浙江工业大学

本科毕业设计外文翻译

（2013 届）

论文题目　RAMClouds: 可扩展的高性能 DRAM 存储

作者姓名　陈佳鹏

指导教师　陈　波

学科 (专业)　软件工程

所在学院　计算机科学与技术学院

提交日期　2013 年 06 月

RAMClouds: 可扩展的高性能 DRAM 存储

摘要： 在线存储的磁盘的方法正成为越来越严重的问题：他们的规模不能很好的满足大型 Web 应用程序，对磁盘容量的改善已经远远超过了在访问延迟和带宽的改善。本文提出一种关于数据中心存储新的方法：RAMCloud，信息完全保存在 DRAM 和大型系统所建立的聚集成千上万的商品服务器的主存储器上。我们认为，RAMClouds 可以提供基于磁盘的系统、100-1000 倍低访问延迟和吞吐量的持久的和可用的存储。低延迟和大规模的结合，提供了一种新的数据密集型应用程序。

关键字： DRAM

一、引言

四十年来，磁盘是存储计算机系统中在线信息的主要手段。在此期间磁盘技术经历了巨大的改进，它一直被更高级别的存储系统利用，如文件系统和关系数据库。但是，磁盘的性能并没有像它的容量那样迅速改善，为了满足大型的 Web 应用程序的需要，开发人员发现越来越难测量基于磁盘的系统。很多人都提出了新的基于磁盘的存储方法来解决这个问题，另外的人建议用 闪存设备更换磁盘。相比之下，我们认为，解决的办法是将磁盘中的主要核心数据转移到随机存取存储器上，磁盘变成备份/归档的角色。

在本文中，我们认为，一类被称为 RAMCloud 的新存储将为许多的未来的应用提供存储基底。一个 RAMCloud 将所有的信息存储在商品服务器的主存储器中，使用数百或数千个这样的服务器创建一个大型存储系统。因为任何时候所有的数据都是在 DRAM，RAMCloud 可提供基于磁盘的系统、100-1000 倍低访问延迟和吞吐量的持久的和可用的存储。虽然个别的内存是不稳定的，RAMCloud 可以使用复制和备份技术确保数据的耐用性和可用性，这等同于基于磁盘的系统。

我们认为 RAMClouds 将在三个方面从根本上改变存储行业的面貌。第一，通过省去了许多的可扩展性问题，这些问题削弱了当今开发人员的工作效率，它们简化了大型的 Web 应用程序的开发。第二，极低的延迟将提供更丰富的查询模型，提供了一种新的数据密集型应用程序。第三，RAMClouds 将提供需要“云计算”的可扩展存储基底和其他数据中心的应用程序。一个 RAMCloud 可以支持一个大型应用程序或许多小应用程序，并允许小型应用程序快速的扩展到大型应用程序，无需为开发人员增加额外的复杂性。

二、RAMCloud 概念

RAMClouds 是最有可能被用于在含有大量服务器的数据中心，粗略地划分分为两类：应用服务器，用来实现应用程序的逻辑，如生成 Web 页面或执行业务规

则;存储服务器,提供长期共享存储的应用程序服务器。传统以来,存储由文件或关系型数据库组成。但在近年来,为了提高可扩展性,开发出了各种新的存储机制,如 Bigtable 和 memcached。每个数据中心通常支持许多应用,从范围从只有一个应用程序的服务器小应用到千上万的专用应用程序和存储服务器组成的大型应用。

RAMCloud 代表了在系统中组织存储服务器一种新的方式。有两个关键属性用来区分 RAMCloud 和其他存储系统。首先,任何时刻所有的信息都保存在 DRAM 中。RAMCloud 不是像 memcached 那样的缓存,且数据不是存储在一个 I/O 设备上。DRAM 是数据永久的家,磁盘仅用于备份。其次,RAMCloud 必须自动扩展以支持数以千计的存储服务器,应用程序看到的是一个单一的存储系统,和存储服务器的实际数目不相关。

存储在 RAMCloud 上的信息必须像存储在磁盘上那样的耐用。例如,单个储存服务器的出错不应该造成数据丢失或几秒钟的系统失效。第四章第二节讨论实现这种级别的持久性和有效性的系统所需要的技术。

表 2.1 总结了现在 RAMCloud 的配置,此配置假设每台服务器有 64GB 的内存,也就是现在最经济有效范围内最大的内存(随着内存大小的增加,价格会有显著的增长)。这个配置提供了 1000 台服务器共 64TB 的储存空间,每 GB60 美元。算上附加的服务器,现在有可能构建容量大约 500TB 的 RAMClouds。在 5 到 10 年内,假设 DRAM 的技术会一直有进步,那么就有可能以小于每 GB5 美元构建容量达 1 至 10PB 的 RAMClouds。

表 2.1: RAMClouds 参考配置

# 服务器	1000
每台服务器的容量	64GB
总容量	64TB
服务器总开销	4000,000$
每 GB 开销	60$
总吞吐量	10^9

三、动机

3.1 应用程序扩展性

RAMClouds 产生的两个动机:应用程序、技术。从应用程序的立场来看,几十年来,关系型数据库被拿来做储存系统,但是关系型数据库无法适应现在大规模应用程序这个级别所需要的数据规模。实质上,每个流行的 Web 应用程序也已经意识到单个关系型数据库已经无法满足吞吐量的要求。随着网站的膨胀,它必须要经历一系列大量的修改,每一个系列都会引进专案技术来扩展其存储系统,

如分割数据到多个数据库中。这些技术在一段时间内有了成效，但是当网站到达一个新级别的规模时，扩展性的问题又出现了，那就需要更特殊更有目的的技术了。

例如，到 2009 年 8 月，Facebook 的储存系统已经包含了 4000 个 MySQL 服务器。数据分布在实例之间，实例之间的一致性明确地由 Facebook 应用程序的代码管理。即使这样，这些数据库服务器也不能满足 Facebook 吞吐量的要求，所以 Facebook 同时附加了 2000 台 memcached 服务器，这些服务器把最近使用的查询结果以 Key-Value 的形式存储在主内存中。不幸的是，memcached 和 MySQL 服务器之间的一致性必须由软件管理，这增加了应用程序的复杂性。

3.2 技术趋势

RAMClouds 的第二动机来源于磁盘技术的演化（见表 3.1 ）。在过去的 25 年中，磁盘容量增加了至少 10000 倍，而且有可能会在未来继续增加。不幸的是，尽管这样，磁盘上信息的访问率提高的更加慢：大数据块之间的传输率只提高了 50 倍，而寻道时间和旋转等待时间只提高了 1/2。

表 3.1: 25 年前和现在的硬盘技术对比

	1980 年代中期	2009 年	改善
硬盘容量	30 MB	500 GB	16667x
最大传输率	2 MB/s	100 MB/s	50x
延迟时间（寻道 + 旋转）	20 ms	10 ms	2x
容量/带宽（大块）	15 s	5000 s	333x
容量/带宽（小块）	600 s	58 days	8333x

参考文献

[1] Ammann A, Hanrahan M, Krishnamurthy R. Design of a memory resident DBMS[C]. IEEE COMPCON, 1985.

[2] Bitton D, Hanrahan M, Turbyfill C. Performance of complex queries in main memory database systems[C]. Proceedings of the Third International Conference on Data Engineering, 1987:72--81.

[3] Copeland G, Keller T, Krishnamurthy R, et al. The case for safe RAM[C]. Proceedings of the 15th international conference on Very large data bases, 1989:327--335.

[4] Copeland G, Franklin M, Weikum G. Uniform object management[J]. Advances in Database Technology—EDBT'90, 1990:253--268.

[5] DeWitt D J, Katz R H, Olken F, et al. Implementation techniques for main memory database systems[M]. Vol. 14.[S.l.]: ACM, 1984.

[6] Eich M H. A classification and comparison of main memory database recovery techniques[M]. [S.l.]: IEEE Press, 1989.

[7] Eich M H. MARS: The design of a main memory database machine[J]. Database Machines and Knowledge Base Machines, 1988, 43:325.

[8] Garcia-Molina H, Salem K. High performance transaction processing with memory resident data[C]. International Workshop on High Performance Transaction Systems, 1987.

[9] Gawlick D, Kinkade D. Varieties of concurrency control in IMS/VS fast path[J]. Database Engineering Bulletin, 1985, 8(2):3--10.

[10] Gray J, Putzolu F. The 5 minute rule for trading memory for disc accesses and the 10 byte rule for trading memory for CPU time[C]. ACM SIGMOD Record, 1987，16:395--398.

[11] Reuter G J, Gray J. Transaction Processing: Concepts and Techniques,>[J]. MorganKaufmann, San Mateo, CA, 1993.

[12] Gruenwald L, Eich M H. MMDB reload algorithms[C]. ACM SIGMOD Record, 1991，20:397--405.

[13] Hagmann R B. A crash recovery scheme for a memory-resident database system[J]. Computers, IEEE Transactions on, 1986, 100(9):839--843.

[14] Kim M Y. Synchronized disk interleaving[J]. Computers, IEEE Transactions on, 1986, 100(11):978--988.

[15] Lehman T J, Carey M J. Query processing in main memory database management systems[C]. ACM SIGMOD Record, 1986，15:239--250.

[16] Lehman T J, Carey M J. A study of index structures for main memory database management systems[C]. Conference on Very Large Data Bases, 1986，294.

[17] Lehman T J, Carey M J. A recovery algorithm for a high-performance memory-resident database system[C]. International Conference on Management of Data: Proceedings of the 1987 ACM SIGMOD international conference on Management of data: San Francisco, California, United States, 1987，27:104--117.

[18] Li K, Naughton J F. Multiprocessor main memory transaction processing[C]. Proceedings of the first international symposium on Databases in parallel and distributed systems, 2000:177--187.

[19] Patterson D A, Gibson G, Katz R H. A case for redundant arrays of inexpensive disks (RAID)[M]. Vol. 17.[S.l.]: ACM, 1988.

[20] Pucheral P, Thévenin J M, Valduriez P. Efficient Main Memory Data Management Using the DBGraph Storage Model[C]. Proceedings of the 16th International Conference on Very Large Data Bases, 1990:683--695.

[21] Reuter A. Performance analysis of recovery techniques[J]. ACM Transactions on Database Systems (TODS), 1984, 9(4):526--559.

[22] Salem K, Garcia-Molina H. DISK STRIPING t[C]. International Conference on Data Engineering, February 5-7, 1986, Bonaventure Hotel, Los Angeles, California, USA., 1986:336.

[23] Salem K, Garcia-Molina H. Checkpointing memory-resident databases[C]. Data Engineering, 1989. Proceedings. Fifth International Conference on, 1989:452--462.

[24] Salem K, Garcia-Molina H. System M: A transaction processing testbed for memory resident data[J]. Knowledge and Data Engineering, IEEE Transactions on, 1990, 2(1):161--172.

[25] Stonebraker M. Managing persistent objects in a multi-level store[C]. International Conference on Management of Data: Proceedings of the 1991 ACM SIGMOD international conference on Management of data: Denver, Colorado, United States, 1991，29:2--11.

[26] Whang K Y, Krishnamurthy R. Query optimization in a memory-resident domain relational calculus database system[J]. ACM Transactions on Database Systems (TODS), 1990, 15(1): 67--95.