SLURM: Simple Linux Utility for Resource Management

ClusterWorld Conference and Expo
June 2003

Morris Jette and Mark Grondona
Lawrence Livermore National Laboratory

www.llnl.gov/linux/slurm

Disclaimer

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livremore National Laboratory under Contract No. W-7405-Eng-48.

Acknowledgments

> Jointly developed by
  - Lawrence Livermore National Laboratory (LLNL) and
  - Linux NetworX

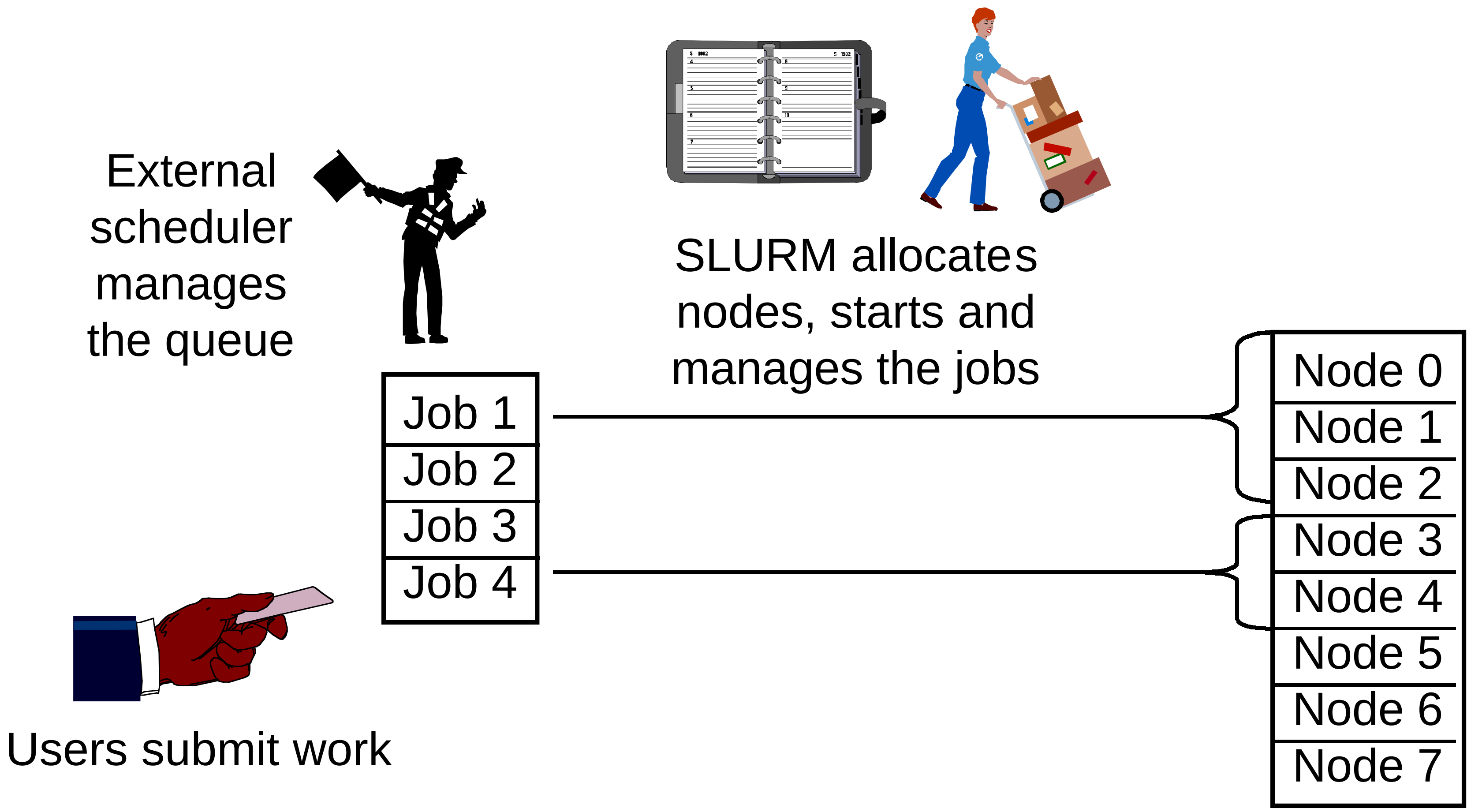
> Additional Developers:
  - Jay Windley, Linux NetworX
  - Joseph Ekstrom, LLNL
  - James Garlick , LLNL
  - Kevin Tew , LLNL

What is SLURM?

- Allocates access to computer nodes
- Distributes work to allocated resources
- Arbitrates requests by managing queue of pending work

- NOT a comprehensive cluster administration or monitoring package
- NOT a sophisticated batch system
  - An external entity can manage the SLURM queues

SLURM in a Nutshell


External scheduler manages the queue
SLURM allocates nodes, starts and manages the jobs
Job 1
Job 2
Job 3
Job 4
Node 0
Node 1
Node 2
Node 3
Node 4
Node 5
Node 6
Node 7
Users submit work

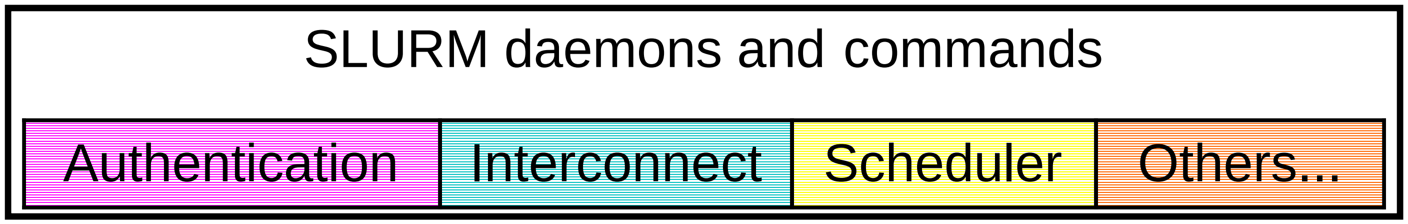
SLURM Design Criteria

- Simple
- Open source: GPL
- Portable
  - C-language, *autoconf*, general-purpose plugin mechanism
- Fault-tolerant
  - For SLURM daemons and (optionally) its jobs
- Secure
- System administrator friendly
  - Simple configuration file, supports heterogeneous clusters
- Scalable to thousands of nodes

SLURM Plugins

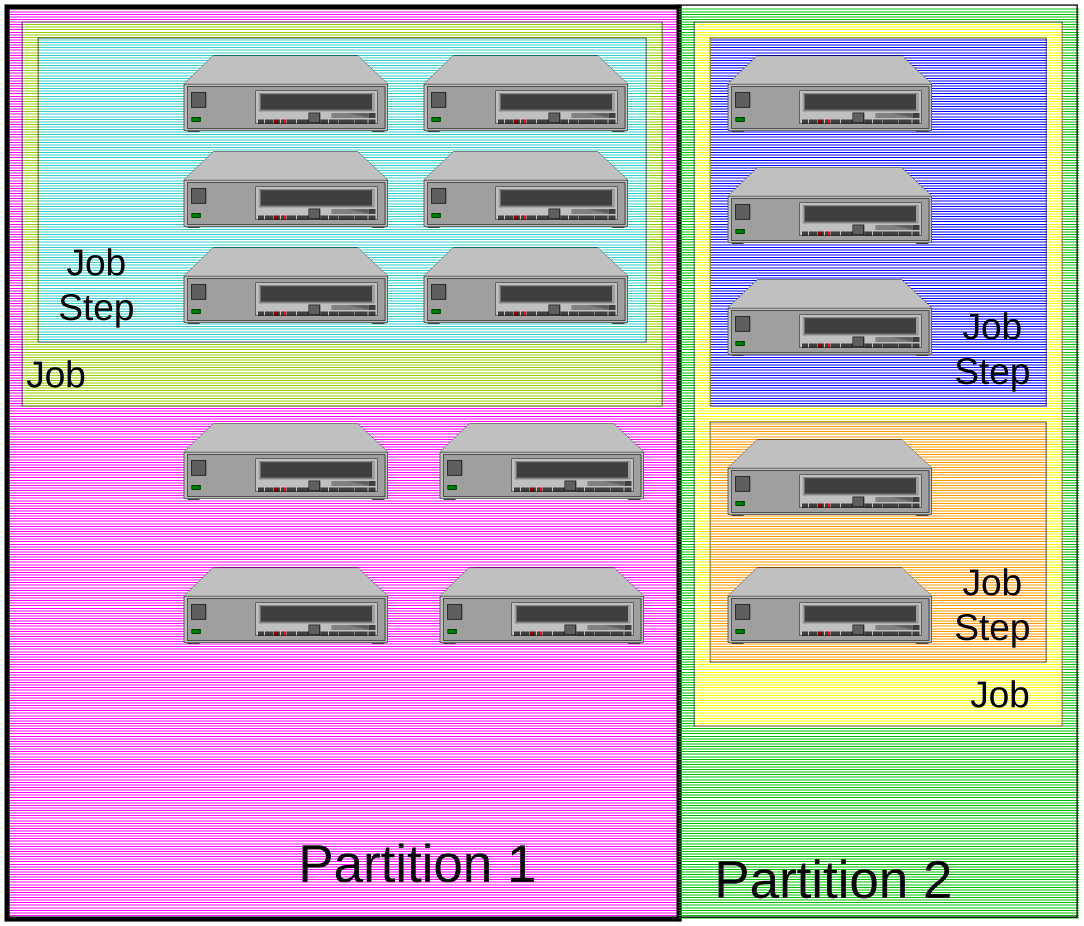
- Dynamically linked objects loaded at run time per configuration file

- Authentication
  - Authd, Munge, or none
- Interconnect
  - UDP/IP, Quadrics Elan3, or Myrinet
- Scheduler
  - Maui or FIFO



SLURM Entities

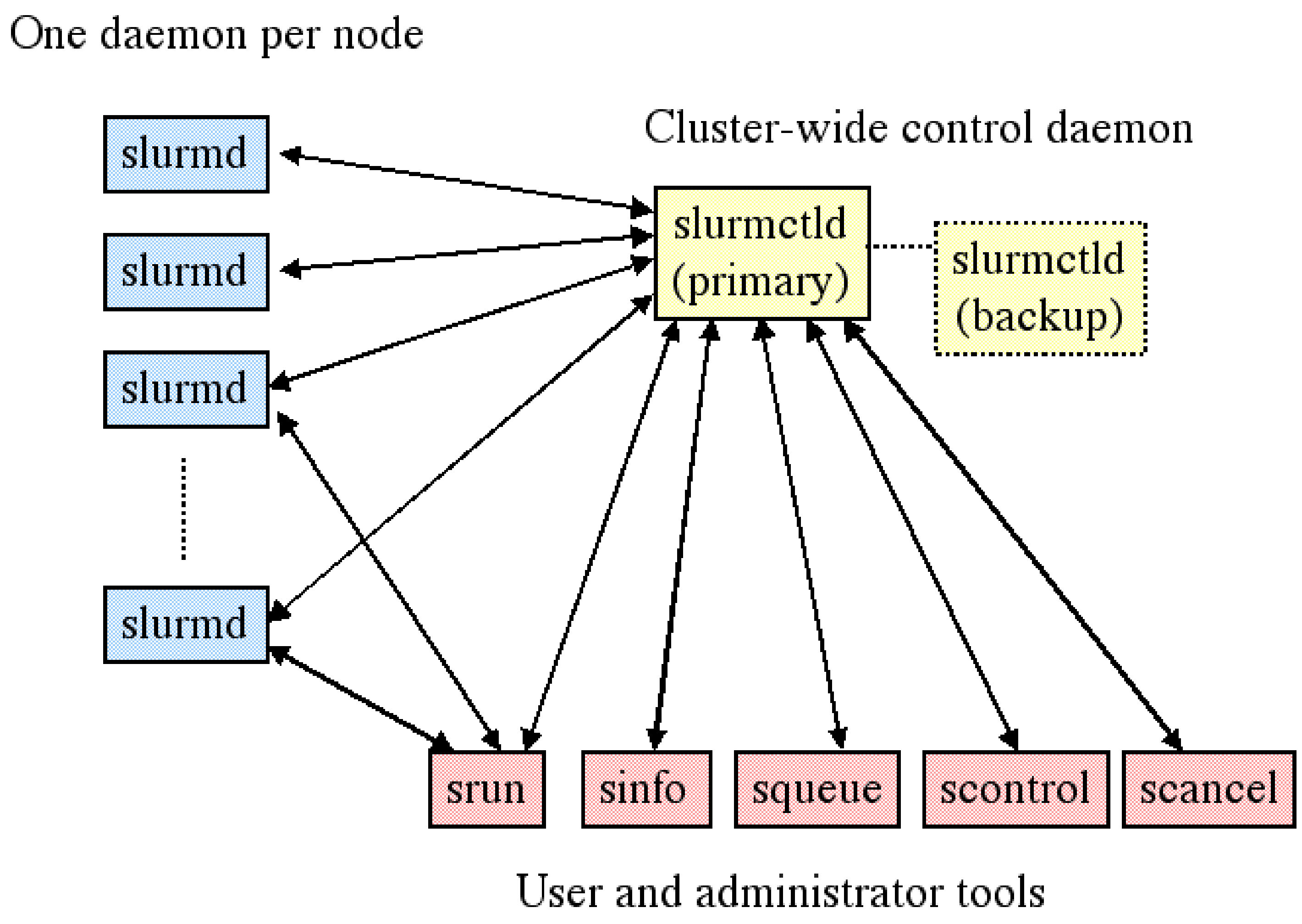
- Nodes
- Partitions
- Jobs
- Job steps



SLURM Architecture

- Two daemons
  - slurmctld - controller, optional backup
  - slurmd - computer node daemon
- Five user commands
  - scontrol - administration tool, get/set configuration
  - sinfo - reports general system information
  - squeue - reports job and job step information
  - srun - submit/initiate job or job step
  - scancel - signal or cancel a job or job step

SLURM Architecture



slurmctld

- Controller of SLURM (with optional backup)
- Multi-threaded
- Independent read and write locks by data structure
- Nodes represented with bit-maps for rapid manipulations

- Components
  - Node Manager - node state information
  - Partition Manager - allocates nodes
  - Job Manager - manages queue of pending jobs

slurmd

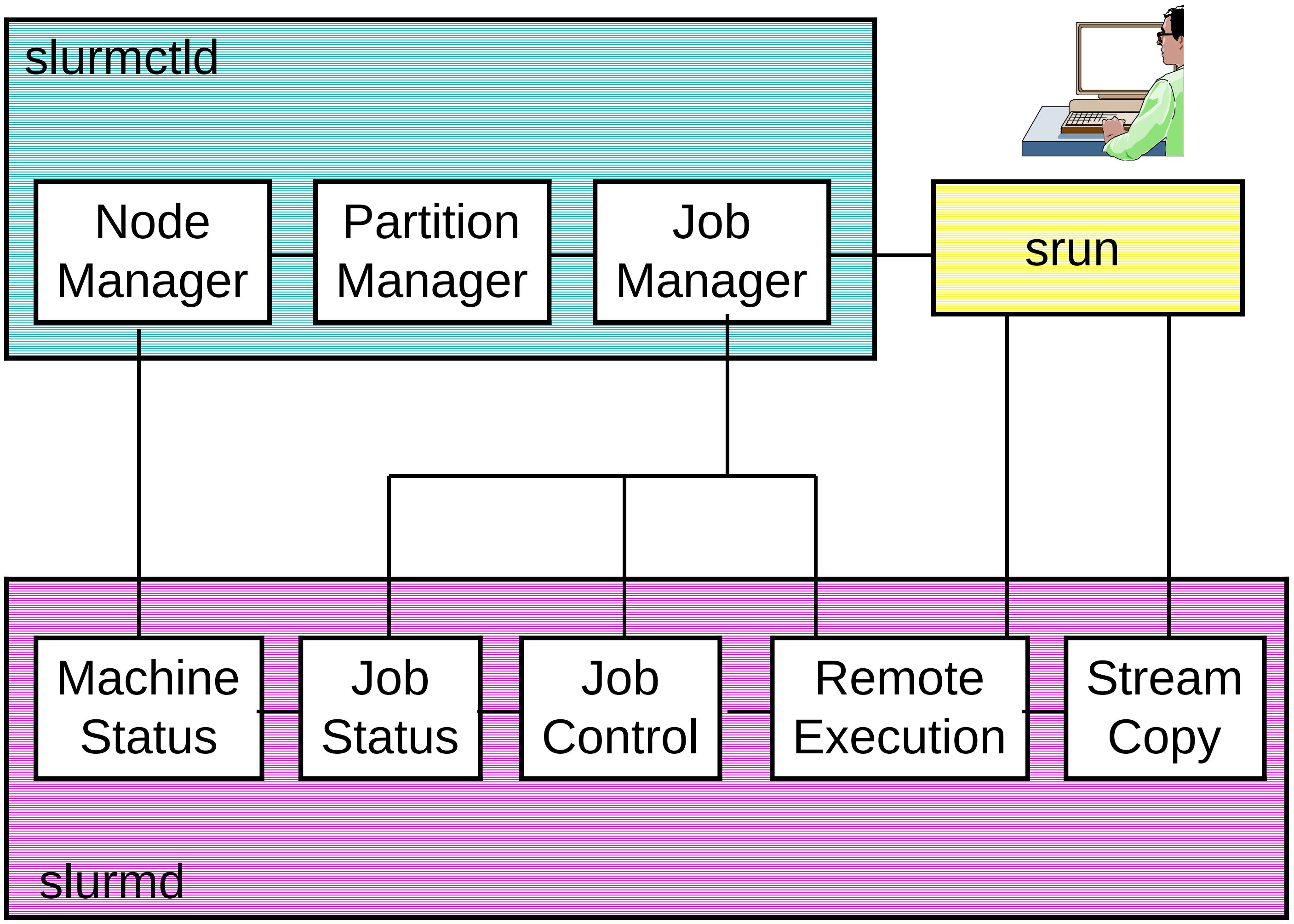
- Daemon executing on each compute node
- Minimal state information
- Performs actions as directed by slurmctld and srun

- Components
  - Machine Status
  - Job Status
  - Remote Execution
  - Stream Copy (stdin, stdout, and stderr)
  - Job Control

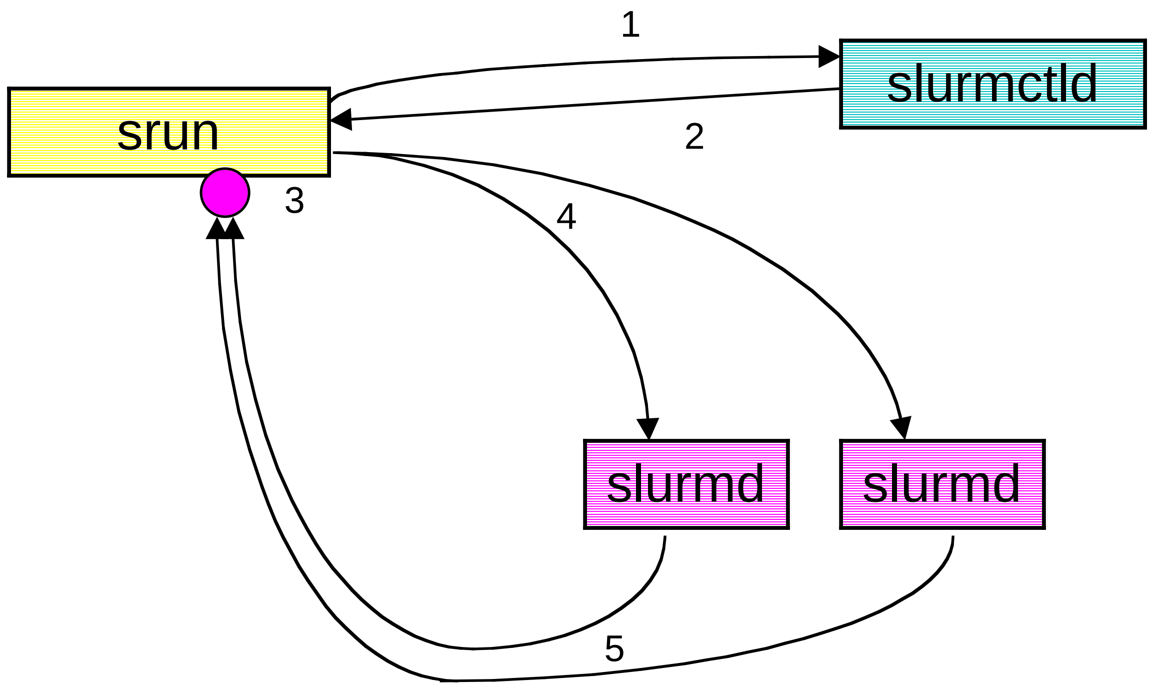
srun

- User tool to initiate jobs and job steps
  - Allocate resources
  - Submit batch jobs
  - Run jobs interactively
  - Attach to currently running job
  - Launch a set of parallel tests (job step)
- 13 options to specify resource requirements
  - Partition, processor count, node count, minimum memory per node, minimum processor count per node, specific nodes to use or avoid, node can be share, etc.

SLURM Subsystems



Interactive Job Initiation Overview



1: srun connects to slurmctld requesting resource allocation and step creation
2: slurmctld responds with node list and job step credential
3: srun opens I/O connection (ephemeral port)
4: srun sends job step requests to slurmd daemons
5: slurmd initiates job step and makes I/O connections to srun

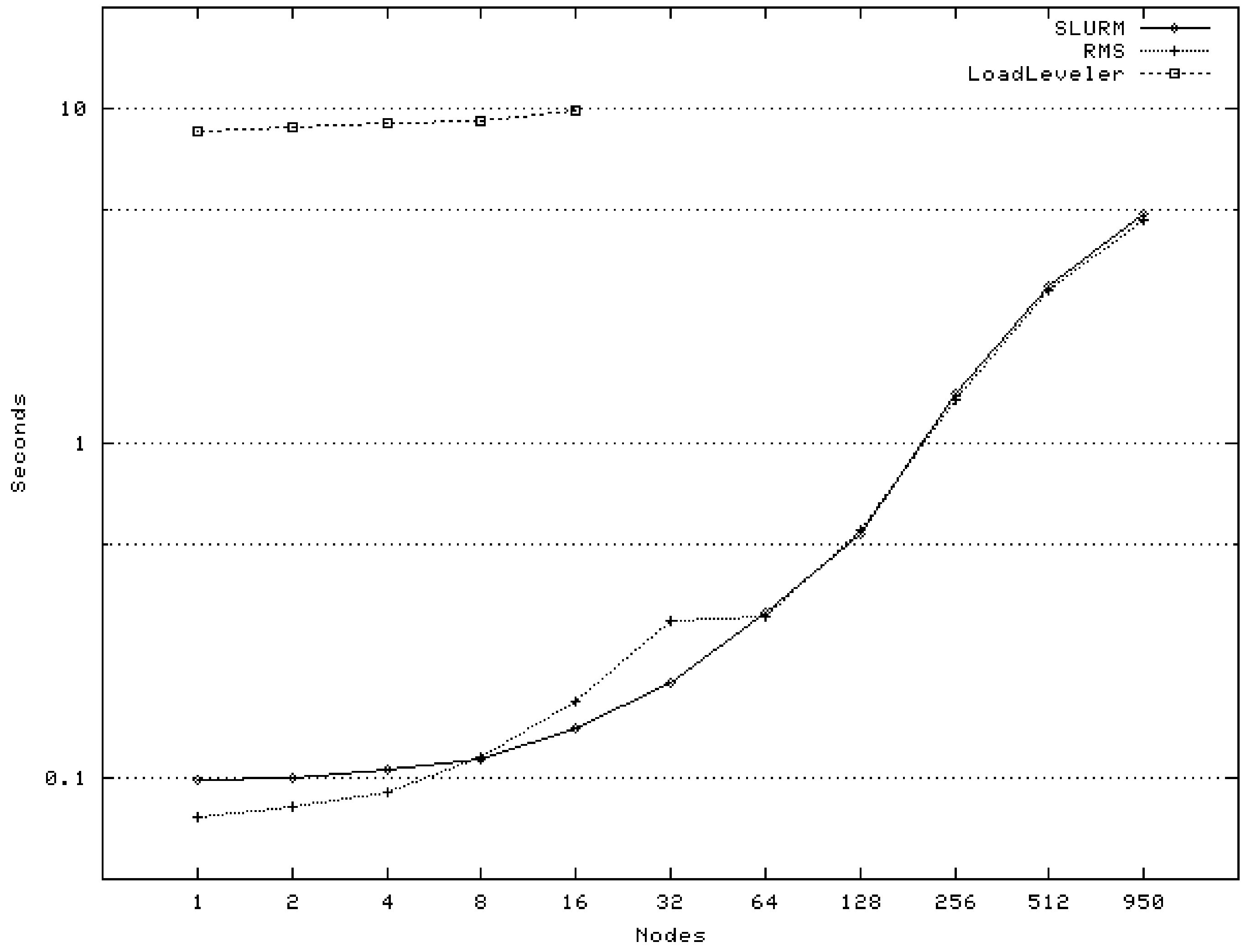
Sample SLURM Configuration

(excerpt)

```
# Sample SLURM configuration (excerpt)
ControlMachine=linux0
BackupController=linux1
#
AuthType="auth/authd"
HeartbeatInterval=60
PluginDir=/usr/lib/slurm
SlurmctldPort=7002
SlurmdPort=7003
SlurmUser=slurm
#
NodeName=DEFAULT Procs=2 TmpDisk=64000
NodeName=linux[2-1000]      RealMemory=16000 Weight=16
NodeName=linux[1001-1016] RealMemory=32000 Weight=32
#
PartitionName=debug Nodes=linux[2-33]      MaxTime=30
PartitonName=batch   Nodes=linux[34-1016] MaxTime=Infinite
```

Performance Results

/bin/hostname, 2 tasks per node



SLURM Plans

- Support more systems
  - Infiniband, IBM Blue Gene/L
- Integrate slurmctld with relational database
- Convert daemon communications to plugin
  - Support broadcast (e.g. STORM)
- Job preempt/resume
- Job checkpoint/restart