Open Science Grid

Workflows: from development to Production
Thursday morning, 10:00 am

Greg Thain <gthain@cs.wisc.edu>

University of Wisconsin - Madison

Overview

- Your DAG runs (once)! Now what?
  - Need to make it run everywhere
  - Need to make it run everytime
  - Need to make it run unattended

Brief note about scientific research

- If others can't reproduce your work, it isn't real research!
  - Work hard to make this happen.
  - 11% of published cancer research reproducible!

The expanding onion

- Laptop (1 machine)
  - You control everything!
- Local cluster (1000 cores)
  - You can ask an admin nicely
- Campus (5000 cores)
  - You better have something for the admin
- OSG (50,000 cores)
  - You don't even know the admins

Making it run everywhere

- What does an OSG machine have?
  - Assume the worst: nothing
- Bring as much as possible with you:
  - Won't that slow me down?
- Bring:
  - Executable
  - Environment
  - Random numbers
  - Tools

Bringing it with you: Matlab

- What's the problem with matlab?
  - Licenses
- What's the solution?
  - "compiling"

How to bring Matlab along

1) Purchase & install matlab “compiler”
2) Run compiler as follows:

```
$ mcc -m -R -singleCompThread -R -nodisplay -R -nojvm –nocache foo.m
```

3) This creates run_foo.sh (et. al.)
4) Create tarball of the runtime

```
$ cd /usr/local/mathworks-R2009bSP1 ; cd ..
$ tar cvzf ~/m.tar.gz matchworks-R2009bSP1
```

More matlab

5) Edit the run_foo.sh

```
tar xzf m.tgz
mkdir cache
chmod 0777 cache
export MCR_CACHE_ROOT=`pwd`/cache
```

Make a submit file:

universe = vanilla
executable = run_foo.sh
arguments = ./mathworks-R2009bSP1
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = m.tgz, foo
queue

Final notes on matlab

- Cache the runtime for extra credit
- Other interpreters similar (R, Python, etc)

But I can't make it work everywhere

- Using
  - Request_memory
  - Request_disk
  - Request_cpus
- GLIDEIN whitelist/blacklist if a site is somehow bad.
  - But talk to the GOC first

Improved storage

- Listen to Derek's methods:
  - Sandbox
  - Caching
  - Prestaging
  - SE horsepower

Making it work everytime

- What could possibly go wrong?
  - Eviction
  - File corruption
  - Performance suprises
    - Network
    - Paging
    - Disk
    - …
  - Maybe even a bug in your code

Performance Surprises

- One bad node can ruin your whole day
- "Black Hole" machines
  - Avoidance tricks and their autoclustering costs
- Using PERIODIC_HOLD / RELEASE
  - To avoid ill performing jobs/machines

File Corruption

- If you don't check, it will happen…
  - ETL
- Running sha1 yourself on both sides
- DAG PRE and POST scripts
  - "Trust, but verify"
- Example here

What to do if a check fails

- Understand something about failure
- Use DAG RETRY
- Let rescue dag continue…
  - Workflow specific

Open Science Grid

Running unattended

- This is the ultimate goal!
- Need to automate:
  - Data collection
  - Data cleansing
  - Submission (condor cron)
  - Analysis and verification
  - LaTex and paper submission ☺

Open Science Grid

If this were a test...

- 20 points for finishing at all
- 10 points for the right answer
- 1 point for every error condition checked

Questions?

- Questions? Comments?
  - Feel free to ask me questions later: