一种基于排队模型和差分进化算法的Web集群节能部署方案

熊 智 刘卫军 崔张伟
(汕头大学计算机科学与技术系 汕头 515063)

摘 要 Web集群的节能问题是急需解决的重要问题,为此提出了一种Web集群节能部署方案。该方案同时采用动态调整CPU频率和动态开关服务节点的措施来进行节能,运用M/G/1 PS排队模型对服务节点进行建模,将Web集群节能部署问题转化为带约束的规划问题。针对该规划问题,提出了一种差分进化求解算法,在迭代过程中采用贪婪思想对不满足约束条件的解进行修正,并将贪婪算法得到的解放入初始种群以提高收敛速度。仿真测试验证了该算法的可行性和有效性。

关键词 Web集群,节能部署,排队模型,差分进化算法
中图法分类号 TP393 **文献标识码** A

Energy-efficient Deployment Scheme Based on Queue Model and Differential Evolution Algorithm for Web Cluster

XIONG Zhi LIU Wei-jun CUI Zhang-wei
(Department of Computer Science and Technology, Shantou University, Shantou 515063, China)

Abstract Energy-saving of Web cluster is an urgent problem to be solved, hence an energy-efficient deployment scheme for Web cluster was proposed. The scheme adopts both CPU dynamic frequency scaling and server node dynamic switching on/off mechanisms to conserve energy. It uses M/G/1 PS queue model to simulate Web server, and transforms the energy-efficient deployment problem of Web cluster to a constrained programming problem. Aiming at the problem, a differential evolution algorithm was proposed to solve it. During the iteration processes, the greedy idea was used to correct the solutions which do not satisfy the constraints. The solutions got by the greedy algorithm were put into the initial population to fasten the convergence speed. Simulation tests show the feasibility and effectiveness of the proposed algorithm.

Keywords Web cluster, Energy-efficient deployment, Queue model, Differential evolution algorithm

1 引言

Web集群服务器(简称Web集群),以其可扩展、高性能、高可靠、高性价比以及对用户透明等特性,已成为当前应用最为广泛的一种提高Web服务器性能和可靠性的解决方案。因此,各种大规模的Web应用,包括电子商务和SaaS(Software as a Service,软件即服务,或软件运营服务)等,通常都采用Web集群对外提供服务。Web集群已成为大规模Web应用基础设施的一部分,通常部署在IDC(Internet Data Center,互联网数据中心)中。

在全球气候日趋变暖和能源日趋紧张的背景下,节能减排、保护环境已经成为各个行业都必须重视的问题。各种数据表明,计算机正在吞噬大量能源[1],IDC中Web集群的节能问题亟待解决。一方面,如今Web集群中的请求基本都是动态请求,这对集群的性能要求很高,运营商在部署集群时通常是按峰值容量部署的,但在实际运行中,大多数时候服务器的利用率都很低,浪费了大量能耗。另一方面,IDC电能消耗所带来的运营成本已成为Web应用运营商的主要开支之一,而服务器能耗是IDC能耗的重要组成部分。因此,如何根据实际负载状况,合理部署Web集群中的节点使得集群的能耗最小,是Web应用运营商们急需解决的问题。

通常Web集群的节能是和QoS(Quality of Service,服务质量)联系在一起的,且可分为两类:另一类是在保证能耗(不超过限定值)的前提下尽可能地提升QoS;一类是在保证QoS的前提下尽可能地降低能耗。本文的研究属于第二类,下面只讨论该类研究的现状。动态调整CPU频率和动态开关服务节点是Web集群节能的两种重要措施。文献[2]仅通过动态调整服务节点的CPU频率进行节能,但服务节点中除了CPU以外,其他部件也耗电,并且节点可供选择的频率是离散的且数量很有限,即便是调到最低频率,CPU的耗电量也不小。文献[3-5]仅通过动态开关服务节点进行节能,但在开启的节点中有的节点可能负载很小,长期让其以最高频率运行将浪费大量能耗。因此,只有两种措施同时采用才能取得最好的节能效果。文献[6]研究的是同构集群的节能,不具有

到稿日期:2012-12-05 返修日期:2013-04-19 本文受国家自然科学基金项目(61202366),广东省自然科学基金项目(S2012010010023)资助。
熊 智(1978-),男,博士,副教授,主要研究方向为集群服务器、云计算、信息系统安全,E-mail:witmmx@gmail.com;刘卫军(1988-),男,硕士生,主要研究方向为集群服务器、服务质量保证;崔张伟(1988-),男,硕士生,主要研究方向为云计算、数字版权保护。

一般性。文献[7]研究的是异构集群的节能，它将服务节点分成多个同构的服务节点组，并让同组中的各节点工作在相同的频率上并且承担相同的负载，但它没有给出如何在不同的组中分配负载的方案。

近几年来，有很多研究人员基于混合整数规划(Mixed Integer Programming，MIP)来研究集群的节能问题[8-12]。他们将服务节点的开关、服务节点的频率调整，以及 QoS 保证约束一并考虑，通过定义一些 0-1 变量和实数变量，将 QoS 保证等作为约束条件，将功耗最小作为目标，从而将集群节能的节点部署问题转化为 MIP 问题，然后求解。但他们都需要对一个节点定义两个或两个以上的规划变量，对于大规模的集群来说，这将给在线求解带来困难。在此类文献中，文献[8,11]没有给出求解方法；文献[10]采用贪婪思想求解，只能得到次优解；文献[9]采用广义 Benders 分解法求解，当集群规模很大时，求解计算量将很大；文献[12]考虑到求解计算量大的问题，提出"先离线求得各种负载下的解，并构建负载和解的对应表，然后供在线查询"的方法，但该方法由于存在如下不足而不具有可操作性：1)要想在线得到的解比较精确，表中负载的间隔必须很小，表的行数将巨大；2)如果表中的某个节点坏掉了，或者想要增加一个节点，则需要重新构建整个表，这将耗费大量的时间。

因此本文提出了一种基于排队模型和差分进化算法的 Web 集群节能部署方案，该方案既动态调整 CPU 频率又动态开关服务节点，可适用于异构集群，并且还给出了请求调度算法。虽然在该方案中也涉及规划问题的求解，但由于规划变量少，并且我们为其提出了高效的差分进化求解算法，因此即使运用于大规模的集群，其规划问题仍然可以在线进行求解。

2 Web 集群节能部署问题的数学描述

2.1 模型及假设

本文的研究对象是如图 1 所示的 Web 集群模型。

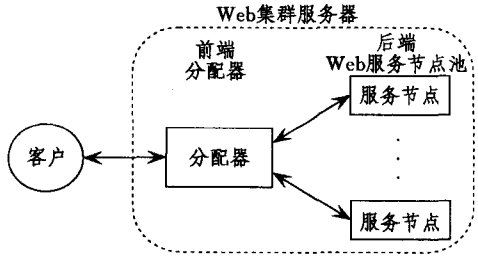

图 1 Web 集群模型

假设 QoS 保证的目标为：请求的平均响应时间不超过某个给定的 SLA(Service Level Agreement，服务等级协议)值。一个请求的响应时间包括两个部分：分配器的处理时间和服务节点的响应时间。已有很多关于分配器扩展的研究可解决其瓶颈问题，因此本文假设分配器的处理时间基本固定，对其不予考虑。

假设后端每个服务节点的 CPU 频率可调整，各节点可以单独关闭和开启，并假设请求的到达为泊松过程，服务节点服务请求的时间服从一般分布，又由于服务节点是采用时间片轮转的方式同时服务多个请求，因此本文采用 M/G/1 PS 排队模型[13]建模每个服务节点。

分配器基于概率来调度请求给各服务节点。与众多文献一样，为了做到集群的实时部署，本研究也需要预测下个周期的负载。每个周期结束时，分配器将预测下个周期的请求到达速率，并根据预测结果调整各个服务节点的开关和频率以及各个服务节点的调度概率，以使在保证 QoS 的前提下，所有服务节点的总功耗达到最小。

2.2 数学描述

假设 Web 集群中有 N 个服务节点 $S_i(1\leqslant i\leqslant N)$。节点 S_i 有 $F(i)$ 个可选的离散频率，从小到大排列的第 k 个为 $F_{i,k}$。节点 S_i 工作在频率 $F_{i,k}$ 时的空载(CPU 利用率为 0)功耗记为 $PI_{i,k}$，满载(CPU 利用率为 100%)功耗记为 $PB_{i,k}$。另外，节点 S_i 关闭时的待机功耗设为 PS_i。

假设预测得到的下个周期的请求到达速率为 λ，节点 S_i 的调度概率为 p_i，显然 $0\leqslant p_i\leqslant 1$，且 $\sum_{i=1}^{N} p_i=1$，那么下个周期节点 S_i 的请求到达仍为泊松过程，且请求到达速率为 $p_i\lambda$。当 $p_i=0$ 时，节点 S_i 将被关闭。

假设主频为 1GHz 的服务节点服务请求的平均速率为 R，节点 S_i 的 CPU 频率为 $F_{i,k}$(以 GHz 为单位)，那么 S_i 服务请求的平均速率为 $F_{i,k}R$。根据 M/G/1 PS 排队模型的公式[13]，节点 S_i 的平均响应时间为

$$T=\frac{1}{F_{i,k}R-p_i\lambda} \tag{1}$$

假设要保证节点的平均响应时间不超过 T_{SLA}，那么，

$$T=\frac{1}{F_{i,k}R-p_i\lambda}\leqslant T_{SLA} \tag{2}$$

进而可得到

$$p_i\leqslant\frac{F_{i,k}R-\frac{1}{T_{SLA}}}{\lambda}\leqslant\frac{F_{i,F(i)}R-\frac{1}{T_{SLA}}}{\lambda} \tag{3}$$

以及

$$F_{i,k}\geqslant\frac{\frac{1}{T_{SLA}}+p_i\lambda}{R} \tag{4}$$

那么，

1)当 $p_i\in(0,1]$ 且满足式(3)时，要做到节能，节点 S_i 的 CPU 频率 $F_{i,f(i)}$ 应设置为大于式(4)右边的最低频率，即

$$f(i)=\begin{cases}1, & F_{i,1}\geqslant\dfrac{\frac{1}{T_{SLA}}+p_i\lambda}{R}\\ k, & F_{i,k-1}<\dfrac{\frac{1}{T_{SLA}}+p_i\lambda}{R}\leqslant F_{i,k},2\leqslant k\leqslant F(i)\end{cases} \tag{5}$$

此时，节点 S_i 的 CPU 利用率为

$$\eta_i=\frac{p_i\lambda}{R\times F_{i,f(i)}} \tag{6}$$

其功耗为

$$Power_i=(1-\eta_i)PI_{i,f(i)}+\eta_i BP_{i,f(i)} \tag{7}$$

2)当 $p_i=0$ 时，节点 S_i 将被关闭，其功耗为 $Power_i=PS_i$。

又记

$$p_{i,MAX}=\mathrm{Min}\{\frac{F_{i,F(i)}R-\frac{1}{T_{SLA}}}{\lambda},1\} \tag{8}$$

那么，Web 集群 QoS 保证下的节能部署问题可描述成如下规划问题：

$$\begin{cases}\min \sum_{i=1}^{N} Power_i \\ \text{s.t. } \sum_{i=1}^{N} p_i = 1 \\ 0 \leqslant p_i \leqslant p_{i,MAX}, i=1,2,\cdots,N\end{cases} \tag{9}$$

式中，p_i 为待求的解，$F_{i,k}$ 为已知的值，T_{SLA} 为给定的值，λ 通过预测得到，$PI_{i,k}$、$PB_{i,k}$ 和 PS_i 可通过实验事先测得，R 也通过实验事先测得；第一个约束条件为概率和约束，第二个约束条件为上下界约束。特别地，在该规划问题中，每个节点只需要一个规划变量，从而规划变量的数量比同类文献少，便于在线求解。

3 Web 集群节能部署问题的求解

3.1 贪婪算法求解

定义 1(功耗因子) 功耗因子表示节点单位频率所消耗的平均功耗，节点 S_i 的功耗因子计算如下：

$$q_i = \frac{1}{F(i)} \sum_{k=1}^{F(i)} \frac{PI_{i,k} + PB_{i,k}}{2F_{i,k}} \tag{10}$$

贪婪算法的基本思想为：将请求尽量调度给功耗因子小的节点，其具体算法如下：

```
所有 pi=0;
psum=0;
while(psum<1) {
  从剩下(即 pi=0)的节点中选择功耗因子最小的节点 Si;
  pi=Min{pi,MAX,1-psum};
  psum=psum+pi;
}
```

3.2 差分进化算法求解

求解带约束规划问题的算法可分为两类：精确算法和进化算法。精确算法通常对目标函数和约束条件有一定的要求，且容易陷入局部最优，并且规模不能太大。进化算法则没有任何特殊要求，不需要借助问题的特征信息，且可求得全局最优解。近年来有许多学者将进化算法如遗传算法、模拟退火算法和粒子群优化算法等用于求解规划问题，取得了满意的效果。

差分进化(Differential Evolution，DE)算法[14]是一种新兴的进化计算技术，其基本操作包括变异、交叉和选择操作。它采用浮点数编码，在连续空间进行优化计算，是一种求解实数变量优化问题的有效方法。在解决复杂的全局优化问题方面，差分进化算法被实践证明是一种有效的全局最优解的搜索算法，其应用领域也越来越广。与其它进化算法相比，差分进化算法具有实现简单、收敛速度快、全局搜索能力强、鲁棒性等优点。因此本文采用差分进化算法来求解式(9)中的规划问题。

当用进化算法求解带约束的规划问题时，已有文献的通常做法都是将约束条件作为罚函数加入到目标函数中，使其变成无约束的规划，然后求解。但惩罚力度通常难以选择，太小太大都不好，并且如果惩罚力度选择不合适，或者如果出于计算量的考虑，迭代的次数不够，那么得到的解有可能不满足约束条件。

根据式(9)中规划问题的特点，本文不采用惩罚函数法，而是在迭代过程中解不满足约束条件时，采用贪婪的思想对其进行修正。具体修正方法如下：

1)当解不满足上下界约束条件时，若 p_i 小于其下界，就让其等于下界；若 p_i 大于其上界，就让其等于上界。此时若概率和约束条件得不到满足，就采用下面 2)中的方法进行修正。

2)当解不满足概率和约束条件时，分两种情况。

情况 1 $\sum_{i=1}^{N} p_i < 1$，修正算法如下：

```
psum=∑(i=1..N) pi;
while (psum<1) {
  从调度概率未达到其上界的节点中选择功耗因子最小的节点 Si;
  pi'=pi;
  pi=Min{pi,MAX,1-(psum-pi')};
  psum=psum-pi'+pi;
}
```

情况 2 $\sum_{i=1}^{N} p_i > 1$，修正算法如下：

```
psum=∑(i=1..N) pi;
while (psum>1) {
  从调度概率大于 0 的节点中选择功耗因子最大的节点 Si;
    pi'=pi;
  pi=Max{0,1-(psum-pi')};
  psum=psum-pi'+pi;
}
```

此外，我们将贪婪算法得到的解放入初始种群，以加快迭代的收敛速度。

4 仿真测试

4.1 仿真场景及参数

本文采用文献[12]中的功耗数据来进行仿真测试。假设 Web 集群的后端节点有如表 1 所列 3 种型号的 CPU，各种节点的功耗数据也在表 1 中给出，并假设各种节点的待机功耗均为 3W。仿真测试的场景有两个：一个为同构场景，采用 6 个编号 3 的节点；另一个为异构场景，每种编号的节点各 2 个。另外假设 $R=250$req/s，$T_{SLA}=0.1$s。

表 1 节点的 CPU 型号及功耗数据

编号	CPU 型号	频率(GHz)	空载功耗(W)	满载功耗(W)
1	AMD Athlon 64 3800+	1.0,1.8,2.0, 2.2,2.4	63.9,67.2,68.7, 69.9,71.6	71.6,85.5,90.7, 96.5,103.2
2	AMD Athlon 64 3500+	1.0,1.8, 2.0,2.2	66.6,73.8, 76.9,80.0	74.7,95.7, 103.1,110.6
3	AMD Athlon 64 3200+	1.0,1.8, 2.0	64.0,74.0, 81.0	73.0,108.0, 124.0

4.2 贪婪算法的求解结果

为了使调度概率精确到 0.01，我们首先采用遍历算法，枚举出调度概率所有可能的组合，得到使总功耗最小的“最优解”(由于精度不可能足够小，因此实际也只能得到次优解)和对应的最小功耗；然后采用贪婪算法进行求解，得到次优解和对应的最小功耗，并与遍历算法得到的最小功耗进行比较。

图 2 和图 3 分别给出了同构场景和异构场景中，上述两种算法在不同的请求速率下得到的最小功耗。图 2 和图 3 表明，即使是在规模很小的集群中，无论同构还是异构场景，贪婪算法得到的最小功耗大多数都比遍历算法得到的最小功耗高出不少，无法进行有效的节能，因此有必要研究更优秀的求解算法。

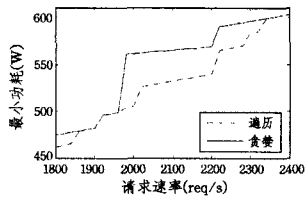

图 2 同构场景中两种算法得到的最小功耗

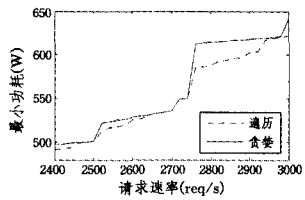

图 3 异构场景中两种算法得到的最小功耗

4.3 差分进化算法的求解结果

我们将贪婪算法得到的两个次优解(在"从剩下的节点中选择功耗因子最小的节点"步骤中,若有多个节点的功耗因子相同,分别采用优先选择编号大的和优先选择编号小的两种策略)放入初始种群,并选择如下的算法参数:种群大小为60,缩放因子为0.2,交叉概率为0.9,用差分进化算法进行迭代求解。在迭代过程中记录下每次迭代得到的最小功耗,观察最终能否得到不比遍历算法差的最小功耗,以及收敛速度如何。每个实验重复5次。

4.3.1 同构场景

当时,遍历算法得到的最优解为 $p=[0,0.2000,0.2000,0.2000,0.2000,0.2000]$,对应的最小功耗为531.7W;贪婪算法得到的两个次优解为 $p=[0.2333,0.2333,0.2333,0.2333,0.0668,0]$ 和 $p=[0,0.0668,0.2333,0.2333,0.2333,0.2333]$,对应的最小功耗为564.6W。差分进化算法的求解结果如图4所示。从图中可以看到,每次实验迭代不超过20次就能找到不比遍历算法差的最小功耗。

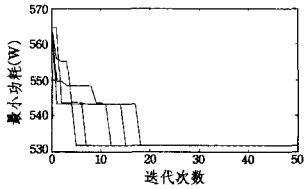

图 4 同构场景中差分进化算法的求解结果

4.3.2 异构场景

当时,遍历算法得到的最优解为 $p=[0.2107,0.2107,0.1500,0.1929,0.1500,0.0857]$,对应的最小功耗为588.0W;贪婪算法得到的两个次优解为 $p=[0.2107,0.2107,0.1929,0.1929,0.1750,0.0178]$ 和 $p=[0.2107,0.2107,0.1929,0.1929,0.0178,0.1750]$,对应的最小功耗为614.4W。差分进化算法的求解结果如图5所示。从图中可以看到,每次迭代不超过20次就能找到不比遍历算法差的最小功耗。

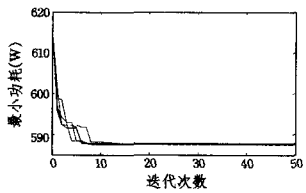

图 5 异构场景中差分进化算法的求解结果

结束语 Web集群已被广泛用来提高Web服务器的性能和可靠性,其节能问题亟待解决,为此本文提出了一种Web集群节能部署方案。该方案具有以下特点和优势:1)同时采用动态调整CPU频率和动态开关服务节点的措施来进行最有效的节能;2)运用M/G/1 PS排队模型对服务节点进行建模,将QoS保证下的Web集群节能部署问题转化为带约束的规划问题,其规划变量的个数比同类文献少,便于在线求解;3)采用差分进化算法求解该规划问题,并将贪婪思想引入到差分进化算法中,不仅保证得到的解满足约束条件,而且能加快算法的收敛速度;4)同构和异构场景的仿真测试验证了求解算法的可行性和有效性。

参 考 文 献

[1] 宋杰,李甜甜,闫振兴,等.一种云计算环境下的能效模型和度量方法[J].软件学报,2012,23(2):201-213

[2] Leite J C B, Kusic D, Mossé D, et al. Stochastic Approximation Control of Power and Tardiness in a Three-Tier Web-Hosting Cluster[C]//Proceedings of the 7th International Conference on Autonomic Computing. 2010:41-50

[3] Krioukov A, Mohan P, Alspaugh S, et al. NapSAC: Design and Implementation of a Power-Proportional Web Cluster[C]//Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking. 2010:15-22

[4] 刘斌,杨坚,赵宇.基于在线负载预测的动态集群节能配置策略[J].计算机工程,2010,36(24):96-98

[5] Guenter B, Jain N, Williams N. Managing Cost, Performance, and Reliability Tradeoffs for Energy-Aware Server Provisioning [C]//Proceedings of IEEE INFOCOM. 2011:1332-1340

[6] 刘峥.嵌入式Web集群服务器节能机制的研究与实现[J].计算机工程,2007,33(13):138-140,143

[7] Sasaki H, Oya T, Kondo M, et al. Power-Performance Modeling of Heterogeneous Cluster-Based Web Servers[C]//Proceedings of the 10th IEEE/ACM International Conference on Grid Computing. 2009:225-231

[8] Petrucci V, Loques O, Mossé D. A Dynamic Optimization Model for Power and Performance Management of Virtualized Clusters [C]//Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking. 2010:225-233

[9] Wang Pei-jian, Qi Yong, Liu Xue, et al. Power Management in Heterogeneous Multi-tier Web Clusters[C]//Proceedings of the 39th International Conference on Parallel Processing. 2010:385-394

[10] Chen Jian-jia, Huang Kai, Thiele L. Power Management Schemes for Heterogeneous Clusters under Quality of Service Requirements[C]//Proceedings of the 2011 ACM Symposium on Applied Computing. 2011:546-553

[11] Petrucci V, Carrera E V, Loques O, et al. Optimized Management of Power and Performance for Virtualized Heterogeneous Server Clusters[C]//Proceedings of 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. 2011:23-32

[12] Bertini L, Leite J C B, Mossé D. Power Optimization for Dynamic Configuration in Heterogeneous Web Server Clusters[J]. Journal of Systems and Software, 2010, 83(4):585-598

[13] Berg J L, Boxma O J. The M/G/1 Queue with Processor Sharing and its Relation to a Feedback Queue[J]. Queueing Systems, 1991, 9(4):365-401

[14] 杨振宇,唐珂.差分进化算法参数控制与适应策略综述[J].智能系统学报,2011,6(5):415-423