StarExec
A Web Service for Evaluating Logic Solvers

Aaron Stump, CS, U. Iowa
Geoff Sutcliffe, CS, U. Miami
Cesare Tinelli, CS, U. Iowa
What is a **Logic Solver**?

Program to test validity of formulas

Formula

Solver

Valid

Invalid
Why are Solvers Useful?

- Can **encode** many problems in logic
  - Verification, static analysis
  - Testcase generation
  - Planning, knowledge representation, etc.

- **Efficient** implementations
  - Boolean reasoning (SAT) is NP-complete
  - Still: can handle huge (megabytes) formulas
  - Many optimizations, heuristics
Logic solvers used heavily for verification

- Express verification problem in logic
- Dispatch formulas to solver

\[ SAT(\text{Reachable}(n) \land \text{Error})? \]
Different Logics

- Many **different kinds** of logics, solvers
  - Restrictions, assumptions => different logics
  - SAT, SMT (SAT Modulo Theories)
  - First-order (subkinds), QBF, MAXSAT, etc.
  - Different research communities

- Different algorithms, characteristics
  - NP-complete, worse (SAT, SMT, QBF)
  - Undecidable (First-order, certain theories)
Example logic: SMT

Propositional logic + theories + quantified axioms

\[(\text{data\_in} \land \neg \text{queue\_full}) \rightarrow \text{enqueue\_next}\]

\[(x = y + z \land f(x) > f(y) > f(0)) \rightarrow x \neq z\]

\[\forall x, y. \ len(\text{nil}) = 0 \land len(\text{cons}(x,y)) = 1 + len(y)\]
Community Infrastructure

- **problem libraries**
  SATLib, SMT-LIB, TPTP, ...

- **recurring competitions**
  CASC, HMC, SAT Race, SMT-COMP, ...

- **execution services**
  SMT-EXEC, SystemOnTPTP, termexec, ...

- **standards and utilities**
  DIMACS, EIGER, SMT-LIB, TPTP, ...
Infrastructure Challenges

For solver users:

- What are the available solvers?
- Which solvers work best for my problem?
- Where can I run my experimental evaluations?
Infrastructure Challenges

For solver implementers:

• How can I compare my solver with the state of the art?

• How can I conveniently test my solver on benchmark problems?
Infrastructure Challenges

For community leaders:

• Where can I store my library of benchmark problems?

• How can I run a periodic solver competition?

• How can I build infrastructure for my community?
StarExec: Cross-Community Service and Infrastructure

Main Idea: create single shared infrastructure

• Avoid duplication across communities
• Reduce start-up costs for new communities
• Invest more resources in shared infrastructure
• Create a single destination for solver users

• Bring communities together (LFSC)
StarExec: Cross-Community Service and Infrastructure

Planned functionality

• ~200 processors, web service frontend

• Registered users can upload solvers, benchmarks; run jobs; download results

• Community leaders control community registration, run competitions, host benchmark libraries
Current Status

First Round of **hardware acquisition**

- 32 dual processor quad-core compute nodes
- 3 head nodes for web service requests
- 5 software development nodes
- 2 mirrored network storage units (22TB)
- Offsite back up facility
Current Status

Software development

- Web service (JSP, Javascript, MySQL)
- Job management (Oracle GridEngine)
- Features in progress:
  - Access control, permissions
  - Organization into spaces
  - Public access
  - Monitoring and administration
Acknowledgments

The National Science Foundation: CRI grant

Development team (past and present)
- Benton McCune, Tyler Jensen
- Todd Elvers, Clifton Palmer, Vivek Sardeshmukh, Skylar Stark, Ruoyu Zhang
- JJ Urich, Hugh Brown (sys admin)

Advisory Board
- Daniel Le Berre, Nikolaj Björner, Ewen Denney, Aarti Gupta, Ian Horrocks, Giovambattista Ianni, Johannes Waldmann
Conclusion

**StarExec**: shared logic-solving infrastructure

- Encourage adoption of solvers
- Foster innovation in logic solving
- Bring solver communities together
- Collect benchmarks
- *Increase power for applications*

http://www.starexec.org