



Towards Improving Regional Contest Software Environment

Andrew Ray, Maung Htay

Presenter: Ian Barland

Outline

- North America Mid-Atlantic region: ~9 sites
- How we currently run our contest
 - Old way
 - New way
 - Reflection
- What could be improved
 - Web based system

Setup

- Decide

- Compiler versions
- Editors
- Environment

Can make a VM image – easy to wipe/restore.

- Communication

- Judges
- Local site administrators

PC²

- Two methods of configuration
 - Each site runs its own server and there is one regional master-server
 - All sites connect to servers running at a central location
- Second method works better than the first
- We pre-configure PC² for all of our sites

Pro/Con

- **Benefits**
 - Network / firewall configuration is simple
 - Local site administration is simple
- **Downsides**
 - Requires a non-trivial server
 - 4 server grade CPUs (Xeon 2.8GHz)
 - 12+ GB RAM
 - Just for the PC² servers

Dry-run

- **Mock contest**
 - Preferably a month before the real contest
 - 2 week minimum
- **All sites participate**
 - Problem submission / grading
- **Eliminates almost all last-minute issues**
 - Having spare computers is the biggest issue on the contest day

Results

- Contest runs well and we have little problem on the administration side
- PC² uses a LOT of resources
 - 10+ GB of RAM for 9 servers
 - 50-100% CPU usage
- Biggest issue this year
 - Symlinks didn't work on the webserver
 - Had to copy the scoreboard to the webserver every 10 seconds

Future

- Going from local servers to several central servers simplified matters
- Next step
 - Firewalled machine + internet connection
- Web interfaces are possible
 - e.g. KATTIS at Worlds
 - Many others are used for grading at universities

Issues

- Current approaches are not designed for programming contests
- Multi-site
 - What happens when the internet dies
- Technology
 - Testing programs
 - Load balancing
 - Security

Issues (2)

- Environment
 - All in one
 - Editor / remote file management / testing / etc...
 - UNIX approach
 - Submit a file, get a result
- We favor the UNIX approach
 - Easier to develop
 - Prevents contestants from having to learn yet another tool while under contest pressure

Interesting issues

- Data mining
 - What are the typical problems teams encounter
 - Best practices from winning teams
 - Submission times for types of problems
- Security
 - Can contestants sabotage others
 - Managing resources

Acknowledgements

- We honor the memory of Sallie Henry who spent years successfully running the contest for the Mid-Atlantic region while managing the programming team at Virginia Tech. Her lessons have inspired us and helped us continue in her footsteps.

Questions?

- We promise we won't respond with "read the problem statement"!
- Specific details or collaboration on how to best run a regional contest (on the systems administrator side)
 - aaray@radford.edu

Steps

- Is there an existing system that could be modified?
 - Doubtful, but possible
- Development of a new system
 - Hard to justify
 - No real short-term benefits for academia / industry