

实时 CORBA 优先级波段划分的研究及算法实现

杜庆伟, 沈卓炜, 杨 鹏

(东南大学计算机网络和信息集成教育部重点实验室, 江苏南京 210096)

摘要: 实时 CORBA 是通用 CORBA 的一个扩展, 首先对实时 CORBA 系统的传输机制进行了一定的分析并给出了相关的仿真试验, 在此基础上提出了计算实时 CORBA 系统所需波段数的试探性算法, 以及基于负载平衡的波段优先级范围划分算法, 并进一步提出了改进的加权算法. 测试结果表明算法能在兼顾到周期和优先级因素的基础上, 对实时 CORBA 系统的优先级波段连接进行合理的优先级划分.

关键词: CORBA; 实时; 优先级; 时间需求分析; 连接

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2004) 05-0854-03

Study on Priority Banded Connections in RealTime CORBA

DU Qingwei, SHEN Zhuowei, YANG Peng

(The key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing, Jiangsu 210096, China)

Abstract: Realtime CORBA specification is an extension of CORBA specification. This paper first introduces the priority banded connections in the realtime CORBA specification, then gives an algorithm to determine the amount of priority banded connections and an algorithm to compute the scope of each priority banded connection. Also an improved algorithm is given. Finally, some experiments are done to prove the algorithm's rationality.

Key words: CORBA; realtime; priority; time demand analysis; connection

1 引言

在分布实时系统中, 传输资源是一类重要的资源, 为了消除通用 CORBA 传输资源的动态管理所造成的时间不确定性, OMG 组织在实时 CORBA 规范中规定: 客户端与服务器端之间在调用发生之前, 事先建立起所需的连接, 即所谓的显式绑定(Explicit Binding) 机制. 该机制有优先级波段连接(Priority Banded Connections) 和专用连接(Private Connection) 两种方式^[1, 2]. 优先级波段连接指客户端和服务器端之间建立有多条连接, 每个连接处理不同优先级范围(R_i) 内的任务的请求; 专用连接指某个任务独占一条连接, 不与其它任务共享, 可以考虑成特殊的优先级波段连接. 优先级波段满足以下条件(E 为用户应用所涉及的所有优先级的集合):

$$\begin{cases} P_i, P_j: R_i \cap R_j = \emptyset \\ G \cap R_i = E \end{cases}$$

文献[3, 4] 对优先级波段连接机制进行了分析; 除此之外, 优先级波段连接还使得系统可以对不同的连接采用不同的 QoS, 使高优先级任务可以获得更好的服务. 文献[5, 6] 中的相关实验表明, 采用优先级波段连接机制可以为分布实时系统带来较好的实时性能. 本文给出了相关算法, 对波段数/连接数的确定, 以及波段连接的优先级范围的划分进行了探讨.

2 算法分析

使用多条连接, 可以确保网络资源满足一定数量的并发调用, 增加了实时应用的可预测性. 但是, 规范并没有给出计算连接数/波段数的方法. 如果连接过少, 传输资源可能成为系统的瓶颈, 增加实时系统的不可预测性. 而连接过多, 则会造成不必要的浪费, 系统的可扩展性不强. 图 1 是传输资源使用的一个仿真试验, 测试 100 个作业在同时运行的情况下, 等待获得连接的时间和. 可见, 连接较少时, 等待时间相当长, 必将对实时系统的可预测性造成影响. 而连接较多时, 随着连接数的增加, 取得的效果并不明显. 因此连接数的确定是一个值得研究的内容. 另外, 在给出了连接数后, 如何

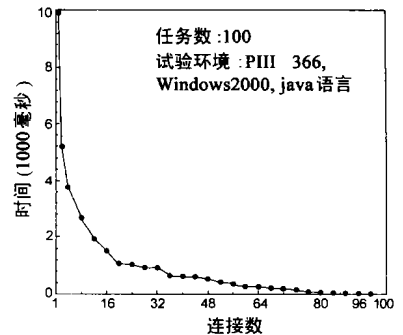


图 1 阻塞时间测试

确定波段所覆盖的优先级范围, 也是一个需要解决的问题. 本文利用 Lehczyky 提出的时间需求分析法^[7] 的思想, 提出了一

一个确定连接数的试探性算法, 和一个基于负载平衡的波段优先级范围划分算法. 算法适用于固定优先级的周期性任务集.

在文献[8]提出的端到端实时 CORBA 系统的调度模型中, 将任务分为计算子任务和传输子任务两类, 认为在前者处理时间不断减少的背景下, 后者的消耗越来越不可忽略, 需要予以考虑, 给出最坏执行时间. 本文算法继承了这个观点, 其思想是: 在假定连接数已知的情况下, 进行波段划分, 进而计算出每个波段中的每个传输子任务的时间需求, 判断是否小于最坏执行时间. 如果所有波段的所有传输子任务都符合要求, 则说明连接数适合, 波段划分成功, 否则增加连接数, 继续试探. 需要指出的是, 传输资源是按照不可抢占的方式调度的, 而时间需求分析法是用于可抢占方式调度的系统的, 因此, 其分析方法不能直接运用于传输资源的分析中. 但是, 基于时间需求分析法的思想, 对传输子任务的分析也是不困难的. 以下为了便于分析, 将传输子任务按优先级由低到高编号, 编号从 1 到 th (th 为传输子任务数), 传输子任务 i 的作业简称为作业 i . 下面分析波段内传输子任务 k 在一个周期内的一次执行(作业 k)的时间需求函数 $w_k(t)$. $w_k(t)$ 由两部分组成: 其它传输子任务对作业 k 的影响(称之为对作业 k 的影响函数 T_{inf}) 时间和作业 k 自身占用连接的时间($t_{o(k)}$). 需要考虑以下三种情况: (1) 当优先级比作业 k 低的作业和作业 k 同时申请连接, 而低优先级作业(作业 l) 恰巧先获得连接时, 对作业 k 的影响函数为:

$$T_{inf(L)} = \max_{i \in \#_L} (t_{o(i)})$$

其中, $\#_L$ 为波段内优先级比传输子任务 k 低的传输子任务的编号集(以下 $\#_x$ 的定义类似, L, E, H 分别代表 lower, equal, higher). 分析如下, 当作业 k 因申请连接而被阻塞时, 经过 $T_{inf(L)}$ 时间后, 作业 l 必然释放连接, 此时即使作业 k 不能获得连接, 也不能调度比它优先级低的作业; (2) 当优先级与作业 k 相同的作业和作业 k 同时申请连接, 同优先级作业之间按照先来先服务的原则, 而作业 k 被排在相同优先级作业的最后时, 对作业 k 的影响函数为:

$$T_{inf(E)} = \sum_{i \in \#_E} t_{o(i)}$$

(3) 当优先级比作业 k 高的作业和作业 k 同时申请连接时, 对作业 k 的影响函数为:

$$T_{inf(H)} = \sum_{i \in \#_H} (7 t_k / p_i \hat{o} @ t_{o(i)})$$

其中, p_i 为传输子任务 i 的周期, t_k 是作业 k 从申请到获得连接的时间. 该影响函数可以使用迭代的思想算出:

$$T_{inf(H)}^{(l+1)} = T_{inf(H)}^{(l)} + \sum_{i \in \#_H} (\hat{o} T_{inf(H)}^{(l)} / p_i \hat{o} @ t_{o(i)}) \quad (1)$$

因为作业 k 的时间需求总要包括比它优先级高的所有传输子任务执行一次的时间, 所以设 $T_{inf(H)}^{(l)}$ 为 $\sum_{i \in \#_H} t_{o(i)}$. 上面的计算过程在 $T_{inf(H)}^{(l+1)}$ 等于 $T_{inf(H)}^{(l)}$ 且 $T_{inf(H)}^{(l)}$ [p_k , 或者 $T_{inf(H)}^{(l+1)}$ 开始大于 p_k 的两个时刻中较早的一个时刻结束, 后者表示计算失败.

最后可得作业 k 的时间需求函数为:

$$w_k(t) = T_{inf(L)} + T_{inf(E)} + T_{inf(H)} + t_{o(k)} \quad (2)$$

3 算法实现

首先给出一个基于负载平衡的、计算波段优先级范围的算法(算法 1)(其中, T 是任意一个大数, 用以标识一个时间段; $M = \sum_{i=1}^{th} (7 T / p_i \hat{o} @ t_{o(i)})$, 是 T 时间内所有释放的作业占用连接的时间和, 因为 T 是一个大数, 可以简单的令 $M = \sum_{i=1}^{th} (T / p_i @ t_{o(i)})$; c 为连接数/波段数; $T_s = M / c$ 是每个连接应处理的工作量):

```

int i = 1, m = 1;
while(i <= th) {
    range.begin[m] = i; // 设置波段的起始优先级
    Mc = 0;
    for(j = i; j <= th && Mc <= T_s + A; j++)
        Mc = Mc + 7 T / p_j \hat{o} @ t_{o(j)}; // 或 Mc = Mc + \frac{T}{p_j} @ t_{o(j)}
    if(j > i && Mc > T_s + A)
        range.end[m] = j - 1; // 设置波段的终止优先级
    else
        range.end[m] = j; // 设置波段的终止优先级
    i = range.end[m] + 1;
    m++;
    if((m > c) && (i <= th)) // 连接已经分配完, 但传输子任务未被全部覆盖
        return error;
}
return ok;
    
```

算法 1 的思想是使每条连接在 T 时间内处理的工作量相同, 但实际上, 并不能够利用 T_s 来精确划分连接的工作量,

$$P \ m, \ v \ n, \ x = \sum_{i=m}^n (T / p_i) @ t_{o(i)},$$

$$y = \sum_{i=m}^{n-1} (T / p_i) @ t_{o(i)}; (1F \ mF \ nF \ th) C(x \ E \ T_s) C(y) < T_s$$

因此在算法中增加了 A 调整因子来设置一个范围. 为防止再加上一个传输子任务后, 连接的负载过大, A 不应太大, 令 A 为 T 时间内传输子任务占用连接的平均值:

$$A = M / th$$

如果某次计算失败, 可以上调 A (幅度为 M / th), 重新计算.

上述算法可以获得一种基于负载平衡的波段优先级范围的划分方法, 下面给出一个计算连接数的试探性算法(算法 2)(其中, e_k 为传输子任务 k 的最坏执行时间):

```

for(i = c.begin; i < th; i++) {
    利用算法(1)计算出 i 个波段的优先级范围;
    for(每一个作业 k) {
        利用公式(2)计算出作业 k 的时间需求 w_k(t);
        if((w_k(t) > e_k) + (利用公式(1)时计算失败))
            break; // i 条连接不能满足要求
    }
    if(所有作业验证完毕)
        break; // 成功
}
连接数为 i;
    
```

其中 $c_{begin} = \max(1, 7M / T\hat{o})$, 是 T 时间内全部作业进行传输所需要的最小连接数.

4 算法测试及改进

图 2 为波段划分的一个测试结果, 共 400 个传输子任务, 优先级从 1 到 400, 占用连接时间相同, 均为 t_0 , 周期为 $p_i = 50t_0 + 25t_0 @ 400 - i / 208$. 测试中, $c_{begin} = 3$.

图 2 表明, 随着任务发送频率的增加, 波段划分越来越细, 即共享一条连接的传输子任务越来越少, 传输子任务获得的服务也越来越好. 因此, 这个算法能对速率单调(rate monotonic) 优先级分配法提供较好的支持. 另外, 大多数作业的时间需求在波段内呈线性递减, 这说明相应的作业在被阻塞的时间内, 波段内不存在高优先级传输子任务执行多次的情况.

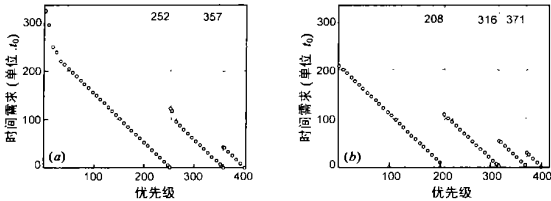


图 2 波段划分后, 作业的时间需求 (a) 3 条连接; (b) 4 条连接

如果实时系统不是采用速率单调优先级分配法, 则可以对上述算法进行改进, 通过加权, 使得低优先级波段的连接在共享程度上比高优先级波段的连接要高, 进而使得优先级高的传输子任务获得较好的服务. 令算法(1)中

$$M = \sum_{i=1}^{th} (7(B_i @ T) / p_i @ t_{0(i)}), \text{ 或 } M = \sum_{i=1}^{th} ((B @ T) / p_i @ t_{0(i)})$$

以及 $M_c = M_c + 7(B @ T) / p_i @ t_{0(i)}$, 或 $M_c = M_c + (B @ T) / p_i @ t_{0(i)}$ 即可. 其中 $B(B \setminus 1)$ 为加权因子, 是传输子任务 i 的优先级的函数, 优先级越高, B_i 越大. 一个简单例子是 $B = P_i$ (P_k 为优先级数, 对应于优先级数越大, 优先级越高的情况, 以下测试采用该函数). 这样, 优先级高的传输子任务将通过加权因子得到调度上的倾斜, 从而获得较大的平均带宽, 保证高优先级传输子任务能顺利执行.

图 3 为带有加权因子的波段划分的测试, 除了周期固定为 $300t_0$ 外, 其它参数同图 2 的测试. 图 3 表明, 虽然周期相同, 但随着优先级的增加, 波段划分越来越细, 因此, 加权的算法在算法 1 的基础上, 进一步考虑了优先级的因素, 使得高优先级的传输子任务可以获得更好的服务.

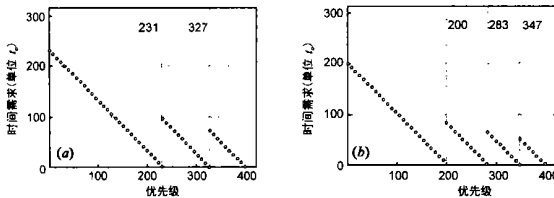


图 3 波段划分后, 作业的时间需求 (带加权因子) (a) 3 条连接; (b) 4 条连接

5 结束语

本文对实时 CORBA 系统的传输机制进行了分析, 提出了

计算连接数和波段优先级范围划分的相关算法, 希望给分布实时系统的开发人员提供一个较为合理的依据, 使之更加科学规范. 应该指出, 采用试探性算法开销较大, 考虑到这部分工作是在正常的远程调用之前完成的, 因此对系统正常情况下的实时性能没有直接的影响. 另外, 本文对优先级加权因子研究不足, 这也限制了对不同应用系统的波段优先级范围进行划分的合理性.

参考文献:

- [1] omg ptc/9206202: RealTime CORBA 1.0[S]. OMG Adopted Specification, www.omg.org.
- [2] omg, White Paper on Realtime CORBA Initial Review Draft[R]. OMG Realtime Platform SIG, December 5, 1996.
- [3] Douglas C Schmidt, Fred Kuhns. An Overview of the realtime CORBA specification[J]. IEEE Computer, 2000, 33(6): 56- 63.
- [4] Carlos O. Ryan, Douglas C Schmidt, Fred Kuhns. Evaluating policies and mechanisms for supporting embedded, realtime applications with CORBA 3.0[A]. Proc. of RTAS Sixth IEEE RealTime Technology and Applications Symposium[C]. IEEE 2000, Washington, D. C., 2000. 188- 197.
- [5] Stefan Lankes, Michael Pfeiffer, Thomas Bemerl. Design and implementation of a SC2based realtime CORBA [A]. Fourth International Symposium on ObjectOriented RealTime Distributed Computing[C]. Magdeburg, 2001.
- [6] S Lankes, A Jabs, T Bemerl. Integration of a CAN2based connection2 oriented communication model into realtime CORBA[A]. Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS 2003), 11th Workshop on Parallel and Distributed RealTime Systems (WPDRTS 2003)[C]. Nice, France, 2003.
- [7] Lechoczy, J L Sha, Y Ding. The rate2monotonic scheduling algorithm: Exact characterization and average case behavior[A]. Proceedings of 10th IEEE RealTime Systems Symposium[C]. Los Alamitos, CA: IEEE Computer Society Press, 1989. 166- 171.
- [8] 谢俊清. 实时 CORBA 系统调度模型及其实现结构的研究[D]. 南京: 东南大学计算机系, 2000.

作者简介:



杜庆伟 男, 1974 年 3 月生于河南郑州, 博士生, 主要研究方向为分布对象计算. Email: duqingwei@seu.edu.cn.



沈卓炜 男, 1974 年 1 月生于江苏常州, 博士生, 主要研究方向为分布对象计算.

杨 鹏 男, 1975 年生于四川成都, 主要研究方向为分布对象计算.