**大数据平台架构评审**

作者：方佩伟

日期：2015-3-9

修订记录

| 日期 | 版本号 | 描述 | 作者 | 审核人 | 批准人 | 批准日期 |
| --- | --- | --- | --- | --- | --- | --- |
| 2015-3-9 | 0.1 | 初稿 | 方佩伟 |  |  |  |
| 2015-3-16 | 0.1 | 根据项目组意见修改 | 方佩伟 |  |  |  |
|  |  |  | 方佩伟 |  |  |  |

# 背景——数据仓库现状

南航的数据仓库存储经历了从oracle到具有mpp架构的teradata一体机的阶段，目前正准备迈入大数据处理平台的阶段。

DSS决策支持系统在2012年1.0项目中申请了一台数据仓库一体机，这台服务器现有2 个节点，总用户空间9TB，目前剩余约1.2TB左右，每月增0.3TB。2014年又增加了4家GDS的MIDT，未来还会引入BIDT，航线成本等数据，存储空间已经成为制约系统发展的瓶颈。为此计划增加2个节点，扩容到20TB的用户空间。

除了存储空间紧张的严峻问题之外，一体机的数据仓库方案还存在存储成本昂贵的问题。

随着数据量的日益增长，眼前的架构方式必将给公司带来沉重的存储成本负担，一体机的扩容并不是长远之计。

随着大数据时代的来临，越来越多的企业开始关注大数据。Gartner最新数据显示：用户在大数据领域的投资将会持续增长；64%的企业正在考虑大数据项目，社交媒体、银行业以及服务行业是目前大数据投资的领先行业；而交通、医疗以及保险等行业则正在积极准备大数据投资。随着数据分析能力的提高，营销人将能更快速高效成本低廉的使用大数据。未来三年，71%的营销将开始使用大数据营销。

关于大数据，摆在南航面前的有两个问题：一是南航如何利用好大数据，给企业带来效益？二是南航的大数据平台该如何建设，采用什么架构？

# 二、什么是大数据

1. 大数据是指无法在一定时间内用传统数据库软件工具对其内容进行抓取、管理和处理的数据集合。PB是大数据层次的临界点： KB->MB->GB->TB->**PB**->EB->ZB->YB->NB->DB。大数据平台除了要解决当前一体机存储空间有限的问题之外，从长远上考虑，随着节点数据的增加，也将支持到PB级别的数据处理。

2. 实时数据分析是大数据平台有别于传统数据库的显著特征。

3. 大数据平台不仅仅能够处理结构化数据，也能处理半结构化数据，比如日志数据。

传统与大数据时代下的数据管理特点对比

|  |  |  |
| --- | --- | --- |
|  | 传统的信息管理 | 大数据时代下的信息管理 |
| 数据获得渠道 | 1. 数据大多是从内部环境中获得 | 1. 数据收集具有实时性，包括了内部数据和外部数据 |
| 数据供给 | 2. 数据提供给高层人员做决策使用 | 2. 数据提供给涉及的人员使用 |
| 数据使用期限 | 3. 数据用于长期的计划 | 3. 可以用于短期的决策支持 |
| 数据作用 | 4. 用于战略指导 | 4. 可以用于战术上的指导 |

# 三、关键需求

## 3.1 非功能性需求

1. 空间需求，通过搭建大数据平台，解决目前数据仓库空间紧张问题，至少解决TB级别的数据存储需要。

2. 扩展性需求，要求平台根据需要能够很方便的进行扩容，满足数据日益增长的需要。

3. 可靠性需求，要求平台在任一数据节点宕机的情况下，不影响数据的完整性。

4. 实时性需求

## 3.2功能性需求

1. 具备传统的数据仓库的数据获取，数据导入，数据处理，数据存储，数据查询，数据分析功能

# 四、南航大数据平台建设目标

数据正在迅速膨胀并变大，它决定着企业的未来发展，虽然很多企业可能并没有意识到数据爆炸性增长带来问题的隐患，但是随着时间的推移，人们将越来越多的意识到数据作为资产对企业的重要性。

* 建设大数据平台，整合数据资产
* 引入管研运研及其他数据
* 弥补现有teradata一体机空间不足劣势
* 实现大数据平台的实时查询
* 为企业内部提供数据分析支持。

# 五、南航大数据平台建设规划

大数据平台建设划分为三个阶段：概念验证阶段，生产应用阶段，hadoop即服务阶段。

1. 概念验证阶段，节点数为10个节点以下，需要完成的任务有，探索学习和使用平台上的各个功能组件，通过实践掌握各个功能组件最适合的应用场景，引入少数几个管研和运研的数据存储到大数据平台，逐步完善数据处理流程。开发南航自己大数据平台的任务调度系统。

2. 生产应用阶段，节点数为100节点以下，全方面引入南航企业内部数据和外部数据。为企业内部提供数据分析支持。

3. hadoop即服务阶段，节点数为100节点以上，将大数据平台作为服务，用户可以自行导入数据，分析数据和报表展示。

# 六、硬件配置

目前我们申请了5台B档PC服务器用来搭建初期的大数据平台，每台服务器有96G内存和5TB的硬盘空间。每台服务器价格为4万元。

参考业界某公司大数据集群的服务器配置如下：30台机器，4T\*12 SAS硬盘，64G内存，2\*6\*2 个cpu内核，他们使用的是社区版本的hadoop2.4， Yarn，MapReduce2, Hbase(0.98), Hive(0.14)。

# 七、物理架构图

前期我们考虑使用一台服务器作为主节点（同时用于监控管理和数据接口服务器），另外三台作为数据节点，剩下的一台作为备份节点，用于后续研究解决主节点的单点故障问题。



# 八、技术架构

我们采用的是Cloudera的开源Apache Hadoop发行版，亦即（Cloudera Distribution including Apache Hadoop，CDH）。选用最新版本的CDH5.3。选择用Cloudera的原因如下：

1. CDH基于稳定版Apache Hadoop，并应用了最新Bug修复或者Feature的Patch。在实际使用过程中CDH表现无比稳定，并没有引入新的问题。
2. 安装维护方便，节省运维时间，通过cloudera manager的gui方式管理集群
3. 整合了hadoop生态系统各个功能组件，包括：hdfs，MapReduce，hive，yarn，impala，sqoop，spark，hbase，solr。省去自己安装配置的麻烦和风险。

# 九、大数据平台架构

 大数据平台按层次划分可以分为源数据层、数据获取层、数据导入层、数据加工层、数据核心存储层、数据分析处理层、数据服务存储层、应用层和服务层。

大数据平台架构主要分四个部分：**离线计算，流式计算，实时分析和实时查询**。

## 9.1源数据层

大数据平台的接入数据包括了原有的teradata一体机数据库里的商业数据，管研和运研的数据，合作第三方数据，政府或公共部门信息网站，社交媒体网站的数据。

平台建设初期我们只考虑接入少数几个管研和运研的数据。

数据主要以文本的方式提供，在主节点搭建ftp，把文本文件上传到主节点，并用自己开发的java工具以文件触发的方式加载数据到大数据平台。

## 9.2数据获取

在hadoop的生态系统中，sqoop组件可以很方便的实现从各种关系型数据库抽取数据并将数据存储到hadoop平台，但是受限于公司的政策，我们不能直接对数据库进行抽取操作。所以数据只能以文本方式从原系统中导出。

分布式网络爬虫是后续我们要研究的方向之一，用于在外网爬取各种有价值的数据并导入到我们的大数据平台

## 9.3数据核心存储

**HDFS**分布式文件系统，有着高容错性（fault-tolerant）的特点，并且设计用来部署在低廉的（low-cost）硬件上。而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。在hdfs中，文件被划分为各个小块，分散到集群的各个节点中。我们可以设置文件数据块的副本数量，默认值是3，来减少因为节点损坏或者网络故障导致的文件丢失的可能性。

应用场景： 我们可以利用hdfs来存储一些数据的源文件，日志文件，以及价值密度较低的文件。

**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。缺点是Hive不能提供实时的查询和基于行级的数据更新操作。

Hive 构建在基于静态批处理的Hadoop 之上，Hadoop 通常都有较高的延迟并且在作业提交和调度的时候需要大量的开销。因此，Hive 并不能够在大规模数据集上实现低延迟快速的查询，例如，Hive 在几百MB 的数据集上执行查询一般有分钟级的时间延迟。因此，

Hive 并不适合那些需要低延迟的应用，例如，联机事务处理（OLTP）。Hive 查询操作过程严格遵守Hadoop MapReduce 的作业执行模型，Hive 将用户的HiveQL 语句通过解释器转换为MapReduce 作业提交到Hadoop 集群上，Hadoop 监控作业执行过程，然后返回作业执行结果给用户。Hive 并非为联机事务处理而设计，Hive 并不提供实时的查询和基于行级的数据更新操作。Hive 的最佳使用场合是大数据集的批处理作业，例如，网络日志分析。

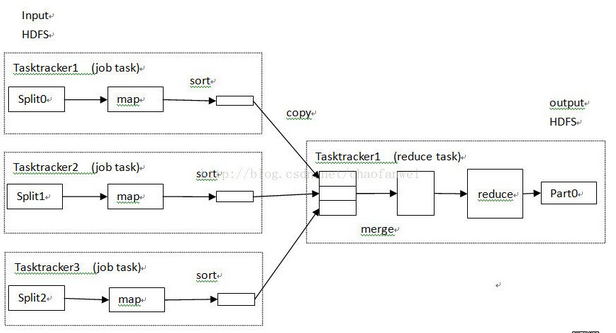
应用场景： 原本存储在teradata一体机的实时性要求不要的的数据就可以存储在hive数据仓库中。以此解决一体机空间紧张的问题。

**HBase**是Hadoop生态系统中继Cassandra之后成为NoSQL数据库的，从设计之初就十分注重系统的扩展性，对集群的动态扩展、负载均衡、容错、数据恢复等都有充分的考虑。相比于传统关系型数据库，HBase更适合于数据量大、读写吞吐量非常高、对数据可靠性一致性及数据操作的事务性要求较低的应用。

应用场景： 像中国移动等运营商就把HBase用于话单数据存储和查询，我们可以根据需要决定哪些数据存放在HBase中。

## 9.4离线计算

**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。



Mapreduce的开发需要开发需要用到java编程，不大适用于数据仓库分析人员。我们的愿景是，尽可能让数据分析人员通过sql语言使用大数据平台，对存储在hive数据仓库中的数据进行分析处理。

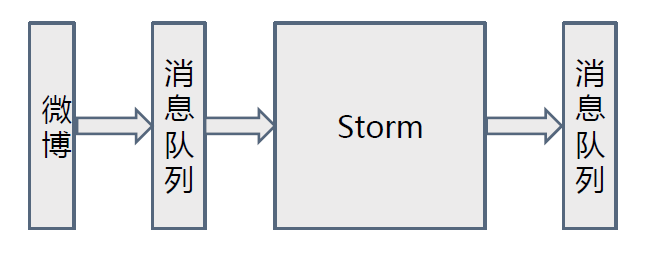
应用场景： 半结构化和非结构化的数据处理。

## 9.5流式计算

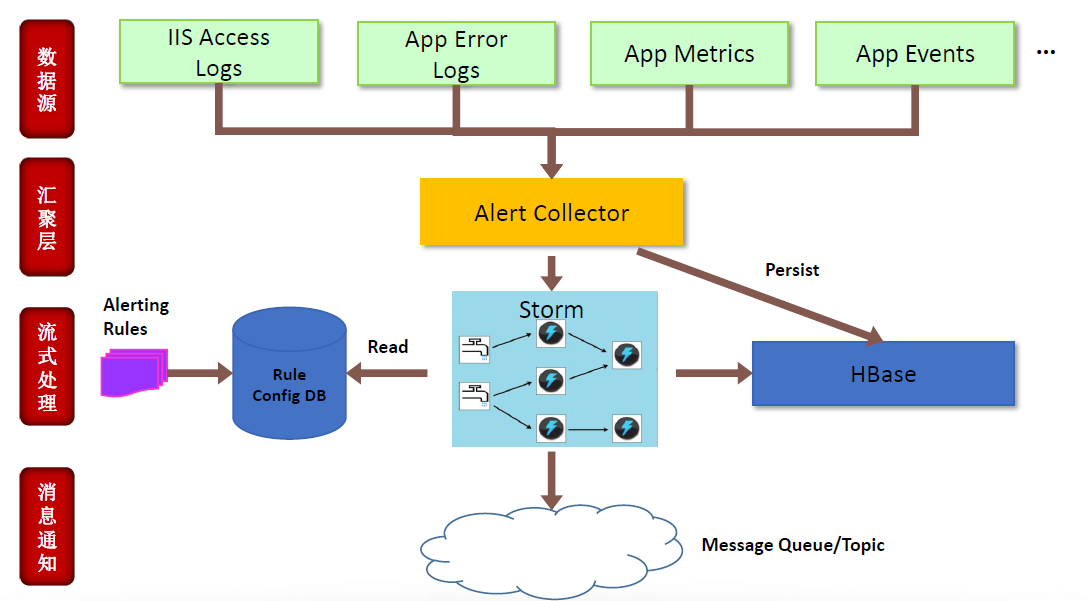
业界中使用较多的是Storm用来作为hadoop的流处理解决方案。Storm是一个分布式的、容错的实时计算系统。Storm可以方便地在一个计算机集群中编写与扩展复杂的实时计算，Storm之于实时处理，就好比Hadoop之于批处理。Storm保证每个消息都会得到处理，而且它很快——在一个小集群中，每秒可以处理数以百万计的消息。可以使用任意编程语言来做开发。

下面我们通过两个简单的案例来了解什么是流式计算：

一个是通过消息队列实时接收微博信息，根据关键字过滤消息的流式计算应用场景。



另一个是一个监控报警系统，通过日志收集器，获取系统异常情况，并存储到Hbase中，Storm则根据匹配的规则决定是否触发监控报警，并将消息发送到队列中。



在大数据平台架构中，我们使用spark streaming用来替代Storm组件。Spark Streaming是大规模流式数据处理的新贵，将流式计算分解成一系列短小的批处理作业。这里的批处理引擎是Spark，也就是把Spark Streaming的输入数据按照batch size（如1秒）分成一段一段的数据（Discretized Stream），每一段数据都转换成Spark中的RDD（Resilient Distributed Dataset），然后将Spark Streaming中对DStream的Transformation操作变为针对Spark中对RDD的Transformation操作，将RDD经 过操作变成中间结果保存在内存中。整个流式计算根据业务的需求可以对中间的结果进行叠加，或者存储到外部设备。

在互联网应用中，网站流量统计作为一种常用的应用模式，需要在不同粒度上对不同数据进行统计，既有实时性的需求，又需要涉及到聚合、去重、连接等较为复杂的统计需求。传统上，若是使用Hadoop MapReduce框架，虽然可以容易地实现较为复杂的统计需求，但实时性却无法得到保证；反之若是采用Storm这样的流式框架，实时性虽可以得到保证，但需求的实现复杂度也大大提高了。Spark Streaming在两者之间找到了一个平衡点，能够以准实时的方式容易地实现较为复杂的统计需求。 下面介绍一下使用Kafka和Spark Streaming搭建实时流量统计框架。

* 数据暂存：Kafka作为分布式消息队列，既有非常优秀的吞吐量，又有较高的可靠性和扩展性，在这里采用Kafka作为日志传递中间件来接收日志，抓取客户端发送的流量日志，同时接受Spark Streaming的请求，将流量日志按序发送给Spark Streaming集群。
* 数据处理：将Spark Streaming集群与Kafka集群对接，Spark Streaming从Kafka集群中获取流量日志并进行处理。Spark Streaming会实时地从Kafka集群中获取数据并将其存储在内部的可用内存空间中。当每一个batch窗口到来时，便对这些数据进行处理。
* 结果存储：为了便于前端展示和页面请求，处理得到的结果将写入到数据库中。

相比于传统的处理框架，Kafka+Spark Streaming的架构有以下几个优点。

* Spark框架的高效和低延迟保证了Spark Streaming操作的准实时性。
* 利用Spark框架提供的丰富API和高灵活性，可以精简地写出较为复杂的算法。
* 编程模型的高度一致使得上手Spark Streaming相当容易，同时也可以保证业务逻辑在实时处理和批处理上的复用。

在基于Kafka+Spark Streaming的流量统计应用运行过程中，有时会遇到内存不足、GC阻塞等各种问题

## 9.6实时分析

Spark是UC Berkeley AMP lab所开源的类Hadoop MapReduce的通用的并行计算框架，Spark基于map reduce算法实现的分布式计算，拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是Job中间输出和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。

提到Spark，我们不得不说一下BDAS（Berkeley Data Analytics Stack），这个伯克利大学提出的关于数据分析的软件栈。从它的视角来看，目前的大数据处理可以分为如以下三个类型。

* + 复杂的批量数据处理（batch data processing），通常的时间跨度在数十分钟到数小时之间。
  + 基于历史数据的交互式查询（interactive query），通常的时间跨度在数十秒到数分钟之间。
  + 基于实时数据流的数据处理（streaming data processing），通常的时间跨度在数百毫秒到数秒之间。

目前已有很多相对成熟的开源软件来处理以上三种情景，我们可以利用MapReduce来进行批量数据处理，可以用Impala来进行交互式查询，对于流式数 据处理，我们可以采用Storm。对于大多数互联网公司来说，一般都会同时遇到以上三种情景，那么在使用的过程中这些公司可能会遇到如下的不便。

* + 三种情景的输入输出数据无法无缝共享，需要进行格式相互转换。
  + 每一个开源软件都需要一个开发和维护团队，提高了成本。
  + 在同一个集群中对各个系统协调资源分配比较困难。

BDAS 就是以Spark为基础的一套软件栈，利用基于内存的通用计算模型将以上三种情景一网打尽，同时支持Batch、Interactive、 Streaming的处理，且兼容支持HDFS和S3等分布式文件系统，可以部署在YARN和Mesos等流行的集群资源管理器之上。Spark可以替代MapReduce进行批处理，利用其基于内存的特点，特别擅长迭代式和交互式数据处理；Shark处理大规模数据的 SQL查询，兼容Hive的HQL。

应用场景：适用于基于内存的快速的数据分析工作。

## 9.7实时查询

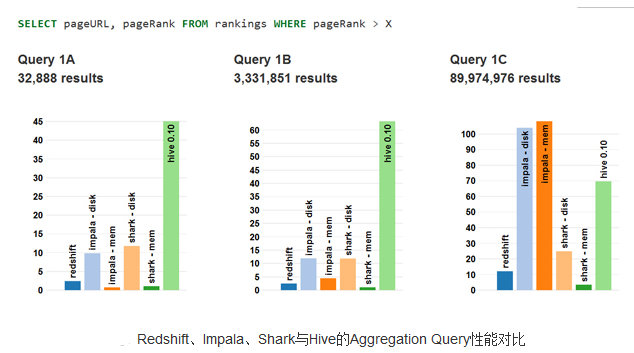
**Impala**是Cloudera公司主导开发的新型查询系统，它提供SQL 语义，能查询存储在Hadoop的HDFS和HBase中的PB级大数据。已有的Hive系统虽然也提供了SQL语义，但由于Hive底层执行使用的是 MapReduce引擎，仍然是一个批处理过程，难以满足查询的交互性。相比之下，Impala的最大特点也是最大卖点就是它的快速。

在Cloudera的测试中，Impala的查询效率比Hive有数量级的提升。从技术角度上来看，Impala之所以能有好的性能，主要有以下几方面的原因。

* Impala不需要把中间结果写入磁盘，省掉了大量的I/O开销。
* 省掉了MapReduce作业启动的开销。MapReduce启动task的速度很慢（默认每个心跳间隔是3秒钟），Impala直接通过相应的服务进程来进行作业调度，速度快了很多。
* Impala完全抛弃了MapReduce这个不太适合做SQL查询的范式，而是像Dremel一样借鉴了MPP并行数据库的思想另起炉灶，因此可做更多的查询优化，从而省掉不必要的shuffle、sort等开销。
* 通过使用LLVM来统一编译运行时代码，避免了为支持通用编译而带来的不必要开销。
* 用C++实现，做了很多有针对性的硬件优化，例如使用SSE指令。
* 使用了支持Data locality的I/O调度机制，尽可能地将数据和计算分配在同一台机器上进行，减少了网络开销。

虽然Impala是参照Dremel来实现的，但它也有一些自己的特色，例如Impala不仅支持Parquet格式，同时也可以直接处理文本、 SequenceFile等Hadoop中常用的文件格式。另外一个更关键的地方在于，Impala是开源的，再加上Cloudera在Hadoop领域 的领导地位，其生态圈有很大可能会在将来快速成长。

跟spark相比之下，Impala由于缺失足够强大的容错机制，其上运行的任务一旦失败就必须“从头来过”，这样的设计必然会在性能上有所缺失。而且Shark是把内存当作第一类的存储介质来做的系统设计，所以在处理速度上也会有一些优势。



 impala能使用的内存无法超过系统的硬件可用内存（GA版，查询需要的内存如果超出硬件内存，则查询将失败），对内存要求高，典型的硬件内存为：32~48G

## 9.8任务调度系统

规划中的任务调度系统应该是具备监控页面和完善的作业调度机制。在初期建设中我们采用简单的java程序写的小工具，以文件触发的方式调度作业，同时具备作业运行日志记录功能。

# 十、风险评估

南航的大数据平台建设目前属于探索建设的阶段，是一个从无到有的过程，技术架构上存在需要实践检验和修正的地方。但是，我们可以通过借鉴业界已有的成熟的方案，结合自己的实际需要，建设适合南航自己的大数据平台。

另一方面，虽然缺乏外面咨询公司的协助，但是顺应公司的政策方向，转变工作作风，去掉厂商的拐杖，有利于培养起南航内部的大数据方面的人才力量。

通过使用Cloudera的hadoop版本，避免了配置错误和管理繁杂上的风险。

由于架构中只有一台名称节点，存在单点故障，后续需要研究探索解决这个问题。