DOI:10.13718/j.cnki.xsxb.2017.07.008

云计算中时间感知应用的资源分配与调度算法①

刘小铭[1], 李宗辉[1], 王俊杰[2], 许旭江[3]

1. 揭阳职业技术学院 信息工程系,广东 揭阳 522000; 2. 百度在线网络技术有限公司,北京 100193;
3. 广东方昱网络有限公司,广东 揭阳 522000

摘要:现有数据中心中时间感知型云计算应用的资源分配算法能耗较高,严重影响了数据中心的服务上限以及云服务商的经济效益,对此提出一种低能耗的云计算资源分配与调度优化算法.算法分为两个阶段:第一阶段,释放并更新请求集的服务器与链接的剩余容量,同时更新能量辅助图中相应的权重;第二阶段,将所有新到达请求按所需时间段以降序排列,为各请求分配资源;第三阶段,检查资源可用性,并基于能量辅助图采用最短权值路径选择算法为资源请求分配虚拟机与流量.基于思科真实设备参数的仿真实验结果表明,本文云计算资源分配与路由算法的能量效率与资源分配性能均优于其他算法.

关 键 词:云计算;时间感知应用;多租户;资源分配;虚拟机;带宽资源

中图分类号:TP393 **文献标志码**:A **文章编号**:1000-5471(2017)07-0046-08

能量效率与计算性能是云计算数据中心的两个重要指标,它们对云服务提供商的经济收益具有直接的影响[1].目前,数据中心能耗的不断增长成为云服务提供商的主要难点,能耗的约束成为了云服务商服务更多租户的瓶颈,并且影响部分潜在租户的商业选择,因此,为了吸引更多的租户,在保证租户资源需求的同时,提高数据中心的能量效率成为一个关键点[2-3].目前,时间感知型应用成为了云计算租户的主流应用形式,租户一般需要向云服务商请求时间感知的虚拟机与带宽资源,而针对多租户场景,虚拟机与带宽资源的分配与调度是一个重要问题.已有的文献将其建模为线性规划问题,并对其进行优化,可获得最优解,但针对大规模的云计算数据中心,已有算法的计算复杂度较高,实用性较低[4-6].

文献[7]针对时间感知与网络感知的资源分配问题进行了研究与优化,但该研究假设所有的虚拟机资源相同,且虚拟机之间可准确地进行网络通信,而实际的应用场景中,虚拟机资源与带宽资源一般为异构形式,难以应用文献[7]的求解模型.此外,其他部分优化算法[8-10]则重点关注可靠性、网络带宽等方面,而并未侧重于数据中心的能量效率性能.

本文针对时间感知的云计算应用提出一种低计算复杂度的资源分配与调度优化算法,开发了一个简单且友好的云资源请求模型,租户可据此模型请求所需的服务器与带宽资源.具体方法为:以资源请求图描述租户的资源请求(包括服务器资源、带宽资源以及预算的时间段),根据动态到达的资源请求为租户分配所需资源.本文的优化目标为最大化资源请求的数量并最小化能耗,因此,首先基于多元素的能量模型设计了一个混合整数线性规划优化问题(MILP, Mixed-integer Linear Programming)[11],该问题结合了虚拟机的分配与路由问题,因此计算量较大,本文为此设计了低计算复杂度的算法.

① 收稿日期:2016-03-20
基金项目:2017 年广东省大数据分析与处理重点实验室开放基金资助项目(2017016);揭阳职业技术学院 2016 年一般科研项目(2016JYCKY05).
作者简介:刘小铭(1982-),女,重庆綦江人,讲师,主要从事计算机应用基础、虚拟化、云计算的研究.

1　系统模型

1.1　数据中心网络模型

本文采用广泛使用的三层数据中心分层拓扑结构[12]，三层分别为：核心层－汇聚层－边缘层，如图 1 所示. 边缘层中与服务器连接的交换机称为 ToR(机架顶端交换机)，ToR 分别通过两条 10 Gbps的上行链路连接到汇聚交换机，汇聚交换机再通过 10 Gbps 的上行链路与每个“Intermediate Switch”(中间交换机)相连.

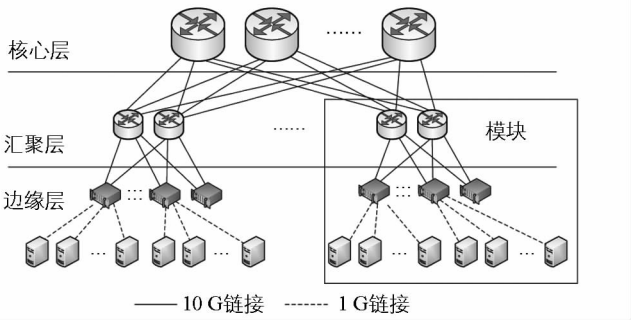


图 1　三层数据中心分层拓扑结构

本文将数据中心表示为带权有向图 $G(V, E)$，其中 V 与 E 分别是网络的节点与边集合. 图中每条有向边$\langle o, p\rangle \in E$ 表示对应的带宽容量 $B_{o,p}$. 节点 $v \in V$ 则表示一个服务器、交换机或外部的客户端设备. 将数据中心以外的客户端表示为 $v*$. 分别采用 S,W 表示数据中心中服务器与交换机的集合，将包含节点 v 的机架与模块分别设为 R_v 与 $\mathfrak{M}_v$. 将服务器的资源抽象为 CPU、内存与存储空间 3 者的度量来量化服务器资源. 服务器资源总体设为 $C_v(v \in S)$.

1.2　请求模型

本文采用文献[13]中时间感知请求模型与应用感知租户资源抽象模型，建立更为实际的请求模型(TTA). 将时间分为 T 个时隙，时隙集合表示为 T，$T=\{t \mid t=1, \cdots, t_{end}\}$. 定义 m 个异质虚拟机的集合为 M，假设虚拟机为 $m \in M$，外部服务器为 $v*$. 将第 k 个 TTA 表示为元组形式$(B_k, \tau_k^{start}, \tau_k^{end})$，其中图 $B_k(H_k, I_k)$ 表示租户应用的资源需求，τ_k^{start} 与 τ_k^{end} 则指明请求估算的时间段. 图 $B_k(H_k, I_k)$ 是一个应用感知型租户的资源抽象，其中 H_k 是应用层的集合，每个应用层基于虚拟机的类型与数量定义，I_k 是所有有向边的集合，每条边表示对应层之间的带宽需求. 每个应用层 $\eta \in H_k$ 与$\langle \Psi_k^\eta, N_k^\eta\rangle$ 关联，该虚拟机类型表示为 $\Psi_k^\eta \in \Psi$，对应类型的数量表示为 $N_k^\eta \in Z_{>0}$.

1.3　能量模型

受文献[14]启发，本文根据设备(服务器或交换机)的资源利用率(交换机的活动端口占比，服务器中虚拟机使用的单位资源数量)定义其能耗，最终各设备的总能耗基于其空闲能耗与设备使用每单位资源的能耗确定，因此，v 的总能耗可如下定义：

$$P_v(n_v) = P_v^0 + n_v P_v^*, \ \forall v \in V \tag{1}$$

式中 P_v^0 与 P_v^* 分别表示设备的空闲能耗与每单位资源的能耗，n_v 为 v 所需的资源量. 对于服务器($v \in S$)，空闲能耗 P_v^0 表示其空闲状态的能耗，$n_v P_v^*$ 则表示当前虚拟机所属的服务器能耗之和；对于交换机($v \in W$)，空闲能耗表示其无流量状态的能耗，而 P_v^* 表示每个活动端口的交换机能耗，n_v 表示交换机 v 的活动端口数量.

2　云计算资源分配与调度算法

2.1　问题模型

考虑一个网络与一个 TTA 请求集合 K(k 个请求)，本文将目标问题建模为 MILP 问题.

2.1.1　目标函数

目标问题是最大化资源请求接收率，同时最小化能耗，其目标函数如下描述：

$$\text{Minimize}: C = C^{\text{Rej}} + C^{\text{Power}} \tag{2}$$

式中：C^{Rej} 由(3)式决定，表示总拒绝成本，C^{Power} 由(4)式决定，表示接收请求前设备启动的总能耗. 假设 c 是每个拒绝请求的成本，则

$$C^{\text{Rej}} = c \times (K - \sum_{k \in K} Z_k) \tag{3}$$

式中$(K-\sum_{k\in K}Z_k)$表示拒绝请求的总数量. 为了最大化C^{Rej}中接收请求的数量,将c值设为恒大于单位时隙、单位请求的最大能耗,从而使得请求被接收的优先级高于被拒绝的优先级. 考虑V_t是时隙t中所有启动服务器与交换机的集合,C^{Power}值计算方法如下:

$$C^{\text{Power}}=\sum_{t\in T,\ v\in V_t}P_v(n_v)=\sum_{t\in T,\ v\in V_t}P_v^0+\sum_{t\in T,\ v\in V_t}n_vP_v^* \tag{4}$$

式中v的能耗由(1)式决定,将(1)式代入(4)式:

$$C^{\text{Power}}=\sum_{t\in T,\ v\in V}P_v^0X_v^t+\sum_{\substack{p;\ (v,\ p)\in E,\\ t\in T,\ v\in W}}P_v^*Y_{v,\ p}^t+\sum_{\substack{t\in T,\ v\in S,\ k\in K,\\ m\in M,\ \eta\in H_k}}P_v^*f_{m,\ v}^{k,\ \eta}d_k^t\psi_k^{\eta} \tag{5}$$

式中第一项决定总空闲能耗,后两项分别是交换机与服务器的总能耗.

2.1.2 约 束

本文的资源分配与调度问题是一种多商品流问题,它必须满足容量约束,流量守恒条件与期望满足约束[15-16],将其分别表示为(6)-(8)式. 对于一个请求$k\in K$,其资源需求图为$B_k(H_k,I_k)$,$r_k^l\in Z_{>0}$指定了每个虚拟机(每层$\eta^{src}\in H_k$)的带宽需求,假设$I_{o,\ p}^{k,\ l}\in Z_{\geqslant 0}$,$l=\langle\eta^{src},\eta^{dst}\rangle$为一个二值变量,表示当前层$\eta^{src}\in H_k$内发送流量的虚拟机数量. 容量约束(6)确保经过每个链接的流量总量不超过其容量;对于交换机$v\in W$,流量守恒约束(7)确保在k(资源请求)的图$B_k(H_k,I_k)$中的总输入、输出流量相等;期望满足约束(8)确保当前层$l\in I_k$中流入、流出服务器的总流量相等:

$$\sum_{k\in K,\ l\in I_k}r_k^ld_k^tI_{o,\ p}^{k,\ l}\leqslant B_{o,\ p},\ \forall t\in T,\ \forall\langle o,\ p\rangle\in E \tag{6}$$

$$\sum_{p;\ (v,\ p)\in E}I_{v,\ p}^{k,\ l}=\sum_{p;\ (v,\ p)\in E}I_{v,\ p}^{k,\ l},\ \forall v\in W,\ \forall k\in K,\ \forall l\in I_k \tag{7}$$

$$\sum_{\substack{o;\ (v,\ o)\in E,\\ v\in[S\cup v*]}}I_{v,\ o}^{k,\ l}=\sum_{\substack{o;\ (v,\ o)\in E,\\ v\in[S\cup v*]}}I_{v,\ o}^{k,\ l},\ \forall k\in K,\ \forall l\in I_k \tag{8}$$

同一应用不同层的虚拟机可能位于同一个服务器[17],将该情况定义为$I_{v,\ v}^{k,\ l}$,其中$\langle v,\ v\rangle$表示相同源、目标服务器v之间的链接,因此,对于一个应用层-对$l=\langle\eta^{src},\eta^{dst}\rangle$,$I_{v,\ v}^{k,\ l}$确定了应用层$\eta^{src}$中虚拟机的数量(属于$v$). 本文假设路由为不可分割型,如果同一服务器$v$中的源虚拟机没有向目标虚拟机发送流量,则流量会到达服务器的输出端口,因此需设置如下的约束条件:

$$\sum_{m\in M}f_{m,\ v}^{k,\ \eta^{src}}=I_{v,\ v}^{k,\ l}+\sum_{p;\ \langle v,\ p\rangle\in E}I_{v,\ p}^{k,\ l},\ \forall k\in k,\ \forall v\in S,\ \forall l\in I_k \tag{9}$$

式中$l=\langle\eta^{src},\eta^{dst}\rangle$. 对于图$B_k(H_k,I_k)$的请求$k\in K$,二值变量$f_{m,\ v}^{k,\ \eta}$表示虚拟机$m\in M$是否位于服务器$v\in S$中,如果应用层对$l$的目标虚拟机均不位于$v\in S$中,则不会有$l$之外的流量发送到$v$中,为此需设置如下的约束条件:

$$\left.\begin{aligned}\sum_{m\in M}f_{m,\ v}^{k,\ \eta^{dst}}&\geqslant\frac{\sum_{p;\ (p,\ v)\in E}I_{p,\ v}^{k,\ l}}{(N_k^{\eta^{src}}+N_k^{\eta^{dst}})},\\ \sum_{m\in M}f_{m,\ v}^{k,\ \eta^{dst}}&\geqslant\frac{I_{v,\ v}^{k,\ l}}{(N_k^{\eta^{src}}+N_k^{\eta^{dst}})},\end{aligned}\right\}\begin{aligned}&\forall k\in k,\ \forall v\in S,\\ &\forall l=\langle\eta^{src},\eta^{dst}\rangle\in I_k\end{aligned} \tag{10}$$

假设二值变量Z_k表示请求$k\in K$是否被满足,对于每个满足($Z_k=1$)的请求$k\in K$,仅属于指定应用层$\eta*$的虚拟机位于节点$v*$中(外部服务器),表示为:

$$\sum_{m\in M}f_{m,\ v*}^{k,\ \eta}=\begin{cases}N_k^{\eta*}Z_k, & \text{if }\eta=\eta*\\ 0, & \text{其他情况}\end{cases}\qquad\forall k\in K,\ \forall\eta\in H_k \tag{11}$$

式中$N_k^{\eta}\in Z_{>0}$表示每个应用层$\eta\in H_k$中实体的所需数量. 然后定义了(12)-(14)式的虚拟机约束:约束(12)确保各虚拟机置于同一个服务器中;约束(13)表示仅当一个请求的所有虚拟机均被成功分配,则认为请求$k\in K$被满足,$d_k^t\in\{0,1\}$表示时隙$t\in T$中请求$k\in K$是否有流量,$\psi_k^{\eta}\in\Psi$表示每个虚拟机的资源需求;约束(14)限制服务器中虚拟机占用的单位资源总量低于服务器的容量.

$$\sum_{\substack{\eta\in H_k,\ k\in K,\\ v\in\{S\cup v^*\}}} f_{m,v}^{k,\eta}\leqslant 1,\ \forall m\in M \tag{12}$$

$$\sum_{\substack{v\in\{S\cup v^*\},\\ m\in M}} f_{m,v}^{k,\eta}=N_k^{\eta}Z_k,\ \forall k\in K,\ \forall \eta\in H_k \tag{13}$$

$$\sum_{\substack{k\in K,\ \eta\in H_k,\\ m\in M}} d_k^t f_{m,v}^{k,\eta}\psi_k^{\eta}\leqslant C_v,\ \forall v\in S,\ \forall t\in T \tag{14}$$

最终，为了计算所有设备启动的总能耗，设置了两个决策变量 $Y_{o,p}^t$ 与 X_v^t，分别表示链接 $\langle o,p\rangle\in E$ 与设备 $v\in V$ 的状态，因此：

$$\left.\begin{aligned} X_o^t &\geqslant d_k^t\frac{I_{o,p}^{k,l}}{(N_k^{\eta^{src}}+N_k^{\eta^{dst}})}\\ X_p^t &\geqslant d_k^t\frac{I_{o,p}^{k,l}}{(N_k^{\eta^{src}}+N_k^{\eta^{dst}})}\\ X_{o,p}^t &\geqslant d_k^t\frac{I_{o,p}^{k,l}}{(N_k^{\eta^{src}}+N_k^{\eta^{dst}})} \end{aligned}\right\}\begin{aligned}&\forall t\in T,\ \forall k\in K\\ &\forall\langle o,p\rangle\in E\\ &\forall l=\langle\eta^{src},\eta^{dst}\rangle\in I_k\end{aligned} \tag{15}$$

(15) 式考虑了所有的链接，如果时隙 t 中某个请求占用了链接 $\langle o,p\rangle\in E$，则认为链接 $\langle o,p\rangle$ 及其端口为活动状态.

3　本文算法

显然，上述资源分配与调度问题计算复杂度较高，本文为此设计了低计算复杂度的算法，本算法的思想是直接考虑资源辅助图的能量与占用状态，并预算候选的路由.

3.1　PER-TTA 算法

本算法分为两个阶段：释放与预约阶段. 在每个时隙的终点，PER-TTAMPC 轮询所有新到达请求，并为每个请求分配所需的资源. 对于给定的输入参数：K^A，K_t，C，B 以及能量辅助图 G^{MPC}，算法总体步骤如算法 1 所示：

算法 1

输入：K^A，K_t，C，B，G^{mp}

1. WHILE（$\exists k$：($k\in K^A$) $\wedge$ ($\tau_k^{end}=t$)）{　　// 第一阶段
2：　$K^A\leftarrow K^A\backslash\{k\}$；
3：　释放 C 与 B 中所有请求 k 的资源；
4.　更新 G^{mp} 中受影响链接的权重；
5. }
6. WHILE $K^A\neq\{\}$ {　　// 第二阶段
7.　$k\leftarrow K^t$ 中时间段最长的请求；
8.　$K^t\leftarrow K^t\backslash\{k\}$；
9.　$[C,B,G^{mp},\mathrm{status}]\leftarrow$ B-statisfaction(k,C,B,G^{mp})；
10.　IF (status == FAIL) 拒绝 k；ELSE $K^A\leftarrow K^A\cup\{k\}$；
11. }

在第一阶段：算法释放 K^A 中请求占用的资源，然后更新活动请求集 K^A 中服务器与链接的剩余容量，更新能量辅助图 G^{MPC} 中相应的权重. 在第二阶段：将所有新到达请求按所需时间段以降序排列，并使用函数 B-satisfaction 为各请求分配资源（算法 2）. 给定一个请求，首先检查资源可用性，然后函数 RoutePlaceVM（算法 3）对虚拟机进行分配与调度. 函数 RoutePlaceVM 基于能量辅助图采用最短权值路径选择算法为资源请求分配虚拟机与流量.

3.1.1　能量辅助图

1.1 小节设计了辅助带权图 $G^{MPC}(V,E)$ 描述数据中心网络，辅助带权图中权值与每个链接关联（基于当前链接的类型与能量状态及其端点的设备），服务器与交换机之间的链接权重表示交换机、服务器以及链接的启动成本. 如果图中元素（服务器、交换机、链接）是启动状态，则该元素启动成本为 0；否则，启动

成本为该元素的空闲能耗.

算法 2

B-statisfaction(k, C, B, G^{mp})
输入：k, C, B, G^{mp}
输出：[C', B', G'^{mp}, SUCCESS]
1. [C', B', G'^{mp}] ← [C, B, G^{mp}];
2. WHILE ($I_k \neq \{\}$) {
3. $\eta^{\max} \leftarrow H_k$ 中最大输出度的应用层;
4. WHILE $\exists l = \langle \eta^{src}, \eta^{dst} \rangle : (l \in I_k) \wedge (\eta^{src} = \eta^{\max})$ {
5. FOR $m^{src} \in \eta^{src}$ {
6. [C', B', G'^{mp}, *status*] ← RoutePlaceVM(C', B', G'^{mp}, m^{src}, η^{dst});
7. IF (status == FAIL) {
8. RETURN [C, B, G^{mp}, FAIL];
9. }
10. }
11. $I_k \leftarrow I_k \backslash \{l\}$;
12. }
13. }

3.1.2 RoutePlaceVM 函数

对于一个给定的源-虚拟机及其目标应用层 η^{dst}，函数 RoutePlaceVM 选择一个服务器放置虚拟机，并选择一条路径将流量路由至目标层 η^{dst} 的虚拟机. 为了实现该目标，RoutePlaceVM 在当前辅助图选择一个服务器对及其关联的路径.

算法 3

RoutePlaceVM(C', B', G'^{mp}, m^{src}, η^{dst})
输入：C', B', G'^{mp}, m^{src}, η^{dst}
1. shortest_path = []; min_cost = ∞;
2. IF(m^{src} 已成功分配) THEN $S^{src} \leftarrow$ {m^{src} 的虚拟机主机};
3. ELSE $S^{src} \leftarrow$ {具有足够剩余容量的服务器};
4$S^{dst} \leftarrow$ {v^{dst} | v^{dst} 是虚拟机 $m \in \eta^{dst}$ 的主机} $\cup$ {v^{dst} | v^{dst} 是具有足够剩余容量的服务器}
5. FOR({$\langle v^{src}, v^{dst} \rangle$}: ($v^{src} \in S^{src}$) $\wedge$ ($v^{dst} \in S^{dst}$)) {
6. IF (G'^{mp} 中 v^{src} 到 v^{dst} 的路径存在) {
7. [path, cost] = Weighted_shortest_path(v^{src}, v^{dst}, G'^{mp})
8. IF cost < min_cost THEN
9. shortest_path ← path; min_cost ← cost; source ← v^{src}; destination ← v^{dst};
10. ELSE IF (cost == min_cost)
11. 选择可用资源最少的对;
12. ENDIF
13. }
14. }
15. IF (*shortest_path*) {
16. 将 m^{src} 置于 *source* 中;
17. IF (!$\exists VM\ m \in \eta^{dst}$) THEN
18. 从目标中选择一个虚拟机 $m \in \eta^{dst}$
19. ENDIF
20. 更新 source 与 destination 服务器的剩余容量与 shortest_path 的所有链接;
21. 更新 G'^{mp} 中受影响的权重;
22. RETURN [C', B', G'^{mp}, SUCCESS];
23. ELSE
24. RETURN [C', B', G'^{mp}, FAIL];
25. }

3.2　计算复杂度

本算法每个时隙的计算时间由新到达请求的数量决定. 本算法的函数 RoutePlaceVMMPC 使用最短带权路径选择算法分配虚拟机与带宽资源，如果使用堆数据结构，则搜索一对多路由路径的计算复杂度是 $O((|E|+|V|)\cdot\log|V|)$，其中 E 与 V 分别是能量辅助图 $G(V, E)$ 中链接与节点的集合. 一对一虚拟机通信模型：一个请求的虚拟机分配与路由选择所需的计算复杂度为 $O((|M_k|\cdot|S|\cdot(|E|+|V|)\cdot\log|V|)$；多对多虚拟机的复杂度为 $O(|M_k|\cdot(|S|+|H_k|-1)\cdot(|E|+|V|)\cdot\log|V|)$. 一般 $|H_k|-1$ 的值远小于 $|S|$，因此本算法每个时隙的计算复杂度为：

$$O(|K_k|\cdot|M_k|\cdot|S|\cdot(|E|+|V|)\cdot\log|V|)$$

4　仿真实验与性能分析

4.1　仿真参数设置

考虑图 1 所示的三层数据中心拓扑结构，本文参考思科公司的产品信息，设置仿真场景：8 个核心交换机，16 个模块，每个模块有 2 个汇聚交换机，12 个 ToR 交换机，每个 ToR 包含两个交换机端口(与汇聚交换机连接)、48 个服务器端口(与 48 个服务器连接)以及两个汇聚开关. ToR 交换机与服务器的总数量分别是 192 与 9216. 参考思科公司的数据中心白皮书(Cisco Data Center Infrastructure SRND 2.1)，分别设置 ToR-汇聚层与汇聚层－核心层的过度定制比(oversubscription ratio)为 2.4∶1 与 1.5∶1. 服务器与 ToR 交换机之间链接的带宽设为 1 Gbps，交换机之间的带宽为 10 Gbps. 将服务器的硬件资源抽象为 CPU、内存以及存储容量的度量，本文为每个服务器分配 10 个单位的资源.

参照文献[18]设置仿真实验的能量参数，ToR、汇聚交换机与核心交换机的空闲能耗分别为 76.4 W，133.5 W 与 555 W，最大能耗分别为 102 W，175 W，656 W，每个活动端口的能耗分别为 0.87 W，0.9 W，0.9 W.

流量参数则参考文献[19]，到达的资源请求为随机分布，请求的时间段则为指数分布，每个请求随机生成资源需求图 B_k，假设 B_k 包含 3 个等级(分别为 2，4，8)，本文参考 Amazon 的真实资源分级方法[20]，将虚拟机分为 4 个等级 $\Psi=\{$"S", "M", "L", "XL"$\}$，各级别虚拟机分别消耗 1，2，4，8 个单位的服务器资源. 为了支持流量分簇，本文考虑低速、高速两种流量的请求，高速流量占总请求的 30%，r_k^l 值服从 10～100 kbps 之间的正态分布，低速流量速度在 1 500～2 500 kbps 之间.

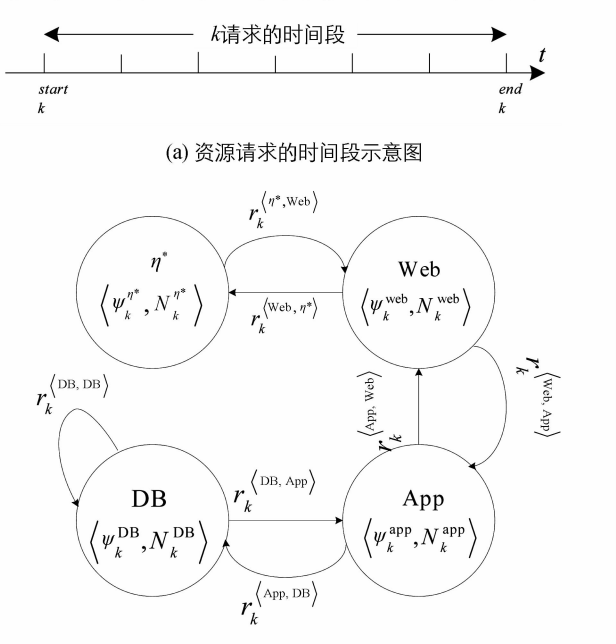


图 2　资源请求的时间段与三阶网络应用的实例

4.2　仿真场景

请求阶数：图 2 所示为一个三阶网络应用的实例，包括 Web，App，DB 层，本仿真实验将请求的阶数设为 2，4，8. 假设共有 1 000 个时隙，平均到达速度为 100 个请求/时隙，请求包含高、低速流量(30∶70). 仿真实验比较算法的两个重要指标：请求接受率和每个时隙、每个接受请求的平均能耗.

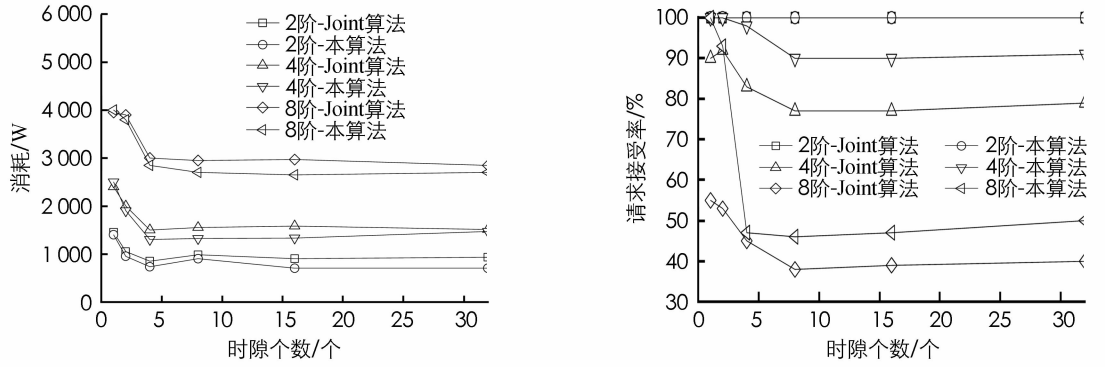
4.3　仿真器

本文基于 Python 语言与编译器实现资源分配与路由算法，并生成随机的资源请求模型，具体参数设置参考 4.1 小节. 本文对每组实验重复若干次，使得实验结果达到 95%的置信区间.

4.4 结 果

将本算法与文献[21](简称为Joint算法)进行对比，Joint算法是一种联合数据中心虚拟机分配与路由算法，该算法建立了合理的模型，但是该求解方法并未考虑数据中心的能耗优化.

图3所示是本算法与Joint算法的性能比较，从图3(a)可看出，对于不同阶数的云计算网络以及请求的时间长度，本算法的平均能耗均低于Joint算法. 从图3(b)可看出，对于不同阶数的云计算网络以及请求的时间长度，本算法的请求接受率(资源分配效果)均优于Joint算法.



(a) 每个时隙、每个接受请求的能耗 (b) 请求的接受率

图3 云计算资源分配算法的能效与资源分配性能结果

5 结束语

本文针对时间感知的云计算应用提出一种低计算复杂度的资源分配与调度优化算法，开发了一个简单且友好的云资源请求模型，租户可据此模型请求所需的服务器与带宽资源. 基于能量辅助图采用最短权值路径选择算法为资源请求分配虚拟机与流量. 基于思科真实设备参数的仿真实验结果表明，本文云计算资源分配与路由算法的能量效率与资源分配性能均优于其他算法.

参考文献：

[1] 邓 维，刘方明，金 海，等. 云计算数据中心的新能源应用：研究现状与趋势[J]. 计算机学报，2013，36(3)：582－598.

[2] 张 伟，宋 莹，阮 利，等. 面向Internet数据中心的资源管理[J]. 软件学报，2012，23(2)：179－199.

[3] 魏祥麟，陈 鸣，范建华，等. 数据中心网络的体系结构[J]. 软件学报，2013，24(2)：295－316.

[4] 张小庆，贺忠堂，李春林，等. 云计算系统中数据中心的节能算法研究[J]. 计算机应用研究，2013，30(4)：961－964.

[5] 李阳阳，王洪波，张 鹏，等. 基于多属性信息的数据中心间数据传输调度方法[J]. 通信学报，2012(S1)：121－131.

[6] 郭力争，张翼飞，赵曙光. 数据中心环境下能耗性能感知的优化方法[J]. 北京邮电大学学报，2015(Z1)：72－76.

[7] DALVANDI A, GURUSAMY M, CHUA K C. Time-Aware Vmflow Placement, Routing, and migration for Power Efficiency in Data Centers[J]. IEEE Transactions on Network & Service Management, 2015, 12(3): 349－362.

[8] 左 成，虞红芳. 可靠性感知下的虚拟数据中心映射算法[J]. 计算机应用，2015，35(2)：299－304.

[9] 荣 超，唐亚哲，胡成臣，等. 基于带宽感知的多租户云数据中心虚拟网络分配算法[J]. 小型微型计算机系统，2015(1)：7－12.

[10] 王光波，马自堂，孙 磊，等. 基于架构负载感知的虚拟机聚簇部署算法[J]. 计算机应用，2013，33(5)：1271－1275.

[11] SIMOGLOU C K, BISKAS P N, BAKIRTZIS A G. Optimal Self-Scheduling of a Thermal Producer in Short-Term Electricity Markets by MILP[J]. IEEE Transactions on Power Systems, 2010, 25(4): 1965－1977.

[12] BARI M F, BOUTABA R, ESTEVES R, et al. Data Center Network Virtualization: a Survey[J]. IEEE Communications Surveys & Tutorials, 2013, 15(2): 909－928.

[13] NADEMBEGA A, HAFID A, TALEB T. Mobility-Prediction-Aware Bandwidth Reservation Scheme for Mobile Networks[J]. IEEE Transactions on Vehicular Technology, 2015, 64(6): 2561－2576.

[14] PADALA P, SHIN K G, ZHU X, et al. Adaptive Control of Virtualized Resources in Utility Computing Environments[J].

Acm Sigops Operating Systems Review，2007，41(3)：289－302.

[15] 万　华，叶耀华. 一种带路径约束的多商品流网络设计问题及其禁忌算法 [J]. 复旦学报(自然科学版)，2005，44(2)：220－223.

[16] MANN V，KUMAR A，DUTTA P，et al. VMFlow：Leveraging VM Mobility to Reduce Network Power Costs in data Centers [M]// International Conference on Networking. Berlin：Springer，2011：198－211.

[17] JIN H，CHEOCHERNGNGARN T，LEVY D，et al. Joint Host-Network Optimization for Energy-Efficient Data Center Networking [J]. Alzheimers & Dementia，2013，9(4)：623－634.

[18] HELLER B，SEETHARAMAN S，MAHADEVAN P，et al. Elastic Tree：Saving Energy in Data Center Networks [J]. Nsdi，2015，51(11)：249－264.

[19] DIVAKARAN D M，LE T N，GURUSAMY M. An Online Integrated Resource Allocator for Guaranteed Performance in Data Centers [J]. IEEE Transactions on Parallel & Distributed Systems，2014，25(6)：1382－1392.

[20] SIDDHARTH JOSHI，STEPHEN BOYD. An Efficient Method for Large-Scale Slack Allocation [J]. IEEE Transactions on Circuits & Systems I Regular Papers，2009，41(12)：1163－1176.

[21] JIANG J W，LAN T，HA S，et al. Joint VM Placement and Routing for Data Center Traffic Engineering [C]//2012 Infocom Proceedings IEEE. New York：IEEE Computer Society Press，2012.

On Resource Allocation and Scheduling Algorithm for Time Aware Applications in Cloud Computing

LIU Xiao-ming[1]，　LI Zong-hui[1]，　WANG Jun-jie[2]，　XU Xu-jiang[3]

1. Department of Information Engineering，Jieyang Vocational & Technical College，Jieyang Guangdong 522000，China；

2. Baidu Online Network Technology (Beijing) Co.，Ltd，Beijing 100193，China；

3. Guangdong Fang Yu Network Co.，Ltd，Jieyang Guangdong 522000，China

Abstract：The existing the resource allocation algorithms of time aware cloud computing applications in the data centre perform high energy consumption，and it significantly affect the ability to support more tenants and the economic benefits of cloud computing service providers，concerned at that problem a low power consumption resource allocation and scheduling algorithm is proposed. The algorithm consists of two phases：in the first phase，the residual capacities of services and links in the request are updated and released，and the corresponding weights in the energy graph are updated；in the second phase，the new arriving requests are ordered in descending order，and the resources are allocated to each request. And lastly，the availability of request is checked and the shortest weighted paths election algorithm is used to allocate the virtual machines and bandwidth for resource requests. Cisco device references based simulation experimental results show that the proposed cloud computing resource allocation and routing algorithm outperforms the other algorithms in energy efficiency and resource allocation performance.

Key words：cloud computing；time aware application；multi-tenancy technology；resource allocation；virtual machine；bandwidth resource

责任编辑　张　枸

崔玉洁