Los Alamos

NATIONAL LABORATORY

EST. 1943

slurm

workload manager

Los Alamos

NATIONAL LABORATORY

EST. 1943

Delivering science and technology
to protect our nation
and promote world stability

NNSA
National Nuclear Security Administration

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

Workload Management & Slurm

Presented by CSCNSI

What is Workload Management?

- We need a way to easily put our many compute nodes to use
- We need a tool that can match work that needs to be done with resources that are available
- It needs to be smart enough to
  - Start multi-node jobs
  - Schedule jobs from a long list of waiting tasks according to policy
  - Know what resources are available
  - Know what jobs to send to what hardware
  - Know how to start a job on our hardware
- And generally a lot of other things, like keeping track of accounting information, etc.

Scheduling vs. Resource Management

Scheduler

- The “Brain” of the system
- Single, centralized daemon
- Obtains job and node data from Resource Manager
- Prioritizes jobs
- Decides:
  - What jobs will run
  - Where they will run
  - When they will run

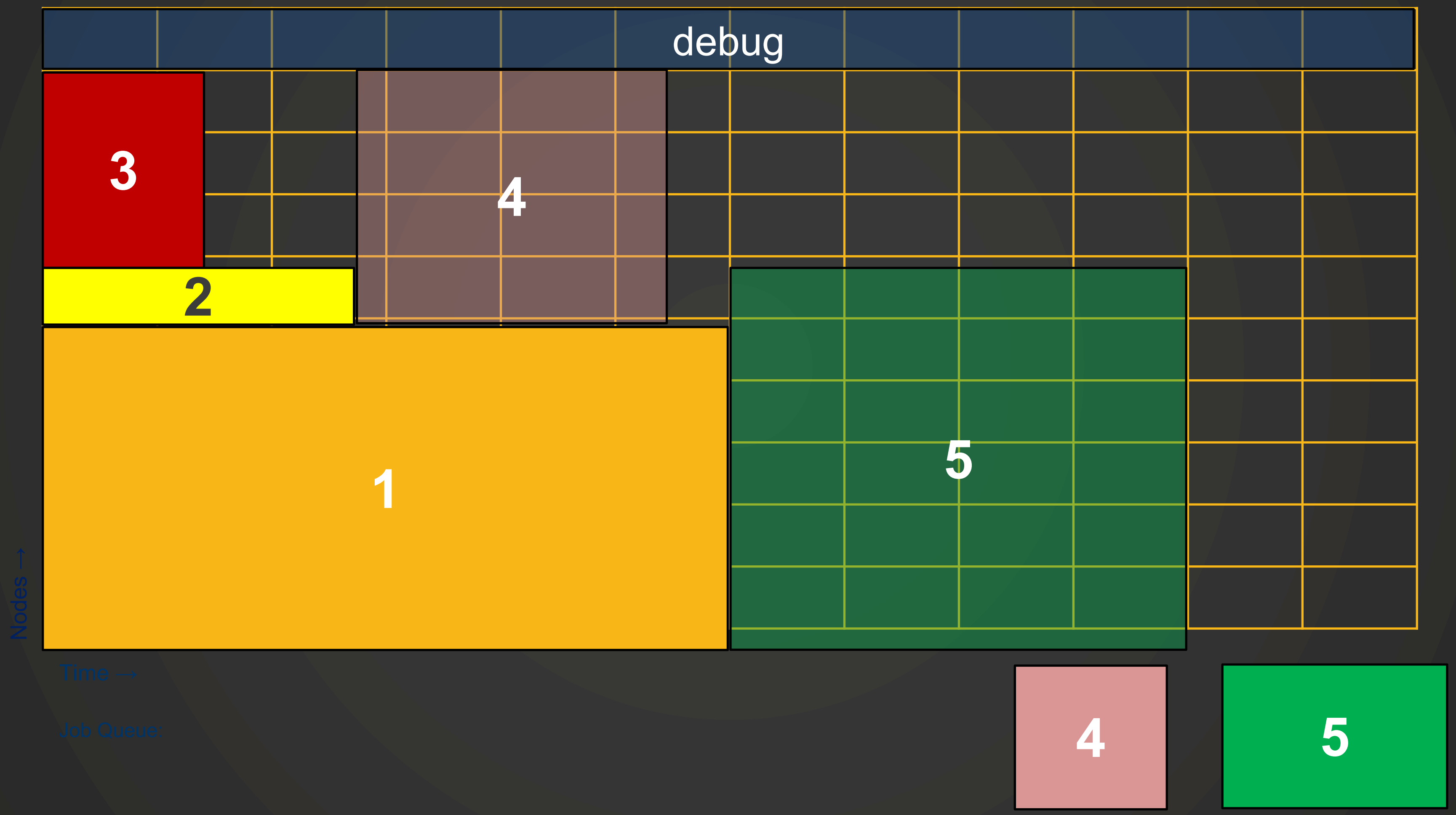
Resource Manager

- The “Worker” of the system
- Server daemon plus per-node client daemons
- Gathers node information (CPUs, memory, disk, load)
- Answers queries for this information
- Spawns jobs when told to
- Can add/remove resources

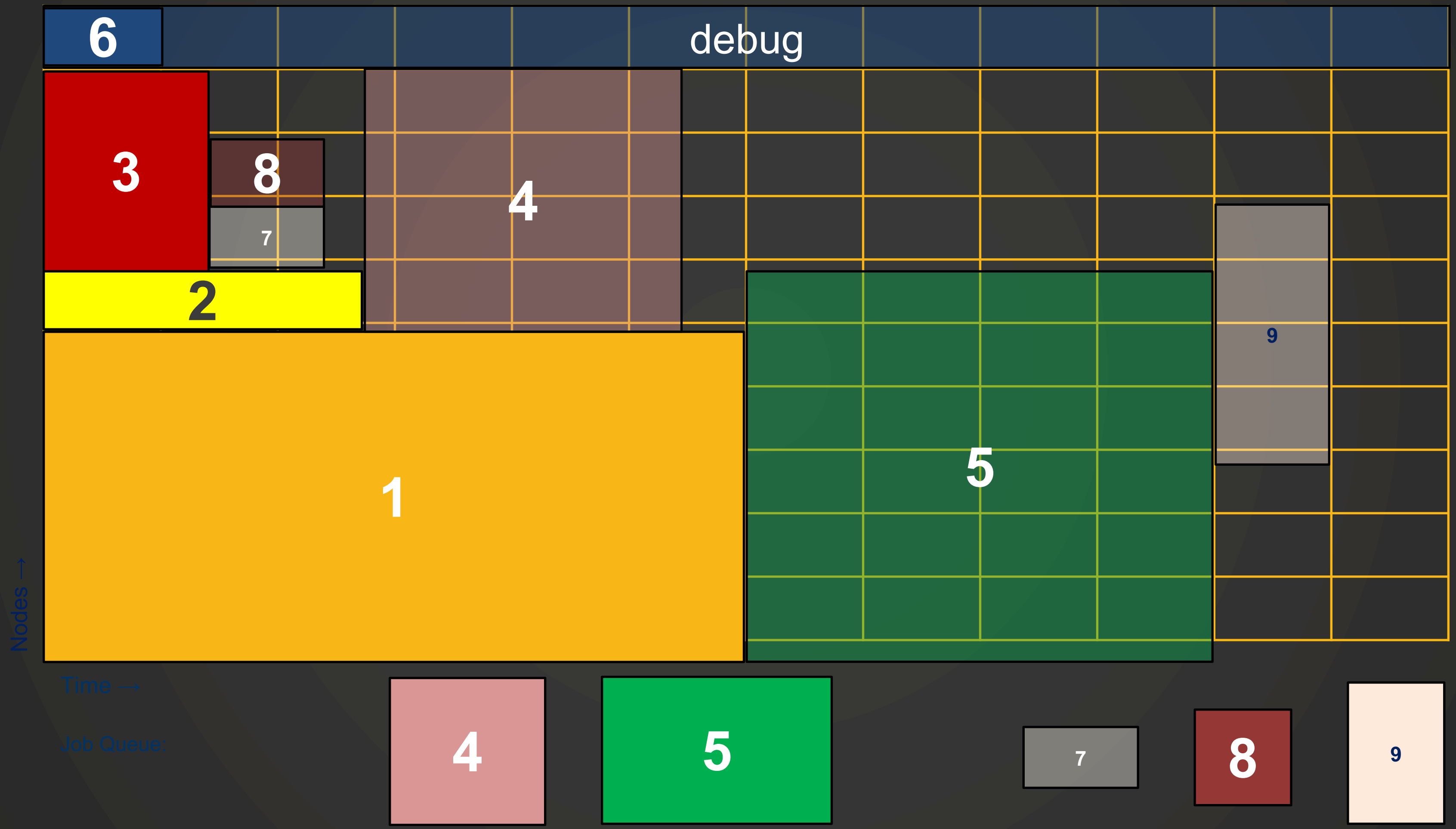
Scheduling Cycle

- Job Ordering
  - Calculate job priorities and sort job list by priority
  - Start all jobs which can start
  - Make reservations for next 1024 jobs
  - Use backfill policy to sort remaining jobs (1024)
- Job Placement (for each job being started)
  - Filter list of all nodes based on requirements
  - Filter list of nodes based on eligibility
  - Sort final node list based on node allocation policy
  - Start job on top n nodes

Scheduling Example – Iteration #1



Scheduling Example – Iteration #2 (backfill)

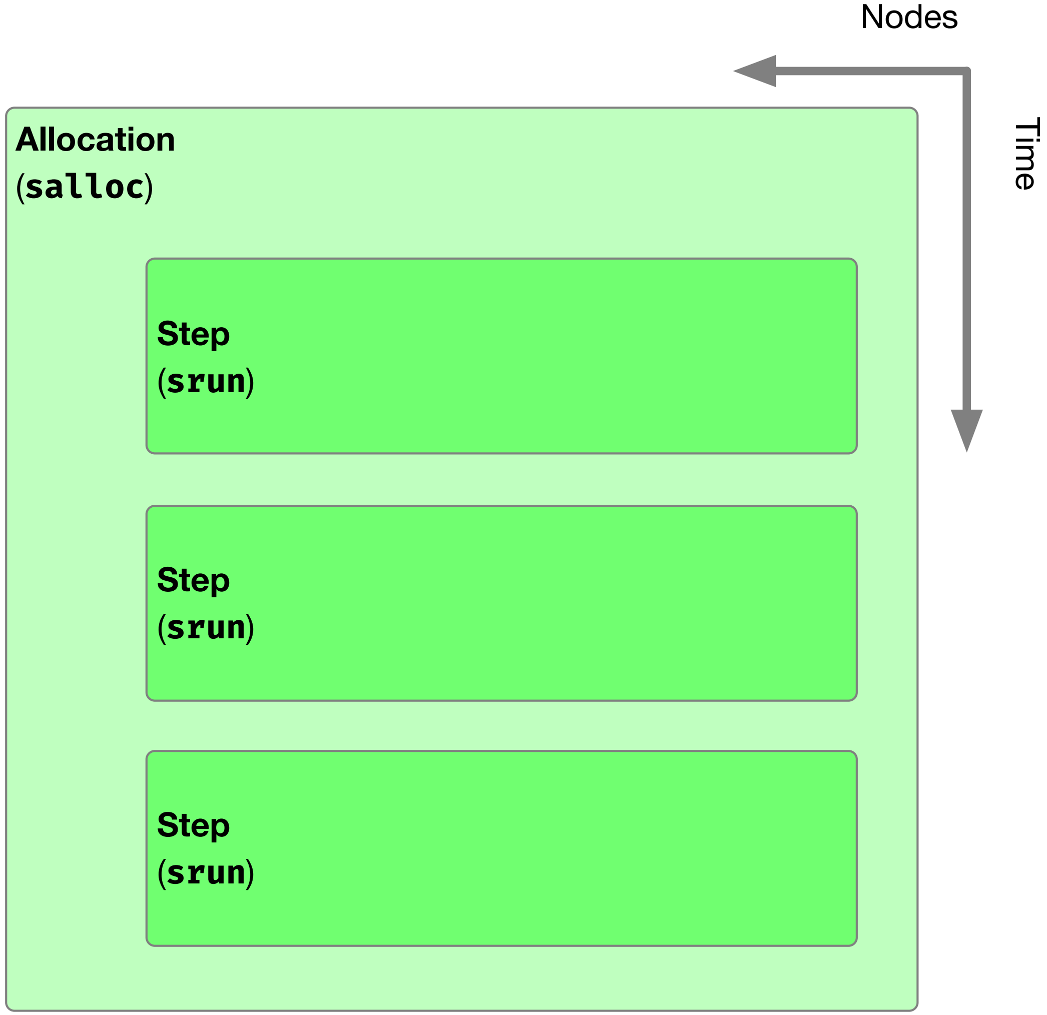


What is Slurm?

- Slurm is a Workload Manager for HPC workloads
- Originally "*Simple Linux Utility for Resource Management*"
  - Was spelled SLURM
- Since it now is also a scheduler, was rebranded *Slurm* (not an acronym)
- Originally written at LLNL
- Now commercially developed & supported by SchedMD
- Used on more than half of the top 10 of the top500

Structure of a Slurm job

- A job needs an “allocation”
  - Specifies needed resources
    - Memory
    - Nodes/CPUs
    - …
  - Specifies a max time
- Within an allocation multiple steps can execute
  - Steps, in total, must fit within allocation



Slurm interactive job

- An allocation is made with the **`salloc`** command
- An allocation is bound to a process
- **`srun`** started within that process create a step
- The job ends when the allocation process ends

```
[fe] $ salloc –n1 /bin/bash
salloc: Granted job allocation 423
[fe] $ srun --pty /bin/bash
[n01] $ ./do_work
[n01] $ exit
[fe] $ srun –pty ./do_more_work
(runs on n01)
[fe] $ exit
salloc: Relinquishing job allocation
423
[fe] $
```

Slurm batch job

- Jobs can be specified as a pre-defined script
- Special *#SBATCH* lines provide Slurm parameters
- Can submit multiples of the same task
  - Or can submit arrays of the same job, keyed by the *$SLURM_ARRAY_TASK_ID*
- Submitted with:
  - *sbatch <script>*

```
#!/bin/bash
#
#SBATCH --job-name=test_emb_arr
#SBATCH --output=res_emb_arr.txt
#SBATCH --ntasks=1
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=100 #
#SBATCH --array=1-8
srun ./my_program.exe \
   $SLURM_ARRAY_TASK_ID
```

Common Slurm commands

Start an allocation	**salloc**
Run a step in an allocation	**srun**
Submit a batch script	**sbatch**
View the queue	**squeue**
Stop a running job	**scancel**
Attach to I/O of current task	**sattach**
View the cluster status	**sinfo**
View job accounting data	**sacct**
Control Slurm	**scontrol**

All of the commands have extensive options. See `man <slurm_command>`

Questions?