



A Task Development Infrastructure

Rob Kolstad, Ph.D.
USA Computing Olympiad

USA/USACO Background

- Population 307M
- Area 9.8M km² (low density → expensive to gather)
- 29,500 high schools
- 15M high school students + 0.3M 'home schooled'
- 3.2M seniors graduated in 2009
- ~160 USACO participants → 0.001% (pitiful)
- 6-24 coaches, 1-2 administrators, all volunteer effort

USACO Annual Task Budget

- Qualification round (sets bronze/silver/gold level)
- Six contests \times ~ 3 divisions \times 3 tasks/contest
→ 54 tasks
- Bonus contests: 1-2 contests \times 3 tasks/contest
→ 3-6 tasks
- Camp contenders: 6 contests \times 3 tasks/contest
→ 18 tasks
- Camp 2nd tier: 4 contests \times 3 tasks/contest
→ ~ 12 tasks

Total: 75+ new tasks annually

USACO Task Properties

- Clear/complete text
- Easy to edit/repair text
- Printable/web-able
- Task ratings and evaluation from multiple coaches
- Algorithm type
- Solution time

→ More...

USACO Task Properties, II

- Test data
 - Validator (mechanically created!?)
 - Easy to manipulate (add, remove, reorder)
 - Feedback style (one case? Many? Full?)
 - Scoring: multiple tests-per-case
- Task-style (batch, reactive, output-file)
- Solutions (with execution, of course)
- Answer-grader & output format-checker
- Analysis Text
- Translations

Task Text Contents

- Names (short & full)
- Author & Owner
- Presentation order
- Difficulty
- Text body
- Input & output formats
- Input & output samples/explanations
- Limits on time/memory

USACO Milieu

- Distributed coaching staff
 - Annually from 6 to 24 coaches
 - Distributed across USA, Canada & other continents
 - Distributed across almost all timezones
 - Medium-speed network link (1.5Mb/sec down; 0.7 up)
 - Small number of servers
-
- Requires web-style development

Automation

- Easy navigation of tasks and contests
- Front page w/pool of tasks including status
- Contests
 - Tasks
 - Configuration (start time, duration, etc.)
 - Status
- Standard task requirements
 - Sandbox, queuer, runner, checker
 - Matches contest & grading environments

Challenges/Requirements

- Security/authentication
- Multiple task development systems (!)
- Paradigms for contest export and administration
- Functionality with reliability
- Meaningful user interfaces
 - Easy to navigate
 - “Shiny” (currently a failure)
 - Efficient use of bandwidth

Current Status

- In production for more than half a decade
- Able to produce 12-20 contests/year with strict deadlines
 - Much reduced stress (due to checking tools)
 - Lower skill level required for task and contest creation
 - Perhaps reduced staff/time requirements as well
 - Vanishingly small number of complaints/clarifications

Current Deficiencies

- Linux task timing is not reliably repeatable
- Web displays are not consistent in their layout
- Web displays for list-of-tasks and list-of-contests are becoming unwieldy due to size
- Entire web layout is not “shiny” and “slick”
- Can’t make one-off custom contests for training (yet)
- Training pages not fully integrated
- Features like tracking solving times not yet implemented

Futures

- Configurable contests for individual training
- ‘Programming bees’
- Modern, consistent web interface (better layout, CSS, better use of color, improved ‘confirmers’)

Conclusion

- The system enables creation of a large volume of good-quality contests (high quality contests have better test data and perhaps better analyses)
- The system virtually eliminates administrative time expenditures and errors
- The system is now maintainable/repairable and expandable
- Basically: It works for us and can provide a model for those who need its use/functionality