

自相似网络流量模拟的分布式系统的设计与实现

张华川, 田 杰, 许 静
(南开大学信息技术科学学院, 天津 300071)

摘 要: 随着网络应用的发展,网络流量不断加大,对于网络应用系统的测试也占有越来越重要的地位. 良好的网络流量模型能较为真实的反映网络的情况,并能给与应用系统更为全面的测试. 根据统计学的结果表明,网络流量在时间上具有自相似性. 因此,对自相似流量模型的研究十分重要. 为了克服单机生成网络流量的不足,本文对于分形高斯噪声(FGN)模型的算法进行了改进,使其能够并行且高速的生成自相似样本. 文中针对改进后的算法提出了分布式系统,并阐述了系统的整体结构和控制机制的设计与实现. 本文的工作还包括实现了一个检测网络流量的检测端,并通过它来分析分布式系统所生成的样本. 结果表明,本系统所生成的网络流量具有自相似性.

关键词: 自相似模型; 网络流量模拟; 分布式系统

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2009) 4A-031-05

The Design and Implementation of Distributed System for Self-Similar Network Traffic Simulation

ZHANG Hua-chuan, TIAN Jie, XU Jing
(College of Information Technical Science, Nankai University, Tianjin 300071, China)

Abstract: With the development of network application, network traffic is constantly increasing, and the web application testing is becoming more important. According to statistical results, the network traffic is self-similar in the time scale. Therefore, the research of self-similar traffic model is very important. Network traffic simulation has limitation by single computer. So this paper improves the Fraction Gaussian Noise (FGN) model so that it becomes a high-speed parallel algorithm to generate self-similar traffic. Using the improved algorithm, we propose a distributed system and present the design and implementation of this system. The paper also includes the work of achieving a network traffic detection tool. The results show that the testing samples generated by the distributed system have self-similar property.

Key words: self-similar model; network traffic simulation; distributed system

1 引言

目前,随着网络技术的发展,网络的业务流量的迅速增加. 网络流量模型可以便于测试人员或研究人员模拟网络行为,监测流量^[1]. 如何构建一个更为真实的反映网络流量的模型,对于网络应用系统性能的测试与评价至关重要^[2].

传统的网络模型主要以泊松过程为依据,即认为网络流量具有短相关性^[3]. 然而,由统计学的结果表明,网络流量在时间尺度上具有普遍的自相似性(长相关性)^[4]. 因此,自相似过程模型成为网络流量研究领域的一个重要课题. 目前,比较经典的自相似模型有:ON/OFF模型,分形布朗运动模型(FBM),分形高斯噪声模型(FGN),FARIMA(p, d, q)模型等^[5].

本文通过对自相似模型的研究,改进了 FGN 生成自相似流量的算法,实现并行化,能够快速并准确的生成自相似样本. 同时,针对该算法实现了分布式系统,克服了单机生成样本的瓶颈. 除此之外,还设计了检测端来验证该系统所生成的样本的自相似性. 本文对自相似模型的实现方法以及在实际网络中的应用具有一定的启发作用.

本文的第二部分将介绍对于 FGN 算法的改进. 第三部分主要介绍根据改进的算法,设计并实现一个生成自相似流量的分布式系统. 第四部分描述了用于检测网络流量的检测端的设计与实现. 第五部分会根据实验数据进行估算 Hurst 参数,进而评价整个系统. 第六部分是对所作的工作的总结以及对以后的研究的展望.

2 FGN算法的改进

自从开始研究网络流量的自相似特性以来,已经出现了许多生成自相似过程的模型^[6].其中,分形布朗运动模型(FBM)生成的样本具有良好的自相似性,FGN是FBM的增量过程,同时也是一种精确的自相似过程.本文也采用FGN模型,并对算法进行改进,来加速样本的生成速度.

本文采用功率谱和快速傅立叶变换的方法来合成分形高斯噪声FGN.FGN的谱密度 $f(\cdot; H)$ 与Hurst参数之间的函数关系式为:

$$f(\cdot; H) = A(\cdot; H) \cdot [|\cdot|^{-2H-1} + B(\cdot; H)] \quad (1)$$

$$A(\cdot; H) = 2(\sin(\cdot H)) \cdot (1 - \cos(\cdot)) \cdot \int_0^{\cdot} x^{2H} e^{-x} dx \quad (2)$$

$$B(\cdot; H) = [2^{p+} - 2^{H-1} + (2^{p-} - 2^{H-1})] \quad (3)$$

其中, $0 < H < 1, 0 < p < 1$.

在式(3)中存在着一个无限求和的运算.为了加快计算速度,同时保证正确性,需要使用一种近似的方法,将式(3)改为:

$$\bar{B}(\cdot; H) = \sum_{p=1}^k (a_p^d + b_p^d) + \frac{a_k^d + b_k^d + a_{k+1}^d + b_{k+1}^d}{8H} \quad (4)$$

其中 $d = -2H - 1, d = -2H, a_k = 2k + 1, b_k = 2k + 2$.

该表达式的最大优点在于,对于一个给定的 k 值,它的求和过程能够快速展开,而且结果表达式能够针对不同的 k 进行快速合成.

接下来的工作是要生成自相似的通信量.首先要经过四个步骤.

(1) 确定Hurst参数,然后通过式(1)产生FGN过程谱密度序列 $\{f_1, f_2, \dots, f_n\}$,其中 f_i 满足 $f_i = f\left(\frac{2}{n}; H\right)$, $(1 \leq i \leq n)$.

(2) 把序列 $\{f_1, f_2, \dots, f_n\}$ 的每一个变量分别乘以一个互不相关的1均值指数随机变量random,从而产生序列 $\{u_1, u_2, \dots, u_n\}$.

(3) 根据 $\{u_1, u_2, \dots, u_n\}$ 构造一个具有随机相位的复数序列 $\{y_1, y_2, \dots, y_n\}$,其中 y_i 满足 $|y_i| = \sqrt{8i}$ 并且均匀分布在0到2之间.

(4) 由 $\{y_1, y_2, \dots, y_n\}$ 拓展构造一个序列 $\{z_1, z_2, \dots, z_n\}$,使 z_i 满足:

$$z_i = \begin{cases} 0, & i = 1 \\ y_i & 1 \leq i \leq n \\ \bar{y}_{2n+1-i} & n+1 \leq i \leq 2n \end{cases}$$

其中, \bar{y}_{2n+1-i} 是 y_{2n+1-i} 的共轭复数.

将序列 $\{z_1, z_2, \dots, z_n\}$ 进行逆离散傅立叶变换(IDFT)即可生成自相似样本 $\{X_1, X_2, \dots, X_n\}$,但这种串

行的方法在数据量较大时会很慢.本文中采用了基于2FFT的快速并行IDFT算法中的频域抽取法(DIF-DFT),来实现并行化.

设序列 $\{x_1, x_2, \dots, x_n\}$ 的长度为 N ,同时 $N = 2^m$,可以得到:

$$\begin{aligned} X(k) &= DFT[x(n)] = \sum_{n=0}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{kn} + \sum_{n=0}^{\frac{N}{2}-1} x(n + \frac{N}{2}) W_N^{k(n+\frac{N}{2})} \\ &= \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x(n + \frac{N}{2})] W_N^{\frac{kN}{2}} W_N^{kn} \end{aligned} \quad (5)$$

其中, $W_N = e^{j\frac{2\pi}{N}}, 0 \leq k \leq N-1$,

$$W_N^{\frac{kN}{2}} = (-1)^k = \begin{cases} -1, & k \text{ 为奇数} \\ 1, & k \text{ 为偶数} \end{cases}$$

由此可见, $X(k)$ 分为奇偶两部分,可以得到式(6),式(7).

当 k 为奇数时,得

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) - x(n + \frac{N}{2})] W_N^{\frac{kN}{2}} \cdot W_N^{kn} \quad (6)$$

当 k 为偶数时,得

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x(n + \frac{N}{2})] W_N^{\frac{kN}{2}} \cdot W_N^{kn} \quad (7)$$

可设 $i = 0, 1, \dots, \frac{N}{2} - 1$,并且令

$$x_1(n) = x(n) + x(n + \frac{N}{2}), n = 0, 1, 2, \dots, N-1 \quad (8)$$

$$x_2(n) = [x(n) - x(n + \frac{N}{2})] W_N^n, n = 0, 1, 2, \dots, N-1 \quad (9)$$

通过式(6)和式(7)可以得到

$$X(2i) = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_N^{in} = DFT[x_1(n)] \quad (10)$$

$$X(2i+1) = \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_N^{in} = DFT[x_2(n)] \quad (11)$$

式(10)和(11)可用蝶形运算表示,如图1.

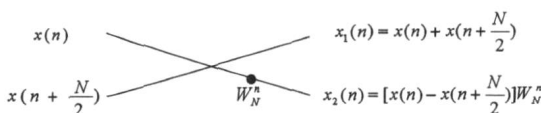


图1 蝶形运算图

旋转因子 W_N 具有对称性和周期性,因此可以减少DFT的运算次数.若 $N = 2^M$,在经过蝶形运算分解后,计算 2^{M-1} 个两点间的DFT即可,降低了计算的复杂度.同时便于通过增加并行计算的程度来提高计算的效率.

该算法在“南开之星”上进行了实现。“南开之星”是一台超级并行计算机,拥有 800 个 Intel Xeon 3.06GHz CPU 节点.实验表明,在生成 2^{16} 个自相似数据时,串行的方法需要 0.0185 秒,而通过并行算法使用四个 CUP,只需要 0.0059 秒,加速比达到了 3.14.在生成 2^{24} 数据量的时候,采用同样的方法,加速比可以达到 3.61.使用更多的 CPU 时会得到更高的加速比,当 CPU 数增加到 8 时,同样是生成 2^{24} 的数据量时,加速比可以达到 7.35.由实验结果说明,改进后的 FGN 算法在生成自相似样本的速度上有明显的提高.

3 分布式系统的设计与实现

根据实验表明,单机的测试系统在生成自相似流量时,不能够生成与真实网络环境相似的测试数据,并且会受到硬件配置的限制.在生成较大数据量,单机的不足表现得更为明显.前文中经过改进的 FGN 算法是一种并行的算法,能够应用在一个分布式的系统上实现自相似流量的生成.下面将针对 FGN 改进算法,设计并实现一个分布式系统.

3.1 分布式系统的拓扑结构

分布式系统主要分为两个部分:控制端和并行 Agent,如图 2 所示.

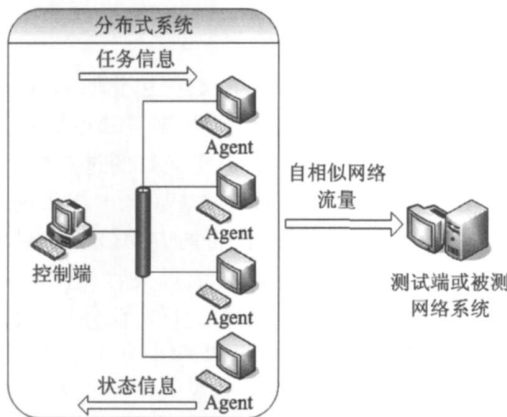


图2 系统拓扑结构图

在整个系统中,控制端和并行 Agent 分别担任不同的职责:

控制端:对每一个 Agent 进行状态监视和任务分配,负责系统整体的任务调度和控制.

并行 Agent:接受控制端发来的任务信息,采用并行算法生成自相似网络流量,并且实时返回自己的状态信息.

控制端会通过 TCP/IP 发送控制信息和任务信息到并行 Agent.并行 Agent 会根据自己所携带的程序运行.所有并行 Agent 都具有自我控制机制,并且能够持续发送状态信息到控制端.并行 Agent 具有包括等待、睡眠、从错误中恢复的功能.

在图 2 中,系统的右方是测试端或被测网络系统.测试端是用来验证这个分布式系统所生成的网络流量是否符合自相似的特性.被测网络系统是所要测试的网络应用服务,根据被测系统服务的不同,控制端会发送具有不同内容的任务信息让并行 Agent 合成有效的测试网络流量.这种测试流量能够保证在宏观统计上符合自相似模型,进而保证在长时间的测试过程中,测试数据与真实网络环境中的数据相吻合,以此来测试被测系统是否在性能上符合要求.

3.2 系统控制结构的设计与实现

为了保证在整个分布式系统流程的可靠运行,尤其是保证在生成自相似网络流量的过程中生成的网络测试的数据包的有效性和准确性.系统的控制端需要在时间上对并行 Agent 进行精确的控制,所以需要建立一种有效的控制结构.

本文提出的分布式系统采用集中式随机控制方式.集中式随机控制方式是由主控端和被控段两部分组成.其中,主控端就是系统的控制端,被控端就是并行 Agent.首先,控制端会获取一个包括所有并行 Agent 的控制列表,根据列表依次对 Agent 进行初始化的通信,并等待回应.针对每个 Agent 的回复信息,控制端会在队列中添加或更新该 Agent 对应的信息,进而来维护所有并行 Agent 的运行状态.在实际中,由于每个 Agent 的性能不同,在运行过程中,时间上会出现随机性,满足具有随机性的请求的需求.

控制端的状态主要包括检查 (CheckUp)、唤醒 (WakeUp)、计算 (CalculateOn)、同步开始 (SynOn) 以及开始 (TestBegin) 这五个状态.每个 Agent 包括睡眠 (Sleep)、等待 (Wait)、同步准备好 (SynReady)、工作中 (Working)、计算完成 (CalculateDone) 这五个状态.控制端和并行 Agent 会发送不同的控制信号来改变各自的状态.表 1 列出了控制端接收并行 Agent 的控制信息后,状态转移的规则,表 2 列出了并行 Agent 接收控制端的控制信号后,状态转移的规则.

表 1 控制端接收的信号及状态转移

控制信号	状态转移
a. Sleep	CheckUp WakeUp
a. Waiting	WakeUp CalculateOn
a. CalDone	CalculateOn SynOn
a. SynReady	SynOn TestBegin
a. Working	Do nothing
a. JobDone	TestBegin TestBegin

表 2 Agent 接收的信号及状态转移

控制信号	状态转移
c. WakeUp	Sleep Wait
c. Calculate	Wait CalculateDone
c. SynOn	CalculateDone SynReady
c. Start	SynReady Working
c. Stop	Any State Wait

控制信号是控制端与并行 Agent 进行通讯的核心内容.无论是控制端,还是并行 Agent 都要对控制信号

进行解析,然后根据目前的运行状态进行下一步的任务.也只有获得了正确的控制信号的前提下,系统才能正常的运转.如果在通讯过程中出现了错误,控制端和并行 Agent 会进入到自己的错误恢复程序,把当前的状态记录到日志中去,便于测试人员发现错误.

由于在实际运行中,由于每个 Agent 的接收的任务的不同,以及所要进行计算的数据量和机器本身的性能的不同,会在运行过程中出现不同步的现象.为了保证整个系统最终生成的网络流量具有自相似的性质,必须有一个唯一的时间标准来使所有的 Agent 能够同时发送数据.这就需要用到“等待—同步”的机制.在每个 Agent 进行计算任务后,就向控制端发送 a. SynReady 信号.当控制端接收到所有 Agent 发送的 a. SynReady 信号后,再以广播的形式向每个 Agent 发送 c. Start 信号,从而使系统开始产生自相似数据流量.

Agent 发送到控制端的通讯内容只是包括当前的控制信号.控制端发送到 Agent 端的通信内容结构中,c. SynOn 和 c. Stop 信号也只是包括控制信号.而 c. WakeUp、c. Calculate 和 c. Start 信号还应包含其它内容,包括任务信息以及相关的数据内容.

3.3 自相似模型的运用

在文章的第二部分对 FGN 算法进行改进,主要计算过程分成两个部分:一个是计算 FGN 谱密度;另一个是进行并行 DFT 计算.对 DIF-DFT 进行横向拆分,将不同的部分分配到 M 个 CPU,其中 $M=2^k$,设需要生成的自相似样本数为 $N=2^c$,那么每个 CPU 所分配到的计算数为 2^{c-k} .设 $M=4$, $N=8$ 时的 DIF-DFT 运算流程图如图 3 所示.

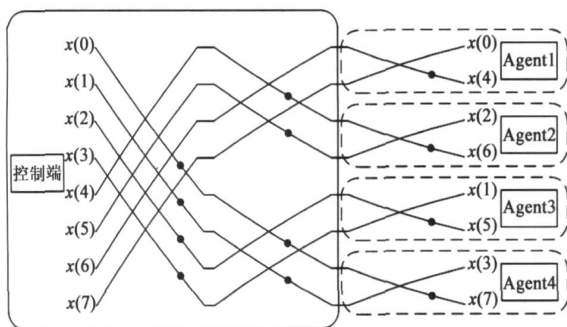


图3 四个Agent的并行计算流程图

由图3可以看出通过分割的方式,把计算部分让控制端和并行 Agent 分别进行计算.在控制端,所要做的就是进行数据的预处理,根据 Agent 数目的多少,决定预处理的步骤,然后分配给各个 Agent. Agent 端在接受到控制端处理后马上进行计算,因为计算的方法都是相同的,所以在结构上容易实现.需要注意的是在 Agent 上进行蝶形运算时,不能使用原始的旋转因子 W_N^n ,需要根据原始的旋转因子求出当前的旋转因子 $W_N^{n'}$,其中

$$W_N^{n'} = W_N^{n \cdot k}$$

若系统需要生成 2^{20} 个样本点,使用 2^2 个 Agent,那么在控制端只需要两步计算,在每个 Agent 上都需要进行 18 步计算,计算自相似的样本数均为 2^{18} 个.如此一来,大大减少了在串行过程中所需要花费的时间.因此,在生成自相似网络流量的过程中,并行自相似样本生成算法所得到的加速比完全依赖于 Agent 数目以及所需要计算的自相似样本队列的数目.并且随着所需生成的自相似样本数量越大,Agent 数目越多,此种结构设计的优越性就越加明显.当需要生成不同的网络流量时,Agent 的数量可以进行伸缩,Agent 的数量始终为 2^n 的形式.这种拓扑结构是以控制端为中心节点,Agent 为中央节点相连的子节点,形成星型的结构.这种将不同的计算部分分配到不同计算机的方法,可以解决一直以来难以生成大量自相似网络流量的问题.

4 检测端的设计与实现

为了对本系统进行有效评价,判断生成的测试数据的自相似性,以及检测系统在实际运行中的运行状态,需要设计和实现一个检测端来接收该生成的自相似数据流量.本文实现的检测端是基于 Winpcap 开发的. Winpcap (windows packet capture) 是 Windows 平台下一个免费、公共的网络访问系统,它为 Win32 应用程序提供访问网络底层的能力.

该检测端的功能主要包括:数据包获取、数据包过滤、协议分析、数据统计和分析.数据包获取和过滤是使用 Winpcap 提供的功能,根据定义好的过滤规则,审查每一个数据包.其中,过滤规则是基于 IP 报头进行制定的,包括数据包长度、源 IP、目的 IP、TCP/IP 协议、端口、ICMP 消息类型等.在获取了 IP 报头后,也就获得了协议类型等其它数据,因此需要进行协议分析.假设协议类型是 TCP,接着就可以从数据包中分离出 TCP 协议头的数据,然后从 TCP 协议头中提取相关的数据,如源端口、目的端口、数据偏移、控制位等.最后一步是要进行数据统计与分析.根据获取的数据包的数量进行计算与分析,对于一定时间内的网络数据流量进行 Hurst 参数的估计,来评判网络的自相似性.为了能够对系统进行直观、有效的评价,检测端还对网络流量进行可视性实现,将数据包的数量用图形显示出来,包括对每一

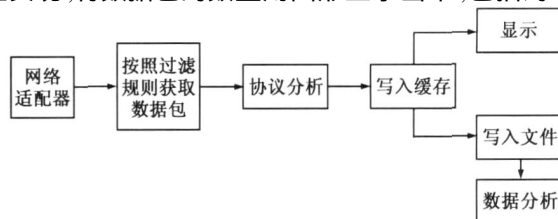


图4 测试端工作流程图

个 Agent 的监测. 除次之外,在数据统计方面,可以按照不同协议层次统计其百分比,来实现对网络流量更全面的评估. 测试端的工作流程图如图 4 所示.

5 分布式系统的评价

对于测试系统的评价,主要是通过系统所生成的网络流量来检测其是否符合自相似性,从而判定该测试系统是否成功. 在测试中选用四台计算机做 Agent 生成网络流量,检测端获取数据包后进行分析,并用估算的方法求出 Hurst 参数. 本文中采用的 Hurst 参数的估测法为 V-T(Variance-Time)图法,即方差时间图法,以及 R/S(Rescaled Range Analysis)图法,即重标度极差分析法. 测试的主要内容如下:

对于每秒生成 10^5 和 10^6 数量级样本的两种情况(合成流量时设定的 Hurst 参数为 0.85),分别对 256 秒和 512 秒的测试数据画出 V-T 图和 R/S 图,并进行 Hurst 参数估计. 实验结果如下:

(1) 10^5 数量级,256 秒:V-T 图和 R/S 图如图 5 所示,Hurst 参数估值分别为 0.8453 和 0.8521.

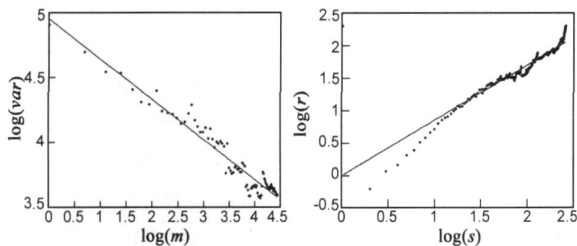


图5 10^5 数量级,256秒的V-T图和R/S图

(2) 10^5 数量级,512 秒:V-T 图和 R/S 图如图 6 所示,Hurst 参数估值分别为 0.8434 和 0.8549.

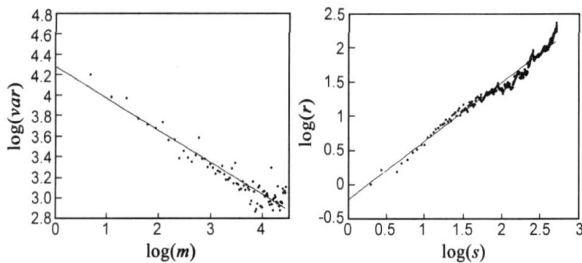


图6 10^5 数量级,512秒的V-T图和R/S图

(3) 10^6 数量级,256 秒:V-T 图和 R/S 图如图 7 所示,Hurst 参数估值分别为 0.8474 和 0.8555.

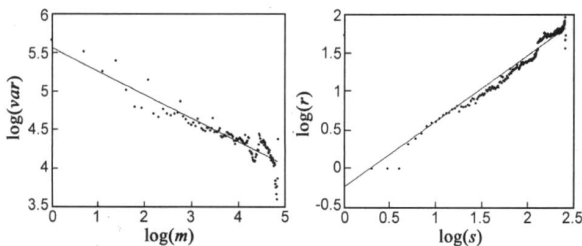


图7 10^6 数量级,256秒的V-T图和R/S图

(4) 10^6 数量级,512 秒:V-T 图和 R/S 图如图 8 所示,Hurst 参数估值分别为 0.8532 和 0.8523.

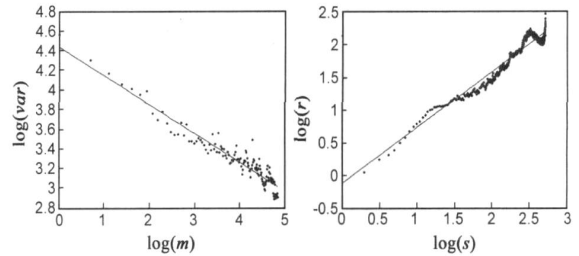


图8 10^6 数量级,512秒的V-T图和R/S图

其中,V-T 图的横纵坐标为样本的聚集过程的阶数 m ,纵坐标为 m 阶聚集过程方差的对数 $\log(\text{var})$;R/S 图的横纵坐标为在序列长度 n 下的样本的标准差的对数 $\log(s)$,纵坐标为在序列长度 n 下的样本的极差的对数 $\log(r)$.

从对于 Hurst 参数的估测结果可以看出,分析测试系统生成的网络流量所得出的 Hurst 估算值与设定值十分接近,符合自相似性质. 并且,随着生成的网络数据量越来越多,时间越来越长,测试系统所生成的数据越能体现出自相似性. 所以,该分布式系统能够满足生成自相似网络流量的要求.

6 总结与展望

本文对使用分形高斯噪声(FGN)生成自相似模型的方法进行了深入的研究,并对其进行了改进,实现并行计算,在提高速度的同时也保证了样本的质量. 通过改进的算法,设计并实现了一个分布式系统,用来生成自相似网络流量. 文中详细介绍了系统的控制机制和并行算法的应用,并利用检测端对该系统所生成的数据进行分析. 结果表明数据具有自相似性,同时也验证了系统的正确性和可用性.

本系统还需要有改进和完善的部分. 首先,系统应该不仅应用于测试网络应用系统,还要能够对一般的网络系统进行有效的测试,把本系统拓展到更低层的网络测试领域中. 除此之外,检测端的设计需要改进,使其能够满足检测和评价更大容量网络数据业务的需要. 同时还应该完善检测端对被测系统的性能参数的监控,从而使整个测试流程能够更加深入,以更好的为测试人员提供分析数据.

作者简介:



张华川 男,1985 年生于天津市,2007 年进入南开大学信息技术科学学院,现在为硕士研究生,从事软件测试方面的研究.
E-mail: cat_w_k@yahoo.com.cn

- analysis [J]. Journal of Computer and System Sciences, 2000 (61) :217 - 235.
- [10] Hayes J H, Dekhtyar A, Osborne J. Improving requirements tracing via information retrieval [A]. Proceedings 11th IEEE International on Requirements Engineering Conference [C]. Monterey, CA, USA, 2003. 138 - 147.
- [11] 王映辉, 王立福, 张世琨, 王琼芳. 一种软件需求变化追踪方法 [J]. 电子学报, 2006, 34(8) :1428 - 1432.
Wang Ying-hui, Wang Li-fu, Zhang Shi-kun, Wang Qiong-fang. A tracing approach of software requirement change [J]. Acta Electronics Sinica, 2006, 34(8) :1428 - 1432. (in Chinese)
- [12] Yu C T, Luk W S, Cheung T Y. A statistical model for relevance feedback in information retrieval [J]. Journal of the Association for Computing Machinery, 1976, 23(2) :273 - 286.
- [13] Hayes J H, Dekhtyar A, Sundaram S K. Advancing candidate link generation for requirements tracing: The study of methods [J]. IEEE Transactions on Software Engineering, 2006, 32(1) :4 - 19.

作者简介:



赖冠辉 男, 1983 年生于广东省东莞市, 现为北京航空航天大学软件工程研究所硕士生, 主要研究领域为软件工程与信息检索.

Email : garylai @sei. baa. edu. cn



王晓博 男, 1982 年生于河南平顶山, 现为北京航空航天大学软件工程研究所博士生, 主要研究领域为程序分析、软件可视化与数据挖掘等.

Email : boby @sei. buaa. edu. cn

(上接第 35 页)

参考文献:

- [1] Abry P, Veitch D. Wavelet analysis of long-range dependent traffic [J]. IEEE Trans Information Theory, 1998, 44(1) :2 - 15.
- [2] Leland W, Taqqu M, Willinger W. On the self-similar nature of Ethernet traffic (extended version) [J]. IEEE ACM Transactions on Networking, 1994, 2(1) :1 - 15.
- [3] Ge X H, Zhu G X, Zhu Y T. On the testing for alpha-stable distributions of ethernet network traffic [J]. The Journal of Electronics, 2003, 1(1) :309 - 312.
- [4] Sikdar B, Vastolak S. The effect of TCP on the self-similarity of network traffic [J]. Computer Communications and Networks, 2003, 1(1) :370 - 373.
- [5] Figueiredo D R, Liu B, Misra V. On the autocorrelation structure of TCP traffic [J]. Computer Networks, 2002, 40(3) :339 - 361.
- [6] Park K, Kim G, Ccrovellam. On the relationship between file sizes transport protocols and self-similarity network traffic [A]. Proceeding of ICNP [C]. Columbus : IEEE Press, 1996. 171 - 180.