

Academy BigData & Spark-Scala

HADOOP

dott. Giampiero ROSA



Indice

#1 Il Framework Hadoop e l'Architettura

#2 Hadoop vs RDBMS

#3 Sistemi computazionali Distributi

#4 Componenti aggiuntive di Hadoop

Indice

#1

Il Framework Hadoop e l'Architettura

#2

Hadoop vs RDBMS

#3

Sistemi computazionali Distributi

#4

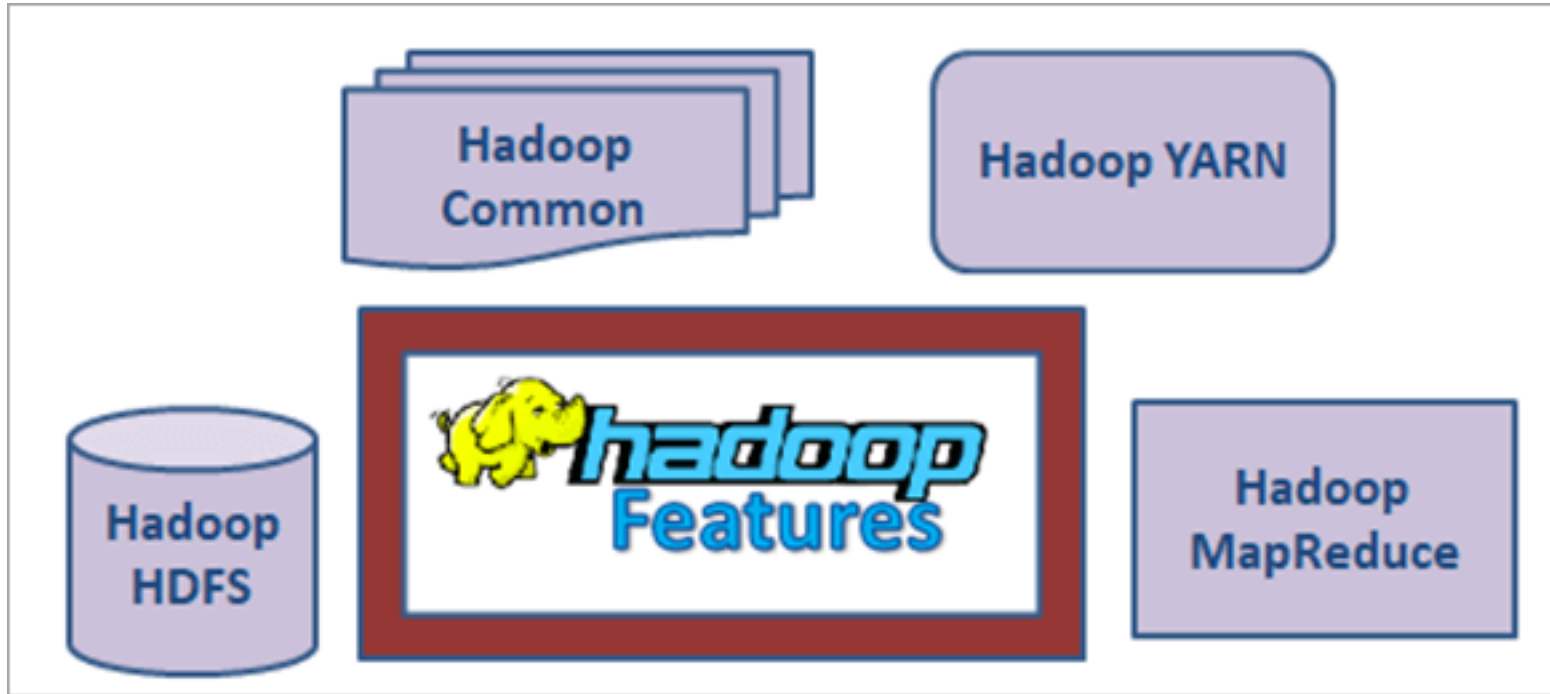
Componenti aggiuntive di Hadoop

Il Framework Hadoop

- **Apache Hadoop** fornisce un framework open source che consente l'elaborazione distribuita di grandi set di dati su cluster di risorse di elaborazione. La sua progettazione consente di scalare da server singoli a migliaia di server, ciascuno dei quali offre funzionalità di elaborazione e di storage locali
- Permette:
 - l'archiviazione e l'elaborazione distribuita di una grande mole di dati
 - di gestire tutti i tipi di dati (strutturati, non strutturati e semi-strutturati)
 - una velocità di elaborazione dei dati molto elevata
 - di creare l'ambiente con un hardware di configurazione economico, semplice e leggero
 - di archiviare dati in formato di **replica** su più macchine in modo tale che se si dovesse verificare un problema o un disastro in una delle posizioni in cui risiedono i dati, ci sarebbe una copia duplicata di quei dati disponibile.
 - di ottenere risultati affidabili poiché dispone di un sistema di tolleranza ai guasti molto elevato

Architettura di Hadoop

Si propone una macro architettura di Hadoop per entrare nei concetti e moduli



Il Framework Hadoop



Hadoop Common

è un layer a **livello software** che contiene i file **jar** e gli **script** necessari per avviare Hadoop.

Contiene inoltre le **librerie** e le **utility** necessarie per tutti gli altri moduli di Hadoop e per fornire accesso al file system di Hadoop.

II Framework Hadoop



Hadoop HDFS (Hadoop Distributed File System)

- È un file system distribuito tra i nodi all'interno di un cluster, che fornisce la modalità di accesso ai dati.
- Consente di archiviare enormi quantitativi di dati in vari formati e di distribuirli in un cluster Hadoop.
- Accetta dati in qualsiasi formato, strutturati e non strutturati

II Framework Hadoop



Hadoop YARN

(Yet Another Resource Negotiator)

Framework responsabile:

- della gestione delle risorse informatiche nei cluster (memoria/CPU/storage)
- dell'utilizzo delle risorse per la pianificazione delle soluzioni applicative degli utenti
- della creazione delle applicazioni o delle infrastrutture per il calcolo distribuito



Hadoop MAP-REDUCE

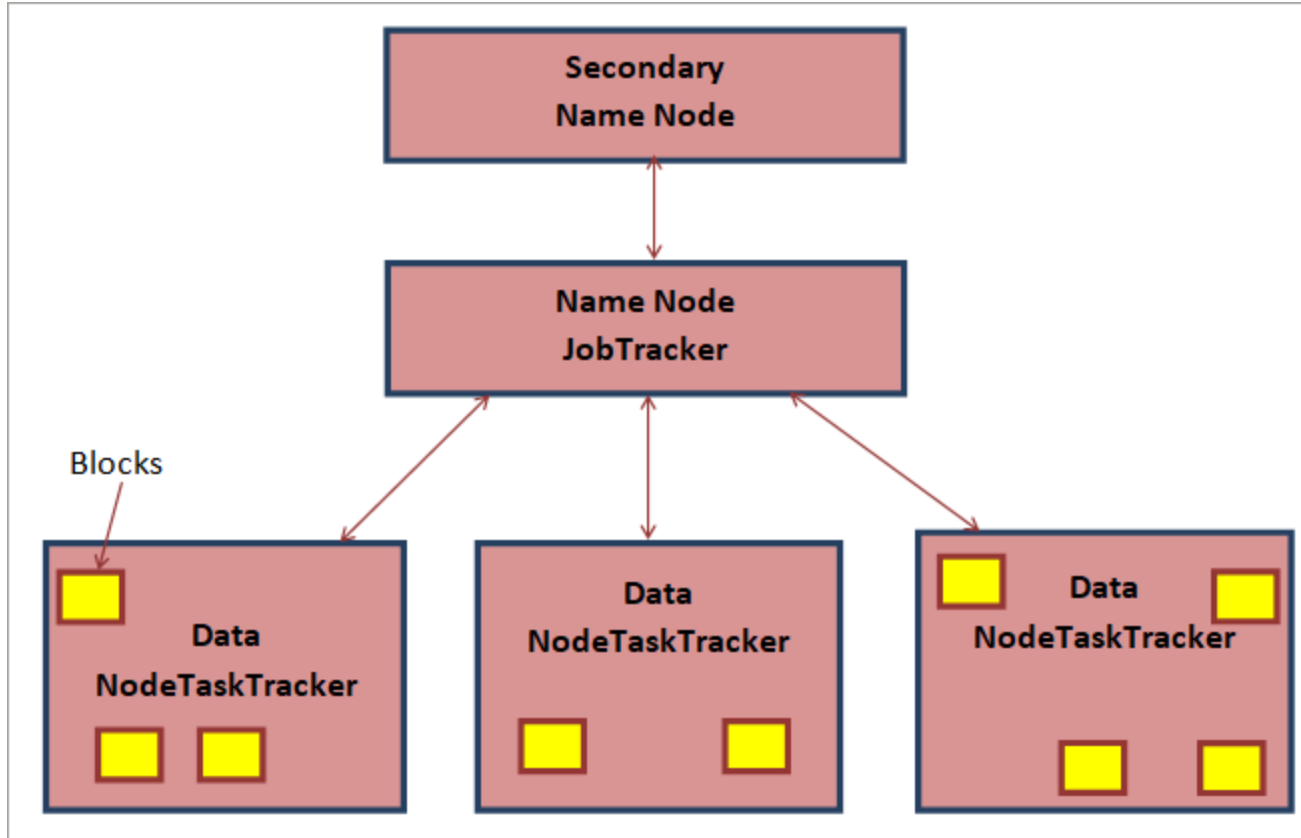
È la caratteristica principale di Hadoop, che si basa sul framework prodotto da Google.

Se implementato, permette di realizzare sistemi di computazione parallela e distribuita di grandi quantità di dati lavorando secondo il principio del «divide-et-impera».

È costituito da due fasi:

- **fase map:** gli input vengono presi e ripartiti in sotto-problemi più piccoli e poi distribuiti per l'elaborazione
- **fase reduce:** trasporta le risposte a tutti i sotto-problemi e le abbina tra loro per produrre degli output.

Hadoop – aspetto fisico



Il Framework Hadoop - terminologie

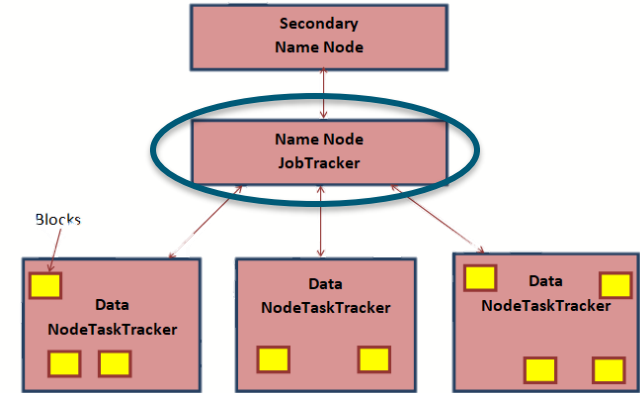
Name Node (Nome nodo)

Il «**Name Node**» è il nodo master in HDFS e:

- ha un contatto diretto con il client
- è responsabile solo dell'accesso ai dati o ai file dal client
- tiene traccia di tutte le transazioni
- tiene traccia di tutte le modifiche apportate ai file.

Esso contiene:

- i vari metadati di HDFS, come le informazioni sui file, la struttura delle directory, le informazioni sui blocchi e tutte le informazioni di Data Node, ecc.
- i dettagli del numero di blocchi presenti per ogni Data Node
- le posizioni dei vari «nodo dati» in cui sono archiviati i dati, le repliche e altri dettagli
- i dettagli della pianificazione del lavoro nel cluster
- il **JobTracker**, che a sua volta contiene i **TaskTracker** dei vari Data Node



II Framework Hadoop - terminologie

Data Node (Nodo dati)

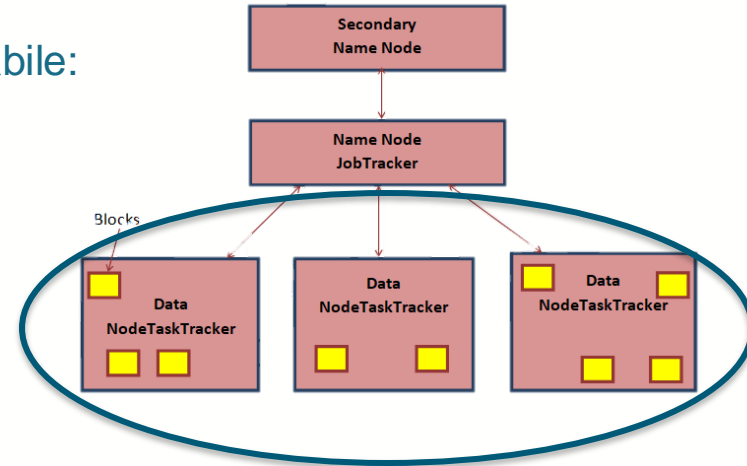
Il «**Data Node**» è il nodo slave in HDFS ed è responsabile:

- dell'archiviazione dei dati
- dell'elaborazione effettive dei dati
- del dividere il lavoro in blocchi (tre) e archiviarlo in diversi nodi di dati.

Ha il **TaskTracker**, che possiede:

- le informazioni complete di ogni blocco
- quale blocco è responsabile di una certa attività,
- quale blocco ha completato l'attività, ecc.

È in contatto con il Name Node al quale comunica informazioni di vario genere, quali ad esempio, la fine dell'elaborazione dei dati, l'avvio del Data Node, ecc.



II Framework Hadoop - terminologie

Secondary Node Name

Il «**Secondary Node Name**», chiamato anche «**CheckPointNode**» viene utilizzato principalmente in caso di tolleranza ai guasti.

Contiene un'immagine dello spazio dei nomi e i «registri di modifica» uguali al Node Name.

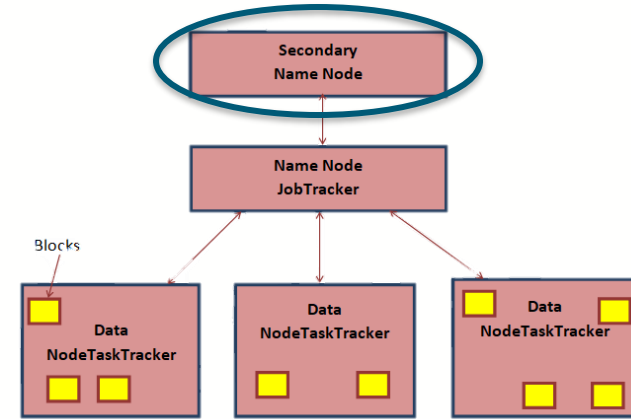
Scenari operatività (caso in cui il Node Name è inattivo):

1. Name Node si riavvia a causa di un problema, con tempo di avvio ampio.
2. caso di arresto anomalo di Name Node e rischio di perdita dei su HDFS, con difficoltà di ripristino in quanto Name Node è l'unico punto di errore.

Periodicamente procede con:

- copia dell'immagine dello spazio dei nomi
- aggiornerà i «registri di modifica» con le varie modifica dal nodo del nome.

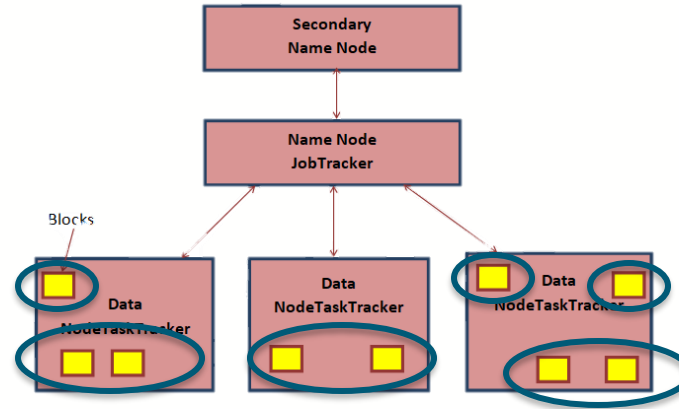
Quindi, in caso di errore del Node Name, il Secondary entra in scena e si comporta come il nodo del nome principale.



II Framework Hadoop - terminologie

Blocchi

- Il «blocco» rappresenta l'unità più piccola nell'HDFS.
- La dimensione di ogni blocco è di 128 MB.
- Hadoop può elaborare un'enorme quantità di file in quanto lo divide in piccoli blocchi. Possiamo dire che i blocchi non sono altro che i dati di un file enorme.
- I blocchi vengono salvati nei Data Node ed elaborano i dati

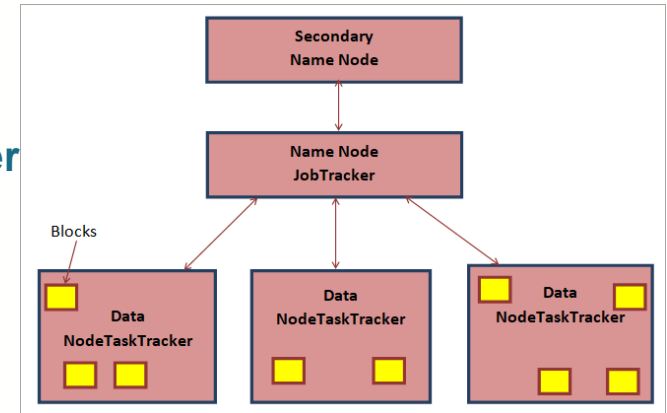


Il Framework Hadoop – overview funzionamento



- **HDFS** è il file system distribuito Hadoop utilizzato per archiviare i dati Hadoop nel cluster.
- HDFS lavora generalmente sull'elaborazione sequenziale dei dati.
- HDFS si basa sull'architettura Master-Slave.

- Tutti i metadati del cluster vengono salvati in **JobTracker**
- I dati effettivi vengono memorizzati nel TaskTracker del Data Node.

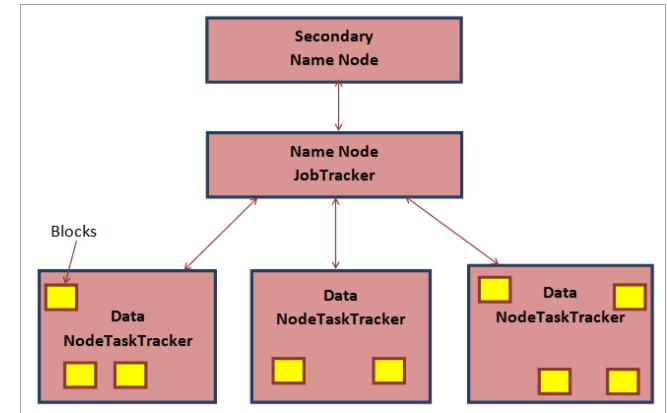


Il Framework Hadoop – overview funzionamento

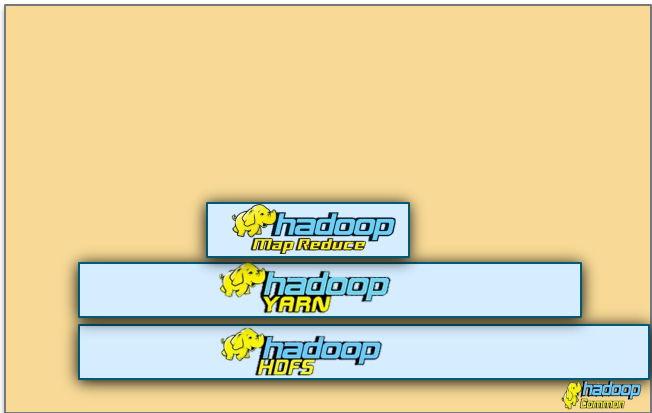


- **MapReduce** è responsabile del trattamento dei dati.
- Ogni volta che un file entra nel cluster per l'elaborazione, il primo **Data Node** lo divide in **blocchi**
- Ogni blocco contiene 64 MB di dati e può memorizzare 128 MB.
- Ogni blocco verrà **replicato** e archiviato in diversi Data Node ovunque nel cluster.

- Le varie informazioni vengono inviate al Name Node che le memorizzerà sotto forma di metadati.
- L'effettiva elaborazione dei dati sarà effettuata nel Data Node
- Il Data Node invia *heartbeat* al Name Node ogni tre secondi in modo che il Name Node abbia le informazioni su cui sta lavorando questo Data Node.

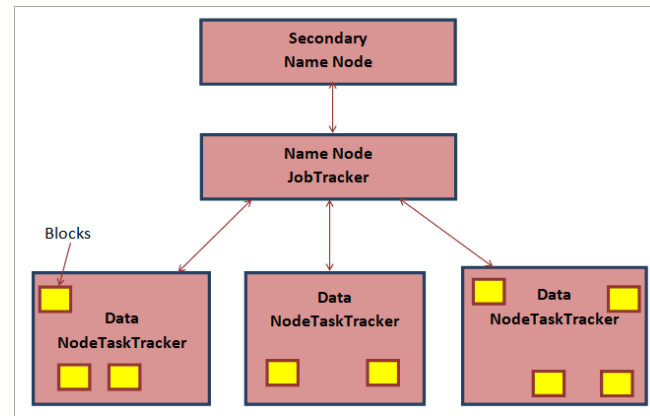


Il Framework Hadoop – overview funzionamento



- Se un Data Node non riesce a inviare heartbeat, il Name Node crea nuovamente la replica di quel blocco su un altro Data Node e avvia l'elaborazione
- Queste informazioni (o istantanee) verranno archiviate in **FsImage**
- se viene eseguita una transazione, il **registro di modifica** unisce le nuove informazioni e mantiene sempre una nuova copia dei registri

- Quando un blocco termina l'attività, il Data Node invia le informazioni al Name Node
- Il Name Node determina ed intraprende l'azione di conseguenza.
- In questo intero processo, YARN supporterà e fornirà le risorse necessarie al sistema
- Dopo l'elaborazione dei dati, i risultati verranno salvati in HDFS per ulteriori analisi



Indice

#1

Il Framework Hadoop e l'Architettura

#2

Hadoop vs RDBMS

#3

Sistemi computazionali Distributi

#4

Componenti aggiuntive di Hadoop

II RDBMS

Abbiamo visto che un **RDBMS**:

- Soluzione migliore se vogliamo **elaborare**, **archiviare** e **gestire** i nostri dati
- ha necessità che i dati siano in un formato **strutturato** per gestirli correttamente.
- presenta limitazioni e non è in grado di gestire dati se la dimensione aumenta
- ha il problema della gestione dei dati con necessità di svecchiamento, che porta a sua volta alla perdita di dati storici. Questa può causare la generazione di risultati poco accurati e poco affidabili in alcuni settori come ad esempio: previsioni del tempo, valutazioni delle vendite, ecc.
- Un altro problema con RDBMS è che se il server principale si blocca, potremmo perdere il nostro importante dati e soffrono molto

Come risolvere il problema con il mondo Big Data?

Hadoop vs DBMS

Caratteristiche	HADOOP	RDBMS
Conservazione	Capacità di stoccaggio molto elevata	Impossibile memorizzare BigData
Architettura	Hadoop è basato su HDFS, MapReduce e YARN	RDBMS si basa sulle proprietà ACID
Volume	Può gestire grandi volumi di dati	Non può gestire grandi volumi
Varietà / tipi di dati	Può gestire dati strutturati, semi-strutturati e non strutturati (video, immagini, file CSV, XML, ecc.)	Gestisce solo dati strutturati
Velocità	Elaborazione rapida di grandi quantità di dati	Molto lento durante elaborazioni di grandi quantità (supportati) di dati
Portata	Elevata produttività	Basso rendimento
Tolleranza ai guasti	Molto bene	Non è in grado di recuperare i dati persi se il server principale non funziona
Affidabile	Molto affidabile e genera rapporti storici e attuali in maniera accurata	Non affidabile in termini di BigData
Schema	I dati vengono semplicemente copiati nel file system e non è richiesta una trasformazione (<i>On-Read</i>)	Lo schema dei dati deve essere creato prima che i dati stessi vengano caricati (<i>On-Write</i>)
Struttura del dato	I dati delle colonne sono estratte durante la fase di lettura	Ogni dato da caricare deve essere trasformato nella struttura interna del database
Aggiunta dati	I nuovi dati possono essere aggiunti ed estratti in qualsiasi momento	Necessario aggiungere le nuove colonne prima che i nuovi dati per tali colonne vengano caricati

Indice

#1

Il Framework Hadoop e l'Architettura

#2

Hadoop vs RDBMS

#3

Sistemi computazionali Distributi

#4

Componenti aggiuntive di Hadoop

Sistemi Computazionali Distribuiti – definizione

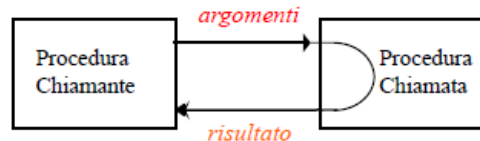
➤ Un **sistema distribuito** è

- Un sistema in cui l'elaborazione delle informazioni è distribuita su più calcolatori, anziché centralizzata su unica macchina
- è una porzione di software che assicura che un insieme di calcolatori appaiano come un unico sistema coerente agli utenti del sistema stesso [*Van Steen*]
- un insieme di calcolatori autonomi, connessi fra loro tramite una rete e un middleware di distribuzione, che permette ai computer di coordinare le loro attività e di condividere le risorse del sistema, in modo che gli utenti percepiscano il sistema come un unico servizio integrato di calcolo [*Wolfgang*]
- Un sistema in cui le componenti HW e SW sono posizionati su calcolatori collegati in rete e comunicano e coordinano le proprie azioni solo attraverso lo scambio di messaggi [*Coulouris & Dellimore*]
- Un sistema in cui il fallimento di un calcolatore di cui non conosci l'esistenza può rendere inutilizzabile il tuo calcolatore [*Lamport*]

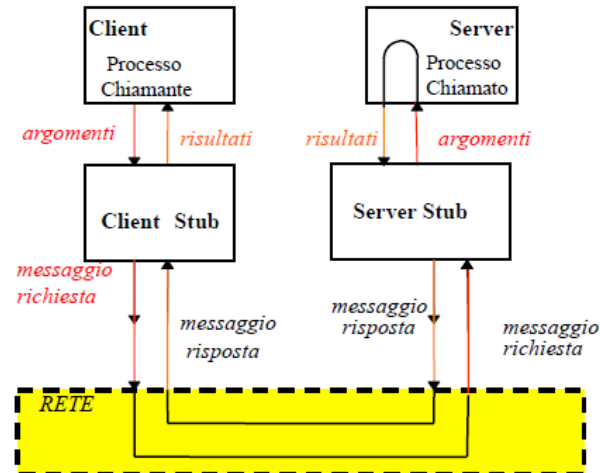
Sistemi Computazionali Distribuiti – definizione

- Un sistema distribuito è un insieme di processori indipendenti (con proprie risorse HW e SW) interconnessi da una rete di comunicazione, che cooperano per condividere alcune delle risorse ovunque distribuite.
- Si indica anche generalmente una tipologia di sistema informatico costituito da un insieme di *processi* interconnessi tra loro le cui comunicazioni avvengono attraverso scambio di opportuni messaggi. Con il termine *processo* si indica una qualsiasi entità capace di comunicare con qualsiasi altra entità e capace di eseguire una sequenza finita di operazioni distribuite. (wikipedia)

Chiamata a procedura locale



Chiamata a procedura remota



Sistemi Computazionali Distribuiti – caratteristiche

Un sistema distribuito dovrebbe soddisfare i seguenti requisiti:

- **Aperto** – deve supportare la portabilità di esecuzione e di interoperabilità secondo standard noti e riconosciuti;
- **Trasparenza** – mascherare:
 - la distribuzione dei processi
 - la distribuzione fisica delle risorse
 - le differenze di rappresentazione dei dati
 - il modo in cui gli utenti accedono alle risorse
 - dettagli e differenze dell'architettura sottostante per permettere un'agevole progettazione e programmazione;
 - la duplicazione delle risorse (migliorano le prestazioni)
 - eventuali guasti all'utente
 - la locazione fisica delle risorse.
- **Integrato** – incorporare al proprio interno sistemi e risorse senza dover utilizzare strumenti ad-hoc

Sistemi Computazionali Distribuiti – caratteristiche

Un sistema distribuito dovrebbe soddisfare i seguenti requisiti:

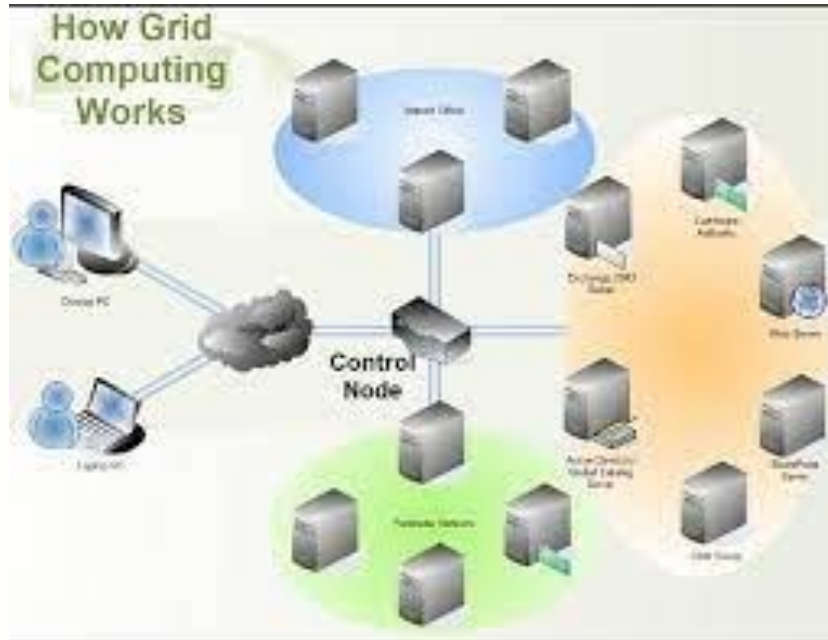
- **Flessibile** – deve rendere semplice la configurazione del sistema, l'aggiunta di nuove componenti;
- **Modulare** – ogni componente deve essere autonoma e con un certo grado di interoperabilità con il resto del sistema;
- **Scalabile** – capacità di erogare le medesime prestazioni, in termini di throughput e latenza, rispetto agli utilizzatori nonostante l'aumento del carico operativo sul sistema. Possibilità di aggiungere risorse.
- **Sicurezza** – non permettere accessi non autorizzati

Sistemi Computazionali Distribuiti – tipologia

Ci sono due tipologie di sistemi distribuiti:

➤ Sistemi Grid

È costituito da nodi tra loro eterogenei (senza alcuna restrizione sull'hardware e sul sistema operativo che ciascun nodo deve possedere) interconnessi tra loro, anche a cavallo di diverse amministrazioni, senza dover necessariamente essere vincolati alla singola rete locale, ma collegate tramite reti dati geografiche (Internet).



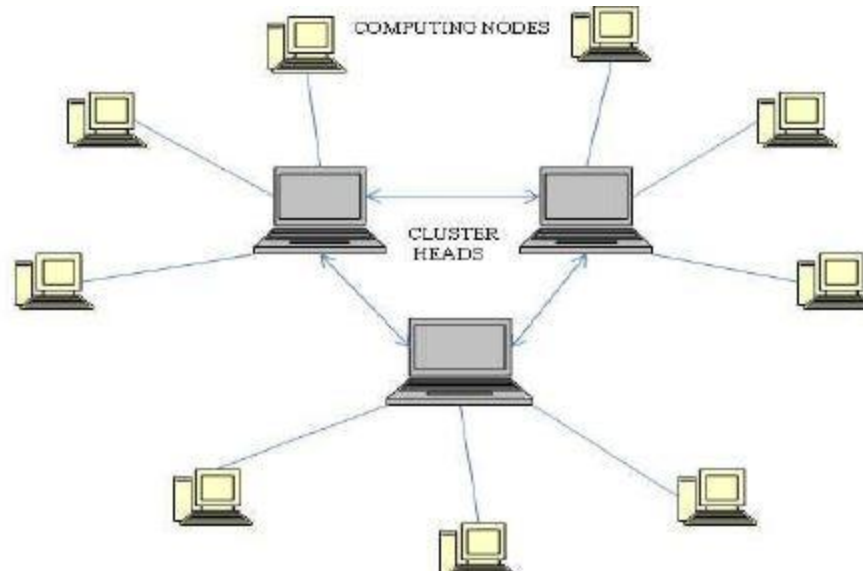
Sistemi Computazionali Distribuiti – tipologia

Ci sono due tipologie di sistemi distribuiti:

➤ Sistemi Cluster

È un insieme di computer connessi tra loro tramite una rete LAN, con l'obiettivo di distribuire elaborazioni complesse su vari computer, aumentando la potenza di calcolo del sistema e/o garantendo una maggiore disponibilità di servizio.

Vi è un *server cluster*, con alte prestazioni, che gestisce i vari *nod*i (macchine che elaborano) e suddivide su di esse il carico di lavoro.



Sistemi Computazionali Distribuiti – tipologia

Ci sono due tipologie di sistemi distribuiti:

- Sistemi Cluster
- Sistemi Grid

Un criterio di scelta tra architettura cluster e grid consiste nel valutare la quantità di dati da scambiare tra i nodi: se i dati da scambiare sono tanti, è necessario utilizzare il cluster computing; Se i dati invece sono rarefatti, è possibile utilizzare il grid computing.

Indice

#1

Il Framework Hadoop e l'Architettura

#2

Hadoop vs RDBMS

#3

Sistemi computazionali Distributi

#4

Componenti aggiuntive di Hadoop

Ecosistema Hadoop



Ecosistema Hadoop – Apache Mahout



Apache Mahout è un progetto open source di Apache Software Foundation utilizzato principalmente per la produzione di algoritmi di **apprendimento automatico** scalabili.

Si focalizza principalmente in tre aree:

- Classification (Naive Bayes distribuite e Naive Bayes complementari)
- Clustering (k-means, Fuzzy k-means, Canopy, Dirichlet e Mean-Shift)
- Recommendation;

Punto di forza: Gli algoritmi principali sono implementati su Apache Hadoop, che si basa sul paradigma MapReduce , quindi il funzionamento è ottimizzato per l'ambiente distribuito

Ecosistema Hadoop



Ecosistema Hadoop – Apache Hive



Apache Hive è uno strumento di **Data Warehouse** dell'infrastruttura di Hadoop, utilizzato per elaborare i dati strutturati.

È un progetto open source di Apache Software Foundation

Hive **non è**:

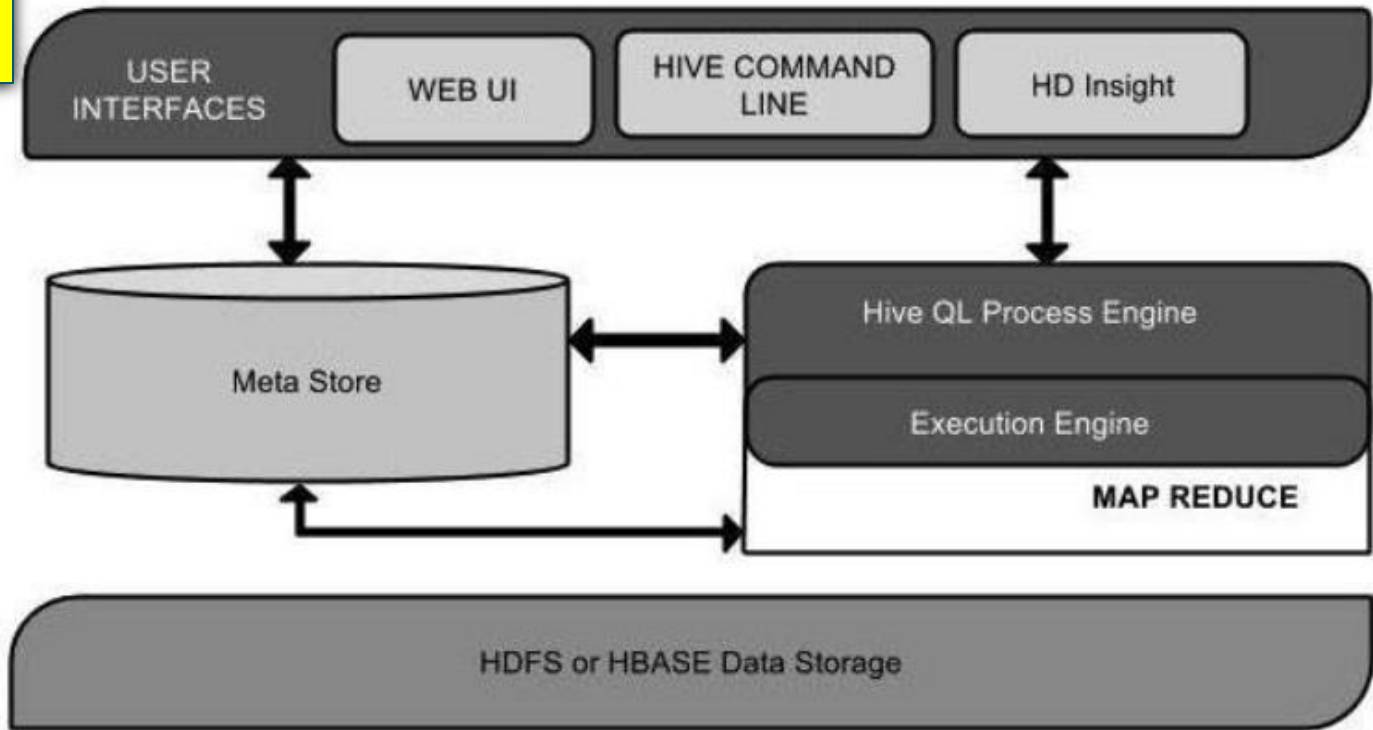
- Un database relazionale
- Un progetto OLTP (OnLine Transaction Processing)
- Un linguaggio per query in tempo reale

Caratteristiche:

- Memorizza lo schema di un DB e i dati elaborati su HDFS
- È progettato per OLAP
- Usa il linguaggio HiveQL (o HQL) simile al SQL
- Veloce, scalabile ed estensibile



Ecosistema Hadoop – Apache Hive



Ecosistema Hadoop



Ecosistema Hadoop – Apache PIG



Apache Pig è una componente software (piattaforma) utilizzata per analizzare set di dati più ampi, rappresentati come flussi di dati.

Pig è generalmente usato con Hadoop ed è **un'astrazione su MapReduce**.

- Con Pig è possibile eseguire tutte le **operazioni di manipolazione** dei dati in Hadoop ed è possibile scrivere programmi di analisi dei dati.
- Pig utilizza un linguaggio proprietario di alto livello noto come **Pig Latin**, che fornisce vari operatori per sviluppare funzioni per la lettura, la scrittura e l'elaborazione dei dati.
- Questi script vengono convertiti internamente, attraverso il **Pig Engine**, in attività di MapReduce.
- Il «**multi-query approach**» di Pig tende a ridurre notevolmente la lunghezza del codice. (Ad esempio: operazione che richiederebbe di codice di 200 LoC in Java può essere facilmente eseguita con codice meno di 10 LoC in Apache Pig).



Ecosistema Hadoop – Apache PIG



- Il Pig Latin è SQL-like language
 - Pig fornisce molti operatori incorporati per supportare le operazioni sui dati come join, filtri, ordini, ecc.
-
- Pig fornisce tipi di dati nidificati come tuple, package e map
 - Consente di creare delle pipeline e definire delle suddivisioni offrendo la possibilità di archiviare dati ovunque nella pipeline.
 - Permette di dichiarare dei piani di esecuzione.
 - Fornisce agli operatori l'esecuzione di funzioni ETL (Extract, Transform e Load)
 - Utilizzato principalmente da Data Scientist per eseguire attività che implicano l'elaborazione ad-hoc e la prototipazione rapida



Ecosistema Hadoop – Apache PIG



Apache Pig vs MapReduce

Apache PIG	MapReduce
È una piattaforma per la gestione dei dati	È un paradigma di elaborazione dei dati
Utilizza un linguaggio ad alto livello	È di basso livello e molto rigido
L'operazione di JOIN dei dati è piuttosto semplice	È complesso eseguire un'operazione di JOIN tra set di dati
È sufficiente conoscere SQL per operare	È necessario conoscere Java per lavorare
Utilizza l'approccio multi-query	È necessario programmare
Non è richiesta alcuna compilazione: le operazioni vengono convertite internamente in MapReduce	Il codice deve essere compilato per essere eseguito in MapReduce



Ecosistema Hadoop – Apache PIG



Apache Pig vs SQL

Apache PIG	SQL
Il linguaggio Pig Latin è di tipo procedurale	Il linguaggio SQL è di tipo dichiarativo
Lo Schema è opzionale	Lo Schema è obbligatorio
Il modello dei dati è di tipo nested relational	Il modello dei dati è di tipo flat relational
Offre opportunità limitate per il Query Optimization	Permette maggiormente l'ottimizzazione delle query

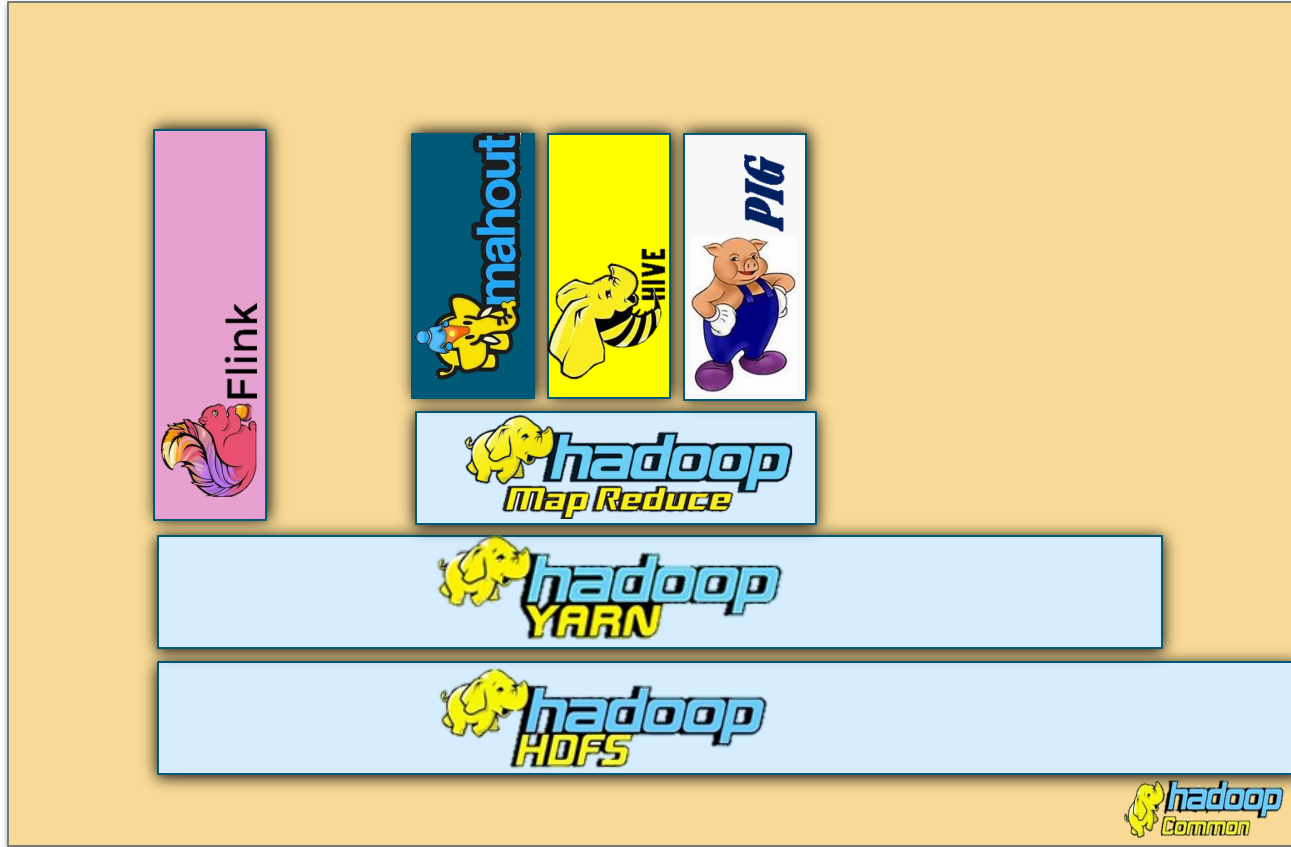
Ecosistema Hadoop – Apache PIG



Apache Pig vs Apache Hive

Apache PIG	Apache Hive
Utilizza il linguaggio proprietario Pig Latin	Utilizza il linguaggio HiveQL
Pig Latin è un linguaggio per il flusso di dati.	HiveQL è un linguaggio di elaborazione delle query
Pig Latin è un linguaggio procedurale	HiveQL è un linguaggio dichiarativo
Può gestire dati strutturati, non strutturati e semi-strutturati	È utilizzato principalmente per dati strutturati

Ecosistema Hadoop



Ecosistema Hadoop – Apache Flink



Apache Flink è un framework Open Source per applicazioni in tempo reale ad alte prestazioni e scalabili.

Oltre ad un framework è un motore di elaborazione distribuito per calcoli con stato su flussi di dati **illimitati** e **limitati**.

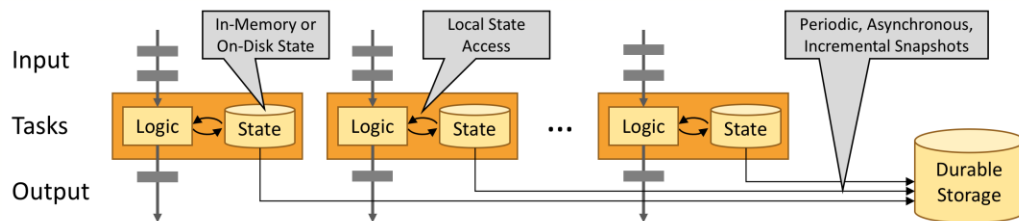
Flusso dati ILLIMITATI	Flusso dati LIMITATI
hanno un inizio ma non una fine definita	i flussi delimitati hanno un inizio e una fine definiti.
devono essere elaborati continuamente, ovvero gli eventi devono essere gestiti tempestivamente dopo essere stati inseriti	possono essere elaborati importando tutti i dati prima di eseguire qualsiasi calcolo
l'input, essendo illimitato, non sarà mai completo in nessun momento e non è possibile attendere l'arrivo di tutti i dati	l'input è completo; l'elaborazione di flussi limitati è anche nota come elaborazione batch
l'elaborazione spesso richiede che gli eventi vengano inseriti in un ordine specifico, ad esempio l'ordine in cui si sono verificati gli eventi, per poter ragionare sulla completezza del risultato	l'importazione ordinata non è necessaria per elaborare i flussi limitati perché un set di dati limitato può sempre essere ordinato

Ecosistema Hadoop – Apache Flink



- Flink è stato progettato per funzionare in tutti i comuni ambienti cluster, eseguire calcoli a velocità in-memory e su qualsiasi scala .

- È progettato per eseguire applicazioni di streaming stateful su qualsiasi scala.
- Le applicazioni sono parallelizzate in migliaia di attività che vengono distribuite ed eseguite contemporaneamente in un cluster.
- Lo stato dell'attività viene sempre mantenuto in memoria o, se insufficiente, in strutture di dati su disco efficienti nell'accesso
- Le attività eseguono tutti i calcoli accedendo allo stato locale



Ecosistema Hadoop – Apache Flink



- Flink è in grado di elaborare real time dati in streaming.
- In runtime fornisce:
 - elaborazione distribuita,
 - tolleranza agli errori,
 - alta affidabilità,
 - capacità di elaborazione iterativa nativa
- Offre API e Librerie:
 - **API Dataset**, che permette una elaborazione con caricamenti in batch
 - **API Datastream**, che si occupa dell'elaborazione del flusso dati.
 - **Flink ML**, utilizzato per l'apprendimento automatico
 - **API Gelly**, utilizzato per l'elaborazione di grafici
- Si integra facilmente con Apache Hadoop, Apache MapReduce, Apache Spark, HBase e altri strumenti per Big Data

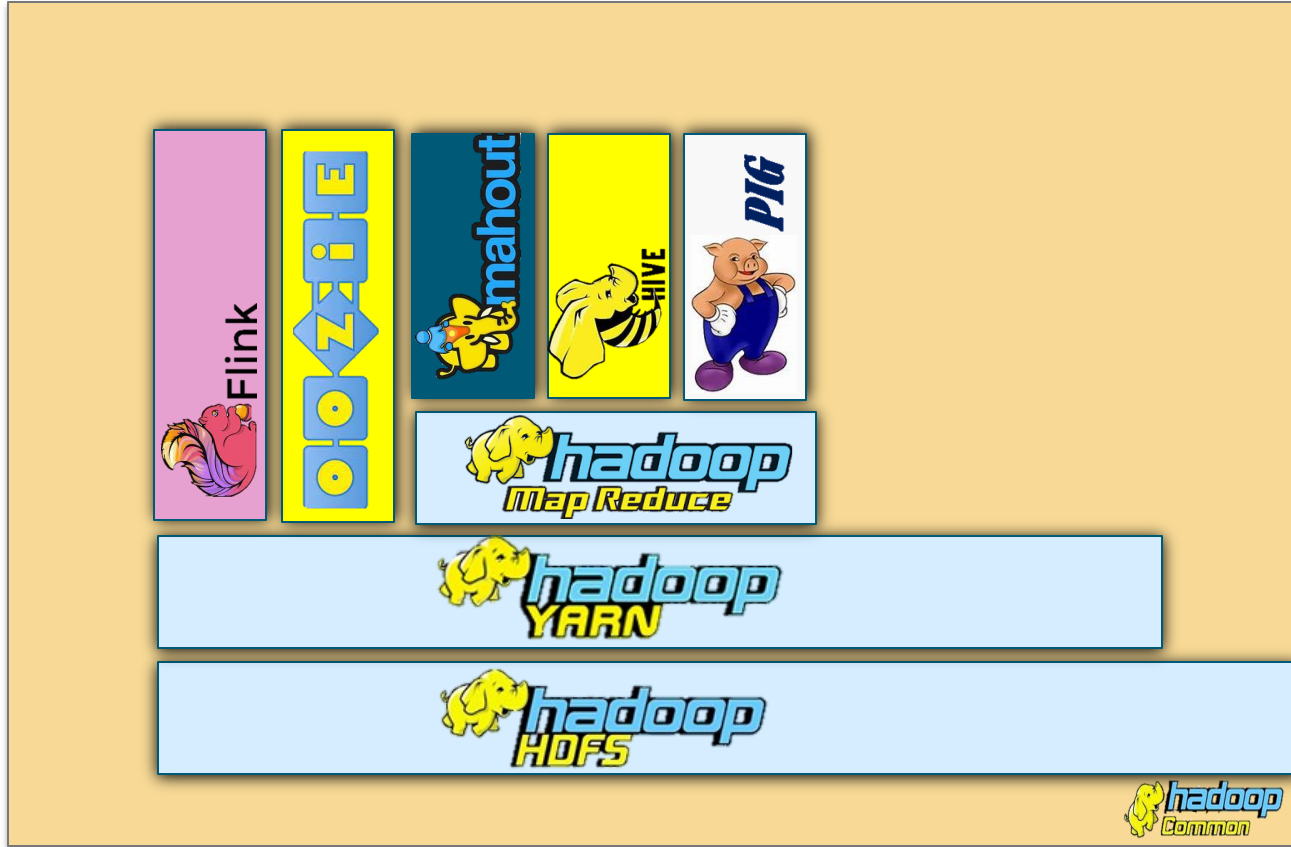
Ecosistema Hadoop – Apache Flink



CURIOSITA'

- Apache Flink è stato utilizzato per elaborazioni, in:
 - applicazioni che elaborano più trilioni di eventi al giorno,
 - applicazioni che mantengono più terabyte di stato
 - applicazioni in esecuzione su migliaia di core.
- Può elaborare i dati a una velocità fulminea
- API disponibili in Java, Scala e Python

Ecosistema Hadoop

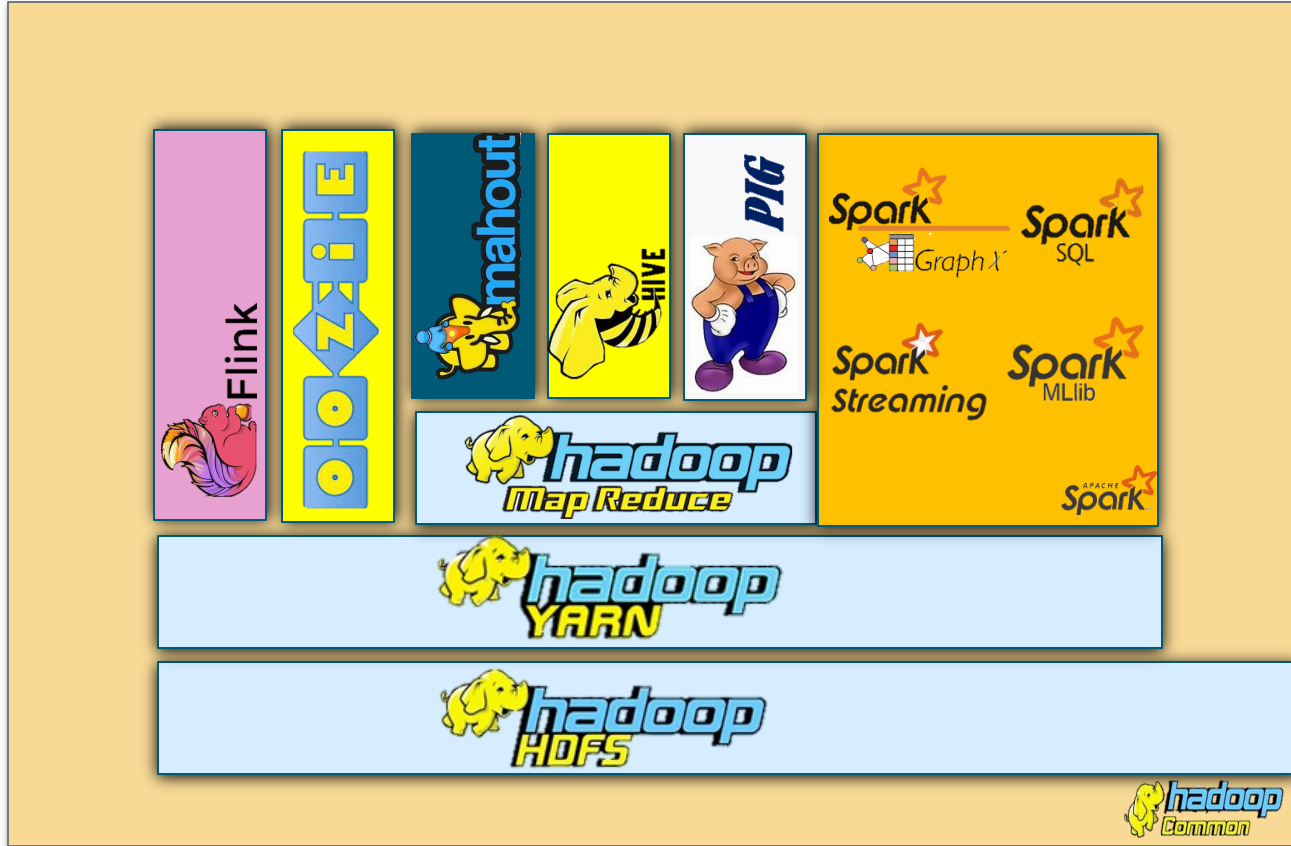


Ecosistema Hadoop – Apache Oozie

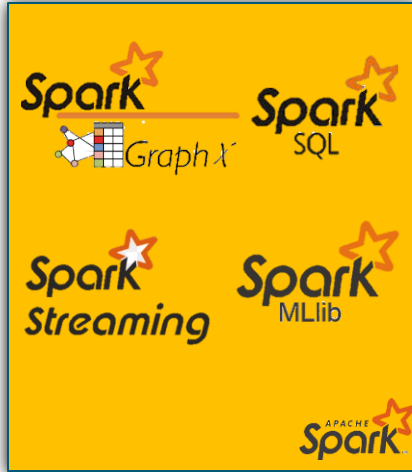


- Oozie è un sistema utilizzato per pianificare le operazioni di Apache Hadoop (schedulazione dei workflow).
- Le operazioni dei flussi di Job Oozie sono grafici aciclici diretti (DAG) che specificano la sequenza di azioni da eseguire.
- Le operazioni Oozie Coordinator sono operazioni dei flussi di job Oozie ricorrenti, attivati dal tempo (frequenza) e dalla disponibilità dei dati.
- Oozie è integrato con il resto dello stack Hadoop supportando diversi tipi di job Hadoop (come operazioni per MapReduce, Pig, Hive e Sqoop) nonché job specifici del sistema (come programmi Java e script di shell).
- Oozie attiva le azioni del flusso di lavoro e Hadoop MapReduce le esegue.
- Oozie Coordinator è in grado di gestire più flussi di job che dipendono dall'esito di flussi di lavoro disposti in sequenza (pipeline di flussi di job)
- Oozie è un sistema scalabile, affidabile ed estensibile.

Ecosistema Hadoop



Ecosistema Hadoop – Apache Spark



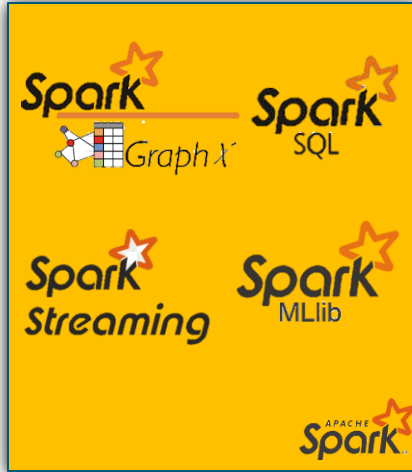
- Apache Spark è una tecnologia di elaborazione in cluster, progettata per un calcolo veloce.
- È basato su Hadoop MapReduce ed estende il modello MapReduce per utilizzarlo in modo efficiente per più tipi di calcoli, che includono query interattive e elaborazione del flusso.

Caratteristiche:

- **in-memory cluster computing.** Aumenta la velocità di elaborazione di un'applicazione memorizzando i dati di elaborazione intermedia in memoria, riducendo gli accessi su disco
- **Supports multiple languages.** Spark fornisce API integrate in Java, Scala o Python.
- **Advanced Analytics.** È studiato appositamente per algoritmi di apprendimento automatico. Permette, oltre a programmazione basata su MapReduce, anche query SQL, dati in streaming, machine learning (ML) e algoritmi di grafici



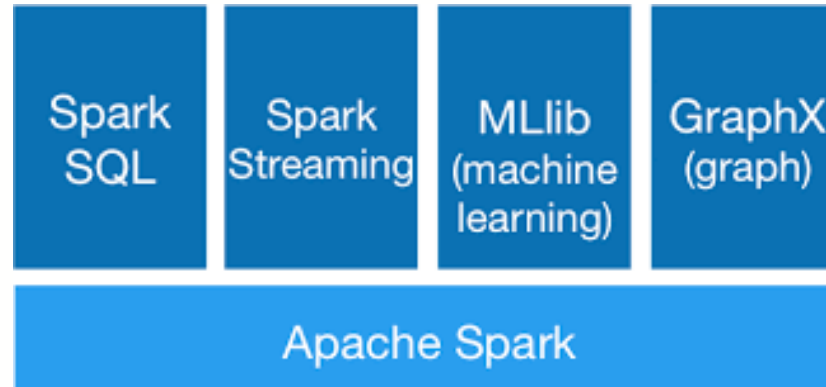
Ecosistema Hadoop – Apache Spark



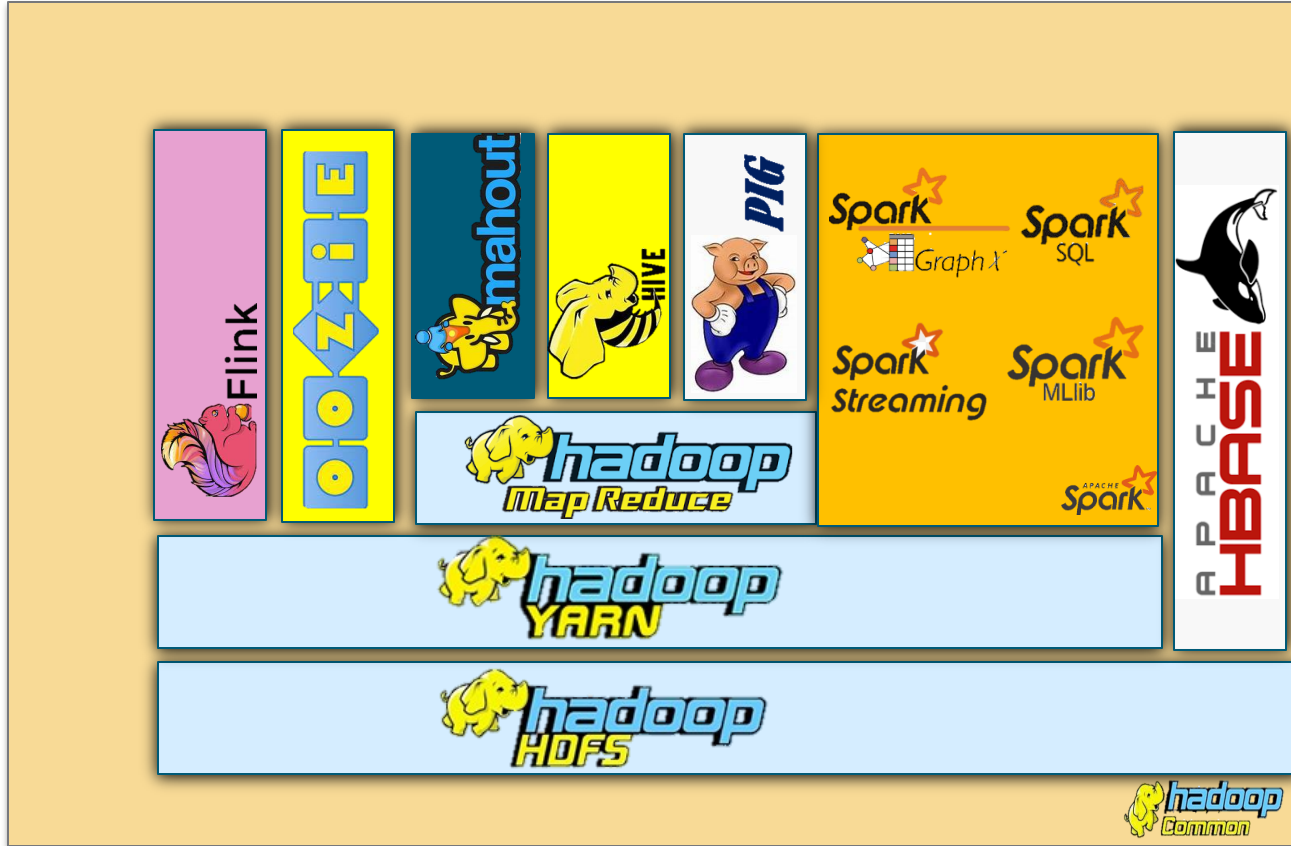
Apache Spark per operare richiede:

- un gestore di cluster – supportato nativamente da Spark o funziona su Hadoop YARN
- un sistema di archiviazione distribuita – si interfaccia con HDFS, Apache Cassandra, Apache Kudu, oltre a soluzioni personalizzabili.

Moduli:



Ecosistema Hadoop

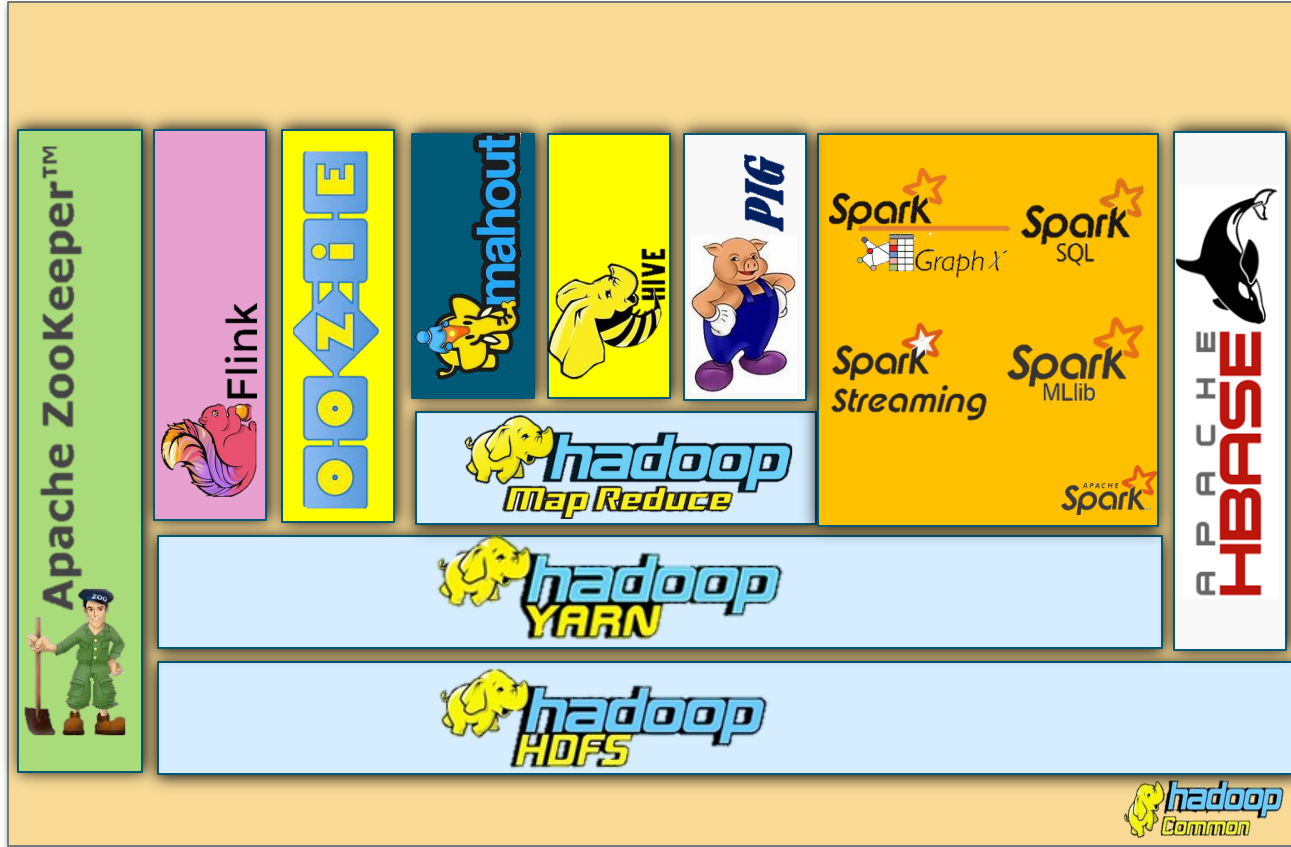


Ecosistema Hadoop – Apache HBase



- **Apache HBase** è un database **NoSQL** Open Source, distribuito, non relazionale, modellato su Bigtable di Google
- È un sistema di archiviazione distribuito per dati strutturati
- Più che un «Data Base» è un **Data Store**, perché manca di molte delle caratteristiche che si trovano in un RDBMS, quali le colonne tipizzate, gli indici secondari, i trigger e i linguaggi di interrogazione avanzati, ecc.
- Le tabelle HBase sono distribuite sul cluster tramite regioni e le regioni vengono automaticamente suddivise e ridistribuite man mano che i dati crescono.
- HBase si integra con HDFS come file system distribuito
- HBase supporta l'elaborazione massicciamente parallelizzata tramite MapReduce

Ecosistema Hadoop



Ecosistema Hadoop – Apache Zookeeper



Apache Zookeeper è un software in grado di fornire supporto per la gestione della sincronizzazione di sistemi distribuiti.

Fornisce:

- servizi operativi per un cluster Hadoop
 - un servizio di **configurazione distribuita**, un servizio di **sincronizzazione** e un **registro dei nomi** per i sistemi distribuiti.
-
- Le applicazioni distribuite utilizzano Zookeeper per archiviare e mediare gli aggiornamenti a importanti informazioni di configurazione
 - È stato progettato per essere facilmente programmabile e utilizza un modello di dati che ricalca la struttura ad albero delle directory dei file system.
 - Zookeeper nasce per sollevare le applicazioni distribuite dalla responsabilità di implementare servizi di coordinamento da zero, che sono notoriamente difficili da realizzare



Ecosistema Hadoop – Apache Zookeeper



Apache Zookeeper consente ai processi distribuiti di coordinarsi tra loro attraverso uno spazio dei nomi gerarchico condiviso.

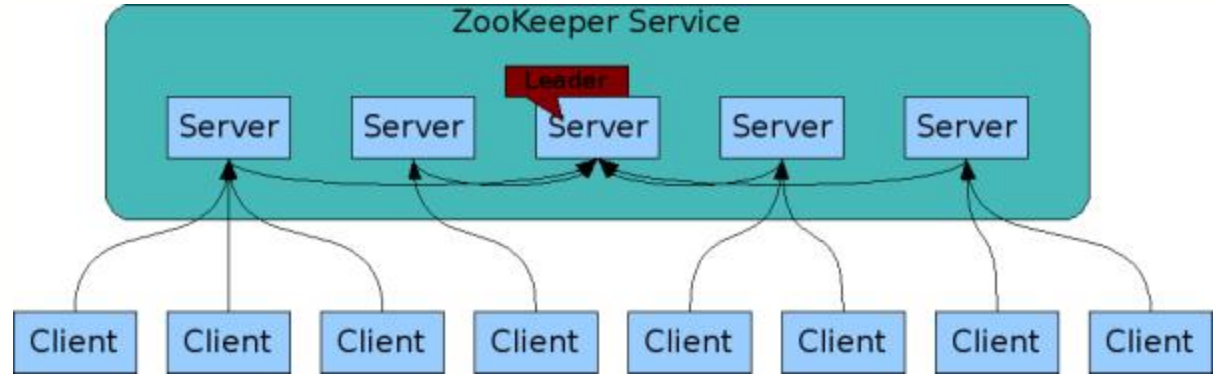
Lo spazio dei nomi consiste in registri di dati, chiamati znodi, che sono simili a file e directory.

A differenza di un tipico file system, che è progettato per l'archiviazione, i dati di ZooKeeper sono conservati in memoria.

- Come i processi distribuiti che coordina, ZooKeeper stesso è destinato a essere replicato su un insieme di host chiamato ensemble.

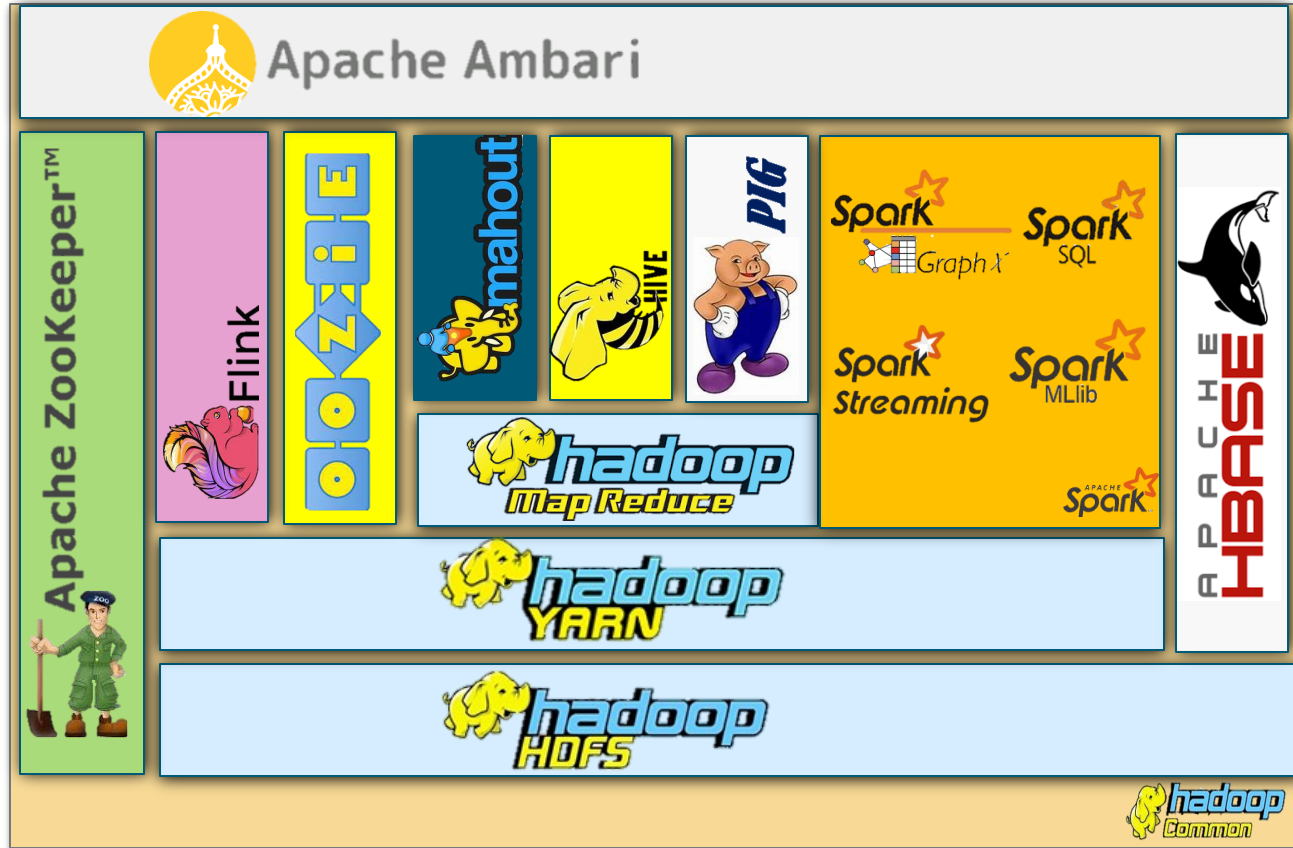


Ecosistema Hadoop – Apache Zookeeper



- I server che compongono il servizio ZooKeeper devono tutti conoscere gli altri.
- Mantengono un'immagine in memoria dello stato, insieme ai registri delle transazioni e alle istantanee in un archivio persistente.
- I client si collegano a un singolo server ZooKeeper.
- Il client mantiene una connessione TCP attraverso la quale invia richieste, riceve risposte, riceve eventi di osservazione e invia heartbeat.
- Se la connessione TCP al server si interrompe, il client si connette a un altro server.

Ecosistema Hadoop



Ecosistema Hadoop – Apache Ambari



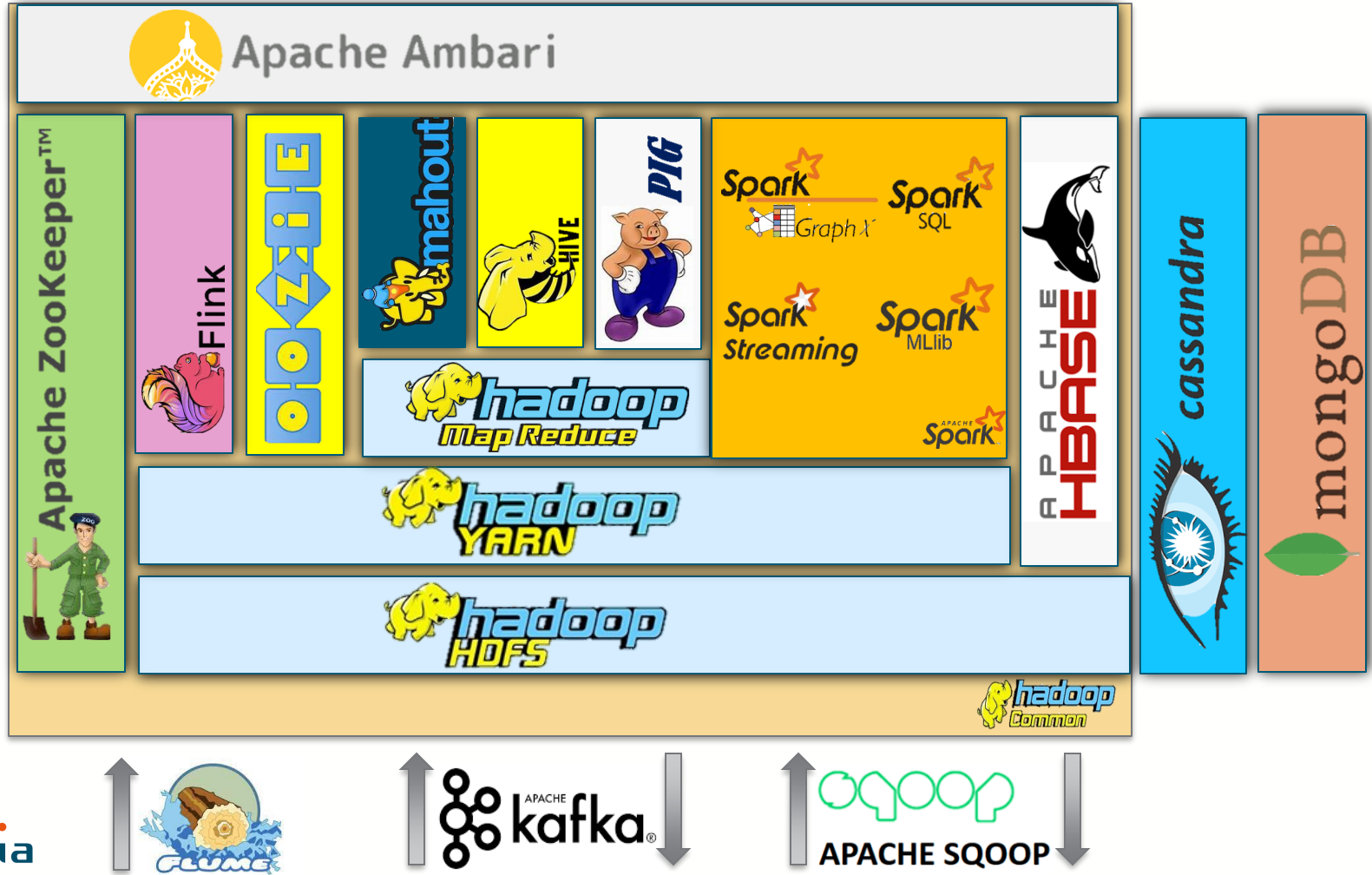
Apache Ambari è una piattaforma completamente Open Source che permettere in modo semplice ed efficiente :

- il **provisioning**,
- la **gestione**,
- il **monitoraggio**
- la **protezione**

di cluster Hadoop su larga scala

- fornisce un'interfaccia utente Web di gestione Hadoop intuitiva e facile da usare
- consente la gestione centrale per l'avvio, l'arresto e la riconfigurazione dei servizi Hadoop nell'intero cluster.
- fornisce una dashboard per il monitoraggio dell'integrità e dello stato del cluster Hadoop
- consente di automatizzare l'installazione e la configurazione delle funzionalità avanzate di sicurezza del cluster come Kerberos e Apache Ranger
- consente di configurare avvisi predefiniti, basati sulle migliori pratiche operative, per il monitoraggio dei cluster

Ecosistema Hadoop



Ecosistema Hadoop – Apache Ambari



Apache Cassandra è un database NoSQL, di tipo colonnare, ottimizzato per la gestione di grandi quantità di dati.



MongoDB è un database NoSQL di tipo documentale



Apache Flume è un servizio distribuito per la raccolta, l'aggregazione e il trasporto di un grande quantitativo di dati di log in maniera efficiente

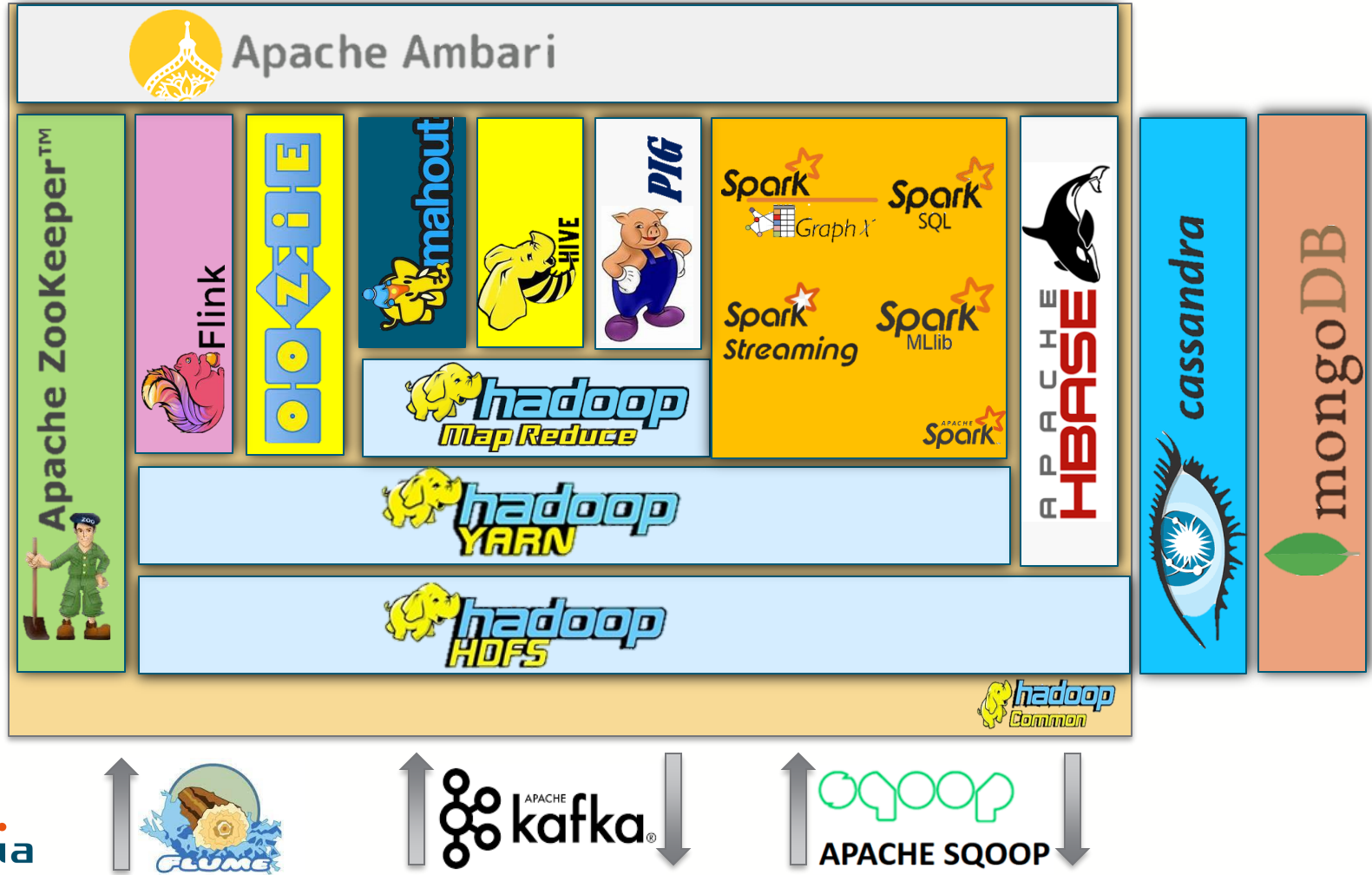


Apache Kafka è una piattaforma per il data streaming distribuita che permette di pubblicare, sottoscrivere, archiviare ed elaborare real time flussi di dati provenienti da più sorgenti distribuendoli a più consumatori.



Apache Sqoop è un'applicazione a riga di comando che permette di trasferire dati da basi di dati relazionali a Hadoop.

Ecosistema Hadoop





Grazie

www.exprivia.it

Diritti d'autore e copyright

Questo documento è proprietà esclusiva della società Exprivia S.p.A e non può essere riprodotto, anche in forma parziale, senza un'autorizzazione scritta della società stessa.