



All Hands Meeting: May 20, 2013

History of SURAGRID

Deployment

Steve Johnson
Texas A&M University



Original Deployment

- Each job would send entire R binary to remote resource.
- **globus-url-copy** used to stage in tgz file.
- Untarred into `$OSG_APP/suragrid/`.
- Input file staged in with **globus-url-copy**.
- Job script staged in and executed with **globus-job-submit**.
- Job output retrieved with **globus-url-copy** and/or **globus-job-get-output**.

Original Deployment - Analysis

- Classic Globus deployment.
- Abuses bandwidth and cpu cycles for application transfer.
- Does not scale well unless wrapped by an end user meta-scheduler.



Second Deployment Condor-G



- This handles staging the R script and associated files.
- Allows for scheduling 100's of jobs.
- Depends on end user running a site preparation script before launching jobs. *Spoiler – not anymore!*
- End user must still keep binaries in their home directory – not feasible.

Third Deployment – OSGMM

- OSG MatchMaker used as a generic meta-scheduler.
- Queries OSG ReSS to collect available resources.
- Uses local Condor on submit host.
- This deployment still depends on end user preparing remote site.
- To be effective, SGVO admin would need to keep grid3-locations.txt up to date on all resources – this is anti-social on resources which primarily serve another VO.

Today's R Deployment



- *Status as of May 2013*

- R-2.15.1 installed on:

- TAMU_BRAZOS
- TAMU_Calclab
- TTU_ANTAUEUS
- TTU_TINGE
- FNAL_FERMIGRID
- GridUNESP_CENTRAL
- UTA_SWT2



unesp

- Rebuilt against OpenBLAS w/ runtime instruction set based on detected CPU.
<https://github.com/xianyi/OpenBLAS>
- Installed under `$OSG_APP/suragrid/`.



Courtesy: Google Maps

Today's R Deployment



- Files owned by `sgvoadmin` user where available.
- Otherwise owned by generic `suragrid` user.
- All scripts must source `$OSG_APP/suragrid/etc/profile` on remote site.
 - Defines several `$SURAGRID_*` variables
 - Adds R to `$PATH` and `$LD_LIBRARY_PATH`
 - Adds required libraries to `$LD_LIBRARY_PATH`
 - Defines `$R_LIBS_SITE` and `$R_LIBS_USER` for storage of R packages

Today's R Deployment



- Simplifies R experience for end user.
 - No binaries for user to download or maintain.
 - No need to copy in R binary and required libraries with every job.
 - Faster job startup.
- Adding R packages:
 - Send requests to the R administrators (currently me). Packages will be installed in `$R_LIBS_SITE`.
 - R experts can build packages and keep them in `$R_LIBS_USER`

Today's R Deployment



- Not modifying site's grid3-locations.txt.
- We may want to create an equivalent SURAGRID resource database to store software info.
 - OSG Site Name, Globus Resource Name, Software Name, Version, \$OSG_APP, \$OSG_DATA, Last Update.
- Consider generalizing installation
 - Binaries under \$OSG_APP/suragrid/bin
 - Libraries under \$OSG_APP/suragrid/lib
- Configuration / package management software?



Running R Jobs

- Having prepared sites greatly simplifies the workflow as well as the job scripts.
- **Two Scenarios:**
 - Simple job with single input file, output on stdout
 - Complex job with pre-staged input files, output stored in `$OSG_DATA/suragrid/$LOGNAME`
- **Three Examples:**
 - Globus Commands
 - Condor-G
 - DAGMan with Condor-G

Simple Job: rbench.sh



```
#!/bin/bash
# http://www.math.tamu.edu/osg/R/rbench.sh - This script.
# version 0.3, 03/25/2013
# Script to run R Benchmark on a collection of resources in SURAGrid.
# This uses a case $OSG_APP to load the SURAGrid profile, which defines
# the various paths used by R.
# http://www.math.tamu.edu/osg/R/R-benchmark.25 - Input file.
# Copyright 2013, Steve Johnson, All Rights Reserved.
# License: GPLv3, http://www.gnu.org/licenses/gpl-3.0.html
#
echo "Starting on $OSG_SITE_NAME node `hostname` in dir `pwd` at `date`"

# Source the SURAGrid profile to pull in the R paths.
if [ -f $OSG_APP/suragrid/etc/profile ]; then
  . $OSG_APP/suragrid/etc/profile
else
  echo "WARNING: cannot find OSG_APP/suragrid/etc/profile" 1>&2
fi

# Run it and save exit status in $ret
R --slave --file=R-benchmark-25.R
ret=$?

echo "Finished on $OSG_SITE_NAME node `hostname` in dir `pwd` at `date`"

# Exit with the return code from R.
exit $ret
```

Simple Job, Globus Commands

- Get a proxy.

- `voms-proxy-init -voms suragrid`

- Stage input file, R-benchmark.25.

- `globus-url-copy file:///`pwd`/R-benchmark.25 \`
`gsiftp://ce.somewhere.edu/~R-benchmark.25`

- Stage shell script, rbench.sh.

- `globus-url-copy file:///`pwd`/rbench.sh \`
`gsiftp://ce.somewhere.edu/~rbench.sh`

- `globus-job-run ce.somewhere.edu/jobmanager-fork \`
`/bin/chmod +x rbench.sh`

- Submit the job.

- `globus-job-submit ce.somewhere.edu/jobmanager-pbs rbench.sh`

- [*note the assigned JobID*]



Simple Job, Globus Commands

- Check Job status.

- `globus-job-status JobID`
[*usually PENDING or DONE*]



- Grab stdout, stderr.

- `globus-job-get-output -out JobID`
- `globus-job-get-output -err JobID`

- Remove job, stdout, stderr.

- `globus-job-clean -force JobID`

- Optionally remove script and input.

- `globus-job-run ce.somewhere.edu/jobmanager-fork \`
`/bin/rm R-benchmark.25 rbench.sh`

Simple Job, Condor-G



- Condor-G handles the staging, status, and cleanup.
- Get a proxy.
 - `voms-proxy-init -voms suragrid`
- Submit the job.
 - `condor_submit rbench.condor`
[*note Condor JobID*]
- Get job status.
 - `condor_q [-long] JobID`
- Cancel job if needed.
 - `condor_rm JobID`

Simple Job, Condor-G Script



```
# http://www.math.tamu.edu/osg/R/rbench.condor
Universe                = grid
Notification            = never
Executable              = rbench.sh
Transfer_Executable     = True
Should_Transfer_Files  = YES
When_To_Transfer_Output = ON_EXIT
Transfer_Input_Files    = R-benchmark-25.R
GlobusRSL               = (maxWalltime=60)

# TAMU_Calclab
Grid_Resource           = gt2 calclab-ce.math.tamu.edu/jobmanager-pbs
Log                     = rbench-$(Cluster)-Calclab.log
Output                  = rbench-$(Cluster)-$(Process)-Calclab.out
Error                   = rbench-$(Cluster)-$(Process)-Calclab.err
queue 1

# TAMU_BRAZOS
Grid_Resource           = gt2 hurr.tamu.edu/jobmanager-pbs
Log                     = rbench-$(Cluster)-Brazos.log
Output                  = rbench-$(Cluster)-$(Process)-Brazos.out
Error                   = rbench-$(Cluster)-$(Process)-Brazos.err
queue 1

# TTU_ANTAEUS
Grid_Resource           = gt2 antaeus.hpcc.ttu.edu/jobmanager-sge
Log                     = rbench-$(Cluster)-Antaeus.log
Output                  = rbench-$(Cluster)-$(Process)-Antaeus.out
Error                   = rbench-$(Cluster)-$(Process)-Antaeus.err
queue 1
```

Complex Job, DAGMan Solving The Output Problem



- Condor-G Jobs will start in a “gram_scratch” directory on remote site, which is also used to stage input and output files. This may have insufficient or slow storage. Typically, `$OSG_DATA` points to bigger/faster/stronger storage.
- Do not assume that compute nodes have Internet access to copy files out! This is a bad model not only because it places firewall requirements on the site owner, it also uses the job's wallclock time to stage out files.
- Condor-G is good at moving files into and out of the `gram_scratch` area. Not so good for output files located elsewhere.
- For this example, R functions and input files are statically stored on the remote resource.
- The output file is much larger and will be written by the job to `$OSG_DATA/suragrid/$LOGNAME/`. Need to bring this back to submit host at job completion and delete it from remote site.



Complex Job, DAGMan

Solving the Output Problem



- *DAGMan* is part of Condor and can help with data transfer. <http://research.cs.wisc.edu/condor/dagman/>
- We use *DAGMan* to call a post-processing *retrieve* script for each job to fetch output data.
 - Runs on the submit host to make calls to remote site
 - Uses GRAM and GridFTP (or SRMv2).
 - Discovers \$OSG_DATA directory on remote site.
 - Args: remote site name, remote file name, local file name, number of jobs.
 - `retrieve TAMU_Calclab out.dat Calclab_out.dat 8`

<http://www.math.tamu.edu/osg/R/myRapp.tgz>



myRapp.dag



```
# Submit with: condor_submit_dag myRapp.dag
# JOB marks start of new job and the Condor-G script it will execute.
# Define 3 variables passed to Condor-G script and epilogue script.
JOB          BRAZOS  myRapp.condor
VARS        BRAZOS  globusRes="hurr.tamu.edu/jobmanager-pbs"
VARS        BRAZOS  rsiteName="Brazos"
VARS        BRAZOS  nJobs="8"
SCRIPT POST BRAZOS  retrieve TAMU_BRAZOS sgmyRapp.out sgmyRapp-Brazos.dat 8

# Second job on different site.
JOB          Calclab myRapp.condor
VARS        Calclab globusRes="calclab-ce.math.tamu.edu/jobmanager-pbs"
VARS        Calclab rsiteName="Calclab"
VARS        Calclab nJobs="4"
SCRIPT POST Calclab retrieve TAMU_Calclab sgmyRapp.out sgmyRapp-Calclab.dat 4

# Third job on different site.
JOB          Antaeus myRapp.condor
VARS        Antaeus globusRes="antaeus.hpcc.ttu.edu/jobmanager-sge"
VARS        Antaeus rsiteName="Antaeus"
VARS        Antaeus nJobs="2"
SCRIPT POST Antaeus retrieve TTU_ANTAEUS sgmyRapp.out sgmyRapp-Antaeus.dat 2
```

myRapp.condor



```
# Condor-G script called by myRapp.dag as part of a workflow.
universe                = grid
grid_type               = gt2
executable              = myRapp
transfer_executable    = True
should_transfer_files  = YES
transfer_input_files   = myRapp.r
arguments              = $(Process)
WhenToTransferOutput   = ON_EXIT
notification            = never
GlobusRSL              = (maxWalltime=10)

# globusRes, rsiteName, and nJobs variables are passed in from DAGMan
Grid_Resource          = gt2 $(globusRes)
log                   = myRapp.log
output                = myRapp-$(rsiteName)-$ (Cluster)-$(Process).out
error                 = myRapp-$(rsiteName)-$ (Cluster)-$(Process).err
queue $(nJobs)
```

myRapp



- A bit too long to reproduce here...
- Takes one arg, an integer.
- Checks if new input or R files were sent by Condor-G and copies into place.
- Runs R:
 - `R -slave -file=$APPDIR/myRapp.r -args $myproc`
 - APPDIR = \$HOME/myRapp
 - myproc = argument \$1 passed to myRapp shell script at run time
 - places output in both stdout and \$OSG_DATA/suragrid/\$LOGNAME/sgmyRapp.dat.\$myproc
 - *retrieve* script on submit host fetches the output file.

Complex Job DAGMan



- Submit the .dag file to Condor

- `condor_submit_dag myRapp.dag`

```
-----  
File for submitting this DAG to Condor           : myRapp.dag.condor.sub  
Log of DAGMan debugging messages                : myRapp.dag.dagman.out  
Log of Condor library output                    : myRapp.dag.lib.out  
Log of Condor library error messages            : myRapp.dag.lib.err  
Log of the life of condor_dagman itself         : myRapp.dag.dagman.log
```

```
Submitting job(s).  
1 job(s) submitted to cluster 111.  
-----
```

- Status/Delete with regular Condor commands

- `condor_q [-long] [jobID]`
- `condor_rm jobID`

What's Next for R on SURAgrid?

- Move master binaries and libraries to Well Known Location.
- Document installation process in Wiki.
- Track new releases of R and libraries.
- Get more users online – feedback and improvement.
- Web interface to a simple meta-scheduler.
- Share with Engage VO?



Examples

<http://www.math.tamu.edu/osg/SGAH13ex.tgz>



Q?