|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **创建人** | **修改人** | **日期** |
| V1.0 | 简德群 |  | 2021-06-25 |
|  |  |  |  |
|  |  |  |  |



平台研发部

数据采架构说明文档

架构说明文档v1.0

目录

[第一章 项目背景 2](#_Toc75530960)

[第二章 系统整体架构及其流程图 2](#_Toc75530961)

[第三章 软件介绍 5](#_Toc75530962)

[1 Apache Flume日志收集系统 5](#_Toc75530963)

[1.1 概述 5](#_Toc75530964)

[1.2 系统架构 6](#_Toc75530965)

[2 Hadoop大数据 6](#_Toc75530966)

[2.1 概述 6](#_Toc75530967)

[2.2 系统架构 6](#_Toc75530968)

[2.3 HDFS 分布式文件系统 7](#_Toc75530969)

[2.4 MapReduce 7](#_Toc75530970)

[2.5 YARN 8](#_Toc75530971)

[3 Kafka实时消息通信系统 9](#_Toc75530972)

[3.1 概述 9](#_Toc75530973)

[3.2 系统架构 10](#_Toc75530974)

[4 HIVE数据仓库 11](#_Toc75530975)

[4.1 概述 11](#_Toc75530976)

[4.2 系统架构 11](#_Toc75530977)

[5 SQOOP数据迁移工具 12](#_Toc75530978)

[5.1 概述 12](#_Toc75530979)

[5.2 架构 12](#_Toc75530980)

[6 Azkaban工作流任务调度系统 12](#_Toc75530981)

[6.1 概述 12](#_Toc75530982)

[6.2 系统架构 13](#_Toc75530983)

[7 MYSQL数据库 13](#_Toc75530984)

[7.1 概述 13](#_Toc75530985)

[7.2 系统架构 14](#_Toc75530986)

[8 报表后台管理系统 15](#_Toc75530987)

[8.1 概述 15](#_Toc75530988)

[8.2 系统架构 15](#_Toc75530989)

# 项目背景

随着大数据的发展，以数据驱动业务发展，已慢慢成为互联网公司业务发展的重要组成部分。大部分互联网企业在日常运营中对用户访问操作产生的海量数据进行分析和处理，可以得出用户的行为，数据业务的各种形态等更能让企业更快速精准的进行业务形态变更或业务拓展。

# 系统整体架构及其流程图

架构图：



业务日志采集流程图



日志采集ETL分析流程图



数据报表通用流程图



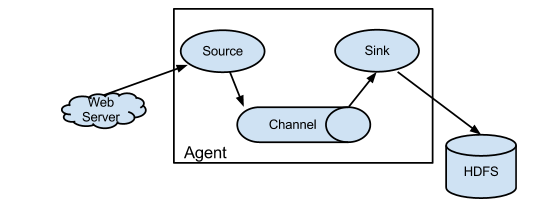
# 软件介绍

## Apache Flume日志收集系统

### 概述

Flume是Cloudera提供的一个高可用的，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。

### 系统架构

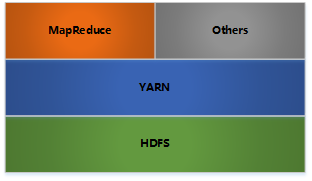


## Hadoop大数据

### 概述

Hadoop是一个开源框架，允许使用简单的编程模型在跨计算机集群的分布式环境中存储和处理大数据。它的设计是从单个服务器扩展到数千个机器，每个都提供本地计算和存储。

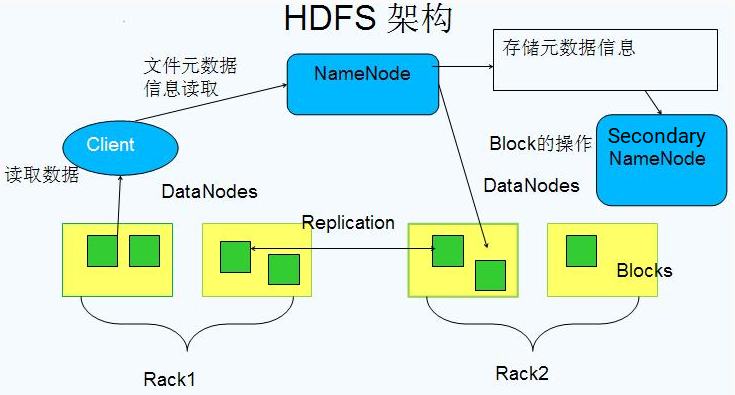
### 系统架构



* HDFS:分布式文件存储
* YARN:分布式资源管理
* MapReduce:分布式计算
* Others:利用YARN的资源管理功能实现其他的数据处理方式

### HDFS 分布式文件系统

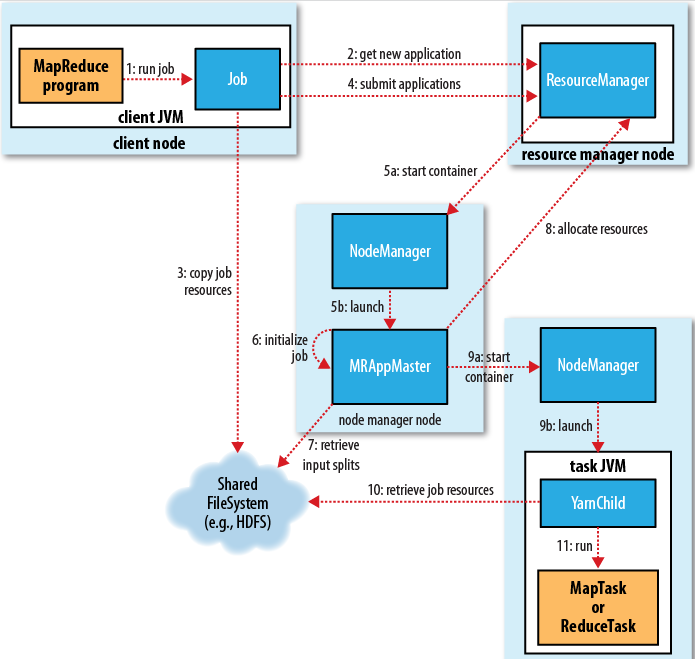
Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。它和现有的分布式文件系统有很多共同点。但同时，它和其他的分布式文件系统的区别也是很明显的。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束，来实现流式读取文件系统数据的目的。HDFS采用master/slave架构。一个HDFS集群是由一个Namenode和一定数目的Datanodes组成。Namenode是一个中心服务器，负责管理文件系统的名字空间(namespace)以及客户端对文件的访问。集群中的Datanode一般是一个节点一个，负责管理它所在节点上的存储。HDFS暴露了文件系统的名字空间，用户能够以文件的形式在上面存储数据。从内部看，一个文件其实被分成一个或多个数据块，这些块存储在一组Datanode上。Namenode执行文件系统的名字空间操作，比如打开、关闭、重命名文件或目录。它也负责确定数据块到具体Datanode节点的映射。Datanode负责处理文件系统客户端的读写请求。在Namenode的统一调度下进行数据块的创建、删除和复制。



### MapReduce

Hadoop Map/Reduce是一个使用简易的软件框架，基于它写出来的应用程序能够运行在由上千个商用机器组成的大型集群上，并以一种可靠容错的方式并行处理上T级别的数据集。一个Map/Reduce作业（job）通常会把输入的数据集切分为若干独立的数据块，由 map任务（task）以完全并行的方式处理它们。框架会对map的输出先进行排序， 然后把结果输入给reduce任务。通常作业的输入和输出都会被存储在文件系统中。 整个框架负责任务的调度和监控，以及重新执行已经失败的任务。通常，Map/Reduce框架和[分布式文件系统](http://hadoop.apache.org/docs/r1.0.4/cn/hdfs_design.html)是运行在一组相同的节点上的，也就是说，计算节点和存储节点通常在一起。这种配置允许框架在那些已经存好数据的节点上高效地调度任务，这可以使整个集群的网络带宽被非常高效地利用。Map/Reduce框架由一个单独的master JobTracker和每个集群节点一个slave TaskTracker共同组成。master负责调度构成一个作业的所有任务，这些任务分布在不同的slave上，master监控它们的执行，重新执行已经失败的任务。而slave仅负责执行由master指派的任务。

应用程序至少应该指明输入/输出的位置（路径），并通过实现合适的接口或抽象类提供map和reduce函数。再加上其他作业的参数，就构成了作业配置（job configuration）。然后，Hadoop的 job client提交作业（jar包/可执行程序等）和配置信息给JobTracker，后者负责分发这些软件和配置信息给slave、调度任务并监控它们的执行，同时提供状态和诊断信息给job-client。



### YARN

Yarn是hadoop集群资源管理器系统；从整体上还是属于master/slave模型，主要依赖于三个组件来实现功能，第一个就是ResourceManager，是集群资源的仲裁者，它包括两部分：一个是可插拔式的调度Scheduler，一个是ApplicationManager，用于管理集群中的用户作业。第二个是每个节点上的NodeManager，管理该节点上的用户作业和工作流，也会不断发送自己Container使用情况给ResourceManager。第三个组件是ApplicationMaster，用户作业生命周期的管理者它的主要功能就是向ResourceManager（全局的）申请计算资源（Containers）并且和NodeManager交互来执行和监控具体的task。架构图如下

## Kafka实时消息通信系统

### 概述

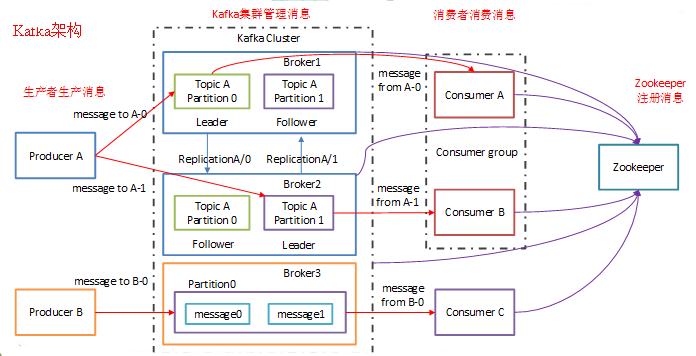
Kafka是最初由Linkedin公司开发，是一个分布式、分区的、多副本的、多订阅者，基于zookeeper协调的分布式日志系统（也可以当做MQ系统），常见可以用于web/nginx日志、访问日志，消息服务等；Kafka用于构建实时的数据管道和流式的app.它可以水平扩展，高可用，速度快。



### 系统架构

Kafka有四个核心API：

* [生产者 API](http://kafka.apache.org/documentation.html#producerapi)：允许应用程序发布记录流至一个或多个Kafka的话题(Topics)。
* [消费者API](http://kafka.apache.org/documentation.html#consumerapi)：允许应用程序订阅一个或多个主题，并处理这些主题接收到的记录流。
* [Streams API](http://kafka.apache.org/documentation/streams)：允许应用程序充当流处理器（stream processor），从一个或多个主题获取输入流，并生产一个输出流至一个或多个的主题，能够有效地变换输入流为输出流。
* [Connector API](http://kafka.apache.org/documentation.html#connect)：允许构建和运行可重用的生产者或消费者，能够把 Kafka主题连接到现有的应用程序或数据系统

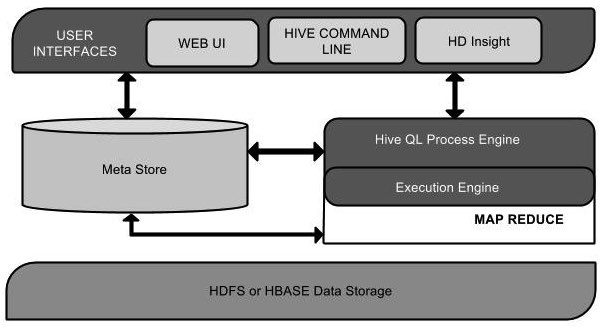


## HIVE数据仓库

### 概述

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SQL查询功能。它架构在Hadoop之上，总归为大数据，并使得查询和分析方便。

### 系统架构



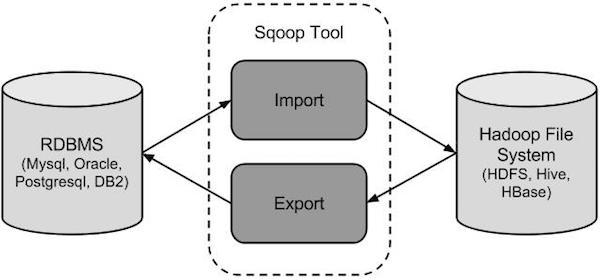
## SQOOP数据迁移工具

### 概述

SQOOP主要用于在Hadoop(Hive)与传统的数据库(mysql、postgresql等)间进行数据的传递，可以将一个[关系型数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93/8999831)*（例如：MySQL ,Oracle ,Postgres等）*中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

### 架构

Sqoop 架构是非常简单的，它主要由三个部分组成：Sqoop client、HDFS/HBase/Hive、Database。下面我们来看一下 Sqoop 的架构图。

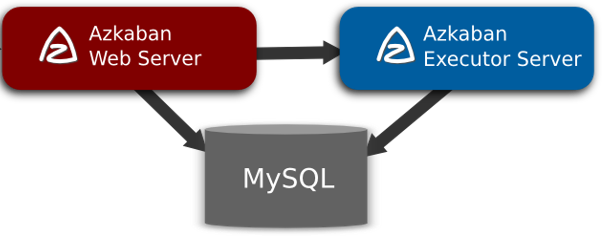


## Azkaban工作流任务调度系统

### 概述

Azkaban是由Linkedin公司推出的一个批量工作流任务调度器，主要用于在一个工作流内以一个特定的顺序运行一组工作和流程，它的配置是通过简单的key:value对的方式，通过配置中的dependencies 来设置依赖关系。Azkaban使用job配置文件建立任务之间的依赖关系，并提供一个易于使用的web用户界面维护和跟踪你的工作流。

### 系统架构



Azkaban Web Server：是整个Azkaban工作流系统的主要管理者，它用户登录认证、负责project管理、定时执行工作流、跟踪工作流执行进度等一系列任务

Azkaban Executor Server:负责具体的工作流的提交、执行，它们通过mysql数据库来协调任务的执行。

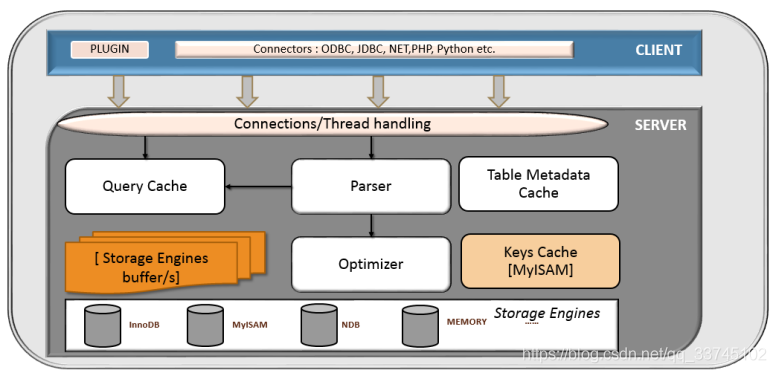
MYSQL:存储大部分执行流状态，AzkabanWebServer和AzkabanExecutorServer都需要访问数据库。

## MYSQL数据库

### 概述

数据库有很多种，最常用的比如 Oracle 10g、DB2、PostgreSQL 等，但最最常用的是我们目前选用的中量级数据库软件 MySQL。 MySQL 是一款非常著名并且流行的开源数据库软件，它部署方便，操作简单，对系统要求不高，它是集多实例、主从复制以及用户安全保障等优点于一身的中量级数据库，特别适用于项目上使用。

### 系统架构



* **Client :**  
  提供连接MySQL服务器功能的常用工具集
* **Server :**  
  MySQL实例，真正提供数据存储和数据处理功能的MySQL服务器进程
* **mysqld:**  
  MySQL服务器守护程序，在后台运行。它管理着客户端请求。mysqld是一个多线程的进程，允许多个会话连接，端口监听连接，管理MySQL实例
* **MySQL memory allocation:**  
  MySQL的要求的内存空间是动态的，比如 innodb\_buffer\_pool\_size (from 5.7.5), key\_buffer\_size。每个会话都有独一无二的执行计划，我们只能共享同一会话域内的数据集。
* **SESSION**  
  为每个客户端连接分配一个会话，动态分配和回收。用于查询处理，每个会话同时具备一个缓冲区。每个会话是作为一个线程执行的
* **Parser**  
  检测SQL语句语法，为每条SQL语句生成SQL\_ID，用户认证也发生在这个阶段
* **Optimizer**  
  创造一个有效率的执行计划（根据具体的存储引擎）。它将会重写查询语句。比如：InnoDB有共享缓冲区，所以，优化器会首先从预先缓存的数据中提取。使用 table statistics optimizer将会为SQL查询生成一个执行计划。用户权限检查也发生在这个阶段。
* **Metadata cache**  
  缓存对象元信息和统计信息
* **Query cache**  
  共享在内存中的完全一样的查询语句。如果完全相同的查询在缓存命中，MySQL服务器会直接从缓存中去检索结果。缓存是会话间共享的，所以为一个客户生成的结果集也能为另一个客户所用。查询缓存基于SQL\_ID。将SELECT语句写入视图就是查询缓存最好的例子。
* **key cache**  
  缓存表索引。MySQL keys是索引。如果索引数据量小，它将缓存索引结构和叶子节点（存储索引数据）。如果索引很大，它只会缓存索引结构，通常供**MyISAM存储引擎**使用。

## 报表后台管理系统

### 概述

该系统主要用于呈现给运营查看具体的报表信息；数据来源主要通过数据采集日志通过ETL数据清理分析后生成了业务上需求的报表数据。

### 系统架构



三层是表述层（WEB层）、业务逻辑层（Business Logic），以及数据访问层（Data Access）

**·WEB层**：包含JSP和Servlet等与WEB相关的内容；

**·业务层**：业务层中不包含JavaWeb API，它只关心业务逻辑；

**·数据层**：封装了对数据库的访问细节；