StarExec

A Web Service for Evaluating Logic Solvers

Aaron Stump

Geoff Sutcliffe

Cesare Tinelli

Acknowledgments

Support

- The National Science Foundation
- The University of Iowa

Development team (past and present)

- Benton McCune, Tyler Jensen
- Todd Elvers, Clifton Palmerm Vivek Sardeshmukh, Skylar Stark, Ruoyu Zhang
- JJ Urich, Hugh Brown (sys admin)

Background

- Many logic-solving subcommunities
  - ASP, Confluence, CSP, MC, QBF, SAT, SMT, Termination, TP,…

Background

- Many logic-solving subcommunities
- They all benefit from 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, …

Background

- Many logic-solving subcommunities
- They all benefit from infrastructure
- Implementing that infrastructure independently in each case can be wasteful

Challenges

For solver users:

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

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?

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

StarExec: Cross-Community Service and Infrastructure

NSF funded project

- 5 NSF programs involved
- Fall 2011 to fall 2015
- $1.95M total funding
- PIs: Stump, Tinelli (Iowa); Sutcliffe (Miami)
- Hardware hosted at Iowa

StarExec: Cross-Community Service and Infrastructure

Planned functionality

- ~200 processors, web service frontend
- Registered users can upload solvers, benchmarks; run jobs; dowload results
- Community leaders control community registration, run competitions, host benchmark libraries

Current Status

Advisor Committee formed

- Daniel Le Berre (University of Artois)
- Nikolaj Björner (Microsoft Research)
- Ewen Denney (NASA Ames)
- Aarti Gupta (NEC Labs)
- Ian Horrocks (Oxford University)
- Giovambattista Ianni (University of Calabria)
- Johannes Waldmann (Leipzig University)

Current Status

First Round of hardware acquisition

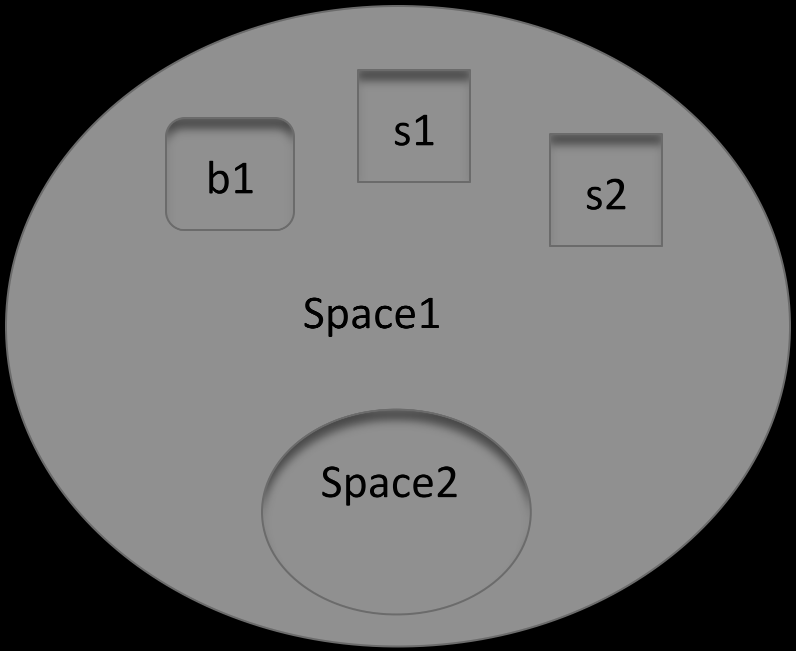
- 32 dual processor quad-core compute nodes
- 3 head nodes for web service requests
- 5 software development nodes
- 2 mirrored network storage units (22TB)
- Offsite back up facility

Primitives

- Benchmarks
- Solvers
- Jobs
- Users

Spaces

- Contain primitives and other spaces



Communities

- Communities are special instances of spaces
- All other spaces are descendants of some community
- New community members automatically get a private space

Permissions

- Add and Remove
- For Spaces and Primitives
- *Space Leaders* may edit the permissions of a space

Community Leadership

- Approve new community members
- Provide benchmark validators and job post processors
- Set community defaults on job settings such as CPU time and post processors

Benchmarks

- Uploaded via a compressed archive
- Can create a space structure mirroring the directory structure
- Benchmarks validated on upload by a community benchmark processor
- Benchmark processor can also provide benchmarks with attributes, a series of key value pairs

Solvers

- Each Solver must be submitted with at least one configuration script
- Configurations tell StarExec how to run the solver
  - e.g.

```
#!/bin/bash

./z3 -smt2 $1
```

Running a Job

- Jobs are initiated from within spaces
- Users may change various settings such as the post processor and the CPU timeout value
- Users may then select the solver/configuration pairs from their space

Running a Job

- Currently, 3 main methods to select the benchmarks you wish to run on
  - Run on all benchmarks in the space hierarchy rooted at your current space
  - Run on all benchmarks in the space
  - Run on selected benchmarks in the space

Running a Job

- Each job pair can be run through one of the communities' post processor to store attributes in the database
- The entire job's output can be downloaded in a compressed archive
- A table of results can be viewed within the web application

System Design

- StarExec runs on a Linux cluster with RedHat 5.8
- Head nodes to send off jobs
- Worker nodes to execute jobs
- 22TB NetApp for general storage
- Node disks for caching

Software Technologies

- Front end implemented with Java Server Pages and Javascript/jQuery
- Backend with Java and MYSQL database
- Apache Tomcat as web server and servlet container
- Oracle Grid Engine to manage the scheduling and the queues

Communities on board

- TPTP
- SMT
- Termination
- ASP
- SAT
- …
- Your community can join too!

Hardware – Control Nodes

- 3 DL380 Gen8 Admin Nodes configured with:
  - 2 Intel E5-2609 2.4 GHZ 4C Processors
  - 128GB RAM
  - 2 HP 600GB 6G SAS 10K 2.5in SC ENT HDD

Hardware – Execution Nodes

- 32 HP SL230 Gen8 nodes each with:
  - 2 Intel E5-2609 2.4GHz 4C Processors
  - 128 GB RAM
  - 1 HP 1TB 6G SATA 7.2k 2.5in SC MDL HDD