白皮书
英特尔 Hadoop
大数据解决方案

英特尔 Hadoop 发行版

英特尔 Hadoop 发行版提供下列核心优势：

1. 经过客户验证的企业级 Hadoop 版本，稳定可靠
2. 图形化安装、管理、监控工具，自动进行集群配置优化
3. 改进的 HDFS 文件 I/O 算法，提高系统扩展性，适合不同配置服务器组成的集群
4. 根据 HDFS 数据的热点程度动态调整数据复制策略，提高 HDFS 系统吞吐量
5. HDFS 和 MapReduce 的高可靠性增强
6. 跨区域数据中心的 HBase 超级大表，位置透明的数据访问和全局汇总
7. HBase大表跨数据中心远程双向复制，适合异地灾备
8. HBase 高级 Region 负载均衡算法，适合多应用、多用户
9. 基于 HBase 的更高性能的分布式聚合和统计
10. HBase 的不同表或不同列族的复制份数精细控制

分布式文件系统 HDFS 简介

Hadoop 分布式文件系统（HDFS）是运行在通用硬件上的分布式文件系统。HDFS 提供了一个高度容错性和高吞吐量的海量数据存储解决方案。HDFS 已经在各种大型在线服务和大型存储系统中得到广泛应用，已经成为海量数据存储的事实标准。

随着信息系统的快速发展，海量的信息需要可靠存储的同时，还能被大量的使用者快速地访问。传统的存储方案已经从构架上越来越难以适应近几年来的信息系统业务的飞速发展，成为了业务发展的瓶颈和障碍。

HDFS 通过一个高效的分布式算法，将数据的访问和存储分布在大量服务器之中，在可靠地多备份存储的同时还能将访问分布在集群中的各个服务器之上，是传统存储构架的一个颠覆性的发展。HDFS 可以提供以下特性：

- 可自我修复的分布式文件存储系统
- 高可扩展性，无需停机动态扩容
- 高可靠性，数据自动检测和复制
- 高吞吐量访问，消除访问瓶颈
- 使用低成本存储和服务器构建

分布式文件系统 HDFS 特性

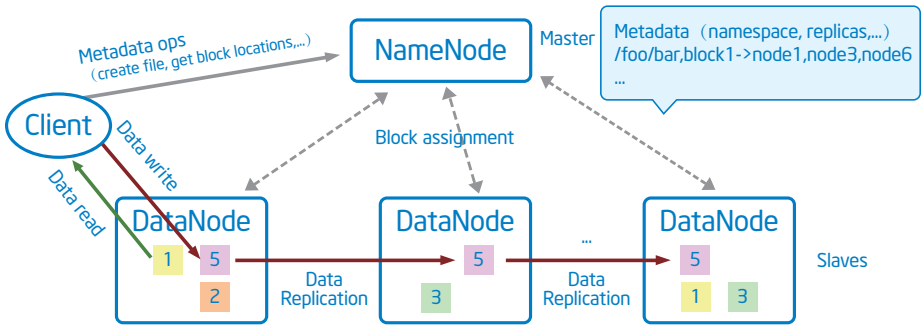
高吞吐量访问

HDFS 的每个数据块分布在不同机架的一组服务器之上，在用户访问时，HDFS 将会计算使用网络最近的和访问量最小的服务器给用户提供访问。由于数据块的每个复制拷贝都能提供给用户访问，而不是仅从数据源读取，HDFS 对于单数据块的访问性能将是传统存储方案的数倍。

对于一个较大的文件，HDFS 将文件的不同部分存放于不同服务器之上。在访问大型文件时，系统可以并行从服务器阵列中的多个服务器并行读入，增加了大文件读入的访问带宽。

通过以上实现，HDFS 通过分布式计算的算法，将数据访问均摊到服务器阵列中的每个服务器的多个数据拷贝之上，单个硬盘或服务器的吞吐量限制都可以数倍甚至数百倍的突破，提供了极高的数据吞吐量。

目录



无缝容量扩充

HDFS 将文件的数据块分配信息存放在 Name Node 服务器之上，文件数据块的信息分布地存放在 DataNode 服务器上。当整个系统容量需要扩充时，只需要增加 DataNode 的数量，系统会自动地实时将新的服务器匹配进整体阵列之中。之后，文件的分布算法会将数据块搬迁到新的 NameNode 之中，不需任何系统当机维护或人工干预。

通过以上实现，HDFS 可以做到在不停止服务的情况下实时地加入新的服务器作为分布式文件系统的容量升级，不需要人工干预文件的重分布。

高度容错

HDFS 文件系统假设系统故障（服务器、网络、存储故障等）是常态，而不是异常。因此通过多方面保证数据的可靠性。数据在写入时被复制多份，并且可以通过用户自定义的复制策略分布到物理位置不同的服务器上；数据在读写时将自动进行数据的校验，一旦发现数据校验错误将重新进行复制；HDFS 系统在后台自动连续的检测数据的一致性，并维持数据的副本数量在指定的复制水平上。

分布式数据库 HBase 简介

HBase 是一个面向列的实时分布式数据库。HBase 不是一个关系型数据库，其设计目标是用来解决关系型数据库在处理海量数据时的理论和实现上的局限性。传统关系型数据库在上世纪七十年代为交易系统设计，以满足数据一致性（ACID）为目标，并没有考虑数据规模扩大时的扩展性，以及系统故障时的可用性。虽然经过多年的技术发展，产生了一些对关系性数据库的修补（并行数据库），然而受限于理论和实现上的约束，扩展性从来没有超过 40 个服务器节点。而 HBase 从一开始就是为 Terabyte 到 Petabyte 级别的海量数据存储和高速读写而设计，这些数据要求能够被分布在数千台普通服务器上，并且能够被大量并发用户高速访问。

分布式数据库 HBase 的特点和优势

高可扩展性

HBase 是真正意义上的线性水平扩展。数据量累计到一定程度（可配置），HBase 系统会自动对数据进行水平切分，并分配不同的服务器来管理这些数据。这些数据可以被扩散到上千个普通服务器上。这样一方面可以由大量普通服务器组成大规模集群，来存放海量数据（从几个 TB 到几十 PB 的数据）。另一方面，当数据峰值接近系统设计容量时，可以简单通过增加服务器的方式来扩大容量。这个动态扩容过程无需停机，HBase 系统可以照常运行并提供读写服务，完全实现动态无缝无宕机扩容。

高性能

HBase 的设计目的之一是支持高并发用户数的高速读写访问。这是通过两方面来实现的。首先数据行被水平切分并分布到多台服务器上，在大量用户访问时，访问请求也被分散到了不同的服务器上，虽然每个服务器的服务能力有限，但是数千台服务器汇总后可以提供极高性能的访问能力。其次，HBase 设计了高效的缓存机制，有效提高了访问的命中率，提高了访问性能。

高可用性

HBase 建立在 HDFS 之上。HDFS 提供了数据自动复制和容错的功能。HBase 的日志和数据都存放在 HDFS 上，即使在读写过程中当前服务器出现故障（硬盘、内存、网络等故障），日志也不会丢失，数据都可以从日志中自动恢复。HBase 系统会自动分配其他服务器接管并恢复这些数据。因此一旦成功写入数据，这些数据就保证被持久化并被冗余复制，整个系统的高可用性得到保证。

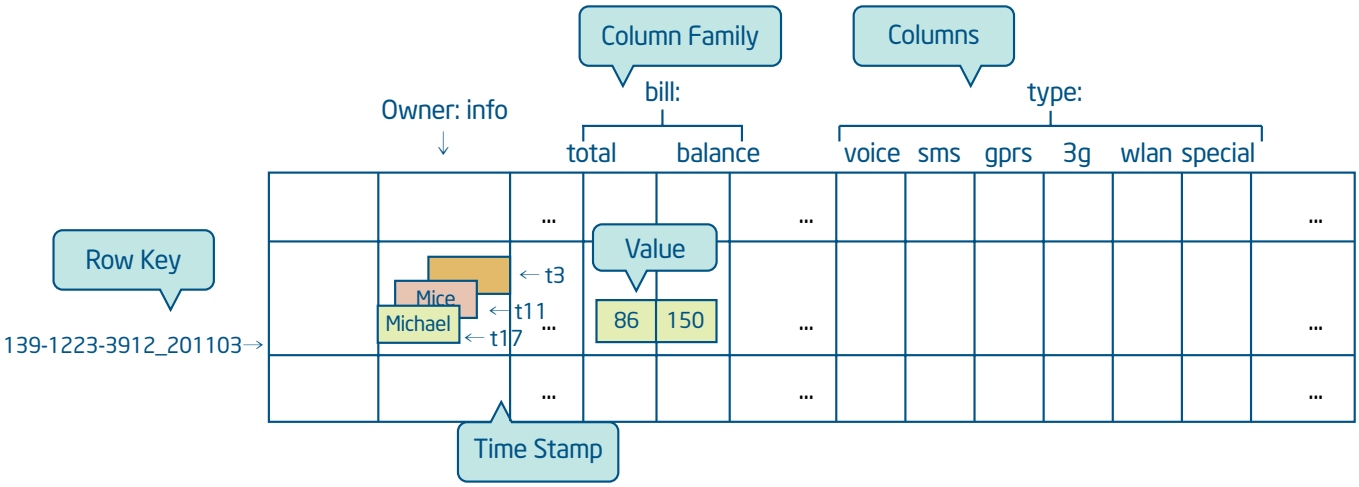
数据模型及其特点

HBase 是一个面向列的、稀疏的、分布式的、持久化存储的多维排序映射表（Map）。表的索引是行关键字、列族名（Column Family）、列关键字以及时间戳；表中的每个值都是一个未经解析的字节数组。

面向列：指的是同一个列族里所有数据都存放在一个文件中，从而在读写时有效降低磁盘 I/O 的开销，并且由于类似类型的数据存放在一起，提高了压缩比。经过压缩后的数据容量通常达到原来的 1/3 到 1/5，极大节省了存储空间。

多维表：这是对传统二维关系表的极大扩充。传统二维表有两维：行和列。列在设计表结构时必须预先固定，而行可以动态增加，也就是说有一个维度可动态改变。HBase 的多维表有四维，列族需要在设计表结构时事先确定，而行、列、时间维都可以动态增加。也就是说有三个维度可动态改变。这种结构非常适合用来表述有嵌套关系的数据。另外，动态增删列的能力也给很多业务带来便利，特别是这些业务在不停的演化，需要的列字段也在不停的增加，多维表结构可以随时进行改变以适应业务发展需求。

稀疏表：由于多维表的列可以动态增加，必然导致不同行相同列的数据大部分为空，也就是说这个表是稀疏的。不像传统关系型数据库，HBase 不存放空值，只存放有内容的表格单元（cell），因此可以支持超大稀疏表，而不会带来任何开销。这对传统的表结构设计也带来了观念上的大改变。



分布式计算框架 MapReduce 简介

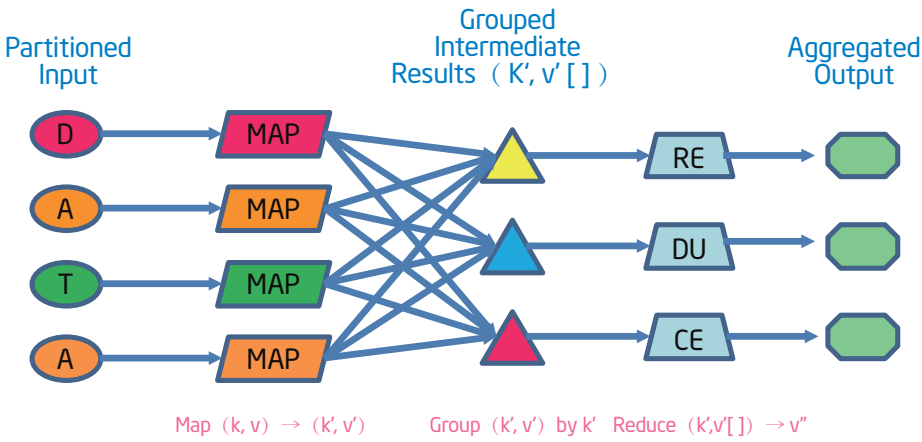
MapReduce 是一个高性能的批处理分布式计算框架，用于对海量数据进行并行分析和处理。与传统数据仓库和分析技术相比，MapReduce 适合处理各种类型的数据，包括结构化、半结构化和非结构化数据。数据量在 TB 和 PB 级别，在这个量级上，传统方法通常已经无法处理数据。MapReduce 将分析任务分为大量的并行 Map 任务和 Reduce 汇总任务两类。Map 任务运行在多个服务器上。目前部署的最大集群有 4000 个服务器。

MapReduce 适合处理的任务

复杂的数据：业务数据不能适合行列的数据库结构。数据可能来源于多种格式：多媒体数据、图像数据、文本数据、实时数据、传感器数据等等。当有新的数据来源时，可能会有新的数据格式的出现。MapReduce 可以存放和分析各种原始数据格式。

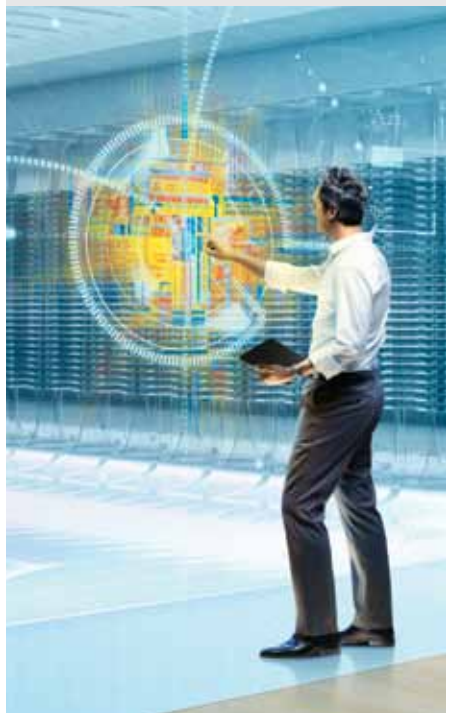
超大规模数据：很多公司仅仅应为数据存放成本过高就放弃了很多有价值的数据。新的数据来源使得问题更为严重，新的系统和用户带来比以往更多的数据。Hadoop 的创新构架使用低成本的常规服务器储存和处理海量的数据。

新的分析手段：海量复杂数据分析需要使用新的方法。新的算法包括自然语言分析、模式识别等。只有 Hadoop 的构架才能方便高效地使用新的算法来处理和分析海量数据。



MapReduce 框架的核心优势：

1. 高度可扩展，可动态增加/削减计算节点，真正实现弹性计算。
2. 高容错能力，支持任务自动迁移、重试和预测执行，不受计算节点故障影响。
3. 公平调度算法，支持优先级和任务抢占，兼顾长/短任务，有效支持交互式任务。
4. 就近调度算法，调度任务到最近的数据节点，有效降低网络带宽。
5. 动态灵活的资源分配和调度，达到资源利用最大化，计算节点不会出现闲置和过载的情况；同时支持资源配额管理。
6. 经过大量实际生产环境使用和验证，最大集群规模在 4000 个计算节点。



数据仓库 Hive 简介

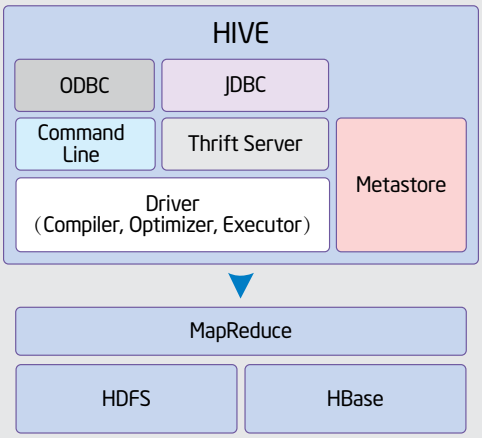
Hive 是一种建立在 Hadoop 之上的数据仓库架构。它提供了：

- 一套方便的实施数据抽取（ETL）的工具。
- 一种让用户对数据描述其结构的机制。
- 支持用户对存储在 Hadoop 中的海量数据进行查询和分析的能力。

Hive 的基本特点是它采用 HDFS 进行数据存储并利用 MapReduce 框架进行数据操作。所以从本质上来说，Hive 就是个编译器，它把用户的操作（查询或者 ETL）变换成 MapReduce 任务，利用 MapReduce 框架执行这些任务以对 HDFS 上的海量数据进行处理。

Hive 被设计成一种批处理系统。它利用 MapReduce 框架来处理数据。因此，它在 MapReduce 任务提交和调度上有比较高的开销。即使对于小数据集（几百兆）来说，延迟也是分钟级的。但其最大的优点是延迟相对于数据集大小是线性增加的。

Hive 定义了一种简单的类 SQL 查询语言 HiveQL，让熟悉 SQL 的用户可以非常容易的进行查询。与此同时，HiveQL 也允许熟悉 MapReduce 框架的程序员在查询中插入自定义的 mapper 和 reducer 脚本以扩展 Hive 内嵌的功能，完成更复杂的分析。



Hive 系统结构

Hive 包括

- 元数据存储（Metastore）
- 驱动（Driver）
- 查询编译器（QueryCompiler）
- 执行引擎（ExecutionEngine）
- 服务器（HiveServer）
- 客户端组件
- 可扩展接口部分

Hive 特点

针对海量数据的高性能查询和分析系统

由于 Hive 的查询是通过 MapReduce 框架实现的，而 MapReduce 本身就是为实现针对海量数据的高性能处理而设计的。所以 Hive 天然就能高效的处理海量数据。

与此同时，Hive 针对 HiveQL 到 MapReduce 的翻译进行了大量的优化，从而保证了生成的 MapReduce 任务是高效的。在实际应用中，Hive 可以高效的对 TB 甚至 PB 级的数据进行处理。

类 SQL 的查询语言

HiveQL 和 SQL 非常类似，所以一个熟悉 SQL 的用户基本不需要培训就可以非常容易的使用 Hive 进行很复杂的查询。

HiveQL 灵活的可扩展性（Extendibility）

除了 HiveQL 自身提供的能力，用户还可以自定义其使用的数据类型、也可以用任何语言自定义 mapper 和 reducer 脚本，还可以自定义函数（普通函数、聚集函数）等。这就赋予了 HiveQL 极大的可扩展性。用户可以利用这种可扩展性实现非常复杂的查询。

高扩展性（Scalability）和容错性

Hive 本身并没有执行机制，用户查询的执行是通过 MapReduce 框架实现的。由于 MapReduce 框架本身具有高度可扩展（计算能力随 Hadoop 机群中机器的数量增加而线性增加）和高容错的特点，所以 Hive 也相应具有这些特点。

与 Hadoop 其他产品完全兼容

Hive 自身并不存储用户数据，而是通过接口访问用户数据。这就使得 Hive 支持各种数据源和数据格式。例如，它支持处理 HDFS 上的多种文件格式（TextFile、SequenceFile 等），还支持处理 HBase 数据库。用户也完全可以实现自己的驱动来增加新的数据源和数据格式。一种理想的应用模型是将数据存储在 HBase 中实现实时访问，而用 Hive 对 HBase 中的数据进行批量分析。

数据处理 Pig 简介

Pig 是一个基于 Hadoop 并运用 MapReduce 和 HDFS 实现大规模数据分析的平台。它为海量数据的并行处理提供了操作以及编程实现的接口。Pig 的编程语言为 Pig Latin，该语言有如下特点：

- **易于编程：**既具有类似 SQL 的灵活可变性，又有过程式语言的数据流特点。
- **优化策略：**系统具备自动优化执行过程的能力，使得用户更加关注于语义。
- **可扩展性：**用户可以自行设计函数来实现特定功能。

日志收集工具 Flume 简介

Flume 是一个高效的收集，聚合和运输海量日志数据的系统。在其实现架构中，最重要的特点是简单灵活的数据流抽象处理。同时，在 Flume 中，通过 ZooKeeper 保证配置数据的一致性和可用性。

Flume 本身具有如下特点：

- **可靠性：**Flume 提供端到端的可靠传输，数据本地化保存等可靠性选项。
- **可管理性：**通过 ZooKeeper 保证配置数据的可用性，并使用多个 Master 管理所有节点。
- **可扩展性：**可以用 Java 语言实现新的自定义功能。

MapReduce 应用场景

视频分析和检索

使用 Hadoop MapReduce 算法，将存放在视频图片库中的海量数据并行分析检索，并可以将分析结果实时汇总，以提供进一步的分析及使用。MapReduce 算法使得原来需要几天的分析计算缩短到几个小时，如果需要甚至可以通过添加服务器的方式线性增加系统的处理能力。新的算法，比如数字城市中的车牌识别、套牌分析、车辆轨迹分析等应用，都通过 MapReduce 算法部署到服务器集群中。

客户流失性分析

风险分析需要在不同数据源的海量数据中使用模式识别技术寻找出具有风险倾向的个体或公司。海量数据的存储、搜索、读取和分析都是需要高计算能力和高吞吐量的系统来实现。使用 MapReduce 算法可以将复杂的计算动态地分布到服务器集群中的各台服务器上并行处理，可以通过服务器的线性扩充轻易突破计算能力的瓶颈，解决海量数据高性能计算的问题。某运行商将所有的通讯记录实时导入到 HBase 中，一方面通过 HBase 提供实时的通讯记录查询功能，另一方面通过 MapReduce 分析用户的历史通讯记录以识别出优质客户；当他们的通讯量显著减少时，意味着这些用户可能已转移到其他运行商，从而可以采取特定优惠措施留住这些用户。

推荐引擎

推荐引擎工具用于找出物品之间的相关性，然后推荐给用户相似的物品，从而达到进一步吸引用户，提高用户粘性的目的。某购物网站采用 MapReduce 分析大量用户的购买记录，计算购买记录间的相似性，从而找出商品间的相关度。然后以商品为索引列出相关的其他商品。在用户购买了某一个商品后，网站根据分析结果推荐给用户可能感兴趣的其他商品。由于用户的购买记录是海量数据，要在特定时间内及时得到分析结果，必需采取 MapReduce 的方法对购买记录进行并行统计和汇总。

数据分析手段

- 全文挖掘
- 建立索引
- 图形创建和分析
- 模式识别
- 协同过滤
- 情感分析
- 风险评估

数据分析应用

- 视频分析和检索
- 现实风险建模
- 客户流失性分析
- 推荐引擎
- 广告目标投放
- 售卖点交易分析
- 网络失效预测
- 安全风险分析
- 商业交易监控
- 搜索质量评估
- 数据 “沙盒”

机器学习 Mahout 简介

Mahout 是一套具有可扩充能力的机器学习类库。它提供机器学习框架的同时，还实现了一些可扩展的机器学习领域经典算法的实现，可以帮助开发人员更加方便快捷地创建智能应用程序。通过和 Apache Hadoop 分布式框架相结合，Mahout 可以有效地使用分布式系统来实现高性能计算。

Mahout 现在提供 4 种使用场景的算法。

- **推荐引擎算法：**通过分析用户的使用行为的历史记录来推算用户最可能喜欢的商品、服务、套餐的相关物品。实现时可以基于用户的推荐（通过查找相似的用户来推荐项目）或基于项目的推荐（计算项目之间的相似度并做出推荐）。
- **聚类算法：**通过分析将一系列相关的物品等划分为相关性相近的群组。
- **分类算法：**通过分析一组已经分类的物品，将其他未分类的其他物品按同样规则归入相应的分类。
- **相关物品分析算法：**识别出一系列经常一起出现的物品组（经常一起查询、放入购物车等）。

Mahout 算法所处理的场景，经常是伴随着海量的用户使用数据的情况。通过将 Mahout 算法构建于 MapReduce 框架之上，将算法的输入、输出和中间结果构建于 HDFS 分布式文件系统之上，使得 Mahout 具有高吞吐、高并发、高可靠性的特点。最终，使业务系统可以高效快速地得到分析结果。

分布式协作服务 ZooKeeper 简介

ZooKeeper 是 Hadoop 和 HBase 的重要组件，它提供分布式应用程序的协调服务。分布式系统可以通过 ZooKeeper 实现系统配置维护，命名服务和同步服务等。

ZooKeeper 致力于将不同协调服务集成在一个简单易用的界面之上。它具有分布性和高度可靠性的特点。

关系数据 ETL 工具 Sqoop 简介

Sqoop 是一个可扩展的机器学习类库，与 Hadoop 结合后可以提供分布式数据挖掘功能。它在 Apache Hadoop 和关系型数据库之间高效传输大量数据的工具。通过使用 Sqoop，可以将外部数据库中数据导入 HDFS，Hive 等相关系统或者从这些系统中导出数据到商用关系型数据库和数据仓库。

为什么使用英特尔 Hadoop 发行版

英特尔 Hadoop 发行版具有下列优势：

- Intel 的 Hadoop 发行版是经过测试和验证的稳定版本，在客户生产环境成功部署运营，可以确保客户生产环境 7x24 小时不间断运行。
- Intel 的 Hadoop 发行版包括了 Intel 针对现有客户在实际使用中出现问题的解决方法以及大量改进和优化。这些改进弥补了开源 Hadoop 在实际使用中的缺陷和不足，并且包含了大量的性能优化。
- Intel 的集群管理工具和安装工具简化了 Hadoop 的安装和配置。可以根据用户的硬件环境自动生成最优化的集群配置，充分发挥集群的计算能力。
- 基于 Intel 在云计算研发上的经验积累，提供从项目规划到实施各阶段专业的咨询服务，帮助客户构建高可扩展高性能的分布式系统。
- 结合 Intel 的硬件部门，提供全面的软硬件解决方案设计。

英特尔 Hadoop 发行版产品与服务

Intel 提供给客户稳定可靠易用的 HDFS、HBase 和 MapReduce 框架商业版本，包括：

分布式文件系统（HDFS）商业套件

- 可自我修复的高带宽集群文件存储系统
- 高可扩展性，无需停机无缝动态扩容
- 高容错性，数据自动复制和校验
- 改进的可靠性和扩展性

分布式数据库（HBase）商业套件

- 分布式、面向列、多维度的数据库系统
- 数据自动切分和分布存储
- 高可扩展性，无宕机线性扩容
- 高性能并发读写

分布式计算框架（MapReduce）商业套件

- 高度并行和可扩展的分布式批处理计算框架
- 高容错能力，支持任务自动迁移和重试
- 公平调度算法，支持任务抢占，兼顾长短任务
- 调度任务到最近的数据节点，有效降低网络带宽
- 灵活的资源分配和调度，达到资源利用最大化

分布式数据仓库（Hive）商业套件

- 高性能分布式海量数据仓库
- 强大的查询与分析功能
- 类 SQL 查询语言

分布式数据处理（Pig）商业套件

- 大规模数据分析实现平台
- 系统自动化最优策略实现
- 高可扩展性，易于实现用户需求

分布式数据分析（Mahout）商业套件

- 可扩充能力的机器学习类库
- 实现了一些可扩展的机器学习领域经典算法
- 有效地使用 MapReduce 实现高性能计算

分布式协作服务（ZooKeeper）商业套件

- 提供分布式应用程序的协调服务
- 完成系统配置维护，命名服务和同步服务等
- 具有高可靠性和可扩展性

分布式日志收集（Flume）商业套件

- 收集，聚合和运输海量日志数据
- 高度一致性和可用性
- 简单灵活的数据流抽象处理方式

关系数据 ETL（Sqoop）商业套件

- 高效传输 Hadoop 和关系型数据库之间大量数据
- 与 Hadoop 结合提供分布式数据挖掘功能

专业咨询服务

- **项目立项阶段：**架构设计、项目规划、容量设计
- **问题分析阶段：**分析系统需求、定义数据处理方案、提出集群系统方案、优化方案设计
- **功能实现阶段：**提供问题解答和咨询

售后技术支持

- 远程 8x5 电话（800-820-1100）和 Web 网络支持
- 现场支持
- 产品升级服务
- 安装调试服务

英特尔 Hadoop 为企业应用而优化

英特尔 Hadoop 发行版针对开源 Hadoop 的源代码做了大量的改进和优化。跟开源 Hadoop 和其他 Hadoop 发行版相比，在核心部分具有下列重要改进：

1. 高可靠性增强，包括分布式文件系统 HDFS 的目录服务器（NameNode）的高可用性和分布式计算框架（MapReduce）的任务调度器（JobTracker）的高可用性以及为 HDFS 的 NameNode 提供双机热备方案。
2. 分布式文件系统 HDFS 扩展性增强，改进了 HDFS 数据的分布和读取算法，移除了读海量文件时的扩展性瓶颈，使得集群的整体性能不再受限于某些较慢的服务器节点。改进后集群系统的 I/O 吞吐量能够随节点数量增加而线性扩展。
3. 根据 HDFS 数据的热点程度，在硬盘容量允许的情况下，动态调整数据复制策略，可提高热点数据的并发访问能力，从而提高 HDFS 系统吞吐量。
4. 分布式计算框架（MapReduce）的调度算法的改进，支持公平调度原则，兼顾短任务和长任务的调度，并且能够很好的处理短（快速）的 Map 任务的并行调度，避免开源 Hadoop 版本出现的并行任务退化成串行执行的问题。
5. 增强了 Map-Only 这类应用的结果处理能力，允许这类应用对 Map 的结果进行自定义的合并和排序处理，这种处理方法的接口和 MapReduce 任务的处理方法一致。
6. 增加了 Hadoop 集群的监控管理，增加了更多的针对 Hadoop 服务的集群监控接口。
7. 改进了 Hadoop 对 NFS/NAS 的支持，使得 MapReduce 可以直接高效访问 NFS 进行 MapReduce 并行计算。
8. 针对 HBase 的查询进行了大量优化，降低了磁盘 I/O。优化后的查询算法兼顾磁盘 I/O 和网络 I/O，最大化吞吐量。
9. 细粒度的 HBase 合并调度控制，更好地避免合并风暴。
10. 支持 HBase 大表跨数据中心远程复制，支持单向、双向、一对多复制，适合异地灾备。
11. 支持在多个异地数据中心上创建统一的 HBase 大表（cross-site big table），提高扩展性和可用性。
12. 根据 HBase 数据局部性、服务器 Region 数、表的 Region 数来实现负载均衡，使得多应用、多用户在 HBase 上运行得更流畅，适合多用户共享集群创建多张大表的应用。
13. 提供基于 HBase 的更高可靠性的分布式聚合函数，包括 sum, avg, count, mean 等统计函数；支持并行扫描大表，以及 Group-By Aggregation，性能优于 MapReduce 方式，比传统方式提高 2~10 倍以上效率。
14. 实现对 HBase 的不同表或不同列族的复制份数精细控制，使结构化数据和非结构化数据共存时存储效率更高。

intel® Software