

基于平均负载混沌预测的服务器老化研究

葛君伟¹, 陈善凤², 方义秋²

(1 重庆邮电大学 软件学院, 重庆 400065; 2 重庆邮电大学 计算机科学与技术学院, 重庆 400065)

摘 要: 首先通过 G-P 算法分析关联维数与嵌入维数的变化识别时间序列混沌特性, 然后用小数据量方法求最大 Lyapunov 指数, 基于最大 Lyapunov 指数对平均负载时间序列建立预测模型. 实验结果表明预测模型能比较精确的对 Web 服务器系统平均负载进行短期预测, 依据平均负载能有效地推测服务器老化情况, 为执行软件自愈提供依据.

关键词: 服务器老化; 时间序列; 平均负载; 软件自愈

中图分类号: TP311.53

文献标识码: A

文章编号: 1000-7180(2011)03-0137-04

Research on Server Aging Based on Average Load Chaotic Forecast

GE Jun-wei¹, CHEN Shan-feng², FANG Yi-qiu²

(1 College of Software, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

2 College of Computer Science and Technology, Chongqing University of Posts
and Telecommunications, Chongqing 400065, China)

Abstract: At first, this paper identify the average load chaotic character which can be obtained by using G-P algorithm to analyze correlation dimension changing with embedding dimension, and then get the largest Lyapunov exponent through small data method, build chaotic time series prediction model based on largest Lyapunov exponent for average load time series. The experimental results show that the prediction model can precisely make short-time prediction to the Web server's average load, which can efficiently predict the aging situation of the sever and provide foundation for the software rejuvenation.

Key words: server aging; time series; average load; software rejuvenation

1 引言

软件在长期不间断的运行过程中, 内在的缺陷随着时间逐渐累积会导致软件性能下降, 最终造成整个软件系统崩溃, 这一过程称为软件老化或软件衰退. 为了解决这一问题, 提出采取主动的、积极的、预防性的软件容错策略^[1]. 目前, 广泛采用两种方法对软件老化进行研究: 一种是基于模型的方法, 另一种是基于测量的方法. 目前主要是通过测试工具收集客户端响应时间, 利用服务器程序监测和收集 Web 服务器平均负载、内存、交换空间等性能参数变化情况^[2], 然后定义的性能评价指标估算服务器

老化. 在所有这些参数中, 平均负载对响应时间影响最大^[3], 响应时间很大部分由平均负载来决定, 而响应时间是一个 Web 服务器健康与否的重要标志, 所以对服务器老化的判断重点在于动态地对平均负载变化状况进行准确估计或预测.

Web 服务器系统是一个非线性系统^[3], 蕴含丰富的动力学信息, 如何提取这些信息并应用到服务器老化中是研究的重要方面. 文中以混沌理论为基础, 验证平均负载时间序列的混沌性, 构建基于平均负载时间序列的 Lyapunov 指数预测模型, 预测服务器平均负载的变化, 依据平均负载的预测数据判定服务器老化情况.

收稿日期: 2010-04-27; 修回日期: 2010-06-24

基金项目: 重庆市教委科学技术研究项目(KJ090519)

2 平均负载混沌性分析方法

系统平均负载被定义为在特定时间间隔内运行队列中的平均进程树数目. 随着访问量不断增加, 系统的可用内存, Buffer, Cache, swpd(被使用的虚拟内存)与 IO 读写次数也会随之发生相应变化, CPU 要不断地处理用户请求, 平均负载会跟随变化, 所以平均负载因受多种因素影响而会出现无序运动. 平均负载只要证明具有混沌性以后, 就可以用混沌时间序列预测方法预测其变化, 所以首先对平均负载进行混沌性分析.

首先利用混沌理论中的重构相空间技术, 设 $(x_1, x_2, \dots, x_{n-1}, x_n)$ 为平均负载时间序列, 根据 Takens 嵌入定理, 构造具有 N 个相点的 m 维相空间, 找到一个合适的嵌入维, 即如果延迟坐标的维数 $m \geq 2d + 1$, d 是动力系统的维数, 则相空间每一点可定义为

$$Y(t_i) = [x(t_i), x(t_i + \tau), \dots, x(t_i + (m-1)\tau)] \quad (1)$$

其中, $i = 1, 2, \dots, N$, $N = n - (m-1)\tau$, 轨迹上的每个相点 $Y(t_i)$ 实际上表示 m 维数据空间上的一个点. 相点在嵌入空间中的轨线将变得光滑, 可以恢复出微分同胚下原动力系统的动力学性质.

在平均负载时间序列分析过程中, 通过 G-P 算法^[4] 计算吸引子维数. 在该算法中, 首先是对嵌入维数 m 取值范围进行估计, 在 m 粗略估计值的范围内对 m 取不同的值, 然后分别求取吸引子的维数 d , 当吸引子的维数 d 达到饱和时的取值, 即为重构相空间的实际维数 m . 从中任取 m 构造相空间, 对所有的 N 个相点, 若给定一个任意小的数 r , 计算所有相点间的欧氏距离, 比较有多少个点对之间的距离 $\|Y(k) - Y(j)\| \leq r$. 把距离小于 r 的点对数占总点对数 N^2 之比记作 $C(m, r)$, 即关联积分:

$$C(m, r) = \frac{1}{N^2} \sum_{k, j=1}^N \theta(r - \|Y(k) - Y(j)\|) \quad (2)$$

其中, θ 为 Heaviside 函数, 即 $\theta(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$

对于给定 r 后, $C(m, r)$ 表示两点间距离小于 r 的概率, 当 r 增大时, $C(m, r)$ 将以 r^{D_m} 的比率增大, 即 $C_m(r) \propto r^{D_m}$. 两端取对数后为 $\ln C_m(r) \propto D_m \ln r$. 对于不同 r 值求出 $C(m, r)$, 将这些不同的 r 值和 $C_m(m, r)$ 代入式(2)中, 即可拟合出吸引子维数 D_m . m 是否存在是随机系统与混沌系统的主要区

别, 对于随机系统, 随着 m 的增大, D_m 也趋于无穷大. 而对于混沌系统, m 增大到一定值以后, D_m 达到饱和, 这表明存在混沌吸引子. 吸引子维数能够很好地度量平均负载运动的复杂性, 通过对吸引子维数分析, 确定了相空间重构的嵌入维数, 识别出动力系统的混沌特性, 这是混沌时间序列预测的前提.

3 最大 Lyapunov 指数预测模型

在混沌学理论中, Lyapunov 指数是对相空间中初始条件不同的两条相邻的轨迹随时间推移按指数规律分离的比率的定量描述, 这决定了混沌时间序列的短期预测是可行的. 随着混沌科学的发展, 从时间序列提取最大 Lyapunov 指数的方法已经很多, 其中小数据量方法操作方便, 计算量小, 对小量数据比较可靠等优点, 所以采用小数据量算法来计算平均负载时间序列的最大 Lyapunov 指数, 结合前面的分析得到求解步骤为:

(1) 对平均负载时间序列 $(x_1, x_2, \dots, x_{n-1}, x_n)$ 进行 FFT 变换, 求出平均周期 P , 由 C-C 算法计算出时间延迟 τ ;

(2) 计算关联维数 D , 并确定嵌入维数 m , 与上步求出的时间延迟 τ 重构相空间 $\{Y_j, j = 1, 2, \dots, M\}$;

(3) 找相空间中每个点 Y_j 的最近邻点 Y_i , 并限制短暂分离, 即: $d_j(0) = \min_i \|Y_j - Y_i\|, |j-i| > p$;

(4) 对相空间中每个点 Y_j , 计算出 k 个离散时间步后的距离 $d_j(k)$, 即 $d_j(k) = \|Y_{j+k} - Y_{j+k}\|$, 其中 $k = 1, 2, \dots, \min(M-j, M-i)$;

(5) 对每个 k , 求出所有 j 的 $\ln d_j(k)$ 平均 $y(k)$, 即 $y(k) = \frac{1}{q\Delta t} \sum_{j=1}^q \ln d_j(k)$. 其中 q 是非零 $\ln d_j(k)$ 的数目, 并用最小二乘法作出回归直线, 该直线的斜率就是最大 Lyapunov 指数.

经过 FFT 变换得到平均周期, 依据计算得到的时间延迟与嵌入维数对平均负载时间序列进行相空间重构, 应用该方法求解 Lyapunov 指数的过程如图 1, 得到最大 Lyapunov 指数为 0.0244.

Lyapunov 指数不仅刻画了耗散系统相空间中相体积收缩、膨胀过程中几何特征变化, 而且对初始轨道的指数发散和估计进行了量化, 是一个很好的预报参数, 下面建立基于平均负载最大 Lyapunov 指数的预测模型.

对于观测平均负载时间序列 $x_1, x_2, \dots, x_i, \dots$,

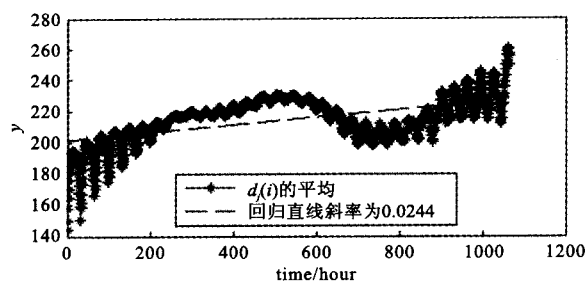


图1 小量数据法求 Lyapunov 指数过程

x_n , 假如需要预测 x_{n+k} , 则取相空间中一点 $X_p, X_p = (x_{n-(m-1)\tau}, x_{n+1-(m-1)\tau}, \dots, x_n)$, 其中 $p = n - (m-1)\tau$.

为预报中心点, 设 X_p 的最近邻点为 X_l , 则 $X_l \in \{X_1, X_2, \dots, X_{p-1}\}$, X_p 与 X_l 的距离为 d , 则有 $d = \min_j \|X_p - X_j\| = \|X_p - X_l\|$, 确定 X_p 的最近邻点 X_l 之后, 根据 X_p 和 X_l 的进一步演化 X_{p+1} 和 X_{l+1} , 有下式成立:

$$\|X_p - X_{p+1}\| = \|X_l - X_{l+1}\| e^{\lambda_1} \quad (3)$$

式中, λ_1 为序列 $x_1, x_2, \dots, x_i, \dots, x_n$ 的最大 Lyapunov 指数. 只有 X_{p+1} 的最后一个分量 x_{n+1} 未知, 故 x_{n+1} 是可预报的, 如果要对 x_{n+k} 进行预测, 解决了一步预测, k 步预测可以类推. 式(3)就是最大 Lyapunov 指数 λ_1 的预报模型, 模型的可预测时间范围通过 $1/\lambda_1$ 来确定, 由求出的最大 Lyapunov 指数, 算得预测时间为 $41d$.

4 实验结果分析

4.1 数据采集与混沌性分析

文中以实验室自主开发的模拟法庭系统为实验环境, 该系统是一个采用 J2EE 标准, Linux 操作系统平台, Apache 为服务器的 Web 服务器软件. 服务器经过很长一段时间的正常运行, 利用 Linux 服务器程序记录服务器端的平均负载 avgload (一分钟的平均负载) 的变化情况, 每隔一小时检测一次, 共获取 2408 个采样点, 见图 2.

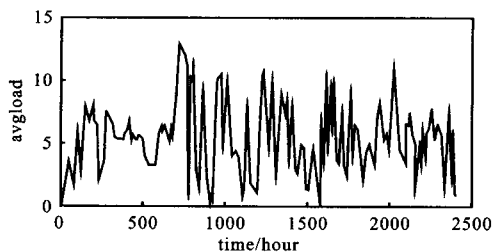


图2 平均负载样本数据曲线

对平均负载时间序列进行相空间重构, 实验选用 C-C 方法, 因为具有抗噪能力, 在实际计算中工

作的比较好, 用 Matlab 程序实现 C-C 算法, 得到时间延迟 $\tau=33$. 然后根据 G-P 算法, 当直线部分的斜率随着嵌入维数的增加不再变化时, 其斜率就是吸引子维数. 根据 $\tau=33$, 取 $m=2, 4, 6, 8, 9, 10, 11$ 时, 用 Matlab 程序实现 G-P 算法, 用平均负载时间序列做出了 $C_m(m, r)$ 随 $\ln r$ 的变化曲线, 如图 3 所示. 当 m 慢慢增大时, 曲线的直线斜率变化减小, 且开始逐渐靠拢. 当 $m \geq 8$ 时, 各条直线开始平行, 斜率趋于 $D=2.554$. 因此可以认为嵌入维数 $m=8$ 时, 吸引子维数 $D=2.554$, 充分说明为混沌系统.

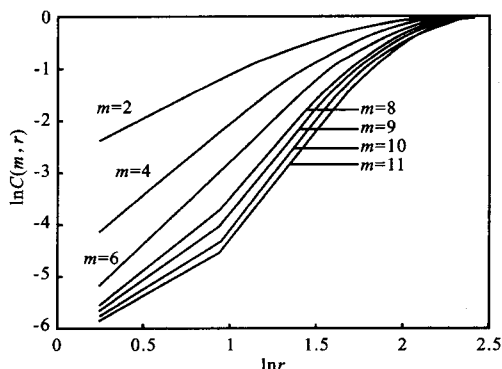


图3 平均负载时间序列 lnC-lnr 图

4.2 基于平均负载多步预测的抗衰应用

根据上述理论及结果分析, 依据平均负载预测模型, 利用平均负载时间序列的前 2367 个样本数据进行模型检验, 对第 2367 个样本以后 41 个数据值进行预测. 文中取平均相对误差 MAPE 作为预测效果的判断依据, 即

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left[\frac{|A(i) - F(i)|}{A(i)} \right] \times 100\% \quad (4)$$

式中, $A(i)$ 和 $F(i)$ 分别表示实际平均负载和预测平均负载结果, n 表示预测数据的个数.

图 4 显示了预测值与实际检测值比较, 计算出预测数据的平均相对误差为 10.04%, 误差小于 15% 的占 81%, 而误差小于 10% 占 71%. 图 5 显示了每个预测数据相对误差情况, 由此得到预测模型的预测误差低, 具有相当好的预测精度, 为服务器老化分析提供了基础.

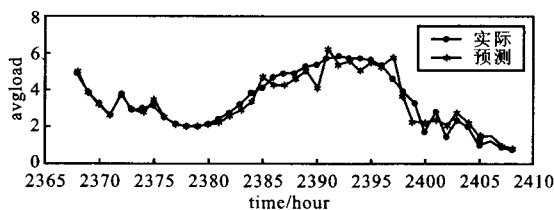


图4 预测数据与实际数据比较

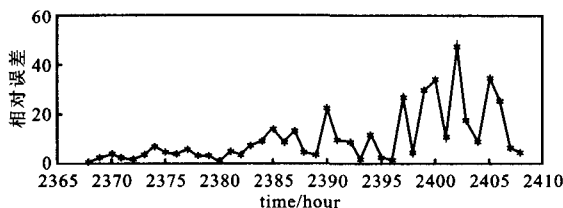


图5 预测数据相对误差

从服务器本身来说,服务器过载的情况分为两种:(1)瞬间过载,这种情况主要是由服务器负载的特点引起;(2)长时间的超载,这种情况一般是由某些特殊因素引起,可能有内存溢出、未释放的文件锁、交换空间不足、错误累积、数据破坏或多种情况的综合等。在服务器老化分析中,前一种不能标志服务器老化,只有后者才有可能出现服务器老化。

利用平均负载的预测数据定义服务器老化评价标准^[5]。针对本服务器硬件,经过观察 Web 服务器长时间运行,定义平均负载超过 5.5 时为瞬间过载比较合适。统计服务器从健康运行到性能衰退过程中平均负载的变化情况,定义 80% 预测数据超过这一阈值为服务器老化的评价指标。对于没有超过这一指标,但平均负载也一直比较高,这也说明出现了一定程度上的可用资源减少,可以不中止 Web 软件,而是通过平缓方式释放内存。而上面实验的预测结果中,预测数据超过定义阈值只占 29.5%,服务器没有出现老化迹象,所以实验时并没有采取相应的软件恢复手段。

不同的服务器根据实际情况可以定义不同的阈值与不同服务器老化评价指标,随着 Web 软件的长时间运行,服务器系统可用的资源逐渐减少,平均负载呈现出缓慢上升的趋势,可以持续检测平均负载的变化,跟踪预测,超过设定指标,就对软件进行中止、重启、或状态载入。

5 结束语

文中利用预测模型对系统老化的关键指标——平均负载进行预测,在可预测的时间内保持较高的预测精度,为服务器老化分析提供依据,可应用于大

型 Web 服务器系统老化估算。假如系统的平均负载在一段时间内一直处在某一阈值以上时,服务器易于宕机的状态,应该对 Web 软件采取相应的抗衰措施,保证服务器正常运行。将混沌理论运用到软件抗衰中还处于起步阶段,有很多问题值得探讨,首先平均负载的预测时间有限,只依靠这一个参数有限时间预测不能完全说明软件运行的全部特征;此外,系统老化也不能由一个参数完全决定。所以今后可以考虑用多变量时间序列分析服务器性能衰退,可能会得到更好的效果。

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] 张洋, 陈文波, 李廉. 基于 SCMSWEB 的网格资源监控研究与实现[J]. 微电子学与计算机, 2008, 25(10): 166-168.
- [3] Jia Yunfei, Zhao Lei, Cai Kaiyuan. A nonlinear approach to modeling of software aging in a web server [C]// 2008 15th Asia - Pacific software engineering conference. USA: IEEE Computer Society, 2008: 77-84.
- [4] Grassberger P, Procaccia I. Measuring the strangeness of strange attractors [J]. Physica D, 1983, 9(1/2): 189-208.
- [5] 鲜学丰, 方巍, 赵朋朋. 一种 Deep Web 数据源质量评估模型[J]. 微电子学与计算机, 2008, 25(10): 47-50.

作者简介:

葛君伟 男, (1961-), 博士, 教授. 研究方向为软件工程.

陈善凤 男, (1983-), 硕士. 研究方向为软件方法学、软件抗衰、混沌理论.

方义秋 女, (1963-), 硕士, 副教授. 研究方向为软件工程、空间信息集成研究、GIS 中间件.