第2章

Microsoft 大数据解决方法

本章内容：

- 认识 Microsoft 大数据战略布局
- Hadoop 生态系统的竞争性
- 决定如何部署 Hadoop

在第 1 章我们学习了构成 Hadoop 生态系统的各类项目的一些知识。本章将集中讨论 Microsoft 的大数据解决方法，并更深入探讨 Hadoop 中更具竞争力的要素。最后，我们将看看部署 Hadoop 的注意事项并评估部署选项。我们将考虑这些部署因素如何在拓扑结构中体现，以及如何予以缓解。

2.1 “优质组合”的故事

时光回溯至2011年，高级副总裁Ted Kummert在PASS峰会演讲中正式宣布Hortonworks成为 Microsoft“大数据”核心战略合作伙伴。这让所有人都吃了一惊。

我们这些 Microsoft 追随者一直在期待它发布一款分布式扩展计算的专利产品(比如名为 Dryad 的 Microsoft 研究项目)，但事与愿违。Microsoft 在合作关系上选择进行投资并与开源社区合作，使 Hadoop 运行于 Windows 平台并和 Microsoft 工具协同工作。这个决定不仅仅是胆大妄为，简直是史无前例。

在那一周晚些时候，Dave DeWitt 在答疑会上发表了对“市场的声音”和选择 Hadoop 的见解。我们更深入地洞察了 Microsoft 做出这个决定的原因，他们起步太晚，已经错过了发布专属产品的最佳时期。但这仅是故事的开始。尽管 Hadoop 的核心是开源的，但是竞争依然无处不在，大量基于 Hadoop 的专利产品已经问世。Microsoft 能开发出专利组件吗？没人知道。重要的是前车可鉴，产品公司期望投资能转化为收益，看来势必会有更多基于 Hadoop 的专利产品陆续问世。

Microsoft 涉足大数据和开源解决方案领域(Open Source Solution，OSS)使更多的战略重心转移到了 Windows Azure 云计算方面并且有了广泛的重叠。这导致了大数据战略中一些很有趣的结果——那些没有这种变化根本不会实现的结果。你能想象到 Linux 会成为 Microsoft 数据平台的一部分吗？反正我是没有！

在有了这些想法后，我强烈建议您继续往下读，学习关于这个引人入胜的生态系统的更多知识。搞清楚 Microsoft 与开源世界的关系，清晰认识 Apache Hadoop 集群的部署抉择。

注意：

假如想了解更多关于 Dryad 项目的信息，http://research.microsoft.com/en-us/projects/dryad/是一个很好的启蒙网站，你会发现一些不可思议的相似之处。

2.2　生态系统中的竞争

Hadoop 的开源性并不意味着就没有竞争，实际上恰恰相反。大多数情况有点类似于玩牌的时候亮牌，大家都可以看到对方的牌，直到看不到牌面之后情况才有会有所改变。许多系统将开源技术作为专利扩展混合组件的一部分，这些专利技术就好像将牌面遮住继而增强了竞争力。我们之后学习 Cloudera 的 Impala 技术时会分析一个这种实例。

Hadoop 当然也不能免俗。为在市场中体现差异性，Hadoop 代理商选择多元化介入而不是仅仅通过单一项目或方案的协作。让我们通过 SQL on Hadoop 领域来突出体现这些是如何运作的。没有哪个领域比下一代 SQL on Hadoop 对分发版的未来更重要，更具有争议性。

2.2.1　SQL on Hadoop 现状

回顾第 1 章：SQL on Hadoop 通过 Hive 项目应运而生，Hive 通过名为 HQL 的类 SQL 语言将 MapReduce 的复杂程度抽象化。注意，这并不意味着 Hadoop 会突然就遵循事务处理的 ACID 规则(原子性、一致性、隔离性、持久性，事务处理的四要素，简称 ACID)，更像是 Hadoop 以 Hive 为终端提供了一种熟悉的查询语法结构。但需要强调的是，Hive 仅能处理 Hadoop 中的数据。

Hive 的挑战在于对 MapReduce 的依赖性。由于 MapReduce 执行引擎和调度功能的紧密结合性，Hive 别无选择只能基于 MR。但是 Hadoop 2.0 和 YARN 项目改变了整个状况。通过将调度分离到单独项目并与执行解耦，Hive 进化呈现出了新的可能性。

2.2.2　Hortonworks 和 Stinger

Hortonworks 已将大部分精力都投入到了 Stinger 上。Stinger 本身并不是 Hadoop 项目，它是一个显著提高 Hive 性能和完善性的方案，目标是将 Hive 处理速度提高 100 倍，这绝非易事。Stinger 的有趣之处在于，所有编码直接对 Hadoop 项目生效，这使得每个人都在改变中受益。这也完全符合 Hortonworks 的承诺和 Hadoop 章程。

那么到底什么是 Stinger？它包含 3 个阶段，前两个已经实现交付。

1. Stinger 阶段 1

阶段 1 主要针对现有 Hive 架构进行优化。已于 2013 年 5 月在 Hive 0.11 中得以交付使用，成为 Hortonworks 数据平台(Hortonworks Data Platform，HDP)1.3 版本的一部分。阶段 1 实现了 3 个重要更改：

- 优化的 RC 文件(Optimized RC file，ORCFile)：对 ORC 文件格式的优化显著改善了 Hive 的数据访问模式。通过在文件和数据块级别增加元数据使查询语句的运行速度更快。另外，只有必要的列从 HDFS 读取，这和 SQL Server 的列存储技术非常类似；降低了 I/O，进一步提升性能。

注意：

ORCFile 支持优化纵栏式记录文件。这种文件格式允许数据被水平(行)垂直(列)划分，实质上就是 Hadoop 的列存储。

- SQL 兼容性：引进十进制数据类型，添加 Truncate 语法，包括窗口函数功能，除 OVER 子句外，Hive 已经可以额外支持 RANK、LAG & LEAD、FIRST & LAST 和 ROW_NUMBER。对核心语法也做了一些改进，使 GROUP BY 可以识别别名，也包含了 ALTER VIEW。
- 查询和连接(JOIN 运算符)优化：与绝大多数数据库软件一样，查询优化是普遍特色，Hive 0.11 自然也不例外。Hive 对此有两个主要更改。首先是移出了计划任务中的冗余运算符，研究发现这类运算符在简单查询中会消耗 10%的 CPU。其次是用 JOIN 运算符去释放 MAPJOIN hint 方式。这是另一个更改，将 Hive.auto.convert.join 的默认配置更改为 true(即 on)。

2. Stinger 阶段 2

阶段 2 于 2013 年 10 月作为 Hive 0.12 的一部分被发布实施。注意此次发布距离阶段 1 发布仅 5 个月而已，Stinger 背后的社区团队正在快速向前推进。

按照 Stinger 针对速度、规模和 SQL 三管齐下的发展方针，阶段 2 需要切换到 Hadoop 2.0。这让 Hive 工程师能利用 YARN，并为 Tez 打好基础。

注意：

参考第 1 章中对 Hadoop 项目 YARN 和 Tez 的定义。

阶段 2 包含以下增强功能：

- 性能：得益于阶段 2 的诸多更改，查询变得更快。一个名为关联优化器的新逻辑型查询优化器被引入。它的功能就是将多个关联的 MapReduce 作业合并成为单个作业以减少数据移动。ORDER BY 被定义为一个并行操作。另外，谓词下推的实现

使 ORCFile 可以忽略行，与 SQL Server 中的段忽略类似。同样还使用 hive.map.groupby.sorted 配置属性为 COUNT(DUSTINCT)添加了优化。

- SQL兼容性：VARCHAR和DATE这两个重要的数据类型被引入。增强了对GROUP BY语法的支持，使其支持结构体类型和联合类型。侧面视图(lateral view)扩展支持“外部”连接行为属性，truncate扩展支持列的截断。针对二进制数据类型添加了新的用户自定义函数功能(User-Defined Functions，UDF)。最终分区切换通过ALTER TABLE..EXCHANGE PARTITION加入产品。

注意：

SQL Server 不支持侧面视图，因为 SQL server 并不支持数组数据类型以及同其交互的功能。访问 https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView 以学习更多关于侧面视图的知识。

- 中止 HCatalog 项目：在 Hive 0.12 中，HCatalog 已经不复存在，因为这个项目已被并入 Hive。

注意：

第 1 章有对 HCatalog 的定义。

3. Stinger 阶段 3

Stinger 阶段 3 仍在开发中，但是可以预料 Hadoop 将引入 Apache Tez，因此将会从批处理转移成为一个具有更多查询/响应交互的引擎。矢量查询(SQL Server 查询处理爱好者的批处理方式)和内存缓存都在规划中。不过，对这个阶段的 Stinger 方案来说一切都言之尚早。

2.2.3　Cloudera 和 Impala

在定义 Hadoop 的 SQL 策略时，Cloudera 选择了一条不同的方向。很明显，他们看到了 MapReduce 的局限性，于是选择实施自己独有的引擎：Impala。

在构建 Impala 时，Cloudera 使用了和 Hortonworks 不同的方法。实际上，他们选择避开整个 Hadoop 中的 MapReduce 遗留问题并重新开始。Cloudera 为 Impala 的运行创建了三个全新的系统服务：

- Impala Daemon
- Impala Statestore
- Impala Catalog Service

1. Impala Daemon

Impala Daemon 作为核心组件运行在 Hadoop 集群的每个节点上。这个处理流程被称为 Impalad，它以分散式多主机的模式运转，也就说任何一个节点都可以是给定查询的控制

“大脑”。因为协调节点是由各个查询决定的，所以许多大规模并行处理系统的常见的单点故障和瓶颈在这种架构下被很恰当地解决了。但注意，提交查询命令连接的 Impala Daemon 将承担协调器的功能，可以通过调用程序来均衡负载，但不是自动的负载均衡。

一旦某个节点被定义为协调器，剩余节点就会被协调器定义为处理器，并在数据子集中执行被分配的任务。每个处理器处理数据并将临时结果反馈给协调器，最后由协调器确定最终结果。

Impala Daemons 不断与 Statestore 进程联系，来确定集群中节点的健康状况和任务接收状态。

2. Impala Statestore

Statestore 是另一个被称为 statestored 的进程，作用是监控所有 Impala 进程，确认它们执行任务的可用性，告知集群中其他 Impala 进程的健康状况。因此它确保了任务不会被分配给目前不可用的节点，由于 Impala 为提高速度牺牲了运行恢复能力，合理分配任务就显得尤为重要。与 MapReduce 不同，经历节点故障的查询会被取消，所以集群越早知道问题越好。

需要注意的是在集群中仅部署一个 Statestore 进程，但这不是一个有效性问题。这个进程对 Impala 的运行而言并不重要，集群会对查询操作的运行时稳定性变得敏感，但是不会离线。

3. Impala Catalog Service

Catalog Service 是第三个进程，名为 catalogd。它的作用是将元数据更改分发给集群中的所有节点。同样，在集群中也仅有一个 Catalog Service 进程运行，而且通常和 Statestore 部署在一个节点上。因为它实际上是利用 Statestore 作为载体将消息传递给 Impala 进程。

Catalog Service 移除了 REFRESH 和 INVALIDATE METADATA 的语句用法，否则在 Impala 中使用数据定义语言(Data Definition Language，DDL)和数据修改语言(Data Modification Language，DML)时它们可能必须运行。通过分发元数据更改，它确保了任何 Impala 进程都被作为调用程序的一部分，能够像一个协调器那样运行，不需要执行任何额外的操作。

和 Statestore 一样，Catalog Service 并不是关键任务。假如 Catalog Service 意外失效，用户在连接中的 Impala 进程中执行插入之后需要执行 REFRESH 语句，或者在进行 DDL 操作之后需要执行 INVALIDATE METADATA 语句。

Impala 是开源的吗？

简单来说是的，Impala 是一个开源产品。但有一个问题，它是 CDH(Cloudera 包含 Apache Hadoop 的分发版)的扩展，最后一点很重要。它仅针对 Cloudera，不能将其用于任何旧的 Hadoop 分发版。因此尽管它是开源软件，但在很多方面是专有的。开源并不意味着没有厂商锁定。

和 Hortonworks 一样，Cloudera 通过提供支持和培训将其在 Hadoop 上的投资转化为收益，Impala 也不例外。实时查询(Real Time Query，RTQ)是 Impala 的技术支持包，并作为 Cloudera 企业版的扩展(基本企业版技术支持产品)。必须同时购买 Cloudera 企业版和 RTQ 才能实现 RTQ。

2.2.4　Microsoft 对 Hadoop 中 SQL 应用的贡献

Microsoft 的初衷其实是想将 Hadoop(特别是 Hortonworks HDP)运行在 Windows 系统上，这是 Mircosoft Hadoop 服务平台——HDIsight 建立的基础。特别是最近，Microsoft 在 Stinger 方案上一直和 Hortonworks 保持着合作。我个人认为，这清晰地指出了 SQL Server 列存储和批处理模式与 Hadoop OCFile 优化和 Tez 中矢量化查询处理之间明显的共通性。Tez 同样为执行复杂的定向无环图(Directed Acyclic Graph，DAG)引入了一种更通用、更有表现力、基于成本的优化器。由于 SQL Server 是当今市面上最精密复杂并基于成本的优化器之一，我确信其开发团队将能为针对 Hadoop 的新型处理模式的发展做出显著积极的贡献。

2.3　Hadoop 的部署

现在你在技术层面和商业角度对环绕在 Hadoop 周围的生态系统有了扎实的了解，你将会考虑亲自尝试部署 Hadoop。

你可能希望在部署 Hadoop 时 Microsoft 能提供若干选择。但是，我们确实需要设定一个基准来协助全方位考虑，评估出最适合既定环境的选项。下一节首先讨论了若干注意事项，进而转向可能的拓扑结构，最终需要一个积分卡来帮助你做出客观决定。

2.3.1　部署要素

部署选项的选择取决于若干因素，其中大部分好像麦片饼干的纤维一样交织在一起。因此通过对本节的学习，有必要记住以下所有要素：

- 弹性
- 灵活性
- 可扩展性
- 安全性
- 相邻性
- 功能性
- 易用性
- 可管理性

1. 弹性

参照橡皮筋来考虑你的弹性需求：

你需要能够拉伸性能需求来满足更快的处理速度或者按照需要应对激增高峰吗？

弹性延伸是云服务的亮点，正如 Windows Azure 提供的服务一样。能随意改变集群的规模和运算能力。有了本地部署服务，尽管缓慢，但始终保持增长，不过绝不可能收缩拓扑结构。曾经购买过这个套件，在上面纠缠了三年——尽管我其实早就不需要它了。

问问自己：

- 你希望当只有少量或者没有工作的时候能够减少开支和产能吗？
- 你知道你的工作量并描述它吗？
- 本质上是可预见和不变的，还是不稳定的？
- 能多么迅速地扩展你的目标环境？
- 这对你重要吗？

这都是很难回答的问题，尤其是在项目开始阶段。每一个部署选项提供了不同的弹性程度，而你对自身环境的理解程度将会是决定所需部署选项的一个重要因素。

2. 灵活性

和弹性密不可分的是灵活性的概念：

- 你明确你的数据处理需求吗？
- 需求是如何动态变化的？
- 驱动项目的愿景有多完整？
- 你已开始大数据的发现之旅，这可能会改变你的想法吗？

在动态变化方面，不同的型号为更好的灵活性提供了契机。硬件采购往往是 3 年固定承诺合同。

3. 可扩展性

你可以很简单地考虑可扩展性因素，回答这个问题：你需要多少个数据节点？

但是这个问题引发了一系列你需要考虑的其他问题：

- 数据节点的位置？
- 如何管理、监控数据节点？
- 谁来管理、监控数据节点？

由于 Hadoop 很像扩展架构，因此首当其冲的数量级问题确实是考虑与部署规模相关的广泛议题的触发点。实际上，规模问题的答案为决定其他因素提供了更多的参考，特别是对灵活性和弹性。

在规模方面，还有关于限制的相关注意事项。例如，Microsoft 在 HDInsight 中最多允许 40 个数据节点。但这不过是人为限制了此服务，其实是可以提升的。从体系结构来看，这是没有限制的。

有人可能会说在本地部署中有同样的问题。当然，世界上最大的集群是本地部署的。但是，现实往往不尽如人意。实际上，Azure存在相同的难题，数据中心必须有能力来处理需求。无论如何，我必须承认我很喜欢为Microsoft制造麻烦。

4. 安全性

Hadoop 没有很完善的方法保障 HDFS 中数据的安全，安全模式的范围由弱到无。因此，在决策过程中达到安全要求的方法是一个重要的考虑因素。在评估所有选项时，除了操作系统和硬件层面，你应该还需要考虑网络层。其他选项包括“默认安全性”配置，在你决定锁定部署时，都是值得完全复制采纳的。

5. 相邻性

应对相邻性问题时，你必须明白数据源于何处。这与许多原因相关，但最主要的是延迟。我们都不希望数据源与分析系统相隔太远，因为假如这样，距离会为数据分析增加延迟。延迟常和费用相关；短小精干的本地网络往往比地理位置分散的网络更便宜，而且对结果影响更小。

数据提供的洞察力价值会随着时间的流逝而大幅下降。因此我们希望尽量保持数据的相邻性来减少在分析时损失的宝贵时间。

另外，系统间距离越远，网络费用越高并且网络会越脆弱。这在为高速传输数据而使用昂贵 InfiniBand 线缆的超低延迟网络拓扑结构中尤其明显。例如，FDR InfiniBand 网络传输速率达到了 56Gbps，但是此性能是基于价格的，所以越短的网络线缆越好。

因此通过简单的例子我们可以看出，假如数据源自云端，那么从常理上而言，提供的分析也应该在云端。这样一来，你的网络将是本地网络，环境之间的传输性能将是局域网传输速度而不是广域网或私有网络传输速度。由于避免了数据出口花费，环境和总成本也将受益。

6. 功能性

尽管可能不会立竿见影，但你需要确定目标平台能提供你想要的功能，不是所有的 Hadoop 分发版都有一样的功能。一个简单例子就是一个相同 Hadoop 分发版中不同版本的 Hive 或 HBase 的支持是不同的。这并不是说你不能为你的部署添加或升级 Hadoop 项目，但当你这样做就意味着你已经脱离了固有的分发版的边界范围。

操作系统的选择同样决定了可用的 Hadoop 分发版和部署选项的范围。

7. 易用性

刚开始使用 Hadoop 时，有时会感到望而生畏，假如你仅使用过微软生态系统，那么这种感觉会更明显。在极端情况下，你可能会去 Apache 网站下载源代码，然后自己编译，最后手动部署代码来建立集群。然而，部署可以像配置向导一样简单。因此，易用性是可变规模，你可接受的启动成本将推动你前行并得到最佳解决方案。

不管选择哪个方案，你可能都想使用自己的 PowerShell 脚本来建立集群，因此它是一个可复制的流程。

8. 可管理性

在考虑部署因素时运营注意事项常常被忽略。思考以下问题：

- 系统如何被监控？
- 支持整个环境需要哪些人力资源？
- 存在什么可用性需求，什么样的灾难恢复策略是合适的？
- 如何应对安全性问题？

这些都是你需要回答的常见运营问题。

2.3.2　部署拓扑结构

我们已经了解了可能影响部署的因素，在继续前进之前，我们可以专注于拓扑结构本身进行相互比较。

在下一节中，我们将比较以下选项：

- 本地部署 Hadoop
- Hadoop 服务基础架构
- Hadoop 服务平台

1. Hadoop 的本地部署

你可按传统方式在本地部署 Hadoop 集群。当今世界上大多数 Hadoop 集群使用 Linux 操作系统，但正如第 1 章中介绍的一样，Hortonworks 有一款适用于 Windows 的分发版。

当选择本地部署 Hadoop 作为部署选择时，最大的挑战是如何确定它的规模，你真正需要多少数据节点？

当然，在考虑清楚后，就需要采购所有的硬件、机柜并进行配置，你还将负责集群的管理和监控工作，因此最好明白如何应对。可能会为此选择 Ambari 项目，或者继续利用 Microsoft 生态系统使用系统中心进行监控。Ambari SCOM 管理套件已经问世，因此可以使用和整个公司一样的基础设施来监控 Hadoop。不管采取何种方式，你必须将新的基础架构集成到环境中并确保所有数据源能访问这个新集群。

那么，为什么要遵循这个模式呢？就个人而言，它和灵活性、相邻性、安全性、功能性都有关联。但这里需要强调的是相邻性。假如数据源自本地并且数量庞大，那么将数据保留在本地部署环境中就更合理。将大量数据推向云端可能是不切实际的，而且在进行分析之前可能出现巨大的延迟。也就是说，一旦初始数据加载完毕，问题可能就已经被缓和了，后面的加载仅包含增量更改，这很有希望能减少延迟的负担。

谨记相邻性的第二个原因就是关于与其他数据环境的集成。业务用户并不关心数据存储的位置或方式，他们只是希望能进行查询(而且可能是跨环境的查询)。

2. 集成并行数据仓库(PDW)

Polybase 项目是 SQL Server PWD 2012 的一个重要功能，跨环境查询是其三个关键设计目标之一。Polybase 为 Hadoop 和数据仓库之间数据的并行输入输出提供了桥梁。但是它也为用户最终提供了统一的异构查询平台。换言之，可以编写以下命令并运行：

```
SELECT     COUNT(*)
,          SUM(s.Value) AS Total_Sales
,          p.Category_name
,          d.CalendarMonth_name
FROM       dbo.hdfs_Sales s
JOIN       dbo.pdw_Product p    ON s.Product_key    = p.Product_key
JOIN       dbo.pdw_Date d       ON s.Date_          = p.Date_key
WHERE      d.CalendarYear_nmbr = 2012
GROUP BY   p.Category_name
,          d.CalendarMonth_name
```

通过上述代码，我能通过单个逻辑声明语句查询 Hadoop 中的数据并将之并入 PDW 的数据中。这个重要性在于它使数据使用者在使用数据时，不必考虑数据来源。它们就是数据表格。

尽管如此，Polybase 也不是完全没有限制。在 PDW 2012 的 RTM 版本中，Polybase 仅仅能支持分隔文件格式和数量有限的分发版：Windows 版 HDP、Linux 版 HDP 和 Cloudera。此外，RTM 版也不能利用 Hadoop 集群的任何计算资源。PDW 只是简单地将 Hadoop 的所有数据高速导入并存放在 PWD 临时表格中。这种模式将随着时间的推移而改变，我们希望将来 MapReduce 作业作为查询优化能够自动生成。希望能看到经过如下轻微修改的查询语句能够触发 MapReduce 作业，使查询中应用的 WHERE 子句成为查询计划中 Hadoop 子树的一部分。

```
SELECT       COUNT(*)
,            SUM(s.Value) AS Total_Sales
,            p.Category_name
,            d.CalendarMonth_name
FROM         dbo.hdfs_Sales s
JOIN         dbo.pdw_Product p   ON s.Product_key   = p.Product_key
JOIN         dbo.pdw_Date d      ON s.Date_key      = p.Date_key
WHERE        s.Date_key >= 20120101
AND          s.Date_key < 20130101
GROUP BY     p.Category_name
,            d.CalendarMonth_name
```

PDW Polybase 集成到 Hadoop 的前景是激动人心的。我们将在第 10 章中对 PDW 和 Polybase 进行更深入详细的探讨。

3. HDInsight

在 HDInsight 作为 GA 版发布时，研发团队同时发布了一款 HDInsight 模拟器，设计用

于本地部署环境的开发者场景。由于针对开发，它只能部署在单一节点配置中。但是，你也可以使用 MSDN 预订版的 Azure 或 Azure 试用版，选择 HDInsight 集群的定制单数据节点配置。

通过链接可以找到更多关于模拟器的初学者使用指南：http://www.windowsazure.com/en-us/manage/services/hdinsight/get-started-with-windows-azure-hdinsight-emulator/。

假如你不喜欢这些选项，Hortonworks 提供了沙盒虚拟机或者你可在 Windows 中搭建单节点开发环境。HDInsight 的优势在于它使用 HDP，你可以在其中很自信地开发可移植代码。不管怎样，你肯定不会缺乏选择。

4. Hadoop 在云端

最近发生在 IT 业界最大的变革无疑是云技术的出现。通过云计算，我们可以利用更动态、更弹性的服务来满足扩展和收缩的需要。很明显，云技术的推进迫使提供商抱着开放的态度接受差异性。例如，2013 年 9 月 Microsoft 宣布了与 Oracle 的新型合作关系，允许 Oracle 软件运行在 Windows Azure 上。谁能想到 Oracle 会提供允许 Microsoft 运行的服务？同样的情况也发生在 Linux 虚拟机身上。

对于 Hadoop 用户而言这都是极好的消息，这意味着客户的选择范围从未像现在这样广泛过。你想使用 Windows 或 Linux 进行本地部署吗？当然可以。你愿意使用 Windows 或 Linux 的云服务吗？当然也可以。你更倾向使用托管服务吗？放手去做吧。所有这些选择不仅仅都是可行的，而且都得到了积极支持。Microsoft 赋予其新的含义，称为数据平台。

本节内容针对 Windows Azure 部署的两个主要选型进行深入探讨。一端是基础架构即服务(Infrastructure as a Service，IaaS)，构建专属的基础架构；另一端是平台即服务(Platform as a Service，PaaS)，提供了托管服务。那种更合适？取决你的需求！

5. 平台即服务(PaaS)

Windows Azure 的 HDInsight 提供了对 Hadoop PaaS 配置的支持。HDInsight 是一个基于 Hortonworks 分发版 HDP 的 Microsoft 品牌增值组件，它是一个安全托管服务，你可以很轻松地进行任务启动(spin-up)和回收(teardown)。spin-up 以分钟计而 teardown 以秒计。就这个角度而言，HDInsight 上手特别容易，它也因为在易用性方面广受好评。

可使用管理门户或 PowerShell 脚本这两种向导中的任意一个来创建 HDInsight Hadoop 集群，卸载 HDInsight Hadoop 集群同样简单。仅需登录 Azure 管理门户，找到集群后单击删除键。删除不必要的集群很重要，因为 Azure 会对 HDInsight 中启动和正在运行的集群持续收取费用。

被删除的数据去哪儿了？

你可能会好奇 HDInsight 集群中被删除的数据去哪儿了，这是一个很好的问题。表面上看这些数据已经丢失，但事实并非如此。HDInsight 最有趣的一个地方就是其实它从计算节点中分离出了数据存储。HDInsight 利用 Windows Azure 存储器(Windows Azure Storage

Blob，WASB)账户来实际保存数据。HDFS 被当作一个数据高速缓存层，默认情况下不会有数据实际存在。当然你可以进行覆盖并且好像“一般”集群那样来看待 HDInsight，但那样一来，如果你想长久保存数据就需要集群保持运行，或者接受在删除集群时失去所有数据的局面。

HDInsight 是一个默认安全的配置，禁用了远程桌面对集群的访问，免费提供了一个拥有授权/验证机制和公开的 433 端口端点的安全网关虚拟机。同样，也可通过 433 端口上的安全网关使用 REST API 检索 Ambari 指标。

HDInsight 使用 HDP 作为其基础分发版，但有所限制。比如现在 HBase 并不被包含在其中。但从好的方向来看，这意味着彻底的普及性。假如你愿意，可将数据集从 HDInsight 导出，然后将其安装在本地或 HDP 的 IaaS 中，它都能运行。

在 2013 年晚些时候，Microsoft 发布了 HDInsight 通用版，现在并不是所有 Azure 数据中心都已安装，但是我预计这种情况会改变，因为每个数据中心都为其预留了空间。

6. 基础架构即服务(IaaS)

现在 HDInsight 和 PaaS 能实现的功能通过结合使用 IaaS 和脚本也能实现。可以使环境创建流程自动化，也可以弹性调整每次启动 Hadoop 集群的资源。当然，你不得不通过脚本来搭建环境，但在安装过程中将得到最大的部署灵活性。你可能会发现会花费更多时间来配置和搭建 IaaS 环境，而不是直接使用 HDInsight 的 PaaS 选项。

但注意，你需要负责一些额外的工作。比如，HDInsight 的安全节点和元存储库并不是自动配备；同样，假如想节约费用和保留数据，必须有分离磁盘、删除虚拟机和执行额外清理的专属自动化流程；创建集群也是如此。

有趣的是 IaaS 提供了在 Windows Azure 中运行 Linux 和 Windows 虚拟机的机令，因此可以在 Windows Azure 上运行你喜欢的任何 Hadoop 版本，包括最新、最好的 Hortonworks 版本甚至是 Cloudera 的 Impala。在 Microsoft 数据平台上基本上不需要将 Windows 局限地认为仅仅是一个操作系统。假如你在几年前对我说这些，我会觉得你很可笑。但世界在进步，相对于通过技术手段减轻工作量，云平台对大规模计算能力资源的管理更感兴趣。

2.3.3　部署计分卡

我们学到了什么？参见图 2-1。

我们已经学了一些相关知识。例如，安全性对任何选项而言都不是最好的，因为这不是 Hadoop 的优势。同样，相邻性取决于数据的实际产生源。当数据来自云端时，云产品提供的相邻性是更优选择(在使用同样服务提供商的前提下)。但是，我对相邻性的评分基于在整个企业中与其他不同数据源的更一般集成能力。在撰写本书时，大部分这种类型的企业数据存在于本地部署中，这也解释了图 2-1 的计分原则。

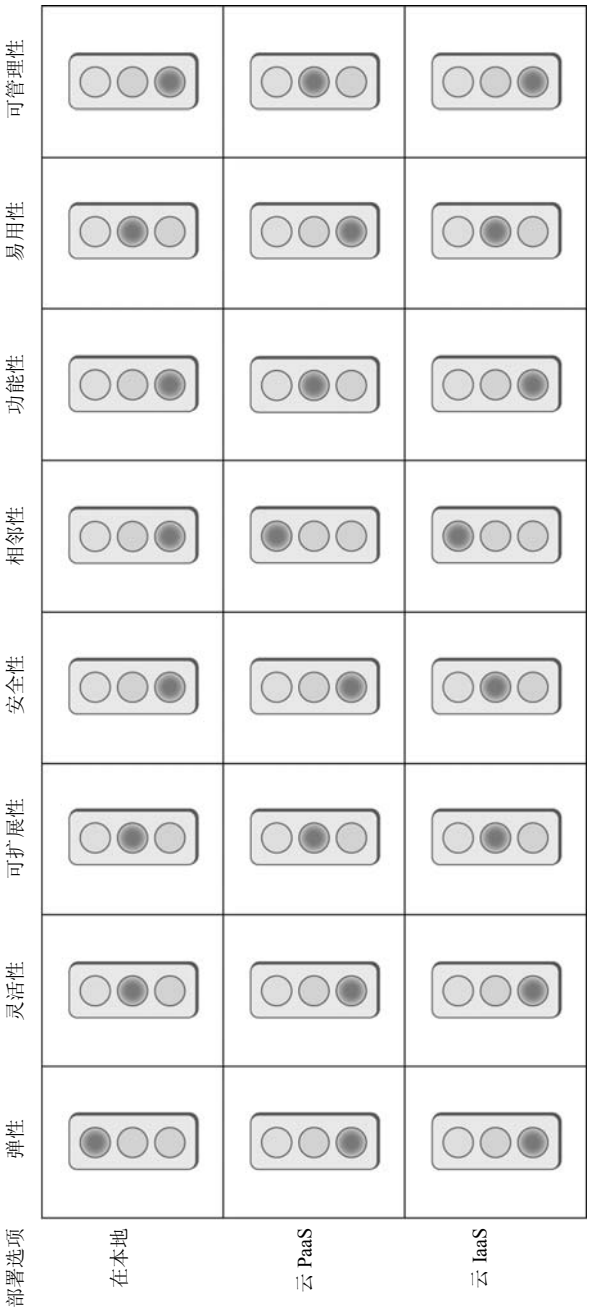

图 2-1　部署计分卡

云部署具有以下优点：

- 弹性扩展
- 检查疏漏(kicking the tires)
- 原型化(prototyping)

在图 2-1 所示的计分卡中，最亮眼的就是云计算为鼓励实验提供难以置信的灵活性和

弹性延伸性，对“突发性”工作量的出色处理。但是，除非数据一直产生于云端，否则相邻性多半会是一个问题。谨记相邻性是双向衡量：

- 与数据管理系统的距离(即带宽问题)
- 与其他企业数据源的距离(即集成分析困境)

假如只是承担初始或历史负载，那么第一个因素的重要性可能会减轻，假如后续增量在可控范围之内，那么这种担忧会被显著减少。第二个因素是一个比例问题，假如本地部署数据源仅有少量或者没有集成分析，那么相邻性问题不需要担心；假如有大量集成分析，相邻性可能就是整个云部署的关键要素。

本地部署具有以下优点：

- 自定义部署
- 数据产生于本地
- 安全数据

本地部署最适合那些数据产生于本地，并且可能需要和当地其他数据源进行集成的环境。拥有专属环境，虽然灵活但要保证计算能力和人力资源。因此它适合能更容易确定可预见工作负载大小的情况。

2.4　本章小结

从本章中你学到了什么？世界已经不像当初看起来那样了。从 SQL Server 2008 演变到 SQL Server 2012 的过程中世界已经发生了翻天覆地的变化。Microsoft 数据平台已经包容和接受运行 Apache Hadoop 开源技术的 Linux 部署。在部署选择方面我们拥有更多样化的可用选项。我们的选择在很大程度上基于业务需求和数据价值(正应该这样)。

现在你应该明白影响部署选择的因素以及如何提前进行评估。同样应该对 Hadoop 生态系统和影响其发展方向的某些驱动因素有更好的理解。