[**一共81个，开源大数据处理工具汇总（上）**](http://www.36dsj.com/archives/24852)

**作者：大数据女神-诺蓝（微信公号：dashujunvshen）。本文是36大数据专稿，转载必须标明来源36大数据。**

本文一共分为上下两部分。我们将针对大数据开源工具不同的用处来进行分类，并且附上了官网和部分下载链接，希望能给做大数据的朋友做个参考。下面是第一部分。

# ****查询引擎****

**一、Phoenix**

贡献者：：[Salesforce](http://www.36dsj.com/archives/tag/salesforce)

简介：这是一个Java中间层，可以让开发者在Apache HBase上执行SQL查询。Phoenix完全使用Java编写，代码位于[GitHub](https://github.com/forcedotcom/phoenix)上，并且提供了一个客户端可嵌入的JDBC驱动。

Phoenix查询引擎会将SQL查询转换为一个或多个HBase scan，并编排执行以生成标准的JDBC结果集。直接使用HBase API、协同处理器与自定义过滤器，对于简单查询来说，其性能量级是毫秒，对于百万级别的行数来说，其性能量级是秒。

Phoenix**最值得关注的一些特性有：**

❶嵌入式的JDBC驱动，实现了大部分的java.sql接口，包括元数据API  
❷可以通过多部行键或是键/值单元对列进行建模  
❸完善的查询支持，可以使用多个谓词以及优化的扫描键  
❹DDL支持：通过CREATE TABLE、DROP TABLE及ALTER TABLE来添加/删除列  
❺版本化的模式仓库：当写入数据时，快照查询会使用恰当的模式  
❻DML支持：用于逐行插入的UPSERT VALUES、用于相同或不同表之间大量数据传输的UPSERT ❼SELECT、用于删除行的DELETE  
❽通过客户端的批处理实现的有限的事务支持  
❾单表——还没有连接，同时二级索引也在开发当中  
➓紧跟ANSI SQL标准

[Phoenix官方网站>>>](http://phoenix.incubator.apache.org/)

**二、Stinger**

贡献者：：[Hortonworks](http://www.36dsj.com/archives/tag/hortonworks)

简介：原叫Tez，下一代Hive,Hortonworks主导开发，运行在YARN上的DAG计算框架。

某些测试下，Stinger能提升10倍左右的性能，同时会让Hive支持更多的SQL，**其主要优点包括：**

❶让用户在Hadoop获得更多的查询匹配。其中包括类似OVER的字句分析功能，支持WHERE查询，让Hive的样式系统更符合SQL模型。

❷优化了Hive请求执行计划，优化后请求时间减少90%。改动了Hive执行引擎，增加单Hive任务的被秒处理记录数。

❸在Hive社区中引入了新的列式文件格式（如ORC文件），提供一种更现代、高效和高性能的方式来储存Hive数据。

❹引入了新的运行时框架——Tez，旨在消除Hive的延时和吞吐量限制。Tez通过消除不必要的task、障碍同步和对HDFS的读写作业来优化Hive job。这将优化Hadoop内部的执行链，彻底加速Hive负载处理。

[Stinger官方网站>>>](http://hortonworks.com/labs/stinger/)

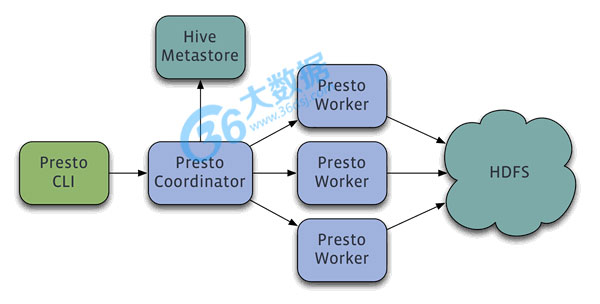
**三、Presto**

贡献者：：Facebook

简介：Facebook开源的数据查询引擎Presto ，可对250PB以上的数据进行快速地交互式分析。该项目始于 2012 年秋季开始开发，目前该项目已经在超过 1000 名 Facebook 雇员中使用，运行超过 30000 个查询，每日数据在 1PB 级别。Facebook 称 Presto 的性能比诸如 Hive 和 Map\*Reduce 要好上 10 倍有多。

Presto 当前支持 ANSI SQL 的大多数特效，包括联合查询、左右联接、子查询以及一些聚合和计算函数；支持近似截然不同的计数(DISTINCT COUNT)等。

[github源代码下载>>>](https://github.com/facebook/presto)



**四、Shark**

**简介**：Shark即Hive on Spark，本质上是通过Hive的HQL解析，把HQL翻译成Spark上的RDD操作，然后通过Hive的metadata获取数据库里的表信息，实际HDFS上的数据和文件，会由Shark获取并放到Spark上运算。Shark的特点就是快，完全兼容Hive，且可以在shell模式下使用rdd2sql()这样的API，把HQL得到的结果集，继续在scala环境下运算，支持自己编写简单的机器学习或简单分析处理函数，对HQL结果进一步分析计算。

❶Shark速度快的原因除了Spark平台提供的基于内存迭代计算外，在设计上还存在对Spark上进行了一定的改造，主要有

❷partial DAG execution：对join优化，调节并行粒度，因为Spark本身的宽依赖和窄依赖会影响并行计算和速度

基于列的压缩和存储：把HQL表数据按列存，每列是一个array，存在JVM上，避免了JVM GC低效，而压缩和解压相关的技术是Yahoo!提供的。

结来说，Shark是一个插件式的东西，在我现有的Spark和Hive及hadoop-client之间，在这两套都可用的情况下，Shark只要获取Hive的配置（还有metastore和exec等关键包），Spark的路径，Shark就能利用Hive和Spark，把HQL解析成RDD的转换，把数据取到Spark上运算和分析。在SQL on Hadoop这块，Shark有别于Impala，Stringer，而这些系统各有自己的设计思路，相对于对MR进行优化和改进的思路，Shark的思路更加简单明了些。

[**Shark官方网站>>>**](http://shark.cs.berkeley.edu/)

**五、Pig**

**简介：**Pig是一种编程语言，它简化了Hadoop常见的工作任务。Pig可加载数据、表达转换数据以及存储最终结果。Pig内置的操作使得半结构化数据变得有意义（如日志文件）。同时Pig可扩展使用Java中添加的自定义数据类型并支持数据转换。

Pig最大的作用就是对mapreduce算法(框架)实现了一套shell脚本 ，类似我们通常熟悉的SQL语句，在Pig中称之为Pig Latin，在这套脚本中我们可以对加载出来的数据进行排序、过滤、求和、分组(group by)、关联(Joining)，Pig也可以由用户自定义一些函数对数据集进行操作，也就是传说中的UDF(user-defined functions)。

[Pig官方网站>>>](http://pig.apache.org/)

**六、Cloudera Impala**

**贡献者：:**[Cloudera](http://www.36dsj.com/archives/tag/cloudera)

**简介：**Cloudera Impala 可以直接为存储在HDFS或HBase中的Hadoop数据提供快速，交互式的SQL查询。除了使用相同的存储平台外， Impala和Apache Hive一样也使用了相同的元数据，SQL语法（Hive SQL），ODBC驱动和用户接口（Hue Beeswax），这就很方便的为用户提供了一个相似并且统一的平台来进行批量或实时查询。

Cloudera Impala 是用来进行大数据查询的补充工具。 Impala 并没有取代像Hive这样基于MapReduce的分布式处理框架。Hive和其它基于MapReduce的计算框架非常适合长时间运行的批处理作业，例如那些涉及到批量 Extract、Transform、Load ，即需要进行ETL作业。

**Impala 提供了**：

❶数据科学家或数据分析师已经熟知的SQL接口

❷能够在Apache Hadoop 的大数据中进行交互式数据查询

❸ Single system for big data processing and analytics so customers can avoid costly modeling and ETL just for analytics

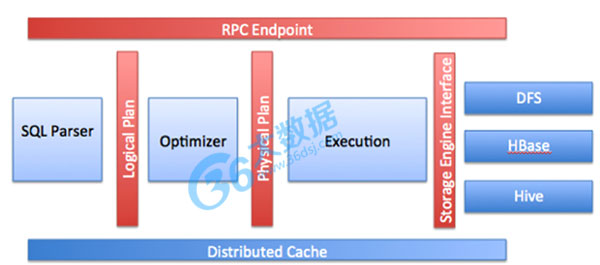
[Cloudera Impala官方网站>>>](http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html)

**七、Apache Drill**

贡献者：：[MapR](http://www.36dsj.com/archives/tag/mapr)

简介：Apache Drill是是一个能够对大数据进行交互分析、开源的分布式系统，且基于Google Dremel实现，它能够运行在上千个节点的服务器集群上，且能在几秒内处理PB级或者万亿条的数据记录。Drill能够帮助企业用户快速、高效地进行Hadoop数据查询和企业级大数据分析。Drill于2012年8月份由Apache推出。

从Drill官方对其架构的介绍中得知，其具有适于实时的分析和快速的应用开发、适于半结构化/嵌套数据的分析、兼容现有的SQL环境和Apache Hive等特征。另外，Drill的核心模块是Drillbit服务，该服务模块包括远程访问子模块、SQL解析器、查询优化器、任务计划执行引擎、存储插件接口（DFS、HBase、Hive等的接口）、分布式缓存模块等几部分，如下图所示：



[Apache Drill官方网站>>>](http://incubator.apache.org/drill/)

**八、Apache Tajo**

**简介：**Apache Tajo项目的目的是在HDFS之上构建一个先进的数据仓库系统。Tajo将自己标榜为一个“大数据仓库”，但是它好像和之前介绍的那些低延迟查询引擎类似。虽然它支持外部表和Hive数据集（通过HCatalog），但是它的重点是数据管理，提供低延迟的数据访问，以及为更传统的ETL提供工具。它也需要在数据节点上部署Tajo特定的工作进程。

Tajo的功能包括：

❶ANSI SQL兼容  
❷JDBC 驱动  
❸集成Hive metastore能够访问Hive数据集  
❹一个命令行客户端  
❺一个自定义函数API

[Apache Tajo官方网站>>>](http://tajo.incubator.apache.org/)

**九、Hive**

**简介：**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，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务进行运行。 其优点是学习成本低，可以通过类SQL语句快速实现简单的MapReduce统计，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。

[Hive官方网站>>>](http://hive.apache.org/)

# ****流式计算****

**一、Facebook Puma**

**贡献者：**Facebook

简介：实时数据流分析

**二、Twitter Rainbird**

**贡献者**：Twitter

**简介**：Rainbird一款基于Zookeeper, Cassandra, Scribe, Thrift的分布式实时统计系统，这些基础组件的基本功能如下：

❶ Zookeeper，Hadoop子项目中的一款分布式协调系统，用于控制分布式系统中各个组件中的一致性。

❷Cassandra，NoSQL中一款非常出色的产品，集合了Dynamo和Bigtable特性的分布式存储系统，用于存储需要进行统计的数据，统计数据，并且提供客户端进行统计数据的查询。（需要使用分布式Counter补丁CASSANDRA-1072）

❸ Scribe，Facebook开源的一款分布式日志收集系统，用于在系统中将各个需要统计的数据源收集到Cassandra中。

❹ Thrift，Facebook开源的一款跨语言C/S网络通信框架，开发人员基于这个框架可以轻易地开发C/S应用。

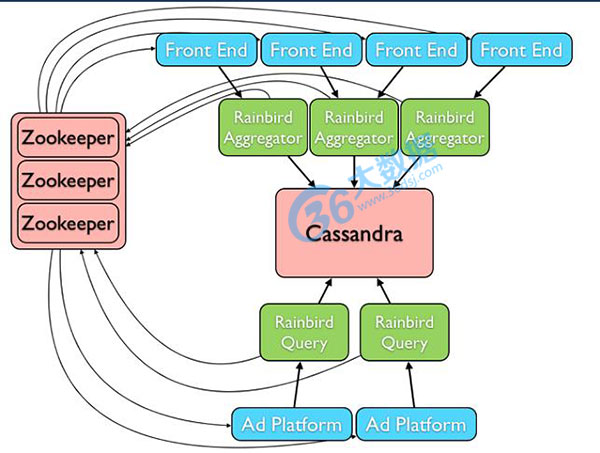
**用处**

Rainbird可以用于实时数据的统计：

❶统计网站中每一个页面，域名的点击次数

❷内部系统的运行监控（统计被监控服务器的运行状态）

❸记录最大值和最小值



**三、Yahoo S4**

**贡献者：**Yahoo

**简介**：S4（Simple Scalable Streaming System）最初是Yahoo!为提高搜索广告有效点击率的问题而开发的一个平台，通过统计分析用户对广告的点击率，排除相关度低的广告，提升点击率。目前该项目刚启动不久，所以也可以理解为是他们提出的一个分布式流计算（Distributed Stream Computing）的模型。

**S4的设计目标是：**

·提供一种简单的编程接口来处理数据流

·设计一个可以在普通硬件之上可扩展的高可用集群。

·通过在每个处理节点使用本地内存，避免磁盘I/O瓶颈达到最小化延迟

·使用一个去中心的，对等架构；所有节点提供相同的功能和职责。没有担负特殊责任的中心节点。这大大简化了部署和维护。

·使用可插拔的架构，使设计尽可能的即通用又可定制化。

·友好的设计理念，易于编程，具有灵活的弹性

[Yahoo S4官方网站>>>](http://incubator.apache.org/s4/)

**四、Twitter Storm**

**贡献者：**Twitter

**简介：**Storm是Twitter开源的一个类似于Hadoop的实时数据处理框架，它原来是由BackType开发，后BackType被Twitter收购，将Storm作为Twitter的实时数据分析系统。

实时数据处理的应用场景很广泛，例如商品推荐，广告投放，它能根据当前情景上下文（用户偏好，地理位置，已发生的查询和点击等）来估计用户点击的可能性并实时做出调整。

**storm的三大作用领域：**

**1.信息流处理（Stream Processing）**

Storm可以用来实时处理新数据和更新数据库，兼具容错性和可扩展性,它 可以用来处理源源不断的消息，并将处理之后的结果保存到持久化介质中。

**2.连续计算（Continuous Computation）**

Storm可以进行连续查询并把结果即时反馈给客户，比如将Twitter上的热门话题发送到客户端。

**3.分布式远程过程调用（Distributed RPC）**

除此之外，Storm也被广泛用于以下方面：

* 精确的广告推送
* 实时日志的处理

[**Twitter Storm官方网站>>>**](http://storm.incubator.apache.org/)

# ****迭代计算****

**一、Apache Hama**

**简介：**Apache Hama是一个纯BSP（Bulk Synchronous Parallel）计算框架，模仿了Google的Pregel。用来处理大规模的科学计算，特别是矩阵和图计算。

❶建立在Hadoop上的分布式并行计算模型。

❷基于 Map/Reduce 和 Bulk Synchronous 的实现框架。

❸运行环境需要关联 Zookeeper、HBase、HDFS 组件。

**Hama中有2个主要的模型:**

– 矩阵计算(Matrix package)

– 面向图计算(Graph package)

[Apache Hama官方网站>>>](https://hama.apache.org/)

**二、Apache Giraph**

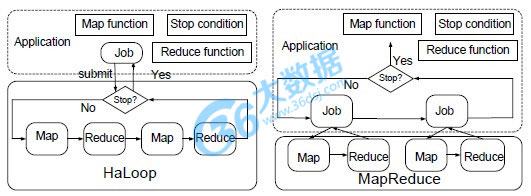
**代码托管地址**： [GitHub](https://github.com/apache/giraph)

**简介：**Apache Giraph是一个可伸缩的分布式迭代图处理系统，灵感来自BSP（bulk synchronous parallel）和Google的Pregel，与它们 区别于则是是开源、基于 Hadoop 的架构等。

Giraph处理平台适用于运行大规模的逻辑计算，比如页面排行、共享链接、基于个性化排行等。Giraph专注于社交图计算，被Facebook作为其Open Graph工具的核心，几分钟内处理数万亿次用户及其行为之间的连接。

**三、HaLoop**

**简介**：迭代的MapReduce，HaLoop——适用于迭代计算的Hadoop 。



Hadoop与HaLoop的不同

与Hadoop比较的四点改变：

1.提供了一套新的编程接口，更加适用于迭代计算；

HaLoop给迭代计算一个抽象的递归公式：

大数据  
2.HaLoop的master进行job内的循环控制，直到迭代计算结束；

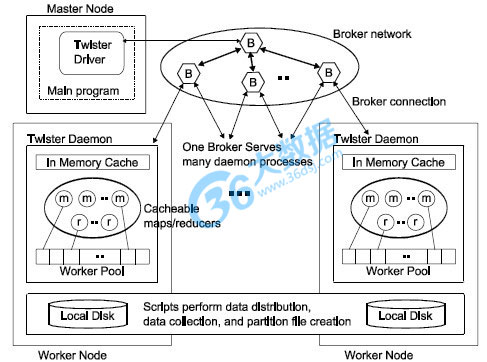
3.Task Scheduler也进行了修改，使得任务能够尽量满足data locality

4.slave nodes对数据进行cache并index索引，索引也以文件的形式保存在本地磁盘。

[HaLoop官网>>>](https://code.google.com/p/haloop/)

**四、Twister**

简介：Twister， 迭代式MapReduce框架，Twister是由一个印度人开发的，其架构如下：



在Twister中，大文件不会自动被切割成一个一个block，因而用户需提前把文件分成一个一个小文件，以供每个task处理。在map阶段，经过map（）处理完的结果被放在分布式内存中，然后通过一个broker network（NaradaBroking系统）将数据push给各个reduce task（Twister假设内存足够大，中间数据可以全部放在内存中）；在reduce阶段，所有reduce task产生的结果通过一个combine操作进行归并，此时，用户可以进行条件判定， 确定迭代是否结束。combine后的数据直接被送给map task，开始新一轮的迭代。为了提高容错性，Twister每隔一段时间会将map task和reduce task产生的结果写到磁盘上，这样，一旦某个task失败，它可以从最近的备份中获取输入，重新计算。

为了避免每次迭代重新创建task，Twister维护了一个task pool，每次需要task时直接从pool中取。在Twister中，所有消息和数据都是通过broker network传递的，该broker network是一个独立的模块，目前支持NaradaBroking和ActiveMQ。

# ****离线计算****

**一、Hadoop MapReduce**

**简介**：MapReduce是一种编程模型，用于大规模数据集（大于1TB）的并行运算。概念”Map（映射）”和”Reduce（归约）”，和它们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。 当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（归约）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

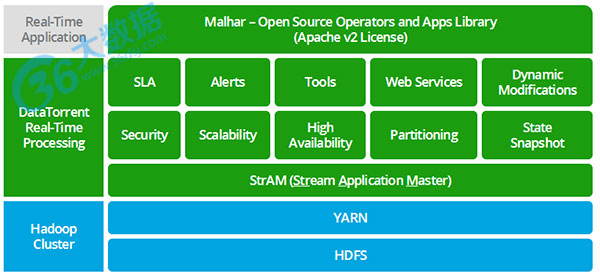
[**Hadoop MapReduce官方网站>>>**](http://hadoop.apache.org/)

**二、Berkeley Spark**

**简介**：Spark是UC Berkeley AMP lab所开源的类Hadoop MapReduce的通用的并行，Spark，拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是Job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的map reduce的算法。

**三、DataTorrent**

**简介**：DataTorrent基于Hadoop 2.x构建，是一个实时的、有容错能力的数据流式处理和分析平台，它使用本地Hadoop应用程序，而这些应用程序可以与执行其它任务，如批处理，的应用程序共存。该平台的架构如下图所示：



相关文章：[DataTorrent 1.0每秒处理超过10亿个实时事件](http://www.36dsj.com/archives/20163)

[DataTorrent 将数据分析速度从“实时”提升至“现在时”](http://www.36dsj.com/archives/1596)

# ****键值存储****

**一、LevelDB**



贡献者：Google

简介：Leveldb是一个google实现的非常高效的kv数据库，目前的版本1.2能够支持billion级别的数据量了。 在这个数量级别下还有着非常高的性能，主要归功于它的良好的设计。特别是LMS算法。

LevelDB 是单进程的服务，性能非常之高，在一台4核Q6600的CPU机器上，每秒钟写数据超过40w，而随机读的性能每秒钟超过10w。

此处随机读是完全命中内存的速度，如果是不命中 速度大大下降。

[LevelDB官方网站>>>](https://code.google.com/p/leveldb/)

**二、RocksDB**

**贡献者**：facebook

**简介**：RocksDB虽然在代码层面上是在LevelDB原有的代码上进行开发的，但却借鉴了Apache HBase的一些好的idea。在云计算横行的年代，开口不离Hadoop，RocksDB也开始支持HDFS，允许从HDFS读取数据。RocksDB支持一次获取多个K-V，还支持Key范围查找。LevelDB只能获取单个Key。

RocksDB除了简单的Put、Delete操作，还提供了一个Merge操作，说是为了对多个Put操作进行合并。

RocksDB提供一些方便的工具，这些工具包含解析sst文件中的K-V记录、解析MANIFEST文件的内容等。RocksDB支持多线程合并，而LevelDB是单线程合并的。

[RocksDB官方网站>>>](http://rocksdb.org/)

**三、HyperDex**

贡献者：Facebook



HyperDex是一个分布式、可搜索的键值存储系统，特性如下：

* 分布式KV存储，系统性能能够随节点数目线性扩展
* 吞吐和延时都能秒杀现在风头正劲的MonogDB，吞吐甚至强于Redis
* 使用了hyperspace hashing技术，使得对存储的K-V的任意属性进行查询成为可能

官网：<http://hyperdex.org/>

**四、TokyoCabinet**

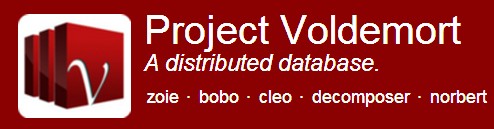


日本人Mikio Hirabayashi（平林干雄）开发的一款DBM数据库。Tokyo Cabinet 是一个DBM的实现。这里的数据库由一系列key-value对的记录构成。key和value都可以是任意长度的字节序列,既可以是二进制也可以是字符串。这里没有数据类型和数据表的概念。

当 做为Hash表数据库使用时，每个key必须是不同的,因此无法存储两个key相同的值。提供了以下访问方法:提供key,value参数来存储，按 key删除记录，按key来读取记录，另外，遍历key也被支持，虽然顺序是任意的不能被保证。这些方法跟Unix标准的DBM,例如GDBM,NDBM 等等是相同的，但是比它们的性能要好得多（因此可以替代它们) 。下一代KV存储系统，支持strings、integers、floats、lists、maps和sets等丰富的数据类型。

[TokyoCabinet官方网站>>>](http://fallabs.com/tokyocabinet/)

**五、Voldemort**



Voldemort是一个分布式键值存储系统，是Amazon’s Dynamo的一个开源克隆。特性如下：

* 支持自动复制数据到多个服务器上。
* 支持数据自动分割所以每个服务器只包含总数据的一个子集。
* 提供服务器故障透明处理功能。
* 支持可拨插的序化支持，以实现复杂的键-值存储，它能够很好的5.集成常用的序化框架如：Protocol Buffers、Thrift、Avro和Java Serialization。
* 数据项都被标识版本能够在发生故障时尽量保持数据的完整性而不会影响系统的可用性。
* 每个节点相互独立，互不影响。
* 支持可插拔的数据放置策略

官网：<http://project-voldemort.com/>

**六、Amazon Dynamo**

贡献者：亚马逊

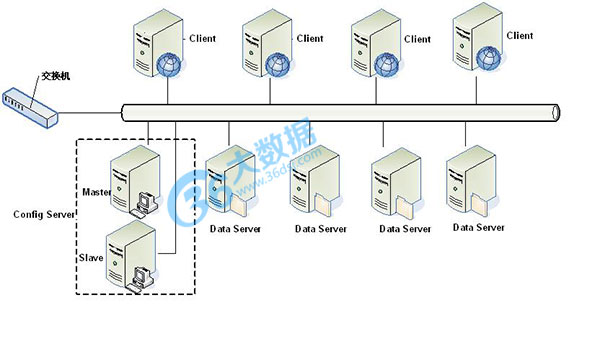
简介：Amazon Dynamo 是一个经典的分布式Key-Value 存储系统，具备去中心化，高可用性，高扩展性的特点，但是为了达到这个目标在很多场景中牺牲了一致性。Dynamo在Amazon中得到了成功的应用，能够跨数据中心部署于上万个结点上提供服务，它的设计思想也被后续的许多分布式系统借鉴。如近来火热的Cassandra，实际上就是基本照搬了Dynamo的P2P架构，同时融合了BigTable的数据模型及存储算法。

[Amazon Dynamo官方网站>>>](https://github.com/dynamo/dynamo)

**七、Tair**

**贡献者：**淘宝

**简介**：tair 是淘宝自己开发的一个分布式 key/value 存储引擎. tair 分为持久化和非持久化两种使用方式. 非持久化的 tair 可以看成是一个分布式缓存. 持久化的 tair 将数据存放于磁盘中. 为了解决磁盘损坏导致数据丢失, tair 可以配置数据的备份数目, tair 自动将一份数据的不同备份放到不同的主机上, 当有主机发生异常, 无法正常提供服务的时候, 其于的备份会继续提供服务.**tair 的总体结构**

tair 作为一个分布式系统, 是由一个中心控制节点和一系列的服务节点组成. 我们称中心控制节点为config server. 服务节点是data server. config server 负责管理所有的data server, 维护data server的状态信息. data server 对外提供各种数据服务, 并以心跳的形式将自身状况汇报给config server. config server是控制点, 而且是单点, 目前采用一主一备的形式来保证其可靠性. 所有的 data server 地位都是等价的.

**八、Apache Accumulo**



Apache Accumulo 是一个可靠的、可伸缩的、高性能的排序分布式的 Key-Value 存储解决方案，基于单元访问控制以及可定制的服务器端处理。Accumulo使用 Google BigTable 设计思路，基于 Apache Hadoop、Zookeeper 和 Thrift 构建。

**官网**：<http://accumulo.apache.org/>

**九、Redis**



Redis是一个高性能的key-value存储系统，和Memcached类似，它支持存储的value类型相对更多，包括string（字符串）、list（链表）、set（集合）和zset（有序集合）。与memcached一样，为了保证效率，数据都是缓存在内存中，区别的是Redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了主从同步。

Redis的出现，很大程度补偿了memcached这类key/value存储的不足，在部分场合可以对关系数据库起到很好的补充作用。它提供了Python、Ruby、Erlang、PHP客户端，使用很方便。

**官网**：<http://redis.io/>

# ****表格存储****

**一、OceanBase**

**贡献者**：阿里巴巴

**相关文章**：[26页PPT解密支撑支付宝交易的分布式数据库系统——OceanBase](http://www.36dsj.com/archives/20317)

**简介**：OceanBase是一个支持海量数据的高性能分布式数据库系统，实现了数千亿条记录、数百TB数据上的跨行跨表事务，由淘宝核心系统研发部、运维、DBA、广告、应用研发等部门共同完成。在设计和实现OceanBase的时候暂时摒弃了不紧急的DBMS的功能，例如临时表，视图(view)，研发团队把有限的资源集中到关键点上，当前 OceanBase主要解决数据更新一致性、高性能的跨表读事务、范围查询、join、数据全量及增量dump、批量数据导入。

目前OceanBase已经应用于淘宝收藏夹，用于存储淘宝用户收藏条目和具体的商品、店铺信息，每天支持4～5千万的更新操作。等待上线的应用还包括CTU、SNS等，每天更新超过20亿，更新数据量超过2.5TB，并会逐步在淘宝内部推广。

OceanBase 0.3.1在Github开源，开源版本为Revision:12336。

**官网**：<http://alibaba.github.io/oceanbase/>

**二、Amazon SimpleDB**

**贡献者**：亚马逊

Amazon SimpleDB是一个分散式数据库，以Erlang撰写。同与Amazon EC2和亚马逊的S3一样作为一项Web 服务，属于亚马逊网络服务的一部分。



正如EC2和S3，SimpleDB的按照存储量，在互联网上的传输量和吞吐量收取费用。 在2008年12月1日，亚马逊推出了新的定价策略，提供了免费1 GB的数据和25机器小时的自由层(Free Tire)。 将其中的数据转移到其他亚马逊网络服务是免费的。

它是一个可大规模伸缩、用 Erlang 编写的高可用数据存储。

**官网**：<http://aws.amazon.com/cn/simpledb/>

**三、Vertica**

**贡献者：**惠普

**简介：**惠普2011年2月份起始3月21号完成收购Vertica。Vertica基于列存储。基于列存储的设计相比传统面向行存储的数据库具有巨大的优势。同时Vertica支持MPP（massively parallel processing）等技术，查询数据时Vertica只需取得需要的列，而不是被选择行的所有数据，其平均性能可提高50x-1000x倍。（查询性能高速度快）

Vertica的设计者多次表示他们的产品围绕着高性能和高可用性设计。由于对MPP技术的支持，可提供对粒度，可伸缩性和可用性的优势。每个节点完全独立运作，完全无共享架构，降低对共享资源的系统竞争。

Vertica的数据库使用标准的SQL查询，同时Vertica的架构非常适合云计算，包括虚拟化，分布式多节点运行等，并且可以和Hadoop/MapReduce进行集成。

**Vertica官网**：<http://www.vertica.com/>

**四、Cassandra**

**贡献者**：facebook

**相关文章**：[开源分布式NoSQL数据库系统——Cassandra](http://www.36dsj.com/archives/20079)   [Cassandra与HBase的大数据对决 谁是胜者？](http://www.36dsj.com/archives/7179)

**简介**：Cassandra是一套开源分布式NoSQL数据库系统。它最初由Facebook开发，用于储存收件箱等简单格式数据，集GoogleBigTable的数据模型与Amazon Dynamo的完全分布式的架构于一身Facebook于2008将 Cassandra 开源，此后，由于Cassandra良好的可扩放性，被Digg、Twitter等知名Web 2.0网站所采纳，成为了一种流行的分布式结构化数据存储方案。



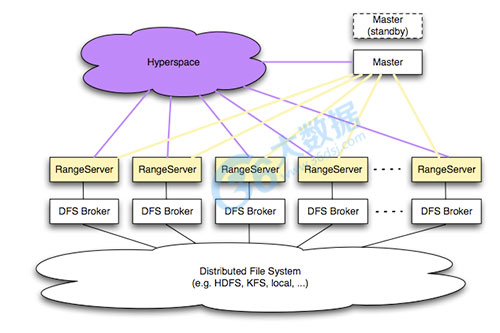
Cassandra是一个混合型的非关系的数据库，类似于Google的BigTable。其主要功能比Dynamo （分布式的Key-Value存储系统）更丰富，但支持度却不如文档存储MongoDB（介于关系数据库和非关系数据库之间的开源产品，是非关系数据库当中功能最丰富，最像关系数据库的。支持的数据结构非常松散，是类似json的bjson格式，因此可以存储比较复杂的数据类型）。Cassandra最初由Facebook开发，后转变成了开源项目。它是一个网络社交云计算方面理想的数据库。以Amazon专有的完全分布式的Dynamo为基础，结合了Google BigTable基于列族（Column Family）的数据模型。P2P去中心化的存储。很多方面都可以称之为Dynamo 2.0。

**Cassandra官网**：<http://cassandra.apache.org/>

**五、HyperTable**



简介：Hypertable是一个开源、高性能、可伸缩的数据库，它采用与Google的Bigtable相似的模型。在过去数年中，Google为在PC集群 上运行的可伸缩计算基础设施设计建造了三个关键部分。



第一个关键的基础设施是Google File System（GFS），这是一个高可用的文件系统，提供了一个全局的命名空间。它通过跨机器（和跨机架）的文件数据复制来达到高可用性，并因此免受传统 文件存储系统无法避免的许多失败的影响，比如电源、内存和网络端口等失败。第二个基础设施是名为Map-Reduce的计算框架，它与GFS紧密协作，帮 助处理收集到的海量数据。第三个基础设施是Bigtable，它是传统数据库的替代。Bigtable让你可以通过一些主键来组织海量数据，并实现高效的 查询。Hypertable是Bigtable的一个开源实现，并且根据我们的想法进行了一些改进。

**HyperTable官网**：<http://hypertable.org/>

**六、FoundationDB**

**简介：**支持ACID事务处理的NoSQL数据库，提供非常好的性能、数据一致性和操作弹性。

2015年1月2日，FoundationDB已经发布了其key-value数据库的3.0版本，主要专注于可伸缩性和性能上的改善。FoundationDB的CEO David Rosenthal在一篇博客上宣布了新的版本，其中展示了FoundationDB 3.0在可伸缩性方面的数据，它可以在一个32位的c3.8xlarge EC2实例上每秒写入1440万次；这在性能上是之前版本的36倍。

除了性能和可伸缩性的改善之外，FoundationDB 3.0还包含了对监控支持的改善。这种监控机制不仅仅是简单的机器检查，它添加了对多种潜在的硬件瓶颈的诊断，并且把那些高层级的信息整合到现有监控基础架构中。

**官网**：<https://foundationdb.com/>

**七：HBase**

**贡献者**： Fay Chang 所撰写的“Bigtable



**简介**：HBase是一个分布式的、面向列的开源数据库，该技术来源于 Fay Chang 所撰写的Google论文“Bigtable：一个结构化数据的分布式存储系统”。就像Bigtable利用了Google文件系统（File System）所提供的分布式数据存储一样，HBase在Hadoop之上提供了类似于Bigtable的能力。HBase是Apache的Hadoop项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。

**官网**：<http://hbase.apache.org/>

# ****文件存储****

**一、CouchDB**

简介：CouchDB是用Erlang开发的面向文档的数据库系统，最近刚刚发布了1.0版本（2010年7月14日）。CouchDB不是一个传统的关系数据库，而是面向文档的数据库，其数据存储方式有点类似lucene的index文件格式，CouchDB最大的意义在于它是一个面向web应用的新一代存储系统，事实上，CouchDB的口号就是：下一代的Web应用存储系统。



**特点：**

一、CouchDB是分布式的数据库，他可以把存储系统分布到n台物理的节点上面，并且很好的协调和同步节点之间的数据读写一致性。这当然也得靠Erlang无与伦比的并发特性才能做到。对于基于web的大规模应用文档应用，分布式可以让它不必像传统的关系数据库那样分库拆表，在应用代码层进行大量的改动。

二、CouchDB是面向文档的数据库，存储半结构化的数据，比较类似lucene的index结构，特别适合存储文档，因此很适合CMS，电话本，地址本等应用，在这些应用场合，文档数据库要比关系数据库更加方便，性能更好。

三、CouchDB支持REST API，可以让用户使用JavaScript来操作CouchDB数据库，也可以用JavaScript编写查询语句，我们可以想像一下，用AJAX技术结合CouchDB开发出来的CMS系统会是多么的简单和方便。

其实CouchDB只是Erlang应用的冰山一角，在最近几年，基于Erlang的应用也得到的蓬勃的发展，特别是在基于web的大规模，分布式应用领域，几乎都是Erlang的优势项目。

**官网**：<http://couchdb.apache.org/>

**二、MongoDB**

**简介**：MongoDB 是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

MongoDB是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似json的bson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

**相关文章**：[MongoDB的基本特性与内部构造](http://www.36dsj.com/archives/22965)  [大数据吃香 创业公司MongoDB估值达16亿美元](http://www.36dsj.com/archives/21104)



**特点**

它的特点是高性能、易部署、易使用，存储数据非常方便。主要功能特性有：

\*面向集合存储，易存储对象类型的数据。

mongodb集群参考

mongodb集群参考

\*模式自由。

\*支持动态查询。

\*支持完全索引，包含内部对象。

\*支持查询。

\*支持复制和故障恢复。

\*使用高效的二进制数据存储，包括大型对象（如视频等）。

\*自动处理碎片，以支持云计算层次的扩展性。

\*支持RUBY，PYTHON，JAVA，C++，PHP，C#等多种语言。

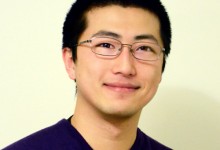
\*文件存储格式为BSON（一种JSON的扩展）。

\*可通过网络访问。

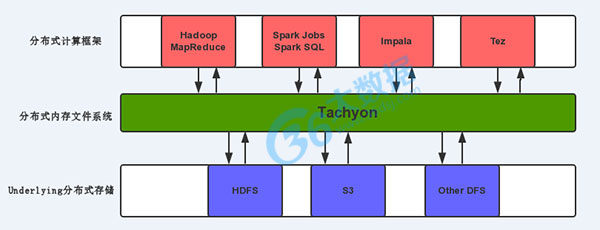
**官网**：<https://www.mongodb.org/>

**三、Tachyon**

**贡献者**：Haoyuan Li（李浩源）



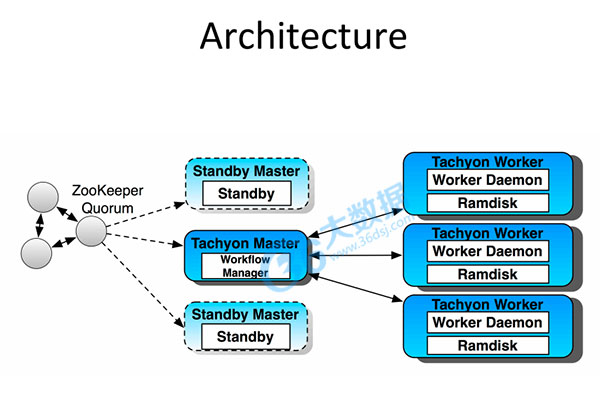
**简介**：Tachyon是一个分布式内存文件系统，可以在集群里以访问内存的速度来访问存在tachyon里的文件。把Tachyon是架构在最底层的分布式文件存储和上层的各种计算框架之间的一种中间件。主要职责是将那些不需要落地到DFS里的文件，落地到分布式内存文件系统中，来达到共享内存，从而提高效率。同时可以减少内存冗余，GC时间等。



**Tachyon架构**

Tachyon的架构是传统的Master—slave架构，这里和Hadoop类似，TachyonMaster里WorkflowManager是 Master进程，因为是为了防止单点问题，通过Zookeeper做了HA，可以部署多台Standby Master。Slave是由Worker Daemon和Ramdisk构成。这里个人理解只有Worker Daemon是基于JVM的，Ramdisk是一个off heap memory。Master和Worker直接的通讯协议是Thrift。

下图来自Tachyon的作者Haoyuan Li：



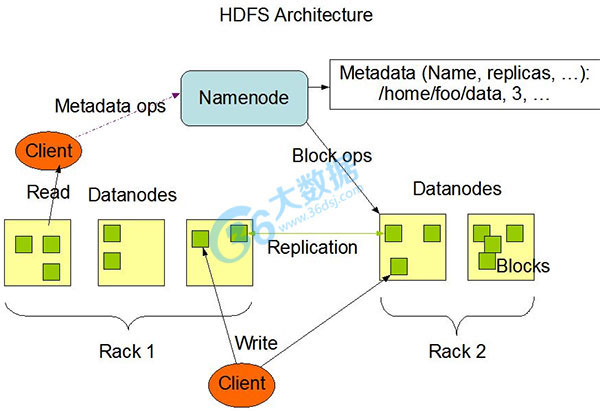
**下载地址**：[https://github.com/amplab/tachyon](http://tachyon-project.org/https:/github.com/amplab/tachyon)

**四、KFS**

**简介：**GFS的C++开源版本，Kosmos distributed file system (KFS)是一个专门为数据密集型应用（搜索引擎，数据挖掘等）而设计的存储系统，类似于Google的GFS和Hadoop的HDFS分布式文件系统。 KFS使用C++实现，支持的客户端包括C++，Java和Python。KFS系统由三部分组成，分别是metaserver、chunkserver和client library。

**官网：**<http://code.google.com/p/kosmosfs/>

**五、HDFS**



**简介**：Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。它和现有的分布式文件系统有很多共同点。但同时，它和其他的分布式文件系统的区别也是很明显的。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束，来实现流式读取文件系统数据的目的。HDFS在最开始是作为Apache Nutch搜索引擎项目的基础架构而开发的。HDFS是Apache Hadoop Core项目的一部分。

**官网**：<http://hadoop.apache.org/>

# ****资源管理****

**一、Twitter Mesos**

**开发者：**Twitter研发人员John Oskasson



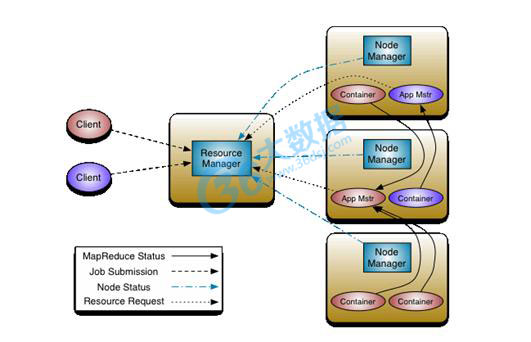
**简介**：Apache Mesos是由加州大学伯克利分校的AMPLab首先开发的一款开源群集管理软件，支持Hadoop、ElasticSearch、Spark、Storm 和Kafka等架构，由于其开源性质越来越受到一些大型云计算公司的青睐，例如Twitter、Facebook等。

**参考文章**：[Mesos渐入主流,Twitter模式有望“无限复制”-CSDN.NET](http://www.csdn.net/article/2014-02-24/2818520-Cloud-Mesosphere-Mesos-Hadoop-Spark)

**官网**：<http://mesos.apache.org/>

**二、Hadoop Yarn**

Hadoop 新 MapReduce 框架 Yarn。为从根本上解决旧 MapReduce 框架的性能瓶颈，促进 Hadoop 框架的更长远发展，从 0.23.0 版本开始，Hadoop 的 MapReduce 框架完全重构，发生了根本的变化。新的 Hadoop MapReduce 框架命名为 MapReduceV2 或者叫 Yarn，其架构图如下图所示：



Yarn 框架相对于老的 MapReduce 框架什么优势呢？我们可以看到：

1、这个设计大大减小了 JobTracker（也就是现在的 ResourceManager）的资源消耗，并且让监测每一个 Job 子任务 (tasks) 状态的程序分布式化了，更安全、更优美。

2、在新的 Yarn 中，ApplicationMaster 是一个可变更的部分，用户可以对不同的编程模型写自己的 AppMst，让更多类型的编程模型能够跑在 Hadoop 集群中，可以参考 hadoop Yarn 官方配置模板中的 mapred-site.xml 配置。

3、对于资源的表示以内存为单位 ( 在目前版本的 Yarn 中，没有考虑 cpu 的占用 )，比之前以剩余 slot 数目更合理。

4、老的框架中，JobTracker 一个很大的负担就是监控 job 下的 tasks 的运行状况，现在，这个部分就扔给 ApplicationMaster 做了，而 ResourceManager 中有一个模块叫做 ApplicationsMasters( 注意不是 ApplicationMaster)，它是监测 ApplicationMaster 的行状况，如果出问题，会将其在其他机器上重启。

5、Container 是 Yarn 为了将来作资源隔离而提出的一个框架。这一点应该借鉴了 Mesos 的工作，目前是一个框架，仅仅提供 java 虚拟机内存的隔离 ,hadoop 团队的设计思路应该后续能支持更多的资源调度和控制 , 既然资源表示成内存量，那就没有了之前的 map slot/reduce slot 分开造成集群资源闲置的尴尬情况。

官网：<http://hadoop.apache.org/>

第二部分将整合大数据日志收集系统、消息系统、集群管理、基础设施、监控管理等开源工具。并将于3月12日发布，尽请期待。

## [一共81个，开源大数据处理工具汇总（下），包括日志收集系统/集群管理/RPC等](http://www.36dsj.com/archives/25042)

End.