

一种基于多重分形的软件衰退分析方法

徐 建^{1,2} 许满武² 严 悍¹ 李千目^{1,2}

(南京大学计算机软件新技术国家重点实验室 南京 210093)¹

(南京理工大学计算机科学与技术学院 南京 210094)²

摘 要 为了提高软件衰退预测的精度,采用了多重分形分析方法,以系统资源参数时间序列为研究对象,提出了一种定性和定量相结合的分析方法,用以研究其波动规律。定性分析阶段,借鉴分形理论分析影响软件性能的系统资源参数,揭示参数的波动具有分形特性;且其多重分形谱特征能刻画系统运行过程中随时间变化的情况。定量预测阶段,提出了一种多维的 Hölder 指数计算方法,用于计算多个资源参数序列的 Hölder 指数,并采用自回归移动平均模型(ARMA)预测 Hölder 指数。最后进行了实证分析,结果表明,该方法具有较好的定性分析和定量预测能力。

关键词 软件衰退,多重分形,预测

中图法分类号 TP302.7 **文献标识码** A

Multi-fractal Based Methodology for Software Aging

XU Jian^{1,2} XU Man-wu² YAN Han¹ LI Qian-mu^{1,2}

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)¹

(Institute of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)²

Abstract This paper discussed a multi-fractal based method to analyze the fluctuation of the parameters of system resource, and proposed a new methodology which combined qualitative analysis with quantitative analysis to predict resource consumption and the trend of software aging. Firstly, this study used fractal theory to discuss the fractal structure of the parameters of system resources that influenced the performance of software system. And the results show that the variations of the parameters are not a stochastic process, but have characteristic properties of fractal. In addition, the characteristics of the spectra can be used to analyze the changes of system parameters during the running time qualitatively. Secondly, this paper put forward a new methodology for calculating multidimensional exponent, which is applied to data of system resource usage. Thirdly, the Auto-Regressive Moving-Average (ARMA) model was adopted in order to carry out the analysis of the multidimensional exponent and build the corresponding forecast model. Finally, the experiment was taken to calculate the multidimensional exponent of parameter series related with several memory resources using the parameters data collected from a realistic software system. The results of the experiment indicate effective ability to predict software aging.

Keywords Software aging, Multi-fractal, Prediction

1 引言

软件衰退的现象指的是一个长时间持续运行的软件系统会发生状态退化和性能降低,最终导致系统崩溃。近年来,研究人员通过对长时间持续运行的软件系统的测量与分析,发现导致软件衰退的主因是系统资源的耗尽,于是提出了一种基于度量的研究思路^[1-4]。该思路的核心思想是:在软件运行过程中,实时监测反映软件性能状况的参数,确定软件所处的性能状态,根据实时负载和抗衰成本等作出抗衰决策。文献[2-4]都采用构造模型来刻画系统衰退状况,不同之处在于模

型的选择(其中文献[2]采用统计模型,文献[3]采用时序模型,文献[4]采用 MSET)。研究表明,不同模型都有其最佳适用场景,从这个角度来说,它们并没有优劣之分。而文献[1]发现系统资源耗费数据展示了一种新的特性,即分形特性,于是采用分形几何的原理、方法来研究衰退现象,进行了初步探讨,表明运用多重分形理论来进行研究是行之有效的。

分形几何能够抽取很多用传统的方法不能获得的隐藏信息,但其建模复杂并会出现混沌现象。本文的研究对象是一组等时间间隔监测、采集得到的系统资源参数序列,具有实时性和突变性。分形几何中的 Hölder 指数揭示了序列的局部

到稿日期:2009-11-04 返修日期:2010-01-06 本文受国家自然科学基金(90718021, 60903027),江苏省自然科学基金企业博士创新项目(BK20095535),高等学校博士学科点专项科研基金(No. 20093219120024)资助。

徐 建(1979-),男,博士,讲师,CCF 会员,主要研究方向为软件工程、可信软件, E-mail: dolphin.xu@mail.njust.edu.cn; 许满武(1944-),男,博士,教授,主要研究方向为新型程序设计; 严 悍(1970-),男,博士,副教授,主要研究方向为软件工程; 李千目(1979-),男,博士,副教授,主要研究方向为软件工程、信息安全。

特征,表明序列中某点的平滑度以及当前点与前后数值间的短期相关性。系统资源耗尽是引起软件衰退的主因,而资源参数序列的剧烈动荡则是系统资源压力增强的体现,Hölder指数恰恰能反映资源参数动荡的规律。因此,借助于对Hölder指数的预测,可实现对软件性能衰退的预测;同时系统资源并非单一的,所以研究Hölder指数估算方法时需考虑多维因素。

本文的研究工作沿袭了基于度量的研究思路,提出了一种基于多重分形的定性、定量相结合的分析方法。与其他工作的不同之处在于,本文针对系统资源耗费数据具备的分形特性,把强相关的资源耗费数据的多维Hölder指数作为研究对象,而它们的研究对象是资源耗费数据。首先利用Hurst指数分析软件性能参数的整体概况;接着运用多重分形理论具体分析局部的数据,研究系统性能参数时间序列的分形特性;然后提出了一种多维Hölder指数计算算法,并应用到系统资源耗费数据,对多维Hölder指数建立ARMA模型进行定量分析,能够较全面地实现软件衰退的预测。

2 衰退分析方法

基于多重分形的软件衰退分析方法包括定性分析、定量预测两个部分,即对系统资源参数进行定性分析,对资源参数对应的多维Hölder指数进行定量预测。如图1所示,括号内表示每个子过程中采用的方法。

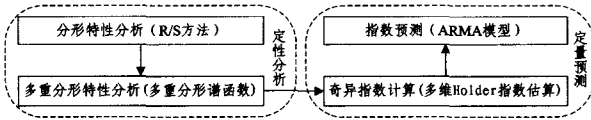


图1 基于多重分形的软件衰退分析过程

2.1 定性分析方法

定性分析的目的在于验证所选择的资源参数序列具有分形特征,并从局部出发通过多重分形过程描述序列的整体波动特征。定性分析的过程可划分为两个阶段。

第一阶段是采用重标极差(R/S)分析法对系统资源耗费参数序列的统计结构进行概要性的研究,利用Hurst指数不同取值时的不同分辨特性,初步分析系统性能参数的时间序列,判断其整体状态上的随机性和分形特性。选择R/S分析方法的原因是R/S对于识别数据长期相关性有着独特的优越性,能够从分形时间序列中区分出随机序列和非随机序列,辨别出一种介于随机结构与确定结构之间的分形结构。具体方法是将定期监测采集到的系统性能参数,作为原始数据进行分析,计算出重标极差序列及其对应的时间增量序列,然后运用最小二乘回归分析方法估计出 h (Hurst指数)。

第二阶段采用多重分形谱刻画奇异指数的变化规律,从局部出发来定性分析性能参数序列的整体波动特征。由于系统的实时动态性特性,仅从单分形角度来分析系统参数的特性是不够的。而多重分形过程是具有随着时间变化的奇异指数函数的函数,通过尺度函数和多重分形谱来刻画时间序列的局部分形特征,能够更加真实地描述变化事物的复杂统计特性,刻画分布在子集上的具有不同标度和标度指数的分形子集的局部标度性。具体方法是将采集的系统性能参数时间序列作为原始样本,计算对应的归一化值、分割函数 $S_q(\delta)$,求

出不同给定权值 q 下 $\log S_q(\delta) \sim \log \delta$ 的曲线簇。根据其是否具备了良好的线性特性,作为判断样本序列是否具有多重分形特性的依据。在此基础上,分析该序列对应的多重分形谱 $f(\alpha)$ 的分布,定性地分析参数序列随时间变化的特征,包括序列的振幅变化以及波动趋势。

2.2 定量分析方法

定量分析过程是利用多维Hölder指数计算方法获取系统资源耗费参数序列的多维Hölder指数,采用ARMA模型分析多维Hölder指数,并给出相应的预测。

函数 $f(t)$ 在时间点 t 的Hölder指数可由下式计算:

$$H_f(t) = \liminf_{h \rightarrow 0} \frac{\log(|f(t+h) - f(t)|)}{\log(|h|)} \quad (1)$$

前提是 $f(t)$ 是随时间变化的连续函数,而本文的研究对象是等时间间隔采集的系统性能参数,是离散信号,显然不满足这一要求。且考虑到导致软件性能衰退的资源参数不止一个,所以选取具有强相关的 p 个参数,作为分析对象,提出了一种多维的Hölder指数估算方法。假设导致软件衰退的关键参数有 p 个,那么某一时刻的观测数据构成一个 p 维空间的点 $\{g^p\}$,算法如下:

(1)若 $p=1$,则执行步骤(2);当 $p \geq 2$ 时,分析 p 个参数的相关性,采用积差法计算相关系数,选择强相关的 q 个参数, $p=q$;

(2)初始化参数窗口尺寸 w 以及加权回归系数 λ ;

(3)计算窗口范围内的数值与第 i 个时间点的对数差异值 $(R_{i,k})$:

$$R_{i,k} = \begin{cases} \frac{\log D_{i,k}}{\log(|k|/N)}, & D_{i,k} \neq 0 \\ 1, & D_{i,k} = 0 \end{cases} \quad (2)$$

式中, $D_{i,k} = \|g_k - g_i\| =$

$\sqrt{(g_k^1 - g_i^1)^2 + (g_k^2 - g_i^2)^2 + \dots + (g_k^p - g_i^p)^2}$; k 是 $-w \leq k \leq w$, $k \neq 0$ 且 $0 < i+k \leq n$ 条件内的任意一个值;

(4)找出窗口范围内差异的最小值,记 $t = \max\{k, w\}$,对于每一个 j , $1 \leq j \leq t$,计算 $h_{i,j} = \min\{R_{i,p} : |p| \leq j\}$;

(5)利用步骤(4)求出的近似值 $h_{i,j}$,采用加权平均值的方法求出计算第 i 个数据对应的Hölder指数 H_i ,且满足越靠近的数据应当具有较大的权值、距离较远的数据具有较小的权值:

$$H_i = \frac{1-\lambda}{1-\lambda^t} (h_{i,1} + \lambda h_{i,2} + \lambda^2 h_{i,3} + \dots + \lambda^{t-1} h_{i,t}) \quad (k=t) \quad (3)$$

上述算法的有效性已在文献[6]中进行了验证,限于篇幅,这里不再重复。

3 实验与分析

实验环境设置如下:在局域网内有一台运行Tomcat 5.5.4的服务器和一台产生负载的客户机。服务器使用Linux操作系统,所有的系统信息都保存在虚拟文件系统/proc下,使用自行开发的监控工具PERFMON来周期性地提取/proc下的信息。客户端使用开源项目JMeter作为服务器的负载生成器以及性能监控器。JMeter提供的服务器性能指标包括响应率、响应时间等。采样间隔设置为5min,持续7天,每个参数共计2016个样本点,期间并没有出现宕机事件。由于采集的系统性能参数的数量级较大,且量纲不统一,为了研究过程

中数据处理的方便,对所有数据均进行了归一化处理。

应用多维霍尔德指数进行软件衰退定量分析时,资源参数的选取需满足:1)参数序列具备分形特征,即满足式(1);2)参数之间必须有较强的关联性。鉴于上述约束条件,选取使用的物理内存空间(used Physical Memory)、空闲的交换空间(free Swap Space)作为研究对象。两者的分形特性将在下文予以验证,这里先验证两者是强相关的。两参数的相关系数采用积差法计算,结果如表1所列。表明它们是强相关的,符合研究需要。

表1 使用的内存空间和空闲的交换空间的相关系数 $|r|$

	Used Physical Memory	Free Swap Space
Used Physical Memory	1	0.637
Free Swap Space	0.637	1

3.1 Hurst 指数分析

以使用的物理内存空间的第一天样本为例,研究该参数序列对应的 Hurst 指数,验证其分形特征。具体过程如下:先计算出重标极差序列及其对应的时间增量序列,其中时间增量 $n=3,4,\dots,144$,而后采用最小二乘拟合函数进行拟合,结果如图2所示。图中 h 表现为其斜率,所得解释变量的系数即为所求的 Hurst 指数。Hurst 值(0.756)是大于0.5的,说明服务器在运行过程中使用的物理内存空间的变化情况绝非单纯地随机游走,其前后事件存在着长期相关性,具有分形特征。

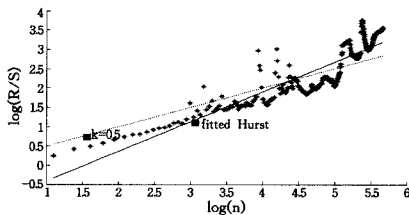


图2 第1天使用的物理内存 $\log(R/S) \sim \log(n)$ 图

用类似的方法研究一周内的两个资源参数序列,结果如表2所列。可以得出,不管对于同一参数不同天内的数据,还是同一天内不同参数的数据,其 Hurst 指数均满足 $0.5 < h < 1$,具有分形特征。

表2 一周内使用物理内存空间、空闲的交换区的 Hurst 指数值

	1	2	3	4	5	6	7
Used Physical Mem.	0.756	0.664	0.633	0.641	0.710	0.966	0.965
Free Swap Space	0.623	0.587	0.708	0.681	0.814	0.989	0.988

3.2 多重分形分析

以使用的物理内存空间7天样本为例,将一天的数据作为一个单元(每单元有 24×12 个数据),研究该参数序列的多重分形特性。若条件具备的话,通过其多重分形谱分析该序列的波动情况。

先分析第1单元的数据, δ 选取值为 $(1/288, 1/144, 1/96, 1/72, 1/36, 1/32, 1/18, 1/16, 1/12, 1/3, 1/2, 1)$;对应的尺度窗口大小为 $(1, 2, 3, 4, 8, 9, 16, 18, 24, 96, 144, 288)$;对于选取的 q 值 $(-80, -40, -20, -10, -8, -2, 0, 2, 8, 10, 20, 40, 80)$,可计算出相应的分割函数的值 $S_q(\delta)$,得到 q 取不同值时 $\log S_q(\delta) \sim \log \delta$ 曲线簇,如图3所示。无论 q 取任何值,图上每条曲线均具备线性关系,表现出良好的标度不变性,由此可知该序列具有多重分形特征。分析这两个参数7天内的任意

一个单元,均可得到同样的结论。

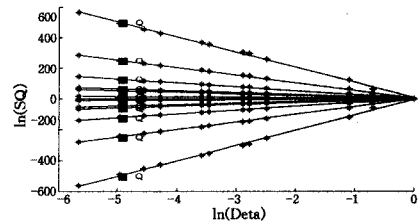


图3 使用的物理内存空间的 $\log S_q(\delta) \sim \log \delta$ 图

上述结论为下文利用多重分形理论来分析系统参数,以及用奇异指数来预测系统资源消耗趋势提供了有力的依据。仍然采用相同的设置,对每个单元内的数据计算多重分形谱,比较各自多重分形谱的变化,选取能充分反映序列波动的振幅和变化趋势的两个指标:取值范围的宽窄 $\Delta\alpha = \alpha_{\max} - \alpha_{\min}$, $f(\alpha)$ 峰值点 $\alpha = \alpha_0$ 的位置。这里仅选取具有典型意义的第1, 2, 7天样本的多重分形谱进行举例分析,如图4所示。

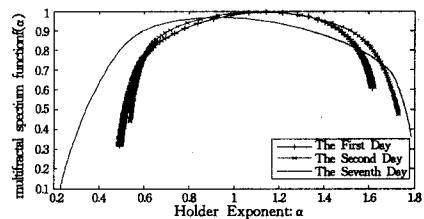


图4 第1,2,7天使用的物理内存空间多重分形谱

为方便描述,设定3个单元的谱宽依次为 $\Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3$,峰值位置分别为 $\alpha_1, \alpha_2, \alpha_3$ 。由图4可知, $\Delta\alpha_1 \approx 1.62 - 0.5 = 1.12$, $\Delta\alpha_2 \approx 1.75 - 0.55 = 1.2$, $\Delta\alpha_3 \approx 1.79 - 0.24 = 1.55$,在误差允许范围内满足 $\Delta\alpha_1 < \Delta\alpha_2 < \Delta\alpha_3$,即多重分形谱的开口宽度是在逐渐增大的。这说明参数序列变得不均匀,波动的幅度有所增强,与实际应用中系统随着运行时间的增长将会出现不稳定的状态是吻合的。另外,还可以看出, $1.1 < \alpha_1 < 1.2$ 且比较偏向1.2, $\alpha_2 \approx 1.1$ 且偏向1.1, $0.8 < \alpha_3 < 1.0$,即多重分形谱的峰值位置逐渐左移,说明分形谱左端区间渐渐强势,即归一化值较高的事件在逐渐起主导作用,此时参数序列值有上扬趋势。这与系统实际运行中使用的物理内存空间资源应当不断增长是相一致的。

3.3 基于多维 Hölder 指数的预测分析

本节将采用多维 Hölder 指数估算方法来计算系统资源参数的 Hölder 指数情况分布,揭示资源参数序列与其多维 Hölder 指数分布之间的关系,并采用自回归滑动平均模型(ARMA)进行预测分析。

图5(a)和图5(c)分别表示资源参数的归一化序列,图5(b)和图5(d)是对应的 Hölder 指数分布。可以看出,系统运行过程中使用的物理内存空间序列波动幅度较为剧烈,且波动较强时,其 Hölder 指数有明显下跌迹象(图中400和1200处竖线标识的区间表现尤为明显)。这验证了我们的假设,即 Hölder 指数能很好地反映数据波动的剧烈程度。图5(e)表示根据资源参数的多维 Hölder 指数,可以看出其波动性与系统资源参数的动荡情况吻合。具体表现为资源参数序列变化较大时,多维 Hölder 指数有明显的跌落现象,如在第400和1200个时间间隔处;此外,其分布与使用的物理内存空间的 Hölder 指数相似,说明物理内存资源的变化情况对多维 Hölder 指数的分布影响较大,同时,空闲的交换空间的变化

也对其产生了一定的影响,比如,在第 1200 个时间间隔处,多维 Hölder 指数的值就受到空闲的交换空间明显的影响。因此,如果能预测多维 Hölder 指数,便能够有效地预测系统资源动荡的情况,从而达到预测软件衰退的目的。

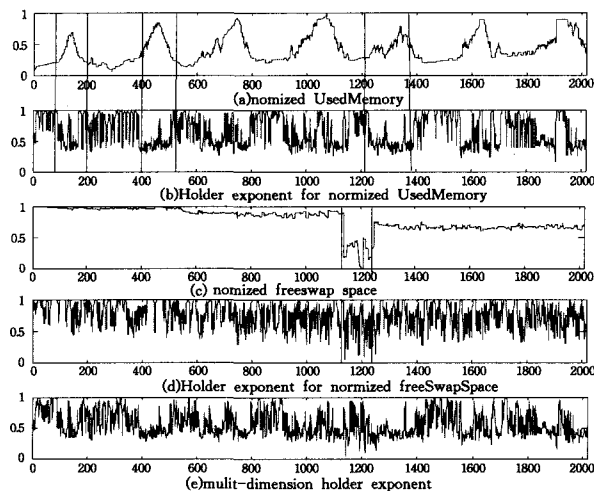


图 5 归一化参数序列及其对应的 Hölder 指数

ARMA 模型能够有效地分析平稳性数据序列的相关性。下面将利用 ARMA 对多维 Hölder 指数进行预测分析。首先建立 ARMA 模型,为了减少计算量,这里采用文献[5]中的 ARMA(2,1)模型来进行预测。由最小二乘估计方法解出相应的参数值: $\phi_1=0.52797, \phi_2=0.33075, \theta_1=0.11287, \delta_\epsilon^2(P)=0.44256$ 。满足 $\phi_1+\phi_2<1, \phi_2-\phi_1<1, |\phi_2|<1$, 根据平稳性判定准则,可判定它是一个平稳时间序列,说明前面的判断是正确的,于是拟合的 ARMA(2,1)模型为:

$$H_t = 0.52797H_{t-1} + 0.33075H_{t-2} + \epsilon_t - 0.11287\epsilon_{t-1} \quad (4)$$

针对上述拟合模型,利用模型递推法来实现 ARMA(2,1)模型 1 步预测。令 $\epsilon_{t-3}=0$, 那么如果要想实现某点的 1 步预测值,只要知道之前的 4 个观测值即可:

$$H_t(1) = 0.52797H_{t-1} + 0.33075H_{t-2} - 0.11287\epsilon_t \quad (5)$$

由于 ARMA 为短相关模型,因此 1 步预测的精度优于多步预测。我们利用前 7 天的资源参数数据的多维 Hölder 指数,对第 8 天第一个小时内的指数进行 1 步预测。如图 6 所示,可以看出,指数的值呈下降趋势,表示对应的系统资源参数的动荡指数将会加剧,这与先前的分析以及实际应用中系统资源随着长时间的持续运行会出现耗尽的情况是一致的。若对指数设定阈值,当预测指数将下降,而且将在可预见时间

内达到指定阈值时,能够提前给出软件衰退预警,以表明系统出现资源耗尽的征兆,需要实施预防性的软件自愈操作。至此达到了预测软件衰退的目的。

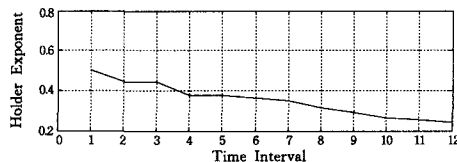


图 6 第 8 天第一个小时多维 Hölder 指数的预测值

结束语 本文给出了一种基于多重分形分析、定性与定量相结合分析软件衰退的方法。通过实证分析说明了与系统内存资源相关的两个参数(使用的物理内存、空闲的交换空间)是具有多重分形特性的,而且系统参数多重分形谱分布为认识参数的发展趋势提供了有价值的信息。同时,采用 ARMA(2,1)模型建模两个参数对应的多维 Hölder 指数,能够对多维 Hölder 指数短期内的数值进行较准确预测,能够估算短期内到达警戒阈值的时间,满足软件衰退预测需求,为制定软件恢复策略提供依据。

参考文献

(上接第 171 页)

- [9] NVIDIA CUDA[OL]. <http://developer.nvidia.com/object/cuda.html>
- [10] Vuduc R W, Moon H J. Fast sparse matrix vector multiplication by exploiting variable block structure[C]// Proceedings of the International Conference on High-performance Computing and Communications. LNCS v3726. Sorrento, Italy, 2005: 807-816
- [11] Strout M M, Carter L, Ferrante J. Compile-time composition of run-time data and iteration reorderings[J]. ACM SIGPLAN Notices, 2003, 38(5): 91-102
- [12] Buatois L, Gaumon G, Levy B. Concurrent Number Cruncher: An efficient sparse linear solver on the GPU[C]// High Per-

- formance Computation Conference (HPCC). Spring Lecture Notes in Computer Sciences, LNCS v4282. 2007: 358-371
- [13] Baskaran M M, Bordawekar R. Optimizing sparse matrix-vector multiplication on GPUs [R]. The Ohio State University, Columbus, USA and IBM TJ Watson Research Center Hawthorne, NY, USA, 2009
- [14] A visual repository of test data for use in comparative studies of algorithms for numerical linear algebra[OL]. <ftp://math.nist.gov/pub/MatrixMarket>, 2007
- [15] Williams S, Oliker L, Vuduc R, et al. Optimization of sparse matrix-vector multiplication on emerging multicore platforms[J]. Parallel Computing, 2009, 35(3): 178-194