

文章编号: 1007-5321(2017)增-0048-05

DOI:10.13190/j.jbupt.2017.s.011

# 基于部分互斥方法的呼叫中心路由并行算法

张文涛<sup>1,2</sup>, 双 锴<sup>1</sup>, 万 能<sup>3</sup>, 詹舒波<sup>1</sup>, 苏 森<sup>1</sup>

(1. 北京邮电大学 网络与交换技术国家重点实验室, 北京 100876; 2. 中国电子科技集团第五十四研究所 中国通信网信息传输与分发技术重点实验室, 石家庄 050081; 3. 北京邮电大学 电子工程学院, 北京 100876)

**摘要:** 针对多技能呼叫中心路由计算的并发性能瓶颈, 提出一种并行的呼叫中心路由方法(PCCRM)。该方法通过临界资源粒度细化, 将互斥范围缩小到局部资源, 可显著减少进程间的同步等待, 同时引入有序竞争模型来保证算法满足呼叫中心路由基本原则。实验结果表明, PCCRM可以有效提升路由算法的系统处理性能和扩展性。

**关键词:** 呼叫中心路由; 座席技能; 临界资源; 互斥

中图分类号: TN911.22

文献标志码: A

## Parallel Routing Algorithm for Call Centers Based on Local Mutual Exclusion

ZHANG Wen-tao<sup>1,2</sup>, SHUANG Kai<sup>1</sup>, WAN Neng<sup>3</sup>, ZHAN Shu-bo<sup>1</sup>, SU Sen<sup>1</sup>

(1. State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing 100876, China)

(2. China Electronics Technology Group Corporation 54th Research Institute, Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, Shijiazhuang 050081, China)

(3. Electric Engineer School, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** A parallel routing method named parallel call center routing method(PCCRM) was proposed to promote the multiprocessing performance of call center system with multi-skill model. Through refinement of resource, PCCRM reduced the general mutual exclusion to local mutual exclusion, which can markedly reduce the waiting of process synchronization. Meanwhile an orderly competition model was introduced to guarantee the algorithm to meet the requirement of general principle of call center's agents. The experimental results showed that the PCCRM could increase system performance and scalability of routing algorithm.

**Key words:** call center routing; agent skills; critical resource; mutual exclusion

云呼叫中心的路由排队算法的扩展性问题是制约呼叫中心云化发展的技术瓶颈。在对多技能组呼叫中心路由分配算法与资源竞争模型分析的基础上, 基于分布式互斥方法<sup>[1-8]</sup>提出一种可以提升路由算法并行程度的并行呼叫中心路由方法(PCCRM, parallel call center routing method)。

PCCRM方法利用技能队列与座席之间的资源竞争关系, 对临界资源粒度进行细化, 形成一种基于局部资源互斥的自由竞争模型, 可以减少临界区资源占用的数量, 提升进程执行路由算法的并行程度。自由竞争模型会破坏路由算法的基本原则, 需要引入约束限制, 形成有序竞争模型。最后通过实

收稿日期: 2016-05-03

基金项目: 国家重点基础研究发展计划项目(2011CB302506)

作者简介: 张文涛(1979—), 男, 博士后, Email: djangot@163.com; 苏 森(1971—), 男, 教授, 博士生导师。

验进行验证和分析。

## 1 呼叫中心多技能模型

呼叫中心座席多技能共享模型如图 1 所示,多个技能队列相互之间分享座席资源,座席具备一个或多个技能,在不同的技能中分配不同的等级。

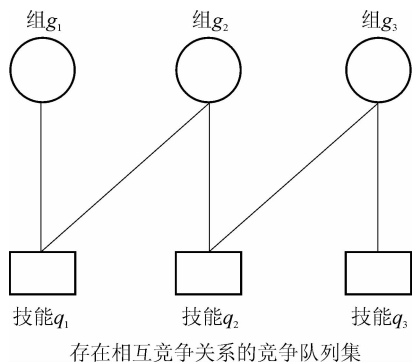


图 1 竞争队列集

针对要解决的问题,建立呼叫中心多技能路由模型  $M$ 。

**模型  $M$  定义:**  $M = (A, S, V)$ ,  $A = \{a_i | 1 \leq i \leq N\}$ ,  $S = \{s_j | 1 \leq j \leq M\}$ ,  $V = \{v_{ij} | 1 \leq i \leq N, 1 \leq j \leq M\}$ ,  $v_{ij} = f(a_i, s_j)$ 。满足以下两个约束:

$$1) \forall v_{ij} \in V, v_{ij} \geq 0;$$

$$2) \forall a_i \in A, \text{存在一个 } 1 \leq k \leq M \text{ 和任意的 } j \neq k, 1 \leq j \leq M, \text{使得 } v_{ik} > v_{ij}.$$

其中  $A$  为系统中的座席集合,  $N$  为系统座席数量,  $S$  为系统中技能集合,  $M$  为系统技能数量,  $V$  为系统中座席和技能的关系, 用一个二元组  $v_{ij} = f(a_i, s_j)$ 。其中  $v_{ij}$  为  $a$  在  $s$  中的技能等级的数值, 如果  $v_{ij} < 0$  表示座席不具备该技能。模型  $M$  的两个约束表示: 每个座席  $a$  均具有全部的技能, 都有且仅有一个主技能。

## 2 呼叫中心路由算法分析

路由算法需从座席状态、座席工作关键性能指标 (KPI, key performance indication) 数据、座席技能等级以及客户等级、请求的服务等多种维度进行综合考虑, 在对座席、客户进行全面的评估基础上, 分配一个最佳匹配座席。

### 2.1 路由算法基本原则

#### 1) 工作量公平性原则

该原则考察座席工作量的均衡。策略为根据座席 KPI 数据, 计算一个综合评估值  $W$ , 从座席集合中取综合值  $W$  排序最大的座席为当前分配座席。

综合评估值的计算方法为  $W = f(p_1 + p_2 + \dots + p_n)$ , 其中  $p_i$  为 KPI 数据的归一化值。

#### 2) 服务匹配原则

该原则考虑客户的等级和座席的技能等级匹配性, 使用客户请求、客户等级和座席技能信息, 指定一个或多个等级座席按优先级来服务。

### 2.2 路由算法复杂度分析

整体算法的时间复杂度可以用  $O(Kn)$  表示, 其中  $n$  代表技能队列对应座席集合的数量,  $K$  为根据座席 KPI 计算综合权值  $W$  的时间复杂度, 是一个常量。

对于模型  $M$ , 每个技能覆盖系统全部的座席, 因此座席路由计算的时间复杂度为  $O(K \cdot N)$ , 其中  $N$  为全部座席的数量。

### 2.3 并行度分析

在多技能模型  $M$  中, 由于每个技能包含了全部座席, 每个技能共享相同的资源, 所有的进程在处理路由算法过程时都会竞争同一个资源, 算法处理的并行程度为 1。

## 3 PCCRM 方法

PCCRM 方法解决路由互斥算法的性能瓶颈, 关键是提升多进程的并行程度, 让多个进程可以同时访问座席资源进行路由分配计算。

### 3.1 多技能建模

基于模型  $M$  的定义, 将技能组作为呼叫中心的基本资源调度单元。呼叫中心按组的多技能路由排队模型可以用一个三元组  $C = (G, Q, V)$  来表示。其中  $G$  表示系统中的技能组,  $G$  中技能组相互没有交集。  $Q$  是系统中的技能队列。  $V$  代表  $G$  和  $Q$  关系, 表示技能组中座席具有某种技能。

**模型  $C$  定义:**  $C = (G, Q, V)$ ,  $G = \{g_i | 1 \leq i \leq n\}$ ,  $Q = \{q_i | 1 \leq i \leq m\}$ ,  $V = \{v_{ij} | v_{ij} = f(g_i, q_j), v_{ij} \in [-1, 0, 1], 1 \leq i \leq n, 1 \leq j \leq m\}$ 。  $VG(g_k) = \{q_j | v_{kj} \geq 0\}$ ,  $VQ(q_k) = \{g_i | v_{ik} \geq 0\}$ , 且具有以下约束:

1)  $1 \leq i, j \leq n, i \neq j, g_i \cap g_j = \emptyset$ ;

2)  $1 \leq i \leq n, 1 \leq j \leq m, v_{ij} \geq 0, VQ(q_j) = G, VG(g_i) = Q$ ;

3)  $1 \leq j \leq m$ , 存在唯一的  $i \in [1, n], v_{ij} = 1$ .

其中:  $g$  组中的座席对技能  $q$  存在多种等级,  $v_{ij} \geq 0$  表示  $g_i$  的座席具有技能  $q_j, v_{ij} = 1$  表示  $q_j$  是  $g_i$  的主技能, 主技能的等级是所有组中最高的;  $VG(g_i)$  表示与技能组  $g_i$  相关的所有技能队列, 即组  $g_i$  中座席拥有的技能,  $VQ(q_i)$  表示拥有技能  $q_i$  的所有技能组. 第1个约束表示任意两个技能组没有交集, 第2个约束表示任意一个技能组  $g_i$  均具有全部技能, 第3个约束表示每个技能有且仅有一个主技能组.

$C$  的3个约束表明, 任意座席只能属于一个组, 每个组均具有全技能, 且每个组都有一个主要技能. 因此模型  $C$  符合模型  $M$  的要求.

### 3.2 自由竞争模型

自由竞争模型就是进程在访问临界资源的时候, 不是一次性地占用全部所需资源, 而是逐个对资源加锁访问, 当完成某个资源的访问后立即释放资源, 让其他的进程可以马上进入临界区.

自由竞争模型中进程的互斥等待造成的影响已经非常小. 对资源进行顺序加锁访问时, 当每个进程都按照  $g_1 \rightarrow g_2 \rightarrow \dots \rightarrow g_n$  的顺序对资源进行互斥访问, 那么在理想情况下, 当多个进程进入稳态运行的时候, 多个进程可以并行访问各自资源, 无需等待.

此时系统整体计算时间为  $O((p+r-1)t)$ ,  $p$  为呼叫数,  $r$  为临界资源数量即座席技能分组数,  $t$  为临界区资源访问时间复杂度  $O(Kn)$ ,  $n$  为技能组座席数量, 是系统座席总数  $N$  的  $q$  分之一 (假设均匀分布). 那么单个呼叫平均的处理时间为  $O((p+q-1)NK/pq)$ , 随着系统持续运行呼叫量  $p$  趋于无穷, 平均呼叫处理时间为  $O(nK)$ , 约等于单个临界区执行的时间复杂度, 系统并行程度约等于  $q$ .

自由竞争模型会产生资源过分占用的情况, 导致资源分配破坏路由算法的两大原则. 在图2例子中, 进程1在  $3 \times t$  结束时刻完成资源分配, 在此之前可能会将  $g_1$  或  $g_2$  资源一直占用 (临时) 直至最后时刻释放, 而被进程1临时占用的资源很可能是进

程2和3的呼叫最匹配座席, 但是却无法分配到进程2和3的呼叫, 那么分配给进程2和3呼叫的座席可能不是技能最匹配的, 也可能是工作量不是最少的, 由此违背了路由算法的两条原则. 由于分布式系统执行的不确定性, 这种问题有可能会在系统运行中经常发生, 导致服务质量下降和分配不均衡, 因此不能满足算法的功能性要求.

由此, 需要对自由竞争模型进行约束, 以保证路由算法的功能性, 即严格遵循座席工作量公平性原则和客服匹配原则.

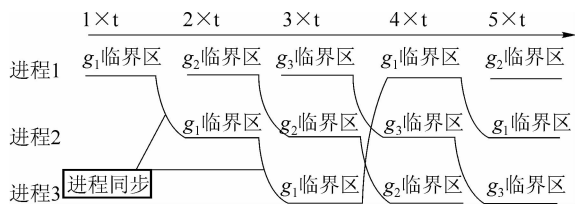


图2 完全竞争的队列

### 3.3 有序竞争模型

有序竞争模型是在自由竞争模型上添加一些约束, 来保证并行方法满足路由算法的两条基本原则.

首先我们提出有序竞争定理: 如果在进程按顺序  $g_1 g_2 g_3 \dots g_n$  访问临界资源过程中, 所筛选的第1个座席就是本次呼叫的最佳匹配座席, 那么就不会破坏路由算法的基本原则.

定理很容易证明: 如果进程第1个筛选的座席来自于  $g_i$ , 则  $g_i$  之前的组中都没有匹配的座席, 自然不会有资源被临时占用. 因此除了  $g_i$ , 其他所有组的座席都没有被临时占用, 其他进程可以从除  $g_i$  之外的组中寻找最匹配座席, 不存在资源的过分占用问题和分配冲突.

为了保证路由算法多进程执行能满足有序竞争定理的要求, 需要下面几条约束.

**第1条约束:** 每个技能组对应一个技能队列, 有且仅有一种技能等级.

**第2条约束:** 任意两个技能组对应同一个技能  $q_{ii}$  的技能等级是不同的.

**第3条约束:** 路由算法根据呼叫所选的技能, 按技能等级由高到低的顺序对具有该技能的技能组进行加锁访问.

根据上述约束,对模型 C 进行增强.

**模型 C 增强 1:**增加二元组  $v_{ij}$  的含义,令  $v_{ij}$  的数值代表技能组  $g_i$  拥有技能  $q_j$  的等级. 如  $v_{ij}=9$  表示  $g_i$  具有  $q_j$  的等级为 9,  $v_{ij}<0$  表示  $g_i$  不具备技能  $q_i$ . 即用  $v_{ij}$  来表示技能组和技能之间的等级.

第 2 条约束要求一个技能对应的多个技能组拥有不同的等级.

**模型 C 增强 2:**对于任一队列  $q_k \in Q, 0 \leq k \leq m$ , 任意两个  $v_{ik} \geq 0, v_{jk} \geq 0$ , 则  $v_{ik} \neq v_{jk}$ .

前两条约束保障了多个技能组对应某个技能的等级是唯一且有序的.

第 3 条约束则要求进程按照技能等级的顺序在选定技能的多个技能组中有序访问. 如果座席路由算法优先考察技能和等级匹配情况,则该约束可以保障进程总是优先去查看最匹配的技能等级座席.

为了满足有序竞争定理要求,还需要增加一条针对路由算法的约束,即座席分配须按技能等级为优先级进行遴选.

**第 4 条约束:**对路由策略的约束,设定路由策略为优先分配技能等级高的座席.

## 4 算法复杂度分析

有序竞争模型对技能组按技能和等级排序,通过约束路由算法的按技能等级高低顺序访问技能组,因此存在最佳情况和最坏情况.

在最佳情况下,每个技能队列都只需访问最佳匹配技能组,即可获得最佳匹配座席. 假设座席分组和呼叫请求都均匀分布,则单个呼叫的计算复杂度为  $O(K \cdot l)$ ,  $l$  为队列长度. 进程并行程度可达到  $m$ ,  $m$  等于技能组数量. 系统整体的呼叫处理时间复杂度达到  $O(Kl/m)$ .

最坏的情况下,所有呼叫需要访问全部的临界资源,单个呼叫的计算复杂度为  $O(KN)$ . 一般情况,这种最坏情况出现的概率非常低,要求从技能队列  $q_k$  对应的组  $g_{k1}$  到  $g_{kn}$  都没有匹配的座席,而且所有队列都没有进入排队(排队状态下呼叫直接进入队列). 这是一个系统临界状态,任何一个呼叫的到来或座席的状态变化,都会打破这种状态. 所以最

坏的计算复杂度  $O(KN)$  在实际运行中只会短暂存在.

## 5 实验验证

在实验中启动了 4 个进程模拟执行路由算法,包括 4 个技能组和 4 个技能队列,每个技能组拥有 4 个技能,每个技能的多个组具有不同的等级.

对 3 种模型进行模拟对比,实验实现方法如下.

1) 普通模型:采用未经优化的分布式互斥算法,设置一个统一的互斥变量.

2) 自由模型:采用自由竞争模式优化的互斥算法,每个进程按固定顺序对技能组进行加锁互斥访问.

3) 有序模型:采用有序竞争模式优化的互斥算法,按每个技能的等级顺序对技能组加锁互斥访问.

通过表 1 和图 3 实验数据的对比分析发现,未经优化的互斥方法相当于是串行执行,需要的处理时间最多,算法处理性能几乎没有提升. 有序模型的全部局部资源访问模式下的并行程度和性能与自由竞争模型非常接近,性能损失很少,而部分局部资源访问模式下,性能远远高于自由竞争模型. 因此,有序竞争模型在绝大部分的情况下可以对呼叫中心路由算法的性能有明显的提升.

表 1 试验结果数据

模型	进程平均 计算时间/ms	理论计 算时间	并行程度	性能提升
普通模式	50 978	52 000	1.02	1.02
自由模式	13 347.75	52 000	3.89	3.89
有序模式——一次访问	4 100.75	16 000	3.90	12.68
有序模式一两次访问	7 929.5	28 000	3.53	6.56
有序模式一三次访问	11 388.25	40 000	3.51	4.57
有序模式一四次访问	16 161.25	52 000	3.21	3.22

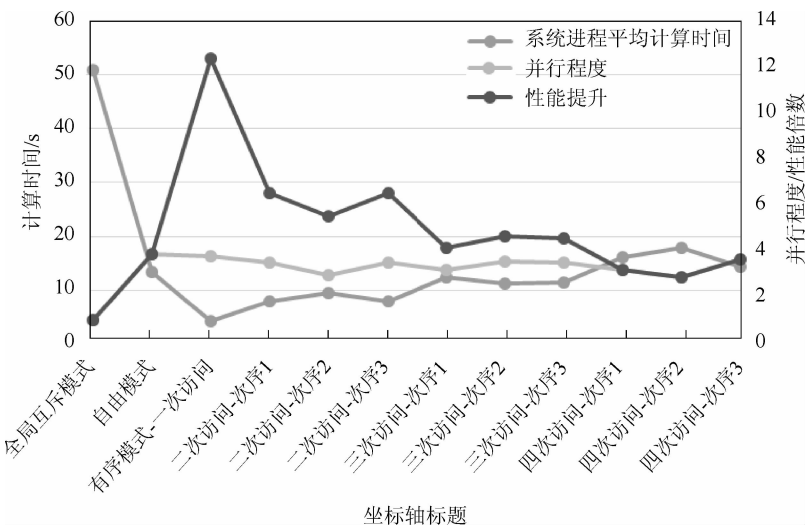


图 3 实验数据对比

6 结束语

多技能模型呼叫中心的座席资源竞争和互斥访问导致系统性能扩展性不佳是实际问题。PCCRM通过资源粒度细化和有序竞争约束,提高了资源利用率和进程并行程度。虽然引入了若干限制,但是没有违反路由算法的基本准则和多技能共享模型要求,在实际应用中不会产生限制。

参考文献:

[1] Lamport L. Time, clocks and the ordering of the events in distributed system[J]. Communications of the ACM, 1978, 21(7): 558-565.

[2] Lamport L. The part-time parliament[J]. ACM Transaction of Computer System, 1998, 16(2): 133-169.

[3] Lamport L. Paxos made simple[J]. ACM SIGACT News, 2001, 32(4): 51-58.

[4] Maekawa M. A  $\sqrt{n}$  algorithm for mutual exclusion in

decentralized systems[J]. ACM Transactions on Computer Systems, 1985, 3(2): 145-159.

[5] Ng W K, Ravishankar C V. Coterie templates-a new quorum construction method[C] // Proceedings of the 15<sup>th</sup> International Conference on Distributed Computing System. Vancouver, BC:IEEE Press, 1995: 92-99.

[6] Li Meian, Liu Xinsong, Wang Zheng, et al. A symmetric quorum algorithm for distributed mutual exclusion based on the properties of cyclic quorum [J]. Chinese Journal of Electronics, 2006, 1(1): 17-20.

[7] Baldoni R, Virgillito A, Petrassi R. A distributed mutual exclusion algorithm for moblie Ad-Hoc networks[J]. International Journal of Computer Networks & Communications, 2012, 413(1): 539.

[8] Lejeune J, Arantes L, Sopena J, et al. A fair starvation-free prioritized mutual exclusion algorithm for distributed systems[J]. Journal of Parallel & Distributed Computing, 2015(83): 13-29.