Working with Inca Reporters

Shava Smallen
ssmallen@sdsc.edu

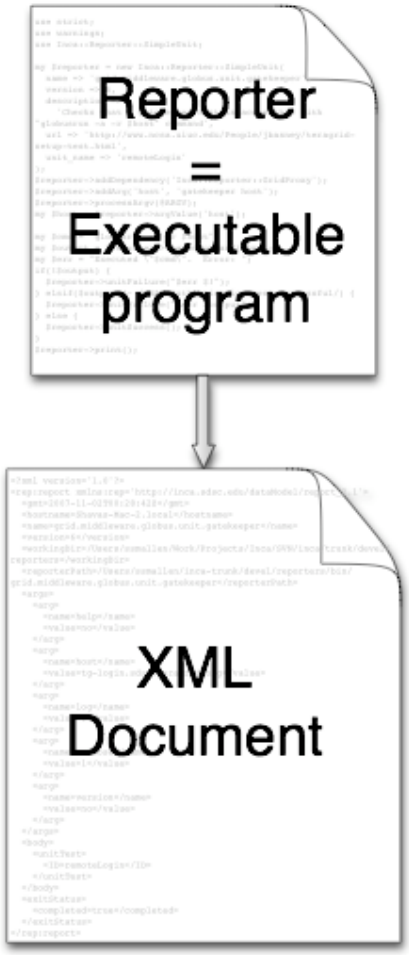
Inca Workshop
August 26, 2010

Outline

- Reporter Definition
- Using the Reporter Libraries
- Reporters in an Inca Deployment
- Meta-Reporters

Reporters Collect Monitoring Data

- A Reporter is an executable program that measures some aspect of the system or installed software
- Requirements:
  - Support four specific command-line options
  - Write XML (Inca Reporter schema) to stdout
- Can be used outside of Inca



Reporters Support Four Options

--help[=yes|no]

If yes, reporter prints help, then exits

--version[=yes|no]

If yes, reporter prints version, then exits

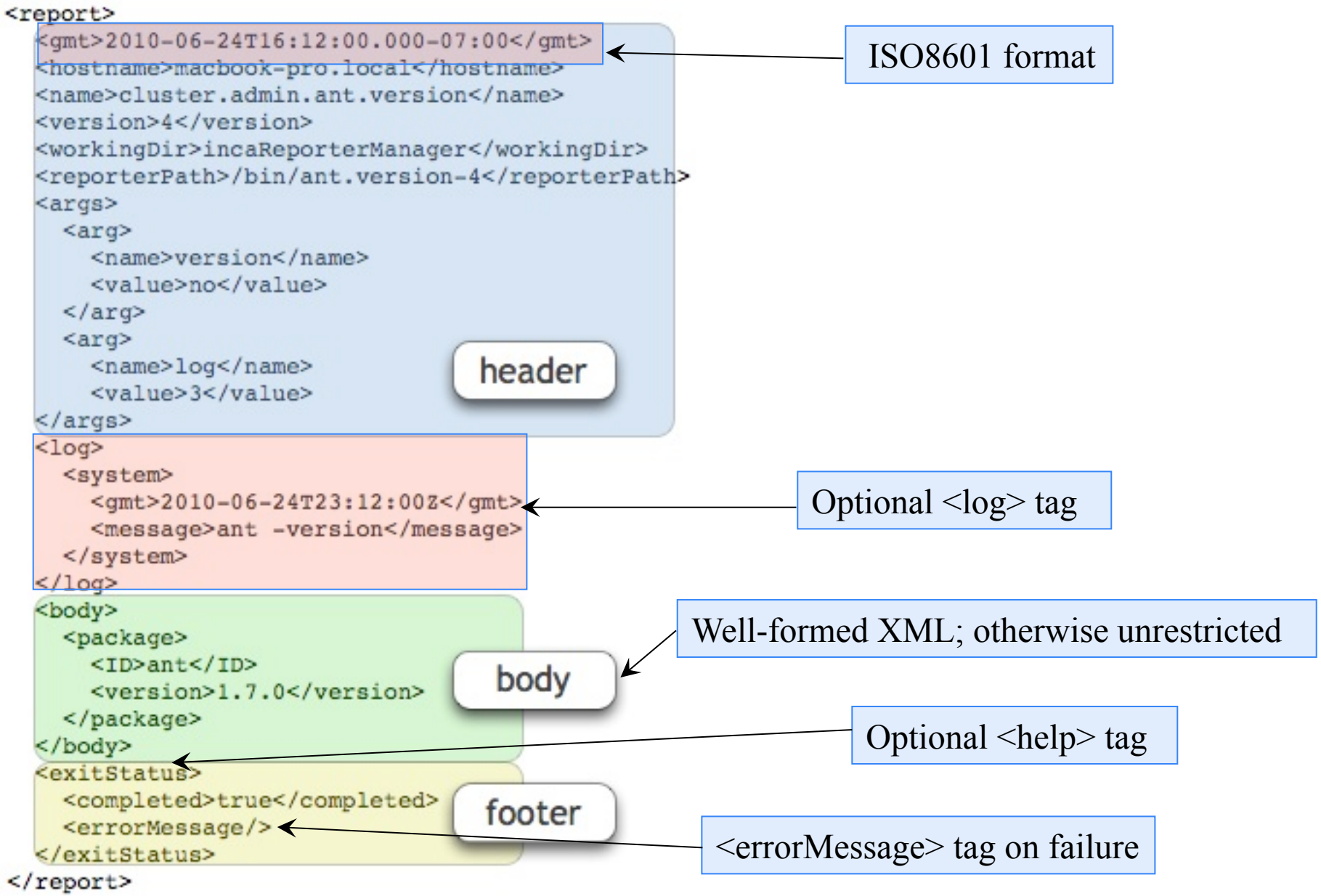
--log=0|1|2|3|4|error|warn|system|info|debug

Selects log messages to include in output

--verbose=0|1|2

Output plain text or Inca Report XML

Reporters Output Inca Report XML



<log> Tag Documents Reporter Execution

```
<log>
  <system>
    <gmt>2008-08-25T21:21:08Z</gmt>
    <message>(gcc src471.c -o src471  &amp;amp;&amp;amp; ./src471)</message>
  </system>
  <system>
    <gmt>2008-08-25T21:21:08Z</gmt>
    <message>/bin/rm -f src471*</message>
  </system>
</log>
```

<help> Tag Describes Reporter

```
<help>
  <ID>help</ID>
  <name>cluster.compiler.any.unit</name>
  <version>2</version>
  <description>Tests that a specified 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::SimpleUnit</ID></dependency>
</help>
```

Outline

- Reporter Definition
- Using the Reporter Libraries
- Reporters in an Inca Deployment
- Meta-Reporters

Libraries Support Common Reporter Tasks

Reporter Purpose	Perl Library	Python Library
General report	Inca::Reporter	inca.Reporter
Software version testing	Inca::Reporter::Version	inca.VersionReporter
Software unit testing	Inca::Reporter::SimpleUnit	inca.SimpleUnitReporter
Globus unit testing	Inca::Reporter::GlobusUnit	inca.GlobusUnitReporter
System performance testing	Inca::Reporter::Performance	inca.PerformanceReporter

Documentation

http://inca.sdsc.edu/releases/latest/repdocs/perl.html

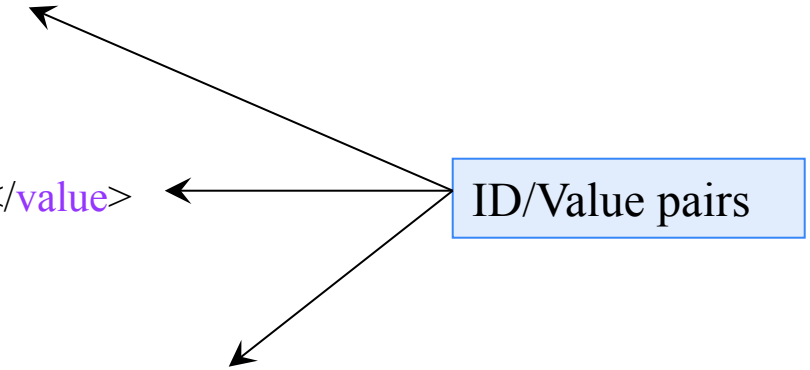
http://inca.sdsc.edu/releases/latest/repdocs/python.html

Reporter Library Implements Base Methods

- Inca::Reporter
  inca.Reporter
- Base class for all reporters
  - Handles command-line parsing
  - Provides interface for log message generation
  - Automates generation of Report header information--hostname, time, reporter name, etc.
  - Supports construction of XML for body
- Directly used only by reporters that gather miscellaneous information--user environment, system CPU, etc.

An Example Base Reporter 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>
```



Perl Base Reporter Code

```
use Inca::Reporter;
my $reporter = new Inca::Reporter
  (version => 3, description => 'Reports all environment settings');
$reporter->processArgv(@ARGV);
my @varXmls;
foreach my $line(split(/\n/, $reporter->loggedCommand('sh -c set'))) {
  my ($var, $value) = $line =~ /(\w+)=(.*)/;
  push(@varXmls,
       $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'), @varXmls));
$reporter->setResult(1);
$reporter->print();
```

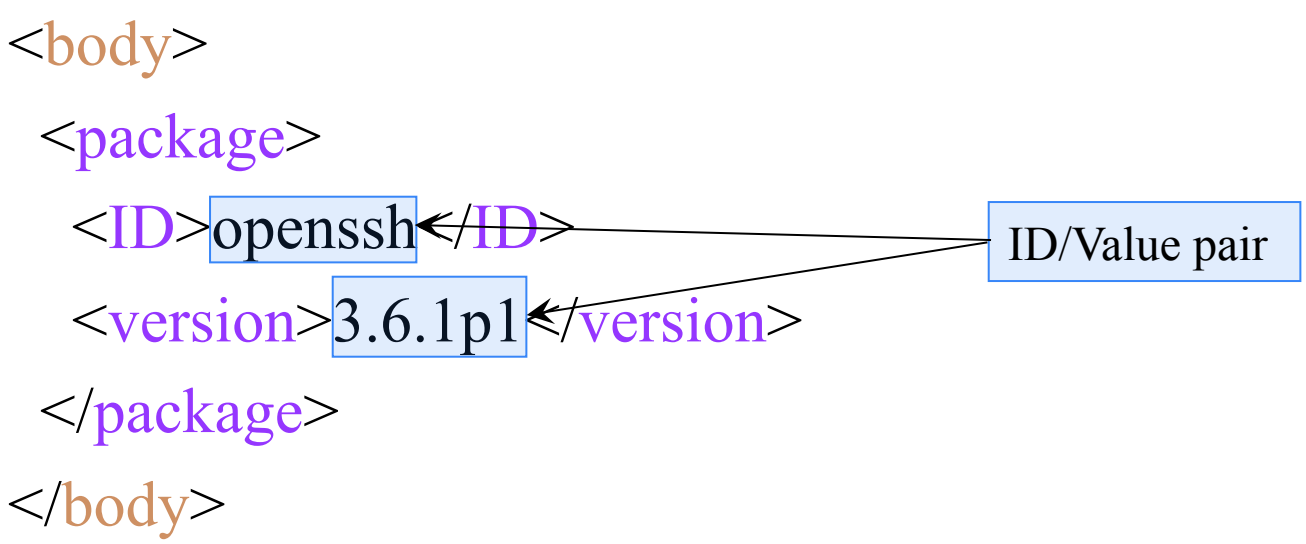
Python Base Reporter Code

```
import re; import sys; from inca.Reporter import Reporter
reporter = Reporter(version=3, description='Reports all environment settings')
reporter.processArgv(sys.argv[1:])
varXmls = []
for line in reporter.loggedCommandOutput('sh -c set').split('\n'):
  parsed = re.match('(\w+)=(.*)', line)
  varXmls.append(
    reporter.xmlElement('var', 0, reporter.xmlElement('ID', 0, parsed.group(1)),
                                  reporter.xmlElement('value', 1, parsed.group(2))))
reporter.setBody(
  reporter.xmlElement('env', 0, reporter.xmlElement('ID', 0, 'env'), *varXmls))
reporter.setResult(1)
reporter.printReport()
```

Version Reporter Library

- Inca::Reporter::Version
  inca.VersionReporter
- Defines common <body> schema for version reporters
- Supports subpackage versions
- Automates common ways of determining version:
  gcc --dumpversion
  grep '#define PVFS_RELEASE_NR': $PVFS_HOME/include/pvfs_config.h
  gpt_query | grep '^condor.*version:'

A Typical Version Reporter Body



Perl Version Reporter Code

```
use Inca::Reporter::Version;
my $reporter =
  new Inca::Reporter::Version(package_name => 'openssh');
$reporter->addArg('ssh', 'ssh command', 'ssh');
$reporter->processArgv(@ARGV);
my $ssh = $reporter->argValue('ssh');
$reporter->setVersionByExecutable
  ("$ssh -V", 'OpenSSH_([\w\.]+)|GSI ([\w\.\-]+)');
$reporter->print();
```

Python Version Reporter Code

```
import sys; from inca.VersionReporter import VersionReporter
reporter = VersionReporter(package_name='openssh')
reporter.addArg('ssh', 'ssh command', 'ssh')
reporter.processArgv(sys.argv[1:])
ssh = reporter.argValue('ssh')
reporter.setVersionByExecutable(
  ssh + ' -V', 'OpenSSH_([\w\.]+)|GSI ([\w\.\-]+)')
reporter.printReport()
```

SimpleUnit Library

- Inca::Reporter::SimpleUnit
  inca.SimpleUnitReporter
- Defines common <body> schema for unit test reporters
- Tests functionality of libraries and applications
- Reporter author determines definition of success
- Provides methods for recording test success or failure

A Typical SimpleUnit Reporter Body

```
<body>
  <unitTest>
    <ID>posixps</ID>
  </unitTest>
</body>
```

Perl SimpleUnit Reporter Code

```
use Inca::Reporter::SimpleUnit;
my $reporter = new Inca::Reporter::SimpleUnit(unit_name => 'posixps');
$reporter->processArgv(@ARGV);
my $output = $reporter->loggedCommand('ps -Af');
if($? == 0) {
  $reporter->unitSuccess();
} else {
  $reporter->unitFailure(output);
}
$reporter->print();
```

Python SimpleUnit Reporter Code

```
import sys; from inca.SimpleUnitReporter import SimpleUnitReporter
reporter = SimpleUnitReporter(unit_name='posixps')
reporter.processArgv(sys.argv[1:])
(status, output) = reporter.loggedCommandStatusOutput('ps -Af')
if status == 0:
  reporter.unitSuccess()
else:
  reporter.unitFailure(output)
reporter.printReport()
```

GlobusUnit Library

- Inca::Reporter::GlobusUnit
  inca.GlobusUnitReporter
- Extends SimpleUnit
- Supports reporters that test Globus components
- Provides methods for execution via Globus--submitJob and submitCSource

Performance Reporter Library

- Inca::Reporter::Performance
  inca.PerformanceReporter
- Defines common <body> schema for system/software performance metric reporters
- Allows recording of a collection of benchmarks, each a set of parameters (name/value) and statistics (name/value/units)

A Typical Performance Reporter 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>roundtrip</ID>
     <value>11.3</value>
     <units>ms</units>
    </statistic></statistics>
  </benchmark>
 </performance>
</body>
```

New in 2.6 – short format option

```
<body>
  <performance ID="ping">
   <benchmark ID="ping">
    <parameters host="cuzco.sdsc.edu"/>
    <statistics roundtrip_ms=”11.3"/>
   </benchmark>
  </performance>
 </body>
```

Perl Performance Reporter Code

```
use Inca::Reporter::Performance;
my $reporter = new Inca::Reporter::Performance(measurement_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 = $reporter->addNewBenchmark('ping');
    $benchmark->setParameter('host', $host);
    $benchmark->setStatistic('round_trip', $1, $2);
    $reporter->setCompleted(1);
  }
}
$reporter->print();
```

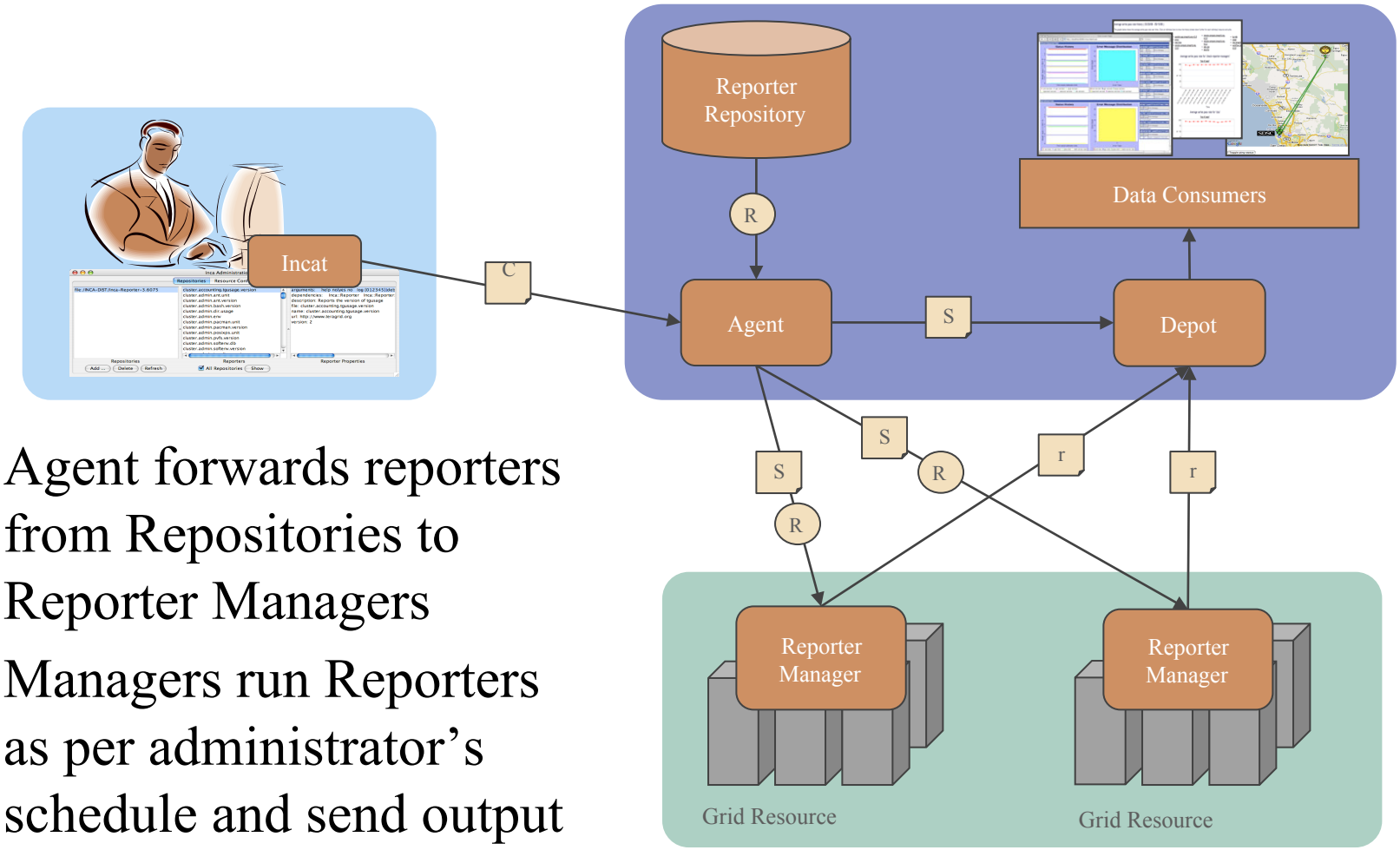
Python Performance Reporter Code

```
import os; import popen2; import re; import sys
from inca.PerformanceReporter import PerformanceReporter
reporter = PerformanceReporter(measurement_name='ping')
reporter.addArg('host', 'target host')
reporter.processArgv(sys.argv[1:])
host = reporter.argValue('host')
try:
  child = popen2.Popen3('ping ' + host)
  line = child.fromchild.readline(); line = child.fromchild.readline()
  parsed = re.search('time *= *([\d.]+) *(\S*)', line)
  benchmark = reporter.addNewBenchmark('ping')
  benchmark.setParameter('host', host)
  benchmark.setStatistic('round_trip', parsed.group(1), parsed.group(2))
  os.kill(child.pid, 9); child.wait()
  reporter.setCompleted(1)
except IOError:
  reporter.setResult(0, 'ping not available')
reporter.printReport()
```

Outline

- Reporter Definition
- Using the Reporter Libraries
- Reporters in an Inca Deployment
- Meta-Reporters

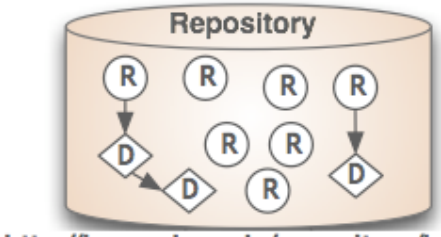
Agent and Managers Work With Reporters



- Agent forwards reporters from Repositories to Reporter Managers
- Managers run Reporters as per administrator's schedule and send output and meta-data to Depot

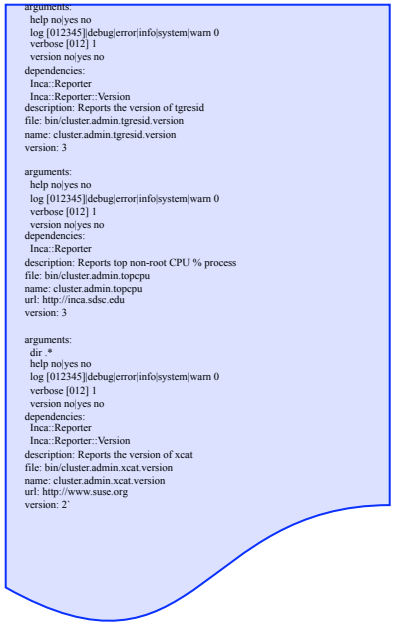
Reporter Repositories Publish Reporters

- Collection of files made available via a URL
- Inca knows how to handle Reporters, Perl/ Python modules, .tar.gz packages
- Inca standard repository (http://inca.sdsc.edu/ repository/latest) contains Reporter libraries and 199 reporters: 96 version, 87 unit, 16 general
- Packages.gz catalog of repository contents allows automated retrieval and update



Packages.gz Lists Reporter Attributes

arguments:
  help no|yes no
  host .*
  log [012345]|debug|error|info|system|warn 0
  verbose [012] 1
  version no|yes no
dependencies:
  Inca::Reporter
  Inca::Reporter::Performance
  Inca::Reporter::Performance::Benchmark
name: grid.benchmark.performance.ping
file: bin/grid.benchmark.performance.ping
description: Reports the ping time to a given host
url: http://inca.ucsd.edu
version: 2



incpack Tool Edits Packages.gz

incpack [-I path …] [-X] path [[-X] path …]

Creates or appends to Packages.gz by running reporters and reading .attrib files

-I -- include library for running reporters

-X -- exclude path (e.g., test directory) from catalog

E.g.,

% incpack –I lib/perl bin/cluster.compiler.gcc.version

Reporters Can Require Other Packages

- Reporter addDependency method notes that a Reporter needs another package to run
- Required package must be in Reporter Repository
- Agent forwards package to Manager, which builds it via, e.g., configure/ make
- Reporter dependency on GridProxy pseudo-package indicates that it requires a credential to run
- Examples:

  $reporter->addDependency('Date::Manip');

  $reporter->addDependency('sampleGridJob');

  $reporter->addDependency('Inca::Reporter::GridProxy');

Multiple package formats supported for dependencies

- Must be a tar.gz
- Installed in $RM_DIST/var/reporter-packages

Standard

Autoconf
configure –prefix=…
make
make install

Make
make INSTALL_DIR=…
make install

Perl

ExtUtils::MakeMaker
perl Makefile.PL …
make
make install

Module::Build
perl Build.PL …
perl Build
perl Build install

Dependencies require an .attrib file

name: Date::Manip
version: 5.54
description: Collection of date manipulation functions
url: http://search.cpan.org/~sbeck/DateManip-5.44
file: DateManip-5.44.tar.gz
dependencies:

- Describes the dependency (tar.gz.attrib)
- E.g.,

  % incpack share/DateManip-5.44.tar.gz

Outline

- Reporter Definition
- Using the Reporter Libraries
- Reporters in an Inca Deployment
- Meta-Reporters

Special Reporter Submits to Batch Queues

- cluster.batch.wrapper reports queue wait time or output of a specified reporter
- Supports cobalt, dql, loadleveler, lsf, pub, sge
- Batch-specific arguments --account, --nodes, --queue, etc.
- Example:

```
cluster.batch.wrapper \
        --scheduler=pbs --account=alf63 \
        --exec='cluster.compiler.gcc.version'
```

Summary Reporters Analyze Series

- New class of reporters to analyze series history and combine series
- Get series data using Inca REST API
- Example: summary.successpct.performance reports collective success % of multiple series

An Example Summary Reporter Body

```
<body>
  <performance xmlns:rep='http://inca.sdsc.edu/dataModel/report_2.1'>
   <ID>successpct</ID>
   <benchmark>
    <ID>successpct</ID>
    <parameters>
     <parameter><ID>filter</ID><value>all2all:gsissh_to_.*</value></parameter>
      …
    </parameters>
    <statistics>
     <statistic><ID>all2all:gsissh_to_bg-login1.sdsc.teragrid.org-fail</ID><value>1</value></statistic>
     <statistic><ID>all2all:gsissh_to_bg-login1.sdsc.teragrid.org-pct</ID><value>94</value></statistic>
     <statistic><ID>all2all:gsissh_to_bg-login1.sdsc.teragrid.org-success</ID><value>18</value></statistic>
     …
    </statistics>
   </benchmark>
  </performance>
</body>
```

Summary

- Flexible schema allows multiple types of data can be reported using Inca reporters
- Perl and Python APIs ease the process of writing reporters
- Reporter and dependencies are published in a repository and automatically deployed by Inca servers
- Meta-reporters allow batch mode execution and summaries

Agenda -- Day 1

9:00 - 10:00	Inca overview
10:00 - 11:00	Working with Inca Reporters
11:15 - 12:00	Hands-on: Reporter API and Repository
1:00 - 2:00	Inca Control Infrastructure
2:00 - 3:00	Administering Inca with incat
3:15 - 4:00	Hands-on: Inca deployment (part 1)

Onto the Tutorial …