Cray XMT™ System Overview

S–2466–201

© 2007–2009, 2011 Cray Inc. All Rights Reserved. This document or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Cray Inc.

Copyright (c) 2008, 2011 Cray Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name Cray Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Your use of this Cray XMT release constitutes your acceptance of the License terms and conditions.

U.S. GOVERNMENT RESTRICTED RIGHTS NOTICE

The Computer Software is delivered as "Commercial Computer Software" as defined in DFARS 48 CFR 252.227-7014.

All Computer Software and Computer Software Documentation acquired by or for the U.S. Government is provided with Restricted Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7014, as applicable.

Technical Data acquired by or for the U.S. Government, if any, is provided with Limited Rights. Use, duplication or disclosure by the U.S. Government is subject to the restrictions described in FAR 48 CFR 52.227-14 or DFARS 48 CFR 252.227-7013, as applicable.

Cray, LibSci, and PathScale are federally registered trademarks and Active Manager, Cray Apprentice2, Cray Apprentice2 Desktop, Cray C++ Compiling System, Cray CX, Cray CX1, Cray CX1-iWS, Cray CX1-LC, Cray CX1000, Cray CX1000-C, Cray CX1000-G, Cray CX1000-S, Cray CX1000-SC, Cray CX1000-SM, Cray CX1000-HN, Cray Fortran Compiler, Cray Linux Environment, Cray SHMEM, Cray X1, Cray X1E, Cray X2, Cray XD1, Cray XE, Cray XEm, Cray XE5, Cray XE5m, Cray XE6, Cray XE6m, Cray XK6, Cray XMT, Cray XR1, Cray XT, Cray XTm, Cray XT3, Cray XT4, Cray XT5, Cray $XT5_h$, Cray XT5m, Cray XT6, Cray XT6m, CrayDoc, CrayPort, CRInform, ECOphlex, Gemini, Libsci, NodeKARE, RapidArray, SeaStar, SeaStar2, SeaStar2+, The Way to Better Science, Threadstorm, and UNICOS/lc are trademarks of Cray Inc.

AMD, AMD Opteron, and Opteron are trademarks of Advanced Micro Devices, Inc. Linux is a trademark of Linus Torvalds. NFS and Lustre are trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Platform is a trademark of Platform Computing Corporation. RSA is a trademark of RSA Security Inc. SUSE is a trademark of Novell, Inc. UNIX, the “X device,” X Window System, and X/Open are trademarks of The Open Group in the United States and other countries. All other trademarks are the property of their respective owners.

RECORD OF REVISION

S–2466–201 Published September 2011 Supports software release 2.0 for Cray XMT hardware with Threadstorm 3 or Threadstorm 4 processors.

S–2466–20 Published May 2011 Supports software release 2.0 GA running on Cray XMT Series compute nodes and Cray XT service nodes. This release uses CLE version 3.1UP02 System Management Workstation (SMW) version 5.1UP03.

1.4 Published December 2009 Supports release 1.4 running on Cray XMT compute nodes and CLE 2.2.UP01 on Cray XT service nodes. This release uses the System Management Workstation (SMW) version 4.0.UP02.

1.3 Published March 2009 Supports release 1.3 running on Cray XMT compute nodes and on Cray XT 2.1.5HD service nodes. This release uses the System Management Workstation (SMW) version 3.1.09.

1.2 Published August 2008 Supports release 1.2 running on Cray XMT compute nodes and on Cray XT 2.0.49 service nodes. This release uses the System Management Workstation (SMW) version 3.1.04.

1.1 Published March 2008 Supports limited availability (LA) release 1.1.01 running on Cray XMT compute nodes and on Cray XT 2.0 service nodes.

1.0 Published July 2007 Supports the 1.0 limited availability (LA) release of the Cray XMT.

Contents

Page

Introduction [1] **9**

1.1 Cray XMT Features 9

1.2 Related Publications 10

1.2.1 Publications for Application Developers 10

1.2.2 Publications for System Administrators 11

Hardware Overview [2] **13**

2.1 Basic Hardware Components 13

2.1.1 Threadstorm 4.0 Processor 13

2.1.2 DIMM Memory 14

2.1.3 Content-Addressable Memory (CAM) 15

2.1.4 Cray SeaStar2 Chip 15

2.1.5 System Interconnection Network 16

2.1.6 RAID Disk Storage Subsystems 16

2.2 Nodes 16

2.2.1 Compute Nodes 17

2.2.2 Service Nodes 18

2.3 Blades, Chassis, and Cabinets 18

2.3.1 Blades 18

2.3.2 Chassis and Cabinets 19

Software Overview [3] **21**

3.1 Cray SeaStar High-speed Network Communication Interfaces 22

3.2 Cray Linux Environment (CLE) Operating System 24

3.2.1 SUSE LINUX Operating System 24

3.2.2 MTK Operating System 24

3.3 File Systems 25

3.3.1 Lustre File System 25

3.3.2 Random Access Memory File System 26

3.3.3 Network File System 27

Page

3.4 User Environment 27
3.5 System Administration 27
3.5.1 System Management Workstation 28
3.5.2 Shared-root File System 28
3.5.3 Configuration and Source Files 28
3.5.4 System Monitoring 29
3.5.5 System Log 29

Application Development [4] 31

4.1 User Runtime Library 31
4.2 Lightweight User Communication Library (LUC) API 31
4.3 Compiling Programs 32
4.3.1 Compiler Commands 32
4.4 Running Applications 32
4.5 Debugging Applications 33
4.6 Monitoring Applications 33
4.7 Measuring Performance 33
4.7.1 Cray Apprentice2 33
4.7.2 Canal 34
4.7.3 Tview 34
4.7.4 Bprof 34
4.7.5 pproc 35
4.7.6 ap2view 35
4.7.7 Tprof 35

Cray Hardware Supervisory System (HSS) [5] 37

5.1 HSS Hardware 37
5.1.1 HSS Network 38
5.1.2 System Management Workstation 38
5.1.3 Hardware Controllers 39
5.2 HSS Software 39
5.2.1 Software Monitors 39
5.2.2 HSS Administrator Interfaces 39
5.3 HSS Actions 40
5.3.1 System Startup and Shutdown 40
5.3.2 Event Probing 40
5.3.3 Event Logging 41
5.3.4 Event Handling 41

Page

Figures

Figure 1. Threadstorm Processor Architecture 14
Figure 2. Cray SeaStar2 Chip 16
Figure 3. Cray XMT Hardware System Architecture 17
Figure 4. Chassis and Cabinet (front view) 20
Figure 5. Software Stack for Service and Compute Nodes 21
Figure 6. LUC Software Stack 23
Figure 7. Lustre Architecture on Cray XMT 26
Figure 8. HSS Components 38

Introduction [1]

This document provides an overview of the second generation of Cray XMT Series systems. The software portion of this document applies also to first generation Cray XMT systems running version 2.0 of the Cray XMT system software. For an overview of the first generation Cray XMT hardware, please refer to an earlier version of this document.

The intended audience is application developers and system administrators. Familiarity with the concepts of high-performance computing and the architecture of parallel processing systems is assumed.

1.1 Cray XMT Features

The Cray XMT Series of supercomputers are scalable, massively multithreaded platforms with a globally shared memory architecture. The second generation Cray XMT system is based on the Cray XT5 infrastructure and uses the Cray massively parallel processing (MPP) system design. The difference is that the Cray XMT compute blades use Threadstorm processors, which are designed to perform multithreaded operations instead of the AMD Opteron-based processors used in the Cray XT5.

The second generation Cray XMT platform has the following features:

- Performs large-scale data analysis.
- Uses Cray Threadstorm 4.0 processors. Each processor is directly connected to a dedicated Cray SeaStar2 interconnect chip, resulting in a high-bandwidth, low-latency network characteristic.
- Scales from 16 to 512 processors providing over half a million threads, using 16 terabytes of system memory.
  - Uses nodes as the most basic scalable unit. There are two types of nodes. Service nodes provide support functions, such as managing the user's environment, handling I/O, and booting the system. Compute nodes run user applications.
  - Uses a global memory model. Applications have access to memory on any compute processor on the machine.

  - Uses the system interconnection network to connect compute and service nodes to maintain high communication rates as the number of nodes increases. Contains Seastar2 chips connected in a full 3-D torus network.

- Contains separately dedicated compute, service, and I/O nodes.
  - Service nodes have AMD Opteron processors and can be configured for I/O, login, network, or system functions.
  - Compute nodes have Threadstorm processors.
- Runs the Cray Linux Environment (CLE) operating system which distributes a multithreaded kernel (MTK) to the compute blades and standard Linux on the service and I/O blades. This enables the compute nodes to focus on the application without being hampered by system administrative functions.
- Includes a development environment that provides compilers, libraries, parallel programming models, debuggers, and performance measurement tools.

1.2 Related Publications

The Cray XMT system runs with a combination of proprietary, third-party, and open-source products, as documented in the following publications.

1.2.1 Publications for Application Developers

For information about the Cray XMT Programming Environment see the following Cray guides.

- *Cray XMT Programming Model*
- *Cray XMT Programming Environment User's Guide*
- *Cray XMT Performance Tools User's Guide*
- *Cray XMT Debugger Reference Guide*
- Cray XMT man pages

1.2.2 Publications for System Administrators

The following publications are available for system administrators.

- *Installing and Configuring Cray XMT System Software*
- *Installing and Configuring the Cray XMT System Management Workstation*
- *Cray XMT System Management*
- *Managing System Software for Cray XE Systems*
- *Cray XT System Overview*
- *Cray XT System Software Release Overview*
- Cray XMT man pages
- System Management Workstation (SMW) man pages for Cray XT and Cray XMT

Hardware Overview [2]

2.1 Basic Hardware Components

The second generation Cray XMT platform includes the following hardware components:

- Threadstorm 4.0 processors on compute nodes and Opteron processors on service nodes
- Dual inline memory modules (DIMMs)
- Cray SeaStar2 chips
- System interconnection network
- RAID disk storage subsystems

2.1.1 Threadstorm 4.0 Processor

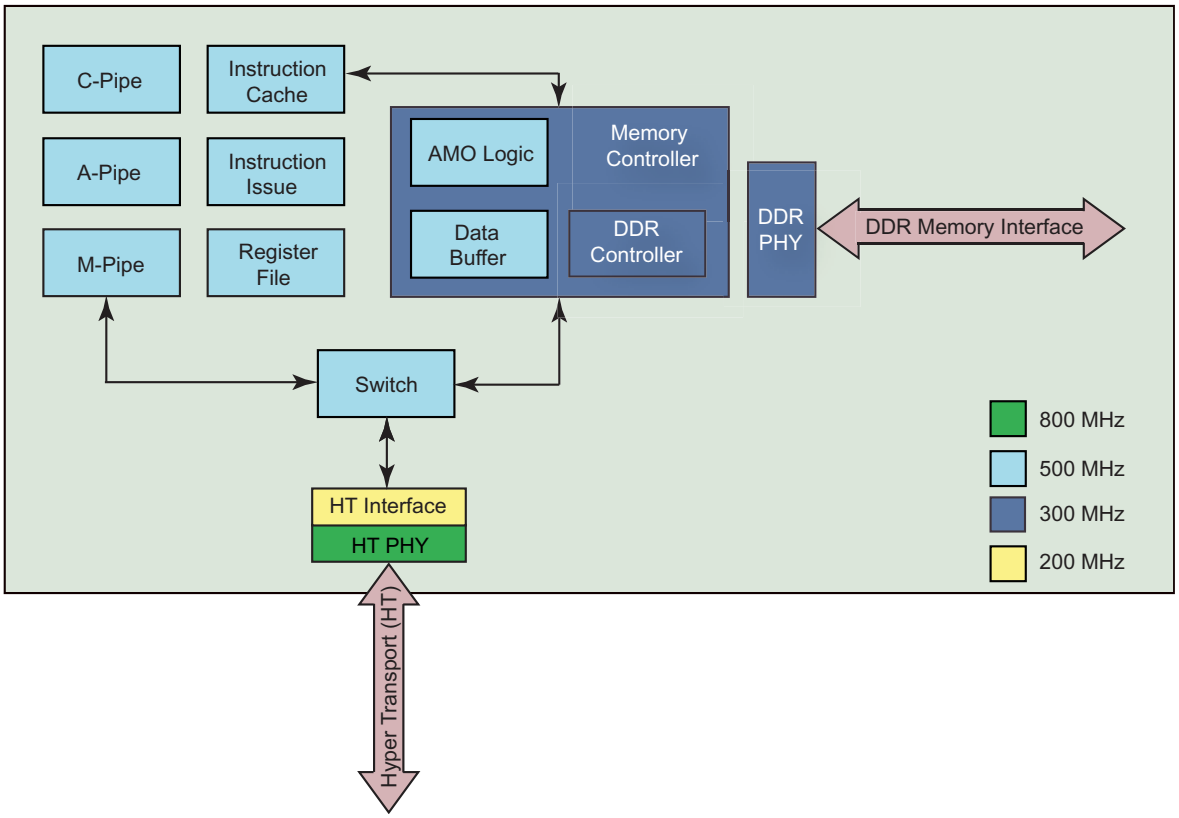
The second generation Cray XMT platform uses Threadstorm 4.0 processors on compute nodes. Threadstorm processors feature:

- Multithreaded processors that support parallel operations.
- Ability to perform remote memory access.
- 128 streams on each processor with 31 general purpose 64-bit registers, 8 target registers, and a status word that includes the program counter. A stream is the hardware used to execute a single thread.
- 16 protection domains on each processor which provide address spaces. Each running stream belongs to one protection domain.
- Three functional units to support operations: the M unit which issues a memory operation, the A unit which executes an arithmetic operation, and the C unit which executes a control or simple arithmetic operation. The Threadstorm ISA is a large instruction word (LIW) where each instruction can specify up to three operations, one for each functional unit.

The Threadstorm architecture includes the following elements:

- Instruction execution logic.
- Double data rate (DDR2) memory controller and data cache.
- Content-addressable memory (CAM).
- HyperTransport (HT) logic and physical interface.
- A switch that connects these three components.

Figure 1. Threadstorm Processor Architecture



2.1.2 DIMM Memory

The Cray XMT supports double data rate dual inline memory modules (DIMMs). The second generation Cray XMT include 8 4-GB or 8-GB DDR2 DIMMs for a maximum physical memory of 64 GB per node. The minimum amount of memory for service nodes is 8 GB.

The Cray XMT use Error-Correcting Code (ECC) memory protection technology.

2.1.3 Content-Addressable Memory (CAM)

Content-addressable memory takes a user-supplied data word and searches the entire memory to see if that data word is stored anywhere in it. If the data word is found, CAM returns a list of one or more storage addresses where the word was found. Because CAM is designed to search its entire memory in a single operation, it is much faster than RAM in virtually all search applications.

The Cray XMT uses CAM in a variety of operations. For example, the data and instruction caches use CAMs to hold the tags, allowing for faster search and retrieval.

In the second-generation Cray XMT, CAM is also used to implement three methods for hotspot avoidance—load combining of up to 32 load operations, fetch-and-add combining, and reference synchronization.

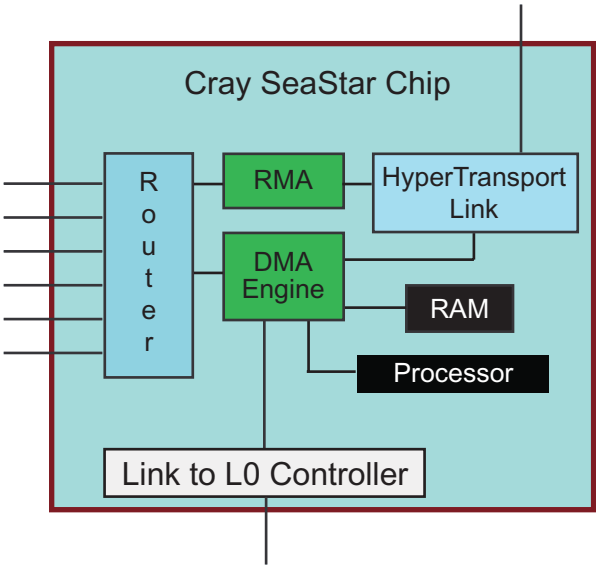
2.1.4 Cray SeaStar2 Chip

The Cray XMT systems use Cray SeaStar2 chips. The Cray SeaStar2 application-specific integrated circuit (ASIC) chip is the system's message processor.

SeaStar2 offloads communications functions from the Threadstorm processor. A SeaStar2 chip contains:

- A HyperTransport Link, which connects SeaStar2 to the Threadstorm processor.
- A 3-D router that connects the chip to the system interconnection network using six high-speed serial links.
- A Remote Memory Access (RMA) block that converts Threadstorm remote memory references to network transactions and back.
- Two Direct Memory Access (DMA) engines, one for sending and the other for receiving, that manage the movement of data to and from node memory. The DMA engines are controlled by an embedded PowerPC processor (described in the next bullet).
- An embedded PowerPC processor to support the network interconnect. The processor programs the DMA engines and assists with other network-level processing needs, such as supporting the Portals message-passing layer of the Cray XT.
- A Portals message passing interface, which provides a data path from an application to memory. Portions of the interface are implemented in Cray SeaStar2 firmware, which transfers data directly to and from user memory without operating system intervention.
- A link to a blade control processor (also known as an *L0 controller*). Blade control processors are used for booting, monitoring, and maintenance. For more information, see Hardware Controllers on page 39.

Figure 2. Cray SeaStar2 Chip



2.1.5 System Interconnection Network

The system interconnection network is the communications center of the second generation Cray XMT system. The network consists of Cray SeaStar2 router links and the cables that connect the compute and service nodes. RMA requests and I/O data are transferred over the network.

The network uses a Cray proprietary protocol to provide fast access to globally shared memory, and Fast I/O (FIO) data transfers.

2.1.6 RAID Disk Storage Subsystems

Cray XMT systems use two types of RAID subsystems for data storage. The System RAID stores the boot image and system files. The Data RAID is configured as a Lustre file system that is accessible from the service nodes. The Lustre file system is not directly accessible from the compute nodes.

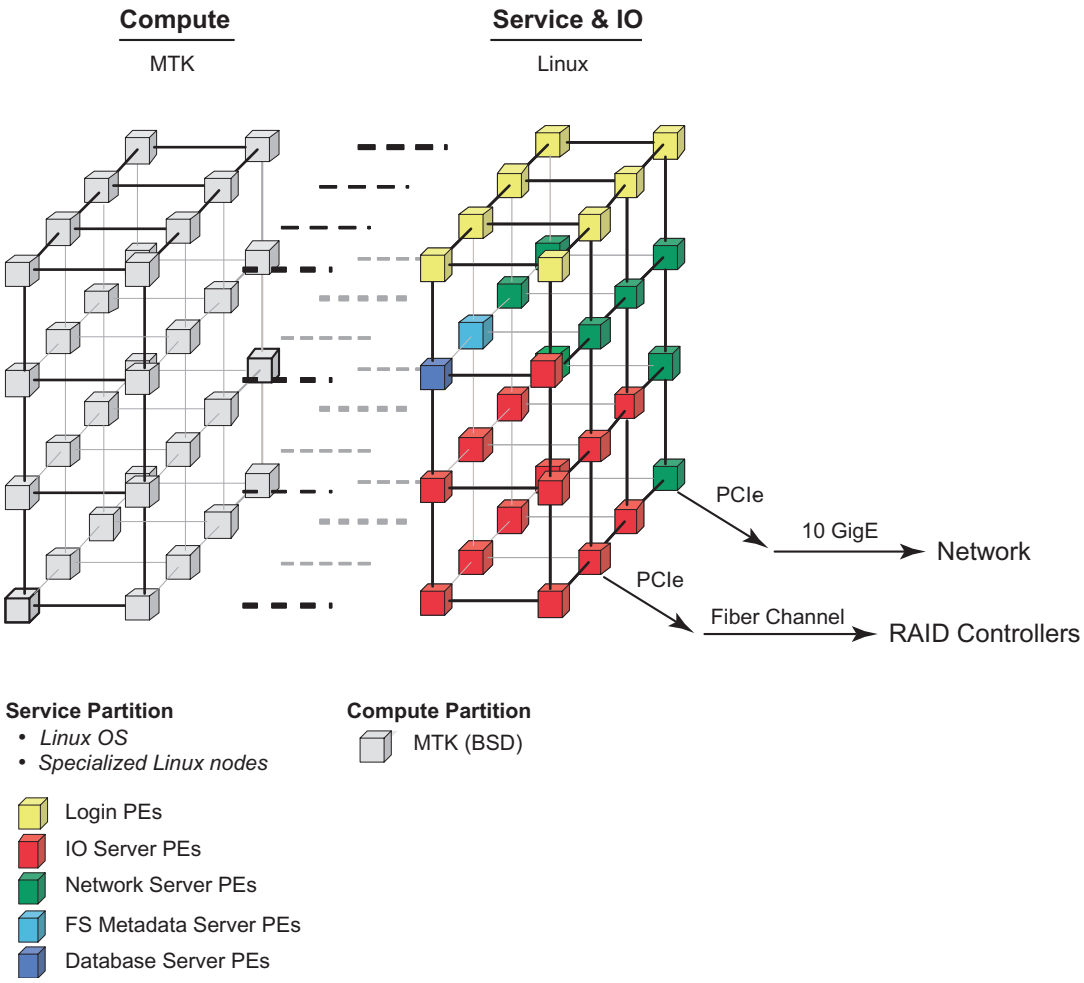
Data on system RAID is globally accessible to the service partition. It is not accessible to the compute partition.

2.2 Nodes

Cray XMT systems use processing components combine to form a node. There are two types of nodes: compute nodes and service nodes. Each node is a logical grouping of a processor, memory, and a data routing resource.

The following diagram shows a conceptual view of the 3-D torus network topology (torus links are not shown) for compute and service nodes. The service nodes connect to the compute nodes through the system interconnection network.

Figure 3. Cray XMT Hardware System Architecture



2.2.1 Compute Nodes

Compute nodes run application programs. Each second generation Cray XMT compute node consists of a Threadstorm 4.0 processor, DIMM memory, and a Cray SeaStar2 chip.

All compute nodes in a logical system use the same processor type.

2.2.2 Service Nodes

Service nodes handle support functions such as user login, I/O, and network management. Each service node contains an Opteron processor, DIMM memory, and a SeaStar2 chip. In addition, each service node may be configured with one or two PCIe Ethernet Network Interface Cards (NIC).

Cray XMT systems include several types of service nodes, defined by the function they perform.

- Login nodes. A login node may have one or two PCIe cards that connect to your network. PCIe cards are supported for both Gigabit Ethernet and 10-Gigabit Ethernet, as well as Fibre Channel. Login nodes that do not have a NIC are accessed by first logging into a login node that has a connection to the network. All login nodes are Lustre clients, and mount the Lustre file system from the disk nodes. You can also use login nodes to run snapshot file system workers. These workers assist in moving data from the Threadstorm memory to the Lustre file system.
- Network service nodes: Each network service node has one or two network interface cards that are connected to the network.
- I/O nodes: Each I/O node uses one or two Fibre Channel cards to connect to RAID storage.
- Boot nodes: Each system requires one boot node. A boot node contains one Fibre Channel card and one Gigabit Ethernet Card. The Fibre Channel card connects to the RAID subsystem, and the PCIe card connects to the SMW (see Chapter 5, Cray Hardware Supervisory System (HSS) on page 37 for further information).
- SDB nodes: Each SDB node contains a Fibre Channel card to connect to the SDB file system. The SDB node manages the state of the Cray XT system.

2.3 Blades, Chassis, and Cabinets

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

2.3.1 Blades

A compute blade consists of four compute nodes, voltage regulator modules, and an L0 controller. Each compute blade within a logical machine is populated with Threadstorm processors of the same type and speed and memory chips of the same speed.

The L0 controller is a Hardware Supervisory System (HSS) component; for more information about HSS hardware, see Chapter 5, Cray Hardware Supervisory System (HSS) on page 37.

A service blade consists of two service nodes, voltage regulator modules, up to two PCIe cards per node, and an L0 controller. A service blade has four SeaStar2 chips to allow for a common board design and to simplify the interconnect configurations. Several different PCIe cards are available to provide Fibre Channel, GigE, and 10 GigE interfaces to external devices.

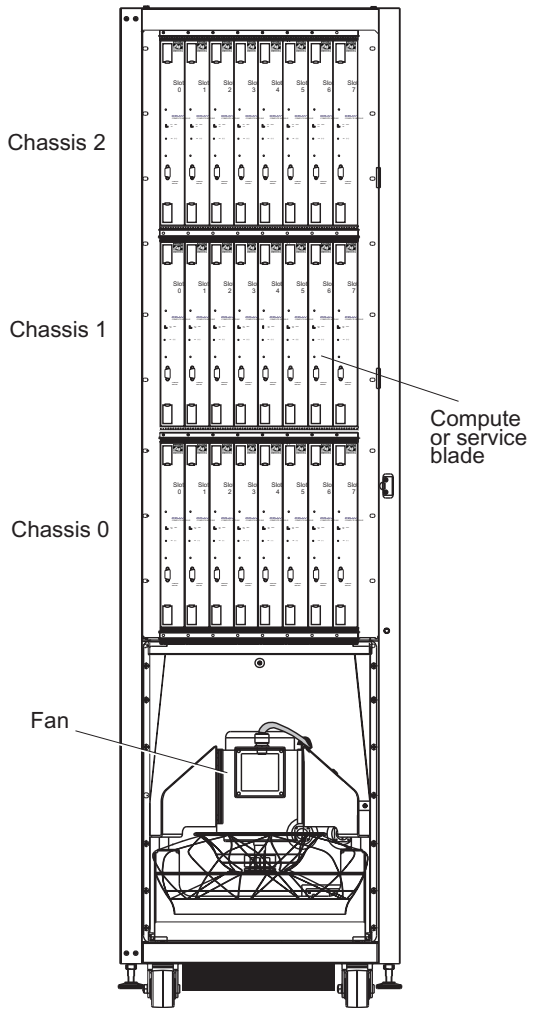
2.3.2 Chassis and Cabinets

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.

Customer-provided three-phase power is supplied to the cabinet Power Distribution Unit (PDU). The PDU routes power to the cabinet's power supplies, which distribute 48 VDC to each of the chassis in the cabinet.

All cabinets have redundant power supplies. The PDU, power supplies, and the cabinet control processor (L1 controller) are located at the rear of the cabinet.

Figure 4. Chassis and Cabinet (front view)



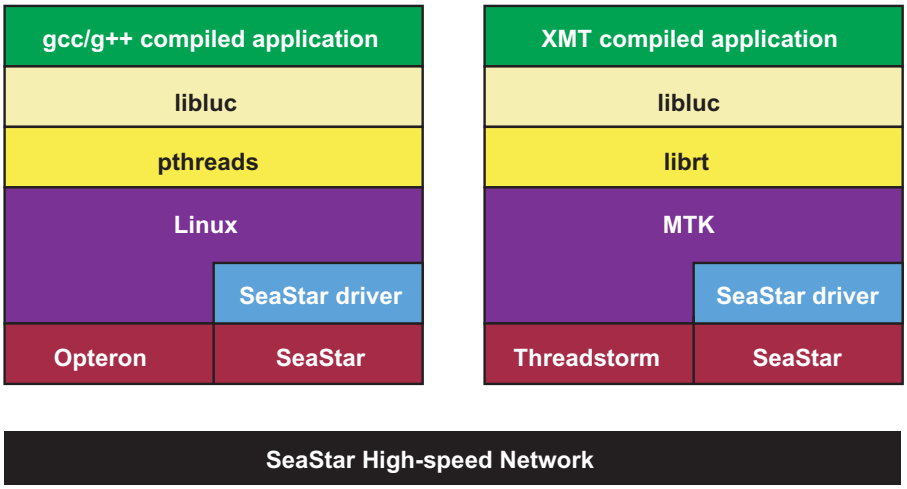
Software Overview [3]

Cray XMT Series 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.

This chapter provides an overview of the Cray XMT operating system, the random access memory file system (RAMFS) file system, the application development environment, and system administration tools.

The software stack for Cray XMT is shown in the following figure. The stack on the left applies to the service nodes, and the stack on the right applies to compute nodes.

Figure 5. Software Stack for Service and Compute Nodes



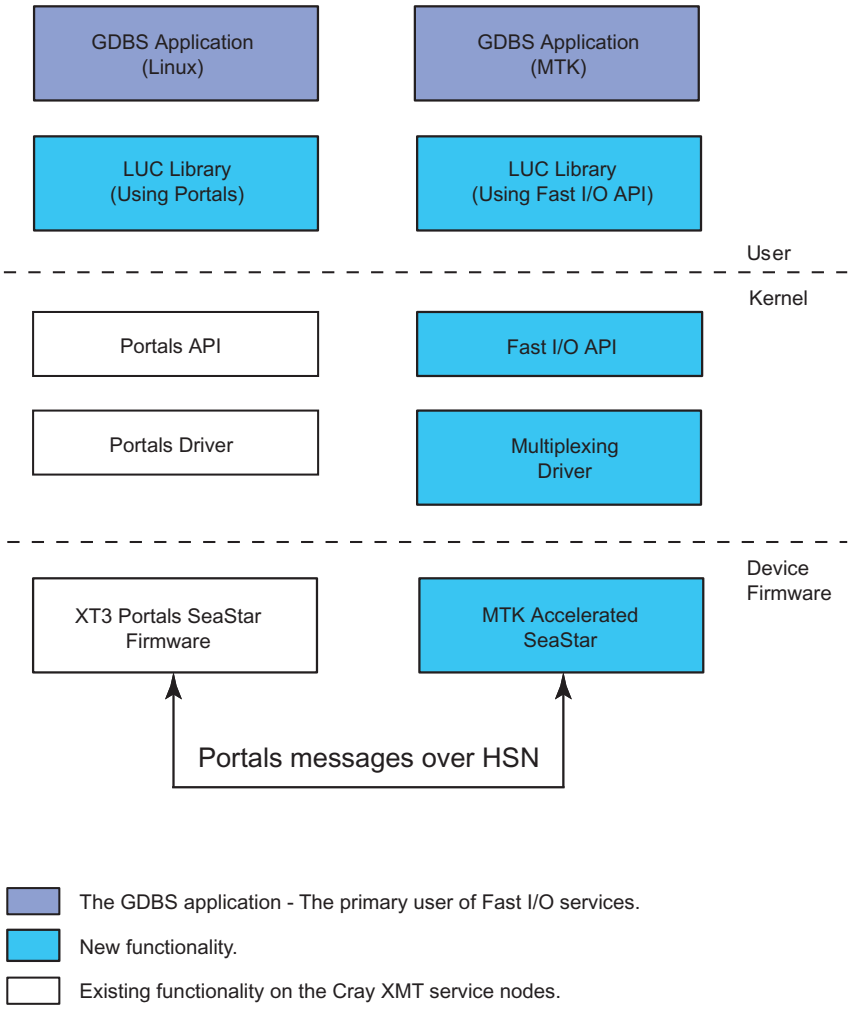
3.1 Cray SeaStar High-speed Network Communication Interfaces

The Cray SeaStar high-speed network supports three communication interfaces:

- The Lightweight User Communication Library (LUC) (user level) for communicating between service and compute nodes
- The Portals API (kernel level) supports communication between the service nodes and is used by Lustre clients and servers for file system data transfers. The compute nodes implement a subset of the Portals API to support Fast I/O (FIO) and LUC communication between the compute nodes and service nodes.
- The FIO API (kernel level) supports LUC on compute nodes. FIO is layered over the MTK Portals SeaStar driver.

Figure 6 describes the LUC software protocol stack. The column on the left represents the software stack on the Linux service nodes. The column on the right represents the software stack on the MTK compute nodes. GDBS is the sample application in this figure. The application is linked with the LUC library. The LUC library presents a uniform application interface for both the service nodes and compute nodes, and abstracts the details of the actual implementation.

Figure 6. LUC Software Stack



On service nodes, the LUC library is implemented using the Linux Portals library and system call interface. The standard Cray XT Portals driver is used with no modifications. The Portals driver interacts with the Portals firmware that executes on the Cray SeaStar chip associated with the service node. The firmware is responsible for sending and receiving Portals messages over the Cray SeaStar chip network. As with the Portals driver, there are no modifications to the Portals firmware.

On compute nodes, the LUC library is implemented using the FIO system call interface. The FIO API is optimized for large data transfers over the Cray SeaStar chip and provides a customized interface for the LUC library. The system call layer is the layer above the FIO multiplexor driver. The multiplexor driver is responsible for multiplexing the operations from multiple FIO streams through a single Cray SeaStar chip. The multiplexor interacts with the firmware executing on the Cray SeaStar chip. The firmware is customized for FIO, and supports a subset of the Portals interface. Limiting the Portals interface reduces the complexity of the MTK driver and greatly simplifies the driver implementation, while providing the necessary operations for supporting fast and efficient transfers of large data packets.

3.2 Cray Linux Environment (CLE) Operating System

The base operating system for the Cray XMT Series is CLE. CLE is a distributed system of service node and compute node components. The service nodes run the SUSE LINUX operating system and the compute nodes run the MTK operating system.

3.2.1 SUSE LINUX Operating System

Service nodes perform the functions needed to support users, administrators, and applications running on compute nodes. Each service node runs a full-featured version of SUSE LINUX. The operating systems on each service node run independently from the others.

Above the operating system level are specialized daemons and applications that perform functions unique to each service node. There are five basic types of service nodes: login, network, I/O, boot, and service database (SDB) service nodes; see Service Nodes on page 18 for further information.

3.2.2 MTK Operating System

The MTK is a single instance of an operating system that runs as a monolithic operating system across all compute nodes on the Cray XMT Series system. This differs from other Cray systems where the operating system runs independently on each compute node. The system calls for this operating system are based on the Berkeley Software Distribution of UNIX (BSD) and include Cray extensions.

The MTK includes the following features:

- Support for standard UNIX system operations such as fork/exec, signals, and so on.
- Scheduling for protection domains rather than for threads. The User Runtime Library provides thread management.
- Memory management that provides a global virtually contiguous data address space.
- Ability to share memory between processors.
- Native support for the RAMFS and network file system (NFS) client.

3.3 File Systems

Cray XMT Series systems use three types of file systems:

- The Lustre file system is used for data storage and is only accessible from the service nodes.
- The RAMFS is used for the root file system for the compute node operating system and has no corresponding physical storage.
- The NFS is used to transfer user applications from the service nodes to the compute node and it is accessible from both types of nodes.

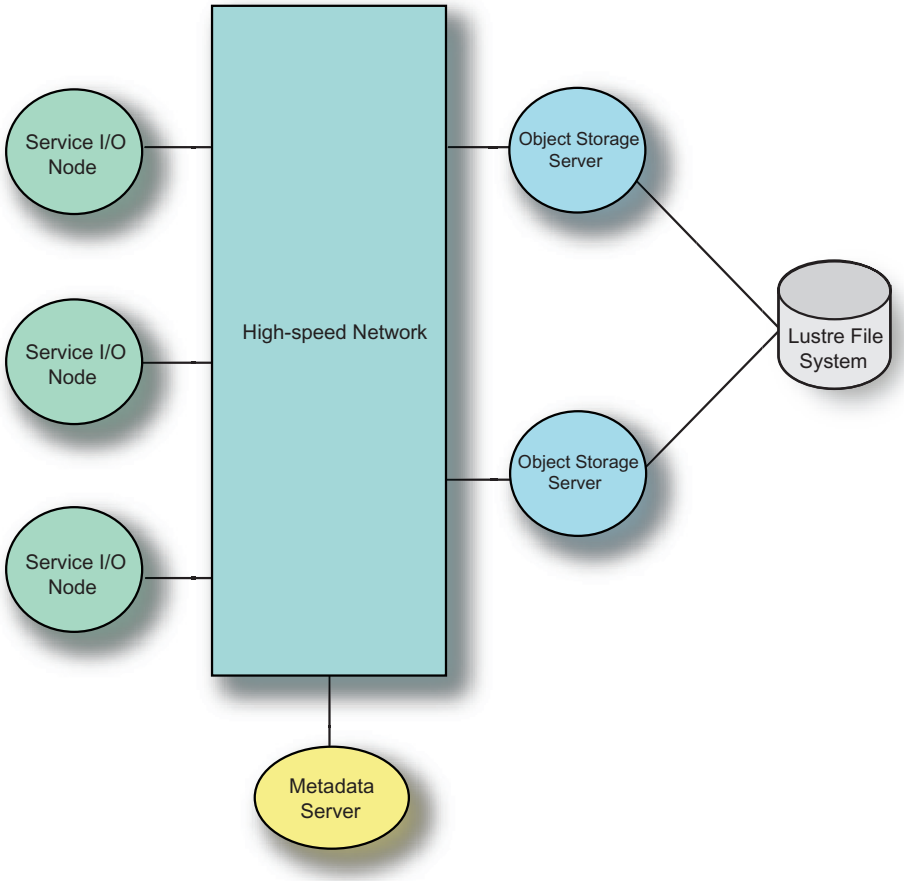
3.3.1 Lustre File System

The Lustre file system is hosted by the I/O service nodes. Lustre is a high-performance, highly scalable, POSIX-compliant parallel 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.

Cray XMT compute nodes access the Lustre file system directly by means of fsworkers, or indirectly by means of server and client applications that pass the data between the compute and service nodes. Use the LUC API to make remote procedure calls between the compute and service nodes and to allocate nearby memory buffers to store data.

Lustre separates file metadata from data objects. Each instance of a Lustre file system consists of one or more Object Storage Servers (OSSs) and a single active Metadata Server (MDS). Each OSS hosts one or more Object Storage Targets (OSTs). Lustre OSTs are backed by RAID storage. Applications store data on OSTs, and files can be striped across multiple OSTs.

Figure 7. Lustre Architecture on Cray XMT



3.3.2 Random Access Memory File System

The Cray XMT compute partition has access to the random access memory file system (RAMFS) file system. The RAMFS is allocated in memory and serves as the root file system for the compute nodes. It has no disk storage associated with it because it is used primarily for system utilities and shared libraries. RAMFS is loaded into Threadstorm memory at boot time and is accessed using standard I/O calls such as read and write.

Although RAMFS appears to be a normal file system, when the system goes down, all files in RAMFS are lost because it has no disk backup. If you create files in RAMFS that you want to retain across system boots, you must create a copy of the files in the NFS available on the service nodes.

In general, do not create large files in RAMFS. It is not a high-speed file system, and it is limited in size.

3.3.3 Network File System

The network file system (NFS) is the only file system on the Cray XMT that is directly accessible by both the service and compute nodes. The MTK operating system on Threadstorm compute nodes provides an NFS filesystem client that can directly access files on an NFS server.

The NFS is the primary means of transferring user applications from the service node to the compute node. By default, one login node exports all user directories to the compute node over NFS. User applications may read and write files in the user's home directory using standard system calls. NFS is not a high-speed file system, but may be used to provide a persistent storage space.

3.4 User Environment

The user environment is similar to the environment on a typical Linux workstation. Users log on to a Cray XMT login node.

Before working on the system, the user must do the following:

1. Set up a secure shell. The Cray XMT 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 XMT 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, 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 *Cray XMT Programming Environment User's Guide* and the `module`(1) and `modulefile`(4) man pages.

3.5 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 XMT Series system hardware, SUSE LINUX, and the Cray XT and Cray XMT utilities and resources.

Note: For information about standard SUSE LINUX administration, see http://www.tldp.org or http://www.suse.com.

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 *Cray XMT System Management*.

3.5.1 System Management Workstation

The System Management Workstation (SMW) 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 through the administration/HSS network; it does not use the system interconnection network.

3.5.2 Shared-root File System

The Cray XMT Series service nodes have 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 *Managing System Software for Cray XE Systems* manual and the *Cray XMT System Management*.

3.5.3 Configuration and Source Files

The administrator uses the 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` utility runs on login nodes and allows the administrator to view files as they would appear on a specified node. The `xtopview` utility also 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.

3.5.4 System Monitoring

There are three tools available in Cray XMT that you can use to monitor activity on the compute nodes.

- `mtatop` — Displays the processes and statistics for CPUs currently running on the Cray XMT for all available processors. Information is displayed in textual form on the command line. See the `mtatop`(1) man page and the *Cray XMT System Management*.
- Dashboard2 — Displays the processes and statistics for CPUs currently running on the Cray XMT for all available processors. Information is displayed using the Dashboard2 graphical user interface. Information is displayed two views: Resources, which shows all the CPU activity on the system as a whole for a particular type of activity, and Processors, which shows all activity on each processor. See the `dash`(1) man page and the *Cray XMT System Management*.
- `xmtconsole` — Displays all output from the Cray XMT compute node and accepts input from the command line and passes it to the MTK. The `xmtconsole` is useful for monitoring warning messages from the system. See the `xmtconsole`(8) man page and the *Cray XMT System Management*.

3.5.5 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.

Application Development [4]

The Cray XMT application development software is the set of software products and services that programmers use to build and run applications on Cray XMT compute nodes.

4.1 User Runtime Library

The user runtime library (`libr t`) contains runtime software support for future variables, synchronization, scheduling, event logging, compiler-generated parallelism, and debugging. The Cray XMT compilers link the user runtime library into your application at compile time.

For a list functions contained in the user runtime library, see the `runtime`(3) man page.

4.2 Lightweight User Communication Library (LUC) API

The Cray XMT Programming Environment contains a user-level library for LUC, `libluc.a`, that uses a C++ interface. The Linux client program and the Cray XMT both use the same sources and interfaces offered by the LUC library.

LUC implementation uses a client/server remote procedure call (RPC) paradigm where communication occurs between endpoints that sit on both the client and server. When the client has a package of data that it needs to deliver to the server, it sends a message to the server endpoint over the high-speed network. On the client side, the LUC library allocates nearby memory in the user buffer for data storage. On the server side, the LUC library allocates memory in nearby memory for the transfer of data, later copies the data to distributed memory, or leaves it in nearby memory. Data is transferred over Cray Seastar chips using Cray Fast I/O (FIO).

When creating a LUC application, you implement the endpoints as objects that are instantiated as either server-only, client-only, or both server and client objects that have a corresponding set of methods. The client and server can be created on either the compute-node or service-node side.

For more information about programming for LUC, see the *Cray XMT Programming Model* and the *Cray XMT Programming Environment User's Guide*.

4.3 Compiling Programs

The Cray XMT Programming Environment includes C and C++ compilers. The compilers translate C and C++ programs into Cray XMT object files.

The command used to invoke a compiler is called a *compilation driver*; it can be used to apply options at the compilation unit level. C or C++ pragmas apply options to selected portions of code or alter the effects of command-line options.

The `mta-pe` module contains the Cray XMT Programming Environment, which compiles for the compute node. This module is loaded by default when you log in. The `mta-linux-lib` module contains the Linux version of the LUC library. For other service node applications, use the standard Linux libraries.

4.3.1 Compiler Commands

The following compiler commands are available:

Compiler	Command
C	`cc`
C++	`c++`

For further information, see the *Cray XMT Programming Environment User's Guide*.

4.4 Running Applications

Applications are launched by using the `mtarun` command. The `mtarun` command connects to a daemon (`mtarund`) on the compute nodes. With this connection, your user environment changes so that your directories on the service node are shared on the compute nodes. The application that you specify in the *program_executable* option is then launched to run on the compute nodes.

After the application is launched, `mtarun` serves as the front end for the program, supplying it with the following services:

- Forwarding of standard I/O, such as `stdin`, `stdout`, `stderr`
- Signal forwarding for all sending signals
- Termination management, which redirects errors that caused the application to terminate or kills the application if `mtarun` terminates

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

4.5 Debugging Applications

You use the `mdb` debugger to debug your applications. The `mdb` debugger is based on the Free Software Foundation GDB debugger (version 3.5).

For more information, see the *Cray XMT Debugger Reference Guide* and the `mdb`(1) man page.

4.6 Monitoring Applications

The `mtatop` command displays the processes and CPU statistics currently running on the machine for all available processes on Threadstorm compute nodes. When run with the `p` option, `mtatop` shows process information for a specified process ID. The following information is provided for this option:

- The name and ID of the process
- The current state of the process, such as running, sleeping, and so on
- The scheduling priority of the process
- The number of CPUs that the process is using
- The amount of memory that the process is using
- The amount of CPU time accumulated by the process

This command can also be run using the `c` option to monitor CPU usage. For more information, see the `mtatop`(1) man page.

4.7 Measuring Performance

The Cray XMT Series system provides tools for the collection, display, and analysis of performance data. You can use the resultant analyses to optimize your application. For more information about the tools described in this section, see *Cray XMT Performance Tools User's Guide*.

4.7.1 Cray Apprentice2

Cray Apprentice2 is an interactive X Window System tool for displaying performance analysis data captured during program execution. It allows you to view traces that are generated when you compile with the `-trace` option, and it provides GUI versions of the `canal` and `bprof` tools. Note that Cray Apprentice2 is not a debugger, nor is it a simulator.

Cray Apprentice2 identifies many potential performance problem areas, including 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:

- Post-execution performance analysis tool that provides information about a program by examining data files that were created during program execution.
- Displays many types of performance data contingent on the data that was captured during execution.
- Reports time statistics for all processing elements and for individual routines.
- Shows total execution time, synchronization time, execution time for a subroutine, communication time, and the number of calls to a subroutine.

4.7.2 Canal

Canal is a compiler analysis tool. It uses information captured during compilation to produce an annotated source code listing showing the optimizations performed automatically by the compiler. You use the Canal listing to identify and correct code that the compiler cannot optimize. See the `canal`(1) man page for more information.

4.7.3 Tview

The `tview` command displays a trace file in one of three formats: XML, Apprentice2, or compressed (ZIP). It uses information captured during program execution to produce graphical displays showing performance metrics over time. Use the `tview` graphs to identify when a program is running slowly. See the `tview`(1) man page for more information.

4.7.4 Bprof

Bprof is a block profiling tool. It uses information captured during program execution to identify which functions are performing what amounts of work. When used with Tview, this can help you to identify the functions which consume the most time while producing the least work. See the `bprof`(1) man page for more information.

4.7.5 pproc

Pproc is a post-processing data conversion tool. It converts the data generated by Canal, Tview, and Bprof into a format that can be displayed within Cray Apprentice2. See the `pproc`(1) man page for more information.

4.7.6 ap2view

Ap2view is an XML data file viewer. It displays data from `.ap2` files that are created when you run `pproc`. See the `ap2view`(1) man page for more information.

4.7.7 Tprof

Tprof is a trace profiling tool. It provides a simple profile of the functions and parallel regions in the code, based on traces.

Cray Hardware Supervisory System (HSS) [5]

The Cray Hardware Supervisory System (HSS) is an integrated, independent system of hardware and software that monitors Cray XMT Series 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 HSS interfaces with all major hardware and software components of the system.

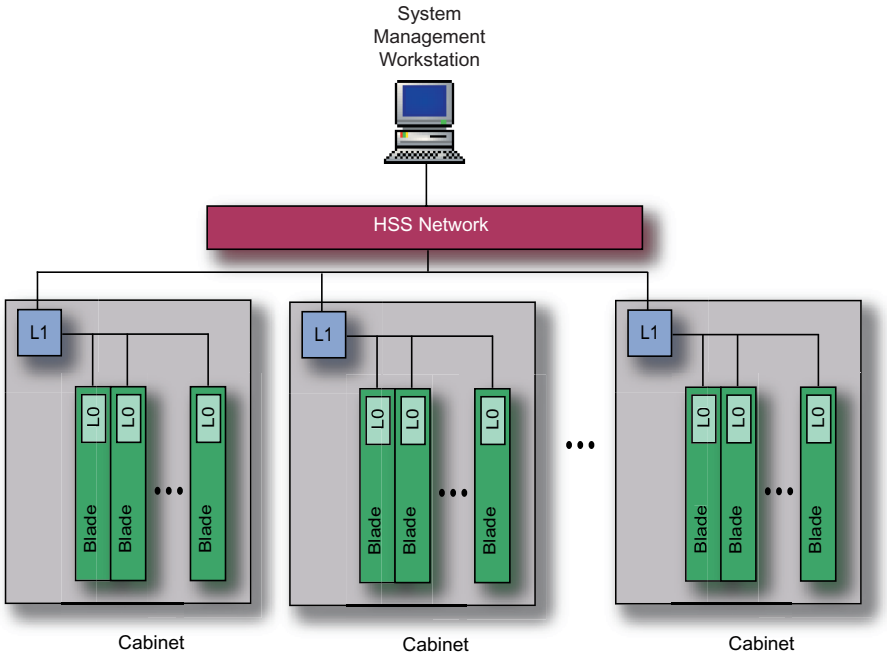
Because the HSS is a completely separate system with its own processors and network, the services that it provides do not take resources from running applications. In addition, if a component fails, the HSS continues to provide fault identification and recovery services and enables the functioning parts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

For detailed information about the HSS, see the *Managing System Software for Cray XE Systems* manual.

5.1 HSS Hardware

The hardware components of HSS are the HSS network, the SMW, the blade control processors (L0 controllers), and the cabinet control processors (L1 controllers). HSS hardware monitors compute and service node components, operating system heartbeats, power supplies, cooling fans, voltage regulators, and RAID systems.

Figure 8. HSS Components



5.1.1 HSS Network

The HSS network is an Ethernet connection between the SMW and the components that the HSS monitors. The network's function is to provide an efficient means of collecting status from and broadcasting messages to system components. The HSS network is separate from the system interconnection network.

Traffic on the HSS network is normally low, with occasional peaks of activity when major events occur. Even during peak activity, the level of traffic is well within the capacity of the network. 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.

5.1.2 System Management Workstation

The System Management Workstation (SMW) is the administrator's single-point interface for booting, monitoring, and managing Cray XMT 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 System Administration on page 27).

5.1.3 Hardware Controllers

At the lowest level of the HSS are the 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 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 L0 controllers also monitor board temperatures and the CLE heartbeat.

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

5.2 HSS Software

The HSS software consists of software monitors; the administrator's HSS 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 HSS Actions on page 40.

5.2.1 Software Monitors

The System Environment Data Collections (SEDC) HSS manager monitors the system health and records the environmental data (such as temperature) and the status of hardware components (such as power supplies, processors, and fans). SEDC can be set to run at all times or only when a client is listening. By default, SEDC is configured to scan the system hardware components automatically.

Resiliency communication agents (RCAs) run on the first compute node to boot.

Each RCA generates a periodic heartbeat message, enabling HSS to know when an RCA has failed. Failure of an RCA heartbeat is interpreted as a failure of the CLE operating system on that node.

5.2.2 HSS Administrator Interfaces

The HS provides both a command-line and a graphical interface. The `xtcli` command is the command line interface for managing the Cray XT system from the SMW. The `xtgui` command launches the graphical interface. In general, the administrator can perform any `xtcli` function with `xtgui` except boot.

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. HSS policy decisions determine how the fault is handled. The SMW logs all information it receives from the system to the SMW disk to ensure the information is not lost due to component failures.

5.3 HSS Actions

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

The HSS 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 HSS initiates some actions (event handling) in response to detected failures that, if left unattended, could cause worse failures. The HSS also initiates actions to prevent failed components from interfering with the operations of other components.

5.3.1 System Startup and Shutdown

The administrator starts a Cray XMT Series system by powering up the system and booting the software on the service nodes and compute nodes. Booting the system sets up the system interconnection network. Starting the operating system brings up the RCA.

A script, set up by the administrator, shuts the system down.

For logical machines, the administrator can boot, run diagnostics, run user applications, and power down without interfering with other logical machines as long as the HSS is running on the SMW and the machines have separate file systems.

For details about the startup and shutdown processes, see *Managing System Software for Cray XE and Cray XT Systems* and the `xtcli`(8) man page.

5.3.2 Event Probing

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

5.3.3 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:

- HSS probes
- Processes communicating through RCA daemons on compute and service nodes
- Other HSS processes running on L0 and L1 controllers

Event messages are time stamped and logged. If a compute or service blade fails, the HSS notifies the administrator.

5.3.4 Event Handling

The event handler evaluates messages from HSS probes and determines what to do about them. The HSS 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 HSS system are:

- Compute node failure. If a compute node fails, the entire compute node partition goes down and needs to be rebooted.
- Power supply failure. Power supplies have an N+1 configuration for each chassis in a cabinet; failure of an individual power supply does not cause an interrupt of a compute node.

In addition, the HSS transmits failure events over the HSS network to those components that have subscribed for the particular event, so that each component can make a local decision about how to deal with the fault. For example, both the L0 and L1 controllers contain code to react to critical faults without administrator intervention.