CATS-Judge: Command-line utility for problem preparation

- Alexander Klenin, Mikhail Babich, Dmitry Bogdan
- Far Eastern Federal University
- Russian Federation

- 2016

… you need more problems

- Multiple contests, different rules:
  ICPC, IOI, AI, machine learning, robots...
- Competitive learning actively used in standard curricula

Problem authoring becomes a commonplace activity.

Problem preparation challenges

- High skill required
- Tight deadline
- Intense collaboration

Problem preparation tools

Standard programmer / copyeditor tools

- Compiler / debugger / IDE / VCS
- Text editor / spellchecker / publishing system

Specialized tools

- Verify packages, validate tests, sandbox, compare solutions, fuzz tes
  scores

Online preparation tools

- Polygon / Codeforces
- Good for novice problem authors
- Holds back experienced authors
- Tied to specific CCS / host

Offline preparation tools

- Polygon-cli, Kattis problemtools
- Much higher competence threshold
- Productive for expecienced authors

- Still some lock-in
- Portability limitations

CATS-Judge: platforms

- Multiple platforms:
  - Windows tier 1
  - Linux – tier 1.1
  - OS X, OpenBSD... – tier 2

CATS-Judge: languages

- Multiple programming languages
  - Automated detection
  - Windows: MinGW vs CygWin vs MSVC …
  - OS X: XCode vs macports vs homebrew …
  - Linux: system Perl/Ruby/PHP vs user's Perl/Ruby/PHP
  - Everywhere: Python 2 vs Python 3 …
  - Rare/exotic: Haskell, Pascal, Erlang ...

CATS-Judge: formats and backends

Format conversion by mixing package formats with backends

- CATS
- Polygon
- *<insert-your-system-here>*

CATS-Judge: basic features

- Download problem package
- Parse/validate problem package
- Install problem package
- Run one or more solutions
- Use sandbox
- Modify/build/test cycle
- Upload problem package

CATS-Judge: console UI

```
j r -pr A.zip -run sol_ask.pp -run sol_naive.pp
  -t 20-25 -result-c 'T|R'
```

sol_ask.pp			sol_naive.pp		
Test	Result	Time	Test	Result	Time
20	OK	0.007947	20	OK	0.007419
21	OK	0.007884	21	OK	0.007502
22	OK	0.01251	22	OK	0.031876
23	OK	0.010969	23	TL	0.992918
24	OK	0.011563	24	OK	0.010854
25	OK	0.011725	25	OK	0.010913

CATS-Judge: advances features

- Interactive problems
- Complicated scoring rules: test groups, partial results etc.
- Inter-package dependencies
- Open source: GPL
- Combined server / client: `j serve` vs `j run`

CATS-Judge: in progress / planned features

- More backends
- Score fuzzing:
  - avoid score collisions with multiple test groups with dependencies
- Multi-run / adversarial problems, e.g. multi-player AI
  - generalized pipes
- Mutli-threaded solutions: CPU affinity, max/total core time

Questions?

- Alexander Klenin, Mikhail Babich, Dmitry Bogdan
- Far Eastern Federal University
- Russian Federaiton
- klenin@gmail.com

- http://github.com/klenin/cats-judge
- http://github.com/klenin/Spawner
- http://imcs.dvfu.ru/cats/?lang=en