C.E.C.I

Consortium des Equipements
de Calcul Intensif
en Fédération Wallonie-Bruxelles

Preparing, submitting and managing jobs with Slurm

damien.francois@uclouvain.be
October 2022

Until now:

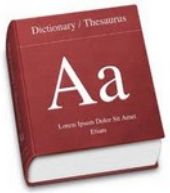
- access the cluster ✔
- copy data to/from the cluster ✔
- choose and activate software ✔
- run software in the command line prompt ✔
- create/write text files ✔
- actually run software on the cluster ❓

tl;dr:

DON'T: run software on the login node

DO: submit a *job* to the *resource manager/job scheduler*

What is a job?



Dictionary

job [1] |jäb|

noun

1 a paid position of regular employment *: jobs are created in the private sector, not in Washington | a part-time job.*

2 a task or piece of work, esp. one that is paid *: she wants to be left alone to get on with the job | you* ***did a good job of*** *explaining.*

- a responsibility or duty *: it's our job to find things out.*
- [in sing.] informal a difficult task *: we thought you'd have a job getting there.*
- [with adj.] informal a procedure to improve the appearance of something, esp. an operation involving plastic surgery *: she's had a nose job | someone had done a skillful paint job.*
- [with adj.] informal a thing of a specified nature *: the car was a blue malevolent-looking job.*
- informal a crime, esp. a robbery *: a series of daring bank jobs.*
- Computing an operation or group of operations treated as a single and distinct unit.

What is a resource manager/scheduler ?

Job scheduler

From Wikipedia, the free encyclopedia

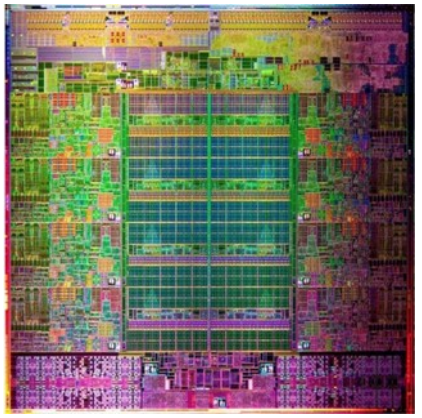
A **job scheduler** is a computer application for controlling unattended background program execution of jobs.[1] This is commonly called **batch scheduling**, as execution of non-interactive jobs is often called batch processing, though traditional *job* and *batch* are distinguished and contrasted; see that page for details. Other synonyms include **batch system**, **distributed resource management system** (**DRMS**), **distributed resource manager** (**DRM**), and, commonly today, **workload automation** (**WLA**). The data structure of jobs to run is known as the job queue.

Resource management (computing)

From Wikipedia, the free encyclopedia

In computer programming, **resource management** refers to techniques for managing resources (components with limited availability).

resources:

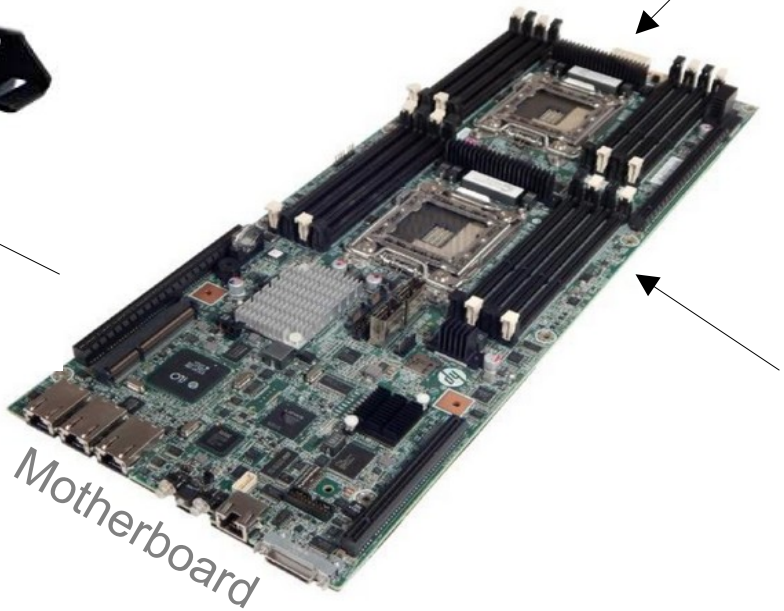


CPU (core)

CPU (socket)

GPU

Compute node

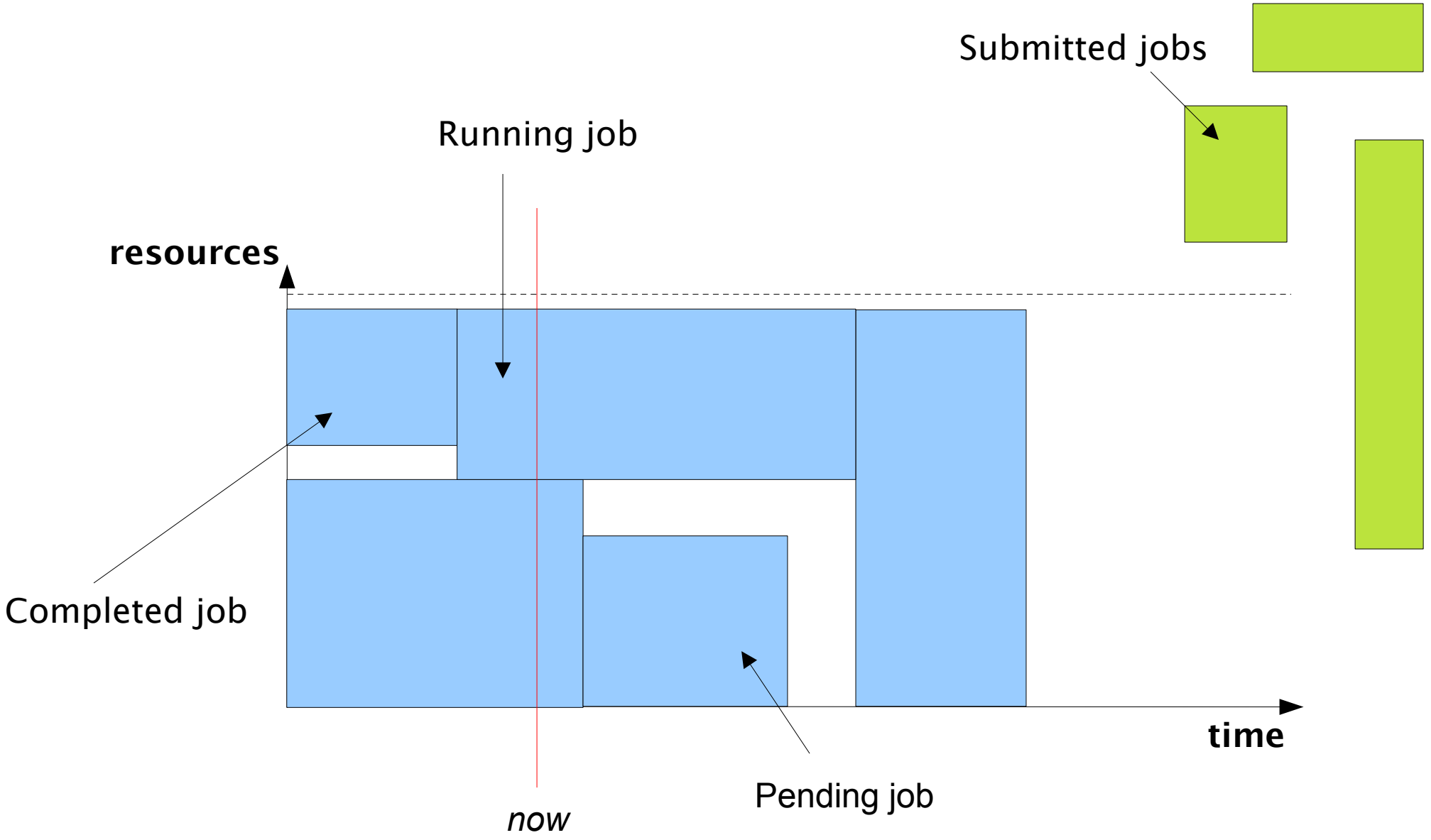


Motherboard

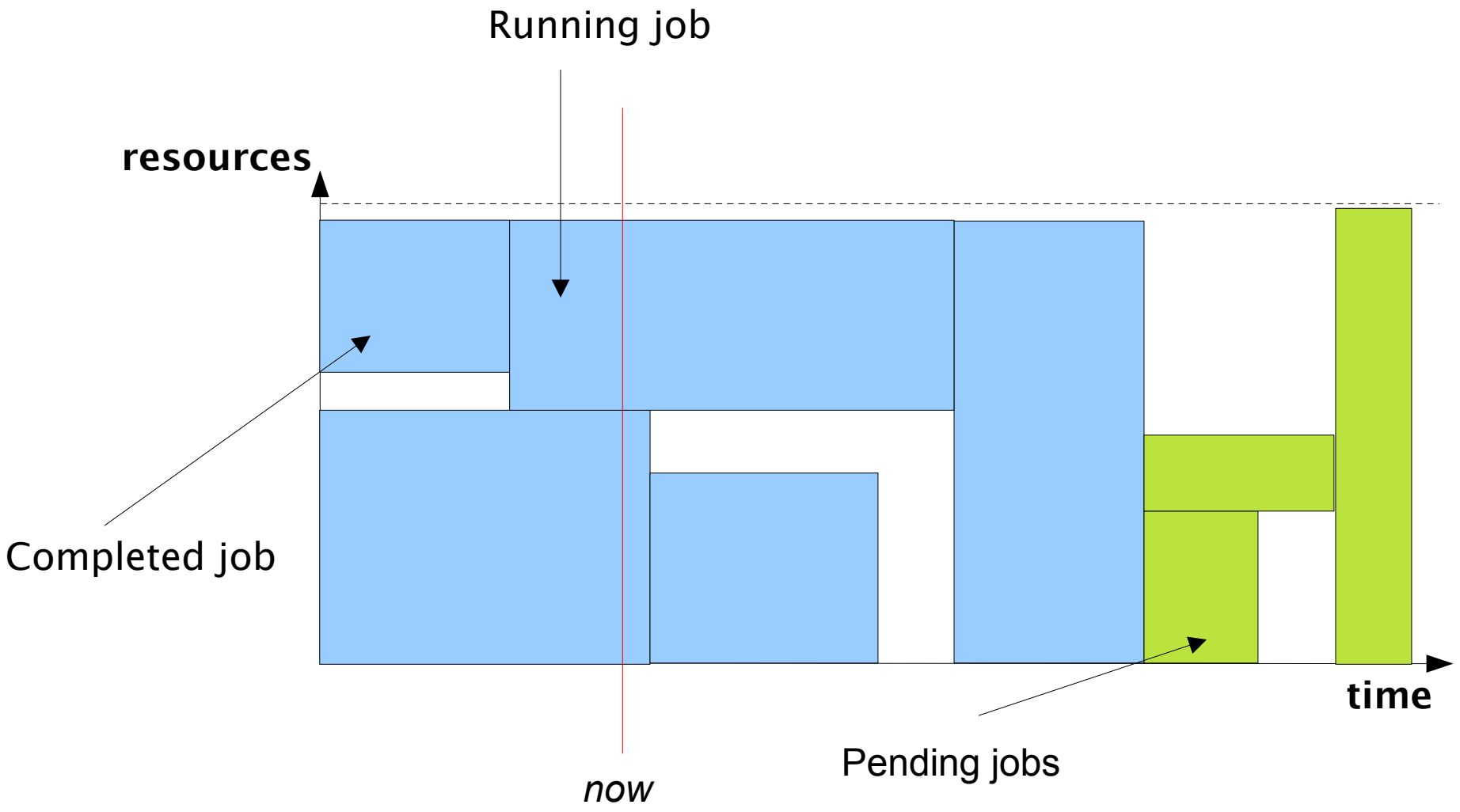
RAM

Disk

scheduling:


Submitted jobs
Running job
resources
Completed job
time
Pending job
now

scheduling:




SCORE
3780
LEVEL
3
LINES
32

scheduling:


Running job
resources
exclusive allocation of resources
to jobs over time
Completed job
time
Pending jobs
now


slurm
workload manager

Slurm

Free and free

Mature (exists since ~2003)

Very active community

Many success stories

Widely used

Also an intergalactic soft drink

Futurama (TV Series, creators David X. Cohen, Matt Groening)
Fry and the Slurm Factory (1999)
20th Century Fox Television

Topics:

1. How to create a job
2. How to choose resources
3. Understand priorities
4. Typical workloads
5. Basic workflows
6. Interactive sessions
7. Workflow management system

Full version: https://www.youtube.com/watch?v=wWWDv4g7TcI

Part I. You will learn how to:

Create and submit a job

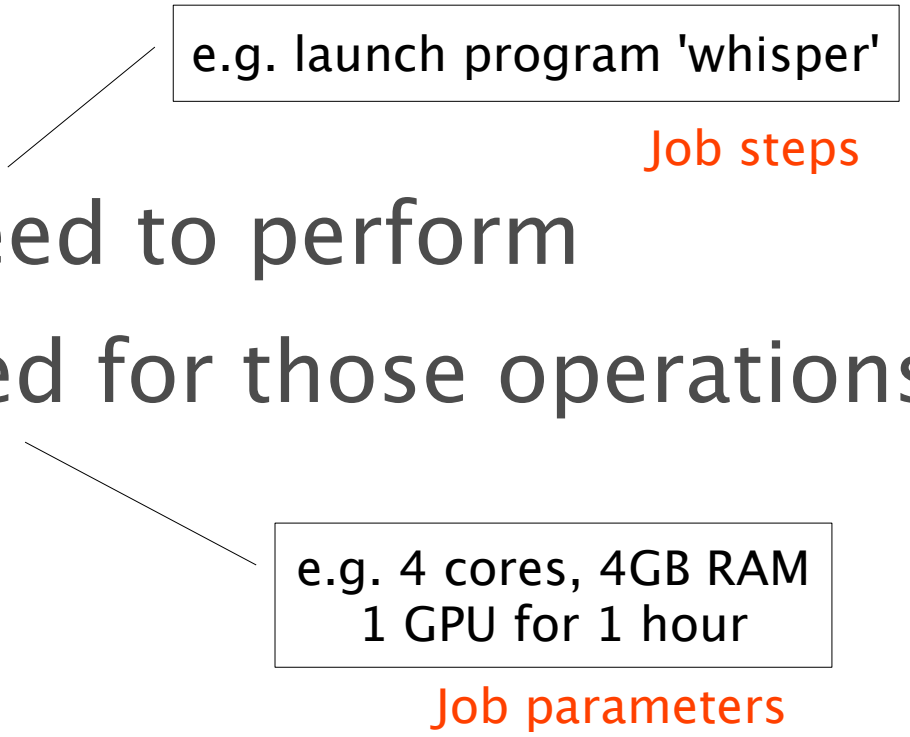
Monitor and inspect jobs

Control (your own) jobs

with

Make up your mind ...

- operations you need to perform
- resources you need for those operations



... then write a submission script...

It is a shell script (Bash)

Bash sees these as comments

Slurm takes them as parameters

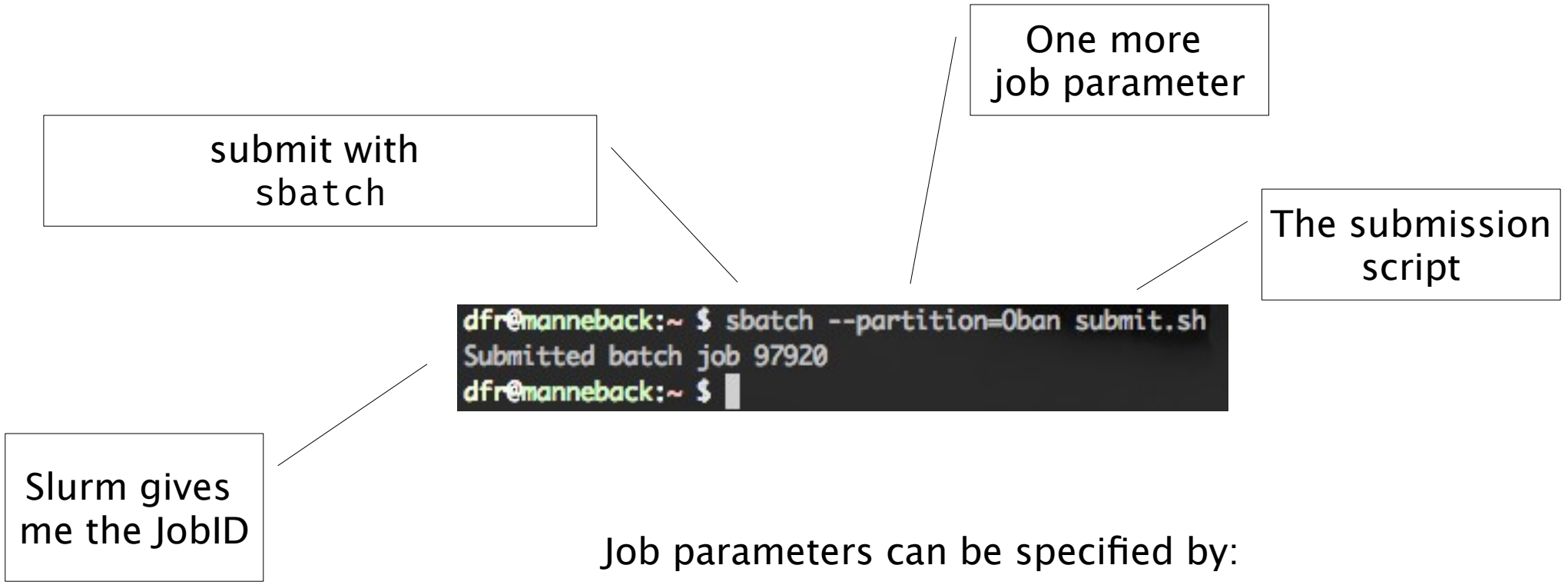
Job step creation

```
#!/bin/bash
# Submission script for demonstrating
# slurm usage.

# Job parameters
#SBATCH --job-name=demo
#SBATCH --output=res.txt
# Needed resources
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=2000
#SBATCH --time=1:00:00

# Operations
echo "Job start at $(date)"
# Job steps
srun ~/bin/myprog < mydata1

echo "Job end at $(date)"
```

Regular Bash comment

Regular Bash commands

... and submit it with sbatch



Job parameters can be specified by:

- #SBATCH directives in the submission script ;
- environment variables ;
- parameters on the `sbatch` command line.

The job ID is used later on to uniquely identify the job.

Submit your first job!

1. Connect to Manneback
2. Open a text editor and write the script for a job that will run the "`hostname`" command
   - The first line must be as in the example
   - All parameters are facultative or have default values
3. Submit the job
4. Look for files created in the directory

Run the squeue command

```
SQUEUE(1)                       Slurm components
                SQUEUE(1)

NAME
       squeue  -  view  information  about jobs
       located in the SLURM scheduling queue.

SYNOPSIS
       squeue [OPTIONS...]

DESCRIPTION
       squeue is used to view job and job  step
       information for jobs managed by SLURM.

OPTIONS
       -A                          <account_list>,
       --account=<account_list>
              Specify  the accounts of the jobs
              to view. Accepts  a  comma  sepa-
              rated list of account names. This
:
```

man squeue

Run the squeue command

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             12324     batch     demo      dfr  R   11:10:02      4 node[001-004]
             12325     batch     demo      dfr PD      00:00      2 (Resources)
             12329     batch   prod_1      bvr PD      00:00      1 (Priority)
             12422     debug   test_2      bvr  R      04:01      1 node005
```

JOBID	the job ID assigned by Slurm
PARTITION	set of nodes the job was submitted to
NAME	name of the job as specified with `--job-name`
USER	username of the user who submitted the job
ST	State of the job: Running, PenDing, ...
TIME	Running time of the job
NODES	Number of nodes requested (`--nodes`)
NODELIST	Nodes assigned to the job by Slurm `node[001-004]` = `node001`, `node002`, `node003`, and `node004`
(REASON)	Reason why the job is pending (`Resources`): your job is next, (`priority`): you need to wait, ...

Use these #SBATCH parameters

You want	You ask
To choose a specific feature (e.g. a processor type or a network type)	`--constraint`
To use a generic resources (e.g. a GPU)	`--gres` (or `--gpu`)
To access a specific licensed software	`--licence`
To chose a partition	`--partition`
To use a specific QOS	`--qos`

To choose the CPU distribution on nodes	`--nodes`
	`--ntasks-per-nodes`
	`--cpus-per-tasks`

Use these #SBATCH parameters

You want	You ask
To set a job name	`--job-name=...`
To attach a comment to the job	`--comment="Some comment"`
To get emails	`--mail-type=BEGIN|END|FAILED|ALL|TIME_LIMIT_90` `--mail-user=my@mail.com`
To set the name of the output file	`--output=result-%j.txt` `--error=error-%j.txt`
To enquiry when it would start	`--test-only`
To specify an ordering	`--dependency=after(ok|notok|any):jobids` `--dependency=singleton`

Full list of options in sbatch manpage

Whisper requires resources

You want	You ask

A GPU with enough memory	`--partition=gpu`
	`--gpus=1`
	`--constraint=Tesla`

Some CPUs and CPU memory	`--cpus-per-task=4` `--mem=16G`
Some time depending on record	`--time=01:00:00`

Whisper requires modules

```
module load releases/2021a
module load whisper whisper-models
```

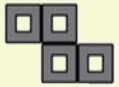
Whisper usage

```
[dfr@mbackf2 ~]$ whisper --help
usage: whisper [-h] [--model {tiny.en,tiny,base.en,base,small.en,small,medium.en,medium,large}]
               [--model_dir MODEL_DIR] [--device DEVICE] [--output_dir OUTPUT_DIR] [--verbose VERBOSE]
               [--task {transcribe,translate}]
               [--language {af,am,ar,as,az,ba,be,bg,bn,bo,br,bs,ca,cs,cy,da,de,el,en,es,et,eu,fa,fi,fo,fr,gl,gu,ha,h
aw,hi,hr,ht,hu,hy,id,is,it,iw,ja,jw,ka,kk,km,kn,ko,la,lb,ln,lo,lt,lv,mg,mi,mk,ml,mn,mr,ms,mt,my,ne,nl,nn,no,oc,pa,pl
,ps,pt,ro,ru,sa,sd,si,sk,sl,sn,so,sq,sr,su,sv,sw,ta,te,tg,th,tk,tl,tr,tt,uk,ur,uz,vi,yi,yo,zh,Afrikaans,Albanian,Amh
aric,Arabic,Armenian,Assamese,Azerbaijani,Bashkir,Basque,Belarusian,Bengali,Bosnian,Breton,Bulgarian,Burmese,Castili
an,Catalan,Chinese,Croatian,Czech,Danish,Dutch,English,Estonian,Faroese,Finnish,Flemish,French,Galician,Georgian,Ger
man,Greek,Gujarati,Haitian,Haitian Creole,Hausa,Hawaiian,Hebrew,Hindi,Hungarian,Icelandic,Indonesian,Italian,Japanes
e,Javanese,Kannada,Kazakh,Khmer,Korean,Lao,Latin,Latvian,Letzeburgesch,Lingala,Lithuanian,Luxembourgish,Macedonian,M
alagasy,Malay,Malayalam,Maltese,Maori,Marathi,Moldavian,Moldovan,Mongolian,Myanmar,Nepali,Norwegian,Nynorsk,Occitan,
Panjabi,Pashto,Persian,Polish,Portuguese,Punjabi,Pushto,Romanian,Russian,Sanskrit,Serbian,Shona,Sindhi,Sinhala,Sinha
lese,Slovak,Slovenian,Somali,Spanish,Sundanese,Swahili,Swedish,Tagalog,Tajik,Tamil,Tatar,Telugu,Thai,Tibetan,Turkish
,Turkmen,Ukrainian,Urdu,Uzbek,Valencian,Vietnamese,Welsh,Yiddish,Yoruba}]
               [--temperature TEMPERATURE] [--best_of BEST_OF] [--beam_size BEAM_SIZE] [--patience PATIENCE]
               [--length_penalty LENGTH_PENALTY] [--suppress_tokens SUPPRESS_TOKENS]
               [--initial_prompt INITIAL_PROMPT] [--condition_on_previous_text CONDITION_ON_PREVIOUS_TEXT]
               [--fp16 FP16] [--temperature_increment_on_fallback TEMPERATURE_INCREMENT_ON_FALLBACK]
               [--compression_ratio_threshold COMPRESSION_RATIO_THRESHOLD] [--logprob_threshold LOGPROB_THRESHOLD]
               [--no_speech_threshold NO_SPEECH_THRESHOLD] [--threads THREADS]
               audio [audio ...]
```

No “one-size-fits-all” optimal set of parameters, default ones ok for most cases

Submit your first useful job!

1. Download MP3 for Abbee Pierre's speeech:
https://universal-soundbank.com/discours-celebres.htm
2. Copy the MP3 file to a new directory on Manneback
3. Create a submission file containing:
   - `#/bin/bash`
   - Resource requests
   - Module loads
   - Whisper launch
4. Submit the job
5. Check its status
6. Inspect files created by Whisper

Part  . You will learn how to:

discover cluster features (resources),
target specific features and tune your jobs,
choose suitable resource values, and
get job actual resource usage.

in your submission scripts for

Use the `squeue -l -j <JOBID>` command

```
[dfr@lemaitre3 ~]$ squeue --me -l
Wed Aug 24 11:00:30 2022
CLUSTER: lemaitre3
             JOBID PARTITION     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
          70786661     batch      dfr  PENDING       0:00      6:00     50 (Resources)
          70786672     batch      dfr  PENDING       0:00      6:00     50 (Priority)
          70786664     batch      dfr  PENDING       0:00      6:00      1 (BeginTime)
          70786673     batch      dfr  PENDING       0:00      6:00      1 (ReqNodeNotAvail)
          70786670     batch      dfr  PENDING       0:00      6:00      1 (Dependency)
          70786657     batch      dfr  PENDING       0:00      6:00      1 (JobHeldUser)
          70786658     debug      dfr  PENDING       0:00      6:00      5 (PartitionNodeLimit)
```

https://slurm.schedmd.com/squeue.html#SECTION_JOB-REASON-CODES

Use the sacct command for **completed** jobs

```
SACCT(1)                          Slurm components
                    SACCT(1)

NAME
       sacct - displays accounting data for all
       jobs and job  steps  in  the  SLURM  job
       accounting log or SLURM database

SYNOPSIS
       sacct [OPTIONS...]

DESCRIPTION
       Accounting  information for jobs invoked
       with SLURM are either logged in the  job
       accounting  log  file  or  saved  to the
       SLURM database.

       The sacct command displays job  account-
       ing  data  stored  in the job accounting
       log file or SLURM database in a  variety
:
```

Use the `sacct` command for **completed** jobs

```
$ sacct --format Jobid,ReqMem,MaxRSS,TimeLimit,AllocCPUS,CPUTime,TotalCPU
       JobID     ReqMem     MaxRSS  Timelimit    Elapsed AllocCPUS    CPUTime  TotalCPU
------------ ---------- ---------- ---------- ---------- ---------- ---------- ---------
   12329           1Gc                00:05:00   00:03:22          2   00:06:44 06:20.781
   12329.ba+       1Gc      13820K                00:03:22          2   00:06:44 06:20.780
   12329.ex+       1Gc       1044K                00:03:22          2   00:06:44 06:20.780
   12329.0         1Gc       1044K                00:00:00          2   00:06:44 00:00.001
   12329.1         1Gc       1044K                00:03:21          2   00:06:44 06:20.780
```

JobID	Job ID . Step ID of the job step
ReqMem	Requested memory (Gc: **G**igaByte per **c**ore)
MaxRSS	Actually-used memory (**R**esident **S**et **S**ize)
Timelimit	Time limit requested for the job with `--time`
Elapsed	Actual time used by the job
AllocCPUs	Number of allocated CPUs to the job
CPUTime	CPUtime allocated to the job (Elapsed * AllocCPUs)
TotalCPU	Actual CPU time consumed by the job

Look at your jobs!

1. Connect to a cluster
2. run the `sacct` command to see your job history

Summary

Write and submit submission scripts

Explore the clusters

Read the man pages of Slurm commands

Use the resources you request

Beware of limits

Submit jobs !