Monitoring User-Level Grid Functionality and Performance using

Shava Smallen

ssmallen@sdsc.edu

SC’07 - November 13, 2007
Motivation: Monitoring TeraGrid

- Over 250 TF
- Over 30 PB of online and archival data storage
- Connected via dedicated multi-Gbps links

Monitor the deployments of CTSS (Coordinated TeraGrid Software & Services)
User-level Grid monitoring

- Runs from a standard user account
- Executes using a standard GSI credential
- Uses tests that are developed and configured based on user documentation
- Centrally manages monitoring configuration
- Automates periodic execution of tests
- Verifies user-accessible Grid access points
- Easily updates and maintains monitoring deployment
Additional features of Inca

- Stores and archives a wide variety of monitoring results
- Captures context of monitoring result as it is collected
- Eases the writing, deploying, and sharing of new tests or benchmarks
- Secure
**Reporters** collect monitoring data

- Executable programs that measure some aspect of the system or installed software
- Supports a set of command-line options and writes XML to stdout
- Schema supports multiple types of data
- Extensive library support for perl scripts (most reporters < 30 lines of code)
- Independent of other Inca components
Benefits of *reporter repositories*

- Collection of reporters available via a URL
- Shared across Inca deployments
- Supports package dependencies
- Packages versioned to allow for automatic updates
- Inca project repository contains 167 reporters
  - Version, unit test, performance benchmark reporters
  - Grid middleware and tools, compilers, math libraries, data tools, and viz tool
Agent provides centralized configuration and management

- Implements the configuration specified by Inca administrator
- Stages and launches a reporter manager on each resource
- Sends package and configuration updates
- Manages proxy information
- Administration via GUI interface (incat)

Screenshot of Inca GUI tool, incat, showing the reporters that are available from a local repository
Depot stores and publishes data

- Stores configuration information and monitoring results
- Provides full archiving of reports
- Uses relational database backend via Hibernate
- Supports HQL and predefined queries
- Supports email notifications
- Web services - Query data from depot and return as XML
**Consumer displays data**

- Current and historical views
- Web application packaged with Jetty
- JSP pages/tags to query data and format using XSLT
- CeWolf/JFreeChart to graph data
Current data views

Cumulative test status by resource

Test status by package and resource

Individual test details
**Historical Data Views**

**compute-0-1.local (rbnb ping)**

- Status history of an rbnb ping test

**Error Message Distribution**

- Distribution of error types

---

**Table:**

<table>
<thead>
<tr>
<th>Executed on (report nickname)</th>
<th>Total Reports</th>
<th>Failed Reports</th>
<th>Passed Reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute-0-1.local (rbnb ping)</td>
<td>8496</td>
<td>139</td>
<td>8357</td>
</tr>
</tbody>
</table>

---

**Error Logs:**

- 1 timed out rbnb ping after 60 seconds
Software status and deployments

Current software version: 2.2

(available from Inca website)

http://inca.sdsc.edu
Inca TeraGrid deployment

- Running since 2003
- Testing for CTSS
- Cross-site tests
- GRAM usage
- CA certificate and CRL checking
- Resource registration in MDS
Inca GLEON deployment

- Sensors in lake: dissolved oxygen level, temperature, velocity (some), etc.
- Monitoring Data Turbine deployments since Oct. 24
- Currently deployed for Lake Sunapee and Lake Erken

More about Data Turbine in Paul Hubbard’s talk at 2pm in SDSC booth
Future work

• Add cumulative statistics to better identify problems
• Improve fault tolerance
• Automatically tune test frequency
• Add ability to create custom views

Mock-up of sample statistics
**Benefits of using Inca**

- Detect problems before the users notice them
- Easy to write and share tests and benchmarks
- Easy to deploy and maintain
- Flexible and comprehensive displays
More information

Website:  
http://inca.sdsc.edu

Announcements:  
inca-users@sdsc.edu

Email:  
inca@sdsc.edu

Funded by:

NSF

TeraGrid™
use Inca::Reporter::SimpleUnit;
my $reporter = new Inca::Reporter::SimpleUnit(
  name => 'grid.globus.gramPing',
  version => 2,
  description => 'Checks gatekeeper is accessible from local machine',
  url => 'http://www.globus.org',
  unit_name => 'gramPing'
);
$reporter->addDependency('Inca::Reporter::GridProxy');
$reporter->addArg('host', 'gatekeeper host');
$reporter->processArgv(@ARGV);
my $host = $reporter->argValue('host');
my $out = $reporter->loggedCommand("globusrun -a -r $host", 30);
if (!$out) {
  $reporter->unitFailure("globusrun failed: $!");
} elsif($out !~ /GRAM Authentication test successful/) {
  $reporter->unitFailure("globusrun failed: $out");
} else {
  $reporter->unitSuccess();
}
$reporter->print();
Scheduling and execution

• Reporter manager
  • Manages and schedules the execution of reporters on a single resource
  • Executes under regular user account
  • Monitors reporter system usage and enforces limits
  • Sends monitoring result to a depot