Gennaio 2020

Big Data: architetture tecnologiche e soluzioni reali

PwC's New Ventures

pwc

Agenda

01 Big Data

Cosa sono e come riconoscerli
Casi d'uso di riferimento

02 Sistemi Distribuiti

Approccio tradizionale
Evoluzioni nei sistemi distribuiti

03 Apache Hadoop

Caratteristiche principali
Componenti di base

Agenda

01 Big Data

Cosa sono e come riconoscerli
Casi d'uso di riferimento

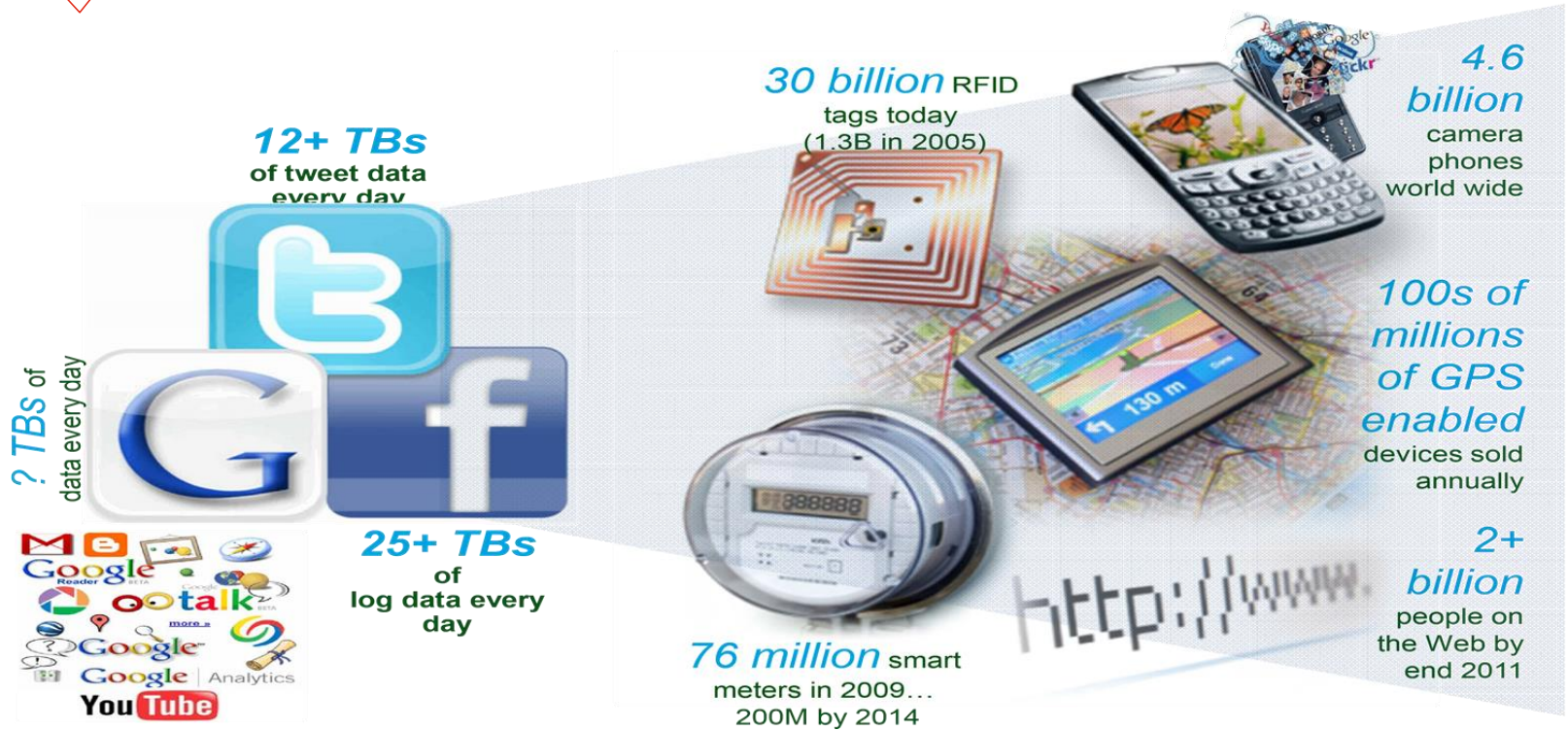
02 Sistemi Distribuiti

Approccio tradizionale
Evoluzioni nei sistemi distribuiti

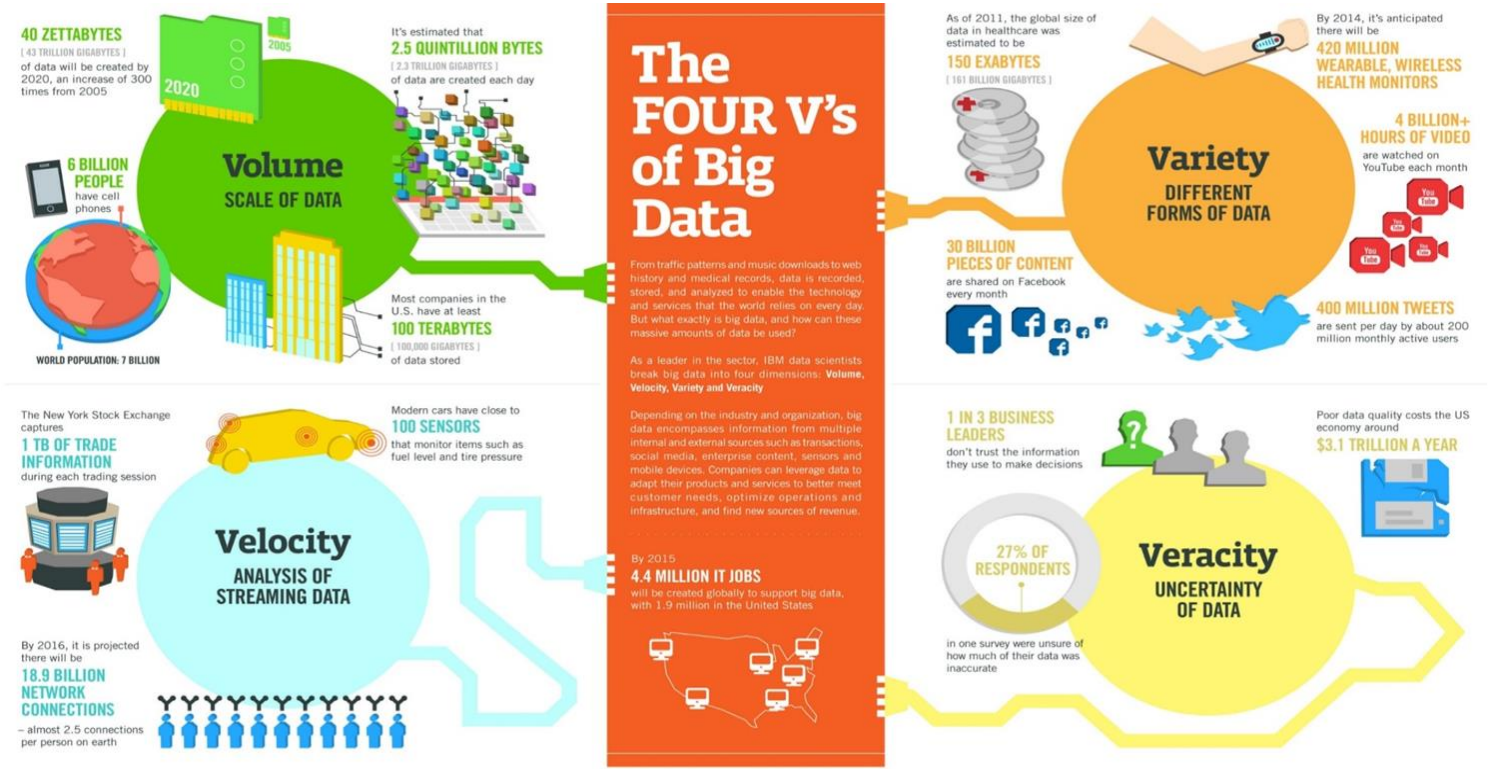
03 Apache Hadoop

Caratteristiche principali
Componenti di base

Big Data



Big Data: le 4 V



Big Data: Volume

Ogni giorno

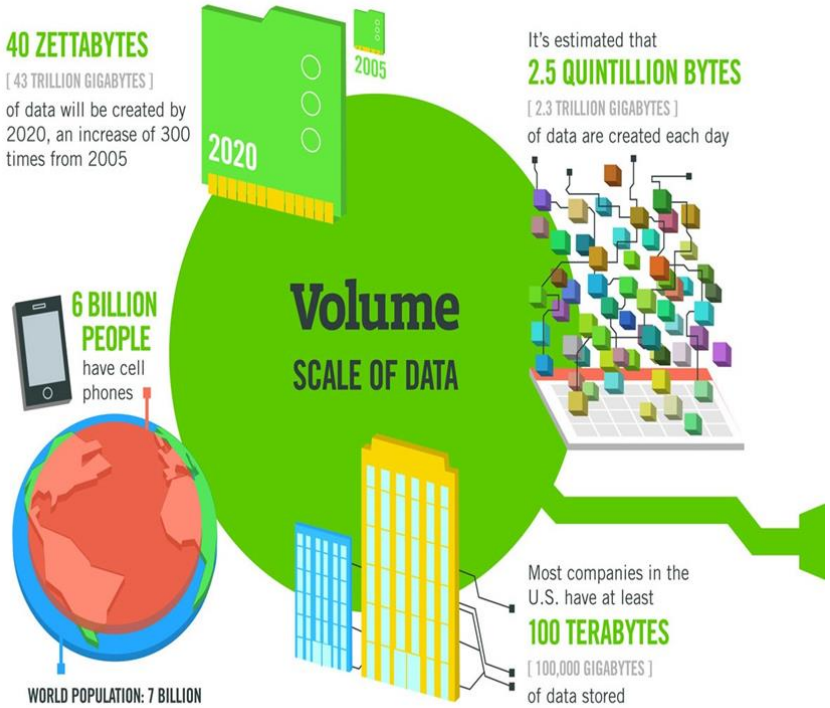
- Generati circa 2,5 Quintilioni di byte (10^18)
- Google processa 24 Petabytes di dati
- Twitter processa 340 milioni di messaggi
- Facebook immagazzina 2,7 miliardi di commenti e ‘Likes’

Ogni minuto

- Foursquare processa più di 2,000 check-in
- Più di 200 milioni di email vengono inviate
- 0,5 milioni di tweet

Ogni secondo

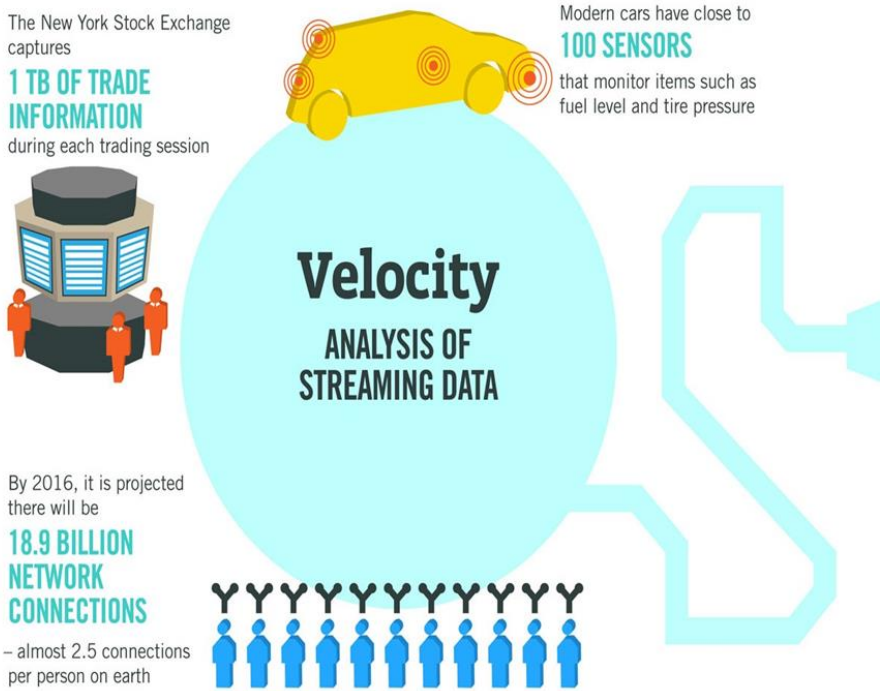
- Le banche processano più di 10 mila transazioni da carte di credito
- Ogni persona genera circa 1,7 MB di contenuti



Big Data: Velocity

Velocity non è solamente la misura di velocità di produzione delle informazioni, ma anche la velocità con cui queste informazioni devono essere consumate.

- I processi di produzione e pubblicazione di informazioni sono sempre più automatizzati
- I sistemi sono interconnessi e hanno necessità di un continuo scambio di informazioni
- Dall'avvento dei social media le interazioni tra gli utenti sono esponenzialmente incrementate
- La vita delle informazioni è molto dinamica:
  - Creazione
  - Modifica
  - Cancellazione

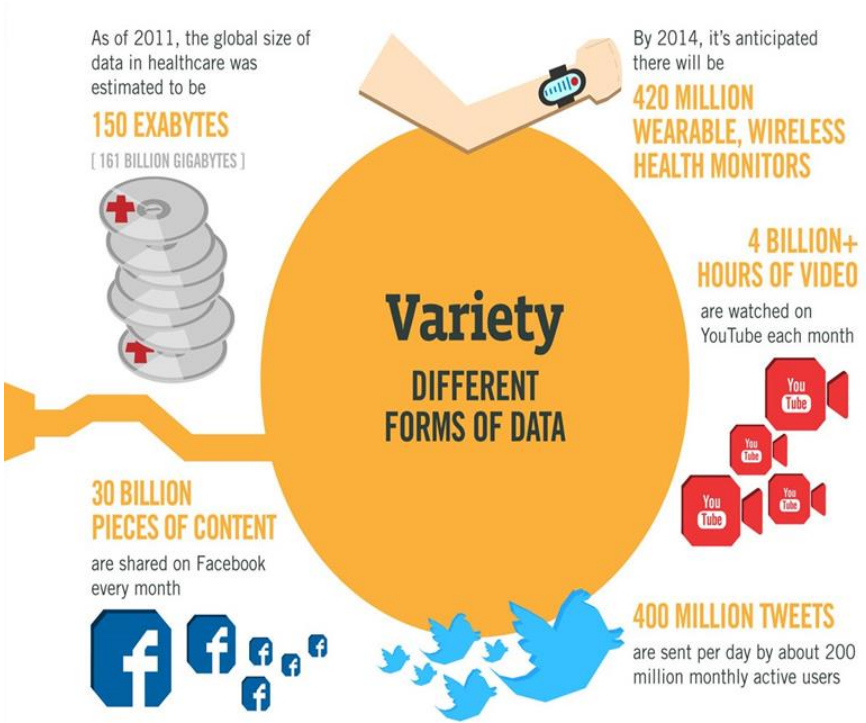


Big Data: Variety

La varietà di tipologia di dati e informazioni non è rappresentabile con un classico modello relazionale.

Quale tipi di dati stiamo producendo?

- Tweet, Post et simili
- Internet click
- RFID e Sensori
- Immagini, audio e video
- Messaggi di testo
- Connessioni su social network
- Log di applicazioni

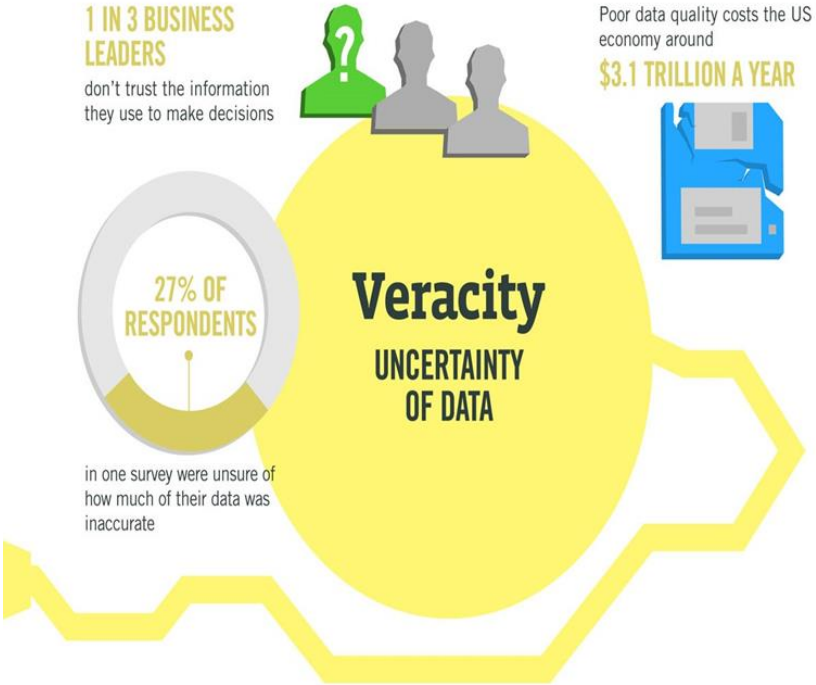


Big Data: Veracity

La consistenza e la completezza dei dati non possono essere garantite perché operazioni di data quality e data cleansing sono critiche in ambito Big Data.

Rinunciare all'esattezza:

- L'incremento dei volumi comporta inesattezza
- L'esattezza può essere sacrificata in favore dell'ampiezza o della frequenza
- Accettare l'inesattezza di pattern estratti o nella struttura dei dati



Big Data: Value

«La vera rivoluzione non sta nelle tecnologie per elaborare i dati, ma nei dati in sé e nel modo in cui li usiamo»

(Mayer-Schoenberger & Cukier 2013)

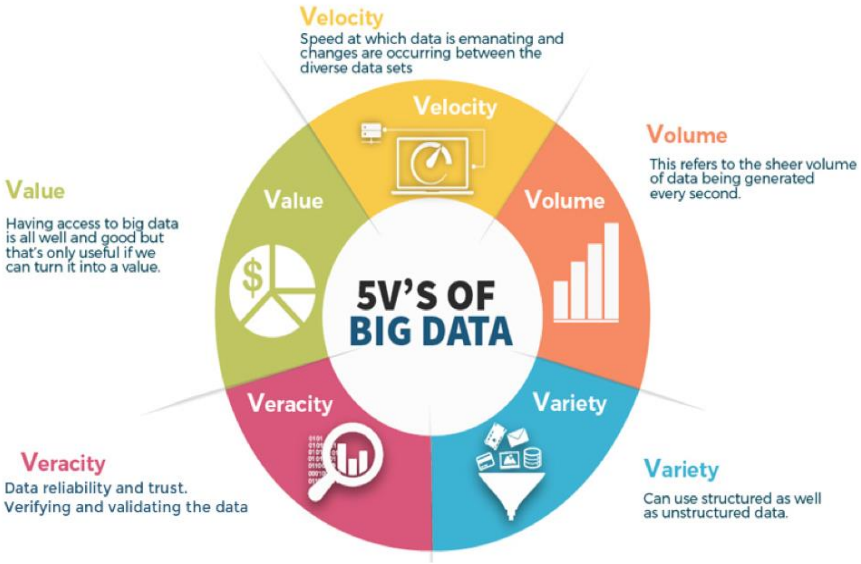
Analizzare tutti i dati disponibili

- Assuefazione al campionamento statistico autolimitazione nell'uso delle informazioni
- Il campionamento è solo un ripiego
- È poco utile quando si vuole scavare in profondità

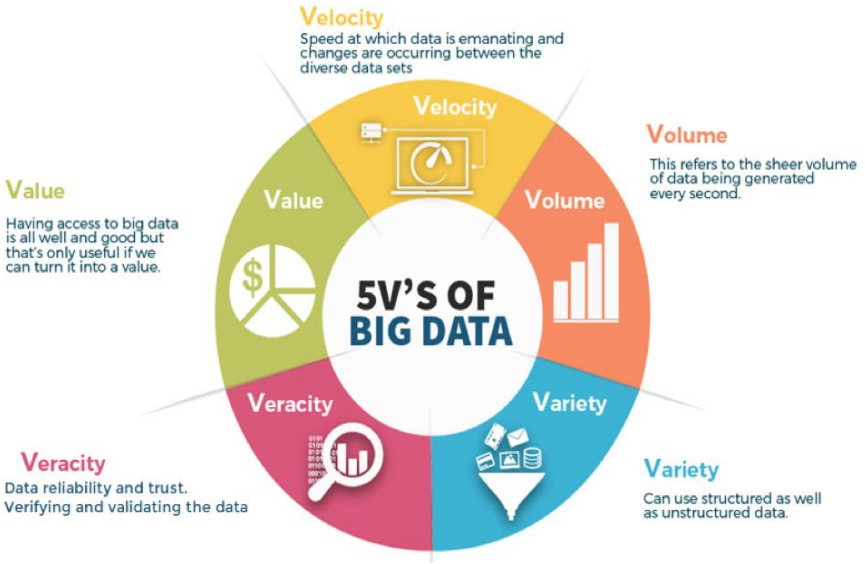
Rinunciare alla causalità in favore della correlazione

- Non conta sapere perché vendo un libro online, ma cosa fa aumentare le vendite
    - In previsione di un uragano aumentano le vendite di torce elettriche, ma anche di merendine e dolci
- La dimostrazione di una causalità è molto più costosa della individuazione di una correlazione
    - Auto usate arancioni sono meno soggette ad avere difetti

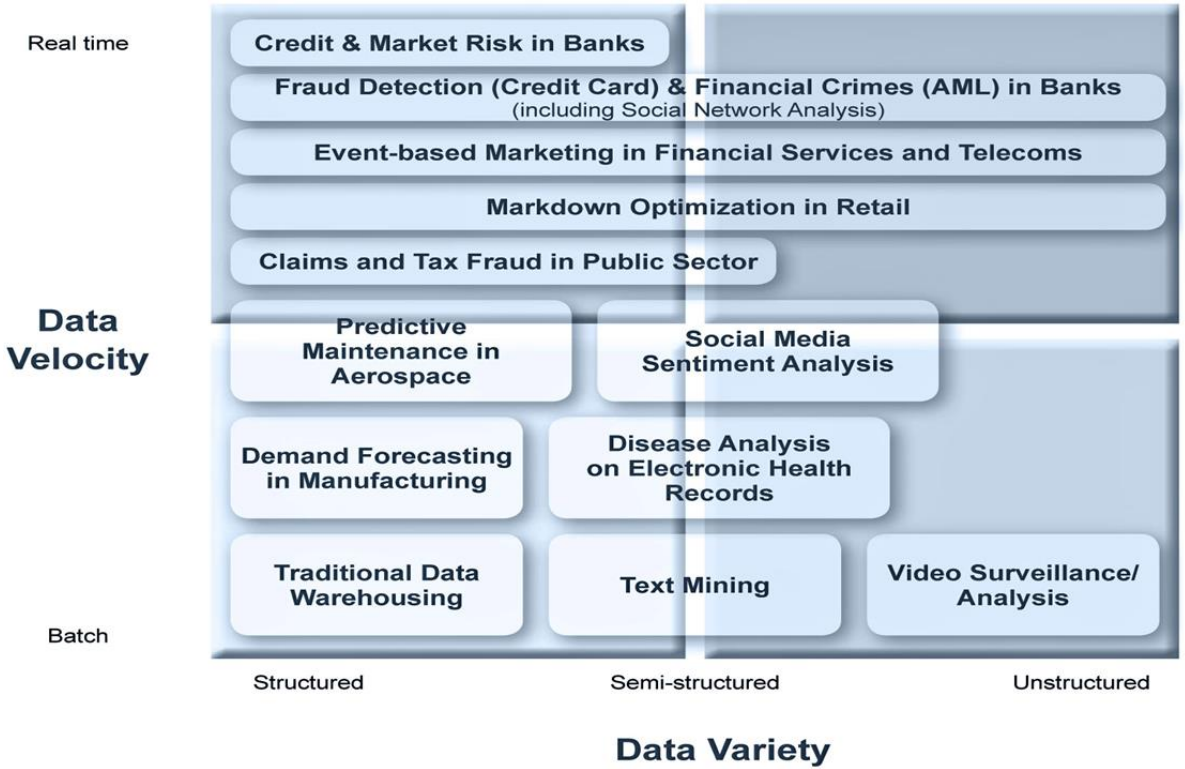
Big Data: continua evoluzione…



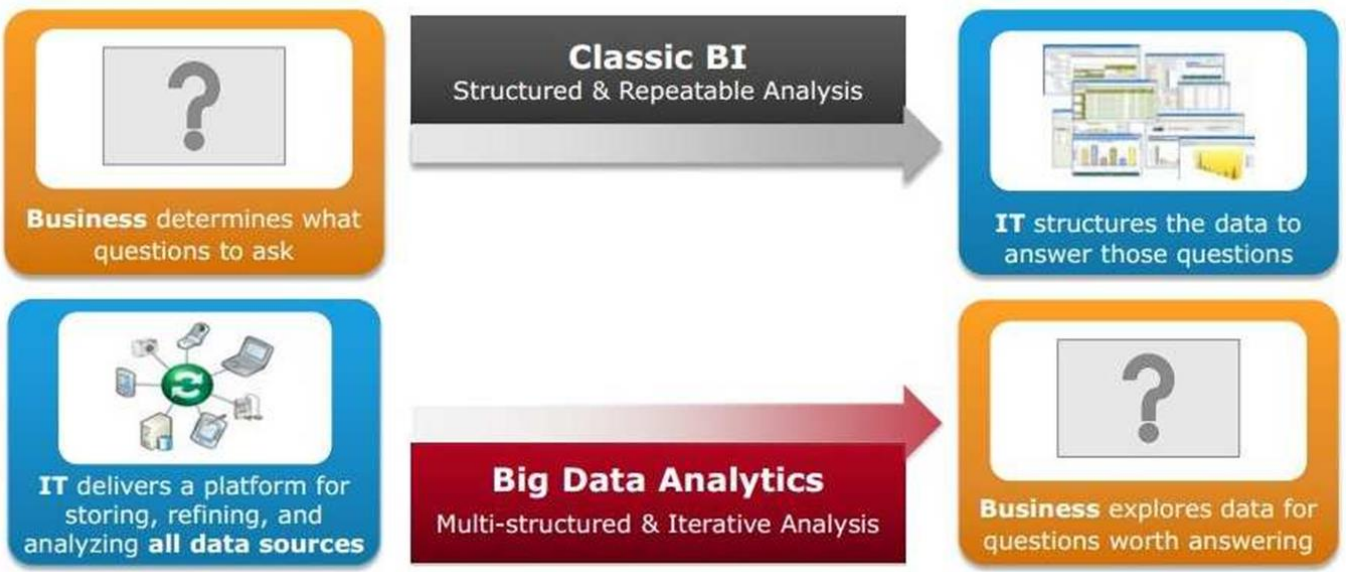
Big Data: continua evoluzione…



Big Data: casi d'uso



Big Data vs Business Intelligence (BI)



Agenda

01 Big Data

Cosa sono e come riconoscerli
Casi d'uso di riferimento

02 Sistemi Distribuiti

Approccio tradizionale
Evoluzioni nei sistemi distribuiti

03 Apache Hadoop

Caratteristiche principali
Componenti di base

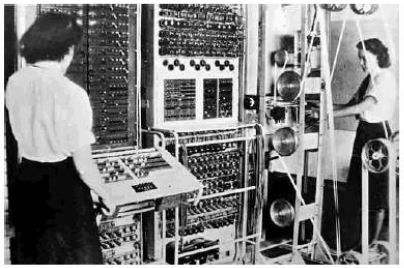
Computazione massiva

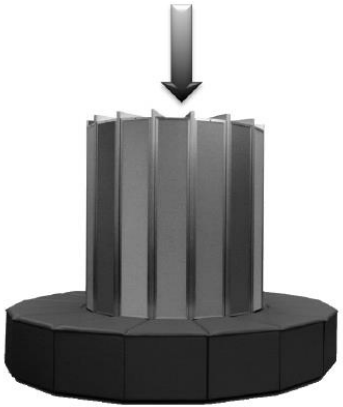
Approccio tradizionale -> Computazione basata sulla quantità e performance dei porcessori

- Quantità relativamente piccole di dati
- Capacità di svolgere elaborazioni complesse

Soluzione iniziale -> Utilizzare macchine più performanti

- Aggiungere processori
- Aumentare la quantità di memoria disponibile

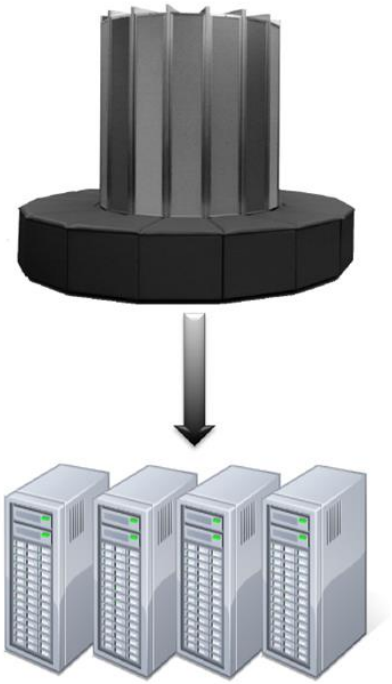




Sistemi distribuiti

"In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, we didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for *more systems* of computers."

– Grace Hopper



Soluzione migliore -> Aumentare il numero di macchine per l'elaborazione

- Creare sistemi di computazione distribuita
- Utilizzare cluster di macchine per l'esecuzione di singoli task di elaborazione

Sistemi distribuiti: caratteristiche principali

Il sistema deve supportare fallimenti parziali

- Per il fallimento di un componente non deve fallire tutto il sistema

Dopo un fallimento non si deve avere nessuna perdita di dati

- Se un componente del sistema fallisce, il suo carico di lavoro deve essere eseguito da un'altra unità del sistema ancora funzionante

Recupero dei componenti

- Se un componente del sistema fallisce e poi recupera, dovrà esser capace di reintegrarsi nel sistema

Consistenza

- Il fallimento dei componenti durante l'esecuzione di un processo non affliggerà il risultato finale del processo stesso

Scalabilità

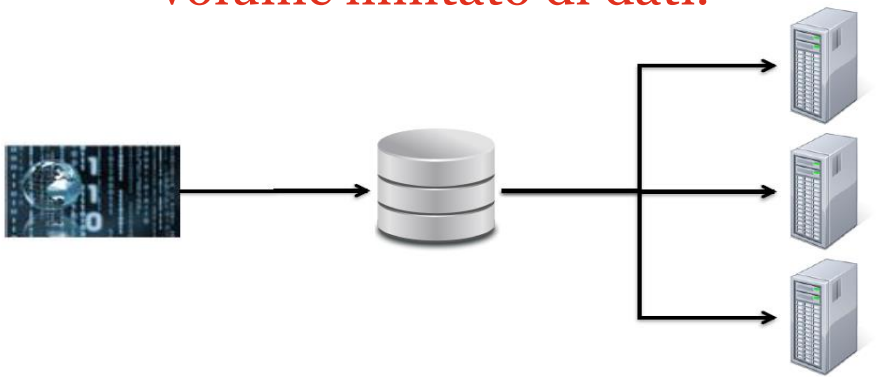
- Deve essere possibile aggiungere risorse fisiche al sistema al presentarsi di nuove esigenze

Sistemi distribuiti tradizionali

Modalità operative dei sistemi distribuiti tradizionali:

- I dati sono memorizzati all'interno di un data-storage centralizzato
- I dati vengono copiati a runtime all'interno delle macchine che effettueranno l'elaborazione

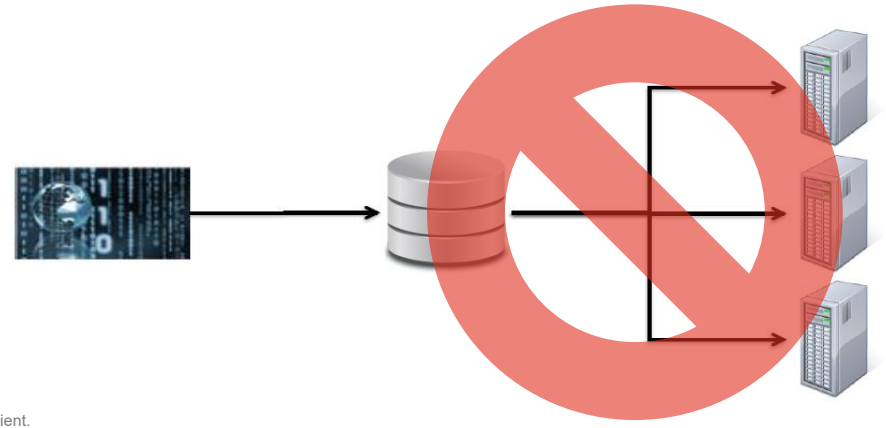
Questa soluzione è valida solo nel caso in cui l'elaborazione prevede un volume limitato di dati.



Sistemi distribuiti tradizionali

Le attuali necessità operative contemplano la gestione di volumi di dati sensibilmente maggiori...

È necessario definire un nuovo approccio tecnologico-architetturale!!



Agenda

01 Big Data

Cosa sono e come riconoscerli
Casi d'uso di riferimento

02 Sistemi Distribuiti

Approccio tradizionale
Evoluzioni nei sistemi distribuiti

03 Apache Hadoop

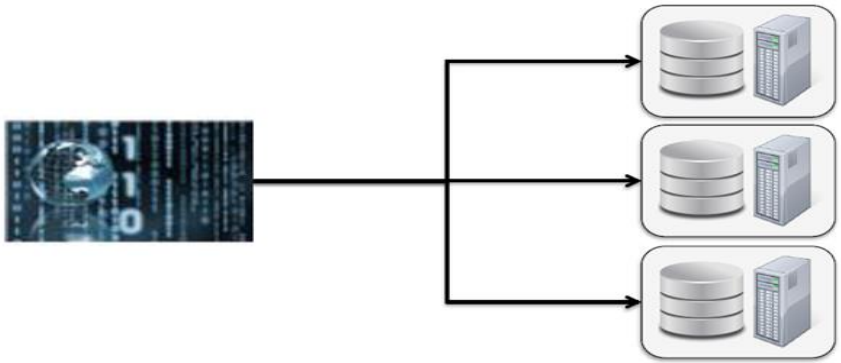
Caratteristiche principali
Componenti di base

Apache Hadoop

- Framework open-source con capacità di memorizzazione, elaborazione ed analisi di *large-scale dataset*
- Basato su studi scientifici effettuati da settore R&D di Google, pubblicati nel 2003/04
- Creatore principale Doug Cutting (creatore anche di Nutch e Lucene)

Nuovo approccio alla computazione distribuita

- Distribuzione dei dati in fase di memorizzazione dei dati
- Esecuzione della computazione dove i dati risiedono



Apache Hadoop: caratteristiche principali

Applicazioni scritte in codice di alto livello

- Gli sviluppatori non si devono preoccupare della programmazione di rete, di dipendenze temporali o delle infrastrutture di basso livello
- I nodi comunicano tra loro il meno possibile (riduzione overhead di rete)

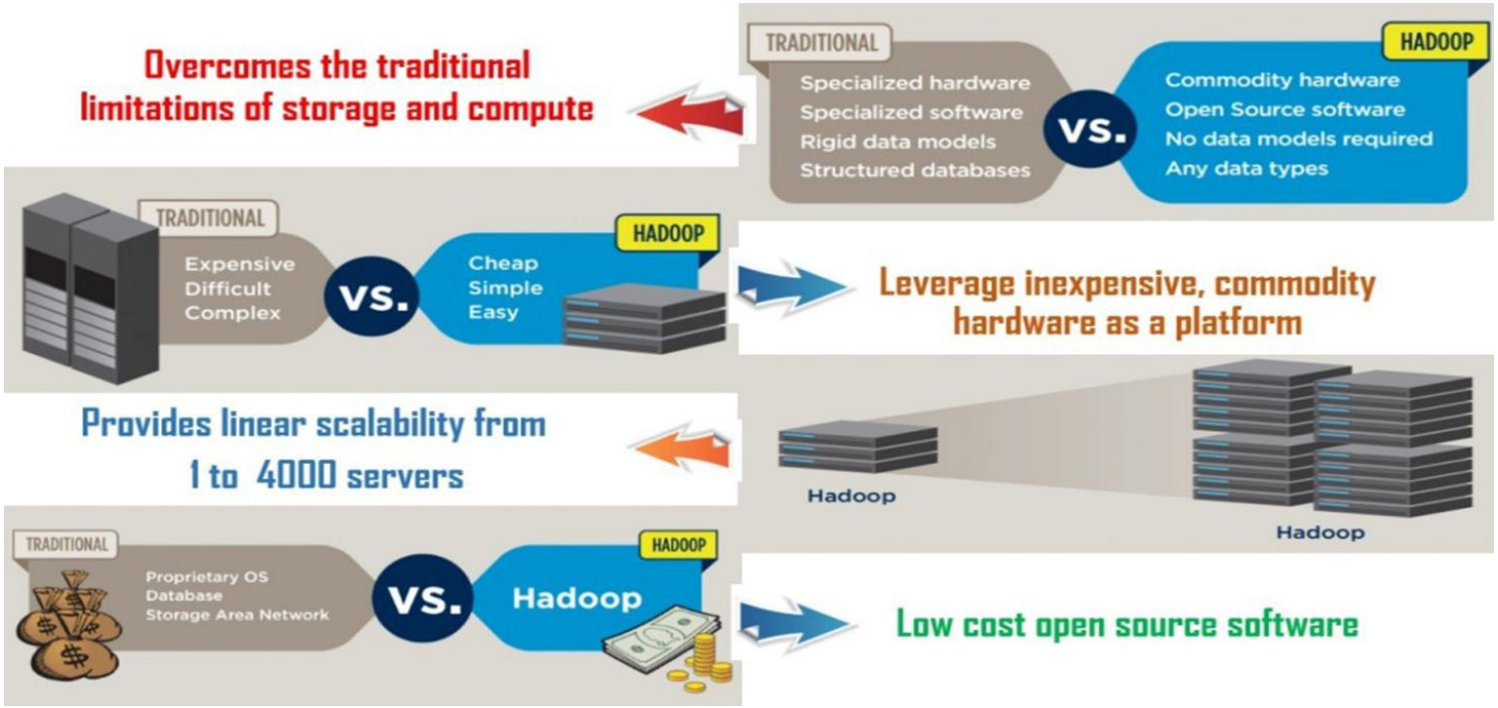
I dati vengono diffusi fra le macchine in anticipo

- Le computazioni avvengono possibilmente dove i dati sono memorizzati
- I dati sono replicati n-volte per migliorare la disponibilità e l'affidabilità

Tolleranza ai guasti

- Se un nodo fallisce, il master individuerà quel fallimento e riassegnerà il carico di lavoro a un nodo differente del sistema
- Se un nodo fallito riparte, è automaticamente aggiunto al sistema e riceve l'assegnazione di nuovi task
- Se un nodo sembra essere lento, il master può in maniera ridondante lanciare l'esecuzione di un'altra istanza dello stesso task

Apache Hadoop: vantaggi



Apache Hadoop: componenti di base

HDFS

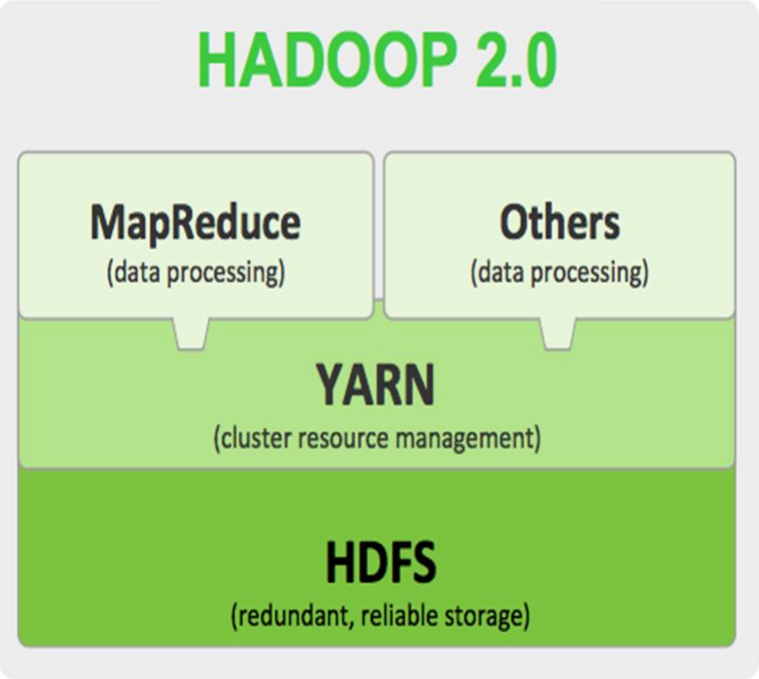
- Filesystem Java-written
- Fornisce memorizzazione ridondante su nodi differenti per assicurare affidabilità e disponibilità
- Ottimizzato per letture di grandi volumi di dati

YARN

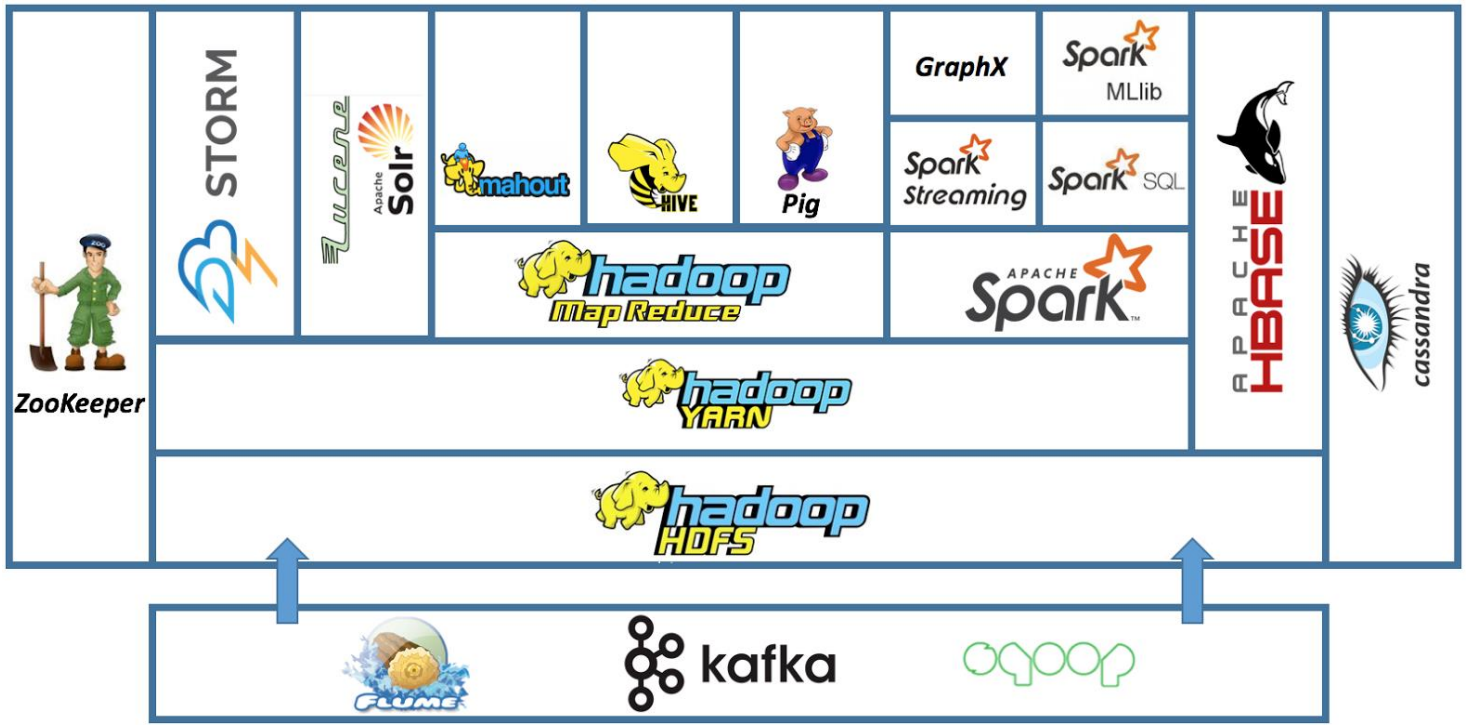
- Piattaforma utilizzata per la gestione delle risorse computazionali e di storage
- Rresponsabile dell'utilizzo delle risorse fisiche per le applicazioni definite dagli utenti

MapReduce

- paradigma di programmazione per il processamento di grandi dataset
- Elimina tutte le problematiche di gestione agli sviluppatori relative a temi di parallelizzazione e distribuzione automatica
- Fault-tolerance



Apache Hadoop: l'ecosistema



Thank you!