



UNIVERSITÀ DEGLI STUDI DI FIRENZE

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

A.A. 2013-2014

# **ANALISI E PROGETTAZIONE DI UN SISTEMA DI MONITORAGGIO PER INFRASTRUTTURE CLOUD**

ANALYSIS AND DESIGN OF A CLOUD INFRASTRUCTURE MONITORING SYSTEM

Relatore

Prof. Paolo Nesi

Prof. Simone Marinai

Candidato

Giuseppe Murtas

Correlatori

Dott. Ing. Daniele Cenni

Dott. Ing. Ivan Bruno

*...a tutti coloro che hanno creduto in me*

# Ringraziamenti

Giunto alla fine della laurea magistrale vorrei rivolgere alcuni ringraziamenti a coloro che mi hanno sostenuto e aiutato. In particolare ringrazio il Prof. Paolo Nesi, l'Ing. Daniele Cenni e l'Ing. Ivan Bruno per la guida, l'infinita disponibilità e il costante aiuto grazie ai quali ho avuto modo di lavorare e appassionarmi a questo progetto.

Vorrei poi ringraziare tutte le persone, che non posso elencare una ad una, ma che mi hanno comunque sostenuto in questo lungo, travagliato e, soprattutto, appassionante percorso.

Infine, ringrazio di cuore la mia ragazza, i miei genitori e tutti i parenti e amici che mi sono stati vicini in questi ultimi mesi, e mi hanno dato la forza per superare i momenti più difficili.

# Indice

<b>RINGRAZIAMENTI .....</b>	<b>2</b>
<b>INDICE .....</b>	<b>3</b>
<b>PREFAZIONE .....</b>	<b>6</b>
<b>I.1. Abstract .....</b>	<b>6</b>
<b>I.2. Struttura della tesi.....</b>	<b>7</b>
<b>I. CLOUD COMPUTING E MONITORAGGIO .....</b>	<b>8</b>
<b>I.1. Introduzione .....</b>	<b>8</b>
<b>I.2. Cloud Computing .....</b>	<b>8</b>
I.2.1. Modelli di erogazione.....	9
I.2.2. Modelli di Servizio .....	10
I.2.3. Vantaggi e svantaggi del Cloud .....	11
<b>I.3. Cloud basati su VMware vSphere.....</b>	<b>13</b>
I.3.1. Gestione delle risorse .....	14
I.3.2. Virtual SMP .....	14
I.3.3. Virtual Machine File System .....	15
I.3.4. vCenter .....	16
I.3.5. vMotion/DRS/HA/FT .....	16
<b>I.4. Il monitoraggio del Cloud.....</b>	<b>17</b>
I.4.1. VMware vSphere Client/Web Client .....	18
I.4.2. Veeam ONE .....	19
I.4.3. VMware vFabric Hyperic.....	21
I.4.4. Riepilogo delle caratteristiche dei sistemi di monitoraggio.....	23
<b>II. ANALISI DELL'APPLICAZIONE .....</b>	<b>26</b>
<b>II.1. Introduzione .....</b>	<b>26</b>
<b>II.2. Analisi dei requisiti.....</b>	<b>26</b>
<b>II.3. Casi d'uso .....</b>	<b>27</b>
<b>III. PROGETTAZIONE DELL'APPLICAZIONE .....</b>	<b>35</b>
<b>III.1. Introduzione .....</b>	<b>35</b>
<b>III.2. Architettura della Webapp .....</b>	<b>35</b>
<b>III.3. La gestione dei thread .....</b>	<b>39</b>



<b>III.4.</b>	<b>Generazione dei grafici .....</b>	<b>43</b>
<b>III.5.</b>	<b>Il database del sistema .....</b>	<b>44</b>
III.5.1.	Descrizione del database .....	44
III.5.2.	Diagramma E-R.....	46
III.5.3.	Le tabelle del database .....	47
III.5.4.	Accesso al database.....	53
<b>III.6.</b>	<b>Le Classi e i Package del sistema (Controller).....</b>	<b>53</b>
III.6.1.	Package tesi.servlet .....	53
<b>III.7.</b>	<b>Le Classi e i Package del sistema (Model).....</b>	<b>55</b>
III.7.2.	Package tesi.db.alarm.....	60
III.7.3.	Package tesi.db.ccrperformance.....	62
III.7.4.	Package tesi.db.condition .....	64
III.7.5.	Package tesi.db.dsperformance .....	66
III.7.6.	Package tesi.db.hsperformance .....	69
III.7.7.	Package tesi.db.inventory .....	72
III.7.8.	Package tesi.db.metric.....	74
III.7.9.	Package tesi.db.vmpformance .....	75
III.7.10.	Package tesi.db.vmsperformance .....	78
III.7.11.	Package tesi.directory.....	80
III.7.12.	Package tesi.mail .....	80
III.7.13.	Package tesi.monitoring .....	81
III.7.14.	Package tesi.security .....	85
III.7.15.	Package tesi.vmware .....	86
III.7.16.	Package tesi.vmware.beans .....	90
III.7.17.	Package tesi.xml .....	98
III.7.18.	Package tesi.xml.beans.....	102
<b>III.8.</b>	<b>Le pagine Jsp del sistema (View) .....</b>	<b>106</b>
III.8.1.	Pagina alarm.jsp.....	106
III.8.2.	Pagina alarmPopup.jsp.....	107
III.8.3.	Pagina ccr.jsp.....	107
III.8.4.	Pagina ccrPerformance.jsp .....	109
III.8.5.	Pagina ccrProperties.jsp .....	110
III.8.6.	Pagina config.jsp.....	110
III.8.7.	Pagina dc.jsp.....	111
III.8.8.	Pagina dcProperties.jsp .....	112
III.8.9.	Pagina ds.jsp .....	113
III.8.10.	Pagina dsPerformance.jsp.....	114
III.8.11.	Pagina dsProperties.jsp.....	115
III.8.12.	Pagina footer.jsp .....	116
III.8.13.	Pagina generalInfo.jsp .....	117
III.8.14.	Pagina header.jsp .....	117
III.8.15.	Pagina home.jsp.....	117
III.8.16.	Pagina hs.jsp.....	118
III.8.17.	Pagina hsPerformance.jsp.....	120
III.8.18.	Pagina hsProperties.jsp.....	120
III.8.19.	Pagina inventarioTree.jsp .....	121
III.8.20.	Pagina login.jsp.....	122
III.8.21.	Pagina loginError.jsp.....	123
III.8.22.	Pagina message.jsp.....	123
III.8.23.	Pagina vm.jsp.....	123
III.8.24.	Pagina vmPerformance.jsp .....	125
III.8.25.	Pagina vmProperties.jsp .....	125

<b>IV. INSTALLAZIONE E CONFIGURAZIONE .....</b>	<b>127</b>
IV.1. <b>Introduzione .....</b>	<b>127</b>
IV.2. <b>Installazione e configurazione dell'Application Server Tomcat 7 .....</b>	<b>127</b>
IV.3. <b>Installazione e configurazione di MySQL 5.6 .....</b>	<b>130</b>
IV.4. <b>Installazione della Web Application .....</b>	<b>135</b>
<b>V. USO SPERIMENTALE E VALIDAZIONE .....</b>	<b>137</b>
V.1. <b>Introduzione .....</b>	<b>137</b>
V.2. <b>L'infrastruttura Cloud .....</b>	<b>137</b>
V.3. <b>Installazione e configurazione del sistema .....</b>	<b>139</b>
V.4. <b>Visualizzazione delle proprietà e delle metriche di performance .....</b>	<b>140</b>
V.4.1.       Datacenter .....	140
V.4.2.       Datastore .....	156
V.4.3.       Cluster .....	163
V.4.4.       Host System .....	175
V.4.5.       Virtual Machine .....	189
V.5. <b>Gestione degli allarmi.....</b>	<b>196</b>
<b>VI. CONCLUSIONI E SVILUPPI FUTURI.....</b>	<b>199</b>
<b>APPENDICE A VMWARE VSPHERE API .....</b>	<b>201</b>
VI.1. <b>Introduzione .....</b>	<b>201</b>
VI.2. <b>vSphere Management APIs.....</b>	<b>201</b>
VI.3. <b>Architettura Client-Server di vSphere.....</b>	<b>202</b>
VI.4. <b>Accesso ai vSphere Server ESX/ESXi e vCenter .....</b>	<b>204</b>
VI.5. <b>Accesso ai Managed Object.....</b>	<b>206</b>
VI.6. <b>Datacenter Inventory .....</b>	<b>208</b>
VI.7. <b>vSphere Performance.....</b>	<b>215</b>
<b>BIBLIOGRAFIA .....</b>	<b>220</b>
<b>INDICE DELLE FIGURE.....</b>	<b>221</b>
<b>INDICE DELLE TABELLE.....</b>	<b>227</b>

# Prefazione

## I.1. Abstract

Negli ultimi anni, grazie all'esponenziale incremento dell'utilizzo dei servizi IT, l'esigenza di poter disporre di sistemi scalabili è diventata sempre più radicata. Inizialmente per eseguire dei semplici calcoli o per immagazzinare i dati bisognava dotarsi di schede perforate ed aspettare il proprio turno per poter usufruire delle risorse di enormi e costosissime unità di elaborazione. Successivamente si passò all'uso di sistemi mainframe con i quali gli utenti interagivano per mezzo di terminali.

Con il boom di Internet iniziò l'era del calcolo distribuito che prevede un'interazione tra molti computer connessi in rete per raggiungere un comune obiettivo. Viene nascosta l'architettura che sta dietro alla rete di computer e all'utente viene presentato un unico sistema in grado di adempiere alle sue necessità.

Grazie all'unione dei concetti di mainframe e di calcolo distribuito nasce il Cloud Computing.

Il Cloud Computing, al giorno d'oggi, è diventato una moda. Con continui riferimenti e pubblicità a prodotti basati sul Cloud, potrebbe sembrare che il paradigma Cloud sia la soluzione, spuntata improvvisamente dal nulla, a tutti i problemi che possono affliggere il comparto ICT.

Può invece essere definito un "rebrand" di tecnologie mature e già ampiamente utilizzate nel passato infatti proprio la fusione tra le tecnologie di virtualizzazione ed il paradigma SOA ha permesso lo sviluppo di applicativi altamente portabili con architettura distribuita, oltre alla possibilità di interconnettere ed utilizzare facilmente software eterogenei ed eventualmente gestiti da altre aziende.

Tra i diversi modelli di Cloud Computing, il modello di erogazione utilizzato maggiormente dalle aziende è il Cloud Privato che garantisce loro un notevole risparmio sui costi IT da sostenere ed un maggiore controllo sui dati.

Per le aziende utilizzatrici dei sistemi di Cloud Privato, mantenere i livelli di servizio desiderati per la propria infrastruttura informatica può risultare complesso e costoso, e non tutte le realtà sono in grado di realizzare il servizio con personale e infrastrutture interne. Al tempo stesso il monitoraggio dei sistemi informatici è imprescindibile per un corretto funzionamento dell'attività aziendale.

Lo stato di salute dei componenti hardware, la saturazione delle risorse di calcolo, l'occupazione dei dischi e il controllo sulla crescita di archivi e database, sono solo alcuni esempi di controlli che possono essere effettuati dai sistemi di monitoraggio. Mantenere sotto controllo la situazione di tutta l'infrastruttura informatica a livello di server, rete e client (ad esempio server che si bloccano, spazio disco su server insufficiente, criticità della RAM e della CPU, consumo energetico degli apparati, ecc) è prioritario per garantire alle aziende la continuità della funzionalità di tutta l'infrastruttura IT.

Una volta individuata la criticità è necessario che si attivi un sistema di alerting via email o SMS al fine di permettere una maggiore rapidità nell'intervento di ripristino e, di conseguenza, una riduzione dei tempi di down dei sistemi. L'utilizzo di soglie di pre-alerting oltretutto permette di evitare, o di ridurre al minimo, l'impatto economico di un'eventuale crash del sistema ed i conseguenti danni economici.

Alla luce delle considerazioni esposte, l'obiettivo della tesi è la progettazione di un sistema di monitoring, reporting e alarming di una infrastruttura Cloud basata su VMware vSphere 5, sistema leader nel mercato della virtualizzazione di sistemi IT.

## **I.2. Struttura della tesi**

La tesi è organizzata nei seguenti capitoli:

### **I Capitolo: Cloud Computing e Monitoraggio**

Nel primo capitolo verranno esposte le definizioni preliminari relative al Cloud Computing, i diversi modelli di erogazione del servizio ed il problema del monitoraggio delle risorse hardware, anche analizzando alcuni sistemi di monitoraggio disponibili sul mercato.

### **II Capitolo: Analisi dell'applicazione**

Nel secondo capitolo verranno analizzati i requisiti che il sistema di monitoraggio deve soddisfare anche in relazione alle funzionalità erogate dai sistemi di monitoraggio esposte nel capitolo precedente. Verranno poi formalizzate le funzionalità dell'applicazione mediante i diagrammi dei casi d'uso.

### **III Capitolo: Progettazione dell'applicazione**

Nel terzo capitolo verrà affrontata la progettazione dell'applicazione, prima dal punto di vista architetturale e successivamente nelle varie componenti costituenti il sistema, con particolare riguardo al database, ai package e alle classi Java ed alle jsp, anche utilizzando i diversi formalismi UML.

### **IV Capitolo: Installazione e Configurazione**

Nel quarto capitolo verrà illustrata la procedura di installazione e configurazione dell'applicazione soffermandosi in particolare sulla configurazione dell'Application Server Tomcat e del DBMS MySQL.

### **V Capitolo: Uso sperimentale e validazione**

Nel quinto capitolo si valuteranno le funzionalità offerte dall'applicazione nel monitoraggio di una infrastruttura Cloud presso l'Università di Cagliari.

### **VI Capitolo: Conclusioni e sviluppi futuri**

Nell'ultimo capitolo verranno analizzati gli obiettivi preposti, l'eventuale raggiungimento degli stessi ed elencate una serie di ulteriori funzionalità che sarebbe utile aggiungere all'applicazione.

# I. Cloud Computing e Monitoraggio

## I.1. Introduzione

In questo capitolo verrà approfondito da un punto di vista tecnico il Cloud Computing ed il problema del monitoraggio, introducendo inizialmente le caratteristiche principali per poi definire quali sono ad oggi i modelli maggiormente riconosciuti e, infine, esporre il problema del monitoraggio delle risorse e le funzionalità di alcuni sistemi di monitoraggio disponibili sul mercato.

## I.2. Cloud Computing

Il Cloud Computing [MG11], conosciuto con l'abbreviazione Cloud, è un modello per condividere un set di risorse computazionali (rete, server, storage, applicazioni ecc.) in maniera conveniente, configurabile, rapida e che necessita di una minima interazione provider-utente finale.

Sfrutta in particolare i vantaggi derivanti dall'utilizzo dalle reti distribuite, utilizza risorse virtualizzate e i protocolli internet.

Affinché si possa parlare di Cloud computing è necessario che:

- l'utente abbia la possibilità, unilaterale, di approvvigionarsi di risorse computazionali automaticamente, senza che ci sia la necessità di una interazione umana con il provider di servizi Cloud.
- le risorse siano accessibili via rete mediante standard che consentono l'utilizzo di piattaforme client eterogenee (smartphone, laptop ecc.).
- le risorse computazionali del provider siano messe in comune per servire più utenti, utilizzando uno schema multi-cliente, che gestisca risorse fisiche e virtuali, dinamicamente, in accordo con le indicazioni degli utenti.
- le risorse siano in grado di essere allocate rapidamente ed in modo flessibile, in alcuni casi automaticamente, per adattarsi, velocemente, alle variabili richieste degli utenti.
- i sistemi controllino automaticamente ed ottimizzino l'utilizzo delle risorse mediante strumenti di misura basati su adeguati livelli di astrazione. L'utilizzo delle risorse deve poter essere monitorato, controllato ed elaborato, sia dal provider del servizio sia dagli utenti finali.

Citando l'ENISA (European Network and Information Security Agency) il Cloud Computing [ENI12] è un nuovo modo di erogare servizi IT, non una nuova tecnologia. La flessibilità e l'efficienza delle architetture Cloud sono rese possibili da meccanismi ormai consolidati quali:

- **Astrazione:** il Cloud Computing astrae i dettagli dell'implementazione a utenti e sviluppatori; infatti le applicazioni sono in funzione su un sistema fisico che non è specificato; i dati sono contenuti in luoghi non specificati; l'amministrazione del sistema è affidata ad altri e gli utenti vi accedono ovunque.

- **Virtualizzazione:** il Cloud Computing virtualizza i sistemi attraverso un pooling di risorse condivise. I sistemi possono essere distribuiti da una infrastruttura centralizzata. I costi sono stimati in base ai consumi; è attivo il multi-tenancy e le risorse sono facilmente scalabili.

Il questa sezione verranno analizzate le due diverse classi di Cloud: quelli basati sui modelli di erogazione, che si riferiscono alla gestione e locazione dell'infrastruttura, e quelli basati sui modelli di servizio, che consistono in particolari tipi di servizi accessibili da una piattaforma Cloud.

Questa distinzione è stata introdotta dal NIST (National Institute of Standards and Technology) nel novembre 2009 per poi giungere alla versione definitiva, nell'ottobre 2011, dopo 15 revisioni. Originariamente infatti, il modello non richiedeva al Cloud di usare un pool di risorse che supportasse il multi-tenancy, proprietà invece richiesta nella definizione finale.

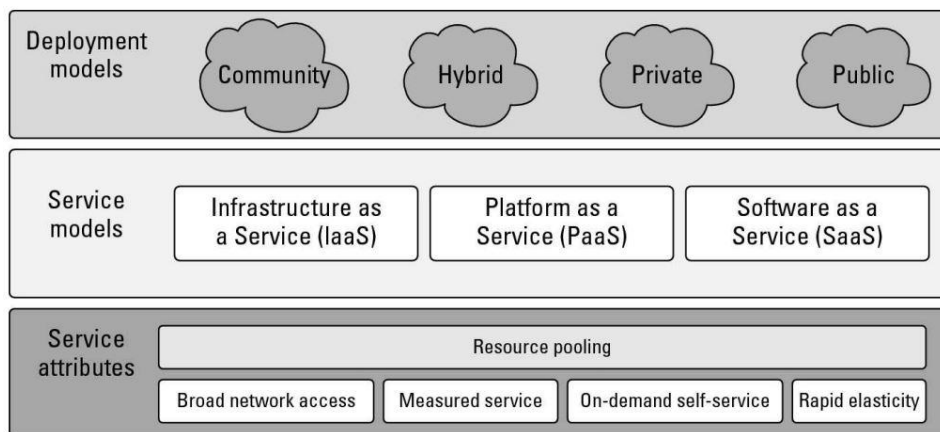


Figura I-1: Modelli di Cloud Computing

### I.2.1. Modelli di Erogazione

Sempre richiamando la definizione proposta dal NIST, i sistemi Cloud, secondo i modelli di erogazione, si dividono in:

- **Cloud Pubblico:** disponibile per un uso pubblico o per un grande gruppo industriale; il Cloud pubblico è gestito da un provider che offre servizi Cloud ed è accessibile attraverso internet. I servizi possono essere offerti in maniera gratuita, con set di risorse limitate, oppure a consumo. Tra i vantaggi troviamo: una gestione semplice che non prevede un esborso di denaro in quanto hardware, applicazioni e costi di connessione sono a carico del provider, una scalabilità del sistema senza limiti e una migliore gestione delle risorse derivante dal modello on-demand.
- **Cloud Privato:** è chiamato anche Cloud interno o Cloud aziendale. Può essere gestito dall'impresa stessa o da una di terze parti ed essere ubicata in sede o fuori sede. Offrendo servizi di hosting ad un numero limitato di utenti protetti da sistemi firewall, viene incontro alle organizzazioni che hanno bisogno di un maggiore controllo sui propri dati rispetto a quello che possono ottenere utilizzando un servizio di terze.
- **Cloud Ibrido:** combina Cloud privati e pubblici, dove ognuno mantiene la propria identità nonostante sia unito agli altri. Può offrire un accesso ai dati ed alle applicazioni di tipo

standardizzato o proprietario e viene generalmente offerto con due soluzioni: un provider dispone di un Cloud privato e costituisce una partnership con un provider che gestisce un Cloud pubblico, o viceversa, un provider che gestisce Cloud pubblici costituisce una partnership con un provider che fornisce piattaforme su Cloud privati. Un Cloud Ibrido è un ambiente di Cloud computing in cui l'organizzazione fornisce e gestisce alcune risorse in-house mentre altre vengono esternalizzate. Idealmente l'approccio ibrido consente alle aziende di trarre vantaggio dalla scalabilità e dal rapporto costo-beneficio che un ambiente di Cloud pubblico offre senza esporre applicazioni critiche e dati privati all'accesso di terze parti, rendendoli quindi meno vulnerabili.

- **Cloud Comunitario:** l'infrastruttura è condivisa da più organizzazioni e supporta una comunità specifica che ne condivide alcuni ambiti (ad esempio, la mission, i requisiti di sicurezza, le policy e le considerazioni di conformità). Esso può essere gestito dalle organizzazioni stesse o da una terza parte e può essere on-premise o off-premise.

Un Cloud privato quindi differisce da uno pubblico non solo per il fatto che viene usato abitualmente sfruttando reti interne, garantendo quindi un maggior livello di sicurezza, ma anche perché viene utilizzato da centinaia a migliaia di nodi, anziché da decine di migliaia come avviene in quello pubblico.

Si contrappone un approccio single-tenancy del Cloud privato a quello multi-tenancy del Cloud pubblico e il modello di pricing si basa sul concetto di capacità utilizzata e non su quello di uso.

Il Cloud ibrido invece include caratteristiche di entrambe: un'azienda potrebbe decidere di mantenere nel Cloud privato applicazioni del core business e dati sensibili, mentre in quello pubblico i servizi "non core" meno vulnerabili dal punto di vista della sicurezza.

### I.2.2. Modelli di Servizio

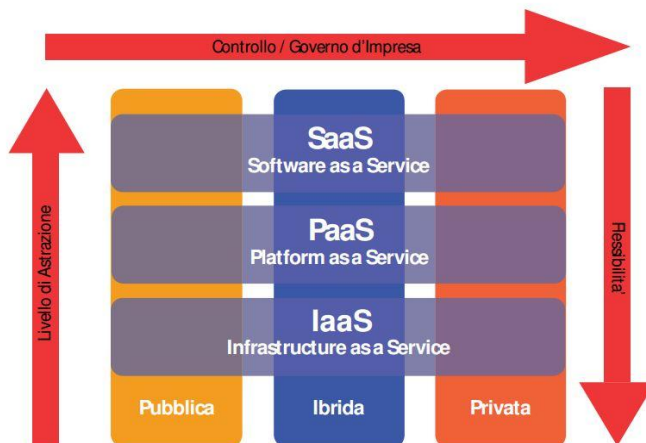


Figura I-2: Modello SPI

Nei modelli di servizio le infrastrutture Cloud sono una espressione della tipologia di infrastruttura che viene erogata. Differenti provider offrono Cloud con differenti servizi associati. Tutti i modelli di servizio assumono la forma: XaaS (<Something> as a Service).

I tre modelli descritti nel modello NIST, conosciuti come "modello SPI" sono:

- **Infrastructure as a Service:** mette a disposizione virtual machine, storage virtuale, infrastrutture virtuali e altre risorse hardware di cui i clienti possono usufruire. Il provider che fornisce il servizio IaaS gestisce tutta l'infrastruttura mentre il cliente si occupa del suo sviluppo che include:

il sistema operativo, le applicazioni e le interazioni utente-macchina. Le sue caratteristiche includono: un servizio di tipo “Utility Computing”, gestione automatica delle attività di tipo amministrativo, scalabilità, virtualizzazione dei desktop, connettività internet.

- **Platform as a Service:** offre virtual machine, sistemi operativi, applicazioni, servizi, framework di sviluppo e strutture di controllo. Il cliente può sviluppare le sue applicazioni utilizzando l'infrastruttura Cloud o le applicazioni programmate per mezzo di linguaggi e strumenti supportati dal provider. Il provider gestisce l'infrastruttura Cloud, i sistemi operativi ed il software di accesso. Il cliente invece installa e gestisce le applicazioni che sviluppa. Tra i vantaggi troviamo: la possibilità di aggiornare o cambiare frequentemente le caratteristiche del sistema operativo; diversi team di sviluppo, distribuiti geograficamente, possono lavorare insieme ad un progetto; i servizi possono essere ottenuti da diverse fonti che attraversano i confini nazionali. I costi iniziali e di mantenimento possono essere ridotti grazie all'utilizzo dei servizi di un'infrastruttura gestita da un solo gestore rispetto all'utilizzo di servizi offerti da diversi hardware che spesso soffrono di problemi di compatibilità. Di contro si potrebbe incappare nel rischio di “lock-in” quando si usano servizi o linguaggi di sviluppo proprietari.
- **Software as a Service:** è un ambiente completamente operativo con applicazioni, sistema gestione e interfaccia utente. Le applicazioni sono offerte al cliente attraverso una interfaccia (solitamente un web browser) e la competenza dell'utente è limitata all'inserimento, alla gestione e all'interazione dei dati. Tutto ciò che utilizza a partire dall'applicazione per arrivare all'infrastruttura è a carico del provider. Tra i benefici di una soluzione di questo genere troviamo: una semplice amministrazione, aggiornamenti automatici, un sistema di patch automatici forniti dal provider del servizio e una totale compatibilità derivante dall'utilizzo della stessa versione del software per tutti gli utenti del sistema, quindi una facile collaborazione e un'accessibilità globale garantita dall'utilizzo dei servizi web.

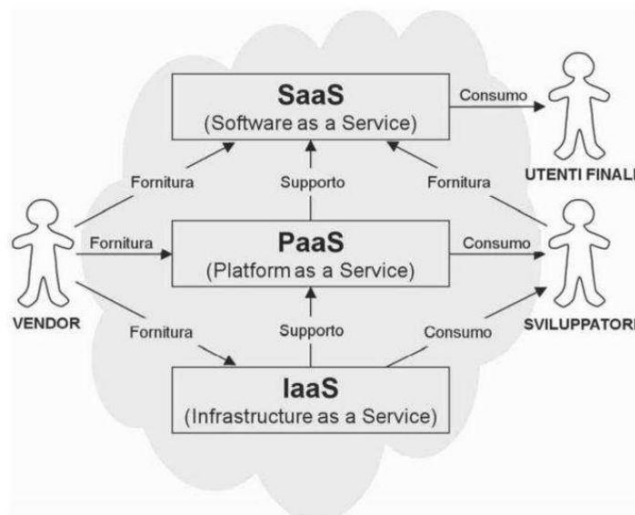


Figura I-3: Confronto modelli SPI

### I.2.3. Vantaggi e svantaggi del Cloud

**Vantaggi:**



Per quanto riguarda i benefici derivanti dall'utilizzo di un modello Cloud, riprendiamo la definizione di Cloud Computing del NIST nella quale sono descritte le cinque principali caratteristiche:

- **Richiesta Self-service:** un utente può gestire le risorse computazionali senza la necessità di interagire con il personale del provider.
- **Ampio accesso dalla rete:** l'accesso al Cloud avviene dalla rete ed è garantito da una moltitudine di piattaforme differenti quali smartphone, laptop, ecc.
- **Pool di risorse:** i sistemi fisici e virtuali sono allocati e riallocati dinamicamente quando necessario. Le macchine fisiche vengono nascoste all'utente al quale vengono invece presentate come: virtual machine, storage, memoria, capacità computazionale, connettività ecc.
- **Flessibilità:** in ogni momento si può richiedere un upgrade o un downgrade del sistema in base alle esigenze del cliente che può acquistare risorse in qualunque momento e in ogni quantità.
- **Servizio Misurato:** il provider fornisce dei tagli specifici di ogni servizio offerto in modo che il cliente possa sempre far riferimento ad una metrica conosciuta, di solito definita come: quantità di storage usato, numero di transazioni, banda, capacità computazionale utilizzata, ecc.

A queste cinque caratteristiche principali si possono affiancare altre caratteristiche quali:

- **Costi Ridotti:** grazie alla maggiore efficienza delle infrastrutture Cloud spesso si ha una forte riduzione dei costi totali.
- **Comodità d'uso:** in relazione al servizio che viene offerto si potrebbe riuscire a implementare il servizio senza la necessità di utilizzare hardware o software con licenza.
- **Qualità del servizio:** La QoS viene offerta per contratto dal provider.

Normalmente i maggiori vantaggi nell'adozione del Cloud si ritrovano nelle piccole organizzazioni rispetto a quelle più grandi, perché queste ultime possono dotarsi di un personale IT interno all'azienda che sviluppa in loco soluzioni software adatte alle proprie necessità.

### Svantaggi:

I benefici del Cloud computing sono innumerevoli, ma presenta anche alcuni svantaggi.

Le applicazioni e i servizi offerti *in the Cloud* spesso non sono personalizzabili secondo le proprie esigenze e sono limitanti rispetto ad un software sviluppato in-house.

Un altro problema del Cloud è quello della latenza legato alla connettività: se la propria applicazione richiede il trasferimento di una grande mole di dati, probabilmente il Cloud non è la scelta migliore da adottare poiché i benefici portati verrebbero in gran parte offuscati dai tempi legati al trasferimento dati.

Se il Cloud si estende a più nazioni allora potrebbe sorgere il problema delle differenti giurisdizioni. Infine le tematiche sulla privacy e sulla sicurezza non devono essere sottovalutate.

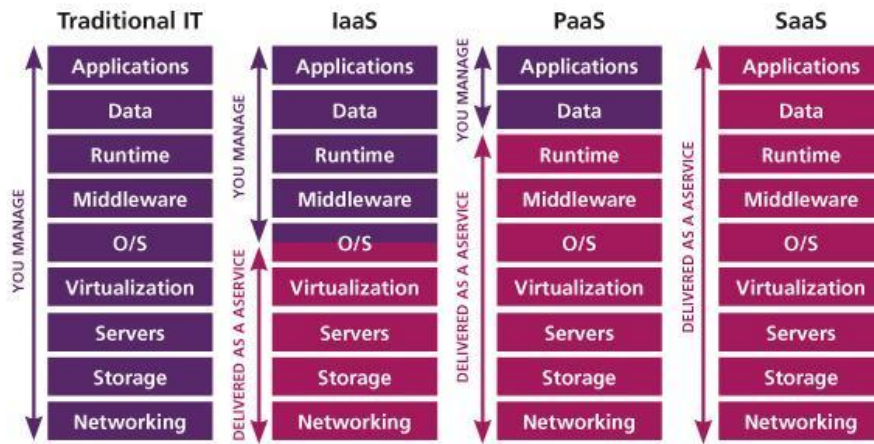


Figura I-4: Sfruttamento dei vantaggi economici del Cloud

### I.3. Cloud basati su VMware vSphere

VMware vSphere [VMW11] è il nome della suite di programmi e funzionalità fornite da VMware per la virtualizzazione di infrastrutture Cloud. Tra i prodotti e le funzionalità che la compongono troviamo:

- ESX e ESXi
- Virtual Symmetric Multi-Processing (vSMP)
- Virtual Machine File System (VMFS)
- vCenter
- vMotion
- Distributed Resource Scheduler (DRS)
- High Availability (HA)
- Fault Tolerance (FT)

ESX ed ESXi [VMW09] [VMW08] sono entrambi hypervisor di tipo nativo, condividono lo stesso motore di virtualizzazione e forniscono lo stesso insieme di funzionalità ad esclusione della Service Console. ESXi infatti non dispone di Service Console, una console basata su Linux, che fornisce un insieme di comandi per interagire con il vmkernel. L'hypervisor a sua volta è suddiviso in due componenti: Virtual Machine Monitor e vmkernel. Il vmkernel è il cuore del sistema e gestisce le principali funzioni del sistema di virtualizzazione come la pianificazione della CPU (scheduling), la gestione della memoria di elaborazione dati provenienti dai virtual switch. Ogni virtual machine viene gestita da un Virtual Machine Monitor separato, situato all'interno dell'hypervisor, che ne soddisfa le richieste di CPU, memoria e I/O.

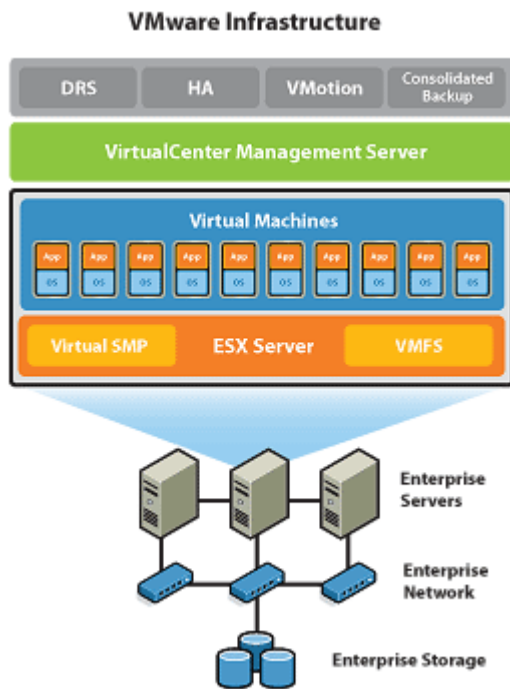


Figura I-5: Infrastruttura di VMware vSphere

### I.3.1. Gestione delle risorse

ESXi e ESX consentono di gestire e limitare la quantità di risorse da assegnare alle virtual machine ospitate al loro interno attraverso l'utilizzo dei seguenti meccanismi:

- **Reservation:** consente di riservare un quantitativo minimo di risorse senza le quali una virtual machine non può essere avviata. Nel caso della RAM questa deve essere fisicamente presente e non può essere fornita attraverso swap su disco.
- **Limit:** consente di limitare l'utilizzo delle risorse quali CPU e RAM.
- **Share:** parametro concettualmente posto tra Reservation e Limit; indica una percentuale che rappresenta la priorità di accesso alle risorse assegnata ad una virtual machine.

### I.3.2. Virtual SMP

Attraverso Virtual SMP [VMW08] una virtual machine ha la possibilità di sfruttare più processori fisici per portare a termine task particolarmente pesanti. Dalla versione ESX 2.1 è possibile sfruttare l'Hyper-Threading [VMW11c]. L'Hyper-Threading è una tecnologia Intel che permette di sfruttare un processore per l'esecuzione contemporanea di due thread distinti, duplicando al suo interno alcune unità di elaborazione maggiormente utilizzate, al fine di poter eseguire simultaneamente alcune operazioni grazie a tecniche di multithreading. Un singolo core è in grado di gestire due thread in contemporanea; quando le istruzioni di un thread rimangono bloccate nella pipeline il processore procede ad elaborare un secondo thread al fine di mantenere le unità di elaborazione sempre attive.

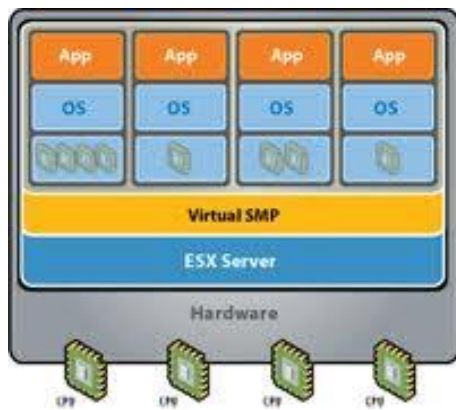


Figura I-6: vSMP esempio di assegnazione dei processori alle macchine virtuali

Abilitando l'Hyper-Threading vSphere permetterà di vedere un quantitativo di processori logici da assegnare alle macchine virtuali pari al doppio di quelli fisici.

### I.3.3. Virtual Machine File System

Dal punto di vista del file system una virtual machine è costituita da un insieme di file: disco, swap, configurazione ecc. Questi file possono essere allocati localmente sui server ESX oppure in maniera centralizzata utilizzando sistemi SAN (Storage Area Network) e NAS (Network Attached Storage). Quando lo spazio dove risiedono le virtual machine è centralizzato, è necessario un meccanismo di concorrenza. Tale meccanismo è garantito dall'utilizzo del file system concorrente VMFS [VMW09c] [VMW08]. Attraverso questo file system, più server ESX possono accedere in maniera concorrente alle risorse presenti all'interno di uno stesso spazio di memorizzazione condiviso. Un sistema di locking sui metadati permette ad un solo server ESX di avviare una virtual machine; in caso di guasto di una virtual machine viene gestito in maniera automatica un sistema di rilascio del lock per sbloccare le risorse. VMFS permette la creazione di file fino a una soglia dipendente dalla versione utilizzata. VMFS è la base di partenza per le funzionalità più avanzate quali vMotion e DRS. VMFS fornisce anche funzionalità e meccanismi di consistenza e recovery tipici di un file system distribuito di classe enterprise.

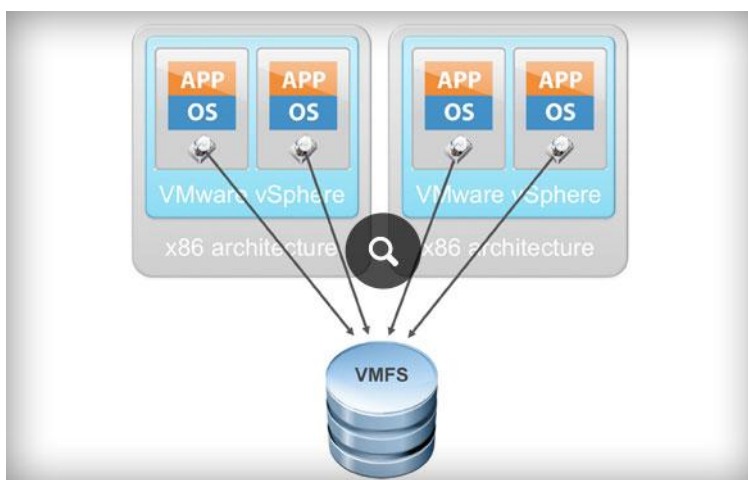


Figura I-7: Architettura VMFS

### I.3.4. vCenter

Attraverso vCenter [VMW11] [VMW08] è possibile gestire in maniera centralizzata i diversi aspetti relativi all'infrastruttura. vCenter è un componente necessario per poter usufruire degli altri componenti che forniscono funzionalità avanzate quali Fault Tolerance, DRS, vMotion Storage ecc.

In particolare vCenter consente:

- La gestione in maniera centralizzata dei componenti dell'infrastruttura e di avere un sistema centralizzato di autenticazione.
- Il monitoraggio delle risorse (CPU, storage, memoria) dei componenti dell'infrastruttura quali virtual machine, host, cluster e datastore.
- La pianificazione di operazioni di management e l'inserimento di allarmi.
- L'aggiunta di nuove virtual machine mediante l'utilizzo dei *template*.
- La creazione di cluster di server nei quali le risorse globali sono la somma delle risorse di ogni host. Ogni virtual machine appartenente ad un cluster potrà utilizzare tutte le risorse del cluster in maniera trasparente.
- La creazione di insiemi di risorse (resource pool) mediante le quali è possibile specificare un limite massimo di risorse per ogni insieme definito. Le virtual machine all'interno di un pool potranno utilizzare esclusivamente tali risorse.

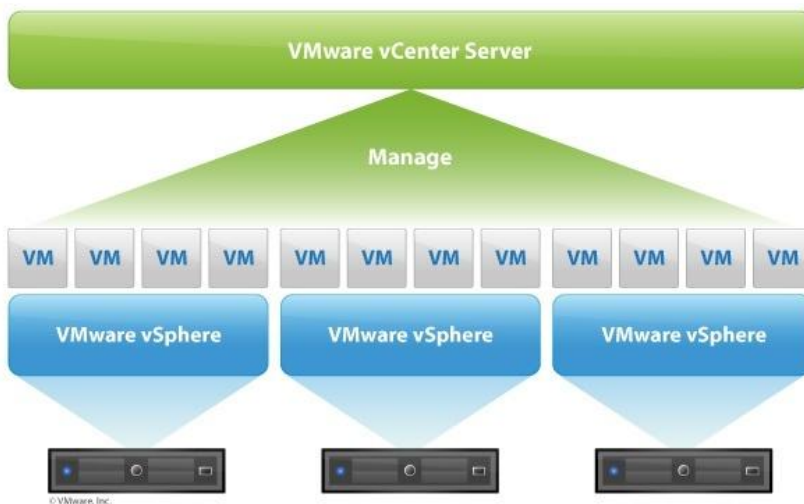


Figura I-8: Architettura VMware vCenter

### I.3.5. vMotion/DRS/HA/FT

**vMotion** [VMW11e] consente di spostare una virtual machine da un host ad un altro senza fermare la stessa. vMotion consente di spostare una o più virtual machine da uno storage a un altro senza interruzioni. La caratteristica principale è la possibilità di modificare le caratteristiche delle virtual machine “a caldo”, cioè a macchina accesa. In sostanza è possibile aggiungere RAM, CPU e hardware, dischi compresi, senza prima spegnere la virtual machine. Questa caratteristica deve essere supportata dal sistema operativo e comunque non è possibile sottrarre risorse ad una virtual machine accesa.

Di seguito le caratteristiche principali:

- Assegnazione di hardware fisico alle virtual machine.
- Virtual Switch distribuito, plugin di terze parti come software Cisco.
- Vshield zones: Creare delle aree tipo DMZ a cui assegnare regole di firewalling; le virtual machine inserite in queste aree assumeranno in automatico le regole di firewalling.
- License server non più necessario.
- Thin provisioning (virtual machine con dischi dinamici, senza assegnazione dell'intero spazio occupato dal disco).
- Modifica delle dimensioni del disco a virtual machine accesa. Incremento della dimensione del datastore a virtual machine accese.

**HA (High Availability)** [VMW11b]: permette il riavvio automatico della virtual machine su un altro host, in caso di crash dell'host che la gestiva. Possibilità di monitorare la virtual machine per intercettare eventuali crash del sistema operativo.

**DRS (Distributed Resource Scheduler)** [VMW11c]: Gestione automatica del carico e gestione avanzata del risparmio energetico. vSphere può gestire in automatico il bilanciamento del carico sui vari host, spostando le virtual machine su host meno carichi. E' inoltre possibile programmare lo spegnimento di alcuni host e lo spostamento automatico delle virtual machine su altri host, in modo che l'infrastruttura consumi meno energia elettrica.

**FT (Fault Tolerance)** [VMW11b]: "Mirroring" di una virtual machine. Abilitando questa funzione, vSphere duplicherà la virtual machine su un altro host, tenendole accese entrambe. La seconda delle quali non viene chiamata in causa durante l'utilizzo normale, ma è una vera e propria copia funzionante della prima. In caso di crash dell'host che ospita la prima, i servizi continuano a funzionare regolarmente poiché la seconda è già accesa e funzionante. Difetto di questa caratteristica rispetto all'HA è la duplicazione dell'intera virtual machine, per cui doppia occupazione di spazio. Se il server che ospita la virtual machine va giù e quindi la seconda lavora al posto della prima, vSphere individua in automatico un altro host su cui copiare la virtual machine per continuare ad avere la fault tolerance.

### I.4. Il monitoraggio del Cloud

Per mantenere i livelli di servizio desiderati il monitoraggio degli elementi del Cloud è un fattore imprescindibile, innanzitutto per un corretto funzionamento dell'attività aziendale.

Lo stato di salute dei componenti hardware, la saturazione delle risorse di calcolo, l'occupazione dei dischi e il controllo sullo spazio disponibile nei datastore, sono solo alcuni esempi di controlli che possono essere effettuati dai servizi di monitoraggio. Mantenere sotto controllo lo stato di tutta l'infrastruttura è fondamentale: ad esempio server che saturano le risorse di calcolo, spazio su disco insufficiente, picchi nell'utilizzo della RAM sono elementi critici per cui è prioritario garantire il monitoraggio degli stessi al fine di prevenire i blocchi dell'intera infrastruttura IT.

Una volta individuata la criticità è necessario che si attivi un sistema di alerting via email o SMS al fine di permettere una maggiore rapidità nell'intervento di ripristino e, di conseguenza, una riduzione dei tempi di down dei sistemi. L'utilizzo di soglie di pre-alerting oltretutto permette di evitare, o di ridurre al minimo, l'impatto economico di un'eventuale crash del sistema.

Nei seguenti paragrafi verranno esposte le caratteristiche e le peculiarità dei sistemi di monitoraggio di infrastrutture Cloud presenti sul mercato, con particolare attenzione a quelli orientati ai Cloud basati su VMware vSphere.

### **I.4.1. VMware vSphere Client/Web Client**

VMware vSphere Client [VMW11] è il prodotto lato client fornito da VMware per l'accesso alle funzionalità dei server vCenter e degli host ESXi (accesso disponibile solo con la versione Client e non Web client). vSphere Client utilizza le VMware API per accedere al vCenter Server e agli host ESXi. Dopo la fase di autenticazione con il server inizia una sessione e l'utente ha visibilità di tutte le risorse disponibili nell'infrastruttura. Mediante la versione Web browser l'utente non ha la necessità di installare il cliente software specifico potendo utilizzare un browser web con l'aggiunta di alcuni plugin quali flash player.

#### **Architettura**

vSphere Client utilizza una architettura client/server essendo il client di default dell'infrastruttura, anche nella versione web browser. La gestione degli utenti è quella nativa di vCenter per cui ad ogni utente dell'infrastruttura è assegnato un set di risorse sulle quali può operare.

#### **Monitoraggio delle risorse**

Per mezzo del client e del web client è possibile accedere al tab performance contenente tutti i parametri del datacenter anche in forma grafica. I set di parametri osservati differiscono in base al tipo di risorsa e sono automaticamente monitorati durante l'arco di funzionamento del software e dal momento in cui un utente aggiunge un componente all'infrastruttura. Nessuna operazione è richiesta per avviare il monitoraggio di un singolo parametro. Il software consente il monitoraggio [VMW11d] delle seguenti risorse:

- Virtual Machine
- Host ESX(i)
- Resource Pool
- Cluster
- Datastore
- Datacenter

Al fine di evitare che il database del sistema cresca troppo velocemente il sistema prevede che i parametri monitorati vengano memorizzati nel vCenter Server database, andando però a perdere di risoluzione via via che si va indietro nel tempo. Periodicamente quindi un algoritmo di vCenter si occupa di ridurre la quantità di dati di performance memorizzati. Ad esempio i parametri monitorati un anno prima vengono conservati con campionamento giornaliero. Più precisamente la situazione è la seguente:

Interval Duration	Save For
5 Minutes	1 Day
30 Minutes	1 Week
2 Hours	1 Month
1 Day	1 Year

**Tabella I-1: Campionamento dei dati di performance storici di VMware vSphere**

Su queste risorse il software è in grado di elaborare alcune statistiche quali ad esempio le virtual machine che in un determinato arco di tempo consumano più risorse.

In aggiunta è presente un sistema di *Alerting* che si occupa di confrontare lo stato dei parametri del sistema con quelli impostati dall'amministratore. In caso di allarme, il sistema può essere configurato per l'invio di una notifica email. Gli allarmi possono essere applicati ai componenti monitorati.

E' disponibile anche un semplice sistema di *Reporting* che consente la generazione di report grafici, riferiti a specifici archi temporali, sui alcuni parametri monitorati dal software.

#### I.4.2. Veeam ONE

Veeam One [VEE13] è un prodotto software per il monitoraggio delle infrastrutture Cloud basate su VMware vSphere e Microsoft Hyper-V. Il prodotto prevede una versione freeware con limitazioni e una completa a pagamento.

Le maggiori limitazioni della versione freeware sono le seguenti:

- L'opzione di installazione distribuita non è disponibile.
- La console di monitoraggio mostra i dati degli ultimi 7 giorni ed i report sulle prestazioni sono limitati alle 24 ore precedenti.
- Opzioni di personalizzazione e-mail non disponibili, i messaggi inviati non includono tutti i dettagli sull'allarme, quali l'orario e i componenti oggetti interessati.
- Resoconto dei dati di performance di CPU, memoria, rete e heartbeat per host, virtual machine e datastore limitato alle 24 ore precedenti.
- Reportistica sulle virtual machine sottoutilizzate o sovrautilizzate limitato a 3 virtual machine.

#### Architettura

Veeam One implementa una architettura client/server ed è supportato da un modulo di gestione degli utenti mediante il quale l'amministratore è in grado di autorizzare l'accesso alle risorse in modo distinto per ogni utente.



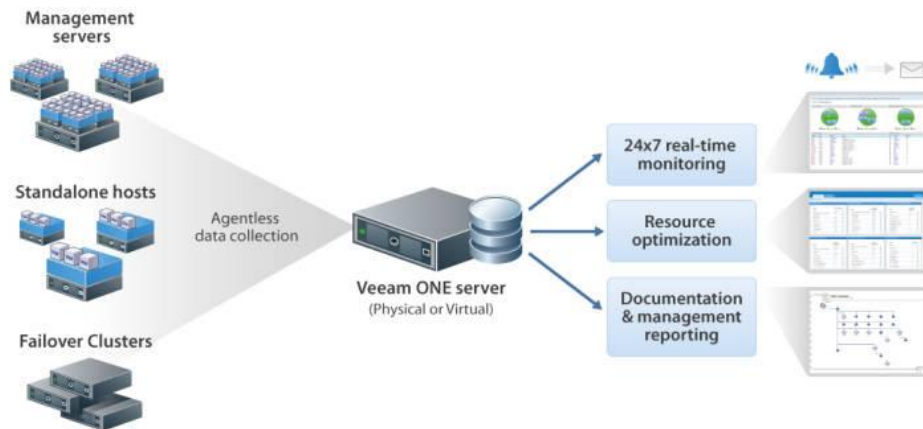


Figura I-9: Architettura Veeam One

### Monitoraggio delle risorse

Il software consente il monitoraggio di tutti i parametri del datacenter mediante grafici 2d e mediante tabelle. I set di parametri osservati differiscono in base al tipo di risorsa e sono automaticamente monitorati durante l'arco di funzionamento del software e dal momento in cui un utente aggiunge un componente all'infrastruttura. Nessuna operazione è richiesta per avviare il monitoraggio di un singolo parametro. Il software consente il monitoraggio delle seguenti risorse:

- Server vCenter
- Host ESX(i)
- Resource Pool
- Virtual Machine
- Datastore

Su queste risorse il software è in grado di elaborare alcune statistiche quali ad esempio le virtual machine che in un determinato arco di tempo consumano più risorse o quali host risultano sotto utilizzati.

In aggiunta a queste caratteristiche è presente un sistema di *Alerting* che si occupa di confrontare lo stato dei parametri del sistema con quelli impostati dall'amministratore. In caso di allarme il sistema può essere configurato per l'invio di una notifica email o per l'invio di messaggi SNMP. Gli allarmi possono essere applicati agli stessi componenti monitorati.

Un'altra caratteristica utile è la possibilità di testare gli allarmi con i dati storici presenti nel database al fine di conoscere orientativamente l'efficacia dell'allarme.

E' disponibile anche un sistema di *Reporting* che consente la generazione di report grafici, riferiti a specifici archi temporali, sui parametri monitorati dal software. Tali report possono essere programmati ed inviati via email oppure visualizzati sul client in tempo reale. I report possono essere esportati nel formato html o xls.

### I.4.3. VMware vFabric Hyperic

VMware vFabric Hyperic [VMW13] è un componente del prodotto VMware vCenter Operations Management Suite, incluso nelle edizioni Advanced ed Enterprise. Disponibile anche come sistema stand-alone. Consente di monitorare, analizzare e controllare tutti i componenti di un datacenter comprese le applicazioni in funzione all'interno delle macchine virtuali.

#### Architettura

VMware vFabric Hyperic implementa una architettura client/server mediante la quale più utenti, con credenziali proprie, possono accedere alle funzionalità offerte.

I componenti principali del software sono:

- **HQ Agent** che si occupano di recuperare ed inviare i valori dei parametri monitorati dalle virtual machine su cui sono installati. Gli Agent vanno installati sia sui server vCenter sia sulle virtual machine.
- **HQ Server** che si occupa di ricevere i dati dagli HQ Agent e memorizzarli nel database di HQ.

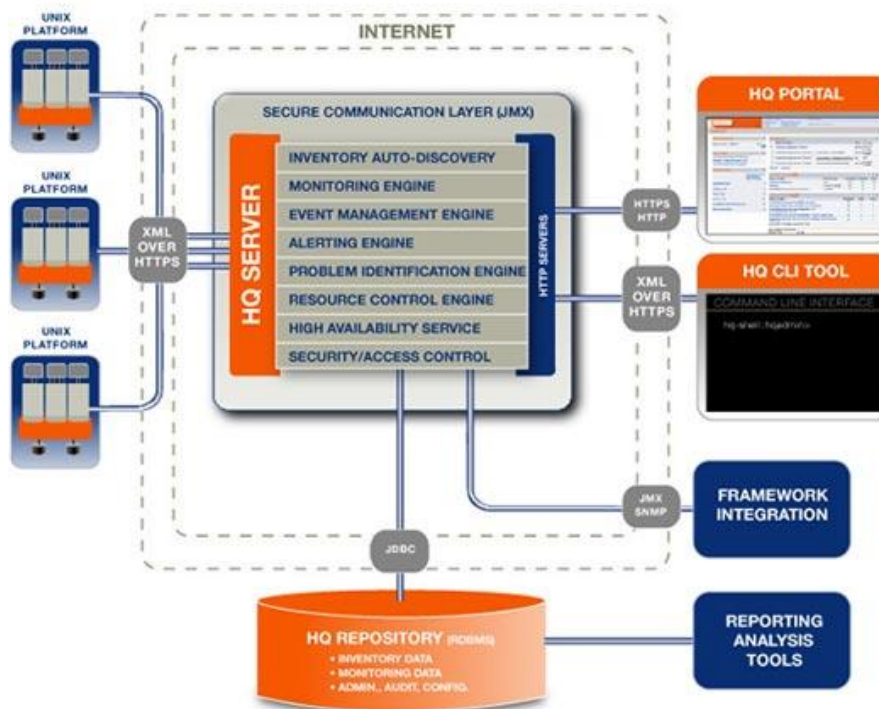
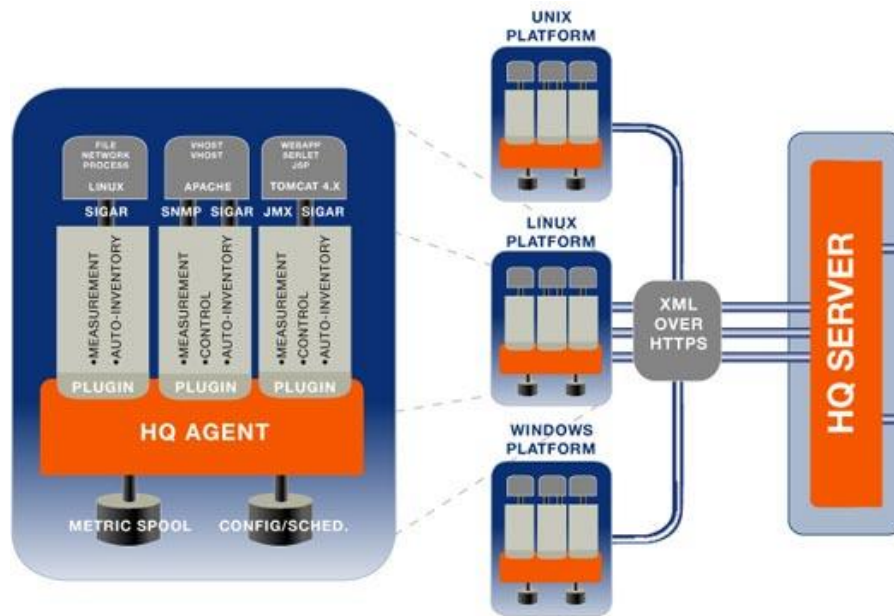


Figura I-10: Architettura Hyperic Server



**Figura I-11: Architttura Hperic Agent**

Insieme al software è inoltre distribuito un plugin di sviluppo mediante il quale è possibile sviluppare Agent personalizzati per componenti hardware non ancora supportati o per estendere quelli esistenti. E' inoltre possibile estendere l'interfaccia grafica e sviluppare script in grado di automatizzare le procedure più frequenti o interfacciare HQ con software terzi.

### Monitoraggio delle risorse

Le caratteristiche principali per quanto riguarda il monitoraggio sono:

- Analisi prestazionale e di utilizzo delle virtual machine.
- Analisi delle prestazioni della rete (es. pacchetti trasmessi e ricevuti ecc.).
- Monitoraggio hardware degli host (es. utilizzo della CPU, della RAM, ecc.).
- Monitoraggio delle risorse hardware delle virtual machine (es. CPU disponibile e usata, memoria primaria virtuale ecc. ).
- Monitoraggio dei componenti di storage (es. spazio disponibile e utilizzato ecc. ).
- Storico dei parametri monitorati e degli allarmi generati.

E' disponibile un sistema di Alarming mediante il quale, impostando delle opportune regole per un componente o un set di essi, informare gli utenti dei problemi che si stanno verificando nell'infrastruttura. Quando un alert è generato, a seconda della configurazione, viene eseguita una o più delle seguenti azioni:

- Presentazione a video dell'alert.
- Notifica via SMS, email, NMs Trap.
- Esecuzione di uno script personalizzato in grado di inviare una notifica ad altri eventuali software.

- Invio di una serie di notifiche, per un determinato periodo di tempo, ad un gruppo di utenti finché l'allarme non viene riconosciuto.

E' disponibile inoltre la possibilità di creare report grafici sui parametri monitorati dal software. Questi report possono essere programmati per l'invio agli utenti via email o visualizzati nei client.

#### I.4.4. Riepilogo delle caratteristiche dei sistemi di monitoraggio

Per quanto riguarda le caratteristiche generali dei software di monitoraggio esposti abbiamo la situazione rappresentata nella seguente tabella:

Software/Caratteristiche	vSphere Client/Web Client	Veeam ONE	VMware vFabric Hyperic
<b>Architettura Client/Server</b>	si	si	si
<b>Accesso via Web browser</b>	Si (nella versione web client)	no	si
<b>Gestione utenti</b>	si	si	si
<b>Assegnazione risorse agli utenti</b>	si	si	si
<b>Sonde software</b>	no	no	si
<b>Autodiscovery dei componenti</b>	si	si	si
<b>Plugin development</b>	si	no	si

Tabella I-2: Comparazione delle caratteristiche generali dei software

Dove le caratteristiche riportate rappresentano rispettivamente:

- **Architettura Client/Server:** indica l'utilizzo di una architettura di tipo Client/Server in cui il Server si occupa di monitorare i componenti del Cloud ed il client di mostrare agli utenti le funzionalità offerte attraverso una GUI.
- **Accesso via Web browser:** indica la possibilità di accedere ai servizi forniti dal server per mezzo di un web browser.
- **Gestione utenti:** possibilità di creazione di più utenti per l'accesso al software di monitoraggio;
- **Assegnazione risorse agli utenti:** nel caso di supporto di più utenti indica la capacità di assegnare il monitoraggio dei componenti in maniera distinta agli utenti.
- **Sonde software:** indica la necessità di installare sistemi software, dette sonde o agent, per il monitoraggio dei componenti.

- **Autodiscovery dei componenti:** indica la capacità di riconoscere la struttura e i componenti dell'infrastruttura.
- **Plugin development:** Indica la possibilità del software di aggiungere nuove funzionalità per mezzo di plugin.

Per quanto riguarda le caratteristiche di monitoraggio, il riepilogo delle funzionalità offerte dai sistemi esposti è riassunto nella seguente tabella:

<b>Software/Caratteristiche</b>	<b>vSphere client/vSphere server (ESXi/vCenter)</b>	<b>Veeam ONE</b>	<b>VMware vFabric Hyperic</b>
<b>Monitoraggio risorse hardware per VM</b>	si	si	si
<b>Monitoraggio risorse hardware per HS</b>	si	si	si
<b>Monitoraggio risorse hardware per CCR</b>	si	si	no
<b>Monitoraggio risorse hardware per DS</b>	si	si	no
<b>Monitoraggio risorse hardware per DC</b>	si	si	no
<b>Statistiche sulle performance (max/min nel periodo)</b>	no	si	si
<b>Statistiche sulle performance (valore medio/deviazione standard nel periodo)</b>	no	no	Solo valore medio
<b>Visualizzazione proprietà degli elementi</b>	si	si	si
<b>Sistema di Alerting</b>	si	si	si
<b>Notifica Email/Sms</b>	si	si	si
<b>Sistema di Reporting per elemento</b>	si	si	si
<b>Reporting con integrazione degli elementi associati</b>	parzialmente	parzialmente	parzialmente
<b>Database con storico dati</b>	parzialmente	si	parzialmente

Tabella I-3: Comparazione delle caratteristiche di monitoraggio dei software

dove le caratteristiche riportate rappresentano rispettivamente:

- **Monitoraggio risorse hardware delle VM:** indica la capacità di monitorare il consumo delle risorse hardware quali CPU, RAM, Storage ecc., dei componenti di tipo Virtual Machine del Cloud.
- **Monitoraggio risorse hardware degli HS:** indica la capacità di monitorare il consumo delle risorse hardware quali CPU, RAM, Storage ecc., dei componenti di tipo Host System del Cloud.
- **Monitoraggio risorse hardware per CCR:** indica la capacità di monitorare il consumo delle risorse hardware quali CPU, RAM ecc., dei componenti di tipo Cluster del Cloud.
- **Monitoraggio risorse hardware per DS:** indica la capacità di monitorare il consumo delle risorse hardware quali la capacità di storage dei componenti di tipo Datastore del Cloud.
- **Monitoraggio risorse hardware per DC:** indica la capacità di monitorare il consumo delle risorse hardware quali CPU, RAM, Storage ecc., dell'intero Datacenter.
- **Statistiche sulle performance (max/min nel periodo):** indica la capacità del sistema di calcolare i valori massimi e minimi, assunti delle metriche di performance, in un determinato arco temporale.
- **Statistiche sulle performance (valore medio/deviazione standard nel periodo):** indica la capacità del sistema di calcolare i valori medi e la deviazione standard, associati alle metriche di performance, in un determinato arco temporale.
- **Visualizzazione proprietà degli elementi:** indica la possibilità di visualizzare le proprietà relative ai componenti del Cloud quali nome, identificativo all'interno del Cloud, stato del componente ecc.
- **Sistema di Alerting:** indica se è offerto all'utente un sistema di notifica degli allarmi inseriti sui componenti del Cloud. L'utente deve quindi avere la possibilità di inserire dei valori soglia sulle metriche di performance, oltrepassati i quali il sistema deve tenere traccia di tale allarme e dei parametri che lo caratterizzano.
- **Notifica Email/Sms:** indica la funzionalità di invio di un messaggio di notifica di un allarme verificatosi, per mezzo di email e/o sms.
- **Sistema di Reporting per elemento:** indica la possibilità di visualizzare tutte le metriche di performance, relative ad un determinato elemento, per mezzo di grafici di tipo TimeSeries e Istogrammi/Grafici a Torta.
- **Reporting con integrazione degli elementi associati:** indica la possibilità di visualizzare il report definito al punto precedente anche con l'integrazione degli elementi associati ad un determinato elemento selezionato dall'utente.
- **Database con storico dati:** Indica la presenza di un database con lo storico dei dati di performance, acquisiti dal software, senza che vi sia perdita di dati nel tempo.

## II. Analisi dell'applicazione

### II.1. Introduzione

In questo capitolo verrà affrontata l'analisi dei requisiti dell'applicazione da progettare e la formalizzazione degli stessi per mezzo dei diagrammi dei casi d'uso.

### II.2. Analisi dei requisiti

Nel capitolo precedente si è affrontato il problema del monitoraggio delle risorse hardware di una infrastruttura Cloud e si sono analizzati alcuni tra i più diffusi sistemi di monitoraggio disponibili sul mercato. Dallo studio delle caratteristiche di tali sistemi derivano i seguenti requisiti che l'applicazione dovrà rispettare:

- **Architettura Client/Server:** tale architettura permette la centralizzazione del sistema di monitoraggio da parte del server e l'accesso da diversi client distribuiti.
- **Accesso via web browser:** l'accesso al server per mezzo di web browser evita l'installazione di software specifico nei client.
- **Autodiscovery degli elementi del Cloud:** l'applicazione dovrà includere la funzionalità di ricerca degli elementi dell'inventario del Cloud e la costruzione della struttura gerarchica dell'inventario.
- **Monitoraggio degli elementi del Cloud:** l'applicazione dovrà provvedere al monitoraggio degli elementi del Cloud delle tipologie Virtual Machine, Host System, Cluster Compute Resource, Datastore.
- **Visualizzazione delle proprietà degli elementi del Cloud:** l'applicazione dovrà consentire la visualizzazione delle proprietà degli elementi del Cloud.
- **Reporting delle metriche e statistiche di performance degli elementi del Cloud e degli elementi associati:** l'applicazione dovrà consentire la visualizzazione in forma grafica e tabellare delle metriche di performance degli elementi del Cloud e degli elementi associati. In particolare dovranno essere visualizzati l'andamento temporale delle metriche nel periodo selezionato dall'utente, ed i valori minimi, massimi, medi e la deviazione standard delle metriche di performance associate ad un determinato elemento. Inoltre dovranno essere visualizzati l'andamento temporale delle metriche e i valori medi nel periodo selezionato anche per gli elementi associati. Gli elementi sono associati secondo la seguente gerarchia:
  - Datacenter -> Cluster -> Host System -> Virtual Machine
  - Datacenter -> Datastore
  - Cluster -> Host System -> Virtual Machine
  - Host System -> Virtual Machine
  - Datastore -> Virtual Machine

- **Dati metriche di performance conservati nel tempo:** i dati relativi alle metriche di performance dovranno essere persistenti nel tempo.
- **Sistema di alarming e alerting via email:** l'applicazione dovrà consentire l'attivazione di allarmi e l'invio di email di alerting.
- **Configurazione del periodo tra due cicli recupero delle metriche:** potrà essere impostato il periodo intercorrente tra due cicli di recupero delle metriche di performance in modo da non creare situazioni di overhead nell'utilizzo delle risorse del Cloud.
- **Non utilizzo di sonde software per i componenti monitorati:** Il sistema di monitoraggio non dovrà prevedere l'installazione di sonde software nei diversi elementi del Cloud.
- **Scalabilità dell'applicazione:** l'applicazione dovrà facilmente consentire l'aggiornamento delle funzionalità quali l'aggiunta di nuove metriche di performance da monitorare ecc.

### II.3. Casi d'uso

Dai requisiti precedentemente elencati segue il seguente diagramma dei casi d'uso:

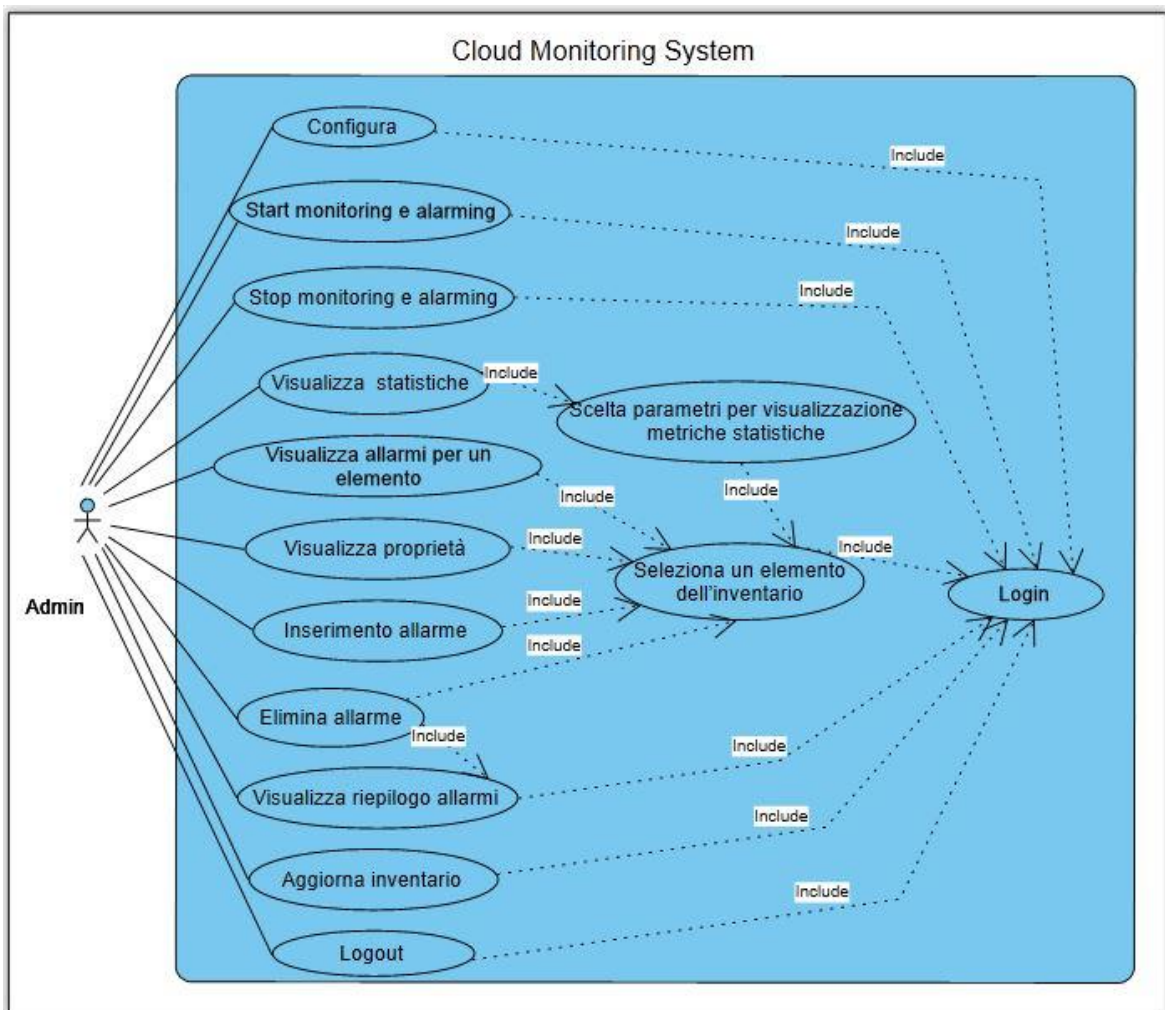


Figura II-1: Diagramma dei casi d'uso

Nella seguente tabella è riportato l'indice dei casi d'uso:



Identificatore del caso d'uso	Nome del caso d'uso	Stakeholder primario	Ambito	Complessità	Priorità
1	Login	Amministratore	Accesso	bassa	1
2	Logout	Amministratore	Accesso	bassa	1
3	Configura	Amministratore	Configurazione	media	1
4	Start monitoring e alarming	Amministratore	Configurazione	media	1
5	Stop monitoring e alarming	Amministratore	Configurazione	media	1
6	Aggiorna inventario	Amministratore	Configurazione	media	1
7	Seleziona un elemento dell'inventario	Amministratore	Reporting	media	1
8	Scelta parametri per visualizzazione metriche statistiche	Amministratore	Reporting	media	2
9	Visualizza statistiche	Amministratore	Reporting	alta	2
10	Visualizza proprietà	Amministratore	Reporting	media	2
11	Inserimento allarme	Amministratore	Alarming	media	2
12	Visualizza allarmi per un elemento	Amministratore	Alarming	media	2
13	Visualizza riepilogo allarmi	Amministratore	Alarming	media	2
14	Elimina allarme	Amministratore	Alarming	Media	2

Figura II-2: Indice dei casi d'uso

Di seguito le specifiche dettagliate dei singoli casi d'uso:

Specifiche del caso d'uso "Login":

Elemento del diagramma	Descrizione
<b>Numero</b>	1
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Login
<b>Descrizione</b>	L'amministratore accede al sistema dopo aver inserito le credenziali di accesso
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore è in possesso delle credenziali di accesso pre

	impostate nel sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	Se le credenziali sono corrette l'amministratore accede al sistema
<b>Flusso alternativo</b>	Se le credenziali sono errate verrà notificato l'errore e l'amministratore potrà inserire nuovamente le credenziali

Figura II-3: Use Case "Login"

Specifiche del caso d'uso "Logout":

Elemento del diagramma	Descrizione
<b>Numero</b>	2
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Logout
<b>Descrizione</b>	L'amministratore esegue il logout dal sistema
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Time out
<b>Flusso principale</b>	L'amministratore esegue il logout dal sistema
<b>Flusso alternativo</b>	Nessuno

Figura II-4: Use Case "Logout"

Specifiche del caso d'uso "Configura":

Elemento del diagramma	Descrizione
<b>Numero</b>	3
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Configura
<b>Descrizione</b>	L'amministratore accede alla pagina di configurazione dell'applicazione ed eventualmente ne modifica le impostazioni
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore accede alla pagina di configurazione dell'applicazione ed eventualmente ne modifica le impostazioni salvandole. L'applicazione mostra un messaggio di conferma dell'operazione.
<b>Flusso alternativo</b>	Nessuno

Figura II-5: Use Case "Configura"

Specifiche del caso d'uso "Start monitoring e alarming":

Elemento del diagramma	Descrizione
<b>Numero</b>	4
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Start monitoring e alarming
<b>Descrizione</b>	L'amministratore avvia il monitoring e alarming del Cloud
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore avvia il monitoring e alarming del Cloud. L'applicazione mostra un messaggio di conferma.
<b>Flusso alternativo</b>	Se il monitoring e alarming è già attivo il sistema mostra un messaggio di notifica

Figura II-6: Use Case "Start monitoring e alarming"

Specifiche del caso d'uso "Stop monitoring e alarming":

Elemento del diagramma	Descrizione
<b>Numero</b>	5
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Stop monitoring e alarming
<b>Descrizione</b>	L'amministratore arresta il monitoring e alarming del Cloud
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore arresta il monitoring e alarming del Cloud. L'applicazione mostra un messaggio di conferma.
<b>Flusso alternativo</b>	Se il monitoring e alarming è già arrestato il sistema mostra un messaggio di notifica

Figura II-7: Use Case "Stop monitoring e alarming"

Specifiche del caso d'uso "Aggiorna inventario":

Elemento del diagramma	Descrizione
<b>Numero</b>	6
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Aggiorna inventario

<b>Descrizione</b>	L'amministratore chiede l'aggiornamento dell'inventario
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore chiede l'aggiornamento dell'inventario
<b>Flusso alternativo</b>	Nessuno

Figura II-8: Use Case "Aggiorna inventario"

Specifiche del caso d'uso "Seleziona un elemento dell'inventario":

Elemento del diagramma	Descrizione
<b>Numero</b>	7
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Seleziona un elemento dell'inventario
<b>Descrizione</b>	L'amministratore seleziona un elemento dell'inventario del Cloud
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore, navigando nella struttura ad albero dell'inventario, seleziona un elemento
<b>Flusso alternativo</b>	Nessuno

Figura II-9: Use Case "Seleziona un elemento dell'inventario"

Specifiche del caso d'uso "Scelta parametri per visualizzazione metriche statistiche":

Elemento del diagramma	Descrizione
<b>Numero</b>	8
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Scelta parametri per visualizzazione metriche statistiche
<b>Descrizione</b>	L'amministratore seleziona i parametri per la visualizzazione delle metriche e statistiche di performance
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema. L'amministratore ha selezionato un elemento dell'inventario del Cloud
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore seleziona l'arco temporale (in termini di date o ultimo periodo), le eventuali tipologie di elementi associati all'elemento precedentemente selezionato, l'ordinamento nei

	grafici, il numero di valori da graficare negli istogrammi e nei pie chart
<b>Flusso alternativo</b>	Nessuno

Figura II-10: Use Case “Scelta parametri per visualizzazione metriche statistiche”

Specifiche del caso d'uso “Visualizza statistiche”:

Elemento del diagramma	Descrizione
<b>Numero</b>	9
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Visualizza statistiche
<b>Descrizione</b>	L'amministratore visualizza le metriche e statistiche di performance in base ai parametri richiesti
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema. L'amministratore ha impostato i parametri di richiesta su un elemento selezionato
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore visualizza i grafici relativi all'elemento selezionato e agli eventuali elementi correlati in maniera aggregata
<b>Flusso alternativo</b>	Nessuno

Figura II-11: Use Case “Visualizza statistiche”

Specifiche del caso d'uso “Visualizza proprietà”:

Elemento del diagramma	Descrizione
<b>Numero</b>	10
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Visualizza proprietà
<b>Descrizione</b>	L'amministratore visualizza le proprietà dell'elemento selezionato
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema. L'amministratore ha selezionato un elemento
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore visualizza le proprietà dell'elemento selezionato, in forma tabellare in una finestra di popup
<b>Flusso alternativo</b>	Nessuno

Figura II-12: Use Case “Visualizza proprietà”

Specifiche del caso d'uso "Inserimento allarme":

Elemento del diagramma	Descrizione
<b>Numero</b>	11
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Inserimento allarme
<b>Descrizione</b>	L'amministratore inserisce un allarme per un elemento
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema. L'amministratore ha selezionato un elemento
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore imposta un allarme su un elemento selezionato indicando i paraetri dell'allarme quali la soglia di trigger, la condizione, la metrica e la durata temporale si superamento della soglia
<b>Flusso alternativo</b>	Nessuno

Figura II-13: Use Case "Inserimento allarme"

Specifiche del caso d'uso "Visualizza allarmi per un elemento":

Elemento del diagramma	Descrizione
<b>Numero</b>	12
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Visualizza allarmi per un elemento
<b>Descrizione</b>	L'amministratore visualizza gli allarmi relativi ad un elemento
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema. L'amministratore ha selezionato un elemento
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore visualizza il riepilogo degli allarmi relativi ad un elemento selezionato
<b>Flusso alternativo</b>	Nessuno

Figura II-14: Use Case "Visualizza allarmi per un elemento"

Specifiche del caso d'uso "Visualizza riepilogo allarmi":

Elemento del diagramma	Descrizione
<b>Numero</b>	13
<b>Applicazione</b>	Sistema di monitoring

<b>Nome</b>	Visualizza riepilogo allarmi
<b>Descrizione</b>	L'amministratore visualizza l'elenco di tutti gli allarmi inseriti
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore visualizza l'elenco di tutti gli allarmi inseriti
<b>Flusso alternativo</b>	Nessuno

Figura II-15; Use Case "Visualizza riepilogo allarmi"

Specifiche del caso d'uso "Elimina allarme":

Elemento del diagramma	Descrizione
<b>Numero</b>	14
<b>Applicazione</b>	Sistema di monitoring
<b>Nome</b>	Elimina allarme
<b>Descrizione</b>	L'amministratore elimina un allarme precedentemente inserito
<b>Attore primario</b>	Amministratore
<b>Pre condizioni</b>	L'amministratore ha eseguito un login al sistema
<b>Trigger</b>	Nessuno
<b>Flusso principale</b>	L'amministratore elimina un allarme dal riepilogo degli allarmi inseriti
<b>Flusso alternativo</b>	L'amministratore elimina un allarme dal riepilogo degli allarmi inseriti per un elemento selezionato

Figura II-16: Use Case "Elimina allarme"

## III. Progettazione dell'applicazione

### III.1. Introduzione

In questo capitolo verrà affrontata la progettazione della Webapp (Web Application) facendo uso del linguaggio Java EE, utilizzato secondo il pattern MVC, e del database MySQL .

### III.2. Architettura della Webapp

L'applicazione verrà realizzata come Webapp, utilizzando la piattaforma Java EE (Enterprise Edition) [JCEGHMS]. Java EE è una piattaforma di programmazione Java, che fornisce API e un ambiente di runtime, per lo sviluppo e l'esecuzione di software per le imprese, compresi i servizi web, e di grandi applicazioni di rete a più livelli, scalabile, affidabile e sicuro.

Una Webapp [SR03] [JCEGHMS] [P05] è definita come una collezione di componenti separati, che possono comunicare fra loro, che costituiscono un sistema eseguibile da un Web Container [JCEGHMS], anche in maniera concorrente.

Nell'ambito dello sviluppo delle Webapp assume un ruolo importante il pattern MVC (Model View Controller) [LR01] [P05] [BCK03] . Questo pattern prevede la separazione dei dati dalla loro presentazione all'utente introducendo un'architettura a livelli. I tre livelli, Model, View e Controller, hanno funzionalità distinte tra loro e introducono un disaccoppiamento tra gli stessi. Il principale vantaggio è che la modifica di uno dei tre non implica la modifica degli altri, facilitando il riutilizzo del codice.

In particolare abbiamo:

- **Model (Modello):** Rappresenta il livello che si occupa della gestione dei dati, tipicamente persistenti su database, e del comportamento. E' indipendente dalla specifica rappresentazione dei dati.
- **View (Vista):** Ha il compito di consentire all'utente di inviare le richieste di elaborazione all'applicazione e di visualizzare all'utente il risultato di tale elaborazione, anche in funzione del dispositivo di accesso utilizzato.
- **Controller (Controllo):** E' l'elemento che mette in comunicazione il Model e la View. Riceve le richieste di elaborazione dalla View e indirizza le richieste di elaborazione al Model. Al termine dell'elaborazione invia i risultati alla View per la presentazione all'utente.



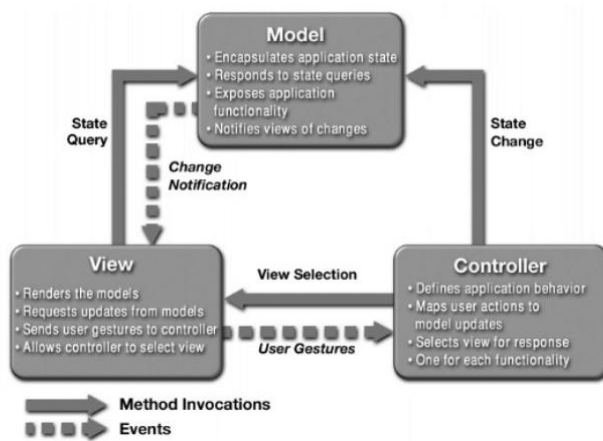


Figura III-1: Model, View e Controller

Nel contesto della Webapp oggetto della tesi, è utilizzato il Model 2 [TB02] [P05] del pattern MVC, detto anche Servlet Centric, in quanto si basa sull'utilizzo delle Servlet [FJ04] [JCEGHMS].

Il Controller è composto da una Servlet che ha il compito di acquisire le richieste in ingresso dalla rete svolgendo il ruolo di “dispatcher”, e di inoltrare ciascuna richiesta verso il corrispondente “handler” per consentirne l’elaborazione. Al termine dell’elaborazione la Servlet determina a quale pagina JSP restituire il controllo, eseguendo il “redirecting”.

Il Model è costituito da classi definite “Manager”, con il compito di interfacciare le funzionalità offerte verso lo strato Controller, da classi DAO (Data Access Object) di accesso al database, e da classi contenitore denominate JavaBean, la cui funzione è di incapsulare più oggetti in uno, utilizzate per il passaggio degli stessi tra classi e tra metodi.

Il livello View consiste in una serie di pagine JSP che hanno lo scopo di permettere all’utente di inoltrare le richieste di elaborazione al sistema e di visualizzare i risultati dell’elaborazione.

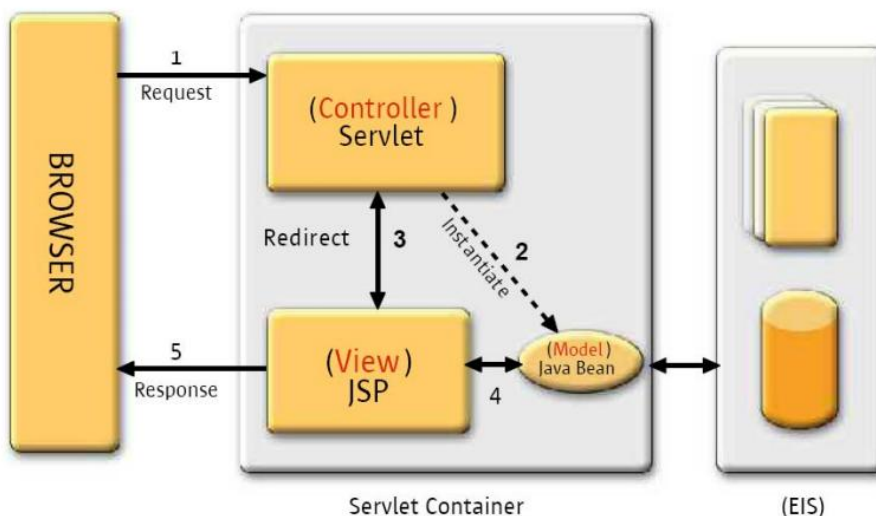


Figura III-2: MVC Model 2 (Servlet - Centric Architecture)

Il sistema prevede il seguente scenario di interazione:

Il browser invia una richiesta (request) di una risorsa al Web Server. La Servlet in funzione nel Web Server, che svolge la funzione di Web Container, si occupa della creazione dell'istanza e della gestione della Servlet durante il suo ciclo di vita. Il Web Container, acquisisce la richiesta e individua l'handler che dovrà gestirla. Vengono istanziati una serie di JavaBean e invocate le classi Manager e DAO che eseguono le elaborazioni richieste, ed eventualmente interagiscono con il back-end del sistema accedendo al database. Al termine della elaborazione il controllo torna alla Servlet che, sulla base del risultato, determina la View e quindi la pagina JSP verso la quale eseguire il redirecting. Quest'ultima estrae i risultati e li visualizza all'utente. Infine, la pagina HTML prodotta dalla JSP viene inviata al browser come risposta (response).

#### **Struttura della Webapp**

La Webapp è costituita dai seguenti elementi disposti secondo una struttura gerarchica:

- Servlet
- Pagine JSP
- JavaBean
- Pagine HTML
- File multimediali (immagini, suoni, video, ...)
- Applet, fogli di stile (CSS) e file JavaScript

All'interno della directory radice (root) è presente la directory WEB-INF, nella quale vengono memorizzate le meta-informazioni della Webapp, al cui interno troviamo le seguenti directory:

- WEB-INF/classes: contiene i package e le relative classi Java che sono utilizzate all'interno dell'applicazione.
- WEB-INF/lib : contiene i file JAR (Java Archive), all'interno dei quali ci sono ulteriori classi distribuite nell'applicazione.

L'ambiente Java consente di distribuire un'applicazione Web mediante un unico file WAR (Web Archive), nel quale sono incluse tutte le risorse necessarie per il funzionamento.

#### **Configurazione della Webapp**

Il Web Application Deployment Descriptor [JCEGHMS] è un file XML, denominato *web.xml*, all'interno del quale sono presenti le informazioni di configurazione della Webapp utilizzate dal Web Container nella fase di avvio della stessa.

Le informazioni principali che costituiscono il file web.xml sono le seguenti:

- definizione della Servlet che funziona da Controller e mapping delle richieste.
- dichiarazione dei parametri iniziali passati alla Servlet in fase di avvio.

- configurazione delle librerie di tag.
- elenco dei file di benvenuto dell'applicazione (Welcome File List).
- utilizzo di eventuali filtri per le richieste.

Per la definizione della Servlet, viene utilizzato il tag `<servlet>`, mediante il quale vengono specificate le seguenti informazioni:

- il nome da assegnare alla Servlet, mediante il quale fare riferimento ad essa nel resto del file (`<servlet-name>`).
- la classe che definisce la Servlet (`<servlet-class>`).

Il passo successivo è quello di fare in modo che ogni richiesta venga smistata dal Web Container verso la Servlet specificata. Questa operazione è nota come “mapping”, in quanto viene eseguita una mappatura fra le tipologie di richieste e la Servlet. Per questo scopo, viene utilizzato il tag `<servlet-mapping>` all'interno del quale sono specificate le seguenti informazioni :

- il nome della Servlet a cui inoltrare le richieste (`<servlet-name>`).
- elenco delle estensioni che devono avere le richieste per essere inoltrate alla Servlet corretta (`<url-pattern>`).

L'utilizzo del tag `<taglib>` è opzionale dalla versione JSP 2.0, in quanto il Web Container cercherà automaticamente tali librerie nelle directory e sottodirectory `/WEB-INF`, `/META-INF` così come nel file JAR.

Infine, è possibile specificare un elenco di risorse, solitamente pagine, che il Web Container invia a seguito di una richiesta incompleta. Ciascun file viene definito con il tag `<welcome-file>`, all'interno del blocco `<welcome-file-list>`.

Il file `web.xml` completo della Webapp è il seguente:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3/EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <display-name>Tesi</display-name>
  <description>Tesi di Laurea - Giuseppe Murtas</description>
  <context-param>
    <param-name>filesDir</param-name>
    <param-value>/resources/</param-value>
  </context-param>
  <listener>
    <listener-class>tesi.servlet.SystemListener</listener-class>
```

```

</listener>
<servlet>
  <servlet-name>SystemService</servlet-name>
  <display-name>Application Servlet</display-name>
  <description>Application Servlet</description>
  <servlet-class>tesi.servlet.SystemServlet</servlet-class>
  <load-on-startup>-1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>SystemService</servlet-name>
  <url-pattern>/tesi.do</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>60</session-timeout>
</session-config>
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>

```

### III.3. La gestione dei thread

I thread Java [JCEGHMS], sono il mezzo mediante il quale Java fa eseguire un'applicazione da più Java Virtual Machine (JVM) contemporaneamente, allo scopo di ottimizzare i tempi del runtime. Per “multi-threading” si intende il processo che porterà un'applicazione, a definire più di un thread, assegnando ad ognuno compiti da eseguire parallelamente. L'ambiente Java, per mezzo delle Virtual Machine, offre uno strato d'astrazione per poter gestire il multi-threading direttamente dal linguaggio.

Ogni thread ha un proprio ciclo di vita. Quando viene invocato il metodo `start()` il thread assume lo stato di *Runnable*, ovvero è pronto per essere eseguito. In generale possono esserci più thread nello stato *Runnable* ma soltanto uno è quello che, in un determinato istante, sarà in esecuzione, ovvero si troverà nello stato *Running*. Un thread in esecuzione continua a rimanere nello stato *Running* fino a quando esso non cessa di essere *Runnable* oppure finché non giunge un thread con priorità maggiore, sempre con lo stato *Runnable*.

Quando un thread cessa di essere *Runnable*, passa allo stato *Blocked*. Esistono diverse ragioni per le quali un thread potrebbe passare dallo stato *Runnable* a quello *Blocked*. Ad esempio, la causa potrebbe essere una chiamata al metodo `Thread.sleep()`, che impone al thread corrente l'interruzione dell'esecuzione del task per un periodo di tempo prefissato. Un thread termina la sua esecuzione quando è terminato l'insieme delle istruzioni che deve eseguire o quando arriva una eccezione di tipo

*InterruptedException*. Quando un thread termina entra nello stato *Dead* e non può essere eseguito nuovamente.

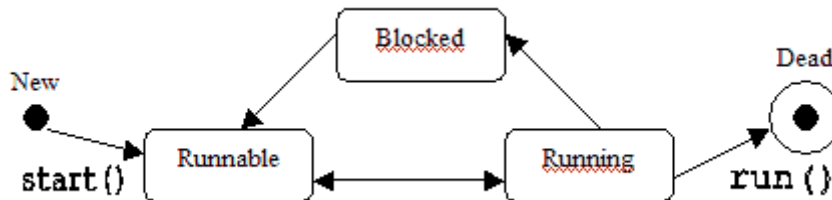


Figura III-3: Diagramma degli stati per un thread

La Webapp deve consentire all'utente di accedere al sistema attraverso il login, di visualizzare le proprietà degli elementi del Cloud, di visualizzare e graficare le metriche e statistiche di performance disponibili. Al contempo l'applicazione deve recuperare, con frequenza indicata nel file di configurazione, le metriche di performance degli elementi del Cloud nonché verificare con la stessa periodicità lo stato degli allarmi inseriti dall'utente. Tali funzionalità possono poi essere sospese dall'utente. Si ha quindi la necessità di avere un flusso di esecuzione principale e di creare una serie di flussi secondari per l'esecuzione parallela delle funzionalità descritte.

La scelta adottata è quindi quella di creare all'avvio dell'applicazione un thread di gestione delle funzionalità di monitoring e alarming. Tale thread sarà quindi nello stato *Running* e, con periodicità prefissata, creerà due thread figli che si occuperanno rispettivamente del recupero delle metriche statistiche degli elementi del Cloud e della verifica degli allarmi. Ognuno dei thread figli avrà il compito di eseguire rispettivamente un unico ciclo di monitoring e di alarming dopo i quali entreranno nello stato *Dead*. L'utente, volendo interrompere il monitoring e l'alarming del sistema, attraverso un apposito comando dell'applicazione, farà sì che il thread secondario di gestione, non avendo istruzioni da eseguire, entrerà anch'esso nello stato *Dead*. Al riavvio del sistema di monitoring e alarming verrà creato un nuovo thread di gestione secondario ed il flusso di esecuzione procederà nuovamente come descritto.

Il seguente diagramma di sequenza illustra la gestione dei thread della Webapp:

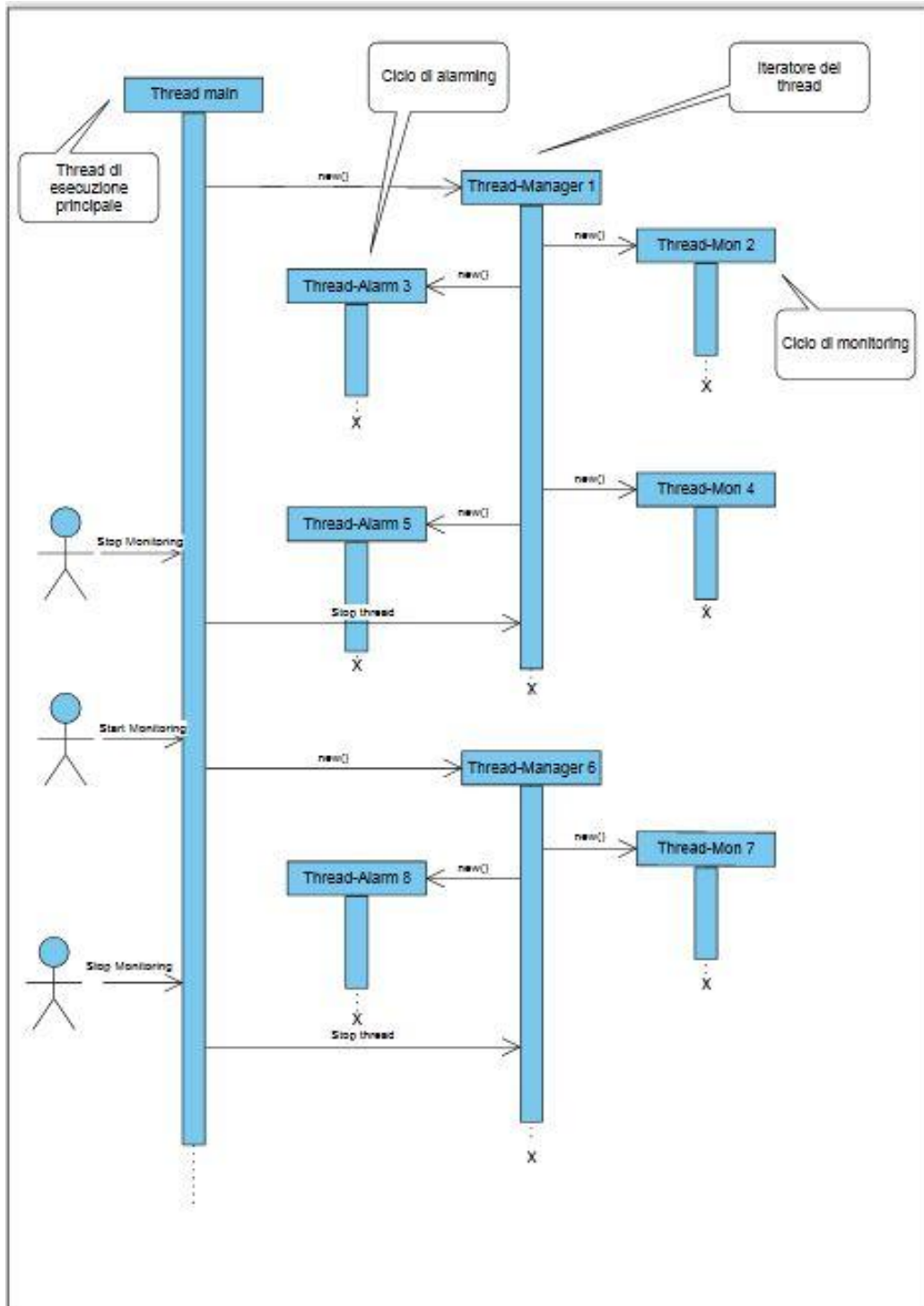


Figura III-4: Diagramma di sequenza gestione dei thread dell'applicazione

Come evidenziato nel diagramma di sequenza, per ogni ciclo di monitoring ed alarming viene creato un thread dedicato. In particolare la classe MonitorDataThread() costituirà il thread responsabile del ciclo di monitoring e la classe AlarmThread() costituirà il ciclo di alarming.

### III. Progettazione dell'applicazione

La classe `MonitorDataThread()` recupera l'inventario del Cloud dal vCenter Server e, per ogni elemento del Cloud, recupera le metriche di performance, sempre dal vCenter Server, inserendole nel database dell'applicazione per mezzo delle classi di tipo Manager e DAO.

Di seguito il diagramma di sequenza relativo ad un ciclo di monitoring della classe `MonitorDataThread()`:

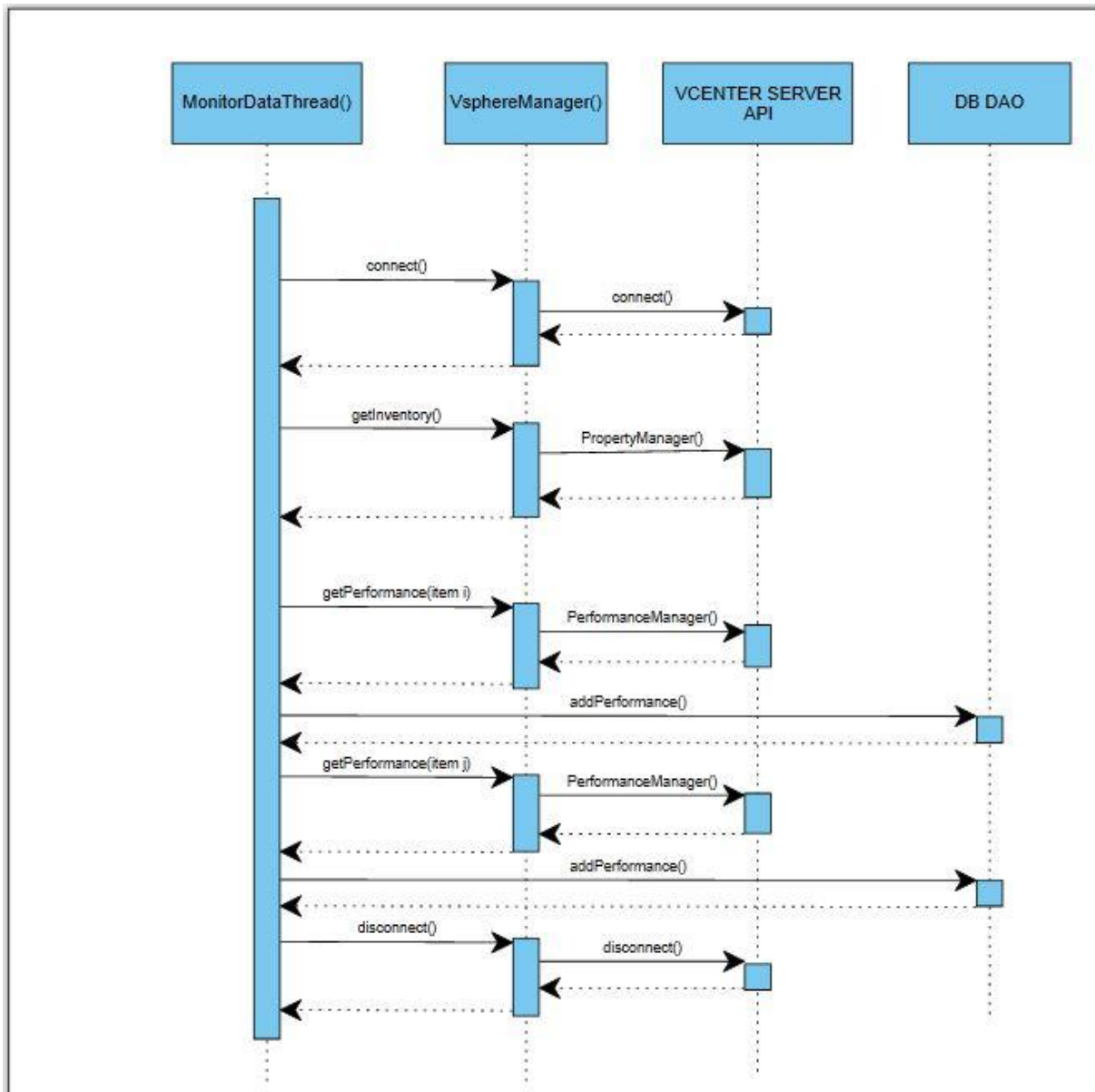


Figura III-5: Diagramma di sequenza della classe `MonitorDataThread()`

La classe `AlarmThread()` recupera l'insieme degli allarmi attivi dal database e, per ogni allarme, verifica il superamento della soglia impostata e gli altri parametri inseriti dall'utente. Qualora l'allarme sia verificato viene inviata una email contenente i parametri dell'allarme all'indirizzo specificato dall'utente in fase di configurazione dell'applicazione.

Di seguito il diagramma di sequenza relativo ad un ciclo di monitoring della classe AlarmThread():

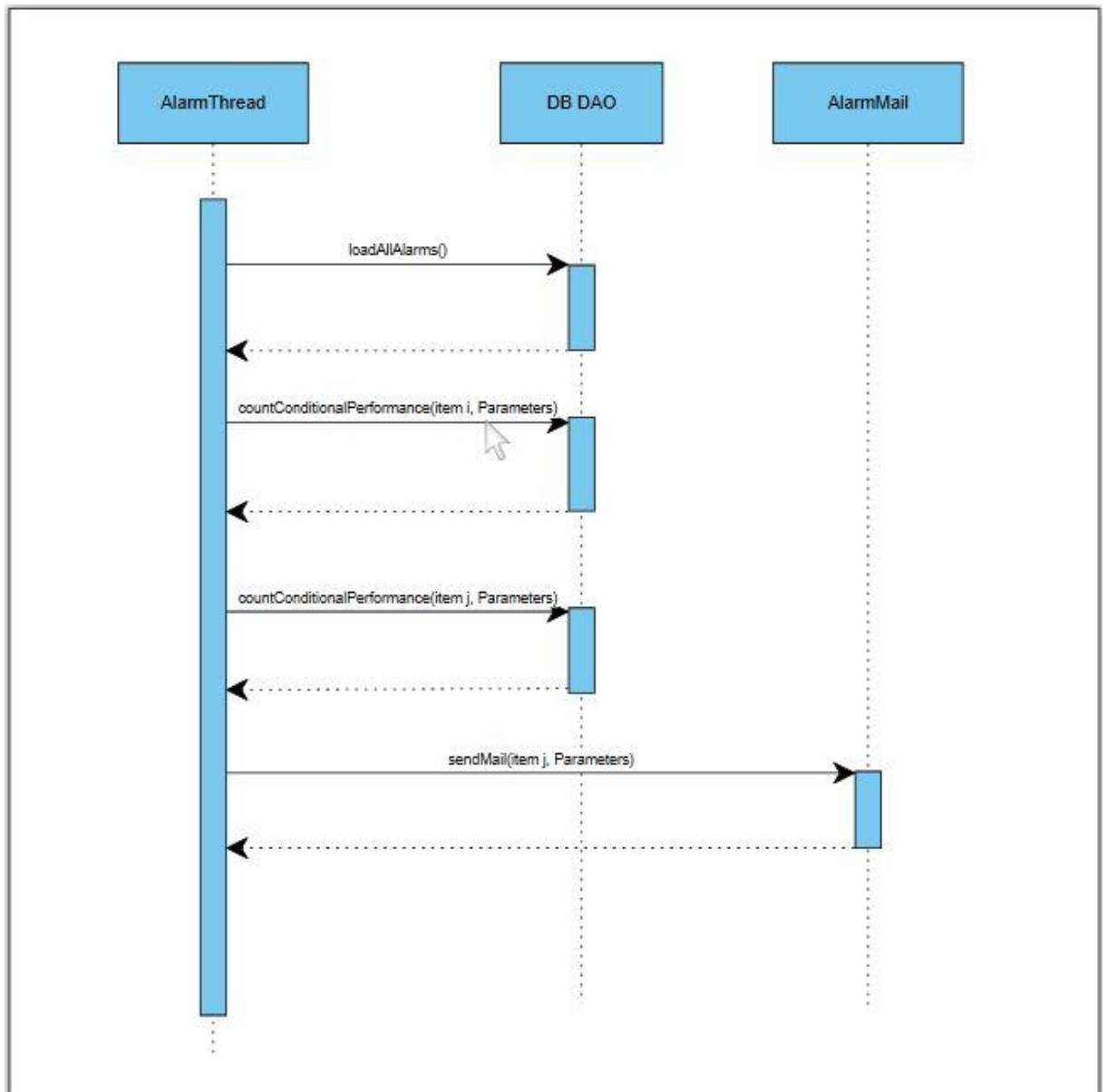


Figura III-6: Diagramma di sequenza della classe AlarmThread()

### III.4. Generazione dei grafici

L'applicazione prevede la visualizzazione delle metriche e statistiche di performance degli elementi del Cloud anche in forma grafica. I grafici dell'applicazione sono generati mediante la libreria JFreeChart [G07] [G08]. JFreeChart è una libreria Java Open Source che consente la generazione dinamica di grafici a partire da una fonte di dati. Tramite JFreeChart è possibile generare grafici di diversa tipologia quali grafici a torta, istogrammi, line chart, time series chart, scatter plot chart, Gantt, ecc. I grafici generati possono essere esportati in diversi formati tra cui *png* e *jpg*.



In particolare, per le esigenze dell'applicazione, le metriche e statistiche di performance sono visualizzate mediante le seguenti tipologie di grafici.

- **Pie Chart:** Grafici a torta utilizzati per confrontare più valori di una grandezza. Sono generati usando la seguente funzione della libreria:

```
JFreeChart chart = ChartFactory.createPieChart();
```

- **Bar Chart:** Grafici a barre verticali utilizzati per confrontare più valori di una grandezza. Sono generati usando la seguente funzione della libreria:

```
JFreeChart chart = ChartFactory.createBarChart();
```

- **Time Series Chart:** Grafici serie temporali per la visualizzazione dell'andamento nel tempo delle metriche di performance. Sono generati usando la seguente funzione della libreria:

```
JFreeChart chart = ChartFactory.createTimeSeriesChart();
```

## III.5. Il database del sistema

La persistenza dei dati della Webapp, aldilà delle informazioni memorizzate nel file di configurazione xml, è garantita dalla presenza del database MySQL [OR13a] del sistema.

In particolare il database del sistema sarà costituito da una serie di entità e di relazioni secondo il modello relazionale.

### III.5.1. Descrizione del database

Il Cloud da monitorare è costituito da un insieme di elementi di varia tipologia (VirtualMachine, HostSystem, ClusterComputeResource ecc). Ogni elemento ha un nome, un identificativo e, come già detto, appartiene ad un tipologia. Sono disponibili metriche di performance per gli elementi di tipo VirtualMachine, HostSystem, ClusterComputeResource, Datastore. Le metriche di ognuna delle suddette tipologie sono associate all'identificativo dell'elemento, al timestamp di recupero delle metriche, e da un insieme di valori riferiti alle metriche disponibili. Sono disponibili anche metriche associate tra elementi di tipo VirtualMachine ed elementi di tipo Datastore. Queste ultime saranno quindi associate agli identificativi degli elementi di tipo VirtualMachine e Datastore. Sono disponibili degli allarmi sulle metriche. Ogni allarme è caratterizzato da uno o due elementi di cui sono state memorizzate le metriche, dalle relative tipologie, dalla metrica da monitorare, da una condizione sul valore attuale (maggiore o minore di una determinata soglia), dalla durata temporale per cui si deve verificare la condizione, dallo stato di attività dell'allarme (attivo/non attivo), dallo stato di triggering dell'allarme (allarme verificatosi o meno) e dal timestamp di inserimento dell'allarme stesso. In caso di verifica positiva dell'allarme viene memorizzato anche il timestamp in cui si è verificata la condizione di allarme. Ogni metrica monitorata è caratterizzata dalla tipologia (VirtualMachine, HostSystem ecc..) dall'identificativo della metrica e dalla sua descrizione.

Si prevedono dunque le seguenti entità:

- allarme

- performance ClusterComputeResource
- condizioni
- performance Datastore
- performance HostSystem
- elementi
- tipologie degli elementi
- metriche
- performance VirtualMachine
- performance VirtualMachine e Datastore associate

Da cui deriva il diagramma E-R del prossimo paragrafo.

### III.5.2. Diagramma E-R

Il diagramma Entity-relationship risultante è mostrato in figura:

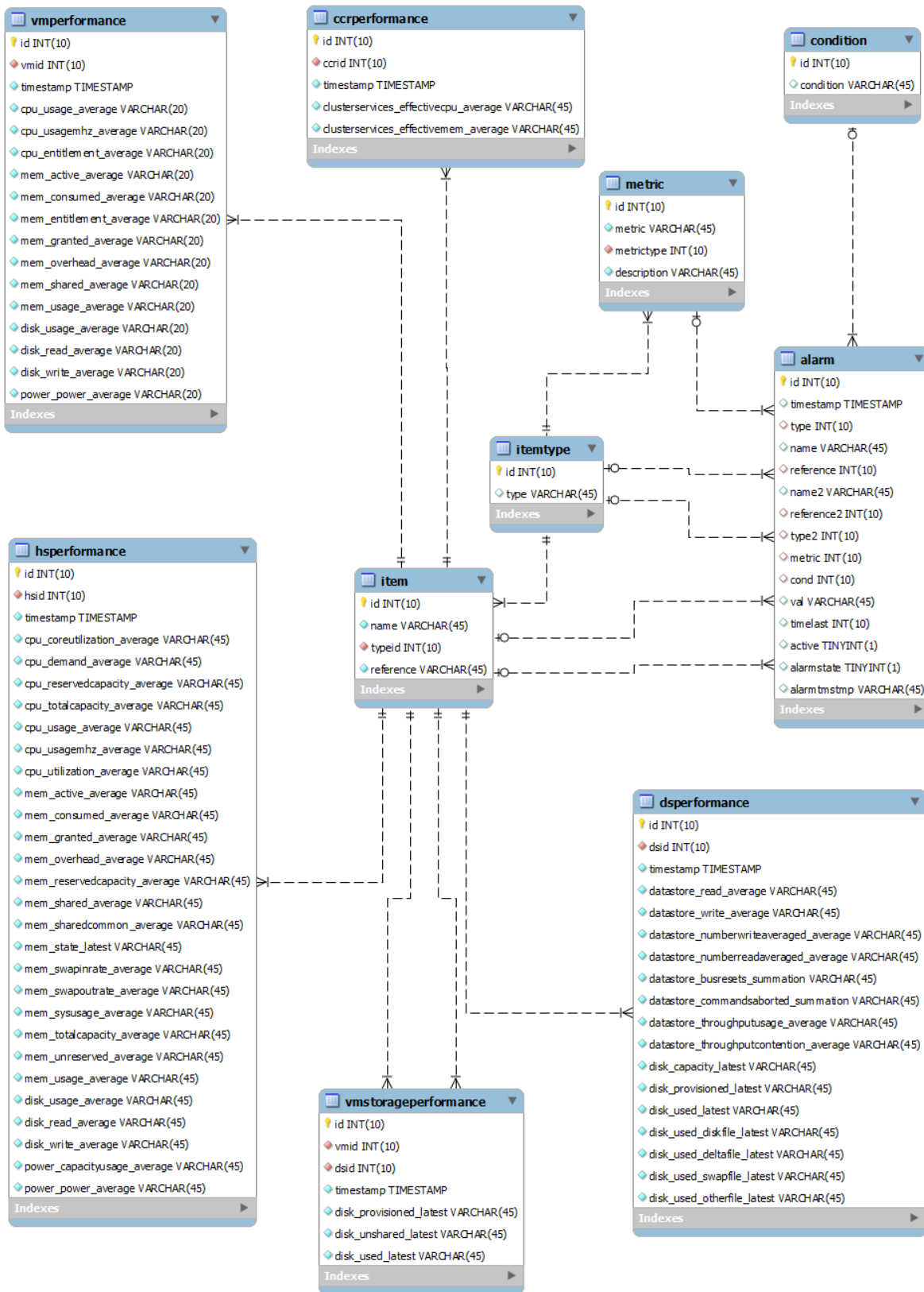


Figura III-7: Diagramma E-R del database

### III.5.3. Le tabelle del database

Il DB è costituito dalle seguenti tabelle

- alarm
- ccrperformance
- condition
- dsperformance
- hsperformance
- item
- itemtype
- metric
- vmperformance
- vmstorageperformance

#### III.5.3.1. Tabella alarm

La tabella *alarm* rappresenta gli allarmi sulle metriche di performance inseriti dall'utente.

Column Name	Column Type
id	INT(10)
timestamp	TIMESTAMP
type	INT(10)
name	VARCHAR(45)
reference	INT(10)
name2	VARCHAR(45)
reference2	INT(10)
type2	INT(10)
metric	INT(10)
cond	INT(10)
val	VARCHAR(45)
timelast	INT(10)
active	TINYINT(1)
alarmstate	TINYINT(1)
alarmtstamp	VARCHAR(45)

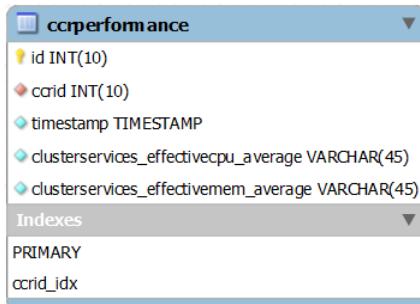
  

Index Name	Index Type
PRIMARY	PRIMARY
reference_idx	INDEX
reference2_idx	INDEX
type_idx	INDEX
type2_idx	INDEX
metric_idx	INDEX
cond_idx	INDEX

Figura III-8: Tabella alarm

### III.5.3.2. Tabella *ccrperformance*

La tabella *ccrperformance* rappresenta le metriche associate agli elementi del Cloud del tipo ClusterComputeResource.

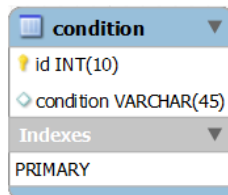


ccrperformance	
id	INT(10)
ccrid	INT(10)
timestamp	TIMESTAMP
clusterservices_effectivecpu_average	VARCHAR(45)
clusterservices_effectivemem_average	VARCHAR(45)
Indexes	
PRIMARY	
ccrid_idx	

Figura III-9: Tabella *ccrperformance*

### III.5.3.3. Tabella *condition*

La tabella *condition* rappresenta l'insieme degli operatori di confronto utilizzati nella verifica dei valori soglia inseriti per gli allarmi.



condition	
id	INT(10)
condition	VARCHAR(45)
Indexes	
PRIMARY	

Figura III-10: Tabella *condition*

### III.5.3.4. Tabella *dsperformance*

La tabella *dsperformance* rappresenta le metriche di performance associate agli elementi del Cloud del tipo Datastore.

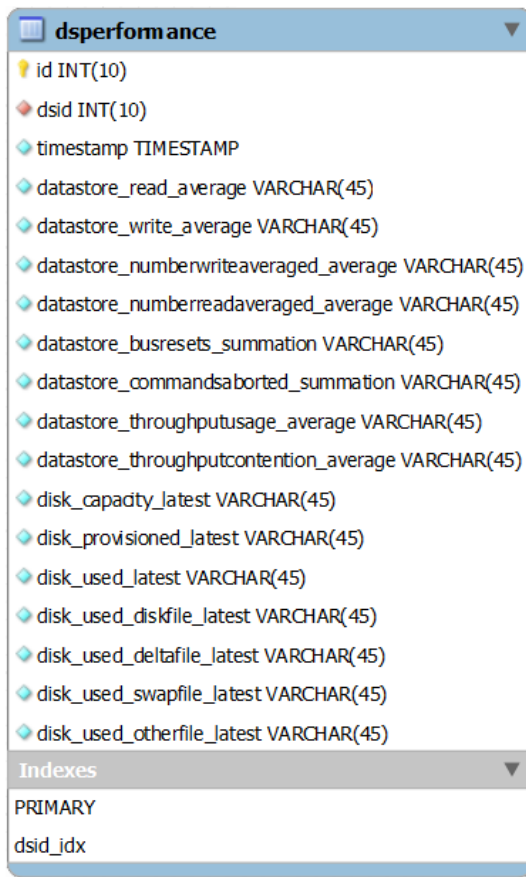


Figura III-11: Tabella dsperformance

### III.5.3.5. Tabella hspperformance

La tabella *hspperformance* rappresenta le metriche di performance associate agli elementi del Cloud del tipo HostSystem.

hspperformance	
id	INT(10)
hsid	INT(10)
timestamp	TIMESTAMP
cpu_coreutilization_average	VARCHAR(45)
cpu_demand_average	VARCHAR(45)
cpu_reservedcapacity_average	VARCHAR(45)
cpu_totalcapacity_average	VARCHAR(45)
cpu_usage_average	VARCHAR(45)
cpu_usagemhz_average	VARCHAR(45)
cpu_utilization_average	VARCHAR(45)
mem_active_average	VARCHAR(45)
mem_consumed_average	VARCHAR(45)
mem_granted_average	VARCHAR(45)
mem_overhead_average	VARCHAR(45)
mem_reservedcapacity_average	VARCHAR(45)
mem_shared_average	VARCHAR(45)
mem_sharedcommon_average	VARCHAR(45)
mem_state_latest	VARCHAR(45)
mem_swapinrate_average	VARCHAR(45)
mem_swapoutrate_average	VARCHAR(45)
mem_sysusage_average	VARCHAR(45)
mem_totalcapacity_average	VARCHAR(45)
mem_unreserved_average	VARCHAR(45)
mem_usage_average	VARCHAR(45)
disk_usage_average	VARCHAR(45)
disk_read_average	VARCHAR(45)
disk_write_average	VARCHAR(45)
power_capacityusage_average	VARCHAR(45)
power_power_average	VARCHAR(45)
Indexes	
PRIMARY	
hsid_idx	

Figura III-12: Tabella hspperformance

### III.5.3.6. Tabella item

La tabella *item* rappresenta gli elementi dell'inventario del Cloud.

item	
id	INT(10)
name	VARCHAR(45)
typeid	INT(10)
reference	VARCHAR(45)
Indexes	
PRIMARY	
reference_UNIQUE	
typeid_idx	

Figura III-13: Tabella item

### III.5.3.7. Tabella itemtype

La tabella *itemtype* rappresenta la tipologia associata agli elementi del Cloud.

itemtype	
id	INT(10)
type	VARCHAR(45)
Indexes	
PRIMARY	

Figura III-14: Tabella itemtype

### III.5.3.8. Tabella metric

La tabella *metric* rappresenta le metriche di performance associate alle diverse tipologie degli elementi del Cloud.

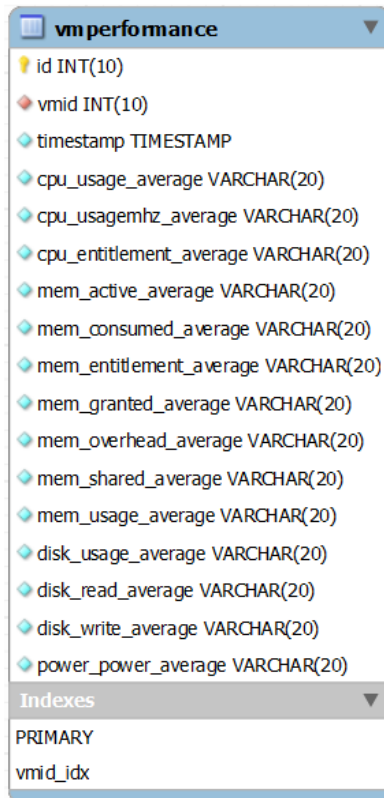
metric	
id	INT(10)
metric	VARCHAR(45)
metrictype	INT(10)
description	VARCHAR(45)
Indexes	
PRIMARY	
metrictype_idx	

Figura III-15: Tabella metric



### III.5.3.9. Tabella *vmperformance*

La tabella *vmperformance* rappresenta le metriche di performance associate agli elementi del Cloud del tipo VirtualMachine.



The screenshot shows the table definition for 'vmperformance'. It lists the following columns and their data types:

- id INT(10)
- vmid INT(10)
- timestamp TIMESTAMP
- cpu\_usage\_average VARCHAR(20)
- cpu\_usagemhz\_average VARCHAR(20)
- cpu\_entitlement\_average VARCHAR(20)
- mem\_active\_average VARCHAR(20)
- mem\_consumed\_average VARCHAR(20)
- mem\_entitlement\_average VARCHAR(20)
- mem\_granted\_average VARCHAR(20)
- mem\_overhead\_average VARCHAR(20)
- mem\_shared\_average VARCHAR(20)
- mem\_usage\_average VARCHAR(20)
- disk\_usage\_average VARCHAR(20)
- disk\_read\_average VARCHAR(20)
- disk\_write\_average VARCHAR(20)
- power\_power\_average VARCHAR(20)

Below the columns, the indexes are listed:

- PRIMARY
- vmid\_idx

Figura III-16: Tabella *vmperformance*

### III.5.3.10. Tabella *vmstorageperformance*

La tabella *vmstorageperformance* rappresenta le metriche di performance sullo storage associate agli elementi del Cloud del tipo VirtualMachine e Datastore.

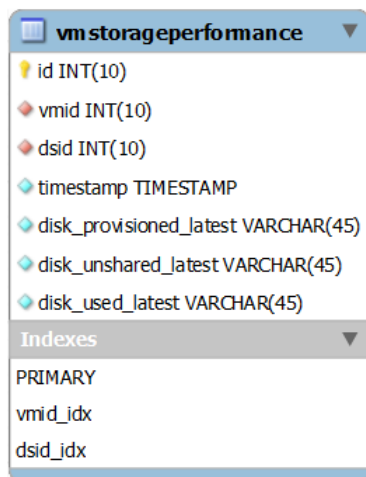


Figura III-17: vmstorageperformance

### III.5.4. Accesso al database

L'accesso al database avviene per mezzo del driver JDBC (Java DataBase Connectivity) [JCEGHMS] per database MySQL. JDBC è l'insieme delle API che definiscono le specifiche per l'accesso ai database relazionali. L'insieme delle classi che offrono una connettività tra Java e un DBMS, implementano le interfacce dei package java.sql e javax.sql.

## III.6. Le Classi e i Package del sistema (Controller)

Le classi del sistema facenti parte del "Controller" sono organizzate in Package secondo la seguente struttura:

- *tesi*
- *tesi.servlet*

I package e le classi ad esso associate sono descritte di seguito.

### III.6.1. Package tesi.servlet

Il package contiene le classi di tipo Listener e Servlet che implementano il Controller dell'applicazione. Le classi appartenenti al package *tesi.servlet* sono mostrate nella seguente figura:

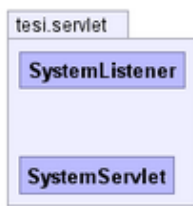


Figura III-18: Classi del package `tesi.servlet`

Di seguito la descrizione delle classi del package `tesi.servlet`:

### III.6.1.1. Classe `SystemListener`

Classe incaricata dell'inizializzazione del sistema all'avvio della Webapp. Crea un'istanza del contesto del sistema e la rende disponibile come attributo del `ServletContext`. Diagramma UML della classe:

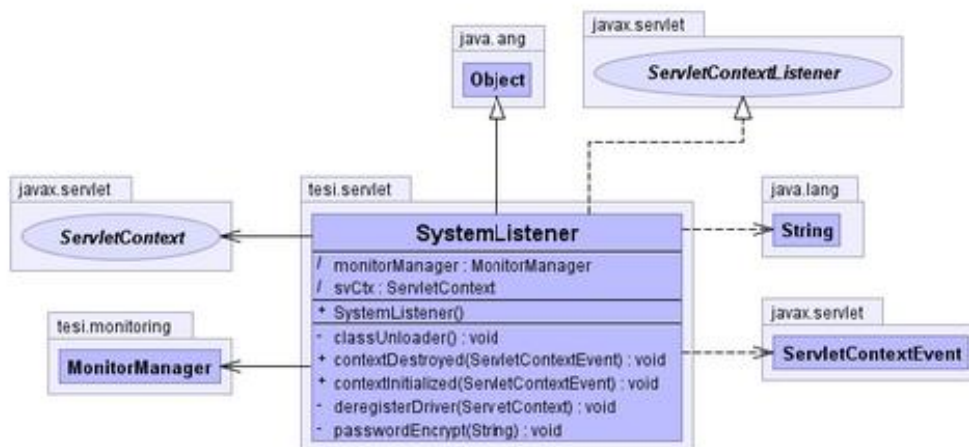


Figura III-19: Diagramma UML della classe `SystemListener`

### III.6.1.2. Classe `SystemServlet`

Classe Servlet di controllo, punto di ingresso per le richieste degli utenti. Predisporre il contesto di richiesta, invoca la corretta jsp per la visualizzazione della risposta. Diagramma UML della classe:

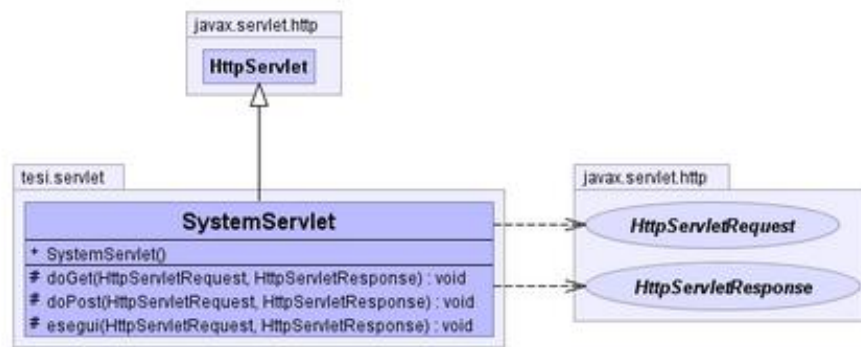


Figura III-20: Diagramma UML della classe SystemServlet

### III.7. Le Classi e i Package del sistema (Model)

Le classi del sistema facenti parte del “Model” sono organizzate in Package secondo la seguente struttura:

- *tesi*
- *tesi.chart*
- *tesi.db*
- *tesi.db.alarm*
- *tesi.db.ccrperformance*
- *tesi.db.condition*
- *tesi.db.dsperformance*
- *tesi.db.hsperformance*
- *tesi.db.install*
- *tesi.db.inventory*
- *tesi.db.metric*
- *tesi.db.vmpformance*
- *tesi.db.vmsperformance*
- *tesi.directory*
- *tesi.mail*
- *tesi.monitoring*
- *tesi.security*
- *tesi.vmware*

- *tesi.vmware.beans*
- *tesi.xml*
- *tesi.xml.beans*

I package e le classi ad esso associate sono descritte di seguito.

#### III.7.1.1. Package *tesi.chart*

Il package contiene le classi atte all'elaborazione delle metriche e statistiche di performance e alla creazione dei relativi grafici. Le classi appartenenti al package *tesi.chart* sono mostrate nella seguente figura:

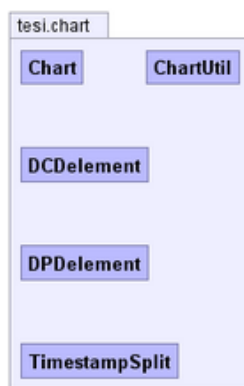


Figura III-21: Classi del package *tesi.chart*

Di seguito la descrizione delle classi del package *tesi.chart*:

#### III.7.1.2. Classe *Chart*

La classe consente la creazione di diversi tipi grafici (Bar chart, Pie chart, Time series chart...) utilizzando la libreria java JfreeChart. Diagramma UML della classe:

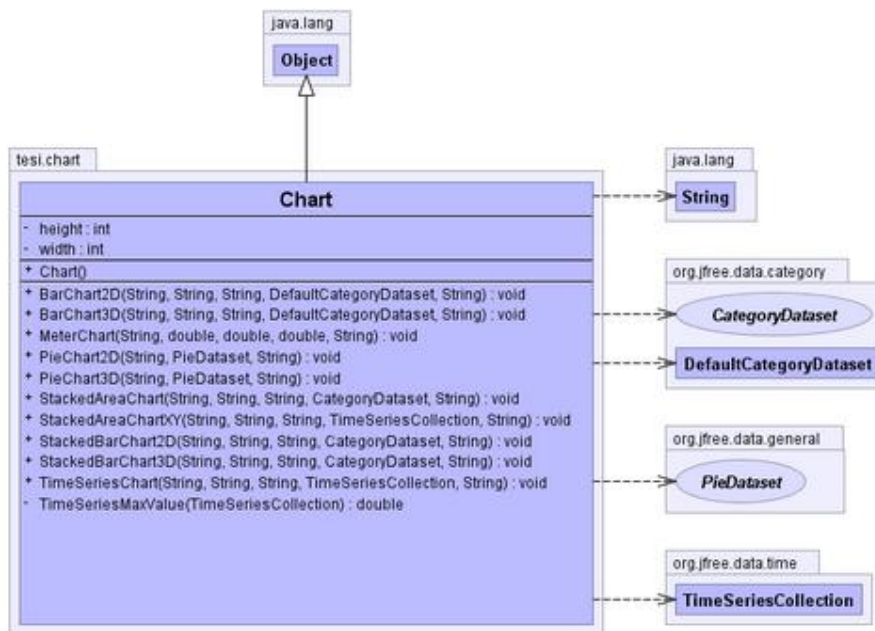


Figura III-22: Diagramma UML della classe Chart

### III.7.1.1. Classe ChartUtil

La classe elabora le metriche statistiche di performance fornite dal database per la costruzione dei diversi tipi di grafici previsti dall'applicazione. L'elaborazione riguarda in particolare l'ordinamento e l'inserimento degli oggetti di tipo *List* nei dataset associati ai grafici, ed il calcolo di eventuali valori minimi, massimi, medi e dell'indicatore statistico deviazione standard. I grafici vengono poi creati in formato PNG per la visualizzazione nel browser. Diagramma UML della classe:

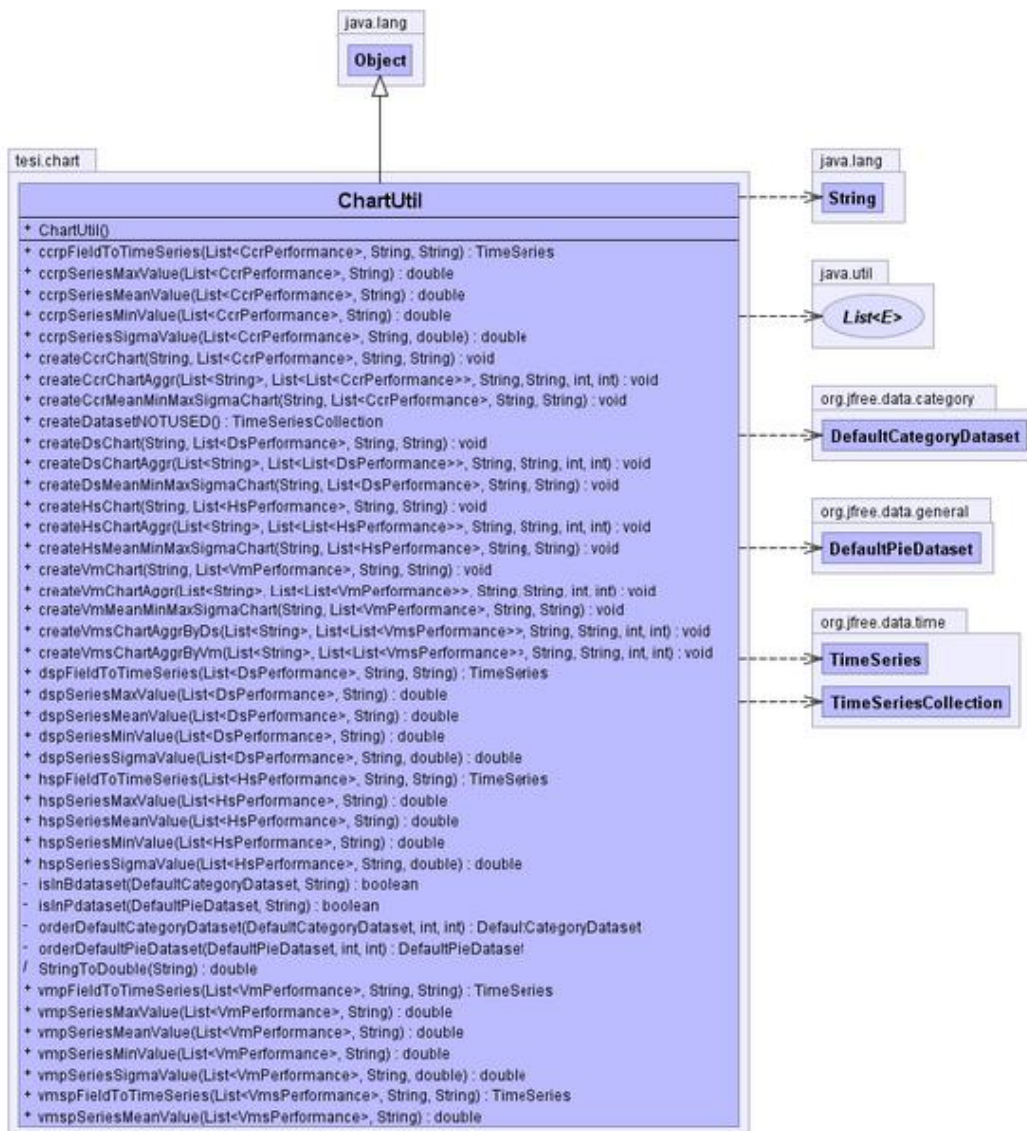


Figura III-23: Diagramma UML della classe ChartUtil

### III.7.1.2. Classe DCDelement

Gli oggetti della classe rappresentano un elemento del dataset *DefaultCategoryDataset* per l'ordinamento secondo l'ordine crescente/decescente dei grafici a barre creati nella classe ChartUtil. Diagramma UML della classe:

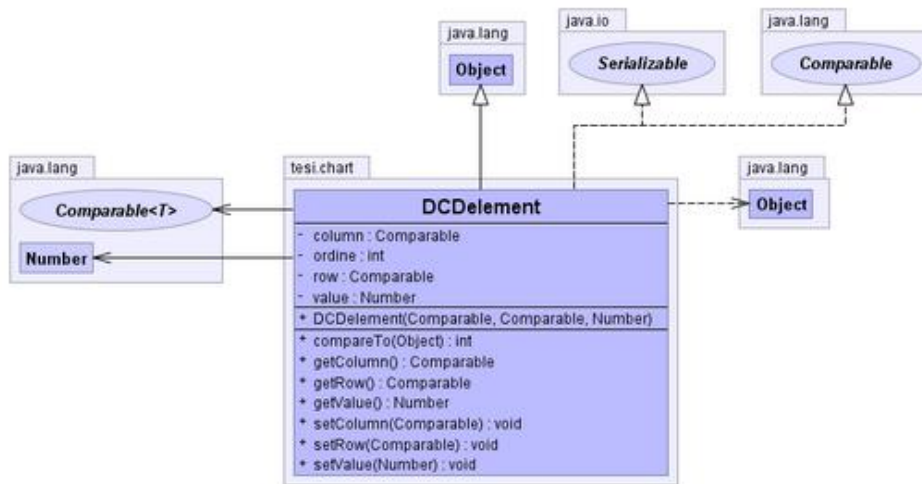


Figura III-24: Diagramma UML della classe DCDelement

### III.7.1.3. Classe DPDelement

Gli oggetti della classe rappresentano un elemento del dataset *DefaultPieDataset* per l'ordinamento secondo l'ordine crescente/decescente dei grafici a torta creati nella classe *ChartUtil*. Diagramma UML della classe:

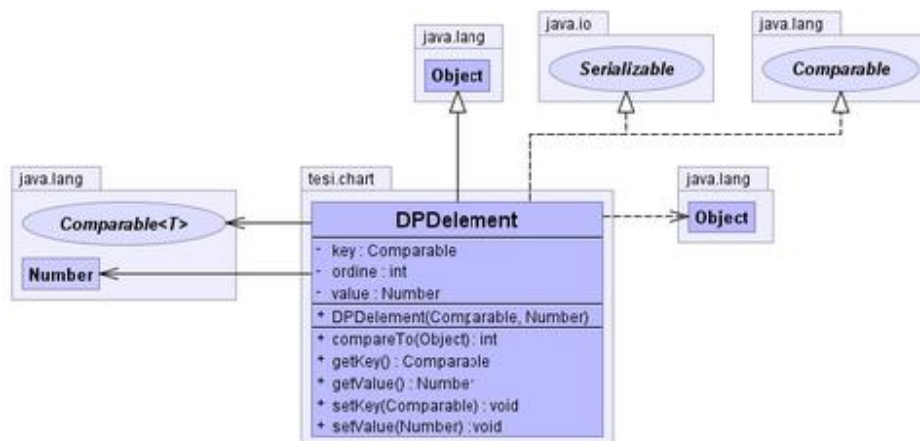


Figura III-25: Diagramma UML della classe DPDelement

### III.7.1.4. Classe TimestampSplit

La classe, di tipo *JavaBean*, consente la separazione in anno, mese, giorno, ora e secondo di una marcatura temporale *timestamp* per l'inserimento nei dataset necessari per la costruzione dei grafici serie temporali (*Timeseries chart*). Diagramma UML della classe:



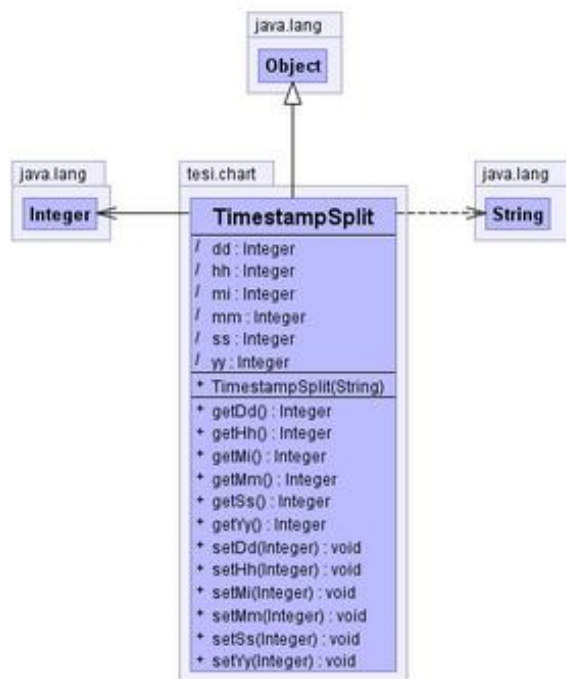


Figura III-26: Diagramma UML della classe TimestampSplit

### III.7.2. Package tesi.db.alarm

Il package contiene le classi, facenti parte del Model, per la gestione degli allarmi del sistema.

Le classi appartenenti al package *tesi.db.alarm* sono mostrate nella seguente figura:



Figura III-27: Classi del package tesi.alarm

Di seguito la descrizione delle classi del package *tesi.db.alarm*:

#### III.7.2.1. Classe Alarm

Classe JavaBean degli allarmi del sistema. Diagramma UML della classe:

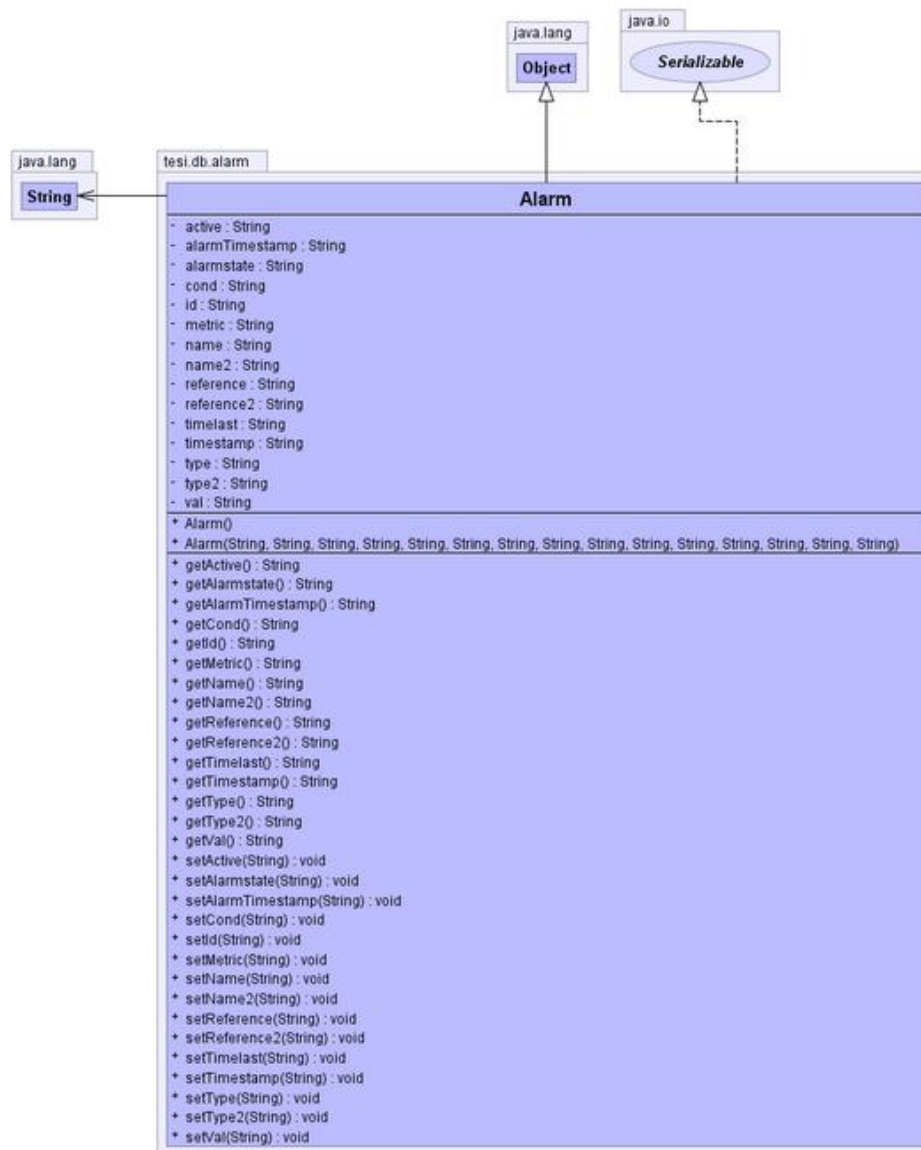


Figura III-28: Diagramma UML della classe Alarm

### III.7.2.2. Classe AlarmDbDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti Alarm accedendo al database del sistema. Diagramma UML della classe:

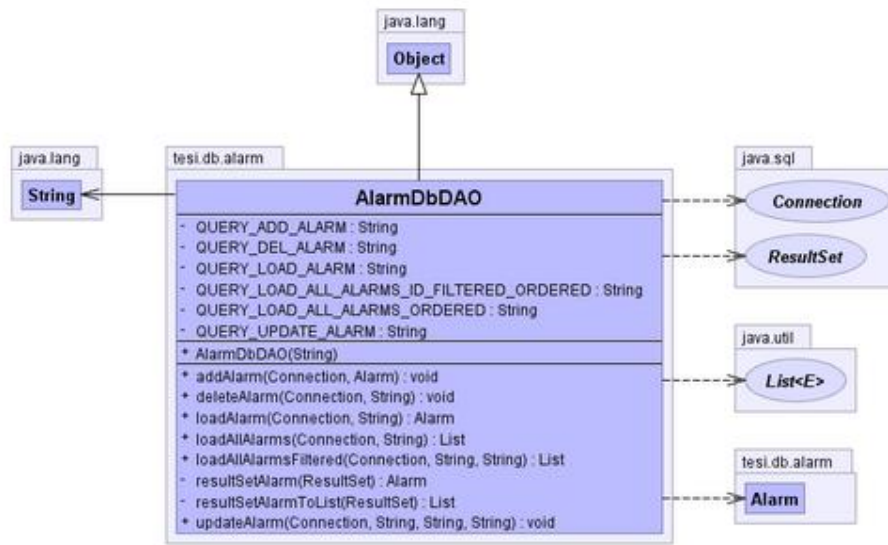


Figura III-29: Diagramma UML della classe AlarmDbDAO

### III.7.2.3. Classe AlarmDbManager

Classe di tipo Manager, per la gestione degli allarmi del sistema. Diagramma UML della classe:

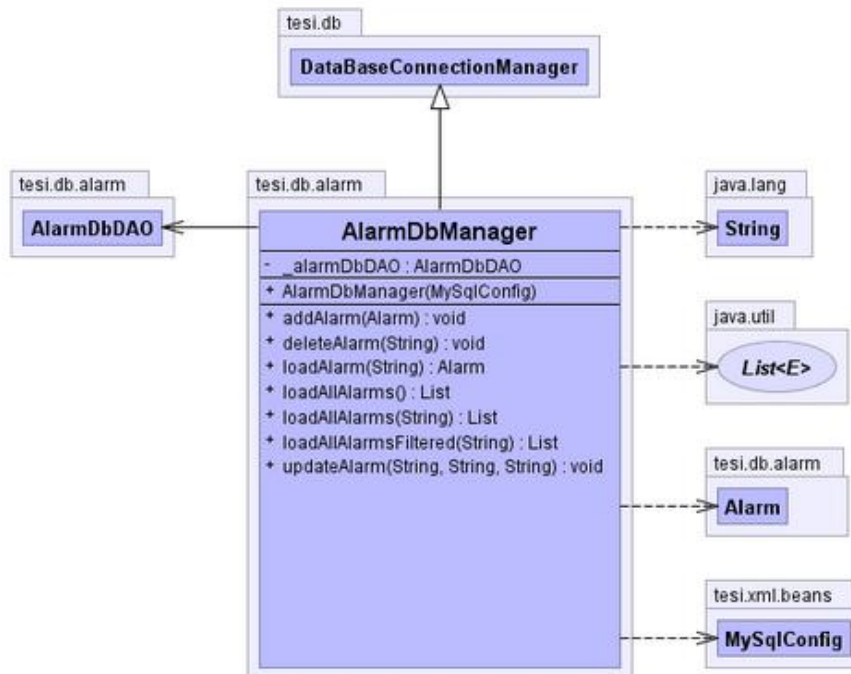


Figura III-30: Diagramma UML della classe AlarmDbManager

### III.7.3. Package tesi.db.ccrperformance

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance degli elementi del tipo ClusterComputeResource del Cloud.

Le classi appartenenti al package *tesi.db.ccrperformance* sono mostrate nella seguente figura:



Figura III-31: Classi del package *tesi.db.ccrperformance*

Di seguito la descrizione delle classi del package *tesi.db.ccrperformance*:

### III.7.3.1. Classe CcrPerformance

Classe JavaBean di un set di metriche e statistiche di performance di un elemento del tipo ClusterComputeResource del Cloud. Diagramma UML della classe:

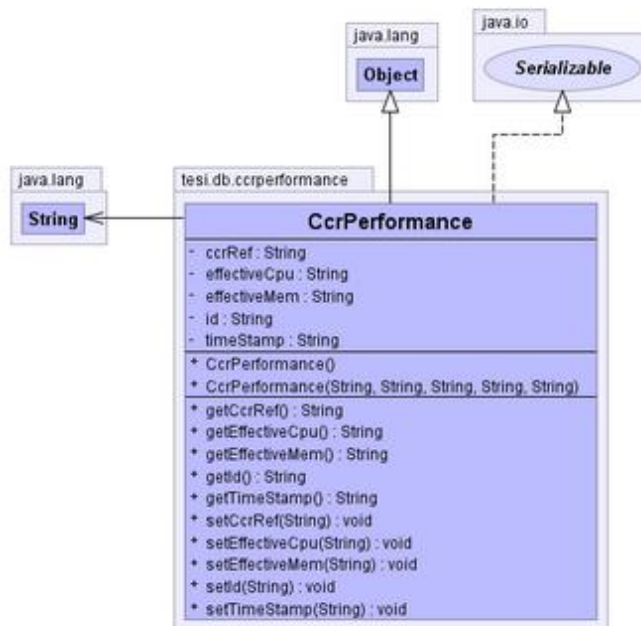


Figura III-32: Diagramma UML della classe *CcrPerformance*

### III.7.3.2. Classe CcrPerformanceDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti *CcrPerformance* accedendo al database del sistema. Diagramma UML della classe:

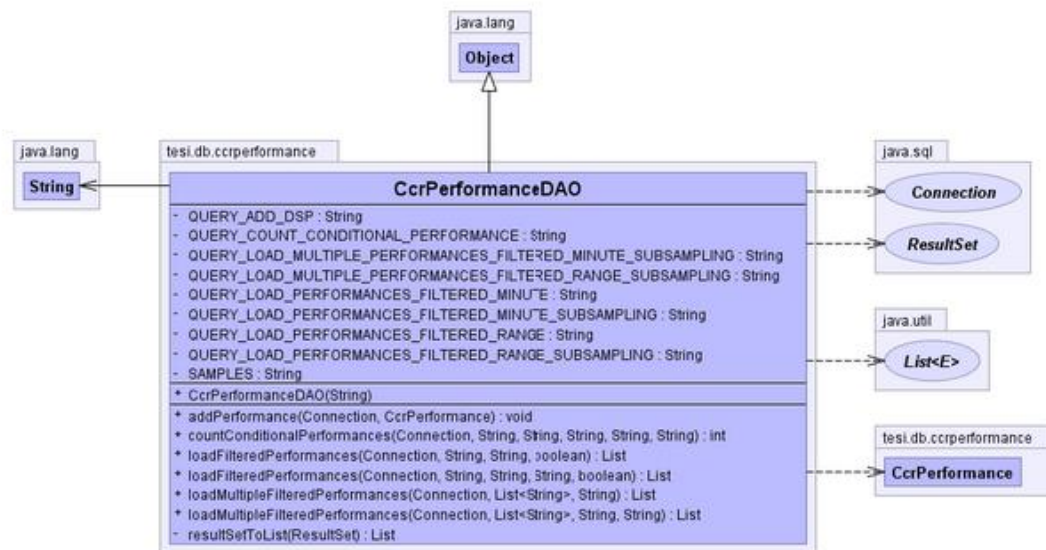


Figura III-33: Diagramma UML della classe CcrPerformanceDAO

### III.7.3.3. Classe CcrPerformanceManager

Classe di tipo Manager, per la gestione degli oggetti CcrPerformance del sistema. Diagramma UML della classe:

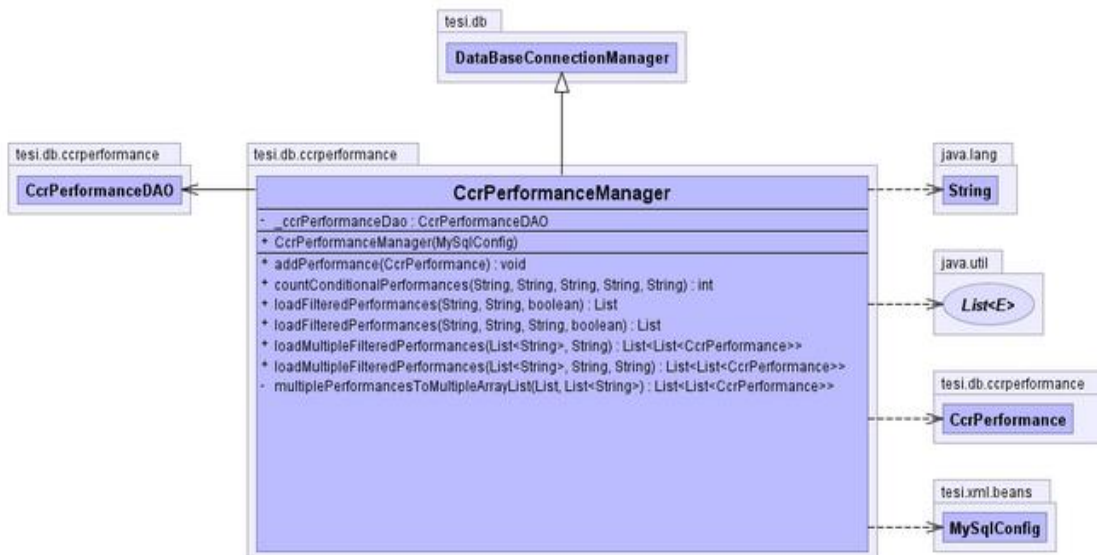


Figura III-34: Diagramma UML della classe CcrPerformanceManager

### III.7.4. Package tesi.db.condition

Il package contiene le classi, facenti parte del Model, per la gestione delle condizioni disponibili sugli allarmi del sistema.

Le classi appartenenti al package *tesi.db.condition* sono mostrate nella seguente figura:

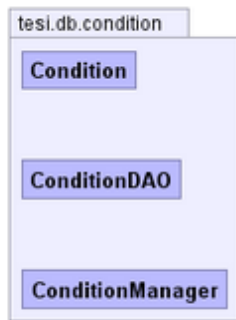


Figura III-35: Classi del package *tesi.db.condition*

Di seguito la descrizione delle classi del package *tesi.db.condition*:

#### III.7.4.1. Classe Condition

Classe JavaBean degli operatori condizionali attivabili sugli allarmi del sistema. Diagramma UML della classe:

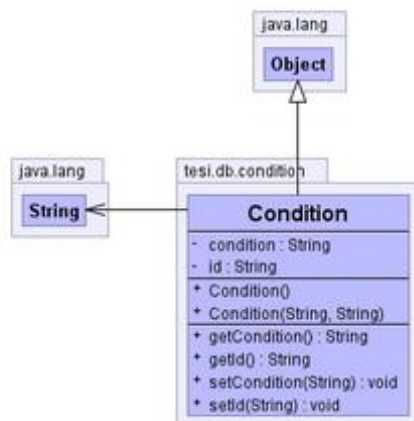


Figura III-36: Diagramma UML della classe Condition

#### III.7.4.2. Classe ConditionDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti Condition accedendo al database del sistema. Diagramma UML della classe:

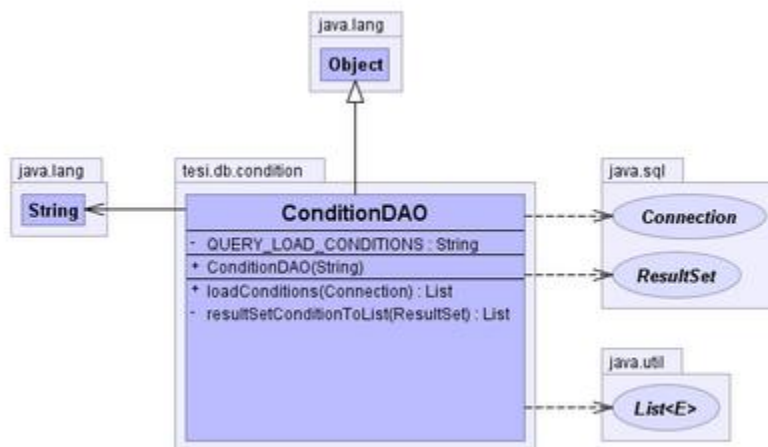


Figura III-37: Diagramma UML della classe ConditionDAO

### III.7.4.3. Classe ConditionManager

Classe di tipo Manager, per la gestione degli oggetti Condition del sistema. Diagramma UML della classe:

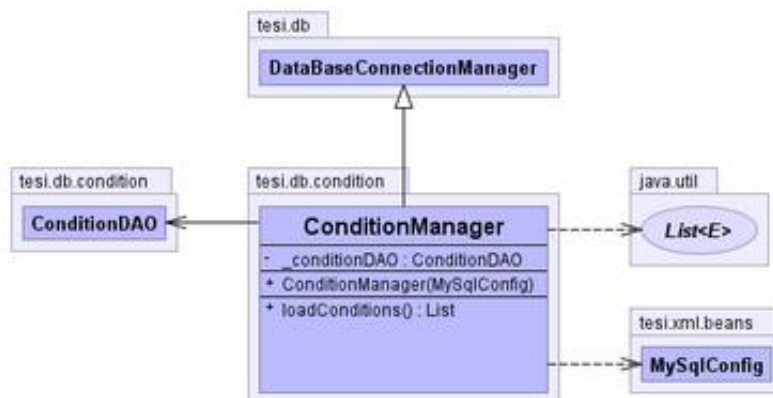


Figura III-38: Diagramma UML della classe ConditionManager

### III.7.5. Package tesi.db.dsperformance

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance degli elementi del Cloud del tipo Datastore.

Le classi appartenenti al package *tesi.db.dsperformance* sono mostrate nella seguente figura:



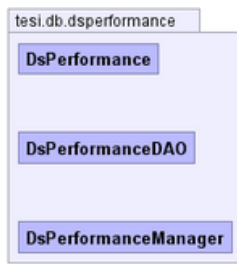


Figura III-39: Classi del package *tesi.db.dsperformance*

Di seguito la descrizione delle classi del package *tesi.db.dsperformance*:

### III.7.5.1. Classe DsPerformance

Classe JavaBean di un set di metriche di performance di un elemento del Cloud del tipo Datastore. Diagramma UML della classe:



Figura III-40: Diagramma UML della classe *DsPerformance*



### III.7.5.2. Classe DsPerformanceDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti DsPerformance accedendo al database del sistema. Diagramma UML della classe:

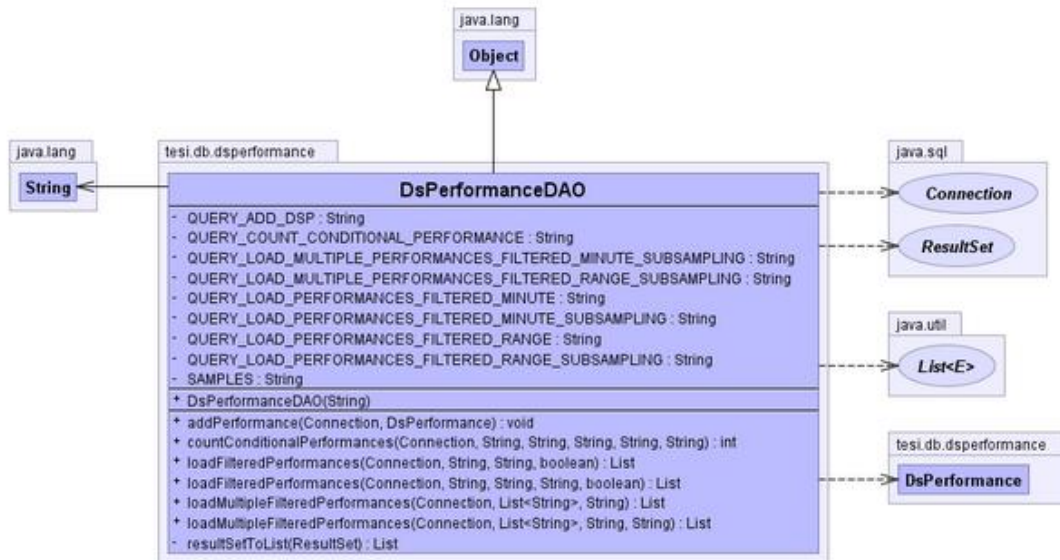


Figura III-41: Diagramma UML della classe DsPerformanceDAO

### III.7.5.3. Classe DsPerformanceManager

Classe di tipo Manager, per la gestione degli oggetti DsPerformance del sistema. Diagramma UML della classe:

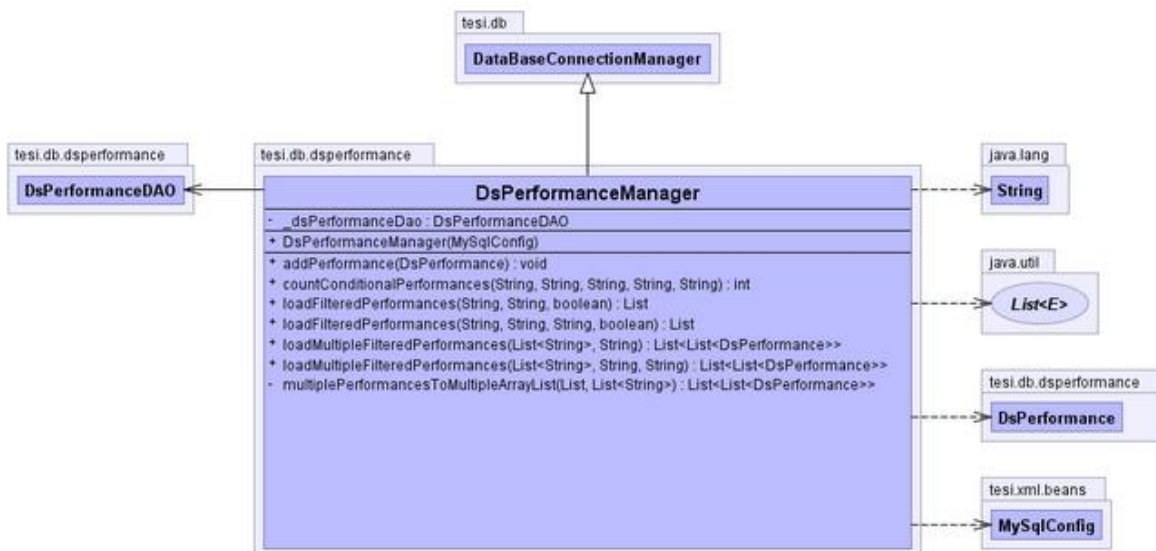


Figura III-42: Diagramma UML della classe DsPerformanceManager

### III.7.6. Package *tesi.db.hsperformance*

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance degli elementi del Cloud del tipo HostSystem.

Le classi appartenenti al package *tesi.db.hsperformance* sono mostrate nella seguente figura:



Figura III-43: Classi del package *tesi.db.hsperformance*

Di seguito la descrizione delle classi del package *tesi.db.hsperformance*:

#### III.7.6.1. Classe *HsPerformance*

Classe JavaBean di un set di metriche di performance di un elemento del Cloud del tipo HostSystem. Diagramma UML della classe:



### III.7.6.2. Classe HsPerformanceDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti HsPerformance accedendo al database del sistema. Diagramma UML della classe:

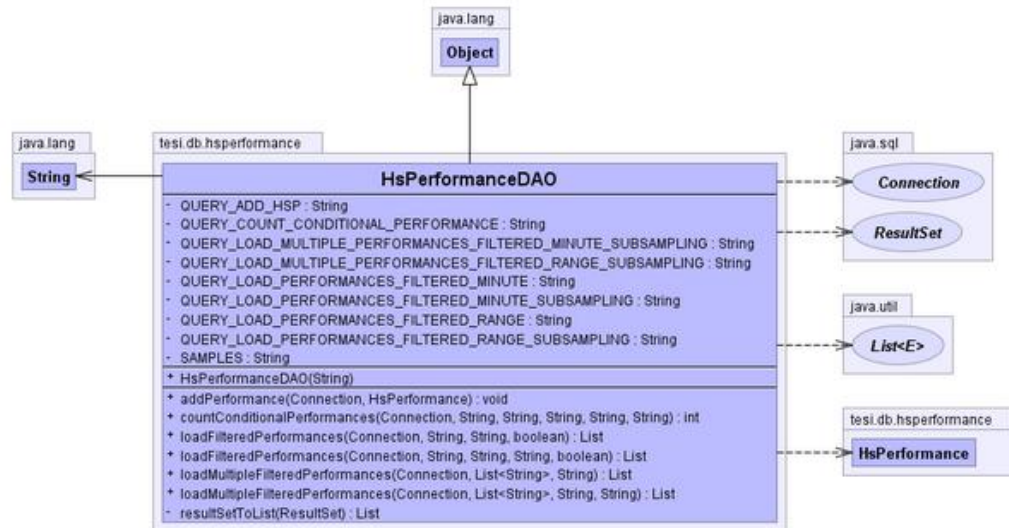


Figura III-45: Diagramma UML della classe HsPerformanceDAO

### III.7.6.3. Classe HsPerformanceManager

Classe di tipo Manager, per la gestione degli oggetti HsPerformance del sistema. Diagramma UML della classe:

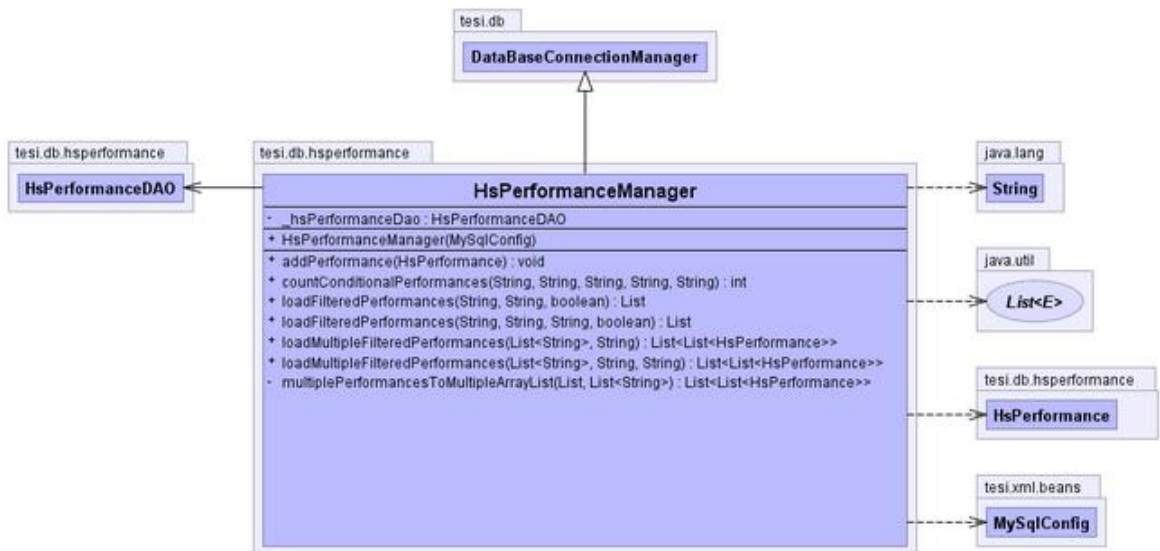


Figura III-46: Diagramma UML della classe HsPerformanceManager

### III.7.7. Package *tesi.db.inventory*

Il package contiene le classi, facenti parte del Model, per la gestione degli elementi dell'inventario del Cloud.

Le classi appartenenti al package *tesi.db.inventory* sono mostrate nella seguente figura:



Figura III-47: Classi del package *tesi.db.inventory*

Di seguito la descrizione delle classi del package *tesi.db.inventory*:

#### III.7.7.1. Classe *InventoryDbDAO*

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti *Item* accedendo al database del sistema. Diagramma UML della classe:

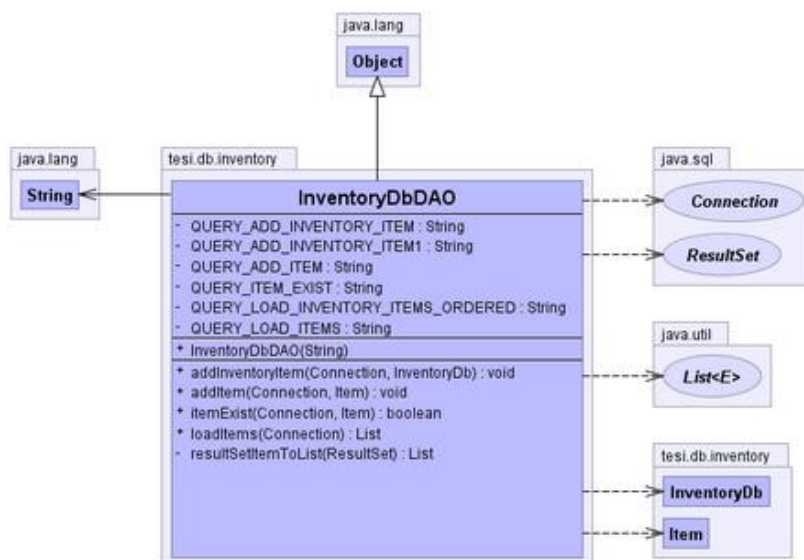


Figura III-48: Diagramma UML della classe *InventoryDbDAO*

### III.7.7.2. Classe InventoryDbManager

Classe di tipo Manager, per la gestione degli oggetti Item del sistema. Diagramma UML della classe:

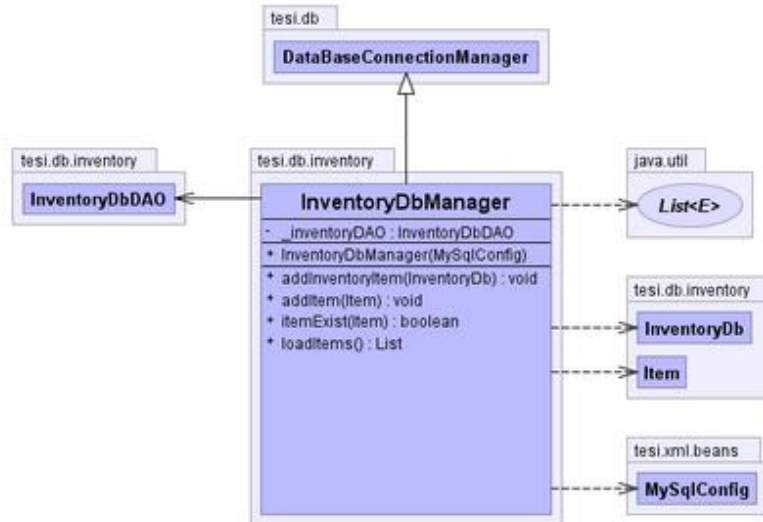


Figura III-49: Diagramma UML della classe InventoryDbManager

### III.7.7.3. Classe Item

Gli oggetti della classe, di tipo JavaBean, rappresentano gli elementi dell'inventario del Cloud. Diagramma UML della classe:

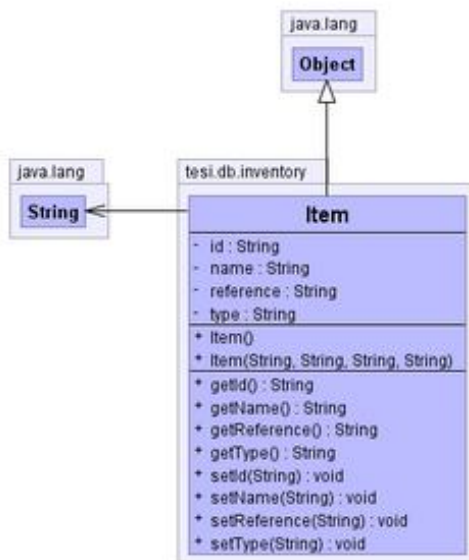


Figura III-50: Diagramma UML della classe Item

### III.7.8. Package *tesi.db.metric*

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance associate ai diversi elementi del Cloud.

Le classi appartenenti al package *tesi.db.metric* sono mostrate nella seguente figura:

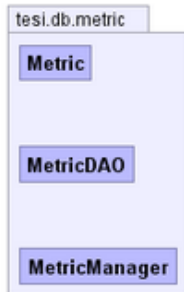


Figura III-51: Classi del package *tesi.db.metric*

Di seguito la descrizione delle classi del package *tesi.db.metric*:

#### III.7.8.1. Classe *Metric*

Gli oggetti della classe, di tipo JavaBean, rappresentano le metriche di performance monitorate dal sistema. Diagramma UML della classe:

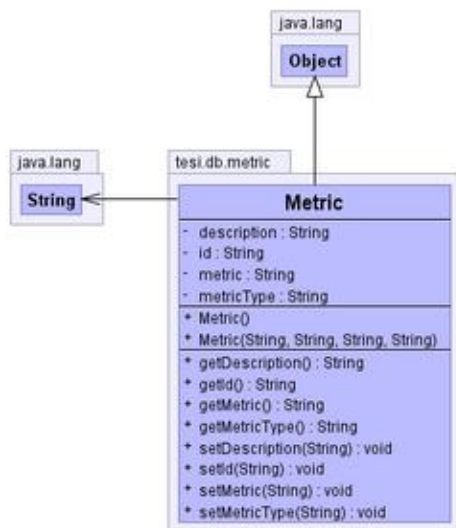


Figura III-52: Diagramma UML della classe *Metric*

#### III.7.8.2. Classe *MetricDAO*

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti *Metric* accedendo al database del sistema. Diagramma UML della classe:



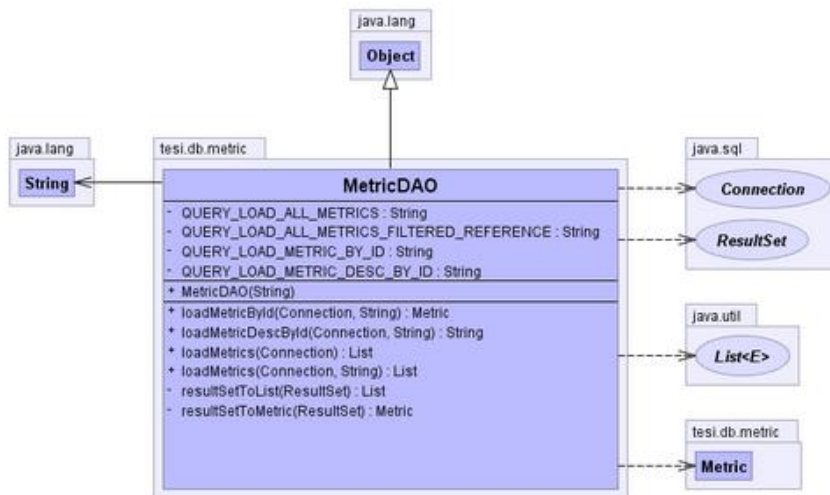


Figura III-53: Diagramma UML della classe MetricDAO

### III.7.8.3. Classe MetricManager

Classe di tipo Manager, per la gestione degli oggetti Metric del sistema. Diagramma UML della classe:

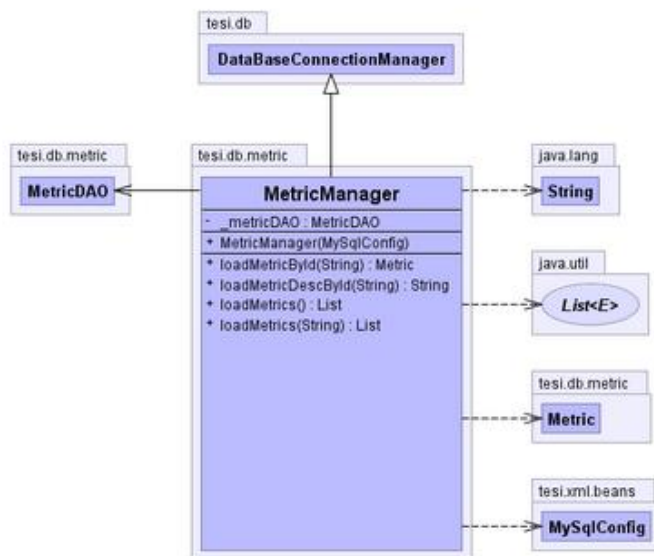


Figura III-54: Diagramma UML della classe MetricManager

### III.7.9. Package tesi.db.vmperspective

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance degli elementi del Cloud del tipo VirtualMachine.

Le classi appartenenti al package *tesi.db.vmperspective* sono mostrate nella seguente figura:



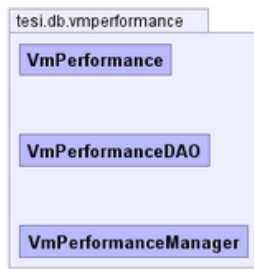


Figura III-55: Classi del package tesi.db.vmpformance

Di seguito la descrizione delle classi del package *tesi.db.vmpformance*:

### III.7.9.1. Classe VmPerformance

Classe JavaBean di un set di metriche di performance di un elemento VirtualMachine del Cloud. Diagramma UML della classe:

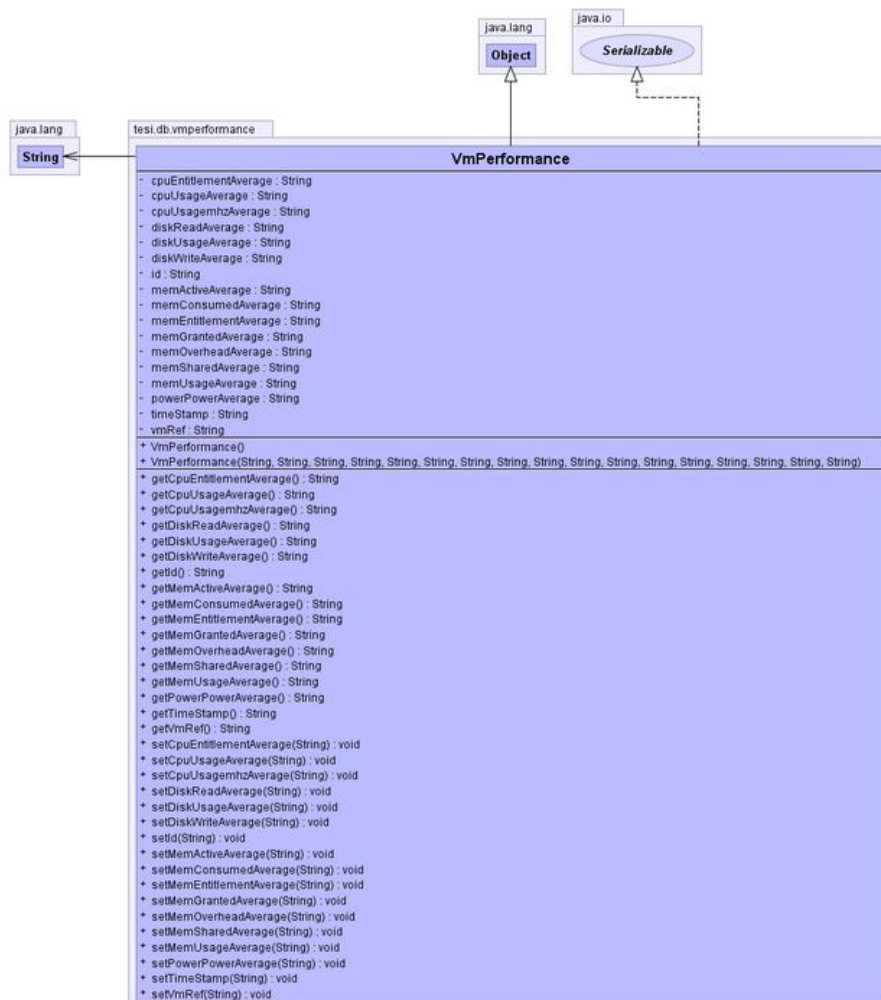


Figura III-56: Diagramma UML della classe VmPerformance

### III.7.9.2. Classe VmPerformanceDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti VmPerformance accedendo al database del sistema. Diagramma UML della classe:

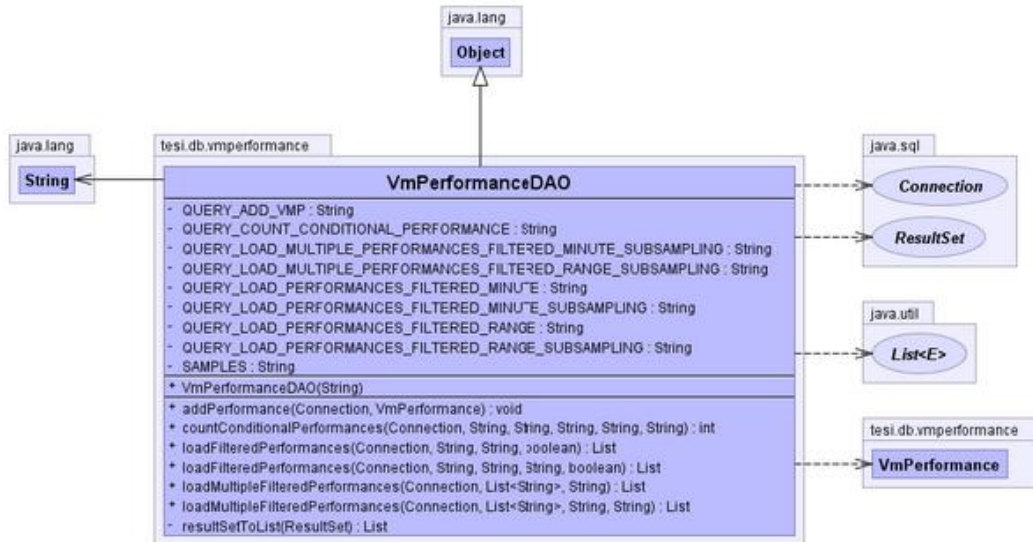


Figura III-57: Diagramma UML della classe VmPerformanceDAO

### III.7.9.3. Classe VmPerformanceManager

Classe di tipo Manager, per la gestione degli oggetti VmPerformance del sistema. Diagramma UML della classe:

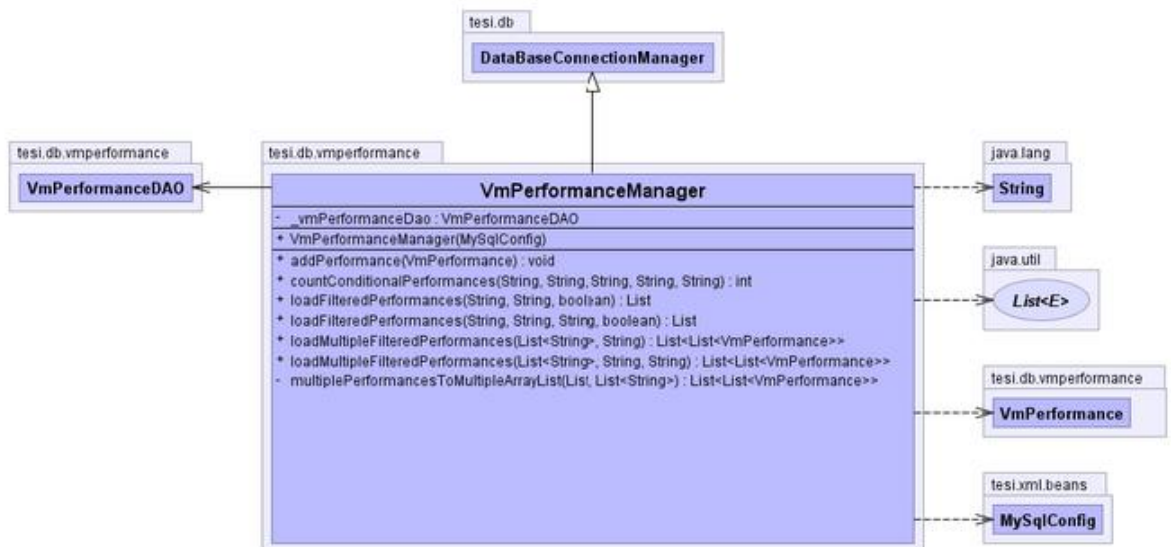


Figura III-58: Diagramma UML della classe VmPerformanceManager

### III.7.10. Package *tesi.db.vmsperformance*

Il package contiene le classi, facenti parte del Model, per la gestione delle metriche di performance degli elementi, associati, del tipo Datastore e VirtualMachine del Cloud.

Le classi appartenenti al package *tesi.db.vmsperformance* sono mostrate nella seguente figura:

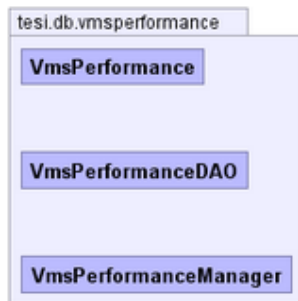


Figura III-59: Classi del package *tesi.db.vmsperformance*

Di seguito la descrizione delle classi del package *tesi.db.vmsperformance*:

#### III.7.10.1. Classe *VmsPerformance*

Classe JavaBean di un set di metriche di performance degli elementi Datastore e VirtualMachine del Cloud. Diagramma UML della classe:

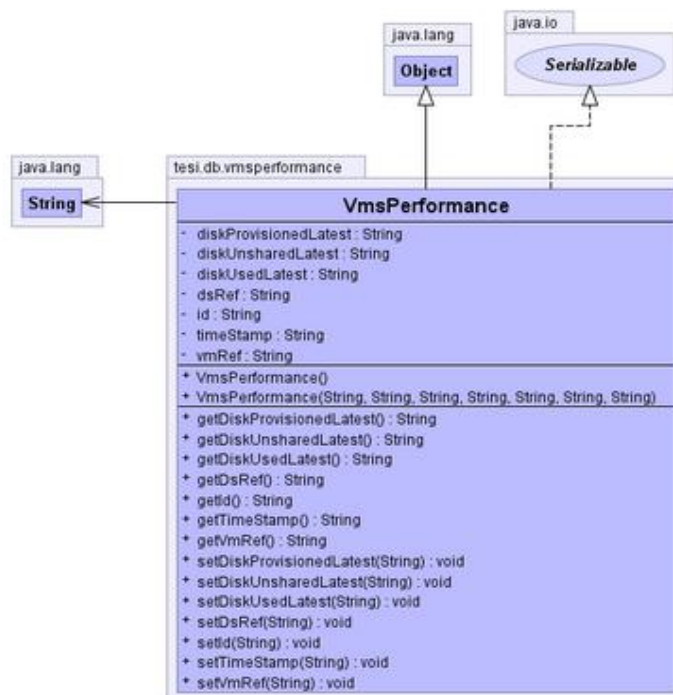


Figura III-60: Diagramma UML della classe *VmsPerformance*

### III.7.10.2. Classe VmsPerformanceDAO

Classe di tipo Data Access Object, consente l'elaborazione degli oggetti VmsPerformance accedendo al database del sistema. Diagramma UML della classe:

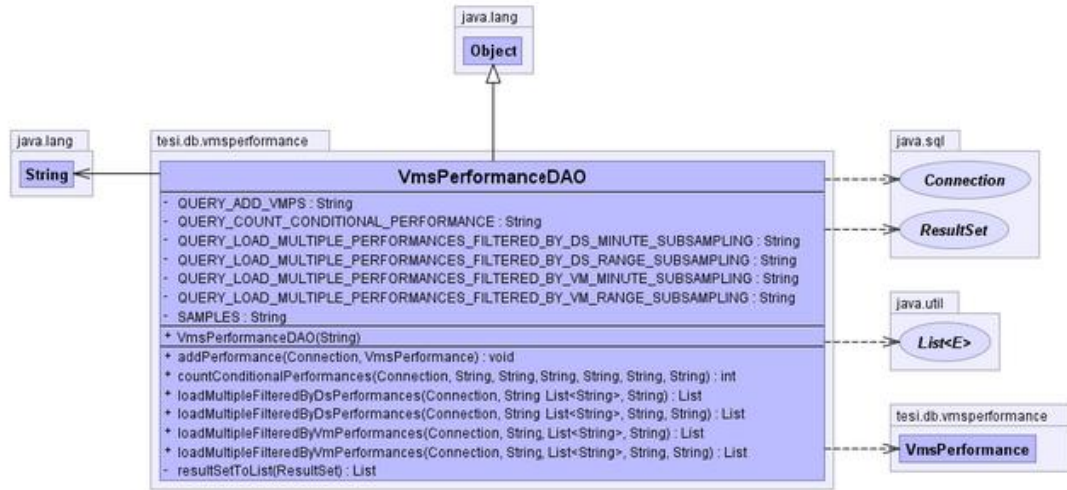


Figura III-61: Diagramma UML della classe VmsPerformanceDAO

### III.7.10.3. Classe VmsPerformanceManager

Classe di tipo Manager, per la gestione degli oggetti VmsPerformance del sistema. Diagramma UML della classe:

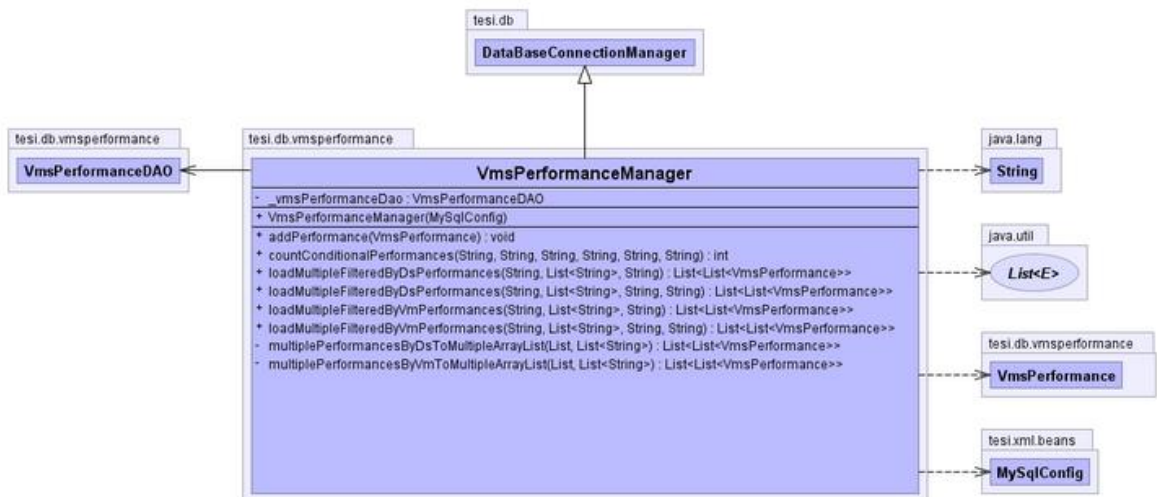


Figura III-62: Diagramma UML della classe VmsPerformanceManager

### III.7.11. Package *tesi.directory*

La classe del package consente la gestione delle directory temporanee create dall'applicazione per la visualizzazione dei grafici delle metriche e delle statistiche di performance.

Le classi appartenenti al package *tesi.directory* sono mostrate nella seguente figura:



Figura III-63: Classi del package *tesi.directory*

Di seguito la descrizione delle classi del package *tesi.directory*:

#### III.7.11.1. Classe *DirUtil*

Classe di gestione delle directory temporanee create dall'applicazione per la visualizzazione dei grafici delle metriche e statistiche di performance. Diagramma UML della classe:

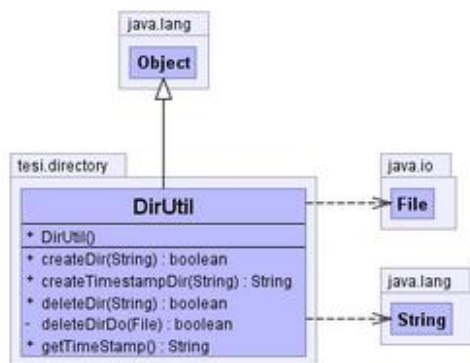


Figura III-64: Diagramma UML della classe *DirUtil*

### III.7.12. Package *tesi.mail*

Il package contiene le classi che permettono l'invio degli alert degli allarmi impostati dall'utente.

Le classi appartenenti al package *tesi.mail* sono mostrate nella seguente figura:



Figura III-65: Classi del package *tesi.mail*

Di seguito la descrizione delle classi del package *tesi.mail*:

### III.7.12.1. Classe AlarmMail

La classe, di tipo Manager, consente la creazione e l'invio delle email di alert degli allarmi del sistema. Diagramma UML della classe:

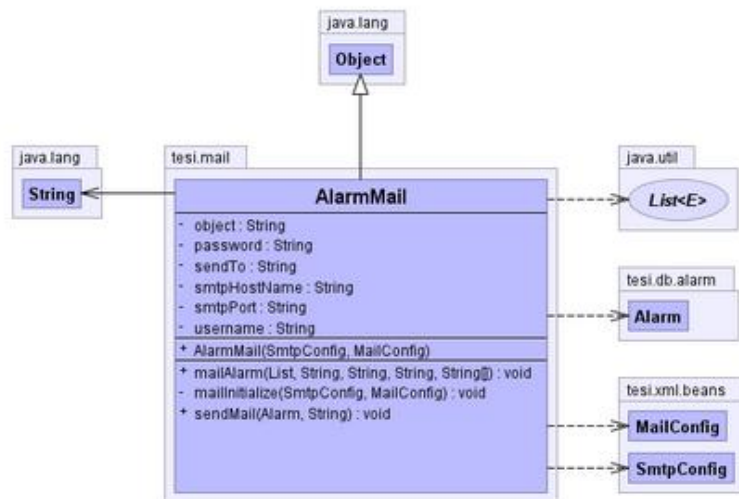


Figura III-66: Diagramma UML della classe AlarmMail

### III.7.12.2. Classe MailSender

La classe permette l'invio di email. Diagramma UML della classe:

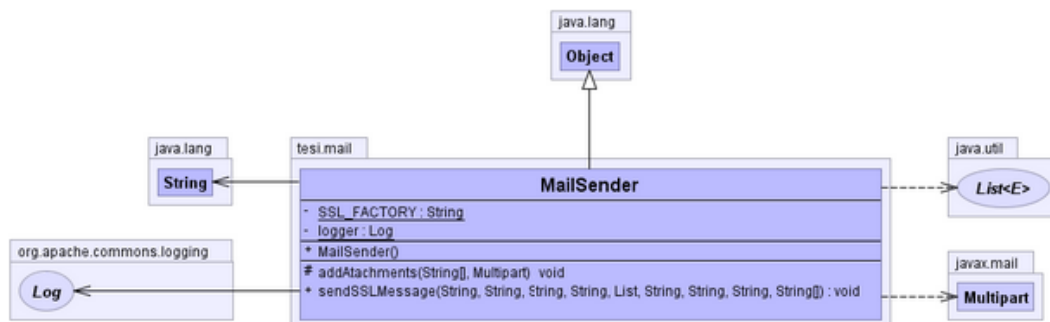


Figura III-67: Diagramma UML della classe MailSender

### III.7.13. Package tesi.monitoring

Il package contiene le classi atte alla gestione dei thread per il monitoring e l'alarmed del sistema.

Le classi appartenenti al package *tesi.monitoring* sono mostrate nella seguente figura:



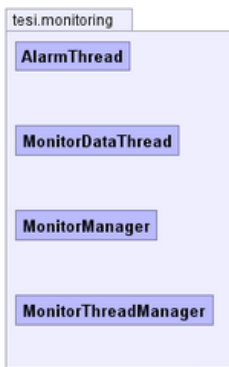


Figura III-68: Classi del package `tesi.monitoring`

Di seguito la descrizione delle classi del package `tesi.monitoring`:

### III.7.13.1. Classe `AlarmThread`

La classe consente di verificare se gli allarmi attivi nel sistema sono verificati e, in caso affermativo, l'invio di una email di alert all'indirizzo configurato nel file di configurazione del sistema. Diagramma UML della classe:

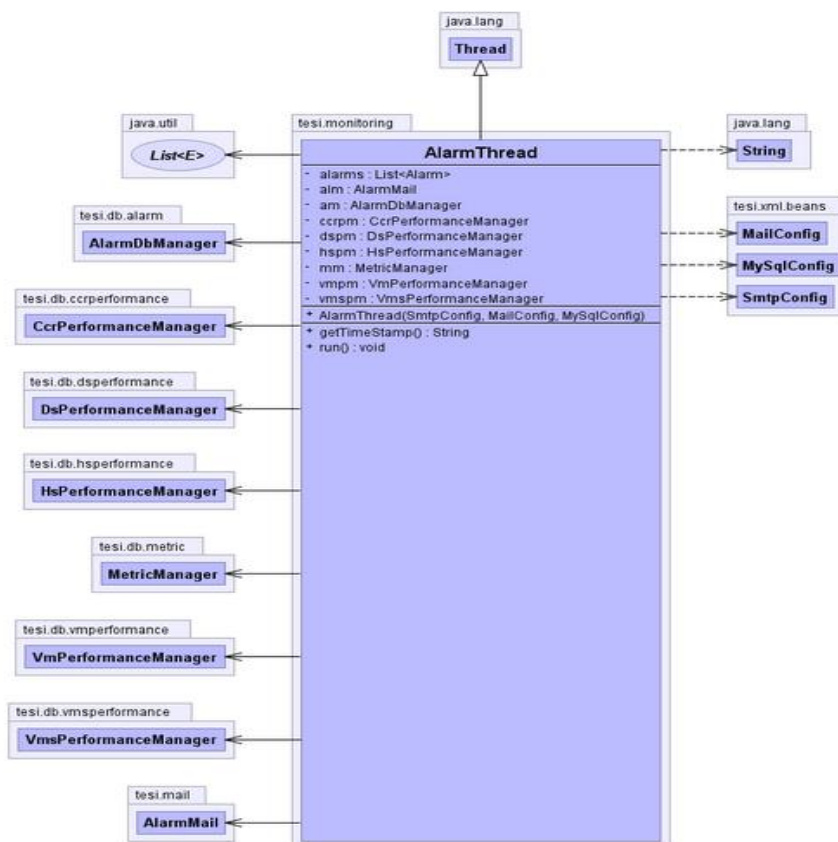


Figura III-69: Diagramma UML della classe `AlarmThread`

### III.7.13.2. Classe MonitorDataThread

La classe consente il recupero di tutte le metriche di performance disponibili per tutti gli elementi attivi del Cloud. Diagramma UML della classe:

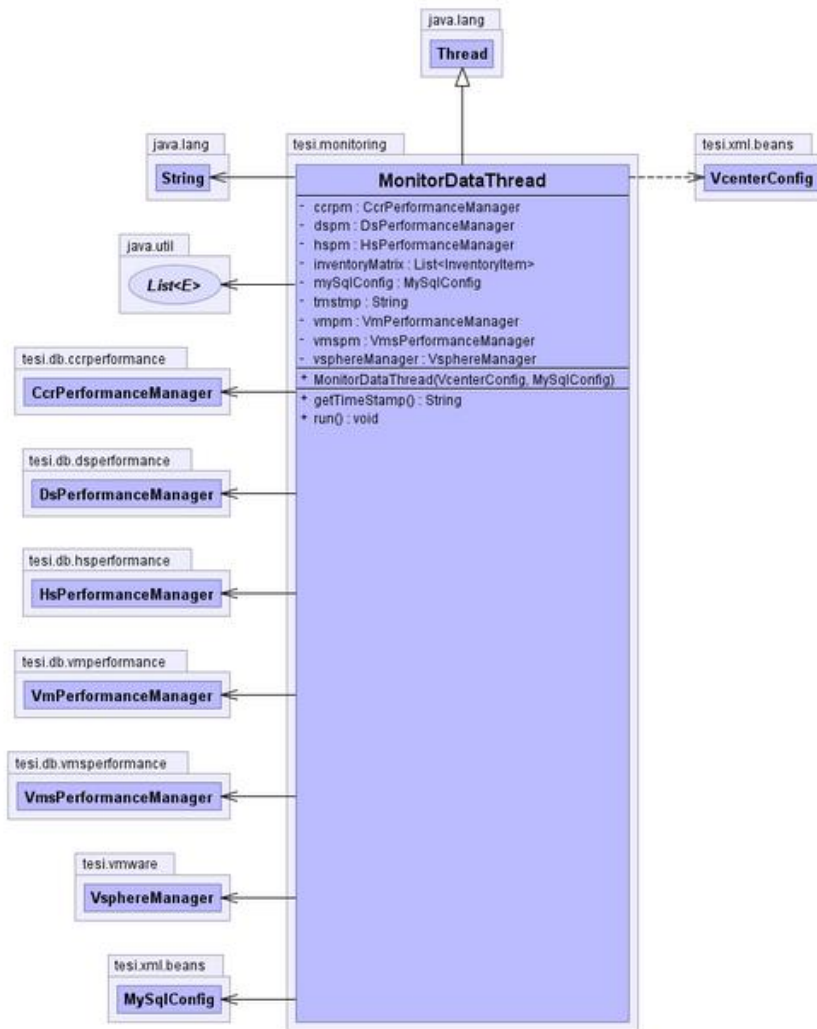


Figura III-70: Diagramma UML della classe MonitorDataThread

### III.7.13.3. Classe MonitorManager

La classe consente l'avvio e lo stop dei cicli di monitoring e alarming del sistema. Diagramma UML della classe:



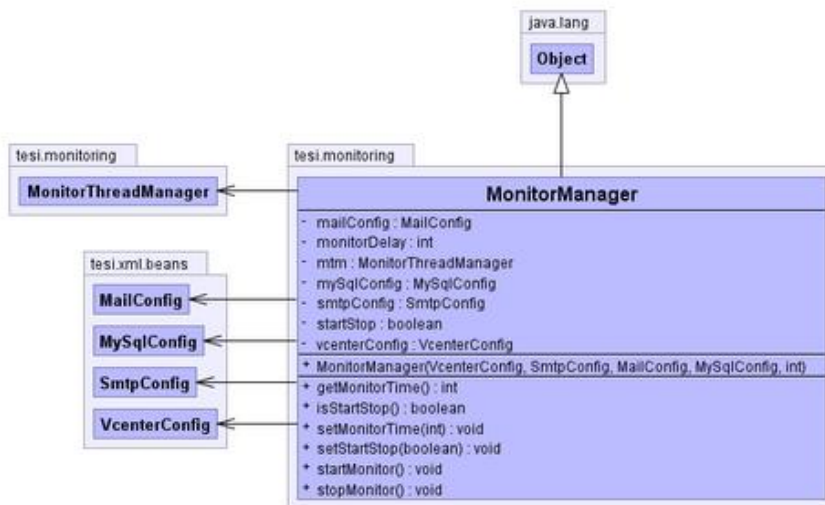


Figura III-71: Diagramma UML della classe MonitorManager

#### III.7.13.4. Classe MonitorThreadManager

La classe consente la creazione dei singoli thread per il monitoring e l'alarmino del sistema con frequenza impostata nel file di configurazione del sistema. Diagramma UML della classe:

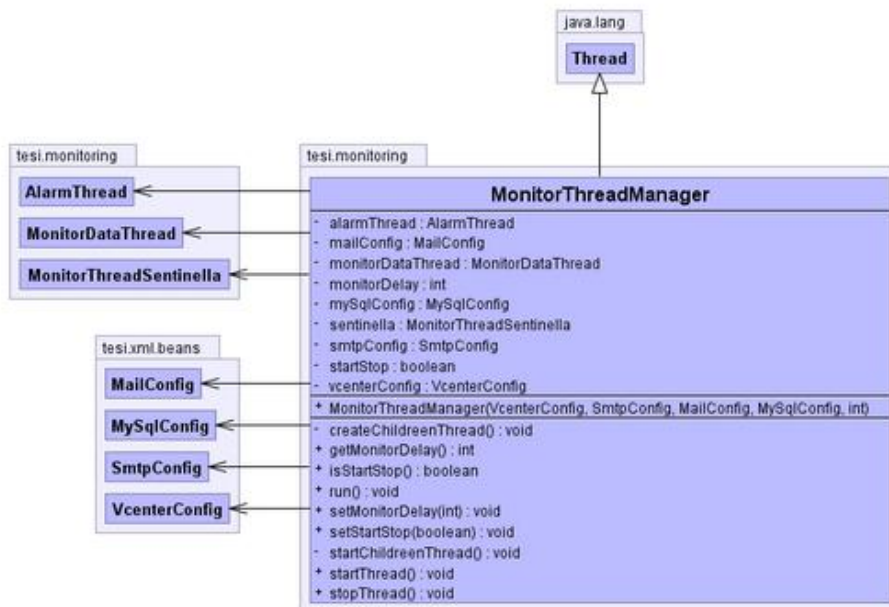


Figura III-72: Diagramma UML della classe MonitorThreadManager

### III.7.14. Package *tesi.security*

Il package contiene le classi utili per la crittografia delle password presenti nel file di configurazione del sistema.

Le classi appartenenti al package *tesi.security* sono mostrate nella seguente figura:



Figura III-73: Classi del package *tesi.security*

Di seguito la descrizione delle classi del package *tesi.security*:

#### III.7.14.1. Classe *DesEncrypter*

La classe consente la codifica e la decodifica DES (Data Encryption Standard) di una stringa. Diagramma UML della classe:

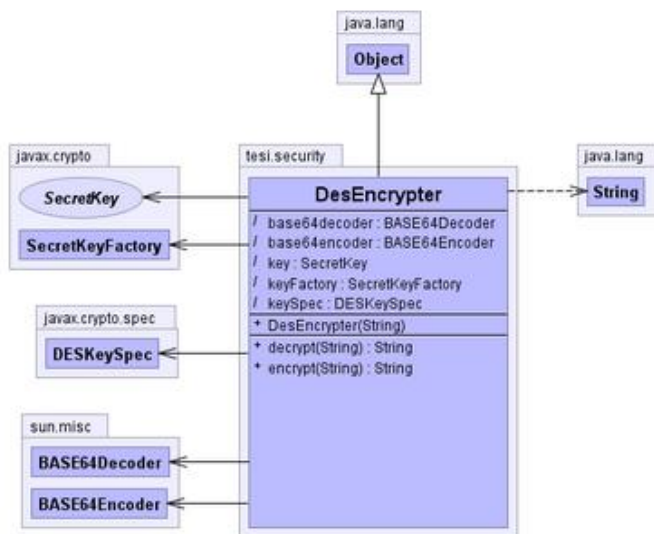


Figura III-74: Diagramma UML della classe *DesEncrypter*

#### III.7.14.2. Classe *TDesEncrypter*

La classe consente la codifica e la decodifica Triple DES (Data Encryption Standard) di una stringa. Diagramma UML della classe:

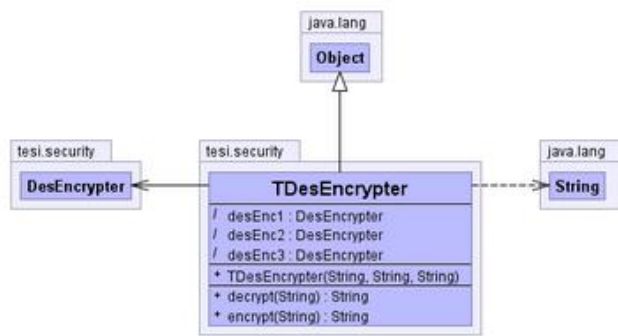


Figura III-75: Diagramma UML della classe TDesEncrypter

### III.7.15. Package *tesi.vmware*

Il package contiene le classi per l'accesso al server vSphere vCenter, il recupero e l'elaborazione dell'inventario del Cloud, il recupero delle proprietà degli elementi dell'inventario ed il recupero delle metriche di performance degli stessi elementi.

Le classi appartenenti al package *tesi.vmware* sono mostrate nella seguente figura:



Figura III-76: Classi del package *tesi.vmware*

Di seguito la descrizione delle classi del package *tesi.vmware*:

### III.7.15.1. Classe PropertyManager

La classe consente il recupero delle proprietà relative ai diversi elementi dell'inventario del Cloud. Diagramma UML della classe:

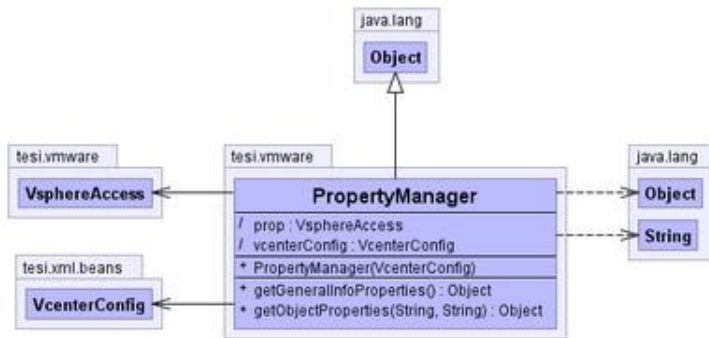


Figura III-77: Diagramma UML della classe PropertyManager

### III.7.15.2. Classe TrustAllTrustManager

La classe consente l'autenticazione del sistema nel vSphere vCenter server. Diagramma UML della classe:

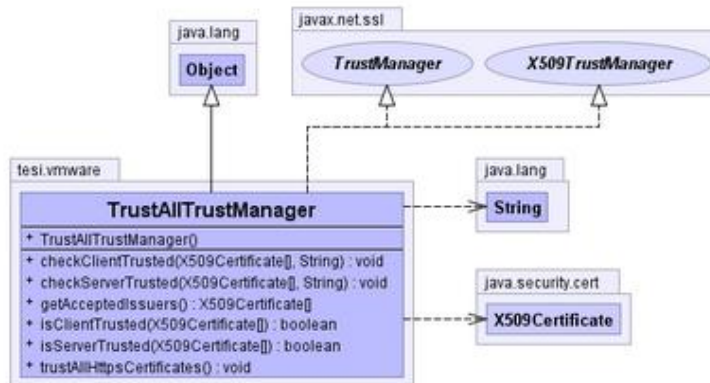


Figura III-78: Diagramma UML della classe TrustAllTrustManager

### III.7.15.3. Classe VsphereAccess

La classe consente l'accesso al server vSphere vCenter, il recupero dell'inventario del Cloud, il recupero delle proprietà degli elementi dell'inventario ed il recupero delle metriche di performance degli stessi elementi. Diagramma UML della classe:

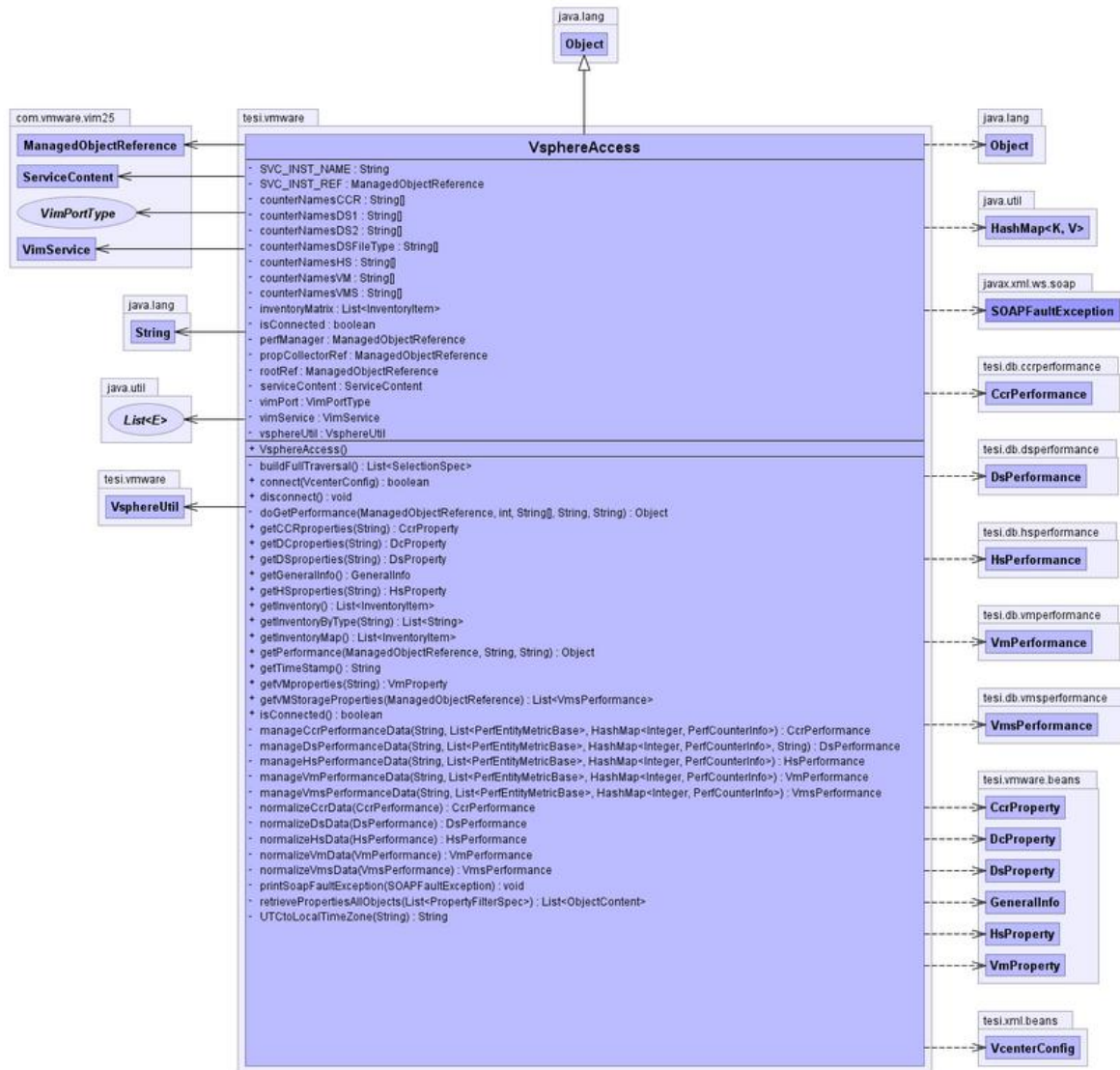


Figura III-79: Diagramma UML della classe VsphereAccess

### III.7.15.4. Classe VsphereManager

Classe, di tipo manager, per la gestione delle funzionalità di recupero dell'inventario del Cloud, di recupero delle proprietà degli elementi dell'inventario e di recupero delle metriche di performance degli stessi elementi. Diagramma UML della classe:

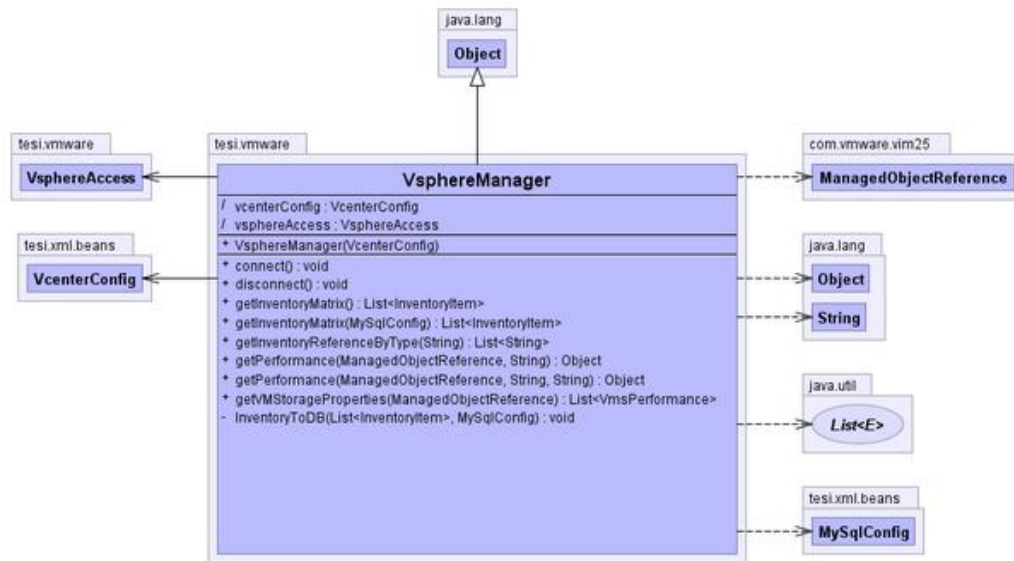


Figura III-80: Diagramma UML della classe VsphereManager

### III.7.15.5. Classe VsphereUtil

La classe contiene le funzionalità di elaborazione delle strutture dati che rappresentano l'inventario del Cloud, recuperato dal vSphere vCenter server. Diagramma UML della classe:

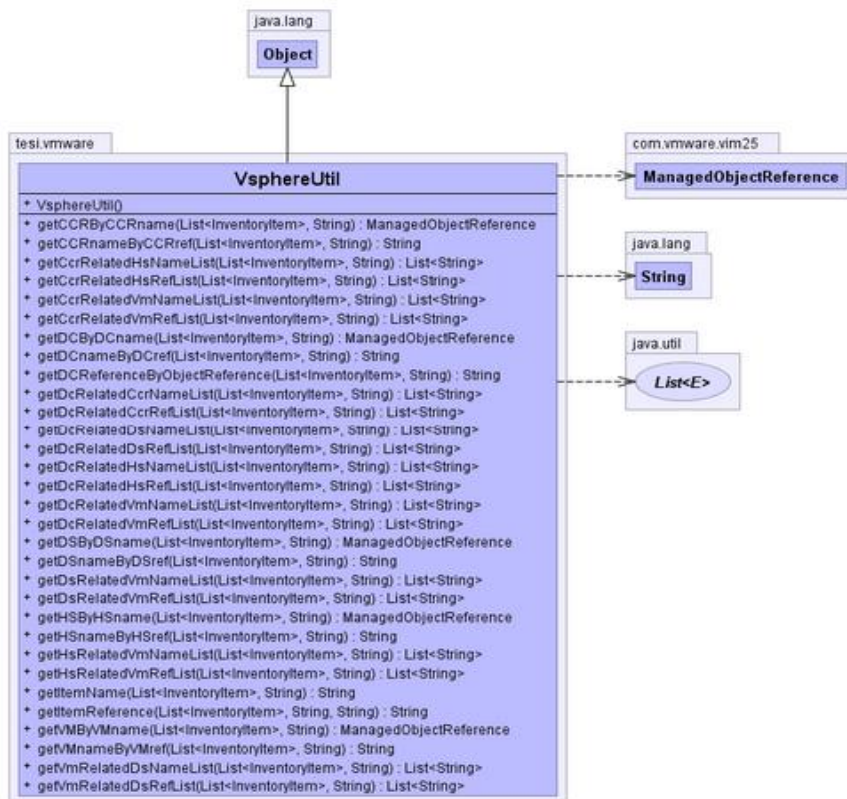


Figura III-81: Diagramma UML della classe VsphereUtil

### III.7.16. Package *tesi.vmware.beans*

Il package contiene le classi JavaBean contenenti le proprietà per le diverse tipologie degli elementi dell'inventario del Cloud.

Le classi appartenenti al package *tesi.vmware.beans* sono mostrate nella seguente figura:

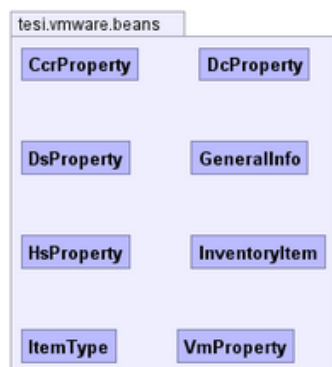


Figura III-82: Classi del package *tesi.vmware.beans*

Di seguito la descrizione delle classi del package *tesi.vmware.beans*:

#### III.7.16.1. Classe `CcrProperty`

Classe JavaBean, rappresenta le proprietà degli elementi `ClusterComputeResource` del Cloud. Diagramma UML della classe:





Figura III-83: Diagramma UML della classe CcrProperty



### III.7.16.2. Classe DcProperty

Classe JavaBean, rappresenta le proprietà degli elementi Datacenter del Cloud. Diagramma UML della classe:

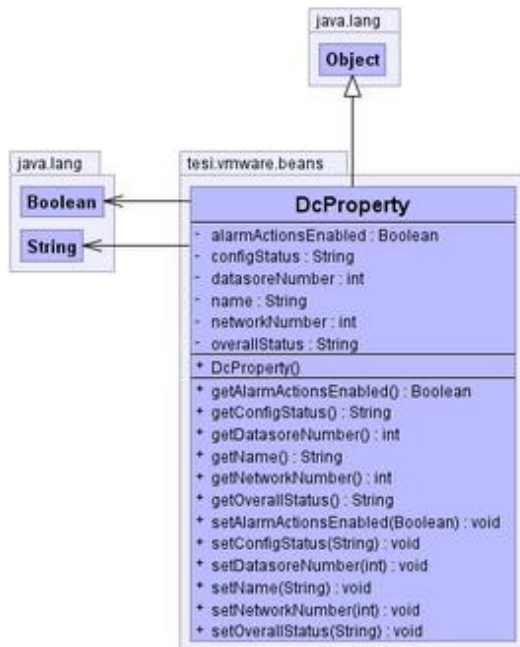


Figura III-84: Diagramma UML della classe DcProperty

### III.7.16.3. Classe DsProperty

Classe JavaBean, rappresenta le proprietà degli elementi Datastore del Cloud. Diagramma UML della classe:



Figura III-85: Diagramma UML della classe DsProperty

#### III.7.16.4. Classe GeneralInfo

Classe JavaBean, rappresenta le proprietà generali del vSphere vCenter server del Cloud. Diagramma UML della classe:

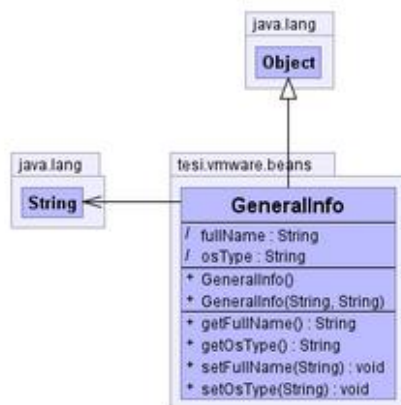


Figura III-86: Diagramma UML della classe GeneralInfo

### III.7.16.5. Classe HsProperty

Classe JavaBean, rappresenta le proprietà degli elementi HostSystem del Cloud. Diagramma UML della classe:



Figura III-87: Diagramma UML della classe HsProperty

### III.7.16.6. Classe InventoryItem

Classe JavaBean, rappresenta le proprietà degli elementi dell'inventario del Cloud per la costruzione della struttura ad albero dell'inventario stesso. Diagramma UML della classe:

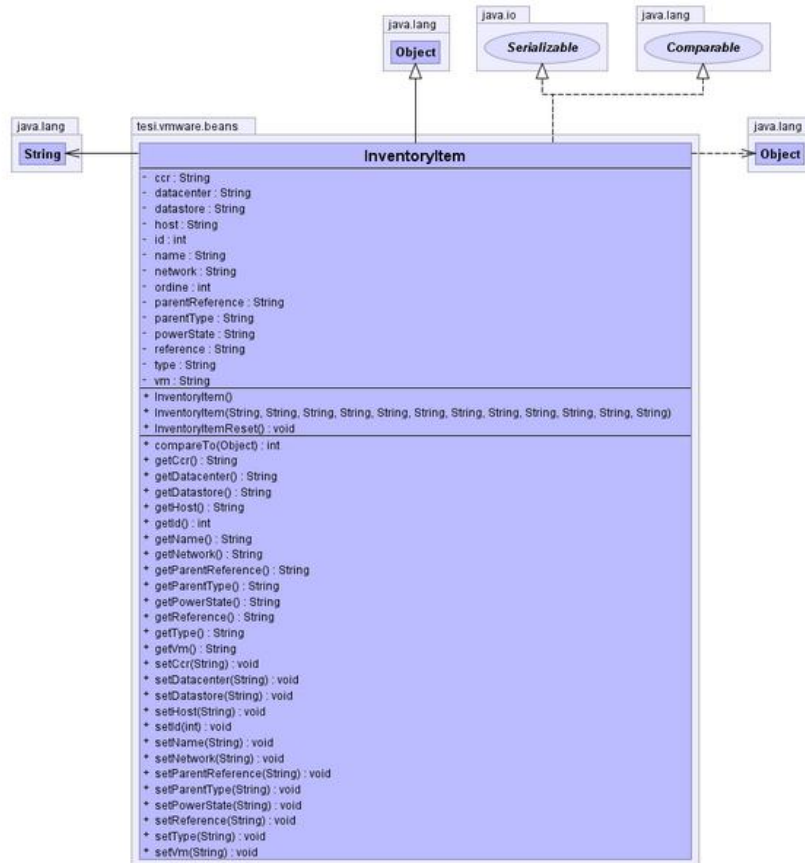


Figura III-88: Diagramma UML della classe InventoryItem

### III.7.16.7. Classe ItemType

Classe JavaBean, rappresenta la tipologia degli elementi dell'inventario del Cloud. Diagramma UML della classe:

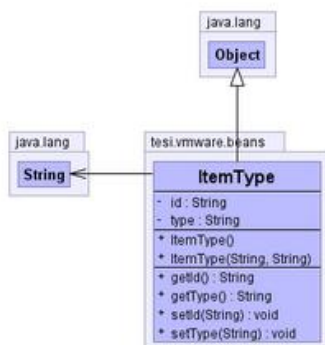


Figura III-89: Diagramma UML della classe ItemType

### III.7.16.8. Classe VmProperty

Classe JavaBean, rappresenta le proprietà degli elementi VirtualMachine del Cloud. Diagramma UML della classe:



Figura III-90: Diagramma UML della classe VmProperty

### III.7.17. Package tesi.xml

Il package contiene le classi per la gestione del file di configurazione xml *appconfig.xml* del sistema.

Le classi appartenenti al package *tesi.xml* sono mostrate nella seguente figura:

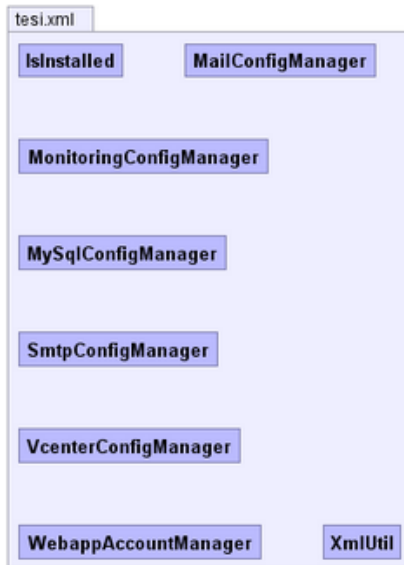


Figura III-91: Classi del package *tesi.xml*

Di seguito la descrizione delle classi del package *tesi.xml*:

#### III.7.17.1. Classe IsInstalled

Classe Manager, permette la gestione del campo `<installed>` del file di configurazione della Webapp, utilizzato per la codifica iniziale DES delle password del file di configurazione stesso. Diagramma UML della classe:

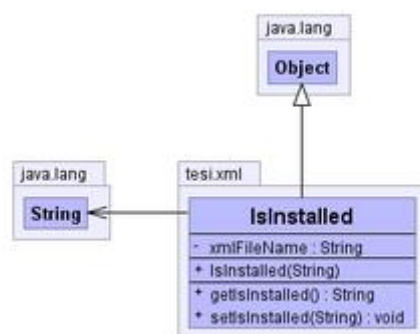


Figura III-92: Diagramma UML della classe `IsInstalled`



### III.7.17.2. Classe MailConfigManager

Classe Manager, permette la gestione delle impostazioni relative alle email spedite dal sistema di alarming. In particolare l'oggetto e l'indirizzo del destinatario della email inviata in caso di allarme. Diagramma UML della classe:

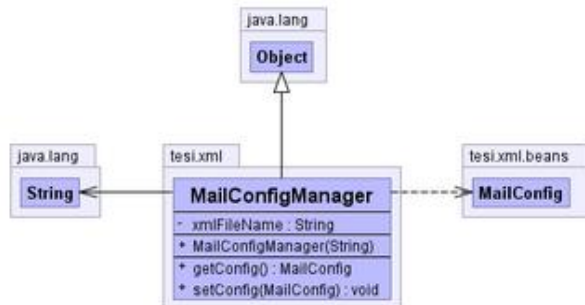


Figura III-93: Diagramma UML della classe MailConfigManager

### III.7.17.3. Classe MonitoringConfigManager

Classe Manager, rappresenta permette la gestione delle impostazioni dell'applicazione memorizzate nel file xml *appCongif.xml*. In particolare il parametro monitoringDelay che rappresenta il periodo tra due cicli di monitoring del sistema. Diagramma UML della classe:

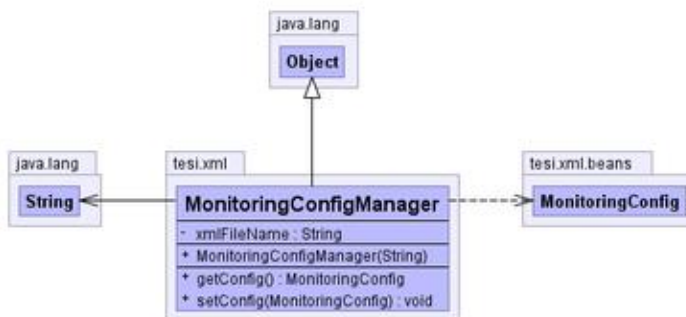


Figura III-94: Diagramma UML della classe MonitoringConfigManager

### III.7.17.4. Classe MySqlConfigManager

Classe Manager, permette la gestione delle credenziali di accesso al DBMS MySQL del sistema. Diagramma UML della classe:



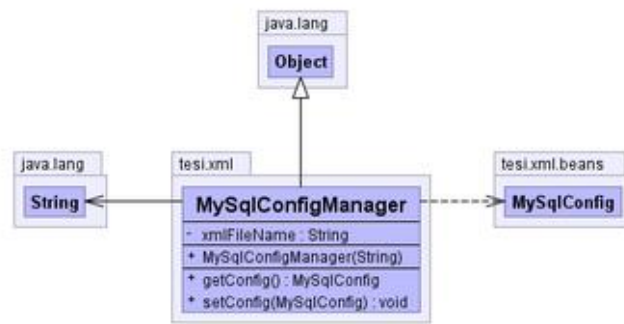


Figura III-95: Diagramma UML della classe MySQLConfigManager

### III.7.17.5. Classe SmtplibConfigManager

Classe Manager, permette la gestione delle credenziali di accesso al Server SMTP per l'invio delle email di allarme. Diagramma UML della classe:

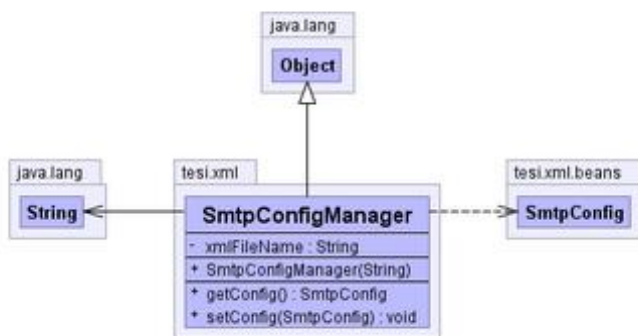


Figura III-96: Diagramma UML della classe SmtplibConfigManager

### III.7.17.6. Classe VcenterConfigManager

Classe Manager, permette la gestione delle credenziali di accesso al vSphere vCenter Server. Diagramma UML della classe:

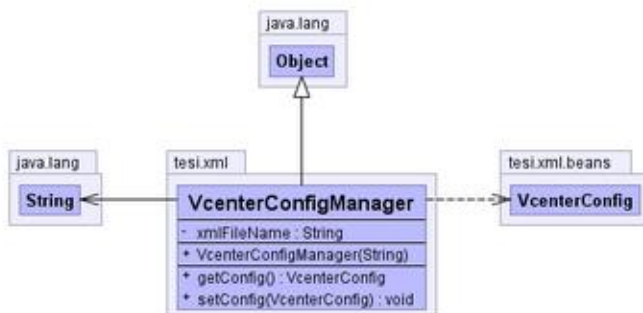


Figura III-97: Diagramma UML della classe VcenterConfigManager

### III.7.17.7. Classe WebappAccountManager

Classe Manager, permette la gestione delle credenziali di accesso alla Webapp. Diagramma UML della classe:

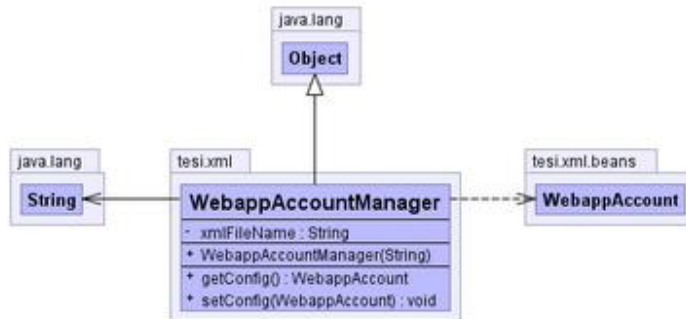


Figura III-98: Diagramma UML della classe WebappAccountManager

### III.7.17.8. Classe XmlUtil

Classe di gestione delle operazioni di lettura e scrittura del file xml di configurazione della Webapp *appconfig.xml*.

Diagramma UML della classe:



Figura III-99: Diagramma UML della classe XmlUtil

### III.7.18. Package tesi.xml.beans

Il package contiene le classi di tipo JavaBean contenenti i parametri del sistema memorizzati in maniera persistente all'interno del file di configurazione *appConfig.xml*

Le classi appartenenti al package *tesi.xml.beans* sono mostrate nella seguente figura:

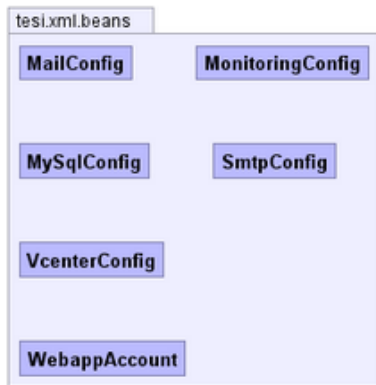


Figura III-100: Classi del package *tesi.xml.beans*

Di seguito la descrizione delle classi del package *tesi.xml.beans*:

### III.7.18.1. Classe MailConfig

Classe JavaBean, rappresenta le impostazioni relative alle email spedite dal sistema di alarming. In particolare contiene l'oggetto e l'indirizzo del destinatario della email. Diagramma UML della classe:

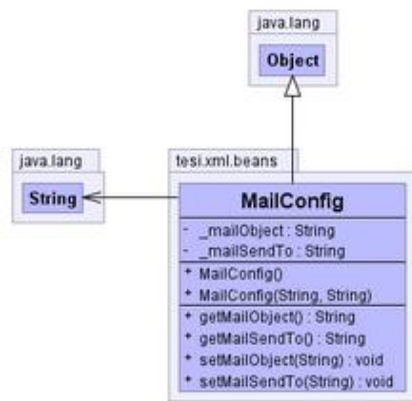


Figura III-101: Diagramma UML della classe MailConfig

### III.7.18.2. Classe MonitoringConfig

Classe JavaBean, rappresenta le impostazioni dell'applicazione memorizzate nel file `xml appCongif.xml`. In particolare contiene il parametro `monitoringDelay` che rappresenta il periodo tra due cicli di monitoring del sistema. Diagramma UML della classe:

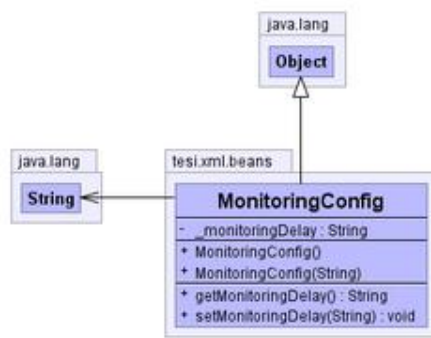


Figura III-102: Diagramma UML della classe MonitoringConfig

### III.7.18.3. Classe MySqlConfig

Classe JavaBean, rappresenta le credenziali di accesso al DBMS MySQL. Diagramma UML della classe:

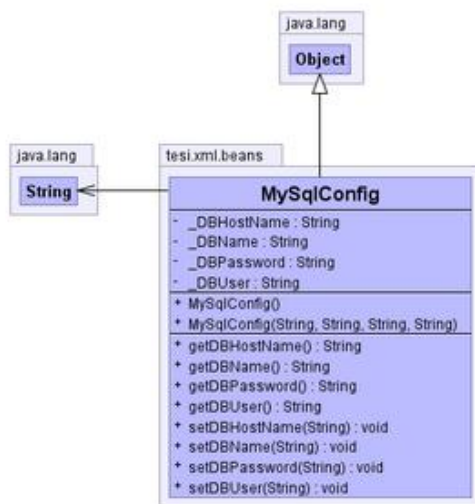


Figura III-103: Diagramma UML della classe MySqlConfig

### III.7.18.4. Classe SmtplibConfig

Classe JavaBean, rappresenta le credenziali di accesso al Server SMTP. Diagramma UML della classe:

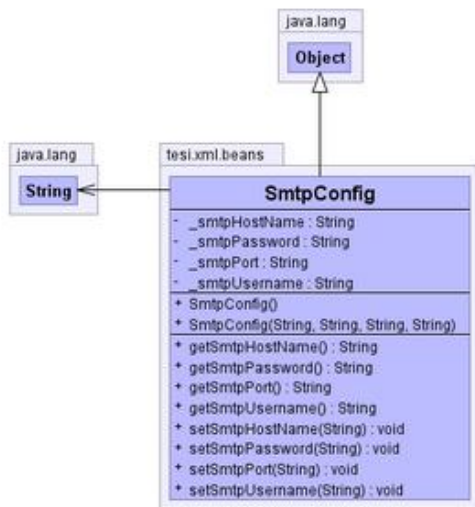


Figura III-104: Diagramma UML della classe SmtplibConfig

### III.7.18.5. Classe VcenterConfig

Classe JavaBean, rappresenta le credenziali di accesso al vCenter Server. Diagramma UML della classe:

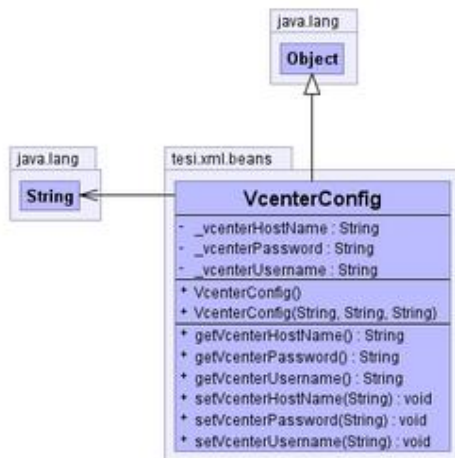


Figura III-105: Diagramma UML della classe VcenterConfig

### III.7.18.6. Classe WebappAccount

Classe JavaBean, rappresenta le credenziali di accesso alla Webapp. Diagramma UML della classe:

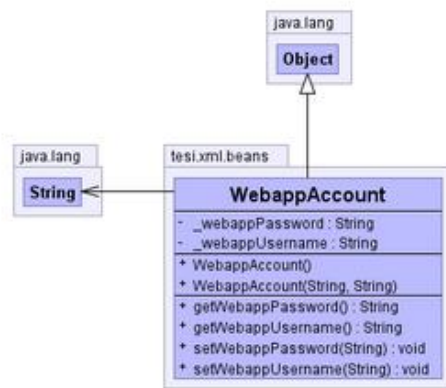


Figura III-106: Diagramma UML della classe WebappAccount

### III.8. Le pagine Jsp del sistema (View)

Come detto in precedenza, la sezione View del pattern utilizzato è costituito da una serie di pagine jsp, contenenti sia codice HTML che Java.

La struttura delle pagine della Webapp è realizzata mediante la pagina *home.jsp*. In particolare l'area di visualizzazione è suddivisa in quattro porzioni: header, footer, inventory, content.

- **header**: porzione superiore dell'area di visualizzazione. Contiene l'intestazione della Webapp.
- **footer**: porzione inferiore dell'area di visualizzazione. Contiene gli eventuali riferimenti personalizzabili della Webapp.
- **inventory**: porzione sinistra dell'area di visualizzazione. Permette la visualizzazione della struttura ad albero dell'inventario del Cloud e la barra dei comandi dell'applicazione.
- **content**: porzione destra dell'area di visualizzazione. Permette la visualizzazione delle pagine richieste dall'utente.

Mediante le pagine di popup è possibile la visualizzazione di informazioni in forma tabellare in finestre di popup esterne alla pagina principale.

L'accesso al sistema è consentito dalla presenza di una pagina di login che consente l'invio delle credenziali di accesso alla Webapp per l'autenticazione.

Nei prossimi paragrafi il dettaglio delle pagine jsp.

#### III.8.1. Pagina alarm.jsp

Pagina di visualizzazione degli allarmi inseriti con il relativo stato. E' possibile attivare/disattivare un determinato allarme e l'eliminazione dello stesso.

Alarm summary													
ID	TIMESTAMP	TYPE	NAME	TYPE2	NAME2	METRIC	CONDITION	VALUE	TIMELAST	ACTIVE	ALARMSTATE	ALARM TMSTMP	DELETE
3	2013-08-18 18:42:33.0	VirtualMachine	Win2k8 x64 VCenter	-	-	Mem Usage Average	>	0	20			2013-08-18 18:42:52	
2	2013-08-18 18:42:18.0	VirtualMachine	Win2k3 SE	Datastore	datastoreX3250	Disk Provisioned Latest	>	0	5				
1	2013-08-18 18:42:03.0	Datastore	datastoreX3400	-	-	Disk Used All Type Latest	>	0	30			2013-08-18 18:45:52	

Figura III-107: Esempio di visualizzazione della pagina alarm.jsp

### III.8.2. Pagina alarmPopup.jsp

La pagina ha le medesime funzionalità della pagina *alarm.jsp* ma relativa ad un determinato elemento del Cloud. E' visualizzata in una finestra di popup.

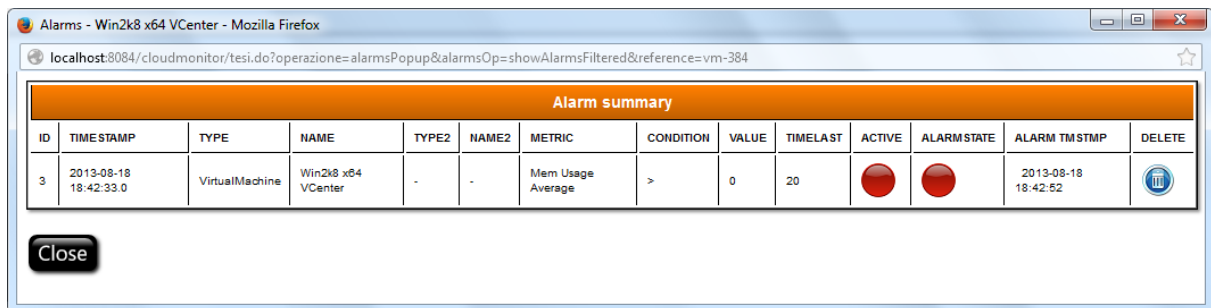


Figura III-108: Esempio di visualizzazione della pagina alarmPopup.jsp

### III.8.3. Pagina ccr.jsp

Pagina di scelta e di visualizzazione delle metriche e statistiche di performance degli elementi ClusterComputeResource e degli elementi HostSystem e VirtualMachine associati. I parametri delle metriche da visualizzare vengono inseriti attraverso un form, per mezzo del quale è anche possibile inserire un allarme relativo all'elemento selezionato. Sono anche presenti i button per la visualizzazione, in finestra di popup, dei valori tabellari delle metriche e statistiche di performance visualizzate, delle proprietà dell'elemento selezionato e degli allarmi associati.



Cluster - Economia Cluster

**Select start/end date**

From:

To:

This Cluster  
 Host System  
 Virtual Machine

Chart sorting:  Desc  Asc

Chart limit:

**Select time last**

Time range:

This Cluster  
 Host System  
 Virtual Machine

Chart sorting:  Desc  Asc

Chart limit:

**Insert alarm**

Alarm type:

Condition:  <  >

Value:

Last time:

Search parameter  
 From: 2013-09-03 12:11 To: 2013-09-04 12:11  
 Sorting type: descendant  
 Chart value limit: 20

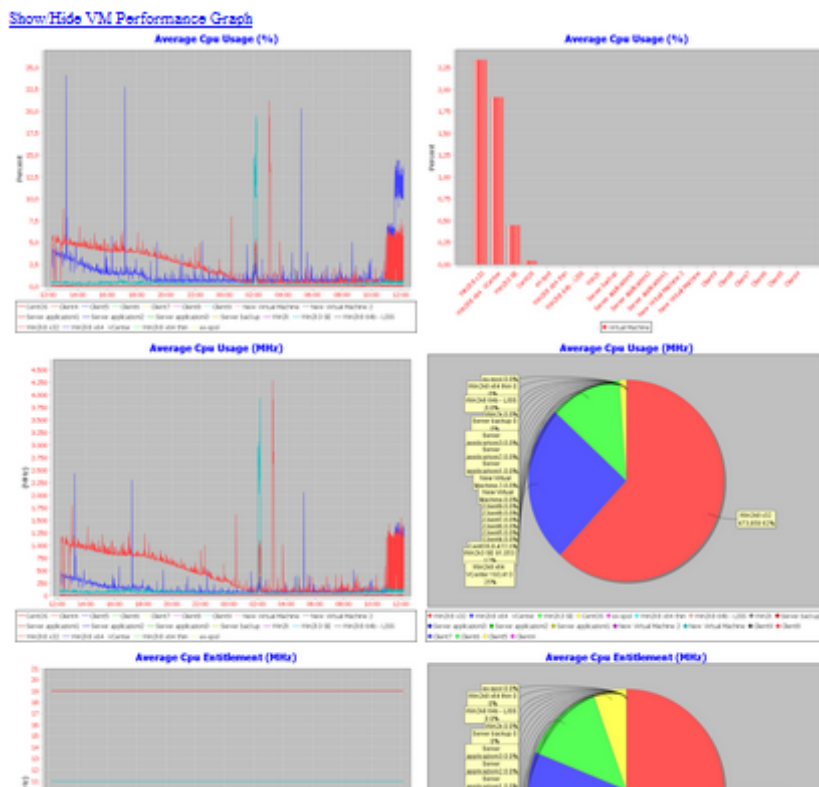
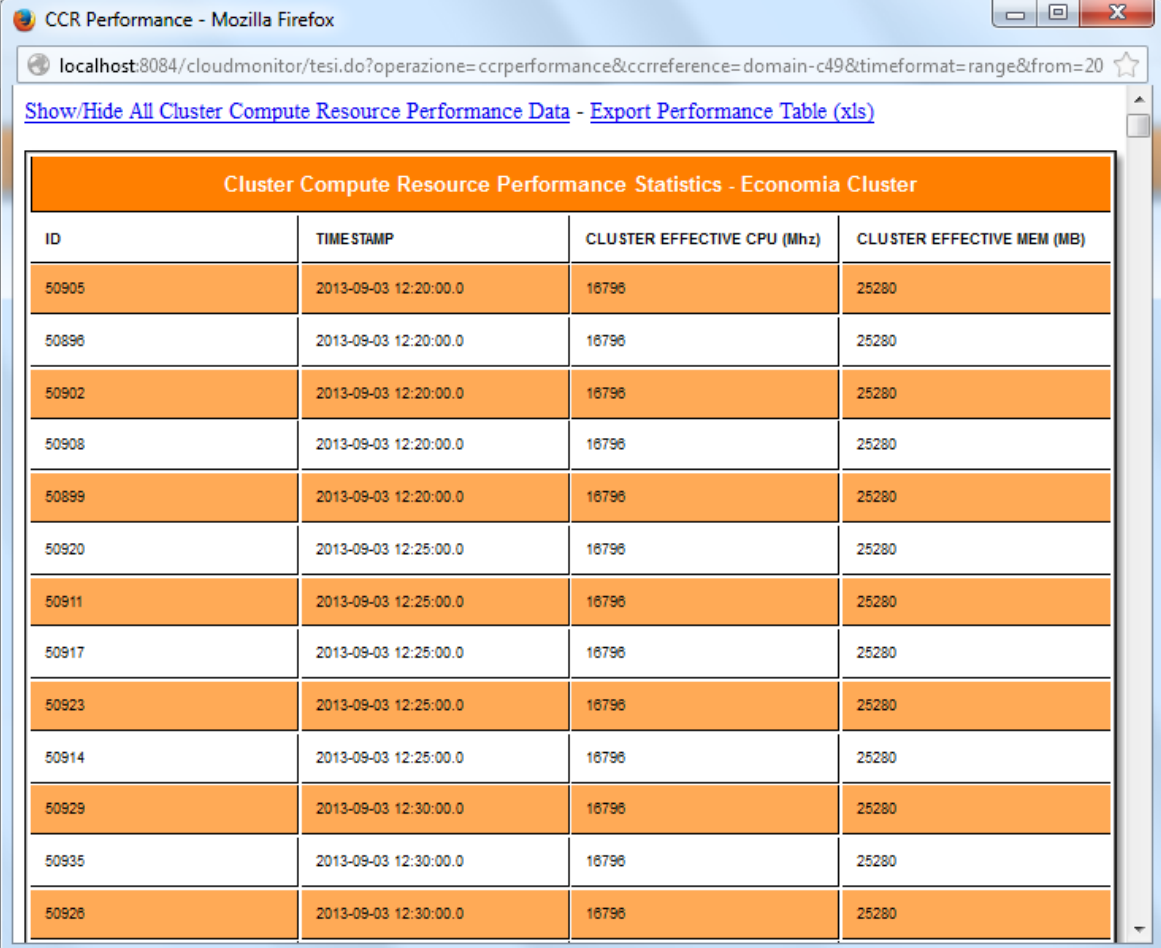


Figura III-109: Esempio di visualizzazione della pagina ccr.jsp

### III.8.4. Pagina ccrPerformance.jsp

Pagina di visualizzazione in forma tabellare delle metriche di performance visualizzate nella pagina ccr.jsp.



CCR Performance - Mozilla Firefox

localhost:8084/cloudmonitor/tesi.do?operazione=ccrperformance&ccreference=domain-c49&timeformat=range&from=20

[Show/Hide All Cluster Compute Resource Performance Data](#) - [Export Performance Table \(xls\)](#)

Cluster Compute Resource Performance Statistics - Economia Cluster			
ID	TIME STAMP	CLUSTER EFFECTIVE CPU (MHz)	CLUSTER EFFECTIVE MEM (MB)
50905	2013-09-03 12:20:00.0	16796	25280
50896	2013-09-03 12:20:00.0	16796	25280
50902	2013-09-03 12:20:00.0	16796	25280
50908	2013-09-03 12:20:00.0	16796	25280
50899	2013-09-03 12:20:00.0	16796	25280
50920	2013-09-03 12:25:00.0	16796	25280
50911	2013-09-03 12:25:00.0	16796	25280
50917	2013-09-03 12:25:00.0	16796	25280
50923	2013-09-03 12:25:00.0	16796	25280
50914	2013-09-03 12:25:00.0	16796	25280
50929	2013-09-03 12:30:00.0	16796	25280
50935	2013-09-03 12:30:00.0	16796	25280
50926	2013-09-03 12:30:00.0	16796	25280

Figura III-110: Esempio di visualizzazione della pagina ccrPerformance.jsp

### III.8.5. Pagina ccrProperties.jsp

Pagina di visualizzazione in forma tabellare delle proprietà dell'elemento selezionato nella pagina ccr.jsp.

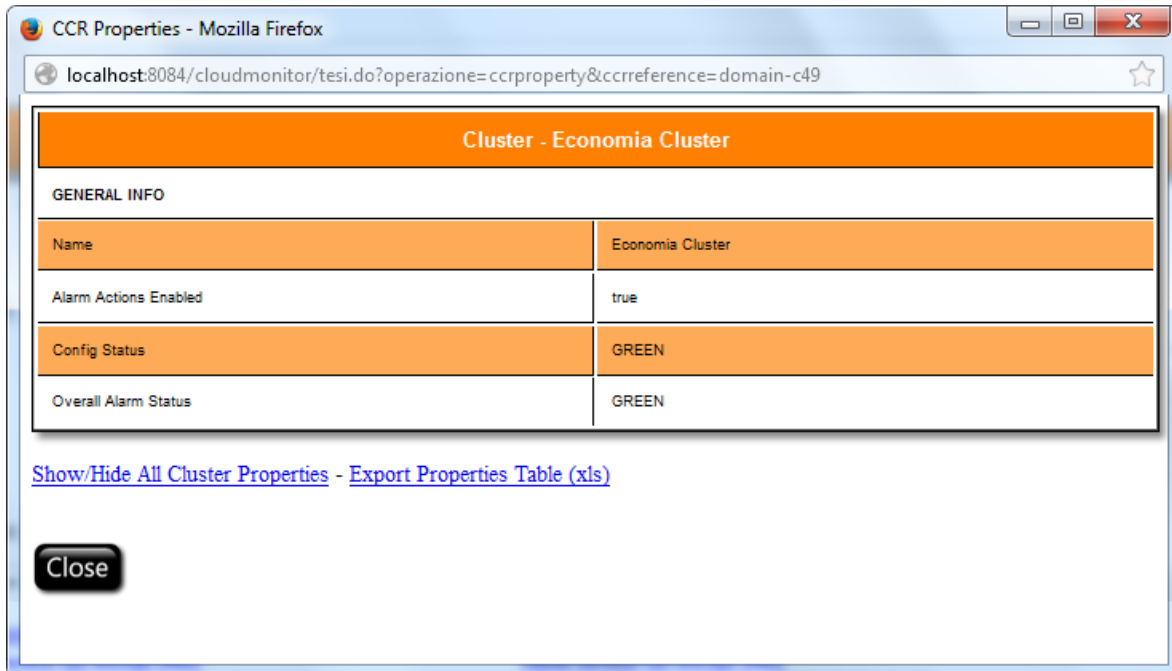


Figura III-111: Esempio di visualizzazione della pagina ccrProperties.jsp

### III.8.6. Pagina config.jsp

Pagina di Visualizzazione e di modifica del file di configurazione *appConfig.xml* dell'applicazione.

<b>VCenter configuration</b>	
vCenter Host Name	<input type="text" value="192.167.147.136"/>
vCenter Username	<input type="text" value="Firenze"/>
vCenter Password	<input type="password" value="....."/>
<b>Webapp configuration</b>	
Webapp Username	<input type="text" value="administrator"/>
Webapp Password	<input type="password" value="....."/>
<b>Data Base configuration</b>	
DB Host Name	<input type="text" value="economia.unica.it"/>
DB Username	<input type="text" value="root"/>
DB Password	<input type="password" value="...."/>
DB Name	<input type="text" value="cloudmonitor"/>
<b>SMTP Mail configuration</b>	
Smtip Host Name	<input type="text" value="smtp.gmail.com"/>
Smtip Port	<input type="text" value="465"/>
Smtip Username	<input type="text" value="vspheremonitoring@gmail.com"/>
Smtip Password	<input type="password" value="....."/>
<b>Alarm Mail configuration</b>	
Mail Object (no spaces allowed)	<input type="text" value="***VSpheere-Monitoring-Alert***"/>
Email Send To Address	<input type="text" value="giusemancio@hotmail.com"/>
<b>Monitoring configuration</b>	
Monitoring delay (second)	<input type="text" value="60"/>
<input type="button" value="Aggiorna"/>	

Figura III-112: Esempio di visualizzazione della pagina config.jsp

### III.8.7. Pagina dc.jsp

Pagina di scelta e di visualizzazione delle metriche e statistiche di performance degli elementi HostSystem, Cluster, VirtualMachine e Datastore associati al datacenter selezionato. I parametri delle metriche e statistiche di performance visualizzate vengono inseriti attraverso un form. E' presente il button per la visualizzazione, in finestra di popup, delle proprietà dell'elemento.

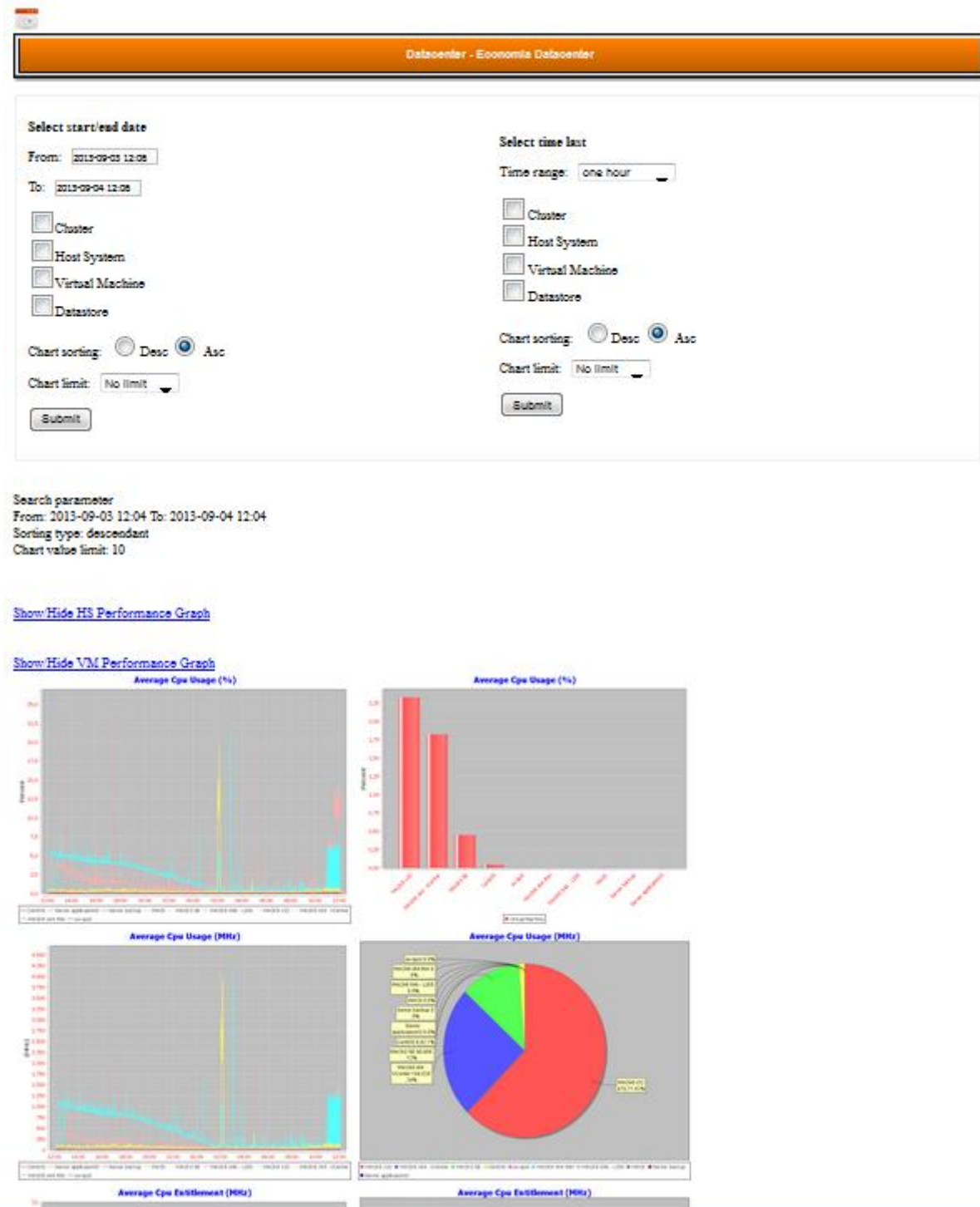


Figura III-113: Esempio di visualizzazione della pagina dc.jsp

### III.8.8. Pagina dcProperties.jsp

Pagina di visualizzazione in forma tabellare delle proprietà dell'elemento selezionato nella pagina dc.jsp.

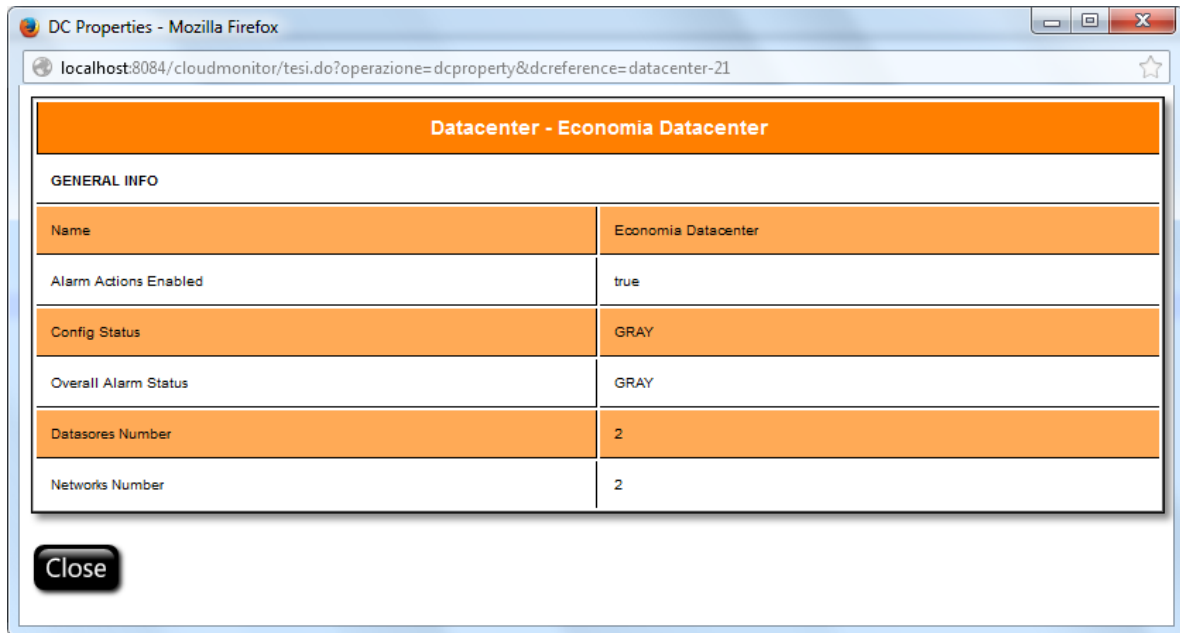


Figura III-114: Esempio di visualizzazione della pagina dcProperties.jsp

### III.8.9. Pagina ds.jsp

Pagina di scelta e di visualizzazione delle metriche e statistiche di performance degli elementi Datastore e degli elementi VirtualMachine associati. I parametri delle metriche e statistiche di performance vengono inseriti attraverso un form, per mezzo del quale è anche possibile inserire un allarme relativo all'elemento selezionato. Sono anche presenti i button per la visualizzazione, in finestra di popup, dei valori tabellari delle metriche visualizzate, delle proprietà dell'elemento selezionato e degli allarmi associati.

Host System - datastoreX3260

---

**Select start/end date**

From:

To:

This Datastore  
 Virtual Machine

Chart sorting:  Desc  Asc

Chart limit:

**Select time last**

Time range:

This Datastore  
 Virtual Machine

Chart sorting:  Desc  Asc

Chart limit:

**Insert alarm**

Alarm type:

Condition:  <  >

Value:

Last time:

Search parameter  
 From: 2013-09-03 12:15 To: 2013-09-04 12:15  
 Sorting type: descendant  
 Chart value limit: 20

[Show/Hide DS Performance Graph](#)

[Show/Hide DS Storage By VM Stats Graph](#)

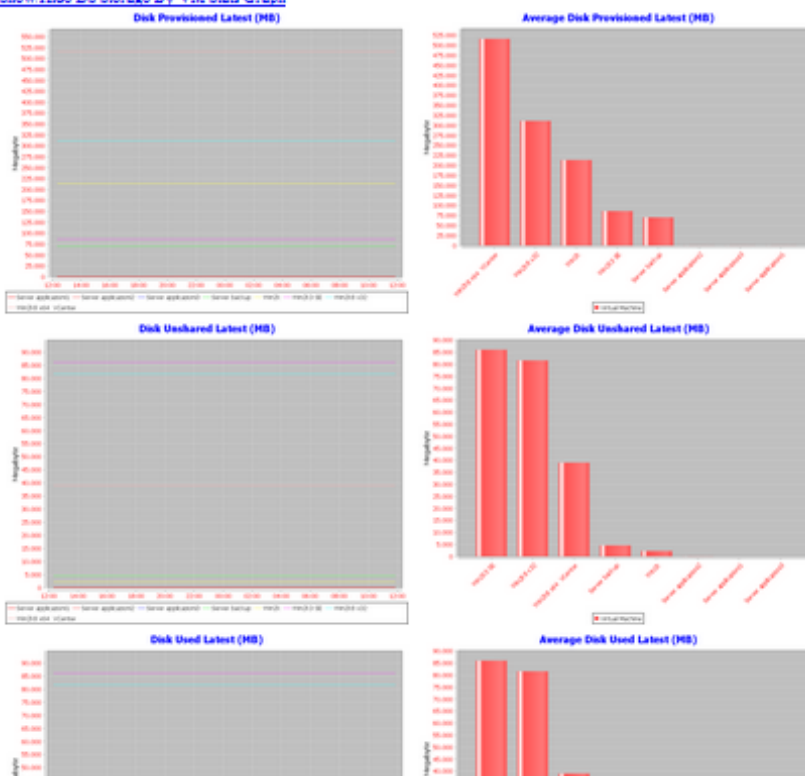


Figura III-115: Esempio di visualizzazione della pagina ds.jsp

### III.8.10. Pagina dsPerformance.jsp

Pagina di visualizzazione in forma tabellare delle metriche di performance visualizzate nella pagina ds.jsp.

ID	TIMESTAMP	DATASTORE READ AVERAGE (KBps)	DATASTORE WRITE AVERAGE (KBps)	DATASTORE NUMBER WRITE AVERAGE (number)	DATASTORE NUMBER READ AVERAGE (number)	DATASTORE BUS RESETS SUMMATION (number)	DATASTORE COMMANDS ABORTED SUMMATION (number)	DATASTORE THROUGHPUT USAGE AVERAGE (KBps)	DATASTORE THROUGHPUT CONTENTION AVERAGE (millisecond)	DISK CAPACITY LATEST (MB)	DISK PROVISIONED LATEST (MB)	DISK USED LATEST (MB)	DISK USED (DISK FILE) (MB)	DISK USED (DELTA FILE) (MB)	DISK USED (SWAP FILE) (MB)	DISK USED (OTHER FILE) (MB)
34009	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34047	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34031	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
33991	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34013	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34035	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
33995	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34017	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34039	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
33999	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	
34021	2013-09-03 12:50:00.0	0	0	0	0	0	0	0	0	471808.0	1199352.447255029811.0	199833.0	288.15025	13312.0	284.099609375	

Figura III-116: Esempio di visualizzazione della pagina dsPerformance.jsp

#### III.8.11. Pagina dsProperties.jsp

Pagina di visualizzazione in forma tabellare delle proprietà dell'elemento selezionato nella pagina ds.jsp.



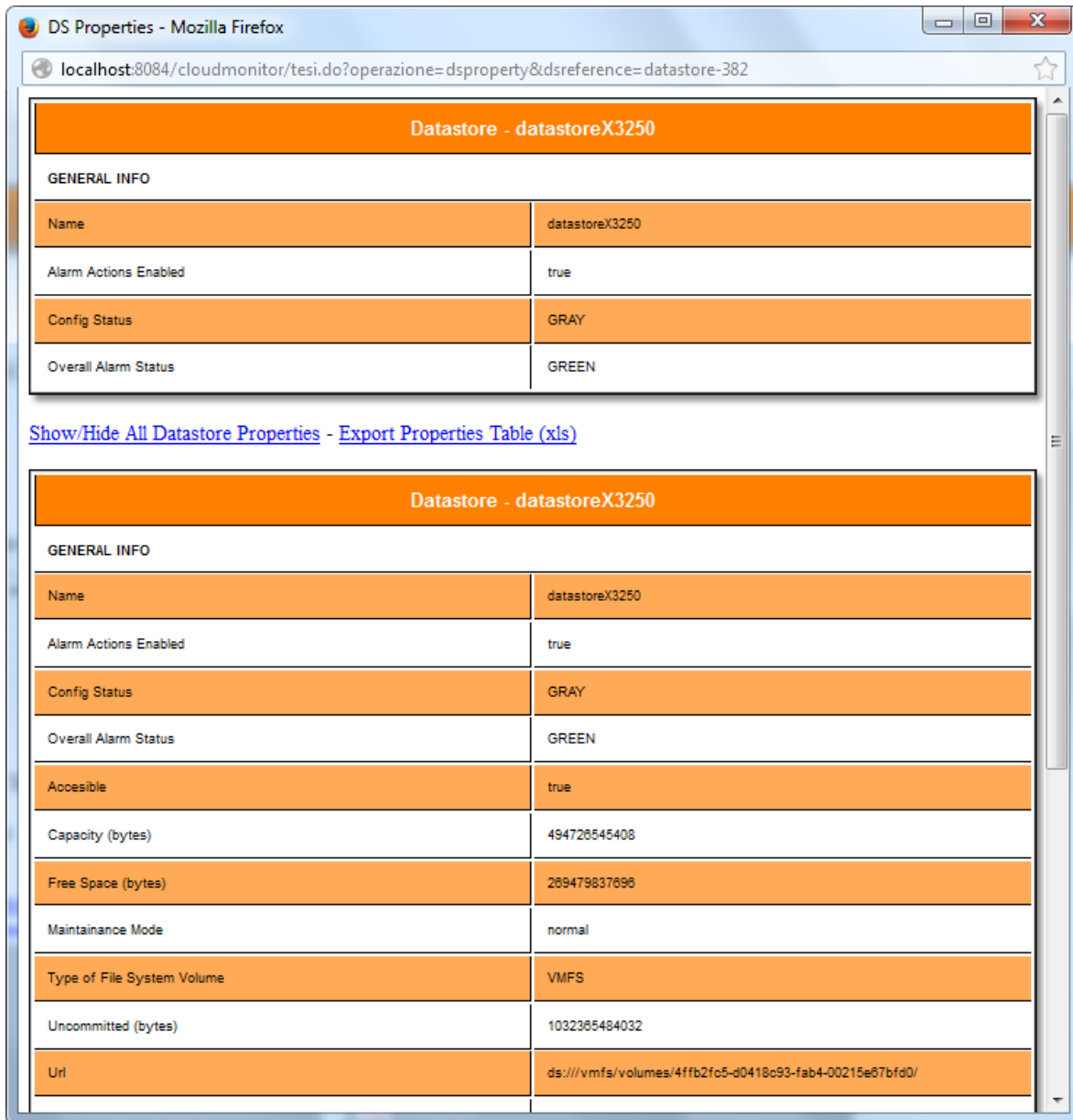


Figura III-117: Esempio di visualizzazione della pagina dsProperties.jsp

### III.8.12. Pagina footer.jsp

Pagina di visualizzazione della sezione footer all'interno di tutte le pagine integrate nella pagina home.jsp.



Figura III-118: Esempio di visualizzazione della pagina footer.jsp

### III.8.13. Pagina generalInfo.jsp

Pagina di visualizzazione delle informazioni relative al server vCenter vSphere del Cloud monitorato. In particolare visualizza la versione del vCenter Server ed il sistema operativo su cui è installato.

VCenter/Host System	
GENERAL INFO	
Full Name	VMware vCenter Server 5.0.0 build-623373
O.S. Type	win32-x84

Figura III-119: Esempio di visualizzazione della pagina generalInfo.jsp

### III.8.14. Pagina header.jsp

Pagina di visualizzazione della sezione header all'interno di tutte le pagine integrate nella pagina home.jsp.

#### VMware vSphere Monitoring Tool

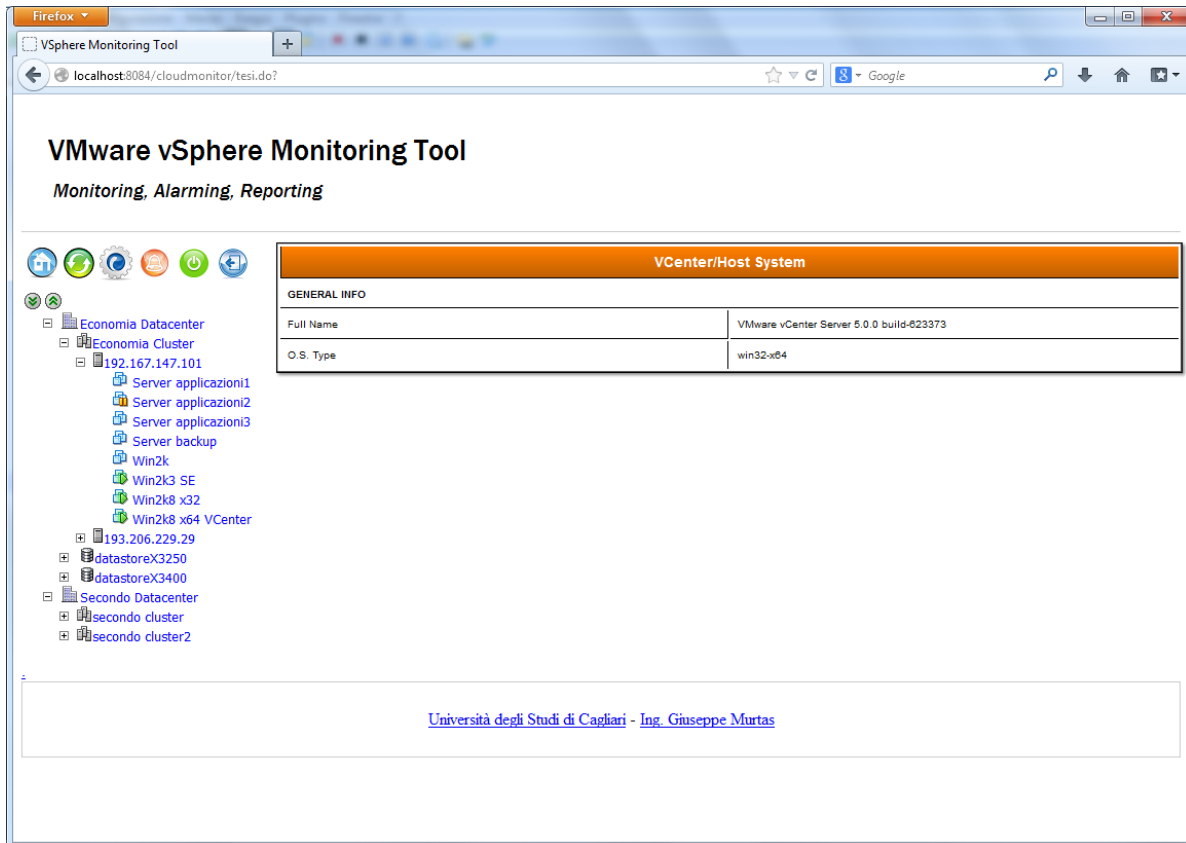
*Monitoring, Alarming, Reporting*

Figura III-120: Esempio di visualizzazione della pagina header.jsp

### III.8.15. Pagina home.jsp

La pagina consente la divisione dell'area di visualizzazione del browser in quattro aree:

- footer: visualizza il contenuto della pagina footer.jsp nell'area inferiore.
- header: visualizza il contenuto della pagina header.jsp nell'area superiore.
- inventario: visualizza il contenuto della pagina inventarioTree.jsp nell'area sinistra.
- content: visualizza le altre pagine jsp nell'area destra.



**Figura III-121: Esempio di visualizzazione della pagina home.jsp**

### III.8.16. Pagina hs.jsp

Pagina di scelta e di visualizzazione delle metriche e statistiche di performance degli elementi HostSystem e degli elementi VirtualMachine associati. I parametri delle metriche e statistiche di performance vengono inseriti attraverso un form, per mezzo del quale è anche possibile inserire un allarme relativo all'elemento dell'inventario selezionato. Sono anche presenti i button per la visualizzazione in finestra di popup dei valori tabellari delle metriche visualizzate, delle proprietà dell'elemento selezionato e degli allarmi associati.

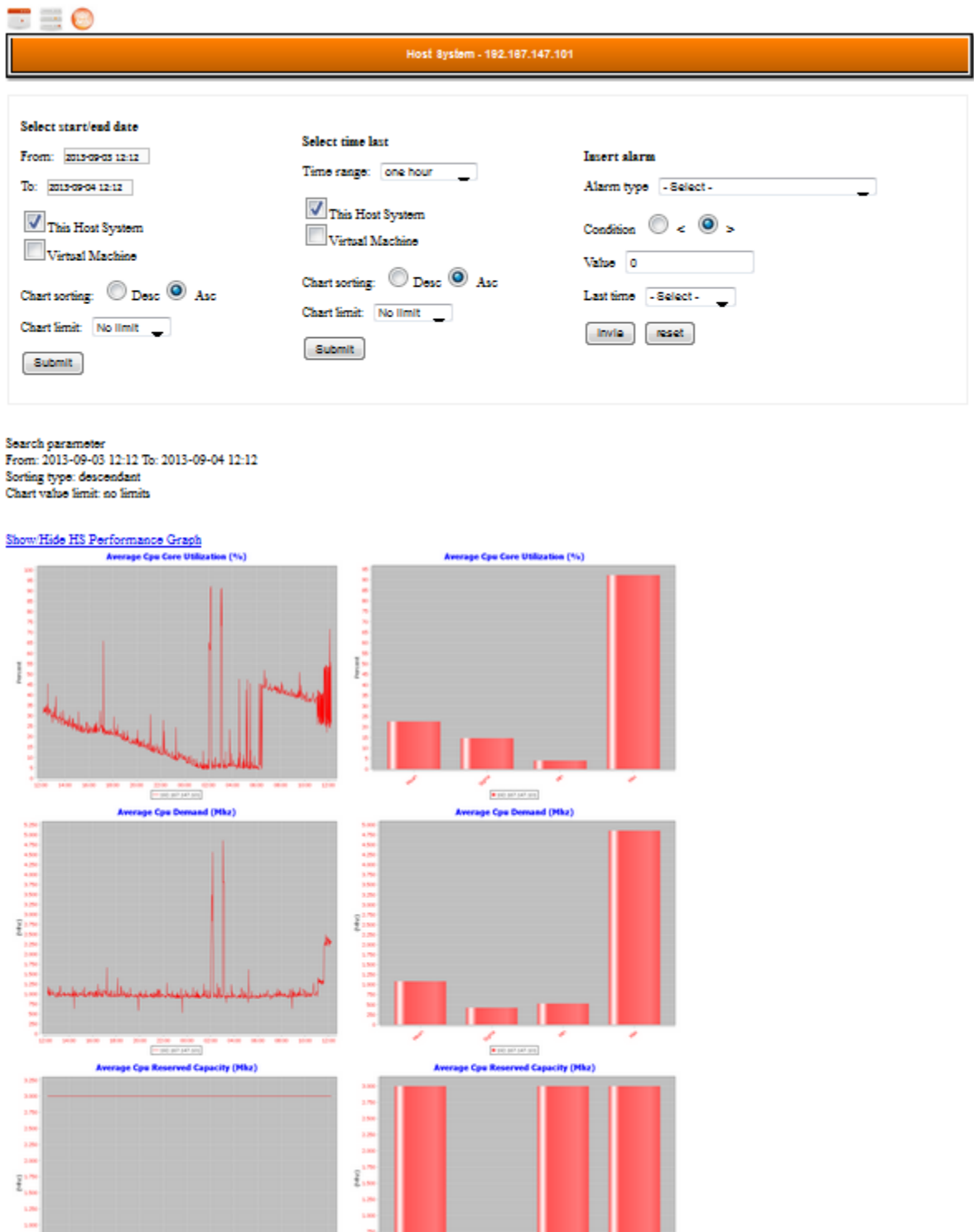


Figura III-122: Esempio di visualizzazione della pagina hs.jsp

### III.8.17. Pagina hsPerformance.jsp

Pagina di visualizzazione in forma tabellare delle metriche di performance visualizzate nella pagina hs.jsp.

ID	TIME	CPU CORE UTILIZATION (%)	CPU DEMAND AVERAGE (MHz)	CPU RESERVE CAPACITY AVERAGE (MHz)	CPU TOTAL CAPACITY AVERAGE (MHz)	CPU USAGE AVERAGE (%)	CPU USAGE MHZ AVERAGE (MHz)	CPU UTILIZATION AVERAGE (%)	MEM UTILIZATION AVERAGE (%)	MEM CONSUMPTION AVERAGE (MB)	MEM OVERHEATED AVERAGE (MB)	MEM RESERVE CAPACITY AVERAGE (MB)	MEM SHARED AVERAGE (KB)	MEM COMMON AVERAGE (KB)	MEM STATE (0:high - 1:soft - 2:hard - 3:low)	MEM SWAP IN RATE AVERAGE (KBps)	MEM SWAP OUT RATE AVERAGE (KBps)	MEM SYS USAGE AVERAGE (MB)	MEM TOTAL CAPACITY AVERAGE (MB)	MEM UNRESERVED AVERAGE (MB)	MEM RESERVE AVERAGE (%)	DISK USAGE AVERAGE (KBps)	DISK READ AVERAGE (KBps)	DISK WRITE AVERAGE (KBps)	POWER CAPACITY AVERAGE (W)	POWER USAGE AVERAGE (W)
32975	2013-09-03 12:29:40	30.8	933	3000	8510	15.14	1531	8.2	1945.05	69878001.175582512.733282059370	265	627272	104	0	0	0	952.74033785590	15325.281290.9	15325.281290.9	86	23	72	0	0		
32977	2013-09-03 12:30:40	30.35	991	3000	8510	14.92	1508	8.08	1904.1853182001.050782252.838645000670	265	627272	104	0	0	0	0	952.74216785590	15325.281290.9	15325.281290.9	81	19	81	0	0		
32979	2013-09-03 12:31:40	36.4	1143	3000	8510	17.83	1812	9.68	2313.7365715000.839532512.839645000670	265	627272	104	0	0	0	0	952.74216785590	15325.281290.9	15325.281290.9	500	13	493	0	0		
32981	2013-09-03 12:32:40	35.02	1112	3000	8510	17.52	1770	9.48	2082.9646783001.75980083.38874630740093285	265	627272	104	0	0	0	0	953.03802250589	15324.864370.9	15324.864370.9	171	12	158	0	0		
32983	2013-09-03 12:33:40	29.98	1069	3000	8510	14.74	1489	7.97	2000.4521878001.148432512.8194800004102805	265	627272	104	0	0	0	0	952.74216785590	15325.281290.9	15325.281290.9	128	12	116	0	0		
32985	2013-09-03 12:34:40	45.32	970	3000	8510	22.4	2264	12.0	2437.125	13001.84453283.2808085.048437865	265	627272	104	0	0	0	0	953.097660385589	15324.525790.9	15324.525790.9	85	21	73	0	0	
32987	2013-09-03 12:35:40	33.72	995	3000	8510	16.55	1673	8.86	2272.7880828001.03282512.75208020640025265	265	627272	104	0	0	0	0	952.80081825590	15325.228980.9	15325.228980.9	87	17	70	0	0		
32989	2013-09-03 12:36:40	31.11	974	3000	8510	15.29	1545	8.29	2108.878003800.91150282.722080206231285	265	627272	104	0	0	0	0	952.80081825590	15325.228980.9	15325.228980.9	86	20	64	0	0		

Figura III-123: Esempio di visualizzazione della pagina hsPerformance.jsp

### III.8.18. Pagina hsProperties.jsp

Pagina di visualizzazione in forma tabellare delle proprietà dell'elemento selezionato nella pagina hs.jsp.

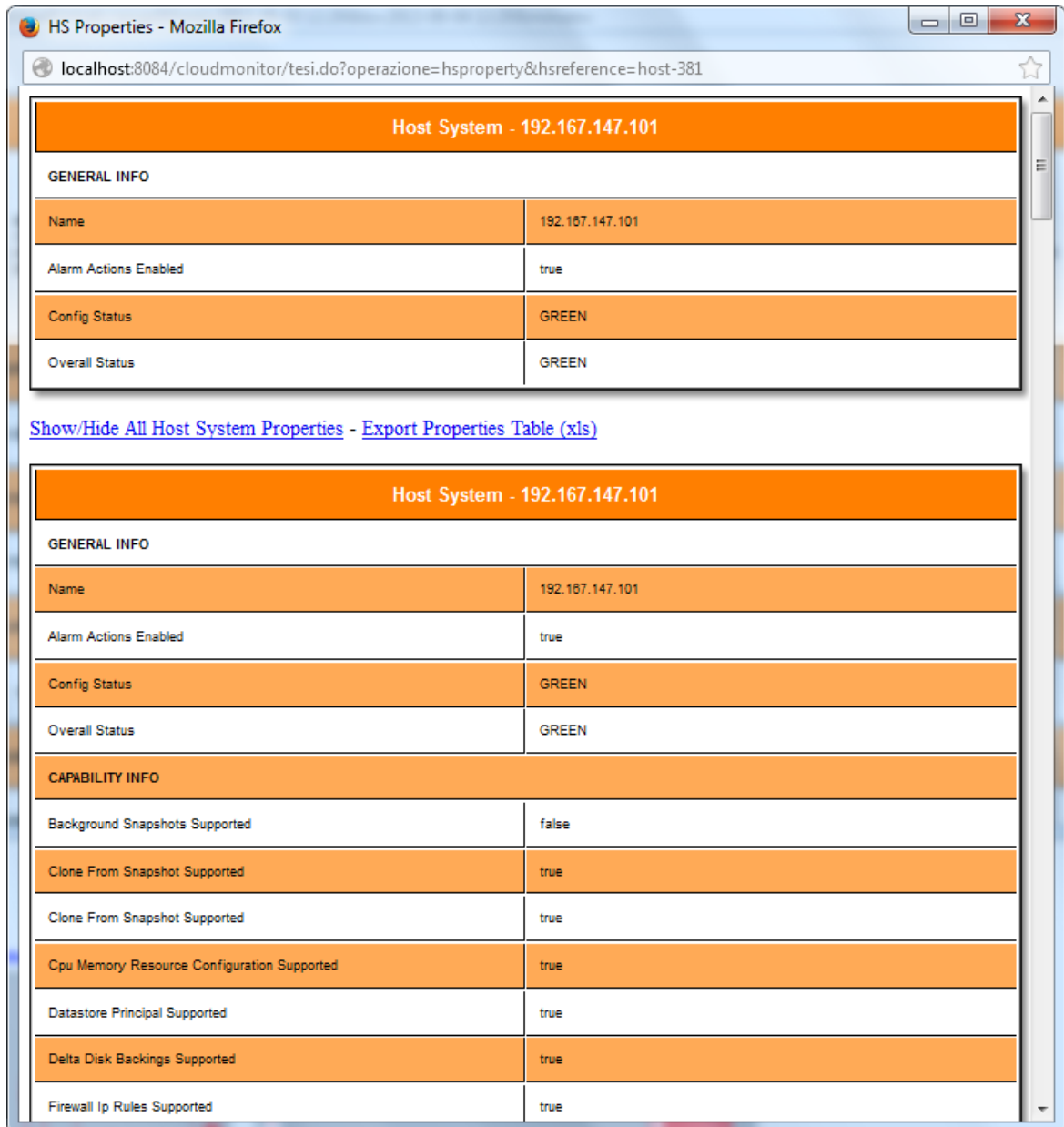


Figura III-124: Esempio di visualizzazione della pagina hsProperties.jsp

### III.8.19. Pagina inventarioTree.jsp

Pagina di visualizzazione dell'inventario del sistema Cloud monitorato e dei controlli dell'applicazione.



Figura III-125: Visualizzazione della sezione comandi della pagina inventarioTree.jsp

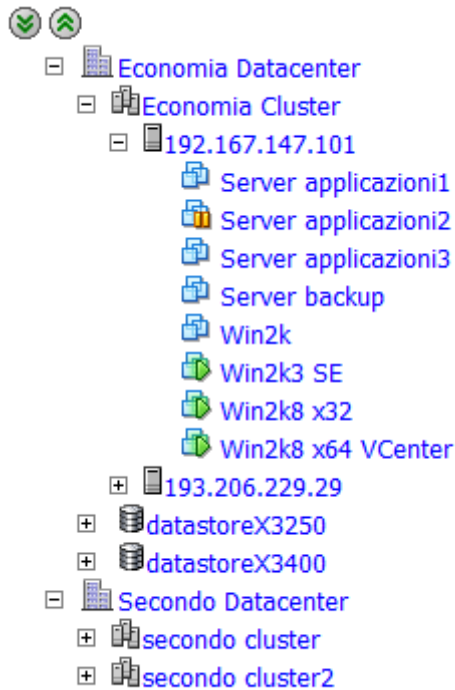


Figura III-126: Esempio di visualizzazione della pagina inventarioTree.jsp

### III.8.20. Pagina login.jsp

Pagina di login al sistema.

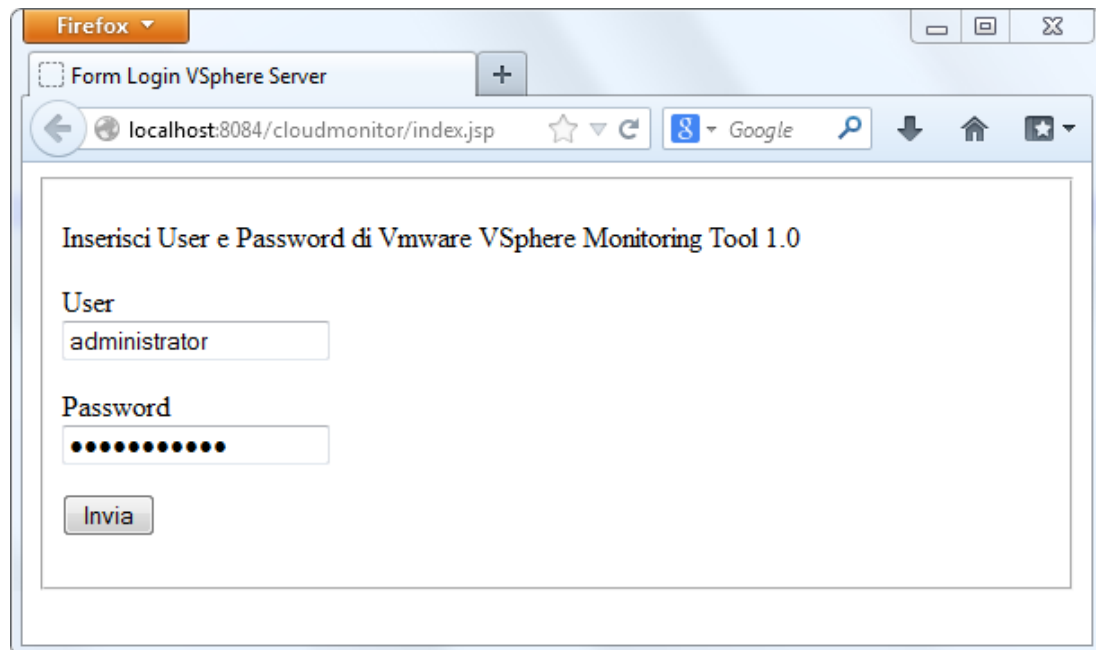


Figura III-127: Esempio di visualizzazione della pagina login.jsp

### III.8.21. Pagina loginError.jsp

Pagina di visualizzazione dell'avviso di login errato.

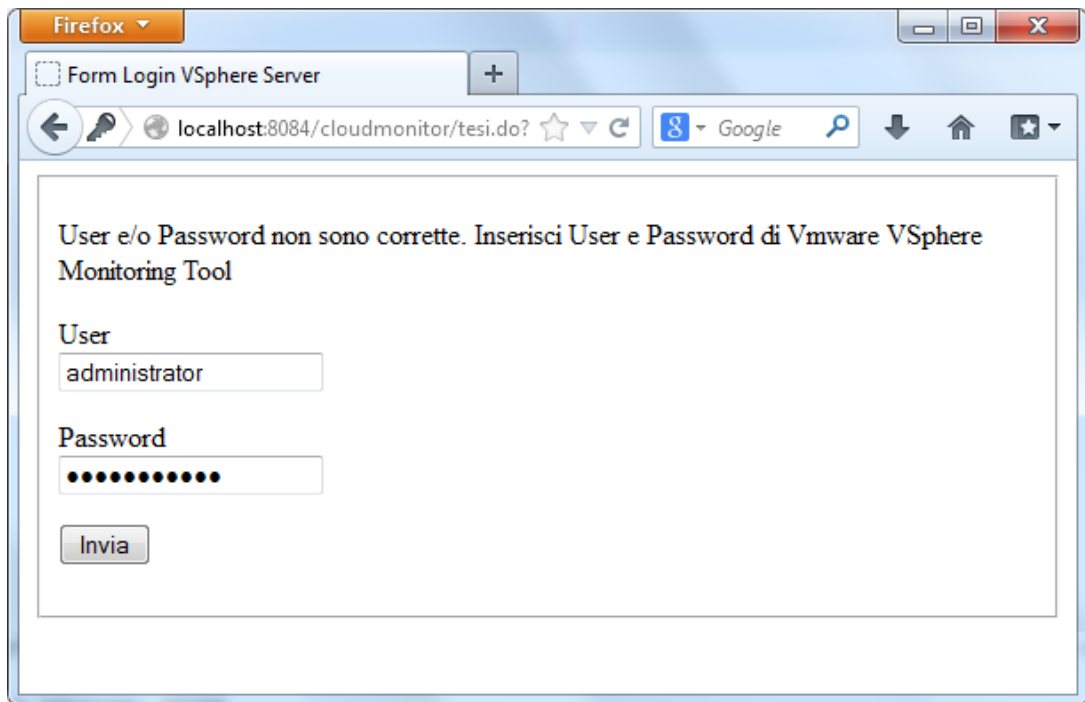


Figura III-128: Esempio di visualizzazione della pagina loginError.jsp

### III.8.22. Pagina message.jsp

Pagina di visualizzazione dei messaggi di conferma e di errore dell'applicazione.

Monitoring disattivato

Figura III-129: Esempio di visualizzazione della pagina message.jsp

### III.8.23. Pagina vm.jsp

Pagina di scelta e di visualizzazione delle metriche e statistiche di performance degli elementi VirtualMachine. I parametri delle metriche e statistiche di performance vengono inseriti attraverso un form, per mezzo del quale è anche possibile inserire un allarme relativo all'elemento dell'inventario selezionato. Sono anche presenti i button per la visualizzazione, in finestra di popup, dei valori tabellari delle metriche visualizzate, delle proprietà dell'elemento selezionato e degli allarmi associati.



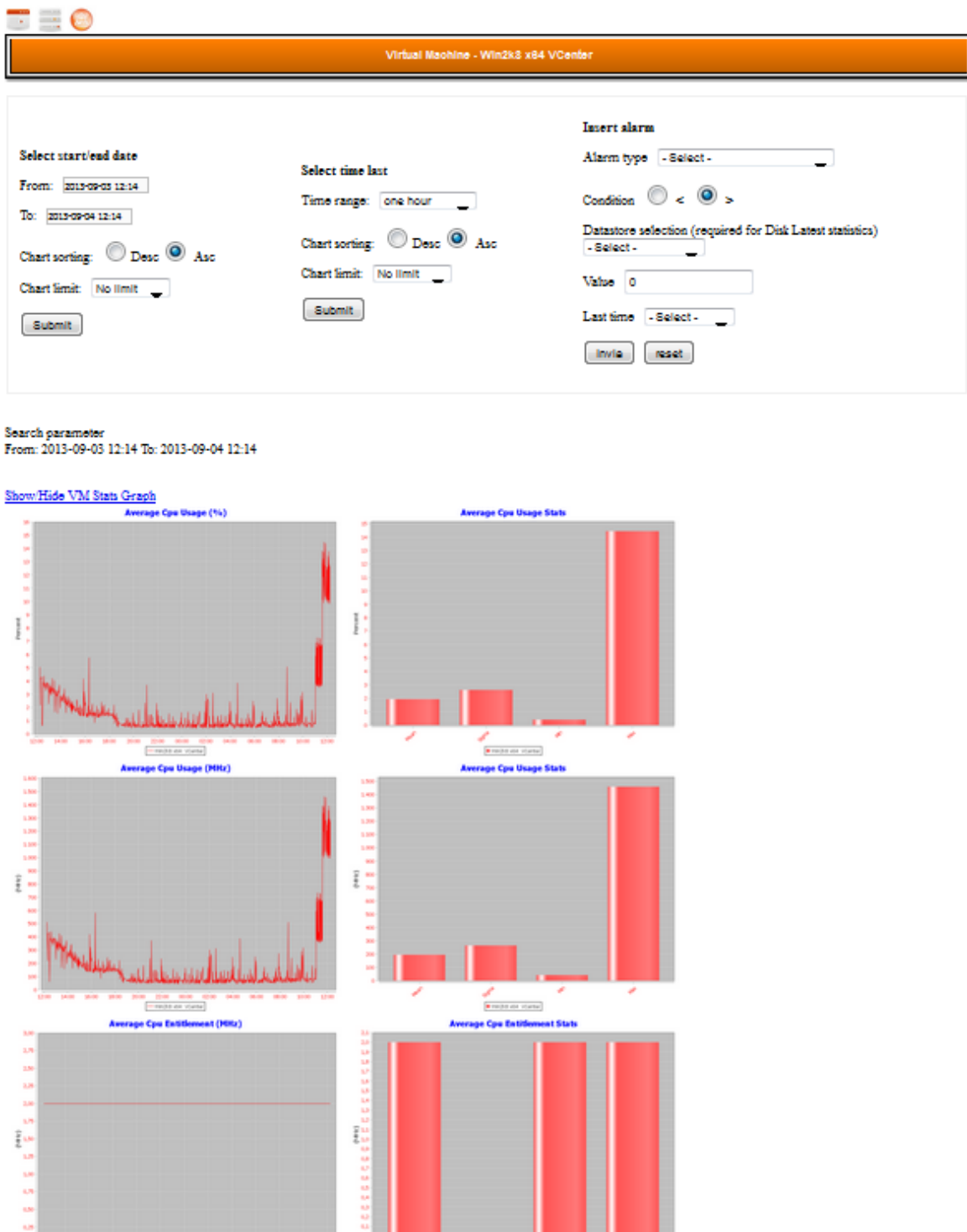


Figura III-130: Esempio di visualizzazione della pagina vm.jsp

### III.8.24. Pagina vmPerformance.jsp

Pagina di visualizzazione in forma tabellare delle metriche di performance visualizzate nella pagina vm.jsp.

ID	TIMESTAMP	CPU USAGE AVERAGE (%)	CPU USAGE MHZ AVERAGE (MHZ)	CPU ENTITLEMENT AVERAGE (MHZ)	MEM ACTIVE AVERAGE (MB)	MEM CONSUMED AVERAGE (MB)	MEM ENTITLEMENT AVERAGE (MB)	MEM GRANTED AVERAGE (MB)	MEM OVERHEAD AVERAGE (MB)	MEM SHARED AVERAGE (KB)	MEM USAGE AVERAGE (%)	DISK USAGE AVERAGE (KBps)	DISK READ AVERAGE (KBps)	DISK WRITE AVERAGE (KBps)	POWER AVERAGE (W)
88703	2013-09-03 12:35:40.0	5.55	1123	5	1310.71875	4096.0	4178.74809875	4096.0	77.51953125	0	31.99	22	0	22	0
88707	2013-09-03 12:35:40.0	4.90	1003	5	1187.83864375	4096.0	4178.825	4096.0	77.3984375	0	28.99	24	2	21	0
88711	2013-09-03 12:37:40.0	5.91	1196	5	1024.0	4096.0	4178.48049875	4096.0	77.25390625	0	25.0	52	0	52	0
88715	2013-09-03 12:39:40.0	5.32	1078	5	1084.95703125	4096.0	4178.4375	4096.0	77.2109375	0	25.99	35	0	35	0
88718	2013-09-03 12:40:40.0	5.19	1050	5	942.078125	4096.0	4178.35549875	4096.0	77.12890625	0	22.99	22	0	21	0
88721	2013-09-03 12:41:40.0	5.08	1028	5	1289.7578125	4096.0	4178.41015625	4096.0	77.18359375	0	30.99	22	0	22	0
88725	2013-09-03 12:42:40.0	5.58	1124	5	1084.95703125	4096.0	4178.44140625	4096.0	77.21484375	0	25.99	84	0	84	0
88729	2013-09-03 12:43:40.0	5.11	1034	5	1084.95703125	4096.0	4178.43359375	4096.0	77.20703125	0	25.99	21	0	21	0
88733	2013-09-03 12:44:40.0	5.33	1078	5	1105.91796875	4096.0	4178.5330625	4096.0	77.3125	0	25.99	24	0	24	0
88737	2013-09-03 12:44:40.0	5.13	1037	5	1145.87890625	4096.0	4178.5545875	4096.0	77.328125	0	27.99	21	0	21	0
88741	2013-09-03 12:46:40.0	5.04	1019	5	1084.95703125	4096.0	4178.359375	4096.0	77.1328125	0	25.99	35	11	24	0
88745	2013-09-03 12:48:40.0	5.14	1039	5	1310.71875	4096.0	4178.48049875	4096.0	77.25590625	0	31.99	22	0	22	0

Figura III-131: Esempio di visualizzazione della pagina vmPerformance.jsp

### III.8.25. Pagina vmProperties.jsp

Pagina di visualizzazione in forma tabellare delle proprietà dell'elemento selezionato nella pagina vm.jsp.

VM Properties - Mozilla Firefox  
 localhost:8084/cloudmonitor/tesi.do?operazione=vmproperty&vmreference=vm-392

**Virtual Machine - Win2k8 x32**

GENERAL INFO

Name	Win2k8 x32
Alarm Actions Enabled	true
Config Status	GREEN
Overall Alarm Status	GREEN
Guest Heartbeat Status	GREEN

[Show/Hide All Virtual Machine Properties](#) - [Export Properties Table \(xls\)](#)

**Virtual Machine Properties - Win2k8 x32**

GENERAL INFO

Name	Win2k8 x32
Alarm Actions Enabled	true
Config Status	GREEN
Overall Alarm Status	GREEN
Guest Heartbeat Status	GREEN

GENERAL CONFIG INFO

Alternate Guest Name	
Annotation	
Change Version	2013-09-02T07:36:03.902121Z
Firmware	bios
Guest Full Name	Microsoft Windows Server 2008 (32-bit)

Figura III-132: Esempio di visualizzazione della pagina vmProperties.jsp

## IV. Installazione e configurazione

### IV.1. Introduzione

In questo capitolo verrà illustrata la procedura di installazione e configurazione del sistema di monitoring, alarming, reporting.

### IV.2. Installazione e configurazione dell'Application Server Tomcat 7

Apache Tomcat 7 [TASF13a] è un prodotto gratuito e open source sviluppato in Java all'interno della fondazione Apache.

Tomcat realizza le seguenti funzioni:

- Web server.
- Container per Webapp conforme alle specifiche Java 2 Enterprise Edition (J2EE). Il che significa che Tomcat funziona da motore per Servlet e Java Server Pages (JSP).

Prima di installare l'application server è necessario avere installato il pacchetto JDK (Java Development Kit). JDK viene fornito insieme alla JVM (Java Virtual Machine) e alle varie librerie di java (API Java), contenendo dunque anche il JRE (Java Runtime Environment).

Ecco i passi da seguire per installare e configurare correttamente l'ambiente Tomcat 7 [TASF13b] in ambiente Windows:

1. Il primo passo è scaricare la versione ufficiale dell'application server dal sito <http://tomcat.apache.org>. Sono disponibili sia le versioni a 32 bit e a 64 bit. Scegliere l'ultima versione disponibile zip (apache-tomcat-7.0.42-windows-x86.zip) ed estrarre il file nella cartella di installazione scelta, per esempio c:\Programmi\apache\tomcat7.
2. E' ora necessario configurare alcune proprietà per consentire a Tomcat di conoscere dove JDK è installato. Creare un file di testo nominato *setenv.bat* con il seguente contenuto:

```
set "JRE_HOME=%ProgramFiles(x86)%\Java\jre6"  
exit /b 0
```

La stringa %ProgramFiles(x86)%\Java\jre6 rappresenta il percorso in cui è stato installato il JRE. Salvare quindi il file nella seguente cartella: c:\Programmi\apache\tomcat7\bin in modo che sia eseguito all'avvio di Tomcat.

3. E' necessario impostare Tomcat come servizio di Windows eseguendo, con diritti da amministratore, le seguenti istruzioni da prompt dei comandi:

```
cd C:\apps\apache-tomcat-7.0.28\bin
```

Eeguire quindi `setenv.bat` creato in precedenza ed eseguire lo script `service.bat` secondo la seguente sintassi:

```
service.bat install/remove [service_name] [/user username]
```

nel nostro caso avremo:

```
service.bat install Tomcat7
```

4. La JVM gestisce due aree di memoria denominate rispettivamente Heap e Permanent Generation [TASF13a].

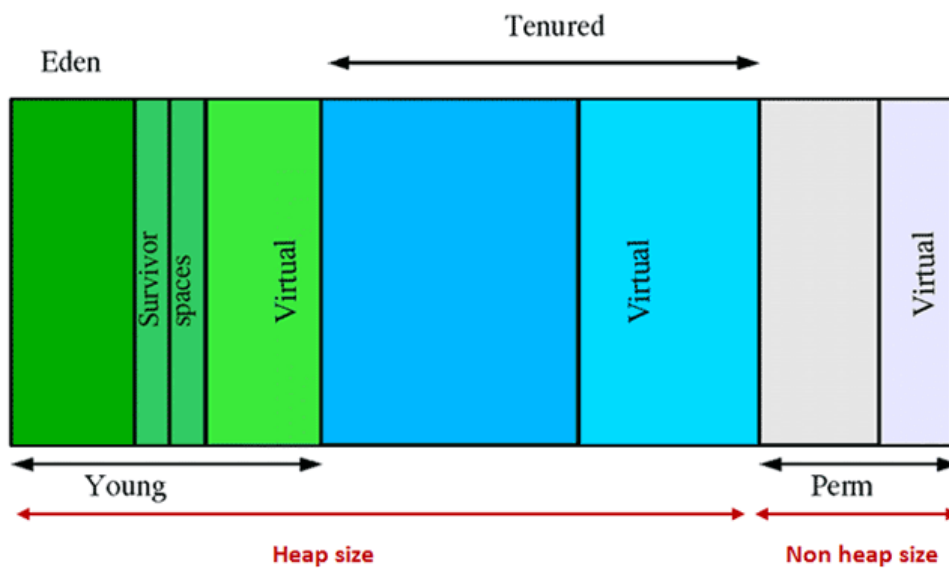


Figura IV-1: Tipologie di memoria di una JVM

Permanent Generation (PermGen) è usata dalla JVM per memorizzare i metadati relativi agli oggetti creati. La dimensione della PermGen può essere variata attraverso i parametri:

**-XX:PermSize**

**-XX:MaxPermSize**

che permettono di inizializzare la dimensione minima e massima. Per evitare il costo, in termini di tempo di esecuzione, del ridimensionamento della memoria allocata è possibile configurare lo stesso valore per la dimensione minima e massima.

La memoria Heap rappresenta l'area di memoria principale dove vengono memorizzati gli oggetti creati dall'utente. A sua volta è suddivisa in Old Generation (OG) e Young Generation (YG). Quest'ultima è ulteriormente suddivisa in Eden e Survivor. La memoria Heap è controllata dai parametri

**-Xms**

**-Xmx**

rispettivamente per il minimo e per il massimo. Esistono anche altri parametri per controllare ciascuna delle aree che compongono la memoria Heap. Tra questi, il parametro

### **-XX:NewRatio=value**

definisce la dimensione dello YG. Quando si crea un oggetto, attraverso l'operatore *new*, questo viene inserito in Eden. Se l'area è saturata, la JVM esegue il Garbage Collector (GC), in modo da individuare e rimuovere eventuali oggetti ancora presenti in memoria ma non più necessari. Gli oggetti che risultano essere ancora validi alle prime raccolte di oggetti inutili vengono spostati in Survivor. La dimensione del Survivor viene specificata attraverso il parametro

### **-XX:SurvivorRatio=value.**

Le applicazioni con un'alta percentuale di oggetti creati ed una bassa percentuale di conservazione dovrebbero aumentare questo parametro. Gli oggetti vecchi presenti in Survivor ma ancora validi, vengono infine spostati in OG.

Per ottimizzare il funzionamento della Webapp è quindi necessario configurare l'utilizzo dei diversi tipi di memoria mediante l'inserimento della seguente stringa all'interno del file *catalina.bat* presente nella cartella *bin*:

```
set JAVA_OPTS=-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512m -
Dlog4j.debug=true
```

5. Il prossimo passo consiste nell'impostare le credenziali di accesso al Tomcat Manager App, utility per la gestione delle applicazioni. Aprire quindi il file *tomcat-users.xml* presente nella cartella *conf* con un editor di testo ed aggiungere le seguenti righe nella sezione *tomcat-users*.

```
<tomcat-users>
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="root" password="yourpassword" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

6. Il servizio risulta attivo con configurazione di avvio "Manual" che noi setteremo come "Automatic" in modo che all'avvio della macchina verrà avviato anche l'application server Tomcat. Per fare questo è necessario avviare l'utility *tomcat7w.exe*, situato nella cartella *bin*, ed impostare il parametro Startup type nel tab "General".

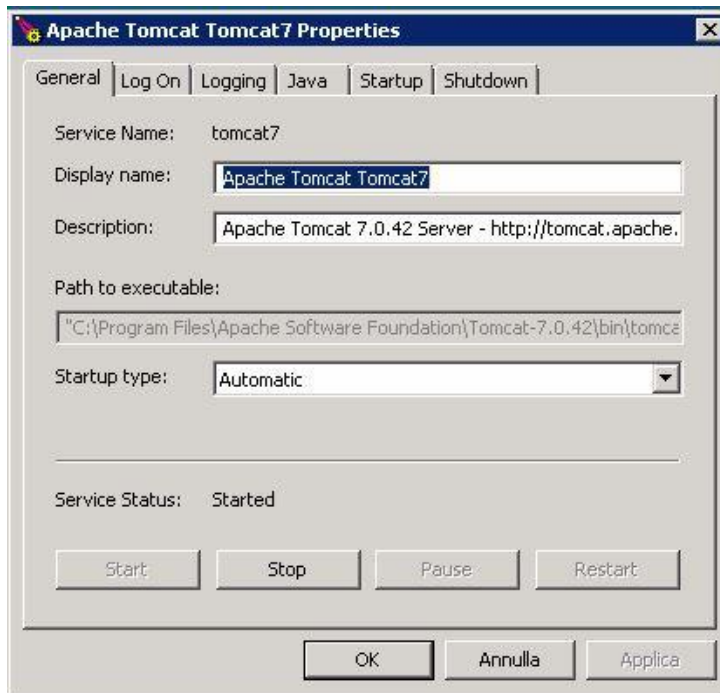


Figura IV-2: Finestra Apache Tomcat Properties

7. Dato che tutte le altre impostazioni di default sono corrette possiamo riavviare la macchina. A questo punto è possibile verificare la correttezza dell'installazione accedendo via web browser al seguente indirizzo <http://localhost:8080>, ricordando che la porta TCP 8080 è la porta usata di default. Tomcat mostrerà la pagina di avvio con l'accesso alle pagine "Server Status", "Manager App" e "Host Manager". L'accesso alla sezione "Manager App" consentirà l'installazione della Webapp.

### IV.3. Installazione e configurazione di MySQL 5.6

MySQL [OR13a] è un RDBMS (sistema di gestione di basi di dati relazionali) cioè permette di gestire database basati sul modello relazionale. La versione che utilizzeremo è denominata Community Edition e si differenzia da quella destinata al mondo aziendale, denominata Enterprise Edition, per i termini di licenza e per le funzionalità. L'url dove scaricare la versione free è la seguente:

*<http://www.mysql.it/downloads/mysql/>*

tramite il menù a tendina scegliamo il Sistema Operativo Windows e scarichiamo la versione di MySQL denominata *Installer for Windows* che include in un solo pacchetto le seguenti componenti:

- MySQL Server 5.5 GA
- MySQL Workbench 5.2 GA
- MySQL Connectors (.Net / ODBC / Java / C / C++) GA
- MySQL Samples and Examples 5.5
- MySQL Documentation 5.5

Avviamo l'installer e clicchiamo su "Install MySQL Products" e poi su "Next". Scegliamo come tipo di Setup quello Custom (personalizzato) in modo da poter scegliere singolarmente i prodotti da

## IV. Installazione e configurazione dell'applicazione

installare. Assicuriamoci quindi che alla voce “Applications” sia selezionato il prodotto MySQL Workbench [OR13b].

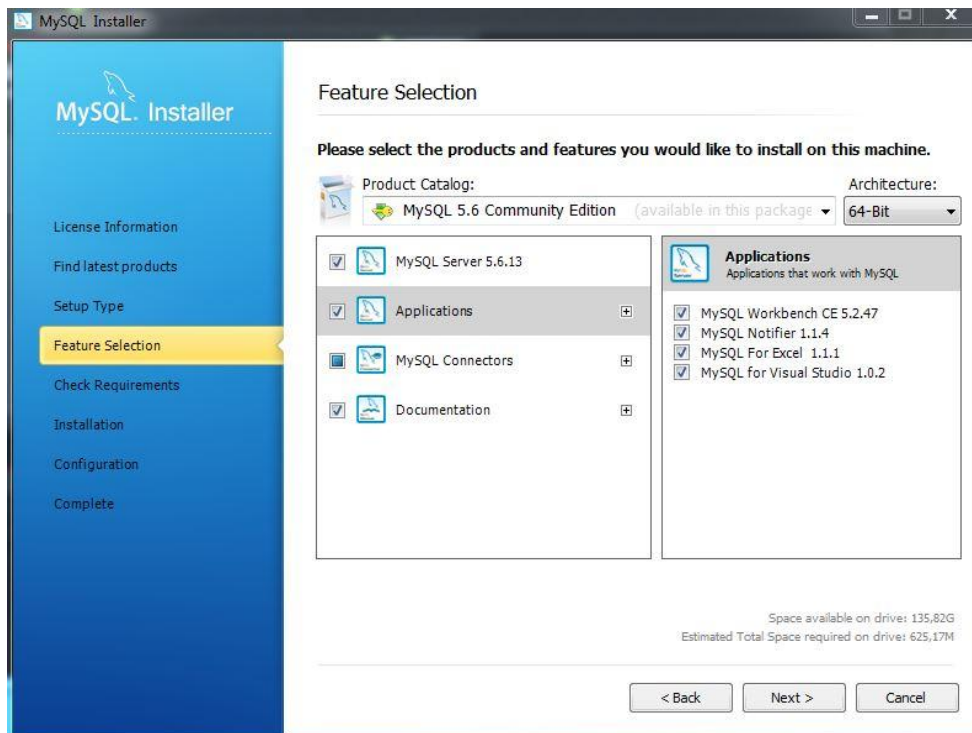


Figura IV-3: Screenshot MySQL Installer (1)

Se tutto procede correttamente dovremmo avere la conferma da parte dell'installer come in figura:

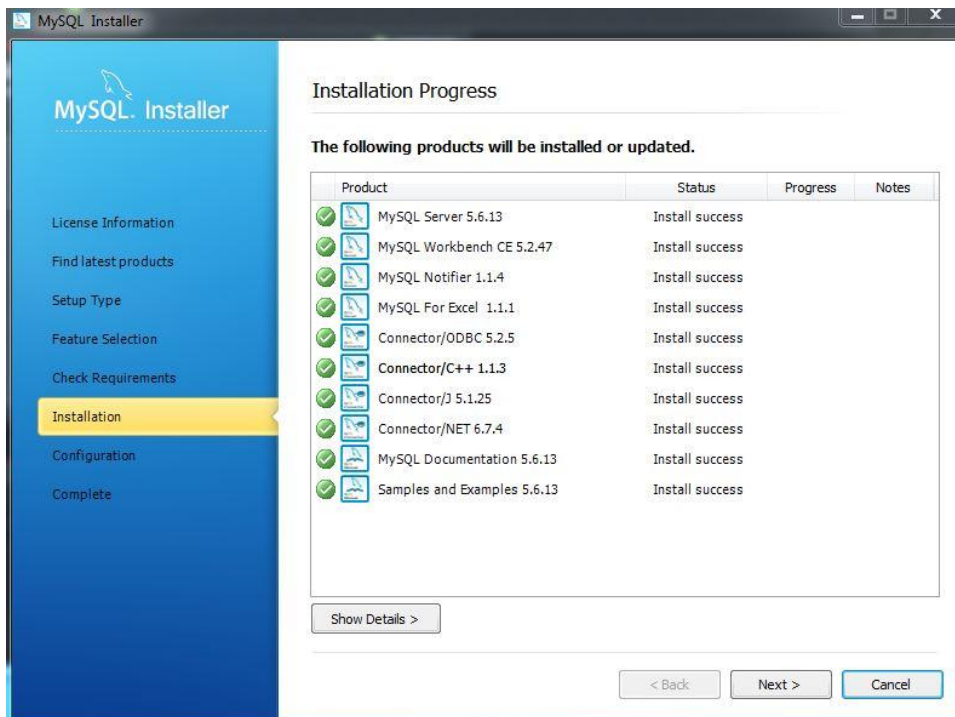


Figura IV-4: Screenshot MySQL Installer (2)



## IV. Installazione e configurazione dell'applicazione

Iniziamo ora la configurazione del MySQL Server selezionando l'opzione Config Type come "Server Machine" e lasciando inalterati gli altri parametri proposti dall'installer come mostrato in nella seguente figura

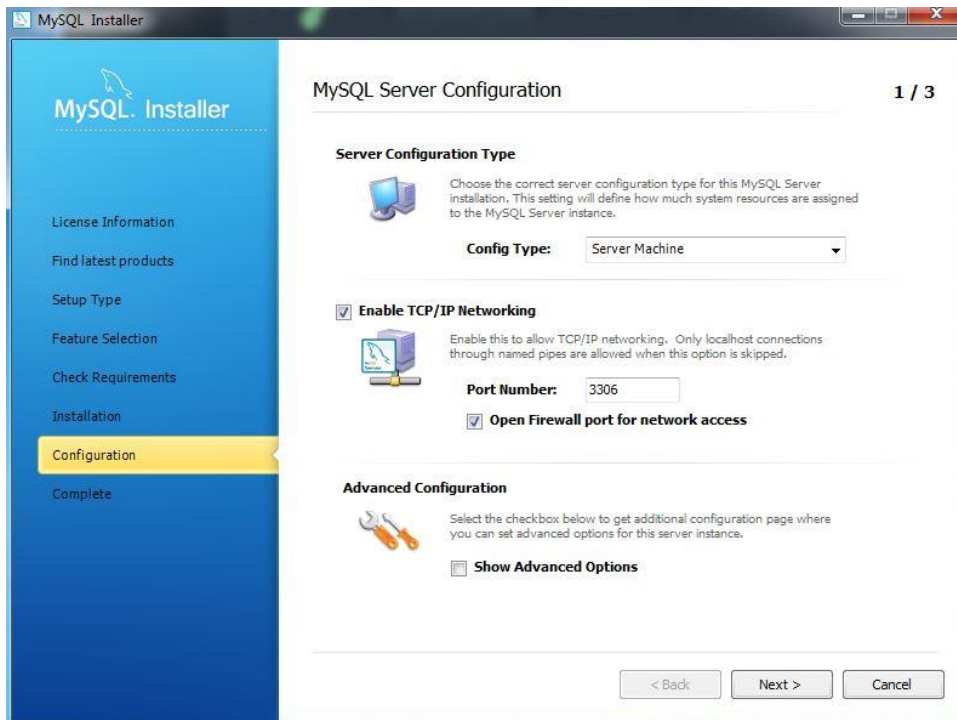


Figura IV-5: Screenshot MySQL Installer (3)

Inseriamo ora la password dell'account utente amministratore root:

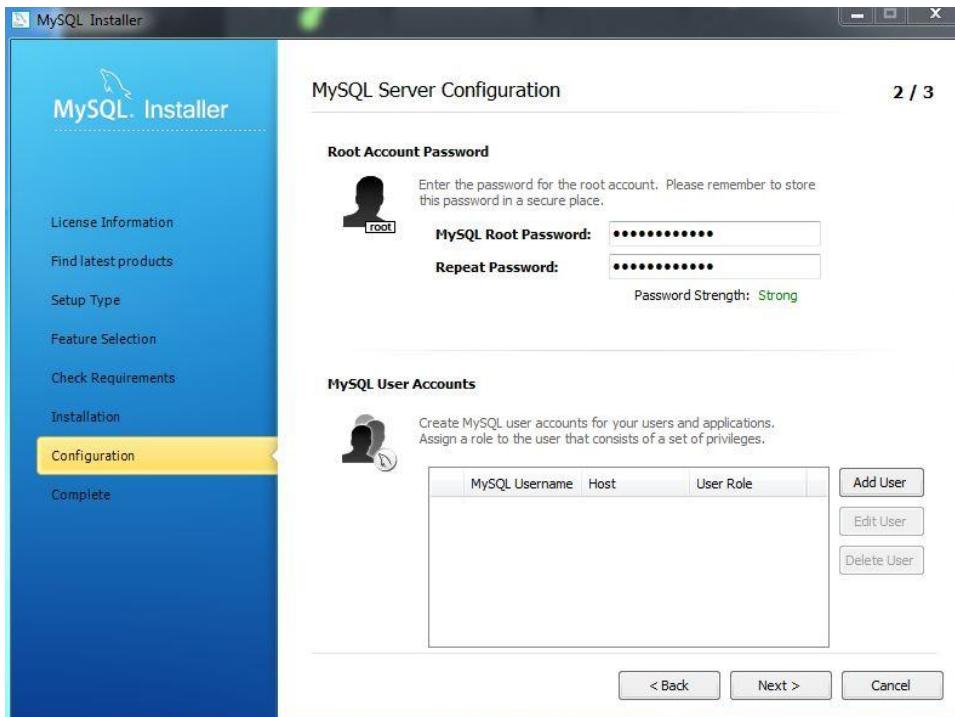


Figura IV-6: Screenshot MySQL Installer (4)

E lasciamo le impostazioni di default per quanto riguarda l'aggiunta tra i servizi di windows:

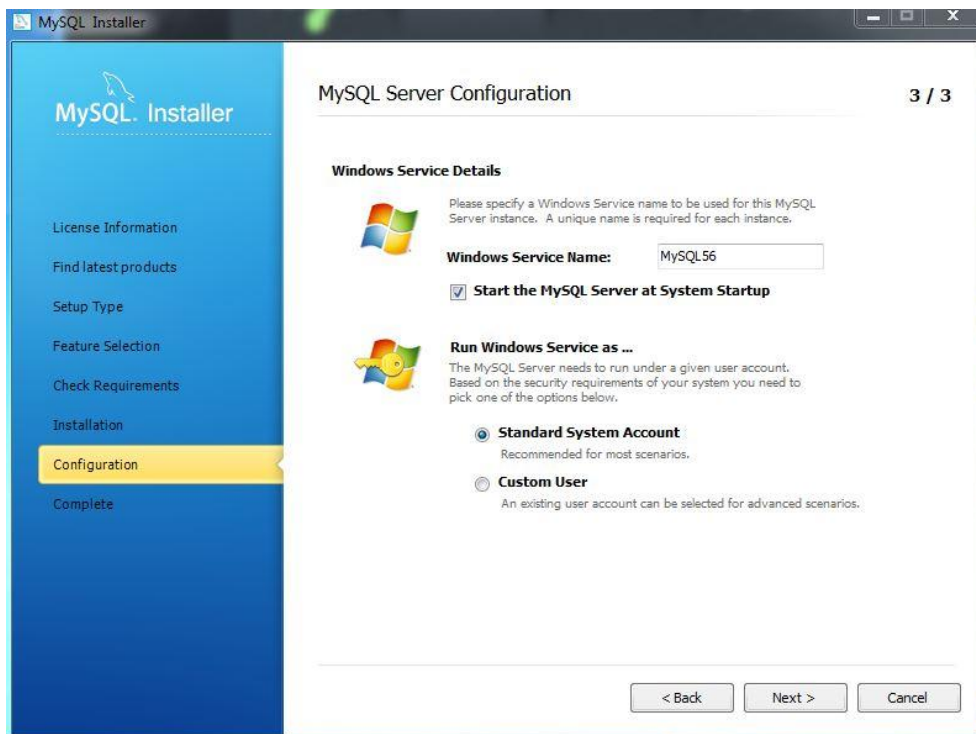
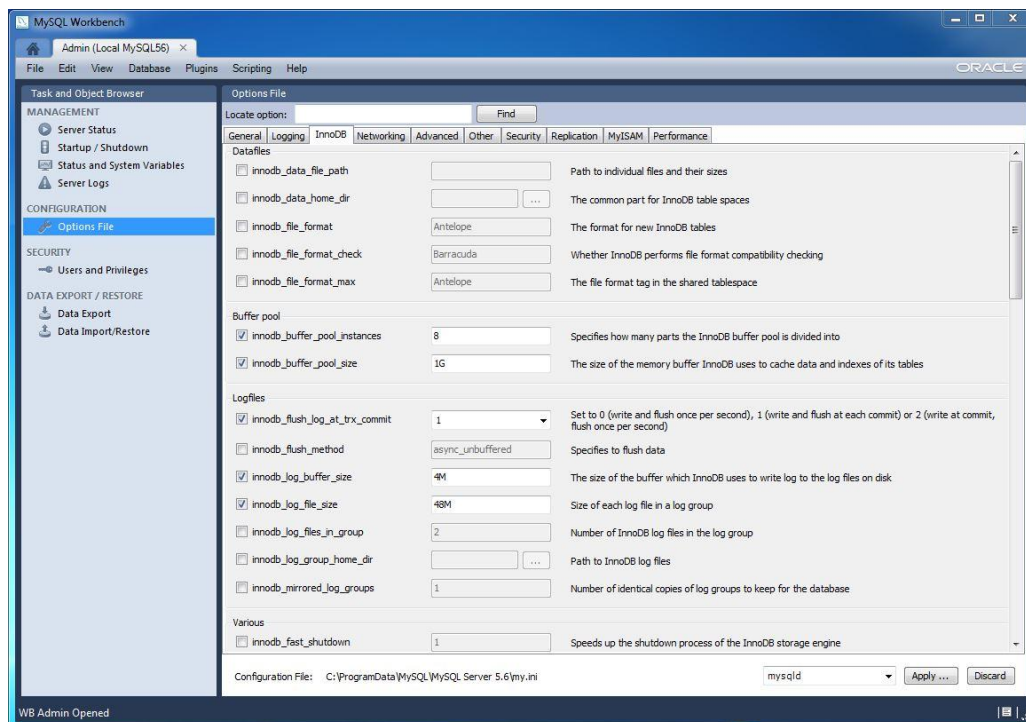


Figura IV-7: Screenshot MySQL Installer (5)

Completata l'installazione e la configurazione di base, andiamo ora a configurare un parametro fondamentale per un efficiente funzionamento della Webapp ovvero il parametro relativo al buffer dei database InnoDB.

**innodb\_buffer\_pool\_size** – Questa proprietà è molto importante utilizzando tabelle di tipo InnoDB. Utilizzando database di tipo InnoDB infatti, dati ed indici vengono memorizzati dal pool, quindi si può pensare di portare al 50% della memoria disponibile nella macchina considerando che il valore minimo di questo parametro è da considerarsi 256M in quanto al di sotto di tale valore la continua scrittura e lettura del disco rallenterebbe notevolmente il sistema.

Per configurare il parametro **innodb\_buffer\_pool\_size** avviamo MySQL Workbench installato, clicchiamo su Server Administration e dopo essersi loggati con le credenziali di amministratore ci portiamo nella sezione Option File e settiamo il parametro di interesse come in figura:



**Figura IV-8: Screenshot MySQL Installer (6)**

Il valore di 1GB dovrebbe essere sufficiente per la gestione da parte del sistema della maggioranza dei sistemi Cloud privati.

Ora che abbiamo installato e configurato il database server procediamo con l'installazione del database della Webapp. Per fare ciò andiamo nella pagina iniziale di MySQL Workbench [OR13b] e selezioniamo questa volta SQL Development. Dopo esserci loggati nuovamente eseguiamo lo script SQL contenuto nel file `db.sql`.

Il codice SQL eseguito consente la creazione del database, delle tabelle, delle relazioni e del popolamento del database con i dati iniziali per un corretto avvio della Webapp.

Il risultato dell'operazione dovrebbe essere quello mostrato in figura:

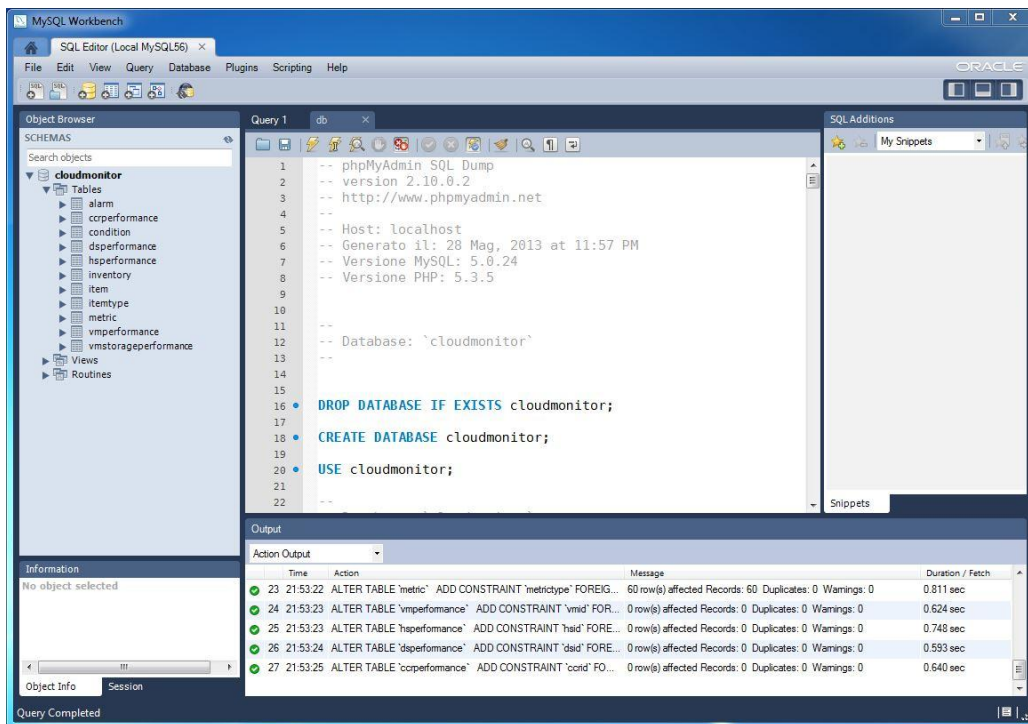


Figura IV-9: Screenshot MySQL Installer (7)

Il database “cloudmonitor” è stato creato con le relative tabelle. E’ ovviamente possibile cambiare il nome del database variando lo stesso nelle prime tre righe dello script.

#### IV.4. Installazione della Web Application

Innanzitutto scompattiamo il file con estensione war per accedere al file xml di configurazione appConfig.xml disponibile al seguente percorso \resources\xml e con un editor di testo settiamo i seguenti parametri all’interno dei rispettivi tag:

Sezione MySQL:

*DBHostName*: indirizzo del DBMS MySQL

*DBUser*: username

*DBPassword*: password

*DBName*: nome del DB

Sezione SMTP

*smtpHostName*: indirizzo del server di posta in uscita

*smtpPort*: porta TCP del server di posta in uscita

*smtpUsername*: username

*smtpPassword*: password

Sezione mailalarm

*mailObject*: oggetto della email di allarme

*mailSendTo*: indirizzo email al quale inviare le email di allarme

### Sezione vcenter

*vcenterHostName*: indirizzo del vCenter server

*vcenterUsername*: username di accesso a vCenter server

*vcenterPassword*: password di accesso a vCenter server

### Sezione Webapp

*webappUsername*: username di accesso alla Webapp

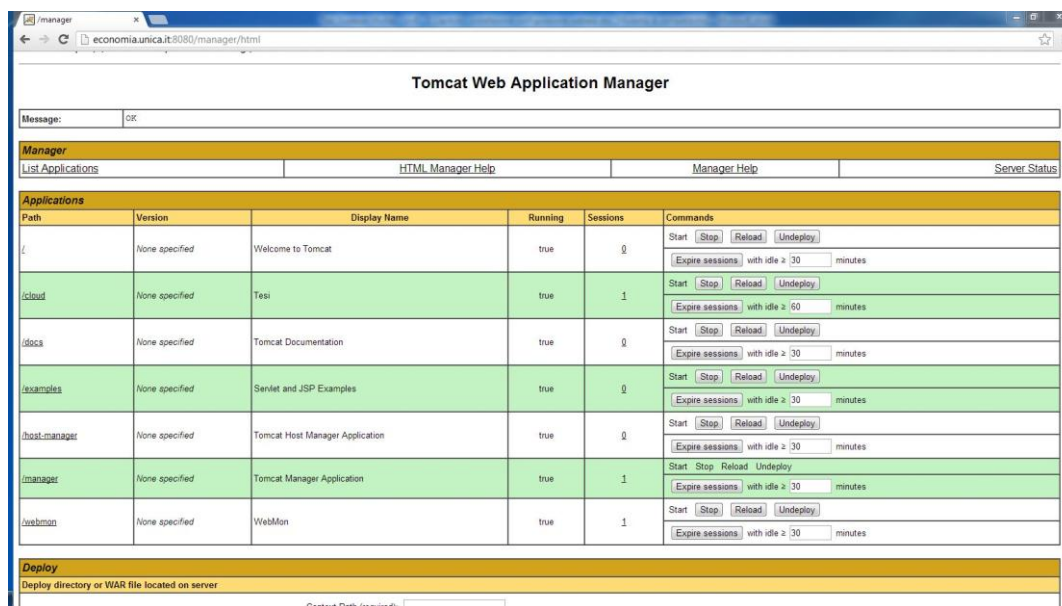
*webappPassword*: password di accesso alla Webapp

### Sezione Monitoring

*monitoringDelay*: tempo, in secondi, tra due cicli di monitoring

Salviamo e ricompattiamo il file prestando attenzione all'estensione dello stesso che dovrà essere .war per poter esser correttamente installato da Tomcat.

A questo punto accediamo con un web browser all'Application server Tomcat installato in precedenza e clicchiamo su "Manager App". Nella sezione "WAR file to deploy" selezioniamo il file war e clicchiamo su "Deploy". Se tutto è andato correttamente la Webapp risulta ora installata



**Figura IV-10: Tomcat Manager App**

e raggiungibile via web browser all'indirizzo:

*http://indirizzodelservertomcat:8080/nomedelfilewar/*

# V. Uso sperimentale e validazione

## V.1. Introduzione

In questo capitolo verrà illustrato il funzionamento del sistema di monitoring, alarming, reporting per l'infrastruttura Cloud della Facoltà di Scienze Economiche, Giuridiche e Politiche dell'Università degli Studi di Cagliari

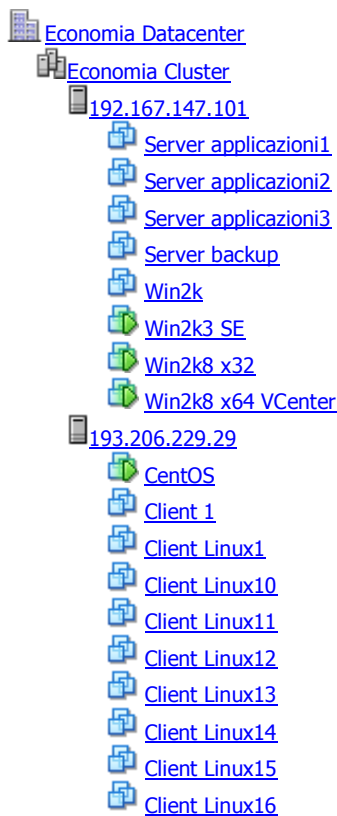
## V.2. L'infrastruttura Cloud















































Il Cloud su cui verrà implementato il sistema di monitoraggio, consente la virtualizzazione dei sistemi server e client nella suddetta Facoltà. Dal punto di vista hardware l'infrastruttura è composta da 2 host IBM xSeries dotati di processore Intel Xeon, da un apparato di rete switch 100/1000 (Gigabit).

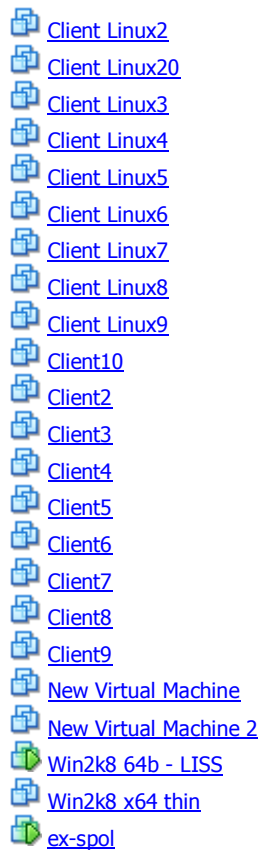
L'infrastruttura Cloud è implementata mediante il prodotto VMware vSphere 5.0 Essential Kit che consente la creazione di Cloud su infrastrutture fino ad un massimo di 3 host fisici con massimo 2 processori l'uno. La versione del prodotto comprende sia l'hypervisor per gli host sia il sistema di gestione centralizzata vCenter, per mezzo del quale è possibile gestire l'intera infrastruttura.

Sull'infrastruttura creata sono in funzione 40 virtual machine dotate di sistema operativo sia server che client (Windows e Linux).

In particolare la struttura dell'inventario è la seguente:



-  [Client Linux17](#)
-  [Client Linux18](#)
-  [Client Linux19](#)
-  [Client Linux2](#)
-  [Client Linux20](#)
-  [Client Linux3](#)
-  [Client Linux4](#)
-  [Client Linux5](#)
-  [Client Linux6](#)
-  [Client Linux7](#)
-  [Client Linux8](#)
-  [Client Linux9](#)
-  [Client10](#)
-  [Client2](#)
-  [Client3](#)
-  [Client4](#)
-  [Client5](#)
-  [Client6](#)
-  [Client7](#)
-  [Client8](#)
-  [Client9](#)
-  [New Virtual Machine](#)
-  [New Virtual Machine 2](#)
-  [Win2k8 64b - LISS](#)
-  [Win2k8 x64 thin](#)
-  [ex-spol](#)
-  [datastoreX3250](#)
-  [192.167.147.101](#)
-  [Server applicazioni1](#)
-  [Server applicazioni2](#)
-  [Server applicazioni3](#)
-  [Server backup](#)
-  [Win2k](#)
-  [Win2k3 SE](#)
-  [Win2k8 x32](#)
-  [Win2k8 x64 VCenter](#)
-  [datastoreX3400](#)
-  [193.206.229.29](#)
-  [CentOS](#)
-  [Client 1](#)
-  [Client Linux1](#)
-  [Client Linux10](#)
-  [Client Linux11](#)
-  [Client Linux12](#)
-  [Client Linux13](#)
-  [Client Linux14](#)
-  [Client Linux15](#)
-  [Client Linux16](#)
-  [Client Linux17](#)
-  [Client Linux18](#)
-  [Client Linux19](#)



### V.3. Installazione e configurazione del sistema

Il sistema di monitoraggio è stato installato su una virtual machine dotata di sistema operativo Windows 2008 Server all'interno della stessa infrastruttura, nello stesso host fisico che ospita la virtual machine relativa al sistema di gestione vCenter, in tal modo si minimizzano i tempi di accesso al sistema di gestione del Cloud mediante le API a disposizione.

Come indicato nel capitolo relativo all'installazione e alla configurazione, il sistema è stato installato in un ambiente dotato di DBMS MySQL 5, e di application container Apache Tomcat 7. Sia il DBMS che l'application container sono stati configurati in modo da ottimizzare l'utilizzo della memoria RAM sia per la gestione delle tabelle del database del sistema sia per le diverse tipologie di memoria della Java Virtual Machine.

L'analisi sperimentale dei tempi di recupero delle metriche di performance per ogni ciclo sono riportate nella seguente tabella:

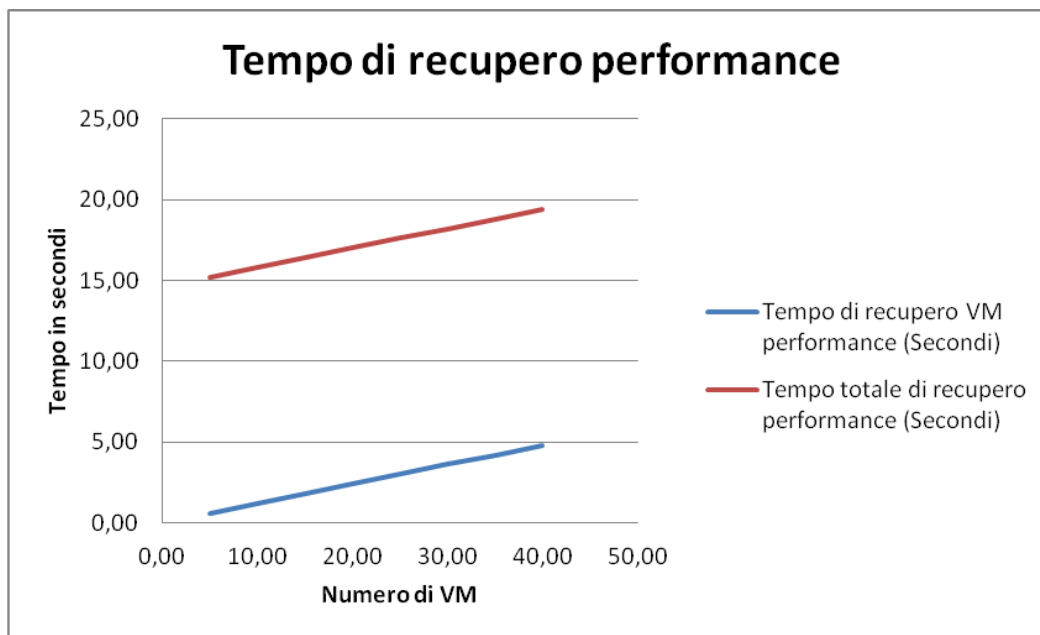
VM Attive	Tempo di recupero performance VM (Secondi)	Tempo totale di recupero performance (Secondi)
5	0.6	15.2
10	1.2	15.8
15	1.8	16.4
20	2.4	17.0



25	3.0	17.6
30	3.6	18.2
35	4.2	18.8
40	4.8	19.4

**Tabella V-1: Tempo di recupero metriche di performance in funzione del numero di Virtual Machine attive**

Il tempo di recupero delle metriche di performance mostra quindi un andamento lineare in funzione del numero di VM attive come mostrato nel seguente grafico:



**Figura V-1: Tempo di recupero delle metriche di performance in funzione del numero di Virtual Machine attive**

## V.4. Visualizzazione delle proprietà e delle metriche di performance

Dal punto di vista logico l'infrastruttura è costituita da un Datacenter. Il Datacenter è costituito a sua volta da un Cluster composto da due Host System. Il Cluster ospita quaranta Virtual Machine. Sono presenti due Datastore associati agli Host System.

Nei prossimi sottoparagrafi è riportato l'utilizzo del sistema per il recupero delle proprietà e delle metriche di performance per alcuni elementi dell'infrastruttura.

### V.4.1. Datacenter

Nell'infrastruttura è presente un unico Datacenter, sul quale sono rilevabili proprietà e metriche e statistiche di performance per gli elementi ad esso associati.

### V.4.1.1. Proprietà

Per gli elementi di tipo Datacenter sono rilevabili 6 proprietà. Un esempio di proprietà rilevate per un elemento di tipo datacenter sono mostrate nella seguente tabella:

Datacenter - Economia Datacenter	
GENERAL INFO	
Name	Economia Datacenter
Alarm Actions Enabled	true
Config Status	GRAY
Overall Alarm Status	GRAY
Datasores Number	2
Networks Number	2

**Tabella V-2: Proprietà rilevate per un elemento Datacenter del Cloud**

### V.4.1.2. Metriche di performance

Per gli elementi di tipo Datacenter non sono disponibili metriche di performance, sono invece disponibili le metriche aggregate degli elementi associati quali Cluster, Host System, Virtual Machine e Datastore. Di seguito un esempio di statistiche visualizzate dal sistema di monitoraggio per un elemento Datacenter del Cloud con i seguenti parametri:

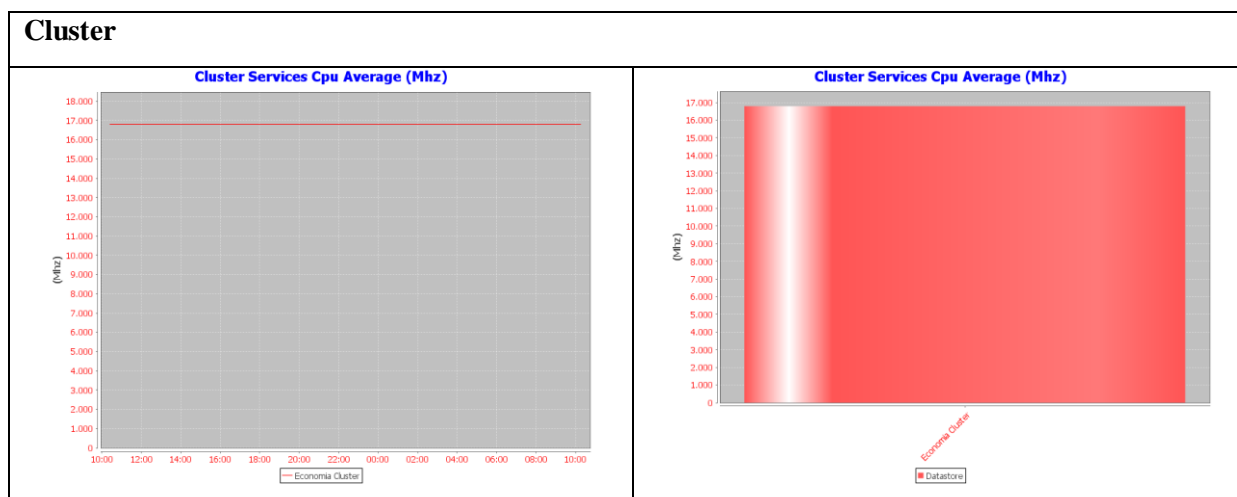
*Datacenter: Economia Datacenter*

*Search parameter*

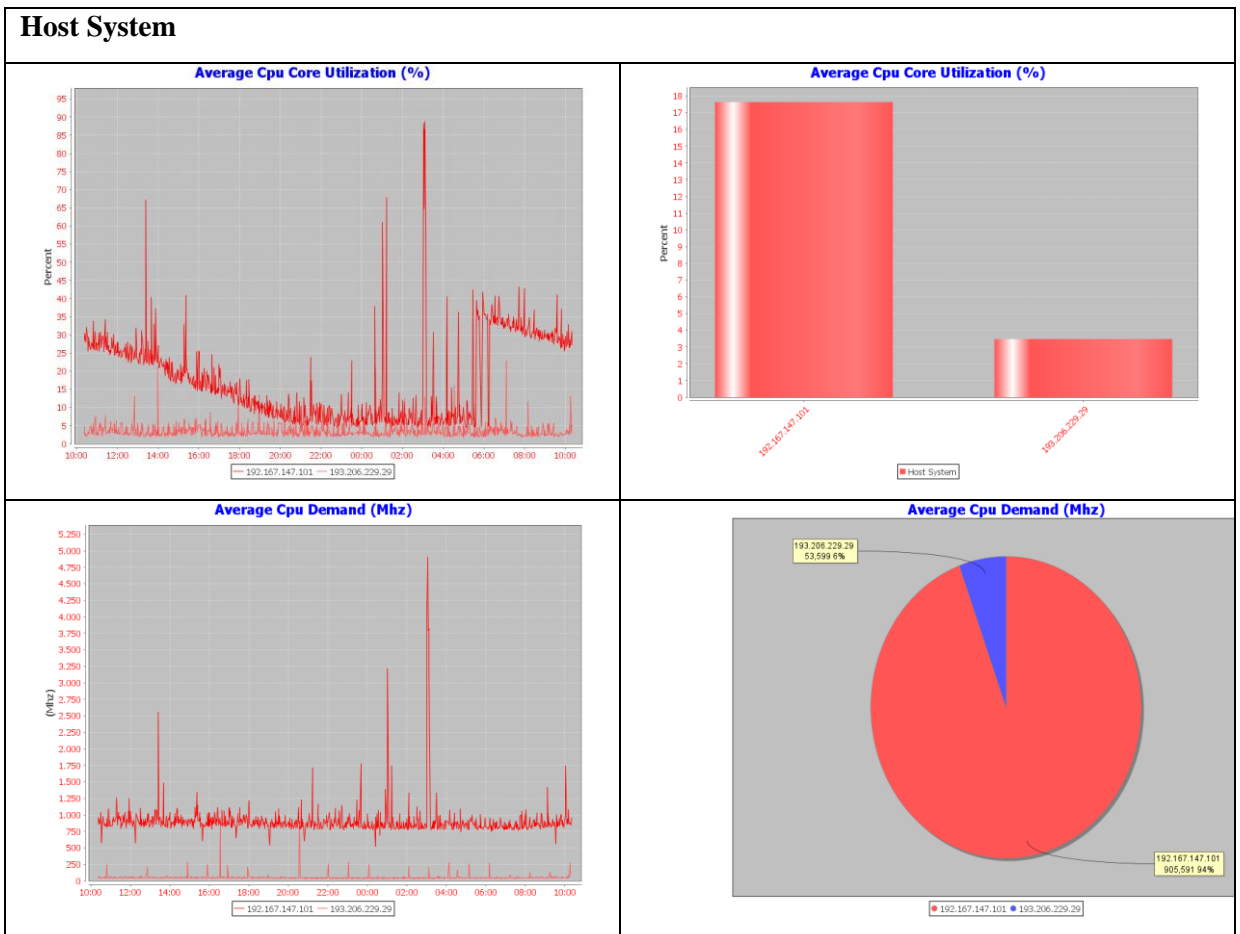
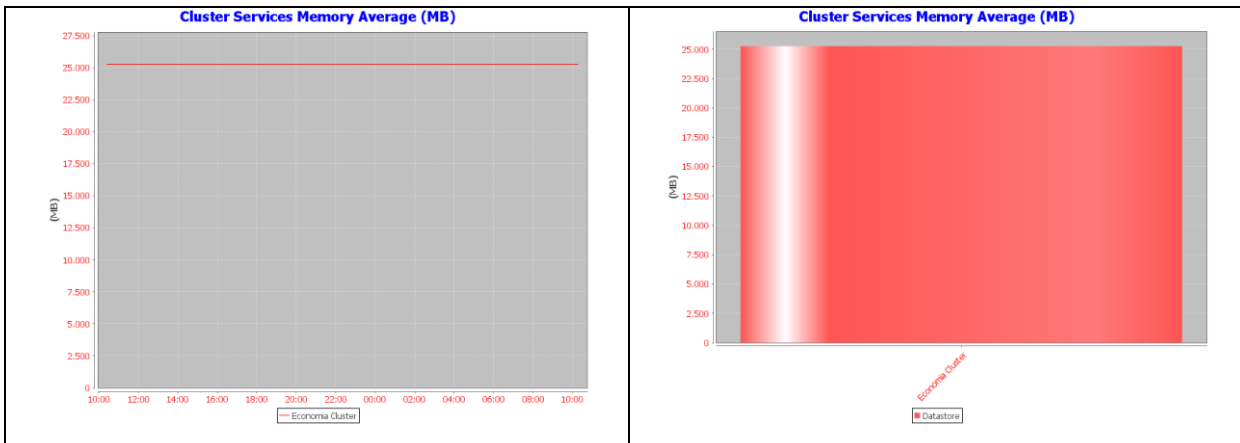
*Minute: one day*

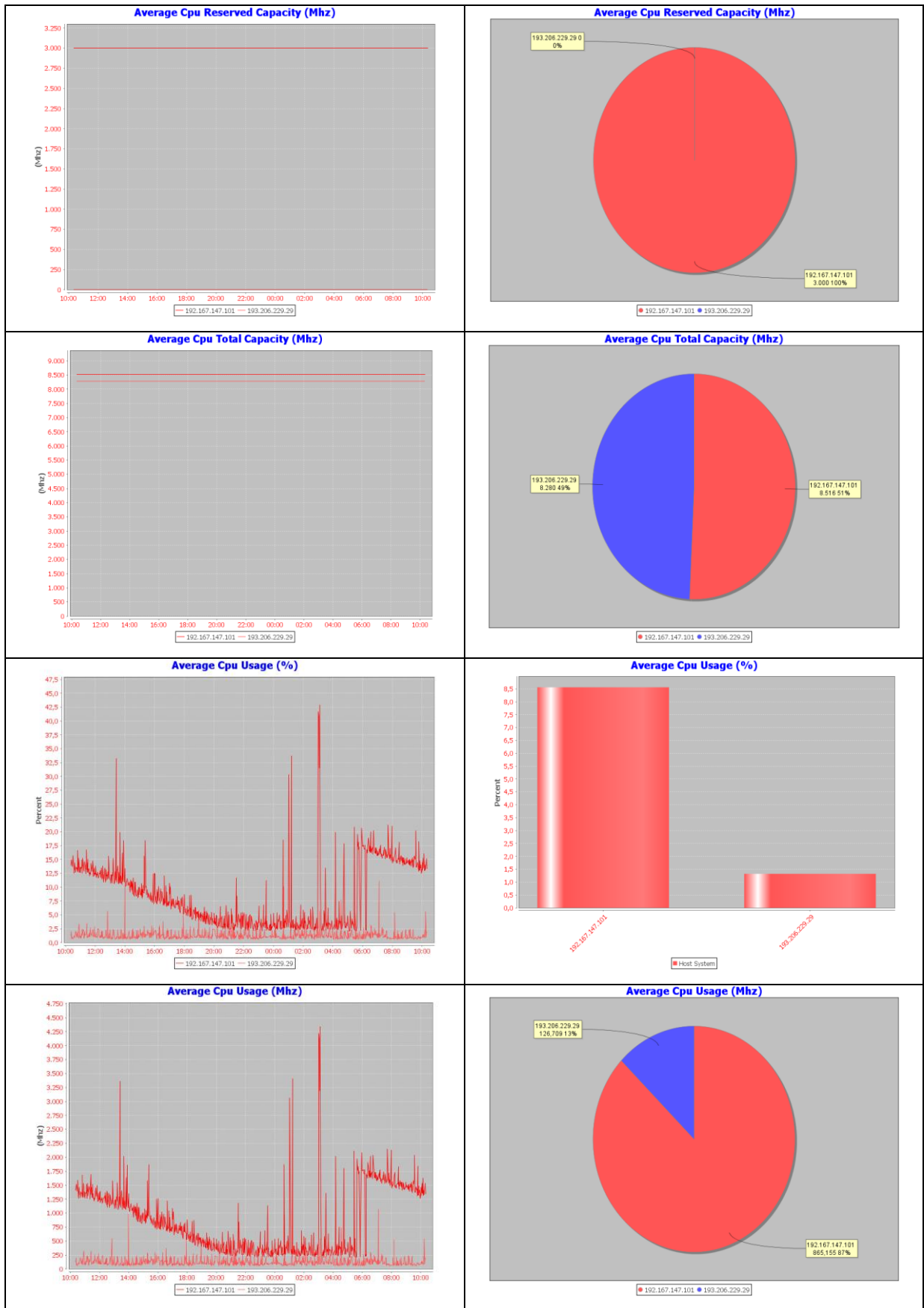
*Sorting type: descendant*

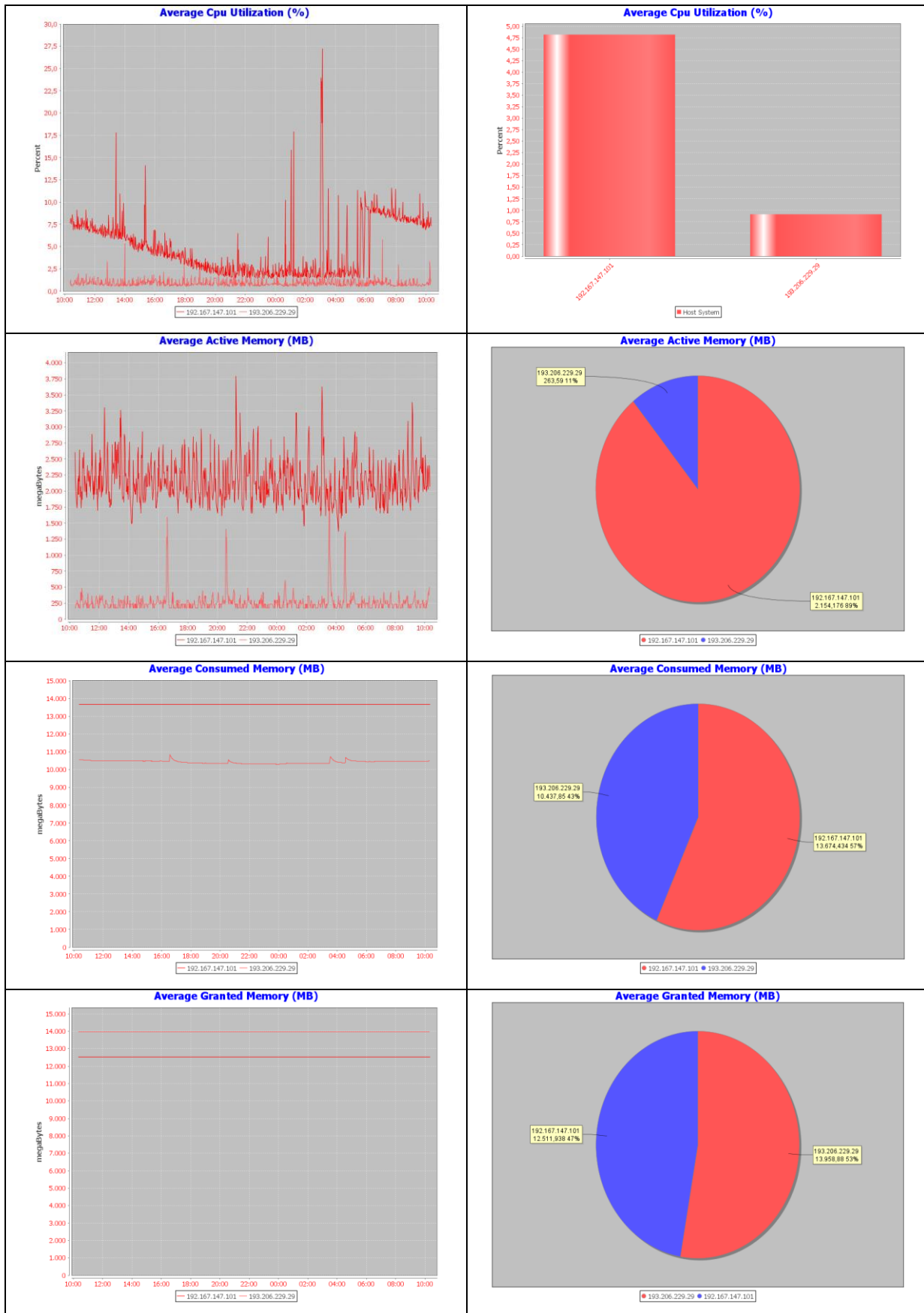
*Chart value limit: 20*

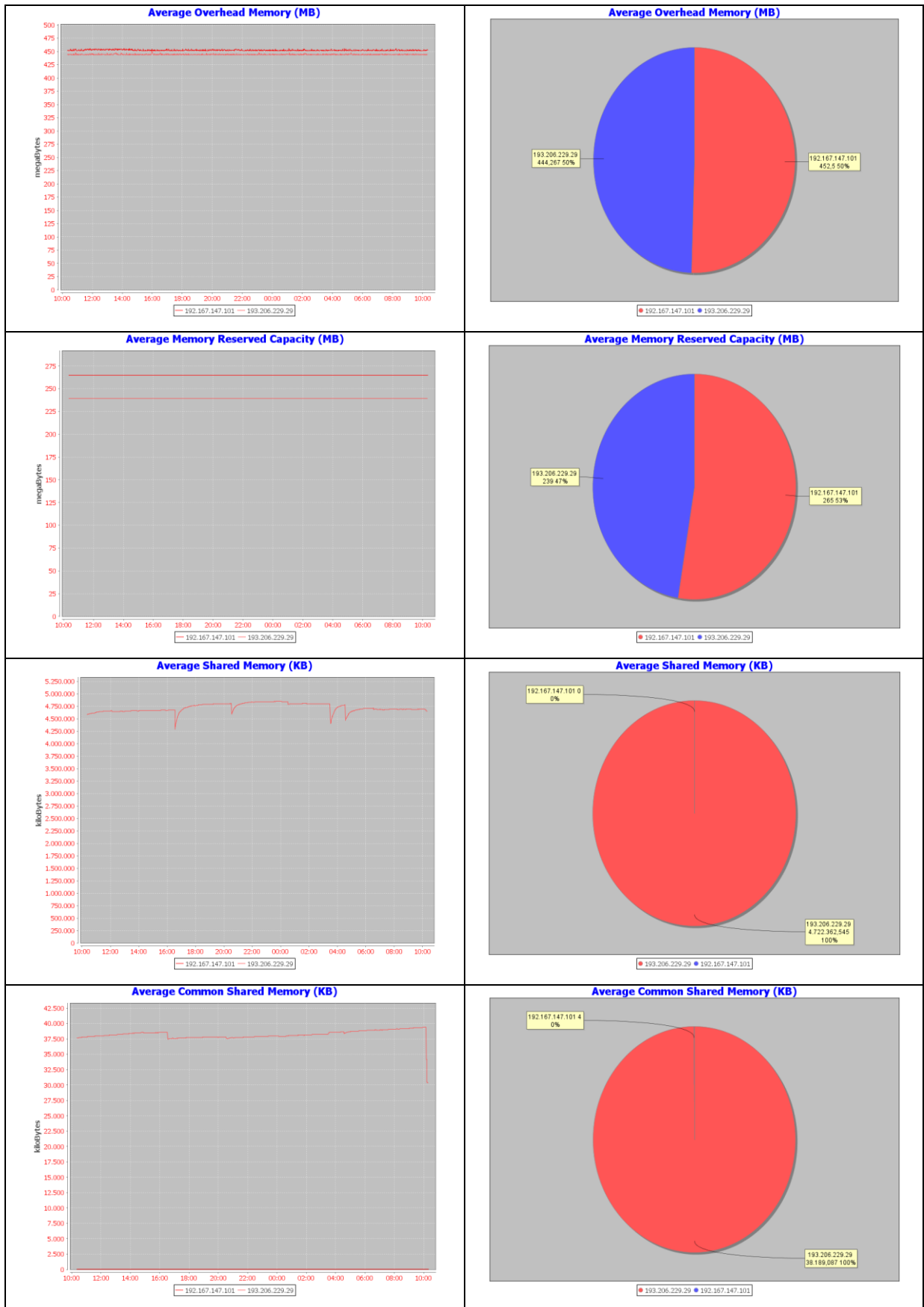


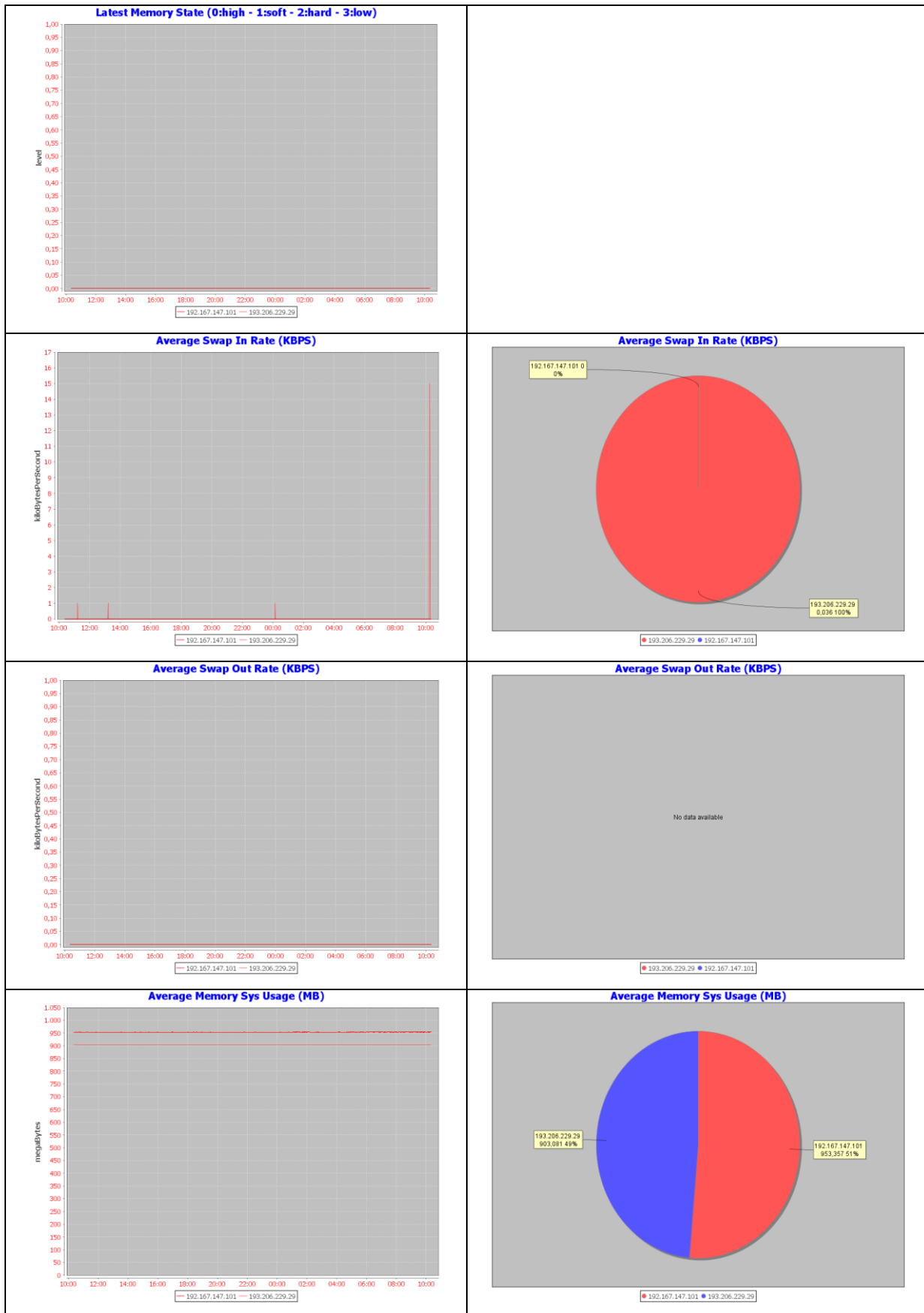
## V. Uso sperimentale - validazione

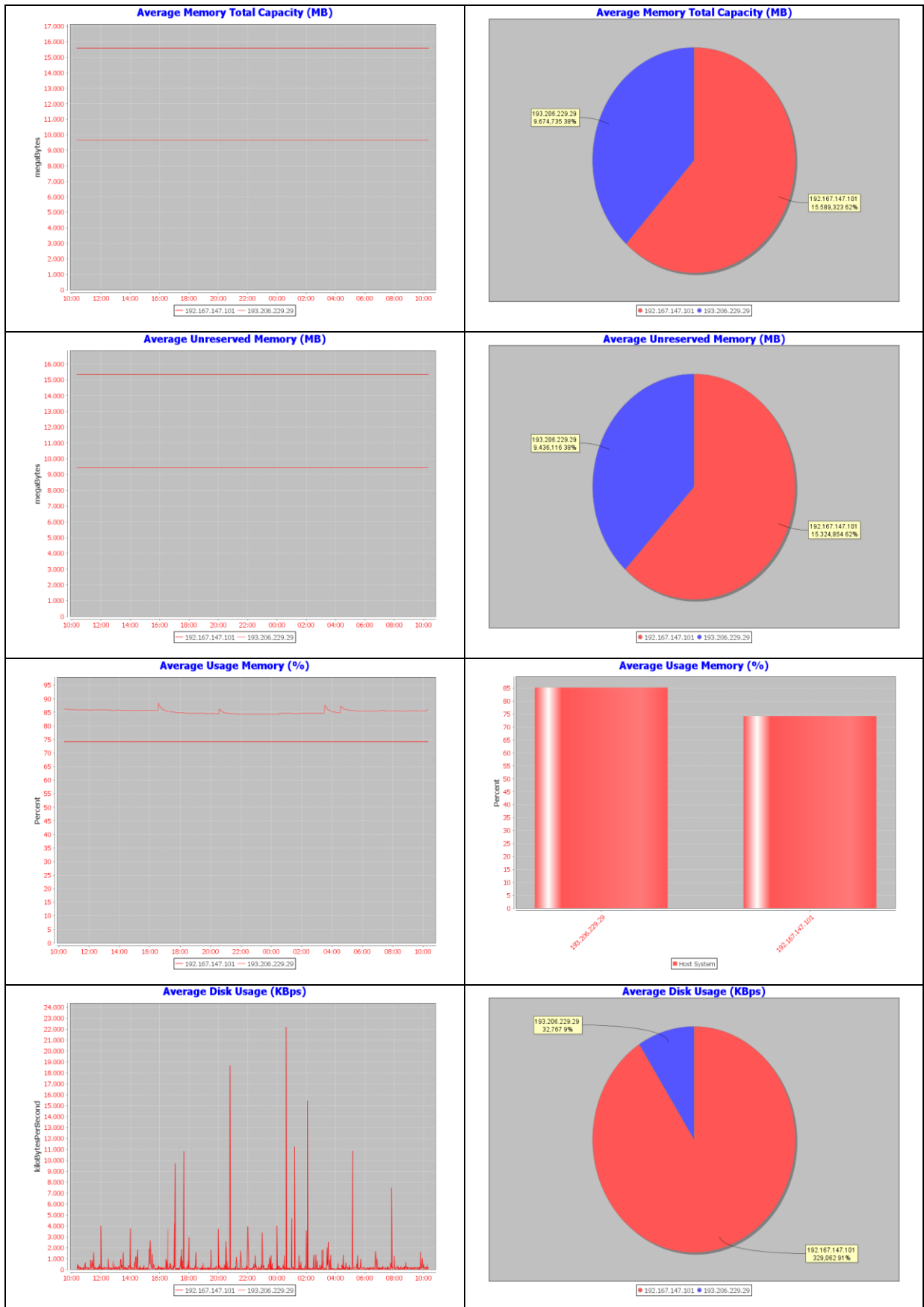






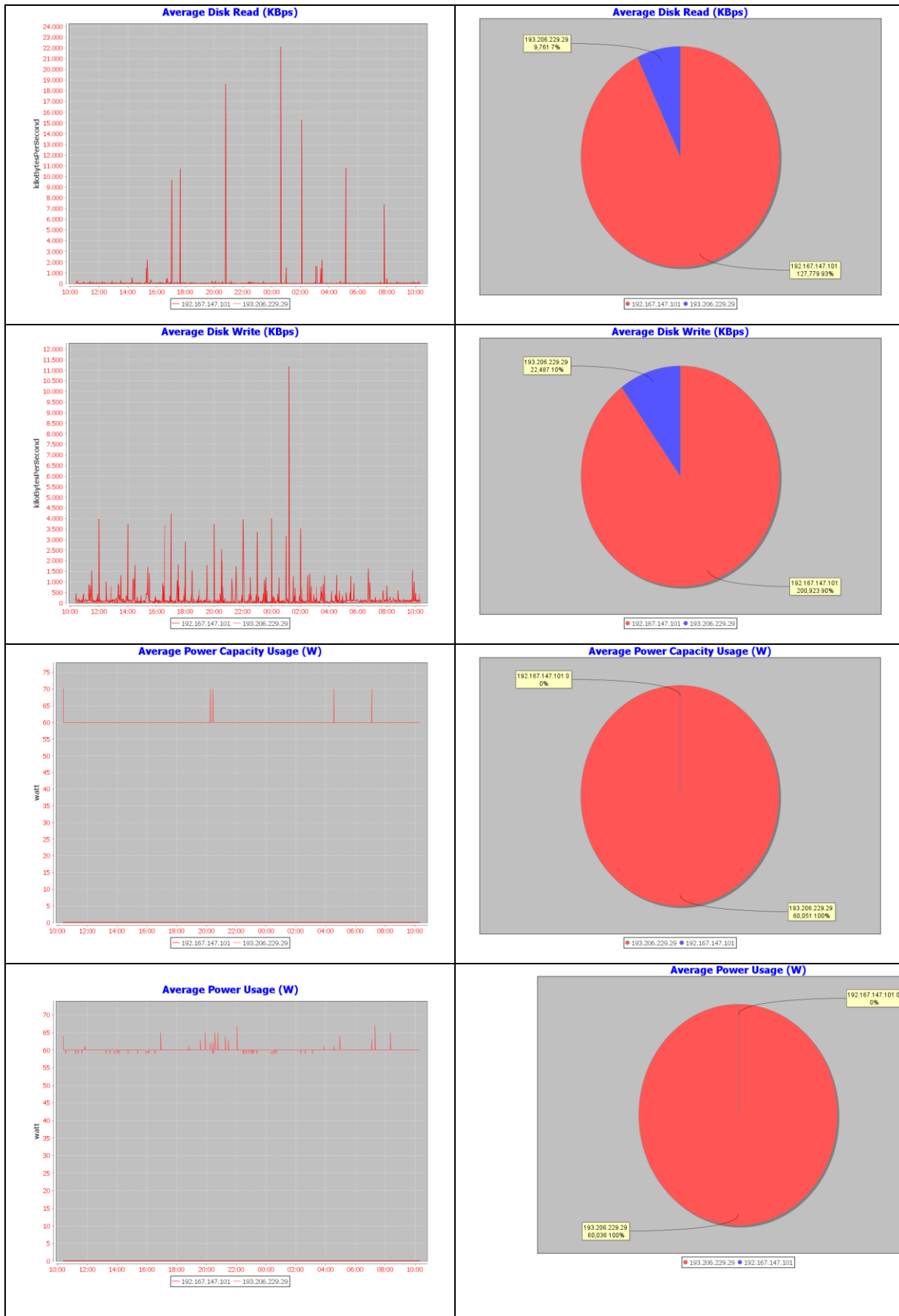




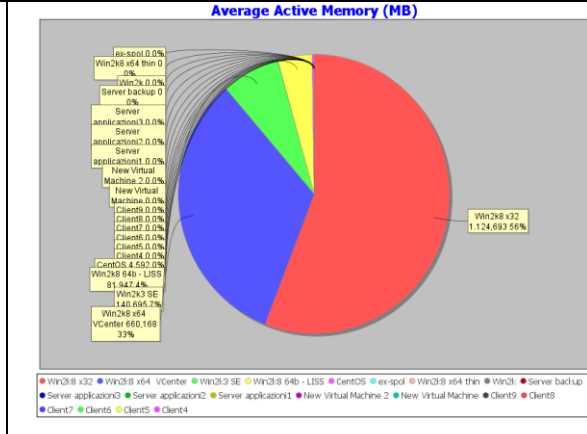
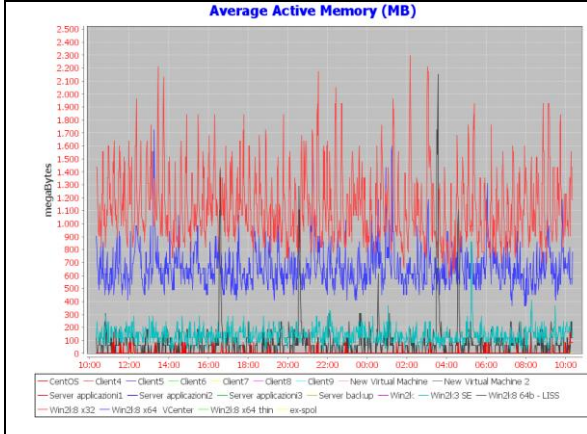
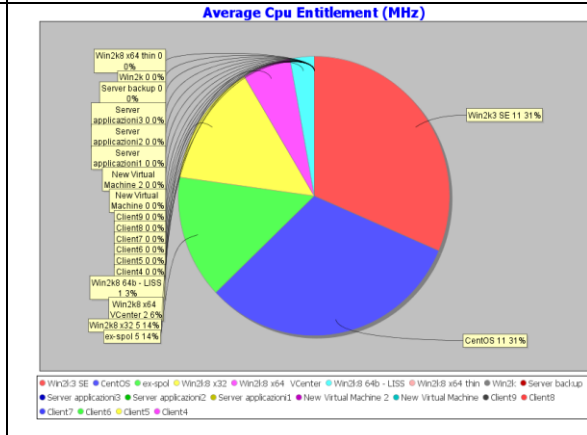
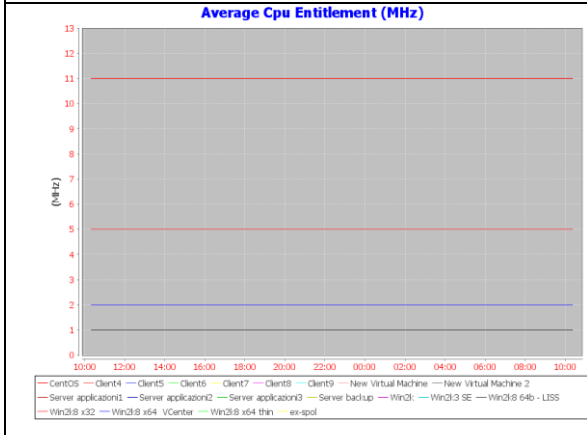
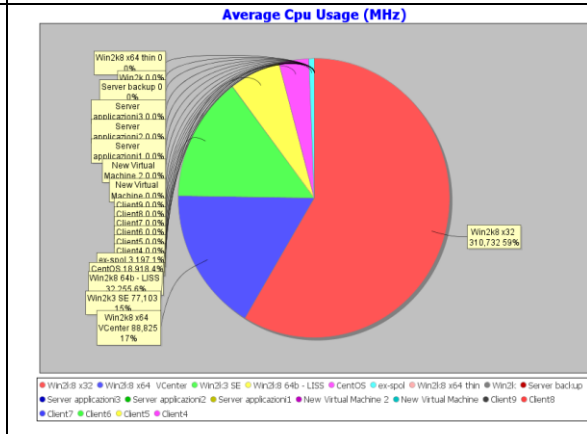
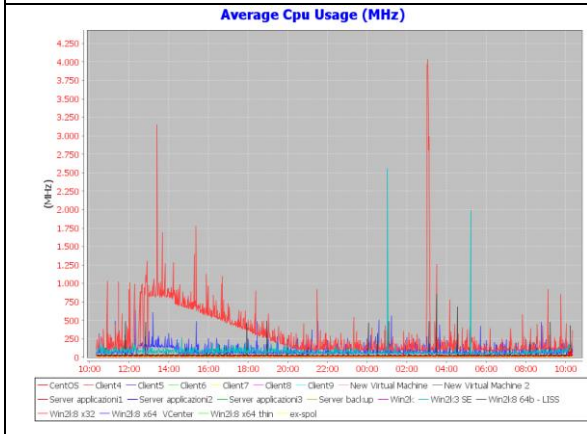
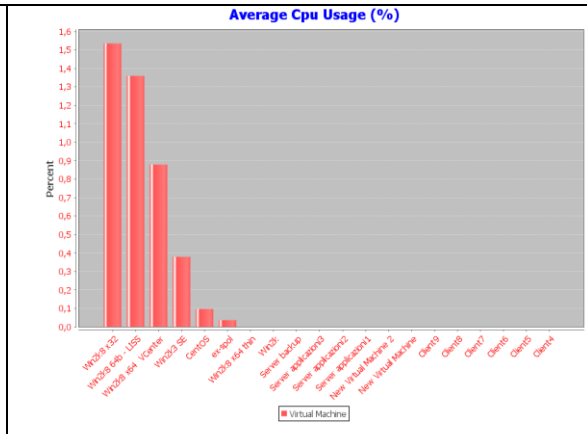
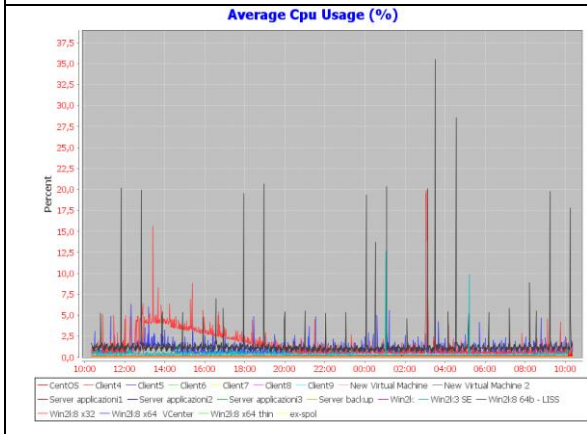


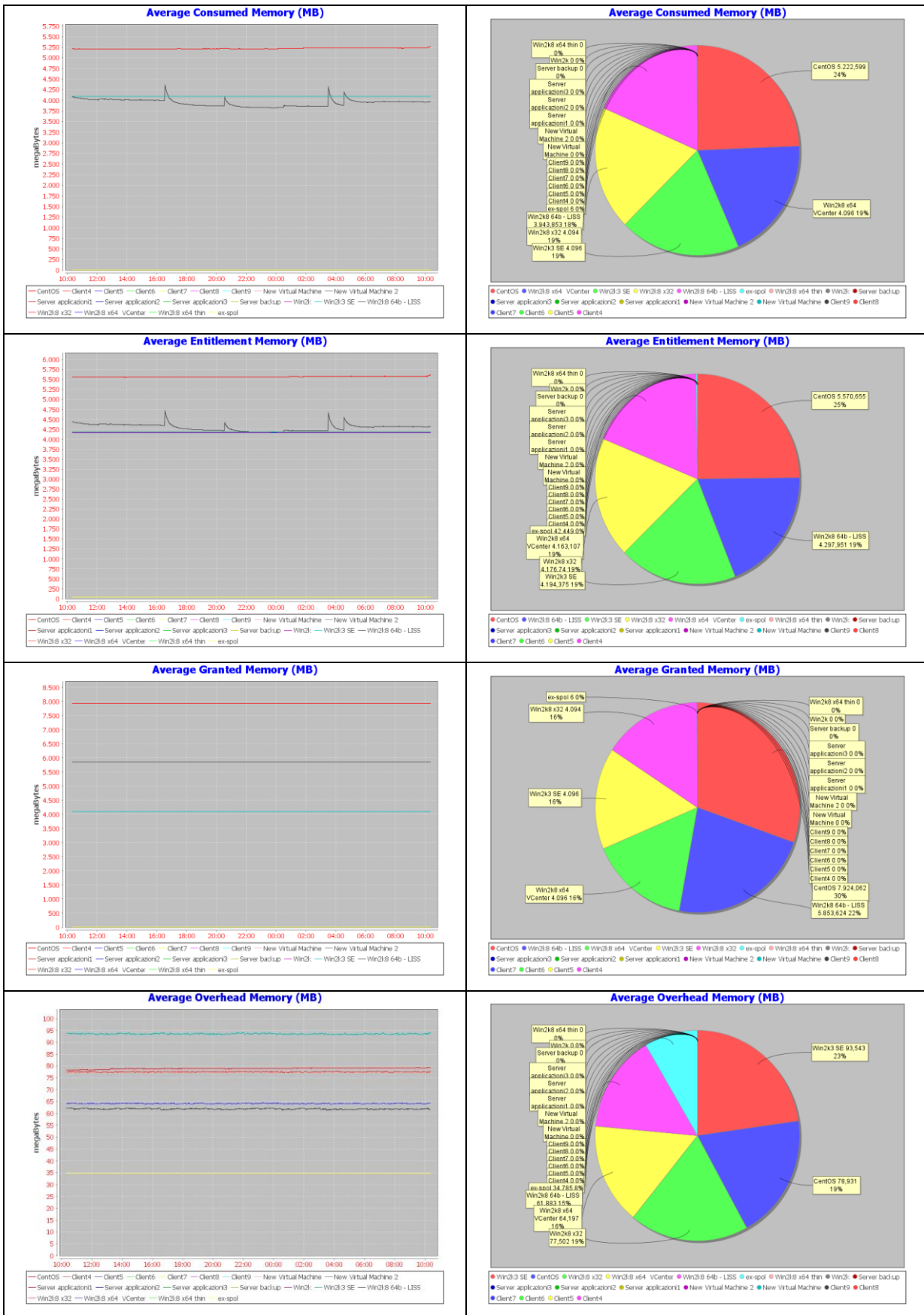


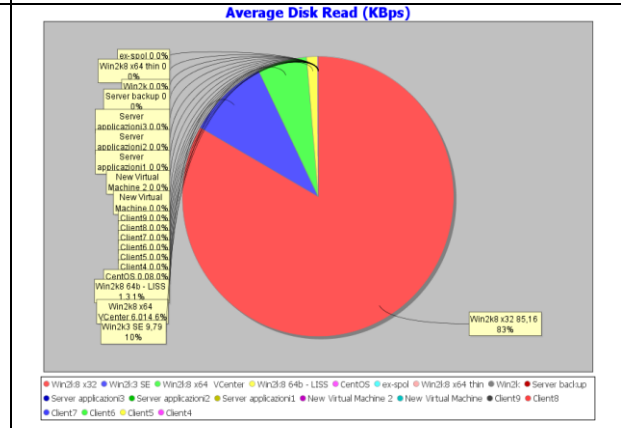
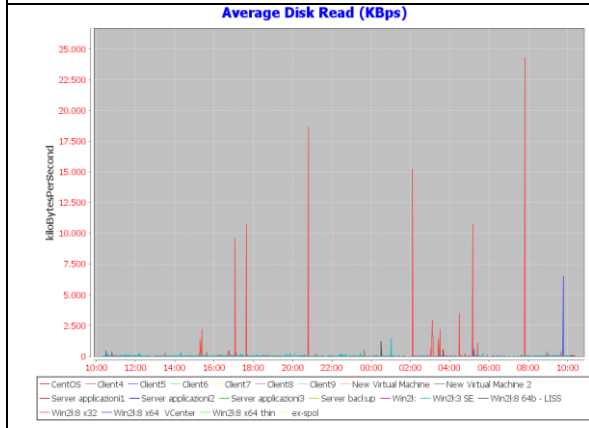
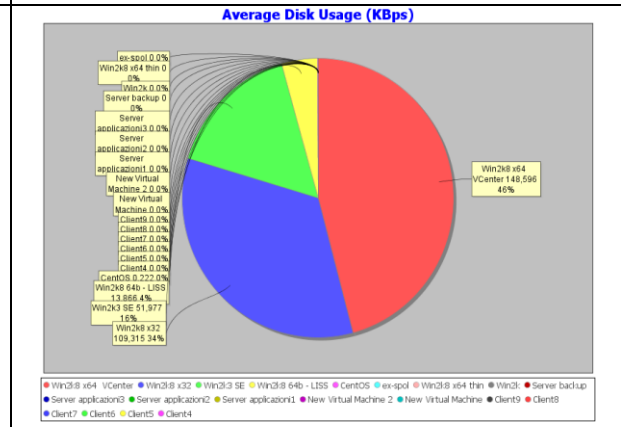
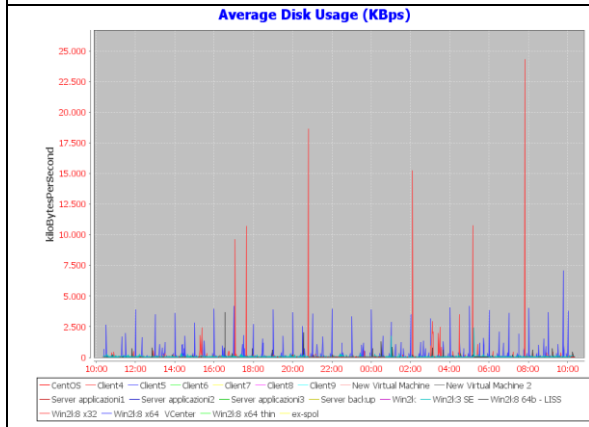
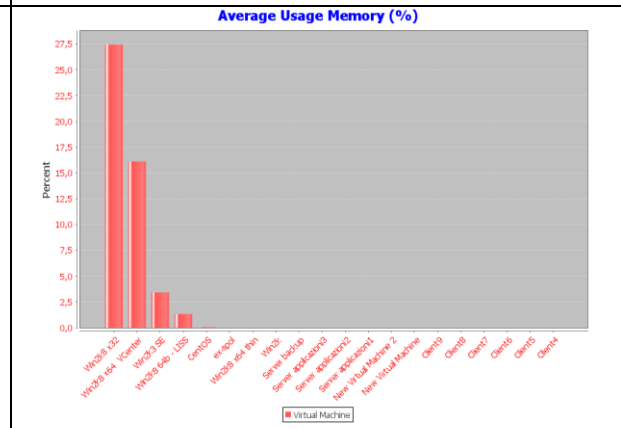
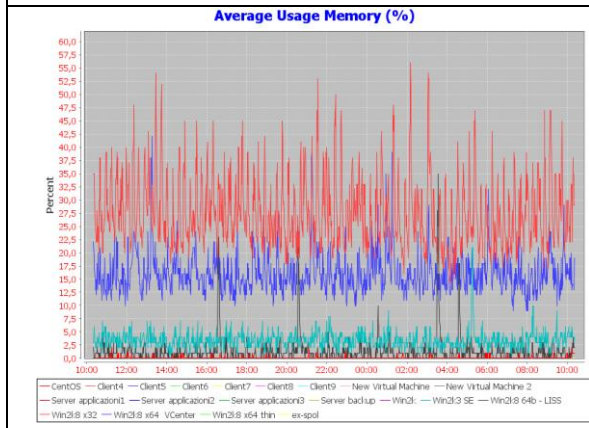
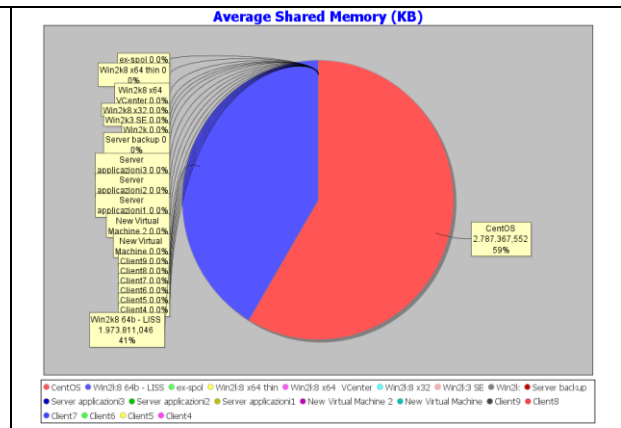
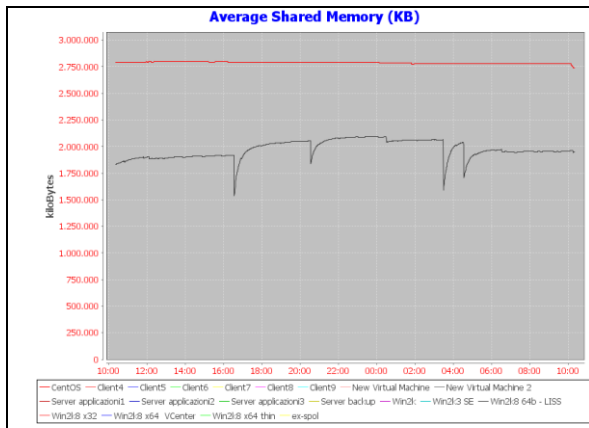
## V. Uso sperimentale - validazione



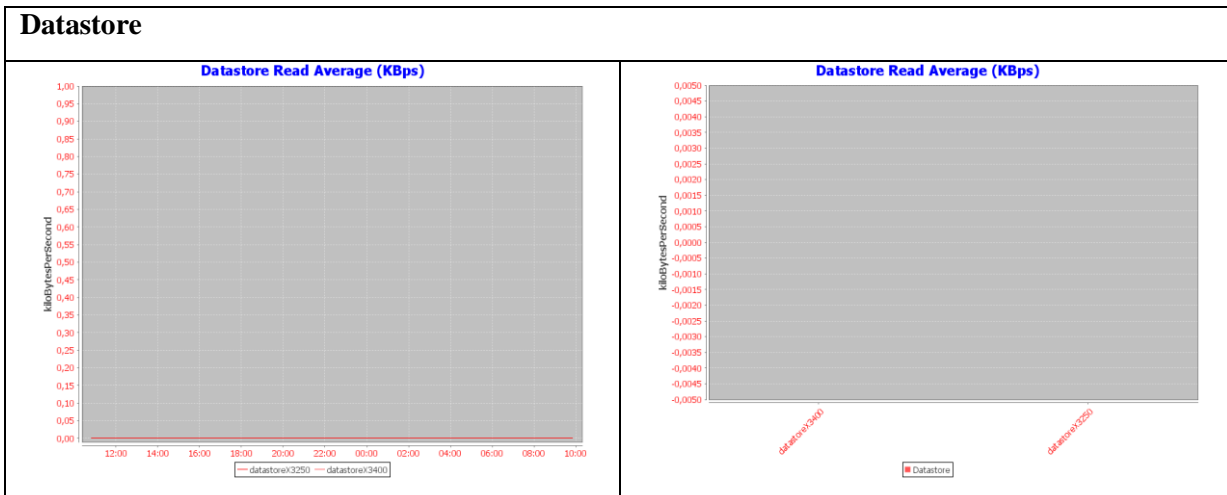
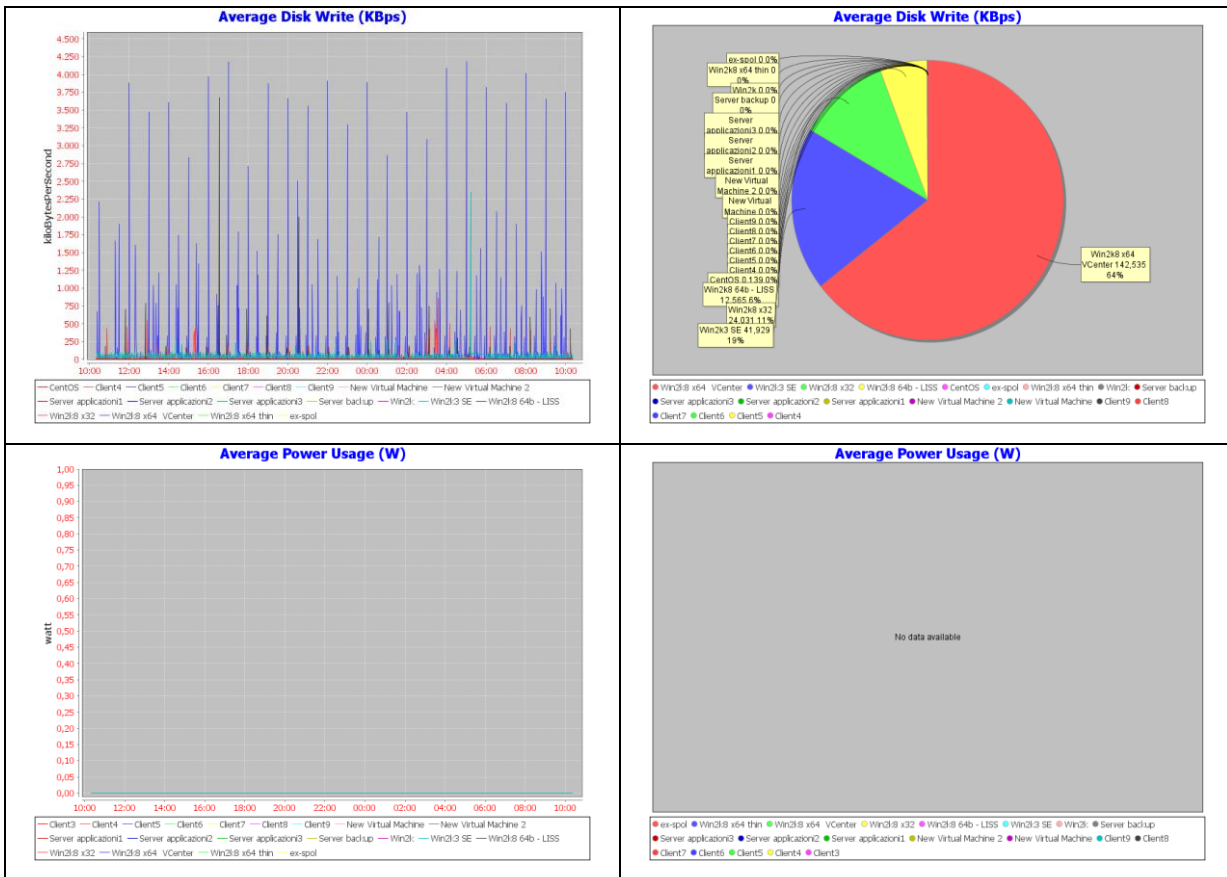
Virtual Machine

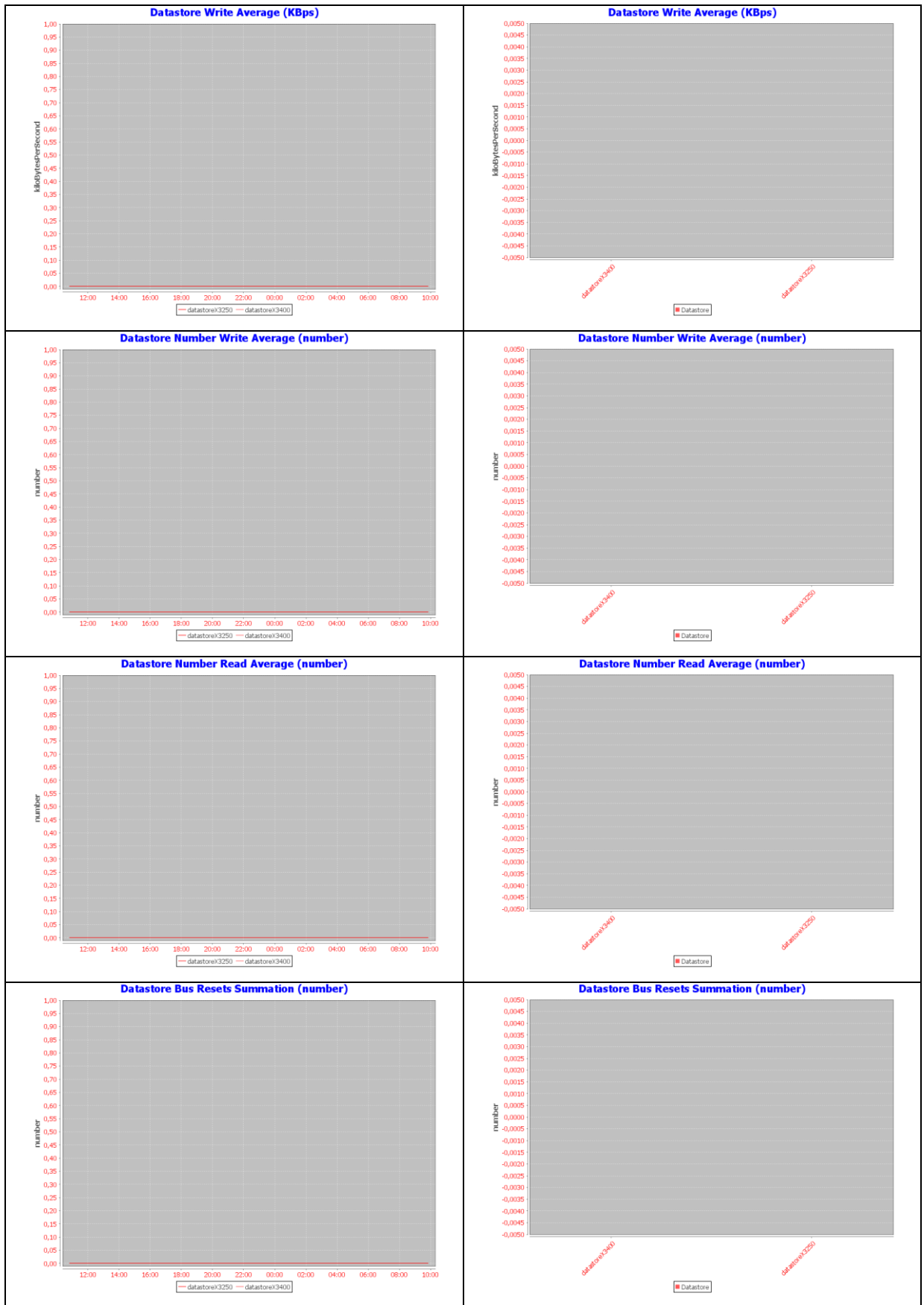




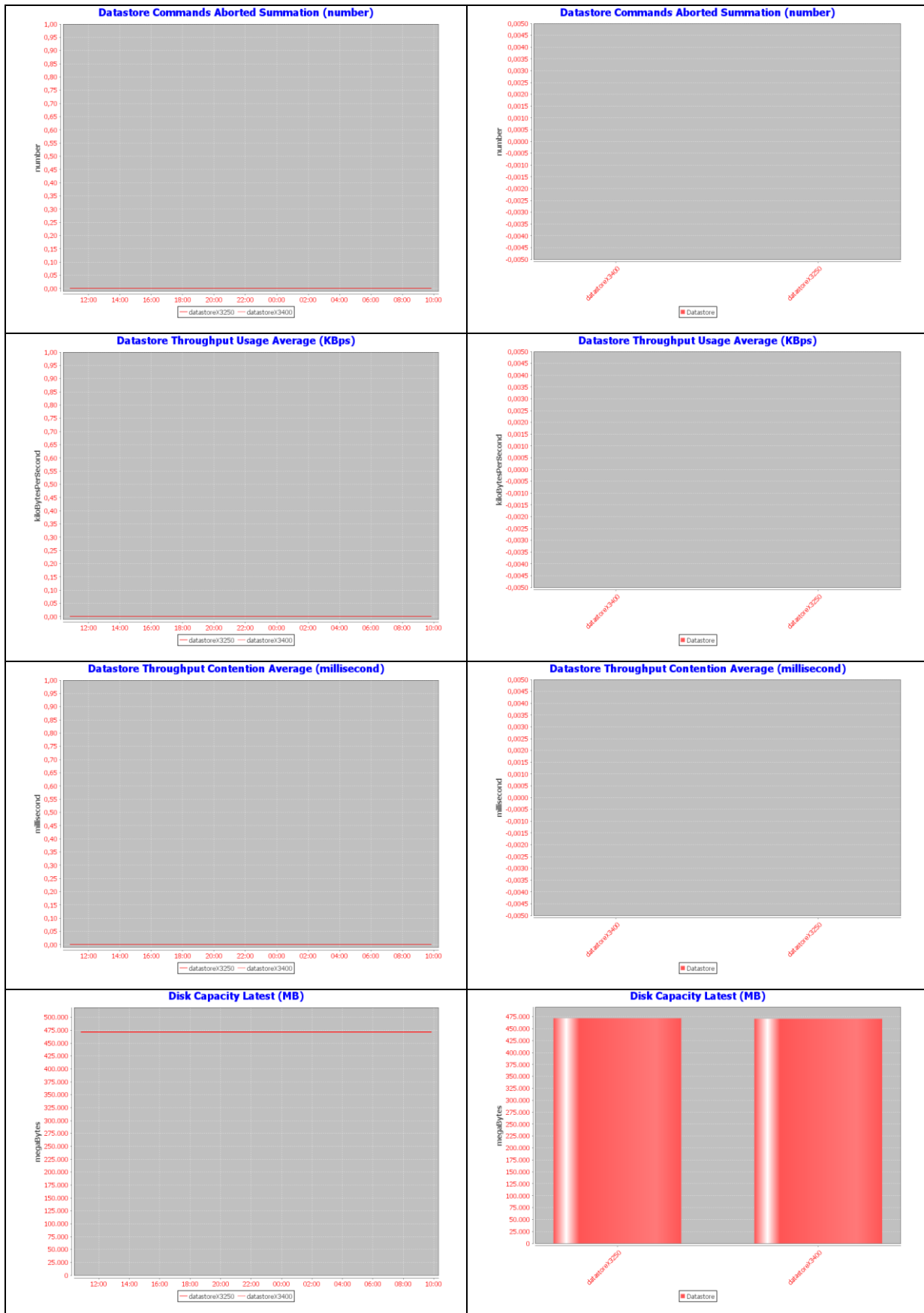


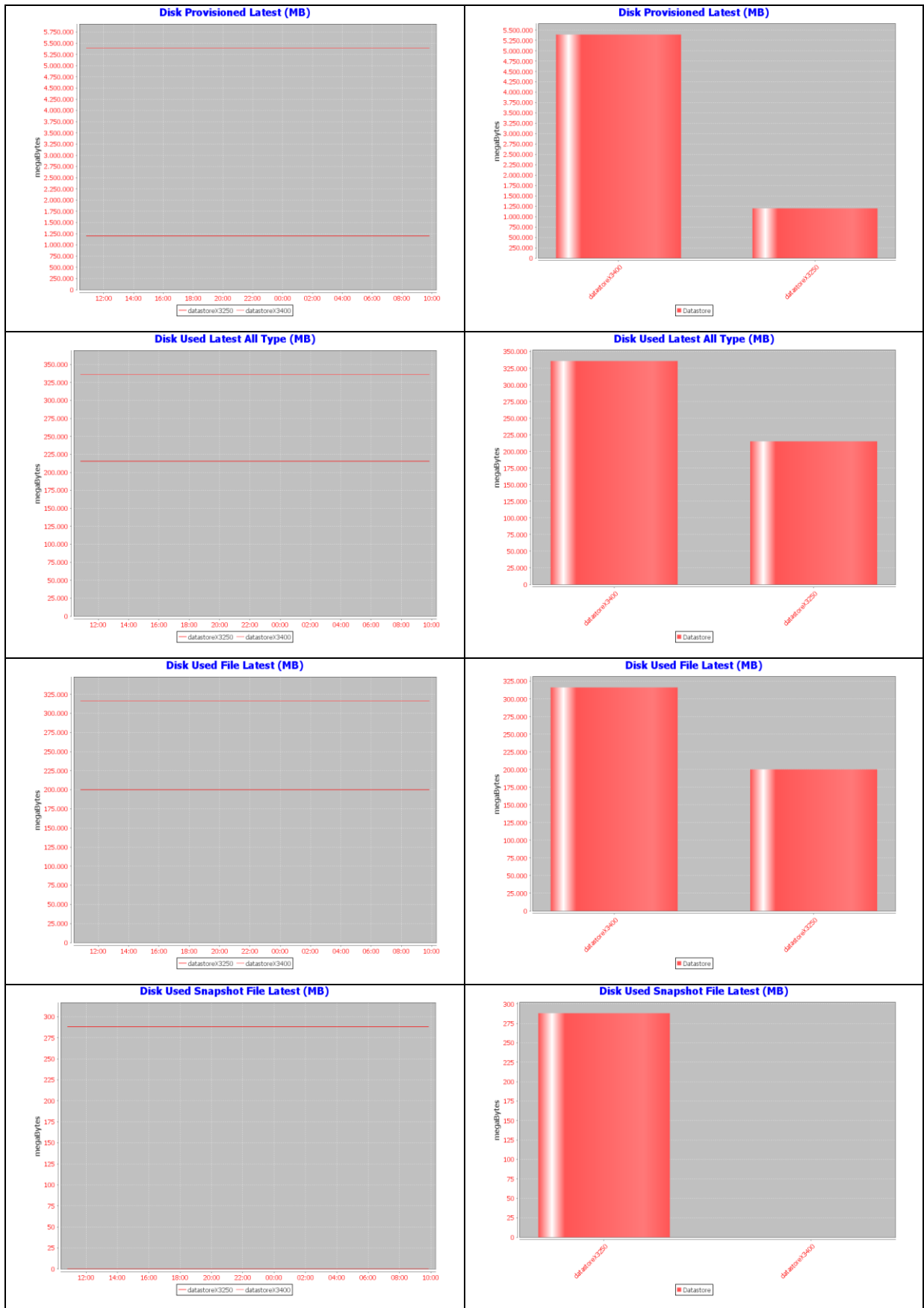
## V. Uso sperimentale - validazione



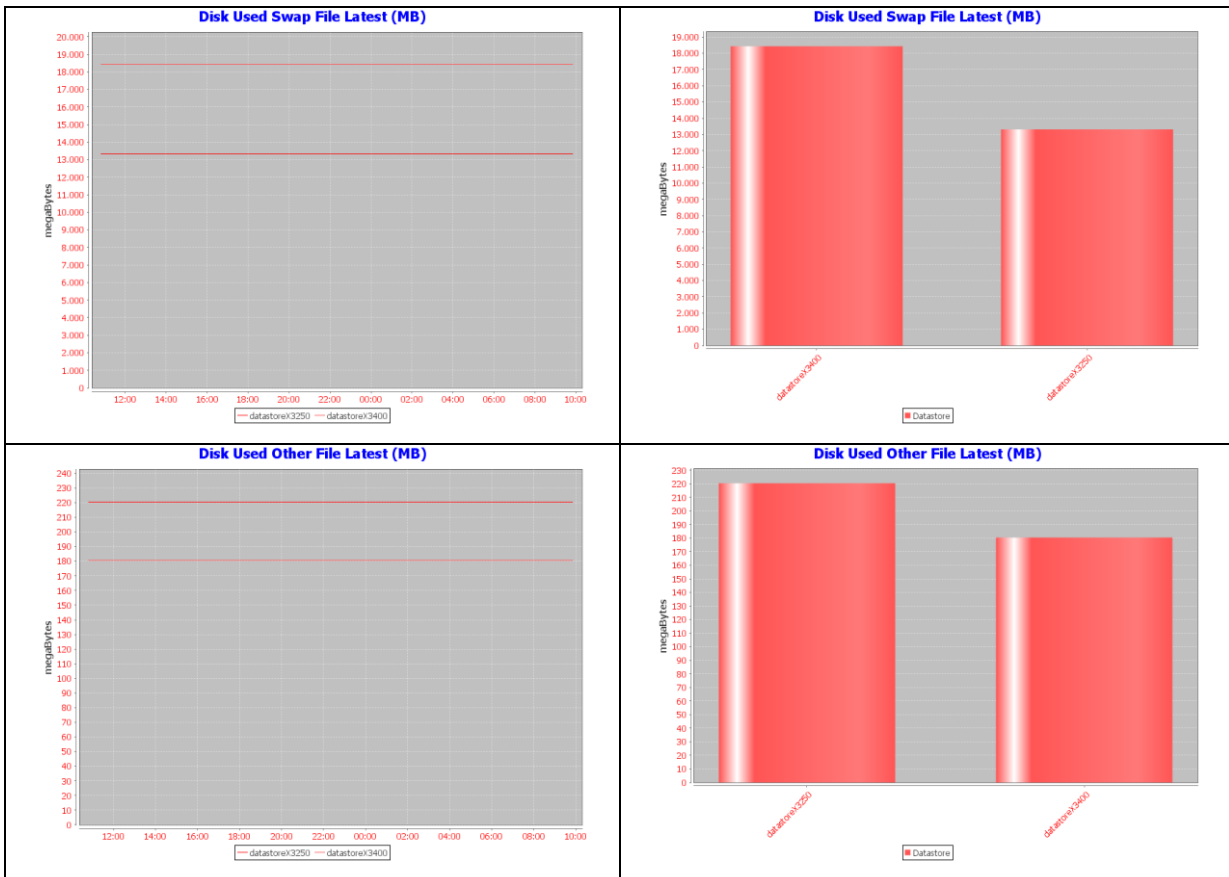












## V.4.2. Datastore

Nell'infrastruttura sono presenti due Datastore, sui quali sono rilevabili proprietà e metriche e statistiche di performance anche per gli elementi Virtual Machine ad essi associati.

### V.4.2.1. Proprietà

Per gli elementi di tipo *Datastore* sono rilevabili 19 proprietà. Un esempio di proprietà rilevate per un elemento di tipo datastore sono mostrate nella seguente tabella:

<b>Datastore - datastoreX3250</b>	
<b>GENERAL INFO</b>	
Name	datastoreX3250
Alarm Actions Enabled	true
Config Status	GRAY
Overall Alarm Status	GREEN
Accessible	true
Capacity (bytes)	4,94727E+11
Free Space (bytes)	2,68978E+11
Maintainance Mode	normal
Type of File System Volume	VMFS
Uncommitted (bytes)	1,0318E+12

Url	ds:///vmfs/volumes/4ffb2fc5-d0418c93-fab4-00215e67bfd0/
Max File Size (bytes)	2,19902E+12
Timestamp	2013-10-23T06:53:07.34175Z
Directory Hierarchy Supported	true
Per File Thin Provisioning Supported	true
Raw Disk Mappings Supported	true
Storage IORM Supported	true
Hosts Number	1
VMs Number	8

**Tabella V-3: Proprietà rilevate per un elemento Datastore del Cloud**

#### V.4.2.2. Metriche di performance

Per gli elementi di tipo Datastore sono disponibili metriche di performance e metriche aggregate degli elementi Virtual Machine associati. Di seguito un esempio di statistiche visualizzate dal sistema per un elemento Datastore del Cloud con i seguenti parametri:

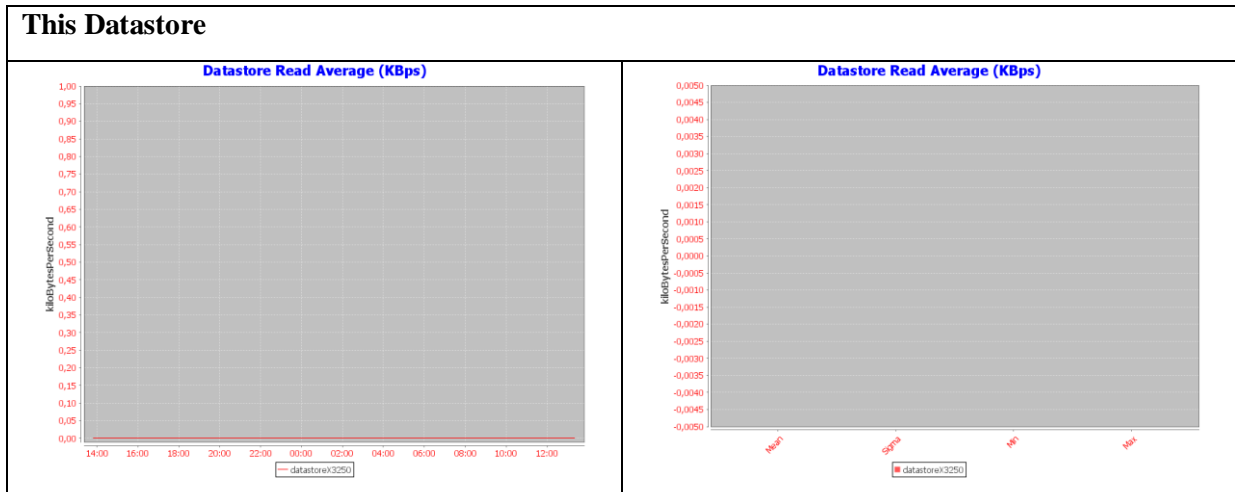
*Datastore: datastoreX3250*

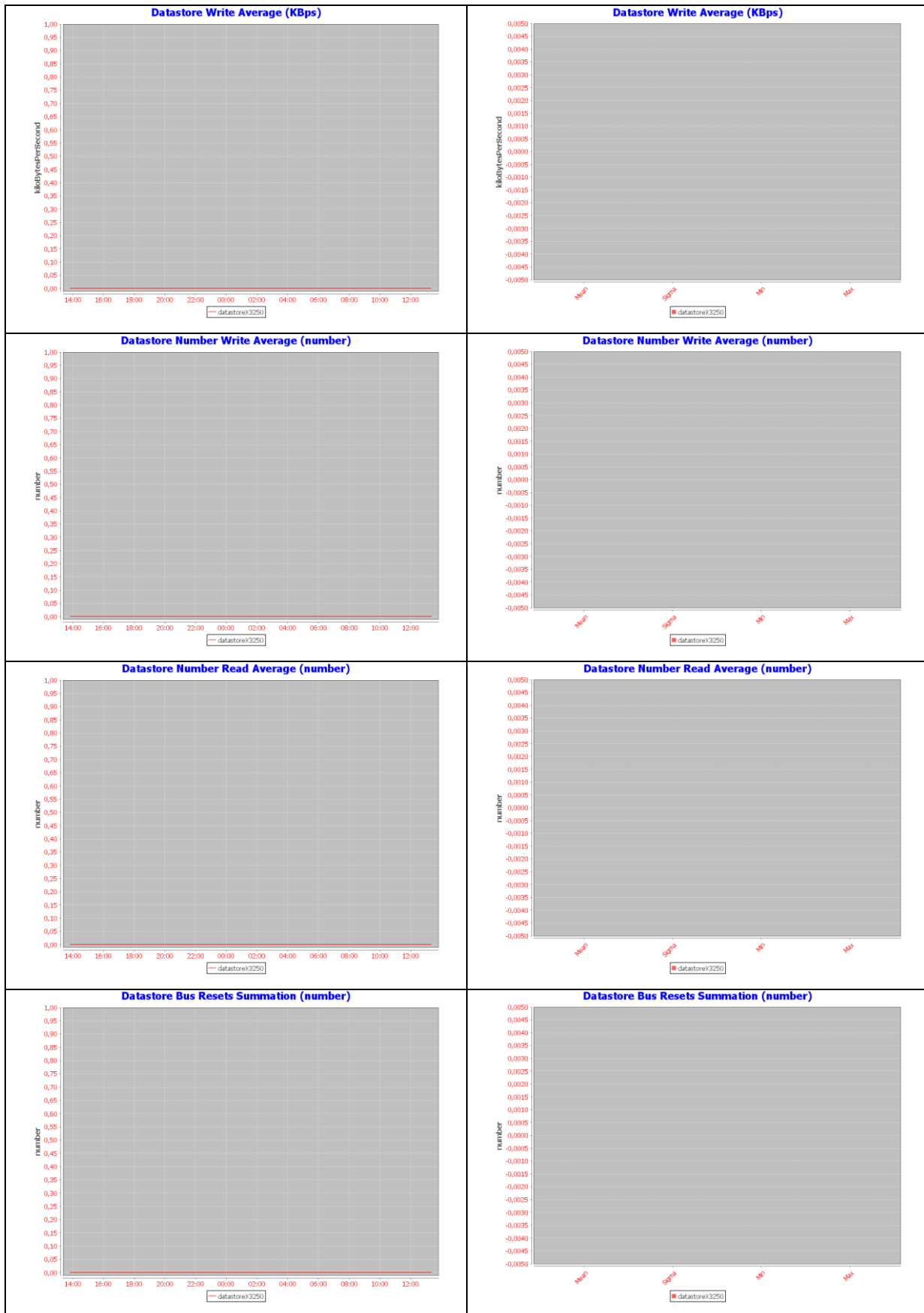
*Search parameter*

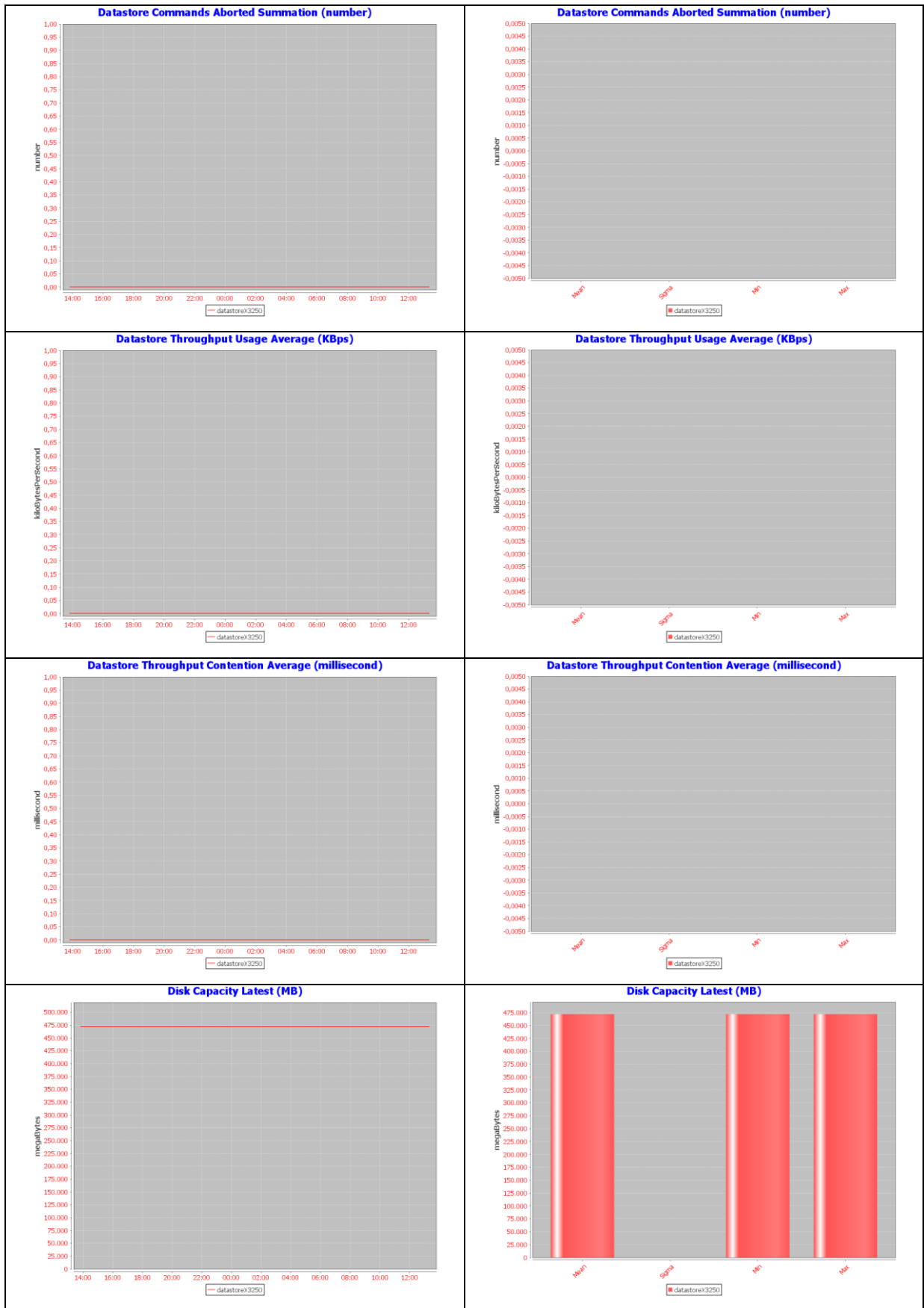
*Minute: one day*

*Sorting type: descendant*

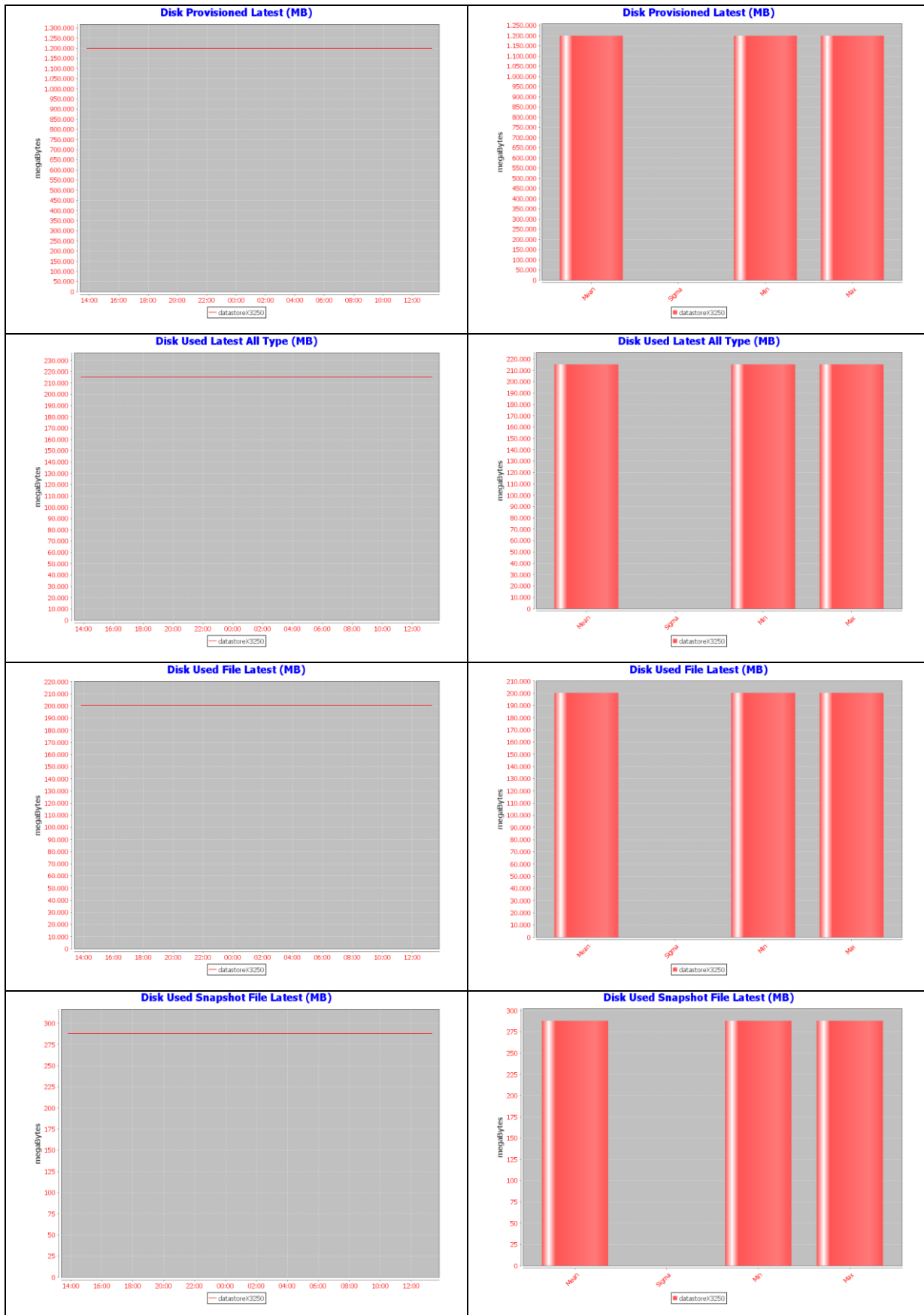
*Chart value limit: 20*

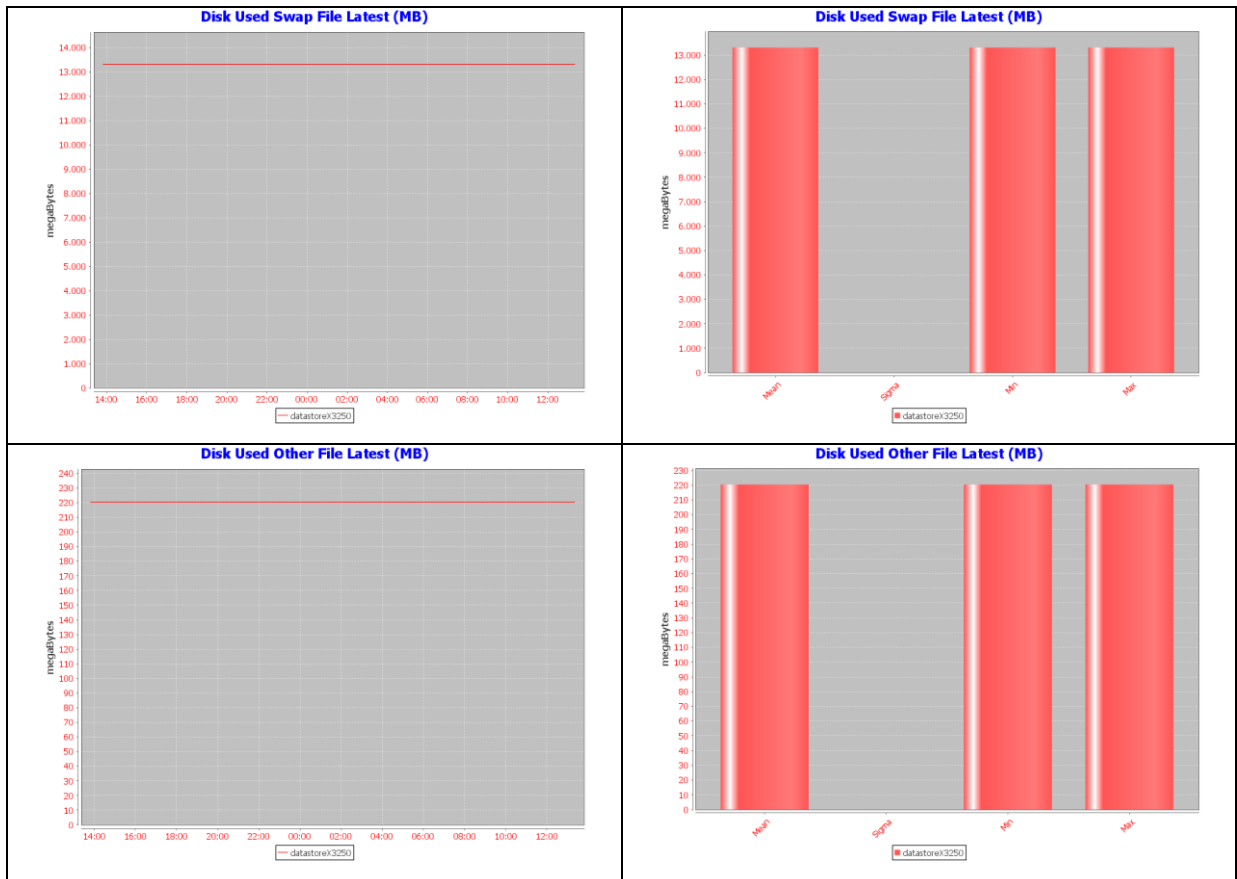




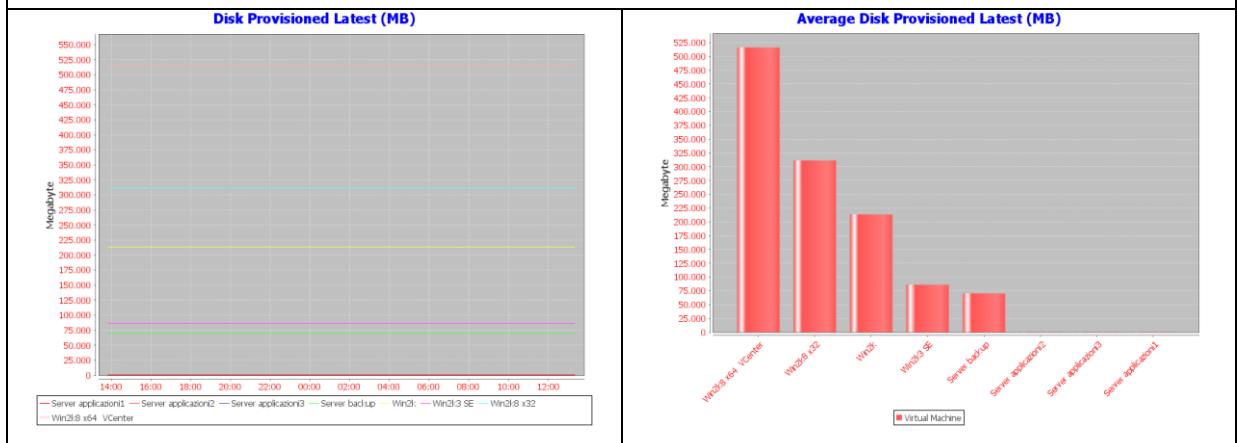


## V. Uso sperimentale - validazione

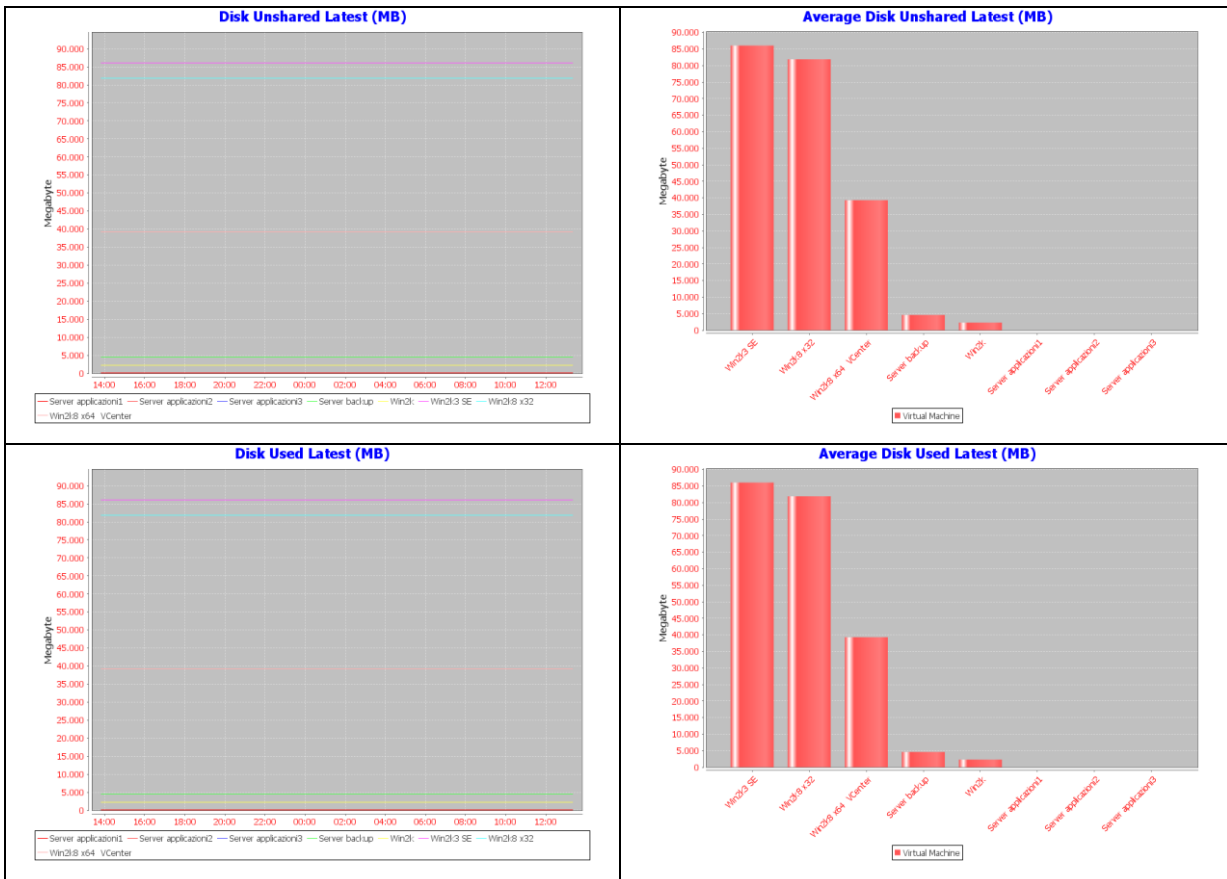




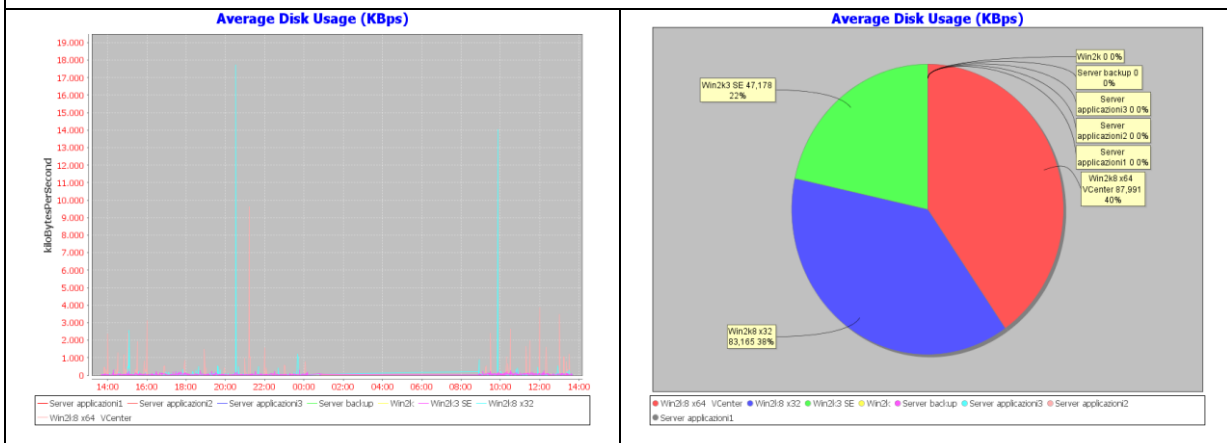
### Datastore Storage by Virtual Machine

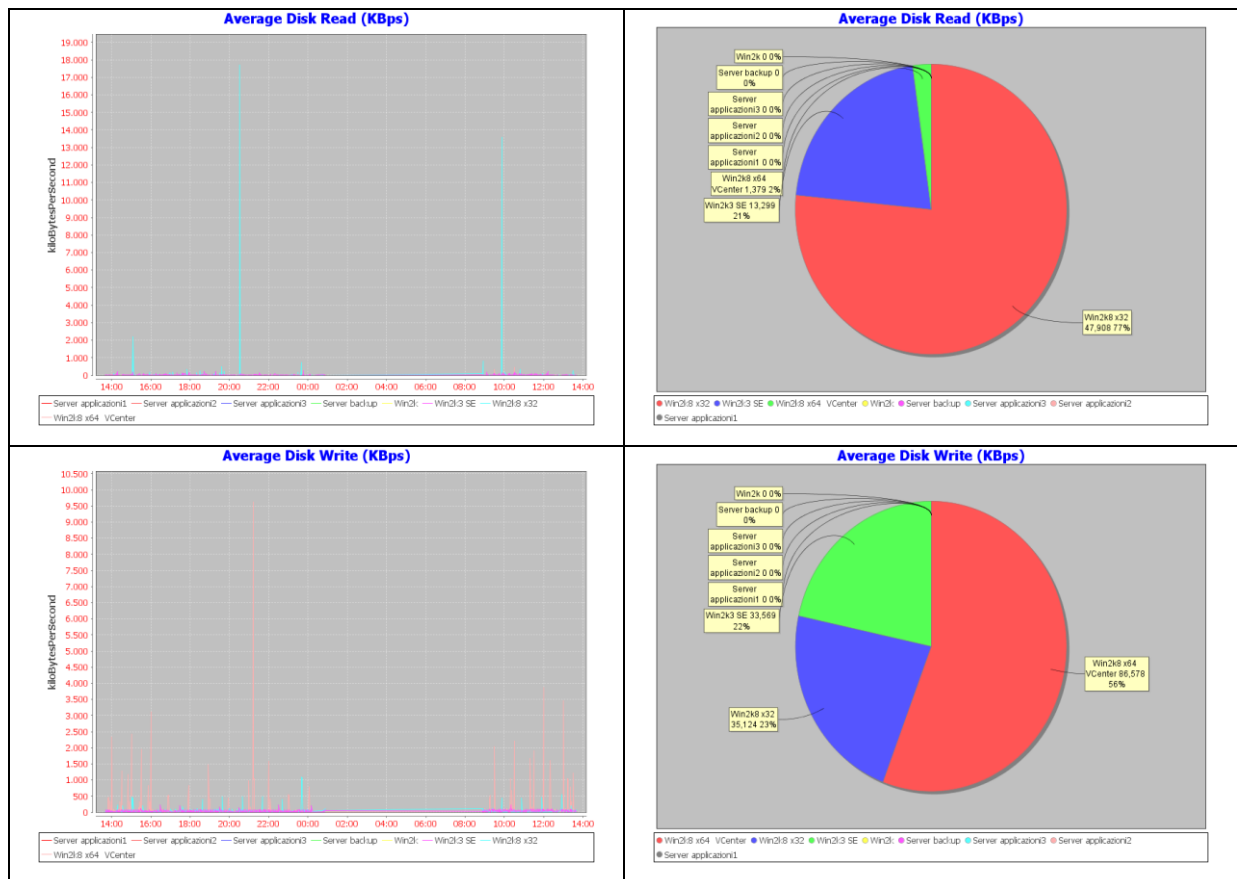


## V. Uso sperimentale - validazione



## Disk usage by Virtual Machine





### V.4.3. Cluster

Nell'infrastruttura è presente un Cluster, sul quale sono rilevabili proprietà e metriche e statistiche di performance, anche per gli elementi Host system e Virtual Machine ad esso associati.

#### V.4.3.1. Proprietà

Per gli elementi di tipo Cluster sono rilevabili 17 proprietà. Un esempio di proprietà rilevate per un elemento di tipo cluster sono mostrate nella seguente tabella:

Cluster - Economia Cluster	
<b>GENERAL INFO</b>	
Name	Economia Cluster
Alarm Actions Enabled	true
Config Status	GREEN
Overall Alarm Status	GREEN
Effective Cpu Resources (MHz)	0
Effective Memory (MB)	25264
Cpu Cores Number	8
Cpu Threads Number	16
Effective Hosts Number	2
Hosts Number	2
Total Cpu Resources (MHz)	19704
Total Memory Resources (bytes)	32154537984



SPBM Enabled	
VM Swap Placement	vmDirectory
Datastores Number	2
Networks Number	2
Hosts Number	2

**Tabella V-4: Proprietà rilevate per un elemento Cluster del Cloud**

**V.4.3.2. Metriche di performance**

Per gli elementi di tipo Cluster sono disponibili metriche di performance e metriche aggregate degli elementi associati quali Host System e Virtual. Di seguito un esempio di statistiche visualizzate dal sistema per un elemento Cluster del Cloud con i seguenti parametri:

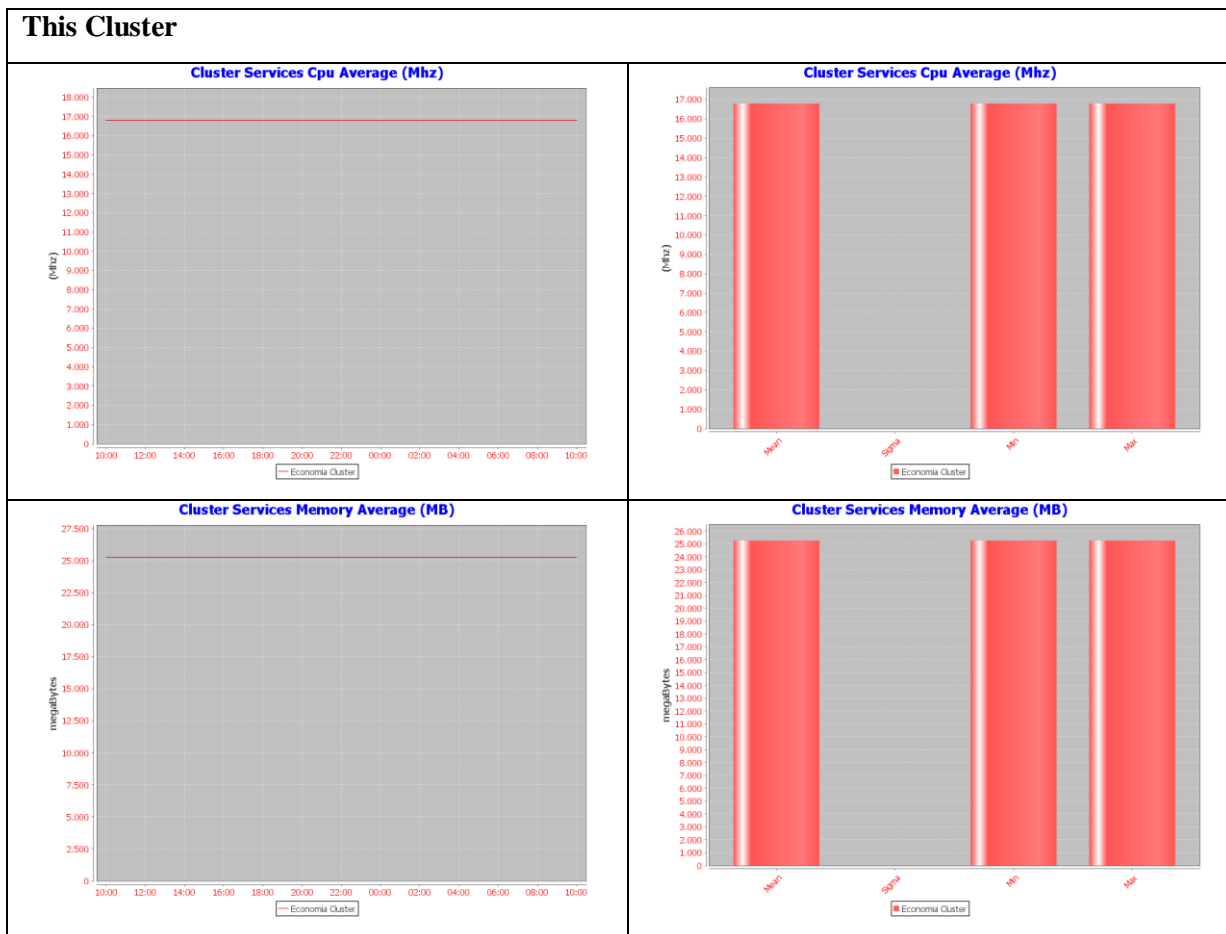
*Cluster: Economia Cluster*

*Search parameter*

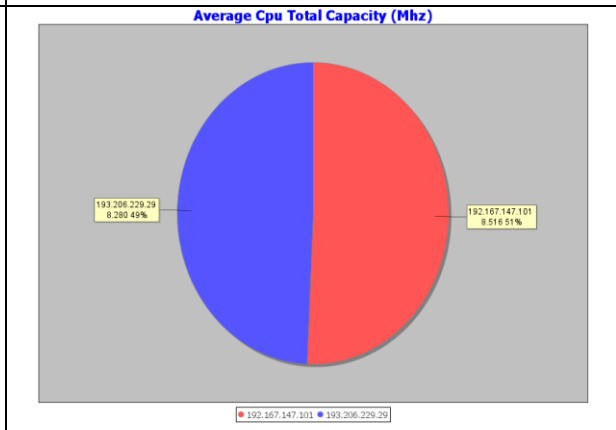
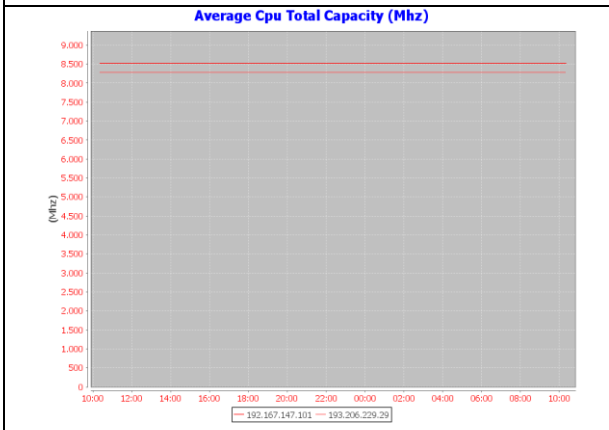
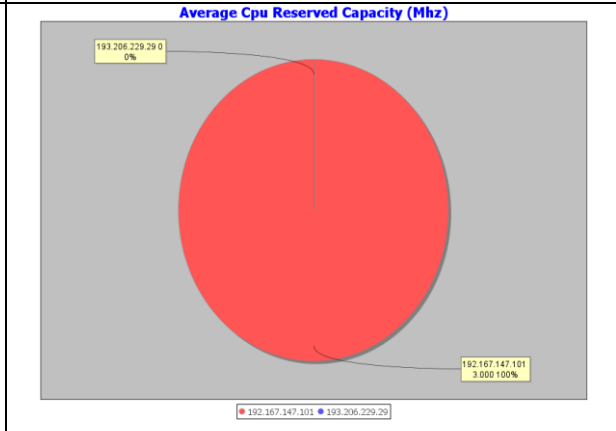
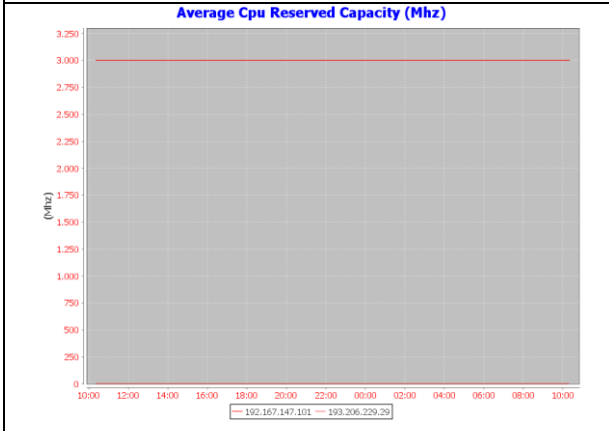
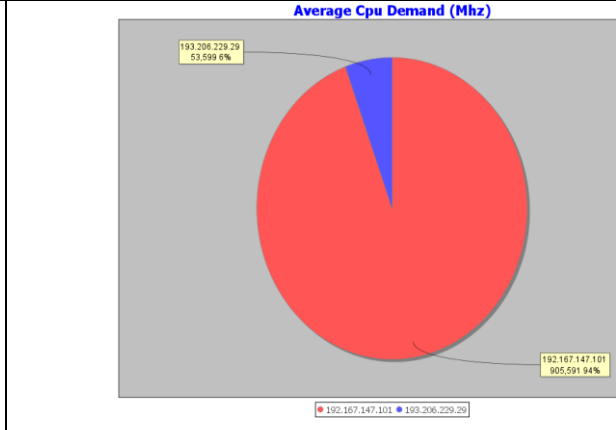
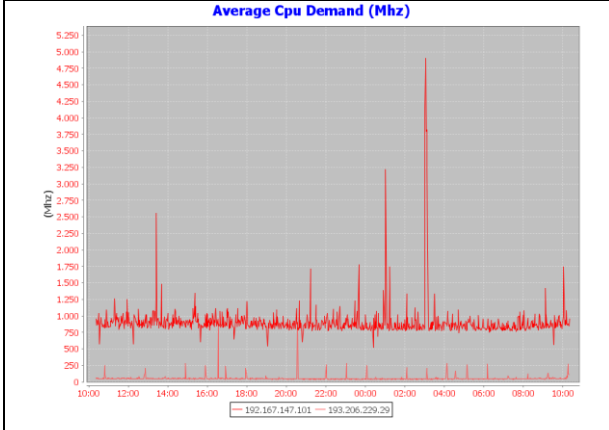
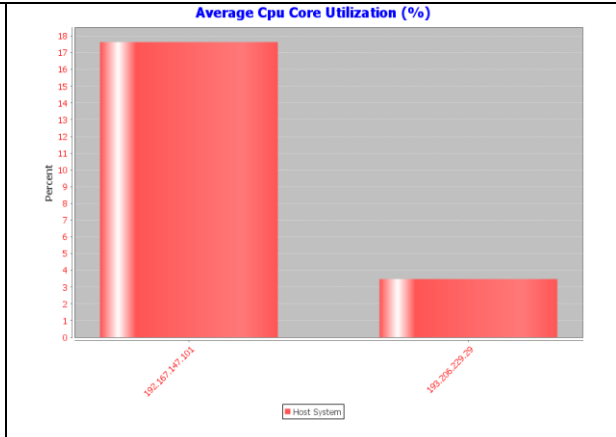
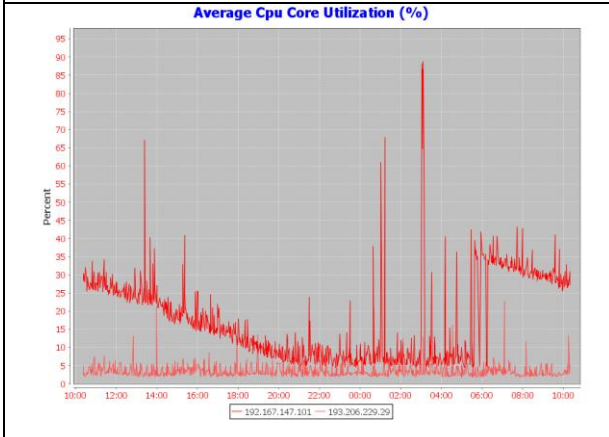
*Minute: one hour*

*Sorting type: descendant*

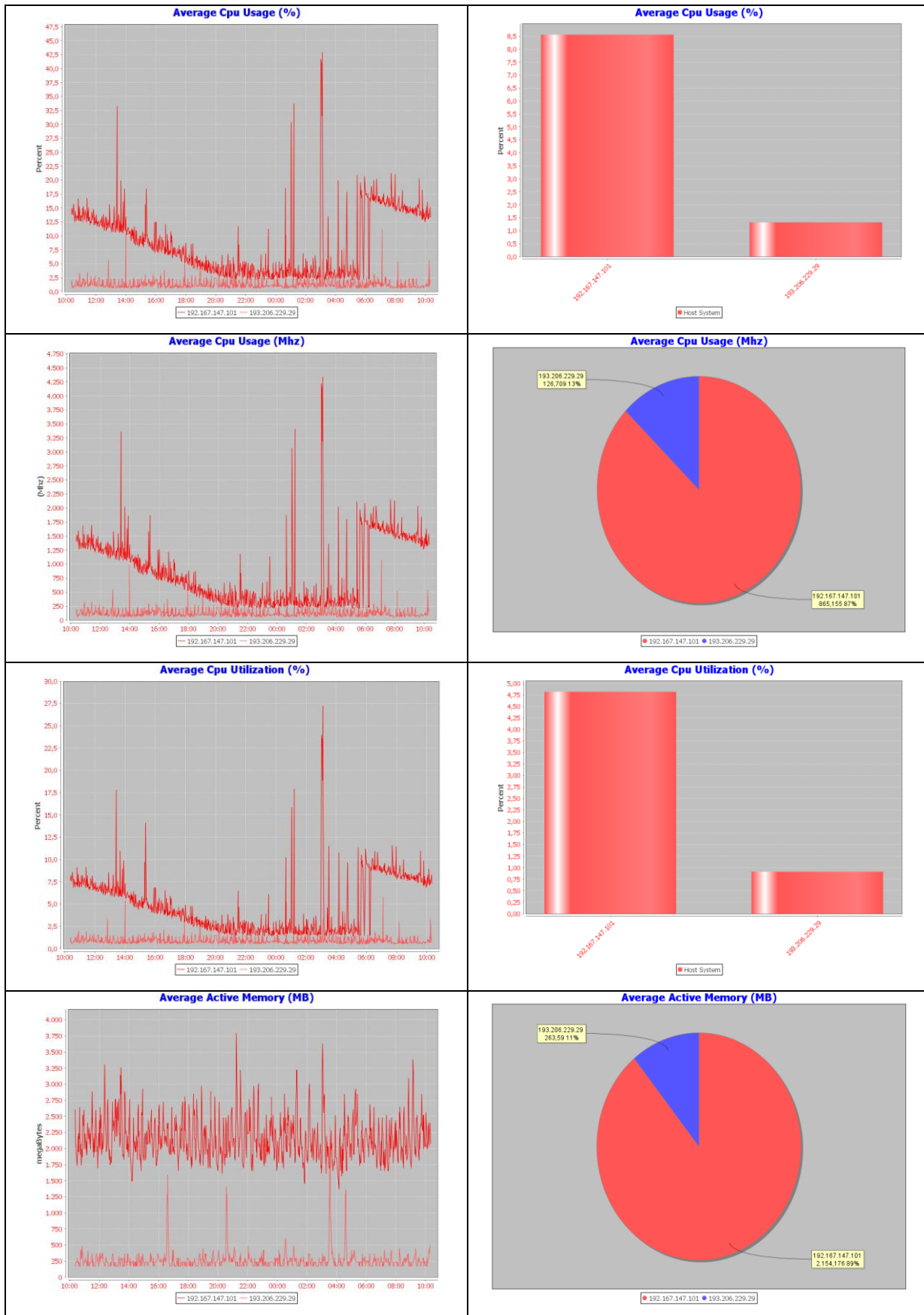
*Chart value limit: 20*

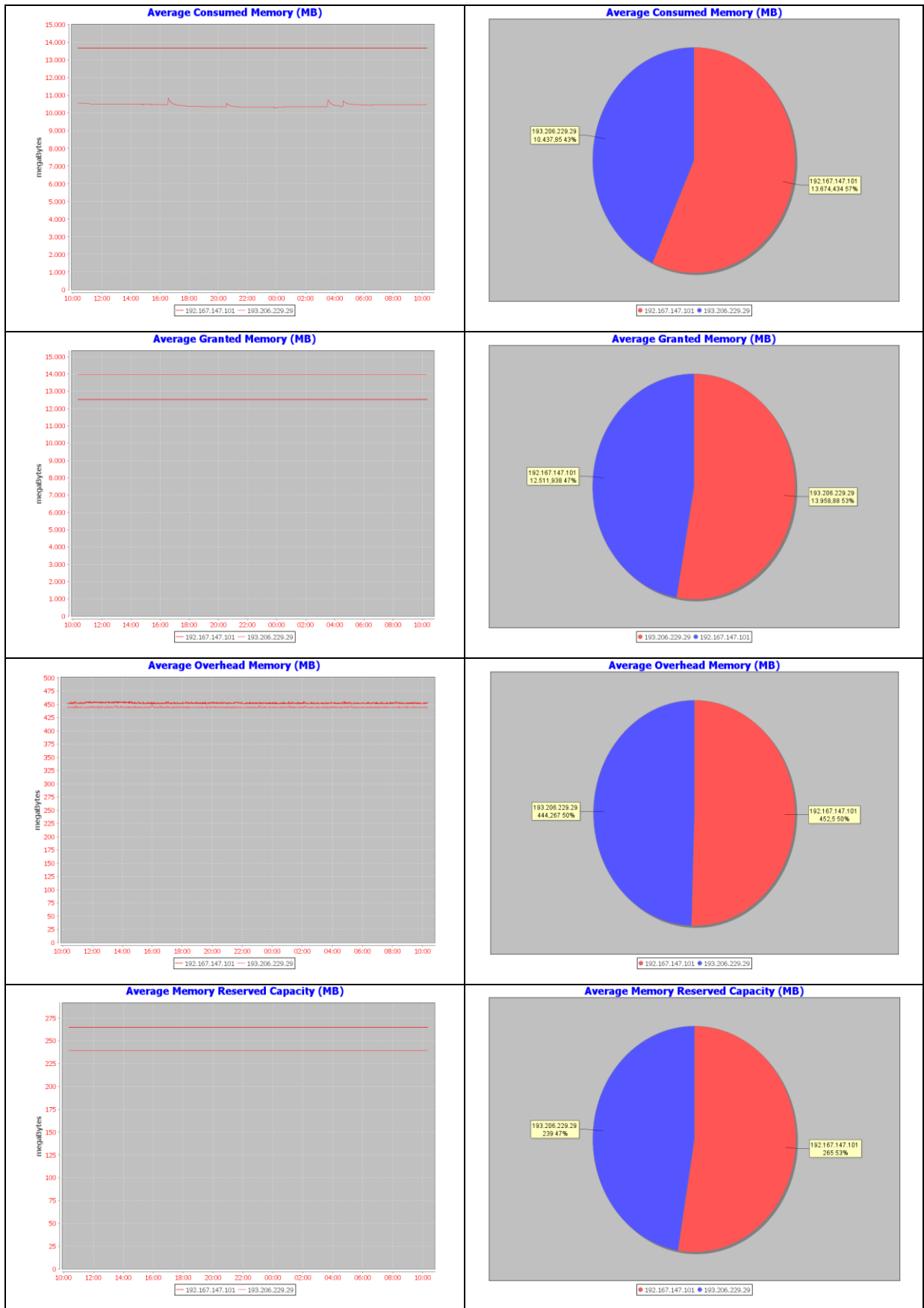


Host System

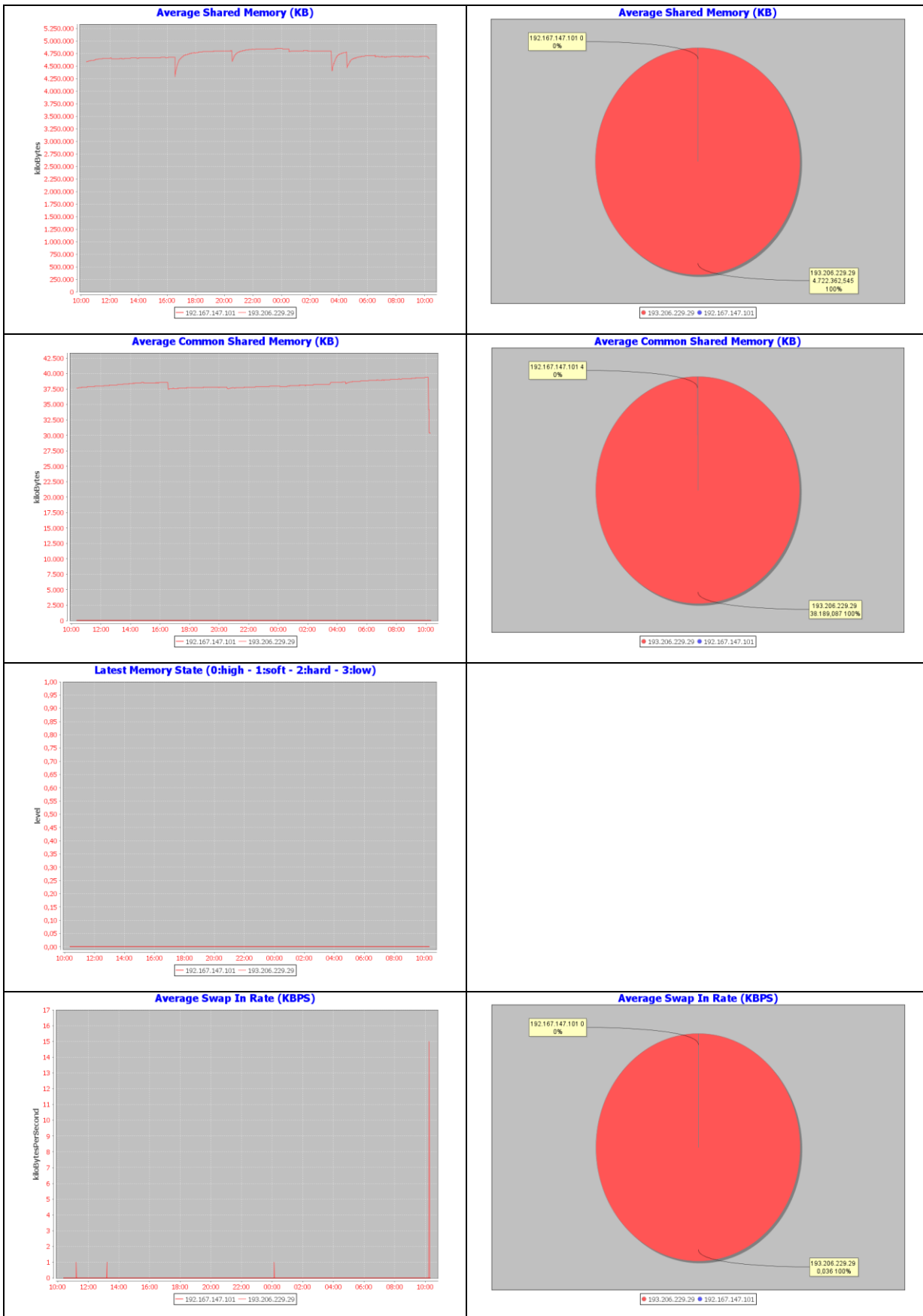


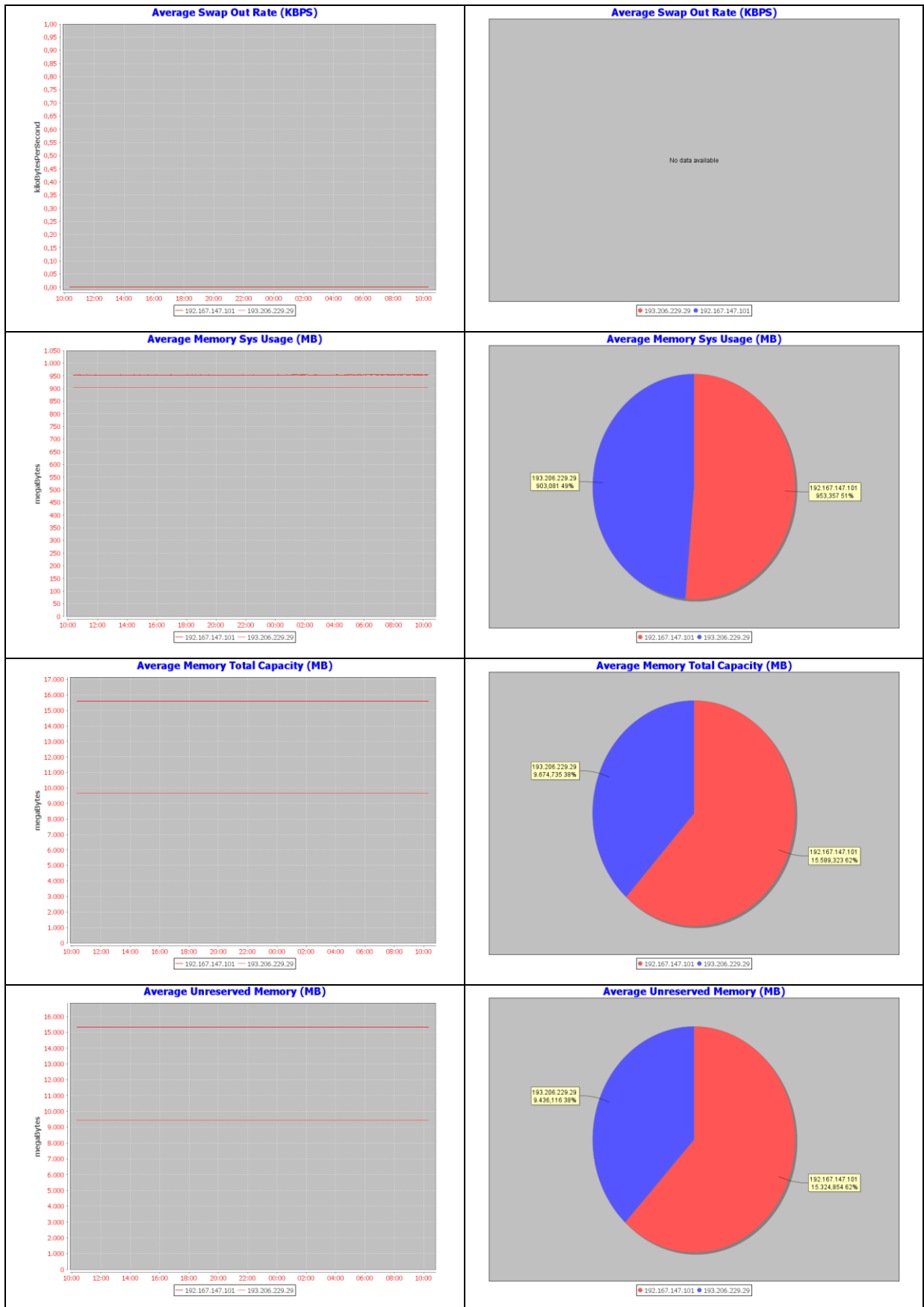
## V. Uso sperimentale - validazione



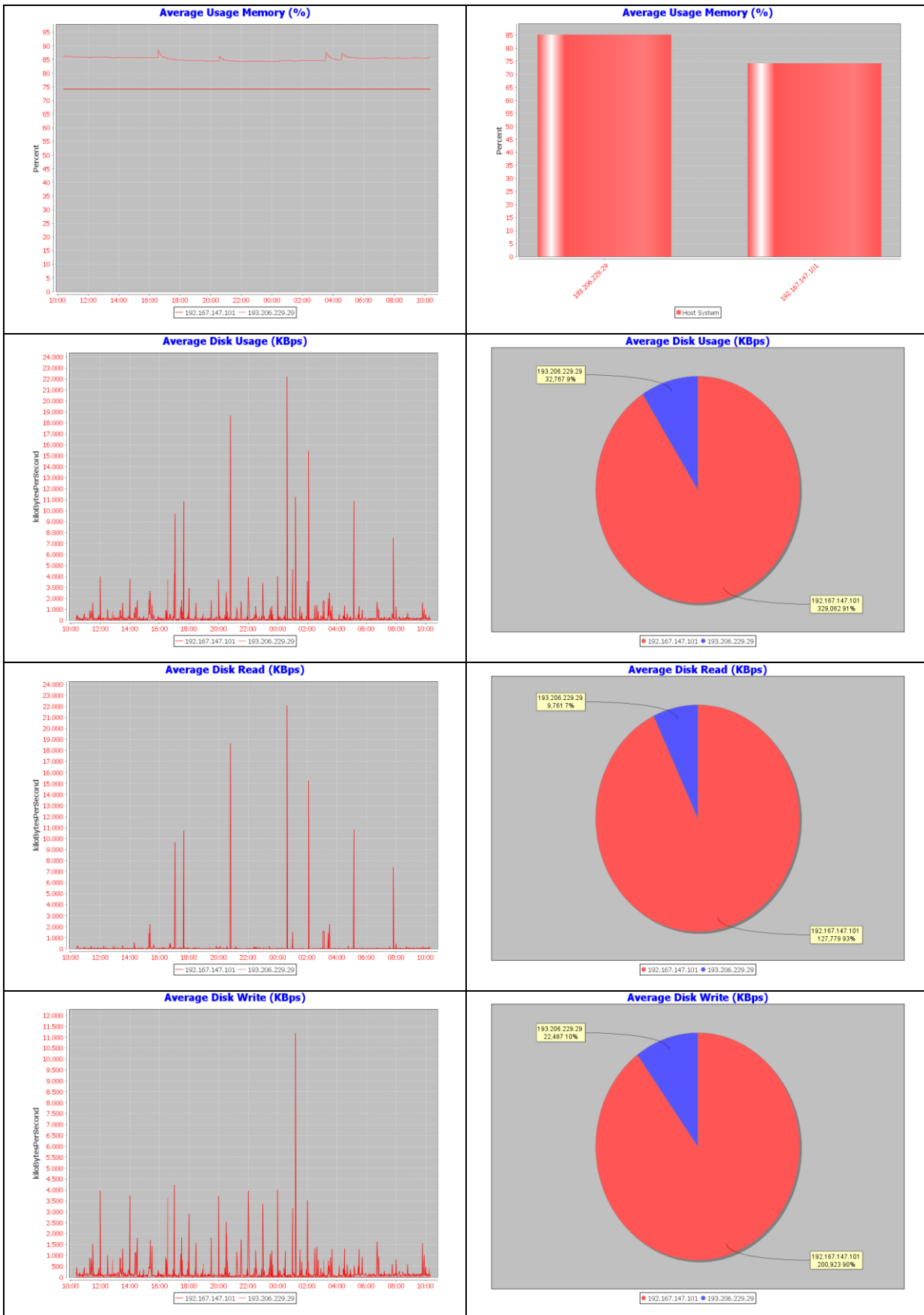


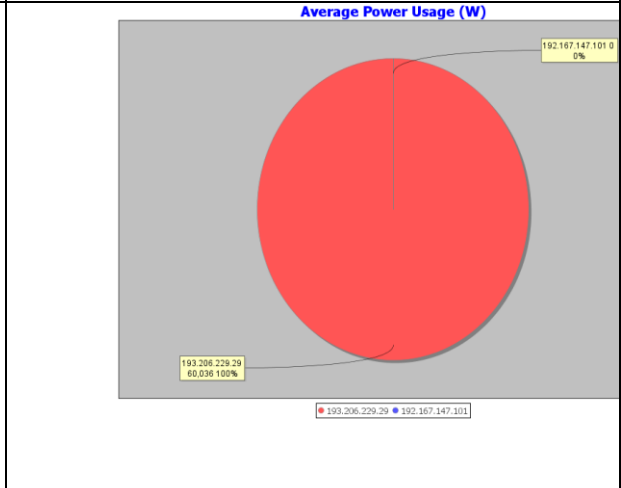
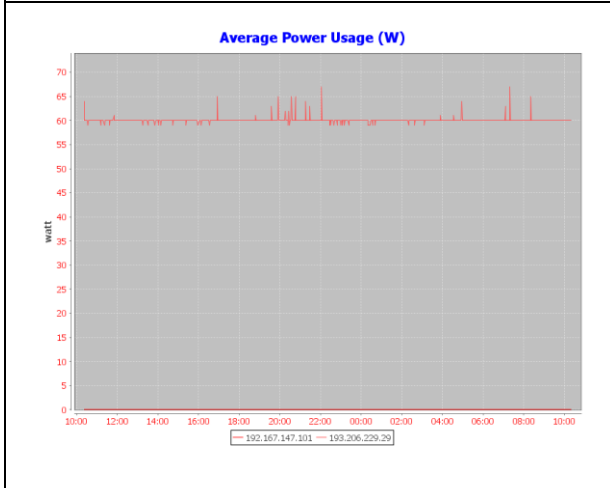
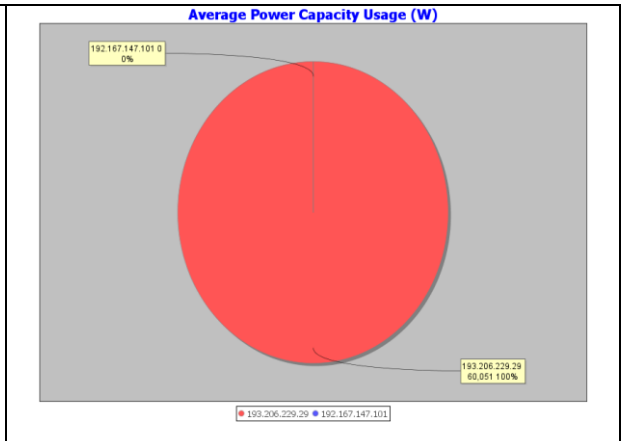
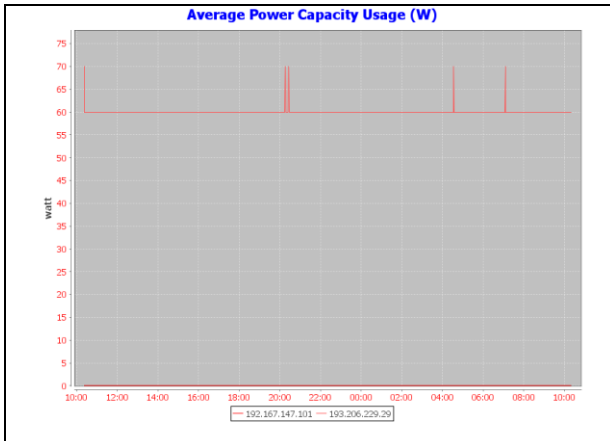
## V. Uso sperimentale - validazione



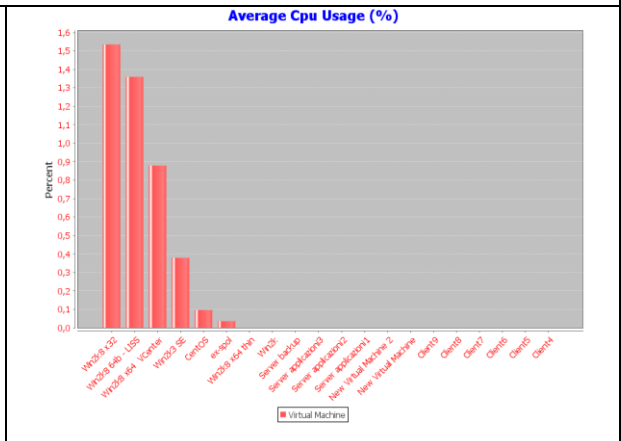
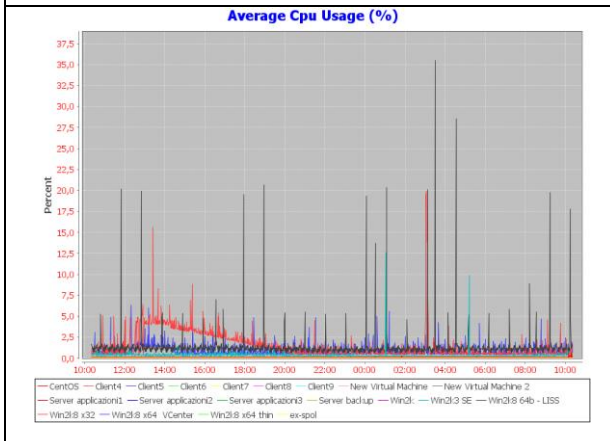


## V. Uso sperimentale - validazione

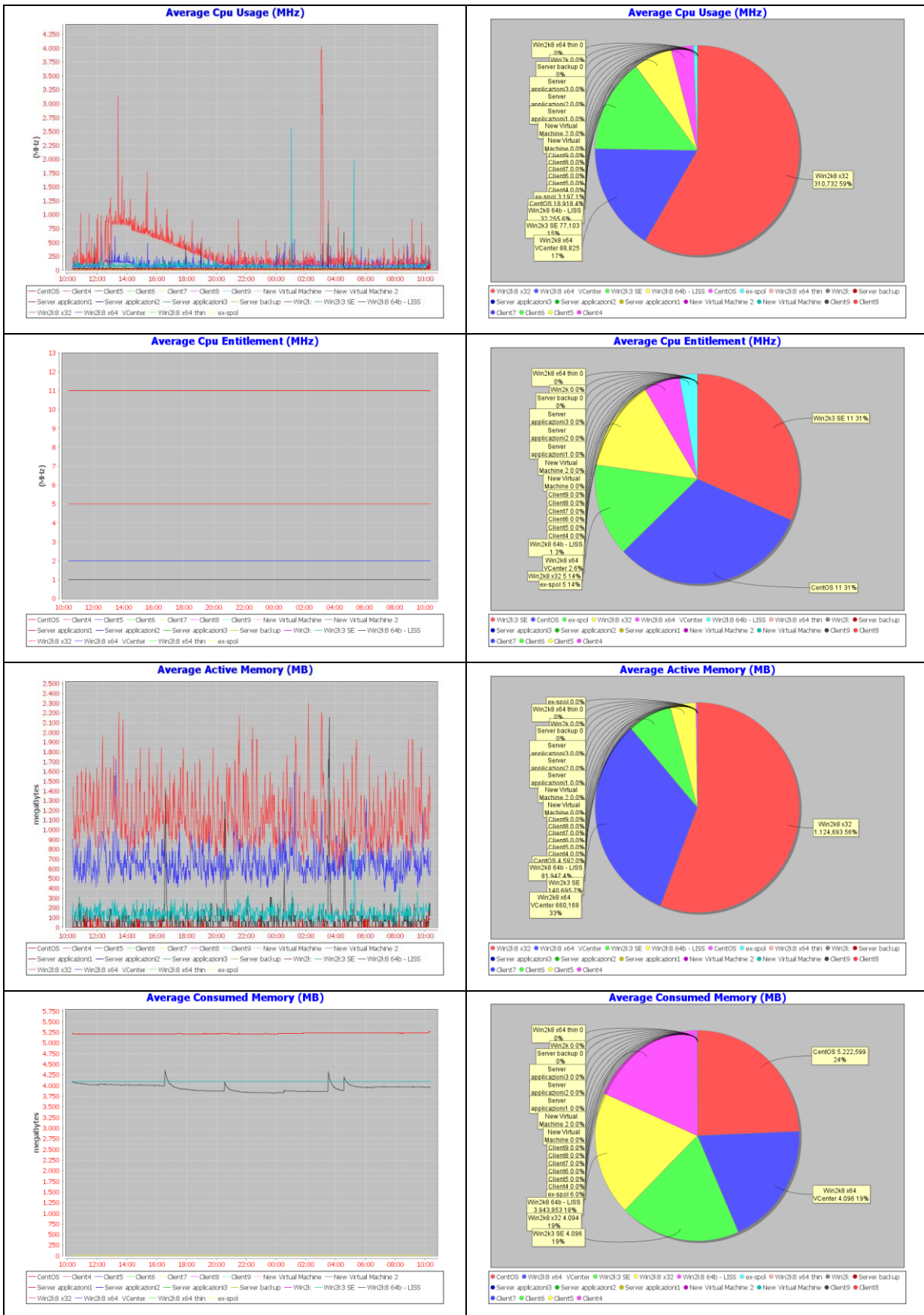


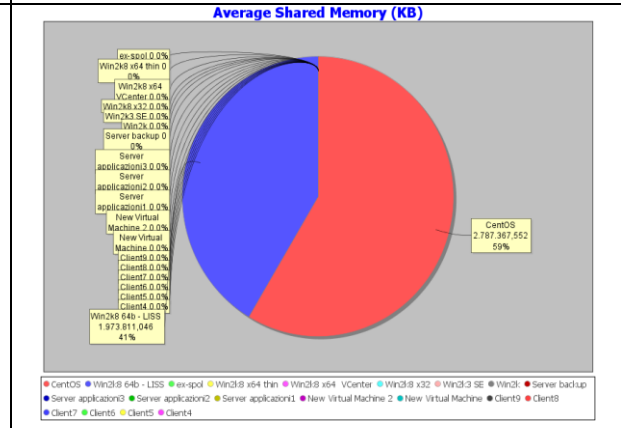
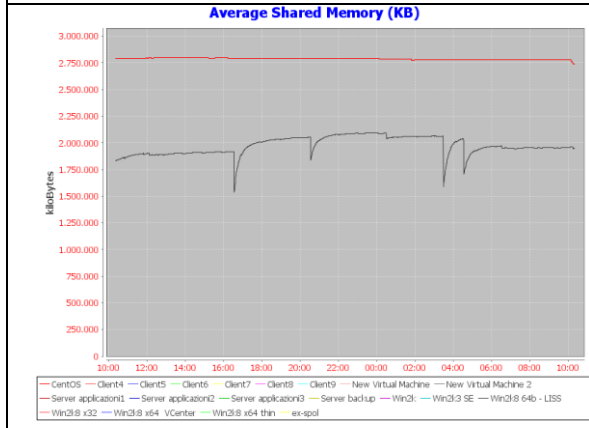
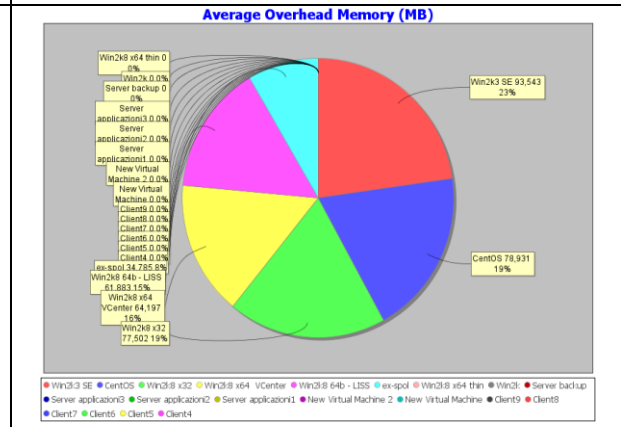
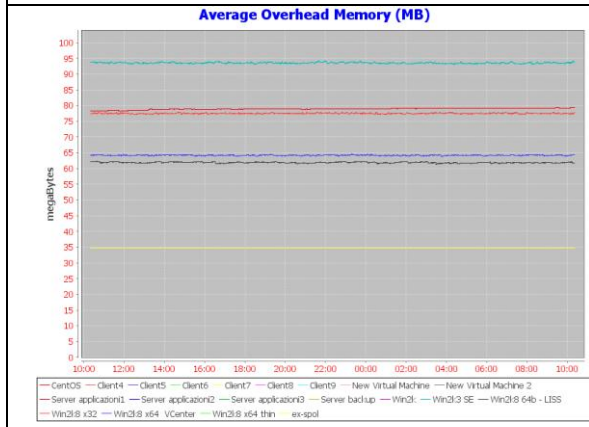
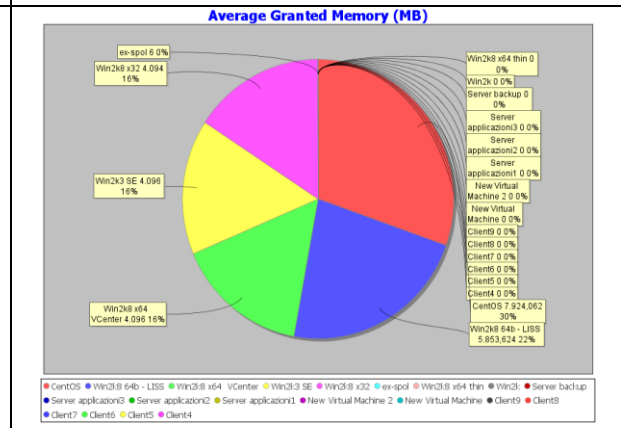
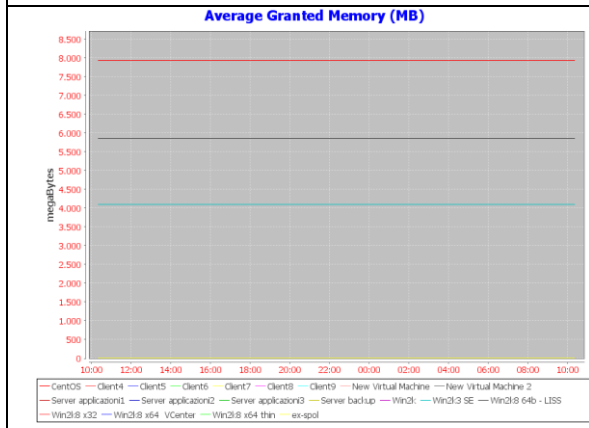
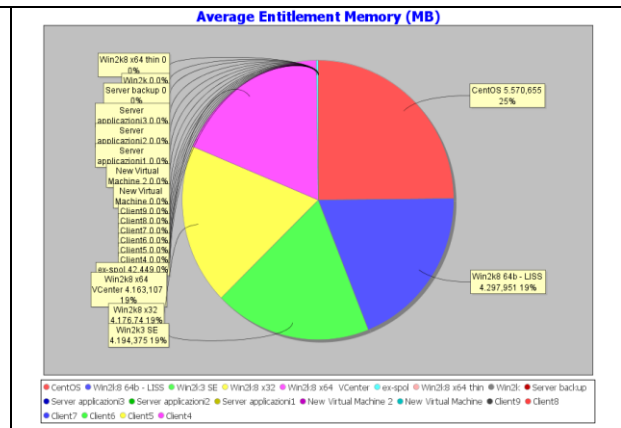
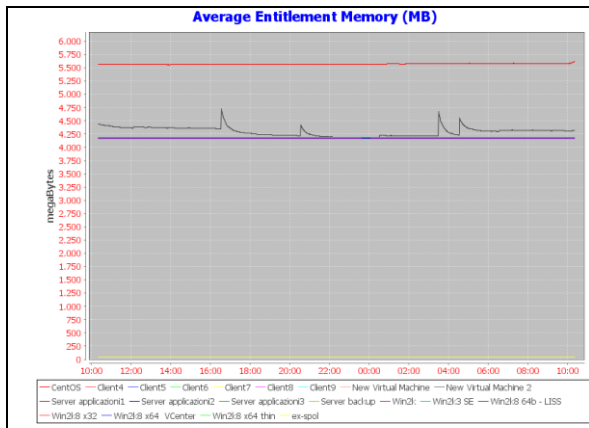


Virtual Machine



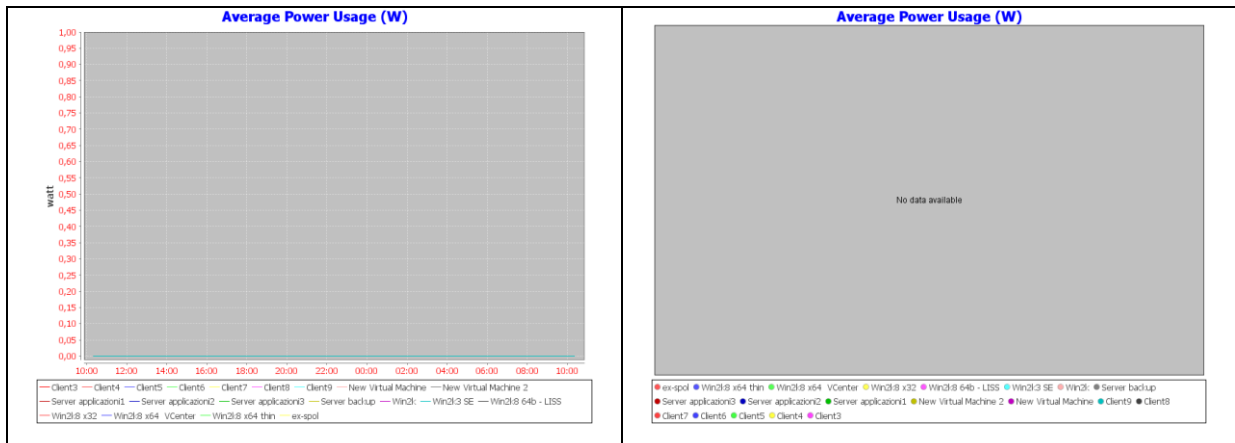






## V. Uso sperimentale - validazione





### V.4.4. Host System

Nell'infrastruttura sono presenti due Host System, sui quali sono rilevabili proprietà e metriche e statistiche di performance anche per gli elementi Virtual Machine ad essi associati.

#### V.4.4.1. Proprietà

Per gli elementi di tipo Host System sono rilevabili 163 proprietà. Un esempio di proprietà rilevate per un elemento di tipo host system sono mostrate nella seguente tabella:

<b>Host System - 192.167.147.101</b>	
<b>GENERAL INFO</b>	
Name	192.167.147.101
Alarm Actions Enabled	true
Config Status	GREEN
Overall Status	GREEN
<b>CAPABILITY INFO</b>	
Background Snapshots Supported	false
Clone From Snapshot Supported	true
Clone From Snapshot Supported	true
Cpu Memory Resource Configuration Supported	true
Datastore Principal Supported	true
Delta Disk Backings Supported	true
Firewall Ip Rules Supported	true
Ft Compatibility Issues	[ftNotLicensed, haAgentIssue, missingFTLoggingNic, missingVMotionNic, vMotionNotLicensed]
Ft Supported	false
High Guest Memory Supported	true
IPMI Supported	true
iSCSI Supported	true
Local Swap Datastore Supported	true
Login By SSL Thumbprint Supported	true
Maintenance Mode Supported	true

Max Host Running VMs	99
Max Host Supported VCPUs	100
Max Running VMs	0
Max Supported VCPUs	8
Max Supported VMs	1200
NFS Supported	true
NIC Teaming Supported	true
Per VM Network Traffic Shaping Supported	false
Per VM Swap Files	true
Pre Assigned PCI Unit Numbers Supported	true
Reboot Supported	true
Record Replay Supported	true
Recursive Resource Pools Supported	true
Replay Compatibility Issues	[]
Replay Unsupported Reason	
Restricted Snapshot Relocate Supported	true
San Supported	true
Scaled Screenshot Supported	true
Screenshot Supported	true
Service Package Info Supported	true
Shutdown Supported	true
Snapshot Relayout Supported	true
Standby Supported	true
Storage IORM Supported	false
Storage VMotion Supported	true
Supported VMFS Major Version	[3, 5]
Suspended Relocate Supported	true
TPM Supported	false
Unshared Swap VMotion Supported	true
Virtual Exec Usage Supported	true
VLAN Tagging Supported	true
VM Direct Path Gen 2 Supported	true
VM Direct Path Gen 2 Unsupported Reason	[]
VM Direct Path Gen 2 Unsupported Reason Extended	
VMFS Datastore Mount Capable	true
Vmotion Supported	false
Vmotion With Storage VMotion Supported	false
VStorage Capable	true
<b>CONFIG INFO</b>	
Active Diagnostic Partition Diagnostic Type	singleHost
Active Diagnostic Partition Disk Name	naa.600508e000000000a19c5345e3f50202
Active Diagnostic Partition Partition	7
Active Diagnostic Partition Slots	1
Active Diagnostic Partition Storage Type	directAttached

Admin Disabled	false
Authentication Manager Info Auth Config	[com.vmware.vim25.HostLocalAuthenticationInfo@17d3d64, com.vmware.vim25.HostActiveDirectoryInfo@14993e0]
Auto Start Defaults Enabled	
Auto Start Defaults Start Delay (s)	120
Auto Start Defaults Stop Delay (s)	120
Auto Start Defaults Stop Action	PowerOff
Auto Start Defaults Wait For Heartbeat	false
Console Reservation Service Console Reserved (MB)	0
Console Reservation Service Console Reserved Cfg (MB)	0
Console Reservation Unreserved (MB)	0
Datastore Capabilities Local Datastore Supported	false
Datastor Capabilities NFS Mount Creation Required	true
Datastore Capabilities NFS Mount Creation Supported	true
Datastore Capabilities VMFS Extent Expansion Supported	true
Datastore Principal	root
Date Time Info Time Zone Name	UTC
Date Time Info Time Zone Key	UTC
Date Time Info Time Zone GMT Offset	0
Date Time Info NTP Config Server	[ntp1.inrim.it, ntp2.inrim.it]
Firewall Default Policy Incoming Blocked	true
Firewall Default Policy Outgoing Blocked	true
Hyper Thread Active	true
Hyper Thread Available	true
Hyper Thread Config	true
IPMI Bmc Ip Address	
IPMI Bmc Mac Address	
IPMI Login	
IPMI Password	
Multipath State Path	[com.vmware.vim25.HostMultipathStateInfoPath@18b6832, com.vmware.vim25.HostMultipathStateInfoPath@165deb3]
Power System Info Current Policy Description	PowerPolicy.off.description
Power System Info Current Policy Key	0
Power System Info Current Policy Name	PowerPolicy.off.name
Power System Info Current Policy Short Name	off
Product Api Type	HostAgent
Product Api Version	5.0
Product Build	623860

Product Full Name	VMware ESXi 5.0.0 build-623860
Product Instance Uuid	
Product License Product Name	VMware ESX Server
Product License Product Version	5.0
Product Locale Build	0
Product Locale Version	INTL
Product Name	VMware ESXi
Product O.S Type	vmnix-x86
Product Line Id	embeddedEsx
Product Vendor	VMware, Inc.
Product Version	5.0.0
System File	
Virtual Machine Reservation Allocation Policy	
Virtual Machine Reservation Virtual Machine Max (bytes)	0
Virtual Machine Reservation Virtual Machine Min (bytes)	0
Virtual Machine Reservation Virtual Machine Reserved (bytes)	0
Wake On Lan Capable	
<b>HARDWARE INFO</b>	
Bios Version	-[GYE120CUS-1.00]-
Bios Release Date	2009-09-25T00:00:00Z
Cpu Speed Per Core (Hz)	2526984074
Cpu Num Cores	4
Cpu Num Packages	1
Cpu Num Threads	8
Cpu Power Management Current Policy	Not supported
Cpu Power Management Hardware Support	Not available
Memory Size (bytes)	19315245056
Smc Present	false
System Model	System x3250 M3 -[4252EBG]-
System Uuid	1365448e-f5a7-11de-ac67-00215e67bfcf
System Vendor	IBM
<b>RUNTIME INFO</b>	
Boot Time	2013-10-20T19:46:59.595194Z
Connection State	CONNECTED
Das Host State	
Maintenance Mode	false
Power State	POWERED_ON
Standby Mode	none
<b>SUMMARY CONFIG INFO</b>	
Config Fault Tolerance Enabled	false
Config Port	443

Config SSL Thumbprint	69:50:2C:33:D7:30:20:0D:B4:5F:64:CE:2D:CD:8E:5E:8C:1D:CF:90
Config Vmotion Enabled	false
Current EVC Mode Key	
<b>SUMMARY HARDWARE INFO</b>	
Cpu Cores Speed (Mhz)	2526
CPU Model	Intel(R) Xeon(R) CPU X3440 @ 2.53GHz
Memory Size (bytes)	19315245056
System Model	System x3250 M3 -[4252EBG]-
Num Cpu Cores	4
Num Cpu Pkgs	1
Num Cpu Threads	8
Num HBAs	5
Num Nics	3
Uuid	1365448e-f5a7-11de-ac67-00215e67bfcf
Vendor	IBM
Management Server Ip	192.167.147.136
Max EVC Mode Key	intel-nehalem
Overall Status	intel-nehalem
<b>SUMMARY QUICK STAT INFO</b>	
Distributed Cpu Fairness (unit)	-1
Distributed Memory Fairness (unit)	-1
Overall Cpu Usage (MHz)	1528
Overall Memory Usage (MB)	13674
Uptime (s)	213261
Reboot Required	false
<b>OTHER INFO</b>	
Datstores Number	1
Networks Number	2
VMs Number	8

**Tabella V-5: Proprietà rilevate per un elemento Host System del Cloud**

#### V.4.4.2. Metriche di performance

Per gli elementi di tipo Host System sono disponibili metriche di performance e metriche aggregate degli elementi associati di tipo Virtual Machine. Di seguito un esempio di statistiche visualizzate dal sistema per un elemento Host System del Cloud con i seguenti parametri:

*Host System: 192.167.147.101*

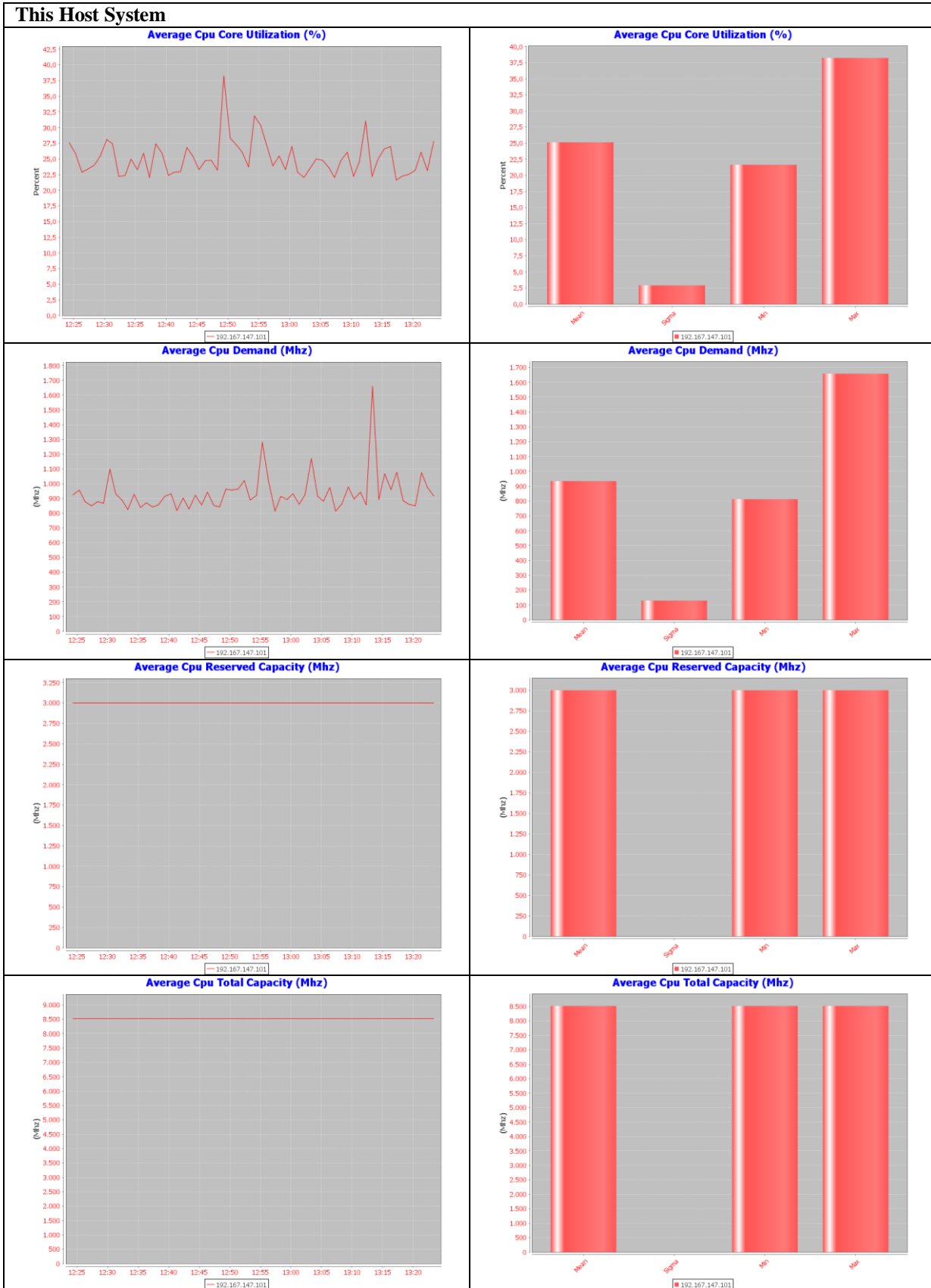
*Search parameter*

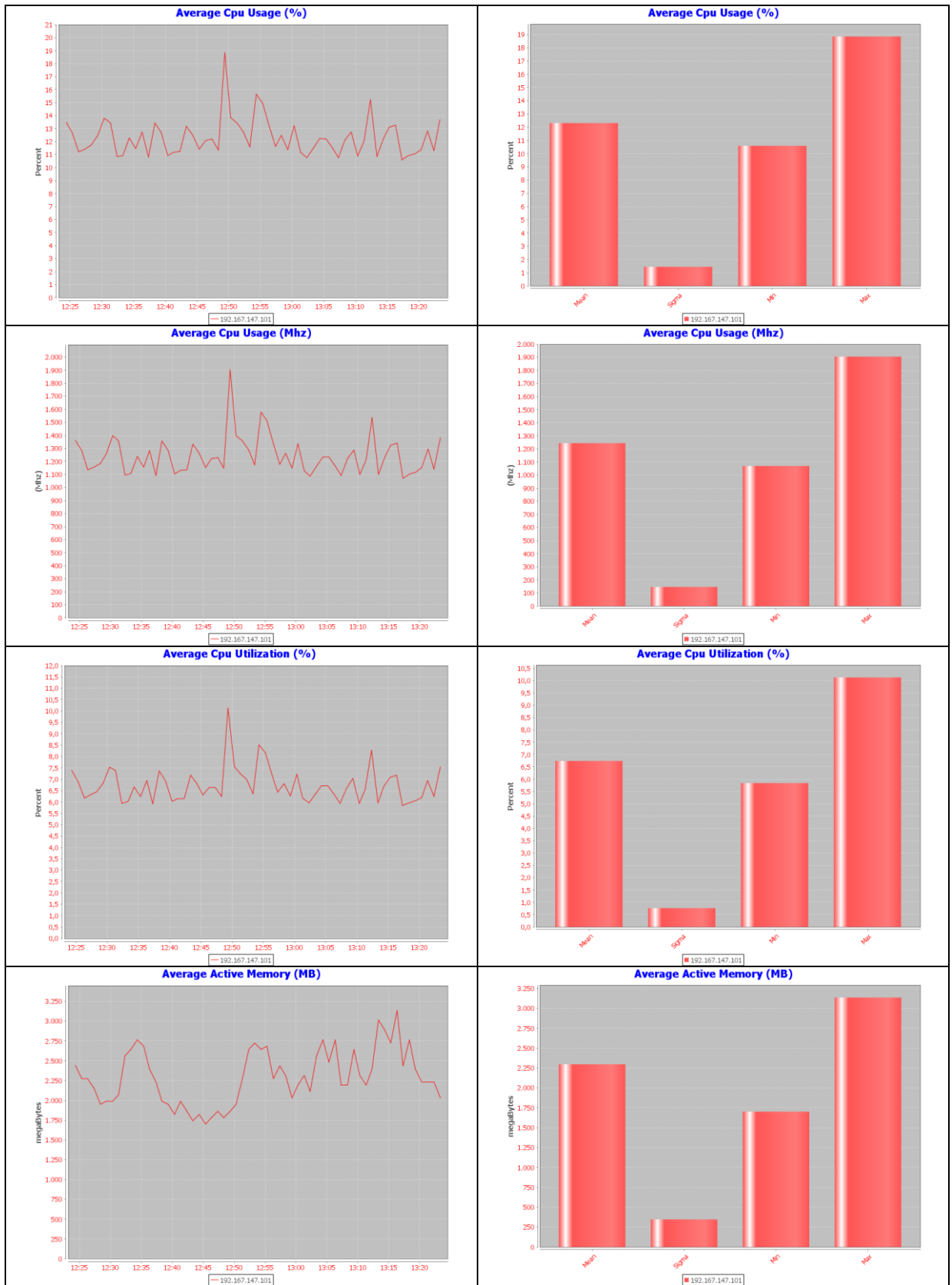
*Minute: one day*

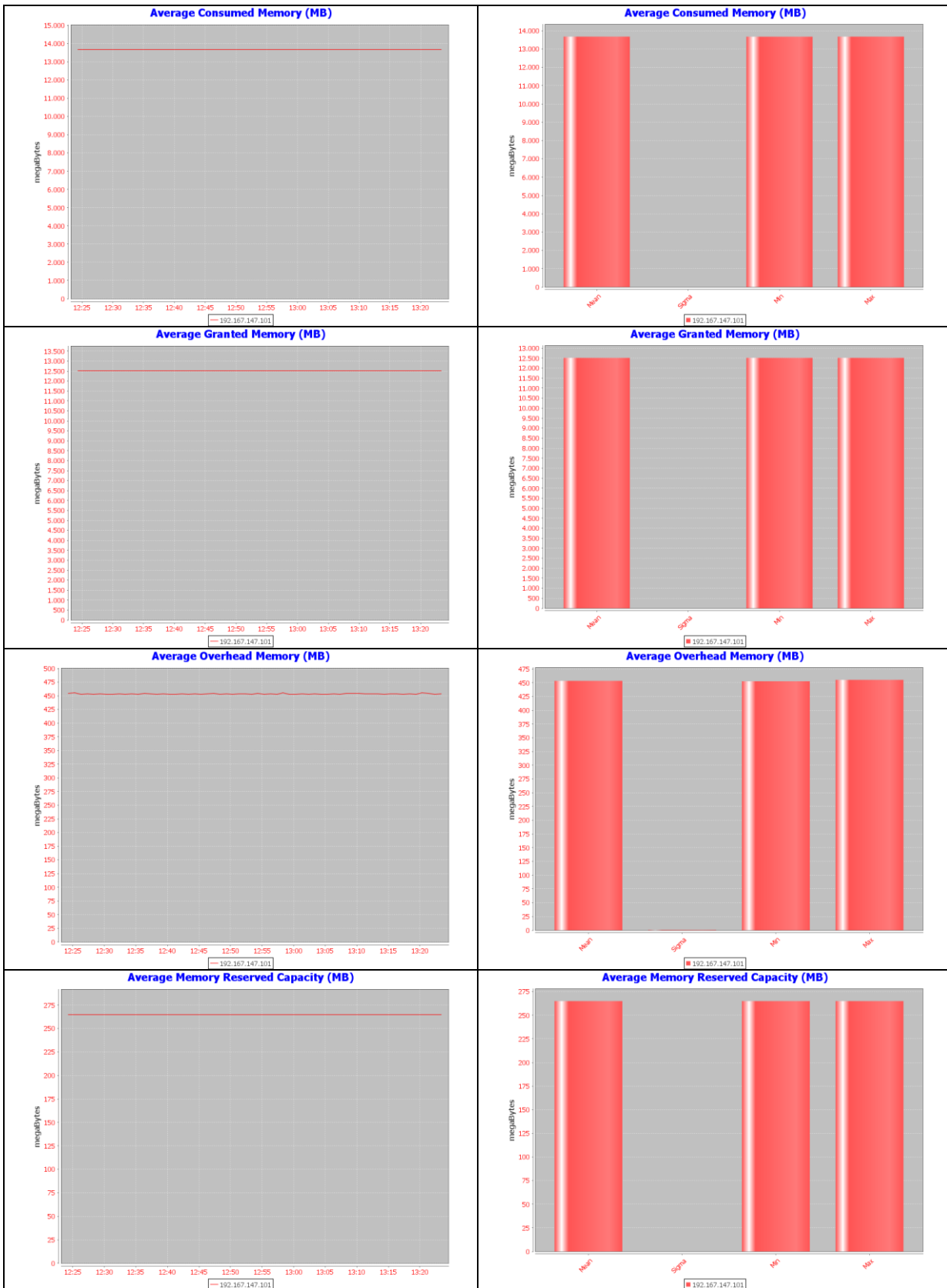
*Sorting type: descendant*

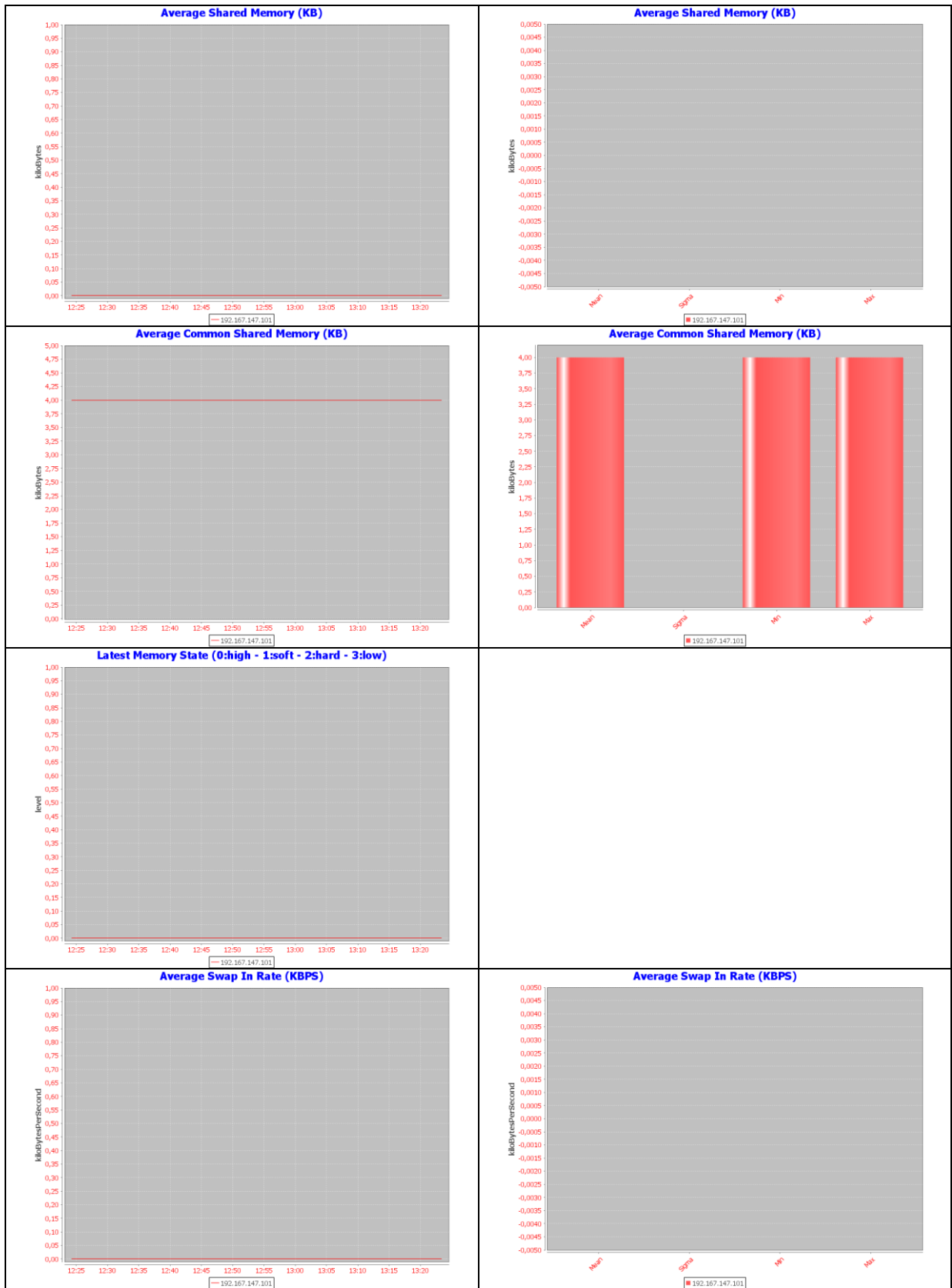
*Chart value limit: 20*

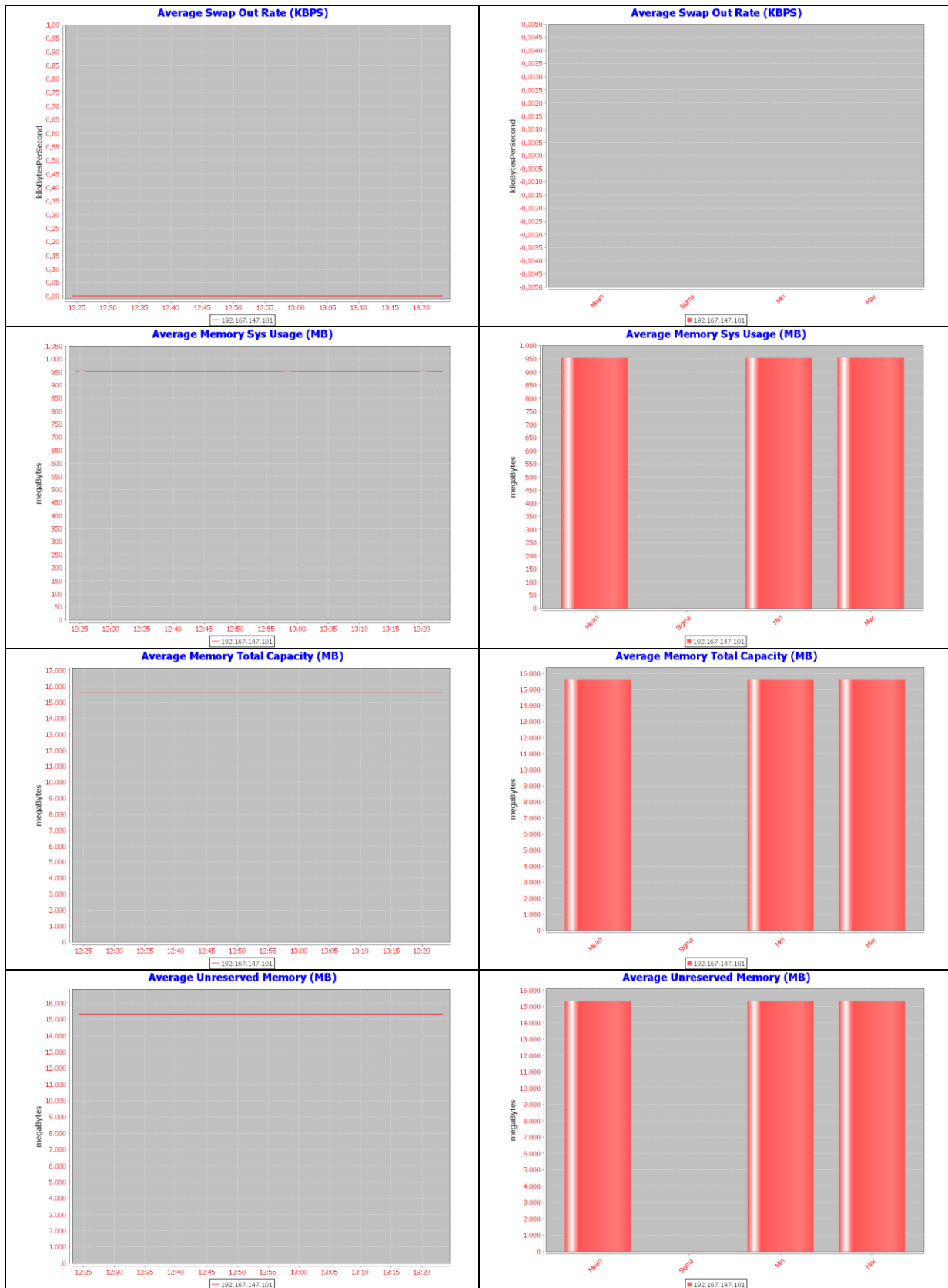


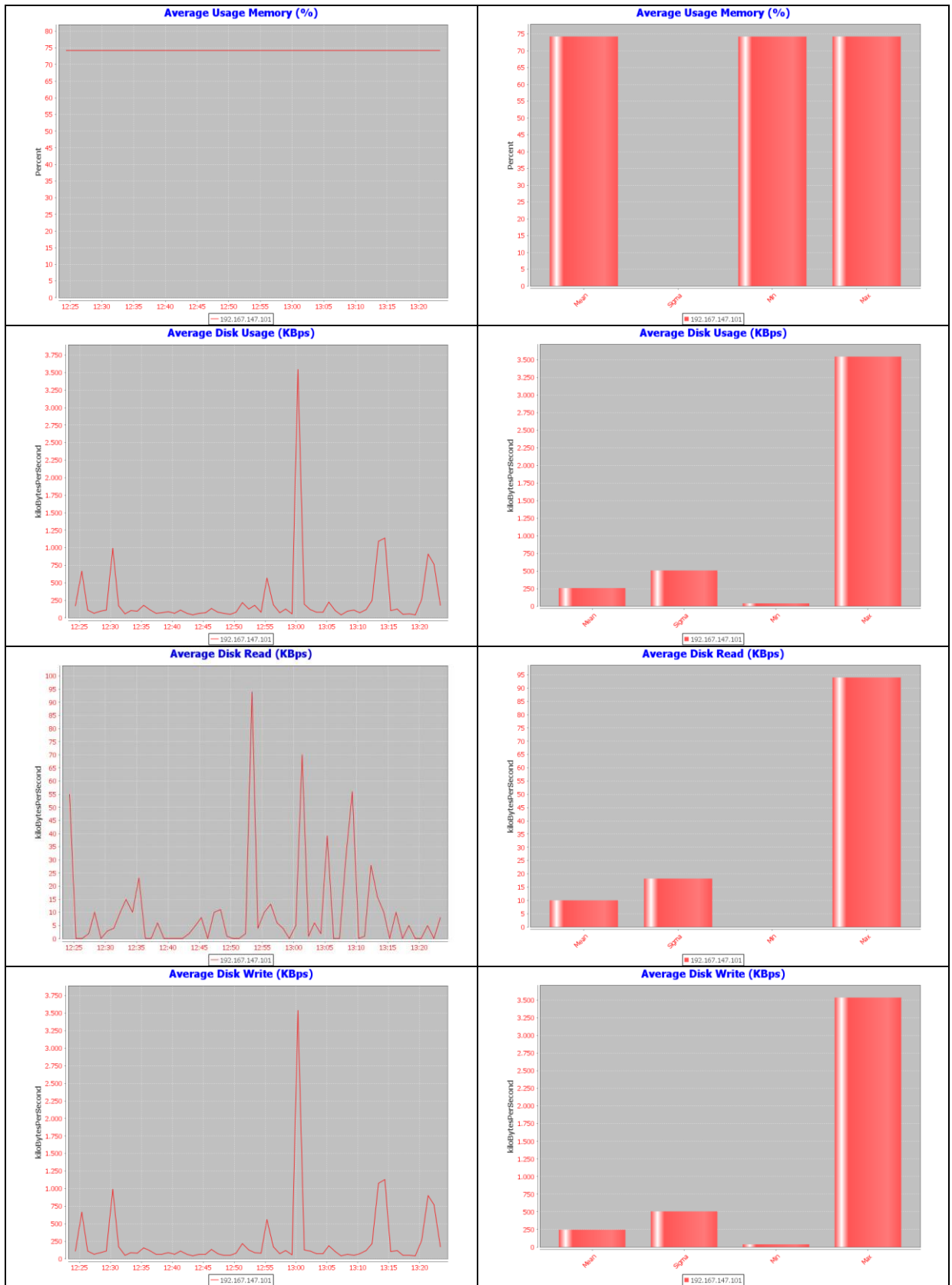




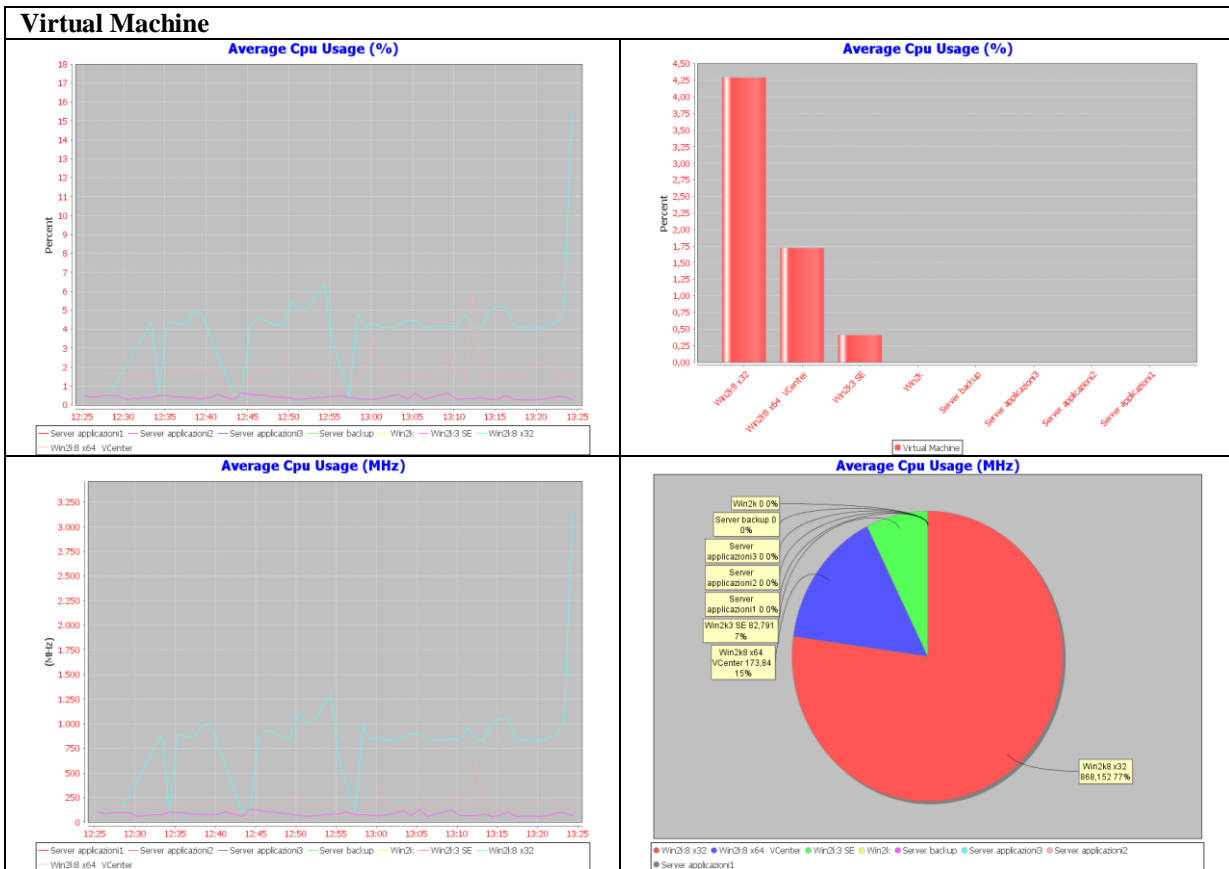
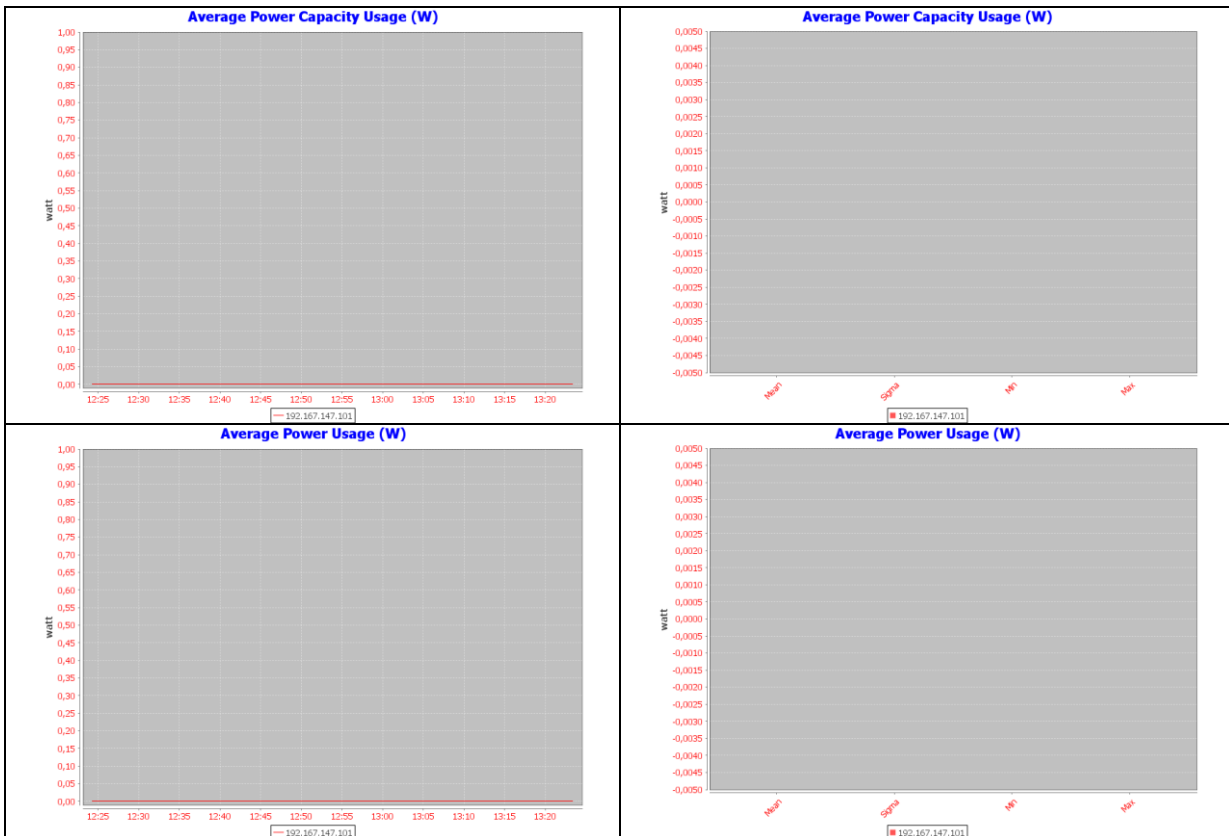


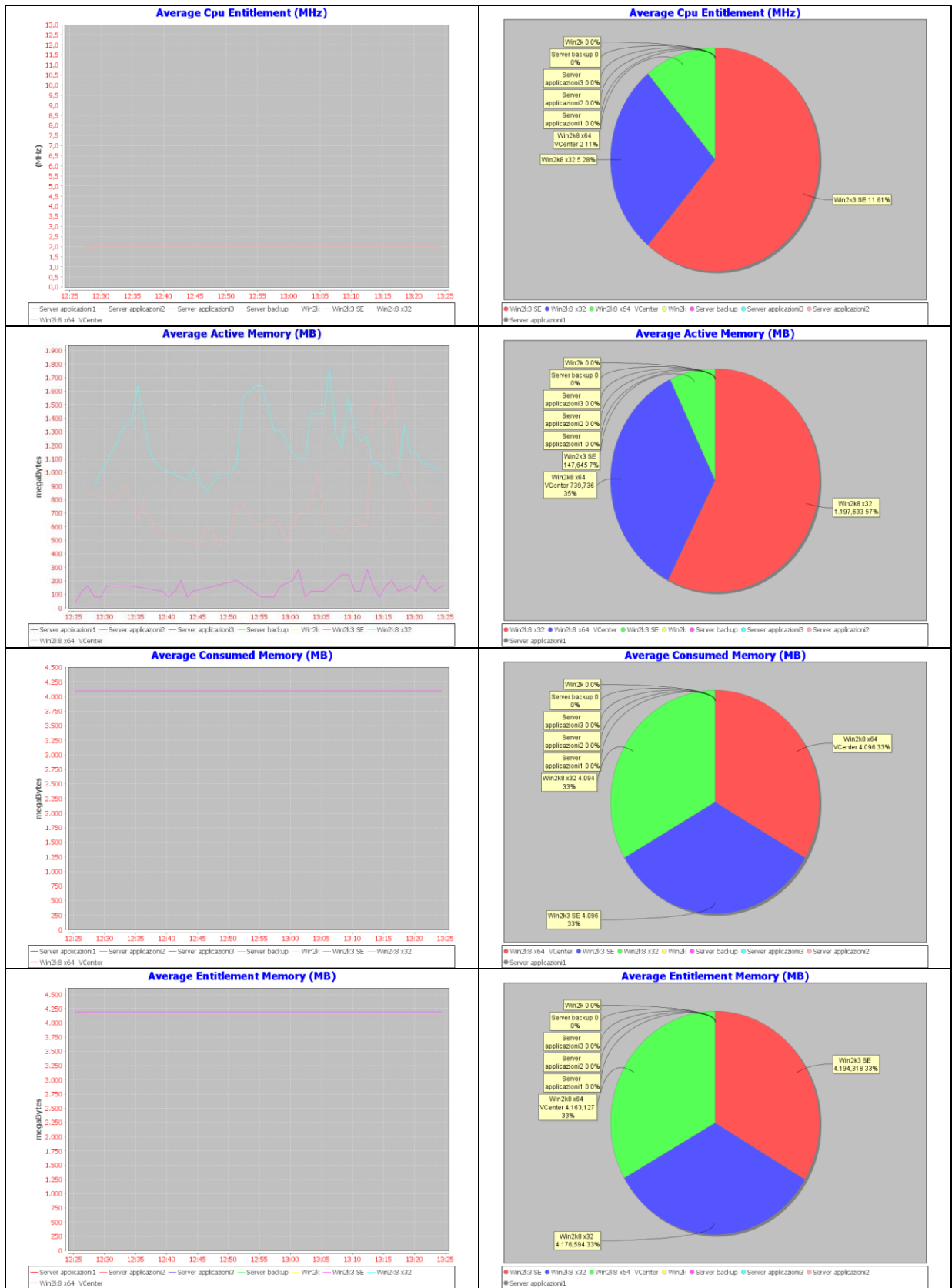






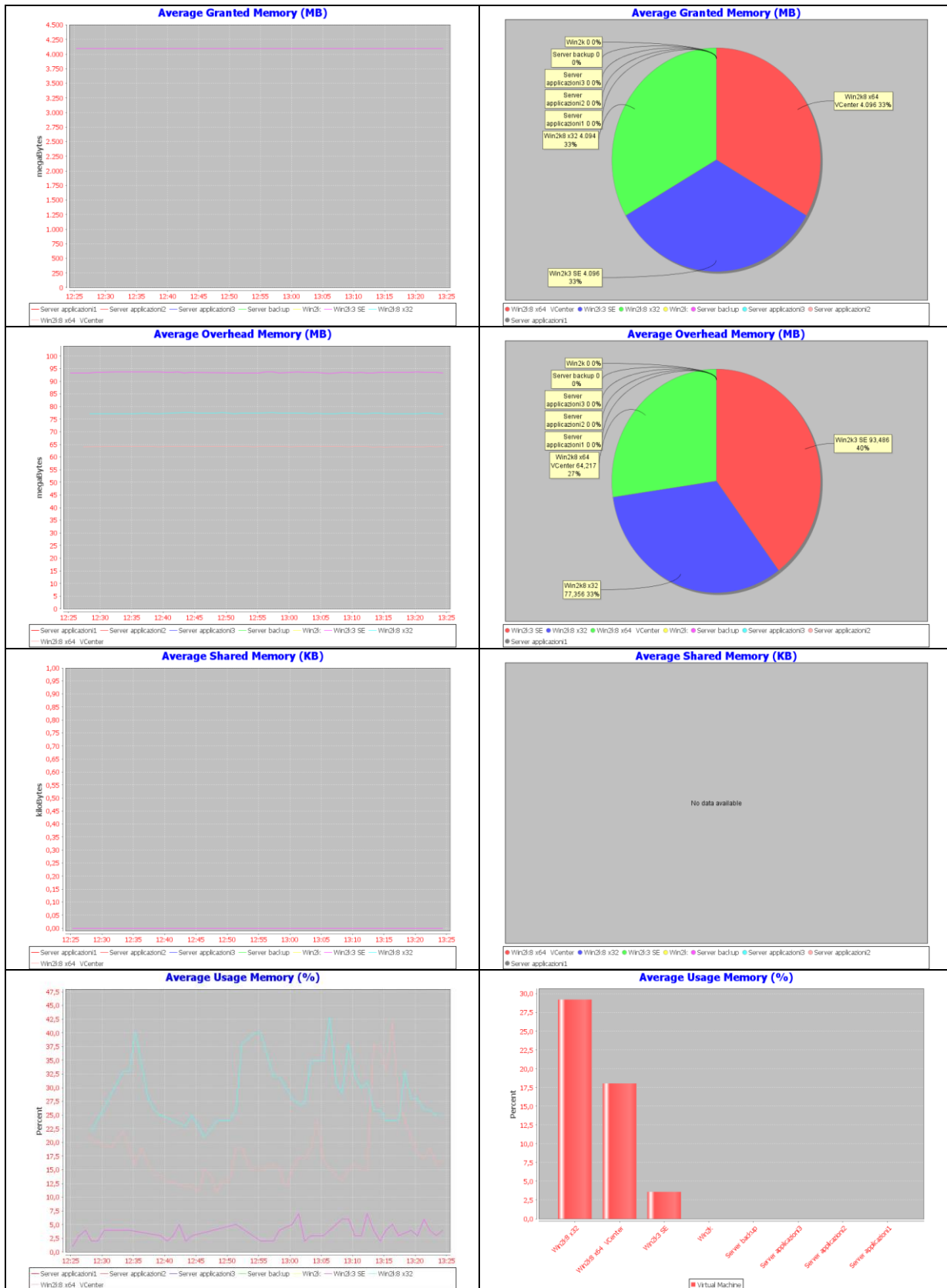
## V. Uso sperimentale - validazione

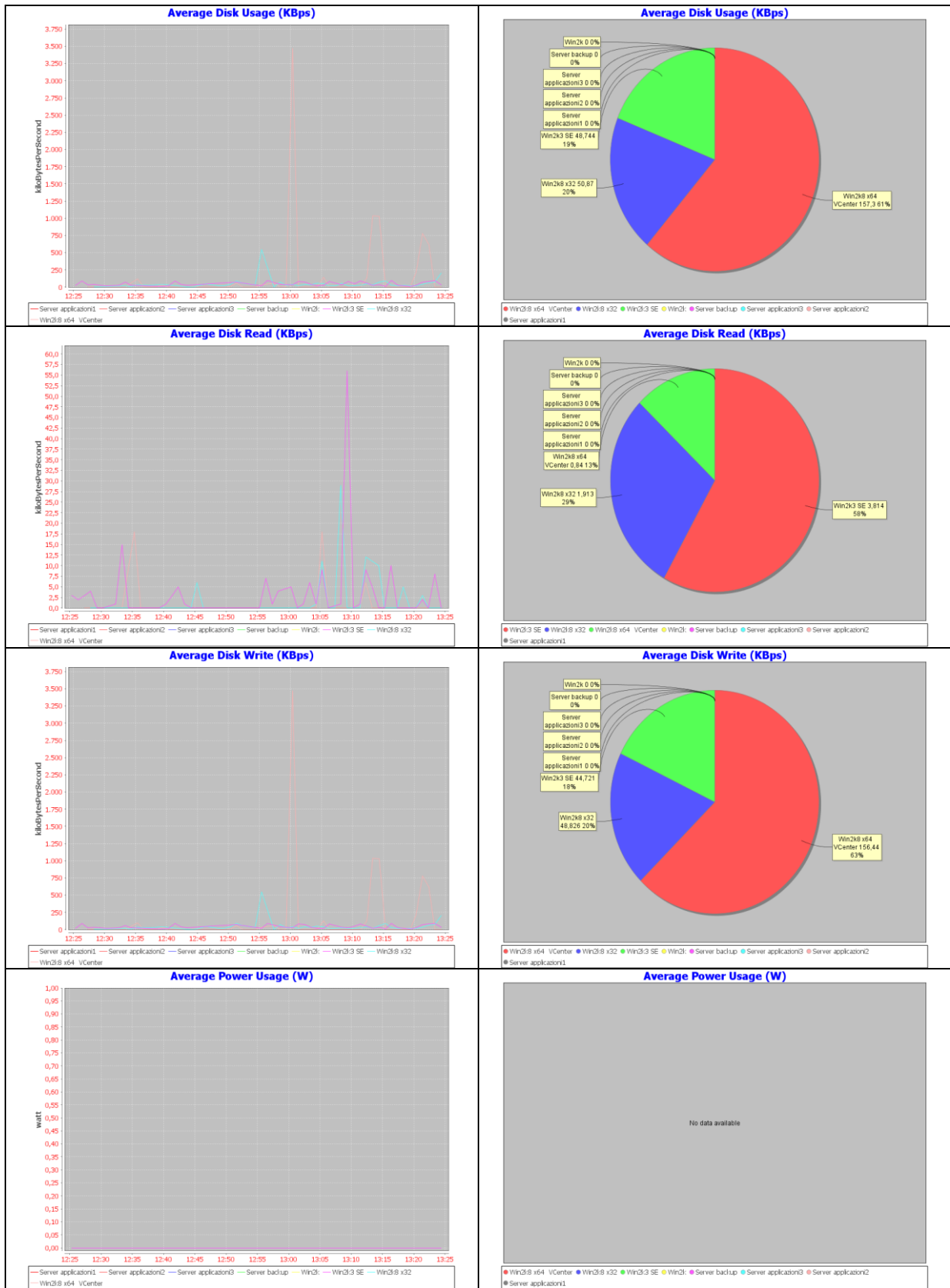






## V. Uso sperimentale - validazione





### V.4.5. Virtual Machine

Nell'infrastruttura sono presenti 40 Virtual Machine, sulle quali sono rilevabili proprietà e metriche e statistiche di performance.

**V.4.5.1. Proprietà**

Per gli elementi di tipo Virtual Machine sono rilevabili 79 proprietà. Un esempio di proprietà rilevate per un elemento di tipo virtual machine sono mostrate nella seguente tabella:

<b>Virtual Machine Properties - Win2k3 SE</b>	
<b>GENERAL INFO</b>	
Name	Win2k3 SE
Alarm Actions Enabled	true
Config Status	GREEN
Overall Alarm Status	GREEN
Guest Heartbeat Status	GREEN
<b>GENERAL CONFIG INFO</b>	
Alternate Guest Name	
Annotation	
Change Version	2013-10-21T12:15:47.761266Z
Firmware	bios
Guest Full Name	Microsoft Windows Server 2003 Standard (32-bit)
Guest Id	winNetStandardGuest
Instance UUID	522f69d4-f51a-fd01-6b79-aa65e8825d59
Modified	1970-01-01T00:00:00Z
Version	vmx-08
<b>GUEST SYSTEM INFO</b>	
App Heartbeat Status	appStatusGray
Guest Family	windowsGuest
Guest Full Name	Microsoft Windows Server 2003 Standard (32-bit)
Guest Id	winNetStandardGuest
Guest Operations Ready	true
Guest State	running
Host Name	utente-1gfv7yp3
Ip Address	192.167.147.159
Screen	1440 x 900
Tools Running Status	guestToolsRunning
Tools Version	8389
Tools Version Status2	guestToolsCurrent
<b>RESOURCE CONFIG INFO</b>	
Change Version	
Last Modified	
Cpu Allocation Limit (MHz)	-1
Cpu Allocation Reservation (MHz)	3000
Memory Allocation Limit (MB)	-1
Memory Allocation Reservation (MB)	0
Resource Pool	resgroup-50
<b>RUNTIME INFO</b>	
Boot Time	2013-10-21T12:15:47.762725Z

Clean Power Off	
Connection State	CONNECTED
Consolidation Needed	false
Vm Protection State	
Fault Tolerance State	NOT_CONFIGURED
Host	
Max Cpu Usage (MHz)	8516
Max Memory Usage (MB)	4096
Need Secondary Reason	
MKS Connections Number	0
Power State	POWERED_ON
Suspend Interval (mS)	0
Suspend Time	
<b>SNAPSHOT INFO</b>	
Current Snapshot	
<b>STORAGE INFO</b>	
Storage Committed (bytes)	90235419366
Storage Timestamp	2013-10-23T06:51:27.578364Z
Storage Uncommitted (bytes)	46137344
Storage Unshared (bytes)	90235419366
<b>GUEST SYSTEM CONFIG INFO</b>	
Cpu Reservation (MHz)	3000
Memory Reservation (MB)	0
Memory Size (MB)	4096
Cpu Number	8
Ethernet Cards Number	2
Virtual Disks Number	1
Template	false
Vm Path Name	[datastoreX3250] Windows Server 2003 SE thin/Windows Server 2003 SE thin.vmx
<b>QUICK STATS INFO</b>	
Ballooned Memory (MB)	0
Compressed Memory (KB)	0
Consumed Overhead Memory (MB)	93
Distributed Cpu Entitlement (MHz)	0
Distributed Memory Entitlement (MB)	0
Guest Memory Usage (MB)	163
Host Memory Usage (MB)	4189
Overall Cpu Demand (MHz)	75
Overall Cpu Usage (MHz)	75
Private Memory (MB)	4096
Shared Memory (MB)	0
Ssd Swapped Memory (KB)	0
Static Cpu Entitlement (MHz)	0
Static Memory Entitlement (MB)	0
Swapped Memory (MB)	0

Uptime Seconds (s)	154233
<b>OTHER INFO</b>	
Datastores Number	1
Networks Number	1

**Tabella V-6: Proprietà rilevate per un elemento Virtual Machine del Cloud**

**V.4.5.2. Metriche di performance**

Per gli elementi di tipo Virtual Machine sono disponibili metriche di performance e metriche aggregate degli elementi di tipo Datastore associati. Di seguito un esempio di statistiche visualizzate dal sistema per un elemento Datacenter del Cloud con i seguenti parametri:

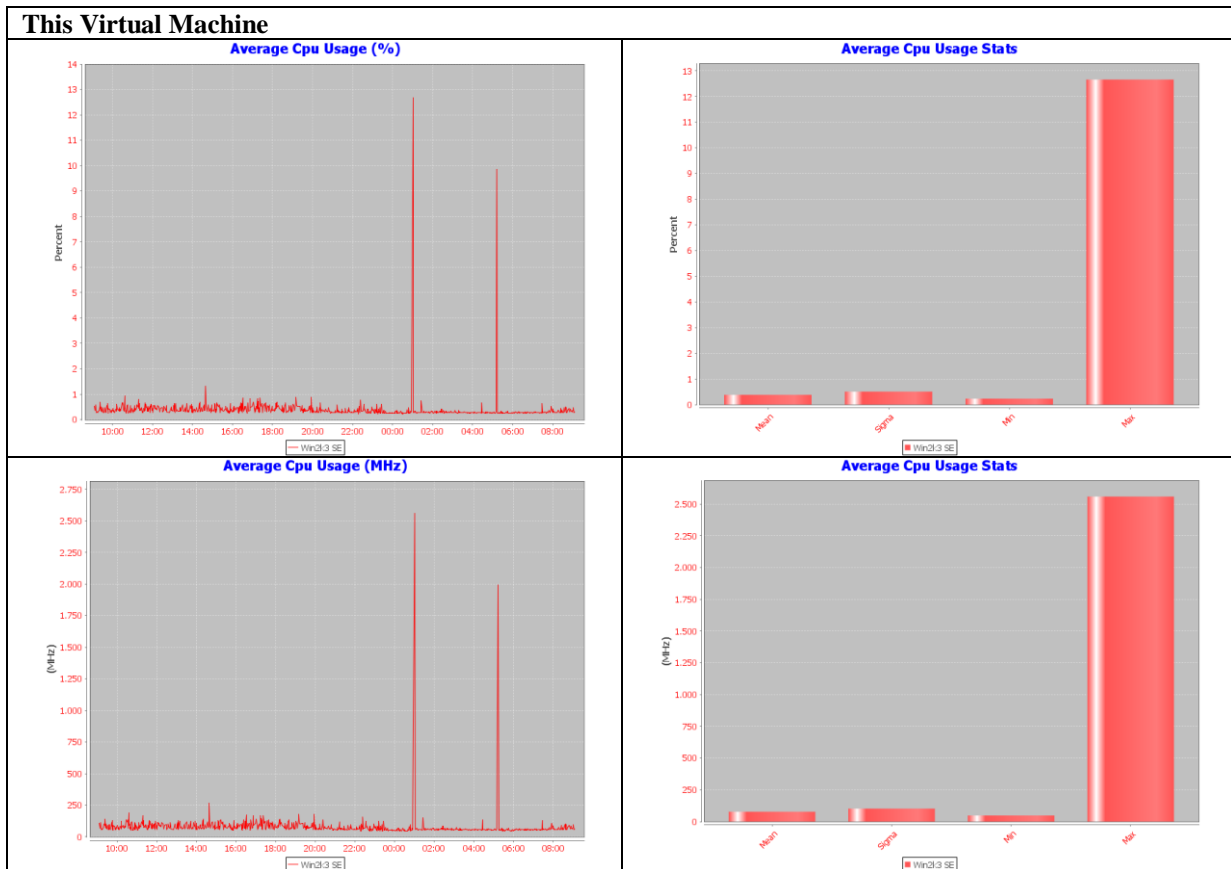
*Virtual Machine: Win2K3 SE*

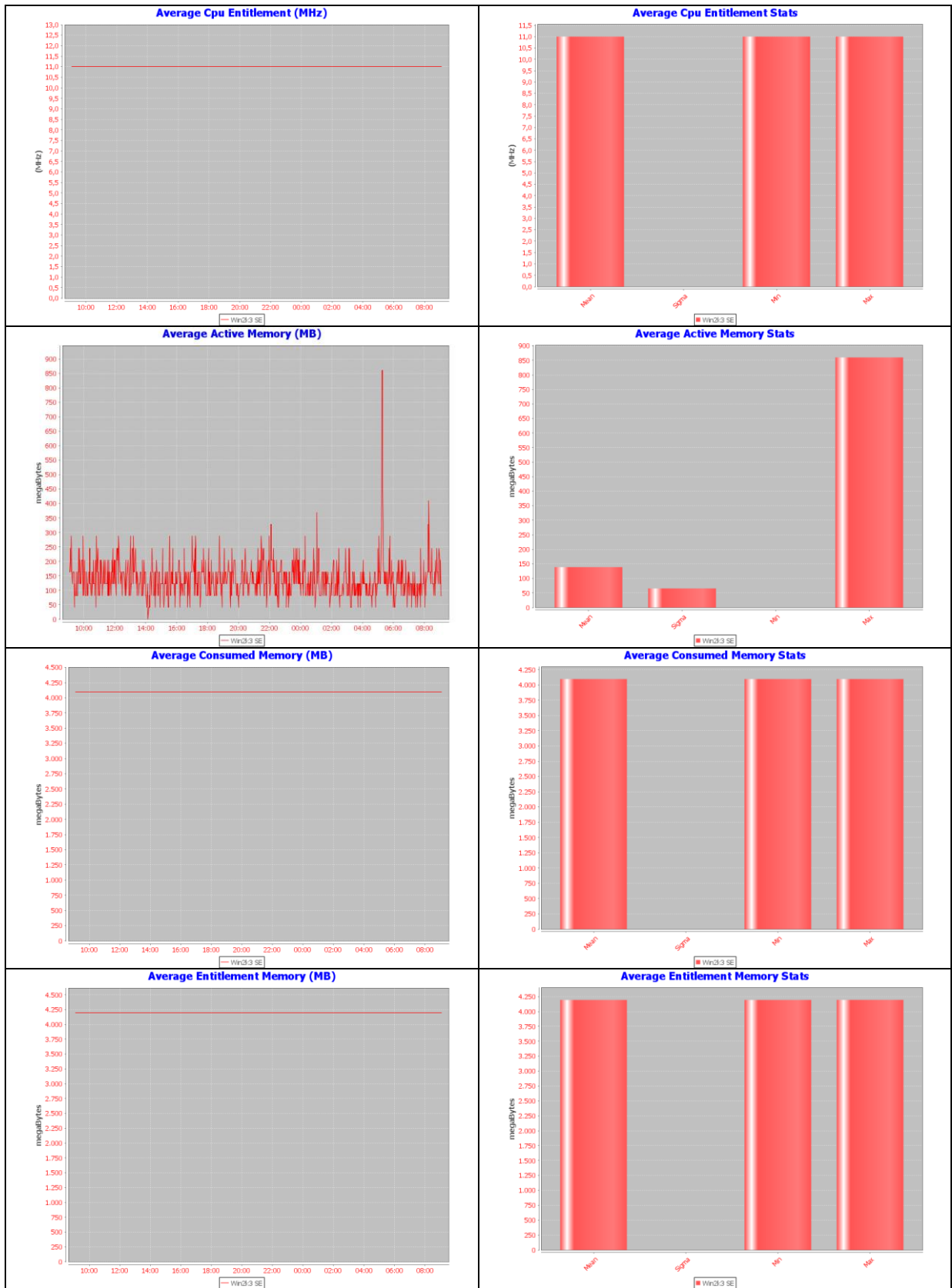
*Search parameter*

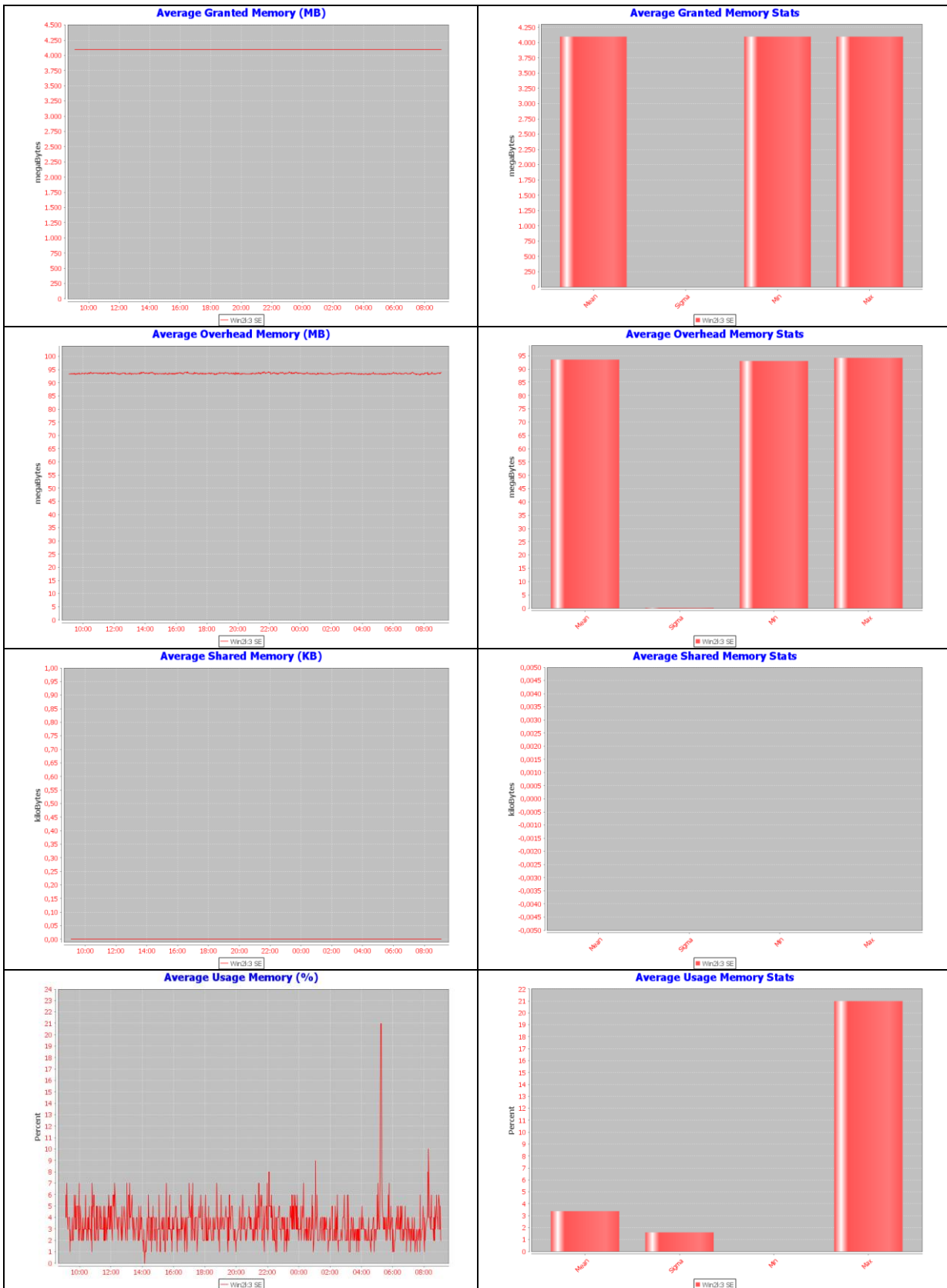
*Minute: one day*

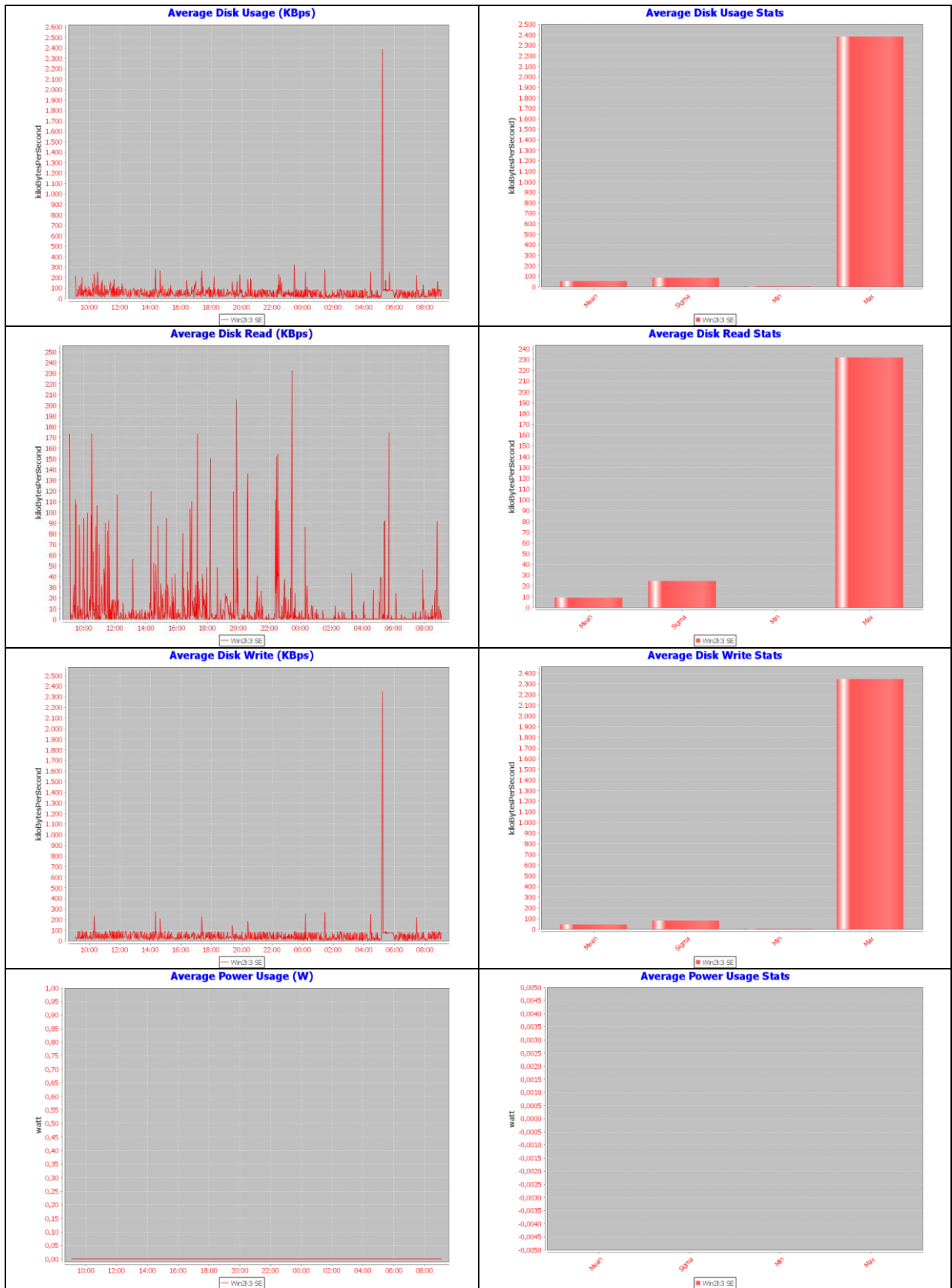
*Sorting type: descendant*

*Chart value limit: 20*

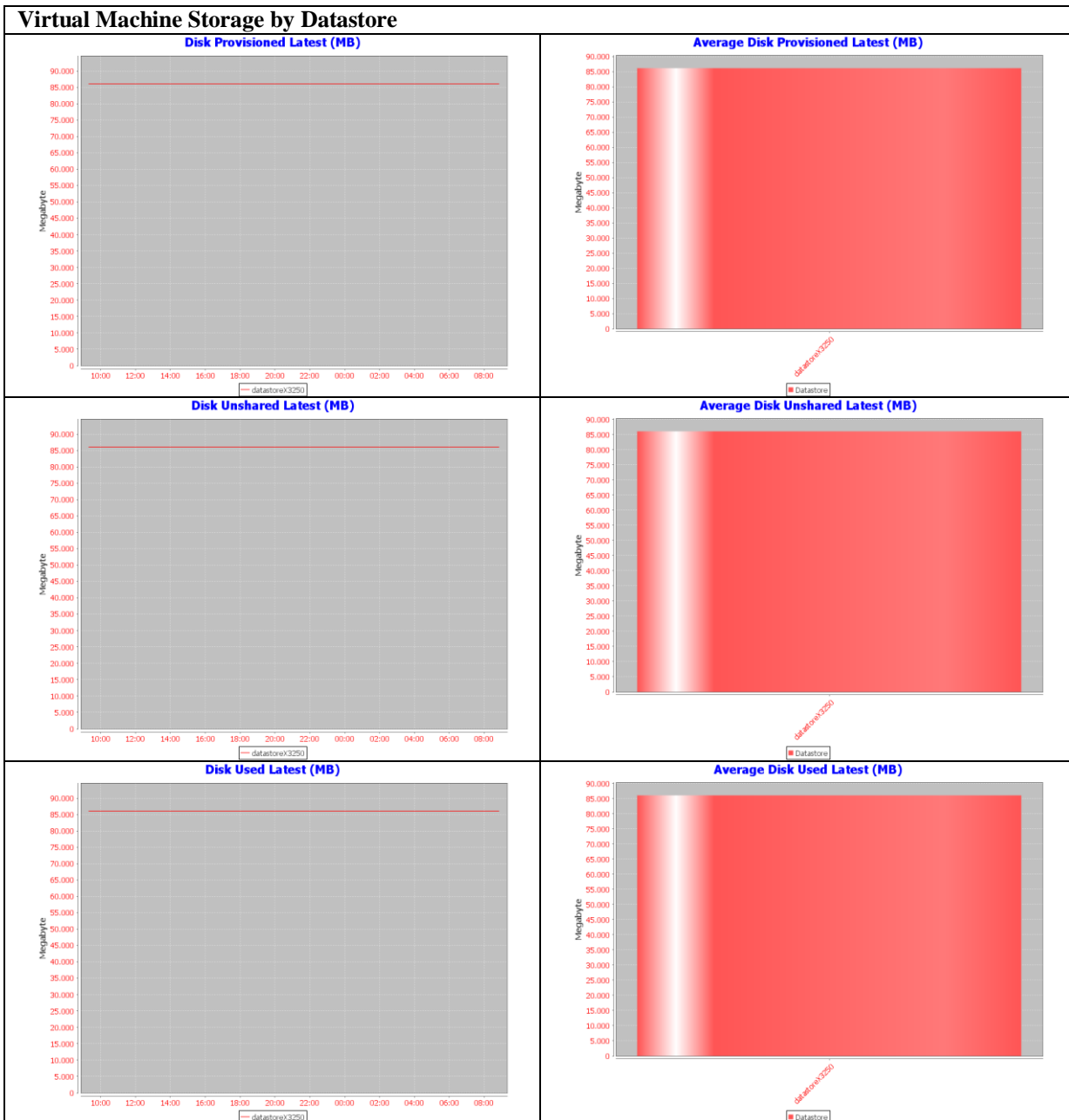












## V.5. Gestione degli allarmi

L'impostazione di un allarme consente di ricevere una email di notifica al verificarsi dello stesso. Un esempio di allarme è il seguente:

*Timestamp inserimento: 2013-10-25 19:56:34.0*

*Tipologia elemento: VirtualMachine*

*Nome : Win2k8 x32*

*Metrica: Cpu Usage Average*

*Condizione:* <

*Valore:* 10

*Durata superamento soglia ( minuti):* 5

*Alarm ring timestamp:* 2013-10-25 19:57:21

Al verificarsi dell'allarme il sistema ha inviato la seguente email di notifica all'indirizzo email dell'amministratore:

Oggetto: \*\*\*VSpere-Monitoring-Alert\*\*\*

Attention

Alarm ID: 5

Item type: VirtualMachine

Item name: Win2k8 x32

Item reference: vm-392

Metric: Cpu Usage Average %

Condition: <

Value: 10

Timelast: 5

Alarm Timestamp: 2013-10-25 19:57:21

L'allarme già verificatosi può essere riattivato oppure eliminato.



## VI. Conclusioni e sviluppi futuri

L'obiettivo della tesi è stato la progettazione di un sistema di monitoraggio per una infrastruttura Cloud basata su VMware vSphere. VMware è l'ecosistema Cloud più diffuso sul mercato anche grazie alla scalabilità dell'architettura e alla varietà di sistemi operativi supportati dal hypervisor ESXi utilizzato. Per la realizzazione del sistema di monitoring è stato scelto il design pattern MVC per via delle caratteristiche di estendibilità e facilità di riutilizzo del codice generato.

La prima fase del lavoro svolto ha riguardato lo studio dei principali sistemi di monitoraggio di infrastrutture Cloud basate su vSphere, analizzandone le caratteristiche e le funzionalità implementate. In particolare si sono valutate le caratteristiche di monitoraggio delle metriche di performance dei componenti hardware dell'infrastruttura e le funzionalità di alerting per cui, a seguito dell'inserimento di un allarme, il sistema notificherà lo stato di allarme attivo all'amministratore.

A seguito di questa fase di studio sono state decise le funzionalità da implementare nel sistema da progettare e codificate mediante i diagrammi UML dei casi d'uso.

Nella seconda fase del lavoro si è proceduto alla progettazione del sistema secondo le funzionalità decise utilizzando il design pattern sopracitato. Dallo studio delle vSphere API si sono verificate le metriche di performance hardware e le proprietà rilevabili attraverso le classi pubbliche a disposizione degli sviluppatori. Si è proceduto quindi con la progettazione concettuale e logica del database dell'applicazione e alla realizzazione della struttura dei tre componenti dell'applicazione: Model, View e Controller.

Si sono realizzate le funzionalità richieste dal sistema per quanto riguarda il recupero delle metriche di performance e delle proprietà dei componenti dell'infrastruttura. Fondamentale è stata la ricostruzione della struttura ad albero dell'inventario dell'infrastruttura Cloud, per consentire l'accesso all'amministratore del sistema ai diversi componenti della infrastruttura monitorata per poi consentirne la visualizzazione della proprietà, delle metriche statistiche anche in forma grafica, e per l'inserimento degli allarmi. Per il recupero delle metriche di performance si sono realizzate classi di accesso al server vCenter e di recupero dei dati, i quali poi sono memorizzati nel database per mezzo di classi di tipo manager e DAO.

L'implementazione delle funzionalità del sistema quali il monitoring, l'alarmino, e la visualizzazione utente, ha richiesto l'utilizzo di più thread. In particolare, dal thread di esecuzione principale vengono creati dei thread di gestione degli altri thread di recupero delle performance e di verifica degli allarmi.

Il sistema è facilmente estendibile al monitoraggio di infrastrutture Cloud basate su altri software grazie alla modularità data dal design pattern e alla separazione dei ruoli delle classi. La parte View, composta da jsp, essendo completamente separata dal Model per mezzo del Controller, che ne realizza un disadattamento di impedenza, permette la completa personalizzazione dell'interfaccia utente senza alcuna modifica delle altre componenti.

Il sistema è stato validato grazie all'utilizzo sperimentale in più infrastrutture Cloud di diverse dimensioni, che hanno consentito di verificare la presenza di eventuali criticità non prevedibili nella fase di progettazione dell'applicazione.

In particolare, l'utilizzo del sistema per il monitoraggio dell'infrastruttura Cloud privata presso la Facoltà di Scienze Economiche Giuridiche e Politiche dell'Università di Cagliari, ha consentito lo studio della corretta ed ottimale configurazione dei parametri sia del database MySQL dell'applicazione, relativamente all'utilizzo della memoria cache delle tabelle, sia dell'application container Apache Tomcat, relativamente all'assegnazione della memoria per le diverse tipologie previste dalla JVM. Un altro parametro fondamentale è il tempo intercorrente tra due cicli di

monitoring e di alarming. Grazie all'uso sperimentale si è potuto verificare quale sia il tempo ottimale per quella determinata infrastruttura e, più in generale, si è verificato un tempo di recupero delle metriche di performance lineare rispetto al numero di virtual machine ed host fisici presenti nell'infrastruttura.

Il sistema, grazie alla visualizzazione grafica dell'andamento delle principali metriche di performance, anche in maniera aggregata, consente di analizzare lo stato complessivo del Cloud mediante poche view; per esempio è possibile verificare l'andamento nel tempo dell'intero datacenter, dei cluster, degli host, delle virtual machine e dei datastore associati in una unica pagina web.

Una caratteristica importante del sistema è la conservazione di tutti i dati sulle metriche rilevati, in tal modo è possibile verificare ex post l'andamento del Cloud in un determinato periodo temporale passato senza perdita di informazione alcuna.

L'accesso web poi consente l'accesso all'applicazione da qualsiasi postazione munita di web browser, comprendendo quindi i moderni smartphone.

Alcune funzionalità che è possibile aggiungere al sistema nel futuro sono: la possibilità di assegnare più utenti al sistema con assegnazione degli elementi dell'inventario su cui ogni utente può operare; l'invio periodico dei report grafici agli amministratori del sistema via email; il monitoraggio di infrastrutture Cloud basate su tecnologie differenti; un sistema di previsione neuronale di saturazione delle capacità hardware e/o basato su medie mobili o estrapolazione statistica;

In generale, il monitoraggio e l>alerting realizzati, consentono un migliore utilizzo delle risorse hardware disponibili nell'infrastruttura, un più immediato intervento in caso di criticità e una conseguente riduzione dei tempi di down della stessa, nonché una riduzione dei costi energetici grazie all'ottimizzazione dell'infrastruttura.

# Appendice A VMware vSphere API

## VI.1. Introduzione

In questo Capitolo verrà illustrata l'architettura delle vSphere Management API [VMW11f] e le principali classi e strutture dati utilizzate.

## VI.2. vSphere Management APIs

vSphere supporta in maniera robusta la virtualizzazione delle applicazioni, il networking e lo storage offrendo moduli e componenti opzionali quali VMware High Availability e VMware vMotion. Il VMware vSphere Web Services SDK permette allo sviluppatore di accedere ai componenti e ai moduli dell'infrastruttura. I moduli base di vSphere, ovvero ESX/ESXi Server, vCenter Server e vSphere Client, consentono la gestione di base delle virtual machine e forniscono il supporto all'intero sistema. In particolare:

- ESX/ESXi hypervisor Server: supporta l'esecuzione di più virtual machine e degli altri componenti accessori quali il networking e lo storage.
- vCenter Server: sistema di gestione dell'intera infrastruttura, dispone di un database integrato come servizio, consente la gestione in maniera centralizzata di più server ESX/ESXi.
- vSphere Client: è la GUI (Graphical User Interface) per la gestione di vSphere. I client vSphere in esecuzione sui sistemi Windows utilizzano il vSphere Web Services SDK per la comunicazione con i server vSphere ESX/ESXi e vCenter Server.

Le API di vSphere sono implementate utilizzando i Web Services, indipendenti dal linguaggio e basati su un meccanismo RMC (Remote Call Procedure), utilizzato dalle applicazioni client per l'accesso ai server vSphere ESX/ESXi e vCenter Server. Ogni applicazione client per connettersi al server deve inviare allo stesso le credenziali di accesso per l'autenticazione.

Le applicazioni client che utilizzano le vSphere API devono eseguire le seguenti operazioni:

- 1- Creare una connessione al vSphere server Web service.
- 2- Istanziare un oggetto locale proxy per referenziare il ServiceInstance e recuperare l'oggetto ServiceContent dal server che contiene il riferimento alla directory root dell'inventario.
- 3- Istanziare un oggetto locale proxy per accedere ai metodi delle vSphere API.
- 4 - Login al server utilizzando le credenziali di accesso utente.
- 5- Accesso agli oggetti server-side per recuperare i dati ed eseguire operazioni di gestione dei componenti.
- 6 - Chiudere la connessione

### VI.3. Architettura Client-Server di vSphere

Le applicazioni client partecipano all'architettura distribuita, utilizzando un modello di comunicazione asincrono. Questa architettura è basata su Managed objects lato server, Managed object references lato client e Data objects. In particolare:

- **Managed object:** sono presenti nei server vSphere (ESX/ESXi e vCenter Server system). Rappresentano i servizi ed i componenti di vSphere. I servizi comprendono PropertyCollector, SearchIndex, PerformanceManager, e ViewManager. I componenti comprendono gli oggetti dell'inventario quali VirtualMachine, Datastore e Folder.
- **Managed object reference (MOR):** sono utilizzati dalle applicazioni client per accedere agli oggetti lato server. I Managed object references identificano in modo univoco gli oggetti all'interno del sistema. Tali oggetti persistono tra più sessioni ed esistono finché esistono gli oggetti a cui fanno riferimento. L'eliminazione di una VirtualMachine, ad esempio, implica la contestuale eliminazione del suo MOR.
- **Data object:** tali oggetti contengono informazioni sui Managed object e Managed object references. L'applicazione client invia e riceve i dati da un server vSphere attraverso questi oggetti. Esempi di questi oggetti sono la specifiche degli oggetti dell'inventario quali VirtualMachineConfigSpec e HostCapability.

Il client ha un riferimento MOR ad una Virtual Machine sul server e una copia delle informazioni di base GuestInfo. Le applicazioni vSphere utilizzano un modello di comunicazione client-server asincrona in cui i metodi sono non bloccanti.

La seguente figura mostra l'architettura Client/Server utilizzata:

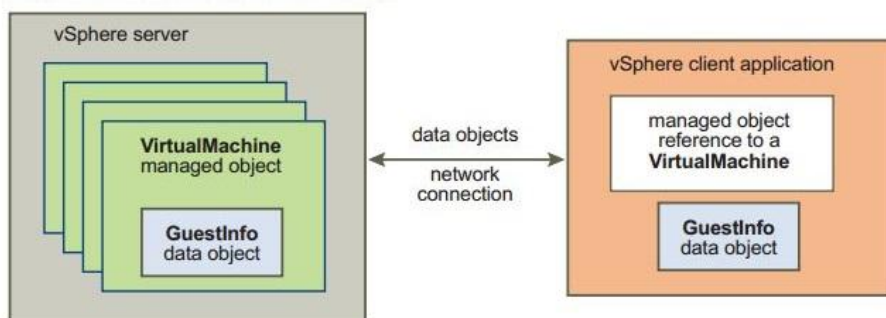


Figura A-0-1: vSphere Server and Client

#### vSphere API e i Web Service

Le vSphere API, in funzione negli server ESX/ESXi e nei vCenter Server, utilizzano i web services secondo le prescrizioni del Web Services Interoperability Organization (WS-I) Basic Profile 1.0.

Il WS-I Basic Profile 1.0 include:

- XML Schema 1.0
- SOAP 1.1

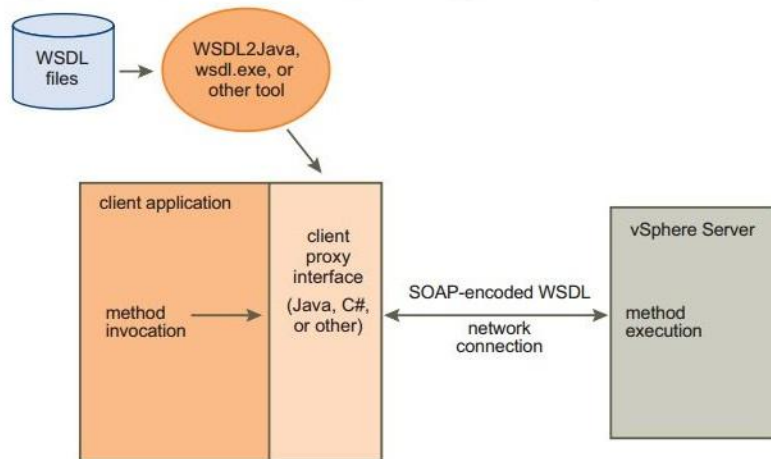
- WSDL 1.1

La tecnologia web services supporta le operazioni, l'equivalente dei metodi negli altri linguaggi di programmazione, e fornisce l'accesso a tutte le operazioni necessarie per il monitoraggio e la gestione dei componenti dell'intera infrastruttura vSphere.

### WSDL Files and the Client-Side Proxy Interface

Il vSphere Web Services SDK fornisce un set di WSDL (Web Services Description Language) file, che definiscono le vSphere Web Services API. I web-services development tools quali JAX-WS usano questi file WSDL per generare il codice proxy client-side definito (stubs).

Il proxy facilita la chiamata remota dei metodi, degli oggetti e dei dati all'interno della architettura OOP distribuita. L'applicazione client utilizza SOAP (Simple Object Access Protocol) per scambiare messaggi WSDL con i server vSphere. Nella seguente figura è rappresentato l'utilizzo delle chiamate ai metodi da parte del client proxy interface, quest'ultima basata sulle definizioni WSDL.



**Figura A-0-2: Client-Server Communication Through a Client Proxy Interface**

Per utilizzare il VMware client proxy interface è necessario importare le vSphere API client libraries nell'applicazione client, in particolare per le applicazioni java verrà importata la seguente libreria:

- `Java import com.vmware.vim25.*;`

L'applicazione client può utilizzare le vSphere API per comunicare con i server vSphere (ESX/ESXi e vCenter) attraverso il protocollo HTTPS alla porta 443 come porta e protocollo di default. E' possibile configurare i server per il supporto del protocollo HTTP per i soli ambienti di test dato che non ne garantisce la sicurezza.

Gli strumenti SOAP generano classi e metodi specifici previsti nelle definizioni WSDL oltre a classi e metodi non previsti. Questi ultimi sono dei seguenti tipi:



- Generated objects. Oggetti aggiuntivi che forniscono l'accesso ai vSphere Web Services, utilizzati per stabilire la connessione client-server (VimServiceLocator, AppUtil, VimPortType, VimService).
- Generated methods. Metodi di accesso e di modifica aggiuntivi di tipo getter e setter.

I Data Type XML sono mappati nei Data Type del linguaggio Java (e nel linguaggio C#) secondo il seguente schema mostrato in figura per i tipi di dato più diffusi:

XML Schema	Java	.NET Data Type
xsd:base64binary	byte[]	Byte[]
xsd:boolean	boolean	Boolean
xsd:byte	byte	SByte
xsd:dateTime	java.util.Calendar	DateTime
xsd:decimal	java.math.BigDecimal	Decimal
xsd:double	double	Double
xsd:float	float	Single
xsd:int	int	Int32
xsd:string	java.lang.String	String

**Figura A-0-3: Mappatura di alcuni Data Type nei rispettivi dei linguaggi di programmazione Java e C#**

Le applicazioni client che utilizzano le vSphere API devono eseguire le seguenti operazioni:

- 1 - Creare una connessione al vSphere Server Web Services
- 2 - Istanziare un oggetto locale proxy per referenziare il ServiceInstance e recuperare l'oggetto ServiceContent dal server, quest'ultimo contiene il riferimento alla directory root dell'inventario.
- 3 - Istanziare un oggetto locale proxy per accedere ai metodi delle vSphere API.
- 4 - Login al server utilizzando le credenziali di accesso utente.
- 5 - Accesso agli oggetti server-side per recuperare i dati ed eseguire operazioni di gestione dei componenti.
- 6 - Chiudere la connessione.

## VI.4. Accesso ai vSphere Server ESX/ESXi e vCenter

VMware vSphere implementa dei meccanismi che consentono l'accesso ai componenti dell'infrastruttura ai soli utenti autorizzati. Per fare ciò, ad ogni proprietà e metodo delle API sono associati dei privilegi e solo gli utenti con tali privilegi possono avere accesso a tali entità. I server ESX/ESXi utilizzano un sistema di autenticazione basato su Linux mentre i server vCenter utilizzano un sistema di autenticazione basato sui servizi di Windows.

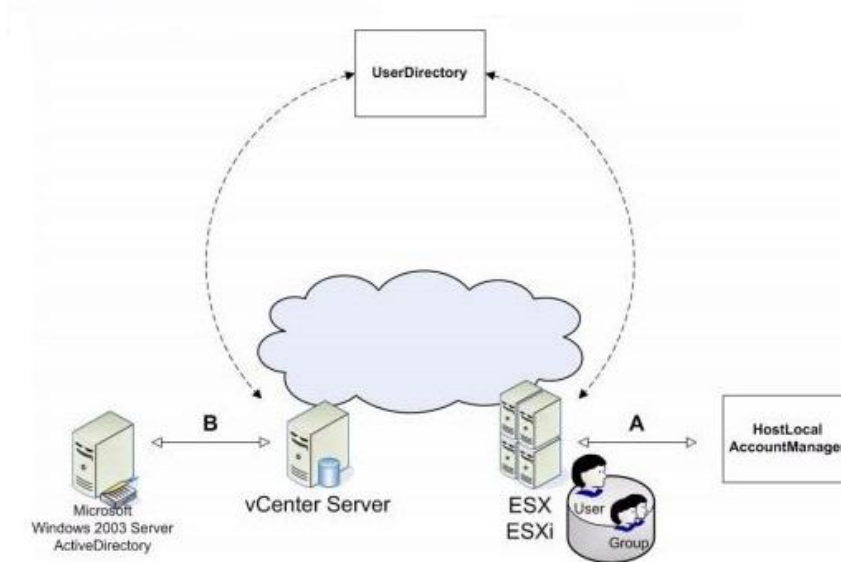


Figura A-0-4: Autenticazione dei server vSphere ESX/ESXi e vCenter

### Modello di autenticazione ESX/ESXi

Quando un utente inserisce le proprie credenziali via client, il server verifica l'autenticità dell'utente. Il server supporta sia password sia certificati quali X.509 certificate. Una volta autenticato l'utente può accedere esclusivamente agli oggetti per i quali è autorizzato. I server ESX/ESXi utilizzano il Linux pluggable authentication module (PAM).

### Modello di autenticazione vCenter Server

vCenter è un servizio di Windows ed utilizza il modello di autenticazione degli utenti di Windows. L'amministratore del sistema deve avere il ruolo di Administrator della macchina in cui è in funzione vCenter. Il sistema supporta il controller di dominio Active Directory.

I passi necessari per l'accesso ai vSphere server (ESX/ESXi o vCenter Server) per mezzo delle vSphere Web Services API sono dettagliati nel seguente esempio che utilizza il linguaggio Java:

1 - Creare un Managed object reference per l'oggetto ServiceInstance presente sul server:

```
ManagedObjectReference SVC_INST_REF = new ManagedObjectReference();
SVC_INST_REF.setType("ServiceInstance");
SVC_INST_REF.setValue("ServiceInstance");
```

2 - Creare un oggetto VimService per ottenere il VimPort binding provider. Questo fornisce l'accesso ai campi del protocollo nei messaggi request/response. Gli oggetti VimServiceLocator e VimPortType forniscono l'accesso ai vSphere Server. Il metodo getVimPort restituisce un oggetto VimPortType che fornisce l'accesso ai metodi delle vSphere API:

```
vimService = new VimService();
vimPort = vimService.getVimPort();
Map<String, Object> ctxt = ((BindingProvider) vimPort).getRequestContext();
```

3 - Memorizzare l'URL del server nel context della request, specificando il valore *true* per mantenere attiva la connessione tra il client ed il server. Il client includerà i cookie del server http nelle proprie richieste per mantenere la sessione.

```
ctxt.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, url);  
ctxt.put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
```

4 - Recuperare l'oggetto ServiceContent ed eseguire il login al server:

```
serviceContent = vimPort.retrieveServiceContent(SVC_INST_REF);  
vimPort.login(serviceContent.getSessionManager(), userName, password, null);  
isConnected = true;
```

5 - Chiudere la connessione utilizzando nuovamente l'oggetto VimPort:

```
try {  
    vimPort.logout(serviceContent.getSessionManager());  
} catch (Exception e) {  
    System.out.println(" Connect Failed ");  
    e.printStackTrace();  
}
```

Quando una applicazione client si connette ad un vSphere server (ESX/ESXi o vCenter Server), il server riconosce la versione delle API utilizzate nello sviluppo dell'applicazione client e rende disponibili esclusivamente le funzionalità supportate dal client. Le informazioni sulla versione delle API utilizzate vengono trasmesse mediante messaggi SOAP dal client. Lato server, le informazioni sulle versioni supportate sono contenute nel file XML vimServiceVersions.xml.

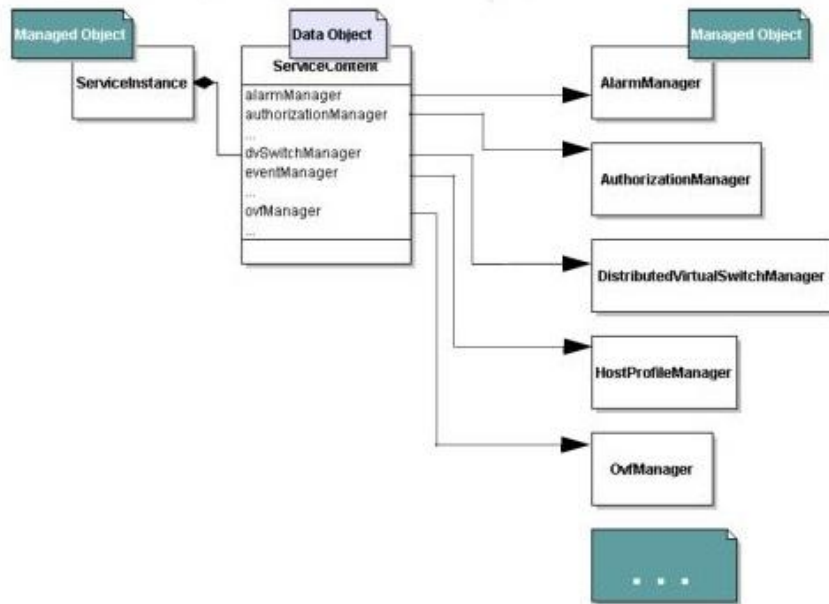
I vSphere Web Services conservano lo stato della sessione per ogni connessione client, inserendo un token nell'header delle request http per identificare la sessione stessa. Il vSphere Server invia il token al client in risposta alla richiesta di connessione e lo invia nelle successive richieste.

## VI.5. Accesso ai Managed Object

L'applicazione client ottiene l'accesso ai Managed object attraverso l'oggetto ServiceInstance e l'oggetto associato ServiceContent. L'oggetto ServiceContent contiene i riferimenti alle altre entità quali servizi manager e alla directory root dell'inventario.

L'oggetto ServiceInstance rappresenta l'oggetto root dell'inventario sia per i server ESX/ESXi che per i server vCenter e funziona quindi da punto di accesso primario all'inventario. I server creano l'oggetto ServiceInstance e le entità manager che forniscono i servizi all'infrastruttura quali PropertyCollector e PerformanceManager.

La seguente figura mostra il modello delle classi ServiceInstance e ServiceContent.



**Figura A-0-5: ManagedObjectReference Data Object**

Mediante i Managed object reference è possibile conoscere le informazioni sullo stato degli oggetti dell’inventario server-side e popolare gli oggetti client-side. Per accedere agli oggetti gestiti sono possibili i seguenti approcci:

- Utilizzare i metodi accessori getter. Tali metodi sono disponibili per ogni proprietà relativa agli oggetti.
- Utilizzare l’oggetto PropertyCollector per raggiungere un determinato oggetto sul server e recuperare delle specifiche proprietà.
- Utilizzare l’oggetto SearchIndex per recuperare il Managed object reference di una entità. L’oggetto SearchIndex è in grado di recuperare il Managed object reference di ComputeResource, Datacenter, Folder, HostSystem, ResourcePool, VirtualMachine, dato il percorso, l’indirizzo IP o il DNS name.

Le proprietà degli oggetti possono essere dei seguenti tipi:

- Tipi di dati semplici quali string, boolean, integer.
- Array di dati di tipo semplice quali array di Managed object reference.
- Tipi di dato enumerati di valori predefiniti quali *poweredOn*, *poweredOff*, e *suspended*.
- Tipi di dato complessi (o compositi). Ad esempio dati contenenti array di oggetti, array di oggetti annidati.

L’accesso ai tipi di dato complessi è possibile nei seguenti modi:

- Utilizzare la notazione puntata per accedere alle proprietà annidate.
- Eseguire il cast degli oggetti con array.

- Utilizzare chiavi o indici per accedere ai valori degli array.

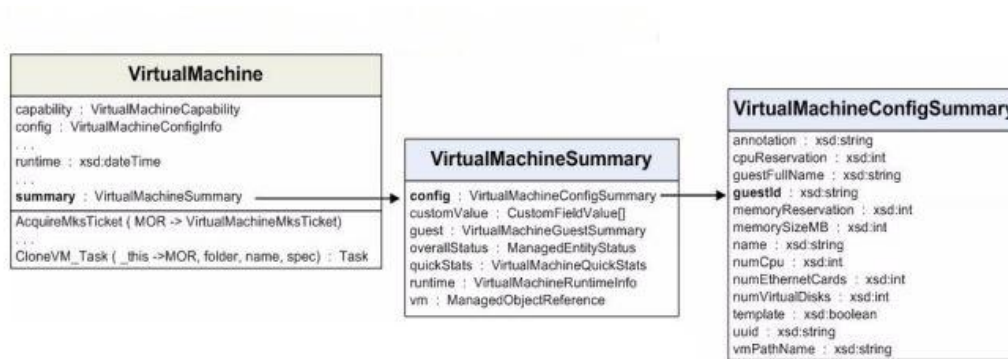


Figura A-0-6: Esempio di oggetti annidati

## VI.6. Datacenter Inventory

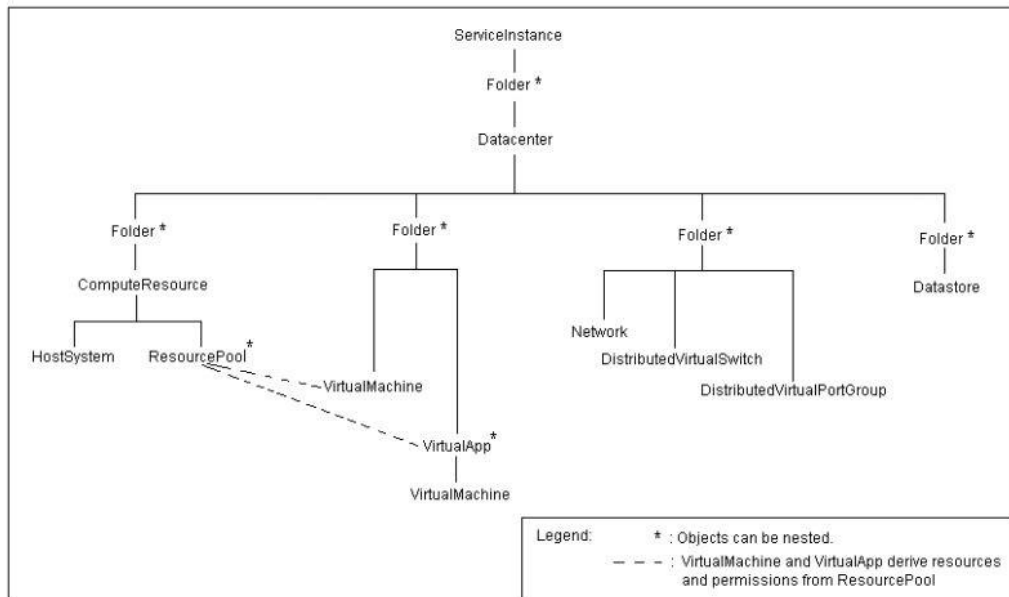
L'inventario di vSphere è una rappresentazione del datacenter e degli oggetti in esso contenuti. Gli oggetti appartengono ad una delle seguenti categorie:

- Componenti del datacenter: Host, VirtualMachine e VirtualApp.
- Componenti di supporto: ComputeResource, Datastore, Network e virtual devices.
- Componenti organizzativi: Folder e Datacenter.

Per accedere agli oggetti dell'infrastruttura il punto di accesso è il ServiceInstance che rappresenta l'oggetto root dell'infrastruttura. Da questo è possibile navigare all'interno della struttura gerarchica. La struttura gerarchica differisce leggermente per le due tipologie di server ESX/ESXi e vCenter.

### Inventario dei vCenter Server

Nella seguente figura è rappresentata la struttura gerarchica dell'inventario di un vCenter Server:



**Figura A-0-7: Struttura dell’inventario di un vCenter Server**

Il vCenter Server consente la creazione di più datacenter. Per ogni datacenter vengono create le seguenti directory:

- Una directory per VirtualMachine, template, e VirtualApp.
- Una directory per una gerarchia ComputeResource.
- Una directory per Network, DistributedVirtualSwitch, e DistributedVirtualPortgroup.
- Una directory per Datastore.

Per infrastrutture di grandi dimensioni è possibile organizzare l’inventario mediante l’utilizzo di più directory annidate.

### **Inventario degli ESX/ESXi server**

Nella seguente figura è rappresentata la struttura gerarchica dell’inventario di un server ESX/ESXi:

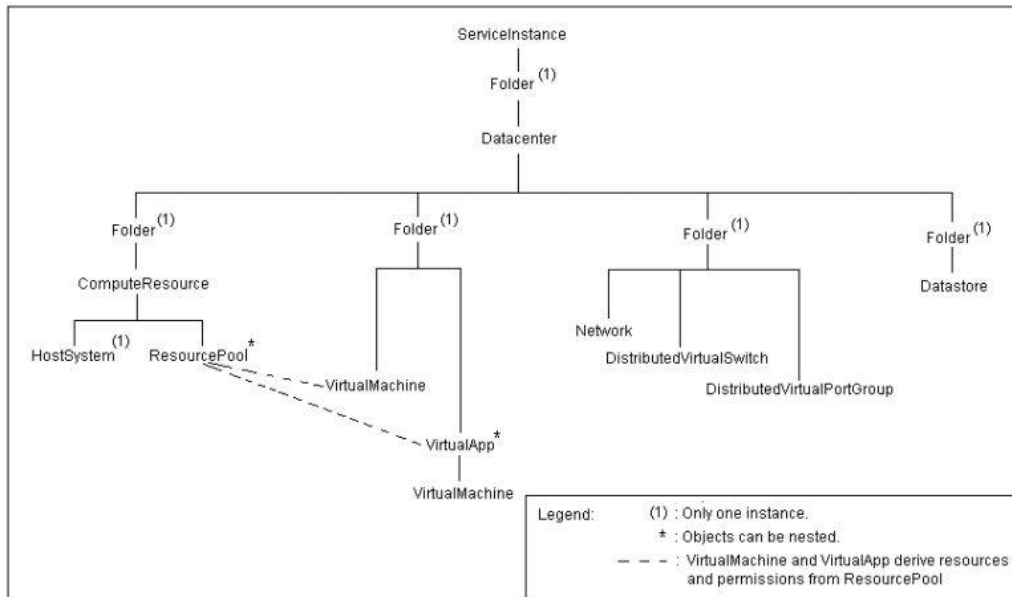


Figura A-0-8: Struttura dell'inventario di un Server ESX/ESXi

I server ESX/ESXi, a differenza dei server vCenter, supportano un solo datacenter all'interno dell'infrastruttura.

### Accesso agli oggetti dell'inventario

L'accesso agli oggetti dell'inventario avviene attraverso l'oggetto TraversalSpec in congiunzione con l'oggetto PropertyCollector. Gli oggetti possiedono i seguenti attributi:

- **Parent:** identifica l'oggetto padre.
- **childEntity:** identifica gli oggetti contenuti dagli oggetti di tipo Folder.

### Property Collector

L'oggetto PropertyCollector consente l'accesso ai dati e alle proprietà degli oggetti dell'inventario. Per il recupero delle proprietà PropertyCollector utilizza i seguenti filtri:

- Punto di accesso alla struttura dell'inventario.
- Inventory traversal path.
- MOR dell'oggetto e proprietà da recuperare.

Un oggetto PropertyCollector viene creato all'inizio di ogni sessione e lo stesso consente la creazione di ulteriori oggetti per l'utilizzo parallelo. Sono disponibili due modalità di recupero dati e proprietà degli oggetti:

- **Property retrieval:** come singola operazione utilizzando i metodi RetrievePropertiesEx e ContinueRetrievePropertiesEx.

- **Incremental property retrieval:** utilizza il metodo WaitForUpdatesEx. Chiamate successive al metodo consentono di ottenere le variazioni avvenute dall'ultima operazione di recupero. Questa modalità è utilizzata per il monitoraggio delle variazioni delle proprietà degli oggetti.

La seguente figura mostra i metodi della classe PropertyCollector:

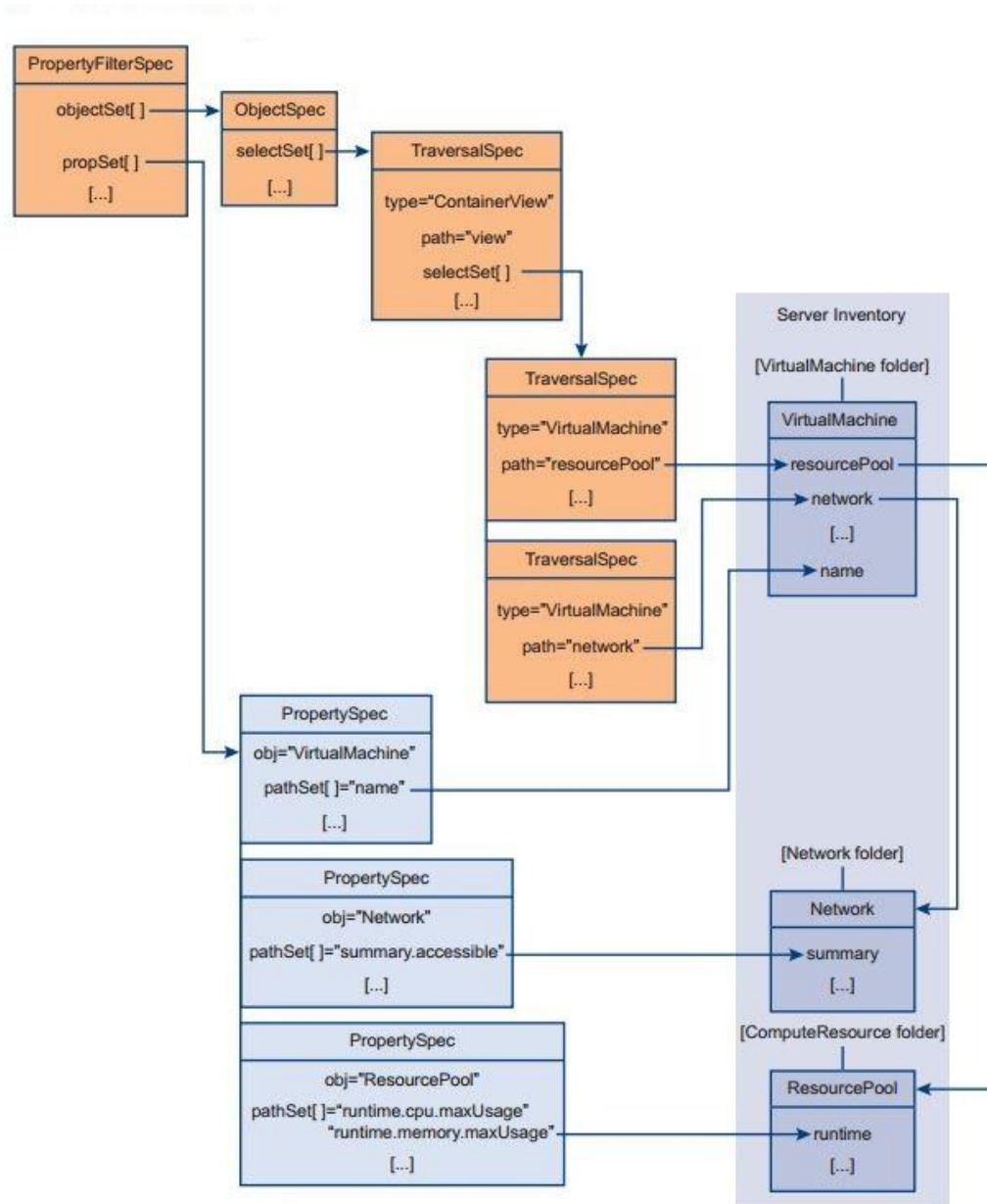


Figura A-0-9: Metodi della classe PropertyCollector

I passi necessari per l'utilizzo dell'oggetto PropertyCollector per una singola operazione di recupero dati e proprietà, nella fattispecie per il recupero della lista delle virtual machine del datacenter, sono i seguenti:



1 - Ottenere i riferimenti agli oggetti ViewManager e PropertyCollector.

```
ManagedObjectReference viewMgrRef = sContent.getViewManager();
ManagedObjectReference propColl = sContent.getPropertyCollector();
```

2 - Creare un container view per la memorizzazione dei riferimenti delle virtual machine.

```
List<String> vmList = new ArrayList<String>();
vmList.add("VirtualMachine");
ManagedObjectReference cViewRef = methods.createContainerView(viewMgrRef,
sContent.getRootFolder(), vmList, true );
```

3 - Creare un oggetto ObjectSpec con le specifiche del punto di accesso alla gerarchia dell'inventario.

```
ObjectSpec oSpec = new ObjectSpec();
oSpec.setObj(cViewRef);
oSpec.setSkip(true);
```

4 - Creare le specifiche di attraversamento TraversalSpec dell'inventario per raggiungere gli oggetti virtual machine all'interno della struttura gerarchica.

```
TraversalSpec tSpec = new TraversalSpec();
tSpec.setName("traverseEntities");
tSpec.setPath("view");
tSpec.setSkip(false);
tSpec.setType("ContainerView");
```

5 - Aggiungere le specifiche TraversalSpec all'oggetto ObjectSpec.

```
oSpec.getSelectSet().add(tSpec);
```

6 - Identificare le proprietà da recuperare (nel nostro caso VirtualMachine.name).

```
PropertySpec pSpec = new PropertySpec();
pSpec.setType("VirtualMachine");
pSpec.getPathSet().add("name");
```

7 - Aggiungere all'oggetto PropertyFilterSpec le specifiche dell'oggetto di cui recuperare le proprietà.

```
PropertyFilterSpec fSpec = new PropertyFilterSpec();
fSpec.getObjectSet().add(oSpec);
fSpec.getPropSet().add(pSpec);
```

8 - Creare la lista dei filtri con le relative specifiche.

```
List<PropertyFilterSpec> fSpecList = new ArrayList<PropertyFilterSpec>();
fSpecList.add(fSpec);
```

9 - Recupero delle proprietà (nomi delle virtual machine).

```
RetrieveOptions ro = new RetrieveOptions();
RetrieveResult props = methods.retrievePropertiesEx(propColl,fSpecList,ro);
```

10 - Stampa a video della lista delle virtual machine.

```
if (props != null) {
    for (ObjectContent oc : props.getObjects()) {
        String vmName = null;
        String path = null;
        List<DynamicProperty> dps = oc.getPropSet();
        if (dps != null) {
            for (DynamicProperty dp : dps) {
                vmName = (String) dp.getVal();
                path = dp.getName();
                System.out.println(path + " = " + vmName);
            }
        }
    }
}
```

La seguente figura mostra gli oggetti utilizzati nell'esempio precedente:

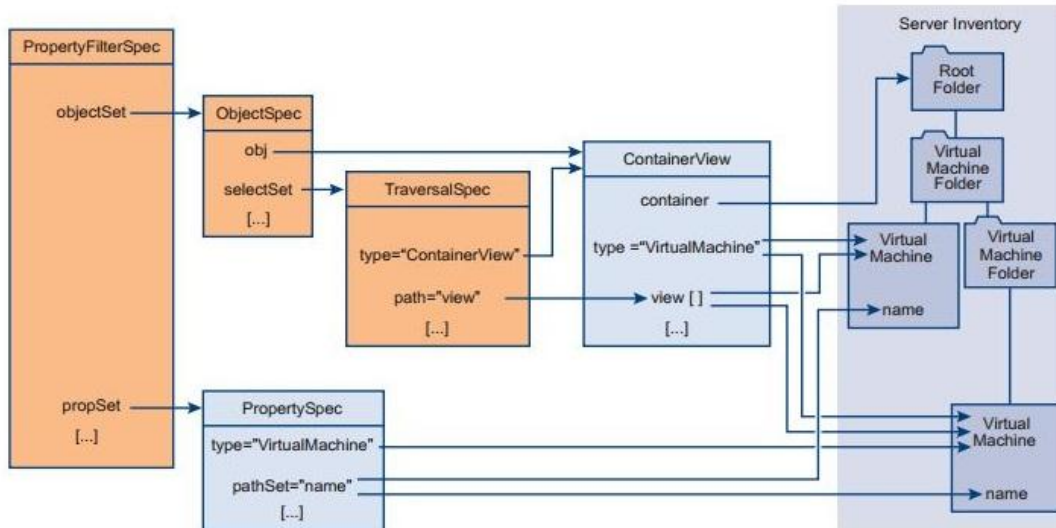


Figura A-0-10: Property Filter Specification per il recupero della lista delle Virtual Machine

### Navigazione dell'inventario

L'esempio precedente utilizza l'oggetto **ContainerView** come oggetto iniziale sul quale iniziare il processo di recupero delle proprietà. Per selezionare un oggetto dell'inventario è quindi necessario impostare un filtro che include l'oggetto **TraversalSpec** e, opzionalmente, l'oggetto **SelectionSpec**. E' anche disponibile l'oggetto **SearchIndex** che fornisce una serie di metodi per il recupero dei riferimenti degli oggetti dell'inventario in base ad un serie di filtri quali il percorso dell'oggetto, l'indirizzo IP, il DNS name ecc.

### TraversalSpec Traversal

L'oggetto **TraversalSpec** è usato per identificare la tipologia dell'oggetto da cercare e le sue proprietà. L'oggetto contiene le seguenti proprietà:

- **type**: identifica la tipologia dell'oggetto dell'inventario.
- **path**: identifica il percorso di un oggetto.
- **selectSet**: specifica una lista opzionale di ulteriori oggetti a partire da quello corrente.
- **skip**: indica se recuperare o meno le proprietà dell'oggetto corrente.

### SelectionSpec Traversal

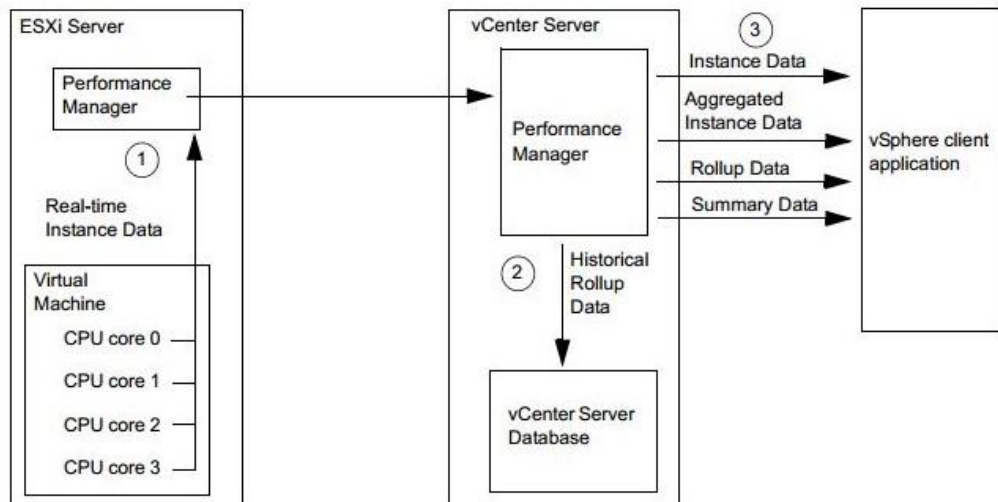
**SelectionSpec** definisce il nome delle proprietà. Utilizzando i riferimenti agli oggetti **SelectionSpec** è possibile riutilizzare gli oggetti **TraversalSpec** e definire la navigazione ricorsiva all'interno della struttura gerarchica per gli oggetti annidati.

## VI.7. vSphere Performance

I server vSphere utilizzano delle metriche di performance per tracciare l'utilizzo delle risorse. A livello di runtime, i diversi componenti dell'infrastruttura generano dati di performance che i server memorizzano nel database. Più precisamente i server ESXi forniscono i dati in real-time e i server vCenter li memorizzano nel database interno. vCenter conserva i dati storici con campionamento variabile in funzione della singola metrica. Per quanto riguarda la frequenza di recupero dei dati si hanno:

- Real-time data collection: i server ESXi raccolgono i dati di performance ogni 20 secondi e li conservano per un ora.
- Historical data rollup : i server vCenter raccolgono i dati da tutti gli host che gestiscono e li memorizzano nel database secondo il parametro *performance counter interval*.

Il modello di recupero dei dati di performance è mostrato in figura:



**Figura A-0-11: Modello di recupero dei dati di performance di vSphere**

- 1 - I server ESXi campionano i dati di performance ogni 20 secondi e li conservano per un ora.
- 2 - Il server vCenter raccoglie i dati dagli host che gestisce e li memorizza nel database interno.
- 3 - Le applicazioni client possono recuperare sia i dati real-time sia quelli storici.

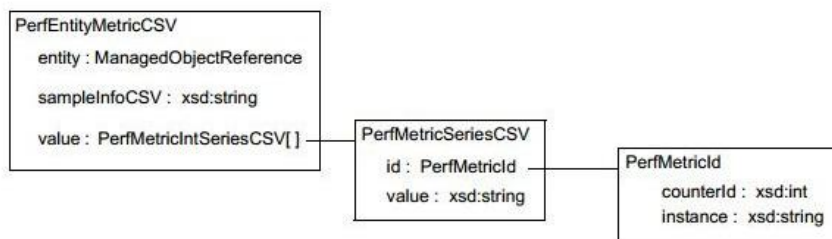
### Recupero dei dati di performance

Per recuperare i dati sulle metriche di performance, le API di vSphere mettono a disposizione la classe PerformanceManager. Nello specifico è necessario impostare una query contenente uno o più oggetti PerfQuerySpec. Ogni oggetto PerfQuerySpec identifica i seguenti parametri:

- PerfQuerySpec.entity: specifica l'entità di cui sono forniti i dati di performance.
- PerfQuerySpec.metricId: specifica la metrica di performance.
- PerfQuerySpec.intervalId – specifica l'intervallo di campionamento dei dati.
- PerfQuerySpec.startTime, PerfQuerySpec.endTime: tempo iniziale e finale di campionamento dei dati.
- PerfQuerySpec.maxSample: quantità di dati da recuperare.
- PerfQuerySpec.format: formato dell'output. Può essere di due tipologie:
  - Normal: valori restituiti come oggetti di dati.
  - Formatted output: valori restituiti nel formato CSV.

La combinazione di entità e metricID determina il set di metriche disponibili. La combinazione di intervallo, tempo iniziale e tempo finale, quantità di valori, consente il recupero dei dati in maniera aggregata o come dati storici.

La seguente figura mostra la differente classificazione dei dati di performance che possono essere recuperati da un vCenter server:



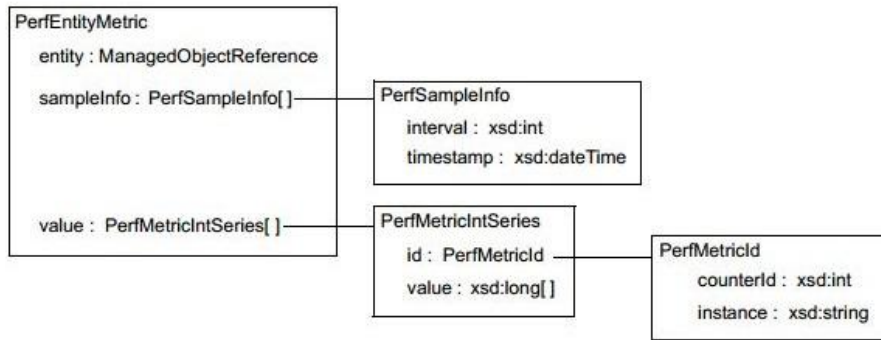
**Figura A-0-12: Classificazione delle metriche di performance per intervallo di recupero**

### Format di output “normal”

Quando viene specificato il formato di output “normal”, ogni PerfEntityMetric restituito contiene le seguenti proprietà:

- entity: riferimento all'entità che ha prodotto i dati di performance.
- sampleInfo: array contenente le informazioni sui dati campionati.
- value: array di oggetti che contengono le seguenti proprietà:
  - id: identifica la metrica di performance metric ID.
  - value: array di interi che corrispondono agli array dei dati (PerfEntityMetric.sampleInfo).

La seguente figura mostra la rappresentazione della gerarchia degli oggetti ottenuti specificando il formato “normal”



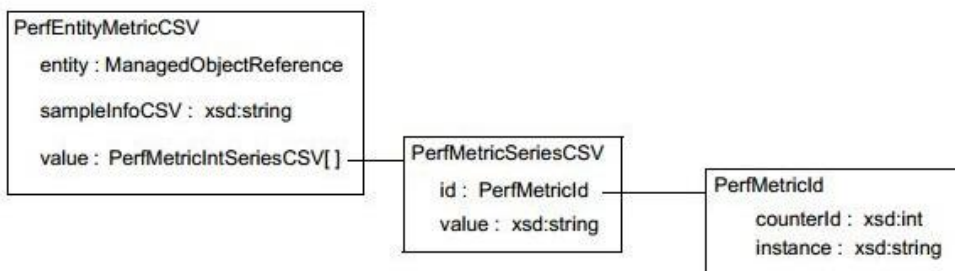
**Figura A-0-13: Oggetto PerfEntityMetric - Formato di output “normal”**

**Formato di output “CSV”**

Quando viene specificato il formato di output “CSV”, sia le informazioni sul campionamento dei dati sia i dati stessi vengono codificati come valori separati da virgola (CSV). L’oggetto PerfEntityMetricCSV utilizzato contiene le seguenti proprietà:

- entity: riferimento all’entità che ha prodotto i dati di performance.
- sampleInfoCSV: stringa contenente le informazioni sui dati campionati.
- value: array di oggetti che contengono le seguenti proprietà:
  - id: identifica la metrica di performance metric ID.
  - value: set di valori campionati in formato CSV, corrispondenti alla lista dei dati di performance (PerfEntityMetricCSV.sampleInfoCSV).

La seguente figura mostra la rappresentazione della gerarchia degli oggetti ottenuti specificando il formato “CSV”



**Figura A-0-14: PerfEntityMetricCSV - Formato di output “CSV”**

### Performance Counter Metadata

Le metriche di performance sono organizzate in gruppi di risorse quali CPU, MEM e disk. vSphere API Reference contiene una tabella per ogni gruppo che include il nome, il tipo di metrica, l'unità di misura ed altre proprietà. L'oggetto PerfCounterInfo identifica le proprietà delle metriche rilevate come mostrato in figura:

Property	Description
groupInfo	Name of the resource group to which this counter belongs, such as disk, cpu, or memory.
key	Unique integer that identifies the counter. Also called the counter ID. The value is unique and it is not static—it might, for example, change between system reboots. The counter key on an ESXi system might not be the same as the counter key for the same counter on the vCenter Server system managing the ESXi system. However, the system maps the keys from ESXi to vCenter Server systems automatically.
level	Number from 1 to 4 that identifies the level at which data values for this counter are aggregated.
nameInfo	Descriptive name for the counter. The name component of a fully qualified counter name, for example "granted" is the nameInfo property for the mem.granted.AVERAGE counter.
rollupType	Indicates how multiple samples of a counter are transformed into a single statistical value. Examples of rollup types are average, summation, and minimum. No conversion of values occurs for counters that specify absolute values, such as the total number of seconds that the system has been running continuously since startup. The PerfSummaryType is an enumeration containing valid constants for this property.
statsType	Type of statistical data that the value represents over the course of the interval, such as an average, a rate, the minimum value, and so on. The PerfStatsType is an enumeration containing valid constants for this property.
unitInfo	Unit of measure, such as megahertz, kilobytes, kilobytes per second, and so on. The ElementDescription's key property is populated using one of the constants available in the PerformanceManagerUnit enumeration.

Figura A-0-15: Proprietà dell'oggetto PerfCounterInfo

### Intervallo di campionamento delle performance per ESXi Server

Come già detto, i server ESXi recuperano i dati sulle metriche di performance ogni 20 secondi. L'array PerformanceManager.historicalInterval per un server ESXi contiene un singolo oggetto PerfInterval che specifica i dati raccolti ogni 5 minuti. Questi dati non possono essere recuperati direttamente dal server ESXi ma possono essere recuperati dal server vCenter che gestisce l'host. Nella seguente figura sono mostrate le proprietà dei dati storici per un server ESXi.

Property	Value	Description
key	1	Numeric identifier for the PerfInterval.
name	PastDay	Name of the PerfInterval.
samplingPeriod	300	Time interval between data sampling events.
length	129600	Number of seconds that statistics associated with the interval are kept by the vCenter Server.
enabled	true	This PerfInterval is enabled on the system.
level	null	Statistics collection level. For an ESXi system, this property is null. The PerfInterval object on an ESXi system defines the baseline interval.

Figura A-0-16: Proprietà dell'oggetto PerfInterval DataObject - ESXi Server

### Intervallo di campionamento delle performance per vCenter Server

La quantità di dati di performance recuperati dal server vCenter dipendono dalla configurazione del parametro *level* associato alle singole metriche. Tale parametro è visibile nelle proprietà PerformanceManager.historicalInterval, costituito a sua volta da un array di oggetti PerfInterval che definisce quattro diversi livelli numerici, dal valore 1 al valore 4. La lista dei valori di default per i diversi periodi di campionamento è disponibile nella seguente figura:

Key	Name	Sampling Period	Length	Enabled	Level
1	Past Day	300	86400	TRUE	1
2	Past Week	1800	604800	TRUE	1
3	Past Month	7200	2592000	TRUE	1
4	Past Year	86400	31536000	TRUE	1

**Figura A-0-17: Proprietà dell'oggetto PerfInterval Data per un vCenter Server**

Per default il parametro *level* assume il valore 1. Utilizzando tale valore il vCenter server memorizza nel database i dati secondo i seguenti periodi di campionamento:

- campionamento di 5 minuti per i dati dell'ultimo giorno.
- campionamento di 30 minuti per i dati dell'ultima settimana.
- campionamento di 2 ore per i dati dell'ultimo mese.
- campionamento di 1 giorno per i dati dell'ultimo anno.

I dati più vecchi di un anno sono eliminate dal database.



# Bibliografia

- [BCK03] L.Bass, P.Clements, R.Kazman, "Software architecture in practice (2nd Ed.)", Addison Wesley, 2003
- [ENI12] "Cloud Computing. Benefits, risks and recommendations for information security", ENISA, 2012
- [FJ04] J. Falkner and K. Jones, "Servlets and JavaServer Pages: The J2EE Technology Web Tier", Addison Wesley, 2004
- [G07] D.Gilbert, "The JFreeChart Class Library Developer Guide", Object Refinery Limited, 2007
- [G08] D.Gilbert, "The JFreeChart Class Library Installation Guide", Object Refinery Limited, 2008
- [JCEGHMS] E.Jendrock, R.Cervera-Navarro, I.Evans, D.Gollapudi, K.Haase, W.Markito, C.Srivathsa, "The Java EE 7 Tutorial, Release 7", Oracle, 2013
- [LR01] A.Leff, T.Rayfield, "Web-Application Development Using the Model/View/Controller Design Pattern", IBM Research Division, 2001
- [MG11] P. Mell, T. Grance, "The NIST definition of Cloud computing", NIST, 2011
- [OR13a] "MySQL 5.6 Reference Manual", Oracle, 2013
- [OR13b] "MySQL Workbench", Oracle 2013
- [P05] P.Patierno, "Framework MVC per lo sviluppo di Web Application: JavaServer Faces e Struts", 2005
- [SR03] L.Shklar, R. Rosen, "Web Application Architecture: Principles, Protocols and Practices", Addison Wesley, 2004
- [TASF13a] "Apache Tomcat Architecture", The Apache Software Foundation, 2003
- [TASF13b] "Apache Tomcat Server Configuration Reference", The Apache Software Foundation, 2003
- [TB02] J.Turner, K.Bedell, "Struts Kick Start", SAMS, 2002
- [VEE13] "Veeam ONE Version 7.0 User Guide for VMware vSphere Environments", Veeam, 2013
- [VMW08] "Building the Virtualize Enterprise with VMware Infrastructure", VMware, 2008
- [VMW09] "VMware ESX and ESXi", VMware, 2009
- [VMW09c] "VMware vStorage VMFS", VMware, 2009
- [VMW11] "VMware vSphere Basics", VMware, 2011
- [VMW11b] "vSphere Availability", VMware, 2011
- [VMW11c] "vSphere Resource Management", VMware, 2011
- [VMW11d] "vSphere Monitoring and Performance", VMware, 2011
- [VMW11e] "What's New in VMware vSphere 5.0 – Storage", VMware, 2011
- [VMW11f] "vSphere Web Services SDK Programming Guide", VMware, 2011
- [VMW13] "VMware vFabric Hyperic Overview", VMware, 2013

# Indice delle figure

Figura I-1: Modelli di Cloud Computing .....	9
Figura I-2: Modello SPI .....	10
Figura I-3: Confronto modelli SPI.....	11
Figura I-4: Sfruttamento dei vantaggi economici del Cloud .....	13
Figura I-5: Infrastruttura di VMware vSphere.....	14
Figura I-6: vSMP esempio di assegnazione dei processori alle macchine virtuali .....	15
Figura I-7: Architettura VMFS.....	15
Figura I-8: Architettura VMware vCenter.....	16
Figura I-9: Architettura Veeam One .....	20
Figura I-10: Architettura Hyperic Server .....	21
Figura I-11: Architettura Hyperic Agent.....	22
Figura II-1: Diagramma dei casi d'uso .....	27
Figura II-2: Indice dei casi d'uso.....	28
Figura II-3: Use Case "Login".....	29
Figura II-4: Use Case "Logout".....	29
Figura II-5: Use Case "Configura" .....	29
Figura II-6: Use Case "Start monitoring e alarming" .....	30
Figura II-7: Use Case "Stop monitoring e alarming".....	30
Figura II-8: Use Case "Aggiorna inventario" .....	31
Figura II-9: Use Case "Seleziona un elemento dell'inventario".....	31
Figura II-10: Use Case "Scelta parametri per visualizzazione metriche statistiche" .....	32
Figura II-11: Use Case "Visualizza statistiche" .....	32
Figura II-12: Use Case "Visualizza proprietà" .....	32
Figura II-13: Use Case "Inserimento allarme" .....	33
Figura II-14: Use Case "Visualizza allarmi per un elemento" .....	33
Figura II-15; Use Case "Visualizza riepilogo allarmi" .....	34
Figura II-16: Use Case "Elimina allarme" .....	34
Figura III-1: Model, View e Controller.....	36
Figura III-2: MVC Model 2 (Servlet - Centric Architecture).....	36
Figura III-3: Diagramma degli stati per un thread .....	40
Figura III-4: Diagramma di sequenza gestione dei thread dell'applicazione .....	41
Figura III-5: Diagramma di sequenza della classe MonitorDataThread().....	42
Figura III-6: Diagramma di sequenza della classe AlarmThread().....	43

Figura III-7: Diagramma E-R del database .....	46
Figura III-8: Tabella alarm.....	47
Figura III-9: Tabella ccrperformance.....	48
Figura III-10: Tabella condition .....	48
Figura III-11: Tabella dsperformance .....	49
Figura III-12: Tabella hperformance .....	50
Figura III-13: Tabella item.....	51
Figura III-14: Tabella itemtype .....	51
Figura III-15: Tabella metric .....	51
Figura III-16: Tabella vmperformance .....	52
Figura III-17: vmstorageperformance.....	53
Figura III-18: Classi del package tesi.servlet .....	54
Figura III-19: Diagramma UML della classe SystemListener .....	54
Figura III-20: Diagramma UML della classe SystemServlet .....	55
Figura III-21: Classi del package tesi.chart.....	56
Figura III-22: Diagramma UML della classe Chart.....	57
Figura III-23: Diagramma UML della classe ChartUtil.....	58
Figura III-24: Diagramma UML della classe DCDelement .....	59
Figura III-25: Diagramma UML della classe DPDelement .....	59
Figura III-26: Diagramma UML della classe TimestampSplit.....	60
Figura III-27: Classi del package tesi.alarm.....	60
Figura III-28: Diagramma UML della classe Alarm .....	61
Figura III-29: Diagramma UML della classe AlarmDbDAO .....	62
Figura III-30: Diagramma UML della classe AlarmDbManager .....	62
Figura III-31: Classi del package tesi. db.ccrperformance.....	63
Figura III-32: Diagramma UML della classe CcrPerformance .....	63
Figura III-33: Diagramma UML della classe CcrPerformanceDAO .....	64
Figura III-34: Diagramma UML della classe CcrPerformanceManager.....	64
Figura III-35: Classi del package tesi.db.condition .....	65
Figura III-36: Diagramma UML della classe Condition.....	65
Figura III-37: Diagramma UML della classe ConditionDAO.....	66
Figura III-38: Diagramma UML della classe ConditionManager .....	66
Figura III-39: Classi del package tesi.db.dsperformance .....	67
Figura III-40: Diagramma UML della classe DsPerformance .....	67
Figura III-41: Diagramma UML della classe DsPerformanceDAO .....	68
Figura III-42: Diagramma UML della classe DsPerformanceManager.....	68

Figura III-43: Classi del package tesi.db.hsperformance .....	69
Figura III-44: Diagramma UML della classe HsPerformance.....	70
Figura III-45: Diagramma UML della classe HsPerformanceDAO .....	71
Figura III-46: Diagramma UML della classe HsPerformanceManager .....	71
Figura III-47: Classi del package tesi.db.inventory .....	72
Figura III-48: Diagramma UML della classe InventoryDbDAO.....	72
Figura III-49: Diagramma UML della classe InventoryDbManager .....	73
Figura III-50: Diagramma UML della classe Item .....	73
Figura III-51: Classi del package tesi.db.metric .....	74
Figura III-52: Diagramma UML della classe Metric .....	74
Figura III-53: Diagramma UML della classe MetricDAO.....	75
Figura III-54: Diagramma UML della classe MetricManager.....	75
Figura III-55: Classi del package tesi.db.vmpformance.....	76
Figura III-56: Diagramma UML della classe VmPerformance .....	76
Figura III-57: Diagramma UML della classe VmPerformanceDAO.....	77
Figura III-58: Diagramma UML della classe VmPerformanceManager.....	77
Figura III-59: Classi del package tesi.db.vmsperformance .....	78
Figura III-60: Diagramma UML della classe VmsPerformance.....	78
Figura III-61: Diagramma UML della classe VmsPerformanceDAO .....	79
Figura III-62: Diagramma UML della classe VmsPerformanceManager .....	79
Figura III-63: Classi del package tesi.directory .....	80
Figura III-64: Diagramma UML della classe DirUtil .....	80
Figura III-65: Classi del package tesi.mail.....	80
Figura III-66: Diagramma UML della classe AlarmMail .....	81
Figura III-67: Diagramma UML della classe MailSender.....	81
Figura III-68: Classi del package tesi.monitoring.....	82
Figura III-69: Diagramma UML della classe AlarmThread.....	82
Figura III-70: Diagramma UML della classe MonitorDataThread.....	83
Figura III-71: Diagramma UML della classe MonitorManager .....	84
Figura III-72: Diagramma UML della classe MonitorThreadManager .....	84
Figura III-73: Classi del package tesi.security .....	85
Figura III-74: Diagramma UML della classe DesEncrypter .....	85
Figura III-75: Diagramma UML della classe TDesEncrypter .....	86
Figura III-76: Classi del package tesi.vmware .....	86
Figura III-77: Diagramma UML della classe PropertyManager.....	87
Figura III-78: Diagramma UML della classe TrustAllTrustManager.....	87

Figura III-79: Diagramma UML della classe VsphereAccess.....	88
Figura III-80: Diagramma UML della classe VsphereManager .....	89
Figura III-81: Diagramma UML della classe VsphereUtil.....	89
Figura III-82: Classi del package tesi.vmware.beans .....	90
Figura III-83: Diagramma UML della classe CcrProperty .....	91
Figura III-84: Diagramma UML della classe DcProperty.....	92
Figura III-85: Diagramma UML della classe DsProperty.....	93
Figura III-86: Diagramma UML della classe GeneralInfo.....	94
Figura III-87: Diagramma UML della classe HsProperty .....	95
Figura III-88: Diagramma UML della classe InventoryItem .....	96
Figura III-89: Diagramma UML della classe ItemType .....	96
Figura III-90: Diagramma UML della classe VmProperty .....	97
Figura III-91: Classi del package tesi.xml .....	98
Figura III-92: Diagramma UML della classe IsInstalled .....	98
Figura III-93: Diagramma UML della classe MailConfigManager.....	99
Figura III-94: Diagramma UML della classe MonitoringConfigManager.....	99
Figura III-95: Diagramma UML della classe MySqlConfigManager.....	100
Figura III-96: Diagramma UML della classe SntpConfigManager .....	100
Figura III-97: Diagramma UML della classe VcenterConfigManager.....	100
Figura III-98: Diagramma UML della classe WebappAccountManager .....	101
Figura III-99: Diagramma UML della classe XmlUtil .....	102
Figura III-100: Classi del package tesi.xml.beans.....	103
Figura III-101: Diagramma UML della classe MailConfig .....	103
Figura III-102: Diagramma UML della classe MonitoringConfig .....	104
Figura III-103: Diagramma UML della classe MySqlConfig .....	104
Figura III-104: Diagramma UML della classe SntpConfig.....	105
Figura III-105: Diagramma UML della classe VcenterConfig.....	105
Figura III-106: Diagramma UML della classe WebappAccount.....	106
Figura III-107: Esempio di visualizzazione della pagina alarm.jsp.....	107
Figura III-108: Esempio di visualizzazione della pagina alarmPopup.jsp .....	107
Figura III-109: Esempio di visualizzazione della pagina ccr.jsp.....	108
Figura III-110: Esempio di visualizzazione della pagina ccrPerformance.jsp .....	109
Figura III-111: Esempio di visualizzazione della pagina ccrProperties.jsp .....	110
Figura III-112: Esempio di visualizzazione della pagina config.jsp.....	111
Figura III-113: Esempio di visualizzazione della pagina dc.jsp.....	112
Figura III-114: Esempio di visualizzazione della pagina dcProperties.jsp .....	113

Figura III-115: Esempio di visualizzazione della pagina ds.jsp .....	114
Figura III-116: Esempio di visualizzazione della pagina dsPerformance.jsp.....	115
Figura III-117: Esempio di visualizzazione della pagina dsProperties.jsp.....	116
Figura III-118: Esempio di visualizzazione della pagina footer.jsp .....	116
Figura III-119: Esempio di visualizzazione della pagina generalInfo.jsp .....	117
Figura III-120: Esempio di visualizzazione della pagina header.jsp .....	117
Figura III-121: Esempio di visualizzazione della pagina home.jsp .....	118
Figura III-122: Esempio di visualizzazione della pagina hs.jsp .....	119
Figura III-123: Esempio di visualizzazione della pagina hsPerformance.jsp.....	120
Figura III-124: Esempio di visualizzazione della pagina hsProperties.jsp.....	121
Figura III-125: Visualizzazione della sezione comandi della pagina inventarioTree.jsp.....	121
Figura III-126: Esempio di visualizzazione della pagina inventarioTree.jsp .....	122
Figura III-127: Esempio di visualizzazione della pagina login.jsp.....	122
Figura III-128: Esempio di visualizzazione della pagina loginError.jsp.....	123
Figura III-129: Esempio di visualizzazione della pagina message.jsp.....	123
Figura III-130: Esempio di visualizzazione della pagina vm.jsp.....	124
Figura III-131: Esempio di visualizzazione della pagina vmPerformance.jsp .....	125
Figura III-132: Esempio di visualizzazione della pagina vmProperties.jsp .....	126
Figura IV-1: Tipologie di memoria di una JVM.....	128
Figura IV-2: Finestra Apache Tomcat Properties .....	130
Figura IV-3: Screenshot MySQL Installer (1).....	131
Figura IV-4: Screenshot MySQL Installer (2).....	131
Figura IV-5: Screenshot MySQL Installer (3).....	132
Figura IV-6: Screenshot MySQL Installer (4).....	132
Figura IV-7: Screenshot MySQL Installer (5).....	133
Figura IV-8: Screenshot MySQL Installer (6).....	134
Figura IV-9: Screenshot MySQL Installer (7).....	135
Figura IV-10: Tomcat Manager App .....	136
Figura V-1: Tempo di recupero delle metriche di performance in funzione del numero di Virtual Machine attive .....	140
Figura A-0-1: vSphere Server and Client.....	202
Figura A-0-2: Client-Server Communication Through a Client Proxy Interface .....	203
Figura A-0-3: Mappatura di alcuni Data Type nei rispettivi dei linguaggi di programmazione Java e C# .....	204
Figura A-0-4: Autenticazione dei server vSphere ESX/ESXi e vCenter .....	205
Figura A-0-5: ManagedObjectReference Data Object.....	207

Figura A-0-6: Esempio di oggetti annidati.....	208
Figura A-0-7: Struttura dell'inventario di un vCenter Server .....	209
Figura A-0-8: Struttura dell'inventario di un Server ESX/ESXi.....	210
Figura A-0-9: Metodi della classe PropertyCollector .....	211
Figura A-0-10: Property Filter Specification per il recupero della lista delle Virtual Machine .....	214
Figura A-0-11: Modello di recupero dei dati di performance di vSphere.....	215
Figura A-0-12: Classificazione delle metriche di performance per intervallo di recupero .....	216
Figura A-0-13: Oggetto PerfEntityMetric - Formato di output "normal" .....	217
Figura A-0-14: PerfEntityMetricCSV - Formato di output "CSV" .....	217
Figura A-0-15: Proprietà dell'oggetto PerfCounterInfo.....	218
Figura A-0-16: Proprietà dell'oggetto PerfInterval DataObject - ESXi Server.....	218
Figura A-0-17: Proprietà dell'oggetto PerfInterval Data per un vCenter Server.....	219

# Indice delle tabelle

Tabella I-1: Campionamento dei dati di performance storici di VMware vSphere .....	19
Tabella I-2: Comparazione delle caratteristiche generali dei software .....	23
Tabella I-3: Comparazione delle caratteristiche di monitoraggio dei software .....	24
Tabella V-1: Tempo di recupero metriche di performance in funzione del numero di Virtual Machine attive.....	140
Tabella V-2: Proprietà rilevate per un elemento Datacenter del Cloud .....	141
Tabella V-3: Proprietà rilevate per un elemento Datastore del Cloud .....	157
Tabella V-4: Proprietà rilevate per un elemento Cluster del Cloud .....	164
Tabella V-5: Proprietà rilevate per un elemento Host System del Cloud .....	179
Tabella V-6: Proprietà rilevate per un elemento Virtual Machine del Cloud.....	192