Using DIANE for astrophysics applications

Ladislav Hluchy, Viet Tran
Institute of Informatics
Slovak Academy of Sciences

SWIFT application

- It is a simulation of the evolutionary scenario of the reservoirs of small bodies in the outer region of the solar system
- The computation consists of a sequence of sub-simulations, each for a defined period of the small-body-orbit evolution
- There are a lot of independent tasks (typically several hundreds) within every sequence
- The necessary requirement is finishing of all the tasks of a given sequence before starting the next sequence

Parametric study

- **SWIFT application is representative for a large class of parametric study applications**
  - Lots of small independent tasks
  - All tasks must be correctly finished
- **gLite built-in support for parametric study is not sufficient:**
  - Probability of job failure is high
  - Long response time (worst case)

Agent-based execution tool

- **Also called as master-worker tool or pilot jobs**
- **Job ≠ task**
  - Job: a grid job, submitted via WMS, runs on CE, has its own job ID, …
  - Task: application computation unit, an execution of application program, reads input data and produces output data
  - A job may repeatedly execute many tasks
- **In agent-based execution tool**
  - Tasks are added to a queue in master process (running in UI)
  - Jobs are worker scripts, submitted to Grid via WMS. When running, worker scripts contact to master process, repeatedly download tasks and execute them, until all tasks are executed

Advantages of agent-based tool

- **Fault tolerance**
  - It is no problem if some worker jobs fail, other workers can take care of task executions
    - Master usually has mechanism for monitoring worker jobs (e.g. heartbeat) and restarting task being executed by failed worker
- **Improving start time**
  - The execution starts when the first worker starts
- **Improving finish time**
  - It is no problem if some worker jobs wait a long time in queue at CEs, other workers can finish all computation even before the delayed jobs start
- **Load balancing**
  - The faster job will execute more tasks, all workers should finish at the nearly same time

Agent-based tool for SWIFT

- **II-SAS shell script**
  - Developed by Jan Astalos at II-SAS, Slovakia
  - Is a set of shell scripts, no additional tools needed
  - Queue are implemented as input directories in SEs, no direct communication between master and worker, no firewall problem
- **DIANE**
  - Developed by CERN
  - Use GANGA for job submission
  - Flexible and portable
- **The next part will focus on our own experiences with using DIANE, not the full guide of DIANE**
  - For full information, look at DIANE webpage http://cern.ch/diane

Installing DIANE

- **Installation is quite easy**
  - Can be installed in user home directory on UI machine, no admin access right needed
  - A simple script for automatic installation DIANE and GANGA
- **Configuration**
  - Setting VO and gLite in GANGA (by user)
  - Opening firewall for omniORB if needed (by admin)

Using DIANE

- **Application need to be written in Python scripting language**
  - Knowing Python will be BIG advantages, and is required for complex applications
  - For advanced application control, GANGA knowledge is required, too
  - For simple apps, it is enough to modify included examples
- **Documentation is still not sufficient**
  - Beta version
  - Developers can give help directly

Using DIANE for SWIFT

- **Use case: 250 small independent tasks in each step, each task has its own input/output**
- **First, put all input files and executable to SE**
  - Using rf* commands (rfmkdir, rfcp)
- **Then, write a simple task script**
  - The script only download input and executable (if not available), run executable, upload output
  - Test the script using globus-job-run
- **Then modify the included example (ExecutableApplication) in DIANE**
  - Just setting executable and arguments in the example, and delete unnecessary commands
- **And run DIANE**

Experiences

- **Users can write task script directly in DIANE**
  - Can exploit all possibilities and flexibility of DIANE
  - Python programming is required
- **or just use standard shell script for tasks**
  - very simple modification of existing Python script from example
  - easy and fast prototype
- **There are many advanced features of DIANE, however, documentations are still missing**
  - Read examples and browse source code of DIANE
  - See demonstrations
  - Contact to developers

Summary

- **Using agent-based execution tools is big benefits for parametric study**
  - Fault tolerance
  - Load balancing
  - Improving start and finish time
- **Using DIANE for SWIFT application is easy and quick**
  - Simple installation
  - Very little coding (writing the task script, modifying included example)
- **Documentation is missing**
  - Can ask help directly from developers
  - Browsing examples and source codes
- **For advanced features, knowledge of Python and GANGA is required**
  - For simple cases like parametric study, it is not needed (but is useful if you can)