Pain-free Programming Contest Management

Lee Wittenberg
Kean University
Greater New York Region
leew@eve.kean.edu

Abstract

Managing a programming contest involves many tasks, the most difficult, time-consuming, and error-prone of which is reconfiguring the machines and networks to be used. This paper describes a Linux-based system that allows a contest to be run without having to reconfigure any machines or the network. This system simplifies other necessary tasks as well. It has been used successfully in the Greater New York Region for competitions since 2005.

1 Introduction

A programming contest environment has very stringent requirements above and beyond those necessary for normal computing: it must isolate contestants from the Internet (and other contestants); it must prevent access to portable data drives and non-contest-related partitions on the hard drive; and it must provide the capability, once the contest is over, to quickly restore the machines used to their original configuration.

The first time I ran the technical end of a programming contest, it was necessary to create separate, hidden, "contest" partitions on each machine used, set these partitions as "bootable" for the contest, and to hide them again (making the original partitions bootable) when the contest was over.[*] In addition, the campus network had to be reconfigured to disallow Internet access.

In contrast, our current system requires no reconfiguration at all. Each PC is booted from a CD-ROM or USB stick containing the entire contest

[*] A number of contest-specific tools had to be developed for this purpose. I still have a small stack of 3.5" floppy disks, labeled "flip-flop," used for switching bootable and hidden partitions.

environment, which is created and thoroughly tested well before the date of the contest.

2 The Server

The key to the system is the server. It need not be a particularly powerful machine.[*] Our current server is a Dell PowerEdge SC1425 with two dual-core Xeon processors (2MB internal cache), 1GB RAM, and dual 72GB SATA drives configured for RAID mirroring. It has served well for the past six years, easily dealing with about 120 clients at a time.[†]

The server provides shared disk space, plus authentication and printing services for its clients. In addition, it runs a web server (for dissemination of contest information) and the contest judging software (and can provide DHCP service, if necessary). The shared disk space allows us to easily clear contestants' home directories after the practice session and before the contest proper. It also prevents teams from losing any saved work if a machine should need to be rebooted during the contest, and allows customization of the environment while the contest is running.[‡] Providing a print server allows client machines to be restricted to specific printers without having to specially configure the clients.

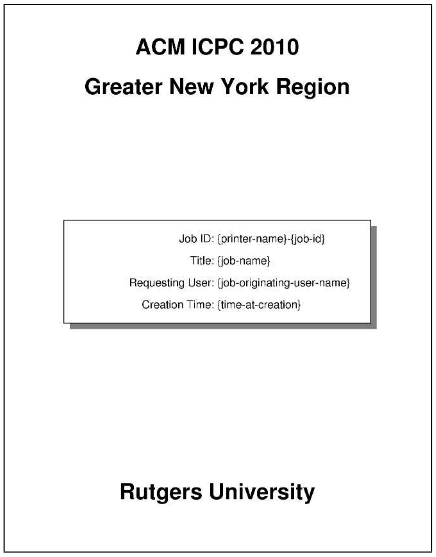
While the host institution sometimes provides its own server, we usually bring our "standard" server to the host site. The use of a portable server, maintained year-round, allows us to set up the contest configuration (networking, printers, problems, test data, etc.) well ahead of the date of the contest. For example, once the type and IP addresses of the printers are known, they can be fully configured, including a customized banner page (figure 1). If the DHCP server at the host site provides machine names as well as IP addresses, printing can be completely configured based on these names. Otherwise, some extra work must be done to assign printers to teams once the server is moved to the host site.

Placing the judging software on the server rather than the client makes it possible to defer decisions about which software (and which version of that software) to use until the last possible moment. In past contests, we have used PC^2 [3]. Changing to KATTIS [4], used in the 2010 Finals, would be

[*] In fact, I have run small contests using an old laptop as a server.

[†] The Greater New York Region is unique in that—for historical reasons—we provide two machines per team rather than one.

[‡] For example, the default screensaver cycles through a collection of contest photographs maintained on the server. New pictures are added throughout the contest, and are automatically included in the rotation.



ACM ICPC 2010

Greater New York Region

Job ID: {printer-name}-{job-id}
Title: {job-name}
Requesting User: {job-originating-user-name}
Creation Time: {time-at-creation}

Rutgers University

Figure 1: Customized Print Banner for 2010 Regionals.

straightforward. Both systems can co-exist on the server (as can multiple versions of the same software), and be activated as necessary. We have even been able, on occasion, to install and test last minute bug fixes to the judging software shortly before contest day, when it would have been impossible to reconfigure the clients.

The server configuration also includes a series of scripts, developed over the years, to ease the tasks of administration and judging. For example, we have a tradition in our region that when a team presses the PC^2 "Test" button, the specified program is run with sample data from the problem statement. However, when the program is judged, it must be run using the judges' test data. The "`run`" script (figure 2) provides this capability (note how the script adjusts for, and hides the complexity of, changes from PC^2 v8.7 to v9). Some other scripts provided are those to synchronize PC^2 and Linux passwords, to generate account information handouts (using LaTeX), to clear team home directories after the practice contest, and to allow Internet and USB drive access on judges' machines. Changing to KATTIS, or some other judging software, would naturally require some adjustment (or even rewriting) of these scripts. However, the necessary changes should be, for the most part, straightforward.

```
#!/bin/sh
#
# usage:  run executable data-file [interpreter]
#    (DATA_HOME env. variable may override data file location}
#
DATA_HOME=${DATA_HOME:-~pc2}    # assume data dirs are under ~pc2
EXEC="$1"
# kludge to deal with $2 being generated via {:ansfile}
# instead of {:problem}, as in previous versions
DATA="`basename $2 .out`.in"
INTERP="$3"
gid=`id -gn`
case "$gid" in
judge)  DATA="$DATA_HOME/judge-data/$DATA"
    ;;
*)  DATA="$DATA_HOME/data/$DATA"
    ;;
esac

cp -f "$DATA" .
exec $INTERP $EXEC < $DATA
```

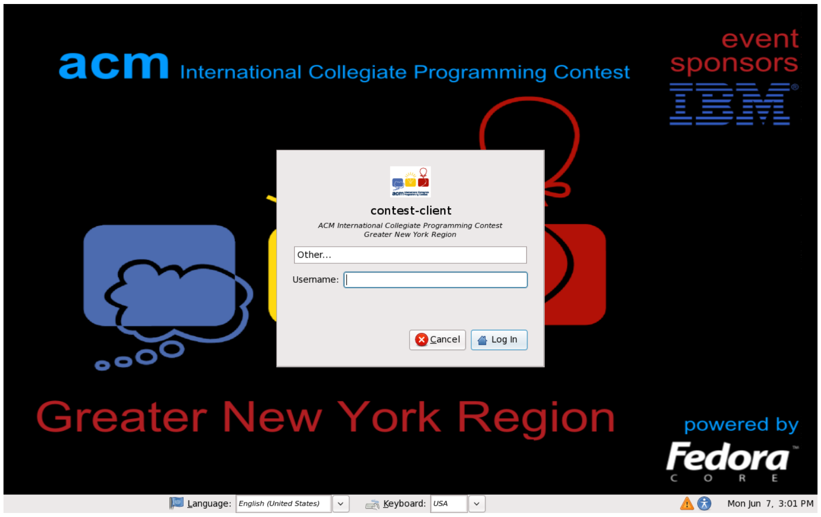
Figure 2: The “run” script for executing submitted programs.

3 The Client CD

Using a bootable CD, of course, removes the necessity of configuring special contest partitions on client machines. While using a network boot image would work as well, it would also necessitate some reconfiguration of the host site network, and the goal here is to eliminate as much reconfiguration as possible, so the CD is preferable. Early images were created using Bernhard Bablock’s BBLCD toolkit [2]; more recently, Tomas Matejicek’s excellent Linux Live [5] has done the job. The current version of the CD works on Intel-based iMacs as well as PC’s.

The client CD provides a complete Linux-based programming environment: editors, compilers, debuggers, etc. It is firewalled to prevent communication with any machine other than the server (thus removing any need to reconfigure the local network), and configured to prevent user access to USB drives (and other portable devices).[*] The environment runs off a ramdisk (so rebooting the CD after the practice session is all that’s necessary to re-

[*] It does, however, allow the use of USB mice and keyboards.



contest-client

ACM International Collegiate Programming Contest
Greater New York Region

Other...

Username:

Language: English (United States)

Keyboard: USA

Mon Jun 7, 3:01 PM

Figure 3: Client CD Login Screen.

move any files a team might have left in a publicly accessible area), and it automatically makes use of any Linux swap partitions that are on the client machine's hard drive.[*]

Judges run the same environment, from the same CD image, as contestants. However, once the judge machines have been booted, a script is run to allow them both Internet and USB drive access. This makes it possible for judges to provide data files and problem information without the necessity of putting that data on the server ahead of time. It also gives judges the flexibility they need to deal with unforeseen problems and questions that may arise during the contest.

The only disadvantage to the CD is that the IP address of the server must be "hard-coded" on the image, which means that each new host site requires a new image. However, this is not a great hardship in practice, as blank CD's are quite inexpensive (we've paid about 15¢ each for CD's over the last six years). Since we regularly update to a new Linux distribution (the current image is based on Fedora Core 10), it turns out to be even less of a disadvantage. Updating takes, on average, two afternoons of work

[*] Most modern machines have enough main memory so that swap space is not necessary. We have never had a problem due to a lack of swap space, and we have several times held regionals at a host institution whose machines had no Linux swap partitions.

(most of which is spent dealing with incompatibilities introduced by new versions of the software packages used).

4 Discussion

We first used this system in the 2005 Greater New York Regionals, and have used it for our regional competitions ever since. Each year the CD image is made publicly available well in advance of the contest date for teams to practice with, enabling them to have complete familiarity with the environment ahead of time, so that precious contest time can be used exclusively for solving problems. Since this image is identical to the one used on contest day, these practice sessions will similarly involve no reconfiguration—users can simply boot from the CD in order to practice (this contrasts with the World Finals environment CD [1], which requires a separate partition on the hard drive).*

The stability of this CD-client/server system has also enabled us to stabilize preparation for the fall contest. In late spring, our technical director visits the prospective host site to determine (and work out) any possible technical problems. By August, the CD image is “frozen” (and given a final test on the actual hardware to be used). One week before the contest date, the server is moved to the host site and used to check network connectivity between the clients and the server (and between the server and the printers); minor adjustments to print server configurations are sometimes necessary at this time. The day before the contest, a “dummy” contest is run as a final check on all software and hardware being used. In addition, when significant changes have been made to the judging software, our head judge and technical director do a “shakedown” run of this software in late summer.

Of course, everything does not always go smoothly: in 2006, we could not run the “weekend-before” checks (so they were moved to the day before the contest and the practice session the next morning had to serve as final checkout); in 2007, we did not have a host site volunteer until September; and in 2009 the host site upgraded the machines in their labs over the summer (the new hardware required last-minute modifications to the CD image). However, the schedule has proven flexible enough to deal with all of these problems and run the contest successfully, with a minimum of problems on contest day.

*Before the 2010 regionals, for example, 27 copies of the image were downloaded to local, non-webcrawler addresses.

5 Conclusion

The system has proved its worth over the past six years, and we expect it to continue to function well in the years ahead. This past year, we tried a significant change in the system: We used bootable USB “thumb” drives[*] rather than CD’s. In addition to the boot partition, containing the operating system, the drive contained a Linux swap partition and a small partition where teams could save copies of their programs and data during the contest proper (this last was inaccessible during the practice session). After the competition, the drives were not only souvenirs for the contestants, but also made it unnecessary for us to provide a separate means for teams to get copies of their work.

We hope to continue using thumb drives for future competitions; the only problem is cost: thumb drives are several orders of magnitude more expensive than blank CD’s.

The system described in this paper is fully explained at the author’s website [6].

Acknowledgments

I would like to thank Adam Florence and John Buck, who first suggested the writing of this paper, and whose comments and criticism improved it significantly.

References

[1] ACM ICPC world finals contest image installation directions. `http://pc2.ecs.baylor.edu/InstallDirections.html`, October 2009.

[2] Bernhard Bablok. The BBLCD toolkit. `httpd://bblcd.berlios.de/`, October 2003. This website is no longer supported.

[3] California State University, Sacramento. CSUS programming contest control (PC^2). `http://www.ecs.csus.edu/pc2/`, June 2009.

[4] KTH Royal Institute of Technology. KATTIS: KTH automated teaching tool. `http://kattis.csc.kth.se/`, May 2010.

[*] Generously provided by Rutgers University, this year’s host institution.

[5] Tomas Matejicek. Linux live for CD & USB. `http://www.linux-live.org/`, September 2009.

[6] Lee Wittenberg. How to run a programming contest. `http://eve.kean.edu/~leew/contest/`, May 2010.