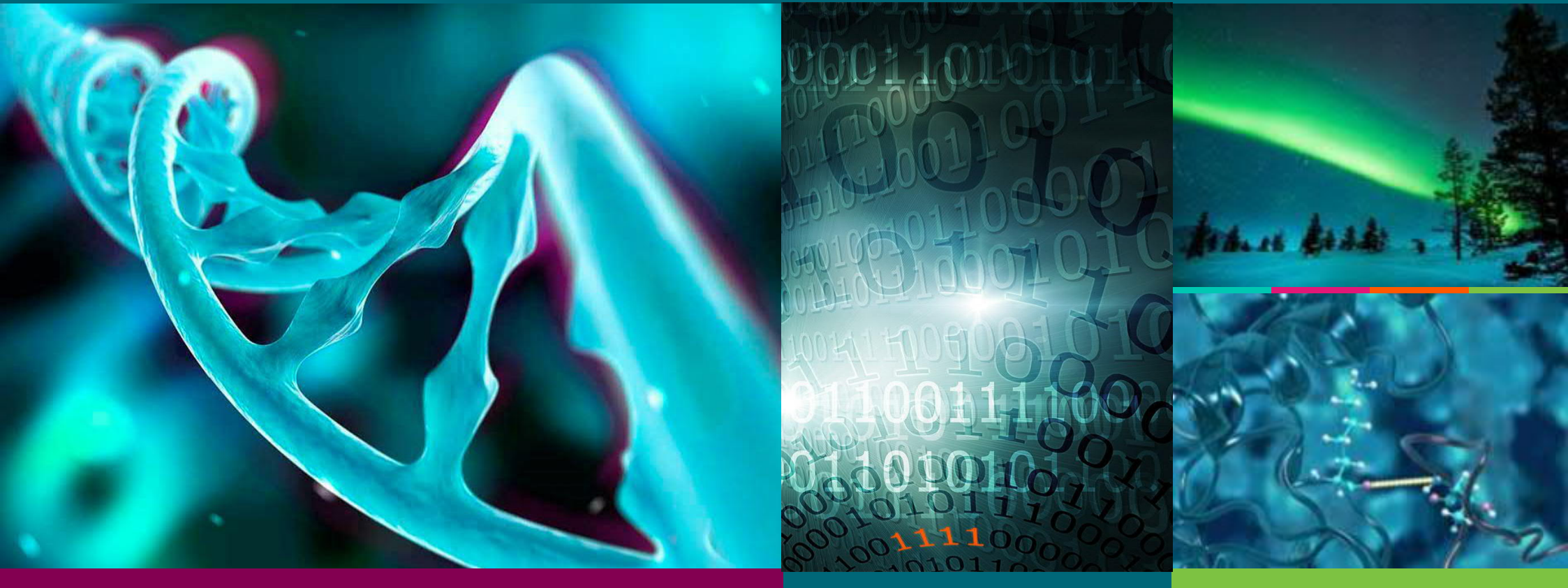


# Monitoring resource usage



## seff

- Command `seff` will print a summary of requested and used resources for both running and finished batch jobs

```
seff <jobid>
```

## Example 1: Core utilization bad

```
>seff 123456
```

```
Job ID: 123456
```

```
Cluster: puhti
```

```
User/Group: user/group
```

```
State: COMPLETED (exit code 0)
```

```
Nodes: 1
```

```
Cores per node: 8
```

```
CPU Utilized: 0:51:01
```

```
CPU Efficiency: 12.48% of 06:56:08 core-walltime
```

```
Memory Utilized: 5.98 GB
```

```
Memory Efficiency: 75.89% of 7.88 GB
```

```
Job consumed X.XX CSC billing units based on cpu reservation  
multiplier
```

## Example 2: Memory utilization bad

```
>seff 123456
```

```
Job ID: 123456
```

```
Cluster: puhti
```

```
User/Group: user/group
```

```
State: COMPLETED (exit code 0)
```

```
Nodes: 1
```

```
Cores per node: 8
```

```
CPU Utilized: 05:49:01
```

```
CPU Efficiency: 83.89% of 06:56:08 core-walltime
```

```
Memory Utilized: 5.98 GB
```

```
Memory Efficiency: 6.25% of 92.59 GB
```

```
Job consumed X.XX CSC billing units based on cpu reservation  
multiplier
```

## Example 3: Job failed due to time reservation running out

```
>seff 123456
```

```
Job ID: 1234566
```

```
Cluster: puhti
```

```
User/Group: user/csc
```

```
State: TIMEOUT (exit code 1)
```

```
Nodes: 1
```

```
Cores per node: 12
```

```
CPU Utilized: 02:06:41
```

```
CPU Efficiency: 70.30% of 03:00:12 core-walltime
```

```
Memory Utilized: 24.70 GB
```

```
Memory Efficiency: 72.27% of 34.18 GB
```

```
Job consumed 6.01 CSC billing units based on cpu reservation multiplie
```

## Example 4: Job failed probably due to memory reservation running out

```
>seff 123456
```

```
Job ID: 1234566
```

```
Cluster: puhti
```

```
User/Group: user/csc
```

```
State: FAILED (exit code 1)
```

```
Nodes: 1
```

```
Cores per node: 12
```

```
CPU Utilized: 02:06:41
```

```
CPU Efficiency: 70.30% of 03:00:12 core-walltime
```

```
Memory Utilized: 35.70 GB
```

```
Memory Efficiency: 101.3% of 34.18 GB
```

```
Job consumed 6.01 CSC billing units based on cpu reservation multiplie
```

# sacct

- Command `sacct` can be used to study past jobs
- Usefull when deciding proper resource requests

<code>sacct</code>	Short format listing of jobs starting from midnight today
<code>sacct -j &lt;jobid&gt;</code>	information on single job
<code>sacct -S YYYY-MM-DD</code>	listing start date
<code>sacct -l</code>	long format output
<code>sacct -o</code>	list only named data fields, e.g.

```
sacct -o jobid,jobname,reqmem,maxrss,averss,state,elapsed -j <jobid>
```

# sacct



Some useful data fields:

jobid	Job id number
jobname	Job name
reqcpus	Cores requested from SLURM
reqmem	Memory requested from SLURM
maxrss	Maximum used memory
averss	Average used memory per process/core
state	Exit status of job
elapsed	Execution time

```
sacct -o jobid,jobname,ntasks,reqnodes,allocnodes,reqcpus,alloccpus,reqmem,maxrss,averss,timelimit,elapsed,state -j 17317981
```

JobID	JobName	NTasks	ReqNodes	AllocNodes	ReqCPUS	AllocCPUS	ReqMem	MaxRSS	AveRSS	Timelimit	Elapsed	State
17317981	cretest		1	1	8	8	504Mc			00:01:00	00:00:01	COMPLETED
17317981.ba+	batch	1	1	1	8	8	504Mc	1580K	1580K		00:00:01	COMPLETED