Introduction to Cluster Computing

Dr. Hrachya Astsatryan,
Institute for Informatics and Automation Problems,
National Academy of Sciences of Armenia,
E-mail: hrach@sci.am

1

QUEUE SYSTEMS

HPC Workloads

- HPC workloads allow run computationally intensive tasks as processes across multiple nodes.
- These different processes work together in parallel as a single application.
- Communication between basks are carried out by messaging passing mechanisms.

Scheduler

- Queues up workloads against the resources of the cluster to orchestrate its use.
- Receive any requests for workloads that need to be scheduled from users of the cluster, keep track of them and then run those workloads as needed when resources are available.
- Aware of any resource availability and utilisation and do their best to consider any locality that might affect performance
- The main purpose is to schedule compute jobs based on optimal workload distribution.

Scheduler functions

- Allow users to submit new jobs
- Allow users to monitor the state of their queued and running jobs
- Allow users and administrators to control running jobs
- Monitor the status of managed resources including system load, memory available, etc.

Understanding Queues

- Queues are collections of jobs submitted to the scheduler for execution.
- Each queue can have specific constraints, including job size limits, time limits, and user permissions.
- Certain nodes within the cluster may be configured to accept jobs only from specific queues, such as those dedicated to specific departments or projects.

Interactive vs Batch Queues

Interactive Queues:

- Designed for short-duration, where users require immediate access to computing resources.
- Typically have shorter time limits and may prioritize responsiveness over resource efficiency.

Batch Queues:

- Intended for longer-duration that do not require immediate user interaction.

Interactive vs Batch Jobs

Interactive jobs

- Requested and immediately run, providing users with a bash shell or preferred shell.
- Typically, only a few nodes in an HPC cluster are dedicated solely to interactive jobs.
- Resources must be available instantaneously for the request to succeed, making fulfillment challenging for requests requiring multiple cores.

Batch jobs

- Users prepare a batch submission script, which requests resources and contains execution commands for a specific program.
- Scheduler adds the job to the chosen queue and runs it when resources become available

Interactive vs Batch Jobs

- If you wish to use a cluster for interactive work and/or running applications like Python with Jupyter notebooks or RStudio using GUIs, you will need to request an interactive job from the scheduler.

- If you wish to use a cluster for large-scale data analysis using packages like Apache Spark or Dask, you will need to request a batch job from the scheduler and prepare an appropriate batch script. Similarly, for software packages like OpenFOAM, GROMACS, NAMD, or LAMMPS, batch jobs are typically required.

How Check Queue Resources

- sinfo - shows a summary of the available partitions along with details such as their names, node count, state (idle, down, draining, etc.), and available resources like CPUs, memory, and GPUs.
- sinfo –l - provides a more extensive listing, including features, node states, node names, and their respective states, as well as detailed information about node properties and available resources.
- sinfo –N - display only the node details without the partition summary.

Schedulers

SLURM

- sudo apt update
- sudo apt install slurmd slurmctld
- sudo chmod 777 /etc/slurm-llnl
- sudo cat << EOF > /etc/slurm-llnl/slurm.conf
- sudo chmod 755 /etc/slurm-llnl/
- sudo systemctl start slurmctld
- sudo systemctl start slurmd
- sudo scontrol update nodename=localhost state=idle
- sinfo

OpenPBS

Altair Grid Engine

MOAB HPC suite

Menti 3: 3287 9162

2

SLURM (Simple Linux Utility for Resource Management)

Overview

Free and open-source job scheduler for Linux and Unix-like kernels, used by many of the world's supercomputers and computer clusters (over 65% of TOP500 systems).

Massive Scalability & Throughput

First Class Resource Management for GPUs

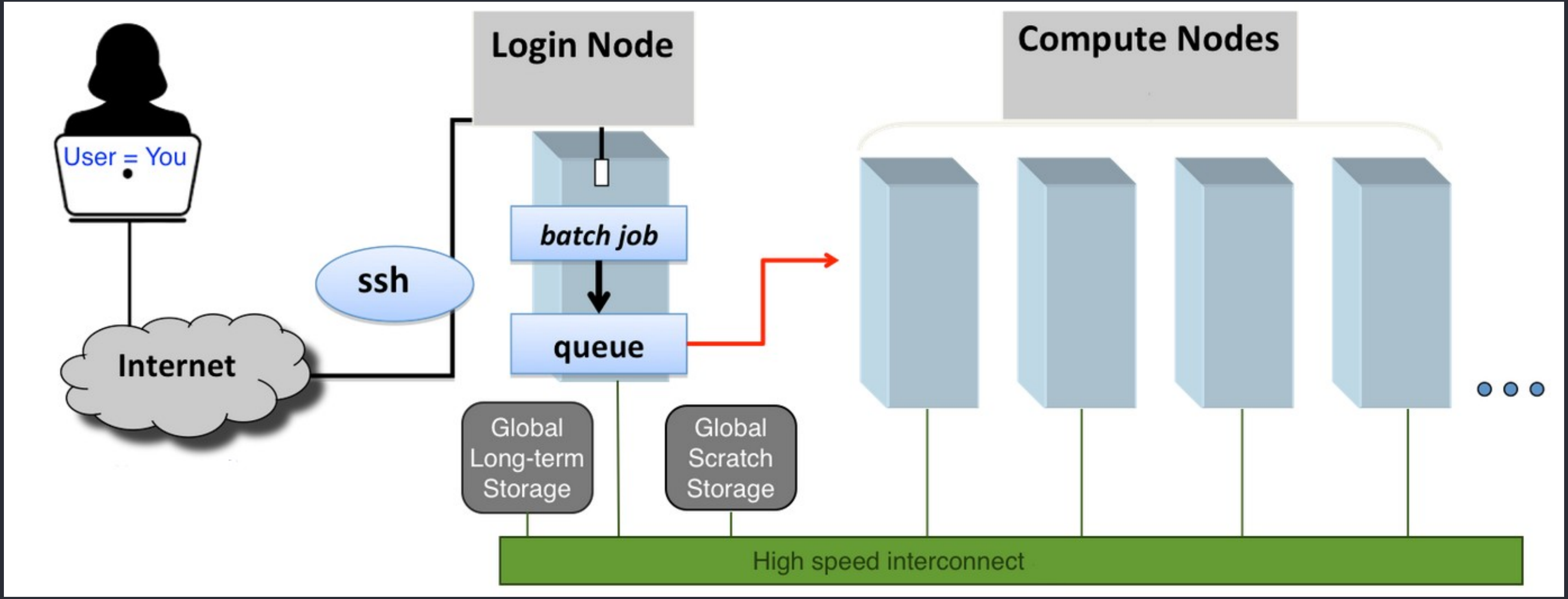
Resource Allocation

Plugin Based Architecture

On-Prem & Cloud

Open Source

Diagram


User = You
Internet
ssh
Login Node
batch job
queue
Compute Nodes
Global Long-term Storage
Global Scratch Storage
High speed interconnect

Hardware/Software Ecosystem

Application	Physics, Weather and Climate Analysis, Pharmaceutical Design, New Materials, etc.				
Application Environment	PVM	MPI	OpenMP	Math Library	
Cluster Software	Cluster Management	Cluster Monitoring	Cluster Deployment	Reporting	Job Scheduling
System Software	OS	Compiler	Library		
Hardware	Server	Storage	Network	Infrastucture	

SLURM

- Cluster resource availability monitoring and reporting
- Job queue management
- Job scheduling optimization
- Automated job resubmission
- Job status notifications
- Resource utilization statistics:

Jobs Submissions

srun is a command used to submit and manage jobs in a Slurm environment.

srun [options] command [command options]

Options

- Resource allocations: specify the number of nodes, tasks, CPUs, memory, and other resources required for the job.
- Output options: redirecting standard output and error streams to files (-o, -e) or the terminal (-t).
- Time limit: Set a maximum time limit for job execution using options like --time or -t.
- Job name : Assign a name to the job using --job-name or -J.
- ...

Interactive Job Submissions

Interactive sessions are useful for tasks that require real-time feedback or debugging, such as software development, testing, debugging, or exploring data interactively. They are also suitable for running interactive applications or tools that require user input or graphical interfaces.

Key points about interactive job submissions:

- Immediate shell access - instant access to a shell prompt on a compute node, allowing them to run commands, execute programs, and perform various tasks directly.
- On-demand resource allocation - request an interactive session from the scheduler, specifying resource requirements such as the number of nodes, CPUs, memory, and duration of the session.
- Dynamic environment - full control over the environment within the interactive session, allowing them to install software, set up configurations, and customize the environment according to their needs.
- Session termination - Interactive sessions automatically terminate after the specified duration or when the user exits the session explicitly.

Interactive Job Submissions

- srun /bin/hostname -i

You can configure the resources available to the interactive session by adding command line options. For example to start an interactive session with access to 1 GB of RAM:

- srun --mem=1G /bin/hostname -i

SLURM ADDITIONAL OPTIONS

Input, Output, Working Directory, Environment

- --output=filename_pattern - Standard output (stdout) of the job script will be connected to the file specified by filename_pattern.
- --error=filename_pattern - Standard error (stderr)
- --input=filename pattern - Standard input

Notifications

- --mail-user=email_address
- --mail-type=[TYPE|ALL|NONE]

SLURM ADDITIONAL OPTIONS

Time Limits

- --time=time -- [[days-]hours:]minutes[:seconds].

Memory Allocation

- --mem=size[K|M|G|T]
- --mem-per-cpu=size[K|M|G|T]
- --mem-per-gpu=size[K|M|G|T]

GPUs

- --gpus=[type:]number - v100:1

BATCH JOB SUBMISSION

1. Prepare a Batch Script - containing the necessary commands to execute your job.

2. Submit the Job - Use the job submission command provided by the scheduler to submit your batch script.

3. Monitor the Job - After submitting the job, you can monitor its progress.

4. Retrieve Results - Once the job is completed, the output files generated will be available for analysis.

SLURM DIRECTIVES

```
#SBATCH <option_1>=<value>
#SBATCH <option_2>=<value>

...
#SBATCH <option_N>=<value>
```

SLURM DIRECTIVES

- --ntasks - number of tasks will be launched from the job
- --ntasks-per-node - number of tasks be invoked on each node
- --nodes - the number of nodes are allocated to the job
- --mem-per-cpu - memory in MB required per node or per vCPU
- --job-name
- --gpus-per-node
- --time

BATCH JOB SUBMISSION

1. Prepare a Batch Script - containing the necessary commands to execute your job.

```
#!/bin/bash
#SBATCH --job-name=singlecpu
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --output=output-%x.%j
#SBATCH --error=error-%x.%j

sleep 40s
/bin/hostname
date
pwd
```

BATCH JOB SUBMISSION

2. Submit the Job - Use the job submission command provided by the scheduler to submit your batch script.

```
sbatch file
```

BATCH JOB SUBMISSION

3. Monitor the Job - After submitting the job, you can monitor its progress.

Status	Code	Description
COMPLETED	CD	completed successfully
COMPLETING	CG	is finishing but some processes are still active
CANCELLED	CA	explicitly cancelled by the user or system administrator
FAILED	F	terminated with a non-zero exit code and failed to execute
PENDING	PD	is waiting for resource allocation
RUNNING	R	currently is allocated to a node and is running

BATCH JOB SUBMISSION

3. Monitor the Job - After submitting the job, you can monitor its progress.

- Basic Usage - Use squeue to display information about jobs in the queue.
- Limit to User's Jobs - Use squeue --user=$USER to display jobs of a specific user.
- Limit to Your Own Jobs - Use squeue --me to display only your own jobs.
- Detailed Information - Use squeue --me --long for detailed information about your jobs.
- squeue --job ID

BATCH JOB SUBMISSION

3. Monitor the Job - After submitting the job, you can monitor its progress.

- Basic Usage - Use squeue to display information about jobs in the queue.
- Limit to User's Jobs - Use squeue --user=$USER to display jobs of a specific user.
- Limit to Your Own Jobs - Use squeue --me to display only your own jobs.
- Detailed Information - Use squeue --me --long for detailed information about your jobs.
- squeue --job ID

BATCH JOB SUBMISSION

3. Monitor the Job - After submitting the job, you can monitor its progress.

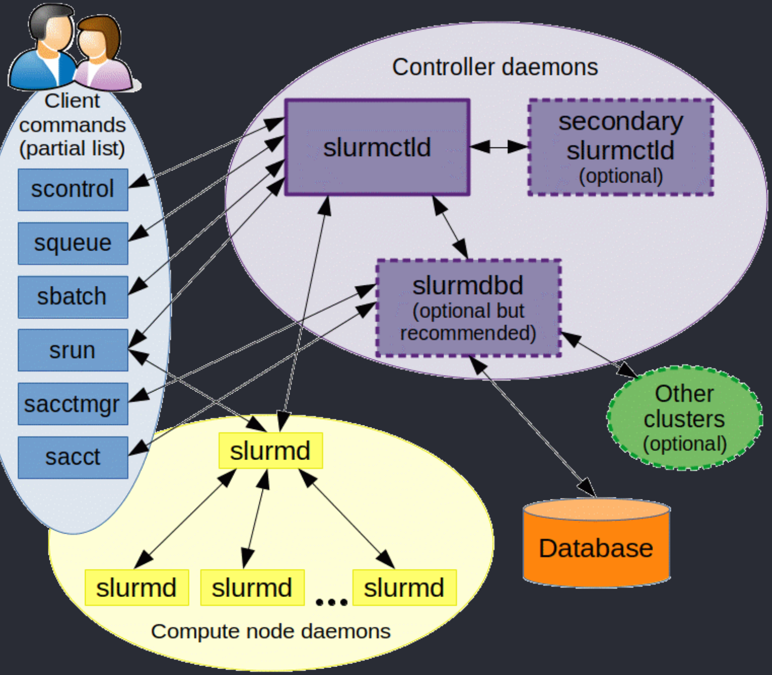
sacct is a command-line tool used to view accounting information for SLURM jobs. It provides detailed information about completed and running jobs, including job IDs, user IDs, job state, start and end times, resource usage, and exit statuses.

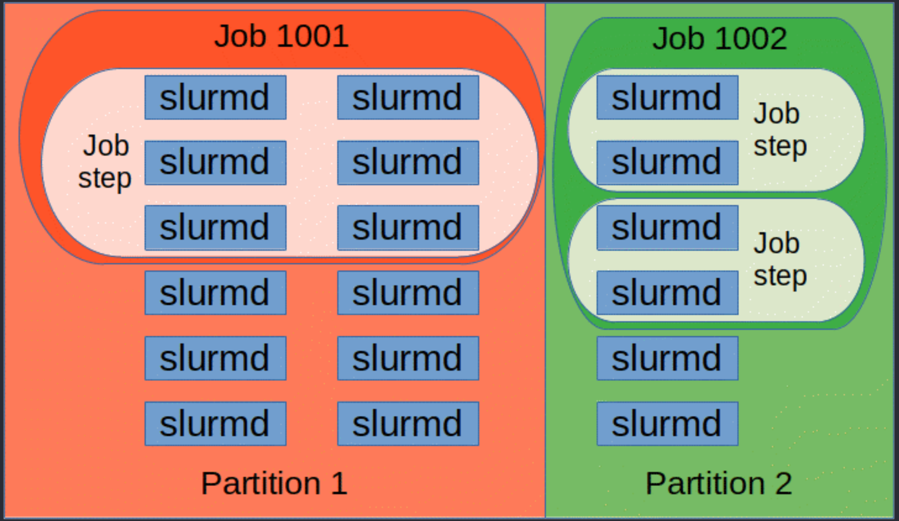
sacct -n - displays job accounting information without headers.

scontrol is a command-line tool used to control various aspects of SLURM's behavior and configuration. It allows administrators and users to view and modify SLURM configuration parameters, job properties, node properties, etc.

scontrol show job <JOB_ID> - displays detailed information about a specific job.

Architecture


Client commands (partial list)
scontrol
squeue
sbatch
srun
sacctmgr
sacct
Controller daemons
slurmctld
secondary slurmctld (optional)
slurmdbd (optional but recommended)
Other clusters (optional)
Database
slurmd
slurmd
slurmd
...
slurmd
Compute node daemons



Monitoring Slurm system: nodes and partitions

```
sbatch -p debug  --wrap="sleep 2m"
scontrol update NodeName=node01 State=down Reason=hung_proc
scontrol update NodeName=node01 State=resume
```

ADVANCED JOB SUBMISSION

An “array job” (multi-task) is a set of tasks run from a single batch job submission script. Examples: Monte Carlo simulations, Parameter sensitivity analysis, Batch file processing

Advantages

- Single job submission for similar tasks
- Efficient utilization of computational resources
- Suitable for Embarrassingly Parallel problems

Disadvantages

- Difficulty in identifying and resubmitting failed tasks
- Overhead if tasks are too small, impacting scheduler efficiency

SLURM EXAMPLE – MULTI TASK

```
#!/bin/bash

#SBATCH --job-name=singlejobs
#SBATCH --ntasks=3
#SBATCH --cpus-per-task=1
#SBATCH --output=output-%x.%j
#SBATCH --error=error

srun --ntasks=1 /bin/hostname
srun --ntasks=1 /bin/hostname
srun --ntasks=1 /bin/hostname
```

SLURM – SUBMIT & MONITOR

- sbatch myslurmscript.sh - submit a job
- squeue --job ID - see the job status in the queue
- scancel ID - cancel a job
- sacct - info about job status
- seff <jobid> - info about completed jobs