Working with Inca Reporters

Jim Hayes
jhayes@sdsc.edu

Inca 2.0 Workshop
February 23, 2006

Outline

- **Creating Reporters**
- **Using the Perl Reporter Libraries**
- **Using Reporters within Inca**

SDSC SAN DIEGO SUPERCOMPUTER CENTER inca

What is an Inca Reporter?

- **Executable program that measures some aspect of the system or installed software**
- **Requirements:**
  - Supports specific command-line options
  - Writes XML (Inca Reporter schema) to stdout
- **Independent of other Inca components**

Required command-line options

- **-help[=yes|no]**
- **-version[=yes|no]**
- **-log=0|1|2|3|4|error|warn|system|info|debug**
- **-verbose=0|1|2**

Inca Reporter Output

```
<?xml version='1.0'?>
<rep:report xmlns:rep='http://inca.sdsc.edu/dataModel/report_2.1'>
  <gmt>2006-02-07T18:31:08Z</gmt>
  <hostname>jhayes-Computer.local</hostname>
  <name>cluster.compiler.any.unit</name>
  <version>1.6</version>
  <args>
    <arg><name>help</name><value>no</value></arg>
    . . .
  </args>
  <body>
    <unitTest>
      <ID>gcc</ID>
    </unitTest>
  </body>
  <exitStatus><completed>true</completed></exitStatus>
</rep:report>
```


Inca Reporter Output

- <gmt> contents ISO 8601 format
- <body> content must be well-formed; otherwise unrestricted
- <errorMessage> after false <completed>
- <log> after <args> if there are log messages, <help> after <exitStatus> if -help=yes
- <help> used by tools (e.g., incpack)

<help> Subtag

```
<help>
  <ID>help</ID>
  <name>cluster.compiler.any.unit</name>
  <version>1.6</version>
  <description>Tests that a compiler compiles hello world</description>
  <url>http://biokdd.informatics.indiana.edu/…</url>
  <argDescription>
    <ID>verbose</ID>
    <accepted>[012]</accepted>
    <description>verbosity level (0|1|2)</description>
    <default>1</default>
  </argDescription>
  . . .
  <dependency><ID>Inca::Reporter</ID></dependency>
  <dependency><ID>Inca::Reporter::Version</ID></dependency>
</help>
```


Outline

- Creating Reporters
- **Using the Perl Reporter Libraries**
- **Using Reporters within Inca**

Perl Reporter Libraries

- **Simple APIs for performing common reporter tasks**
  - Inca::Reporter general reporter API
  - Inca::Reporter::Version software versions
  - Inca::Reporter::SimpleUnit software unit tests
  - Inca::Reporter::GlobusUnit Globus unit tests
  - Inca::Reporter::Performance system performance metrics
- **Perldoc on-line, http://inca.sdsc.edu/2.0/Reporters**

Inca::Reporter

- **Base class for all types of reporters**
- **Automates determination of hostname, gmt, reporter name, etc.**
- **Handles command-line parsing**
- **Provides interface for log messages**
- **Handles XML generation**

Example: cluster.admin.env

```
use Inca::Reporter;
my $reporter = new Inca::Reporter
  (version => 1.5, description => 'Reports all environment settings');
$reporter->processArgv(@ARGV);
my @tags;
foreach my $line(split(/\n/, `sh -c set`)) {
  my ($var, $value) = $line =~ /(\w+)=(.*)/;
  push(@tags,
    $reporter->xmlElement('var', 0, $reporter->xmlElement('ID', 0, $var),
                                    $reporter->xmlElement('value', 1, $value)));
}
$reporter->setBody
  ($reporter->xmlElement('env', 0, $reporter->xmlElement('ID', 0, 'env'), @tags));
$reporter->setCompleted(1);
$reporter->print();
```


cluster.admin.env <body>

```
<body>
  <env>
    <ID>env</ID>
    <var>
      <ID>HOME</ID>
      <value>/Users/jhayes</value>
    </var>
    <var>
      <ID>JAVA_HOME</ID>
      <value>/Library/Java/Home</value>
    </var>
    <var>
      <ID>PERL5LIB</ID>
      <value>/sw/lib/perl5:/sw/lib/perl5/darwin</value>
    </var>
  </env>
</body>
```

Inca::Reporter::Version

- Common <body> schema for version reporters
- Support for subpackage versions
- Provides convenience methods for common ways of determining version

Example: cluster.interactive_access.openssh.version

```
use Inca::Reporter::Version;
my $reporter = new Inca::Reporter::Version(
  version => 1.8,
  description => 'Reports the version of openssh',
  url => 'http://www.openssh.org',
  package_name => 'openssh'
);
$reporter->processArgv(@ARGV);
$reporter->setVersionByExecutable('ssh -V', 'OpenSSH_([\w\.]+)|GSI ([\w\.]+)');
$reporter->print();
```

cluster.interactive_access.openssh.version <body>

```
<body>
  <package>
    <ID>openssh</ID>
    <version>3.6.1p1</version>
  </package>
</body>
```


Inca::Reporter::SimpleUnit

- **Common <body> schema for unit test reporters**
- **Provides methods for recording results of unit test**

Example: grid.admin.gpt.unit

```
use Inca::Reporter::SimpleUnit();
my $reporter = new Inca::Reporter::SimpleUnit(
  version => 1.4,
  description => 'Checks coherant builds using gpt-verify',
  url => 'http://www.gridpackagingtools.org',
  unit_name => 'gpt-verify'
);
$reporter->processArgv(@ARGV);
my $output = $reporter->loggedCommand('gpt-verify');
if($?) {
  $reporter->unitFailure("call to gpt-verify failed: $output $!");
} else {
  $reporter->unitSuccess();
}
$reporter->print();
```


grid.admin.gpt.unit <body>

```
<body>
  <unitTest>
    <ID>gpt-verify</ID>
  </unitTest>
</body>
```

Inca::Reporter::GlobusUnit

- **Unit tests for Globus components**
- **Provides methods for running Globus jobs**

Example: grid.middleware.globus.unit.date

```
use Inca::Reporter::GlobusUnit;
my $reporter = new Inca::Reporter::GlobusUnit(unit_name => 'globus date');
$reporter->addArg('host', 'hostname where gatekeeper is running');
$reporter->addArg ('timeout', 'kill the job after this many minutes', '60', '\d+');
$reporter->processArgv(@ARGV);
my $host = $reporter->argValue('host');
my $timeout = $reporter->argValue('timeout');
my ($date, $err) = $reporter->submitJob
  (executable => '/bin/date', host => $host, timeout => $timeout, remote => 1);
my $VALID_DATE = '\w{3} \w{3} (\s|\d)\d \d{2}:\d{2}:\d{2}.* \w{3} \d{4}';
if(!defined($date) || $date eq '') {
  $reporter->unitFailure("test failed" . (defined($err) ? ": $err" : ''));
} elsif($date !~ /$VALID_DATE/) {
  $reporter->unitFailure("job completed but result is suspect: $date");
} else {
  $reporter->unitSuccess();
}
$reporter->print();
```

grid.middleware.globus.unit.date <body>

```
<body>
  <unitTest>
    <ID>globus date</ID>
  </unitTest>
</body>
```


Inca::Reporter::Performance

- **Common <body> schema for system/software performance metric reporters**
- **Produces a collection of benchmarks, each a set of parameters (name/value) and statistics (name/value/units)**

Example: grid.benchmark.performance.ping

```
use Inca::Reporter::Performance;
use Inca::Reporter::Performance::Benchmark;
my $reporter = new Inca::Reporter::Performance(test_name => 'ping');
$reporter->addArg('host', 'target host');
$reporter->processArgv(@ARGV);
my $host = $reporter->argValue('host');
if(!open(INPUT, "ping $host|")) {
  $reporter->setResult(0, 'ping not available');
} else {
  my $line = <INPUT>; $line = <INPUT>;
  if($line =~ /time *= *([\d.]+) *(\S*)/) {
    my $benchmark = new Inca::Reporter::Performance::Benchmark();
    $benchmark->setParameter('host', $host);
    $benchmark->setStatistic('round_trip', $1, $2);
    $reporter->addBenchmark('ping', $benchmark);
  }
}
$reporter->print();
```


grid.benchmark.performance.ping <body>

```
<body>
  <performance>
    <ID>ping</ID>
    <benchmark>
      <ID>ping</ID>
      <parameters><parameter>
          <ID>host</ID>
          <value>cuzco.sdsc.edu</value>
        </parameter></parameters>
      <statistics><statistic>
          <ID>round_trip</ID>
          <value>11.3</value>
          <units>ms</units>
        </statistic></statistics>
    </benchmark>
  </performance>
</body>
```

Outline

- Creating Reporters
- Using the Perl Reporter Libraries
- **Using Reporters within Inca**

Reporter Repository

- **Collection of files made available via a URL**
- **Reporters, (Perl) libraries, RPM (planned)**
- **Packages.gz catalog of repository contents contains name:value pairs of package attributes**

Inca Standard Repository

- **http://inca.sdsc.edu/2.0/repository**
- **Perl Reporter libraries and 100 reporters--58 version, 36 unit, 6 general**
- **grasp/network reporters upgraded to v2.0 "soon"**

Creating Packages.gz Catalog

- **incpack [-a yes|no] [-I path …] path …**
- **Creates Packages.gz by running Perl reporters and reading .attrib files**
- **Important attributes name, dependencies, arguments**
- **Attribute file provided by incpack**
- **Optional attributes version, description, url**

*Sample *.attrib*

arguments: help no|yes no; verbose [012] 1; version no|yes no

dependencies: Inca::Reporter;Inca::Reporter::Version

description: Reports the version of gcc

name: cluster.compiler.gcc.version

url: http://gcc.gnu.org

version: 1.5