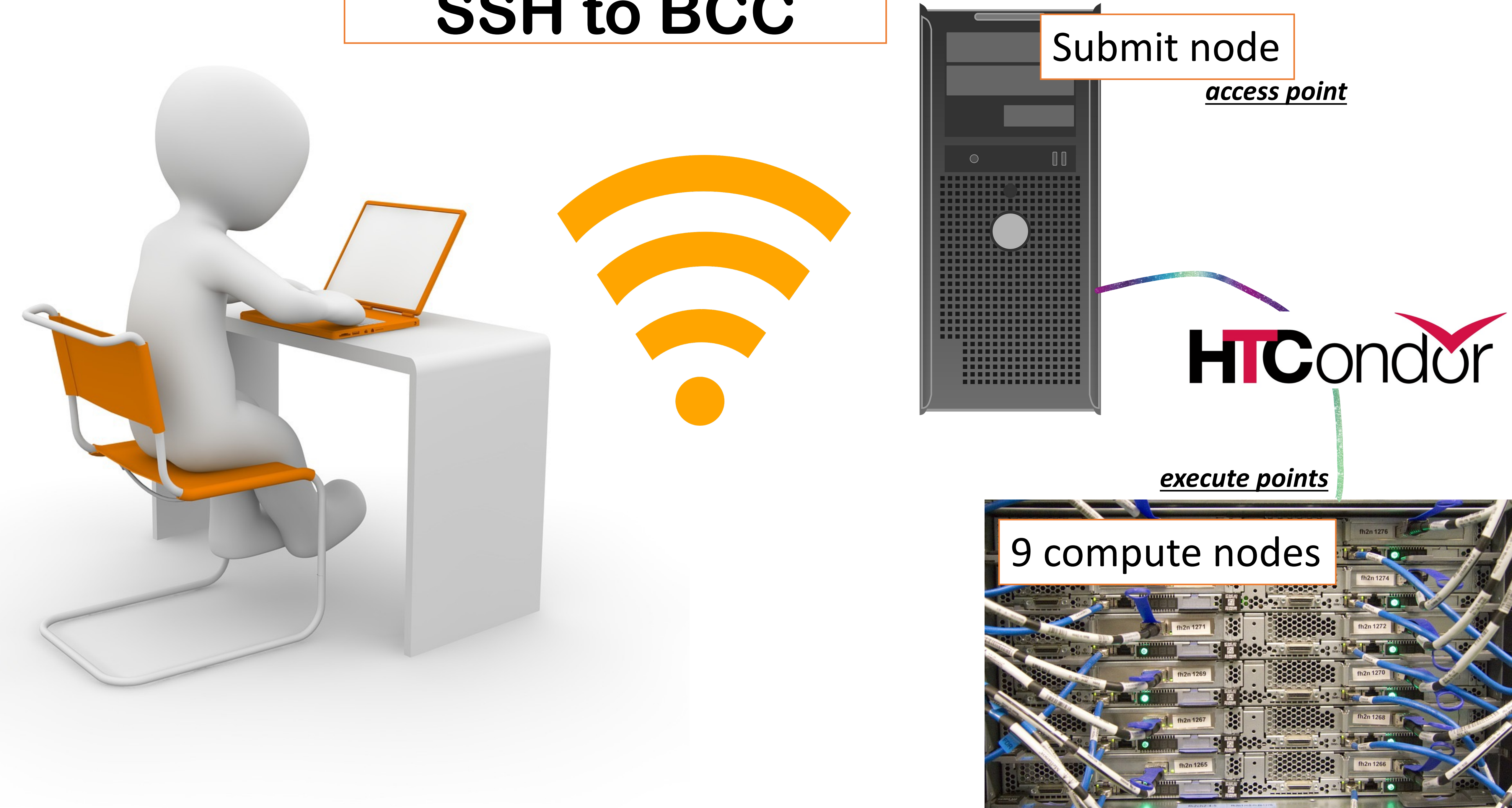
HTCondor on BCC

Dec 14, 2023

SSH to BCC



Submit node

access point

HTCondor

execute points

9 compute nodes

Credits

Lauren Michael – Open Science Grid, 2021

https://osg-htc.org/virtual-school-2021/materials/htcondor/files/osgvsp21-htc-htcondor.pptx

Christina Koch Researching Computer Facilitator

https://research.cs.wisc.edu/htcondor/HTCondorWeek2016/presentations/Koch_UserTutorial.pptx

OS Grid Materials

`https://osg-htc.org/virtual-school-2021/materials/`

Terminology: *Job*

***Job*: An independently-scheduled unit of computing work**

Three main pieces:

- **Executable:** the script or program to run
- **Input:** any options (arguments) and/or file-based information
- **Output:** files printed by the executable

In order to run *many* jobs, executable must run on the command-line **without any graphical input** from the user

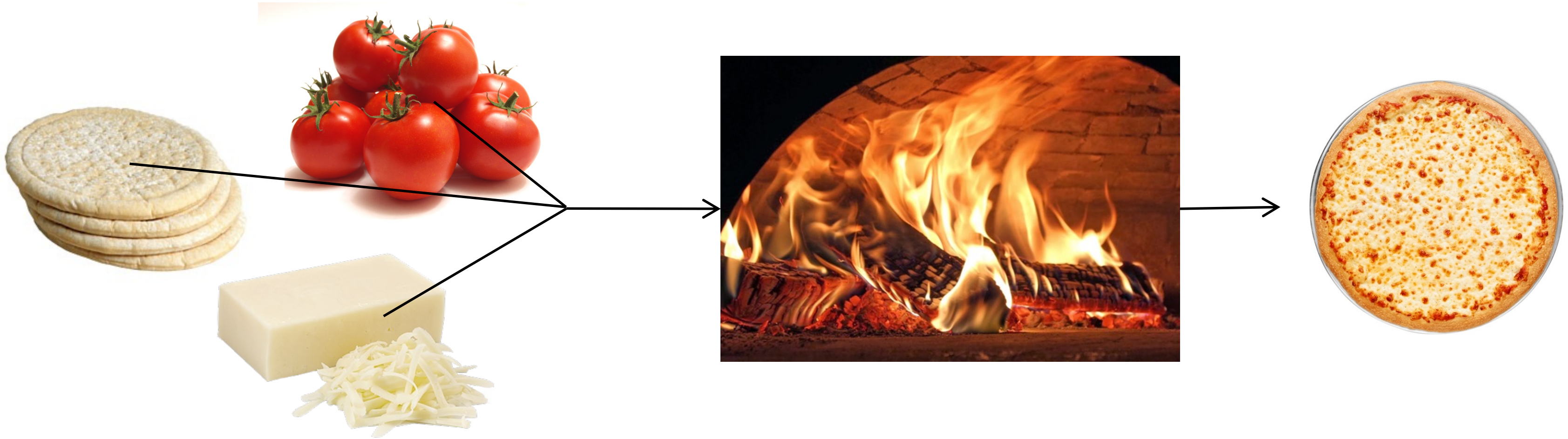
Basic Job Submission

Jobs

A single computing task is called a “job”

Three main pieces of a job are:

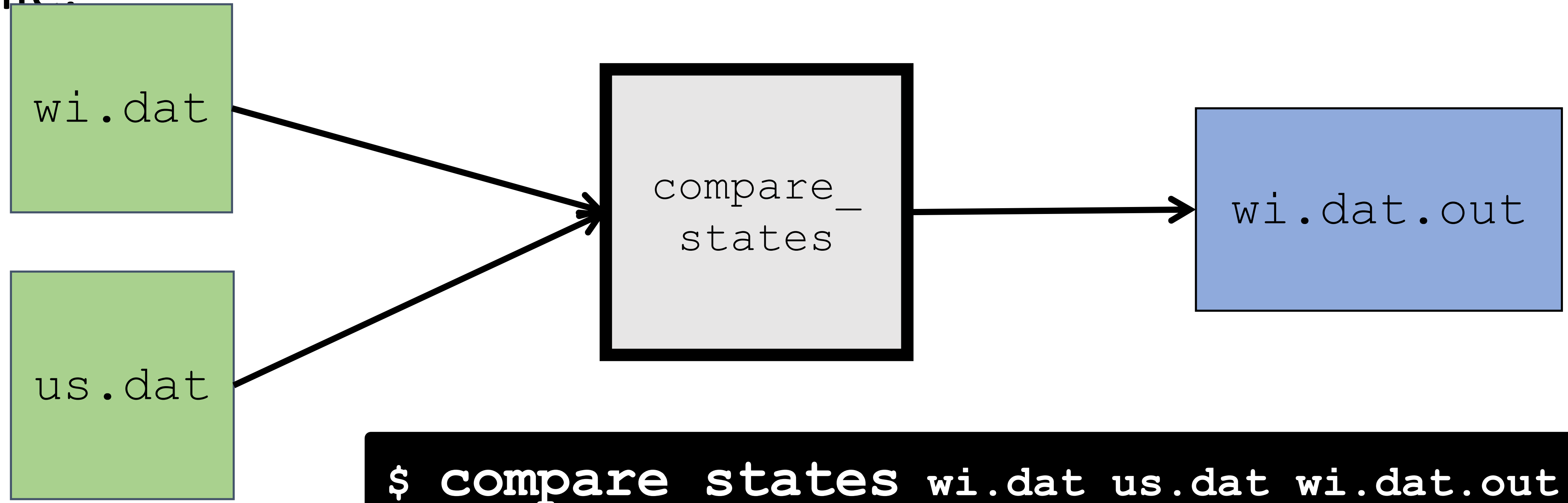
- input
- executable (program
- output



Executable must be runnable from the command line without any interactive input

Job Example

- program called “compare_states” (executable), which compares two data files (input) and produces a single output file.



```
$ compare_states wi.dat us.dat wi.dat.out
```

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat wi.dat.out

transfer_input_files = us.dat, wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat
wi.dat.out

transfer_input_files = us.dat,
wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

List your **executable** and any **arguments** it takes

Arguments are any options passed to the executable from the command line

```
$ compare_states wi.dat us.dat
wi.dat.out
```

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat
wi.dat.out

transfer_input_files =
us.dat, wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

comma-separated list of **input files to transfer** to the slot

wi.dat

us.dat

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat
wi.dat.out

transfer_input_files =
us.dat, wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

HTCondor will transfer back all new and changed files (output) from the job, automatically.

`wi.dat.out`

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat
wi.dat.out

transfer_input_files =
us.dat, wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

`log`: file created by HTCondor to track job progress

- *Explored in exercises!*

`output`/**`error`**: captures stdout and stderr from your program (what would otherwise be printed to the terminal)

Basic Submit File

```
executable = compare_states
arguments = wi.dat us.dat
wi.dat.out

transfer_input_files =
us.dat, wi.dat

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 20MB
request_memory = 20MB

queue 1
```

`request` the resources your job needs.

(More on this later)

`queue`: *final* keyword indicating “create 1 job” according to the above

Submitting and Monitoring

To submit a job/jobs: `condor_submit` *`submit_file`*

To monitor submitted jobs: `condor_q`

```
$ condor_submit job.submit
Submitting job(s).
1 job(s) submitted to cluster 128.
```

```
$ condor_q
-- Schedd: learn.chtc.wisc.edu : <128.104.101.92> @ 05/01/17 10:35:54
OWNER  BATCH_NAME            SUBMITTED   DONE   RUN    IDLE  TOTAL
JOB_IDS
alice  CMD: compare_states  5/9  11:05    _      _      1      1
128.0

1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
```

Job Idle

```
$ condor_q -nobatch
-- Schedd: submit-5.chtc.wisc.edu : <128.104.101.92>
 ID          OWNER       SUBMITTED     RUN_TIME ST PRI SIZE CMD
128.0        alice       5/9  11:09   0+00:00:00 I  0    0.0 compare_states wi.dat us.dat

1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
```

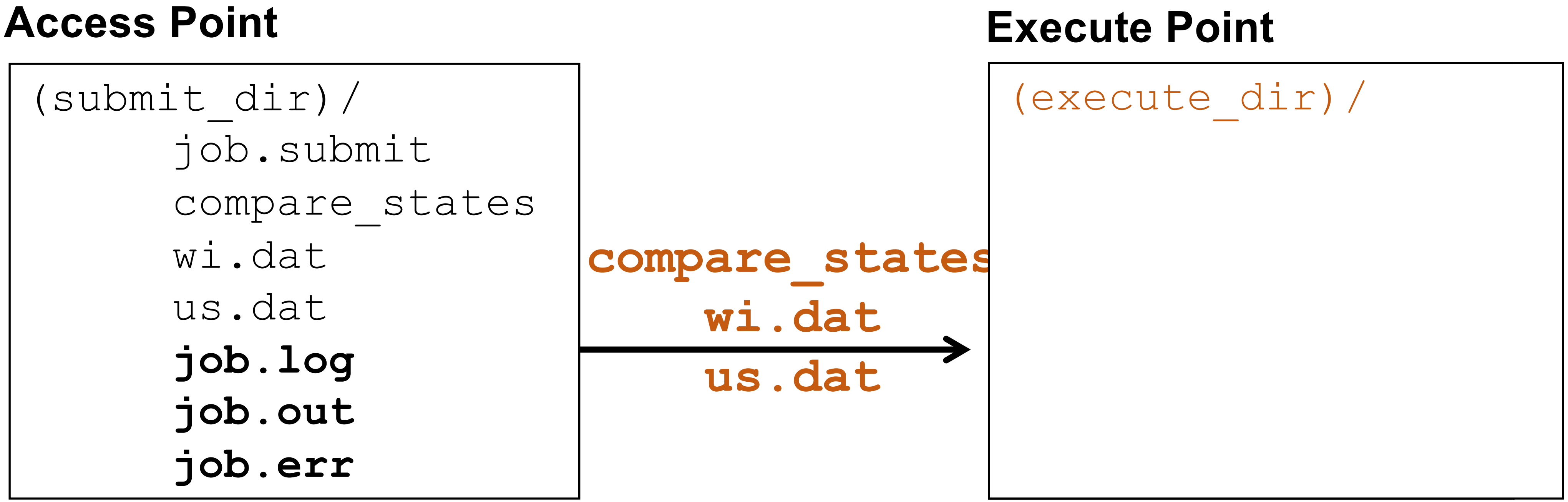
Access Point

```
(submit_dir)/
        job.submit
        compare_states
        wi.dat
        us.dat
        job.log
        job.out
        job.err
```

Job Starts by doing File Transfer

```
$ condor_q -nobatch
-- Schedd: submit-5.chtc.wisc.edu : <128.104.101.92:9618>
 ID          OWNER       SUBMITTED      RUN_TIME ST PRI SIZE CMD
128.0        alice       5/9  11:09    0+00:00:00 <  0    0.0 compare_states wi.dat us.dat

1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```



Access Point

(submit_dir)/
job.submit
compare_states
wi.dat
us.dat
job.log
job.out
job.err

compare_states
wi.dat
us.dat

Execute Point

(execute_dir)/

Job Running

```
$ condor_q -nobatch
-- Schedd: submit-5.chtc.wisc.edu : <128.104.101.92>
 ID          OWNER        SUBMITTED      RUN_TIME  ST PRI SIZE CMD
128.0        alice        5/9  11:09    0+00:01:08  R  0   0.0 compare_states wi.dat us.dat

1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```

Access Point

(submit_dir)/
job.submit
compare_states
wi.dat
us.dat
job.log
job.out
job.err

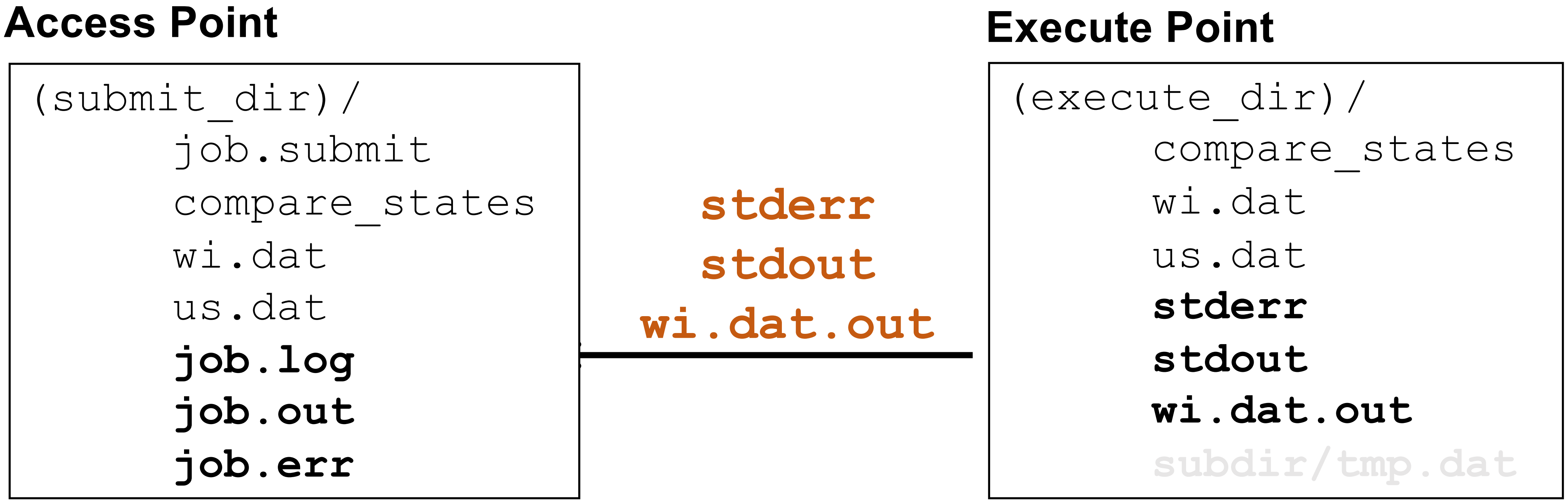
Execute Point

(execute_dir)/
compare_states
wi.dat
us.dat
stderr
stdout
wi.dat.out
subdir/tmp.dat

Job Completes

```
$ condor_q -nobatch
-- Schedd: submit-5.chtc.wisc.edu : <128.104.101.92>
 ID          OWNER      SUBMITTED     RUN_TIME ST PRI SIZE CMD
128          alice      5/9  11:09   0+00:02:02 >  0   0.0 compare_states wi.dat us.dat

1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```



Access Point

```
(submit_dir)/
       job.submit
       compare_states
       wi.dat
       us.dat
       job.log
       job.out
       job.err
```

stderr
stdout
wi.dat.out

Execute Point

```
(execute_dir)/
       compare_states
       wi.dat
       us.dat
       stderr
       stdout
       wi.dat.out
       subdir/tmp.dat
```

Job Completes (cont.)

```
$ condor_q -nobatch

-- Schedd: submit-5.chtc.wisc.edu : <128.104.101.92:9618?...
 ID      OWNER                 SUBMITTED      RUN_TIME ST PRI SIZE CMD

0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
```

Access Point

`(submit_dir)/`
- `job.submit`
- `compare_states`
- `wi.dat`
- `us.dat`
- **`job.log`**
- **`job.out`**
- **`job.err`**
- **`wi.dat.out`**

RESEARCH
DRIVE

Connecting research drive

https://chtc.cs.wisc.edu/uw-research-computing/transfer-data-researchdrive

Research drive

The information on the CHTC web page has all the details needed to connect to your research drive from your Biochemistry Cluster login using the `smbclient` communication software.

Connect to ResearchDrive:

```
$ smbclient -k //research.drive.wisc.edu/PI-Name
```

`Prompt changes to:`

```
smb: \>
```

Move files: To move files, you will use the get and put commands:

To move files from BCC to ResearchDrive:

```
smb: \> put filename
```

To move files from ResearchDrive to BCC:

```
smb: \> get filename
```

Finish: End connection to ResearchDrive:

```
smb: \> exit
```

See the CHTC web page https://chtc.cs.wisc.edu/uw-research-computing/transfer-data-researchdrive

on using `mput` and `mget` or the wild card `*` for batch transfer of multiple files.