多层服务系统选择性再生框架

田冠华[1),2)] 詹剑锋[1)] 孟 丹[1)]

[1)](中国科学院计算技术研究所国家智能计算机研究开发中心 北京 100190)

[2)](中国科学院研究生院 北京 100049)

摘 要 随着人们对网络服务的依赖性日益增强,网络服务系统的服务质量和可用性变得至关重要.然而,服务系统的日益复杂化及第三方软件和异构中间件系统的大量部署,为各种软件缺陷提供更多隐蔽空间.这对保证服务质量和系统可用性提出挑战.文中提出一个自适应的选择性再生框架,以保证系统的服务质量.选择性再生框架,把基于请求处理路径的性能故障分析诊断技术和软件再生技术相结合,通过及时诊断和再生系统中性能故障的部件,改善请求的响应时间,保证系统服务质量.文中提出了一个新的基于请求路径的性能故障的诊断方法.同时,文中实现了一个层次化的选择性的再生策略.基于 RUBiS 系统的实验结果表明文中方法可以有效保证多层服务系统的服务质量.

关键词 软件再生;服务质量;请求跟踪技术;多层服务系统;选择性再生

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2009.01938

Selective Rejuvenation Framework for Multi-Tier Services

TIAN Guan-Hua[1),2)] ZHAN Jian-Feng[1)] MENG Dan[1)]

[1)](*National Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[2)](*Graduate University of Chinese Academy of Sciences, Beijing* 100049)

Abstract Quality of service and availability are crucial for Internet service system. However, service systems become increasingly complex, and deployed with various heterogeneous middleware and commercial off-the-self software without source code. These introduce more hidden spaces for software defects, which challenge the traditional approaches for guarantee system's Service Level Objectives, or SLOs. In this paper, the authors propose an Adaptive Selective Rejuvenation Framework to guarantee SLOs in multi-tier service systems. The selective rejuvenation framework combines software rejuvenation technique with request tracing technique to identify and selectively rejuvenate the performance faulty components. The authors implement a novel request tracing technique in the rejuvenation framework, which reconstruct request causal paths precisely, and analyze request causal paths by clustering similar paths into path patterns and selecting relevant path patterns. Meanwhile, the authors implement a hierarchical rejuvenation scheme, which identify and do proactive rejuvenation of performance faulty components, before performance degradation becomes severe. The evaluations are given with RUBiS benchmark, and the results show that the request path analysis approach is effective, and selective rejuvenation framework can effectively improve system's Quality of Services.

Keywords software rejuvenation; quality of service; request tracing technique; multi-tier service system; selective rejuvenation

收稿日期:2009-07-15;最终修改稿收到日期:2009-08-31.本课题得到国家自然科学基金(60933003)、国家青年自然科学基金(60703020)、国家"八六三"高技术研究发展计划项目基金(2009AA01Z139)资助.**田冠华**,男,1980 年生,博士研究生,主要研究方向为集群系统服务质量.E-mail: tianguanhua@ncic.ac.cn. **詹剑锋**,男,1976 年生,博士,副研究员,主要研究方向为集群操作系统、分布式系统、高可靠性软件. **孟 丹**,男,1965 年生,研究员,博士生导师,主要研究领域为高性能计算机体系结构和系统软件、高效通信协议、分布式文件系统与存储服务器、机群操作系统.

1 引　言

随着网络时代的到来,人们的生活对网络服务的依赖性日益增强,系统的服务质量和可用性变得至关重要.然而,服务系统的日益复杂化及第三方软件模块和异构中间件系统的大量部署,为各种软件缺陷提供更多隐蔽空间,这对保障服务系统的服务质量和可用性提出严峻挑战.软件缺陷严重地影响着系统的服务质量和可用性.最近的一个研究表明,对于一个典型的服务系统,性能故障及宕机带来的平均损失在每小时 125000 美元以上[1].

软件再生技术是克服系统老化及大量原因不明的软件缺陷引起的性能故障的有效手段.采用主动再生的方式,系统可以通过对内在的运行状态作及时清理,消除长期积累的潜在错误和一些瞬时错误,使系统重新进入正常状态[2-3].软件再生技术简洁有效,得到广泛的应用和研究.研究人员先后引入多种数学模型来分析系统状态及演化规律,以获得系统最优的再生策略[2-5].同时也有研究人员通过检测系统的资源使用情况来预测系统的资源耗尽趋势,以主动再生的方式避免宕机带来的损失[6-8].模型分析方法基于严格的数学基础,刻画系统状态及演进规律,但缺乏对系统变化的适应能力.相反检测处理方法,对系统变化有很好的调整和适应能力,但缺少对请求在系统各部件间的相互关联性分析.本文,从保证多层服务系统上用户关注的性能参数的角度,提出一个自适应选择性再生框架.

本文的选择性再生框架,通过把基于请求路径的性能故障诊断方法和系统软件再生技术相结合,对服务系统的各层作性能故障诊断和有选择的再生.我们提出并实现了一个新的基于请求路径方式的性能故障分析诊断技术.我们把系统中各层功能模块看作是黑盒,使用各层中收集的 trace 作请求处理路径的重构,进而使用统计分析方法对请求路径作分析,实现对系统的性能故障的诊断,确定需要再生的部件.同时,为避免对由于负载变化和波动造成的系统短期性能异常作不必要的软件再生,我们实现一个动态多门限判定算法.在此基础上,我们提出并实现一个层次化的有选择的再生策略.在这个层次化结构里,底层模块使用多门限判定算法,判断部件的性能特征在统计意义上是否发生改变,以避免不必要的再生;顶层借助请求的依赖关系选择最需要作再生的部件.我们基于 RUBiS 平台做了故障注入实验,证明本文框架可以正确判定和再生性能故障的部件,进而保证服务质量.同时,我们的请求处理路径方法引入的性能损失可以忽略.

2 请求跟踪技术

我们提出的选择性再生框架,通过基于请求处理路径的性能故障诊断分析技术,确定性能异常点,进而通过实现对多层服务系统的选择性再生,改善请求响应时间,保证服务系统的服务质量.

2.1 请求跟踪技术

在多层服务系统里,外部请求通常会在各层功能模块之间引起一系列的子请求序列.外部请求和子请求之间存在因果关系.请求跟踪技术正是利用这种请求在分布式系统里处理过程中存在的因果关系,实现请求路径的重构.由于当前的服务系统大多是对请求采取并发处理的方式,请求路径构建需要对并发处理的请求作区分隔离以重构路径.为此研究人员提出多种处理方法,或是通过修改系统处理模块,为每个请求添加逻辑号,来区分并发的请求(如文献[9]);或是通过信号处理方法,把请求的延迟时间作统计处理,进而把并发请求重新关联成请求处理路径(如文献[10]).前一种方法需要有系统实现的内在信息,涉及对系统的修改,对于大量异构中间件和第三方模块有局限性;后一种方法把系统看作黑盒,避免上面的局限,但对请求路径的构建依赖统计方法处理,精确程度受到影响.

当前的服务系统,普遍采用进程池(或线程池)系统架构,通过消息交互方式在多层系统间交换消息.针对这种系统架构,我们把系统里各层功能模块当作黑盒处理,提出实现一种精确的关联方法.我们的方法不涉及对服务系统的修改,而只收集各层功能模块之间的交互消息,包括请求的 begin、end、send 和 receive 等类型的消息,使用交互信息之间内在的因果关系来重构请求路径.我们收集的信息包括:进程上下文和消息上下文信息.进程上下文中用于路径构建的主要信息是进程号,而消息上下文中包括目标地址和源地址 IP 及两者的端口号、时间戳等.我们的精确关联方法包括两方面的关联:进程上下文的关联和消息上下文的关联.进程上下文的关联,是把同一时间窗口里同一进程号的 receive 和 send 消息对作关联,消息上下文关联是把不同的进程号的目标地址和源地址 IP 及端口号对应的 send 和 receive 消息对作关联.这样把请求的整个路径关

联起来.例如在一个三层服务系统里,请求处理路径的关联结果如图1,其中 $M^{p}_{s,d}$ 表示消息的进程上下文 p、消息源 IP 及端口号为 s、目标 IP 及端口号为 d;B、E、S 和 R 分别代表消息类型 begin、end、send 和 receive,其中 $R^{j1}_{m,j}$ 和 $S^{j1}_{j,a}$ 是通过进程上下文(即进程号 $j1$)作关联,$S^{j1}_{j,a}$ 和 $R^{a1}_{j,a}$ 通过消息上下文(即源地址 j 和目标地址 a)作关联.这样一条请求路径的关联结果可以表示为一个消息序列,例如,图1中,请求1(request1)的路径关联结果可以表示为 $B^{a1}_{c,a}-S^{a1}_{a,j}-R^{j1}_{a,j}-S^{j1}_{j,m}-R^{m1}_{j,m}-S^{m1}_{m,j}-R^{j1}_{m,j}-S^{j1}_{j,a}-R^{a1}_{j,a}-E^{a1}_{a,c}$.

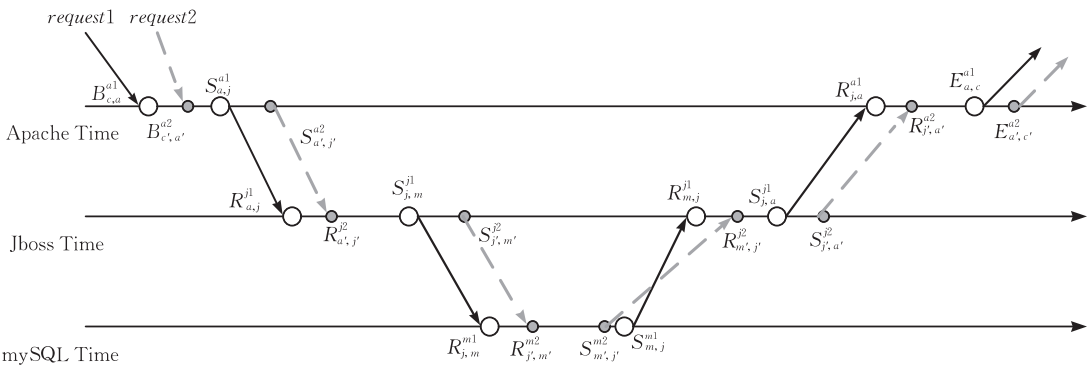

图1 请求路径的关联

对于关联起来的请求路径,我们可以计算请求在路径上各段的处理延迟.这里由于进程上下文关联是对同一进程中的消息作关联,具有一致的时钟,可以得到精确的延迟信息.而消息上下文关联涉及对不同进程,甚至不同物理节点上消息的关联,系统时钟可能存在偏移,所以得到的延迟需要作相应的时间对齐处理.我们的系统采用网络时间协议(network time protocol)实现处理.

2.2 用于再生框架的挑战

利用请求跟踪技术作性能故障诊断有几方面的挑战:(1)服务系统会生成大量的杂乱无章的请求路径,需要找出有代表性的路径,分析系统状态和诊断性能异常;(2)服务系统负载的动态特性,对请求路径分析存在干扰,需要排除奇异点的影响.

对于第1个挑战,我们借用 Aguilera[10] 提出的请求路径模式(request path pattern)的概念,把大量重复出现的具有相同结构和相似时延的路径归为同一路径模式.同时,在多层服务系统里,各类请求的系统特征有较强的差异性,我们对各类请求作独立处理,分析各自的行为特征.因此,我们把请求路径模式作如下定义.

定义1. 请求路径模式.两条请求路径属于同一路径模式,当且仅当满足以下条件:(1)两个请求的类型相同;(2)两条请求路径的消息序列长度相同,且其中对应位置上的消息相似.

定义2. 消息相似.两个消息相似,当且仅当满足以下条件:(1)两个消息的类型相同;(2)两个消息的源地址 IP 和目标地址 IP 对应相同.

基于请求路径模式的概念,我们一方面可以通过对正常的路径模式和检测到的路径模式作比较,找出异常模式,一方面可以通过分析路径模式内部的性能特征是否异常,确定性能故障点.

对于第2个挑战,我们利用动态多门限方法处理.Avritzer[11] 等人利用排队论推导出并在系统级模拟平台上证实:在一个典型的 M/M/c 系统里,有限的样本数量即可以拟合正态分布的均值及方差参数,进而可以用于确定分布的变化.根据这一结论,我们可以利用动态多门限处理方法确定异常变化是否属于统计意义上的变化,消除奇异点的干扰.

3 再生框架体系结构

我们的再生框架(Adaptive Selective Rejuvenation Framework, ASRF)实现对请求 trace 的在线收集、请求路径的精确重构、路径模式的统计分析、请求行为特征在统计意义上是否发生变化的判定、性能故障点检定和选择性再生.再生框架系统架构如图2,包括请求 trace 收集器(tracer)、请求路径构造器(request path constructor)、请求路径分析器(request path analyzer)、请求路径模式库(request path pattern library)和再生管理器(hierchical rejuvenator).请求 trace 收集器布置在多层系统的每个节点上,而其余的模块没有这个要求.出于对 trace 数据量和开销的考虑,trace 搜集器采用变频采样模

式,请求路径生成器使用 trace 作路径重构.路径分析器对每个时间间隔的数据作分析,选择相关的请求路径模式,获得各路径模式中各层模块的依赖关系和行为特征.再生管理器使用各层模块依赖关系和行为特征作选择性再生.

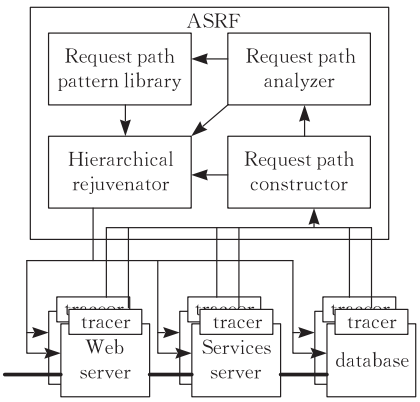

图 2　选择性再生框架系统架构图

4　请求路径构造器

我们在之前的文章中实现了 PreciseTracer[12],一个基于黑盒方式的精确的请求路径构造器的原型系统.我们的路径构造方式,适用于基于进程池(或线程池)处理架构及消息通信方式的多层服务系统.PreciseTracer 通过把 send、receive 等交互消息作关联操作,把每个请求路径构造成一个部件活动图(Component Activity Graph,CAG).PreciseTracer 原型系统,对作关联的请求 trace 有严格的完整性要求,而在实际情况下,请求 trace 搜集很难做到没有丢失的情况.为此,本文对这个原型系统作了改进,主要是加入对请求 trace 丢失情况的处理.我们引入 recycle list 和 garbage list 两个结构.对于一个不完整的请求 trace,大致有以下几种情况:(1)请求路径的头序列丢失;(2)请求路径的尾序列丢失;(3)请求路径的中间序列丢失.对于第 1 种情况,我们使用 recycle list 把 trace 缓存起来,直到时间窗口结束.对于第 2 种情况,我们搜索 recycle list,去寻找匹配的 trace,如果找到就作关联重构,否则,当所有时间窗口中的 trace 都用完,就把 trace 放入 garbage list.对于第 3 种情况,我们把 trace 直接放入 garbage list.我们的改进提高了 PreciseTracer 的适应性,使之可以方便地工作于采样模式.这可以减少 trace 收集和处理的开销,对于在线的检测很重要.

5　请求路径分析方法

路径分析是要分析每个时间窗口中存在的各个路径模式,提取各个路径模式中部件之间的依赖关系和处理延迟的均值及方差等系统行为特征信息.

5.1　请求路径

在 PreciseTracer 中,每个请求路径是一个部件活动图(Component Activity Graph,CAG).我们主要是从系统各层延迟特征的角度分析和确定性能故障点.我们把每一个 CAG 转化为一个含两个元素的向量.这个向量里,第 1 个元素是记录请求路径的顶层信息,包括请求开始的时间戳、端到端响应时间、消息数量、输入数据量、输出数据量等;第 2 个元素是个变长的向量,其中每个元素记录请求路径上的一个消息的信息.

5.2　路径分析

路径分析包括对路径结构和系统处理延迟等行为特征的分析.

路径结构分析:我们针对 CAG 转化后的向量中的第 2 个元素的信息(即路径的底层信息中的消息序列)作遍历,提取依赖关系.

对于行为特征的分析:请求路径上各层功能模块的处理延迟信息可以通过消息之间的时间戳算出.这需要面对双重挑战.一方面,对于多层服务系统,各类请求的处理路径不完全相同,要确定性能异常点,需要对请求作分类处理,在各类请求中分析路径模式.另一方面,虽然各种请求处理延迟有差别,且不符合完全相同的分布,但各层功能模块的延迟分布有很多重合,加之多层系统中缓存等因素的影响,单纯使用处理延迟信息作聚类分析效果并不理想[13].为此,我们使用了服务系统前端的运行日志信息.由于多层服务系统前端普遍采用 log4j 格式①,本文对这种格式日志提取相关外部请求的时间戳、URL、进程号信息,用以匹配请求路径.我们的实验发现,使用前端运行日志信息的方法,比单纯依赖请求路径信息作聚类分析的方法有效.

为分析系统各层功能模块处理延迟变化情况,我们提出两个参量来刻画系统各部件的行为特征:每类请求的端到端的响应时间均值(End-to-End performance mean,E2Epm)和特定类型的请求在特定层的处理延迟均值(Component Performance Mean,CPM).其中,CPM 定义如下:

① Log4j 日志格式参见 http://logging.apache.org/log4j

$$CPM_{j,k}=\sum_i p_{j,k}\times N_{j,k,i}\times latency_{j,k,i}\Big/\sum_i N_{j,k,i},$$

其中 $N_{j,k,i}$，$latency_{j,k,i}$ 和 $P_{j,k}$ 分别代表第 j 类请求流经系统模块 k 的各路径模式的样本数量、延迟和特征参量. 当前的特征参量统一取为 1. 这样 $CPM_{j,k}$ 代表第 j 类请求中各种路径模式的路径样本在模块 k 上处理延迟的均值.

5.3 有效路径模式的选择

上面的方法对于处理给定的请求路径模式中各层功能模块的延迟变化情况是有效的. 但由于多层服务系统的复杂性，实际中会出现大量的请求路径模式，加之负载变化及缓存等因素的影响，这会使路径模式的数量爆炸. 进而影响分析处理效率，同时奇异点也带来准确率下降的问题，影响整体效果. 为此，我们提出一个方法对请求路径模式作有效性选择. 这个与数据挖掘里的有效参数的选择类似. 服务系统通常采用服务水平协议(Service Level Agreement，SLA)规定服务质量①. 具体的，SLA 中规定服务水平目标(Servive Level Objectives，SLOs)，量化度量系统的状态，分为 SLO violation 和 SLO compliance. 为了实现路径模式选择，我们定义一个请求路径有效性的度量标准. 我们把待选择的请求路径模式对系统异常的区分能力作为选择的标准. 对于一个给定的路径模式，如果无法用数据挖掘的方法训练出区分度达到一定标准的分类器，就认为这个路径模式不属于有效模式的集合. 我们构造训练数据，对 SLO violation 和 SLO compliance 两种状态下请求路径数据作标注，训练贝叶斯网络分类器. 贝叶斯网络[14]是一种功能强大且实用的分类工具，可以方便描述多变量间的相互关系. 同时贝叶斯网络对训练的模型具有良好的解释能力，可以使我们更好地分辨路径模式对性能故障的表征能力. 我们使用树状增强型的贝叶斯网络(Tree-Augmented Bayesian Networks，TANs)方法来构建网络结构. TAN[14]要求贝叶斯网络各个节点的入度不超过 2. 这可以避免对数据集的过度拟合，同时也克服朴素贝叶斯网络对各参量间相互完全独立的假设而引入的局限性.

我们把数据集对分类器的训练能力，作为评价给定的路径模式样本所具有的对系统性能故障的表征能力的评价工具. 因此，这里对于训练所得的分类器性能的评价参数的选择是很关键的. 我们选择了 Balanced Accuracy 作为度量参数. Balanced Accuracy 由 Cohen 等人提出[15]，是把样本分类结果的 Ture Positive 和 Ture Negative 取均值得到的. 这样可以同时保证对于 SLO violation 和 SLO compliance 的区分能力，避免偏分现象.

6 再生管理器

再生管理器，采用一个层次化的模式，如图 3. 底层是一个多门限判定模块(identify algorithm)，用以判定功能模块的行为特征是否发生统计意义上的变化. 顶层是一个仲裁模块(arbitrary strategy)，对于多故障的情况，判定最需要再生的模块. 当前，我们的仲裁模块采用倒序优先再生策略，即在底层判定出多个连续性能故障点时，利用处理路径的依赖关系，优先再生距离系统接入点较远的模块. 同时我们对仲裁模块的设计具有良好的扩展性，可以方便地加入各种语义信息和专家知识.

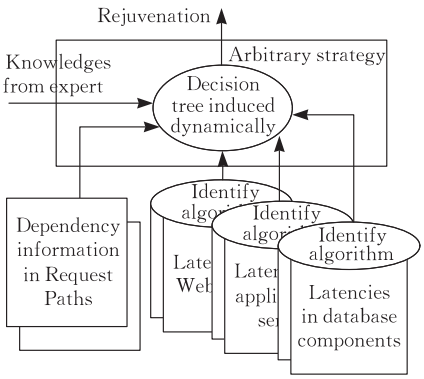

图 3 层次化再生策略框图

底层是对部件级性能特征的分布变化情况的判定模块. 对于部件处理延迟，我们实现了静态均值再生(static rejuvenation algorithm with averaging)算法[11]，来检测给定层的处理延迟的分布情况是否发生变化. 这是一个动态多门限的检测方式，我们设置一个门限值来检测处理延迟是否变化，并使用多个门限值确定这种变化是否表明处理延迟在统计意义上已发生变化，而不是奇异点的影响. 这种方式使用一种装桶的方式处理. 具体来说，设延迟的平均值为 u，其标准差为 d. 初始门限值设为延迟的平均值 u，恒定步长取标准差 d. 设有 m 个桶，每个桶可以装 k 个球. 当前 m 设为 3，k 设为 5. 检测路径模式样本，如果延迟值超过当前门限值，往桶里装入一个球；反之，从桶里拿出一个球. 当前桶满了，就将门限值增加一个步长，并使用一个空的桶作当前桶；反之，当

① Service Level Agreement(SLA)可以参见 http://www.service-level-agreement.net

前桶空了,门限值减一个步长,并把上一个桶作为当前桶.当所有的桶都满了,判定该部件的处理延迟在统计意义上已发生变化.

7　实验评价

7.1　实验负载及平台

我们采用 RUBiS 系统评价本文提出的再生框架.RUBiS① 是由 Rice 大学开发的,用以模拟 eBay.com 的三层服务系统的测试平台.RUBiS 是一个类 TPC-W 的测试平台,共提供了 27 种请求类型,使用马尔卡夫转移矩阵确定负载的动态信息(如下一个请求的类型).RUBiS 提供两类负载模式:默认模式(default)和浏览模式(browse).每种模式有特定的转移矩阵刻画各自的请求类型及转移概率.我们对 RUBiS 的模拟客户端作修改,使负载量可以随时调整.基于此,我们抽取 98 世界杯的负载数据②中有关负载量的波动特征用于实验.

在我们实验里,对系统的服务质量和可用性,使用服务水平协议(Service Level Agreement,SLA)中对服务水平目标(Service Level Objectives,SLOs)的常见的规定方式来量化度量系统当前的状态.其中 SLO 按照如下方式规定:系统处于 SLO Violation 状态,是指系统在一个给定时间段(t)内,请求的响应时间超过某一个门限值(s)的请求个数占该时间段内请求总数的百分比超过某个给定的门限比例(r).反之系统处于 SLO compliance 状态.其中,门限值 t,s,r 的具体取值可以按需要情况设置.

我们的实验平台由 7 个节点的集群组成,用 100Mbps 以太网互联.每个节点上配置两个 PIII 的处理器和 1GB 内存,运行 RedHat 4.1.1-30 Linux 系统,内核版本 2.6.19,使能 kprobe 特性.实验部署如图 4.

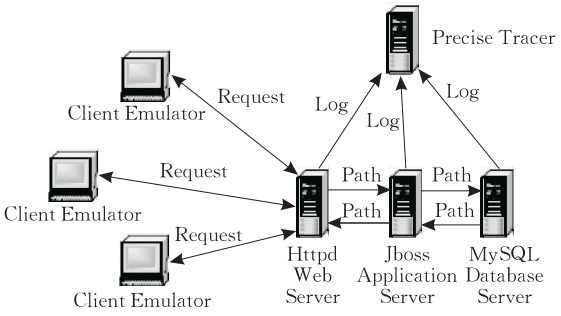

图 4　实验平台部署

7.2　路径分析方法评价

为了评价本文提出的路径分析方法,我们做了两个实验.第 1 个实验对给定路径样本的处理行为特征的分析方法作评价.我们对 RUBiS 提供的默认模式(default)和浏览模式(browse)两种负载类型分别做了实验.我们在 RUBiS 的 BrowseCategories 模块里注入一个 50ms 的延迟.实验结果如图 5.图 5 中各个 Component Performance Mean(CPM)值对应 RUBiS 的各种请求类型的路径样本在 JBoss 层的平均延迟时间.可以看出,在两种负载下,请求类型 BrowseCategories 的 CPM 明显高于其它请求类型路径样本的 CPM.我们的分析方法可以很清晰地分辨出延迟异常的请求路径样本.

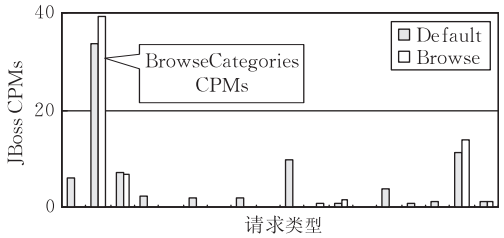

图 5　JBoss 上各请求类型的 CPM 分析结果

第 2 个实验是对有效路径模式的选择方法作评价.我们使用 RUBiS 的默认模式作负载,提取 1998 年世界杯负载中负载量波动特征,生成动态负载.考虑到实验时间和实验平台性能的限制,我们把世界杯决赛日的负载,在时间上按 3min 模拟 1h 作 1∶20 的压缩,在负载量上按 1∶1000 作压缩.本实验里我们使用在 7.1 节中对 SLO 的定义方式,其中门限值 t,s,r 取值为 $t=1$min, $s=300$ms, $r=10\%$,即当在 1min 内请求的响应时间超过 300ms 的次数占全部请求数的比例大于 10%,认定系统处于 SLO violation 状态;反之,系统处于 SLO compliance 状态.我们在 JBoss 节点,动态设置网卡的工作带宽,将带宽减小到 10Mbps,以模拟网络连接的故障,共插入 4 次故障,每次持续 3min,间隔 10min.图 6 中 AveResponseTime 表示平均响应时间(ms);

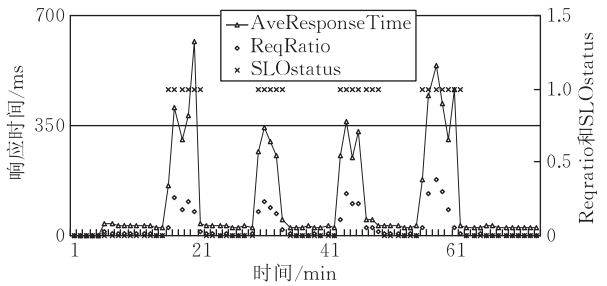

图 6　1998 年世界杯决赛日负载及故障注入实验中的 SLO

① RUBiS 可以参见 http://rubis.ow2.org
② 1998 年世界杯的负载数据可以参见 http://ita.ee.lbl.gov/html/contrib/WorldCup.html

ReqRatio 表示各时间段内请求响应时间大于门限值s的请求数所占的比例，取值范围为 0 到 1；SLOstatus 表示系统当前状态(1 表示 SLO violence，0 表示 SLO compliance).

我们使用请求路径样本来训练贝叶斯网络分类器，使用 10 迭交叉评价方法，用 Balanced Accuracy 度量参数作评价，筛选有效的路径模式. 我们使用 WEKA 数据挖掘工具①提供的 API 实现数据的预处理和贝叶斯网络的训练及测试. 处理结果见表 1. 可以看出被选择的有效路径模式的分辨力明显优于未经筛选的路径数据.

表 1 有效路径模式选择结果

请求类型	路径长度	Blanced Accuracy
withoutSelection		0.56
PutBid	34	0.83
BrowseCategories	48	0.84
BrowseCategories	14	0.85
AboutMe	128	0.86
StoreBid	60	0.89

7.3 再生框架评价

我们使用 RUBiS 的浏览模式作负载. 我们在 JBoss 的 EJBs 里加入随请求处理次数增长的延迟来模拟性能故障. 我们使用在 7.1 节中对 SLO 的定义方式，其中门限值 t,s,r 取值为 $t=5$s，$s=300$ms，$r=20\%$，即，当 5 秒钟内，请求的响应时间超过 300ms 的次数占全部请求数的比例大于 20%，认定系统处于 SLO violation 状态；反之，系统处于 SLO compliance 状态. 实验结果如图 7 所示，其中 AveResponseTime，ReqRatio，SLOstatus 的含义与图 6 中一致. 结果表明我们的再生框架可以确定 JBoss 中的性能异常并作必要再生. 同时再生策略中采用动态多门限处理方法. 该方法使用多次采样方式按照动态调整门限值策略，判定延迟数据是否发生统计分布意义上的变化. 从图 7 中可以看出，该方法对于延迟中瞬态激增的奇异点干扰有很好的屏蔽能力，进而可以避免不必要的再生带来的损失，保证系统的服务质量和可用性.

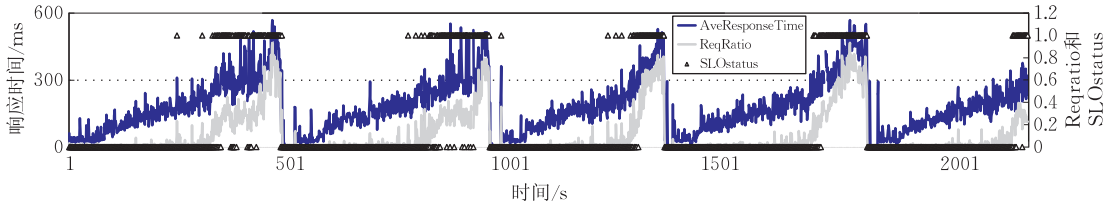

图 7 再生框架下延迟故障注入实验的结果

7.4 框架的性能开销

我们使用 RUBiS 的浏览模式作负载，对使能本文框架(Enable)和不使能本文框架(Disable)作比较，结果如图 8. 我们的方法引入的性能开销非常小. 在负载低于 500 个客户端并发的情况下，开销几乎可以忽略不计. 随着负载加大，开销会有所增加，但不会超过 4%.

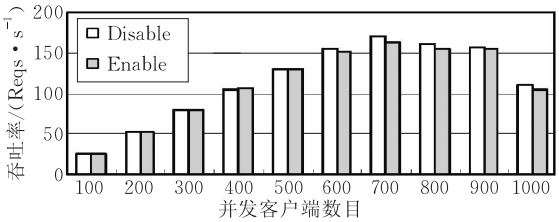

图 8 框架对性能的开销

8 相关工作

软件再生技术引起广泛的研究兴趣. 研究工作主要分两大类：基于分析模型的方法和基于监测的方法. 基于分析模型的方法，通过引入各类分析模型(如 Markov model[2]，Markov regenerative Petri net model[4]，Stochastic Activity Net[3])，来得到系统的最优再生策略. 各类分析模型主要的差别在于对不同的目标系统的状态演进规律的基本假设存在差别，进而引入不同的数学模型加以刻画. 这类方法的优点在于有严格的数学基础，可以实现对系统状态的刻画；不足在于缺乏对系统变化的及时调整和适应能力. 基于检测的方法，通过预测资源的使用情况，实现主动的再生，如文献[6-7]. 这类方法通过对系统参量的在线监测和处理，实现对系统状态的判定和预测，对系统变化有很好的调整和适应能力，但缺少对系统的演进规律的整体刻画. 最近 Vaidyanathan 和 Trivedi[5]提出一种层次化的模型方法，结合两种方法的优点，在底层用监测方法，提取系统状态参量作处理，之后在顶层用分析模型确定再生的时机. Silva 等人[8]通过检测方法，针对 SOA 系统找到一个系统处理能力随系统处理过的请求个数而

① WEKA 网址 http://www.cs.waikato.ac.nz/ml/weka

变化的确定性模型,进而使用主动的再生方式克服系统资源泄漏类的软件故障. Avritzer 等人[11]针对用户关心的性能参数(如响应时间),提出基于检测的再生判定算法,并在系统级模拟平台上作验证. 我们的再生框架基于请求路径的分析技术,实现诊断和定位系统性能故障点,作有选择地再生.

请求跟踪技术是分布式系统性能诊断的有效方法. 研究人员先后提出过多种请求处理路径技术,按请求 trace 的抓取和构建过程中是否对被检测系统进行修改可以分为两类:黑盒方法和白盒方法. 白盒方法(如 Magpie[13]、Pinpoint[9]等)涉及对检测系统的中间件层或应用层作修改,以方便请求路径的构建. 黑盒方法(如 Project5[10]、E2Eprof[1]等)把检测的系统看成黑盒,采用信号处理和统计处理方法实现请求路径的构建. Pinpoint 通过对 j2ee 平台作修改,为每个请求加入逻辑标签,实现请求处理路径的构建. Magpie 在系统的不同点插入探针,并使用应用特定的关联规则构建路径. Project5 使用信号处理方法实现对分布式系统部件间的请求消息的关联. E2eprof 提出类似的基于卷积算法的路径构建算法,并使用了 trace 的压缩表示方法,实现在线的性能诊断. 我们 PreciseTracer[12]的请求路径的构建方法不涉及对被检测系统的修改,属于黑盒方法. 同时我们的路径关联方法通过使用请求处理路径上消息之间的因果关系作精确的关联,避免了传统黑盒方法使用统计方法精度不高的问题.

Candea 等人[16]提出的自动恢复技术把软件再生技术和 pinpoint[9]结合起来,对 j2ee 中间件系统作修改,针对"web 页面访问,返回空白页"一类的应用故障问题作诊断分析和自动恢复. 而我们的工作是通过请求路径构建和分析的手段定位性能故障类问题,作选择性地再生,改善用户关心的响应时间参数,以保证系统的服务质量.

9　结　论

本文针对多层服务系统,提出一个选择性再生框架,把请求路径分析方法与软件再生技术相结合,通过分析请求路径,提取各层部件的行为特征,诊断性能故障,采用选择性再生方式,改善请求响应时间,保证系统的服务质量. 在本文的再生框架里,我们提出一个新的基于请求路径的性能故障分析诊断方法. 实验表明,我们的分析方法可以有效地聚类路径模式,并很好地区分路径模式里的性能异常. 实验也表明,有效请求路径模式的选择方法,可以对生成的大量路径模式作有效地筛选,提高路径分析方法的性能. 同时,我们实现的动态多门限判定方法,作多次采样,按照门限值的动态调整策略,判定系统模块的处理延迟是否出现统计意义上的变化. 这可以屏蔽掉由于瞬态奇异点带来的干扰,进而避免对系统作不必要的再生而引入的损失. 实验结果表明我们的方法可以有效诊断性能故障,及时再生,保证系统服务质量.

参　考　文　献

[1] Agarwala S, Alegre F, Schwan K, Mehalingham J. E2EProf: Automated end-to-end performance management for enterprise systems//Proceedings of the 37th DSN 2007. Edinburgh, UK, 2007: 749-758

[2] Huang Y, Kintala C, Kolettis N, Fulton N D. Software rejuvenation: Analysis, module and applications//Proceedings of the 25th Symposium on Fault Tolerant Computer Systems. Pasadena, California, 1995: 381-390

[3] Tai Ann T, Tso Kam S, Sanders William H, Chau Savio N. A performability-oriented software rejuvenation framework for distributed applications//Proceedings of the 35th DSN 2005. Yokohama, Japan, 2005: 570-579

[4] Garg S, Puliafito A, Telek M, Trivedi K S. Analysis of software rejuvenation using Markov regenerative stochastic Petri net//Proceedings of the 6th ISSRE1995. Location: Toulouse, France, 1995: 180-187

[5] Vaidyanathan K, Trivedi K S. A comprehensive model for software rejuvenation. IEEE Transactions on Dependable and Secure Computing, 2005, 2(2): 124-137

[6] Vaidyanathan K, Trivedi K S. A measurement based model for estimation of resource exhaustion in operational software systems//Proceedings of the 10th ISSRE. Boca Raton, Florida, USA, 1999: 84-93

[7] Shereshevsky M, Crowell J, Cukic B, Gandikota V, Liu Y. Software aging and multi-fractality of memory resources//Proceedings of the 33th DSN 2003. San Francisco, CA, USA, 2003: 721-730

[8] Luis Silva, Henrique Madeira, Gabriel Silva Joao. Software aging and rejuvenation in a SOAP-based server//Proceedings of the 5th IEEE International Symposium on Network Computing and Applications 2006. Cambridge, MA USA, 2006: 56-65

[9] Chen M Y, Kiciman E, Fratkin E, Fox A, Brewer E. Pinpoint: Problem determination in large, dynamic Internet services//Proceedings of the 32th DSN 2002. Bethesda, MD, USA, 2002: 595-604

[10] Aguilera M K, Mogul J C, Wiener J L, Reynolds P, Muthitacharoen A. Performance debugging for distributed systems

of black boxes//Proceedings of the 19th SOSP 2003. Bolton Landing, NY, USA, 2003: 74-89

[11] Avritzer Alberto, Bondi Andre, Grottke Michael, Trivedi Kishor S, Weyuker Elaine J. Performance assurance via software rejuvenation: Monitoring, statistics and algorithms// Proceedings of the 36th DSN 2006. Philadelphia, Pennsylvania, USA, 2006: 435-444

[12] Zhang Z H, Zhan J F, Li Yong, Wang Lei, Meng Dan, Sang Bo. Precise request tracing and performance debugging for multi-tier service of black boxes//Proceedings of the 39th DSN. 2009: 337-346

[13] Barham P, Donnelly A, Isaacs R, Mortier R. Using magpie for request extraction and workload modeling//Proceedings of the 6th OSDI 2004. San Francisco, California, USA, 2004: 259-272

[14] Han J, Kamber Micheline. Data Mining: Concepts and Techniques. 2nd Edition. San Francisco, CA, USA: Morgan Kaufmann, 2006

[15] Cohen I, Goldszmidt M, Kelly T, Symons J, Chase J S. Correlating instrumentation data to system states: A building block for automated diagnosis and control//Proceedings of the 6th USENIX OSDI. San Francisco, CA, 2004: 231-244

[16] Candea George, Kiciman Emre, Kawamoto Shinichi, Fox Armando. Autonomous recovery in componentized Internet applications//Proceedings of the Cluster Computing. Barcelona, Spain, 2006: 175-190

TIAN Guan-Hua, born in 1980, Ph. D. candidate. His research interests focus on quality of service in cluster system.

ZHAN Jian-Feng, born in 1976, Ph. D., associate professor, M. S. supervisor. His research interests include cluster operating system, distributed system, high availability software.

MENG Dan, born in 1965, Ph. D., professor, Ph. D. supervisor. His research interests include architecture of high performance computing system, system software, high performance communication protocol, distributed file system, operating system for cluster system.

Background

Quality of service and availability are crucial for Internet service system. Developed from and deployed with various heterogeneous middleware and commercial off-the-self software without source code, service systems become increasingly complex, and introduce more hidden space for software defects. It is challenging to guarantee QoS and availability of service system.

Software rejuvenation technique, as an effective proactive measure against the progressive performance faults or soft faults, has received wide attention in research fields. Related research works fall into two broader categories: model based approaches and measurement based approaches. Model based approaches introduce various models to depict system state and evolvement, in order to get the optimal rejuvenation schedule. Measurement based approaches do necessary proactive rejuvenation, based on detection and prediction of system resource usage metrics. The former category has sound mathematics foundation, but lack of adaptability for system changes; the letter one has the adaptability and pays more attention on proactive recovery.

This work, selective rejuvenation framework, belongs to the measurement based approaches. The authors take request causal paths into consideration, instead of resources usage metrics which is used in traditional approaches. They combine request tracing technique with software rejuvenation technique to identify and selectively rejuvenate the most potential faulty component in multi-tier service systems, and introduce a novel request causal path analysis approach and implement a hierarchical rejuvenation scheme.

This work is supported by the National Science Foundation for Young Scientists of China (Grant No. 60703020), the National High Technology Research and Development Program (863 Program) of China (No. 2009AA01Z139), and National Science Foundation of China (No. 60933003). The projects focus on improving the availability of large scale network systems. This work improves QoS and availability of multi-tier services and the selective rejuvenation approach is applicable to large scale systems.

The group is always doing researches in the field of dependability and availability for large scale systems, and has developed Fire Phoenix system, a scalable and fault-tolerant cluster management system for Dawning4000 series, and Dawning5000 series. The work is published in Proceedings of the 7th IEEE Cluster Computing 2005. Meanwhile, the group also got some interesting achievements in the relative fields, e. g., quality of service guarantee, fault diagnosis, performance debugging. Jiang Y. et al. introduced a promising admission control policy to guarantee QoS of cluster systems, and her work is published in Proceedings of the 8th IEEE Cluster Computing 2006. Wu L. et al. proposed a fault diagnose algorithm for cluster systems, and his work is published in Proceedings of the 20th IEEE IPDPS 2006. Zhang Z. H. et al. proposed a precise request causal path reconstruction algorithm, and his work is published in Proceedings of the 39th IEEE DSN 2009.