# 云存储中一种基于布局的虚拟磁盘节能调度方法

李建敦[1,2],彭俊杰[1],张 武[1,3]
(1.上海大学计算机工程与科学学院,上海 200072; 2.上海电机学院电子信息学院,上海 201306;
3.上海大学高性能计算中心,上海 200072)

**摘 要:** 在云存储中,如何有效地调度用户请求到目标磁盘以实现绿色节能存储是一个热点问题.鉴于云存储对节能调度算法提出的新要求,如请求响应时间敏感与对动态优化的限制等,本文提出了一种基于布局的虚拟磁盘节能调度方法.该方法将磁盘阵列动态划分为工作区与就绪区,以工作区为主向用户分发资源,并以未连接虚拟机的虚拟磁盘为单位,根据实时负载情况对虚拟磁盘布局进行动态优化.实验结果表明,这种方法不仅能够降低磁盘阵列的能耗,而且能够有效地缓解响应时间延长的问题,还能够使虚拟磁盘布局达到更高的负载均衡水平.

**关键词:** 云计算; 云存储; 虚拟磁盘; 节能调度; 负载均衡
**中图分类号:** TP301 **文献标识码:** A **文章编号:** 0372-2112 (2012) 11-2247-08
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.11.017

# A Layout-Based Energy-Aware Approach for Virtual Disk Scheduling in Cloud Storage

LI Jian-dun[1,2], PENG Jun-jie[1], ZHANG Wu[1,3]
(1. *School of Computer Engineering & Science, Shanghai University, Shanghai* 200072, *China*;
2. *Electronics Information School, Shanghai Dianji University, Shanghai* 201306, *China*;
3. *High Performance Computing Center, Shanghai University, Shanghai* 200072, *China*)

**Abstract:** In cloud storage, how to effectively schedule user's request to specific disk becomes to be a hot topic. Considering the new requirements related to energy-aware scheduling algorithms proposed by cloud storage, e.g. sensitivity to request's response time and limited space left for dynamic optimization, this paper proposes a layout-based scheduling approach for virtual disks. The disk array is dynamically partitioned into two sets, working set and standby set respectively using this approach, and requests are mainly addressed by working set. Additionally, according to real-time workloads, the layout of virtual disks is optimized with standalone virtual disk as a unit. Results of experiments show that this approach can cut down the power of disk array, effectively alleviate the problem of prolonged response time, and achieve higher level of load balancing.

**Key words:** cloud computing; cloud storage; virtual disk; energy-aware scheduling; load balancing

## 1 引言

云计算是一种新的分布式计算模式,它能够充分利用规模效应,以请求即响应的方式向用户提供几乎所有与 IT 相关的服务,且只按用量收费[1,2].它改变了硬件设计与供求的方式,使软件因"软件即服务"而变得更有吸引力[3].它能够较大程度地减少能源的消耗,提高资源的利用率,以弹性高可扩展的方式为用户提供服务[4~6].

在云对外提供的服务中,存储服务[7~9]正受到越来越多研究人员的关注.其中,如何有效地调度用户的存储请求到目标节点以实现绿色节能存储是当前的一个热点问题.同时,除了降低能耗这个要求外,云存储对调度算法还有效率上的要求,包括用户效率(如请求响应时间)与系统健壮水平(如负载均衡)两个方面.

一般来讲,云存储服务是通过广域网交付用户使用的,网络延迟的存在使用户对于存储请求的响应时间较敏感,而传统的节能调度算法又往往是以牺牲请求的响应时间为代价的,因此如何缓解节能的负面效应,尽可能地不延长或少延长用户的等待时间是节能云存储对

收稿日期:2011-11-21;修回日期:2012-05-16
基金项目:国家自然科学基金(No.61103054);教育部博士点基金(No.20113108120022);上海市重点学科(No.J50103);上海市科委重点项目(No.11510500300)

调度算法提出的新要求;为了实现系统负载的均衡,常规的调度算法都包括对资源分配布局的更新与动态优化,但是在云存储场景中,当虚拟磁盘未断开到虚拟机的连接时,虚拟磁盘的动态迁移过程很有可能直接影响用户的 I/O 数据流,因此面向负载均衡的动态优化就受到了限制,如何对资源分配布局进行动态优化就成为云存储对调度算法提出的又一个新要求.

然而,已有的研究主要是从各个方面(重塑 I/O 请求、模式切换等)对磁盘的节能做了优化,而未将用户及系统性能的要求完全考虑在内.因此,虚拟磁盘的节能问题,即如何在保证用户性能及系统健壮水平基础上节能降耗,是个值得研究的问题.

针对云存储对节能调度算法提出的新要求及当前研究的不足,本文提出了一种基于布局的虚拟磁盘调度方法.该方法将磁盘阵列动态划分为工作区与就绪区,以工作区为主向用户分发资源,并以未连接虚拟机的虚拟磁盘为单位,根据实时负载情况对虚拟磁盘布局进行动态优化.

## 2 相关工作

目前,国外的许多知名大学都开展了对节能云存储的研究工作.

科罗拉多大学波尔得分校等[10~13]从磁盘的读写请求入手,整合与优化了 I/O 请求队列,使磁盘请求呈现堆效应,便于磁盘控制器集中处理,为磁盘的多时间段、长时间休眠提供了便利.与此类似,斯坦福大学 E Y Chung 等[14]通过滑动窗口与二维线性插值,为单个磁盘的节能优化提出了一种自适应的方法.然而由于优化目标是单个磁盘,对多磁盘间的协调访问指导意义不大.

罗格斯大学等[15~19]通过磁盘冗余,为磁盘的节能问题提出了相应方案.然而由于冗余客观上增加了空间上的开销(>100%),再加上数据自有的备份机制等,虽然提高了数据的可用性,但是却显著增加了成本,降低了磁盘空间的利用率,节能量有限.

加州大学伯克利分校 M Armbrust 等[1]对磁盘的节能管理做了深入细致的定量分析,并指出为了达到最小化能耗的目的,应当在磁盘休眠 2s 后将磁盘置于休眠模式.受此启发,本文采用主动方式来休眠磁盘,以达到节能降耗的目的.

伊利诺伊大学厄本那香槟分校等[20~22]充分利用磁盘就绪模式下,Cache 仍可用的特性,有效地提高了 Cache 的命中率,从而让磁盘持续保持休眠模式,以达到深度节能的目标.然而由于 Cache 的命中率与具体的应用密切相关,而本文研究的场景只与用户有关,因此方法的应用效果受到了限制.

加州大学圣克鲁兹分校等[23,24]在收到每个磁盘访问请求后都会启动一个计时器,计时器超过预设的空闲阈值时,就会触发休眠指令,将目标磁盘置于休眠模式.然而目标阈值的设定存在着潜在的因空闲模式过长(阈值太大)或模式切换过于频繁(阈值太小)而使能耗升高的问题,因此本文采用主动休眠的方法.

斯坦福大学等[25~29]能够根据应用的访存模式,大胆地提前休眠磁盘.方法基于应用的访存历史统计或对应用的实时监测,来获得应用的特定模式.然而一方面模式的生成,需要额外的开销,另一方面,如果模式失效,尤其是那些弱模式的应用,会对应用的执行效率造成明显影响.再者,与应用密切相关的属性与本文基于大规模用户共享的场景不尽符合.

在虚拟化环境下,卡尔斯鲁厄大学[30]也提出了一个多层架构来规划能源与监控资源,并且能够将节能限制量化到客户端操作系统.然而架构只考虑磁盘的两种模式(即活动与空闲),节能效果不足.

通过有效地重塑虚拟机的 I/O 操作,尤其是写操作,以及虚拟化层上的休眠前缓冲区的提前释放(flush),亚利桑那大学 L Ye 等[31]最大化了磁盘的休眠时间,从而节约了大量能源.然而由于方法的应用场景基于虚拟机与磁盘,与虚拟机紧耦合的关系,属于系统盘的调度范畴,与独立于虚拟机的数据盘的调度不尽相同.

由此可以看出,云存储的节能调度问题目前已经得到了相当一部分研究人员的关注,而且主要集中在三个层面上,即虚拟机访问物理内存的调度,自带虚拟磁盘(系统盘)的调度与虚拟数据盘的调度.由于物理内存的内在要求(不支持低功耗模式),节能的空间并不是很大;同时,由于虚拟机自带的磁盘空间较小,且与虚拟机紧耦合,会随虚拟机的注销而消亡,因此用户一般不会将数据存储到系统盘,而往往倾向于申请一块独立的虚拟磁盘.由于这种虚拟磁盘可方便地与任意虚拟机及客户操作系统进行耦合与解耦合,数据始终独立,因此受到了用户认可,也是节能云存储关注的重点.

## 3 虚拟磁盘的节能调度

### 3.1 虚拟磁盘

虚拟磁盘(Virtual disk),是云计算数据中心对外提供的一种远程存储服务,支持与虚拟机实例(VM instance)的绑定,是用户数据的集中存储点.通过云计算提供的服务,客户端就可以大大地“瘦身”,只需要一组标准输入输出设备和到数据中心的网络连接就可以正常工作,从而有效地降低了用户的投入成本.

### 3.2 磁盘的工作模式

现有磁盘(IDE、SCSI、SATA 等)一般都支持多种工

作模式,归纳起来基本上有以下四种[32]:

(1)活动模式(Active):磁盘处于全速工作状态,盘片在高速运转,能够在最短时间内响应用户的存储请求;

(2)空闲模式(Idle):盘片保持旋转状态,但磁头臂停止运转,其他多数电子器件处于关闭状态,能耗较活动模式稍低;

(3)就绪模式(Standby):磁盘停止了运转,处于低功耗状态中;

(4)休眠模式(Sleeping):磁盘处于关闭状态,功耗达到了最小值.

上述四种模式中,活动模式与空闲模式是常规模式,能够在最短时间内完成对任务的响应;而就绪模式与休眠模式需要先恢复到常规模式后,才能对任务完成响应.而恢复时间会随磁盘的不同而有所差异,一般是在秒级.对于休眠模式,需要硬重启或软重启指令以恢复到就绪模式,最后到常规模式.另外由表 1 可以看出,休眠模式的节能水平与就绪模式基本持平,而恢复到常规模式却要先经过就绪模式,因此为了提高任务的响应时间,本文选用就绪模式来节能.

**表 1　磁盘工作模式[32]**

| 模式 | 磁头/盘片 | 缓冲区 | 功率(W) |
|---|---|---|---|
| 活动 | 运转 | 可用 | 6.19 |
| 空闲 | 运转 | 可用 | 4.60 |
| 就绪 | 停止 | 可用 | 0.79 |
| 休眠 | 停止 | 不可用 | 0.79 |

### 3.3　虚拟磁盘节能调度的目标

鉴于云存储对节能调度算法提出的新要求,我们选用了三个调度指标,即请求响应时间、节能量与负载均衡来衡量调度性能的优劣.

响应时间指从用户提交虚拟磁盘请求起,到用户可以将虚拟磁盘绑定到 VM 之时止的这段时间.如果目标磁盘工作在常规模式下,那么这段时间较短(系统调度时间与划分虚拟磁盘时间之和);然而,如果目标磁盘处于就绪模式下,那么响应时间就较长(原有时间加上唤醒就绪磁盘时间).

节能量的多少是衡量节能调度优劣的直接指标,可以用所有磁盘在就绪模式上运行的时间总和或占总体运行时间的百分比来表征.

负载均衡是作业调度中一个重要的衡量指标,要求载体上的负载平均分布.在本文中,由于节能与负载均衡指标的共存,牵涉到了性能与节能这一对矛盾,这里通过四个范式来阐述这对矛盾体.

假定系统中有磁盘(以下简称节点)$N$ 个,每个最多能够支持虚拟磁盘 $M$ 个.全集 $U$ 由这 $N$ 个节点组成.子集 $S$ 由当前处于就绪模式的节点组成.这里我们引入两个整型函数来表征这些节点的状态,其中函数 $F(i)$表示节点 $i$ 上的负载,$Z(i)$表示节点 $i$ 的工作模式.$F(i)$的值域为$[0,M]$,而 $Z(i)$的值域为$\{0,1\}$(其中"0"表示常规模式,"1"表示就绪模式).

**范式 1**　全集 $U$ 中的每个元素都满足 $F(i)>0$,或者 $Z(i)=1$,即

$$\prod_{i\in U}(F(i)+Z(i))>0 \tag{1}$$

**范式 2**　在满足范式 1 的基础上,补集$\overline{S}$的规模$|\overline{S}|$达到最小值,即

$$M\cdot(|\overline{S}|-1)<\sum_{i\in\overline{S}}F(i)\leqslant M\cdot|\overline{S}| \tag{2}$$

**范式 3**　在满足范式 1 与范式 2 的基础上,负载均匀地分布在补集$\overline{S}$中,即

$$\sum_{i\in\overline{S},j\in\overline{S},i\neq j}(|F(i)-F(j)|)+\sum_{k\in S}(|Z(k)-1|)=0 \tag{3}$$

有时补集$\overline{S}$中的负载并未达到严格意义上的均衡,但由于虚拟磁盘的原子性,不宜再优化,因此我们提出了范式 4.

**范式 4**　在满足范式 1 与范式 2 的基础上,负载在补集$\overline{S}$中的分布不宜再均衡,即

$$\sum_{i\in\overline{S},j\in\overline{S},i\neq j}((|F(i)-F(j)|)(|F(i)-F(j)|-1))+\sum_{k\in S}(|Z(k)-1|)=0 \tag{4}$$

## 4　一种基于布局的虚拟磁盘节能调度方法

我们首先来考察两个经典的调度方法,轮盘赌(Round-robin,RR)[33]与贪婪法(Greedy)[34].

轮盘赌是面向负载均衡调度众多方法中最简单、最常用的方法之一.它按照环形顺序依次为各节点分配等量的任务,所有节点同等看待,无优先级之分.对于负载均衡指标来说,这种方法简单有效,方便实用;然而对于节能指标,却无能为力.

贪婪法是节能调度中最简单的方法之一,它将工作持续分配到第一个节点上,直到剩余空间不足为止才往下一个节点分配.而其它节点在空闲一段时间后,就会自动进入就绪模式.这种方法对于节能目标较有效,但是由于唤醒处于就绪模式的节点需要一段时间的等待,因此会延长响应时间,另外负载在节点间的分配也达不到均衡的状态.

由此我们可以看出,在传统的节能存储方法中,至少有两点需要优化:①当目标节点处于就绪模式时,唤醒过程需要一定时间,如何减少调度到就绪模式节点的可能性是个值得优化的问题;②负载在节点间未能达到均衡.针对这两个优化点,我们提出了一种基于布局的优化调度方法,它由预处理、资源选择与后处理三

个部分组成.

### 4.1 预处理

云计算之所以受到如此多企业、高校与研究机构的关注,一个重要原因是云计算的弹性机制.弹性机制可从云服务的前端和后台两个方面来理解.

①前端:用户可动态变更服务请求,如虚拟机个数、虚拟磁盘的挂载与虚拟网络的规模等;

②后台:云数据中心资源池的规模可以随着用户需求规模的实时变化而进行动态伸缩.而这里的动态伸缩就是通过节点的休眠与唤醒来实现的.

而资源池伸缩的时机直接影响任务的执行效率,尤其在云环境中.资源池的收缩时机目前一般采用分布式与被动式相结合的方式,即由各个节点上设置的空闲阈值来被动触发;而在资源池扩张的目标选择上也采用被动方式,当所有处于活动模式的节点均不能完成当前任务请求时,自动调度任务请求到就绪节点,而未考虑响应时间的滞后问题.

不同于这种被动的方式,本文采用主动休眠的方式.首先设定目标区间[$BL$, $BU$]作为云存储资源池的伸缩依据.目标区间反映了任务执行效率与节能目标之间的权衡,其值可依照历史统计值数据来设定.当资源池的当前剩余服务能力 $LC$ 发生上溢时,就主动休眠一些节点,以保证服务能力落在目标区间内;反之,当发生下溢时,就主动去唤醒一些处于就绪模式的节点.如此一来,在保证系统弹性服务模式的基础上,既克服了被动方式中因空闲等待所造成的资源浪费问题,也大大降低了请求响应时间因唤醒节点而造成的延迟问题.同时,目标区间的设定将虚拟资源池划分为两个动态子集 $S$ 与$\overline{S}$,其边界随着休眠/唤醒指令而发生偏移.

### 4.2 资源选择

对于负载均衡的调度目标,经典算法 RR 能够较好的完成任务;然而它未考虑节能目标,不能直接应用.本文采用最小负载优先法(Min-load-first)来选择节点为请求分发资源,目的是达到较高的负载均衡水平.在具体的选择过程中,首先在工作区$\overline{S}$中考察,如果能够得到满足,那么将会以最少的请求响应时间向用户分发资源.如果$\overline{S}$中剩余能力不足以向当前请求提供服务,那么就要向就绪区 $S$ 请求资源,具体步骤如下:

**Step 1** 比较请求规模 $R$ 与最大虚拟磁盘容量 $M$ 之间的大小关系,如果 $R > M$,那么当前数据中心不支持,退出(-1);否则,下一步.

**Step 2** 查看工作区中最后一个节点,即节点 $Mark-1$,如果满足 $M - F(OT[Mark-1]) \geqslant R$,就选定节点 $Mark-1$,并更新负载水平 $F(OT[Mark-1])$、当前服务能力 $LC$ 及有序表 $OT[N]$,退出($Mark-1$);否则,下一步.

**Step 3** 选定就绪区中的第一个节点 $Mark$,并唤醒,更新 $Z(OT[Mark])$、$F(OT[Mark])$、$LC$、$OT[N]$与 $Mark$,将 $Miss$ 增 1,退出($Mark$).

可以看出,资源选择步骤非常简单,不需要遍历所有节点,甚至不需要遍历工作区中的每个节点,因此能够最大程度地减少因调度而产生的延迟,有效地提高了用户请求的响应效率.当然,算法步骤简化的基础是有序表 $OT[N]$的维护,然而维护是渐进式的,主要是对元素位置的更新,包括节点删除与节点插入操作,因此时间复杂度为 $O(N)$.

当用户不再需要虚拟磁盘时,要对所占用的虚拟磁盘进行释放操作,而数据中心要进行相应的资源回收操作,包括虚拟磁盘所在节点负载 $F$ 的更新,当前剩余能力 $LC$ 的更新及有序表 $OT[N]$的更新等.

在资源选择或释放之后,原有的面向三个指标平衡的布局就被打破了,因此需再针对三个指标进行后处理,以达到新的平衡.

### 4.3 后处理

在资源选择或资源释放之后,要对数据中心当前的资源分发布局进行调整,以满足三个调度指标的要求,具体有:

①目标区间的设置是否合理?(节能与响应时间之间的平衡问题)

②当前布局是否满足各范式的要求?(节能与均衡负载之间的平衡问题)

③当前剩余服务能力是否落在目标区间内?(节能与响应时间之间的平衡问题)

首先来看目标区间的更新操作,其依据是请求在$\overline{S}$中失效的次数与频率.如果虚拟磁盘请求在一个周期内(如一个工作日或一周),发生的失效次数较多且较频繁,那么就需要将目标区间向上调整,以适应短期内较多、较密集的资源请求;反之失效过少时,说明区间设定的资源边际较大,当前的请求较少且较稀疏,需要将其往下调整.

接下来看目标区间的维护问题.对于下溢,只需要在$\overline{S}$中启动一个就绪节点,再更新相关变量就可以了;然而对于上溢来说,可能存在这样的情况,即不能通过休眠一个或多个常规节点来使 $LC$ 重新被目标区间捕获,原因是当前所有的常规节点上均有未完成的任务在运行.也就是说,对于空闲节点来说,可以通过直接休眠指令将目标节点置于低能耗模式上,从$\overline{S}$转到 $S$ 中去,以达到节能目标;但是对于有任务在运行的活动节点,直接休眠不可行,需要先考虑迁移.

然而在迁移之前,我们需要注意到,当虚拟磁盘已连接到虚拟机,那么迁移不可行.因为此时的迁移操作

直接影响用户的正常使用,尤其是当磁盘有大量数据时,另一方面,连接或断开虚拟机是用户的权利,数据中心不应干预.因此,迁移只应发生在虚拟磁盘未连接或已经断开虚拟机时才合理.

在后处理中,我们首先将这些未连接的磁盘调度入就绪区,再进行范式达标情况的检查,因此,后处理不需要对范式2(或范式4)进行检验,因为即便当前布局未达到范式2的要求,也不能通过迁移来操作.对于范式1来说,只需判断没有空闲节点即可.

考虑到引起布局变更的缘由有两种,即资源分发与资源回收,而资源分发只可能引起下溢,只有资源回收才会引起资源过剩,从而可能发生上溢.因此,范式达标的情况也只在发生上溢时才去检查并处理.

后处理涉及多个方面,处理顺序如下:

①目标区间的更新;

②面向未连接虚拟磁盘的迁移;

③面向目标区间的上溢处理:当前剩余服务能力面向目标区间发生上溢时(由于资源释放、未连接虚拟磁盘的连接或迁移),可通过范式1的检查将空闲节点休眠;

④面向目标区间的下溢处理:当前剩余服务能力发生下溢时(由于资源请求),需要唤醒就绪区中的节点.

### 4.4 算法小结

针对云存储对节能调度算法提出的新要求,以及已有算法在虚拟磁盘节能调度上的不足,提出了一种基于布局的虚拟磁盘节能调度方案.算法的主要操作是对未连接虚拟机的虚拟磁盘的迁移操作,而操作需要考察每个(共 $T$ 个)虚拟磁盘的连接情况,如果未连接,那么就从 $OT[N]$ 中最后一个元素开始查找可行的迁移目,因此算法的最大时间复杂度为 $O(T\cdot N)$.

## 5 实验及讨论

为了说明本文提出方法的作用,我们做了一些仿真实验.实验平台由4台PC机(HP Compad dc 7900)构成,每台有4核(Intel(R) Core(TM) 2 Quad CPU Q8400 2.66GHz),4GB内存与320GB磁盘(ST3320418AS),通过100Mbps以太网互联.我们选其中一台PC作为云调度节点,而其余的($N=3$)作为虚拟磁盘执行节点.分布式存储架构采用AOE(ATA Over Ethernet),操作系统采用Ubuntu Server 9.10 AMD64.

我们采用Eucalyptus[35]建立基础云平台.不失一般性,虚拟磁盘请求工作流服从高斯分布 $N(2400,1550)$,请求容量为75GB($M=4$).磁盘休眠是通过hdparm指令实现的,虚拟磁盘的迁移采用快照(Snapshot)来实现.经过多次实验,得到的最佳目标区间为[150GB,300GB].

我们分别用轮盘赌、贪婪法与本文基于布局的优化方法来调度工作流,相关数据与比较见图1至图3以及表2.图中,横坐标代表时间,纵坐标代表节点上的负载.负载用百分比表示,其中 -25% 表示当前节点负载为0且处于就绪模式.

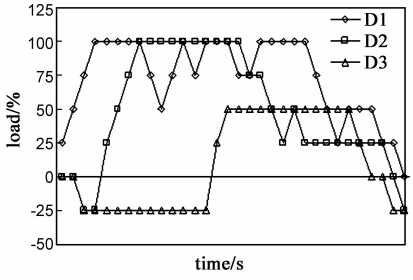

图1 运用贪婪法调度

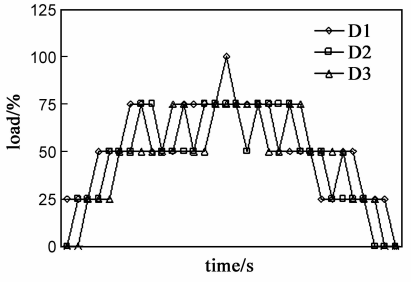

图2 运用轮盘赌调度

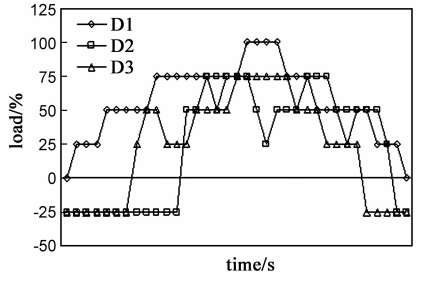

图3 运用本文方法调度

表2 详细数据与比较

| 指标 | 贪婪法 | 轮盘赌 | 本文方法 |
|---|---|---|---|
| 节能量(%) | 25.6 | 0 | 32.8 |
| 响应时间(s) | 8.69 | 4.44 | 5.25 |
| 范式4(s) | 1508 | 660 | 2846 |
| 范式2(s) | 2938 | 660 | 3197 |
| 范式1(s) | 4512 | 4626 | 6595 |

在开始阶段,Greedy算法持续将负载置于节点D1上,直到D1饱和(100%).当新的虚拟磁盘请求到达时,不得以才将业已休眠的节点D2唤醒.同理,当D2亦饱和时,再去唤醒处于就绪模式的节点D3.在结尾阶段,当节点上的负载因虚拟磁盘请求的注销而降到0时,节点再次通过休眠阈值的控制而切换到就绪模式,如图1所示.总之,Greedy算法在负载均衡方面不理想,尤其是在开始阶段.

与Greedy算法不同的是,RR算法总是用下一个可调度的节点来为虚拟磁盘请求分发资源,因此各计算节点的负载变化曲线像金字塔(如图2),很好地均衡了负载.然而RR算法不将节点置于就绪模式(负载没有 -25%),因此不节能.

利用本文所提出的基于布局的优化方法来调度工作流如图3所示.在开始阶段,直接将节点D2与D3休眠,而只保留D1来为虚拟磁盘请求提供服务.当D1上

的负载达到一定值(75%)时(此时未有虚拟磁盘请求到达),由于所剩服务能力不足而唤醒就绪节点 D2.在资源分发节点的选择上总是选择当前负载最小的节点.在结尾阶段,直接将过剩的节点休眠以节能.从图中可以看出,除去就绪的节点,负载得到了较好地均衡.

图 4 显示了不同方法在响应时间上的差异,Greedy 算法无疑是最差的,而本文的优化方法比 RR 算法稍差.

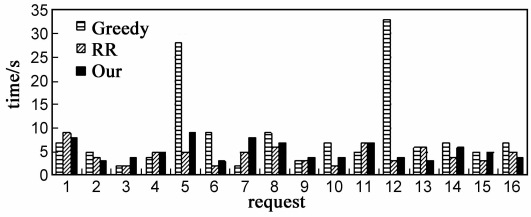


**图4 响应时间的比较**

各范式的达标情况如图 5 所示.从中可以看出,本文的优化方法较优,能够在更大的时间跨度内达到更高的范式要求;然而,由于虚拟磁盘的特殊性,迁移受到了较大地限制,致使优化方法在很长的时间内所能达到的范式水平较低.

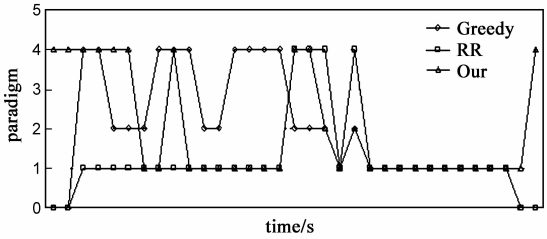


**图5 范式达标情况**

不同方法在不同衡量指标下的具体数据如表 2 所示.接下来,再从具体指标出发,对不同方法的实验结果进行深入的比较与讨论.

(1)响应时间:在这一指标的对比中,Greedy 算法是最差的.为了最大可能地降低能耗,它将任务尽可能地放到磁盘 D1 上,而其它的节点在空闲一段时间后都自动进入就绪模式.当 D1 不能满足需求时,才不得不去唤醒 D2.而唤醒过程需要额外的时间开销,因此响应时间就被延长了,图 4 很好地说明了这点.另外,与 RR 算法相比,优化方法的响应时间稍长,原因在于算法的时间复杂度.

(2)节能量:RR 算法不节能,而 Greedy 算法与本文方法都节约了大量的能源(使磁盘较长时间地运行在就绪模式上),分别是 25.6%与 32.8%.在 Greedy 算法的开始阶段,空闲节点(D2 与 D3)都处于常规模式,直到过了休眠阈值(300s)才自动休眠.在优化方法中,在期望边界之外的节点都直接被置于就绪模式,只留下一个节点(D1)来分发资源.然而,当第三个虚拟磁盘请求到达时,本文方法直接唤醒了一个额外的节点(D2).在这种情况下,第四个请求就被调度到节点 D2 上去执行了.在这一时间点上,工作区的规模比 Greedy 算法要稍大,致使其状态达不到范式 2 的要求;然而它却显著提高了第五个请求的响应时间.同时,通过将未连接 VM 的虚拟磁盘迁移到就绪区,调度器为工作区腾出了额外的空间,同时将未被使用的资源集中休眠,从而得到了可观的节能量,详见表 2.

(3)负载均衡:显然,就一般的负载均衡指标而言,RR 算法是最优的(如图 2 所示).本文的优化方法其次,而 Greedy 算法最差.但是,对于本文所提出的综合负载均衡指标来说,却是另外一种情况.从图 5 可以看出,RR 算法的曲线大部分时间都处于范式 1 或更低的位置,其中只有小部分达到了范式 4 的要求(当请求的规模接近系统最大服务规模时).再看 Greedy 算法,它的曲线基本上均匀地分布在各个范式上.值得注意的是,RR 算法与 Greedy 算法在开始阶段与结束阶段较相似,那就是都未对空闲节点作休眠处理,结果达不到范式 1 的要求,而在中间阶段却都能达到较高的范式水平.与此形成对比的是,本文方法横跨多个范式,而且高范式达标情况主要集中在两端,而不是中间.虽然两种情况看似区别不大,但对于高斯分布来说,两端时间刻度的跨度要比中间大得多,而且由于两端是请求较稀疏的时间段,也就是最需要节能优化的时间段,因此,与其他两种方法相比,本文方法在复合负载均衡指标上优势明显(如表 2).

## 6 结论

通过分析云存储对节能调度算法提出的新要求,结合现有研究中存在的一些效率问题,本文提出了一种基于布局的虚拟磁盘节能调度方法.该方法将磁盘阵列动态划分为工作区与就绪区,以工作区为主向用户分发资源,并以未连接虚拟机的虚拟磁盘为单位,根据实时负载情况对虚拟磁盘布局进行动态优化.实验结果表明,这种方法不仅能够降低磁盘阵列的能耗,而且能够有效地缓解响应时间延长的问题,还能够使虚拟磁盘布局达到更高的负载均衡水平.

然而,本文的工作尚存不足,表现在对节能量的度量上.文中对节能优化前后的比较只是通过低功耗模式所占的时间百分比来表达的,未得出精确的功率差异.这其中隐含了一个问题,那就是目标节点在模式切换时能耗激增对全局节能形势的影响,因此我们将此作为本文下一步的工作.

**参考文献**

[1] M Armbrust, A Fox, R Griffith, A D Joseph, R H Katz, A Konwinski, G Lee, D A Patterson, A Rabkin, I Stoica, M Zaharia.

Above the Clouds:A Berkeley View of Cloud Computing[R]. Berkeley,California,USA:University of California at Berkley, 2009.1 – 23.

[2] 李建江,崔健,王聃,严林,黄义双.MapReduce 并行编程模型研究综述[J].电子学报,2011,39(11):2635 – 2642.
LI Jian-jiang,CUI Jian,WANG Dan,YAN Lin,HUANG Yi-shuang.Survey of MapReduce parallel programming model[J]. Acta Electronica Sinica,2011,39(11):2635 – 2642.(in Chinese)

[3] I Foster,Y Zhao,I Raicu,S Lu.Cloud computing and grid computing 360-degree compared[A].Proceedings of Grid Computing Environments Workshop[C].Washington,DC: IEEE Computer Society,2008.1 – 10.

[4] J D Li,W Zhang,J J Peng,Z Lei,Y Lei.A carbon 2.0 framework based on cloud computing[A].Proceedings of 2010 International Conference on Information Systems[C].Washington, DC:IEEE Computer Society,2010.153 – 158.

[5] A Berl,E Gelenbe,M D Girolamo,G Giuliani,D H Meer,M Q Dang,K Pentikousis.Energy-efficient cloud computing[J].The Computer Journal,2010,53(7):1045 – 1051.

[6] F S Chu,K C Chen,C M Cheng.Toward green cloud computing[A].Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication[C]. New York:ACM,2011.1 – 5.

[7] K D Bowers,A Juels,A Oprea.HAIL:a high-availability and integrity layer for cloud storage[A].Proceedings of the 16th ACM Conference on Computer and Communications Security [C].New York:ACM,2009.187 – 198.

[8] 吴吉义,傅建庆,平玲娣,谢琪.一种对等结构的云存储系统研究[J].电子学报,2011,39(5):1100 – 1107.
WU Ji-yi,FU Jian-qing,PING Ling-di,XIE Qi.Study on the P2P cloud storage system[J].Acta Electronica Sinica,2011,39 (5):1100 – 1107.(in Chinese)

[9] 孙大为,常桂然,李凤云,王川,王兴伟.一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法[J].电子学报, 2011,39(8):1824 – 1831.
SUN Da-wei,CHANG Gui-ran,LI Feng-yun,WANG Chuan, WANG Xing-wei.Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference[J].Acta Electronica Sinica,2011,39(8):1824 – 1831.(in Chinese)

[10] D Colarelli,D Grunwald.Massive arrays of idle disks for storage archives[A].Proceedings of the 2002 ACM/IEEE Conference on Supercomputing[C].Washington,DC:IEEE Computer Society,2002.1 – 11.

[11] E Pinheiro,R Bianchini.Energy conservation techniques for disk array-based servers[A].Proceedings of the 18th Annual ACM International Conference on Supercomputing[C].New York:ACM,2004.68 – 78.

[12] Q Zhu,Z Chen,L Tan,Y Zhou,K Keeton,J Wilkes.Hibernator:helping disk arrays sleep through the winter[A].Proceedings of the 20th ACM Symposium on Operating Systems Principles[C].New York:ACM,2005.177 – 190.

[13] D Narayanan,A Donnelly,A Rowstron.Write off-loading: practical power management for enterprise storage[A].Proceedings of the 6th USENIX Conference on File and Storage Technologies[C].New York:ACM,2008.253 – 267.

[14] E Y Chung,L Benini,A Bogliolo,Y H Lu,G D Micheli.Dynamic power management for nonstationary service requests [J].IEEE Transactions on Computing,2002,51(11):1345 – 1361.

[15] E Pinheiro,R Bianchini,C Dubnicki.Exploiting redundancy to conserve energy in storage systems[A].Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems[C].New York:ACM,2006.15 – 26.

[16] D Li,J Wang.Eeraid:energy efficient redundant and inexpensive disk array[A].Proceedings of the 11th ACM SIGOPS European Workshop[C].New York:ACM,2004.29 – es.

[17] X Yao,J Wang.Rimac:A novel redundancy-based hierarchical cache architecture for energy efficient,high performance storage systems[A].Proceedings of 1st ACM SIGOPS/EuroSys European Conference on Computer Systems[C].New York:ACM,2006.249 – 262.

[18] K M Greenan,D D E Long,E L Miller,T J E Schwarz,J J Wylie.A spin-up saved is energy earned:Achieving power-efficient,erasure-coded storage[A].Proceedings of 4th Workshop on Hot Topics in System Dependability[C].Berkeley, CA:USENIX Association Berkeley,2008.1 – 6.

[19] C Weddle,M Oldham,J Qian,A A Wang.Paraid:a gear-shifting power-aware raid[J].ACM Transactions on Storage (TOS),2007,3(3):13:1 – 13:33.

[20] Q Zhu,F M David,C F Devara,Z Li,Y Zhou,P Cao.Reducing energy consumption of disk storage using power-aware cache management[A].Proceedings of the 10th International Symposium on High Performance Computer Architecture[C]. Washington,DC:IEEE Computer Society,2004.118 – 129.

[21] S D Carson,S Setia.Analysis of the periodic update write policy for disk cache[J].IEEE Transactions on Software Engineering,1992,18(1):44 – 54.

[22] J C Mogul.A better update policy[A].Proceedings of the USENIX Summer Technical Conference[C].Berkeley,CA: USENIX,1994.99 – 111.

[23] F Douglis,P Krishnan,B N Bershad.Adaptive disk spin-down policies for mobile computers[A].Proceedings of the 2nd Symposium on Mobile and Location-Independent Computing [C].Berkeley,CA:USENIX,1995.121 – 137.

[24] R Golding,P Bosch,C Staelin,T Sullivan,J Wilkes.Idleness is not sloth[A].Proceedings of the USENIX Winter Technical Conference[C].Berkeley,CA:USENIX Association Berke-

ley,1995.201 – 212.
[25] E Y Chung,L Benini,G D Micheli. Dynamic power management using adaptive learning tree[A]. Proceedings of the 1999 IEEE/ACM International Conference on Computer-Aided Design[C]. Washington,DC: IEEE Computer Society,1999.274 – 279.
[26] C H Hwang, A C H Wu. A predictive system shutdown method for energy saving of event-driven computation[J]. ACM Transactions on Design Automation of Electronic Systems,2000,5(2):226 – 241.
[27] M B Srivastava, A P Chandrakasan, R W Brodersen. Predictive system shutdown and other architectural techniques for energy efficient programmable computation[J]. IEEE Transactions on Very Large Scale Integration Systems,1996,4(1):42 – 55.
[28] A E Papathanasiou,M L Scott. Energy effcient prefetching and caching[A]. Proceedings of the USENIX Annual Technical Conference[C]. Berkeley, CA: USENIX Association Berkeley,2004.255 – 268.
[29] Y H Lu, E Y Chung, T Simunic, L Benini, G D Micheli. Quantitative comparison of power management algorithms [A]. Proceedings of the Conference on Design, Automation and Test in Europe[C]. Washington,DC: IEEE Computer Society,2000.20 – 26.
[30] J Stoess,C Lang,F Bellosa. Energy management for hypervisor-based virtual machines[A]. Proceedings of the USENIX Annual Technical Conference[C]. Berkeley, CA: USENIX Association Berkeley,2007.1 – 14.
[31] L Ye,G Lu,S Kumar,C Gniady,J H Hartman. Energy-efficient storage in virtual machine environments[A]. Proceedings of the 2010 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments[C]. New York: ACM,2010.75 – 84.
[32] Barracuda 7200.12 Serial ATA. Seagate Product Manual[S]. 2010.
[33] Round-robin (RR) on Wikipedia[R/OL]. http://en.wikipedia.org/wiki/Round-robin_scheduling.2012.
[34] Greedy on Wikipedia[R/OL]. http://en.wikipedia.org/wiki/Greedy_algorithm.2012.
[35] D Nurmi,R Wolski,C Grzegorczyk,G Obertelli,S S oman,L Youseff, D Zagorodnov. The eucalyptus open-Source cloud-computing system[A]. Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid [C]. Washington,DC: IEEE Computer Scoiety,2009.124 – 131.

## 作者简介

**李建敦** 男,1982 年 11 月出生,河北张家口人.2005 年毕业于河北经贸大学信息技术学院,取得工学学士学位;随后考入上海大学计算机工程与科学学院,2011 年博士毕业;现为上海电机学院电子信息学院教师,从事高性能计算与云计算等方面的研究.
E-mail:jdli@shu.edu.cn

**彭俊杰** 男,1977 年 1 月出生,湖北红安人.副教授,硕士生导师.1999 年、2001 年和 2005 年在哈尔滨工业大学分别获工学学士、硕士和博士学位.现为上海大学计算机学院高性能计算研究室主任.主要从事云计算、物联网、光学计算及嵌入式系统等相关领域的研究工作.
E-mail:jjie.peng@shu.edu.cn

**张 武** 男,1957 年 11 月出生,江西武宁人.教授,博士生导师.1988 年于西北工业大学获博士学位.现为上海大学计算机学院执行院长,上海大学高性能计算中心主任.主要从事高性能计算与应用及生物信息学等领域的研究工作.
E-mail:wzhang@shu.edu.cn