

# Intro to tensor compute cluster @ MEB

Robert Karlsson Rikard Öberg

2024-10-01

# Workshop program

- **9:00** Presentation
  - What is tensor, and what can tensor do for me?
  - Using tensor
  - Other systems
  - Queue etiquette
- **9:45** Break
- **10:00** Practical: simple Slurm jobs
- **10:?? – 10:45** Q&A

# What is tensor?

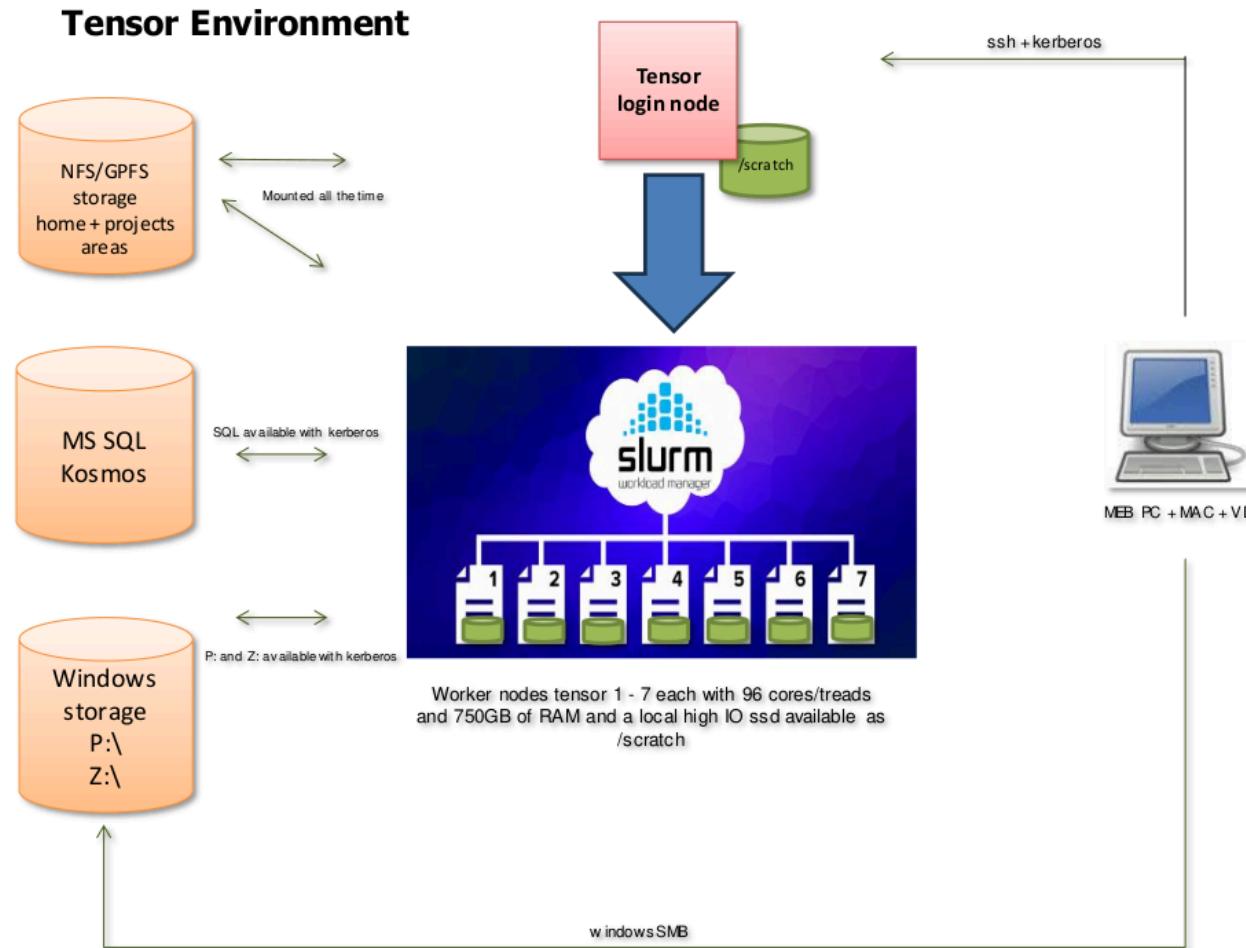
# What is tensor?

- MEB's new(ish)<sup>1</sup> local High Performance Computing system
- For medium-size computing, data management, statistical analysis, bioinformatics, ...
- Replaces previous servers vector, matrix<sup>2</sup>
- Runs Red Hat Enterprise Linux 9.4 and Slurm 23.11.1

1. online since April 2024

2. but keep using matrix for SAS, for now

# What is tensor?



# What can tensor do for me?

- run analyses that take *longer time* than I'd like to keep my own computer running
  - "I usually start the analysis, then leave the laptop open over the weekend..."
- run analyses that use *more resources* than my computer has (CPU cores and/or memory)
  - "My analysis data consists of approximately 500 million rows..."

# What can tensor do for me (continued)?

- run many independent analyses in parallel
  - Simulation over a range of parameters, data analysis in batches, ...
- run analyses that require software that is only available for, or runs better on, Linux
- all of the above with access to the MEB network file storage and databases
  - `P:`, `Z:`, `kosmos`, ...

# Using tensor

# The user guide

- Available at <https://meb-ki.github.io/meb-tensor-docs/>
- Holds answers to many common questions (continuously updated)

User documentation for the tensor compute cluster at MEB | meb-tensor-docs - Google Chrome

User documentation for the tensor compute cluster at MEB

meb-ki.github.io/meb-tensor-docs/

## Connecting

The login node `tensor.meb.ki.se` or just `tensor` is only accessible from within MEB's network (or through VDI).

### Windows

Use KiTTY, (available in MEB Software Center), or your own preferred SSH program to connect to the host `tensor`.

*[!TIP] Create a saved KiTTY session with the hostname `tensor`, and tick the box in Connection -> Data to "Use system username". With this setting you should be able to connect without typing anything (your Windows login will be re-used).*

### Mac/Linux

Use `ssh` in the terminal, or some other interface that you prefer, to connect to `tensor` with your MEB username and password.

*[!TIP] Use SSH keys for passwordless login (see for example [instructions at Uppmax](#))*

# Slurm workload manager

- To use tensor, you must use Slurm<sup>1</sup>
- Why Slurm?
  - allocates resources efficiently
  - keeps jobs separated while running



1. “Simple Linux Utility for Resource Management”

# Slurm in practice

# Slurm (batch jobs)

- `sbatch` – submits batch jobs
  - You already have a script: send it to the job queue

```
# instead of  
./run_analyses.sh  
# or  
Rscript simulate_something.R  
  
# send jobs to the Slurm queue  
sbatch -c 2 -t 0-8 --memory=10G run_analyses.sh  
sbatch -c 2 -t 0-8 --memory=10G --wrap='Rscript simulate_something.R'
```

- Tell Slurm:
  - resources needed (cpu cores, memory)
  - for how long (roughly)?
- Slurm will run your job as soon as possible

# Slurm (batch jobs)

- Manage batch jobs: `squeue`, `scancel`

```
[robkar@tensor ~]$ squeue
   JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
 98108      core BT_K60_H  hamkha  R  6:16:46      1 tensor4
 95475      core interact robkar  R 1-19:57:08      1 tensor7
 98123      core DCSM_bmi  pegler  R  3:36:49      1 tensor3
```

- See also `squeue -l`, `squeue -O ...`, `squeue --me`
- When you change your mind:

```
scancel 95475 # cancel a specific job
scancel --me # cancel all my jobs
```

# Slurm (interactive)

- When developing your analysis code, ask for an interactive session with `salloc`
- Like sbatch, specify time, cpu cores (memory)
- Exit when done to free up resources

```
# interactive job using 4 cores for 8 hours
salloc -t 0-8 -c 4
```

# Slurm (interactive)

- X11 graphics available (see user guide for details)

```
[robkar@tensor01 ~]$ ml add Stata
[robkar@tensor01 ~]$ xstata
[robkar@tensor01 ~]$ [REDACTED]
Stata/BE 18.0@tensor01.meb.ki.se
```

The screenshot shows the Stata/BE 18.0 graphical user interface. At the top, there's a terminal window with the command 'xstata'. Below it is the Stata application window. The title bar says 'Stata/BE 18.0@tensor01.meb.ki.se'. The menu bar includes 'File', 'Edit', 'Data', 'Graphics' (which is underlined in red), 'Statistics', 'User', 'Window', and 'Help'. The 'Graphics' menu has several options like 'Plot', 'Graph', etc. The main window has two tabs: 'History' and 'Results'. The 'Results' tab is active and displays the Stata license information. It shows the Stata logo, version 18.0 BE-Basic Edition, copyright information from 1985-2023, and details about the 80-user network perpetual license. It also mentions Unicode support and the command 'Running /nfs/sw/eb/software/Stata/18/sysprofile.do ...'.

```
[robkar@tensor01 ~]$ ml add R/4.3.2
[robkar@tensor01 ~]$ R
```

R version 4.3.2 (2023-10-31) -- "Eye Holes"  
Copyright (C) 2023 The R Foundation for Statistical Computing  
Platform: x86\_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

```
> library(ggplot2); library(datasauRus)
> ggplot(subset(datasaurus_dozen, dataset=="dino"), aes(x, y)) + geom_point()
> [REDACTED]
```

The screenshot shows the R terminal and R Graphics Device. The terminal window at the top shows the command 'R' being run. Below it is the R Graphics Device window titled 'R Graphics: Device 2 (ACTIVE)'. The device shows a scatter plot of data points. The x-axis is labeled 'x' and ranges from approximately 25 to 100. The y-axis is labeled 'y' and ranges from approximately 25 to 100. The plot shows a dense cloud of points forming various shapes, including a large central cluster and several smaller clusters and outliers.

# Slurm job parameters

- Specify in job script or on sbatch command line
  - on command line: `sbatch -c 4 -t 0-4 ...`
  - in job script (must come before any commands):

```
#SBATCH -c 4  
#SBATCH -t 0-4  
...
```

# Slurm job parameters

- Time: `-t` or `--time`
  - format: days-hours or hours:minutes:seconds or minutes
- CPU threads: `-c N`, `--cpus-per-task N` for N threads
- Memory: `--mem`
  - format: nnn for nnn megabytes, or use suffixes [K|M|G|T]
- Defaults if left unspecified: 4 hours, 2 threads, 7 GB memory per thread

# Asking for the right amount of time/memory/CPU

- Think first (for a rough estimate)
- Test run(s) – interactive or batch
- Checking on resource usage:
  - `seff [jobid]` (after the test job finishes)
  - `/usr/bin/time -v [your command...]` (see results in `slurm-$JOBID.out`)
  - `srun --jobid=[jobid] --overlap --pty htop` (monitor the job while it runs)
    - replace “htop” with “bash” for a shell to examine files in job-specific /scratch

# When you asked for too little: Not enough time

```
[robkar@tensor ~]$ salloc -t 1:00
salloc: Granted job allocation 113359
salloc: Nodes tensor4 are ready for job

[robkar@tensor4 ~]$ salloc: Job 113359 has exceeded its time limit and its allocation has been
revoked.
slurmstepd: error: *** STEP 113359.interactive ON tensor4 CANCELLED AT 2024-09-25T12:00:52 DUE TO
TIME LIMIT ***
srun: Job step aborted: Waiting up to 32 seconds for job step to finish.
srun: error: tensor4: task 0: Killed

[robkar@tensor ~]$ seff 113359
Job ID: 113359
...
State: TIMEOUT (exit code 0)
...
```

# When you asked for too little: Not enough memory

```
[robkar@tensor ~]$ sbatch -t 0-1 -c 2 --mem=4G --wrap='ml add R; Rscript -e "x <- rnorm(1e9)"'
Submitted batch job 113367

[robkar@tensor ~]$ cat slurm-113367.out
/var/spool/slurm/d/job113367/slurm_script: line 4: 2590672 Killed          Rscript -e "x <-
rnorm(1e9)"
slurmstepd: error: Detected 1 oom_kill event in StepId=113367.batch. Some of the step tasks have been
OOM Killed.

[robkar@tensor ~]$ seff 113367
Job ID: 113367
...
State: OUT_OF_MEMORY (exit code 0)
...
Memory Utilized: 3.00 MB
Memory Efficiency: 0.07% of 4.00 GB
```

Note that `seff` memory statistics do not always capture spikes in memory usage (but the system OOM killer will)

# More Slurm job parameters

- Job name: `-J` or `--job-name`
  - give your job a nice label in squeue
- Job output files: `-e` and `-o`
  - save your job's error messages and output in named files  
(`-e` and `-o` can point to the same file, by default  
`"slurm-$JOBID.out"`)
- Advanced: job arrays `--array=1-N`
- Advanced: job dependencies `--dependency=...`

# Login vs compute nodes

- Please don't use the login node for computing
  - Impolite: can interfere with others' work
  - Slow: login node tasks are confined to a single core per login and 12G memory
- Instead use `salloc` or `sbatch`
- OK to stay on login node for text editing, file management, other small tasks

# Software modules

- Growing collection of preinstalled software, but must enable packages before using
- Show what's currently available: `ml avail` (or user guide)
- Enable a package: `ml load package/version`
- When a package you need is not in modules:
  - *and only you need it*: /nfs/home/\$USER installation
  - *and many others need it too*: ask for /nfs/sw central installation
- `apptainer` (previously Singularity) is installed for containerized workflows

# Storage folders

- tensor-specific network folders:
  - /nfs/home/\$USER
  - /projects/\$PROJECT
- Regular MEB network folders
  - /cifs/Z/\$USER “Windows Z:”
  - /cifs/P/\$PROJECT “Windows P:”
  - `ml add mebauth` for P/Z access
- tensor node-local storage /scratch/tmp – use it! It is fast! But:
  - **/scratch is machine-(and job-)specific**  
 $\text{tensor1:}/\text{scratch} \neq \text{tensor2:}/\text{scratch}$
  - **/scratch is cleared when the job ends**

# What tensor is not set up to do

- GPU
- Multi-node parallel jobs (MPI)
- Windows-only software
- Really big jobs ( $N = 7$  will be quickly saturated)

# When tensor is not big enough

- NAISS (previously SNIC) national resources
- Clusters Bianca (UU) for sensitive data, Dardel (KTH) for non-sensitive data



# Queue/tensor etiquette

- Do read the manual
- Don't run analyses on the login node
- Don't reserve resources you don't need
- Release interactive sessions when done



Running a long command as the last thing in an interactive session

```
some_command --param 1 --param 2 && exit
```

- Rules and limits for heavy use still under development – your input most welcome!
- When there is a queue, jobs from heavy users get lower priority
  - `sprio` to see the priority for jobs in queue

# part 2: tensor practical

# Practical: simple slurm jobs

- Use VDI (to ensure we all have the same environment) to
  - log in to tensor
  - create a small job script using a specific R version (or Stata)
  - submit to queue (sbatch)
  - check status (squeue)
  - examine output (cat slurm-\$JOBID.out)
- Example scripts available from  
<https://github.com/robkar/meb-tensor-intro/>

**part 3: general Q&A, help  
with specific tasks**