Handling Data on OSG

Wednesday, August 9
Mats Rynge

This work was supported by NSF grants MPS-1148698, OAC-1836650, and OAC-2030508
You are here

Your job is here

You are here
You are here

Your job is here
From yesterday…

container_image = py-cowsay.sif
From yesterday…

container_image = py-cowsay.sif

queue 10000
Like all things

- I always think of HTC/OSG usage as a spectrum:

More Resources, More Planning

Laptop  Cluster  OSG
Handling Data on OSG

• Overview / Things to Consider
• HTCondor File Transfer
• OSDF
• Shared File Systems
What is big data?

- In reality, “big data” is relative
  - What is ‘big’ for you? Why?
What is big large data?

- In reality, “big data” is relative
  - What is ‘big’ for you? Why?

- Volume, velocity, variety!
  - think: a million 1-KB files, versus one 1-TB file
Determining In-Job Needs

- "Input" includes any files needed for the job to run:
  - executable
  - transfer_input_files
  - data and software

- "Output" includes any files produced for the job that need to come back:
  - output, error
Data Management Tips

1. Determine your per-job needs
   a. minimize per-job data needs

2. Determine your batch needs

3. Leverage HTCondor and OSG data handling features!
First! Try to minimize your data

- split large input for better throughput
- eliminate unnecessary data
- file compression and consolidation
  - job input: prior to job submission
  - job output: prior to end of job
  - moving data between your laptop and the submit server
**‘Large’ data: The collaborator analogy**

What method would you use to send data to a collaborator?

<table>
<thead>
<tr>
<th>amount</th>
<th>method of delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>email body</td>
</tr>
<tr>
<td>tiny – 100MB</td>
<td>email attachment (managed transfer)</td>
</tr>
<tr>
<td>100MB – GBs</td>
<td>download from Google Drive, Drop/Box, other web-accessible repository</td>
</tr>
<tr>
<td>TBs</td>
<td>ship an external drive (local copy needed)</td>
</tr>
</tbody>
</table>

*Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.*

Andrew S. Tanenbaum (1981) – Professor Emeritus, Vrije Universiteit Amsterdam
# Large *input* in HTC and OSG

## File Size Method of Delivery

<table>
<thead>
<tr>
<th>File Size</th>
<th>Method of Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>within executable or arguments?</td>
</tr>
<tr>
<td>Tiny – 1GB per file</td>
<td>HTCondor file transfer (up to 1GB total per job)</td>
</tr>
<tr>
<td>1GB – 20GB</td>
<td>OSDF (regional replication)</td>
</tr>
<tr>
<td>20 GB – TBs</td>
<td>shared file system (local copy, local execute servers)</td>
</tr>
</tbody>
</table>
Transfers

More Data

HTCondor
File Transfer

OSDF

Local
Storage
Rule of thumb - many dimensions

Number of jobs

Input Size
Rule of thumb - many dimensions

- Should this be HTCondor file transfer, OSDF, or shared filesystem?
Rule of thumb - many dimensions

- Should this be HTCondor file transfer, OSDF, or shared filesystem?

Number of jobs vs. Job length vs. Input Size
Handling Data on OSG

- Overview / Things to Consider
- HTCondor File Transfer
- OSDF
- Shared File Systems and Other Options
Review: HTCondor Data Handling

HTCondor

submit
server

submit file
executable
dir/ input
output

exec
server

(exec dir)/
executable
input
output
Network bottleneck: the submit server

HTCondor

submit server
- submit file
- executable
- dir/ input
- output

exec server
- (exec dir)/
- executable
- input
- output
Network bottleneck: the submit server

Input transfers for many jobs will coincide

submit server
submit file
executable
dir/ input
output

HTCondor

exec server
(exec dir)/
executable
input
output
Network bottleneck: the submit server

Input transfers for many jobs will coincide

Output transfers are staggered

submit file
executable
dir/ input
output

exec server
(exec dir)/
executable
input
output
Hardware transfer limits

HTCondor

1GB total

submit server

submit file
executable
dir/ input
output

exec server

(exec dir)/
executable
input
output

1GB total
Handling Data on OSG

- Overview / Things to Consider
- HTCondor File Transfer
- OSDF
- Shared File Systems
### Large input in HTC and OSG

<table>
<thead>
<tr>
<th>file size</th>
<th>method of delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>within executable or arguments?</td>
</tr>
<tr>
<td>tiny – 100MB per file</td>
<td>HTCondor file transfer (up to 1GB total per-job)</td>
</tr>
<tr>
<td>100MB – 1GB, shared</td>
<td>download from web server (local caching)</td>
</tr>
<tr>
<td>1GB – 20GB, unique or shared</td>
<td>OSDF (regional replication)</td>
</tr>
<tr>
<td>10 GB - TBs</td>
<td>shared file system (local copy, local execute servers)</td>
</tr>
</tbody>
</table>
Open Science Data Federation (OSDF)
OSDF Considerations

- Available at ~95% of OSG sites
- Regional caches on very fast networks
  - Recommended max file size: 20 GB
- Can copy multiple files totaling >10GB
- Change name when update files
Placing Files in OSDF

• Place files in `/ospool/PROTECTED/username/`

```
local server
/ospool/PROTECTED/username/
```

```
“OSDF” origin
file
```

```
regional cache
```

```
any OSG submit
```

```
exec server
```
Obtaining Files in OSDF

- Use HTCondor transfer for other files

```
local server
```

```
/ospool/PROTECTED/username/
```

```
any OSG submit
```

```
“OSDF” origin
```

```
file
```

```
regional cache
```

```
file
```

```
HTCondor
```

```
exec server
```
Obtaining Files in Stash

- Download using stashcp command
Open Science Data Federation (OSDF)

Your job is here

Institution Cache Site

I2/Backbone Cache Site
Open Science Data Federation (OSDF)
Open Science Data Federation (OSDF)
Open Science Data Federation (OSDF)

You are here

Data

Job 2

Institution Cache Site

I2/Backbone Cache Site
Open Science Data Federation (OSDF)
In the Submit File

```
transfer_input_files = osdf:///ospool/PROTECTED/USERNAME/...
```
How about output?
## Output for HTC and OSG

### Executable Method of Delivery

<table>
<thead>
<tr>
<th>Amount</th>
<th>Method of Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>within executable or arguments?</td>
</tr>
<tr>
<td>tiny – 1GB, total</td>
<td>HTCondor file transfer</td>
</tr>
<tr>
<td>1GB - 20GB, unique or shared</td>
<td>OSDF</td>
</tr>
<tr>
<td>20GB+, total</td>
<td>shared file system (local copy, local execute servers)</td>
</tr>
</tbody>
</table>
Output for HTC and OSG

<table>
<thead>
<tr>
<th>amount</th>
<th>method of delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>within executable or arguments?</td>
</tr>
<tr>
<td>tiny – <strong>1GB, total</strong></td>
<td>HTCondor file transfer</td>
</tr>
<tr>
<td>1GB – 20GB, unique or shared</td>
<td>OSDF</td>
</tr>
<tr>
<td>20GB+, total</td>
<td>shared file system (local copy, local execute servers)</td>
</tr>
</tbody>
</table>
Writing to stash

```python
transfer_output_remaps = "Output.txt = osdf://ospool/PROTECTED/<username>/Output.txt"
```
Other Considerations

• Only use these options if you MUST!!
  – Each comes with limitations on site accessibility and/or job performance, and extra data management concerns

<table>
<thead>
<tr>
<th>file size</th>
<th>method of delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>within executable or arguments?</td>
</tr>
<tr>
<td>tiny – 10MB per file</td>
<td>HTCondor file transfer (up to 1GB total per-job)</td>
</tr>
<tr>
<td>10MB – 1GB, shared</td>
<td>download from web server (local caching)</td>
</tr>
<tr>
<td>1GB - 10GB, unique or shared</td>
<td>OSDF (regional replication)</td>
</tr>
<tr>
<td>10 GB - TBs</td>
<td>shared file system (local copy, local execute servers)</td>
</tr>
</tbody>
</table>
Cleaning Up Old Data

Make sure to delete data when you no longer need it in the origin!!!

Servers do NOT have unlimited space!
Some may regularly clean old data for you. Check with local support.
# Quick Reference

<table>
<thead>
<tr>
<th>Option</th>
<th>Input or Output?</th>
<th>File size limits</th>
<th>Placing files</th>
<th>In-job file movement</th>
<th>Accessibility?</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTCondor file transfer</td>
<td>Both</td>
<td>100 MB/file (in), 1 GB/file (out); 1 GB/tot (either)</td>
<td>via HTCondor submit node</td>
<td>via HTCondor submit file</td>
<td>anywhere HTCondor jobs can run</td>
</tr>
<tr>
<td>OSDF</td>
<td>Both</td>
<td>20 GB/file</td>
<td>via HTCondor submit server</td>
<td>transfer_*_file / stashcp command</td>
<td>OSG-wide (most sites), by anyone</td>
</tr>
<tr>
<td>Shared filesystem</td>
<td>Input, likely output</td>
<td>TBs (may vary)</td>
<td>via mount location (may vary)</td>
<td>use directly, or copy into/out of execute dir</td>
<td>local cluster, only by YOU (usually)</td>
</tr>
</tbody>
</table>
Acknowledgments

- This work was supported by NSF grants OAC-1836650, and OAC-2030508
Additional Slides

Shared Filesystem Details
(Local) Shared Filesystems

- data stored on file servers, but network-mounted to local submit and execute servers
- use local user accounts for file permissions
  - Jobs run as YOU!
  - readable (input) and writable (output, most of the time)
- \textit{MOST} perform better with fewer large files (versus many small files of typical HTC)
Shared FS Technologies

• *via network mount*
  - NFS
  - AFS
  - Lustre
  - `/staging` (may use NFS mount)
  - Isilon (may use NSF mount)

• *distributed file systems (data on many exec servers)*
  - HDFS (Hadoop)
  - CEPH
Shared FS Configurations

1. Submit directories \textit{WITHIN} the shared filesystem
   - most campus clusters
   - limits HTC capabilities!!

2. Shared filesystem separate from local submission directories
   - supplement local HTC systems
   - treated more as a repository for VERY large data (>GBs)

3. Read-only (input-only) shared filesystem
   - Treated as a repository for VERY large input, only
Submit dir within shared FS

Shared FS

(submit dir)/
  file.sub
  input, software
  log, error, output
Submit dir within shared FS

Shared FS
(submit dir)/
file.sub
input, software
log, error, output

# file.sub:
should_transfer_files = NO
transfer_input_files =

execute on server...
Separate shared FS

Submit file
Executable
dir/ input
output

(exec dir)/
Executable
input
output

Separate FS
Separate shared FS - Input

1. Place compressed input into FS

Separate FS
/path/to/lgfile

HTCondor

submit server

server server server server server (exec dir)/
Separate shared FS - Input

2. Executable copies and decompresses the file
Separate shared FS - Input

3. Executable must remove the file in the exec dir after use
Separate shared FS - Output

1. Executable creates and compresses the output file
Separate shared FS - Output

2. Executable copies the file

Separate FS
/path/to/

submit server

HTCondor

exec server

(exec dir)/

Igfile

Igfile

Igfile
Separate shared FS - Output

1. Submit server to HTCondor
2. HTCondor assigns task to exec server
3. Executable removes the file in the exec dir

Separate FS
/path/to/lgfile
At UW-Madison (Ex. 3.1-3.2)

learn.chtc.wisc.edu

submit

server

HTCondor

Separate FS

/mnt/gluster/user/

lgfile

exec

server

server

server

server

(exec dir)/