

System Administration Data Under CLE 2.2 and SMW 4.0

Jason W. Schildt
Cray Inc.

Talking Points

- Cray Management Services (CMS) – Group Charter
- The Problem with Log and State Information
- Solutions – CMS Log Manager
- Solutions – CMS State Daemon
- Future Functionality
- Summary, Questions, and Contact Information

Abstract

- With SMW-4.0 and CLE-2.2, Cray is making significant improvements in how system administrators can access information about jobs, nodes, errors, and health/troubleshooting data. This talk and paper will explain the changes and how administrators can use them to make their lives easier.

Cray Management Services (CMS) Group Charter

- The purpose of the Cray Management Services group (CMS) is to provide a common set of system management tools and infrastructure that allow customers to administer Cray supercomputers and maximize system reliability, stability, customer usability, without unreasonably impacting performance.

The Problem With Log and State Information

Everything in its right place vs. Everything all over the place

Lack of centralized log and state information

- Console logs
- Events
- System Database (SDB)
- *syslog*
- ALPS Reservations and Claims
- RAID errors
- Boot Node *syslog*
- Node state information from multiple sources
- Sources store data in different locations and formats
- No defined API or method to update or access data

How the Log Manager helps to resolve these problems

- Storing *syslogs*, events, and ALPS information in one place as they arrive
 - Storing hostname and *c-name* (*physloc*) for more consistent searches
 - Single log queries and search summaries.
 - Live log and event watching
 - Customize actions based upon user defined event triggers
 - Provide an API to access log data
- Performance and scalability enhancements for large and active systems
 - Granular table structures
 - Smaller indexes
 - Daily table drops vs. search and delete individual messages
 - Replicate messages in a 1-sec window
 - Buffered 1-sec window
 - Ability to store data on remote MySQL server

How the CMS State Daemon provides single source state aggregation

- Provide unified representation and format of node state information
- A set of APIs that provides access to node state information
- Resiliency and performance – State Daemon mirroring and caching
- Examples of stored information:
 - ALPS – Upon application create/start and destroy/stop:
 - Job account id, reservation start/end time, execution hostname, batch id
 - HSS - Node id, node state, node type
 - HSS - Processor type, speed, memory speed

CMS Future Functionality

- Further scaling optimizations
- Provide an APIs to access log data from anywhere on the system, utilizing access controls
- Enable log insertion via a lightweight C API, or a command
- Data streaming into the log
- Support Additional attributes by State Daemon

Questions and Follow-up Information

- Questions?
- Contact and Follow-up Information

Jason W. Schildt
CMS Software Group, Manager
Cray Inc. - Seattle, WA

(w) 206-701-2065

jschildt@cray.com

CRAY
THE SUPERCOMPUTER COMPANY