Cray XT3™ System Overview

S–2423–12
## Record of Revision

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.0     | December 2004  
         | Draft documentation to support Cray XT3 early-production systems. |
| 1.0     | March 2005  
         | Draft documentation to support Cray XT3 limited-availability systems. |
| 1.1     | June 2005  
         | Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.1,  
         | System Management Workstation (SMW) 1.1, and UNICOS/lc 1.1 releases. |
| 1.2     | September 2005  
         | Supports Cray XT3 systems running the Cray XT3 Programming Environment 1.2,  
         | System Management Workstation (SMW) 1.2, and UNICOS/lc 1.2 releases. |
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>vii</td>
</tr>
<tr>
<td>Accessing Product Documentation</td>
<td>vii</td>
</tr>
<tr>
<td>Conventions</td>
<td>viii</td>
</tr>
<tr>
<td>Reader Comments</td>
<td>ix</td>
</tr>
<tr>
<td><strong>Introduction [1]</strong></td>
<td>1</td>
</tr>
<tr>
<td>Related Publications</td>
<td>4</td>
</tr>
<tr>
<td>Publications for Application Developers</td>
<td>4</td>
</tr>
<tr>
<td>Publications for System Administrators</td>
<td>6</td>
</tr>
<tr>
<td><strong>Hardware Overview [2]</strong></td>
<td>7</td>
</tr>
<tr>
<td>Basic Hardware Components</td>
<td>7</td>
</tr>
<tr>
<td>AMD Opteron Processor</td>
<td>7</td>
</tr>
<tr>
<td>DIMM Memory</td>
<td>8</td>
</tr>
<tr>
<td>Cray SeaStar Chip</td>
<td>8</td>
</tr>
<tr>
<td>System Interconnection Network</td>
<td>9</td>
</tr>
<tr>
<td>Nodes</td>
<td>9</td>
</tr>
<tr>
<td>Compute Nodes</td>
<td>10</td>
</tr>
<tr>
<td>Service Nodes</td>
<td>10</td>
</tr>
<tr>
<td>Blades, Chassis, and Cabinets</td>
<td>11</td>
</tr>
<tr>
<td>Blades</td>
<td>11</td>
</tr>
<tr>
<td>Compute Blade</td>
<td>11</td>
</tr>
<tr>
<td>Service Blade</td>
<td>12</td>
</tr>
<tr>
<td>Chassis and Cabinets</td>
<td>12</td>
</tr>
<tr>
<td>I/O System</td>
<td>13</td>
</tr>
</tbody>
</table>
# Software Overview [3]

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNICOS/lc Operating System</td>
<td>15</td>
</tr>
<tr>
<td>Lustre File System</td>
<td>16</td>
</tr>
<tr>
<td>Cray XT3 Development Environment</td>
<td>16</td>
</tr>
<tr>
<td>User Environment</td>
<td>16</td>
</tr>
<tr>
<td>System Access</td>
<td>17</td>
</tr>
<tr>
<td>Compilers</td>
<td>17</td>
</tr>
<tr>
<td>Libraries</td>
<td>18</td>
</tr>
<tr>
<td>Linker</td>
<td>19</td>
</tr>
<tr>
<td>Runtime Environment</td>
<td>19</td>
</tr>
<tr>
<td>Interactive Jobs</td>
<td>20</td>
</tr>
<tr>
<td>Batch Jobs</td>
<td>21</td>
</tr>
<tr>
<td>Debugging Applications</td>
<td>23</td>
</tr>
<tr>
<td>Measuring Performance</td>
<td>23</td>
</tr>
<tr>
<td>Performance API (PAPI)</td>
<td>23</td>
</tr>
<tr>
<td>CrayPat</td>
<td>23</td>
</tr>
<tr>
<td>Cray Apprentice(^2)</td>
<td>24</td>
</tr>
<tr>
<td>System Administration</td>
<td>25</td>
</tr>
<tr>
<td>System Management Workstation (SMW)</td>
<td>25</td>
</tr>
<tr>
<td>Shared-Root File System</td>
<td>25</td>
</tr>
<tr>
<td>Lustre File System Administration</td>
<td>26</td>
</tr>
<tr>
<td>Lustre UFS Like File System</td>
<td>26</td>
</tr>
<tr>
<td>Lustre Parallel File System</td>
<td>27</td>
</tr>
<tr>
<td>Configuration and Source Files</td>
<td>28</td>
</tr>
<tr>
<td>System Log</td>
<td>29</td>
</tr>
<tr>
<td>Service Database (SDB)</td>
<td>29</td>
</tr>
<tr>
<td>System Activity Reports</td>
<td>30</td>
</tr>
</tbody>
</table>

# Cray RAS and Management System (CRMS) [4]

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRMS Hardware</td>
<td>31</td>
</tr>
<tr>
<td>CRMS Network</td>
<td>32</td>
</tr>
<tr>
<td>Contents</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>System Management Workstation</td>
<td>33</td>
</tr>
<tr>
<td>Hardware Controllers</td>
<td>33</td>
</tr>
<tr>
<td>CRMS Software</td>
<td>33</td>
</tr>
<tr>
<td>Software Monitors</td>
<td>33</td>
</tr>
<tr>
<td>CRMS Administrator Interfaces</td>
<td>34</td>
</tr>
<tr>
<td>CRMS Actions</td>
<td>34</td>
</tr>
<tr>
<td>System Startup and Shutdown</td>
<td>35</td>
</tr>
<tr>
<td>Event Probing</td>
<td>35</td>
</tr>
<tr>
<td>Event Logging</td>
<td>35</td>
</tr>
<tr>
<td>Event Handling</td>
<td>36</td>
</tr>
<tr>
<td>Glossary</td>
<td>37</td>
</tr>
<tr>
<td>Index</td>
<td>43</td>
</tr>
<tr>
<td>Figures</td>
<td></td>
</tr>
<tr>
<td>Figure 1. Cray XT3 Supercomputer System</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2. Cray SeaStar Chip</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3. Cray XT3 System Compute Node</td>
<td>10</td>
</tr>
<tr>
<td>Figure 4. Cray XT3 System Service Node</td>
<td>11</td>
</tr>
<tr>
<td>Figure 5. Chassis and Cabinet (front view)</td>
<td>13</td>
</tr>
<tr>
<td>Figure 6. I/O System</td>
<td>14</td>
</tr>
<tr>
<td>Figure 7. Launching Interactive Jobs</td>
<td>21</td>
</tr>
<tr>
<td>Figure 8. Launching Batch Jobs</td>
<td>22</td>
</tr>
<tr>
<td>Figure 9. Lustre UFS Like File System</td>
<td>27</td>
</tr>
<tr>
<td>Figure 10. Lustre Parallel File System</td>
<td>28</td>
</tr>
<tr>
<td>Figure 11. CRMS Components</td>
<td>32</td>
</tr>
</tbody>
</table>
The information in this preface is common to Cray documentation provided with this software release.

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:

CrayDoc     The Cray documentation delivery system that allows you to quickly access and search Cray books, man pages, and in some cases, third-party documentation. Access this 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   Access man pages by entering 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

Access third-party documentation not provided through CrayDoc according to the information provided with the product.
Conventions

These conventions are used throughout Cray documentation:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>This fixed-space font denotes literal items, such as file names, pathnames, man page names, command names, and programming language elements.</td>
</tr>
<tr>
<td>variable</td>
<td>Italic typeface indicates an element that you will replace with a specific value. For instance, you may replace filename with the name datafile in your program. It also denotes a word or concept being defined.</td>
</tr>
<tr>
<td>user input</td>
<td>This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Brackets enclose optional portions of a syntax representation for a command, library routine, system call, and so on.</td>
</tr>
<tr>
<td>...</td>
<td>Ellipses indicate that a preceding element can be repeated.</td>
</tr>
<tr>
<td>name(N)</td>
<td>Denotes man pages that provide system and programming reference information. Each man page is referred to by its name followed by a section number in parentheses. Enter:</td>
</tr>
<tr>
<td></td>
<td>% man man</td>
</tr>
<tr>
<td></td>
<td>to see the meaning of each section number for your particular system.</td>
</tr>
</tbody>
</table>
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:**
Software Publications
Cray Inc.
1340 Mendota Heights Road
Mendota Heights, MN 55120–1128
USA
This document provides a basic overview of the Cray XT3 system for application developers and system administrators. It assumes readers are familiar with the concepts of high-performance computing and the architecture of parallel processing systems.

**Note:** Functionality marked as deferred in this documentation is planned to be implemented in a later release.

Cray XT3 supercomputer systems are powerful, massively parallel processing (MPP) systems. Cray has combined commodity and open source components with custom-designed components to create a system that can operate efficiently at immense scale.

Cray XT3 systems are based on the Red Storm technology that was developed jointly by Cray Inc. and the U.S. Department of Energy’s Sandia National Laboratories.

Cray XT3 systems are designed to run applications that require large-scale processing, high network bandwidth, and complex communications. Typical applications are those that create detailed simulations in both time and space, with complex geometries that involve many different material components. These long-running, resource-intensive applications require a system that is programmable, scalable, reliable, and manageable.

The major features of Cray XT3 systems are:

- Cray XT3 systems scale from 200 to 30,000 processors. The ability to scale to such proportions stems from the design of system components:
  - The basic scalable component is the node. There are two types of nodes. Compute nodes run user applications. Service nodes provide support functions, such as managing the user’s environment, handling I/O, and booting the system. Each compute node and service node is a logical grouping of a processor, memory, and a data routing resource.
  - Cray XT3 systems use a simple memory model. For applications distributed across numerous nodes, each instance of the application has its own processor and local memory. Remote memory is the memory on the nodes running the associated application instances. There is no shared memory.

Cray XT3 systems use Chipkill memory, IBM’s advanced Error Checking and Correcting (ECC) technology. Chipkill memory can correct a
maximum of four bits of memory errors per each 128-bit data request. Normal ECC chips will fix only a single bad bit per data request. On a 10,000 processor system, normal ECC protection results in 900 memory failures over a three-year period; using Chipkill ECC protection reduces the number of failures to 6.

– The *system interconnection network* is the data-routing resource that Cray XT3 systems use to maintain high communication rates as the number of nodes increases. The system interconnection network enables the system to achieve an appropriate balance between processor speed and interconnection bandwidth.

• The Cray XT3 system development environment extends standard Fortran, C, and C++ via compiler directives, library routines and intrinsic functions, and parallel programming model constructs. The system supports the Message Passing Interface (MPI) and Cray SHMEM parallel programming, distributed memory models. The message passing interface is Cray MPICH2, an implementation of MPI-2 by the Argonne National Laboratory Group. Performance across processors is uniform, simplifying Cray MPICH2 and Cray SHMEM application design.

• Integrated software development tools support the creation of complex multinode applications. Examples are PGI Fortran, C, and C++ compilers; AMD Core Math Library (ACML) functions; Cray MPICH2 and Cray SHMEM parallel processing libraries; and the TotalView debugger (available from Etnus, LLC).

• The UNICOS/lc operating system is tailored to the requirements of computation and service components. Nodes that process user applications run an operating system microkernel (also referred to as the *quintessential kernel* or Qk). The microkernel’s limited interaction with applications significantly minimizes OS interference and allows for the addition of compute nodes without a corresponding increase in operating system overhead. It also allows for reproducible run times, a feature of major significance for sites running production systems. A full-featured operating system runs on service nodes.

• The Cray XT3 Lustre file system scales to thousands of clients and petabytes of data. Lustre separates file metadata from data objects. Each instance of a Lustre file system consists of Object Storage Servers (OSSs) and a Metadata Server (MDS). Each OSS hosts two Object Storage Targets (OSTs). Applications use Lustre object storage targets (OSTs) to transfer data objects; these data objects can be striped across RAID storage devices. Lustre can be configured as a parallel file system optimized for large-scale serial access or
as a UFS like file system, optimized for accessing many small files typically seen with root file systems and home directories.

- Reliability, availability, and serviceability (RAS) are designed into system components:
  - Cray XT3 system cabinets have only one moving part (a blower that cools the components) and redundant power supplies, reducing the likelihood of cabinet failure.
  - Cray XT3 system processor boards (called blades) have several reliability features. The blades have redundant voltage regulator modules (VRMs), which are the solid state components most likely to fail. All components are surface mounted, and the blades have a minimal number of components to facilitate airflow.
  - All Cray XT3 nodes are diskless; the availability of a node is not tied to the availability of a moving part.
  - DataDirect Networks (DDN) is the supplier of the system RAID. Each RAID device has multiple redundant RAID controllers with automatic failover capability and multiple Fibre Channel connections to the disk storage.

- The Cray XT3 single-system view (SSV) is a set of operating system features that provide users and administrators with one view of the system, significantly simplifying the administrator’s tasks.

- The Cray RAS and Management System (CRMS) monitors and manages all major Cray XT3 system components. The CRMS is independent of computation and service components and has its own network. If a blade fails, the CRMS detects it and informs the administrator, who performs a hot swap (Deferred implementation).
1.1 Related Publications

Cray XT3 systems run with a combination of proprietary, third party, and open source products, as documented in the following publications.

1.1.1 Publications for Application Developers

- *Cray XT3 System Overview* (this manual)
- *Cray XT3 Programming Environment User’s Guide*
- *Cray XT3 Systems Software Release Overview*
- *PGI User’s Guide*
- PGI Tools Guide
- PGI Fortran Reference
- Compiler commands man pages: cc(1), CC(1), ftn(1), f77(1)
- Modules utility man pages: module(1), modulefile(4)
- Commands related to application launch: yod(1), xtshowmesh(1)
- Cray MPICH2 man pages: read the intro_mpi(1) man page first
- Cray SHMEM man pages: read the intro_shmem(1) man page first
- LAPACK man pages
- ScALAPACK man pages
- BLACS man pages
- AMD Core Math Library (ACML)
- SuperLU Users’ Guide
- PBS Pro 5.3 Quick Start Guide, PBS-3BQ01
- PBS Pro 5.3 User Guide, PBS-3BU01
- PBS Pro man pages
- PBS Pro 5.3 External Reference Specification, PBS-3BE01
- PAPI User’s Guide
- PAPI man pages
- PAPI Programmer’s Reference
- PAPI Software Specification
- CrayPat 2 man pages: start with craypat(1)
- Cray Apprentice2 app2(1) man page
- TotalView documentation
- UNICOS/lc man pages (start with intro_xt3(1))

1 PBS Pro is an optional product from Altair Grid Technologies. See http://www.altair.com.
2 An optional Cray product.
3 TotalView is an optional product available from Etnus, LLC (http://www.etnus.com).
1.1.2 Publications for System Administrators

- Cray XT3 System Overview (this manual)
- Cray XT3 Systems Software Release Overview
- Cray XT3 Software Installation and Configuration Guide
- Cray XT3 System Management
- UNICOS/lc administrator man pages
- Cray LINUX man pages
- SUSE LINUX man pages
- Lustre man pages
- PBS Pro Release Overview, Installation Guide, and Administration Addendum for Cray XT3 Systems \(^4\)
- PBS Pro 5.3 Administrator Guide, PBS-3BA01 \(^4\)
- Linux documentation – see Linux Documentation Project - www.tldp.org and www.suse.com

\(^4\) PBS Pro is an optional product from Altair Grid Technologies. See http://www.altair.com.
Cray XT3 system hardware consists of computation components, service components, the system interconnection network, and CRMS components. This chapter describes the computation and service components and the system interconnection network. For a description of CRMS hardware, see Chapter 4, page 31.

2.1 Basic Hardware Components

The basic hardware components of the Cray XT3 system are the:

- AMD Opteron processor
- Dual in-line memory modules (DIMMs)
- Cray SeaStar chip
- System interconnection network

2.1.1 AMD Opteron Processor

The AMD Opteron processor fully supports the x86 instruction set and is fully compatible with AMD Socket 940 design.

The processor supports out-of-order execution and can issue a maximum of nine instructions simultaneously.

Sixteen 64-bit registers and a floating-point unit support full 64-bit IEEE floating-point operations. An integer processing unit provides full 64-bit integer arithmetic.

The processor also has four 48-bit performance counters that can be used to monitor the number or duration of processor events, such as the number of data cache misses or the time to return data from memory after a cache miss.

The AMD Opteron processor memory controller supports Chipkill memory protection, enabling the system to significantly reduce the number of memory failures.
2.1.2 DIMM Memory

The Cray XT3 system supports double data rate Dual In-line Memory Modules (DIMMs). DIMM sizes are 512 MB, 1 GB, and 2 GB. With four DIMM slots per node, the physical maximum amount of memory is 8 GB per node. However, the maximum amount of memory supported by the operating system is 4 GB per node.

2.1.3 Cray SeaStar Chip

The Cray SeaStar application-specific integrated circuit (ASIC) chip is the system’s message processor. It offloads communications functions from the AMD Opteron processor. The chip contains:

- The HyperTransport Link connecting the AMD Opteron processor to the Cray SeaStar chip.
- The Direct Memory Access (DMA) engine that manages the movement of data to and from DIMMs. The DMA engine is controlled by an on-board processor.
- A router that connects the chip to the system interconnection network via high-speed network links (see Section 2.1.4, page 9).
- The Portals message-passing interface, portions of which are implemented in Cray SeaStar firmware, that provides a data path from an application to memory. The firmware transfers data directly to and from user memory without operating system intervention.
- A link to a Cray XT3 system L0 controller. The Cray XT3 system L0 controller is used for booting, monitoring, and maintenance (see Section 4.1.3, page 33).
2.1.4 System Interconnection Network

The system interconnection network is the communications center of the Cray XT3 system. The network uses a Cray proprietary protocol to provide fast node-to-node message passing and fast I/O to a global, shared file system.

The system interconnection network consists of the Cray SeaStar router links and the cables that connect the computation and service components, combined in a three-dimensional torus topology. Network cables can be configured in multiple ways to create various topologies.

2.2 Nodes

Cray XT3 processing components combine to form a node (also referred to as a processing element or PE). Cray XT3 systems have two types of nodes: compute nodes and service nodes.
2.2.1 Compute Nodes

Compute nodes run application programs. Each compute node consists of one processor socket populated with a minimum of one single-core AMD Opteron processor, four DIMM slots providing 1-8 GB of local memory, and a Cray SeaStar chip that connects the processor to the system interconnection network.

All compute nodes in a system use the same processor type. Because the processors are socketed, customers can upgrade nodes as faster processors become available.

The set of all compute nodes is referred to as the compute partition.

2.2.2 Service Nodes

Service nodes handle support functions such as system startup and shutdown, user login, I/O, and network management. Service nodes use the same processors, memory, and SeaStar chips as compute nodes. In addition, each service node has two PCI-X slots. PCI-X cards plug into the slots and interface to external I/O devices. Different PCI-X cards are used for different types of service nodes. For a description of the types of service nodes, see Section 3.1, page 15.)
The set of all service nodes is referred to as the service partition.

2.3 Blades, Chassis, and Cabinets

This section describes the main physical components of the Cray XT3 system and their configurations.

2.3.1 Blades

While the node is the logical building block of the Cray XT3 system, the basic physical component and field-replaceable unit is the blade. There are two types of blades: compute blades and service blades.

2.3.1.1 Compute Blade

A compute blade consists of four nodes (one AMD Opteron processor socket, four DIMM slots, and one SeaStar chip per node), voltage regulator modules, and a Cray XT3 system L0 controller.

Each compute blade is populated with AMD processors and memory chips of the same type and speed.

The Cray XT3 system L0 controller is a CRMS component; for more information, see Chapter 4, page 31.
2.3.1.2 Service Blade

A service blade consists of two nodes, voltage regulator modules, PCI-X cards, and a Cray XT3 system L0 controller. Each node contains an AMD Opteron processor socket, four DIMM slots, and a Cray SeaStar chip. (Service blades have four Cray SeaStar chips to allow for a common and interchangeable board design.) Several different PCI-X cards are available to provide Fibre Channel, GigE, and 10 GigE interfaces to external devices.

2.3.2 Chassis and Cabinets

Each cabinet contains three vertically stacked chassis, and each chassis contains eight vertically mounted blades. A cabinet can contain compute blades, service blades, or a combination of compute and service blades. A single variable-speed blower in the base of the cabinet cools the components.

A three-phase AC delivers power to the cabinet where the power is converted to DC. All cabinets have redundant power supplies. The power supplies and the cabinet control processor (L1 controller) are located at the rear of the cabinet.
2.4 I/O System

The I/O system handles data transfers to and from storage devices, user applications, and the CRMS. The I/O system consists of service nodes and RAID devices:

- Service nodes provide I/O, network, login, and boot services.
  - I/O service nodes use Fibre Channel PCI-X cards to provide connections to external devices, such as Fibre Channel RAID storage.
  - Network service nodes use 10 GigE PCI-X cards to provide connections to network storage devices.
– Login service nodes use GigE PCI-X cards to provide connections to user workstations.

– Boot service nodes use Fibre Channel PCI-X cards to provide connections to the boot RAID devices and GigE PCI-X cards to provide connections to the System Management Workstation (SMW). For further information on the SMW, see Section 4.1.2, page 33.

• DataDirect Networks (DDN) parallel RAID systems are used for storing user application data, supporting the system boot process, and supporting system management and system administration functions.

RAID storage includes one or more RAID subsystems. A subsystem consists of a singlet or couplet controller and all Fibre Channel switches and disk enclosures that attach to the controller. Each disk enclosure has 16 disk drive slots with disks arranged in tiers. Each tier consists of data drives and a parity drive. All RAID components are fully redundant.

Figure 6. I/O System
Software Overview

Cray XT3 systems run a combination of Cray developed software, third-party software, and open source software. The software is optimized for applications that have fine-grain synchronization requirements, large processor counts, and significant communication requirements.

Message passing via Cray MPICH2 and Cray SHMEM and the underlying Portals layer optimizes communication among applications. A single-point interface to the administrator’s environment and a shared-root file system simplify system administration.

This chapter provides an overview of the UNICOS/lc operating system, the Lustre file system, the development environment, and system administration tools. For a description of CRMS software, see Chapter 4, page 31.

3.1 UNICOS/lc Operating System

The UNICOS/lc operating system consists of service node and compute node components.

- Service nodes perform the functions needed to support users, applications, and administrators. Service nodes run a full-featured version of SUSE LINUX. Above the operating system level are specialized daemons and applications that perform functions unique to each service node type.

There are four basic types of service nodes:

- **login nodes** where users log in and where PBS Pro resource managers run
- **I/O nodes** that manage file system metadata and transfer data to and from storage devices and applications
- **network nodes** that provide TCP/IP connections to external systems
- **system nodes** that perform special services such as system boot, system administration, Service Database (SDB) management (see Section 3.4.6, page 29), and syslog management (see Section 3.4.5, page 29)

A system administrator can reconfigure service nodes to improve system efficiency. For more information, see the *Cray XT3 System Management* manual.
• Compute nodes run a microkernel named *Catamount*. *Catamount* was developed by Sandia National Laboratories to provide support for application execution without the overhead of a full operating system image.

The microkernel interacts with an application in very limited ways. It provides virtual memory addressing and physical memory allocation, memory protection, access to the message-passing layer, and a scalable job loader. The microkernel provides limited support for I/O operations. Compute node applications are limited to the amount of physical memory on a node; the microkernel does not support demand paged virtual memory.

### 3.2 Lustre File System

I/O nodes host the file system. Cray XT3 systems run Lustre, a high-performance, highly scalable, POSIX-compliant shared file system. Lustre is based on Linux and uses the Portals lightweight message passing API and an object-oriented architecture for storing and retrieving data.

Lustre’s file I/O operations are transparent to the application developer. The I/O functions available to the application developer—Fortran, C, and C++ I/O calls; C and C++ stride I/O calls (*readx*, *writex*, *ireadx*, and *iwritex*); and system I/O calls—are converted to Lustre library calls. Lustre handles file I/O between applications and I/O nodes, and between I/O nodes and RAID storage devices.

For a description of Lustre administration, see Section 3.4.3, page 26.

### 3.3 Cray XT3 Development Environment

The Cray XT3 system development environment is the set of software products and services that developers use to build and run applications.

#### 3.3.1 User Environment

The user environment on a Cray XT3 system is similar to the environment on a typical Linux workstation. However, there are steps that the user needs to take before starting to work on a Cray XT3 system:

1. Set up a secure shell. Cray XT3 systems use `ssh` and `ssh`-enabled applications for secure, password-free remote access to login nodes. Before using `ssh` commands, the user needs to generate an RSA authentication key.
2. Load the appropriate modules. Cray XT3 systems use the Modules utility to support multiple versions of software, such as compilers, and to create integrated software packages. As new versions of the supported software become available, they are added automatically to the Programming Environment and earlier versions are retained to support legacy applications. By specifying the module to load, the user can choose the default or another version of one or more Programming Environment tools.

For details, see the Cray XT3 Programming Environment User’s Guide and the module(1) and modulefile(4) man pages.

3.3.2 System Access

To access the Cray XT3 system, users enter the ssh command to log in from a standard terminal window.

Logins are distributed among login nodes using a load-leveling service (Deferred implementation) that intercepts login attempts and directs them to the least heavily used login node.

A login node provides all of the standard Linux utilities and commands, a variety of shells, and access to application development tools.

3.3.3 Compilers

The Cray XT3 System Programming Environment includes Fortran 90/95, FORTRAN 77, C, and C++ compilers from The Portland Group (PGI), a wholly owned subsidiary of STMicroelectronics. Developers use the PGI compilers to compile single and multiple compute node applications.

The compilers translate Fortran 90/95, FORTRAN 77, C, and C++ source programs into Cray XT3 system object files. Interlanguage communication functions enable developers to create Fortran programs that call C or C++ routines and C or C++ programs that call Fortran routines.

The command used to invoke a compiler is called a compilation driver; it can be used to apply options at the compilation unit level. Fortran directives and C or C++ pragmas apply options to selected portions of code or alter the effects of command-line options.
Cray XT3™ System Overview

The following compiler commands are available:

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran 90/95</td>
<td>ftn</td>
</tr>
<tr>
<td>FORTRAN 77</td>
<td>f77</td>
</tr>
<tr>
<td>C</td>
<td>cc</td>
</tr>
<tr>
<td>C++</td>
<td>CC</td>
</tr>
</tbody>
</table>

For details about compiler command options, see the `ftn(1)`, `f77(1)`, `cc(1)`, and `CC(1)` man pages, the Cray XT3 Programming Environment User’s Guide, and the PGI user documentation.

### 3.3.4 Libraries

Developers can use PGI Fortran, C, and C++ library functions and the following Programming Environment libraries:

- Cray MPICH2, an implementation of the Message-Passing Interface 2 (MPI-2) by the Argonne National Laboratory Group. The dynamic process (spawn) functions in Cray MPICH2 are not supported at this time, but otherwise the libraries are fully MPI 2.0 compliant.

- Cray SHMEM logically shared, distributed memory access routines. Cray SHMEM routines are similar to the Cray MPICH2 routines; they pass data between cooperating parallel processes.
  
  Cray SHMEM routines can be used in programs that perform computations in separate address spaces and that explicitly pass data to and from different processing elements in the program.

  The Fortran module for Cray SHMEM is not supported. Use the `INCLUDE mpp/shmem.fh` statement instead. Also, support of Cray SHMEM atomic operations is deferred (Deferred implementation).

- 64-bit AMD Core Math Library (ACML), which includes:
  
  - Level 1, 2, and 3 Basic Linear Algebra Subroutines (BLAS)
  
  - A full suite of Linear Algebra (LAPACK) routines
A suite of Fast Fourier Transform (FFT) routines for single-precision, double-precision, single-precision complex, and double-precision complex data types

- Cray XT3 LibSci scientific libraries. Cray XT3 LibSci contains ScaLAPACK, BLACS, and SuperLU. ScaLAPACK is a set of LAPACK routines redesigned for use in Cray MPICH2 applications. The BLACS package is a set of communication routines used by ScaLAPACK and the user to set up a problem and handle the communications. SuperLU is a set of routines that solve large, sparse, nonsymmetric systems of linear equations. Cray XT3 LibSci library routines are written in C but can be called from Fortran, C, or C++ programs.

- A subset of the GNU C Language Runtime Library, glibc. For details, see the Cray XT3 Programming Environment User’s Guide.

### 3.3.5 Linker

After correcting compilation errors, the developer again invokes the compilation driver, this time specifying the application's object files (filename.o) and libraries (filename.a) as required.

The linker extracts the library modules that the program needs and creates the executable file (named a.out by default).

### 3.3.6 Runtime Environment

There are two methods of running applications: as interactive jobs and as batch jobs. There are two types of interactive jobs:

- Jobs launched directly via yod commands
- PBS Pro interactive jobs launched via the qsub -I and yod commands.

Both SPMD (single-program-multiple-data) and MPMD (multiple-program-multiple-data) applications are supported.

Before launching a job, the user can run the xtshowmesh utility to view the current state of the system. This utility displays the status of the compute nodes—whether they are up or down, designated for interactive or batch processing, and free or in use. For more information, see the xtshowmesh(1) man page and the Cray XT3 Programming Environment User’s Guide.

Some runtime components must run on a service node. To facilitate this, the yod utility executes on a service node on behalf of the application.
3.3.6.1 Interactive Jobs

To run an interactive job, the user enters the `yod` command, specifying all executables that are part of the job and the number of nodes on which they will run. The user can request a number of compute nodes or, using `xtshowmesh` data, a list of specific compute nodes.

The `yod` utility queries the Compute Processor Allocator (CPA) to get a list of nodes on which to run the job. It then contacts the process control thread (PCT) for the first node on the list. After connecting with the PCT, `yod` maps the executables, one at a time, into its address space and sets up a memory region for the PCT to fetch the executable.

In response to the load request, the PCT propagates the executable to all compute nodes selected to run that program. Once the first PCT has finished loading the executable, it maps its memory for load propagation, contacts the next set of PCTs, and forwards the load request. This process is repeated until all compute nodes have loaded. Once a compute node and all of its descendants have finished loading, the PCT starts the executable.

While the application is running, `yod` provides I/O services, propagates signals, and participates in cleanup when the application terminates.
3.3.6.2 Batch Jobs

The Cray XT3 system uses PBS Pro to launch batch jobs. PBS Pro is a networked subsystem for submitting, monitoring, and controlling a workload of batch jobs on one or more systems.

A batch job is typically a shell script and a set of attributes that provide resource and control information about the job. Batch jobs are scheduled for execution at a time chosen by the subsystem according to a defined policy and the availability of resources.

The user logs on to the Cray XT3 system and creates a script containing the `yod` command(s) to run the application. The user then enters the PBS Pro `qsub` command to submit the job to a PBS Pro server.

A PBS Pro server, executing on a login node, maintains a set of job queues.
Each queue holds a set of jobs available for execution and each job has a set of user-specified resource requirements.

The user can specify the number of compute nodes to allocate to the job. If no number is given, the job is allocated one node.

The PBS Pro scheduler is the policy engine that examines the set of ready-to-run jobs and selects the next job to run based on a set of criteria. The PBS Pro scheduler negotiates with the CPA to allocate the compute node or nodes on which the job is to run. PBS Pro communicates with a PBS Pro daemon running on the login node or a dedicated batch support node. As with interactive jobs, the `yod` utility queries the CPA to get the list of compute nodes allocated to the job, then asks the PCT to fetch the executable and propagate it to all compute nodes selected to run that program.

Figure 8. Launching Batch Jobs
3.3.7 Debugging Applications

The Cray XT3 system supports a special implementation of the Etnus TotalView debugger, available from Etnus, LLC. TotalView provides source-level debugging of applications and is compatible with the PGI Fortran 90, C, and C++ compilers.

TotalView can debug applications running on 1-360 compute nodes, providing visibility and control into each process individually or by groups. It supports access to MPI-specific data, such as the message queues.

To debug a program, the developer invokes TotalView with the `totalview` command. TotalView parses the command line to get the number of nodes, then makes a node allocation request to the CPA. TotalView directs `yod` to load but not start the application. The `yod` utility then loads the application onto the compute nodes, after which TotalView can perform initial setup before instructing `yod` to start the application.

For more information, see the Cray XT3 Programming Environment User’s Guide and Etnus TotalView documentation (Section 1.1.1, page 4).

3.3.8 Measuring Performance

Cray XT3 systems provide tools for collecting, displaying, and analyzing performance data.

3.3.8.1 Performance API (PAPI)

The Performance API (PAPI) from the University of Tennessee and Oak Ridge National Laboratory is a standard interface for access to hardware performance counters.

A PAPI event set maps AMD Opteron processor hardware counters to a list of events such as Level 1 data cache misses, data translation lookaside buffer (TLB) misses, and cycles stalled waiting for memory accesses. Developers can use the API to collect data on those events.

3.3.8.2 CrayPat

CrayPat is a performance analysis tool. The developer can use CrayPat to perform trace experiments on an instrumented application and analyze the results of those experiments. Trace experiments count an event such as the number of times a specific system call is executed.

---

1 An optional product available from Cray Inc.
The developer uses the `pat_build` command to instrument programs. No recompilation is needed to produce the instrumented program.

After instrumenting a program, the developer sets environment variables to control run time data collection, runs the instrumented program, then uses the `pat_report` command to view the resulting report.

### 3.3.8.3 Cray Apprentice$^2$

Cray Apprentice$^2$ is an interactive X Window System tool for visualizing data captured by CrayPat during program execution.\(^1\)

Cray Apprentice$^2$ can help identify and correct:

- Load imbalance
- Excessive serialization
- Excessive communication
- Network contention
- Poor use of the memory hierarchy
- Poor functional unit use

Cray Apprentice$^2$ has the following capabilities:

- It is a post-execution performance analysis tool that provides information about a program by examining data files that were created during program execution. It is not a debugger or a simulator.
- At present, Cray Apprentice$^2$ supports two types of performance data: call stack sampling and Cray MPICH2 tracing. More types of performance data will be supported in future releases.
- It reports time statistics summed across all processing elements for the whole program, as well as for individual routines.
- It shows total execution time, synchronization time, time to execute a subroutine, communication time, and the number of calls to a subroutine.
3.4 System Administration

The system administration environment provides the tools that administrators use to manage system functions, view and modify the system state, and maintain system configuration files. System administration components are a combination of Cray XT3 system hardware, SUSE LINUX, Lustre, and Cray XT3 system-specific utilities and resources.

Note: For details about standard SUSE LINUX administration, see http://www.tldp.org or http://www.suse.com. For details about Lustre functions, see the Cray XT3 System Management manual and http://www.lustre.org/documentation.html.

Many of the components used for system administration are also used for CRMS (such as powering up and down and monitoring the health of hardware components). For details about CRMS, see Chapter 4, page 31.

3.4.1 System Management Workstation (SMW)

The SMW is a server and display that provides a single-point interface to an administrator’s environment. The SMW provides a terminal window from which the administrator performs tasks like adding user accounts, changing passwords, and monitoring applications. The SMW accesses system components via the administration/CRMS network; it does not use the system interconnection network.

3.4.2 Shared-Root File System

A Cray XT3 system has a shared-root file system where the root directory is shared read-only on the service nodes. All nodes have the same default directory structure. However, the /etc directory is specially mounted on each service node as a node-specific directory of symbolic links. The administrator can change the symbolic links in the /etc directory by the process of specialization, which changes the symbolic link to point to a non-default version of a file. The administrator can specialize files for individual nodes or for a class (type) of nodes.

The administrator’s interface includes commands to view file layout from a specified node, determine the type of specialization, and create a directory structure for a new node or class based on an existing node or class. For details, see the Cray XT3 System Management manual.
3.4.3 Lustre File System Administration

The administrator can configure the Lustre file system to optimize I/O for a broad spectrum of application requirements. At one extreme is Lustre as a UFS like file system. Configuring Lustre as a UFS like file system optimizes it for access patterns of directories with many small files such as is typically seen with root file systems and often with home directories. At the other extreme is Lustre as a parallel file system. Configuring Lustre as a parallel file system optimizes it for large-scale, serial access typical of many Cray MPICH2 and Cray SHMEM applications.

When a file is created, the client contacts a metadata server (MDS), which creates an inode for the file. The inode holds metadata rather than references to the actual data. The MDS handles namespace operations, such as opening or closing a file, managing directory listings, and changing permissions. The MDS contacts Object Storage Targets (OSTs) to create data objects. The OSTs handle block allocation, enforce security for client access, and perform parallel I/O operations to transfer file data. The administrator can create and mount more than one instance of Lustre. One MDS plus one or more OSTs make up a single instance of Lustre and are managed together.

Objects allocated on OSTs hold the data associated with the file. Once a file is created, read and write operations take place directly between the client and the OST, bypassing the MDS. The OSTs use Linux ext3 file systems for backend storage. These file systems are used to store Lustre file and metadata objects and are not directly visible to the user. Lustre does not support building a parallel file system with NFS file systems.

Lustre configuration information is maintained in the SDB. For details, see the Cray XT3 System Management manual.

3.4.3.1 Lustre UFS Like File System

Lustre supports a version of UFS that features serial access to files and a uniform file space with full POSIX compliance. A single metadata server and a single OST provide the paths for data transfers between applications and RAID devices.
3.4.3.2 Lustre Parallel File System

The administrator can configure Lustre as a parallel file system by creating multiple object storage targets (OSTs). The file system optimizes I/O by striping files across many RAID storage devices.

The administrator can specify a default system-wide striping pattern at file system creation time. The administrator specifies:

- The default number of bytes stored on each OST
- The default number of OSTs across which each file is striped
3.4.4 Configuration and Source Files

The administrator uses boot nodes to view files, maintain configuration files, and manage the processes of executing programs. Boot nodes connect to the SMW and are accessible through a login shell.

The xtopview utility runs on boot nodes and allows the administrator to view files as they would appear on any node. The administrator uses this tool to coordinate changes to configuration files and software source files. All operations are logged.
The `xtopview` utility maintains a database of files to monitor as well as file state information such as checksum and modification dates. Messages about file changes are saved via a Revision Control System (RCS) utility.

### 3.4.5 System Log

Once the system is booted, console messages are sent to the system log and are written to the boot RAID system. System log messages generated by service node kernels and daemons are gathered by syslog daemons running on all service nodes. Kernel errors and panic messages are sent directly to the SMW.

The administrator can configure the syslog daemon to write the messages to different files, sorted by message generator or degree of importance.

### 3.4.6 Service Database (SDB)

A database node hosts the Service Database (SDB), which is accessible from every service processor.

The SDB, implemented in MySQL, contains the following information:

- Global state information for the CPA. CPA daemons synchronize resource allocation and store their persistent state in database tables in the Service Database. The CPA interacts with the system accounting utilities to keep track of free and allocated jobs. It logs job parameters such as start time, nodes allocated, and error conditions.

- Accounting information (processor usage, disk storage usage, file I/O demand, application execution time, etc.). The information is accessed by accounting software that accumulates resource usage information, system availability, and system utilization and generates system accounting reports.

- System configuration tables that list and describe the configuration files.

Cray XT3 system accounting is a set of utilities that run on a service node. Accounting information is collected in the Service Database (SDB). The administrator uses accounting commands to generate the following reports from the SDB:

- Job and system use
- All jobs in a time period
- System-wide statistics
3.4.7 System Activity Reports

The `sar(1)` command collects, reports, or saves system activity information for service nodes. To get system activity information (such as accounting information) for compute nodes, use the `xtgenacct(8)` command instead. For more information, see the `sar(1)` and `xtgenacct(8)` man pages.
The Cray RAS and Management System (CRMS) is an integrated, independent system of hardware and software that monitors Cray XT3 system components, manages hardware and software failures, controls startup and shutdown processes, manages the system interconnection network, and displays the system state to the administrator. The CRMS interfaces with all major hardware and software components of the system.

Because the CRMS is a completely separate system with its own processors and network, the services that it provides do not take resources from running applications. In addition, if a component fails, the CRMS continues to provide fault identification and recovery services and allows the functioning parts of the system to continue operating.

For detailed information about the CRMS, see the Cray XT3 System Management manual.

4.1 CRMS Hardware

The hardware components of CRMS are the CRMS network, the System Management Workstation (SMW), the Cray XT3 system blade control processors (L0 controllers), and the cabinet control processors (L1 controllers). CRMS hardware monitors compute and service node components, operating system heartbeats, power supplies, cooling fans, voltage regulators, and RAID systems.
4.1.1 CRMS Network

The CRMS network is an Ethernet connection between the SMW and the components that the CRMS monitors.

The CRMS network’s function is to provide an efficient means of collecting status from and broadcasting messages to system components. The CRMS network is separate from the system interconnection network.

Traffic on the CRMS network is normally low, with occasional peaks of activity when major events occur. There is a baseline level of traffic to and from the hardware controllers. All other traffic is driven by events, either those due to hardware or software failures or those initiated by the administrator. The highest level of network traffic occurs during the initial booting of the entire system as console messages from the booting images are transmitted onto the network. This level of traffic is well within the capacity of the network.
4.1.2 System Management Workstation

The system management workstation (SMW) is the administrator’s single-point interface for booting, monitoring, and managing Cray XT3 system components. The SMW consists of a server and a display device. Multiple administrators can use the SMW, locally or remotely over an internal LAN or WAN.

Note: The SMW is also used to perform system administration functions (see Section 3.4, page 25).

4.1.3 Hardware Controllers

At the lowest level of the CRMS are the Cray XT3 system L0 and L1 controllers that monitor the hardware and software of the components on which they reside.

Every compute blade and service blade has a Cray XT3 system blade control processor (L0 controller) that monitors the components on the blade, checking status registers of the AMD Opteron processors, the Control Status Registers (CSRs) of the Cray SeaStar chip, and the voltage regulation modules (VRMs). The Cray XT3 system L0 controllers also monitor board temperatures and the UNICOS/lc heartbeat.

Each Cray XT3 system cabinet has a Cray XT3 system cabinet control processor (L1 controller) that communicates with the Cray XT3 system L0 controllers within the cabinet and monitors the power supplies and the temperatures of the air cooling the blades. Each Cray XT3 system L1 controller also routes messages between the Cray XT3 system L0 controllers in its cabinet and the SMW.

4.2 CRMS Software

The CRMS software consists of software monitors; the administrator’s CRMS interfaces; and event probes, loggers, and handlers. This section describes the software monitors and administrator interfaces. For a description of event probes, loggers, and handlers, see Section 4.3, page 34.

4.2.1 Software Monitors

Resiliency communication agents (RCAs) run on all compute nodes and service nodes. RCAs are the primary communications interface between the node’s operating environment and the CRMS components external to the node. They monitor software services and the operating system instance on each node.
Each RCA provides an interface between the CRMS and the system processes running on a node for event notification, informational messages, information requests, and probing. It also provides a subscription service for processes running on the nodes. This service notifies the current node of events on other nodes that may affect the current node or that require action by the current node or its functions.

Each RCA generates a periodic heartbeat message so that the CRMS can know when an RCA has failed. Failure of an RCA heartbeat is interpreted as a failure of the UNICOS/lc operating system on that node.

RCA daemons running on each node start a CRMS process called failover manager. If a service fails, the RCA daemon broadcasts a service-failed message to the CRMS.

Failover managers on other nodes register to receive these messages. Each failover manager checks to determine if it is the backup for any failed services that relate to the message and, if it is, directs the RCA daemon on its node to restart the failed service.

4.2.2 CRMS Administrator Interfaces

The CRMS provides both a command-line and a (Deferred implementation) graphical interface.

The SMW is used to monitor data, view status reports, and execute system control functions. If any component of the system detects an error, it sends a message to the SMW. The message is logged and displayed for the administrator. CRMS policy decisions determine how the fault is handled. The SMW logs all information it receives from the system to a RAID storage device to ensure the information is not lost due to component failures.

4.3 CRMS Actions

The CRMS manages the startup and shutdown processes and event probing, logging, and handling.

The CRMS collects data about the system (event probing and logging) that is then used to determine which components have failed and in what manner. After determining that a component has failed, the CRMS initiates some actions (event handling) in response to detected failures that, if left unattended, could cause worse failures. The CRMS also initiates actions to prevent failed components from interfering with the operations of other components.
4.3.1 System Startup and Shutdown

The administrator starts a Cray XT3 system by powering up the system, booting the software on the compute nodes and service nodes, adding the booted nodes to the system interconnection network, starting the RCA daemons, and starting the compute processor allocator (CPA). The administrator stops the system by reserving, removing, and stopping components and powering off the system. For details about the startup and shutdown processes, see the Cray XT3 System Management manual.

4.3.2 Event Probing

The CRMS probes are the primary means of monitoring hardware and software components of a Cray XT3 system. The CRMS probes that are hosted on the SMW collect data from CRMS probes running on the Cray XT3 system L0 and L1 controllers and RCA daemons running on the compute nodes. In addition to dynamic probing, the CRMS provides an offline diagnostic suite that probes all CRMS-controlled components.

4.3.3 Event Logging

The event logger preserves data that the administrator uses to determine the reason for reduced system availability. It runs on the SMW and logs all status and event data generated by:

- CRMS probes
- Processes communicating through RCA daemons on compute and service nodes
- Other CRMS processes running on Cray XT3 system L0 and L1 controllers

Event messages are time stamped and logged. Before the message reaches the input queue of the event handler, an attempt is made to recover from a failure. If a compute or service blade fails, the CRMS notifies the administrator. The administrator can hot swap the blade without affecting other jobs in the system or the system as a whole (Deferred implementation). After blade replacement, the administrator re-introduces those processors into the system.
4.3.4 Event Handling

The event handler evaluates messages from CRMS probes and determines what to do about them. The CRMS is designed to prevent single-point failures of either hardware or system software from interrupting the system. Examples of single-point failures that are handled by the CRMS system are:

- Compute node failure. A failing compute node is automatically isolated and shut down. The failure affects only the application running on that node; the rest of the system continues running and servicing other applications.

- System interconnection network failure. A system interconnection network failure triggers a remapping of network paths to route around the failed component. An underlying reliability protocol ensures that no messages are lost. Only the node with the failed component is affected.

- Power supply failure. Power supplies have an N+1 configuration for each chassis in a cabinet; failure of an individual power supply will not cause an interrupt of a compute node.

In addition, the CRMS broadcasts failure events over the CRMS network so that each component can make a local decision about how to deal with the fault. For example, both the Cray XT3 system L0 and L1 controllers contain code to react to critical faults without administrator intervention.
**blade**
1) A Cray XT3 field-replaceable physical entity. A service blade consists of two AMD Opteron sockets, memory, four Cray SeaStar chips, up to four PCI-X cards, and a blade control processor. A compute blade consists of four AMD Opteron sockets, memory, four Cray SeaStar chips, and a blade control processor. 2) From a system management perspective, a logical grouping of nodes and blade control processor that monitors the nodes on that blade.

**blade control processor**
A microprocessor on a blade that communicates with a cabinet control processor via the CRMS network to monitor and control the nodes on the blade. See also blade; Cray RAS and Management System (CRMS).

**cabinet control processor**
A microprocessor in the cabinet that communicates with the CRMS via the CRMS network to monitor and control the devices in a system cabinet. See also Cray RAS and Management System (CRMS).

**cage**
A chassis on a Cray XT3 system. See chassis.

**Catamount**
The microkernel operating system developed by Sandia National Laboratories and implemented to run on Cray XT3 compute nodes. See also compute node.

**chassis**
The hardware component of a Cray XT3 cabinet that houses blades. Each cabinet contains three vertically stacked chassis, and each chassis contains eight vertically mounted blades. See also cage.

**class**
A group of service nodes of a particular type, such as login or I/O. See also specialization.
compute blade
See blade.

compute node
Runs a microkernel and performs only computation. System services cannot run on compute nodes. See also node; service node.

compute partition
The logical group that consists of all compute nodes.

compute processor allocator (CPA)
A program that coordinates with yod to allocate processing elements. See also yod.

Cray RAS and Management System (CRMS)
A system of software and hardware that implements reliability, availability, and serviceability (RAS) and some system management functions for a Cray XT3 system. The CRMS components use a private Ethernet network, not the system interconnection network. See also system interconnection network.

Cray SeaStar chip
The component of the system interconnection network that provides message routing and communication services. See also system interconnection network.

CrayDoc
Cray’s documentation system for accessing and searching Cray books, man pages, and glossary terms from a web browser.

deferred implementation
The label used to introduce information about a feature that will not be implemented until a later release.

distributed memory
The kind of memory in a parallel processor where each processor has fast access to its own local memory and where to access another processor’s memory it must send a message via the interprocessor network.
**Etnus TotalView**
A symbolic source-level debugger designed for debugging the multiple processes of parallel Fortran, C, or C++ programs.

**L0 controller**
See *blade control processor*.

**L1 controller**
See *cabinet control processor*.

**login node**
The service node that provides a user interface and services for compiling and running applications.

**metadata server (MDS)**
The component of the Lustre file system that stores file metadata.

**module**
See *blade*.

**Modules**
A package on a Cray system that allows you to dynamically modify your user environment by using module files. (This term is not related to the module statement of the Fortran language; it is related to setting up the Cray system environment.) The user interface to this package is the *module* command, which provides a number of capabilities to the user, including loading a module file, unloading a module file, listing which module files are loaded, determining which module files are available, and others.

**node**
For UNICOS/1c systems, the logical group of processor(s), memory, and network components acting as a network end point on the system interconnection network. See also *processing element*.

**object storage target (OST)**
The component of the Lustre file system that handles file activities.
**parallel processing**
Processing in which multiple processors work on a single application simultaneously.

**processing element**
The smallest physical compute group in a Cray XT3 system. The system has two types of processing elements. A compute processing element consists of an AMD Opteron processor, memory, and a link to a Cray SeaStar chip. A service processing element consists of an AMD Opteron processor, memory, a link to a Cray SeaStar chip, and PCI-X links.

**reliability, availability, serviceability (RAS)**
System hardware and software design that achieves increased system availability by avoiding or recovering from component and system failures.

**resiliency communication agent (RCA)**
A communications interface between the operating environment and the CRMS. Each RCA provides an interface between the CRMS and the processes running on a node and supports event notification, informational messages, information requests, and probes. See also Cray RAS and Management System (CRMS).

**service blade**
See blade.

**service database (SDB)**
The database that maintains the global system state.

**service node**
A node that performs support functions for applications and system services. Service nodes run SUSE LINUX and perform specialized functions. There are six types of predefined service nodes: login, IO, network, boot, database, and syslog.

**service partition**
The logical group of all service nodes.
specialization
The process of setting files on the shared-root file system so that unique files can be present for a node or for a class of node.

system interconnection network
The high-speed network that handles all node-to-node data transfers.

system management workstation (SMW)
The workstation that is the single point of control for the CRMS and Cray XT3 system administration. See also Cray RAS and Management System (CRMS).

UNICOS/lc
The operating system for Cray XT3 systems.

xtshowmesh
A utility that identifies the state of the nodes.

yod
Application launching utility. See also compute processor allocator (CPA).
Index

A
Accessing the system, 17
Accounting, 29
ACML, 18
Administration, 25, 33
Application development tools, 2
Application launch, 19
Authentication, 16

B
Batch jobs, 19, 21
BLACS, 19
Blade, 3, 11
   compute, 11
   service, 12
Blade control processor, 8, 31, 33
Board (processor)
   See blade
Boot nodes, 15, 28
Boot service nodes, 14

C
C and C++ interlanguage communications, 17
C command, 17
C++ command, 17
Cabinet, 11–12
Cabinet control processor, 31, 33
Catamount, 16
Chassis, 11–12
Compiler commands, 17
Compilers
   C and C++, 17
   Fortran, 17
Compute blade, 11
Compute node, 10
Compute partition, 10
Compute Processor Allocator (CPA), 20, 29
Control Status Register (CSR), 33
Controller
   L0, 31
CPU
   Opteron processor, 7
Cray LibSci, 19
Cray MPICH2, 18
Cray SHMEM, 18
CRMS, 31
CRMS actions, 34
CRMS event handling, 36
CRMS event logging, 35
CRMS event probing, 35
CRMS hardware, 31
CRMS network, 32
CRMS software, 33

D
Database nodes, 29
Debugging, 23
Development environment, 16
Development tools, 2
DIMMs, 8
Disk storage, 3, 14

E
Etnus TotalView, 23

F
Failover Manager, 33
Fibre Channel, 13
File system
   Lustre, 2, 26
   Object Storage Target, 26
   UFS, 2
Fortran commands, 17
Fortran interlanguage communications, 17
Cray XT3™ System Overview

H
Hardware components, 7
Hardware controllers, 33
HyperTransport Link, 8

I
I/O nodes, 15
I/O service nodes, 13
I/O system, 13
Interactive jobs, 19–20
Interconnect, 2, 9
Interlanguage communications, 17

J
Jobs
batch, 19, 21
interactive, 19

L
L0 controller, 8, 33
L1 controller, 33
Libraries, 18
AMD Core Math Library (ACML), 18
BLACS, 19
BLAS, 18
Cray MPICH2, 18
Cray SHMEM, 18
FFT, 19
glibc, 19
LAPACK, 18
math, 18
ScalAPACK, 19
scientific, 18
SuperLU, 19
LibSci, 19
Linker, 19
Linux, 15
Logging in, 17
Login nodes, 15
Login service nodes, 14
Lustre
parallel file system, 26–27
shared-root file system, 25
UFS like file system, 26
Lustre file system, 26

M
Memory, 8
Chipkill memory protection, 1, 7
Direct Memory Access (DMA), 8
Memory model, 1
Message passing
Portals interface, 8
Metadata server (MDS), 26
Microkernel, 16
Module, 11
Module (processor)
See blade
Modules utility, 17
MPP, 1

N
Network
CRMS, 3
node-to-node communication, 2
system interconnection network, 9
Network nodes, 15
Network router, 8
Network service nodes, 13
Node, 1, 9
calculate, 10
Lustre service, 10

O
Object Storage Target (OST), 26
Operating system, 2, 15

P
Parallel file system, 27
Parallel programming models, 2
PBS Pro, 21
PE, 9
Performance API (PAPI), 23
Performance counters, 7
Performance measurement, 23
Portals interface, 8, 16
Process control thread (PCT), 20
Processing element (PE), 9
Processor, 7

Q
quintessential kernel (Qk), 2

R
RAID, 3
RAS (reliability, availability, serviceability), 3
Reliability, Availability, Serviceability (RAS), 31
Resiliency communication agent (RCA), 33
Router, 8
Running applications, 19

S
ScalAPACK, 19
SeaStar chip, 8
Secure shell, 16
Service blade, 12
Service Database (SDB), 15, 29
Service node, 10, 15
  boot, 14, 28
database, 29
I/O, 13
login, 14
network, 13
Service node classes
  boot, 15
  I/O, 15
  login, 15
  network, 15
Service Database SDB, 15
  syslog, 15
Service partition, 11
Shared-root file system, 25
Single-system view, 3
SMW, 14
SMW software, 34
Storage
  RAID, 14
Striping, 27
SuperLU, 19
SUSE LINUX, 15
Syslog, 29
Syslog nodes, 15
System administration, 25
System interconnection network, 2, 9
System Management Workstation (SMW), 14, 25, 33

T
TotalView, 23

U
UFS, 26
UNICOS/lc
  microkernel, 16
UNICOS/lc operating system
  Catamount, 16
  SUSE LINUX, 15
User environment, 16

X
  xtshowmesh utility, 19

Y
  yod utility, 20