金钻芯大数据平台

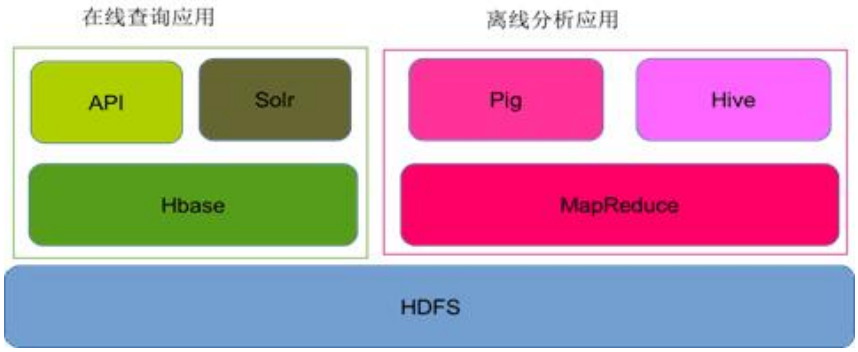
1 大数据平台的需求分析

1.1 现状描述

Hadoop 是一个由 Apache 基金会所开发的开源的分布式系统基础框架。目前，Hadoop 已成为大数据的代名词，是事实上的大数据标准。Hadoop 框架最核心的设计就是：分布式文件系统 HDFS 和分布式计算编程模型 MapReduce。HDFS 为海量的数据提供了存储，而 MapReduce 为海量的数据提供了计算。

图　Hadoop 架构

目前，业界普遍基于开源 Hadoop 的大数据解决方案框架，如图：



■ 基于 Hadoop 解决方案的**优点**：

高扩展性：Hadoop 采用物理服务器集簇方式部署，可以方便地横向扩展，实现存储空间的动态线性扩充。

高效性：充分利用集群的威力进行分布式高速运算，并能够在节点之间动态地移动数据。

高容错性：Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

低成本：利用中低端机架式服务器集群部署，无需采用小型机，无需建设大规模的集中存储。

■ 当前基于 Hadoop 解决方案的**缺点**：

1、数据没有统一视图，存储架构混乱。

2、MapReduce 应用场景受限，不适合低延迟数据访问，即适用于离线批处理统计，不适用于实时交互分析。

3、在线查询应用只能使用 Hbase，而 Hbase 只支持行键查询，使用场景单一。实际应用中会将 Solr 集成到方案中，和 Hbase 配合使用，但又造成了索引的额外存储（无法和 HBASE 存储在一起），这就就存在相互间关联的问题，势必造成时延。

4、HDFS 采用块存储方式，无法高效存储大量小文件。

5、HDFS 不支持多用户写入及任意修改文件。

6、开源系统的商业支持性差，实施复杂，无法快速构建，开源软件或多或少都存在一定的系统 BUG 或优化不足等问题，由此定会涉及意料之外的实施、管理和支持成本。

1.2 系统需求分析

最重要的现实是对大数据进行分析，只有通过分析才能获取很多智能的，深入的，有价值的信息。之前只是一直没有足够的基础设施和技术来对这些数据进行有价值的挖据。随着存储成本的不断下降、以及分析技术的不断进步，尤其是云计算的出现，不少公司已经发现了大数据的巨大价值：它们能揭示其他手段所看不到的新变化趋势，包括需求、供给和顾客习惯等等。比如，银行可以以此对自己的客户有更深入的了解，提供更有个性的定制化服务；银行和保险公司可以发现诈骗和骗保；零售企业更精确探知顾客需求变化，为不同的细分客户群体提供更有针对性的选择；制药企业可以以此为依据开发新药，详细追踪药物疗效，并监测潜在的副作用；安全公司则可以识别更具隐蔽性的攻击、入侵和违规。

这对企业和组织的信息分析和管理提出了挑战，传统的分析方法和处理能力已无法满足海量数据环境下的需求，主要表现在以下几方面：

- **高速海量数据的采集和存储变得困难**

一分钟内，Twitter 上新发的数据数超过 10 万;社交网络 Facebook 的浏览量超过 600 万.....我们正处于一个信息大爆炸的时代:宽带普及带来的巨量日志和通讯记录,社交网络每天不断更新的个人信息,视频通讯、医疗影像、地理信息、监控录像等视频记录,传感器、导航设备等非传统 IT 设备产生的数据信息,以及持续增加的各种智能终端产生的图片及信息,这些爆炸性增长的数据正在充斥整个网络。据权威市场调查机构 IDC 预测,未来每隔 18 个月,整个世界的数据总量就会翻倍;到 2020 年,整个世界的数据总量将会增长 44 倍,达到 35.2ZB(1ZB=10 亿 TB)。如此海量的数据规模的采集和存储，对传统的系统来说是不可完成的任务，这使得使用传统技术进行高速海量安全数据的采集和存储已不可行。

- **异构数据的存储和管理变得困难**

与之前的数据相比,互联网时代的大数据的数据结构更加多样化,图像、视频和文档的比例占据大半江山。数据显示,每年诸如邮件、视频、微博、帖子、手机呼叫、网页点击等类型的非结构化数据增长率就达 80%。而且这些数据里面

包含了很多有价值的信息。如果能有效地把它们的价值挖掘出来,这无疑会为企业带来巨大的经济效益。但是这都是建立在如何存储和管理的基础上，后期再进行分析。

- **孤立少量的数据分析价值较小**

其实也就是说孤证不立，如果想通过数据分析得出某种结论，那么待分析的数据样本一定要够完整丰富，否则无法得到比较正确的结论。

- **对海量数据的检索能力很弱**

传统的系统通过对关系型数据库的查询来实现业务的开展，随着数据量增大，查询效率变得非常低，查询延时，再加上组合查询条件，查询效率和时延无法满足现在需求。

系统相互独立，协同工作困难

现代企业和组织已建设了大量的信息基础设施和系统，系统之间相互独立，各自完成不同的功能，条块分割，各管一摊的现状造成了组织在信息管理过程中形成了信息孤岛，如何解决这些信息孤岛，形成完整的信息技术体系。这是迫切需要考虑的问题。

- **分析的方法较少**

越来越多的应用涉及到大数据，这些大数据的属性，包括数量，速度，多样性等等都是呈现了大数据不断增长的复杂性，所以，大数据的分析方法在大数据领域就显得尤为重要，可以说是决定最终信息是否有价值的决定性因素。

预测性分析能力：数据挖掘可以让分析员更好的理解数据，而预测性分析可以让分析员根据可视化分析和数据挖掘的结果做出一些预测性的判断。

数据质量和数据管理：数据质量和数据管理是一些管理方面的最佳实践。通过标准化的流程和工具对数据进行处理可以保证一个预先定义好的高质量的分析结果。

可视化分析：不管是对数据分析专家还是普通用户，数据可视化是数据分析工具最基本的要求。可视化可以直观的展示数据，让数据自己说话，让观众听到结果。

语义引擎：我们知道由于非结构化数据的多样性带来了数据分析的新的挑战，我们需要一系列的工具去解析，提取，分析数据。语义引擎需要被设计成能够从“文档”中智能提取信息。

数据挖掘算法：可视化是给人看的，数据挖掘就是给机器看的。集群、分割、孤立点分析还有其他的算法让我们深入数据内部，挖掘价值。这些算法不仅要处理大数据的量，也要处理大数据的速度。

- **对于趋势性的东西做态势感知困难**

态势感知就是针对用户的一类或多类数据，利用数据挖掘手段（统计、分类、聚类等）展示数据统计分类结果，感知数据关联关系，并且根据需求对数据发展趋势进行预测的一个系统。

态势感知的目的是发挥数据价值，提高决策水平，直观描述用户所关心事件的发展过程和未来趋势。

目前态势感知面临的问题：有数据但没有分析、不知道该如何挖掘、懂业务的人看不懂数据。

2 金钻芯大数据基础平台总体设计

2.1 总体架构设计

金钻芯大数据基础平台包含三个组成部分：**采集层、存储层、计算管理层**。

金钻芯大数据基础平台具有完整的上下游产品支撑，为用户提供了数据的采集、存储、批处理、数据流分析、全文搜索以及数据共享服务，在解决方案中：

- 采集层

使用金钻芯数据集成系统支持数据的实时增量采集和清洗，主要功能为：数据转换、数据合并和分拆、数据过滤、数据去重、数据校验。

- 存储层

使用金钻芯分布式数据库系统支持结构化数据和非结构化数据的存储（基于HDFS），此外还有独特的文件系统 UFS 支持海量小文件的存储。

- 计算管理层

使用金钻芯分布式数据库系统 UDB 作为搜索引擎，金钻芯算法库 UMT 作为机器学习，Spark 作为交互式实时分析计算框架，Spark Steaming 作为流式计算引擎，UMS 作为管理系统。

2.2 核心功能设计

2.2.1 数据集成

金钻芯数据集成系统（以下简称：金钻芯数据集成系统）是一套完整的数据加工处理工具，具备数据清洗、数据转换、数据影射、数据分配、数据跟踪、数据质量检查以及数据汇总等功能。

金钻芯数据集成系统满足政府部门建设数据仓库及数据集市、数据集中、对内对外信息处理加工等应用中的数据加工处理需求。

金钻芯数据集成系统可以访问所有类型、结构或来源的所有数据——从各种数据库系统到 XML 文档和电子表格；可以支持不断变化的 IT 环境，拥有开放式的、独立于硬件平台的体系结构；可以简化数据集成过程并加速开发、部署以及维护的一个统一的体系结构；基于元数据和开放标准的共享服务方法，可提供透明性、互操作性和灵活性。

2.2.2 数据存储

- HDFS——分布式计算的存储基石

HDFS 是分布式计算的存储基石，Hadoop 分布式文件系统对于整个集群有单一的命名空间；具有数据一致性，都适合一次写入多次读取的模型，客户端在文件没有被成功创建之前是无法看到文件存在的；文件会被分割成多个文件块，每个文件块被分配存储到数据节点上，而且会根据配置由复制文件块来保证数据的安全性。

HDFS 通过三个重要的角色来进行文件系统的管理：NameNode、DataNode 和 Client。NameNode 可以看做是分布式文件系统中的管理者，主要负责管理文件系统的命名空间、集群配置信息和存储块的复制等。NameNode 会将文件系统的 Metadata 存储在内存中，这些信息主要包括文件信息、每一个文件对应的文件块的信息和每一个文件块在 DataNode 中的信息等。DataNode 是文件存储的基本单元，它将文件块（Block）存储在本地文件系统中，保存了所有 Block 的 Metadata，同时周期性地将所有存在的 Block 信息发送给 NameNode。Client 就是需要获取分布式文件系统文件的应用程序。

- **UDB——分布式 NoSQL 数据库**

UDB 分布式数据库是专门针对大数据分析应用场景而设计，兼具事务处理功能，底层采用分布式架构，计算引擎设计遵循 SQL99 标准，提供 PLSQL 接口，成功解决了普遍存在的采用分布式架构同时兼容基于 SQL 和 ORACLE 应用的难题，全面支持结构化和非结构化数据的处理，同时 UDB 数据库拥有独特的压缩存储专算法，极大地提高了计算和存储速度，支持高并发的分析统计和查询，具有高安全、高速度、易使用、易维护、低成本等特点，在普通商用服务器集群的环境下，成功实现海量数据（1PB，1000 亿条记录）处理的秒级响应，速度和性能远超在同样配置下的国际先进数据库软件，或者达到其在豪华配置下才能实现的速度和性能。大大提高了数据处理效率，节省了数据处理成本。

对结构化数据，UDB 分布式数据库全面支持 SQL99，借鉴了目前国内外所有的领先技术，集百家之所长，功能上与国际先进数据库（Oracle、Microsoft SQLServer 等）一样强大，在无缝切换的同时使用分布式技术，提高性能、降低成本。

对于非结构化数据，UDB 分布式数据库很好地实现磁盘资源的合理利用，全面支持对于海量文件的管理诸如检索、过期等，并且还可以与结构化数据联动，为业务系统的开发提供了更大的弹性。

在整体使用方面，UDB 分布式数据库也非常重视并提供了诊断、调优、运维、备份恢复等方面不可或缺的基础功能。

- **UFS——分布式云存储系统**

金钻芯云存储系统 UFS（Universal File System）基于 scale-out 存储架构设计，具有强大的横向扩展能力，能够支持存储容量无限扩展和满足处理数千客户端并发读写需求，可广泛用于海量图片文件、视频片段等任意大小文件的存储。

UFS 主要解决了海量的文件（主要是图片、视频、音频等）存储和高并发访问的问题，并在文件存取时实现了负载均衡。同时它使得应用程序可以在任意地点通过 WEB 访问到这些文件，当然也支持各种移动设备对 UFS 系统的访问。与其它存储系统相比，UFS 最大的特点在于它的高性能。

2.2.3 离线计算

MapReduce 是一个高性能的批处理分布式计算框架，用于对海量数据进行并行分析和处理。与传统数据仓库和分析技术相比，MapReduce 适合处理各种类型的数据，包括结构化、半结构化和非结构化数据。数据量在 TB 和 PB 级别，在这个量级上，传统方法通常已经无法处理数据。MapReduce 将分析任务分为大量的并行 Map 任务和 Reduce 汇总任务两类。Map 任务运行在多个服务器上。

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。把一堆杂乱无章的数据按照某种特征归纳起来，然后处理并得到最后的结果。Map 面对的是杂乱无章的互不相关的数据，它解析每个数据，从中提取出 key 和 value，也就是提取了数据的特征。经过 MapReduce 的 Shuffle 阶段之后，在 Reduce 阶段看到的都是已经归纳好的数据了，在此基础上我们可以做进一步的处理以便得到结果。

2.2.4 实时分析

Spark 是一种与 Hadoop 相似的开源集群计算环境，启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。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 MapReduce 的算法。Spark 是为了支持分布式数据集上的迭代作业，但是实际上它是对 Hadoop 的补充，可以在 Hadoop 文件系统中并行运行。

2.2.5 流式计算

Spark Streaming 是构建在 Spark 上处理 Stream 数据的框架，基本的原理是将 Stream 数据分成小的时间片断（几秒），以类似 batch 批量处理的方式来处理这小部分数据。Spark Streaming 构建在 Spark 上，一方面是因为 Spark 的低延迟执行引擎（100ms+），虽然比不上专门的流式数据处理软件，也可以用于实时计算，另一方面相比基于 Record 的其它处理框架（如 Storm），一部分窄依赖的 RDD 数据集可以从源数据重新计算达到容错处理目的。

2.2.6 搜索引擎

UDB 分布式数据库的查询引擎根据关系型数据结构的特点为 SQL 实现了类似 Google Map/Reduce 的并行处理技术：表数据已经被系统预先分割成多个小表，可以作为查询引擎的多个输入，主服务器解析 SQL 查询，生成语法解析树，然后根据解析树以及小表数据在从服务器中的分配情况生成可执行的优化树，UDB 分布式数据库引擎在执行优化树时，把优化树构造成一系列的作业（Job），而每

个作业又由很多同类型的任务（Task）组成。作业被 UDB 分布式数据库的查询引擎按生成的顺序执行，而在执行每一个作业时，它的组成任务被均匀分配到从服务器上并行执行，从而极大地提高了数据库系统的查询性能。主服务器收集最终的查询结果，返回给用户。UDB 数据库的作业由以下类型组成：Restrict，NormalJoin，OuterJoin，ExistsJoin，GroupBy，MergeGroupBy，Sort，MergeSort，Project 等组成，而一个相对复杂的查询（比如 TPCH 规范中的查询）往往由几十个作业组成，作业数量越多，并行程度越高，如果小表服务器（TabletServer）越多，查询速度也越快。

UDB 分布式数据库对非结构化文本文件的全文索引（如常用的 PDF，Word，Excel，PPT，Txt 以及 Html 等）实现全文索引功能，实现了基于 HDFS 的索引存储，保证了索引数据的安全性，并对索引数据进行自动分段，由多服务器均衡管理。全文检索时，多服务器对索引段并行检索，这样就提高了查询效率。处理 Bfile 类型的文件时，利用现有的解析类库，从不同格式的文档中（例如 HTML, PDF, Doc, Txt），侦测和提取出元数据和结构化内容。

全文检索的查询方法与其他支持全文检索的数据库类似，使用 CONTAINS 谓词进行全文检索，UDB 的全文检索支持多个查询词之间的 AND、OR、NOT 等逻辑操作。

2.2.7 管理系统

金钻芯大数据平台管理系统 UMS 包含两个子系统：

一、平台资源管理系统

金钻芯率先使用 YARN 在融合多个主流大数据计算框架之上，进行统一的资源管理框架。金钻芯改进了 Apache YARN 资源管理框架，使得可在由分布式数据库 UDB 提供的同一个数据视图上，进行动态创建 SQL 交互式分析集群、Map/Reduce 批处理集群、实时计算 Spark 集群以及流式计算 Spark Streaming 集群，提供多任务的计算资源配额管理、动态资源调配、资源共享的能力，企业客户不再需要东拼西凑的混合架构，不需要孤立的多个集群，无需繁琐的数据迁移与重复存储，为企业建立一体化数据平台提供有力支持。

实现主要功能有：

1、具备完整的资源分配与调度机制，能够根据数据处理任务的级别，本身的属性等，根据当前平台的资源进行动态的调度分配，可对作业按照负荷、磁盘运行状况、网络运行状况、资源分配状态等，对数据处理作业进行有序、高效的调度，确保数据处理服务效率。

2、在一个动态共享的物理资源池中运行多种业务框架逻辑集群。

3、在同一个集群上运行多个实例，以隔离生产和实验作业，甚至是多个版本的业务作业。

4、无需重新设计底层基础设施就可以构建新的集群计算框架，并使其与现存框架共存。

5、采用相同的物理配置，管理方便、扩展便利；统一管理、统一监控、统一部署、统一运维。

6、业务资源运行时按需分配，周期性释放资源；最大限度的发挥平台中全部计算资源和 I/O 资源的价值。

二、集群可视化部署与监管系统

该系统是金钻芯大数据基础平台自动化部署工具，提供全中文与可视化图形界面，解决了开源系统部署步骤繁琐、命令行式命令容易出错等问题。系统提供了丰富的日常运维管理工具，能够管理集群各节点状态、进行服务配置与生命周期管理、数据库管理等各种方面。系统提供性能监控及分析工具，通过性能监控能够获取系统当前负载、瓶颈，为管理员进一步优化提供依据。同时，系统提供异常报警机制，当集群出现异常时，可及时以邮件等多种方式通知管理员，以便尽快解决问题。

2.3 关键技术

- ●数据采集加工处理技术
- ●分布式海量数据存储技术
- ●分布式实时流式计算分析技术
- ●基于架构的批量数据处理技术
- ● 大规模并行处理(MPP)数据库技术
- ● 弹性搜索引擎技术
- ● 交互式查询技术
- ● 多种机器学习和数据挖掘技术
- ● 云计算技术

3 金钻芯大数据基础平台功能介绍

3.1 数据集成——监控器

金钻芯数据集成系统有独立的监控器，具有以下功能：

系统监控：监控服务器资源使用情况以及单个服务资源使用情况：CPU、内存、硬盘。

服务监控：统一监控所有 UTL 服务实时状态、实时流量等信息。

服务管理：服务的添加、修改、删除、控制、调度。

服务审计：用于按条件查询所有服务的总调度审计日志、任意一项服务的调度次数、每次调度的总出库/总入库记录数、以及每次调度的控件出库/入库记录数。

报警查询：用于按条件查询所有服务的报警日志信息。

用户管理：用于管理访问管理中心的所有用户。

操作审计：用于按条件查询所有用户在金钻芯数据集成系统中的操作信息。当用户执行用户登录、修改服务、部署服务、用户退出、删除服务、启动服务、修改密码、创建服务、停止服务等操作时，将记录其操作审计信息。

白名单功能：用于设置可以访问金钻芯数据集成系统的客户端 IP 地址范围。管理员登录系统时，将首先验证客户端的 IP 地址，如果超出限制的 IP 范围，则拒绝其登录。

监控器以 B/S 的方式，使用浏览器运行，客户端无需安装任何其它程序。

监控器提供用户名/口令方式的登录和证书登录两种方式，确保赋予权限的用户才能使用。

3.2 数据集成——流程设计器

在线流程设计器主要为用户提供向导式的数据集成规则设置，从类别、处理阶段、具体整合功能（动作）等，将设计过程划分为阶段、步骤、动作。最小的调度单元是任务，任务之间可以有关联（某个任务必须在另一个任务结束之后才可以执行），手工调度则可以执行任何一个层次，最小到动作。

在线流程设计器目前提供的功能有:作业流程设计、转换流程设计、转换文件导入、作业与多个转换相关联。

提供大量的作业和转换组件支持，可以组合成一个完整的数据转换作业流程。通过条件测试控件配置和错误处理流程设置，即使转换过程出现各种异常情况，都能确保转换按照设定的流程运行。

3.3 全文索引

对结构化文本文件实现全文索引功能，实现了基于 HDFS 的索引存储，保证了索引数据的安全性，并对索引数据进行自动分段，由多服务器均衡管理。全文检索时，多服务器对索引段并行检索，这样就提高了查询效率。处理 Bfile 类型的文件时，利用现有的解析类库，从不同格式的文档中（例如 HTML, PDF, Doc,Txt)，侦测和提取出元数据和结构化内容。

全文检索的查询方法与其他支持全文检索的数据库类似，使用 CONTAINS 谓词进行全文检索，UDB 的全文检索支持多个查询词之间的 AND、OR、NOT 等逻辑操作。

创建分词索引：中文分词建立索引速度在 100MB/秒 。

关键词检索速度：总量在 15TB 的文本数据，关键词检索响应时间小于 3 秒。

3.4 大数据离线计算

数据划分和计算任务调度。将一个作业（Job）待处理的大数据划分为很多个数据块，每个数据块对应于一个计算任务（Task），并自动 调度计算节点来处理相应的数据块。作业和任务调度功能主要负责分配和调度计算节点（Map 节点或 Reduce 节点），同时负责监控这些节点的执行状态，并 负责 Map 节点执行的同步控制。

减少数据通信，一个基本原则是本地化数据处理，即一个计算节点尽可能处理其本地磁盘上所分布存储的数据，这实现了代码向 数据的迁移；当无法进行这种本地化数据处理时，再寻找其他可用节点并将数据从网络上传送给该节点（数据向代码迁移），但将尽可能从数据所在的本地机架上寻 找可用节点以减少通信延迟。

减少数据通信开销，中间结果数据进入 Reduce 节点前会进行一定的合并处理；一个 Reduce 节点所处理的数据可能会来自多个 Map 节点，为了避免 Reduce 计算阶段发生数据相关性，Map 节点输出的中间结果需使用一定的策略进行适当的划分处理，保证相关性数据发送到同一个 Reduce 节点；此外，系统还进行一些计算性能优化处理，如对最慢的计算任务采用多备份执行、选最快完成者作为结果。

3.5 大数据实时分析

交互式查询、实时流的数据处理。Spark Streaming 已支持了丰富的输入接口，大致分为两类：一类是磁盘输入，如以 batch size 作为时间间隔监控 HDFS 文件系统的某个目录，将目录中内容的变化作为 Spark Streaming 的输入；另一类就是网络流的方式，目前支持 Kafka、Flume、Twitter 和 TCP socket。在 WordCount 例子中，假定通过网络 socket 作为输入流，监听某个特定的端口，最后得出输入 DStream（lines）。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。

3.6 兼容 SQL 99

UDB 分布式数据库依据传统的关系型数据库理论开发，遵循 SQL99 标准，但底层采用分布式架构，实现横向无限扩展、无单点故障，易于维护。与传统的关系型数据库 ORACLE 保持了高度兼容（PL/SQL 兼容），能够提供数据和程序两个层面的平滑迁移。