

融合通信系统中分布式存储引擎的研究^①

史坤鹏^{1,2}, 贾正锋²

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

摘要: 融合通信是当今计算机应用领域研究热点之一, 人们对融合通信系统中应用服务的要求也越来越高. 在数据存取方面, 基于传统关系型数据库或者基于传统文件系统的存储方式已经越来越不能满足应用的需求. 随着 Hadoop 技术以及相关子系统的发展, 分布式存储的优势日渐明显. 因此, 本文在分析 HBase、Hive 各自特点及其体系结构的基础上, 结合融合通信具体项目提出了基于 HBase-Hive 集成设计的存储引擎设计方法, 以此来解决融合通信系统中数据安全性、数据获取效率等方面不满足的情况. 通过对比实验表明, 该设计方案提高系统数据查询获取效率, 也为后续数据挖掘方面的开发做好准备.

关键词: 融合通信; 分布式存储; HBase; Hive

Research on Distributed Storage Engine in Unified Communications Systems

SHI Kun-Peng^{1,2}, JIA Zheng-Feng²

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: Unified Communications is one of the research hotspots in today's computer applications, and people's requirements of application service in unified communications system applications and services are increasingly higher. In terms of data access, based on the traditional relational database or file system-based storage, traditional way has become increasingly unable to meet the needs of the application. With the development of Hadoop technology and related subsystems, the advantages of distributed storage become obvious. Therefore, on the basis of analyzing respective characteristics and system architectures of HBase and Hive, this article combines with specific projects of Unified Communications and proposes the design method of storage engine based HBase-Hive integrated design, in order to solve the unsatisfied problem of data security, data access efficiency etc. in Unified Communications systems. The contrast experiments show that the design scheme not only improves the efficiency in data query, but also makes fully prepare for the development of subsequent data mining.

Key words: unified communications; distributed storage; HBase; Hive

引言

随着互联网、移动互联网的飞速发展, 人们的通信方式也发生了深刻的变革, 融合通信以及越来越多的融合通信产品也慢慢为人们所熟知, 与此同时, 对数据的存储和处理提出革命性的需求, 存储基础架构已经成为 IT 核心之一. 从项目实际的角度来讲, 在融合通信系统中, 需要就消息、通话、文档、通讯录、组织架构等关键数据提供完善的存储引擎机制. 传统的基于关系数据库或者基于传统文件系统的存储方式,

在存储数据安全性、获取数据效率以及后续数据挖掘与分析等方面都存在不满足的情况.

本文以沈阳计算所融合通信产品企发号为项目依托, 提出一套可行的基于 HBase/Hive 的分布式存储引擎设计方案.

1 融合通信

融合通信(Unified Communication), 旨在通过管理、集成各种通信媒介以统一的方式改善通信用户的

^① 收稿时间:2016-06-03;收到修改稿时间:2016-07-14 [doi:10.15888/j.cnki.csa.005618]

可用性和协调性。UC 诞生,很好的缓解了企业用户在分布式环境中所面临的各种设备,工具和媒体类型的负担。具体而言,在 UC 的背景下,一系列的通信工具都集成在一个方式,无论企业和个人都能够在一个实体来管理所有的通信,UC 部署通过 IP 通信技术,推动了在线应用和即时消息相关技术的进步。

1.1 融合通信中数据特点

在融合通信系统中,其核心目标是:让人们无论何时何地,都可以通过网络,获得图像、声音、视频等数据的自由通信。为此,在用户体验获得前所未有的突破时,数据特点也呈现出前所未有的多元化。

(1) 数据类型多种多样

目前,融合通信系统中的数据类型并不仅仅是以文本或是数字的类型存在,图片、音频、视频数据甚至是地理位置信息等,这些数据越来越多的出现在融合通信系统中,其中,个性化的数据占到了很大比例。

(2) 数据体量巨大

随着各种技术的发展,人们的通信轨迹都能够以数据的形式被记录下来,而这些数据将会以更大的数据体量来进行记录和储存。

(3) 响应速度要求高

为了获得更好的用户体验,用户在对数据进行存取时必须要求系统以更快的速度进行响应,这时后台数据的存储引擎就起到了至关重要的作用。

1.2 融合通信的发展前景

融合通信有效的将文本、语音、视频等数据与不同平台上的通信应用统一起来,使得通常使用的即时消息、电话会议、电子邮件、网络电话、办公协同等多种功能融合在一个统一的系统平台上实现,使得人们可以更加有效的进行信息交流,同时提高人们工作中信息传递的效率,也丰富了用户体验。融合通信具体发展前景可以概括为如图 1 所示。

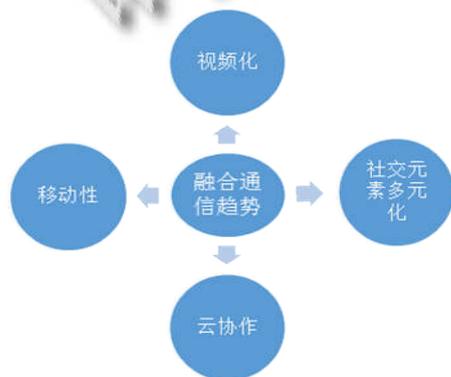


图 1 融合通信发展前景

社交元素多元化的手机通信录的融合通信产品是近期发展重点。以智能手机为载体、基于手机通信录的沟通类移动互联网应用型融合通信产品,可以通过 3G、Wi-Fi 等通道提供免费的通话/对讲、短信、IM、视频、状态以吸引用户,通过通信录的真实社会关系网以发展用户,通过用户和好友间的互相推荐以实现病毒式传播,正在成为移动互联网的“杀手级”业务^[3]。目前,在国内互联网应用市场上流行着很多融合通信的应用,如中国移动推出的云企信和云视讯、工作宝等应用。随着中国在 2015 年世界移动通信大会上推出基于融合通信的“三新手机”,并且由华为公司定义了 5G 网络架构,这将会大大刺激国内的融合通信业务的迅猛发展。

2 数据存储

2.1 关系模型

1970 年由 IBM 公司的一名研究员 E.F.Codd 博士首先提出关系模型,关系模型可以理解作为一种二维表格模型,而关系型数据库是根据二维表及其之间的联系所组成的一个数据组织。事务一致性、ACID、行式存储是其突出特点。比较典型的代表 MySQL、SQL Server 等。随着融合通信技术的不断发展、大数据时代的到来以及生活服务业务与 IT 日益密切的联系,传统基于关系模型的行式数据库正面临着一系列挑战,比如:成本、速度、整合、安全、能源管理等,越来越不能满足大数据应用的需求。

2.2 分布式存储

为解决传统行式数据所面临的问题,我们提出了分布式存储。而存储引擎更多的可以理解作为一种数据存、取的应用服务。分布式存储系统主要是通过大量的物理 I/O 设备组合成一个逻辑存储设备来提供存储的容量和性能。分布式存储系统架构如图 2 所示。

2.3 Hive/HBase

以 Hadoop 为平台的分布式存储系统,提供了一个稳定的共享存储和分析分布式系统,在云计算、云存储以及大数据领域得到了广泛的应用,存储由 HDFS 实现,分析由 MapReduce 实现。基于 Hadoop 平台开发的子系统 HBase,是一种类似于 BigTable 的分布式数据库,其底层存储建立在 HDFS 之上,可以提供高可靠性、高性能、基于列存储、可伸缩、实时读写的数据系统服务。HBase 是一种典型的非关系型

开源分布式数据存储系统,其主要特点面向列,包括面向列(族)的存储和权限控制,列(族)独立检索.针对每个查询,每个数据库集(至少是很大一部分)都会被处理.并且只能通过行健和行健范围进行检索,只支持单行事务,但是可以通过 Hive 支持来实现多表联合等复杂操作, HBase 在分布式集群上主要依靠由 HRegion、HMaster、HClient 组成的体系结构从整体上管理数据,克服了传统关系型数据库对融合通信系统中非结构化数据存储性能差的缺点,具体其物理结构图如图 3 所示.

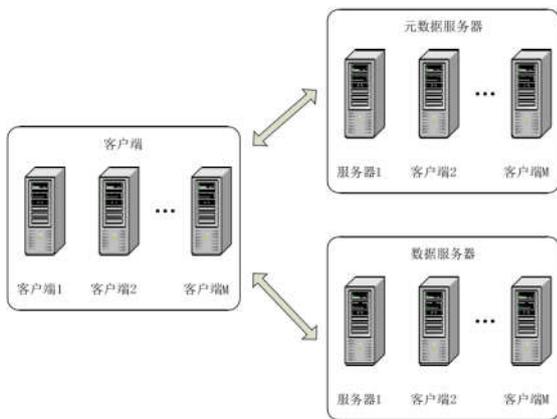


图 2 分布式存储系统架构

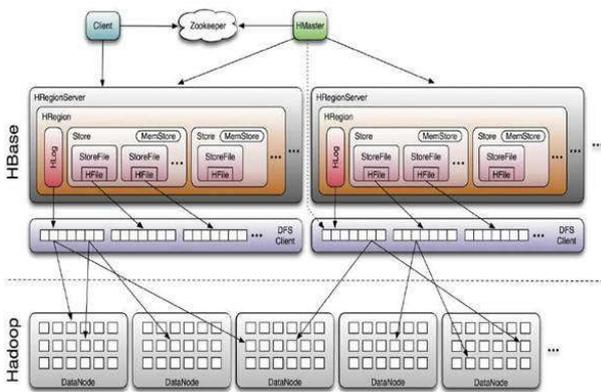


图 3 HBase 物理结构

Facebook 开发的 Hive 数据仓库应用工具构建于 Hadoop 集群之上,提供类 SQL 查询接口,本质是将命令行查询转换为相应的 Map Reduce 程序,这使得我们更容易学习掌握的 Hadoop.基于本文的研究内容,结合融合通信项目中所采用的 Hive 体系架构,得出 Hive/Hadoop 架构如图 4 所示.

由图中可以看出 Hive 数据访问客户端有三种方式.方式一:是大家最经常用的方式,采用命令行接

口 Shell/CLI 直接访问 Hive;方式二:采用 JDBC 或 ODBC 连接 Hive 的 Trift 服务器的方法进行数据访问;方式三:用 HWI(Hive Web Interface)通过浏览器访问 Hive.以上三种方式其最根本的特点还是 Hive 在做数据读取时所采用的 MapReduce 程序,这也是 Hive 具有很好的数据分析处理能力的原因.

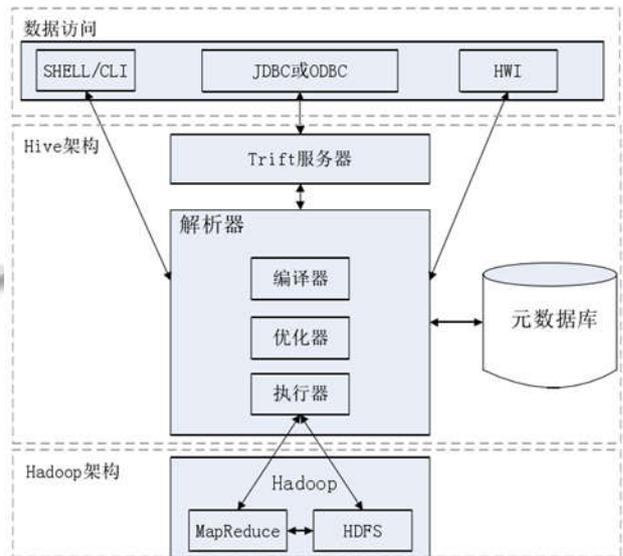


图 4 系统 Hive/Hadoop 架构图

3 分布式存储引擎设计

通过上述对融合通信系统中数据特点的总结以及 HBase、Hive 体系结构及其功能特点的分析,我们发现 HBase 更加侧重于查询的性能,这一优势能够给平台的 MapReduce 提供高效的存取支撑.但是 HBase 本身对数据的操作性能不足,若单独依靠 HBase 来管理日常数据,那么稍微复杂的数据操作都需要大量的 MapReduce 程序,这样就大大增加了数据管理成本. Hive 是被设计用来并行处理分析大规模数据,可以进行高级而且复杂的数据计算操作.但是出于体系结构简洁优化的角度考虑, Hive 忽略了对表的高效设计,对于表的遍历也会带来在数据处理上的负担. HBase 和 Hive 分别有各自的优缺点,而且两者在一些地方可以进行交互互补.为此,我们提出 HBase-Hive 整合集成的设计方案,形成一个分布式数据存储管理系统,以完成融合通信系统中存储引擎的相关功能,具体包括 HBase 的并行加载,集成设计以及系统的综合查询应用.

3.1 HBase 并行加载

HBase 被设计时主要是适用于大量数据的存储与

查询, 本身提供了几种不同的数据加载方法, 同时提供了加载数据的 API、供程序员使用. 在这里, 本文借助 Hadoop 中 MapReduce 并行框架, 通过调用 HBase 的接口 API 实现数据的并行加载. 加载操作可以直接在 Map 函数中完成, 无需编写 Reduce 函数. 具体流程大致如下: 首先是在 MapReduce 中 Run()方法中建立与 HBase 的连接, 然后在 Map()函数中调用 HBase 接口将数据加载到 HBase 中. 以下是 Map 核心代码.

```
public class MapHBaseLoad extends Configured
implements Tool{
public static class Map
extends Mapper<LongWritable,Text,NullWritable,NullWritable>{
Configuration conf = null;
HTable xtab = null;
private String tableName = "";
private Put put = null;
protected void setup(Context context) throws
IOException, InterruptedException{
super.setup(context);
conf = context.getConfiguration();
tableName = conf.get("tableName");
}
protected void map(LongWritable key ,Text value,Context context)
throws IOException , InterruptedException{
String all[] = value.toString().split(",");
put= new Put(Bytes.toBytes(all[0]));
put.add(Bytes.toBytes("att1"),Bytes.toBytes("20160531"),
Bytes.toBytes(all[1]));
}
protected void cleanup(Context context)
throws IOException , InterruptedException{
super.cleanup(context);
xtab.flushCommits();
xtab.close(0);
}
}
}
```

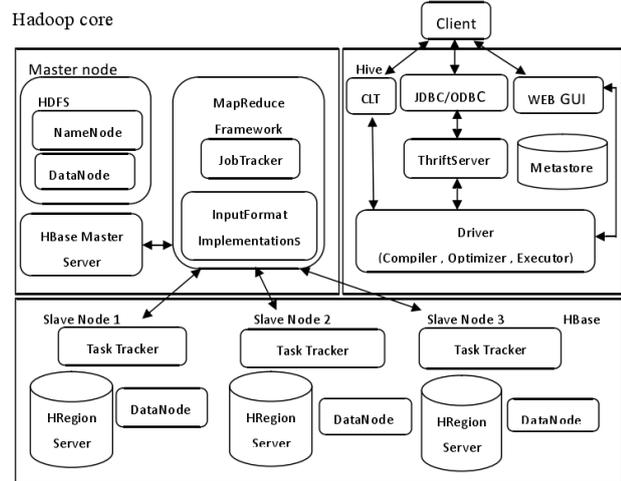


图 6 集成框架

3.3 Hbase-Hive 集成

本文根据 HBase 和 Hive 各自的优缺点, 将

HBase Map Reduce、HDFS 和 Hive 结合使用, 将以并行加载的 HBase 作为数据的存储端, 考虑到将来会对存储端数据做进一步的处理程序, 比如数据挖掘等, 因此将 Hive 作为查询端. 在这个过程中, HBase 可作为系统中的实时存储媒介, Hive 将解析的查询计划 (HQL 语句) 存储于 HDFS 中, 通过 MapReduce 访问存储在 HBase 中的数据, 执行查询操作, 以及表与表之间的关联操作, 具体的 HBase-Hive 集成框架图如图 6 所示.

为了更好地利用 HBase 行更新、列索引等存储特性, 需要在 Hive 中建立 HBase 和 Hive 的关联表, 也就是建立 HBase 表和 Hive 表之间的映射关系, 实际项目中所建立的映射关系如下所示.

```
create Table Hive.urls(id int, url String ,hit_count int ;)
STOREDBY 'org.apache.hadoop.hive.hbase.HbaseStorageHandler'
WITH SERDEPROPERTIES
("hbase.columns.mapping"=":key,u:url,s:Hits")
TBLPROPERTIES("Hbase.table.name"="hbase_urls")
```

3.4 系统查询

在系统中的 Hive 进程调用 MapReduce 函数查询 HBase 中存储的数据时, 其具体的数据流图如图 7 所示.

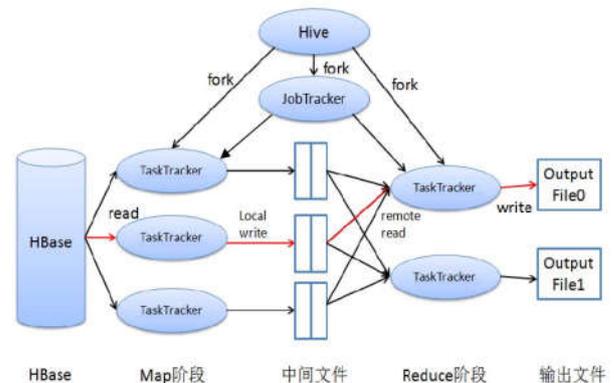


图 7 系统数据流图

具体的数据流主要步骤如下: 首先, Hive 进程向 Job Tracker 提交查询请求, Job Tracker 读取 HBase 中的数据, 同时获取数据存储的具体位置. 然后使用 MapReduce 运行机器集群中的处理程序, 具体实施包括一个 Map 阶段的作业追踪器 Job Tracker 和一组 Reduce 阶段的任务追踪器 Task Tacker. Job Tracker 分发不同的 Map 任务 或 Reduce 任务给处于不同阶段的 Task Tacker. Map 阶段的 Task Tacker 接受分配的 Map 任务后, 首先从 HBase 中读取数据, 并将读取结

果以键值对(key/value)的形式提交给 Map 函数,而 Map 函数处理后产生的中间文件的结果也为该形式.中间结果在缓存区中暂时存放,而且会被定期的写入硬盘.同时,Job Tracker 也会获得中间结果所处的本地硬盘位置信息,并将中间结果的位置信息传递给 Reduce 阶段的 Task Tacker,也就是分配了 Reduce 任务的 Task Tacker.该 Task Tacker 调用远程处理读取缓存中的中间数据,按照数据的 key 值对进行排序形成键值对列表.最后 Task Tacker 把所有的键值对列表、中间结果集合发送给 Reduce,并产生输出结果.最后当所有 Map/Reduce 任务完成后,作业追踪器激活 Hive 程序,回到最初的调用点.

4 性能测试

本文提出的面向融合通信的分布式存储引擎的设计已经在项目中的服务器中得以验证,为了验证本文提出的 HBase-Hive 整合应用具有较好的效率,我们通过设计实验将本系统与传统的基于关系模型的数据库作对比分析,具体的实验过程为:对实际融合通信系统中的人员信息表进行查询,并根据表中数据条目设置三组不同的数据量,通过在服务器端使用 wireshark 抓包,获取各自的查询耗时,定量分析不同条目数量的查询时间,从而在整体的效率上做出对比分析.

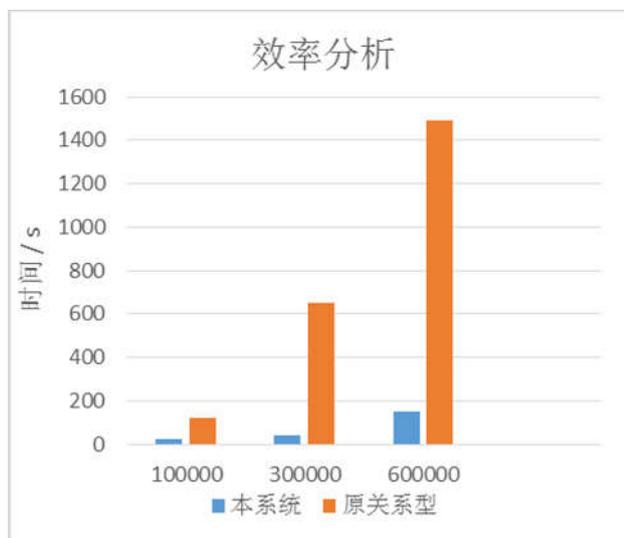


图8 耗时对比

根据图表中的信息可以看出,两种查询所消耗的时间与数据量的大小基本上是成正相关的.在实际的实验时,我们发现当表中数据量较少的时候,两种查

询所消耗时间相差并不大,甚至原基于关系模型的系统所消耗的时间会更少一些.但是随着表中数据量的增加,本系统基于 Hive-HBase 框架结构的查询能力的优势会表现的越来越明显.

5 结语

针对融合通信系统后台数据库实时性响应、数据安全性以及数据分析等方面的不足,本文结合融合通信系统中数据存储的具体特点,并详细分析了 HBase、Hive 各自的优缺点,提出了一种基于 HBase/Hive 整合集成的分布式存储方案,充分利用了 HBase 列式存储可以快速查询的优点,以及 Hive 数据处理的方面的优越性能,实践充分证明了系统设计的合理性,很好地解决了上述提出的问题.但是,由于自身知识储备不足等原因,系统中仍有可操作性差等不足之处,包括在数据挖掘方面仍大有可为,期待后续进一步的扩展开发.

参考文献

- 1 陆嘉恒.Hadoop 实战.北京:机械工业出版社,2012.
- 2 田秀霞,周耀君,毕忠勤.基于 hadoop 架构的分布式计算和存储技术及其应用.上海电力学院学报,2011.
- 3 姚群峰,张玉莹.电信运营商发展融合通信的战略思考.电信科学,2011.
- 4 高金标,何利力,邹云阳.基于分布式存储系统的 Hive 与 Hbase 的研究.工业控制计算机,2015.
- 5 Russom P. Big data analytics. TDWI Best Practices Report, Fourth Quarter, 2011.
- 6 George L. HBase: The definitive guide. O'Reilly Media, Inc, 2011.
- 7 Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system. Mass Storage Systems and Technologies (MSST) IEEE. 2010. 1-10.
- 8 黄贵,庄明强.Oceanbase 分布式存储引擎.华东师范大学学报,2014.
- 9 王伟.基于 Hive 的物流数据仓库研究与实现.东华大学, 2016.
- 10 张冬.网络存储系统原理精解与最佳实战.北京:清华大学出版社.
- 11 付东华.基于 HDFS 的海量分布式文件系统的研究与优化 [学位论文].北京:北京邮电大学,2012.