Cray XT5™ System Overview
S–2472–10
### Record of Revision

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.0 LA  | December 2007  
Draft documentation to support limited availability (LA) release of Cray X2 compute blades for Cray XT5\(_h\) systems running the Cray X2 Programming Environment 6.0, System Management Workstation (SMW) 3.1, and UNICOS/lc 2.0.33 or later releases. |
| 1.0 GA  | July 2008  
Supports the general availability (GA) release of Cray X2 compute blades for Cray XT5\(_h\) systems running the Cray X2 Programming Environment 6.0, System Management Workstation (SMW) 3.1.06, and UNICOS/lc 2.0.53y or later releases. |
## Contents

### Preface
- Accessing Product Documentation .......................................................... vii
- Conventions ......................................................................................... viii
- Reader Comments .................................................................................. ix
- Cray User Group .................................................................................. ix

### Introduction [1]
- Feature Summary .................................................................................. 2
- Related Publications ............................................................................ 4
  - Publications for Application Developers .......................................... 4
  - Publications for System Administrators ......................................... 5

### Hardware Overview [2]
- Basic Cray X2 Hardware Components ................................................ 8
  - Compute Nodes ................................................................................. 8
  - Local Memory .................................................................................... 10
  - Network Router ............................................................................... 12
    - YARC Router Chip ........................................................................ 12
    - Fat-tree Topology ......................................................................... 13
  - StarGate / Bridge Blade Interface ................................................... 13
  - Hardware Supervisory System ......................................................... 14
    - Cray System Management Workstation ...................................... 15
    - Private Ethernet Network ............................................................. 16
- Physical Components ........................................................................... 16
  - Blades and Modules ........................................................................ 16
  - Compute Blade .................................................................................. 16
  - Bridge Blade ...................................................................................... 17
<table>
<thead>
<tr>
<th>System Overview</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Modules and Blades</td>
<td>18</td>
</tr>
<tr>
<td>Chassis and Cabinets</td>
<td>18</td>
</tr>
<tr>
<td>System Reliability and Management</td>
<td>18</td>
</tr>
<tr>
<td>Software Overview [3]</td>
<td>21</td>
</tr>
<tr>
<td>Cray X2 Linux Kernel</td>
<td>21</td>
</tr>
<tr>
<td>Lustre File System Client</td>
<td>21</td>
</tr>
<tr>
<td>System Monitoring and Control</td>
<td>22</td>
</tr>
<tr>
<td>Hardware Supervisory System</td>
<td>22</td>
</tr>
<tr>
<td>Mazama System Administration Software</td>
<td>23</td>
</tr>
<tr>
<td>Command and Control</td>
<td>24</td>
</tr>
<tr>
<td>Monitoring</td>
<td>24</td>
</tr>
<tr>
<td>Image Management</td>
<td>24</td>
</tr>
<tr>
<td>Partition Management</td>
<td>24</td>
</tr>
<tr>
<td>Administrative States</td>
<td>25</td>
</tr>
<tr>
<td>Reliability, Availability, and Serviceability</td>
<td>25</td>
</tr>
<tr>
<td>Network Management</td>
<td>25</td>
</tr>
<tr>
<td>Configuration</td>
<td>25</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>26</td>
</tr>
<tr>
<td>Cray X2 and Cray XT Common Software Components [4]</td>
<td>27</td>
</tr>
<tr>
<td>Operating System</td>
<td>27</td>
</tr>
<tr>
<td>User Environment Services</td>
<td>28</td>
</tr>
<tr>
<td>File Systems</td>
<td>29</td>
</tr>
<tr>
<td>System Administration</td>
<td>29</td>
</tr>
<tr>
<td>System Management Workstation</td>
<td>30</td>
</tr>
<tr>
<td>Shared-root File System</td>
<td>30</td>
</tr>
<tr>
<td>Lustre File System Administration</td>
<td>30</td>
</tr>
<tr>
<td>Configuration and Source Files</td>
<td>31</td>
</tr>
<tr>
<td>System Log</td>
<td>32</td>
</tr>
<tr>
<td>Service Database</td>
<td>33</td>
</tr>
</tbody>
</table>
## Cray XT\textsuperscript{TM} System Overview

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 6.</td>
<td>StarGate Interconnect</td>
<td>14</td>
</tr>
<tr>
<td>Figure 7.</td>
<td>Hardware Supervisory System Block Diagram</td>
<td>15</td>
</tr>
<tr>
<td>Figure 8.</td>
<td>Cray X2 Bridge Blade</td>
<td>17</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:

**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.
Conventions

These conventions are used throughout Cray documentation:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>command</strong></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><strong>variable</strong></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><strong>user input</strong></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. Section numbers are used to group man pages into categories, as defined by the usage of the commands in that section. For example, section 1 man pages are typically user commands and section 8 man pages are typically system administrator commands. To find the meaning of each section number for your particular system, enter the following command:</td>
</tr>
</tbody>
</table>

```
% man man
```
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

Cray User Group

The Cray User Group (CUG) is an independent, volunteer-organized international corporation of member organizations that own or use Cray Inc. computer systems. CUG facilitates information exchange among users of Cray systems through technical papers, platform-specific e-mail lists, workshops, and conferences. CUG memberships are by site and include a significant percentage of Cray computer installations worldwide. For more information, contact your Cray site analyst or visit the CUG website at www.cug.org.
This document provides an overview of the Cray X2 1.0 software release for Cray XT5\textsuperscript{h} systems. The release of Cray XT5\textsuperscript{h} systems marks the beginning of Cray’s Adaptive Supercomputing vision; it provides a software infrastructure for Cray systems now and in the future. The intended audiences are application developers and system administrators. Prerequisite knowledge is a familiarity 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 X2 blades provide second-generation scalable vector systems. They build on the technology introduced with Cray X1 and Cray X1E systems but include an enhanced instruction set, improved scalar performance, and a peak performance of more than 200 GFLOPS (100 GFLOPS per node) (gigaFLOPS, one billion floating-point operations per second).

As with other Cray systems, this system combines commodity and open-source components with custom-designed components.

The Cray X2 compute processors (also called CPUs) are Cray proprietary vector CPUs, delivered as four-way SMP compute nodes. **Chapter 2, page 7** provides details on the architecture of Cray X2 compute blades.

The Cray X2 blades are connected with a proprietary Cray interconnect, called YARC, which enables systems to be scaled using a fat-tree topology. **Fat-tree topology** is a tree structure with redundant interconnections on its branches; the number of interconnections increases as the root is neared. The purpose is to increase the bandwidth at higher levels, where it is most needed. The system design enables the Cray X2 nodes to be scaled to many CPUs. For larger systems, special router cabinets are also used as part of this system scaling.
The blades of the Cray X2 interconnect with a Cray XT system, and rely on the Cray XT system for providing service and I/O functions. The Cray XT system can also contain optional scalar compute blades in a separate partition. The two systems are connected with a high-speed interface called StarGate, which serves as a bridge between the Cray XT system SeaStar or SeaStar2 interconnect and the YARC interconnect of the Cray X2 blades.

**Note:** In this guide, Cray X2 refers only to the vector compute nodes, the YARC network, and the StarGate interface. The Cray XT system includes the optional scalar compute partition, the I/O service partition, the SeaStar network, and all I/O devices connected to Cray XT cabinets. Together, this hybrid system is known as the Cray XT5h.

The Cray X2 compute node operating system is based on a distributed architecture, where most services are provided by service-I/O (SIO) nodes on the adjoining Cray XT system. The Cray X2 compute node, like the Cray XT CNL compute node, is configured with a minimal OS capability, designed to limit the impact of OS activities on user applications. Each Cray X2 compute node runs a Linux kernel configured with a minimum of other system software. The primary functions of the operating system on the compute nodes is to initiate programs, manage node memory, and terminate programs.

The programming environment supports Fortran, C, and C++ languages, plus a wide variety of parallel programming models, including MPI, Co-Array Fortran, UPC, and OpenMP (single node only, mixed models are not supported). The programming environment is an evolution of the mature Cray X1 and Cray X1E programming environment.

### 1.1 Feature Summary

Cray XT5h features include:

- SLES 10 Kernel
- Large page support
- Resiliency features, including error detection and logging, ability to survive a compute SMP failure, and firewall support
• Scaling up compute processors

Release 1.0 supports the following configurations:

– A single cabinet with up to 15 compute blades
– Two compute cabinets and one router cabinet with up to 30 compute blades
– Six compute cabinets and one router cabinet with up to 90 compute blades

• Accounting features:

– Application time reporting
– Linux process accounting
– (Deferred implementation) Comprehensive System Accounting (CSA), open-source software that includes changes to the Linux kernel to collect more types of system resource usage data than under standard Fourth Berkeley Software Distribution (BSD) process accounting

• Application core files
• Application global address space
• CPU and memory performance counters
• Dump and crash support
• Limited compute node commands
• Application Level Placement Scheduling (ALPS) features, including aprun launch, application monitoring and management, resource reservation, ALPS interface to batch subsystems (for example, PBS Pro), and context switching (sometimes called gang scheduling) (Deferred implementation)

• Support for Lustre Linux clients
• IP support
• RSIP client support
• Mazama managers (see Section 3.3.1, page 22 for more information)

For more information about Cray XT5h features, see the Cray XT5h Software Release Overview. For more information about programming environment features, see the Cray Programming Environment Releases Overview and Installation Guide.
1.2 Related Publications

The Cray XT5™ proprietary, third-party, and open-source products are documented in the following publications.

1.2.1 Publications for Application Developers

- *Cray XT5h System Overview* (this manual)
- *Getting Started on Cray X2 Applications*
- *Cray XT5h Software Release Overview*
- *Cray Fortran Reference Manual*
- *Cray C and C++ Reference Manual*
- Modules utility man pages: `module(1), modulefile(4)`
- Commands related to application launch: `aprun(1), apstat(1), apkill(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
- Cray LibSci FFT man pages (read the `intro_fft(3)` man page first)
- *SuperLU Users’ Guide*
- *PBS Professional 8.0 Quick Start Guide* ¹
- *PBS Professional 8.0 User Guide* ¹
- *PBS Pro man pages* ¹
- *PBS Professional 8.0 External Reference Specification* ¹
- *PAPI User’s Guide*
- PAPI man pages

¹ PBS Pro is an optional product from Altair Grid Technologies available from Cray Inc. See [http://www.altair.com](http://www.altair.com).
• **PAPI Programmer’s Reference**
• **PAPI Software Specification**
• **Using Cray Performance Analysis Tools**
• CrayPat man pages: ² (start with intro_craypat(1))
• Cray Apprentice2 man page (app2(1)) ²
• Cray X2 Linux kernel man pages
• UNICOS/lc man pages
• System Management Workstation (SMW) man pages
• SUSE LINUX man pages
• Linux documentation (refer to the Linux Documentation Project at http://www.tldp.org and to SUSE documentation at http://www.suse.com)
• Lustre documentation (refer to Lustre documentation at http://www.sun.com/software/products/lustre and http://www.lustre.org)

### 1.2.2 Publications for System Administrators

• **Cray XT5h System Overview** (this manual)
• **Cray System Management Workstation (SMW) Software Installation Guide**
• **Cray XT System Management**
• **Cray XT5h Installation, Configuration, and Management Supplement**
• **Cray XT5h Software Release Overview**
• **Cray Programming Environment Releases Overview and Installation Guide**
• **SUSE LINUX man pages**
• Lustre documentation (refer to Lustre documentation at http://www.sun.com/software/products/lustre and http://www.lustre.org)
• **PBS Pro Release Overview, Installation Guide, and Administration Addendum** ³

² An optional Cray product.
³ PBS Pro is an optional product from Altair Grid Technologies available from Cray Inc. (see http://www.altair.com).
• *PBS Professional 8.0 Administrator Guide* ³

• Linux documentation (refer to the Linux Documentation Project at [http://www.tldp.org](http://www.tldp.org) and to SUSE documentation at [http://www.suse.com](http://www.suse.com)).
The Cray XT5 system represents a fusion of vector and scalar technologies in a heterogeneous environment. Cray X2 vector processors are integrated with a scalar Cray XT environment over a custom high-speed interconnect. These systems interoperate through a protocol bridge chip (StarGate) housed on a bridge blade that allows data traffic to transfer between the networks of the two systems. The bridge blade plugs directly into the Cray X2 mechanical cabinet and connects to the Cray XT system using network cables, as shown in Figure 1.

Figure 1. Integrated Cray X2 and Cray XT Components
2.1 Basic Cray X2 Hardware Components

Basic Cray X2 hardware-processing components include:

- Compute nodes
- Local memory
- Network router
- StarGate interface
- Hardware Supervisory System (HSS)

2.1.1 Compute Nodes

Cray X2 blades are built with four-way SMP nodes sharing a global address space. Each node has four CPUs and local node memory. The CPU is implemented on a single chip that includes L1 and L2 caches and four bidirectional network interface (NIF) ports. Local memory is comprised of 16 memory daughter cards (MDC), each of which contains a Weaver memory controller with a distributed L3 cache. Each CPU has uniform memory access to local node memory using a narrow high-speed memory channel to each of the 16 MDCs. Figure 2, page 9 shows the layout of a compute blade containing two nodes.
Figure 2. Cray X2 Compute Blade
Access to remote node memory is nonuniform across a high bandwidth, low latency, node interconnection network via the NIF ports. Caches are kept coherent across the entire system. Based on powerful vector processors, each node is capable of more than 100 GFLOPS peak performance in 64-bit mode. Figure 3 shows the layout of a node.

**Figure 3. Cray X2 Node Architecture**

### 2.1.2 Local Memory

Each node contains 16 memory daughter cards that contain the node’s local memory. Each MDC contains one Weaver chip driver that drives 20 DDR II SDRAM memory packages, arranged as four channels of 5 packages. Each channel is 40 bits wide, comprised of five 8-bit wide packages. The 40-bit width includes 32 bits of data, 7 bits of ECC, and 1 spare bit. **Figure 4, page 11** shows the configuration of a Cray X2 memory daughter card.
The Weaver memory controller chips provide two primary functions. First, they serve as pin expanders, converting a small number of high-speed channels from the processors into a large number of low-speed signals to interface to commodity DDR SDRAM chips. Second, they implement an 8 MB shared L3 cache and cache coherence directories.
2.1.3 Network Router

The Cray X2 processors are connected with a proprietary Cray interconnect containing a YARC chip, which enables systems to be scaled using a fat-tree topology. Figure 5 shows a rank 1 network router.

![Network Router Diagram](image)

Figure 5. Rank 1 Network Router

The system design enables Cray XT5 systems to be scaled to many thousands of CPUs. For larger systems, special router cabinets are used as part of this system scaling.

2.1.3.1 YARC Router Chip

The YARC router is a single chip with 64 ports. Each port supports a bidirectional link pair (rail) with an aggregate bandwidth of 3.8 GBs, or 1.9 GBs in each direction.

Internally, YARC is arranged as an 8x8 array of tiles, each containing a link control buffer (LCB) for reliable input and output.
2.1.3.2 Fat-tree Topology

Each Cray X2 processor has a network injection port into the network. An injection port consists of four bidirectional links that connect to the rank 1 routers. The fat-tree network is composed of one, two, or three ranks of routers, with the rank indicating distance from the processors. The smallest networks require only rank 1 (R1) routers. Larger networks require rank 2 (R2) or rank 3 (R3) routers.

2.1.4 StarGate / Bridge Blade Interface

Cray X2 components and Cray XT systems are tightly connected using a high-speed network bridge called StarGate.

The StarGate chip is a protocol bridge chip that provides the conversion between the Cray XT SeaStar 3-D network and Cray X2 YARC fat-tree network. User login nodes are housed in the Cray XT cabinets and can launch applications on the Cray XT compute blades or the Cray X2 compute blades. All Cray X2 input/output packets are directed to a StarGate. The StarGate chip provides the necessary protocol conversion between the YARC network protocol and the SeaStar network protocol.

The StarGate chip enables I/O and network traffic to seamlessly travel between the two systems. This connection is a network bridge, not a processor to processor interface. Figure 6, page 14 shows how the StarGate connects the two networks.
2.1.5 Hardware Supervisory System

The Hardware Supervisory System (HSS) monitors the hardware components of the system and proactively manages the health of the system. It communicates with nodes and with the management processors over the private Ethernet network. Figure 7, page 15 shows a block diagram of the HSS. The following subsections provide more detail on the components.
2.1.5.1 Cray System Management Workstation

The Cray System Management Workstation (SMW) connects to several controllers via a private Ethernet network. The SMW provides a single-point interface for booting, monitoring, and managing Cray system components. Multiple administrators can use the SMW, locally or remotely over an internal LAN or WAN. The SMW is an Opteron-based server running Linux.
2.1.5.2 Private Ethernet Network

The SMW is connected to each cabinet with a fast Ethernet connection via a 10/100 switch. This is a private Ethernet network, used only by the SMW and HSS.

The HSS hardware is implemented with a controller at each hierarchical level of hardware. This scheme provides an efficient method of producing and consuming control data.

2.2 Physical Components

This section describes the main Cray X2 physical components of the Cray XT5\textsubscript{h} system and their configurations.

2.2.1 Blades and Modules

*Blades* and *modules* are field-replaceable units containing some combination of compute processors, router chips, memory, power regulators, and connectors.

The Cray X2 compute blades and bridge blades are implemented in vertically mounted blades into the chassis; they provide power and connectivity to the interconnection network. Modules are horizontally mounted.

2.2.1.1 Compute Blade

A Cray X2 compute blade contains two Cray X2 compute nodes. The two nodes implemented on a Cray X2 blade use the YARC interconnect in the same manner as nodes on separate blades. Each blade contains the eight Cray X2 chips (CPU and network access links), two voltage regulator modules (VRM) per CPU, 32 memory daughter cards, a blade controller for supervision, and a back panel connector.

Each memory daughter card contains 20 DDR2 SDRAM memory chips (1 Gb) and one Weaver control chip. The 20 memory chips include 16 data chips and 4 parity chips to implement full single error correction/double error detection (SECDED).

Memory available for the Cray X2 blade is 64 or 128 GB, which is divided evenly between the two nodes, providing 32 or 64 GB for each four-way node. The 128 GB memory is implemented with stacked versions of 1 Gb memory chips.
2.2.1.2 Bridge Blade

A Cray X2 bridge blade connects the Cray X2 components with its supporting Cray XT system. The blade includes eight StarGate chips, and is installed in a Cray X2 compute chassis in place of a compute blade. A single bridge blade can deliver total sustained bandwidth between the two systems of approximately 8 GBs (1 GBs per StarGate chip). Cray recommends that bridge blades be installed in pairs in adjacent slots in the same chassis, because the bridge blade runs the network at a slower speed than do compute blades; having bridge and compute blades in the adjacent slots hurts performance of the latter. Multiple pairs of bridge blades can be installed for greater bandwidth. Figure 8 shows the configuration of the bridge blade.

Note: When estimating the peak performance of Cray XT5h system, take into account that cabinets containing bridge blades have fewer compute blades, and thus lower peak performance, than cabinets without bridge blades.

Figure 8. Cray X2 Bridge Blade
2.2.1.3 Router Modules and Blades

Two types of router module/blades exist:

- One provides rank 1 (R1) router connections within the compute cabinet; this is called a router module. The R1 router module contains two YARC chips.

- The other, called a router blade, is used to build the R2 and R3 connections and is housed in a separate router cabinet. The R2 and R3 router blades contain four YARC chips.

2.2.2 Chassis and Cabinets

Chassis and cabinets house blades and modules, as follows.

- A chassis is a blade enclosure together with power distribution and four R1 router cards. A chassis houses from one to eight blades (either compute or bridge blades).

- Cabinets contain chassis or router modules:
  - A compute cabinet contains two chassis.
  - A router cabinet has three chassis and contains a router backplane housing up to 16 router modules.

2.3 System Reliability and Management

In the Cray XT5\textsubscript{h} system, Cray X2 component design, component selection, and system environment minimizes component failures. The Cray X2 components tolerate the following failures if they do occur:

- Error correction coding (ECC) in memory provides single-bit error correction and double-bit error detection.

- CRC-based error checking on network links with retry and graceful degradation when multiple links are used.

- N+1 redundant power supplies.
These and other features provide a significant level of fault tolerance. The system isolates and contains failures to allow user tasks to continue in the presence of component failures in unrelated parts of the system. The system management network uses Ethernet, PCI, I²C, and JTAG (Joint Test Access Group) to connect the components of the system, providing system monitoring capability.
Most services for the Cray X2 compute nodes are provided by service-I/O (SIO) nodes on the adjoining Cray XT system. For more information about the common software components, see Chapter 4, page 27.

3.1 Cray X2 Linux Kernel

Cray X2 compute nodes are configured with a minimal operating system capability to limit the impact of operating system activities on user applications. Each compute node runs a Cray X2 Linux kernel called CNL that is configured with a minimum of other system software. The primary function of CNL on the compute nodes is to initiate programs, manage node memory, and terminate programs. The other system software includes a client for the Lustre file system and daemons used to communicate with the job scheduler, Application Level Placement Scheduler (ALPS).

3.2 Lustre File System Client

The Lustre file system is used as the local file system for the Cray XT5h system. Lustre is a highly scalable and highly modular file system designed for large systems. It combines established, open standards with the Linux operating system and innovative protocols into a reliable, network-neutral data storage and retrieval system. Lustre provides high I/O throughput in clusters and shared-data environments. Features include independence from the physical location of data, protection from single points of failure, and fast recovery from downtime stemming from cluster reconfiguration or server/network outages.

Each Cray X2 node has a Lustre client, and I/O nodes within the Cray XT system run the Lustre Meta Data Servers (MDSs), Object Storage Servers (OSSs), and Object Storage Targets (OSTs). Lustre is a 64-bit file system. As such, the size of files and file systems is limited only by the use of 64-bit fields. Lustre includes comprehensive recovery features and takes advantage of journaling features in the underlying native file system to ensure the consistency of the data actually stored on disk managed by the MDSs, OSSs, and OSTs.
The Lustre framework enables you to structure your file system installation to match your data transfer requirements. One MDS plus one or more OSTs make up a single instance of Lustre and are managed together. Client nodes mount the Lustre file system over the network and access files with POSIX file system semantics. Each client mounts Lustre, uses the MDS to access metadata, and performs file I/O directly through the OSTs.

For more information about Lustre and how data is stored in a Lustre file system, see *Cray XT System Management*.

### 3.3 System Monitoring and Control

Two subsystems provide system monitoring and control for Cray X2 components of Cray XT5™ systems:

- Hardware Supervisory System (HSS)
- Mazama system administration software

#### 3.3.1 Hardware Supervisory System

In addition to the high-speed interconnect between Cray X2 components and Cray XT systems, a Hardware Supervisory System (HSS) exists among all major hardware components. This subsystem enables command and control operations to be communicated via the HSS network to the hardware controllers to enable the system to be administered.

The HSS exists on each cabinet control processor and is responsible for in-cabinet control and monitoring component hardware. The administrative interface to this subsystem is via the event mechanism and by database references. Each cabinet control processor maintains an independent database of components and state.

An aggregate system view of various HSS components is available on the System Management Workstation (SMW). Changes in component state are communicated to the Mazama software by database table updates and events. It is the responsibility of the event router to propagate the notification event to all registered programs, one of which is Mazama. Events can be generated by Mazama, when a state change is administratively required (boot), or by the HSS when a component state change is monitored (for example, a voltage drop).
The HSS also interacts with the Resiliency Communication Agent (RCA) component of the kernel; RCA is the communication interface between the operating environment and HSS. RCA enables HSS to provide the following services for Cray X2 components:

- A message path to and from the operating environment, for events, for informational messages, and for information requests.
- A periodic heartbeat message, used so that the HSS can know when the operating environment has failed and needs to be restarted (if possible).
- A probe function for software components on its node.

### 3.3.2 Mazama System Administration Software

Mazama, the Cray X2 blades administration framework, integrates both hardware and software environments to provide monitoring and administrative functionality of the Cray X2 components. Mazama provides software tools for tasks such as booting nodes, asserting routes, and collecting log information. The software provides a hierarchy of control and authority that originates on the SMW, delegating control to local agents such as HSS cabinet and blade controllers.

The main purpose and benefit of the Mazama software is to provide a single point of control and administration for Cray X2 components. Despite multiple copies of the operating system (possibly hundreds in large systems), you can easily monitor and manage Cray X2 components. The Mazama software automates many of the tasks that are typically necessary when you manage many nodes in a system of this size. It permits you to partition the system into a number of logical regions, each with a common set of attributes, and manage them through a common interface.

This section does not describe specific Mazama commands; it describes the components of the Mazama architecture.

Mazama includes a configuration database that describes all the hardware and software used within Cray X2 components. Mazama also includes logging and error handling for Cray XT components.

Although you install and run Mazama on the SMW, most Mazama commands affect only the Cray X2 components. In contrast, most administrative commands affect only Cray XT system components; however, `xtdumpsys(8)` and `xtbootsys(8)` can affect both systems. See the appropriate man page for more information.
This section describes the main features of Mazama that support system administrators access, monitoring, and control of Cray X2 components. Commands enable these features; for information about specific Mazama commands, see Cray XT5h Installation, Configuration, and Management Supplement.

3.3.2.1 Command and Control

The command and control component enables the system administrator to control Cray X2 hardware components. An administrator can send commands to the cabinet control processors. When a command is sent to the cabinet control processors (for example, a boot command), the cabinet control processors determine the internals of boot, complete the operation, and report the status back to the administration server.

3.3.2.2 Monitoring

Mazama monitoring is the ability to display the following:

- System state, for example, node state
- Network statistics
- Software and hardware errors throughout the system
- System usage, for example, Cray X2 nodes running jobs

3.3.2.3 Image Management

Image management enables the system administrator to create or upgrade OS distributions for deployment within Cray X2 blades. The system administrator creates base OS distributions (releases of Cray X2 software) on the SMW, copies images from the installed master copy, and applies site modifications. This modified image is then packaged as Cray X2 bootable archived for deployment to the cabinet controller, and the cabinet control processor software forwards these images to Cray X2 nodes for booting.

3.3.2.4 Partition Management

The administrator can assign compute nodes, OS distributions, and kernels to a single entity using partition management. The administrator can then use this entity for administration purposes such as startup, shutdown, or status query.
3.3.2.5 Administrative States

Mazama displays and sets various processor flags for Cray X2 processors; these values are called administrative states. Administrative states control how the Cray X2 processors are allocated by the batch scheduler. Valid states include batch, interactive, available, and unavailable. After setting processor flags, the updated status of the affected processors is made available to ALPS.

States can be set on an individual node basis, by partition grouping, or on all nodes within the system.

3.3.2.6 Reliability, Availability, and Serviceability

Cray X2 components provide a system of software and hardware that implements reliability, availability, and serviceability (RAS). RAS components enable administrators to:

- Monitor and report on hardware and software components.
- Monitor all logs and errors generated within the system.
- Filter all messages generated within the cabinet and, based on site-defined policies, send them back to the SMW for storage and analysis.

3.3.2.7 Network Management

Network management is the administration of the internal/external interfaces, network, and topological routing. It includes managing the high-speed network interfaces (site-defined base address, self-calculating node addresses), managing the external interfaces, and routing within and external to the system. It also provides hostname resolution within the system.

3.3.2.8 Configuration

Configuration provides a system view of the configuration, control, and status. It includes a complete view of administration data within the system, in terms of a logical view (partitions, processors, images) and a physical view (represented by HSS).
3.3.2.9 Infrastructure

*Infrastructure* provides an extensible and scalable framework for administration and control based on the Cray XT event router. All managed hardware components must have access to the event stream, thus providing a network between all components for command and control functions.
Cray XT and Cray X2 components run a combination of Cray developed software, third-party software, and open-source software. System software provides the functions needed to support users, applications, and administrators. It comprises a full-featured operating system, user environment services, file systems, I/O functions, and system administration tools. System software runs on Cray XT service nodes.

Cray XT application software and OS kernels run on Cray XT compute nodes. Cray X2 application development software and the OS kernel run on Cray X2 compute nodes. Compilation drivers reside on either the Cray XT login node or on a stand-alone Linux system, depending on the site’s choice of PE packages.

This chapter provides an overview of the common system software and processes. Chapter 5, page 35 describes the Cray X2 application development tools and processes.

4.1 Operating System

The Cray XT operating system is UNICOS/lc. UNICOS/lc consists of service-node and compute-node components. This chapter describes the service-node components. The compute-node components (kernels) are described in Cray XT Series System Overview.
Service nodes run a full-featured operation system derived from SUSE LINUX. Above the operating-system level are specialized daemons and applications that perform functions unique to each of the following service node types:

- Login nodes provide user login, user environment management services, and host-batch processing resource managers.
- I/O nodes manage file system metadata and transfer data to and from storage devices and applications.
- Network nodes provide Transmission Control Protocol/Internet Protocol (TCP/IP) connections to external systems.
- System nodes perform special services such as system boot, system administration, Service Database (SDB) management, and syslog management. A system administrator can reconfigure service nodes to improve system efficiency. For more information about configuration options, see *Cray XT System Management* and *Cray XT System Software Installation and Configuration Guide*.

### 4.2 User Environment Services

Cray XT5h system users log in from a standard terminal window to Cray XT login nodes. The user environment on the login nodes 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 the Cray XT system:

1. Set up a secure shell. The Cray XT system uses `ssh` and `ssh-enabled` applications for secure, password-free remote access to login nodes. Before using `ssh` commands, the user must generate an RSA authentication key.

2. Load the appropriate modules. The Cray XT system uses 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, the Modules utility adds them automatically to the Programming Environment, and retains earlier versions 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 XT Programming Environment User’s Guide* and the `module(1) and modulefile(4)` man pages.

The system distributes logins among login nodes using a load-leveling service that intercepts login attempts and directs them to the least used login node.
Login nodes provide all of the standard Linux utilities and commands, a variety of shells, and access to application development tools.

### 4.3 File Systems

I/O nodes host the file system. The Cray XT system runs Lustre, a high-performance, highly scalable, POSIX-compliant shared file system. Lustre is hosted on Linux and uses the Portals lightweight message passing API and an object-oriented architecture for storing and retrieving 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 one or more Object Storage Targets (OSTs). Applications use Lustre OSTs to transfer data objects; these data objects can be striped across RAID storage devices.

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 4.4.3, page 30.

### 4.4 System Administration

The Mazama system administration environment provides the tools that administrators use to manage Cray X2 node functions, view and modify the system state, and maintain system configuration files. System administration components are a combination of Cray XT system hardware, SUSE LINUX, Lustre, and Cray XT system utilities and resources.


Many of the components used for system administration are also used for system monitoring and management (such as powering up and down and monitoring the health of hardware components). For details, see the [Cray XT5h Installation, Configuration, and Management Supplement](http://www.sun.com/software/products/lustre) manual.
4.4.1 System Management Workstation

The System Management Workstation (SMW) is a server and display that provides a single-point interface to an administrators environment. The SMW accesses system components through the administration network; it does not use the system interconnection network.

Note: A system that contains Cray X2 compute nodes requires SMW release 3.1 or later.

For more information about the SMW, see the Cray XT System Management manual.

4.4.2 Shared-root File System

The Cray XT 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 nondefault version of a file. The administrator can provide specialized 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 XT System Management manual.

4.4.3 Lustre File System Administration

Three software components of Lustre can run across all or selected nodes of the Cray XT system.

- Clients

Clients are services or programs that access the file system. You do not set up clients as part of the Lustre setup, but you must specify the nodes on which they run.
• Object storage target (OST)

You can configure one or more object storage target (OST)s, which are software interfaces to backend storage volumes. The OST is a logical device. The OSTs handle file data and enforce security for client access. Object storage targets provide a networked interface to other object storage. The client performs parallel I/O operations across multiple OSTs. Each OST is backed by unique storage (a LUN on the RAID array.) Lustre clients communicate with the OSTs.

You configure the characteristics of the OSTs as part of the Lustre setup.

• Object storage server (OSS)

An object storage server (OSS) is a node that hosts the OSTs. Each OSS node, referenced by node ID (NID), has one or more fibre channel connections to a RAID couplet. Where the OST is a logical device, the OSS is the physical node.

On Cray XT systems, although you can run many OSTs per OSS, Cray recommends that you run a maximum of four, and preferably two, OSTs per OSS.

• Metadata server

The metadata server (MDS) owns and manages information about the files in the Lustre file system. It handles namespace operations such as file creation, but it does not contain any file data. It stores information about which file is located on which OSTs, how the blocks of files are striped across the OSTs, the date and time the file was modified, and so on. The MDS is consulted whenever a file is opened or closed, and it may be referenced during I/O to get the block layout on the OSTs. Because file namespace operations are done by the MDS, they do not impact operations that manipulate file data.

The metadata server transforms client requests into journaled, batched metadata updates on persistent storage. The MDS can batch large numbers of requests from a single client or it can batch large numbers of requests generated by different clients, such as when many clients are updating a single object.

You configure the characteristics of the MDS, such as which file is stored on which OST, as part of the Lustre setup.
Each pair of subsystems acts according to protocol:

- **MDS-Client**: The MDS interacts with the client for metadata handling such as the acquisition and updates of inodes, directory information, and security handling.
- **OST-Client**: The object storage target interacts with the client for file data I/O, including the allocation of blocks, striping, and security enforcement.
- **MDS-OST**: The MDS and OST interact to preallocate resources and perform recovery.

Lustre configuration information is maintained in the SDB. For details, see the Cray XT System Management manual and the Cray XT System Software Installation and Configuration Guide.

### 4.4.4 Configuration and Source Files

The administrator uses a boot node 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(8)` utility runs on boot nodes and enables 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 through a Revision Control System (RCS) utility.

### 4.4.5 System Log

After 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.
4.4.6 Service Database

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

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

- Node attributes used by the application launch utilities to schedule jobs. Node attributes include the number of cores present on a processor, the processor clock speed, the amount of memory available to the processor, the architecture type of the node processor, and the type of OS kernel running on the node.

- System configuration tables that list and describe the configuration files.
Cray X2 application development software is the set of products that programmers use to build and run applications on Cray X2 compute nodes. The Cray X2 blade architecture is optimized for applications that require powerful vector processors, shared memory, and high memory bandwidth. This chapter provides an overview of the Programming Environment and related products and processes used to develop applications.

For instructions about setting up a user environment, see Section 4.2, page 28 and the Getting Started on Cray X2 Applications manual.

5.1 UNICOS/lc Operating System

UNICOS/lc consists of service-node and compute-node components.

Cray X2 compute nodes run the Cray X2 kernel and a minimum of other software. The kernel is based on SMP Linux. In addition to the kernel, each node runs a client for the Lustre file system and daemons used to communicate with ALPS. The primary function of the kernel is to initiate programs, manage node memory and terminate programs. Each compute-node kernel services application system requests.

5.2 Compiling Programs

The Cray X2 Programming Environment includes Cray C, C++, and Fortran compilers.

The compilers translate C, C++, and Fortran source programs into Cray X2 object files. Fortran's C interoperability features enable developers to call C or C++ functions directly from Fortran and create Fortran subprograms that can be called directly from C or C++. In addition, global data objects can be shared between Fortran modules and C or C++ functions.

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.
5.2.1 Cray Compiler Commands

The following Cray compiler commands are available:

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray C</td>
<td>cc</td>
</tr>
<tr>
<td>Cray C++</td>
<td>CC</td>
</tr>
<tr>
<td>Cray Fortran</td>
<td>ftn</td>
</tr>
</tbody>
</table>

Table 1. Source and Object Files

For a detailed description of compiler driver command options, see the cc(1), CC(1), and ftn(1) man pages; the Cray C and C++ Reference Manual; and the Cray Fortran Reference Manual.

5.2.2 Cray Fortran Programming Environment

Except as noted in the Cray Fortran Reference Manual, the Cray Fortran Compiler supports the Fortran language as specified by the Fortran 2003 standard, which is the International Organization of Standards ISO/IEC 1539-1:2004. Selected features from the proposed Fortran 2008 standard are also supported.

The Cray Fortran Compiler supports 8-, 16-, 32-, and 64-bit integer data types; 8-, 16-, 32-, and 64-bit logical data types; 32-, 64-, and 128-bit real data types; 64-, 128-, and 256-bit complex data types; and the character data type.

In addition, the Cray X2 development environment extends standard Fortran via compiler directives, calls to Cray library routines and intrinsic functions, and parallel programming model constructs. The supported parallel programming models are the Message Passing Interface (MPI), shared memory access (SHMEM) interface, Fortran language extensions for co-array Fortran (CAF), and OpenMP.

5.2.3 Cray C and C++ Programming Environment

The C++ compiler supports the C++ language in accordance with the International Organization of Standards ISO/IEC 1998, with some exceptions. (For a description of the exceptions, see the Cray C and C++ Reference Manual.) The C compiler supports the C language per the standard ISO/IEC 9899:1999 (C99).
The C and C++ compilers and libraries support the LP64 data type model, which includes the 16-bit short, the 32-bit int, the 64-bit long, the 32-bit float, the 64-bit double, and 8-bit char data types.

In addition, the Cray X2 development environment extends standard C and C++ via pragmas, calls to Cray library routines and intrinsic functions, parallel programming model constructs, and a subset of the GNU Compiler Collection (GCC) C and C++ language extensions. The supported parallel programming models are the Message Passing Interface (MPI), shared memory access (SHMEM) interface, Unified Parallel C (UPC), and OpenMP.

5.3 Using Library Functions

Developers can use C, C++, and Fortran library functions and the following libraries.

- GNU C Language Runtime Library, glibc, functions. Some glibc functions are not supported for compute-node applications; see the Getting Started on Cray X2 Applications manual for details.
- Fortran library procedures. For details, see the man pages for the Fortran library procedures, Fortran intrinsic procedures, and Fortran application programmers I/O.
- C and C++ library routines. For details, see the man pages for the C and C++ library functions and the C and C++ intrinsic procedures.
- MPICH2, an implementation of 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. For details, see the intro_mpi(1) man page.
- Shared-memory access (SHMEM) library routines. For details, see the intro_shmem(1) man page.
- Unified Parallel C (UPC) routines.
- LibSci, the Cray X2 optimized scientific library routines. LibSci provides support for 32- and 64-bit data types. The default 32-bit version of LibSci provides both single- and double-precision routines. The 64-bit version of LibSci contains single-precision routines.
Included in LibSci are:
- Fast Fourier transform (FFT), filter, and convolution routines
- Basic Linear Algebra Subprograms (BLAS)
- Linear Algebra Package (LAPACK)
- Scalable LAPACK (ScaLAPACK) (distributed memory parallel set of LAPACK routines)
- Basic Linear Algebra Communication Subprograms (BLACS)
- Sparse direct solvers
- The Iterative Refinement Toolkit includes support for serial and parallel mixed-precision iterative refinement. The toolkit can provide solutions to linear systems with similar error qualities to a full-precision solver, but with a potential speed up of up to two times faster. For further information, see the intro_irt(1) man page.

For additional information about LibSci, see the intro_libsci(3s) man page.

5.4 Linking Applications

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 requires and creates the executable file (named a.out by default).

5.5 Running Applications

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

- Jobs launched directly using the aprun command
- PBS Pro jobs launched using the qsub and aprun commands
5.5.1 Launching Jobs with aprun

To launch a job directly using the aprun command, the user must specify all executables that are part of the job and the number of cores (also referred to as processing elements) on which they will run. Each Cray X2 processor has one core. The user can request a number of compute nodes or, using data from the cnselect, xtshowcabs, or xtshowmesh command, a list of specific compute nodes.

The aprun client sends the aprun command information to apsys client, which forwards it to the apsched daemon. The apsys agent gets the compute node placement list and relays the list to the aprun client. The aprun client sends the placement list and the executable binary data to the apinit daemon running on the first node assigned to the application (PE 0). On each node allocated to the application, an application shepherd is created to manage the process of the application on that node. The application shepherds propagate the node placement list and executable binary data to all processing elements allocated to the job. The apinit daemon on PE 0 recreates the user's login node environment on each processing element and passes control to the application.

While the application is running, the application shepherds monitor the application processes. If the aprun client or an application instance catches a signal, the signal is propagated to all processing elements. Applications rely on aprun to manage the stdin, stdout, and stderr streams and handle signal management.

5.5.2 Launching Jobs with PBS Pro

The Cray XT5 system also uses PBS Pro to launch jobs. PBS Pro is a networked subsystem for submitting, monitoring, and controlling a workload of batch jobs.

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 creates a script containing the aprun command(s) to run the application, then invokes 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 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 ALPS 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 aprun utility uses ALPS to get the list of compute nodes allocated to the job, fetch the executable, and propagate it to all compute nodes selected to run that program.

5.6 Debugging Applications

Users can debug programs by analyzing compile time and runtime messages and making the appropriate corrections. The explain ftn-msgnumber and explain CC-msgnumber commands provide explanations of error messages.

The Cray XT5h system supports the TotalView debugger from TotalView Technologies for single-process and multiprocess debugging. TotalView provides source-level debugging of applications and is compatible with the Cray C, C++, and Fortran compilers.

TotalView can debug applications running on 1 to 1024 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 using TotalView, 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. TotalView directs aprun to load but not start the application. The aprun utility then loads the application onto the compute nodes, after which TotalView can perform initial setup before instructing aprun to start the application.

For more information about TotalView, see the totalview(1) and totalviewcli(1) man pages and the Using TotalView manual.

5.7 Monitoring and Managing Applications

ALPS provides commands for monitoring and managing applications. The apstat command reports the status of applications, including the number of processing elements, number of threads, a map of the application’s address space, and a map showing the placement of team members. The apkill command sends a kill signal to an application team. For more information about these commands, see the apstat(1), and apkill(1) man pages.
5.8 Measuring Performance

The Cray XT5\textsubscript{n} system provides optional tools for the collection, display, and analysis of performance data. All of these tools require a separate license. For further information, see the Using Cray Performance Analysis Tools manual.

5.8.1 Performance API

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 Cray X2 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.

5.8.2 CrayPat

CrayPat is a performance analysis tool. It is an optional product available from Cray Inc. The developer can use CrayPat to instrument programs for performance data collection, execute the instrumented program under controlled conditions, and then analyze the performance analysis data that was collected during program execution. Refer to the intro_craypat(1) man page for an overview of CrayPat.

While CrayPat requires relinking, no recompilation is needed in order to produce the instrumented program.

After instrumenting a program, the developer sets environment variables to control run-time data collection, runs the instrumented program, and then uses the pat_report(1) command to either generate performance analysis reports or export the data for use in other applications.

5.8.3 Cray Apprentice2

Cray Apprentice2 is an interactive X Window System tool for displaying data captured by CrayPat during program execution.
Cray Apprentice2 identifies the following conditions:
- Load imbalance
- Excessive serialization
- Excessive communication
- Network contention
- Poor use of the memory hierarchy
- Poor functional unit use

Cray Apprentice2 has the following capabilities:
- It provides information about a program by examining data files that were created during program execution.
- It displays two types of performance data: call-stack sampling and Cray MPICH2 tracing, contingent on the data that was captured by CrayPat during execution.
- It reports time statistics for all processing elements and 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.

### 5.9 Optimizing Applications

The Cray C, C++, and Fortran compilers provide compiler command options, directives, and pragmas that the developer can use to optimize code. For further information, see the *Cray C and C++ Reference Manual* and the *Cray Fortran Reference Manual*. 
blade
1) A field-replaceable physical entity. A Cray XT service blade consists of AMD Opteron sockets, memory, Cray SeaStar chips, PCI-X cards, and a blade control processor. A Cray XT compute blade consists of AMD Opteron sockets, memory, Cray SeaStar chips, and a blade control processor. A Cray X2 compute blade consists of eight Cray X2 chips (CPU and network access links), two voltage regulator modules (VRM) per CPU, 32 memory daughter cards, a blade controller for supervision, and a back panel connector. 2) From a system management perspective, a logical grouping of nodes and blade control processor that monitors the nodes on that blade.

cabinet control processor
A microprocessor in the cabinet that communicates with the HSS via the administrative network to monitor and control the devices in a system cabinet. See also Hardware Supervisory System (HSS).

chassis (Cray XT systems)
The hardware component of a Cray XT cabinet that houses blades. Each cabinet contains three vertically stacked chassis, and each chassis contains eight vertically mounted blades.

chassis (Cray X2 component)
A blade enclosure together with power distribution and four R1 router cards. A chassis houses from one to eight blades (either compute or bridge blades). Cabinets contain chassis or router modules: a compute cabinet contains two chassis; a router cabinet has three chassis and contains a router backplane housing up to 16 router modules.

CNL
CNL is a Cray XT and Cray X2 compute node operating system. CNL provides a set of supported system calls. CNL provides many of the operating system functions available through the service nodes, although some functionality has been removed to improve performance and reduce memory usage by the system.

compute blade
See blade.
**compute node**
A node that runs application programs. A compute node performs only computation; system services cannot run on compute nodes. Compute nodes run a specified kernel to support either scalar or vector applications. See also node; service node.

**Cray Fortran Compiler**
The compiler that translates Fortran source code into executable object files. The Cray Fortran Compiler supports the Fortran language through the Fortran 2003 Standard, ISO/IEC 1539-1:2004, with some exceptions and extensions, as well as selected features of the proposed Fortran 2008 Standard, as described in the Cray Fortran user documentation.

**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.

**Hardware Supervisory System (HSS)**
Hardware and software that monitors the hardware components of the system and proactively manages the health of the system. It communicates with nodes and with the management processors over the private Ethernet network. See also system interconnection network.

**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.
node
For UNICOS/lc 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 server (OSS)
A node that hosts the OSTs of the Lustre file system.

object storage target (OST)
The component of the Lustre file system that handles file activities.

processing element
The smallest physical compute group.

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

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

service node
A node that performs support functions for applications and system services. The service node operating system is based on SUSE Linux and performs specialized functions. There are six types of predefined service nodes: login, I/O, 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 exist for a node or for a class of nodes.
StarGate
A high-speed interconnect that serves as a bridge between the Cray XT system’s SeaStar interconnect and the YARC interconnect on Cray X2 compute nodes.

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 HSS and system administration. See also Hardware Supervisory System (HSS).

TLB
A table (Translation Lookaside Buffer) in the processor that contains cross-references between the virtual and real addresses of recently referenced pages of memory.

TotalView
An optional symbolic source-level debugger designed for debugging the multiple processes of parallel Fortran, C, or C++ programs.

UNICOS/lc
The operating system for Cray XT systems.
Index

A
Accessing the system, 28
Adaptive Supercomputing
definition, 1
Administration, 29
Application development, 35
Application launch, 38
Applications
debugging, 40
linking, 38
measuring performance, 41
monitoring and managing, 40
optimizing, 42
running, 38
aprun utility, 38–39
Authentication, 28

B
Batch jobs, 38–39
Blades
bridge, 17
compute, 16
Cray X2, 16
Boot nodes, 28, 32

C
C and C++
compilers, 35
interlanguage communications, 35
ISO/IEC 1998, 36
ISO/IEC 9899:1999 (C99), 36
programming environment, 36
C command, 35
C++ command, 35
Cabinet, Cray X2, 18
Chassis, Cray X2, 18
Chip
CPU router, 8
protocol bridge chip, 7
StarGate, 7
Weaver, 10–11
YARC router, 12
Compiler commands, 35
Compilers
C and C++, 35
Fortran, 35
Compute nodes, Cray X2, 8
Compute Processor Allocator (CPA), 33
Configuration and source files, 32
Cray Apprentice2 (CrayPat data display tool), 41
Cray X2
application development, 35
common software components with Cray XT
systems, 27
compute node kernel, 35
calculate nodes, 2
fat-tree topology, 1
overview, 1
YARC interconnect, 1
Cray X2 hardware components, 8
calculate nodes, 8
Hardware Supervisory System (HSS), 14
local memory, 10
memory daughter cards (MDC), 8
network router, 12
StarGate chip, 13
Weaver chip, 8
YARC router chip, 12
Cray X2 physical components
blades, 16
cabinet, 18
chassis, 18
Cray XT systems
common software components with Cray X2, 27
CrayPat (performance analysis tool), 41

S–2472–10
D
Database nodes, 33
Debugging, 40
Documentation
  application developers, 4
  system administrators, 5

F
Fat-tree network topology, 1, 13
Fault tolerance, 18
File system
  Lustre, 21
  shared-root, 30
File systems
  Lustre, 29
Fortran
  2003 standard, 36
  2008 standard, 36
  commands, 35
  compilers, 35
  interlanguage communications, 35
  programming environment, 36

H
Hardware components
  Cray X2, 7
  Cray XT, 7
Hardware performance counters, 41
Hardware Supervisory Subsystem (HSS), 22
Hardware Supervisory System (HSS), 14, 22

I
I/O nodes, 28
Interactive jobs, 38–39
Interlanguage communications, 35

J
Jobs
  batch, 38
  interactive, 38

K
Kernel, 21

L
Libraries, 37
  math, 37
  scientific, 37
Linker, 38
Linux, 28
Local memory, Cray X2, 10
Logging in, 28
Login load leveling, 28
Login nodes, 28
Lustre file system, 21, 29
  administration, 30
  client interaction with MDS and OST, 30
  clients, 21
  layout of file system, 30
  Meta Data Servers (MDSs), 21
  metadata server (MDS), 30
  object storage target (OST), 30
  Object Storage Targets (OSTs), 21
  shared-root file system, 30

M
Mazama, 14, 22–23
  administrative states, 25
  command and control, 24
  configuration, 25
  features, 23
  image management, 24
  infrastructure, 26
  monitoring, 24
  network management, 25
  partition management, 24
  reliability, availability, and serviceability, 25
  MDS, 30
Memory daughter cards (MDC), 8
Metadata server (MDS), 30
Modules
  Cray X2, 16
  router, 18
N
Network
- Cray X2 and Cray XT interconnect, 13
  - fat-tree topology, 13
  - injection port, 13
  - router, 12
  - router chip, 12
- Network bridge
  - StarGate, 13
- Network nodes, 28
- Nodes
  - Cray X2 compute, 2
  - Cray XT system service-I/O (SIO), 2

O
Object storage server (OSS), 30
Object storage target (OST), 30
Operating system, 27, 35

P
PBS Pro, 38
Performance
- analysis tool, 41
- API (PAPI), 41
- measurement, 41
Programming environment
- C and C++, 36
  - debugging applications, 40
  - Fortran, 36
  - libraries, 37
- linking applications, 38
- measuring performance, 41
- monitoring and managing applications, 40
- optimizing performance, 42
- running applications, 38
Publications
- application developers, 4
- system administrators, 5

R
Reliability and system management, 18
Running applications, 38

S
SeaStar interconnect, 2
Secure shell, 28
Service Database (SDB), 28, 33
Service node, 28
  - boot, 32
  - database, 33
Service node classes
  - boot, 28
  - I/O, 28
  - login, 28
  - Login, 28
  - network, 28
  - Service Database SDB, 28
  - syslog, 28
Shared-root file system, 30
StarGate, 2, 13
SUSE LINUX, 28
Syslog, 32
Syslog nodes, 28
System
- administration, 29
  - common software components, 27
  - configuration and source files, 32
  - file system, 29
  - logging, 32
  - operating (UNICOS/lc), 27
  - shared-root file system, 30
- System Management Workstation (SMW), 30
  - user environment, 28
System control, 22
  - Hardware Supervisory Subsystem (HSS), 22
  - Mazama system administration software, 23
System Management Workstation (SMW), 14–15, 30
System monitoring, 22
  - Hardware Supervisory Subsystem (HSS), 22
  - Mazama system administration software, 23
## Cray XT5™ System Overview

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T</strong></td>
<td><strong>SUSE LINUX, 28</strong>&lt;br&gt;TotalView, 40&lt;br&gt;User environment, 28</td>
</tr>
</tbody>
</table>
| **U** | **Y**<br>YARC interconnect, 1–2<br>YARC router chip, 12
| UNICOS/lc operating system | YARC interconnect, 1–2 |
| Cray X2 compute node kernel, 35 | YARC router chip, 12 |