Cray Linux Environment (CLE) 3.1 Software Release Overview

S–2425–31

Version 3.0 Published March 2010 Supports the 3.0 release of the Cray Linux Environment (CLE) operating system running on Cray XT6 systems.

Version 3.1 Published June 2010 Supports the 3.1 release of the Cray Linux Environment (CLE) operating system running on Cray XT and Cray XE systems.

Contents

Page

Page

Page

Tables

Introduction [1]

This document provides an overview of the Cray Linux Environment (CLE) 3.1 operating system release package and highlights new functionality and changes from previous CLE releases.

The CLE 3.1 release supports Cray XE and Cray XT systems. Throughout this document, any reference to *Cray systems* includes all supported Cray systems unless otherwise noted. For a complete description of hardware platforms supported with the CLE 3.1 release, see Table 1.

Chapter 2, Software Enhancements on page 13 and Chapter 3, Compatibilities and Differences on page 41 describe changes made since the 2.2 version of the CLE operating system software. This information is provided as a service to users and administrators who are familiar with the CLE 2.2 release.

The previous release, CLE 3.0, was not generally available. Therefore, this document is focused on the differences between the CLE 2.2 and CLE 3.1 releases. If you have a Cray XT6 system running CLE 3.0, not all features and differences described in this document are new to you; see Appendix B, Differences Between CLE 3.0 and CLE 3.1 on page 79 for a list of changes introduced with CLE 3.1. Throughout this document, this content is flagged as follows: **(CLE 3.0 – 3.1 Change)**. If you are performing a new installation or are migrating from CLE 2.2, ignore these notes.

This document does **not** describe hardware, software, or installation of related products, such as the Cray Compiling Environment or products that Cray does not provide. To determine the release levels of other software products that are compatible with CLE 3.1, see Software Requirements on page 67.

1.1 Emphasis for the CLE 3.1 Release

The CLE 3.1 release provides the following key enhancements:

- **Cray XT6 and Cray XT6m Hardware Support.** This release supports Cray XT6 and Cray XT6m systems that have AMD Opteron 6100 Series compute blades, a Cray SeaStar based system interconnection network, and SIO service blades with AMD 940 Opteron processor sockets.
- **Cray XE Hardware Support.** This release supports Cray XE6 and Cray XE5 systems that have a Cray Gemini based system interconnection network and the new XIO service blades.

- **Cluster Compatibility Mode (CCM).** New CCM functionality enables cluster-based independent software vendor (ISV) applications to run without modification on Cray systems.
- **Performance Analysis.** CrayPat can access memory mapped registers (MMRs) on the Gemini ASIC, which serve as performance counters in either the Network Interface Controller (NIC) or router tile domains.
- **SUSE Linux Enterprise Server (SLES) 11 Upgrade.** Cray's customized version of the Linux operating system is upgraded to SLES 11.
- **Lustre File System Upgrade.** The Lustre file system from Oracle is upgraded to version 1.8.
- **Dynamic Shared Objects and Libraries.** Support is provided for using dynamic shared objects with the compute node root runtime environment.
- **Core Specialization.** Functionality is provided to allow a user to bind a set of Linux kernel-space processes and daemons to a single core within a compute node to enable the software application to fully utilize the remaining cores within its `cpuset`.
- **Node Health Checker (NHC) Enhancements.** Suspect mode returns healthy nodes to the resource pool faster than in previous releases; NHC automatically recovers if a login node crashes; NHC tests are improved.
- **Stripe Parallel Mode for Cray Data Virtualization Service (Cray DVS).** Depending on the DVS block size and file offset, DVS automatically determines which server handles the client transaction.

The SMW 5.1 release also provides several key enhancements that directly impact the operation of your Cray system running the CLE 3.1 release, for example Gemini network resiliency and the capability to warm swap a compute blade. For more information, see *Cray System Management Workstation (SMW) 5.1 Software Release Overview* (S–2482–51).

1.2 Supported System Configurations

Cray Linux Environment (CLE) release 3.1 supports Cray XE and select Cray XT systems.

The base release (CLE 3.1.UP00) supports only new or initial software installations on the following platforms: Cray XE6, Cray XT6, Cray XT6m, and Cray XT5m systems. The base release also supports upgrade installations from CLE 3.0 on the following platforms: Cray XT6 and Cray XT6m systems.

CLE 3.1 update releases will support additional Cray hardware platforms and migration from CLE 2.2, releases as indicated in Table 1. Installation types in this table are defined as follows:

Initial — A new or fresh software installation involves installing and configuring the entire system and is generally performed for new hardware. If an initial installation is performed on an existing system, the previous configuration is lost.

Update — A software update installation involves applying an update release package for a major release that is already running on your system. For example, installing CLE 3.1.UP02 on a system that is already running an earlier version of CLE 3.1 is considered an update installation.

Upgrade — A software upgrade installation involves moving to the next release of a software package. For example, installing CLE 3.1 on a system that is running CLE 3.0 is considered a software upgrade.

Migration — A CLE migration installation involves moving to a new release level of the CLE software package. It includes an upgrade, plus additional steps to convert the system software to a new level of SuSE Linux. A migration is required when the target release includes a newer level of SLES. For example, upgrading to CLE 3.1 from a system that is running CLE 2.2 requires a migration from SLES 10 SP2 to SLES 11.

Table 1. CLE 3.1 Installation Support by Hardware Platform

Hardware Platform	Installation Type	Target Availability
Cray XE6 (new system)	Initial	June 2010 (base release)
Cray XT6 or Cray XT6m	Initial or Upgrade from CLE 3.0	June 2010 (base release)
Cray XT5m	Initial	June 2010 (base release)
Cray XE6 (XT6 HW upgrade)	Initial or Upgrade from CLE 3.0	September 2010 (update release)
Cray XE6 (XT5 HW upgrade)	Initial	September 2010 (update release)
Cray XE5 (XT5 HW upgrade)	Initial	September 2010 (update release)
Cray XT4, Cray XT5 or Cray XT5m	Migration from CLE 2.2 (UP02 or later)	December 2010 (update release)
Cray XE5 or Cray XE6 (XT5 HW upgrade)	Migration from CLE 2.2 (UP02 or later)	December 2010 (update release)

1.3 Description of the CLE 3.1 Release Software

CLE is a Linux-based operating system that runs on Cray XE and Cray XT systems. The CLE 3.1 release includes Cray's customized version of the SLES 11 operating system. All software is installed by means of scripts and RPM Package Manager (RPM) files. RPMs include related security fixes.

Important: The base CLE 3.1.UP00 release supports initial software installations or upgrades from CLE 3.0. Upgrades from CLE 2.2 to CLE 3.1 will be supported in a CLE 3.1 update release. For more information, see Table 1.

For complete information about the release package, including detailed information about prerequisites for other Cray software products and the supported upgrade path, see Chapter 5, Release Contents on page 67.

1.4 CLE 3.1 Support Policy

Cray continually enhances the Cray Linux Environment (CLE) with new releases and periodically discontinues support for older releases. Our current policy is to support the latest major release of CLE and the previous major release.

Note: Because CLE 3.0 was restricted to a single hardware platform, for support purposes, the previous major release is CLE 2.2.

- During the 12 months after initial release, CLE software is supported with update packages at approximately three- to six-month intervals, depending on need.
- Cray will provide patches for available critical and urgent bug fixes for a period of 18 months following an initial (generally available) CLE release.
- Beyond 18 months, support is limited to critical fixes on a best-effort basis.

All applicable recommended and security-related SUSE Linux updates released by Novell are included in the CLE releases and update packages. Security-related patches are also available through Field Notices (FNs).

Contact your Cray representative for information about current software availability and release schedules.

Software Enhancements [2]

This chapter describes the software enhancements made to the Cray Linux Environment (CLE) since the CLE 2.2 release. This information is provided as a service to users and administrators who are familiar with earlier CLE versions.

For information about issues that you may encounter when using, installing or maintaining CLE 3.1 (when compared to previous CLE releases), see Chapter 3, Compatibilities and Differences on page 41.

In addition to the documentation noted in each feature description, see Cray-developed Books Provided with This Release on page 61.

2.1 Cray XE Systems Supported

CLE 3.1 adds support for Cray XE6 and Cray XE5 systems. Cray XE systems incorporate the new Cray Gemini based system interconnection network, XIO service blades, and Cray X6 compute blades.

Note: The base CLE 3.1.UP00 release supports initial software installations for new Cray XE6 systems. Support for Cray XE5 and Cray XE6 hardware upgrades (Cray XT5 systems upgraded to Cray Gemini) will be included in a CLE 3.1 update release. For more information, see Table 1.

2.1.1 Gemini Based System Interconnection Network

The Cray Gemini application-specific integrated circuit (ASIC) provides an interface between the processors and the interconnection network through the HyperTransport 3.0 technology. Key features of the Gemini ASIC include hardware support for a global address space, improved message rate, off-load capabilities, fast memory access, atomic memory operations, adaptive routing and hardware support for large block transfers.

Each ASIC includes two Network Interface Controllers (NICs), and an embedded interconnection switch (router). Each NIC is an independent, addressable endpoint in the network, therefore a single ASIC supports two nodes.

The Cray Gemini based system interconnection network and associated software provides the following features to support scalable programming. For additional information about these features, see *Using the GNI and DMAPP APIs* (S–2446–31) and *Workload Management and Application Placement for the Cray Linux Environment* (S–2496–31).

- Message passing interface. Support for message passing, one-sided operations, and global address space programming models.
- Gather/scatter performance. A symmetric address translation mode allows access to all nodes in a job without needing to modify the fast memory access (FMA) window. This reduces processor and network overhead on operations involving a small amount of data on a large number of nodes. Network packet overhead is reduced so that network efficiency is high during these operations.
- Flat collectives. Support for atomic memory operations plus efficient scatters allows collectives to be programmed in a vector-like manner to scale much better than typical message-based algorithms.
- Flexible memory mapping. The Memory Relocation Table (MRT) allows software to use a contiguous address space when directly accessing memory allocated by your program.
- I/O performance enhanced by RDMA transfers from I/O adapters to remote memory throughout the system.

The Cray Gemini based system interconnection network provides the following features to support system resiliency. For additional information about network resiliency and adaptive routing, see *Managing System Software for Cray XE and Cray XT Systems* (S–2393–31).

- Network resiliency. Improved handling of network link failures to keep the system operational if a network or network endpoint fails. This includes the capability to warm swap a compute blade.
- Adaptive routing. Gemini ASIC hardware spreads data packets over the four or eight channels which comprise each of the torus links. Adaptive routing may be used for most network data, reducing sensitivity to network hot spots.
- End-to-end data protection. Hardware support is provided so that all packets between the sender and receiver receive a cyclic redundancy check (CRC) to detect data corruption.
- Link-level error correcting code (ECC). Link-level data is resent if an error occurs while data is transiting a link.
- Internal NIC and router protection. The Gemini ASIC uses ECC to protect major memories and data paths within the device.

2.1.2 XIO Service Blades for Cray XE Systems

The Cray XIO service blades contain four service I/O nodes and four PCI Express 2.0 (PCIe) slots. XIO blades support the Gemini based system interconnection network only. XIO blades support three types of I/O devices: Ethernet, Fibre Channel, and InfiniBand. Each node on an XIO blade has a six-core AMD Opteron Series 2000 processor coupled to 16 GB of DDR2 memory (four DDR2 DIMMS per socket), an AMD SR5670 bridge chip, and a PCIe slot. The XIO node has vastly improved I/O throughput compared to the SIO node of a Cray XT system.

XIO boot blades are a special type of XIO blade. XIO boot blades have one node with two PCIe slots (node 1). Of the remaining three nodes on the blade, node 0 has no PCIe I/O connectivity and nodes 2 and 3 have the normal service node configuration of one PCIe slot per node.

Documentation may use one of the following terms to reference SDB and I/O nodes: *service I/O node*, *XIO node* (Cray XE service and I/O node, with Gemini ASICs), or *SIO node* (Cray XT service and I/O node, with SeaStar ASICs).

2.1.3 Software for the Cray Gemini ASIC

The Cray Gemini ASIC provides an address translation mechanism, communication modes, and low-latency synchronization necessary to support the abstraction of a global, shared address space across the entire machine. System libraries available with CLE 3.1 provide low-level communication services to user-space software. *uGNI* directly exposes the communications capabilities of the Cray Gemini ASIC and *DMAPP* implements a logically shared, distributed memory (DM) programming model. The uGNI and DMAPP APIs allow system software to realize as much of the hardware performance of the Cray Gemini ASIC as possible while being reasonably portable to its successors. For more information, see *Using the GNI and DMAPP APIs* (S–2446–31).

Layered on top of uGNI and DMAPP are portable communication libraries, such as MPI and Cray SHMEM, and the partitioned global address space (PGAS) compilers, such as UPC and Co-array Fortran.

Kernel modules, such as the Lustre network driver (LND), communicate via the kernel Gemini Network Interface, *kGNI*.

2.1.4 Performance Analysis for Cray XE Systems

CLE 3.1 includes the ability for CrayPat (Cray performance analysis tool) to access memory mapped registers (MMRs) on the Gemini ASIC that serve as performance counters in either the network interface controller or router tile domains. These MMRs can provide useful performance statistics to application developers when optimizing their programs for use with a Cray XE system.

The Network Interface Controller (NIC) or processor tile MMRs track network transfers within its client node while router tile MMRs increment based on matches with desired bit patterns. The Application Level Placement Scheduler (ALPS) is modified to interface with the kernel module, `gpcd`, to support access to the Gemini ASIC performance counters for Cray Performance Analysis Tools. Reserved access to the Gemini ASIC performance counters is managed both by ALPS and `gpcd` and requested by the application programmer through CrayPat.

There are two virtual channels within the Cray Gemini ASIC. Each router tile in the ASIC has four control registers and five data registers per virtual channel. One control-data register pair per virtual channel can be used as a filtering counter. When a control register is set to active, it will compare a 24-bit pattern with incoming transfers. When all programming control registers match their supplied pattern, the filtering counter will increment.

The `gpcd` kernel module also allows applications to collect MMR data on remote nodes that are not connected to the application; this can provide useful information about performance. However, remote MMR access can potentially inhibit application performance.

For more information, see *Cray Performance Analysis Tools Release Overview and Installation Guide* and *Using Cray Performance Analysis Tools*.

The `apstat -av` command display is modified to show which, if any, Cray Gemini ASIC MMR tiles are reserved in three specific types:

- Processor attached tiles
- Both processor and network attached tiles
- Processor attached tiles and **all** system network attached tiles

`apstat` will also indicate if performance counters are the cause of application placement failure.

For more information, see the `apstat`(1) man page.

2.2 Cray X6 Compute Blades Supported

Cray XE6, Cray XT6, and Cray XT6m systems include Cray X6 compute blades. Support for Cray X6 compute blades is included in the CLE 3.1 release.

Note: For Cray XT6 and Cray XT6m systems, this support was initially provided with the CLE 3.0 release.

Each Cray X6 compute blade includes four compute nodes for high scalability in a small footprint; up to 96 processor cores per blade, or 2,304 processor cores per cabinet. The Cray X6 compute node has two AMD Opteron 6100 Series processors (8-core or 12-core), each coupled with its own memory and a connection to the network ASIC (Gemini or SeaStar). Each Cray X6 compute node is designed to efficiently run up to 24 MPI tasks or a hybrid of MPI and OpenMP parallelism.

The AMD processor's on-chip and highly associative data cache supports aggressive out-of-order execution. The integrated memory controller eliminates the need for a separate Northbridge memory chip and provides a high-bandwidth path to local memory.

Each Cray X6 compute node can be configured with 32 GB or 64 GB DDR3 memory. Memory on Cray X6 compute nodes is registered and memory controllers provide the additional protection of x4 device correction, ensuring highly reliable memory performance.

2.3 SUSE Linux Enterprise Server (SLES) 11 Upgrade

The CLE 3.1 release is based on the SUSE Linux Enterprise 11 (SLES 11) version of the Linux operating system and a Linux 2.6.16.27 kernel. Cray-specific kernel features, the user-level environment, software subsystems, and the installation/build system are now based on SLES 11.

The update to SLES 11 aligns with newer versions of the SLES operating system to keep pace with the SLES product life-cycle and to ensure that CLE releases include relevant bug fixes and current software. SLES 11 incorporates a significant number of enhancements when compared to the previous two CLE releases, which were based on SLES 10, Service pack 1. Additionally, some new features and hardware supported by the CLE 3.1 release require specific functionality that is available with SLES 11.

2.3.1 OpenFabrics InfiniBand Upgrade

OpenFabrics Enterprise Distribution (OFED) is now part of the SLES 11 release. As part of the upgrade to SLES 11, OFED is upgraded to version 1.4. OFED is deployed on service I/O nodes to provide InfiniBand connectivity to the Cray system. InfiniBand is the highest-bandwidth, lowest-latency commodity interconnect available today.

For additional information about OFED, see *Installing and Configuring Cray Linux Environment (CLE) Software* and *Managing System Software for Cray XE and Cray XT Systems*.

2.3.2 SLES 11 Documentation

For information about the contents of SLES 11 and Linux in general, refer to the following third-party and open-source websites:

- SLES 11 Documentation — See www.novell.com/linux
- The Linux Documentation Project — See www.tldp.org

Updated Linux man pages are included with the CLE 3.1 release. For complete information regarding changes to specific commands due to the upgrade to SLES 11, see the associated man pages. To access Linux man pages, use the `man` command on a login node.

2.3.3 Differences and Limitations Introduced with SLES 11

For information regarding specific changes associated with the upgrade to SLES 11, see the documentation described in the previous section. Some changes likely to impact users and administrators of Cray systems are described in Chapter 3, Compatibilities and Differences on page 41 as follows:

`make` Utility Enhancements and Differences on page 43

`/dev/sd*` Device Ordering is Not Guaranteed on page 49

Lustre File System Configuration Requires Persistent Device Names on page 54

`scsidev` is Obsolete on page 55

2.4 Configuration and Packaging Enhancements

Cray releases CLE software by using RPM Package Manager (RPM) files. The CLE 3.1 release includes a number of changes to the location of Cray-specific packages and associated configuration files.

The naming conventions for many Cray-specific RPMs now conform to Linux Assigned Names and Numbers Authority (LANANA) standards for packages to ensure that Cray produced packages do not conflict with SLES RPMs. Most Cray-specific RPMs have a name that begins with `cray-`.

Changes were made to file and package locations to comply with the Filesystem Hierarchy Standard (FHS) for UNIX file and directory placement. Most Cray-specific software applications have moved to `/opt/cray` and associated Cray-specific configuration files have moved to `/etc/opt/cray`.

Cray implemented a new build service, openSUSE Build Service (OBS), for CLE software to improve internal Cray development processes; OBS does not directly impact users or administrators. However, because monolithic packages have been broken up into smaller pieces, the process for creating and releasing patches is simplified; critical patches can be generated for individual components.

Users and system administrators who are familiar with older CLE release packages need to be aware of the re-factored packages and package locations. For a summary of new package locations and a partial list of files that have moved, see Appendix A, Configuration and Packaging Changes on page 75.

Cray-specific man pages have been updated to reflect changes in file locations. Source for Cray-specific man pages is included in the associated RPM and is installed in `/opt/cray/share/man`.

2.5 Lustre 1.8 File System Upgrade

The CLE 3.1 release includes support for the Lustre 1.8 release from Oracle. Lustre 1.8 includes several new features that will be useful to end-users, site analysts, and administrators, including:

- Adaptive Time-outs
- Version-Based Recovery (VBR)
- Object Storage Target (OST) Pools
- Object Storage Server (OSS) Read Cache

Another feature, client interoperability, enables Lustre 1.8 clients to send requests to a Lustre server running a later release of Lustre (when available). CLE 3.1 also includes a Cray specific feature, imperative recovery, that potentially allows file systems to recover faster. For a complete description, see Lustre Imperative Recovery on page 21. For additional information about Lustre, see the *Lustre Operations Manual* and *Managing Lustre for the Cray Linux Environment (CLE)*.

2.5.1 Adaptive Time-outs

Lustre servers track remote procedure call (RPC) transaction completion times and clients use those times for future RPC transaction time-out values. Servers also send early replies to clients if queued RPCs approach their time-outs. These features allows clients to avoid unnecessarily retrying RPCs. Adaptive time-outs are active by default in CLE 3.1; you can disable adaptive time-outs by invoking `lctl set_param at_max=0` on each client.

2.5.2 Version-based Recovery (VBR)

In VBR, data objects are given versions in order to track and detect conflicting transactions during recovery. If the expected version is associated with the object, the transaction is replayed during recovery. If there is a version mismatch, the client is no longer allowed to participate in data operations on the object in question and the client is evicted. This allows for a flexible, out-of-order recovery of clients. In VBR, recovery continues even if multiple clients do not reconnect.

2.5.3 OST Pools

OST pools are used to group a subset of OSTs together for easier object placement and manipulation. Administrators may want to group data objects together because of the relative performance of the underlying technology. Furthermore, users may want application objects to execute on certain types of targets for performance reasons. Striping policies may be assigned to OST pools using the `lfs` command.

Note: CLE 2.2 clients, using Lustre 1.6.5, are not compatible with Lustre 1.8 data objects that are assigned to an OST pool.

2.5.4 OSS Read Cache

Lustre 1.8 uses the Linux page cache to provide read-only caching of data on object storage servers (OSS). This strategy reduces disk access time caused by repeated reads from an OST. OSS read cache is enabled by default, but you can disable it by setting `/proc` parameters. For example, invoke the following on the OSS:

```
# lctl set_param obdfilter.*.read_cache_enable 0
# lctl set_param obdfilter.*.writethrough_cache_enable 0
```

2.6 Lustre Imperative Recovery

(CLE 3.0 – 3.1 Change) Lustre imperative recovery is a Cray-specific Lustre feature provided with CLE that works automatically in the background to facilitate faster recovery in the event of a Lustre failover. During Lustre failover and recovery, imperative recovery utilities (if enabled) notify Lustre clients to switch server connections immediately rather than waiting for connections to the primary service to time out. The recovery window can be tuned to limit the amount of time servers wait for clients that do not reconnect.

This functionality is disabled by default but the system administrator may enable it by setting the `ENABLE_IMP_REC` parameter in *mylustre*`.fs_defs` to `yes`. The tunable recovery window is configured by setting `RECOVERY_TIME_HARD` and `RECOVERY_TIME_SOFT`, also in *mylustre*`.fs_defs`. A new utility, `xtlusfoevntsndr`, is provided to notify clients that the secondary services are available. This command is generally invoked automatically by the `xt-lustre-proxy` daemon. For additional information, see *Managing Lustre for the Cray Linux Environment (CLE)* and the `xtlusfoevntsndr`(8), `xt-lustre-proxy`(8), and `lustre.fs_defs`(5) man pages.

2.7 Dynamic Shared Objects and Libraries (DSL)

Users can link and load dynamic shared objects in their applications by using the compute node runtime environment in the Cray Linux Environment (CLE). CLE includes software that enables linking and loading with dynamic libraries, using an alternate to the `initramfs` file system on the compute nodes, called the compute node root. The compute node root is essentially the read-only DVS-projected shared root file system.

The main benefit of this feature is expanded use of programs and libraries that require shared objects on Linux systems. Users are able to effectively reduce memory and binary footprint when shared objects, called multiple times, use the same segment of memory address space. Users can create applications that no longer need relinking or recompiling when the system software is upgraded.

Administrators enable this option at installation time by modifying parameters in `CLEinstall.conf`.

The default compute node root is the Cray shared root file system. Administrators can change this by editing `/etc/opt/cray/cnrte/roots.conf`. Symbolic names in this file define the available roots. The user may then choose the appropriate compute node root based on the requirements of their application by setting the environment variable `CRAY_ROOTFS` to a symbolic name that is defined in `roots.conf` (only needed if different from the default). For example, to revert to using the `initramfs` compute node root, set `CRAY_ROOTFS` to `INITRAMFS`.

Both the C/C++ and Fortran compilers use the `dynamic` option when invoking the compiler:

```
% cc -dynamic sample.c -o sample
```

```
% ftn -dynamic samplef.f -o samplef
```

Please see the `cc` and `ftn` man pages for further information.

The compilers also accept the `shared` option to create a shared object file; for example, *libname*`.so`.

When setting up the dynamic libraries feature, it is important to remember the following constraints, as they may affect system administrators or end-users:

- You must have available service or compute nodes to act as internal DVS servers. These servers should not be Lustre servers because the load will likely degrade the performance of that node if both DVS and Lustre are active. Compute nodes used as internal DVS servers will no longer be available to the compute node pool.
- You must recompile programs that were statically compiled (usually with archive files) to include dynamic libraries. Afterward, when you use dynamic libraries you no longer need to recompile when a dynamic library is updated.
- Support is limited to providing an environment for dynamic linking and loading with user applications and ISV applications. However, Cray does not support or warrant third-party products.

For additional information, see *Managing System Software for Cray XE and Cray XT Systems* (S–2393–31) and *Workload Management and Application Placement for the Cray Linux Environment* (S-2496-31).

2.8 Cluster Compatibility Mode (CCM)

(CLE 3.0 – 3.1 Change) Cluster Compatibility Mode (CCM) provides the services needed to run most cluster-based independent software vendor (ISVs) applications "out of the box".

A Cray XE or Cray XT system is not a cluster but a massive parallel processing (MPP) computer. An MPP is simply one computer with many networked processors used for distributed computation, and, in the case of Cray XT and Cray XE architectures, a high-speed communications network that facilitates optimal bandwidth and memory operations between those processors. When operating as an MPP machine, the Cray compute node kernel (Cray CNL) typically does not have a full set of the Linux services available that are used in cluster ISV applications.

CCM is tightly coupled to the workload management system. It enables users to execute cluster applications alongside workload-managed jobs running in a traditional MPP batch or interactive queue. Support for dynamic shared objects and expanded services on compute nodes, using the compute node root runtime environment (CNRTE), provide the services to compute nodes within the cluster queue. Essentially, CCM uses the batch system to logically designate part of the Cray system as an emulated cluster for the duration of the job.

CCM requires that the administrator install both CNRTE and Realm-Specific Internet Protocol (RSIP). For procedures to configure these features, see *Installing and Configuring Cray Linux Environment (CLE) Software*. When you set the following parameters in the `CLEinstall.conf` file, the `CLEinstall` program automatically configures your system for CCM.

`CCM=`**`yes`** Set this parameter to `yes` to enable Cluster Compatibility Mode and install the appropriate RPMs.

`CCM_ENABLERSH=`**`yes`**

Enables services or daemons that most ISV applications need to run. Examples of these services are `xinetd`, `portmap`,`rsh`, and `rlogin`. If you set `CCM_ENABLERSH` to `no` some ISV applications will not work. If you don't specify this parameter in `CLEinstall.conf`, `rsh` is enabled by default.

`CCM_QUEUES="`***`ccm_queue1, ccm_queue2`***`"`

Specifies one or more batch queues used in the workload management system. If this parameter is not specified, the default value is `ccm_queue`.

`CCM_WLM=`***`pbs`***

Set the value to either `pbs` or `torque` to choose your preferred workload management software.

You may also set these parameters after installation by editing `/etc/opt/cray/ccm/ccm.conf`. This file also contains options that are effective only after the installation.

`CCM_ENABLENIS=`**`no`**

Set this option to `yes` to enable NIS and start `ypservices` on the compute node. NIS is optional and is disabled by default. If not properly configured, the calls will time out to the network, significantly slowing down CCM startup.

`CCM_DEBUG=`**`no`**

Set this option to `yes` to turn on CCM debugging. Log files for CCM sessions are generated in `/var/log/crayccm`. Consider using facilities like `logrotate`(8) to rotate debug logs according to site policy.

Users must load the `cray/ccm` module on the login node before launching a job. They then provision the desired resources by using their workload management system, for example, PBS or Moab, or TORQUE. The user allocates these nodes to a cluster batch queue. A new command, `ccmrun`, starts the cluster application, propagates the application through the requested nodes, and then terminates and cleans up the application on exit. Compute nodes reserved for the cluster application are returned to the MPP-architecture pool when the job terminates. Users can interact with a cluster job by using the `ccmlogin` command to log in to the head node of the cluster job where processing element or rank 0 resides. This is implemented using SSH and will accept SSH arguments. For more information, see the `ccmrun`(1) and `ccmlogin`(1) man pages.

Users are encouraged to install cluster programs on their own shared storage. However, if a program requires root access or if administrators want to install a suite of applications for a site, it may be necessary to install programs on the shared root in `xtopview.`

For more information about CCM, see *Workload Management and Application Placement for the Cray Linux Environment*.

2.8.1 Limitations for Using CCM

The following limitations apply to supporting cluster queues with CLE 3.1 on Cray systems:

- Applications must fit in the physical node memory because swap space is not presently supported in CCM.
- Core specialization is not supported in CCM.
- CCM does not include support for applications built in Cray Compiling Environment (CCE) with Co-Array Fortran (CAF) or Unified Parallel C (UPC) compiling options, nor any Cray built libraries built with these implementations. Applications built using the Cray SHMEM library are also not compatible with CCM.

2.9 Core Specialization

(CLE 3.0 – 3.1 Change) Some applications may experience performance degradation as a result of overhead interrupts or noise. These interrupts can have an aggregate effect upon synchronizations within the application. Testing indicates that low-frequency, high-duration noise causes the greatest performance degradation. Conversely; application performance is less affected by high-frequency, low-duration noise.

To address this issue, CLE 3.1 offers new core-specialization functionality. Core specialization binds a set of Linux kernel-space processes and daemons to a single core within a Cray compute node to enable the software application to fully utilize the remaining cores within its `cpuset`.

Users can specify the number of specialized cores by selecting the `-r` and `-B` options of the `aprun` command. The `-B` option passes batch options that correspond with `-n`, `-N`, `-d`, and `-m` to the `aprun` command. Table 2 shows representative values for core specialization scenarios on Cray systems.

The `apstat` command output now includes specialized cores as in the following example:

```
$ apstat -n
   NID Arch State HW Rv Pl  PgSz     Avl    Conf  Placed  PEs Apids
...
    84   XT UP  B  8  8  7+   4K 4096000 4096000 4096000    8 1577851
    85   XT UP  B  8  2  1+   4K 4096000 4096000 4096000    8 1577851
    86   XT UP  B  8  8  8    4K 4096000 4096000 4096000    8 1577854
...
```

For `Apid` 1577851, the + signs indicate that one core on each node is specialized.

A new command, `apcount`, was introduced to calculate the batch reservation width required when you use the `aprun -r` option.

This functionality is documented in the following man pages: `apstat`(1), `aprun`(1), and `apcount`(1). For additional information, see *Workload Management and Application Placement for the Cray Linux Environment*.

Table 2. Core/PE Distribution for `r=1`

Compute Blade Type	# of Cores	Service Affinity Cores	Compute Cores	N_{MAX}
Cray XT5 or XE5	8	7	0-6	7
Cray XT5 or XE5	12	11	0-10	11
Cray XT6 or XE6	16	15	0-14	15
Cray XT6 or XE6	24	23	0-22	23

2.9.1 Limitations for Using Core Specialization

Core specialization will not work with jobs launched in Cluster Compatibility Mode.

With CLE 3.1, only `r=1` is supported.

2.10 Node Health Checker (NHC) Enhancements

NHC (sometimes referred to as *NodeKARE*) is automatically invoked by the Application Level Placement Scheduler (ALPS) when an application terminates. ALPS passes a list of compute nodes associated with the terminated application to NHC. NHC performs tests, which are specified in the NHC configuration file, to determine whether compute nodes allocated to the application are healthy enough to support running subsequent applications.

For an overview of NHC, see the `intro_NHC`(8) man page. For detailed information about NHC, see *Managing System Software for Cray XE and Cray XT Systems*.

This CLE release provides the following NHC enhancements.

2.10.1 Suspect Mode Returns Healthy Nodes to Resource Pool Faster

NHC is now more efficient in Suspect Mode, so NHC finds healthy nodes and returns them to the system resource pool faster than in previous releases. This means a higher overall availability of nodes.

- Every five minutes, NHC searches for nodes that have become healthy, compared with every fifteen minutes in the CLE 2.2 version (the value is specified through the Suspect Mode `suspectfreq:` variable).
- NHC marks unhealthy nodes as `admindown` 85 minutes sooner than in the CLE 2.2 version, because of a reduction in the time-out value of the `Filesystem` test. Although the `Filesystem` test was deferred in the CLE 2.2 version, its long time-out value (80 minutes) required the `suspectend` variable to be set at 2 hours. The `Filesystem` test time-out value has been reduced from 80 minutes to 30 minutes. This reduction allowed NHC to reduce its overall time-out value from 2 hours to 35 minutes (the value is specified through the Suspect Mode `suspectend:` variable).
- In Suspect Mode, NHC now continually restarts tests that fail. (This is indicated by the new *restart_delay* variable used with each NHC test.)
- Previously, Suspect Mode was a separate binary that was called by NHC; the NHC software has incorporated the Suspect Mode software. As a result of this enhancement, the behavior between Normal Mode and Suspect Mode is now synchronous, allowing the previously deferred NHC tests to be supported (see Enhancements to NHC Tests on page 28) and the `xtok2` binary to be removed.

2.10.2 NHC Verifies ALPS Acknowledgement of a Node's State Change

NHC now verifies that ALPS acknowledges a change that NHC has made to a node's state. If ALPS does not acknowledge a change, then NHC recognizes this disagreement between itself and ALPS. NHC will then change the node's state to `admindown` and then exit.

2.10.3 Automatic Recovery If a Login Node Crashes While `xtcheckhealth` Monitoring Nodes

If a login node crashes while some `xtcheckhealth` binaries on that login node are monitoring compute nodes that are in `suspect` state, those `xtcheckhealth` binaries will die when the login node crashes. When the login node that crashed is rebooted, an automatic recovery action takes place, and manual intervention by the system administrator is no longer required.

For additional information, see the `intro_NHC`(8) man page that is provided with the CLE 3.1 release package.

(CLE 3.0 – 3.1 Change) NHC automatically recovers the nodes in `suspect` state when the crashed login node is rebooted because the recovery feature runs on the rebooted login node. If the crashed login node is not rebooted, then manual intervention is required to rescue the nodes from `suspect` state. This manual recovery can commence as soon as the login node has crashed. To recover from a login node crash when a login node will not be rebooted, the `nhc_recovery` binary can help you release the compute nodes owned by the crashed login node. For additional information, see the new `nhc_recovery`(8) man page and *Managing System Software for Cray XE and Cray XT Systems*.

2.10.4 `xtcheckhealth` Logs Dumped by `xtdumpsys`

(CLE 3.0 – 3.1 Change) The `dumpsys-plugin` file now dumps `xtcheckhealth` logs from login nodes and saves them in the `login-node-logs.tar.gz` file in the dump directory `/opt/craydump/`*`YYMMDDhhmm`*`/`.

2.10.5 Enhancements to NHC Tests

- NHC can now run all NHC tests in both Normal Mode and Suspect Mode. However, a test's action determines whether it is actually run in Suspect Mode. Tests that have an action of `Log` do **not** run in Suspect Mode. In addition, if you use plugin scripts with an action of `Log`, the script will only be run once, in Normal Mode, which makes log collecting and various other maintenance tasks easier to code.
- Guidance information about using NHC tests is provided in *Managing System Software for Cray XE and Cray XT Systems*.
- The `Application Exited Check` test and the `Apinit Ping` test running in Suspect Mode are faster and more reliable than in previous versions. The `Application Exited Check` test was previously named the `Application` test, the `Apinit Ping` test was previously named the `Alps` test.
- The following (previously deferred) NHC tests are now supported with this release:
  - `Free Memory Check`, examines how much memory is consumed on a compute node while applications are not running. The `Free Memory Check` test is enabled in the default NHC configuration file that Cray provides. The `Free Memory Check` test was previously named the `Memory` test. (The previously deferred `Root File System` test functionality was incorporated into this test.)
  - `Filesystem`, ensures that the compute node is able to perform simple I/O to the specified file system. The `Filesystem` test is enabled in the default NHC configuration file.

– `Plugin`, enables scripts and executables not built into node health to be run, provided they are accessible on the compute node. To enable the use of local configuration settings, no plugins are configured by default. `Plugin` was previously named the `Site` test.

Note: For backward compatibility the `Site` name is also recognized in the NHC configuration file. Refer to the following example:

```
Site: Log 600 2400 300 0 0 /scratch/QuickIMB.sh "/tmp 100 1"

Plugin: Log 600 2400 300 0 0 /scratch/QuickIMB.sh "/tmp 100 1"
```

The technical note *Writing a Node Health Checker (NHC) Plugin Test* is provided with the CLE 3.1 release.

(CLE 3.0 – 3.1 Change) Node health now prints out a summary message at the end of Normal Mode and Suspect Mode when at least one test has failed on a node. For example:

```
<node_health:3.1> APID:100 (xtnhc) FAILURES: The following tests have failed in normal mode:
<node_health:3.1> APID:100 (xtnhc) FAILURES: (Admindown) Apinit_Ping
<node_health:3.1> APID:100 (xtnhc) FAILURES: (Admindown) Plugin /example/plugin
<node_health:3.1> APID:100 (xtnhc) FAILURES: (Log Only ) Filesystem_Test on /mydir
<node_health:3.1> APID:100 (xtnhc) FAILURES: (Admindown) Free_Memory_Check
<node_health:3.1> APID:100 (xtnhc) FAILURES: End of list of 5 failed test(s)
```

2.10.6 New and Changed NHC Variables

- The new `nhcon` global variable turns NHC on or off. In the default NHC configuration file that Cray provides, NHC is `on`.
- Because NHC in Suspect Mode now continually restarts tests that fail, the new *restart_delay* argument was added to each NHC test. It specifies how long NHC will wait, in seconds, to restart the test after the test fails. The *restart_delay* argument is valid only when NHC is running in Suspect Mode.
- The Suspect Mode `suspectend:` variable is now set to `2100` seconds in the default NHC configuration file. Previously, it was set to `7200` seconds.
- The Suspect Mode `recheckfreq:` variable is now set to `300` seconds in the default NHC configuration file. Previously, it was set to `900` seconds.
- The *iterationlimit* variable was removed from the NHC configuration file because it was not used. The purpose of the *iterationlimit* variable was to set the limit for how many times NHC could be called by ALPS for the same job; however, NHC ignores ALPS if it calls NHC more than once per job.

2.11 Cray Data Virtualization Service (Cray DVS) Enhancements

CLE 3.1 includes the following DVS enhancements. For more information about DVS on a Cray system, see *Introduction to Cray Data Virtualization Service* and the `dvs`(5) man page.

2.11.1 Stripe Parallel Mode for Cray DVS

Cray DVS is an I/O forwarding service that can parallelize the I/O transactions of an underlying POSIX-compliant file system. In the previous release, DVS supported serial and cluster parallel modes. Serial mode provides access to the underlying file system through one DVS server. In cluster parallel mode, multiple DVS servers are active and each is responsible for a particular segment of the address space of the underlying file system. Based on the file inode number, DVS automatically determines which server a client will forward its request to.

Stripe parallel mode introduces a block-level granularity to the I/O forwarding in DVS. Based on the DVS block size and file offset, DVS automatically determines which server the client transaction is relayed to.

Stripe parallel mode provides the opportunity for greater aggregate I/O bandwidth when forwarding I/O from a coherent cluster file system. IBM GPFS and PanFS have been tested extensively using this mode. NFS cannot be used in stripe parallel mode because NFS implements close-to-open cache consistency; striping data across the NFS clients could result in data integrity issues.

As with other DVS modes, the system administrator mounts the underlying file system by creating an `/etc/fstab` entry on compute node clients and updating the boot images. The administrator manually lists the DVS servers in the `fstab` entry or enters a `nodefile` in place of the manual list. The `maxnodes` option in the `fstab` entry specifies the difference between cluster parallel and stripe parallel. In cluster parallel mode, `maxnodes=1`. In stripe parallel mode, `maxnodes` is either not included or is greater than one, specifying that DVS will stripe across more than one DVS server. For example, the following `fstab` entry configures DVS stripe parallel mode for the */user* file system:

```
/user /user dvs path=/user,nodename=c0-0c2s2n0:c0-0c2s2n1:c0-0c2s2n2,clusterfs
```

Note: DVS no longer requires a `node-map` file (`/etc/dvs/node-map`) for mapping DVS node ids to node ordinals as this is automatically determined internally.

2.11.2 Failover and Failback for Cray DVS Stripe Parallel and Cluster Parallel Modes

(CLE 3.0 – 3.1 Change) DVS now supports failover and failback for loadbalance mode as well as cluster and stripe parallel modes by adding a `failover` option. If a DVS server in the `/etc/fstab` list fails, client requests will automatically failover to the next available server in the list. Once the DVS server is rebooted, client requests will automatically failback to include it in future I/O operations. Failover is on by default; specify `nofailover` to disable it.

This feature provides greater resiliency for DVS file system projection within CLE. In previous CLE releases, DVS supported failover for loadbalance mode only, and did not support failback for any modes.

The following entries in `/etc/fstab` indicate cluster parallel and stripe parallel projection modes. Cluster parallel mode is indicated with a `maxnodes=1` option:

```
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0,maxnodes=1,failover
/dvs-shared /dvs dvs path=/dvs,nodename=c0-0c2s1n0:c0-0c2s1n3:c0-0c2s2n0,failover
```

2.11.3 Collecting Statistics for Cray DVS

(CLE 3.0 – 3.1 Change) DVS statistics are available for both client and server nodes in CLE. A count of file system operations are available via the `/proc/fs/dvs/stats` file. Each line of this file displays a file system operation and a count of successful and failed operations of that type. The `/proc/fs/dvs/stats` file is used for file system operations that can not be correlated to a specific DVS mount point, and thus is most useful for DVS servers. Per-mountpoint statistic files are available on compute nodes via `/proc/fs/dvs/mounts/*/stats`. In addition, DVS IPC statistics are available on all nodes via `/proc/fs/dvs/ipc/stats`.

2.12 Application Level Placement Scheduler (ALPS) Enhancements

2.12.1 ALPS Exclusive Access

A new `-F` option is available for `aprun` to provide a program with exclusive access to all the processing and memory resources on a node. This option was initially introduced with the CLE 2.2.UP01 update package.

This option assigns all compute node cores and compute node memory to the application's `cpuset`. Using it together with the `-cc` option allows an application programmer to bind processes to those mentioned in the affinity string.

There are two modes: `exclusive` and `share`. The `share` mode restricts the application specific `cpuset` contents to only the application reserved cores and memory on Non-Uniform Memory Access (NUMA) node boundaries. For example, if an application requests and is assigned cores and memory on NUMA node `0`, then only NUMA node `0` cores and memory are contained within the application `cpuset`. The application will not have access to the cores and memory on other NUMA nodes on that compute node.

Administrators can modify `/etc/alps.conf` to set a policy for access modes. If `nodeShare` is not specified in this file, the default remains `exclusive`; setting to `share` makes the default share access mode. Users can override the system-wide policy by specifying `aprun -F` at the command line or within their respective batch scripts. For additional information, see the `aprun`(1) man page.

2.12.2 Batch Application Scheduler Interface Layer (BASIL) Protocol Updated

BASIL provides an XML-based interface to ALPS. BASIL is used primarily by batch systems to monitor and manage compute node resource allocations. The BASIL protocol has been updated to provide new capabilities:

- Node exclusive allocation support in ALPS
- Cray X6 processor support
- Resource list as part of response to ALPS reservation method
- Cray Gemini network ASIC support
- Performance counters support
- Core specialization support

In addition, the `admin_cookie` and `alloc_cookie` attributes have been deprecated; they were required for Cray XT systems with the Catamount compute node OS and Catamount is no longer supported. A new attribute, `pagg_id`, takes the place of `admin_cookie` and is used to communicate the process aggregate ID.

BASIL changes have no direct dependencies outside of ALPS other than adoption by third party batch vendors and integration into their products. All applicable batch vendors have been provided the BASIL updates.

For additional information, see *Managing System Software for Cray XE and Cray XT Systems* and the `basil`(7) man page.

2.12.3 Comprehensive System Accounting and ALPS Interface

Comprehensive System Accounting (CSA) now includes an interface with the ALPS application management systems so that application accounting records that include application start, termination, and placement information can be entered into the system accounting database.

You can view this information using the `csacom -A` command. For more information, see the `intro_csa`(8) man page.

2.12.4 ALPS Interface to Cray Management Services (CMS)

Deferred Implementation: This functionality will be implemented in a future release.

(CLE 3.0 – 3.1 Change) ALPS reservation and claim information is forwarded to the Cray Management Services (CMS) state daemon on the SMW. Forwarding this information from the SDB node to the SMW allows a persistent store of the data and enables system administrators to search the history of jobs, error messages and the job utilization on the system.

ALPS notifies CMS of application create or start and destroy or stop. The system state cache daemon also stores the information associated with a reservation, including reservation ID, start and end time of the reservation, execution host name, batch job identification, and user information. Multiple node assignments could be made during the lifetime of a reservation. The node assignment information, such as the start time and end time of the assignment, the name of the application, and exit status, is also stored.

The new `/etc/alps.conf` configuration file parameter `cms` indicates whether or not ALPS uses CMS to store reservation and claim information. Valid settings are `no` and `yes`; the default setting is `no`.

For additional information, see *Using Cray Management Services (CMS)* and *Managing System Software for Cray XE and Cray XT Systems*.

2.12.5 New `aprun -B` Option

(CLE 3.0 – 3.1 Change) The `aprun-B` option passes options specified with the PBS or Moab batch reservation. When the user specifies this option, there is no need to specify `-n`, `-N`, `-d`, and `-m` with the `aprun` command.

2.13 Out of Memory (OOM) Killer

To help avoid out of memory (OOM) errors that lead to a compute node becoming unresponsive, the OOM killer has been extended to terminate all of the processes on a compute node that belong to a specific job (called a *cpuset* or *cgroup*).

The OOM killer radically frees memory by terminating processes in the situation where no progress can be made due to memory not being available. The OOM kill enhancement allows not only one process from a job to be terminated, but all of the processes on a compute node belonging to a specific job. This halts the majority of allocations on a node, greatly reducing the odds of a compute node becoming unresponsive.

With this release, the extended OOM killer feature is enabled by default. System administrators can disable the feature by setting system control `sysctl oom_appkill` to `0`; invoke the following on a compute node:

```
# /sbin/sysctl -w vm.oom_appkill=0
```

Or

```
# echo 0 > /proc/sys/vm/oom_appkill
```

Default behavior can be restored, that is, OOM killer enabled by setting system control `sysctl oom_appkill` to `1`; invoke the following on a compute node:

```
# /sbin/sysctl -w vm.oom_appkill=1
```

Or

```
# echo 1 > /proc/sys/vm/oom_appkill
```

When enabled, `oom_appkill` can be controlled on a per-process basis when a programmer determines that, for a given application, certain processes in the `cpuset` should be exempt from the `oom_appkill` logic. To make a process exempt, use the following command:

```
echo 1 > /proc/self/oom_appkill_exempt
```

2.14 XPMEM Kernel Module

(CLE 3.0 – 3.1 Change)

XPMEM is a low-level kernel driver that handles on-node shared memory usage. When XPMEM is enabled, calls to shared memory on a node stay on the node. When used by parallel programming libraries, XPMEM enables processes to share portions of their memory address space with other local processes. Once these segments of memory are "exported," a different process can "attach" and then reference the exported addresses as if they are in its native address space. XPMEM can enhance performance of local memory sharing; without XPMEM, calls to shared memory on a node are routed to the network and back.

Because XPMEM is a standalone driver, it does not present a dependency risk with other Cray products.

Cray SHMEM (Cray Message Passing Toolkit 5.0 release) uses XPMEM. For more information, see the `intro_shmem`(3) man page. The Cray Compiling Environment (CCE) also uses XPMEM for applications that use UPC extensions or Fortran coarrays. For more information about UPC and Fortran with coarrays, see *Cray C and C++ Reference Manual* and *Cray Fortran Reference Manual*.

2.15 System Resiliency Enhancements

2.15.1 Support for Multipathing I/O

(CLE 3.0 – 3.1 Change) Multipathing functionality enables you to configure multiple I/O paths to the controllers on a disk array. One path is designated as the active primary path and the remaining paths are considered inactive or alternative paths. When the primary path to the array is lost due to a failure, disk-specific multipathing functionality automatically switches the data access to an alternative path.

For Data Direct Networks (DDN) devices, multipathing functionality is provided using Device Mapper (DM) functionality that is included in the Linux kernel. CLE 3.1 supports DM multipathing I/O only on DDN 9900 arrays.

The LSI Redundant Disk Array Controller (RDAC) provides multipathing functionality for LSI devices. LSI RDAC is a self-contained module that operates as a device driver. This module has no external interfaces; it interacts directly with Linux kernel I/O functionality. For Cray systems, the LSI RDAC driver module must be integrated into the OS boot image so that the RDAC module is loaded before the Fibre Channel Driver is loaded. You must configure system boot scripts to recognize service nodes with LSI connections and load the RDAC and Qlogic driver modules in the correct order. For more information, contact your Cray service representative.

During system initialization and startup, DM multipathing or RDAC automatically scans the available I/O paths and sets the primary and alternate; no additional configuration or manual intervention is required.

Failover is automatic. If the current active path fails, the alternate paths are reviewed and I/O processing moves to one of the alternate paths. Failback is also automatic. DM or RDAC maintains heartbeat functionality on the primary connection path. When the primary connection is recognized as restored, data access returns to the primary path's controller.

Multipathing functionality does not negatively impact to I/O performance during normal system operations. In the event of a failure, I/O completion may be slightly delayed, but there is no impact to data access.

2.15.2 SDB Node Failover

(CLE 3.0 – 3.1 Change) The SDB node is integral to the operation of a Cray system. Critical services like ALPS and Lustre rely on the SDB and will fail if the SDB node is unavailable. The CLE release provides functionality to create a standby node that will automatically act as a backup SDB in the event of primary node failure. SDB node failover functionality does not provide automatic failback.

The CLE implementation of SDB node failover includes several installation configuration parameters that facilitate automatic configuration, a `chkconfig` service called `sdbfailover`, and a `sdbfailover.conf` configuration file for defining site-specific commands to invoke on the backup SDB node.

The backup SDB node uses `/etc` files that are class or node specialized for the primary SDB node and not for the backup node itself; the `/etc` files for the backup node will be identical to those that existed on the primary SDB node.

The following list summarizes requirements to implement SDB node failover on your Cray system.

- Designate a service node to be the alternate or backup SDB node. The backup SDB node requires a QLogic Host Bus Adapter (HBA) card to communicate with the RAID. This backup node is dedicated and cannot be used for other service I/O functions.
- Enable the STONITH capability on the blade or module of the primary SDB node in order to use the SDB node failover feature.
- STONITH is a per blade setting and not a per node setting. Ensure that your primary and backup SDB nodes are located on separate blades from services with conflicting STONITH requirements, such as Lustre.
- Enable SDB node failover by setting the `sdbnode_failover` parameter to **`yes`** in the `CLEinstall.conf` file prior to running the `CLEinstall` program.

When this parameter is used to configure SDB node failover, the `CLEinstall` program verifies and turns on `chkconfig` services and associated configuration files for `sdbfailover` and provides command hints for additional manual steps.

- Specify the primary and backup SDB nodes in the boot configuration. For more information, see the `xtcli`(8) man page.
- **(Optional)** Populate `/etc/opt/cray/sdb/sdbfailover.conf` with site-specific commands.

  When a failover occurs, the backup SDB node invokes all commands listed in the `/etc/opt/cray/sdb/sdbfailover.conf` file. Include commands in this file that are normally invoked during system start-up via boot automation scripts. In a SDB node failover situation, these commands must be invoked on the new (backup) SDB node. For example, you may include commands to start batch system software (if not started via `chkconfig`) or commands to add a route to an external license server.

For additional information and procedures to configure SDB node failover, see *Installing and Configuring Cray Linux Environment (CLE) Software* or *Managing System Software for Cray XE and Cray XT Systems*.

2.15.3 Compute Node Failover Manager

(CLE 3.0 – 3.1 Change) The new compute node failover manager daemon (`cnfomd`) facilitates communication from the compute nodes to the backup boot or SDB node in the event of a primary boot or SDB node failure. When a node failed event from the primary boot or SDB node is detected, `cnfomd` updates the `arp` cache entries for the boot or SDB node virtual IP address to point to the backup node. The daemon runs on the compute nodes and is similar to the failover manager daemon (`fomd`) on the service nodes. If both boot and SDB node failover are disabled, the `cnfomd` process exits immediately after start up.

2.15.4 Cray DVS Failover

DVS enhancements in CLE 3.1 provide greater resiliency for DVS file system projection within CLE. For more information, see Failover and Failback for Cray DVS Stripe Parallel and Cluster Parallel Modes on page 31.

2.16 System Administration Enhancements

In addition to the new functionality described in the following sections, administrators should be aware of the administrative compatibilities and differences described in Installation and Configuration Changed Functionality for System Administrators on page 47 and General System Administration Differences on page 52.

2.16.1 New `CLEinstall` Options and Configuration Parameters

The CLE installation program provides several new configuration parameters to support automatic installation and configuration of new features. For additional information, see *Installing and Configuring Cray Linux Environment (CLE) Software* and the `CLEinstall`(8) man page.

Note: The installation program is now called `CLEinstall`. For more information, see `XTinstall` Utilities Renamed `CLEinstall` on page 49.

2.16.1.1 `CLEinstall` Support for New Features

The following parameters are now available in the `CLEinstall.conf` file to automatically configure the following features during a CLE software installation or upgrade.

- `DSL_`* parameters enable or disable DSL and configure DSL for your system.
- `CCM_`* parameters enable or disable CCM and configure CCM for your system. **(CLE 3.0 – 3.1 Change)**
- `sdbnode_failover_`* parameters enable or disable SDB node failover and optionally modify the SDB node failover network configuration. **(CLE 3.0 – 3.1 Change)**.

For additional information, see the new `CLEinstall.conf`(5) man page.

2.16.1.2 NFS Tuning Support in `CLEinstall.conf`

The `CLEinstall` program supports NFS tuning via `/etc/sysconfig/nfs` and `/etc/init.d/nfsserver` on the boot node.

The `nfs_mountd_num_threads` parameter in the `CLEinstall.conf` installation configuration file controls an NFS `mountd` tuning parameter that is added to `/etc/sysconfig/nfs` and used by `/etc/init.d/nfsserver` to configure the number of `mountd` threads on the boot node.

By default, NFS `mountd` behavior is unchanged (a single thread). For systems with more than 50 service I/O nodes, Cray recommends that you configure multiple threads by setting this parameter to `"4"`. If you have a larger Cray system (greater than 50 service I/O nodes), contact your Cray service representative for assistance changing the default setting.

2.16.2 New Node Allocation Mode of `other`

A new allocation mode (`alloc_mode`) of `other` is supported for compute nodes. Valid values for `alloc_mode` are now `batch`, `interactive`, and `other`. Nodes with an allocation mode of `batch` or `interactive` become part of the available pool of batch or interactive nodes respectively. Nodes with an allocation mode of `other` are no longer available to the compute node pool.

Compute nodes used as internal DVS servers to support dynamic shared objects use this allocation mode. For additional information, see Dynamic Shared Objects and Libraries (DSL) on page 21.

The new commands `xtalloc2db` and `xtdb2alloc` access the `alloc_mode` field in the Service Database (SDB). The `xtprocadmin` and `xtnodestat` commands have been modified to support an allocation mode of `other`. For additional information, see the `xtalloc2db`(8), `xtdb2alloc`(8), `xtnodestat`(1) and `xtprocadmin`(8) man pages.

2.16.3 Call of `rc.local` Script During Boot

The file `/etc/init.d/rc.local` is now available for local customization of the boot process. If this file/script is present, it is executed during the compute node boot. This script is executed after `/init`, before any of the scripts in `/etc/init.d/rc3.d` and before `/etc/fstab` is processed.

Note: This change was first released with the CLE 2.2.UP02 update package.

2.16.4 New `clerelease` File Indicates Installed Release Level

Following a successful installation, the file `/etc/opt/cray/release/clerelrease` is populated with the installed release level. For example,

```
% cat /etc/opt/cray/release/clerelease
3.1.UP00
```

2.16.5 `ldump` Enhancements

For more information about the following enhancements, see the `ldump`(8) man page.

2.16.5.1 New `-q` Option Turns Off Printing of a Dump's Progress

By default, the `ldump` utility prints the progress of the dump; use the new `-q` option to turn off printing of the dump's progress.

2.16.5.2 For Cray XE Systems: New Default Access Method Command-line String Name, `xt-bhs`

The new `xt-bhs` command-line string name is provided for the slow `ldump -r` method to access a Cray XE system and read node memory. The `xt-bhs` access method is equivalent to the `xt-ssi` access method supported on Cray XT systems, but the `xt-bhs` access method uses a basic hardware system server that runs on the SMW to access and read node memory.

The `xt-bhs` method is the default method used by `ldump` to access the Cray XE system and read node memory. For Cray XT systems, `xt-ssi` remains the default access method.

An updated `ldump`(8) man page is provided with the CLE 3.1 release package.

2.16.5.3 For Cray XE Systems: `ldump` Uses New NIC Mapping

Because the SMW 5.1 and CLE 3.1 software releases implement a new way of managing node IDs, `ldump` now uses Network Interface Controller ID (NIC) mapping of a Cray system. For Cray XT systems, NIC mapping is equivalent to NID mapping. For more information about NIC mapping, see *Cray System Management Workstation (SMW) 5.1 Software Release Overview* (S–2482–51).

2.17 Bugs Addressed Since the Last Release

The list of customer-filed critical and urgent bug reports that were closed with the CLE 3.1 release is included in the *CLE 3.1 Errata*; this document is provided with the release package.

Compatibilities and Differences [3]

This chapter compares Cray Linux Environment (CLE) 3.1 with CLE 2.2 and lists compatibility issues and functionality changes.

Note: The *CLE 3.1 Limitations* document describes temporary limitations of the release. The *CLE 3.1 Errata* document describes any installation and configuration changes identified after documentation for this release was packaged; it also includes a list of customer-filed critical and urgent bug reports that are closed with this release. These documents are included with the release package and are also available from your Cray representative.

3.1 Binary Compatibility

Binary compatibility is maintained from CLE 2.2 to CLE 3.1 for dynamically linked binaries. Users with applications that were built to use dynamic shared objects and libraries on a Cray XT system with CLE version 2.2 do not need to relink or recompile those applications on a Cray XT system running the CLE 3.1 operating system release.

Applications built on a Cray XT system that has a Cray SeaStar based system interconnection network will not run on a Cray XE system that has a Cray Gemini based system interconnection network.

For applications that use static libraries, some CLE 2.2 binaries will fail when run with CLE 3.1, due to differences between SLES 10 and SLES 11. For example, `syscall` numbers for `pfm/perfmon` changed, however, this is not a commonly used application interface. Cray tested a significant number of applications that were compiled and statically linked under CLE 2.2 and successfully ran them under CLE 3.1. However, this does **not** guarantee that all statically linked applications from CLE 2.2 are compatible with CLE 3.1.

CLE 2.2 service node commands that use the `libalps.a` library need to be re-linked on a CLE 3.1 system. For additional information, see ALPS Reservations File Format Changed on page 45.

If you have a Cray XT6 or Cray XT6m system running the CLE 3.0 release, you do not need to relink or recompile when upgrading to the CLE 3.1 release. Cray tested a significant number of applications that were compiled under CLE 3.0 and successfully ran them under CLE 3.1. However, this does **not** guarantee that all applications from CLE 3.0 are compatible with CLE 3.1.

Note: While relinking or recompiling may not be required, doing so may result in improved application performance.

3.2 PCI-X Network Interface Cards are Not Supported

The CLE 3.1 release supports PCIe network interface cards; PCI-X cards are currently not supported. Limited PCI-X support will be available in a future CLE 3.1 update release.

3.3 Changing the Default `syslog-ng` Configuration

CLE uses the Linux `syslog-ng` daemon and associated `syslog-ng.conf` configuration file to log system messages. The procedure for changing where the log information is saved has been simplified with the CLE 3.1 release.

For more information see *Managing System Software for Cray XE and Cray XT Systems*, and the `syslog-ng`(8) and `syslog-ng.conf`(5) man pages.

3.4 Commands Removed from the Release

The `xthostname` command has been removed from the CLE release. The `/etc/xthostname` file is automatically populated by the `CLEinstall` program during software installation.

The `xtchecklink` command has been removed from the CLE release. Comparable functionality is available by using the `xtverifyshroot` command. For additional information, see the `xtverifyshroot`(8) man page.

The `xtappackage` command was previously deprecated and has been removed from the CLE release. Comparable functionality is available using dynamic shared objects. For additional information, see Dynamic Shared Objects and Libraries (DSL) on page 21.

3.5 Deprecated Commands

The `xtuname`, `xtps`, `xtwho` and `xtkill` commands have been deprecated and will be removed in a future release. These utilities were developed to support the Catamount option for compute nodes, which is no longer available.

Comparable functionality for `xtps`, `xtwho` and `xtkill` is available using `pdsh` and the equivalent Linux command. For more information, see the `pdsh`(1) man page.

Comparable functionality for `xtuname` options is available as follows:

`-N` (node) `% rca-helper -i`

`-C` (class) `% xtnce `rca-helper -i``

`-S` (boot string)

`% cat /proc/cmdline`

3.6 Requirements for Cray Performance Analysis Tools

If you are doing performance tuning on an application by using hardware counters through CrayPat, or if you are using the Performance API (PAPI) directly, you must upgrade your performance tools package to version 5.1.

3.7 `make` Utility Enhancements and Differences

SLES 11 includes an upgrade of the GNU `make` utility, from version 3.80 to version 3.81. There are a number of enhancements and compatibility differences introduced with version 3.81. See the `NEWS` file included with the GNU make 3.81 package for a description of the changes.

3.8 Software Packages/Releases That Must be Reinstalled

If you are migrating from CLE 2.2, the following programming tools must be reinstalled to ensure compatibility with CLE 3.1. This is due to changes in CLE 3.1 included with the upgrade to SLES 11.

- STAT 1.0 and later versions
- PGI Compilers
- Cray Performance Tools (CrayPat, Apprentice2 and Papi)
- Performance tools 5.0.2 (Performance Tools 5.1.0 or later is recommended)
- GCC compilers
- Intel compilers

These products are not compatible with SLES 11. They must be uninstalled and a SLES11 compatible version must be installed after the upgrade if available.

- PathScale 3.2.99
- DDT 2.5

3.9 Changes in Setting System-wide Default Modulefiles

Important: Beginning with CLE 3.1, the location of the module files changed, all of the RPM names changed, and the set of module files that are loaded by default increased substantially. You should verify local scripts that include module file names or paths against the new names and paths, and update the scripts as needed.

The `Base-opts` module file now loads two lists of module files: a default list and a site-specified local list.

The default list differs between the SMW and the Cray system. On the SMW, the file `/etc/opt/cray/modules/Base-opts.default.SMW` contains the list of the CLE module files to load by default. On the Cray system, the file `/etc/opt/cray/modules/Base-opts.default` contains the list of CLE module files to load by default.

Also, all the module files listed in the file `/etc/opt/cray/modules/Base-opts.default.local` are loaded. Edit this file to make site-specific changes.

The `/etc/opt/cray/modules/Base-opts.default.local` file initially includes the `admin-modules` module file, which now loads a full set of module files. System administrators do not need to manually load the `admin-modules` module file, unless the site removes it from the default list. The CLE installation process removes `admin-modules` module file from the default list on login nodes.

An example file, `/etc/opt/cray/modules/Base-opts.default.local.example`, is also provided. The example file is a copy of the `/etc/opt/cray/modules/Base-opts.default.local` file provided for an initial installation. For additional information, see *Managing System Software for Cray XE and Cray XT Systems*.

3.10 ALPS Reservations File Format Changed

To support Cray X6 systems with 24 cores per node, the ALPS reservations file format has changed between the CLE 2.2 and CLE 3.1 releases. This change in file format has two consequences:

- Recovery of applications is not possible when making the transition between ALPS on CLE 2.2 and CLE 3.1; that is, all applications must have exited before shutting down ALPS on a CLE 2.2 system and restarting with the CLE 3.1 version of ALPS. Normally ALPS (specifically `apsched`) can recover running applications when it is shutdown and restarted, but the difference in the reservations file format makes this impossible.
- Locally developed service node administration commands that use the `libalps.a` library (not user applications, but commands that use `libalps` functionality to interpret the ALPS reservations and appinfo files) need to be re-linked to continue to function on a CLE 3.1 system. To avoid this issue in the future, in CLE 3.1, ALPS service node shared libraries have been provided in `/usr/lib/alps/libalps.so` and `/usr/lib/alps/libxmlrpc.so`; linking commands with these shared libraries ensures that further changes in the ALPS reservations file format will be transparent to the user.

3.11 Removed Obsolete Routing Entry in `xtnodestat` Legend

The `R` flag (`node is routing`) is obsolete and has been removed from the `xtnodestat` command output and legend.

3.12 Node Health Checker (NHC) Changed Functionality

3.12.1 NHC Release Upgrade Compatibility

When you upgrade to CLE 3.1, your previous release's `/etc/opt/cray/nodehealth/nodehealth.conf` file will not be overwritten if the file already exists. This will preserve any local modifications that you made to your previous NHC configuration file. However, you should always compare your `nodehealth.conf` file content with the `/opt/cray/nodehealth/default/etc/nodehealth.conf.example` file provided with each release to identify any changes, and then update your `nodehealth.conf` file on the shared root accordingly. (The example NHC configuration file is a copy of the `/etc/opt/cray/nodehealth/nodehealth.conf` file provided for an initial installation.)

If the `/etc/opt/cray/nodehealth/nodehealth.conf` file does **not** exist, then the `/opt/cray/nodehealth/default/etc/nodehealth.conf.example` file is copied to the `/etc/opt/cray/nodehealth/nodehealth.conf` file.

3.12.2 NHC Files, Binary, and Script Moved

To comply with new package and file naming conventions, the following NHC files and commands have new locations. For additional information, see Configuration and Packaging Enhancements on page 19.

`/etc/opt/cray/nodehealth/nodehealth.conf`

`/opt/cray/nodehealth/default/etc/nodehealth.conf.example`

`/opt/cray/nodehealth/default/bin/xtcheckhealth`

`/opt/cray/nodehealth/default/bin/xtcleanup_after`

3.12.3 `xtcleanup_after` Script Changes

The `xtcleanup_after` script now writes its normal launch information to the `/var/log/xtcheckhealth_log` file and writes error output (launch failure information) to the `/var/log/xtcheckhealth_log` file, to the console file on the SMW, and to the syslog.

In addition, a new `-f` *alt_NHCconfigurationfile* option enables you to specify which NHC configuration file to use with the `xtcleanup_after` script instead of using the default NHC configuration file.

For additional information, see the `xtcleanup_after`(8) man page.

3.12.4 `xtcheckhealth` Binary Interface Changes

The interface to the `xtcheckhealth` binary changed to:

`xtcheckhealth -h [-a` *apid*`] [-e` *exit_code*`] [-s]` *nodelistfile NHCconfiguration_file*

The application identifier (APID) and exit code arguments are now options. Also, *NHCconfiguration_file* is now an argument, not a redirected input file. (The new `-s` option indicates that `xtcheckhealth` should start in Suspect Mode, but only the login node crash recovery code uses this option.)

If you manually execute the `xtcheckhealth` binary to do a "dry run" and verify your configuration file settings, you now only need to specify the *nodelistfile* and *NHCconfiguration_file* arguments on the command line. The *nodelistfile* should contain all of the nodes that NHC is to check; these nodes should be listed as NIDs, one per line. There should be no blank line at the end of the *nodelistfile* file.

For additional information, see the `xtcheckhealth`(8) man page.

3.12.5 CLE 3.0–CLE 3.1: Additional NHC Changes

(CLE 3.0 – 3.1 Change) If you are upgrading from CLE 3.0 to CLE 3.1, you should note these additional changes:

- A node no longer remains in `suspect` state during a panic or crash. Instead, it changes state to `down`. After this happens, NHC recognizes the state change and stops monitoring the node within the time specified by the `recheckfreq` variable in the NHC configuration file, if not sooner.
- If you warm boot a compute node that is in `suspect` state, the node will no longer remain in `suspect` state after it is warm booted; the node will come back in the `up` state if the boot is successful. After this happens, NHC recognizes the state change and stops monitoring the node within the time specified by the `recheckfreq` variable in the NHC configuration file, if not sooner.
- The following NHC tests are renamed:
  - The `Application` test is now named `Application Exited Check`
  - The `Alps` test is now named `Apinit Ping`
  - The `Memory` test is now named `Free Memory Check`
  - The `Site` test is now named `Plugin`
- The example file `/opt/cray/nodehealth/default/etc/nodehealth.conf.example` provided with the CLE 3.1 release includes an example for the `Free Memory Check` test that shows how to set hard limits and soft limits for the `Free Memory Check` test.
- The `xtcheckhealth` binary added the `-h` usage option.

3.13 Installation and Configuration Changed Functionality for System Administrators

In addition to the feature information described in Chapter 2, Software Enhancements, system administrators should also note the following compatibility issues and differences that are associated with the CLE 3.1 release.

For detailed initial and update installation procedures, see *Installing and Configuring Cray Linux Environment (CLE) Software*. Migrations from CLE 2.2 (UP02 or later) will be supported in an update release; migration documentation will be provided with the update package. Supported System Configurations on page 10 describes the types of installations on CLE systems.

For temporary limitations of this release and changes identified after the documentation for this release was packaged, see the *CLE 3.1 Limitations* document provided with the release package. Additional information may be included in the *CLE 3.1 README* document provided with update packages.

3.13.1 Supported Upgrade Path

The CLE 3.1.UP00 release supports new system installations and upgrade installations from CLE 3.0. Migrations from CLE 2.2 on an existing Cray XT system will be supported in an update release package. Migration documentation will be provided with the update package. For more information, see Supported System Configurations on page 10.

Note: You must be running release version CLE 2.2.UP02 or later on your Cray XT system to migrate to the CLE 3.1 release.

3.13.2 System Management Workstation (SMW) Upgrade Requirements

You must install or upgrade the System Management Workstation (SMW) to the SMW 5.1 release before you install or upgrade to CLE 3.1.

Only Restriction of Hazardous Substances (RoHS) compliant SMWs are supported with the SMW 5.1 release.

For information about the content of the SMW 5.1 release, see *Cray System Management Workstation (SMW) 5.1 Software Release Overview* (S–2482–51).

3.13.3 Installation Time Required

The time required to install the CLE 3.1 release depends on a large number of site-specific variables. As with past releases, much of the installation or upgrade requires a dedicated system. The time required to install CLE 3.1 is comparable to installation times experienced with CLE 2.2.

However, migrations from CLE 2.2 (to be supported in a update release) involve moving from SLES 10 SP1 to SLES 11 and will require additional time to complete. For more information, see the migration documentation that will be provided with the update release package.

3.13.4 `XTinstall` Utilities Renamed `CLEinstall`

(CLE 3.0 – 3.1 Change) To more accurately reflect the command's purpose, the `XTinstall` installation program has been renamed to `CLEinstall`.

These additional changes to installation utilities support this change: The `CLEinstall` log files have been renamed from `/var/adm/cray/logs/XTinstall.*` to `/var/adm/cray/logs/CLEinstall.*`. The `--XTmedia` option for this command is now `--CLEmedia`. The `XTinstall.conf` installation configuration file is renamed to `CLEinstall.conf`. The `CRAYXTinstall.sh` installation script is renamed to `CRAYCLEinstall.sh`.

This is a name change only and does not impact functionality of the installation program or the associated configuration file. For additional information, see the `CLEinstall` and `CLEinstall.conf`(5) man pages.

3.13.5 `/dev/sd*` Device Ordering is Not Guaranteed

SCSI device names (`/dev/sd*`) are not guaranteed to be numbered the same from boot to boot under SLES 11. This inconsistency can cause serious system problems following a reboot. When installing CLE 3.1, you **must** switch to persistent device names for file systems on your Cray system.

Cray recommends that you use the `/dev/disk/by-id/` persistent device names. Use `/dev/disk/by-id/` for the root file system in the initramfs image and in the `/etc/sysset.conf` installation configuration file as well as for other file systems, including Lustre (as specified in `/etc/fstab` and `/etc/sysset.conf`). For more information, see *Installing and Configuring Cray Linux Environment (CLE) Software*.

Alternatively, you can define persistent names using a site-specific `udev` rule or `cray-scsidev-emulation`. However, only the `/dev/disk/by-id` method has been verified and tested. For information about `cray-scsidev-emulation`, see `scsidev` is Obsolete on page 55.

Caution: You must use `/dev/disk/by-id` when specifying the root file system. There is no support in the initramfs for `cray-scsidev-emulation` or custom `udev` rules.

3.13.6 Changes to the `CLEinstall.conf` Installation Configuration File

Changes to the `CLEinstall.conf` installation configuration file are reflected in the new `CLEinstall.conf`(5) man page.

The following parameters have been added to `CLEinstall.conf`.

`CCM`
`CCM_ENABLERSH`
`CCM_QUEUES`
`CCM_WLM`
`CCM_ENABLENIS`

New feature; see Cluster Compatibility Mode (CCM) on page 22.

`DSL`
`DSL_nodes`
`DSL_mountpoint`
`DSL_attrcache_timeout`

New feature; see Dynamic Shared Objects and Libraries (DSL) on page 21.

`sdbnode_failover`
`sdbnode_failover_IPaddr`
`sdbnode_failover_netmask`
`sdbnode_failover_interface`

New feature; see SDB Node Failover on page 36.

`nfs_mountd_num_threads`

New functionality; see NFS Tuning Support in `CLEinstall.conf` on page 38

`persistent_var_hostname`

Specifies the host name of the node that serves as the NFS server for persistent var.

The following parameters have default values that are different for a Cray XE system as compared to Cray XT system; the installation utilities install a template with default values appropriate for your system.

```
HSN_byte1
HSN_byte2
bootimage_bootnodeip
bootimage_bootifnetmask
bootimage_bootproto
bootnode_failover_IPaddr
bootnode_failover_netmask
bootnode_failover_interface
sdbnode_failover_IPaddr
sdbnode_failover_netmask
sdbnode_failover_interface
```

The following parameters are deprecated and have been removed from `CLEinstall.conf`.

```
bootimage_maxnid               # Cray XT5h only
forwarding_network
forwarding_netmask
forwarding_bootnode_interface
forwarding_sdbnode_interface
forwarding_syslognode_interface
lustre_max_procs_per_node      # Catamount only
OFED                           # Installed by default in CLE 3.1
```

3.13.7 `shell_bootimage_`*`label`*`.sh` Script Differences

The installation program creates a script called `shell_bootimage_`*`label`*`.sh`; this script is used to prepare boot images for the system set with the specified *label* in `/etc/sysset.conf`.

This script is now created in `/var/opt/cray/install` on the SMW; it is no longer created in `/tmp`.

A new `-b` *bootimage* option is available with the `shell_bootimage_`*`label`*`.sh` script to specify the boot image disk or file name. When creating the `shell_bootimage_`*`label`*`.sh` script during software installation, the `CLEinstall` program uses the `/etc/sysset.conf` file to determine the default *bootimage*. Use the new `-b` option to override the default. System administrators can use this option to manage multiple boot images. Note that when *bootimage* is an archive (cpio) file, you must manually copy the boot image file to the boot root.

A new `-C` *coldstart_dir* option is available with the `shell_bootimage_`*`label`*`.sh` script to specify the path to the HSS coldstart applets directory. The default is `/opt/hss-coldstart+gemini/default/xt` for Cray XE systems with Cray Gemini network interconnect or `/opt/hss-coldstart/default/xt` for Cray XT systems with Cray SeaStar network interconnect. Use this option to override the default. For more information, see the `xtbounce`(8) man page.

3.13.8 `shell_*` Installation Scripts on the Boot Node Have Moved

Installation scripts are installed in `/var/opt/cray/install` on the boot node; they are no longer copied to `/tmp`. During the installation process, you are directed to invoke the following scripts on the boot node:

```
/var/opt/cray/install/shell_bootnode_first.sh
/var/opt/cray/install/shell_bootnode_second.sh
/var/opt/cray/install/shell_ssh.sh
/var/opt/cray/install/shell_sshkeys.sh
```

3.13.9 Local Changes to `*rc.local` Scripts are Maintained After a CLE Upgrade

The `prgenv` RPM adds a section to the `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local` scripts, which set default modulefiles to be loaded. `##BEGIN` and `##END` tags delimit the contents of this section. In previous CLE releases, when the RPM was reinstalled, this section was removed and re-appended to the scripts. Any local changes that had been added after the original section were moved in the process.

In CLE 3.1, the `/etc/bash.bashrc.local` and `/etc/csh.cshrc.local` scripts have clearly delimited sections for operating system changes. A CLE upgrade modifies these sections in place, maintaining any local changes system administrators have made outside of the delimited block and, more importantly, the order of the blocks within the file.

Note: This change was first released with the CLE 2.2.UP02 update package.

3.14 General System Administration Differences

3.14.1 For Cray XE Systems: Gemini Component Naming and Numbering

System administrators familiar with Cray XT systems will notice changes in how nodes are named and numbered on Cray XE systems with Cray Gemini network interconnect when compared to Cray XT systems.

The integer node IDs (NIDs) are numbered sequentially starting in cabinet `c0-0`. Each additional cabinet continues from the highest value of the previous cabinet; that means cabinet `0` has NIDs `0-95`, cabinet `1` has NIDs `96 - 191`, and so on.

Because a single Gemini ASIC connects to two nodes, the node numbering within a cabinet is different from a SeaStar based system. This changes the physical name to NID mapping. For example, `c0-0c0s2n3` is node 11 on a Cray XT system and node 27 on a Cray XE system.

To support the Gemini ASIC and network interface controllers, the physical component naming convention has been expanded.

For more information, see *Cray System Management Workstation (SMW) 5.1 Software Release Overview* (S–2482–51).

3.14.2 Configuration and Packaging Changes

The CLE 3.1 release changes the location of various Cray-specific packages and associated configuration files. For information about the changes, see Configuration and Packaging Enhancements on page 19. For a summary of new package locations and a partial list of files that have moved, see Appendix A, Configuration and Packaging Changes on page 75.

3.14.3 Quotes are Allowed in the System Configuration File

The system configuration file (`/etc/sysconfig/xt`) file is now structured according to conventions for a `sysconfig` file. Both quoted and non-quoted values are allowed. If a value contains a quote (or double quotes) at both the beginning and end, the quotes are removed. All other data is unaltered.

Note: In previous CLE releases, the `/etc/sysconfig/xt` file was located in `/etc/xt.conf`.

3.14.4 IP Forwarding No Longer Required on the Boot Node

(CLE 3.0 – 3.1 Change) A new `mzproxy` daemon on the boot node provides controlled communication between the Service Database (SDB) and Cray Management Services (CMS) on the SMW when IP forwarding/routing is disabled on the boot node.

The CLE 2.2 and SMW 4.0 releases required that IP forwarding be enabled on the boot node to allow ALPS and syslog information to pass from the SDB to the SMW. IP forwarding is no longer a requirement in CLE 3.1 and is off by default. For more information, see the `mxproxy`(8) man page.

3.14.5 Virtual IP Addresses for Boot and SDB Nodes

Virtual IP addresses are now configured for the boot and SDB nodes. This change supports failover for these nodes. However, the virtual IP addresses are configured by default, regardless of the boot or SDB node failover configuration on your system.

To support this functionality, these default IP addresses have changed in the `CLEinstall.conf` installation configuration file: `bootimage_bootnodeip`, `persistent_var_IPaddr`, `bootnode_failover_IPaddr`, and `sdbnode_failover_IPaddr`. You can change these defaults according to your site requirements; however, the default values are acceptable in most cases.

As part of this change, the default user-prompts on the active boot and SDB nodes use generic `boot` and `sdb` hostnames and no longer include a boot node or node id number. For example,

```
boot:~ #
sdb:~ #
crayadm@boot:~>
crayadm@sdb:~>
```

3.14.6 Pluggable Authentication Module (PAM) Now Controls `/etc/nologin`

OpenSSH transferred control of the `/etc/nologin` file to PAM. System administrators must configure PAM to require `pam.nologin.so`. This change is needed to prevent users with key-based authentication from logging in when `/etc/nologin` exists. On the login node, edit `/etc/pam.d/sshd` and add this line:

```
account required pam_nologin.so
```

3.14.7 ALPS Adds Additional Job Information to Syslog Messages

ALPS now collects more complete job information and preserves it in the system log. Commands `aprun` and `apsys` both add the batch job ID to the syslog messages written about an application. In addition, if `apsys` receives an application exit message, the exit code or signal is also added to the syslog message.

3.14.8 Lustre 1.8 File System Compatibility

Lustre 1.8 clients and servers can access older Lustre 1.6 file systems. Therefore, you do not have to reformat a file system that was originally formatted with Lustre 1.6. However, a file system formatted by Lustre 1.8 is not backward compatible.

3.14.9 Lustre File System Configuration Requires Persistent Device Names

Because SCSI device names (`/dev/sd*`) are not guaranteed to be numbered the same from boot to boot, the configuration specified in the Lustre file system definition file must use persistent device names. If non-persistent device names are used, the file system may fail to start when the system is rebooted. For more information, see `/dev/sd*` Device Ordering is Not Guaranteed on page 49.

Cray Lustre utilities have new routines that ease the migration to persistent device names. System administrators use new action options with `lustre_control.sh` (`verify_config`, `update_config` and `dump_target_devnames`) to generate a Lustre file system definition file based on the current correct device name configuration of the system.

For procedures involving these utilities, see *Installing and Configuring Cray Linux Environment (CLE) Software* and *Managing Lustre for the Cray Linux Environment (CLE)*. For additional information, see the `lustre_control.sh`(8) man page.

3.14.10 scsidev is Obsolete

SLES 11 no longer supports the scsidev method for SCSI device naming. Cray recommends that you create persistent device names by using udev to create device nodes with persistent names in /dev/disk/by-id/. For more information, see /dev/sd* Device Ordering is Not Guaranteed on page 49.

As an alternative, CLE 3.1 includes a limited, as-is capability to emulate the basic functionality of scsidev called cray-scsidev-emulation. A device alias is created in /dev/scsi for any devices that match entries in the alias file, /etc/scsi.alias. Format the alias file as follows, where *SN* is the serial number of the hard disk, *PN* is the partition number of the disk, and *devname* is the desired alias name.

```
serial_number="SN", devtype=disk, [partition=PN,] alias=devname
```

For example, the following entry creates a device entry /dev/scsi/cab2-3-c1-shroot for partition 2 on the disk with the serial number 030B9ED30300.

```
serial_number="030B9ED30300", devtype=disk, partition=2, alias=cab2-3-c1-shroot
```

Note: The following limitations apply when using cray-scsidev-emulation:

- This capability is not implemented in the initramfs; it cannot be used to specify the boot root.
- Only the format shown is supported; scsidev supported a number of additional formats.
- Only one alias per disk is supported.
- Only symbolic links are supported.

3.14.11 cnos Specialization Class for Compute Node Shared Root

The CLEinstall program creates a default cnos specialization class. This class allows an administrator to specialize files specifically for compute nodes; it is used with dynamic shared objects and libraries (DSL). If the cnos specialization class exists and DSL is enabled, those specialized /etc files are automatically mounted on the compute node roots.

3.14.12 xtpackage and xtclone Must be Invoked as root

System administrators must now have root privileges to invoke the xtpackage and xtclone commands on the SMW. This change was made to increase security and integrity of files in the initramfs.

3.14.13 `xtrelswitch` No Longer Supports Switching the SDB

The `xtrelswitch` command no longer supports switching of the SDB. Because the SDB software is now a separate RPM in the CLE release package, it is no longer necessary to support release switching of the SDB.

3.14.14 `xtprocadmin` Command Differences

Obsolete columns removed from output. The `PSLOTS` and `FREE` columns are no longer used and have been removed from the `xtprocadmin` output.

Sending of RCA event no longer optional. The `xtprocadmin` command automatically sends an RCA event to indicate nodes that have been updated for processor and allocation state. The `-e` and `-E` options that force sending an RCA event are obsolete and have been removed.

3.14.15 `ldump` Command Differences

`xt` access method no longer accepted. The `xt` access method command-line string name is no longer accepted for use with the `ldump -r` option. The `xt` command-line string name was a previously deprecated alias for the `xt-ssi` method.

`xt-ssi` access method change. For the `xt-ssi` access method, if *host* is not specified, `smw` is used. Previously, if *host* was not specified, `ldump` connected to the event router daemon on the L0 for the node being dumped and used it to read the node memory.

"NID Mapping" now referred to as "NIC Mapping". Because the SMW 5.1 and CLE 3.1 software releases implement a new way of managing node IDs, `ldump` now uses Network Interface Controller ID (NIC) mapping of a Cray system. However, for Cray XT systems, NIC mapping is equivalent to NID mapping.

3.14.16 `xtcdr2proc` Supported Only on the Boot Node

The `xtcdr2proc` command must be invoked on the boot node. It is no longer supported on the SMW because it requires RCA to perform NID and NIC conversions.

3.15 Documentation Differences

(CLE 3.0 – 3.1 Change) The following new books are provided with the release.

- *Workload Management and Application Placement for the Cray Linux Environment* (S–2496): Introduces ALPS user features and the other infrastructure of CLE and programming environments provided for application programming on Cray systems. Gives users basic tips on command line ALPS and workload management tools for launching jobs. Includes user-level documentation for Cluster Compatibility Mode (CCM).
- *Using the GNI and DMAPP APIs* (S–2446): Provides overview, tasks and scenarios, and C++ reference information for the GNI and DMAPP APIs.
- *Writing a Node Health Checker (NHC) Plugin Test* (S–0023): Provides information to help you write NHC plugin tests.

For improved usability, information in several CLE books and other documents has moved to new or existing books. For more information, see Cray-developed Books No Longer Provided with this Release on page 62.

Documentation [4]

This chapter describes the documentation that supports the Cray Linux Environment (CLE) 3.1 release.

4.1 Cray Documentation Website

The Cray Documentation website is accessible to the public. It contains documentation for each Cray product, release announcements, single-subject technical articles, and links to other information resources, see docs.cray.com.

4.2 CrayPort Website

The CrayPort website is updated with product documentation for each Cray software release and is accessible to CrayPort registered users, see crayport.cray.com.

4.3 CrayDoc Documentation Delivery System

The CrayDoc documentation delivery system, along with product documentation, is provided with each Cray software release and can be installed at your site. The CrayDoc software runs on many UNIX-based operating systems including Solaris and Linux. The installation and administration of the CrayDoc server software and Cray documentation are described in *CrayDoc Installation and Administration Guide*.

4.4 Accessing Product Documentation

With each software release, Cray provides books and man pages, and in some cases, third-party documentation. These documents are provided in the following ways:

CrayPort

CrayPort is the external Cray website for registered users that offers documentation for each product. CrayPort has portal pages for each product that contains links to all of the documents that are associated to that product. CrayPort enables you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. You access CrayPort by using the following URL:

crayport.cray.com

CrayDoc

CrayDoc is the Cray documentation delivery system. CrayDoc enables you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. Access the HTML and PDF documentation via CrayDoc at the following locations.

- The local network location defined by your system administrator
- The CrayDoc public website: docs.cray.com

Man pages

Man pages are textual help files available from the command line on Cray machines. To access man pages, enter the `man` command followed by the name of the man page. For more information about man pages, see the `man`(1) man page by entering:

```
% man man
```

Third-party documentation

Third-party documentation that is not provided through CrayPort or CrayDoc is included with of the third-party product.

4.5 Cray-developed Books Provided with This Release

The books provided with this release are listed in Table 3, which also indicates whether each book was updated. Books are provided in HTML and PDF formats.

Table 3. Books Provided with This Release

Book Title	Number	Updated
Cray Linux Environment (CLE) Software Release Overview (this document)	S–2425–31	Yes
Installing and Configuring Cray Linux Environment (CLE) Software	S–2444–31	Yes
Managing System Software for Cray XE and Cray XT Systems	S–2393–31	Yes
Managing Lustre for the Cray Linux Environment (CLE)	S–0010–31	Yes
Introduction to Cray Data Virtualization Service	S–0005–31	Yes
Writing a Node Health Checker (NHC) Plugin Test	S–0023–31	New
Workload Management and Application Placement for the Cray Linux Environment	S–2496–31	New
Using the GNI and DMAPP APIs	S–2446–31	New
CrayDoc Installation and Administration Guide	S–2340–411	No

4.5.1 Additional Cray-developed Release Documents

Two additional documents are provided with the CLE 3.1 release package. These documents are also available from your Cray representative.

CLE 3.1 Limitations

Describes temporary limitations of the release.

CLE 3.1 Errata

Describes any installation and configuration changes that were identified after documentation for this release was packaged; also includes a list of customer-filed critical and urgent bug reports closed with this release.

You should also contact your Cray representative about CLE-related information addressed in Field Notices (FNs).

4.5.2 Cray-developed Books No Longer Provided with this Release

For improved usability, information from the following technical documents, provided with the CLE 2.2 release package, has been incorporated into other new or existing books.

- *Application Cleanup by ALPS and Node Health Monitoring* (S–0014–22) now documented in *Managing System Software for Cray XE and Cray XT Systems* (S–2393–31).
- *OpenFabrics Interconnect Drivers for Cray XT Systems* (S–2483–22) now documented in *Managing System Software for Cray XE and Cray XT Systems* (S–2393–31) and *Installing and Configuring Cray Linux Environment (CLE) Software* (S–2444–31).
- *Configuring and Using Dynamic Shared Objects and Libraries for the Cray Linux Environment (CLE)* (S–0012–22) now documented in *Managing System Software for Cray XE and Cray XT Systems* (S–2393–31), *Workload Management and Application Placement for the Cray Linux Environment* (S–2496–31) and *Installing and Configuring Cray Linux Environment (CLE) Software* (S–2444–31).
- *Cray XT Programming Environment User's Guide* (S–2396–22) now documented in *Cray Application Developer's Environment User's Guide* (S–2396-50) and *Workload Management and Application Placement for the Cray Linux Environment* (S–2496–31).

4.6 Third-party Books Provided with This Release

The CLE 3.1 release package includes the following book from Oracle:

Lustre Operations Manual (S–6540–1813)

4.7 Changes to Man Pages

Updated Linux man pages are included with the CLE 3.1 release. For complete information regarding changes to specific commands due to the upgrade to SLES 11, see the associated man pages. To access Linux man pages, use the `man` command on a login node.

4.7.1 New Cray Man Pages

These man pages are new with this release.

- `apcount`(1): Calculates the scaled width of the batch reservation that users will need to use with core specialization.
- `ccmrun`(1): Launches a cluster application within the Cray Cluster Compatibility Mode (CCM) environment.
- `ccmlogin`(1): Provides a mechanism to log in to the head node in a CCM job; implemented using SSH and accepts SSH arguments.
- `CLEinstall.conf`(1): Describes the parameters in the `CLEinstall.conf` installation configuration file. The man page is new; the file is not new but it has been renamed.
- `nhc_recovery`(8): Displays and updates information stored in the SDB for login node crash recovery.
- `xtalloc2db`(8) and `xtdb2alloc`(8): Support for `alloc_mode` in the Service Database (SDB).
- `xtlusfoevntsndr`(8): Lustre failover imperative recovery event sending agent.

4.7.2 Removed Cray Man Pages

The following Cray man pages were removed with this release:

- `xtappackage`(1)
- `xtchecklink`(8)
- `xthostname`(8)
- `xtok2`(8)

4.7.3 Changed Cray Man Pages

Most Cray-specific man pages have been updated to reflect changes in file locations. Source for Cray-specific man pages is included in the associated RPM and is installed in a new location, `/opt/cray/share/man`.

The following Cray man pages have additional updates and enhancements:

- `aprun`(1): Adds `-F` option to specify `exclusive` (default) or `share` mode; adds `-B` option to keep options specified at batch reservation; adds `-r` option to enable core specialization; adds information on setting `HUGETLB_DEFAULT_PAGE_SIZE`; miscellaneous other changes.
- `apstat`(1): Adds support for Cray Gemini performance counters and core specialization.
- `basil`(7): Documents the changes to the `basil.h` header file.
- `intro_NHC`(8), `xtcheckhealth`(8), and `xtcleanup_after`(8): Reflects the NHC enhancements for this release.
- `lustre.fs_defs`(5): Documents several new parameters for Lustre imperative recovery and includes changes for persistent device naming.
- `xt-lustre-proxy`(8): Adds support for Lustre imperative recovery.
- `lustre_control.sh`(8): Adds persistent device naming options.
- `xtprocadmin`(8): Reflects changes to support `alloc_mode` in the SDB.
- `xtnodestat`(1): Reflects changes to support `alloc_mode` in the SDB; removed obsolete `R` from legend.

4.8 Other Related Documents Available

The following publications contain additional information that may be helpful in setting up your Cray system; they are not provided with this release but are supplied with other products purchased from Cray. You can access these publications from the CrayPort website or you can order them using the CrayDoc CD from the Cray Software Distribution Center (see Ordering Documentation on page 66). You can also order the printed form of release overviews and installation guides from Cray.

Table 4. Other Related Documents Available

Book Title	Number
Cray System Management Workstation (SMW) Software Release Overview	S–2482–51
Installing Cray System Management Workstation (SMW) Software	S–2480–51
Using Cray Management Services (CMS)	S–2484–50
Using and Configuring System Environment Data Collections (SEDC)	S–2491-50
Cray Application Developer's Environment Installation Guide	S–2465
Cray Compiling Environment Release Overview and Installation Guide	S–5212

4.9 Additional Documentation Resources

Table 5 lists additional resources for obtaining documentation not included with this release package.

Table 5. Additional Documentation Resources

Product	Documentation Source
Linux	Documentation for SLES and Linux is at www.novell.com/linux and documentation for the Linux Documentation Project is at www.tldp.org
Lustre	Additional Lustre documentation is available at wiki.lustre.org/index.php/Lustre_Documentation and www.oracle.com/us/products/servers-storage/storage/storage-software
MySQL	MySQL documentation is available at www.mysql.com/documentation
RPM	RPM documentation is available at www.rpm.org
PBS Professional	Documentation for the PBS Professional work load manager system software is available from Altair Engineering, Inc. at www.altair.com
Moab and TORQUE	Documentation for Moab and TORQUE work load manager system software is available from Adaptive Computing: www.adaptivecomputing.com/
Platform LSF	Documentation for Platform LSF software is available from Platform Computing Corporation at www.platform.com/
Berkeley Lab Checkpoint/Restart (BLCR)	Documentation for BLCR is available from Berkeley Lab at upc-bugs.lbl.gov/blcr/doc/html/

4.10 Ordering Documentation

To order Cray software documentation, contact your Cray representative or contact the Cray Software Distribution Center in any of the following ways:

E-mail:
orderdsk@cray.com

Telephone (inside U.S., Canada):
1–800–284–2729 (BUG CRAY), then 605–9100

Telephone (outside U.S., Canada):
+1–651–605–9100

Fax:
+1–651–605–9001

4.11 Cray Glossary

The entire Cray Glossary is available on the CrayDoc website (docs.cray.com). A Cray Glossary of terms for Cray XT and Cray XE systems is included with the installable CrayDoc documentation delivery system.

Release Contents [5]

5.1 Hardware Requirements

The Cray Linux Environment (CLE) 3.1 release supports new or initial installations on Cray XE6, Cray XT6, Cray XT6m, and Cray XT5m systems. Upgrade installations from CLE 3.0 are supported for Cray XT6 and Cray XT6m systems.

Support for other Cray hardware platforms will be provided in a CLE 3.1 update release; for more information, see Supported System Configurations on page 10, Table 1.

5.2 Software Requirements

The following sections list the required or recommended release levels for products that run on Cray systems but are released separately from CLE 3.1.

5.2.1 Release Level Requirements for Other Cray Software Products

The product versions listed in Table 6 are the minimum release level required for verified compatibility with CLE 3.1. Support for these products is provided in the form of updates to the latest released version only. Unless otherwise noted in the associated release documentation, Cray recommends that you continue to upgrade these releases as updates become available.

Table 6. Minimum Release Level Requirements for Other Software Products with CLE 3.1

Product	Minimum Release Level	Release Information
System Management Workstation (SMW)	Release 5.1 or later.	*Cray System Management Workstation (SMW) Software Release Overview* (S–2482–51)
Cray Application Developer's Environment (CADE)	Release 5.0 or later.	*Cray Application Developer's Environment Installation Guide* (S–2465)

Product	Minimum Release Level	Release Information
Cray Performance Analysis Tools	CrayPat 5.0.2 and Cray Apprentice2 5.0.2 or later. Release 5.1.0 or later is recommended.	*Cray Performance Analysis Tools Release Overview and Installation Guide* (S–2474)
Cray Compiling Environment (CCE)	Release 7.2.4 or later.	*Cray Compiling Environment Release Overview and Installation Guide* (S–5212)

5.2.2 Third-party Software Requirements

Third-party compiler products are available for Cray systems as noted in Table 7. The release level indicated has been tested with CLE 3.1. Cray recommends that you continue to upgrade these products as updates become available.

Table 7. Minimum Release Level Requirements for Third-party Compilers with CLE 3.1

Product	Minimum Release Level	Release/Ordering Information
PGI Compiler	Release 10.2 or later.	Contact your Cray representative for licensing/purchasing information. For product information see The Portland Group, Inc.: www.pgroup.com
PathScale Compiler	No SLES 11 compatible version currently available. PathScale version 3.2 and 3.2.99 C++ compilers are not supported on CLE 3.1 (SLES 11) systems.	Open source version available via PathScale Inc. at the following web site: www.pathscale.com
Intel Compiler	Release 11.1.064 or later.	Intel Corporation. See: software.intel.com

Batch system software products are available for Cray systems as indicated in Table 8. Information regarding supported and certified batch system software release levels is available on the CrayPort website at crayport.cray.com. Click on **3rd Party Batch SW** in the menu bar. For information about accessing CrayPort, see CrayPort on page 73.

Table 8. Third-party Batch System Software Products Available for Cray Systems

Product	Minimum Release Level	Release/Ordering Information
Moab and TORQUE:	Moab Version 5.3.4 or later. TORQUE 2.3.4 or later.	Contact your Cray representative for licensing/purchasing information. For product information see Adaptive Computing: www.adaptivecomputing.com/
PBS Professional:	Release 10.0 or later.	Contact your Cray representative for licensing/purchasing information. For product information see Altair Engineering, Inc.: www.altair.com/
Platform LSF:	Release 7 Update 3 or later.	Contact Platform Computing Corporation. See: www.platform.com/

5.3 Supported Upgrade Path

The CLE 3.1.UP00 release supports new system installations and upgrade installations from CLE 3.0. Migrations from CLE 2.2 on an existing Cray XT system will be supported in an update release package. Migration documentation will be provided with the update package. For more information, see Supported System Configurations on page 10.

Note: You must be running release version CLE 2.2.UP02 or later on your Cray XT system to migrate to the CLE 3.1 release.

The System Management Workstation (SMW) must be Restriction of Hazardous Substances (RoHS) compliant and must be running the SMW 5.1 release before you install the CLE 3.1 operating system release package.

5.4 Contents of the Release Package

The release package includes:

- All necessary RPMs and installation utilities for the components listed in CLE 3.1 Software Components on page 70
- CrayDoc software suite and the documentation, described in Chapter 4, Documentation on page 59

- A printed copy of this release overview
- A printed copy of the *Installing and Configuring Cray Linux Environment (CLE) Software*
- A printed copy of the *CLE 3.1 Limitations*
- A printed copy of the *CLE 3.1 Errata*
- A printed copy of the *CLE 3.1 README*

5.4.1 CLE 3.1 Software Components

The CLE 3.1 release includes, but is not limited to, the following system software products:

- Cray's customized version of the SLES 11 operating system with a Linux 2.6.16.27 kernel
- CNL compute node operating system
- Lustre file system (Version 1.8) from Oracle.
- Application Level Placement Scheduler (ALPS)
- Cray Data Virtualization Service (Cray DVS)
- Checkpoint/Restart (CPR)
- Cluster Compatibility Mode (CCM)
- Comprehensive System Accounting (CSA)
- Cray Audit
- Dynamic Shared Objects and Libraries (DSL)
- Linux `ldump` and `lcrash` Utilities
- MySQL Pro
- Node Health Checker (NHC)
- OpenFabrics InfiniBand
- Realm-Specific Internet Protocol (RSIP)

5.5 Licensing

The CLE release is covered under a software license agreement for Cray software. Upgrades to this product are provided only when a software support agreement for this Cray software is in place.

Cray licenses the following as separate products for Cray systems under a Cray license agreement:

- Cray XT/XE OS binary (which provides rights to the CLE operating system and its components)

  Note: Source Code Option: The Cray XT/XE OS license for Cray XE and Cray XT systems is binary by default. Certain U.S. customers may be eligible to obtain a buildable OS source license on Cray XE and Cray XT systems for an additional fee. For more information regarding source code, contact your sales representative.

- Lustre Parallel File System

In addition, effective with the CLE 3.1 release and SMW 5.1 release, the version of MySQL software included with the releases is a proprietary version. All customers are required to sign a new MySQL addendum to their software license agreement with Cray before they can receive the CLE 3.1 and SMW 5.1 releases. For some customers, this is an additional addendum to a previous MySQL addendum that was required with CLE 2.2 and SMW 4.0 releases.

For more information about licensing and pricing, contact your Cray sales representative, or send e-mail to crayinfo@cray.com.

5.6 Ordering Software

This release package is distributed by order only to customers who have signed a license agreement for the Cray software that includes this product. The most current revision of the release package is supplied. To receive any upgrades to a given Cray product, the customer must also have a signed support agreement for this Cray software.

Note: Customers outside the United States and Canada must sign a Letter of Assurance before software can be shipped to them. For questions about whether you have signed this agreement, or questions about which software requires this letter, send e-mail to crayinfo@cray.com.

You can order the release package from the Cray Software Distribution Center in any of the following ways:

E-mail:
orderdsk@cray.com

CrayPort (for subscribers):
crayport.cray.com

Click on the **Order Cray Software** link.

Telephone (inside U.S., Canada):
1–800–284–2729 (BUG CRAY), then 605–9100

Telephone (outside U.S., Canada):
+1–651–605–9100

Fax:
+1–651–605–9001

Cray will be ship the software via ground service (US), or International Economy, unless otherwise requested.

Customer Services [6]

6.1 Technical Assistance with Software Problems

If you experience problems with Cray software, contact your Cray service representative. Your service representative will work with you to resolve the problem. If you choose to have full- or part-time support on site, your on-site personnel are your primary contacts for service. If you have elected not to have on-site support, please call or send e-mail to the Cray Customer Support Center:

E-mail:
support@cray.com

Telephone (inside U.S., Canada):
1–800–950–2729 (CRAY)

Telephone (outside U.S., Canada):
+1–715–726–4993

CrayPort (for subscribers):
crayport.cray.com

As a CrayPort subscriber, you can request technical assistance using the Case Management interface. The Case Management interface lets you search, view, update, and close your cases online. If you signed up to participate in the Bug tracking portion of CrayPort, you can also view all of the Bugs for a particular system, even those Bugs that originate from other sites. Under this new system, you can quickly locate solutions to problems that have been encountered by other customers.

6.2 CrayPort

CrayPort provides information and problem reporting to Cray customers who subscribe to CrayPort. You are a potential CrayPort subscriber if your site has a software license agreement and software support agreement.

Some of the tasks a CrayPort subscriber can perform include:

- Read news and helpful information about Cray Service
- Report software problems
- Read about software problems reported at other sites (if you elected to share Bugs)
- Request technical assistance through the Case Management interface
- View, update, and close cases that you originate
- Order Cray software
- View Cray software product documentation
- View Cray service documentation
- View third-party batch software information

To access CrayPort, use the following link: crayport.cray.com. If you need an account, click the **Customer Registration Form** button on the login page.

6.3 Training

To find out more about Cray training, contact your Cray representative or contact us in any of the following ways:

E-mail:
registrar@cray.com

Web:
www.cray.com/training/

Fax:
+1–715–726–4991

6.4 Cray Public Website

The Cray public website offers information about a variety of topics and is located at:

www.cray.com

Configuration and Packaging Changes [A]

The CLE 3.1 release includes a number of changes to the location of software packages and associated configuration files. These changes were made to align with Cray-specific naming conventions. In general, these conventions follow LANANA standards for package naming and Filesystem Hierarchy Standard (FHS) for file locations. This appendix compares locations and names to those used in CLE 2.2 and earlier releases. This is not a comprehensive list and not all Cray-specific software conforms to these standards. For more information about LANANA and FHS, see www.lanana.org and www.pathname.com/fhs.

The following conventions were used to determine the new locations for files or package names (RPMs) on the Cray system and on the System Management Workstation (SMW).

Cray specific packages installed on the boot root and shared root have changed location and/or RPM names following this convention:

CLE 2.2: /opt/*old_package_name*/*package_version*

CLE 3.1: /opt/**cray**/*new_package_name*/*package_version*

For example, `/opt/xt-boot/default` is now `/opt/cray/boot/default`.

Cray-specific packages installed on the SMW have changed location and/or RPM names following this convention:

CLE 2.2: /opt/*old_package_name*/*package_version*

CLE 3.1: /opt/**cray-xt-***new_package_name*/*package_version*

For example, `/opt/xt-rsipd` is now `/opt/cray-xt-rsipd` and `/opt/cray-dropbear` is now `/opt/cray-xt-dropbear`.

Cray-specific CLE configuration files have changed location following this convention:

CLE 2.2: /etc/*configfile_name*

CLE 3.1 (on the Cray system):

/etc/**opt/cray/*package_name***/*configfile_name*

CLE 3.1 (on the SMW):

/etc/**opt/cray-xt-*package_name***/*configfile_name*

For example, `/etc/node_classes` is `/etc/opt/cray/sdb/node_classes` on the Cray system and `/etc/poller.conf` is `/etc/opt/cray-xt-lbnamed/poller.conf` on the SMW.

Table 9. New Locations for Specific Configuration Files on the Shared Root

CLE 2.2 Location	CLE 3.1 Location
`/etc/xt.conf`	`/etc/sysconfig/xt`
`/etc/xtrelease`	`/etc/opt/cray/release/xtrelease`
n/a	`/etc/opt/cray/release/manifests/*`
`/etc/lustre/*`	`/etc/opt/cray/lustre-utils/*`
`/etc/attr.defaults`	`/etc/opt/cray/sdb/attr.defaults`
`/etc/attr.xthwinv`	`/etc/opt/cray/sdb/attr.xthwinv`
`/etc/db.conf`	`/etc/opt/cray/sdb/db.conf`
`/etc/node_classes`	`/etc/opt/cray/sdb/node_classes`
`/etc/sdbfailover.conf`	`/etc/opt/cray/sdb/sdbfailover.conf`
`/etc/serv_cmd`	`/etc/opt/cray/sdb/serv_cmd`
`/etc/attributes`	`/etc/opt/cray/sdb/attributes`
`/etc/fstab_alias.conf`	`/etc/opt/cray/sdb/fstab_alias.conf`
`/etc/processor`	`/etc/opt/cray/sdb/processor`
`/etc/segment`	`/etc/opt/cray/sdb/segment`
`/etc/nids`	`/etc/opt/cray/configuration/nids`
`/etc/machines`	`/etc/opt/cray/pdsh/machines`
`/etc/fomd.conf`	`/etc/opt/cray/rca/fomd.conf`
`/etc/rca_dispatcher.conf`	`/etc/opt/cray/rca/rca_dispatcher.conf`
`/etc/rca_svcs.conf`	`/etc/opt/cray/rca/rca_svcs.conf`
`/etc/xtshutdown.conf`	`/etc/opt/cray/init-service/xtshutdown.conf`

CLE 2.2 Location	CLE 3.1 Location
/etc/xt_shutdown_local	/etc/opt/cray/init-service/xt_shutdown_local
/etc/hosts_alias.conf	/etc/opt/cray/hosts/hosts_alias.conf
/etc/service_alias.conf	/etc/opt/cray/hosts/service_alias.conf
/etc/rsipd.conf	/etc/opt/cray/rsipd/rsipd.conf
/etc/sysconfig/nodehealth	/etc/opt/cray/nodehealth/nodehealth.conf
/etc/csa.conf	/etc/opt/cray/csa/csa.conf
/etc/csa.holidays	/etc/opt/cray/csa/csa.holidays
/etc/roots	/etc/opt/cray/cnrte/roots
/etc/shared_roots.conf	/etc/opt/cray/cnrte/shared_roots.conf
/etc/xtfaillog.conf	/etc/opt/cray/pam/faillog.conf
/etc/my.cnf	/etc/opt/cray/MySQL/my.cnf

Table 10. New Locations for Miscellaneous Packages, Associated Files and Executables

CLE 2.2 Location	CLE 3.1 Location
/opt/xt-service/default/etc/\ sysconfig/nodehealth.example	/opt/cray/nodehealth/default/etc/\ nodehealth.conf.example
/opt/xt-service/default/bin/\ snos64/xtcheckhealth	/opt/cray/nodehealth/default/bin/\ xtcheckhealth
/opt/xt-service/default/bin/\ snos64/xtcleanup_after	/opt/cray/nodehealth/default/bin/\ xtcleanup_after
/usr/sbin/rsipd	/opt/cray/rsipd/default/sbin/rsipd

Differences Between CLE 3.0 and CLE 3.1 [B]

For Cray XT6 and Cray XT6m systems upgrading from Cray Linux Environment (CLE) 3.0:

Because CLE 3.0 was not a generally available release, this document is focused on the differences between the CLE 2.2 and CLE 3.1 releases. Much of the information in this overview was previously documented in *Cray Linux Environment (CLE) Software Release Overview*, S-2425-**30**; the `30` version was part of your CLE 3.0 release package.

The following content is new with CLE 3.1 and is flagged with this note: **(CLE 3.0 – 3.1 Change)**. If you are performing a new installation or are migrating from CLE 2.2, all of the information in this book is applicable and you can ignore these notes.