第41卷 第8期 西 安 交 通 大 学 学 报 Vol.41 №8
2007年8月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Aug. 2007

网格测试引擎:一种构建网格测试环境的基础架构

王寅峰，董小社，何秀强，朱正东
（西安交通大学电子与信息工程学院，710049，西安）

摘要：针对动态、开放、分布式网格环境动态组织资源的需要，提出了一种构建网格测试环境的基础架构——网格测试引擎.该引擎支持测试项目动态的部署、更新和执行，支持基于故障紧急等级的预警和自动处理.采用令牌方式来控制目标服务的测试压力，通过令牌迁移实现分布式测试的负载平衡.在ChinaGrid环境下的实验结果表明，基于测试引擎架构所构建的监测环境，对主机系统影响不会超过1%，并可生成符合要求的测试压力，能够相对准确地测试网格资源与服务，反映目标服务性能的变化趋势.

关键词：网格测试引擎；分布式测试；网格资源
中图分类号：TP016.1 **文献标识码**：A **文章编号**：0253-987X(2007)08-0884-05

Grid Testing Engine: a Infrastructure for Constructing Grid Testing Environment

Wang Yinfeng, Dong Xiaoshe, He Xiuqiang, Zhu Zhengdong
(School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: Focusing on the requirements of dynamically organizing resources in dynamic, open, and distributed grid environment, an infrastructure for constructing grid test environment—grid testing engine is proposed, which supports dynamic deployment, update and execution of testing items as well as alarming and auto-handling based on the emergence category of fault. The test pressure of target service is controlled by using tokens and the load balance of distributed testing is realized by token migration. Experiments in ChinaGrid environment show that the impact of the test environment constructed by the grid testing engine on the host system performance is less than 1%, the environment can create test pressure in accord with demands, test resources and service exactly, and reflect the performance trend change.

Keywords: grid testing engine; distributed testing; grid resource

网格监测的目的是对网格平台及应用的功能和性能等进行监测，是为了提高网格系统的可用性和可维护性服务.

网格监测环境基础架构的研究主要有Inca[1]和DiPerF[2]. Inca是一种对网格虚拟组织服务约定进行自动验证的客户端-服务器架构，而DiPerF是一种分布式性能测试框架，可实现对服务性能的自动测试. DiPerF可对一组地理位置分布的测试机器进行协调，对目标服务进行并发测试，启动、协调测试机器.这2种架构的测试脚本指令（或客户端代码）类型和执行信息，需要在构建网格系统监测环境时预先定义，因此缺乏灵活性，也不支持测试脚本指令的动态更新，限制了网格监测环境的动态性.文献[3]对基于面向架构服务（SOA）的分布式测试流程进行了分析，文献[4]对网格服务测试、网格性能测试和网格软件测试的管理进行了分类，并开发出一种模拟测试工具GridTestMaker.

本文提出了一种基于网格测试引擎的分布式网

收稿日期：2006-12-06. 作者简介：王寅峰(1974～)，男，博士生；董小社(联系人)，男，教授，博士生导师. 基金项目：国家高技术研究发展计划资助项目(2006AA01A109)；教育部教育科研网格计划资助项目(CG2003-CG008).

格监测环境基础架构，它无需在网格测试时预先构建，避免了更新、增加测试内容的困难，解决了故障预警以及自动处理等问题.

1　网格测试引擎

定义 1　测试项目是测试引擎的基本执行单位，用于监测网格系统的某个功能或性能指标，它由测试描述文件和测试代码 2 部分组成.

定义 2　测试描述文件包含了测试引擎执行测试项目的所有信息，包括测试代码名称，测试执行的启动时间、持续时间、执行频率，故障预警和自动处理的触发条件以及执行时的操作信息.

定义 3　测试代码分为测试执行代码和故障处理代码. 测试引擎通过运行测试执行代码可获得目标网格资源/服务的功能、性能监测结果. 当网格出现故障或者有故障发生趋势时，测试引擎将执行故障处理代码来完成故障的预警和自动处理，而故障处理代码可指示故障处理的具体执行动作.

1.1　测试引擎架构

测试引擎由 2 部分组成：集中式控制器(CC)和多个分布式控制器(DC)，如图 1 所示. CC 是测试引擎的管理模块，负责分发测试项目，组织和同步各个 DC 执行测试项目、处理监测结果等. DC 可部署在各个网格资源所在的结点或测试机器之上，负责 CC 分发测试项目，收集并返回监测结果.

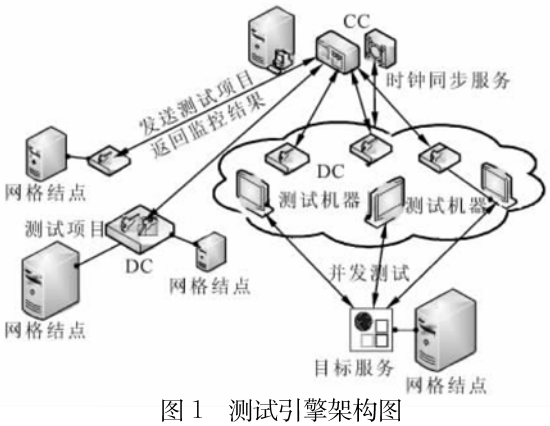


图 1　测试引擎架构图

测试引擎支持测试项目的部署、执行类型有以下 2 种.

(1) 测试项目直接部署在资源所在网格结点的 DC 之上，这样可以监控、测试网格资源内部的运行状态和机器的性能.

(2) 测试项目部署在专门的测试机器之上，通过时钟同步服务器完成测试机器之间的同步运行，并发执行测试项目完成目标服务的功能、性能测试.

1.2　测试引擎执行流程

测试引擎的执行流程如图 2 所示.

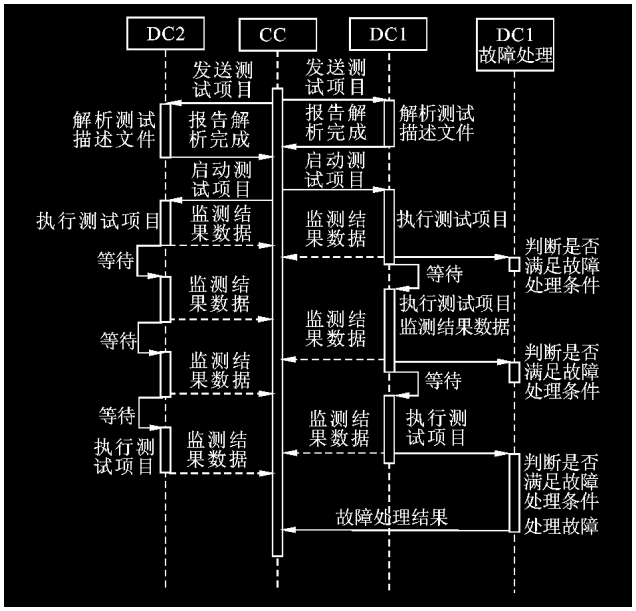


图 2　测试引擎执行流程

2　测试引擎关键技术

2.1　网格故障的预警、处理和调度

测试引擎将网格故障处理作为测试项目定义的基本组成部分，测试描述文件可以定义故障处理的触发条件、紧急等级和故障处理代码，而触发条件取决于测试项目的监测结果.

在测试引擎中定义了 3 种类型的故障处理请求——Info(信息提示)、Warn(预警)和 Error(错误处理)，并为每种类型的故障请求定义了 3 种等级——Normal(普通)、Emergent(紧急)和 Disastrous(最高紧急)，目前测试引擎使用了其中的 5 种，其他紧急等级留作扩展时使用，故障处理请求、紧急等级定义策略如表 1 所示.

测试引擎的调度功能由 DC 的故障处理执行模块负责完成. 处理时应优先考虑高紧急等级的故障处理请求，同一等级的事件按照请求提交的先后顺序来处理.

采用上述故障处理调度策略的优点如下.

(1)由事件处理模块触发事件执行模块，不需要持续或周期性地搜索请求队列，这样可以减小系统消耗，提高事件处理的整体效率.

(2)当故障处理请求过期时，将请求转移到存档文件之中，这样可以减少故障处理等待队列中的请求数，满足故障处理的实时性要求.

表1 故障处理请求、紧急等级定义策略

	Info	Warn	Error
Normal	I-N	W-N√	E-N√
Emergent	I-E	W-E√	E-E√
Disastrous	I-D√	W-D	E-D

注:√为本文架构所使用.

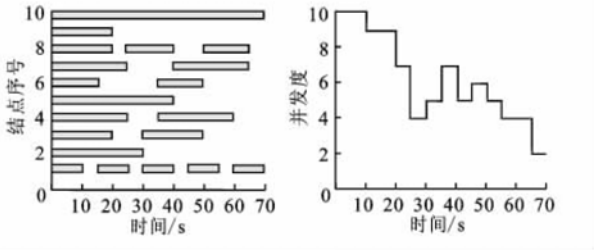
2.2 测试压力控制和动态分配

测试引擎采用令牌机制来解决压力控制与任务动态分配问题.由CC按照测试描述文件的要求产生令牌并分配给DC,令牌是DC能够执行测试项目的标志.每个令牌都由独立的开始、结束时间来指定其有效周期,在令牌的有效周期之内,持有令牌的测试机器才能执行测试项目.

2.2.1 测试压力控制

定义4 在测试项目的运行过程中,某时刻的有效令牌总数代表了允许DC并发执行测试任务的数量,称之为测试项目并发度.

利用令牌的启动和失效描述有效令牌总数的动态变化,可以动态控制测试项目的并发度,进而实现测试压力的控制.牌是一个XML文件,令牌的离散有效周期是可以指定的.令牌机制的工作原理如图3所示.



矩形长条的起点和终点分别表示令牌有效时间的开始和结束

(a)结点的令牌离散有效时间 (b)测试项目并发度

图3 令牌机制工作原理

图3显示CC共颁发了10个令牌,每个测试结点上有1个DC.测试项目的并发访问将对目标资源/服务产生压力,并发度的大小等同于压力的大小.

符合一定规则的令牌文件集合最终将映射为针对目标资源/服务的压力集合,合理地分配测试机器上的令牌文件就可以有效控制目标资源/服务的测试压力.

2.2.2 测试任务动态分配 分布式测试的一个重要特点是测试项目中的所有测试机器都周期性地执行相同的测试任务,而且在每个测试机器上都保存执行测试项目所需要的信息,如执行频率、测试代码等.基于分布式测试的这个特点,依据令牌在测试机器间的动态迁移,可以在测试机器上实现测试任务负载的合理分配.令牌动态迁移如图4所示.

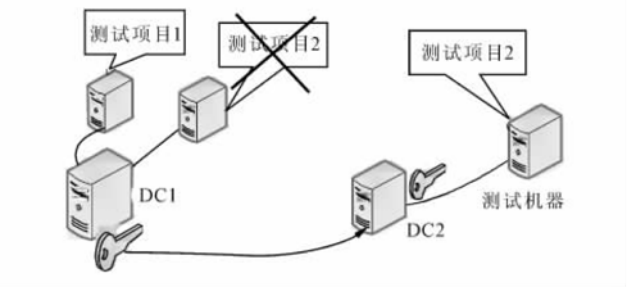


图4 令牌的动态迁移示意图

在图4中,如果DC1持有的令牌在有效时间结束之前转发到DC2,DC2就可以在令牌的有效时间内执行测试任务,相应地,在DC1上停止执行测试任务,测试任务将转移到DC2.令牌的迁移机制为测试任务在DC间的动态分配提供了实现条件.

3 实验及性能分析

本文定义了噪音指数,用于描述和定量分析测试引擎对网格系统的影响.

定义5 测试资源/服务所在结点的系统负载为L,标准服务在测试项目执行前后的响应时间分别为T_B和T_A,则系统负载为L时的噪音指数

$$N_L=\frac{(T_A-T_B)}{T_B}$$

由于测试引擎原型系统是基于Globus Toolkit (GT)4.0容器实现的,所以在计算噪音指数时可选用GT4.0容器系统的标准服务.

3.1 效率测试分析

实验环境1 使用11台测试机器,硬件配置为CPU Pentium(R)4, 主频2.4 GHz,内存1.00 GB.软件配置为Redhat9.0,ChinaGrid网格中间件CG-SP[5],测试引擎TE.网络配置为1 000 Mb/s局域网互连.选取1台机器安装测试引擎的CC功能模块,其余10台安装DC模块.

3.1.1 测试项目部署效率的实验 从向CC提交测试项目到所有测试项目在DC部署、解析完毕,称为测试项目的部署时间.实验中使用了不同的测试项目,其文件大小分别为5 kB、10 kB、20 kB,它们分别同时部署在10台DC上,部署时间如图5所示.

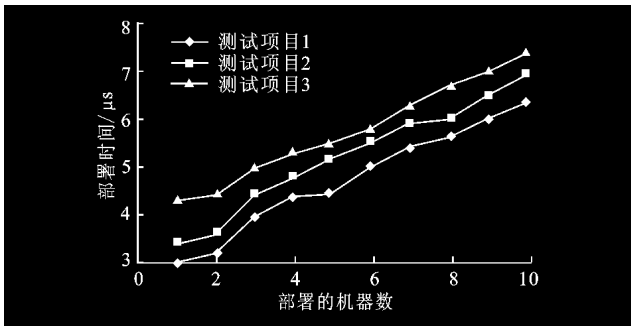


图 5　测试项目的部署效率

从图 5 可以看出,部署测试项目所需的时间与测试项目的大小有关,这主要是由数据传输量以及不同的解析时间引发的,其中测试项目 3 的文件最大,部署时间也最长,但部署 10 台测试机器所需要的时间不会超过 10 s.

3.1.2　*噪音指数实验*　通过在标准服务上驻留的实验结点循环运行任务,可形成不同的负载等级,结点负载分别为 Lev1(11%～20%),Lev2(21%～30%),…, Lev8(81%～90%),Lev9(91%～100%),测试项目执行频率分别为 1/10 s,1/5 s,1/1 s和1/0.5 s,测试引擎噪音指数实验结果如图 6 所示.

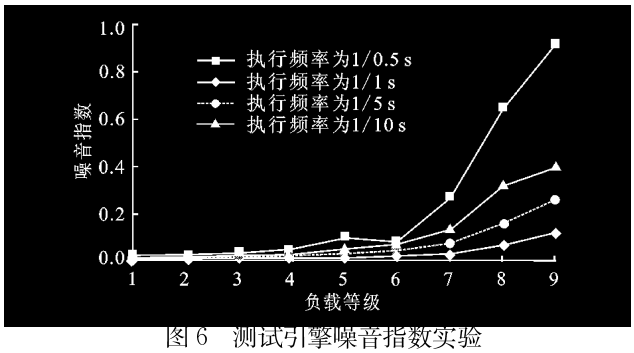


图 6　测试引擎噪音指数实验

实验结果表明,噪音指数与系统负载有关,并随系统负载的增加而增大,而当测试频率较低时,测试引擎对测试结点的影响比较小. 所以,在结点负载较重时(大于 70%),应避免执行高频率测试,DC 触发测试项目的令牌应向负载较轻的机器上迁移(见图 4),以免结点性能急剧恶化.

在迁移过程中,为了防止令牌丢失,测试引擎采用了类似 TCP 协议的握手机制,它要求 DC 在接收到令牌之后,必须向发送令牌者返回确认消息,而发送令牌者接收到确认消息后才能删除令牌文件,否则在接收超时之后,将向接收令牌 DC 重新发送确认信息. 在结点负载不超过 60%的情况下,基于测试引擎的测试项目对原网格系统的影响不会超过 1%,测试结果也证明了基于测试引擎的网格监测环境是有效的.

3.2　分布式并发测试

实验环境 2　在中国教育科研网格(ChinaGrid)环境中进行了实验,其中有 4 个高校网格结点,每个结点部署了 10 台测试机器,每台机器上均部署测试引擎 DC 模块. CC 部署在清华大学结点上,软件配置和分布式测试环境均如同 3.1 节所述. 目标服务部署在清华大学结点的 CGSP2.0 JobManager 作业执行管理服务器上,以测试服务响应时间.

3.2.1　*有效性和准确性实验*　首先定义了 2 组不同的压力条件,CC 生成相应的令牌文件,令牌文件测试项目以分布式测试环境的形式部署在 40 台机器上. 测试项目执行频率为 1 次/s,在所维护的监测结果中,测试引擎 CC 记录了每个测试项目执行周期的开始和持续时间. 通过调用 CC 提供的监测结果文件标准访问接口,可在 2 组压力条件下每间隔 100 ms 获得一个测试项目的并发度. 压力准确性实验结果如图 7 所示.

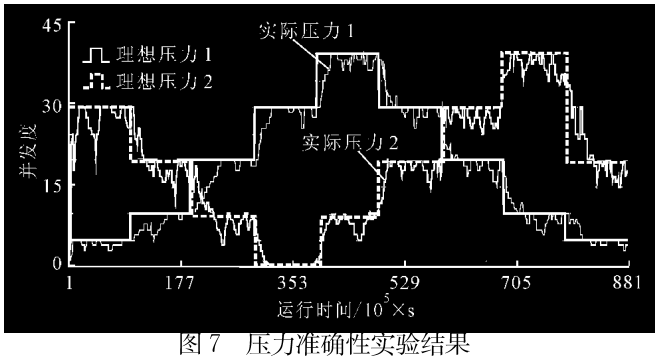


图 7　压力准确性实验结果

图 7 中的理想压力 1、压力 2 是实验中预先定义的 2 组压力条件,CC 根据该条件产生令牌文件,并分配给各个测试机器,以控制测试机器对 JobManager 的并发测试. 实际压力 1、压力 2 分别对应于理想压力 1、压力 2. 尽管实际压力与理想压力之间存在着一定的误差,但其变化规律符合理想压力的条件.

3.2.2　*目标资源/服务性能测试实验*　该实验是测试 JobManager 的服务响应时间,实验条件与 3.2.1 节所述相同. 在测试结果中记录了每个 DC 测得的服务响应时间和每次执行测试项目的开始、结束时间等,从中可以获得目标服务的吞吐率、最大并发请求数以及响应时间等多种性能,实验结果如图 8 所示.

从图 8 可以看出,目标服务的吞吐率和并发服务数基本符合理想压力的要求,其响应时间随着压力的增大而增加,若并发度小于30,响应时间应在

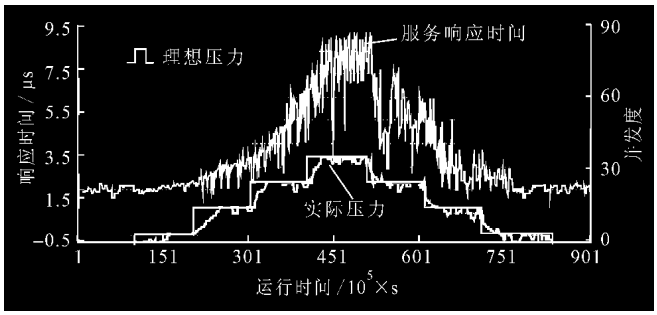


图8 JobManger服务响应时间

可接受的范围之内,这说明目标服务在此范围内可以保证QoS服务.另外,测试引擎不仅能反映目标服务的性能变化,还可以获得某个压力条件下的大致响应时间等.

4 结 论

本文提出了一种称之为网格测试引擎的分布式网格监测环境基础架构,其利用令牌机制来平衡负载.实验结果表明,该引擎具有高效、低噪等特点,可以有效地组织分布式测试环境,并可针对目标服务产生相对准确的压力,测试目标服务的性能,发现系统瓶颈资源,为优化系统性能提供服务.

下一步研究将考虑构建多个对等管理域间的测试引擎机制,完善构建网格监测环境基础架构的测试引擎,并将其扩展到其他网格系统之中.

参考文献:

[1] Smallen S, Ericson K, Hayes J, et al. User-level grid monitoring with Inca 2, SDSC TR-2007-1 [R]. San Diego: Supercomputer Center Technical, 2007.

[2] Raicu I, Dumitrescu C, Ripeanu M, et al. The design, performance, and use of DiPerF: an automated distributed performance testing framework [J]. Journal of Grid Computing, 2006, 4(3): 287-309.

[3] Gerardo C, Massimiliano D. Testing services and service-centric systems: challenges and opportunities [J]. IT Professional, 2006, 8 (2): 10-17.

[4] 郭勇,邓波,衣双辉. 面向服务的网格软件测试环境[J]. 软件学报,2006, 17(11): 2335-2340.
Guo Yong, Deng Bo, Yi Shuanghui. Service oriented grid software testing environment [J]. Journal of Software, 2006, 17(11): 2335-2340.

[5] Wu Yongwei, Wu Song. CGSP: an extensible and reconfigurable grid framework [C] // Proceedings of International Conference on Advanced Parallel Processing Technologies. Berlin: Springer-Verlag, 2005: 292-300.

(编辑 苗凌)