Workload Managers

A Flexible Approach

Blaine Ebeling
Marlys Kohnke
Carl Albing
HPCS R & D
Operating Systems and I/O
St. Paul, USA
Email: bce@cray.com

Abstract---Workload Managers (WLMs) are the main user interfaces for running HPC jobs on Cray systems. Application Level Placement Services (ALPS) is a resource placement infrastructure provided on all Cray systems to support WLMs. Until now, WLMs have interfaced with ALPS through the BASIL protocol for resource reservations, and the aprun command (and apinit daemon) is used for launching applications.
Over the last several years, the requirement to support more platforms, new processor capabilities, dynamic resource management and new WLM features led Cray to investigate alternative ways to provide methods for supporting and expanding WLM capabilities and new WLMs on Cray systems.
This paper will highlight Cray's plans to expose low level hardware interfaces by refactoring ALPS to allow 'native' WLM implementations that do not rely on the current ALPS interface mechanism, and the expansion of WLMs on Cray systems. Native implementations allow a WLM to present its own user commands in place of ALPS commands.

A description of the process for Cray testing, certification and support of WLMs will be included. Partnering with both our vendors and customers in the use of both commercial and open source WLMs enhances the Cray experience.

I. Introduction

Workload Managers (WLM) are supported on Cray systems as a front-end to ALPS. They work with the BASIL interface communicating with ALPS. In 2012, Cray embarked on a mission to support Workload Managers on a native level, without the use of BASIL or ALPS, on CLE systems running on Cray hardware.

This was an important and exciting decision but not one without significant challenges and even a bit of controversy. This initiative was customer driven and based on data from customers. Additionally, prospects began requesting WLMs that were not in the Cray list of supported WLMs on CLE. Cray made a decision to investigate and address these new customer requirements. This would ultimately create a more versatile and flexible approach to workload management.

II. Traditional Model

WLMs are the user interface to run HPC applications on Cray systems. The internal Cray interface to the WLMs has traditionally been BASIL, which works directly with the Application Level Placement Scheduler (ALPS). ALPS provides the placement infrastructure for the WLM. It manages resources, application launch and Cray specific services. ALPS uses aprun as its launch command and supports Cray PMI applications. Cluster Compatibility Mode (CCM) can be used to launch third party or non-Cray PMI applications. Today, Cray supports multiple WLMs on their systems, including

Moab/Torque and PBS Professional. Cray remains neutral in terms of recommending workload managers or suggesting which WLM a customer or prospect might use. The customer should acquaint themselves with the features and attributes of the WLMs and choose the WLM that is the best for their use.

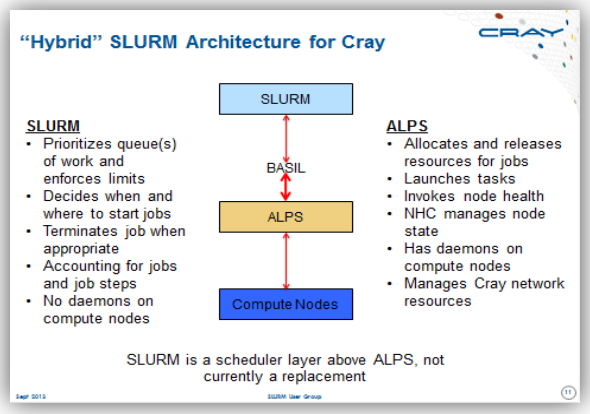
III. The Customer Voice and Motivation for Change

In 2012, once it was clear customers wanted more support for WLMs on Cray systems, Cray was interested in determining the motivation for this demand. After all, Cray had gone for several years with a limited number of supported WLMs. Customers were requesting support for LSF, GridEngine, and Slurm. Some of these requests were to deliver native WLM support on systems.

Cray asked the customers for the motivation for their requests. What was determined was that customers with large labs and datacenters that had heterogeneous systems found it was inefficient and costly to require their administration staff to know multiple WLMs with different commands, parameters and syntax. If administrators were forced to move around or administer different machines, they would frequently find they would have to learn a new WLM or would be less efficient doing their job because they were not familiar with all the WLMs.

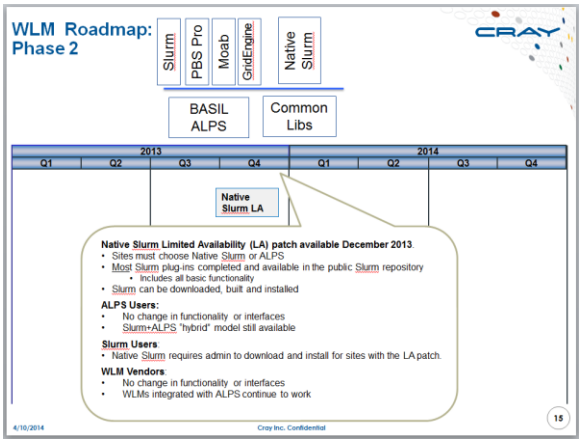
Customers wanted to limit the training on Workload Managers to a single WLM allowing their administration staff to be able to move from system to system with familiarity of the WLM being run on each system. Sites were looking to solve this problem, reduce training costs and create a more efficient operations staff. Cray started crafting a solution to support more WLMs and native WLMs.

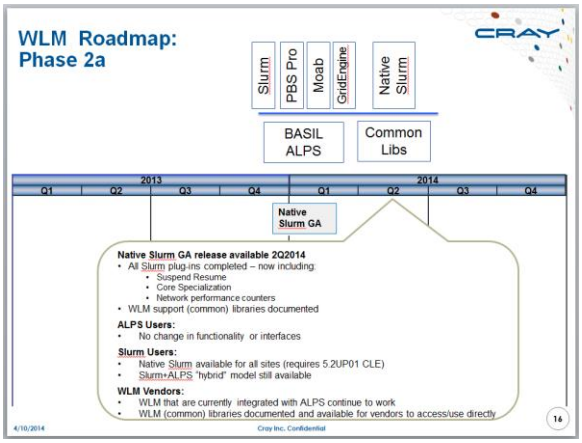
One thing that was clear was that it would take longer than customers were willing to wait to get to a full native solution. An interim solution was proposed to provide a hybrid version of Slurm to bridge the time gap. Hybrid Slurm would allow customers to run Slurm commands with a set of wrappers, making it look like Slurm, even though it was really ALPS commands that were being executed. It was an attempt to solve the problem and have administrators only have to know one set of WLM commands. The following chart represents the Slurm and ALPS functionality and commications:

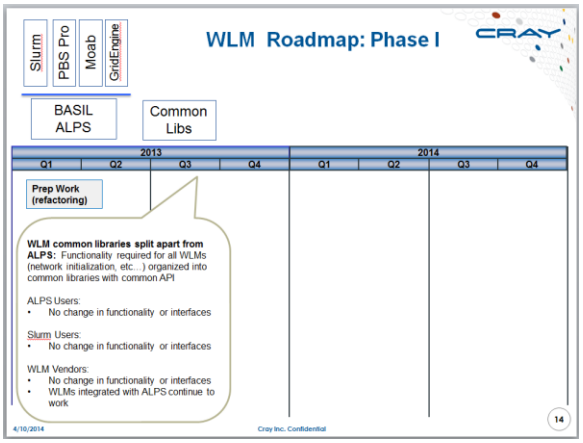


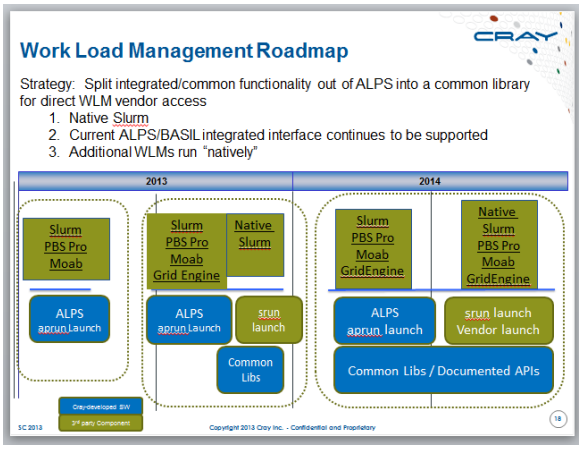
Cray immediately put a two phase plan in place to deliver Slurm and make the way for any WLM to run natively on top of CLE on Aries systems (not Gemini systems). Native mode would present new and interesting opportunities for Cray. In native WLM mode, there will be no ALPS user commands (i.e. aprun, apstat, apmgr, apkill and apbasil). There will be no visible ALPS presence for end users. The ALPS service node and compute node daemons will not be present. Three new daemons to support certain ALPS provided services will be available. The following slides depict the phases and the current state of Workload Management over time.

These charts represent the two-phase plan to develop the infrastructure for native WLMs to run on Cray systems and a two-year roadmap defining the WLMs, application launchers and the application interfaces each uses:









IV. Providing the Infrastructure for Native Workload Managers

The primary requirement for native WLM support is that a WLM be able to successfully launch Cray PMI based applications and out of the box applications on an Aries system. The first step was to create a set of library APIs that were independent of ALPS. That meant determining what functionality was required and extracting as much of this code as possible from ALPS and placing it in the common libraries.

Next was to address the Cray specific functionality required for a set of plugins. In the case of Slurm, many of the plugins Cray would need were already in place. They would only require a set of Cray specific code to be added. SchedMD created a few new plugins to be used by Cray. The plugins would be open source and would be delivered with the Slurm code.

Cray would need a method of dealing with protection keys. A new daemon was developed to run on a service node to dispense and remove cookies, which include the protection keys. A Slurm plugin makes a common library call to request cookies per application launch. This cookie information is provided through Slurm to the assigned compute nodes. A Slurm plugin makes a common library call to release the cookies following the application exit. A new command, `viewcookies`, provides information about the managed cookies.

V. Native Workload Managers

Not all workload managers have their own launch mechanism. Today our popular ones use aprun as their launcher. Since aprun would not be available in native mode another solution had to be found as we developed a general solution for native WLMs.
Using srun is an option that was explored and was determined to be one convenient option. A majority of mpirun implementations already interface to srun as a launcher. The native WLM functionality could use srun as a launcher. The WLM would still provide resource management, scheduling and reporting functions.

The following is a list of some of the infrastructure changes that had to be made to provide for native WLMs to run on top of CLE Aries systems:

- ALPS code extracted to stand alone C interface library APIs for Cray services
- Service node and compute node libraries provided
- Some API services provided in WLM plugins
- No changes in programming model code from PE
- No Node Translation Table (NTT) on the Aries systems
    - Global pKeys contained within cookies are used instead
- Three new daemons
    - **ncmd** – network cookie management daemon. One per system, runs on boot node
    - **aeld** – provides application placement info to HSS

Used in network congestion management

- **apptermd** – application termination daemon
  Kills apps as directed by network congestion management

This is a list of the specific Cray services provided:

- Cookie/protection Key management
- Configure Aries driver
- Configure reserved access to NPC
- Provide app info used with congestion mgmt.
- Memory compaction
- Compute Node Clean Up
- Flush Lustre caches
- Provide info to IAA for third party application launches
- Suspend/Resume
- Provide info requested by Cray PMI
- Provide dynamic node state change info
- Provide system topology info
- Invoke NHC following application and batch exit
- Manage PMI port assignment when more than
  one application per compute node

VI. Tools and debuggers:

Both Totalview and DDT debuggers work with Slurm and were available at the limited availability release. For the general availability release, we will add both ATP and lgdb. It is important to note that debugger tools link dynamically with ALPS libraries. ALPS provides an application startup barrier synchronization service not available in a typical WLM.

VII. Functionality Provided for use in Native Mode

- A `viewcookies` command is provided to display info about assigned protection Keys and cookies
- New ALPS APIs
- srun launcher
- WLM will provide its own user commands
  WLM can use its own daemons on the compute nodes
- Application launch will need to know if suspend/resume is enabled
  Allows for network resources to be scaled for each launch
- Third party MPI launch commands will be supported for non-Cray PMI based apps
- Native WLM will provide its own status and admin commands
- For native WLMs, batch jobs will be initiated on a compute node not a service node as happens today
- All commands within a batch job execute on a compute node
- Launcher when invoked in a batch job will execute directly on a compute node
- Interactive srun will still execute on a service node

The following options will NOT be supported. Sites that require any of these services will need to continue to run ALPS rather than a native WLM.

- aprun -P option provides read/write pipes used by certain Cray debugger tools to control when
  an application is released from its startup barrier.
- aprun -p option accepts a user defined protection domain identifier.
- aprun -mhs option requests huge pages and strict enforcement. A user can set various
  libhugetlbfs environment variables for hugepage management, which no longer requires root privilege

- aprun -R option requests relaunching an application following certain node failures.
- aprun -C option requests reconnecting an ALPS fanout control tree around a failed node,
  allowing the application to continue executing in a degraded mode.
- aprun -T option requests synchronous tty output. Slurm and perhaps other WLMs allow
  users to request application per rank separate output files, which can take the place of this aprun option.

Note: Applications linked statically prior to CLE 5.1UP01 will need to be relinked.

VIII. Functionality NOT provided in Native Mode and Native Mode Limitations

- Native mode is not supported on the Gemini interconnect
- Checkpoint/Restart is not supported on Aries systems
- user defined (user and system) protection domains feature not supported
- Following srun options are not supported
  - reboot
  - checkpoint
  - tmp
- No CSA support (Slurm accounting is provided)
- Pre-reservation of huge pages will not be supported in native WLM mode
- No RCA or event support on compute node
  - WLM must detect node failures on its own
  - No ALPS user commands in native mode
  - No ALPS log messages (tools use these)
- No autonomous mode KNC support (accelerated mode works)
  for nodes that are shared by multiple applications (e.g. MAMU)
- Per-job power data and per-job power capping are not supported
  both are supported for nodes used exclusively
- Only one job can use the GPU on a node (this may change to requiring exclusive use of a node when using a GPU, depending on details of nvidia proxy)
- The Slurm database should run on an external nodes, accessible from the node running slurmctld
- Slurm affinity options apply to PE ranks only. Any threads and/or child processes created by PEs will inherit the affinity of their parent.
- Warmswap will require manually editing/regenerating Slurm.conf; the updates will
  not happen automatically.

There are a few limitations which are specific to compute nodes and service nodes. Those are listed here:

IX. Compute Node Limitations

- No application startup barrier synchronization
- No debugger assistance involving application startup barrier synchronization(including aprun -P)
- No automatic killing of tool helper processes following application exit (i.e. debugger processes)
- No huge pages reserved as part of application launch (application and end user to manage huge pages)
- No delay in sending SIGKILL signal to application process dumping core
- No NTT; by default the NTT is not configured on the Aries interconnect

X. Service Node Limitations

- No NUMA node specific launch options beyond what may be provided by the launch command
- No synchronous TTY writes (aprun -T)
- No user defined or system protection domains
- Network performance counters for global and blade level reserved access only
- No application relaunch and reconnect resiliency beyond what may be provided by the WLM
- No consolidated exit code and exit signal information returned to application stdout beyond what may be provided by the WLM
- No consolidated resource usage information returned to application stdout beyond what may be provided by the WLM
- No application and reservation information provided through mazama
- No syslog messages for application start, finish and error beyond what may be provided by the WLM

XI. Running Slurm in Native Mode

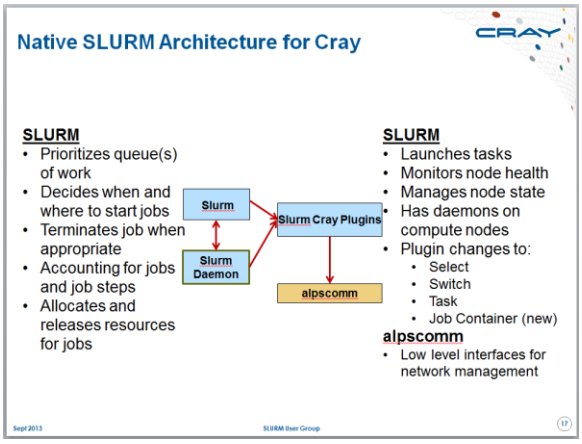
Cray support is only available for Slurm with the 5.2UP01. This is the general availability release. A customer intending to run Slurm would need to acquire the Slurm package from the SchedMD repository. All components required to build, install and run Native Slurm are in the repository.

Customers running Native Slurm will be on the Slurm community feature roadmap. They would have immediate access to the features planned and developed by SchedMD and community developers.
Below is the Roadmap for Slurm and seven features that are currently planned:

- Energy Accounting and External Sensor Plugins
- Debugging Large Machines
- Fault Tolerant Workload Management
- Slurm Layouts Framework
- License Management
- Multi-cluster Management
- Depth Oblivious Hierarchical Fairshare Priority Factor

Similarly, for each of the vendors running native mode would allow a customer to be on that vendor's roadmap of feature functionality. This could mean earlier availability of certain features.

This chart represents the functionality controlled by Slurm in native mode and the paths of communication between Slurm and the Cray library application interfaces:



XII. The Future of ALPS

ALPS is not being deprecated or removed and will remain a perfectly viable option for customers who want to continue to run as they have been for some time using a WLM running in tandem with ALPS.
All existing WLMs will remain as options. There is no intent to add Cray functionality to any WLMs or vice versa add any WLM functionality to ALPS. It should also be noted that there is no WLM specific code in any of the alpscomm library APIs.

Although the first endeavor into native workload management was with Slurm, the

decision to do Native Workload Management was made with all WLMs in mind. The coding effort was done independently of any specific WLM. Once the CLE5.2UP01 release is available, it is possible to port any WLM to CLE and have it run natively. The decision to port a WLM to CLE natively will be made individually by our vendors and not by Cray.

The initial release to support native Slurm was made available on a limited basis in December 2013. Next up is the general availability release in 2014, which is expected at the end of the second quarter.

The additional features which will be provided are listed below:

- Core Specialization
- Network Performance Counters
- GPU Support
- MIC Support for accelerated mode
- Suspend/Resume – Job Preemption
- lgdb and ATP debugger support
- Third Party Application Launch Support
- Defect fixes

It is likely the Slurm base will be 14.03 and an additional patch will also need to be provided.

XIII. Workload Manager Support Site

A web portal with a great deal of information about the Workload Managers is available for all customers. There is a separate tab for each WLM with direct links to the actual vendor WLM web site. Information is provided about the supported versions of WLM software. Certified versions are listed along with a definition of what it means to be Cray certified. Known problems are documented and cross reference tables are provided to determine which versions of WLMs can be used with unique versions of CLE and which versions of BASIL are included in each release of CLE.

The link to this site is the following:

- **http://crayport.cray.com/3rdPartyBatchSW/Forms/compatibility_info.aspx**

After clicking on this link you will be asked for your CrayPort credentials or asked to register for an account.

Cray would be interested in any feedback you have on this site. Please send feedback to spswlm@cray.com