Exadata V2 架构分析

中国 Oracle 用户组

作者：黄凯耀(Kaya)

http://www.acoug.org

版本	发布时间
1.0	2011/3/15

目录

概述：本篇文章是由作者在其博客上发表的一系列关于 Exadata 的文章中摘录 6 篇，汇总而成的，重点解析 Exadata 的架构。文章语言风格比较诙谐，希望这样的风格能让纯技术的文章显得不那么枯燥，同时可以帮助对 Exadata 感兴趣的朋友从中获得想要的知识。

1 Flash Cache 特性分析

准备写一系列的文章，专门分析下 Exadata 的架构。时间真过得很快，自从 Exadata 问世以来，一直围绕着它做各种各样的性能测试，从 V1 过渡到 V2。Exadata 在大踏步地前进着，我总认为，这应该是一个很优秀的产品，而我也相信时间会证明这一点的。

这些文章不会有很严谨的结构，或许某天，某时发现的一个特性，一个特点，一个最佳实践，一个有意思的地方，一个令人惊讶的数据，一个令人鼓舞的图形，都会成为这一系列文章的一部分。

Flash Cache 是 V2 里引入的一个特性，大家或许都认为 Flash Cache 主要是用于 OLTP，为 OLTP 提高更高的 IOPS，这点当然没错，Flash Cache 相比于一般磁盘，会带来上十倍的 IOPS 的性能提升。但即使对于 DSS 场合，Flash Cache 也是提升性能的一个利器。

下面是看图说话时间：

上图中，MBPS 达到了 4.5 GB/s。但是存贮节点上的 CPU 和数据库节点上的 CPU 的利用率却在 20%左右。如果在存贮节点加入 Flash Cache，整个系统会发生什么样的改变呢？下图就是加入了 Flash Cache 后的 CPU/IO 情况。MBPS 达到了 14G/s，存贮节点的 CPU 利用率超过了 60%，DB 节点上的 CPU 利用率也超过了 40%。

仔细地对比研究这个图，对系统间资源的协作会有进一步的认识。

可以说，Flash Cache 除了提供更高的带宽外，还大大解放了存贮节点上的 CPU 处理能力。这种处理能力可是单纯的高端存贮都不具备的，这就是 Exadata 上独有的 Smart Scan 特性。**Exadata V1 实现了对磁盘的 Smart Scan，而 V2 则实现了对 Flash Cache 的更快，更强大的 Smart Scan**。当然，在软件设计上，还有另一个重头戏，它更是大大的利用起了存贮节点上的 CPU 处理能力，同时还能减少对带宽的争用。

2 Flash Cache 对 OLTP 的性能影响

既然提到了 Flash Cache，如果不提下对 OLTP 的提速好象会缺少点什么。对 OLTP 系统而言，缓存是一个极其重要的设计，不管是数据库节点的内存上的 Buffer Cache，还是存贮节点上的 Flash Cache（Exadata)，还有数据库节点上的 Flash Cache(某些平台，如 Linux)。

前几天在 Eygle 的性能调优课程上听过他提过的一句话，缓存为王（致敬，呵呵），深以为然！这应该是

他极其重要的实践总结，对于 OLTP 系统而言，这真是一点不为过。

下面的表格列出了一个实际场景，分别测试了不同的 Buffer 命中率和引入 Flash Cache 之后的性能变化。

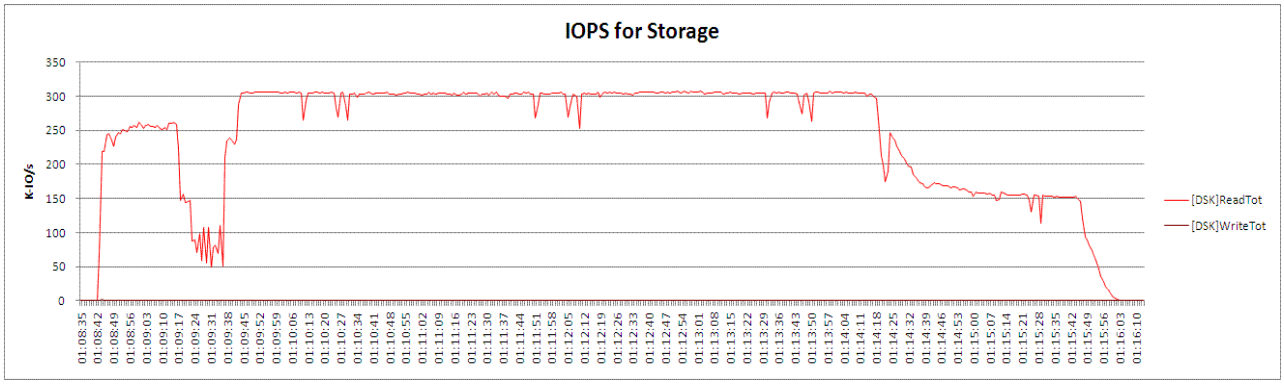
Cache	Executions	Buffer Hit %	CPU Time (s)	User IO Time (s)	Elapsed Time	Total Elapsed Time (s)	CPU %	Response Time	SpeedUp Factor
Buffer Cache	30000	85	70	4082	00:02:20	4138	3	0.138	1
Flash Cache	30000	85	69	265	00:00:12	323	36	0.011	13
Keep Buffer Cache	30000	100	19	0	00:00:02	26	61	0.001	158

上面的三行对应的 OLTP 负载是相同的，但总的执行时间却是大大不同的，从 2 分多钟到 10 几秒到 2 秒。唯一的区别就在于 Cell Flash Cache，Buffer Cache 的介入。

上面这些数据起码可以得出几个结论：

1. Buffer Cache 会减少 CPU Time，但 Cell Flash Cache 不会。这从另一个方面说明，IO 调度是需要 CPU 的。
2. Cell Flash Cache 和 Buffer Cache 都大大减少了 User IO Time。最终的结果就是大大提升了响应时间，例如上面我们得到了从 13 倍到 158 倍的性能提升。
3. Cell Flash Cache 和 Buffer Cache 都提高了 DB 节点的 CPU 利用率。例如上面我们的 CPU 利用率从 3% 提高到 36%和 61%。

Exadata V2 宣传的一个令人惊诧的数字之一是每秒钟一百万个 8K 的随机 IO。很恐怖吧，1,000,000 IO/s。其实，如果只是进行读操作而没有写操作的话，这个数字会更令人恐怖。下面是一个截图。把里面的峰值(300k)除以 3 乘以 14，就得到整个机器 14 个 cell 可以达到的一个 IOPS： 1.4 million IO/s。



3 Flash Cache 究竟有何帮助

关于 Cell Flash Cache，好象大家都余兴未尽，例如：

1. 一个真正的生产系统，真的需要 1,000,000 IOPS 吗？
2. Cell Flash Cache 对用户带来的真正的好处在哪里？

假设用户的逻辑 IOPS 达到 1,000,000 IO/s。Buffer Cache 命中率为 98%。则落在 Cell Flash Cache 中的 IOPS 为 20,000 次。这个数据对于 1/4 配，即一个 quarter 的 Exadata 来说，当然是小菜一碟，不过对于一般磁盘呢？一般的磁盘每秒大约为 300 IOPS，这意味着需要有 20,000/300=66 个磁盘，这已经是一个不低的配置了。

再进一步考虑，如果系统想支持混合负载，即系统同时支持数据仓库查询和 OLTP 在线系统。那么这时 OLTP 的性能将会受到严重影响。要知道，数据仓库查询要求的 MBPS，而 OLTP 要求的则是 IOPS，这两个指标是会相互影响的，用户一般考虑的是 OLTP 优先。甚至于出现一种情形，我不清楚下面这种情况会多普遍：

在 IO 受限的情形下，不敢对 OLAP 查询启用并行，由于不启用并行，OLAP 查询很久不返回结果，进一步地，用户会在表上建更多的索引来“优化”这些 OLAP 查询。最终，整个混合型的系统就变成了一个类似 OLTP 的系统了。

在这种情形下，Cell Flash Cache 或许会带有性能上的实质提升，通过 Flash Cache 会大大提升 OLTP 的查询性能，同时，后端的磁盘和 Flash Cache 会一起提供足够的带宽给 OLAP 查询(很聪明吧)，这在硬件上保证了两者并存的可能性。另一方面，由于在 IO 上的财大气粗，对原有 OLAP 的过度优化终于可以停止了，系统设计会回归简单，与此对应地，系统维护成本也会大大降低。

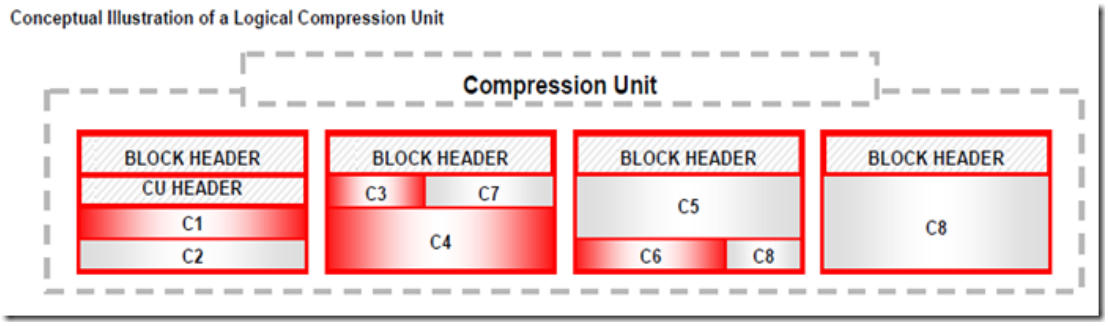
或许这是 Cell Flash Cache 可能给大家带来的一个很诱人的地方。

4 Hybrid Columnar Compression 特性介绍

下一个要出场的是 HCC，Hybrid Columnar Compression。目前它是 Exadata 上面才有的一个特性，所以又叫做 Exadata Hybrid Columnar Compression。

在 Exadata V2 架构分析 (1)中，曾提到“在软件设计上，还有另一个重头戏，它更是大大的利用起了存贮节点上的 CPU 处理能力，同时还能减少对带宽的争用”。Exadata 的很多设计，或许从根本上讲，就在于充分利用起存贮节点上的处理能力，Smart Scan 和这里所要提及的 HCC，就是两个典型的代表了。HCC 中文翻译过来或许就叫做混合列压缩，它是在单纯的行存贮和列存贮之间取得的一个折衷。

这篇白皮书是一个比较好的参考文档：Exadata Hybrid Columnar Compression。



上图中，典型地，4 个 8KB 的 Blocks 被当成一个 Compression Unit。在这个 CU 所能存贮的 Rows 中，每个 Column 被分开存贮。可以想像到，每个 Column 里的内容是很相似的，如果与 Row 之间的内容做比较的话。于是，对每个 Column 的内容进行压缩，会得到很好的压缩率。根据压缩算法的不同，Oracle 提供了四种不同的压缩等级，详见上面提到的白皮书，这里就不详细列出了。

到底 EHCC 的压缩率可以达到多少呢？白皮书中提到两个数据，可以做为参考，对于 Warehouse Compression，有 10x 的压缩率，对于 Archive Compression，有 15x 的压缩率。

EHCC 相对于单纯的 Column Compression 而言，有一个极其突出的优点，这点是不得不提及的。当进行行级访问数据时，如根据 Rowid 返回一行数据，EHCC 只要一个 IO 就够了，不管所访问的表有多少列，而对

于单纯的 Column Compression 而言，对于每个 Column，都必须有一个 IO 操作。那么，随着表设计的复杂，如一个表拥有成百上千列，两种存贮方式的性能就能体现出成百上千倍的差距了。

5 Storage Index 的实现

Exadata 上另一个聪明的软件设计是实现了 Storage Index。（另一篇白皮书中有对 Storage Index 的一个描述 A Technical Overview of the Sun Oracle Exadata Storage Server and Database Machine）

Storage Indexing

Storage Indexes are a very powerful capability provided in Exadata storage that helps avoid I/O operations. The Exadata Storage Server Software creates and maintains a Storage Index in Exadata memory. The Storage Index keeps track of minimum and maximum values of columns for tables stored on that cell. When a query specifies a WHERE clause, but before any I/O is done, the Exadata software examines the Storage Index to determine if rows with the specified column value exists in the cell by comparing the column value to the minimum and maximum values maintained in the Storage Index. If the column value is outside the minimum and maximum range, scan I/O for that query is avoided. Many SQL Operations will run dramatically faster because large numbers of I/O operations are automatically replaced by a few in-memory lookups. To minimize operational overhead, Storage Indexes are created and maintained transparently and automatically by the Exadata Storage Server Software.

如果 Exadata 给你的印象就是有很强大的硬件，却不会利用传统的性能优化方法，比如索引，去加快查询速度的话，那么 Storage Index 的出现或许会改变你的这种观念。而且，Storage Index 是完全自动化的，它甚至不需要人工的干涉就能工作得很好。

Smart Scan 之所以冠名以 Smart，是因为它在扫描数据的时候同时做过滤操作，只把必需的数据返回给数据服务器端；而 Storage Index 在这里却直接避免了对 Disks 的访问，其 Smart 的程度与 Smart Scan 相比，或许已经到了超凡脱俗的境界了。

如果真要找一个类比的话，DB2 中有个特性或许可以与 Storage Index 做下对比，那就是 Multi Dimensional Clustering。仔细阅读下 MDC 的介绍，两者真的有异曲同工之妙。

6 总结

做下回顾与思考。

从第一篇开始到现在，Cell Flash Cache， Exadata Hybrid Columnar Compression， Storage Index 轮番上场，加上 V1 版本里出现的 Smart Scan，Infiniband 等等，多少会给人以眼花缭乱的感觉。

最根本的一点，当然在于 Exadata 本身是一个 Balanced System。

不过，这些技术做为一个整体对实际应用会带来多大的好处呢？这不是一个很好回答的问题，当然也可以用一句话回答——具体问题具体分析。

思路应该是这样的，首先明了当前系统的现状

- 如果当前系统运作得很好，简直完美的设计，分区，并行，并发控制等都无可挑剔，业务量也没超过系统极限，那么也许只有一些技术会对性能提升比较明显，如 Smart Scan。举个例子，如果原来系统

没存在 IO 上的瓶颈，Cell Flash Cache 就英雄无用武之地了。

- 如果当前系统运作得很好，简直完美的设计，分区，并行，并发控制等都无可挑剔，业务量大大超过系统极限，系统出现 CPU 或者 IO 或者 Network 上的瓶颈了，这时或许就是考虑升级的时候了。
- 如果当前系统为小数据量所设计，如刚开始没有分区，但随着业务量的增长，系统出现 CPU 或者 IO 或者 Network 上的瓶颈了，这时可以有两条途径，改进原来的设计，或者考虑硬件升级了。

设计上的改进包括

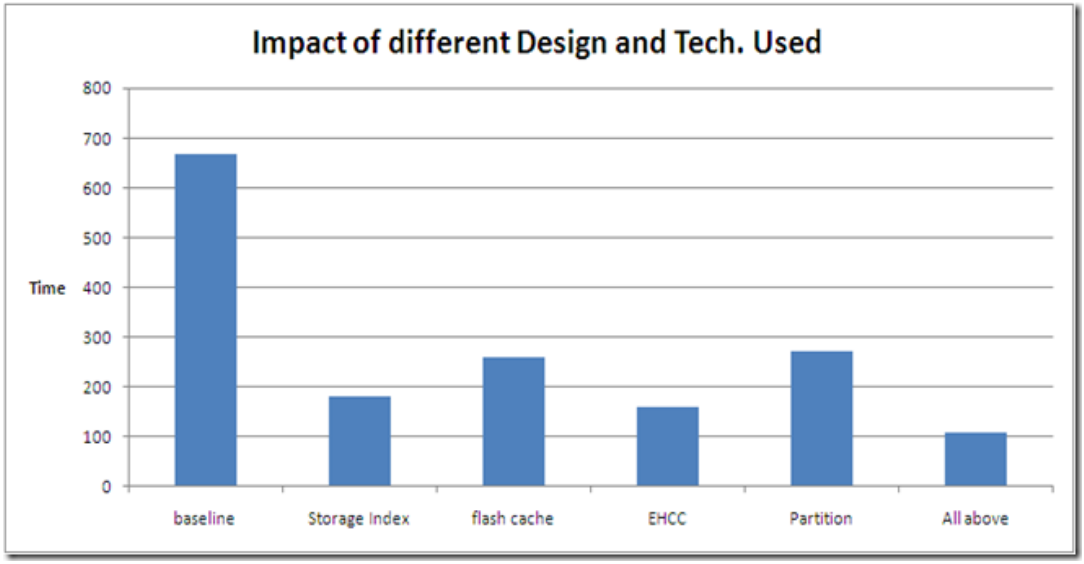
- 分区
- 压缩
- etc…

硬件上的升级包括

- Smart Scan
- Flash Cache
- Storage Index
- etc…

最后，做为一个例子，可以考核一个原来的系统（没分区，没压缩，没 Flash Cache，没 Storage Index），分区，压缩，Flash Cache，Storage Index，分别能带来的性能上的提升，及做为一个整体带来的性能提升。

感性认识还是更重要的，虽然我把它放在最后面了。



Kaya 的个人简介

黄凯耀，英文名 Kaya

目前工作于 Oracle RealWorld Database Performance Group，一个隶属于 Oracle 公司总部数据库产品管理的核心团队。

大学及研究生时期专注于 Linux 应用开发和 Linux 内核开发工作，

2006 年加入 Oracle 公司至今，主要专注于
现实世界的数据库高性能，高可扩展性，高并发高压力的项目实践
Oracle Exadata Database Machine 上的客户真实项目的性能测试与优化
Oracle 客户所碰到的典型的性能问题的解决与方案建议
DSS 技术，如并行运算，数据分区，数据压缩，资源管理，ETL 等
OLTP 技术，如连接池管理，Cursor/Session 管理，Index 设计，应用架构优化等
SQL 性能优化，如 SQL 重写，执行计划分析，CBO 优化等
操作系统与数据库的资源监控与性能分析

个人站点：http://www.os2ora.com
Email： kaiyao.huang@gmail.com