

华为 FusionInsight HD 2.6 技术白皮书


文档版本 01
发布日期 2016-03-20

版权所有 © 华为技术有限公司 2016。 保留一切权利。

非经华为技术有限公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI、华为、 是华为技术有限公司的商标或者注册商标。

在本手册中以及本手册描述的产品中，出现的其他商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。

免责声明

本文档可能含有预测信息，包括但不限于有关未来的财务、运营、产品系列、新技术等信息。由于实践中存在很多不确定因素，可能导致实际结果与预测信息有很大的差别。因此，本文档信息仅供参考，不构成任何要约或承诺。华为可能不经通知修改上述信息，恕不另行通知。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.huawei.com>

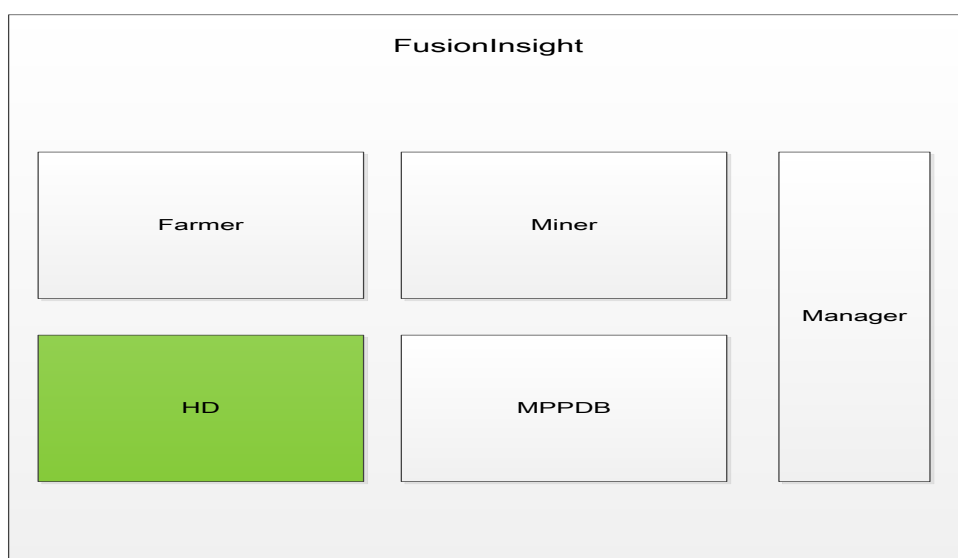
目 录

1 简介	3
1.1 FusionInsight HD 概述	3
1.2 FusionInsight HD 基础介绍	4
2 重点组件介绍	6
2.1 分布式文件系统 HDFS	6
2.1 分布式批处理引擎 MapReduce	6
2.2 统一资源管理和调度框架 YARN	7
2.3 数据仓库组件 Hive	8
2.4 分布式数据库 HBase	8
2.5 分布式内存计算框架 Spark	9
2.6 全文检索组件 Solr	10
2.7 Hadoop 集成开发工具 Hue	11
2.8 数据集成	13
2.8.1 Flume	13
2.8.2 Loader (Sqoop)	14
2.9 流处理 (Streaming)	17
2.9.1 Storm	17
2.9.2 StreamCQL	18
2.10 Redis	19

1 简介

1.1 FusionInsight HD 概述

FusionInsight 是华为企业级大数据存储、查询、分析的统一平台，能够帮助企业快速构建海量数据信息处理系统，通过对巨量信息数据实时与非实时的分析挖掘，发现全新价值点和企业商机。



FusionInsight 解决方案由 5 个子产品 FusionInsight HD、FusionInsight MPPDB、FusionInsight Miner、FusionInsight Farmer 和 FusionInsight Manager 构成。

- **FusionInsight HD：**企业级的大数据处理环境，是一个分布式数据处理系统，对外提供大容量的数据存储、分析查询和实时流式数据处理能力。
- **FusionInsight MPPDB：**企业级的MPP关系型数据库，基于列存储和MPP架构，是为面向结构化数据分析而设计开发的，能够有效处理PB级别的数据量。

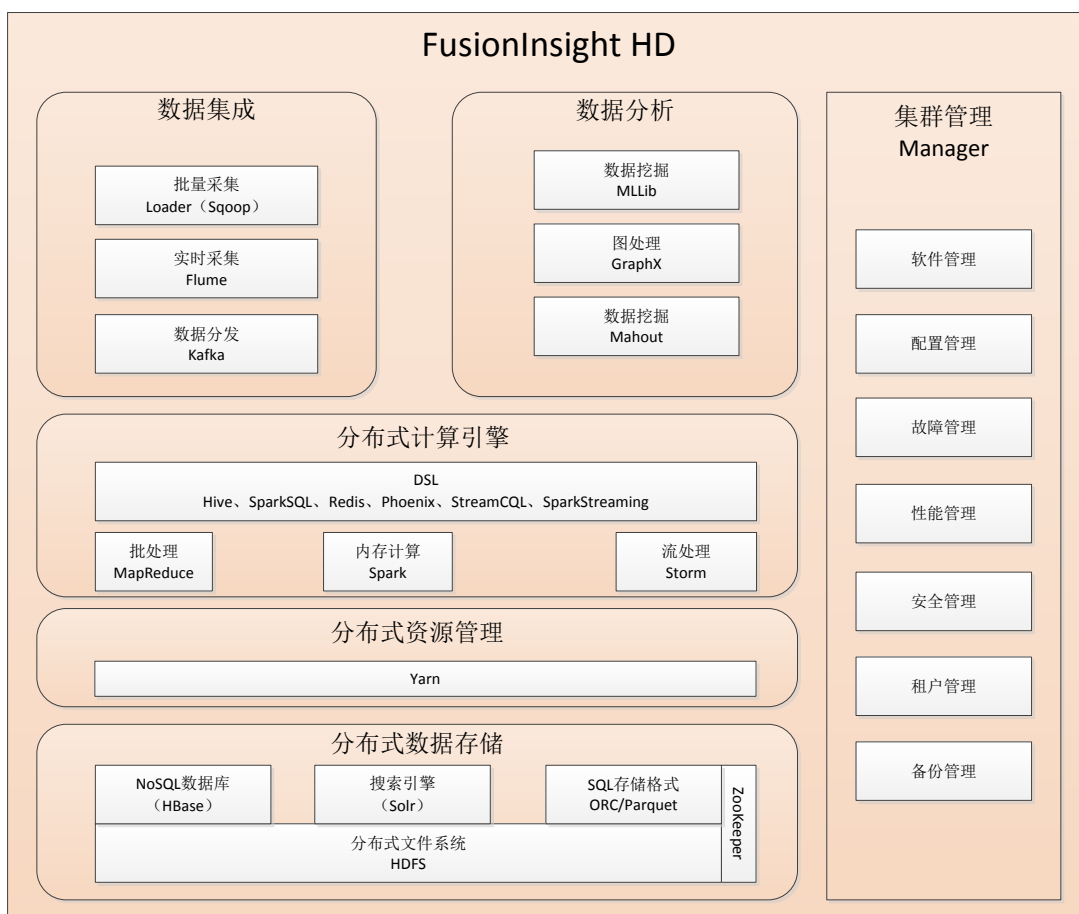
FusionInsight MPPDB在核心技术上跟传统数据库有巨大差别，可以解决很多行业用户的数据处理性能问题，可以为超大规模数据管理提供高性价比的通用计算平

台，并可用于支撑各类数据仓库系统、BI（Business Intelligence）系统和决策支持系统，统一为上层应用的决策分析等服务。

- **FusionInsight Miner：**企业级的数据分析平台，基于华为FusionInsight HD的分布式存储和并行计算技术，提供从海量数据中挖掘出价值信息的平台。
- **FusionInsight Farmer：**企业级的大数据应用容器，为企业业务提供统一开发、运行和管理的平台。

FusionInsight Manager：企业级大数据的操作运维提供，提供高可靠、安全、容错、易用的集群管理能力，支持大规模集群的安装部署、监控、告警、用户管理、权限管理、审计、服务管理、健康检查、问题定位、升级和补丁等功能。

1.2 FusionInsight HD 基础介绍



FusionInsight HD 需要对开源组件进行封装和增强，对外提供稳定的大容量的数据存储、查询和分析能力。各自组件提供功能如下：

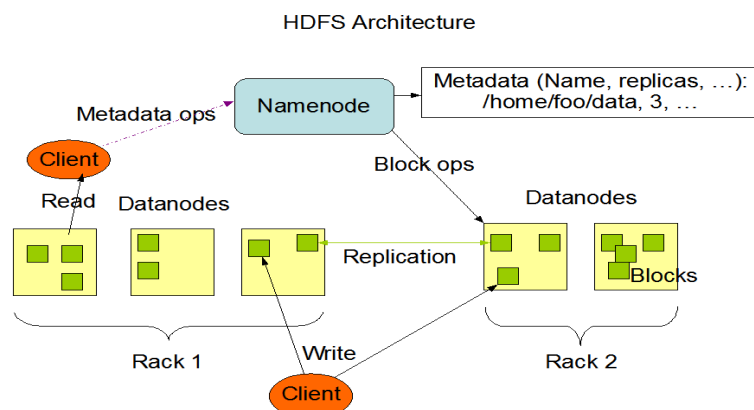
- **Manager:** 作为运维系统，为 FusionInsight HD 提供高可靠、安全、容错、易用的集群管理能力，支持大规模集群的安装/升级/补丁、配置管理、监控管理、告警管理、用户管理、租户管理等。
- **HDFS:** Hadoop 分布式文件系统 (Hadoop Distributed File System)，提供高吞吐量的数据访问，适合大规模数据集方面的应用。
- **HBase:** 提供海量数据存储功能，是一种构建在 HDFS 之上的分布式、面向列的存储系统。
- **Oozie:** 提供了对开源 Hadoop 组件的任务编排、执行的功能。以 Java Web 应用程序的形式运行在 Java servlet 容器（如：Tomcat）中，并使用数据库来存储 工作流定义、当前运行的工作流实例（含实例的状态和变量）。
- **ZooKeeper:** 提供分布式、高可用性的协调服务能力。帮助系统避免单点故障，从而建立可靠的应用程序。
- **Redis:** 提供基于内存的高性能分布式 K-V 缓存系统。
- **Yarn:** Hadoop 2.0 中的资源管理系统，它是一个通用的资源模块，可以为各类应用程序进行资源管理和调度。
- **Mapreduce:** 提供快速并行处理大量数据的能力，是一种分布式数据处理模式和执行环境。
- **Spark:** 基于内存进行计算的分布式计算框架。
- **Hive:** 建立在 Hadoop 基础上的开源的数据仓库，提供类似 SQL 的 Hive QL 语言操作结构化数据存储服务和基本的数据分析服务。
- **Loader:** 基于 Apache Sqoop 实现 FusionInsight HD 与关系型数据库、ftp/sftp 文件服务器之间数据批量导入/导出工具；同时提供 Java API/shell 任务调度接口，供第三方调度平台调用。
- **Hue:** 提供了开源 Hadoop 组件的 WebUI，可以通过浏览器操作 HDFS 的目录和文件，调用 Oozie 来创建、监控和编排工作流，可操作 Loader 组件，查看 ZooKeeper 集群情况。
- **Flume:** 一个分布式、可靠和高可用的海量日志聚合系统，支持在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，并写入各种数据接受方（可定制）的能力。
- **Solr:** 一个高性能，基于 Lucene 的全文检索服务器。Solr 对 Lucene 进行了扩展，提供了比 Lucene 更为丰富的查询语言，同时实现了可配置、可扩展，并对查询性能进行了优化，并且提供了一个完善的功能管理界面，是一款非常优秀的全文检索引擎。
- **Kafka:** 一个分布式的、分区的、多副本的实时消息发布-订阅系统。提供可扩展、高吞吐、低延迟、高可靠的消息分发服务。
- **Streaming:** 基于 Apache Storm 的一个分布式、可靠、容错的实时流式数据处理的系统，并提供类 SQL (StreamCQL) 的查询语言。
- **SparkSQL:** 基于 Spark 引擎的高性能 SQL 引擎，可与 Hive 实现元数据共享。
- **Mahout:** 提供基于 MapReduce 的数据挖掘算法库
- **MLLib:** 提供基于 Spark 的数据挖掘算法库
- **GraphX:** 提供基于 Spark 的图处理算法库

2 重点组件介绍

2.1 分布式文件系统 HDFS

HDFS 是 Hadoop 的分布式文件系统，实现大规模数据可靠的分布式读写。HDFS 针对的使用场景是数据读写具有“一次写，多次读”的特征，而数据“写”操作是顺序写，也就是在文件创建时的写入或者在现有文件之后的添加操作。HDFS 保证一个文件在一个时刻只被一个调用者执行写操作，而可以被多个调用者执行读操作。

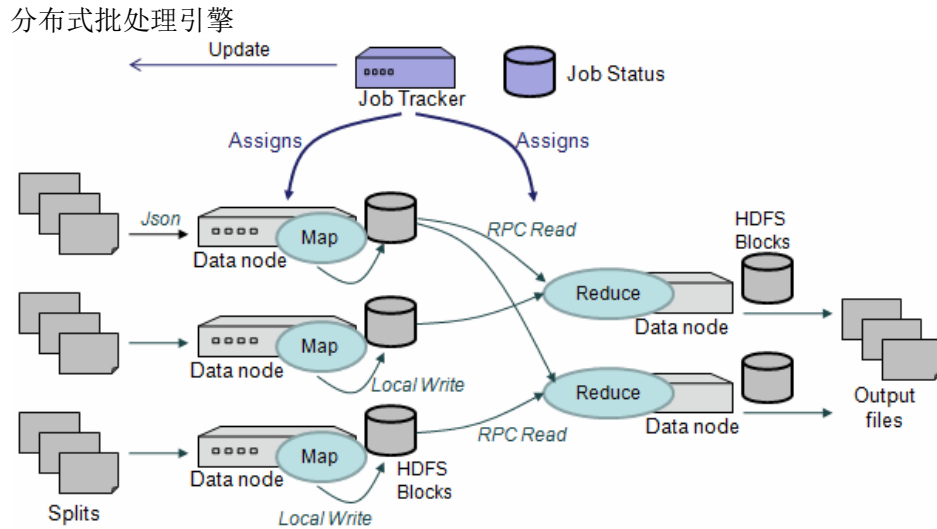
图1 分布式文件系统 HDFS



2.2 分布式批处理引擎 MapReduce

MapReduce 是 Hadoop 的核心，是 Google 提出的一个软件架构，用于大规模数据集（大于 1TB）的并行运算。概念“Map（映射）”和“Reduce（化简）”，及他们的主要思想，都是从函数式编程语言借来的，还有从矢量编程语言借来的特性。

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（化简）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。



MapReduce 是用于并行处理大数据集的软件框架。MapReduce 的根源是函数性编程中的 map 和 reduce 函数。Map 函数接受一组数据并将其转换为一个键/值对列表，输入域中的每个元素对应一个键/值对。Reduce 函数接受 Map 函数生成的列表，然后根据它们的键缩小键/值对列表。MapReduce 起到了将大事务分散到不同设备处理的能力，这样原本必须用单台较强服务器才能运行的任务，在分布式环境下也能完成了。

2.3 统一资源管理和调度框架 YARN

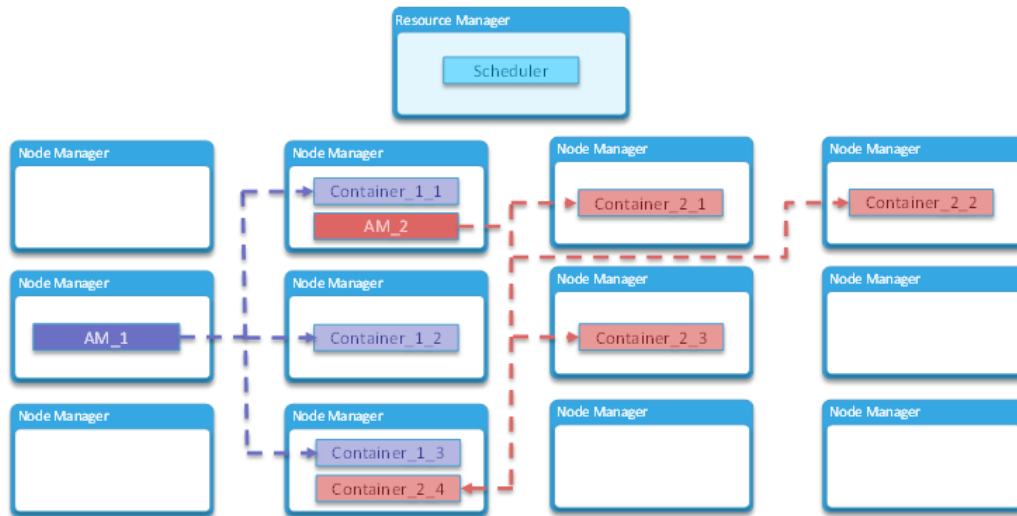
为了实现一个 Hadoop 集群的集群共享、可伸缩性和可靠性，并消除早期 MapReduce 框架中的 JobTracker 性能瓶颈，开源社区引入了统一的资源管理框架 YARN。

YARN 分层结构的本质是 ResourceManager。这个实体控制整个集群并管理应用程序向基础计算资源的分配。ResourceManager 将各个资源部分（计算、内存、带宽等）精心安排给基础 NodeManager（YARN 的每节点代理）。ResourceManager 还与 Application Master 一起分配资源，与 NodeManager 一起启动和监视它们的基础应用程序。在此上下文中，Application Master 承担了以前的 TaskTracker 的一些角色，ResourceManager 承担了 JobTracker 的角色。

Application Master 管理一个在 YARN 内运行的应用程序的每个实例。Application Master 负责协调来自 ResourceManager 的资源，并通过 NodeManager 监视容器的执行和资源使用（CPU、内存等的资源分配）。请注意，尽管目前的资源更加传统（CPU 核心、内存），但未来会带来基于手头任务的新资源类型（比如图形处理单元或专用处理设备）。从 YARN 角度讲，Application Master 是用户代码，因此存在潜在的安全问题。YARN 假设 Application Master 存在错误或者甚至是恶意的，因此将它们当作无特权的代码对待。

NodeManager 管理一个 YARN 集群中的每个节点。NodeManager 提供针对集群中每个节点的服务，从监督对一个容器的终生管理到监视资源和跟踪节点健康。MRv1 通过插槽管理 Map 和 Reduce 任务的执行，而 NodeManager 管理抽象容器，这些容器代表着可供一个特定应用程序使用的针对每个节点的资源。

图2 统一资源管理和调度框架 YARN



2.4 数据仓库组件 Hive

Hive 是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。同时，这个语言也允许熟悉 MapReduce 开发者的开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作。

Hive 体系结构：

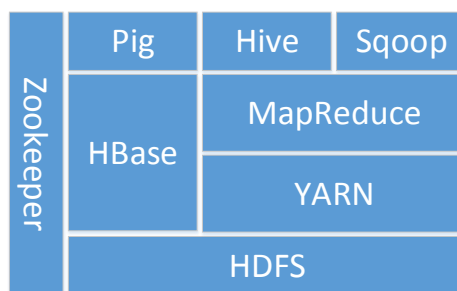
用户接口：用户接口主要有三个：CLI，Client 和 WUI。其中最常用的是 CLI，CLI 启动的时候，会同时启动一个 Hive 副本。Client 是 Hive 的客户端，用户连接至 Hive Server。在启动 Client 模式的时候，需要指出 Hive Server 所在节点，并且在该节点启动 Hive Server。WUI 是通过浏览器访问 Hive。

元数据存储：Hive 将元数据存储于数据库中，如 mysql、derby。Hive 中的元数据包括表的名字，表的列和分区及其属性，表的属性（是否为外部表等），表的数据所在目录等。

2.5 分布式数据库 HBase

数据存储使用 HBase 来承接，HBase 是一个开源的、面向列（Column-Oriented）、适合存储海量非结构化数据或半结构化数据的、具备高可靠性、高性能、可灵活扩展伸缩的、支持实时数据读写的分布式存储系统。

图3 分布式数据库 HBase



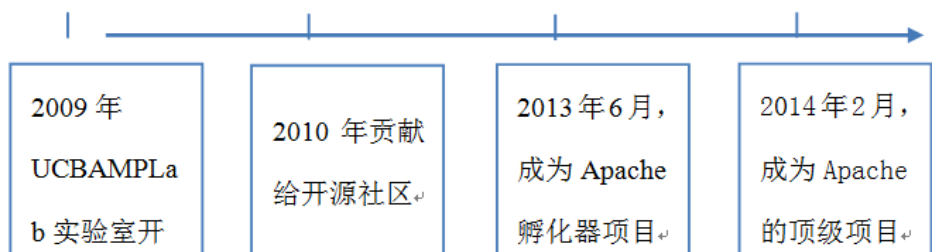
存储在 HBase 中的表的典型特征：

- 大表 (BigTable)：一个表可以有上亿行，上百万列
- 面向列：面向列(族)的存储、检索与权限控制
- 稀疏：表中为空(null)的列不占用存储空间

2.6 分布式内存计算框架 Spark

Apache Spark 是一个开源的，通用的分布式集群计算引擎。Spark 发展历程：

图4 Spark 发展历程



FusionInsight Spark 是一个开源的，并行数据处理框架，能够帮助用户简单的开发快速，统一的大数据应用，对数据进行，协处理，流式处理，交互式分析等等。

Spark 具有如下特点：

- 快速：数据处理能力，比 MapReduce 快 10-100 倍。
- 易用：可以通过 Java, Scala, Python, 简单快速的编写并行的应用处理大数据量，Spark 提供了超过 80 种高层的操作符来帮助用户组件并行程序。
- 普遍性：Spark 提供了众多高层的工具，例如 Spark SQL, MLib, GraphX, Spark Stream, 可以在一个应用中，方便的将这些工具进行组合。

与 Hadoop 集成：Spark 能够直接运行于 Hadoop 2.0 的集群，并且能够直接读取现存的 Hadoop 数据。尤其，Spark 和 FusionInsight 紧密结合，可以通过 FusionInsight Manager 部署安装 Spark。

Spark 提供了一个快速的计算，写入，以及交互式查询的框架。相比于 Hadoop，Spark 拥有明显的性能优势。Spark 使用 in-memory 的计算方式，通过这种方式来避免一个 MapReduce 工作流中的多个任务对同一个数据集进行计算时的 IO 瓶颈。Spark 利用 Scala 语言实现，Scala 能够使得处理分布式数据集时，能够像处理本地化数据一样。

除了交互式的数据分析，Spark 还能够支持交互式的数据挖掘，由于 Spark 是基于内存的计算，很方便处理迭代计算，而数据挖掘的问题通常都是对同一份数据进行迭代计算。除此之外，Spark 能够运行于安装 Hadoop 2.0 Yarn 的集群。之所以 Spark 能够在保留 MapReduce 容错性，数据本地化，可扩展性等特性的同时，能够保证性能的高效，并且避免繁忙的磁盘 IO，主要原因是因为 Spark 创建了一种叫做 RDD（Resilient Distributed Dataset）的内存抽象结构。

原有的分布式内存抽象，例如 key-value store 以及数据库，支持对于可变状态的细粒度更新，这一点要求集群需要对数据或者日志的更新进行备份来保障容错性。这样就会给数据密集型的工作流带来大量的 IO 开销。而对于 RDD 来说，它只有一套受限制的接口，仅仅支持粗粒度的更新，例如 map，join 等等。通过这种方式，Spark 只需要简单的记录建立数据的转换操作的日志，而不是完整的数据集，就能够提供容错性。这种数据的转换链记录就是数据集的溯源。由于并程序，通常是对一个大数据集应用相同的计算过程，因此之前提到的粗粒度的更新限制并没有想象中的大。事实上，Spark 论文中阐述了 RDD 完全可以作为多种不同计算框架，例如 MapReduce，Pregel 等的编程模型。

并且，Spark 同时提供了操作允许用户显示的将数据转换过程持久化到硬盘。对于数据本地化，是通过允许用户能够基于每条记录的键值，控制数据分区实现的。（采用这种方式的一个明显好处是，能够保证两份需要进行关联的数据将会被同样的方式进行哈希）。如果内存的使用超过了物理限制，Spark 将会把这些比较大的分区写入到硬盘，由此来保证可扩展性。

2.7 全文检索组件 Solr

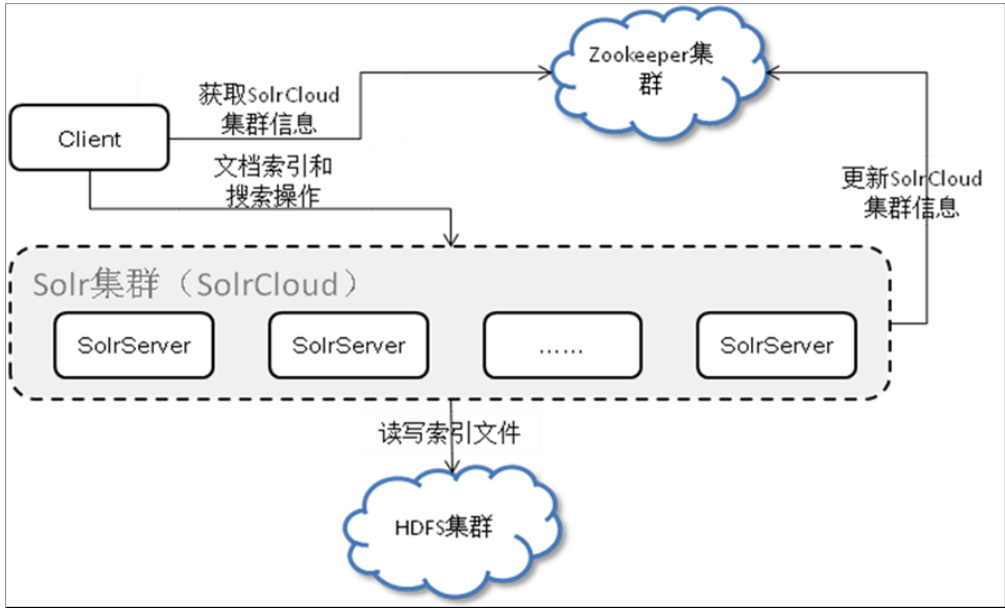
Solr 是基于 Apache Lucene 的独立的企业级应用搜索服务器。它对外提供了类似于 REST 的 HTTP/XML 和 JSON 的 API。其主要功能包括强大的全文检索，高亮显示，层面搜索，近实时索引，动态聚类，数据库整合，丰富的文档（如 Word 中，PDF 格式）处理和地理信息搜索等。

Solr 作为业界优秀的企业搜索服务器具有以下特性：

- 先进的全文搜索功能
- 优化的高容量网络流量
- 基于标准的开放接口——XML，JSON 和 HTTP
- 综合的 HTML 管理界面
- 采用 JMX 监控服务器统计信息
- 线性可扩展性，自动索引复制，自动故障转移和恢复
- 近实时索引
- 采用 XML 配置达到灵活性和适配性

➤ 可扩展的插件架构

Solr 集群方案逻辑组成



名称	说明
Client	Client 使用 HTTP 协议同 Solr 集群（SolrCloud）中的 SolrServer 进行通信，进行分布式索引和分布式搜索操作。
SolrServer	<p>SolrServer 负责提供创建索引和全文检索等服务，是 Solr 集群中的数据计算和处理单元。</p> <p>SolrServer 一般与 HDFS 集群中的 DataNode 合并。从而可以提供更高性能的索引和搜索服务。</p>
ZooKeeper 集群	ZooKeeper 为 Solr 集群中各进程提供分布式协作服务。各 SolrServer 将自己的信息（collection 配置信息、SolrServer 健康信息等）注册到 Zookeeper 中，Client 据此感知各个 SolrServer 的健康状态来决定索引和搜索请求的分发。
HDFS 集群	HDFS 为 Solr 提供高可靠的文件存储服务，Solr 的索引文件全部存储在 HDFS 中。

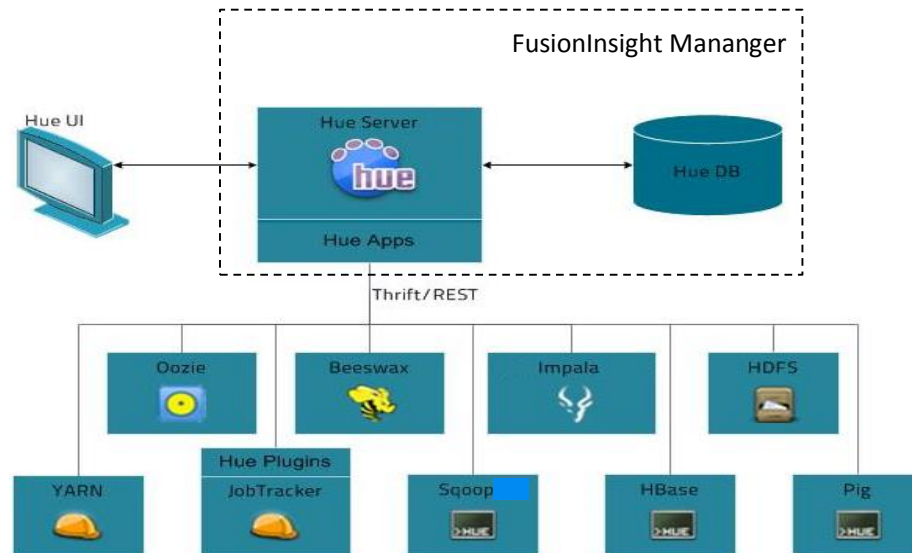
2.8 Hadoop 集成开发工具 Hue

Hue 是一组 web 应用，用于和 FusionInsight 平台进行交互，能够帮助用户浏览 HDFS，进行 Hive 查询，启动 MapReduce 任务以及 Oozie 工作流等。

Hue 运行于浏览器，在 FusionInsight 中，HUE 被集成在 FusionInsight Manager 中。

下图是 Hue 的整体架构，描述了 Hue 的工作机制。Hue Server 是一个集成在 FusionInsight Manager 上的 web 应用的容器。它承载了与所有 FusionInsight 组件交互的应用。

图5 集成开发工具 Hue



Hue 主要包括了如下的组件及功能：

- 文件浏览器——该应用能够允许用户直接通过界面浏览以及操作 HDFS 的不同目录，主要包含如下功能：
 - ✓ 创建文件及目录，上传下载文件，重命名，移动，删除文件及目录。修改文件以及目录的属主，权限。
 - ✓ 搜索文件，目录，文件所有人，所属用户组。
 - ✓ 查看编辑文件。
- 查询编辑器——用户能够通过查询查询编辑器，编写简单的 SQL，查询存储在 Hadoop 之上的数据。例如 HDFS，HBase，Hive。用户可以方便的，创建，管理，执行 SQL，并且能够以 Excel 的形式下载执行的结果。主要功能如下：
 - ✓ SQL 编辑，执行，SQL 模板保存，模板复制，模板编辑。SQL 解释，查询，历史记录。
 - ✓ 数据库展示，数据表展示。
 - ✓ 支持多种 Hadoop 存储。
- 工作流控制——主要包含如下功能：
 - ✓ 任务浏览器
 - 提供任务列表，根据具体任务找到对应子任务的相关信息，状态，开始，结束时间
 - 查看任务日志

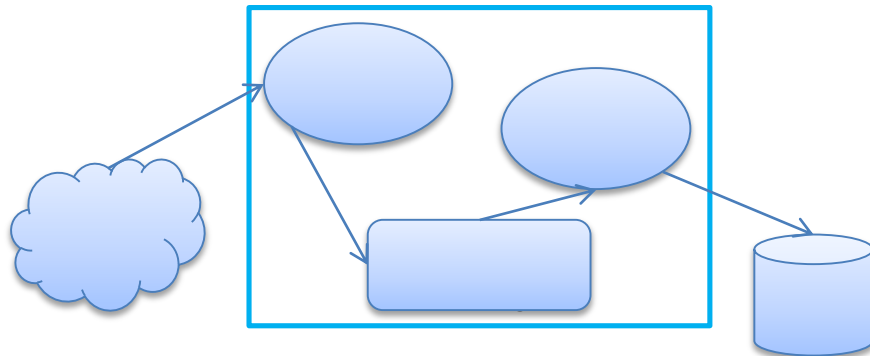
- ✓ 任务定制器
 - 能够帮助用户方便的创建提交任务。
 - 可以给具体的任务输入变量，参数
- ✓ Oozie 编辑器——Oozie 编辑器允许用户定义 Oozie 工作流以及协调器。
 - 工作流是一组任务的结合，控制任务执行顺序。工作流能够自动控制所属节点任务的执行，停止，克隆等等操作。
 - 协调器应用允许用户定义和执行周期性以及相互依赖的工作流任务，并配置工作流能够执行的条件。
 - 用户管理——类似于常规的 Web 应用，Hue 也提供了用户管理的功能，能够添加删除管理用户信息。

2.9 数据集成

包括 Loader、Flume、FTP Server。Loader 实现 FusionInsight 与关系型数据库、文件系统之间交换数据和文件的数据加载工具，集成开源社区 Sqoop 功能，同时提供 REST API 接口，供第三方调度平台调用。Flume 提供日志导入功能。

2.9.1 Flume

Flume 是一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统，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。其中 Flume-NG 是 Flume 的一个分支，其目的是要明显简单，体积更小，更容易部署。其最基本的架构如下图所示：



Flume-NG 由一个个 Agent 来组成，而每个 Agent 由 Source、Channel、Sink 三个模块组成，其中 Source 负责接收数据，Channel 负责数据的传输，Sink 则负责数据向下一端的发送。

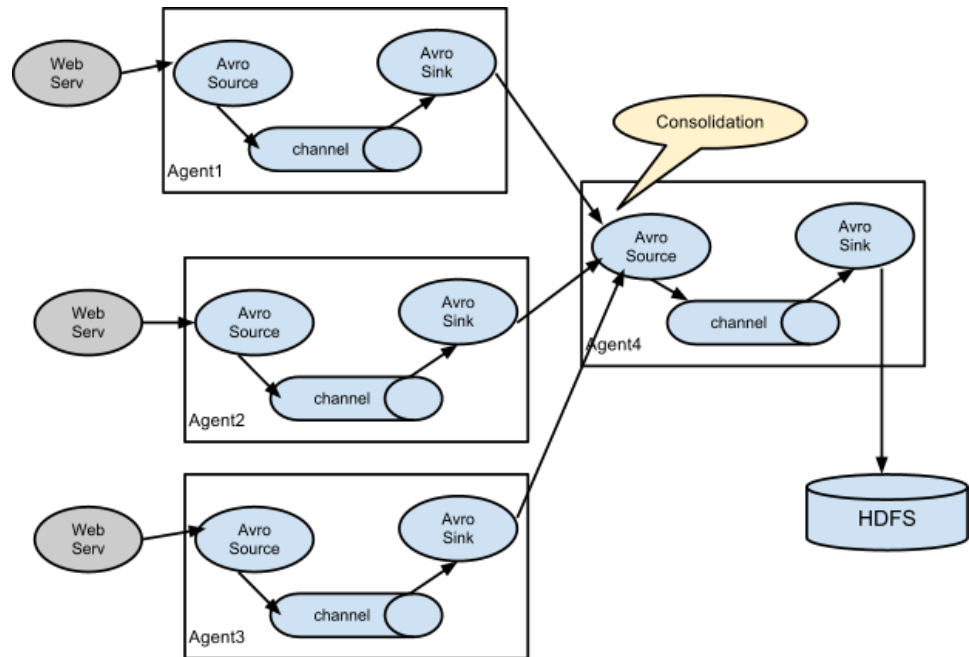
Source: 完成对日志数据的收集，分成 transtion 和 event 打入到 channel 之中。

Channel: 主要提供一个队列的功能，对 source 提供中的数据进行简单的缓存。

Sink: 取出 Channel 中的数据，进行相应的存储文件系统，数据库，或者提交到远程服务器

Flume 的可靠性基于 Agent 间事务的交换，下一个 Agent down 掉，Channel 可以持久化数据，Agent 恢复后再传输。可用性则基于内建的 Load balancing 和 Failover 机制。Channel 及 Agent 都可以配多个实体，实体之间可以使用负载分担等策略

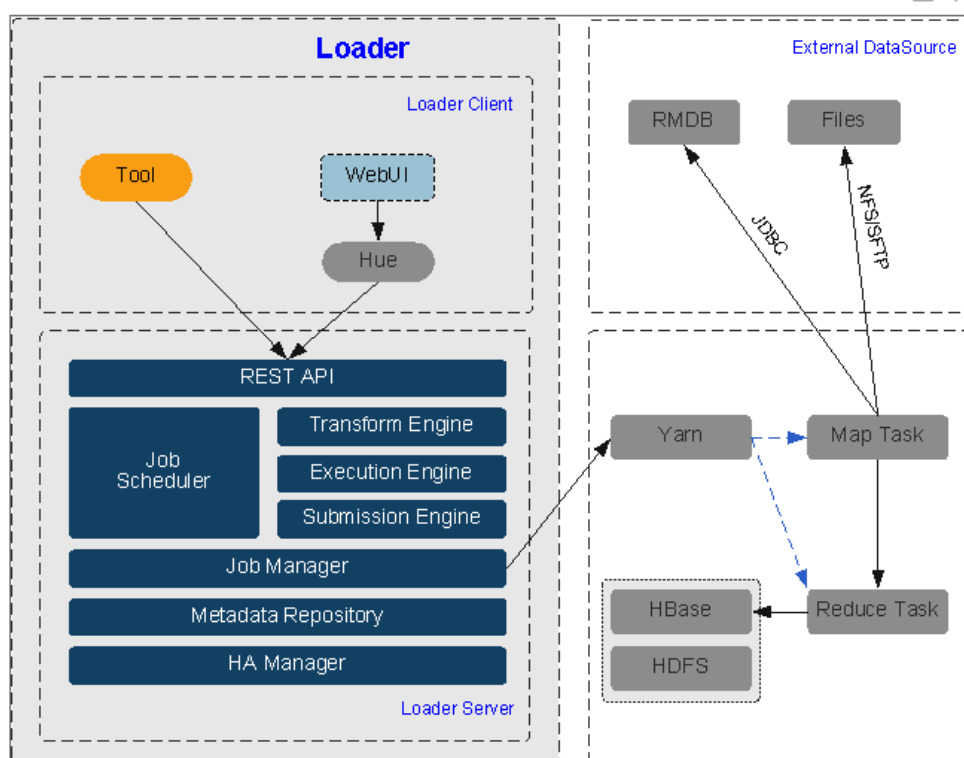
每个 agent 为一个 JVM 进程，同一台服务器可以有多个 agent。收集节点(agent1, 2, 3)负责处理日志，汇聚节点(agent4)负责写入 HDFS，每个收集节点的 agent 可以选择多个汇聚节点，这样可以实现负载均衡。



2.9.2 Loader (Sqoop)

Loader 是在开源 Sqoop 组件的基础上进行了一些扩展，实现 FusionInsight 与关系型数据库、文件系统之间交换“数据”、“文件”，同时也可以将数据从关系型数据库或者文件服务器导入到 FusionInsight 的 HDFS/HBase 中，或者反过来从 HDFS/HBase 导出到关系型数据库或者文件服务器中。

Loader 模型主要由 Loader Client 和 Loader Server 组成，如图 1 所示。



图中各部分的功能说明如表 1 所示。

名称	描述
Loader Client	Loader 的客户端，包括 WebUI 和 CLI 版本两种交互界面。
Loader Server	Loader 的服务端，主要功能包括：处理客户端操作请求、管理连接器和元数据、提交 MapReduce 作业和监控 MapReduce 作业状态等。
REST API	实现 RESTful（HTTP + JSON）接口，处理来自客户端的操作请求。
Job Scheduler	简单的作业调度模块，支持周期性的执行 Loader 作业。
Transform Engine	数据转换处理引擎，支持字段合并、字符串剪切、字符串反序等。
Execution Engine	Loader 作业执行引擎，支持以 MapReduce 方式执行 Loader 作业。
Submission Engine	Loader 作业提交引擎，支持将作业提交给 MapReduce 执行。
Job Manager	管理 Loader 作业，包括创建作业、查询作业、更新作业、删除作业、激活作业、去激活作业、启动作业、停止作业。
Metadata Repository	元数据仓库，存储和管理 Loader 的连接器、转换步骤、作业等数据。

名称	描述
HA Manager	管理 Loader Server 进程的主备状态，Loader Server 包含 2 个节点，以主备方式部署。

通过 MapReduce 实现并行执行和容错

Loader 通过 MapReduce 作业实现并行的导入或者导出作业任务，不同类型的导入导出作业可能只包含 Map 阶段或者同时 Map 和 Reduce 阶段。

Loader 同时利用 MapReduce 实现容错，在作业任务执行失败时，可以重新调度。

数据导入到 HBase

在 MapReduce 作业的 Map 阶段中从外部数据源抽取数据。

在 MapReduce 作业的 Reduce 阶段中，按 Region 的个数启动同样个数的 Reduce Task，Reduce Task 从 Map 接收数据，然后按 Region 生成 HFile，存放在 HDFS 临时目录中。

在 MapReduce 作业的提交阶段，将 HFile 从临时目录迁移到 HBase 目录中。

数据导入 HDFS

在 MapReduce 作业的 Map 阶段中从外部数据源抽取数据，并将数据输出到 HDFS 临时目录下（以“输出目录-ldtmp”命名）。

在 MapReduce 作业的提交阶段，将文件从临时目录迁移到输出目录中。

数据导出到关系型数据库

在 MapReduce 作业的 Map 阶段，从 HDFS 或者 HBase 中抽取数据，然后将数据通过 JDBC 接口插入到临时表（Staging Table）中。

在 MapReduce 作业的提交阶段，将数据从临时表迁移到正式表中。

数据导出到文件系统

在 MapReduce 作业的 Map 阶段，从 HDFS 或者 HBase 中抽取数据，然后将数据写入到文件服务器临时目录中。

在 MapReduce 作业的提交阶段，将文件从临时目录迁移到正式目录

2.10 流处理（Streaming）

2.10.1 Storm

Apache Storm 是一个分布式、可靠、容错的实时流式数据处理系统。在 Storm 中，先要设计一个用于实时计算的图状结构，我们称之为拓扑（topology）。这个拓扑将会被提交给集群，由集群中的主控节点（master node）分发代码，将任务分配给工作节点（worker node）执行。一个拓扑中包括 spout 和 bolt 两种角色，其中 spout 发送消息，负责将数据流以 tuple 元组的形式发送出去；而 bolt 则负责转换这些数据流，在 bolt 中可以完成计算、过滤等操作，bolt 自身也可以随机将数据发送给其他 bolt。由 spout 发射出的 tuple 是不可变数组，对应着固定的键值对。

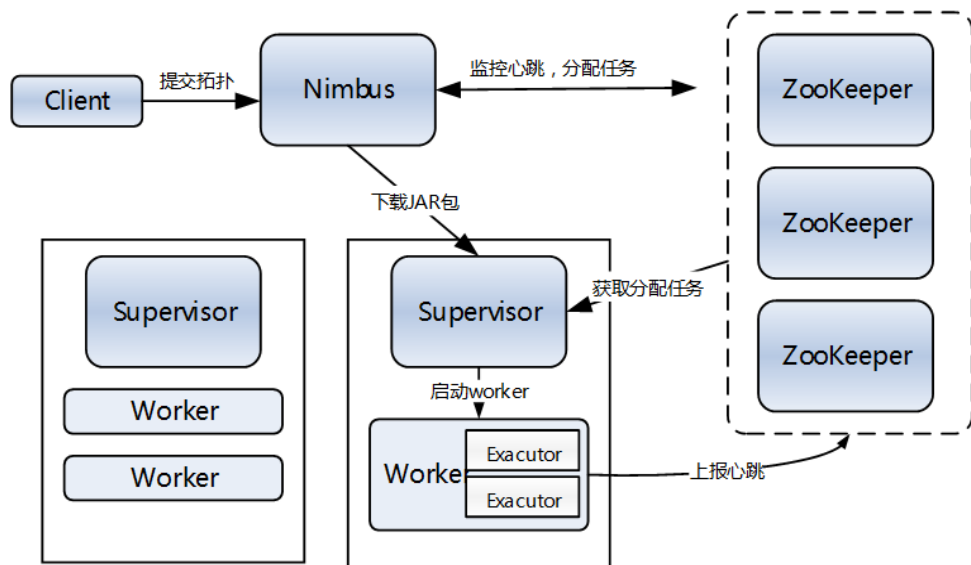


图2-1 Storm 系统架构

业务处理逻辑被封装进 Storm 中的 topology 中。一个 topology 是由一组 Spout 组件（数据源）和 Bolt 组件（逻辑处理）通过 Stream Groupings 进行连接的有向无环图（DAG）。Topology 里面的每一个 Component（Spout/Bolt）节点都是并行运行的。在 topology 里面，可以指定每个节点的并行度，storm 则会在集群里面分配相应的 Task 来同时计算，以增强系统的处理能力。

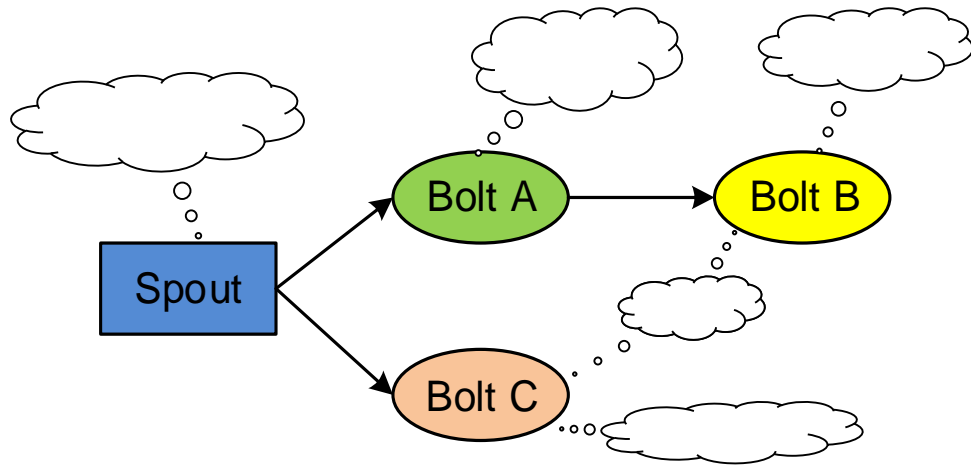


图2-2 Topology

2.10.2 StreamCQL

StreamCQL (Stream Continuous Query Language), 流式查询语言, 是一种用于实时数据流上的查询语言, 它是一种类 SQL 语言, 相对于 SQL, StreamCQL 中增加了 (时序) 窗口的概念, 将待处理的数据保存在内存中, 进行快速的内存计算, StreamCQL 的输出结果为数据流在某一时刻的计算结果。使用 StreamCQL, 可以快速进行业务开发, 并方便地将业务提交到 Storm 平台开启实时数据的接收、处理及结果输出; 并可以在合适的时候中止业务。

StreamCQL 具有如下几个特点:

- 使用简单: StreamCQL语法和标准SQL语法类似, 只要具备SQL基础, 都可以快速进行开发。
- 功能丰富: StreamCQL除了包含标准SQL的各类基本表达式等功能之外, 还特别针对流处理场景增加了窗口, 窗口前过滤, 窗口后过滤, 并发度设置等功能, 满足多种实时业务处理场景。
- 易于拓展: StreamCQL提供了拓展接口, 以支撑日益复杂的业务场景, 用户可以自定义输入、输出、序列化、反序列化等并结合已有功能来满足灵活的业务场景。
- 易于调试: StreamCQL提供了详细的异常码说明, 降低了用户对于各种错误的处理难度, 提升了易用性

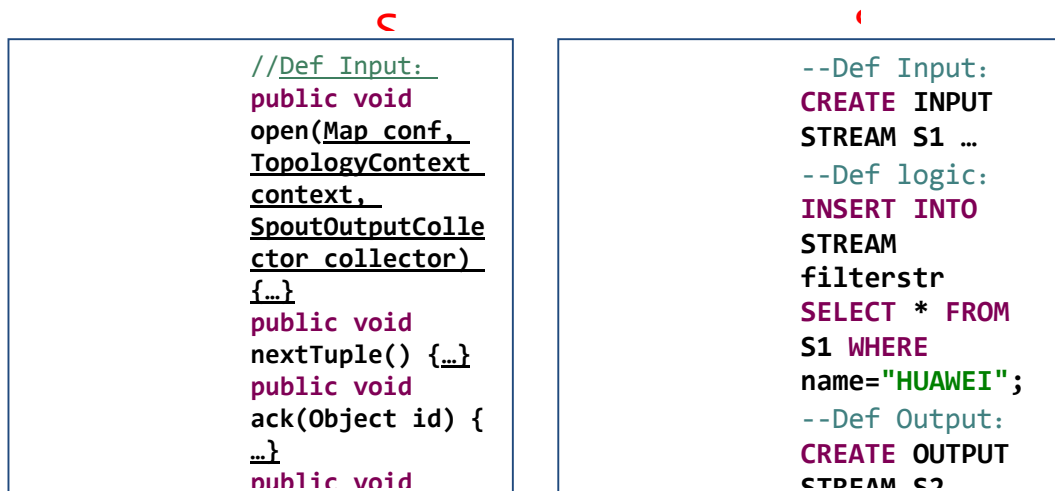


图2-3 Storm 原生 API 与 StreamCQL 比较

2.11 Redis

Redis (REmote DIctionary Service), 是 C 语言编写的高性能 Key-Value 内存数据库, 支持多种数据类型, 包括 string (字符串)、list (链表)、set (集合)、zset (有序集合)、hash 等。Redis 集群模式具有更多优点, 适合生产环境使用, 但 Redis 集群管理复杂, 容易出错, 社区版本部分功能不完善。FusionInsight HD 提供图形化的 Redis 集群管理功能。

- 向导式创建 Redis 集群系统

FusionInsight HD 支持一主一从模式的 Redis 集群, 系统自动计算节点上可安装的 Redis 实例个数并分配主从关系。

- 集群扩容、减容

当集群需要提供大规模的处理能力时, 可以一键式扩容一对或多对主从实例。在此过程中, 系统会自动完成数据迁移和数据平衡, 用户无需其他操作。

- Balance

出现扩容异常、部分实例掉线等异常场景时, Redis 集群中的数据可能会分布不均匀, 此时可以通过管理界面上提供的 Balance 功能, 让系统自动对集群数据进行平衡, 保证集群的健康运行。

- 性能监控与告警

系统提供 Redis 集群的性能监控功能, 可以通过直观的曲线图方式, 了解当前 Redis 集群、实例的 TPS 吞吐量情况。

- 集群可靠性保证

社区自带的集群创建工具还不完善, 只是按顺序在节点上分配主从实例。有可能将同一组主从实例排在同一节点上, 如此不能处理节点故障场景。

FusionInsight HD 在创建 Redis 集群的时候，能够自动将同一组主从实例安排在不同节点上，同时在进行扩容和减容的操作时，仍然会保证该原则。这样可以保证集群内任一节点发生故障，都能够通过主从实例倒换来保证业务不中断。

- 优化集群性能

内置了 OS 层、应用层的性能调优；比社区版性能更好，此调优开箱即用，不需额外开发、操作。