

一种新的 DCCEM 动态云任务分配算法研究

刘军华,蔡卫红,雷超阳

(湖南邮电职业技术学院, 湖南 长沙 410015)

摘 要: 提出了一种动态的云任务分配算法架构模型,其算法能将云任务平均分配到云群服务器中,充分考虑了不同云服务器针对不同业务能力,解决了云服务器的负载平衡,且能高效率地使用 CPU 资源,让使用者得到较高的服务效率,并让云群环境的计算及存储资源得到了有效管理.通过仿真比较分析,该技术是有效的,既能充分利用系统资源,又能实现资源的自动调节和负载均衡.

关键词: 云任务;分配;算法;研究

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2013)10-0058-04

A New DCCEM Dynamic Cloud Task Allocation Algorithm Research

LIU Jun-hua, CAI Wei-hong, LEI Chao-yang

(Hunan Post and Telecommunication College, Changsha 410015, China)

Abstract: This paper proposes a dynamic cloud task allocation algorithm model which can equally allocate the cloud tasks into the cloud group servers and gives full consideration to the different cloud servers for different business abilities. The algorithm solves the load balance of the cloud servers and uses the CPU resources efficiently so that the user can obtain higher service efficiency and the effective management of the cloud cluster environment calculation and storage resources can be achieved. By simulation comparison analysis, the technique is proved to be effective, which can not only make full use of the system resources, but also realize the automatic regulation and load balancing of the resources.

Key words: cloud task; allocation; algorithm; research

1 引言

随着互联网技术的快速发展,互联网访问数据量出现了高速增长,特别是随着 Web2.0 的发展,导致了网络数据处理能力的相对不足,网络资源的利用率处于一种不平衡状态.这样,一种好的云任务分配算法及系统架构模型对于充分利用系统资源、实现资源的自动调节和负载均衡、发挥云计算的优点具有重要的意义.常见的云任务分配算法有 Random 算法和 RRM 算法^[1],这两种算法在一定程度上都可提高工作效率,但 Random 算法和 RRM 算法都没有考虑云群服务器的能力去分配任务,无法高效率的使用 CPU 资源.本文提出一种 DCCEM

动态云任务分配算法,其算法在考虑当前不同云群服务器对不同云任务服务能力不同的基础上尽量将云任务平均分配到云群服务器中,更能高效率使用 CPU 资源.

2 DCCEM 云任务分配算法研究

2.1 Random 云任务分配算法

在云任务分配中,Random 算法原理如图 1 所示,网络使用者输入的云任务会首先进入 Load Balancer 负载均衡器队列进行暂存,然后 Load Balancer 按照 Random 分配算法以随机方式选择一个目的云群服务器(S_i)进行云任务的派送与处理.

收稿日期: 2012-11-22; **修回日期:** 2013-01-16

基金项目: 湖南省教育厅资助科研项目(10C720)

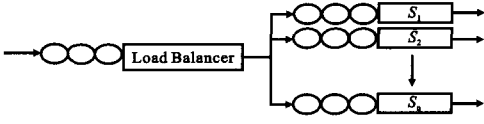


图1 Random云任务分配算法架构模型图

2.2 TRRM云任务分配算法

针对 Random 云任务分配算法没有充分考虑云群服务器负载过重、云群服务器资源不足,云任务遗失率过高的缺点,在 Random 云任务分配算法架构模型的基础上,提出了一种具有高吞吐量的 TRRM 云任务分配算法^[2],其算法架构模型如图2所示。

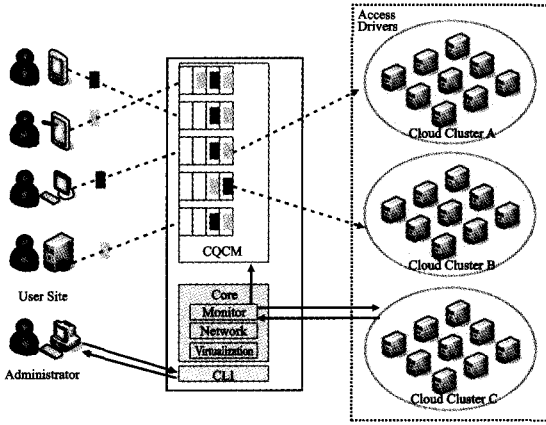


图2 TRRM云任务分配算法架构模型图

图2中,Core主要用于云计算的管理,包括三个子元件:Network子元件通过虚拟专用网络VPN将虚拟设备相互连接,用于虚拟资源网络的建立和管理;Monitor子元件通过监控代理来监控目的云群服务器资源的使用率,并将有用信息传给CQCM;Virtualization子元件用于集中管理控制下层虚拟设备的软件。CLI(命令行界面)用于给网络操作员提供操作界面来建立云群计算服务或其它相关设置。Access Drivers安装在实体资源中,它通过XEN虚拟机软件和KVM虚拟机软件来虚拟化实体资源,同时,它还通过插件让上层Core中的Virtualization来被统一管理控制。CQCM(交叉队列容量管理)用于云任务的接收和云任务的分配。

CQCM的基本工作原理如图3所示^[3]。CQCM由N个云任务输入Input i ($0 \leq i \leq N-1$)和N个云群服务器Cloud Cluster (CC_j) ($0 \leq j \leq N-1$)组成,每个Input i 中的N个Group Queue (GQ_{ij})用于暂存从Client送来的任务,而Cloud Cross Buffer (CXB_{ij})用于暂存到云群服务器 CC_j 的任务。Cloud Client Group (CCG_i)代表云服务使用者所聚集的分群,当 CCG_i 分群有用户需云服务时, CCG_i 先将任务送至Input i ,CQCM然后会根据任务的目的地把任

务送Input i 中的 GQ_{ij} 来暂存,为避免任务的抢先传送, GQ_{ij} 中的云任务再通过Threshold Round Robin Match (TRRM)分配算法传给交叉缓冲区和 CXB_{ij} 缓存。

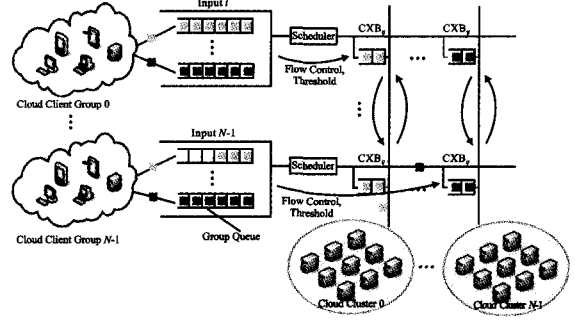


图3 CQCM工作原理图

为提高云计算的服务质量,通过对 CC_j 使用Threshold 门限机制来避免云任务被分配到云群服务器资源不足的 CC_j 中。CQCM首先初始化计算 CC_j 的 CCt_j ,其计算公式如式(1)所示,同时通过Value Function (VF_j)的最大值 $VF_{j,max}$ 来计算 CC_j 的最大服务能力,在服务过程中Monitor子元件通过Monitor Agent 监控代理定期收集 CC_j 的CPU、Memory和存储器Storage的剩余使用量等信息,再根据式(2)计算出实时的 VF_j 后传给CQCM, CQCM会根据 VF_j 和 CCt_j 去进行判断,只有当 $CCt_j > VF_j$ 时 GQ_{ij} 的任务才有资格传送到 CXB 中。

$$CCt_j = VF_{j,max} \times \text{Threshold} \quad (1)$$

Threshold: 门限值的百分比;

CCt_j : CC_j 的门限值;

$$VF_j = \sum_{i=1}^n W_i f(x_{ij}) = W_1 f(x_{1j}) +$$

$$W_2 f(x_{2j}) + \dots + W_n f(x_{nj}) \quad (2)$$

x_{ij} : CC_j 决策变数 i 的值;

$f(x_{ij})$: CC_j 决策变数 i 的函数值;

VF_j : 估测 CC_j 的实时能力值;

W_i : 每个变异值的权重,且 $\sum_{i=1}^n W_i = 1$;

$VF_{j,max}$: 估测 CC_j 的最大能力值。

为降低Input i 和 CXB_{ij} 之间任务的丢失率,在Input i 和 CXB_{ij} 之间加入TRRM任务流分配机制,其TRRM分配算法流程如下所示^[4]。当 CXB_{ij} 无法暂存任务时, CXB_{ij} 将反馈信息给Input i ,使 GQ_{ij} 暂时不传送任务给 CXB_{ij} 。当条件满足时, GQ_{ij} 才传送任务给 CXB_{ij} 。

TRRM算法如下:

Threshold Round Robin Match

```

1 while  $GQ_{ij}$  do
2    $j = (j+1) \text{ Mod } N$ 
3   if  $GQ_{ij} \sim \text{Flow control}$  then
4     if  $CC_{t_j} > VF_j$  then
5        $CBX_{ij} \leftarrow GQ_{ij}$ 
6     end if
7   end if
8 end while

```

2.3 DCCEM 云任务分配算法

尽管 TRRM 云任务分配算法可提供较高的服务质量,但仍然忽略了云任务服务类型的不同和不同云任务服务类型对云计算资源消耗不同两个方面影响,因而可能造成满载的状况.为能让云任务动态有效的分配到最合适的云群中去执行并且能够均匀的使用云计算资源^[5],提出一种新的动态云任务分配算法即 DCCEM 算法.

其算法架构模型如图 4 所示.此架构也是由 N 个 Input 和 Colud Cluster 所组成,每一个 Input 都有 N 个 Group Queue(GQ_{ij})用来暂存到 CC_j 的云任务.当使用者所聚集成的人群 Colud Client Group (CCG_i)有云任务服务需求时, CCG_i 会通过 DCCEM 算法将云任务发送到适合的 Input,系统将通过云任务的服务类型来查询 Load-Balancing List(LB_j),DCCEM 再根据 Load-Balancing List 中的信息选择适合云任务服务类型的 CC_j ,将云任务送到 Input 中的 GQ_{ij} 进行暂存,再将云任务传送到 CC_j 中.

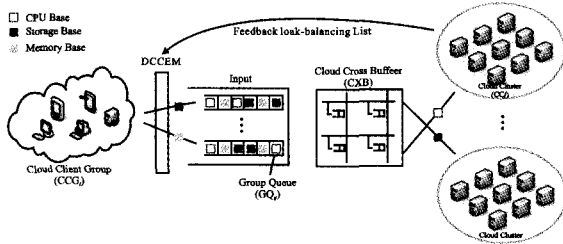


图 4 DCCEM 动态云任务分配算法架构模型图

每个 CC_j 通过式(3)来统一量化不同云服务类型的能力值.

$$R_{\text{type}} = \frac{\text{resource}_{\text{type}}}{\text{resource}_{\text{type max}}} \quad (3)$$

type : 服务类型;

resource_{type} : type 类资源目前的服务能力值;

resource_{type max} : type 类资源最大可提供的服务能力值;

再根据 CPU Base、Memory Base、Storage Base 云任务服务类型代入式(4)来求得 LB_j .

$$LB_j = R_{\text{storage}} \times W_1 + R_{\text{cpu}} \times W_2 + R_{\text{memory}} \times W_3 \quad (4)$$

LB_j : 为不同云任务服务类型 CC_j 的总的服务能力值;

i : 为函数中的第 i 个决策因子,共有 N 个决策因子;

$$W_i : \sum_{i=1}^n w_i = 1, \text{ 为各个决策因子的权重值.}$$

因不同的云任务服务类型所需求的能力不同,对应的权重值也不同.例如对于 Storage Base 云任务服务类型业务,所需的 Storage 的比重就会较高,则新的 VF_j 的权重就变为式(5).

$$VF_j = \text{Storage} \times (0.8) + \text{CPU} \times (0.1) + \text{Memory} \times (0.2) \quad (5)$$

通过 Monitor Agent 将 CC_j 的负载状态汇报给 DCCEM 的 Load-Balancing List. DCCEM 根据 Load-Balancing List 来动态调整云任务分配,从而使 CC_j 的资源达到负载平衡.

DCCEM 云任务分配算法流程如下:

Dynamic Cloud Cluster Election Mechanism

```

1 while  $CGB_i$  do
2    $\text{Min}(LB_j)$ 
3   if  $GQ_j \neq \text{full}$  then
4      $GQ_j \leftarrow \text{Task}$ 
5   end if
6 end while

```

3 仿真比较分析

通过以上 Random、TRRM、DCCEM 三种云任务分配算法的基本理论分析,很显然,DCCEM 分配算法最优.为比较 Random、TRRM、DCCEM 三种分配算法的系统性能差异^[6],通过随机函数产生 CPU Base、Memory Base、Storage Base 三种不同云任务至 8 个实验分群中,分别选择 Random 分配算法、TRRM 分配算法和 DCCEM 分配算法做 MATLAB 仿真分析,每 30 秒分别统计整体云群服务器负载平均水平、云任务已服务数量和 Storage 的 Mean Average Deviation(MAD)三大指标数据.通过云群服务器平均负载水平可大致了解整体云群服务器的 CPU 和 Memory 的使用率,通过云任务已服务数量(Number of Task)指标可了解不同分配算法的工作效率,通过检测 Storage 的 MAD 可大致了解 Storage 档案的存储负载不均衡状态.其 Storage 的 MAD 的计算公式如下:

$$\frac{1}{n} \sum_{j=1}^n | \text{Storage}_j - m(\text{Storage}) | \quad (6)$$

j :分群的编号;
 $Storage_j$:分群 j 的 Storage 使用量;
 $m(Storage)$:所有分群的平均 Storage 使用量;
 n :分群的数量.

通过仿真比较在不同云资源分配算法下 CPU Load、Memory Load、Storage MAD、Number of Task 四个方面的性能,其仿真效果如图 5 所示.

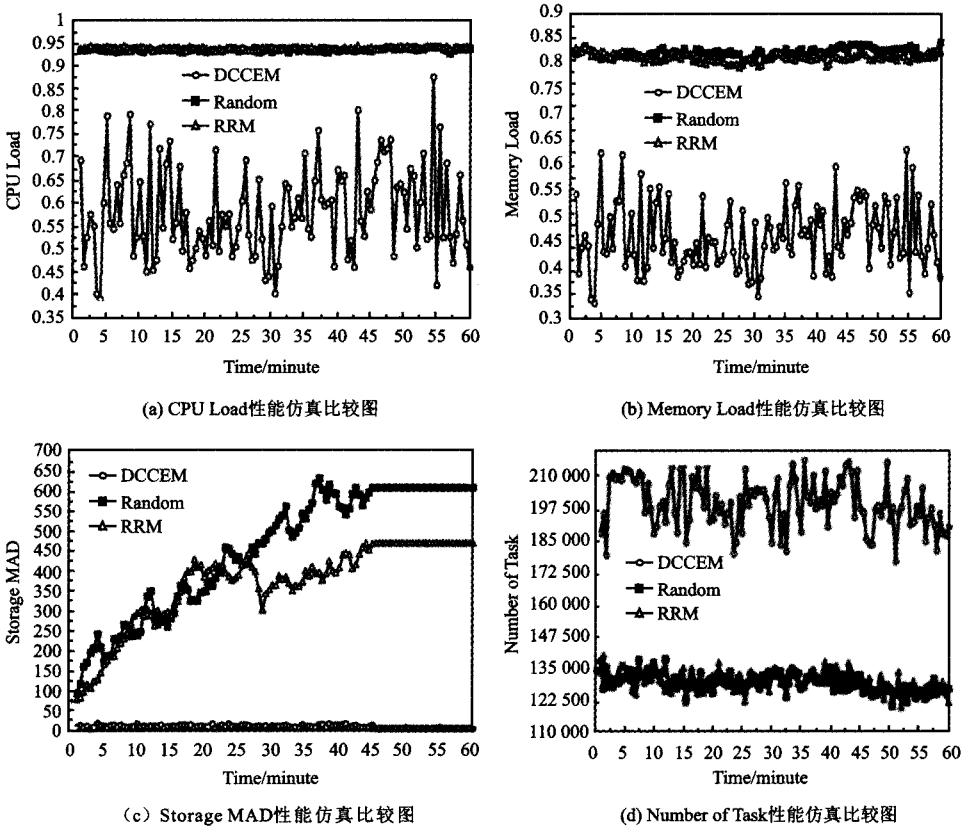


图 5 不同云任务分配算法不同性能仿真比较图

由图 5(a)可知,DCCEM 算法比 Random 和 TRRM 算法好. DCCEM 算法比 Random 和 TRRM 算法工作效率高,进而提高了任务服务数量.

由图 5(b)可知,DCCEM 算法比 Random 和 TRRM 算法好. DCCEM 算法能将云任务分配至最适合的云群服务器,让拥有足够 CPU 和 Memory 资源的云群服务器接受工作,Memory 资源使用效率高,故 DCCEM 不会因 CPU 满载而浪费了大量的 Memory 资源.

由图 5(c)可知,DCCEM 算法比 Random 和 TRRM 算法好. DCCEM 算法不仅能将云任务平均的分配到云群服务器中,而且能选择 Storage 资源使用率最低的群服务器,所以 DCCEM 算法具有最低的 MAD.

由图 5(d)可知,DCCEM 算法比 Random 和 TRRM 算法好. DCCEM 算法能够将更多的云任务给云群服务器,DCCEM 算法中工作量忽多忽少的

状况是因为有些工作需要较大的 CPU 需求,造成服务的数量突然降低.

4 结束语

通过比较 TRRM 和 DCCEM 分配算法的原理与仿真性能可知,TRRM 云任务分配算法架构模型虽然达到了降低云任务丢失率和云任务队列等待时间的效果,拥有较平均 Task Loss Rate 和 Queue Wait Time 的 TRRM 分配算法适合在多种而且复杂的网络服务环境中使用^[7]. 但没有考虑不同云服务器对不同业务的能力,DCCEM 算法解决了云群服务器负载平衡的问题,服务效率及资源管理问题. 仿真实验结果表明,本方法可以让使用者得到较佳的服务效率,并让云群环境的计算及存储资源能得到有效的管理,达到负载均衡. 此外,由于资源可以达到负载均衡,则在任务的分配上,也能更有效率的进行.

(下转第 65 页)

具有更好的安全性.

表 1 相邻像素相关系数

方向	原始图像	加密后图像
水平	0.953 3	0.003 3
垂直	0.993 3	0.012 9
对角线	0.904 0	0.029 3

6 结束语

本文提出了一种新的六维混沌算法,其结构是由 Lorenz 混沌算法和 Duffing 混沌算法融合后形成的,由于两个混沌算法的相互作用,融合后的新六维混沌算法动力学结构复杂,算法中共有 13 个可变参数和变量可作为密钥来使用,密钥空间大,更难被破译.结合 Arnold 变换对彩色图像进行加密,理论分析和仿真实验证实了所设计的加密算法的性能和加密效果.

参考文献:

- [1] 卢辉斌,张鹏,国宪鹏,等.一种新的基于双混沌系统的图像加密算法[J].计算机工程与应用,2012,48(2):90-92.
- [2] 王相生,甘骏人.一种基于混沌的序列密码生成方法[J].计算机学报,2002(4):16-21.
- [3] 陈华,冯登国,范丽敏.一种关于分组密码的新的统计检测方法[J].计算机学报,2009(4):7-13.
- [4] 刘金梅,丘水生.基于超混沌系统的图像加密算法的安全性分析[J].计算机应用研究,2010(3):1042-1044.
- [5] 廖晓峰,肖迪,张勇,等.2009 混沌密码学原理及应用[M].北京:科学出版社,2009(7):135-179.
- [6] 于津江,曹鹤飞,许海波,等.复合混沌系统的非线性动力行为学分析[J].物理学报,2006,55(1):29-34.
- [7] 甘建超,肖先赐.混沌的可加性[J].物理学报,2003,52(5):1085-1090.

作者简介:

张帆 男,(1981-),硕士,讲师.研究方向为计算机技术.

(上接第 61 页)

参考文献:

- [1] 吕良干.云计算环境下资源负载均衡调度算法研究[D].乌鲁木齐:新疆大学,2010:25-40.
- [2] Costanzo A di, Assuncao M D de, Buyya R. Harnessing cloud technologies for a virtualized distributed computing infrastructure[J]. IEEE journal of Internet Computing, 2009, 13(5):24-33.
- [3] Stage A, Setzer T, Bichler M. Automated capacity management and selection of infrastructure as a service providers[C]//IFIP/IEEE International Symposium on Intrgrated Network Management Workshops. New York, 2009.
- [4] Lawton G. Developing software online with plantform as a service technology[J]. IEEE. Journal of Computer, 2008, 41(6):13-15.
- [5] Hou Z, Zhou X, Gu J, et al. ASAAS: application software as a service for high performance cloud computing

[C]// 12th IEEE International Conference on High Performance Computing and Communications. Melbourne, 2010:156-163.

- [6] Head M R, Sailer A, Shaikh H, et al. Towards self-assisted troubleshooting for the deployment of private clouds[C]// 5th IEEE International Conference on Cloud Computing. Miami, 2010:156-163.
- [7] Vouk M A. Cloud computing - issues, research and implementations[C]//30th International Conference on Information Technology Interfaces. USA: Northcarolina University, 2008:31-40.

作者简介:

刘军华 男,(1979-),硕士,讲师.研究方向为软件工程、图像处理、信息安全与保密、人工神经网络.

雷超阳 男,(1971-),博士,教授.研究方向为图像处理、模式识别.