

基于分布式离散事件系统监控理论的云资源动态调度

刘富春, 胡 芹

广东工业大学计算机学院, 广东 广州 510006

基金项目: 国家自然科学基金资助项目(61273118, 61673122); 广东省高校省级重大项目(2014KZDXM033); 广东省公益研究与能力建设专项资金资助项目(2015A030402006); 广东工业大学计算机学院重大奖项培育项目(2016PY01)

通信作者: 刘富春, fliu2011@163.com 收稿/录用/修回: 2016-08-30/2016-11-13/2017-02-22

摘要

鉴于复杂云环境具有较强的动态变化与分散性特点且云数据中心各实体又具有离散特性, 提出了一种基于分布式离散事件系统监控理论的云资源动态调度策略. 先将云环境构建为一个由物理服务器类、虚拟机类、云任务类等多个子系统构成的分布式离散事件系统(DES); 再将分布式离散事件系统监控理论应用于云资源动态调度. 通过系统局部控制器的局部控制及其与全局控制器之间的信息交互, 实现对云资源调度的整体优化, 保障各个子系统资源分配的合理性. 同时, 系统通过调控资源负载均衡, 提高云资源整体利用率与云服务质量, 提升整个云平台性能. 实验结果表明, 在任务量不断增加的情况下, 所提的云资源动态调度策略与传统云资源调度策略相比, 在资源利用率和任务执行时间等方面都具有较明显的优势.

关键词

离散事件系统
分布式系统
云资源分配
动态调度策略

中图分类号: TP13

文献标识码: A

Dynamic Scheduling of Cloud Resource Based on Control Theory of Distributed Discrete Event Systems

LIU Fuchun, HU Qin

School of Computers, Guangdong University of Technology, Guangzhou 510006, China

Abstract

Given the complex dynamic and distributed nature of the cloud environment together with the discrete characteristic of cloud data center entities, we propose a dynamic scheduling strategy of cloud resources based on the distributed discrete event systems (DES) theory. First, we construct the cloud environment as a distributed DES composed of numerous subsystems, which include the physical server subsystem, the virtual machine subsystem, and the cloud task subsystem. Then, we apply supervisory control theory of distributed DESs to deal with the dynamic scheduling of the cloud resources. Through the control from local controllers and the information exchange between the global and local controllers, the proposed strategy achieves the overall optimization of the cloud resources and guarantees the rational allocation of the subsystems' resources. Moreover, the scheduling of the resources' load balancing improves the overall utilization of the cloud resources and the quality of the cloud service and enhances the performance of the entire cloud platform. Experiments indicate that with the increase in tasks, the proposed control strategy has obvious advantages in the utilization rate and the execution time compared with the traditional strategies.

Keywords

discrete event system;
distributed system;
cloud resource allocation;
dynamic scheduling strategy

1 引言

云计算为用户提供一种超强计算能力和高效数据存储能力^[1-3], 云资源动态调度是按需分配或释放虚拟机资源^[4]. 如何调度资源保证负载均衡, 提高利用率是当前云

计算技术的研究热点之一. 文[5-7]主要从云资源提供商收益角度出发, 平衡供应商与用户之间的费用; 文[8-10]主要考虑云资源利用率, 通过合理管理资源达到资源最大化利用.

根据资源实体的离散特性, 本文作者最近提出了一种

基于离散事件系统 (DES) 监控理论的云资源分配控制模型及相应的分配算法, 实现了云资源利用率的最大化^[11]. 分布式 DES^[12] 已广泛应用于计算机网络、通信网络、军事系统、机器人技术、柔性制造等应用领域, 文[13]有这方面较详细的介绍. 本文作者也对分布式 DES 进行了较深入探讨, 分别提出了分布式 DES 的监控机制^[14-15].

本文运用分布式 DES 监控理论研究云资源动态调度问题, 将文[11]的工作拓展至分布式情形, 建立一种基于分布式 DES 监控理论的云资源动态调度模型和调度算法. 实验表明, 在任务量不断增加的情况下, 与传统云资源调度策略相比, 本文提出的调度策略在资源利用率和任务执行时间等方面都具有较明显的优势. 这种将 DES 监控理论应用于云资源调度是一种尝试与创新, 为进一步深入研究云资源动态调度最优化算法等相关问题奠定基础.

2 云资源分配模型分析

根据云资源实体的离散特性构造可量化的云资源模型, 对云资源系统参数化, 并将资源池中的资源按类型记为 $R = \{R_1, \dots, R_n\}$, 其中 R_i 表示系统第 i 类型资源, $i = 1, \dots, n$. 将任务子系统状态集表示为 $T_i = \{T_1, \dots, T_n\}$, 其中 $T_x = \{R_{T_x}, V_i\}$ 为第 x 个请求任务, 这里 $R_{T_x} = \{C_{R_1}^{T_x}, \dots, C_{R_n}^{T_x}\}$ 为任务的资源需求量, 而 $C_{R_i}^{T_x}$ 表示在第 x 请求任务中所需要 R_i 类资源的容量, 若请求中无该类资源, 则将 $C_{R_i}^{T_x}$ 置为 0. $V_i \subset V_i \cup \{0\}$ 表示为该任务提供服务的虚拟机, 取值为 0 表明暂未资源分配.

将虚拟机子系统的状态集表示为 $V_i = \{V_1, \dots, V_n\}$, 其中每个 $V_y = \{R_{V_y}, M_i, T_j\}$ 表示系统第 y 个虚拟机, 这里 $R_{V_y} = \{C_{R_1}^{V_y}, \dots, C_{R_n}^{V_y}\}$ 表示第 y 个虚拟机中的资源量分配: $C_{R_i}^{V_y}$ 表示第 R_i 类资源的分配量, 若无该类资源, 则 $C_{R_i}^{V_y}$ 置为 0, 任务执行完毕后所占资源量都置为 0; M_i 表示支持该虚拟机的物理服务器, M_i 为 0 时表示该虚拟机已经被销毁; T_j 表示该虚拟机服务的任务, T_j 为 0 时表示当前虚拟机空闲.

物理服务器子系统的状态集可表示为 $M_i = \{M_1, \dots, M_n\}$, 其中 $M_k = \{R_{M_k}, \text{token}\}$ 表示第 k 个物理节点; $R_{M_k} = \{C_{R_1}^{M_k}, \dots, C_{R_n}^{M_k}\}$ 表示第 k 个物理服务器的资源占有量; $C_{R_i}^{M_k}$ 表示第 R_i 类资源的占有量, 若无该资源 $C_{R_i}^{M_k}$ 置为 0; token 表示该物理服务器挂载虚拟机状态, token 为 1 表示已经挂载虚拟机, token 为 0 表示没有挂载. 资源模型标记每一类资源的不同状态, 为资源动态分配提供基础.

同时, 将云资源分配模型分为用户任务子系统、虚拟机子系统和物理节点子系统, 如图 1 所示, 物理节点层 m_1, \dots, m_n 为分布在各地的物理服务器, 各服务器可按需划分出多个虚拟机, 每个虚拟机都是一个具有计算能力的资源提供点^[16].

在云资源分配的调度过程中, 先按需申请虚拟机, 服务完毕后回收资源, 保留常用的虚拟器配置. 本文优先将可拆分的大任务分解为多个子任务, 分发到不同的服务器来保障负载均衡. 目前, Hadoop 技术能有效实现任务拆

分^[17], 服务器负载态可从系统响应时间 t 是否超出预算及性能参数是否低于阈值来判断, 其中响应时间为

$$t = k \cdot \frac{1}{N \cdot 1024} \cdot g \cdot I \quad (1)$$

其中, k 表示时钟周期个数, g 表示机器周期的个数, N 表示当前虚拟机所分配的内存容量, I 为当前任务的平均指令数.

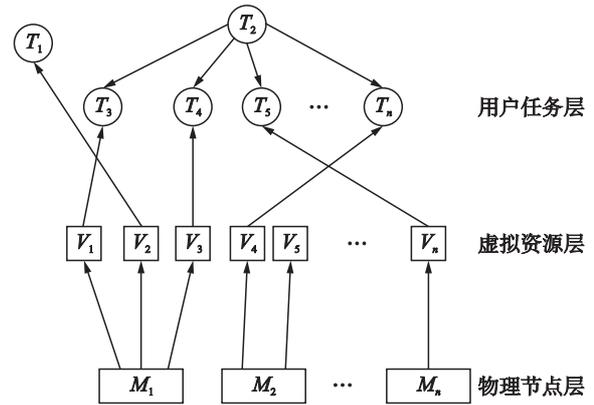


图 1 云资源 3 层模型

Fig.1 The three-layer model of the cloud resource

本文引入欧氏距离来表示参数间的贴进度. 2 个资源载体 T_x 与 T_y 之间的资源欧氏距离为

$$D(x, y) = \sqrt{(C_{R_1}^{T_x} - C_{R_1}^{T_y})^2 + \dots + (C_{R_n}^{T_x} - C_{R_n}^{T_y})^2} \quad (2)$$

为求出最大贴进度, 需找出使得多参资源体的欧氏距离 $D(x, y)$ 最小的 y 值. 当云资源分配时, 本文优先利用碎片资源最大化资源利用率 U :

$$U = \frac{\sum_{x=1}^n \sum_{i=1}^k C_{R_i}^{T_x}}{\sum_{y=1}^n \sum_{i=1}^k \text{token} \cdot C_{R_i}^{M_y}} \quad (3)$$

在不超过系统性能阈值情况下, 缩短完成所有任务的执行时间之和 t :

$$t = \sum_{i=0}^n k \cdot \frac{1}{N_i \cdot 1024} \cdot g \cdot I_i \quad (4)$$

3 基于分布式 DES 监控理论的云资源调度

在分布式云资源 DES 系统中, 任务子系统状态转移如图 2 所示, $t_0, \dots, t_5 \subset T_i$ 表示某任务的各种状态, t_0 表示任务初始化, t_1 表示任务进入等待队列, t_2 表示任务得到分配, t_3 表示任务被拆解后的状态, t_4 表示任务执行完毕; $a, b, c, d, e, f, g \subset \Sigma$ 表示事件, a 表示任务由初始状态进入等待状态, b 表示由等待状态获得资源分配, c 表示存在满足条件的虚拟机资源, d 表示将任务 T_i 分解为多个子任务 T_{i1}, \dots, T_{in} 并进入子任务状态, e 表示执行任务的划分, g 表示为子任务重新分配虚拟机资源, f 表示任务正常执行完毕. 因正在执行的任务不一定可划分, 所以当出现任务执行超负荷时, 事件 e 的发生与否是不可控的. 因此, 系统的可控事件集为 $L_c = \{a, b, c, d, g, f\}$.

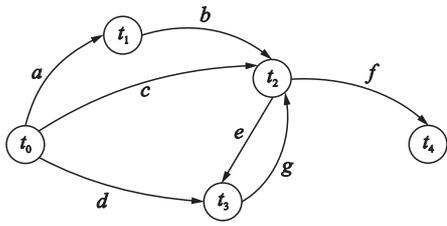


图2 任务子系统
Fig.2 The subsystem of tasks

虚拟机的状态从创建到销毁共有 7 个状态，如图 3 所示。其中， v_0 表示虚拟机的初始化， v_1 表示虚拟机失效状态， v_2 表示虚拟机扩容后状态， v_3 表示虚拟机正在执行任务， v_4 表示虚拟机超负荷， v_5 表示虚拟机迁移后状态， v_6 表示虚拟机被销毁。各事件分别为： i 表示当前虚拟机不能满足任务 T_x ， j 表示将扩容的虚拟机分配给任务， k 表示超负荷运行事件， h 表示为虚拟机分配任务， w 表示虚拟机正常提供服务至任务完成， o 表示虚拟机迁移， q 表示虚拟机在新宿主上提供正常服务直至任务完成， x 表示虚拟机失效事件， y 表示虚拟机人为或非人为的重启事件， u 表示虚拟机销毁事件因虚拟机是否将失效是不因人为改变，所以属于不可控事件。同理，因为某种原因导致任务执行出现超负荷运算也属于不可控事件。因此，系统可控事件集为 $L_c = \{i, j, h, w, o, q, u\}$ ，而不可控事件集为 $L_{uc} = \{x, y, k\}$ 。

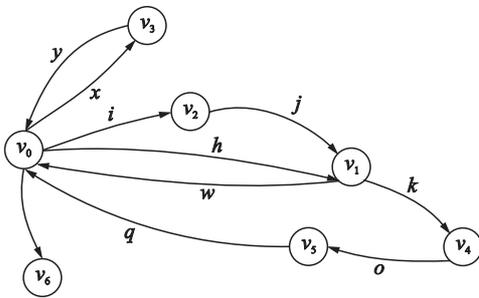


图3 虚拟机子系统
Fig.3 The subsystem of the virtual machines

物理服务器子系统主要有初始化、死机、分配虚拟机等状态，其状态转移如图 4 所示。其中， $m_0, \dots, m_n \in M_i$ 表示物理服务器的各种状态， m_0 表示初始化的物理服务器， m_1 表示物理服务器失效状态（如死机故障）， $m_2 \sim m_n$ 表示物理服务器上挂载的虚拟机； $\alpha, \beta, \chi, \eta \in \Sigma$ 表示系统事件， α 表示物理服务器故障事件， β 表示人为或非人为的物理服务器恢复事件， χ, γ 表示资源分配事件（如创建虚拟机）， η 表示资源回收事件（如虚拟机销毁事件）。同理，物理服务器的失效是不可控的，因此可控事件集 $L_c = \{\chi, \gamma, \eta\}$ 。

根据云任务动态配置虚拟机资源，如图 5 所示，系统中存在任务和资源全局控制器 C_g 和局部控制器 C_i, C_v, C_m ，在各局部控制系统中，当子系统处于不可接受状态时，通过控制可控事件的发生使子系统到达可接受状态，

若子系统中出现不可控事件，局部控制器向全局控制器反馈，由全局控制器协调控制其它子系统控制器，使整个系统最终到达可接受状态。

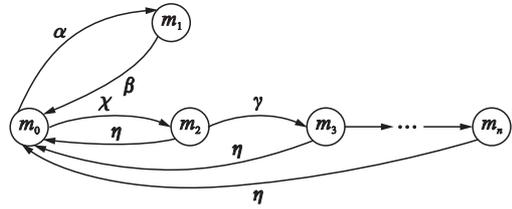


图4 物理节点子系统
Fig.4 The subsystem of the physical nodes

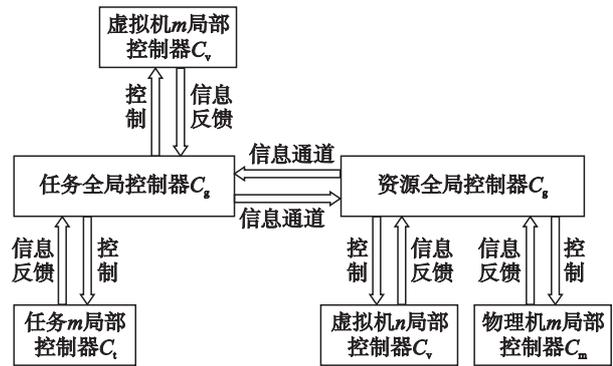


图5 分布式 DES 控制结构
Fig.5 The distributed DES control structure

基于分布式 DES 云资源动态调度策略主要作用为：

- (1) 保障资源利用率. 先通过全局控制器 C_g 计算空闲状态的虚拟机的各类资源量与任务的各类资源量之间的贴进度，通过对比找出最合适的虚拟机，再由局部控制器 C_i 控制事件 c 的发生；若监测到任务处于等待状态，则局部控制器 C_i 需要将消息传给全局控制器 C_g ，全局控制器控制局部控制器 C_v ，监控是否符合扩容条件：若符合，则控制事件 i 发生；否则，全局控制器 C_g 控制局部控制器 C_m ，控制事件 x 产生，生成新的虚拟机，局部控制器 C_i 控制事件 b 发生，实现资源分配。
- (2) 提高服务性能. 当监控到任务执行出现超负荷状态，判断任务是否可划分为子任务：若可分，则局部控制器 C_i 控制事件 eg 的产生划分子任务，并分配新虚拟机；若不可分，则将消息传递给全局控制器 C_g ，由 C_g 控制局部控制器 C_i ，控制事件 o 发生，将虚拟机迁移到性能较好的物理服务器。

4 云资源调度的算法实现

云资源动态调度流程图如图 6 所示，图中“VM”表示“虚拟机(virtual machine)”。

云资源动态调度算法描述如下：

- 输入：任务请求队列 T_i 、资源池中物理服务器 M_{i_m} 、虚拟机 V_{i_v} ；
- 输出：资源分配表 A 、资源利用率、任务执行时间之和和执行；

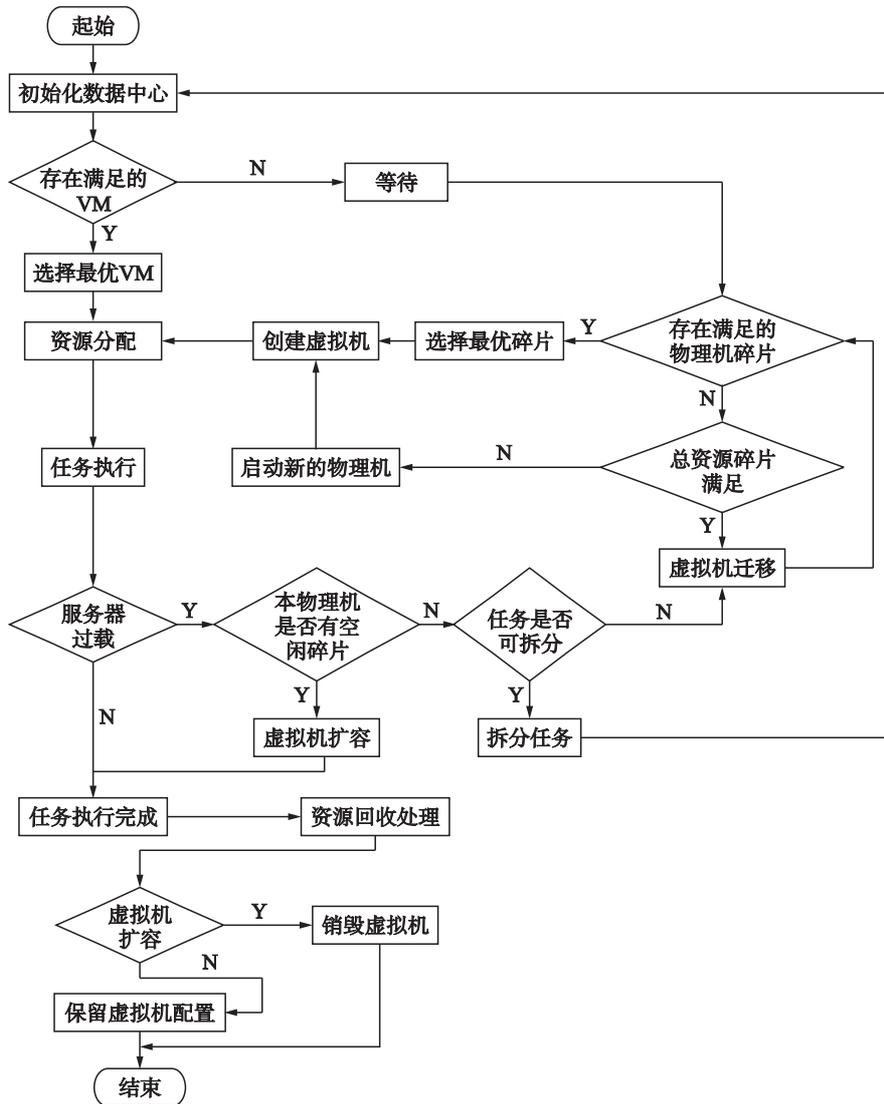


图 6 云资源动态调度流程图

Fig.6 Flow chart of the cloud resources dynamic scheduling

(1) 初始化资源分配:

初始化 T_{l_i} 、 V_{l_i} 、 M_{l_i} ;

$\forall T_x = \{R_{T_x}, 0\} \subset T_{l_i}$, $\exists V_y = \{R_{V_y}, M_i, 0\} \subset V_{l_i}$,

根据式(2)求 $D(x, y)$;

if $(\forall i)(R_i \subset R) \wedge (C_{R_i}^{T_x} < C_{R_i}^{V_y}) \wedge \min D(X, Y)$

then 将 V_y 分配给 T_x , 并修改资源载体状态,

$T_x = \{R_{T_x}, V_y = y\}$, $V_y = \{R_{T_x}, M_i, T_x = x\}$.

(2) 资源动态调度调整:

$\forall T_x = \{R_{T_x}, 0\}$

$\forall V_y = \{R_{T_x}, M_i, 0\} \subset V_{l_i} \mid \exists R_i \subset R, C_{R_i}^{T_x} > C_{R_i}^{V_y}$

if $\exists M_k = \{R_{M_k}, 1\}$

for $\{ \forall V_y = \{R_{V_y}, M_k, T_j \cup 0\}$

标记资源碎片 $R' = \{C_{R_i}^{M_k} - C_{R_i}^{V_y}, \dots, C_{R_i}^{M_k} - C_{R_i}^{V_y}\}$

if $\forall R_i \subset R(R_{T_x} + \tau < R') \wedge \min D(X, Y)$

then 建立新虚拟机并修改状态

$V^{new} = \{R_{T_x} + \tau, M_k, T_x\}$, $M_k = \{R_{M_k} - R_{T_x} - \tau, 1\}$

else break;

else $\forall M_k = \{R_{M_k}, 1\}$

for $\forall V_y = \{R_{V_y}, M_k, T_j \cup 0\}$

if $\forall R_i \subset R, R_{T_x} < \sum_k^n R'_k$, 虚拟机迁移;

while $\forall R_i \subset R, \exists R_{T_x} < R'_k$, 建立虚拟机并修改状态,

$V^{new} = \{R_{T_x} + \tau, M_k, T_x\}$, $M_k = \{R'_k - R_{T_x} - \tau, 1\}$

else 新建虚拟机, 分配任务并调整状态

$M^{new} = \{R_{M^{new}}, 1\}$, $V^{new} = \{R_{T_x} + \tau, M^{new}, T_x\}$.

(3) 优化调整:

$\forall V_y = \{R_{V_y}, M_k, T_x\} \subset V_{l_i}$, 由式(1)得响应时间 t ;

for 执行时间 $T > t$

if $\exists R'$ 满足 need
 then 扩容虚拟机并修改状态,
 $V_y = V_y^{\text{enlarge}} = \{R_{V_y} + \text{need}, M_k, T_x\}$;
 else $T_x = \{T_{x1}, T_{x2}, T_{x3} \dots T_{xi}\}$ 可拆分标识,
 then $\forall T_{xi} = \{R_{xi}, 0\}$, 转至第 1 步;
 else $\exists M_j, j \neq k, \forall R_i | C_{R_i}^{M_j} < C_{R_i}^{V_y}$,
 then V_y 迁移至 $M_j, V_y = \{R_{V_y}, M_j, T_x\}$.
 (4) 返回任务资源分配并计算资源利用率:
 $T_x = \{R_{T_x}, V_y\}$, 由式(3)求得资源利用率 U .
 (5) 全局资源回收调整:
 回收扩容的虚拟机 $V_y^{\text{enlarge}} = (R_{V_y}, M_k, 0) = (0, 0, 0)$.

5 仿真实验

实验采用墨尔本大学云计算与分布式系统实验室研发的 CloudSim^[18] 框架模拟云环境中的资源分配. 本文首先初始化数据中心 1 和数据中心 2, 每个数据中心共有 20 台物理服务器, 每个数据中心预先初始化 4 台物理服务器, 一个虚拟机节点. 初始化任务队列中有 200 个任务, 每个任务的长度区间为 [500, 2 000] 条指令数. 依次以 20 个任务递增的形式采集实验结果, 并将本文的调度算法实验结果与目前较为成熟且广泛应用的 Min-Min 调度算法^[19] 及 FCFS (first come first service) 调度算法^[20] 的实验结果进行对比, 3 种算法执行 200 个任务的完成时间和资源利用率的对比如图 7 和图 8 所示.

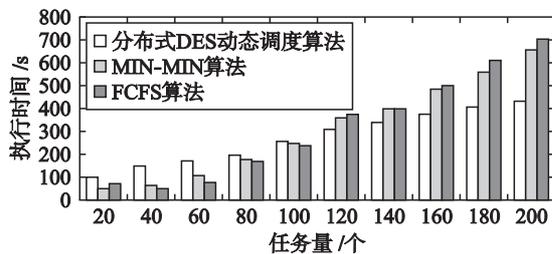


图 7 执行时间对比图

Fig. 7 The comparison of the execution time

参考文献

- [1] Mell P, Grance T. The NIST definition of cloud computing[J]. Communications of the ACM, 2010, 53(6): 50.
- [2] 刘鹏. 云计算[M]. 北京: 电子工业出版社, 2010.
Liu P. Cloud computing[M]. Beijing: Publishing House of Electronics Industry, 2010.
- [3] 于雷. 一种基于协作博弈的虚拟网络嵌入策略[J]. 信息与控制, 2016, 45(4): 449-455.
Yu L. Virtual network embedding strategy based on cooperative game theory[J]. Information and Control, 2016, 45(4): 449-455.
- [4] 龚素文, 艾浩军, 袁远明. 基于迁移技术的云资源动态调度策略研究[J]. 计算机工程与应用, 2014, 50(5): 51-54, 78.
Gong S W, Ai H J, Yuan Y M. Research of cloud resource dynamic scheduling strategy on migration technology[J]. Computer Engineering and Applications, 2014, 50(5): 51-54, 78.
- [5] 肖鹏, 胡志刚. 云环境中基于混合博弈的虚拟资源定价模型[J]. 计算机集成制造系统, 2014, 20(1): 198-206.
Xiao P, Hu Z G. Hybrid game based virtual resource pricing model in cloud environment[J]. Computer Integrated Manufacturing Systems, 2014, 20(1): 198-206.
- [6] 孙佳佳, 王兴伟, 高程希, 等. 云环境下基于神经网络和群搜索优化的资源分配机制[J]. 软件学报, 2014, 25(8): 1858-1872.
Sun J J, Wang X W, Gao C X, Huang M. Resource allocation scheme based on neural network and group search optimization in cloud environment[J]. Journal of Software, 2014, 25(8): 1858-1872.
- [7] Li K, Liu C, Li K, et al. A framework of price bidding configurations for resource usage in cloud computing[J]. IEEE Transactions on Paral-

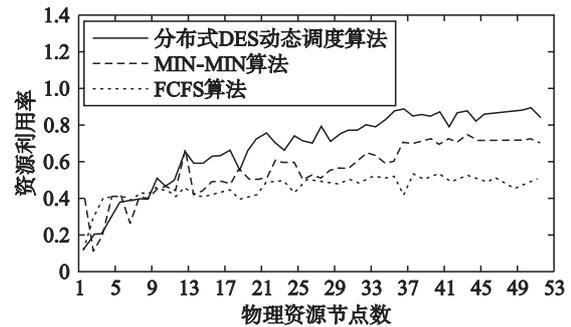


图 8 资源利用率对比图

Fig. 8 The comparison of the resources utilization

从图 7 可以看出, 本文提出的分布式 DES 动态调度算法在任务量较小时, 执行时间并没有表现出优势, 但随着任务量的增加, 其执行时间增速缓慢, 平均执行时间维持在 3 s 以内, 较 Min-Min 算法和 FCFS 算法都有明显优势. 在资源利用率方面, 从图 8 可知, Min-Min 算法由于性能强的资源可能会一直被占用, FCFS 算法也由于先到任务可能占用大量资源而导致其它任务响应超时, 而本文提出的分布式 DES 动态调度算法则通过获取资源载体状态, 能动态申请和调度资源, 因此, 随着任务数和资源节点数的不断扩大, 其资源利用率曲线上升并维持在 80% 左右, 具有较明显优势.

6 结语

本文提出了一种基于分布式离散事件系统监控理论的云资源动态调度方法, 能从全局角度动态对虚拟机进行合理扩充、迁移及释放, 并通过局部控制器与全局控制器之间的信息交互, 实现了云资源的动态调度. 实验表明, 随着任务数量的不断递增, 该调度方法在提高资源利用率及缩短任务执行时间上都具有较明显优势. 在后续的研究中, 本文将从云资源的价格收益及分配代价的角度出发, 结合启发式算法, 通过丰富资源属性对该动态调度方法进一步优化.

- lel and Distributed Systems, 2016, 27(8): 2168 – 2181.
- [8] Xiao Z, Song W, Chen Q. Dynamic resource allocation using virtual machines for cloud computing environment[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1107 – 1117.
- [9] 匡桂娟, 曾国荪. 一种基于时分复用的云资源管理方法[J]. 同济大学学报, 2014, 42(5): 782 – 789.
Kuang G J, Zeng G S. Time-division multiplexing-based cloud resource management methods[J]. Journal of Tongji University, 2014, 42(5): 782 – 789.
- [10] 师雪霖, 徐格. 云虚拟机资源分配的效用最大化模型[J]. 计算机学报, 2013, 36(2): 252 – 262.
Shi X L, Xu G. Utility maximization model of virtual machine scheduling in cloud environment[J]. Chinese Journal of Computers, 2013, 36(2): 252 – 262.
- [11] 胡芹, 刘富春. 基于离散事件系统的云资源分配优化控制[J]. 广东工业大学学报, 2016, 33(1): 29 – 35.
Hu Q, Liu F C. Optimization control of cloud resource allocation based on DES[J]. Journal of Guangdong University of Technology, 2016, 33(1): 29 – 35.
- [12] Cassandras C, Lafortune S. Introduction to discrete event systems[M]. New York, USA: Springer Science + Business Media LLC, 2008: 199 – 211.
- [13] Schmidt K, Breindl C. Maximally permissive hierarchical control of decentralized discrete event systems[J]. IEEE Transactions on Automatic Control, 2011, 56(4): 723 – 737.
- [14] Liu F, Lin H. Reliable supervisory control for general architecture of decentralized discrete event systems[J]. Automatica, 2010, 46(9): 1510 – 1516.
- [15] 刘富春. 非确定型离散事件系统双模拟控制的实现[J]. 控制理论与应用, 2015, 32(1): 75 – 79.
Liu F C. Realization of bisimilarity control of nondeterministic discrete event systems[J]. Control Theory & Applications, 2015, 32(1): 75 – 79.
- [16] 向洁, 丁恩杰. 基于虚拟机调度的数据中心节能优化[J]. 计算机应用, 2013, 33(12): 3331 – 3353.
Xiang J, Ding E J. Energy-saving optimization in datacenter based on virtual machine scheduling[J]. Journal of Computer Applications, 2013, 33(12): 3331 – 3353.
- [17] 马汉达, 郝晓宇, 马仁庆. 基于 Hadoop 的并行 PSO-kmeans 算法实现 Web 日志挖掘[J]. 计算机科学, 2015, 42(6A): 470 – 473.
Ma H D, Hao X Y, Ma Y Q. Parallel PSO-kmeans algorithm implementing web log mining based on Hadoop[J]. Computer Science, 2015, 42(6A): 470 – 473.
- [18] Humane P, Varshapriya J N. Simulation of cloud infrastructure using CloudSim simulator; A practical approach for researchers[C]//International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials. Piscataway, NJ, USA: IEEE, 2015: 207 – 211.
- [19] Miriam D D H, Easwarakumar K S. A double Min-Min algorithm for task metascheduler on hypercubic P2P grid systems[J]. International Journal of Computer Science Issues, 2010, 7(4): 8 – 18.
- [20] Hsieh S Y, Chen C T, Chen C H, et al. Novel scheduling algorithms for efficient deployment of map reduce applications in heterogeneous computing environments[J]. IEEE Transactions on cloud Computing, 2016; DOI 10.1109/TCC. 2016.2552518.

作者简介

刘富春(1971 –), 男, 博士, 教授. 研究领域为控制理论与控制工程, 数理逻辑与模糊系统.

胡 芹(1989 –), 女, 硕士生. 研究领域为离散事件系统的监督控制理论与应用.