# Job Execution Monitoring

Markus Mechtel


4th HEPCG Workshop, Siegen

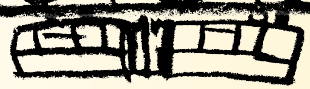
December 13th, 2007

# Outline

# JEM Architecture

- supervisor for jobscripts
- monitoring execution of bash and python scripts
- written in Python
- automatically invoked using a special job submit
- direct data communication via R-GMA, socket (, MonALISA)
- indirect data flow via output sandbox

# Features of JEM



- add-on to existing monitoring architecture
  - acessable command properties:
    script filename, line number, function name, time stamp, command, stdin/stdout/stderr, exit code
  - no need to modify job
- emergency mode
  - activated on error conditions within JEM itself
  - runs job without JEM
  - prevents additional job errors
- little overhead
  - largest overhead was meassured with python scripts
  - small overhead during bash script execution
  - no overhead during binary execution

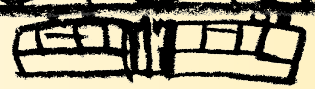
# Example

```
JEM_PROCESS_PYTHON_5127 < EXCEPTION >
    Filename              : /afs/cern.ch/sw/lcg/external/Python/2.4.2/slc4_ia32_gcc345/lib/py
    Line number           : 131
    Function              : __getitem__
    Time stamp            : 2007-09-27 18:47:44
    Error                 : exceptions.IndexError
    Reason                : list index out of range
    Local variables       :
        self : '[('branch', (None, [[('max_repeat', (1, 65535, [('in', [('category', 'catego
        index : '1'

    Code vincinity        :
        127  :           return len(self.data)
        128  :       def __delitem__(self, index):
        129  :           del self.data[index]
        130  :       def __getitem__(self, index):
        131 -->           return self.data[index]
        132  :       def __setitem__(self, index, code):
        133  :           self.data[index] = code
        134  :       def __getslice__(self, start, stop):
        135  :           return SubPattern(self.pattern, self.data[start:stop])

    Traceback             :
          File "/afs/cern.ch/sw/lcg/external/Python/2.4.2/slc4_ia32_gcc345/lib/python2.4/sre_
            return self.data[index]
```

# Benefits of JEM



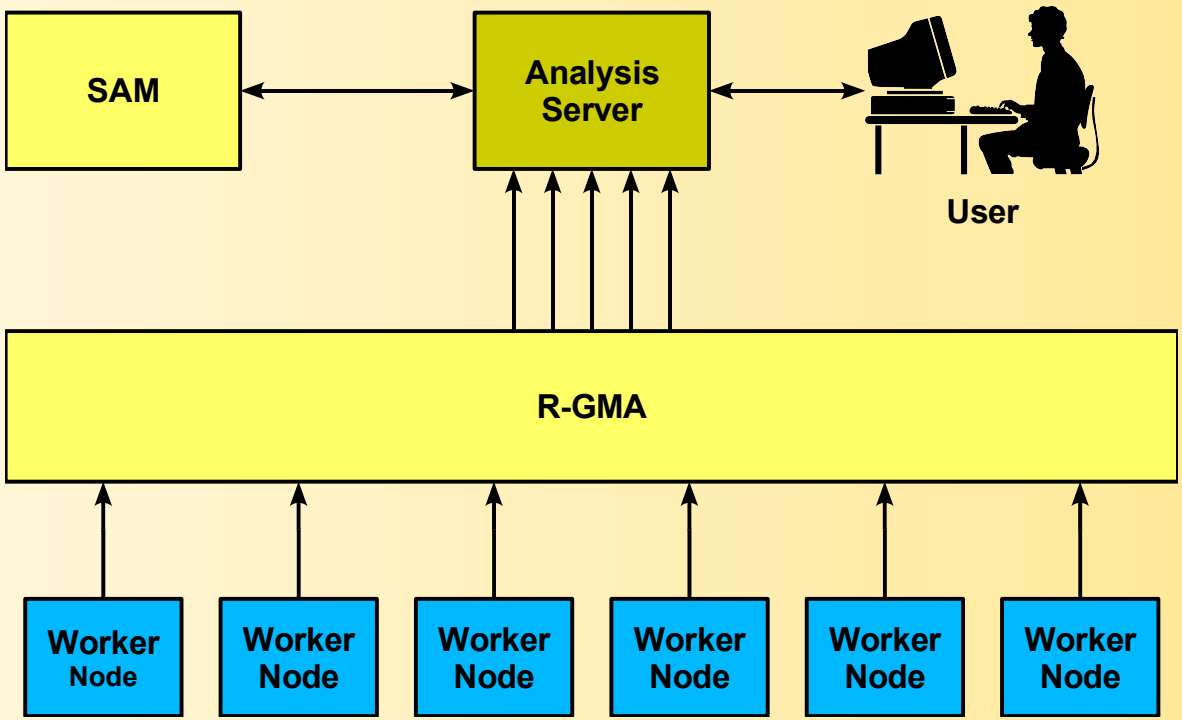
- command logging with different verbosity levels
- values of python variables during errors
- known-critical commands may be modified/hardened
- history of system resources available
- Realtime Information
  - currently executed command
  - access to stdout/stderr and exitcode

# Grid eXPertsystem

- explains error messages
- gives advice on job errors
- makes sugestions for solutions

Frequently asked questions

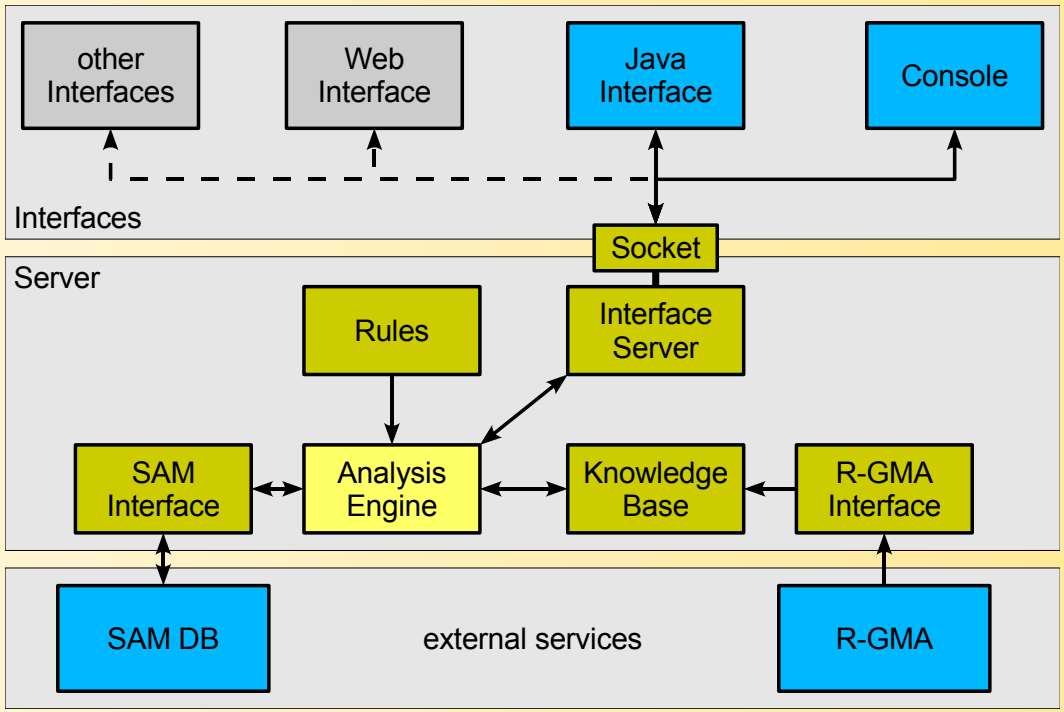
- Who is responsible for the error?
- Which node is affected?
- What is the recommended solution?



- server collects job information from R-GMA
- dynamically queries SAM database for additional information
- matches all information against rules
- provides results to user
- may report error conditions and possible solutions to site admins

many user client interfaces possible

- commandline
- Java GUI
- Web Interface
- ...



# Status

Job Execution Monitor

- JEM version 2 released on November 23$^{\mathrm{rd}}$, 2007
- improved robustness of JEM
- attempt to monitor ATLAS production jobs with JEM
- complete documentation and manual available

GridXP

- ready for release in December/January

# Outlook

## JEM

- create test suite for JEM
- monitor a reasonable amount of real jobs with JEM ($\sim$ 2000 jobs)
- evaluate results of these tests
- integrate JEM with existing tools (e.g. Ganga, gLite, . . . )

## GridXP

- set up server in Wuppertal
- analyse monitoring data from JEM
- enlarge rule base

## Webpage

http://www.grid.uni-wuppertal.de/grid/jem