# 前言

本文档使用Word 2016编辑，大家可以借助导航功能来方便快捷地找到自己想要的东西。Hustoj作为一个开源的项目，帮助到了很多学校建立起了属于自己的在线ACM评测系统，在这里向作者zhblue致敬。由于存在网上资料零散，以及作者blog和github\_wiki中的东西有大量重叠及时效性的问题，在这里我将各方面资料以及我实际安装运行中的各种问题汇总起来，供大家参考。

我作为受益于开源项目者，愿意贡献出自己的一份力量。**为了方便大家复制代码，我以未加密的Word文档形式发布，转载请保持完整性，对于窃取它人劳动成果的人加以鄙视。**

文中使用了夏夏的数据库解读、bigballon的vjudge搭建流程、zhblue老师的github\_wiki及个人博客，还有我自己的一点润色以及使用随笔等。如有不足，敬请指正。

字体：正文为微软雅黑 五号，标题1为微软雅黑 小二，标题2为微软雅黑 四号，标题3为微软雅黑 小四，代码为*Consolas 小五 斜体*，注释为正文+红色。

qq2663797538@gmail.com，更新于2017年2月5日

# HUSTOJ结构简析

HUSTOJ分为两大部分，core和web，分别对应判题和数据管理两大功能。

core和web之间数据交换有两种方式：1、通过数据库，轮询。2、通过wget实现的http请求。两种方式的选择在判题端的配置文件/home/judge/etc/judge.conf中，HTTP\_JUDGE=1则启用后者，默认为前者。

## 一．Core解析

core分3部分，judged、judge\_client、sim。源码解读见以下链接：

<http://blog.csdn.net/legan/article/details/40746829>

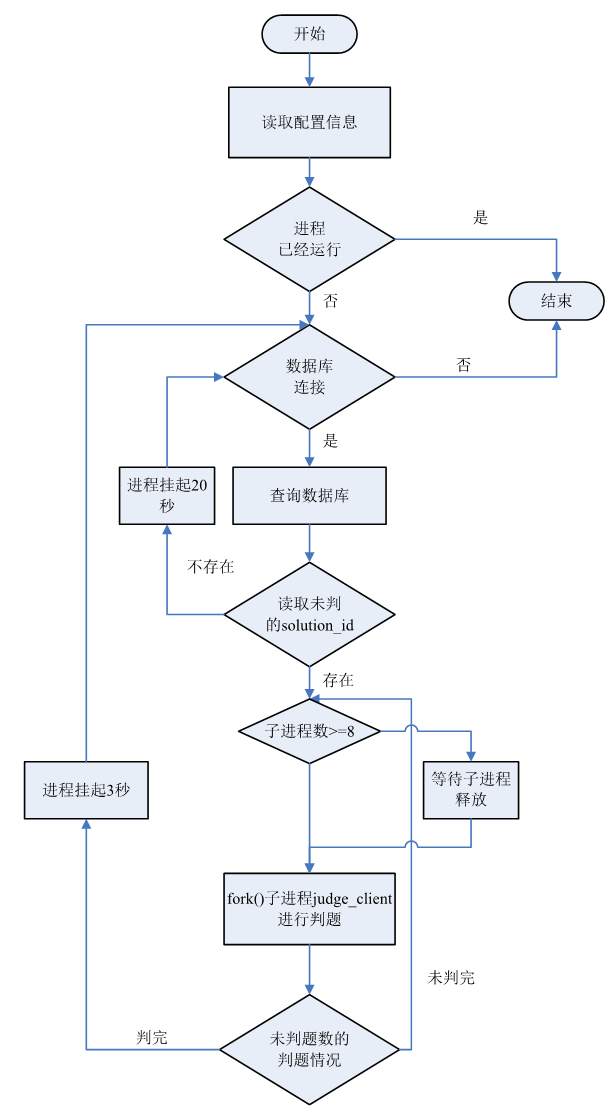
<http://blog.csdn.net/legan/article/details/40789939>

### judged

##### 简介

judged为服务进程，d即daemon。负责轮询数据库或web端，提取判题队列。单个主机运行多个judged，分别负责不同的OJ判题。

##### 工作流程图



##### 基本参数

judged 可以接受一个参数作为自己的主目录，默认是/home/judge/。

如：*sudo judged /home/judge/local*

不指定参数将自动以单进程运行。当指定的参数不为/home/judge时，就会有多个进程出现。

每个主目录可以有自己的etc/judge.conf 数据目录可以共享，runX目录需要独立。

##### judged调试模式

judged 接受参数指定目录的情况下，还可以再接受一个debug作为调试模式开关。

如：*sudo judged /home/judge/local debug*

调试模式的judged将不会进入后台，并且将输出大量调试信息，其调用的judge\_client也工作在debug模式。

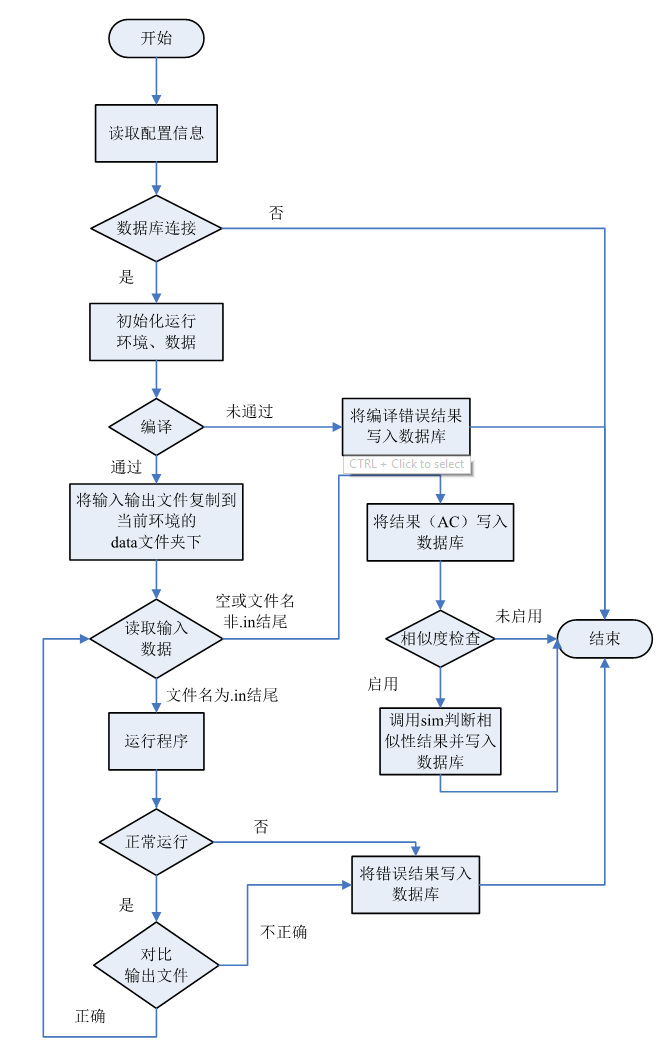
### judge\_client

##### 简介

当发现新任务时产生judge\_client进程。

judge\_client进程为实际判题程序，负责准备运行环境、数据，运行并监控目标程序的系统调用，采集运行指标，判断运行结果。

##### 工作流程图



##### 后台安全机制

在UNIX中有上百个[<http://www.ibm.com/developerworks/cn/linux/kernel/syscall/part1/appendix.html> 系统调用]，有一大部分是在用户程序运行过程中不需要的，比如说mkdir,mount等，还有一部分会对系统造成安全隐患的，比如fork,kill,exec等，还有一些比如socket等会造成敏感信息，比如测试数据的泄漏等。 因为以上情况的存在，所以需要在运行用户程序的时候对用户加以限制，linux下的ptrace在这里是一个非常好用的工具，它可以在用户态和内核态之间切换之前和之后，将进程暂停，以方便控制进程的处理，控制进程通过ptrace可以读取到当前进程想要去做什么，这样就可以在用户程序造成破坏之前将程序中止。 限制非法系统调用，最好的办法是使用白名单机制，只允许程序使用一个小集合里的调用，对于其它调用，即使它是安全的，也不会被允许，比如mkdir。 由于Pascal,Java,C/C++的机制有些区别，因此，三种不同语言的白名单各不相同。

##### judge\_client调试模式

judge\_client接受参数

judge\_client [工作主目录] [调试]

如：judge\_client 2001 5 /home/judge/demo debug

将在/home/judge/demo/run5目录中对2001号提交进行重判，并打开调试模式，输出大量调试信息，运行后不删除中间结果。

这个模式可以帮助调试题目数据，发现数据问题和了解提交RE的详细错误原因

### sim

##### 简介

当配置为启用抄袭检查时，judge\_client将调用sim，判断相似性结果，并写回数据库或web端。

sim为第三方应用程序，可进行语法分析判断文本相似度，通过检验的程序将由judge\_client复制进题目数据的ac目录，成为新的参考样本。

### 配置文件judge.conf注释

*OJ\_HOST\_NAME=127.0.0.1* 如果用mysql连接读取数据库，数据库的主机地址  
*OJ\_USER\_NAME=root* 数据库帐号 *OJ\_PASSWORD=root* 数据库密码 *OJ\_DB\_NAME=jol* 数据库名称 *OJ\_PORT\_NUMBER=3306* 数据库端口 *OJ\_RUNNING=4* judged会启动judge\_client判题，这里规定最多同时运行几个judge\_client  
*OJ\_SLEEP\_TIME=5* judged通过轮询数据库发现新任务，轮询间隔的休息时间，单位秒  
*OJ\_TOTAL=1* 老式并发处理中总的judged数量  
*OJ\_MOD=0* 老式并发处理中，本judged负责处理solution\_id按照TOTAL取模后余数为几的任务。 *OJ\_JAVA\_TIME\_BONUS=2* Java等虚拟机语言获得的额外运行时间。 *OJ\_JAVA\_MEMORY\_BONUS=512* Java等虚拟机语言获得的额外内存。 *OJ\_SIM\_ENABLE=0* 是否使用sim进行代码相似度的检测 *OJ\_HTTP\_JUDGE=0* 是否使用HTTP方式连接数据库，如果启用，则前面的HOST\_NAME等设置忽略。  
*OJ\_HTTP\_BASEURL=http://127.0.0.1/JudgeOnline* 使用HTTP方式连接数据库的基础地址，就是OJ的首页地址。  
*OJ\_HTTP\_USERNAME=admin* 使用HTTP方式所用的用户帐号（HTTP\_JUDGE权限），该帐号登录时不能启用VCODE图形验证码，但可以登录成功后启用。 *OJ\_HTTP\_PASSWORD=admin* 密码 *OJ\_OI\_MODE=0* 是否启用OI（信息学奥林匹克竞赛）模式，即无论是否出错都继续判剩余的数据，在ACM比赛中一旦出错就停止运行。  
*OJ\_SHM\_RUN=0* 是否使用/dev/shm的共享内存虚拟磁盘来运行答案，如果启用能提高判题速度，但需要较多内存。  
*OJ\_USE\_MAX\_TIME=1* 是否使用所有测试数据中最大的运行时间作为最后运行时间，如果不启用则以所有测试数据的总时间作为超时判断依据。 *OJ\_LANG\_SET=0,1,2,3,4* #判哪些语言的题目

## 二．Web解析

web分两大部分，前端和admin目录下的管理程序。

### 1．前端

无非是数据库的CRUD操作，关键功能是将用户提交的程序源码加入数据库的任务队列（solution表、souce\_code表）。

### 2．管理程序

提供具有administrator等高级权限的账号管理试题、账号等方面的功能。其中FPS导入导出程序主要为XML格式的数据处理。

\*\*特别的，judged可以多重启动，通过增加基准目录参数指定启动位置（默认/home/judge），从而确定judge.conf的位置，并确定其他参数。因此不但可以一个web服务器下挂多个判题服务器，也可以一台物理机器上同时启动任意多个相互独立的OJ系统。

实际使用中，使用开源的ispcp虚拟主机管理系统搭建多Web环境与hustoj协同工作取得了良好效果。

### 3．比赛方面

比赛根据数据通过率排名，而不只看AC数量

1．数据库solution表pass\_rate字段表示改条通过率。

2．把contestrank.php中的solved字段变成浮点对待。

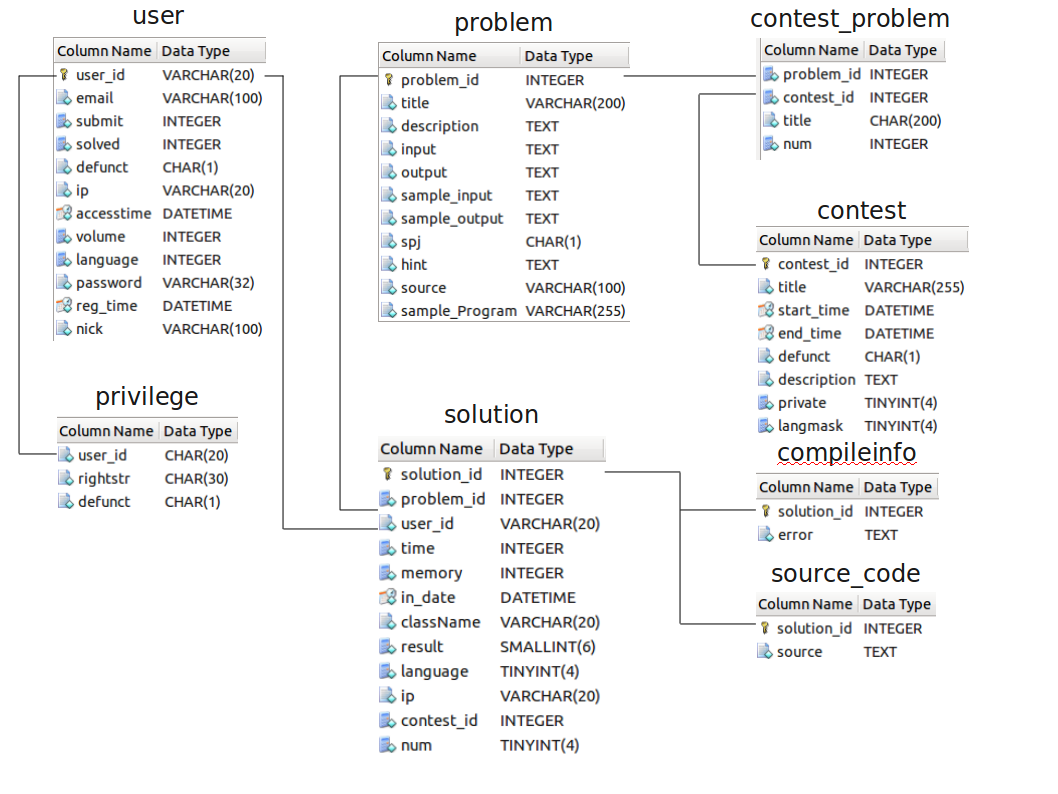
3．修改积分方式，按照希望的方式积分。可能需要给TM增加字段$p\_wa\_best\_rate记录每题最大通过率。

### 4．配置文件db\_info.inc.php注释

*static $DB\_HOST="localhost";* 数据库的服务器地址 *static $DB\_NAME="jol";* 数据库名 *static $DB\_USER="root";* 数据库用户名 *static $DB\_PASS="root";* 数据库密码 *// connect db* 数据库连接部分 *static $OJ\_NAME="HUSTOJ";* OJ的名字，将取代页面标题等位置HUSTOJ字样。 *static $OJ\_HOME="./";* OJ的首页地址 *static $OJ\_ADMIN="root@localhost";* 管理员email *static $OJ\_DATA="/home/judge/data";* 测试数据所在目录，实际位置。 *static $OJ\_BBS="discuss";*//"bbs"论坛的形式，discuss为自带的简单论坛，bbs为外挂论坛，参考bbs.php代码。 *static $OJ\_ONLINE=false;* 是否使用在线监控，需要消耗一定的内存和计算，因此如果并发大建议关闭  
*static $OJ\_LANG="en";* 默认的语言，中文为cn *static $OJ\_SIM=true;* 是否显示相似度检测的结果。 *static $OJ\_DICT=true;* 是否启用在线英字典 *static $OJ\_LANGMASK=1008;* //1mC 2mCPP 4mPascal 8mJava 16mRuby 32mBash 1008 for security reason to mask all other language 用掩码表示的OJ接受的提交语言，可以被比赛设定覆盖。1008为只使用C/CPP/Pascal/Java。  
*static $OJ\_EDITE\_AREA=true;* 是否启用高亮语法显示的提交界面，可以在线编程，无须IDE。  
*static $OJ\_AUTO\_SHARE=false;*//true: 自动分享代码，启用的话，做出一道题就可以在该题的Status中看其他人的答案。 *static $OJ\_CSS="hoj.css";* 默认的css,可以选择dark.css和gcode.css,具有有限的界面制定效果*。  
static $OJ\_SAE=false;* //是否是在新浪的云平台运行web部分  
*static $OJ\_VCODE=true;* 是否启用图形登录、注册验证码。 *static $OJ\_APPENDCODE=false;* 是否启用自动添加代码，启用的话，提交时会参考$OJ\_DATA对应目录里是否有append.c一类的文件，有的话会把其中代码附加到对应语言的答案之后，巧妙使用可以指定main函数而要求学生编写main部分调用的函数。  
*static $OJ\_MEMCACHE=false;*是否使用memcache作为页面缓存，如果不启用则用/cache目录  
*static $OJ\_MEMSERVER="127.0.0.1";* memcached的服务器地址  
*static $OJ\_MEMPORT=11211;* memcached的端口  
*static $OJ\_RANK\_LOCK\_PERCENT=0;* //比赛封榜时间的比率，如5小时比赛设为0.2则最后1小时封榜。  
*static $OJ\_SHOW\_DIFF=false;* //显示WrongAnswer时的对比

## 三．Core与Web的连接方式解析

### 1．简化ER图



### 2．数据库连接（默认）

1.Web插入Solution表

2. core轮询solution表，发现新纪录

3.core更新solution表result等字段

4.Web端轮询soltuion显示result等字段。

### 3．HTTP方式连接

1.Web插入Solution表

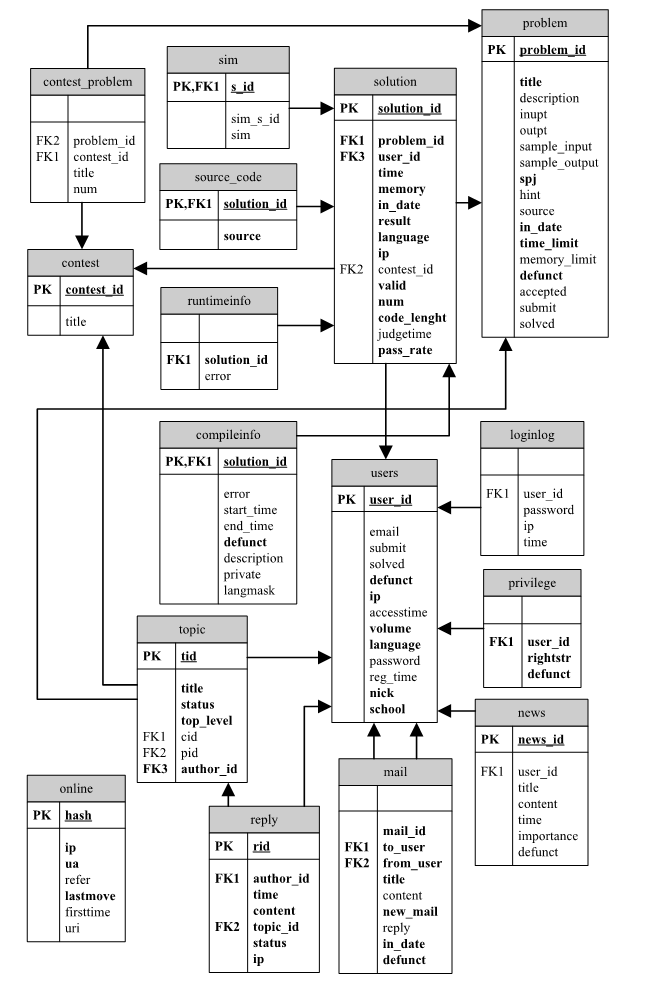
2. core访问Web端admin/problem\_judge.php，发现新纪录

3. core向Web端admin/problem\_judge.php提交数据，problem\_judge.php更新solution表result等字段

4. Web端轮询soltuion显示result等字段。

## 四．数据库解析

### 1．数据库关系图



### 2．数据库表分析（by夏夏）

|  |  |  |  |
| --- | --- | --- | --- |
| **HustOj数据库（~~r1292~~）** | | | |
| 序号 | 表名 | 作用 | 备注 |
| 1 | [compileinfo](#compileinfo) | 记录编译错误的记录 |  |
| 2 | [contest](#contest) | 竞赛表 |  |
| 3 | [contest\_problem](#contest_problem) | 竞赛题目 |  |
| 4 | [loginlog](#loginlog) | 登入日志 | 记录正确与错误的登入日志 |
| 5 | [mail](#mail) | 消息列表 |  |
| 6 | [news](#news) | 新闻表 |  |
| 7 | [online](#online) | 用户在线数据统计？ |  |
| 8 | [privilege](#privilege) | 权限授予 |  |
| 9 | [problem](#problem) | 题目表 |  |
| 10 | [reply](#reply) | 论坛（帖子及回复）表 |  |
| 11 | [runtimeinfo](#runtimeinfo) | 运行错误信息（Runtime Error） |  |
| 12 | [sim](#sim) | 相似度检测表 | 用于防作弊 |
| 13 | [solution](#solution) | 程序运行结果记录 |  |
| 14 | [source\_code](#source_code) | 提交的源码 |  |
| 15 | [topic](#topic) | 论坛帖子表 |  |
| 16 | [users](#users) | 用户信息 |  |
| 17 | [custominput](#custominput) | 用于在线IDE |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| compileinfo | | 记录编译错误的提交号（id）及原因 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| solution\_id | int | 11 | N | 主键（提交id，即RunID） |
| error | text |  | Y | 编译错误原因 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| contest | | 竞赛表 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| contest\_id | int | 11 | N | 竞赛id（主键） |
| title | varchar | 255 | Y | 竞赛标题 |
| start\_time | datetime |  | Y | 开始时间(年月日时分) |
| end\_time | datatime |  | Y | 结束时间(年月日时分) |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |
| description | text |  | Y | 描述（在此版本中未用） |
| private | tinyint | 4 |  | 公开/内部（0/1） |
| langmask | int | 11 |  | 语言 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| contest\_problem | | | | 竞赛题目 | | | | |
| 字段名 | | 类型 | | 长度 | | 是否允许为空 | | 备注 |
| problem\_id | | int | | 11 | | N | | 题目id |
| contest\_id | | int | | 11 | | Y | | 竞赛id |
| title | | char | | 200 | | N | | 标题 |
| num | | int | | 11 | | N | | 竞赛题目编号 |
| loginlog | | | | 登入日志(不管是否登入成功都记录) | | | | | | |
| 字段名 | | 类型 | | 长度 | | 是否允许为空 | | 备注 | | |
| user\_id | | varchar | | 20 | | N | | 用户id | | |
| password | | varchar | | 40 | | Y | | 密码（不一定正确） | | |
| ip | | varcahr | | 100 | | Y | | 登录的ip | | |
| time | | datetime | |  | | Y | | 登入时间 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mail | | 站内消息系统 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| mail\_id | int | 11 | N | 消息编号 |
| to\_user | varchar | 20 | N | 接收者 |
| from\_user | varchar | 20 | N | 发送者 |
| title | varchar | 200 | N | 标题 |
| content | text |  | Y | 内容 |
| new\_mail | tinyint | 1 | N | 新消息（1/0） |
| reply | tinyint | 4 | Y | 回复 |
| in\_date | datetime |  | Y | 时间 |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| news | | 新闻（首页显示） | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| news\_id | int | 11 | N | 新闻编号（主键） |
| user\_id | varchar | 20 | N | 用户账号 |
| title | varchar | 200 | N | 新闻标题 |
| content | text |  | N | 内容 |
| time | datetime |  | N | 更新时间 |
| importance | tinyint | 4 | N | 关键字 |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| online | |  | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| hash | varchar | 32 | N | 主键 |
| ip | varchar | 20 | N | Ip地址 |
| ua | varchar | 255 | N | 浏览器发出的浏览器相关的标识字符串 |
| refer | varchar | 255 | Y | 浏览器发出的一个表示访问的上个页面的网址。 |
| lastmove | int | 10 | N | 最后一次修改时间 |
| firsttime | int | 10 | Y | 第一次访问时间 |
| uri | varchar | 255 | Y | （统一资源指示器），包括URL（统一资源定位符）和URN（统一资源名称）两种 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| privilege | | 用户分组 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| user\_id | char | 20 | N | 用户帐号 |
| rightstr | char | 30 | N | 分组 |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| problem | | 题目表 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| problem\_id | int | 11 | N | 题目编号，主键 |
| title | varchar | 200 | N | 标题 |
| description | text |  | Y | 题目描述 |
| inupt | text |  | Y | 输入说明 |
| output | text |  | Y | 输出说明 |
| sample\_input | text |  | Y | 输入参照 |
| sample\_output | text |  | Y | 输出参照 |
| spj | char | 1 | N | 是否为特别题目 |
| hint | text |  | Y | 暗示 |
| source | varchar | 100 | Y | 来源 |
| in\_date | datetime |  | Y | 加入时间 |
| time\_limit | int | 11 | N | 限时（秒） |
| memory\_limit | int | 11 | N | 空间限制(MByte) |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |
| accepted | int | 11 | Y | 总ac次数 |
| submit | int | 11 | Y | 总提交次数 |
| solved | int | 11 | Y | 解答（未用） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| reply | | 论坛（帖子及回复）（Web Board） | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| rid | int | 11 | N | 帖子序号（主键） |
| author\_id | varchar | 20 | N | 作者帐号 |
| time | datetime |  | N | 发布时间 |
| content | text |  | N | 帖子内容 |
| topic\_id | int | 11 | N | 帖子分组 |
| status | int | 2 | N | 状态（0：正常，1：锁定，2：删除） |
| ip | varchar | 30 | N | 发帖子者ip |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| runtimeinfo | | 运行错误信息（Runtime Error） | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| solution\_id | int | 11 | N | 运行id（主键） |
| error | text |  | Y | 错误记录 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sim | | 相似度检测 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| s\_id | int | 11 | N | 提交号soltiotn\_id(主键) |
| sim\_s\_id | int | 11 | Y | 与s\_id相似的soltion\_id |
| sim | int | 11 | Y | 相似度（50-100） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| solution | | 程序运行结果记录 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| solution\_id | int | 11 | N | 运行id（主键） |
| problem\_id | int | 11 | N | 问题id |
| user\_id | char | 20 | N | 用户id |
| time | int | 11 | N | 用时（秒） |
| memory | int | 11 | N | 所用空间（） |
| in\_date | datetime |  | N | 加入时间 |
| result | smallint | 6 | N | 结果（4：AC） |
| language | tinyint | 4 | N | 语言 |
| ip | char | 15 | N | 用户ip |
| contest\_id | int | 11 | Y | 所属于竞赛组 |
| valid | tinyint | 4 | N | 是否有效？？？ |
| num | tinyint | 4 | N | 题目在竞赛中的顺序号 |
| code\_lenght | int | 11 | N | 代码长度 |
| judgetime | datetime |  | Y | 判题时间 |
| pass\_rate | decimal | 2 | N | 通过百分比（OI模式下可用） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| source\_code | | 记录源代码 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| solution\_id | int | 11 | N | 运行id（主键） |
| source | text |  | N | 源代码 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| topic | | 论坛（答辩）帖子主题 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| tid | int | 11 | N | 帖子编号（主键） |
| title | varbinary | 60 | N | 标题 |
| status | int | 2 | N | 状态（0：未锁定，1：锁定） |
| top\_level | int | 2 | N | 置顶等级（0，1：题目置顶，2：分区置顶，3：总置顶） |
| cid | int | 11 | Y | 竞赛编号 |
| pid | int | 11 | N | 竞赛中题目编号 |
| author\_id | varchar | 20 | N | 作者id |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| users | | 用户表 | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| user\_id | varchar | 20 | N | 用户id（主键） |
| email | varchar | 100 | Y | 用户E-mail |
| submit | int | 11 | Y | 用户提交次数 |
| solved | int | 11 | Y | 成功次数 |
| defunct | char | 1 | N | 是否屏蔽（Y/N） |
| ip | varchar | 20 | N | 用户注册ip |
| accesstime | datetime |  | Y | 用户注册时间 |
| volume | int | 11 | N | 页码（表示用户上次看到第几页） |
| language | int | 11 | N | 语言 |
| password | varchar | 32 | Y | 密码（加密） |
| reg\_time | datetime |  | Y | 用户注册时间 |
| nick | varchar | 100 | N | 昵称 |
| school | varchar | 100 | N | 用户所在学校 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| custominput | | 用于IDE | | |
| 字段名 | 类型 | 长度 | 是否允许为空 | 备注 |
| solution\_id | int | 11 | N | 用户id（主键） |
| Input\_text | text |  |  | 输入测试数据 |

## 五．LiveCD解析

### 1．LiveCD的实现

通过uck工具解压出ubuntulivecd的chroot环境，并在其中删除oo、gnome等大型程序释放空间，然后用apt工具安装基础环境，安装配置lxde和hustoj。再使用uck重新打包形成iso。

### 2．升级方式

利用googlecode的svn服务，用svn客户端分别升级core和web,再编译core，并通过web提供可能的数据库升级。

livecd中的升级脚本为update-hustoj，可以用which命令查找其实际位置。

# HUSTOJ建立分布式判题系统

HUSTOJ 支持一台数据库服务器，多台web服务器和多台判题服务器，以承担较高的访问负荷。

### 1．创建用于从远程连接数据库的帐号

*GRANT ALL PRIVILEGES ON jol.\* TO ‘judge’@’%’*

*IDENTIFIED BY ‘judge\_pass’ WITHOUT GRANT OPTION;*

*flush privileges*

其中jol为数据库，judge为帐号，judge\_pass为密码。  
注意：检查/etc/mysql/my.cnf 确保

*bind-address = 0.0.0.0*

高负载的情况下最好设置更多的连接数。*max\_connections = 512*

### 2．配置各web程序连接到数据库

修改include/db\_info.inc.php

*static  $DB\_HOST=”数据库服务器ip”;*

*static  $DB\_NAME=”jol”;*

*static  $DB\_USER=”judge”;*

*static  $DB\_PASS=”judge\_pass”;*

### 3．配置各判题程序连接到数据库，分配任务

*OJ\_HOST\_NAME=数据库服务器ip*

*OJ\_USER\_NAME=judge*

*OJ\_PASSWORD=judge\_pass*

*OJ\_DB\_NAME=jol*

*…*

*OJ\_TOTAL=判题机总数*

*OJ\_MOD=本机编号，从0开始*

*…*

\* 自R784版本开始，不再需要分别配置OJ\_TOTAL和OJ\_MOD,全部设为1和0即可，所有正常工作的judge节点将自动分配当前任务。

### 4．复制测试数据目录到各判题机

\*从r1520开始，使用HTTP\_JUDGE方式不必单独复制数据，数据将从web服务器按需下载。

先要准备好远程访问,在主服务器执行：

*sudo apt-get install openssh-server*

从主机向判题机复制：

*scp -r /home/judge/data  root@判题机ip:/home/judge/*

或用同步命令：

*rsync -vzrtopgh –delete /home/judge/data root@判题机ip:/home/judge/*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

判题机从主机复制：

*scp -r root@主机ip:/home/judge/data  /home/judge/*

或用同步命令：

*rsync -vzrtopg –progress –delete root@主机ip:/home/judge/data /home/judge/*

### 5．在各判题机重启判题程序

*sudo pkill judged&&sudo judged*

# HUSTOJ升级

## 一．通过SVN方式安装的

### 1．检出最新web，复制原upload目录到新目录，测试后切换

*sudo svn checkout https://github.com/zhblue/hustoj/trunk/trunk/web /var/www/new/*

*sudo cp -a /var/www/JudgeOnline/include/db\_info.inc.php  /var/www/new/include/*

*sudo cp -a /var/www/JudgeOnline/upload  /var/www/new/*

### 2．检出最新core，./make.sh

*sudo svn checkout https://github.com/zhblue/hustoj/trunk/trunk/core core*

*cd core*

*sudo pkill -9 judged*

*sudo ./make.sh*

*sudo judged*

### 3．管理员登陆后台，更新数据库

访问http://原OJ地址/new->登陆，后台，更新数据库。->测试无误后

*sudo mv /var/www/JudgeOnline /var/www/oldOJ*

*sudo mv /var/www/new /var/www/JudgeOnline*

## 二．通过.tar.gz源码安装的

### 1．到安装文件目录找到hustoj-read-only目录

### 2．执行命令

*sudo svn up hustoj-read-only*

*cd hustoj-read-only/core*

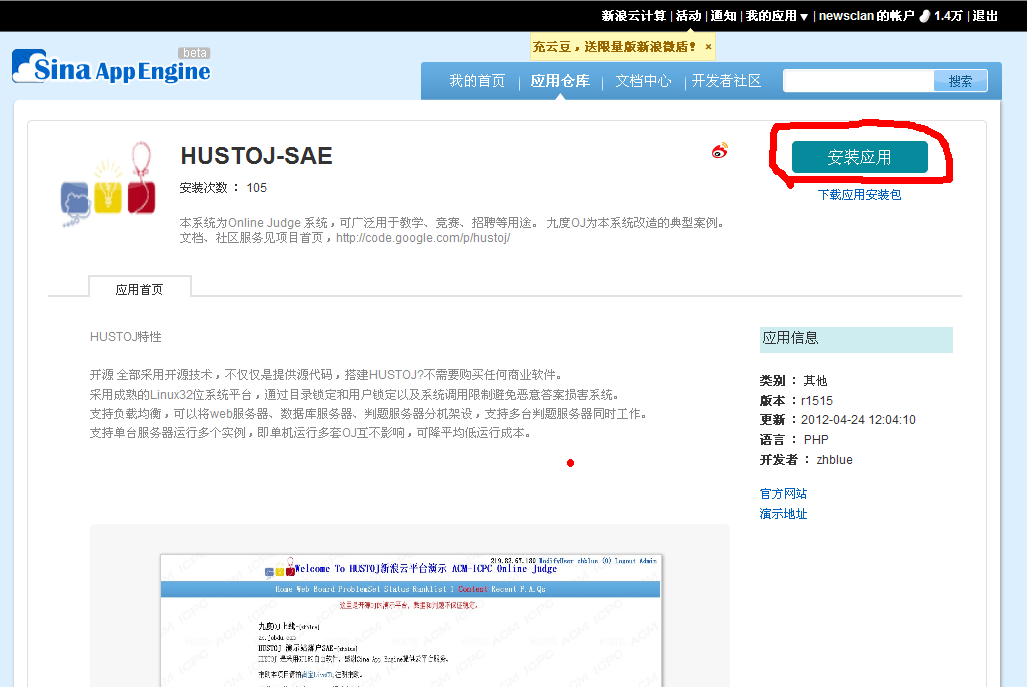
*sudo ./make.sh*

*sudo svn up /var/www/JudgeOnline*

# HUSTOJ新浪云安装步骤（by zhblue）

### 1．使用新浪微博账号登陆新浪云计算，打开应用，选择安装。

<http://sae.sina.com.cn/?m=apps&a=detail&aid=92>

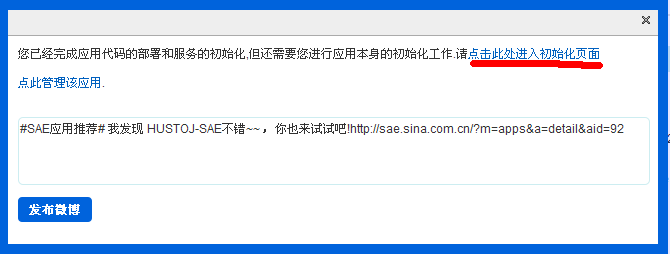


### 2．输入域名，不能与已有应用重名

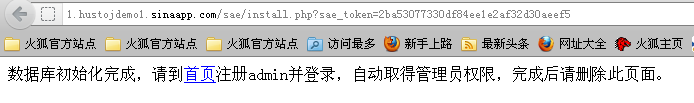


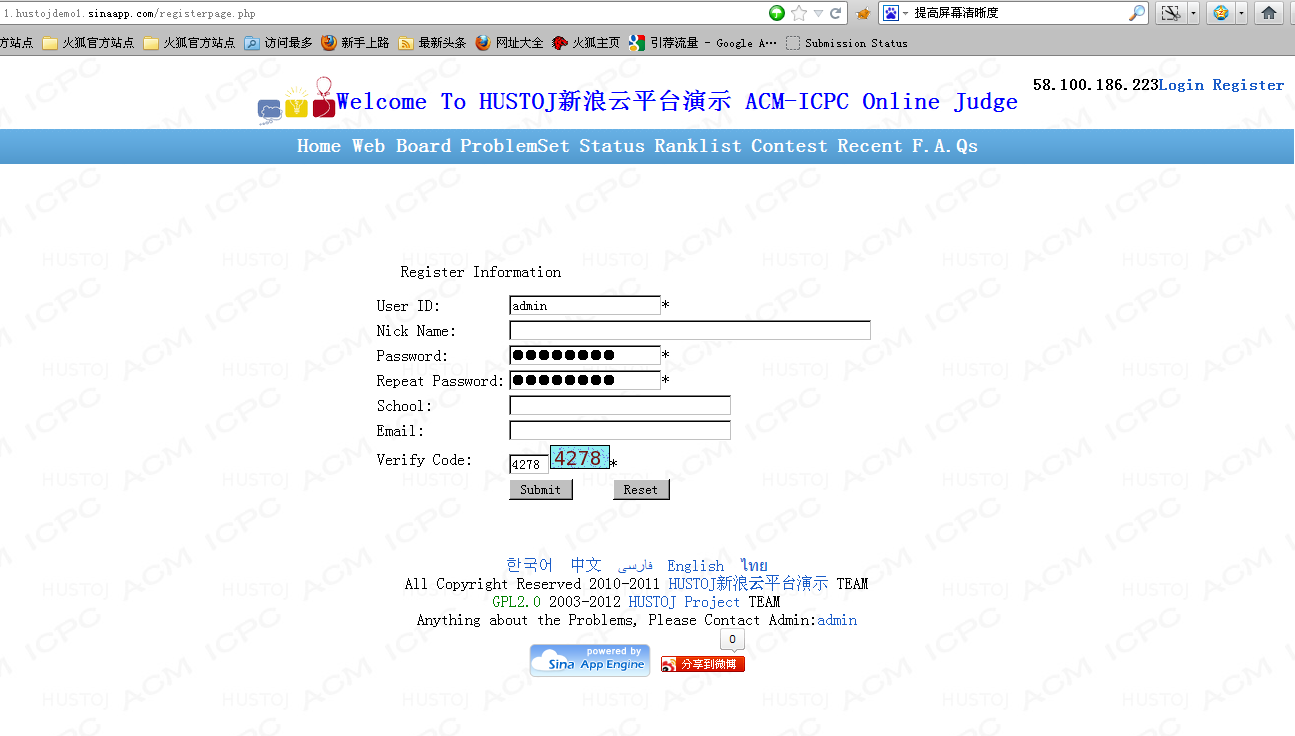


### 3．初始化应用（建库）

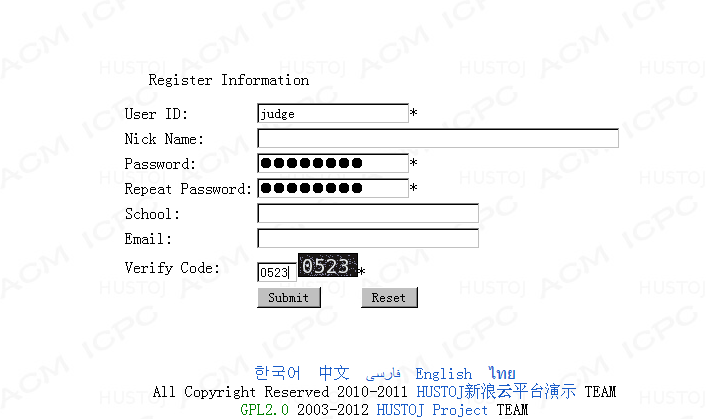


### 4．注册admin账号





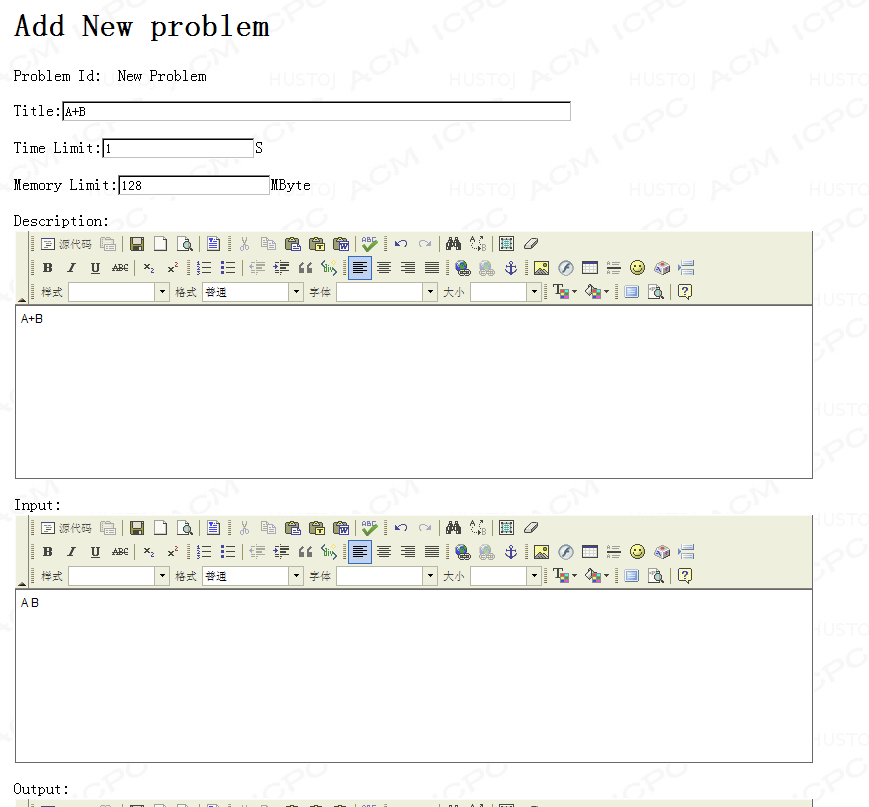
### 5．注销管理员，再注册一个判题账号judge



### 6．注销判题账号，重新登录管理员，给judge添加http\_judge权限。



### 7．增加题目A+B测试



### 8．管理应用



### 9．修改data为私有



### 10．管理测试数据

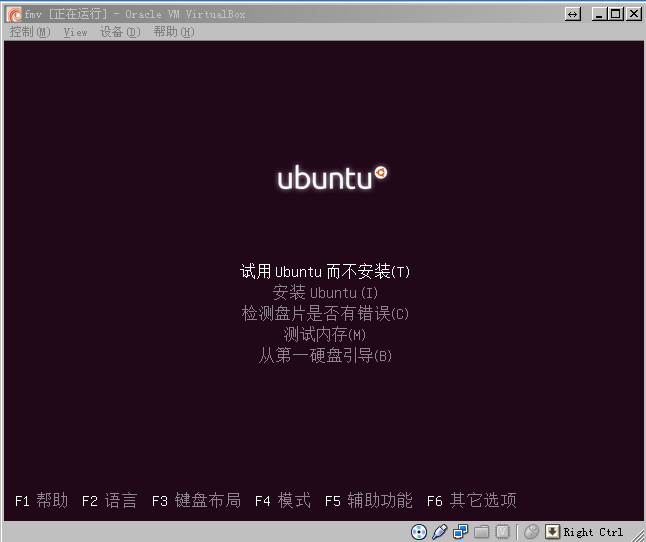


可以增加更多测试数据

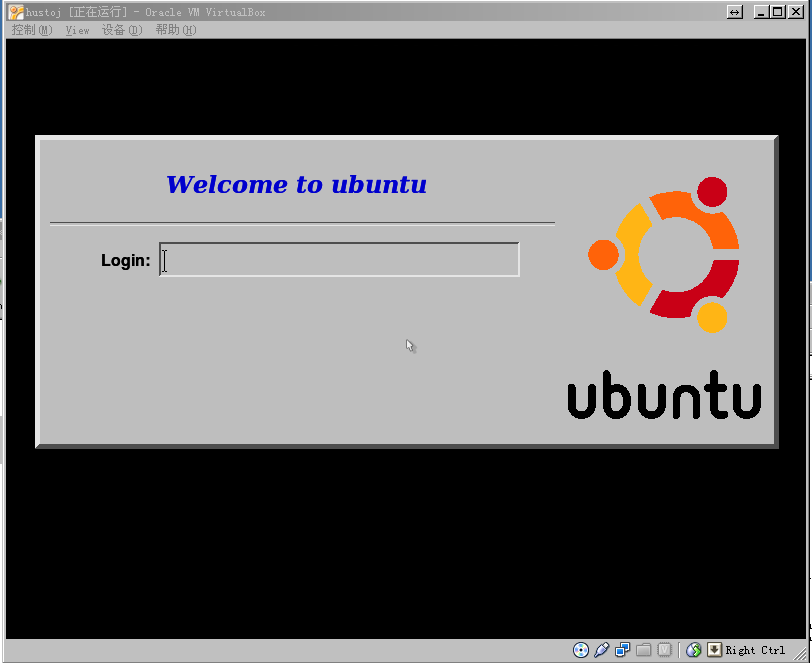


### 11．安装开源虚拟机Vbox

启动LiveCD，启动时按下任意键，选择试用。

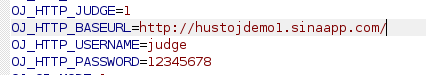


用户ubuntu密码freeproblemset.



### 12．编辑LiveCD中的配置文件

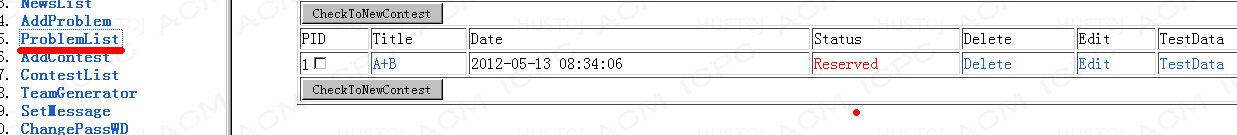




重启服务



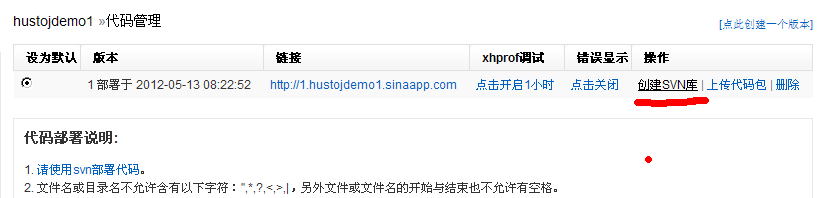
找到题目



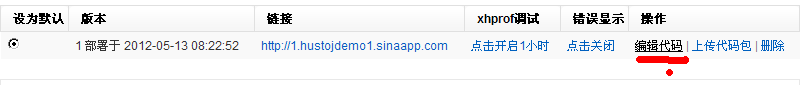
### 13．测试提交



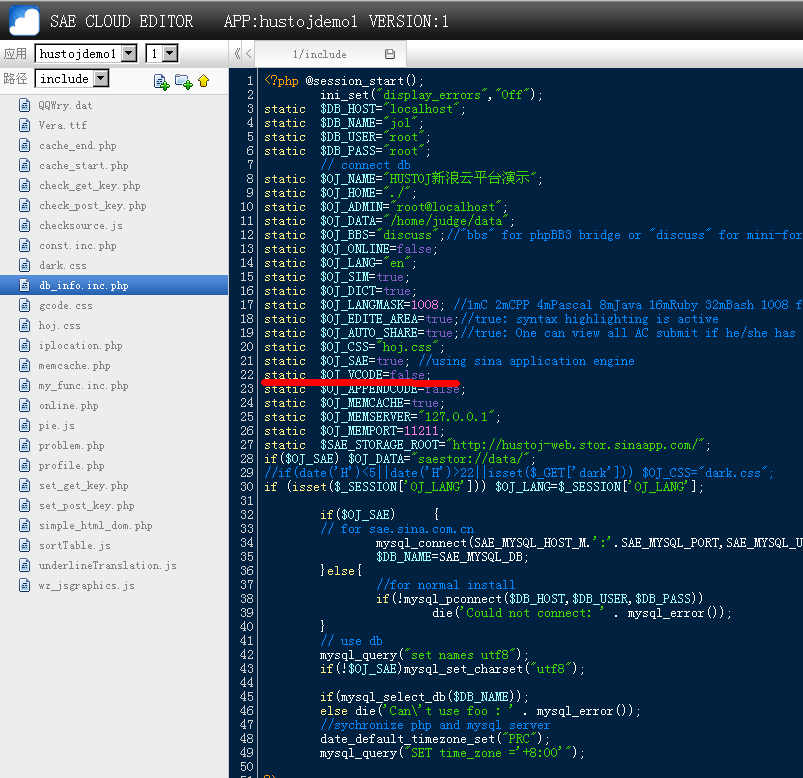
### 14．创建SVN库

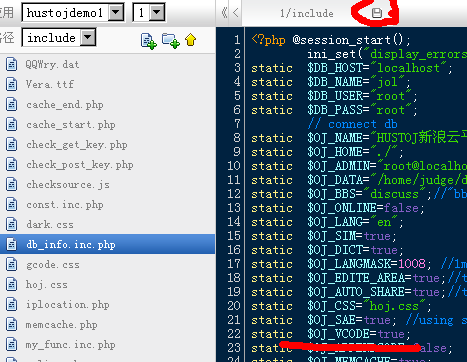


编辑代码



15．临时设vcode为false，测试，待题目AC再改回。





# HUSTOJ腾讯云安装步骤及错误排查

### 1．安装Ubuntu 14.04 LTS 64bit

### 2．开启root用户

*sudo passwd root*

*sudo vi /etc/ssh/sshd\_config*

*i→PermitRootLogin yes→esc→:wq*

*sudo service ssh restart*

### 3．安装hustoj

*sudo apt-get update*

*sudo apt-get install subversion*

*sudo svn co https://github.com/zhblue/hustoj/trunk/trunk/install hustoj*

*cd hustoj*

*sudo bash install-interactive.sh*

### 4．添加管理员账号

*mysql -u root -p*

*use jol*

*insert into privilege(user\_id,rightstr) values('root','administrator');*

root 为需要加管理员权限的帐号

### 安装phpMyAdmin

*sudo apt-get install phpmyadmin*

安装中的Web Server选择Apache2。

建立phpMyAdmin与Apache2的连接：

如www目录在/var/www，phpMyAdmin在/usr/share/phpmyadmin目录，则执行

*sudo ln –s /usr/share/phpmyadmin /var/www*

### 6．特别注意

分析脚本install-interactive.sh

*sudo apt-get update*

*sudo apt-get install make flex g++ clang libmysqlclient-dev libmysql++-dev php apache2 mysql-server php-mysql php-gd php-cli mono-gmcs subversion redis-server php-redis*

*sudo apt-get install make flex g++ clang libmysqlclient-dev libmysql++-dev php5 apache2 mysql-server php5-mysql php5-gd php5-cli mono-gmcs subversion redis-server php5-redis*

*sudo /etc/init.d/mysql start*

*HTTP\_START="sudo /etc/init.d/apache2 restart"*

可知该脚本采用LAMP方式，运行环境为Ubuntu、Apache2、PHP5以及MySQL。

# HUSTOJ FAQ

### 1．如何进入后台？

以管理员身份登录，点击Admin/管理进入后台。

### 2．如何添加题目？

进入后台，点击左侧NewProblem。

### 3．如何添加测试数据？

添加题目时，可以在test input/test output添加一组测试数据，大规模的数据（10kb+）和更多的数据，可以在添加完题目后，通过ftp/sftp,上传到题目对应目录，通常是/home/judge/data/题号。命名规则是输入数据以.in结尾，输出数据以.out结尾，主文件名相同。

### 4．如何编辑题目？

后台中点击ProblemList,找到需要编辑的题目，点击Edit。编辑时不能修改测试数据，测试数据请使用ftp工具修改。

### 5．如何启用题目？

题目添加后，默认是停用状态，以防比赛提前漏题，后台中点击ProblemList，找到题目，点击Resume启用题目，或者组织比赛，比赛中的题目将自动启用。

### 6．如何组织比赛？

在题目列表ProblemList中选择使用的题目，在PID一栏打钩，点击CheckToNewContest按钮，进入到比赛添加页面，输入比赛名称，设定比赛时间，语言类型，访问权限提交即可。

也可以使用管理菜单中的NewContest,需要手动输入题目编号，用英文逗号分隔。

### 7．如何修改、删除比赛？

点击比赛列表ContestList，选择Edit或Delete。

### 8．如何修改公告信息？

点击SetMessage。

### 9．如何修改用户密码？

点击ChangPassWD

### 10．如何重新判题？

点击Rejudge,输入题号或运行编号。

### 11．如何增加用户权限？

Addprivilege, administrator为管理员，source\_browser为代码审查，contest\_creator为比赛组织者。

通常给使用系统的老师分配代码审查和比赛组织者权限即可。

### 12．如何导入、导出题目？

使用ImportProblem，上传FPS文件。

使用ExportProblem，输入起始编号，结束编号，或题号列表，如果输入了列表，起始结束将不起作用。

### 13．如何更新数据库结构？

系统升级中，有对数据库的修改，这些修改不能通过SVN实现自动更新，如果发现升级web/core代码后系统报错，可以执行update database操作，进行数据库升级。因为脚本中有测试代码，所以重复执行不会造成影响。

### 14．如何下载新题目？

访问FreeProblemSet,查看Downloads列表。

*https://github.com/zhblue/freeproblemset/*

### 15. 如何让判题程序忽略行尾的空白字符？

在judge\_client.cc头部增加宏定义 IGNORE\_ESOL

### 16．如何上传多组数据？

加好题目后在题目列表找TestData，点击上传。 主文件名一样的\*.in \*.out，如test1.in test1.out

# HUSTOJ出错问题分析及解决

### 1．无法正常判题，一直pending

可能是judged服务未能正常启动，执行如下命令即可。

*sudo judged*

或者检查/home/judge/etc/judge.conf中的数据库账号配置，修正后再次重启服务器或执行*sudo pkill -9 judged*

等待一会儿再执行

*sudo judged*

### 2．Runtime error

若出现Runtime Error:[ERROR] A Not allowed system call: runid:10735 CALLID:20时，编辑okcalls64.h或okcalls32.h（取决于您使用的Linux版本uname -a出现x64字样则64位，i686字样则32位），在对应的语言数组里增加内容。 如C或C++：

*int LANG\_CV[256] = { 85, 8,140, SYS\_time, SYS\_read, SYS\_uname, SYS\_write, SYS\_open, SYS\_close, SYS\_execve, SYS\_access, SYS\_brk, SYS\_munmap, SYS\_mprotect, SYS\_mmap2, SYS\_fstat64, SYS\_set\_thread\_area, 252, 0 };*

将上述报错中CALLID:后的数字，增加到数组中非末尾的位置，如果这个数字是0，请加在首位。

*int LANG\_CV[256] = { 20, 85, 8,140, SYS\_time, SYS\_read, SYS\_uname, SYS\_write, SYS\_open,SYS\_close, SYS\_execve, SYS\_access, SYS\_brk, SYS\_munmap, SYS\_mprotect,SYS\_mmap2, SYS\_fstat64, SYS\_set\_thread\_area, 252, 0 };*

修改完成，重新在core目录执行sudo ./make.sh 然后重新测试，如果发现再次出现类似错误，请留意CALLID数字变化，重复上述步骤直至问题消失。

### 3．Ubuntu下Apache2报错

Ubuntu 环境，当apache2重启提示时：

*\* Starting web server apache2*

*apache2: Could not reliably determine the server's fully qualified domain name,*

*... waiting apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName*

解决的方法是：

*sudo vim /etc/apache2/sites-available/default*

打开default文件后，在default文件顶端加入：

*ServerName 127.0.0.1*

重启apache2 就不会提示上述错误了。

*sudo /etc/init.d/apache2 restart*

### 4．页面总是需要刷新才能显示

如果您使用的是ie6浏览器，请禁用服务器上的deflate模块。

*sudo rm /etc/apache2/mods-enabled/deflate.\* sudo /etc/init.d/apache2 restart*

### 5．添加题目时出现warning/题目目录下数据没有自动生成

需要修改系统php.ini,给予php操作数据目录的权限。以下是推荐修改的设置。

*sudo gedit /etc/php5/apache2/php.ini open\_basedir =/home/judge/data:/var/www/JudgeOnline:/tmp*

*max\_execution\_time = 300 ; Maximum execution time of each script, in seconds max\_input\_time = 600 memory\_limit = 128M ; Maximum amount of memory a script may consume (16MB) post\_max\_size = 64M upload\_tmp\_dir =/tmp upload\_max\_filesize = 64M*

重启Apache即可。

### 6．添加的题目普通用户看不到

题目默认为删除状态，只有管理员能访问，当管理员确认题目没有问题后，可以点击ProblemList中的Resume启用题目。

### 7．Java总是CE/RE

目前只支持sun原版jdk和openjdk，其他jdk暂不能保证支持。

### 8．管理员不能查看别人的源码

请给自己增加source\_browser权限。

### 9．fckeditor上传的图片在题目中无法显示

 如果web安装位置不在/JudgeOnline，需要手工修改/fckeditor/editor/filemanager/connectors/php/config.php37行 $Config['UserFilesPath'] = '/JudgeOnline/upload/'.date("Ym")."/" ;

将JudgeOnline修改为对应的OJ web路径,如oj。

# HUSTOJ 功能自定义

## 一．开启sim模块的防作弊功能

### 1．在judge.conf里添加如下行

*OJ\_SIM\_ENABLE=1*

### 2．在db\_info.inc.php里添加如下行

*$OJ\_SIM=true;*

### 3．在MYSQL中创建如下table

*CREATE TABLE sim (*

*s\_id int(11) NOT NULL,*

*sim\_s\_id int(11) NULL,*

*sim int(11) NULL,*

*PRIMARY KEY (s\_id)*

*) ENGINE=MyISAM DEFAULT CHARSET=utf8;*

### 4．特别说明

若长时间运行本功能则会占用大量存储空间！

## 二．制定自己的前台模板、论坛

### A．制定自己的前台模板

针对很多学校希望自定义OJ界面风格的需求，对原代码进行了初步的模板分离，在template目录下存放模板，原有风格定义为classic。在include/db\_info.inc.php文件中新建$OJ\_TEMPLATE用于指定模板名。

如果需要制作新的模板，可以选择复制classic目录。然后修改其中的php、css、imgae等文件，在db\_info.inc.php中修改$OJ\_TEMPLATE变量为新模板名即可。

有条件的也可以请美工重新设计整个页面，然后切图形成html,再针对每个页面进行代码嵌入，从classic的php中提取显示用的代码段，放入新模板的html中，并命名为正确的\*.php文件，放入新模板目录即可。

##### 1．复制template/bs目录，放置在template目录中，并改为新模板名。

##### 2．在db\_info.inc.php中修改$OJ\_TEMPLATE变量为新模板名

##### 3．浏览前台，修改新目录中对应的php、css、imgae等文件

##### 4．实现admin/problem\_judge.php的仿真，方便直接集成原版core。

### B．论坛

##### 1．建议集成GPL的phpbb，参考

*https://github.com/zhblue/hustoj/blob/master/trunk/web/bbs.php*

##### 2．集成Discuz建议购买商业许可。

## 三．给HUSTOJ添加多语言支持

HUSTOJ本身有对汉语、英语、韩语的支持。如果想要对HUSTOJ进行本地化，请按如下步骤操作。（以日语为例）

### 1．把/lang/en.php拷贝为/lang/jp.php

### 2．把/faqs.php拷贝为/faqs.jp.php

### 3．对jp.php和faqs.jp.php进行本地化

### 4．在/include/db\_info.inc.php中把*static $OJ\_LANG="en"*设置为*"jp"*

## 四．给HUSTOJ添加更多编程语言支持

HUSTOJ默认安装C和C++的编程环境，若要添加如下语言的支持，请安装对应的环境。

* Pascal：*sudo apt-get install fpc*
* Java：*sudo apt-get install openjdk-6-jdk*（需配置环境变量）
* Ruby：*sudo apt-get install ruby*
* Objective C：*sudo apt-get install gnustep gnustep-devel gobjc*
* Python：*sudo apt-get install python*

## 五．开启HUSTOJ的OI（信息学奥林匹克竞赛）模式

### 1．升级core、web和judge.conf的版本，保证不低于r1210 +

### 2．修改judge.conf，使得OJ\_OI\_MODE=1

### 3．在core文件夹输入*pkill judged && ./make.sh*来重新编译内核

### 4．特别注意

当开启OI模式后，系统会发生的变化：

即使有测试实例出错，它依旧会测试下去

比赛根据数据通过率排名，而不只看AC数量。

竞赛排名和练习排名不会在OI模式下发生改变，通过率只会在status.php中显示。

每题设定的最大运行时间的优先级在系统默认每题最大运行时间之上。

## 六．使用HTTP方式与core连接

### 1．执行apt-get install wget

### 2．注册用户并赋予http\_judge权限

### 3．编辑/home/judge/etc/judge.conf

*OJ\_HTTP\_JUDGE=1* 开启本功能

*OJ\_HTTP\_BASEURL=<http://127.0.0.1/JudgeOnline>* 填写OJ-web的地址

*OJ\_HTTP\_USERNAME=admin* 上述创立的具有http\_judge权限的用户名

*OJ\_HTTP\_PASSWORD=admin* 密码

# HUSTOJ Special Judge的说明及使用

## 说明

通常的ACM题目包括以下几项内容：题目描述(Description)、输入描述(Input)、输出描述(Output)、样例输入(Sample Input)、样例输出(Sample Out)，在后台则包括测试输入(Input Data)和测试输出(Output Data)两项。在评测用户提交的程序正确与否时，系统会将样例输入和测试输入重定向作为程序的标准输入，通过判断程序对应的输出是否与期待的输出完全相同，来判断解答是否正确。

对于同一道题目，用户可能使用各种不同的方法来解答，所以对于某些特殊的题目，其结果可能不唯一，但都符合题目要求。此类题目就需要进行特判(Special Judge)。HUSTOJ便提供了特判功能。

这些题目主要有：

1、答案不唯一。见下例一。

2、控制精度。题目要求输出精度在某eps之内。见下例二。

3、其他。

spj的文件可以是c或cpp编写的程序，编译为spj文件，并设执行权限。

spj.c 或 spj.cc，需编译为spj，执行spj时传3个文件名参数：输入，参考输出，用户输出。

spj.c 或 spj.cc 应放置于对应题目测试数据目录下，导出题目时将自动导出 。

执行判定是自动的。

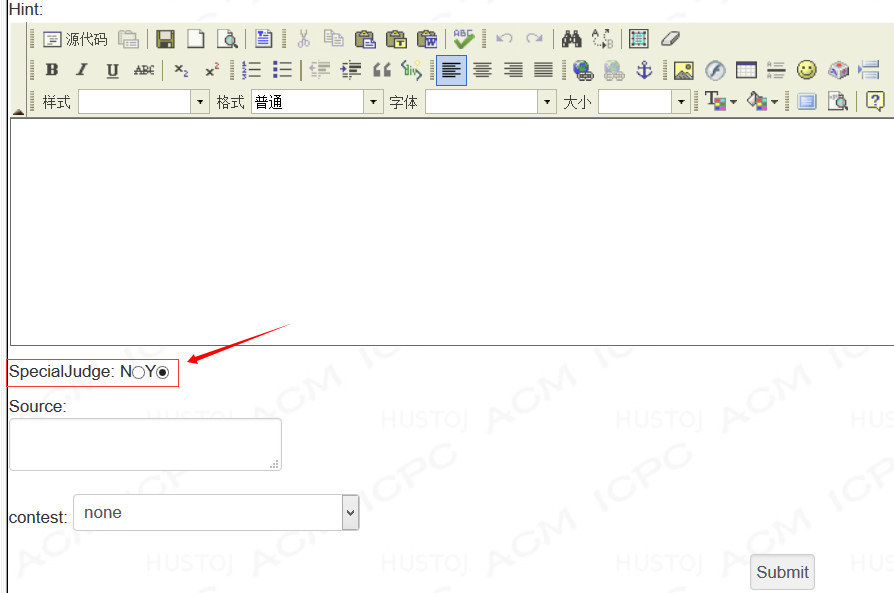
spj的退出值决定判断结果，成功退出(0)表示AC,其余表示WA.

把spj本身看做一道题目，输入是前述三个文件，输出是程序退出时的返回值,返回值是0表示AC,非零表示WA。

可以到freepeoblemset找到标有Spj的题目下载。

## 二、Special Judge在HUSTOJ中的使用

### 1．在题目后台编辑页面启用该选项



### 2、编制data.in、data.out、spj.cc（或spj.c）。

### 3、编译出spj，设执行权限，连同2中的三个文件一起上传至服务器。

## 三．例子

### 1．A+B spj.c

*#include<stdio.h>*

*int main(int argc,char \*args[])*

*{*

*FILE \* f\_in=fopen(args[1],"r");*

*FILE \* f\_out=fopen(args[2],"r");*

*FILE \* f\_user=fopen(args[3],"r");*

*int ret=0;*

*int a,b,c;*

*while(fscanf(f\_in,"%d %d",&a, &b) != EOF)*

*{*

*fscanf(f\_user,"%d",&c);*

*if(a+b!=c){ret=1; break;}*

*}*

*fclose(f\_in);*

*fclose(f\_out);*

*fclose(f\_user);*

*return ret;*

*}*

### 2. 给出一个不小于12的正整数n，请你输出两个合数，使他们的和等于n。

分析题意，可以得出系统判断用户输出是否正确的两个条件是：

输出的两个数是否均为合数；

输出的两个数之和是否为n。

首先分别制定data.in（测试输入）、data.out（测试输出）如下：

filename: data.in

|  |  |
| --- | --- |
| 1  2  3  4 | 3  12  15  1000 |

filename: data.out

|  |  |
| --- | --- |
| 1  2  3 | 8 4  9 6  500 500 |

然后编写spj.cc（特判程序）。HUSTOJ中的spj.cc需要有3个文件名参数传入：测试输入、测试输出、用户输出。  
spj的返回值决定着判断结果，成功返回(0)表示AC，其他非零值表示WA。  
下面给出一个本题spj.cc的示例代码。

*#include <stdio.h>*

*bool is\_prime(int x)//判断素数，伪代码*

*{*

*if(x是素数)*

*return true;*

*else*

*return false;*

*}*

*int main(int argc,char \*args[])//主函数*

*{*

*FILE \* f\_in=fopen(args[1],“r”);//测试输入*

*FILE \* f\_out=fopen(args[2],“r”);//测试输出*

*FILE \* f\_user=fopen(args[3],“r”);//用户输出*

*int ret=0;//返回值*

*int T,n,a,b;*

*fscanf(f\_in,“%d”,&T);//从输入中读取数据组数T*

*while(T–)*

*{*

*fscanf(f\_in,“%d”,&n);*

*fscanf(f\_user,“%d%d”,&a,&b);*

*if(a+b!=n || is\_prime(a) || is\_prime(b))*

*ret = 1;//Wrong Answer*

*}*

*fclose(f\_in);*

*fclose(f\_out);*

*fclose(f\_user);*

*return ret;*

*}*

编制完spj.cc后，将这些文件上传至服务器。

登录服务器以后进入/home/judge/data/[题号]文件夹，执行以下操作：

如/home/judge/data/1000

*sudo vim spj.cc #Edit the file spj.cc*

*sudo g++ -o spj spj.cc # gcc -o spj spj.c*

*sudo chmod +x spj*

测试：

*acm@acm:/home/judge/data/1000$ ./spj data.in data.out data.out*

*acm@acm:/home/judge/data/1000$ echo $?*

*0 #Accepted*

*acm@acm:/home/judge/data/1000$ ./spj data.in data.out data.in*

*acm@acm:/home/judge/data/1000$ echo $?*

*1 #Wrong Answer*

在本地HUSTOJ测试，通过网页提交，通过。

### 3．经过一定复杂计算后输出一个实数结果，精度要求控制在10−4内。

分析题意，可知spj需要判断测试输出a与用户输出x之差是否在eps之内。

代码如下

*#include <stdio.h>*

*const double eps = 1e-4;*

*int main(int argc,char \*args[])//主函数*

*{*

*FILE \* f\_in=fopen(args[1],“r”);//测试输入*

*FILE \* f\_out=fopen(args[2],“r”);//测试输出*

*FILE \* f\_user=fopen(args[3],“r”);//用户输出*

*int ret=0;//返回值*

*int T;*

*double a,x;*

*fscanf(f\_in,“%d”,&T);//从输入中读取数据组数T*

*while(T–)*

*{*

*fscanf(f\_out,“%lf”,&a);*

*fscanf(f\_user,“%lf”,&x);*

*if(fabs(a-x)>eps)*

*ret = 1;//Wrong Answer*

*}*

*fclose(f\_in);*

*fclose(f\_out);*

*fclose(f\_user);*

*return ret;*

*}*

# 在Ubuntu 14.04下搭建Vjudge（by BIGBALLON）

### 编者前言

本人是一名ACM弱菜，由于集训队需要，我们决定架设自己的OnlineJudge和Virtual Judge，于是开始了噩梦般的搭建。

OJ的搭建，非常顺利，参考的是 UbuntuServer12.04搭建hustoj 这篇博客，写得非常好，其他不多说，如果有需要搭建自己学校的OJ的朋友，可以参考，主要就是架设LAMP，其他也没什么了。

Vjudge的搭建，我却遇到了相当多的问题，由于JSP方面的知识不是很够，我只能不断地在网络上寻找资料和解决办法，无奈，泛滥的大量博文中，信息相当杂，精品相当少，大多数都是转载来转载去，内容相同还没有营养，搭建Vjudge方面的博文更是稀少，因此，搭建非常不顺利，连续两天我都没有能够搭建起来，最后在jcf的帮助下，终于解决了各种问题，成功搭建起了Virtual Judge。

在网上搜罗了一番，找到了一篇写得很好的文章 VirtualJudge 环境搭建与配置 ，没有这篇文章，我也搭建不起Virtual Judge，大家可以去参考。

另外，这里还找到了原作者的简易教程 How to deploy your own Vjudge，对于熟悉linux的同学来说，足矣搭建起Virtual Judge。但是对于不熟悉Linux和JSP的同学，这篇文章也许太过简单，然而，我再也没有找到更好的参考博文或是文档，于是，决定自己花一点时间来完成这个空缺，这篇文章由此诞生。

本人不才，修为尚浅，如果你有任何问题、建议或是意见，非常欢迎一起讨论。

联系方式：

Email：[605718199@qq.com](mailto:605718199@qq.com)

Blog：<http://www.cnblogs.com/BigBallon/>

BIGBALLON

## 1. 前期准备工作

### 1.1 一个Linux系统

这里我测试用的是ubuntu14.04 Desktop 64bit ,当然选择Server会更好一些.  
系统的安装不再赘述，作为服务器请选用Server版本。

### 1.2 更新源

在搭建环境之前，请确保你的源是有效的，速度是给力的，

建议选择一个国内的源 14.04 LTS 更新源。

将原来的源覆盖并保存

*sudo gedit /etc/apt/sources.list*

更新源

*sudo apt-get update*

**1.3 必要文件准备**

我们需要下载这几个文件（部分链接需要翻墙才能访问）：

1．\*.sql

2．http\_client.json

3．remote\_accounts.json

4．virtual judge源文件 <https://github.com/trcnkq/virtual-judge>

备用链接：<http://pan.baidu.com/s/1o6JVKtc>

**2. 必要环境搭建**

**2.1 JDK的安装和配置**

首先需要下载JDK，地址：[https://jdk7.java.net/download.html](https://jdk7.java.net/download.html" \t "_blank) (之前的云盘链接内有64位的,请认准这个jdk版本)  
注意系统是32位还是64位

1.) 解压下载的文件

*tar -xzvf jdk1.7.0\_80*

2.) 移动文件夹到指定目录下

*sudo mkdir /usr/lib/jvm*

*sudo mv jdk1.7.0\_80/ /usr/lib/jvm/*

3.) 设置环境变量

*sudo vi /etc/profile*

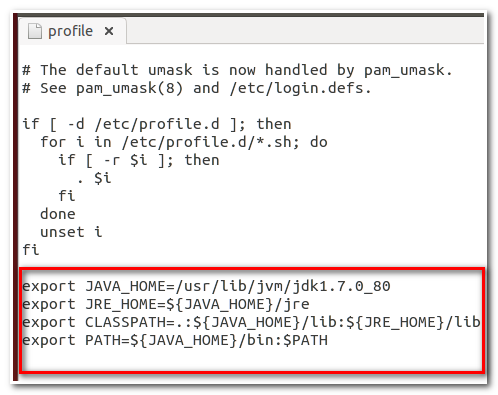
在文件最后加入如下内容：

*export JAVA\_HOME=/usr/lib/jvm/jdk1.7.0\_80*

*export JRE\_HOME=${JAVA\_HOME}/jre*

*export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib*

*export PATH=${JAVA\_HOME}/bin:$PATH*



4.) 使修改生效：

*sudo source /etc/profile*

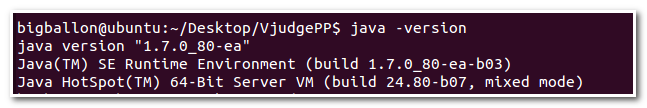
*//如果提示source command not found*

*//请执行以下两步:*

*//sudo -s*

*//source /etc/profile*

这时候在终端输入 java –version 查看当前 JDK 版本



至此，JDK 配置完成。

**2.2 Redis的安装和配置**

 1.)    下载安装Redis:

*wget http://download.redis.io/releases/redis-2.8.9.tar.gz*

*tar xvzf redis-2.8.9.tar.gz*

*cd redis-2.8.9/*

*make*

*sudo make install*

2.)    配置init脚本：

*wget https://github.com/ijonas/dotfiles/raw/master/etc/init.d/redis-server*

*wget https://github.com/ijonas/dotfiles/raw/master/etc/redis.conf*

*sudo mv redis-server /etc/init.d/redis-server*

*sudo chmod +x /etc/init.d/redis-server*

*sudo mv redis.conf /etc/redis.conf*

3.)    初始化用户和日志路径

第一次启动Redis前，建议为Redis单独建立一个用户，并新建data和日志文件夹

*sudo useradd redis*

*sudo mkdir -p /var/lib/redis*

*sudo mkdir -p /var/log/redis*

*sudo chown redis.redis /var/lib/redis*

*sudo chown redis.redis /var/log/redis*

4.) 设置开机自动启动，关机自动关闭

*update-rc.d redis-server defaults*

5.) 启动Redis：

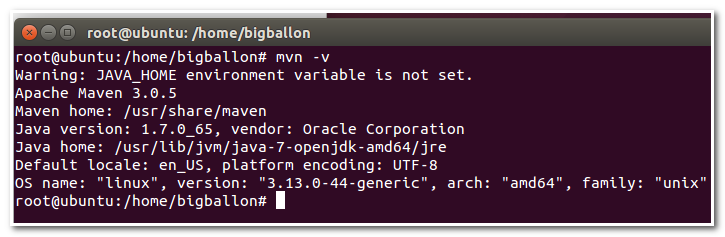
*/etc/init.d/redis-server start*

**2.3 Maven3的安装和配置**

1.)    通过apt-get安装Maven3

*sudo apt-get install maven –y*

安装完后，*sudo su* 进入root  
用 *mvn –v* 查看一下Maven的版本，如下图：

  
安装完毕后，默认的JDK版本可能会被更改。  
因为Maven3安装的同时把openjdk也一并安装了。

2.)  修改系统默认的jdk

*update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.7.0\_80/bin/java 300*

*update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.7.0\_80/bin/javac 300*

*update-alternatives --config java     //请选择jdk1.7.0\_80版本的jdk*

*update-alternatives --config javac*

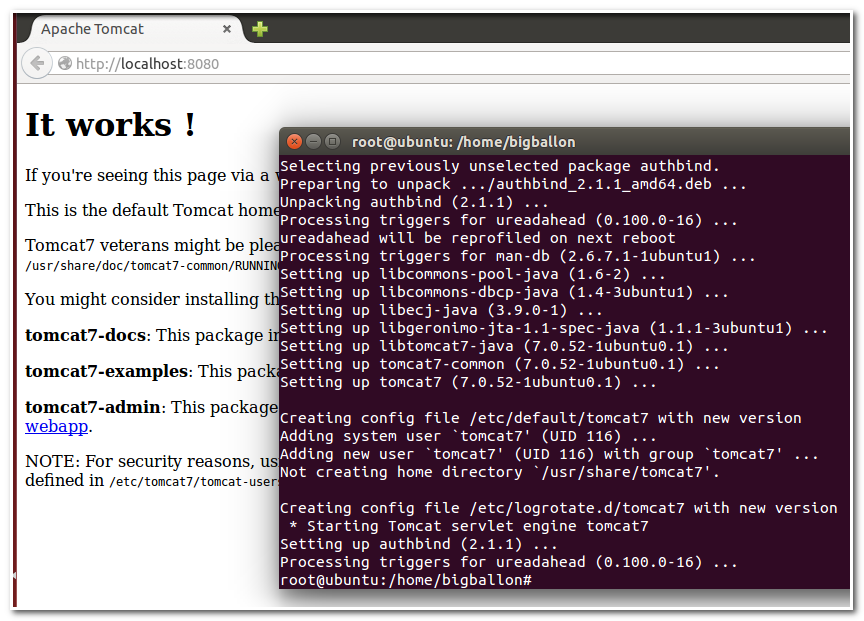
使用*java -version*或者*mvn -v*再次查看，JDK版本已更改。

**2.4 Tomcat7的安装和配置**

1.) apt-get安装tomcat7：

*sudo apt-get install tomcat7 -y*

tomcat7默认会被安装在目录/var/lib/tomcat7/下，安装完之后在浏览器下输入*localhost:8080*查看是否安装成功。如果出现下图，即为安装成功。



2.) 安装tomcat7-admin

安装成后，我们还需要安装一个tomcat7-admin

*s*udo apt-get install tomcat7-admin

3.) 修改账户

安装完毕之后，我们进入tomcat7的conf目录下修改tomcat-users.xml文件

*cd /var/lib/tomcat7/conf/*

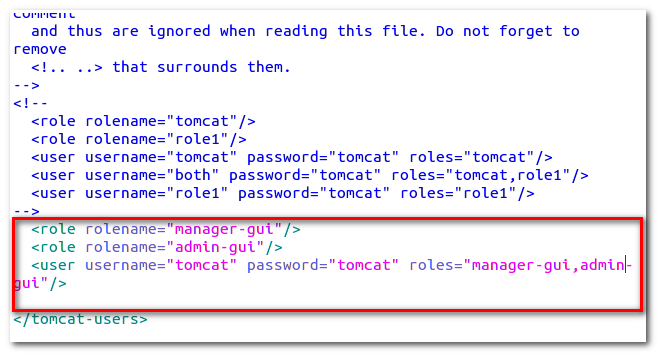
*gedit tomcat-users.xml*

我这里用的是

*<role rolename="manager-gui"/>*

*<role rolename="admin-gui"/>*

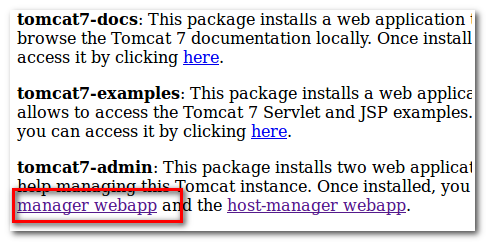
*<user username="tomcat" password="tomcat" roles="manager-gui,admin-gui"/>*



4.)    重启tomcat:

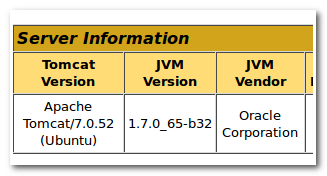
*sudo /etc/init.d/tomcat7 restart*

重新在浏览器打开tomcat界面  
点击manager webapp，用刚才我们建立的用户登陆。



5.) 修改JDK默认的JDK版本

1. 检查 tomcat7 的 Server Information，版本可能不是我们自己的jdk版本。
2. 这里我们要修改tomcat使用的JDK版本，这步很重要，否则会出现很多问题。



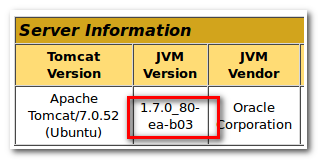
*sudo gedit /etc/default/tomcat7*

加入如下内容：

*JAVA\_HOME=/usr/lib/jvm/jdk1.7.0\_80*

再次重启tomcat7：

*sudo /etc/init.d/tomcat7 restart*



OK， tomcat7的JDK版本修改完毕。

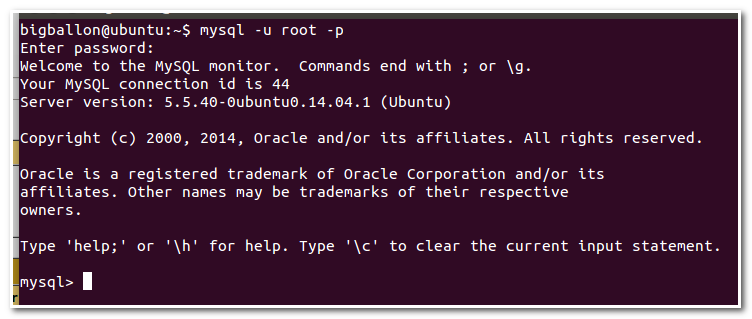
**2.5 Mysql的安装**

*sudo apt-get install mysql-server –y*

安装过程中会要求你输入数据库的密码，这里我直接用的123456，连续输入两次即可。

进入数据库测试一下

*mysql -u root –p*



**3. 工程代码实施**

准备好四个文件，就是一开始下载的那四个。

**3.1 Vjudge的打包**

进入virtual-judge-master 目录

*cd virtual-judge-master/*

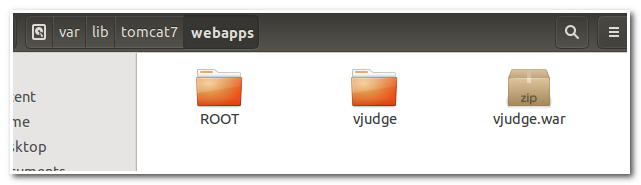
用 Maven 将 Virtual Judge 打包

*mvn clean package*

里面会有一个vjudge.war文件，就是我们打包完成的 Virtual Judge，把这个war文件拷到tomcat7的webapps目录下。

打包的过程可能会相当长，特别是网络不好的情况下，万一掉包了，非常蛋疼，如果你不想等待太久，可以直接下载此文件，在文章开头的百度云网盘链接里。

放到webapps目录后，会自动生成一个vjudge文件夹，如果没有生成，你也可以自行解压。

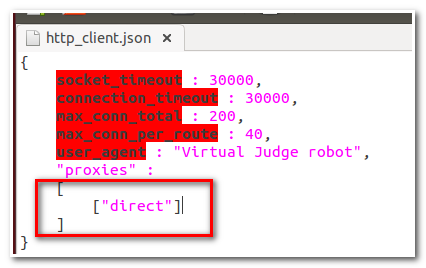


**3.2 remote\_accounts.json文件的编辑**

把各个OJ的提交账号添加到remote\_accounts.json里。

**3.3 config.properties文件的编辑**

如不需要代理或VPN访问国外OJ，保留http\_client.json里面的*[“direct”]*即可。



**3.4 vjudge的简单部署**

把 remote\_accounts.json 和 http\_client.json 两个文件放在指定的文件夹下  
这里我放在 /var/lib/tomcat7/webapps/vjudge/ 目录下。

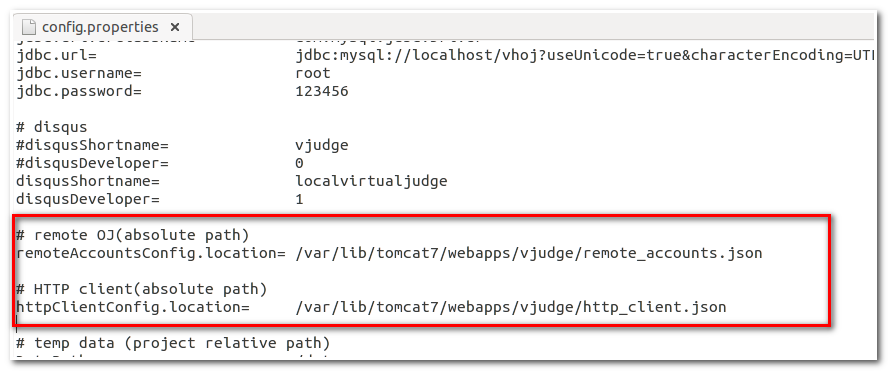
*sudo mv http\_client.json /var/lib/tomcat7/webapps/vjudge/*

*sudo mv remote\_accounts.json /var/lib/tomcat7/webapps/vjudge/*

更改/webapps/vjudge/WEB-INF/classes/的目录下config.properties文件  
将remote\_accounts.json和http\_client.json的绝对路径改为正确的路径

*cd /var/lib/tomcat7/webapps/vjudge/WEB-INF/classes/*

*sudo gedit config.properties*



PS:如果你的数据库密码不是123456，那么上面的root密码你也需要修改，默认为123456

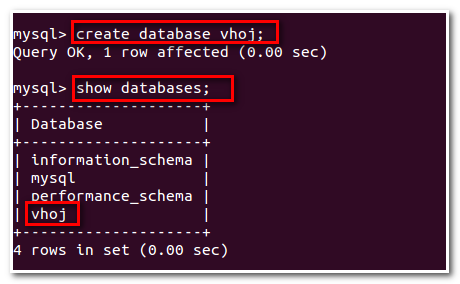
**3.5 vhoj数据库的建立**

在mysql中新建一个数据库，名字为：vhoj，  
并把表vhoj\_20141109.sql导入(可能我的版本略微旧了点)。

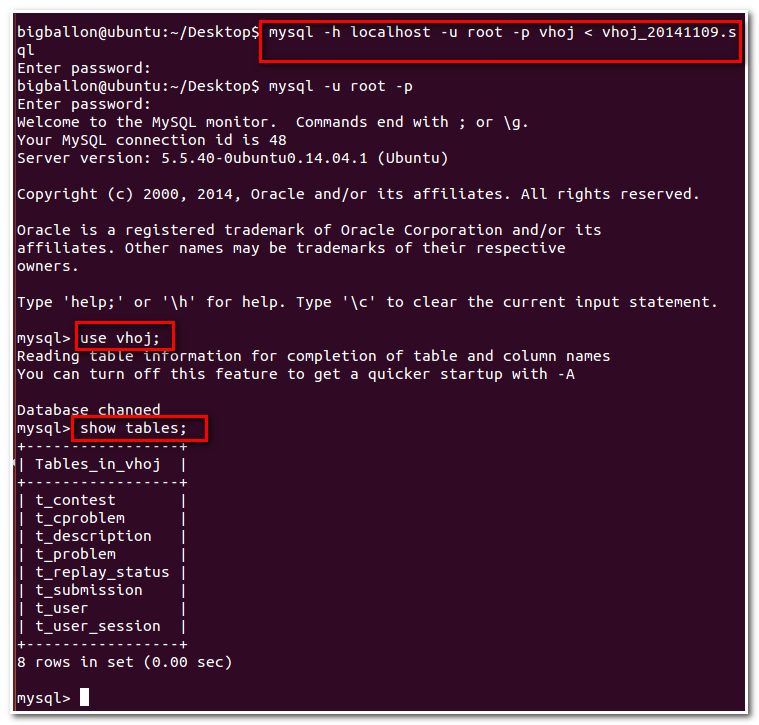
*mysql -u root -p*

*create database vhoj;*

*exit;*



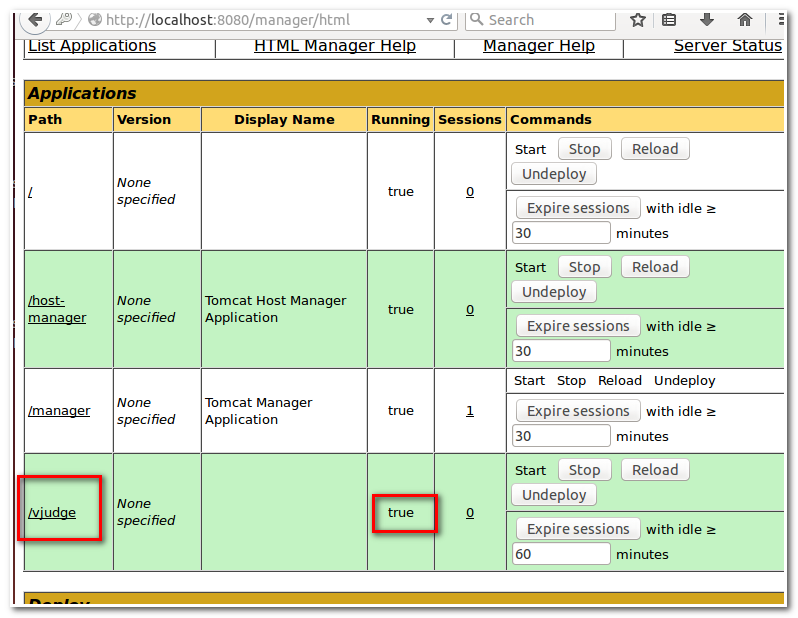
*mysql -h localhost -u root -p vhoj < vhoj\_20141109.sql*



*sudo /etc/init.d/tomcat7 restart*

最后，重启tomcat7，进入manager

查看Application，可以看到vjudge已经处于running状态了。



**4. 大功告成的Vjudge**

在地址栏输入localhost:8080/vjudge，进入搭建成功的vjudge。



**到此为止，Virtual Judge 终于搭建成功，just enjoy it！**