第33卷 第20期 计 算 机 工 程 2007年10月
Vol.33 No.20 Computer Engineering October 2007
·软件技术与数据库· 文章编号：1000—3428(2007)07—0089—02 文献标识码：A 中图分类号：TP368

基于实时监控的 Apache 自适应调节机制

吴 飞，罗 军，李 慰
（国防科学技术大学计算机学院，长沙 410073）

摘 要：为了改善Apache服务器的性能及服务质量，提出了一种基于实时监控的Apache自适应调节机制。该机制运用分析性能模型结合组合搜索技术，根据负载变化，自适应地完成调节任务。该文结合Apache的体系结构设计了实现该机制的模型，描述了模型中各模块的具体功能，给出了实现技术和相关算法，并对其功效进行了评估。

关键词：Apache；实时监控；分析性能模型；服务质量

Adaptive Tuning Mechanism for Apache Based on Real-time Monitoring

WU Fei, LUO Jun, LI Wei
(School of Computer Science, National University of Defense Technology, Changsha 410073)

【Abstract】In order to improve the performance and the QoS of Apache server constantly, an adaptive tuning mechanism for Apache based on real time monitoring is introduced, which uses an analytic performance model combined with combinatorial search techniques to complete some tuning tasks as the workload changing. A model with this mechanism is designed into Apache, and the function of each module, the technology and algorithm of model is depicted. An experiment is conducted for evaluating the efficiency of this mechanism.

【Key words】Apache; real-time monitoring; analytic performance model; quality of service(QoS)

Apache 的 MPM 模块采用多路并行处理技术，可以动态调节系统中空闲的服务进程或线程数，使其保持一定数量，实现调节任务，但修改配置需要重新启动 Apache 才能生效，并不能满足负载变化的特性。

Bennani和Menascé等应用自主管理技术对Apache进行了扩充，其主要思想是在Apache中建立一种满足服务质量需求的机制。并且论证了在负载高可变的条件下，保证服务质量持续满足的健壮性[1~3]。为实时调整和改善Apache服务器的性能及服务质量，本文设计了一个基于实时监控的Apache自适应调节模型，该模型通过获取性能及服务质量的偏差量，运用分析性能模型结合组合搜索技术，将合理的配置策略传递给Apache，并实时进行调节。

1 模型结构

Apache实时监控需要对Apache透明，为了让自适应调节机制与Apache有机地结合，建立了如图 1 所示的模型。模型主要由 5 部分组成：负载监控模块，性能监控模块，服务质量监控模块，性能模型处理模块和配置控制模块。

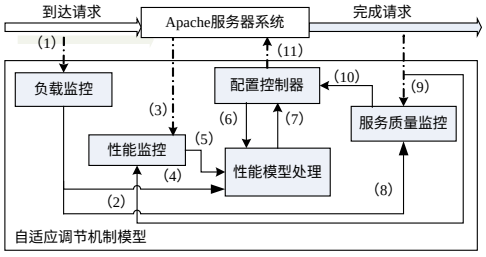

图 1 自适应调节机制模型

模型建立在 Apache 的基础之上，收集控制间隔内服务器工作负载的数据，监控当前服务器性能及服务质量值与期望值的偏差，结合分析性能模型搜索配置参数空间，根据搜索结果实时修改配置策略，使系统性能和服务质量一直保持最优。

1.1 负载监控模块

负载监控模块对到达请求的信息(1)进行采集和存储，计算控制间隔时间内请求的平均到达率。不同类型的请求对负载的影响程度不同，仅获得请求平均到达率是不够的，还必须明确请求的类型。该模块在进行数据采集的同时，在特定时刻统计请求类型信息，并给收集到的数据打上时间戳。

1.2 性能监控模块

性能监控模块作为整个控制部分的主要输入部分，采集和存储各种设备的性能数据，统计设备(如 CPU、Disk)利用率(3)，根据完成请求数(2)计算吞吐量，并根据设备利用率和吞吐量计算每个设备在控制间隔期间的服务需求。服务需求定义为每个请求在给定设备上的总服务时间，总服务时间不包括在该设备上的排队时间，比如 CPU 设备的服务需求为 Apache 的 CPU 利用率与吞吐量的商。

1.3 服务质量监控模块

服务质量监控模块处理 2 种信息：对完成请求(9)的信息进行采集、存储和接收负载监控模块收集到的负载数据。获得当前服务器的性能及服务质量值，在每个控制间隔的末尾

基金项目：国家“863”计划基金资助项目(2002AA1Z2101)；国家科技攻关计划基金资助项目(2005BA112A02)
作者简介：吴 飞(1981－)，男，硕士，主研方向：操作系统，软件工程；罗 军，研究员；李 慰，硕士
收稿日期：2006-10-30 **E-mail：**wufei81@gmail.com

检查二者与实际需求的偏差，判断是否需要改变配置，若需要，向配置控制器发出重新配置的通知(10)。

1.4 性能模型处理模块

性能模型处理模块接收 3 种信息：负载监控模块获得请求到达率(4)，性能监控模块计算出的服务需求(5)和配置控制模块输出的配置策略(6)。本文运用分析性能模型为 Apache 服务器进行建模。

分析性能模型由Markov链和排队网络(QN)模型组合而成，用来计算平均响应时间R、平均吞吐量X和拒绝概率P_{rej}，其中，Markov链为m个服务进程及等待服务进程的队列进行建模，排队网络模型用来计算服务进程处理请求的完成率。定义$X(k)(k=1,\cdots,m)$为有k个请求被处理时的完成率，可通过解包含k个正在处理请求服务进程的、由CPU和Disk组成的硬件子系统的排队网络模型获得，如图 2 所示。排队网络的解由均值分析(MVA)获得。

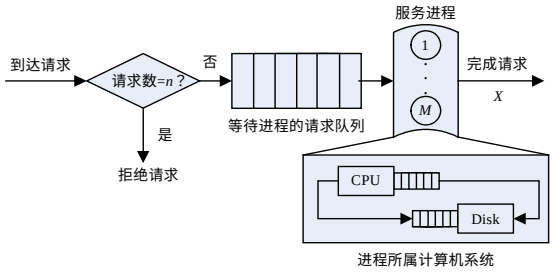

图 2 Apache 服务器队列模型

Markov链模型有n+1 个状态，如图 3 所示，状态k代表系统中请求数为k个（包括正在等待服务进程或正在占用服务进程的请求），图 3 中到达率 λ 是指请求到达速率，$X(m)$是指当m个服务进程完成请求服务的完成速率。因为系统中最多只有m个服务进程，所以当 $k>m$ 时，完成速率一直为$X(m)$。其中，R, X, P_{rej}可根据状态概率及Little规则[1,4,5]计算求得。利用该模型计算出对应于不同配置的服务质量值，并将结果(7)返回给配置控制模块。

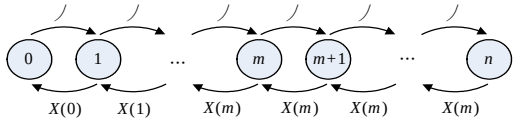

图 3 Markov 链模型

1.5 配置控制模块

配置控制模块运用组合搜索技术搜索配置空间，结合性能模型处理模块返回的服务质量值(7)，获得适应于当前服务器工作状态的优化配置，并将配置策略传递给 Apache，重新配置系统(11)。

2 实现技术与相关算法

2.1 性能及服务质量值的偏差计算

服务器的性能及服务质量由平均响应时间 R、平均吞吐量 X、拒绝服务概率 P 这 3 个主要性能分量来衡量，为使服务质量达到期望值，应考虑实际值与期望值的差别。

(1)定义 ΔQoS_R 为实际平均响应时间与期望响应时间的偏差为

$$\Delta QoS_R=\frac{R_{\max}-R_{\text{measured}}}{\max(R_{\max},R_{\text{measured}})}$$

其中， $R_{\max}$ 为可以容忍的最大平均响应时间； R_{measured} 为测量的平均响应时间。

(2)定义 ΔQoS_P 为实际拒绝概率与期望值的偏差为

$$\Delta QoS_P=\frac{P_{\max}-P_{\text{measured}}}{\max(P_{\max},P_{\text{measured}})}$$

(3)实际吞吐量与期望值得偏差定义为

$$\Delta QoS_X=\frac{X_{\text{measured}}-X^*_{\min}}{\max(X_{\text{measured}},X^*_{\min})}$$

其中， X_{measured} 为实际测量的吞吐量； $X^*_{\min}=\min(\lambda,X_{\min})$ ，定义 $X^*_{\min}$ 是为了保证系统负载很小情况，即 X_{measured} 不足以达到吞吐量需求最小值时计算的合理性。

(4)定义 ΔQoS 为当前服务器的性能及服务质量与期望值的偏差：

$$\Delta QoS=\omega_R\times\Delta QoS_R+\omega_P\times\Delta QoS_P+\omega_X\times\Delta QoS_X$$

其中，ω_R， ω_P， ω_X 为加权系数，取值范围为[0,1]; $\Delta QoS\ge 0$ 代表性能及服务质量满足服务水平协议(SLA)，否则，至少有一个指标不满足需求[2~3]。

2.2 配置空间组合搜索技术

配置空间由Apache的可调参数组成，包括等待服务的请求队列最大长度c_1、同时并行处理客户端请求的最大数量c_2、最小空闲进程的总数c_3、最大空闲进程总数c_4、组成的配置向量为(c_1, c_2, c_3, c_4)，考虑到c_1, c_2改动对性能影响比较大，改动对c_3, c_4对性能影响较小，且c_3, c_4依赖于c_1,c_2。为提高搜索效率，先搜索c_1, c_2，再根据负载变化确定c_3, c_4。算法如下：

```
NumHops ← 1;
C_curr ← C_0;
C_new ← C_0;
Repeat;
Improved ← False;
MaxQos ← Qos(C_curr)
For  i = 1  to  2  do
Begin
  If   c_i − 1 ≥ c_i^min
  Then  If  Qos(C_curr − I_i) > MaxQos
    Then  Begin
          MaxQos ← Qos(C_curr − I_i);
          C_new ← C_curr − I_i;
          Improved ← True;
    End
  If   c_i + 1 ≤ c_i^max
  Then  If  Qos(C_curr + I_i) > MaxQos
    Then  Begin
          MaxQos ← Qos(C_curr + I_i);
          C_new ← C_curr + I_i;
          Improved ← True;
    End
End
C_curr ← C_new;
NumHops ← NumHops + 1;
Until(¬Improved | (NumHops = MaxHops))
```

$$c_3=\frac{k}{\vec{I}_1\times\vec{C}_0^{\,T}}\times(\vec{I}_1\times\vec{C}_{curr}^{\,T});$$

$$c_4=\vec{I}_1\times\vec{C}_{curr}^{\,T}$$

其中，k代表系统中占用服务进程的请求数。

3 自适应调节机制评估

为了对自适应调节机制的效果进行评估，进行了如下实

(下转第 93 页)