

基于时间序列分析的应用服务器性能衰退模型

徐文彬^{1,2}, 齐勇¹, 侯迪¹, 刘亮³

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 常熟理工学院计算机系, 215500, 常熟;
3. IBM 中国研究中心, 100087, 北京)

摘要: 针对 JUFram 应用服务器老化情况, 设计了多种类型的客户请求程序和服务器端程序, 记录了各种参数共计 5 类 36 个. 对影响应用服务器中间件性能的主要指标“内存利用率”, 采用时间序列分析方法建模, 经采样数据的统计分析得出了自相关函数明显拖尾、偏自相关函数明显截尾的结果. 模型的仿真结果表明, 预测值与原始数据拟合良好, 且可作为系统再生的依据, 它适用于预测实际的运行系统. 与其他软件老化预测方法比较, 所提模型更切合系统的实际运行状况, 有利于提高系统的可用率.

关键词: 时间序列; 软件老化; 自相关函数

中图分类号: TP31 **文献标识码:** A **文章编号:** 0253-987X(2007)04-0426-04

Performance Degradation Model of Application Server System Based on Time Series Analysis

Xu Wenbin^{1,2}, Qi Yong¹, Hou Di¹, Liu Liang³

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;
2. Department of Computer, Changshu University of Technology, Changshu 215500, China;
3. IBM Research Center for China, Beijing 100087, China)

Abstract: Many kinds of client request applications and server programs are designed for the aging of JUFram application sever, in which 36 parameters in 5 groups are recorded. The key index “memory usage” that affects the performance of middleware of application server is modeled based on time series analysis. The results that the auto correlation function (ACF) is clearly tail-dragged and the partial auto correlation function is clearly tail-truncated can be drawn via detailed statistical analysis of sampled data. The result of model simulation shows that the predicted value fits the original data well and can be regarded as the basis of system rejuvenation, and is applicable for predicting runtime systems. Compared to other software aging prediction methods, the proposed method is more suitable for the practical case of systems, and the availability of the system is improved.

Keywords: time series; software aging; auto correlation function

软件在长期运行的过程中, 内在的缺陷随着时间逐渐累积会导致软件性能下降, 最终造成系统失效和停机, 这一过程就称为软件老化. 近年来的研究表明, 许多软件系统都存在老化现象^[1-2], 为了解决这一问题, Huang 等人提出了一种主动、预防

性的软件容错策略, 称为软件再生^[1]. 目前, 广泛采用的有两种方法: 一是基于模型的方法, 即通过数学方法求得软件再生时间的最优值, 从宏观上这类方法一般不能反映实际系统的真实老化情况; 另一是基于测量的方法, 即根据系统的实际老化情况决定

再生动作,如通过对某些老化现象导致的低效或停机系统资源进行监视,然后用统计学模型估算和验证软件老化情况^[3],或将资源消耗的速度视为时间的函数和系统负载的函数^[4],以及采用基于时序分析的方法来测量和估算 Web 服务器的软件老化情况等^[5]。

上述研究基本上集中在研究操作系统老化的问题上,对应用服务器中间件的老化研究均未涉及。本文采用基于测量的方法对西安交通大学计算机系研制的一个实际 JUFrame 应用服务器进行实验设计,不但测量了必要的系统参数,而且也测量了内部相关组件的特征参数,同时为老化的标志性指标建立了自回归(AR)模型,所得模型能很好地捕获系统的动态特征。

1 实验方案

JUFrame 是西安交通大学自主开发的应用服务器中间件软件,它采用 J2EE 标准,为企业资源集成、事务处理、业务组件管理和部署等基础服务的中间件运行提供了应用支撑平台。为了全面考察该应用服务器的老化情况,需要设计客户端程序(发出合适的请求并记录响应时间)和服务器端程序(记录各种参数,包括服务器本身的一些指标和各个组件的性能指标)。实验设计如下。

(1)确定请求类型。将请求分成 4 种类型:有状态会话、有事务请求,有状态会话、无事务请求,无状态会话、有事务请求,无状态会话、无事务请求。另外,还有一种模拟客户请求的真实场景。

(2)在当前负载情况下确定服务器处理的最大峰值。通过逐渐增大请求率并记录响应时间,如果发现某个请求率下的响应时间超时、丢包且随之丢包率也在增加,则可认为服务器处理的最大峰值就是产生丢包之前的请求率。

(3)将测得的服务器处理的峰值从低到高分档,以确定客户端的请求率。

(4)在当前请求状态下,按一定的时间间隔记录客户端、服务器端的状态。

本实验共测得了 5 类 36 个参数,基本上能反映系统的运行情况。从实验结果看出,每次系统宕机都是在“已用内存”达到一定值以后出现的,所以可针对其建立动态模型,以用于实际系统的预测。

2 自回归滑动平均模型

将系统观测所得并按照时间顺序排列的数据称

为时间序列数据,计为 $\{X_t, t=1, 2, \dots, n\}$ 。一个变量在某时刻 t 的取值(记为 X_t)很可能与它在 t 以前的取值($X_{t-1}, X_{t-2}, \dots, X_{t-p}$)有关。采用时间序列分析的目的就是要研究 X_t 与 $X_{t-1}, X_{t-2}, \dots, X_{t-p}$ 之间的线性关系,这种线性关系分 3 种情况,第 1 种情况是

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \dots + \varphi_p X_{t-p} + a_t \quad (1)$$

式中: φ_p 为自回归系数, p 为模型的自回归阶次; $\{a_t\}$ 为零均值的白噪声序列。式(1)可简记为 $M_{AR}(p)$, 称 AR 模型。

第 2 种情况是

$$X_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2)$$

式中: q 为模型滑动平均阶次。式(2)可简记为 $M_{MA}(q)$, 称为滑动平均(MA)模型。

第 3 种情况是

$$X_t - \varphi_1 X_{t-1} - \dots - \varphi_p X_{t-p} = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (3)$$

式(3)可简记为 $M_{ARMA}(p, q)$, 称为自回归滑动平均(ARMA)模型。

上述 3 种模型的阶次 p, q 以及各个系数 $\varphi_i (i=1, 2, \dots, p), \theta_i (i=1, 2, \dots, q)$ 均需要根据实测数据来推断。

3 时间序列分析步骤

3.1 数据预处理

时间序列分析的前提是,时间序列数据必须是平稳的。在分析之前必须先去除孤立点数据并平稳化,剔除孤立点可采用 Tukey 提出的方法,其基本思想是:先产生一个曲线的平滑估计,然后把它从数据中减掉。这样做,易于识别孤立点。该方法利用了“中位数”是均值的鲁棒(robust)估计这一事实,其步骤如下。

(1)从 $x(i)$ 构造一个新序列 $x_1(i)$, 即取 $x(1), \dots, x(5)$ 的中位数作为 $x_1(3)$, 然后舍去 $x(1)$ 、加入 $x(6)$ 取中位数得 $x_1(4)$, 依次类推,直到加入最后一个数据。换言之,总是在相邻 5 个数据中择取中位数。显然, $x_1(i)$ 的项数比原 $x(i)$ 项数少 4。

(2)用类似的方法在 $x_1(i)$ 的相邻的 3 个数据中择取中位数,构成序列 $x_2(i)$ 。

(3)由序列 $x_2(i)$ 按 $x_3(i) = x_2(i-1)/4 + x_2(i)/2 + x_2(i+1)/4$ (这是一海宁平滑滤波器)构成序列 $x_3(i)$ 。

分析序列 $x(i) - x_3(i)$, 若满足 $|x(i) - x_3(i)| > k$ (预定值), 则用一内插值代替 $x(i)$ 。假设滤波以

后的数据记为 $y_i (i=1, 2, \dots, n)$, 可将前一半的数据用于建模, 将后一半的数据用于模型检验。

3.2 趋势项提取

使用 Matlab 中的函数 polyfit, 可以很容易地得到内存使用与时间的线性关系 $y_i = 10\,726 + 8\,089.2t$ 。去除原始数据中的关系项, 便可获得残差序列, 如图 1 所示。

从图中可以看出, 残差序列是一平稳序列, 以下用 ARMA 为该序列建模。

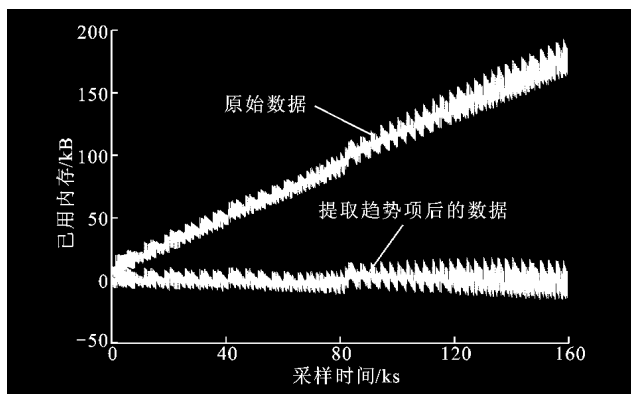


图1 已用内存的原始数据和提取趋势项以后的数据

3.3 自相关函数及偏自相关函数曲线

自协方差函数计算公式为

$$\hat{\gamma}_k = \frac{1}{N} \sum_{i=1}^{N-k} x_i x_{i+k} \quad k = 0, 1, \dots, N-1$$

式中: N 为建模的数据个数, 则自相关函数 (ACF) 的计算公式为

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} \quad k = 0, 1, \dots, N-1$$

偏自相关函数 (PACF) φ 的计算流程为

$$\left. \begin{aligned} \varphi_{11} &= \rho_1 \\ \varphi_{(k+1)(k+1)} &= (\rho_{k+1} - \sum_{j=1}^k \rho_{k+1-j} \varphi_{kj}) (1 - \sum_{j=1}^k \rho_j \varphi_{kj})^{-1} \\ \varphi_{(k+1)j} &= \varphi_{kj} - \varphi_{(k+1)(k+1)} \varphi_{k(k+1-j)}, j = 1, 2, \dots, k \end{aligned} \right\}$$

根据前一半数据可计算得到自相关函数和偏自相关函数, 如图 2、图 3 所示。

3.4 模型定阶

计算自相关函数和偏自相关函数, 可以看出系统的部分特征并初步判别模型的阶次。根据时间序列的相关理论可知: AR 模型的自相关函数具有拖尾性, 偏自相关函数具有截尾性; MA 模型的自相关函数具有截尾性, 偏自相关函数具有拖尾性; AR-MA 模型的自相关函数和偏自相关函数均具有拖尾性。从图 3、图 4 中还可以看出, 自相关函数渐近趋于 0 并且没有明显的周期性, 而偏自相关函数大概

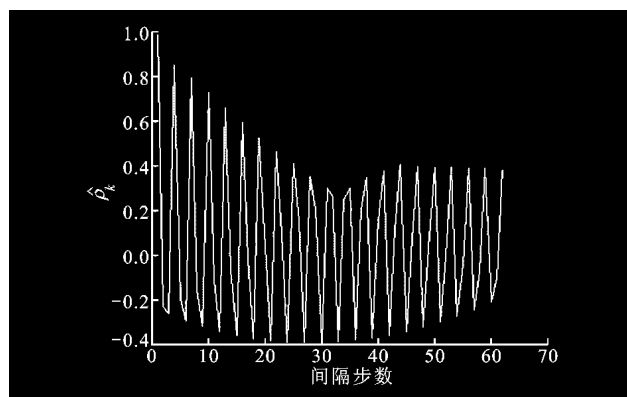


图2 自相关函数

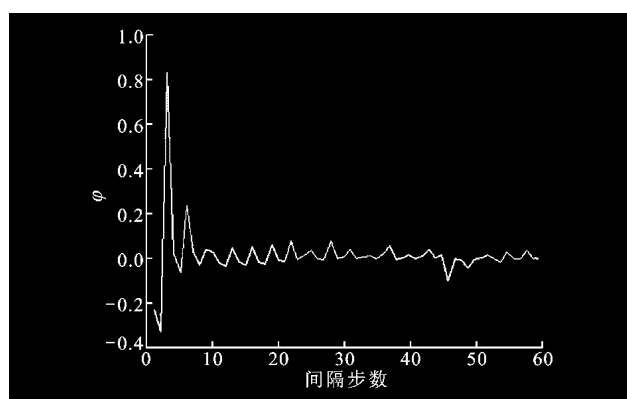


图3 偏自相关函数

在第 4 拍以后截尾, 故可考虑用 AR 模型建模。由于计算机的计算存在误差, 所以可采用判断准则来判断截尾性, 即 $\hat{\rho}_k$ 的分布渐近于正态分布 $N(0, (1/(N)^{1/2}))^2$, 则认为 $\hat{\rho}_k$ 存在截尾性。首先计算 $\hat{\rho}_1, \dots, \hat{\rho}_M$, 如果 $\hat{\rho}_{q+1}, \dots, \hat{\rho}_M$ 满足 $|\hat{\rho}_k| \leq \frac{1}{(N)^{1/2}}$ 的比例占总数的 68.3%, 那么第 1 个满足条件的 q 就是 $\hat{\rho}_k$ 的截尾处。 $\hat{\varphi}_{kk}$ 的分布与 $\hat{\rho}_k$ 相同, 故其截尾性判断也同上。经程序计算, 模型阶次初定为 4。

3.5 参数识别

(1) AR 模型参数识别的矩估计计算式为

$$\begin{pmatrix} \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_p \end{pmatrix} = \begin{pmatrix} 1 & \hat{\rho}_1 & \cdots & \hat{\rho}_{p-1} \\ \hat{\rho}_1 & 1 & \cdots & \hat{\rho}_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\rho}_{p-1} & \hat{\rho}_{p-2} & \cdots & 1 \end{pmatrix}^{-1} \begin{pmatrix} \hat{\rho}_1 \\ \hat{\rho}_2 \\ \vdots \\ \hat{\rho}_p \end{pmatrix}$$

(2) 利用 F 检验进行筛选, 可最终确定模型的阶数。

在 F 检验中, $H_0: M_{AR}(p-1)$ 模型是合适的, 即 $\varphi_p = 0$; $H_1: M_{AR}(p-1)$ 模型是不合适的, 即 $\varphi_p \neq 0$ 。

统计量 $F = (A_1 - A_0)(N - 2p) / A_0$ 服从 $F(1, N - 2p)$ 分布, 其中 A_0 是 $M_{AR}(p)$ 的残差平方和, A_1

是 $M_{AR}(p-1)$ 的残差平方和, N 是量测数据数目. 给定显著性水平 α , 查 F 分布表可得临界值 F_α . 若 $F > F_\alpha$, 则拒绝 H_0 , 即 $M_{AR}(p-1)$ 模型是不合适的, 否则接受 H_0 , 前者称为 F 检验显著, 后者为 F 检验不显著. 通过编程实现了 F 检验, 利用其确定的阶次为 6, 并同时获得了模型的参数.

3.6 模型检验

利用后一半数据可进行模型检验, 为此给出模型的一步预报递推公式

$$\hat{\varphi}_k(1) = \varphi_1 X_k + \varphi_2 X_{k-1} + \cdots + \varphi_p X_{k-p+1}$$

式中: $\hat{\varphi}_k(1)$ 表示在 k 时刻已知当前的采样值 X_k 预测 $k+1$ 时刻的值, 也就是一步预测值. 可以证明, 随着预测步数的增加, 预测误差的方差会越来越大. 此处仅进行一步预测, 预测数据和原始数据的曲线如图 4 所示.

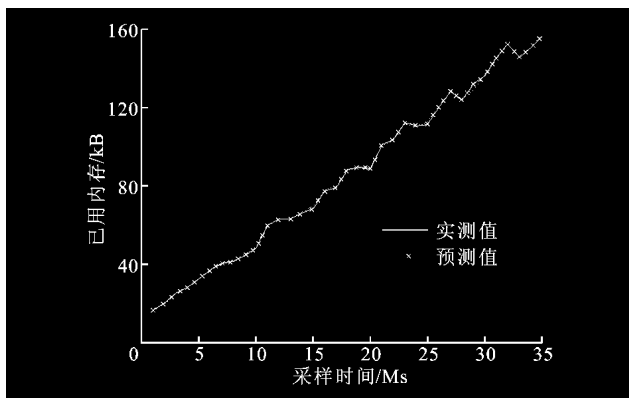


图4 模型检验图

从图 4 可以看出, 预测曲线与原始曲线的拟合结果很好, 这说明所用模型是合适的. 由于这是一步预测结果, 所以预测方差比较小, 如果是多步预测, 则预测方差将随之增加. 由于软件老化是一种缓慢的变化过程, 采样间隔往往很大, 所以多步预测就不是很必要了.

4 结 论

本次实验为系统老化的关键指标“已用内存”建

立了动态 AR 模型, 该模型的预测曲线与原始曲线能很好地拟合, 故它可应用于实际系统的老化预测. 假如系统的指标到达某一阈值时仍处于易于宕机的状态, 则监控软件将启动系统重新载入. 一般情况下的系统老化指标不是单一的, 所以今后的工作应该考虑用多维时间序列建模. 另外, 时间序列分析本质上是一种线性逼近, 有时效果未必很好, 应该考虑采用非参数方法进行拟合, 或者直接采用机器学习方法对系统的老化过程进行建模、预测, 这样效果会更好.

参考文献:

- [1] Huang Y, Kintala C, Kolettis N, et al. Software rejuvenation: analysis, module and applications [C]//Proceedings of 25th IEEE International Symposium on Fault Tolerant Computing, Los Alamitos, USA: IEEE Computer Society, 1995:381-390.
- [2] Avritzer A, Weyuker E J. Monitoring smoothly degrading systems for increased dependability [J]. Empirical Software Engineering, 1997, 2 (1):59-77.
- [3] Garg S, van Moorsel A, Vaidyanathan K, et al. A methodology for detection and estimation of software aging [C]//Proceedings 9th International Symposium on Software Reliability Engineering, Los Alamitos, USA: IEEE Computer Society, 1998:283-292.
- [4] Vaidyanathan K, Trivedi K S. A measurement-based model for estimation of resource exhaustion in operational software systems [C]//Proceedings 10th International Symposium on Software Reliability Engineering, Los Alamitos, USA: IEEE Computer Society, 1999:84-93.
- [5] Li Lei, Vaidyanathan K, Trivedi K S. An approach for estimation of software aging in a Web server [C]//Proceedings of 2002 International Symposium on Empirical Software Engineering, Los Alamitos, USA: IEEE Computer Society, 2002:91-100.

(编辑 苗凌)