

1 本节我们讨论内存数据库 HANA

2 数据处理系统提供大数据计算处理能力和应用开发平台。从计算架构的角度，将数据处理系统分为数据算法层、计算模型层、计算平台层、计算引擎层等。

与大数据相关的计算算法包括机器学习算法和数据挖掘算法。计算模型是指不同类型的大数据在不同场景下的处理方式，包括批处理、流计算、结构化数据

的大规模并发处理(MPP)模型、内存计算模型和数据流图模型。

就计算平台和引擎而言，通常具有代表性的大数据处理平台有 Hadoop、Spark、storm、Pregel 等，

本节我们讨论下内存计算模型中的内存数据库 HANA

3 HANA- 内存数据库的主要特点

在硬件和软件架构方面，该行业已经取得了多方面的进步，这使得基于内存的计算在今天比过去更加重要，如图所示。

简而言之，支持 64 位 CPU 的新型硬件现在可以在单个设备上支持 2TB。此外，面向分布式体系结构和云的软件体系结构和解决方案的进步使得利用这些新的硬件功能更加容易。

内存计算，在许多方面，内存计算是内存数据库的近亲。与许多数据库一样，它的设计目的是支持传统数据库通常期望的所有数据管理方面，如查询和事务，区别在于数据是在 RAM 设备上管理，而不是在磁盘上管理，因此根据各种基准测试，可能具有比传统数据库好 1000 倍的性能和延迟。

传统的内存数据库和基于内存的计算之间的主要区别是: 1. 为分布式和弹性环境而设计 2. 专为内存数据处理而设计

执行数据所在的代码:

我们可以将数据存储在与应用程序相同的地址空间中，这是最大的好处。

与磁盘甚至闪存设备不同，我们可以通过引用访问数据，从而执行复杂的数据操作，而不需要任何序列化/反序列化开销。有了新的动态语言类(如 Java、JavaScript、JRuby 和 Scala)，通过网络传递复杂逻辑并在远程设备上执行它也变得非常容易。

内存计算在很大程度上依赖于该功能，并通过实时映射/还原和基于流的处理作为其体系结构的核心元素，提供了一种新的复杂数据处理功能，非常适合数据的分布式特性。

HANA 结合了硬件和软件的优势，使内存数据库成为可能。

硬件技术的发展，多核架构，跨刀片并行扩展，64 位地址空间，当前服务器板 2Tb, 25Gb /s 的数据吞吐量。

所有这些都使得性价比迅速下降，这意味着先进的高速处理硬件变得更便宜和负担得起。

Hana 数据库还包含了一些先进的软件技术和特殊的方法，包括结合基于行和基于列的存储、压缩、分区、不聚合表、只插入和动态扩展。

1)基于行的存储用于支持事务处理，OLTP 和基于列的存储用于支持分析处理，如 OLAP。

2) .因为内存的大小不能像硬盘那么大，我们必须尝试任何方法来减小 DB 的大小，HANA 使用了很多压缩的方法。

3) .提高 DB 性能，可以对数据库进行分区，实现并行化，提高性能。

4) 消除聚合表，通常是为了数据分析，建立数据仓库从多个数据源收集相关的表数据，并进行一些聚合，以方便进一步的分析。

但在 HANA 中，它不需要聚合表，内存中的聚合可以快速完成，不需要提前做，动态聚合性能不会受到影响，并且聚合表产生冗余数据，消耗内存空间。

5) 只插入的方法有什么好处？

嗯，因为主存储是压缩的(这有利于读取性能)，更改将是代价比较大的。想象一下，您想要在压缩文本文件中进行更改。

您需要解压缩，进行更改，然后再次压缩。由于这个原因，数据只追加到未压缩的增量存储区，在那里

更改操作要消耗少得多。

合并将数据从增量移动到主增量，因此只会时不时地压缩，而不是每次更改都压缩。

设置删除标志是基于 Row IDs 的，适用于压缩和未压缩的数据。

4 HANA 只执行插入数据

在列存储中 HANA 将只做数据插入。也就是说，当您更新或删除数据时，HANA 将会只是插入新数据。出于性能原因，列存储分为主存储部分和增量存储部分。

新的数据正在产生到 delta 存储，并将频繁合并到主存储。

更改语句的工作原理如下：

插入：INSERT 语句将在增量中插入一条新记录。合并过程将从增量 delta 存储中的数据合并到主存储。

删除：DELETE 语句将选择该记录，并通过设置标志(用于 main 或 delta)将其标记为无效。一旦它不再有打开的活动事务合并

进程将从内存中删除该记录。

更新：UPDATE 语句将插入记录的新版本。合并过程将最新数据从增量 delta 存储中合并到主存储。

一旦没有为它们打开的活动事务，旧的版本将被删除。

因此更新 在内部由插入+删除组成。

下图简要示意说明了这一原理。请注意这张照片不是真实的

列存储。可以看到主存储中一个已经删除的记录 (id 7)和两个旧版本的记录(用灰色标记) (id 8 和 14)，其中名称已被更新，这将导致插入记录的新版本。

5 我们来对比下硬盘和内存

HANA 是内存中的数据库。但这到底是什么意思呢？不是所有的数据库都使用内存吗？是都是用内存的，但是有很大的不同。

其他数据库将数据存储存储在硬盘驱动器上的物理页中，并将一些数据的“副本”移动到内存中；这就是所谓的缓存。

当数据从内存缓存中丢失时，必须从磁盘中获取数据，这将花费非常长的时间，最小延迟为 10,000 微秒，从而降低了查询和计算性能。

不仅如此，由于内存中的数据是磁盘的副本，因此以反映磁盘结构的方式存储数据，例如链接的页、扇区等，这在获取数据时增加了额外的开销。

然而，HANA 直接将数据存储存储在内存中，它不是针对数据如何存储在磁盘上进行优化，而是针对高效访问进行优化，

它是高度压缩的，使用计算机科学家能够有效应用的的最有效的内存表示。

即使在缓存丢失的情况下，从 DRAM 中获取丢失的数据也只需要 0.1 微秒。这是相较于存储在硬盘的 10 万倍的速度，这非常重要的。

请注意，当事务更改了一段数据时，HANA 会记录该更改的日志，以便在发生崩溃时可以恢复该更改。

6 摩尔定律：内存的价格变化

有两个行业趋势促成了 HANA 的性能。第一个趋势是内存价格在过去几年里惊人的降低了。

几年前还难以想象的昂贵的价格，现在已经变得完全可以负担得起了，正如你从这张图中看到的，这张图显示了 1 GB DRAM 的价格随时间的变化的趋势。

注意从 100 美元/GB 降到 10 美元/GB 只花了很短的时间。

如今，1TB 内存的成本远低于 1 万美元，计算机制造商现在正在组装具有 4Tb 内存的服务器，不久将在单个系统中达到 12Tb 内存。

7 越来越多的核的处理能力：

第二个趋势是计算能力的提升。十多年前，我们会买一台只有一个处理器的电脑。

几年后，英特尔推出了双核处理器，不久之前又推出了拥有 8 个双线程核的 64 位处理器。

HANA 的第一批服务器将拥有多达 4 个这样的处理器。现在，我们看到的是 12 核的 Intel Sandy Bridge

处理器，以及多达 16 核的服务器。

这不仅仅是 10 年里性能的 200 倍飞跃，这也预示着现代软件面临的挑战——内存计算还意味着软件现在必须利用现代处理器的多核架构和多处理器服务器来编写软件，甚至还继续扩展。

8 重新思考数据管理

以磁盘为中心的计算也是迫使事务工作负载和分析工作负载分离的一个主要因素。

用于报告和分析的专门数据仓库需要事务数据的移动、转换和预处理。

这带来了巨大的复杂性：有时企业可能拥有同一数据的三个不同副本.....

通过利用内存计算的能力，HANA 可以将 OLTP(事务处理)和 OLAP(数据分析)重新整合到一个数据库中。

HANA 扭转了分离的、专门的分析数据库的行业趋势。

9 我们来看下 SAP HANA 架构这张图是 SAP HANA 的简化架构图。

HANA 通过了 IBM、HP、富士通、思科、戴尔等合作伙伴提供的强大企业服务器的认证和交付。

这些高度优化的服务器可扩展到 192 个内核，由合作伙伴(如 SUSE 和 Red Hat)的 Linux 操作系统管理。

HANA 软件针对其所运行的硬件进行了精心优化，并利用了其所运行的现代处理内核的大量并行性和内存架构。包括固态或闪存盘的多种存储系统可用于高可用性和备份支持。

HANA 与其他数据库系统完全兼容，并支持 SQL、JDBC、ODBC 和 MDX-Multi - Dimensional Expressions 等分析数据源接口。

这意味着现有的应用程序无需任何修改就可以使用 HANA。另外，HANA 也支持基于 web 的接口。

10 并行化：HANA 从一开始就被设计成在每个操作级别上都是大规模并行的，这是它表现如此出色的主要原因之一。第一级很简单：为多个并发用户和查询分配各自的计算资源。

然后，HANA 优化器创建一个执行计划，以并行化查询中的操作。

例如，我可以在不同的处理核上同时扫描两列。

通过对数据进行分区，我实际上可以使这些操作更加并行化，甚至可以跨多个服务器。

例如，我可以将表 A 相加为四个独立的并发聚合，最后将四个结果相加。

11 内存计算 - 数据结构

HANA 的另一个非常重要的属性是：数据存储在内存中的方式。

传统上，数据库按行存储数据。一个原因是，当数据库存储在磁盘上时，这样做要方便得多：您只需在文件的末尾写入一个新行。

另一方面，HANA 按列顺序存储表。

这意味着执行聚合(例如销售额总和)的效率要高得多，因为所有相关数据都位于连续的内存位置，大多数值将由处理器预取。

不仅如此，通过将数据存储在这样的列中，我们可以比传统数据库更有效地对数据进行排序和压缩。

12 消除聚合

在传统数据库中，通常不可能对原始的事务数据执行数据分析。

因此，建模专家被迫引入额外的表，其中包含预先计算的总和、平均值等，以提高分析性能，放弃细节信息。

例如，我们可以在上面的表中按国家合计销售额，从而生成另一个更小的表，以便更快地访问以生成报告。

这些额外的表被称为“materialized aggregations”。

这些聚合的问题是它们使应用程序非常复杂。

每个事务还需要后续重新计算和更新所有受影响的聚合表，这可能会大大降低操作速度。

另一种方法是定期更新聚合，这意味着报告通常不是最新的，或者与事务数据不匹配。

但是 HANA 可以以高性能聚合大量数据集，每个核每秒聚合超过 1500 万次，所以通常真的不再需要预先存储聚合数据了。

这会带来:更简单的数据模型，更少的表;不遗漏细节;更简单的应用程序逻辑，更少的写锁和日志;更好的 up-to-datedness 数据的实时性

所有这些都极大地提高了性能。

这就是客户报告了数百倍性能改进的原因之一。

13 OLAP 引擎 用于对基于列存储的星型模式进行聚合查询的专用引擎:

具有多个属性表(维度)的大型事实表

为得到最佳结果模型创建基于星型结构的模型:如果必要，通过将多个事实表合并为一个；组合多个维度(雪花模型到星形模型)

商业智能和数据分析经常使用规范化的数据模型，称为星型模式。

这涉及将所有事实数据保存在一个大表中，并将对其他信息的引用保存在单独的表(称为维)中。

例如，我们可以将所有的销售数据放在中心事实表中，而将国家和产品等细节放在维度表中。

HANA 针对这些类型的模型有一个特别优化的执行引擎，它通常会将对这些表的操作加快 10 倍或更多，并对 HANA 的整体性能影响做出很大贡献。

14 基于列的表的其他优势

扫描列数据非常快，通常不需要很多索引；

1) HANA 默认只索引主键。这节省了内存和索引更新时间'

2) 动态视图可以动态计算;在每秒超过 10 亿条记录的情况下，聚合速度也非常快。

3) 您可以看到，更改表是多么容易，例如添加一个新列，而不需要完全重新组织表;

每个列都是单独管理的。

4) 当然，只查看某些列(一种称为投影的操作)在面向列的数据库中是非常简单的。

5) 压缩减少了对主存的访问。这不仅代表了由于改进的缓存行为而带来的性能奖励，而且说明了内存数据库的另一个优势:相同的内存容量可以容纳更多的数据;

数据压缩比为 5 倍的情况并不少见。

15 本节我们讨论了以 SAP HANA 为例讨论了内存数据库，今天的内容就到这里，谢谢大家