服务覆盖网中保证 QoS 的服务路由算法①

刘 妍 李 俊 吴 刚 (中国科学技术大学 自动化系 安徽 合肥 230027)

摘 要： 针对服务覆盖网的服务路径选择问题，提出改进的 KCP 算法。在后继结点选择时，使用节点延迟、剩余资源、链路带宽、延迟等表征节点和链路性能的多项指标，组合成综合性能函数；引入调节因子兼顾负载均衡和节点复用率。仿真试验表明，此算法能够有效地进行服务路由选择，并且达到良好的负载均衡和高节点复用率。

关键词： 服务覆盖网；服务组合；服务路由；服务质量；负载均衡

QoS-Guaranteed Service Routing in Service Overlay Network

LIU Yan, LI Jun, WU Gang (Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: In this paper, a modified KCP algorithm is presented to replace single delay parameter by an aggregate function when we decide which path to choose, the aggregate function takes nodes' and links' comprehensive performance parameters such as the nodes' delay, available resource and the links' delay, bandwidth into consideration, besides, a new adjusting factor is proposed to give attention to both load-balancing and node-multiplexing. The simulation results verify the validity of this new algorithm in the service routing selection, it can also adjust the whole system to a well load-balancing and high node-multiplexing state at meanwhile.

Keywords: SON(Service Overlay Networks); service composition; service routing; QoS(Quality of Service); load-balancing

1 引言

随着互联网服务在种类和数量上的飞速增长，如何有效地利用互联网上已有的服务为用户提供种类多样、功能强大的个性化服务成为一个急需解决的问题。服务覆盖网络 SON(Service Overlay Networks)对这个问题提供了一种解决方案。典型的 SON 如图 1 所示。服务组合就是通过覆盖节点之间的相互通信和协作，将部署在节点上的功能简单的服务组件组合成复杂的、具有新功能的增值服务的过程。

但是一个服务组合可以有多条服务路径来实现，服务路由算法直接影响着服务质量和用户体验，是服务组合的核心。服务路由需要考虑以下几个问题：第一，满足用户的 QoS 约束，用户服务质量是首要问题；第二，负载均衡，为确保整个网络能长期有效的运行，必须保证负载在各个网络节点之间均衡分配；第三，提高节点复用率，以减少路由次数、缩短服务路径长度；第四，为用户提供各方面性能尽可能好的服务，如延迟、服务价格、稳定性等。

目前服务路由算法主要有 QUEST[1]算法、LIAC[2]算法和 KCP[3]算法。QUEST 算法对 QoS 和负载均衡度量指标因素考虑的比较全面，LIAC 算法在层次图上寻找最短路径实现负载均衡。但 Jaffe[4]指出，利用线

① 基金项目:国家高技术研究发展计划(863);安徽省高校自然科研项目(KJ2008A106)
收稿时间:2010-04-19;收到修改稿时间:2010-05-23

性函数拟合多重约束无法满足所有的 QoS 约束，因此 KCP 算法提出从出口节点开始搜索服务组件，并生成最小延迟树，深度遍历此树并保留符合 QoS 约束的服务路径。虽然 KCP 算法在组合成功率和满足 QoS 约束上较 QUEST 和 LIAC 算法有较大改进，但是它考虑各项性能指标时过于简单，得到的路径并非最佳，也不能同时兼顾节点复用和负载均衡。因此，本文对 KCP 算法进行了改进，提出 MKCP(Modified KCP)算法，在保留 KCP 原有优点的基础上，更加全面考虑了表征节点和链路性能的指标，如节点延迟、剩余资源和链路延迟、带宽等；同时引入调节因子[5]，兼顾整个 SON 的负载均衡和节点复用。

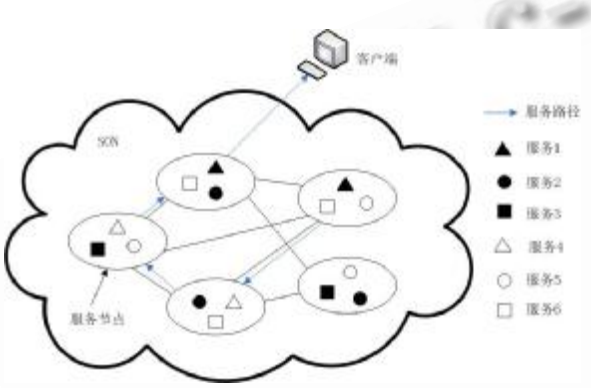

图 1 服务覆盖网

2 MKCP服务路由算法

2.1 问题描述

服务覆盖网中有 $O_1, O_2, ..., O_N$ 共 N 个服务节点，能提供 $S_1, S_2, ..., S_M$ 共 M 种服务组件。每种服务可以在多个节点部署副本，每个服务节点也能同时部署多种服务。用户请求的服务序列为 $S_{req}=\left(S_K^1, S_J^2, ..., S_T^n\right)$，由 n 个服务组成，其中 s_i^j 代表服务组合序列中的第 j 个服务为 s_i。另外，对此服务组合的 QoS 约束为 $QoS(S_{req})=\left(QoS(S_K^1), QoS(S_J^2), ..., QoS(S_T^n)\right)$。服务路由需要在整个网络中找到一条服务路径 $Sp=\left(S_K^1/O_k, S_J^2/O_j, ..., S_T^n/O_t\right)$，使其能够提供用户要求的服务、满足用户的 QoS 约束并实现性能最优。其中 S_J^i/O_j 代表服务组合中的第 i 个服务 S_J 在节点 O_j 上完成。

2.2 性能指标

服务路由中要考虑多方面的性能指标，主要包括服务节点性能指标和链路性能指标。

2.2.1 服务节点性能指标

$q_p(S_j/O_i)$ 服务价格，服务节点 O_i 提供服务 S_j 的费用；

$q_d(S_j/O_i)$：节点延迟；

$q_{abi}(O_i)$：节点剩余服务能力；

$q_{repu}(S_j/O_i)$：信誉度，使用者对服务的评价；

$q_{relia}(O_i)$：可靠性，O_i 节点成功提供服务的次数与总的调用次数的比值。

2.2.2 链路性能指标

$q_d(l_i)$：链路延迟，链路 l_i 的传输时间；

$q_{abw}(l_i)$：可用带宽，链路 l_i 当前可用带宽；

$q_{relia}(l_i)$：可靠性，即(1-丢包率)。

2.2.3 性能指标标准化

根据以上性能指标，QoS 约束可细化为 $QoS(S_J^i)=(q_p^{req}(S_J^i), q_d^{req}(S_J^i), ..., q_{relia}^{req}(l_{S_J^i-S_k^{i+1}}))$，其中每一分项对应前面讨论的性能指标。

由于各性能指标量纲不同，取值范围的数量级相差很大，并且有些指标取值越大，性能评价越低，有些则相反，所以需要对它们进行如下标准化处理：

$$q'=q/q^{req} \quad (1)$$

$$q'=q^{req}/q \quad (2)$$

q 代表节点或链路实际的性能，q^{req} 代表用户对节点或链路的 QoS 约束。在前面所考虑的各项性能指标中，服务价格、节点延迟、链路延迟取值越小，性能越好，因此适用于公式(1)。而节点剩余服务能力、信誉度、节点可靠性、链路可用带宽和链路可靠性适用于公式(2)。如果性能指标不满足用户的需求，那么标准化之后的值将大于 1，所以在后面的算法中，只有各项性能指标都小于 1 的节点和链路才会作为备选。

2.3 性能函数

综合上节的各项性能指标，得到节点的性能函数为：

$$F(S_j/O_i)=\{w_1[q'_p(S_j/O_i)]^2+w_2[q'_d(S_j/O_i)]^2+w_3[q'_{abi}(O_i)]^2+w_4[q'_{repu}(S_j/O_i)]^2+w_5[q'_{relia}(O_i)]^2\}^{1/2} \quad (4)$$

其中$w_1+w_2+w_3+w_4+w_5=1$

链路的性能函数为：

$$F(l_i)=\{l_1[q'_d(l_i)]^2+l_2[q'_{abw}(l_i)]^2+l_3[q'_{relia}(l_i)]^2\}^{1/2} \quad (5)$$

其中$l_1+l_2+l_3=1$

综合性能函数为：

$$F=\left[j_1F^2(S_j/O_i)+j_2F^2(l_i)\right]^{1/2} \quad (6)$$

其中$j_1+j_2=1$

在(4)(5)(6)三个公式中的权值可以根据用户的具体需求来选择。

2.4 调节因子

在服务路径的选择上，为了同时达到负载均衡和高节点服用率两个目标，引入调节因子进行定量讨论。引入调节因子后的性能函数为：

$$F'=\begin{cases}pF & \text{前驱节点仍是自身}\\ F & \text{其他}\end{cases} \quad (7)$$

p 的取值范围是 $(0,1]$。p 越小，综合性能函数越小，就越倾向于选择在同一个节点完成相连的两个或多个服务，服务路径就越短。反之则倾向于选择在尽可能多的节点上完成服务，使负载均衡。

至此得到了最终的综合性能函数 F'，服务路由算法的最终目标是在所选节点和链路各项性能指标值小于 1 的前提下，选择一条使综合性能函数值最小的服务路径。

2.5 算法描述

假设已知用户请求的服务序列为 $(S_k^1,S_j^2,...,S_r^n)$，服务出口节点为 O_r，出口节点不提供任何服务，仅作为服务的出口。

算法分为两步：构建一个 K 叉树，每一条从叶到根的路径代表一条可能的服务路径；深度优先遍历此树得到最终服务路径。

在第一步中，首先设出口节点 O_r 为树的根节点，即第 0 层。在选择树的第 i 层时，需满足条件：

a) 能够提供 S_x^{n+1-i} 服务；

b) 节点和链路的各项性能指标标准化之后小于 1；

c) 在满足条件 a)和 b)的节点中，根据综合性能函数 F' 从小到大排序的前 K 个。

直到第 n+1 层，k 叉树生成完毕。

在第二步中，对 K 叉树进行深度优先遍历的同时要完成两件任务。一是进行 QoS 约束检查，如果满足客户提出的总的 QoS 约束，继续遍历，否则将此节点以下截掉，不再考虑。二是对于满足 QoS 约束的路径，始终选择性能合计函数最小的一条。

遍历后如果没有一条路径满足条件，就拒绝服务。如果有结果，那么就根据此结果为用户提供服务。

3 仿真实验

在实验中，有服务节点 20 个，客户端节点 60 个，服务共有 10 种，每种服务有 6 个副本，每个服务节点上部署 3 种服务。每次客户端发起的服务由 2~10 种服务组成。其中，各参数取值为 $K=3$，$w_1=w_2=w_3=w_4=w_5=1/5$，$l_1=l_2=l_3=1/3$，

$j_1=j_2=1/2$，$p=0.9$。

3.1 实验一：服务成功率

为比较 KCP 和 MKCP 两种算法的服务成功率，进行以下实验：60 个客户端分别每隔 180 秒、90 秒、60 秒、45 秒、36 秒、30 秒发送一次服务组合请求，对应的请求率为 20 次/分钟、40 次/分钟、60 次/分钟、80 次/分钟、100 次/分钟、120 次/分钟，每次服务持续 5~30 秒之间的随机值。实验结果如图 2 所示。

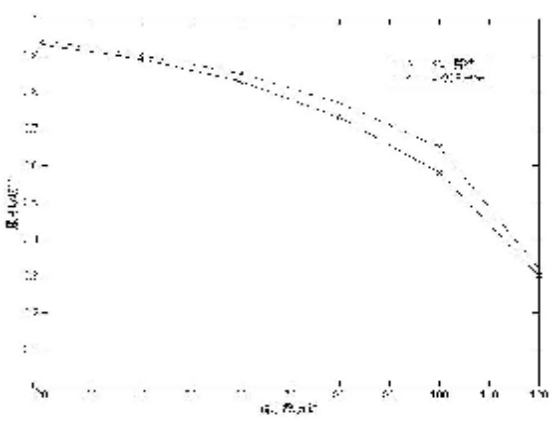

图 2　KCP 算法和 MKCP 算法成功率比较

从图 2 可以看到，在请求率比较低，系统负载较

低，两种算法的成功率相差不多，这是因为服务节点还不存在因负载过重无法提供服务的情况，但随着请求率的不断增加，MKCP 的成功率明显高于 KCP，因为 MKCP 考虑了节点因负载过重而无法提供服务的情况，同时还考虑了节点的可靠性等方面的因素，因此能够提高服务的成功率。但随着请求率的进一步增加，对于整个网络来说负载过重，两种算法的成功率趋于一致。

因此通过分析实验一的结果，可以看出，在负载较轻或过重时，KCP 和 MKCP 两种算法在成功率方面的性能大致相同，但在负载适中时，MKCP 算法性能更优。实验数据表明，MKCP 算法的服务成功率比 KCP 算法最多高 7%，最少高 1%，平均高 2.8%。

3.2 实验二：路径长度

为比较两种算法的路径长度，进行以下实验：60 个客户端每隔 60 秒发起一次服务组合请求，请求率为 60 次/分钟每次持续时间为 5~30 秒之间的随机数。对于 MKCP 算法，进行两次实验，调节因子 p 分别取 0.9 和 0.7，实验结果如图 3 所示。

从图 3 可以看到，MKCP 算法的服务路径长度要小于 KCP 算法的。实验数据表明，KCP 算法的平均路径长度为 5.77，而 MKCP 算法当 p=0.9 时平均路径长度为 5.13，当 p=0.7 时平均路径长度为 4.56。对于 MKCP 算法的两次实验，p 参数越小，服务路径越短。这与前面对 p 的讨论相一致。

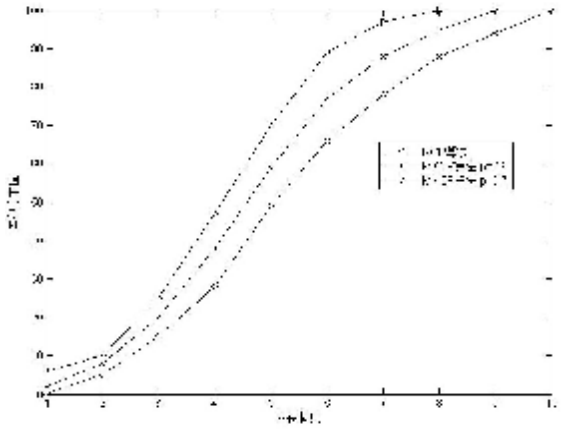

图 3 KCP 算法和 MKCP 算法路径长度比较

3.3 实验三：负载均衡

为测试 MKCP 的负载均衡能力，进行以下实验：60 个客户端分别每隔 180 秒、90 秒、60 秒、45 秒、36 秒、30 秒发送一次服务组合请求，对应的请求率为 20 次/分钟、40 次/分钟、60 次/分钟、80 次/分钟、100 次/分钟、120 次/分钟，每次服务持续 5~30 秒之间的随机值。对于 MKCP 算法，分别取 p 因子值 0.9 和 0.7 进行两次实验。把每个服务节点参与的服务组合数量的方差作为衡量负载均衡的指标。得到的结果如图 4 所示。

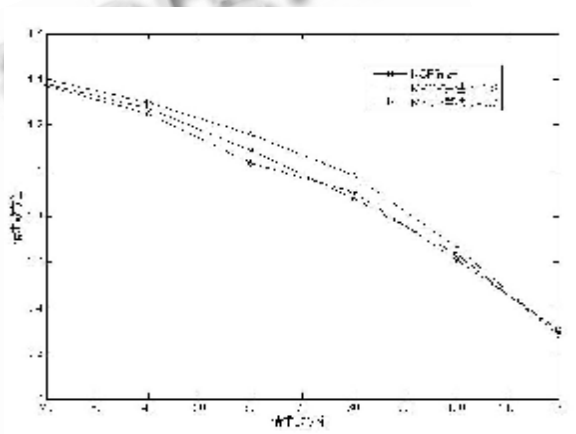

图 4 KCP 算法和 MKCP 算法的负载均衡比较

从图 4 可以看出，MKCP 算法能具有与 KCP 算法在负载均衡方面的性能相当。

3.4 总结

分析以上三个实验的结果，MKCP 算法不仅兼顾了负载均衡、节点复用两方面的性能，而且与 KCP 算法相比具有更高的服务成功率。

4 结论

本文对 KCP 算法进行了改进，提出 MKCP 算法。它在后继结点选择上使用如节点延迟、剩余资源和链路带宽、延迟等表征节点和链路性能的多项指标组合而成的综合性能函数代替原算法中单一的延迟参数作为判断条件，引入调节因子来兼顾负载均衡和节点复用率。仿真实验表明，此算法能够有效进行服务路由选择并且达到良好的负载均衡和高节点复用率。但 MKCP 算法中有 K 和 p 两个不定的因子，这两个参数的具体取值以及各种取值的组合会对结

(下转第 250 页)

(上接第 215 页)

果造成什么影响尚未考虑，这些问题留待今后进一步研究。

参考文献

1 Gu X, Nahrstedt K, Chang RN, Ward C. QoS-Assured Service Composition in Managed Service Overlay Networks. In: Proc. of the 23rd Int'l Conf. on Distributed Computing Systems(ICDCS 2003). Providence: IEEE Computer Society, 2003,194 − 203.

2 Raman B, Katz RH. Load balancing and stability issues in algorithms for service composition. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003). San Franciso: IEEE Communications Society. 2003,1477 − 1487.

3 Manish Jain, Puneet Sharma, et al. QoS-guaranteed Path Selection Algorithm for Service. 14th International Workshop on Quality of Service (IEEE Cat No. 06EX1425). 2006:288 − 289.

4 Jaffe JM. Algorithms for finding paths with multiple constraints. Networks. 1984,vol.14,pp.95 − 116.

5 陈香兰,李曦,龚育昌. 服务组合中一种服务组合路径优化方法研究. 小型微型计算机系统, 2008,29(9): 1569 − 1573.