



Regione Toscana



Documento con Modello di valutazione e ottimizzazione del cloud, implementazione dell'algoritmo Deliverable 3.27.1

Versione 1.2
Data: 30/12/2013



Progetto iCaro

La piattaforma cloud per l'accelerazione
del business delle PMI toscane
[CUP 6408.30122011.026000074]

COMPUTER
GROSS


liberologico.com



UNIVERSITÀ
DEGLI STUDI
FIRENZE
DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE



 **altro
lavoro**
Agenzia per il lavoro



Informazioni sul documento

ID Deliverable	3 27.1
Titolo Deliverable	Documento con Modello di valutazione e ottimizzazione del cloud, implementazione dell’algoritmo.
ID Attività	3.5
N. Versione / Revisione	1.2
Natura: Bozza / Definitivo	Definitivo
Partner responsabile	UNIFI
Distribuzione: Riservato / Pubblico	Pubblico
Riferimenti Autore	Brunelli
Data redazione	04/12/2012
Riferimenti revisore	Paolo Nesi
Data revisione	30/12/2013
Riferimenti soggetto che approva	Paolo Nesi
Data approvazione e consegna	30/12/2013

Controllo delle revisioni

Oggetto	Numero	Data
Prima stesura	1.0	04/12/2013
Seconda stesura	1.1	19/12/2013
Chiusura	1.2	30/12/2013



Nota di riservatezza

Il presente documento sarà utilizzato esclusivamente ai fini del progetto ICARO, ha carattere riservato e non potrà quindi essere divulgato se non in seguito ad esplicita autorizzazione scritta da parte dell’ATS, salvo il caso in cui di richieste di ottemperare ad obblighi di legge o a richieste di pubbliche autorità.



Indice

1. Introduzione	5
2. Analisi della letteratura	5
3. Funzione multi obiettivo	6
3.1 Allocazione dinamica delle risorse	7
1.2 Fascia oraria e tariffazione	9
Bibliografia.....	13

Indice figure

Figura 1 – Struttura di allocazione dinamica delle risorse (fonte [153]).....	7
Figura 2 – Esempio di curva della domanda mostrando il rapporto tra quantità di richieste e prezzo (fonte [153])	8
Figura 3 – Approccio di memory provisioning (fonte [94])	9
Figura 4 – Funzioni utilità di prezzo per il consumer (a) e per il provider (b), (fonte [149]).....	10
Figura 5 – Funzioni utilità di fascia d’orario per il consumer (fonte [149]).....	10
Figura 6 – Fase di generazione (a destra) e di valutazione (a sinistra) delle proposte (fonte [149])....	11
Figura 7 – Fase di generazione (a destra) e di valutazione (a sinistra) delle proposte (fonte [113])....	11
Figura 8 – Fase di generazione delle proposte nel meccanismo Price-Timeslot-QoS negotiation (fonte [113])	12

Indice tabelle

Tabella 1 – Obiettivo dell’ottimizzazione matematica.....	6
Tabella 2 – Risorse oggetto di ottimizzazione	6

Legenda Acronimi e sigle

Acronimo / Sigla	Dettaglio
CC	Cloud Computing
VM	Virtual Machine
SLA	Service Level Agreement
QoS	Quality of Service
CP	Cloud Provider



1. Introduzione

In questo documento, a partire dalla *review* riportata nel documento *ICARO_Deliverable 3_26 1_Compportamento dei broker* verranno analizzate e discusse nel dettaglio le funzioni multi obiettivo presentate in letteratura. In particolare nel prossimo capitolo verrà ripreso lo schema di classificazione utilizzato nell'analisi sopra citata. Infine, nel Capitolo 3 verranno presentati alcuni esempi significativi di funzioni multi obiettivo.

2. Analisi della letteratura

Seguendo il processo descritto nel capitolo 2 Metodologia del documento *ICARO_Deliverable 3_26 1_Compportamento dei broker*, i paper sono stati analizzati usando sette dimensioni di classificazione mutate, in parte, dalla letteratura.

1. *Tipologie di implementazione*, si distinguono le tre principali tipologie di implementazione del cloud computing: cloud privati, cloud pubblici e cloud ibridi.
2. *Metodi di analisi e tecniche risolutive*. La prima parte tratterà dei metodi di analisi utilizzati nei contributi, come problemi di ottimizzazione combinatoria, euristiche, analisi di benchmark e analisi di scenario. La seconda parte approfondisce la natura delle tecniche risolutive adottate, facendo distinzione tra programmazione matematica, simulazione e algoritmi euristici.
3. *Funzione obiettivo e risorse considerate*. In questa sezione i lavori che presentano modelli di ottimizzazione del cloud sono classificati in relazione alla funzione obiettivo ed al tipo di risorse considerate.
4. *Modalità della validazione*. In questa sezione verrà illustrata la modalità attraverso la quale i modelli presentati sono stati validati.
5. *Cloud commerciali*. Questa sezione presenta e descrive i principali cloud commerciali presenti sul mercato.
6. *Software per la simulazione*. In questo paragrafo sono raggruppati i principali software dedicati alla simulazione in ambito cloud computing.
7. *Cloud broker*. Infine l'ultimo criterio riguarda i sistemi di brokering adottati.

I criteri 1, 3, 5, 6 sono mutuati da Sakellari et al. (2013) [107], mentre 2, 4 da Cardoen et al. (2010). Il prossimo paragrafo riporterà per esteso il punto 3 "*Funzione obiettivo e risorse considerate*".

In questo capitolo verranno riclassificati i contributi presenti in letteratura facendo riferimento alla dimensione 3: "*Funzione obiettivo e risorse considerate*".

Per quanto riguarda le funzioni obiettivo, in prima istanza è possibile ravvisare come queste considerino in maniera preminente due tipologie di prestazioni: efficienza energetica e QoS.

In particolare con efficienza energetica si indica la capacità di sfruttare l'energia per soddisfarne un determinato fabbisogno. Minori sono i consumi relativi al soddisfacimento di suddetto fabbisogno, migliore sarà l'efficienza energetica del sistema.

Con QoS si individua in generale l'ottimizzazione delle performance del sistema, esprimibili in termini di qualità del servizio fornito all'utente. Quando è previsto un contratto di servizio con relativi



parametri di QoS (con relative penali nel caso questi parametri non vengano rispettati) si parla anche di Service Level Agreement (SLA).

Nella Tabella 1 si riporta come i diversi articoli sono classificati in base al diverso obiettivo dell'ottimizzazione condotta. Per posizionare ciascun contributo all'interno della matrice di classificazione, a ciascun articolo è stato assegnato un numero identificativo (vedi Bibliografia).

OBIETTIVO DELL'OTTIMIZZAZIONE	
Efficienza energetica	[1-3, 17, 24, 25, 60, 61, 63, 64, 70, 96, 105, 118, 130, 137]
Performance/QoS	[4-16, 18-23, 26-59, 62, 65-68, 70-95, 97-104, 106-117, 119-129, 131-136, 138-147]

Tabella 1 – Obiettivo dell'ottimizzazione matematica

I modelli presenti in letteratura considerano un insieme limitato delle risorse che caratterizzano un sistema cloud. Tra queste le risorse maggiormente considerate sono:

- CPU (CORE, computing power);
- Storage;
- RAM (memory);
- Rete (IP, bandwidth);

Nella tabella che segue (Tabella 2) riportiamo i contributi distinti per risorsa.

RISORSE	EFFICIENZA ENERGETICA	PERFORMANCE/QoS
CPU	[24, 25, 63, 69, 126]	[1, 6, 9, 10, 16, 23, 34, 50, 52, 62, 66, 73, 74, 98, 104, 110, 111, 117, 136]
Storage	[64]	[16, 49, 80, 125, 126, 136, 140]
RAM	[17, 60, 61, 76, 96, 105, 118, 130, 137]	[5, 7, 11, 13, 18-21, 26, 29-32, 40, 42, 45, 46, 48, 53, 56, 58, 75, 85, 88, 94, 95, 99, 101, 102, 104, 106, 112, 114, 116, 126, 131, 139, 141, 145, 147]
Rete		[93]

Tabella 2 – Risorse oggetto di ottimizzazione

Dalle precedenti tabelle si nota come in prevalenza le risorse maggiormente studiate siano la memoria (virtuale) e a seguire la CPU. Da molti autori la RAM, intesa come memoria istanziata a ciascuna VM, è percepita come una risorsa critica.

3. Funzione multi obiettivo

Il paragrafo precedente ha introdotto il concetto di funzione multi obiettivo, con riferimento a due differenti tipologie di prestazione: l'efficienza energetica e la Performance/QoS.



Riguardo le tecniche di efficienza energetica per la progettazione di un sistema cloud verde, in letteratura sono presenti molti contributi che definiscono il problema e propongono soluzioni per realizzare un sistema con basse emissioni di CO2. Questa tipologia di obiettivo non sarà oggetto, almeno in fase preliminare, di questo lavoro.

L’altra funzione obiettivo si interessa delle performance del cloud, intese come qualità del servizio (QoS) e rispetto degli accordi sul livello di servizio (SLA). La proprietà di elasticità, cioè la capacità di rilasciare delle risorse rapidamente è spesso indicata come una delle caratteristiche chiave, in quanto consente di adattare dinamicamente la sottostante infrastruttura informatica virtuale per le esigenze dinamiche di esecuzione delle applicazioni. I modelli presenti in letteratura considerano un insieme limitato di risorse (CPU, storage, RAM, rete). Questo è particolarmente vero a livello IaaS, dove gli utenti richiedono/rilasciano risorse, pagando per il loro utilizzo.

3.1 Allocazione dinamica delle risorse

In Figura è stata mostrata la struttura di allocazione dinamica delle risorse (Dynamic Resource Allocation) proposta dal contributo [153]. Ogni macchina fisica esegue diverse tipologie per istanziare le VM. In alcuni casi l’istanza è garantita, mentre in altri tali istanze sono “spot”. In fase di esecuzione il Market Analyzer analizza periodicamente la situazione del mercato e attua una previsione del livello futuro di domanda e offerta. Sulla base della previsione effettuata, il Capacity Planner decide il prezzo atteso di ciascun tipo di VM in ogni mercato. Questo permette al VM Scheduler di prendere delle decisioni sulla pianificazione delle risorse e massimizzare i ricavi.

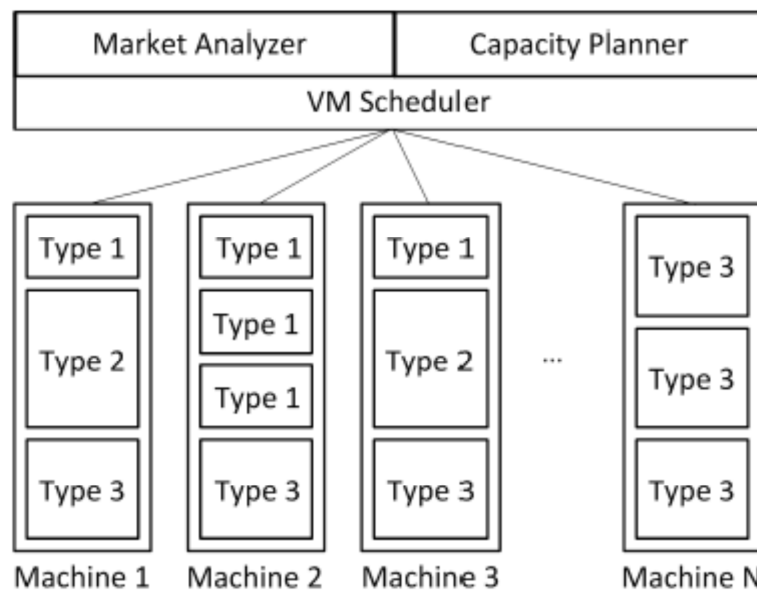


Figura 1 – Struttura di allocazione dinamica delle risorse (fonte [153])

Riguardo la modellazione e previsione della domanda, in ogni momento t (tra t_0 e t_0+T) può essere costruita una curva della domanda per catturare il rapporto tra quantità di richieste accettabili e prezzo dell’offerta, come è stato mostrato in Figura 2.

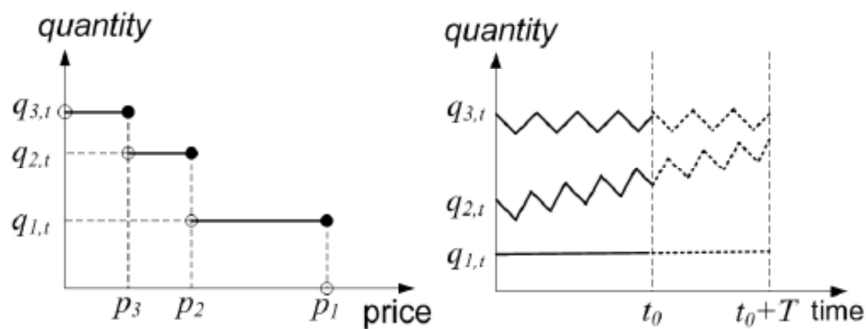


Figura 2 – Esempio di curva della domanda mostrando il rapporto tra quantità di richieste e prezzo (fonte [153])

L'obiettivo principale del problema è di pianificare le richieste di ciascun mercato per massimizzare i ricavi attesi nel corso del periodo successivo di previsione, senza eccedere le capacità delle singole macchine.

In [94] è stato riportato che l'elasticità, intesa come capacità di adattare dinamicamente la sottostante infrastruttura informatica virtuale per le esigenze dinamiche di esecuzione delle applicazioni, è divisa in due tipologie: orizzontale e verticale (come in Figura 1). L'elasticità orizzontale, da un lato, ha la capacità di un rapido provisioning, di rilasciare nodi per affrontare un cambiamento importante del carico di lavoro e di evitare costi aggiuntivi.

Un esempio tipico utilizza l'elasticità orizzontale in un'applicazione web-based con una flotta di n VM. Le richieste in arrivo sono gestite da un sistema di bilanciamento del carico che distribuisce le richieste alle n macchine virtuali. Ogni volta che il carico (CPU) di una VM supera una certa soglia, la flotta incrementa le VM ad un valore m (dove $m > n$). Altrimenti se il carico diminuisce alcune VM sono spente al fine di ridurre i costi. Dall'altra parte l'elasticità verticale ha la proprietà di modificare rapidamente la capacità di ciascuna VM, tipicamente in termini di CPU e RAM.

Negli ultimi anni gli sforzi della ricerca si sono concentrati su l'elasticità orizzontale, mentre attualmente alcuni contributi affrontano l'elasticità verticale. Inoltre una relazione della Commissione Europea sul futuro del Cloud Computing¹ afferma che l'elasticità verticale è una delle aree meno affrontate, anche se si riconosce la sua importanza per un adeguamento efficace delle infrastrutture alle applicazioni.

Il contributo [94] descrive le implicazioni principali dell'integrazione di tecniche di elasticità verticale sull'esecuzione di applicazioni con requisiti di memoria dinamica in ambienti virtualizzati. In particolare, una gestione dinamica della memoria ha permesso l'adattamento di VM ai cambiamenti di consumo di memoria con applicazioni in fase di esecuzione. Con riferimento alla Figura 3, in A, dopo che l'applicazione ha richiesto più memoria di quella disponibile è possibile aumentare la dimensione della VM fino ad un certo livello. A seconda dell'incremento più o meno rapido della memoria dell'applicazione e la regola di elasticità, la nuova dimensione della memoria della VM potrebbe non essere sufficiente per soddisfare i requisiti dell'applicazione (B).

Questo stato richiederebbe un ulteriore incremento della VM, fino al momento in cui l'applicazione avviandosi libera memoria. Il monitor rileva che tale memoria è disponibile e decide di scalare fino ad una certa quantità (C), fino ad arrivare in (D) con eventuali aggiustamenti. Questa fase di assestamento della VM può ospitare completamente l'applicazione nella memoria residente.

¹ E. G. Report, The Future of Cloud Computing. Opportunities for European Cloud Computing Beyond 2010, Tech. rep., European Commission (2010).

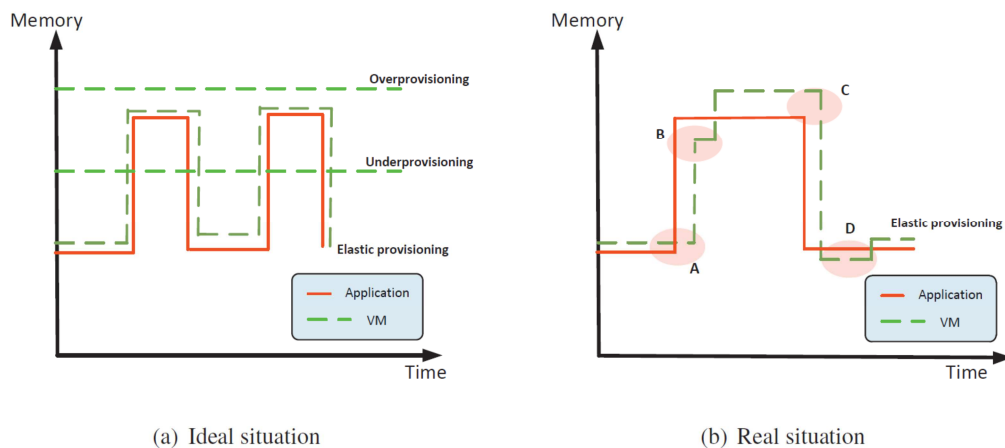


Figura 3 – Approccio di memory provisioning (fonte [94])

I risultati dello studio dimostrano come sia possibile adattare le dimensioni della memoria per ridurre il consumo, pur mantenendo durante l’esecuzione dell’applicazione adeguati livelli di prestazione (SLA). Questa regolazione è vantaggiosa per il provider di risorse, dal momento che l’host usufruendo di tale riduzione può dedicarsi ad altri processi.

Altri contributi propongono algoritmi basati sulla teoria dell’ottimizzazione convessa (convex optimization) come ad esempio Song et al. 2013 [114] che formulano alcuni algoritmi per risolvere il problema di placement delle VM. In particolare l’algoritmo assegnerà le macchine virtuali agli host fisici più adatti, in base alla condizione del carico di lavoro (come CPU, RAM, storage) durante la fase di mappatura. Queste strategie consentono di raggiungere il minimo costo globale delle macchine fisiche.

In una di queste è stato proposto un algoritmo di allocazione delle risorse che integri un meccanismo di re-indirizzamento del carico che, quando necessario, distribuisca le richieste in ingresso tra i vari siti [7].

L’obiettivo generale è quello di minimizzare i costi delle risorse assegnate in termini di macchine virtuali, garantendo nel contempo i vincoli di SLA espressi come soglia il tempo medio di risposta. E’ stata proposta una soluzione che integri sia tecniche di previsione del carico di lavoro, che tecniche di ottimizzazione non lineare. Gli esperimenti che sono stati effettuati (Normal day, heavy day e noisy day scenarios) dimostrano come le soluzioni migliorano altre euristiche proposte in letteratura (Oracle, Euristica1, Euristica2 e Euristica3) senza penalizzare gli SLA.

Il punto di vista adottato è quello di un provider che offre dei servizi Web-based (WS) presenti in diversi siti di un provider di servizi IaaS. I servizi racchiusi rappresentano applicazioni eterogenee in termini di richieste di risorse, intensità del carico di lavoro, e requisiti SLA.

Solitamente i provider di servizi IaaS retribuiscono i provider di software su base oraria. Quindi, il provider di WS deve affrontare il problema di Capacity Allocation (CA) che consiste nel determinare ad ogni ora il numero ottimale di VM per ogni classe di WS in ogni sito IaaS, a seconda del carico medio predetto su base oraria e garantendo in contemporaneamente i vincoli di SLA.

1.2 Fascia oraria e tariffazione

Un’altra funzione multi obiettivo considera la performance in termini di “funzione di utilità totale”. In particolare abbiamo due sotto obiettivi: il prezzo e la fascia d’orario. Un esempio è il meccanismo



Price- and-Time-Slot-Negotiation, presentato in [149], dove la funzione di utilità si basa sul concetto che i consumatori tendono ad un prezzo basso per l’acquisto di un servizio, mentre i fornitori vogliono vendere i loro servizi a prezzi più alti. La funzione multi-obiettivo di utilità globale considera il prezzo e la fascia d’orario (time slot).

Se P è il prezzo che entrambi i soggetti accettano come accordo, la funzione di utilità di prezzo U(P) è definita come:

$$U_P^C(P) = \begin{cases} u_{\min}^P + (1 - u_{\min}^P) \cdot \left| \frac{RP_C - P}{RP_C - IP_C} \right|, & IP_C \leq P \leq RP_C \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$U_P^P(P) = \begin{cases} u_{\min}^P + (1 - u_{\min}^P) \cdot \left| \frac{P - RP_P}{IP_P - RP_P} \right|, & RP_P \leq P \leq IP_P \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Figura 4 – Funzioni utilità di prezzo per il consumer (a) e per il provider (b), (fonte [149])

Nel caso in cui non si raggiunga un accordo tra il provider ed il consumer, la funzione di utilità è pari a 0. Per la determinazione della funzione di utilità della fascia d’orario è elaborata una funzione composta da x funzioni parziali, le quali corrispondono alle diverse fasce d’orario ritenute accettabili dal consumer e dal provider. Con un esempio grafico sono riportati gli step per la generazione della funzione di utilità (fascia oraria) del consumer. Nel grafico di sinistra sono stata riportate le funzioni parziali che riportano la prima scelta fino alla quarta per le fasce d’orario. Mentre in quello di destra è stata riportata la funzione completa di utilità, somme delle quattro precedenti.

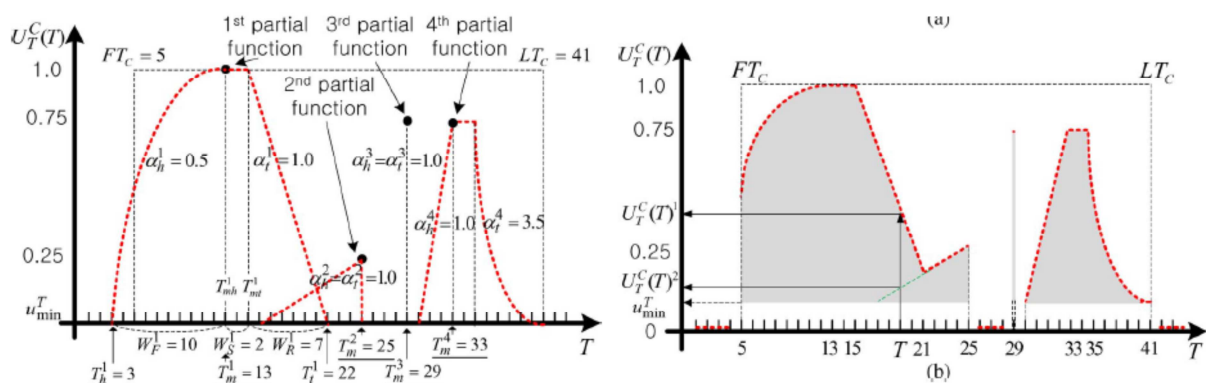


Figura 5 – Funzioni utilità di fascia d’orario per il consumer (fonte [149])

Riguardo la funzione di utilità del provider questo selezionerà (ragionevolmente), le fasce d’orario in base a:

- L’orario in cui è prevista una bassa domanda di servizio (o di carico delle risorse).
- La prima fascia d’orario disponibile, in quanto risorse inutilizzate corrispondono ad un mancato guadagno.
- L’orario in cui la mole del lavoro può ottimizzare l’utilizzo delle risorse.



Nella figura 6 è riportata una rappresentazione grafica della funzione aggregata. In particolare a sinistra è riportata la fase di **generazione** e a destra la fase di **valutazione delle proposte**. In particolare sono state generate “7-burst” proposte. Queste proposte sono disposte su una linea che rappresenta la combinazione di prezzo e fascia d’orario che generano un valore di utilità aggregata.

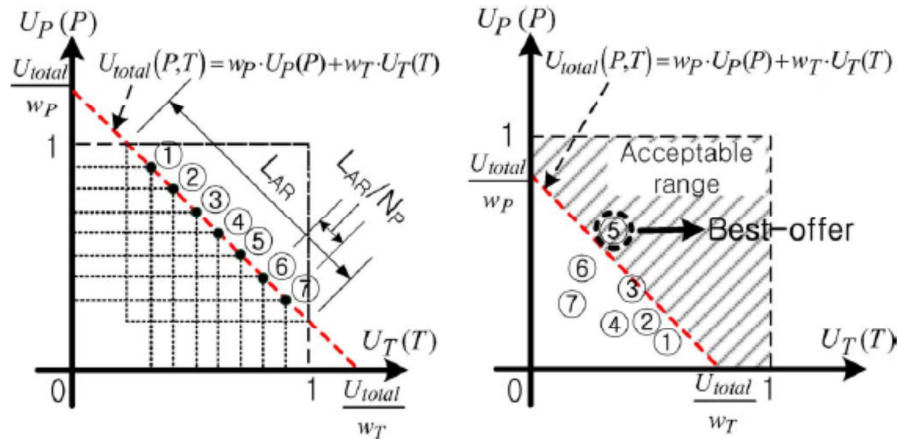


Figura 6 – Fase di generazione (a destra) e di valutazione (a sinistra) delle proposte (fonte [149])

Il meccanismo Price- and-Time-Slot-Negotiation trova l’estensione in [113]. Il contributo riporta la progettazione di un meccanismo di negoziazione multi-problema per facilitare la contrattazione e la ricerca del trade-off tra i prezzi, fascia oraria e QoS. L’algoritmo proposto, “burst mode adaptive”, è stato appositamente progettato per aumentare la velocità di negoziazione, l’utilità totale e la riduzione del carico computazionale per la valutazione delle soluzioni, adeguandosi e generando una serie di proposte simultanee.

La funzione di utilità $U(x)$ rappresenta il livello di soddisfazione di un operatore riguardo il risultato in una trattativa. Poiché ogni utente ha diverse preferenze di prezzo, fascia d’orario e QoS, è definita una funzione di utilità per ciascun elemento. In particolare la QoS è suddivisa in tre problemi distinti: performance di servizio, disponibilità del servizio e affidabilità del servizio.

$$U_{total}(P, T, Q_{performance}, Q_{availability}, Q_{reliability}) = \begin{cases} 0 & \text{if either } U_P = 0, U_T = 0, U_{Qp} = 0, U_{Qa} = 0, \text{ or } U_{Qr} = 0 \\ w_P \cdot U_P + w_T \cdot U_T + w_{Qp} \cdot U_{Qp} + w_{Qa} \cdot U_{Qa} + w_{Qr} \cdot U_{Qr}, & \text{otherwise.} \end{cases}$$

Figura 7 – Fase di generazione (a destra) e di valutazione (a sinistra) delle proposte (fonte [113])

I risultati ottenuti dalle simulazioni suggeriscono che utilizzando il meccanismo di negoziazione proposto:

- Un operatore e un provider trovano un accordo reciprocamente soddisfacente sul prezzo, sulla fascia oraria e sulle questioni di QoS, in termini di utilità aggregata.
- Si ottiene una maggiore velocità di negoziazione, con un numero relativamente basso di proposte valutate durante la negoziazione.

Il ruolo principale degli agenti del provider e dell’operatore è un negoziato bilaterale. Per ciascun negoziato l’agente genera una proposta da offrire il proprio avversario.



Le funzioni di utilità e la strategia di negoziazione che consiste in algoritmi di concessione e tradeoff, sono incorporati per generare una proposta. La sessione di negoziazione si divide in 3 fasi. Ci sono due condizioni per terminare la seduta: la proposta viene accettata oppure il termine della negoziazione viene raggiunto. In Figura 8 è stata riportata la generazione delle proposte, riportando graficamente ciascuna funzione obiettivo, tale procedura è chiamata BP o “burst proposal”.

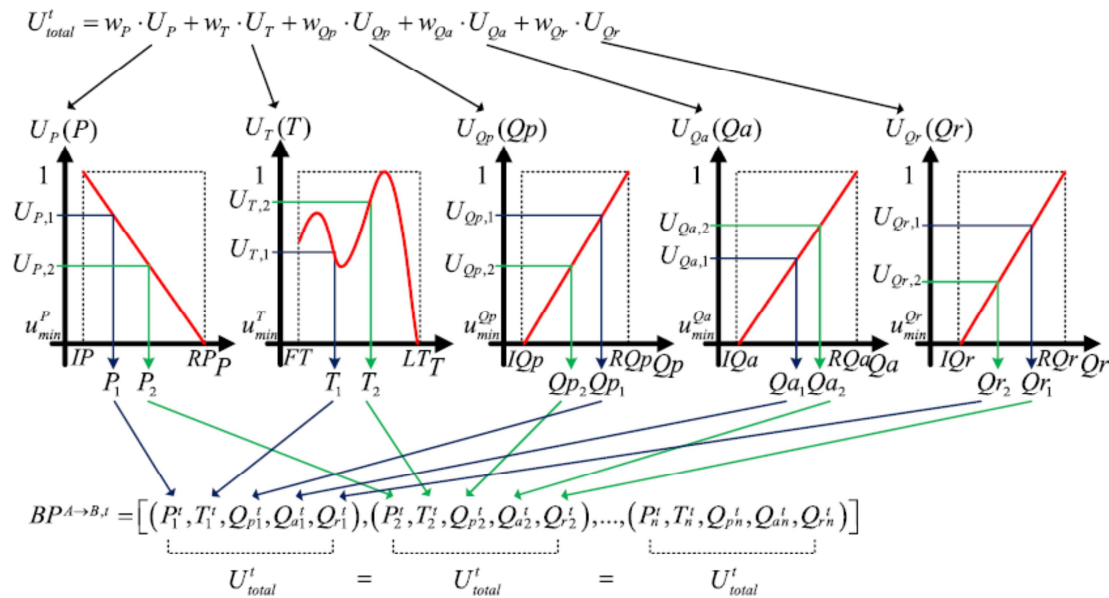


Figura 8 – Fase di generazione delle proposte nel meccanismo Price-Timeslot-QoS negotiation (fonte [113])

In fine [28,152] trattano le analisi degli effetti sulle performance strettamente riferite a due diversi sistemi di tariffazione, subscription pricing e pay-per-use pricing. Mentre [72], in controtendenza, riporta come, nonostante si evidenzi nei test che meccanismi di tariffazione pay-per-use sono adeguati ed equi, le esperienze maturate con le tecnologie ADSL, 3G e Wi-Fi mostrano che nel tempo il meccanismo di tariffazione forfettario (flat rate pricing) prevale sempre.

Tali contributi approfondiscono il concetto di funzione multi obiettivo, presentando per esteso lo sviluppo di algoritmi, modelli ed risultati di molteplici simulazioni. Resta difficile indicare, al momento, quale sia la miglior soluzione per l’implementazione di un algoritmo. Si ritiene pertanto che tali modelli e loro eventuali evoluzioni possano costituire una base di partenza per le prossime fasi di test.



Bibliografia

- [1] Abdullah, M. and M. Othman, *Cost-based Multi-QoS Job Scheduling Using Divisible Load Theory in Cloud Computing*. Procedia Computer Science, 2013. **18**: p. 928-935.
- [2] Abrishami, S. and M. Naghibzadeh, *Deadline-constrained workflow scheduling in software as a service Cloud*. Scientia Iranica, 2012. **19**(3): p. 680-689.
- [3] Abrishami, S., M. Naghibzadeh, and D.H.J. Epema, *Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds*. Future Generation Computer Systems, 2013. **29**(1): p. 158-169.
- [4] Afgan, E., et al., *Galaxy CloudMan: delivering cloud compute clusters*. BMC Bioinformatics, 2010. **11 Suppl 12**: p. S4.
- [5] Amoretti, M., F. Zanichelli, and G. Conte, *Efficient autonomic cloud computing using online discrete event simulation*. Journal of Parallel and Distributed Computing, 2013. **73**(6): p. 767-776.
- [6] Angeli, D. and E. Masala, *A cost-effective cloud computing framework for accelerating multimedia communication simulations*. Journal of Parallel and Distributed Computing, 2012. **72**(10): p. 1373-1385.
- [7] Ardagna, D., et al., *Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems*. Journal of Parallel and Distributed Computing, 2012. **72**(6): p. 796-808.
- [8] Arutyunov, V.V., *Cloud computing: Its history of development, modern state, and future considerations*. Scientific and Technical Information Processing, 2012. **39**(3): p. 173-178.
- [9] Assunção, M.D., A. Costanzo, and R. Buyya, *A cost-benefit analysis of using cloud computing to extend the capacity of clusters*. Cluster Computing, 2010. **13**(3): p. 335-347.
- [10] Atif, M. and P. Strazdins, *Adaptive parallel application resource remapping through the live migration of virtual machines*. Future Generation Computer Systems, 2013.
- [11] Begnum, K., *Simplified cloud-oriented virtual machine management with MLN*. The Journal of Supercomputing, 2010. **61**(2): p. 251-266.
- [12] Benedict, S., *Performance issues and performance analysis tools for HPC cloud applications: a survey*. Computing, 2012. **95**(2): p. 89-108.
- [13] Bittencourt, L.F. and E.R.M. Madeira, *HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds*. Journal of Internet Services and Applications, 2011. **2**(3): p. 207-227.
- [14] Brandic, I., et al., *VieSLAF Framework: Enabling Adaptive and Versatile SLA-Management*. Grid Economics and Business Models, 2009. **5745**: p. 60-73.
- [15] Broberg, J., S. Venugopal, and R. Buyya, *Market-oriented Grids and Utility Computing: The State-of-the-art and Future Directions*. Journal of Grid Computing, 2007. **6**(3): p. 255-276.
- [16] Buyya, R., *CloudAnalyst A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications*. AINA, 2010.
- [17] Caballer, M., et al., *EC3: Elastic Cloud Computing Cluster*. Journal of Computer and System Sciences, 2013. **79**(8): p. 1341-1351.
- [18] Calcavecchia, N.M., et al., *DEPAS: a decentralized probabilistic algorithm for auto-scaling*. Computing, 2012. **94**(8-10): p. 701-730.
- [19] Calheiros, R.N., et al., *A coordinator for scaling elastic applications across multiple clouds*. Future Generation Computer Systems, 2012. **28**(8): p. 1350-1362.
- [20] Calheiros, R.N., et al., *The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds*. Future Generation Computer Systems, 2012. **28**(6): p. 861-870.



- [21] Canali, C. and R. Lancellotti, *Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems*. Automated Software Engineering, 2013: p. 1-26.
- [22] Chang, Y.-S., et al., *Scaling database performance on GPUs*. Information Systems Frontiers, 2011. **14**(4): p. 909-924.
- [23] Chapman, C., et al., *Software architecture definition for on-demand cloud provisioning*. Cluster Computing, 2011. **15**(2): p. 79-100.
- [24] Chen, H., et al., *Energy-aware application performance management in virtualized data centers*. Frontiers of Computer Science, 2012. **6**(4): p. 373-387.
- [25] Chen, M., et al., *Enabling Technologies for Future Data Center Networking: A Primer*. Ieee Network, 2013. **27**(4): p. 8-15.
- [26] Chen, T., R. Bahsoon, and G. Theodoropoulos, *Dynamic QoS Optimization Architecture for Cloud-based DDDAS*. Procedia Computer Science, 2013. **18**: p. 1881-1890.
- [27] Choi, S., K. Chung, and H. Yu, *Fault tolerance and QoS scheduling using CAN in mobile social cloud computing*. Cluster Computing, 2013.
- [28] Chun, S.-H. and B.-S. Choi, *Service models and pricing schemes for cloud computing*. Cluster Computing, 2013.
- [29] Chung, L., et al., *A goal-oriented simulation approach for obtaining good private cloud-based system architectures*. Journal of Systems and Software, 2013. **86**(9): p. 2242-2262.
- [30] Coelho da Silva, T.L., et al., *Non-Intrusive Elastic Query Processing in the Cloud*. Journal of Computer Science and Technology, 2013. **28**(6): p. 932-947.
- [31] Cuomo, A., et al., *An SLA-based Broker for Cloud Infrastructures*. Journal of Grid Computing, 2012. **11**(1): p. 1-25.
- [32] D'Agostino, D., et al., *A QoS-aware broker for hybrid clouds*. Computing, 2012. **95**(S1): p. 89-109.
- [33] Deboosere, L., et al., *Efficient resource management for virtual desktop cloud computing*. The Journal of Supercomputing, 2012. **62**(2): p. 741-767.
- [34] Delamare, S., et al., *SpeQuloS: a QoS service for hybrid and elastic computing infrastructures*. Cluster Computing, 2013.
- [35] Dhar, S., *From outsourcing to Cloud computing: evolution of IT services*. Management Research Review, 2012. **35**(8): p. 664-675.
- [36] Dou, W., et al., *An evaluation method of outsourcing services for developing an elastic cloud platform*. The Journal of Supercomputing, 2010. **63**(1): p. 1-23.
- [37] Durkee, D., *Why cloud computing will never be free*. Communications of the ACM, 2010. **53**(5): p. 62.
- [38] Faniyi, F., R. Bahsoon, and G. Theodoropoulos, *A Dynamic Data-Driven Simulation Approach for Preventing Service Level Agreement Violations in Cloud Federation*. Procedia Computer Science, 2012. **9**: p. 1167-1176.
- [39] García García, A., I. Blanquer Espert, and V. Hernández García, *SLA-driven dynamic cloud resource management*. Future Generation Computer Systems, 2014. **31**: p. 1-11.
- [40] Garg, S.K. and R. Buyya. *NetworkCloudSim: modelling parallel applications in cloud simulations*. in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. 2011. IEEE.
- [41] Garrison, G., S. Kim, and R.L. Wakefield, *Success factors for deploying cloud computing*. Communications of the ACM, 2012. **55**(9): p. 62.
- [42] Ghorbannia Delavar, A. and Y. Aryan, *HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems*. Cluster Computing, 2013.
- [43] Gomez-Migueluez, I., V. Marojevic, and A. Gelonch, *Deployment and management of SDR cloud computing resources: problem definition and fundamental limits*. EURASIP Journal on Wireless Communications and Networking, 2013. **2013**(1): p. 1-11.
- [44] Goyal, T., A. Singh, and A. Agrawal, *Cloudsim: simulator for cloud computing infrastructure and modeling*. Procedia Engineering, 2012. **38**: p. 3566-3572.



- [45] Graciani Diaz, R., et al., *Belle-DIRAC Setup for Using Amazon Elastic Compute Cloud*. Journal of Grid Computing, 2011. **9**(1): p. 65-79.
- [46] Gulisano, V., et al., *StreamCloud: An Elastic and Scalable Data Streaming System*. Ieee Transactions on Parallel and Distributed Systems, 2012. **23**(12): p. 2351-2365.
- [47] Habib, S.M., et al., *Trust as a facilitator in cloud computing: a survey*. Journal of Cloud Computing, 2012. **1**(1): p. 1-18.
- [48] Han, R., et al., *Enabling cost-aware and adaptive elasticity of multi-tier cloud applications*. Future Generation Computer Systems, 2014. **32**: p. 82-98.
- [49] Han, Y., *IaaS cloud computing services for libraries: cloud storage and virtual machines*. OCLC Systems & Services, 2013. **29**(2): p. 87-100.
- [50] He, Y., et al., *A simulation cloud monitoring framework and its evaluation model*. Simulation Modelling Practice and Theory, 2013. **38**: p. 20-37.
- [51] Höfer, C.N. and G. Karagiannis, *Cloud computing services: taxonomy and comparison*. Journal of Internet Services and Applications, 2011. **2**(2): p. 81-94.
- [52] Hovestadt, M., et al., *Adaptive Online Compression in Clouds—Making Informed Decisions in Virtual Machine Environments*. Journal of Grid Computing, 2013. **11**(2): p. 167-186.
- [53] Hu, Y., et al. *Resource provisioning for cloud computing*. in *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*. 2009. IBM Corp.
- [54] Huang, J. and D.M. Nicol, *Trust mechanisms for cloud computing*. Journal of Cloud Computing: Advances, Systems and Applications, 2013. **2**(1): p. 9.
- [55] Issarny, V., et al., *Service-oriented middleware for the Future Internet: state of the art and research directions*. Journal of Internet Services and Applications, 2011. **2**(1): p. 23-45.
- [56] Iyer, R., et al., *VM3: Measuring, modeling and managing VM shared resources*. Computer Networks, 2009. **53**(17): p. 2873-2887.
- [57] Jararweh, Y., et al. *Teachcloud: a cloud computing educational toolkit*. in *Proceedings of the 1st International IBM Cloud Academy Conference (ICA CON 2012), IBM, Research Triangle Park, NC, USA*. 2012.
- [58] Javadi, B., J. Abawajy, and R. Buyya, *Failure-aware resource provisioning for hybrid Cloud infrastructure*. Journal of Parallel and Distributed Computing, 2012. **72**(10): p. 1318-1331.
- [59] Jing, S.-Y., et al., *State-of-the-art research study for green cloud computing*. The Journal of Supercomputing, 2011. **65**(1): p. 445-468.
- [60] Katsaros, G., et al., *A service framework for energy-aware monitoring and VM management in Clouds*. Future Generation Computer Systems, 2013. **29**(8): p. 2077-2091.
- [61] Kim, K.H., A. Beloglazov, and R. Buyya. *Power-aware provisioning of cloud resources for real-time services*. in *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*. 2009. ACM.
- [62] Kliazovic, *A Holistic Model for Resource Representation in Virtualized Cloud Computing Data*. 2013.
- [63] Kliazovich, D., et al., *GreenCloud: A Packet-level Simulator of Energy-aware Cloud Computing Data Centers*. 2010 Ieee Global Telecommunications Conference Globecom 2010, 2010.
- [64] Kliazovich, D., P. Bouvry, and S.U. Khan, *DENS: data center energy-efficient network-aware scheduling*. Cluster Computing, 2011. **16**(1): p. 65-75.
- [65] Kossmann, D. and T. Kraska, *Data Management in the Cloud: Promises, State-of-the-art, and Open Questions*. Datenbank-Spektrum, 2010. **10**(3): p. 121-129.
- [66] Kousiouris, G., et al., *Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms*. Future Generation Computer Systems, 2014. **32**: p. 27-40.
- [67] Krebs, R., C. Momm, and S. Kounev, *Metrics and techniques for quantifying performance isolation in cloud environments*. Science of Computer Programming, 2013.
- [68] Lai, K.-C. and Y.-F. Yu, *A scalable multi-attribute hybrid overlay for range queries on the cloud*. Information Systems Frontiers, 2011. **14**(4): p. 895-908.



- [69] Lee, H.M., Y.-S. Jeong, and H.J. Jang, *Performance analysis based resource allocation for green cloud computing*. The Journal of Supercomputing, 2013.
- [70] Lee, Y.C. and A.Y. Zomaya, *Energy efficient utilization of resources in cloud computing systems*. The Journal of Supercomputing, 2010. **60**(2): p. 268-280.
- [71] Li, B.H., et al., *New advances of the research on cloud simulation*, in *Advanced Methods, Techniques, and Applications in Modeling and Simulation 2012*, Springer. p. 144-163.
- [72] Li, C.-F., *Cloud Computing System Management Under Flat Rate Pricing*. Journal of Network and Systems Management, 2011. **19**(3): p. 305-318.
- [73] Li, C. and L. Li, *Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and SaaS provider in cloud environment*. The Journal of Supercomputing, 2013. **65**(2): p. 866-885.
- [74] Li, C. and L.Y. Li, *Optimal resource provisioning for cloud computing environment*. The Journal of Supercomputing, 2012. **62**(2): p. 989-1022.
- [75] Li, J., et al., *Online optimization for scheduling preemptable tasks on IaaS cloud systems*. Journal of Parallel and Distributed Computing, 2012. **72**(5): p. 666-677.
- [76] Liao, X., L. Hu, and H. Jin, *Energy optimization schemes in cluster with virtual machines*. Cluster Computing, 2009. **13**(2): p. 113-126.
- [77] Lim, S.H., et al., *MDCSim: A Multi-tier Data Center Simulation Platform*. 2009 IEEE International Conference on Cluster Computing and Workshops, 2009: p. 317-325.
- [78] Lin, J., L. Zha, and Z. Xu, *Consolidated cluster systems for data centers in the cloud age: a survey and analysis*. Frontiers of Computer Science, 2012. **7**(1): p. 1-19.
- [79] Lin, K.W. and Y.-C. Lo, *Efficient algorithms for frequent pattern mining in many-task computing environments*. Knowledge-Based Systems, 2013. **49**: p. 10-21.
- [80] Liu, L., *Computing infrastructure for big data processing*. Frontiers of Computer Science, 2013. **7**(2): p. 165-170.
- [81] Liu, T., et al., *SDMS-O: A service deployment management system for optimization in clouds while guaranteeing users' QoS requirements*. Future Generation Computer Systems, 2012. **28**(7): p. 1100-1109.
- [82] Liu, X., et al., *Cloud-based computer simulation: Towards planting existing simulation software into the cloud*. Simulation Modelling Practice and Theory, 2012. **26**: p. 135-150.
- [83] Lucas-Simarro, J.L., et al., *Scheduling strategies for optimal service deployment across multiple clouds*. Future Generation Computer Systems, 2013. **29**(6): p. 1431-1441.
- [84] Maciel, P.D., et al., *Business-driven short-term management of a hybrid IT infrastructure*. Journal of Parallel and Distributed Computing, 2012. **72**(2): p. 106-119.
- [85] Malawski, M., K. Figiela, and J. Nabrzyski, *Cost minimization for computational applications on hybrid cloud infrastructures*. Future Generation Computer Systems, 2013. **29**(7): p. 1786-1794.
- [86] Manuel, P., *A trust model of cloud computing based on Quality of Service*. Annals of Operations Research, 2013.
- [87] Martens, B. and F. Teuteberg, *Decision-making in cloud computing environments: A cost and risk based approach*. Information Systems Frontiers, 2011. **14**(4): p. 871-893.
- [88] Mateescu, G., W. Gentsch, and C.J. Ribbens, *Hybrid Computing-Where HPC meets grid and Cloud Computing*. Future Generation Computer Systems-the International Journal of Grid Computing and Esience, 2011. **27**(5): p. 440-453.
- [89] Mauch, V., M. Kunze, and M. Hillenbrand, *High performance cloud computing*. Future Generation Computer Systems, 2013. **29**(6): p. 1408-1416.
- [90] Maurer, M., I. Brandic, and R. Sakellariou, *Adaptive resource configuration for Cloud infrastructure management*. Future Generation Computer Systems, 2013. **29**(2): p. 472-487.
- [91] Maurer, M., et al., *Cost-benefit analysis of an SLA mapping approach for defining standardized Cloud computing goods*. Future Generation Computer Systems, 2012. **28**(1): p. 39-47.



- [92] Mazhelis, O. and P. Tyrväinen, *Economic aspects of hybrid cloud infrastructure: User organization perspective*. Information Systems Frontiers, 2011. **14**(4): p. 845-869.
- [93] Miguelez, G., *Deployment and management of SDR cloud computing resources: problem definition and fundamental limits*. JWCN, 2013.
- [94] Moltó, G., et al., *Elastic Memory Management of Virtualized Infrastructures for Applications with Dynamic Memory Requirements*. Procedia Computer Science, 2013. **18**: p. 159-168.
- [95] Moschakis, I.A. and H.D. Karatza, *Evaluation of gang scheduling performance and cost in a cloud computing system*. The Journal of Supercomputing, 2010. **59**(2): p. 975-992.
- [96] Nathuji, R., et al., *VPM tokens: virtual machine-aware power budgeting in datacenters*. Cluster Computing, 2009. **12**(2): p. 189-203.
- [97] Niezgodá, S.R., A.K. Kanjarla, and S.R. Kalidindi, *Novel microstructure quantification framework for databasing visualization and analysis of microstructure data*. Integrating Materials and Manufacturing Innovation, 2013. **2**(1): p. 1-27.
- [98] Núñez, A., et al., *iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator*. Journal of Grid Computing, 2012. **10**(1): p. 185-209.
- [99] Oliveira, D., et al., *A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds*. Journal of Grid Computing, 2012. **10**(3): p. 521-552.
- [100] Prodan, R., M. Sperk, and S. Ostermann, *Evaluating High-Performance Computing on Google App Engine*. Ieee Software, 2012. **29**(2): p. 52-58.
- [101] Quiroz, A., et al. *Towards autonomic workload provisioning for enterprise grids and clouds*. in *Grid Computing, 2009 10th IEEE/ACM International Conference on*. 2009. IEEE.
- [102] Ranjan, R. and L. Zhao, *Peer-to-peer service provisioning in cloud computing environments*. The Journal of Supercomputing, 2011. **65**(1): p. 154-184.
- [103] Redmond, A., et al., *Exploring how information exchanges can be enhanced through Cloud BIM*. Automation in Construction, 2012. **24**: p. 175-183.
- [104] Rodero-Merino, L., et al., *Using clouds to scale grid resources: An economic model*. Future Generation Computer Systems, 2012. **28**(4): p. 633-646.
- [105] Rodero, I., et al., *Energy-Efficient Thermal-Aware Autonomic Management of Virtualized HPC Cloud Infrastructure*. Journal of Grid Computing, 2012. **10**(3): p. 447-473.
- [106] Rogers, O. and D. Cliff, *A financial brokerage model for cloud computing*. Journal of Cloud Computing, 2012. **1**(1): p. 1-12.
- [107] Sakellari, G. and G. Loukas, *A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing*. Simulation Modelling Practice and Theory, 2013. **39**: p. 92-103.
- [108] Sakr, S., *Cloud-hosted databases: technologies, challenges and opportunities*. Cluster Computing, 2013.
- [109] Sethia, P. and K. Karlapalem, *A multi-agent simulation framework on small Hadoop cluster*. Engineering Applications of Artificial Intelligence, 2011. **24**(7): p. 1120-1127.
- [110] Sharma, S., et al., *Decentralized allocation of CPU computation power for web applications*. Performance Evaluation, 2010. **67**(11): p. 1187-1202.
- [111] Shiraz, M., et al., *Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing*. The Journal of Supercomputing, 2013.
- [112] Son, S., G. Jung, and S.C. Jun, *An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider*. The Journal of Supercomputing, 2013. **64**(2): p. 606-637.
- [113] Son, S. and K.M. Sim, *Adaptive and similarity-based tradeoff algorithms in a price-timeslot-QoS negotiation system to establish cloud SLAs*. Information Systems Frontiers, 2013.
- [114] Song, F., et al., *An Optimization-Based Scheme for Efficient Virtual Machine Placement*. International Journal of Parallel Programming, 2013.



- [115] Streitberger, W. and T. Eymann, *A simulation of an economic, self-organising resource allocation approach for application layer networks*. Computer Networks, 2009. **53**(10): p. 1760-1770.
- [116] Sun, G., et al., *A cost efficient framework and algorithm for embedding dynamic virtual network requests*. Future Generation Computer Systems, 2013. **29**(5): p. 1265-1277.
- [117] Taylor, R.C., *An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics*. BMC Bioinformatics, 2010. **11**(Suppl 12): p. S1.
- [118] Tian, W., Q. Xiong, and J. Cao, *An online parallel scheduling method with application to energy-efficiency in cloud computing*. The Journal of Supercomputing, 2013. **66**(3): p. 1773-1790.
- [119] Truong Huu, T., et al., *Joint Elastic Cloud and Virtual Network Framework for Application Performance-cost Optimization*. Journal of Grid Computing, 2010. **9**(1): p. 27-47.
- [120] Van den Bossche, R., K. Vanmechelen, and J. Broeckhove, *Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds*. Future Generation Computer Systems, 2013. **29**(4): p. 973-985.
- [121] Vanmechelen, K., S. De Munck, and J. Broeckhove, *Conservative distributed discrete-event simulation on the Amazon EC2 cloud: An evaluation of time synchronization protocol performance and cost efficiency*. Simulation Modelling Practice and Theory, 2013. **34**: p. 126-143.
- [122] Vázquez, C., et al., *On the use of clouds for grid resource provisioning*. Future Generation Computer Systems, 2011. **27**(5): p. 600-605.
- [123] Vouk, M.A., *Cloud computing - Issues, research and implementations*. Proceedings of the Iti 2008 30th International Conference on Information Technology Interfaces, 2008: p. 31-40.
- [124] Walraven, S., E. Truyen, and W. Joosen, *Comparing PaaS offerings in light of SaaS development*. Computing, 2013.
- [125] Wang, D., *An Efficient Cloud Storage Model for Heterogeneous Cloud Infrastructures*. Procedia Engineering, 2011. **23**: p. 510-515.
- [126] Wang, J.-z., P. Varman, and C.-s. Xie, *Optimizing storage performance in public cloud platforms*. Journal of Zhejiang University SCIENCE C, 2011. **12**(12): p. 951-964.
- [127] Wang, L.Z., et al., *Cloud Computing: a Perspective Study*. New Generation Computing, 2010. **28**(2): p. 137-146.
- [128] Wang, W.-J., et al., *Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments*. The Journal of Supercomputing, 2013. **66**(2): p. 783-811.
- [129] Wang, W., et al., *Cloud-DLS: Dynamic trusted scheduling for Cloud computing*. Expert Systems with Applications, 2012. **39**(3): p. 2321-2329.
- [130] Wang, X., Z. Du, and Y. Chen, *An adaptive model-free resource and power management approach for multi-tier cloud environments*. Journal of Systems and Software, 2012. **85**(5): p. 1135-1146.
- [131] Warneke, D. and O. Kao, *Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud*. Ieee Transactions on Parallel and Distributed Systems, 2011. **22**(6): p. 985-997.
- [132] Wei K. Tsai, M.I.a.J.R., *Thresholds Revisit the Strings Versus Clouds Debate for the Internet Architecture, Part II QoS, Control, Management, and TCP*. Journal of Network and Systems Management, 2002.
- [133] Wei K. Tsai, M.I.a.J.R., *Thresholds Revisit the Strings versus Clouds Debate for the Internet Architecture. Part I Control, Scalability, and QoS*. Journal of Network and Systems Management, 2002.
- [134] Weinhardt, C., et al., *Cloud Computing - A Classification, Business Models, and Research Directions*. Business & Information Systems Engineering, 2009. **1**(5): p. 391-+.
- [135] Wickremasinghe, B., R.N. Calheiros, and R. Buyya. *Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications*. in *Advanced*



- Information Networking and Applications (AINA), 2010 24th IEEE International Conference on.* 2010. IEEE.
- [136] Wright, P., et al., *A constraints-based resource discovery model for multi-provider cloud environments.* Journal of Cloud Computing, 2012. **1**(1): p. 1-14.
- [137] Wu, C.-M., R.-S. Chang, and H.-Y. Chan, *A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters.* Future Generation Computer Systems, 2013.
- [138] Wu, L., S. Kumar Garg, and R. Buyya, *SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments.* Journal of Computer and System Sciences, 2012. **78**(5): p. 1280-1299.
- [139] Xu, B., et al., *Job scheduling algorithm based on Berger model in cloud environment.* Advances in Engineering Software, 2011. **42**(7): p. 419-425.
- [140] Yan-hua, Z., F. Lei, and Y. Zhi, *Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm.* Procedia Engineering, 2011. **15**: p. 3341-3345.
- [141] Yang, J., et al., *A cost-aware auto-scaling approach using the workload prediction in service clouds.* Information Systems Frontiers, 2013.
- [142] Yang, K. and X. Jia, *Data storage auditing service in cloud computing: challenges, methods and opportunities.* World Wide Web, 2011. **15**(4): p. 409-428.
- [143] Yang, S.Q., *Move into the Cloud, shall we?* Library Hi Tech News, 2012. **29**(1): p. 4-7.
- [144] Yoo, C.S., *Cloud Computing: Architectural and Policy Implications.* Review of Industrial Organization, 2011. **38**(4): p. 405-421.
- [145] Zaman, S. and D. Grosu, *Combinatorial auction-based allocation of virtual machine instances in clouds.* Journal of Parallel and Distributed Computing, 2013. **73**(4): p. 495-508.
- [146] Zhang, Q., L. Cheng, and R. Boutaba, *Cloud computing: state-of-the-art and research challenges.* Journal of Internet Services and Applications, 2010. **1**(1): p. 7-18.
- [147] Zhang, T., et al., *Typical Virtual Appliances: An optimized mechanism for virtual appliances provisioning and management.* Journal of Systems and Software, 2011. **84**(3): p. 377-387.
- [148] Bennani M.N., et al., *Resource Allocation for Autonomic Data Centers using Analytic Performance Models.* Computer Society, 2005
- [149] Son S., et al., *A Price-and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations,* Systems, Man, And Cybernetics, 2012.
- [150] Xiao Z., et al., *Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment,* Parallel And Distributed Systems, 2013
- [151] Yang, J., et al., *A new method based on PSR and EA-GMDH for host load prediction in cloud computing system.* J Supercomput, 2014.
- [152] Yeo C.S., et al., *Autonomic Metered Pricing for a Utility Computing Service,* Future Generation Computer Systems, 2009
- [153] Zhang Q., et al., *Dynamic Resource Allocation for Spot Markets in Clouds,* Utility and Cloud Computing, 2013