计算机科学1999Vol.26№.9

事务处理系统的性能模型

Performance Model of Transaction Processing System

孙乐昌　胡劲松

（中国人民解放军电子工程学院　合肥230037）

Abstract　This paper mainly presents the modeling method for performance of transaction processing system, and discusses how to analyse its performance using the performance model.

Keywords　Transaction processing system, Performance model, Transaction volume, Response time

事务处理(TP)系统为实时多用户系统，其性能(即响应时间)受系统瓶颈—并发事务的共享资源的制约。评价、分析和改善事务处理系统的性能，需要建立事务处理系统的性能模型，本文就性能模型的建立方法加以论述。

1.　性能模型

可以用一个性能模型来描述通过系统的主要事务的事务流程。标识每个处理步骤，突出表示那些可能引起事务处理延迟的结点。利用性能模型可以为事务处理所需时间建立起一个表达式。这一处理时间是指用户所能忍受的响应时间。在响应时间变得不可接受之前，这段时间的系统调用称为系统的处理容量。性能建模为的是要预测系统处理容量。

2.　一个简单的事务处理系统的性能模型

2.1　一个简单的事务处理系统

假设有一个简单的请求-服务事务处理系统如图1所示。事务请求由一个请求服务进程来接收，由两个服务进程来处理，其中：查询服务进程只处理查询事务，而更新服务进程只处理更新事务。每个服务进程都直接向数据库管理进程发送存取数据库的请求，即查询服务进程将请求读取数据库的记录，而更新服务进程将请求对数据库的读写操作以更新数据库的记录。又假设服务进程都是单线程的，即一次只能处理一个事务项。

每个服务进程在完成对事务的处理后都要归纳整理出对用户的应答，并通过应答处理进程返回给用户。查询事务的应答包含用户请求的数据；更新事务则包含完成更新后的状态。

通信信息量的大小、服务进程所需要的处理时间以及事务类型的概率分布情况如表1所示。

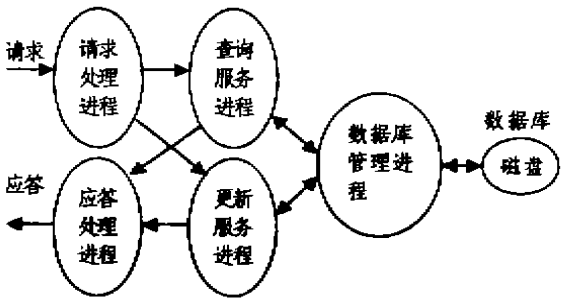

图1　一个简单的TP系统

表1

事务	概率	信息量大小(bytes)		磁盘存取	处理时间(ms)
		请求	应答		
查询	35%	20	400	1	10
更新	65%	200	15	2	15

再假设所有用户都通过9600波特率的点到点异步传输线路与系统连接。所有的通信I/O操作全部由请求/应答进程处理，而由数据库管理进程进行磁盘的存取；系统中没有其它的I/O处理进程；且所有进程的优先级相同，在此忽略中断处理过程。系统的事务流程如图2所示。

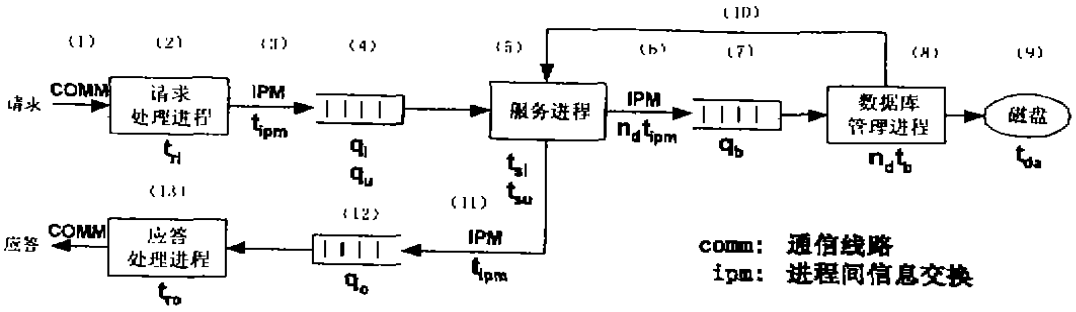

图2 系统的事务处理流程

2.2 性能建模

现在我们可以利用性能模型来导出系统响应时间关于系统调用的函数关系。在单处理器系统中各进程的有效处理时间难以直接求得，因为它们必须竞争使用一个共享的处理器。

一般来说，共享资源上的时间开销由两部分组成：在该资源队列中的等待时间和实际的资源占有时间。它们分别被称为等待时间和占有时间。这两部分时间的总和称为延迟时间，也就是事务由于请求资源造成的延迟。延迟时间作为资源的一种重要属性，其计算公式如下：

$$Delay = T/(1-L) \qquad (1)$$

其中 T 表示资源的平均占有时间，L 表示资源的调用率，也称为资源的占用率。术语 $1/(1-L)$ 可以认为是一个"扩展因子"，它随着资源调用的增加而对资源上的时间开销进行扩展，显然已考虑到了在队列中等待的时间。事实上，当资源调用趋近与100%（L=1），资源的队列可能非常长。性能模型所需的各项参数如表2所示。

对于每一个事务，在如表3所示的情况下需要获取处理器，性能模型的各项结果如表4所示。

2.3 模型分析

系统的平均响应时间 T 是事务的处理速率 R 的函数。将上述模型中的各项参数代入，则得到 T 和 R 在该系统中的函数关系，如图3所示，注意到图中的曲线突然变陡。这通常是由系统中的"潜在危机"造成的：在系统调用率低时，产生的延迟很小；在高系统调用率下，就转化为系统的瓶颈。事实上，随着系统调用的增加，系统可能会突然"瘫痪"。

如果假设示例系统每秒处理的事务容量为4项事务（总容量的67%），则系统的响应时间大约为0.6秒，系统运作相当理想。注意：如果我们将系统每秒处理的事务容量提高25%到5项事务，系统响应时间

表2 性能模型参数

参数类型	参数	说明
结论参数	T	系统平均响应时间（秒）
输入变量	R	事务的处理速率（事务/每秒）
输入参数	m_{ql}	查询请求信息的量（比特）
	m_{qo}	查询应答信息的量（比特）
	m_{ul}	更新请求信息的量（比特）
	m_{uu}	更新应答信息的量（比特）
	p_l	查询事务的概率
	p_u	更新事务的概率
	s	通信线路的传输速度（比特位/每秒）
	t_b	数据库管理进程处理每个磁盘存取请求的平均时间（秒）
	t_{da}	磁盘存取所需的平均时间
	t_{ipm}	进程间信息交换时间（秒）
	t_{rl}	请求处理进程处理请求的平均时间（秒）
	t_{ro}	应答处理进程处理应答的平均时间（秒）
	t_{sl}	查询进程处理每项查询事务的平均时间（秒）
	t_{su}	更新进程处理每项更新事务的平均时间（秒）
中间参数	L_b	数据库管理进程及磁盘存取调用率
	L_o	应答处理进程的调用率
	L_p	处理器的调用率
	L_{sl}	对查询服务进程的调用率
	L_{su}	对更新服务进程的调用率
	r_d	处理器的处理速率（进程/每秒）
	t_d	获取处理器的平均时间（秒）
	t_p	处理器的平均处理时间（秒）
	t_{ql}	查询服务进程的响应时间（排队和处理时间）（秒）
	t_{qu}	更新服务进程的响应时间（秒）
	T_b	数据库管理进程的响应时间（秒）
	T_c	通信线路的响应时间（秒）
	T_l	请求处理进程的响应时间（秒）
	T_o	应答处理进程的响应时间（秒）
	T_s	服务进程的响应时间（秒）

$$\sum_{T_{v,o}\in H_{i,j}}\left\lceil\frac{t+R_{v,o-1}}{p_v}\right\rceil\tau_{v,r}\Bigg\}$$

(b)计算第 m 个实例的 IEER 时间：

$R_{i,j}(m)=C_{i,j}(m)+R_{i,j-1}-(m-1)p_i$

4)计算最大 IEER 时间：$R_{i,j}=\max\{R_{i,j}(m)\}$，$1\leqslant m\leqslant M_{i,j}$

5)最后一个子任务的 IEER 时间就是端端任务 $T_{i,j}$的最大响应时间 EER。

当任务响应时间小于端端时限时，该任务可调度。如果系统中所有端端任务都是可调度的，则该系统是可调度系统。否则需重新划分子任务或分配优先级或增加处理机计算能力，直到满足要求。

三、展望

本文对并行与分布式实时系统的调度理论进行了研究。目前单机系统的实时调度理论比较成熟，已经提出静态和动态的最优调度算法，并得到了广泛应用。原则上，多机和分布式实时系统的调度是NP难的，所以本文的研究主要集中在静态、固定优先级的独立或具有简单约束关系的实时任务集调度，今后可以进一步深入的领域有：

1)优先级分配：可利用当前一些先进的组合优化问题的解决思想、算法开发有效的优先级分配算法，减少系统的响应时间。

2)增加任务同步模型的复杂性：例如，一个子任务可以具有两个或多个前趋子任务；一个子任务的完成也可以释放两个或多个子任务；多个具有前趋关系的相邻或不相邻子任务可以在同一台处理机上执行等等。

3)各种实时网络的可调度性：包括共享介质网络和点到点网络，并与主机的调度有机结合起来进行研究。调度的粒度可以从消息减小到报文，使分析更加准确。

4)动态调度：包括优先级的动态分配。研究基于EDF(earliest deadline first，最早期限优先)调度理论的并行与分布式实时系统的调度。这对于硬实时系统来说很困难，需要限制系统模型的复杂度。

5)针对具体的系统模型研究调度算法：面向实际应用，具体问题具体分析。

主要参考文献

1 Tindell K, et al. Allocating real-time tasks An NP-hard problem made easy, Real-Time Systems Journal, 1992, 4(2)

2 Garcia J, Harbour M. Optimized priority assignment for tasks and messages in distributed hard real-time systems. In. The Third Workshop on Parallel and Distributed Real-Time Systems April 1995. 124～132

3 Dhall S K, Liu C L. On a Real-time Scheduling Problem. in Operations Research, 1978, 26: 127～140

4 Liu C L, Layland J. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment J. Assoc. Comput. Machinery, 1973, 10(1): 174～189

5 Lehoczky J, Sha L, Ding Y. The Rate Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior. IEEE Real-Time Symposium 1989. 166～171

6 Lehoczky J. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In. 11th IEEE Real-Time Systems Symposium. 1990. 201～209

7 Stankovic J A, Spuri M. Implications of Classical Scheduling Results For Real-Time Systems, June 23, 1994

8 Sun J, Liu J W S. Synchronization protocols in distributed real-time systems. In. The 16th Intl. Conf. on Distributed Computing Systems. Hong Kong, May 1996

(上接第89页)

模型可被用来评估所要做的变动对系统性能产生的影响，如果所要做的变动完全是为了增强系统的性能，那么所做变动的损/益比将可以精确地确定。

·**系统配置**。作为面向市场的产品，系统经常要包含许多选项，使用户能够根据自己的需要来配置系统，比如，磁盘的数量、处理器的能力、通信线路的传输速率。性能模型可以作为一种工具，帮助用户配置新系统。

参考文献

1 Tanenbaum A S. Distributed Operating System. Prentice Hall Intl., Inc., 1996

2 汤子瀛，哲凤群，汤小丹. 计算机操作系统. 西安电子科技大学出版社，1998

3 Highlevman W H. Performance Analysis of Transaction Processing System. Prentice Hall, Englewood Cliffs, New Jersey, 1992