

Overlay 服务网络中的多因子服务放置问题^{*}

潘 耘¹ 余镇危² 冯烟利² 高 斌²

(中国传媒大学计算机与软件学院 北京 100024)¹ (中国矿业大学北京校区计算机系 北京 100083)²

摘 要 Overlay 服务网络是一种通用的服务框架,它利用覆盖网络技术来向用户提供各种各样的服务。本文在讨论 Overlay 服务网络的基础上,提出了多因子服务放置的问题,对该问题进行了形式化建模,并提出了相应的算法,对算法进行了相应的仿真。

关键词 Overlay 服务网络,服务因子,分层技术

On the Multiple-Factors Service Placement Problems on the Overlay Service Networks

PAN Yun¹ YU Zhen-Wei² FENG Yan-Li² GAO Bin²

(Computer Science & Software Institute, Communication University of China, Beijing 100024)¹

(Dept. Computer, China University of Mining and Technology-Beijing, Beijing 100083)¹

Abstract The multiple factors service placement problem(MFPP) has been investigated carefully. Corresponding programming models and a heuristic algorithm for the MFPP problem have been proposed. The complexity analysis of the algorithm and the numerical experiments are also presented.

Keywords Overlay service network, Factor service, Layer technology

1 引言

Overlay 服务网络是一种通用的服务框架^[1],无需对基础网络层的协议进行修改,即可向用户提供各种各样的服务。一个 Overlay 服务网络通常由一组覆盖节点(Overlay Nodes)构成,而这些覆盖节点又是由第三方,即覆盖服务提供商(Overlay Service Provider, OSP)按照一定的放置策略(Placement Strategy)从基础网络层的物理节点中选择、建设、部署而成^[1]。目前,Overlay 服务网络研究的核心是其拓扑结构问题,即假定覆盖节点都已经选定,探讨如何在这些节点之间建立覆盖链路(Overlay Links),以更加有效地支持提供各种覆盖应用(Overlay Applications)。不同的拓扑结构往往被称为不同的覆盖服务网络,主要有如下几种:应用层组播网络^[2~7],弹性网络^[8]等等。这些种类各异的拓扑结构往往是针对不同的特定服务而优化选择的,对一种服务适合的拓扑结构的另外的服务就不一定是高效的。

文[2]提出了通用 OSN 的概念,这种通用的 OSN 为各种各样不同的应用所共享;为了实现这一目标,文[2]引入了 Overlay 代理(Overlay Brokers, OBs)的概念,这些 OBs 按照“特定策略”放置在 Internet 之上。如何在这些代理之上放置服务能更有效地提供通用的服务,这是在 Overlay 网络上进行路由和构建其它覆盖应用的基础。现有的研究文献大多是针对特定的服务(例如只考虑组播)研究适合的拓扑结构,在实际应用中往往需要同时考虑多个服务(例如组播、压缩、缓存、重新编码等)的相互配合,这时原有的针对特定服务而选择的 OSN 拓扑结构就不一定是最有效的。尽管文[2]也提出了通用 OSN 的概念,但是并没有对服务的放置问题做进一步的研究。

2003 年,本文作者提出并展开了对服务放置问题的研

究^[9]。本文可以看作是这方面研究的阶段性成果。限于篇幅,本文侧重对多因子服务放置问题(MFPP)进行建模,并设计了求解该模型的启发式算法。

2 服务分类和服务因子

OSN 支持的服务可分为两类:一类是从源端来的数据流接受该服务后流向唯一的节点(如图 1),只是数据流的流量,或者负荷中的数据内容发生了改变。如压缩服务改变了数据流的流量,加密解密和其他编码服务改变了负荷中的数据内容等,所以它们都可看作为一对一的服务。另一类是接受该服务后流向多个目的节点(图 2),称为一对多服务。组播通信服务就可归为一对多的服务,组播节点对数据流进行复制,发向同组的多个目的节点。

我们可以把一对一的服务与一对多的服务进行组合,来完成更复杂的服务。例如加密服务与组播服务进行组合,这样每个组播组的目的节点都可获得加密服务。

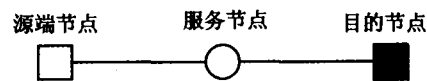


图 1 一对一的服务

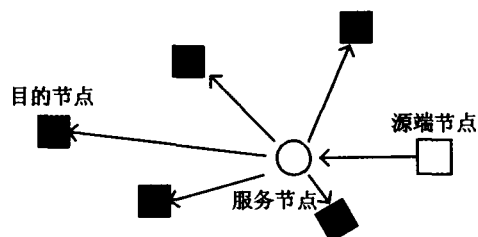


图 2 一对多的服务

^{*}教育部博士点基金项目(20030290003)。潘 耘 博士,研究方向:计算机网络。余镇危 博士生导师,主要研究方向:计算机网络。冯烟利 博士生,副教授。高 斌 硕士。

在研究服务放置问题时,还需要考虑服务因子的影响。因为有些端系统之间的通信需要接受中间节点提供的一次服务,有些通信需要组合多个相关的服务才能完成。所以 OSN 上的服务放置问题分为:单因子和多因子的服务放置问题。单因子的服务放置问题是指如何为多个需要同一服务的会话对选取合适的主动节点来放置适量数目的该类服务。多因子的服务放置问题是指如何放置多个相关联的服务因子才能满足多个会话组按不同次序来组合服务的需要。

3 多因子服务放置问题

给定网络拓扑结构 $G=(V,E;O)$, 其中 V 是基础网络层中节点的集合; E 是物理链路(也称为边)集合, O 为覆盖节点的集合。并设

$e:V \times V \rightarrow (0, \infty)$, $e(i,j)=e(j,i)$ 为边权函数, 可解释为延迟、代价等等;

$F=\{F_1, \dots, F_m\}$ 为要放置的多个服务因子的集合;

$c:O \times F \rightarrow [0, \infty]$, $c(i,j)$ 表示在节点 $i \in O$ 上部署第 j 类服务因子 F_j 的代价($c(i,j)=\infty$ 表示不可以在节点 i 上放置第 j 类服务因子 F_j);

$S=\{\langle s_1, d_1 \rangle, \langle s_2, d_2 \rangle, \dots, \langle s_n, d_n \rangle\}$ 为所关心的会话组。对每一个会话对 $\langle s_i, d_i \rangle$, 设偏序集 $P(s_i, d_i) \subseteq F \times F$ 指明了从源端 s_i 到目的端 d_i 的数据流所需要经历的服务因子以及经历各服务因子的次序关系。

定义 1 多因子服务标志函数 用来标志覆盖节点上放置的服务因子的类别, 定义为

$$z:O \times F \rightarrow \{0, 1\}, z(i,j) = \begin{cases} 1, & \text{节点 } i \text{ 上放置服务因子 } F_j \\ 0, & \text{节点 } i \text{ 不放置服务因子 } F_j \end{cases} \quad (1)$$

定义 2 先行服务因子链 对于 s 到 d 的路径 $path(s, d)$, 提供的服务因子组成的从 s 到 d 的链, 称为会话 $\langle s, d \rangle$ 的先行服务因子链, 用 $a(s, d)$ 表示。如图 3 所示, 若 $path(s, d)$ 上有覆盖节点依次提供第 k 类、第 j 类服务, 则 $a(s, d) = \{k, j\}$ 。



图 3 先行服务节点链

于是, MFPP 问题可定义为:

$$\begin{aligned} \min & \sum_{i=1}^n l_G(s_i, d_i) + \sum_{i \in O} \sum_{j \in F} z(i,j) c(i,j) \\ \text{s.t. } & a(s_i, d_i) \subseteq \bar{P}(s_i, d_i), i=1, 2, \dots, n \quad (a) \\ & z \in \{0, 1\}^{O \times F} \quad (b) \end{aligned} \quad (2)$$

其中: $l_G(s_i, d_i)$ 表示图 G 中从 s_i 到 d_i 的最短路径长度; $\bar{P}(s_i, d_i)$ 为偏序关系 $P(s_i, d_i)$ 按照任一拓扑序定义的一个全序关系(即链), 约束条件(a)表示 $a(s_i, d_i)$ 是 $\bar{P}(s_i, d_i)$ 的子链, 即会话的先行服务因子链满足给定的服务次序约束; 约束条件(b)指明了决策变量的取值空间, 从形式上看, 此模型属于泛函优化模型。但是, 当服务放置策略的决策函数 z 给定时, 先行服务因子链函数 a 也随之确定, 所以此模型等价于下列 0-1 规划模型:

$$\begin{aligned} \min & \sum_{i=1}^n l_G(s_i, d_i) + \sum_{i \in O} \sum_{j \in F} z_{ij} c(i,j) \\ \text{s.t. } & a(s_i, d_i) \subseteq \bar{P}(s_i, d_i), i=1, 2, \dots, n \quad (a) \\ & z_{ij} \in \{0, 1\}, \forall i \in O, \forall j \in F \quad (b) \end{aligned} \quad (3)$$

一般的 0-1 规划问题本身是 NP-hard 的, 常用的算法是“分支-定界法”, 随着问题规模的增大, 该算法具有指数复杂度。从模型(2)或(3)可以看出, MFPP 问题是特殊的 0-1 规划问题, 其特殊性在于模型中包含有复杂的离散数据结构约束, 例如子链包含这样的约束条件。所以, MFPP 问题的复杂性至少不低于一般的 0-1 规划模型, 即 MFPP 问题也是 NP-hard 的。由于目前还不存在确定性的多项式时间算法可以求解这类问题的最优解, 所以我们下面给出一个求解该问题次优解的启发式算法。

4 求解 MFPP 问题的启发式算法

我们基于 Choi 等人提出的分层思想^[17], 设计了一个求解 MFPP 问题的启发式算法。该算法的主要思想是: 分别按照每对会话要求的服务因子的次序计算源端到目的端的最短距离, 然后统计每个主动节点在各个会话中承担各种服务因子的次数, 选择次数最大的主动节点来承担所对应的主动服务, 若次数都相同, 随机选择一个。同时, 该算法通过迭代选择较优的服务放置策略; 如果某次迭代后解的质量没有显著改进, 则结束迭代。

我们先给出算法的大致框图(图 4), 再给出详细的算法描述, 然后通过一个例子来说明算法的流程。

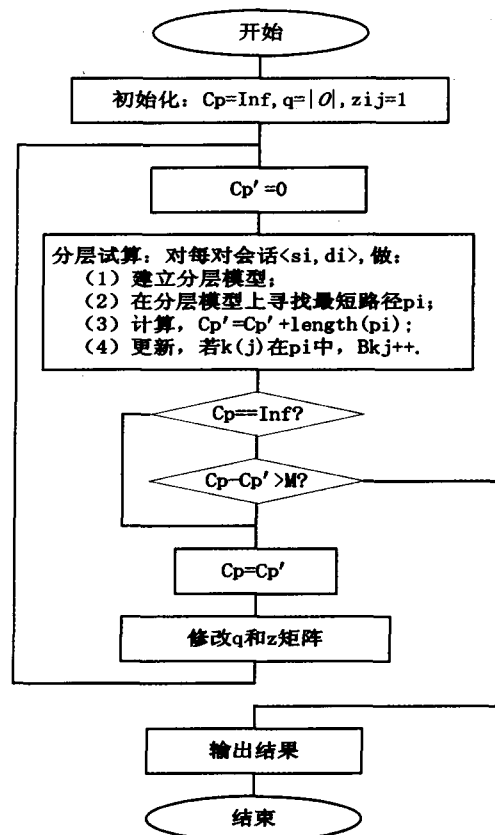


图 4 算法框图

Algorithm MFPP(G, O, F, S, M)

Input: G is the network graph; O is the set of active overlay service nodes; F denotes the set of service factors; S is the set of session group $\{\langle s_1, d_1 \rangle, \langle s_2, d_2 \rangle, \dots, \langle s_n, d_n \rangle\}$; M is the iteration control parameter.
Output: Matrix $(z_{ij})_{|O| \times |F|}$ is an optimal placement policy.
Begin
Initialization: Let $C_p = \infty, q = |O|, z_{ij} = 1 (\forall i \in O, \forall j \in F)$;
L1: Let $C_{p'} = 0, B_{ij} = 0 (\forall i \in O, \forall j \in F)$;
For each session pair $\langle s_i, d_i \rangle \in S, (i=1, 2, \dots, n)$, do
First, according network G and $\bar{P}(s_i, d_i)$, build-layered

model of graph G' as follows;

i) G' contains $|\bar{P}(s_i, d_i)|$ layers while each layer is a copy of G ;

ii) Edge $\langle i^{(j-1)}, i^{(j)} \rangle$ span from $(j-1)^{\text{th}}$ layer to j^{th} layer if and only if $z_{ij}=1$;

iii) Set the cost of edge $\langle i^{(j-1)}, i^{(j)} \rangle$ as $c(i, j)$;

Second, find the shortest path, denotes by p_i , in G' from s_i to d_i , using Dijkstra algorithm;

Third, denotes the length of p_i by l_i and let $C'_P = C'_P + l_i$;

Fourth, let $B_{ij} = B_{ij} + 1$ for each j^{th} layer node i in path

p_i ;

If $C_P - C'_P \leq M$ then goto L2;

For each j in F , do

First, sort $\{B_{i,j}, i \in O\}$ as $B_{i_1,j} \geq B_{i_2,j} \geq \dots \geq B_{i_q,j} \geq \dots$

$B_{i_q,j}$ order;

Second, for each $i \in O$, do

If, $i \in \{i_1, i_2, \dots, i_q\}$, let $z_{ij}=1$;

Otherwise, let $z_{ij}=0$;

Let $q=q-1$, then goto L1;

L2: Output current placement policy matrix $(z_{ij})|O| \times |F|$, halt.

End

算法复杂性分析(注意区分覆盖节点集合 O 和计算复杂性函数 $O(\cdot)$): 计算 n 个会话从源端到目的端最短路径的计算复杂度代价为 $O(n \cdot (|V| \log |V| + |E| + |O|))$; 而算法中针对 q 的寻优操作最多重复 $|O|-1$ 次, 故该算法的复杂度以 $O(|O| \cdot n \cdot (|V| \log |V| + |E| + |O|))$ 为上限, 实际计算量要远小于这个上限。

图 5(a) 给出了一个为 10 个节点的网络拓扑部署多因子服务的例子。设主动节点集合为 $A = \{2, 4, 7, 8, 9, 10\}$, 在每个主动节点上建设每类服务的代价如表 1。

表 1 服务代价

主动节点	2	4	7	8	9	10
服务因子 1	10	20	30	∞	30	10
服务因子 2	20	10	10	10	∞	10
服务因子 3	20	20	10	10	10	∞

设有 5 组会话, 其服务次序约束如表 2。

表 2 服务次序约束

会话组	源端	目的端	服务因子次序约束
S1	1	4	1
S2	1	6	1→2→3
S3	1	12	2→1
S4	11	5	2→3
S5	7	1	3→2

下面通过示例详细解释该算法一次运行中的几个关键步骤:

1) 置 $z_{ij}=1, B_{ij}=0, q=|O|-1=5, (\forall i, j), C_P=\infty$, 设 $M=10$;

2) 对会话 S1, 在分层图 5(b) 上可求得从第 0 层的 1 号节点到第 1 层 4 号节点的最短路径 $\min \text{Path}(1^{(0)}, 4^{(1)}) = \{1^{(0)}, 2^{(1)}, 2^{(1)}, 3^{(1)}, 4^{(1)}\}$, 其路径长度为 15。在此路径中, 2 号主动节点承担了第 1 类服务, 故 $B_{2,1}++$;

3) 对会话 S2, 在分层图 5(c) 上可求得从第 0 层的 1 号节点到第 3 层 6 号节点的最短路径 $\min \text{Path}(1^{(0)}, 6^{(3)}) = \{1^{(0)}, 2^{(1)}, 2^{(1)}, 9^{(1)}, 8^{(1)}, 7^{(1)}, 7^{(2)}, 7^{(3)}, 6^{(3)}\}$, 其路径长度为 37。在此路径上, 2 号节点承担了第 1 类服务, 7 号节点同时承担了第 2 和第 3 两类服务, 故 $B_{2,1}++, B_{7,2}++, B_{7,3}++$;

4) 对会话 S3, 在分层图 5(d) 上可求得从第 0 层的 1 号节点到第 3 层 6 号节点的最短路径 $\min \text{Path}(1^{(0)}, 12^{(2)}) = \{1^{(0)}, 10^{(0)}, 10^{(1)}, 10^{(2)}, 11^{(2)}, 12^{(2)}\}$, 其路径长度为 25。在此路径上, 10 号节点承担了第 2 和第 3 类的服务, 故 $B_{10,2}++$,

$B_{10,1}++$;

5) 对会话 S4, 在分层图 5(e) 上可求得从第 0 层的 11 号节点到第 2 层 5 号节点的最短路径 $\min \text{Path}(11^{(0)}, 5^{(2)}) = \{11^{(0)}, 8^{(0)}, 8^{(1)}, 7^{(1)}, 7^{(2)}, 4^{(2)}, 5^{(2)}\}$, 其路径长度为 25。在此路径上, 8 号节点承担了第 2 类服务, 7 号节点承担了第 3 两类服务, 故 $B_{8,2}++, B_{7,3}++$;

6) 对会话 S5, 在分层图 5(f) 上可求得从第 0 层的 7 号节点到第 2 层 1 号节点的最短路径 $\min \text{Path}(7^{(0)}, 1^{(2)}) = \{7^{(0)}, 8^{(0)}, 9^{(1)}, 9^{(1)}, 10^{(1)}, 10^{(2)}, 1^{(2)}\}$, 其路径长度为 26。在此路径上, 9 号节点承担了第 3 类服务, 10 号节点承担了第 2 两类服务, 故 $B_{9,3}++, B_{10,2}++$;

7) 这样, 5 对会话总的路径长度为 $C'_P=128$, 因 $C_P - C'_P > M$, 故还需进一步寻优。令新的 $C_P=128, q=q-1=4$ 。此时 B 矩阵变为:

	1	2	3
2	2	0	0
4	0	0	0
7	0	1	2
8	0	1	0
9	0	0	1
10	1	2	0

对 B 矩阵的每一列, 选值最大的 q 行(在此可引入其它启发式策略)。例如, 第 1 列需要从节点 4、7、8、9 中淘汰 2 个, 不妨选建设代价最大的那个, 这里选 8 号和 9 号节点, 置 $z_{8,1}=0, z_{9,1}=0$, 而其它的不变。即淘汰了 8 号和 9 号节点部署服务因子 1 的权利。类似, 淘汰 2 号和 9 号节点部署服务因子 2 的权利($z_{2,2}=0, z_{9,2}=0$), 淘汰 2 号和 10 号节点部署服务因子 3 的权利($z_{2,3}=0, z_{10,3}=0$)。

8) 在新 $z_{ij}, B_{ij}, q, (\forall i, j), C_P$ 条件下, 重复第 2 步到第 7 步, 直到 $C_P - C'_P \leq M$ 。

5 仿真

为了测试 MFPP 算法的性能, 采用 BRUTE 工具^[12]来生成带有 AS 区域的拓扑图。在 AS 图上随机地选择覆盖节点, 覆盖节点之间的边由 KMST 算法^[1]生成, 如图 6 所示。在我们的实验中, 网络有如下属性: a) 无向图; b) 边的费用在 $[5, 10]$ 上均匀分布; c) 平均每个路由器节点按 Waxman 模型来产生, 其中连接边数 $m=4, \alpha=0.15, \beta=0.2$; d) 服务因子相对应的服务代价在集合 $[5, 10]$ 上均匀分布; e) 路由器节点的个数为 100, AS 为 10 个, AS 的分配按均匀分布; f) 选择 20% 的节点覆盖节点, 均匀分布在各个 AS 域中。

在模拟实验中, 把会话的节点对随机地接入到网络拓扑的覆盖节点上。假设有 4 个服务因子, 每组会话的服务次序和服务因子的个数随机生成(不能大于 4 个)。随着会话个数的变化, 观察解的质量(即生成的代价)和算法的执行效率。仿真结果如图 7、8 所示。可以看出, 随着会话数的增多, 代价增大, 执行时间也加长; 服务因子越多, 代价越大, 执行时间也越长。

为了对比算法 MFPP 的性能, 我们还设计了随机放置算法和遗传算法作为参照系。随机给定 5 组会话和相应的服务因子次序, 遗传算法的种群初始规模为 30, 最大进化代数为 100。图 9 给出了三个算法求得的代价与服务因子个数之间的关系。可以看出: 从解的质量看, MFPP 算法优于随机算法, 并且代价随着服务因子的增加呈线性增加; 从算法的效率看, MFPP 算法远远优于遗传算法。综合地看, MFPP 算法是解的质量和执行时间两方面的合理的折衷。

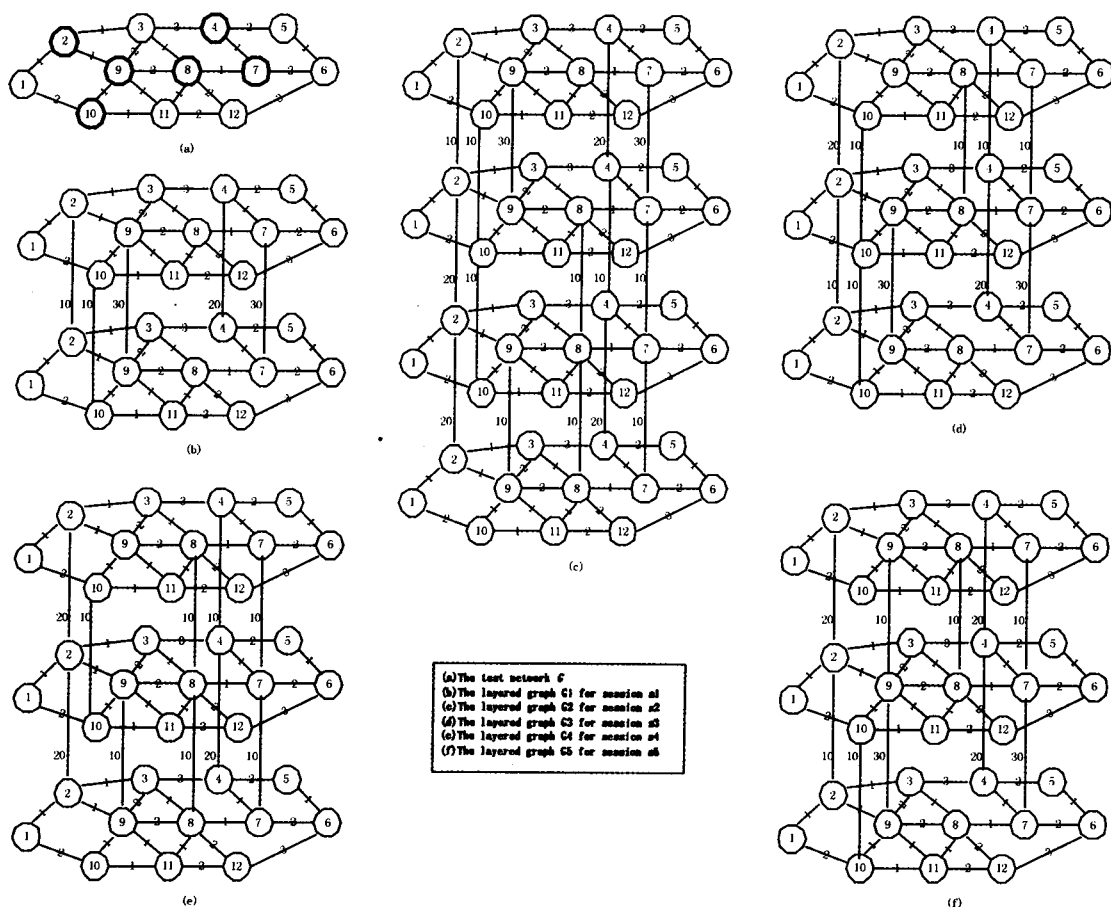


图 5 多因子的服务放置问题示例

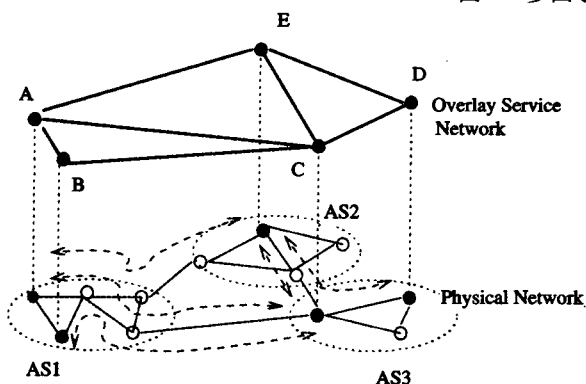


图 6 测试用例的拓扑结构示意图

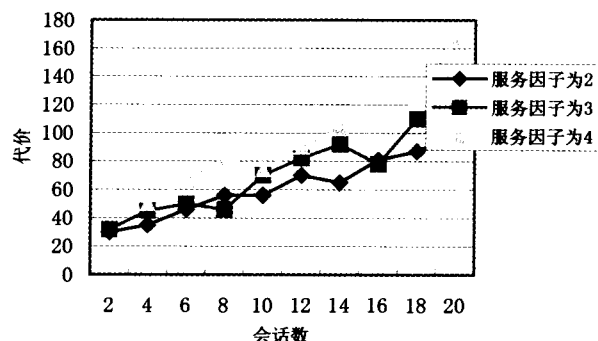


图 7 服务因子、会话数、代价之间的关系

求解该模型的启发式算法,分析了该算法的复杂度。最后,对该算法进行了仿真实验。通过实验分析对比,表明本文的算法有较好的效果。

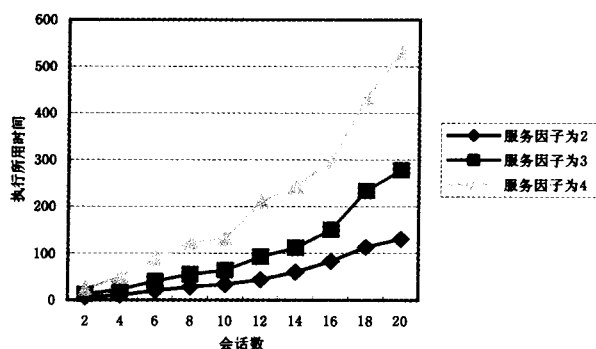


图 8 服务因子、会话数、执行所用时间之间的关系

结束语 本文首先讨论了覆盖服务网络(OSN)上的服务分类问题,引入了服务因子的概念,进而提出了多服务因子的服务放置问题(MFPP),建立了该问题的优化模型,并设计了

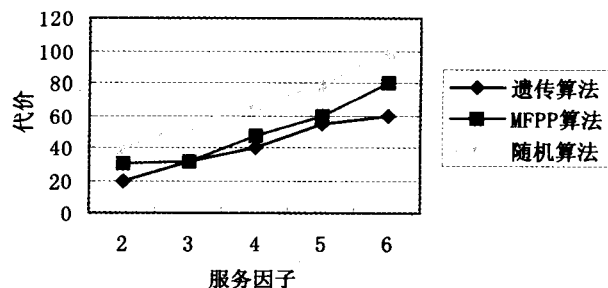


图 9 三个算法的性能比较

- 3 Kotz D, Gray R S. Mobile agents and the future of the Internet. *ACM SIGOPS Operating Systems Review*, 1999, 33(3): 7~13
- 4 Palathingal P, Chandra S. Agent approach for service discovery and utilization. In: *Proc. of the 37th Annual Hawaii International Conference on System Sciences*, Jan. 2004
- 5 Bellavista P, Corradi A, Stefanelli C. Mobile agent middleware for mobile computing. *IEEE Computer*, 2001, 32(3): 73~81
- 6 Cao Jiannong, Tse D C K, Chan A T S. PDAgent: A platform for developing and deploying mobile agent-enabled applications for wireless devices. In: *ICPP 2004*. Montreal, Canada, Aug. 2004. 510~517
- 7 Liu R, Chen F, et al. Agent-based web services evolution for pervasive computing. In: *Proc. of the 11th Asia-Pacific Software Engineering Conference*, Busan, Korea, 2004. 726~731
- 8 Kotz D, Gray D, Rus D. Future directions for mobile agent research. *IEEE Distributed System online*, 2002
- 9 Aleksy M, Kouthaus A, Schader M. CARLA - a CORBA-based architecture for lightweight agents. In: *Proc. of the IEEE/WIC International Conference on Intelligent Agent Technology*, Halifax, Canada, Nov 2003. 111~118
- 10 Montanari R, Tonti G, Stefanelli C. Policy-based separation of concerns for dynamic code mobility management. In: *Proc. of 27th Annual International Computer Software and Applications Conference*. Dallas, Texas, Nov 2003. 82~90
- 11 Smith B C. Reflection semantics in a procedural programming language: [dissertation]. MIT, 1982
- 12 Maes P. Concepts and experiments in computational reflection. *ACM SIGPLAN Notice*, 1987, 22(12): 147~155
- 13 Lange D B. Mobile objects and mobile agents: The future of distributed computing? In: *Proc. of the ECOOP'98*. Brussels, Belgium, July 1998. 1~12
- 14 SUN Microsystems. Java 2 Platform, Standard Edition, Version 1.4.2, 2003
- 15 Chiba S. Load-time Structural Reflection in Java. In: *Proc. of the ECOOP'2000*. Malaga, Spain, Jun. 2000. 313~336
- 16 Gamma E, Helm R, Johnson R, et al. Design patterns: Elements of reusable object-oriented software. Boston: Addison-Wesley Professional, 1995
- 17 Picco G P. μ Code: A lightweight and flexible mobile code toolkit. In: *Proc. of 2nd International Workshop on Mobile Agents*. Berlin, Germany, 1998. 160~171
- 18 Ledoux T. Adaptability in mobile agent systems using reflection. *RM 2000, IFIP/ACM Workshop on Reflective Middleware, Middleware'2000*. New York, 2000
- 19 Watanabe T, Noriki A, Shinbori K. A reflective framework for reliable mobile agent. *ECOOP 2000, Workshop on Reflection and Metalevel Architectures*. Cannes, 2000
- 20 Lange D B, Oshima M, Mitsuru O. Programming and deploying mobile agents with aglets. Boston: Addison-Wesley Professional, 1998
- 21 Wong D, Paciorek N, Walsh T, et al. Concordia: An infrastructure for collaboration mobile agents. In: *Proc. 1st International Workshop on Mobile Agents 97*. Berlin, Germany, Apr 1997. 98~110

(上接第4页)

参 考 文 献

- 1 Li Zhi, Mohapatra P. Impact of Topology On Overlay Routing Service. http://www.ieee-infocom.org/2004/Papers/09_1.PDF
- 2 Chu Yang-hua, Rao S, Zhang Hui. A Case for End System Multicast. In: *Proceedings of ACM Sigmetrics*, June 2000. 1~12
- 3 Jannotti J, Gifford D K, Johnson K L, et al. Overcast: Reliable Multicasting with an Overlay Network. *Proceedings of Operating Systems Design and Implementation(OSDI)*, October 2000
- 4 Francis P. Yoid: Your own Internet distribution: [Tech Rep]. www.aciri.org/yoid, UC Berkeley ACIRI Tech Report, Apr. 2000
- 5 Chawathe Y. Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service: [PhD thesis]. Department of EECS, UC Berkeley, Dec. 2000
- 6 Pendarakis D, Shi S, Verma D, et al. ALMI: An application level multicast infrastructure. *3rd Usenix Symposium on Internet Technologies and Systems(USITS)*, Mar. 2001
- 7 Li Zhi, Mohapatra P. QRON: QoS-Aware Routing in Overlay Networks. *QRON IEEE Journal on Selected Areas in Communications*, 2004, 22(1): 29~40
- 8 Andersen D G, Balakrishnan H, Kaashoek M F, et al. Resilient Overlay Networks. In: *Proc. 18th ACM SOSP*, Banff, Canada, October 2001
- 9 Duan Zhenhai, Zhang Zhili, Hou Yiwei T. Service overlay networks, SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Trans Netw*, 2003, 11(6): 870~883
- 10 Subramanian L, Stoica I, Balakrishnan H, et al. OverQoS: An Overlay Based Architecture for Enhancing Internet QoS. *NSDI*, 2004. 78~84
- 11 Yun Pan. Research on Active Overlay Network and its key technology: [Ph D Thesis]. the China of Mining and Technology University, December 2003
- 12 Charikar M, Guha S. Improved combinatorial algorithms for the facility location and k-Median problems. *IEEE Symposium on Foundations of Computer Science*
- 13 Shmoys D B, Tardos E, Aardal K. Approximation algorithms for facility location problems. In: *29th ACM Symposium on Theory of Computing*, 1997. 265~274
- 14 Medina A, Lakhina A, Matta I, et al. BRITE: An Approach to Universal Topology Generation. In: *Proc. of MASCOTS '01*, August 2001
- 15 Waxman B. Routing of Multipoint Connections. *IEEE Journal on Selected Areas in Communications*, 1988, 6(9): 1617~1622
- 16 Yu Z W, Pan Y, Wang L C. On the model of OoS multicast routing problems in active networks. *International Conference on Computer Networks and Mobile Computing, Proceedings*, 2003. 419~422
- 17 Choi S, Turner J, Wolf T. Configuring Sessions in Programmable Networks. *Computer Networks*, 2003, 41(2): 269~284