SAND2021-7894C

Sandia National Laboratories

Moving to a Multi-cluster HPC Slurm Environment

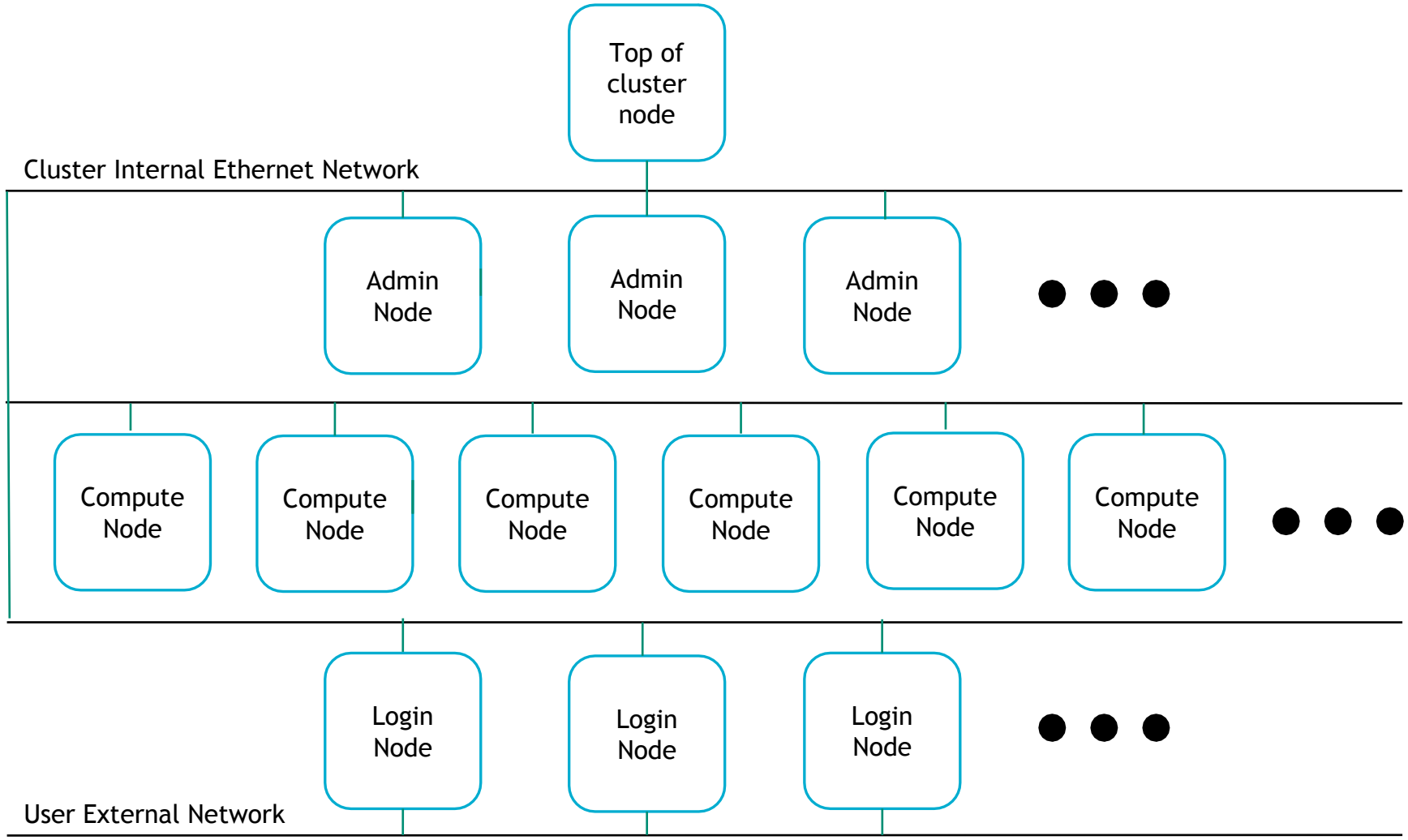
Presented by Mark Schmitz

July 16, 2021

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Basic Cluster Layout

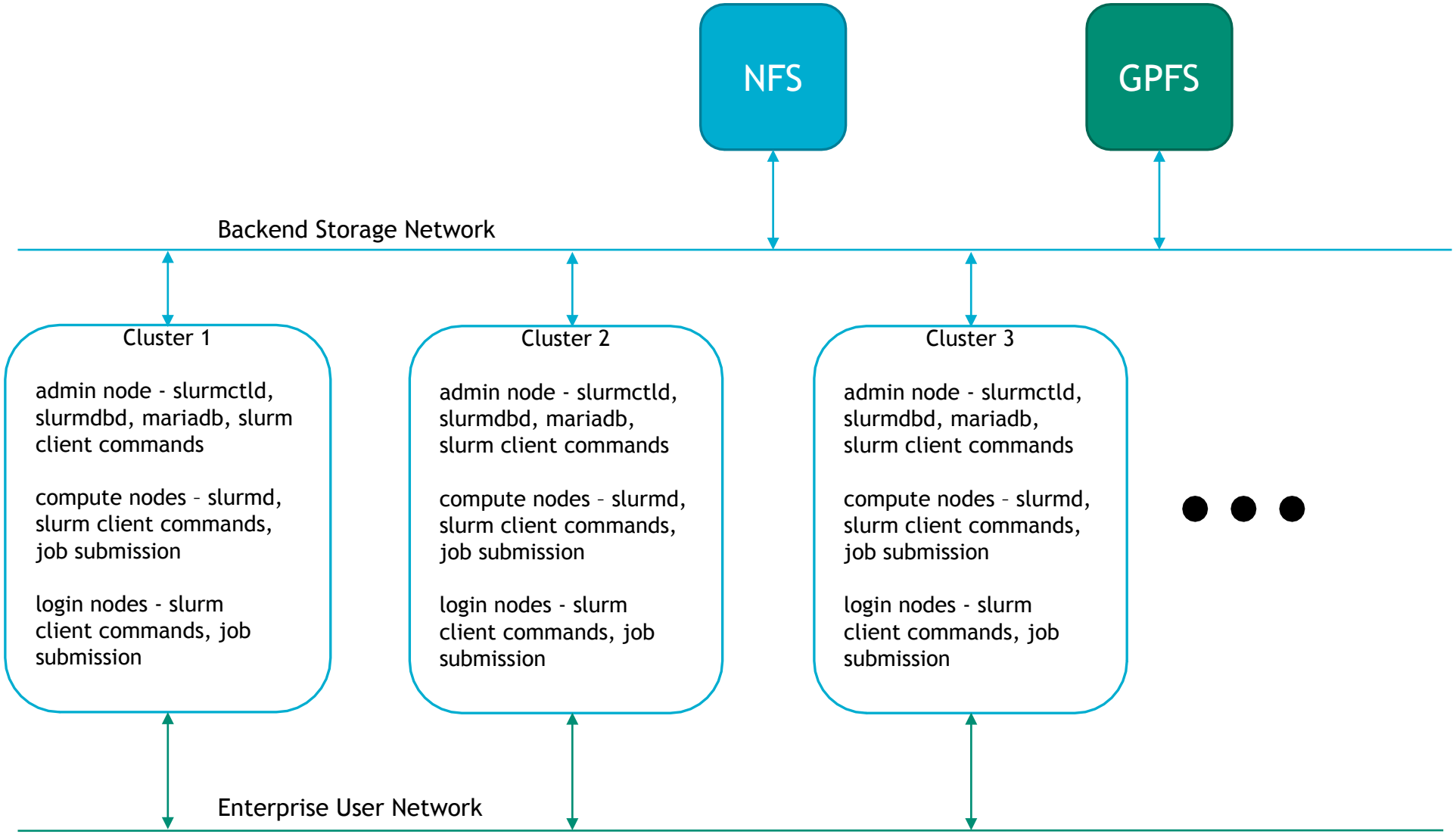


Note: The Infiniband/OPA high speed fabric and lustre filesystems have been left off of the diagrams in this presentation to simplify the complexity. While the high speed fabric is certainly an important part of HPC cluster performance and job execution it is not directly involved in job scheduling.

Some Slurm basics

- Slurm job scheduler – slurmctld (running on an admin node)
- Slurm database controller – slurmdbd (running on an admin node)
- Slurm job management – slurmd (runs on all compute nodes)
- Slurm database controller (slurmdbd) uses an underlying database like MySQL/Mariadb (also running on an admin node)
- Slurm provides failover mechanisms for slurmctld, slurmdbd, and the database controller (Mariadb) so no additional High Availability (HA) mechanism is needed.

Legacy Cluster Layout



Legacy HPC Scheduling Environment

- Each HPC cluster did its own Slurm scheduling
- Each HPC cluster has internal non-routable IP network spaces
- Globally shared filesystems:
  - NFS for /home and /projects via backend ethernet network
  - GPFS via backend ethernet network
  - Lustre filesystems via IB/OPA fabric gateways to backend Lustre storage clusters
- User accounts globally shared, with only a couple of restricted clusters
- User association with accounts provided by a separate utility (WC-Tool)
- Group/Project accounts or Workload Characterization ID (WCID) are managed in a fair share tree
- User and account updates to the Slurm database occur on every cluster separately

Project Goals

- With an increasing number of available clusters there is a growing need to:
  - Allow users to view and interact with all HPC clusters from a single login to any of the clusters
  - Allow jobs to be submitted to multiple clusters without needing to log on to each cluster and submit jobs locally
- Provide a more integrated user environment with better workflow
- Better redundancy and maintainability
- Prepare for the next generation of HPC workload scheduling systems like Flux
  - with exascale clusters and
  - workload scheduling at the data center level

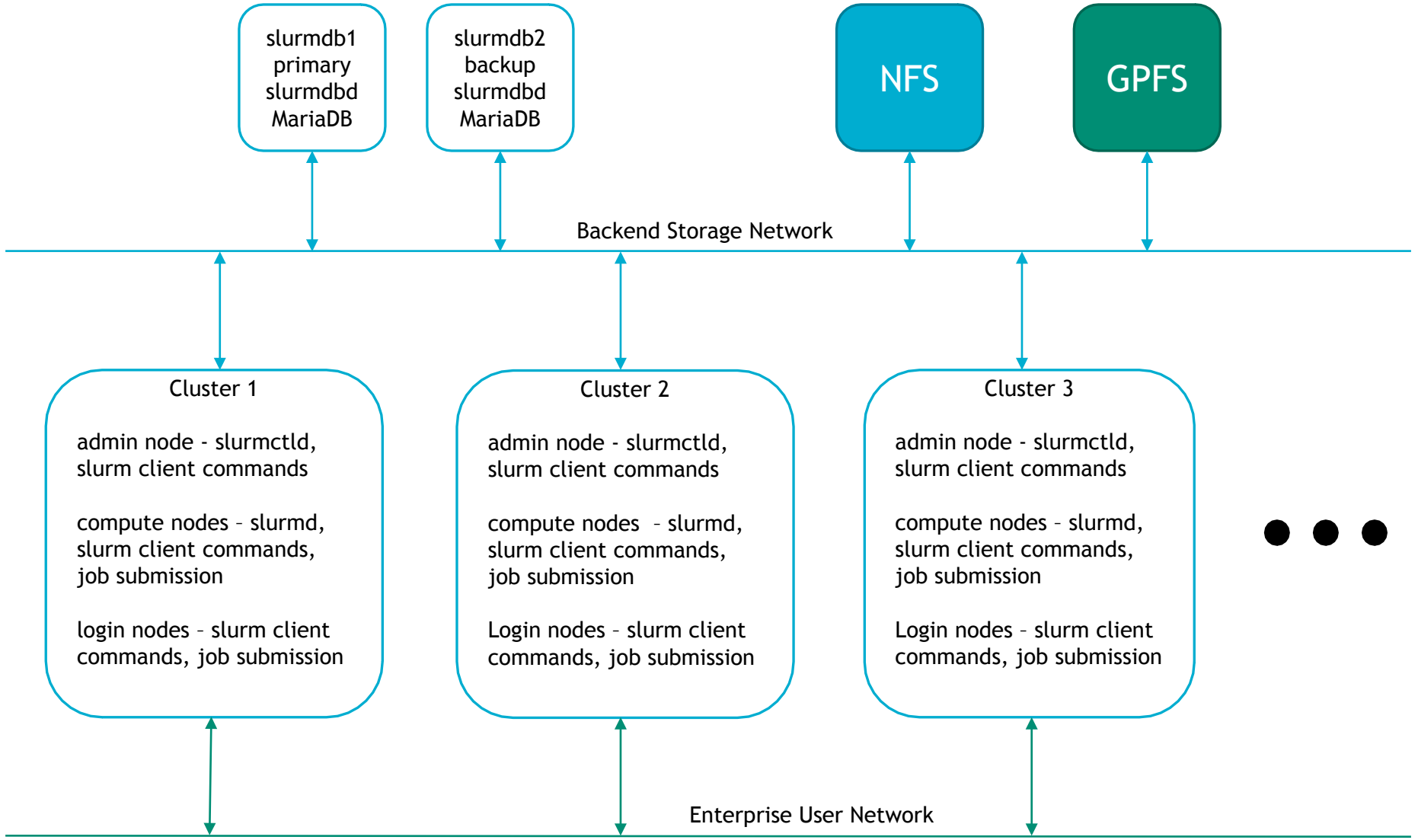
Transition to a multi-cluster scheduling environment

- Moving to a multi-cluster scheduling environment was a major architectural change
- Cluster internal networks need to be reachable across all clusters
  - Routing between all cluster node types through the existing backend storage network
  - Cluster internal non-routable network IP address range overlap need to be resolved
- New common slurm database servers were provisioned with a primary and backup server
  - Faster processors, more memory and NVMe SSD storage to increase database performance
  - Used latest version of MariaDB not available in TOSS distribution for better performance and features
  - MariaDB replication used to keep the primary and backup databases in sync
  - Placed on the high speed backend storage ethernet network backbone so all clusters have access

Transition to a multi-cluster scheduling environment (continued)

- Slurm configuration defines and handles failover of common slurm database server pair
- Contend with cluster specific Slurm QoS definitions which are now global to all clusters
  - Updated job_submit and cli-filter scripts for conversion from global QoS names to cluster specific names
- Updated cli-filter to prevent jobs from being remotely submitted to restricted clusters
- Users will possibly need to modify submit scripts to handle different cluster architectures
  - In a heterogeneous cluster environment submit scripts need to focus on cores and be more intelligent about selecting executables and the environment to match the cluster the jobs lands on

Multi-cluster Scheduling Layout



Benefits of the multi-cluster HPC Slurm environment

- Users can log on to any HPC cluster login node and use slurm client commands to view and interact with any of the HPC clusters:
  - Slurm commands like squeue, sacct, sshare include a `--cluster` or `-M` switch to specify which cluster(s) they would like to view.
- Job submission commands like sbatch, salloc and srun also include the `--cluster or -M` switch to enable job submission to the local or remote cluster, or multiple clusters
- When multiple clusters are specified in a job submission slurm will determine where the job can be started first and submit the job to that cluster

Benefits of the multi-cluster HPC Slurm environment (continued)

- The slurm database function is running on optimized hardware to provide better performance and redundancy
- User and account management consolidated on the centralized slurm database servers
- Better prepared for future transition to next generation HPC workload scheduling systems like Flux
- Future reduction in the number of login nodes needed on any given cluster or consolidation of login nodes into a dedicated login cluster with an HPC cluster compute farm

Support Vendor Information and Questions

- Slurm support provided by SchedMD. Website at https://slurm.schedmd.com
- Slurm documentation is available on that website
- For more information on Slurm network configuration see: https://slurm.schedmd.com/network.html

- Questions