Hadoop
及其相关系统介绍

石立元　**2012年07月19日**

PresentationPoint

总提纲

- Hadoop的基本原理与基本操作
- HIVE的基本原理与基本操作
- SQOOP的基本原理与基本操作
- Hadoop其它相关系统介绍

提纲

- Hadoop的构成与原理
- Hadoop的性能特点与应用
- 使用Hadoop
- Hadoop优缺点与应用方案

提纲

- Hadoop的构成与原理
- Hadoop的性能特点与应用
- 使用Hadoop
- Hadoop优缺点与应用方案

Hadoop是什么

Hadoop是什么？ Hadoop原来是Apache Lucene下的一个子项目，专门负责分布式存储以及分布式运算。简单地说来，Hadoop是一个可以更容易开发和运行处理大规模数据的软件平台。 Hadoop框架最核心的设计：MapReduce和HDFS。

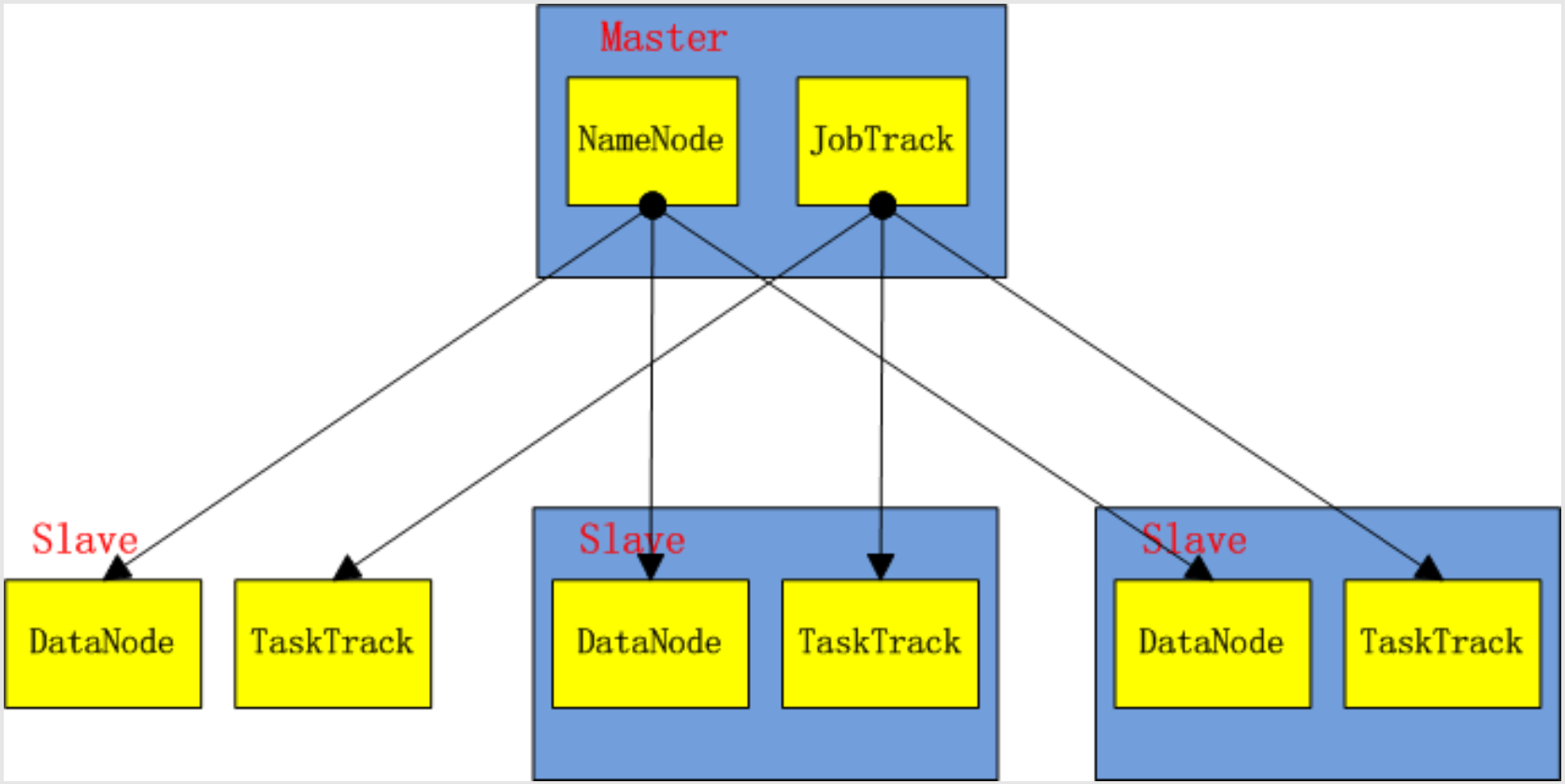
HDFS

- Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有着高容错性的特点，并且设计用来部署在低廉的硬件上。而且它提供高传输率来访问应用程序的数据，适合那些有着超大数据集的应用程序。

MapReduce

- 简单的一句话解释MapReduce就是“任务的分解与结果的汇总”
- MapReduce将应用程序的工作分解成很多小的工作小块。HDFS为了做到可靠性创建了多份数据块的复制，并将它们放置在服务器群的计算节点中，MapReduce就可以在它们所在的节点上处理这些数据了。

Hadoop机群整体结构



HDFS交互关系说明

文件写入

- Client向NameNode发起文件写入的请求。
- NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。
- Client将文件划分为多个Block，根据DataNode的地址信息，按顺序写入到每一个DataNode块中。

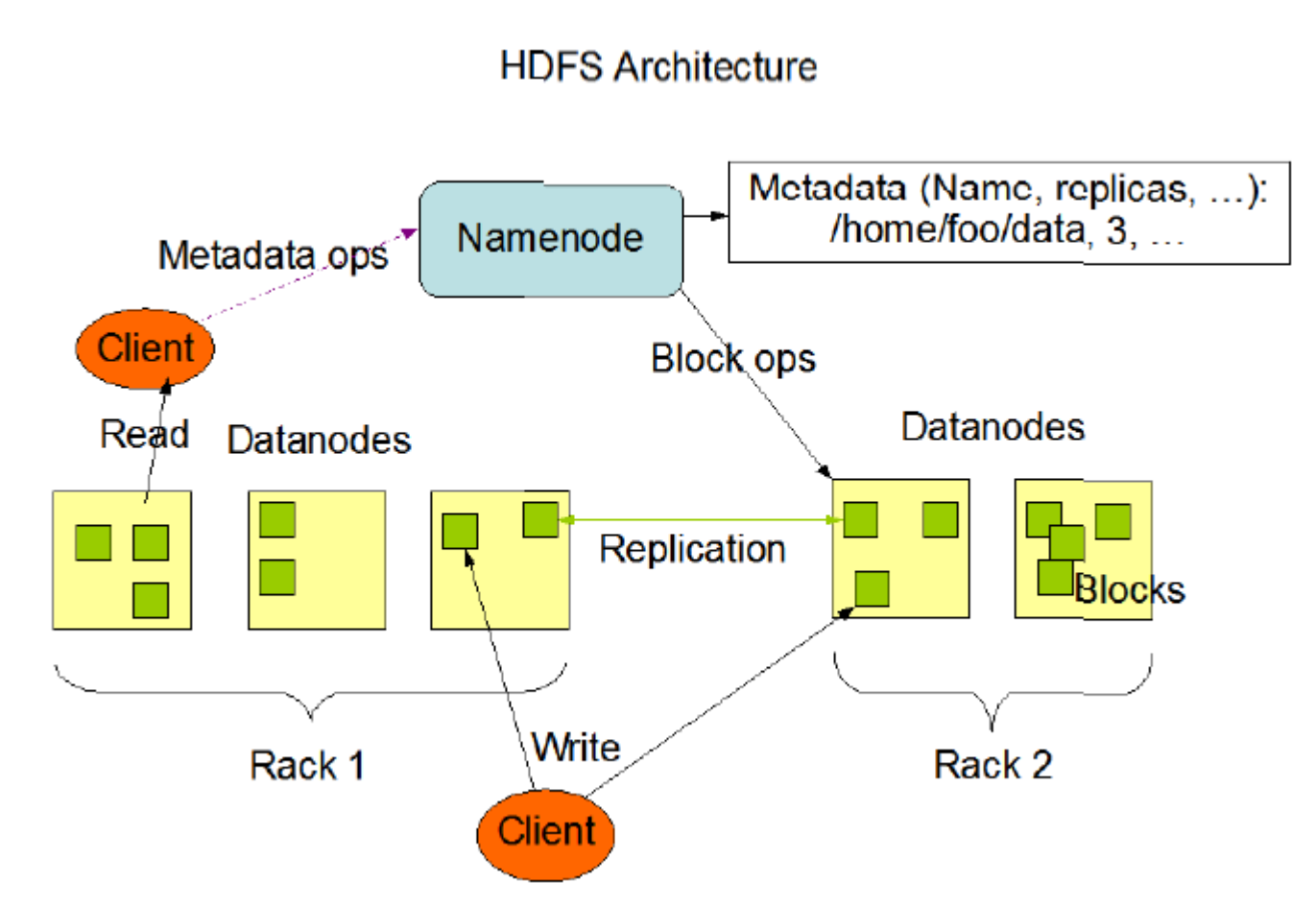
文件读取

- Client向NameNode发起文件读取的请求。
- NameNode返回文件存储的DataNode的信息。
- Client读取文件信息。

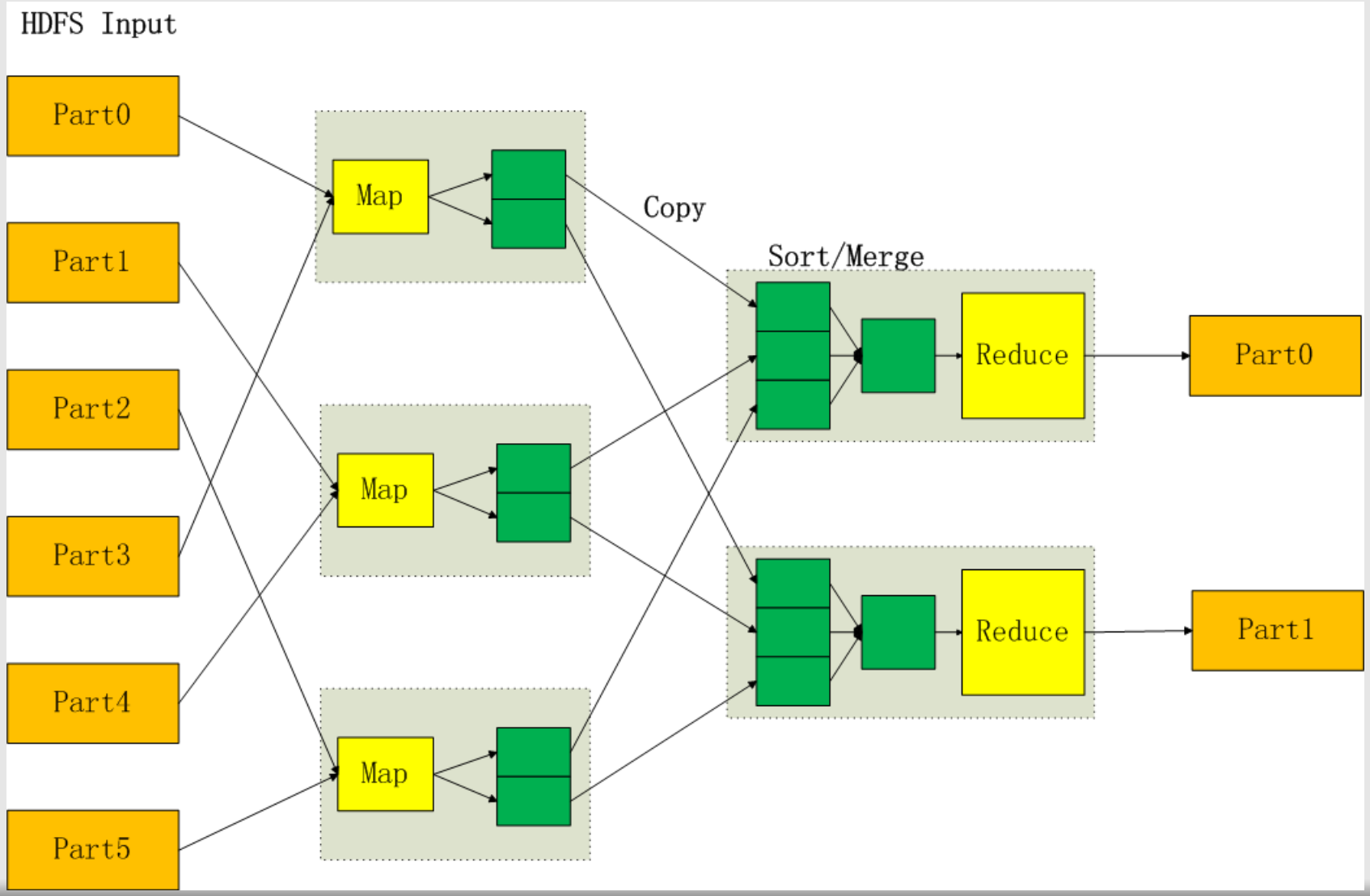
文件Block复制

- NameNode发现部分文件的Block不符合最小复制数或者部分DataNode失效。
- 通知DataNode相互复制Block。
- DataNode开始直接相互复制。

HDFS结构示意图


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

MapReduce结构示意图



提纲

- Hadoop的构成与原理
- Hadoop的性能特点与应用
- 使用Hadoop
- Hadoop优缺点与应用方案

Hadoop特点

hadoop主要的一些特点：

- **扩容能力**：能可靠地存储和处理千兆字节（PB）数据。
- **成本低**：可以通过普通机器组成的服务器群来分发以及处理数据。这些服务器群总计可达数千个节点。
- **高效率**：通过分发数据，hadoop可以在数据所在的节点上并行地处理它们，这使得处理非常的快速。
- **可靠性**：hadoop能自动地维护数据的多份复制，并且在任务失败后能自动地重新部署计算任务。

为什么选择Hadoop

- Hadoop性能测试（环境：11台hadoop机群）
  - **在一个月的浏览数据6亿多条中找出首页的浏览次数**
    - 单机脚本：110分钟
    - 同样的脚本放入Hadoop：2分40秒
    - 扩展性：增加机器后计算时间仍可继续缩短
  - 月UV统计（7亿级去重）
    - 1分40秒，足够机器下可达30秒左右
  - 年UV统计（63亿以上去重）
    - 16分钟，足够机器下可达2分钟以内
- Hadoop在相关领域的发展状态
  - Yahoo ：34个集群，总数超过3万台机器，最大的集群是 4000台左右，总存储容量超过100PB
  - 淘宝：单个集群规模2000台，实际存储数据超过17PB，日运行mapreduce job达6万个，开发团队240余人

Hadoop在当当

一、Hadoop系统当前的状态

1、目前hadoop平台拥有布有36台机器。

2、每台机器 的配置为：2C四核，32G，1T_RAID0*6，CentOS5.4 64bit。

3、其中35台为计算节点，共设280个cpu计算资源，存储容量175T

二、目前在hadoop上运行的项目

1、ddclick：

a.流量数据的存储与常用指标计算

2、研究开发组：

a.当首馆首流量(各专题单品)

b.专题页统计

c.首页轮转统计

d.推荐效果统计

e.当首所有链接分析

f. 基础数据(浏览树等)生成

g.未设定时执行但随时可运行的任务：常见搜索引擎带来的流量订单统计；任意指定路径的流量收订情况；任意起始位置流量收订统计；等等

提纲

- Hadoop的构成与原理
- Hadoop的性能特点与应用
- 使用Hadoop
- Hadoop优缺点与应用方案

使用HDFS

- Hadoop包括一系列的类shell的命令，可直接和HDFS以及其他Hadoop支持的文件系统进行交互。
- 查看目录“hadoop dfs –ls /root”
- 将本地文件存储到HDFS“hadoop dfs –put test.txt /root/.”
- 将HDFS文件存储到本地“hadoop dfs –get /root/test.txt test.txt”
- 删除文件“hadoop dfs –rm /root/test.txt”
- 移动文件或目录 “hadoop dfs –mv /root/ /test/”
- 创建文件夹 “hadoop dfs –mkdir /root/”

Hadoop开发实例mapper.py


```
#!/usr/bin/env python

import sys
import re

for line in sys.stdin:
   url= line.strip()
   if len(re.findall('^http://product.dangdang.com/product.aspx\?product_id=\d+', url))
   > 0:
            print '%s\t%s' % (url, 1)
```

Hadoop开发实例reducer.py


```
#!/usr/bin/env python
import sys
urlcount = {}
for line in sys.stdin:
   line = line.strip()
   url, count = line.split()
   try:
        count = int(count)
        urlcount[url] = urlcount.get(url, 0) + count
    except ValueError:
       pass
for url, count in urlcount :
    print '%s\t%s'% (url, count)
```

在**hadoop**上执行


```
bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \
  -mapper /home/hadoop/mapper.py\
  -reducer /home/hadoop/reducer.py \
  -input doc/* \
  -output python-output
```

其他语言以此类推

提纲

- Hadoop的构成与原理
- Hadoop的性能特点与应用
- 使用Hadoop
- Hadoop优缺点与应用方案

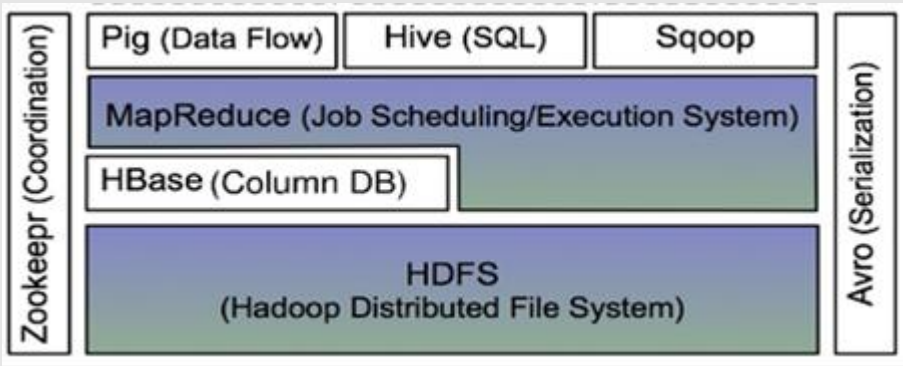
Hadoop优缺点

- Hadoop的长处
  - 系统开源免费
  - 操作相对简单
  - HDFS与MapReduce框架比传统的SQL数据库更适于数据仓库大数据
  - 大吞吐量
  - 高性价比：其精巧设计使大数据计算性能在低成本下得到极大提高

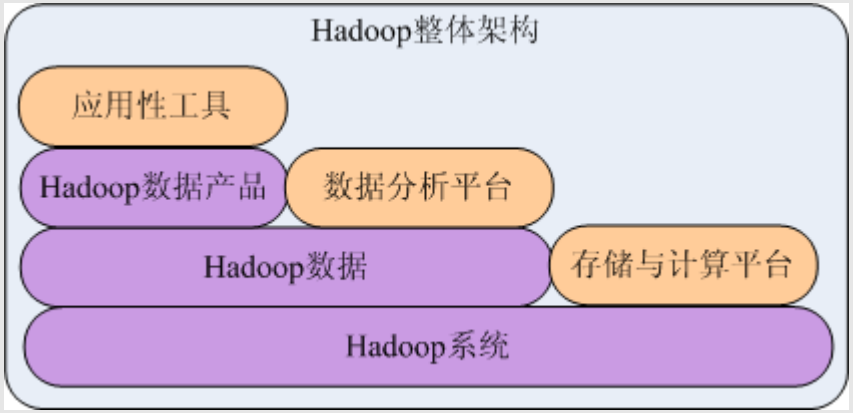
- Hadoop的不足
  - 多版本之间的兼容性较差
  - 开源产品没有服务保障
  - 无法实时响应
  - 调试较为复杂开发周期长

Hadoop系统架构

- Hadoop系统基础架构



- Hadoop系统整体架构



- HIVE简介
- HIVE表结构
- HIVE操作
- HIVE实现原理
- HIVE特点

- HIVE
  - HIVE是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供类似sql查询功能，可以将sql语句转换为MapReduce任务进行运行。

- HIVE表
  - Metastore
  - HDFS基础数据

  - 内部表/外部表
  - 分区partition

- 界面

```
[shiliyuan@h252021 ~]$ hive
Hive history file=/tmp/shiliyuan/hive_job_log_shiliyuan_201202281353_279075496.txt
hive> show databases;
```

- HQL
  - Show tables
  - Create Table
  - Alter Table
  - SELECT
  - Join

- 例：某需求需要得到网页每个查询词的查询次数
  - hive> select query,count(1) from web where uigs_type='pv' group by query;
- 例：计算UV
  - hive>select count(distinct perm_id) from data_table;
- 优点：**简单**

Create table

- CREATE [EXTERNAL] TABLE [IF NOT EXISTS] table_name [(col_name data_type [COMMENT col_comment], ...)]
- [table_comment]
- [PARTITIONED BY (col_name data_type [col_comment], ...)]
- [ROW FORMAT row_format]
- [STORED AS file_format]
- [LOCATION hdfs_path]

- 例：
- create table web(time string,ip string,suv string,url string,uigs_type String, p string,w String,query string) PARTITIONED BY (ds string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

- **Loading files into tables**
  - LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)]
  - 例：
  - LOAD DATA INPATH '/root/user/webhivedata/200912/20091220' INTO TABLE web PARTITION (ds=20091220);

- **Select**
  - SELECT [ALL | DISTINCT] select_expr, select_expr, ...
  - FROM table_reference
  - [**WHERE** where_condition]
  - [**GROUP BY** col_list]
  - [**CLUSTER BY** col_list
  - | [**DISTRIBUTE BY** col_list] [SORT BY col_list]
  -]
  - [**LIMIT** number]

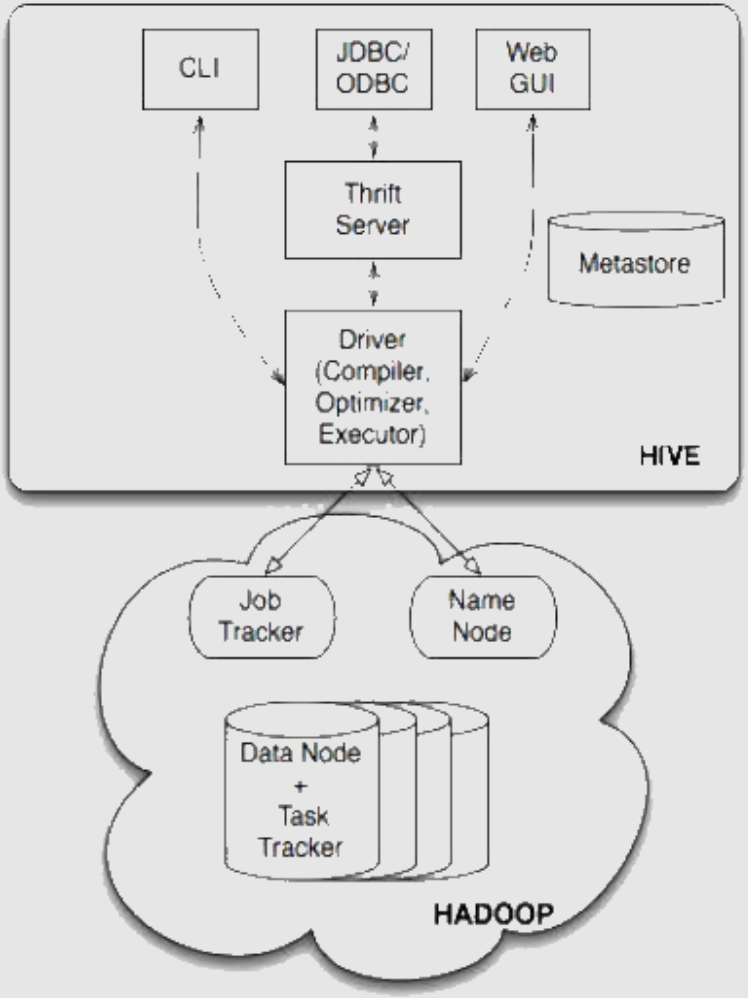
  - 例：
  - select query,count(1) as qnum,count(distinct suv) as quvnum from web where ds='20091225' and uigs_type='pv' and query is not null sort by qnum desc limit 100

- 其它用法
  - Insert overwrite table
  - Drop Table
  - Describe Table

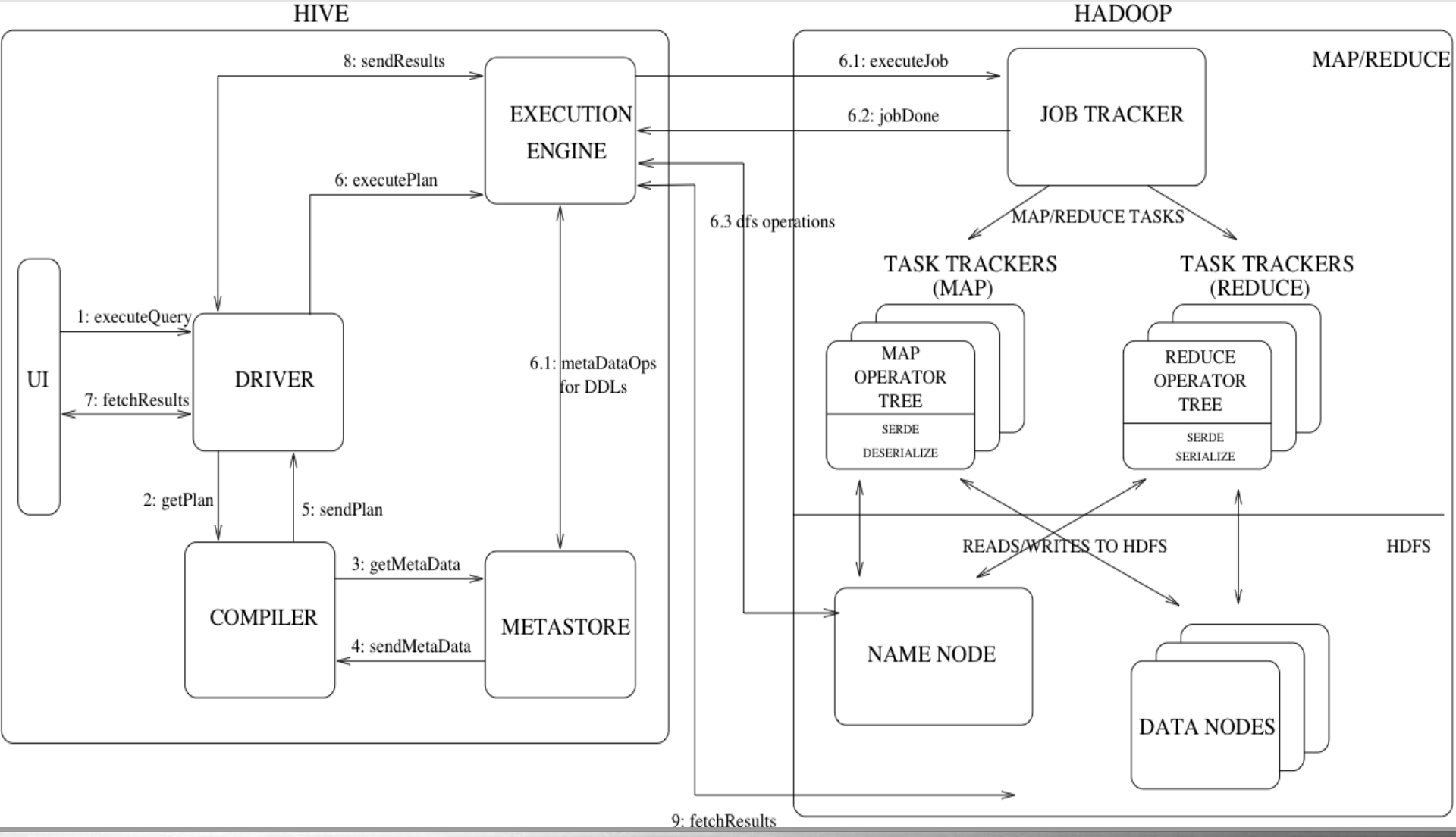
  - Hive语法参考手册
    http://wiki.apache.org/hadoop/Hive/LanguageManual

HIVE实现原理

- HIVE操作原理



HIVE实现原理


HIVE
HADOOP
MAP/REDUCE
8: sendResults
6.1: executeJob
EXECUTION ENGINE
6.2: jobDone
JOB TRACKER
6: executePlan
6.3 dfs operations
MAP/REDUCE TASKS
TASK TRACKERS (MAP)
TASK TRACKERS (REDUCE)
1: executeQuery
UI
DRIVER
7: fetchResults
6.1: metaDataOps for DDLs
MAP OPERATOR TREE
SERDE DESERIALIZE
REDUCE OPERATOR TREE
SERDE SERIALIZE
2: getPlan
5: sendPlan
READS/WRITES TO HDFS
HDFS
3: getMetaData
COMPILER
METASTORE
4: sendMetaData
NAME NODE
DATA NODES
9: fetchResults

HIVE特点

- Hive主要应用于大规模结构化数据分析、统计。
- Hive实质上是**hadoop**的一个客户端,只是把产生mapreduce任务用一个sql编译器自动化了。
- Hive不支持对数据的修改、删除操作。
- Hive查询是非实时，根据数据量不同统计时间一般在分钟级。
- HQL不支持having，in，exist等用法
- HQL只支持等值连接
- HQL可执行脚本
- HQL没有SQL丰富，但可自定义UDF
- 复杂的HQL需要拆分，通过中间临时表过渡
- 有时没有专门的mapreduce处理高效

- 更多参考
  - https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Transform
  - https://cwiki.apache.org/confluence/display/Hive/GettingStarted

- SQOOP简介
- SQOOP操作
- SQOOP原理

- SQOOP
  - Sqoop是一个用来将Hadoop和关系型数据库中的数据相互转移的工具，可以将一个关系型数据库（例如:MySQL ,Oracle ,Postgres等）中的数据导入到Hadoop的HDFS中，也可以将HDFS的数据导入到关系型数据库中。
  - 注意：Sqoop目前在Apache 版本的Hadoop 0.20.2上是无法使用的

SQOOP使用

- 界面

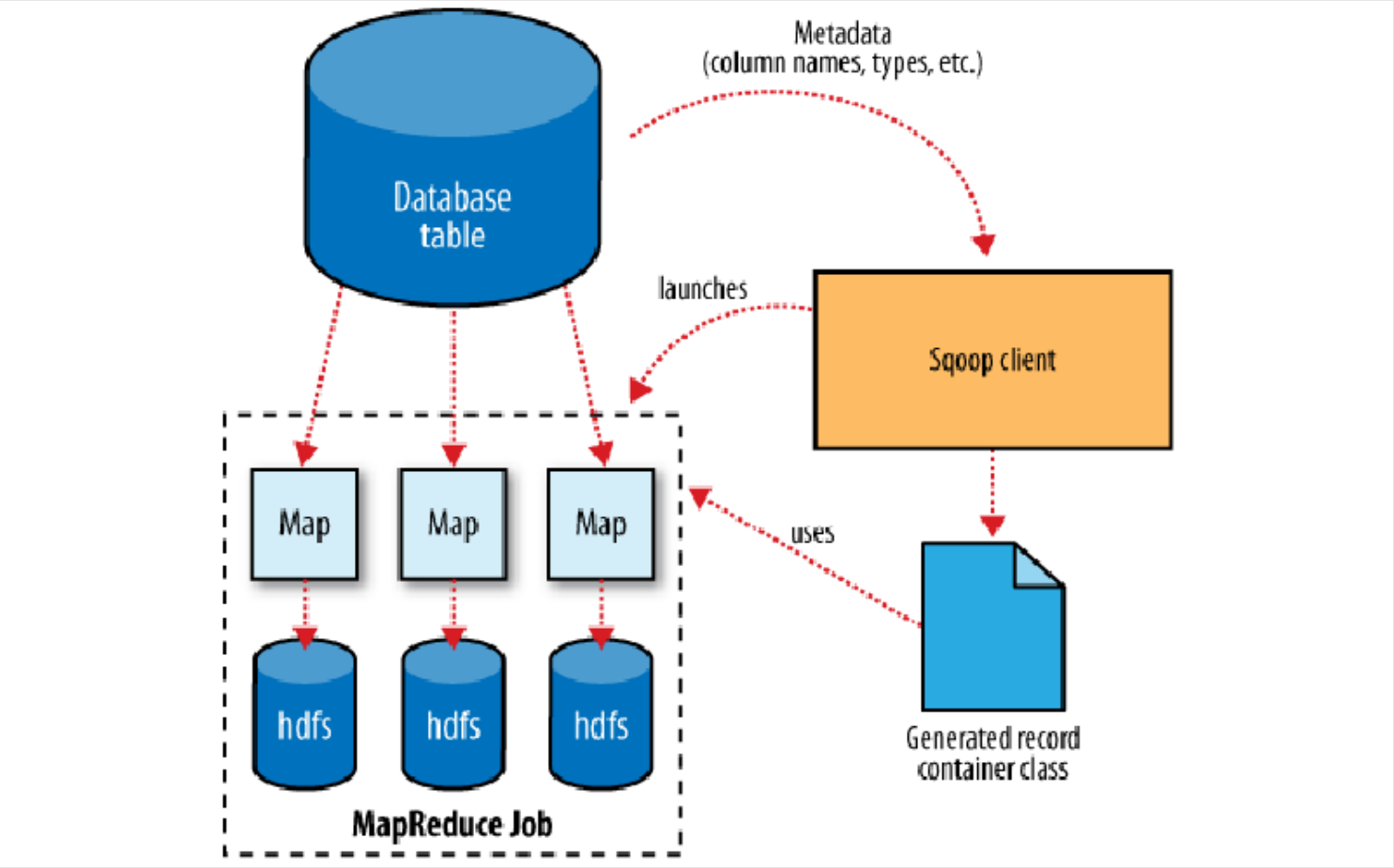
```
[shiliyuan@h252021 bin]$ ./sqoop help
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information
```

- 导入示例
  - ./sqoop import --connect jdbc:mysql://172.16.252.21:3306/DBname --table （表名） --username writeuser -P --hive-import --split-by indexORkey

SQOOP原理

- SQOOP导入过程



- Hbase简介
- Hbase表结构
- Hbase操作
- Hbase原理
- Hbase特点
- Hbase与HIVE

- Hbase
  - HBase (Hadoop Database)是Apache的Hadoop项目的子项目，HBase是一个分布式的、面向列的开源数据库，该技术来源于Google Bigtable。
  - HBase利用Hadoop HDFS作为其文件存储系统，利用Hadoop MapReduce来处理HBase中的海量数据，利用Zookeeper作为协同服务。

HBase表结构

- HBase的表结构
  - HBase以表的形式存储数据。表有行和列组成。列划分为若干个列族/列簇(column family)
  - 主要包含：
    - 行键
    - 列族
    - 单元
    - 时间戳

Row Key	column-family1		column-family2			column-family3
	column1	column2	column1	column2	column3	column1
key1	t1:abc t2:gdxdf		t4:dfads t3:hello t2:world			
key2	t3:abc t1:gdxdf		t4:dfads t3:hello		t2:dfdsfa t3:dfdf	
key3		t2:dfadfasd t1:dfdasddsf				t2:dfxxdfasd t1:taobao.com

HBase表结构

- 行键 Row key
  - 可以是任意字符串(最大长度是 64KB，实际应用中长度一般为 10-100bytes)，在hbase内部，row key保存为字节数组。存储时，数据按照Row key的字典序(byte order)排序存储。
- 列族 column family
  - hbase表中的每个列，都归属与某个列族。列族是表的chema的一部分(而列不是)，必须在使用表之前定义。列名都以列族作为前缀。例如courses:history ，courses:math 都属于 courses 这个列族。
- 单元 Cell
  - HBase中通过row和columns确定的为一个存贮单元称为cell。
- 时间戳 timestamp
  - 每个cell都保存着同一份数据的多个版本。版本通过时间戳来索引。时间戳的类型是 64位整型。时间戳可以由hbase(在数据写入时自动)赋值，此时时间戳是精确到毫秒的当前系统时间。时间戳也可以由客户显式赋值。如果应用程序要避免数据版本冲突，就必须自己生成具有唯一性的时间戳。每个cell中，不同版本的数据按照时间倒序排序，即最新的数据排在最前面。

- 界面

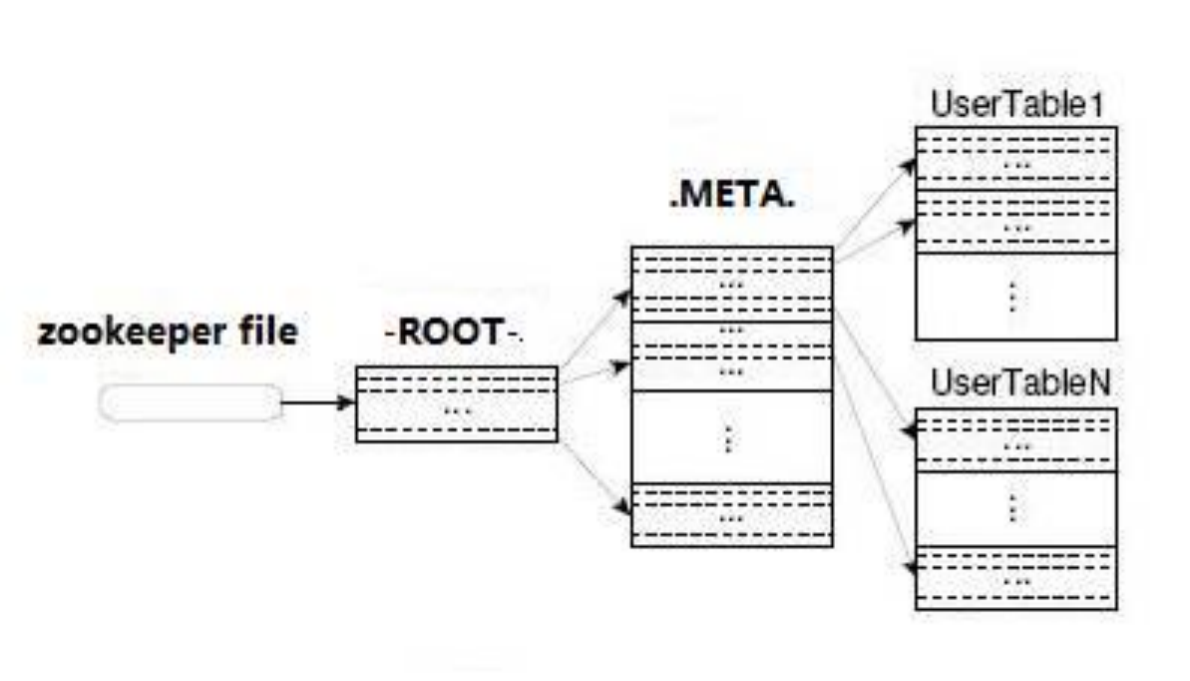
```
[shiliyuan@h252021 ~]$ hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.90.4-cdh3u2, r, Thu Oct 13 20:32:26 PDT 2011

hbase(main):001:0> list
```

- 常见命令
  - list
  - create
  - put
  - get
  - scan

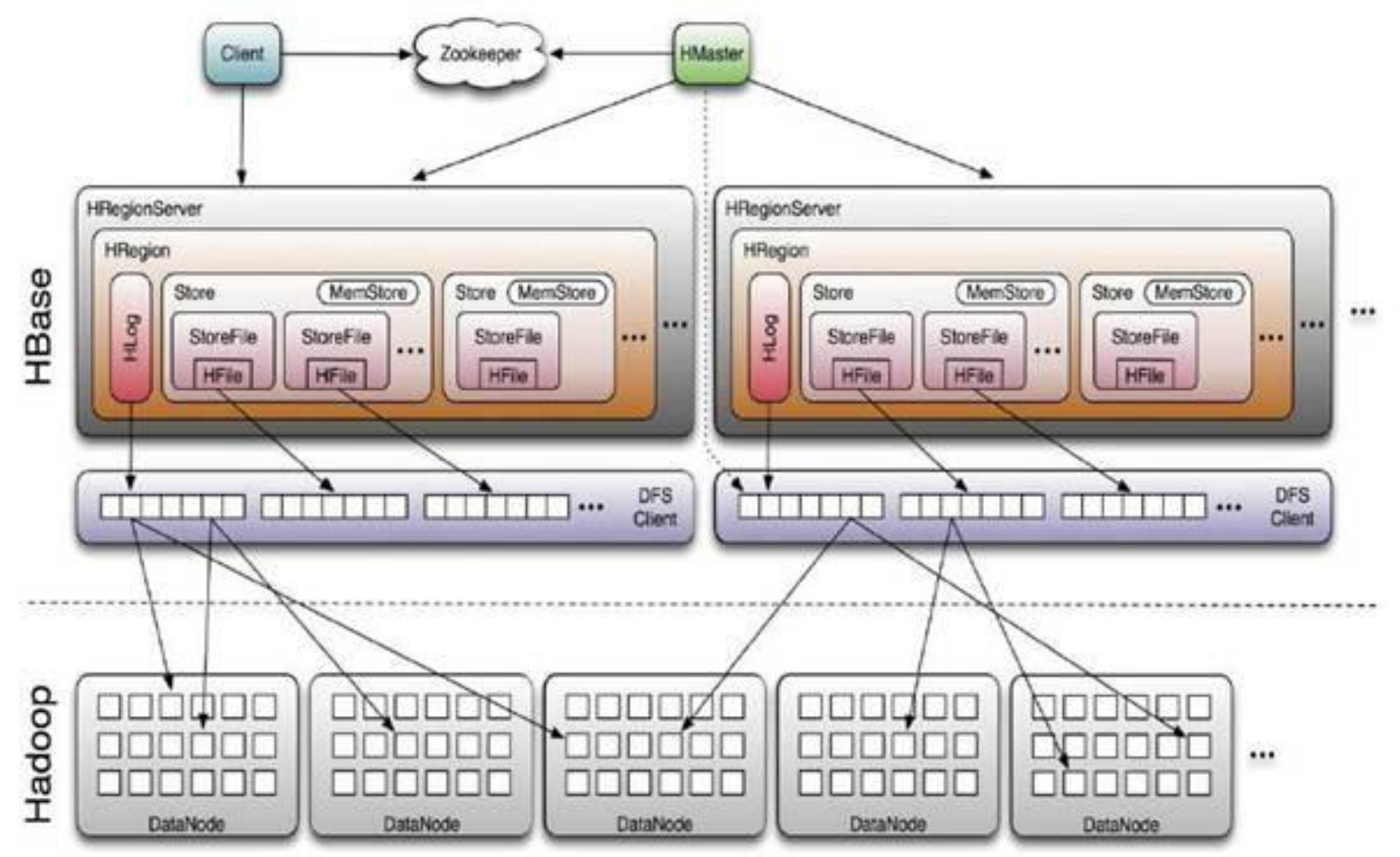
Hbase的原理

- Hbase的访问



Hbase的原理

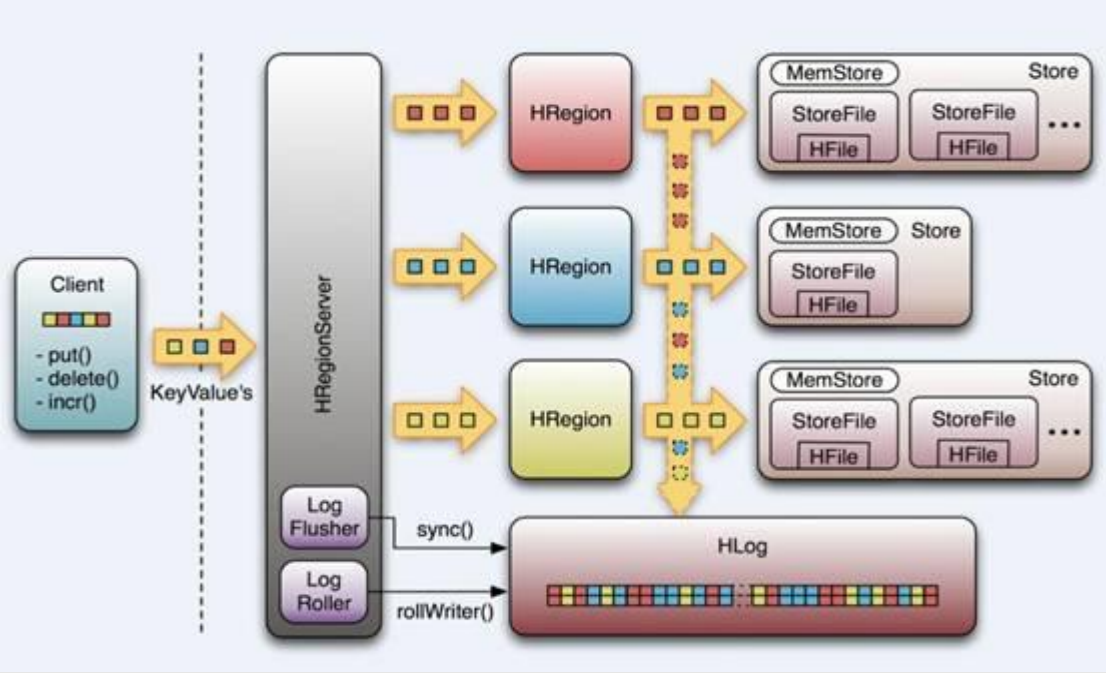
- HBase系统架构



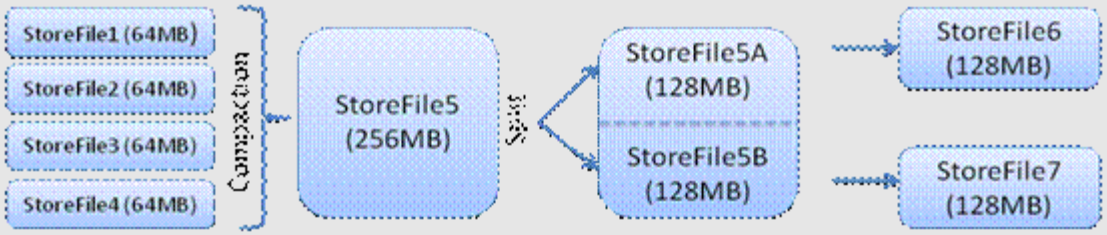
- Hmaster
  - HMaster没有单点问题，HBase中可以启动多个HMaster，通过Zookeeper的Master Election机制保证总有一个Master运行，HMaster在功能上主要负责Table和Region的管理工作：
    - 管理用户对Table的增、删、改、查操作
    - 管理HRegionServer的负载均衡，调整Region分布
    - 在Region Split后，负责新Region的分配
    - 在HRegionServer停机后，负责失效HRegionServer 上的Regions迁移

Hbase的原理

- HRegionServer
  - HRegionServer主要负责响应用户I/O请求，向HDFS文件系统中读写数据，是HBase中最核心的模块。
  - HRegionServer内部管理了一系列HRegion对象，每个HRegion对应了Table中的一个Region，HRegion中由多个HStore组成。每个HStore对应了Table中的一个Column Family的存储

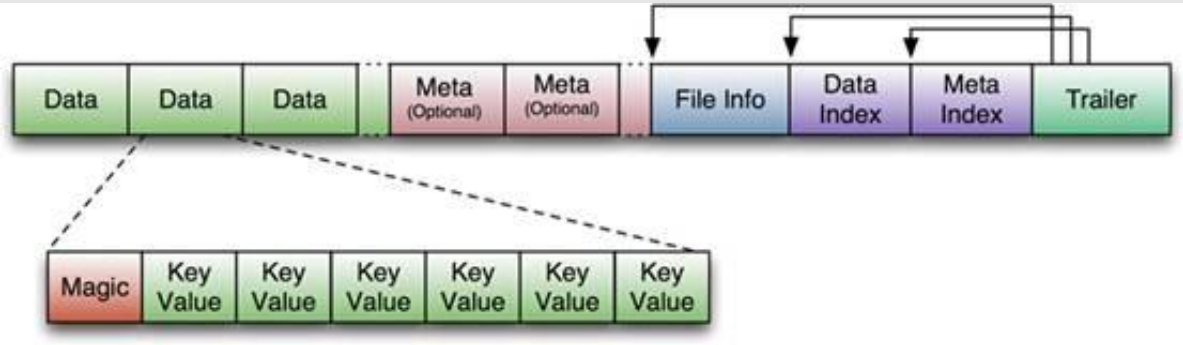


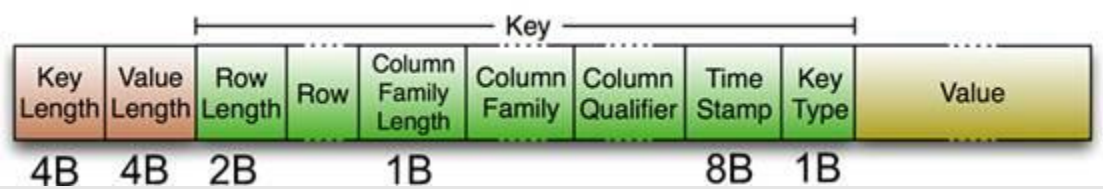
- Hstore
  - HStore存储是HBase存储的核心，其中由两部分组成，一部分是MemStore，一部分是StoreFiles。MemStore是Sorted Memory Buffer，用户写入的数据首先会放入MemStore，当MemStore满了以后会Flush成一个StoreFile（底层实现是HFile），当StoreFile文件*数量*增长到一定阈值，会触发Compact合并操作，将多个StoreFiles合并成一个StoreFile。当StoreFiles Compact后，会逐步形成越来越大的StoreFile，当单个StoreFile*大小*超过一定阈值后，会触发Split操作，同时把当前Region Split成2个Region，父Region会下线，新Split出的2个孩子Region会被HMaster分配到相应的HRegionServer上，使得原先1个Region的压力得以分流到2个Region上。



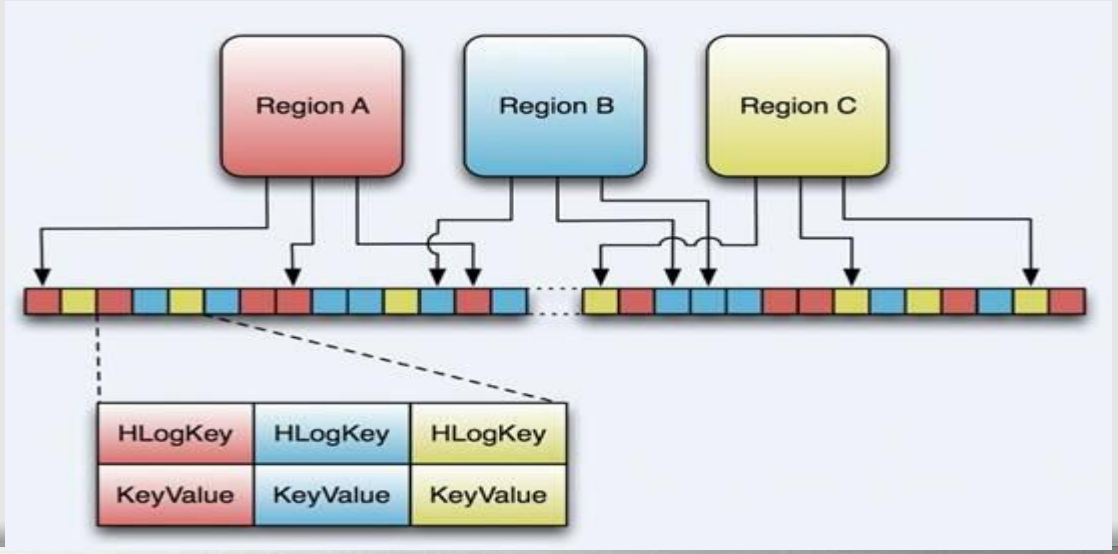
Hbase的原理

- Hfile
  - HFile文件是不定长的，长度固定的只有其中的两块：Trailer和FileInfo
  - Trailer中有指针指向其他数据块的起始点
  - File Info中记录了文件的一些Meta信息
  - Data Index和Meta Index块记录了每个Data块和Meta块的起始点





- Hlog
  - 在每次用户操作写入MemStore的同时，也会写一份数据到HLog文件中，HLog文件定期会滚动出新的，并删除旧的文件。当HRegionServer意外终止后，HMaster会通过Zookeeper感知到，HMaster首先会处理遗留的 HLog文件，将其中不同Region的Log数据进行拆分，分别放到相应region的目录下，然后再将失效的region重新分配
- HLogFile
  - HLog文件就是一个普通的Hadoop Sequence File



Hbase的特点

- 特点：
  - 大:一个表可以有上亿行，上百万列
  - 依靠横向扩展，通过不断增加廉价的商用服务器，来增加计算和存储能力。
  - 必须有主键(row key)
  - 仅能通过主键和主键的range来检索数据，不支持二级索引
  - 高实时性
  - 面向列:面向列(族)的存储和权限控制，列(族)独立检索。
  - 稀疏:对于为空(null)的列，并不占用存储空间，因此，表可以设计的非常稀疏。
  - 不支持join，不支持多行事务

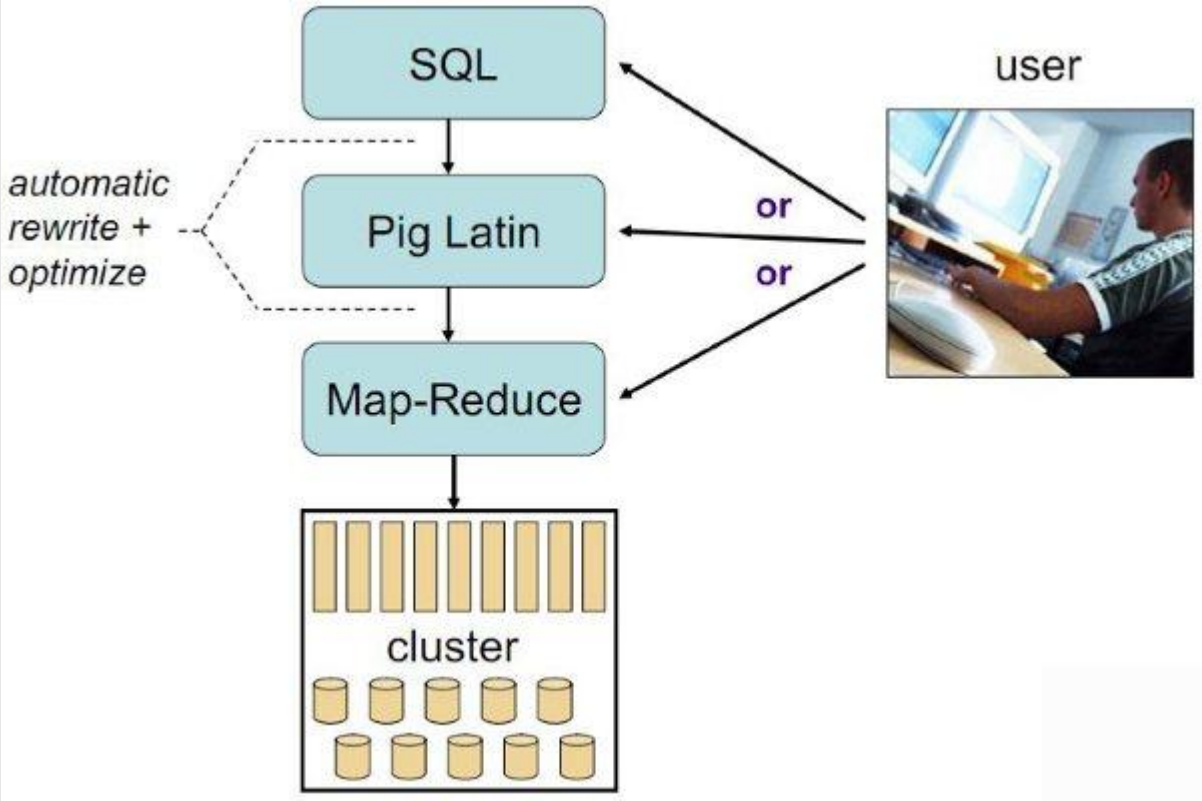
- HBase与HIVE的比较
  - 都是表结构
  - 基础数据都存放在HDFS上
  - Hbase有主索引，HIVE无
  - Hbase高实时性，HIVE非实时
  - Hbase可随时随意添加列，HIVE必须事先定义
  - Hbase支持修改删除，HIVE不支持
  - Hbase是列存储，HIVE不是
  - Hbase主键必须有，HIVE无要求
  - Hbase不能多表join，HIVE支持
  - Hbase开发较复杂，HIVE的HQL易学易用

- HBase与HIVE的结合
  - 在HIVE中生成数据在Hbase的表
    - CREATE TABLE hbase_table_1(key string, value string)
    - STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    - WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key, col_cluster_name:col_name ")
    - TBLPROPERTIES ("hbase.table.name" = "new_hbase_name");
  - 优点：
    - Hbase的表和HIVE的表数据完全相同，表名列名可各自定义
    - HIVE可以对应Hbase的全部列或某几列
    - 可以在Hbase中增加新列，重建HIVE的对应表，达到增加HIVE列的目的
    - 可以修改Hbase中的值从而修改了HIVE表中值
    - 可以通过HIVE对Hbase多表数据进行join
  - 不足：
    - HIVE目前只能读取Hbase数据的最新version

OTHER

- PIG
- OOZIE
- MAHOUT
- HUE
- FLUME

- PIG
  - Pig是对处理超大型数据集的抽象层，用Pig提供的类SQL语句可以简化MapReduce的开发。



- PIG界面

```
[shiliyuan@h252021 ~]$ pig
2012-02-28 22:34:18,177 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/shiliyuan/pig_1330439658171.log
2012-02-28 22:34:18,556 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://h252020:9000/
2012-02-28 22:34:18,882 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker at: hdfs://h252020:9001/
grunt>
```

- PIG与HIVE的区别
  - Hive中有 “表”的概念，但是Pig中基本没有表的概念，所谓的表建立在Pig Latin脚本中。
  - Pig可以执行一些 ls 、cat 这样很经典的命令，但是在使用Hive的时候没有。
  - Pig允许开发简洁的脚本用于转换数据流以便嵌入到较大的应用程序，无需事先定义表；HIVE操作HDFS数据必须先定义表并制定列和列对应的数据
  - Hive有JDBC/ODBC，Pig无
  - HQL比Pig Latin更像SQL

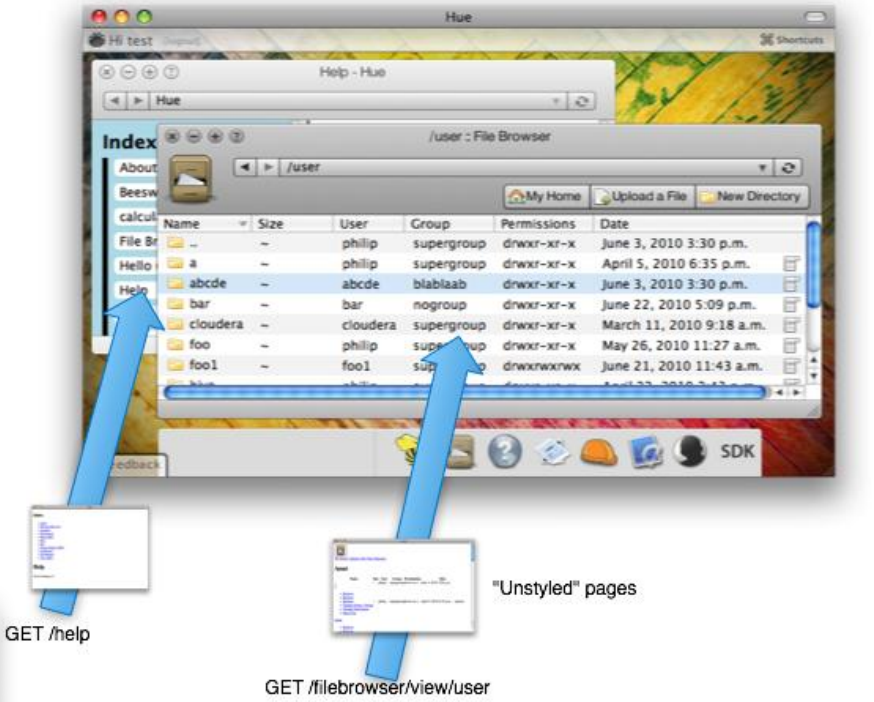
- OOZIE
  - Oozie是一个开源的工作流和协作服务引擎，基于 Apache Hadoop 的数据处理任务。Oozie 是可扩展的、可伸缩的面向数据的服务，运行在 Hadoop 平台上。

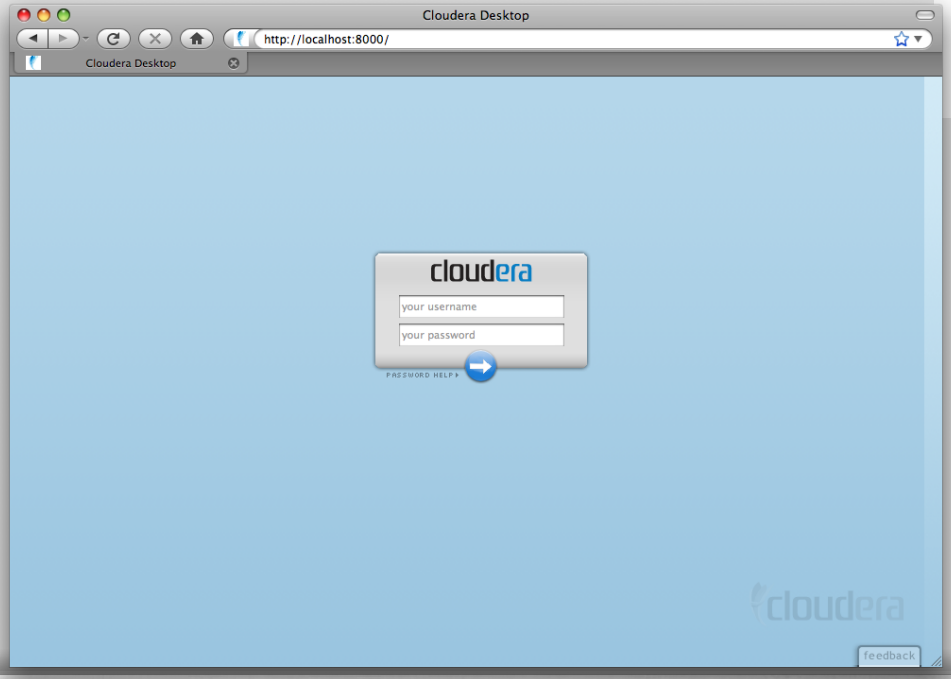
- MAHOUT
  - Mahout 是Apache Software Foundation (ASF)旗下的一个开源项目，提供一些可扩展的机器学习领域经典算法的实现，旨在帮助开发人员更加方便快捷地创建智能应用程序。

```
[hadoop@h253014 mahout]$ examples/bin/build-cluster-syntheticcontrol.sh
examples/bin/build-cluster-syntheticcontrol.sh: line 25: [: =: unary operator expected
Please select a number to choose the corresponding clustering algorithm
1. canopy clustering
2. kmeans clustering
3. fuzzykmeans clustering
4. dirichlet clustering
5. meanshift clustering
Enter your choice :
```

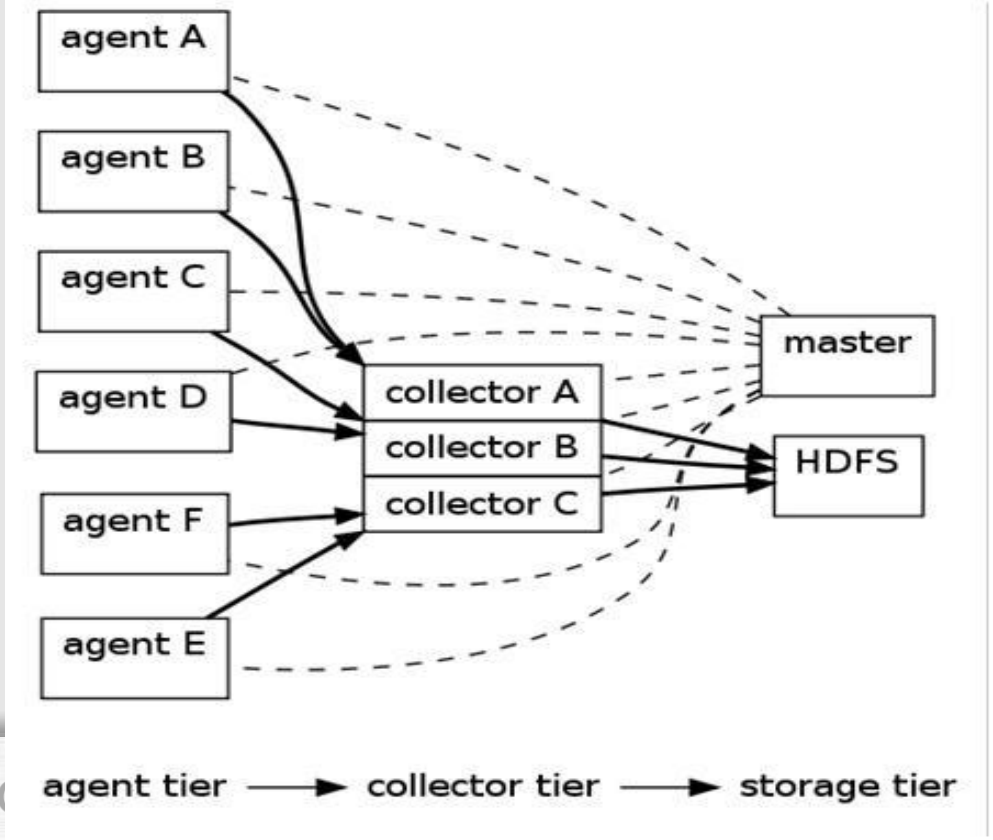
OTHER

- HUE
  - HDFS高级视图
  - 支持jar插件开发





- Flume
  - Flume是Cloudera提供的日志收集系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。
  - Flume是一个分布式、可靠、和高可用的海量日志采集、聚合和传输的系统。



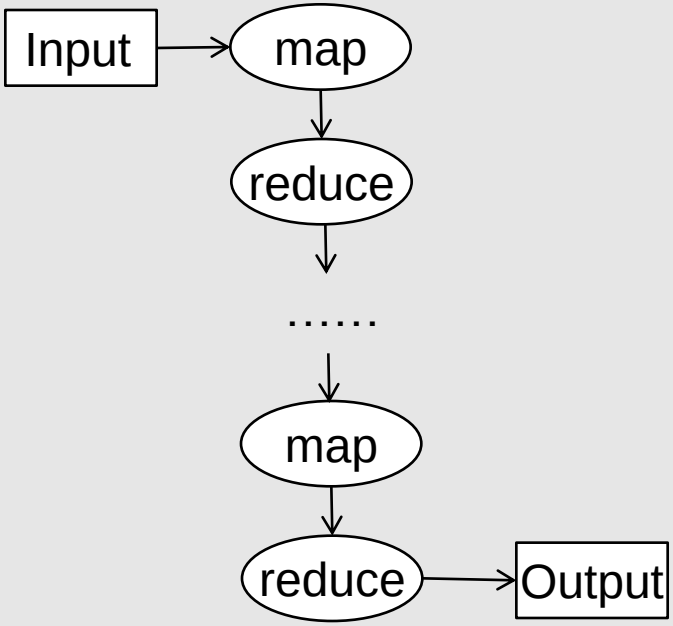
- Avro
  - Avro是一个数据序列化的系统，它可以提供：
  - 丰富的数据结构类型
  - 快速可压缩的二进制数据形式
  - 存储持久数据的文件容器
  - 远程过程调用RPC
  - 简单的动态语言结合功能，Avro和动态语言结合后，读写数据文件和使用RPC协议都不需要生成代码，而代码生成作为一种可选的优化只值得在静态类型语言中实现。
- Thrift
  - Apache thrift是一个比较流行的序列化框架，用来进行可扩展且跨语言的服务的开发。它结合了功能强大的软件堆栈和代码生成引擎，以构建在不同编程语言间无缝结合的、高效的服务。

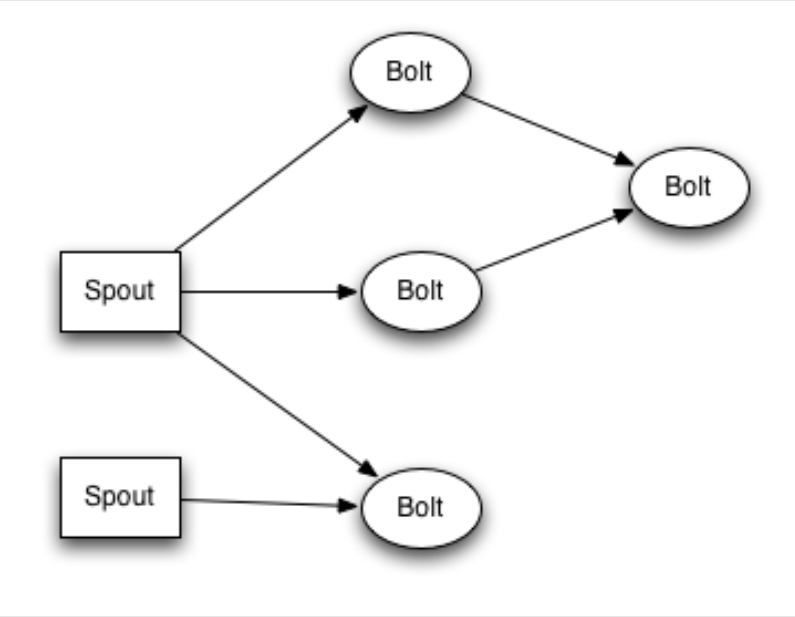
- STORM简介
- STORM与Hadoop
- STORM示例
- 较理想的数据处理架构

- Twitter Storm
  - Storm是一个分布式的、容错的实时计算系统（原来是由BackType开发，后BackType被Twitter收购，将Storm作为Twitter的实时数据分析）
  - Storm集群由一个主节点和多个工作节点组成。主节点运行了一个名为“Nimbus”的守护进程，用于分配代码、布置任务及故障检测。每个工作节点都运行了一个名为“Supervisor”的守护进程，用于监听工作，开始并终止工作进程。

- 特点：
  - 简单的编程模型。类似于MapReduce降低了并行批处理复杂性，Storm降低了进行实时处理的复杂性
  - 可以使用各种编程语言。
  - 容错性。Storm会管理工作进程和节点的故障。
  - 水平扩展。计算是在多个线程、进程和服务器之间并行进行的。
  - 可靠的消息处理。Storm保证每个消息至少能得到一次完整处理。
  - 快速。系统的设计保证了消息能得到快速的处理，使用ØMQ作为其底层消息队列。
  - 本地模式。Storm有一个“本地模式”，可以在处理过程中完全模拟Storm集群。

- Hadoop与Storm的处理方式
  - MapReduce与Topology



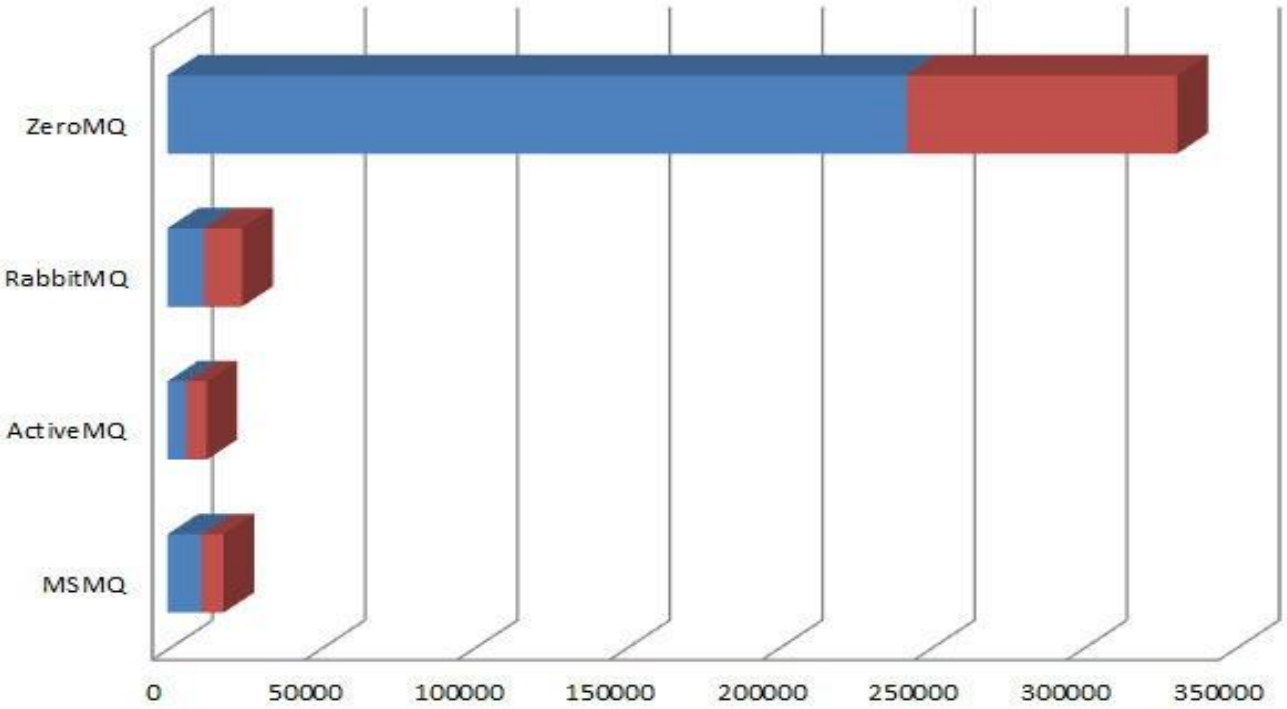


Storm与Hadoop

- 处理方式：图式处理与链式处理
- 数据存放：内存与磁盘
- 通讯机制：**ZeroMQ**与TCP/IP
- 响应时间：实时与非实时

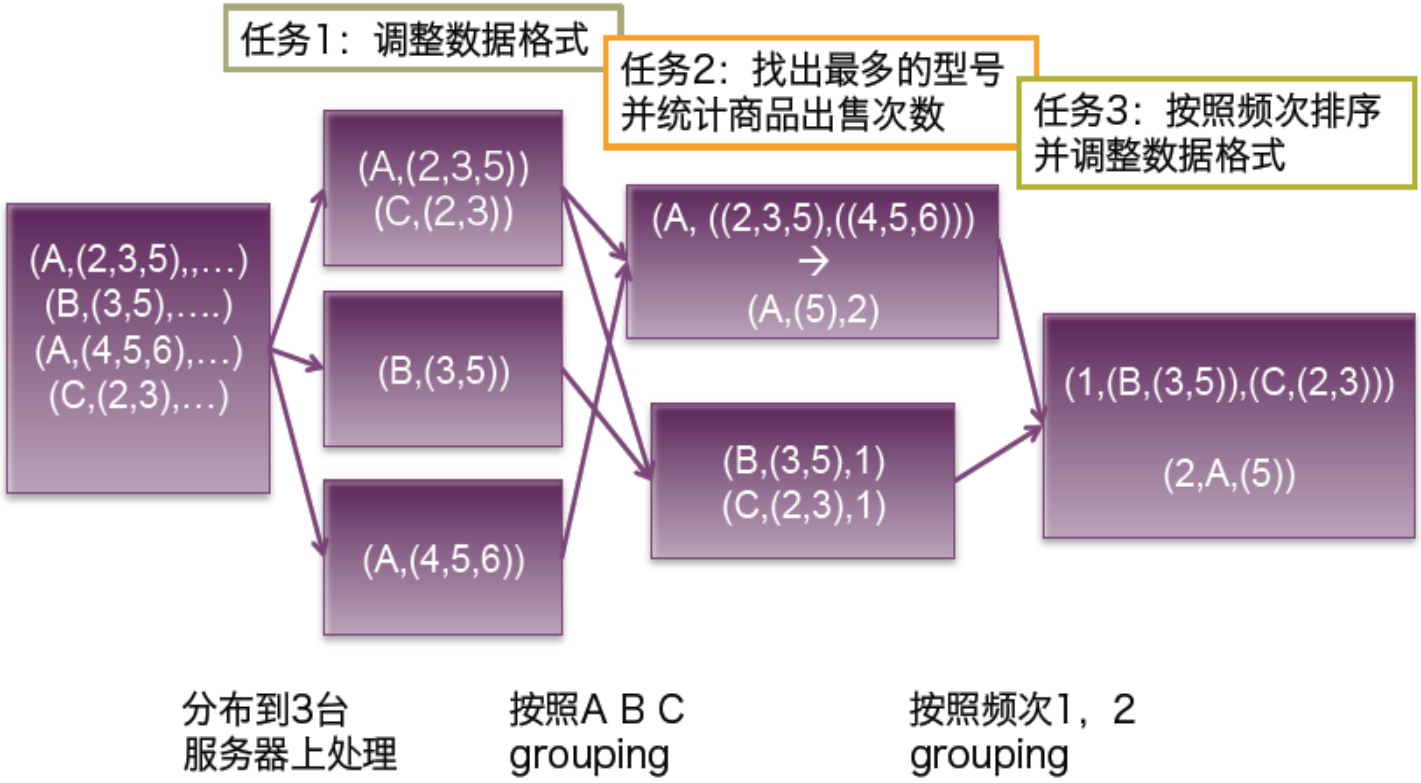
ZeroMQ

- **ZeroMQ**是一个轻量级消息内核。它实现了30微秒的端到端延迟和每秒超过300万的信息。它可用于C、C++、Python、.NET /Mono、Fortran和Java语言。

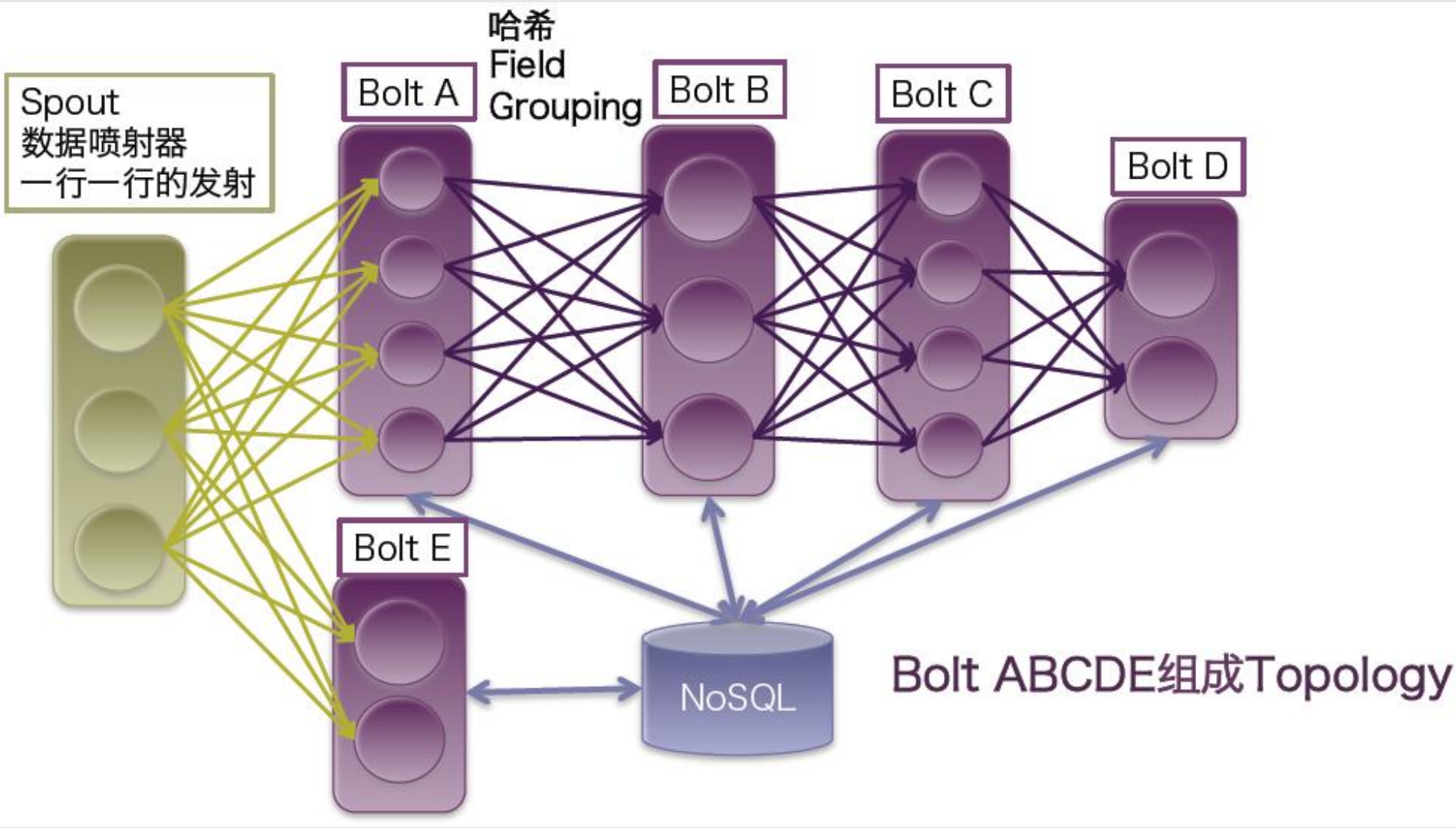


	MSMQ	ActiveMQ	RabbitMQ	ZeroMQ
Send	11406	6452	12278	242659
Receive	7021	6453	12281	88699

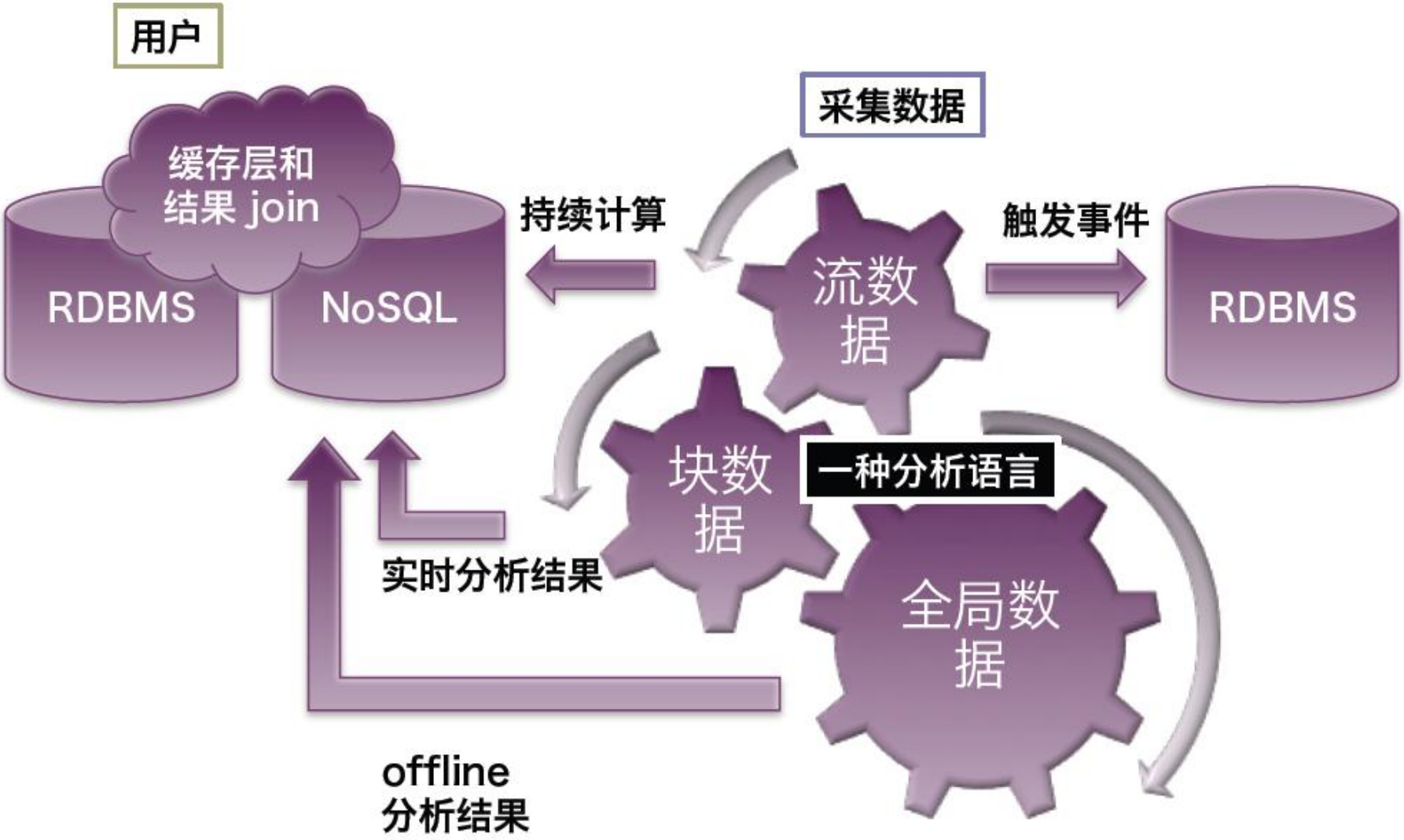
按照客户购买的频次1，2，3列出商品，同时每种商品列出最热卖的型号 ABC 商品 (2,3) 型号



STORM处理过程


Spout
数据喷射器
一行一行的发射
Bolt A
哈希
Field
Grouping
Bolt B
Bolt C
Bolt D
Bolt E
NoSQL
Bolt ABCDE组成Topology

较理想的数据处理架构


用户
缓存层和
结果 join
RDBMS
NoSQL
采集数据
持续计算
流数
据
触发事件
RDBMS
块数
据
一种分析语言
全局数
据
实时分析结果
offline
分析结果

Q&A