

文章编号: 1005-8451 (2010) 07-0014-03

## 基于 JMX 与 J-SIM 的业务仿真平台研究

林碧霞, 尹治本

(西南交通大学 信息科学与技术学院, 成都 610031)

**摘要:** 存储行业的飞速发展, 存储阵列是否能够承受得了复杂业务场景下发的输入/输出 (I/O), 成为各大厂商与客户关心的问题。为了避免复杂业务场景的搭建, 本文通过模拟业务场景对存储阵列进行性能测试。利用 JMX 与 J-Sim 技术进行存储阵列的业务平台仿真系统研究。

**关键词:** 存储阵列; 虚拟用户组; 虚拟资源库; 虚拟机

**中图分类号:** TP39

**文献标识码:** A

### Research on performance simulation platform based on JMX and J-SIM

LIN Bi-xia, YIN Zhi-ben

(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** The storage trades were rapid developed. As vendors and customers concern, whether storage arrays were able to be issued I/O in complicated business scenes. In order to avoid the building of complex business structures, the paper simulated the scenes to test the performances for storage arrays. This article used JMX with the J-Sim technology for Storage Array System simulation platform for research.

**Key words:** storage arrays; virtual user group; virtual resource; virtual machine

随着 Internet 的高速发展, 网络规模不断扩大, 存储系统成为 Internet 背后的功臣, 也促成了存储行业的高速发展。存储业的发展, 面临着如何有效地对存储阵列的性能情况进行评价与分析的问题。因为存储阵列在不同的业务场景下, 可能体现出不同的性能, 所以如何在不同的业务场景下对存储阵列进行性能测试, 从而分析出存储阵列的性能, 成为各厂家在测试过程中的重点与难点。

本文基于某公司的需求构建了一个新的网络仿真平台。通过 JMX 结合 J-Sim, 利用模拟各种业务场景, 对存储阵列进行性能测试。平台开发目的是创建一个可能快速实现上层应用模拟的环境。

### 1 JMX 技术概述

JMX<sup>[1]</sup> (Java Management Extension) 是 J2EE 中的通用管理规范, 它定义了一套标准的代理和服务, 可为应用程序植入管理功能的轻型框架。JMX 可以跨越异构操作系统平台、系统体系结构和网络传输协议, 从而对网络上多种资源 (包括应用程序、设备、服务等) 统一进行管理, 提供

一致的管理视图。因此逐渐成为管理复杂软件系统的优秀解决方案<sup>[2]</sup>。JBoss、IBM 的 TivoliJMX 和 BEA 的 Weblogic 等商业软件已成功的应用了 JMX, 有效地提高了系统的动态配置和管理能力。

JMX 是一个用来管理网络、设备、应用程序等资源的管理体系结构。JMX 采用三级分而治之的体系和结构化方法来降低网络管理的复杂性, 每层都是高度组件化的并由良好定义的接口进行划分。

(1) 工具层: 可管理端点 (设备、软件服务等), 可通过 JMX 指定的接口被访问。这是通过创建公开可配置属性、可访问操作和事件的 Java 对象实现的。这些对象称为 Managed Bean (简称 MBean)。在规范中将可通过这些对象管理的端点称为 JMX 可管理的资源。

(2) 代理层: JMX 代理是软件组件, 它向远程管理组件公开一组标准化代理服务, 并通过 JMX 可管理资源的 MBean 接口直接控制这些资源。实际上, 在 JMX 代理内可通过能够动态地装入和卸装 MBean 的 MBean 服务器来管理 MBean。

(3) 分布式服务层: 本层定义 JMX 管理平台对代理层进行操作的接口和组件。应用程序可以访问代理或代理组来管理由代理公开的 JMX 可管

收稿日期: 2009-11-11

作者简介: 林碧霞, 在读硕士研究生; 尹治本, 教授。

理资源。

为了让业务仿真平台可以动态管理, 本文提出一种基于 JMX 的性能业务仿真平台设计方案, 具有动态的配置管理能力、动态的调度策略选择机制和可动态配置的可视化管理视图。

## 2 业务仿真平台的设计

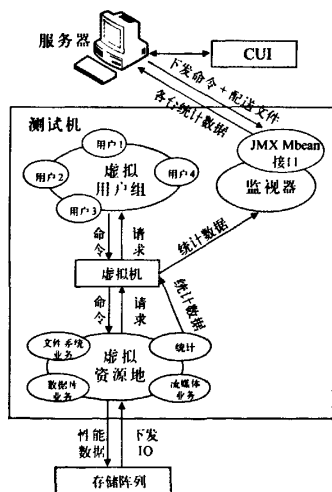
## 2.1 功能模块

性能仿真平台主要用于对存储设备进行性能的测试, 这些网络设备分布在 Internet 和 Intranet 网络的环境中, 一般通过 HTTP、TCP/IP 协议通信, 所以将仿真平台分为服务端与测试机端 (1 对多的关系)。

## 2.2 系统结构

系统分为服务端和测试机端。图1为仿真图。

### 2.2.1 服务端



**图 1 仿真平台总体框架**

在功能上负责用户与测试机的图形界面接口，包括每台测试机的用例配置、待测试存储的指定、性能测试的项目、性能分析数据、报表的查询与显示，并对各测试机的相关系统故障和告警的查询与显示。与一般JMX的应用不同，将JMX的MBean Server与HTTP浏览服务器放在这一层，通过多台测试机同时对存储进行IO的下发，增加对存储的压力。在服务端采用被动接收来自各测试机的数据信息，对每台测试机发来的信息进行存储，显示瞬时的各台测试机的统计性能数据，并

打印出相关性能数据曲线,同时对应 CPU 占用过大、系统崩溃等进行告警。

### 2.2.2 测试机端

仿真平台的测试机端是核心系统。每台测试机需要在 Server 中注册，通过适配器 invoke 方法实现对 MBean 进行调用，将性能数据根据服务端设置的时间段发给服务端。测试机端注册了信息发送模块、告警管理、日志管理。根据需要定义了 2 种适配器：(1) 本地调用的适配器；(2) 基于 HTTP 的远程调用的适配器。每台测试机根据分配的用例运行一个核心系统，对存储阵列进行 IO 的下发操作，并在纳秒级别对信息进行统计汇总，发送给服务端。

### 3 测试机端的核心系统

每个测试机端都运行着一个核心子系统，这个子系统使用 J-SIM 设计开发。

### 3.1 J-Sim 平台概述

J-Sim<sup>[31]</sup>(以前也被称作 JavaSim)是用纯 Java 语言开发的一种开源的、基于组件合成的网络模拟环境,该环境建立在组件自治体系结构(Autonomous Component Architecture, ACA)和可扩展 Internet 网络框架(Extensible Internetworking Framework, INET)之上。在不同的网络场景之下采用“即插即用”的组件叠加方式就可以构建出所需要的仿真环境和网络协议栈,如无线传感器网络。

J-Sim 是一个层次化、可扩展的网络仿真平台。在 J-Sim 平台下, 组件是一个基本的单位。从网络环境的角度看, 主机、路由器、TCP / IP 协议等网络实体都是组件。而整个平台是以自治组件框架 (ACA) 为基础的, 所谓的自治组件架构是指它的架构思想仿效数字电路中 IC 芯片的设计和制造, 一个组件与另一个组件相互独立。软件系统由组件有机地组合起来类似于一块印刷电路板上焊接芯片, 甚至在系统运行时还可以添加组件。J-Sim 对网络环境的模拟是通过在 ACA 的基础上建立可扩展网络模拟 (INET) 来实现的。这个模型采用了基于包的通讯机制。此外, 它定义了节点 (主机或路由器) 的结构和基本网络组件。根据需要, 新定义的网络组件都是它们的子类, 这些网络

组件以TCP/IP模型为依据,进行层次化地编写。对于编写之后的组件代码,需要时可以直接调用,也可以它的基础上扩展新的网络组件。

J-Sim仿真通过执行线程表来实现。仿真运行开始,先构建各节点,随后目标节点开始生成监测数据,目标节点将监测数据选择发送给最佳的传感器节点,传感器节点获得数据后在汇聚节点分配的时限内将数据上传。

### 3.2 核心子系统设计思想及其实现

本仿真平台的核心子系统包括业务场景的抽象,子系统模拟业务开发。目前存储所涉及的业务场景有文件系统业务、流媒体业务(包括VOD、视频监控等)、数据库业务(支持Oracle、SQL、MySQL等)等业务场景。比如移动公司所涉及到的业务场景。

本平台业务模拟是在对实际测试的具体业务进行抽象后,在本仿真平台上通过在服务端更改业务配置文件(对应于测试用例)来实现。

子系统模拟业务开发模块包括虚拟用户组(Virtual User Group)、虚拟机(Virtual Machine)、虚拟资源池(Virtual Resource Pool)、监视器(Monitor)等。

(1) 虚拟用户组:用户组是用户群体行为的发起源,负责调试用户群体行为、收集用户的反馈信息,并传递给监控器。用户组是由多个用户组成,并根据具体用户是否在进行业务操作分为Active user和Idle user。通过对存储阵列所涉及较多的文件系统业务、流媒体业务、数据库业务进行模拟。根据实际对存储进行的IO操作,将用户抽象出来,并将用户分为用户群体与单个用户。通过用户数目与用户行为来模拟真实业务。用户组有一个端口,用于把用户反馈传递给监视器。

(2) 虚拟机:虚拟机是用于模拟用户请求的紧急度进行排序和发送相关的请求。它负责任务的调试和用户与资源之间的通信,VM(虚拟机)包括线程池及任务调度。用户与资源之间的任何通信都是以任务的方式提交给VM调试。VM主要用于模拟存储系统的上层业务的离散事件模型。

(3) 虚拟资源池:对用户请求的各种资源进行存储阵列的访问,并根据对存储阵列的资源请求进行相关性能数据的统计与分析。根据实际业务分为:文件系统资源池、多媒体资源池、数据库

资源池。资源池端有一个资源代理(Resource Agent),用于接受用户端(User)发来的创建的资源类型和数量,并转发用户的具体请求。

(4) 监视器:根据多线程调度,对资源池进行的纳秒级别的数据进行汇总,并对数据进行分类汇总、显示。

(5) 资源代理(Resource Agent):主要功能是根据用户请求生成不同的资源池,分发user请求到不同的资源池里,并控制user的并发数。

(6) 资源池(Resource pool):由一系列同类的资源组成,资源池是一个Parent Component,它有两个控制端口(a.把用户请求转发给资源;b.控制用户对某个资源的最大连接数)。当用户要操作资源时,必须先与Pool Component取得联系,然后Pool Component在资源池里随机选取一个资源给用户,增加一个端口,端口数至少为1,这个默认端口是接受pool Component发来的有用户要进行操作的信息,向资源池专门收集统计信息的端口(statistic port)发送信息。

Resource Port是在用户和资源池通信的过程中动态生成的一组端口,用来与资源进行通信。

Fork Port用来接收Link发送给自己的fork事件,从而产生不同的用户。

## 4 结束语

本文利用JMX的动态配置管理能力、动态调度策略选择机制开发了服务端与测试机端,测试机端利用J-Sim模拟业务框架进行开发,从而让本业务仿真系统具有动态配置、更高的可维护性和扩展性。本业务仿真平台已实现,在某公司的性能测试工作中投入使用,节约了人力、物力和财力,并且达到了较高的置信度。

### 参考文献:

- [1] 廖 渊,王保进,李明树,等.智能应用服务协议iASP的设计和实现[J].计算机工程与设计,2003,24(12):13-17.
- [2] 廖 渊,李明树,王 青.基于Linux的网络计算机服务器的设计与实现[J].计算机工程与应用,2003,39(14):12-16.
- [3] ACA:The autonomous component architecture[EB/OL].<http://www.j-sim.org/whitepapers/aca.html>, 2003.