Open Science Grid

Handling Data on OSG

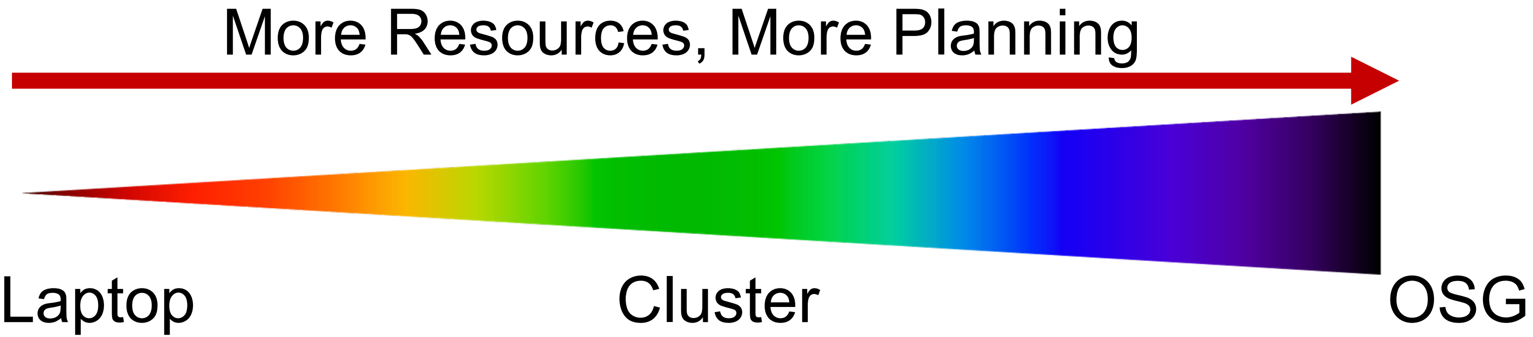
Friday, August 6

Derek Weitzel (dweitzel@unl.edu)

Open Science Grid

Like all things

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



Planning?

- Can’t control a cluster like your laptop, where you can install any software and place files (until they flat-out don’t fit)
- OSG: heterogeneity, borrowed resources (including network and disk), lack of on-the-fly troubleshooting

Open Science Grid

Benefits!

- On a cluster & OSG you can access 1000+ cores!
- Automate job tasks (with HTCondor)!
- Doesn’t burn up your laptop!

Handling Data on OSG

- Overview / Things to Consider
- HTCondor File Transfer
- Web Proxy
- Stash
- Shared File Systems

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*** software

- “**Output**” includes any files produced for the job that *need to come back*
  - `output, error`

Data Management Tips

- **Determine your per-job needs**
  - minimize per-job data needs
- Determine your *batch* needs
- 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

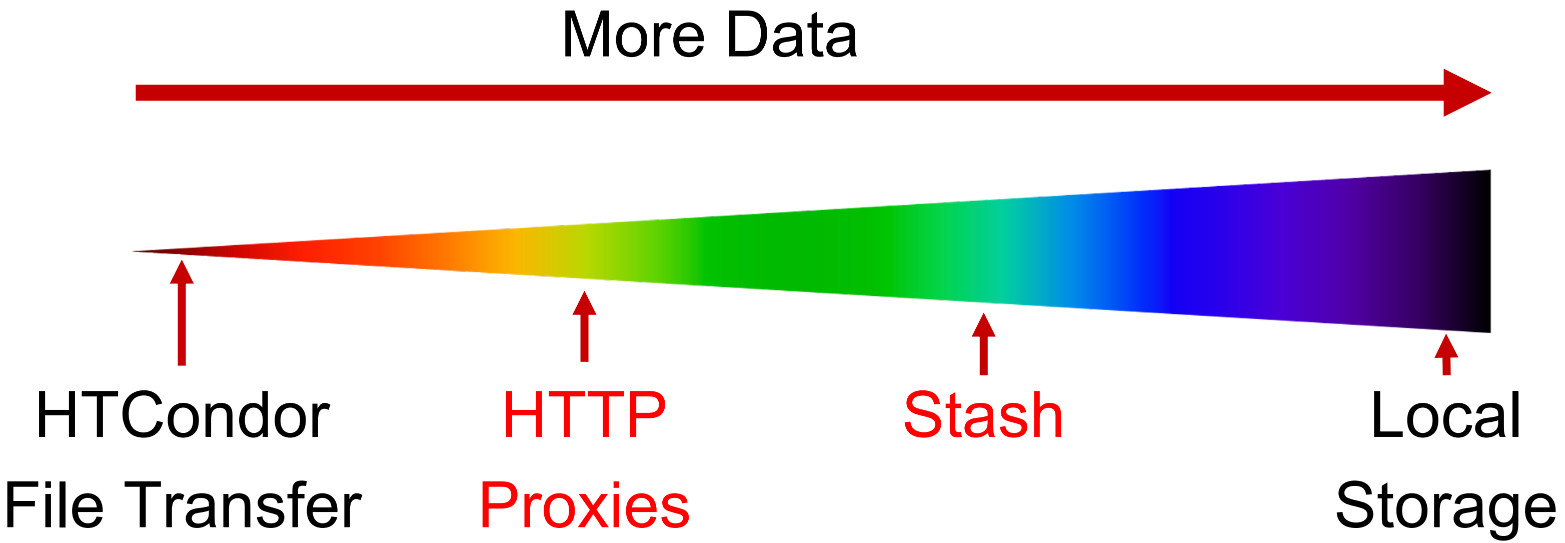
What method would you use to send data to a collaborator?

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

Transfers



More Data

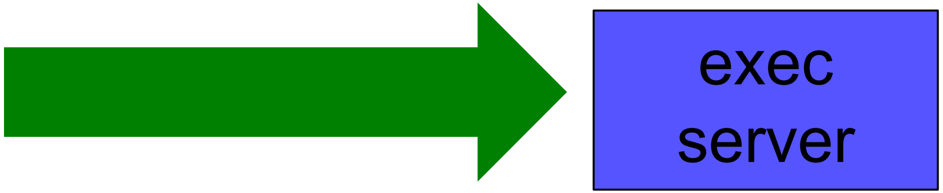
HTCondor File Transfer

HTTP Proxies

Stash

Local Storage

Large *input* in HTC and OSG

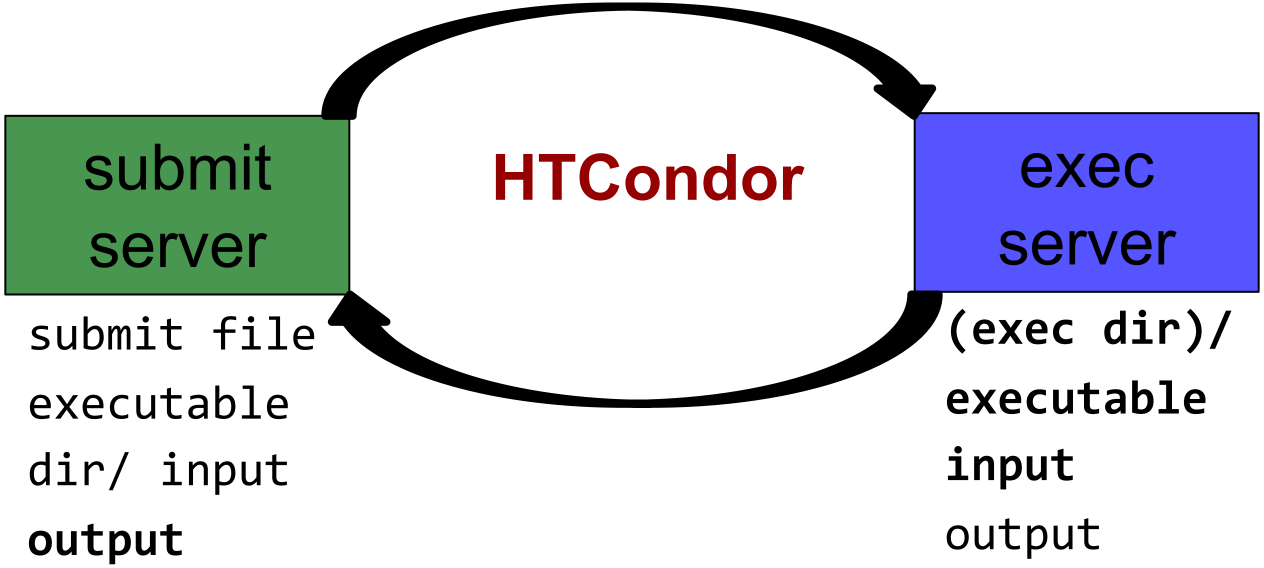


file size	method of delivery
words	within executable or arguments?
tiny – 100MB per file	HTCondor file transfer (up to 500MB total per job)
100MB – 1GB, shared	download from web server (local caching)
1GB – 10GB, **unique** or shared	Stash (regional replication)
10 GB – TBs	shared file system (local copy, local execute servers)

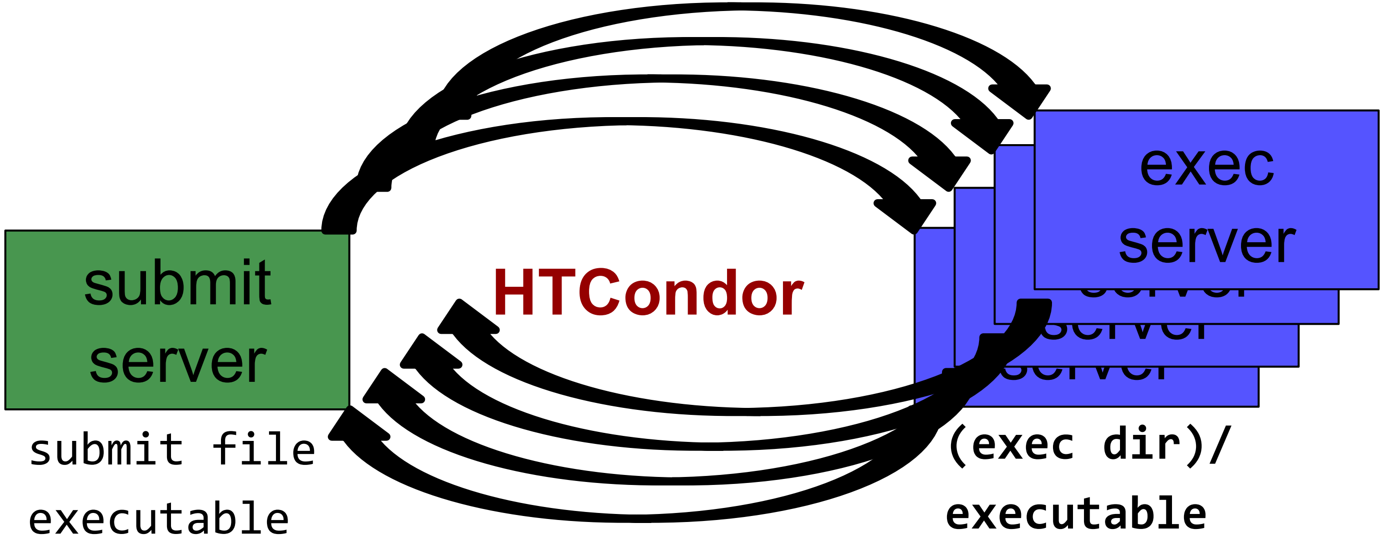
Handling Data on OSG

- ~~Overview / Things to Consider~~
- **HTCondor File Transfer**
- Web Proxy
- Stash
- Shared File Systems and Other Options

Review: HTCondor Data Handling



Network bottleneck: the submit server



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

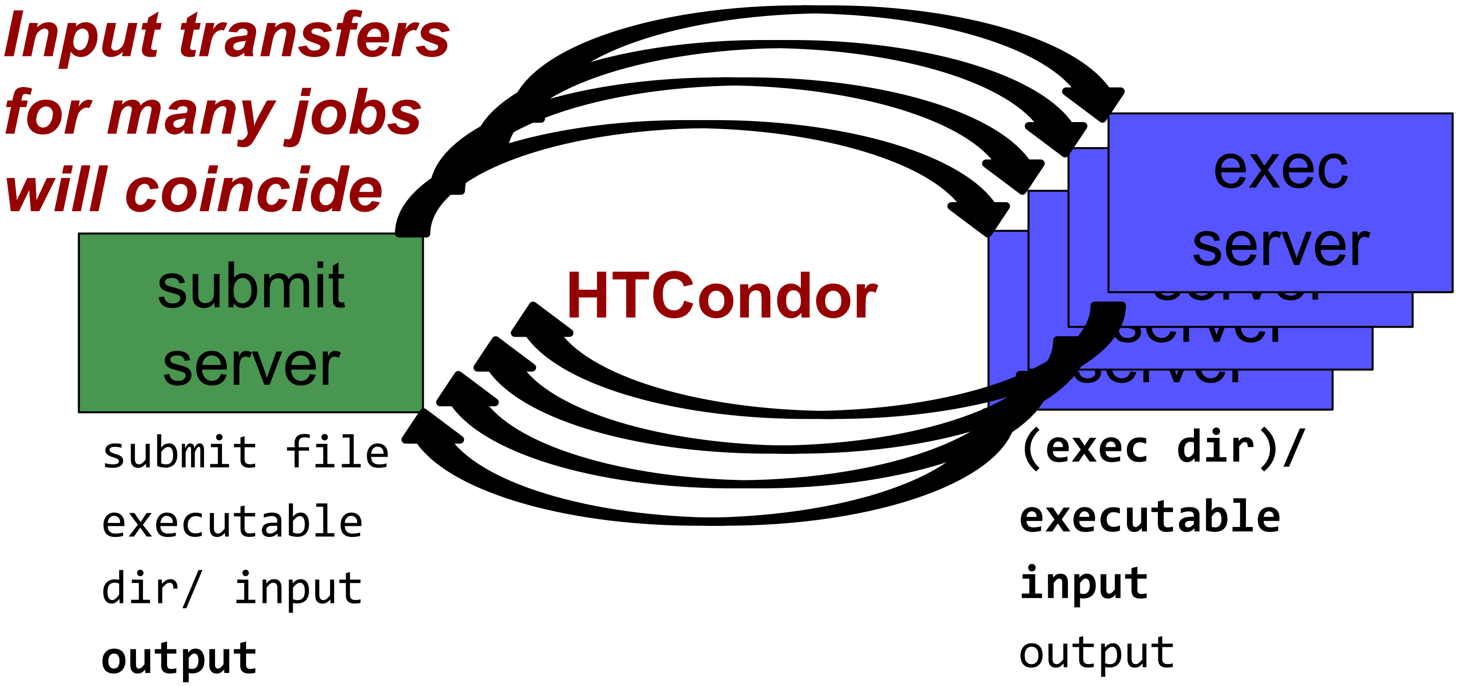
(exec dir)/

executable

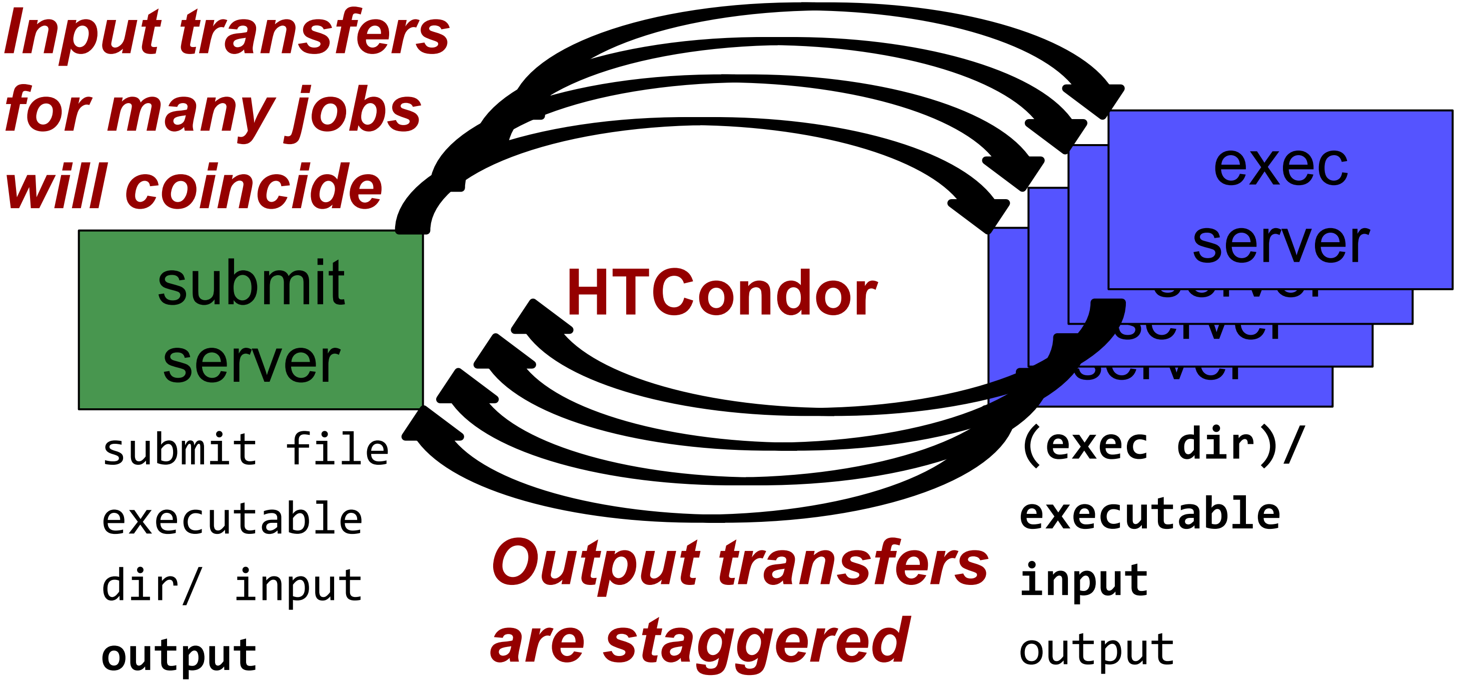
input

output

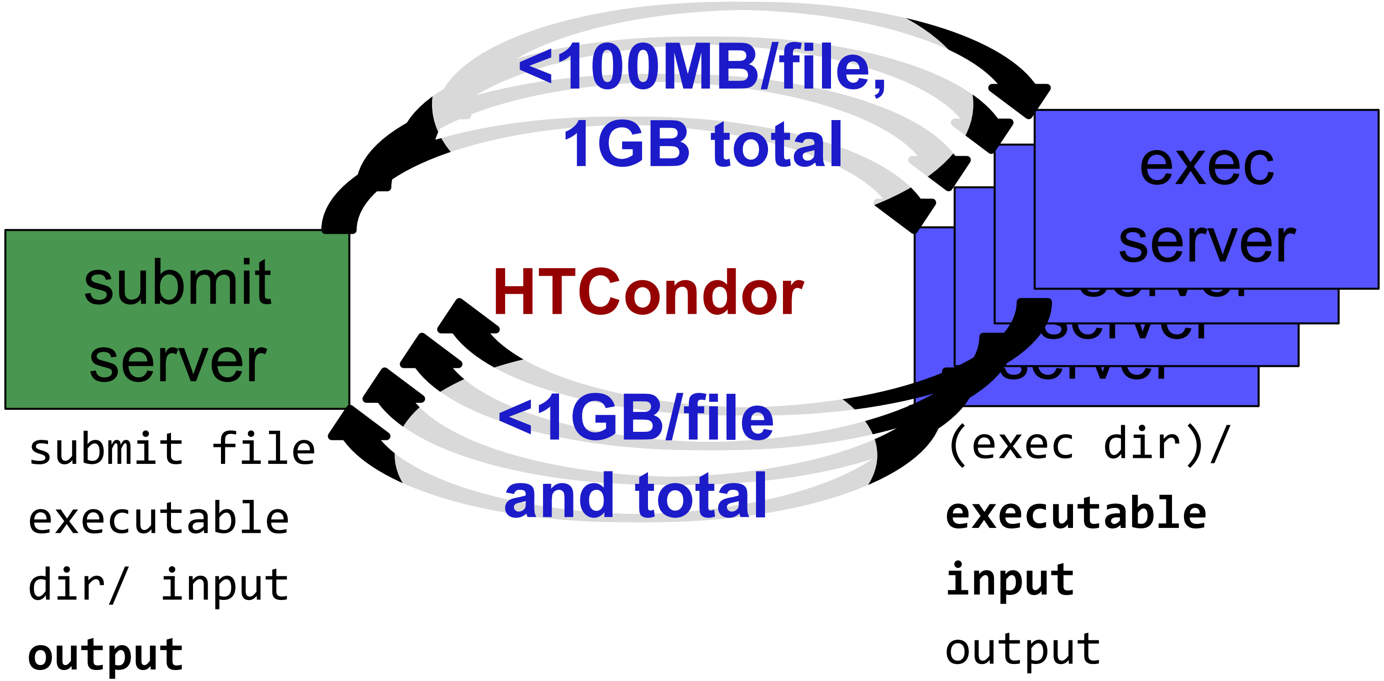
Network bottleneck: the submit server



Network bottleneck: the submit server



Hardware transfer limits



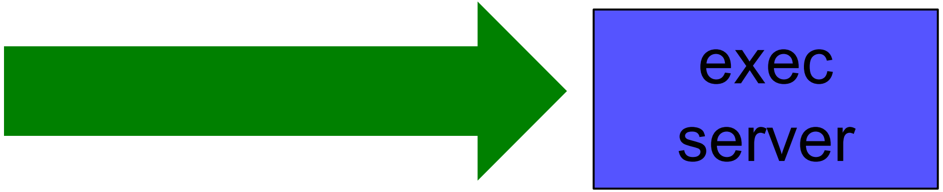
submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Handling Data on OSG

- ~~Overview / Things to Consider~~
- ~~HTCondor File Transfer~~
- **Web Proxy**
- Stash
- Shared File Systems and Other Options

Large input in HTC and OSG

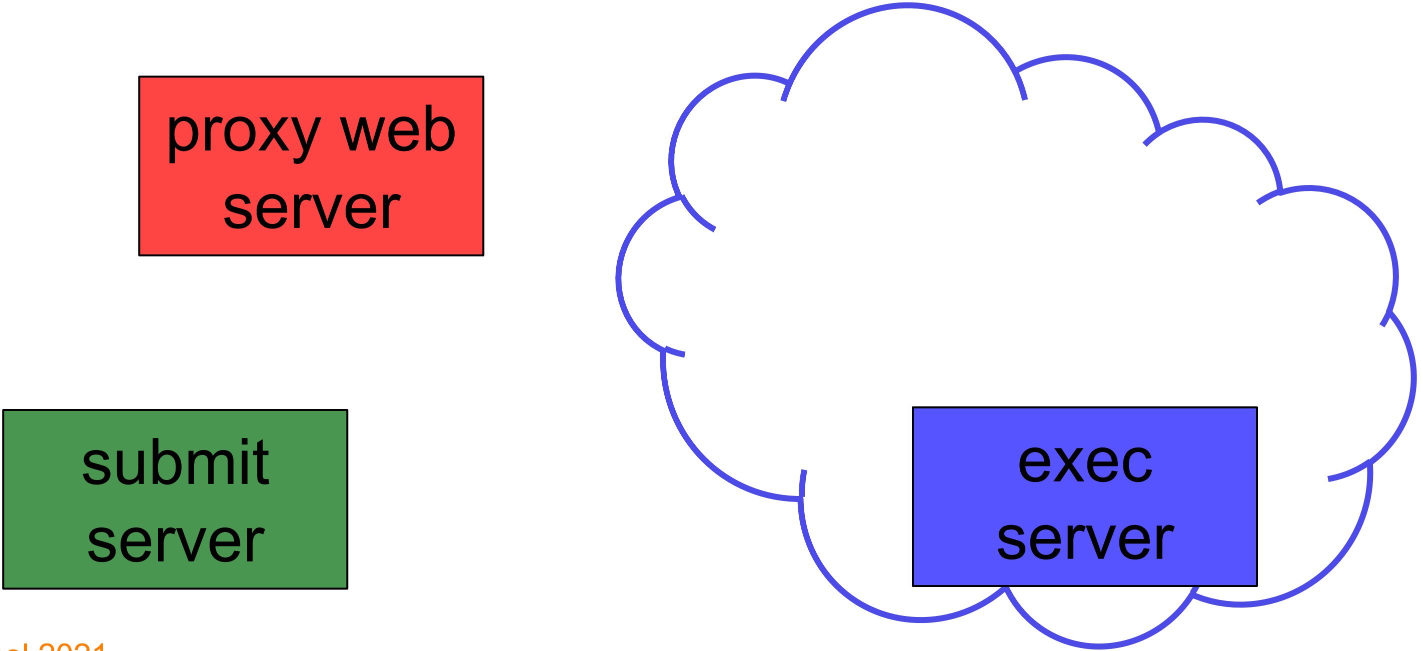


exec server

file size	method of delivery
words	within executable or arguments?
tiny – 100MB per file	HTCondor file transfer (up to 1GB total per-job)
100MB – 1GB, shared	download from web server (local caching)
1GB – 20GB, unique or shared	Stash (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

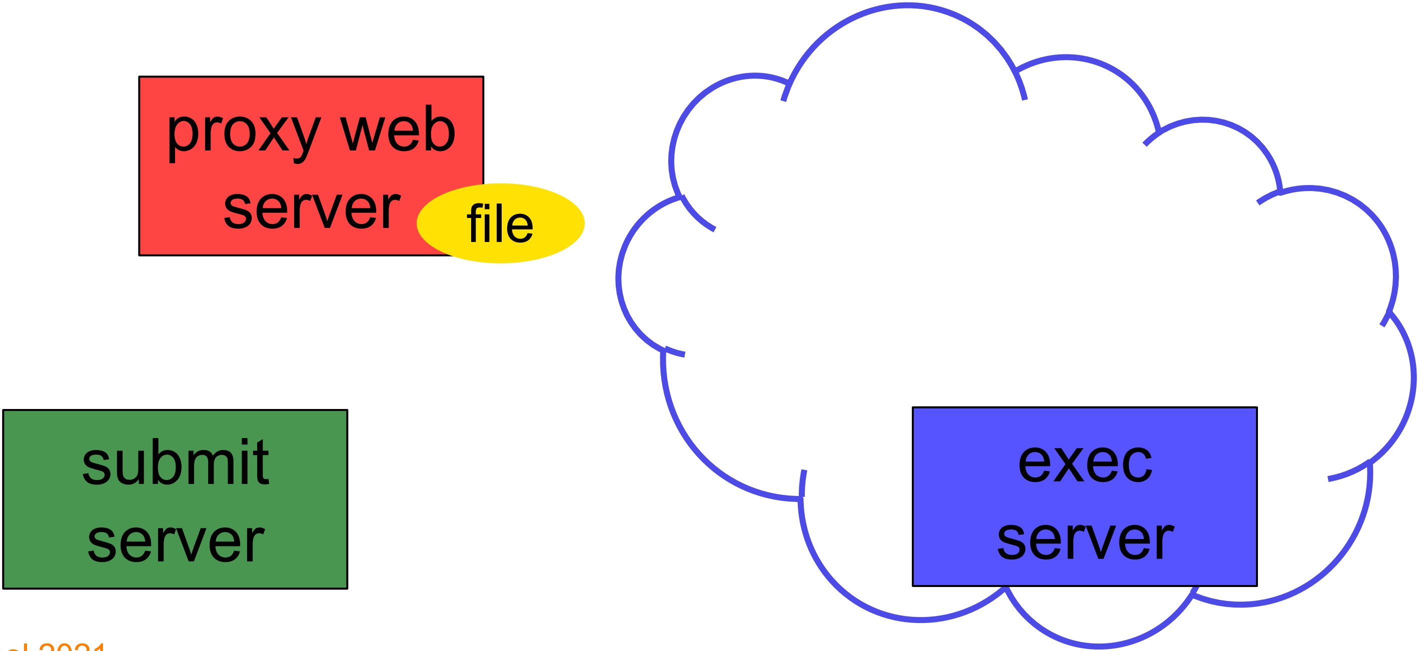
Using a Web Proxy

- Place the file onto a local, proxy-configured web server
- Have HTCondor download via HTTP address



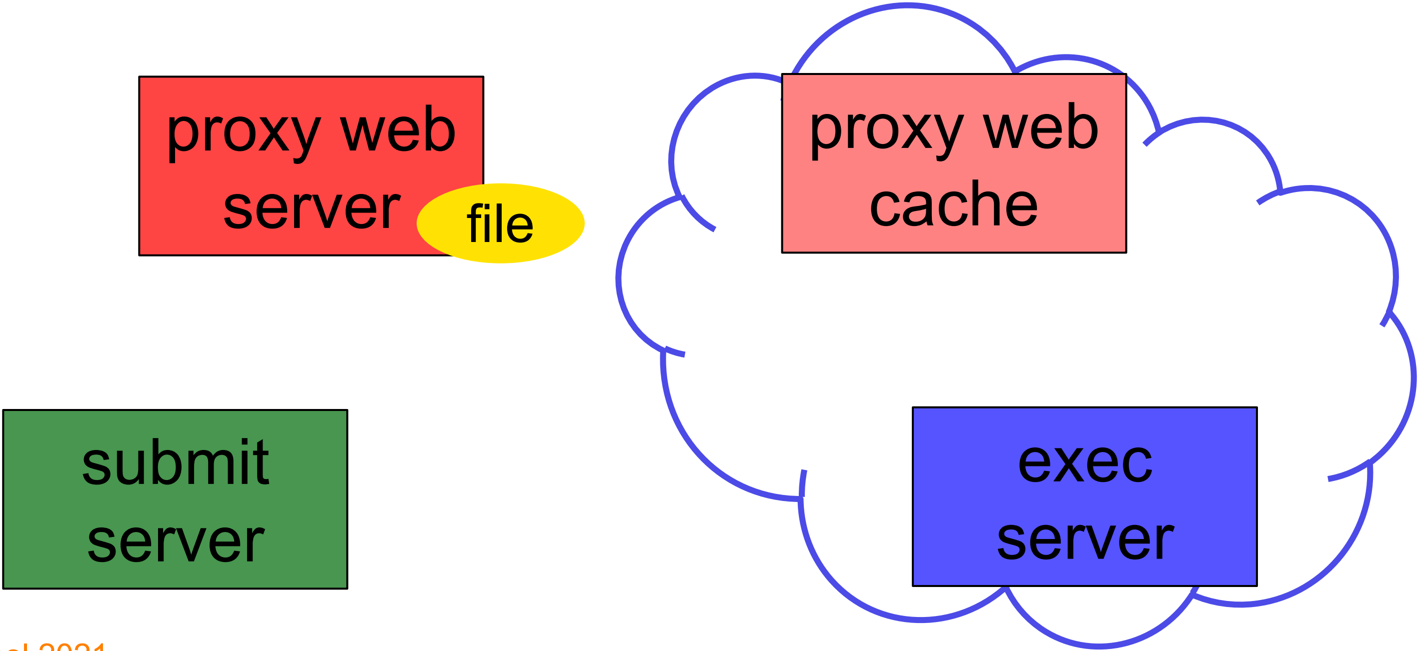
Using a Web Proxy

- Place the file onto a proxy-configured web server
- Have HTCondor download via HTTP address



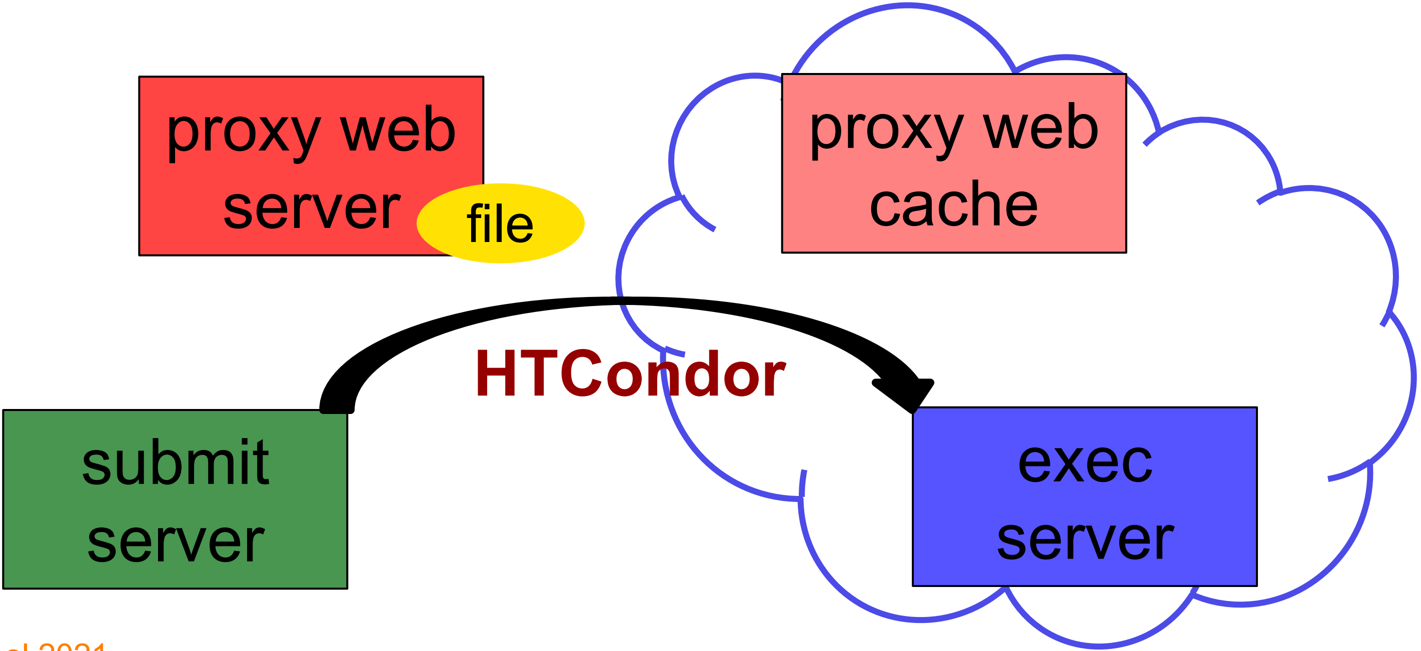
Using a Web Proxy

- Place the file onto a proxy-configured web server
- Have HTCondor download via HTTP address



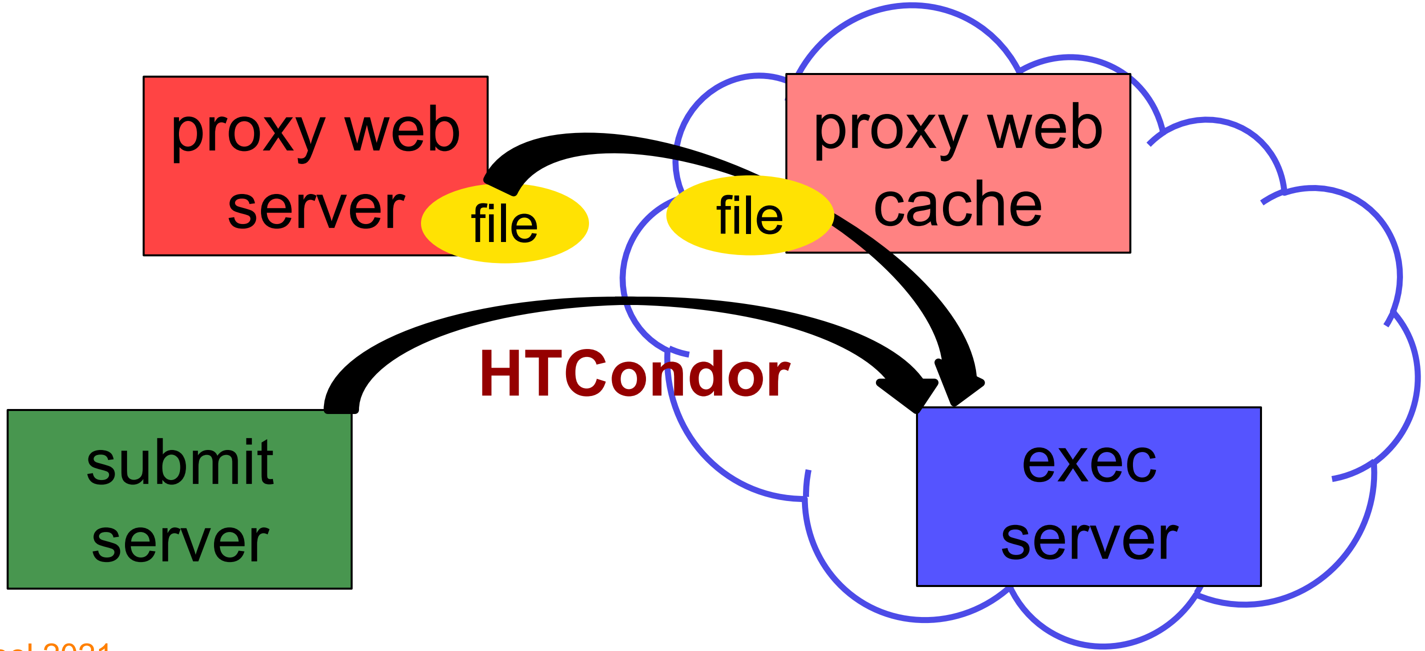
Using a Web Proxy

- Place the file onto a proxy-configured web server
- Have HTCondor download via HTTP address



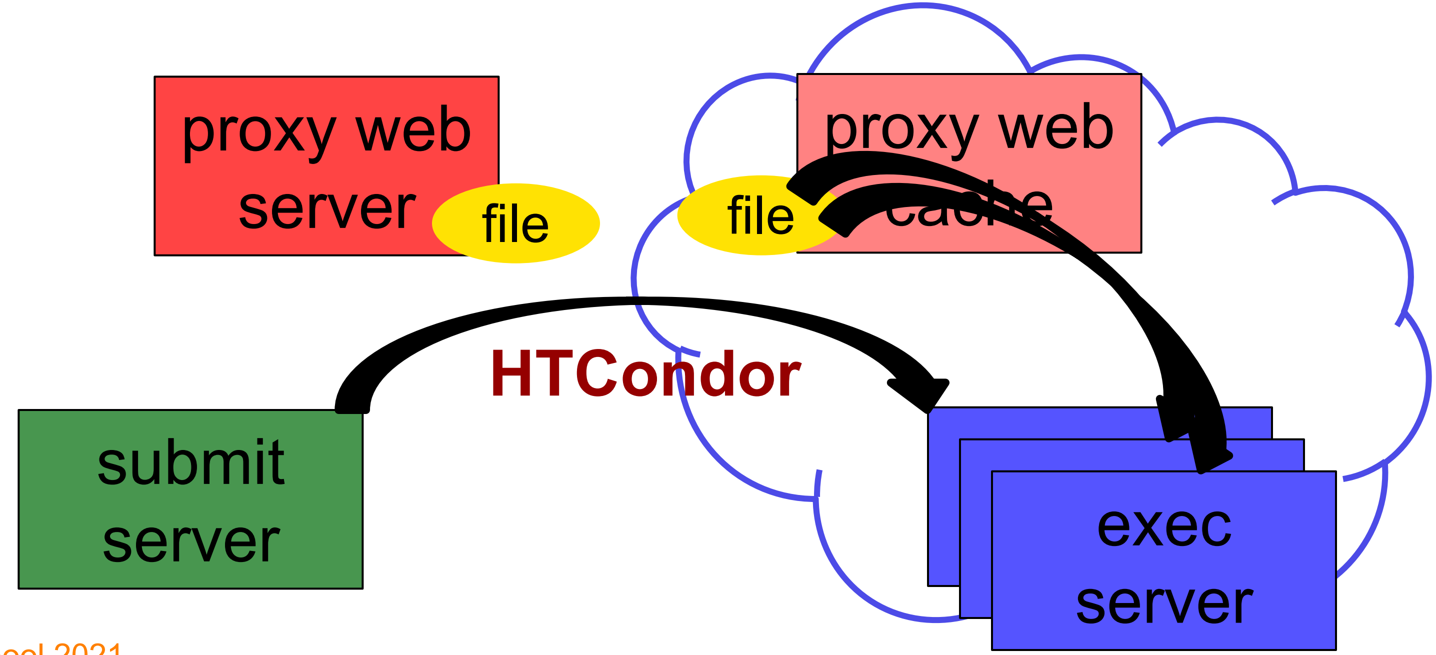
Using a Web Proxy

- Place the file onto a proxy-configured web server
- Have HTCondor download via HTTP address



Using a Web Proxy

- Place the file onto a proxy-configured web server
- Have HTCondor download via HTTP address



Downloading HTTP Files

- HTCondor submit file:

```
transfer_input_files=http://host.univ.edu/path/to/shared.tar.gz
```

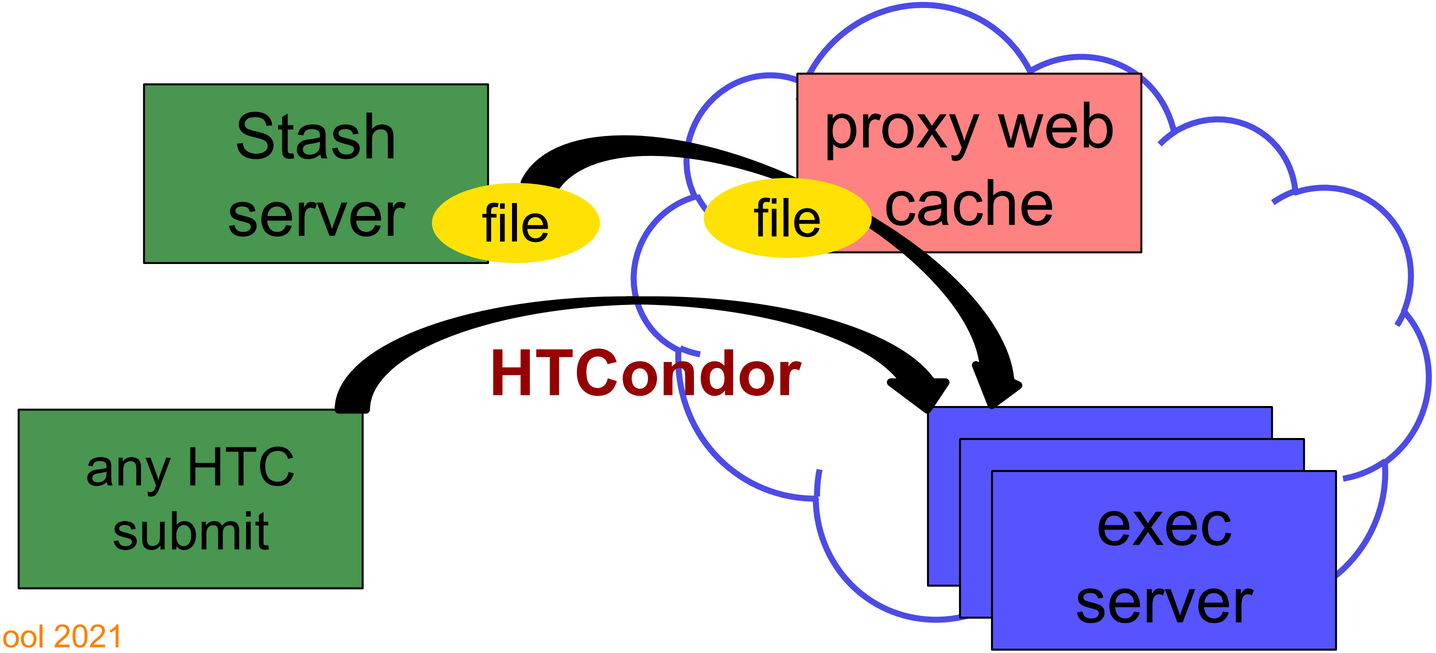
- Virtually any host or existing web server but ensure multiple downloads are permissible.

Web Proxy Considerations

- Memory limited, **max file size: 1 GB**
- Local caching at OSG sites
  - good for *shared* input files
  - perfect for software and common input
  - renaming changed files recommended
- Files are downloadable by **ANYONE** who has the specific HTTP address
  - Will work on 100% of OSG sites, though not all sites will have a local cache

In the OSG (Ex. 2.1)

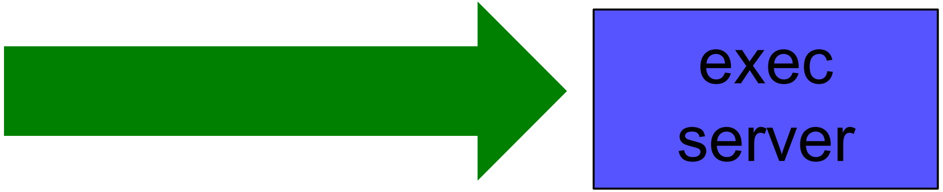
- place files in `/public/username/`
- address: `http://stash.osgconnect.net/public/user/shared.tar.gz`



Handling Data on OSG

- ~~Overview / Things to Consider~~
- ~~HTCondor File Transfer~~
- ~~Web Proxy~~
- **Stash**
- Shared File Systems

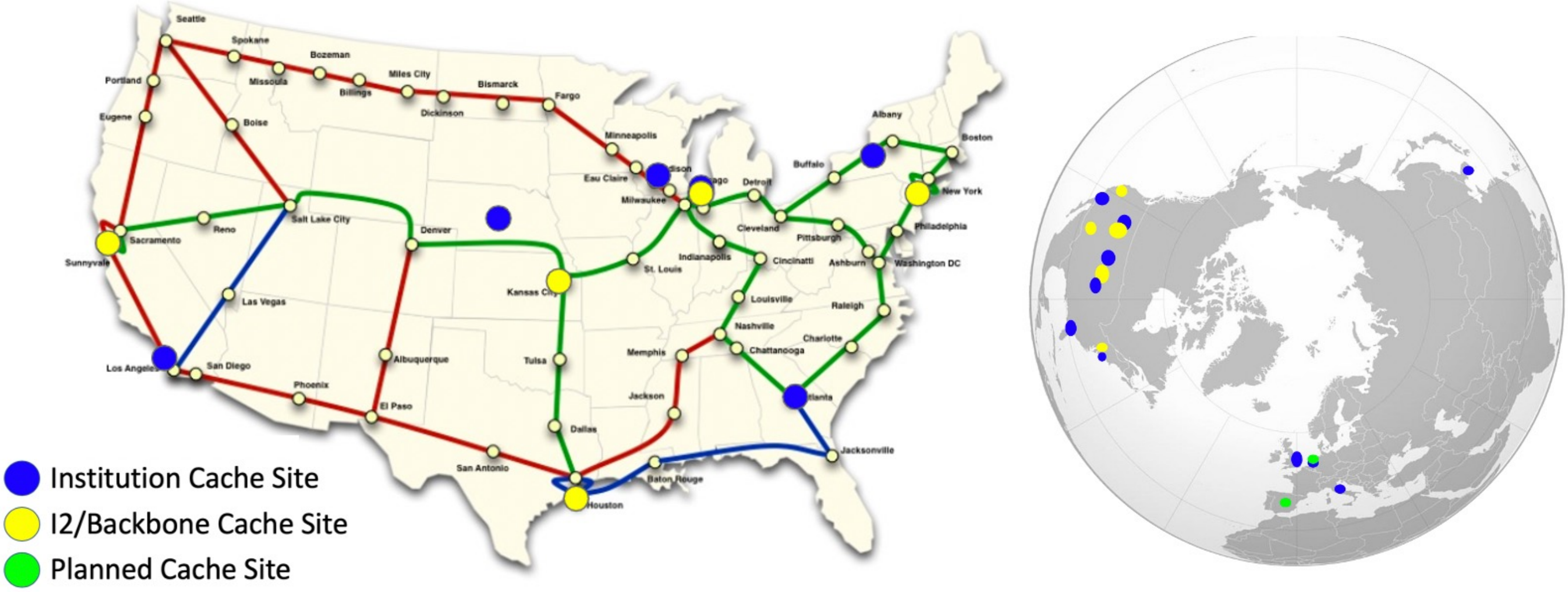
Large input in HTC and OSG



file size	method of delivery
words	within executable or arguments?
tiny – 100MB per file	HTCondor file transfer (up to 1GB total per-job)
100MB – 1GB, shared	download from web server (local caching)
1GB – 20GB, unique or shared	Stash (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

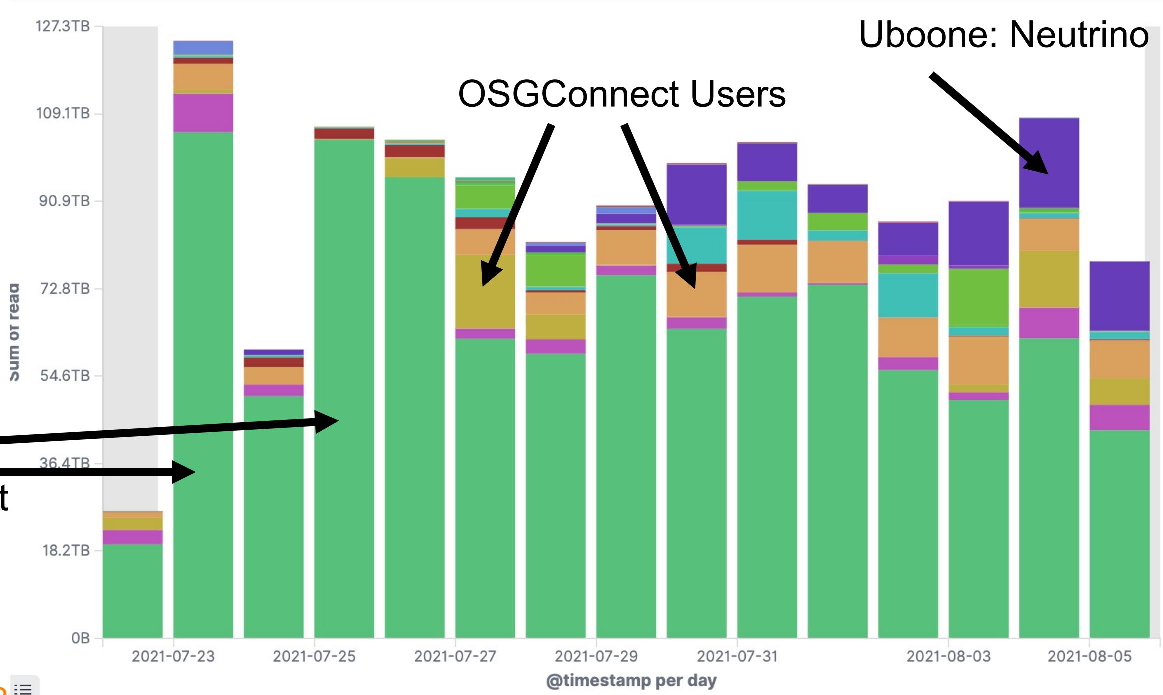
Using Stash for Input

- regionally-cached repository managed by OSG Connect



Stash Usage on OSG

- Lots of experiments use Stash



Uboone: Neutrino

OSGConnect Users

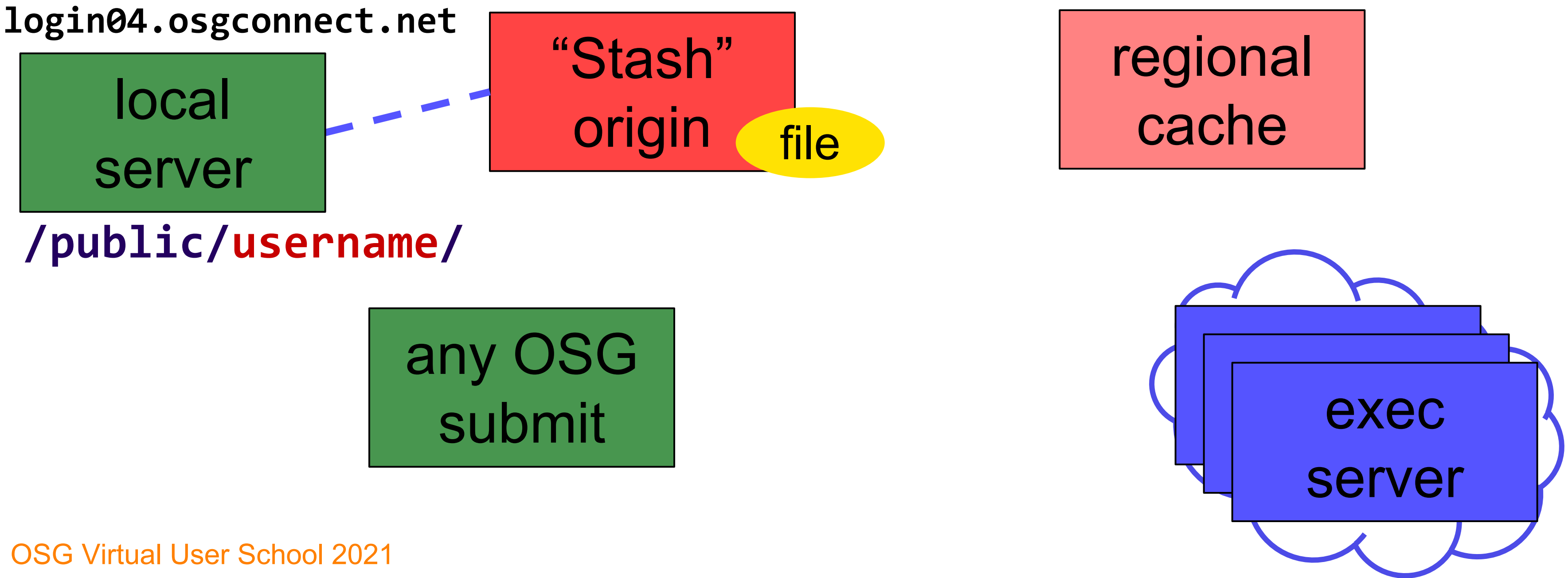
Minerva:
Neutrino Experiment

Stash Considerations

- Available at ~90% of OSG sites
- Regional caches on *very fast* networks
  - **Recommended max file size: 20 GB**
  - *shared* OR *unique* data
- Can copy multiple files totaling >10GB
- Just like HTTP proxy, change name when update files

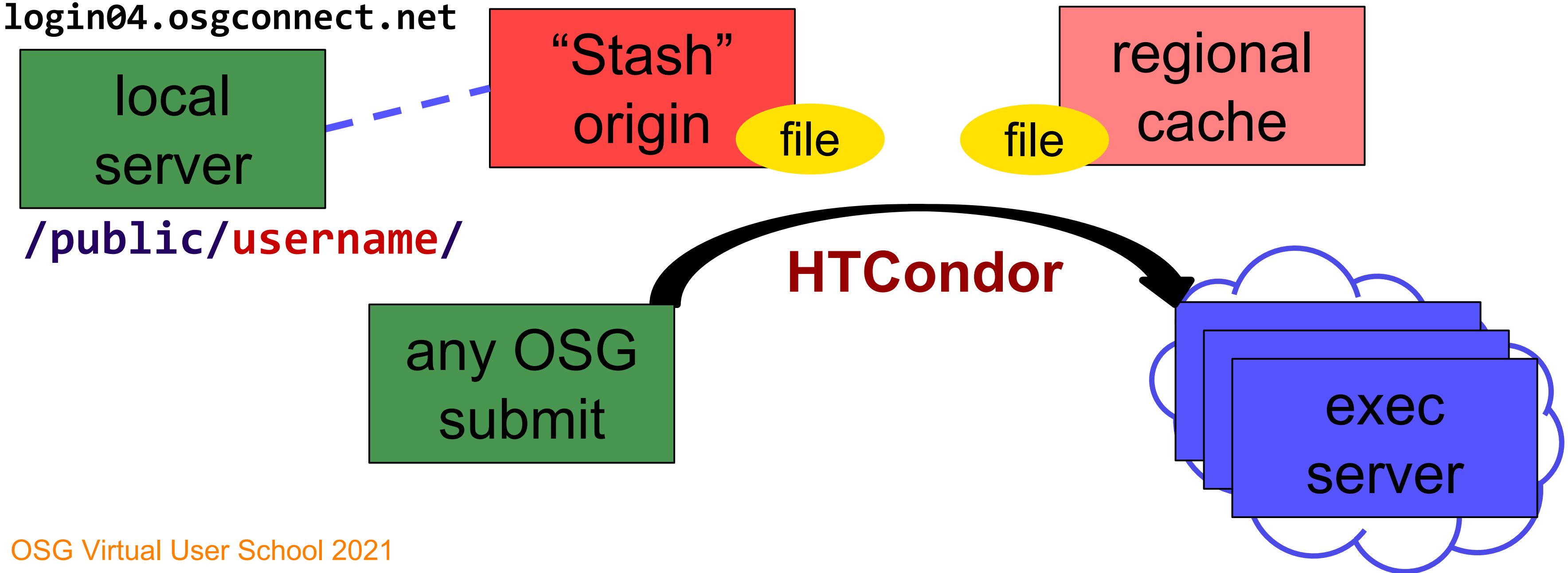
Placing Files in Stash

- Place files in `/public/username/` on `osgconnect.net`



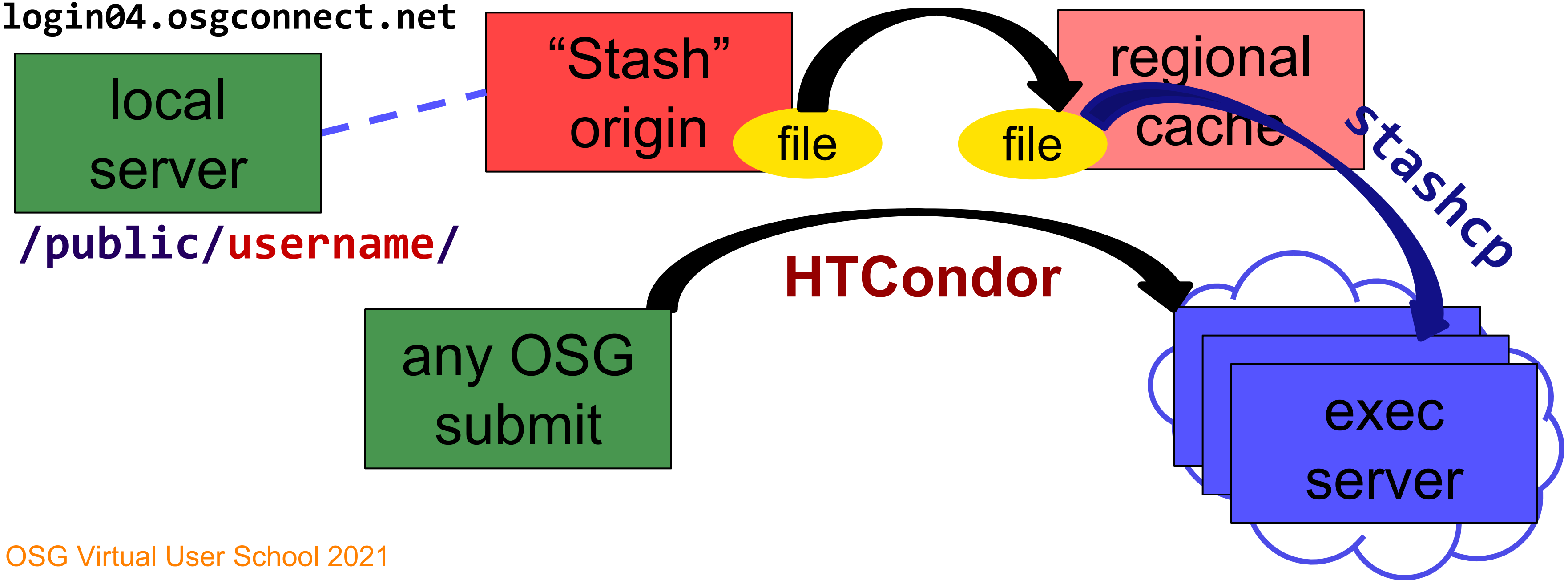
Obtaining Files in Stash

- Use HTCondor transfer for other files



Obtaining Files in Stash

- Download using `stashcp` command (available as an OASIS software module)



In the Submit File

- Require StashCashe sites in the submit file

```
+WantsStashCache = true
```

- Require sites with OASIS modules (for `stashcp`)

```
Requirements = <OTHER REQUIREMENTS>
                           && (HAS_MODULES =?= true)
```

In the Job Executable

```
#!/bin/bash
# setup:
module load stashcache
stashcp /public/username/file.tar.gz ./

<untar, then remove the tarball>
<job commands>

<remove all files from Stash>
# END
```

What’s Different for Output?

- always unique (right?), so caching won’t help
- files not associated with your local username
  - security barriers outside of local context
- security issues with world-writability
  - (versus okay world-readability for input)

Output for HTC and OSG

exec
server

amount	method of delivery
~~words~~	~~within executable or arguments?~~
tiny – **1GB, total**	HTCondor file transfer
1GB - 20GB, unique or shared	Stash
20GB+, total	shared file system (local copy, local execute servers)

Output for HTC and OSG

exec
server

amount	method of delivery
~~words~~	~~within executable or arguments?~~
tiny – **1GB, total**	HTCondor file transfer
1GB – 20GB, unique or shared	Stash
20GB+, total	shared file system (local copy, local execute servers)

Writing to Stash

In the submit file:

- Require StashCashe sites in the submit file

```
+WantsStashCache
```

- Require sites with OASIS modules (for `stashcp`)

```
Requirements = <OTHER REQUIREMENTS> && (HAS_MODULES =?= true)
```

In the job (wrapper script):

- Use `stashcp` within the job to transfer desired output

```
stashcp output.dat stash:///osgconnect/public/username/
```

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

file size	method of delivery
words	within executable or arguments?
tiny – 10MB per file	HTCondor file transfer (up to 1GB total per-job)
10MB – 1GB, shared	download from web server (local caching)
1GB - 10GB, unique or shared	Stash (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

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

file size	method of delivery
words	within executable or arguments?
tiny – 10MB per file	HTCondor file transfer (up to 1GB total per-job)
10MB – 1GB, shared	download from web server (local caching)
1GB - 10GB, unique or shared	Stash (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

Open Science Grid

Cleaning Up Old Data

- For Stash *AND* web proxies:

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

- Stash and VO-managed web proxy servers do NOT have unlimited space!
  - Some may regularly clean old data for you. Check with local support.

Handling Data on OSG

- ~~Overview / Things to Consider~~
- ~~HTCondor File Transfer~~
- ~~Web Proxy~~
- ~~Stash~~
- **Shared File Systems**

(Local) Shared Filesystems

- data stored on file servers, but network-mounted to local submit and execute servers
- Available on **some** submit servers
  - CHTC ✓ **Yes**
  - OSG Connect ✗ **No**

More details at the end of this presentation…

Open Science Grid

Filesystem Quotas

System	Location	Quota	Transfer Mechanism
CHTC	`/home`	20 GB	HTCondor file transfer
	`/staging`	20 GB 20 files total	Accessed directly from within job
OSG Connect	`/home`	50 GB	HTCondor file transfer
	`/public`	500 GB	Web Proxy, **`stashcp`**

Tips:

- Choose data location and transfer carefully based on the size and type of the data
- Remove unnecessary files
- Configure workflow to discard unneeded intermediate files

To request increases contact:

- CHTC: **chtc@cs.wisc.edu**
- OSG Connect: **support@osgconnect.net**

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 submit node	via HTCondor submit file	anywhere HTCondor jobs can run
Web proxy	Shared input only	1 GB/file	Service specific - OSGConnect in `/public/user/`	HTTP download	anywhere, by anyone
Stash	Both	20 GB/file	via OSG Connect submit server	via stashcp command (and module)	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)

Required Exercises

- 1.1 Understanding a job’s data needs
- 1.2 Using data compression with HTCondor file transfer
- 1.3 Splitting input (prep for large run in 2.1)

- 2.1 Using a web proxy for shared input
  - place the blast database on the web proxy
- 2.2 Stash for shared input
  - place the blast database in Stash
- 2.3 Stash for unique input
  - convert movie files

Bonus Exercises

- 3.1 Shared Filesystem for Large Input
- 3.2 Shared Filesystem for Large Output

Acknowledgments

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

Open Science Grid

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)

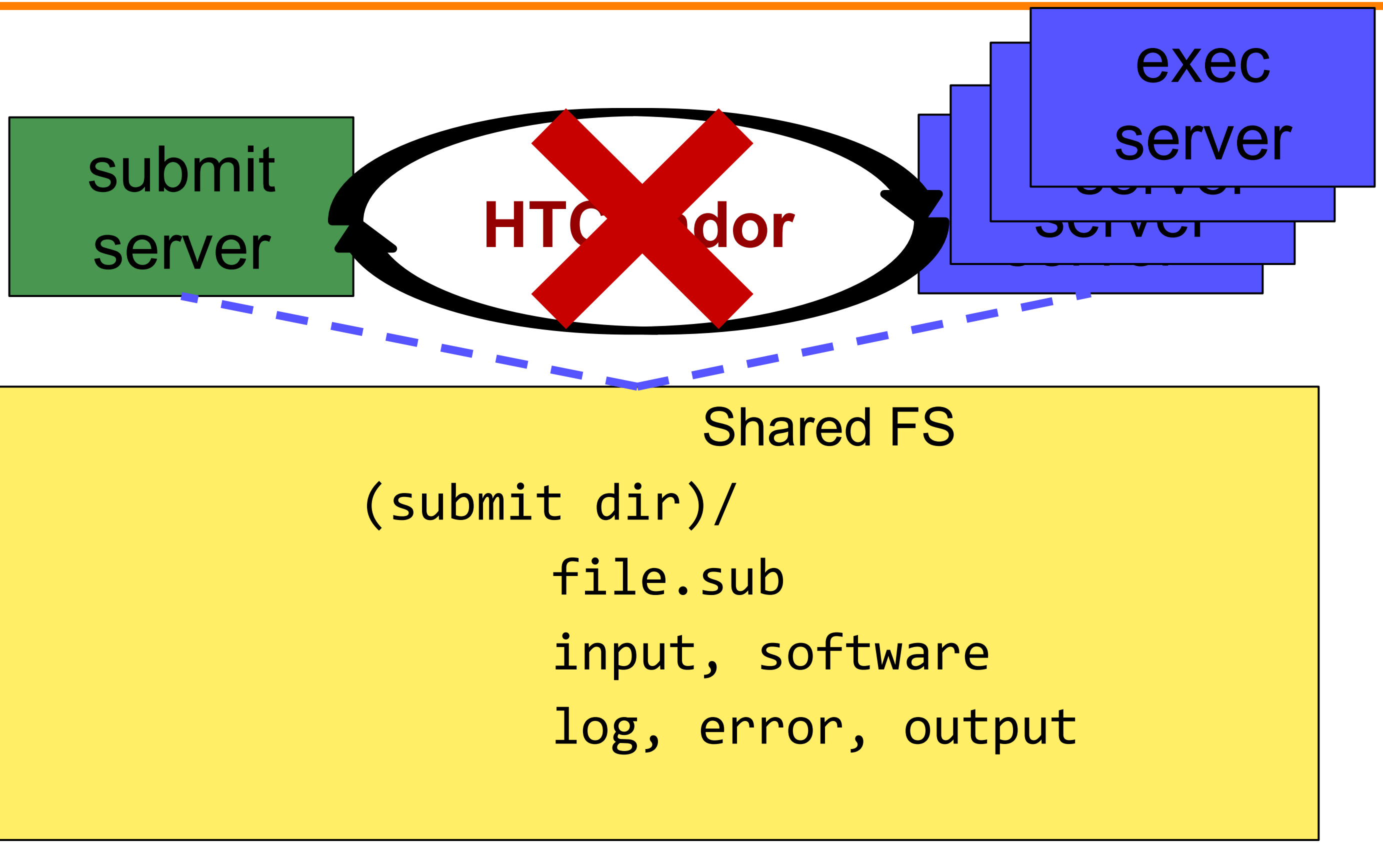
Shared FS Technologies

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

Shared FS Configurations

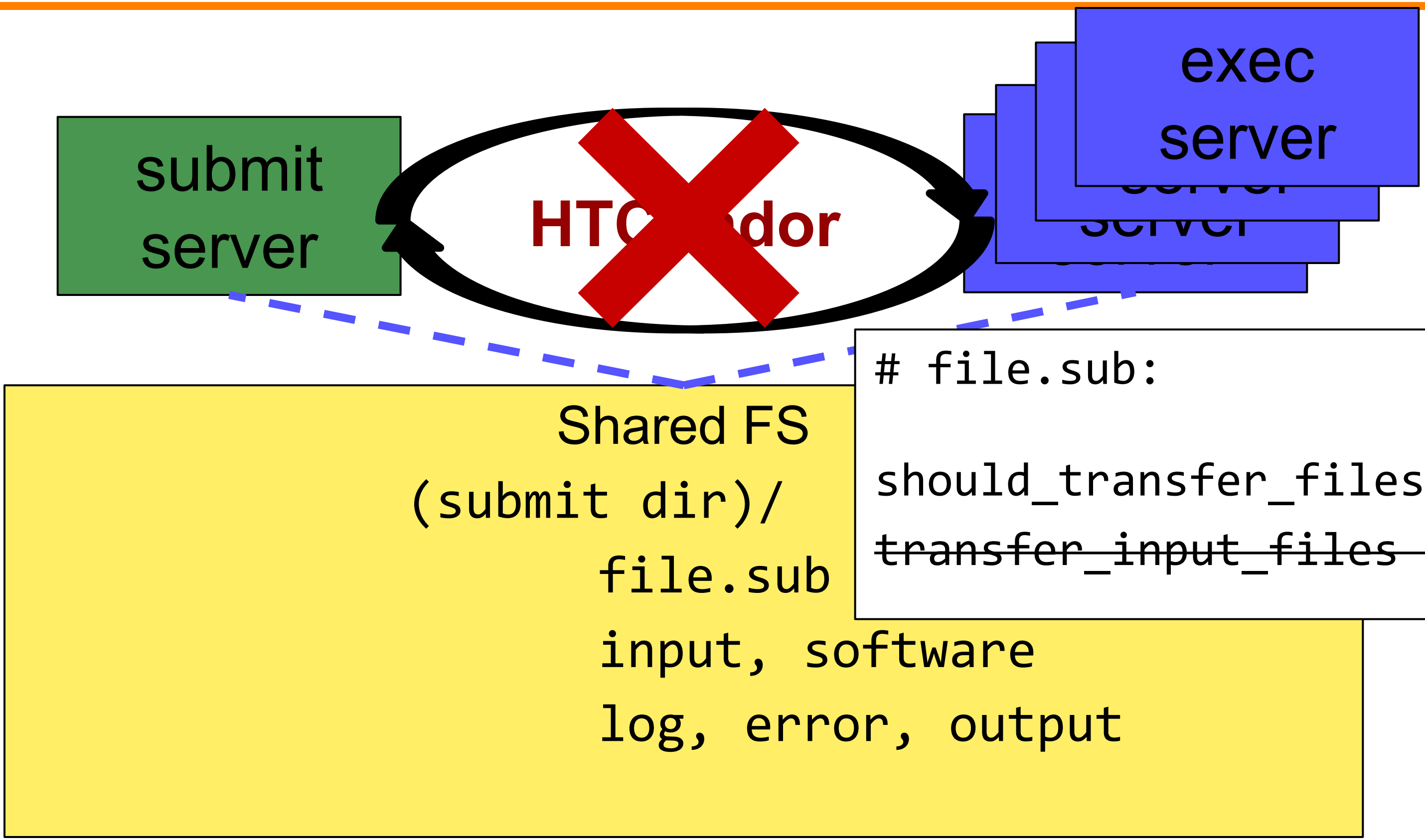
1. Submit directories *WITHIN* the shared filesystem
   - most campus clusters
   - limits HTC capabilities!!
2. Shared filesystem separate from local submission directories
   - supplement local HTC systems
   - treated more as a repository for VERY large data (>GBs)
3. Read-only (input-only) shared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



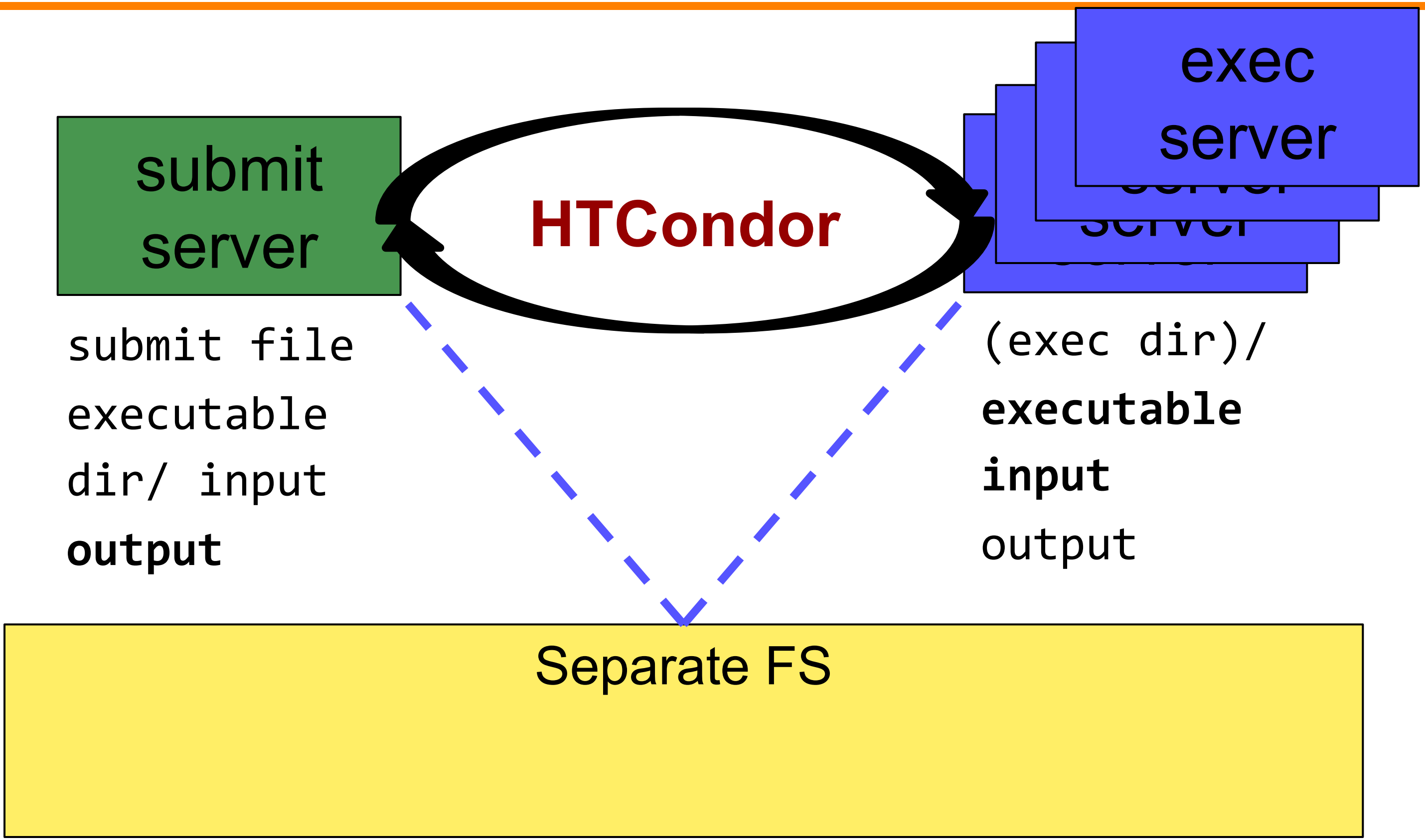
Shared FS

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

```
# file.sub:

should_transfer_files = NO
~~transfer_input_files =~~
```

Separate shared FS



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

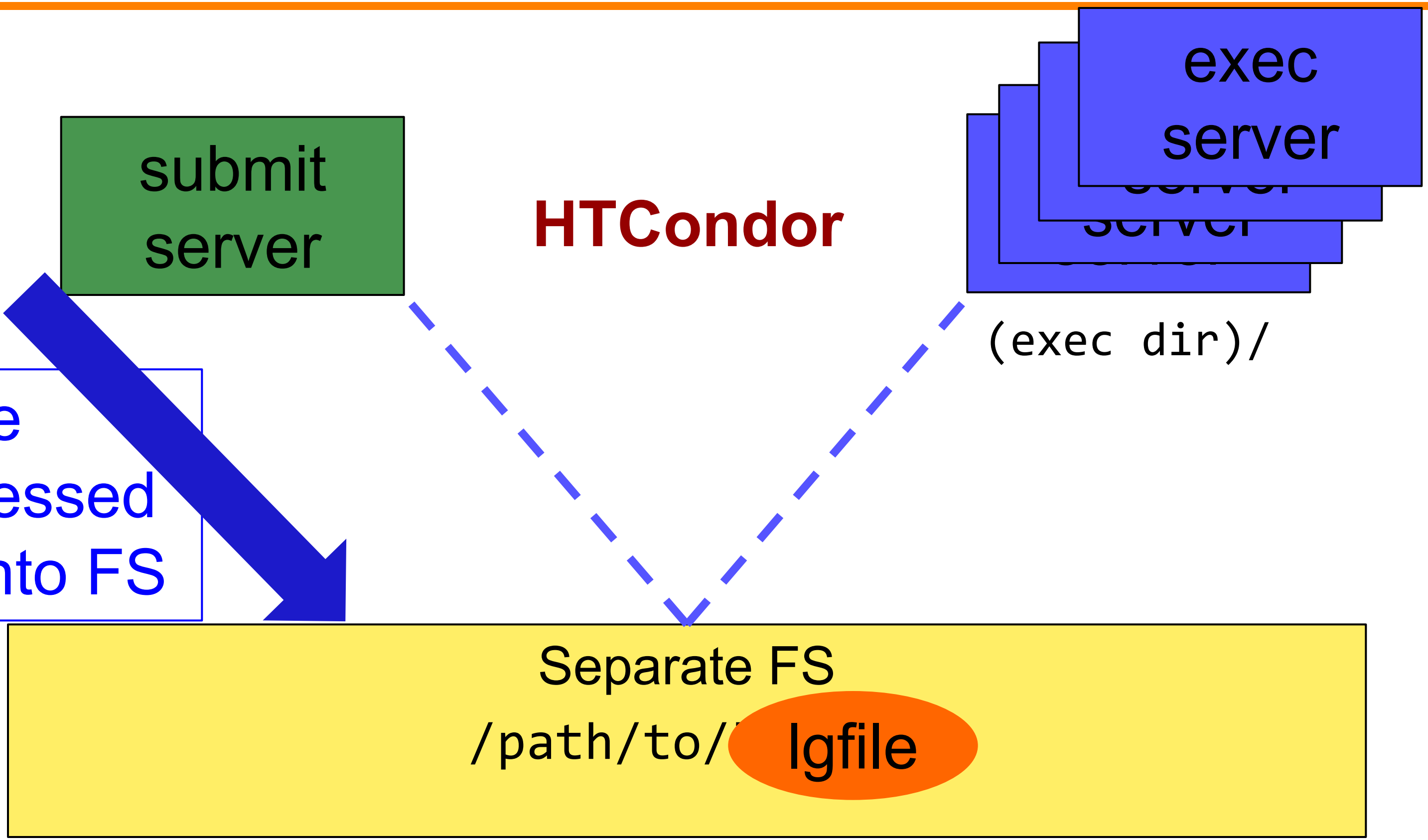
output

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

executable

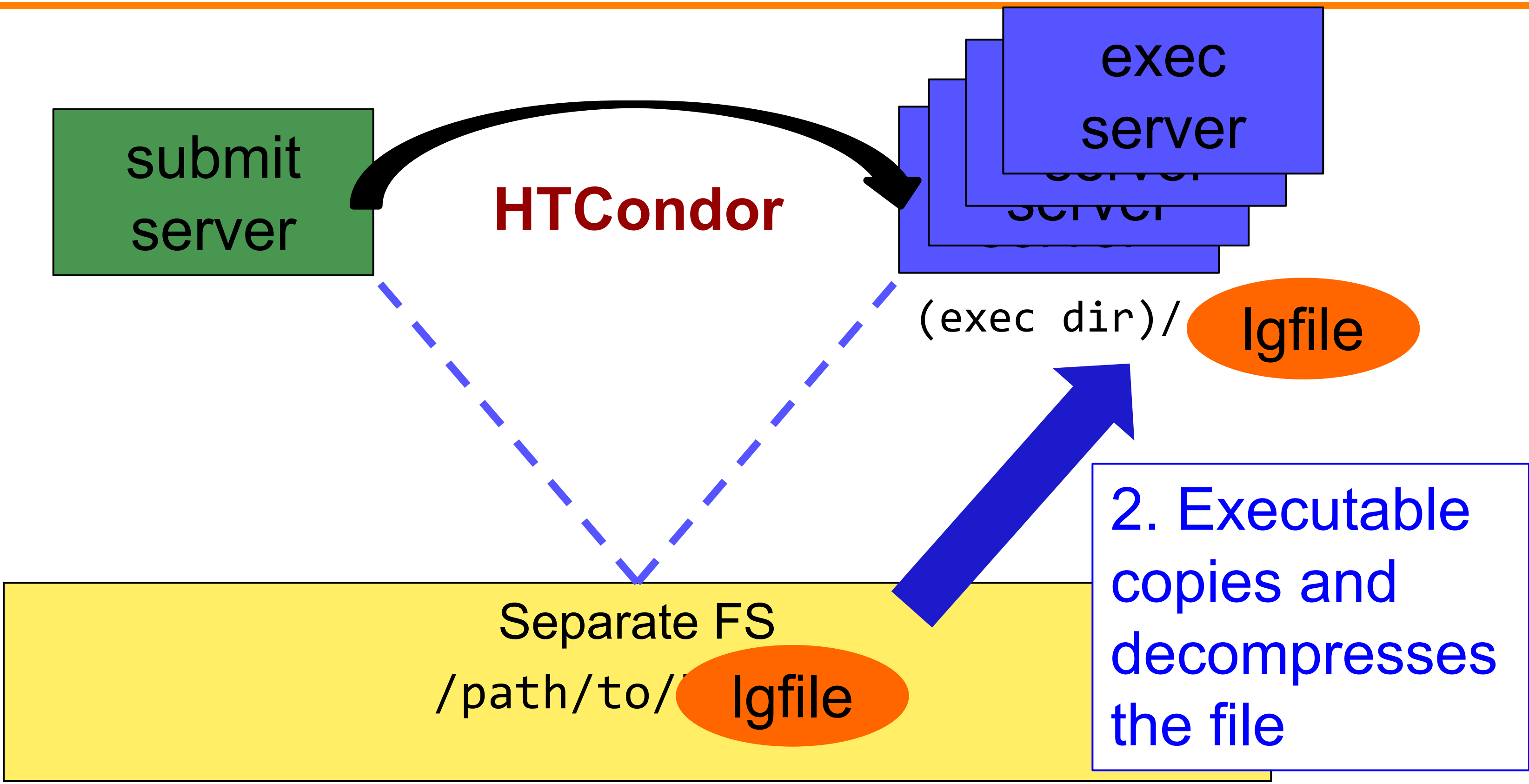
input

Separate shared FS - Input



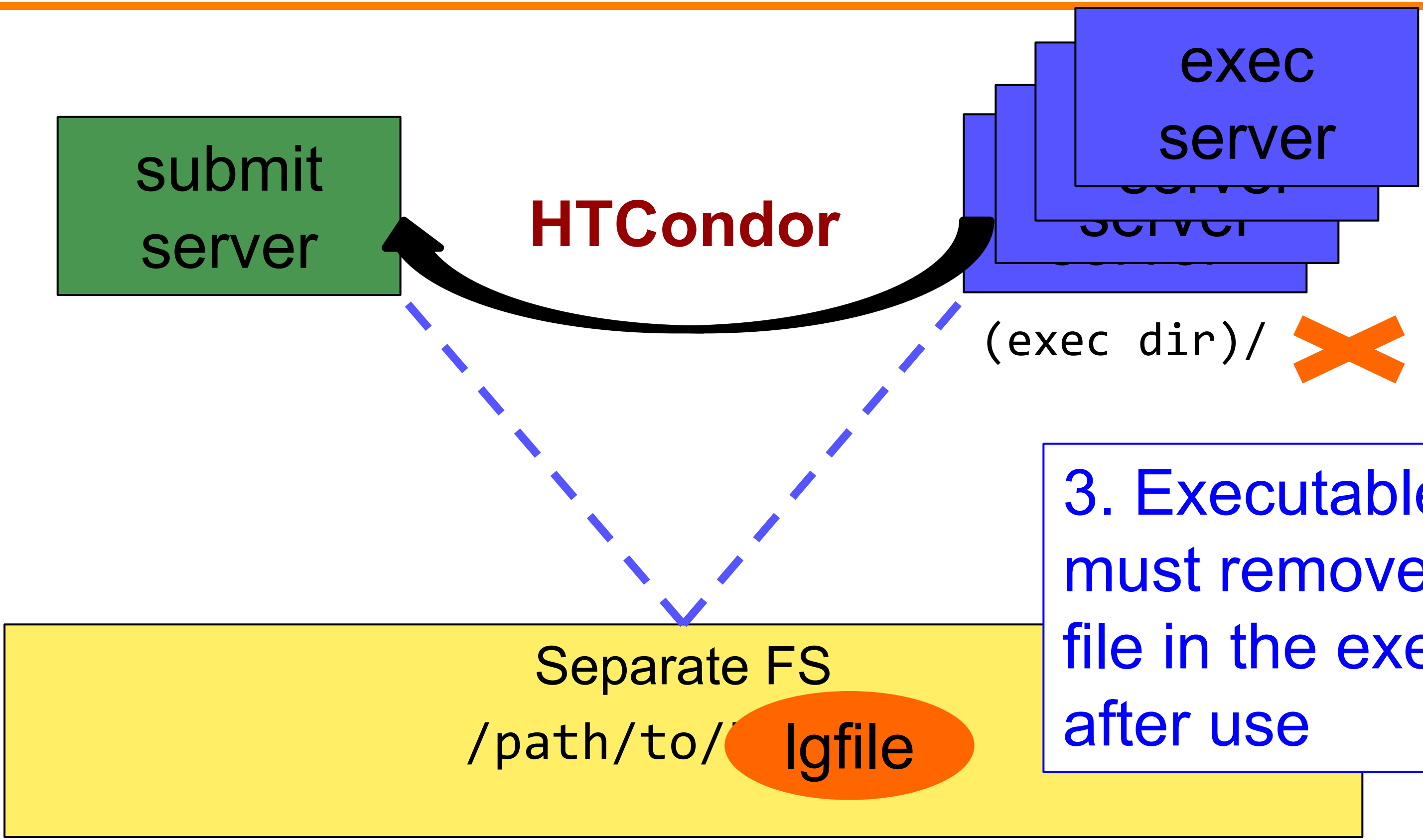
1. Place compressed input into FS

Separate shared FS - Input



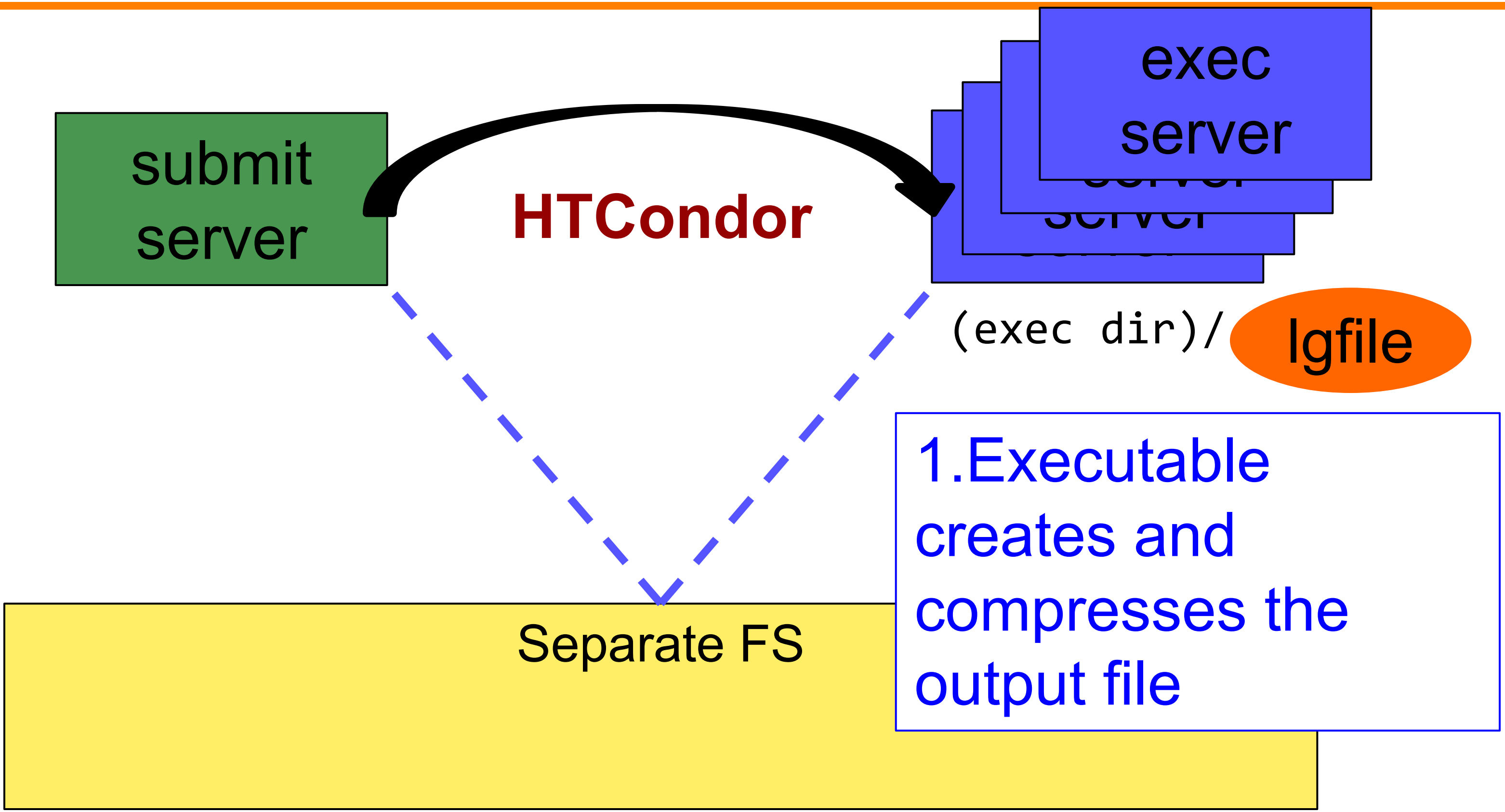
2. Executable copies and decompresses the file

Separate shared FS - Input



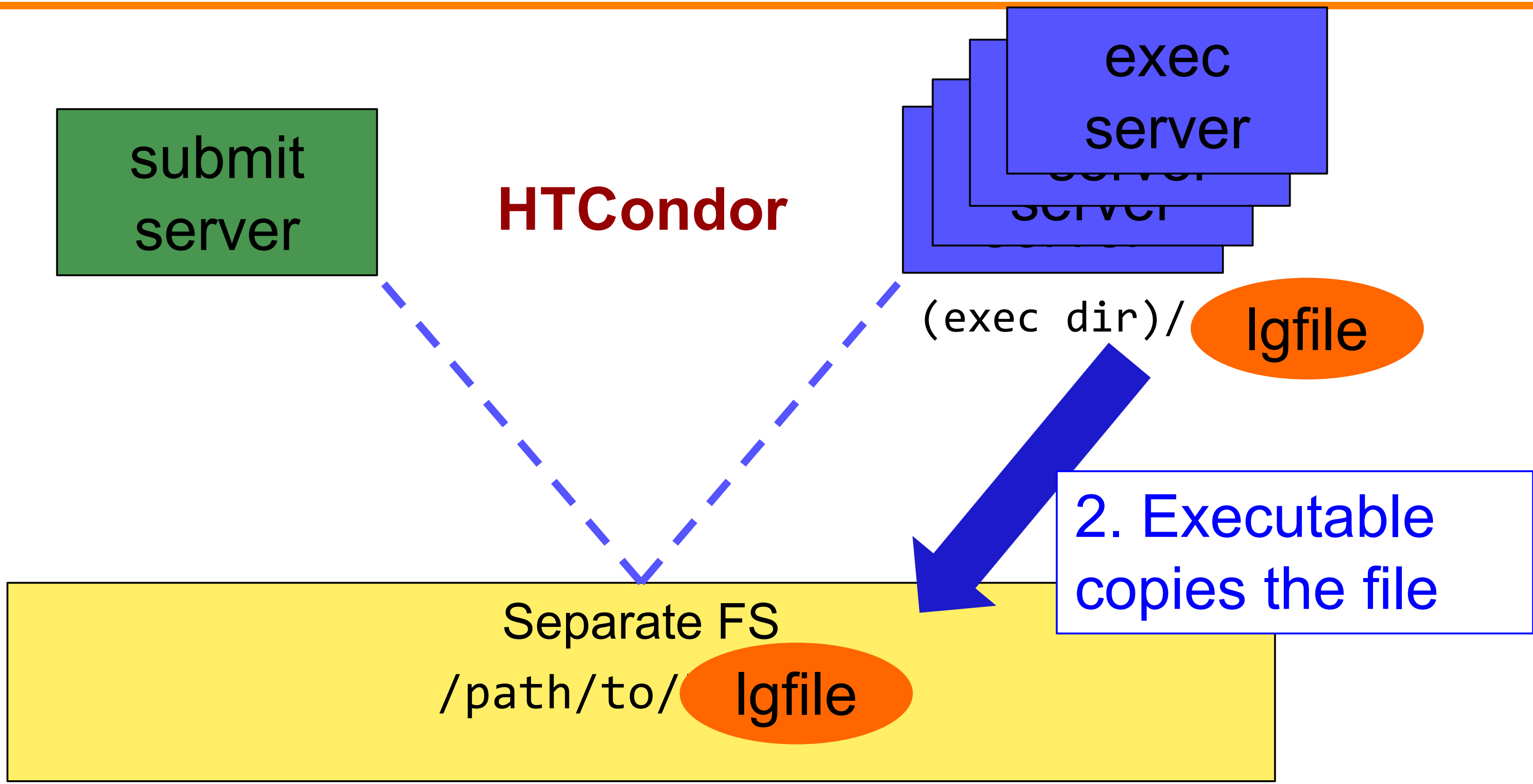
3. Executable must remove the file in the exec dir after use

Separate shared FS - Output



1. Executable creates and compresses the output file

Separate shared FS - Output



submit server

HTCondor

exec server

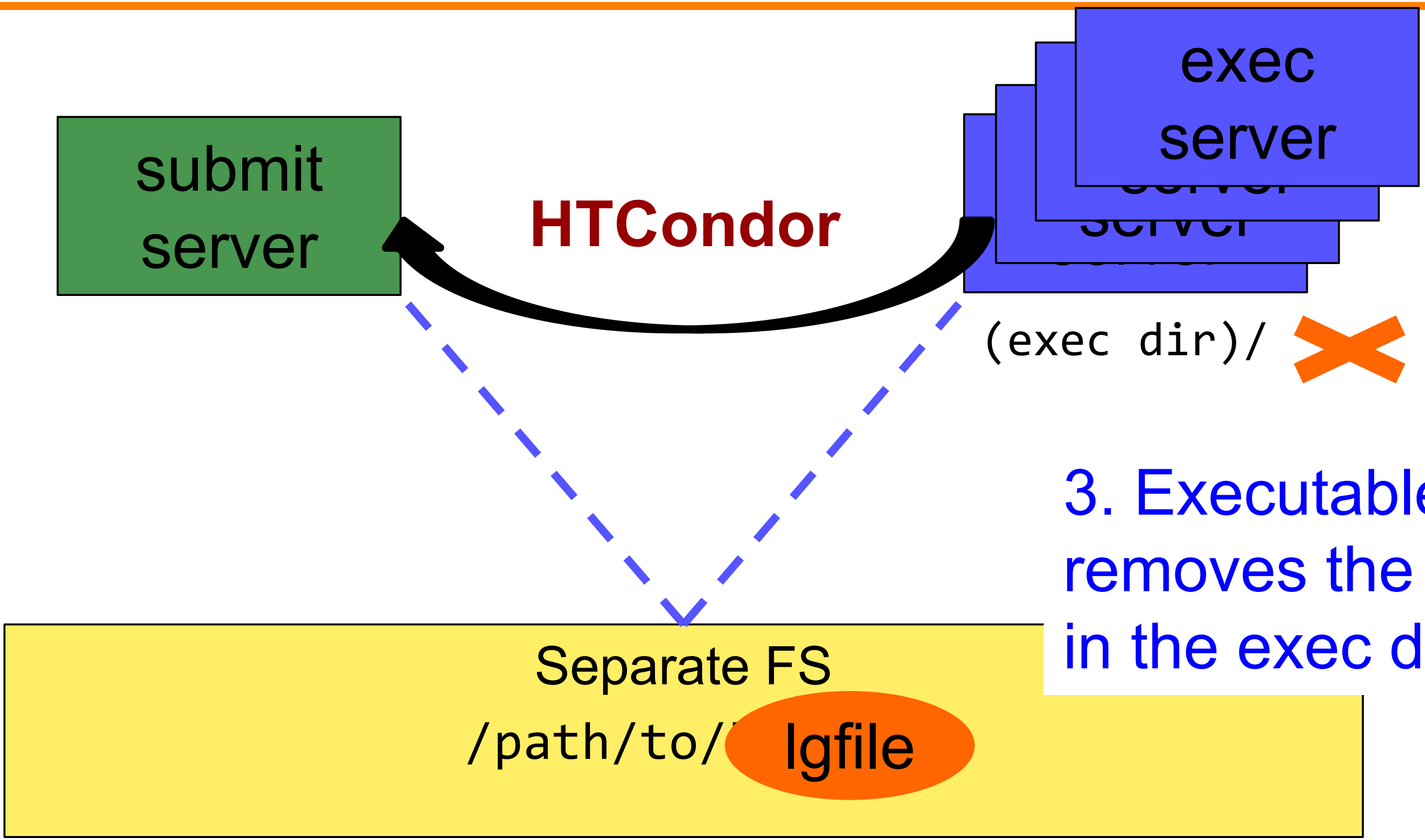
(exec dir)/ lgfile

Separate FS

/path/to/ lgfile

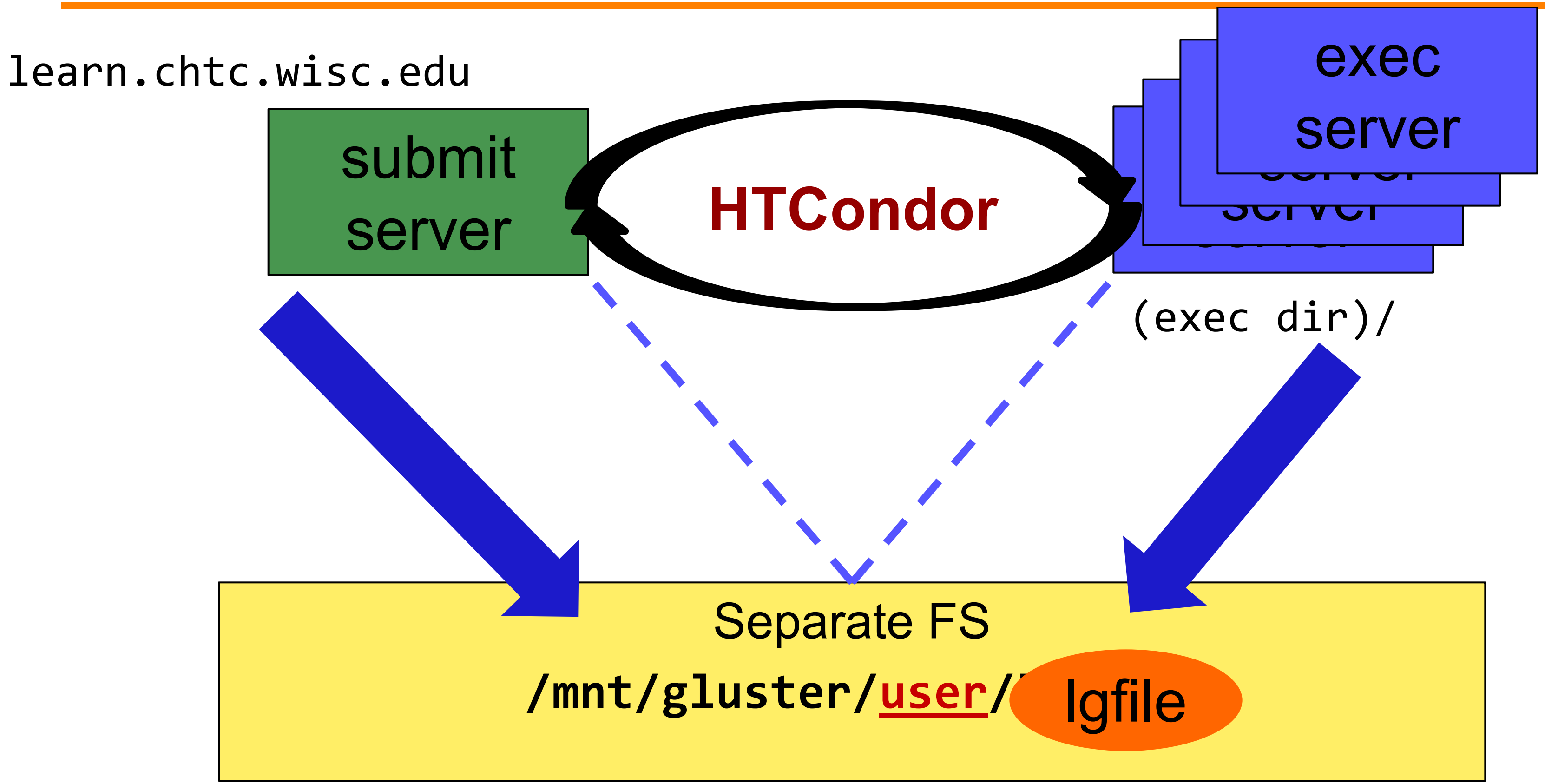
2. Executable copies the file

Separate shared FS - Output



3. Executable removes the file in the exec dir

At UW-Madison (Ex. 3.1-3.2)



learn.chtc.wisc.edu

submit server

HTCondor

exec server

(exec dir)/

Separate FS

/mnt/gluster/user/

lgfile