# Slurm Native Workload Management on Cray Systems

Morris Jette
jette@schedmd.com

SchedMD LLC
http://www.schedmd.com

# Outline

- Slurm operation with Cray ALPS resource manager
- Native Slurm design
- New Slurm capabilities (for all most system types)

# Outline

- Slurm operation with Cray ALPS resource manager
- Native Slurm design
- New Slurm capabilities (for all most system types)

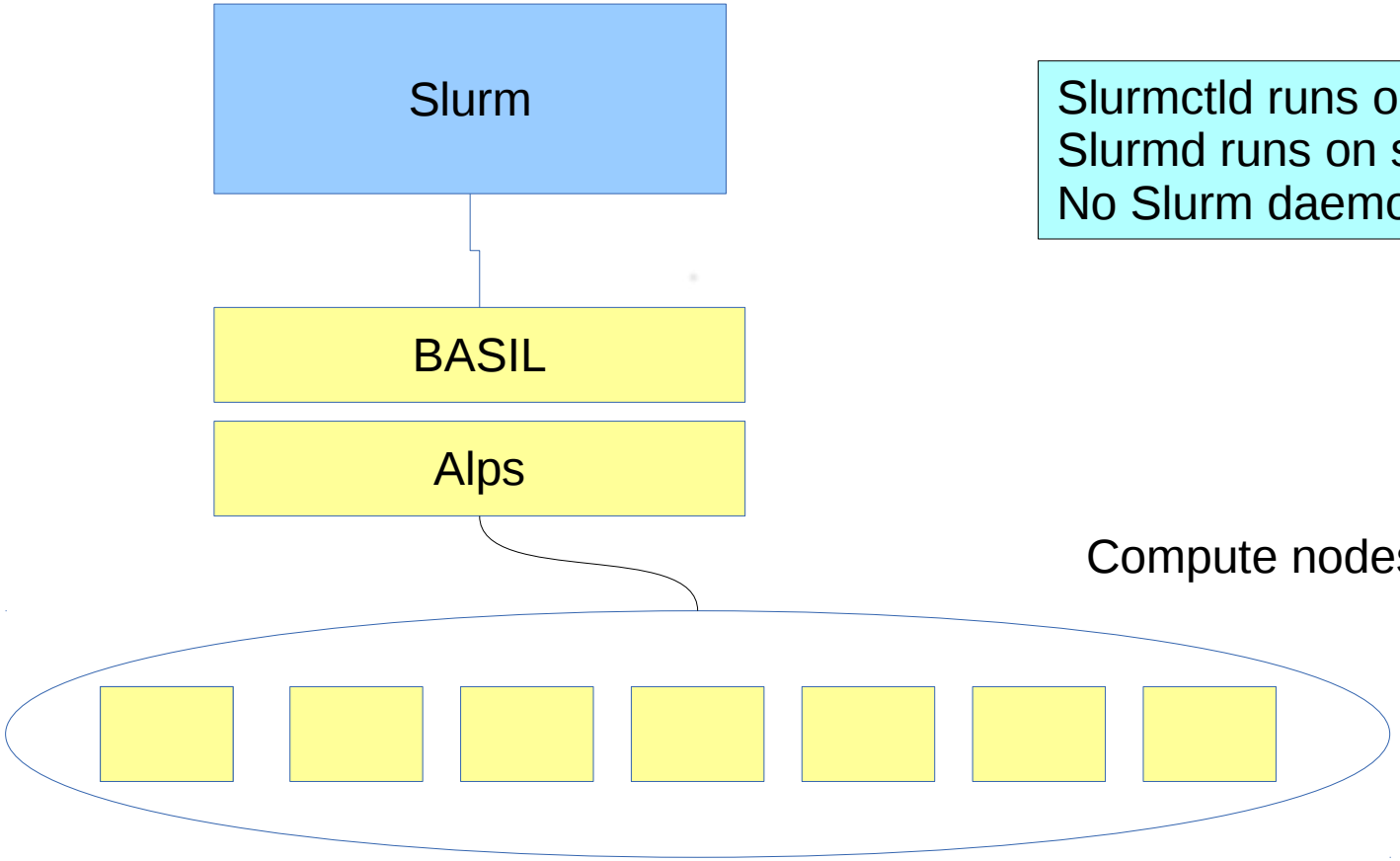
# ALPS and BASIL

- **ALPS** – **A**pplication **L**evel **P**lacement **S**cheduler
  - Cray's resource manager
  - Six daemons plus variety of tools
    - One daemon runs on each compute node to launch user tasks
    - Other daemons run on service nodes
  - Rudimentary scheduling software
    - Dependent upon external scheduler (e.g. Slurm) for workload management
- **BASIL** – **B**atch **A**pplication **S**cheduler **I**nterface **L**ayer
  - XML interface to ALPS

# Slurm and ALPS Functionality

- Slurm
  - Manages resources and jobs
  - Prioritize queues and enforces limits
  - Scheduling and accounting of jobs
- ALPS
  - Allocates and releases reservations for jobs (as directed by Slurm)
  - Launches tasks
  - Monitors node health

# Slurm Architecture for Cray




Slurmctld runs on sdb
Slurmd runs on service node(s)
No Slurm daemons on compute nodes

# Job Launch Process

- User submits a job script
- Slurmctld creates an ALPS reservation
- Slurmctld sends the job script to slurmd
- Slurmd claims the reservation for the session ID
- Slurmd launches the user script
- Aprun (ALPS tool) launches the tasks on compute nodes (invoked directly by the user or run by srun)
- When the job finishes the reservation is released

# Outline

- Slurm operation with Cray ALPS resource manager
- Native Slurm design
- New Slurm capabilities (for all most system types)

# Motivation for Native Slurm

- Current architecture has limitations due to the translation from Slurm to ALPS
- Not all features of Slurm are supported by ALPS
  - Spawning multiple concurrent jobs per login session
  - Running multiple applications (job steps) per job allocation
  - Running multiple jobs per node
  - Job profiling
- Improved performance
- Allow native Slurm functionality scheduling, resource management and reporting
- Majority of MPI implementations already interface to Slurm as launcher with the srun command

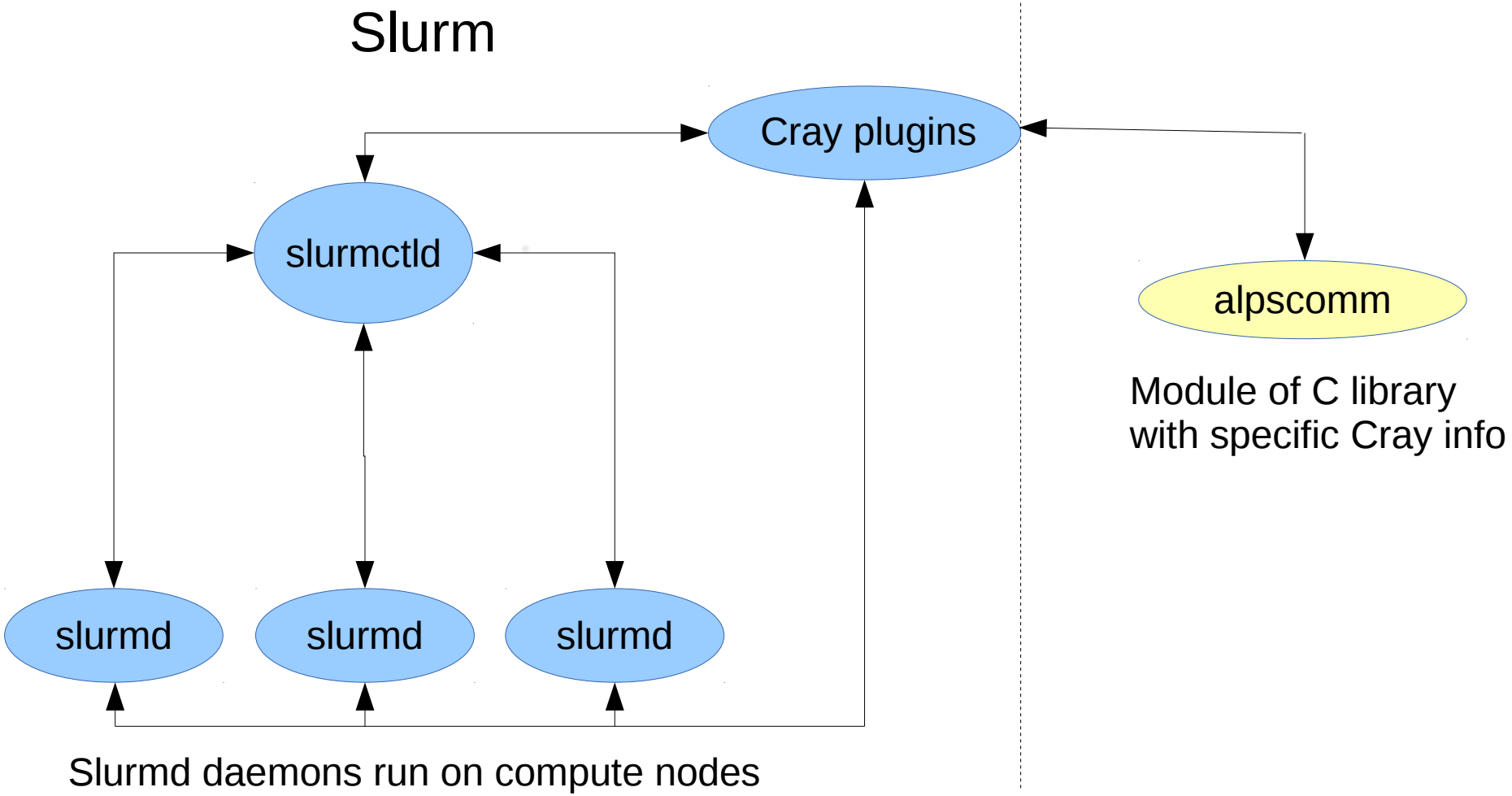
# ALPS Refactored

- ALPS divided into
  - Library with underlying functions
    - Network management
    - Node health check
  - Daemons, commands, etc.
    - Preserve previous functionality

# Slurm Native Implementation

- Cray and SchedMD developed plugins to provide the following services:
  - Dynamic node state change information
  - System topology information
  - MPMD (Multiple-Program Multiple-Data) support
  - Node Health Check Support (can be disabled in Slurm)
  - Network performance counter management
  - Congestion management information for Cray Hardware Supervisory System

# Slurm Native Architecture



# Slurm Cray Specific Feature

- Network Performance Counters (NPC)
  - To access the Cray's NPC, use –network option in sbatch/salloc/srun commands
  - --network=system for the system wide NPC
  - --network=blade for the blade NPC
- Core Specialization
  - To specify count, use -S/--core-spec=# option in sbatch/salloc/srun
  - Ability to reserve number of cores allocated to the job and not used by the application
  - All non-application processes are migrated to the specialized cores to reduce application jitter (system noise)

# Slurm Configuration for Cray

- Configure plugins to use Cray without ALPS
- CoreSpec (bind system programs to specialized cores)
  - Set CoreSpecPlugin=core_spec/cray
- Job Submit (sets “–gres=craynetwork” to limit number of simultaneous applications)
  - Set JobSubmitPlugin=job_submit/cray
  - Also set craynetwork GRES count on each node to 4 (or 2 if node includes Xeon Phi)
- Process tracking (uses Cray job container to purge files)
  - Set ProctrackType=proctrack/cray
- Select (manages network counters, plus wrapper for select/cons_res/)
  - Set SelectType=select/cray
- Switch
  - Set SwitchType=switch/cray
- Task (configures some environment variables)
  - Set TaskPlugin=cray (other task plugins could also be used)

# Outline

- Slurm operation with Cray ALPS resource manager
- Native Slurm design
- New Slurm capabilities (for all most system types)

# New Slurm Functionality

- Core specialization (extended to generic Linux clusters by Bull using cgroup)
- CPU governor under user control
- Gang scheduling support for user controlled CPU governor and frequency
- New function calls added to several plugins (greater flexibility)

# Slurm Limitation on Cray

- Number of running applications per node limited due to network constraints
  - 2 or 4 simultaneous applications (depending upon hardware)