Moving Data on the OSPool

Wednesday, June 25

Showmic Islam

Slides adapted from Mats Rynge, Andrew Owen
This work was supported by NSF grants MPS-1148698, OAC-1836650, and OAC-2030508

Outline

- What is ~~big~~ large data?
- Data Management Tips
- Characteristics of OSPool
- Solutions to moving data
  - HTCondor File Transfer
  - OSDF/Pelican

What is ~~big~~ large 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*** <u>software</u>

- “**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 OSPool 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

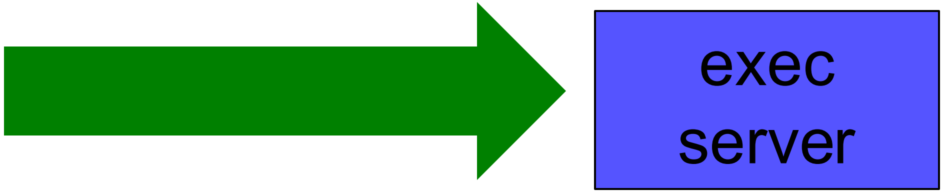
What method would you use to send data to a collaborator?

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

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

Andrew S. Tanenbaum (1981) – Professor Emeritus, Vrije Universiteit Amsterdam

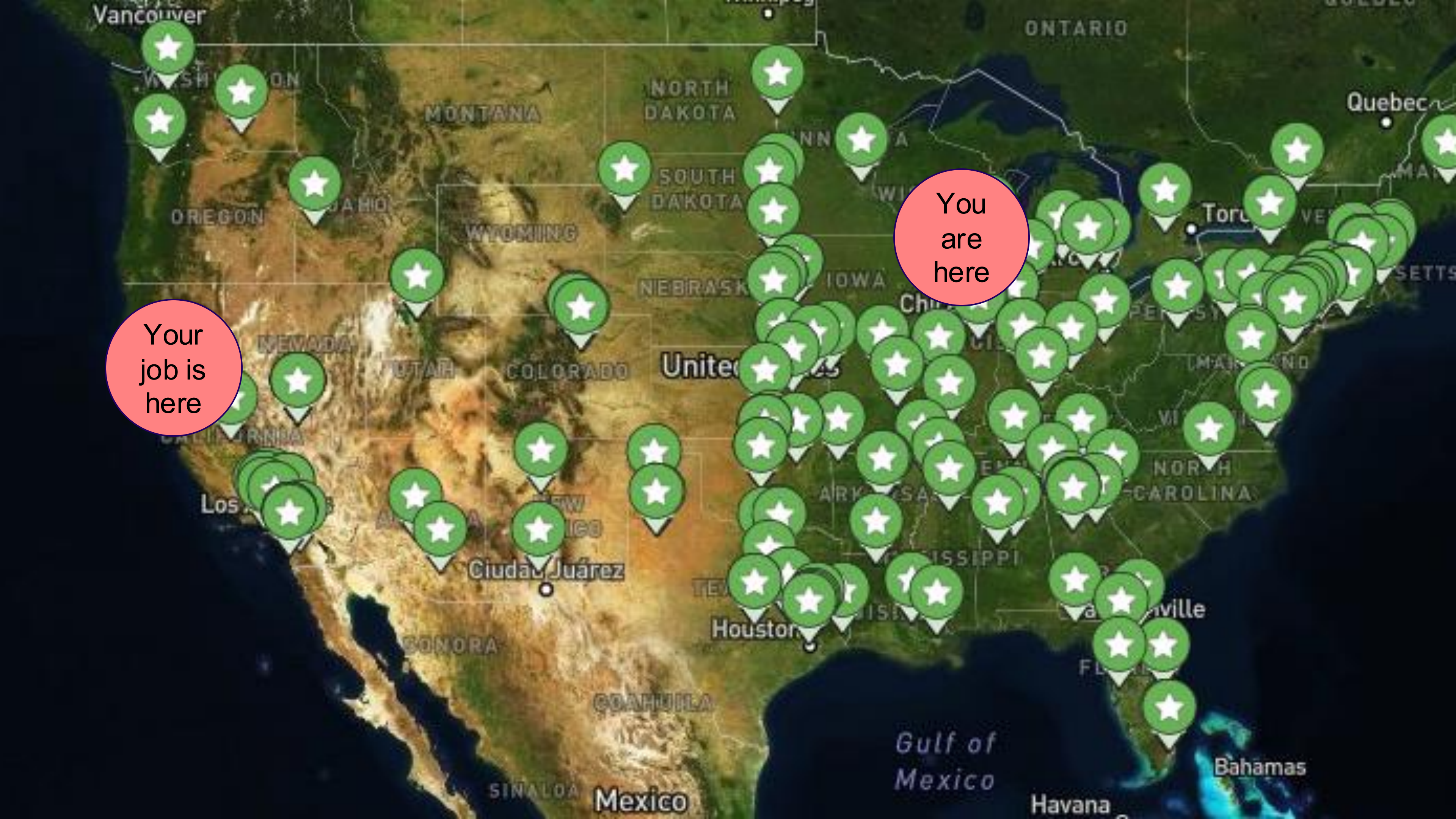
Large *input* in HTC and OSPool



file size	method of delivery
words	within executable or arguments?
tiny – 1GB per file	HTCondor file transfer (up to 1GB total per job)
1GB – 20GB	OSDF (regional replication)
20 GB – TBs	shared file system (local copy, local execute servers)

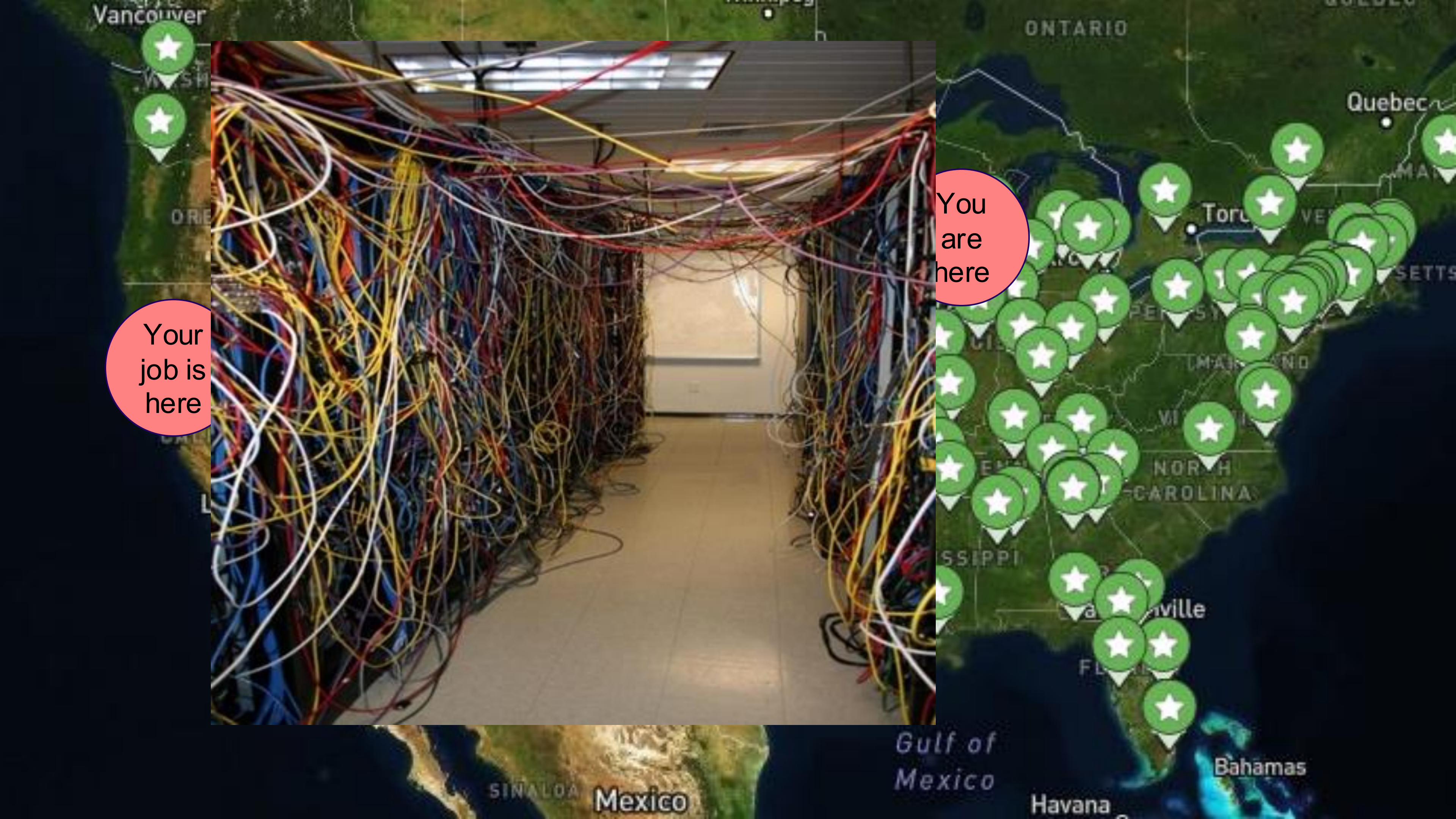
OSPool Characteristics

- No Shared FS (File System)
- Execute Point does not have access to data on the Access Point



Your job is here

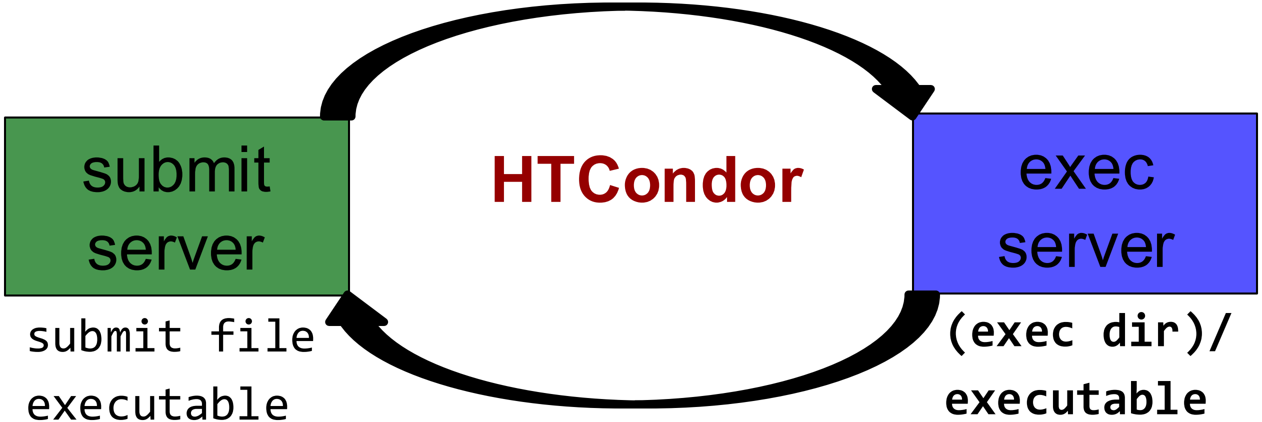
You are here



Your job is here

You are here

Review: HTCondor Data Handling



submit server

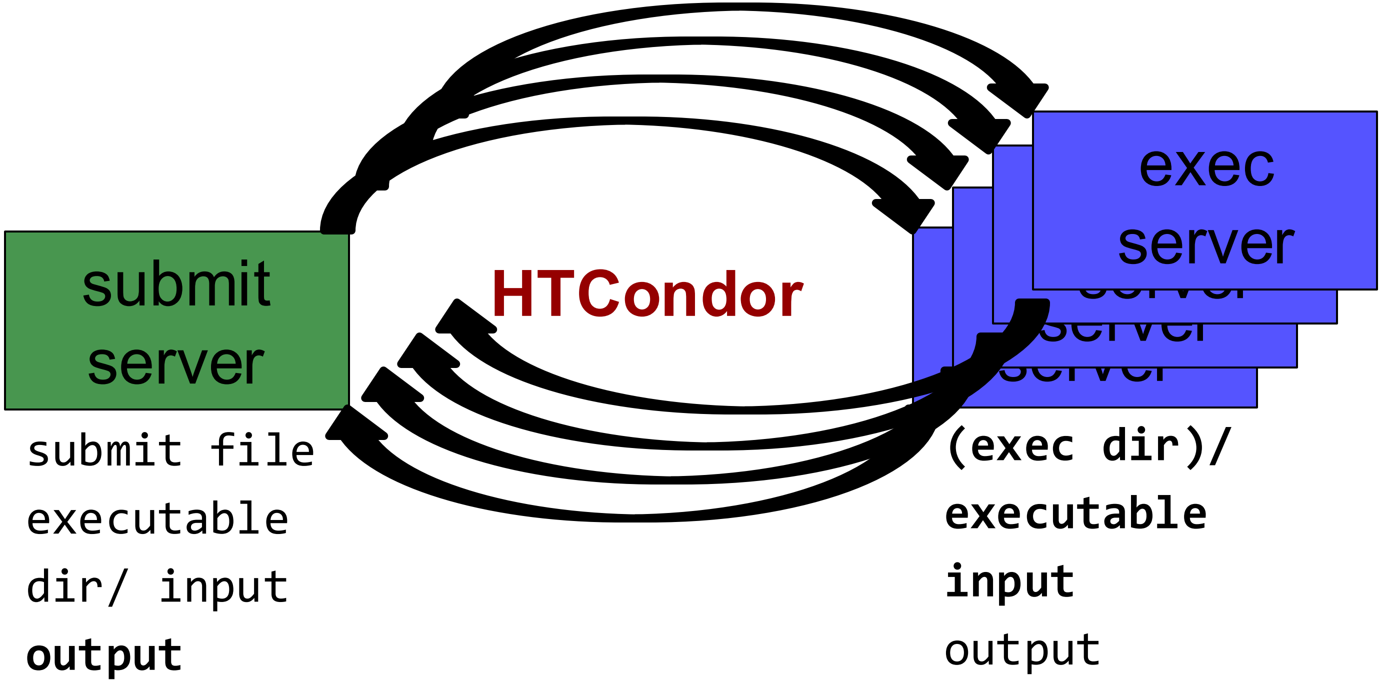
HTCondor

exec server

submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Network bottleneck: the submit server

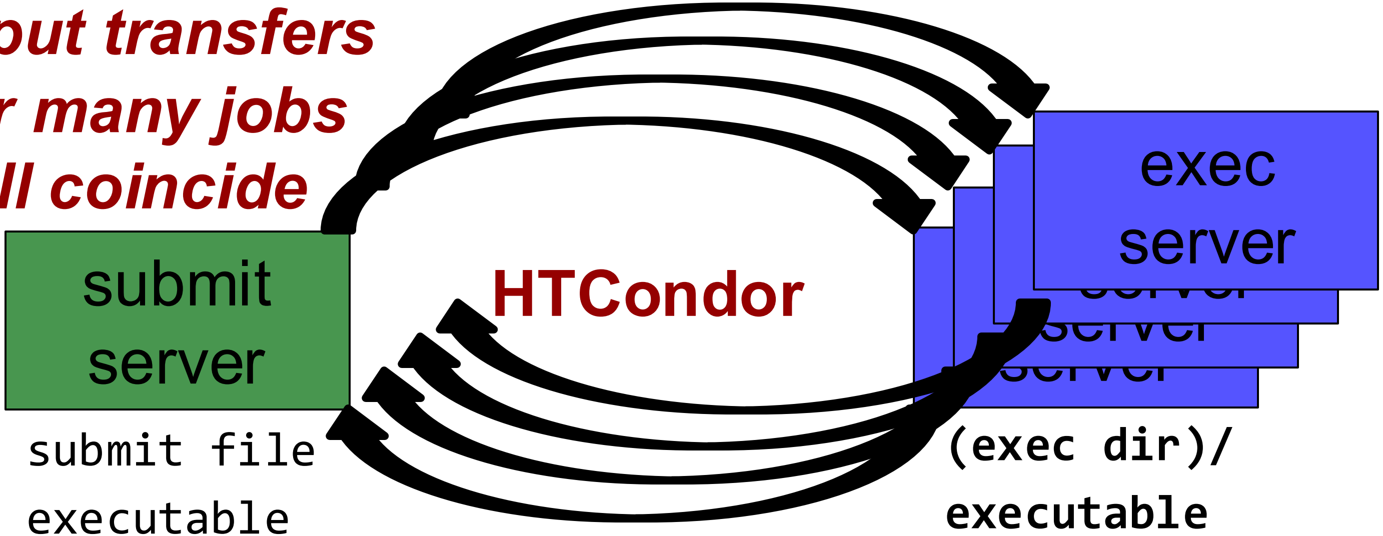


submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Network bottleneck: the submit server

Input transfers for many jobs will coincide

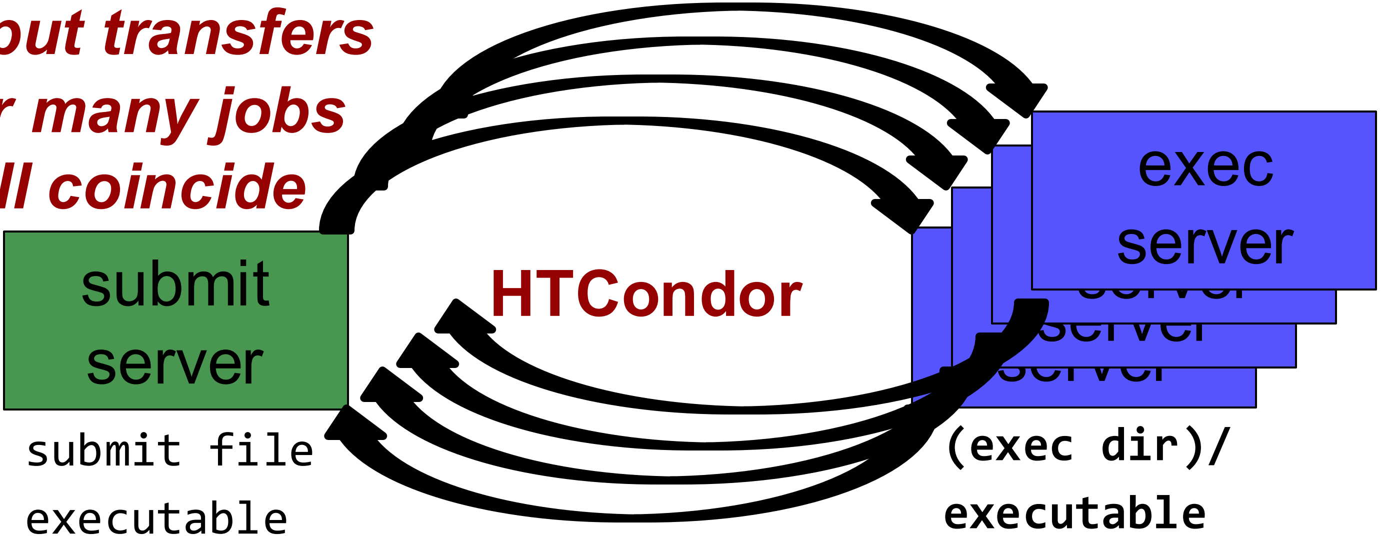


submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Network bottleneck: the submit server

Input transfers for many jobs will coincide

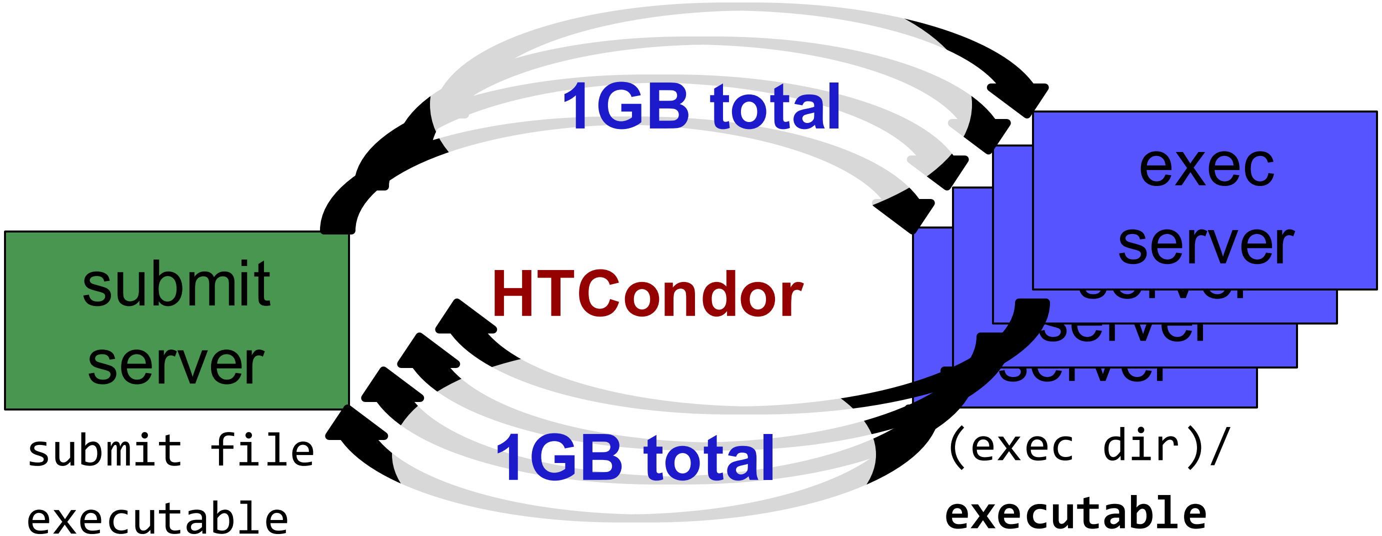


```
submit file
executable
dir/ input
output
```

```
(exec dir)/
executable
input
output
```

Output transfers are staggered

Hardware transfer limits



submit server

HTCondor

1GB total

1GB total

exec server

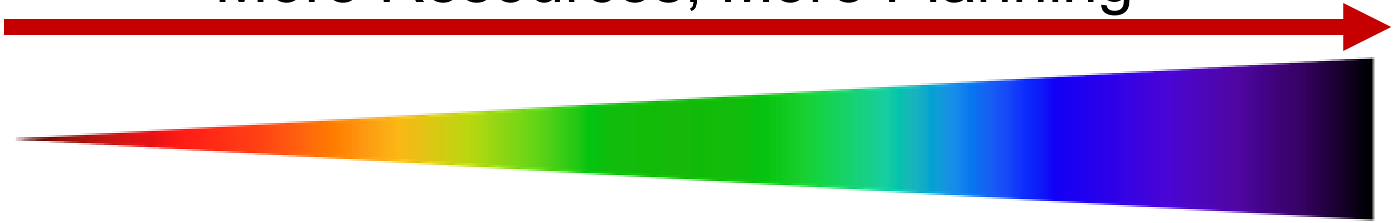
```
submit file
executable
dir/ input
output
```

```
(exec dir)/
executable
input
output
```

Like all things

We like to think of HTC/OSPool usage as a spectrum:

More Resources, More Planning



Laptop

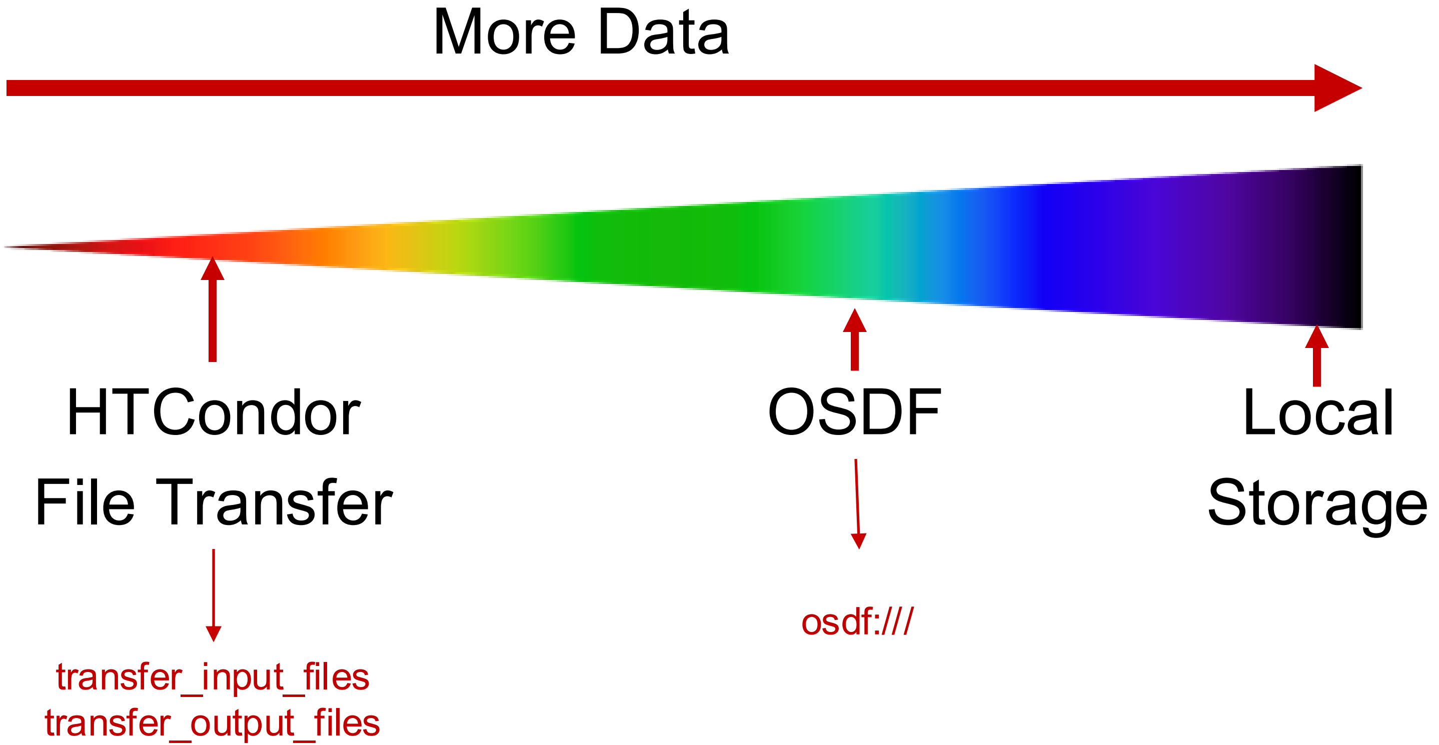
Cluster

OSPool

Outline

- What is ~~big~~ large data?
- Data Management Tips
- Characteristics of OSPool
- Solutions to moving data
  - HTCondor File Transfer
  - OSDF/Pelican

Transfers



More Data

HTCondor File Transfer

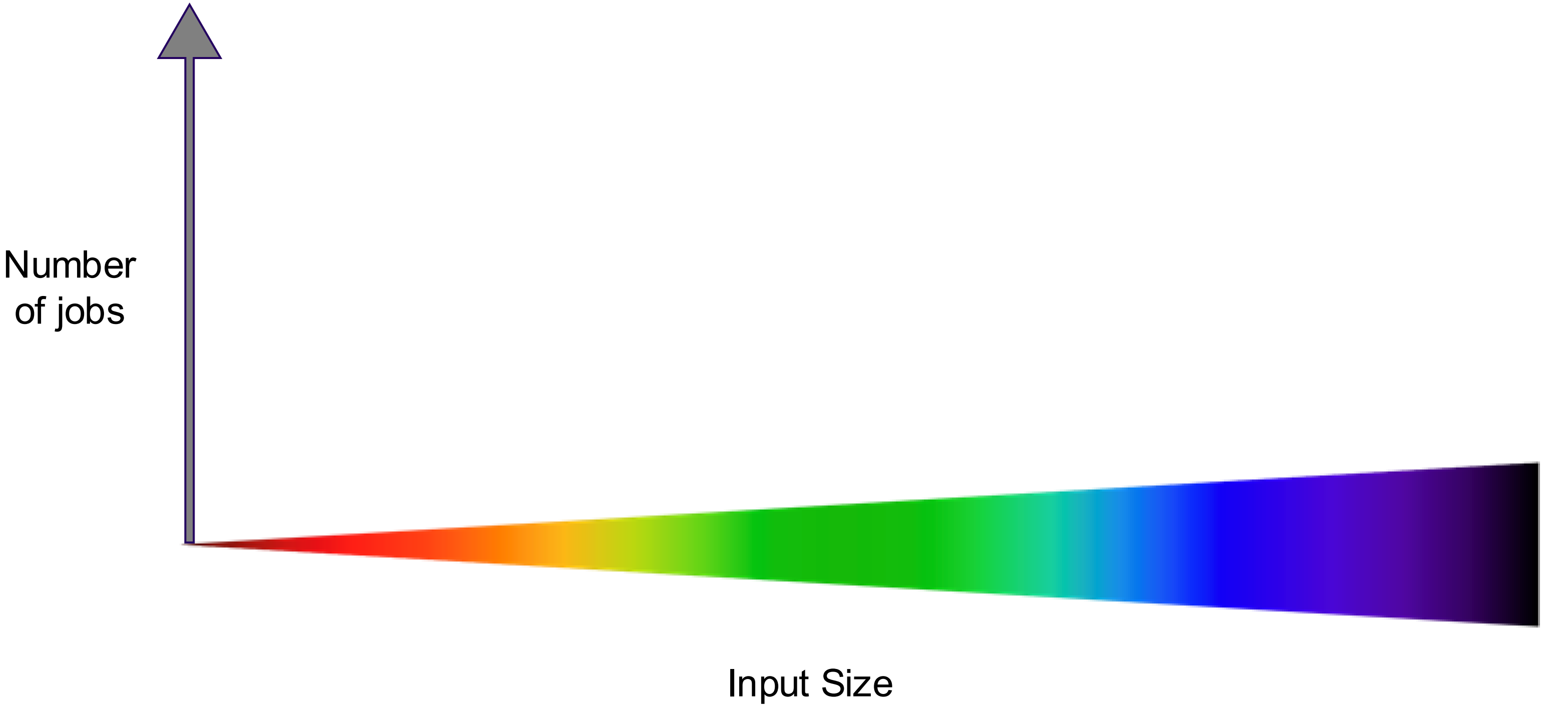
transfer_input_files
transfer_output_files

OSDF

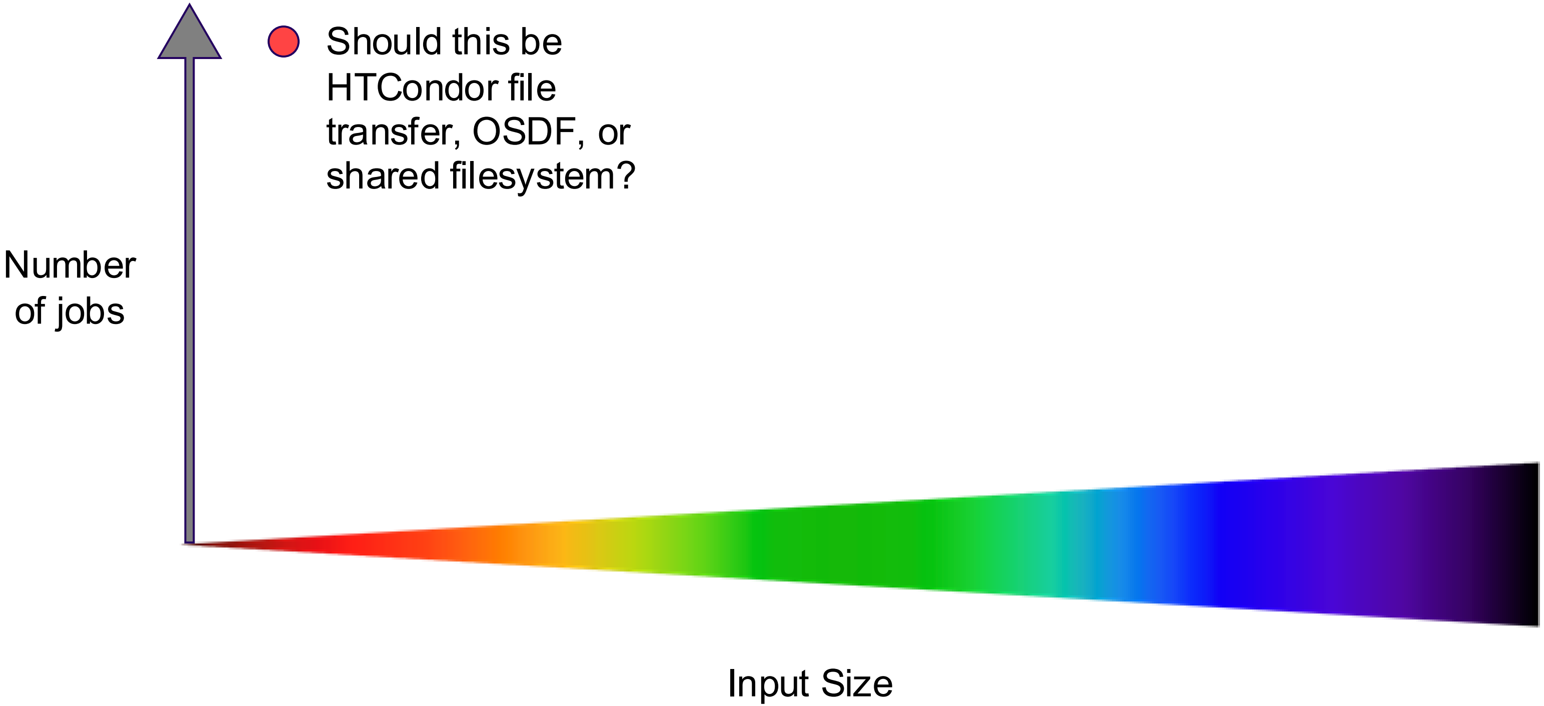
osdf:///

Local Storage

Rule of thumb - many dimensions



Rule of thumb - many dimensions

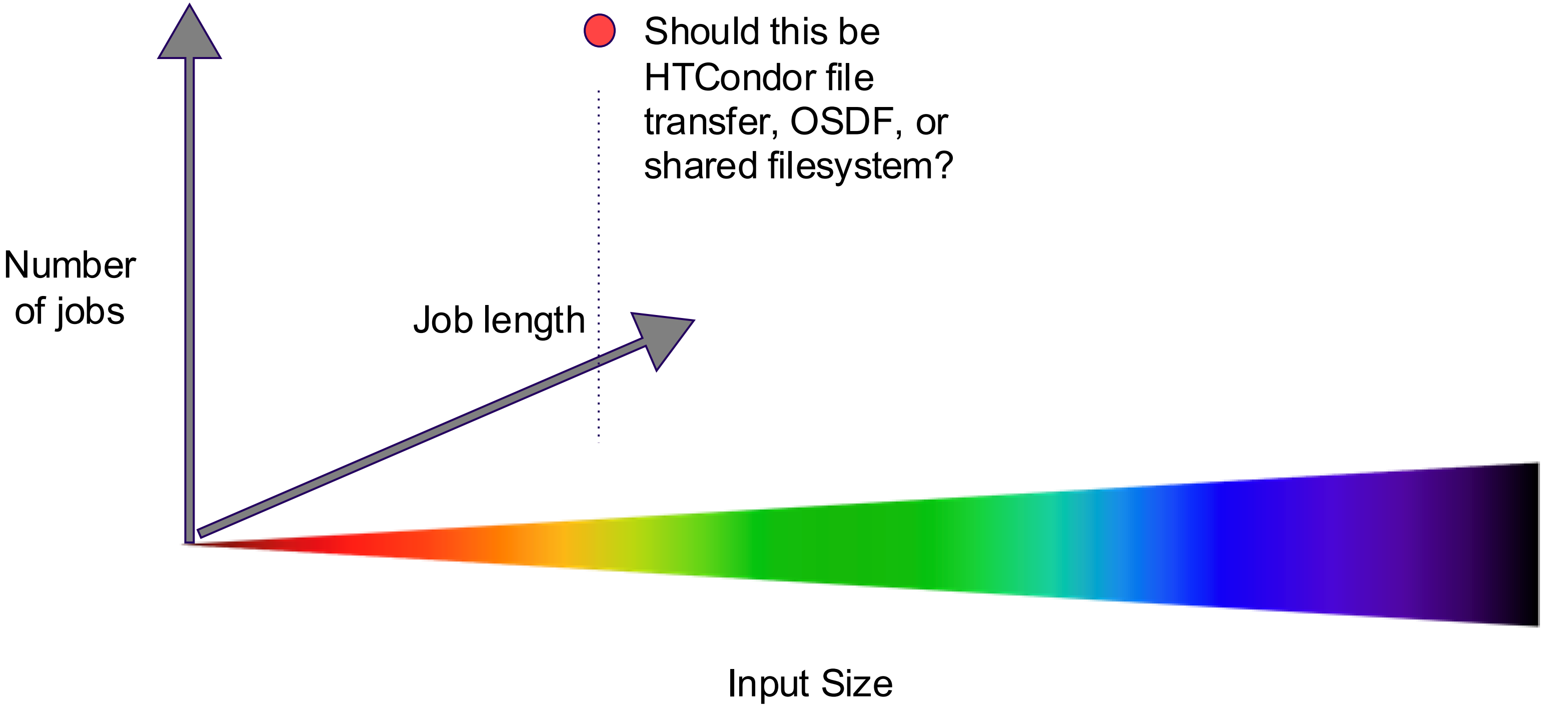


● Should this be HTCondor file transfer, OSDF, or shared filesystem?

Number of jobs

Input Size

Rule of thumb - many dimensions



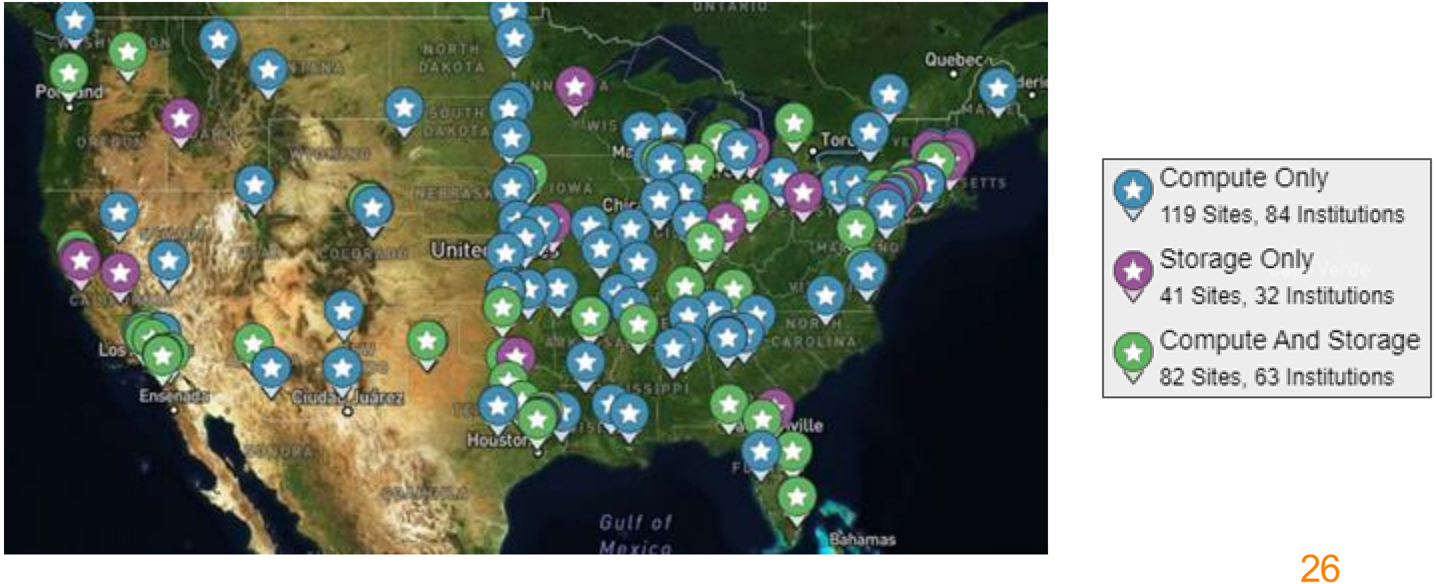
● Should this be HTCondor file transfer, OSDF, or shared filesystem?

Outline

- What is ~~big~~ large data?
- Data Management Tips
- Characteristics of OSPool
- Solutions to moving data
  - HTCondor File Transfer
  - OSDF/Pelican

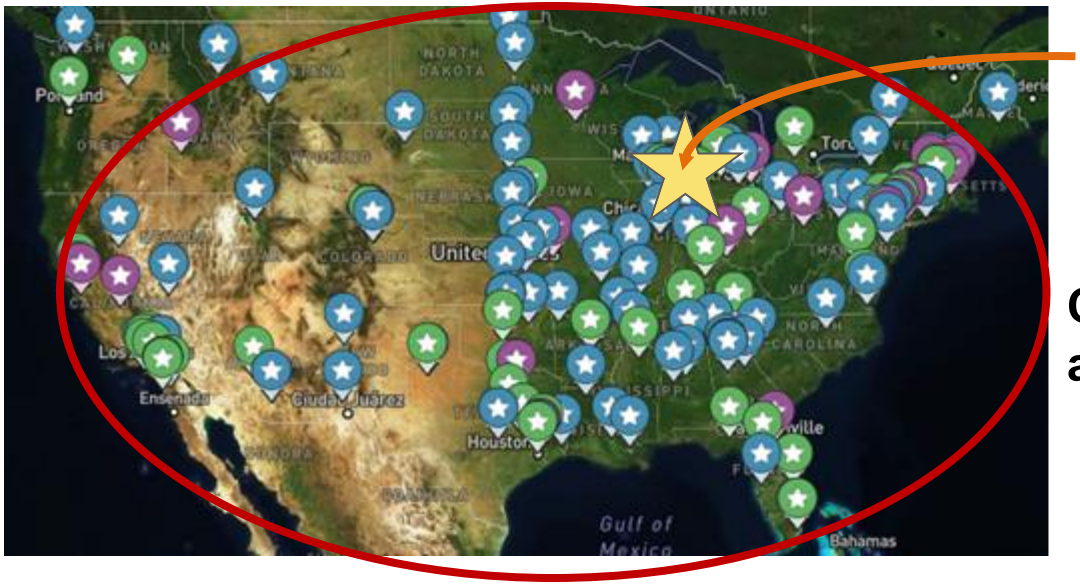
OSPool and the Open Science Data Federation (OSDF)

The OSPool is a High Throughput Computing system distributed across most of the United States, that runs 500,000 - 1,000,000+ jobs ***per day***



OSPool and the Open Science Data Federation (OSDF)

With distributed computing comes the need for data distribution that works at large scale and large volume



*Submitting Jobs Here**

Could run anywhere!

OSPool and the Open Science Data Federation (OSDF)

Submitting many jobs that use the same large file can quickly flood the network

10,000 jobs

x

10 GB input file

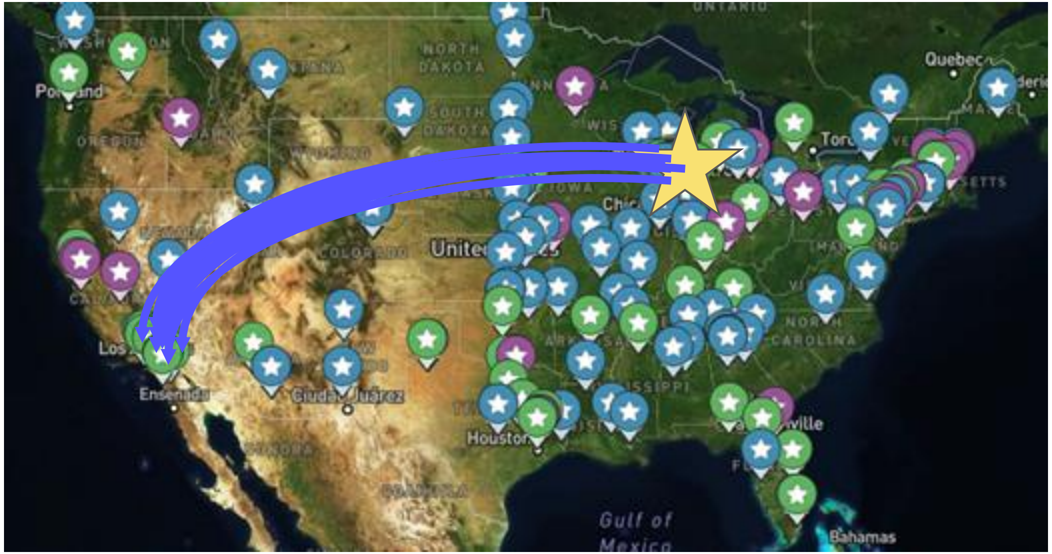
x

1 transfer / job

=

100,000 GB

network transfer



OSPool and the Open Science Data Federation (OSDF)

Enter the OSDF - a system of data caches that can stage large, repeatedly used files closer to the actual compute resources

10,000 jobs

x

10 GB input file

x

1 transfer total

=

10 GB

network transfer

Use OSDF to Transfer Large Input Files

OSPool users can use the OSDF to transfer large data for their HTCondor jobs

- Place large file(s) in /ospool/ap40/data/[Username]/large_file
- Use OSDF plugin in submit file:

```
transfer_input_files = osdf:///ospool/ap40/data/[Username]/large_file
```

3 slashes, not 2!

- HTCondor & OSDF automatically handle transfer of data when the job starts

https://portal.osg-htc.org/documentation/htc_workloads/managing_data/osdf/

- By default, only the OSPool user who placed the data can use that data

Use OSDF to Transfer Large Output Files

OSPool users can use the OSDF to transfer large data for their HTCondor jobs

- In your submit file, specify the output file(s) you want transferred with
  transfer_output_files = large_file
- Also in your submit file, remap the output location using OSDF plugin:
  transfer_output_remaps = "large_file = osdf:///ospool/ap40/data/[Username]/large_file”

**Use semicolons (;) to separate multiple entries*

HTCondor & OSDF automatically handle transfer of data when the job finishes

https://portal.osg-htc.org/documentation/htc_workloads/managing_data/osdf/

Good Practices for OSDF

- If you modify a file in OSDF please give the file a ***unique*** name, otherwise:
  - OSDF won’t know whether it’s a new/older file
  - Some jobs may run new version of the file, some will run with the old one
- Make sure to **delete data** when you no longer need it in the origin!!!

When to use HTCondor file transfer vs OSDF?

HTCondor File transfer:

Data Location: /home/<user.name>

Perfect for:

- Smaller files (<5GB)
- Repeated changed/updated files
- Submit Files
- Executables
- Temporary intermediate files

OSDF File transfer:

Data Location: /ospool/<ap##>/data/<user.name>

Perfect for:

- Larger files (>5GB)
- Repeated used files
- Containers

Pelican and the OSDF

Just like how OSG uses

HTCondor as the <u>software</u> that runs the *OSPool,*

OSG is transitioning to use

Pelican as the <u>software</u> that runs the *OSDF.*

The benefits for the OSDF (as the flagship instance of Pelican):

- More reliable, robust software stack
- Lots more room for new features, improvements
- More extensible to other contributors and data stores

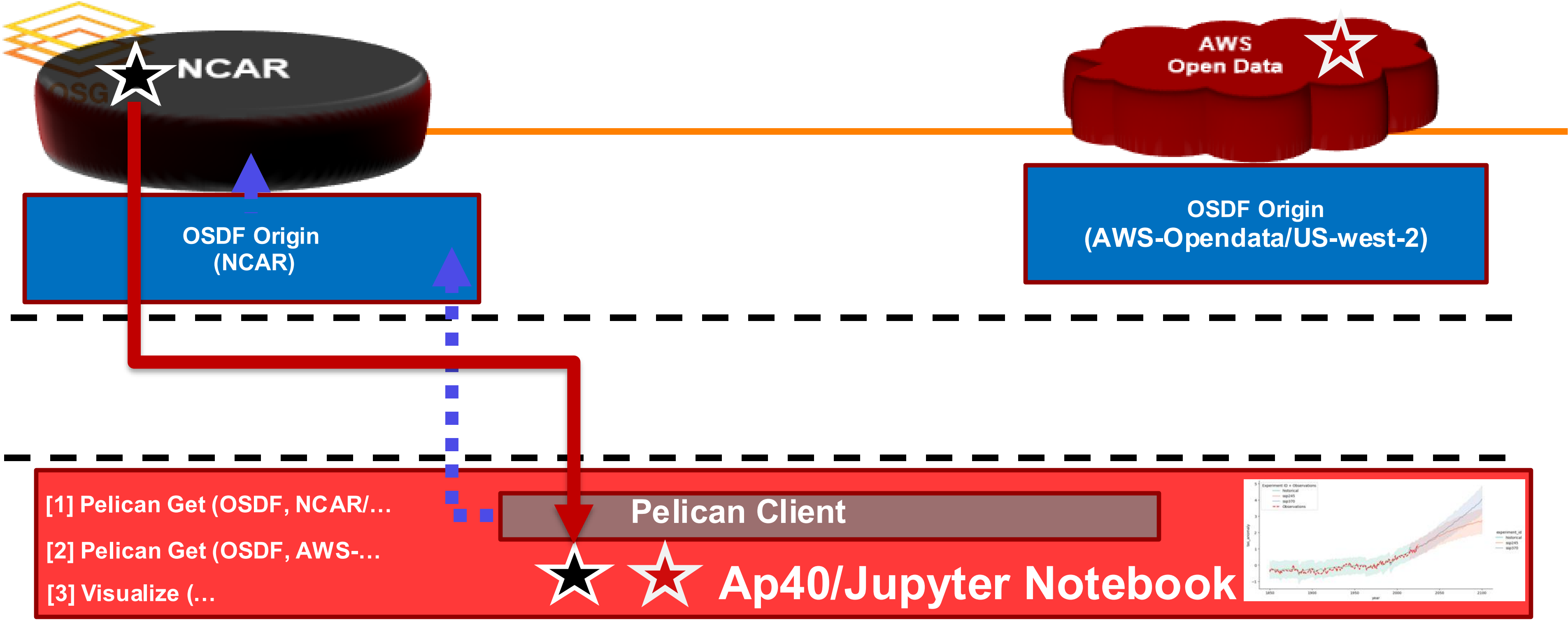
What is Pelican?

Like HTCSS, the Pelican Platform is an open-source software being developed at CHTC (Center for High Throughput Computing) at University of Wisconsin - Madison

pelicanplatform.org

Overall goals for Pelican development include

- empowering infrastructure for target domains, such as climate data
- supporting a wide range of storage backends to support user needs
- making the setup and use of Pelican services convenient and easy



OSDF Origin
(NCAR)

OSDF Origin
(AWS-Opendata/US-west-2)

[1] Pelican Get (OSDF, NCAR/…

[2] Pelican Get (OSDF, AWS-…

[3] Visualize (…

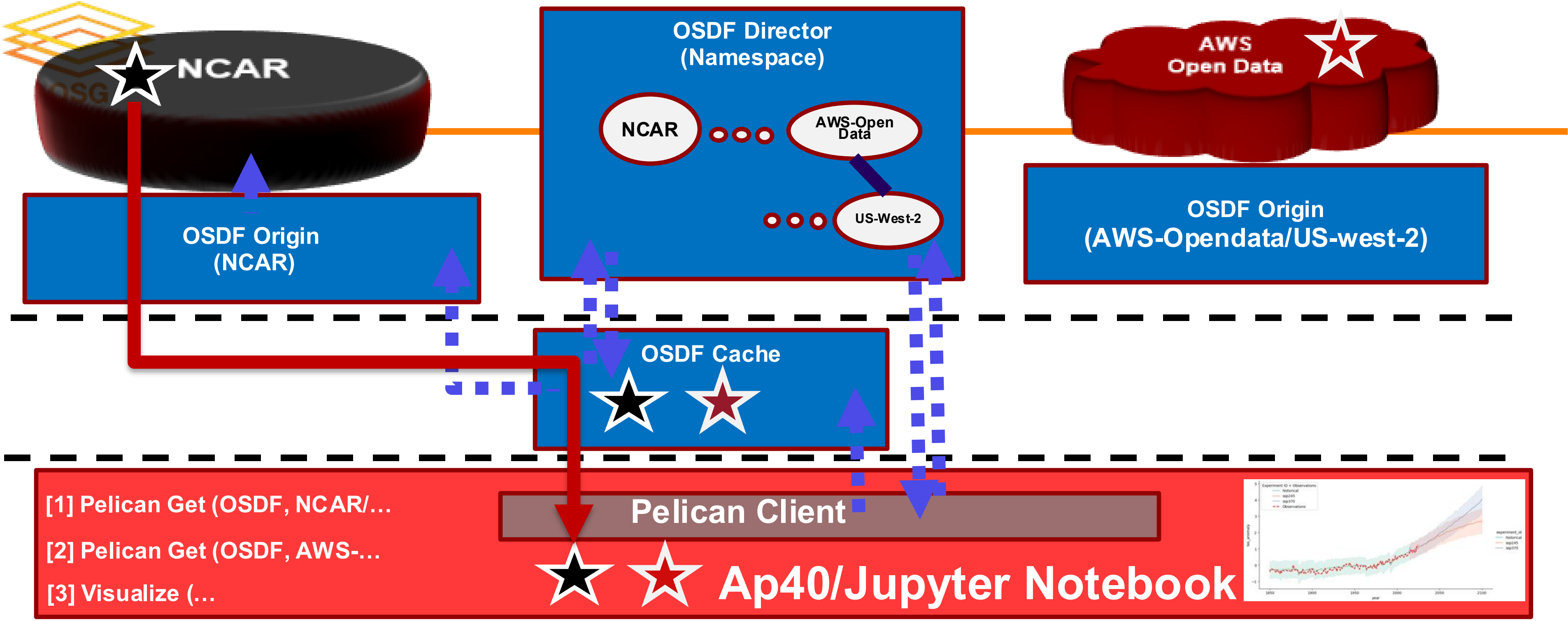
Pelican Client

★ ★ Ap40/Jupyter Notebook

Researcher uses a Jupyter Notebook to create a visualization that requires two objects:

★ **NCAR**/ rda/harshah/osdf_data/HadCRUT.5.0.2.0.analysis.summary_series.global.monthly.zarr

★ **AWS-OpenData/US-West-2**/cmip6-pds/CMIP6/CFMIP/NCAR/CESM2/aqua-4xCO2/r1i1p1f1/Amon/co2mass/gn/v20190816



[1] Pelican Get (OSDF, NCAR/…

[2] Pelican Get (OSDF, AWS-…

[3] Visualize (…

Researcher uses a Jupyter Notebook to create a visualization that requires two objects:

★ **NCAR**/ rda/harshah/osdf_data/HadCRUT.5.0.2.0.analysis.summary_series.global.monthly.zarr

★ **AWS-OpenData/US-West-2**/cmip6-pds/CMIP6/CFMIP/NCAR/CESM2/aqua-4xCO2/r1i1p1f1/Amon/co2mass/gn/v20190816

More info about Pelican: HTC24 talks

- "Deployment Scale and Use of OSDF" session: https://agenda.hep.wisc.edu/event/2175/contributions/30968/
- "Introducing Pelican: Powering the OSDF" https://agenda.hep.wisc.edu/event/2175/contributions/30967/
- "Pelican under the hood: how the data federation works" https://agenda.hep.wisc.edu/event/2175/contributions/31334/
- "Connecting Pelican to your data" https://agenda.hep.wisc.edu/event/2175/contributions/31335/
- "Data in Flight: Delivering Data with Pelican – Tutorial" https://agenda.hep.wisc.edu/event/2175/contributions/31337/

Questions?

Quick Reference

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

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)
- *MOST* perform better with fewer large files (versus many small files of typical HTC)

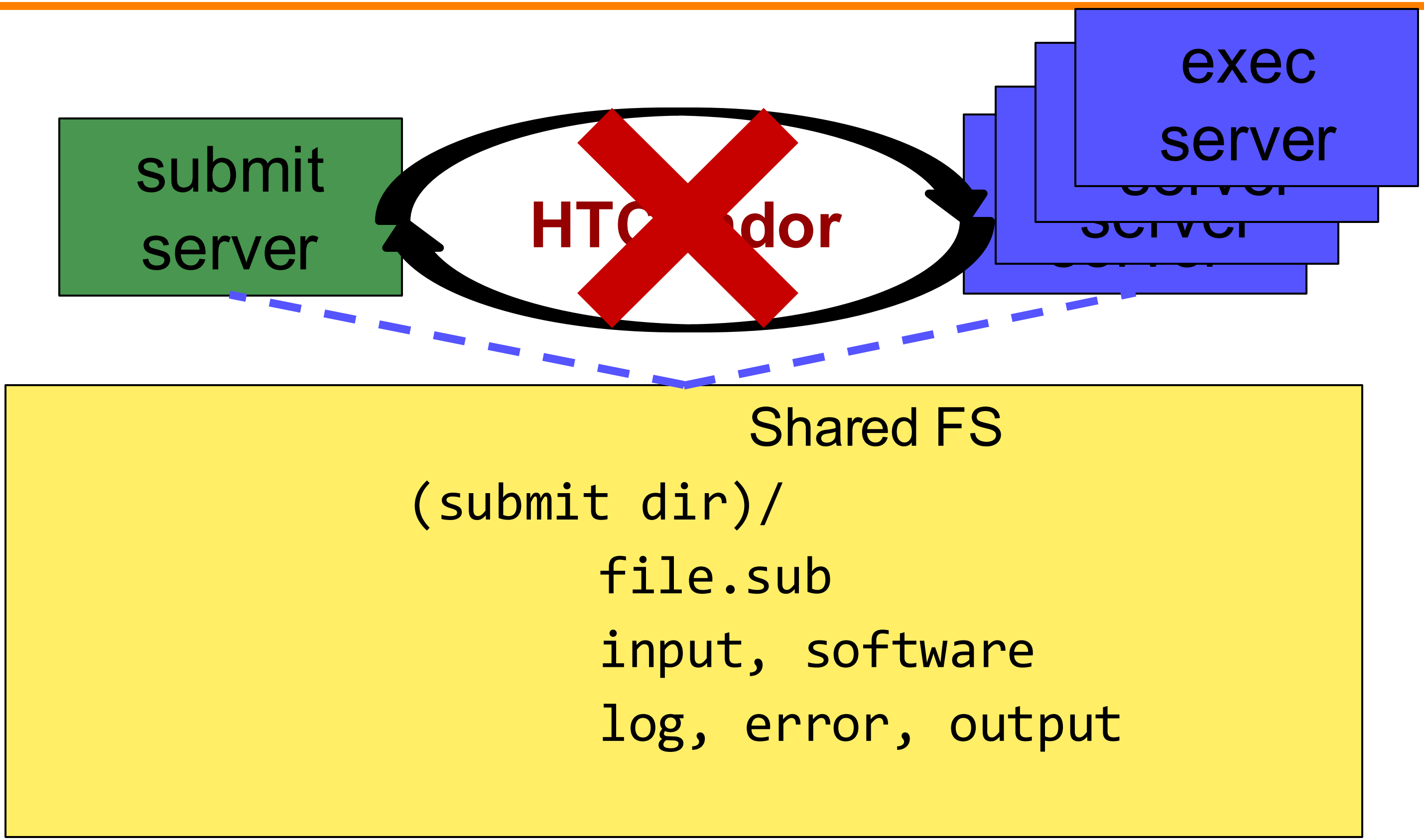
Shared FS Technologies

- *via network mount*
  - NFS
  - AFS
  - Lustre
  - Isilon (may use NSF mount)
- *distributed file systems (data on many exec servers)*
  - HDFS (Hadoop)
  - CEPH

Shared FS Configurations

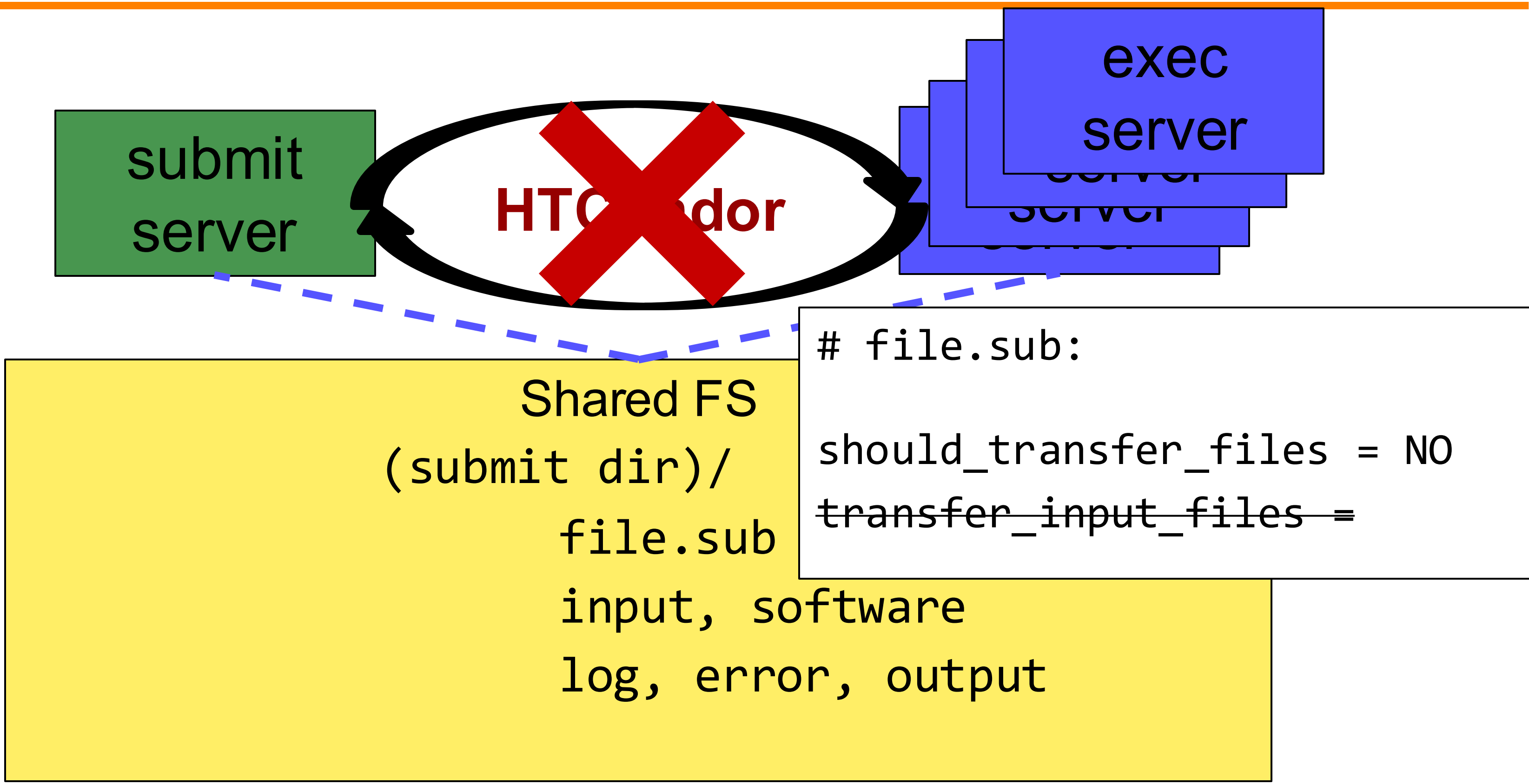
1. Submit directories *WITHIN* the shared filesystem
   - most campus clusters
   - limits HTC capabilities!!
2. Shared filesystem separate from local submission directories
   - supplement local HTC systems
   - treated more as a repository for VERY large data (>GBs)
3. Read-only (input-only) shared filesystem
   - Treated as a repository for VERY large input, only

Submit dir within shared FS



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

Submit dir within shared FS

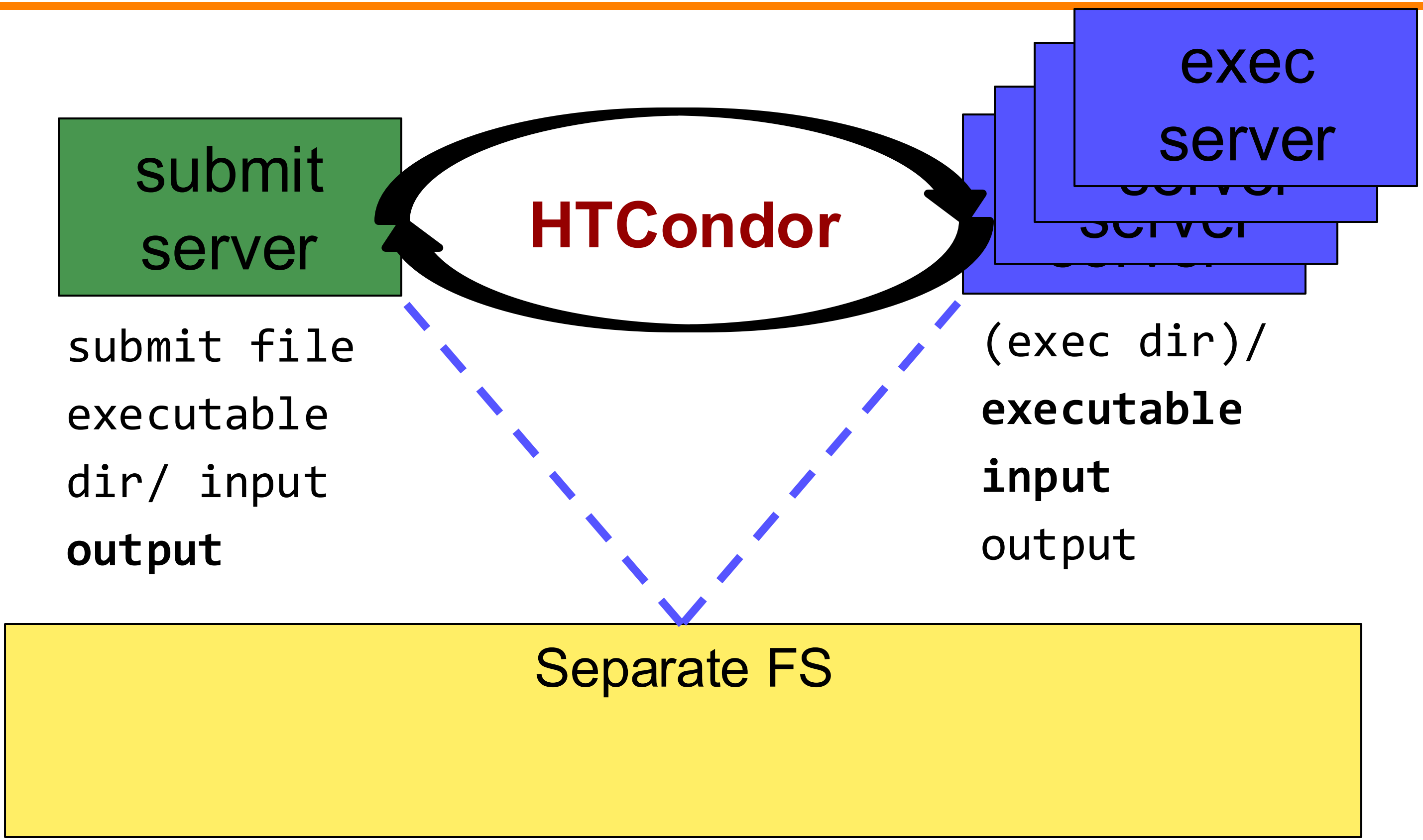


# file.sub:

should_transfer_files = NO

~~transfer_input_files =~~

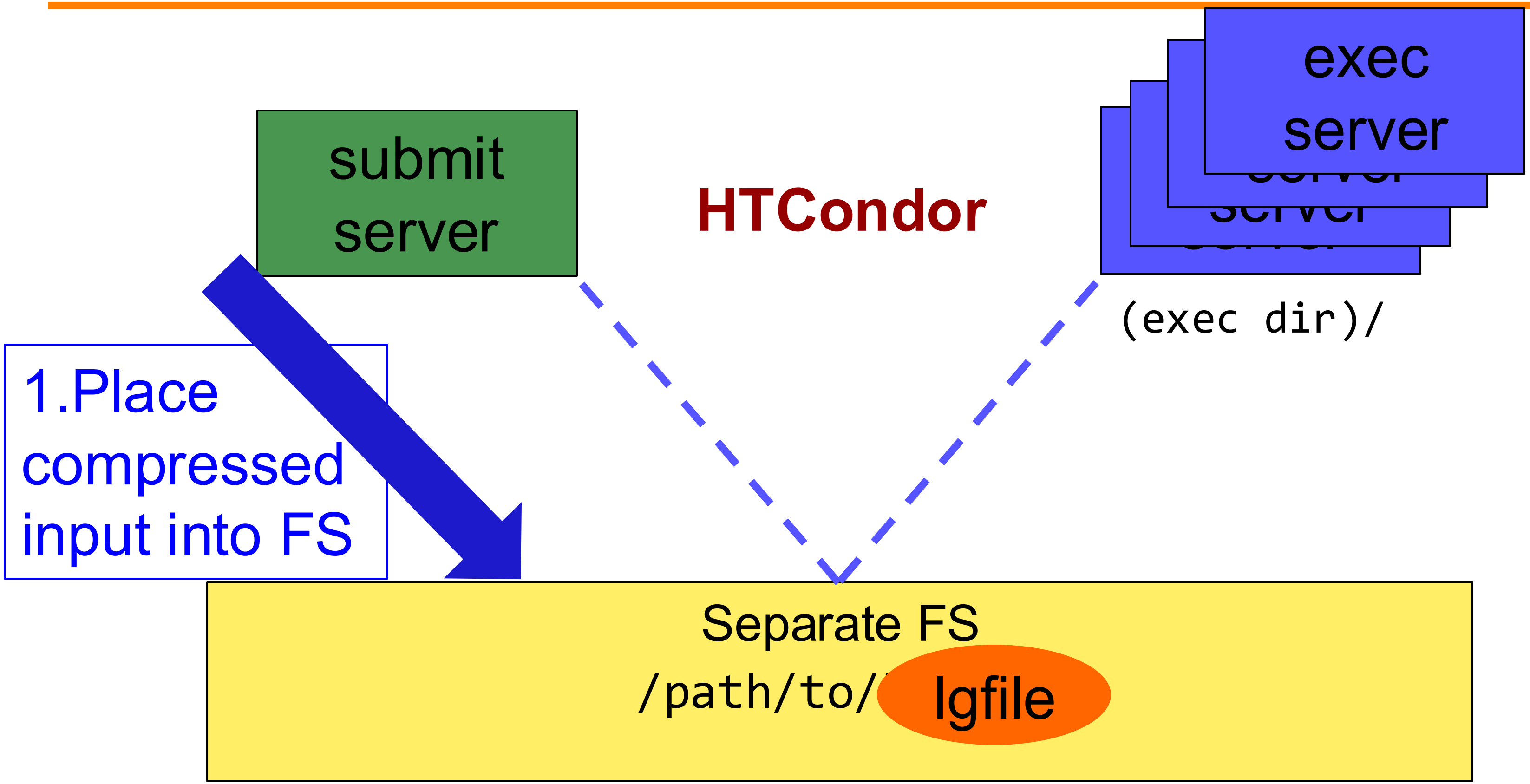
Separate shared FS



```
submit file
executable
dir/ input
output
```

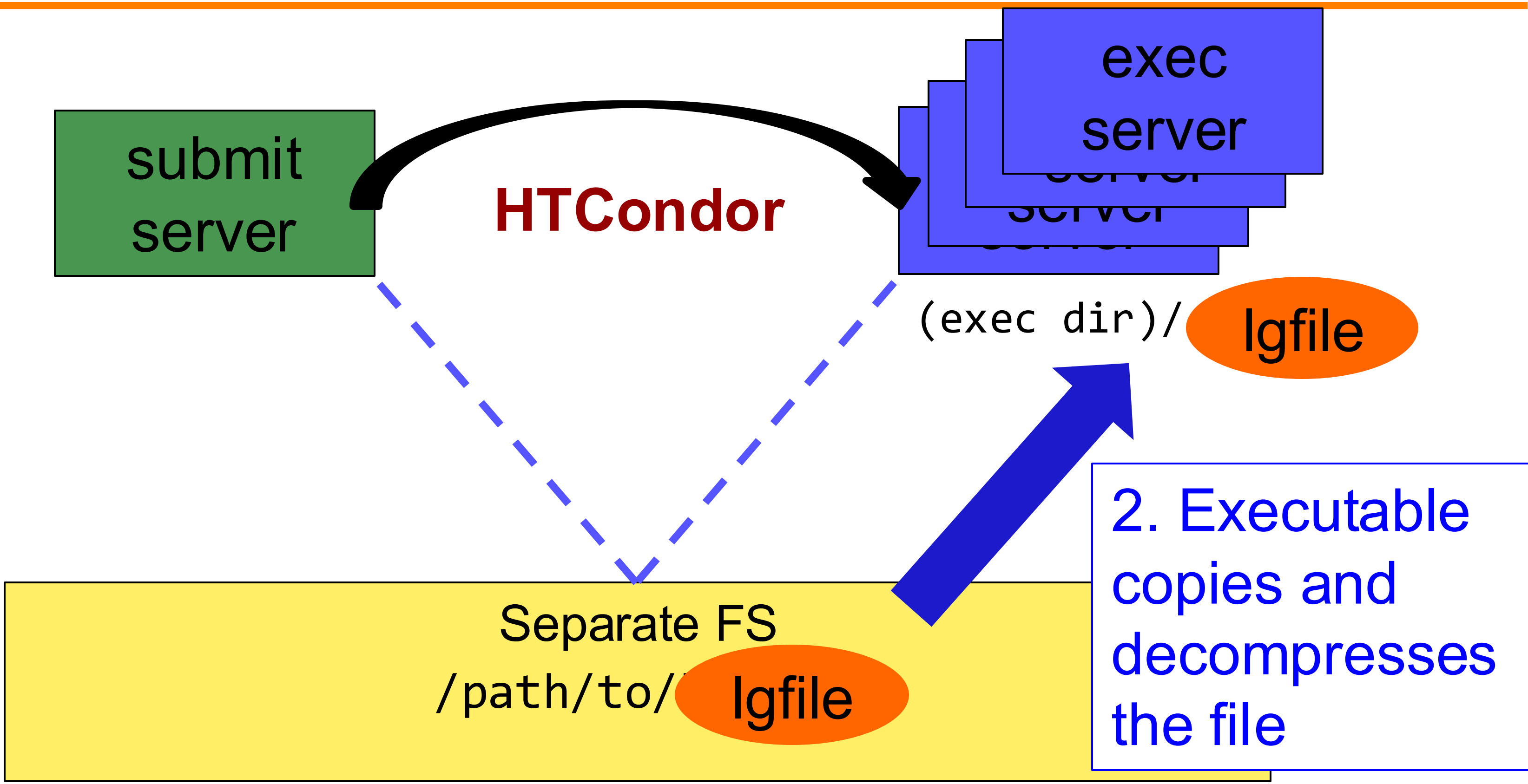
```
(exec dir)/
executable
input
output
```

Separate shared FS - Input



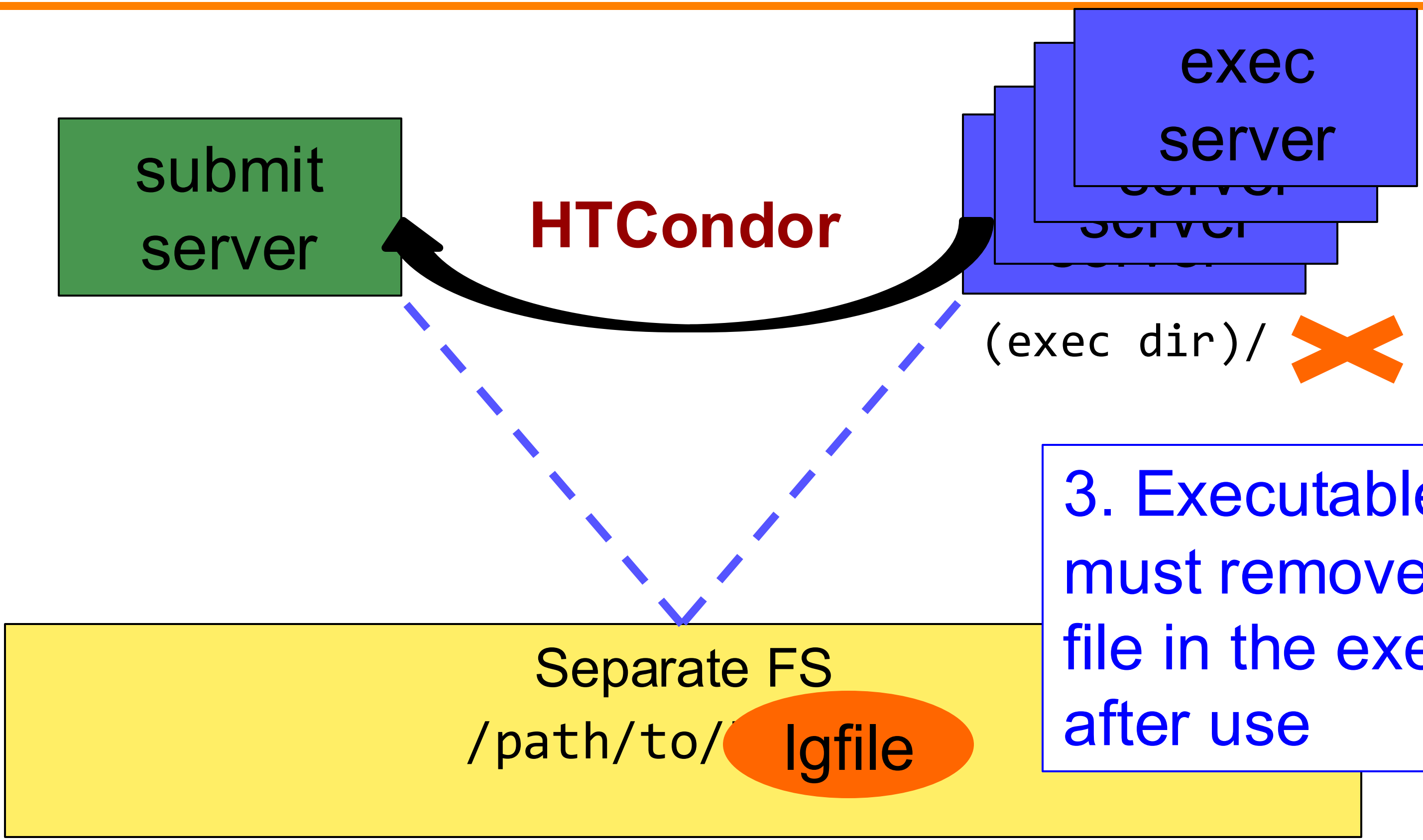
1.Place compressed input into FS

Separate shared FS - Input



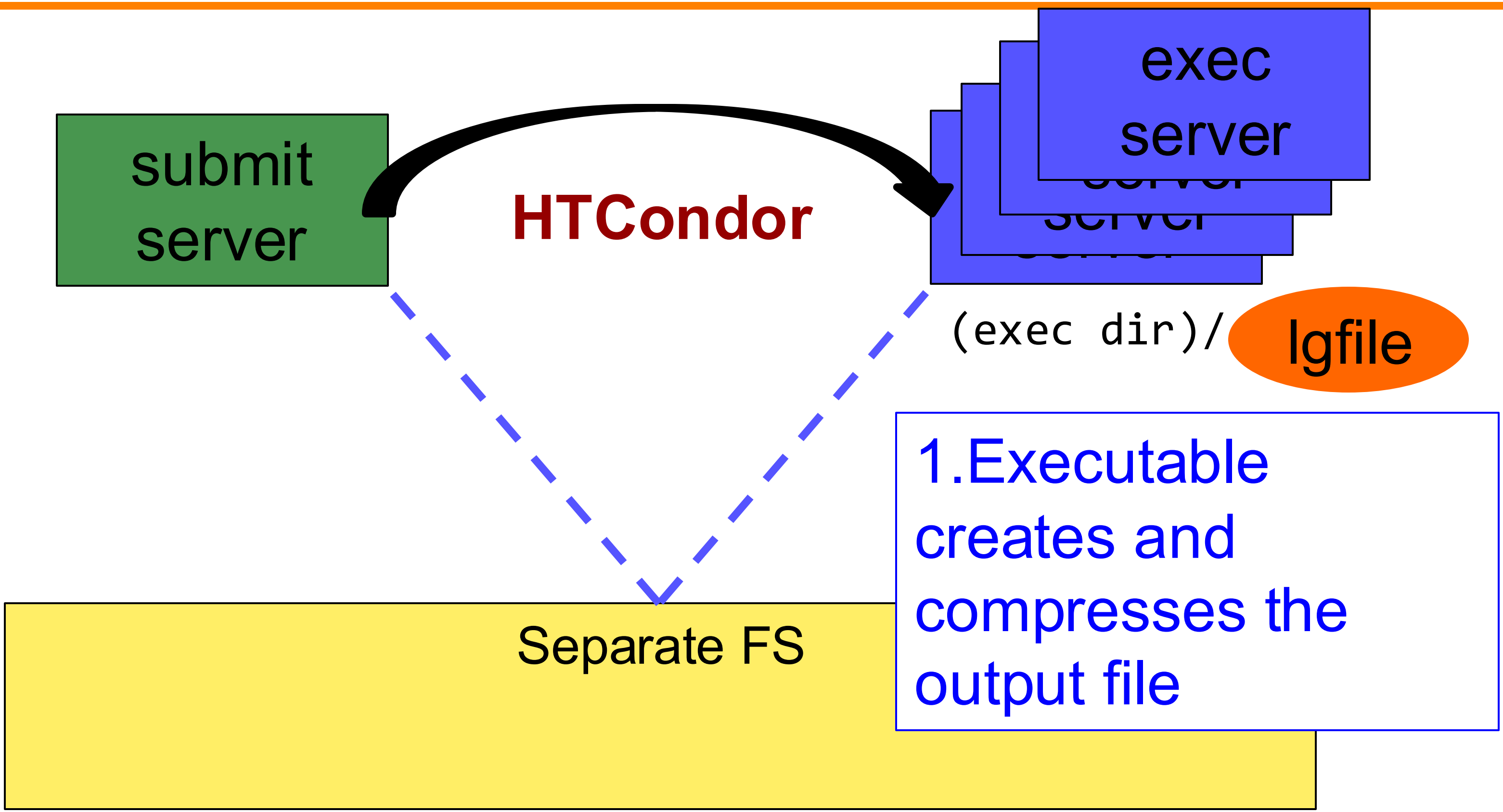
2. Executable copies and decompresses the file

Separate shared FS - Input



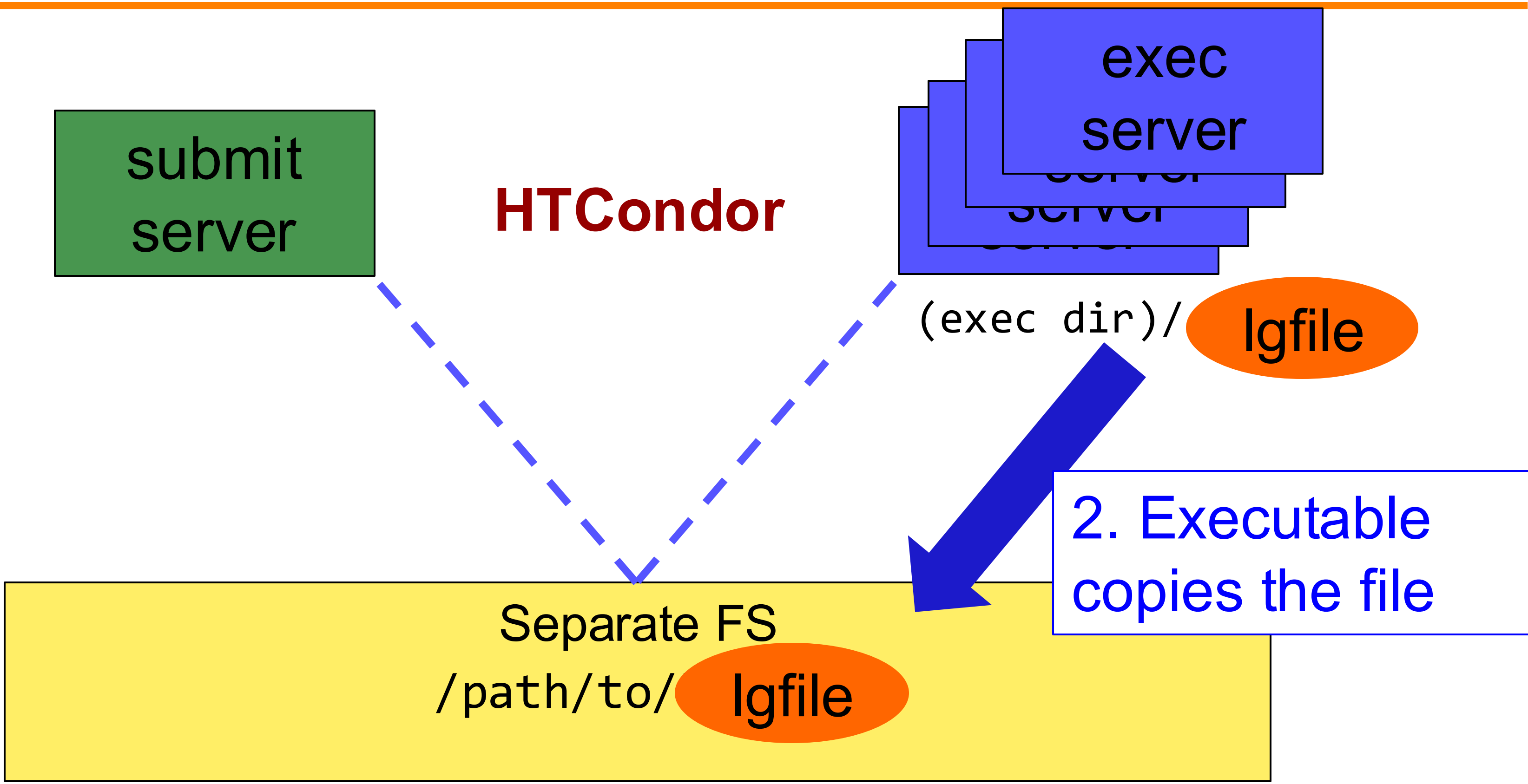
3. Executable must remove the file in the exec dir after use

Separate shared FS - Output



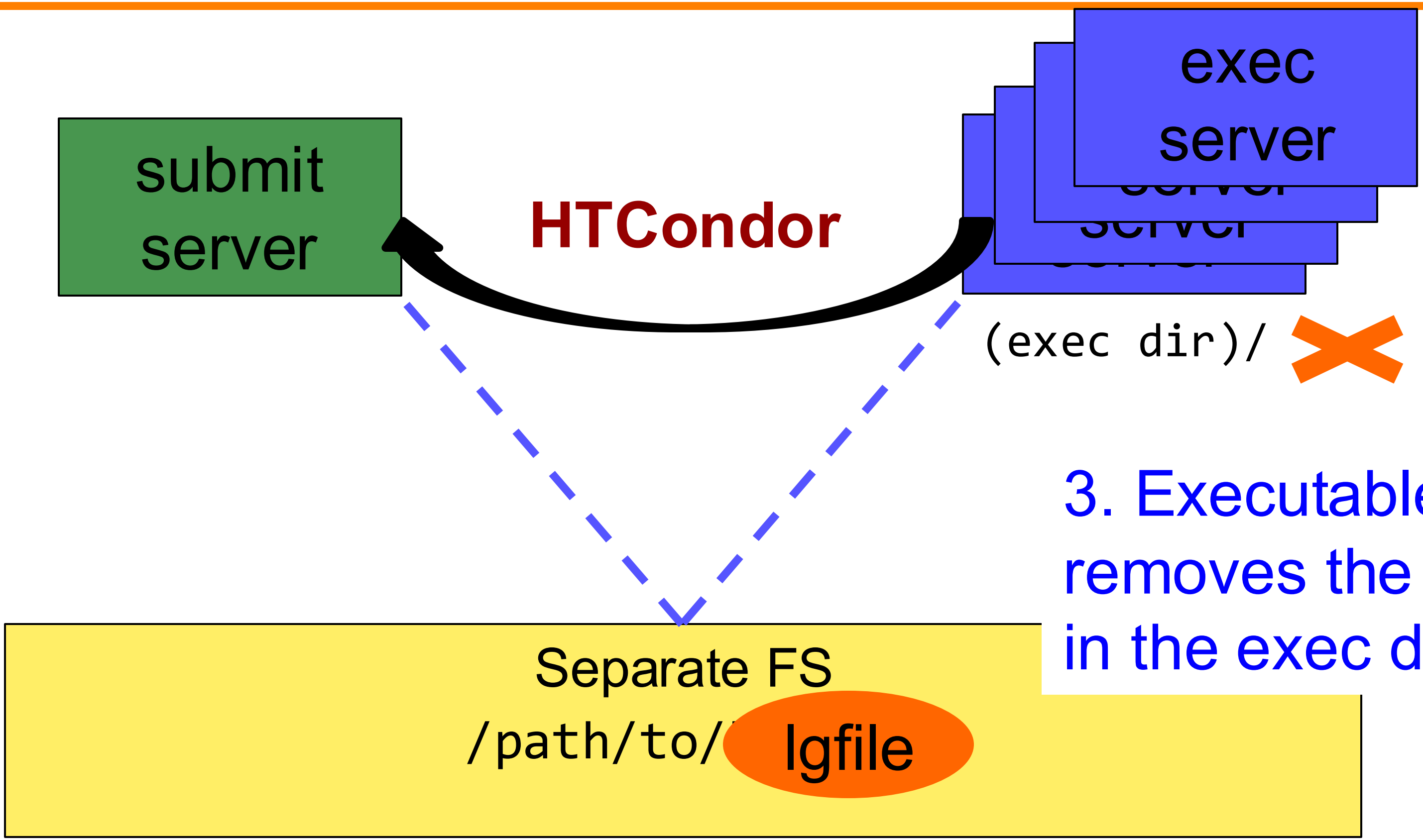
1.Executable creates and compresses the output file

Separate shared FS - Output



2. Executable copies the file

Separate shared FS - Output



3. Executable removes the file in the exec dir