Microsoft HDIndight 大数据平台

2019 年 2 月

BESPIN GLOBAL

目录

1．微软大数据平台-HDInsight

大数据，对于如今的互联网世界来讲已经不再是陌生的词汇。各行各业的企业中都已经渗透了大数据的理念。互联网企业更是身先士卒，各大互联网巨头都已在大数据领域投资试水。微软也不例外。

什么是大数据?

大数据，故名思议，首先它是一套数据集的集合。然后这个集合非常大，非常复杂，以至于使用一般的数据库管理工具或者传统的数据处理程序会很难对它进行处理。

那哪些数据是属于大数据的范畴?根据大数据的定义，我们可以举出一些大数据的例子：

比如，传统的大数据有物理实验数据，各种感应器的数据，卫星数据等等。

随着人类社会的发展，计算机技术的发展，现在的大数据还包括一些计算机本身操作的日志，网店客户的一些行为表现，在线社交程序，微博，微信，twitter,facebook 之类的在线交互的内容， 等等。

我们可以发现，从上世纪九十年代的数据规模到现在，这个数据量是呈爆炸式的发展。从当初的 Terabytes，也就是兆兆字节，发展到现在的 PetaBytes,相信到不久的将来会有 Zetabytes 的数据量出现。

Apache Hadoop

那了解了大数据的特征，我们可以把处理大数据的平台作为一种机会来看待。事实 上从商业应用的价值，大数据是一个金矿，就看你如何去挖掘。而要想从

大数据中进行数据挖掘获取真正有价值的东西，那么首先，你需要这样一个大数据处理的平 台。这样的平台能够帮助你存储大数据，同时提供机制使得你能够去挖掘大数据的价值。这样一个平台正是你挖掘大数据的商业价值的基础。

而一个好的大数据处理平台，我们可以想到这样一些特性：

我们希望用多台机器，而且是多台普通的机器架设这样的平台。我们可以不需要非常高端的硬件。

而它能够有很好的扩充性。你能够随时在群集中增减机器的数量，我们称为 ScaleOut

它使用的是完全开源的软件，降低成本。

那 Apache Hadoop 就是这样一个系统。

那么 Hadoop 究竟包含些什么组件呢。Hadoop 并不仅仅是一个存储数据的架构，它同时提供了一个机制能够用来编写分布式可扩展的访问数据的应用。

Hadoop 包含下面两个重要的组成部分：

Hadoop Distributed File System

HDFS, Hadoop 分布式文件系统。它是一个分布式的，可扩展的，可移植的针对 Hadoop 架构下的文件系统。它是由 Java 编写的。

MapReduce

Map Reduce，是一种专门用于在群集中处理分布式数据的编程模型。

Hadoop 提供了这样一个实现。

Hadoop 可以把大数据分布到成百上千的普通计算机上。传统的软件程序通常是把数据拿到本地来进行处理，而 MapReduce 却是把执行代码推送到数据所在的节点上运行。那我们可以看到这样可以有效的避免网络带宽的限制，而且处 理的速度基本上同数据集大小本身保持线性关系，而性能取决于有多少个存储数据的节点。数据节点多，那么处理同样的数据量就快。

Relational Database vs. Hadoop

那 Hadoop 究竟跟传统的关系型数据库有些什么样的联系和不同呢?为了理解他们的关系和区别，我们有必要理解 schema 在 Hadoop 中是如何工作的：

Relational		Hadoop
Required on write	schema	Required on read
Reads are fast	speed	Writes are fast
Standards and structured	governance	Loosely structured
Limited, no data processing	processing	Processing coupled with data
Structured	data types	Multi and unstructured
Interactive OLAP Analytics Complex ACID Transactions Operational Data Store	best fit use	Data Discovery Processing unstructured data Massive Storage/Processing

首先，在关系型数据库中，schema 必须在写入数据之前就生成的。这可以使得数据必须符合某种模型的规则。

而对于 Hadoop，导入的数据保持了它原始的格式，是没有任何 schema 的。而当数据被提取的时候，这个时候 schema 才会根据你应用的需要被采用。

另外从读写来看，Hadoop 主要是基于批量操作，因此大批量的存储和大批量的处理是它的优势所在。而相比较而言，关系型数据库可以进行事务型读写操作，可能随机读写更具优势。由于他们针对的数据不同，处理方式也不同，所以我们不能够单纯的从某种操作的性能上来评价两者孰胜孰劣。

需要注意的是，Hadoop 并不是为了取代传统的关系型数据库。Hadoop 是为了存储大数据。这类数据通常因为数据量的的大小或者要求的环境限制等导致你不能存在数据库中。因此即使你打算使用 Hadoop,

你仍然需要你的关系型数据库。

1.1. 什么是 HDInsight?

Apache Hadoop 它是一个支持数据密集型的分布式应用的开源软件框架。那么 HDInsight 就是 Apache

Hadoop 这个框架的微软的分发实现。

HDInsight 是开源的，微软的修改可以回馈给主项目。

离开 Hadoop 本身的定义，我们可以认为 HDInsight 它是一种通过简单的编程模型，在计算机集群中对大数据集进行分布式处理的平台。它的重要特点是具有容错性，可以处理结构化的数据和非结构化的数据。

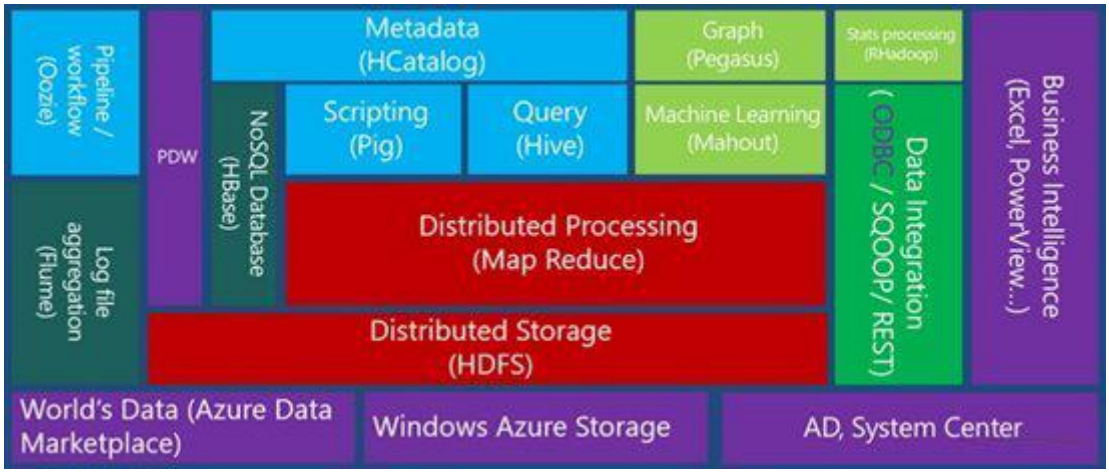
微软的 HDInsight 提供了两种部署模式，Azure HDInsight Service 和 HDInsightServer (on premise). 同时，微软也在 Parallel Data Warehouse 的 Appliance 中提供了 HDInsight 的实现

另外 HDInsight 的实现是通过微软同 HortonWorks 公司进行合作推出的。微软把 Hadoop 技术冠名为 HDInsight,

但它的基础其实是 HortonWorks 的 HortonWorks

Data Platform for Windows. 我们介绍的核心部分其实也适用于 HDP for Windows。

HDInsight 生态系统



从整个 HDInsight 生态系统来看，首先我们需要有一个分布式的存储机制，它就是 HDFS 框架。

在这个之上，我们需要有一个分布式的处理机制，那这个机制就是 MapReduce。

那基于 MapReduce 的机制，我们衍生了出了两种更上层的语言机制来帮助实现数据挖掘的功能，它们就是 Pig 和 Hive。

为了管理 Metadata，特别是使得其他工具能够在 Hive metastore 和 Pig,MapReduce 之间能互相共享数据和 Metadata,又衍生了一个 metadata，

table 的管理系统，叫做 HCatalog.HCatalog 是 Hive 的扩展，它可以把 Hive 中的表的 Schema 提供给其他工具。比如说，它提供了两个接口 HCatLoader 和 HCatStorer 给 Pig 来读和写 HCatalog 托管的表。

Hbase 是一个 Hadoop 的 Database，它是一个 NoSQL 的 Database。

在 MapReduce 挖掘数据部分又延伸出一些专门领域的工具，比如机器学习，图像挖掘，状态处理等等。

为了与不是 Hadoop 的关系型数据达到整合，我们又有了 ODBC driver, SQOOP，REST 接口等等。

有了这些与外部的接口，我们就可以用 BI 的工具进行大数据处理的分析和展现。

另一方面，为了能够有效的管理和调度 Hadoop 的 Job，又应运而生了一个工作流工具，叫做 Oozie。
还有关于日志的 Aggregation。

那有了这些，我们可以把 Azure 的数据，以及 PDW 的数据也可以联系进来，我们可以通过 AD, System Center，以及各式各类的世界数据都整合进来。这样就形成了一个完整的生态系统。

2. Azure HDInsight Service

HDInsight 是在 Microsoft Azure 上快速扩展 Apache Hadoop 技术堆栈（作为大数据分析的首选解决方案）的云实现。它包括 Storm、HBase、Pig、Hive、Sqoop、Oozie 等的实现。HDInsight 还可集成商业智能 (BI) 工具，例如 Excel、SQL Server Analysis Services 和 SQL Server Reporting Services。

按需灵活扩展

HDInsight 是一种云技术驱动的 Hadoop 发行版。这意味着 HDInsight 架构能够处理任何数量的数据，按需将数据处理容量从数 TB 扩展至数 PB 级别。您可以随时快速创建任意数量的节点。我们只对您实际使用的计算和存储收取费用。

结构化、半结构化、非结构化，所有数据一网打尽

由于完全符合 Apache Hadoop 标准，HDInsight 能够处理来自网络点击流、社交媒体、服务器日志、设备和传感器等来源的非结构化或半结构化数据。借此您能够分析新的数据集，从中寻找新商机，推动组织向前发展。

使用您惯用的语言进行开发

HDInsight 具有强大的编程扩展能力，适用于多种语言，包括 C#、Java、.NET 等。您可在 Hadoop 上使用自己习惯的编程语言进行 Hadoop 作业的创建、配置、提交和监控。

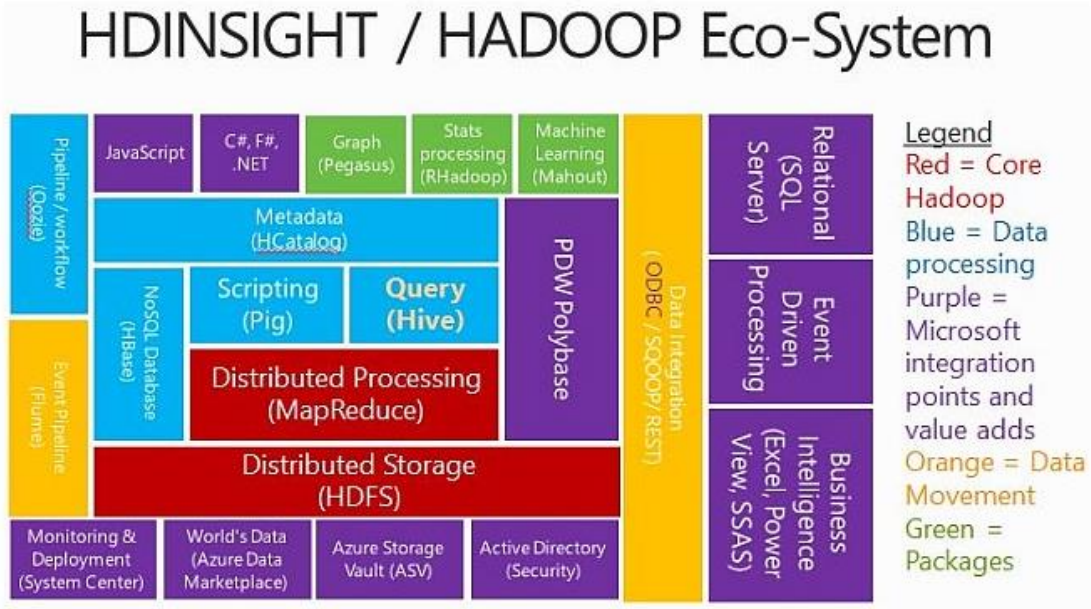
无需采购或维护硬件

使用 HDInsight，您可在云中部署 Hadoop，无需购买新硬件，也无需其他

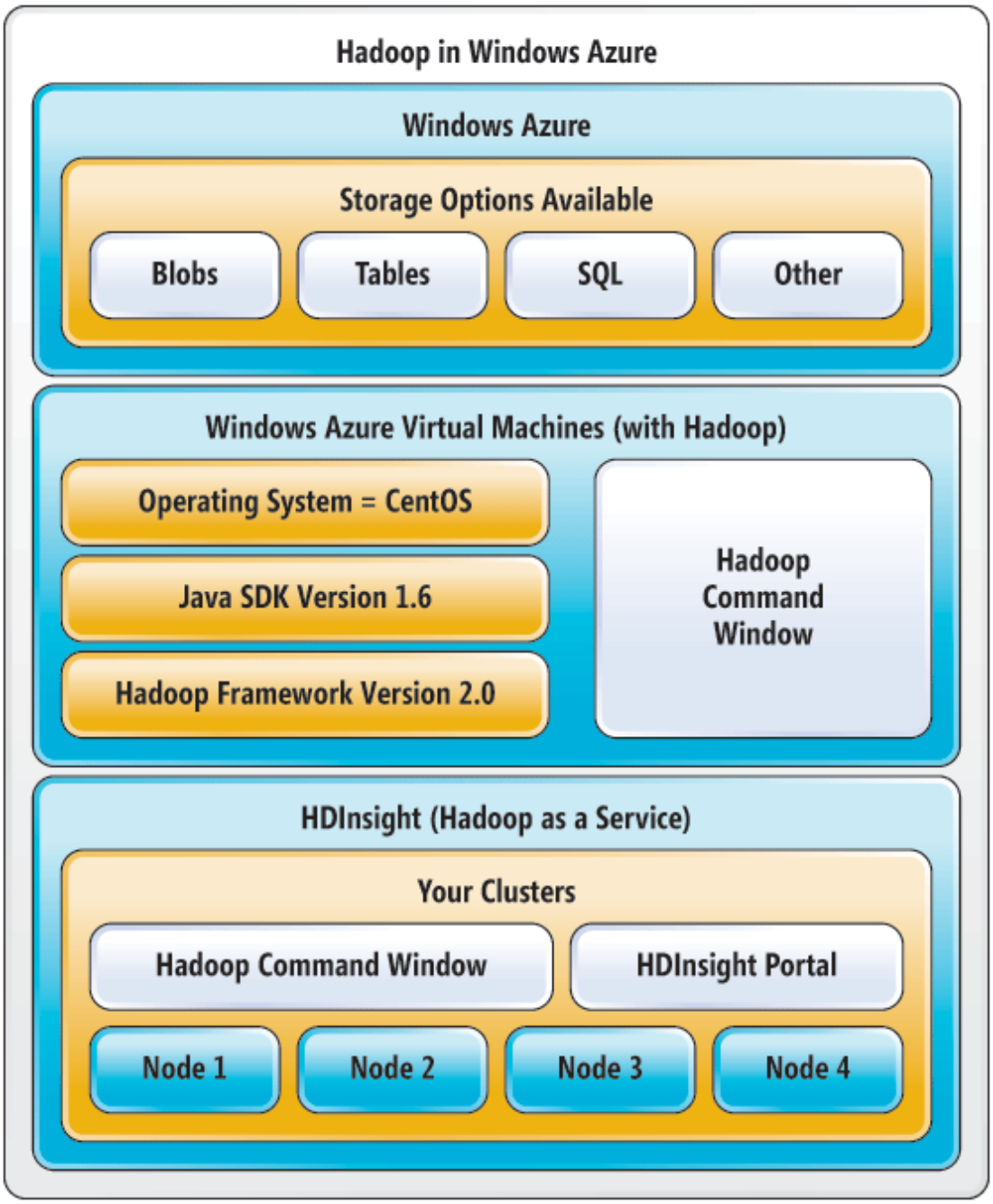
前期成本。无需花费大量时间进行安装或设置, Microsoft Azure 可以为您完成这些工作。您可在几分钟内启动第一个群集。

2.1 HDInsight 到底是什么

HDInsight 是一个可以在 Azure 云中部署并且提供 Apache Hadoop 集群的服务，提供了对大数据进行管理，分析和报表的软件工具框架。它与 Hadoop 有着类似的生态圈，但又具有 Microsoft 的种种特色：



Hadoop Ecosystem in Microsoft Azure



2.2 Why HDInsight

HDInsight is an Apache Hadoop implementation that runs in globally distributed Microsoft datacenters. It' s a service that allows you to easily build a Hadoop cluster in minutes when you need it, and tear it down after you run your MapReduce jobs. As Microsoft Azure Insiders,

we believe there are a couple key value propositions of HDInsight. The first is that it’ s 100 percent Apache-based, not a special Microsoft version, meaning that as Hadoop evolves, Microsoft will embrace the newer versions. Moreover, Microsoft is a major contributor to the Hadoop/Apache project and has provided a great deal of its query optimization know-how to the query tooling, Hive.

微软希望通过支持 Windows Server 和 Microsoft Azure 的 Hadoop 发布版，提供可移植、性能优越、安全且易部署等特性，促进 Hadoop 的应用。微软还将通过在 HDInsight 中集成 Active Directory 来增强 Hadoop 的安全性。此举将使 IT 部门能够将同样的一致性安全策略用于包括 Hadoop 集群在内的所有 IT 资产。

此外，通过与 System Center 集成，HDInsight 简化了 Hadoop 的管理，并支持 IT 部门在同一面板上管理 Hadoop 集群、SQL Server 数据库和应用程序。

基于 Hadoop 的 Windows 平台应用程序集成了如 Excel、Power View 和 PowerPivot 等微软的商业智能（BI）工具，可以很容易地分析大量的业务信息，从而创造独特的、差异化的商业价值。
为实现与 Apache Hadoop 百分之百的兼容性，微软的 Hadoop 发布版 HDInsight 是基于 Hortonworks Data Platform（HDP）构建的。因此，客户能够将其 MapReduce 作业从自己的 Windows 服务器移到云中，甚至是移到运行在 Linux 上的 Apache Hadoop 发布版中。目前还没有其他厂商提供该

功能。此外，在 Windows Server 和 Azure 平台上提供这些功能，也使客户能够利用熟悉的工具（如 Excel、PowerPivot for Excel 和 Power View）轻松地从数据中抽取可行的观点。

3. HDInsight HBase 的概述

3.1 什么是 HBase 的?

HBase 的是建立在的 HadoopApache 的开源的 NoSQL 数据库，它提供了大量的非结构化和半结构化数据的随机存取能力强的一致性。它是仿照谷歌的 BigTable，是一个以家庭为中心的列式数据库。数据被存储在一个行内的表和数据的行由列族分组。 HBase 的是在这个意义上，无论是列也不存储在其中的数据的类型，需要使用它们之前，定义一个无模式数据库。开放源代码是首次发布由 Mike Cafarella 于 2007 年，线性扩展处理 PB 级数据的数千个节点。它可以依赖于数据的冗余，批量处理和通过在的 Hadoop 生态系统的分布式应用程序中提供的其他功能。

3.2 什么是 AzureHDInsight HBase 的?

HDInsight 的 HBase 提供一个管理的集群集成到 Azure 环境。该簇被配置为直接在 Azure 斑点存储，这提供了在性能/成本选择低等待时间和增加的弹性存储数据。这使客户能够构建大型数据集工作的交互式网站，构建存储传感器和遥测数据，从数以百万计的端点的服务，以及分析这些数据与 Hadoop 作业。 HBase 的和的 Hadoop 都是很好的出发点，在 Azure 大数据项目，特别是，可以实现实时应用与大型数据集工作。

在 HDInsight 实现利用 HBase 的的横向扩展架构，可提供自动分片表，强一致性读取和写入，和自动故障转移。性能提高了内存高速缓存的读取和高通量流式写入。虚拟网络的配置也可用于 HDInsight HBase 的。

3.3 如何在 HDInsight HBase 的数据管理?

数据可以在 HBase 的使用创造 GET，PUT 和扫描从 HBase 的 shell 命令进行管理。数据通过表决，并阅读使用 get 命令写入到数据库中。扫描命令用于获得在一个表中，从多行数据。数据也可以使用 HBase 的 C＃的 API，它提供了一个客户机库的 HBase 的 REST API 的顶端管理。一个 HBase 的数据库也可以使用 Hive 查询。

场景：什么是用例 HBase 的?

BigTable，推而广之，HBase 的创建为其典型用例是网页搜索。搜索引擎建立一个映射条款，包含它们的网页索引。但也有很多其他的用例 HBase 的适用哪几个的，都逐项本节。

用例 1：key-value 存储

HBase 的可作为一个键值存储，适用于管理信息系统。 Facebook 的 HBase 的使用他们的邮件系统，它是理想的存储和管理网络通信。 WebTable 使用 HBase 的搜索和管理从网页中提取表。

用例＃2：传感器数据

Hase 的是用于捕获是从各种来源的增量收集的数据是有用的。这包括社交分析，时间序列，保持交互式仪表盘了解最新的趋势和专柜，以及管理审计日志

系统。例子包括彭博交易终端和开放时间序列数据库（OpenTSDB），它存储并提供访问收集了服务器系统的健康指标。

用例 3：实时查询

Phoenix 是 Apache HBase 的一个 SQL 查询引擎。它是作为一个 JDBC 驱动程序和能使查询和使用 SQL 管理 HBase 的表。

用例 4：HBase 的一个平台

应用程序可以在 HBase 的顶部使用它作为数据存储上运行。例子包括凤凰城，OpenTSDB，KIJI，和 Titan。应用程序还可以整合 HBase 的。例子包括 Hive，Pig，Solr 的，风暴，水槽，黑斑羚，星火，神经节和钻孔。

4. 使用 HDInsight 进行开发

Microsoft AzureHDInsight 提供了运行 Apache Hadoop 的动态供应群集来处理大数据(Big Data)的能力。您可以在这个系列的第一篇博客中找到更多信息，您也可以点击这里开始在 Microsoft Azure 门户网站中使用它。这篇文章列举了开发人员与 HDInsight 交互的几种不同方法，首先通过讨论不同的场景，然后深入讨论 HDInsight 中各种不同的功能。因为我们的产品是建立在 Apache Hadoop 之上，所以开发人员可以利用一个有广泛且丰富的工具和功能的生态系统。

说起场景，就我们合作过的客户而言，有两个截然不同的情形，创建，使用工具来处理大数据的工作，以及**在应用程序中整合 HDInsight，将工作的**输入和输出整合为一个较大的应用程序架构的一部分。HDInsight 的一个关键设计是集成了 Microsoft Azure Blob Storage 作为默认的文件系统。这意味着您可

以使用现有的和工具和 API's 访问 blob 存储中的数据。该文章更详细地解释了我们如何利用 Blob Storage。

就创建工作这一点而言，有大量的工具可用。深层次说，它有一套作为现有 Hadoop 生态系统的一部分的工具，以及一组我们建立的项目帮助.NET 开发人员开始学习 Hadoop，同时我们已经开始了新的项目帮助开发人员利用 JavaScript 与 Hadoop 交互。

4.1 创建工作

4.2 现有的 Hadoop 工具

HDInsight 是通过 Hortonworks Data Platform 来使用 Apache Hadoop，对于 Hadoop 的生态系统有很高的保真度。因此，许多功能都和原来的完全一样。这意味着你在下面列出的任何工具的投资和知识都在 HDInsight 中可用。分布式处理群集由下面的 Apache 项目创建：

- Map/Reduce
    - 在 Hadoop 上，Map/Reduce 是分布式处理的基础。为了编写工作，程序员可以使用 Java，或者通过 Hadoop Streaming 使用其他语言和运行时.
    - 这里提供了在 HDInsight 上编写 Map/Reducejobs 的简易指南.
- Hive
    - Hive 使用一种类似于 SQL 的语法来表达编译一组被编译成 Map/Reduce 程序的查询。Hive 支持 SQL 中的许多结构（聚合、分组、 过滤等），并轻松地在您的群集中的各节点并行化这些查询。

    - 这里提供了使用 Hive 的方法
- Pig
    - Pig 是一种数据流语言，使用一种叫做 Pig Latin 的语言编译成一系列的 Map/Reduce 程序。
    - 这里提供了在 HDInsight 上使用 Pig 的入门指南.
- Oozie
    - Oozie 是一种工作流调度程序，用来管理行动的有向无循环图，其中的行动可以是 Map/Reduce, Pig, Hive 或其他工作。下表表示为当前预览版中各组件的版本：

Apache Hadoop
Apache Hive
Apache Pig
Apache Sqoop
Apache Oozie
Apache HCatalog
Apache Templeton

此外，Hadoop 空间的其他项目，例如 Mahout（参见示例） 或 Cascading，也可以方便地用在 HDInsight 之上。有关这些主题我们将在今后另外写文章介绍。

4.3 .NET 工具

我们正努力开发一组工具让开发人员能利用他们的.NET 技能和投资来使用 Hadoop。这些项目放在 CodePlex 上，你可以从 NuGet 上下载这个工具包创建运行在 HDInsight 上的工作。

- .NET Map/Reduce
- LINQ to Hive

4.4 运行工作

想要运行任意一项工作，有如下几个方法：

- 直接从头节点运行它们。若要执行此操作，远程连接到您的群集，打开 Hadoop 命令提示符，并直接使用命令行工具
- 在群集上使用 REST API's 远程提交它们
- 利用 HDInsight 仪表板上的工具。创建您的群集后，群集仪表板中提供了一些功能用来提交工作：
    - 创建工作
    - 交互式控制台

4.5 集成 HDInsight 到您的应用程序

4.6 打开 REST API's

为了提供一个简单的接口供客户端应用程序集成，我们努力确保群集上的所有功能都通过一组安全的 REST API's 暴露给客户端。

- WebHCatalog— — 元数据管理及远程工作提交、 历史记录和管理
- Ambari— — 运行中群集的监测
- Oozie— — 管理和调度 Oozie 工作流

我们目前已经对这些 API 提供了.NET 客户端类库，点击这里下载，你能够自行在其他语言中轻松地使用 HTTP 堆栈构建客户端。

4.7 通过 ODBC 连接

利用 ODBC 客户端，就可以轻松地整合现有的应用程序 (例如 Excel) 访问存储在 HDInsight 上 Hive 表中的数据。

4.8 调试/测试

为了能在 Azure 上不连接群集都可以工作，我们开发了 HDInsight Developer Preview，你可以轻松从 Web Platform Installer 上一键安装。您可以利用它通过一小组数据来实验、调试和测试前面提及的所有技术。然后您可以将项目部署到 Azure 并运行 Blob Storage 上的大数据。若要安装它，只需在 Web Platform Installer 上搜索 HDInsight，或点击这里直接从 web 安装。

5. Spark 集群三种部署模式的区别

Spark 最主要资源管理方式按排名为 Hadoop Yarn, Apache Standalone 和 Mesos。在单机使用时，Spark 还可以采用最基本的 local 模式。

目前 Apache Spark 支持三种分布式部署方式，分别是 standalone、spark on mesos 和 spark on YARN，其中，第一种类似于 MapReduce 1.0 所采用的模式，内部实现了容错性和资源管理，后两种则是未来发展的趋势，部分容错性和资源管理交由统一的资源管理系统完成：让 Spark 运行在一个通用的资源管理系统之上，这样可以与其他计算框架，比如 MapReduce，公用一个集群资源，最大的好处是降低运维成本和提高资源利用率（资源按需分配）。本文将介绍这三种部署方式，并比较其优缺点。

1. Standalone 模式

即独立模式，自带完整的服务，可单独部署到一个集群中，无需依赖任何其他资源管理系统。从一定程度上说，该模式是其他两种的基础。借鉴 Spark 开发模式，我们可以得到一种开发新型计算框架的一般思路：先设计出它的 standalone 模式，为了快速开发，起初不需要考虑服务（比如 master/slave）的容错性，之后再开发相应的 wrapper，将 stanlone 模式下的服务原封不动的部署到资源管理系统 yarn 或者 mesos 上，由资源管理系统负责服务本身的容错。目前 Spark 在 standalone 模式下是没有任何单点故障问题的，这是借助 zookeeper 实现的，思想类似于 Hbase master 单点故障解决方案。将 Spark standalone 与 MapReduce 比较，会发现它们两个在架构上是完全一致的：

1） 都是由 master/slaves 服务组成的，且起初 master 均存在单点故障，后来均通过 zookeeper 解决（Apache MRv1 的 JobTracker 仍存在单点问题，但

CDH 版本得到了解决）；

2) 各个节点上的资源被抽象成粗粒度的 slot，有多少 slot 就能同时运行多少 task。不同的是，MapReduce 将 slot 分为 map slot 和 reduce slot，它们分别只能供 Map Task 和 Reduce Task 使用，而不能共享，这是 MapReduce 资源利率低效的原因之一，而 Spark 则更优化一些，它不区分 slot 类型，只有一种 slot，可以供各种类型的 Task 使用，这种方式可以提高资源利用率，但是不够灵活，不能为不同类型的 Task 定制 slot 资源。总之，这两种方式各有优缺点。

2. Spark On Mesos 模式

这是很多公司采用的模式，官方推荐这种模式（当然，原因之一是血缘关系）。正是由于 Spark 开发之初就考虑到支持 Mesos，因此，目前而言，Spark 运行在 Mesos 上会比运行在 YARN 上更加灵活，更加自然。目前在 Spark On Mesos 环境中，用户可选择两种调度模式之一运行自己的应用程序（可参考 Andrew Xia 的“Mesos Scheduling Mode on Spark”）：

1） 粗粒度模式（Coarse-grained Mode）：每个应用程序的运行环境由一个 Dirver 和若干个 Executor 组成，其中，每个 Executor 占用若干资源，内部可运行多个 Task（对应多少个“slot”）。应用程序的各个任务正式运行之前，需要将运行环境中的资源全部申请好，且运行过程中要一直占用这些资源，即使不用，最后程序运行结束后，回收这些资源。举个例子，比如你提交应用程序时，指定使用 5 个 executor 运行你的应用程序，每个 executor 占用 5GB 内存和 5 个 CPU，每个 executor 内部设置了 5 个 slot，则 Mesos 需要

先为 executor 分配资源并启动它们，之后开始调度任务。另外，在程序运行过程中，mesos 的 master 和 slave 并不知道 executor 内部各个 task 的运行情况，executor 直接将任务状态通过内部的通信机制汇报给 Driver，从一定程度上可以认为，每个应用程序利用 mesos 搭建了一个虚拟集群自己使用。

2) 细粒度模式（Fine-grained Mode）：鉴于粗粒度模式会造成大量资源浪费，Spark On Mesos 还提供了另外一种调度模式：细粒度模式，这种模式类似于现在的云计算，思想是按需分配。与粗粒度模式一样，应用程序启动时，先会启动 executor，但每个 executor 占用资源仅仅是自己运行所需的资源，不需要考虑将来要运行的任务，之后，mesos 会为每个 executor 动态分配资源，每分配一些，便可以运行一个新任务，单个 Task 运行完之后可以马上释放对应的资源。每个 Task 会汇报状态给 Mesos slave 和 Mesos Master，便于更加细粒度管理和容错，这种调度模式类似于 MapReduce 调度模式，每个 Task 完全独立，优点是便于资源控制和隔离，但缺点也很明显，短作业运行延迟大。

3. Spark On YARN 模式

这是一种很有前景的部署模式。但限于 YARN 自身的发展，目前仅支持粗粒度模式（Coarse-grained Mode）。这是由于 YARN 上的 Container 资源是不可以动态伸缩的，一旦 Container 启动之后，可使用的资源不能再发生变化，不过这个已经在 YARN 计划中了。

spark on yarn 的支持两种模式：

1) yarn-cluster：适用于生产环境；

2) yarn-client：适用于交互、调试，希望立即看到 app 的输出

yarn-cluster 和 yarn-client 的区别在于 yarn appMaster，每个 yarn app 实例有一个 appMaster 进程，是为 app 启动的第一个 container；负责从 ResourceManager 请求资源，获取到资源后，告诉 NodeManager 为其启动 container。yarn-cluster 和 yarn-client 模式内部实现还是有很大的区别。如果你需要用于生产环境，那么请选择 yarn-cluster；而如果你仅仅是 Debug 程序，可以选择 yarn-client。

总结：

这三种分布式部署方式各有利弊，通常需要根据实际情况决定采用哪种方案。进行方案选择时，往往要考虑公司的技术路线（采用 Hadoop 生态系统还是其他生态系统）、相关技术人才储备等。上面涉及到 Spark 的许多部署模式，究竟哪种模式好这个很难说，需要根据你的需求，如果你只是测试 Spark Application，你可以选择 local 模式。而如果你数据量不是很多，Standalone 是个不错的选择。当你需要统一管理集群资源（Hadoop、Spark 等），那么你可以选择 Yarn 或者 mesos，但是这样维护成本就会变高。

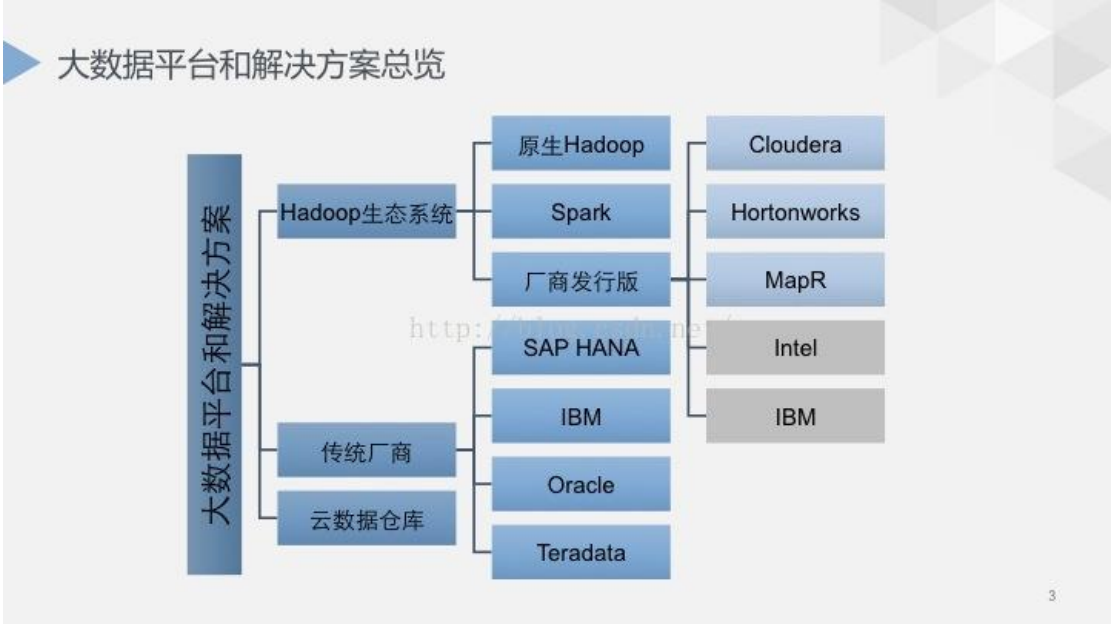
· 从对比上看，mesos 似乎是 Spark 更好的选择，也是被官方推荐的

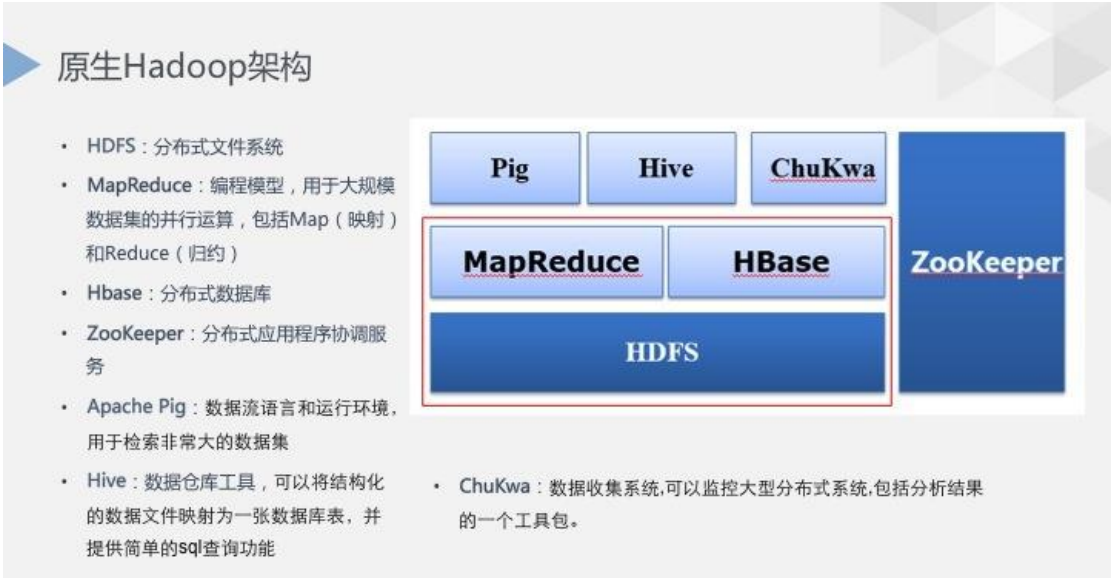
· 但如果你同时运行 hadoop 和 Spark,从兼容性上考虑，Yarn 是更好的选择。 · 如果你不仅运行了 hadoop，spark。还在资源管理上运行了 docker，

Mesos 更加通用。

· Standalone 对于小规模计算集群更适合!

6. 主流大数据平台及解决方案对比





原生Hadoop的优劣

- Hadoop框架很有吸引力，提供给企业分析数据的能力
- 扩展性强
- 成本低
- 可灵活处理结构化数据和非结构化数据
- 扩展和优化Hadoop 集群涉及大量编程工作，对于数据分析开发人员是障碍
- 原本设计不具备太多安全功能
- 与现存数据库和应用的集成较为困难
- 需要使用独立的扩展查询语言HQL

实施企业需要具备较强的编程开发能力
适用于规模大、数据量大、数据多样化的企业
虽然开源，但存在隐性开销

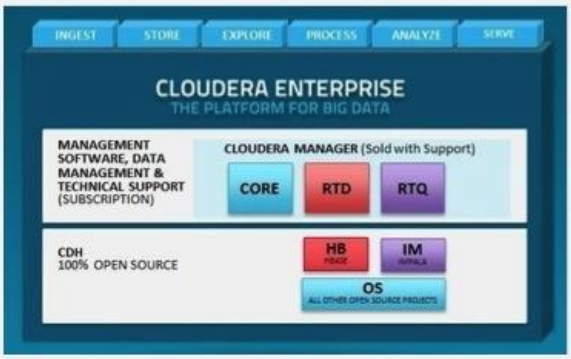
Spark

- Spark最初由伯克利大学AMPLab于2009年启动，2010年开源，2013年成为Apache软件基金会资助的项目，2014年2月成为Apache顶级项目。今年发布最新版本为2.0.
- 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。
- Shark (Hive on Spark): Shark基本上就是在Spark的框架基础上提供和Hive一样的HQL命令接口，为了最大程度的保持和Hive的兼容性
- Spark streaming: 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。
- Bagel: Pregel on Spark，可以用Spark进行图计算

Spark VS Hadoop

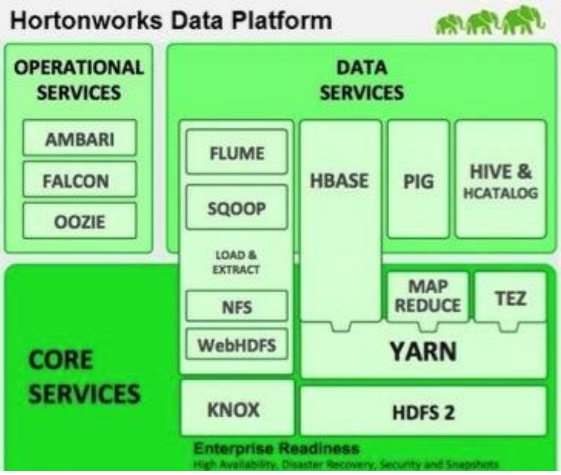
	Hadoop	Spark
文件系统	HDFS	支持HDFS、MESOS、S3等文件系统，可以直接将Spark集成到Hadoop上，可以从HDFS读取和写入文件
中间结果存储	存储到磁盘	内存存储
开发语言	Java	Scala、Java、Python
易用性	Java API、无交互式界面	提供丰富的Scala、Java、Python API及交互式Shell来提高可用性
容错性	数据冗余、任务失败重计算	Checkpoint机制，RDD支持重计算
通用性	只提供了Map和Reduce两种操作	提供多种数据集操作类型，把map, filter, flatMap, sample等多种操作类型统称为Transformations。同时还提供Count, collect, reduce, lookup, save等多种actions操作。
性能	频繁读写磁盘、低	数据缓存内存、高
应用场景	适用于大数据量、迭代次数少、无时延要求的业务	适用于中等数据量，需要多次操作特定数据集，且频繁迭代计算的数据业务场合

Cloudera



- 分析数据库管理系统:Hbase，以及Cloudera Impala，支持SQL在Hadoop顶层的查询。
- 内存数据库管理系统:支持Apache Spark作为Hadoop顶层的内存分析
- Hadoop分布式系统:CDH开源分布式系统、Cloudera标准版（Standard）、Cloudera企业版（Enterprise）
- 流处理技术:包括Storm（风暴）的Hadoop上开源流处理
- 硬件/软件系统:合作伙伴工具和预设硬件，两者也可来自Cisco、Dell、HP、IBM、NetApp和Oracle等系统。

HortonWorks



- 分析数据库管理系统:Hbase，Hive，作为Hortonworks提供的在Hadoop顶层实现SQL查询的不错选择
- 内存数据库管理系统:支持Apache Spark作为Hadoop顶层进行内存分析
- Hadoop分布式系统:Hortonworks数据平台（HDP）2.0，HDP for Windows，Hortonworks Sandbox
- 流处理技术:Hadoop上的开源流处理技术选项，包括Storm
- 硬件/软件系统:合作伙伴工具和预配置的硬件，或都可从HP、Teradata和其它平台上获得

MapR

- 分析数据库管理系统:HBase，支持Drill、Hive、Impala、Shark和其它Hadoop上SQL查询选项
- 内存数据库管理系统:MapR通过Drill和Shark等的开源项目来实现内存分析
- Hadoop分布式系统:MapRM3、MapRM5、MapRM7
- 流处理技术:支持通过Storm或与Informatica HParser整合的方式进行的流分析
- 硬件/软件系统:硬件配置可通过包括Cisco、HP、IBM和NetApp在内的合作伙伴获得

Cloudera、Hortonworks、MapR横向对比

	cloudera Ask Bigger Questions	Hortonworks	MAPR TECHNOLOGIES
开发特点	开源组件为辅，专注功能基础的专有技术	关注开源组件的完善	开源组件为辅，专注功能基础的专有技术
盈利模式	工具产品路线，收入以来软件授权费用	收入依赖于产品支持和服务	工具产品路线，收入以来软件授权费用
管理组件	提供额外管理组件		提供额外管理组件
发布版优点	CDH提供用户友好界面和其他易用的工具，如Impala	唯一支持windows平台的Hadoop发布版	最快的、带有多节点直接访问功能的Hadoop发布版
不足	CDH相比较MapR稍慢	AMBARI客户端的功能较基础，不具备丰富功能	客户端界面不如CDH
相似点	• 均使用Hadoop核心框架并捆绑企业应用，提供应用支持服务和订阅服务 • 均提供免费下载版 • 均有相应的技术社区		

SAP HANA

- HANA是一个软硬件结合体，提供高性能的数据查询功能，用户可以直接对大量实时业务数据进行查询和分析。用户拿到的是一个装有预配置软件的设备。
- 基于内存计算技术的高性能实时数据计算平台
- SAP HANA不会替代BW，其初衷是一个用于极致性能的通用数据库和应用平台，BW则类似一个应用软件，是构建和维护数据仓库的工具。

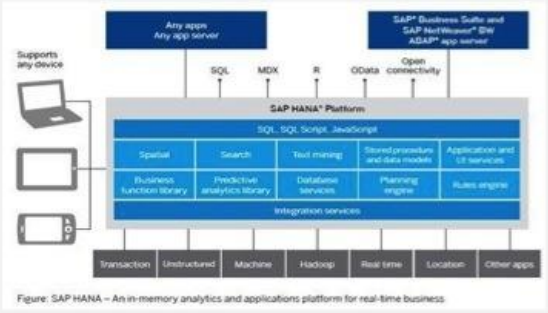
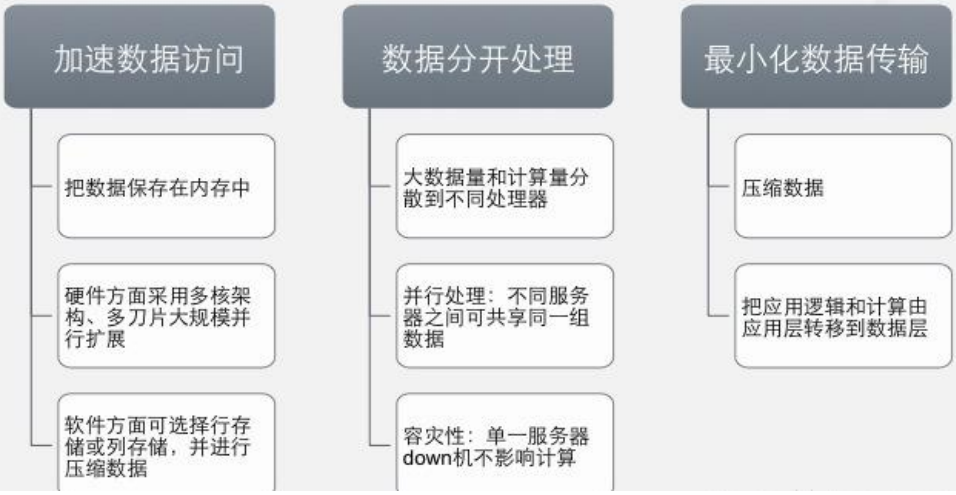


Figure: SAP HANA – An in-memory analytics and applications platform for real-time business

- 分析数据库管理系统:SAPHana、SAPIQ
- 内存数据库管理系统:SAPHana
- 流分析选项:SAP事件流处理（Event Stream Processing）
- Hadoop分布式系统:代售并支持Hortonworks、Intel，由Cloudera和MapR认证的Hadoop集群
- 硬件/软件系统:多个硬件配置合作伙伴，包括Dell、Cisco、Fujitsu（富士通）、Hitachi（日立）、HP和IBM

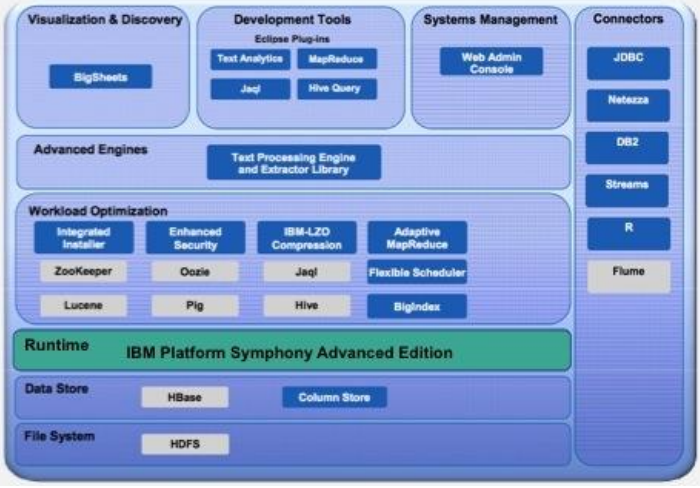
SAP HANA性能特点



IBM大数据平台

IBM的BigData方案

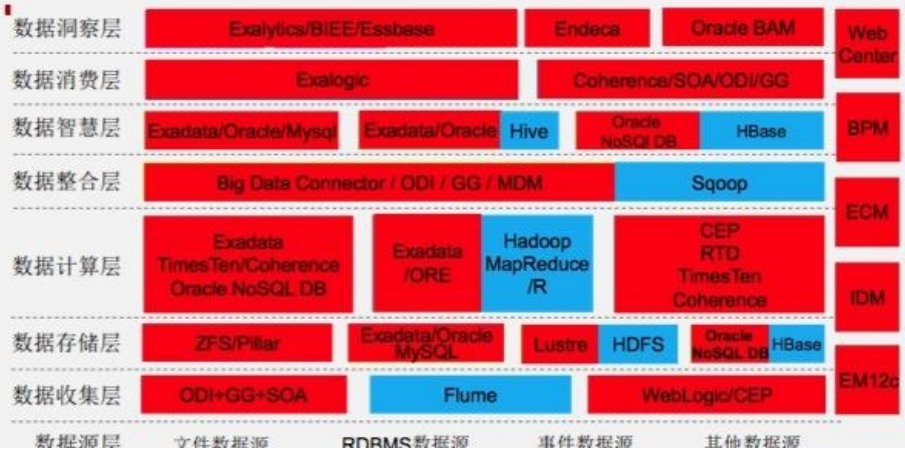
IBM BigInsights & Platform Symphony



- IBM Platform Symphony 替代了开源Hadoop中的原生工作和任务跟踪设施，采用了经优化的低延迟MapReduce实现方式，完全兼容开源Hadoop以提供增强的容量。

Oracle大数据解决方案

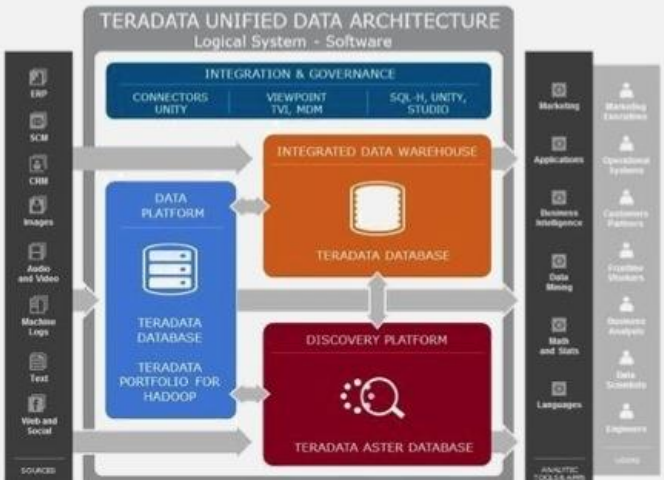
- 从软件上，Oracle的大数据解决方案更多以来现有Oracle产品，同时整合Cloudera的相关Hadoop产品组成解决方案。



Oracle大数据解决方案

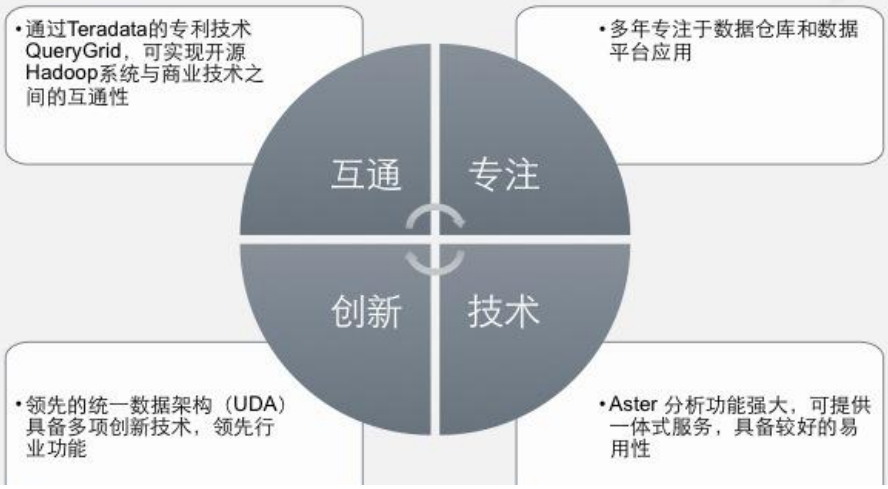
- 解决方案中，Oracle将内存性能与其旗舰数据库关联，使用Oracle Times Ten内存数据库与其硬件相配合，从而实现与SAP HANA相竞争的解决方案。

Teradata大数据平台



- 分析数据库管理系统:Teradata、Teradata Aster
- 内存数据库管理系统:虽然并不是一个内存数据库管理系统，但Teradata智能存储监视器仍实现了对最热数据的查询，并且自动将这些数据送至可用的最快速存储层，附带一些选项，包括RAM（随机存取存储器）、flash、SSD，以及不同速度的传统旋转磁盘。
- 流分析选项:无
- Hadoop分布式系统:代售并支持Hortonworks数据平台
- 硬件/软件系统:Teradata和Teradata Aster是集成的软硬件系统。Hadoop由两个Teradata组件和标准的Dell配置来

Teradata特点



传统厂商的横向对比

	SAP HANA	IBM BigInsights	Oracle	Teradata
开发特点	• 基于内存的管理平台	• 在Hadoop平台基础上优化的专属解决方案	• 现有产品与Hadoop平台的结合	• 现有产品与Hadoop平台的结合
优势	• SAP专属的分布式大数据平台，数据运算效率高，与SAP其他组件的配合度高	• 在Hadoop原生平台上的改进升级 • 与IBM其他工具配合度较好	• 与Oracle本身硬件结合，方案完整性较好	• UDA数据架构，使得联通性和扩展性强 • Aster数据分析功能强大
不足	• 与其他Hadoop平台组件的连接、配合不足	• 组件、工具多，全面掌握难，开发难度高	• 更多依赖Oracle原有工具 • 需要大规模硬件投资	
适用场景	• 已使用SAP ERP及其他产品的公司	• 已建立IBM数据仓库的公司； • 要求少量开发，同时寻求全面解决方案的公司	• 可进行大规模投资的企业，并且具备对于Oracle各产品组件相对熟悉的开发能力	• 数据平台的初期投资，对扩展性和灵活性要求高的企业

云计算分析平台