Journal of System Simulation

Volume 28 | Issue 3 Article 12

7-2-2020

Data Management of Data Processing Framework in Green Data Center

Zhang Xiao
1. State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China;

Gao Yuan
2. NARI Technology Co, Ltd, Nanjing 210023, China;

Xiaoliang Wang
1. State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China;

Yiyong Ge
2. NARI Technology Co, Ltd, Nanjing 210023, China;

See next page for additional authors

Data Management of Data Processing Framework in Green Data Center

Abstract

Abstract: Using renewable energy in data center is an environment-friendly way to solve the problem of high energy consumption of data center. Since renewable energy is variable, delaying the jobs which has no strict deadline i a widely used strategy to maximize the usage of renewable energy. Meanwhile, turning the idle servers off can further reduce energy consumption. If the data required by the jobs to be processed are unavailable, some servers in sleep state need to be reactivated to guarantee that the data required by the jobs are available. Such operation may lead to energy waste due to the frequent reactive processes. *An effective data management algorithm was proposed, which copied the data required by the jobs in waiting queue to active servers in advance. By doing so, the times that the sleep servers were reactivated could be reduced*. Simulation results show that the times can be reduced by 43% on average.

Keywords

green data center, data processing framework, energy, renewable energy, data management

Authors

Zhang Xiao, Gao Yuan, Xiaoliang Wang, Yiyong Ge, Haixiang Yang, and Shupeng Wan

Recommended Citation

Zhang Xiao, Gao Yuan, Wang Xiaoliang, Ge Yiyong, Yang Haixiang, Wan Shupeng. Data Management of Data Processing Framework in Green Data Center[J]. Journal of System Simulation, 2016, 28(3): 592-599.

第 28 卷第 3 期 系统仿真学报© Vol. 28 No. 3
2016 年 3 月 Journal of System Simulation Mar., 2016

绿色数据中心数据处理型框架中的数据管理

张啸[1]，高原[2]，王晓亮[1]，葛以踊[2]，杨海祥[1]，万书鹏[2]
(1. 南京大学计算机软件新技术国家重点实验室南京大学计算机科学与技术系，南京 210023)
2. 国电南瑞科技股份有限公司，南京 210023)

摘要：使用绿色能源已成为解决数据中心能耗问题的一种有效方式。为了降低绿色能源变化幅度大的特点带来的影响，通常将可延迟作业放入等待队列，将相应空闲服务器置为休眠状态，降低系统能耗，在新能源可用的时候执行作业。当新作业执行时，需要重新开启休眠状态服务器来保证数据可用性。数据放置与作业执行时间的不统一，会导致频繁开启休眠服务器，带来能源浪费。*针对绿色数据中心提出一种数据调度策略，根据数据处理型框架中等待队列作业调度次序，通过将未来一段时间内需要被读取的数据块提前复制在活跃服务器上，降低休眠状态服务器开启的次数，从而降低总体能耗。*实验模拟结果显示，该算法可平均减少 43%的休眠状态服务器重复开启次数。

关键词：绿色数据中心；数据处理型框架；能耗；新能源；数据管理
中图分类号：TP391.9 文献标识码：A 文章编号：1004-731X (2016) 03-0592-08

Data Management of Data Processing Framework in Green Data Center

Zhang Xiao[1], *Gao Yuan*[2], *Wang Xiaoliang*[1], *Ge Yiyong*[2], *Yang Haixiang*[1], *Wan Shupeng*[2]
(1. State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)
2. NARI Technology Co, Ltd, Nanjing 210023, China)

Abstract: Using renewable energy in data center is an environment-friendly way to solve the problem of high energy consumption of data center. Since renewable energy is variable, delaying the jobs which has no strict deadline i a widely used strategy to maximize the usage of renewable energy. Meanwhile, turning the idle servers off can further reduce energy consumption. If the data required by the jobs to be processed are unavailable, some servers in sleep state need to be reactivated to guarantee that the data required by the jobs are available. Such operation may lead to energy waste due to the frequent reactive processes. *An effective data management algorithm was proposed, which copied the data required by the jobs in waiting queue to active servers in advance. By doing so, the times that the sleep servers were reactivated could be reduced.* Simulation results show that the times can be reduced by 43% on average.

Keywords: green data center; data processing framework; energy; renewable energy; data management

引言

近些年来，用新能源供能的绿色数据中心引起了人们的极大关注。众所周知，高能耗一直都是数据中心中的突出问题。目前，数据中心每年消耗了全球 1.3%的电能，到 2020 年，这个比例将达到 8%[1]。随着传统能源的日益枯竭，依赖于传统能源的电力成本不断上升，极大地增加了高能耗数据中心的运营成本。此外，传统能源会产生大量的碳排放，加剧了日益恶化的环境问题。而太阳能、风

收稿日期：2014-07-15 修回日期：2014-10-24；基金项目：国家自然科学基金(61370028, 91218302, 61321491)；江苏省自然科学基金(BK2011191)；江苏省科技支撑计划(BE2013116)；中央高校基本科研业务费专项资金(20620140514)；国家电网公司科技项目；
作者简介：张啸(1991-)，男，山东，硕士生，研究方向为数据中心。

能、潮汐能等新型能源技术的发展为解决数据中心高能耗成本和高碳排放量问题提供了新的思路。新型能源是采用新技术开发利用可再生能源，它们有着取之不尽、碳排放量低等诸多优点。因此，使用新能源的数据中心悄然兴起。目前，不仅仅是 Apple，FaceBook 等 IT 巨头在大力发展新能源供能的绿色数据中心，许多中小型数据中心也在使用新能源以求降低能耗成本，这种情形变得越来越普遍[2]。科研机构也广泛开展了对绿色数据中心中新能源的高效利用的相关研究，如文献[2-6,11-14]。

新能源的产量受天气等外界因素影响较大，具有不稳定、变化幅度大的特点。以太阳能为例，太阳能发电是使用光伏板组件将太阳能转化为电能的过程，这种转换与光照强度有着直接的关系。不同地方年均日照时数有着巨大的差异；在同一地方，一天当中太阳能转化效率也有很大的变化。如图 1 为美国 Texas 某天太阳能电池板输出功率曲线变化图[10]所示。

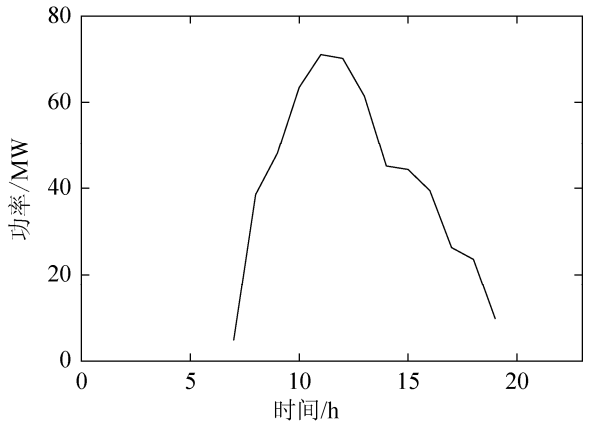


图 1 美国 Texas 某天太阳能电池板输出功率曲线图[10]

正午时分太阳能转换效率达到最高，而夜晚则不能利用太阳能进行发电。对于要求具有高性能、高可靠性的数据中心而言，如果直接将新能源应用到数据中心中有可能导致断电等问题，严重影响数据中心的性能。为降低新能源间歇性、变化幅度大的特点带来的影响，可以人为调整工作负载，使新能源产量高的时段内运行更多的工作负载。这一般是通过延迟低优先级的批处理型作业实现的。这些低优先级的作业并不需要立即处理，只要在规定的截止日期之前完成即可，所以可以尽量将这些作业延迟到未来新能源可用的时候再执行，以最大化新能源的利用率，同时只要保证不超过这些作业的最终截止期限即可。Goiri 等人提出了 GreenHadoop[3] 负载调度系统。GreenHadoop 是通过延迟部分低优先级作业在新能源充足的时刻运行，达到了最大化新能源利用率的目的。

由于要延迟部分低优先级的作业，所以 GreenHadoop 在运行工作负载的时候，会分为 2 个队列：等待队列和运行队列，如图 2 所示。在作业提交阶段，优先级高的作业直接进入运行队列，立即处理；其余作业则进入等待队列，在合适的时机被选择进入运行队列进行执行。在每一个时段(时段是调度的最小单位，默认 5 mins)的开始，GreenHadoop 针对运行队列和等待队列中的作业，会根据历史数据来计算平均运行时间和能耗，再根据这些数据来计算这一时段内所需的能源，以期在此时段内采用最佳数量的服务器来运行作业。

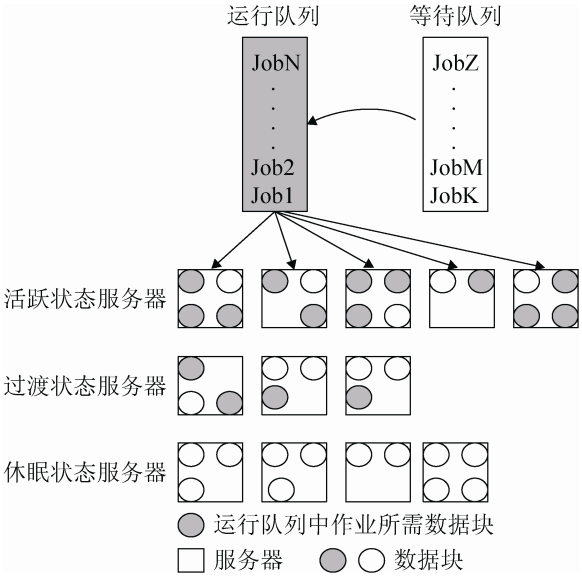


图 2 GreenHadoop 作业调度

在运行作业期间，要保证运行作业所需数据块是可用的。数据块以三副本策略存储在各个服务器上。在 GreenHadoop 中，服务器被分为 3 种状态：活跃状态、过渡状态和休眠状态。活跃状态服务器

第 28 卷第 3 期 系统仿真学报 Vol. 28 No. 3
2016 年 3 月 Journal of System Simulation Mar., 2016

不仅能够运行作业，也能存储数据块并提供对数据块的访问。过渡状态服务器只提供对数据块的访问，而不允许其再存放新的数据块；过渡状态的设置避免了服务器由活跃状态直接变为休眠状态所带来的运行队列作业所需数据块不可访问和正在运行作业中断的问题，在关闭过渡状态服务器之前会将运行队列中所需数据块从过渡状态服务器上复制到活跃状态上，同时让本来在其上运行的作业继续运行完毕，如此就不会中断作业的执行。休眠状态指的是 ACPI S3 状态，即挂起到内存状态，该状态下系统的耗电量极低且系统被唤醒速度极快。在不同的时段内，活跃服务器的数量是动态变化的。若当前活跃服务器的数量小于需要的数量，就会将过渡状态服务器变为活跃状态，如果此时还需要更多的活跃服务器，就要开启休眠状态的服务器；若当前活跃服务器的数量大于需要的数量，就将一部分活跃服务器变为过渡状态。同时，当过渡状态服务器上不再有作业执行，就将过渡服务器置为休眠状态。在服务器状态变化过程中，要保证运行作业所需数据块的可用性。

GreenHadoop 保证数据可用性的方式基于 Covering Subset[7]策略。Covering Subset 策略是一个静态的方法，它维持了一个特定的服务器集合，在这个服务器集合中存放了每个数据块的至少一个副本。不管该服务器集合上数据是否是当前所需要的，这个服务器集合都要一直处于活跃状态。GreenHadoop 在 Covering Subset 策略基础上做了改进，GreenHadoop 依赖于数据的动态复制。数据的动态复制指的是在将过渡状态服务器置为休眠状态过程中，会将过渡状态服务器上的运行队列中作业所需数据复制到活跃服务器的可用空间中，并关闭这部分的过渡状态服务器以降低能耗。GreenHadoop 仅仅保证了运行队列中的作业的数据是可用的，所以所需要的活跃状态服务器数量也要更少。

在 GreenHadoop 保证数据可用性的算法中，只保证当前运行队列中的作业的数据块是可用的，所以不可避免地带来一些问题。若等待队列一个作业进入运行队列，而这个作业所需数据是在休眠状态服务器上，此时就需要开启此休眠状态服务器以提供需要处理的数据，造成不必要的能源浪费。由于未来运行作业所需数据块以及下一时段活跃状态服务器数量的不可预知，导致该问题不可能求得最优解，但是，充分利用已知条件，我们可以得到一种解决该问题的有效算法。

针对休眠状态服务器频繁开启的问题，本文提出一个基于数据块提前放置的数据管理的算法。在传统 Hadoop 中，每到一个作业就会立即执行，不会有等待队列的存在，也不会延迟作业。所以下一时段内具体到达哪些作业、需要处理哪些数据块是无法提前获取的。GreenHadoop 中的等待队列给数据块的提前放置提供了一个很好的机会，我们通过遍历等待队列，将下一个时段内可能被选入运行队列的作业所需数据块提前放置在活跃服务器上，从而达到了减少休眠状态服务器开启次数的目的。同时若当前时段复制的数据块在下一时段并没有被使用而服务器却被置为休眠状态，这就导致了复制的数据块白白浪费。针对这个问题，我们通过对复制数据块设置是否被作业处理过的标记，若在需要关闭的休眠状态服务器上有未被处理过的复制数据块，则一并复制到活跃状态服务器上。

本文的主要贡献如下：

(1) 为保证数据块可用性，GreenHadoop 中休眠状态服务器会频繁开启，本文提出了绿色数据中心中该如何降低休眠状态服务器开启次数的问题。

(2) 本文设计了一个数据块提前放置算法。在调度作业过程中，会将下一时段可能所需数据块提前放置在活跃服务器上。如此可以降低服务器重复开启的次数。同时针对复制数据块未被使用，其所在服务器就被置为休眠状态的情况，通过对数据块加标记并重新复制到活跃状态服务器上的方式予以解决。通过模拟实验发现，休眠状态服务器重复开启次数平均降低 43%。

1 问题描述及算法设计

1.1 问题定义

我们采用了 GreenHadoop 中的作业调度机制，并在其数据管理算法基础上提出了数据块提前放置的策略。当从等待队列选择一个作业进入运行队列时，而这个作业所需的数据块并不在活跃服务器或过渡服务器中，此时，就需要将拥有该数据块但处于休眠状态的服务器开启。我们的目标是使服务器由休眠状态开启为过渡状态的次数尽可能少。

为减少休眠状态服务器开启的次数，我们考虑将下一时段内可能会运行的作业所需数据块提前放置在活跃服务器上。提前放置的数据块集合 preBlock_t 包含 2 个部分：P_t 和 R_t(参数含义如表 1 所示)。获取 R_t 是为了解决可能会带来的复制数据块浪费问题。即上一时段存放复制数据块的活跃服务器在本时段内需要被置为过渡状态，而处理该服务器上的数据块的作业在本时段内并没有到来，之后该过渡服务器由于不再执行作业就被置为休眠状态了，这就导致了复制的数据块白白浪费。

综上，数据块提前放置问题可形式化如下，各个参数含义如表 1 所示。

$$\text{minimize} \quad \sum_{t=0} S(B_t - D_t - H_t - \text{preBlock}_t) \quad (1)$$

$$\text{subject to} \quad T = A_t + |D_t| \quad (2)$$

$$|\text{preBlock}_t| \leqslant A_t \quad (3)$$

$$P_t \subseteq (B_t - D_t - H_t) \cap C_t \quad (4)$$

$$\text{preBlock}_t = P_t \cup R_t \quad (5)$$

目标函数(1)表示最小化休眠状态服务器开启的次数，B_t -D_t -H_t- preBlock_t 表示下一时段内被选择作业所需的位于休眠状态服务器上的数据块集合；公式(2)表示下一时段活跃状态服务器上的可用空间；公式(3)表示本时段内需要复制的数据块的量不会超过下一时段服务器上的可用空间；公式(4)表示一部分需要复制的数据块首先是下一个时段内作业运行所需要的，其次是在需要关闭的过渡状态服务器集合中；公式(5)表示该过渡状态服务器上若有复制的但从未被处理的数据块，也要将其复制到活跃服务器可用空间上。问题的难点在于未知变量太多，导致无法求得最优解。例如对于当前时段来说，B_t 和 A_t 是未知的，所以我们只能利用已知条件，得到与 B_t 尽可能相似的数据块集合进行求解。

表 1 参数及其含义

参数	参数含义
preBlock_t	本时段内要复制的数据块集合
P_t	根据等待队列和过渡服务器选择的要复制的数据块集合
R_t	本时段内要关闭的过渡服务器上复制但未处理过数据块集合
B_t	下一时段内，被选择的作业所需数据块集合
D_t	下一时段内活跃状态服务器上数据块集合
H_t	下一时段继续维持过渡状态的服务器数据块集合
C_t	本时段要关闭的过渡服务器上非复制数据块集合
A_t	下一时段内活跃服务器上可用空间数量
T	下一时段内活跃服务器能存储的数据块的总量
$S(X)$	包含数据块集合 X 的最小数量的休眠状态的服务器集合的数量

1.2 算法设计

1.2.1 算法概述

我们采取的方法是提前将作业需要的数据块放置到有空闲位置的活跃服务器上，如图 3 所示。该数据管理算法主要分为 2 个部分：获取提前放置数据块算法(算法 2)和数据块放置算法(算法 3)。算法 2 的目的是得到需提前放置的数据块集合，需要分别遍历需要关闭的过渡状态服务器集合和等待队列。算法 3 是将算法 2 中得到的数据块集合放置到活跃状态服务器的可用空间中，需要遍历活跃状态服务器集合以及数据块集合。数据块提前放置算法在把过渡状态服务器置为休眠状态之前执行。

第 28 卷第 3 期 系统仿真学报 Vol. 28 No. 3
2016 年 3 月 Journal of System Simulation Mar., 2016

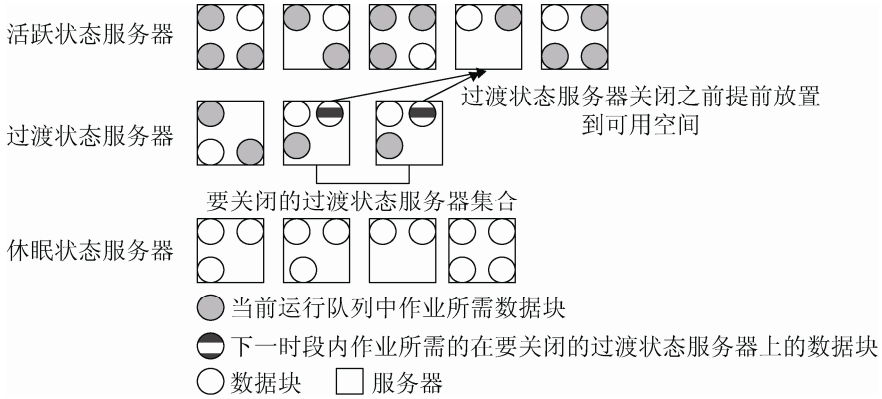


图 3 基于数据块提前放置的数据管理

1.2.2 数据块可用性保证算法细节

算法 1. 数据管理算法

输入：当前活跃状态服务器数量 currentNumber；本时段内所需要的活跃服务器数量 neededNumber；

调整活跃服务器数量：

IF(currentNumber>neededNumber)

将(currentNumber- neededNumber)个活跃状态服务器置为过渡状态；

在将过渡状态服务器置为休眠状态之前：

维持一个需要关闭的过渡服务器集合 ***s***；

检测活跃服务器上可用空间数量 avaiSpace；

IF(avaiSpace > 0)

PreBlock = getBlocks()；(算法 2)

putBlocksOnServers(PreBlock)；(算法 3)

算法 2. 获取需提前放置数据块算法

输入：需要关闭过渡状态服务器集合 ***s***，等待队列 ***q***；

活跃状态服务器集合 ***t***；

输出：需要复制的数据块集合 *PreBlock*；

getBlocks()：

FOR(each job i in ***q***)

mark i using<T_i ， N_i>；

sort(***q***)；

FOR(each job i in ***q***)

FOR(each block b_{ij} that i needs)

IF(b_{ij} is not on ***t*** && b_{ij} is on ***s***)

PreBlock.add(b_{ij})；

FOR(each server *t* in ***s***)

IF(block *b* is replicated on *t* && b.state == unused)

PreBlock.add(b)；

ELSE IF(*b* is replicated on *t* && b.state == used)

delete *b*；

算法 3. 数据块放置算法

输入：活跃状态服务器集合 ***s***，数据块集合 PreBlock；

putBlocksOnServers(PreBlock):

FOR(each block *i* in *PreBlock*)

FOR(each server *j* in ***s***)

IF(*j*.remainingSpace>0)

put(*i*，*j*)；

break；

首先会调整当前时段活跃服务器的数量，若当前活跃状态服务器数量要多于本时段内需要的数量，就需要将部分活跃状态服务器置为过渡状态。活跃状态服务器选择规则如下：选择复制过来的数据块的数量跟运行队列中作业所需数据块数量之和最小的服务器。其次，当过渡状态服务器不再运行作业，会维护一个需要关闭的过渡状态服务器集合。在关闭过渡状态服务器之前复制数据块。

在算法 2 中，由于将等待队列中的作业选到运行队列中，必须满足 3 个条件[3]：(1)作业即将到达必须开始的截止期限；(2)作业所需数据已都在活跃服务器上；(3)活跃服务器还有多余计算资源。其中，条件(2)跟此数据可用性保证算法无关。所以根据条件(1)(3)，对等待队列中的作业以二元组标记并进行排序。比如作业 j 用一个二元组<T_j ，N_j>来表示。T_j 表示作业 j 的截止时间，而 N_j 表示提供作业 j 所需的在休眠状态服务器上数据块的数量。首先根据 T_j 对二元组进行排序，T_j 越小表示开始时间越早，作业优先级越高；对于 T_j 相同的作业，再根据 N_j 排序，N_j 越小，表示作业 j 所需的在休眠状态服务器上数据块的数量越小，作业优先级越高。排序之后，依次判定等待队列中的作业所需数据块是否在需要关闭的过渡状态服务器集合中，若所需数据块在需要关闭的过渡状态服务器集合中，则加入到数据块集合 PreBlock 中，否则，遍历下一个作业，直到遍历完整个等待队列。同时，针对复制数据块浪费的问题，对每个数据块设置标记，表示作业是否处理过该数据块。遍历需要关闭的过渡服务器集合，若是该服务器上复制的数据块还没有被处理，那也接入到数据块集合 PreBlock 中。最后，将算法 2 中得到的数据块集合 PreBlock 中每一个元素依次放置到活跃状态服务器的可用空间中即可。

获取数据块集合 PreBlock 最坏情况下时间复杂度为 $O(mn)$，m 是作业最多所需数据块的数量，n 是等待队列中的作业数量。在多数情况下，通过一些参考文献的数据分析，例如：“FaceBook-Drived”工作负载[9]，若每个存储在服务器上数据块大小默认为 64 MB，则每个作业最小处理 4 个数据块，最大处理 600 个数据块，m 最大为 600 且在每一时段作业到达数量是有限的，则复杂度是可以接受的。

2 实验模拟分析

2.1 仿真设置

假设集群拥有 16 台服务器。假设每个时段内运行作业数量以及作业所需数据块数量都是随机的。每组实验服从不同区间的均匀分布。例如第 10 组实验每个时段运行作业量服从区间[1，21]上的均匀分布，平均每个时段到达作业数量为 11 个。每个时段内活跃状态服务器数量随该时段运行作业数量变化而变化。

2.2 实验结果及分析

2.2.1 理论最优解

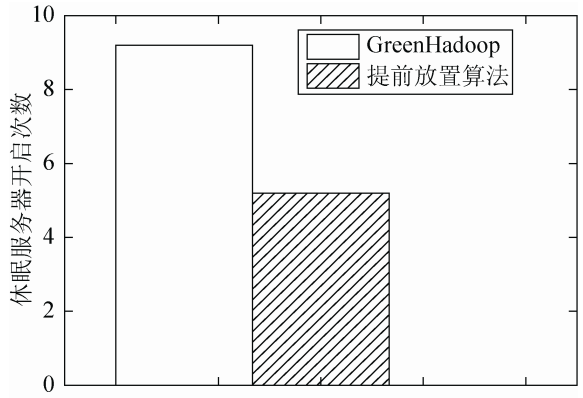
在理想条件下，服务器由休眠状态开启为过渡状态次数应为 0。理想条件是指：(1)活跃服务器上可用空间无穷大，不论何时放置多少数据块都是可以的；(2)准确得知作业的运行顺序，并且作业运行所需数据块必须都在活跃服务器和要关闭的过渡服务器上。然而在实际环境下，活跃服务器上的可用空间是有限的；同时我们只能对等待队列中作业进行排序，选择最有可能被选择进入运行队列中的作业，而这与作业实际被调度的顺序不一定是相同的，同时这些作业所需数据块并不一定全在活跃服务器和要关闭的过渡服务器集合中。所以，我们的算法是一种贪心策略，每一次都尽量填满活跃服务器上的可用空间，尽可能保证未来运行作业所需数据块会在服务器上。

2.2.2 实验结果分析

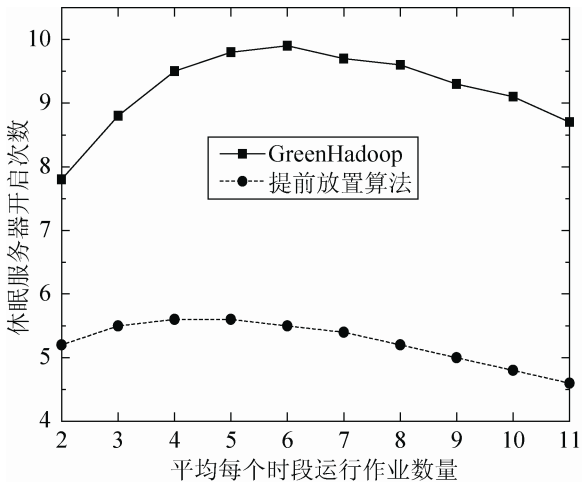
我们进行了 10 组实验，每组实验对 3000 个时段进行了模拟。每组实验中，每个时段内平均运行作业数量有所不同，实验结果如图 4 所示。图 4(a)为 10 组实验中在每个时段内休眠状态服务器的平均开启次数。其中，GreenHadoop 平均每个时段开启休眠状态服务器的次数为 9.2 次，而运行了提前放置算法后平均每个时段开启次数为 5.2 次，下降了 43%。图 4(b)中，横坐标表示每个时段平均运行

第 28 卷第 3 期 系统仿真学报 Vol. 28 No. 3
2016 年 3 月 Journal of System Simulation Mar., 2016

作业的数量，纵坐标表示当前时段休眠服务器开启的次数。可以看出随着每个时段内运行作业数量的增多，休眠状态服务器开启的次数随之增加，但是当达到一定数量时，休眠服务器开启次数将保持在相对稳定的水平，此时主要是受到作业所需数据块数量的制约。之后休眠状态服务器开启次数会略有下降的趋势，这是因为随着运行作业数量的增多，活跃状态服务器的数量也随之增多，所以导致休眠状态服务器开启的次数会有所下降。值得注意的是，数据块提前放置算法只是延长了过渡状态服务器关闭的时间，如果不提前放置数据块，那么在下一时段就必须开启休眠服务器为过渡状态，在运行作业处理完该数据块之前，该服务器都不能再被置为休眠状态，与此相比数据块提前放置算法所带来的能耗是极低的。



(a) 平均开启次数对比



(b) 休眠服务器开启次数跟每个时段平均运行作业数量关系

图 4 实验结果图

算法主要受活跃服务器上可用空间的量和要关闭的过渡状态服务器集合上存放数据块数量两个因素制约。在不同时段，活跃服务器的量是动态变化的，若之前时段活跃服务器的数量较少，也就是可用空间较少，能够存储的预测数据块是非常有限的，这会导致实验效果并没有那么明显。若是活跃服务器的数量在各个时段相对稳定，而且数量维持在较高水平，实验效果会更明显。

在同一时段内，活跃状态服务器数量不同，导致下一时段服务器重复开启的次数也不同。从图 5 可以看出，随着活跃状态服务器数量的增加，即可用空间的增多，服务器重复开启的次数会随之减少，但是不会一直递减下去。例如在本实验中，当活跃状态服务器数量增加到 2 时，不管之后活跃状态服务器如何增加，服务器重复开启次数一直维持不变，这是因为此时虽然可用空间的数量是足够的，但是关闭的过渡状态服务器上并没有未来作业所需的数据块，此时就需要开启休眠状态的服务器来保证数据的可用性。同时，GreenHadoop 中随着活跃状态服务器的增加，服务器重复开启的次数也有减少的趋势，这是因为未来作业所需数据块可能恰好在该活跃状态服务器上，但是，服务器重复开启次仍保持较高数值，例如当活跃状态服务器数量为 1 时，GreenHadoop 中服务器重复开启次数为 7 次，而运行了数据块提前放置算法之后，次数仅为 4 次，再次证明了数据块提前放置算法的有效性。

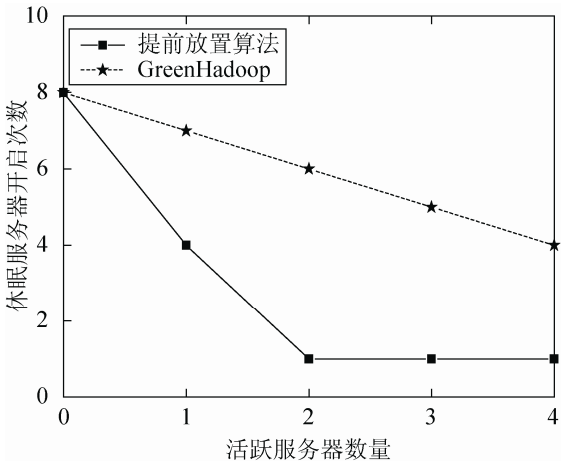


图 5 活跃状态服务器数量跟服务器重复开启次数的关系

3 结论

本文针对绿色数据中心环境中数据处理型框架下的数据管理问题提出一个算法。在节能型集群中，一般会关闭一部分空闲服务器达到降低能耗的目的。而关闭服务器可能导致服务器上数据不可用的问题，为了读取数据，就需要开启处于休眠状态的服务器。本文提到的数据管理的算法就是为了降低服务器开启的次数，进一步降低了能耗。通过模拟实验，我们的算法使休眠服务器开启次数平均降低 43%。

算法的优点在于：首先，算法能有效降低为保证作业所处理数据可用性，休眠状态服务器开启的次数；其次，复制到活跃状态服务器可用空间上的数据块在处理完毕后会立即删除，这并不违反为保证数据可靠性所采用的三副本放置策略。

参考文献:

[1] Gao P X, Curtis A P, Wong B, Keshav S. It's not easy being green [C]// Proceedings of the ACM SIGCOMM 2012, Helsinki, Finland. USA: ACM, 2012: 211-222.

[2] Deng W, Liu F, Jin H, et al. Harnessing renewable energy in cloud datacenters: opportunities and challenges [J]. IEEE Network (S0890-8044), 2014, 28(1): 48-55.

[3] Goiri Í, Le K, Nguyen T D, et al. GreenHadoop: leveraging green energy in data-processing frameworks [C]// Proceedings of the 7th ACM European Conference on Computer Systems. USA: ACM, 2012: 57-70.

[4] Goiri Í, Beauchea R, Le K, et al. GreenSlot: scheduling energy consumption in green datacenters [C]// Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. USA: ACM, 2011: 20.

[5] Goiri Í, Katsak W, Le K, et al. Parasol and GreenSwitch: Managing datacenters powered by renewable energy [C]// Proceedings of the eighteenth international conference on Architectural support for programming languages and operating systems. USA: ACM, 2013: 51-64.

[6] Aksanli B, Venkatesh J, Zhang L, et al. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers [J]. ACM SIGOPS Operating Systems Review (S0163-5980), 2012, 45(3): 53-57.

[7] Leverich J, Kozyrakis C. On the energy (in) efficiency of hadoop clusters [J]. ACM SIGOPS Operating Systems Review (S0163-5980), 2010, 44(1): 61-65.

[8] Lang W, Patel J M. Energy management for mapreduce clusters [J]. Proceedings of the VLDB Endowment (S2150-8097), 2010, 3(1-2): 129-139.

[9] Zaharia M, Borthakur D, Sarma J S, et al. Job scheduling for multi-user mapreduce clusters [M]// Tech. Rep. UCB/EECS-2009-55, 2009. Berkeley, USA: EECS Department, University of California, 2009.

[10] Ibanez E, Brinkman G, Lew D. Sub-hour solar data for power system modeling from static spatial variability analysis [M]. USA: National Renewable Energy Laboratory, 2012.

[11] Liu Z, Chen Y, Bash C, et al. Renewable and cooling aware workload management for sustainable data centers [C]// ACM SIGMETRICS Performance Evaluation Review. USA: ACM, 2012, 40(1): 175-186.

[12] Li C, Zhang W, Cho C B, et al. SolarCore: Solar energy driven multi-core architecture power management [C]// High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on. USA: IEEE, 2011: 205-216.

[13] Li C, Qouneh A, Li T. iSwitch: coordinating and optimizing renewable energy powered server clusters [C]// Computer Architecture (ISCA), 2012 39th Annual International Symposium on. USA: IEEE, 2012: 512-523.

[14] Sharma N, Barker S, Irwin D, et al. Blink: managing server clusters on intermittent power [C]// ACM SIGARCH Computer Architecture News. USA: ACM, 2011, 39(1): 185-198.

