第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

一种基于 qemu 的动态迁移模型

王 森[1] 朱常鹏[1] 韩 博[2]

(重庆理工大学计算机科学与工程学院 重庆 400050)[1] (西安交通大学计算机系 西安 710049)[2]

摘 要 在线迁移已经成为数据中心的一个核心管理工具,广泛用于负载平衡、服务器整合和系统维护等方面。精确地预测在线迁移性能是制定有效迁移决策的前提。在广泛用于开源云计算的 qemu-kvm 虚拟化平台中,迁移策略与传统的预拷贝策略存在差异,导致已有的迁移模型无法有效地应用于该平台。为此,提出一种基于 qemu-kvm 平台的迁移策略的建模方法,基于模型抽取影响在线迁移性能的关键因素,分析它们与迁移性能之间的数学关系,最后针对这些关键参数建立相应的测试环境,以此测试评估模型的正确性与精确性。测试结果表明模型预测迁移时间和迁移数据总量的精确度在 95% 以上。

关键词 虚拟机,在线迁移,性能模型

中图法分类号 TP316 文献标识码 A

Analytical Model for Qemu-based Live Migration Strategy

WANG Sen[1] ZHU Chang-peng[1] HAN Bo[2]

(College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400050, China)[1]

(Department of Computer and Science, Xi'an Jiaotong University, Xi'an 710049, China)[2]

Abstract Live migration is a powerful management tool in data center and has been widespreadly applied for virtual machine load balancing, fault tolerance, power management and other applications. Whether evaluation for performance of live migration of virtual machines is precise or not has directly influence on effects of live migration decisions. Therefore, we proposed an analytical model for qemu-based live migration of virtual machines. Based the model, we extracted key parameters that affect the performance of live migration, and analyzed mathematic relations between these parameters and performance of live migration. Finally, we built some experiments to evaluate and verify the correctness and precision of the analytical model by comparing experiential and analysis results. Our experiential results show that the model yields higher than 95% prediction accuracy in migration time and total transferred data.

Keywords Virtual machine, Live migration, Performance model

1 引言

虚拟机(Virtual Machine, VM)作为一种计算资源容器,实现了软件与硬件之间的分离[12],有利于提高硬件资源利用率、隔离应用程序和容错处理,已被广泛地应用于现代数据中心,支持基础设施即服务的云计算。虚拟机动态迁移(Live Migration of Virtual Machines)是指在不中断虚拟机执行的前提下,将虚拟机从源主机迁移到目的主机,其中迁移时间、数据传输量与停机(downtime)时间是衡量迁移性能的 3 个关键指标。在线迁移已经成为数据中心的一个核心管理工具,广泛用于负载平衡、服务器整合和系统维护等方面。如何优化虚拟机在线迁移是虚拟化技术和云计算等当前热点研究领域共同面临的重大挑战。

近年来,虚拟机的动态迁移性能模型已被提出,这为量化分析资源对迁移性能的影响和研究优化迁移性能的方法提供了理论基础。比如,Akoush[3] 提出了动态迁移性能预测模型,Aldhalaan[4] 提出一种性能分析模型,分析网络速率对迁移时间影响的方法,并采用非线性优化方法求解满足网络速率约束下的最短迁移时间。

已有的动态迁移性能模型均基于经典的预拷贝(pre-copy)策略,即通过预拷贝、循环拷贝和停机拷贝 3 步完成虚拟机的动态迁移。基于 qemu-kvm 的虚拟化平台采用一种新的拷贝策略支持虚拟机的动态迁移,两种策略的不同之处主要在于:

. 后者取消了预拷贝阶段,即并不预先传输虚拟机的所有内存数据到目的端。

. 在循环拷贝阶段,后者要求每次传输的数据量到达一阈值就终止传输,并进入到下一次循环传输,而前者要求传输完在上次传输时间段内产生的所有脏页面之后才能进入下一次循环传输。

. 尽管后者可视为前者的一种变种,但是上述差异导致已有的动态迁移模型无法有效地用于描述和预测基于 qemu-

本文受国家自然科学基金(61173040)资助。

王 森(1978—),男,硕士,讲师,主要研究方向为虚拟机与虚拟化,E-mail:wangsen@cqut.edu.cn;朱常鹏(1981—),男,博士,讲师,主要研究方向为虚拟机与虚拟化;韩 博 博士,高级工程师,硕士生导师,主要研究方向为虚拟机与虚拟化、信息管理与融合。

kvm 的虚拟化平台的动态迁移，为此，本文提出一种基于 qemu-kvm 的虚拟化平台动态迁移性能模型，分析影响迁移性能的关键参数，并通过实验验证性能模型的精确性，为进一步研究优化提高动态迁移性能的方法提供基础。

2 动态迁移分析模型

本节首先描述基于 qemu-kvm 的动态迁移过程，然后通过数学分析建立动态迁移的理论模型，图 1 展示了该动态迁移过程，主要由循环拷贝、停机拷贝两个阶段组成。循环拷贝主要以循环方式将脏页面传输到目的主机 每当传输的数据量超过某一指定阈值时，本次循环传输结束并开始下一次循环。若剩余的脏页面低于某一指定阈值时，整个循环拷贝阶段结束转而进入停机拷贝阶段。该阶段首先终止虚拟机的运行，然后将剩余的脏页面一次传输到目的主机，然后在目的端重启虚拟机，完成虚拟机的动态迁移，为进一步地研究优化迁移性能的方法提供基础。

图 1 基于 qemu 的动态迁移过程

迁移时间、停机时间和数据传输总量是衡量动态迁移性能的 3 个关键指标。通过量化分析上述两个过程可为这些指标建立分析模型。为了便于描述建模过程，我们需使用若干符号(notations)。比如，使用符号 V_T 和 V_P 分别表示终止单次循环传输和进入停机拷贝阶段的阈值，符号 $P(i)$ 表示第 i 次循环传输的页面数。此外，在动态迁移过程中页面可分为正常页面、空页面和相似页面，其中空页面的传输可转化为对页面自身信息(比如页面的地址)而不是页面内容的传输，而相似页面的传输可以使用基于异或的压缩算法传输。针对后两种页面传输的优化可以极大地减少传输数据总量。基于此，我们使用符号 P_n，P_z 和 P_x 分别表示正常页面、空页面和相似页面，并且为了简化描述，我们使用符号 u，v 和 w 分别表示这 3 种页面占总页面的百分比，显然 $u+v+w=1$。

表 1 动态迁移性能分析模型中的符号及其注释

B	可用网络带宽(MB/sec)
S	页面大小(kB)
d	内存脏页率(pages/sec)
P(i)	第 i 次循环传输的页面数
V	虚拟机的内存大小(MB)
V_T	终止单次循环传输的阈值
V_P	终止整个循环传输的阈值(进入停机传输阶段的阈值)
$P_n(i)$	第 i 次循环传输中的正常页面数
$P_z(i)$	第 i 次循环传输中的空页面数
$P_x(i)$	第 i 次循环传输中可压缩的页面数
u	$P_n(i)$ 与 $P(i)$ 的比值
v	$P_z(i)$ 与 $P(i)$ 的比值
w	$P_x(i)$ 与 $P(i)$ 的比值
K_z	空页面的压缩比值
K_x	可压缩页面的压缩比值

在循环拷贝阶段，每次循环传输并不是所有的脏页面全部被传输，因此可以使用符号 $D(i)$ 表示每次循环传输完后还剩余的脏页面数，$D(i)$ 主要可分为 3 部分：上次循环传输剩余的脏页面数、本次循环传输产生的新脏页面数以及已被传输的脏页面数，具体表示如下：

$$D(i)=D(i-1)+dt-P(i) \tag{1}$$

其中，$t=V_T/B$。

每次循环传输的最大数据量为 V_T，并且传输不同页面产生的实际数据量也不尽相同，因此 V_T 与 $P(i)$ 之间的关系可表示如下：

$$uP(i)*S+vP(i)*S/K_z+wP(i)*S/K_x=(u+v/K_z+w/K_x)P(i)S\leqslant V_T \tag{2}$$

因此 $P(i)$ 可表示如下：

$$P(i)\leqslant V_T/S(u+v/K_z+w/K_x)\leqslant \alpha V_T/S \tag{3}$$

其中，$\alpha=1/(u+v/K_z+w/K_x)$，可视为一个页面的平均压缩比。

基于式(1)和式(3)有：

$$\begin{aligned} D(n) &\geqslant D(n-1)+dt-\alpha V_T/S \\ &=D(0)+n(dt-v/S) \\ &=D(0)+n(dV_T/B-\alpha V_T/S) \\ &=V/S+nV_T(d/B-\alpha/S) \end{aligned} \tag{4}$$

其中，$D(0)S=V$。

假设循环传输阶段总共进行 $n+1$ 次传输(从第 0 次到第 n 次)，则 $D(n+1)$ 表示为停机拷贝阶段的脏页面数。

根据进入停机拷贝阶段的条件有：

$$D(n+1)\geqslant V/S+(n+1)V_T(d/B-\alpha/S) \tag{5}$$

$$D(n+1)S\leqslant V_P \tag{6}$$

基于式(5)和式(6)有：

$$V+(n+1)(dSV_T/B-\alpha V_T)\leqslant V_P \tag{7}$$

dSV_T/B 表示在每次循环拷贝时间段内新产生的脏数据量，而 αV_T 则表示在每次循环拷贝时间段内实际的数据传输，显然，$\alpha V_T\leqslant dSV_T/B$，否则迁移无法正常终止。因此，式(7)可表示为：

$$\begin{aligned} &V+(n+1)V_T(\tau-\alpha)\leqslant V_P \\ &(n+1)V_T(\tau-\alpha)\leqslant V_P-V \\ &n+1\geqslant (V_P-V)/V_T(\tau-\alpha) \\ &\text{由于}(\tau-\alpha)<0 \\ &n+1=\lceil (V_P-V)/V_T(\tau-\alpha)\rceil \end{aligned} \tag{8}$$

其中，$\tau=dS/B$。

可以看出，页面的平均压缩比 α、网速 B 和脏页率 d 直接影响循环次数 n。

停机时间 T_{down} 即是第 $n+2$ 次循环拷贝所有剩余脏页面的时间，可表示为：

$$T_{down}=D(n+1)S/B\leqslant V_P/B \tag{9}$$

迁移时间 T_{mig} 由循环拷贝时间与停机拷贝时间组成，其中每次循环拷贝的时间 $t=P(i)S/B$，因此 T_{mig} 具体可表示为：

$$T_{mig}=\sum_{i=0}^{n}P(i)S/B+T_{down} \tag{10}$$

由于每次循环次拷贝的最大数据量为 V_T，因此 $P(i)S\leqslant V_T$，式(10)可表示为：

$$\begin{aligned} T_{mig} &\leqslant (n+1)\ V_T/B+V_P/B \\ &=(\lceil (V_P-V)/V_T(\tau-\alpha)\rceil\ V_T+V_P)/B \\ &=(\lceil (V_P-V)/(\tau-\alpha)\rceil+V_P)/B \\ &=\lceil (V+(\alpha-1-\tau)V_P)/(\tau-\alpha)\ \rceil/B \end{aligned} \tag{11}$$

总的传输数据 $P_{TotalMig}$ 即是循环拷贝阶段传输的总数据与停机阶段拷贝的数据之和，因此可以表示为：

$$P_{totalmig}=\sum_{i=0}^{n}P(i)S+D(n+1)S$$
$$\leqslant(n+1)V_T+V_P$$
$$=\lceil(V+(\alpha-1-\tau)V_P)/(\tau-\alpha)\rceil \quad (12)$$

3 实验

本节主要描述如何通过若干实验验证上述模型的准确性。实验环境由两台联想 ThinksStation 工作站组成，它们通过千兆交换机相连接。工作站的硬件配置为一颗 intel E3-1225 CPU 和 12GB 内存，软件环境为 Ubuntu 14.04LTS、qemu-2.0 和 libvirt 1.26；客户机的硬件配置为一颗 VCPU 和 1GB 内存，操作系统为 Centos 5.2。

首先，通过调整网络带宽测试迁移时间和数据传输总量的变化，揭示网络带宽与迁移时间之间和数据传输总量之间的关系，然后基于测试数据验证式(11)和式(12)的有效性。图 2 和图 3 分别展示了测试结果。在图 2 中，方形点表示基于式(11)计算出的理论迁移时间，而菱形点表示实际测得的迁移时间。当网络带宽从 10MB/s 上升到 100MB/s 时，迁移时间的理论数据分别从 31.25s 下降到 3.12s，而它的实验数据则从 32.35s 下降到 2.99s，两种数据的差距在 5% 以内，表示式(12)不仅有效地反映出迁移时间与网络带宽之间的内在关系，而且精确度在 95% 以上。

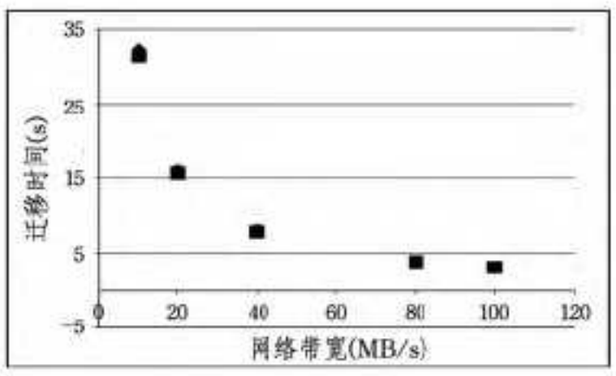

图 2 网络带宽与迁移时间

在图 3 中，方形点表示基于式(11)计算出的理论数据传输总量，而菱形点表示实际测得的数据传输总量。首先理论数据值在 327.6MB 左右波动，而实际数据在 336.8MB～340.8MB 之间波动，但是两种数据之间的差距在 5% 以内。这表示式(11)不仅有效地反映出迁移时间与网络带宽之间的内在关系，而且精确度在 95% 以上。其次，图 3 中的数据还表明，随着网络的改变，数据传输总量基本不发生变化，这是因为与虚拟机的内存大小(1GB)相比，在迁移过程中产生的脏数据量太小，从而导致总的数据传输总量独立于网络带宽。

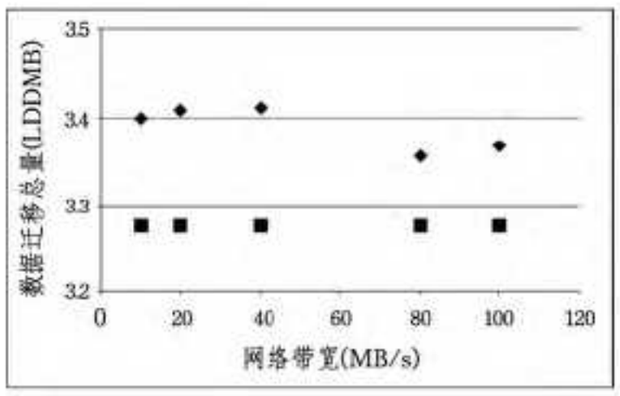

图 3 网络带宽与数据传输总量

其次，通过整 V_T 测算迁移时间和数据传输总量之间的关系，揭示 V_T 与迁移时间之间和数据传输总量之间的关系，然后基于测试数据验证式(11)和式(12)的有效性。图 4 和图 5 分别展示了测试结果。在图 4 中，V_T 从 0.4MB 增加到 40MB，迁移时间则仅在 8.126 秒附近波动，且波动的幅度在 0.5%～2% 以内。该测试结果表明，改变 V_T 并不会影响迁移时间。这与式(11)是一致的，验证了该等式的有效性。

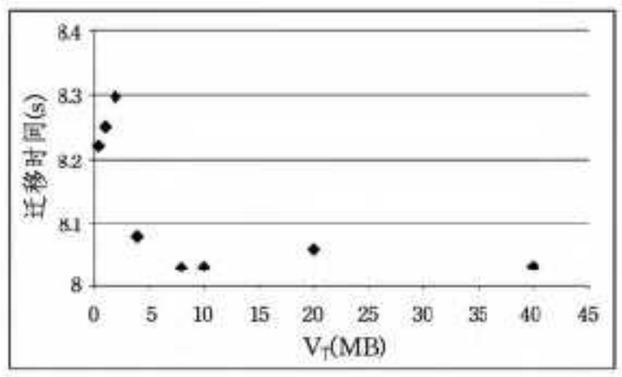

图 4 V_T 与迁移时间

在图 5 中，随着 VV_T 的改变，数据迁移总量仅在 340.2 MB 附近波动，且波动的幅度在 0.4%～1.4% 以内。该测试结果表明，改变 V_T 并不会影响数据迁移总量。这与式(12)是一致的，验证了该等式的有效性。

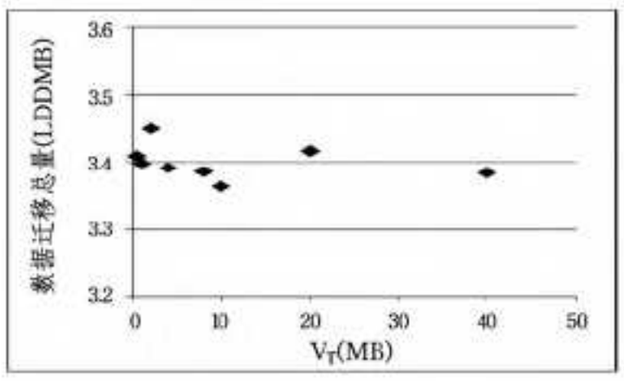

图 5 V_T 与数据迁移总量

最后，通过整 V_P 测算迁移时间和数据传输总量之间的关系，揭示 V_P 与迁移时间之间和数据传输总量之间的关系，然后基于测试数据验证式(11)和式(12)的有效性。图 6 和图 7 分别展示了测试结果。在图 6 中，V_P 从 2.4MB 减少到 0.15MB，理论迁移时间则相应地从 7.771s 增加到 7.810s，实际测试的迁移时间则从 8.032s 增加到 8.295s，并且两种数据之间的差值在 6% 以内。该测试结果表明：随着 V_P 的增加，迁移时间相应地减少但减少的幅度较小，此外也表明式(11)不仅有效地反映出 V_P 与迁移时间之间的内在关系，且精确度在 94% 以上。

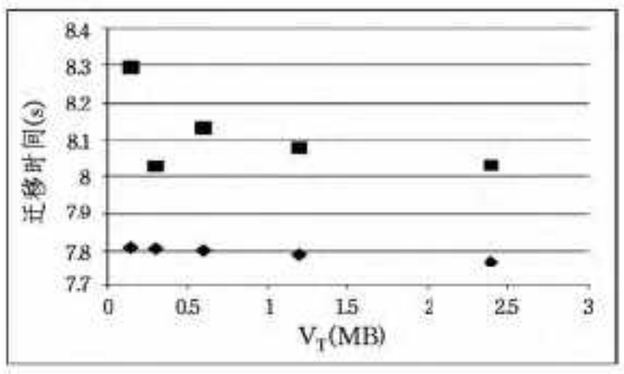

图 6 V_P 与迁移时间

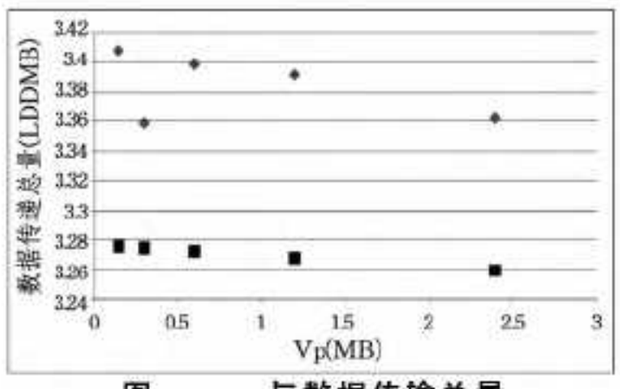

图 7 V_P 与数据传输总量

4 相关工作

在建模方面，Akoush 等[3]采用理论分析建模单虚拟机在线迁移，揭示网络速率与迁移时间和停机时间之间的相互关系。Wu 等[11]采用实验测试与资源剖析技术，建模 CPU 对迁移时间影响的实验模型，量化分析 CPU 分配对迁移时间的影

响。刘海坤等[8]结合理论分析与指数加权移动平均方法，建模单虚拟机在线迁移，量化分析网络速率和虚拟机大小等因素对迁移性能的影响。但是上述模型均是基于传统的预拷贝策略，而不是基于 qemu 中的预拷贝策略。

在页面压缩方面，金海等[6]提出一种基于内存页面特征的自适应压缩算法压缩内存数据，降低数据传输量，缩短在线迁移的时间。Gupta 等[5]采用基于哈希的指纹技术(hash-based fignerprints)计算相似页面，并使用 Xdelta 计算相似页面之间的差异，实现 delta 页面传输。Svard 等[10]使用 2 路组相联高速缓存技术(2-way set associative cache)缓存已发送页面，然后使用异或操作计算待传页面与已传页面之间的差异，实现 delta 页面传输。张翔等[1]提出一种基于位图和异或压缩的方法计算磁盘文件的变化，并采用流水线技术同步文件传输，提高在线迁移的性能。

在页面传输方面，罗英伟等[9]提出一种基于块位图(block-bitmap)的方法记录内存数据块是否被改写，以此计算需要传输的页面，减少数据重传。陈阳等[2]结合主动推送和按需复制两种传输机制，提出一种混合内存复制方法，实现脏页面的快速复制，减少数据传输量。胡亮等[7]也提出一种类似的混合内存复制方法，但是他们将 delta 页面压缩机制融入到方法中，进一步减少数据传输量。

结束语　已有的动态迁移模型均针对传统的预拷贝迁移策略。开源虚拟化平台 qemu-kvm 中的动态迁移策略与之存在一些差异，这导致已有的动态迁移模型无法有效应用于 qemu-kvm 环境下的动态迁移性能预测。为此，本文提出一种基于 qemu-kvm 的动态迁移性能模型，着重分析影响迁移性能的关键因素，并针对性地进行实验测试，通过比较实验数据与基于模型推算的理论数据之间的偏差，验证模型的有效性与精确性。实验结果表明，本文提出的性能评估模型在估算迁移时间与数据传输总量方面的精确性在 95% 以上。

参 考 文 献

[1] 张翔，霍志刚，马捷，等. 虚拟机快速全系统在线迁移[J]. 计算机研究与发展，2012，49(3)：661-668

[2] 陈阳，怀进鹏，胡春明. 基于内存混合复制方式的虚拟机在线迁移机制[J]. 计算机学报，2011，34(12)：2278-2291

[3] Akoush S，Sohan R，Rice A，et al. Predicting the performance of virtual machine migration[C]// 2010 IEEE International Symposium on Modeling，Analysis & Simulation of Computer and Telecommunication Systems(MASCOTS). 2010：37-46

[4] Aldhalaan A，Menascé D A. Analytic Performance Modeling and Optimization of Live VM Migration[C]// Computer Performance Engineering. Springer，2013：28-42

[5] Gupta D，Lee S，Vrable M，et al. Difference engine：Harnessing memory redundancy in virtual machines[J]. Communications of the ACM，2010，53(10)：85-93

[6] Jin Hai，Li Deng，Wu Song，et al. Live virtual machine migration with adaptive memory compression[C]// IEEE International Conference on Cluster Computing and Workshops，2009(CLUSTER'09). 2009：1-10

[7] Hu Liang，Zhao Gao，Xu Gao-chao，et al. HMDC：Live Virtual Machine Migration Based on Hybrid Memory Copy and Delta Compression[J]. Applied Mathematics & Information Sciences，2013，7：639-646

[8] Liu Hai-kun，Jin Hai，Xu Cheng-zhong，et al. Performance and energy modeling for live migration of virtual machines[J]. Cluster computing，2013，16(2)：249-264

[9] Luo Ying-wei，Zhang Bin-bin，Wang Xiao-lin，et al. Live and incremental whole-system migration of virtual machines using block-bitmap[C]// 2008 IEEE International Conference on Cluster Computing. 2008：99-106

[10] Svärd P，Hudzia B，Tordsson J，et al. Evaluation of delta compression techniques for efficient live migration of large virtual machines[J]. ACM Sigplan Notices，2011，46(7)：111-120

[11] Wu Yang-yang，Zhao Ming. Performance modeling of virtual machine live migration[C]// 2011 IEEE International Conference on Cloud Computing. 2011：492-499

[12] Zhu Chang-peng，Zhao Yin-liang，Bo Han，et al. Runtime support for type-safe and context-based behavior adaptation[J]. Frontiers of Computer Science，2014，8(1)：17-32

(上接第 336 页)

[9] Chu Yu. 淘宝 TFS 的 wiki[OL]. http://code.taobao.org/p/tfs/wiki/index/

[10] McAuley A J. Reliable broadband communication using a burst erasure correcting code[J]. ACM SIGCOMM Computer Communication Review，1990，20(4)：297-306

[11] Weatherspoon H，Kubiatowicz J D. Erasure coding vs. replication：A quantitative comparison[M]// Peer-to-Peer Systems. Springer Berlin Heidelberg，2002：328-337

[12] Wu L，Liu B，Lin W. A Dynamic Data Fault-Tolerance Mechanism for Cloud Storage[C]// 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies(EIDWT). IEEE，2013：95-99

[13] 林伟伟. 一种改进的 Hadoop 数据放置策略[J]. 华南理工大学学报：自然科学版，2012，40(1)：152-158

[14] 利业鞾，林伟伟. 一种 Hadoop 数据复制优化方法[J]. 计算机工程与应用，2012，48(21)：58-61

[15] 林伟伟，刘波. 基于动态带宽分配的 Hadoop 数据负载均衡方法[J]. 华南理工大学学报：自然科学版，2012，40(9)：42-47

[16] 林伟伟，贺品嘉，刘波. 云存储系统的能耗优化节点管理方法[J]. 华南理工大学学报：自然科学版，2014，42(1)：104-110

[17] Megiddo N，Modha D S. ARC：A Self-Tuning，Low Overhead Replacement Cache[C]// FAST. 2003，3：115-130

[18] 罗象宏，舒继武. 存储系统中的纠删码研究综述[J]. 计算机研究与发展，2012，49(1)：1-11

[19] Lin W K，Chiu D M，Lee Y B. Erasure Code Replication Revisited[C]// Peer-to-Peer Computing. 2004：90-97

[20] 康殿统，王文娟，杨雯. 关于 Pareto 分布的一个综合研究[J]. 河西学院学报，2008，24(2)：1-5