



CSC

ICT Solutions for
Brilliant Minds



Troubleshooting batch jobs in CSC HPC environment

2025-03-19, Ari-Matti Sarén





A little goes a long way

Sometimes troubleshooting can get really involved, require specialized software tools and expertise...

A little goes a long way

Sometimes troubleshooting can get really involved, require specialized software tools and expertise...

... but in great majority of failed jobs the problem is something simple that can be identified and corrected with beginner skills

A little goes a long way

Sometimes troubleshooting can get really involved, require specialized software tools and expertise...

... but in great majority of failed jobs the problem is something simple that can be identified and corrected with beginner skills

You just have to check the most common problems

So your batch job failed.

What to do?

Step 0

1. Don't panic
2. Take a deep breath
3. Get a cup of coffee
4. (Optional) Shake your fist at the heavens and curse you fate. Swearing like a sailor may help (but is not recommended in polite company)

Step 1

- Check the error file captured by the batch job script
 - By default *stderr* and *stdout* (i.e. the text printed to screen when running interactively) is captured in file "slurm-<jobid>.out"
 - Can be directed to separate files by specifying `--error` and `--output` in the batch job script
- Check any other error files and logs the program itself may have produced
 - In case of workflows try to find the error message/log produced by the application that actually failed

Step 1 (continued)

- Error messages can sometimes be long, cryptic and a bit intimidating, but try skimming through them and see if you can spot something "human readable" instead of "nerd readable"
 - Often you can spot the actual problem if you go through the whole message. Something like "required input file so-and-so missing" or "parameter X out of range" etc.

Troubleshooting checklist 1

- Did the job fail at once or did it run for some time?
 - Jobs failing immediately are often due to something simple, like typos in command line, missing inputs, bad parameters *etc.*

- Solution: Check your batch job script
 - Use interactive sessions and "test" partition to check command syntax and batch job script

Troubleshooting checklist 1 (continued)

- Some common problems
 - "command not found": Make sure necessary modules are loaded or your application is added to \$PATH
 - Input file not found:
 - Check spelling
 - Remember Linux filenames are case sensitive
 - Make sure you are looking in the correct directory (default working directory for a batch job is the directory where sbatch was run).
 - Spaces in filenames must be escaped ("my\ file.txt")
 - Some characters can cause problems (,\$><|:;%# etc) because they are interpreted by bash. They should be avoided or escaped
 - Application specific problems in syntax: Check software documentation

Troubleshooting checklist 2

- Did the job run out of time?

- Check the Slurm output

```
slurmstepd: error: *** JOB 20364046 ON r18c43 CANCELLED AT  
2024-02-08T09:37:30 DUE TO TIME LIMIT ***
```

- You can also check seff/sacct output

```
State: TIMEOUT
```

- Solution: Increase time reservation

- Also check #4

Troubleshooting checklist 3

- Did the job run out of memory?

- Check the Slurm output

```
slurmstepd: error: StepId=20364048.batch exceeded memory limit (1598536704 > 1048576000), being killed
```

```
slurmstepd: error: Exceeded job memory limit
```

- You can also check seff output

```
Memory Efficiency: 152.45% of 1000.00 MB
```

- Solution: Increase memory reservation

- Also check #4

Troubleshooting checklist 3 (continued)

- Note: Some applications (especially Java applications) may also need adjusting software parameters
 - For Java check/set maximum heap size (-Xmx) or set `_JAVA_OPTIONS="-Xmx "`
`export _JAVA_OPTIONS=" -Xmx6g "`

Troubleshooting checklist 4

- Did the job actually get allocated the resources you specified?
 - Remember to run with sbatch
 - Problems in the batch job script can cause parameters to be ignored and default values getting used instead
 - Check the allocated resources with seff/sacct and compare to reservations
- Solution: Check your batch job script
 - Pay attention to possible hidden characters at end of lines
 - If script originally written on a Windows computer try command `dos2unix myjob.sh`

Troubleshooting checklist 5

- Did you run out of storage?
 - Error messages vary, but are typically about failing to write a file
 - Check the status of your disk areas with command `csc-workspaces`
 - Problems with `/tmp` filling up common
- Solution:
 - If problem with `$HOME`, `/projappl`, `/scratch`: Make room
 - If problem with `$LOCAL_SCRATCH`: Reserve more
 - If problem with `/tmp`: Make sure `$TMPDIR` is defined and points to location with more space
 - Also check if the application has some parameter for temp files
 - For Java applications try: `export _JAVA_OPTIONS=-Djava.io.tmpdir=/new/tmp/dir`

Failing all that

- If you can't figure out the problem, please don't hesitate to contact us at:

`servicedesk@csc.fi`



facebook.com/CSCfi



twitter.com/CSCfi



youtube.com/CSCfi



linkedin.com/company/csc---it-center-for-science



github.com/CSCfi