­

**基于NVM与RDMA的高速缓存在分布式数据库系统中**

**的研究与实现**

|  |  |
| --- | --- |
| **申请者姓名** |  |
| **所属机构** | 华东师范大学 |
| **提交日期** |  |

**目录**

**1. 项目介绍**

1.1. 题目

1.2. 项目背景及研究意义

1.3. 研究目标

1.4. 研究方法

**2. 研究计划**

2.1. 项目期限

2.2. 进度安排

**3. 预期成果**

**4. 人力、设备等投入及项目预算**

4.1. 人力投入

4.2. 设备投入

4.3. 项目预算

**1. 项目介绍**

1.1. 项目名称

基于NVM与RDMA的高速缓存在分布式数据库系统中的研究与实现

1.2 项目背景及研究意义

1.2.1 研究背景

互联网行业的飞速发展，使得数据规模、用户请求日趋庞大。这一发展对后台数据库系统也提出了更高的需求。工业界开始倾向于使用分布式数据库系统，以应对数据量和负载的快速增长。比如，一些开源系统如CockroachDB，TiDB， 阿里的多款数据库产品如X-DB，PolarDB, Google用于支持其广告业务的分布事数据库Spanner等等。除了在数据库本身架构与原理上以得到性能的突破，另一种提升性能的“捷径”是在分布式数据库系统中使用已经出现并正在逐渐量化生产的硬件。对数据库而言，数据存储、访问以及网络传输尤其重要，而新出现的两款硬件(或与之相关技术)非易失性内存(NVM)与远程内存访问技术(RDMA)在数据访问、网络传输上能产生较大的收益。

从硬件特性角度看，NVM与内存、磁盘、SSD相比有其独特的优势。在相同价格下，NVM较内存存储容量有数倍的提升，且数据存储具有非易失性。相对于SSD而言，NVM在读写速度上具有数量级的优势，并且可擦除次数更高。此外，SSD、磁盘存储为传统的块状设备，在访问方式上以块/页为单位，而intel即将面世的NVM设备3D-Xpoint允许字节级操作。 因此，在存储层次结构上，NVM可以看作是具备非易失性的、可支持按字节操作的、位于内存与SSD之间的全新存储层。

另一方面，基于Infiniband和RoCE的RDMA技术相对于传统万兆网络在传输速度上有了数量级的提升，在一个集群内部，可以一个数据包的传输时延从毫秒级别提升到微秒级别。

学术界和工业界均非常看好这两种新硬件在数据库的应用前景。但如何充分利用NVM与RDMA以提高数据库系统性能目前仍是一个值得探讨的命题。 工业界目前采用的做法是简单使用它们替代当下数据库的组成部分，如使用NVM替换磁盘、使用RDMA作为网络通信方式。这种简单暴力的做法可以获得一定收益，但未考虑如何做到最优性价比。学术界更偏重于研究利用NVM做一些精细化的优化工作，比如存储模型、查询处理和日志管理等，侧重于提高系统的吞吐性能(如TPC-C的吞吐率)。 由于学术界尚未获得真正的硬件，相关的实验多依赖于系统原型且很多结论基于模拟的结果，能否真正在数据库系统中发挥作用还不得而知。

1.2.2 研究目的

本项目探讨如何利用NVM与RDMA实现一个高速的共享缓存，并将其部署在分布式数据库系统中以提高性能。相比于内存，NVM因为价格优势可以构建容量更大的缓存。这样，缓存数据内容可以做到更加灵活，缓存命中率得以提高。而RDMA的访问方式使得数据无需经过被访问节点的CPU，而是可以直接访问存储设备本身。这一性质使得NVM设备之间可以快速联通，实现高效的共享缓存。共享缓存内部设备间的访问是高速的，客户端对共享缓存的访问也是高速的。 为方便起见，我们将这一缓存称之为Shared Cache (SC)。SC主要发挥三种作用 ：

1. 作为缓存，SC既可以支持行数据缓存也可以支持块数据缓存，从而减少数据库对磁盘的访问，包括：（a) 利用NVM byte-addressable的特性实现行级缓存，在单点访问请求中，避免对磁盘数据的直接访问，提升单点查询以及事物处理性能；（b) 实现块状数据缓存，在范围查询中，避免对磁盘数据的大量扫描，大幅提升查询效率。通常由于缓存容量的限制，传统数据库对范围数据的缓存效果往往并不理想，而SC由于容量更加庞大，对范围查询的缓存支持将更加有效。
2. 作为共享缓存，SC不仅可以缓存本地数据，也可以缓存异地数据，这有助于实现整个系统的负载均衡，实质上也进一步提升了缓存的容量（本地放不下的热数据可以放到异地）。
3. SC提供写缓存的功能。在事物处理过程中，将部分数据物化到SC中，由于NVM的非易失性，这些数据是持久化的。这样可以减少日志管理的代价，并有效提升数据恢复的性能。

1.2.3 待研究的问题（研究内容）

1. SC本身如何做到高性能？  
   对外完全透明，数据存储在NVM设备中，设备通过RDMA互连。SC对外能够提供较大的数据存储，极高的吞吐率，较低的数据访问延迟。具体包括SC中统一地址的管理，设备间基于RDMA通信协议的设计，SC中数据存储空间分配，高效索引构建，数据写入方式等一系列问题。
2. SC在数据库中如何发挥读缓存作用？  
   研究缓存的数据形态和粒度。行、列、块、行组似乎都有可行性，需要对比它们各自的优缺点。研究缓存的替换策略，·包括换出块的选择，以及本地缓存和异地缓存的选择。
3. SC在数据库中如何发挥写缓存作用？  
   研究如何减少日志和提升恢复效率。基于写缓存设计新的日志和恢复机制。研究如何处理好写缓存与读缓存之间的关系。

1.2.4 相关研究工作

1. NVM应用于日志管理

[1] Joy Arulraj, Matthew Perron, Andrew Pavlo: Write-Behind Logging. PVLDB 10(4): 337-348 (2016)

[2] Ru Fang, Hui-I Hsiao, Bin He, C Mohan, and Yun Wang. High performance database logging using storage class memory. In ICDE, pages 1221–1231. IEEE, 2011.

[3] Jian Huang, Karsten Schwan, and Moinuddin K Qureshi. Nvram-aware logging in transaction systems. VLDB, 8(4):389–400, 2014.

1. NVM及RDMA提供存储系统

[4] Alexander van Renen, Viktor Leis, Alfons Kemper, Thomas Neumann, Takushi Hashida, Kazuichi Oe, Yoshiyasu Doi, Lilian Harada, Mitsuru Sato: Managing Non-Volatile Memory in Database Systems. SIGMOD Conference 2018: 1541-1555

[5] Youyou Lu, Jiwu Shu, Youmin Chen, Tao Li: Octopus: an RDMA-enabled Distributed Persistent Memory File System. USENIX Annual Technical Conference 2017: 773-785

[6] FaRM: Fast Remote Memory.Aleksandar Dragojević, Dushyanth Narayanan, Orion Hodson, and Miguel Castro, Microsoft Research.

1. 与查询优化相关

[7] Feng Li, Sudipto Das, Manoj Syamala, Vivek R. Narasayya:

Accelerating Relational Databases by Leveraging Remote Memory and RDMA. SIGMOD Conference 2016: 355-370

1.3 研究目标

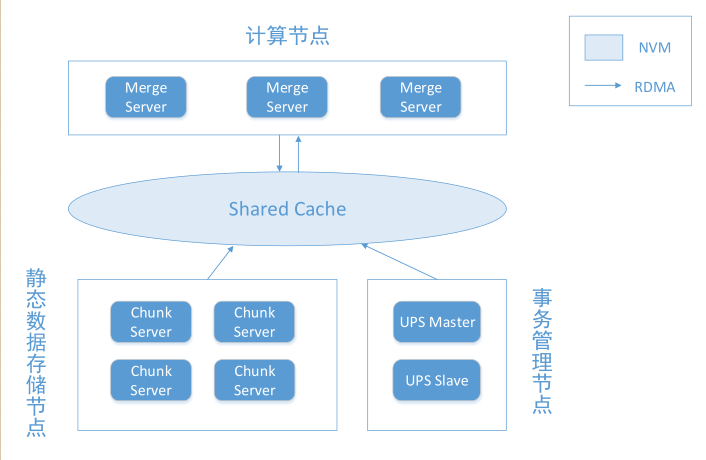
1. 在保证低延迟的情况下，提升事务处理吞吐量。

(2) 降低SQL查询响应时间。

1．4 技术路线（研究方法）

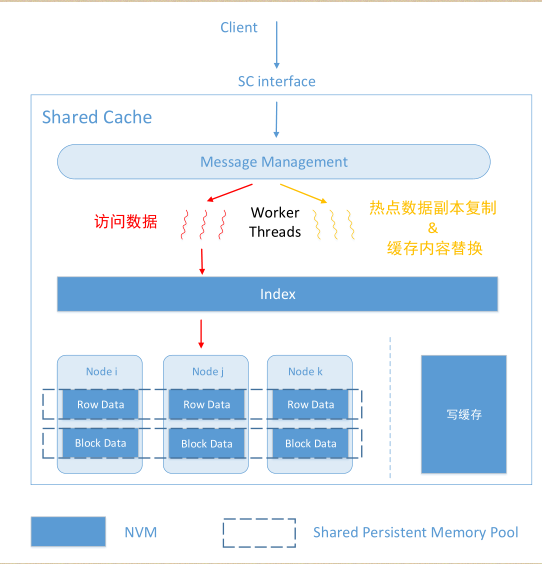
我们拟将SC部署在分布式数据库系统Cedar中，研究内容主要有三大方面，SC在Cedar中的架构、SC本身的架构设计、SC写缓存对Cedar日志模块的影响。

1. SC在Cedar中的架构。 如图1所示。 Cedar架构包括MS(计算节点)， CS（静态数据存储节点），UPS(事物管理节点)三种。我们将扩展Cedar的存储层次，在内存计算(MS)与存储(CS)之间增加一个NVM设备构成的SC。 SC通过RDMA与MS，CS，UPS互连。 UPS与CS为SC提供需要缓存的数据，MS从SC中获取数据用于查询计算。具体需要在Cedar系统的查询引擎中调整当前数据的访问路径，增加对SC的访问路径。



图表 1-SC在Cedar中的架构

1. SC本身架构。如图2所示。 主要包括几个模块：
   1. SC的接口。NVM设备通过RDMA完全联通，共享存储，对外透明。这使得SC对外操作是完全透明的，客户端通过RDMA，使用SC的接口访问。 当在SC中加入数据时，仅需要知道数据大小，SC内部自动实现空间分配。
   2. SC的数据存储。在SC内部，我们在基于的RDMA通信协议基础上，实现一个统一地址空间的数据存储。SC的内部存储包括行存储与块状存储两大类。 统一的地址空间将非常便利于我们存储数据以及在上面设计索引结构。
   3. SC内部索引结构。 SC对外提供按主键查询及范围查询的接口，因此在同一地址空间基础上，将设计高效的索引结构。特别是对数据的扫描访问，虽然数据通过RDMA互连，但由于分布在不同机器上，为避免数据在一个访问中出现频繁跨机器访问的问题，需要仔细考虑块状数据的存储分布。
   4. 普通工作线程。通过索引判断数据的存在性，进行数据访问。另外工作线程基于某种缓存替换策略实现块数据与行数据的内容替换。也可以考虑设计专门的后台替换线程，拆分工作线程的任务，以提高数据访问的效率。需要解决的是基于一个统一地址空间的并发操作问题。对读写冲突和写写冲突，需要结合数据库本身的并发机制进行处理。 对并发的写入操作而言，申请地址的操作也是冲突的。一种减缓该冲突的方法是对地址空间分区，使得并发的写入操作在申请地址时尽可能不落到同一个分区。
   5. SC访问消息管理。对SC的访问消息，进行统一的管理与记录，为每个消息分配工作线程， 另外辅助决策缓存的数据替换， 热点数据副本复制功能。
   6. 写缓存模块。负责写入事物管理中的某些数据。并且对外能够访问，以便于在事物处理节点进行数据恢复时能够提供数据。
   7. (可选) 热点数据副本复制。对热点数据，由于落在一台机器上，性能可能产生瓶颈。因此，对热点数据进行自动识别复制，增加访问的并行度。



图表 2-SC架构

1. 写缓存及Cedar日志管理模块重新设计。由于写缓存数据已经是持久化的。这对数据库的日志及恢复会产生影响。对这部分数据，其日志管理方式是可以有变化的。而其最直观的效用是减少日志恢复过程中的日志访问，减少日志恢复时间。具体而言，研究在提升日志恢复性能的前提下，保证事务的ACID性质的解决方案。

这里强调利用NVM来提升日志性能的方式是复杂的，方法也不唯一，当前整个学术与工业界都处于探索阶段。我们当下的一种思路如下：

数据存储在内存中(这提供了最高效的数据访问方式，也是内存数据库的立足之本)。同时利用NVM 和 SSD一起保证的数据持久化。在SSD上我们使用日志保证数据持久化。在NVM中我们可以直接物化数据(随机写)。

进一步而言, 将数据按照某种规则分为两个分区 N 和 S。属于N分区的数据使用NVM保证持久性，属于S分区的数据通过日志保证持久性。这样会产生三种不同类型的事务:

(1) 只访问S分区数据的事务（记为Ts），在日志落盘后提交。并且这里日志是成组(group commit)落盘的。

(2) 只访问N分区数据的事务（记为Tn），在NVM物化后提交。

(3) 同时访问两个分区的事务(记为Tsn)，需要两者均做到物化后才能提交。

恢复协议：恢复与当前数据库系统的思路类似。恢复时，我们只需要在日志与NVM中寻找到每个数据在已提交事务中的最近更新就行了。由于N分区与S分区没有任何依赖关系，所以两者的恢复完全可以同时进行。 对S分区可完全按照当下内存数据库的恢复来操作。对N分区的恢复更简单，只需要扫描所有数据项即可(可以轻易做到并行)。我们可以看到这样的恢复协议在恢复时间上具有优势： 首先，增加了并行性；其次，日志量减少（N区的日志大量减少），从而减少了恢复时间。

提交协议： 比当下内存数据库会复杂。 需要保证两件事， (1)Ts, Tn，Tsn事务之间不存在日志号比较靠后，但数据先完成持久化并提交的现象。(2) Tsn事务，只有在S区及N区均物化成功后才能提交。解决(1)、(2)设计思路不止一种。一种简单的解决思路是以S区的事务为标准，也就是Ts，及Tsn在N区的数据物化之后，通知S区。S区成组罗盘，对成功持久化的事务提交并返回客户端。简单而言，提交协议包括下述几个步骤:

(a), 事务请求LSN

(b), 对Ts 在日志缓冲区填充日志， 对Tn 在NVM上物化数据， 对Tsn的两部分，分别在NVM上物化，在日志缓冲区填充日志。

(c). 每一个Ts， Tsn在N区完成物化后都告知日志落盘线程它们已经物化完成了。这种告知可以通过将某些内存比特位置值实现。

(d), 落盘线程结合缓冲区的填充情况以及N区的物化情况，选择当下已经完成日志填充或NVM物化的最大LSN值，成组罗盘并提交。

以上只是一种最简单（但未必非常高效）利用物化缓存来提升日志恢复性能的协议。如何在保证事务处理性能的前提下利用物化缓存来提升日志性能，需要逐步探索完成。

此外，我们需要一种灵活的机制决定哪些事务是Ts事务，哪些事务是Tn事务。一方面在事务处理性能和恢复性能上取得折中。另一方面平衡SC缓存的换出机制，对SC缓存本身的性能不造成负面影响。（注意：写缓存的内容不能随便换出，否则数据无法恢复；因此不是所有事务都适合作为Tn事务，否则挤占缓存空间；另一方面，原本就应该被缓存的热数据更适合Tn事务，因为可以不付代价地获得Tn事务的好处。）

**2 研究计划**

2.1．项目期限：项目起止时间

2.2．进度安排：如果是一年的项目，按每季度设定预期的工作。如果是多年的项目，第一年按每季度写，后面几年按每年写

|  |  |  |  |
| --- | --- | --- | --- |
| **阶段** | **研究内容** | **时间** | **提交结果** |
| 1 | … |  | Demo、项目报告书等 |
| 2 | … |  | 源代码、原型系统、项目报告书等 |
| 3 | … |  | 源代码、项目报告书、专利、论文等 |

1. **预期成果**

论文：CCF-A类论文1篇（重点关注SIGMOD、VLDB、ICDE）

专利：2项

原型系统：比工业界同等系统性能提高50%以上，在Cedar上完成验证。