门限 Byzantine quorum 系统及其在分布式存储中的应用

张 薇[1,2], 马建峰[1], 王良民[1], 郭渊博[3]
(1. 西安电子科技大学计算机网络与信息安全教育部重点实验室, 陕西西安 710071;
2. 武警工程学院电子技术系, 陕西西安 710086;
3. 解放军信息工程大学电子技术学院, 河南郑州 450004)

摘 要: 针对纠删编码和门限方案在分布式存储中的应用, 提出门限 Byzantine quorum 系统(TBQS)的概念. 该系统与数据分离算法相结合, 可以构造可靠性较强的存储系统, 当故障服务器个数不超过服务器总数的 1/4 时, 利用 TBQS 设计存储策略可以实现容错和无间断服务. 讨论了 TBQS 的存在性, 构造了两类 TBQS 并对其效率进行分析, 在此基础上设计了基于 TBQS 的分布式读写协议.

关键词: 分布式存储; 容错; quorum 系统; Byzantine 故障
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2008) 02-0314-06

Threshold Byzantine Quorum System and Distributed Storage

ZHANG Wei[1,2], MA Jian-feng[1], WANG Liang-min[1], GUO Yuan-bo[2]
(1. *Key Laboratory of Computer Network and Information Security under Ministry of Education, Xidian University, Xi'an, Shaanxi 710071, China;*
2. *Engineering Institute of the Armed Police, Xi'an, Shaanxi 710086, China;*
3. *The School of Electronic Technology, Information Engineering University of PLA, Zhengzhou, Henan 450004, China*)

Abstract: Distributed storage use erasure coding and threshold scheme to provide security and reliability. We present threshold Byzantine quorum system (TBQS), which can be used to design reliable storage strategy where up to one fourth of the servers might exhibit Byzantine faults. TBQS can provide fault-tolerance and serviced without interruption together with erasure coding or threshold schemes. We discussed the existence of TBQS, and proposed two types of TBQS called threshold f-masking system and grid TBQS respectively. We also yield a read-write protocol base on TBQS that is both time-efficient and reliable.

Key words: distributed storage; fault tolerance; quorum system; Byzantine fault

1 引言

存储系统的可靠性是一个不容回避的研究课题. 对于存储系统的设计者和使用者来说, 最为关心的问题是当网络中某些服务器出错时, 正常的数据服务是否够继续进行.

为提高数据服务的可靠性, 分布式存储常使用备份、纠删编码与门限方案等方法处理数据. 在备份系统中, 所有服务器上都保存着数据完整的副本, 因而只要系统中有一个未失效服务器, 就可以为合法用户提供数据服务. 然而, 为保证数据的一致性, 在更新时要对所有副本更新, 这将带来较大的通信量和管理开销. 为解决这一问题, 人们利用 quorum 系统来制订存储策略, quorum 系统是一个全集上某些子集的集合, 这些子集之间满足交汇性(intersection property), 即其中任意两个子集均相交, 每个子集称为一个 quorum. 如果将全体服务器集合视为全集, 则一个 quorum 由其中的一部分服务器构成. 所有 quorum 构成的集合系统称为 quorum 系统.

Quorum 系统的交汇性可以较好地满足分布式系统中对共享数据一致性的要求, 因而常常被用于构建对可靠性要求较高的分布式数据服务. 在基于 quorum 的备份服务中, 用户无论读还是写数据, 均要与一组服务器进行通信, 如果用户将数据写入一个 quorum, 则从另一个 quorum 中读数据时必然可以读到最后一次写入的数据. 这样在减少系统负载的同时保证了数据的一致性. 如果考虑服务器故障的情形, 则还要对 quorum 的交汇

收稿日期: 2007-01-22; 修回日期: 2007-10-31
基金项目: 国家 863 高技术研究发展计划(No. 2007AA01Z429, 2007AA01Z405); 国家自然基金重点项目(No. 60633020); 国家自然科学基金(No. 60573036, 60503012)

性作更多的限制, 为此 Malkhi 和 Reiter[1] 研究了服务器出现 Byzantine 故障时的情形, 而他们提出的屏蔽 quorum 系统 MQS(masking quorum system) 可以在此时保证数据的可用性.

与备份相比, 纠删编码和门限方案在安全性和效率上具有较大优势. 纠删编码可以提供数据冗余而避免复制所带来的系统负载. 门限方案则可以避免多个副本的存在带来的安全隐患. 出于对安全性与效率的考虑, 近年来人们更倾向于用纠删编码和门限方案取代备份来构造存储系统. 然而这两种方案的使用却带来了新的可靠性问题.

MQS 只适用于备份系统, 而在基于纠删编码和门限方案技术的存储系统中, 只有当未失效服务器数量大于某个门限值 m 时, 才有可能恢复数据. 用户向一个 quorum Q1 中写入数据, 从另一个 quorum Q2 中读出, 为了正确恢复数据, Q1 与 Q2 的交集中至少要包含 m 个服务器. 针对这种情形, Frolund 和 Merchant 等提出了门限 quorum 系统(m-quorum system)[2] 的概念, 门限 quorum 是这样一种集合系统, 其中任意两个子集的交集中至少包含 m 个元素. 他们利用纠删编码构造存储系统, 用门限 quorum 系统制订存储策略. 门限 quorum 适用于像门限方案和纠删编码这样利用部分份额恢复数据的系统, 然而如果 Q1 与 Q2 恰好相交于 m 个节点, 此时只要其中有一个发生故障便无法恢复数据. 因此, 这种系统不能全面地解决存储系统的可靠性问题. 许多系统对这一问题的处理方法是重新选择读 quorum Q2, 这样做势必会给系统带来新的负载.

本文主要研究门限 quorum 系统的容错问题. 在此我们提出另一种解决方法, 不用重新选择 Q2, 而是通过构造一种新型的 quorum 系统, 门限 Byzantine quorum 系统 TBQS(Threshold Byzantine Quorum System), 来提高数据服务的可靠性, 这种系统通过增强的交汇性来解决服务器故障问题, 避免了因重新选择读 quorum 而使服务效率降低. TBQS 可以用在基于纠删编码和门限方案的分布式存储中, 与现有 quorum 系统不同的是, 这种系统既可容忍服务器的 Byzantine 故障, 又可充分利用纠删编码和门限方案的优点, 保证数据的完整性及可靠性.

2　相关工作

Quorum 系统是一种利用冗余设计来提供容错的集合系统, 每个 quorum 由多个节点构成, 这些节点间具有特殊的冗余结构, 从而满足交汇性. 基于 quorum 的容错技术具有高可用性、并行性、灵活升级等优点. 对 quorum 的研究主要包括 quorum 系统的构造方法, 性能及应用.

Quorum 系统的构造方法可分为三类, 第一类是根据多数原则(majority consensus)[4], 简单地要求每个 quorum 中包含超过半数的节点从而满足交汇性, 此时构造的每个 quorum 的规模 $\geqslant \frac{N+1}{2}$, (N 为节点总数), 消息复杂度为 $O(N)$; 第二类是将服务器排列为一个虚拟的几何结构之后再进行构造, 主要有网格结构[5], 树形结构[6], 三角网格结构[7]等等, 所构造的 quorum 具有较小的规模; 第三类是利用代数方法(差分对)[8]或有限射影平面[9, 10]. 另外, 为了提高 quorum 系统的性能, 还可以将多种方法组合起来, 形成较复杂的构造方法, 如 crumbling walls[11].

文献[12]中总结了 quorum 系统及类似的数据结构如 coterie, bicoterie 等系统的定义, 并提出了利用组合构造 quorum 系统的方法.

Naor 和 Wool[13] 首先对 quorum 系统的负载、容量及可靠性进行定义并作了详细讨论, 他们提出的这些性能参数成为通用的衡量 quorum 系统性能的标准.

在应用方面, Quorum 系统通常用于构造互斥系统[9, 10, 14, 15]和存储管理系统. 人们提出了许多基于 quorum 的备份管理协议[4, 5, 8, 16~18]来解决存储系统的可靠性和一致性问题.

Malkhi 和 Reiter[1] 研究了 Byzantine quorum 系统, 他们对 quorum 系统的交汇性做了一定的限制, 提出了屏蔽 quorum 系统(MQS), 当服务器发生 Byzantine 故障时可以确保数据的可用性. 他们还提出了 dissemination quorum 系统(DQS) 的概念, 可以用于支持自验证数据的服务中.

在此基础上, 人们又根据不同的应用场合及可靠性要求, 对 quorum 系统的交汇性做了更多限制, 提出了其他一些可容忍 Byzantine 故障的系统[19].

Frolund 和 Merchant[2] 等提出了门限 quorum 系统的概念, 他们利用纠删编码构造存储系统, 存取结构为一个 quorum. 门限 quorum 适用于像秘密共享和纠删编码这样的利用部分份额恢复数据的系统, 然而对数据服务的可靠性考虑不足.

3　预备知识

3.1　故障模型

定义 1　(Quorum 系统)　设全集为 U, m 为正整数, 一个 quorum 系统 $Q=\{Q_1, Q_2, \cdots, Q_m\}$ 是 U 的子集的集合, 并且其中任意两个子集均相交, 每个 $Q_i \in Q$ 称为一个 quorum.

系统的可靠性可以用预期失效系统来形式化的表示, 我们采用文献[1]中对预期失效系统的定义:

定义 2　(预期失效系统)　一个预期失效系统 $B=\{B_1, B_2, \cdots, B_k\}$ 是 U 的子集的集合, 且对 $\forall B_i \in B$, 不

存在 $B_j \in B$, 使得 $B_i \subset B_j$. 从而在某一确定时刻, 必定存在某个 $B_i \in B$, 其中包含了所有的失效服务器.

预期失效系统是对所有可能发生的失效情形的一种推测. 一般来说, 服务器发生损毁的概率较低, 而攻击者的能力也总是有限的, 因而系统中总有一部分服务器处于正常工作状态. 在预期失效系统已知时, 可以设制适当的存储策略, 屏蔽失效服务器, 使正常的存储服务在未失效服务器中进行.

然而在实际中, 存储系统的普通用户不太可能预见到服务器何时发生损毁或被入侵, 要求用户知道哪些服务器失效是一种过强的假设. 因而在研究中通常会对预期失效系统作一些简化, 比如, 假定失效服务器个数的上限为 f (称为 f-threshold 模式), 或者假设错误个数是固定的.

3.2 分布式存储中数据的读写协议

设服务器集合 U 中的每个服务器上保留着数据 x 的副本, 客户对 x 进行读写操作, 客户在写入数据副本时为每个服务器分配一个时戳. 协议要求不同的客户选择不同的时戳, 客户 C 的公开时戳集合为 T_C, 各个客户的时戳集合不相交. 读写协议如下.

写: 客户 C 要写入数据 v 时, 首先选定写 quorum Q_1, 向其中所有的服务器发出请求, 得到由这些服务器传来的时戳集合 $A=\{\langle t_u\rangle\}_{u\in Q_1}$, 选择时间 $t \in T_C$, t 大于 A 中的最大值, 并且大于 C 以往所选择的任何一个时戳值. C 传送更新后的值 $\langle v', t\rangle$ 到 Q_1, 并从各个服务器处得到应答信息.

读: 客户 C 要读取 x 的值时, 选定读 quorum Q_2, 向 Q_2 中的服务器发送请求, 得到服务器传来的数据/时戳集合 $A=\{\langle v_u, t_u\rangle\}_{u\in Q_2}$, 选择其中具有最大时戳 t_m 的服务器传来的数据 v_m 作为有效数据.

3.3 m-quorum 系统与 MQS

纠删编码与门限方案在对数据的处理上有某种相似之处, 它们均作用于一个多节点系统中, 利用数据分离算法将数据分散保存, 而恢复数据时, 利用部分节点上保存的数据运行恢复算法来得到原始数据.

当使用 3.2 节中的读写协议时, 为了保证数据的一致性, 要求读 quorum Q_1 与写 quorum Q_2 至少相交于 m 个节点, 即 $|Q_1| \cap |Q_2| \geqslant m$. 如果采用 f-threshold 故障模式, 并且所有服务器发生故障的概率均相同, 则预期失效系统 B 可定义为 U 中所有包含 f 个节点的集合. 因而 m-quorum 系统[2]的定义如下:

定义 **3** 一个 m-quorum 系统 Q 满足以下两个性质:

M-Consistency: $\forall Q_1, Q_2 \in Q, |Q_1 \cap Q_2| \geqslant m$;

M-Availability: $\forall B_i \in B$, 存在 $Q_j \in Q$, $B_i \cap Q_j = \Phi$.

文献[2]中讨论了 m-quorum 系统的存在性.

引理 **1**[2] 当且仅当 $n \geqslant 2f + m$ 时, 存在一个 m-quorum 系统.

文献[1]中定义了能容忍 Byzantine 故障的 quorum 系统, 称为屏蔽 quorum 系统 MQS (masking quorum systems). 给定预期失效系统 B, 关于 B 的 MQS 定义为:

定义 **4** (屏蔽 Quorum 系统 MQS) 如果以下条件成立, 则 Q 是一个关于预期失效系统 B 的 MQS:

M-consistency: $\forall Q_1, Q_2 \in Q, \forall B_1, B_2 \in B$, 有 $Q_1 \cap Q_2 \setminus B_1 \not\subset B_2$;

M-availability: $\forall B \in B, \exists Q \in Q$, 使得 $B \cap Q = \Phi$

引理 **2**[1] 令 B 为一个预期失效系统, 则当且仅当 $Q=\{U \setminus B_i : B_i \in B\}$ 是一个 MQS 时, 才存在关于 B 的 MQS.

4 门限 Byzantine Quorum 系统

4.1 定义

TBQS 的定义如下:

定义 **5** 如果以下条件成立, 则 Q 是关于预期失效系统 B 的 TBQS:

(1) TB-consistency: $\forall Q_1, Q_2 \in Q, \forall B_i \in B$, 有 $|(Q_1 \cap Q_2) \setminus B_i| \geqslant m$;

(2) TB-availability: $\forall B_i \in B, \exists Q_j \in Q$, 使得 $B_i \cap Q_j = \Phi$.

TBQS 的定义综合考虑了效率、安全性及可靠性. 其中条件(1)表示从任意两个 quorum 的交集中除去故障节点外仍有至少 m 个节点, 从而可以利用纠删编码或门限方案来保存数据, 而当系统中有故障节点时能保证读写 quorum 中仍存在足够的交集来恢复数据. 条件(2)表示在任何一个时刻系统中除去故障节点外仍至少存在一个可用的 quorum, 保持数据的一致性.

TBQS 的效率和安全性体现在它可以应用在使用纠删编码和门限方案的存储系统中, 与备份系统相比, 大大减小了单个服务器上保存的数据量, 并且数据的安全性也随之提高. TBQS 的可靠性则依赖于其本身的定义和预期故障系统.

4.2 TBQS 的存在性

设 n 为总节点数, m 为读 quorum 中包含的最小节点数, f 为失效节点上限. 在讨论 TBQS 的存在性时, 我们假设预期失效系统由 f 个节点构成的子集的集合构成, 即故障节点的数量上限为 f.

定理 **1** 设服务器集合为 U, $|U| = n$, 预期失效系统为 $B=\{B_i \in 2^U, |B_i| = f\}$, 则当且仅当 $Q=\{Q_i \in 2^U | |Q_i| = n-f\}$ 是一个 TBQS 时, U 中存在关于 B 的 TBQS.

证明: 充分性显然.

必要性

假设 Q' 为 U 中关于 B 的 TBQS, 而 Q 不是 TBQS, 因为对于 $\forall B_i \in B, |B_i| = f$, 故 Q 显然满足 TBQS 的条件 2, 因而必不满足条件 1, 从而存在 $Q_1, Q_2 \in Q, B \in B$, 使得 $|(Q_1 \cap Q_2) \setminus B| < m$.

令 $B_1 = U \setminus Q_1, B_2 = U \setminus Q_2$, 则 $|B_1| = |B_2| = f$, 故 $B_1 \in B, B_2 \in B$. 因而由条件 2, 存在 $Q_1', Q_2' = Q'$, 使得 $Q_1' \cap B_1 = \Phi, Q_2' \cap B_2 = \Phi$, 从而有 $Q_1' \subseteq Q_1, Q_2' \subseteq Q_2$, 故 $Q_1' \cap Q_2' \subseteq Q_1 \cap Q_2$. 因此 $\left|\left(Q_1' \cap Q_2'\right) \setminus B\right| \leqslant |(Q_1 \cap Q_2) \setminus B| < m$, 这与 Q' 是 TBQS 矛盾.

定理 2　如果 $n \geqslant 3f + m$, $B = \{B_i \in 2^U \mid |B_i| \leqslant f\}$, 则 $Q = \{Q_i \in 2^U \mid |Q_i| = n - f\}$ 是 U 上关于预期失效系统 B 的 TBQS.

证明: 显然 Q 满足条件 2, 因此只须证明满足条件 1.

对 $\forall Q_1, Q_2 \in Q, \forall B_i \in B$,

$$|(Q_1 \cap Q_2) \setminus B_i| \geqslant |Q_1 \cap Q_2| - |B_i| \geqslant |Q_1 \cap Q_2| - f$$
$$= |Q_1| + |Q_2| - |Q_1 \cup Q_2| - f \geqslant 2(n - f) - n - f$$
$$= n - 3f \geqslant m$$

从而 Q 也满足条件 1, 因而是一个 TBQS.

定理 3　如果 $B = \{B_i \in 2^U \mid |B_i| \leqslant f\}$, $Q = \{Q_i \in 2^U \mid |Q_i| = n - f\}$ 是关于 B 的 TBQS, 且 $m \geqslant f$, 则 $n \geqslant 3f + m$.

证明: 如果 Q 为 TBQS, 则对 $\forall Q_1, Q_2 \in Q$, 因为 $|Q_1 \cap Q_2 \neq \Phi|$, 而 $|Q_1| = |Q_2| = n - f$, 所以 $n \geqslant 2f$. 故必存在 $Q_1, Q_2 \in Q$, 使得 $|Q_1 \cup Q_2| = U$. 因为 $Q_2 = (U \setminus Q_1) \cup (Q_1 \cap Q_2)$, 而 $(U \setminus Q_1) \cap (Q_1 \cap Q_2) = \Phi$, 所以 $|Q_2| = |U \setminus Q_1| + |Q_1 \cap Q_2|$, 故

$$|Q_1 \cap Q_2| = |Q_2| - |U \setminus Q_1| = n - f - f = n - 2f$$

因为 $|Q_1 \cap Q_2| \geqslant m \geqslant f$, 故必存在 $B_i \in B, |B_i| = f$, 且 $B_i \subseteq Q_1 \cap Q_2$, 所以 $|(Q_1 \cap Q_2) \setminus B_i| = n - 2f - f = n - 3f \geqslant m$, 即 $n \geqslant 3f + m$.

以上三个定理讨论了 TBQS 的存在性, 由定理二、三知 TBQS 存在的充要条件是: $m \geqslant f$, 且 $n \geqslant 3f + m$, 或者 $n \geqslant 4f$. 因而从容错的角度看, TBQS 最多可以容忍所有节点中有 1/4 产生 Byzantine 故障.

5　TBQS 的构造及效率分析

由文献[1]中的 f-masking 系统和 quorum 的网格构造出发, 我们构造了以下两种 TBQS.

5.1　门限 f-masking 系统

令 $n \geqslant 3f + m$, $m \geqslant f$, $B = \{B_i \in 2^U \mid |B_i| = f\}$, 则 $Q = \left\{ Q_i \subset U \mid |Q_i| = \left\lceil \frac{n + m + f}{2} \right\rceil \right\}$ 是 U 上关于 B 的 TBQS.

TB-consistency: $\forall Q_1, Q_2 \in Q, |Q_1 \cap Q_2| \geqslant m + f$, 因而对于 $\forall B_i \in B, |(Q_1 \cap Q_2) \setminus B_i| \geqslant m + f - f = m$.

TB-availability: 因为 $n \geqslant 3f + m$, 故 $\left\lceil \frac{n + m + f}{2} \right\rceil \leqslant n - f$, 所以对 $\forall B_i \in B$, 存在 $Q_j \in Q$, 使得 $B_i \cap Q_j = \Phi$.

5.2　网格 TBQS

令 $B = \{B_i \in 2^U \mid |B_i| = f\}$, 假设 $n = k^2, k \geqslant 2f + m$, 将 n 个节点排列为 $k \times k$ 的网格, 令 R_i 表示第 i 行, C_j 表示第 j 列, 则 $Q = \{C_j \cup \bigcup_{i \in I} R_i \mid \{j\} \subseteq \{1, 2, \cdots, k\}, |I| = m + f\}$ 是 U 上关于 B 的 TBQS. 其中每个子集的选取方法是在网格中任取一列, 再任取 $m + f$ 行, 两者的并即构成一个 quorum.

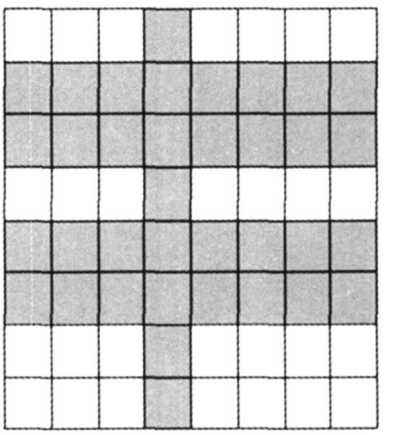

图 1　网格 TBQS, 阴影部分为一个 quorum. 这里 $f=2$, $m=2$, $k=8$

TB-consistency: $\forall Q_1, Q_2 \in Q, |Q_1 \cap Q_2| \geqslant m + f$, 故对于 $\forall B_i \in B, |(Q_1 \cap Q_2) \setminus B| \geqslant m$;

TB-availability: 因为 $k \geqslant 2f + m$, 所以对 $\forall B_i \in B$, 存在 $Q_j \in Q$, 使得 $B_i \cap Q_j = \Phi$.

图 1 给出了网格构造的一个例子.

5.3　效率分析

Quorum 系统 Q 的规模定义为其中最小子集 Q_i 中包含的结点个数, 记为 $\mathrm{Size}(Q) = |Q_i|$, quorum 系统的规模越大, 其内部结点之间的通信费用就越高, 为获得数据一致性的同步时间就越长. 相反, 规模越小, 其内部结点对相同数据的冗余量就越小, 含有效数据的结点比例就会越低.

负载是衡量 quorum 系统性能的重要参数. 在分布式系统中, 访问策略表示为对每个 quorum 的访问频率, 系统采取的访问策略为每个元素或节点引入了负载, 即对该节点所在的所有 quorum 的访问频率之和. 对于一个给定的 quorum 系统 Q, 以及所有的访问策略, 负载 $L(Q)$ 是指其中最"忙"元素的最小负载. 负载越小, 则每个元素被访问的次数也越小, 从而有更多的时间和资源来处理其他事务.

Naor 和 Wool 研究了与 quorum 系统的负载有关的一系列问题[2].

给定 quorum 系统 $Q = \{Q_1, Q_2, \cdots, Q_m\}$, 设读写数据时选择子集 Q_i 的概率为 $W(Q_i)$, 则访问策略 w 表示 Q 为关于中元素的概率分布, 即 $\sum_{i=1}^{m} w(Q_i) = 1$, 负载的

定义如下:

定义 **5** 给定服务器集合 U, quorum 系统 Q 及访问策略 w, 则 w 对某个特定的 $u \in U$ 诱导的负载为 $l_w(u) = \sum_{u \in Q_i} w(Q_i)$, w 对 quorum 系统 Q 诱导的负载为 $L_w(Q) = \max_{u \in U}\{l_w(u)\}$, 而 Q 的系统负载为 $L(Q) = \min_w\{L_w(Q)\}$.

引理 **3**[2] 如果 Q 是在 n 个元素的全集上的 quorum 系统, 则 $L(Q) \geqslant \max\left\{\frac{1}{C(Q)}, \frac{C(Q)}{n}\right\}$, 因而 $L(Q) \geqslant \frac{1}{\sqrt{n}}$. 其中 C(Q) 为 Q 的规模.

与 BQS 相比, 在容忍相同数量故障节点的前提下, TBQS 要求更强的交汇性, 这意味着 quorum 规模的增加, 或者数据访问时的通信量的增大.

5.1 节中构造的门限 f-masking 系统规模为 $\left\lceil \frac{n+m+f}{2} \right\rceil$, 而 5.2 节中的网格 TBQS 规模为 $(m+f+1)\sqrt{n}-(m+f)$, 均大于文献[1]中具有相同参数的 BQS 的规模.

根据引理 3, 当采用均衡访问策略, 即访问每个 quorum 的概率均相同时, 门限 f-masking 系统的负载为 $\frac{1}{n}\left\lceil \frac{n+m+f}{2} \right\rceil$, 网格 TBQS 的负载为 $\frac{(m+f+1)k-(m+f)}{n}$. 均小于文献[1]中类似 BQS 的负载.

与传统的基于复制的 quorum 系统相比, 由于 TBQS 是针对纠删编码或者门限方案而设计的, 因而不可避免地具有更大的规模, 但是负载却有所减小. 同时, 采用纠删编码或者门限方案代替复制可以解决复制带来的效率和安全问题, 提高了数据服务的可靠性和安全性. 因而 TBQS 在利用这两种方案优点的同时, 也体现了效率与可靠性的折衷, 是一种实用性很强的存储策略.

6 应用

由 3.2 节中的读写协议出发, 我们设计了如下基于 TBQS 的数据读写协议.

设系统中共有 n 个存储节点, Q 为 n 个节点的全集上的规模为 r 的 TBQS. 采用的数据分离算法为 DIS(v), DIS() 是现有的纠删编码或秘密共享算法, 其输入为数据 v, 输出是一个 r 维向量 $\boldsymbol{V} = (v_1, \cdots, v_r)$. 假设恢复数据时至少需要 m 个份额, $m < r$, 则相应的数据恢复算法为 RET(), 其输入为 m 维向量, 在此用 Shareset 表示, 输出为 $\boldsymbol{v}$.

设客户 C 的时间戳为 τ, 每个服务器 P_i 上保存着本地副本 v_i 和本地时间 τ_i.

图 2、图 3 分别给出了基于 TBQS 的读写协议, 与 3.2 节中协议不同之处在于, 我们的协议在写入数据时加入了数据分离的过程, 在读取数据时客户选择读 quorum, 从中收集份额, 收集到 m 个份额后运行恢复算法. 使用 TBQS 的优点在于, 如果收集到的份额中有错误的, 或者某个服务器没有返回数据, 仍不影响用户运行恢复算法. 而现有的其他协议则需要重新选择读 quorum, 否则无法恢复数据.

```
Protocol Write for data v:
    τ++;                                          //Client C
    V:= DIS(v);
  select a quorum Q= (Q1, Q2, …, Qr);
    send(write, vi, τ) to Qi;
    wait for ACKs;
  upon receiving(write, vi, τ) from C             //Server Qi
    if τ> τi then (vi, τi) ← (v, τ);
    return ACK;
  upon receiving ACKs from all servers in Q       //Client C
    return abort;
  End.
```

图 2 写入协议

```
Protocol Read for data v:
  t:= 0;
  Shareset:= NULL;
  while (t< m)
    select a quorum Q;                            //Client C
    broadcast (read, v) to all servers in Q;
      maxtimestamp:= 0;
    upon receiving(read, v)                       //Server Qi
      send(vi, τi) to C;
    upon receiving all (vi, τi) from Server Qi    //Client C
      find max(τi);
      maxtimestamp:= τi
      for j= 1 to r do
        if τj= maxtimestamp
          t++;
            add vj to Shareset;
  v:= RET(Shareset);                              //Client C
  End.
```

图 3 读取协议

7 结论

存储系统的容错性能是分布式存储设计中的热点问题, 利用纠删编码和门限方案分离数据是提高存储系统安全性和可靠性的重要措施, 本文提出了一种新型的 quorum 系统 TBQS, 利用 TBQS 与这两种数据分离算法相结合, 可以构造对可靠性要求较高的存储系统. 我们证明了当系统中故障节点比例不大于 1/4 时, 用

TBQS 设计存储策略可以实现容错与无间断服务. TBQS 的使用既能充分利用纠删编码与门限方案的优点, 又能提供容错. 所构造的两类 TBQS 体现了效率与可靠性的折衷, 具有较高实用价值.

参考文献:

[1] D Malkhi, M Reiter, Byzantine quorum systems[J]. Distributed Computing, 1998, 11(4): 203– 213.

[2] S Frolund, A Merchant, U Saito, S Spence, A Veitch. A decentralized algorithm for erasure-coded virtual disks[R]. Technical Report HPL-2004-46, HP Labs, 2004.

[3] 郭渊博, 马建峰. 异步及不可靠链路环境下的先应式秘密共享[J]. 电子学报, 2004, 32(3): 399– 403.
Guo Yuanbo, Ma Jianfeng, Proactive secret sharing in asynchronous networks with unreliable links[J]. Acta Electronica Sinica, 2004, 32(3). 399– 403. (in Chinese)

[4] D K Gifford. Weighted voting for replicated data[A]. Proc of 7th Symp on Operating Systems Principles[C]. New York: ACM Press, 1979. 150– 162.

[5] S Y Cheung, M H Ammar, M Ahamad. The grid protocol: A high performance scheme for maintaining replicated data[J]. IEEE Trans Knowledge and Data Eng, 1992, 4(6): 582– 592.

[6] D Agrawal, A E Abbadi. The generalized tree quorum protocol: An efficient approach for managing replicated data (corrigenda) [J]. ACMTDS: ACM Transactions on Database Systems, 18, 1992, 17(4): 689– 717.

[7] C H Cho, J T Wang. Triangular grid protocol: an efficient scheme for replica control with uniform access quorums[A]. Proc Euro-Par' 96 Parallel Processing Conf[C]. Lyon, France: Springer, 1996. 843– 851.

[8] C M Lin, G M Chiu. A new quorum-based scheme for managing replicated data in distributed systems[J]. IEEE Transactions on Computers, 2002, 51(12): 1442– 1447.

[9] W S Luk, T T Wong. Two new quorum based algorithms for distributed mutual exclusion[A]. Proc Int' l Conf Distributed Computing Systems[C]. New York: ACM Press, 1997. 100– 106.

[10] M Maekawa. A √n algorithm for mutual exclusion in decentralized systems[J]. ACM Transactions on Computer Systems, 1985, 3(2): 145– 159.

[11] D Peleg, A Wool. Crumbling walls: A class of practical and efficient quorum systems[J]. Distributed Systems, 1997, 10(2): 120– 129.

[12] M L Neilsen, M Mizuno. A general method to define quorums [A]. Proceedings of the 12th International Conference[C]. YoKohama, Japan: IEEE Computer Society Press, 1992. 657– 664.

[13] M Naor, A Wool. The load, capacity and availability of quorum systems[A]. SIAM Journal on Computing, 1998, 27(2): 423– 447.

[14] Y J Joung. Quorum-based algorithms for group mutual exclusion[J]. IEEE Transactions, 2003, 14(5): 463– 476.

[15] Harada T, Yamashita M. Transversal merge operation: A non-dominated coterie construction method for distributed mutual exclusion[J]. IEEE Transactions, 2005, 16(2): 183– 192.

[16] M P Herlihy. Replication methods for abstract data types[D]. Massachusetts Institute of Technololgy, Cambridge, MA, 1984.

[17] K Tanaka, M Takizawa. Replica management in object-based systems[A]. Proceedings of the 15th International Conference on Information Networking [C]. New York: IEEE Press, 2001. 367– 372.

[18] Kong L, Manohar D J, A Subbiah, Sun M. Agile store: experience with quorum-based data replication techniques for adaptive Byzantine fault tolerance[A]. SRDS 2005, 24th IEEE Symposium[C]. New York: IEEE Press, 2005. 143– 154.

[19] J P Martin, L Alvisi, M Dahlin. Small byzantine quorum systems[A]. Proceedings International Conference on Dependable Systems and Networks[C]. New York: IEEE Press, 2002, 374– 383.

作者简介:

张　薇　女, 1976 年生于陕西西安, 讲师, 西安电子科技大学博士生. 研究方向为密码学、分布式存储系统、信息系统可生存性.
E-mail: zhaangweei@yeah. net

马建峰　男, 1963 年生于陕西临潼, 西安电子科技大学计算机学院院长, 教授. 研究方向为信道编码、信息安全及信息系统可生存性.