DESY HPC User Meeting – Operational Model

> Submitting HPC Jobs - Scheduling and Reservation

- Role of a job scheduler
- Introduction to SLURM
- Group specific resources and their integration
- Reservation model
- Beta feature – Docker container virtualization

Role of a Job Scheduler

- Usually there is more work than resources
  - Need to share resources
- Job scheduler manages queue(s) of jobs supporting complex scheduling algorithms
  - Provides fair resource sharing
  - Supports resource limits (user, group, etc.)
  - Optimized for network topology
  - Supports reservations
- Does it automatically
  - No big effort for users
  - Less effort for us
- Provides most efficient usage of the resources

Introduction to SLURM

- Simple Linux Utility for Resource Management
  - Manages resources on the compute nodes
  - Schedules jobs using those resources
  - About 500,000 lines of C code
  - Free and Open source (GPL license, active world-wide development, plugins)
  - Fast (1000 job submissions per second)
  - Fault-tolerant
  - Used by many supercomputer centers (including top ones)

Introduction to SLURM

> Resources in SLURM

- Nodes – description of a compute node (number of CPUs, memory, etc.)
- Partitions – group of nodes with specified properties/restrictions

```
$ cat /etc/slurm/slurm.conf
```

> Basic commands

- **sinfo** - information about nodes and partitions
- **squeue** – shows current job queue
- **sbatch** – submits a job in a batch mode
- **salloc** – request resources for an interactive job

> More info

- http://slurm.schedmd.com/
- https://confluence.desy.de/display/IS/Using+the+Maxwell+Cluster

Introduction to SLURM

> Batch job

```
$ sbatch my_job.sh
```

my_job.sh

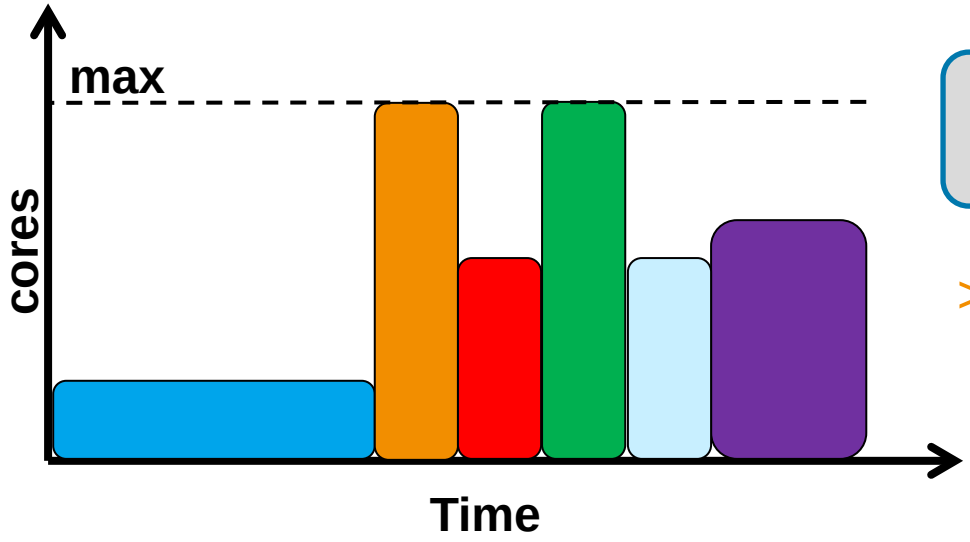
```
#!/bin/bash

#SBATCH --ntasks=128
#SBATCH --nodes=2
#SBATCH --cpus-per-task=1
#SBATCH --partition=all
#SBATCH –time=00:30:00

module load mpi/openmpi-x86_64 intel
mpirun -n 128 /home/yakubov/opt/benchmarks/hpcg-master/bin/xhpcg
```

- Job output to <jobid>.out

> Interactive job

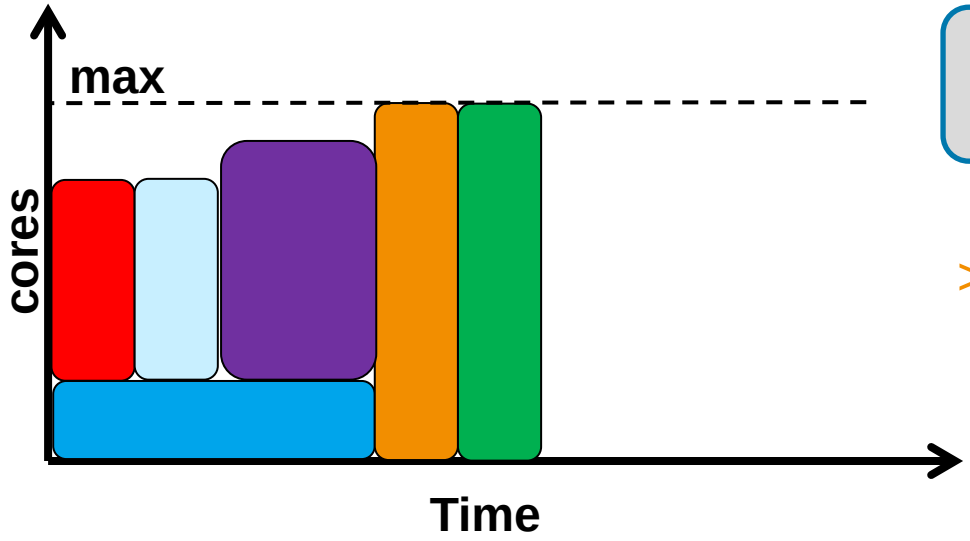
```
$ salloc --nodes 1 --partition=all
```

Introduction to SLURM – Schedulers



FIFO

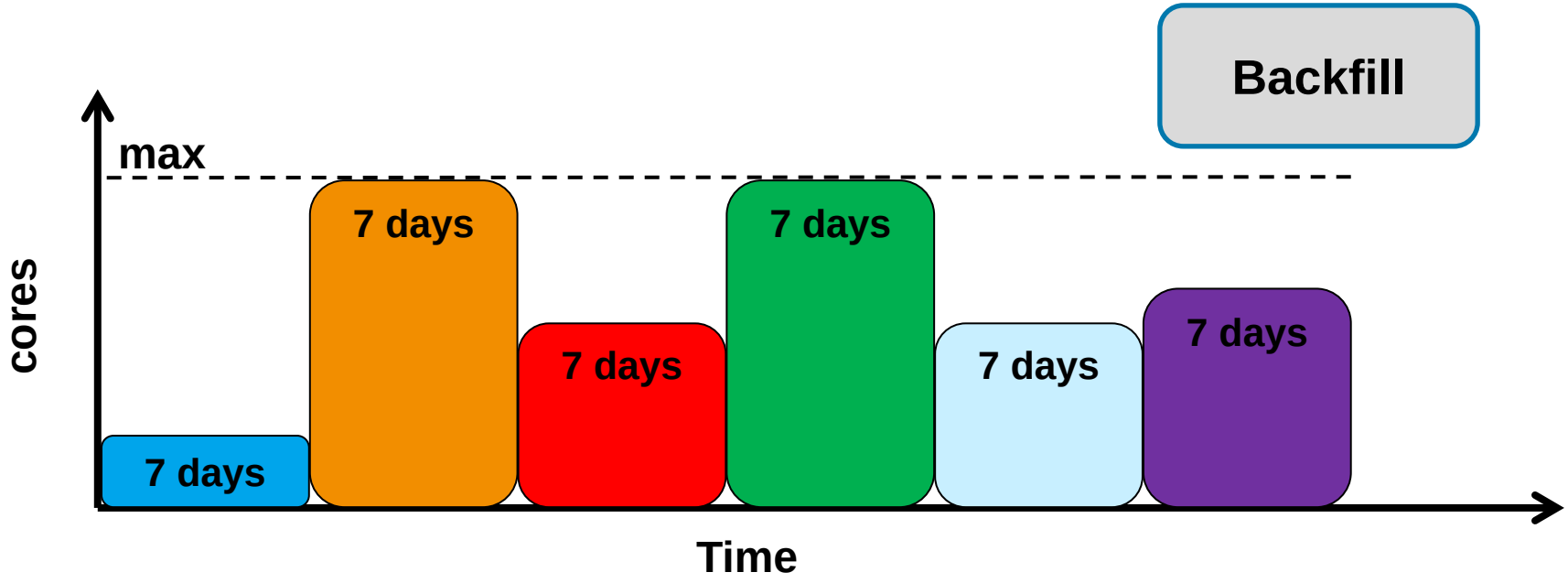
> Jobs started strictly in priority (submission) order



Backfill

> Allow to start lower priority jobs if it won't delay start of higher priority ones

Introduction to SLURM – Schedulers



Will not work if you don't set job time appropriately!

#SBATCH –time=…

Group specific resources and their integration

- All is/will be done via SLURM partitions
- Partition *maxwell*
  - Includes IT group nodes
  - Every registered user can use it
- Group can have “their” partition
  - Contains compute nodes assigned to the group
  - Only users of a group can submit job (no need to get registered as Maxwell user)
  - Has higher priority than common partition
- Common partition *all*
  - Includes all compute nodes
  - Every Maxwell user can submit a job
  - Jobs from common partition will be preempted (killed) if running on a nodes of other groups
  - Supposed to be used for short(er) jobs

Group specific resources and their integration

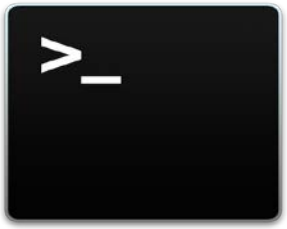
- Currently several groups are using their nodes as workgroup servers
  - Direct access to the nodes
  - No active resource scheduling
  - Inefficient
  - Is not shared with other users when not in use
- Can be included into a queuing system
  - Petra III analysis is an exception (for now) as they need direct access

Reservation model

- We will provide several nodes for use via a new reservation system
  - In-house python-based software (hpcReservations by Lene Stampa)
  - An improvement to the old it-hpc-reservation system
  - Can automatically provide the next free slot
  - Fair-use will be implemented
  - Rely on SLURM reservations
- Constraints
  - Account past reservations of the user (not of the group) - the user's total share
  - Booking will be possible 24 hours from the time of the request
  - And maximum 2 months in advance

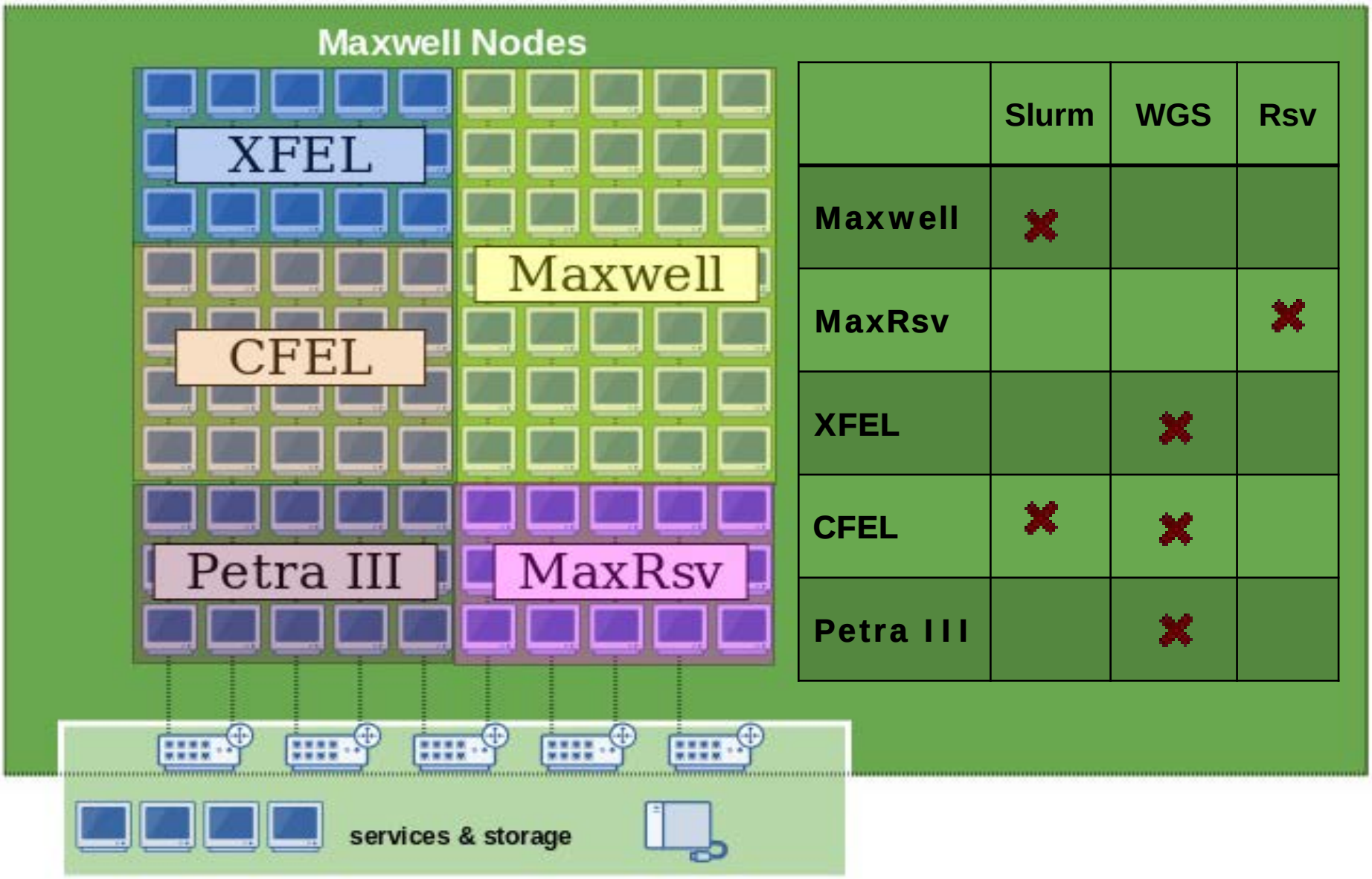
Reservation model - example

- User requests
  - Reservation of 8 nodes for 4 hours
  - Need special node max-… (GPU, …)
  - Between Monday and Thursday
  - Between 9-00 and 17-00
  - **Give me the earliest timeslot !**
- hpcReservations
  - REST-API
  - Web-interface
  - Command line interface
  - **Suggest x earliest timeslots**
- User
  - **Reserve preferred timeslot**



or

Group specific resources and their integration



	Slurm	WGS	Rsv
Maxwell	✖		
MaxRsv			✖
XFEL		✖	
CFEL	✖	✖	
Petra III		✖	

Beta feature – Docker container virtualization

- Lightweight software containers
  - Run as an isolated process in userspace on the host operating system
  - All containers share the same system kernel
  - Based on LXC (Linux Containers), cgroups, kernel namespaces and a union-capable file system.
- Application is deployed inside a container
  - All dependencies are installed only once
  - Can be run on any operating system (Linux, Windows, MacOs)
  - Can be run anywhere (laptop, cluster, cloud)
- Installed on the Maxwell cluster
  - For each job a virtual HPC cluster of Docker containers is created
  - Looking for beta-users!