SLURM Integration with IBM Parallel Environment

SLURM User Group Meeting
October 9-10, 2012
Barcelona, Spain

Morris Jette and Danny Auble
[jette,da]@schedmd.com

IBM Parallel Environment (PE)

- Parallel application development and execution environment
  - Parallel Operating Environment (POE) to submit and manage jobs
  - IBM MPI and LAPI communication libraries
  - Parallel debugger (PDB)
  - Performance analysis toolkit
- Tightly integrated with LoadLeveler
  - Provides management of network resources

Integration Goal

- Keep entire IBM Parallel Environment
- Eliminate the need for LoadLeveler

New Infrastructure

- Developed new job step launch plugin infrastructure
  - Developed *launch/poe* plugin to interface with POE
  - Moved existing launch logic into *launch/slurm* and *launch/runjob* (for IBM BlueGene) plugins
- Expanded capabilities of existing SLURM switch plugin infrastructure
  - Developed *switch/nrt* plugin to interface with IBM network API (NRT = Network Resource Table)

Launch/POE Plugin Operation

- Translates *srun* options to the extent possible and executes *poe* to launch tasks
  - The *poe* command can also be invoked directly
- *poe* spawns its own daemon (*pmdv12*) on compute nodes
  - New library for *poe* to interact directly with SLURM (e.g. allocate resources, set environment variables, spawn tasks)
  - A complication: *poe* spawns *pmdv12* on compute nodes in a piecemeal fashion, not all at one time
- *pmdv12* spawn user tasks, manages I/O, signals, exit codes

Launch/POE Plugin Operation

1. User invokes *srun* command
2. *srun* creates job allocation (if needed) and job step allocation
3. *srun* invokes *poe* command
4. *poe* spawns *pmdv12* processes using SLURM library with *launch/slurm* plugin (can't use *launch/poe* again!)
5. *poe* tells pmdv12 to spawn application tasks

Switch Plugin Overview

- Network specific information managed within the plugin
- The plugin uses opaque data types to maintain node and step specific information
- SLURM kernel uses plugin functions to perform all operations on these opaque data types (e.g. read, write, allocate, etc.)
- Existing infrastructure for IBM Federation switch could be partly reused, but the network interface was completely different

Switch/NRT Plugin Operation Node State

- slurmd daemon (compute node)
  - Read network state at startup using NRT API
    - Switch windows and other resources
  - Clears network state on cold-start using NRT API
  - Send state to slurmctld daemon
- slurmctld daemon
  - Maintains state of all network resources on all nodes

Switch/NRT Plugin Operation Step State

- slurmctld daemon
  - Allocates and deallocates resources to job steps
    - Retry logic if insufficient resources
  - Maintains record of network resources allocated to all job steps
  - Sends step network resource information to slurmd daemons
- slurmd daemon (compute node)
  - Performs network resource allocation/deallocation operations for job steps using NRT API

Switch/NRT Plugin Operation

1. Slurmd daemon starts, discovers “InfiniBand” switch with 128 windows, reports state to slurmctld daemon
2. Slurmctld daemon maintains switch state information about all nodes in opaque data type
3. User submits MPI job step requesting some number of switch windows
4. Slurmctld allocates specific switch windows on each node and records in job step opaque data type which is included in credential used to authenticate task spawn request
5. Slurmd daemon allocates specified switch resources and spawns job step

Switch/NRT Plugin Operation

- Reverse the process for job step termination
- Support also available for job suspend/resume, preventing MPI timeouts

Status

- Available in SLURM version 2.5
- IBM Parallel Environment User and Administrator Guide for SLURM: http://www.schedmd.com/slurmdocs/ibm-pe.html