Reducing Cluster Compatibility Mode (CCM) Complexity

Marlys Kohnke
Cray Inc.
St. Paul, MN USA
kohnke@cray.com

***Abstract*—Cluster Compatibility Mode (CCM) provides a suitable environment for running out of the box ISV and third party MPI applications, serial workloads, X11, and compiling on Cray XE/XC/XK compute nodes.**

At times, customers have experienced CCM issues related to setting up or removing that environment. The tight coupling of CCM to workload manager prologue and epilogue services has been a primary source of issues. A new configurable ALPS prologue and epilogue service specific to CCM has been provided. Removing the workload manager dependency has reduced the CCM complexity. Other problem areas have been identified, and solutions have been implemented to avoid or correct those issues.

This paper describes changes made to CCM to reduce the CCM complexity and provide a more robust, workload manager independent product.

Keywords-cluster compatibility mode; CCM; ISV applications; third party MPI applications; Cray XE/XC/XK

I. Introduction

Cluster Compatibility Mode (CCM) on Cray Cascade systems provides a cluster environment required by ISV and third party MPI applications to run unmodified out of the box using either ssh or rsh for the application launch. The current CCM complexity in providing that environment will be reduced, leading to a more robust, workload management (WLM) independent CCM product.

Third party MPI applications and ISV applications expect to be launched from the batch job head node, which is the first compute node within the list of compute nodes reserved for the batch job. CCM configures the reserved set of compute nodes to allow ssh or rsh access by the batch job owner. CCM starts the application launcher (e.g. mpirun, mpiexec) on just the head node. The application launcher uses ssh or rsh to access each of the reserved compute nodes. It is the responsibility of the application launcher to get the actual application executing on all of the reserved compute nodes.

CCM has the following requirements:

- install and run ISV and third party MPI applications out of the box without modification (some applications may require minimal modifications to the application launch script)
- support passwordless ssh and rsh application launch within the compute nodes assigned to the batch job and restricted to the batch job owner
- provide resources for external network connectivity (e.g. access licensing)
- provide resources for user identity information (e.g. name service resolution)
- avoid interference with extreme scalability mode (ESM) applications

CCM provides two user commands to access the cluster environment set up by CCM. The **ccmrun** command is used to initiate the launch of ISV and third party MPI applications on that set of reserved compute nodes. The **ccmlogin** command is used to start an interactive ssh session to a compute node within that set of reserved compute nodes.

CCM initialization provides the following services in support of ccmrun and ccmlogin. These services are provided through a set of scripts and binaries. A WLM is required for ccmrun and ccmlogin use.

- provide passwordless ssh login access to the batch job head compute node restricted to the job owner
- provide a temporary writeable /tmp on compute nodes
- provide infrastructure support for user authentication services from compute nodes
- start additional network services on compute nodes (e.g. rpcbind, xinetd, nscd, ypbind)
- start the dbus daemon for potential application process to process communication
- provide batch job owner passwordless restricted access for ssh or rsh access between the set of reserved compute nodes
- acquire Aries cookies and configure the Aries driver per application launch and per compute node login
- perform various bind mounts on compute nodes
- provide per application information to ISV Application Acceleration (IAA)
- support ssh and rsh application launch using TCP/IP over the Aries interconnect

Both ccmrun and ccmlogin use the ALPS aprun command to launch a CCM **ccmlaunch** program, which starts the required system daemons (e.g. xinetd, dbus, nscd, etc.) on the compute nodes prior to executing the application launcher. To reduce unnecessary jitter on compute nodes for ESM applications, this set of system daemons is only run within a CCM environment. A non-busybox, application compliant sshd daemon listening on a CCM defined port is started and configured by CCM to restrict access to just the batch job owner for ccmrun and ccmlogin.

Since these additional network services daemons are started by ccmlaunch, these daemons are stopped by ccmlaunch after the application exits. These daemons do not continue to execute between ccmrun and ccmlogin invocations.

The ALPS application startup barrier synchronization service is used by ccmlaunch to ensure that the CCM required setup is complete on all the reserved compute nodes before the application launcher command is executed on the head compute node by ccmlaunch.

CCM works together with IAA to provide an application transparent interface to use the Aries interconnect network in place of an infiniband network the application assumes is available. CCM bind mounts the following infiniband related files into standard locations within the CCM environment on compute nodes:

- libibverbs.so (Cray modified)
- libibgni.so
- ibstat

The modified libibverbs.so library is used by IAA to intercept ibverbs calls made by the application and direct that activity to use the Aries interconnect for better performance. The maximum IAA supported number of application processes has increased from 2048 to 4096.

II. Issues

At times, users have experienced CCM issues related to setting up or removing the CCM environment. CCM has had a tight coupling with WLMs through batch job prologue and epilogue services responsible to invoke the CCM prologue and epilogue scripts at batch job boundaries. Each WLM determines constraints and behavior related to execution of those scripts, which can negatively impact CCM activities.

CCM provides a cray-ccm-prologue script to be used with a WLM batch job prologue service associated with a CCM specific batch queue or resource. This script can only be executed by root and initiates CCM setup activities for the batch job prior to an application launch by ccmrun or a compute node login by ccmlogin. That queue/resource also invokes a CCM provided cray-ccm-epilogue script to do CCM post batch job exit cleanup.

A new CCM prologue and epilogue service is now provided by ALPS. The WLM batch prologue and epilogue services are no longer used. See the implementation section for further details.

The following are some examples of issues which have been identified and corrected.

1) The CCM prologue script was using xtxqtcmd to cat the hosts nodelist file (e.g. PBS_NODELIST) to the compute nodes. This could fail due to the number of nodelist entries and the kernel maximum arguments limit.

The hosts nodelist file is now created locally on each compute node by CCM, which removes the need to move the file contents from a MOM node to the compute nodes.

2) WLM failover use of temporary files can affect WLM status command output used by the CCM prologue script, which can cause the prologue script to fail.

The CCM prologue no longer uses the WLM prologue service and no longer does screen scraping of apstat and the WLM status command (e.g. qstat, scontrol).

3) The CCM prologue used apstat and the WLM status command output to gather information. Any change in the apstat or WLM status command output layout could break the CCM screen scraping of that data, which can cause the prologue script to fail.

Since the CCM prologue is now invoked by ALPS, there is no screen scraping within the prologue script.

4) Under certain conditions involving multiple batch job submissions within a short period and delays involving NFS caching of ALPS spooled files accessed by apstat, the CCM prologue could encounter delays in getting the required apstat information. Depending upon the length of the delays, the prologue script could fail.

ALPS now provides the required information to the CCM prologue through environment variables. This avoids the screen scraping delays.

5) The CCM prologue script required WLM specific code to parse the WLM command output for use by CCM. CCM support for each new WLM required code changes to CCM.

This WLM specific code has been removed since ALPS now provides the required information and services.

6) The CCM epilogue script was not run by the WLM or a WLM timeout was hit, which caused the CCM epilogue to be killed before it completed its cleanup activities.

ALPS is now providing a CCM epilogue service. There is a configurable ALPS timeout for the epilogue.

7) NHC was invoked directly from the CCM epilogue script, which could be executing at the same time as NHC invoked by ALPS for either application or reservation checks.

A CCM plugin test has been added to NHC. The CCM epilogue no longer invokes NHC. This NHC CCM test unmounts the CCM created bind mounts. The NHC nhc_ccm_test.sh has to be enabled within the NHC configuration file or the CCM prologue will error exit.

III. Implementation

Changes were made to CCM and ALPS to provide a more robust, less complex CCM product. ALPS now provides a CCM prologue and epilogue invocation service to replace the WLM provided prologue and epilogue service. New CCM ccm-prologue and ccm-epilogue scripts are available, which have been modified to remove the screen scraping of apstat and the WLM status command output and to get that information from environment variables set by ALPS.

The prior CCM cray-ccm-prologue and cray-ccm-epilogue scripts and WLM provided prologue and epilogue services will continue to be supported for just one CLE update release once the new CCM scripts and ALPS services are available within a CLE update release. CLE release notes will describe how to disable CCM use of the WLM prologue and epilogue services. The ALPS CCM prologue and epilogue services are always enabled and are triggered by aprun being part of a CCM invocation.

1) Prologue Changes

The following changes have been made to the new CCM ccm-prologue script.

- The CCM ccm-prologue script no longer does any screen scraping. The required information is obtained from environment variables set by ALPS. If any of the required environment variables are not found, the prologue script will error exit.
- An appropriate hosts nodelist file is created locally on the compute nodes using a combination of local node placement information from ALPS and a compact string created by ccm-prologue giving the PE/rank counts per node in the placement list. This compute node local nodelist file will have the same contents as the original service node file.

  For example, if the service node nodelist contains host names for nids 30,30,31,31,32,32,33, then the compact string format or PE-count:number-of-nodes would be 2:3,1:1, indicating there are 2 PEs/ranks for the first 3 nodes and 1 PE/rank for the remaining node. The ALPS compute node placement information contains nids 30,31,32,33. That information is used with the compact string to expand out to a hosts nodelist of host names for nids 30,30,31,31,32,32,33 again on the compute nodes.

  Xtxqtcmd is now used to cat the compact string to the compute nodes, not the entire expanded hosts nodelist file.
- The prologue script verifies that NHC is configured with the CCM nhc_ccm_test.sh plugin. If this plugin is not enabled, the prologue script will error exit. This plugin runs as part of the reservation set when the CCM batch job reservation is ready to be canceled by ALPS. The plugin does CCM environment cleanup activity that used to be done by the CCM epilogue.
- The prologue script checks if the CCM prologue has already been run for this batch job. If so, the script immediately returns since there is no additional initialization work to be done.

ALPS now provides a CCM prologue and epilogue invocation service. The following changes have been made to ALPS components.

- The /etc/opt/cray/alps/alps.conf file has configurable timeout values related to CCM:
  - prologTimeoutCCM, default 120 seconds
  - epilogTimeoutCCM, default 120 seconds

  There are also prologPathCCM and epilogPathCCM configuration variables, but those are intended for internal testing use. Customers can ignore those path variables.
- When aprun is started as part of ccmrun or ccmlogin, aprun sets a flag to tell ALPS apsys and apsched that this is a CCM related launch. Apsched stores that flag as part of the batch reservation data. After a placement list is assigned by apsched for that aprun claim, apsys invokes the new CCM ccm-prologue script. Apsys sets environment variables to provide required information to the ccm-prologue script.

  Apsys will invoke the CCM ccm-prologue script as part of each ccmrun or ccmlogin within a CCM batch job. CCM itself will process the first prologue script invocation and immediately return for any subsequent invocations for that batch job. The apsys child waits for the prologue script to complete and exit.

  This ALPS CCM prologue service is provided at the aprun level, not at the batch job reservation level. There is currently no scalable way at reservation creation time to know if this reservation is for a CCM batch job. Therefore, the first ccmrun or ccmlogin within a batch job will trigger the CCM prologue setup activities.

  An error exit by the CCM prologue script is considered a fatal launch error. Aprun will write a message to stderr and syslog before exiting. If there is no error, then apsys will invoke any configured prologPath executable defined in alps.conf (currently used by RUR). Following that, the assigned placement list will be forwarded to aprun to start the CCM launch on the compute nodes.
- Apsys sets the following environment variables required by the ccm-prologue script:
  - ALPS_PREP_BATCHID, batch job identifier
  - ALPS_PREP_UID, the aprun user id
  - ALPS_PREP_NIDFILE, /tmp/apsysCCMXXXXXX unique filename
- Apsys writes placement information into the ALPS_PREP_NIDFILE file with a nid entry per PE/rank on each reserved node.

2) CCM Program Changes

The following changes have been made to ccmrun, ccmlogin, and ccmlaunch.

- Ccmrun sets an environment variable to tell aprun this is a CCM invocation. The aprun "-n" option value is no longer determined and set by ccmrun. Aprun itself calculates the appropriate value from reservation information.
- Ccmlogin also sets the environment variable before invoking aprun. Netcat continues to be used to check if the CCM started compute node sshd is

listening on the special CCM port number. A new CCM SSH_MAX_CONNECTION_TIMEOUT configuration variable is used (or a default of 20 netcat calls with a 1 second sleep between calls) with netcat.

- Ccmlaunch has been changed to get additional placement information from ALPS on the compute nodes. As described earlier, this information is used along with a CCM created compact string to locally create the hosts nodelist file.

3) *Aprun Changes*

The following changes have been made to aprun.

- If this is a CCM launch, aprun makes a library call using the aprun PAGG job id as a primary key to get the batch job reservation list of reserved compute nodes. Aprun then calculates an appropriate "-n" number of processes value to provide to apsched and apinit.
- Aprun sets a flag value to tell apsys and apsched that this is a CCM launch. This flag causes apsys to invoke the CCM prologue after a placement list is assigned by apsched. Apsched stores that flag and provides it later to apsys when the reservation is ready to be canceled. Apsys then invokes the CCM epilogue script.

4) *Epilogue Changes*

The CCM ccm-epilogue script has been simplified to no longer directly invoke NHC or do CCM cleanup activities following the batch job exit. NHC has a CCM plugin to do the CCM environment cleanup work.

IV. Design Decisions

The CCM user interface remains the same. An end user can still invoke ccmrun and ccmlogin without change.

CCM copies the batch job owner ssh public keys to the set of reserved compute nodes and modifies a copy of the ssh and sshd configuration files to allow compute node passwordless ssh access to the CCM batch job owner. These modified configuration files are used with the sshd started by CCM.

Trusted host ssh authentication was considered to replace the ssh user key authentication used by CCM. This idea was rejected since trusted host authentication is less secure and would require sites to become familiar with configuring the Cray system for this type of authentication. This trusted host ssh configuration change would be persistent, not just for CCM batch job activities.

The CCM prologue is invoked by ALPS at an aprun boundary rather than a batch job boundary. The CCM environment setup work done by the prologue remains in effect until after the batch job exits. Consideration was given to modifying the BASIL reservation creation method for the WLM to provide CCM queue name or CCM resource information to ALPS. However, there are usually delays in WLM support of new BASIL revisions. The decision was made to invoke the CCM prologue at an aprun boundary and have CCM manage only executing that ccm-prologue script once per batch job.

V. Troubleshooting

In addition to the current CCM troubleshooting data, the following files contain information about CCM:

- NHC writes error information to the console logfile related to the CCM nhc_ccm_test.sh plugin.
- Apsys writes start and end information about each CCM prologue and epilogue invocation within the apsys logfile. With the default apsys debug level, apsys also writes information about the temporary file containing placement list information.
- CCM writes additional information to the ccm-YYYYMMDD file on the SMW.

Here are examples of nhc_ccm_test.sh failure messages written to the console logfile:

```
2015-03-24T13:00:33.140248-05:00 c0-0c0s5n0 <node_health:5.1> RESID:39314901 (nhc_ccm_test.sh) Unable to unmount CCM file system: /var/crayccm/var/ccm/ccm.run (/var/run)
2015-03-24T13:00:33.221287-05:00 c0-0c0s5n0 <node_health:5.1> RESID:39314901 (nhc_ccm_test.sh) Unable to unmount CCM file system: /var/crayccm/tmp (/tmp)
2015-03-24T13:00:33.297372-05:00 c0-0c0s5n0 <node_health:5.1> RESID:39314901 (nhc_ccm_test.sh) Unable to unmount CCM file system: /dev/urandom (/dev/random)
2015-03-24T13:00:33.424113-05:00 c0-0c0s5n0 <node_health:5.1> RESID:39314901 (Plugin) WARNING: Process (nhc_ccm_test.sh) returned with exit code 2
```

Here are examples of apsys logfile entries for the CCM prologue and epilogue activity. The batch job id is included in the messages. The apid is for the aprun which is invoked by ccmrun or ccmlogin.

```
2015-03-30 10:58:01: [21057] makePEnidFile:2879: Using appinfo file to create CCM PE nid file /tmp/apsysCCMpuz7aN for batch_id 2960.opal-p1, resid 2147818, apid 122224
2015-03-30 10:58:01: [21061] Agent for batch_id 2960.opal-p1 apid 122224 CCM prolog: /opt/cray/ccm/default/etc/ccm-prologue
2015-03-30 10:58:05: [21057] Agent for batch_id 2960.opal-p1 apid 122224 CCM prolog done (0)
...
2015-03-30 11:06:02: [21309] Agent for batch_id 2960.opal-p1 apid 2147818 CCM epilog: /opt/cray/ccm/default/etc/ccm-epilogue
2015-03-30 11:06:03: [21305] Agent for batch_id 2960.opal-p1 apid 2147818 CCM epilog done (0)
```

Here are examples of CCM messages which used to be written to the batch job stdout and are now written to the CCM logfile on the SMW. Since apsys is invoking the CCM prologue and epilogue scripts, there is no connection to the batch job stdout and stderr.

```
2015-03-30T10:58:02.256131-05:00 c0-0c0s1n1 xtlog 21077 [ccm-prologue:2960.opal-p1@34] In CCM JOB: 2960.opal-p1 USER kohnke WLM pbs
2015-03-30T10:58:02.895982-05:00 c0-0c0s1n1 xtlog 21121 [ccm-prologue:2960.opal-p1@34] 2960.opal-p1 Mon 30 Mar 2015 10:58:02 AM CDT
...
2015-03-30T11:06:02.305341-05:00 c0-0c0s1n1 xtlog 21283 [ccm_epilogue_debug@34] CCM epilogue - jobid: 2960.opal-p1 user: kohnke
```

VI. Future Work

Work may be done in the future to replicate name servers within the Cray system and configure compute nodes to use the internal name servers. Support of external name servers has been an ongoing support issue for CCM.

VII. Conclusion

CCM had tight coupling with workload managers through batch job prologue and epilogue services. Each WLM determines constraints and behavior related to execution of the prologue and epilogue scripts, which can negatively impact CCM.

The WLM prologue and epilogue services will no longer be used. ALPS will provide a service to run the CCM ccm-prologue and ccm-epilogue scripts at the appropriate times. This removes the CCM dependency upon the WLMs and the WLM specific code related to these services. With ALPS providing a CCM prologue and epilogue invocation service, additional WLMs can be supported without CCM changes.

The changes described in this document reduce the CCM complexity and allow Cray more direct control over the services needed by CCM. This should result in a more stable CCM product.

Both the prior and the new CCM implementations will be supported for one CLE update release to allow sites time to convert from WLM to ALPS provided CCM prologue and epilogue services. The prior script names remain cray-ccm-prologue and cray-ccm-epilogue while the new script names are ccm-prologue and ccm-epilogue.