

1. Aspetti di Data Management

- CouchBase

2. Aspetti Architettureali

- Infrastruttura per i Big Data
- Sistemi SMP (*Symmetric Multi Processing*)
- Sistemi MMP (*Massive Parallel Processing*)

- E' un **NoSQL database**, *Open Source* adatto a *sistemi mission-critical*.
- **Non** offre il supporto a **transazioni multi-operazionali**.
- Le *applicazioni basate su couchbase* sono **facilmente scalabili** verso un *numero "infinito" di utenti*.
- Appartiene alla famiglia dei **document DB** **schema-less** (offre buona *flessibilità*, e *capacità di indicizzazione e query simili a quelle dei RDBMS*).



- Offre **elevate prestazioni e bassissima latenza.**
 - *Operazioni di lettura e scrittura sotto i millisecondi*
 - *Nessun calo di prestazioni allo scalare delle applicazioni.*
- **Replicazione automatica.**



Couchbase

CouchBase - Caratteristiche

API: Supporto di varie SDK per Java, C, C#, PHP, Ruby.

Protocollo: *Interfacce Memcached di tipo REST* per la configurazione e la gestione del cluster.

Scritto in: *C/C++ e Erlang* (per la clusterizzazione).

Replicazione: Peer to Peer, fully consistent.

Misc: *Trasparente alle variazioni topologiche durante la fase di esecuzione*, disponibilità di una versione commerciale con supporto.

CouchBase - Caratteristiche

Memcached compatibile

- Permette alle applicazioni di **leggere e scrivere dati con una latenza inferiore al millisecondo** garantendo un elevato rendimento. **Gli utenti possono allocare le diverse risorse di cache per database** a seconda delle esigenze applicative.

Zero-downtime di Manutenzione

- Le operazioni di manutenzione possono essere eseguite mantenendo il cluster attivo. Ciò non riguarda solo le **modifiche alla topologia** come l'aggiunta o la rimozione di server che non richiedono tempi di inattività delle applicazioni, ma ***anche gli aggiornamenti software possono essere fatti su un sistema in esecuzione.***



CouchBase - Caratteristiche

Clonazione per aumentare le dimensioni del cluster con supporto dell'*Auto-Sharding*

- E' possibile **aggiungere (o rimuovere) i server nel cluster con il clic di un pulsante** adattando le risorse alle mutevoli esigenze della applicazione in esecuzione.
- La semplice aggiunta di più server **realizza una scalabilità orizzontale** del sistema con l'aggiunta di RAM e di capacità di I/O.
- L'Auto-sharding distribuisce uniformemente i dati tra server e consente l'instradamento diretto di richieste al server appropriato senza la necessità di eventuali modifiche alle applicazioni.



CouchBase - Caratteristiche

Couchbase Edit internal settings • Documentation • Support Forums • About • [Sign Out](#)

[Cluster Overview](#) **Server Nodes** [Data Buckets](#) [Views](#) [XDCR](#) [Log](#) [Settings](#)

Servers

Active Servers Pending Rebalance [Rebalance](#) [Add Server](#)

Server Node Name	RAM Usage	Swap Usage	CPU Usage	Items (Active / Replica)	
▶ 10.166.242.91	Up 24.3%	0%	5.58%	42.2 K/ 42.9 K	Fail Over Remove
▶ 10.166.243.248	Up 24.1%	0%	13%	42.4 K/ 42.4 K	Fail Over Remove
▶ 10.167.13.233	Up 24.7%	0%	5.69%	42.4 K/ 42.3 K	Fail Over Remove
▶ 10.188.1.14	Up 23.9%	0%	5.75%	42.4 K/ 42.7 K	Fail Over Remove
▶ 10.188.1.47	Up 26.1%	0%	23.2%	42.9 K/ 42.4 K	Fail Over Remove
▶ 10.188.1.136	Up 23.8%	0%	11.8%	42.7 K/ 42.5 K	Fail Over Remove
▶ 10.188.9.168	Up 24%	0%	11.2%	42.4 K/ 42.2 K	Fail Over Remove
▶ 10.188.13.105	Up 23.7%	0%	9.84%	42.5 K/ 42.4 K	Fail Over Remove

CouchBase - Caratteristiche

Gestione e monitoraggio production-ready

- Fornisce un monitoraggio avanzato grazie ad una ricca interfaccia web di amministrazione. Gli utenti possono controllare l'intero cluster in utilizzando un sistema di monitoraggio che realizza grafici e statistiche in tempo reale.

Architettura di storage affidabile

- Tutti i **dati vengono immagazzinati in modo asincrono sul disco** in modo da consentire la memorizzazione di dataset più grandi della dimensione fisica della RAM. *Couchbase sposta automaticamente i dati tra la RAM e il disco e mantenendo il “working set” nella cache a livello-oggetto.*



CouchBase - Caratteristiche



CouchBase - Caratteristiche

Replicazione dei dati con auto-failover

- Per garantire **un'elevata disponibilità** delle applicazioni basate su Server CouchBase, la capacità di replica dei dati consente di **mantenere più copie dei dati all'interno di un cluster**. Le repliche dei dati sono distribuite su tutti i nodi riducendo così l'impatto di un fallimento sui singoli nodi.

SDK professionali per un'ampia varietà di linguaggi

- **E' offerto un pieno** supporto alle SDK per Java, C#, PHP, C, Python e Ruby.



CouchBase - Vantaggi

Prestazioni

- **Il modello document-data** mantiene i dati correlati in un'unica locazione fisica in memoria e su disco (un documento). **Ciò consente sempre l'accesso ai dati con una bassa latenza** – le operazioni di lettura e scrittura avvengono con pochissimo ritardo.

Elasticità dinamica

- Poiché l'approccio document-based conserva i record "in un'unica locazione" è molto più facile spostare i dati da un server all'altro, garantendone e mantenendone la coerenza.

CouchBase - Vantaggi

Flessibilità di schema

- Tutti i database NoSQL forniscono la flessibilità di schema, ma i DB chiave-valore e i document DB godono ancor più di flessibilità. **Questi db, infatti, non richiedono alcuna manutenzione del database per modificare lo schema del database** (per aggiungere e rimuovere campi o elementi di dati da uno specifico record).

Flessibilità di query

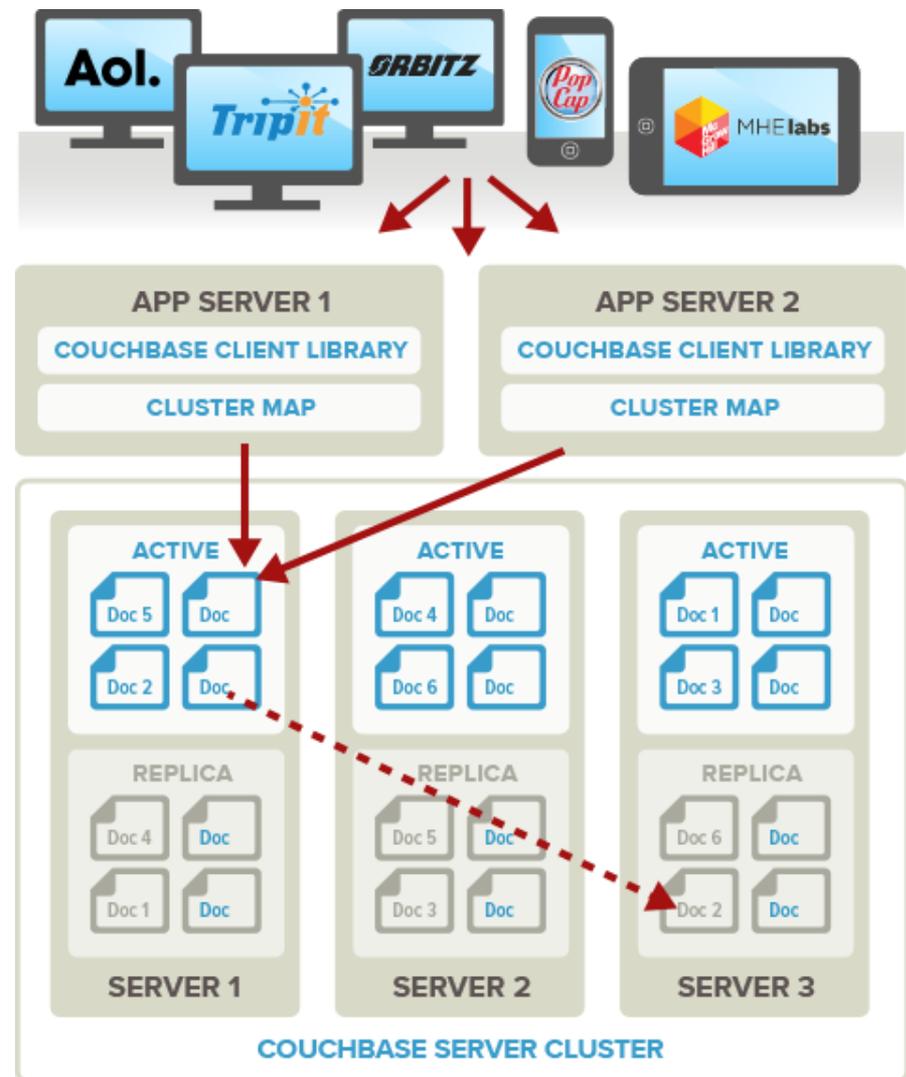
- **Espressività Query:** è la capacità di porre domande ad un db, come "restituiscimi un elenco di tutte le aziende in cui un impiegato ha guadagnato XX il mese scorso".

I Document-database forniscono il miglior equilibrio tra la flessibilità di schema e la possibilità di fare query sofisticate.

CouchBase – Operazioni di base

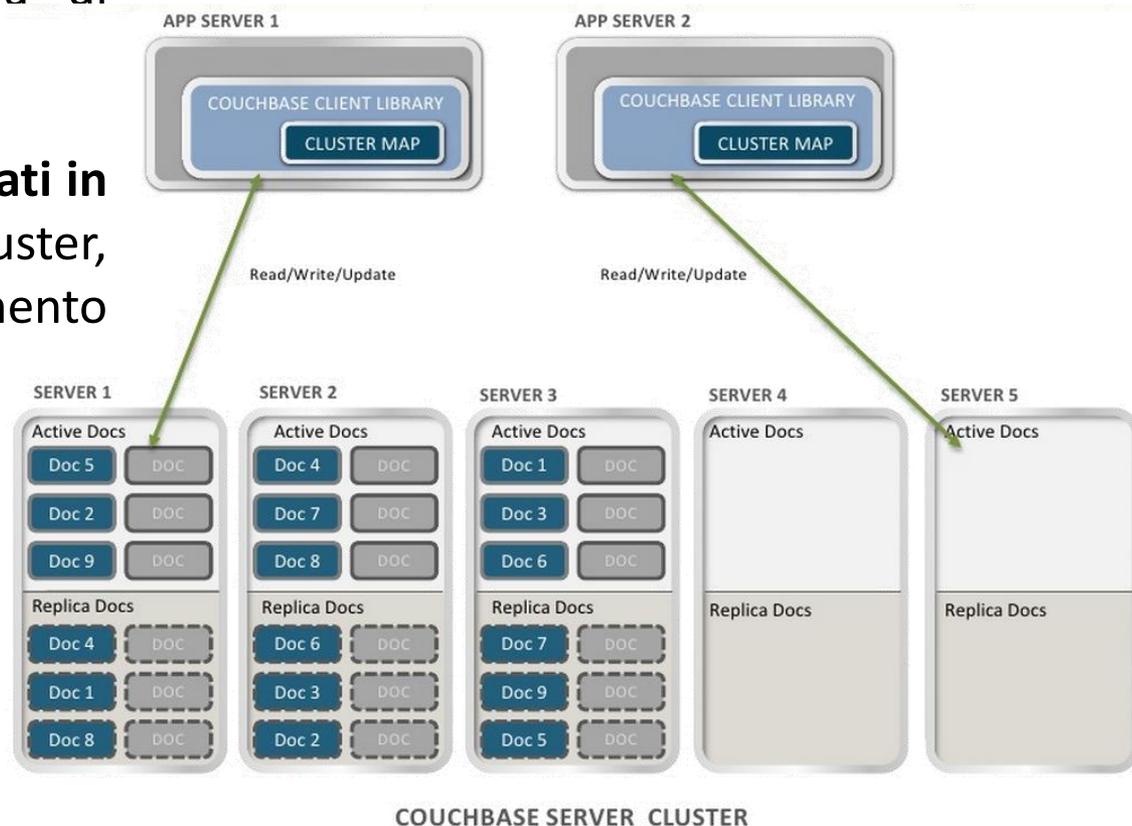
- I documenti sono distribuiti in modo uniforme tra i server all'interno del cluster.
- Ogni server memorizza entrambi i documenti, sia quelli attivi che la replica.
- Le App leggono, scrivono e aggiornano i documenti.

App Server multipli possono accedere contemporaneamente ad uno stesso documento.



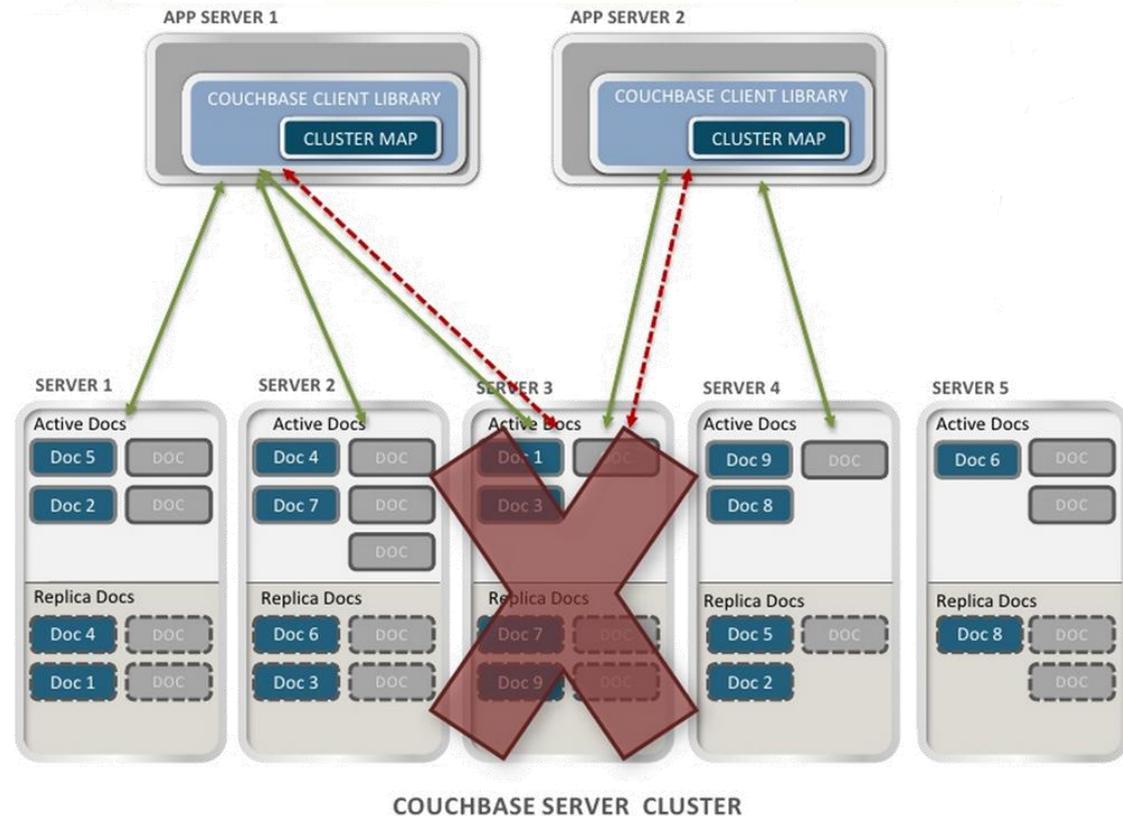
CouchBase – Aggiunta dei nodi

- L'aggiunta di altri server ad un cluster si realizza facilmente per mezzo dell'interfaccia di monitoraggio.
- I documenti sono **ribilanciati in modo automatico** sul cluster, secondo lo spostamento minimo di documenti.
- Aggiornata la Cluster Map le chiamate alle App Database saranno distribuite su un numero maggiore di server.



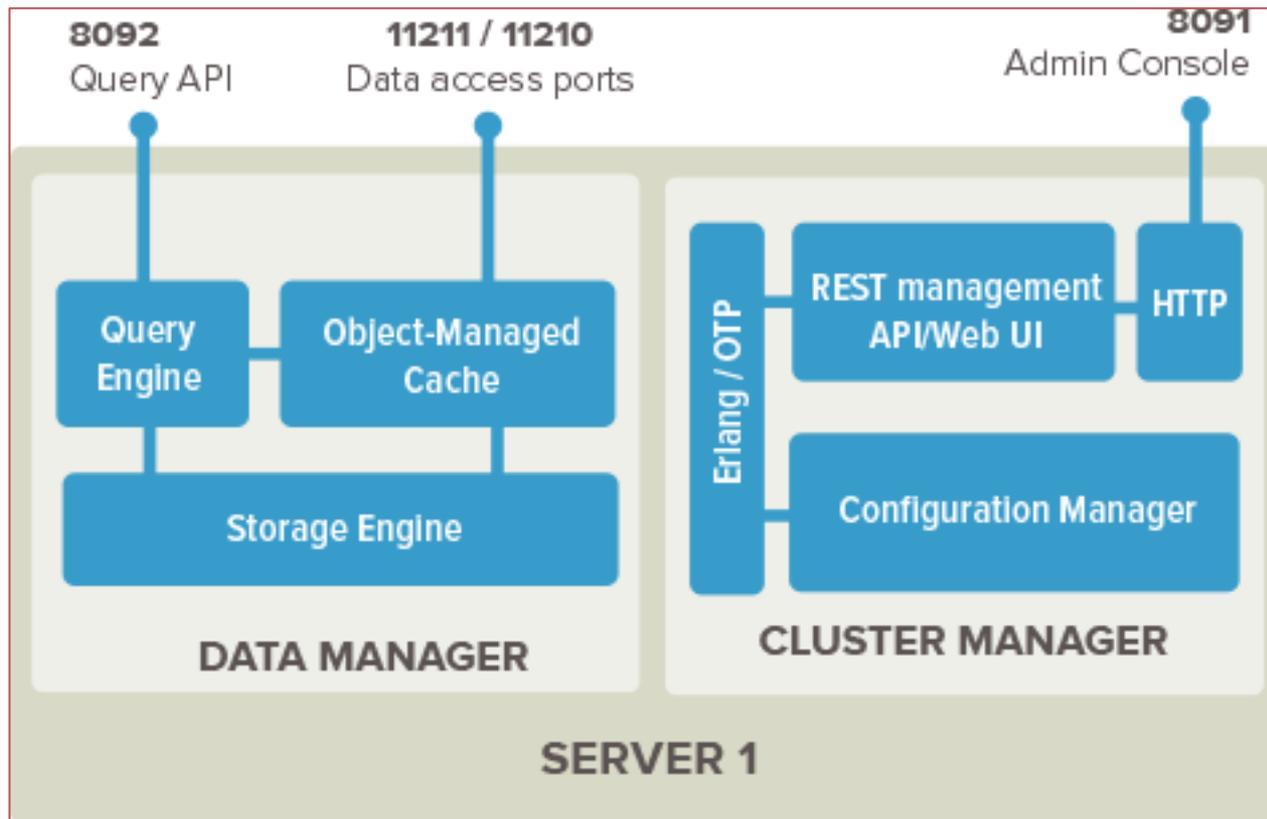
CouchBase – Rimozione dei nodi

- Un cluster rileva il fallimento di un server, di conseguenza favorisce l'uso delle repliche dei documenti persi e aggiorna la Cluster Map.
- Le richieste di documenti da parte delle App devono essere indirizzate ai server appropriati.
- Viene quindi realizzato un ribilanciamento dei dati tra i nodi.



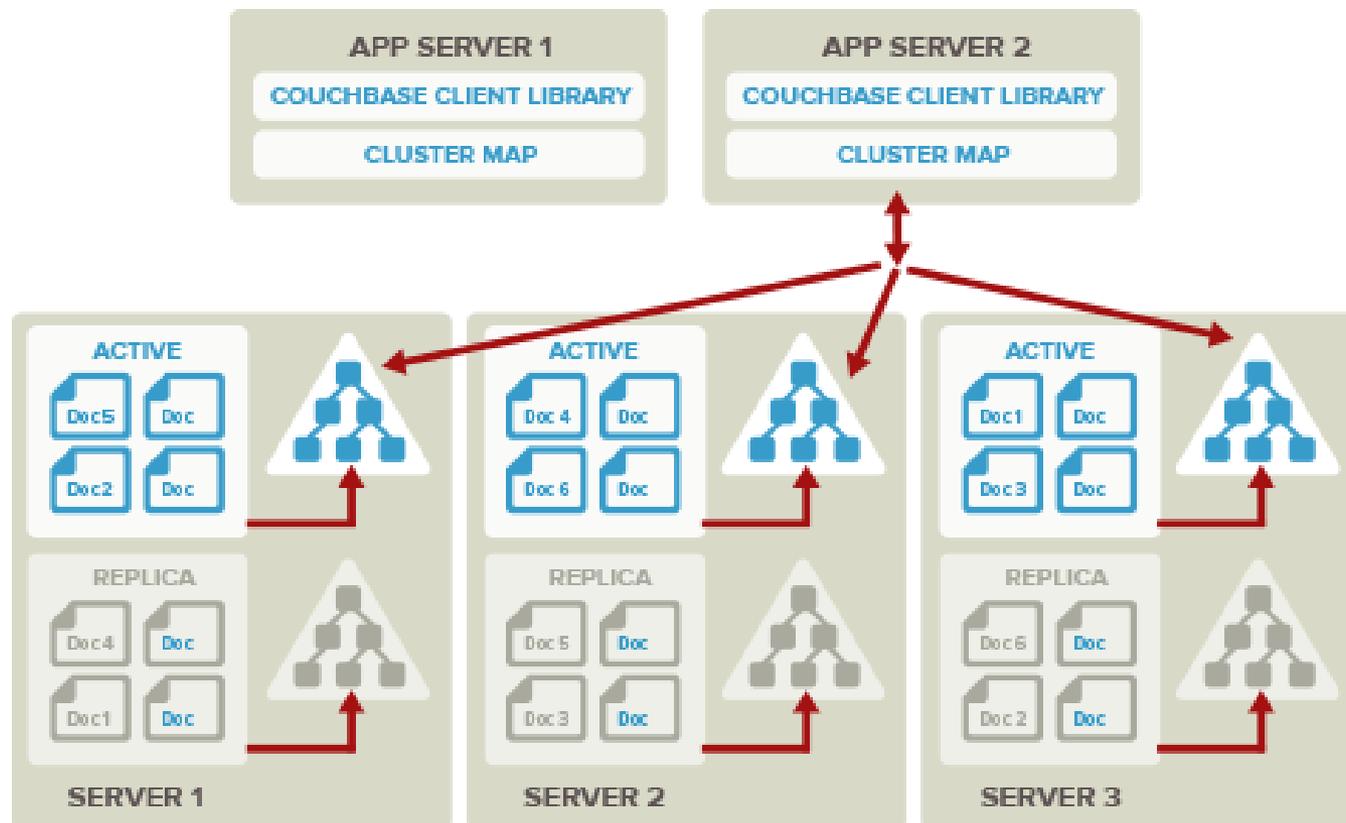
CouchBase – Architettura

- **Cluster Manager** – Gestisce e supervisiona i server di un cluster.
- **Data Manager** – Si occupa della gestione dei dati, primari e repliche.



CouchBase – Indicizzazione e Query

- Utilizzo di **indici per organizzare i documenti** sui nodi e di un **approccio map-reduce per realizzare query** su sottoinsiemi di documenti.



CouchBase - diffusione

Internet Companies



Enterprises



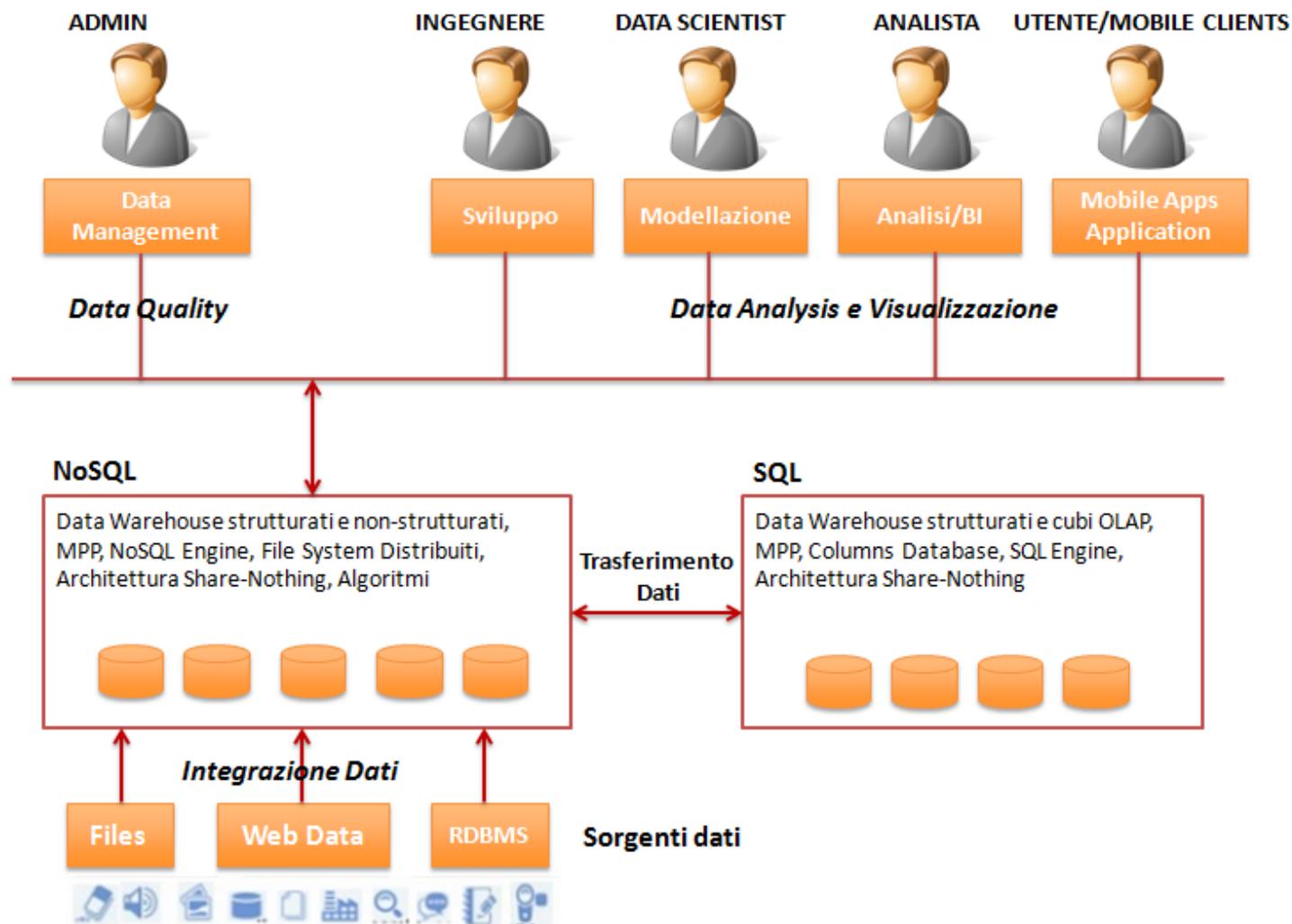
Panoramica sulle tecnologie Big Data

Considerando le tecnologie e le soluzioni adatte a lavorare con i **Big Data** bisogna prestare attenzione a **4 classi di aspetti fondamentali**:

- ***Aspetti Architetture***

Nell'elaborazione di un insieme molto grande di dati è importante **ottimizzare il carico di lavoro**, ad esempio adottando un'architettura parallela (distribuita), o fornendo una **allocazione dinamica delle risorse di calcolo**.

Panoramica Infrastruttura Big Data



Tecnologie per i Big Data

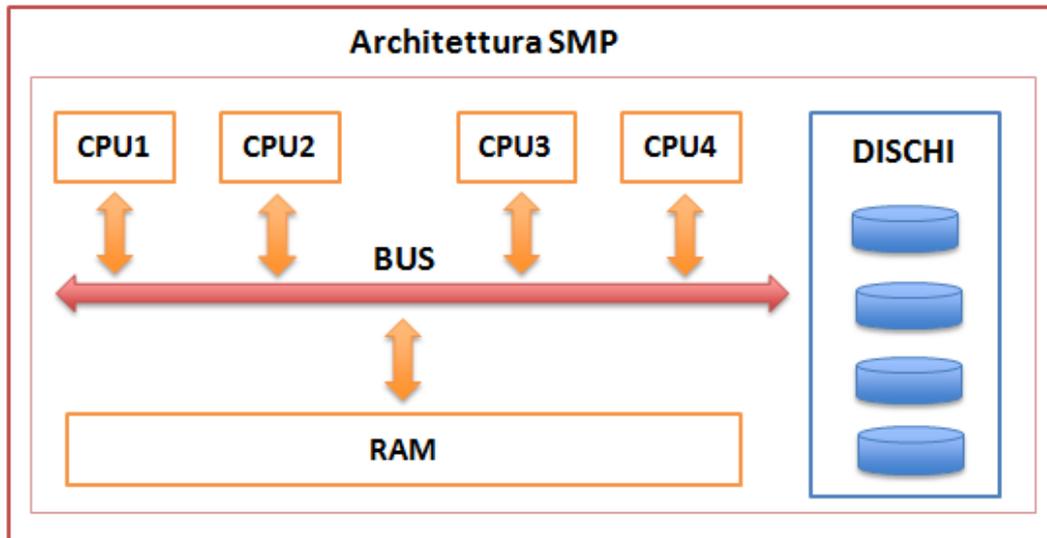
- Tra le **tecnologie** che permettono la gestione e l'analisi *grandi moli di dati non-strutturati (Big Data)* troviamo sicuramente:
 - **Hadoop** (con le “componenti” **HDFS** e **MapReduce**).
 - Meccanismi di *sharding* e *replicazione* dei **NoSQL DB**.
- Queste soluzioni permettono:
 - L'uso di **HW di costo medio/basso**, ottenendo comunque buone capacità di calcolo.
 - La **gestione dei fallimenti** (*fault-tolerance*).
 - L'**ottimizzazione delle performace** via SW.

Osservazione

I **Big Data** sono caratterizzati dalla **Variabilità**, cioè una *mancaza di struttura ben definita*. Ciò complica la loro rappresentazione in **modello relazionale**.

Tecnologie per i Big Data - SMP

- **Dati caratterizzati solo da *Volume e Velocità***. In questo caso gli RDBMS potrebbero fornire un supporto di memorizzazione, come?!
- Generalmente gli **RDBMS** sono ospitati su **sistemi di tipo SMP** (*Symmetric Multi Processing*):
 - Composti da **più processori diversi** che **condividono** lo stesso **OS**, la stessa **memoria RAM** e lo stesso **bus Input/Output** (canale di collegamento tra processore e device esterni che consente il passaggio dei dati).



shared everything

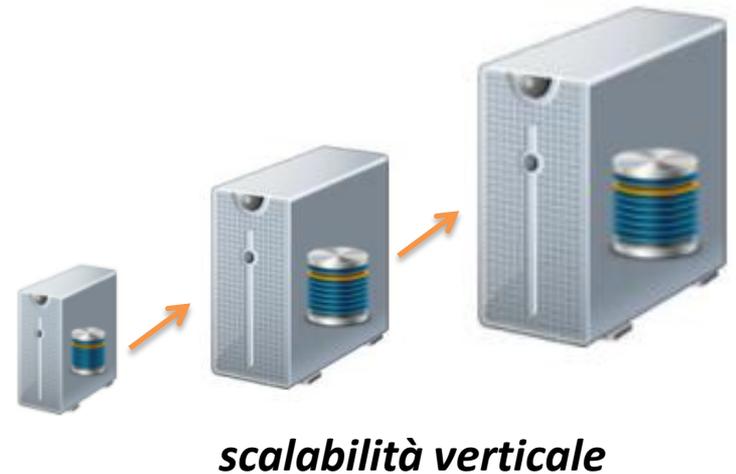
Tecnologie per i Big Data - SMP

- Gli **SMP** sono una **soluzione economica** per la **scalabilità dei sistemi**, realizzata aggiungendo RAM, CPU e dischi fino al limite installabile.
- **Applicazioni** sviluppate per far **lavorare in parallelo** i vari processori ottimizzando la **suddivisione del carico di lavoro**.
- In **ottica RDBMS**, questi sistemi sono **efficienti in applicazioni OLTP** (*On Line Transaction Processing – molti inserimenti, update e scritture*).

(-) **Non adatti** ad elaborare **grandi moli di dati**, poiché il **sovraccarico del BUS** può portare all'**effetto collo di bottiglia**.

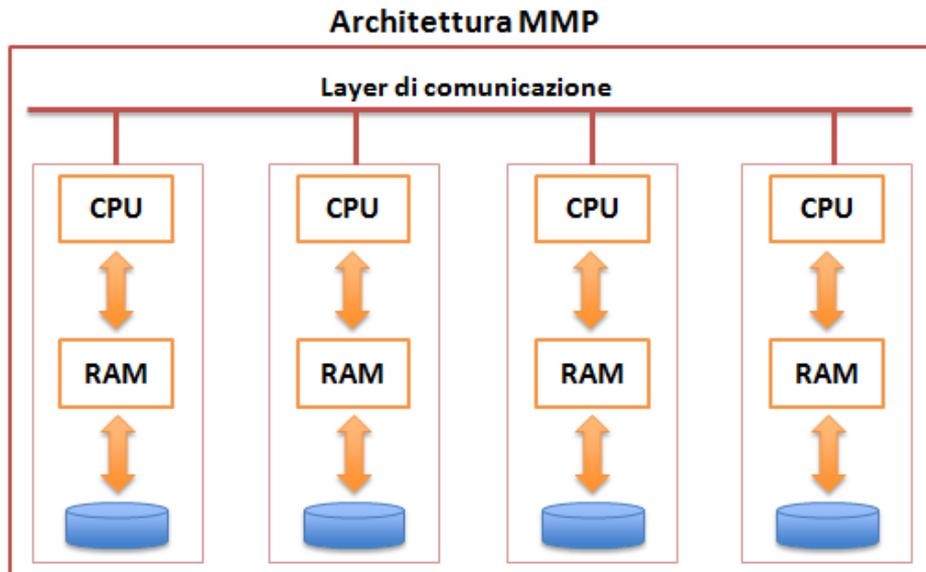
- **Possibili soluzioni** sono le **architetture NUMA** (*Non Uniform Memory Access*).

Processori accorpati in moduli, possono accedere alla memoria del proprio modulo o di altri (con tempi di accesso differenti).



Tecnologie per i Big Data - MMP

- Quando il **Volume dei dati aumenta** in modo considerevole, i **sistemi MPP** (*Massive Parallel Processing*) sono **maggiormente adatti** per l'elaborazione.
- **Ogni processore** ha un **sistema di I/O** e una **RAM dedicati**.
- **Operazioni** da svolgere sono **suddivise in *task* paralleli e indipendenti**.
- **Processori comunicano** tra loro mediante un'interfaccia di ***messaging*** (scompaiono i limiti legati alla *condivisione del bus*).



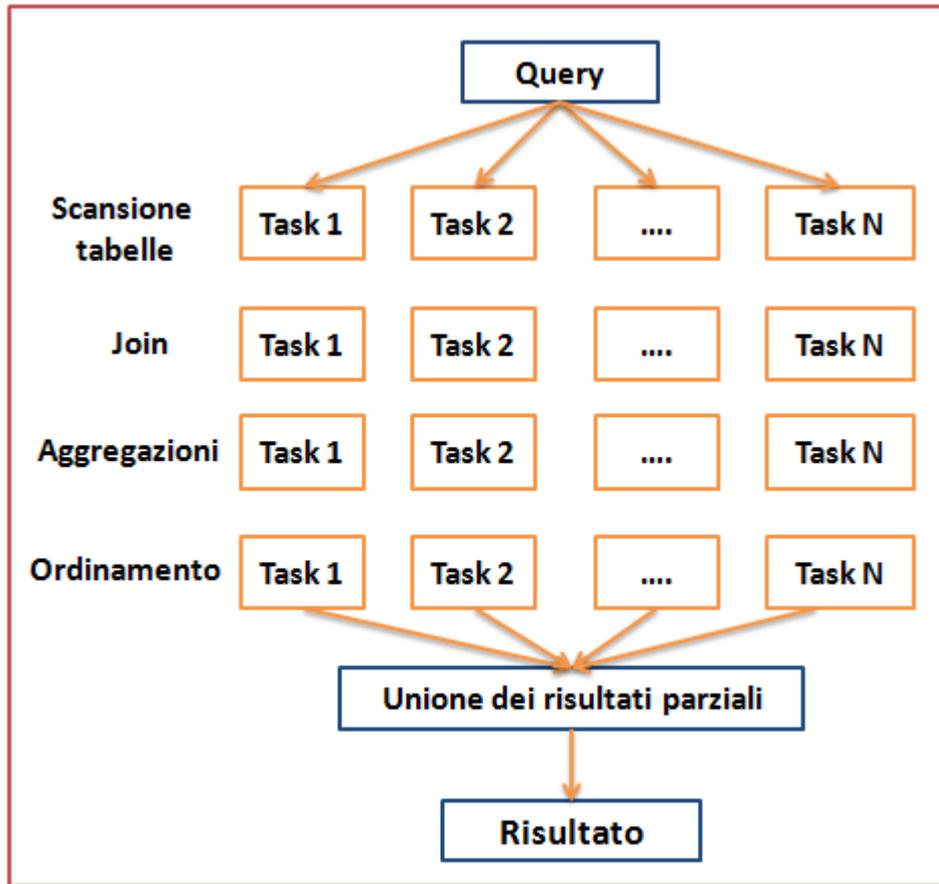
shared nothing
o
parallel everything

Tecnologie per i Big Data - MMP

- **Funzionamento** dei sistemi **MMP** composto solitamente da **due fasi**:
 1. **Fase di comunicazione**, *definizione dei processi da elaborare.*
 2. **Fase di calcolo**
- Questi sistemi hanno una forte **impostazione distribuita**, la **rete che connette i vari nodi è estremamente importante** e deve essere regolata in modo che non si presentino dei colli di bottiglia.
- Una corretta e attenta **gestione delle repliche dei dati tra i nodi** permette di **raggiungere e garantire un'alta disponibilità**.

- Architetture **shared-nothing** (o parallel everything)
- Risorse **HW dedicate** (CPU, RAM, dischi)
- SW ad-hoc per sfruttare la **parallelizzazione delle operazioni**

Tecnologie per i Big Data - MMP



Schema di esecuzione query su architettura MMP

- Una **query SQL** è **suddivisa** in una **serie di operazioni** che vengono **eseguite** in *parte in sequenza* e in *parte in parallelo*.

- Ogni **operazione** è a sua volta **scomposta in N task**, in modo da sfruttare *l'architettura parallela* *minimizzando la contesa di risorse*.

- Si può osservare che gli *unici punti critici* sono all'**inizio della query** e al **termine del processo**, nel momento in cui vanno riuniti i risultati provenienti da diversi nodi.

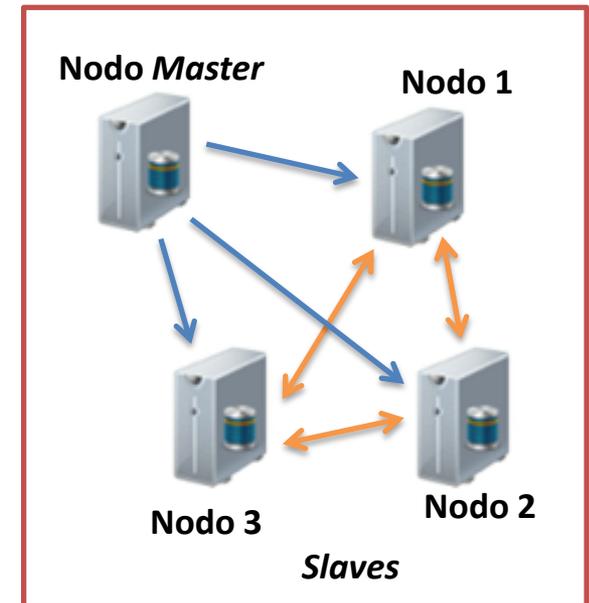
Tecnologie per i Big Data – MMP Vs. Hadoop

- I **sistemi MPP** sono uno strumento **ideale** per la **gestione di dati** che presentano una **rappresentazione tabellare**, e riescono anche a far fronte al fattore **Volume** dei Big Data.
- Rispetto alla **Variabilità** (assenza di struttura), i sistemi MPP presentano gli stessi limiti degli RDBMS tradizionali.



Hadoop è una piattaforma software open source finalizzata al calcolo su un sistema di computer autonomi (nodi) collegati in rete.

- **HDFS: file system distribuito** per salvare dati su un cluster di computer.
- **MapReduce: paradigma di programmazione** realizzato per offrire scalabilità e tolleranza ai guasti.



Aspetti Architetture

Dimensione del Cluster

- Il **numero di nodi** in ogni cluster influisce i **tempi di completamento di un processo** (*job, funzione, etc.*).
- Un **numero maggiore di nodi** corrisponde ad un **tempo di completamento minore** di un job.

Dati di input

- **Maggiore** è la **dimensione del dataset** inizialmente in ingresso, più sarà il **tempo di elaborazione dei dati** e di **produzione dei risultati**.

Data Node

- Un **numero maggiore di nodi in un cluster** corrisponde ad una **maggiore potenza computazionale** e quantità di **memoria disponibile**, ciò comporta un **minor tempo nel completamento di un job**.

Localizzazione dei Dati

- **Non è possibile assicurare** che i **dati** siano **disponibili localmente su un nodo**. Spesso è necessario **ricercare i blocchi di dati** da elaborare con un significativo **aumento dei tempi di completamento**.

Network

- Il **network influisce in modo importante sulle performance** finali di un sistema di gestione dei Big Data.
- Le **connessioni tra diversi nodi e cluster** possono risultare utili nelle operazioni di lettura/scrittura.
- E' ideale un **network che assicuri un'elevata disponibilità e resilienza**, ottenute con meccanismi di ridondanza.

Aspetti Architettureali

CPU

- Un **numero maggiore di processi da eseguire** si traduce in un **utilizzo più intenso della CPU**.
- La potenza della CPU a disposizione influisce quindi sulle performance dell'intero sistema.

Memoria

- In caso di **applicazioni "memory-intensive"** è consigliabile avere una **quantità di memoria su ciascun nodo server che sia in grado di coprire le esigenze dell'intero cluster.**
- Spesso 2-4 GB di memoria su ciascun server potrebbero **non risultare sufficienti** (es. SiiMobility 12GB memoria).