DRMAA: Distributed Resource Management Application API

Hrabri Rajic, Intel
Daniel Templeton, Sun Microsystems
Peter Tröger, University of Potsdam

GGF 12 Tutorial

Brussels, Sept 21, 2004

Agenda

- A few details about available DRMAA implementations
- Detailed example: Povray wrapper application for distributed rendering of raytracing pictures
- Brief example: GT3 job manager for DRMAA-enabled DRM systems

- Note: All code will be available at http://www.drmaa.org

DRMAA-enabled cluster software

- Sun Grid Engine 6.0
  http://gridengine.sunsource.net
- Condor 6.7 series (Red Hat RPM, TGZ)
  http://www.cs.wisc.edu/condor
- Condor 6.7 series (Debian Package)
- http://www.dcl.hpi.uni-potsdam.de/debian
- Announcements for DRMAA implementation
  - GridWay project
    http://asds.dacya.ucm.es/GridWay
  - Rocks Cluster Distribution
    http://www.rocksclusters.org/Rocks/

Availability Matrix (18.9.2004)

	C library	Java library	Perl wrapper
Sun Grid Engine 6.0.0	Complete	Complete	Works
Condor 6.7.1	Partial implementation	Not available	Not working, but needs only SWIG adoption

↙start with SGE now, things will become better during the next months ...

Availability Matrix (21.9.2004)

	C library	Java library	Perl wrapper
Sun Grid Engine 6.0.0	Complete	Complete	Works
Condor 6.7.1	Partial implementation	Not available	Needs SWIG adoption
GridWay	???	Not available	Needs SWIG adoption

↙start with SGE now, things will become better during the next months ...

Condor 6.7.1

- Missing DRMAA functionality
  - Job categories
  - Some mandatory job template attributes
  - Drmaa_run_bulk_jobs()
  - Drmaa_wtermsig()
  - Drmaa_wcoredump()
  - DRMAA_IDS_SESSION_ALL
  - DRMAA_IDS_SESSION_ANY
- A few bugs (we will see them)
- Condor team is working hardly on these issues, should be fixed with the next Condor releases

Sun Grid Engine 6.0.0

- $SGE_ROOT/lib/libdrmaa.so
- $SGE_ROOT/include/drmaa.h
- $SGE_ROOT/examples/drmaa/
- C and Java HowTo documents
- Integrated DRMAA test suite
- No open bugs for the moment

Povray Example

- Popular open source raytracer
  http://www.povray.org
- Command line tool, tons of options, renders scene files
- Several patches for MPI and PVM (farmer-worker)
- PVM options
  - NTnnn = Spawn nnn tasks
  - NAs = Spawn only on architecture s
  - NWnnn = Width of each chunk
  - NHnnn = Height of each chunk
  - NSs = Name of slaves (executable)
  - NDnnn = Name of the working directory
  - pvm_hosts = n,n,.. = List of hostnames to use

- Who is starting the executable on the nodes ?
  - Host file, RSH

DRMAA Povray Goals

- Application which takes the Povray command line arguments and creates DRMS jobs through DRMAA interface
- Slicing of the picture is done by our application, not by patched code inside PVM/MPI-Povray
- Real DRM-independent application
  - SGE: drmaa.h, libdrmaa.so
  - Condor: lib_condor_drmaa.h (Bug), lib_condor_drmaa.a (Open issue)

What do we need ?

- C application with Povray-like command line arguments („dpovray“)
- One way to specify the number of nodes to be used
- An automatic way of splitting up the work into smaller pieces
- A way to concatenate the resulting picture slices
- Cross-DRMS compilation ↙ Configure script
- Let‘s go ...

Command Line Interface

- Povray arguments:
  - +I<input file>
  - +O<output file> (will always be PPM format)
  - +H<height>, +W<width>
- dpovray arguments:
  - +SL<number of nodes to be used>
- Check the number of arguments, read values, ...

- [dpovray1.c]

Start the Rendering

- Get the DRMAA header file
  - #include DRMAA_H
- Calculate number of slices
- Initialize the library
  - drmaa_init(), drmaa_allocate_job_template()
- Fill job template
  - Povray arguments in DRMAA_V_ARGV array
  - Executable name (/usr/bin/env), output path, output files
  - Consider DRM specific options (Condor universe)
- Run jobs and wait for finishing
  - drmaa_run_job(), drmaa_wait(), drmaa_wif...()

- [dpovray2.c]

Combine the Results

- PPM picture format is suitable
  (no compression, fixed header length)
- +FP povray switch in the job template
- Remove the first 3 lines of all output files, concatenate the results
- Save the resulting picture, clean up (drmaa_exit)

- [dpovray3.c]

Build the Software

- Special configure script that searches for well-known DRMAA libraries:
  - AC_ARG_WITH(sge-drmaa)
  - AC_ARG_WITH(condor-drmaa)
  - AC_CHECK_HEADER, AC_CHECK_FILE
  - AC_DEFINE_UNQUOTED(DRMAA_H)
  - LIBS=„$LIBS $drmaa_lib“
  - AC_OUTPUT
- Pluggability issues are in discussion within the DRMAA-WG
- [DEMO]

Conclusion

- Yes, one can write DRMS-independent code (if we ignore the Condor bugs for a moment)
- Further development of the tool:
  - Support enough command line arguments so that Povray GUI frontends can use it
  - Support more Povray file output formats (for example by using gd library for concatenation)
- It is necessary to recompile the application in presence of a new DRMS
  - Solution could be a meta library, DRMAA-WG is investigating this issue

Any questions until here ?

Globus Job Manager Example

- Globus 3.2 supports different job schedulers for local DRM systems (PBS, Condor, ...)
- Realization as Perl module with specific interfaces (see *GRAM job manager tutorial*)
- DRMAA job manager would allow to integrate DRM systems that do not have an explicit scheduler module in Globus distribution
- Unsolved question of resource monitoring (MDS integration)
- Implementation based on DRMAA perl library by *Tim Harsch <harsch1@llnl.gov>* needs compilation for the locally installed DRMS

Necessary Steps

- Write a module constructor
- Implement *submit* method
  - Globus writes submission parameters to $self hash
  - Parse Globus parameters, create DRMAA job template, set DRMAA parameters (at least executable name)
- Implement *poll* method
  - drmaa_job_ps()
- Implement *cancel* method
  - drmaa_control(DRMAA_CONTROL_TERMINATE)
- Install the code in Globus, build a GPT package (not part of this tutorial)

- [DRMAA.in]

GT3 Job Manager Status

- Ongoing work, some issues with SGE
  - Missing status codes
  - Who sets SGE_ROOT ?
- Final GPT package will be available at drmaa.org, watch the mailing list
- GT4 adoption is planned
- For specific questions please contact *Alexander Saar* < *alexander.saar@hpi.uni-potsdam.de*>

Are there any questions ?

Your Homework

- Let your application use DRMS resources through DRMAA interfaces
- Send bug reports to SGE and Condor authors
- Write a DRMAA library for your favorite DRMS (PBS, LLF, XGrid, fork, ...)
- Design the ultimate DRMAA GUI application
- Please, let us know about your experiences

Our Homework

- Continuous inspection of available DRMAA libraries (subscribe to drmaa-wg@gridforum.org for latest news)
- More bindings (WSDL, WSRF,Python, ...)
- More implementations (Mono, XGrid, Rocks, GridWay, GT frontend, ...)
- Collection of DRMAA 1.0 issues as tracker items in GridForge

Thank you for your attention!

drmaa-wg@gridforum.org