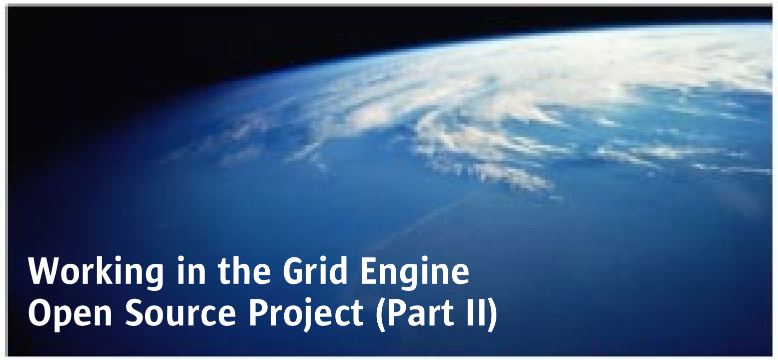


Working in the Grid Engine Open Source Project (Part II)

Andreas Haas

Software Engineering

Sun Grid Engine

Topics

- Data Structures (CULL)
- Interfaces
- Debugging
- Testsuite

Data structures (CULL)

Motivation

- CULL is used by SGE internally for implementing most data structures
- allows reuse of code impl linked lists
- all CULL optimization become effective in the whole system (hashing)
- why not C++? portability benefits from C based solution

Data structures (CULL)

Common basis data types

- Ulong (positive integers, flags)
- Double (big numbers, floats)
- Host (special handling of hostnames)
- String (everything else)
- Sub-list (subordinated list elements)
- Reference (reference associated

Data structures (CULL)

Efficient CULL programming

- low level CULL programming requires some coding effort and is fault-prone
- in many cases high level funcs can be used instead for common operations:
  - Quicker
  - more efficient (hashing)
  - ease of code maintainance

Data structures (CULL)

Efficient CULL programming (samples)

- search list element using a certain key: lGetElemUlong(), lGetElemStr(), ...
- create, initialize list element with key and chain it into list: lAddElemStr(), lAddSubHost(), ...
- Copy certain elements/fields of a list (data base funcs): lSelect(), lWhere(), lWhat(), ...

Data structures (CULL)

Important CULL based SGE data structures

- Job description (JB_Type), job instance (JAT_Type)
- Queue (QU_Type)
- Exechost (EH_Type)
- PE (PE_Type) libs/gdi/sge_peL.h
- See libs/gdi/sge_all_listsL.h for a complete list of all SGE data structures

Data structures (CULL)

Common CULL pitfalls I

- Runtime type errors, e.g. lGetUlong(job, JB_job_name) instead of lGetString(job, JB_job_name)
  - causes an abort()
  - diagnosis information is printed/logged
  - comparable easy to locate

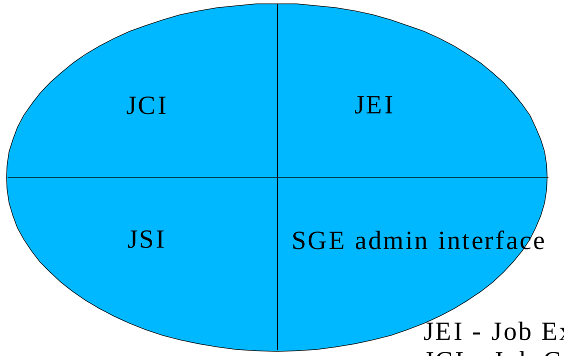
Data structures (CULL)

Common CULL pitfalls II

- Heap corruption errors, e.g due to write/read access to already free()'d CULL elements
  - actually not CULL specific
  - hard to find (runtime debugger)
- Memory leaks
  - use of convenient high level funcs helps avoiding mem leaks
  - hard to find (runtime debugger)

Interfaces

Interface Overview Map



JEI - Job Execution Interfac
JCI - Job Control Interface
JSI - Job Scheduling Interfa

Interfaces

JCI - Job Control Interface (classification)

- Adressee: ISVs, portal integrators
- Covers all high level functionality which is necessary at client side to consign an application as a job to the DRM system
- common operations on jobs like termination or suspension
- There are different JCI occurencies ...

Interfaces

JCI - Job Control Interface (SGE CLI)

- Clients: qsub, qdel, qalter, qmod, qalter, ...
- SGE CLI follows POSIX 1003.2d (standard for batch queuing systems)
- Proven in countless integrations

Interfaces

JCI - Job Control Interface (SGE GDI)

- GDI (Grid Engine Database Interface) is an internal low level interface
- implementation of all CLI clients bases on GDI functionality: sge_gdi()
- But GDI not suited as API b/c too low level and still evolving
- Samples: clients/qsub/qsub.c and

Interfaces

JCI - Job Control Interface (DRMAA JCI)

- DRMAA group has "Work Group" status within Global Grid Forum (GGF)
- Goal is to agree on a stable JCI standard for the prevalent DRM systems
- Current participants: Cadence, Intel, Sun, Veridian (PBS), ...

Interfaces

JCI - Job Control Interface (DRMAA JCI)

- Status: the two API proposals from GGF4 get joined by Hrabri Rajic (Intel)
- Coordination by biweekly conference calls and mailing list
- For details on how to participate see http://www-unix.mcs.anl.gov/~schopf/ggf-sched/WG/drmaa-wg.html

Interfaces

JEI - Job Execution Interface (classification)

- Adressee: integrators, administrators
- Provides all the flexibility necessary to make specific jobs run in a certain environment at each execution host
- allows customization of job start, suspension, termination, preemption operations and of

Interfaces

JEI - Job Execution Interface (SGE JEI)

- prolog/epilog procedure
- Parallel environment procedures start/stop
- Checkpoint interface procedures ckpt/migrate
- Queue terminate/suspend/unsuspend method
- Queue job starter

Interfaces

JEI - Job Execution Interface (SGE JEI)

- further customization of job execution is possible by building a customer specific sge_shepherd(8)
- Usually not needed if job setup enhancement does not requires root priviledges
- Good sample is IRIX project id setting in setosjobid()

Interfaces

JSI - Job Scheduling Interface (classification)

- Adressee: mostly academic sites
- A JSI provides a means for external schedulers to decide which job will be dispatched when and to what compute resource
- Currently there is no JSI API available

Interfaces

JSI - Job Scheduling Interface (SGE sched framework)

- Allows reusing nearly all building blocks of sge_schedd(8)
- Only questionable parts must be implemented
- See schedd link below http://gridengine.sunsource.net/project/gridengine/module_doc.html

Interfaces
JSI - Job Scheduling Interface (Maui/SGE integration)

- Maui: preempting and planning scheduler
- Integration was announced recently (see www.supercluster.org)
- Maui get updates via SGE events
- Passes decisions to SGE via GDI
- Supports only a subset of Maui and SGE features

Interfaces
SGE Admin Interface

- Specific to SGE
- Covers means for administering a SGE system
- Administrator uses qconf CLI or qmon GUI for accomplishing administrative tasks
- qconf/qmon are implemened via GDI

Debugging
Overview

- First make use of regular product diagnosis capabilities (Troubleshooting)
- If necessary use SGE rmon monitoring
- Shepherd debugging specialities
- Commd debugging specialities

Debugging
Troubleshooting I

- SGE builtin diagnosis capabilities should exploited first (customers view)
- See http://gridengine.sunsource.net/project/gridengine/howto/troubleshooting.html
- Troubleshooting covers
  - Why is my job not being dispatched?
  - Why went my job/queue into error state?
  - What logging files are worth to be inspected?

Debugging

Troubleshooting II

- truss/strace/.. tools can be used to investigate problems with applications having problems when run as SGE job

Debugging

SGE rmon monitoring

- It's a developer tool!
- Provides a means to trace into problems with SGE daemons/clients
- Iteratively adding DPRINTF statements usually leads to a good understanding of the problem
- See rmon link below http://gridengine.sunsource.net/project/gridengine/module_doc.html

Debugging

SGE shepherd debugging specialities

- Shepherd writes trace information into 'trace' file
- Admin mail delivers complete 'trace' file for failed jobs
- Adding shepherd_trace() statements plus admin mail usually suffices
- Shepherd can also be started manually in difficult cases

Debugging

SGE commd debugging specialities

- Commd has no 'messages' file (at the moment)
- Creation of /tmp/commd directory causes commd log file being written
- With commdcntl -t 255 already running commd's can be monitored online
- SGE rmon monitoring can also be used

Testsuite
Overview

- Tool for qualitiy assurance (QA) http://gridengine.sunsource.net/project/gridengine/documentation.html see the "testsuite page" link
- Tool for compatibility tests http://gridengine.sunsource.net/project/gridengine/standards.html see the "testsuite page" link
- Bases technically on expect/tcl

Testsuite
Properties I

- Automatic compile and preinstall of SGE (nightly build cron job)
- Test basic product functions (covering install procedures, daemons, client functionality, features)
- Well extensible (ease of creating new tests)

Testsuite
Properties II

- Compatibility test tool (used to estimate compatibility of different product versions or enhancements)
- Saves time through automation (does update sources, compile, install, run tests in one step)

Testsuite
Code structure

- Testsuite framework (used for controlling testsuite system)
- Scripting library (tools library can be used by everyone and is not especially coupled to framework)
- Test template (to speedup creation of new tests)

Andreas Haas

andreas.haas@germany.sun.com