Open Science Grid

Data Considerations

Thursday AM, Lecture 2

Lauren Michael

CHTC, UW-Madison

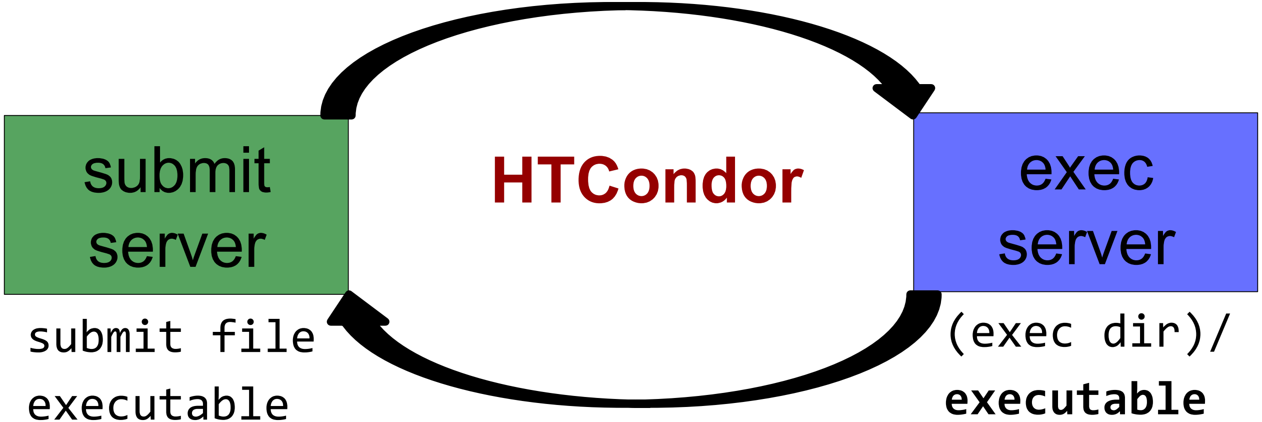
Overview – Data Handling

- Review of HTCondor Data Handling
- What is ‘Large’ Data?
- Data Management Tips
- Next talks: Dealing with Large Data

Overview – Data Handling

- **Review of HTCondor Data Handling**
- What is ‘Large’ Data?
- Data Management Tips
- Next talks: Dealing with Large Data

Review: HTCondor Data Handling



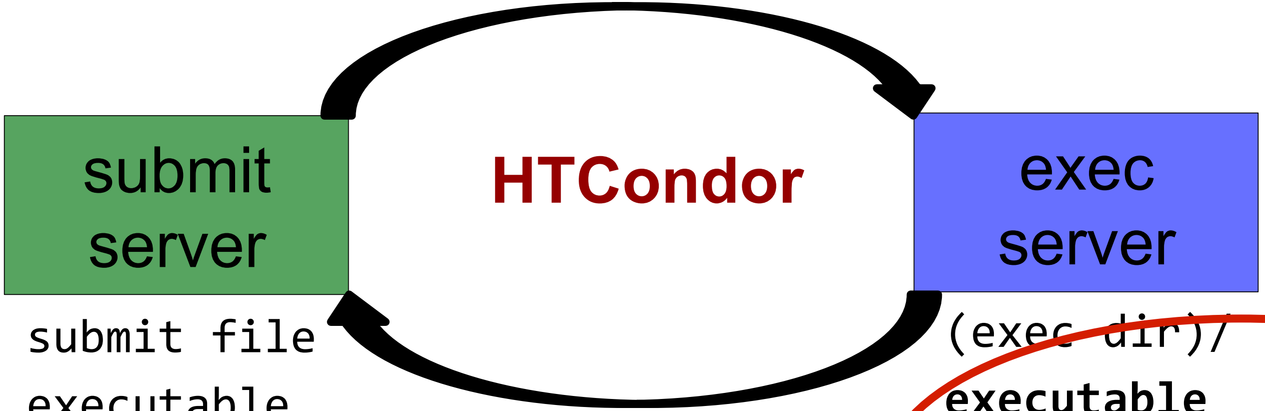
```
submit file
executable
dir/ input
output
```

output in bold under the submit server.

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

executable and **input** in bold under the exec server.

Server-specific data needs

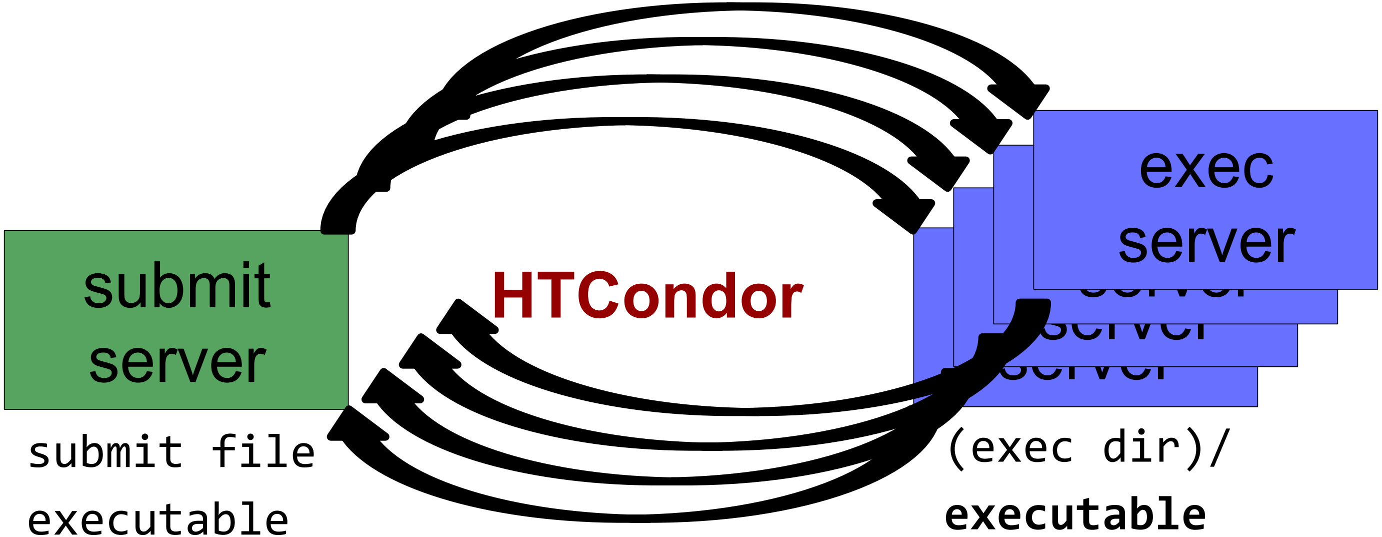


```
submit file
executable
dir/ input
```
`output (incl. err, out)`
```
log
```

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

informs request_disk

Network needs: the submit server bottleneck



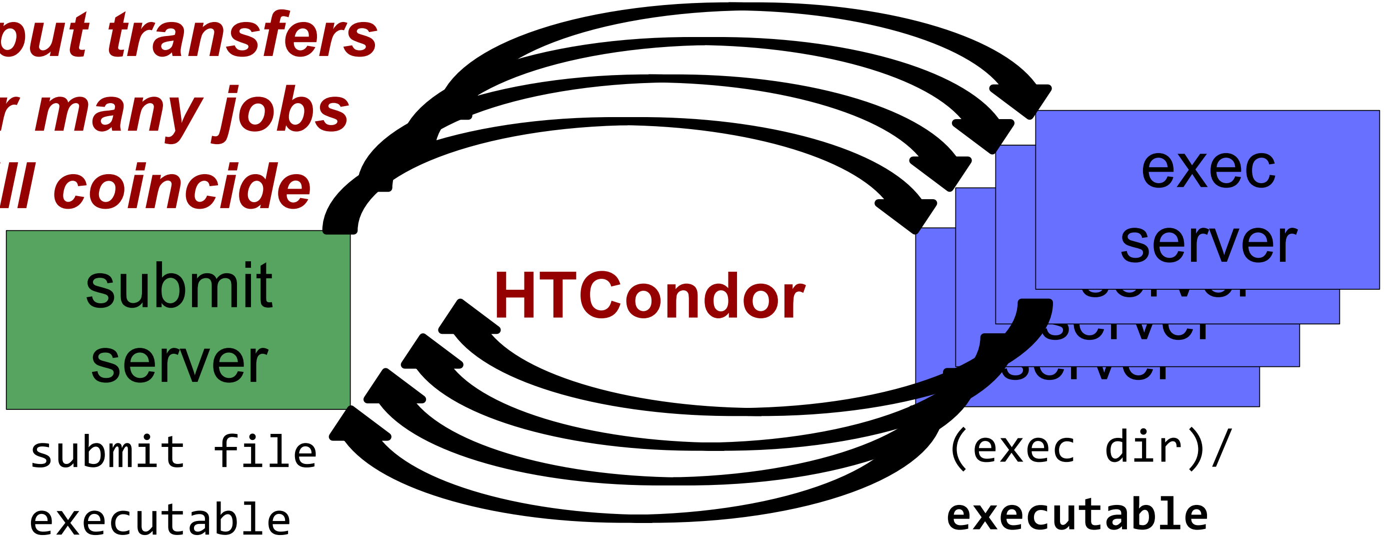
submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Open Science Grid

Network needs: the submit server bottleneck

Input transfers for many jobs will coincide



submit server

HTCondor

exec server

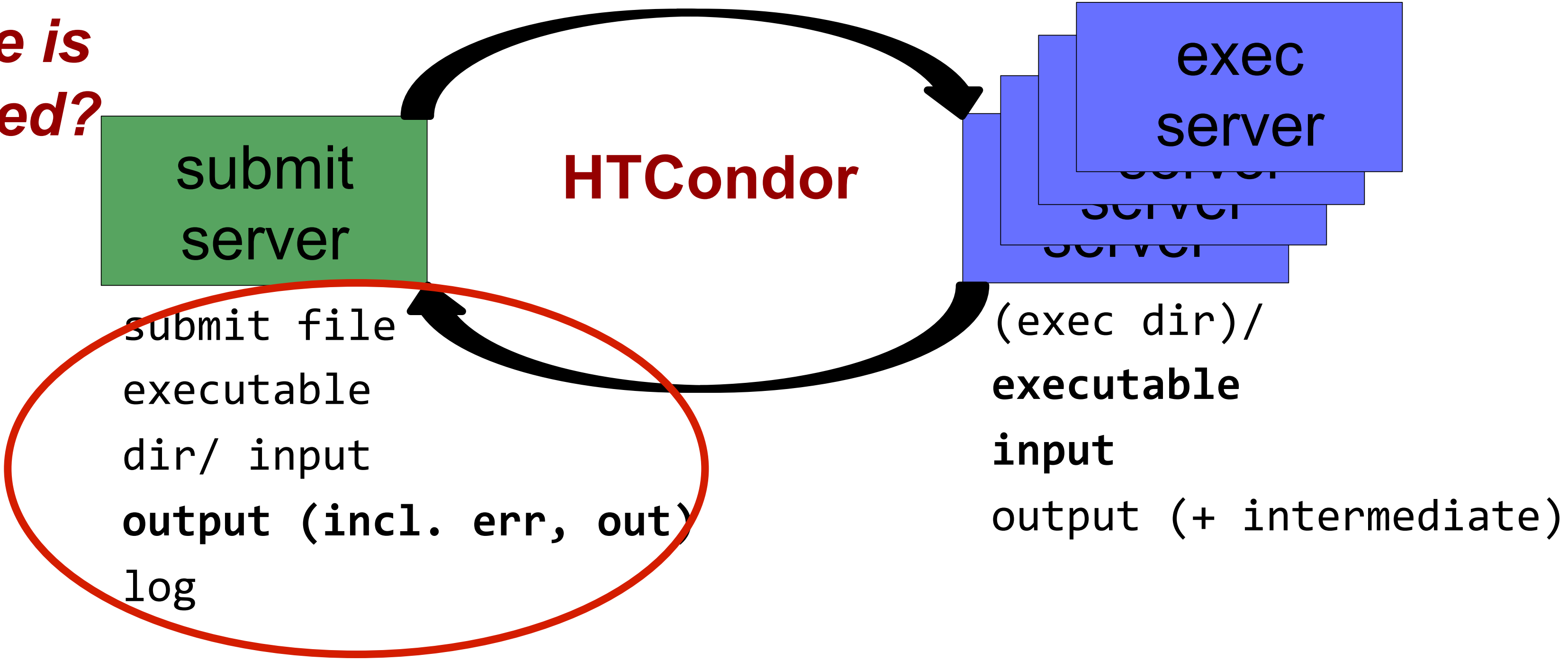
submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Output transfers are staggered

Server-specific data needs

What total disk space is needed?



```
submit file
executable
dir/ input
```
`output (incl. err, out)`
```
log
```

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

Overview – Data Handling

- Review of HTCondor Data Handling
- **What is ‘Large’ Data?**
- Data Management Tips
- Next talks: Dealing with Large Data

What is ~~big~~ large data?

- For researchers "big data" is relative
  - What is 'big' for you? Why?

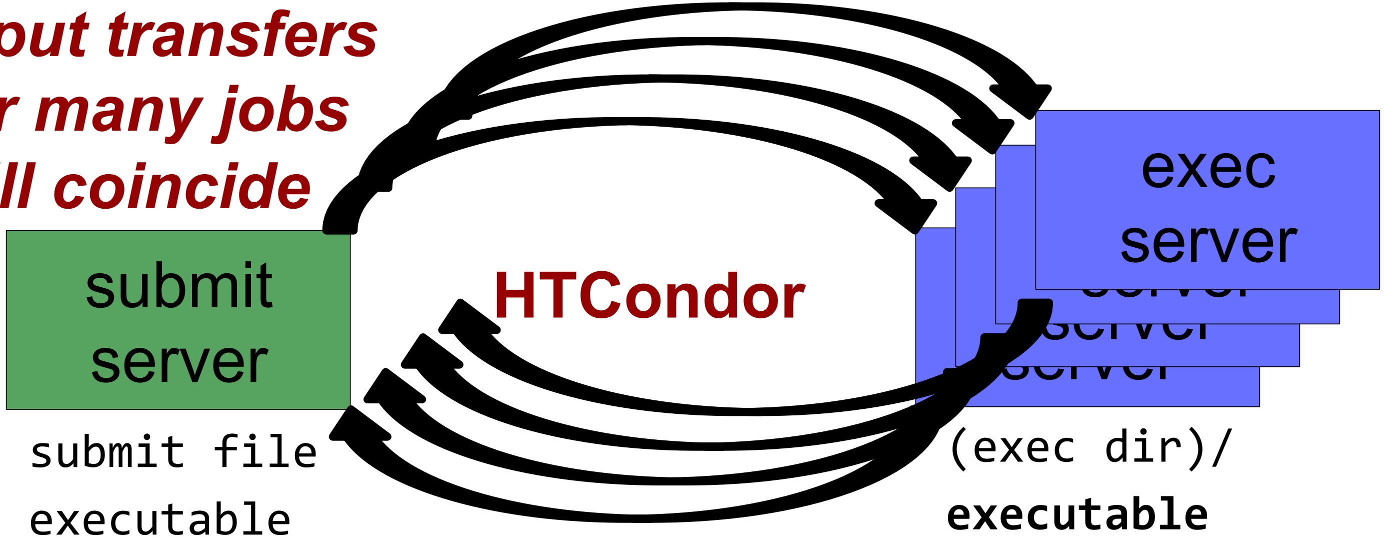
What is ~~big~~ large data?

- For researchers “big data” is relative
  - What is ‘big’ for you? Why?

- Volume, velocity, variety!
  - think: a million 1-KB files, versus one 1-GB file

HTCondor transfer limits

Input transfers for many jobs will coincide



submit server

HTCondor

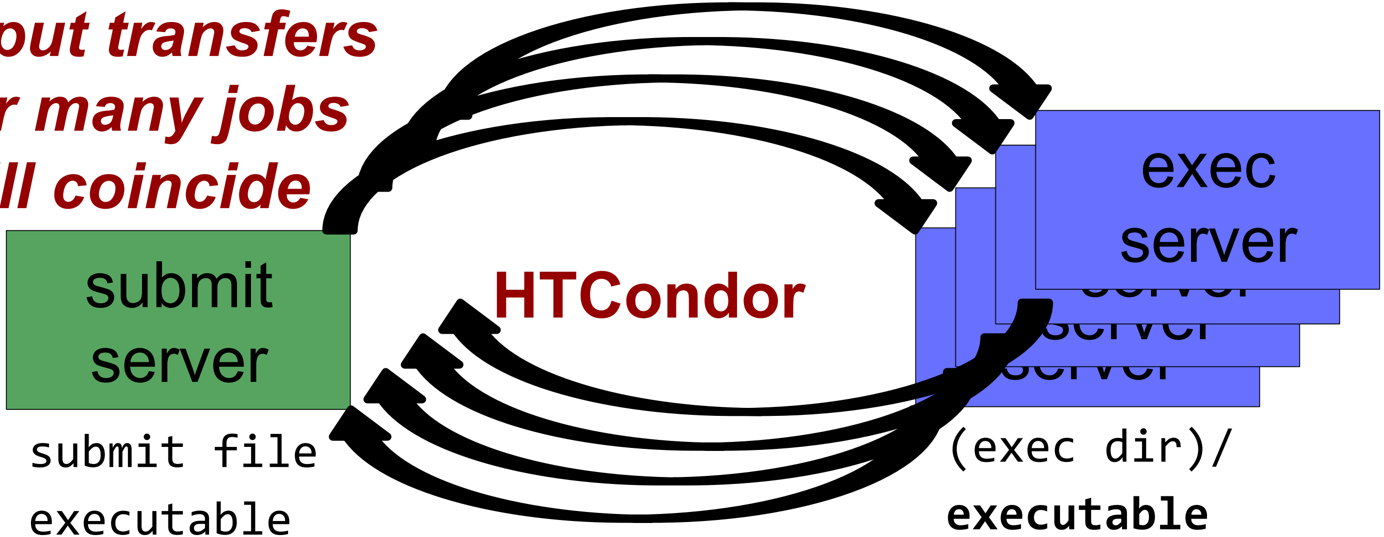
exec server

submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Output transfers are staggered

Per-job transfer limits



Input transfers for many jobs will coincide

submit file
executable
dir/ input
output

(exec dir)/
executable
input
output

Output transfers are staggered

‘Large’ input: The collaborator analogy

‘Large’ input: The collaborator analogy

amount	method of delivery
words	
tiny – 10MB	
10MB – GBs	
TBs	

‘Large’ input: The collaborator analogy

amount	method of delivery
words	email body
tiny – 10MB	
10MB – GBs	
TBs	

‘Large’ input: The collaborator analogy

amount	method of delivery
words	email body
tiny – 10MB	email attachment (managed transfer)
10MB – GBs	
TBs	

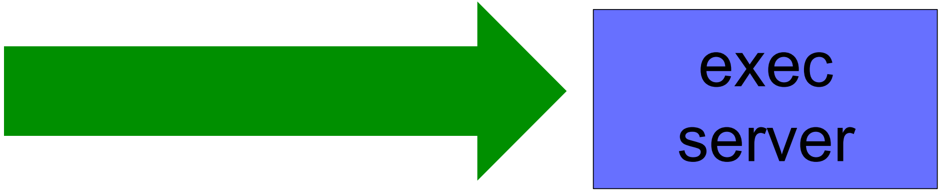
‘Large’ input: The collaborator analogy

amount	method of delivery
words	email body
tiny – 10MB	email attachment (managed transfer)
10MB – GBs	download from Google Drive, Drop/Box, other web-accessible server
TBs	

‘Large’ input: The collaborator analogy

amount	method of delivery
words	email body
tiny – 10MB	email attachment (managed transfer)
10MB – GBs	download from Google Drive, Drop/Box, other web-accessible server
TBs	ship an external drive (local copy needed)

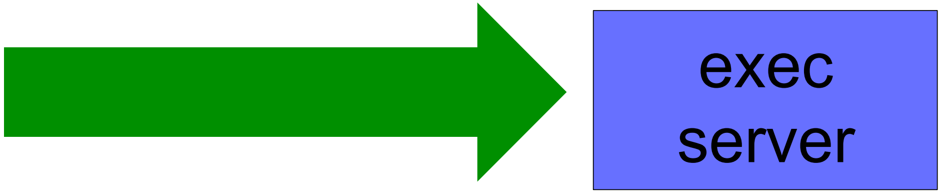
Large input in HTC and OSG



exec server

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 proxy (network-accessible server)
1GB - 10GB, unique or shared	StashCache (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

Large input in HTC and OSG

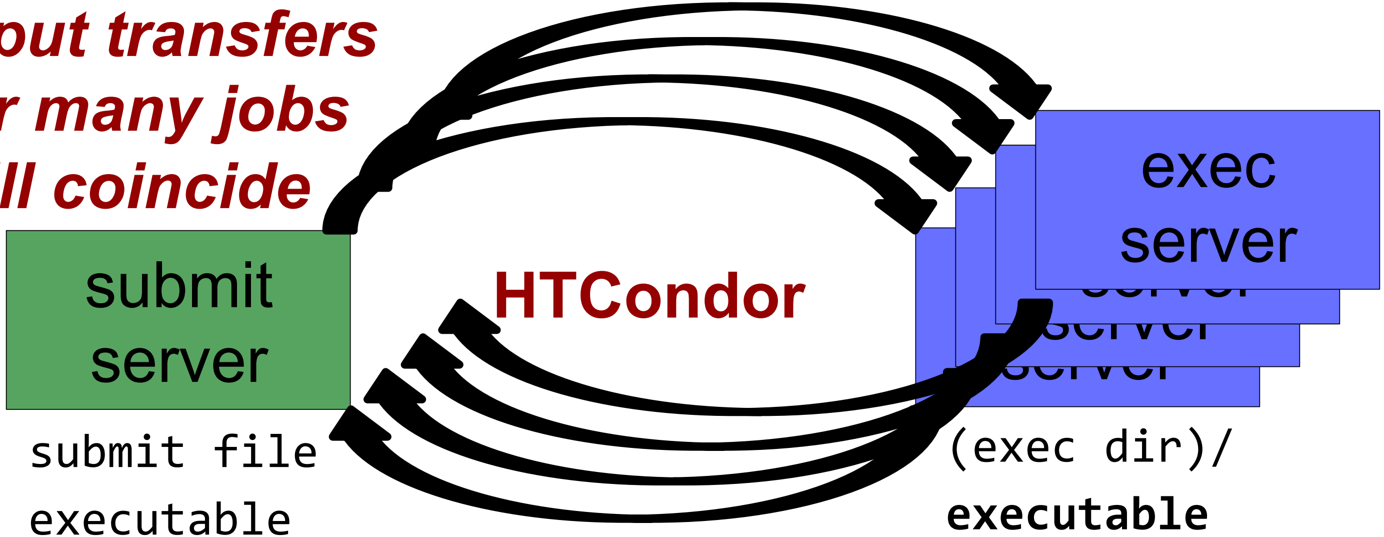


exec server

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 proxy (network-accessible server)
1GB - 10GB, unique or shared	StashCache (regional replication)
10 GB - TBs	shared file system (local copy, local execute servers)

HTCondor transfer limits

Input transfers for many jobs will coincide

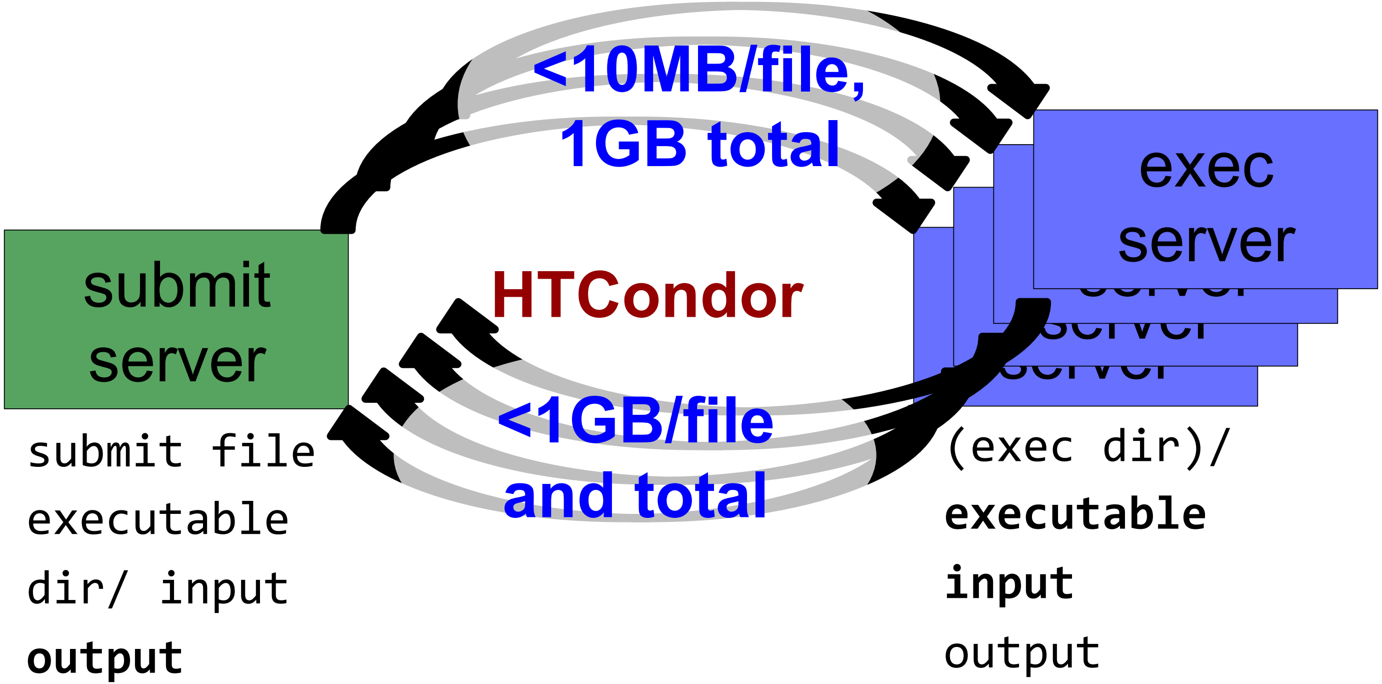


submit file
executable
dir/ input
output

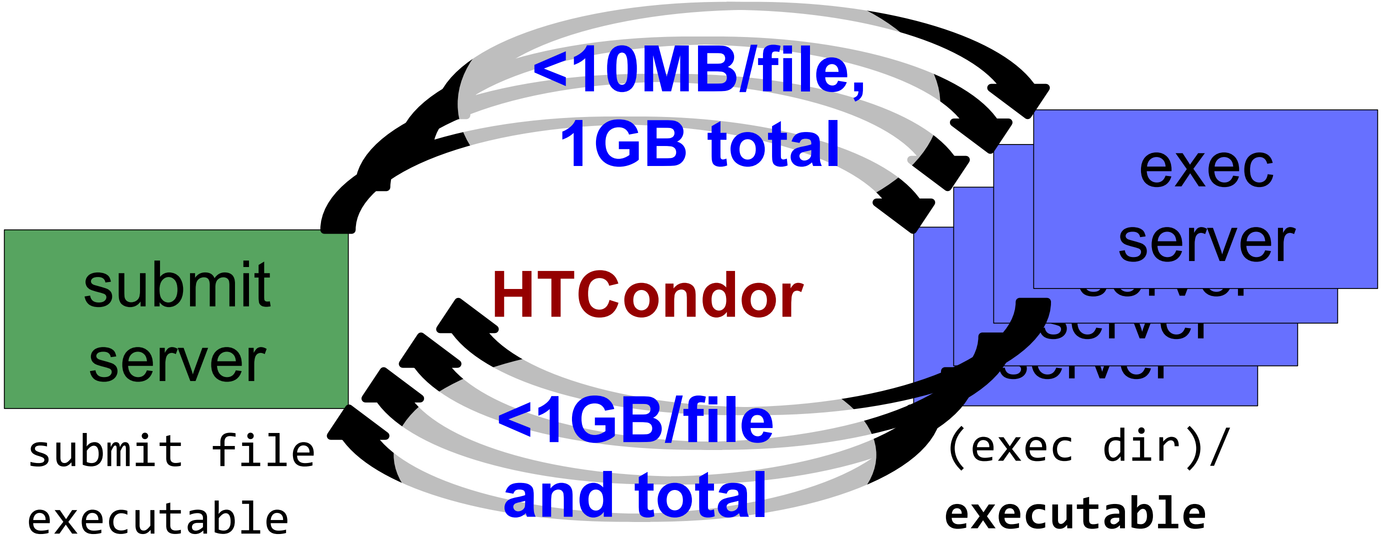
(exec dir)/
executable
input
output

Output transfers are staggered

Per-job transfer limits



Per-job transfer limits



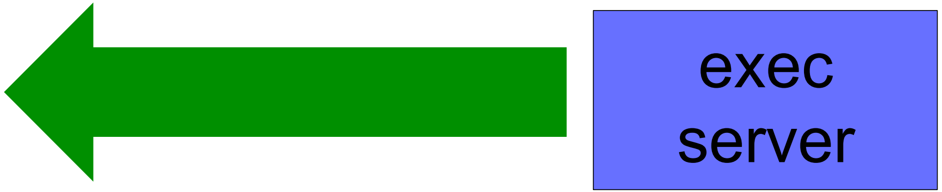
```
submit file
executable
dir/ input
output
```

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

Number of jobs matters!

Open Science Grid

Output for HTC and OSG


exec server

file size	method of delivery
~~words~~	~~within executable or arguments?~~
tiny – **1GB**	HTCondor file transfer (up to 1 GB total per-job)
1GB+	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**	HTCondor file transfer (up to 1 GB total per-job)
1GB+	shared file system (local copy, local execute servers)

- Why are there fewer options?

A OSG-wide File System?

WHY NOT?

- Technology barriers:
  - need local copies everywhere
    - too many users, too much data
  - need unrealistic network
  - security: need global user accounts
- Social barriers:
  - local sites unwilling/able to do the above

Overview – Data Handling

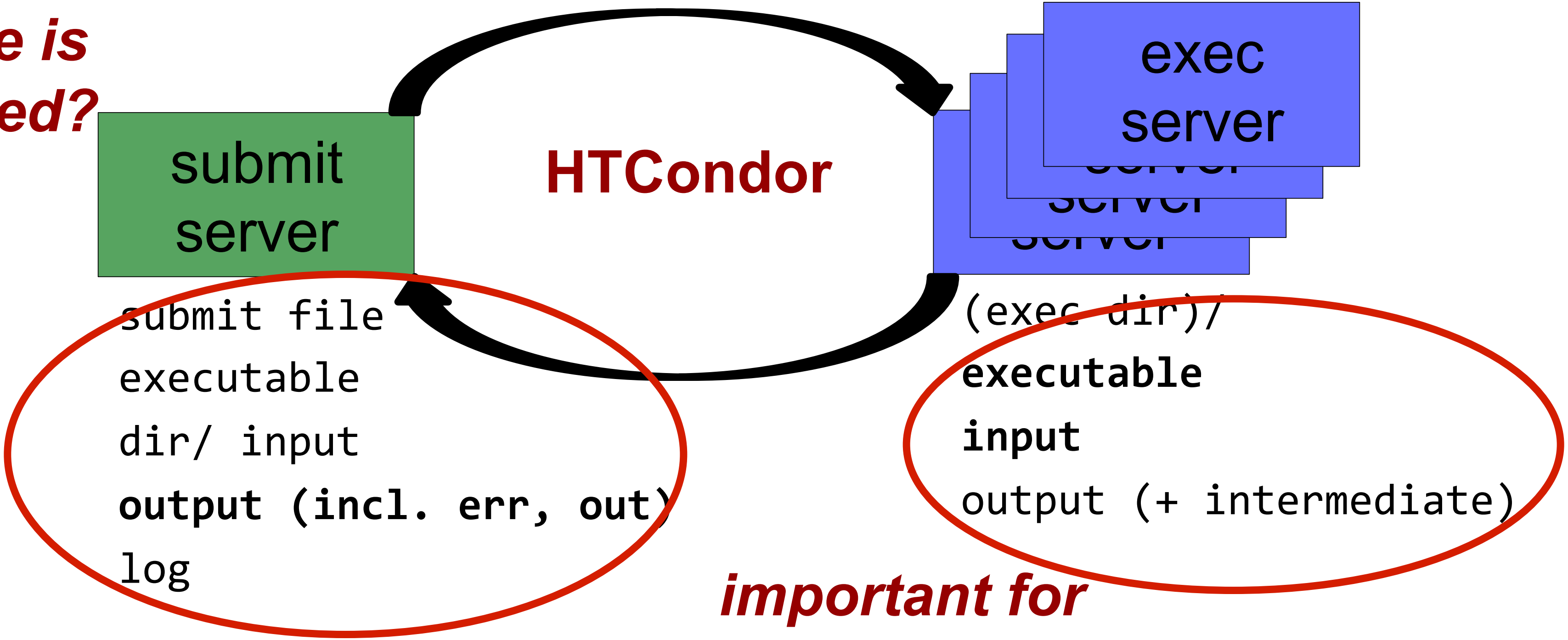
- Review of HTCondor Data Handling
- What is ‘Large’ Data?
- **Data Management Tips**
- Next talks: Dealing with Large Data

Data Management Tips

- **Know your job needs**
- Know your *batch* needs
- Reduce per-job data needs
- Test, then scale up gradually

Review: HTCondor Data Handling

What total disk space is needed?

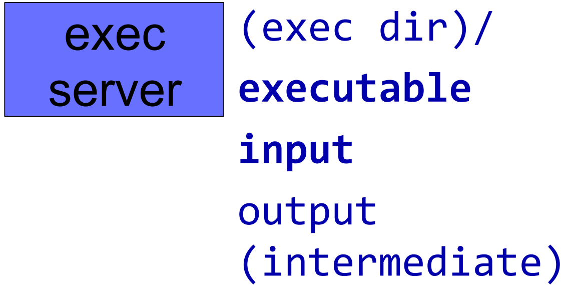


submit file
executable
dir/ input
output (incl. err, out)
log

(exec dir)/
executable
input
output (+ intermediate)

important for `request_disk`

In-Job Data Needs



exec server

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

- “Input” includes:
  - data *and* <u>software</u>
    - `executable`
    - `transfer_input_files`
  - files downloaded into the job (not transferred by HTCondor)
- “Output” includes:
  - your output, `output`, `error`
  - intermediate data (sub-directories and deleted files)
- **Total informs `request_disk`, *<u>but request a little extra for first tests!</u>***

Open Science Grid

More on memory and disk requests

- Memory *might* be informed by input file sizes, but this really depends on your software

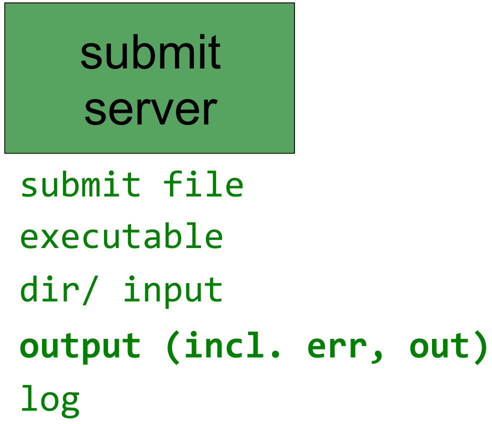
- A good place to start for tests?
  - If your files are smaller than 1 GB:

```
request_memory = 1GB
request_disk = 1GB
```

- Be prepared to increase or decrease for a batch!

Per-job Submit Node Data

- Submit file, log file, dag file
  - only exist on the submit node
  - *usually* fairly small (KBs at most)
- All files transferred *by HTCondor*
  - `executable, transfer_input_files`
- Any files copied back by HTCondor
  - `stdout, stderr`
  - default: any *files* remaining at end of job (*not* subdirectories)



```
submit file
executable
dir/ input
```
`output (incl. err, out)`
```
log
```

Data Management Tips

- Know your job needs
- **Know your *batch* needs**
- Reduce per-job data needs
- Test, then scale up gradually

In-job Batch Needs

exec server

- Per-job values remain the same
  - Same `request_memory` and `request_disk` as these are per-job values
  - Go by your tests!
  - Some software will vary considerably (e.g. our blastx example, today)

Submit Node Batch Needs

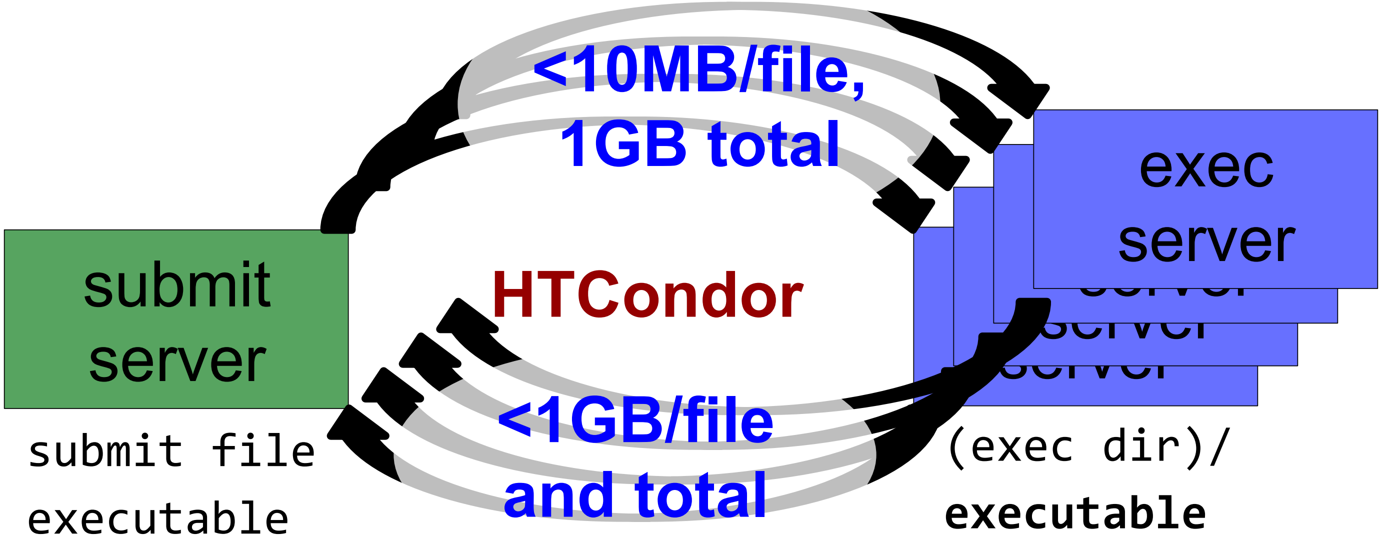
submit server

- Mostly multiplicative!
  - total can get BIG, *quickly*
- What will *NOT* multiply?
  - "shared" data: software and common input used by every job
  - versus "unique" input for each job, which multiplies

Data Management Tips

- Know your job needs
- Know your *batch* needs
- **Reduce per-job data needs**
- Test, then scale up gradually

Per-job transfer limits



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

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

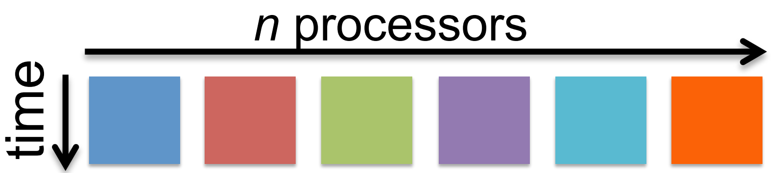
Reducing data needs

- ***An HTC best practice!***

- split large input for better throughput *and* less per-job data
- eliminate unnecessary data
- file compression and consolidation

Split Up Your Input!

- Think throughput!
  - Can the data be processed as smaller pieces?
  - (Exercises 2.3 and 3.1)



- Rather than “shared” input …
  - If each job only needs *some* of the data from a big file
  - (e.g. parameter sweeps)

Eliminate Unnecessary Data!

- Test for files on your local computer that aren't actually necessary.
  - maybe: strip down complex software.
- Remove excess data from input.
- Transfer input carefully. Remember:

```
transfer_input_files = file1,dir,dir_contents/
```

Open Science Grid

Compress and Combine!

1. Input: prior to job submission, decompress in-job
2. Output: as part of the job
3. Prior to transfer between your computer and the submit server.

```
tar czf newfile.tar.gz <contents>
tar xzf newfile.tar.gz
```

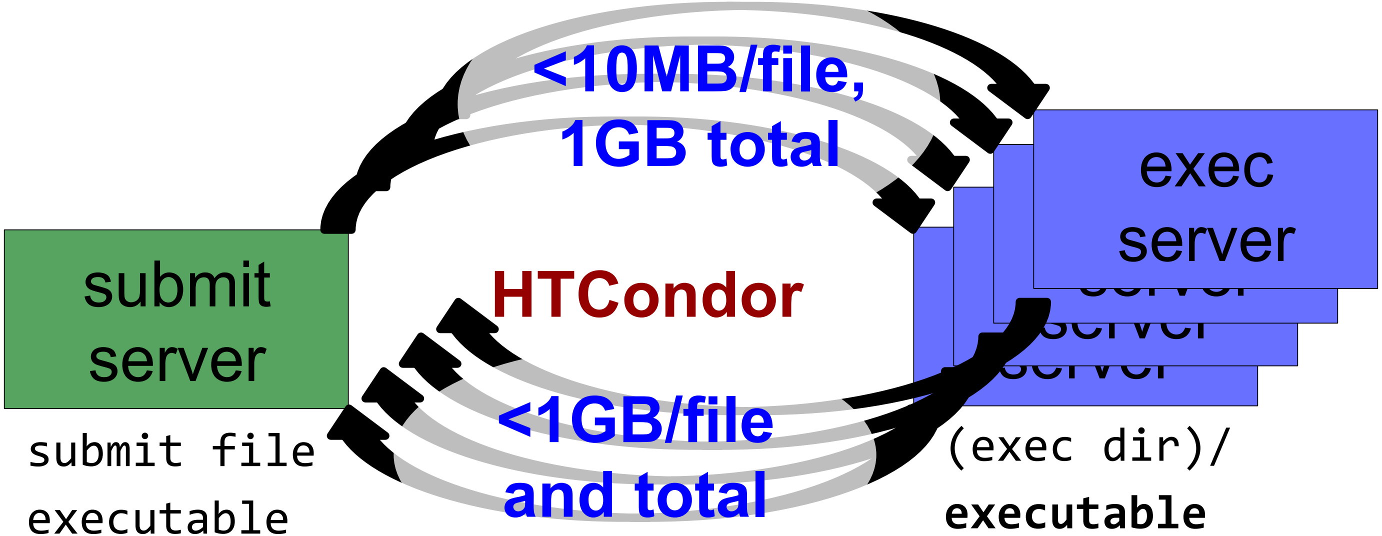
- Also use to combine many files, especially for “shared” data and software

```
transfer_input_files = file1,many_files.tar.gz,dir.tar.gz
```

Data Management Tips

- Know your job needs
- Know your *batch* needs
- Reduce per-job data needs
- **Test, then scale up gradually**

Per-job transfer limits



submit server

<10MB/file, 1GB total

HTCondor

<1GB/file and total

exec server

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

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

Number of jobs matters!

Test, and Scale Up *Gradually*

- Capabilities of submit servers and local networks may vary!

- Just because it isn't a problem for **10 jobs** doesn't mean you won't kill the submit server or network performance when you submit **100 or 1000 jobs.**

Overview – Data Handling

- Review of HTCondor Data Handling
- What is ‘Large’ Data?
- Data Management Tips
- **Next talks: Dealing with Large Data**

Exercises

- 2.1 Understanding a job's data needs
  - revisit the "blast" example from yesterday
- 2.2 Using data compression to reduce transfer load
  - compress/combine the database files
- 2.3 Splitting input (prep for large run in 3.1)
  - split large blast input

Questions?

- Feel free to contact me:
  - lmichael@wisc.edu

- Next: Exercises 2.1-2.3
- Later: Handling large input data