Using Maxwell

HPC Seminar Series Part 2

Maxwell Team
Hamburg, 23.04.2018

Agenda

Basic Information
- Resources & Lists
- What is Maxwell?
- Getting access
- Maxwell specialities
- What is slurm

Getting started
- Submission hosts
- Login to Maxwell
- My first slurm jobs
- Importance of setting limits

SLURM User commands
- Getting infos about the cluster
- Advanced jobs
- Constraints
- Installing your own software
- Parallel jobs
- Running …

Organizing jobs
- Preemption & signals
- Array jobs
- Chaining jobs

Topics - if time permits. If not - we could continue at a later point

Please ask whenever you've got a question etc

Please complain if progress is too fast or too slow etc

Basic Information

Basic Information

Resources & Lists

Type of information	Resource
General Information	https://confluence.desy.de/display/IS/Maxwell
Information about available software	https://confluence.desy.de/display/IS/Software
Slides from the last user meeting	https://indico.desy.de/indico/event/19753/
Maxwell question from Users to IT to Users – no subscription	maxwell.service@desy.de
Maxwell announcements from IT to Users – subscribe! Users to Users – moderated!	maxwell-user@desy.de
Seminar announcements from IT – self subscription	hpc-seminar-announce@desy.de
Feedback/Request for seminar from users – no subscription	hpc-seminar@desy.de
Any questions or issues you might have, or if in doubt	uco@desy.de
Some of the (trivial) scriplet used in the tutorial	/software/slurm/tutorial/
Man pages!	man sinfo …

Hint: Subscribing to lists

- https://lists.desy.de/sympa and subscribe via web. Some lists are only visible upon login
- Send email to sympa@desy.de with empty subject and body subscribe hpc-seminar-announce@desy.de;
  - mailto:sympa@desy.de?body=subscribe hpc-seminar-announce@desy.de
  - Use hyperlinks in the table above

Basic Information

What is the Maxwell cluster?

- A bunch of computers named max-something sharing common features
  - connected via infiniband (fast low latency network) as well as 10GE
  - with the same GPFS home-directory
  - with access to GPFS storage infrastructures
  - with access to BeeGFS storage
  - with access to dCache instances (on demand)
  - with access to identical software stacks
  - with (unusually) high memory/core counts
  - **SLURM controlled and/or serving as a SLURM submission host**

- max-p3a* aka max-fsc, max-fsg, aka desy-ps-cpu, desy-ps-gpu are here **NOT** considered a part of Maxwell here

Basic Information

What does it mean?

	Maxwell	Bird
Bandwidth (2 nodes)	~4000 MB/s	~1800 MB/s
0kb msg latency	~3 usec	~20 usec
4kb msg latency	~6 usec	~17 usec
64kb msg latency	~35 usec	~83 usec
4mb msg latency	~1000 usec	~2200 usec
jobs / node	1	~ #of cores
max # of nodes / job	**100‘s**	**~1**
max # of cores / job	**1000‘s**	**Typical 1**
max GB / core	1500	8
# of GPUs	~70	0
costs per node	**6-25k€**	**<5k€**
# of concurrent jobs	**100‘s**	**10000‘s**

Basic Information

What does it mean?

- **We require that jobs on Maxwell to utilize at least one of the special features**
  - fast, low latency interconnect
  - multi-core capabilities
  - memory utilization
  - [GPUs]

- **This does not apply to group owned maxwell resources**
  - Do whatever you like – but you'll get "monitored" anyway

Basic Information

Getting access

- **You would like to use general (IT) resources in Maxwell (the maxwell and all partitions)**
  - Drop a message to maxwell.service@desy.de briefly explaining your use case, so we have an idea that you really need maxwell resources (rather than bird, grid, etc)

- **You would like to use group owned resources in Maxwell**
  - None of our business!
  - Drop a message to your groups administrators. Only the group admins can grant the resources!
  - See next slide

- **You are uncertain if the maxwell cluster is useful and would like to make some tests**
  - Drop a message to maxwell.service@desy.de briefly explaining your use case. We can provide temporary access using dedicated functional accounts.

- **You intend to run hands-on tutorials, schools, etc on maxwell**
  - Drop a message to maxwell.service@desy.de briefly explaining your use case. We can provide temporary access using dedicated functional accounts.
  - Note: reservation of substantial resources might be difficult.

- **You need software only available on maxwell, or access to GPFS, or BeeGFS**
  - Not good enough a reason.
  - You need group owned resources in such cases

- **Not an option at all**
  - Any commercial/industrial use. This includes services for industry
  - Any commercial/industrial user
  - Crypto coin mining
  - Illegal/cracked software

	Nodes	Cores	GPUs	Resource[-users]	Admin	#Jobs	#Nodes/Job	Time Limit	PreEmption
Maxwell	86	3720	12(+)	maxwell	IT	**4**	**6**	**7 days**	**no**
all	0	22280	71(+)	any resource	all	**10**	**32**	**14 days**	**yes/reque**
cfel	20	864	6	cfel-wgs	CFEL	**-**	**6**	**14 days**	**no**
cms	8	320	8	max-cms-*	CMS-*	**-**	**-**	**24 hours**	**yes/chkpt**
cms-desy	4	160	4	max-cms-desy	CMS-DESY	**-**	**-**	**24 horus**	**no**
cms-uhh	4	160	4	max-cms-uhh	CMS-UHH	**-**	**-**	**24 hours**	**no**
cssb	12	554	6	max-cssb	FS/CSSB	**-**	**-**	**-**	**no**
exfel	183	13104		exfel-wgs	EXFEL	**-**	**16**	**14 days**	**no**
exfel-spb	4	288	4	exfel-theory	EXFEL	**-**	**4**	**7 days**	**no**
exfel-theory	31	1728	6	exfel-theory	EXFEL	**-**	**6**	**14 days**	**no**
exfel-th	9	384	2	exfel-theory	EXFEL	**-**	**6**	**14 days**	**no**
exfel-wp72	18	1056	0	exfel-theory	EXFEL	**-**	**6**	**14 days**	**no**
ferrari	24	960	0	max-ferrari	MPY	**-**	**-**	**-**	**no**
fspetra	1	40	1	max-fspetra	FS	**-**	**1**	**14 days**	**no**
p10	1	64	1	p10 staff+users	automatic	**-**	**-**	**unlimited**	**no**
ps	20	800	6	max-ps	FS/IT	**3**	**1**	**14 days**	**no**
psx	6	240	0	max-psx	FS/IT	**3**	**1**	**7 days**	**no**
petra4/p4	2	80	0	p4(*)	M	**-**	**-**	**14 days**	**no**
uke	16	1024	0	max-uke	FS/IT	**-**	**16**	**21 days**	**no**
upex	183	9224	18	upex	EXFEL	**-**	**16**	**14 days**	**no**

Basic Information

Getting access

- **Verifying access to partitions**
  - ssh to max-wgs. If that doesn't work you certainly don't have any of the maxwell resources
  - IF it works, you have at least one of the resources
  - getent netgroup maxwell-users | grep $USER # if empty you don't have access to maxwell partition. Likewise for all other partitions (except all)
  - module load maxwell tools; my-partitions # lists all partitions you are entitled to use

- **Access to max-fsc, max-fsg**
  - As mentioned: not considered part of maxwell for this seminar
  - Not a partition, so my-partitions won't tell
  - getent netgroup hasy-users | grep $USER # if empty you can't user max-fsc, max-fsg
  - getent netgroup psx-users | grep $USER # for external users. if empty, you can't use desy-ps-cpu, desy-ps-gpu

Basic Information

Maxwell specialties

- **tokens and tickets**
  - Very limited support of KRB5 and AFS on maxwell
    - No tickets or tokens in jobs!
    - No automatic ticket or token renewal
  - NOT needed usually
  - Interactive sessions will inherit tickets, tokens. No automatic renewal

Basic Information

Maxwell specialties - Storage

- Several group specific storage locations (gpfs, beegfs, nfs, …)
  - Talk to your admins!

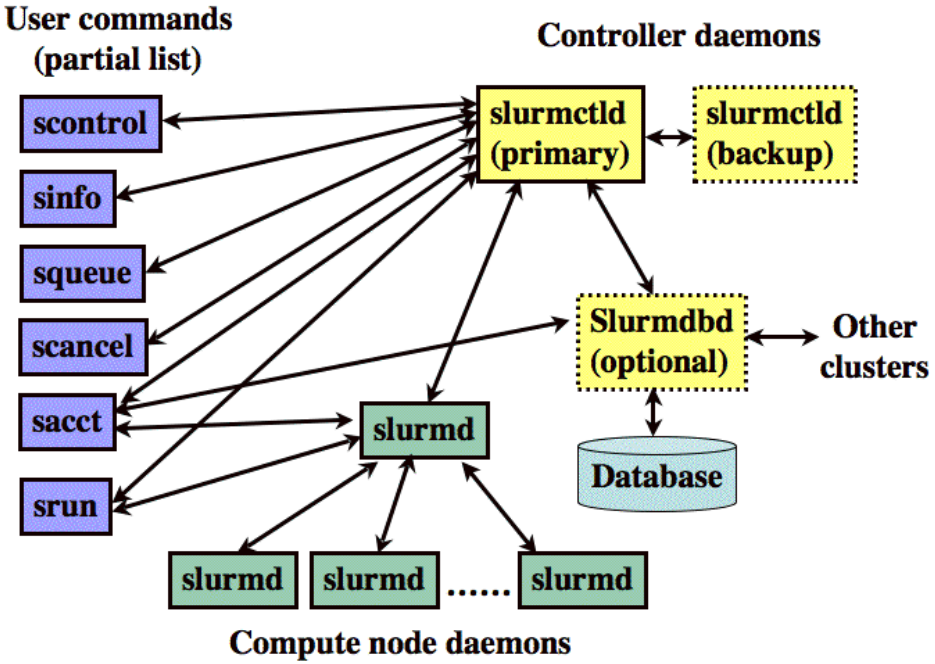
- **generic options**
  - $HOME – GPFS hosted, 20GB hard quota
  - /beegfs/desy – BeeGFS scratch space
  - /data/netapp – NFS space
  - /pnfs/desy.de – dCache space
  - desyCloud – unlimited cloud storage
  - /afs/ – simply avoid it on maxwell

Storage	Location	Features	Purpose	Policy
GPFS	/home/$USER	fast, snapshots	Personal data	remove after account expired
BeeGFS	/beegfs/desy/user	fast, no backup, no snapshots	temporary data	remove after account expired. remove aged data
NFS	/data/netapp	slow, secure, no backup, no snapshots	software	removed after account expired
dCache	/pnfs/desy.de/	slow	scientific data	permanent
desycloud	via gvfs	slow, world-wide accessible, sharable, disaster-recovery in preparation	documents	removed after account expired(?)
afs	/afs/desy.de/	very slow, world-wide accessible, sharable, backup & snapshot (.OldFiles)	documents	archived after account expired

Basic Information

What is slurm

- In simple word, SLURM is a workload manager, or a batch scheduler
- SLURM stands for Simple Linux UTlity for Resource Management
- SLURM unites the cluster resource management (such as Torque) and job scheduling (such as Moab) into one system. Avoids inter-tool complexity.



Getting started

Getting Started

Submission / Login nodes

- Several group specific login & submission hosts
  - Check the documentation
  - Talk to your admins!

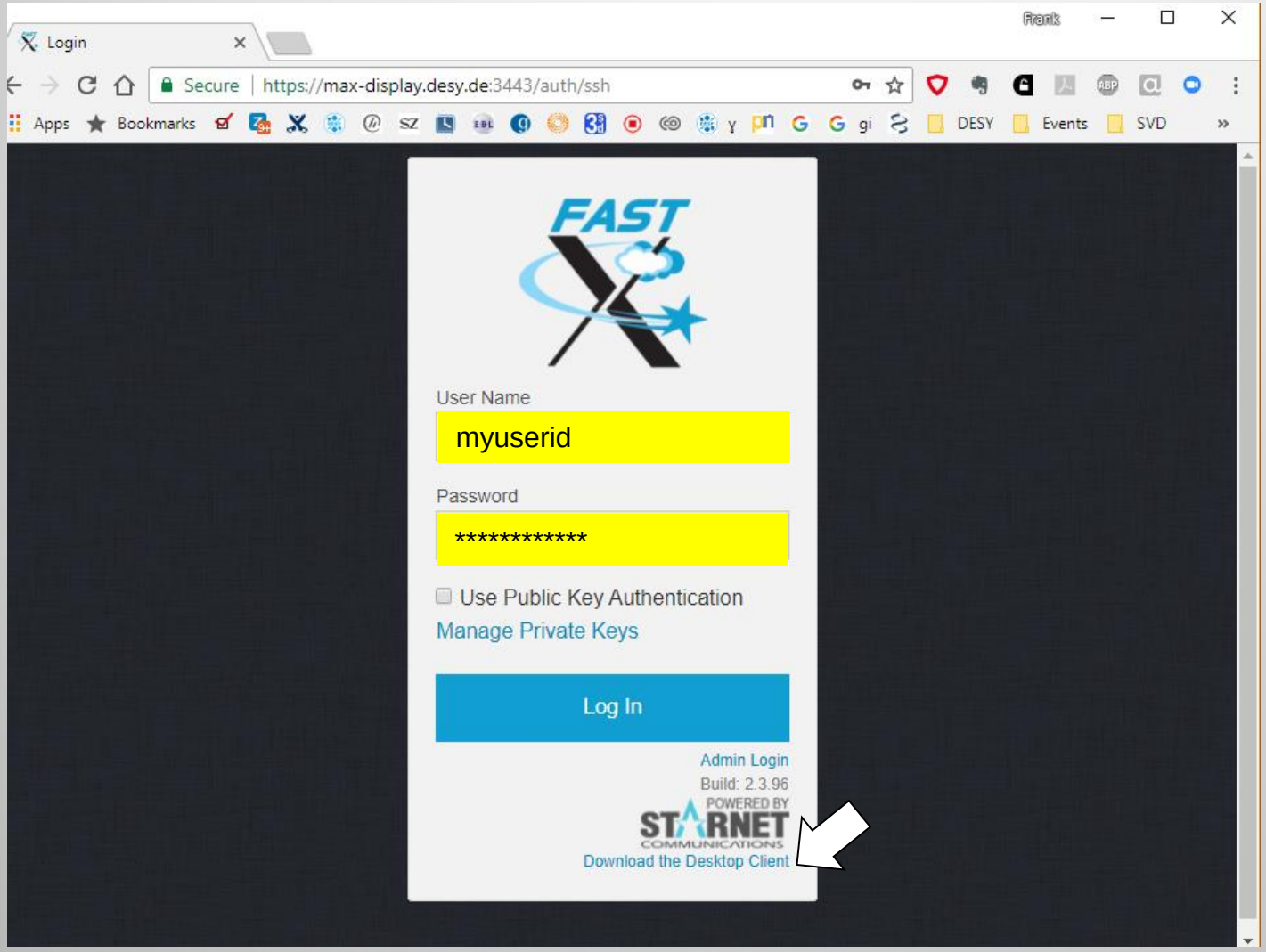
- **generic options**
  - ssh max-wgs.desy.de - connect to WGS
  - https://max-display.desy.de:3443/ - graphical session in your browser
  - fastx2_client - graphical session in FastX2 client

- **what are they good for**
  - composing & submitting jobs! - all nodes.
  - compiling, testing, debugging profiling - use max-wgs
  - graphical applications requiring GPU acceleration - use max-display
  - **NOT**: multi-core, high-memory, long-running production jobs

- **max-display favorable features**
  - load balanced
  - persistent sessions (up to a reboot)
  - GPU accelerated sessions
  - open from outside world (no ssh-tunnel, vpn-client etc)

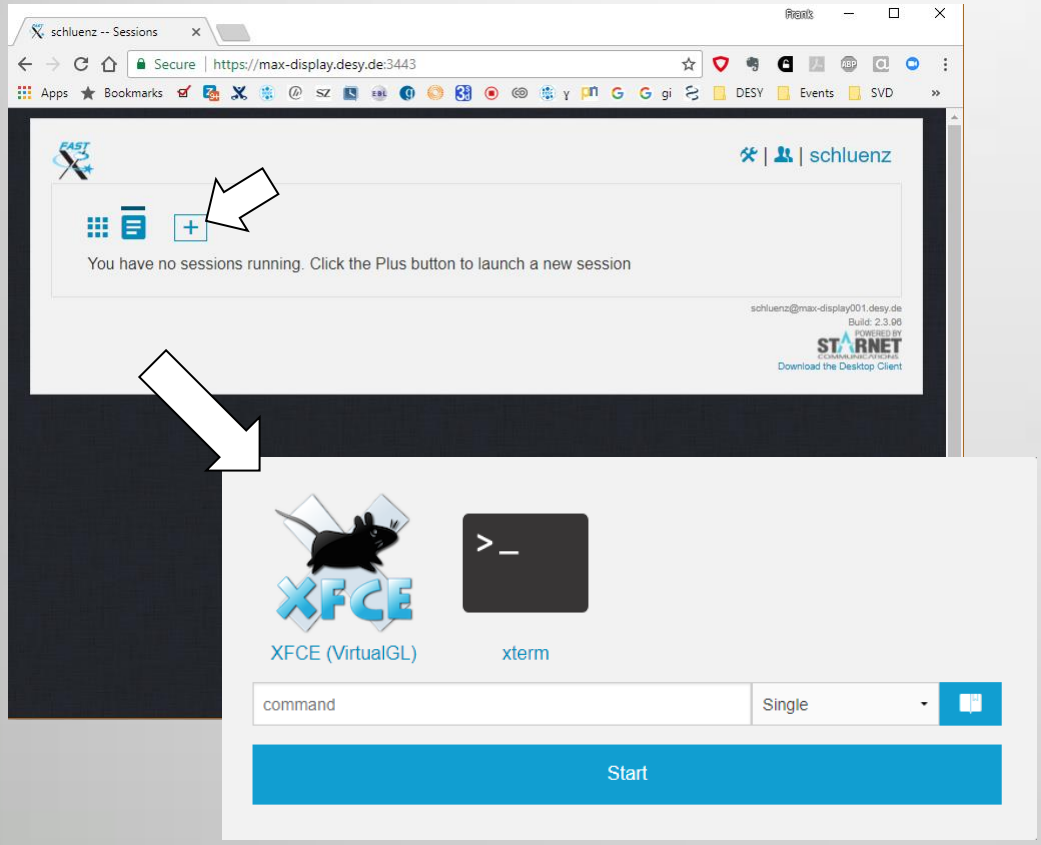
Getting Started

Submission / Login nodes



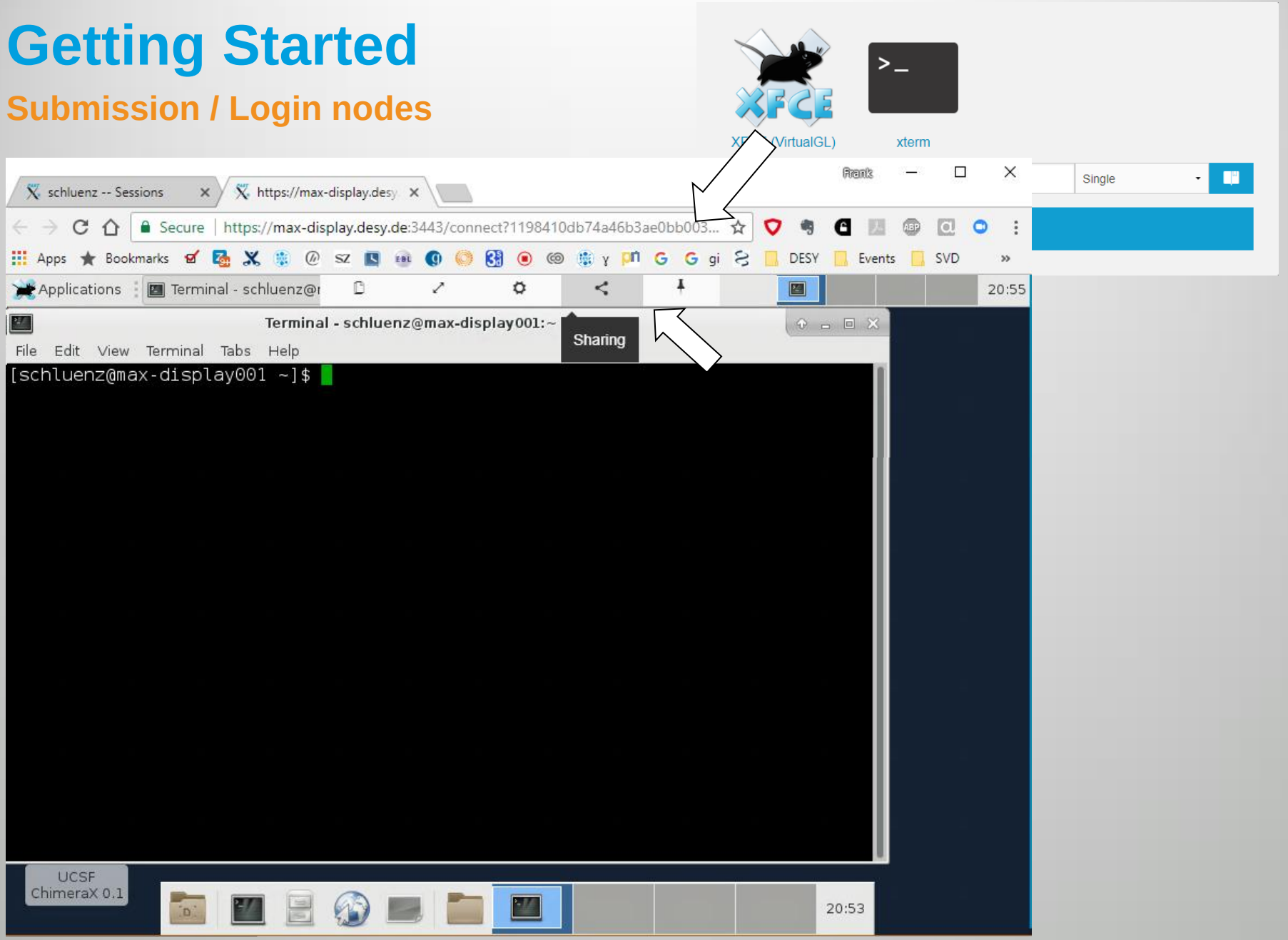
Getting Started

Submission / Login nodes



Getting Started

Submission / Login nodes



Getting Started

Hands-On – Getting Overview of the Cluster

- Login to the cluster
  - ssh max-wgs.desy.de (maybe ssh bastion.desy.de before that)
- Look at the partitions
  - sinfo
  - sinfo –a (all partitions)
  - sinfo –p <name> (specific partition)
- Look at the nodes
  - sinfo -N --long
  - sinfo -a -o "%R %.40N %c %m %f %g" (you can try something else, see SLURM sinfo format help)
- Look at the current queue
  - squeue
  - squeue –u <username>
- sview graphical view -- nothing for „purists“

Getting Started

First Job Script

```
#!/bin/bash
#SBATCH --time      0-00:01:00
#SBATCH --nodes     1
#SBATCH --partition maxwell
#SBATCH --job-name  slurm-01
export LD_PRELOAD="„
source /etc/profile.d/modules.sh
echo "my first job„
echo "SLURM_CLUSTER_NAME    $SLURM_CLUSTER_NAME"
echo "SLURM_JOB_NAME        $SLURM_JOB_NAME"
echo "SLURM_JOB_PARTITION   $SLURM_JOB_PARTITION"
echo "SLURM_SUBMIT_HOST     $SLURM_SUBMIT_HOST"
echo "SLURM_JOB_NUM_NODES   $SLURM_JOB_NUM_NODES"
echo "SLURM_JOB_RESERVATION $SLURM_JOB_RESERVATION"
echo "SLURM_JOB_NODELIST    $SLURM_JOB_NODELIST"
sleep 300
exit
```

```
#!/bin/bash
#SBATCH -t     0-00:01:00
#SBATCH -N     1
#SBATCH -p     maxwell
#SBATCH -J     slurm-01-short
export LD_PRELOAD="„
source /etc/profile.d/modules.sh
echo "my first short job"
sleep 300
Exit
```

See the sbatch manual for flags, environments, etc

- we mostly use the long form
- easier to remember; matter of taste really

Getting Started

First Job Script

```
#!/bin/bash
#SBATCH --time      0-00:01:00
#SBATCH --nodes     1
#SBATCH --partition maxwell
#SBATCH --job-name  slurm-01
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
echo "my first short job"
sleep 30
exit
```

- Specify which shell to use – recommend to use bash

Getting Started

First Job Script

```
#!/bin/bash -l
#SBATCH --time       0-00:01:00
#SBATCH --nodes      1
#SBATCH --partition maxwell
#SBATCH --job-name   slurm-01
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
echo "my first short job"
sleep 30
exit
```

Set the environment

Don’t rely on the host environment

- Define what you need for the job

Getting Started

First Job Script

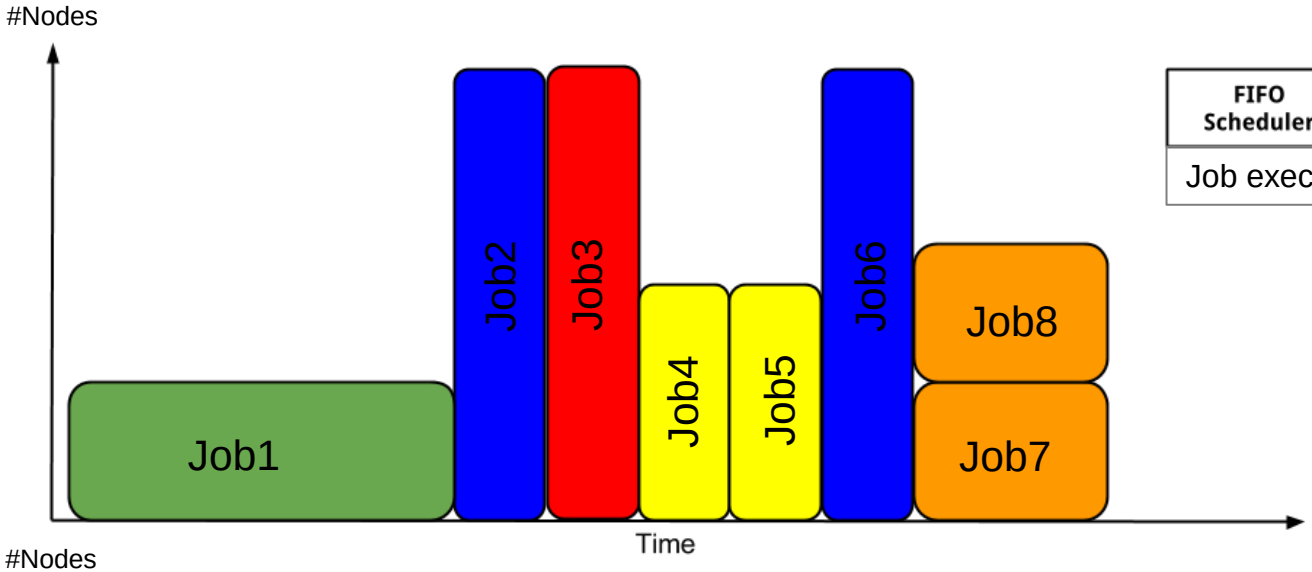
```
#!/bin/bash -l
#SBATCH --time       0-00:01:00
#SBATCH --nodes      1
#SBATCH --partition maxwell
#SBATCH --job-name   slurm-01
export LD_PRELOAD="„
source /etc/profile.d/modules.sh
echo "my first short job"
sleep 30
exit
```

Job directives: instructions for the batch system

Always specify time, partition, number of nodes

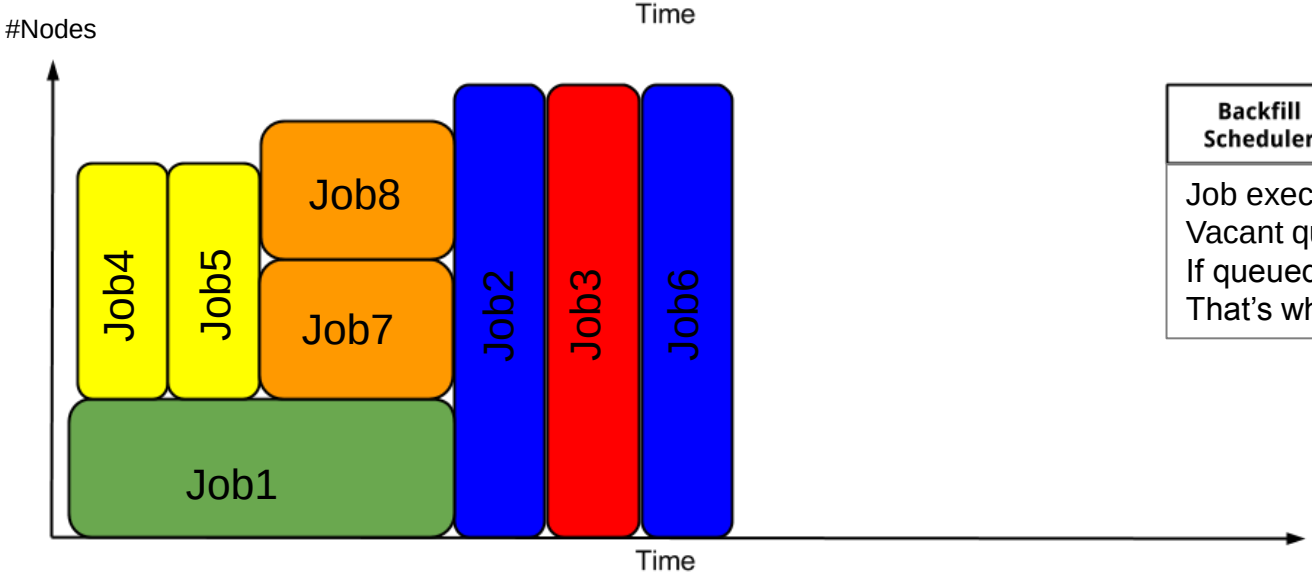
- partition determines limits
- time & number of nodes determine time of execution

Getting Started



FIFO Scheduler

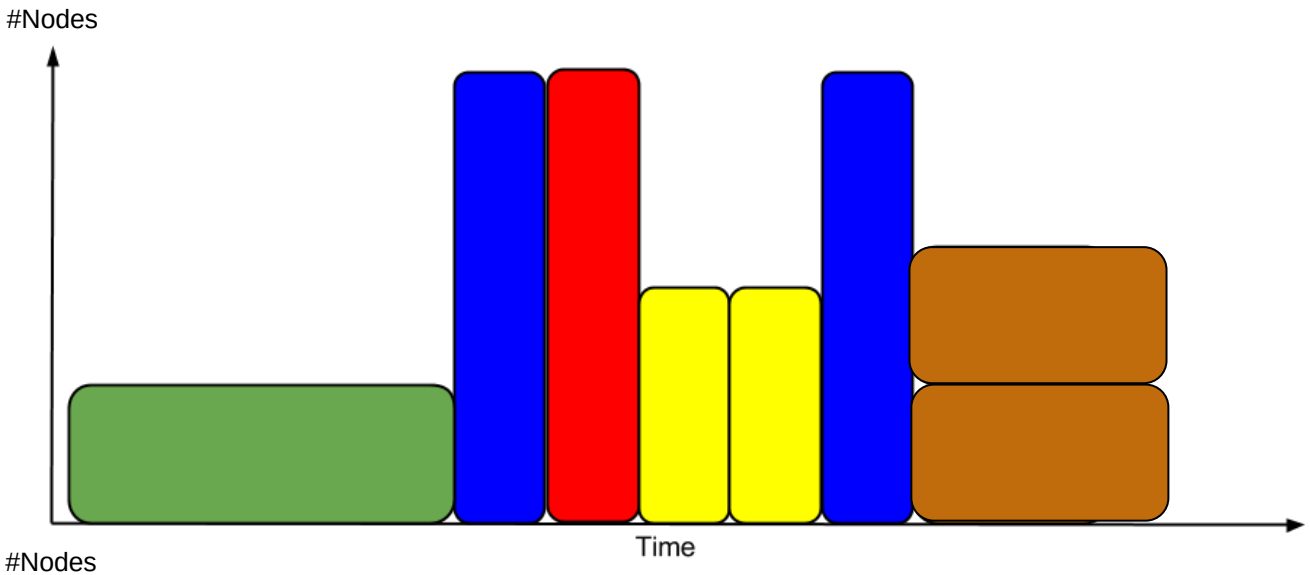
Job executed in order of submission

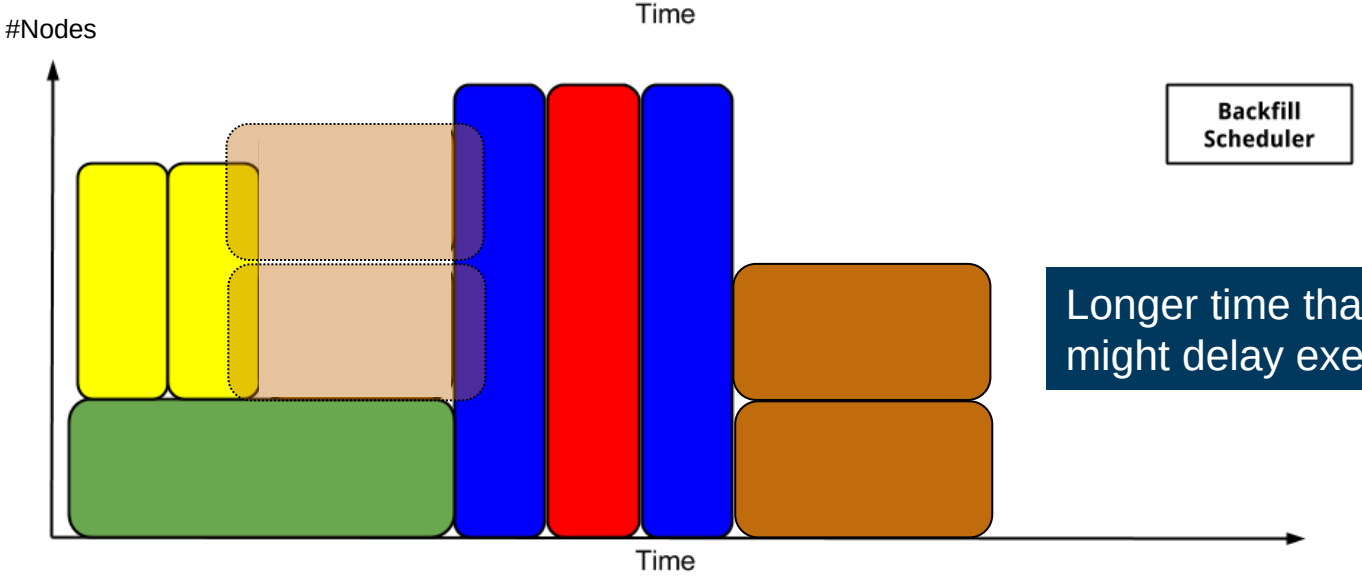


Backfill Scheduler

Job executed in order of submission
Vacant queues are backfilled with jobs –
If queued jobs don't get delayed.
That's what we have in maxwell

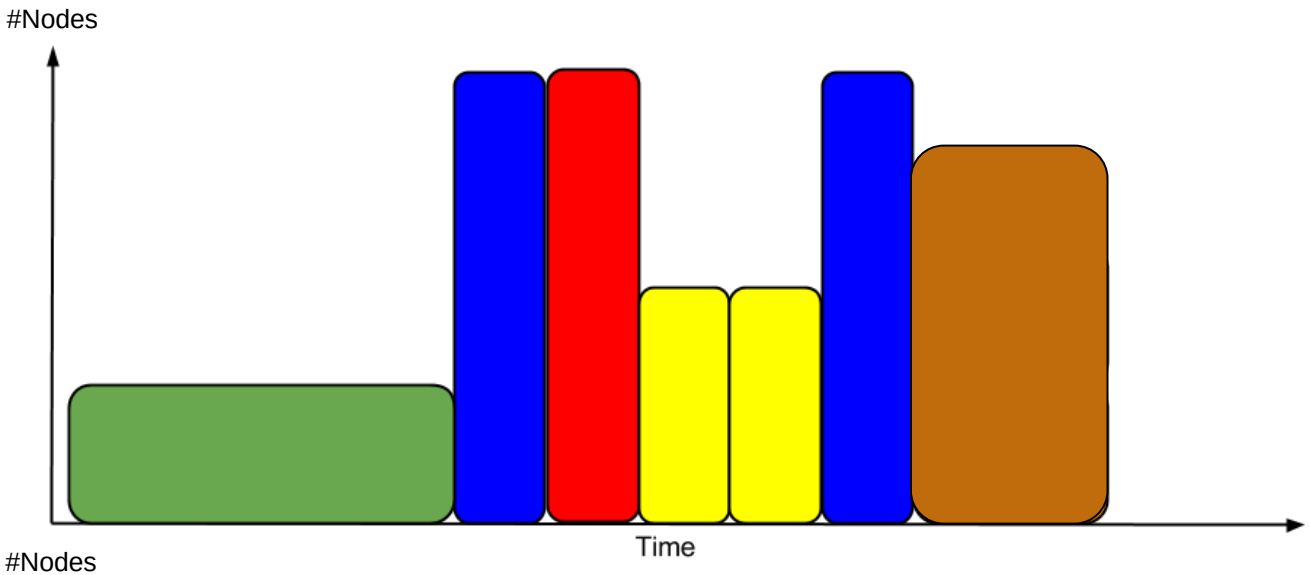
Getting Started

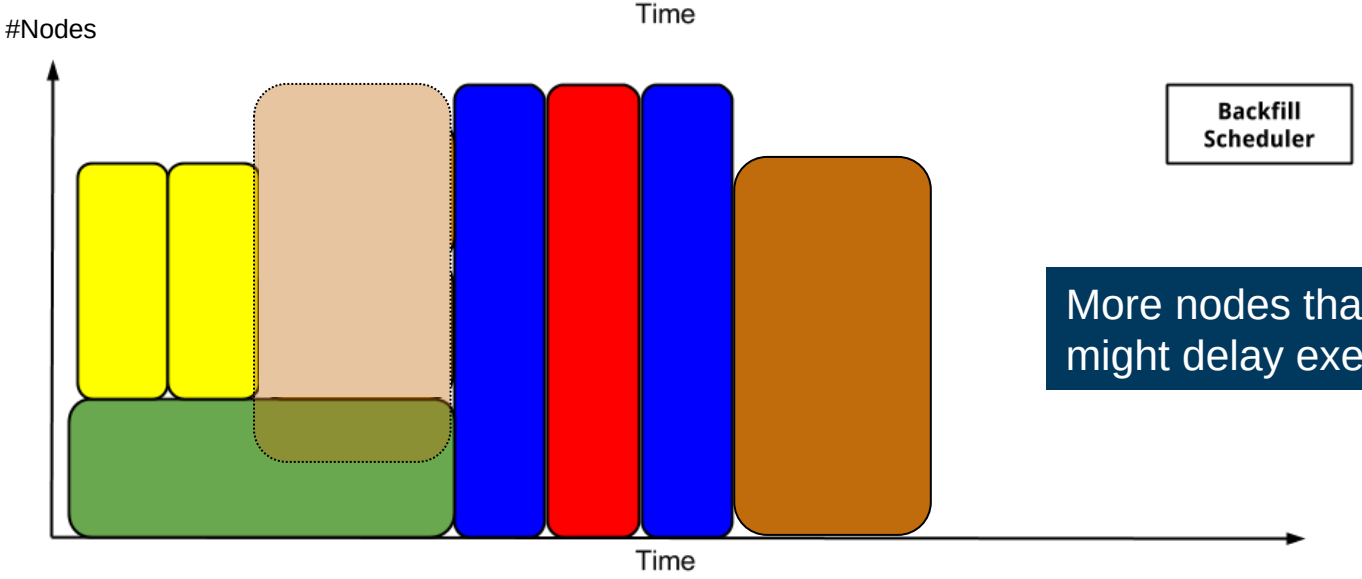




Longer time than needed might delay execution

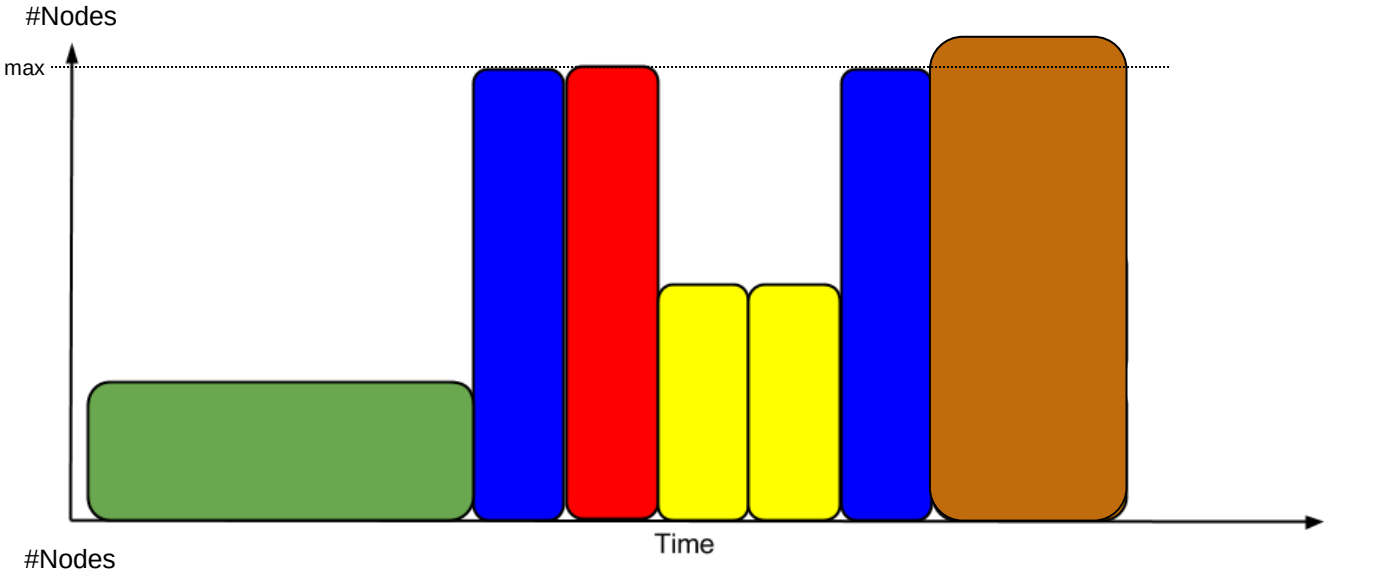
Getting Started

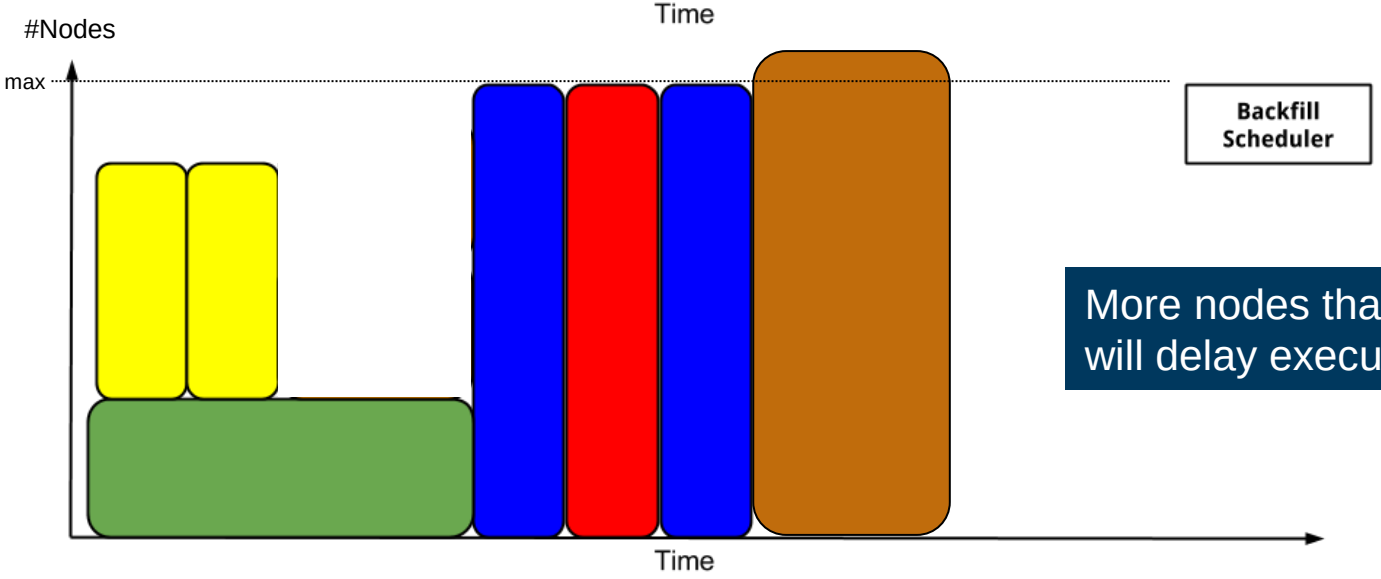




More nodes than needed might delay execution

Getting Started



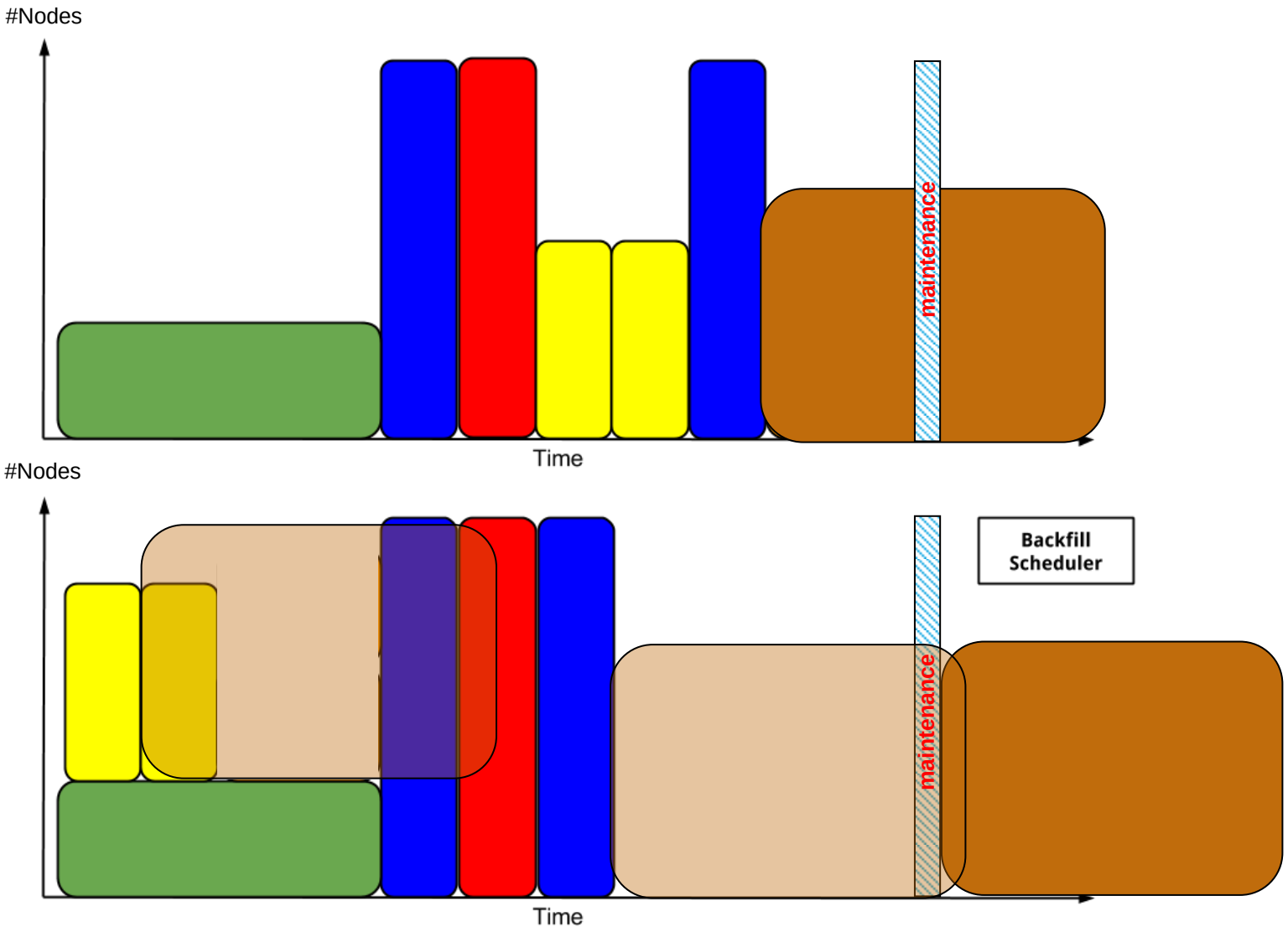


Backfill Scheduler

More nodes than allowed
will delay execution indefinitely

Getting Started

Jobs not fitting into will be delayed until after e.g. maintenance



Getting Started

First Job Script

```
[@max-display002 tutorial]$ sbatch --partition maxwell --time 30-0 slurm-02-no-resources.sh
[@max-display002 tutorial]$ sbatch --partition maxwell --nodes 20 --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 1-0 slurm-01.sh \
                            --constraint has-no-effect
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 0-00:02:00 slurm-01.sh
```

```
[@max-display002 tutorial]$ squeue -u $USER
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
977794   maxwell slurm-02 myuserid PD       0:00     20 (PartitionNodeLimit)
977798   maxwell slurm-01 myuserid PD       0:00     20 (PartitionNodeLimit)
977795   maxwell slurm-02 myuserid PD       0:00      1 (PartitionTimeLimit)
977799       all slurm-01 myuserid PD       0:00      1 (PartitionTimeLimit)
977828       all slurm-01 myuserid PD       0:00      1 (Priority)
977827       all slurm-01 myuserid  R       0:18      1 max-wn009
```

Status: PD pending, R running.

SLURM tells why a job is pending (at least one of the reasons)!

Getting Started

First Job Script

```
[@max-display002 tutorial]$ sbatch --partition maxwell --time 30-0 slurm-02-no-resources.sh
[@max-display002 tutorial]$ sbatch --partition maxwell --nodes 20 --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 1-0 slurm-01.sh \
                            --constraint has-no-effect
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 0-00:02:00 slurm-01.sh
```

```
[@max-display002 tutorial]$ squeue -u $USER
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
977794   maxwell slurm-02 myuserid PD       0:00     20 (PartitionNodeLimit)
977798   maxwell slurm-01 myuserid PD       0:00     20 (PartitionNodeLimit)
977795   maxwell slurm-02 myuserid PD       0:00      1 (PartitionTimeLimit)
977799       all slurm-01 myuserid PD       0:00      1 (PartitionTimeLimit)
977828       all slurm-01 myuserid PD       0:00      1 (Priority)
977827       all slurm-01 myuserid  R       0:18      1 max-wn009
[@max-display002 tutorial]$ sbatch --test-only -p maxwell -N 20 -t 30-0 slurm-01.sh
allocation failure: Requested node configuration is not available
[@max-display002 tutorial]$ scontrol update jobid=977794 NumNodes=1-2
# OR
[@max-display002 tutorial]$ scancel -u $USER
```

- You can test a job without submission using --test-only
- Modify a pending job using scontrol
- Simply cancel all your jobs using scancel

Getting Started

First Job Script

```
[@max-display002 tutorial]$ sbatch --partition maxwell --time 30-0 slurm-02-no-resources.sh
[@max-display002 tutorial]$ sbatch --partition maxwell --nodes 20 --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --time 30-0 slurm-01.sh
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 1-0 slurm-01.sh \
                            --constraint has-no-effect
[@max-display002 tutorial]$ sbatch --partition all --nodes 1 --time 0-00:02:00 slurm-01.sh

[@max-display002 tutorial]$ squeue -u $USER
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
977794   maxwell slurm-02 myuserid PD       0:00     20 (PartitionNodeLimit)
977798   maxwell slurm-01 myuserid PD       0:00     20 (PartitionNodeLimit)
977795   maxwell slurm-02 myuserid PD       0:00      1 (PartitionTimeLimit)
977799       all slurm-01 myuserid PD       0:00      1 (PartitionTimeLimit)
977828       all slurm-01 myuserid PD       0:00      1 (Priority)
977827       all slurm-01 myuserid  R       0:18      1 max-wn009
[@max-display002 tutorial]$ sbatch --test-only -p maxwell -N 20 -t 30-0 slurm-01.sh
allocation failure: Requested node configuration is not available
[@max-display002 tutorial]$ scontrol update jobid=977794 NumNodes=1-2
# OR
[@max-display002 tutorial]$ scancel -u $USER
```

Options on the command line override settings in the script!

Be aware: options after the job-script have no effect!

Getting Started

First Job Script

```
#!/bin/bash -l
#SBATCH -t      0-00:01:00
export MAX_NODES=8
#SBATCH -N      1-$MAX_NODES   ← this and the following are ignored!
#SBATCH -p      maxwell
#SBATCH -J      slurm-01-short
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
echo "my first short job"
sleep 30
exit
```

Likewise:

any options after the first SHELL command will be ignored

BIRD/SGE in contrast will scan the entire file for job directives

Getting Started

Hands-On – Create Job Script and Submit a Job

- Login to the cluster
  - ssh max-wgs.desy.de (maybe ssh bastion.desy.de before that)
  - create a script example1.sh

```
#!/bin/bash
#SBATCH --time      0-00:01:00
#SBATCH --nodes     1
#SBATCH --partition maxwell
#SBATCH --job-name  slurm-01-yourname

echo "my first job"
sleep 12000
```

- Submit the job: sbatch example1.sh
- Check it is running: squeue –u <yourname>, job id will be shown
- Check job output
- ssh into node allocated to job and check process list; log out
- Kill the job on the submit host: scancel <jobid>

Getting Started

Hands-On –Submit a Job To a Node with a Specific Feature

- Use --constraint parameter
  - create a script example2.sh

```
#!/bin/bash
#SBATCH --time       0-00:01:00
#SBATCH --nodes      1
#SBATCH --partition maxwell
#SBATCH --job-name   slurm-02-yourname
#SBATCH --constraint E5-2640

lscpu
```

```
#!/bin/bash
#SBATCH --time       0-00:01:00
#SBATCH --nodes      1
#SBATCH --partition all
#SBATCH --job-name   slurm-02-yourname
#SBATCH --constraint GPU

nvidia-smi
```

- Submit the job: sbatch example2.sh
- Check job output

Getting Started

Hands-On –Submit a Job To a Node with a Specific Feature

```
# to list all valid combinations of features
[@max-display002 bin]$ sinfo -p all -o "%10c %10m %45f"

# To know how many nodes with P100 and other features are available
[@max-display002 bin]$ sinfo -p all -o "%30n %10c %10m %45f" | grep P100 | \
                       awk '{print $4}' | sort -k 1 | uniq -c

# ------------------------------------------------------------------------

-p all --constraint="P100"        # will get you a node with 1-4 P100 GPUs
-p all --constraint="P100|K40X"   # either P100 or K40X. If both are available,
                                  # it will be a K40X.
-p all --constraint="P100&GPUx2" # one of the nodes with dual P100

# ------------------------------------------------------------------------

[@max-display002 bin]$ sbatch --test-only -p all --constraint="P100&K40X" [...]
# would tell you if your combination of constraints is valid
# and when to expect execution
```

Getting Started

Hands-On – Load module with MPI and start an MPI job

- Check available modules: module avail
- Load module in the script

```
#!/bin/bash
#SBATCH --time      0-00:01:00
#SBATCH --nodes     2
#SBATCH --partition maxwell
#SBATCH --job-name  slurm-01-yourname
#SBATCH --mail-type ALL
#SBATCH --mail-user <my-mail@desy.de>

module load mpi/openmpi-x86_64
mpirun hostname
```

- Check job output
- Check your email

Getting Started

Hands-On – Install Your Own Software

- module load maxwell; module avail # lots of software pre-installed. If not:
- Python
  - create requirements.txt file containing the modules you need

```
argparse>=1.2.1
datetime
dill
```

  - pip install --user -r requirements.txt
- C/C++ and Co
  - downloads sources
  - unpack, configure (--prefix $HOME/opt/…), compile, install
  - set paths to binary/libraries
  - Try to install zlib: https://zlib.net/zlib-1.2.11.tar.gz

Advanced Topics

Advanced topics

Hands-On – Requeue and catching signals

- ***all*** partition offers much more than any other partition
  - more concurrent jobs
  - more nodes per job
  - Many more nodes
- but jobs might be killed (preempted)
  - Your choice

```
# do not restart a job
#SBATCH --no-requeue
# restart a job, the default
#SBATCH --requeue
```

- preempted jobs first get a SIGHUP, then a SIGKILL (use kill –l to list SIGNALS)
- Your job gets 5 minutes before it's being terminated
  - if you're able to catch the signal!

Advanced topics

Hands-On – Requeue and catching signals

- Find an idle node in the maxwell partition (sinfo)
- Submit a job to an idle node using the all partition
- Submit a job to the same node using the maxwell partition
- Watch squeue…

Advanced topics

Hands-On – Requeue and catching signals - bash

```
#!/bin/bash

function sighup {
   echo "signal HUP received"
   mkdir -p /scratch/$USER
   echo "trapped" > /scratch/$USER/trapped
}

trap sigquit QUIT
trap sigint  INT
trap sighup  HUP

echo "test script started. My PID is $$"

sleep 30
```

- Start the script in a terminal
- Open another terminal and send the HUP signal to that process
- Wait for the sleep to finish

Advanced topics

Hands-On – Requeue and catching signals - C

```
#include<stdio.h>
#include<signal.h>
#include<unistd.h>

void sig_handler(int signo)
{
  if (signo == SIGHUP)
    printf("received SIGHUP\n");
}

int main(void)
{
  if (signal(SIGHUP, sig_handler) == SIG_ERR)
  printf("\ncan't catch SIGHUP\n");
  // A long long wait so that we can easily issue a signal to this process
  while(1)
    sleep(1);
  return 0;
}
```

Stolen from: https://www.thegeekstuff.com/2012/03/catch-signals-sample-c-code/

Advanced topics

Hands-On – Requeue and catching signals - python

```
#!/usr/bin/env python
import signal
import sys
def signal_handler(signal, frame):
        print(' HUP received!')
        sys.exit(0)
signal.signal(signal.SIGHUP, signal_handler)
print(' Waiting for HUP')
signal.pause()
```

Advanced topics

Hands-On – Requeue and catching signals

- Find an idle node in the maxwell partition (sinfo)
- Submit a signal-catching job to an idle node using the all partition
- Submit a job to the same node using the maxwell partition
- Watch squeue…

Advanced topics

Hands-On – Job arrays

```
#!/bin/bash
#SBATCH --time    0-00:01:00
#SBATCH --nodes             1
#SBATCH --partition all
#SBATCH --array 1-10
#SBATCH --job-name job-array
#SBATCH --output array-%A_%a.out
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
echo "SLURM_JOB_ID           $SLURM_JOB_ID"
echo "SLURM_ARRAY_JOB_ID     $SLURM_ARRAY_JOB_ID"
echo "SLURM_ARRAY_TASK_ID    $SLURM_ARRAY_TASK_ID"
echo "SLURM_ARRAY_TASK_COUNT $SLURM_ARRAY_TASK_COUNT"
echo "SLURM_ARRAY_TASK_MAX   $SLURM_ARRAY_TASK_MAX"
echo "SLURM_ARRAY_TASK_MIN   $SLURM_ARRAY_TASK_MIN"
sleep 10
exit
```

```
--array 1-0                 # run through 1-10
SLURM_JOB_ID                # increments for each job
SLURM_ARRAY_JOB_ID          # job id of the parent job    == %A
SLURM_ARRAY_TASK_ID         # interates over –array       == %a
# use squeue and check the output files!
```

Advanced topics

Hands-On – Job dependencies

```
#!/bin/bash
#SBATCH --time    0-00:01:00
#SBATCH --nodes             1
#SBATCH --partition all
# wait for all jobs with the same name to finish before running this one
#SBATCH --dependency singleton
#SBATCH --job-name singleton
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
sleep 10
exit
```

```
[@max-display001 ~]$ for i in {1..10}; do sbatch singleton.sh; done
[myuserid@max-display001 ~]$ squeue -u $USER
   JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
  979510       all singleto myuserid PD       0:00      1 (Dependency)
  979511       all singleto myuserid PD       0:00      1 (Dependency)
  979512       all singleto myuserid PD       0:00      1 (Dependency)
  979513       all singleto myuserid PD       0:00      1 (Dependency)
  979514       all singleto myuserid PD       0:00      1 (Dependency)
  979515       all singleto myuserid PD       0:00      1 (Dependency)
  979516       all singleto myuserid PD       0:00      1 (Dependency)
  979517       all singleto myuserid PD       0:00      1 (Dependency)
  979509       all singleto myuserid  R       0:09      1 max-exfl079
```

Advanced topics

Hands-On – Running mathematica, matlab, consol … in parallel

In most cases quite trivial, check the documentation. Some examples

- Matlab: https://confluence.desy.de/display/IS/Parallel+Matlab+on+Maxwell
- Comsol: https://confluence.desy.de/display/IS/Parallel+COMSOL+on+Maxwell
- Mathematica:

```
#!/bin/bash
#SBATCH --time    0-00:01:00
#SBATCH --nodes             1
#SBATCH --partition all
#SBATCH --job-name mathematica
export LD_PRELOAD=""
source /etc/profile.d/modules.sh
module load mathematica
export nprocs=$((`/usr/bin/nproc` / 2))  # use half of available cores (HT)
math -noprompt -run '<<math-trivial.m'
exit
```

```
tmp = Environment["nprocs"]
nprocs = FromDigits[tmp]
LaunchKernels[nprocs]
Do[Pause[1];f[i],{i,nprocs}] // AbsoluteTiming        >> "math-trivial.out"
ParallelDo[Pause[1];f[i],{i,nprocs}] // AbsoluteTiming  >>> "math-trivial.out"
Quit[]
```

- Fin –

any feedback highly appreciated! hpc-seminar@desy.de