polo innovazione
ICT Calabria

ICAR CNR
Sede di Cosenza
Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

UNIVERSITÀ DELLA CALABRIA
DIMES
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Seminario

Strumenti Open Source utilizzabili nel contesto Big Data

31 Marzo 2015

Ing. Cassavia Nunziato (ICAR-CNR)
Ing. Dicosta Pietro (UNICAL)

Sommario

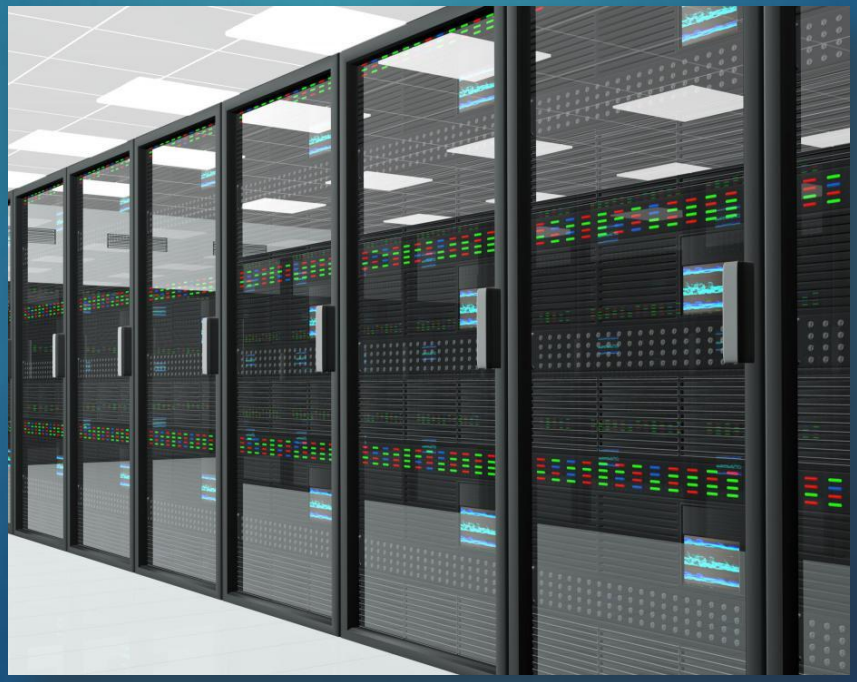
- Gli strumenti Open Source
  - HDFS
  - Hadoop
  - Hive
  - HBase
  - Solr (Carrot)
  - Indexing (Batch ed NRT)
- Possibili approcci ed architetture
  - Analisi Batch
  - Ricerca Full Text
- Cloudera (con breve DEMO)
- Conclusioni

HDFS

Introduzione

- **HDFS (Hadoop Distributed File System)** è un file system distribuito progettato per eseguire su commodity hardware

- HDFS fornisce alto throughput per le applicazioni che utilizzano enormi dataset

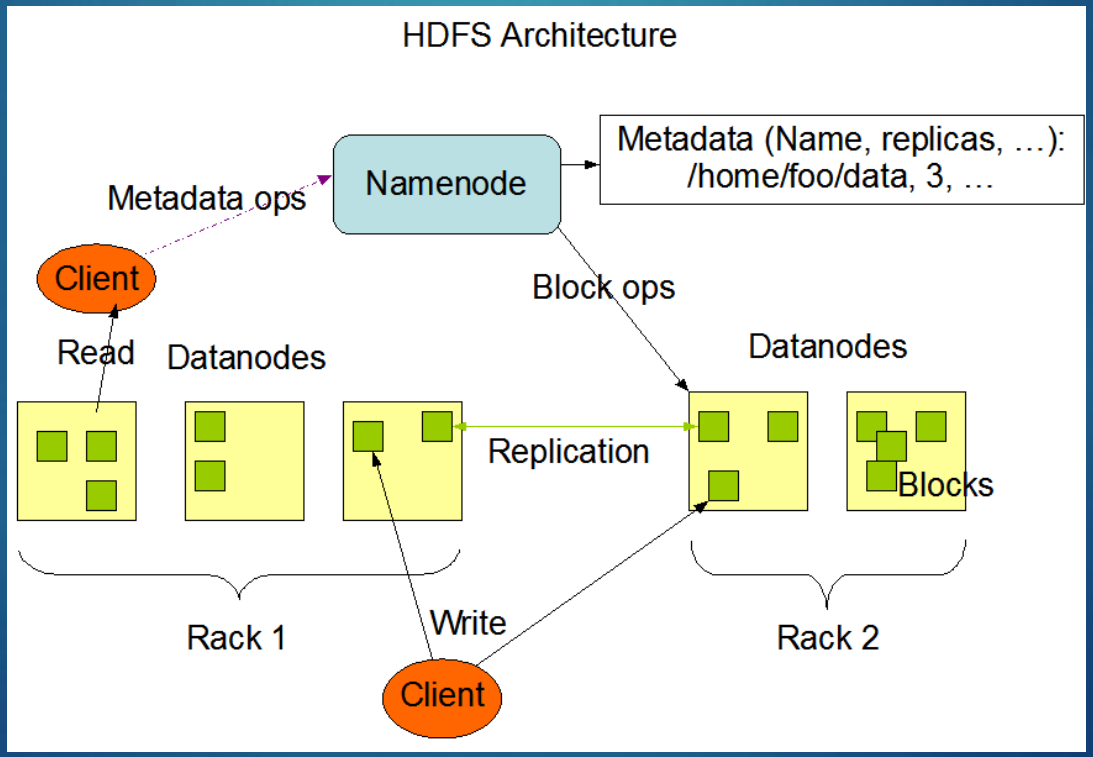


Presupposti e scopo

- **Fallimenti Hardware**
  - Il fallimento dei nodi hardware è la norma come l'eccezione
- **Enormi Dataset**
- **Accesso Streaming ai dati**
  - Alto throughput preferito a bassa latenza
- **Modello Simple Coherency**
  - Modello di accesso ai file "write-once-read-many"
- **Muovere la computazione è più conveniente che muovere i dati**
- **Portabilità fra piattaforme hardware e software eterogenee**

Filesystem Namespace e Architettura

- **Namespace**
  - Organizzazione dei files gerarchica tradizionale
- **Architettura Master-Slave**
  - NameNode & DataNode



Accessbilità

- **Java API**
- **C wrapper API**
- **Fs shell**
  - HDFS fornisce una interfaccia da linea di comando chiamata FS shell
  - La sintassi di questi comandi è molto simile a quella dell altre shell POSIX

Azione	Comando
Creazione di una directory chiamata foodir	`> bin/hdfs dfs -mkdir /foodir`
Rimozione della directory foodir	`> bin/hdfs dfs -rmr /foodir`
Visualizzazione del contento del file /foodir/Myfile.txt	`> bin/hdfs dfs -cat /foodir/myfile.txt`

Map-Reduce

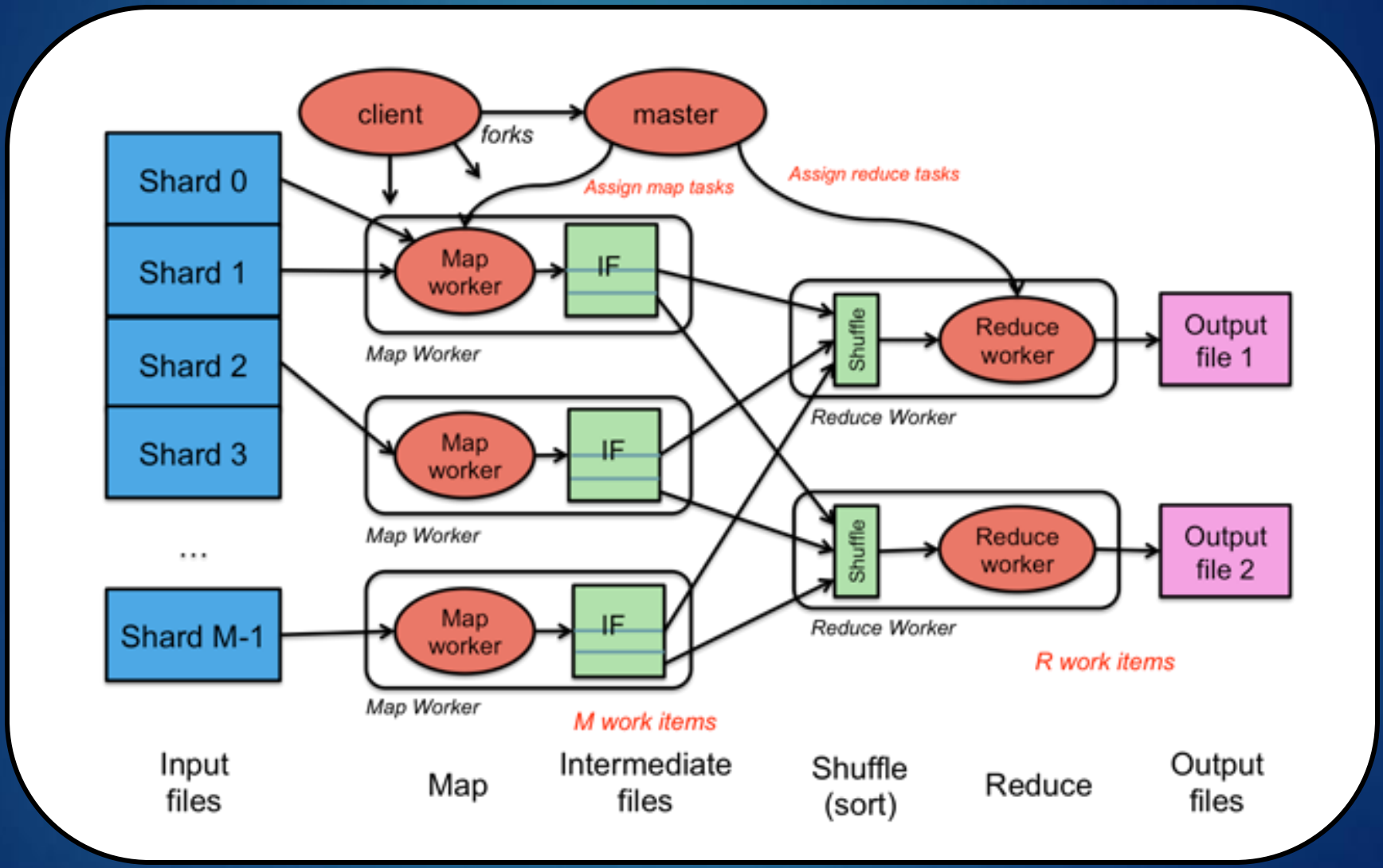
Introduzione

- **Map-Reduce** è un modello di programmazione, proposto da Google nel 2006, utile a processare e/o generare grandi data sets

- Gli utenti devono solo specificare una funzione "map" (che processa coppie chiave-valore e ne genera di altre intermedie), ed una funzione "reduce" (che fa il merge di tutti quei valori intermedi associati alla stessa chiave)

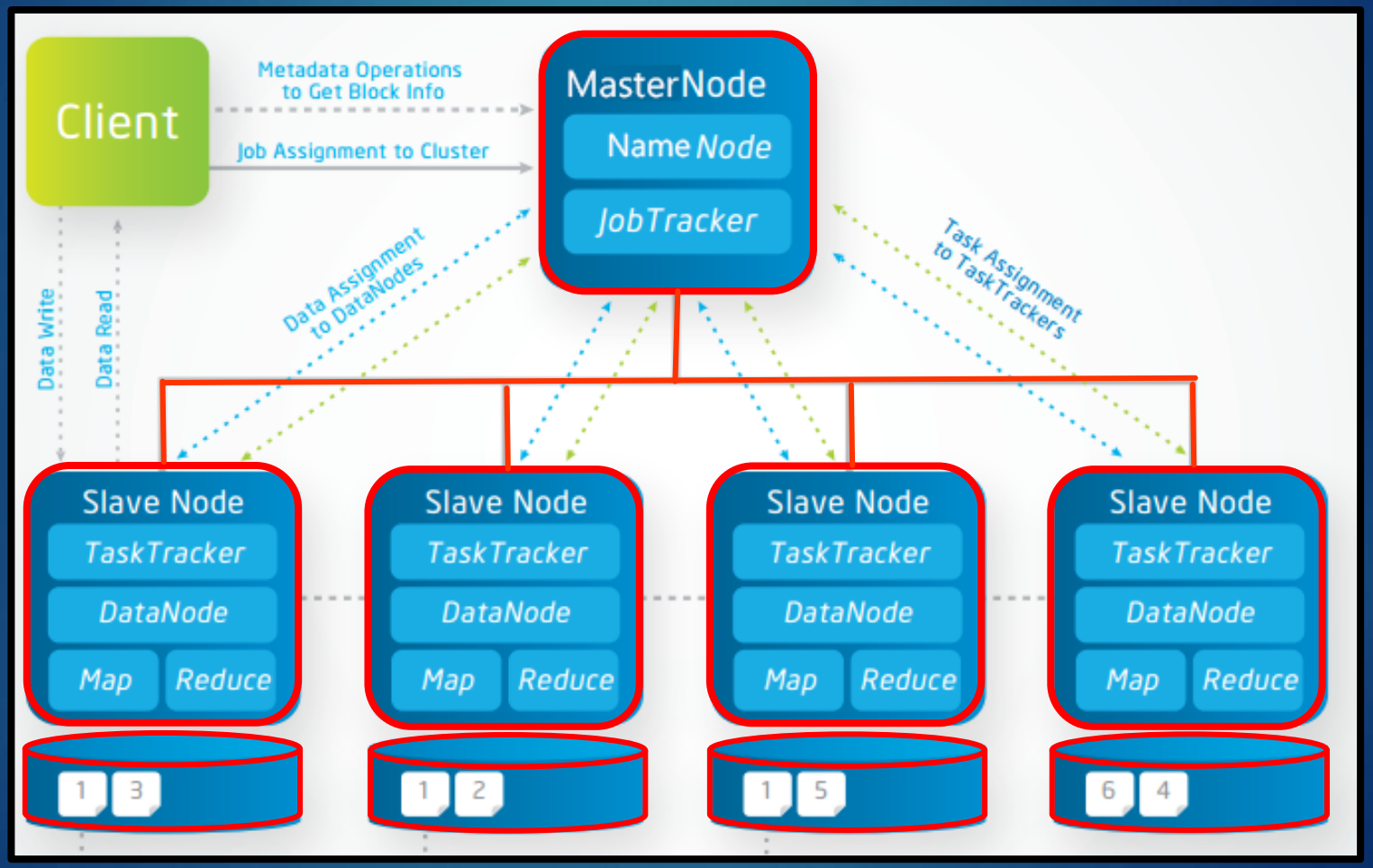
- Rappresenta una soluzione per molti problemi reali e inoltre i programmi scritti secondo questo modello, risultano automaticamente eseguibili in parallelo (anche su clusters di commodity hardware)

MapReduce - Funzionamento

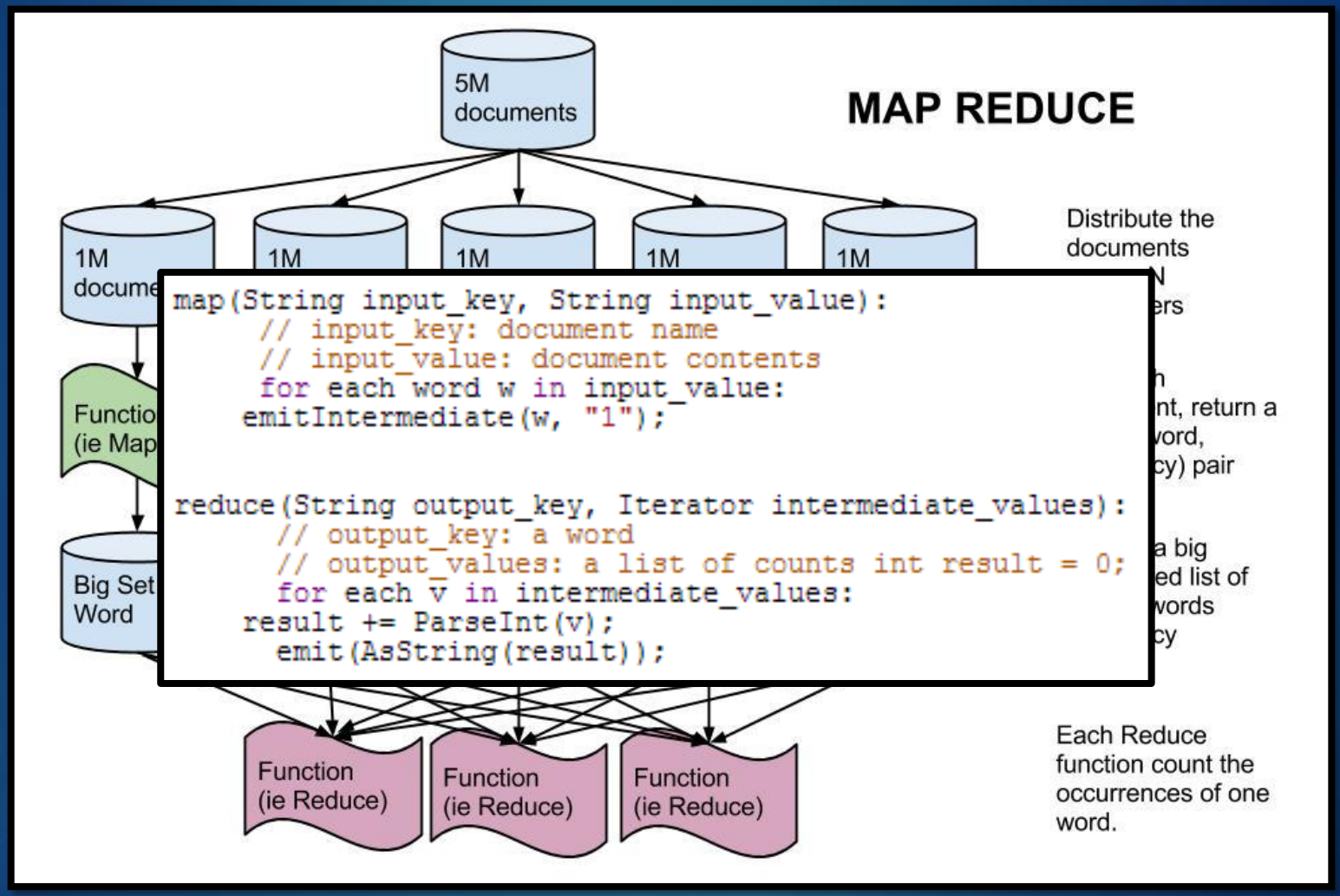


Apache Hadoop

Una implentazione **Open Source del** framework MapReduce



MapReduce – Esempio WordCount

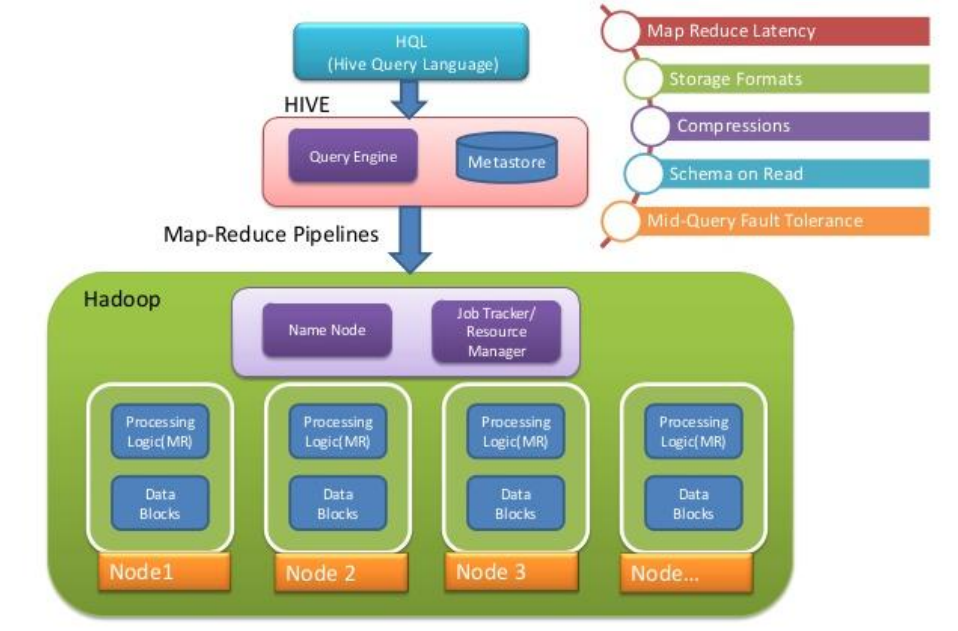


Hive

Introduzione

- **Apache Hive** è un infrastruttura per Data Warehousing creata su Apache Hadoop per fornire dati aggregati, query e analisi

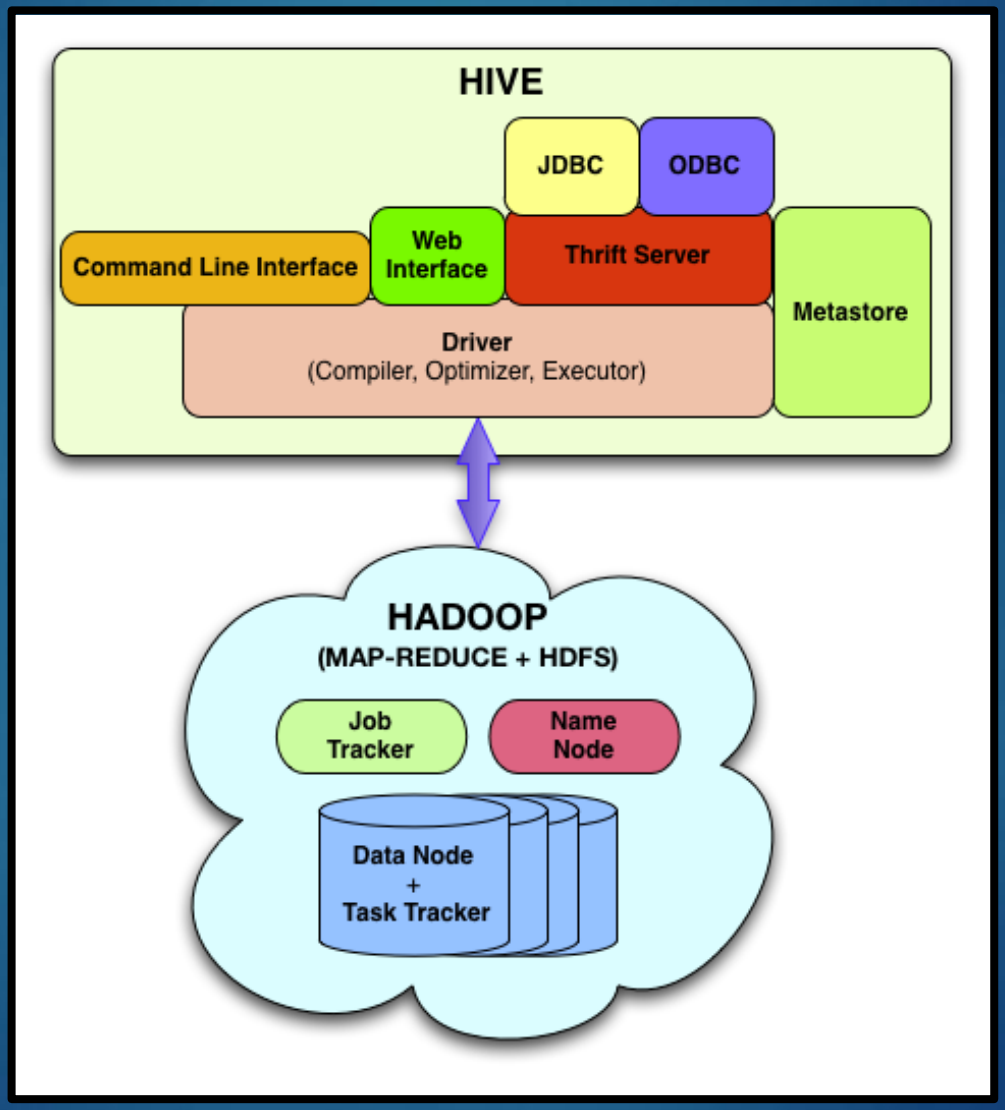
- Hive fornisce un meccanismo per creare delle strutture sui dati ed effettuare delle query utilizzando un linguaggio SQL-like chiamato HiveQL
- Trasparentemente all'utilizzatore converte le query HiveQL in job map-reduce



Caratteristiche

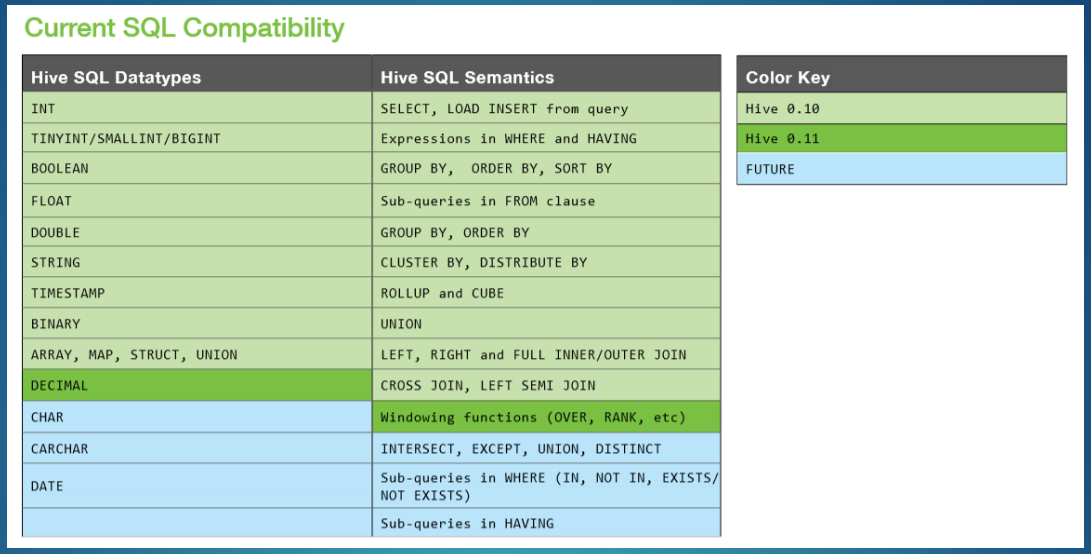
- **Tipi di storage**
  - **Plain Text on HDFS**
  - **RCFile**
  - **HBase**
  - **ORC**
- **Supporto agli indici**
- **Metadati memorizzati su RDMS riducono il tempo di esecuzione di check semantici**
- **Operazioni sui dati compressi tramite algoritmi DEFLATE, bwt, snappy**

Architettura



HiveQL

- SQL



Current SQL Compatibility

Hive SQL Datatypes	Hive SQL Semantics
INT	SELECT, LOAD INSERT from query
TINYINT/SMALLINT/BIGINT	Expressions in WHERE and HAVING
BOOLEAN	GROUP BY, ORDER BY, SORT BY
FLOAT	Sub-queries in FROM clause
DOUBLE	GROUP BY, ORDER BY
STRING	CLUSTER BY, DISTRIBUTE BY
TIMESTAMP	ROLLUP and CUBE
BINARY	UNION
ARRAY, MAP, STRUCT, UNION	LEFT, RIGHT and FULL INNER/OUTER JOIN
DECIMAL	CROSS JOIN, LEFT SEMI JOIN
CHAR	Windowing functions (OVER, RANK, etc)
CARCHAR	INTERSECT, EXCEPT, UNION, DISTINCT
DATE	Sub-queries in WHERE (IN, NOT IN, EXISTS/ NOT EXISTS)
	Sub-queries in HAVING

Color Key
Hive 0.10
Hive 0.11
FUTURE

- DML
  - LOAD
  - INSERT into Hive tables from queries
- DDL
  - CREATE

Hive Caso d'Uso: Facebook

facebook

Hive & Hadoop Usage @ Facebook

- Types of Applications:
  - Reporting
    - Eg: Daily/Weekly aggregations of impression/click counts
    - Measures of user engagement
    - Microstrategy dashboards
  - Ad hoc Analysis
    - Eg: how many group admins broken down by state/country
  - Machine Learning (Assembling training data)
    - Ad Optimization
    - Eg: User Engagement as a function of user attributes
  - Many others

Hive Caso d'Uso: Facebook

facebook

Hadoop & Hive Cluster @ Facebook

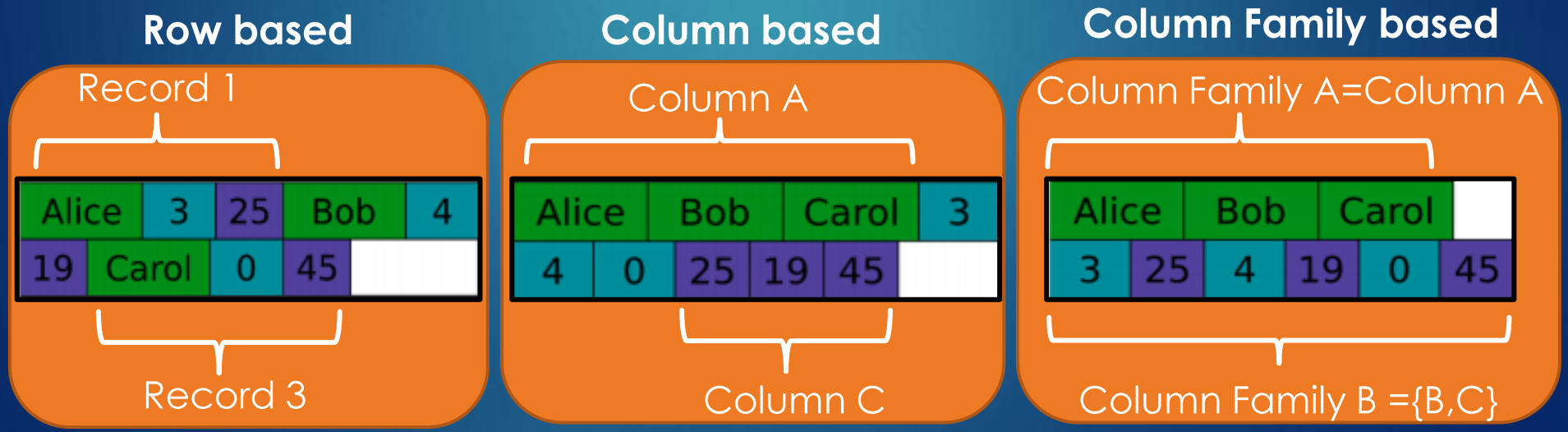
- **Production Cluster**
  - 300 nodes/2400 cores
  - 3PB of raw storage
- **Adhoc Cluster**
  - 1200 nodes/9600 cores
  - 12PB of raw storage
- **Node (DataNode + TaskTracker) configuration**
  - 2CPU, 4 core per cpu
  - 12 x 1TB disk (900GB usable per disk)

HBase

Column Family Stores

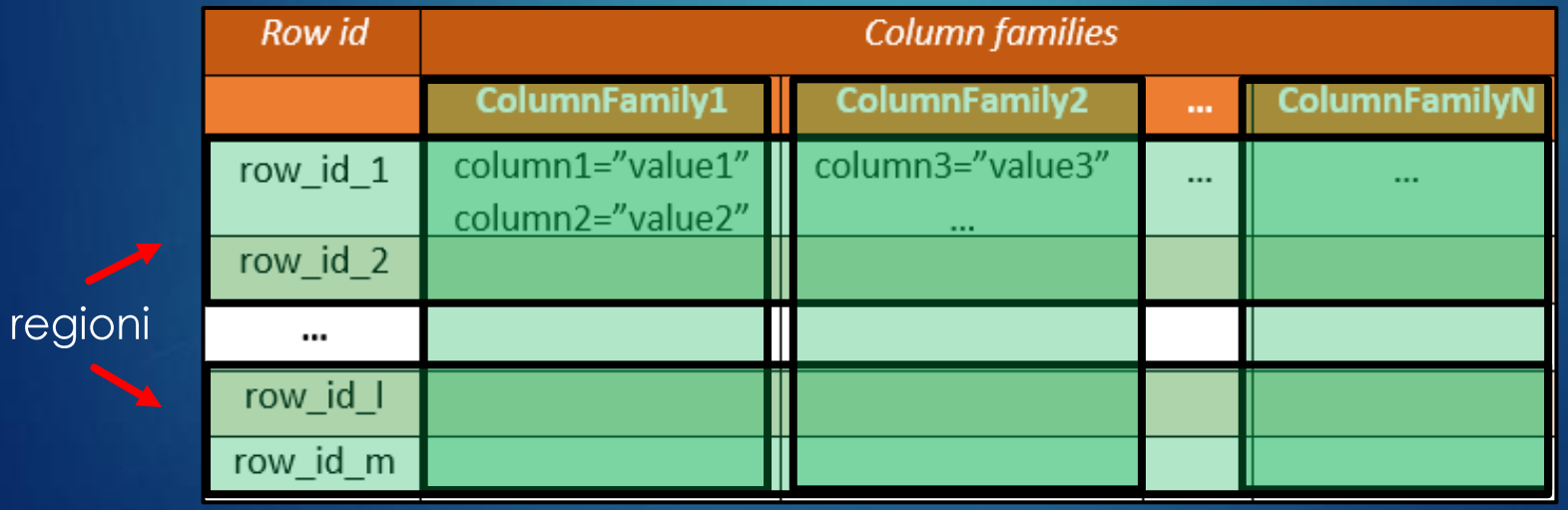
Permettono di memorizzare i dati su un cluster, partizionandoli sia orizzontalmente che verticamente. Il partizionamento orizzontale (sharding) delle tabelle, avviene in base alla chiave, mentre le colonne sono distribuite sulla base di un predefinito raggruppamento

- Essenzialmente sono mappe multi-livello
- Per accedere ad una cella: get('row', 'column', 'timestamp')
- I valori di una singola colonna sono memorizzati in maniera contigua



Apache HBase

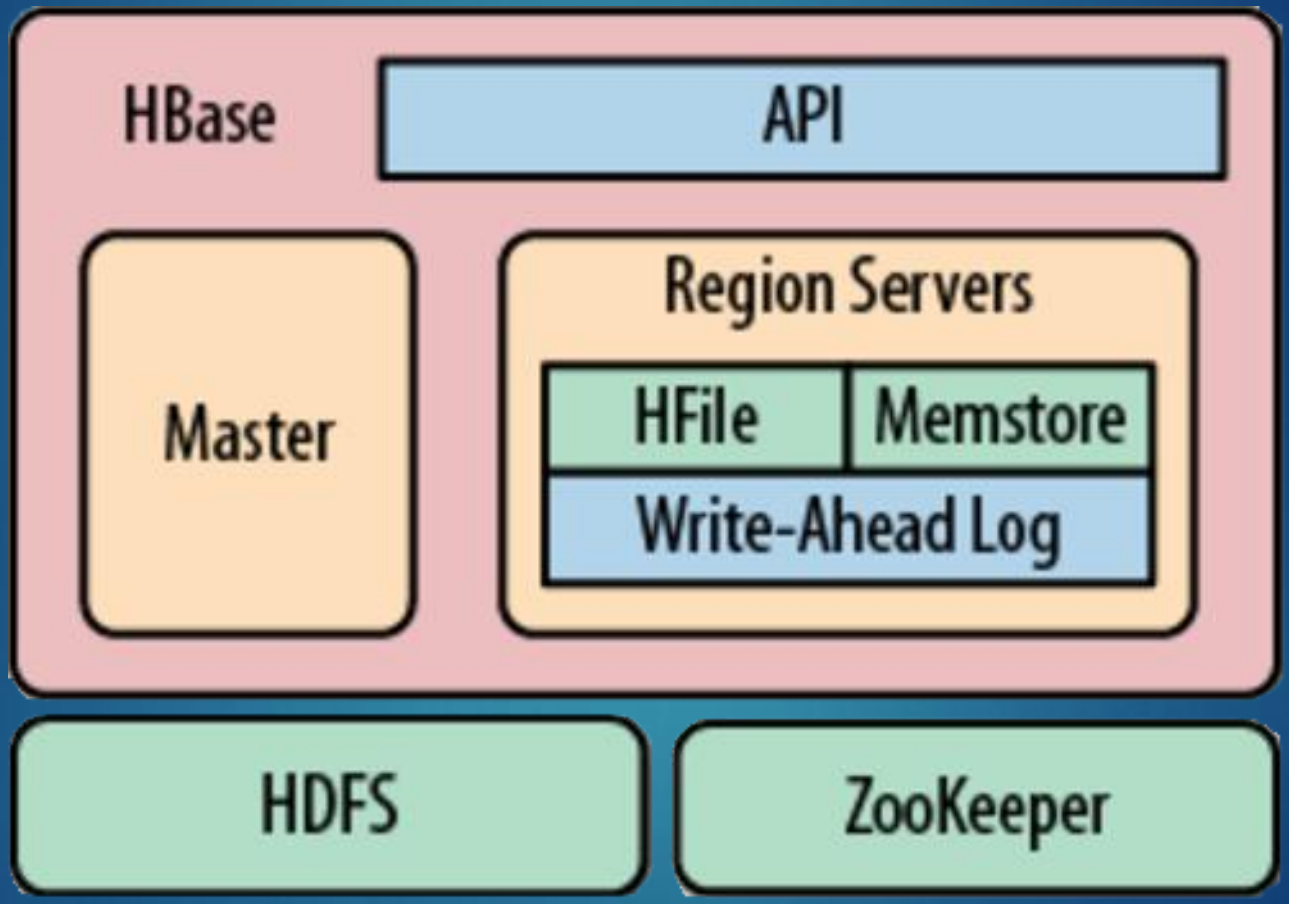
- Basato su Big Table (soluzione proposta da Google)
- Le colonne di una tabella sono raggruppate in Column Families (CFs)
- Le CFs sono statiche e definite in anticipo
- Entro la stessa CF, differenti righe possono avere differenti colonne
- Esiste un solo tipo di dato: il Byte-array



Operazioni:

- get
- put
- delete
- scan

L'architettura di HBase



Gestione di HBase

- HBase Shell
- API Java Native (soluzione più veloce)
- REST Server (poco efficiente)
- Thrift Server (compatile con diversi linguaggi)

Esiste un alto livello di integrazione con il Map Reduce

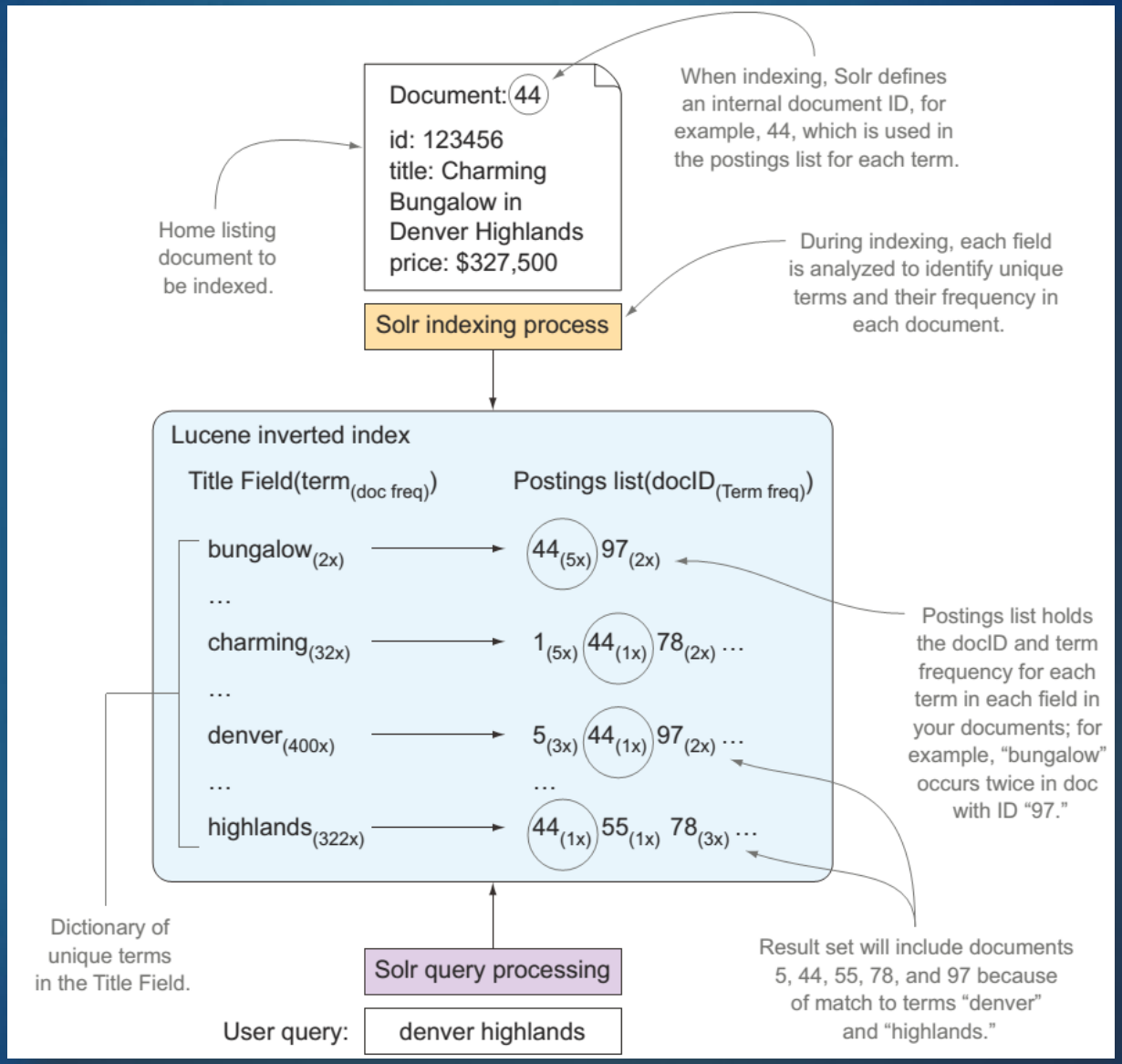
Permette di memorizzare miliardi di tuple, tuttavia rispetto ai classici RDBMS non è adattabile ad ogni situazione, per via delle API limitate

Solr

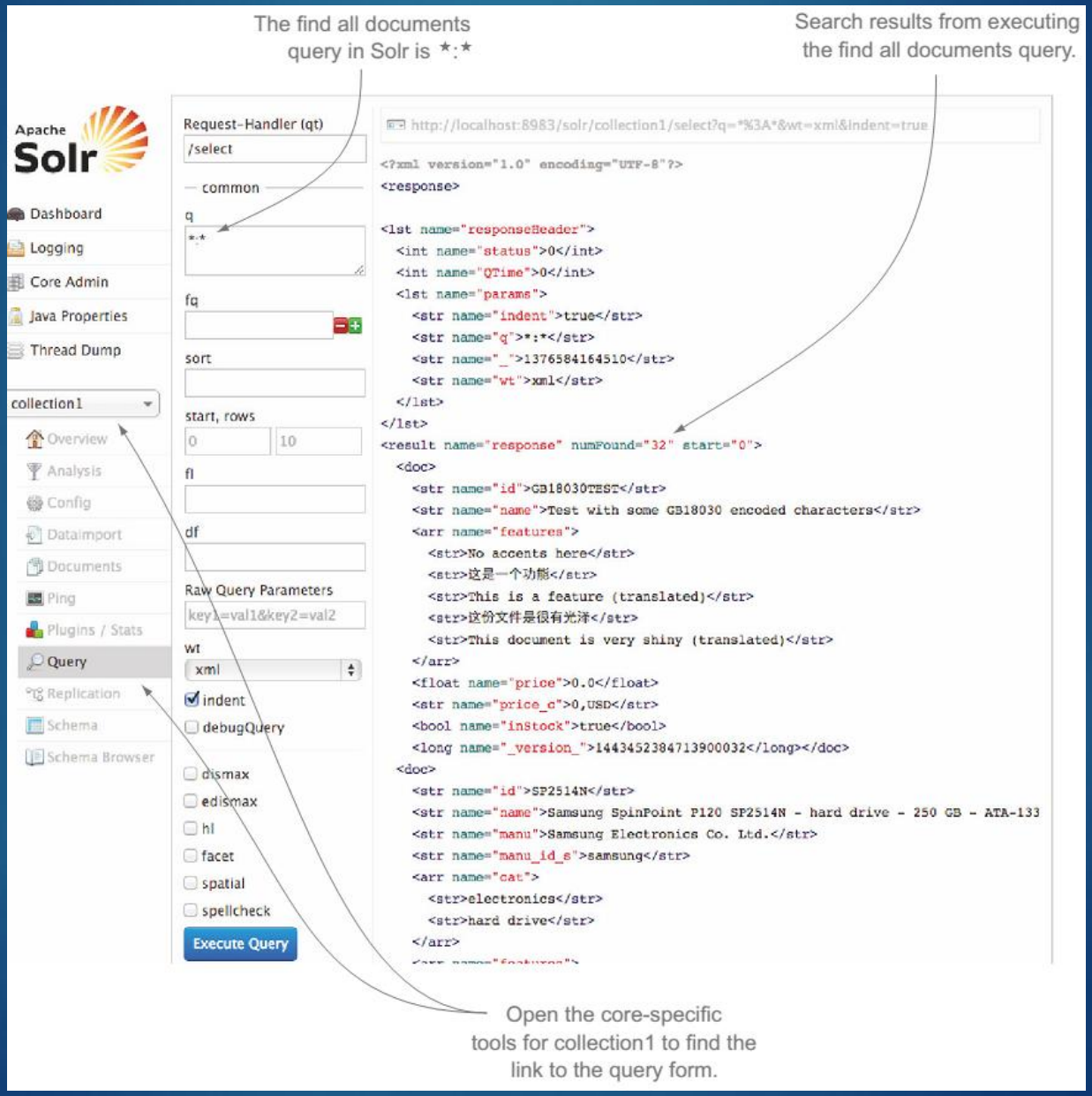
Introduzione

- **Apache Solr** è un motore di ricerca standalone scritto in java che utilizza il progetto Apache Lucene come backend.
- http://lucene.apache.org/solr/
- Solr è realizzato sotto forma di servlet e per funzionare necessita di un server per applicazioni web come Tomcat o Jetty
- Utilizzato pesantemente da: LinkedIN, Twitter, Netflix, Cnet, Digg


Indice Invertito



Solr Admin


The find all documents query in Solr is *:*
Search results from executing the find all documents query.
Apache Solr
Dashboard
Logging
Core Admin
Java Properties
Thread Dump
collection1
Overview
Analysis
Config
Dataimport
Documents
Ping
Plugins / Stats
Query
Replication
Schema
Schema Browser
Request-Handler (qt)
/select
common
q
:
fq
sort
start, rows
0
10
fl
df
Raw Query Parameters
key1=val1&key2=val2
wt
xml
indent
debugQuery
dismax
edismax
hl
facet
spatial
spellcheck
Execute Query
http://localhost:8983/solr/collection1/select?q=*%3A*&wt=xml&indent=true
<?xml version="1.0" encoding="UTF-8"?>
<response>
<lst name="responseHeader">
<int name="status">0</int>
<int name="QTime">0</int>
<lst name="params">
<str name="indent">true</str>
<str name="q">*:*</str>
<str name="_">1376584164510</str>
<str name="wt">xml</str>
</lst>
</lst>
<result name="response" numFound="32" start="0">
<doc>
<str name="id">GB18030TEST</str>
<str name="name">Test with some GB18030 encoded characters</str>
<arr name="features">
<str>No accents here</str>
<str>这是一个功能</str>
<str>This is a feature (translated)</str>
<str>这份文件是很有光泽</str>
<str>This document is very shiny (translated)</str>
</arr>
<float name="price">0.0</float>
<str name="price_c">0,USD</str>
<bool name="inStock">true</bool>
<long name="_version_">1443452384713900032</long></doc>
<doc>
<str name="id">SP2514N</str>
<str name="name">Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133
<str name="manu">Samsung Electronics Co. Ltd.</str>
<str name="manu_id_s">samsung</str>
<arr name="cat">
<str>electronics</str>
<str>hard drive</str>
</arr>
Open the core-specific tools for collection1 to find the link to the query form.

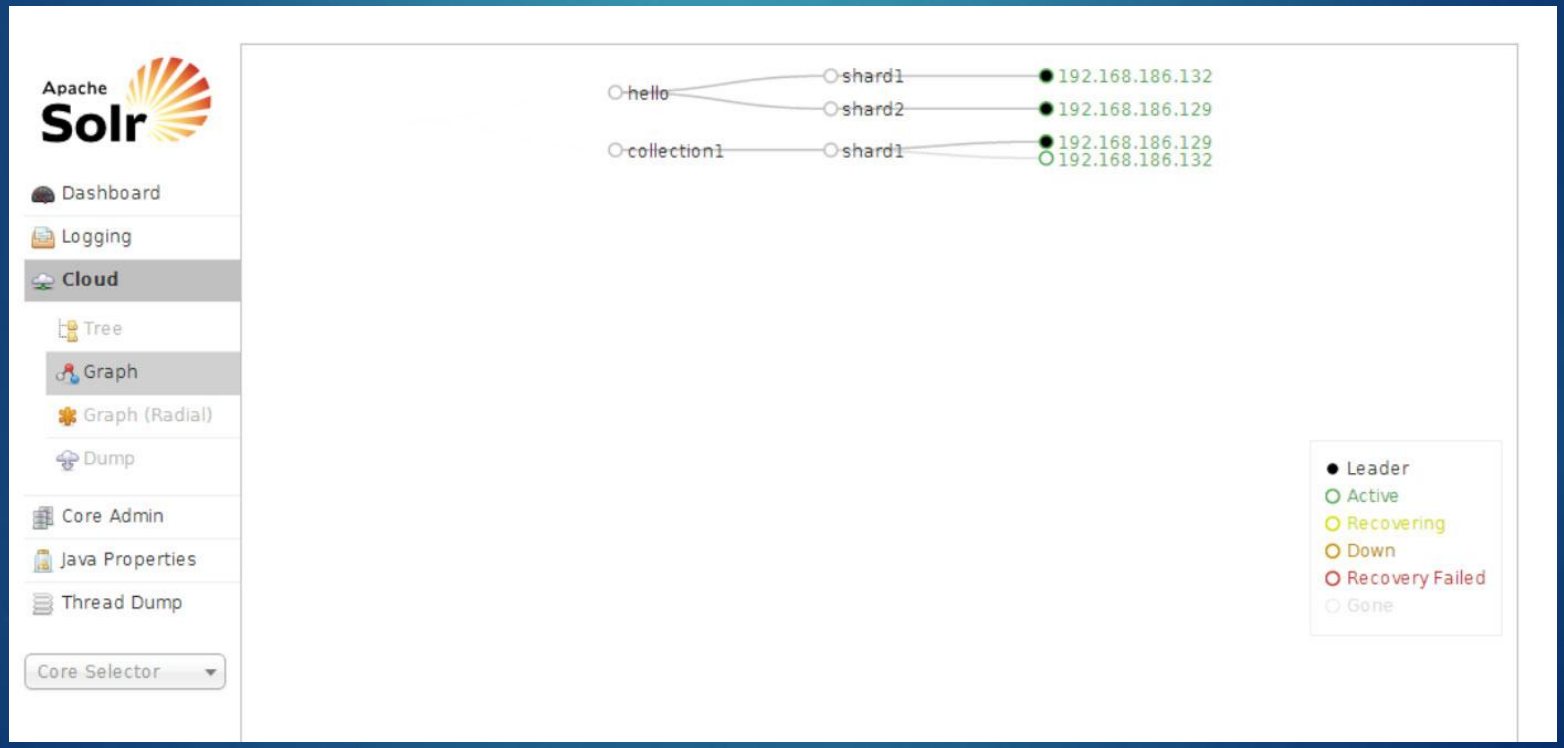
Indicizzazione e ricerca

- **Indicizzazione dei documenti**
  - HTTP post di documenti XML
  - DataImportHandler per basi di dati
  - SolrCell per documenti complessi (PDF o Office)

- Ricerca
  - Tramite URL:
  - Es. *http://localhost:8983/solr/select?indent=on&version=2.2&q=*%3A*&start=0&rows=10&fl=*%2Cscore&qt=standard&wt=standard*

SolrCloud

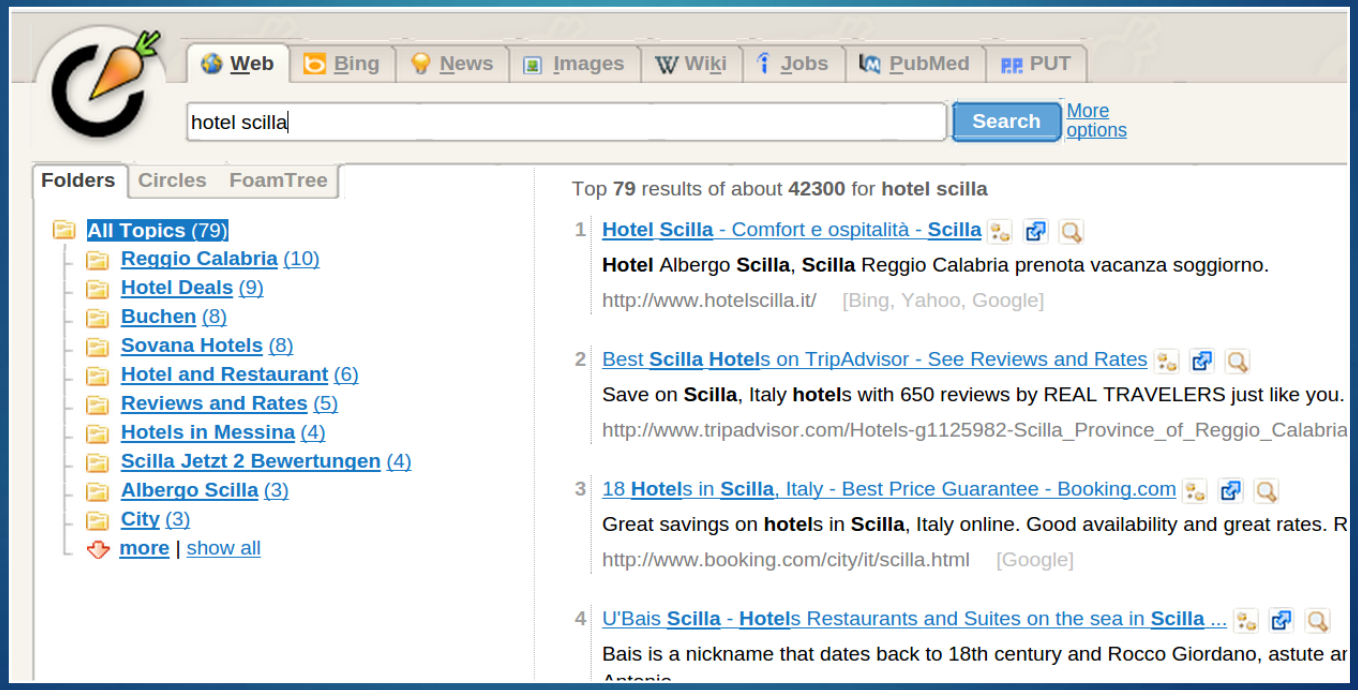
- Memorizzazione indice su HDFS
- Segmentazione e replicazione core su più nodi
- Gestione automatica sharding



Carrot2

Introduzione

- **Carrot2** è una libreria con la quale è possibile clusterizzare risultati provenietnti da motori di ricerca
- Nessuna tassonomia o contenuti preclassificati
- Possibilità di interfacciamento con Solr o motori di ricerca già esistenti come Google o Bing



Algoritmi di Clustering

- **Algoritmi integrati in Carrot2**
  - Lingo
  - STC
  - K-Means

Caratteristica	Lingo	STC	k-means
Diversità dei cluster	Alta. Vengono evidenziati molti piccoli clusters	Bassa. Pochi cluster evidenziati	Bassa. Pochi cluster evidenziati
Etichette dei Clusters	Lunghe, dunque molto descrittive.	Corte, ma ancora appropriate	Una sola parola ma non sempre descrittiva di tutti i documenti nel cluster.
Scalabilità	Bassa. Per un numero superiore a 1000 documenti, l'algoritmo Lingo potrebbe impiegare molto tempo e molta memoria.	Alta	Bassa, basata su strutture dati simili a Lingo.

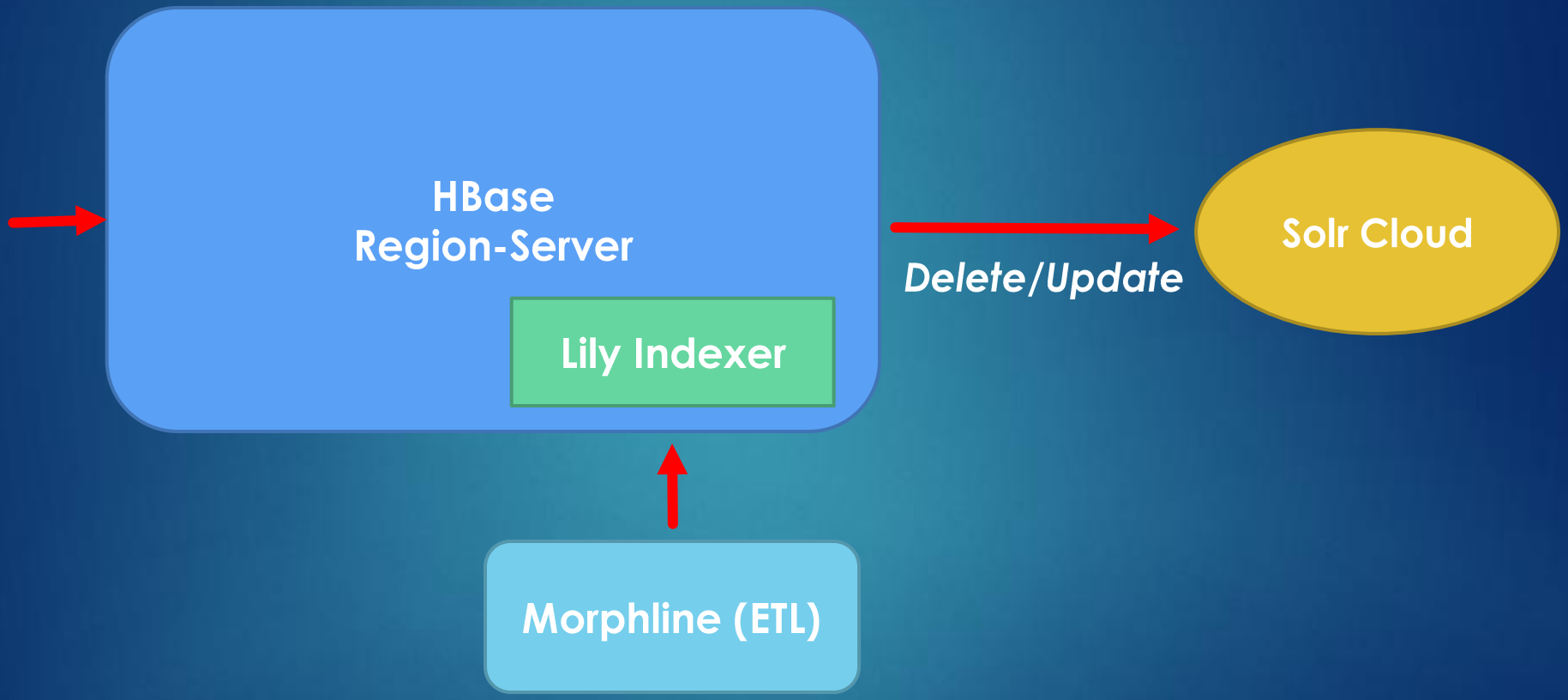
NG DATA

Indicizzazione Lily / Map-Reduce

Introduzione

- **Lily** è un sistema di indicizzazione NRT (Near Real Time) per HBase
- Rappresenta una valida soluzione per ottenere una ricerca Full-Text efficiente dei dati memorizzati in HBase
- Simula il comporatmento di un Hbase Region Server, intercettando le richieste di scrittura e cancellazione, delegandole a Solr Cloud (dopo aver applicato un ETL "al volo")

L’architettura di Lily



Un esempio di morphline per l'ETL

```
        {
           inputColumn : "cfAnagrafica:sorgenti"
           outputField : "sorgenti"
           type : "string"
           source : value
        }
        {
           inputColumn : "cfAttributi:servizio_*"
           outputField : "servizi"
           type : "string"
           source : value
        }
      ]
    }
  }

  {
    addValues { location : "@{latitudine},@{longitudine}"}
  }
```

Indicizzazione con Map Reduce

Nei casi in cui è necessario indicizzare dati in maniera statica (per esempio caricamento iniziale del DW) è conveniente utilizzare il Map Reduce

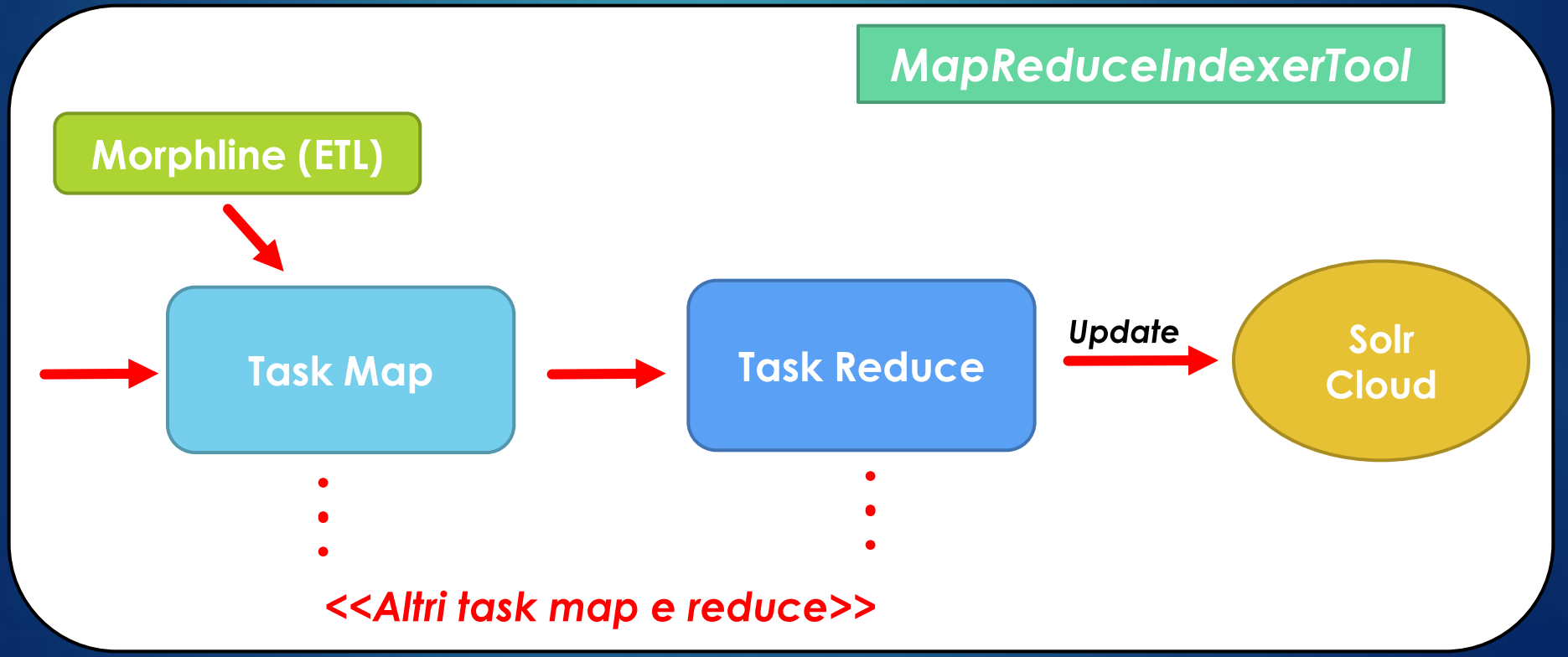
In questo caso:

- la Map produce una coppia termine univoco e ID del documento in cui il termine è stato trovato
- mentre nella fase di Reduce ci si ritrova per ogni termine la lista dei documenti in cui esso occore

Perciò la Reduce non dovrà fare altro che contare per ogni termine le relative frequenze nei documenti e costruire "l'indice invertito"

Architettura di Map Reduce Indexing

Come per Lily viene utilizzato un morphline per eseguire dell'ETL sui dati prima del loro caricamento su Solr

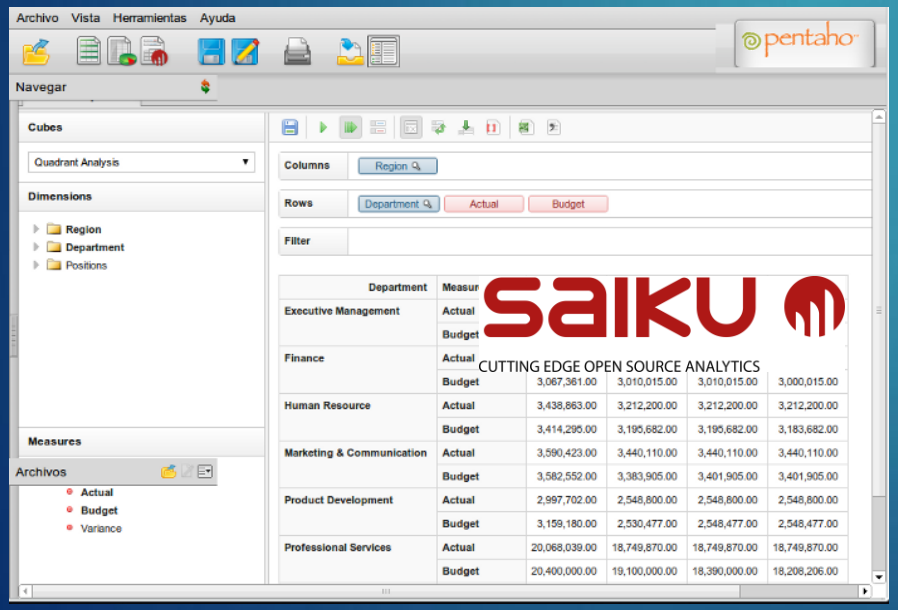


Architettura per Analisi Batch

Data Analysis Tools

Mondrian

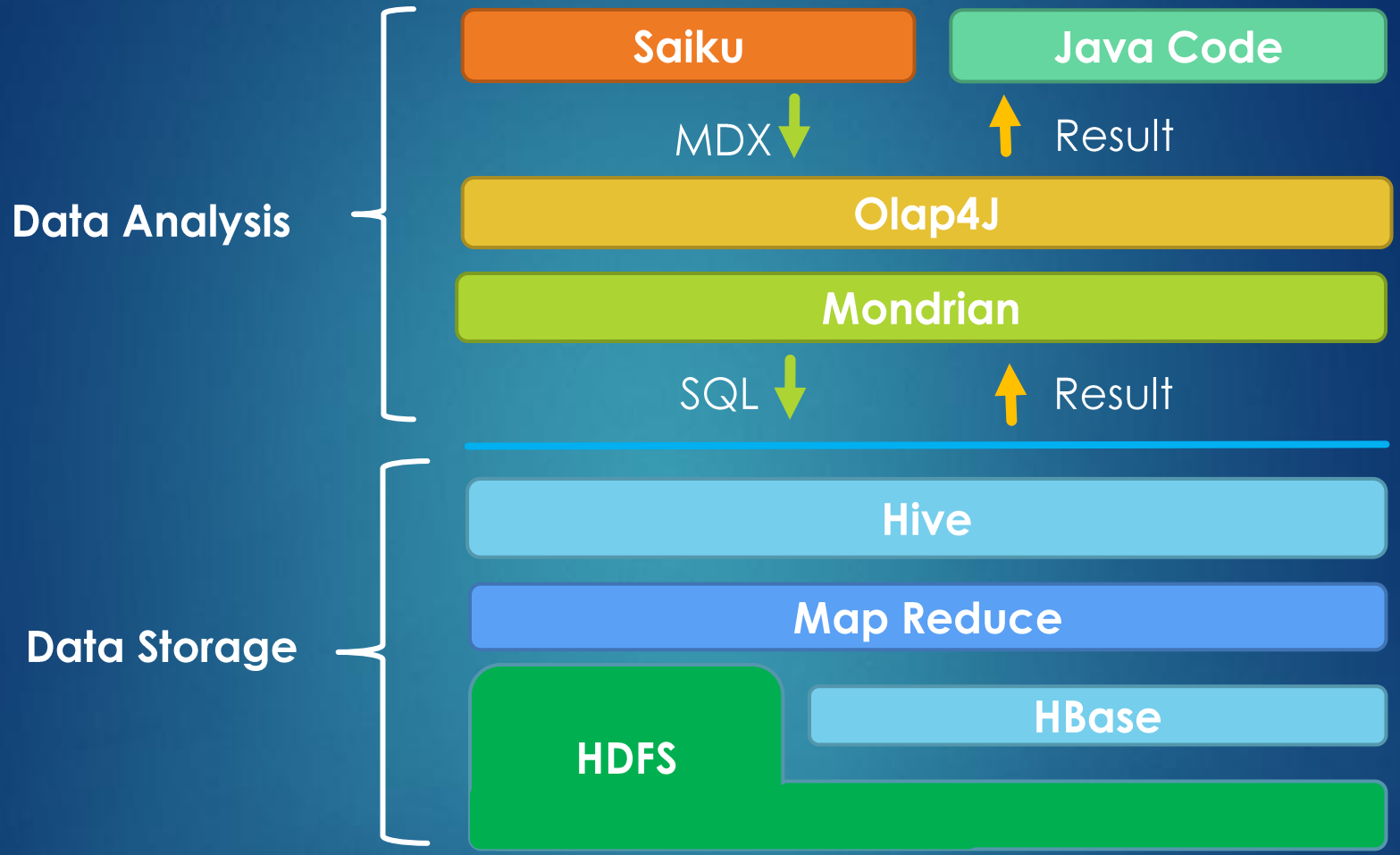
Modrian èl'OLAP server fornito da Penthao . Si basa su tecnologia ROLAP, consente di tradurre query MDX in SQL, basandosi su un modello multidimensionale definito dagli utenti.



Saiku

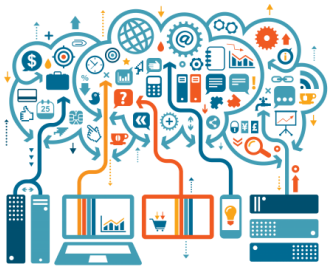
Saiku è un plugin che offre molte possibilità di visualizzazione dei dati in Penthao. Consente la selezione dei dati tramite drag and drop

Architettura del sistema (generazione reports)



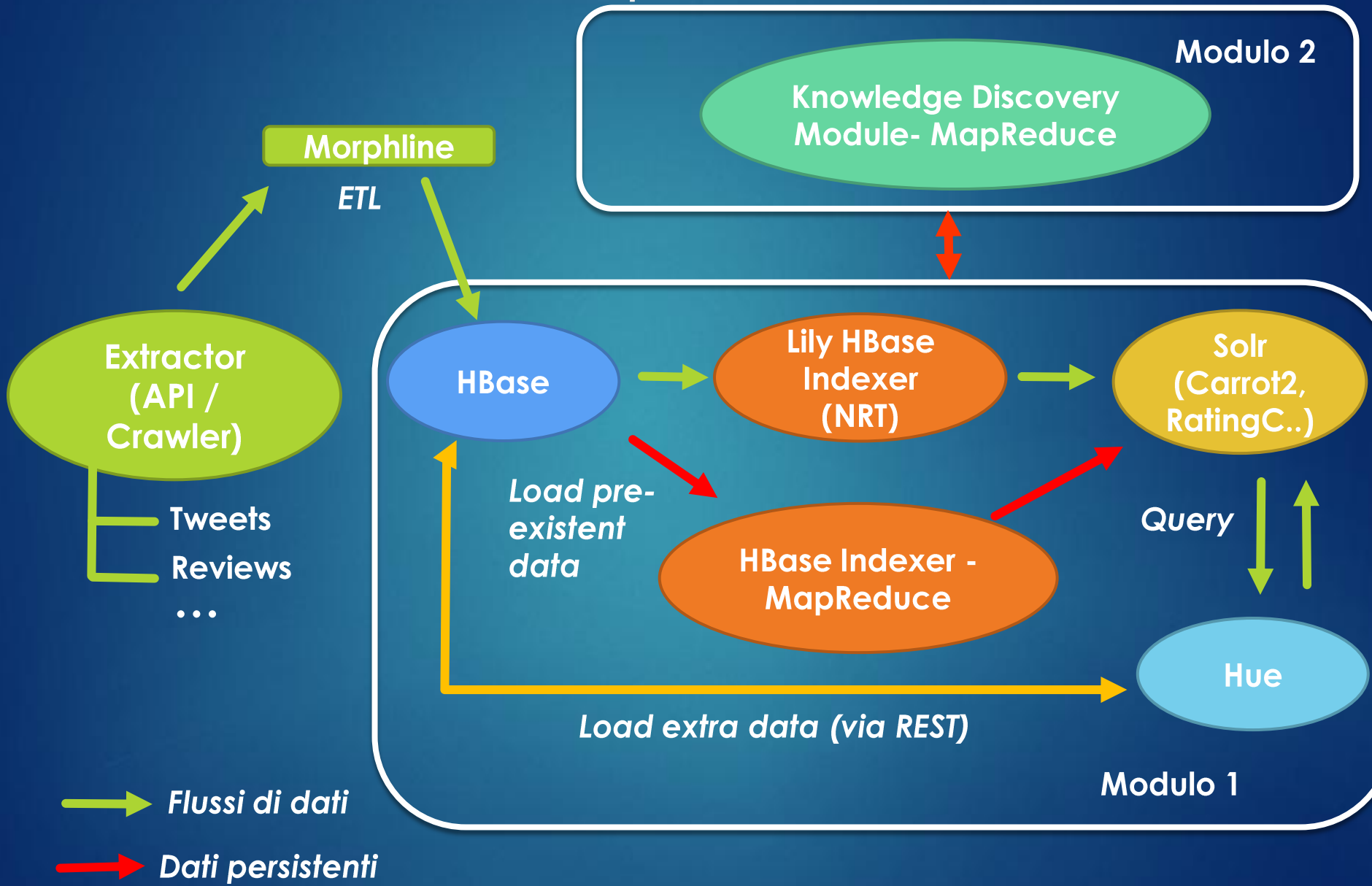
L'architettura proposta è composta da 3 moduli:

1. **Mondrian come OLAP Server**
2. **Hive come esecutore di query su Map Reduce**
3. **HBase o HDFS come Data Storage**

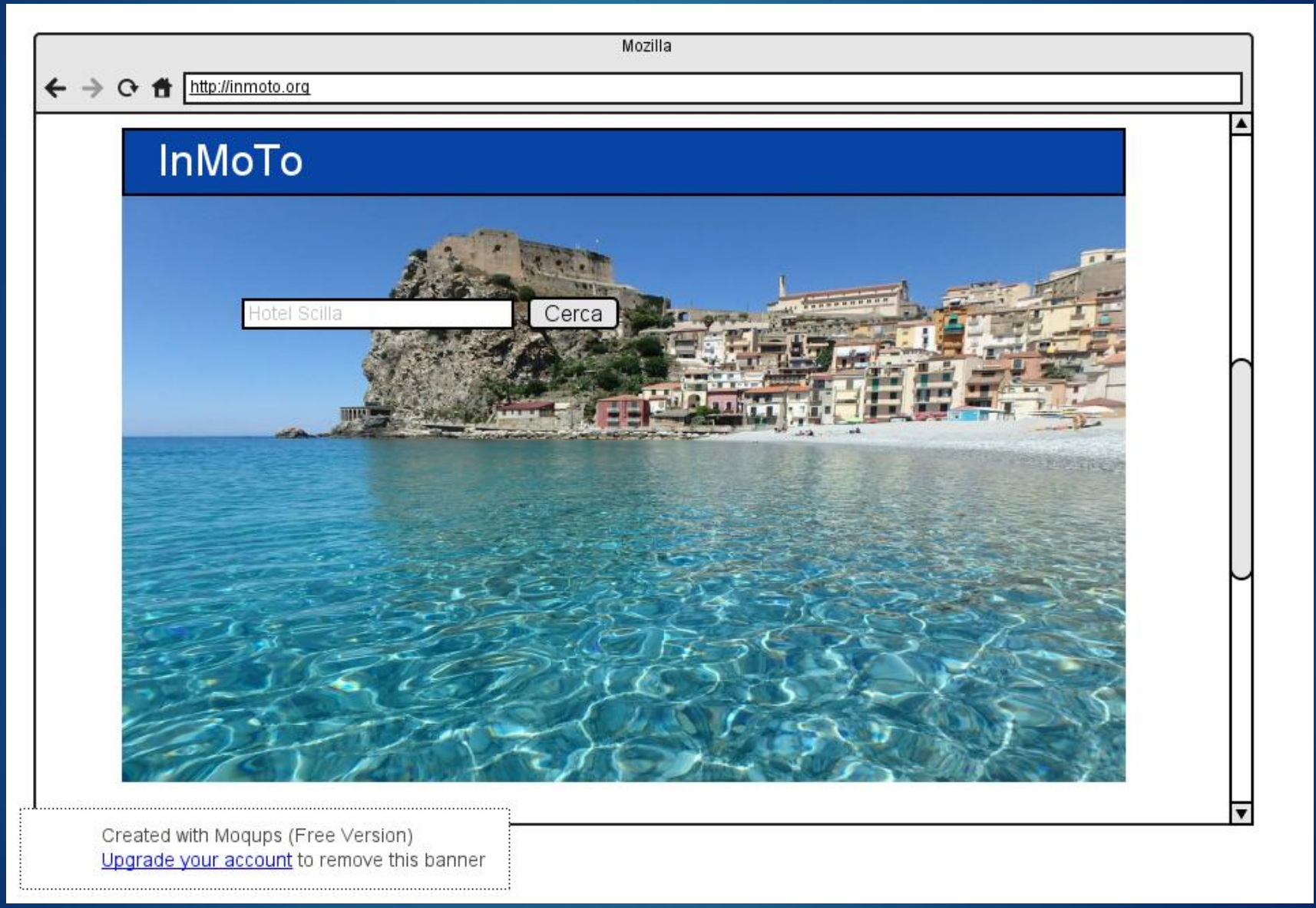


Architettura per Ricerca Full Text

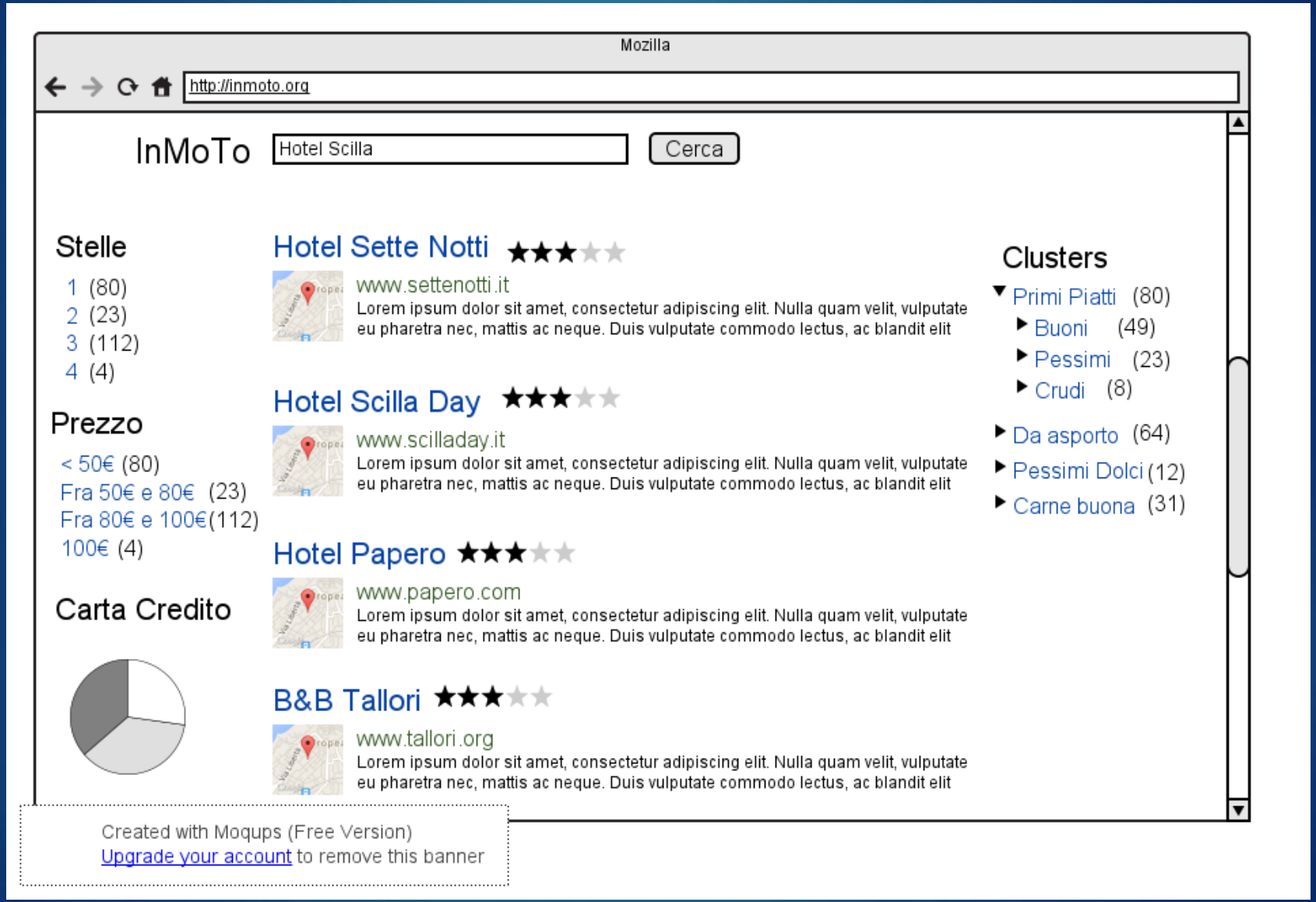
Architettura completa



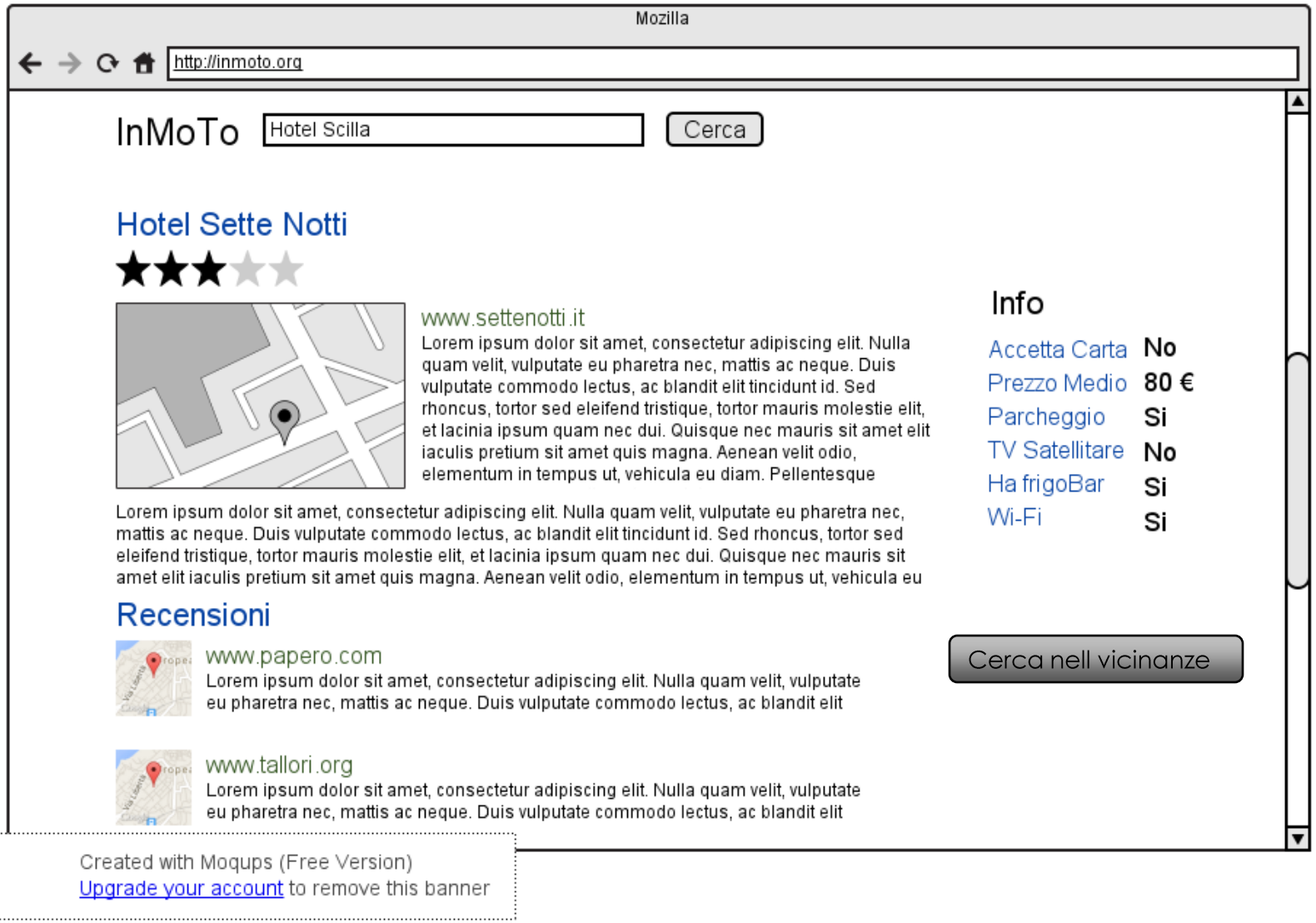
Caso d'uso: Ricerca



Caso d'uso: Visualizzazione Risultati



Caso d'uso: Dettaglio



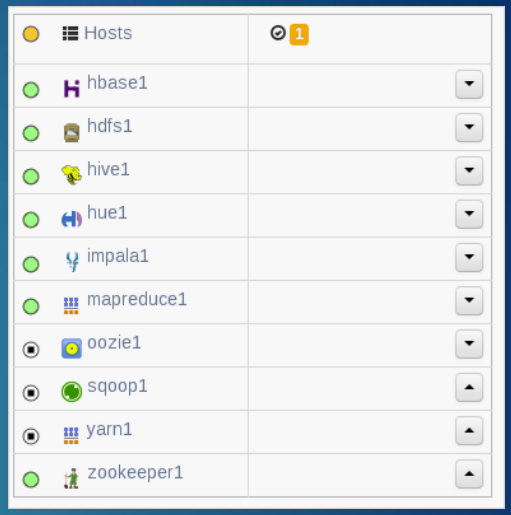
cloudera®

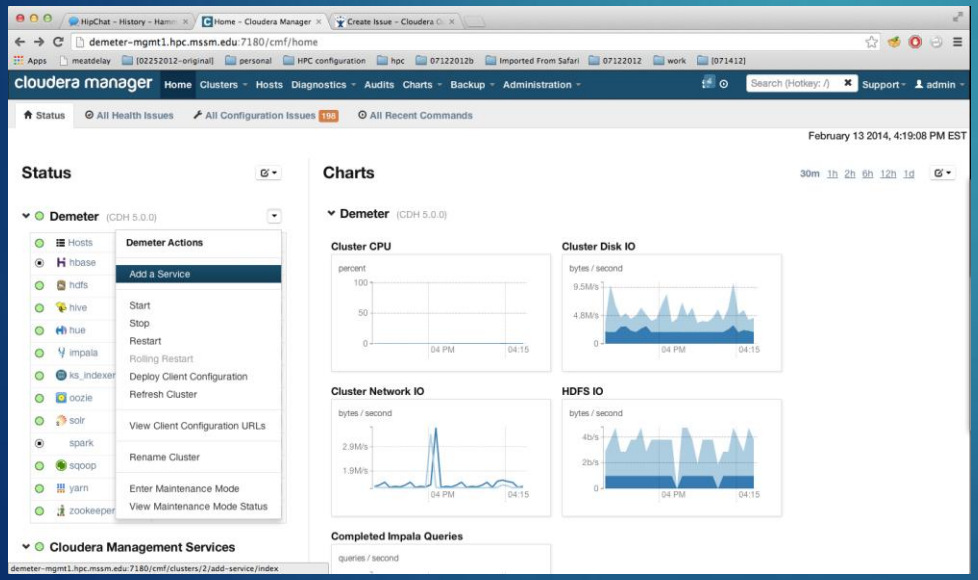
Ask Bigger Questions

La suite Cloudera

La Suite

Cloudera è una suite completa di tutti i componenti piu' utili all'interazione con l'ecosistema Hadoop. Viene chiamata brevemente **CDH** (Cloudera Distribution for Apache Hadoop) ed include un manager che permette tramite browser, l'installazione e la gestione dei vari servizi sui nodi del cluster.





- HDFS
- HBase
- Hadoop (Map-Reduce -Yarn)
- Hive
- Impala
- Sqoop
- Hue
- Zookeeper
- ….

Requisiti

L'installazione di Cloudera richiede l'uso delle più comuni distribuzioni Linux (Centos, Ubuntu, etc..)

Per una installazione minimale è necessario avere tre macchine con la seguente configurazione:

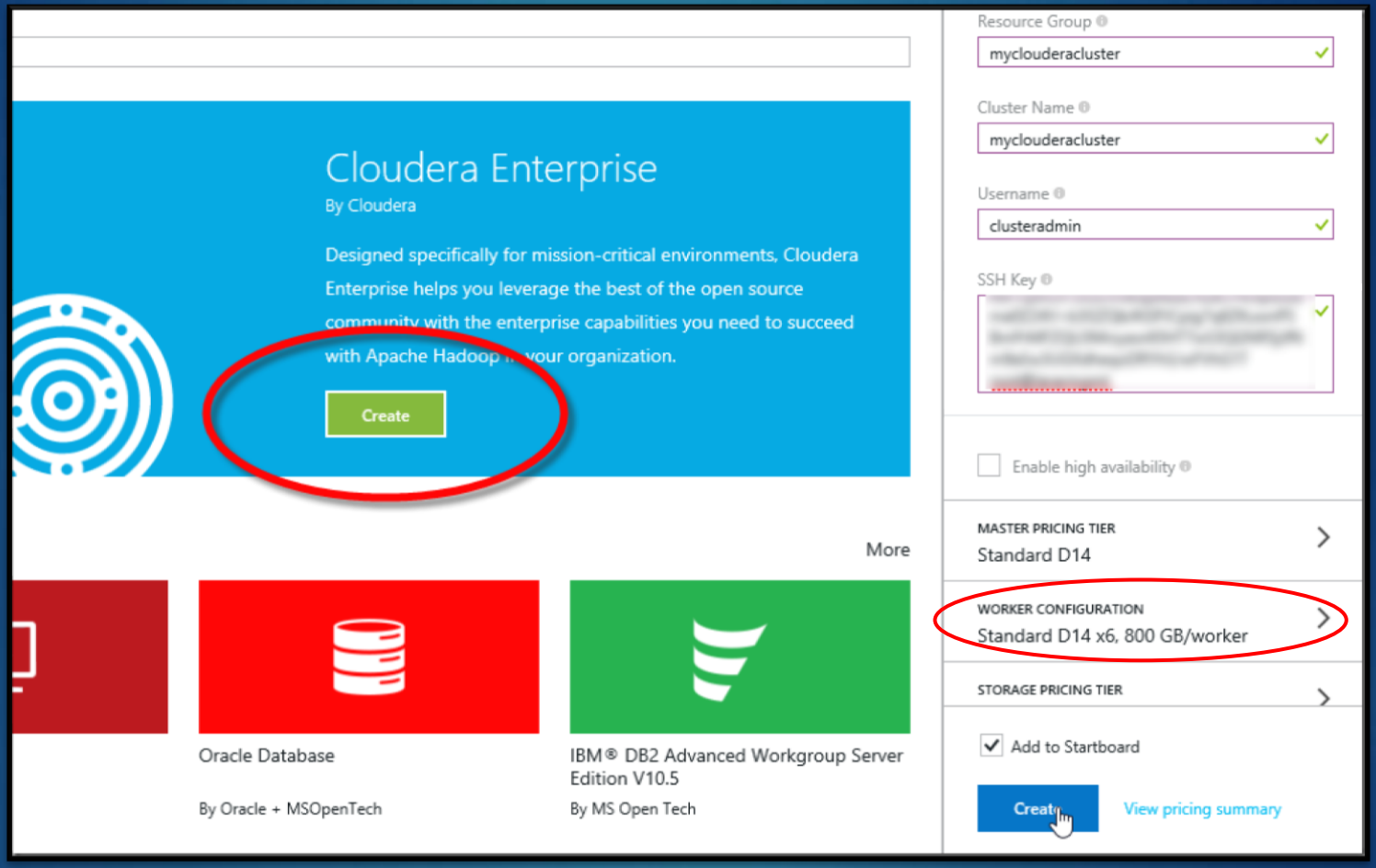
- 8 GB RAM sul nodo master e 4GB sui due nodi slave
- CPU con due core
- 100 GB di hardisk a macchina (N.B. collo di bottiglia del sistema)

I requisiti raccomandati prevedono almeno 10 nodi con la seguente configurazione:

- 32 GB RAM sul nodo master e 16GB sui nodi slave
- CPU con 8 core sul master e almeno 4 sui nodi slave
- 3X500GB di hardisk SSD "dedicato"
- Rete Gigabit

L'installazione di Cloudera è anche supportata "nativamente" dalle soluzioni di cloud più note (per es. Microsoft Azure)

Cloudera su Windows Azure



ISTANZA	CORE	RAM	DIMENSIONI DEI DISCHI	PREZZO
D14	16	112 GB	800 GB	€1,9445/ora (~€1.447 al mese)

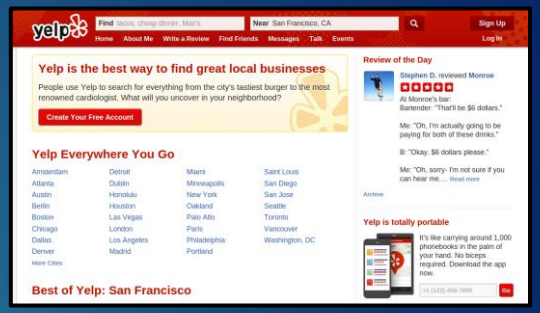
cloudera®

Ask Bigger Questions

DEMO

Yelp su Solr

Yelp è un **social network** in cui le persone si scambiano **pareri**, **opinioni** e "**dritte**" sui posti migliori del luogo in cui abitano e di quelli in cui vanno per lavoro, viaggio o altri motivi.




```
<field name="id" type="string" indexed="true" stored="true" required="true" />
<field name="phone" type="string" indexed="true" stored="true" />
<field name="review_count" type="tint" indexed="true" stored="true" />
<field name="is_closed" type="boolean" indexed="true" stored="true" />
<field name="is_claimed" type="boolean" indexed="true" stored="true" />
<field name="url" type="string" indexed="true" stored="true" />
<field name="name" type="text" indexed="true" stored="true" />
<field name="mobile_url" type="string" indexed="true" stored="true" />
<field name="categories" type="string" indexed="true" stored="true"/>
<field name="rating" type="tfloat" indexed="true" stored="true" />
<field name="image_url" type="string" indexed="true" stored="true" />
<field name="display_phone" type="text" indexed="true" stored="true" />
<field name="distance" type="double" indexed="true" stored="true" />
<field name="snippet_text" type="text" indexed="true" stored="true" />
<field name="snippet_image_url" type="text" indexed="true" stored="true" />
<field name="location" type="text" indexed="true" stored="true" />
<field name="review_id" type="string" indexed="true" stored="true" multiValued="true" />
<field name="review_rating" type="tfloat" indexed="true" stored="true" multiValued="true" />
<field name="review_excerpt" type="text" indexed="true" stored="true" multiValued="true" />
<field name="review_time_created" type="long" indexed="true" stored="true" multiValued="true" />
<field name="user_id" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="user_image_url" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="user_name" type="string" indexed="true" stored="true" multiValued="true"/>
```

Problemi legati all'estrazione dei dati (imitidelle API):

- 25.000 richieste al giorno
- 1.000 risultati ad interrogazione
- 1 recensione per attività

http://192.168.0.16:8983/solr/yelp_shard1_replica1/clustering?q=text:ristorante+AND+text:milano&rows=100

Google Book n-grams

- **N-grams**
  - Un **n-gramma** è una sottosequenza di n elementi di una data sequenza.
  - In questo caso gli elementi sono parole estratte da Google Book
- Dataset
  - File compressi plain text tab-separated
    - Ngram: l'attuale n-gramma
    - Year: anno di aggregazione
    - Occurrences: Numero di volte che l'n-gramma appare nell'anno
    - Book: numero di libri in cui appare l'n-gramma

Raggruppamento termini per deacde con ratio

- ```
  CREATE TABLE by_decade (
   gram string,
   decade int,
   ratio double
  );
  ```
- ```
  INSERT OVERWRITE TABLE by_decade
  SELECT
   a.gram,
   b.decade,
   sum(a.occurrences) / b.total
  FROM
   normalized a
  JOIN (
   SELECT
    substr(year, 0, 3) as decade,
    sum(occurrences) as total
   FROM
    normalized
   GROUP BY
    substr(year, 0, 3)
  ) b
  ON
   substr(a.year, 0, 3) = b.decade
  GROUP BY
   a.gram,
   b.decade,
   b.total;
  ```

Trend dei termini per decade


```
SELECT
 a.gram as gram,
 a.decade as decade,
 a.ratio as ratio,
 a.ratio / b.ratio as increase
FROM
 by_decade a
JOIN
 by_decade b
ON
 a.gram = b.gram and
 a.decade - 1 = b.decade
WHERE
 a.ratio > 0.0001 and
 a.decade >= 190
SORT BY
 decade ASC,
 increase DESC;
```

polo innovazione
ICT Calabria

ICAR CNR
Sede di Cosenza
Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

UNIVERSITÀ DELLA CALABRIA
DIMES
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Seminario

Strumenti Open Source utilizzabili nel contesto Big Data

31 Marzo 2015

Ing. Cassavia Nunziato (ICAR-CNR)
Ing. Dicosta Pietro (UNICAL)