

具有不同类服务器的 C/S 系统的优化设计¹

刘 波 李益强 李 华*

(西安电子科技大学 西安 710071)

*(西北工业大学 西安 710072)

摘 要 该文利用系统分析的方法研究客户 / 服务器系统 (简称 C/S 系统) 的优化设计。首先用排队论分析具有不同类服务器的 C/S 系统, 得到系统性能的解析解。在此基础上建立了系统优化设计的数学模型, 给出了不同条件下的优化算法及优化实例。该方法对于分析系统瓶颈、优化系统的体系结构有重要意义。

关键词 不同类服务器, 客户 / 服务器, 系统分析

中图分类号 TP39, O224

1 引 言

客户 / 服务器系统 (简称 C/S 系统) 是当今分布式环境中计算机应用发展的主流, 其产生和应用是网络技术和计算机应用发展的结果, 反映了独立的自治进程之间的关系。服务器提供服务, 客户则使用服务。在地理位置上, 两者可以位于同一计算机上, 也可以通过网络分布到不同的计算机上。C/S 系统以消息作为载体完成服务请求和结果应答信息的传递, 具有较好的用户透明性, 服务的封闭性以及易于升级, 资源共享等特点。一般在实际应用中, 客户主要完成数据检验和其他一些简单的分析、计算、表达、显示等工作, 而服务器则要完成大量的处理任务, 是系统的核心, 其效率和可靠性对整个系统的性能有很大的影响。

目前, 有关 C/S 系统的研究主要集中在体系结构和应用编程上^[1], 而对系统的分析工作做的很少。文献 [2] 研究了具有同类服务器的 C/S 系统配置优化的一般算法, 文献 [3] 研究了局域网环境下, 同类服务器的 C/S 系统优化配置的遗传算法。但是上述研究仍有以下局限性:

(1) 都是针对同类服务器的 C/S 系统; (2) 优化算法中, 同时只能为一项配置作最优设计。本文将用系统分析法分析在不同类服务器情况下系统的稳态性能, 并用降维逐次逼近算法实现同时为多项配置作出最优设计。

2 系统描述

考虑如图 1 所示的系统, 它由多个客户机, 多个不同类的服务器及相关的通信设备组成。客户机产生任务请求并通过网络传给任务分发服务器, 该服务器把任务分发给指定的空闲服务器。若服务器都忙, 则把任务暂存缓冲区, 直到有空闲的服务器为止 (服务器 # ij 表示该服务器属于第 i 类服务器中的第 j 台)。共有 M 类服务器, 其中第 i 类服务器的台数为 $h_i (i = 1, 2, \dots, M)$, 为便于分析, 作以下假设:

(1) N 个客户机独立地、随机地提出任务请求, 且每个客户提出请求的时间间隔 T 服从负指数分布, 即 $P(T \geq t) = e^{-\lambda t}$ 。

(2) $\sum_{i=1}^M h_i$ 个服务器独立工作, 每个任务只通过一个服务器即算完成, 第 i 类服务器对每个任务的服务时间是随机的, 且服从负指数分布 $P(T \geq t) = e^{-\mu_i t}$ 。

¹ 2001-05-23 收到, 2001-10-10 定稿

国家自然科学基金资助项目 (50075070); 中国博士后基金资助项目

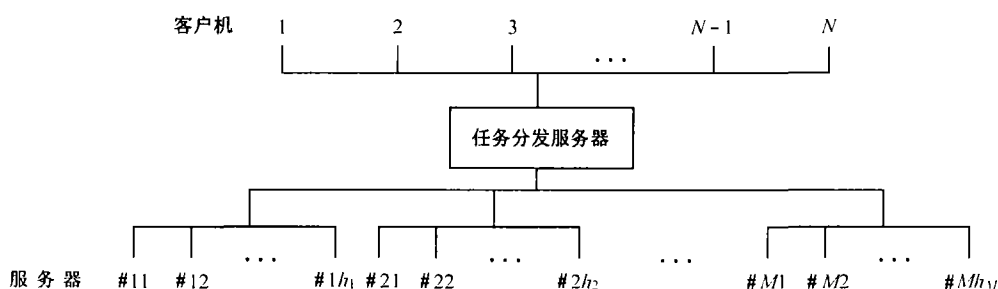


图 1 具有不同类服务器的 C/S 系统

(3) 所有的任务都按先来先服务的规则排队。

(4) 客户机随机提出的任务请求是要求第 i 类服务器服务的概率为 $q_i (i = 1, 2, \dots, M)$ ，显然 $\sum_{i=1}^M q_i = 1$ 。

3 系统性能分析

在上述的 C/S 系统中，由假设可知，每个客户进程每 $1/\lambda$ min 请求一次服务，这次请求恰好是请求第 i 类服务器的概率为 q_i 。因此，每个客户进程每 $[1/(\lambda q_i)]$ min 请求第 i 类服务器一次，而第 i 类服务器进程平均每处理一个请求需花费 $(1/\mu_i)$ min。可以证明， N 个客户机与第 i 类服务器构成一个有限源的生灭排队系统^[4]。这样，整个系统便分成 M 个子系统，且在第 i 个子系统中，每个客户提出请求的时间间隔 T 服从负指数分布，即 $P(T \geq t) = e^{-\lambda t q_i}$ 。

对于第 i 个子系统，系统参数为

$$\lambda_j = (N - j)\lambda q_i, \quad j = 0, 1, 2, \dots, N - 1$$

$$\mu_j = \begin{cases} j\mu_i, & j = 1, 2, \dots, h_i - 1 \\ h_i\mu_i, & j = h_i, h_i + 1, \dots, N \end{cases}$$

系统状态瞬时强度转移图如图 2 所示：

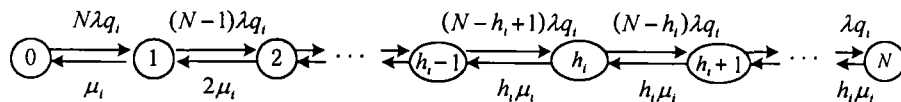


图 2 系统状态瞬时强度转移图

相应的平衡方程为 (其中 $(p_j (j \in \{0, 1, 2, \dots, N\}))$ 是系统的平稳分布)：

$$\begin{aligned} N\lambda q_i p_0 &= \mu_i p_1 \\ ((N-1)\lambda q_i + \mu_i) p_1 &= N\lambda q_i p_0 + 2\mu_i p_2 \\ &\vdots \\ \lambda q_i p_{N-1} &= h_i \mu_i p_N \end{aligned}$$

且

$$\sum_{i=0}^N p_i = 1 \quad (1)$$

于是

$$p_j = \begin{cases} C_N^j \left(\frac{\lambda q_i}{\mu_i}\right)^j p_0, & j = 1, 2, \dots, h_i - 1 \\ C_N^j \frac{j!}{h_i! h_i^{j-h_i}} \left(\frac{\lambda q_i}{\mu_i}\right)^j p_0, & j = h_i, h_i + 1, \dots, N \end{cases}$$

其中

$$p_0 = \left[\sum_{j=0}^{h_i-1} C_N^j \left(\frac{\lambda q_i}{\mu_i}\right)^j + \sum_{j=h_i}^N C_N^j \frac{j!}{h_i! h_i^{j-h_i}} \left(\frac{\lambda q_i}{\mu_i}\right)^j \right]^{-1}$$

在此基础上, 可求得系统性能的解析解如下:

(1) 在第 i 个子系统中, 处于等待状态的客户机数

$$LQ_i = \sum_{j=h_i}^N (j - h_i) p_i \quad (2)$$

$$\bar{\lambda}_i = \lambda_i \sum_{j=0}^N (N - j) p_j = \lambda q_i (N - \sum_{j=0}^N j p_j)$$

(2) 每个客户机任务通过第 i 个服务系统时在队列中的平均等待时间

$$WQ_i = \frac{LQ_i}{\bar{\lambda}_i} \quad (3)$$

(3) 系统中第 i 类服务器的平均利用率

$$QS_i = \sum_{j=0}^{h_i-1} \frac{j}{h_i} p_j + \sum_{j=h_i}^N p_j = 1 - \sum_{j=0}^{h_i-1} \frac{h_i - j}{h_i} p_j \quad (4)$$

下面分析客户进程数 N , 服务器数 M 以及服务器的服务能力参数 $\rho = \lambda/\mu$ (即客户进程请求的到达率和服务器的服务率之间的比值) 变化时的系统相对队长和服务器利用率的变化趋势。

分析解析表达式 (2) 式, (3) 式和 (4) 式, 可得如下结果:

当 N 增加时, 相对队长和服务器的利用率都逐渐增加, 而且随着 N 值的不断增大, 会呈现出加快的趋势。

当保持 N 不变, 每增加一个服务器, 相对队长减小, 因此极大地提高了系统的性能, 但同时也降低了服务器的平均利用率。

当服务器的服务能力参数 $\rho = \lambda/\mu$ 增加时, 相对队长加长, 使得系统可能出现阻塞情况, 但服务器的平均利用率提高。

一般地, 当系统出现阻塞时, 可以采取的措施有: (1) 减少正在工作的客户进程数, 如关闭部分客户进程; (2) 降低客户进程请求的频度, 以减少客户进程请求的到达率 λ ; (3) 增加服务器的数量; (4) 提高服务器的性能, 如更新设备、修改软件等, 以此缩短每个客户进程的服务时间 $1/\mu$ 。因此在实际应用中应根据客户机、服务器等多方面的实际情况进行分析和权衡, 选择一个合理的配置, 使得系统性能达到最优。

4 优化配置的降维逐次逼近算法

4.1 目标函数

为了更大限度地利用系统资源, 提高服务的质量, 减少客户的等待时间, 需建立系统优化配置的目标函数。在各种系统指标中, 用户较关心的有: (1) 服务器的平均利用率; (2) 系统中的排队队长; (3) 每个客户在队列中的排队时间。由 (3) 式知: (2) 和 (3) 有相同的变化趋势, 只是变化率有所差别, 所以目标函数中只需包括 (2)。但目标 (1) 和 (2) 却是相互矛盾的, 比较 (1)、(2) 两个指标, 它们的量纲各不相同, 因此需进行归一化处理。对排队队长而言, 排队队长的绝对值在客户机数量变化时没有可比性, 而排队队长占系统中所有客户机数量的比例具有可比性, 因此得到归一化的性能指标向量 $(QS_i, LQ_i/N)^T$, 服务器的效益可以认为与 QS_i 成正比, 而客户机的效益则不能看作与排队的客户比例均匀变化, 因为其中需要考虑系统用户的承受能力。因此, 由线性加权和法定义本系统的目标函数为

$$\begin{aligned} \text{Max } z &= \sum_{i=1}^M (W_i QS_i + W_0 f(\frac{LQ_i}{N})) \\ \text{s.t. } W_0 + \sum_{i=1}^M W_i &= 1, \quad W_i \geq 0, \quad i = 0, 1, \dots, M \end{aligned} \quad (5)$$

其中 $f(\cdot) \in [0, 1]$, $f(0) = 1$, $f(1) = 0$ 且 f 为单调下降函数, 函数 f 的其它值由系统的使用者根据实际情况和偏好来定义。

特别地, 当 $M = 1$ 时, (5) 式演化为

$$\begin{aligned} \text{Max } z &= W_1 QS + W_0 f(\frac{LQ}{N}) \\ \text{s.t. } W_1 + W_0 &= 1, \quad W_0, W_1 \geq 0 \end{aligned} \quad (6)$$

这便是具有同类服务器的 C/S 系统的优化配置目标函数。

4.2 降维逐次逼近算法

由目标函数 (5) 式易证其最大值存在。

从模型的决策变量可以看出, 在实际的问题中目标解集是 $(N, h_1, h_2, \dots, h_M)$ 或 $(\lambda, \mu_1, \mu_2, \dots, \mu_M)$, 它们都是 $M+1$ 维的向量, 这是一个多目标的优化问题。

降维逐次逼近法的基本思想就是首先降低目标解集的维数, 然后采用最小摄动搜索方法逐步优化各个目标变量, 最后得到目标函数的最优解。

在实际应用中, 我们通常是在已知 $(\lambda, \mu_1, \mu_2, \dots, \mu_M)$ 的条件下, 优化 C/S 系统中 $(N, h_1, h_2, \dots, h_M)$ 的配置。其算法步骤如下:

步骤 1 确定用户偏好函数, 输入已知量;

步骤 2 设定 N 的初始值, 并确定 N 的搜索步长, 设为 ε ;

- 步骤 3 设定 h_j 的初始值, 并且对每一目标变量 h_j 确定一个整数摄动步长 δ , 定义向量 δ_j , 它的第 j 个分量为 δ , 其余的分量全为零 ($j = 1, 2, \dots, M$);
- 步骤 4 计算目标函数的值, 记为 Z_{temp} ;
- 步骤 5 设定 j 的初始值, 令 $j = 1$;
- 步骤 6 给服务器配置向量 $(h_1, h_2, \dots, h_M)$ 加上向量 δ_j , 再计算目标函数的值 Z_j , 若 $Z_j < Z_{\text{temp}}$, 则令 $j = j + 1$ 转到步骤 6; 若 $Z_j \geq Z_{\text{temp}}$, 令 $Z_{\text{temp}} = Z_j$ 继续步骤 6; 这样一直到 $j = M$ 为止, 最后利用跳出此循环的约束条件找到 $(h_1, h_2, \dots, h_M)$ 的最优值;
- 步骤 7 $N = N + \varepsilon$, 转到步骤 4 继续; 利用跳出此循环的约束条件找到 N 的最优值;
- 步骤 8 输出目标函数的最优解 $(N, h_1, h_2, \dots, h_M)$; 结束。

一般来说, 降维逐次逼近算法由于受到初值的影响, 计算的结果不一定收敛到最优解。但是在本题的计算中, 由于有实际的意义, 所有的目标变量都从最小的正整数开始, 所以只要目标函数的最优解存在, 此算法必定能收敛到最优解。

4.3 计算实例

一个具有两类不同服务器的系统, 设 $\lambda = 0.2/\text{min}$, $\mu_1 = 3/\text{min}$, $\mu_2 = 2/\text{min}$, $q_1 = 0.2$, $q_2 = 0.8$, $W_0 = W_1 = W_2$, 函数 $f(\cdot)$ 如表 1 所示, 未说明的参数采用四舍五入的办法进行处理。

按上述算法我们设计了程序, 求得最佳的客户机数为 151, 第一类服务器的最佳个数为 2, 第二类服务器的最佳个数为 11。

顺便指出, 用本文的算法计算文献 [2] 中的算例, 求得客户进程最优数量为 27。而文献 [2] 所得的客户进程最优数量为 28 的结果是错误的。因为根据目标函数的定义, 在 $N = 27$ 时目标函数才到达最大值。

表 1 函数 $f(x)$ 的定义

x	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
$f(x)$	1.0	1.0	0.99	0.98	0.97	0.95	0.93	0.90	0.85
x	0.09	0.10	0.11	0.12	0.13	0.14	0.15	0.16 - 1.0	
$f(x)$	0.80	0.70	0.60	0.40	0.25	0.10	0.00	0.00	

5 结 束 语

本文对具有不同类服务器的 C/S 系统进行了系统分析, 建立了系统的最优配置模型, 并且提出了一种有效的算法, 实现了系统的优化设计。本文的研究工作使得我们可以更加方便、直观地了解 C/S 系统的动态行为, 并且实现系统的优化, 因此有较强的实际意义。

同时, 我们还将进一步研究用 FUZZY-AHP 法对系统的性能优化程度进行综合评价, 以此指导系统的设计决策。

参 考 文 献

[1] Satoshi Moriguchi, Performance and Performability Evaluation of Client Server System by Petri Nets, Transaction A of IFIP, 1992, 622-628.

[2] 金尊和, 吴 澄, 具有多服务器的客户 / 服务器系统的配置优化算法, 系统工程理论与实践, 1996, Vol.16, 27-31.

[3] 戴晓晖, 李敏强, 寇纪淦, 客户 / 服务器系统配置优化的遗传算法, 系统工程, 1999, 17(1), 57-63.

- [4] 孟玉珂, 排队论基础及应用, 上海, 同济大学出版社, 1989, 130-162.

OPTIMAL DESIGN FOR CLIENT/SERVER SYSTEM WITH MULTIPLE TYPE SERVERS

Liu Bo Li Yijiang Li Hua*

(Xidian University, Xi'an 710071, China)

**(Northwestern Polytechnical University, Xi'an 710071, China)*

Abstract Optimal design for C/S system is studied using system analysis method. Firstly, C/S system with multiple type servers based on queuing theory and analytic performance results for the system are presented. A mathematical model for system optimal design and relative algorithm under different conditions are then obtained. Finally, an example is followed to illustrate the computing process. This method is important to analyze the bottleneck of C/S system and optimize the system programming.

Key words Multiple type servers, Client/Server (C/S), System analysis

刘 波: 男, 1957 年生, 研究员, 研究方向: 计算机应用.

李益强: 男, 1978 年生, 硕士, 研究方向: 系统工程, 决策分析.

李 华: 男, 1963 年生, 副教授, 研究方向: 系统工程, 工业工程.