基于开放式云平台的开源在线评测系统设计与实现

张浩斌
（浙江传媒学院新媒体学院 杭州 310018）

摘 要 从应用角度出发，设计并开发了基于开放云平台的开源在线评测系统。该系统从系统实现、系统搭建、平台运维及题目的获得等诸多方面解决了大学和科研机构搭建在线评测系统的困难。从理论上论证并用实际代码和实际系统检验了系统的可行性，开放了全部源代码，为在线评测系统的改进找到了新途径。

关键词 在线评测系统，分布式系统，开放式云平台，开源软件，XML
中图法分类号 TP391.76 **文献标识码** A

Design and Implementation of the Open Cloud Platform Based Open Source Online Judge System

ZHANG Hao-bin
(Department of New Media, Zhejiang University of Media and Communications, Hangzhou 310018, China)

Abstract Designed and Implemented the open cloud platform based open source online judge system from the applicable view. The system resolved a set of problems in deploying online judge system by universities or research institutions, which included implementing, deploying, maintenance and questions acquiring. Discussed the Feasibility theoretically and proved it by running codes and applicable system. By releasing all the source code, provided a new way to improvement online judge system.

Keywords Online judge, Distribution system, Open cloud platform, Open source software, XML

1 在线评测系统的现状与应用难点

在线评测系统(Online Judge,OJ)起源于国际大学生程序设计竞赛(ACM/ICPC)，是一种采用黑盒测试原理进行计算机程序正确性判断的自动化程序[1]。该系统具有比赛公平性好、评测自动化程度高、节约人力成本等优点，广泛应用于各类计算机程序设计比赛和教学[2]。

在线评测系统既可以指该系统的源码，也可以指该系统运行的一个实例。从信息系统的角度看，由于在线评测系统安装、维护上的难度，应当将系统软件整个生命周期中的所有软硬件资源和参与的管理员、用户综合起来，认为它们都是系统的一部分。从这个角度来说，目前的在线评测系统仍然存在以下缺点。

1.1 系统程序的获得困难

在线评测系统的设计和开发涉及到操作系统进程调度、Web界面、数据库管理、大规模并发访问等诸多方面的计算机应用技术。该系统开发难度大、要求高，而且设计上一旦存在漏洞将严重危害系统安全；许多学校在自行过开发后，都因为各种原因而没有得以大范围应用。国内少数高校开发和使用的系统，虽然对系统源代码进行了开放，但是由于文档不足且缺乏技术支持，其源码很难进行二次利用，应用范围也非常有限，许多只有源码没有文档的系统因为难于安装配置而无法推广。

1.2 系统平台的搭建困难

国内外在线评测系统普遍基于Linux平台进行开发，在给开发带来便利的同时，也给系统的使用带来巨大障碍。虽然近几年国内的Linux社区发展迅速，但是由于盗版的根深蒂固和计算机教育对Windows系统的全面迎合，国内高校真正能够熟练使用Linux操作系统平台的教师和学生并不多，即使取得了系统程序的源代码，也很难实际部署成功。

1.3 系统运行与维护的困难

在线评测系统因为需要实时对使用者提交的源代码进行编译、运行、监控，同时又要完成大量Web请求的并发访问，所以对服务器系统的性能要求极高。普通的计算机服务器虽然可以进行小规模的练习，但对于在线人数成百上千的大型比赛就无力应对。虽然可以对Web服务器、数据库服务器、判题服务器进行分机运行，实现简单的分布式计算，但是对于普通大学和科研机构，购买数量繁多的服务器又是一个经济上难以承受的负担。

1.4 练习题目获得的困难

因为在线评测系统基于黑盒测试技术，所以测试数据对于题目而言就是必不可少的基本要素。目前许多高校的在线评测系统对外提供题目查看，但是其测试数据都严格保密。目前，在线评测系统使用上的便利性一定程度上是建立在题目编写的困难性之上的，即使是一流大学的教授编写一道新题目也要付出非常艰辛的劳动。这就是像北大这样的一流大

本文受浙江传媒学院《程序设计类教学辅导平台设计与开发》教改项目，HUSTOJ社区用户资助。
张浩斌(1980—)，男，硕士，讲师，主要研究领域为计算机应用技术、互联网应用技术，E-mail:newsclan@gmail.com。

万方数据

学为什么会在互联网上重金征集新颖题目的原因[3]。

针对以上存在的问题和困难，本文提出了建立基于开放云平台的开源在线评测系统，为高校和科研机构使用在线评测系统提供了新的途径。

2 开源在线评测系统

2.1 开源在线评测系统现状

互联网上可以访问到的在线评测系统数量很多，开放源代码的在线评测系统数量则相对有限。在开源的系统中，很多是属于学校本科生以毕业设计为目的而开发的实验性系统，因而缺乏实用价值。国内知名的在线评测系统中，浙江大学在线评测系统(ZOJ)[4]和北京交通大学在线评测系统(BOJ)[5]都进行了开源的尝试。但是开源不仅仅是开放源代码，还需要有文档、社区和有效的技术支持；由于文档的缺失和代码更新缓慢，上述两所知名大学开发的在线评测系统几乎没有其他学校使用。而且 ZOJ 使用的是 J2EE 框架和基于 Socket 的 C/S 结构的分布式应用，其系统复杂程度和对硬件系统的要求都比较高，很难在开放式云平台进行部署。而其他小型的开源评测系统，则往往缺失可扩展性，对于分布式应用或云平台的使用都存在基础性缺陷。

2.2 开源在线评测系统改进

HUSTOJ 是由华中科技大学毕业生王良晶等开发并在 2008 年底以 GPL 协议进行开源的一个在线评测系统[6]，它采用 LAMP 平台进行开发，其早期版本在华中科技大学的服务器上稳定运行至今。早期的 HUSTOJ 在开源方面与 ZOJ、BOJ 一样，缺少应有的文档和技术支持，甚至没有提供数据库结构，使得在开源后长达一年的时间中没有任何其他的学校和组织进行部署。作者从 2009 年 9 月开始加入开源在线评测系统 HUSTOJ 的开发，并在创始人的支持下成为 HUSTOJ 的项目管理员。为了使 HUSTOJ 更容易被其他院校有效使用，真正实现开源共享，作者对 HUSTOJ 系统进行了大量改进，主要包括：

1)开发了安装脚本，实现命令方式的安装，并编写了详细的安装步骤文档。

2)增加了可视化的题目管理界面，采用 FCKEditor 实现了超文本编辑和图片上传功能。

3)重构和改写了判题内核程序，提高了系统代码的可读性和多线程判题的性能，对多核心处理器有更好的支持。

4)在原有沙箱技术上进行扩充，增加了 Ruby、Bash、Python、Perl、C#、PHP 等多种编程语言的支持。

5)增加了对多国语言支持的机制，吸引了国外用户和开发者的关注。

6)创建论坛和即时通讯群，并组织创建了用户社区。

7)为用户提供免费的咨询和收费的在线安装、维护服务。

8)开发了基于 Ubuntu 的 LiveCD 系统，使用户无需安装即可试用其完整功能，并可以图形化方式完成系统安装，降低了使用门槛。

9)开发了集成虚拟机的 Windows 版本，使得 Windows 用户也可以免安装使用。

改进后的 HUSTOJ 主体架构分为判题内核 Judged 和 Web 界面两部分，它们通过公用一个 MySQL 服务器实现任务队列和数据通信。判题服务 Judged 是一个系统服务进程，随操作系统一起启动，并随时监视数据库服务器中的任务队列；Web 界面由一组 PHP 程序构成，运行在 Apache 服务器上，也可以支持在 Nginx、lighthttpd 等目前流行的非阻塞式 Web 服务器上运行。当用户提交新的答案后，Web 程序就将判题任务加入数据库的任务队列，随后判题服务程序 Judged 则自动锁定任务并完成，可以防止同一任务被多次判断而浪费系统资源。其系统结构如图 1 所示，其中 MySQL 服务器、Web 服务器、判题服务 Judged 既可以由同一台服务器承担，也可以分别由不同的服务承担，以实现分布式计算。Web 服务器可以采用 Apache 集群，MySQL 服务器可以使用主从式结构，Judged 服务器可以用提交编号静态分配判题任务，也可以通过数据库的锁机制实现动态任务分配。这样的设计不但能够适应日常教学的小范围使用，而且可以从多个层面扩大服务器集群的规模，高效应对大规模的并发服务[7]。这种可扩充的结构设计，也使得 HUSTOJ 相对于其他在线评测系统能够更容易地部署在开放式的云平台之上。相关技术细节请参考作者在文献[6]中提供的源代码。

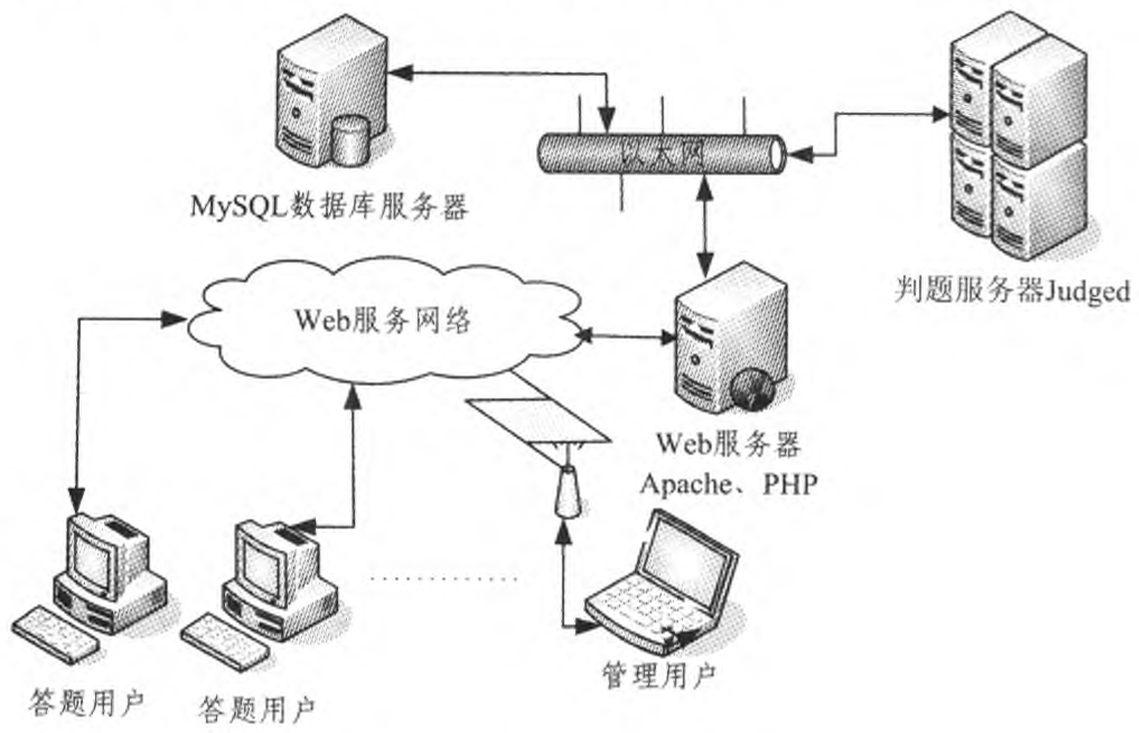

图 1 HUSTOJ 系统结构

在作者和社区的努力下，现在有近 50 所高校、科研机构采用了改进的 HUSTOJ 搭建自己的在线评测系统，用户涉及中国大陆、香港、台湾、韩国、美国、马其顿、巴西、伊朗等多个国家和地区。HUSTOJ 项目的发展壮大，系统代码不断升级更新，有效地解决了系统代码获取和平台搭建这两大难题。

3 开放式云平台

3.1 开放式云平台现状

开放式云平台是近几年由各大互联网巨头提出的新型分布式计算应用平台。首先，由资金雄厚、技术实力强的互联网公司设计和搭建一个庞大的基础结构，其包含分布在世界各地的成千上万的计算节点构成的分布式服务网络。然后，由系统构建方用一种透明的方式发布给需要搭建 Web 服务的个人、公司和组织。开放式云平台为个人、中小型机构的高性能 Web 服务搭建提供了一种经济、可靠的实现方式。这正符合高校和其他科研机构对于搭建高性能在线评测系统所提出的要求。

万方数据

3.2 新浪开放云平台

国外最流行的云平台有亚马逊[8]和谷歌[9]各自提供的云平台，国内目前较好的开放式云平台服务当属知名上市公司新浪提供的 SAE(Sina App Engine)[10]。新浪云平台提供国内 Web 开发最为流行的 PHP5、MySQL5、Memcache、Crontab 等服务，相对于国外平台其最大的优势在于系统几乎完全兼容常见的 LAMP(Linux、Apache、MySQL、PHP)平台，许多 PHP 应用程序几乎不用修改或做少量修改就可以在新浪云流畅运行。这给在线评测系统从非云平台向云平台进行移植带来了极大便利。图 2 展示的是新浪云平台的整体架构，可以看出，新浪云平台通过 Service Router 将 Web Service Pools 中的底层服务与应用完全隔离，实现了运行中系统资源的无缝调配，可以在不影响应用运行的情况下动态地对参与服务的节点进行增加和删除。而成熟的反向代理机制又极大地保证了系统服务的效率，在提高响应速度的同时也有效降低了基础服务的运行负荷。

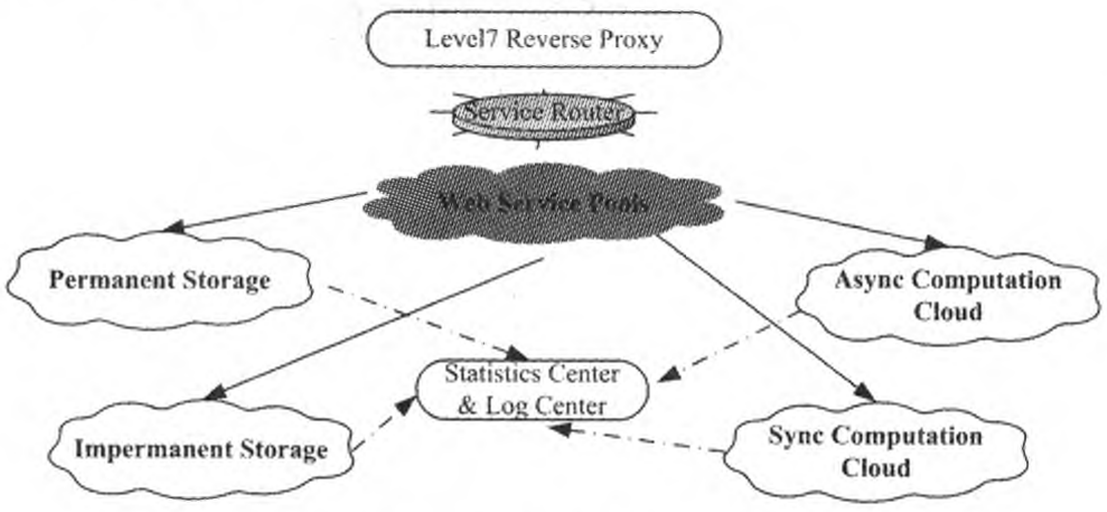

图 2 SAE 整体架构图[11]

Web 服务池由一些不同特性的 Web 服务池组成。每个 Web 服务池实际是由一组 Apache Server 组成的。每个 Web 服务进程实际处理用户的 HTTP 请求，进程运行在 HTTP 服务沙盒内，同时还内嵌同样运行在 SAE 沙盒内的 PHP 解析引擎。用户的代码最终通过接口调用各种服务。

日志和统计中心：负责对用户所使用的所有服务的配额进行统计和资源计费，通过以云豆方式的配额机制来保证整个平台的稳定，同时还提供了计费手段[11]。

此外，图中其他云型标识代表各种分布式服务，如存储服务、计算服务等，还可以不断扩充新的服务与功能。

新浪的云计算平台是分布式计算的一种新形式和新应用，但是相对于普通的分布式计算而言，它具有价格低廉(对低端用户基本免费)、可扩展性更强、性能更高的特点。一般高校如果要完全自行搭建类似的分布式服务平台，需要巨额的初期投入和长期的维护投入才能够实现类似的效果。

4 在开放云平台应用开源评测系统的问题及对策

4.1 在开放云平台应用开源评测系统的问题

虽然新浪云平台几乎可以完美兼容 LAMP 平台，但是由于其云平台的特殊性，直接将 HUSTOJ 部署在 SAE 上仍然会遇到有以下几个问题：

1)SAE 只支持 Web 平台，不支持 Linux 编译运行环境，因此只适用于 HUSTOJ 的 Web 服务界面运行，无法进行编译和判题。这就要求判题端必须与 SAE 通过网络进行数据传输、任务转发。但是，鉴于安全上的考虑和技术上的限制，HUSTOJ 所使用的 MySQL 远程访问在 SAE 上是不对外开放的。在图 2 中也可以看到，Service Router 将后台分布式服务透明化的同时，也将 MySQL 封装在 SAE 内部，使其不能通过互联网进行访问。

2)SAE 对 MySQL 的使用有着严格的限制，不允许频繁地使用耗时长、扫描行数多的查询，并对此类查询进行了软硬限制。一旦超越限制，将导致服务被暂停。

3)SAE 不支持文件系统的访问，原有的 HUSTOJ 代码中存在很多访问文件系统的代码，而当这些代码运行失败时就会导致部分功能的失效。

4.2 开源评测系统针对开放云平台的改进

4.2.1 基于 HTTP 的远程过程调用

从通信角度考虑，无论是 ZOJ 的 Socket 通信还是 HUSTOJ 的 MySQL 连接通信，都是通过网络对任务数据进行访问。具体使用什么样的网络，采用什么样的协议，都只是一个形式问题。这样一来，4.1 节中问题 1)就转化成了如何从新浪云平台外部的网络访问其内部 MySQL 数据库的问题，而后者有一个显而易见的答案，即 SAE 的 Web 服务。所以，如果能够分别编写运行在新浪云平台的 PHP 程序和运行在判题服务器的 Web 客户端程序，就可以实现经由 Web 服务访问 MySQL 数据。这种操作机制与 JSON[12]类似，可以看作是一种基于 HTTP 协议的远程过程调用(RPC)。因为判题系统承担着判断用户答案正确性的关键任务，所以它和 SAE 的 Web 服务的通信过程必须有身份认证机制。作者采用 Unix 传统的简单设计组合复用的思想，直接使用了基于 Cookie 的 PHP Session 机制来实现身份认证；在 Web 客户端实现上以子进程方式复用了知名的开源命令行 Web 客户端 WGET[13]。图 3 描述了这一解决方案。

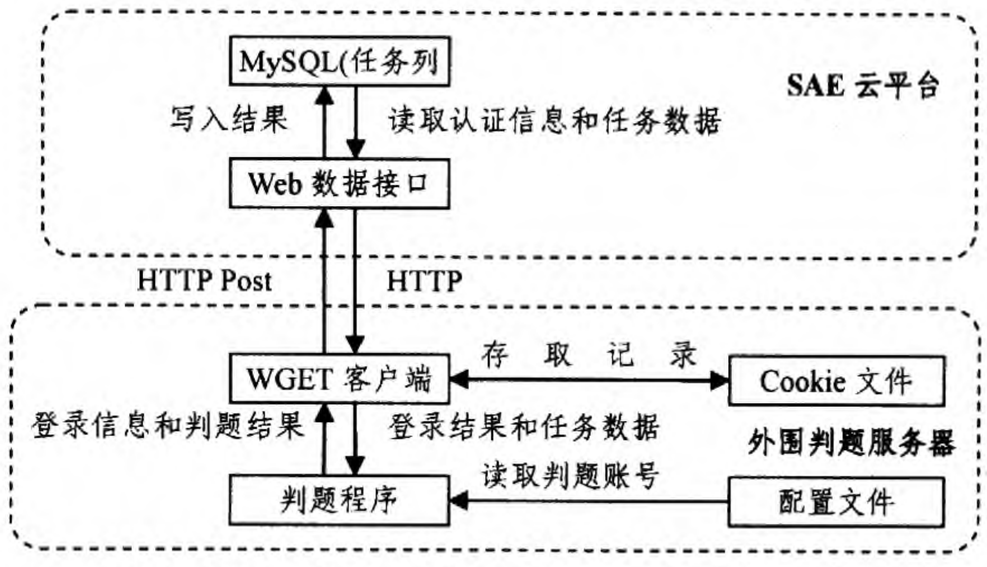

图 3 判题服务器与 SAE 云平台的带认证 HTTP 数据接口

在实际运行中，云平台上的判题任务数量比普通的在线评测系统更多，因此可以大量部署判题服务器。因为判题服务仅需要 HTTP 通道即可与 SAE 完成远程调用，所以判题服务器不必具有公网 IP 地址，只需要最基础的 Web 浏览能力即可。这样的要求在学校机房很容易满足，可以实现快速部署大量判题服务器。

为了方便这一操作，作者在 Ubuntu Desktop 的基础上制定了 HUSTOJ 专用 LiveCD，即可以在任何 X86 体系结构的 PC 机上用一张光盘或虚拟机快速启动出 HUSTOJ 完整的评测系统，只需要通过网络传输少量的测试数据，就可成为新的判题节点。在添加判题节点的过程中，完全不影响 SAE 上的 Web 服务，实现了判题节点的动态增加和删除，使得系统规

万方数据

模能够很容易得到扩充。

4.2.2 基于内存缓存技术的页面加速

SAE 虽然限制了 MySQL 服务的使用量，却提供了另一种性能更为优秀的存储机制——Memcache。Memcache 是目前互联网服务领域广泛使用的基于内存 Hash 机制的缓冲系统[14]，因为其数据完全存储在内存并采用 Hash 算法进行键-值(Key-Value)检索，所以它具有磁盘存储系统无法比拟的存取速度。但是也正由于这一点，Memcache 的使用成本更高，而且不支持 SQL 的关系化查询方式。Memcache 系统的这些特点使得它非常适合作为高速缓冲机制。为了充分发挥 Memcache 的缓冲作用，降低 MySQL 负荷，作者又对 HUSTOJ 做出了两个方面的改进。

1)使用 Memcache 缓存 MYSQL 查询

因为 HUSTOJ 对于数据库的查询都是通过 SQL 语句来实现的，所以可以在查询 MySQL 的时候使用 SQL 语句的 MD5 散列值作为键值，将查询到的结果放入 Memcache 中。在下一次查询的时候，先在 Memcache 中查找缓存数据，如果存在有效数据，就不再进行 MySQL 查询，而直接使用 Memcache 缓存的数据。

2)使用 Memcache 直接缓存页面

HUSTOJ 的 PHP 代码调用 MySQL 进行查询，最终目的是生成 HTML 文本，并反馈给浏览器客户端。而生成一个页面往往需要进行多次、大量的查询使得页面打开速度变慢，用户在等待时常常使用浏览器的 F5 快捷键刷新页面，就造成了更大的查询压力，这也是移植初期 MySQL 常常超出配额的原因。所以如果能够将页面代码直接缓存到 Memcache 中，那么不但降低了 MySQL 的压力，而且还能减少 PHP 代码运行的次数和时间，可以进一步提高页面访问速度。为了实现尽可能多的页面缓存，作者在所有页面的首尾都加入了使用 Memcache 进行缓存的代码，并且对每个页面设置了个性化的缓存周期，以保证缓存不会给用户访问数据的即时性带来太多影响。

启用 Memcache 功能前、后，用 ApacheBench 模拟 500 个并发用户对系统中使用最为频繁的 status. php 进行 1000 次访问，得到的测试结果如表 1 所列，其中左侧为启用之前的测试结果，右侧为启用之后的测试结果。

表 1 使用 Memcache 前后的测试数据对比

启用 Memcache 之前	启用 Memcache 之后
Server Software: Apache/2.2.14 Server Hostname: 127.0.0.1 Server Port: 80 Document Path: /JudgeOnline/status.php Document Length: 8772 bytes Concurrency Level: 500 Time taken for tests: 32.239 seconds Complete requests: 1000 Failed requests: 207 (Connect: 0, Receive: 0, Length: 207, Exceptions: 0) Write errors: 0 Total transferred: 7262294 bytes HTML transferred: 6956196 bytes Requests per second: 31.02 [#/sec](mean) Time per request: 16119.251 [ms](mean) Time per request: 32.239 [ms](mean, across all concurrent requests) Transfer rate: 219.99 [kbytes/sec]received Connection Times (ms) min mean[+/-sd] median max Connect: 1 390 388.8 229 3000 Processing: 296 7872 8889.2 2009 32200 Waiting: 0 3371 5151.1 1002 15725 Total: 358 8262 8683.9 2687 32237 Percentage of the requests served within a certain time (ms) 50% 2687 66% 13366 75% 14030 80% 14341 90% 15726 95% 31406 98% 31993 99% 32219 100% 32237 (longest request)Server Software:	Server Software: Apache/2.2.14 Server Hostname: 127.0.0.1 Server Port: 80 Document Path: /JudgeOnline/status.php Document Length: 8753 bytes Concurrency Level: 500 Time taken for tests: 8.777 seconds Complete requests: 1000 Failed requests: 250 (Connect: 0, Receive: 0, Length: 250, Exceptions: 0) Write errors: 0 Total transferred: 9363105 bytes HTML transferred: 8967841 bytes Requests per second: 113.93 [#/sec](mean) Time per request: 4388.660 [ms](mean) Time per request: 8.777 [ms](mean, across all concurrent requests) Transfer rate: 1041.74 [kbytes/sec]received Connection Times (ms) min mean[+/-sd] median max Connect: 9 241 228.2 178 856 Processing: 359 1749 2590.3 768 8743 Waiting: 5 1553 2646.1 484 8639 Total: 382 1991 2507.9 1006 8775 Percentage of the requests served within a certain time (ms) 50% 1006 66% 1326 75% 1397 80% 1534 90% 8494 95% 8610 98% 8766 99% 8773 100% 8775 (longest request)Server Software:

从表 1 中可以看到，启用 Memcache 后页面打开速度提高到启用前的 367%，极大地提高了系统的响应速度。这里需要说明两点：1)由于 status. php 涉及的表为系统中条目数量最多的表，因此记录的数量会影响测试的结果；数据条数越多，前后差异越大，缓存效果越明显。2)由于 SAE 平台能够根据访问量动态调整服务的参与节点数量，并且在不启用 Memcache 的情况下少量的并发访问就足以触发配额限制，导致系统暂停服务，因此直接在 SAE 进行测试很难得到有效数据。基于以上两点，表 1 中的测试数据是使用了浙江传媒学院内网教学用评测系统两年左右的实际运行数据在一台

万方数据

IBM ThinkPad T43p 上完成的模拟测试，使用的操作系统平台是 Ubuntu 10.4 LTS，相关服务器软件均为 Ubuntu 软件库中最新版本。

4.2.3 文件系统限用兼容性

针对 SAE 文件系统访问受限的问题，作者在 HUSTOJ 的配置文件中设置了 SAE 运行标识，当运行在 SAE 上时就不再运行会出错的文件操作，巧妙地回避了这一问题。另一方面，为了实现 SAE 系统数据的导入、导出，可以采用在本地安装镜像系统的方式。即在本地服务器上安装和配置与 SAE 上完全相同的代码和数据，当需要向 SAE 添加文件、增加数据时，先在本地系统进行操作，然后使用 SVN 工具完成本地系统到远程系统的差量更新，这样既可以间接保留 HUSTOJ 本地运行时的很多文件处理功能，又可以作为 SAE 的数据、代码备份。

5 基于 XML 的开源题目交换格式

5.1 题目来源问题分析

要解决题目来源的问题，首先是充分利用现有的题目资源，最重要的是实现学校之间的题目共享，特别是题目的测试数据。在这个问题上，有情理和技术两方面的障碍。首先，因为在线评测系统是基于黑盒测试的，这就意味着，如果有了测试数据，理论上可以编写出能够骗过判题程序而没有完成指定功能的答案。这种可能性的存在，使得使用评测系统进行辅助教学时就有方便学生抄袭、舞弊的风险。其次，因为各个大学自行开发在线评测系统，其数据库结构、文件组织方式都不尽相同，即使他们打破界限，愿意共享数据，也会因为没有统一的标准而使得数据交换的过程涉及大量人力操作，带来各种人为的意外错误，这一点作者在整理题目的过程中深有感触。

5.2 基于 XML 的数据交换格式设计

在互联网领域中，解决异构平台数据交换的首选工具就是 XML[15]。由于 XML 的开放性和跨平台特性非常优秀，几乎所有的主流编程语言都对 XML 提供支持[16]，因此无论是哪一所大学用何种主流语言编写的在线评测系统都可以轻易地兼容这种文件格式[17]。作者为了保证这种数据交换格式的独立性，打消其他评测系统在兼容时的顾虑，特意在 Google 的开源项目平台上建立了新的 GPL 项目 FreeProblemSet(FPS)[18]。FPS 基于 XML，结合 Base64 编码技术[19]集成了题目中图片数据的存储，完全开放其结构定义。作者花费大量业余时间从网络上、教材中搜集整理了超过 400 道各类中英文题目，以 FPS 格式共享在 Google 平台上，供广大师生使用，目前累计下载 12227 人次。

5.3 抄袭侦测系统对 FPS 交换机制的补充

FPS 的实现为大学之间的题目数据共享扫清了技术障碍，但是仍有数据泄漏评测失效的顾虑，针对这一问题，作者也提出了技术上的应对方式——代码相似度检测即反抄袭系统。因为计算机编程语言与人类的自然语言类似，但又有结构严谨的特点，所以对程序代码的相似度检测与近些年的论文抄袭检测系统原理上类似，又相对更加容易实现。出于解决问题的紧迫性和重要性，作者并没有从头开始编写语法解析器和词条统计等工具，而是在互联网上寻找可以二次利用的类似程序。最后，作者找到了荷兰阿姆斯特丹 Vrije 大学的 Dick Grune 教授编写的 software and text similarity tester (SIM)[20]，并顺利地取得了 Grune 教授的允许，将 SIM 集成到 HUSTOJ 中，实现代码相似度的检测，这样即使测试数据泄漏，也只有极个别人能够用它骗过系统，如果有更多的人使用，就会被相似度检测程序发现并报告出来。在集成了 SIM 系统以后，在线评测系统不但克服了交换题目带来的数据泄漏问题，而且还能够有效地侦测到学生作业抄袭、考试作弊的问题。集成时采用了基于命令行参数的进程调用接口，而没有修改 SIM 的任何代码，因此如果有需要可以很容易地将该部分转换为其他类似的相似度检测程序。

6 基于开放云平台的开源在线评测系统的实际应用

为了验证基于 SAE 的 HUSTOJ 系统的可行性，作者在新浪云平台开设了演示性在线评测平台，地址是 http://hustoj.sinaapp.com。因为所有相关代码全部在 HUSTOJ 系统主线代码上进行开发，所以本文所述的所有代码实现均已同步开源[6]。任何人都可以通过互联网查看到开发过程中的每一次修改和更新的详细细节。如对本文所述技术细节存有疑问，也可通过项目主页上的联系方式进行咨询探讨。

九度社区是一家专注于计算机学生的社区网站，因为有越来越多的大学在研究生复试中采用在线评测系统，所以九度社区的创办方试图为其用户提供各校往年复试题目的在线评测服务。由于在线评测系统的复杂性和用户规模的庞大，其创办方最终选择了基于新浪开放云平台的 HUSTOJ 作为基础进行二次开发和系统集成，他们通过互联网联系到本文作者，在作者的帮助下最终实现了该平台的商业化运行。其服务器在自有域名下的地址是 http://ac.jobdu.com，在新浪云平台的二级域名是 http://jobdu.sinaapp.com。

结束语 综上所述，基于开放云平台的开源在线评测系统可以满足从日常教学到大规模在线竞赛的应用需求，不但能够提供更好的使用体验，而且从系统实现、系统搭建、运维平台及题目的获得 4 个方面解决了大学科研机构搭建在线评测系统的困难。重点针对新浪云平台进行了系统改造，其相对于普通的分布式 Web 服务平台具有扩展性更强、成本更低的技术优势。从理论上论证并用实际代码和实际系统检验了系统的可行性，同时基于 GPL 协议开放了全部源代码，为在线评测系统的改进和推广找到了新途径。

参考文献

[1] 曾宗根. 源程序在线评测系统技术改进[J]. 计算机工程与应用，2011，4：68-71

[2] 陈湘骥，徐东风，杨秋妹. 在线评判在 C 语言课程设计教学中的应用[J]. 计算机教育，2010，3：97-100

[3] 北京大学在线评测系统. 题目有偿征集 [EB/OL]. http://poj.org/challenge.html，2011-09-02

[4] 浙江大学在线评测系统. 开源主页[EB/OL]. http://code.google.com/p/zoj/，2011-09-02

[5] 北京交通大学在线评测系统. 开源主页[EB/OL]. http://code.google.com/p/open-boj/，2011-09-02

（下转第 348 页）

万方数据

RAID5＞RAID6，其中，RAID1 是直接从镜像盘读取数据，RAID3、RAID5 和 RAID6 需要进行一次校验操作来得到损坏的数据。在实际情况中，由于阵列控制器的不同，RAID1 的相对位置可能不同，这与校验操作的处理时间有关。

由于 RAID6 和 RAID5x 的容错度为 2，因此还进行了 RAID6 和 RAID5x 损坏两个磁盘的读性能测试。

在损毁两个磁盘的情况下，RAID6 读性能要次于 RAID5x，因为 RAID6 需要将两个校验式联立求解，计算过程需要耗费大量的 CPU 资源，这个操作计算过程非常复杂，一般需要独立的硬件来处理。其结果如表 4 所列。

表 4　损坏两个盘的读性能测试(单位：MB/s)

请求	Ca	R0	R1	R3	R5	R6	R5x
1. txt	—	—	—	—	—	2.7	2.6
2. ppt	—	—	—	—	—	4.9	6.3
3. doc	—	—	—	—	—	7.7	10.8
4. pdf	—	—	—	—	—	6.6	12.7
5. bmp	—	—	—	—	—	6.2	10.4
6. jpg	—	—	—	—	—	7.6	11.8
7. mp3	—	—	—	—	—	5.5	12.7
8. wm	—	—	—	—	—	6.2	13.3
9. flv	—	—	—	—	—	6.3	11.5
10. mv	—	—	—	—	—	6.4	11.9

RAID5x 数据布局采用镜像垂直校验交错放置：磁盘数为 N，$N-2$ 个磁盘的数据分带进行校验计算，得到一个校验分带子段对应存放到剩下的两个磁盘中的一个，另一个连续存在这 $N-2$ 个数据分带子段的副本，容许双磁盘故障失效错误。RAID5x 可以从镜像中读取数据，在损毁两个磁盘的情况下，一般只需进行一次异或校验，在一些特定的数据布局下，RAID5x 可以直接从镜像中将对应的数据读出，不需进行校验，如图 5 所示。如果磁盘 0 与磁盘 1 两块磁盘坏掉，则直接从磁盘 2 与磁盘 3 中的镜像区将备份数据读出即可，不需要校验。

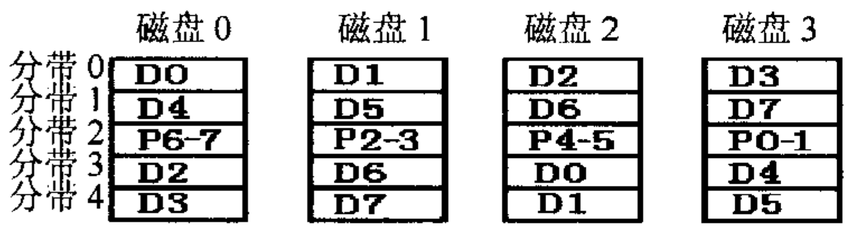

图 5　一种磁盘个数为 4 的 RAID5x 的数据布局

结束语　测试结果验证了，在正常情况下 RAID0 的请求处理性能最高，但其不具备容错性；在损坏一个磁盘的情况下，RAID5x 请求处理性能最高，因其直接从镜像盘中将损坏的数据读出，不需校验操作，同时实现了磁盘的并行处理；在损毁两个磁盘的情况下，RAID5x 的性能也要优于 RAID6，因为 RAID5x 数据布局采用镜像垂直校验交错放置，一般只需进行一次异或校验，在一些特定的数据布局下，RAID5x 可以直接从镜像中将对应的数据读出，不需进行校验，而 RAID6 需要将两个校验式联立求解，计算过程需要耗费大量的 CPU 资源。所以，在需要双故障容错的情况下，RAID5x 是一个不错的选择。

该软件为磁盘阵列性能、可靠性以及阵列优化方法的研究提供了一个平台，也为以后网络存储、云存储等存储系统的研究和模拟奠定了基础。软件现已完成主要功能，但由于时间的限制，系统仍有一些需要改进、优化之处，例如，磁盘模块没有考虑固态硬盘，阵列的重建过程暂未实现，暂未加入 Cache 的对存储系统性能的影响，里德-所罗门编码的处理等。

参 考 文 献

[1] 张锋. 计算机仿真技术及其应用[J]. 人工智能及识别技术，2007(09)：233-234

[2] Bucy J S，Schindler J，Schlosser S W，et al. The DiskSim Simulation Environment Version4. 0 Reference Manual[M]. CMU-PDL-08-101，May 2008

[3] 田磊，冯丹. RAID 在线数据重建方法仿真器设计[J]. 华中科技大学学报：自然科学版，2010，38(5)

[4] Hu Ming，Jiang Ming-hua. RAID5x：A Performance-optimizing Scheme against Double Disk Failures[C]//Proceedings of 2006 International Symposium on Distributed Computing and Applications for Business，Engineering，and Science (DCABES 2006). Shanghai University Press，Volume II，ISBN 7-8118-023-5/TP.037，2006：1060-1063

[5] 胡鸣. 磁盘阵列评估与优化方法研究[D]. 武汉：华中科技大学，2005

[6] Worthington B L，Ganger G R. Scheduling algorithms for modern disk drives[J]. ACM SIGMETRICS Performance，1994

[7] Lee E K，Katz R H. An Analytic Performance Model of Disk Arrays[J]. ACM SIGMETRICS Performance Evaluation Review，1993

(上接第 343 页)

[6] 华中科技大学在线评测系统. 开源主页[EB/OL]. http://hustoj. googlecode. com，2011-09-02

[7] 管赋胜. 基于集群技术的源代码自动评测系统研究[J]. 硅谷，2009(16)：48-49

[8] 亚马逊云平台. 主页[EB/OL]. http://aws. amazon. com/，2011-09-02

[9] 谷歌云平台. 主页[EB/OL]. http://code. google. com/intl/zh-CN/appengine/，2011-09-02

[10] 新浪云平台. 主页[EB/OL]. http://sae. sina. com. cn/，2011-09-02

[11] 新浪云平台. 运行原理[EB/OL]. http://sae. sina. com. cn/? m=devcenter&catId=164，2011-10-27

[12] 李德贤，陆歌皓，姚绍文. JSON-RPC 协议分析、扩展及其应用[J]. 中国科技论文在线，2008，2：125-130

[13] 命令行客户端 WGET. 主页[EB/OL]. http://www. gnu. org/software/wget/，2011-09-02

[14] 杨立身，曹志义. 内存缓存技术在门户网站开发中的应用研究[J]. 电脑知识与技术，2008，25：1415-1419

[15] 何国辉. 卿银波. 基于 XML 的数据交换系统设计[J]. 计算机工程与设计，2007，3：583-587

[16] 何国辉，卿银波. 基于 XML 的数据交换系统设计[J]. 计算机工程与设计，2007，3：583-587

[17] 周娟，周尚超，谭炳文. 基于 JSP 和 XML 的在线裁判系统[J]. 计算机应用与软件，2009，12：177-178，181

[18] FreeProblemSet. 开源主页[EB/OL]. http://freeproblemset. googlecode. com/，2011-09-02

[19] 黄蓉刚. 基于 XML 和 Base64 方法实现通用通信平台[J]. 微计算机信息，2009，6：137-138

[20] Software and text similarity tester. HomePage [EB/OL]. http://www. dickgrune. com/Programs/similarity_tester/，2011-09-02

万方数据

万方数据 WANFANG DATA 文献链接

基于开放式云平台的开源在线评测系统设计与实现

作者：张浩斌， ZHANG Hao-bin
作者单位：浙江传媒学院新媒体学院 杭州310018
刊名：计算机科学 ISTIC PKU
英文刊名：Computer Science
年，卷(期)：2012,39(z3)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_jsjkx2012z3089.aspx