Max-Planck-Institut für Polymerforschung
Max Planck Institute for Polymer Research

Preparing and running jobs on the THINC Cluster

Robert Klein*

February 24, 2025

Considerations when preparing jobs

Where to read, store and archive data:

Do not write many files to any bee during a job. Instead pack the files at the end of the job and copy the packed file.

- copy/read from */data/bee8/$LOGNAME* or */data/bee14/$LOGNAME* before your computations start.
- Create */usr/scratch/$LOGNAME* and use it during computation.
- At the end of the job, pack the data (e.g. using *tar* and copy the archive to */data/bee8/$LOGNAME* or */data/bee14/$LOGNAME*. Not enough space? Mail *helpdesk@mpip-mainz.mpg.de* to get more. Note, there is a file count quota also which will not be increased.

Choosing the Queue / Available Queues

- Also use *sinfo* command

Get information about available queues and their respective time limit using the *sinfo* command.

The Job script

A job script has four parts:

1. Shebang and Slurm options (Those have to come before anything else)

*Robert.Klein@mpip-mainz.mpg.de

Table 1: Queues on the THINC cluster (as of February 2025)

Partition name	Cores / Threads	Memory	Maximum CPU time	Remarks
CPU_Std20	20	192 GB	36 h	default partition
CPU_Std32	32	128 GB	36 h	
CPU_IBm32	32	32 GB	36 h	up to 36 nodes per job
GPU_Std16	16	192 GB	36 h	NVIDIA V100S GPU (32GB)

2. Set up environment
    - ensure there's a scratch directory to use as work directory
    - ensure programs needed are available
    - copy data you need to work directory
3. run job
4. cleaning up
    - pack result data
    - copy result archive away
    - remove work directory

Shebang and Slurm options

A Slurm job always begins with a shebang line:

```
#!/bin/bash -l
```

Listing 1: First line of a Slurm Script

Then, before anything else come the Slurm options. Here is an example:

```
## Slurm Options

# which partition to use (here: the 20 core nodes)
#SBATCH --partition=CPU_Std20

# Number of nodes:
#SBATCH --nodes=1

# Number of cores (i.e. 'rank' in MPI):
#SBATCH --ntasks=20

# mails? and to whom
#SBATCH --mail-type=END --mail-user=MY_EMAIL_ADDRESS

# asks Slurm to send the USR1 signal 300 seconds (5 minutes) before
```

```
# end of the time limit, so you can run a cleanup job
#SBATCH --signal=B:USR1@300
```

Listing 2: Example of setting Slurm options

Set up the environment

Here you typically set environment variables, put some functions, create directories you need later.

As an example, here is a cleanup function to be run when the USR1 signal requested in the Slurm options above arrives.

```
SRCDIR="/data/bee8/{$LOGNAME}/input12"

SCRATCHBASE="/data/scratch/{$LOGNAME}"
SCRATCHDIR="${SCRATCHBASE}/job12"

RESULTDIR="/data/bee8/{$LOGNAME}/output12"
```

Listing 3: Example for environment variable use

```
# usage: 'ensure_dir newdir'
# ensures a directory exists or aborts script.
ensure_dir()
{
    if [ ! -d "$1" ]; then
        mkdir -p "$1"
        if [ $? != 0 ]; then
            printf "Couldn't␣create␣%s.\n" "$1" >&2
            printf "Bailing␣out...\n" >&2
            exit 1
        fi
    fi
}

# ensuring I have scratch and result directories
ensure_dir "${SCRATCHDIR}"
ensure_dir "${RESULTDIR}"
```

Listing 4: creating scratch and result directories

```
cleanup()
{
    # pack results
    cd "${SCRATCHBASE}"
```

```
    tar cfz job12.tar.gz job12
    scp job12.tar.gz "${RESULTDIR}"
    cd
    rm -rf "${SCRATCHBASE}"
}

# The function 'cleanup' should exist before calling trap.
trap 'cleanup' USR1
```

Listing 5: cleanup function

```
cd "${SCRATCHDIR}"
tar fxz "${SRCDIR}/job12_data.tar.gz"
```

Listing 6: fetching and unpacking the source data

Run the job

```
./job12_data/my_binary
```

Listing 7: running the actual job

cleaning up

As I already have defined a cleanup function above, I only need to disable the trap again and call this very cleanup function.

```
# disable the trap, as we're going to call cleanup anyway and don't
# want it to be invoked again (if Slurm sends SIGUSR1 just in that
# moment) while already running it.
trap - USR1
cleanup
```

Listing 8: cleanup