Cray XT™ System Software 2.1 Release Overview

S–2425–21

Contents

Page

Introduction [1] **1**

Naming of the Release 1

Emphasis for the CLE 2.1 Release 1

CLE 2.1 Release Package Description 2

Reader Comments 3

Software Enhancements [2] **5**

SLES 10 SP1 Upgrade 5

SLES 10 SP1 Upgrade and Configuration 6

SLES 10 SP1 Man Pages 6

EAL3 Evaluation 7

Security Auditing and Cray Audit Extensions for CNL 7

ALPS Interface to Security Auditing 9

Lustre File System Requirements for Cray Audit 9

Security Auditing Impact on System Performance 9

PAM Module to Log Failed Login Attempts 10

Lustre 1.6 Upgrade 11

Lustre Control for Cray XT Systems 12

Lustre 1.4 Compatibility 13

Lustre 1.6 Commands 13

Improvements in System Resiliency 14

New `ldump` and `lcrash` Utilities 14

Using `ldump` and `lcrash` Utilities for Node Memory Dump and Analysis 14

Boot-node Failover 15

Lustre Failover (Deferred implementation) 18

Preserving the System's Ability to Run Applications After Abnormal Terminations (Node Health) 19

Page

New `xthotbackup` Command 20
Comprehensive System Accounting (CSA) for CNL 20
CSA Compatibility and Limitations 21
PerfMon 2.3 Upgrade for CNL 21
NUMA Kernel 22
Portals Enhancements 22
Huge Pages for CNL 22
Huge Pages on Cray XT3, Cray XT4, and Cray XT5 Systems 22
Huge Pages Requirements and Limitations 24
Huge Pages User-visible Functionality 25
Huge Pages for Cray X2 Nodes in Cray $XT5_h$ Systems 25
Memory Affinity for CNL 26
CPU Affinity for CNL 29
New XTAdmin Database `segment` Table 31
Cray Data Virtualization Service (Cray DVS) for CNL 32
Checkpoint/Restart (Deferred implementation) 33
Additional Enhancements to System Administration and System Maintenance 33
General System Administration Enhancements 34
ALPS Node Attribute Resync 34
`xtprocadmin` Options to Control RCA Events 34
Service Node Boot Format 35
Release Upgrade, Configuration, and Installation Enhancements 37
Shared Root Configuration Tools 37
New `XTinstall` Boot Parameters Options 38
New `XTinstall --noforcefsck` Option 38
Configuring the Contents of the CNL Boot Image 38
Changing the Default HSN IP Address 39
Changes to the `XTinstall.conf` Installation Configuration File 39
File System and I/O Enhancements 39
Lustre Health Checker 39

Page

Lustre Option for Panic on LBUG for CNL 40
NFS version 4 Support 41
Virtual Channel 2 (VC2) Supported 41
PCI Express Supported 42
IP Routes for CNL 42
Additional New and Enhanced Commands 43
`apstat` Enhancements 44
Bugs Addressed Since the Last Release 45

Compatibilities and Differences [3] 47

Users Must Recompile Applications 47
Compilers Must Be Reinstalled 47
`apstat` Display Changes 47
Catamount-specific Commands Deprecated for CNL Systems 47
SUSE Man Page Packaging 48
PBS Professional No Longer Packaged as an Optional Product 48
PerfMon 2.3 Upgrade for CNL 49
Differences when Moving from SLES 9 SP2 to SLES 10 SP1 49
SLES 10 SP1 Changes to the User Interface 49
`g77` Command No Longer Supported 50
Installed `ksh` Version Changed to AT&T `ksh` Package 50
`ulimit` Stack Size Limit 51
PAM Configuration Files 51
SLES 10 SP1 Administrative Changes 51
SMW Device Name Conflicts 52
`ssh` Protocol Version 1 Disabled 52
PAM Login Failure Logging Differences 52
MySQL Version 5.0 52
`syslog-ng` Differences 53
TotalView Debugger Differences 53
Lustre 1.6 Backward Compatibility 53

Page

Additional System Management Compatibility Issues and Differences 53
Release 2.1 Upgrade-related and Configuration-related Changes 54
Supported Upgrade Path 54
SUSE Linux RPMs Loaded 54
Installation Time Required 55
Upgrading System Software Requires Service Database (SDB) Update 55
RSIP Configuration is Automated 56
General System Administration Differences 56
`e2fsprogs` Upgraded 56
STONITH Feature Default Changed to Disabled 56
SMW `xtbootimg` and `xtcli_boot` Command Usage Differences 56
Operational-related Changes 58
`ldump -r xt` Access Method Renamed 58
`ldump` Directories to Manually Copy Changed 58
Native IP Default 58
CRMS Renamed to HSS 59

Documentation [4] **61**

CrayPort Website 61
CrayDoc Documentation Delivery System 61
Accessing Product Documentation 62
Documentation Changes 63
SUSE Man Page Packaging 63
CrayDoc Requires GCC 4.1.1 63
File System and Storage Documentation has been Restructured 63
Books Provided with This Release 64
Third-party Books Provided with This Release 64
Other Related Documents Available 65
Changes to the Man Pages Document Set Since the UNICOS/lc 2.0 Release 65
New Cray Man Pages 65
New Third-party Man Pages 70

Page

Cray Glossary 74
Additional Documentation Resources 74
Ordering Documentation 75

Release Contents [5] **77**
Cray XT System Configurations 77
Software Requirements 77
Contents of the Release Package 79
Components for All Systems 79
Additional Components for Cray $XT5_h$ Systems 80
Licensing 80
Ordering Software 81

Customer Services [6] **83**
Technical Assistance with Software Problems 83
CrayPort 83
Training 84
Cray Public Website 85

Appendix A Package Differences **87**

Introduction [1]

This document provides an overview of the 2.1 general availability (GA) release for Cray XT systems running the Cray Linux Environment (CLE) 2.1 operating system. Throughout this document, only reference to (*Cray XT systems*) includes Cray XT3, Cray XT4, Cray XT5 and Cray XT5$_h$ systems, unless otherwise noted. For a complete description of hardware platforms supported with this release, see Section 5.1, page 77.

This document does **not** describe hardware, software, or installation of related products, such as Cray Programming Environment, or products not provided through Cray. For specific requirements for other Cray software products supported with this release, see Section 5.2, page 77.

1.1 Naming of the Release

The UNICOS/lc operating system was renamed Cray Linux Environment (CLE). Documentation associated with this release may use the terms UNICOS/lc and CLE interchangeably. The transition to the CLE name will be complete in the next release.

1.2 Emphasis for the CLE 2.1 Release

The CLE 2.1 release provides the following key enhancements:

- **SUSE Linux Upgrade.** The operating system includes an upgrade to SUSE Linux Enterprise Server 10, Service Pack 1 (SLES 10 SP1).
- **EAL3 Evaluation.** An early release of the CLE 2.1 operating system is in evaluation for conformance with EAL3+ (Evaluation Assurance Level) requirements as defined by the National Information Assurance Partnership (NIAP).
- **Security Auditing.** The SLES 10 Linux security auditing utilities are available along with Cray enhancements to facilitate auditing across a large number of nodes.
- **Lustre file system upgrade.** The Lustre file system includes an upgrade to the Lustre 1.6 release from Sun Microsystems, Inc. For the CLE 2.1 GA release, this is Lustre version 1.6.5.

- **Non-Uniform Memory Access (NUMA) Kernel.** The kernel contains updates to include configuration changes required to enable Non-Uniform Memory Access (NUMA). This change minimizes traffic between sockets on Cray XT5 compute nodes by using socket-local memory whenever possible.
- **Comprehensive System Accounting (CSA).** The CSA open-source software is supported in this release.
- **Huge Pages.** The CNL operating system contains enhancements that support 2 MB huge pages for CNL applications.
- **System Resiliency Enhancements.** The system administration tools include new features that assist system administrators in recovering from system or node failures, including node dump and dump analysis commands, a hot backup utility, health monitoring features, and enhanced failover abilities.
- **Cray DVS.** The Cray Data Virtualization Service (Cray DVS) is a distributed network service that provides compute nodes with transparent access to NFS file systems that reside outside of the Cray XT network.

1.3 CLE 2.1 Release Package Description

CLE 2.1 release includes the base operating system software required to run on Cray XT systems. The software is based on SUSE Linux Enterprise Server 10, Service Pack 1 (SLES 10 SP1) which includes a Linux 2.6.16 kernel. Software is also provided for compute nodes to run either the CNL compute node operating system or the Catamount compute node operating system. CNL is supported on Cray XT compute nodes, including Cray X2 nodes.

Note: CLE 2.1 is the last release to include Catamount compute node software.

Note: The System Management Workstation (SMW) must be upgraded to the SMW 3.1 GA release running SLES 10 SP1 before upgrading to the CLE 2.1 release.

For detailed information about the release package, see Chapter 5, page 77. For detailed information about other prerequisites and the supported upgrade path, see Section 5.2, page 77.

1.4 Reader Comments

Contact us with any comments that will help us to improve the accuracy and usability of this document. Be sure to include the title and number of the document with your comments. We value your comments and will respond to them promptly. Contact us in any of the following ways:

E-mail:
docs@cray.com

Telephone (inside U.S., Canada):
1–800–950–2729 (Cray Customer Support Center)

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

Mail:
Customer Documentation
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Software Enhancements [2]

This chapter describes software enhancements provided with the Cray Linux Environment (CLE) 2.1 release. The intended audience is people who currently use or manage a Cray XT system.

For compatibility issues and differences that you should be aware of when installing or using this release, see Chapter 3, page 47.

Note: The *Limitations for CLE 2.1 GA* document includes a description of temporary limitations of this release. The *CLE 2.1 Release Errata* document includes installation and configuration changes identified after the installation documentation for this release was packaged and lists customer-filed critical and urgent bug reports closed with this release. A printed copy of these documents is included with the release package; they are also available from your Cray representative.

In addition to the documentation noted in each feature description, see Section 4.5, page 64 for a list of revised manuals provided with this release.

Note: Throughout this document, the use of the term *Catamount* refers to either the Catamount kernel running on compute nodes or the Catamount Virtual Node (CVN) capability that supports dual-core processing on compute nodes running the Catamount kernel.

2.1 SLES 10 SP1 Upgrade

The CLE 2.1 release for Cray XT systems incorporates the SUSE Linux Enterprise Server 10, Service Pack 1 (SLES 10 SP1). With this upgrade, both the user-level interfaces and the kernel running on the service nodes and CNL compute node kernel are now based on SLES 10. The kernel was initially upgraded to the Linux 2.6.16 kernel with the UNICOS/lc 2.0 release.

Except for the kernel, the UNICOS/lc 2.0 release was based on SLES 9, which was released in April of 2005. A significant number of updates and fixes have been generated since that time and are included in SLES 10 SP1, which was released in March 2007. If you currently have machines that run UNICOS/lc 2.0, you benefit by running the updated version of SUSE Linux.

SLES 10 SP1 includes a variety of new features and enhancements to many existing products. The following open-source and third-party websites may contain useful information regarding SLES 10 SP1:

- SUSE Linux Documentation — See http://www.novell.com/linux
- The Linux Documentation Project — See http://www.tldp.org

2.1.1 SLES 10 SP1 Upgrade and Configuration

A number of configuration changes are required in order to upgrade your software from the UNICOS/lc 2.0 release, which is based on SLES 9 Service Pack 2 (SLES 9 SP2), to the CLE 2.1 release and SLES 10 SP1. The process for upgrading involves steps to specifically update any specialized files in the shared root `/etc` directory. Cray has provided new utilities to archive and upgrade specialized files. These utilities and the associated documentation provide you with all the tools necessary to complete a successful upgrade. The new utilities are described further in Section 2.17.2.1, page 37. For complete upgrade documentation, see the *Cray XT System Software Installation and Configuration Guide*.

2.1.2 SLES 10 SP1 Man Pages

Updated SUSE Linux man pages are included with the CLE 2.1 release. For complete information regarding SLES 10 SP1 changes to specific commands, see the associated man pages. To access SUSE Linux man pages, use the `man` command on a login node.

2.2 EAL3 Evaluation

An early release of the CLE 2.1 operating system on Cray XT hardware is in evaluation for conformance to EAL3+ (Evaluation Assurance Level) by the National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Validation Scheme. The scope of a Common Criteria evaluation is not only the product itself but also the procedures and tools that Cray uses to develop and support it. This evaluation also includes conformance to Common Criteria ALC_FLR.1 for the defect handling procedures that Cray has in place. Additionally, this evaluation includes conformance to the Controlled Access Protection Profile (CAPP) developed by the Information Systems Security Organization within the National Security Agency. The evaluation is being carried out by Atsec Information Security, an independent NIAP-accredited Common Criteria Testing Lab. The evaluation does not include Catamount and Cray X2 compute nodes.

The Linux audit capabilities provided with SLES 10 SP1 provide a security auditing system that is compliant with CAPP to collect information about a variety of security events. For more information about Linux security auditing, see Section 2.3, page 7. For more information about CAPP, see the Security Audit section of the CAPP document at the following website: http://www.commoncriteriaportal.org/files/ppfiles/capp.pdf.

For complete information regarding the EAL3+ evaluation of CLE 2.1, contact your Cray representative.

2.3 Security Auditing and Cray Audit Extensions for CNL

The Linux security auditing utilities included with SLES 10 SP1 are fully implemented and available for use on Cray XT systems running the CLE 2.1 release. Security auditing is not supported on Cray X2 compute nodes.

Cray has developed extensions to the standard Linux audit utilities to facilitate administration and review of audit data across login and compute nodes. For more information about Linux audit, see SUSE Linux documentation at the following website: http://www.novell.com/linux.

Cray Audit is a set of Cray specific extensions to Linux security auditing. It provides a simple means for system administrators to configure auditing using the standard Linux audit features. When the Cray auditing enhancements are used, separate audit logs are generated for each compute and login node on the Cray XT system. New utilities simplify administration of auditing across many nodes. You can use these utilities to obtain a coherent picture of system activity across compute and login nodes on the Cray XT system.

Cray extensions to security auditing include the following changes:

- A new Linux audit configuration option, `cluster`, has been added to the `/etc/auditd.conf` file to enable clustered auditing. The `auditd` daemon has been enhanced to implement this new configuration option. The clustered configuration provides a mechanism to collect audit data on many nodes and store the data in a central location. The configuration script creates a separate directory for each node and names and manages the auditing log file in the same way as on a single-node system. This includes tracking log size, responding to size-related events, and rotating log files.
- A new command called `xtauditctl` is provided to distribute `auditctl` administrative commands to compute nodes on the system. This command traverses a list of all running compute nodes and invokes commands that deliver a signal to the audit daemons on each node. This utility allows an administrator to apply configuration changes without having to restart every node in the system.
- A new command named `xtaumerge` merges clustered audit logs into a single log file. When you use this tool to generate a single audit file, you can also use Linux audit tools to report on and analyze system-wide audit data. An additional benefit is that `xtaumerge` maintains compatibility with the Linux audit tools; you can move audit data to another Linux platform for analysis.

  Note: When you run `xtaumerge`, the resulting merged data stream loses one potentially useful piece of information: the node name of the node on which the event originated. In order to maintain compatibility with standard Linux utilities, the merged audit log does not include this information. Instead, you can use Linux audit utilities directly on the per-node log files to find a specific record if you require that level of information.

By default, the configuration setting for security auditing is off. To enable auditing, use standard Linux `auditctl` command options or edit the `/etc/auditd.conf` and `/etc/audit.rules` files and set the `-e` option to `1`. To enable Cray Audit extensions, set `cluster` to `yes` in the `/etc/auditd.conf` file on both the compute and login nodes. If you run an audit on a Cray XT system without Cray Audit extensions, auditing data from the various nodes collide and generate a corrupt audit log. Because of this, Cray Audit extensions are enabled by default.

Note: Because the boot node does not have access to the Lustre file system being used for the audit logs, you should configure the `/etc/auditd.conf` file on the boot node with `cluster` set to `no` and `log_file` set to the name of a local file.

For more information about Cray Audit, see *Cray XT System Management* and *Cray XT System Software Installation and Configuration Guide.*

2.3.1 ALPS Interface to Security Auditing

For Cray systems with Cray XT CNL compute nodes, the Application Level Placement Scheduler (ALPS) version 1.1 is enhanced to support security auditing functionality. ALPS instantiates an application on behalf of the user on specific compute nodes. After instantiating the application, the ALPS interface calls the auditing system to begin auditing the application. At job start and end, auditing system utilities write the audit record to the audit log.

2.3.2 Lustre File System Requirements for Cray Audit

Cray recommends that you configure auditing to use a Lustre file system to hold the audit log files. For login nodes, use the `xtopview` command to set `log_file = lustre_pathname` in the `/etc/auditd.conf` file. The audit system stores audit data in a directory tree structure that uses a naming scheme based on the directory name provided by the *lustre_pathname* string. For example, if you set `log_file` to `/lus/audit/audit.log`, the auditing system stores audit data in files named `/lus/audit/`*`node_specific_path`*`/audit.log`, where *node_specific_path* is a directory structure generated by Cray Audit.

Warning: If you run auditing on compute nodes without configuring the audit directory, audit records are written to the local ram-disk; writing these records may consume all your resources and cause data loss.

With the exception of the boot node, each audited node in the system must have access to the Lustre file system that contains the audit directory. Because each node has its own audit log file, sufficient space must be made available to store audit data. For more information on Lustre file system requirements for Cray Audit, see *Cray XT System Management*.

2.3.3 Security Auditing Impact on System Performance

With auditing turned off there is no performance impact from this feature. With auditing turned on, system performance is impacted. The more frequently the system is audited, the greater the impact there will be on performance. It is the responsibility of the auditor or administrator to determine what is the acceptable level of performance loss versus security benefits obtained by auditing. The administrator can decrease the impact by setting options and adjusting the frequency of audits.

The performance costs for running Linux audit and the associated Cray extensions vary greatly, depending on the site-defined audit event selection criteria. Auditing of judiciously chosen events, for example login or su attempts, do not impact overall system performance. However, auditing of frequently used system calls has a negative impact on system performance because each occurrence of an audited system call triggers a file system write operation to the audit log. It is the responsibility of the administrator or auditor to design the site security policy and configure auditing to minimize this impact.

2.4 PAM Module to Log Failed Login Attempts

The cray_faillog module is a pluggable authentication module (PAM) that, when configured, provides information to the user at login time about any failed login attempts since their last successful login. The module provides:

- Date and time of last successful login
- Date and time of last unsuccessful login
- Total number of unsuccessful logins since their last successful login

Note: To use this feature, you must install the pam_tally and configure the PAM module. The PAM configuration files provided with the CLE 2.1 (SLES 10 SP1) release allow you to manipulate a common set of configuration files that are active for all services. With the UNICOS/lc 2.0 (SLES 9 SP2) release, PAM configuration files forced you to configure PAM for each service (sshd, sudo, su, login).

The cray_faillog module requires an entry in the PAM common-auth and common-session files or an entry in the PAM auth section and an entry in the PAM session section of any PAM application configuration file. Use of the common files is typically preferable so that other applications such as su also report failed login information; for example:

```
boot-p2001:/etc/pam.d # su -
2 failed login attempts since last login.
Last failure Thu May  8 11:41:20 2008 from smw.
boot-p2001:~ #
```

When a login occurs, cray_faillog in the auth section saves the pam_tally counter information in a per-user temporary file in the directory specified in the /etc/xtfaillog.conf file. By default this is set to /var/xtfaillog; optionally, you can edit the /etc/xt.faillog.conf file to reflect a root-writable directory for cray_faillog to store the temporary files.

Limitations:

- If a login attempt fails, `cray_faillog` in the `auth` section creates a temporary file; but because the login attempt failed, the `session` section is not called and, as a result, the temporary file is not removed. This is harmless because the file is overwritten at the next login attempt and removed at the next successful login.
- Logins that occur outside of the PAM infrastructure are not noted.
- Host names are truncated after 12 characters. This is a limitation in the underlying `faillog` recording.
- The `cray_faillog` module requires `pam_tally` to be configured.

  Note: To configure `cray_faillog` with the CLE 2.1 release, see *Cray XT System Management* and the *Cray XT System Software Installation and Configuration Guide* provided with the CLE 2.1 release package. For additional information on using the PAM module, see the `pam`(8) and `pam_tally`(8) man pages.

2.5 Lustre 1.6 Upgrade

The CLE 2.1 release includes support for the Lustre 1.6 release from Sun Microsystems, Inc. In addition to a significant number of bug fixes and other features, the 1.6 release of Lustre includes the following significant new features:

- Simplified configuration system: A new *management server (MGS)* centralizes configuration information for Lustre file systems. Cray-supplied configuration utilities interface with the MGS on the boot node.
- Automatic space-based object storage target (OST) assignment: Lustre 1.6 creates new files on the OST with the most available space, reducing errors caused by an OST that has run out of space. This is the default behavior—no action is required to activate this feature.
- Fragmentation measurement tools: System administrators can use these tools to monitor fragmentation and determine when fragmentation may be having a significant impact on performance.
- Adaptive time-outs: Adaptive time-outs enable Lustre to track actual remote procedure call (RPC) completion times and adjust time-out intervals accordingly. Adaptive time-outs are disabled by default. Support for this feature on Cray XT systems is deferred until a later CLE release.

- New `ldiskfs` allocator: Small file handling is updated to improve performance and reduce file system fragmentation.
- New `e2scan` fast file system scanner: Includes associated backup scripts. For more information, see the `e2scan`(1) man page.

For more information about Lustre 1.6 feature content and general Lustre information, see the *Lustre Operations Manual* and other Lustre-related publications available at the following websites:

- http://manual.lustre.org
- http://www.sun.com/software/products/lustre

In addition to the features included with the Lustre 1.6 release from Sun Microsystems, Inc., Cray also ships new utilities as part of a Lustre control feature. You use these utilities to scale Lustre 1.6 configuration management on Cray XT systems that have many nodes.

2.5.1 Lustre Control for Cray XT Systems

Cray provides new Lustre control utilities to work with the Lustre 1.6 configuration enhancements. These new utilities implement a centralized configuration file and provide a layer of abstraction to the standard Lustre configuration and mount utilities. Lustre control utilities provide you with a mechanism to easily transition from Lustre 1.4 to Lustre 1.6.

To configure Lustre file systems, you use the Cray Lustre control scripts that interface with Lustre's new Mount Configuration or MountConf system. You no longer use the `lconf` and `lmc` commands and the associated `xml` file.

MountConf involves new utilities (`mkfs.lustre` and `tunefs.lustre`) and two new lustre components, the Management Client (MGC) and the Management Server (MGS). The MGS compiles configuration information about all Lustre file systems running at a site. When using the Cray-supplied configuration utilities, you do not need to use these new commands and components directly.

The MGS can use its own disk for storage. However, the Cray upgrade utilities create a default configuration in which the MGS shares a disk (co-located) with a Meta Data Target (MDT) of a single file system.

For more information about using Lustre control utilities, see the *Cray XT System Software Installation and Configuration Guide.*

2.5.2 Lustre 1.4 Compatibility

You must upgrade all Lustre clients and servers running with your CLE 2.1 system and Lustre 1.6 at the same time. Sun Microsystems, Inc., states that Lustre 1.4.11 clients should work with Lustre 1.6.5 servers, however, Cray currently does not support this configuration because it has not been verified on a Cray XT system.

A Lustre file system that was formatted under Lustre 1.4 is compatible with either Lustre 1.4 or Lustre 1.6. You do not need to reformat Lustre file systems in order to upgrade. There are no known limitations or reduced functionality in using file systems formatted with Lustre 1.4 on a system running Lustre 1.6.

Warning: A Lustre file system formatted under CLE 2.1 running Lustre 1.6 does not work with UNICOS/lc 2.0 running Lustre 1.4.

2.5.3 Lustre 1.6 Commands

Lustre 1.6 contains modifications to existing Lustre system administration commands, the addition of new commands, and the deletion of some commands.

Modifications to the `lfs` and `lctl` commands:

`lfs` Supports new option formats in the `setstripe` argument. See the `lfs`(1) man page for more information.

`lctl` Supports adding/removing OSTs. See the `lctl`(8) man page for more information.

New commands from Sun Microsystems:

`mkfs.lustre`

Formats a disk for a Lustre file system

`tunefs.lustre`

Modifies the Lustre configuration information on a disk

New commands from Cray:

`generate_config.sh`(8)

Generates Lustre file system configuration

`lustre_control.sh`(8)

Controls Lustre file system configuration

Deleted commands from Sun Microsystems:

`lmc`(1)	Lustre configuration maker
`lconf`(8)	Lustre file system configuration utility

2.6 Improvements in System Resiliency

2.6.1 New `ldump` and `lcrash` Utilities

Note: Initial support for these utilities was provided with the UNICOS/lc 2.0.35 update package. Initial support for using the SeaStar network to dump node memory of Cray XT nodes to a file was provided in the UNICOS/lc 2.0.51 update package.

You use the new `ldump` and `lcrash` utilities to analyze the memory on any Cray XT service node, Cray XT CNL compute node, or Cray X2 compute node. The `ldump` command dumps node memory to a file. After `ldump` completes, you use the `lcrash` utility on the dump file generated by `ldump`.

For Cray XT nodes, Cray recommends running the `ldump` utility only if a node has panicked or is hung, or if a dump is requested by Cray. For Cray X2 compute nodes, if need be, `ldump` can be run manually; however, the preferred method is to use the `mzdumpsys` command.

2.6.2 Using `ldump` and `lcrash` Utilities for Node Memory Dump and Analysis

To select the desired access method for reading node memory, use the `ldump -r` *`access`* option. Valid access methods are `xt-ssi`, `xt-hsn`, and `x2-hss`. The default access method is `xt-ssi`. For backward compatibility, method `xt` is still accepted as an alias for method `xt-ssi`.

To dump Cray XT node memory, *access* takes the following form:

`method``[@`*`host`*`]`

Note: For the `ldump -r` *`access`* option, you can use the alias `xt` as shorthand for the `xt-ssi` method.

The `xt-ssi` method uses the L0 synchronous serial interface (SSI) channel to the Cray SeaStar to read node memory. The `xt-hsn` method utilizes a proxy that reads node memory by using the SeaStar network. The `xt-hsn` method is faster than the `xt-ssi` method, but there are situations where it will not work (for example, when the Cray SeaStar is not functional).

For the `xt-ssi` method, if you specify a host, `ldump` connects to the event router daemon on the specified host and uses it to read the node memory. If you do not specify a host, `ldump` connects to the L0 on which the node resides. For example, if you specify node c0-0c0s0n0, `ldump` connects to the event router daemon on c0-0c0s0.

For the `xt-hsn` method, you can specify *host* as *host* or as *smw*/*host*. *host* specifies where the SeaStar network proxy is running. If you do not specify *host*, a default of `boot` is used. *smw* specifies the host where an event router daemon is running. `ldump` sends events to the state manager through the event router on the specified host to determine the topology and, thus, the node-id (NID) mapping of the Cray XT system. If you do not specify *smw*, `smw` is used by default.

When using the `xt-hsn` method:

- The SeaStar network proxy is normally run on the boot node because the boot node is connected to the SeaStar network and the SMW has access to it. You must manually copy the SeaStar network proxy binary (`/opt/cray/bin/ldump_hsn_proxy`) to the boot node and start it. If `ldump_hsn_proxy` is not running, `ldump` issues an error when it tries to connect. You must run `ldump_hsn_proxy` as root.
- You cannot dump the node where `ldump_hsn_proxy` is running (usually the boot node) using the `xt-hsn` method.
- If the boot node has crashed and you want a full dump of the system, you can dump the boot node and reboot that node only. You can start the high-speed network (HSN) proxy on the boot node and use it to dump the rest of the system.

To dump Cray X2 compute node memory, the only valid access method is `x2-hss`. The `ldump` utility uses the JTAG interface of the specified host (blade controller) to read node memory. *access* takes the form *method@host*.

For additional information, see the `ldump`(8) and `lcrash`(8) man pages.

2.6.3 Boot-node Failover

When a secondary (backup) boot node is configured, boot-node failover occurs automatically when the primary boot node fails. This failover allows the system to keep running without a system interrupt.

The following services run on the boot node:

- NFS shared root (read-only)
- NFS persistent `/var` (read-write)
- Boot node daemon, `bnd`
- Hardware supervisory system (HSS) and system database (SDB) synchronization daemon, `xtdbsyncd`
- Application Level Placement Scheduler (ALPS) daemons `apbridge`, `apres`, and `apwatch`

When the primary boot node is booted, the backup boot node also begins to boot. However, the backup boot node makes a call to the `rca-helper` utility before it mounts its root file system, causing the backup boot node to be suspended until a primary boot-node failure event is detected.

The `rca-helper` daemon running on the backup boot node waits for a primary boot-node failure event, `ec_node_failed`. When the heartbeat of the primary boot node stops, the L0 begins the heartbeat checking algorithm to determine if the primary boot node has failed. When the L0 determines that the primary boot node has failed, it sends an `ec_heartbeat_stop` event to set the alert flag for the primary node. The primary boot node is halted through STONITH. Setting the `alert` flag on the node triggers the HSS state manager on the SMW to send out the `ec_node_failed event`.

When the `rca-helper` daemon running on the backup boot node receives an `ec_node_failed` event alerting it that the primary boot node has failed, it allows the boot process of the backup boot node to continue. Any remaining boot actions occur on the backup boot node. Booting of the backup boot node takes approximately two minutes.

Each service node runs a failover manager daemon (fomd). When each service node's fomd receives the `ec_node_failed` event, it takes appropriate action. The fomd process accesses the SDB to get configuration information and remaps the virtual IP address associated with boot-node failover to point to the backup boot node. Some services start up; however, you must start some services manually.

Note: The boot-node failover feature does not provide a failback capability.

The purpose of this implementation of boot-node failover is to ensure that the system continues running, not to guarantee that every job will continue running. Therefore, note the following:

- During the time the primary boot node has failed, any service node that tries to access its root file system will be I/O blocked until the backup boot node is online, at which time the request will be satisfied and the operation will resume. In general, this means if an application is running on a service node, it can continue to run if the application is in memory and does not need to access disk. If it attempts to access disk for any reason, it will be blocked until the backup boot node is online.
- Applications running on compute nodes are affected only if they cause a service node to access its root file system, in which case the service node function would be blocked until the backup boot node is online.

Caution: When enabling boot-node failover on a system that uses alternate system sets to boot multiple versions of CLE, you must disable the boot-node failover feature before booting a pre-2.1 CLE operating system. Failure to do so will cause boot RAID file system corruption.

REQUIREMENTS:

- The backup boot node must be in the same service class as the primary boot node.
- The backup boot node must have a Fibre Channel card connected to the boot RAID.

  Note: You must configure the backup boot node in the same zone as the primary boot node.
- You must ensure that the boot RAID host port can see the desired LUNs; for DDN, use the host port mapping; for LSI (Engenio), use SANshare in the SANtricity Storage Manager.
- The backup boot node also requires a Gigabit Ethernet card connected via a Gigabit Ethernet switch to the same port on the SMW as the primary boot node (typically port 4 of the SMW quad Ethernet card).
- The STONITH capability must be enabled on the blade of the primary boot node in order to use the boot-node failover feature. Due to this STONITH requirement, do **not** set up boot node failover on any blades that are providing Lustre file system services.

Configuration procedures are documented in *Cray XT System Software Installation and Configuration Guide* as part of the CLE installation and update procedures. *Cray XT System Management* includes the procedure to configure boot-node failover separately from the CLE installation or update process; it also includes the procedure to disable the boot-node failover feature.

2.6.4 Lustre Failover (Deferred implementation)

Lustre object storage server (OSS) and metadata server (MDS) failover is a service that switches to a standby server when the primary server fails or the service is temporarily shut down for maintenance. After cabling and configuring the file system, you have the option of creating a failover on a Lustre node to a backup node. The CLE 2.1 release provides a mechanism so that you can configure Lustre to invoke a failover automatically. With this new feature, when a Lustre node fails, it notifies a Lustre proxy service that runs the failover commands without administrator intervention.

> **Note:** Implementing failover requires specific cabling and configuration of the storage devices and the Lustre file system.

The automatic Lustre failover framework from Cray includes the `xt-lustre-proxy` process, the SDB, and a set of database utilities. The `xt-lustre-proxy` daemon is a Linux process that is responsible for Lustre automatic failover in the event of a Lustre service failure. The SDB stores information about Lustre configuration and failover states. Console and log messages record the status of a failover.

Using the Lustre failover utilities, it is possible to reverse the process and *failback* to the primary node. To failback, you stop failover services on the backup node and restore services on the primary node. If the failover was automatic, you will also have to reset a state in the SDB.

To configure Lustre automatic failover, you set up three new SDB tables: `lustre_failover`, `lustre_service`, and `filesystem`. Each SDB table can be populated or backed up by executing the related Lustre database table utility. Each utility uses a formatted data file to generate database entries.

For complete Lustre failover and failback documentation, see *Managing Lustre on a Cray XT System*. For more information on Lustre automatic failover commands and utilities, see the following man pages: `xt-lustre-proxy`(8), `xtlusfoadmin`(8), `xtlustrefailover2db`(8), `xtdb2lustrefailover`(8), `xtlustreserv2db`(8), `xtdb2lustreserv`(8), `xtfilesys2db`(8), and `xtdb2filesys`(8).

2.6.5 Preserving the System's Ability to Run Applications After Abnormal Terminations (Node Health)

When the `aprun` command cannot complete in a controlled manner (for example, when a user executes a `kill -9` or an `apkill -9` command), then node health software attempts to preserve the system's ability to run applications. These uncontrolled terminations cause the `apsys` daemon to invoke the `xtcleanup_after` node health-check script. This feature is not supported on Cray X2 or Catamount compute nodes.

As it is supplied, the `xtcleanup_after` script invokes the `xtok2` command on all compute nodes that were running that application. The `xtok2` command queries the health of the compute node ALPS daemon, `apinit`, on each compute node. If the `apinit` daemon fails to respond or reports that the `apid` is still present (it should have terminated by now), the state of the associated compute nodes is changed by `xtok2` to `ADMINDOWN`, and ALPS is reset to remove the nodes flagged as `ADMINDOWN` from the candidate list for future user applications. This information is logged to the `/var/log/xtoklog` file; the `/var/log/xtoklog` file is present on every login node (every node from which applications are launched). This prevents new applications from being placed on nodes that, for one reason or another, fail to properly run the `apinit` service.

The Cray Linux Environment (CLE) installation process installs and enables this node health software by default, and it is fully automatic; there is no need for you to issue any commands.

When a user application terminates in a controlled manner (for example, non-fatal `apkill` commands or a `Ctrl-c`), the `xtcleanup_after` node health-check script does not run. A future version of ALPS will allow a site to configure this behavior to be on or off.

> **Note:** The `xtcleanup_after` script is intended as a user exit. With this release, although it is a site-modifiable step in the cleanup process, when you apply each CLE software update, you will need to update the `xtcleanup_after` script manually to carry forward your local changes on your new CLE software version. Both your current `xtcleanup_after` file and the new update's default file are located in `/opt/xt-service/`*`filename`*. A future CLE release will provide a version of the node health software that will be more easily customized.

For more information about using this feature, see the `xtcleanup_after`(8) and `xtok2`(8) man pages and *Cray XT System Management*.

2.6.6 New xthotbackup Command

The xthotbackup command creates a bootable backup. The system set labels in /etc/sysset.conf define disk partitions for backup and source system sets which are used by xthotbackup to generate the appropriate dump and restore commands. The entire contents of the disk partitions defined in a source system set are copied to the corresponding disk partitions in the backup system set. The backup and source system sets must be configured with identical partitions. The disk partitions in the backup system set are formatted prior to the dump and restore commands. The --configfile option uses a configuration file other than the default /etc/sysset.conf file. The -b option changes key files within the backup to make it bootable, for example, /etc/fstab in the boot node and shared root file systems.

For more information about using the new xthotbackup command, see the xthotbackup(8) man page included on the release media.

2.7 Comprehensive System Accounting (CSA) for CNL

Comprehensive System Accounting (CSA) open-source software includes changes to the Linux kernel so that the CSA can collect more types of system resource usage data than under standard Fourth Berkeley Software Distribution (BSD) process accounting. CSA software also contains interfaces for the Linux process aggregates (paggs) and jobs software packages. The CSA software package includes accounting utilities that perform standard types of system accounting processing on the CSA generated accounting files. CSA, with Cray modifications, provides:

- Project accounting capabilities, which provide a way to charge computer system resources to specific projects
- An interface with various other job management systems in use at Cray sites
- A data management system for collecting and reporting accounting data
- An interface that you use to create the project account and user account databases, and to later modify them, as needed.
- (Deferred implementation) 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.

Jobs are created on the system using either a batch job entry system (when such a system is used to launch jobs) or by the PAM job module for interactive sessions.

Complete features and capabilities of CSA are described in the `csa`(8) and `intro_csa`(8) man pages.

CSA RPMs (`csa`, `projdb`, `job`, and `account`) are installed at system installation. In addition, administrators must configure CSA for their system. Complete configuration and management tasks are provided in the *Cray XT System Management* manual.

A number of new Cray supplied and third-party man pages are included to support this feature. For a complete list of new man pages, see Section 4.8, page 65.

2.7.1 CSA Compatibility and Limitations

Systems that run CNL on compute nodes can use both Linux accounting and CSA accounting packages to produce job and system accounting reports; however, any CSA accounting files must be processed by CSA commands, and any BSD accounting files must be processed by BSD commands.

CSA runs **only** on login nodes and compute nodes. The SMW, boot node, SDB node, Luster MDS nodes, and Lustre OST nodes do not support CSA.

CSA requires the following work load manager release levels: PBS Professional 9.2 or later; Moab / TORQUE torque-2.4.0-snap.200809251409 or later. Compute node project accounting for applications submitted through workload managers (for example, PBS Professional) depends on the ability of the workload manager to obtain and propagate the project ID to ALPS at job submission time. If the workload manager does not support the ability to obtain and propagate the project ID to ALPS at job submission, the project ID for processes running on compute nodes is not available.

The open-source CSA `ja` command is not supported for Cray installations; there is no reasonable way to support it on the Cray XT architecture.

2.8 PerfMon 2.3 Upgrade for CNL

The PerfMon performance monitoring tool is a component in the Linux kernel. The version of PerfMon in the CLE kernel is upgraded to release 2.3. This upgrade impacts you only if you use CrayPat (Cray performance analysis tool) or PAPI (Performance Application Programming Interface). For more information about the impact of the upgraded PerfMon tool in CLE 2.1, see Section 3.7, page 49.

2.9 NUMA Kernel

In previous releases, Cray XT CNL compute node and service node kernels use symmetric multiprocessing (SMP) to manage memory accesses. With the CLE 2.1 release, Cray has modified the kernels to use Non-Uniform Memory Access (NUMA). The NUMA kernel is not supported on Cray X2 compute nodes.

The CLE 2.1 release supports Cray XT5 dual-socket compute nodes. Switching from SMP to NUMA minimizes traffic between sockets by using socket-local memory whenever possible. The CLE 2.1 release includes all kernel configuration changes required to enable NUMA. No action is required. The switch to NUMA memory architecture applies to all Cray XT systems. There are no known issues or impacts to end-users resulting from this change.

2.10 Portals Enhancements

Portals heap and memory allocations were modified to recognize the NUMA architecture. CLE 2.1 includes Portals software changes to increase concurrency in the processing of HSN traffic at the operating system level. These changes result in a decreased average latency but with a slightly higher minimum latency. Additionally, the number of messages that can be processed per second has increased. The effects of these Portals changes varies depending on the characteristics of your applications.

2.11 Huge Pages for CNL

For Cray systems with CNL compute nodes, the ALPS version 1.1 now supports huge pages functionality.

The CLE 2.1 installation process installs and enables the huge pages functionality and the `hugetlbfs` library by default; you do not need to take any action.

2.11.1 Huge Pages on Cray XT3, Cray XT4, and Cray XT5 Systems

Cray XT systems have been enhanced to provide support of 2 MB huge pages for CNL applications. Previous versions of CNL supported only 4 KB base pages. For applications that use a large amount of virtual memory, 4 KB pages can put a heavy load on the virtual memory subsystem. Huge pages can provide a significant performance increase under heavy virtual memory load.

The 4 KB base pages remain the default. Users specify huge pages by doing the following.

1. Linking the huge pages library, `libhugetlbfs.a`, during the linking/loading phase. For example:

   ```
   % cc -c my_hugepages_app.c
   % cc -o my_hugepages_app my_hugepages_app.o -lhugetlbfs
   ```

   The `-lhugetlbfs` argument, as shown, is required. Do not use the separated form, `-l hugetlbfs`.

2. Setting the huge pages environment variable. For example, if you are using `csh`:

   ```
   % setenv HUGETLB_MORECORE yes
   ```

   If this environment variable is not set or is set to `no`, the system uses 4 KB pages. Otherwise, the system uses huge pages.

3. Adding a huge pages suffix to the `aprun -m` *`size`* option:

`-m` *`size`*`h`	Requests *size* of huge pages to be allocated to each processing element. All nodes use as much huge page memory as they were able to allocate and 4 KB pages afterward.
`-m` *`size`*`hs`	Requires *size* of huge pages to be allocated to each PE. If the request cannot be satisfied, an error message is issued and `aprun` terminates the request.

   The following command requests 700 MB of huge pages per processing element, or 1400 MB per node on a dual-core system:

   ```
   % aprun -m700h -n2 -N2 ./my_hugepages_app
   ```

Users can run base-page and huge-page applications on the same machine at the same time. Base-page and huge-page applications can run in any order in succession on any groups of nodes.

For example, users can run a base-page application, a huge-page application, and then a base-page application (the following example assumes that you are using csh):

```
% aprun -n64 -N2 ./my_4kbpage_app

% setenv HUGETLB_MORECORE yes

% aprun -m700h -n64 -N2 ./my_hugepages_app

% aprun -n64 -N2 ./my_4kbpage_app
```

2.11.1.1 Huge Pages Requirements and Limitations

Only the heap is placed on huge pages. All other program segments (code, initialized data, BSS data, and the stack) are on 4 KB pages. The heap is not placed on huge pages if the program uses an allocation function other than `glibc malloc()` or overrides the `glibc malloc morecore()` function.

The memory available for huge pages is less than the total memory on the node. Users must leave enough memory for CNL and I/O buffers. Also, because of memory fragmentation, less memory is available for huge pages after a node has run other jobs.

There is no guaranteed amount of huge page memory available to an application. Cray recommends that users not request more than the following values for `total_node_memory`:

2GB of memory on the node

1000MB is available for huge pages

4GB of memory on the node

3000MB is available for huge pages

8GB of memory on the node

6400MB is available for huge pages

Memory allocated into huge pages whether used by the application or not is unavailable for I/O. Less memory for I/O buffers may result in performance degradation.

If users do not include the `-m`*`size`*`hs` option on the `aprun` command and not enough huge pages are available, run times may be inconsistent.

2.11.1.2 Huge Pages User-visible Functionality

The `-lhugetlbfs` argument is required on the compiler driver command line during the linking/loading phase. See the `CC`(1), `cc`(1), and `ftn`(1) man pages.

The `h` or `hs` suffix to the `aprun -m` *`size`* option is required. See the `aprun`(1) man page.

For more information about how to use huge pages, see *Cray XT Programming Environment User's Guide.*

2.11.2 Huge Pages for Cray X2 Nodes in Cray $XT5_h$ Systems

Although Cray X2 node software uses Linux `hugetlbfs`, several differences exist in the huge page implementation for Cray X2 nodes in Cray $XT5_h$ systems.

- For Cray X2 nodes, there is a dedicated memory pool for huge pages.
- There is no huge page swapping; therefore, there is no memory oversubscription.
- Huge pages are allocated through libdmapp, which provides:
  - Assistance with distributed memory application launch and startup
  - Setup for RTT/NTT for distributed memory applications

Huge page sizes are set using kernel command-line arguments, usually specified as an argument to the `mzinstall` command. For example, to specify a huge page size of 65536 KB, you would create the following `site-kernel-options` file:

```
smw:~ # vi site-kernel-options
"site-kernel-options" [New file]
hugepagecode=4
:wq
smw:~ #
```

You would then install the Release Master using the `site-kernel-options` file.

```
smw:~ # mzinstall --initial --release RM-n.n.nnnn --CLOfile site-kernel-options \
/opt/mazama/media/X2release/*.rpm
```

You can also use the `mzimage --edit` command to edit the image properties or kernel command-line options, including the `hugepagecode` value of an image.

```
smw:~ # mzimage --edit --image 5

Current property values are in parentheses:
  hit <Enter> to keep current value, <ctrl-D> to clear property
Image description (X2 Release Master):
Image path (file:///opt/mazama/PM/test103008/):
Image comment ():
Image name (test103008):
Image kernel arguments (ramdisk=0 rhash_entries=50000 thash_entries=50000 grad_timeout=625000000 \
gc_warning=250000 hugepagecode=4 runlevel=3 3 ):
Image huge mem page size in KB [ 65536 | 262144 | 1048576 | 4194304] (1048576): 1048576
```

The following values for `hugepagecode` are supported:

`hugepagecode=4` indicates 65536 KB

`hugepagecode=5` indicates 262144 KB

`hugepagecode=6` indicates 261,048,576 KB (or 1 GB)

`hugepagecode=7` indicates 4,194,304 KB (or 4 GB)

For more information about how to use huge pages for Cray X2 nodes in Cray $XT5_h$ systems, see *Cray XT5h Installation, Configuration, and Management Supplement*.

2.12 Memory Affinity for CNL

The Cray XT system now provides memory affinity optimization tools for Cray XT5 applications. Cray XT5 systems use dual-socket, quad-core compute nodes. Each compute node has two *NUMA nodes*. A NUMA node is a quad-core processor and its local memory. This feature is not supported on Cray X2 compute nodes.

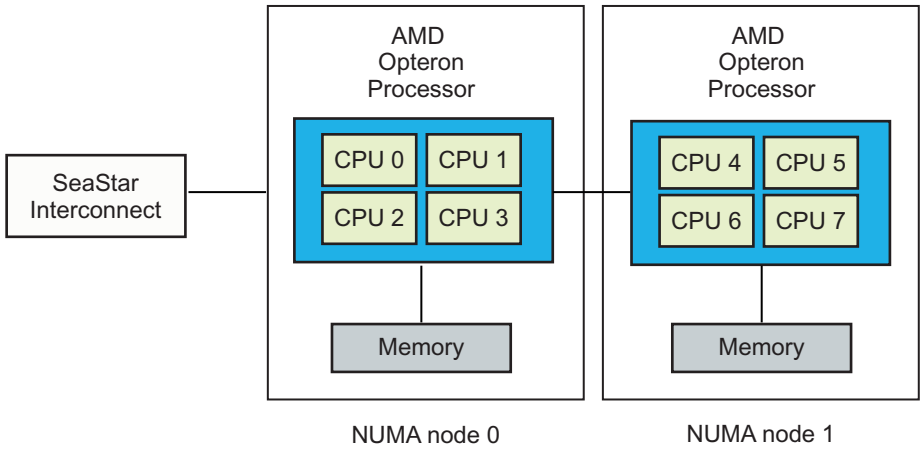

Figure 1. Memory Affinity

Applications can execute using one or both NUMA nodes of a Cray XT5 compute node. If an application is placed using only one NUMA node of a compute node, the other NUMA node is not used. In this case, the application processes are restricted to using local-NUMA-node memory. This memory usage policy is enforced by running the application processes within a *cpuset*. A cpuset is a process container that controls memory and CPU usage.

When an application is placed using both NUMA nodes of a compute node, the cpuset includes all node memory and CPUs. In this case, the application processes allocate local-NUMA-node memory first. If sufficient free local-NUMA-node memory is not available, the allocation may be satisfied using remote-NUMA-node memory.

Because Cray XT5 systems can run more tasks simultaneously, they can increase overall performance. However, off-NUMA-node memory references, such as a NUMA node 0 process accessing NUMA node 1 memory, can adversely affect performance. To give users run-time controls that can optimize memory references, Cray has added the following `aprun` memory affinity options:

`-S` *`pes_per_numa_node`*

Specifies the number of processing elements (PEs) to allocate per NUMA node; *pes_per_numa_node* can be `1`, `2`, `3`, or `4`.

`-sn` *`numa_nodes_per_node`*

Specifies the number of NUMA nodes per compute node to be allocated; *numa_nodes_per_node* can be `1` or `2`.

`-sl` *`list_of_numa_nodes`*

Specifies the NUMA node or nodes (comma separated or hyphen separated) to use for application placement; `-sl` *`list_of_numa_nodes`* can be:

```
-sl 0
-sl 1
-sl 0,1 or -sl 0-1
```

`-ss` Specifies strict memory containment per NUMA node. This option applies to Cray XT5 compute nodes. When `-ss` is specified, a PE can allocate only the memory local to its assigned NUMA node.

The default is to allow remote-NUMA-node memory allocation. For example, by default any PE running on NUMA node 0 can access NUMA node 1 memory. Developers can use this option to find out if restricting each PE's memory access to local-NUMA-node memory affects performance.

Developers can use these new `aprun` options for each element of a multiple-program, multiple-data (MPMD) application and can vary them with each MPMD element.

Only Cray XT5 compute nodes are considered for the application placement if any of the following are true:

- The `-sn` value is `2`
- The `-sl` list has more than one entry
- The `-sl` list is NUMA node `1` (Cray XT3 and Cray XT4 systems have single-NUMA-node compute nodes, defined as NUMA node `0`)
- The `-S` value along with a `-N` value requires two NUMA nodes (such as `-N 4 -S 2`)

Developers can use the `cnselect coremask.eq.255` command to create a list of Cray XT5 compute nodes and the `aprun -L` option or `qsub -l mppnodes` option to specify that list.

2.13 CPU Affinity for CNL

CPU affinity options let users bind a process to a particular CPU or a subset of CPUs on a node. These options apply to all Cray XT multicore compute nodes. CPU affinity is not supported on Cray X2 compute nodes.

CNL can dynamically distribute work by allowing PEs and threads to migrate from one CPU to another within a node. In some cases, moving processes from CPU to CPU increases cache misses and translation lookaside buffer (TLB) misses and therefore reduces performance. Also, there may be cases where an application runs faster by avoiding or targeting a particular CPU. The `aprun` CPU affinity options let users bind a process to a particular CPU or the CPUs on a NUMA node. These options apply to all Cray XT multicore compute nodes.

Applications are assigned to a cpuset and can run only on the CPUs specified by the cpuset. Also, applications can allocate memory only on memory defined by the cpuset. A cpuset can be a compute node (default) or, for Cray XT5 systems, a NUMA node.

The following CPU affinity options have been added to the aprun command:

`-cc` *cpu-list*| *keyword*

The *cpu-list* option requests CPU binding, where *cpu-list* is a comma-separated list of virtual CPU numbers and/or CPU ranges. As processes or threads are created, they are bound to the CPU in a *cpu-list* corresponding to the number of processes that have been created at that point. This option applies to all multicore Cray XT systems.

Developers can use the following *keyword* values:

- The cpu keyword (the default) binds each PE to a CPU within the assigned NUMA node. Developers do not have to indicate a specific CPU. The `-cc cpu` option is the typical use case for an MPI application.
- The numa_node keyword causes a PE to be constrained to the CPUs within the assigned NUMA node. CNL can migrate a PE among the CPUs in the assigned NUMA node but not off the assigned NUMA node. For example, if PE2 is assigned to NUMA node 0, CNL can migrate PE2 among CPUs 0-3 but not among CPUs 4-7.
- The none keyword allows PE migration within the assigned *cpuset*.

`-cp` *cpu_placement_file_name*

(Deferred implementation) The `-cp` *cpu_placement_file_name* option provides the name of a CPU binding placement file. This file must be located on a file system accessible from the compute nodes. The CPU placement file provides more extensive CPU binding instructions.

This option applies to all multicore Cray XT systems.

These new aprun options can be used for each element of an MPMD application and can vary with each MPMD element.

If developers do not use any of the new aprun options, this feature does not change the current placement and launch behavior for single NUMA nodes used as Cray XT compute nodes.

2.14 New XTAdmin Database segment Table

The XTAdmin database contains a segment table that supports the new memory affinity optimization tools for Cray XT5 applications and CPU affinity options for all Cray compute nodes with two or more NUMA nodes. The CPU affinity options apply to all Cray multicore compute nodes. See Section 2.12, page 26 and Section 2.13, page 29. The segment table is only supported by Cray systems running CNL.

The new segment table is similar to the attributes table but differs in that a node may have multiple segments associated with it; the attributes table continues to provide summary information for each node.

In order to address the application launch and placement requirements for compute nodes with two or more NUMA nodes, ALPS requires additional information that characterizes the intranode topology of the system. This data is stored in the segment table of the XTAdmin database and acquired by apbridge when ALPS is started, in much the same way that node attribute data is acquired.

The segment table contains the following fields:

- node_id is the node identifier that maps to the nodeid field of the attributes table and processor_id field of the processor table.
- socket_id contains a unique ordinal for each processor socket.
- die_id contains a unique ordinal for each processor die; with this release, die_id is 0 in the segment table and is otherwise unused (reserved for future use).
- coremask is the processor core mask. The coremask has a bit set for each core of a CPU. Quad core CPUs will have a value of 15 (binary 01111, hex 0xF).
- mempgs represents the amount of memory available, in Megabytes, to a single segment.

When you change the hardware on the machine, at the next system boot, you must invoke the SMW xthwinv utility to populate the attribute and segment tables. The /etc/attr.xthwinv file, which contains information to generate the hardware attributes for each node, populates the segment table. Like the attributes table, you must reinitialize the segment table at boot. Any changes that you make manually to the table do not persist at reboot. For additional information about the xthwinv command, see the xthwinv(8) man page.

The `/etc/xt.conf` file contains a new field, `SDBSEG`, which specifies the location of the `segment` table file.

The following service database commands enable a system administrator to update the `segment` table: `xtdb2segment`, which converts the data into an ASCII text file that can be edited, and `xtsegment2db`, which writes the data back into the database file. For more information, see the `xtdb2segment`(8) and `xtsegment2db`(8) man pages.

After manually updating the `segment` table, you can use the `apmgr resync` command to request ALPS to reevaluate the configuration node segment information and update its information.

Note: If ALPS or any portion of the feature fails in relation to segment scheduling, ALPS reverts to the standard scheduling procedure.

For the CLE 2.1 release, the Batch Application Scheduler Interface Layer (BASIL) specification is updated for use with CNL nodes and is being distributed to third-party batch vendors supporting Cray XT systems.

2.15 Cray Data Virtualization Service (Cray DVS) for CNL

The Cray Data Virtualization Service (Cray DVS) is a distributed network service that provides transparent access to file systems on the service I/O (SIO) nodes from the compute nodes. Cray DVS provides a service analogous to NFS. The key difference is Cray DVS provides I/O scalability to large numbers of nodes, far beyond the typical number of clients supported by a single NFS server.

Cray DVS configurations are limited to NFS file system access. Other file system formats are not presently supported in CLE 2.1. Cray DVS is not supported on Cray X2 compute nodes.

The CLE 2.1 release includes Cray DVS support for access to NFS file systems. This allows applications running on the compute nodes to read and write small data files to the user's home directory. A typical Cray DVS use case would be for a user to access small applications and input files on an NFS mounted `/home` directory using DVS. Most applications and all large data files would continue to be accessed using the Lustre file system.

For users who are migrating from Catamount to CNL, Cray DVS provides functionality similar to `yod` I/O from Catamount compute nodes to NFS-mounted file systems. Normal system calls such as `open()`, `read()`, and `write()` work without modification. Impact on compute node memory resources, as well as operating system jitter, is minimized in the Cray DVS configuration.

Administration of Cray DVS is very similar to configuring and mounting any Linux file system. DVS-specific options to the mount command enable client access to a network file system being projected by DVS server nodes. Cray DVS clients run on compute nodes and use the SeaStar network to communicate with a Cray DVS server running on an SIO node.

Caution: DVS servers must not be run on the same SIO nodes as Lustre servers.

The SIO node also runs an NFS client for the file system being projected. See the `mount`(8) and `dvs`(5) man pages for more information.

For information about installing and configuring Cray DVS, see *Cray XT System Software Installation and Configuration Guide*.

2.16 Checkpoint/Restart (Deferred implementation)

The CLE 2.1 release includes software to provide an initial implementation of Berkeley Lab Checkpoint/Restart (BLCR). Support for this feature is deferred until a later CLE release.

2.17 Additional Enhancements to System Administration and System Maintenance

This section describes additional software enhancements for Cray XT system administration and system maintenance. The following topic areas are discussed.

- General system administration enhancements
- Release upgrade, configuration, and installation enhancements
- File system and I/O enhancements

For compatibility issues and differences that you should be aware of when upgrading your Cray XT system to the CLE 2.1 release, see Section 3.10, page 53.

2.17.1 General System Administration Enhancements

2.17.1.1 ALPS Node Attribute Resync

The new `apmgr resync` command eliminates the need to restart ALPS after node attributes change. This feature is supported on Cray XT and Cray X2 compute nodes running CNL.

> **Note:** This feature was provided initially with the UNICOS/lc 2.0.36 update package.

Manual changes to node attributes and status can be reflected in the Application Level Placement Scheduler (ALPS) by using the `apmgr resync` command. The `apmgr resync` command requests ALPS to reevaluate the configuration and attribute information and update its information. For example, on a Cray XT system, after making manual changes to the SDB using the `xtprocadmin` command, use the `apmgr resync` command so that the changes take effect. For Cray X2 compute nodes, when you make changes using the `mzutil procadmin` command, the changes can be reflected in ALPS by using the `apmgr resync` command.

The ALPS `apres` event watcher restart daemon resides on the boot node and registers with the event router daemon to receive `ec_service_started` events. When the service is the SDB, ALPS updates its data to reflect the current values in the SDB. The `apres` daemon is invoked as part of the ALPS startup on the boot node. The `apres` daemon is not intended for direct use.

The `apmgr`(8) man page and the new `apres`(8) man page are provided with the release package.

2.17.1.2 `xtprocadmin` Options to Control RCA Events

When using multiple `xtprocadmin` commands for set operations, you can avoid sending an RCA update event after every command. This is especially important for Cray XT compute nodes running CNL because each time this RCA event is sent, it causes ALPS to be restarted.

To control sending of an RCA event, use the `xtprocadmin` [`-e` | `--noevent`] and [`-E` | `--forceevent`] options. By default, when a database change is made using the `xtprocadmin` command, a special RCA event is sent. Use the `-e` option to suppress sending an event or the `-E` option to force an event to be sent even if you did not change the database.

If your system is running ALPS and you have scripts that include multiple `xtprocadmin` commands, consider adding the `-e` option to each command to avoid restarting ALPS multiple times. Then use the `apmgr resync` command (or an `xtprocadmin -E` command) to notify ALPS of the changes via a single event.

The `xtprocadmin`(8) man page documents the [`-e` | `--noevent`] and [`-E` | `--forceevent`] options.

Note: This feature was provided initially with the UNICOS/lc 2.0.36 update package.

2.17.1.3 Service Node Boot Format

The Cray XT system CNL compute nodes and service nodes both use RAM disks for booting. Prior to the CLE 2.1 release, they used different formats and environments. A new feature is available in CLE 2.1 so that service nodes use the same `initramfs` format used by the CNL compute nodes. Although the format has changed, there are still two distinct and unique `initramfs` files (both called `initramfs.gz`); one for compute nodes and one for service nodes.

The default CLE 2.1 installation and build processes and tools generate service node images using the `initramfs` format. The `xtbootimg` command has been updated to process service node boot images in the same manner as CNL compute node images. The `xtbootimg` and `xtcli boot` commands now process service node boot images in the same manner as CNL compute node images. The *Cray XT System Software Installation and Configuration Guide* details the specific `xtbootimg` command and procedures that you need to generate a service node `initramfs` and add it to the boot image. See also Section 3.10.2.3, page 56 for specific `xtbootimg` and `xtcli boot` differences.

The `xtcli boot` command has been updated to support the `initramfs` format and RAM disk environments feature. For more information regarding this feature, see Section 2.17.1.3, page 35.

Note the following `xtcli_boot` usage differences:

- To boot CNL compute nodes or service nodes, you **must** specify the `loadfile_name` option.
- You can also specify the `-o` *`boot_type`* option with the `loadfile_name` option to specify a boot type. Valid types are `bootnode`, `service`, and `compute`.
- The `boot_type` option is deprecated but maintained for backward compatibility. A new NOTES section on the `xtcli_boot`(8) man page documents the `boot_type` strings to load file name mapping.
- The `boot_params` option is deprecated but maintained for backward compatibility. If you specify `boot_params`, however, and also specify the `loadfile_name` option, any `boot_params` are ignored.
- New examples 1 through 6 have been added showing the use of the `loadfile_name` option.

In addition, the `all` and `all_serv` arguments to the `boot_type` option (and examples showing the same) have been removed from the man page; these arguments have never been valid.

The `xtcli_boot`(8) man page provided with the SMW 3.1 release package documents these usage differences.

Several other system changes have been made to support this feature but should be transparent to users and system administrators. These changes include the following:

- Default boot automation files have been updated to support the `initramfs` load files in place of `initrd` file names.
- The `xtpackage` and `xtclone` commands have been generalized so that they now create a service node, `initramfs`, when you select the new `-s` option.
- New logic was added to the `init` boot script to handle all three cases; boot nodes, service nodes and compute nodes.
- The service node boot script, `boot.xt`, is simplified. Most of the logic has been added to the `initramfs` image. Backward compatibility is provided.
- The `xtdumpsys` command is updated to account for the new locations of some components.

- The `kernel-ss` RPM is now installed into the service node `initramfs`, the boot root and the shared root. The `xt-lustre-ss` and `xt-os` RPMs are now installed in the boot root and the shared root.
- The `keep_initrd` parameter in `XTinstall.conf` has been removed from the template. The `XTinstall` program generates an informational message suggesting this parameter be removed from your local `XTinstall.conf` file.

Since the service and compute nodes use the same RAM disk environment, the requirements of one may lead to increased size for the other. For example, the compute nodes require more utilities than service nodes for bootstrapping. This makes the `initramfs` slightly larger than necessary for service nodes.

2.17.2 Release Upgrade, Configuration, and Installation Enhancements

The following installation and configuration tools and enhancements are provided in the CLE 2.1 release. For more information on these features and upgrading to CLE 2.1, see *Cray XT System Software Installation and Configuration Guide*.

2.17.2.1 Shared Root Configuration Tools

Much of the effort required to configure a CLE operating system involves the specialization of various configuration files to create instances that are specific to individual nodes or classes of nodes. A new set of tools is provided with the CLE 2.1 release to assist system administrators in tracking, archiving and recreating these specialized files. These tools can also be used to better understand and manage configuration data. For more information about file specialization, see *Cray XT System Management*.

Changes to specialized shared root files in your current UNICOS/lc system have been tracked using the `xtopview` command and the Revision Control System (RCS). Four new utilities provide easy access to the RCS information stored by the `xtopview` command: `xtoprlog`, `xtopco`, `xtoprdump`, and `xtoparchive`.

You can use the `xtoprlog` command to review RCS revision information for a configuration file. To restore previous versions of a file, use the `xtopco` command. Using RCS information, combined with the `xtopview` specialization information, `xtoprdump` generates a comprehensive summary report of changes to configuration files within the shared root. The `xtoprdump` output can also be used as input to the `xtoparchive` command. The `xtoparchive` command allows you to archive and restore shared root configuration information, including the specialization information.

The scope of these tools is limited to identification and manipulation of `/etc` configuration data within the shared root. Configuration files on the boot root file system or on the SMW are not managed by these utilities.

System administrators can use these new utilities to archive and migrate specialized files as part of a system upgrade process. Using these utilities along with the upgrade processes documented in the *Cray XT System Software Installation and Configuration Guide*. You can easily upgrade the Cray XT system configuration to the CLE 2.1 release.

2.17.2.2 New `XTinstall` Boot Parameters Options

Two new options have been added to the `XTinstall` utility to support boot parameters files. The `--bootparameters` option specifies the parameters file to be used when making a service node boot image. The `--CNLbootparameters` option specifies the parameters file to be used when making a CNL node boot image. `XTinstall` modifies the specified parameters file as needed to reflect the configuration specified in `XTinstall.conf` and `sysset.conf`.

If these options are not specified for a release upgrade, `XTinstall` uses the parameters file within an existing boot image as the template. In the case of a new system installation, `XTinstall` modifies the default parameters file.

2.17.2.3 New `XTinstall --noforcefsck` Option

A new `--noforcefsck` option was added to the `XTinstall` utility. When this option is specified, `XTinstall` does not force an `fsck` of file systems during an upgrade. Use of the `--noforcefsck` option is not recommended for installation of a new system or for normal use during upgrades. However, if you experience problems during the upgrade and `XTinstall` fails, this option can be useful when restarting `XTinstall`.

2.17.2.4 Configuring the Contents of the CNL Boot Image

You can configure which optional RPMs are installed into the CNL boot image for your system. Several parameters are available in the `XTinstall.conf` file to control whether specific RPMs are included during installation or upgrade of your system software. An additional method is to add or remove specific RPMs by editing the `shell_bootimage_`*`label`*`.sh` command used when preparing boot images for CNL compute nodes. For a list of new parameters in `XTinstall.conf`, see Section 2.17.2.6, page 39.

2.17.2.5 Changing the Default HSN IP Address

The default HSN IP address can be changed during and installation or upgrade. Several parameters are available in the `XTinstall.conf` to change the default. The `XTinstall` program will propagate the change to appropriate system files. Additionally, the CNL and SNL parameters file in the boot image are updated to include the new IP address.

2.17.2.6 Changes to the `XTinstall.conf` Installation Configuration File

There are several new variables defined in the `XTinstall.conf` installation configuration file. The `XTinstall` program uses the following new variables to configure Realm Specific IP Addressing (RSIP), boot-node failover and optional CNL RPMs during system installation or upgrade.

`bootnode_failover`

`bootnode_failover_IPaddr`

`bootnode_failover_netmask`

`rsip_nodes`

`rsip_interfaces`

`CNL_audit`

`CNL_csa`

`CNL_dvs`

`CNL_rsip`

2.17.3 File System and I/O Enhancements

2.17.3.1 Lustre Health Checker

System administrators can use a new Lustre health check utility to determine the readiness of Lustre nodes and associated storage devices. After setting up a Lustre configuration and booting Lustre nodes, this script can be run to help you verify the configuration and locate potential problems.

You provide the `lustre_readiness.sh` script with the location of the `fs_defs` file that describes the current Lustre configuration. When run, the script parses the configuration to determine whether a Lustre mount can succeed. The `lustre_readiness.sh` provides the following information regarding the readiness of the Lustre file system configuration:

- Basic configuration check, reporting if Lustre was configured and installed correctly and if the correct Lustre modules are loaded
- Node status, indicating if Lustre nodes are up or down
- Storage status, indicating if the associated storage is attached or if there is a potential problem at the disk or controller level
- Application status, indicating if the system is ready for jobs to run on CNL or Catamount compute nodes

Note: For the script to work properly, the boot node needs passwordless root access to all Lustre nodes and the SMW.

2.17.3.2 Lustre Option for Panic on LBUG for CNL

A new Lustre configuration option, `panic_on_lbug`, is available to control Lustre behavior when processing a fatal file system error.

When a Lustre file system hits an unexpected data condition internally, it produces an `LBUG` error to guarantee overall file system data integrity. This renders the file system on the node inoperable. In some cases, an administrator wants the node to remain functional, for example, when there are dependencies such as a login node that has several other mounted file systems. However, there are also cases where the desired effect is for the `LBUG` to cause a node to panic. Compute nodes are good examples, because when this state is triggered by a Lustre or system problem, a compute node is essentially useless.

The new `panic_on_lbug` functionality is configured off by default on service nodes and on by default on compute nodes. Administrators can change the default behavior by modifying the `modprobe.local.conf` configuration file to include the following line:

```
options libcfs libcfs_panic_on_lbug=1
```

2.17.3.3 NFS version 4 Support

The Network File System (NFS) version 4 distributed file system protocol is now supported. NFS is enabled on service nodes but is not enabled on compute nodes.

2.17.3.4 Virtual Channel 2 (VC2) Supported

To improve performance of a system when it is under a heavy communication load, the second virtual channel class present within the SeaStar network (VC2) is now supported. The SeaStar Portals firmware in the CLE 2.1 release and the SMW 3.1 release software have been enhanced to support VC2. A system that is operating under a heavy communication load should see improved performance when using VC2. This feature is not applicable to Cray X2 compute nodes.

Table 1 gives an indication of potential performance improvements using HPC (High Performance Computing) Challenge benchmarks with VC2 on a dual-core system. Performance increases generally correlate with HSN bandwidth usage. Bandwidth intensive programs see a bigger improvement.

Table 1. Summary of HPC Challenge Benchmarks Run on 2772 Cores

HPC Challenge Benchmark	Improvement with VC2
PTRANS	17.5%
GUPS	5%
G-FFTE	7.9%
NR BW (NaturalRing Bandwidth)	10.2%
RR BW (RandomRing Bandwidth)	35.2%

By default, VC2 is **not** enabled in the SMW 3.1 release. If you have a Cray XT system with dual-core or quad-core processors, Cray strongly recommends that you enable VC2 using the `xtbounce` command as described here.

Caution: The `xtspider` and `xtnxn2` tests must be run to test the unused cache and confirm functionality before starting to use VC2. The versions of these diagnostics available with SMW 3.x and later will test VC2, even when VC2 is disabled for production.

The new SMW `xtbounce portals_algorithm` initialization file variable was added with the SMW 3.1 release. It is a numeric value indicating which algorithm the Portals firmware is to use. You may select which algorithm to use. Valid values are `0` and `1`. Algorithm `0` indicates that the Portals firmware should use virtual channel 0 (VC0) exclusively. Algorithm `1` indicates that the Portals firmware should utilize virtual channel 0 (VC0) and virtual channel 2 (VC2). By default, algorithm `0` is used.

Note: When Cray XMT processors are available on the system, only algorithm `0` is permitted.

The SMW `xtfwstat` command output now indicates the state of virtual channel 2 (VC2) usage within the firmware. This information is displayed as a diagnostic and a site-administration aid to confirm that VC2 is configured correctly.

Updated `xtbounce`(8) and `xtfwstat`(8) man pages are provided with the SMW 3.1 release package.

2.17.3.5 PCI Express Supported

Device drivers have been upgraded to support PCI Express (PCIe) cards on service nodes for connecting to external devices. The PCIe cards (GbE, FC4, and 10 GbE) are functional replacements for PCI-X cards. Login nodes connect to the external network using a GbE PCIe card; network nodes use a 10 GbE PCIe card; SIO nodes use the FC4 fiber channel for storage connectivity.

2.17.3.6 IP Routes for CNL

The `/etc/routes` file can now be edited in the CNL template image on the SMW to provide route entries for CNL nodes. This change was made to provide a simple mechanism for administrators to configure routing access from CNL compute nodes to login and network nodes using external IP destinations without having to traverse RSIP tunnels. This mechanism is not intended to be used for general-purpose routing of internal HSN IP traffic. It is intended only to provide IP routes for CNL nodes that need to reach external networks.

A new `/etc/routes` file is created in the CNL images and is examined during startup. Non-comment, non-blank lines are passed to the `route add` command. The empty template file contains comments describing the syntax.

2.18 Additional New and Enhanced Commands

In addition to the commands noted in this document that changed to support specific new features, the following new and enhanced commands are provided for this release.

- `xtnodestat`: The new `xtnodestat` command provides current job and node status summary information. This command combines the functionality of the `xtshowmesh` and `xtshowcabs` utilities and provides a simplified interface to ALPS and jobs running on CNL compute nodes. You must be running ALPS in order for `xtnodestat` to report job information.
- `xtpackage` and `xtclone`: The new `-s` option creates a service node image instead of a compute node image.
- `apmgr`:
  - The new `ping -g` option indicates success if the supplied *apid* is not present. If the `-a` option is used to specify an `apid`, then the `-g` option causes `apmgr ping` to return 0 if that *apid* is not present on the node. Similarly, if no *apid* is specified (that is, the `-a` option is not used), then the `-g` option causes *apmgr ping* to return 0 if the node that is pinged returns no `apid`.
  - The new `ping -o` option outputs the node number rather than the *apid* found on that node; if the *apid* is not found on the node, then no number is printed. If an error occurs when connecting to the node, then the node number (rather than an error message) is printed. When using the `-o` option, you must also use the `-a` *apid* option.
- `apmgrcleanup.sh`: A loop was added to `apmgrcleanup` so that when `apmgrcleanup` kills an application, it loops until that application is truly gone from all the nodes. A user exit allows system administrators some variability in operational responses to the condition(s) indicated by the persistence of an application after it has been terminated.
- The new `rca-helper -M` option prints the highest-numbered node ID.
- `aprun`: Added memory affinity, CPU affinity, and huge pages options
- `cnselect`: Added core mask values of 15 for Cray XT4 quad-core nodes and 255 for Cray XT5 dual-socket quad-core nodes
- `CC`, `cc`, `ftn`: Added an option for loading the huge pages library, added an environment variable that suppresses the INFO target information

2.18.1 apstat Enhancements

The apstat -n option now includes core status in its display:

```
% apstat -nv
NID Arch State HW Rv Pl PgSz     Avl   Conf Placed PEs Apids
 59   XT UP B   1  -  -   4K 1024000      0      0   0
 60   XT UP B   1  1  -   4K 1024000 512000      0   0
 61   XT UP B   1  1  -   4K 1024000 512000      0   0
 62   XT UP B   1  1  -   4K 1024000 512000      0   0
 63   XT UP B   1  1  -   4K 1024000 512000      0   0
 76   XT UP B   1  -  -   4K 1024000      0      0   0
 77   XT UP B   1  -  -   4K 1024000      0      0   0
 87   XT UP B   1  -  -   4K 1024000      0      0   0
 88   XT UP I   1  1  1   4K 1024000    256    256   1 156143
 89   XT UP I   1  1  1   4K 1024000    256    256   1 156143
 90   XT UP I   1  1  1   4K 1024000    256    256   1 156143
 91   XT UP I   1  1  1   4K 1024000    256    256   1 156143
 92   XT UP I   1  1  1   4K 1024000    256    256   1 156143
 93   XT UP I   1  -  -   4K 1024000      0      0   0
```

Where HW is the number of cores in the node, Rv is the number of cores held in a reservation, and Pl is the number of cores being used by an application.

The apstat -z option replaces the - in the HW, Rv, and Pl fields with a 0:

```
% apstat -nv -z
  NID Arch State HW Rv Pl PgSz     Avl   Conf Placed PEs Apids
   59   XT UP B   1  0  0   4K 1024000      0      0   0
   60   XT UP B   1  1  0   4K 1024000 512000      0   0
   61   XT UP B   1  1  0   4K 1024000 512000      0   0
   62   XT UP B   1  1  0   4K 1024000 512000      0   0
   63   XT UP B   1  1  0   4K 1024000 512000      0   0
   76   XT UP B   1  0  0   4K 1024000      0      0   0
   77   XT UP B   1  0  0   4K 1024000      0      0   0
   87   XT UP B   1  0  0   4K 1024000      0      0   0
   88   XT UP I   1  1  1   4K 1024000    256    256   1 156143
   89   XT UP I   1  1  1   4K 1024000    256    256   1 156143
   90   XT UP I   1  1  1   4K 1024000    256    256   1 156143
   91   XT UP I   1  1  1   4K 1024000    256    256   1 156143
   92   XT UP I   1  1  1   4K 1024000    256    256   1 156143
   93   XT UP I   1  0  0   4K 1024000      0      0   0
```

`apstat` also has two new verbosity levels, enhanced summary information, and a new batch system job ID has been added to the `r` and `av` displays. The node list is now shown as a rangelist in `-avv` and a full PE list is still available in `-avvv`. For the `apstat -n` option, the "`----`" notation that shows the number of cores with the letters `C` and `P` to indicate "confirmed" and "placed" is being replaced by simple counts.

2.19 Bugs Addressed Since the Last Release

The list of customer-filed critical and urgent bug reports closed with the CLE 2.1 release is included in the *CLE 2.1 Release Errata* provided with the release package.

Compatibilities and Differences [3]

This chapter describes compatibility issues and functionality changes to be aware of after upgrading to Cray Linux Environment (CLE) 2.1 from UNICOS/lc 2.0. For temporary limitations of this release and changes identified after the documentation for this release was packaged, see *CLE 2.1 Release Limitations*.

3.1 Users Must Recompile Applications

Because of changes made with the CLE 2.1 release, users must recompile applications when moving from the UNICOS/lc 2.0 release. Applications that run on Cray X2 compute nodes do not need to be recompiled.

If you are running the CLE 2.1 limited availability (LA) release with CNL compute nodes, you do not need to recompile when moving to the CLE 2.1 general availability (GA) release.

Caution: Not recompiling an application can result in undefined behavior and may cause the application to fail or hang or may cause a Cray XT node to fail.

3.2 Compilers Must Be Reinstalled

The PGI and PathScale compilers must be uninstalled and reinstalled after upgrading to CLE 2.1 from the UNICOS/lc 2.0 release.

3.3 `apstat` Display Changes

The `apstat` command has been changed to provide core status, enhanced summary information, additional verbosity levels, and a batch system job ID. Users or system administrators who have developed tools based on the previous output format will need to update their tools to use the new `apstat` output format. See Section 2.18.1, page 44 for details.

3.4 Catamount-specific Commands Deprecated for CNL Systems

The following Catamount-specific commands are deprecated for use on systems without Catamount compute nodes: `xtshowcabs`, `xtshowmesh`, `yod`, `core`, `sysio_init`, `ping_node`, `xtcpa`, `xtcpaconfig2db`, `xtdb2cpaconfig` and `xtgenacct`.

These commands will no longer be available in the next major CLE release.

You can use the new `xtnodestat` command to view the node status on a CNL compute node. This command provides the same functionality for the CNL operating system as the `xtshowmesh` and `xtshowcabs` commands provide for the Catamount operating system. For more information, see the `xtnodestat`(1) man page.

3.5 SUSE Man Page Packaging

SUSE Linux man pages are provided without modification. In previous releases, the implementation section of the man pages was modified to indicate if the man page applied to UNICOS/lc and, more specifically, to CNL or Catamount compute nodes. These modifications have not been made to the SUSE Linux Enterprise Server (SLES) 10 man pages provided with the CLE 2.1 release.

Some of the SUSE Linux man pages provide documentation for commands or files that do not apply to Cray XT systems and are not supported. For information regarding Cray specific differences in the behavior of certain Linux commands, see *Cray XT Programming Environment User's Guide*. In some cases, Cray XT specific implementation information is also provided in introductory man pages, for example `intro_mpi`(3).

3.6 PBS Professional No Longer Packaged as an Optional Product

The PBS Professional batch system software for Cray systems is now available directly from Altair Engineering Inc. PBS Professional software and the *PBS Pro Release Overview, Installation Guide, and Administration Addendum* (S-2438) are no longer available as an optional product with the operating system release package. You can order PBS Professional software by contacting the Cray Distribution Center as described in Section 5.5, page 81. For licensing, setup and configuration information, or to order and receive a license manager key and the PBS Professional software for a Cray XT system, including documentation, please contact Cray. For additional information on PBS Professional, see http://www.altair.com

Note: PBS Professional 9.2 requires FLEX*lm* licensing. In order to implement this software on a Cray XT system, network connectivity must exist between the license server and the SDB node. By default, the SDB node is connected to the Cray XT high-speed network (HSN) and cannot access an external FLEXlm server. Various options available to overcome this restriction are detailed in *Cray XT System Management*.

3.7 PerfMon 2.3 Upgrade for CNL

In order to resolve problems related to using CrayPat (Cray performance analysis tool) to capture hardware counter values to analyze the behavior of multithreaded applications such as those developed using OpenMP, the version of PerfMon in the CLE kernel has been upgraded to release 2.3. If you do not use CrayPat, this change may safely be ignored. If you are licensed to use CrayPat and/or have PAPI installed on your system, verify that you are running the following versions of the software.

CrayPat 4.3 (or later)

PAPI 3.5.99c (or later)

Because PerfMon is a kernel component, there is no separate utility to access it. Instead, a library included with CrayPat is linked into the user application and issues the required system calls. Any applications that access hardware counters on OpenMP programs must be re-instrumented using CrayPat version 4.3 or later, or re-linked using PAPI version 3.5.99c or later. Applications instrumented using earlier versions of CrayPat and PAPI do not function on the CLE 2.1 release of CNL, and if executed abort in an error state.

Similarly, the CrayPat `pat_build` utility now performs PerfMon version checking during the instrumentation process. Attempting to use CrayPat version 4.3 to instrument a program for hardware counter analysis or OpenMP experiments, when running on an earlier version of CNL, produces an error message, and the `pat_build` utility exits without building an instrumented executable.

3.8 Differences when Moving from SLES 9 SP2 to SLES 10 SP1

Due to the upgrade to SUSE Linux Enterprise Server 10 Service Pack 1 (SLES 10 SP1), the CLE 2.1 release includes a large number of changes that impact users and administrators. These changes are documented as part of the SUSE Linux release and are not included in this overview. The changes that are most likely to impact Cray users and administrators are highlighted here.

3.8.1 SLES 10 SP1 Changes to the User Interface

End-users may notice these changes following the upgrade to CLE 2.1 running SLES 10 SP1.

3.8.1.1 g77 Command No Longer Supported

The GNU compiler GCC 4.2.4 does not support the g77 command. The g77 module was removed effective with the upgrade from SLES 9 SP2 to SLES 10 SP1. (See Table 8, page 87.) Users attempting to use the g77 command receive the following message:

```
GCC 4.2.4 does not support f77. You may alias f77 to ftn, but GNU does not  \
provide a g77 interface with gcc 4.1.1. and there may be some problems with \
conflicts of entry point names.
```

GCC 4.2.4 does support the gfortran command for compiling Fortran 77 programs. Further, all newer Fortran compilers support programs written in Fortran 77. Cray recommends that users load the appropriate PrgEnv module and then use the ftn command to compile Fortran programs. Fortran 77 compilers from PGI and Pathscale are available but are not supported.

3.8.1.2 Installed ksh Version Changed to AT&T ksh Package

The version of ksh installed by SLES 9 SP2 is pdksh. The SLES 10 SP1 ksh is the AT&T version ksh package.

There are a number of differences between these shells, and you should verify that your scripts are not impacted by this change. In particular, two differences are of note:

- The default behavior of the echo built-in command with regard to special characters has changed. While the behavior of the -E and -e options (disable or enable the special meaning of special characters) is the same, the behavior of echo without an -E or -e option has changed. The SLES 9 SP2 ksh (pdksh) default behavior is to retain the special meaning; for example:

  ```
  # echo "hello \n world"
  hello
  world
  #
  ```

  The SLES 10 SP1 ksh (AT&T version) default behavior of echo is to ignore the special meaning. This is consistent with the behavior of the bash shell; for example:

  ```
  # echo "hello \n world"
  hello \n world
  #
  ```

To get the same behavior as with the SLES 9 SP2 `ksh` (retain the special meaning), include the `-e` option, for example:

```
# echo -e "hello \n world"
hello
world
#
```

- Also, the use of "local variables" has changed. In SLES 9 SP2 `ksh` (`pdksh`), there is an alias for `local` (`local='typeset'`). This alias is not available in SLES 10 SP1 `ksh` (AT&T version).

3.8.1.3 `ulimit` Stack Size Limit

With UNICOS/lc 2.0 and SLES 9, the default user environment was set up with an unlimited stack size resource limit. With SLES 10, the login environment now defaults to the kernel default stack size limit. To restore the old behavior, add the following to `/etc/profile.local`.

```
ulimit -Ss unlimited
```

3.8.1.4 PAM Configuration Files

The PAM configuration files provided with the CLE 2.1 (SLES 10 SP1) release allow you to manipulate a common set of configuration files that will be active for all services. With the UNICOS/lc 2.0 (SLES 9 SP2) release, PAM configuration files forced you to configure PAM for each service (`sshd`, `sudo`, `su`, `login`).

3.8.2 SLES 10 SP1 Administrative Changes

System administrators should note the following differences when moving from UNICOS/lc 2.0 release running SUSE Linux Enterprise Server 9 Service Pack 2 (SLES 9 SP2) to the CLE 2.1 release running SLES 10 SP1.

3.8.2.1 SMW Device Name Conflicts

The RoHS-compliant SMW has internal SATA disk drives which were known as `/dev/hde`, `/dev/hdg`, etc. on SLES 9. On an SMW running SLES 10 SP1 with SMW 3.1 software (or later), these SATA disk drives appear as `/dev/sda`, `/dev/sdb`, etc. This causes a conflict with the boot RAID disk devices which were `/dev/sda`, `/dev/sdb`, etc. on SLES 9. On SLES 10, the boot RAID disk devices have been shifted up so that the old `/dev/sda` boot RAID becomes `/dev/sdc` and the old boot RAID `/dev/sdb` becomes `/dev/sdd`, etc. This requires system administrators to update the `/etc/sysset.conf` file on the SMW before starting the CLE 2.1 upgrade installation on the Cray XT system. For more information about upgrading your system, see *Cray XT System Software Installation and Configuration Guide*.

Note: NonRoHS-compliant SMWs are not impacted because they have the internal IDE disk drives which appear to the SLES 10 SP1 operating system as `/dev/hda`, `/dev/hdc`, and so on.

3.8.2.2 `ssh` Protocol Version 1 Disabled

With SLES 10, the `ssh` protocol version 1 is enabled. The default setting in `/etc/ssh/sshd_config` under SLES 9 was `Protocol 2,1`, which accepts either version of the protocol. With SLES 10 SP1, this setting is `Protocol 2`. A site may change `/etc/ssh/sshd_config` to include an entry for `Protocol 2,1` rather than `Protocol 2` if the previous behavior is desired.

3.8.2.3 PAM Login Failure Logging Differences

The `reset` and `no_magic_root` options for `pam_tally.so` are not supported with SLES 10 SP1 and should not be used on CLE 2.1 systems. If you were using the `cray_faillog` feature with UNICOS/lc 2.0, ensure that the following configuration files do not include these options.

```
/etc/pam.d/common-auth
/etc/pam.d/common-account
/etc/pam.d/common-session
```

For more information, see *Cray XT System Software Installation and Configuration Guide*.

3.8.2.4 MySQL Version 5.0

MySQL has been upgraded to version 5.0. For more information about MySQL 5.0, see http://www.mysql.com/documentation.

3.8.2.5 `syslog-ng` Differences

SLES 10 uses `syslog-ng` to log system messages and deprecates the older `syslog` program. The Mazama log manager software included with the SMW 3.1 release also uses the `syslog-ng` daemon instead of the older `syslog` program. Cray's software installation program configures your system for `syslog-ng` but does not automatically translate older `syslog` configurations to newer `syslog-ng` configurations; you must manually translate `syslog` configurations to the new format.

For more information, see the *Cray XT System Software Installation and Configuration Guide* and the *Cray System Management Workstation (SMW) Software Release Overview*.

3.8.2.6 TotalView Debugger Differences

The TotalView debugger is now able to perform a debug session on a subset of an application. For this feature to be enabled it requires an enhancement to `aprun`. This enhancement is in this release.

3.9 Lustre 1.6 Backward Compatibility

A Lustre file system formatted under CLE 2.1 running Lustre 1.6 does not work with UNICOS/lc 2.0 running Lustre 1.4.

Lustre 1.6.5 user and group limits for Lustre quotas and the data tracking usage against these limits are not compatible with earlier Lustre versions. This data is no longer valid when accessed from a system running Lustre 1.4.11 or earlier or from a system running Lustre 1.6.3. If you use Lustre quotas under the CLE 2.1 GA release or later and then revert to an earlier release of the operating system, you must manually regenerate Lustre quota data.

3.10 Additional System Management Compatibility Issues and Differences

In addition to the feature information described in Chapter 2, page 5, system administrators should also note the following compatibility issues and differences when upgrading to the CLE 2.1 release.

For temporary limitations of this release and changes identified after the documentation for this release was packaged, see the *CLE 2.1 Release Limitations*.

3.10.1 Release 2.1 Upgrade-related and Configuration-related Changes

The following information is provided to help you prepare for installing the CLE 2.1 release. Installation procedures are included in the *Cray XT System Software Installation and Configuration Guide*, which is provided with the CLE 2.1 release package. For installation procedures specific to Cray X2 compute nodes, see *Cray XT5h Installation, Configuration, and Management Supplement*.

3.10.1.1 Supported Upgrade Path

The System Management Workstation (SMW) must be upgraded to the SMW 3.1 GA release running SLES 10 SP1 before upgrading to the CLE 2.1 release.

You must be running release version UNICOS/lc 2.0 GA or later on your Cray XT system in order to upgrade to the CLE 2.1 release.

Caution: Upgrading your Cray XT from operating system version 1.5 is not supported and requires an intermediate step involving UNICOS/lc 2.0. Contact your Cray representative if you plan to upgrade from version 1.5 to CLE 2.1. For additional information about compatibility issues to be aware of when upgrading your Cray XT system from the 1.5 release, see *Cray XT Series Software Release Overview* (S-2425-20) for the 2.0 release.

The CrayDoc documentation server has been updated to require GCC version 4.1.1 or later for new installations. For more information, see *CrayDoc Installation and Administration Guide* (S-2340-411).

Additionally, the PGI and PathScale compilers must be uninstalled and reinstalled after upgrading to SLES 10 SP1.

3.10.1.2 SUSE Linux RPMs Loaded

Because the upgrade from SLES 9 to SLES 10 SP1 is more like an initial installation, the time it takes to load the software is increased due to the number of RPMs that are loaded.

CLE 2.1 running SLES 10 includes a different set of RPMs from the set that was included with SLES 9 on UNICOS/lc 2.0. Several RPMs are replaced with functionally equivalent or newer RPMs. Others are no longer available under SLES 10. For a complete list, see Table 8, page 87.

3.10.1.3 Installation Time Required

The total time required to install or upgrade to the CLE 2.1 release is dependent on a large number of site-specific variables. However, system administrators should be aware that, when compared to the UNICOS/lc 2.0 release, the time required to install CLE 2.1 has increased significantly. The following steps are specific to CLE 2.1:

Note: As with past releases, much of the installation or upgrade requires a dedicated system. However, with the CLE 2.1 release, there are a number of pre-installation steps you can perform on a running system.

- Archiving specialized files in preparation for the upgrade. This step is required for the SLES 9 to SLES 10 upgrade. Estimate an additional 15 to 20 minutes. Note that this step does not require dedicated time.
- Converting Lustre configuration from 1.4 to 1.6. Estimate an additional 60 minutes plus time to analyze and confirm your changes. Part of this conversion can be completed on a running system, but some dedicated time and a reboot is required.
- Loading of additional SLES 10 RPMs. Estimate 60 to 90 minutes for the `XTinstall` program to install and configure the software.
- Comparing and converting specialized files. Estimate 45 to 50 minutes to run the conversion scripts. Additionally, you will need time to analyze the output and resolve differences. The time required for this analysis depends on your site-specific configuration.
- Reinstalling compilers. Estimate 30 minutes.
- Installing and configuring new optional features. Estimate several minutes to several hours, depending on which features are installed.

3.10.1.4 Upgrading System Software Requires Service Database (SDB) Update

If upgrading to CLE 2.1, you must update the SDB database schema. You must also upgrade from MySQL 4.0 to MySQL 5.0. The procedure to update the database schema and MySQL is documented in *Cray XT System Software Installation and Configuration Guide.*

3.10.1.5 RSIP Configuration is Automated

Realm-Specific Internet Protocol (RSIP) configuration and startup is now automated using the `XTinstall` program. For more information, see *Cray XT System Software Installation and Configuration Guide*.

3.10.2 General System Administration Differences

3.10.2.1 `e2fsprogs` Upgraded

The CLE 2.1 release includes an updated version of the `e2fsprogs` RPM from Sun Microsystems. This RPM will be updated in future updates to the CLE 2.1 as needed to address specific problems. The README file included with CLE update packages will indicate when the `e2fsprogs` RPM is updated.

3.10.2.2 STONITH Feature Default Changed to Disabled

The default setting of the STONITH feature changed; it is now disabled by default after it is installed.

> **Note:** Cray recommends that you keep the STONITH feature disabled until further notice; a Field Notice will be issued when this recommendation changes.

STONITH is implemented by using the `l0sysd` daemon on the L0 controller, which halts a node that has triggered its heartbeat alert event. Nodes that have triggered a heartbeat alert can continue to function. However, if STONITH is enabled and a heartbeat alert occurs, these nodes are halted. If the node is a compute node, the application dies; if the node is a Lustre I/O node, Lustre dies and the system goes down.

> **Note:** STONITH is a per blade setting and not a per node setting.

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

3.10.2.3 SMW `xtbootimg` and `xtcli_boot` Command Usage Differences

The SMW `xtbootimg` and `xtcli boot` commands process service node boot images in the same manner as CNL compute node images. The *Cray XT System Software Installation and Configuration Guide* details the specific `xtbootimg` command and procedures needed to generate a service node `initramfs` and add it to the boot image. For more information regarding this feature, see Section 2.17.1.3, page 35.

Note the following SMW `xtbootimg` command usage differences:

- You cannot invoke the `xtbootimg` command with only the `-c` action parameter and produce a working boot image. (The `-c` action parameter creates a boot image.)
- The `-L` *`path`* option is required for both the `CNL0.load` and `SNL0.load` files. In CLE 2.1, the service node boot image, like the CNL image, is an `initramfs` image. Prior to the CLE 2.1 release, the service node image was an `initrd` file while the CNL image was an `initramfs` file. Use the `-L` *`path`* option to specify the path to the service node load file. If you do not specify the `-L` *`path`* option, only the default `CVN0` load file will be created.
- The `-i` *`path`*, `-k` *`path`*, and `-P` *`path`* options are deprecated but are maintained for backward compatibility.

Note the following SMW `xtcli_boot` command usage differences:

- To boot CNL compute nodes or service nodes, you **must** specify the `loadfile_name` option.
- You can also specify the `-o` *`boot_type`* option with the `loadfile_name` option to specify a boot type. Valid types are `bootnode`, `service`, and `compute`.
- The `boot_type` option is deprecated but maintained for backward compatibility. A new NOTES section on the `xtcli_boot`(8) man page documents the `boot_type` strings to load file name mapping.
- The `boot_params` option is deprecated but maintained for backward compatibility. If you specify `boot_params`, however, and also specify the `loadfile_name` option, any `boot_params` are ignored.
- The `xtcli_boot`(8) man page includes new examples to show the use of the `loadfile_name` option.
- The `all` and `all_serv` arguments to the `boot_type` option (and examples showing the same) have been removed from the man page; these arguments have never been valid.

The `xtbootimg`(8) and `xtcli_boot`(8) man pages provided with the SMW 3.1 release package document these usage differences.

3.10.3 Operational-related Changes

3.10.3.1 `ldump -r xt` Access Method Renamed

The `ldump -r` *`access`* method `xt` is now named `xt-ssi`. To dump Cray XT node memory, the valid access methods you use for `ldump` are `xt-ssi` and `xt-hsn`. The default is `xt-ssi`. For backward compatibility, method `xt` is still accepted as an alias for method `xt-ssi`.

3.10.3.2 `ldump` Directories to Manually Copy Changed

When a dump is taken, you must manually copy the contents of a set of directories into the dump directory. With the CLE 2.1 release, the directories that you must manually copy have changed.

For UNICOS/lc 2.0, the directories were:

```
/opt/xt-images/kernel/boot
/opt/xt-images/kernel/lib/modules
/opt/xt-os/release/linux/ss-lustre26/boot
/opt/xt-os/release/linux/ss-lustre26/lib/modules
```

For CLE 2.1, the directories are:

```
/opt/xt-images/kernel/compute/boot
/opt/xt-images/kernel/compute/lib/modules
/opt/xt-images/kernel/service/boot
/opt/xt-images/kernel/service/lib/modules
```

where *kernel* corresponds to the image that was used to boot the system. It is important to copy the directories corresponding to the type of node that was dumped (service or compute). It is also important to keep the directories in a similar structure when being copied because some of the file names are identical. These directories contain the `System.map`, `vmlinux`, and kernel module files that are needed by `lcrash` to analyze the dump.

3.10.3.3 Native IP Default

Effective with the CLE 2.1 release, native IP (SSIP) is the only protocol supported for communication between Cray XT nodes. Previous releases supported IP over Portals (IPPO); this option is no longer available. The `bootimage_bootproto` parameter in `XTinstall.conf` must be set to `ssip`. The `XTinstall` utility enforces this requirement for the boot image protocol.

3.10.3.4 CRMS Renamed to HSS

As of the SMW 3.1 release, the Cray RAS and Management System (CRMS) was renamed to the Hardware Supervisory System (HSS). This change was previously announced in the *Cray XT Series Software Release Overview* provided with the Cray XT series 2.0 release package.

The HSS is the next generation Cray hardware and software management system. The HSS is an integrated system of hardware and software that monitors Cray XT system components and proactively manages the health of the system. It communicates with nodes and with the management processors over the private Ethernet network, and displays the system state to the administrator. The HSS is being built upon to support future Cray platforms.

Documentation [4]

This chapter describes the documentation that supports the Cray Linux Environment (CLE) 2.1 release.

4.1 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.

4.2 CrayDoc Documentation Delivery System

The CrayDoc documentation delivery system, along with product documentation, is provided with each Cray software release. The CrayDoc software runs on any operating system based on UNIX systems or systems like UNIX including Mac OS X, Linux, BSD, and anywhere else that Perl and Apache can be compiled from source code with freely available (GNU) tools. The installation and administration of the CrayDoc server software and Cray documentation are described in *CrayDoc Installation and Administration Guide.*

4.3 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 documentation that is associated to that particular product. CrayPort allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. You access CrayPort using the following URL:

crayport.cray.com

CrayDoc

CrayDoc is the Cray documentation delivery system. CrayDoc allows 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 as part of the third-party product.

4.4 Documentation Changes

If you are upgrading from UNICOS/lc 2.0, you should be aware of several changes to Cray supplied documentation.

4.4.1 SUSE Man Page Packaging

SUSE Linux man pages are provided without modification. In previous releases, the implementation section of the man pages was modified to indicate if the man page applied to UNICOS/lc and, more specifically, to CNL or Catamount compute nodes. These modifications have not been made to the SUSE Linux Enterprise Server (SLES) 10 man pages provided with the CLE 2.1 release.

Some of the SUSE Linux man pages provide documentation for commands or files that do not apply to Cray XT systems and are not supported. For information regarding Cray specific differences in the behavior of certain Linux commands, see *Cray XT Programming Environment User's Guide.* In some cases, Cray XT specific implementation information is also provided in introductory man pages, for example `intro_mpi`(3).

4.4.2 CrayDoc Requires GCC 4.1.1

The CrayDoc documentation server has been updated to require GCC version 4.1.1 or later for new installations. For more information, see *CrayDoc Installation and Administration Guide* (S-2340-411).

4.4.3 File System and Storage Documentation has been Restructured

Cray specific Lustre file system installation and administration documentation is now included in *Managing Lustre on a Cray XT System*. This information was previously included in *Cray XT System Management*. Lustre installation and configuration information is also documented in *Cray XT System Software Installation and Configuration Guide.* Boot RAID configuration is now documented in both *Cray System Management Workstation (SMW) Software Installation Guide* and *Cray XT System Software Installation and Configuration Guide.* Storage RAID documentation was removed from the *Cray XT System Software Installation and Configuration Guide.*

4.5 Books Provided with This Release

The books provided with this release are listed in Table 2, which also notes whether each book was updated. Most books are provided in HTML and all are provided in PDF.

Note: Two additional documents are provided with this release. The *Limitations for CLE 2.1 LA* includes a description of temporary limitations of this release. The *CLE 2.1 Release Errata* includes installation and configuration changes identified after the installation documentation for this release was packaged and lists customer-filed critical and urgent bug reports closed with this release. A printed copy of these documents is included with the release package; they are also available from your Cray representative. You should also contact your Cray representative about Cray XT system-related information addressed in Field Notices (FNs).

Table 2. Books Provided with This Release

Book Title	Number	Updated
Cray XT System Software Release Overview (this document)	S-2425–21	Yes
Cray XT System Software Installation and Configuration Guide	S-2444-21	Yes
Cray XT System Overview	S-2423-21	Yes
Cray XT System Management	S-2393-21	Yes
Managing Lustre on a Cray XT System	S-0010-21	Yes
Cray XT Programming Environment User's Guide	S-2396-21	Yes
CrayDoc Installation and Administration Guide	S-2340-411	Yes

4.6 Third-party Books Provided with This Release

Table 3. Third-party Books Provided with This Release

Book Title	Number	Updated
Lustre Operations Manual	S-6540-16	Yes

4.7 Other Related Documents Available

The following publications contain additional information that may be helpful in setting up your Cray XT system; they are not provided with this release but are supplied with other products purchased from Cray. They can also be ordered on a CrayDoc CD from the Cray Software Distribution Center (see Section 4.11, page 75). Release overviews and installation guides can also be ordered from Cray in printed form.

Table 4. Other Related Documents Available

Book Title	Number
Cray System Management Workstation (SMW) Software Release Overview	S-2482-31
Cray System Management Workstation (SMW) Software Installation Guide	S-2480-31
Cray Programming Environments Installation Guide	S-2465-30
Cray Programming Environment Releases Overview and Installation Guide	S-5212-60
Cray XT5h System Overview	S-2472-10
Cray XT5h Installation, Configuration, and Management Supplement	S-2477-10

4.8 Changes to the Man Pages Document Set Since the UNICOS/lc 2.0 Release

4.8.1 New Cray Man Pages

The following Cray man pages are new with this release:

`apres`(8) ALPS database event watcher restart daemon

`dvs`(5) Cray DVS fstab format and options

`ldump`(8) and `lcrash`(8)
Utilities for node memory dump and analysis

`xtcleanup_after`(8) and `xtok2`(8)
Utilities for node health-check

`xtdb2segment`(8) and `xtsegment2db`(8)
Database utilities for segment table

`xthotbackup`(8)

Creates a bootable backup

`xtoparchive`(8)

Performs archive operations on shared root files

`xtoprdump`(8) Lists shared root file specification and version information

`xtoprlog`(8) Provides RCS log information about shared root files

`xtopco`(8) Checks out RCS versions of shared root files

`xtnodestat`(1)

Provides current job and node status summary information

`xt-lustre-proxy`(8)
`xtlusfoadmin`(8)
`xtlustrefailover2db`(8)
`xtdb2lustrefailover`(8)
`xtlustreserv2db`(8)
`xtdb2lustreserv`(8)
`xtfilesys2db`(8), and
`xtdb2filesys`(8)

Lustre automatic failover commands and utilities

General CSA man pages added for the Cray CSA implementation:

`intro_csa`(8)

Describes changes to standard CSA pages for the Cray implementation of CSA

`csanodeacct`(8)

Sets up a transfer of node `pacct` files to a common file system

`csanodemerg`(8)

Merges all individual node `pacct` files to a common system `pacct` file

`csanodesum`(8)

Reads the `pacct` file, verifies the contents, creates application account summary records, and writes to the `pacctsum` file

`account`(1)

Shows accounts or changes an active account number

`appacct`(3)

Initiates application termination accounting on a compute node

CSA Job man pages added for the Cray CSA implementation:

`job_getacctid`(3)

Gets the account ID for a process

`job_getapid`(3)

Gets the application ID for a process

`job_getapjid`(3)

Gets the job ID for an application

`job_getpjid`(3)

Gets the parent job ID for a job

`job_setacctid`(3)

Sets the account ID for a process

`job_setapid`(3)

Sets the application ID for a process

`job_setpjid`(3)

Sets the parent job ID for a job

CSA Project database man pages added for the Cray CSA implementation:

`projdb`(8)

Creates and updates system project database

`db_add_project`(3)

Adds a project to the database `projects_accounts` table of the project database

`db_add_user`(3)

Adds a user account entry to the project database `user_accounts` table

`db_connect`(3)

Connects to a database

`db_disconnect`(3)

Disconnects from a database

`db_get_proj_acct`(3)

Gets the project account from the project database

`db_get_proj_name`(3)

Gets the project name from the project database

`db_get_user_accts`(3)

Fetches user accounts from the project database

`db_has_table`(3)

Verifies if a table exists in a database

`db_print_table`(3)

Formats and prints the contents of a database table

`db_truncate_table`(3)

Truncates a database table

4.8.2 New Third-party Man Pages

The following third-party man pages are provided for the Cray CSA implementation:

`csa`(8)

Provides an overview of the Comprehensive System Accounting (CSA)

`acctdisk`(8)

Produces consolidated accounting records

`acctdusg`(8)

Reads standard input and computes the disk resource consumption

`csaaddc`(8)

Combines `cacct` records

`csabuild`(8)

Organizes accounting records into job records

`csachargefee`(8)

Charges a fee to a user

`csackpacct`(8)

Checks the size of the process accounting file

`csacms`(8)

Summarizes command usage from per-process accounting records

`csacon`(8)

Condenses records from the `sorted pacct` file

`csacrep`(8)

Reports on consolidated accounting data

`csaedit`(8)

Displays and edits the accounting information

`csagetconfig`(8)

Searches the accounting configuration file for the specified argument

`csajrep`(8)

Prints a job report from the `sorted pacct` file

`csaperiod`(8)

Runs periodic accounting

`csarecy`(8)

Recycles unfinished job records into the next accounting run

`csarun`(8)

Processes the daily accounting files

`csaswitch`(8)

Checks the status of, enables, or disables the different types of Comprehensive System Accounting (CSA), and switches accounting files for maintainability

`csaverify`(8)

Verifies that the accounting records are valid

`dodisk`(8)

Performs disk accounting

`lastlogin`(8)

Records the last date on which each user logged in

`nulladm`(8)

Creates `file` with permission bits set to 0664 and owner set to the value defined by the `CHGRP` parameter in file `/etc/csa.conf`

`csa_auth`(3)

Checks if the caller has the necessary capabilities

`csa_check`(3)

Checks a kernel, daemon, or record accounting state

`csa_halt`(3)

Stops all accounting methods

`csa_kdstat`(3)

Gets the kernel and daemon accounting status

`csa_rcdstat`(3)

Gets the record accounting status

`csa_stop`(3)

Stops the specified accounting method(s)

`csa_wracct`(3)

Writes the accounting record to a file

CSA Job man pages provided for the Cray CSA implementation:

`job`(7)

Provides an overview of the Linux Jobs kernel module

`jkill`(1)

Sends a signal to a job

`jstat`(1)

Displays the job status information

`jwait`(1)

Awaits completion of a job

`jattach`(8)

Attaches a processes to a job

`jdetach`(8)

Detaches a group of processes from a job

`jsethid`(8)

Enables unique job ID values

`csajastart`(3)

Starts job accounting

`csa_jastop`(3)

Stops job accounting

`csa_start`(3)

Gets the user ID of a job

`job_attachpid`(3)

Attaches a process to a requested job

`job_create`(3)

Creates a new job

`job_detachjid`(3)

Detaches all the processes from a job

`job_detachpid`(3)

Detaches a process from its current job

`job_getjid`(3)

Returns the job ID for the given process

`job_getjidcnt`(3)

Returns the number of jobs currently on the system

`job_getjidlist`(3)

Gets the currently active job IDs

`job_getpidcnt`(3)

Gets the number of processes attached to a job

`job_getpidlist`(3)

Gets the list of process IDs attached to a job

`job_getprimepid`(3)

Gets the prime process ID for a job

`job_getuid`(3)

Gets the user ID of a job

`job_killjid`(3)

Sends a signal to all processes in a job

`job_sethid`(3)

Sets the handle ID for new job IDs

`job_waitjid`(3)

Waits for a job to complete

For more information about these commands, see the associated man pages, which have been included without modification.

4.9 Cray Glossary

A Cray Glossary of terms specific the Cray XT system is included with CrayDoc. The entire Cray Glossary is available on the CrayDoc public website:

docs.cray.com

4.10 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 SUSE Linux is at http://www.novell.com/linux and documentation for the Linux Documentation Project is at http://www.tldp.org
Lustre	Additional Lustre documentation is available at http://manual.lustre.org and http://www.sun.com/software/products/lustre

Product	Documentation Source
MySQL	MySQL documentation is available at http://www.mysql.com/documentation
PBS Professional	Documentation for the PBS Professional work load manager system software is available from Altair Engineering, Inc. at http://www.altair.com
Moab/TORQUE	Documentation for Moab/TORQUE work load manager system software is available from Cluster Resources, Inc. at http://www.clusterresources.com/
Platform LSF	Documentation for Platform LSF software is available from Platform Computing Corporation at http://www.platform.com/
RPM	RPM documentation is available at http://www.rpm.org
glibc	glibc documentation is available at http://gcc.gnu.org/onlinedocs
GNet	GNet documentation is available at http://www.gnetlibrary.org
GLIB	GLIB documentation is available at http://developer.gnome.org/doc/API/2.0/glib/index.html

4.11 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

Mail:
Software Distribution Center
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Release Contents [5]

5.1 Cray XT System Configurations

The Cray Linux Environment (CLE) 2.1 release continues to provide support for Cray XT3 single- and dual-core systems and Cray XT4 dual-core systems. The following new system configurations are also supported in the CLE 2.1 release.

- **Cray XT4 Quad-core systems.** Initial support for Quad-core AMD Opteron processors began in an update of the UNICOS/lc 2.0 release.
- **Cray $XT5_h$ systems.** Initial support for Cray $XT5_h$ systems with Cray X2 compute blades began in the Cray $XT5_h$ 1.0 GA (General Availability) release and UNICOS/lc 2.0.53y update release. Support for additional features was provided in the UNICOS/lc 2.0.62 update release. The CLE 2.1 release continues support for Cray $XT5_h$ systems.
- **Cray XT5 systems.** A variant of the CLE 2.1 LA release, 2.1 HD, provided initial support for Cray XT5 systems.

5.2 Software Requirements

The System Management Workstation (SMW) must be upgraded to the SMW 3.1 GA release running SUSE Linux Enterprise Server 10, Service Pack 1 (SLES 10 SP1) **before** upgrading to the CLE 2.1 release.

You must be running release version UNICOS/lc 2.0 GA or later on your Cray XT system in order to upgrade to the CLE 2.1 release.

Caution: Upgrading your Cray XT from operating system version 1.5 is not supported and requires an intermediate step involving UNICOS/lc 2.0. Contact your Cray representative if you plan to upgrade from version 1.5 to CLE 2.1.

Several Cray software packages are available as separately released products. Table 6 and Table 7 list the minimum release levels required to use these products with CLE 2.1. 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 Levels for Other Cray Products Supported with CLE 2.1

Product	Minimum Release Level	For Release Information
System Management Workstation (SMW)	Release 3.1 or later running SLES 10 SP1	*Cray System Management Workstation (SMW) Software Release Overview* (S-2482-31)
Cray Message Passing Toolkit (MPT)	Systems with CNL compute nodes: MPT 3.0 or later; Systems with over 64K processing elements: MPT 3.1 or later; Systems with over 32K processing elements for Cray SHMEM applications: MPT 3.1 or later; Systems with Catamount compute nodes: MPT 2.1[1]	*Cray Programming Environments Installation Guide* (S-2465-30)
Cray Programming Environments for Cray XT systems	Release 3.0 or later	*Cray Programming Environments Installation Guide* (S-2465-30)
Cray Performance Analysis Tools	Release 4.3 or later. CrayPat and Cray Apprentice2 are included in the 4.3 release.	*Cray Performance Analysis Tools Release Overview and Installation Guide* (S-2474-43)

If you have a Cray $XT5_h$ system, some Cray products require different or additional release levels to provide support for Cray X2 compute nodes.

Table 7. Additional Requirements for Other Cray Products on Cray $XT5_h$ Systems.

Product	Required Release Level	For Release Information
System Management Workstation (SMW)	Systems with Cray X2 compute nodes must run release level 3.1.09 or later.	*Cray System Management Workstation (SMW) Software Release Overview* (S-2482-31)
Cray Message Passing Toolkit (MPT)	MPT 1.0	
Cray Programming Environments	Release 6.0	*Cray Programming Environment Releases Overview and Installation Guide* (S-5212-60)

1 Unlike later MPT releases, which are released separately, MPT release 2.1 is only distributed with the CLE 2.1 release. CNL software and MPT 2.1 are not compatible.

Information regarding supported and certified batch system software products is available on the CrayPort website at crayport.cray.com. Click on **3rd Party Batch SW** in the menu bar. For additional information on CrayPort, see Section 6.2, page 83.

5.3 Contents of the Release Package

5.3.1 Components for All Systems

The CLE 2.1 release package includes the following operating system software:

- Linux 2.6.16 kernel and SUSE Linux Enterprise Server 10, Service Pack 1 (SLES 10 SP1) service node software.
- CNL 2.1 compute node software.
- Catamount 2.1 compute node software.

  Catamount, which is also known as the quintessential kernel or Qk, was developed by Sandia National Laboratories. Compute nodes running the Catamount operating system support dual-core processing through the Catamount Virtual Node (CVN) capability. Support for Catamount/Qk on quad-core Opteron processors is not supported.

  Note: CLE 2.1 is the last operating system release to include Catamount compute node software.

The CLE 2.1 release includes the following additional system software products:

- Lustre file system (Version 1.6) from Sun Microsystems, Inc.
- Application Level Placement Scheduler (ALPS)
- Cray Data Virtualization Service (Cray DVS)
- Comprehensive System Accounting (CSA)
- Cray Audit
- MySQL (Version 5.0) from Sun Microsystems, Inc.
- Realm-Specific Internet Protocol (RSIP)
- Linux `ldump` and `lcrash` Utilities

The CrayDoc software suite and the documentation, described in Chapter 4, page 61 is included with the release package.

5.3.2 Additional Components for Cray XT5$_h$ Systems

For sites with Cray X2 compute nodes, the release provides these additional components:

- Cray X2 RPMs, which are for the Cray NV-2 architecture
- The `CRAYX2install.sh` script for installation and configuration
- The `CRAYXTinstall.sh` script for validating that the minimum version of Cray XT software is installed on the SMW
- The `CRAYSMWinstall.sh` script for validating that the minimum version of SMW software is installed on the SMW
- Various supporting files for the preceding scripts

5.4 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 XT systems under a Cray license agreement:

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

  Note: Source Code Option: The Cray XT OS and OS Components for Cray X2 Systems licenses are binary by default. Certain U.S. customers may be eligible to obtain a buildable source license for an additional fee. For more information regarding source code for the Cray XT systems, contact your sales representative.

- Lustre Parallel File System (contractual rights to Lustre are included with the Cray XT OS license for some initial customers)

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

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.

5.5 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.

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

Mail:
Software Distribution Center
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA

Software will be shipped by ground service or 5-day international service.

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. You can also view all of the Bugs for a particular system, even those bug reports 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 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
- Request technical assistance through the Case Management interface
- View, update, and close cases that you originate
- Communicate with other Cray system users in a moderated forum
- Learn about solutions to various problems
- Order Cray software
- View Cray software product documentation
- View Cray service documentation
- View batch system software information

To access CrayPort, use the following link: crayport.cray.com. If you need a password, 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:

ttd_online@cray.com

Web:
www.cray.com/training/

Fax:
+1–715–726–4991

Mail:
Technical Training
Cray Inc.
P.O. Box 6000
Chippewa Falls, WI 54729–0080
USA

6.4 Cray Public Website

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

www.cray.com

Package Differences [A]

The following table displays the differences between SLES 9 SP2 and SLES 10 SP1 packages Cray provides. For the Cray Linux Environment (CLE) 2.1 release, if a package has a different name in the SLES 10 SP1 column, it was replaced completely by the new package (for example, apache was replaced by apache2 in SLES 10 SP1). If a package says "*name* retained" in the SLES 10 SP1 column, two versions of the package were provided in SLES 9 SP2 but only the "*name* retained" package is provided in SLES 10 SP1 (for example, `gconf` and `gconf2` exist in SLES 9 SP2, but only `gconf2` exists in SLES 10 SP1). If a package has no new name, there is no replacement provided in SLES 10 SP1 and it was removed entirely (for example, `bastille` has no SLES 10 SP1 equivalent).

Table 8. Packages Differences Between SLES 9 SP2 and SLES 10 SP1

SLES 9 SP2	SLES 10 SP1	Description
alice-compat	Removed	Automatic Linux Installation and Configuration
antiword	Removed	convert files from word to text
apache	apache2	
bastille	Removed	security hardening
bonobo	Removed	CORBA interfaces
bug-buddy	Removed	graphical bug reporting utility
core-release	Removed	
cryptplug	Removed	plugins for crypto engines
directory_administrator	Removed	LDAP admin tool
evlog	syslogd retained	Enterprise eVent Logging - syslog+
freeswan	Removed	VPN
gal	gal2	
gcc-g77	gcc-fortran	

SLES 9 SP2	SLES 10 SP1	Description
gconf	gconf2 retained	
gettyps	Removed	
gkermit	Removed	file transfer program
glade	Removed	GTK2+ user interface builder
gnokii	Removed	nokia connectivity program
gnome2-SuSE	gnome2-NLD	
gq	Removed	LDAP client for GTK
heimdal	krb5	
hotplug	Removed	automatic configuration of hotplugged devices
imwheel	Removed	mouse control for MS intellimouse
inetd	xinetd	
java2	java-1_4_2-sun	
kdebase3-SLES	kdebase3-NLD	
libghttp	Removed	GNOME library for HTTP access
libgtkhtml	Removed	GTK2.0 library for HTML support
libPropList	Removed	properties list for WindowMaker
libselinux	Removed	security enhanced linux
libungif	Removed	uncompressed GIF images handling library
libunicode	Removed	Unicode handling library
libxml	libxml2 retained	

SLES 9 SP2	SLES 10 SP1	Description
linc	Removed	network client/server library
linux-iscsi	Removed	iSCSI driver
listexec	Removed	graphical or audio directory lister
logsurfer	Removed	system log handling tool
mad	Removed	MPEG audio decoder library
marsnwe	Removed	Novell server emulation
metacity-themes	Removed	themes for metacity window manager
modules	Modules	
mozilla	MozillaFirefox	
newpg	Removed	s/mime variant of GnuPG
ntop	Removed	Web based network traffic monitor
oaf	Removed	GNOME Object Activation Framework library
orbit	orbit2 retained	
papi	Removed	Performance Application Programming Interface
pdksh	ksh	
perl-OPENSSL	Removed	Perl interface to OpenSSL
php4	php5	
python-japanese	Removed	Japanese codecs for Python

SLES 9 SP2	SLES 10 SP1	Description
python-korean	Removed	Korean codecs for Python
qt3-non-mt	Removed	Qt library for developing GUIs
raidtools	mdadm retained	
release-notes	release-notes-sles	
rp-pppoe	Removed	PPP-over-Ethernet redirector for pppd
rstatd	Removed	RPC kernel statistics server
ruby	Removed	scripting language
saint	Removed	(Security Administrators Integrated Network Tool
saxident	sax2-ident	
saxtools	sax2-tools	
schedutils	Removed	utilities for manipulating process scheduler attributes
submount	Removed	Auto mounting of removable media
tarfix	Removed	recovers damaged tar files
tripwire	aide	
words-words	words	
xbanner	Removed	X window system background writing & images
xchat	Removed	IRC client
XFree86	xorg-x11	
xmms	Removed	extensible media player

SLES 9 SP2	SLES 10 SP1	Description
yast2-ipsec	Removed	SUSE configuration tool
yast2-packagemanager	Removed	SUSE configuration tool
yast2-packagemanager-devel	Removed	SUSE configuration tool
yast2-phone-services	Removed	SUSE configuration tool
yast2-theme-SuSELinux	yast2-theme-NLD	
yast2-you-server	Removed	SUSE configuration tool