

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>

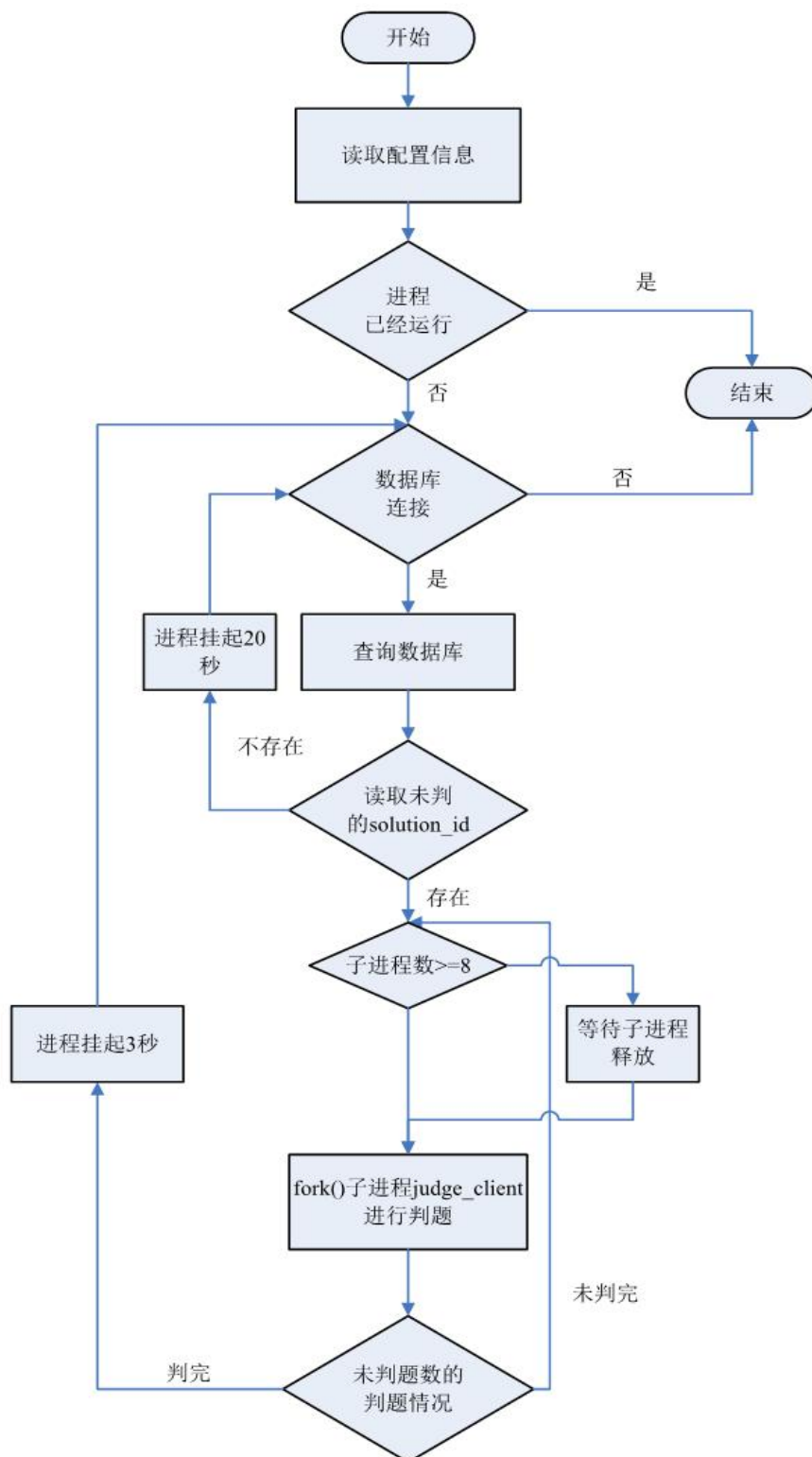
1 . judged

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。

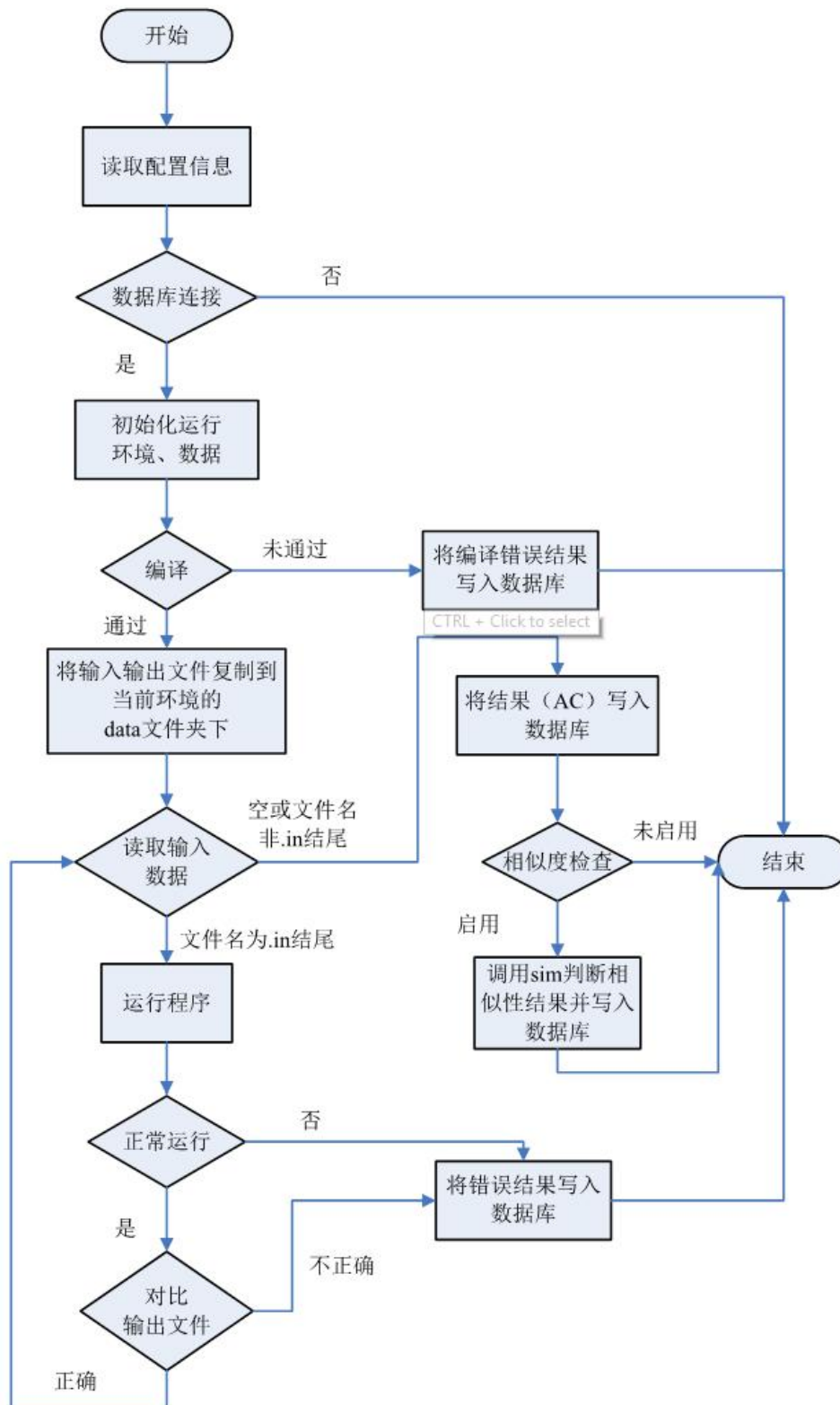
2 . judge_client

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

[\[http://www.ibm.com/developerworks/cn/linux/kernel/syscall/part1/appendix.html\]](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

3 . 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。

4 . 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、 source_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

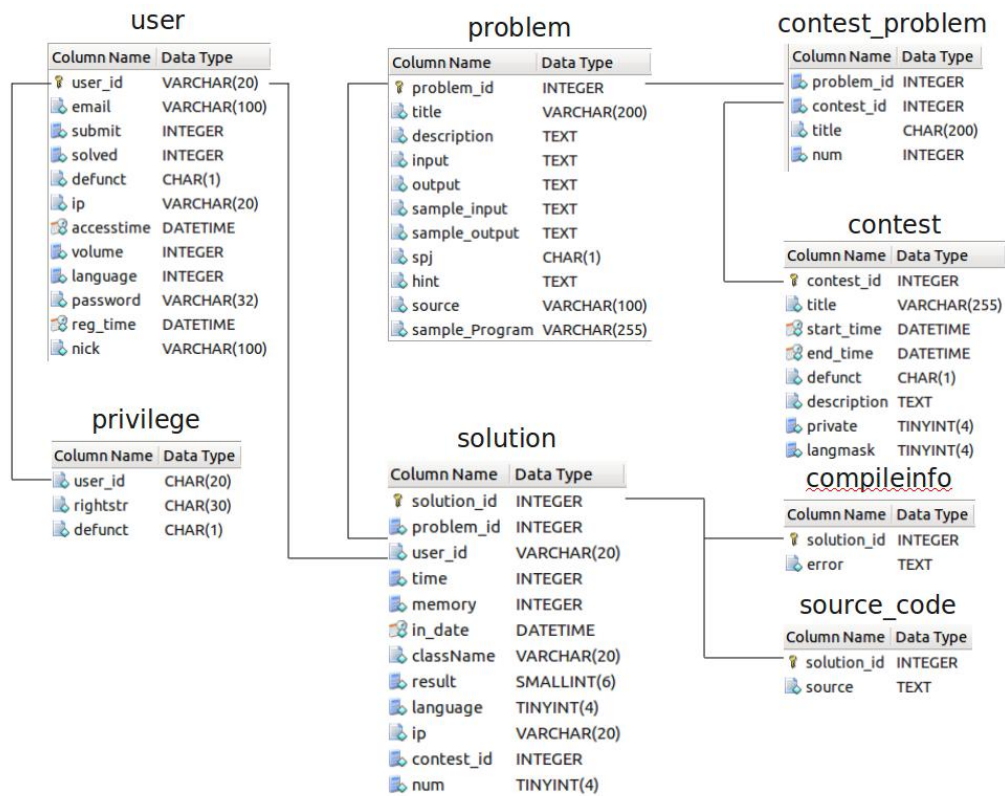
`static $OJ_MEMPORT=11211;` memcached 的端口

`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

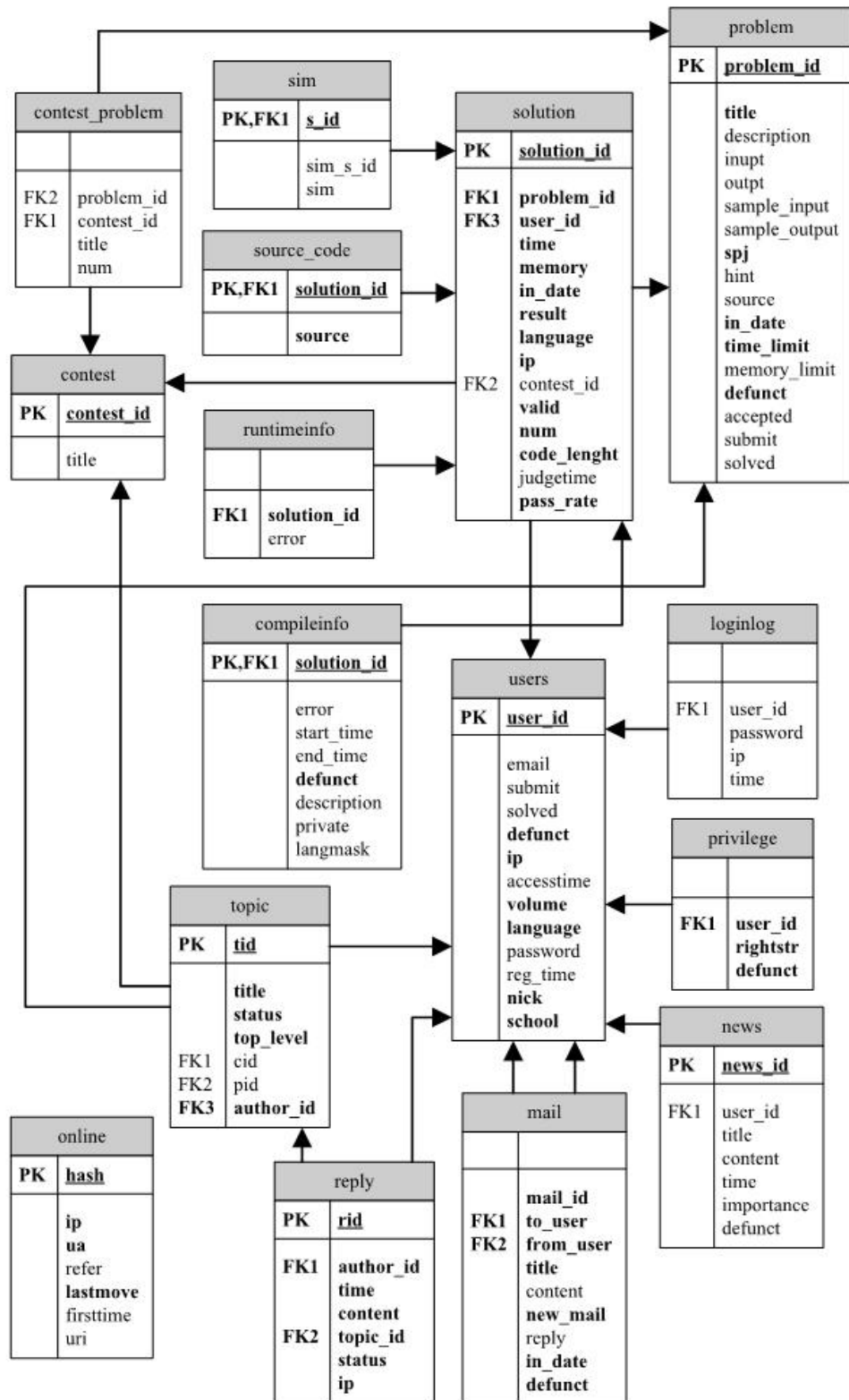
- 1.Web 插入 Solution 表
2. core 访问 Web 端 admin/problem_judge.php , 发现新纪录

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 solution 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)

Hust0j 数据库 (r1292)			
序号	表名	作用	备注
1	compileinfo	记录编译错误的记录	
2	contest	竞赛表	
3	contest_problem	竞赛题目	
4	loginlog	登入日志	记录正确与错误的登入日志
5	mail	消息列表	
6	news	新闻表	
7	online	用户在线数据统计?	
8	privilege	权限授予	
9	problem	题目表	
10	reply	论坛(帖子及回复)表	
11	runtimeinfo	运行错误信息 (Runtime Error)	
12	sim	相似度检测表	用于防作弊
13	solution	程序运行结果记录	
14	source_code	提交的源码	
15	topic	论坛帖子表	
16	users	用户信息	
17	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	datetime		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

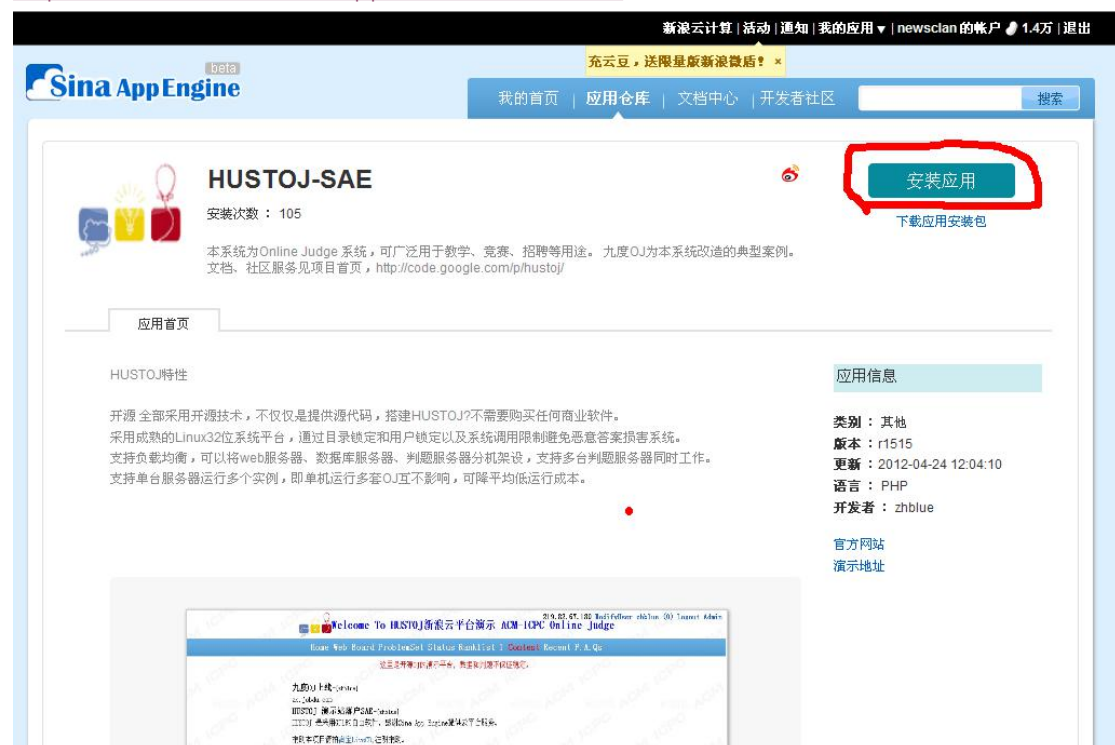
2 . 执行命令

```
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

应用向导 - 应用安装

安装代码

SAE的应用安装基于应用安装包(.zip文件)。

通过应用仓库安装的应用不可直接修改代码，如有修改需求，安装以后需要到“代码管理”面板创建SVN仓库。

- 您可以输入应用安装包的URL或者从本地上传应用安装包进行安装。
- 应用安装包应为ZIP格式，并包含符合SAE标准的 应用安装描述脚本。[详细说明](#)

基于URL安装

☐

选择文件

hustoj-sae.zip

1.3/1.9 MB,上传速度: 158 KB/秒; 剩余时间: 00:08; 进度 67%

填写二级域名

.sinaapp.com

安装到以上位置

☒

选择文件

hustoj-sae.zip

3. 初始化应用（建库）

您已经完成应用代码的部署和服务的初始化,但还需要您进行应用本身的初始化工作.请点击[此处进入初始化页面](#) 点此管理该应用.

#SAE应用推荐# 我发现 HUSTOJ-SAE不错~~，你也来试试吧!http://sae.sina.com.cn/?m=apps&a=detail&aid=92

发布微博

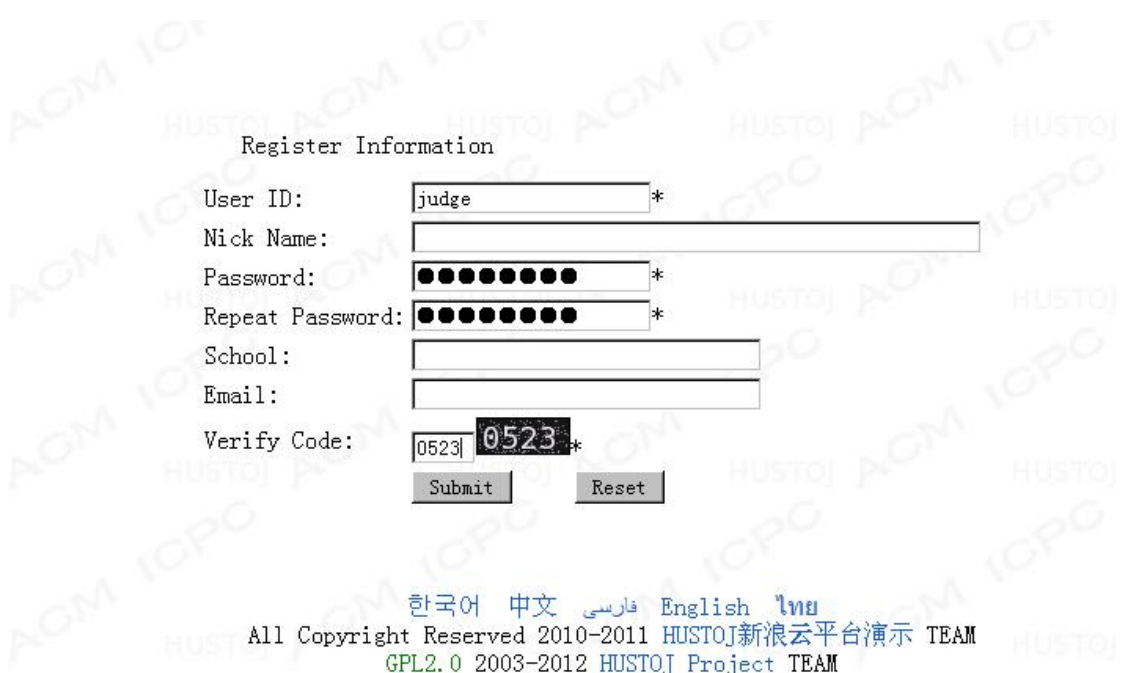
4. 注册 admin 账号



数据库初始化完成，请到[首页](#)注册admin并登录，自动取得管理员权限，完成后请删除此页面。



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

Add New problem


Problem Id: New Problem

Title:

Time Limit: S

Memory Limit: MByte

Description:



A+B

Input:



AB

Output:



8. 管理应用



9. 修改 data 为私有



10 . 管理测试数据

Domain管理

Domain Name	访问权限	简介	防盗链	操作
web	public	数据存储	否	管理 修改属性 删除
data	private	数据存储	否	管理 修改属性 删除

防盗链提示

可以增加更多测试数据

Storage»data»1»

[\[新建文件夹\]](#) [\[上传文件\]](#) [\[返回上一级\]](#)

<input type="checkbox"/> 名称	类型	上传日期	大小	缓存过期	操作
<input type="checkbox"/> sample.in	in	2012/05/13 08:34:08	3 Bytes	未设置	删除
<input type="checkbox"/> sample.out	out	2012/05/13 08:34:08	1 Bytes	未设置	删除
<input type="checkbox"/> test.in	in	2012/05/13 08:34:08	3 Bytes	未设置	删除
<input type="checkbox"/> test.out	out	2012/05/13 08:34:08	1 Bytes	未设置	删除
<input type="checkbox"/> 删除					

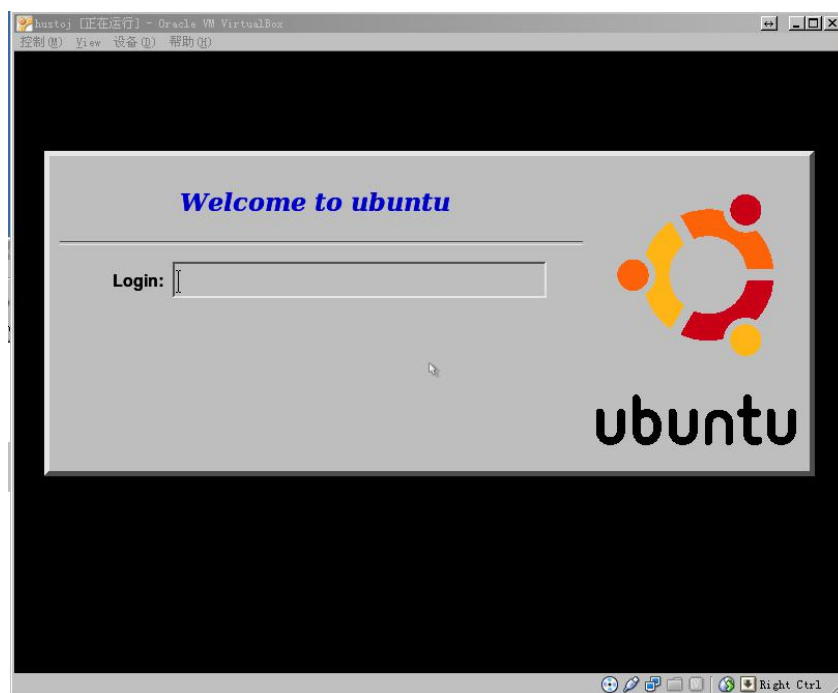
转到: 返回条数: [GO](#) [上页](#) [下页](#)

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




```
OJ_HTTP_JUDGE=1
OJ_HTTP_BASEURL=http://hustojdemo1.sinaapp.com/
OJ_HTTP_USERNAME=judge
OJ_HTTP_PASSWORD=12345678
```

重启服务

```
ubuntu@ubuntu:/home/judge/etc$ sudo /etc/init.d/judged restart
```

找到题目

1. NewsList
2. AddProblem
3. ProblemList
4. AddContest
5. ContestList
6. TeamGenerator
7. SetMessage
8. ChangePassWD

CheckToNewContest

PID	Title	Date	Status	Delete	Edit	TestData
1	A+B	2012-05-13 08:34:06	Reserved	Delete	Edit	TestData

CheckToNewContest

13. 测试提交

58.100.180.225 root@ubuntu:~\$ ssh root@100.100.100.100

Welcome To HUSTOJ新浪云平台演示 ACM-ICPC Online Judge

Home Web Board ProblemSet Status Ranklist Contest Recent F.A.Qs

Problem 1

Language: C++

8pt CPP

1#include <iostream>
2using namespace std;
3int main()
4{
5 int a, b;
6 while(cin >> a >> b)
7 cout << a+b << endl;
8 return 0;
}

Position: Ln 8, Ch 2 Total: Ln 8, Ch 125

☒ Toggle editor

Submit Reset

14. 创建 SVN 库

hustojdemo1 » 代码管理

[点此创建一个版本]

设为默认	版本	链接	xhprof调试	错误显示	操作
	1 部署于 2012-05-13 08:22:52	http://1.hustojdemo1.sinaapp.com	点击开启 1小时	点击关闭	创建SVN库 上传代码包 删除
代码部署说明: 1. 请使用svn部署代码。 2. 文件名或目录名不允许含有以下字符: ",*,?,<,>, ", 另外文件或文件名的开始与结束也不允许有空格。					

编辑代码

设为默认	版本	链接	xhprof调试	错误显示	操作
	1 部署于 2012-05-13 08:22:52	http://1.hustojdemo1.sinaapp.com	点击开启 1小时	点击关闭	编辑代码 上传代码包 删除

15. 临时设 vcode 为 false，测试，待题目 AC 再改回。

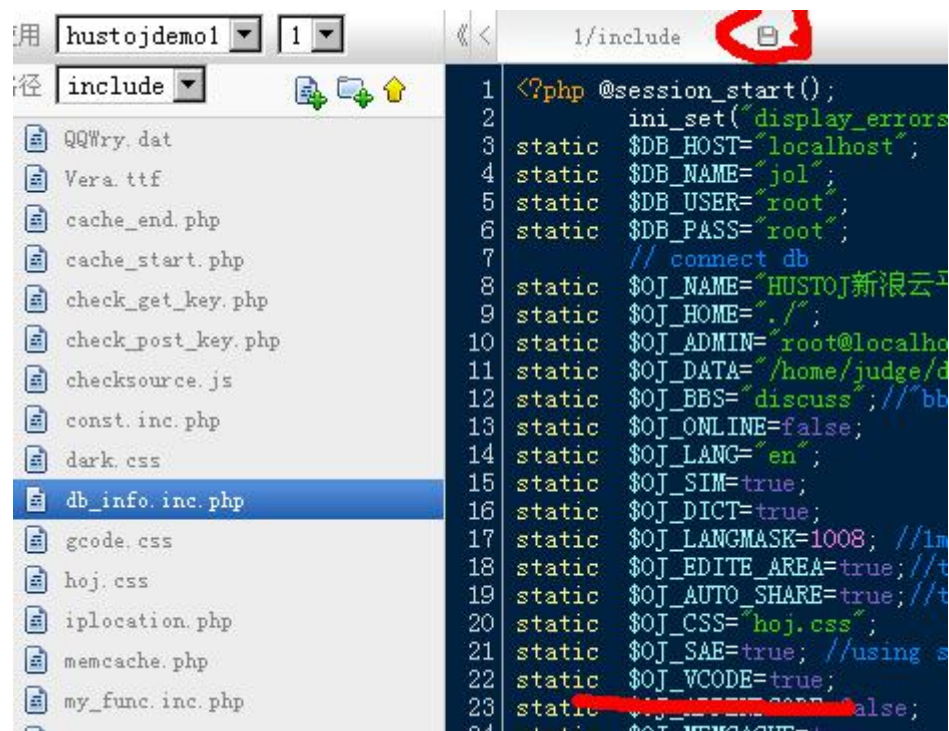
SAE CLOUD EDITOR APP:hustojdemo1 VERSION:1

应用 hustojdemo1 1 1/include

路径 include

- QQWry.dat
- Vera.ttf
- cache_end.php
- cache_start.php
- check_get_key.php
- check_post_key.php
- checksource.js
- const.inc.php
- dark.css
- db_info.inc.php
- gcode.css
- hoj.css
- iplocation.php
- memcache.php
- my_func.inc.php
- online.php
- pie.js
- problem.php
- profile.php
- set_get_key.php
- set_post_key.php
- simple_html_dom.php
- sortTable.js
- underlineTranslation.js
- wz_jsgraphics.js

```
1 <?php @session_start();
2 ini_set("display_errors","Off");
3 static $DB_HOST="localhost";
4 static $DB_NAME="hoj";
5 static $DB_USER="root";
6 static $DB_PASS="root";
7 // connect db
8 static $OJ_NAME="HUSTOJ新浪云平台演示";
9 static $OJ_HOME="http://1.hustojdemo1.sinaapp.com/";
10 static $OJ_ADMIN="root@localhost";
11 static $OJ_DATA="/home/judge/data/";
12 static $OJ_BBS="discuss"; //bbs for phpBB3 bridge or "discuss" for mini-for
13 static $OJ_ONLINE=false;
14 static $OJ_LANG="en";
15 static $OJ_SIM=true;
16 static $OJ_DICT=true;
17 static $OJ_LANGMASK=1008; //1mC 2mCPP 4mPascal 8mJava 16mRuby 32mBash 1008 f
18 static $OJ_EDITE_AREA=true; //true: syntax highlighting is active
19 static $OJ_AUTO_SHARE=true; //true: One can view all AC submit if he/she has
20 static $OJ_CSS="hoj.css";
21 static $OJ_SAE=true; //using sina application engine
22 static $OJ_VCODE=false;
23 static $OJ_APPENDCODE=false;
24 static $OJ_MEMCACHE=true;
25 static $OJ_MEMSERVER="127.0.0.1";
26 static $OJ_MEMPORT=11211;
27 static $SAE_STORAGE_ROOT="http://hustoj-web.stor.sinaapp.com/";
28 if($OJ_SAE) $OJ_DATA="saestor://data/";
29 //if(date('H')<5||date('H')>22||isset($_GET['dark'])) $OJ_CSS="dark.css";
30 if (isset($_SESSION['OJ_LANG'])) $OJ_LANG=$_SESSION['OJ_LANG'];
31
32 if($OJ_SAE) {
33     // for sae.sina.com.cn
34     mysql_connect(SAE_MYSQL_HOST_M.':'.SAE_MYSQL_PORT,SAE_MYSQL_U
35     $DB_NAME=SAE_MYSQL_DB;
36 }else{
37     //for normal install
38     if(!mysql_pconnect($DB_HOST,$DB_USER,$DB_PASS))
39         die("Could not connect: ".mysql_error());
40 }
41 // use db
42 mysql_query("set names utf8");
43 if(!$OJ_SAE)mysql_set_charset("utf8");
44
45 if(mysql_select_db($DB_NAME));
46 else die("Can't use foo: ".mysql_error());
47 //synchronize php and mysql server
48 date_default_timezone_set("PRC");
49 mysql_query("SET time_zone ='+8:00'");
50
```



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

5 . 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['UserFilePath'] =  
'/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 `kill 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

一．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 ϵ 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。

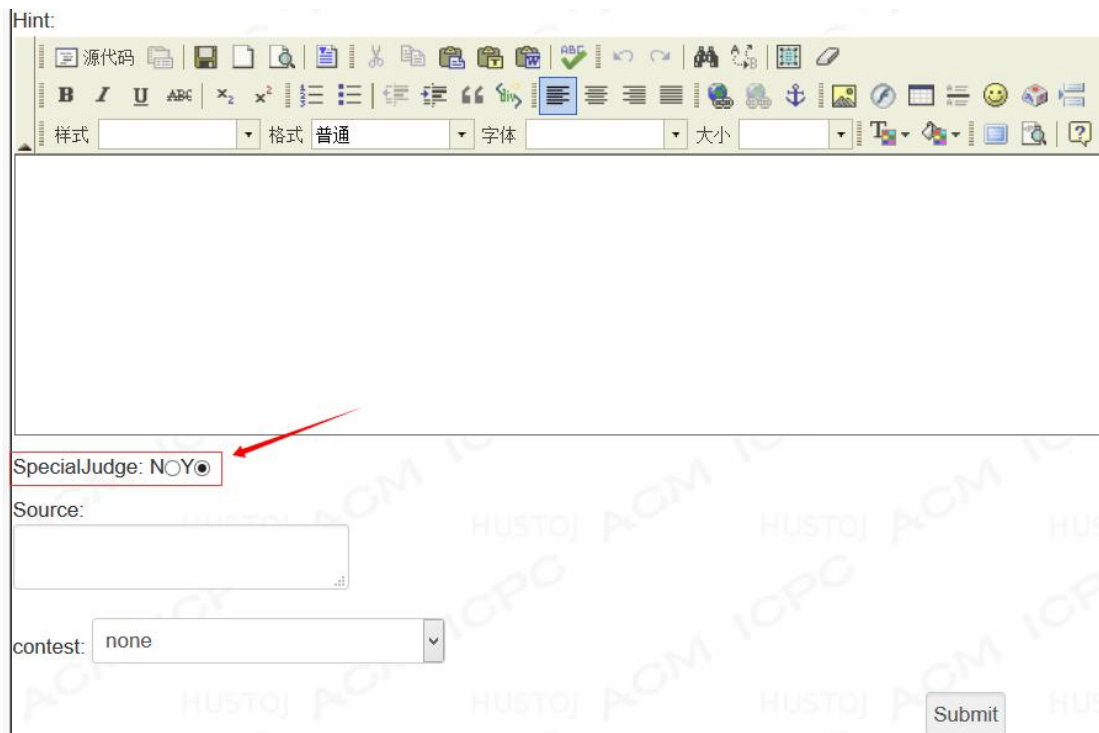
把 spj 本身看做一道题目,输入是前述三个文件,输出是程序退出时的返回值,返回值是 0 表示 AC,非零表示 WA。

可以到 [freeproblemset](#) 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

Hint:



SpecialJudge: NOY

Source:

contest: none

Submit

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	3
2	12
3	15
4	1000

filename: data.out

1	8	4
2	9	6
3	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 方面的知识不是很多 , 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

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> (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 -xvzf jdk1.7.0_80
```

2.) 移动文件夹到指定目录下

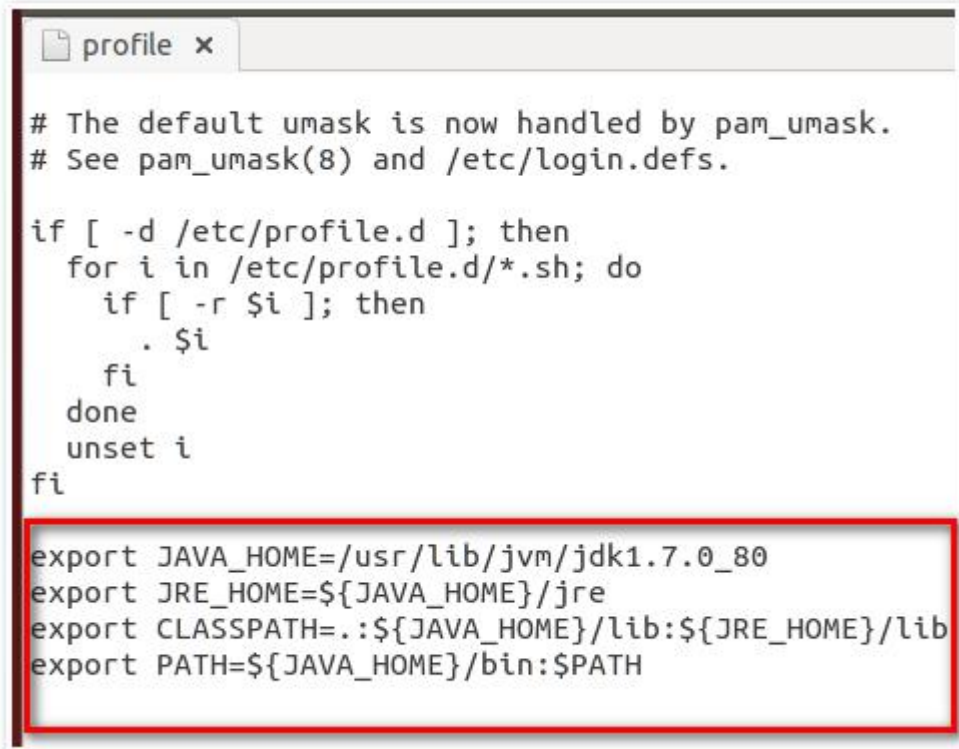
```
sudo mkdir /usr/lib/jvm
sudo mv jdk1.7.0_80/ /usr/lib/jvm/
```

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
```



```
profile x
# The default umask is now handled by pam_umask.
# See pam_umask(8) and /etc/login.defs.

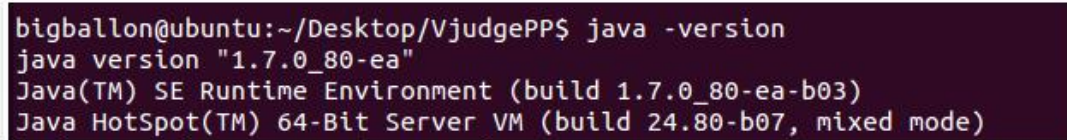
if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

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



```
bigballon@ubuntu:~/Desktop/VjudgePP$ java -version
java version "1.7.0_80-ea"
Java(TM) SE Runtime Environment (build 1.7.0_80-ea-b03)
Java HotSpot(TM) 64-Bit Server VM (build 24.80-b07, mixed mode)
```

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 xvf 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：

```
root@ubuntu: /home/bigballon
root@ubuntu:/home/bigballon# mvn -v
Warning: JAVA_HOME environment variable is not set.
Apache Maven 3.0.5
Maven home: /usr/share/maven
Java version: 1.7.0_65, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-7-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.13.0-44-generic", arch: "amd64", family: "unix"
root@ubuntu:/home/bigballon#
```

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。

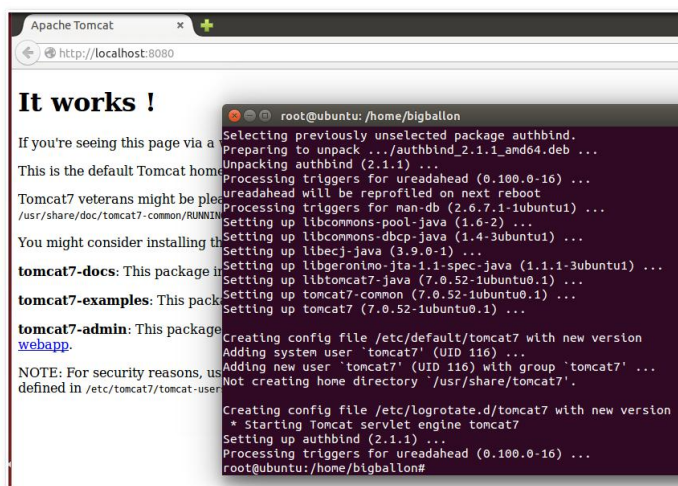
2.4 Tomcat7 的安装和配置

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

安装完成后，我们还需要安装一个 tomcat7-admin

```
sudo apt-get install tomcat7-admin
```

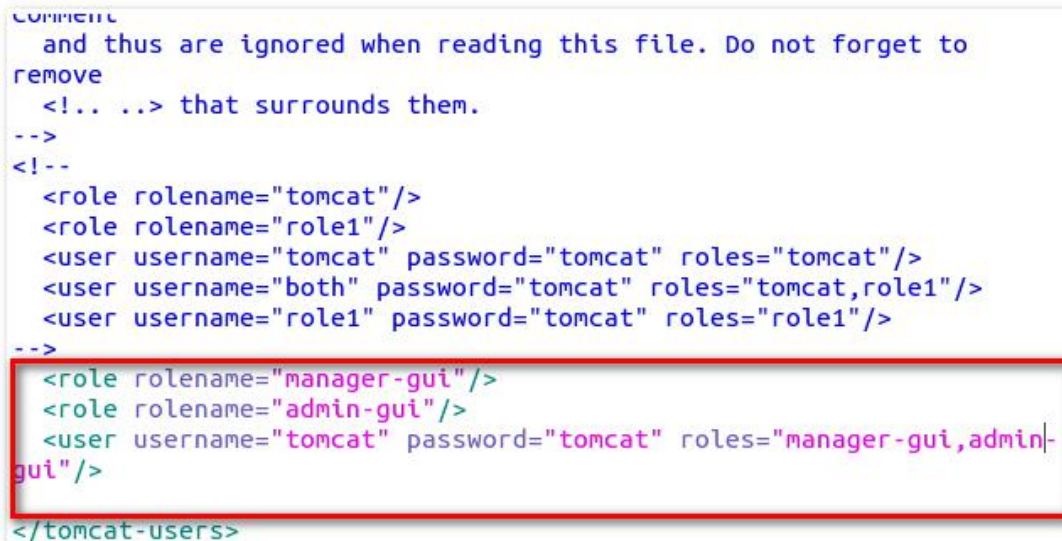
3.) 修改账户

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"/>
```



```
Comment  
and thus are ignored when reading this file. Do not forget to  
remove  
<!-- ... --> that surrounds them.  
-->  
<!--  
<role rolename="tomcat"/>  
<role rolename="role1"/>  
<user username="tomcat" password="tomcat" roles="tomcat"/>  
<user username="both" password="tomcat" roles="tomcat,role1"/>  
<user username="role1" password="tomcat" roles="role1"/>  
-->  
<role rolename="manager-gui"/>  
<role rolename="admin-gui"/>  
<user username="tomcat" password="tomcat" roles="manager-gui,admin-  
gui"/>  
</tomcat-users>
```

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。

tomcat7-docs: This package installs a web application that allows you to browse the Tomcat 7 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat7-examples: This package installs a web application that allows you to access the Tomcat 7 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat7-admin: This package installs two web applications that help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

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。

Server Information		
Tomcat Version	JVM Version	JVM Vendor
Apache Tomcat/7.0.52 (Ubuntu)	1.7.0_65-b32	Oracle Corporation

```
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
```

Server Information		
Tomcat Version	JVM Version	JVM Vendor
Apache Tomcat/7.0.52 (Ubuntu)	1.7.0_80-ea-b03	Oracle Corporation

OK，tomcat7 的 JDK 版本修改完毕。

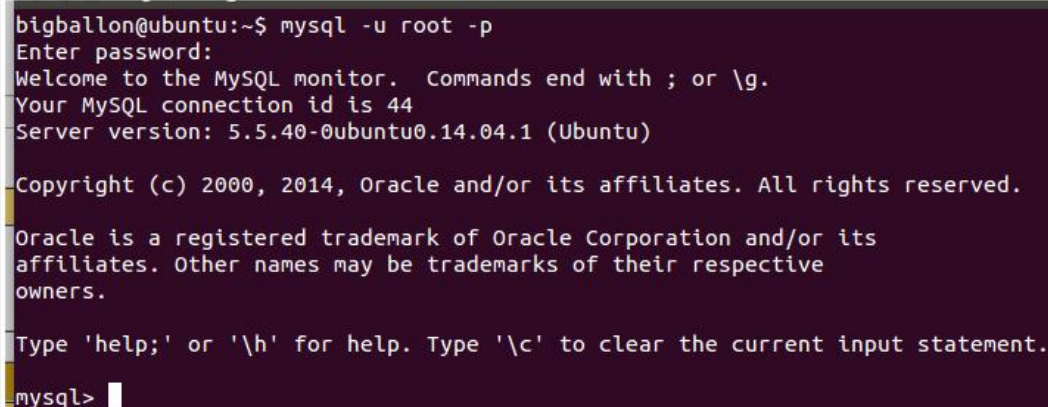
2.5 Mysql 的安装

```
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
```

A terminal window with a dark purple background and light green text. The user 'bigballon' is at the 'ubuntu' prompt. They run 'mysql -u root -p'. The prompt changes to 'Enter password:'. After an invisible password entry, the MySQL monitor starts. It displays: 'Welcome to the MySQL monitor. Commands end with ; or \g.', 'Your MySQL connection id is 44', and 'Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)'. It also shows copyright information for 2000, 2014, Oracle and its affiliates. The prompt is 'mysql>' with a cursor.

```
bigballon@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

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/
```

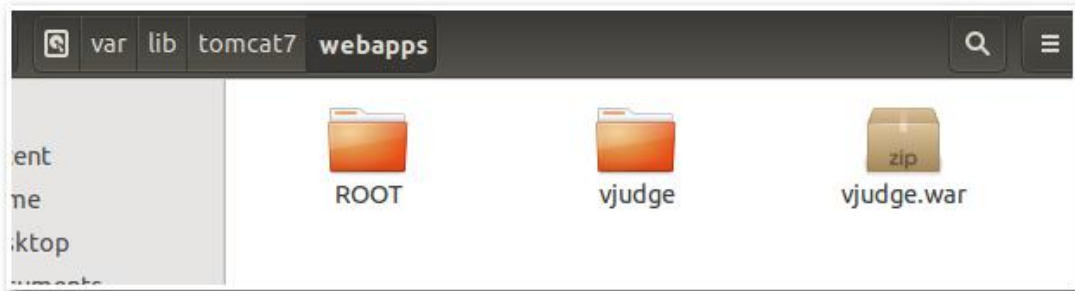
用 Maven 将 Virtual Judge 打包

```
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。

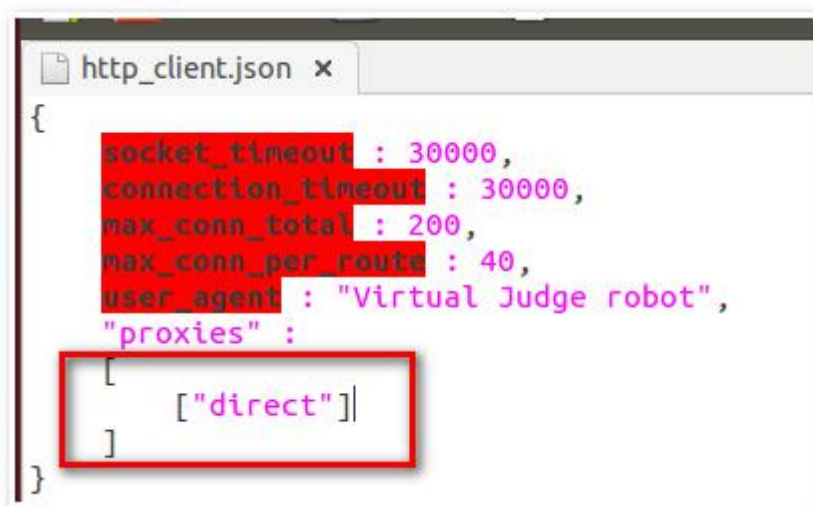


3.2 remote_accounts.json 文件的编辑

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

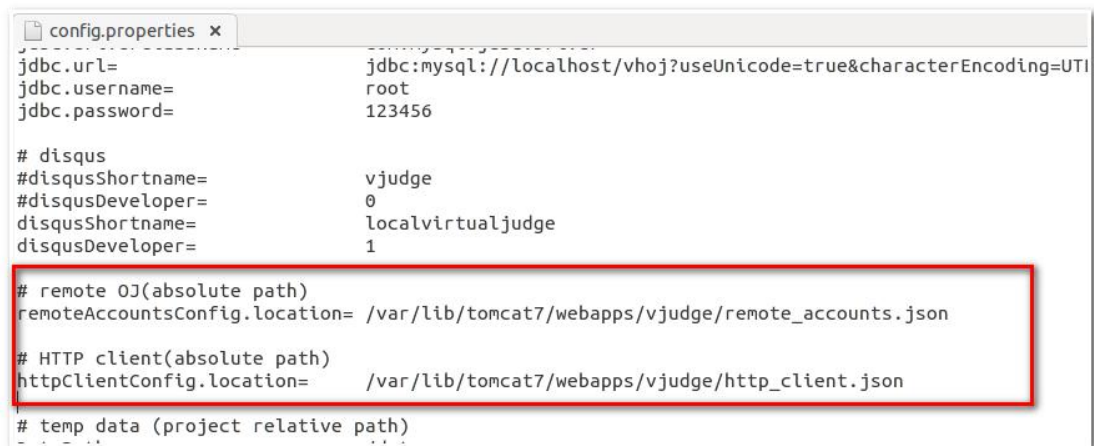
这里我放在 `/var/lib/tomcat7/webapps/vjudge/` 目录下。

```
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
```



```
config.properties x
jdbc.url=jdbc:mysql://localhost/vhoj?useUnicode=true&characterEncoding=UTF-8
jdbc.username=root
jdbc.password=123456

# Disqus
#disqusShortname=vjudge
#disqusDeveloper=0
disqusShortname=localvirtualjudge
disqusDeveloper=1

# remote OJ(absolute path)
remoteAccountsConfig.location=/var/lib/tomcat7/webapps/vjudge/remote_accounts.json

# HTTP client(absolute path)
httpClientConfig.location=/var/lib/tomcat7/webapps/vjudge/http_client.json

# temp data (project relative path)
```

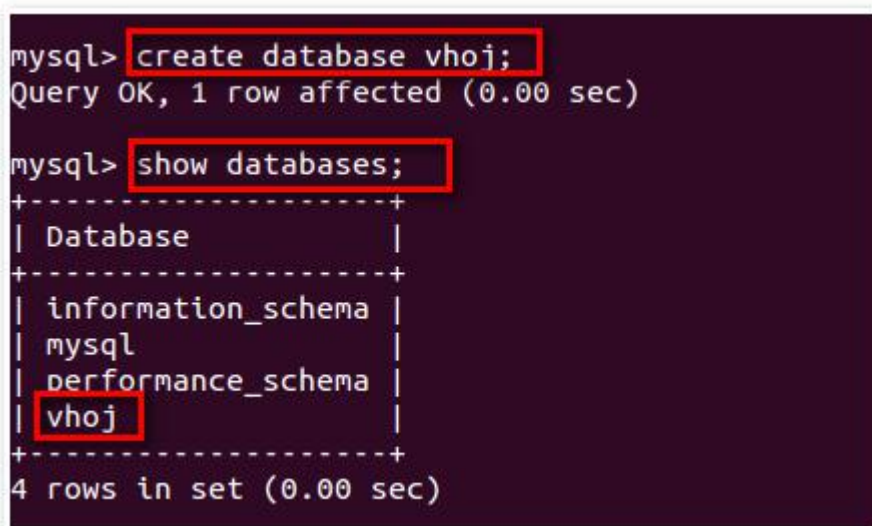
PS:如果你的数据库密码不是 123456，那么上面的 root 密码你也需要修改，默认为 123456

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> create database vhoj;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| vhoj |
+-----+
4 rows in set (0.00 sec)
```

```
mysql -h localhost -u root -p vhoj < vhoj_20141109.sql
```

```

bigballon@ubuntu:~/Desktop$ mysql -h localhost -u root -p vhoj < vhoj_20141109.s
ql
Enter password:
bigballon@ubuntu:~/Desktop$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.5.40-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use vhoj;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_vhoj |
+-----+
| t_contest      |
| t_cproblem     |
| t_description  |
| t_problem      |
| t_replay_status|
| t_submission   |
| t_user         |
| t_user_session |
+-----+
8 rows in set (0.00 sec)

mysql>

```

`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。

http://localhost:8080/manager/html

List Applications HTML Manager Help Manager Help Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/vjudge	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 60 minutes

4. 大功告成的 Vjudge

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

Virtual Judge

http://localhost:8080/vjudge/toIndex.action

HOME PROBLEM STATUS CONTEST REGISTER LOGIN

Virtual Judge

Virtual Judge is not a real online judge. It can grab problems from other regular online judges and simulate submissions to other online judges. It aims to enable holding contests when you don't have the test data.

Currently, this system supports the following online judges:

POJ	ZOJ	UVALive	SGU	URAL	HUST
SPOJ	HDU	HYSBZ	UVA	CodeForces	Z-Trening
Aizu	LightOJ	UESTC	NBUT	FZU	CSU
SCU	ACdream				

What's new:

- 2014-10-15 UI update. Please [contact me](#) if you find any bugs. Thanks very much!
- 2014-09-20 Currently Virtual Judge has two access address:
 - <http://acm.hust.edu.cn/vjudge> : Since it was redirected to vjudge.net permanently incorrectly, you need clear your browser cache to use it. Recommended for Chinese visitors.
 - <http://vjudge.net> : Reverse proxy hosted overseas. Recommended for visitors who can't use the above entrance.
- 2010-10-07 Refer to [this page](#) for recent change of Virtual Judge.

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