doi:10.3969/j.issn.1000-436x.2013.07.013

复本存储机制的效率研究

罗香玉，汪芸，陈笑梅，袁飞飞，李聪
(东南大学 计算机科学与工程学院 计算机网络和信息集成教育部重点实验室，江苏 南京 211189)

摘 要：建立独立于具体算法的复本存储模型，识别影响效率的关键参数和无关参数；分析系统规模增长对效率的影响，通过实验得出空间效率与 I/O 效率的乘积同系统节点数量近似成反比这一结论，并从理论上解释结论背后的成因。研究结果可用于指导复本存储机制的工程应用，为空间效率和 I/O 效率之间的权衡以及大规模存储系统 I/O 性能的预测提供理论依据。

关键词：存储系统；复本存储；负载均衡；I/O 效率；空间效率
中图分类号：TP393 **文献标识码：**B **文章编号：**1000-436X(2013)07-0111-13

Research on the efficiency of replication-based storage mechanism

LUO Xiang-yu, WANG Yun, CHEN Xiao-mei, YUAN Fei-fei, LI Cong
(CNII, MoE, School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: An algorithm-independent description model was established for the ism, and both efficiency-related and efficiency-independent parameters were identified. Besides, the impact of system scale on efficiency was investigated, and an important observation was obtained that the product of the two utilization ratios (i.e., the utilization ratio of disk space and that of I/O bandwidth) is approximately inversely proportional to the number of nodes contained in the system. The observation was verified through experimental resu reason behind was revealed through theoretical analysis. The study provides guidelines for the application of replication-based storage mechanism in engineering, and lays a theoretical foundation for the trade-off between the two utilization ratios and for the prediction of I/O performance in large-scale storage systems.

Key words: storage system; replication-based storage; load balance; I/O efficiency; space efficiency

1 引言

随着云计算的兴起和大数据时代的降临，大规模存储系统在 IT 基础设施中的重要性日益突出，同时，其在存储容量、I/O 性能及系统成本等方面所面临的挑战也愈加严峻。国际数据公司的一项研究报告指出，2007 年全球信息总量首次超过存储介质的总容量，并且信息总量的增长速度(年增长率为 57%)远高于存储介质总容量的增长速度(年增长率为 35%)[1]，存储容量压力巨大。作为低成本的主流存储设备，机械硬盘的 I/O 带宽增长缓慢，极大地限制了存储系统 I/O 性能的提升。存储系统的成本危机日益凸显，单就能耗成本而言，2011 年全美 3%的电能用于供给数据中心的计算、存储及网络设备和其他相关设施[2]。随着存储系统规模的持续增长，其成本已在 IT 基础设施总成本中占据相当高的比重[3]。降低存储成本已被 Gartner 列为降低 IT 开支的 10 种关键行为之一。

在存储容量、I/O 性能及系统成本等诸多挑战面前，研究存储系统的效率问题具有重要意义。效率表征系统对资源（包括存储空间资源和 I/O 带宽资源）的有效利用程度。效率的提高可在给定资源条件下增加系统的服务能力（如存储更多的数据、服务更多的用户访问请求），或在满足给定服务需

收稿日期：2012-11-05；修回日期：2013-06-07
基金项目：国家高技术研究发展计划（“863”计划）基金资助项目(2011AA040502)；国家自然科学基金资助项目(60973122)
Foundation Items: The National High Technology Research and Development of China(863 Program) (2011AA040502); The National Natural Science Foundation of China (60973122)

求条件下节约资源成本。此外，数据存储量和访问量的持续增长要求系统不断扩容，系统规模增长对效率的影响也是需要研究的重要问题。

存储系统的效率包括空间效率和I/O效率。影响空间效率的主要因素是冗余存储。常见的冗余存储方式包括复本和纠删码[4]。空间效率采用冗余度的倒数表示。冗余度越高则空间效率越低。影响I/O效率的主要因素是节点间所承担I/O负载的非均衡性。这种非均衡性可由文件访问热度分布的偏斜性引起[5]。偏斜性是指热点文件和冷门文件的访问热度差异。常见的文件访问热度分布如Zipf-like[6]以及泛化的Zipf-like[7]。此外，文件访问热度往往是动态变化，先前的热点文件可能变为冷门文件，反之亦然。即使某时刻节点间可实现I/O负载均衡，后续仍可能因为文件访问热度的变化而不再均衡。在I/O负载不均衡系统中，负载最重的节点将成为系统I/O性能的瓶颈点。节点间I/O负载不均程度越严重，则I/O效率也越低。

大规模存储系统通常采用复本存储机制。复本存储机制主要包括复本放置算法和请求分发算法。前者的核心任务是确定各文件的复本数量和复本位置。后者在访问请求到达时，根据所请求文件各复本所在节点的负载状态，为访问请求指定最佳服务节点。在复本存储机制下，各个文件均在多个节点中存有复本。复本的存在一方面增加了文件访问请求分发时的灵活性，从而有效缓解文件访问热度偏斜性和动态性对I/O负载均衡的影响，提高了系统I/O的效率，另一方面造成数据冗余度增加，从而导致空间效率下降。此外，复本存储机制还是提高数据可靠性的重要手段，相关方面的研究也非常重要，在本文中主要关注其对存储空间资源和I/O带宽资源的使用效率。

复本存储机制获得了广泛的研究，根据复本数量和位置的可变性大致分为2类：静态复本存储机制和动态复本存储机制。静态复本存储机制为各文件存储相同数量的复本，并且文件的复本数量和位置均不随时间变化；复本放置完毕后，仅依靠复本间的请求分发实现I/O负载均衡。该类存储机制的相关工作包括Kinesis[5]、CFS[8]、Chain[9]等。较之同类机制的其他算法，Kinesis算法[5]在选择复本位置时具备较高灵活性，并且采用基于“multiple-choice”的请求分发策略，因而拥有较强的I/O负载均衡能力。动态复本存储机制对节点负载状态及文件访问热度进行实时监测，并适时调整部分文件的复本数量或存储位置。除依靠复本间的请求分发之外，动态复本存储机制还可通过改变复本数量和位置的方式保证I/O负载均衡。典型算法如RMAS[10]、INDSC[11]、CDRM[12]、CAGW_PD[13]等。然而，动态复本存储机制增加了工程实现的复杂性[14]；复本位置调整会消耗系统I/O带宽等资源，影响访问请求的及时处理；并且其对I/O负载均衡的改善效果依赖于文件访问热度的准确预测，实际效果不稳定。

在获得广泛学术研究的同时，复本存储机制存在大量工程应用。比如，谷歌分布式文件系统GFS[15]、淘宝分布式文件系统TFS[16]、开源分布式文件系统HDFS[17]等均采用复本存储机制。为降低工程实现的复杂性，上述系统通常为各文件指定相同的复本数量，且各文件的复本数量和位置均静态不变[18]。尽管HDFS[17]的设计兼容动态复本存储机制（体现在rebalancer模块），但在大多数实际应用中，HDFS[17]主要以静态复本存储机制工作，各文件默认的复本数量为3[19]。基于静态复本存储机制在工程应用中的普遍性，本文主要分析该类机制的效率。

现有研究在对所提出算法进行分析时，通常采用如下方式：在一定节点数量及复本数量条件下测试系统的I/O性能。用以衡量I/O性能的指标主要包括请求响应延迟、I/O吞吐量、节点I/O负载均衡度等。但是，现有研究尚未形成复本存储机制效率的规律性认识，如空间效率和I/O效率之间的关系、系统规模增长对效率的影响等。这种规律性认识的缺乏造成工程上复本方案选择的盲目性，使得人们难以预见系统在方案采纳后的I/O性能，更难以预见系统在未来规模扩展后的I/O性能。

本文通过建立独立于具体算法的复本存储模型和相关理论分析，得出有关复本存储机制效率的一般性结论；同时，以典型算法Kinesis[5]为例，对复本存储机制的效率进行实验分析。主要贡献包括以下3个方面。

1) 识别影响效率的关键参数和无关参数，指出复本存储机制的效率与系统所采用节点的硬件配置（包括存储容量和I/O带宽）无关，但与节点数量密切相关。

2) 从理论上证明当文件访问热度服从偏斜度大于1的Zipf-like分布[6]时，若保持空间效率不变，则系统的I/O吞吐量存在上限（即使节点数量和文件数

量无限制地增加)。该结论解释了空间效率和 I/O 效率乘积随节点数量增加而不断下降的原因。

3) 通过实验得出空间效率和 I/O 效率的乘积同系统中节点的数量近似成反比这一结论。该结论可为 2 种效率间的权衡提供参考依据，并可用于预测大规模存储系统的 I/O 性能（本文选择 I/O 吞吐量作为 I/O 性能的衡量指标)。

2　复本存储模型及假设条件

2.1　复本存储模型

工程上广泛采用的复本存储机制通常为各文件静态指定相同的复本数量[18]。本文所研究的复本存储模型主要针对此类机制。所建立模型是对此类存储机制共性的抽象，而独立于所采用的具体复本放置及请求分发算法。

除复本存储机制外，缓存机制也是提高 I/O 性能的重要手段。在 Web 和数据库查询等强调请求响应延迟的应用中，缓存机制发挥了重要作用。然而，在以大文件（如视频文件）为主、强调 I/O 吞吐量的应用中，多复本依然是提高 I/O 性能的主要技术。一方面，当前磁盘的 I/O 带宽与网络带宽比较相近，或者通过一个节点连接多个磁盘的方式可获得与网络带宽相近的 I/O 带宽[5]，因此，即使采用缓存机制，I/O 吞吐量因受限于网络带宽而难以获得显著提升。另一方面，为保证数据存储的可靠性，即使采用了缓存机制，复本存储机制仍然不可缺少；若可单独通过复本存储机制保证较高的 I/O 性能，则可减少缓存机制产生的额外成本[20]。基于上述原因，本文的复本存储模型没有考虑缓存机制。

本文所建立的复本存储模型分为系统资源、存储及访问需求、复本方案 3 个部分。其中，前两部分是复本放置及请求分发算法的输入，复本方案则是算法的输出。

系统资源　存储系统 M 由 n 个节点构成，即 $M=\{N_1,\cdots,N_n\}$；节点 N_i 用二元组 $<c_i,b_i>$ 表征，$i=1,\cdots,n$，其中，c_i 和 b_i 分别表示节点 N_i 的存储容量和 I/O 带宽。此外，以 C 表示系统的存储空间总容量，以 B 表示系统的 I/O 带宽总量，则

$$C=\sum_{i=1}^{n}c_i$$

$$B=\sum_{i=1}^{n}b_i$$

存储及访问需求　待存储文件集合 F 包含 m 个文件，即 $F=\{f_1,\cdots,f_m\}$；文件 f_j 用二元组 $<s_j,\ ?_j(t)>$ 表征，$j=1,\cdots,m$，其中，s_j 和 $?_j(t)$ 分别表示文件 f_j 的大小和访问请求到达率，s_j 和 $?_j(t)$ 的乘积称为文件 f_j 所产生的 I/O 负载率（即单位时间所产生的 I/O 负载量)。

以 S 表示全部文件大小之和，则

$$S=\sum_{j=1}^{m}s_j$$

以 $W(t)$ 表示全部文件所产生的 I/O 负载率之和，以 $?(t)$ 表示全部文件的访问请求到达率之和，以 $p_j(t)$ 表示文件 f_j 被访问比例，则

$$W(t)=\sum_{j=1}^{m}s_j l_{\ j}(t)$$

$$l\ (t)=\sum_{j=1}^{m}l_{\ j}(t)$$

$$p_j(t)=l_{\ j}(t)/l\ (t)$$

由于各文件访问热度并非静止不变，$?_j(t)$、$W(t)$、$?(t)$ 及 $p_j(t)$ 均是时间 t 的函数。

复本方案　复本方案包含各文件的复本数量、复本位置以及各文件访问请求在多个复本节点间的分发比例，可通过复本数量 r、复本放置矩阵 $\boldsymbol{D}_{n\times m}$ 及请求分发矩阵 $\boldsymbol{E}_{n\times m}(t)$ 唯一确定。

在本文所研究的复本存储模型下，复本数量和位置均静态确定，因此，复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$ 不随时间变化。该矩阵的元素 d_{ij} 只有 0 和 1 两种取值。若 $d_{ij}=0$，则表示节点 N_i 不存有文件 f_j 的复本；否则，表示节点 N_i 存有文件 f_j 的复本。

由于访问请求分发至各复本节点的比例随文件访问热度的变化而动态调整，因此，矩阵 $\boldsymbol{E}_{n\times m}(t)$ 是关于时间 t 的变量。该矩阵的元素 $e_{ij}(t)$ 满足 $0\le e_{ij}(t)\le 1$（$i=1,\cdots,n;\ j=1,\cdots,m$），用以表示访问文件 f_j 的请求被分发至节点 N_i 的比例。$e_{ij}(t)>0$ 的必要条件是节点 N_i 存有文件 f_j 的复本，即 $d_{ij}=1$。

复本方案 $<r,\ \boldsymbol{D}_{n\times m},\ \boldsymbol{E}_{n\times m}(t)>$ 需满足如下 5 个约束条件。

$$\sum_{i=1}^{n}d_{i\,j}=r(1\le j\le m)\tag{1}$$

$$\sum_{j=1}^{m}d_{i\,j}s_j\le c_i(1\le i\le n)\tag{2}$$

$$e_{i\,j}(t)\le d_{ij}(1\le i\le n,1\le j\le m)\tag{3}$$

$$\sum_{i=1}^{n} e_{ij}(t) = 1 (1 \le j \le m) \tag{4}$$

$$\sum_{j=1}^{m} e_{ij}(t) s_j l_j(t) \le b_i (1 \le i \le n) \tag{5}$$

式(1)约束各文件的复本数量为 r；式(2)约束各节点所存储的数据量不超过其存储容量；式(3)约束访问请求仅能分发至存有文件复本的节点；式(4)约束各文件的访问请求被分发至复本节点的比例之和为 1；式(5)约束分发至各节点的 I/O 负载率不超过其 I/O 带宽。

2.2 假设条件

本文假设节点同构（即 $c_i=c$, $b_i=b$, $i=1,\cdots,n$，该假设可在一些集群存储系统中成立）、文件大小相同（即 $s_j=s$, $j=1,\cdots,m$，该假设可通过分块存储保证，如 GFS[15]的文件分块大小为 64 MB），重点关注文件访问热度分布和节点数量对效率的影响。

文件访问热度分布属于文件访问特征建模研究的一部分。文件访问特征的建模非常复杂，目前仍处于不断研究之中[21~23]。其中，文件访问热度分布的偏斜性特征对存储系统 I/O 性能的影响受到研究者的普遍重视[12,19]。Zipf-like 分布[6]是描述偏斜性的重要工具，在多种不同类别复杂系统的研究中产生了深远影响[24]，并且已被实验研究证实可用于描述 Web 文件和视频文件的访问热度分布[25,26]，即访问热度排名为 x 的文件在全体文件中被访问的比例 $G(x)$满足如下函数关系式。

$$G(x) = (1/x^a)/(\sum_{k=1}^{m} 1/k^a)$$

其中，a 是大于 0 的常数，称为偏斜度。偏斜度越大，表示热点文件和冷门文件的访问热度差异越显著。

本文采用 Zipf-like 分布[6]描述文件访问热度的偏斜性。文件访问热度分布的动态性体现为各文件访问热度排名的变化以及偏斜度 a 的变化。

此外，本文假定系统中的文件访问符合“Write-once-read-many”，即“一次写多次读”模式或称 Worm 模式。

3 复本存储机制效率的理论分析

复本放置及请求分发算法所产生的复本方案直接影响系统对存储空间资源和 I/O 带宽资源的使用效率。在系统资源确定时，复本方案决定系统可容纳的数据量和所能提供的 I/O 吞吐量；在存储及访问需求确定时，复本方案决定系统所需投入的资源成本。

然而，复本方案的解空间为指数级别，无法穷尽所有可能性。本节对复本存储机制效率的理论分析结论独立于具体的复本方案，从而独立于用于产生复本方案的复本放置及请求分发算法，因此具有较普遍的意义。

3.1 效率的计算

本文所关注的效率包括空间效率和 I/O 效率。在本文所定义的复本存储模型下，复本方案给定后，复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$ 便保持不变，但请求分发矩阵 $\boldsymbol{E}_{n\times m}(t)$可根据文件访问热度分布的变化进行动态调整。

空间效率　给定复本方案下，空间效率 e 用复本数量 r 的倒数表示，即 $e=1/r$。

数据容量 $S_{\max}$：满足式(1)和式(2)这 2 个约束条件下，全部文件大小之和 S 的最大值称为系统的数据容量。一般情况下，对相同存储空间总容量 C 而言，e 越大（即 r 越小），则 $S_{\max}$ 越大。

I/O 效率　给定复本方案下，I/O 效率 $?$ 用系统 I/O 吞吐量的期望 $\mathrm{E}(W_{\max})$与 I/O 带宽总量 B 的比值表示，即 $?=\mathrm{E}(W_{\max})/B$。在 2.2 节的假设条件下，$B=nb$。由于复本方案给定后，文件的复本数量和位置便保持不变，文件访问热度分布的变化（体现为偏斜度的变化或文件访问热度排名的变化）可引起系统 I/O 吞吐量的不同。此处 I/O 吞吐量的期望 $\mathrm{E}(W_{\max})$是指偏斜度给定条件下，各种文件访问热度排名顺序下 I/O 吞吐量 $W_{\max}$ 的平均值。

I/O 吞吐量 $W_{\max}$：在给定复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$ 约束下，保持各文件 f_j 被访问比例 $p_j(t)$不变（即文件访问热度排名顺序和偏斜度 a 均保持不变），通过调节各文件访问请求到达率之和 $?(t)$可调节 I/O 负载率之和 $W(t)$；仅依靠改变请求分发矩阵即可满足式(1)~式(5)共 5 个约束条件的 $W(t)$的最大值 $W_{\max}$，即为系统在给定复本方案、偏斜度及文件访问热度排名顺序条件下的 I/O 吞吐量。

I/O 吞吐量的期望 $\mathrm{E}(W_{\max})$：在给定复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$ 条件下，偏斜度 a 保持不变，通过改变文件访问热度排名的顺序，可改变 I/O 吞吐量 $W_{\max}$。各种文件访问热度排名顺序下 I/O 吞吐量的平均值称为给定偏斜度和复本方案条件下 I/O 吞吐量的期望，记为 $\mathrm{E}(W_{\max})$。

3.2　效率无关参数的识别

复本方案直接影响空间效率和 I/O 效率。此外，效率的可能影响因素还包括资源参数和需求参数。其中，资源参数主要指节点的 I/O 带宽 b、存储容量 c 以及节点数量 n；需求参数主要指文件大小 s、文件数量 m 以及文件访问热度分布（可用函数 $G(x)$ 或偏斜度 a 表示）。

本节的定理 1 识别出 3 个效率无关参数，即 b、c 和 s。

定理 1　在节点数量 n、文件数量 m、用以描述文件访问热度分布的函数 $G(x)$以及节点存储容量 c 与文件大小 s 的比值确定条件下，复本存储机制的效率与文件大小 s、节点存储容量 c 及 I/O 带宽 b 的具体取值无关。

证明　若可证得在定理所述条件下，任一给定复本方案$<r, \boldsymbol{D}_{n\times m}, \boldsymbol{E}_{n\times m}(t)>$下的 I/O 效率 $?$ 均与文件大小 s、节点存储容量 c 及 I/O 带宽 b 的具体取值无关，即可证得复本存储机制的效率与上述 3 个量无关（空间效率 e 已由复本数量 r 完全确定，因此与 b、c、s 无关）。

1) 证明 I/O 带宽 b 与 I/O 效率 $?$ 的无关性。

任一给定复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$，在定理 1 所述条件下，假设节点 I/O 带宽为 b_0 时的 I/O 效率为 $?_0$，I/O 吞吐量的期望为 $\mathrm{E}(W_{\max_0})$；不失一般性，令 $b_1=x_1b_0$（$x_1>0$），假设相同复本数量和复本放置矩阵在定理 1 所述条件下节点 I/O 带宽为 b_1 时的 I/O 效率为 $?_1$，I/O 吞吐量的期望为 $\mathrm{E}(W_{\max_1})$，则

$$?_0=\mathrm{E}(W_{\max_0})/B_0=\mathrm{E}(W_{\max_0})/(nb_0) \tag{6}$$

$$?_1=\mathrm{E}(W_{\max_1})/B_1=\mathrm{E}(W_{\max_1})/(nx_1b_0) \tag{7}$$

不失一般性，在任意一种给定的文件访问热度排名顺序下（该顺序下文件 f_j 被访问比例记为 p_j），假设节点 I/O 带宽为 b_0 时的 I/O 吞吐量为 $W_{\max_0}$，对应的访问请求到达率为 $?_{\max_0}$，请求分发矩阵为 $\boldsymbol{E}_{n\times m}$，并假设节点 I/O 带宽为 b_1 时的 I/O 吞吐量为 $W_{\max_1}$，对应的访问请求到达率为 $?_{\max_1}$，请求分发矩阵为 $\boldsymbol{E}'_{n\times m}$。于是

$$\sum_{j=1}^{m} e_{ij}\,sp_j\,?_{\max_0} \le b_0 \tag{8}$$

$$\sum_{j=1}^{m} e'_{ij}\,sp_j\,?_{\max_1} \le b_1 \tag{9}$$

在请求分发矩阵为 $\boldsymbol{E}_{n\times m}$ 条件下，将节点 I/O 带宽由 b_0 变为 b_1、访问请求到达率由 $?_{\max_0}$ 变为 $x_1?_{\max_0}$，不影响式(1)~式(4)的可满足性。此外，由式(8)可得

$$\sum_{j=1}^{m} e_{ij}\,sp_j\,(x_1\,?_{\max_0}) \le x_1b_0 = b_1$$

可见，在节点 I/O 带宽为 b_1、访问请求到达率为 $x_1?_{\max_0}$ 时，请求分发矩阵 $\boldsymbol{E}_{n\times m}$ 可保证式(5)也成立。根据 I/O 吞吐量的定义，$W_{\max_1}\ge sx_1?_{\max_0}$，即

$$W_{\max_1}\ge x_1W_{\max_0} \tag{10}$$

同理，在请求分发矩阵为 $\boldsymbol{E}'_{n\times m}$ 条件下，将节点 I/O 带宽由 b_1 变为 b_0、访问请求到达率由 $?_{\max_1}$ 变为 $?_{\max_1}/x_1$，不影响式(1)~式(4)的可满足性。此外，由式(9)可得

$$\sum_{j=1}^{m} e'_{ij}\,sp_j\,(?_{\max_1}/x_1) \le b_1/x_1 = b_0$$

可见，在节点 I/O 带宽为 b_0、访问请求到达率为 $?_{\max_1}/x_1$ 时，请求分发矩阵 $\boldsymbol{E}'_{n\times m}$ 可保证式(5)也成立。根据 I/O 吞吐量的定义，$W_{\max_0}\ge s?_{\max_1}/x_1$，即

$$W_{\max_0}\ge W_{\max_1}/x_1 \tag{11}$$

由式(10)和式(11)可得

$$W_{\max_1}=x_1W_{\max_0}$$

由于在任意一种给定文件访问热度排名顺序下可得 $W_{\max_1}=x_1W_{\max_0}$，于是

$$\mathrm{E}(W_{\max_1})=x_1\mathrm{E}(W_{\max_0}) \tag{12}$$

由式(6)、式(7)和式(12)可得

$$?_1=?_0$$

可见，在定理 1 所述条件下，任一复本方案下的 I/O 效率不因系统所采用节点 I/O 带宽 b 的改变而不同。

2) 证明文件大小 s 及节点存储容量 c 的具体数值与 I/O 效率 $?$ 的无关性。

任一给定复本数量 r 和复本放置矩阵 $\boldsymbol{D}_{n\times m}$，在定理 1 所述条件下，假设文件大小为 s_0、节点存储容量为 c_0 时的 I/O 效率为 $?_0$，I/O 吞吐量的期望为 $\mathrm{E}(W_{\max_0})$；不失一般性，令 $s_1=y_1s_0$，$y_1>0$，由于定理的条件中限定 c 和 s 的比值不变，可得 $c_1=y_1c_0$。假设相同复本数量和复本放置矩阵在定理 1 所述条件下文件大小为 s_1、节点存储容量为 c_1 时的 I/O 效率为 $?_1$，I/O 吞吐量的期望为 $\mathrm{E}(W_{\max_1})$，则

$$?_0=\mathrm{E}(W_{\max_0})/B \tag{13}$$

$$?_1=\mathrm{E}(W_{\max_1})/B \tag{14}$$

不失一般性，在任意一种给定的文件访问热度排名顺序下（该顺序下文件 f_j 被访问比例记为 p_j），

假设文件大小为 s_0、节点存储容量为 c_0 时的 I/O 吞吐量为 $W_{\max_0}$，对应的访问请求到达率为 $?_{\max_0}$，请求分发矩阵为 $\boldsymbol{E}_{n\times m}$，并假设文件大小为 s_1、节点存储容量为 c_1 时的 I/O 吞吐量为 $W_{\max_1}$，对应的访问请求到达率为 $?_{\max_1}$，请求分发矩阵为 $\boldsymbol{E'}_{n\times m}$。于是

$$\sum_{j=1}^{m} d_{ij} s_0 \le c_0 \tag{15}$$

$$\sum_{j=1}^{m} e_{ij} s_0 p_j l_{\max_0} \le b \tag{16}$$

$$\sum_{j=1}^{m} d_{ij} s_1 \le c_1 \tag{17}$$

$$\sum_{j=1}^{m} e'_{ij} s_1 p_j l_{\max_1} \le b \tag{18}$$

在请求分发矩阵为 $\boldsymbol{E}_{n\times m}$ 条件下，将文件大小由 s_0 变为 s_1，节点存储容量由 c_0 变为 c_1，访问请求到达率由 $?_{\max_0}$ 变为 $?_{\max_0}/y_1$，不影响式(1)、式(3)和式(4)的可满足性。此外，由式(15)和式(16)分别可得

$$\sum_{j=1}^{m} d_{ij} y_1 s_0 \le y_1 c_0 = c_1$$

$$\sum_{j=1}^{m} e_{ij} s_1 p_j (l_{\max_0}/y_1) \le b$$

可见，在文件大小为 s_1、节点存储容量为 c_1、访问请求到达率为 $?_{\max_0}/y_1$ 时，请求分发矩阵 $\boldsymbol{E}_{n\times m}$ 可保证式(2)和式(5)也成立。根据 I/O 吞吐量的定义，$W_{\max_1} \ge s_1 ?_{\max_0}/y_1$，即

$$W_{\max_1} \ge W_{\max_0} \tag{19}$$

同理，在请求分发矩阵为 $\boldsymbol{E'}_{n\times m}$ 条件下，将文件大小由 s_1 变为 s_0，节点存储容量由 c_1 变为 c_0，访问请求到达率由 $?_{\max_1}$ 变为 $y_1 ?_{\max_1}$，不影响式(1)、式(3)和式(4)的可满足性。此外，由式(17)和式(18)分别可得

$$\sum_{j=1}^{m} d_{ij} s_1/y_1 \le c_1/y_1 = c_0$$

$$\sum_{j=1}^{m} e'_{ij} s_0 p_j y_1 l_{\max_1} \le b$$

可见，在文件大小为 s_0、节点存储容量为 c_0、访问请求到达率为 $y_1 ?_{\max_1}$ 时，请求分发矩阵 $\boldsymbol{E'}_{n\times m}$ 可保证式(2)和式(5)也成立。根据 I/O 吞吐量的定义，$W_{\max_0} \ge s_0 y_1 ?_{\max_1}$，即

$$W_{\max_0} \ge W_{\max_1} \tag{20}$$

由式(19)和式(20)可得

$$W_{\max_1} = W_{\max_0}$$

由于在任意一种文件访问热度排名顺序下可得 $W_{\max_1} = W_{\max_0}$，于是：

$$\mathrm{E}(W_{\max_1}) = \mathrm{E}(W_{\max_0}) \tag{21}$$

由式(13)、式(14)和式(21)可得

$$?_1 = ?_0$$

可见，在定理 1 所述条件下，任一复本数量和复本放置矩阵下的 I/O 效率不因系统所采用节点存储容量 c 和文件大小 s 的改变而不同。

定理 1 的进一步解释。

复本存储机制的效率不因系统所采用节点的硬件配置（包括存储容量和 I/O 带宽）以及所存储文件的大小而变化。

3.3 2 种效率间的关系及节点数量对效率的影响

本节分析空间效率和 I/O 效率之间的关系，并分析节点数量 n 对效率的影响。

引理 1 文件访问热度服从偏斜度 $a>1$ 的 Zipf-like 分布，在节点 I/O 带宽 b 给定时，若保持空间效率 e 不变，则无论采用何种复本放置和请求分发算法，系统的 I/O 吞吐量 $W_{\max}$ 存在上限（即使节点数量 n 和文件数量 m 无限制增加）。

证明 在复本数量 r 和偏斜度 a 给定条件下，任一复本放置矩阵、请求分发矩阵、文件访问热度排名顺序、文件数量 m 和节点数量 n 下，当全体文件所产生 I/O 负载率之和等于 I/O 吞吐量 $W_{\max}$ 时，最重载节点 N_h 恰好满负荷，此时分发至节点 N_h 的 I/O 负载率恰好等于 b（即式(5)在 $i=h$ 时恰好取等号）。以 P 表示分发至节点 N_h 的 I/O 负载率与全体文件所产生 I/O 负载率之和的比值，则

$$W_{\max} = b/P \tag{22}$$

其中，N_h 为最重载节点，因此，分发至节点 N_h 的 I/O 负载率不小于访问热度排名第 1 的文件 $f_{\max}$ 所产生 I/O 负载率的 $1/r$（最重载节点 N_h 不一定存有文件 $f_{\max}$ 的复本，但系统中必存在存有文件 $f_{\max}$ 复本的节点 N'，且访问文件 $f_{\max}$ 的请求被分发至节点 N' 的比例不小于 $1/r$，而分发至最重载节点 N_h 的 I/O 负载率不小于节点 N'）。

以 $p_{\max}$ 表示文件 $f_{\max}$ 所产生 I/O 负载率占全体文件所产生 I/O 负载率之和的比值，则

$$P \ge p_{\max}/r \tag{23}$$

当文件访问热度服从 Zipf-like 分布时，

$$p_{\max}=1/(\sum_{k=1}^{m}1/k^{a})$$

因为 p 级数在 $p>1$ 条件下收敛，因此，当 $a>1$ 时，

$$\lim_{m\to\infty}\sum_{k=1}^{m}1/k^{a}=C_a$$

其中，C_a 由参数 a 完全确定。此外，令函数 $f(m)$ 定义为

$$f(m)=\sum_{k=1}^{m}1/k^{a}$$

则 $f(m)$是关于 m 的严格单调增函数，因此

$$p_{\max}>1/C_a \tag{24}$$

由式(23)和式(24)可得

$$P>1/(rC_a) \tag{25}$$

由式(22)和式(25)可得

$$W_{\max}<rbC_a$$

可见，若文件访问热度服从偏斜度 $a>1$ 的 Zipf-like 分布，在节点 I/O 带宽 b 给定时，若空间效率 e 不变（从而 r 不变），无论采用何种复本放置和请求分发算法，I/O 吞吐量 $W_{\max}$ 存在上限（即使节点数量 n 和文件数量 m 无限制增加）。

引理 1 的进一步解释。

1) 当文件访问热度服从偏斜度 $a>1$ 的 Zipf-like 分布时，最热门文件被访问比例不会少于某个下限值；无论采用何种复本方案，在不牺牲空间效率条件下，均无法单独通过增加节点数量的方式使得系统 I/O 吞吐量获得显著提升。

2) 尽管引理 1 指出在所述条件下，即使节点数量无限制增加，系统的 I/O 吞吐量存在上限，但并非意味着节点数量的增加对 I/O 吞吐量没有贡献。实际上，系统的 I/O 吞吐量仍然随节点数量增加而上升，但上升速度越来越缓慢；I/O 吞吐量无限接近却永远无法达到某个确定的上限。

3) 引理 1 的结论独立于具体的复本方案，但并非意味着复本方案对 I/O 吞吐量没有影响。实际上，引理 1 所给出的 I/O 吞吐量上限是所有复本方案的最高上限。若采用 I/O 效率较差的复本方案，则系统 I/O 吞吐量远小于所给出的上限值。

4) 引理 1 结论中所给出的上限值只取决于节点的 I/O 带宽 b、复本数量 r 以及访问热度分布偏斜度 a，而与复本存储模型中的其他参数（如文件大小 s、文件数量 m 等）无关。

可通过如下所给出的一个具体的复本方案实例，更加直观的解释引理 1 的含义。假设某复本方案下各文件均存储一个复本，即 $r=1$；各节点仅存储一个文件，从而节点数量和文件数量相等，即 $n=m$。文件访问热度服从 Zipf-like 分布，且偏斜度 $a=1.05$。该复本方案实例较简单，可采取直接求解的方法得出节点数量 n 和 I/O 吞吐量 $W_{\max}$ 之间的关系：首先，该实例中最重载的节点是存储访问热度排名第 1 文件的节点，在系统达到 I/O 吞吐量 $W_{\max}$ 时，分发至该节点的 I/O 负载率为 b；其次，分发至最重载节点的 I/O 负载率占全体文件所产生 I/O 负载率之和的比值 P 等于访问热度排名第 1 的文件在全部文件中的访问比例，即

$$p=1/\sum_{k=1}^{n}k^{-1.05}$$

从而，

$$W_{\max}=\frac{b}{p}=b\sum_{k=1}^{n}k^{-1.05}$$

n 和 $W_{\max}$ 的关系如图 1 所示(图中 b 代表节点 I/O 带宽)。可见，尽管 I/O 吞吐量 $W_{\max}$ 随节点数量 n 的增加而不断增长，但增长速度越来越缓慢，$W_{\max}$ 存在上限，该上限值为 bC_a，其中,

$$C_a=\lim_{n\to\infty}\sum_{k=1}^{n}k^{-1.05}$$

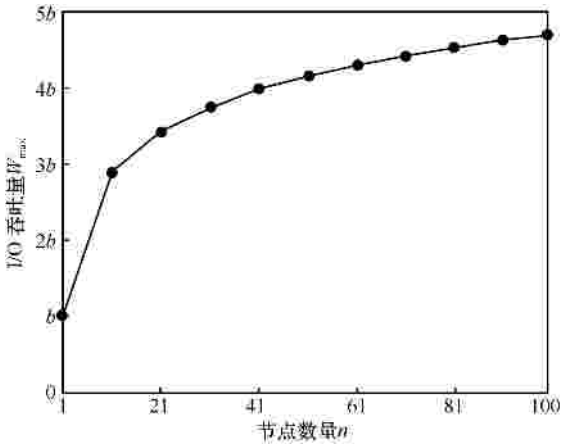

图 1　节点数量和 I/O 吞吐量之间的关系

定理 2　在文件访问热度服从偏斜度 $a>1$ 的 Zipf-like 分布时，复本存储机制所能获得的空间效率 e 和 I/O 效率 $?$ 乘积的最大值随节点数量 n 的增加而减少。

证明　根据空间效率和 I/O 效率的定义可得

$$e?=(1/r)(\mathrm{E}(W_{\max})/B) \tag{26}$$

由引理1的证明过程可知，任意文件访问热度排名顺序下均有 $W_{\max}<rbC_a$，从而

$$\mathrm{E}(W_{\max})<rbC_a \tag{27}$$

由式(26)和式(27)可得

$$e?<(1/r)(rbC_a/B)= C_a/n$$

其中，C_a 由参数 a 完全确定。$e?<C_a/n$ 在任意复本方案下均可成立。因此，复本存储机制所能获得的 e 和 $?$ 乘积的最大值随节点数量 n 的增加而减少。定理2表明，在系统规模增加时，至少有一个效率值会发生衰减。

定理2的进一步解释。

1) 随着节点数量的增加，若保持原有I/O效率不变，即保证I/O吞吐量 $W_{\max}$ 随节点数量线性增长，则空间效率呈现近似反比下降，从而使得系统数据容量 $S_{\max}$ 近似保持不变。

2) 随着节点数量的增加，若保持空间效率不变，即保证系统的数据容量 $S_{\max}$ 随节点数量线性增长，则I/O效率呈现近似反比下降，从而使得系统I/O吞吐量 $W_{\max}$ 近似保持不变。

3) 节点数量增加时，复本存储机制无法同时保持原有的空间效率和I/O效率，即无法同时实现数据容量 $S_{\max}$ 和I/O吞吐量 $W_{\max}$ 的线性增长。一般需要在2种效率的损失之间进行权衡，以使得系统数据容量 $S_{\max}$ 和I/O吞吐量 $W_{\max}$ 均获得一定程度的提升。

比如，系统节点数量增加至原来的16倍时，系统的存储空间容量和I/O带宽总量同时增至原来的16倍。但是，在文件访问热度服从偏斜度 $a>1$ 的Zipf-like分布时，无法实现系统数据容量 $S_{\max}$ 和I/O吞吐量 $W_{\max}$ 同时增至原来的16倍。假设系统数据容量 $S_{\max}$ 增至原来的2倍（空间效率降至原来的1/8），则I/O吞吐量 $W_{\max}$ 近似增至原来的8倍（I/O效率近似降至原来的1/2）。空间效率和I/O效率的乘积近似降至原来的1/16。

4) 复本存储机制效率随节点数量增加而不断下降的结论，揭示了当前大规模存储系统下的效率危机。本文的理论分析主要针对工程上获得广泛采用的静态复本存储机制，该效率危机间接论证了研究实用的动态复本存储机制的必要性和紧迫性。

4 复本存储机制效率的实验研究

独立于具体复本放置及请求分发算法，第3节通过理论分析得出有关复本存储机制效率的一般性结论，即：复本存储机制的效率与系统所采用节点的硬件配置（包括存储空间容量和I/O带宽）无关，但与节点数量紧密相关；空间效率和I/O效率的乘积随节点数量增加而不断减小。本节以Kinesis算法[5]为例，对前述研究结论进行实验验证，得出复本存储机制下权衡空间效率和I/O效率的一般性方法，以及大规模存储系统I/O性能的预测方法。

此外，第3节分析了4个模型参数的影响，识别出3个效率无关参数（即节点I/O带宽 b、节点存储容量 c 和文件大小 s）与一个效率相关参数（即节点数量 n）。本节将通过实验分析其余2个参数（即文件数量 m 和偏斜度 a）对效率的影响。

4.1 实验方法

本文的实验含原型系统实验和仿真实验。

原型系统实验　实验环境由24台PC机和一台以太网交换机构成。交换机为思科Catalyst 2690系列，含24个100 Mbit/s自适应端口。PC机的配置为英特尔酷睿2四核CPU、2.96 GB内存、320 GB SATA硬盘和百兆以太网网卡，其中，9台运行Kinesis服务器程序，15台运行Kinesis客户端程序。通过调节客户端发出请求的频率，可改变I/O负载率之和 $W(t)$（各请求随机选择文件进行访问，各文件被选中的概率由Zipf-like分布确定）；通过观察服务器端的请求队列是否发生溢出，可判断系统是否过载。通过多次测试可确定满足系统不过载的 $W(t)$ 的最大值，从而确定系统的I/O吞吐量 $W_{\max}$。

仿真实验　通过C++编程实现了一个网络排队系统，对Kinesis的复本放置和请求分发机制进行模拟，并用蒙特卡罗方法求取系统的I/O效率等指标。其中，节点的I/O带宽 b 和存储容量 c 没有采用绝对意义的数值，而根据其他量求取相对值。例如，若节点数量 n=10，相对I/O负载率 $W'(t)$=0.5，其中，$W'(t)=W(t)/B$，则在请求平均到达时间间隔内系统整体可处理请求的数量 $B'=1/W'(t)=2$，单个节点可处理请求的数量 $b'=B'/n=0.2$。节点的相对存储容量为 $c'=c/s$，表示节点可容纳文件的数量。

在原型系统和仿真2类实验中，均通过二分查找法得到系统在给定复本放置和文件访问热度分布下的I/O吞吐量。

在原型系统实验中，以 W_{top} 和 W_{bottom} 表示 $W_{\max}$ 取值的上界和下界。首先令 $W_{\text{bottom}}=0$，$W_{\text{top}}=nb$，$W(t)=nb/2$。若测得系统过载，则保持 W_{bottom} 不变，并先根据公式 $W_{\text{top}}=W(t)=nb/2$ 更新 W_{top}，再根据公

式 $W(t)=(W_{bottom}+W_{top})/2=nb/4$ 更新 $W(t)$；否则，保持 W_{top} 不变，并先根据公式 $W_{bottom}=W(t)=nb/2$ 更新 W_{bottom}，再根据公式 $W(t)=(W_{bottom}+W_{top})/2=3nb/4$ 更新 $W(t)$。$W(t)$更新后，重新测试系统是否过载。如此迭代，直至 $W_{top}-W_{bottom}<tnb$ 时(t 表征测试精度，实验中选择 $t=0.01$)，求得 $W(t)=(W_{bottom}+W_{top})/2$。

在仿真实验中，以 W'_{top} 和 W'_{bottom} 表示 W'_{max} 取值的上界和下界。首先令 $W'_{bottom}=0$，$W'_{top}=1$，$W'(t)=1/2$，若测得系统过载，则保持 W'_{bottom} 不变，并先根据公式 $W'_{top}=W'(t)=1/2$ 更新 W'_{top}，再根据公式 $W'(t)=(W'_{bottom}+W'_{top})/2=1/4$ 更新 $W'(t)$；否则，保持 W'_{top} 不变，先根据公式 $W'_{bottom}=W'(t)=1/2$ 更新 W'_{bottom}，再根据公式 $W'(t)=(W'_{bottom}+W'_{top})/2=3/4$ 更新 $W'(t)$。$W'(t)$更新后，重新测试系统是否过载。如此迭代，直至 $W'_{top}-W'_{bottom}<t$ 时，求得 $W'(t)=(W'_{bottom}+W'_{top})/2$（实验中选择 $t=0.01$）。

在原型系统和仿真 2 类实验中，判断系统是否过载的依据均是观察系统在给定运行时间 T 内是否发生服务器节点的请求队列溢出（各服务器节点请求队列长度设为 100）。在原型系统实验中，T 取值 30 min。在仿真实验中，由于不需要实际数据读取操作，测试所花费的时间代价较小。仿真系统运行数分钟即可模拟原型系统数天的运行结果。因此，仿真实验所模拟的系统运行时间 T 远大于 30 min。

4.2　相关结论的实验验证

4.2.1　节点硬件配置的效率无关性

由定理 1 可知，复本存储机制的效率独立于节点的硬件配置。由于仿真实验不限定单个存储节点的具体配置，实验 1 通过仿真实验和原型系统实验结果的比较，验证节点硬件配置的效率无关性。

实验 1　节点数量 $n=9$，文件数量 $m=200$，复本数量 $r=2$，偏斜度 a 介于 0.8~1.2 之间（定理 1 不限定 $a>1$）。文件放置完毕后，随机为其指定访问热度排名。通过测试系统的 I/O 吞吐量，求取 I/O 效率 $?$。原型系统和仿真 2 类实验均通过随机指定的多种不同访问热度排名下所测得的 I/O 吞吐量的平均值近似表示 I/O 吞吐量的期望值。

由于节点数量较少，文件访问热度排名指定的随机性使得相同 a 取值条件下的多次实验结果之间存在较大差异性。图 2 显示了以 I/O 吞吐量平均值代替期望值所测得的 I/O 效率近似值。可见，仿真和实测结果十分接近，从而验证了节点硬件配置和效率的无关性。

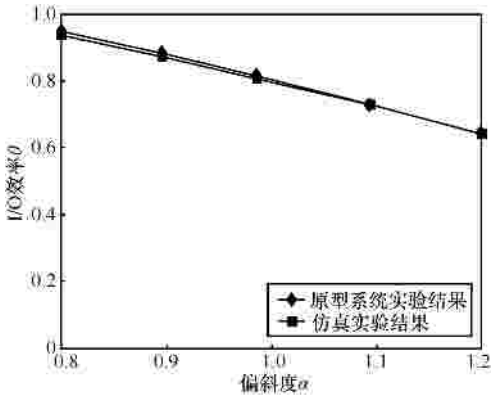

图 2　仿真实验和原型系统实验结果比较

从图 2 还可以看出，1）仿真实验结果略小于原型系统实验结果。这是因为，原型系统实验以系统运行 30 min 的结果判断其是否过载，由于文件访问具有一定的随机性，运行 30 min 不过载并不能完全确定系统不会过载，即原型系统实验的测试方法可能导致测得的 I/O 吞吐量偏大。由于仿真实验的代价较小，数分钟即可模拟原型系统数天的运行结果，更有利于过载事件的检测。2）文件访问偏斜度越大，则 2 类实验结果的差别越不显著。这是因为，文件访问偏斜度越大，则访问请求越可能集中于少数几个文件，如此，I/O 负载率稍稍超出 I/O 吞吐量即可能导致存有热点文件的服务器节点过载，反之，访问请求较均匀的分布于不同文件之间，I/O 负载率稍大时很难在较短时间内出现某服务器节点的请求队列溢出现象，从而更容易导致测试结果偏高。

由于仿真实验可近似表示原型系统实验的结果，且测试所花费时间远小于原型系统实验，后文中的实验（即实验 2~实验 8）均指仿真实验。另外，大量仿真实验发现：随着节点数量的增加，文件访问热度排名指定的随机性对 $?$ 的影响越来越微弱。例如，在 $n=50$、$a=1.0$、$r=2$ 时，10 次随机实验均测得 $?=0.35$。可见，在节点数量较大时，文件访问热度排名的变化对 Kinesis 算法[5]的效率几乎没有影响；尽管文件访问热度排名动态变化，只要偏斜度保持不变，则 Kinesis 算法[5]的 I/O 吞吐量保持相对稳定。

4.2.2　节点数量对效率的影响

引理 1 指出，在文件访问热度服从 Zipf-like 分布且偏斜度 $a>1$ 条件下，即使节点数量无限制增加，系统的 I/O 吞吐量存在上限。在 3.3 节中已通过一个极端的复本方案对引理 1 的结论进行了解释（如

图 1 所示)。实验 2 进一步通过 Kinesis 算法[5]所产生的复本方案对该结论进行验证。

实验 2 复本数量 r=3，偏斜度 a=1.05，节点数量 n 介于 50~650 之间，文件数量 m=100n。节点数量和 I/O 吞吐量之间的关系如图 3 所示（图中 b 表示节点 I/O 带宽)。可见，随着节点数量的增加，I/O 吞吐量的增长速度越来越缓慢。

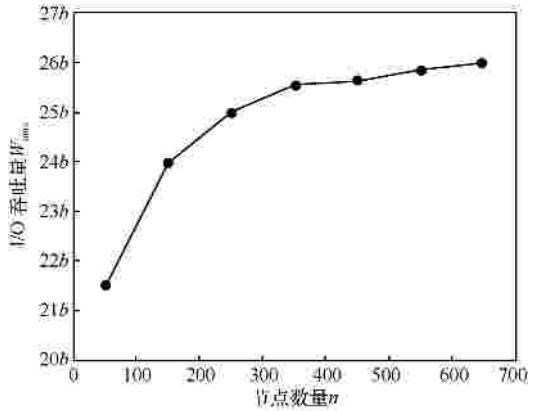

图 3 节点数量和 I/O 吞吐量的关系

实验 2 同时给出在空间效率固定条件下，节点数量和 I/O 效率之间的关系，如图 4 所示。可见，随着节点数量的增加，I/O 效率近似成反比下降。

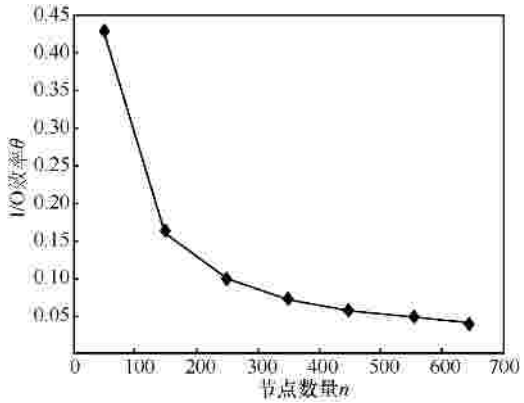

图 4 节点数量和 I/O 效率的关系

实验 3 在不同节点数量条件下求取满足给定 I/O 效率需求的最小复本数量，进而获得空间效率和节点数量的关系。

实验 3 偏斜度 a=1.05，节点数量 n 介于 50~650 之间，文件数量 m=100n/r，求满足 I/O 效率不低于 0.4 的最小复本数量。如图 5 所示。可见，随着节点数量的增加，为满足给定 I/O 效率所需的最小复本数量几乎呈线性增长，因此，空间效率成近似反比下降。

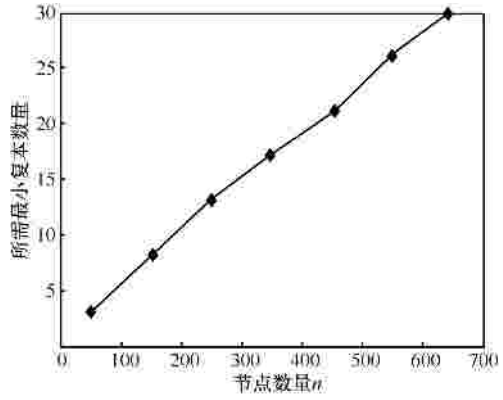

图 5 节点数量与满足给定 I/O 效率需求的最小复本数量之间的关系

4.3 相关结论的主要应用

4.3.1 2 种效率间的权衡

实验 4 节点数量 n=200，偏斜度 a=1.05，文件数量 m=100n/r，复本数量 r 介于 2~20 之间，e 和 $?$ 乘积的变化如图 6 所示。

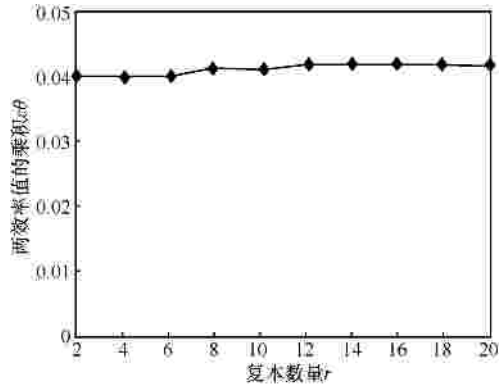

图 6 两效率值乘积的变化

由图 6 可知，e 和 $?$ 的乘积近似为常量 0.04。根据 $e?$=0.04，可估计 $?$ 随 e 的变化情况。实测结果和估计值的比较如图 7 所示，二者之间的差异性很小。因此，在 n 和 a 给定时，可以将 e 和 $?$ 的乘积视为常量。该常量可表征给定规模系统的整体效率。

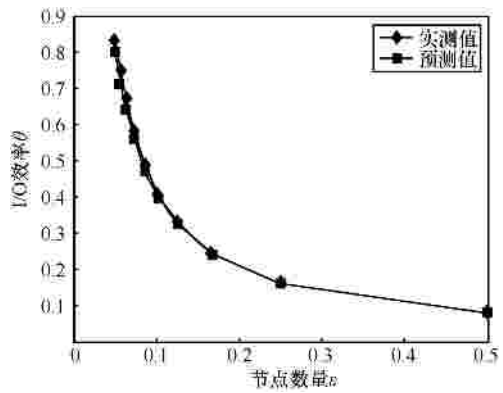

图 7 空间效率 e 和 I/O 效率 $?$ 之间的关系

因此，可得出 2 种效率权衡的一般性方法。对给定节点数量 n 和偏斜度 a，若已测得某一空间效率 e_0 所对应的 I/O 效率 $?_0$，则根据函数关系式 $?=e_0?_0/e$ 可获取 $?$ 和 e 的关系曲线，并据此对二者进行权衡。

4.3.2　大规模存储系统 I/O 性能的预测

本文以 I/O 吞吐量 W_{max} 表征系统 I/O 性能。通过小规模系统的实验结果，可预测大规模系统在给定复本数量 r 下可获得的 I/O 性能。

假设在给定偏斜度条件下，已测得系统在小规模 n_0 下的一组效率值 e_0、$?_0$，根据 2 个效率值的乘积与系统中节点数量成反比的结论，可预测系统在大规模 n 下，空间效率 e 所对应的 I/O 效率 $?$，即根据 $e?n\tilde{}\ e_0?_0n_0$ 得出：$?\tilde{}\ (e_0?_0n_0)/(en)$。另外，根据 I/O 效率的定义，$?=W_{max}/(nb)$。因此，

$$W_{max}=nb?\tilde{}\ (nb)(e_0?_0n_0)/(en)=be_0?_0n_0r$$

当节点数量 n_0=50、偏斜度 a=1.0 时，已测得 $e_0?_0\tilde{}\ 0.175$。因此，当节点数量为 $n(n>n_0)$、复本数量为 r 时，$W_{max}\tilde{}\ 0.175rn_0b$。

实验 5　偏斜度 a=1.0，节点数量 n 介于 100~500 之间，r 取值 2~20 之间，W_{max} 的预测结果和实验结果分别如图 8 和图 9 所示（图中 b 表示节点 I/O 带宽）。

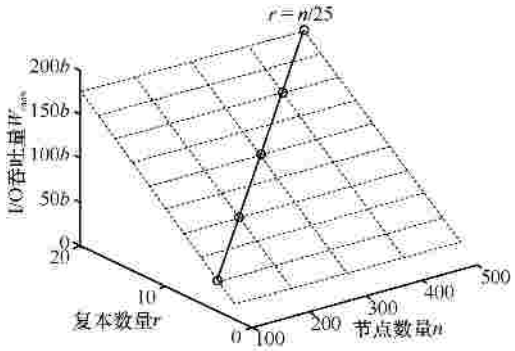

图 8　I/O 吞吐量 W_{max} 的预测结果

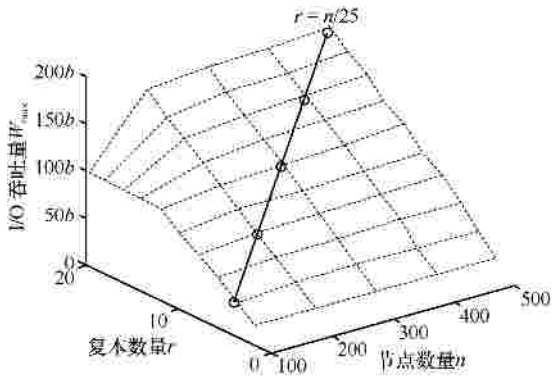

图 9　I/O 吞吐量 W_{max} 的实验结果

在图 8 中，n、r 和 W_{max} 位于同一平面。而图 9 中，n 介于 100~200 之间、r>12 的区域发生凹陷，即实验结果小于预测结果（这是因为 n=100、r=12 时，W_{max} 已接近 B，r 的继续增加对 W_{max} 没有贡献）。

一般情况下，系统扩容后可存储的文件数量不减少，因此，复本数量 r 的增长倍数不会大于节点数量 n 的增长倍数，即 $r\leq r_0(n/n_0)$。在图 8 和图 9 中，分界线的解析式为 $r=r_0(n/n_0)$，r_0 取 4，n_0 取 100。可以看出，$r\leq n/25$ 时，预测结果和实测结果十分接近。

此外，尽管引理 1 和定理 2 成立的充分条件是 a>1，实验中发现，该条件不成立但 a 较接近 1 时，结论仍近似成立。

因此，可得出大规模存储系统 I/O 性能预测的一般性方法。首先，在给定偏斜度 a（a>1）条件下，测试节点数量为 n_0、空间效率为 e_0 时的 I/O 效率 $?_0$，则相同偏斜度下，节点数量为 n（$n>n_0$）的大规模系统在复本数量为 r 时的 I/O 吞吐量可预测为

$$W_{max}\tilde{}\ be_0?_0n_0r$$

其中，n_0 越大，预测结果越精确，但进行测试的难度越高。实验中发现，a=1.05 条件下，n_0>50 时即可保证较高的预测精度。

4.4　文件数量和偏斜度对效率的影响

4.4.1　文件数量的影响分析

实验 6　偏斜度 a=1.05，节点数量 n=200，复本数量 r=3，文件数量 m 介于 10 000~100 000 之间。结果如图 10 所示。随着文件数量的增加，I/O 效率变化较小，一直在 0.12~0.14 之间。可见，当文件数量较大（比如满足 m>100n）条件下，其具体取值对效率的影响较小。

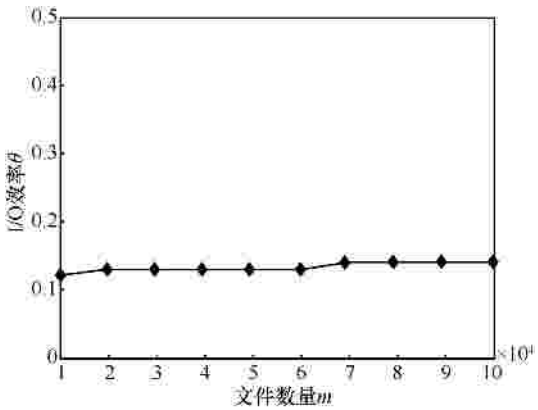

图 10　文件数量和 I/O 效率之间的关系

4.4.2　偏斜度的影响分析

由定理 2 可知，随着系统规模增长，复本存储

机制的整体效率降低。然而，基于复本机制的大规模存储系统在某些实际应用中能够较好的满足其存储空间容量、I/O 性能以及系统成本等需求。这是因为，复本存储机制的效率除和节点规模相关外，还和具体应用的数据访问特征密切相关。本小节对给定空间效率条件下的偏斜度 a 和 I/O 效率之间的关系进行实验分析。

实验 7 节点数量 n=100，复本数量 r=3，偏斜度 a 介于 0.75~1.0 之间。实验所测得的 I/O 效率 $?$ 和偏斜度 a 之间的关系如图 11 所示。可见，随着偏斜度的减小，I/O 效率急速上升。当偏斜度 a 较小时，Kinesis 算法[5]可保证系统较高的 I/O 效率。

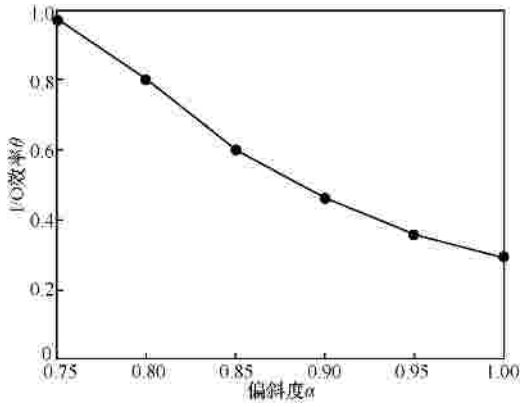

图 11 偏斜度 a 对 I/O 效率 $?$ 的影响

实验 8 偏斜度 a=0.7，复本数量 r=3，文件数量 m=100n，节点数量 n 介于 100~500 之间，I/O 效率和节点数量之间的关系如图 12 所示。

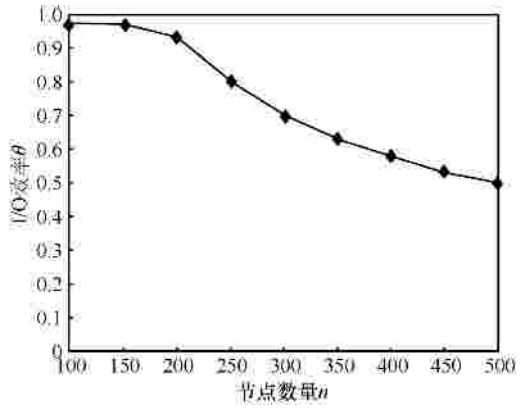

图 12 节点数量和 I/O 效率之间的关系（a=0.7）

可见，即使偏斜度较小，当节点数量增加时，I/O 效率依然下降，只是下降速度较慢（不再与节点数量成简单反比关系）。

5 结束语

本文通过建立独立于具体算法的复本存储模型，对复本存储机制的效率进行理论分析，识别出影响效率的关键参数和无关参数，证明在文件访问热度服从偏斜度大于 1 的 Zipf-like 分布时，复本存储机制的 I/O 吞吐量存在上限，并得出空间效率和 I/O 效率的乘积同系统中节点的数量近似成反比这一结论。本文对 Kinesis 算法[5]的实验研究同时验证了理论分析的结果。实验结果表明，本文的结论可作为系统扩容时对空间效率和 I/O 效率进行权衡的依据，并可用于预测大规模存储系统的 I/O 性能。本文的研究结论独立于节点本身的硬件配置和具体的复本方案，具有普遍指导意义。

此外，本文的研究结论揭示了当前大规模存储系统所面临的严重效率危机。由于本文的理论分析主要针对工程上广泛采用的静态复本存储机制，该效率危机间接论证了研究实用的动态复本存储机制的必要性和紧迫性。

然而，本文的约束条件较强，如节点同构、访问热度服从 Zipf-like 分布等，未来可开展的进一步工作为该机制的工程应用提供更全面的理论指导。

参考文献：

[1] 聂雪军. 内容感知存储系统中信息生命周期管理关键技术研究[D]. 武汉: 华中科技大学, 2011.
NIE X J. Research on Key Technologies of Information Lifecyc Management in Content Aware Storage System[D]. Wuhan: Huazhong University of Science and Technology, 2011.

[2] KIM J, CHOU J, ROTEM D. Energy proportionality and performance in data parallel computing clusters[A]. Proceedings of Scientific and Statistical Database Management Conference[C]. Portland, USA, 2011. 414-431.

[3] 陆承涛. 存储系统性能管理问题的研究[D]. 武汉: 华中科技大学, 2010.
LU C T. Performance Management in Storage Systems[D]. uhan: Huazhong University of Science and Technology, 2010.

[4] AGUILERA M K, JANAKIRAMAN R, XU L. Using erasure codes efficiently for storage in a distributed system[A]. Pr ceedings of International Conference on Dependable Systems and Networks[C]. Yokohama, Japan, 2005. 336-345.

[5] MACCORMICK J, MURPHY N, RAMASUBRAMANIAN V, *et al*. Kinesis: a new approach to replica placement in distributed storage systems[J]. ACM Transactions on Storage, 2009, 4(4):1-28.

[6] BRESLAU L, CAO P, FAN L, *et al*. Web caching and zip-like distributions: evidence and implications[A]. Proceedings of INFOCOM[C]. New York, USA, 1999. 126-134.

[7] TAN W, FU Y, CHERKASOVA L, *et al*. MediSyn: a synthetic streaming media service workload generator[A]. Proceedings o the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video[C]. Monterey, USA, 2003. 12-21.

[8] DABEK F, KAASHOEK M F, KARGER D, *et al*. Wide-area cooperative storage with CFS[A]. Proceedings of the 18th ACM mposium on Operating Systems Principles[C]. Banff, Canada, 2001.202-215.

[9] VAN RENESSE R, SCHNEIDER F B. Chain replication for supporting high throughput and availability[A]. Proceedings of the Symposium on Operating Systems Design and Implementation[C]. San Francisco, USA, 2004. 91-104.

[10] XIE C, CAI B. A decentralized storage cluster with hig liability and flexibility[A]. Proceedings of the 14th Euromicro International Conference on Parallel, Distributed and Network-Based Processing[C]. Montbeliard-Sochaux, France, 2006.116-123.

[11] WANG W, ZHAO Y. A novel network storage scheme: intelligent network disk storage cluster[A]. Proceedings of the 5th International Conference on Networking, Sensing and Control[C]. Sanya, China, 2008. 142-147.

[12] WEI Q. CDRM: a cost-effective dynamic replication management scheme for cloud storage cluster[A]. Proceeding of the IEEE International Conference on Cluster Computing[C]. Heraklion, Greece, 2010. 88-196.

[13] WANG Z, LI T, XIONG N, *et al*. A novel dynamic network data replication scheme based on historical access record and ve deletion[J]. The Journal of Supercomputing, 2012, 62(1):227-250.

[14] KHAN S U, AHMAD I. A pure Nash equilibrium-based game theoretical method for data replication across multiple servers[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(4):537-553.

[15] GHEMAWAT S, GOBIOFF H, LEUNG S T. The google file system[A]. Proceedings of the 19th ACM Symposium on Operating Systems Principles[C]. Bolton, UK, 2003. 29-43.

[16] TFS(taobao file system)[EB/OL]. http://tfs.taobao.org/.

[17] HDFS(hadoop distributed file system)[EB/OL]. http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdf s/Federation.html.

[18] QU Y, XIONG N. RFH: a resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage[A]. Proceedings of the 41st International Conference on Parallel Processing[C]. Pittsburgh, USA, 2012. 520-529.

[19] ANANTHANARAYANAN G, AGARWAL S, KANDULA S, *et al*. Scarlett: coping with skewed content popularity in mapreduce clusters[A]. Proceedings of the 6th Conference on Computer Systems[C]. Salzburg, Austria, 2011. 287-300.

[20] **陈康, 余宏亮, 张堃. 对等网络中基于位置信息和文件流行度的自适应复本管理算法**[J]. **计算机学报**, 2009, 32(10):1927-1937.
CHEN K, YU H L, ZHANG K. Adaptive replication manageme t algorithm based on location and file popularity for peer-to-peer network[J]. Chinese Journal of Computers, 2009, 32(10):1927-1937.

[21] KAVALANEKAR S, WORTHINGTON B, ZHANG Q, *et al*. Characterization of storage workload traces from production indows servers[A]. Proceedings of IEEE International Symposium on Workload Characterization[C]. Seattle, USA, 2008. 119-128.

[22] YADWADKAR N J, BHATTACHARYYA C, GOPINATH K, *et al*. Discovery of application workloads from network file traces[A]. Proceedings of the 8th USENIX Conference on File and Storage Technologies[C]. San Jose, USA, 2010.183-196.

[23] OSTROWSKI J R, SARHAN N J. Characterization of social video[A]. Proceedings of IS&T/SPIE Electronic Imaging[C]. San Jose, USA, 2009.347-356.

[24] GAO S, WANG H, WANG J. Research and application of web caching workload characteristics model[A]. Proceedings of the 2nd IEEE International Conference on Information Management and E gineering[C]. Chengdu, China, 2010. 105-109.

[25] CHOI J, REAZ A S, MUKHERJEE B. A survey of user behavior in VoD service and bandwidth-saving multicast streaming schemes[J]. IEEE Communications Surveys & Tutorials, 2012, 14(1):156-169.

[26] YU H, ZHENG D, ZHAO B Y, *et al*. Understanding user behavior in large-scale video-on-demand systems[A]. Proceedings of EuroSys'06 [C]. Leuven, Belgium, 2006. 333-344.

作者简介：

罗香玉（1984-），女，河北邢台人，东南大学博士生，主要研究方向为分布式存储系统。

汪芸（1967-），女，江苏苏州人，博士，东南大学教授、博士生导师，主要研究方向为容错理论、分布式计算。

陈笑梅（1989-），女，江苏南通人，东南大学硕士生，主要研究方向为分布式存储系统。

袁飞飞（1988-），男，江苏南通人，东南大学硕士生，主要研究方向为分布式存储系统。

李聪（1987-），女，江苏徐州人，东南大学硕士生，主要研究方向为工作流调度。