计算机科学1999Vol. 26№. 7

基于 Web 服务器的性能模型与参数分析*)

Performance Model and Parameter Analysis of Web Server

喻 莉 石冰心

(华中理工大学电子与信息工程系 武汉 430074)

Abstract In this paper, a novel performance analysis model for Web server system is presented, this model is also extended to multiple-server system. Simulation results confirmed the validity of the model. After analysis the effects of the model parameters on the performance, the method to avoid deadlock and the speed rule to add new server are investigated. Finally, several schemes for improving Web server performance based on different congest conditions are evaluated and compared. It has practical meaning for network managing, planning, design and upgrade.

Key words Web server, Model, Markov queue network, Performance analysis

Web 是全球范围的信息浏览系统，建立在"客户机/服务器"模型之上。Web 服务器具有高度的集成性，能把各种类型的信息(如文本、图像、声音、动画等)和服务(如 News, FTP, Gopher, Mail 等)无缝连接起来。因此，如何使 Web 服务器服务速度更快、服务质量更好已经成为人们关注的一个问题。过去有一些关于客户/服务器系统的研究，但往往集中在容错或存储特性的研究上，很少考虑 Web 服务器系统的性能特征。文[1]只是在一些实验的基础上，提出了一个简单的关于 Web 服务器模型的想法，考虑因素少，且未做验证。

本文提出了一个新的 Web 服务器系统的性能分析模型，将 Web 服务器与通信网络一起看作一个服务系统。并将模型引伸到了多服务器系统，在此模型上对增加新服务器对系统服务性能的影响进行了理论上的证明和分析。通过仿真实验，验证了该模型的有效性。基于该分析模型，得到了一些有意义的结论，这些结论有益于指导人们进行网络规划、升级和管理。

1 Web 服务系统的马尔可夫排队模型

1.1 HTTP 会话过程

Web 客户机与 Web 服务器之间一次 HTTP 会话的过程如下：1)客户机与服务器的80端口之间建立一条 TCP 连接；2)客户机向服务器发送 GET 请求；3)服务器回应信息，传送请求的文件；4)释放 TCP 连接。

从上面我们注意到 Web 访问的性能不仅与服务器有关，也与网络的传送和接收有关。端到端的响应时间就是由上面几个步骤所花时间之和决定的。因此我们把 Web 服务器和 Internet 网络一起看作一个 Web 服务系统来建模。

建模中做了这样的简化和假设：a)不考虑 HTTP 和 TCP/IP 协议底层的细节，仅抽象出与服务性能有关的处理。把 DNS 解析和建立 TCP 连接的时间看作服从平均服务时间为 A 的负指数分布。b)HTTP 协议采用简单请求信息和简单回应信息格式。不做"if-modified-since"等信息头的操作。因此 HTTP GET 请求信息与文件长度比起来可忽略。c)请求文件长度是负指数分布的，因此服务时间也是负指数分布的。d)到达为泊松流，到达率为 λ。

1.2 单服务器系统模型

图1为 Web 单服务器系统的性能分析模型。该模型包括四个节点，每个节点是一个单排队系统，其中节点 P_A 和 P_B 是 Web 服务器本身的模型，P_C 和 P_D 两个节点是 Internet 通信网络模型。文件请求(即任务)以频率 λ 到达服务器，在节点 P_A 进行 TCP 连接等初始化处理，在 P_B 节点一个缓存的数

*)国家"九五"重点科技攻关资助项目(96-743-01-04-02)。喻莉 博士生。

据从文件中读取、处理，然后传给网络，在节点 P_C 以服务器端的传输速率将这一块数据传送到 Internet。这些数据通过 Internet 传送，最终被客户端的浏览器接收，即节点 P_D 接收。若该文件还未被完全传送，则返回到节点 P_B 做进一步处理；否则，文件已被完全传送，则退出网络。

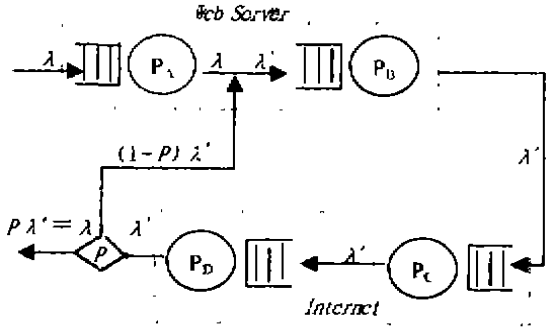

图1 Web 服务系统的性能分析模型

节点 P_A 为单 M/M/1排队系统，到达率为 λ，服务率为1/A；节点 P_B 为单 M/M/1排队系统，到达率为 λ'，HTTP 信息头的固定处理时间为 I，而对文件进行读取的速度为 U，缓存长度假设为 B，则节点 P_B 的服务率为1/[I+B/U]；节点 P_C 也是 M/M/1单排队系统，到达率为 λ'，网络速度为 S，因此其服务率为 S/B；节点 P_D 由许多浏览器程序接收，看作 M/M/∞系统，到达率为 λ'，服务率为 R/B，R 为客户端程序接收数据的平均速率。在节点 P_D，任务以概率 p 转向节点 P_B。整个模型构成了一个开放的马尔可夫排队网络。

给出平均文件长度 F 和缓存长度 B，则文件被完全传送的概率是 P=B/F。排队理论要求离开任一稳态节点的速率必须等于其到达率。由 Jackson 定理，节点 P_B 的到达率 λ' 等于网络到达率与任务从节点 P_D 返回 P_B 的速率之和，即

$$\lambda'=\lambda+(1-p)\lambda' \quad \text{则 } \lambda'=\lambda/p=\lambda F/B$$

节点 P_A：
$$\rho_A=\frac{\lambda}{1/A}=\lambda A$$
$$E[N_A]=\frac{\lambda A}{1-\lambda A}$$

节点 P_B：
$$\rho_B=\frac{\lambda'}{1/(I+B/U)}=\frac{\lambda F(B+UI)}{BU}$$
$$E[N_B]=\frac{\lambda F(B+UI)}{BU-\lambda F(B+UI)}$$

节点 P_C：
$$\rho_C=\frac{\lambda'}{S/B}=\frac{\lambda F}{S}$$
$$E[N_C]=\frac{\lambda F}{S-\lambda F}$$

节点 P_D：
$$\rho_D=\frac{\lambda'}{R/B}=\frac{\lambda F}{R}$$
$$E[N_D]=\frac{\lambda F}{R}$$

最后得平均响应时间为：

$$T=\frac{F}{R}+\frac{A}{1-\lambda A}+\frac{F}{S-\lambda F}+\frac{F(B+UI)}{BU-\lambda F(B+UI)} \quad (1)$$

由简化条件 a)和 b)，节点 P_A 的初始化时间 A 和节点 P_B 的固定处理时间 I 都很短，且由(1)式看出对响应时间影响小，因此可对模型进一步简化，见图2。

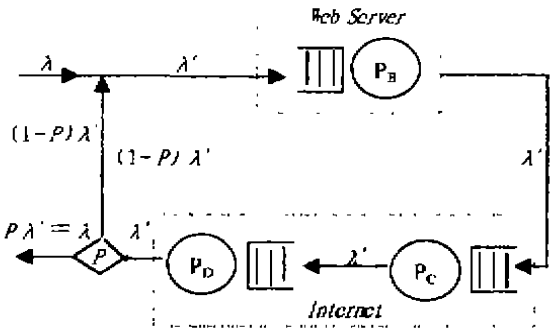

图2 简化的 Web 服务系统性能分析模型

相应地，式(1)中取 A=0，I=0，得：

$$T=\frac{F}{R}+\frac{F}{S-\lambda F}+\frac{F}{U-\lambda F} \quad (2)$$

1.3 多服务器系统模型

为了在后面分析增加服务器对网络服务性能的影响，我们将该模型扩展到了多服务器系统，建立模型如图3所示。

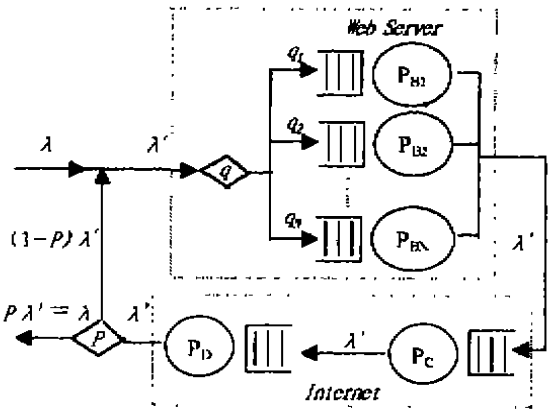

图3 多服务器系统性能分析模型

假设有 N 个服务器，分别以概率 q_i 访问服务器 i，各服务器的处理速率为 U_i，则服务率为 U_i/B(i=

1,2,…,N)。仿上分析,利用 Jackson 定理得到多服务器系统的平均响应时间:

$$T=\frac{F}{R}+\frac{F}{S-\lambda F}+\frac{q_1F}{U_1-q_1\lambda F}+\frac{q_2F}{U_2-q_2\lambda F}+\cdots\cdots+\frac{q_nF}{U_n-q_n\lambda F}$$

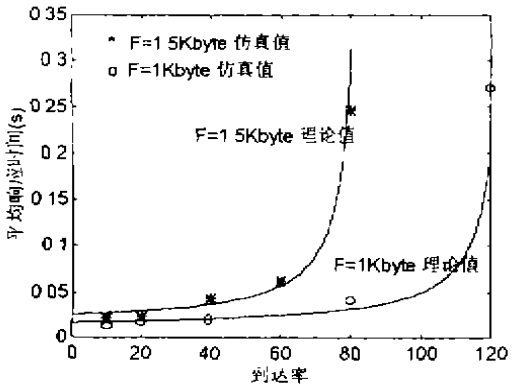

图4 不同文件长度下的仿真结果

3 Web 服务模型的参数分析

由公式(2),服务系统响应时间涉及到五个参数:a)网络到达率 λ;b)平均文件长度 F;c)服务器服务速率 U;d)服务网络带宽 S;f)客户网络接收速率 R。下面逐一分析,

1)到达率 λ 的影响。显然,λ 越大,响应时间 T 越长,见图6(略)。但 T 的增长并不是线性的,而是存在一个服务容量的极限问题,一旦到达率接近于服务容量的极限值,响应时间将无限增大,服务系统以越来越慢的速度服务,以致使很少有文件能成功地被服务,系统陷入死锁状态。

文件长度越长,服务容量极限值越小,则系统越容易陷入死锁状态。如何才能避免陷入死锁状态呢?对于到达率的监视较难实现,我们利用 Little 公式,N=λT,当接近极限值时,λ 增大,T 亦增大,则 N 会急剧增大,这里 N 即对应于同时的 TCP 连接数目,因此可以在服务器软件中对 HTTP 请求加以限制,当系统接近死锁状态时,限制 TCP 连接数目。

2)文件长度 F 的影响。见图7(略),文件增大,响应时间迅速增加,且到达率越大增长越迅速。

3)网络带宽 S 的影响。见图8(略),网络带宽增大,响应时间呈双曲递减,而且由图9(略)看到 S 增大,服务容量极限值也明显增大,则系统工作范围将明显增大。

2 仿真结果

我们用仿真软件进行了仿真,其结果与模型分析结果比较吻合,只取其中部分结果示于图4、图5,因而验证了分析假设的合理性以及模型的有效性。

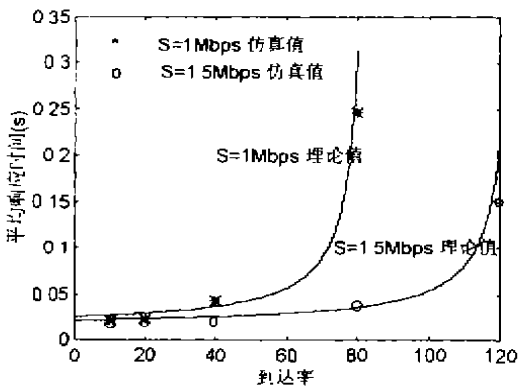

图5 不同带宽下的仿真结果

4)服务器处理速度 U 的影响。见图10(略),服务器处理速度加快,则响应时间呈双曲递减。但由图10也可以看到,处理速度增加到一定程度后,对响应时间的影响就很小了,几乎不再变化。这是因为当服务器处理速度足够大时,网络瓶颈可能已是网络带宽,这时无限制地增加处理速度是无济于事的,下节将对比提高网络性能的方法。又由图11(略),处理速度对服务容量极限值几乎无影响。

5)增加服务器对响应时间的影响。在这里,我们利用公式(2)、(3),从理论上加以证明。取 N=2,$q_1=q_2=1/2$,假设原来一台服务器的处理速度是 U,增加的一台服务器的处理速度为 U'=kU(k>0)。则 单服务器时,

$$T=\frac{F}{R}+\frac{F}{S-\lambda F}+\frac{F}{U-\lambda F} \quad (3)$$

多服务器时:

$$T'=\frac{F}{R}+\frac{F}{S-\lambda F}+\frac{\frac{1}{2}F}{U-\frac{1}{2}\lambda F}+\frac{\frac{1}{2}F}{kU-\frac{1}{2}\lambda F} \quad (4)$$

(4)式减去(3)式得:

$$T'-T=F\cdot\frac{(U-\lambda F)^2+(1-2k)U^2}{(2kU-\lambda F)(2U-\lambda F)(U-\lambda F)}$$

对稳定系统,需保证 $\frac{\frac{1}{2}\lambda F}{kU}<1$、$\frac{\frac{1}{2}\lambda F}{U}<1$、$\frac{\lambda F}{U}<1$。

因此分母>0,只需考查分子,显然分子是关于

(下转第79页)

if O_{id_j} in $WO_{deleted}$ then
$WO_{updated} = WO_{updated} + \{O_{id_i}\}$;
end if;
endfor;
$WO_{inserted} = WO_{inserted} - WO_{deleted}$;
$WO_{updated} = WO_{updated} - WO_{inserted}$;
$WO_{updated} = WO_{updated} - WO_{deleted}$;
算法结束

算法2.3(ViewUpdate)

/*这个算法的思想是假设视图 V_{id} 是由 n 个类 C_1, C_2, …, C_n 直接导出的，利用 $W_{instance}(C_1)$、$W_{instance}(C_2)$……$W_{instance}(C_n)$ 中插入、删除和修改的对象的集合对视图 V_{id} 进行维护和物化。*/
Input：基类 C_1, C_2, …, C_n 和直接由这些类导出的视图 V_{id}
Output：$W_{instance}(V_{id})$。
步骤：
for i:=1 to n do
IDU(C_i, V_{id});
for every object VO_{id} $\{O_{id_1}, O_{id_2}, \cdots, O_{id_n}\}$ in $W_{instance}(V_{id})$ do
if $\{O_{id_1}, O_{id_2}, \cdots, O_{id_n}\}$ 中含有 $WO_{updated}$ 和 $WO_{deleted}$ 中的元素 then
$W_{instance}(V_{id}) = W_{instance}(V_{id}) - VO_{id}$;
endif;
endfor;
$WO_{inserted} = WO_{inserted} + WO_{updated}$;
for 分别对应属于 $W_{instance}(C_1)$, …, $WO_{inserted}$, … $W_{instance}(C_n)$ 的每个对象 O_{id_1}, …, O_{id_i}, …, O_{id_n} do
If $O_{id_1}, O_{id_2}, \cdots, O_{id_n}$ 满足 $P_{expression}(V_{id})$ 的属性表达式 then
$W_{instance}(V_{id}) = W_{instance}(V_{id}) + \{O_{id_1}, O_{id_2}, \cdots, O_{id_n}\}$;Endif;
Endfor;Endfor;
算法结束

结论 在本文中我们提出一种在面向对象数据库中基于多个类的视图维护模式，并给出了简单的算法，这种算法比较节省时间和费用，尤其对于大型面向数据库来讲，其优势很明显。这种模式还可以应用到关系数据库和嵌套数据库的视图维护中。

参考文献

1 Alhajj R, Polat F. View Maintenance in Object-Oriented Databases. In: Database and Expert System Applications, Proc. 7th Intl. Conf. DEXA'96 Zurich, Switzerland, 1996
2 Alhajj R, Polat F. An Object-Oriented Query Model Enforcing Closure and Reusability. Journal of Mathematical Modeling and Computing, 1996, 6(April)
3 何炎详，郑振楣，石树刚. 面向对象数据库. 武汉大学出版社

(上接第89页)

k 的单调递减函数。当 $k=\frac{1}{2}+\frac{1}{2}\left(1-\frac{\lambda F}{U}\right)^2$ 时，$T'-T=0$。显然，取 k=1/2时 $T'-T>0$ 说明若增加的服务器速度为原来的一半，响应时间会增大，即性能反而下降；取 k=1时 $T'-T<0$ 说明若增加一个同等速度的服务器，响应时间会减少，即性能提高。

所以所增加服务器的速率稍大于1/2倍原速率就可以优于原来的性能，当然，增加服务器的速度越快，性能提高得越多，但代价也就越大，在下一节将看到有时增加服务器还不如直接用一个更高速率的服务器代替更经济。

4 改善服务系统性能的方法比较

从上面的理论分析，我们已经知道提高 Web 服务系统的性能有三种方法：1)增加网络带宽；2)提高服务器处理速度；3)另外增加服务器。哪种方法更好呢?我们进行了对比，结果如下：

a)网络为瓶颈时，方法从优到劣的顺序依次为：增加网络带宽；增加服务器处理速率；增加一台服务器；而如果增加一台原速率一半的服务器，反而会降低性能。

b)服务器为瓶颈时，最好的方法是增加服务器的处理速率；其次是增加一台服务器还是增加网络带宽取决于网络状况，在我们设置的环境下，如果到达率低于38，增加网络带宽比增加服务器更有效，高于38时，选择增加一台服务器的方法更好；同样，如果增加的服务器速率低于原来的一半，则会更糟。

结束语 本文提出了一种新的 Web 服务系统的性能分析模型，研究了避免死锁状态的方法和增加新服务器的原则，这对于网络管理和网络升级都具有实际意义，而且利用本文模型直接对一个 Web 服务系统的性能进行分析，可节省分析时间，避免了用仿真分析耗时长的缺点，因此对于网络规划设计也有重要作用。

参考文献

1 Fujita Y, et al. Analysis of Web Server Performance Evaluation of Web System. In: The 6th IEEE Intl. Conf. on Network'98. Singapore, July 1998
2 Tanenbaum A S. Computer Networks. 3rd Edition. Prentice Hall, 1996
3 Trivedi K S, et al. Performance Evaluation of Client-Server Systems. IEEE trans on PDS, 1993, 4(11)