CoG Workflow

http://dev.globus.org/wiki/Incubator/CoG_Workflow

Code Name: Karajan

Committers:
Gregor von Laszewski
Mihael Hategan

Contributions from Non-Committers:
Guru Prasad, Deepak Agrawal, A. Anand,

Alumni:
Deepti Kodeboyina

Goals of the Project

- Make the use of Grids through workflows easier
- Workflow Language
  - simple parallelism: coarse-grained tasks, Concurrent sequential processes specification, DAG specification, futures
  - simple structure: conditions; user defined templates, functions, modules, and namespaces; recursion; asynchronous multi-valued functions, ...
  - (simple) integration: Agents/MPI/MPICHG4/OpenMP (possible future direction)
- Workflow Framework
  - scalable lightweight runtime engine for thread handling; convenient client interfaces: GUIs, command line, services
  - GSI enabled workflow service (at this time non WSRF)

gregor@mcs.anl.gov, http://www.mcs.anl.gov/~gregor

Grid Integration

- (cont.) Workflow framework
  - Scheduling: Just-in-time task to resource mapping (bring your own if you do not like ours)
  - Throttles: to avoid resource over-provisioning
- Generic asynchronous Grid abstraction library as part of the original CoG Kit
  - GT2, GT4, PBS, Cobalt, GridFTP, FTP, WebDAV, GSI
  - Connection caching (GridFTP/FTP)

Current Status

- downloadable from
  - http:/www.cogkit.org
- Adoption into Swift
- I would classify us as “hibernated”
  - Higher priorities were given to “functionality” than in the incubation process. This is hopefully changing soon ?

Future

- Performance Improvements
  - Faster byte code through JIT
- Ease of Use Improvements
  - more generic scheduler interface for plugins such as provided by WS-GRAM
  - language
  - Debugging tools
  - Integration of our Python strategy
- Infrastructure
  - Testing
  - Complete the Incubation process (over next 6 month)
  - all of CoG to SVN (???)

gregor@mcs.anl.gov, http://www.mcs.anl.gov/~gregor