Investigating a Job Failure

Jobs do not always run correctly. There is a list of reasons why jobs or applications stop or fail. The most common causes are:

- Exceeding requested resource limits, such as memory or time limit, for example.
- Errors specific to the software.
- Errors in data processing.

> **Note**
>
> It is important to collect error/output messages either by writing such information to the default location or by specifying specific locations using the `--error/--output` options, for example, although this will depend on individual software.
>
> Error and output messages are the starting point for investigating an error in the job.

Exceeding Resource Limits

Each partition defines maximum and default limits for runtime and memory usage. Within the job specification the current limits can be defined within the ranges. For better scheduling, the job requirements should be estimated and the limits should be adapted to the needs. The lower the limits the better SLURM can find a spot. Furthermore, the less resource overhead is specified the less resources are wasted, e.g. for memory.

> **Info**
>
> If a job exceeds the execution time or memory limit, SLURM will kill the job.

Error information

In both cases, the error file provides appropriate information:

- Time limit:

```
(...)
slurmstepd: error: *** JOB 41239 ON fnode01 CANCELLED AT 2016-11-30T11:22:57 DUE
TO TIME LIMIT ***
(...)
```

- Memory limit:

```
(...)
slurmstepd: error: Job 41176 exceeded memory limit (3940736 > 2068480), being killed
slurmstepd: error: Exceeded job memory limit
slurmstepd: error: *** JOB 41176 ON fnode01 CANCELLED AT 2016-11-30T10:21:37 ***
(...)
```

Software Errors

The exit code of a job is captured by Slurm and saved as part of the job record. **For sbatch jobs the exit code of the batch script is captured. For srun, the exit code will be the return value of the executed command. Any non-zero exit code is considered a job failure, and results in job state of FAILED**. When a signal was responsible for a job/step termination, the signal number will also be captured, and displayed after the exit code (separated by a colon).

Depending on the execution order of the commands in the batch script, it is possible that a specific command fails but the batch script will return zero indicating success. Consider the following simplified example:

fail.R

```
var<-sq(1,1000000000)
```

job.sbatch

```
#!/bin/bash
# Slurm options
#SBATCH --mail-user=mustermann@unibe.ch
#SBATCH --mail-type=begin,end,fail
#SBATCH --job-name="Simple Example"
#SBATCH --time=00:05:00
#SBATCH --mem-per-cpu=2G
#SBATCH --constrains=sandy

# Put your code below this line
Rscript fail.R
echo "Script finished"
```

The exit code and state wrongly indicates that the job finished successfully:

```
sbatch job_slurm.sh
Submitted batch job 41585

sacct -j 41585
      JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
41585        Simple E +        all         id          1  COMPLETED      0:0
41585.batch       batch                  id          1  COMPLETED      0:0
```

Only the R-specific output file shows the error:

fail.Rout

```
(...)
> var<-sq(1,1000000000)
Error: could not find function "sq"
Execution halted
```

You can bypass this problem by exiting with a proper exit code as soon as the command failed:

job_sbatch.sh

```
#!/bin/bash

# Slurm options
#SBATCH --mail-user=nico.faerber@id.unibe.ch
#SBATCH --mail-type=begin,end,fail
#SBATCH --job-name="Simple Example"
#SBATCH --time=00:05:00
#SBATCH --mem-per-cpu=2G
#SBATCH --constrains=sandy

# Put your code below this line
Rscript fail.R || exit 91
echo "Script finished"
```

Now, the exit code and state matches the true outcome:

```
sbatch job_slurm.sh
Submitted batch job 41925

sacct -j 41925
      JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
41925        Simple E +        all         id          1     FAILED     91:0
41925.batch       batch                  id          1     FAILED     91:0
```

Always check application-specifc output files for error messages.