基于RDMA和NVM的大数据系统一致性协议研究

吴昊[1,2]，陈康[1,2]，武永卫[1,2]，郑纬民[1,2]
1. 清华大学计算机科学与技术系，北京 100084；2. 北京信息科学与技术国家研究中心，北京 100084

摘要
分布式的存储系统以及计算系统是构造大数据处理系统的基础。系统的高可用性是任何一个分布式系统的基石，高可用技术一般依赖于一致性协议。讨论了经典的非拜占庭的分布式一致性协议以及新技术发展下的RDMA通信协议与NVM存储介质，通过RDMA和NVM的结合获得了更高性能的高可用系统。改进了一致性协议，使其能够更好地利用RDMA与NVM的特性。实现的系统在保证系统数据一致和可用的同时，有效地提高了协议实现的性能。实验表明，相比于现有的系统，实现的系统能够得到40%的性能提高。

关键词
大数据；分布式系统；一致性协议；远程内存直接访问；非易失存储器

中图分类号：TP302.8 文献标识码：A doi: 10.11959/j.issn.2096-0271.2019034

Research on the consensus of big data systems based on RDMA and NVM

WU Hao[1,2], CHEN Kang[1,2], WU Yongwei[1,2], ZHENG Weimin[1,2]
1. Department of Computer Science and Technology,Tsinghua University, Beijing 100084, China
2. Beijing National Research Center for Information Science and Technology, Beijing 100084, China

Abstract
Distributed storage systems and computing systems are the foundation for constructing big data processing systems. High availability of the system is the cornerstone of any distributed system. High-availability technologies generally rely on consensus protocols. The classic non-Byzantine distributed consensus protocol was discussed, as well as the RDMA communication protocol and NVM storage media under the development of new technologies to achieve higher performance high availability systems by combining them. The consensus protocol to make the better use of the features of RDMA and NVM was modified. The implemented system effectively improves the performance of the protocol while ensuring the consistency and availability of the system data. Experiments show that the system implemented in this paper can achieve 40% performance improvement compared to existing systems.

Key words
big data, distributed system, consensus protocol, RDMA, NVM

1 引言

随着分布式系统规模的增大，服务器宕机和网络服务中断等各种故障使得分布式系统中断服务的可能性越来越高。为了解决单点故障问题，维护系统的高可用性，分布式系统在实际应用中一般采用多副本的备份技术。这种方法不仅可以保证系统的高可靠性，还可以对请求进行相应的负载均衡，以提高服务的并发数量和扩展性。但是，这种多副本技术同样会带来系统一致性、可用性的问题。设计者通常使用分布式一致性协议来保证多副本系统中多台机器上数据的强一致性。

另外，随着远程内存直接访问（remote direct memory access，RDMA）等新型网络技术和非易失存储器（non-volatile memory，NVM）等新型存储介质的生产和普及，所有的系统服务在网络通信和存储过程中的性能都会得到巨大的提高，这为提高一致性系统的性能提供了机会。目前的一致性协议在新的网络技术和存储介质中的实现都不能完全地利用RDMA和NVM的特性。本文将展开这方面的研究，结合RDMA和NVM的特性，设计更加高效的分布式一致性协议。

2 概述

2.1 一致性协议概述

一致性问题是指集群节点在网络状态和服务器节点的运行状态都不可以控制的情况下，仍然能够对整个集群的某一问题或者集群的状态达成一致的共识。一般来说，如果分布式系统中有2f+1个副本服务器在工作，一致性协议一般可以容忍f个副本服务器出现故障，即只要系统中存在f+1个副本服务器正常工作，那么分布式系统就可以正常地对外提供服务。

图1展示了分布式一致性协议的简要发展过程和工业应用。其中，两阶段提交协议[1]通过投票阶段和正式提交阶段的两轮消息交互，使得某项决定达成一致。而三阶段提交协议[2]是两阶段提交协议的阻塞问题的优化方案。Quorum算法[3]通过限定参与读写操作的副本的最小数目，实现读写操作的互斥，从而支持多副本的并发更新。在Lamport提出了Paxos算法[4]后，很多类Paxos算法陆续出现，以适应不同的应用环境。如Multi-Paxos[5]就是通过选举出一个领导者，减少了Paxos协议的第一轮通信，避免了多个节点同时提交指令时的冲突问题。ZAB（ZooKeeper atomic broadcast）协议[6]是Multi-Paxos的变种，是一种能够支持崩溃恢复的原子广播算法。Raft算法[7]是研究者为了降低Paxos算法的复杂性而设计的一个管理日志的一致性协议。还有一些弱领导者协议（如Fast Paxos[8-9]、Generalized Paxos[10]、FGGC[11]、EPaxos[12]）也有不同方面的优化，但是协议相对复杂，实际应用很少。

一致性系统在工业界也有很多产品。目前工业界使用的一致性算法大多是Paxos算法的变种，如Google公司的Chubby[13]和Megastore[14]使用的就是Multi-Paxos协议，Apache公司的ZooKeeper[15]使用的是ZAB协议。

2.2 RDMA概述

RDMA网络技术是一种新型网络技术，它可以绕过远端CPU直接读取远端内存，同时还具有高吞吐量和低时延的特性，这些特性使其可以有效地降低集群中

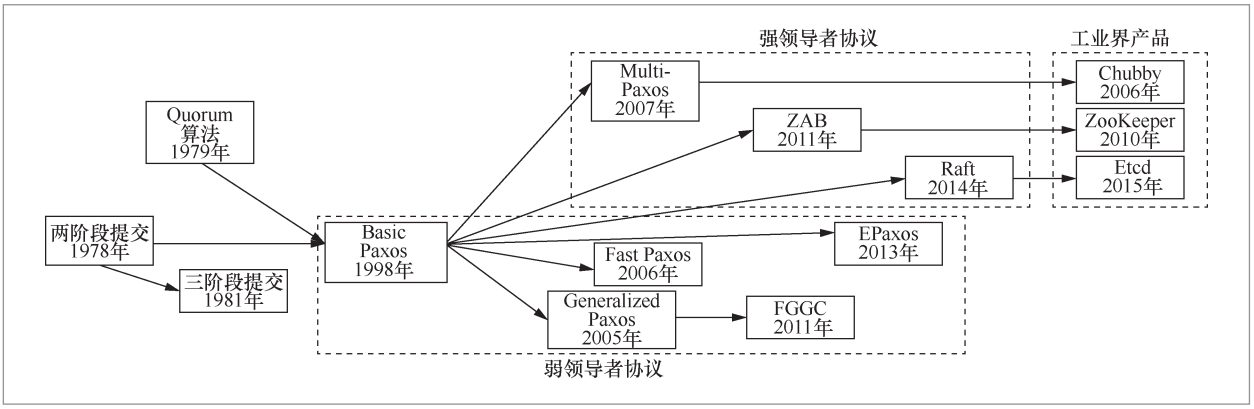


图1 分布式一致性协议的简要发展过程与工业应用

网络通信的开销。目前，RDMA网络在数据中心已经被逐渐普及[16]。

RDMA的通信模式分为可靠连接（reliable connection，RC）、不可靠连接（unreliable connection，UC）和不可靠的数据报（unreliable datagram，UD）3种模式。RDMA支持的通信原语分别是RDMA Write、RDMA Read、Send、Receive以及原子操作（Atomic）。其中，Read和Write是单边的操作。单边的读写操作可以绕过远端的CPU。Send和Receive是双边的操作，双边的操作要求通信双方的CPU必须同时参与才能完成通信。原子操作包括Fetch and Add和Compare and Swap等。不同的连接模式支持不同的RDMA原语操作，见**表1**。之前的工作表明，在速度方面单边的操作比双边的操作快两倍左右[17-18]，因此本文采用RC模式以及单边的读写操作。

表1 不同模式下的RDMA原语和最大传输单元

原语	UD	UC	RC
Send	√	√	√
Receive	√	√	√
RDMA Write		√	√
RDMA Read			√
Atomic			√

2.3 NVM概述

NVM[19]是新型存储设备，它能够在设备断电之后仍然保存其中存储的数据。新型非易失性存储器（如PCM[20]）具有可字节寻址、低时延、高密度等特点，其能耗比磁盘低90%以上，同时读写的时延也和DRAM相近。但是与DRAM不同的是，NVM的读写速度不对称，其读数据的速度快于写数据的速度。而且类似于固态硬盘（solid state drive，SSD），其擦写次数存在上限，因此其寿命有限，也需要考虑磨损均衡的问题。

表2比较了不同类型的存储介质在能耗、读写时延以及最大擦写次数等方面的性能。从**表2**可以明显看出各种不同存储介质的区别。

NVM在很多方面和系统内存使用的动态随机存取存储器（dynamic random access memory，DRAM）相似，尤其是其读写速度快、容量大、可字节寻址的特性，使其有望代替主存。NVM还具有DRAM

表2 不同类型存储介质的比较

特性	磁盘	PCM	DRAM
读能耗/(J/GB)	65	1	0.8
写能耗/(J/GB)	65	6	1.2
读时延	~5 ms	~60 ns	20~50 ns
写时延	~5 ms	50~120 ns	20~50 ns
最大擦写次数/次	∞	10^6~10^8	∞

不具有的非易失性的特点，这使其可以应用到一致性系统中，作为日志存储的存储器，提高一致性系统的性能。

3 基于RDMA和NVM的一致性协议

3.1 RDMA、NVM与一致性协议的结合

在传统的一致性协议中，协议除了要处理指令的共识问题，还要保证指令的持久化。考虑到集群可能出现全部宕机的问题，因此要保证指令在系统重启时可以从持久化的存储中恢复出来。传统的一致性协议使用磁盘或SSD作为持久化指令的存储介质。随之而来的问题是协议中的领导者在复制日志的时候，不知道日志是否已经被持久化。领导者必须等待其他副本进程回复日志已经被持久化的消息后，协议算法才能继续。由于存储介质的问题，即使在RDMA通信环境下，这个问题也同样存在。如APUS[21]也只是在系统的网络层设计中用RDMA网络替换了传统的以太网，副本进程通过RDMA网络接收复制的指令消息，将其存入SSD后回复领导者。这种方式并没有完全利用RDMA的特性，尤其是RDMA绕过远端CPU的特性。

将NVM作为一致性系统中的存储介质，可为解决上述问题提供帮助。首先，NVM可以替代内存与RDMA直接进行通信，即服务器可以使用RDMA对远端的NVM直接进行读写；其次，NVM还具有持久性，使得使用RDMA通信后的消息同时也被持久化。这样就使得通信和持久化两个问题都通过RDMA的Write操作解决了。使用NVM后，领导者通过RDMA复制日志时，就不需要等待副本进程的回复，通过检查自身的状态就可以判断日志是否被持久化到副本进程中，进而继续判断日志的共识过程是否成功。当领导者使用RDMA的Write操作成功之后，远端的RDMA网卡会发送一个ACK到本地的网卡中，在本地网卡的完成队列（completion queue，CQ）中生成一个发送消息完成的状态。在RDMA的设计中，本地服务器首先调用RDMA Write操作向远端服务器发送消息，远端的服务器网卡收到该消息后会立即返回给本地服务器网卡一个ACK消息。因此Write操作的ACK消息不能确保数据被写入NVM中，即数据不能确保是否被持久化。为了解决这个问题，笔者设计在每个Write操作之后，立刻发送一个RDMA中的Read操作。该Read操作读取大小为0 byte的数据，读取的远端内存地址即Write操作中写入远端内存的地址，如Hyper Loop[22]中的设计。这个Read操作会保证对应位置的数据已经被成功写入NVM中。本地程序只需要检测本地网卡的CQ中对应Read操作的状态，就可以判断对应的数据是否被成功写入NVM。

使用RDMA网络和NVM存储为网络中断后的领导者选举引入了新的问题。传统协议的解决方法是每次发送指令时，指令都会附带一个系统版本信息。这是一个整型数值，用来存储系统版本。每次系统重新选举领导者的时候，这个版本号就会增加1。当副本进程收到指令的时候，副本进程判断指令附带的版本信息，如果是旧的版本就忽略掉，从而避免旧的领导者对整个日志的破坏。但是由于笔者提出的方案绕过了副本机器的CPU，副本进程无法对消息进行判断，因此还需要对选举的流程进行修改，以保证协议的正确性。

本文针对RDMA网络和NVM存储介质的特性对Multi-Paxos协议进行了修改，并对修改后的协议进行了实现。本文修改了协议流程中对一致性成功的判定方式以及领导者选举和日志恢复中的部分流程，

利用RDMA Write操作的绕过目标服务器CPU的特性、NVM的持久性和可字节寻址的特性，使得修改后的协议具有低时延、高吞吐量和低CPU消耗等特点。下面将介绍修改后的具体流程。

3.2 日志复制

图2为日志复制流程。在协议正常运行的情况下，日志的编号和复制都由领导者负责。客户端通过RDMA的Write操作将指令发送给领导者服务器。领导者服务器会不断轮询每一个客户端对应的在本地注册的内存，检测是否有客户端发送指令过来。当领导者服务器接收到从客户端发送过来的指令后，领导者为该条指令分配一个序列号和这个指令在整个系统日志里的地址。然后，领导者服务器将该指令复制到其他的服务器上，领导者通过RDMA的Write操作直接将这条指令写入对应副本机器中的NVM中，指令的地址即刚才分配的地址。当领导者将指令发送到副本机器后，领导者会继续发送对应位置的一个Read操作，这个Read操作读取的字节为0，用于保证数据被持久化到NVM中。同时这个Read操作可以在Write操作后直接使用，不需要等待Write操作的返回。领导者可以通过从RDMA的完成队列中获取相应Read操作对应的消息，判断指令是否被成功写入。领导者通过计算单个指令发送成功的次数判断指令是否可以提交，当单个指令发送成功的次数为f+1次，即领导者已经成功地将该指令发送到f+1个副本机器中时，即可认为该指令已经完成了共识过程。领导者提交该指令，发送commit消息给所有的副本服务器，同时执行该指令，并将执行的结果返回给相应客户端。如果指令的发送次数没有超过f+1次，领导者将不断发送该指令给未接收到的副本服务器，直到其接收到该指令为止。

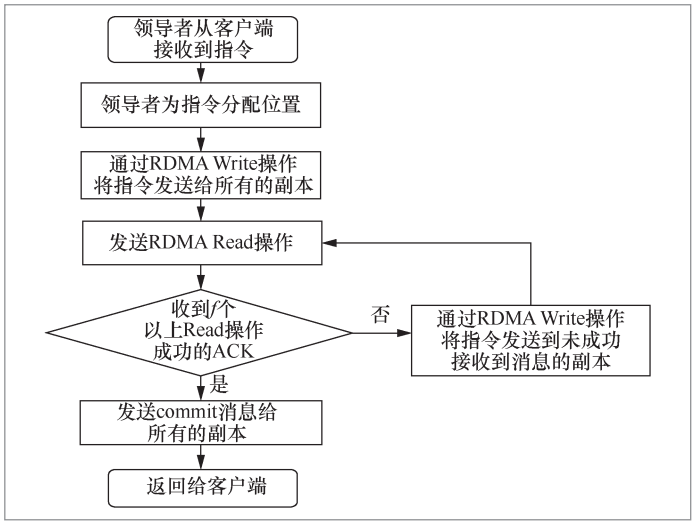


图 2 日志复制流程

3.3 领导者选举与日志恢复

由于协议是强领导者协议，因此当领导者出现故障时，整个系统就要重新选举出一个新的领导者，才能继续对外提供服务。每个领导者都对应着一个任期编号（term）。当副本进程检测到与领导者的连接中断时，该副本进程会进入选举领导者的状态。它会将term值增加1，并向其他的副本服务器发送选举其为领导者的消息，在该消息中附带自己的term值。然后等待其他副本进程为其投票。其他副本进程收到该消息后，根据自身情况回复相应消息：如果副本进程检测到与领导者的连接畅通，那么回复不投票给发送该指令的副本进程，并将领导者的编号信息发送给该副本进程；如果其检测到与领导者也断开了连接，但是已经投票给了其他term值更高的副本进程，那么将回复不投票给发送该指令的副本进程；如果其检测到与领导着也断开了连接，并且没有投票给任何其他的副本进程，那么回复投票给发送该指令的副本进程。当某一个副本进程收到大于f+1个为其投票的信息后，该副本进程的角色变为领导

者，并进行日志恢复的过程，同时阻止其他副本进程进行新一轮的选举。

系统在复制指令的时候使用了RDMA的Write操作将指令写入副本进程的NVM中，领导者直接绕过了副本进程的CPU和操作系统。这样的好处是减少了副本进程的CPU消耗，同时降低了时延；坏处是副本进程不会判断写入的指令是否是由新的领导者写入的。为了解决这个问题，系统在选举领导者的时候，当副本进程收到某个副本进程选举领导者成功消息的时候，该副本进程就会断开与旧的领导者的RDMA连接。这样使得旧的领导者无法写入任何新系统中存活的副本进程的NVM中，就不会破坏整个日志。图3为领导者选举流程。

当新的领导者被选举出来后，新的领导者会对所有的日志进行恢复操作。重新选举出来的领导者发送恢复指令给其他与之相连的副本机器，询问其所持有的日志中的序号最大的指令和没有提交的指令。当领导者收到所有的回复信息后，领导者整理回复信息，更新已经提交的日志，然后重新提交那些没有成功提交的指令。对于没有编号的日志，即日志中的空洞，领导者同样提交空洞的指令。当所有的指令都提交完成后，日志恢复过程结束，系统可以继续对外提供服务。

3.4 故障检测

一致性系统中只要保证副本服务器发生故障的数量不超过f个，系统就仍然能够继续提供服务。新设计的一致性协议为强领导者协议，因此不同服务器之间不需要互相监控状态，副本服务器只需要监控领导者的状态即可。

副本服务器会通过心跳机制的方法检测领导者的故障状态。系统规定每一个副本服务器每间隔100 ms向领导者发送一次心跳请求。由于通信网络是RDMA网络，因此心跳

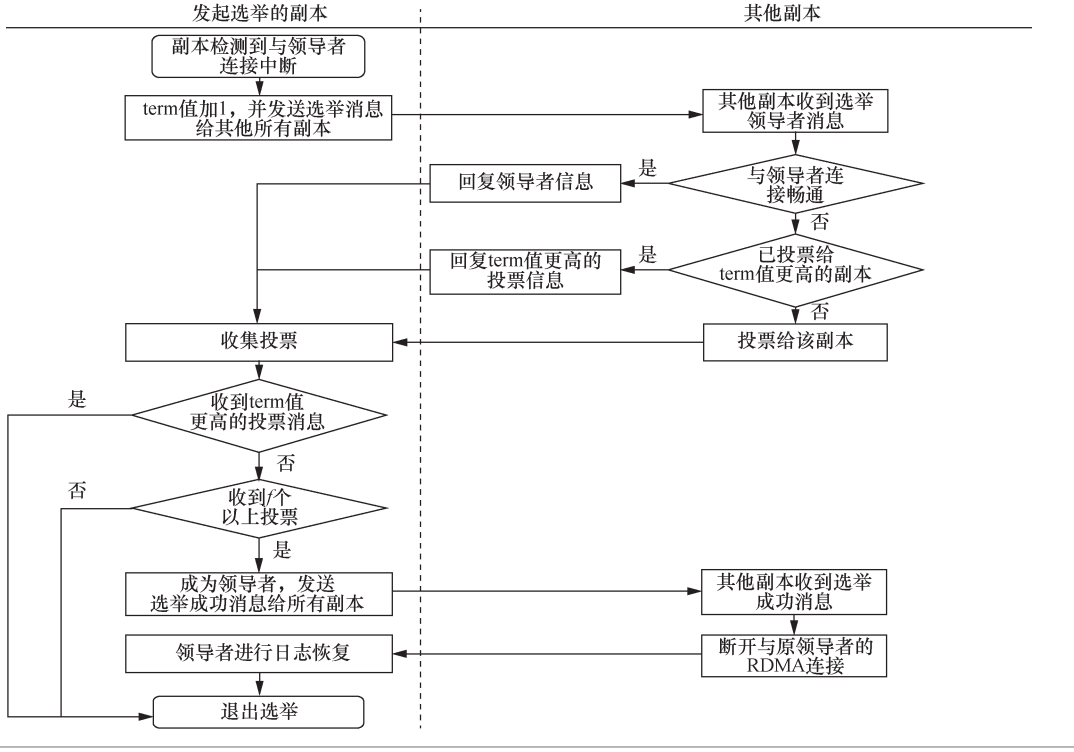


图 3 领导者选举流程

消息的实现也是用RDMA的原语实现的。副本服务器会通过RDMA的Read原语读取领导者内存中的一处地址。Read操作结束后会在RDMA相应的完成队列中存入一个完成事件。之后读取这个完成事件，并通过其中的状态信息判断这次Read操作是否成功。如果此次操作成功，那么表明领导者是在线状态，没有发生故障；如果此次Read操作的状态是失败，那么表明当前副本服务器与领导者的连接出现故障，当前副本服务器进入领导人选举的状态，向其他副本服务器发送选举领导人信息，选举其为新的领导人。

进入选举过程后，收到选举消息的副本机器会把与之前的领导者之间的连接切断，销毁相应的资源。断开连接的副本服务器若想重新加入系统集群中，要向领导者发送join指令，该指令表明有新的副本服务器要加入系统。领导者收到该指令后，同样为该指令分配其在日志中的位置，并将其复制到其他副本服务器中。当领导者成功将其发送到了f+1个副本服务器上时，表明该指令已经成功提交到日志中，该副本服务器可以加入系统中。同时，系统中副本机器的多数派的数量也要发生相应的改变，领导者在复制日志的时候也要将日志复制到该副本机器中。

4 相关工作

近年来，使用硬件和一致性协议结合的方法对一致性协议进行优化的工作也有很多。NetPaxos[23]利用网络交换机中消息的顺序实现Paxos算法，同时该工作还能够兼容OpenFlow[24]的应用程序编程接口（application programming interface，API），具有很高的应用价值。NOPaxos[25]则首先设计了一个特殊的网络，以支持有序不可靠广播（ordered unreliable multicast，OUM），并在该网络下实现一致性协议。由于该网络是有序的网络，因此要比Paxos等简单很多。同样还有将Paxos协议实现在可编程路由器中，如使用P4语言[26]实现Paxos协议的Switchcy-Paxos[27]。也有学者使用现场可编程门阵列（field programmable gate array，FPGA）[28]在网络层实现ZAB协议[29]，性能也非常优秀。

目前使用RDMA网络实现一致性协议的工作有DARE[30]系统和APUS。其中，DARE可以看作将Raft协议在RDMA环境下的实现，其利用RDMA的原语对Raft进行了修改。APUS则是Multi-Paxos算法在RDMA中的实现，同时使用SSD存储日志。与本文的实现比较，二者各有不同的缺陷。DARE由于协议的设计，需要两轮通信才可以提交指令，时延比本文的工作要高很多，同时DARE只能运行在内存中，并没有做日志的持久化工作。APUS则存在之前提到的问题，虽然它是一轮通信的协议，但是其领导者必须等待副本的回复才可以判断日志是否可以提交，时延也比本文的工作高。在吞吐量方面，三者相差不大。由于APUS没有提供直接测试的客户端程序，因此第5节中将本文工作和DARE进行了对比。

除了一致性协议的工作，使用RDMA进行数据复制的工作还有Tailwind[31]。Tailwind的复制过程中使用RDMA的Write操作进行数据复制，与本文的工作有一定程度的重叠。不过Tailwind和本文的工作也有3点区别。一是Tailwind使用的是Primary-Backup的模式，使用f+1个副本容忍f个副本发生错误，而本文使用的是传统的一致性协议设计，使用2f+1个副本容忍f个副本发生错误，提供了更高的可用性。在f个副本发生错误后，仍然能够继续对外提供服务。二是Tailwind的工作中提供了关于数据恢复的方法，但是没有关于选举领导者的工作。三是在数据持久化的工作中，本文使用NVM硬件和RDMA中的Read操

作确保数据的持久性。但是Tailwind在保证数据持久性方面没有任何的工作，只是假设服务器存在备用的电源设备，并且服务器能够在断电后利用备用的电源设备将PCIe控制器和RDMA网卡缓存中的数据写入SSD中。因此Tailwind并不具有任何应用价值。同时Tailwind没有提供其实现的代码，因此本文工作也无法和其进行测试对比。

5 系统性能测试

5.1 测试环境

本文使用C++语言对修改后的Multi-Paxos协议进行了实现，同时使用基于InfiniBand[32]网络设施的服务器集群对实现的一致性系统进行测试。集群由10台服务器构成，每台服务器的配置见**表3**。其中，笔者使用5台服务器作为一致性系统的服务端，另外5台服务器作为客户端，负责和服务端进行通信并发送请求。由于目前市面上没有可用的NVM硬件，因此使用内存模拟测试，同时由于NVM的写时延比内存高，因此在测试时，笔者修改了CentOS内核，使得系统内存的写时延增加150 ns，从而匹配NVM性能。

5.2 性能测试

在性能测试中，本文主要从时延和吞吐量两个方面对一致性系统进行测试。在测试系统时延的过程中，笔者分别对数据大小为8 byte、32 byte、128 byte、512 byte和2 048 byte的指令提交时延进行了测试。每次测试都将指令分别提交，即当客户端收到一个指令提交成功的回复后，再提交下一个指令。每次测试提交10 000条指令，然后对所得的结果取平均值。在测试吞吐量时，笔者随机生成测试数据，数据量大小为32 byte，通过不同数量的客户端同时进行提交，并测试性能。

表 3 实验服务器配置

配置	具体信息
操作系统	CentOS 7.2
CPU	Intel Xeon CPU E5-2643 v4 12核，主频3.40 GHz
内存	128 GB
网卡	Mellnaox ConnectX-3 Infiniband网卡，56 Gbit/s

测试结果如**图4**所示。由于本文工作是单轮通信协议，而DARE是两轮通信协议，因此本文工作的时延比DARE低。在时延测试中，当数据大小为8 byte时，单次提交指令的时延最低可达到6 μs，相比DARE的9 μs时延来说约减少了33%。由于RDMA操作的时延随着数据大小的增加而增加，因此二者的时延也随着数据大小的增加而增加。

在吞吐量测试中，二者的吞吐量都随着客户端连接数量的增加而增大。DARE协议在第二轮通信中要比本文工作多发送一轮日志的尾指针，发送数据量比本文多一些，因此本文工作的吞吐量要大一些。本文的协议中请求可以并行处理，而DARE只可以等待前面的请求处理结束后才开始处理之后的请求。为了提高性能，DARE在设计中使用了批量处理操作，将相同操作类型的请求打包到一起。随着连接的客户端数量增加，领导者服务器接收到的请求也随之增多，DARE每一次批量处理后的数据大小也随着增大。RDMA的带宽利用率是随着传输数据大小增加而增加的，因此本文工作和DARE的吞吐量差距也随着客户端连接数的增加而减小。

6 结束语

本文调研了分布式一致性协议的研究现状以及新的网络技术RDMA和新的存储介质NVM的特性。针对RDMA和NVM

的特性，设计了适应RDMA和NVM的新协议。该协议能够充分地利用新技术以及新硬件的特性，使得分布式一致性系统的性能得到了巨大的提升。

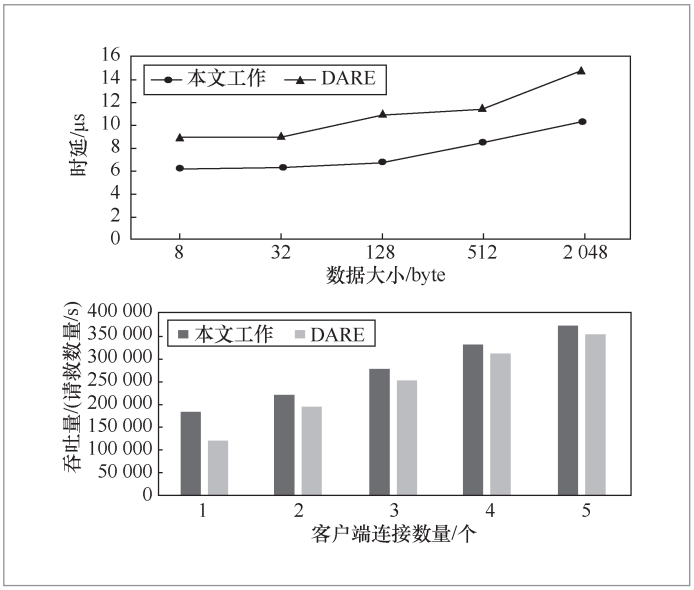


图 4 时延和吞吐量测试结果

参考文献:

[1] GRAY J, LAMPORT L. Consensus on transaction commit[J]. ACM Transactions on Database Systems (TODS), 2006, 31(1): 133-160.

[2] SKEEN D. Nonblocking commit protocols[C]// The 1981 ACM SIGMOD International Conference on Management of Data, April 29-May 1, 1981, Ann Arbor, USA. New York: ACM Press, 1981: 133-142.

[3] GIFFORD D K. Weighted voting for replicated data[C]//The 7th ACM Symposium on Operating Systems Principles, December 10-12, 1979, Pacific Grove, USA. New York: ACM Press, 1979: 150-162.

[4] LAMPORT L. Paxos made simple[J]. ACM Sigact News, 2001, 32(4): 18-25.

[5] CHANDRA T D, GRIESEMER R, REDSTONE J. Paxos made live: an engineering perspective[C]//The 26th Annual ACM Symposium on Principles of Distributed Computing, August 12-15, 2007, Portland, USA. New York: ACM Press, 2007: 398-407.

[6] HUNT P, KONAR M, JUNQUEIRA F P, et al. ZooKeeper: wait-free coordination for Internet-scale systems[C]//USENIX Annual Technical Conference, June 23-25, 2010, Boston, USA. [S.l.: s.n.], 2010.

[7] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm[C]//2014 USENIX Annual Technical Conference, Junuary 17-20, 2014, Philadelphia, USA. [S.l.: s.n.], 2014: 305-319.

[8] LAMPORT L. Fast Paxos[J]. Distributed Computing, 2006, 19(2): 79-103.

[9] ZHAO W. Fast Paxos made easy: theory and implementation[J]. International Journal of Distributed Systems and Technologies (IJDST), 2015, 6(1): 15-33.

[10] LAMPORT L B. Generalized Paxos: U.S. patent 7698465[P]. 2010-04-13.

[11] SUTRA P, SHAPIRO M. Fast genuine generalized consensus[C]//2011 IEEE 30th International Symposium on Reliable Distributed Systems, October 4-7, 2011, Madrid, Spain. Piscataway: IEEE Press, 2011: 255-264.

[12] MORARU I, ANDERSEN D G, KAMINSKY M. There is more consensus in egalitarian parliaments[C]//The 24th ACM Symposium on Operating Systems Principles, November 3-6, 2013, Farminton, USA. New York: ACM Press, 2013: 358-372.

[13] BURROWS M. The chubby lock service for loosely-coupled distributed systems[C]// The 7th Symposium on Operating Systems Design and Implementation, November 6-8, 2006, Seattle, USA. Berkeley: USENIX Association, 2006: 335-350.

[14] BAKER J, BOND C, CORBETT J C, et al. Megastore: providing scalable, highly available storage forinter active services[C]// The 5th Biennial CIDR Conference, January 9-11, 2011, Asilomar, USA. [S.l.: s.n.],

2011: 223-234.

[15] HUNT P, KONAR M, JUNQUEIRA F P, et al. ZooKeeper: wait-free coordination for internet-scalesystems[C]//USENIX Annual Technical Conference, June 23-25, 2010, Boston, USA. Berkeley: USENIX Association, 2010.

[16] GUO C, WU H, DENG Z, et al. RDMA over commodity ethernet at scale[C]//The 2016 ACM SIGCOMM Conference, August 22-26, 2016, Florianopolis, Brazil. New York: ACM Press, 2016: 202-215.

[17] TSAI S Y, ZHANG Y. Lite kernel RDMA support for datacenter applications[C]// The 26th Symposium on Operating Systems Principles, October 28-31, 2017, Shanghai, China. New York: ACM Press, 2017: 306-324.

[18] MACARTHUR P, RUSSELL R D. A performance study to guide RDMA programming decisions[C]//2012 IEEE 14th International Conference on High Performance Computing and Communication, June 25-27, 2012, Liverpool, UK. Piscataway: IEEE Press, 2012: 778-785.

[19] CHEN A. A review of emerging non-volatile memory (NVM) technologies and applications[J]. Solid-State Electronics, 2016(125): 25-38.

[20] MAOW, LIU JN, TONG W, et al. A review of storage technology research based on phase change memory[J]. Chinese Journal of Computers, 2015, 38(5): 944.

[21] WANG C, JIANG J, CHEN X, et al. Apus: fast and scalable Paxos on RDMA[C]// The 2017 Symposiumon Cloud Computing, September 25-27, 2017, Santa Clara, USA. New York: ACM Press, 2017: 94-107.

[22] KIM D, MEMARIPOUR A, BADAM A, et al. Hyperloop: group-based NIC-offloading to accelerate replicated transactions in multi-tenant storage systems[C]//The 2018 Conference of the ACM Special Interest Group on Data Communication, August 20-25, 2018, Budapest, Hungary. New York: ACM Press, 2018: 297-312.

[23] DANG H T, SCIASCIA D, CANINI M, et al. Netpaxos: consensus at network speed[C]// The 1st ACM SIGCOMM Symposium on Software Defined Networking Research, June 17-18, 2015, Santa Clara, USA. New York: ACM Press, 2015.

[24] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[25] LI J, MICHAEL E, SHARMA N K, et al. Just say noto Paxos overhead: replacing consensus with network ordering[C]//The 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI16), November 2-4, 2016, Savannah, USA. Berkeley: USENIX Association, 2016: 467-483.

[26] BOSSHART P, DALY D, GIBB G, et al. P4: programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.

[27] DANG H T, CANINI M, PEDONE F, et al. Paxos made switch-y[J]. ACM SIGCOMM Computer Communication Review, 2016, 46(2): 18-24.

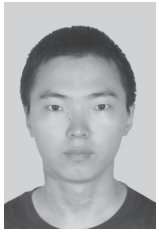
[28] SULAIMAN N, OBAID Z A, MARHABAN M, et al. Design and implementation of FPGA-based systems-areview[J]. Australian Journal of Basic and Applied Sciences, 2009, 3(4): 3575-3596.

[29] ISTVÁN Z, SIDLER D, ALONSO G, et al. Consensus in a box: inexpensive coordination in hardware[C]//The 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI16), March 16-18, 2016, Santa Clara, USA. Berkeley: USENIX Association, 2016: 425-438.

[30] POKE M, HOEFLER T. DARE: high-performance state machine replication on RDMA networks[C]//The 24th International Symposium on High-Performance Parallel and Distributed Computing, June 15-19,

2015, Portland, USA. New York: ACM Press, 2015: 107-118.
[31] TALEB Y, STUTSMAN R, ANTONIU G, et al. Tailwind: fast and atomic RDMA-based replication[C]//The 2018 USENIX Annual Technical Conference, July 11-13, 2018, Boston, USA. [S.l.: s.n.], 2018: 851-863.
[32] PANDA D K, KOOP M, BALAJI P. Tutorial: infiniband and 10-Gigabit Ethernet for dummies[C]// IEEE Symposium on High Performance Interconnects, August 25-27, 2009, New York, USA. Piscataway: IEEE Press, 2009.

作者简介



吴昊(1992-),男,清华大学计算机科学与技术系硕士生,主要研究方向为分布式系统。

陈康(1976-),男,清华大学计算机科学与技术系副教授,中国计算机学会(CCF)会员,主要研究方向为分布式系统、存储系统等。

武永卫(1974-),男,清华大学计算机科学与技术系教授,CCF高级会员,主要研究方向为并行和分布式处理、云计算和存储等。

郑纬民(1946-),男,清华大学计算机科学与技术系教授、博士生导师,CCF会士,主要研究方向为计算机架构、操作系统、存储和分布式计算等。

收稿日期: 2019-05-05
基金项目: 国家重点研发计划基金资助项目(No.2016YFB1000504);国家自然科学基金资助项目(No.61433008, No.61373145, No.61572280, No.U1435216);中国博士后科学基金资助项目(No.2018M630162)
Foundation Items: National Key Research and Development Program of China(No.2016YFB1000504), The National Natural Science Foundation of China (No.61433008, No.61373145, No.61572280, No.U1435216), China Postdoctoral Science Foundation (No.2018M630162)