LA-UR-12-24015

Title: Cielo Computational Environment Usage Model With Mappings to ACE Requirements for the General Availability User Environment Capabilities Release Version 1.1

Author(s):

Vigil,Benny Manuel
Ballance, Robert
Haskell, Karen

Intended for: Required usage model for computing system

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Cielo Computational Environment Usage Model

## With Mappings to ACE Requirements for the General Availability User Environment Capabilities

## Release Version 1.1

Version 1.0: Bob Tomlinson, John Cerutti, Robert A. Ballance (Eds.)
Version 1.1: Manuel Vigil, Karen Haskell, Robert A. Ballance (Eds.)

Prepared by the Alliance for Computing at Extreme Scale (ACES),
a partnership of Los Alamos National Laboratory and Sandia National Laboratories.

Approved for public release, unlimited dissemination
**LA-UR 10-07492 (TBD)**

August 2012

**Los Alamos National Laboratory**
**Sandia National Laboratories**

## Disclaimer

Unless otherwise indicated, this information has been authored by an employee or employees of the Los Alamos National Security, LLC. (LANS), operator of the Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 with the U.S. Department of Energy. The U.S. Government has rights to use, reproduce, and distribute this information. The public may copy and use this information without charge, provided that this Notice and any statement of authorship are reproduced on all copies. Neither the Government nor LANS makes any warranty, express or implied, or assumes any liability or responsibility for the use of this information.

Bob Tomlinson – Los Alamos National Laboratory
Robert A. Ballance – Sandia National Laboratories
Karen H. Haskell – Sandia National Laboratories
(Editors)

# Abstract

Cielo is a massively parallel supercomputer funded by the DOE/NNSA Advanced Simulation and Computing (ASC) program, and operated by the Alliance for Computing at Extreme Scale (ACES), a partnership between Los Alamos National Laboratory (LANL) and Sandia National Laboratories (SNL). The primary Cielo compute platform is physically located at Los Alamos National Laboratory. This Cielo Computational Environment Usage Model documents the capabilities and the environment to be provided for the Q1 FY12 Level 2 Cielo Capability Computing (CCC) Platform Production Readiness Milestone. This document describes specific capabilities, tools, and procedures to support both local and remote users. The model is focused on the needs of the ASC user working in the secure computing environments at Lawrence Livermore National Laboratory (LLNL), Los Alamos National Laboratory, or Sandia National Laboratories, but also addresses the needs of users working in the unclassified environment.

The Cielo Computational Environment Usage Model maps the provided capabilities to the tri-Lab ASC Computing Environment (ACE) Version 8.0 requirements. The ACE requirements reflect the high performance computing requirements for the Production Readiness Milestone user environment capabilities of the ASC community. A description of ACE requirements met, and those requirements that are not met, are included in each section of this document. The Cielo Computing Environment, along with the ACE mappings, has been issued and reviewed throughout the tri-Lab community.

# Acknowledgements

Thanks are extended to the following individuals who contributed to the content and review of this document:

Hal Armstrong
Don Bragg
John Cerutti
Rob Cunningham
Doug Doerfler
Kim Edlund
Parks Fields
David Gunter
Karen Haskell
Lisa Ice
Barbara Jennings
Jeff Johnson
David Karelitz
Sue Kelly
Brett Kettering
Sue King
Jim Lujan
Susan McRee
Laura Monroe
David Montoya
John P. Noe
Dino Pavlakos
Mahesh Rajan
Phil Sena
Joel Stevenson
Judy Sturtevant
Manuel Vigil
Cheryl Wampler
Paul Weber
Meghan Wingate
John Zepper

# Change Table

| Version | Date | Comments |
|---|---|---|
| 1.0 | June 2011 | Initial release |
| 1.1 | July 2012 | Revised to incorporate Lustre as the parallel file system |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Table of Contents

## Figures and Tables

# Acronyms and Abbreviations

| | |
|---|---|
| ACE | ASC Computational Environment |
| ACES | Alliance for Computing at Extreme Scale (ACES), a partnership between Los Alamos National Laboratory (LANL) and Sandia National Laboratories (SNL) |
| ASC | Advanced Simulation and Computing |
| CCB | Change Control Board |
| CCC | Capability Computing Campaign |
| CEC | Capability Executive Committee |
| CPAC | Capability Planning Advisory Committee |
| CPI | Capability Performance Indicator |
| DAT | Dedicated Application Time |
| DST | Dedicated System Time |
| DSW | Directed Stockpile Work |
| EPR | (Tri-Lab) Expedited Priority Run |
| GA | General Availability –The machine is ready for production computing and any of the class of users the machine is targeted to serve may request and be granted accounts. |
| HPC | High Performance Computing |
| HPSS | High Performance Storage System |
| LA | Limited Availability. A machine is available for use but not for general availability (GA). A limited number of user accounts are added. |
| LANL | Los Alamos National Laboratory |
| LLNL | Lawrence Livermore National Laboratory |
| SARAPE | Synchronized Account Request Automated Process |
| SNL | Sandia National Laboratories |
| SWL | Synthetic Work Load |
| WAN | Wide Area Network |

# Introduction

The National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) Program Cielo computing system is the next-generation tri-Lab capability platform beyond the Purple system, which was retired in 2010. Cielo represents a 6x to 10x increase in the classified compute capability for NNSA over its predecessor. Cielo will be used by scientists at three national laboratories: Los Alamos (LANL), Sandia (SNL), and Lawrence Livermore (LLNL), to solve our nation's most demanding stockpile stewardship problems; that is, the large-scale application problems at the edge of our understanding of weapon physics. This fully functional, petascale system serves a diverse scientific and engineering workload. Cielo provides a robust code development and production environment scaled to meet or exceed the workload requirements.

Cielo's capabilities were designed and developed jointly by LANL and SNL under the Advanced Computing at Extreme Scale (ACES) partnership. The system is physically located at LANL in the Nicholas Metropolis Center for Modeling and Simulation. The facility is designed to scale with the increasingly powerful compute resources to deliver the required end-to-end services for users. Within the center are networking, archival storage, visualization servers, global file systems, and system software, all enhanced to support Cielo's size and architecture. The ACES partnership is responsible for the support of Cielo, including interactions with the vendor.

This Usage Model serves as a service level agreement or contract with the users of the system. It documents the Cielo system at the time of publication. For current information on the state of the system, including detailed usage guidelines and recommendations, users should refer to the ACES Web pages (Section 1.2).

This document provides a description of the expected user model for computing on Cielo and will describe capabilities planned to be available to users of Cielo when it is production ready. The usage model will be used as the capability definition for the 2012 "Cielo Capability Computing Platform Production Readiness" Level 2 Milestone.

The Level 2 milestones for this system are as follows:

December 30, 2010 Cielo Capability Computing Platform Integration Readiness

Cielo is ready for integration into the LANL computing center. System hardware delivery to LANL has been completed. The system installation at LANL has been completed. System software for the system has been delivered, tested, and demonstrated. On-site capability scaling testing has been completed. Cielo is ready for on-site integration into the local and remote computing infrastructure, including the user software environment.

This milestone was completed on December 8, 2010.

December 30, 2011 Cielo Capability Computing Platform Production Readiness

Cielo shall achieve Production Capability Readiness as defined by the Capability Platform Level 2 Milestones Working Group. In summary, this includes the following: the platform is made available for Capability Computing Campaign (CCC) capability work; all system software, tools, utilities, and user support processes are available and fully functional; ASC applications targeted for the platform are ported and made available to designers, analysts, and engineers; the platform has demonstrated acceptable reliability performance targets.

The ACES partners along with LLNL share responsibility for the Cielo user environment. As the hosts for Cielo, LANL and SNL have the greatest responsibilities. LANL and SNL User Support Groups will jointly provide customer support for Cielo to the tri-Labs and as such have the lead for user documentation. The ACES partnership is responsible for negotiating the Cielo computational environment, mapping of the ASC Computational Environment (ACE) requirements to the Cielo environment, and demonstrating that those requirements have been met. In addition, ACES project teams are charged with demonstrating important capabilities of the computing environment including full functionality of visualization tools, file transport between Cielo and remote site file systems, and the build environment for principal ASC codes. LANL, LLNL, and SNL are individually responsible for delivering unique capabilities in support of their users, porting important applications and libraries, and demonstrating remote capabilities. For example, capabilities tested by SNL and LLNL include user authorization and authentication, data transfer, file system, data management, and visualization, as well as porting, optimizing, and running in production mode a few key applications on a substantial number of Cielo nodes. ACES will provide application support in the form of “application readiness teams” who can assist code teams to get their applications running in production.

Production usage on the full platform is mediated by the Capability Compute System Scheduling Governance Model. The stockpile stewardship workload on ASC Capability Systems is apportioned through the Capability Computing Campaign (CCC) process.

Users of Cielo are expected to engage in up to eight activities, all of which are supported in the Capability Production Readiness milestone, and described in detail throughout this document:

1. Getting started (learning about the system, gaining access etc.)
2. Setting up the work environment
3. I/O and data migration
4. Application and system code development
5. Problem setup
6. Running the application to solve the problem
7. Processing simulation output
8. Tri-Lab coordinated operational support

Additionally, the Cielo computational environment maps the provided capabilities to the tri-Lab ACE Version 8.0 requirements. The ACE requirements reflect the high performance computing requirements for the General Availability user environment capabilities of the ASC community. A description of ACE requirements—those expected to be met and those not expected to be met—is included in italics at the beginning of each section of this document. Requirements that will be met are displayed in black text; requirements (or parts of requirements) not expected to be met are displayed in red text.

In summary, this document describes specific capabilities, tools, and procedures to support both local and remote users of Cielo according to ACE requirements. The ACE model is focused on the needs of the ASC user working in the secure computing environments at LANL, LLNL, and SNL.

# 1. Getting Started (Learning About the System, Gaining Access, etc.)

## 1.1. Platform Overview

Cielo is a massively parallel system consisting of three types of nodes:

- Compute nodes run application programs. All compute nodes run a Cray XE6 lightweight operating system known as Compute Node Linux (CNL). Compute nodes are further characterized based on their designated purpose as either normal compute nodes or visualization nodes.
- Service nodes handle support functions such as login management, I/O, and network management. All service nodes run a full Linux-based Cray operating system based on SuSE Linux Enterprise Server. Service nodes are further characterized based on their specific functionality, such as login nodes or file system support node.
- “External” login nodes provide additional user services, such as the application development environment, but they are not directly connected to the interconnection network (the mesh). The external login nodes run a stock SuSE SLES 11 Linux operating system.

Cielo compute and service nodes are tightly coupled with an interconnection network that supports fast MPI traffic, advanced synchronization and communication features such as globally addressable memory and atomic memory operations, as well as a fast I/O to a global, shared file system. The interconnection network is based on a 3D Torus topology using Cray’s Gemini high-speed interconnect. The Cray Gemini chip on each pair of nodes functions as the network interface and the router. The interconnection network architecture of Cielo is shown in Figure 1. The Cielo interconnect will be configured as an 18x8x24 3D Torus in Phase 1 (early FY11) and as a 16x12x24 in Phase 2 (late FY11).

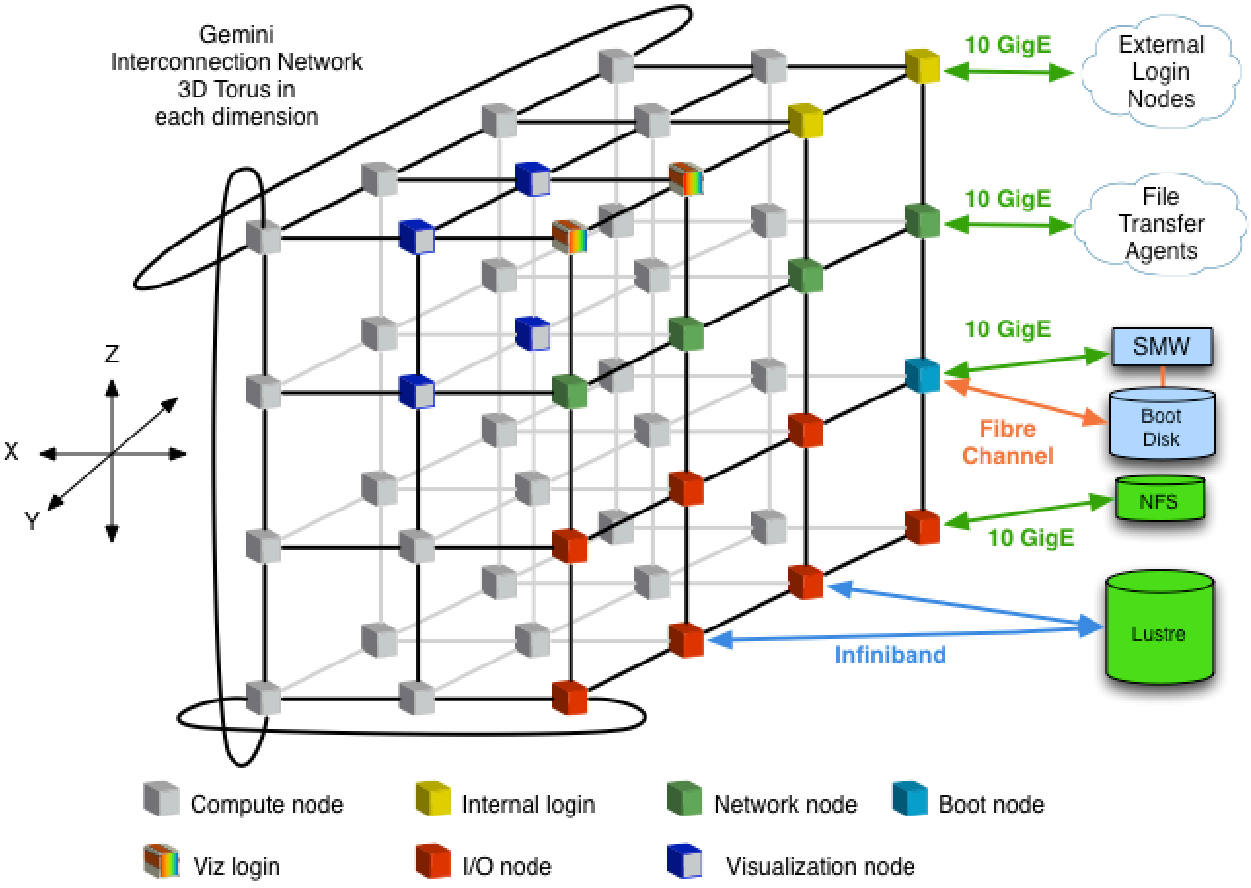


**Figure 1. Cielo System Architecture**

### 1.1.1. Compute Nodes

The compute partition of the complete Cielo system contains 8944 compute nodes. Each compute node contains 2 AMD G34 Opteron Magny-Cours 2.4 GHz 8 core processors for a total of 143,104 cores. Each core of the Magny-Cours processor is capable of 4 floating-point operations per clock period. Each compute node has eight slots for DDR3 memory DIMMS, four per socket. Each compute node will have a total of 32 GB of local memory. These AMD processor dies include L1 and L2 caches dedicated to each core as well as two 6 MB L3 caches shared by all the cores of each of the two dies, a DDR3 internal memory controller, and an HT3 HyperTransport interface. Two socket G34 processors with the associated memory and network interconnection constitute a Cielo compute node (see Figures 2 and 3).

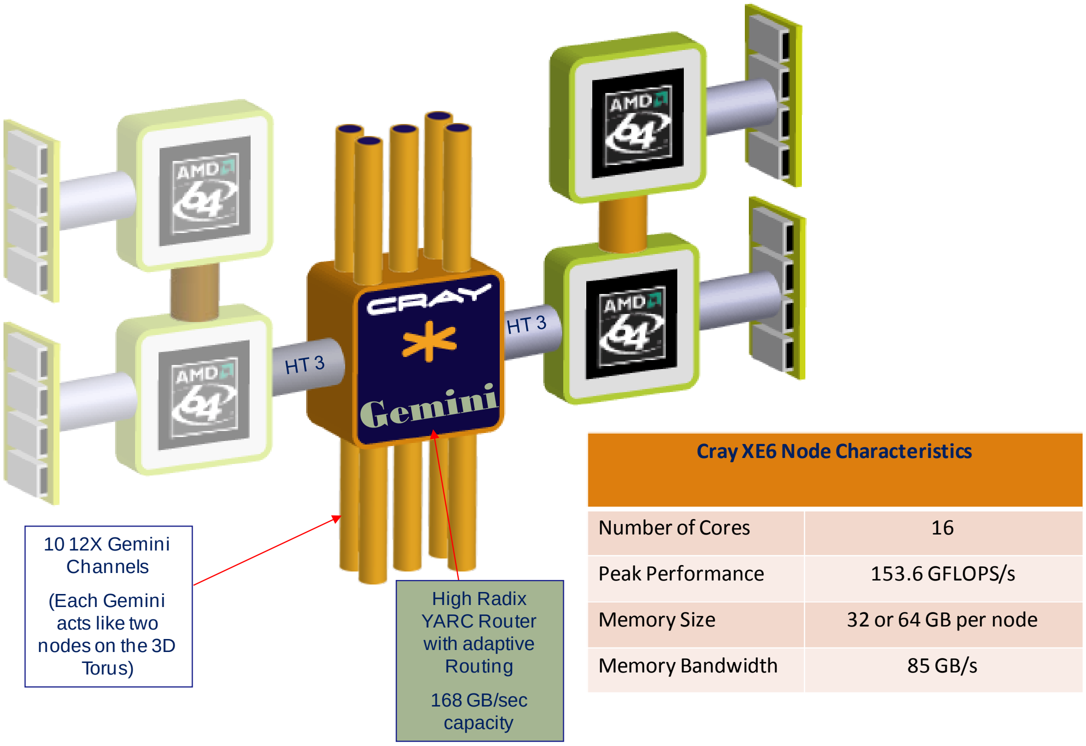
Using four DDR3 channels, each socket G34 has a peak socket memory bandwidth of 42.66 GB/s (85 GB/s per node) and memory latency under 100 ns. Memory is direct attached, with no FBDIMM or motherboard buffer chips. For fast cache coherency, two full bandwidth and two half bandwidth HT3 links are enabled between the sockets in a node. There are four memory channels per socket, with one DIMM per channel for maximum memory bandwidth.

Application communication protocols available between different nodes include MPI, SHMEM, UPC, Fortran 2008 with coarrays, and Global Arrays. Shared memory parallelization via OpenMP is supported between the cores of a single node. This provides a 16-way ccNUMA shared memory node where a core can address all of the node's available memory.

The compute node includes the following components:

- 2 AMD G34 sockets with Magny-Cours 2.4 GHz 8-core processors
- 8 DIMMs of DDR3 memory providing 32 GB of capacity
- 16 cores in two sockets sharing memory in ccNUMA mode
- Cray Gemini interconnect chip (supports two nodes)
- Memory bandwidth of 85 GB/s per node
- Memory latency less than 100 ns

**Visualization nodes** – approximately 5% of the compute nodes are configured for interactive visualization and analysis tasks. These nodes are configured with twice as much memory as other compute nodes, 64 GB per node.


AMD 64
AMD 64
AMD 64
AMD 64
HT 3
HT 3
CRAY
Gemini
10 12X Gemini Channels
(Each Gemini acts like two nodes on the 3D Torus)
High Radix YARC Router with adaptive Routing
168 GB/sec capacity

| Cray XE6 Node Characteristics | |
|---|---|
| Number of Cores | 16 |
| Peak Performance | 153.6 GFLOPS/s |
| Memory Size | 32 or 64 GB per node |
| Memory Bandwidth | 85 GB/s |

**Figure 2. Cielo Compute Node**

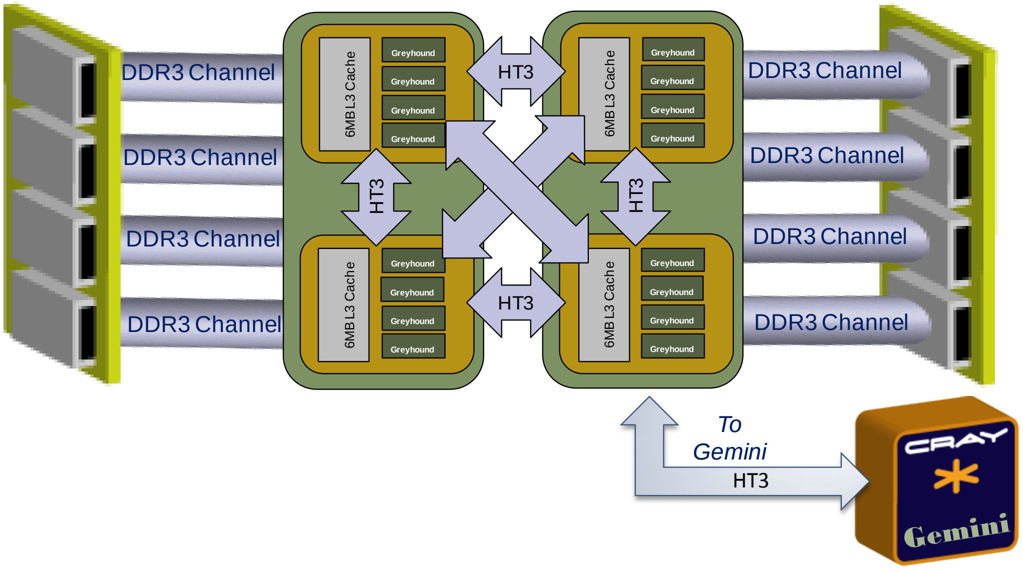


**Figure 3. Cielo Compute Node Details**

### 1.1.2. Service Nodes

Service nodes run a full Linux operating system and support standard I/O blades in addition to providing access to the high-speed mesh. Each service blade contains four service nodes. Service nodes are single-socket nodes using AMD Opteron 6-core processors, with 16 GB of DDR2 memory per node, and the same Gemini interconnect processors as compute nodes.

Cielo includes several types of service nodes, based on their hardware configuration. Each type performs dedicated functions and requires at least one PCIe card. The types include:

**Network Service Node:** Each network service node contains one dual-port 10 Gigabit Ethernet PCIe card that is connected to customer network storage devices. Network service nodes are used for user logins, visualization services, and file transfer activities.

**I/O Node:** Each I/O node contains one dual-port QDR Infiniband (IB) PCIe card. The IB ports connect to the Cielo file systems.

**Boot Node:** Each system requires one boot node. A boot node contains one 4-Gbit FC HBA and one GigE PCIe card. The FC HBA shall connect to the boot RAID and the GigE card shall connect to the System Management Workstation of the HSS.

Each service node has four slots for DDR2 memory DIMMs supporting 16 GB per node. Each DIMM has 72 bits: 64 data bits and 8 error-correcting code (ECC) bits. The four DIMM slots are configured on two memory buses to the Opteron processor.

### 1.1.3. External Login Nodes

External login nodes are nodes that allow users to compile and submit batch jobs to Cielo. They are “external” in the sense that they are not directly connected to Cielo’s high-speed mesh interconnect. External login nodes have four twelve-core AMD Opteron processors and 128 GB of memory and run a full Linux operating system. In Cielo documentation we refer to these as “external login nodes.” In other contexts users may also see them called “esLogin” or “front-end” nodes.

### 1.1.4. Cray Gemini Interconnect Network

Cielo includes a new generation of Cray-designed communication processor called Gemini. This communication accelerator is an active chip that is directly connected to the compute nodes through a HyperTransport3 port and to the 3D Torus through high-bandwidth custom “YARC” ports. The Gemini chip is a dual core communication accelerator, each port being dedicated to a compute node as illustrated in Figure 4.

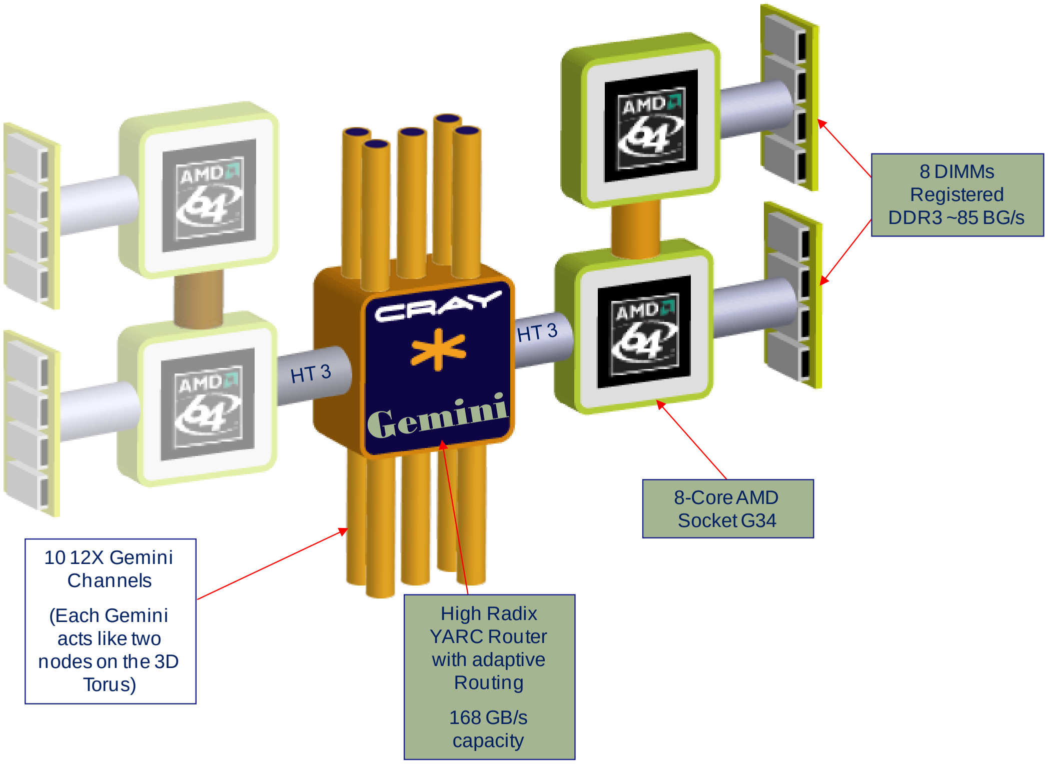

AMD 64
AMD 64
AMD 64
AMD 64
CRAY
Gemini
HT 3
HT 3
8 DIMMs Registered DDR3 ~85 BG/s
8-Core AMD Socket G34
10 12X Gemini Channels
(Each Gemini acts like two nodes on the 3D Torus)
High Radix YARC Router with adaptive Routing
168 GB/s capacity

**Figure 4. Gemini Interconnect for Two Cielo Compute Nodes**

An overall description of Cielo appears in Table 1.

**Table 1: Cielo Configuration Description**

| **Compute Partition** | Phase 1<br>(early FY11) | Phase 2<br>(late FY11) | |
|---|---|---|---|
| Blades | 1,676 (1,728-52) | 2,236 (2,304-68) | |
| Nodes | 6,704 (4 * 1,676) | 8,944 (4*2,236) | Compute Nodes<br>(4 nodes/blade) |
| Processors | 13,408 (2 * 6,704) | 17,888 (2*8,944) | 2-way AMD Magny-Cours Processors<br>(2 processors/node) |
| Cores | 107,264<br>(8 * 13,408) | 143,104<br>(8*17,888) | AMD Opteron cores<br>(8 Cores/Processor) |
| **Compute Node** | | | |
| Socket | | | AMD G34 Socket |
| Processor | | | 8-Core AMD Magny-Cours |
| CPU Clock | 2.4 | | GHz per Core |
| **FLOPs** | 4 | | FLOPs per Clock |
| Core | 9.6 (4 * 2.4) | | GigaFLOPs/S per Core |
| Processor | 76.8 (8 * 9.6) | | GigaFLOPs/S per Processor/Socket |
| Node | 153.6 (2 * 76.8) | | GigaFLOPs/S per Node |
| **Total Compute Partition FLOPs** | **1,029.73**<br>**(6,704 * 153.6)** | **1,373.79**<br>**(8,944*153.6)** | **TeraFLOPs/S** |
| **Memory** | | | |
| Processor | 16 (4 * 4) | | 4 GB DDR3 DIMMS per processor/socket |
| Compute Node | 32 ( 2 * 16) | | Gigabytes |
| Compute Memory | 202,496<br>(6,328 * 32) | 274,176<br>(8,568 * 32) | Gigabytes |
| Visualization Processor | 32 ( 4*8) | | 8 GB DDR3 DIMMs per processor/socket |
| Visualization Node | 64 ( 2*32) | | Gigabytes |
| Visualization Memory | 24,064 (376 * 64) | | Gigabytes |
| **Total Compute Partition Memory** | **226,560** | **298,242** | **Gigabytes** |

### 1.1.5. Cielito: Application Development System

In addition to Cielo itself, a compatible, but small-scale, application development system known as Cielito is available. Cielito resides on LANL's unclassified protected (Yellow) network.

The primary differences between Cielo and Cielito arise from their difference in size and the environment in which they live. Cielo is much larger than Cielito, and comprises more nodes, more disks, more networking, and a larger interconnect switch. Additionally, Cielo is installed on LANL's classified secure (Red) network, whereas Cielito is installed on LANL's unclassified protected (Yellow) network. Table 2 compares the two systems based upon their differences in size.

**Table 2: Cielo and Cielito Configuration Comparison**

| Configuration Description | Cielo (Phase 2) | Cielito |
|---|---:|---:|
| Number of nodes | 8,944 | 96 |
| Batch compute nodes | 8,568 | 64 |
| Batch compute cores | 137,088 | 1,024 |
| Memory per compute node (GB) | 32 | 32 |
| Interactive / visualization nodes | 376 | 4 |
| Visualization cores | 6016 | 64 |
| Memory per Viz node (GB) | 64 | 64 |
| Visualization login nodes | 8 | |
| External login nodes | 5 | 2 |
| Memory per login node | 128 | 128 |
| Service I/O nodes | 272 | 28 |
| Theoretical peak performance (Tflop) | 1,374 | |
| Parallel file systems (PB) | 10 | 0.5 |
| Levels of interconnect | 1 | 1 |

Because these two systems reside in different computing networks they have different file systems and different network connectivity to external systems including HPSS, visualization systems, outside networks, and other clusters.

There are several smaller platforms in addition to the primary Cielo platform; such satellite systems are located either at LANL or SNL, and will be dedicated to well-defined roles in support of Cielo.

## 1.2. Learning About the System

1. *Shall provide machine policy information. This will include but not be limited to limitations on interactive use. (moved from Section 1.1)*
2. *Shall clearly specify platform security policies. These shall include but not be limited to export control policies. (moved from Section 1.1)*

The ACES partners will provide complete documentation covering all areas of interest to users of the ASC Cielo systems, including policy information, machine configurations, system status and usage, code development, software, tools, news, manuals, tutorials, and training. This information provides both high level overviews and detailed descriptions of how to conduct work in the Cielo computational environment.

This documentation is Web-based. All ACES documentation is accessible from `http://aces.sandia.gov`. Non-Sandia tri-Lab Cielo account holders will be provided an unclassified cross-realm account in order to obtain their Web-based content from this site.

Policy information is clearly specified in the documents located on the policy Web page under `http://aces.sandia.gov/policies.` Policies include:

- Access protocols for the Cielo system
- Cielo usage model
- Purge policies and file quotas
- Required computer security briefings
- Relevant computing policies and procedures
- Code of conduct for classified computer users
- Cielo CCC process

These policy documents may in turn describe additional policies such as waste, fraud, and abuse; export control; good citizenship; resource allocation and job scheduling; software licensing; third-party software; and computer security.

### 1.2.1. Web-based System Documentation

1. *Shall provide web-based system information.*
2. *The web-based information system shall include open, secure, non-password protected pages that are available to the tri-labs.*
3. *The web-based information system shall include links to platform-specific web pages.*
4. *Shall provide current information about the system that is both open and secure.*
5. *Shall provide web pages that are searchable.*
6. *Shall provide a single, well-known point of entry for system information. This point of entry shall provide links to policy information.*
7. *Shall provide system information in a way to allow increasingly more detail as topic areas are explored.*
8. *Shall provide information about the main compute platform. It shall also provide information about supporting platforms; this shall include but not be limited to information about development, visualization, and open computing.*
9. *The ACE system information shall cover utilities for moving data between machines and within a platform group.*
10. *System information shall be kept current and shall include a last modification date and email address for the personnel responsible for the page content.*
11. *Shall provide a user feedback system.*
12. *Shall provide links to application project web pages that provide information about what applications run on what systems. (This requirement may only apply to secure environments given security policies.)*
13. *Shall maintain list of FAQs.*
14. *Shall provide access to current system configuration information.*
15. *Shall publish a "good neighbor" policy to advertise the policy for submitting jobs to the queues.. (moved from Section 6.1)*

In addition to the ACES Web portal, the tri-Labs provide a few well-established portals for all of their production computing platforms (see Table 3). Instructions for accessing Web pages on the unclassified protected and classified secure networks are typically available from the appropriate open network Web pages. In some cases full access to these Web pages may require authentication via cross-realm or a cryptocard.

**Table 3: Tri-Labs Web documentation links**

| Site | Open Network | Protected Network | Secure Network |
|---|---|---|---|
| LANL | http://hpc.lanl.gov | http://hpc.lanl.gov | http://hpc.lanl.gov |
| LLNL | http://hpc.llnl.gov | http://computing.llnl.gov | https://lc.llnl.gov |
| SNL | http://hpc.sandia.gov | https://computing.sandia.gov | https://computing-s.sandia.gov |

From these starting points, users can select from a range of topics associated with high performance computing at the Laboratories, including detailed user information for policies, getting started, systems status, hardware and software environment, code development, running jobs, data transfer, visualization and technical assistance contacts. Information on obtaining user accounts and accessing the system can also be obtained. Navigation is intuitive and allows users to progressively dive deeper for additional detail and related information.

Of special interest to Cielo users is the machine information available to the user community from the ACES Web pages. Information includes current configurations, planned activities, important announcements, software upgrades, dedicated time, and scheduled preventive maintenance.

Users are encouraged to provide feedback by means of a “Customer Feedback Form” located at `http://aces.sandia.gov/feedback`.

### 1.2.2. On-line System Information

1. *Shall provide a current set of man pages for each platform to include locally developed software.*

The path to man pages is automatically set for users upon login to a Cielo platform. A current set of man pages is available using the `man` utility. Most locally developed software and utilities also have man pages. Information for particular software products or tools is generally kept in the installation directory or in modulefile documentation for that product or tool, such as `/usr/projects/packages`. All systems also display the standard Department of Energy login banner.

### 1.2.3. Written System Documentation

1. *Shall provide a current set of (easily printable) documentation for each platform, especially compilers, libraries, MPI-IO and MPI that are available locally to users.*

2. *Shall provide a "quickstart" guide for users of new platforms. (moved from Section 4.2)*

Cielo users have a wealth of printable system documentation available covering topics related to all ACES platforms, and topics specific to Cielo. Documentation common to all platforms can be found in the ACES Web pages `http://aces.sandia.gov`. The wide range of topics includes manuals on using the ACES Moab batch system, links to Cray documentation, file transfer tools, libraries, and math packages. Tutorials on a number of parallel programming topics are also available, as well as linked external locations. Tutorials are in HTML format, which can easily be printed directly or converted to PDF format for printing.

### 1.2.4. Training

1. *Shall provide vendor and locally developed training information.*
2. *Shall provide training materials that are electronically accessible.*
3. *Shall provide trainer contact information.*
4. *Shall provide periodic tailored training for remote users at their sites. Targeted to specific groups like end-users and developers.*

Periodically, ACES offers on-site workshops focusing on parallel programming, parallel tools, and the use of the Cielo machine. Introductory level workshops are intended for new users, with the goals of improving user productivity and minimizing the obstacles typically encountered by new users of such complex systems. Introductory level workshops typically include both lectures and hands-on exercises using the actual machines.

Other workshops are targeted towards more experienced users and can cover a range of topics related to new technologies, performance/programming tools, vendor product training, and other topics as requested by ACES users, researchers, and staff. Some workshops include "hands-on" exercise time. Invited trainers from other institutions and vendor companies often teach at these topic-specific and advanced workshops.

Training/workshop materials are accessible from the ACES Web pages. Announcements and additional advertising takes place through tri-Laboratory email lists.

### 1.2.5. Consulting

Users may also learn about the Cielo system by contacting ACES User Support staff via the methods described in Section 8.

## 1.3. Gaining Access to the System

### 1.3.1. Governance of System Usage

1. *Shall provide the ability to request accounts online. This shall include instructions and criteria for getting accounts.*
2. *Shall provide information about tri-lab access policies and procedures. This shall cover all ASC computing resources with links to platform specific web pages.*
3. *Shall provide site-specific access policies and procedures that specify differences from tri-lab standards. The information system shall also provide links to tri-lab and platform specific web pages.*

4. *Shall provide guidance to users concerning which platform they should consider applying for an account based on the machine maturity and their intended usage.*
5. *Shall provide clear guidance on the authorization approval criteria.*
6. *Shall provide a tri-lab service to process account requests, with user transparent links to site-local authorization mechanisms.*

A tri-Lab team was formed in the fall of 2005 to establish a new governance model for allocating and scheduling the stockpile stewardship workload on the ASC Capability Systems according to programmatic priority. The "Capability Compute System Scheduling Governance Model" document produced by this team described the model fully. Cielo will continue to use the Capability Computing Campaign (CCC) model. The model governs the allocation of capability computing resources for Weapons Laboratory deliverables that cannot be reasonably attempted on other resources and that merit priority on this class of resource. The process outlined in this document describes how capability work can be evaluated and allocated resources, while also preserving a highly effective use of the systems in order to provide the broadest possible benefit to the Program.

This model intends to make effective use of the machine, as far as possible within the context of its two major objectives, both by minimizing idle cycles and by enhancing the probability of productive and useful capability calculations. The two major objectives of this model are:

- Ensure that the capability system resources are allocated on a priority-driven basis according to the program mission
- Use ASC Capability Systems for performing the large capability jobs for which they were intended

Another objective is to keep the prioritization and allocation processes as simple and effective as possible, in order to help ensure success of the objectives. Considerations include how to organize the workload for submitting proposals and allocating resources, definitions for capability mode computing, how to prioritize the proposals, implementation of the model, and reporting requirements. This governance model provides clear guidance on the authorization approval criteria for getting an account on the Cielo platforms.

Cielo will be allocated and scheduled according to the CCC policy. Calls for proposals and the method of responding will be announced by the ASC program.

A separate process to gain access to Cielo, called Standby, will be provided to each Laboratory to be managed by that Laboratory. Site allocations for the standby resources will be determined by the Capability Executive Committee.

**Note on early access during system deployment:**

Cielo CCC 1 will initially allocate a major portion of the machine for this activity. Additional nodes will be added as the Cielo system grows through the end of CCC 1. A capability job is defined as a job that uses 30% or more of the available nodes. CCCs that propose running capability jobs for most of their allocation are the most desired. Usage will be tracked to ensure time is used as allocated and job sizes meet proposal requirements.

The CPAC plans to allocate Cielo for time periods of roughly 6 months. PIs who submitted proposals for earlier campaigns on Purple must resubmit for Cielo. Cielo CCC 1 ranking will be made independent of any previous campaign ranking. PIs of chosen CCCs are expected to prepare a presentation on their results upon completion of their CCC.

Cielo CCC 1 will provide a transition period before full Cielo General Availability (GA). The early part of CCC 1 will be based on a Limited Availability period, where there may still be a few issues to be resolved before the machine becomes GA. During this time there will likely be system down time periods for resolving machine issues. GA is when the machine is ready for production computing.

Proposals for CCC 1 should provide plans for scaling of applications to use the larger number of nodes and processors available on Cielo. During the early part of Campaign 1 each CCC is highly encouraged to work on demonstrating this application scalability. Priority will be given to those proposals that have a well-defined plan to attempt this application scalability ramp up.

Cielo CCC 2 call for proposals will take place in the summer of 2011. Cielo CCC 2 will allocate the Cielo resources, including the scheduled FY11 upgrades.

### 1.3.2. Account and Password Management (Authorization)

1. *Shall minimize the number of authentications users are required to make while meeting security requirements.*
2. *Shall support cross-laboratory honoring of credentials.*
3. *Automated authentication support for batch jobs (e.g., cron), for nightly builds, regression testing, etc. (including documented solutions)*
4. *Provide the ability for multiple users to have access control of a common running applicatio*n

The methods for requesting accounts on ACES platforms, including instructions, criteria, policies, and procedures, can be found online at the open network Web pages at `http://hpc.sandia.gov/aces` or from the ACES Web pages (`https://aces.sandia.gov` for those users who already have the credentials required to access to that site).

### 1.3.3. Gaining Access to the Machine (Authentication)

1. *Shall provide real-time information via the web about system availability and system loading.*
2. *Shall provide an estimated uptime when the system is unavailable.*
3. *Shall provide the capability for users to determine the size of a job that can be run.*
4. *Shall be able to determine the state of the current queue and job limits.*
5. *Shall support use of lightweight scripts access to real-time availability information. (e.g., Perl, CSH)*
6. *Shall provide users with current maintenance and upgrade schedules.*
7. *Shall provide accurate information about resources available to the user, to include disk and node resources (memory, CPU, etc.). It is expected that the system will have removed any transient resources allocated the previous user.*
8. *Shall limit access to secure job status information.*

9. *A utility that returns the set of queues (authorized for user) that meet resource requirements (nodes, memory, etc.)*
10. *Every user should be able to get an interactive login (at least one) (with modules) for compiles or whatever, but there may have to be a limit on the number they can have.*
11. *Shall offer a single web site to provide information about platform status and availability to the tri-lab environment. This site shall also provide recommendations about which platforms to avoid due to other commitments.*

As with other tri-Lab production systems, access to Cielo systems differs between the unclassified protected network and the classified secure network, and whether access is coming from within LANL or from outside LANL. Access procedures are summarized below. A complete discussion with examples can be found on the ACES Web pages (https://aces.sandia.gov).

#### 1.3.3.1. Unclassified System Access

Authorized unclassified users can access the Cielito Application Development cluster from inside the LANL unclassified protected network or from outside LANL over the Internet; the process for each site may differ. From within the LANL unclassified protected (Yellow) network, users are required to use an SSH client (version 2 compatible) to connect to Cielito.

Access originating outside LANL over the Internet, such as from another institution, home, hotel, etc., will SSH to a Cielito bastion host using a LANL Cryptocard, then SSH to the Cielito login node. Further details about the Cielito host, login node, and other help are available from the ACES Web pages. LANL Cryptocards can be obtained through processes documented there, under “Getting an Account.”

#### 1.3.3.2. Classified System Access

Access to the classified Cielo system may originate from within the LANL classified secure network, or from outside LANL over SecureNet. From within LANL, SSH (version 2 compatible) must be used to connect to a login node. Authentication may be accomplished by use of a LANL Secure Restricted Network (Red) Cryptocard or from forwarded Kerberos credentials.

Access from within the LLNL or SNL classified networks requires an initial SSH connection to a LANL File Transfer Agent (FTA). After the LLNL or SNL user logs into FTA and obtains a local Kerberos credential (using their LLNL or SNL identity), they may then SSH to Cielo without the need to enter a password.

Other DOE sites may access Cielo over SecureNet using SSH and a Red LANL Cryptocard.

### 1.3.4. System Availability and Scheduling Information

In addition to ACES Web-based status information, LANL provides a unique email status list for every machine. Users are automatically and unconditionally subscribed to an email list for every HPC cluster on which they have an account. These lists receive notification of important events for specific platforms. Email notifications are communicated in both the unclassified protected (Yellow) and classified secure (Red) networks.

### 1.3.5. Requests for Priority Jobs

Cielo provides for expedited priority runs according to the Dedicated Application Time (DAT) procedure. Expedited priority requests, DATs, and user-initiated Dedicated System Times (DSTs) are requested through an online form on the ACES Web pages; these requests are reviewed by the standard programmatic Expedited Priority Run (EPR) review process. The expectation is that most DATs and DSTs will occur during normal working hours in order to facilitate system support for the effort. Approved DATs and DSTs will be published on the Web and via email.

To encourage scalability and testing on large portions of Cielo, ACES will support regularly scheduled DATs for large, but short-running jobs.

# 2. Setting Up the Work Environment

## 2.1. File System Standards and Documentation

1. *Shall use standardized path names for the location of applications and project areas. (e.g., /projects/<project-name>/) (Note: To the extent standards are documented)*
2. *Shall use configurations for home directories, scratch areas and archival storage that are available and consistent across platforms.*
3. *Shall provide current documentation on configurations of home directories, scratch space and parallel file systems.*
4. *File names should implement similar policies. (e.g., scratch, netscratch, vizscratch.)*
5. *Shall provide appropriate information for all file systems. This shall include but not be limited to back-up policy, quota and purge policy.*
6. *Shall provide tri-Lab access to common disk space to enable a remote build capability.*
7. *At least one file system shall be provided backup for source code.*

LANL employs a consistent file system directory structure and naming conventions across all of its production platforms, including Cielo. To the extent that tri-Lab standards for path naming conventions exist and are documented, the standards are met in this directory structure. Examples of these are described below, and summarized in Table 4. Note that the italicized "*N*" indicates a numeric substitution, depending on the situation. Characteristics of the file systems will be discussed in more detail in later sections of this document.

- `/users/`*`username`* denotes a user's home directory that is automatically created when a new account is created.
- `/netscratch/`*`username`* refers to a user's directory on a globally mounted NFS file system (not a parallel file system).
- `/lscratch`*`N/username`* indicates a directory on the Lustre parallel file system. The naming convention distinguishes Lustre-based file systems from other, Panasas-based file systems visible on the FTA nodes at LANL.
  A scratch directory is automatically created when a new account is created. Scratch space is intended for parallel I/O. It is not backed up and is subject to periodic purges.
- `/usr/projects/` Project space for code teams that have a direct correlation to the program funding the compute cycles on the machine. These directories are for source code repositories, common data sets, EOS info, binaries of codes from an ASC or IC that support multiple users and/or platforms, or multiple release versions for users (i.e., Production version, test version, future release).

  Each project must have an Owner. The Owner must understand programmatic need to know for access to the source code and binaries and input decks for the project. Access to the space must be controlled through a UNIX group. Project spaces are officially requested through ACES User Support. In the case of Cielo, project spaces should support a Capability Campaign on the machine.

  All `/usr/projects/` spaces are automatically mounted from servers elsewhere in the LANL network. They are mounted on demand and may be unmounted after long idle times. If your project space appears to be missing, simply `cd` to your `/usr/projects/directory` to mount it. The `ls` command alone will not force a mount.

LANL prefers to automount in order to simplify management of the mount points and to reduce kernel tables on the file system servers. Automounting file systems has also helped keep machines from “hanging” on external NFS servers during the odd server outage. Note that home directories are also automounted, but a user login triggers the mount when the Linux shell starts.

- `/usr/projects/packages/` Subdirectories created here are NFS-mounted and intended for the installation of packages which are unofficially supported by users of the machine but that do NOT have need to know requirements or may be of benefit to all the users of the machine. This area is typically where tools that all code teams use that some person has managed to collect and support. (e.g., special editors, text conversion tools, special versions of some GNU products). The HPC administrative team also installs products that are not officially supported under a maintenance agreement into `/usr/projects/packages`.

  Every package must have an Owner. The owner of the “package” is responsible for keeping it updated and creating any support mechanisms necessary for the user community (modules, start-up scripts, etc.). `/usr/projects/packages` is allocated a fixed amount of space as a whole. It is small.

  On Cielo, directories within `/usr/projects/packages/` are automatically mounted on first use. This means that a user may not “see” a directory (e.g., via the ‘`ls`’ command) until the directory has been accessed.

- `/udsl/fs1/` denotes a file system configured to support dynamically shared libraries and other files that require scalable read-access from the compute nodes. Files located in `/udsl/fs1` are writable from login nodes, but read-only from the compute nodes. This file system is managed in the same way as `/usr/projects`.

**Table 4: Cielo File System Naming Conventions**

| File System / Directory | Purpose | Backup | Quota |
|---|---|---|---|
| /users/*username* | Home directory | Yes | Yes |
| /users/*username*/.snapshot | Home directory online backup | Yes | Yes |
| /netscratch/*username* | NFS mounted global scratch space | No | Yes |
| /lscratch*N*/*username* | Lustre parallel file system (subject to purge) | No | No |
| /usr/projects/ | User project directories | Yes | Yes |
| /usr/projects/packages/ | Third party and user-supported software | Yes | Yes |
| /udsl/fs1/ | Project directories; configured to support scalable loading of read-only files, including dynamic shared objects | No | Yes |

Documentation for Cielo file systems, including backup, quota and purge policies, may be found in the ACES Website: `http://aces.sandia.gov/red_filesystems/.`

## 2.2. Setting up User Groups, Environment Variables, Modules, etc.

1. *Shall provide a documented and maintained skeleton template cshrc file and others that provides the basics for setting up the user environment.*
2. *Shall allow users to change the login shell without requiring sys admin intervention. (Note: with caveats.)*
3. *Shall provide home directories that are portable across multiple architectures*
4. *Include a standard architecture-specific environment initialization capability. (e.g., cshrc.aix, .cshrc.tru64, etc.)*
5. *Shall have a standard module capability on all platforms to manage software dependencies.*
6. *Shall standardize the system environment variables across platforms (to the degree possible) to enable portable user scripts.*
7. *Shall provide paths to common software that are standardized across platforms. (Note: To the degree that paths are standardized and documented)*
8. *Provide other initialization scripts (termcap, emacs, vim)*
9. *Shall provide documented processes that allow tracking additions and deletions to groups.*
10. *Shall provide a service for group owners to set up and manage groups.*
11. *Shall provide a hierarchical group capability. (Used to expand the number of groups that a user can belong to.)*
12. *Shall provide a user level mechanism to change the default group.*
13. *Shall provide the capability to set up groups for export control purposes.*
14. *Shall have a single gid space across the complex to enable imported tar files to map to the correct group.*

The Cielo user environment resembles other tri-Lab production systems. User accounts are created with a complete set of default, standardized startup scripts (dot files). The master scripts (.profile, .cshrc, .login, etc.) serve to initialize the user environment. Supported shells include tcsh and bash. These scripts combine to set up the user environment with necessary environment variables, paths, aliases, standard host type environment variables, etc. Users can modify any of these files, and also obtain fresh copies at any time from the standard location of `/etc/skel/`. Most software packages and libraries, including third-party software, have an associated modulefile, and users are expected to use these modulefiles to access the software.

The following commands are useful for viewing and loading modulefiles

```
module list
module avail
module load [module_name]
```

More information on modulefiles may be found using the module man pages and the user documentation at the ACES Web pages.

User groups are managed by the Cielo account specialists. Users having LANL crytpocard credentials can originate group via the network registry Website, `http://register.lanl.gov`, in the unclassified protected (Yellow) network, or the classified secure (Red) network. After establishing a file-sharing group, users may then add themselves and other users to this group. Users without LANL cryptocards must request group creation and management by contacting ACES User Support.

All file-sharing group management is propagated to HPC platforms in both networks. Cielo also recognizes “secure only” file-sharing groups. Assignment of the default user group occurs when the user account is first created and managed by Cielo system administrators. In order to promote security, the user default group is the same as the userid and may not be changed.

# 3. I/O and Data Migration

ASC efforts are directed at developing and providing tools and services that enable Distance Computing (DisCom). A goal of ACES, working closely with the previous DisCom program element, is to provide adequate tools and high-performance throughput for both local and remote users.

High performance data movement and interactive visualization over the Wide Area Network (WAN) require common solutions based on the DisCom network architecture. Throughput issues between sites can be resolved only through coordination and cooperation among the sites.

The DisCom WAN was implemented by Qwest in 2006. The WAN has bandwidth of 10 Gbps, and deploys redundant bandwidth among the tri-Labs. Redundant bandwidth incorporates a "north" route (through Denver) and a "south" route through El Paso (see Figure 5). The WAN uses TCP/IP over Ethernet.

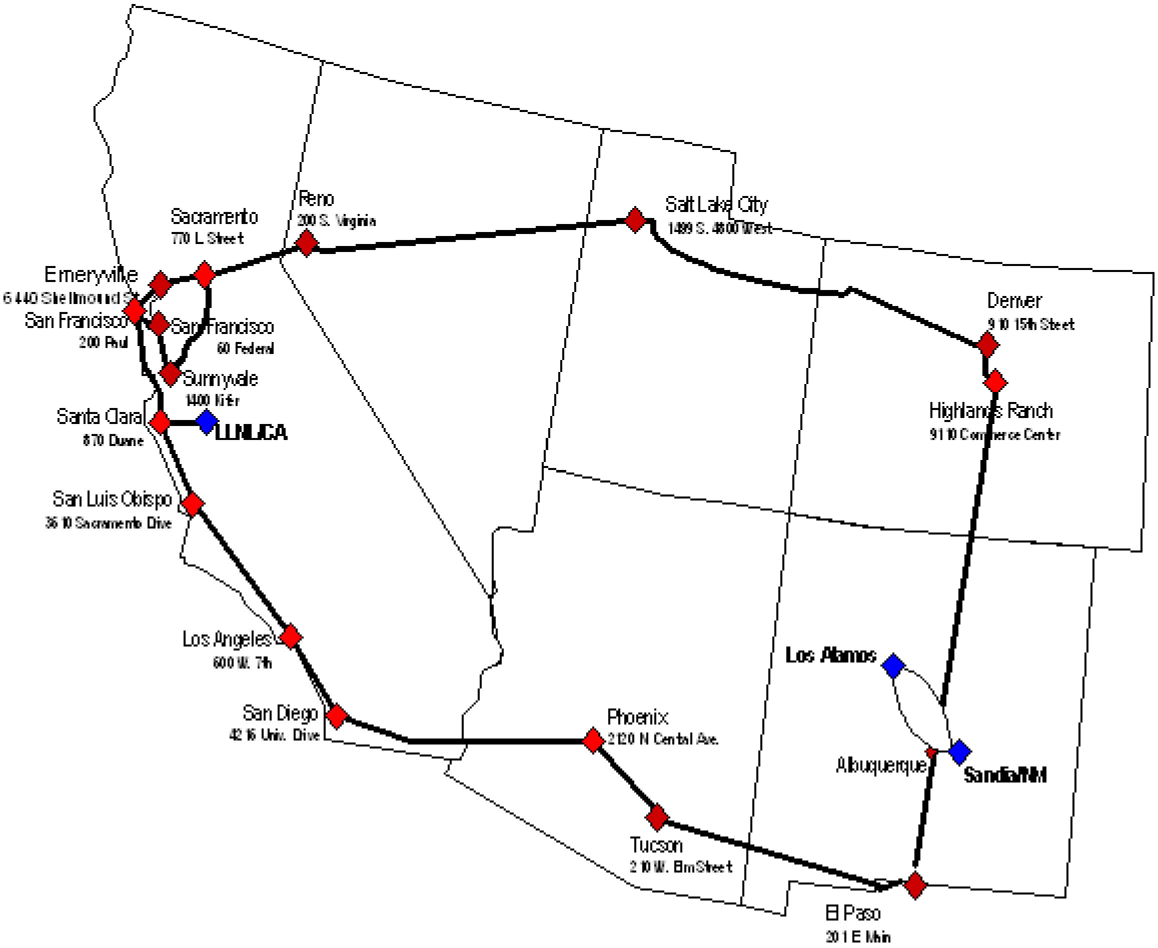


**Figure 5. DisCom WAN 2009**

## 3.1. Tools for Transferring Files

1. *Shall provide the infrastructure to support high speed remote bulk data transfer that can be executed unattended and is resilient..*
2. *Shall provide tools for high performance and scalable data transfer that can be initiated from either side. (currently pftp)*
3. *Shall support all combinations of high-speed (parallel) data transfer from local parallel file system/archive to remote parallel file system/archive on both the classified and unclassified networks. This shall include local to local, local to remote, remote to local and remote to remote.*
4. *Shall provide local files systems that are accessible over the WAN.*

Cielo offloads all file transfer operations to a cluster of File Transfer Agents (FTA) that have tools for file transfers and archiving. A FTA will be used for any data movement to remote sites or local HPSS. The parallel file system, along with home directories and `/usr/projects/` are mounted on each FTA. The infrastructure support for high speed bulk data transfers includes jumbo frame 10 Gigabit Ethernet links on the Red-FTAs, and some of HPSS storage movers. On the classified network, the FTA nodes are `red-fta01.lanl.gov` … `red-fta10.lanl.gov`.

The Cielo data transfer tool mix includes HSI, HTAR, PFTP, and PSI. The tools are described below. *However, users are cautioned that HSI and PSI each maintains its own data structures in HPSS, and so are not completely inter-operable.*

HSI, HTAR, and PFTP are all part of the tri-Labs common data transfer interface and may be “wrapped” at sites external to Cielo for user convenience (e.g., SNL and LLNL use of Hopper.) This allows remote site users to move data using familiar tools. The ASC tri-Lab community is developing tools to transfer files directly from one HPSS to another without having to stage data to a file system first. Figure 6 shows a high-level diagram of the relationships among tri-Labs sites and tools. The arrows indicate the direction of the originating connection and do not indicate the data transfer direction.

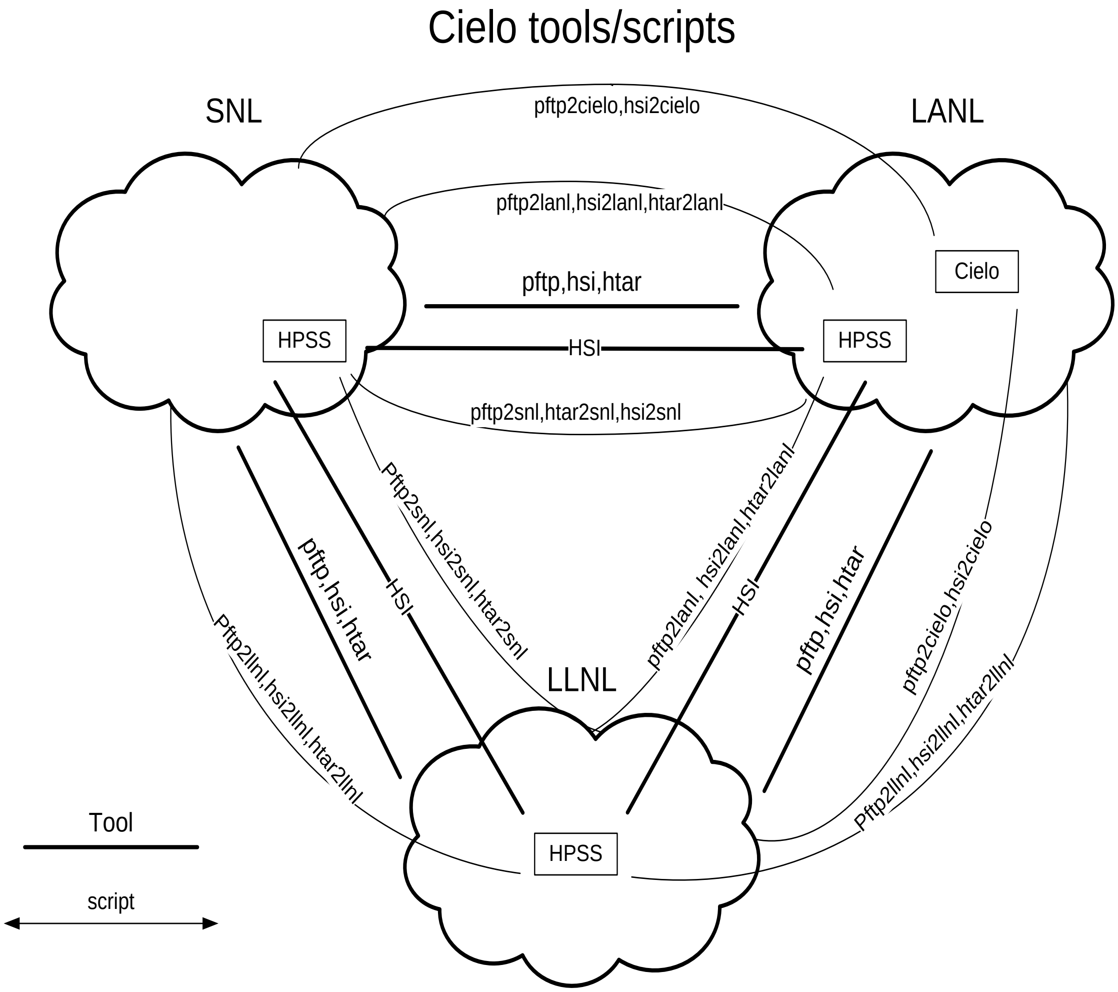

Cielo tools/scripts
SNL
pftp2cielo,hsi2cielo
LANL
pftp2lanl,hsi2lanl,htar2lanl
Cielo
pftp,hsi,htar
HPSS
HSI
HPSS
pftp2snl,htar2snl,hsi2snl
Pftp2snl,hsi2snl,htar2snl
pftp2lanl, hsi2lanl,htar2lanl
pftp,hsi,htar
HSI
HSI
pftp,hsi,htar
pftp2cielo,hsi2cielo
LLNL
Pftp2llnl,hsi2llnl,htar2llnl
Pftp2llnl,hsi2llnl,htar2llnl
Tool
HPSS
script

**Figure 6. Data Management Tools**

Multiple tools are provided to the user in order to meet individual needs, which may include specific functionality, performance, resiliency, and reliability. Documentation and assistance will be provided by ACES User Support to aid in the user's selection of tools and expected outcomes, including expected transfer rates, resilience, and authentication requirements and tool availability. The Storage and Data transfer teams will provide input and tier-three support for problems that cannot be resolved by ACES User Support.

The following tools are supported at one or more endpoints in the Cielo computational environment:

**HSI** is both a powerful HPSS-aware and intra-platform data transfer tool. For example, HSI provides HPSS-to-HPSS transfer capability – a way of transferring files directly between HPSS systems without staging files. HSI provides Unix-like commands, familiar to the user (e.g., `rm`, `find`, `chgrp`, etc.) HSI has a non-HPSS server, allowing an HSI client to access a platform's file system, using the same tool. HSI services are universally available at all three laboratories.

**HTAR** is a parallel transfer tool used at all three sites to aggregate small files into a single data stream. This pooling of data connections improves HPSS and network performance.

**PFTP** provides high-speed parallel transfer capabilities to HPSS and platforms' file systems. A compatible DisCom PFTP server is provided for platform services. A compatible multi-node PFTP client is also provided to improve large file performance. The ASC tri-Lab community supports PFTP, universally used within and between the laboratories. PFTP servers exist at all three labs, and scripts are provided to transfer files between selected classified ASC tri-Lab machines. Interactive authentication is not permitted (e.g., entering a user name and password). Instead, PFTP automatically performs the authentication if a valid Kerberos credential is available.

**PSI** is a powerful and high performance user interface to HPSS systems, which is modeled largely after UNIX file system commands. PSI uses parallelization to significantly increase the performance of a single file transfer, to provide simultaneous file transfers, to allow for conditional and restart of transfers, and to increase the performance of metadata operations (e.g., `chmod`, `du`, `find`, `ls`.) within HPSS. In addition, via HTAR, PSI provides the optional ability to do automatic and near-transparent aggregation of small file transfers to HPSS, thereby reducing the amount of meta-data stored within an HPSS system. PSI also has the capability to transfer data and files from one local file system to another.

**NOTE:** If file aggregation is selected for a recursive PSI transfer, then PSI aggregates in such a way as to preserve a client's directory structure. If the remote user expects to perform HSI HPSS-to-HPSS transfers, the user must not initially select PSI aggregation to store to LANL HPSS.

Scripts will be provided at each site to simplify common data transfer paths, that is, paths to HPSS storage systems and Cielo. The same naming convention will be used: <*tool*>`2`<*site*> where <*tool*> can be `pftp`, `htar` or `hsi` and <*site*> is one of `lanl`, `llnl`, or `snl`. For example, the command `hsi2llnl` will initiate a compatible connection to HPSS at LLNL. Other destination <*tool*> and <*site*> scripts may be added at each site's discretion using the same naming convention. All tools will be present, without wrapping, to allow users to specify unique destinations.

Unclassified FTP between the tri-Lab sites is largely unsupported due to firewalls and other network restrictions. Secure Copy (SCP) can be used to copy files between hosts on a network. The Intersite HPC environment will also be used for file transfer among the Unclassified Cielo systems, when it becomes generally available.

The infrastructure support for high-speed bulk data transfers from Cielo includes jumbo frame 10 Gigabit Ethernet links on the FTAs, and some of HPSS storage movers. The Transfer Agent (TA) provides HPSS and host-to-host transfer performance and resilience. PFTP and HSI both have Transfer Agent (TA) capabilities. The TA is supplied as part of the HPSS software suite and is deployed at LANL and SNL.

### 3.1.1. LANL Site-Specific Data Transfer Configurations & Documentation

LANL-specific documentation can be found at http://hpss-info.lanl.gov/hpss

### 3.1.2. LLNL Site-Specific Data Transfer Configurations & Documentation

LLNL-specific documentation can be found at the following locations:
HPSS (including FTP/PFTP): `https://computing.llnl.gov/LCdocs/hpss/`

HSI: `https://computing.llnl.gov/LCdocs/hsi/`

HTAR `https://computing.llnl.gov/LCdocs/htar/`

### 3.1.3. SNL Site-Specific Data Transfer Configurations & Documentation

SNL-specific documentation can be found at
`https://computing.sandia.gov/data_management/recommended_toolset`

## 3.2. Staging Data to the Machine

1. *Shall provide system environment components that facilitate development of common procedures for setting up similar types of problems on the platforms. (e.g., scripts, common file movement facilities, etc.)*
2. *Shall provide a location where libraries and tools associated with an application can be maintained.*

Users have several options for data placement/staging: home directories, project space, or the parallel filesystems. Home directories and project space are both located on sequential filesystems based on NFS. These are intended for staging data, but not for running large parallel jobs. User home directories are large (16 GB per user), safe from purging, and backed up regularly. Additionally they provide an online backup directory (`.snapshot`), which allows file recovery without the need for contacting a system administrator or ACES User Support. Project directories can be even larger, and permit group and world sharing based upon Unix permissions. These provide a safe (backed up), permanent location for project source code, libraries, input decks, documentation, and any data related to the project.

The parallel filesystems are not to be used for long-term storage, nor to store valuable data that should be replicated and backed up. No portion of the parallel filesystem is backed up. The parallel file systems are entirely subject to periodic file purges.

Users are expected to transfer files between unclassified protected and classified secure environments using tools available at their home sites. This means, for example, that LLNL and SNL users may move files to Cielo only from a classified secure machine at LLNL or SNL, respectively.

## 3.3. Archival Storage Policies

1. *The archive implementation shall provide high reliability. Provide disaster recovery mechanisms for user identified files to attain even higher reliability. (moved from Section 7.4)*
2. *The archive implementation shall provide high availability. Provide failover for archive write.(moved from Section 7.4)*
3. *Provide archive data management and organizational services (to include audit capabilities), consistent with security policies (e.g., GREP, FIND, LS, Head, Tail, Bulk file management). (moved from Section 7.4)*

4. *The archive implementation shall provide data security. To include access control lists, portable across HPSS platforms/sites implementations. (moved from Section 7.4.)*
5. *The archive implementation shall provide the capacity sufficient to track user needs.*
6. *The archive implementations shall provide common functionalities. (moved from Section 7.4.)*
7. *The archive performance (bandwidth delivered to a single file archive) shall track user needs for archival storage as determined by problem size and expectations for time to archive. (e.g., a constant time to archive that tracks problem size). (moved from Section 7.4.)*
8. *Shall provide a common (across systems and tri-Lab) user interface to the archive. (moved from Section 7.4.)*
9. *Shall provide utilities to consolidate many small files into a single file. (currently htar). (moved from Section 7.4.)*
10. *Shall provide a persistent archiving capability with seamless transition across archival systems. (moved from Section 7.4.)*

HPSS is the standard for archival storage within the HPC community. It provides a scalable parallel storage system for massively parallel computers as well as traditional supercomputers and workstation clusters. The tri-Labs have been actively involved in the collaborative project to develop the High Performance Storage System (HPSS) for a number of years, with the ASC program's extreme archival requirements being the primary driver for the HPSS development effort.

Cielo will use HPSS to meet the high-end storage system and data management requirements of its users. All LANL production systems are connected via 10gigabit and gigabit Ethernet to an HPSS system.

Access to HPSS from SNL or LLNL over the classified network has been facilitated with tri-Lab common `pftp` scripts, `hsi/ta`, `htar`, and `psi`, described in Section 3.1.

HPSS security relies upon standard Unix security mechanisms.

While LANL does not currently charge a fee for archival storage, mechanisms are in place where this is possible should it become necessary at the direction of program management.

No purging of archival storage is planned.

## 3.4. Effective Use of the File Systems, Serial, and Parallel I/O

1. *Need an API to determine whether a file system is local or global, and serial vs. parallel. (moved from Section 4.5)*
2. *Shall provide guidance for code developers to make effective use of serial and parallel file systems, including the main file systems primarily to include but not limited to large parallel file system-topology. (moved from Section 4.5)*
3. *Shall provide platform specific single global parallel file system visible from all nodes with homogeneous node IO performance to the file system. (moved from Section 4.5)*
4. *Shall provide a single parallel file system that spans platforms. (moved from Section 4.5)*
5. *Shall provide links to information about the system parallel file system. (moved from Section 4.5)*

6. *Shall provide optimized and standard compliant parallel I/O libraries: MPI IO, HDF5, PNetCDF, and UDM. (moved from Section 4.5)*
7. *Shall allow for the number of concurrent open files (including system files) equal to approximately ten times the number of processes. (moved from Section 4.5)*
8. *Shall provide IO libraries that are thread safe with scalable performance. (moved from Section 4.5)*

Each user-accessible file system is designed for specific purposes and styles of I/O. The filesystems are listed in Table 6. The characteristics and intended usage of these file systems are extensively documented within the ACES Web pages so that users may easily comply with conventions and benefit from optimum usage.

Most user file systems are NFS mounted and suitable only for serial I/O and are low bandwidth compared to the Lustre parallel filesystem. Parallel I/O is specifically intended for the parallel file systems attached to a given compute platform. For Cielo systems, Lustre file systems are designated for all application parallel I/O.

### 3.4.1. Parallel I/O

At the heart of the Cielo I/O infrastructure are Lustre file systems, as implemented and supported by Cray, Inc. Communication to Lustre from Cielo occurs over a dedicated InfiniBand network. The InfiniBand network is used only to server Lustre to Cielo, Cielo's front-ends, and the File Transfer Agents.

Lustre file layouts can be controlled at the file level or at the directory level, so that all files created within a directory will have a certain file layout. Using MPI hints or ioctls allows the user to specify more file layout.

Lustre is a standards-based file system. All existing codes that compile to the POSIX standard, i.e., standard open, read, and write calls, will work with Lustre.

In general, transfers from a user application constructed as an aligned multiple of the underlying file system stripe unit size will provide better performance from the system.

Lustre is a parallel file system; to scale up to faster bandwidth, operations should be done in parallel. Using "`ls -l`" to gauge progress of jobs should be avoided. Long listing will take longer when jobs are writing or creating files in the same directory and will slow down the I/O rates for files in the directories being listed. The same slow behavior is caused by `ls` options or shell settings which color-code files by type. When in doubt, use `/bin/ls` to ensure that "`ls`" is not executing "`ls –l`".

Users should contact ACES User Support to help resolve problems using Lustre. More information about file system usage can be found at the ACES Web pages.

### 3.4.2. File System Support for Shared Libraries

The `/udsl/fs1` partition is tuned to provide best possible delivery of user-provided dynamic shared library access on the compute nodes. Files located in `/udsl/fs1` are writable from login nodes, but read-only from the compute nodes. The `/udsl/fs1` file system is specifically dedicated to support applications that need to load dynamically shared libraries and other read-only input files when running at scale. It will be managed in the same way as the `/usr/projects/` file system.

### 3.4.3. File System and I/O Libraries

Parallel I/O libraries supported on Cielo systems include Cray's thread-safe MPI implementation, HDF5 and NetCDF. Cray's MPI library includes all of MPI-1 and all of MPI-2 except dynamic tasks. Parallel I/O is fully supported by the MPI-IO interface and is optimized for use with the parallel file system. HDF5, like previous versions of the Hierarchical Data Format software from NCSA, is specifically designed for the efficient storage of large scientific datasets. This most recent version incorporates improvements and support for larger files, parallel I/O and threads. More information can be found at http://www.hdfgroup.org/HDF5/.

# 4. Application and System Code Development

## 4.1. Gaining Access to a Machine for Code Development

1. *Provide resources configured to facilitate code development (system and application.)*

The development environment for Cielo systems offers resources and support for both application and system code development. Resources comprise hardware, networks, storage, and software. Cielo developers have access to Cielo production and development machines for development, porting codes, debugging and making smaller production runs (all considered “routine” work) in preparation for capability computing (considered to be “priority” work). All users have access to the parallel file systems (discussed in Sections 2.1 and 3.4) and mass storage for their development work. The Cray Application Development Environment (CADE) comprises all the software for development, execution, debug, and analysis on the XT/XE6 (see http://docs.cray.com and links from ACES Web pages).

Applications and projects that require their own project-managed storage are provided with directories in the `/usr/projects/` file system. ‘Project’ directories are created by first submitting a request to ACES User Support. Each project must have an Owner. The Owner must understand programmatic need to know for access to the source code and binaries and input decks for the project. Access to project directories is controlled using normal Unix user and group permissions. This mechanism allows designated users to manage their own subdirectories. Application teams can use their project directories to install and update their codes. When projects are set up, a module directory will also be created so that the project team can manage their own module files. These modules are in addition to the required, vendor-provided modules.

In addition, users are permitted to create their own subdirectories in the `/usr/projects/packages/` directory, with full access permissions. Both the `/usr/projects/` and `/usr/projects/packages/` directories are subject to quotas and are backed up.

## 4.2. Peculiarities of the System

1. *Shall document deviations from POSIX and ANSI/ISO standards.*
2. *Shall document deviations from compliance with language and run-time standards. (e.g., MPI, I/O, Cray pointers, etc.)*
3. *Shall publish deviations from IEEE arithmetic.*
4. *Single location (per platform) (e.g., a web site.) with pointers to commonly known or encountered deviations from industry HPC standards–could be implemented via an updated FAQ as items are identified, via release notes, or a bulletin board*

From a user's perspective, ASC Cielo systems are very similar to the previous ASC system Red Storm. The Cray XT/XE6 features a Linux-based programming environment, parallel and sequential file systems, diskless compute nodes, MPI-based message passing, and the use of a lightweight operating system on compute nodes.

Service partition nodes are assigned to one of several different classes based on their hardware configuration and their intended use. Cielo also incorporates external login nodes, which are service nodes that are not directly connected to the high-speed mesh. They are used for compilation, job setup, and job submission. The presence of the external login nodes allows users to review files, perform job setup (not submission) and compile statically linked applications, even when the Cielo platform is unavailable (see Tables 5 and 6). (For performance reasons, `/udsl/fs1` is located directly on Cielo; it may not be available to users on external login nodes if the main system is unavailable.) External login nodes also provide disk-based swap space, unlike the diskless internal login nodes (see next section). Users log in to Cielo by requesting a specific external login node; a list of those nodes is available on the ACES Web pages. Thus, a user can ssh into one of several login nodes.

Within the compute partition there are a set of nodes configured with additional memory for problem setup, data analysis, and visualization. These nodes are "internal" to Cielo and will be dedicated to a particular visualization application. To support visualization, this sub-partition has dedicated visualization login nodes.

Network nodes feature high-speed connections to the FTAs. The I/O disk nodes are configured with high-speed disk connections, and are configured to run as parallel file system components. End users will not have direct access to the I/O disk nodes.

File Transfer Agents are also hosts external to Cielo, but having high-speed connectivity into the system. Users will log in to the FTAs to initiate file transfer operations to and from the Cielo platform.

**Table 5: Summary of User-accessible Nodes**

| Nodes | Connected to Mesh | Access |
|---|---|---|
| Compute Node | Yes | Allocated by scheduler |
| Internal Login Node | Yes | Allocated by interactive jobs |
| Viz Login Node | Yes | Allocated by scheduler; high-speed LAN connection |
| External Login Node | No | Direct user login |
| FTA | No | Direct user login; for data transfer |

**Table 6: Visibility of File Systems to Each Node Type**

| | `/users` | `/usr/projects` | `/udsl/fs1` | `Cielo Root` | `/lscratchN` | `Other LANL file systems` |
|---|---|---|---|---|---|---|
| **Type** | NFS | NFS | NFS | NFS | Lustre | Panasas |
| Compute Node | Yes | Yes | Yes, read-only | Yes, read-only | Yes | No |
| Internal Login Nodes | Yes | Yes | Yes | Yes | Yes | No |
| Viz Login Nodes | Yes | Yes | Yes | Yes | Yes | No |
| External Login Nodes | Yes | Yes | Yes | No | Yes | No |
| FTA | Yes | Yes | Yes | No | Yes | Yes |

### 4.2.1. Linux in the Cielo Environment

Service nodes on Cielo run the portion of the Cray Linux Environment (CLE) that is a diskless incarnation of SuSE Linux. Without a disk, processes on Cielo Linux node will not be able to use swap space other than that available as a portion of main memory. Thus, memory exhaustion is a real possibility. The various classes of Linux nodes have been configured with this limitation in mind. Other than the lack of a local disk, service nodes are essentially full, standard distributions of SuSE Linux. Service nodes, including login nodes, may be shared with other users.

Users can log into the Linux (service) nodes involved in the job, but not the compute nodes. In addition, TORQUE will report only the service nodes in the usual TORQUE node lists; the compute node assignments can be retrieved by running the utility `apstat` with the `–n` option.

The operating system is the Cray Linux Environment, which features both a lightweight compute-node kernel and a full-featured service-node Linux OS. The supported compilers are from PGI, GNU, and Cray. The Cray programming environment uses wrappers (`cc`, `CC`, `ftn`) for building codes that run on the compute nodes using MPI and OpenMP. The compiler suite must be invoked by selecting a Programming Environment module loaded through the use of the module facility.

The PGI Programming Environment is the default. Users must unload or swap modules to use other compiler environments.

For building codes on the service nodes, users may use the PGI, GNU, or Cray compilers directly. For vendor-provided training material refer to http://docs.cray.com/ and documentation on the ACES Web pages.

The service partition of Cielo should not be mistaken for a conventional Linux cluster. While the two architectures share many similar attributes, the Cielo environment is sufficiently different that porting will be required.

### 4.2.2. The Compute Node Linux (CNL) Lightweight Kernel

Compute Node Linux (CNL) runs on the compute nodes, and is provisioned to provide only those services required to run an application started by the `aprun` utility. This utility is somewhat similar to `mpirun`, but is designed for higher scalability. Running on an internal login service node, `aprun` transfers an application binary to the lowest numbered compute node assigned to the batch job. That node, which will be MPI rank 0, will fan out the executable binary to 32 more compute nodes assigned to the job. This process continues in a logarithmic fashion until all assigned compute nodes have the binary. At that point, the application begins running on the specified number of cores per node and is able to use the bulk of the node resources (e.g., CPU cycles, memory, and network chip).

CNL has been designed to minimize its memory footprint and need for CPU cycles. Very few of the traditional Linux daemons are available. For example, it is not possible for a user to log into a compute node. Nor is it possible to start two different application binaries. CNL does not support virtual memory. There is no disk drive to use as swap space should the application attempt to use more than the amount of physical memory on the node. Although scalability remains a concern, CNL on Cielo has been configured to support dynamically linked binaries.

### 4.2.3. Compute Node Linux (CNL) with Cluster Compatibility Mode (CCM)

In addition to CNL, there is a ‘Cluster Compatibility Mode (CCM)’ provided and supported by Cray. This mode adds back a number of the services that were present in the full-featured Linux, but not part of the base CNL. Cray created CCM in order to support Independent Software Vendor (ISV) codes that are provided only in binary form and that assume standard Linux cluster utilities, such as `mpirun` or `ssh`. CCM makes use of pre- and post-script invocation hooks provided with the batch resource manager (TORQUE), so the request to use CCM mode is tied to the job’s submission. CCM will likely support on the order of 1,000 nodes. It is not intended to scale to the full size of Cielo. CCM may be required in order to run some visualization applications on Cielo.

### 4.2.4. Page Size Control

Users may specify a page size for their application at run time using an option for `aprun,` in order to tune page size to their application. Incorrectly set page size can provide significant performance impact. Users are encouraged to consult the performance recommendations on the ACES Web pages for current information.

### 4.2.5. Multi-core Multi-Socket Compute Nodes

Two types of operations on multicore compute nodes—remote-NUMA-node memory references and process migration—can affect performance. On Cray XT/XE6 systems, processes accessing remote-NUMA-node memory can reduce performance. To restrict applications to local-NUMA-node memory, developers can use `aprun` memory affinity options. On Cray XE multicore systems, the compute node kernel can dynamically distribute workload by migrating processes and threads from one CPU to another. In some cases, this migration reduces performance. Developers can bind a process or thread to a particular CPU or a subset of CPUs by using `aprun` CPU affinity options.

The XE6 operating system distributes the memory available on a node across all of the cores in use. For example, on the standard compute node having 16 active CPUs and 32GB of memory, each of the CPUs can access 2GB of RAM. To run a job in which each process needs more memory, the `msub` resource `ppn` can be used to request fewer active CPUs per node.

In addition to these optimization options, the PGI, GNU, and Cray compilers provide compiler command options, directives, and pragmas that the developer can use to optimize code. For further information, see the compiler documentation links on the ACES Web pages. In addition, see the *Software Optimization Guide for AMD64 Processors* at http://www.amd.com/.

## 4.3. Parallel Programming Models and Run-Time Systems

1. *Shall provide a standard compliant MPI library optimized for both on-box and across-box messaging.*
2. *Shall provide POSIX compliant thread package.*
3. *Shall provide support for hybrid OpenMP/MPI parallel programs.*
4. *Shall provide documentation with guidelines for running OpenMP and MPI programs (especially for thread-safe code.) (e.g., how to tune buffer sizes).*

5. *Shall make it clear to users which MPI library is loaded, especially when wrappered compiler commands are used.*
6. *Shall provide resources for the installation and support of locally developed libraries that can be centrally accessed. (e.g., LA-MPI, UPS, HYPRE).*
7. *Need a mechanism to force a specific version of MPI to be linked into the code.*
8. *Need a standard interrupt propagation mechanism to tasks associated with a parallel application.*
9. *Consistent integration with the batch system and access to resource availability in order to build the MPI run command.*
10. *Provide mechanisms in MPI for enhanced reliabilit–e.g., survive NIC failures.*

The parallel programming models on the Cielo are:

1. Message passing using MPI. The `intro_mpi(3)` man page on the system provides a quick overview. It provides information on Cray MPICH2.
2. SHMEM programming model is a Cray-developed alternate messaging model using one-sided messaging and uses the Cray MPICH2 routines. The `intro_shmem(3)` man page provides a quick overview.
3. OpenMP shared memory programming model is industry-standard shared memory programming model; information can be obtained from http://www.openmp.org. On Cielo OpenMP applications can be used in hybrid MPI/OpenMP applications.

## 4.4. Third Party Libraries and Utilities

1. *Shall manage changes to center supported third party libraries via a change control procedure.*
2. *Shall provide information about different versions of libraries that are on the system and guidance to the user as to which one to use, including but not limited to MPI.*
3. *Shall provide system documentation that includes a POC for each center supported third party product on the system.*

Third party supported libraries and utilities are managed and supported by the ACES support staff. Information about updates and versions are posted on the ACES Web pages, documented in news items, and sent out by email to specific mail lists via ACES support staff. When new versions of existing libraries and software are installed on Cielo or Cielito, notices will be sent to users that these new versions are available. After a fixed time period during which users have become accustomed to the new version, default module settings will be changed to use the new version. Modules are used to set appropriate paths and other environment variables necessary to use these libraries.

The ACES User Support staff acts as the POC for third party products. They will be the first line of support and can answer most questions. In the event of further escalation of the problem, they will contact the appropriate staff member. Official problem reporting mechanisms are in place to report vendor library and software issues.

### 4.4.1. Math Libraries (including solvers)

1. *Shall provide support for standard sparse matrix operations.*
2. *Shall provide support for parallel sparse linear solvers.*

3. *Shall provide optimized BLAS and LAPACK.*
4. *Shall provide system documentation that points to what solvers are installed and the support POC.*

Math libraries on Cielo include Cray LibSci, PETSc, ACML, FFTW. Brief descriptions follow:

Cray XT/XE LibSci scientific libraries contain:

- Basic Linear Algebra Subroutines (BLAS) and LAPACK linear algebra routines
- ScaLAPACK and Basic Linear Algebra Communication Subprograms (BLACS) routines
- Iterative Refinement Toolkit (IRT), a library of factorization routines, solvers, and tools that can be used to solve systems of linear equations more efficiently than the full-precision solvers in Cray XT/XE-LibSci or ACML.
- SuperLU, a set of routines that solve large, sparse, nonsymmetric systems of linear equations. XT/XE-LibSci library routines are written in C but can be called from Fortran, C, or C++ programs.
- CRay Adaptive Fast Fourier Transform (CRAFFT), a library of Fortran subroutines that compute the discrete Fourier transform in one, two, or three dimensions; of arbitrary input size; and of both real and complex data.
- CRAFFT provides a simplified interface to FFT and allows the FFT library itself to choose the fastest FFT kernel.
- Portable, Extensible Toolkit for Scientific Computation (PETSc), an open source library of sparse solvers.

AMD Core Math Library (ACML), which includes:

- A suite of Fast Fourier Transform (FFT) routines for single-precision, double-precision, single-precision complex, and double-precision complex data types.
- Fast scalar, vector, and array math transcendental library routines optimized for high performance.
- A comprehensive random number generator suite. S –2423 –22 31

FFTW is a C subroutine library with Fortran interfaces for computing the discrete Fourier transform in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, such as the discrete cosine/sine transforms). The Fast Fourier Transform (FFT) algorithm is applied for many problem sizes. Distributed memory parallel FFTs are available only in FFTW 2.1.5. For further information, see the `intro_fftw2(3)` and `intro_fftw3(3)` man pages.

Fast_mv is a library of high-performance math intrinsic functions. The functions can be used in PGI and Cray compiled applications. For further information, see the `intro_fast_mv(3)` man page.

### 4.4.2. Networking and Other Libraries

1. *Shall support libraries that are callable from C++. (Note: If important please supply which libraries.)*
2. *Shall support secure interprocess communications such as sockets, HTTPS, and/or ssh.*
3. *Shall support the deployment of client/server socket-based parallel tools (including visualization) on platforms.*

In addition to the math libraries discussed above, Cielo supports a variety of other libraries common to the C/C++/Fortran HPC environment. These libraries comprise a wide range of functionality. For example, CLE system libraries are used for networking and communications, security, system management, X11, Java, national language support, OpenGL, diagnostics, system performance and tuning, and user management. Third-party libraries include programming (MPI, Pthreads, Perl, Tcl/Tk, Python, shell scripting, etc.) and a range of tools used for profiling, memory analysis, debugging, performance analysis and tuning, and visualization. Additionally, application libraries developed locally by various tri-Lab CCE groups are available on the Cielo systems. Not all of these features are available on compute nodes if they require capabilities beyond Compute Node Linux or Compute Node Linux with Cluster Compatibility Mode (see Sections 4.2.2 and 4.2.3).

### 4.4.3. **Compilation**

1. *The ACE system information shall provide current documentation for compilers, libraries and development tools. (moved from Section 4.2)*
2. *Shall provide user recommendations for recommended compiler options.*
3. *Shall provide the latest 64-bit versions of the gnu utilities. (e.g., make, g++, gawk, gdb, gawk, etc.).*
4. *Shall provide standard compliant compilers. C, C++, and FORTRAN9x.*
5. *Shall provide parallel make and compilation in build.*
6. *Shall provide information about different compiler versions resident on the system and guidance to the user as to which one to use.*
7. *Shall provide compiler support for OpenMP. C & OpenMP, C++ & OpenMP, and Fortran9X & OpenMP.*
8. *Shall provide compiler support for F77 codes.*
9. *Shall provide JAVA support. (editorial note  this is an attempt to clarify a previous requirement): JAVA  via standard JDK toolkit.*
10. *Shall provide common scripting languages. (e.g., perl, python, and Tcl/Tk, sh, csh, tcsh, ksh, expect)*
11. *Shall provide support for dynamic/shared libraries.*

The Cray XT/XE system Programming Environment includes Cray compilers and compiler suites from The Portland Group (PGI) and the GNU Compiler Collection (GCC). The compilers translate C, C++, and Fortran source programs into Cray XT/XE object files. Developers can use interlanguage communication functions to create Fortran programs that call C or C++ routines and C or C++ programs that call Fortran routines. The command used to invoke a compiler, called a *compilation driver*, 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. In addition to the Cray, PGI, and GCC compilers, the Cray XT/XE Programming Environment includes the Java compiler for developing applications to run on service nodes only. For details, see http://java.sun.com/javase/6/docs/. The Cray wrappers for C, C++, Fortran 90 are `cc`, `CC`, and `ftn` respectively. Selection of an appropriate Programming Environment to use PGI, GNU, or other compilers that are called from within the Cray provided `cc`, `CC`, and `ftn` wrappers are enabled by loading the appropriate module.

## 4.5. Debugging and Correctness Testing

1. *Shall provide the ASC standard parallel debugger. (currently TotalView).*
2. *Shall provide a usable debugging capability for jobs that span across 1/3 of the system.*
3. *Shall provide common tools for parallel aware memory checking including user callable routines to track memory use and availability. (e.g., Valgrind type functionality)*
4. *Shall provide machine resources to be used for code debugging – (this requires dedicated interactive use of the resource - and make it clear to users which resources are to be used for this purpose).*
5. *Shall provide tools for static analysis of source. (e.g., flint)*
6. *Shall provide tools that measure testing code coverage for all supported standard languages.*

The Cray XT/XE system supports the TotalView debugger for single-process and multiprocess debugging and the GNU `lgdb` debugger for single-process applications.

The TotalView debugger, available from TotalView Technologies, LLC, provides source-level debugging of applications. It is compatible with the PGI, GNU, Cray, and other compilers. TotalView can debug applications running on 1 to 4096 compute nodes, providing visibility and control into each process individually or by groups. It also supports access to MPI-specific data, such as the message queues. TotalView typically is run interactively.

To debug a program using TotalView, the developer invokes TotalView either from the graphical interface (`totalview`) or the command line (`totalviewcli`). TotalView parses the command to get the number of nodes requested, then makes a node allocation request. TotalView directs `aprun` to load but not start the application. The `aprun` utility 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 using TotalView (including other ways to interact with a job), see the ACES Web pages, the *Cray XT/XE Programming Environment User's Guide*, the `totalview(1)` man page, and TotalView documentation at http://www.totalviewtech.com/Documentation/. To find out what levels of the compilers TotalView supports, see the *TotalView Platforms and System Requirements* document at the TotalView Website.

For more information about `lgdb`, see the *Cray XT/XE Programming Environment User's Guide* `lgdb(1)` man page.

Cray provides a facility called Abnormal Termination Processing, or ATP. ATP monitors a running application and should it take a system trap, ATP provides an analysis of the stack traces from each process in the job prior to termination. See the `atp(1)` man page for usage instructions and limitations.

## 4.6. Performance Measurement, Analysis and Tuning.

1. *Shall provide a message tracing tool that is usable for jobs that span across 1/3 of the system. (e.g., Vampir).*
2. *Shall provide a message profiling interface (e.g., mpiP).*
3. *Shall provide loop level application profiling tools.*
4. *Shall provide tools to analyze cache performance.*

5. *Shall provide user level access to hardware performance counters, preferably via the PAPI API.*
6. *Shall provide lightweight high-resolution timers.*
7. *Shall make it clear which tools can be used together. (e.g., compiler and memory tool, or debugger and mpi).*
8. *Shall provide tools to access thread performance.*
9. *Shall provide tools to characterize the IO performance of a code.*
10. *Shall document usage of environment variables and other settings that optimize application performance (NEW)*

The Cray XT/XE system provides tools for collecting, analyzing, and displaying performance data.

### 4.6.1. **Performance API**

The Performance API (PAPI) from the University of Tennessee and Oak Ridge National Laboratory is a standard interface for accessing hardware performance counters. A PAPI event set maps AMD Opteron processor hardware counters to a list of events, such as Level 1 data cache misses, data translation lookaside buffer (TLB) misses, and cycles stalled waiting for memory accesses. Developers can use the API to collect data on those events.

### 4.6.2. **CrayPat**

CrayPat is a performance analysis tool. A developer can use CrayPat to perform sampling and tracing experiments on an instrumented application and analyze the results of those experiments. Sampling experiments capture information at user-defined time intervals or when a predetermined event occurs, such as the overflow of a user-specified hardware performance counter. Tracing experiments capture information related to both predefined and user-defined function entry points, such as the number of times a particular MPI function is called and the amount of time the program spends performing that function.

The developer uses the `pat_build` command to instrument programs. No recompilation is required to produce the instrumented program. Alternatively, the developer can use the `pat_hwpc` command to instrument the program for collecting predefined hardware performance counter information, such as cache usage data. After instrumenting a program, the developer sets environment variables to control run time data collection, runs the instrumented program, then uses the `pat_report` command to either generate a report or export the data for use in Cray Apprentice2 or other applications.

### 4.6.3. **Cray Apprentice2**

Cray Apprentice2 is an interactive X Window System tool for displaying performance analysis data captured during program execution. 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:

- It is a postexecution performance analysis tool that provides information about a program by examining data files that were created during program execution. It is not a debugger or a simulator.
- Cray Apprentice2 displays many types of performance data contingent on the data that was captured 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.

## 4.7. Configuration Control

1. *ASC platform status and configuration information shall be maintained and accessible (both push and pull model) to users. This information includes the current, historical, and planned (1) operating system and system libraries, (2) compiler and embedded, (3) MPI library, (4) other "standard and customary" libraries, and (5) system status e.g., maintenance down time, dedicated mode, etc. Attributes shall identify the entity including name, location, version, and patches; shall include statement regarding backward compatibility with previous versions; shall include the dates planned for delivery, actually validated, planned for withdrawal, and actually withdrawn.*
2. *Shall provide access to source repository (e.g., SourceForge, Razor, CVS, etc.)*
3. *Shall provide the capability for developers and support personnel to maintain products without requiring sys admin privileges.*
4. *Verify the environment by running a set of application based regression tests to exercise changes to the operating system, compilers, libraries, etc. (moved from Section 1.1)*

### 4.7.1. Change Management

The configuration for each ACES/Cielo system will be managed by a change control board (CCB). The CCB will be staffed by the members of the ACES partnership and will approve all hardware and software updates applied to the platforms. The Information System Security Officer (ISSO) for each Cielo classified platform will be a member of the Change Control Board.

This approach will ensure that users are provided with a consistent operating environment. Changes will be announced in advance via the ACES Web pages and other user communication mechanisms. All changes to the classified systems will be managed in accordance with the system security plan. Regression tests will automatically be applied to any software configuration changes applied by ACES administrators. Configuration control for software installed by ACES staff also includes regression testing. When new software is installed, such as a new Fortran compiler or MPI library, support staff will test the new software against a suite of codes designated for that purpose. Any variances found during testing will be posted appropriately on the ACES Web pages.

For major hardware/software installations, the ACES administrators run a series of regression tests that include user applications and benchmarking codes. These are run as a part of the synthetic workload (SWL) suite for the acceptance of the machine. A review team determines whether the results from the regression tests are acceptable for user access and general availability (GA).

### 4.7.2. Source Code Control

ACES users and support staff have the standard Unix environment configuration control utilities available on all production systems, including Cielo platforms. These include RCS and CVS. Documentation for these configuration control systems is readily available in man pages and on the Web. All of these configuration control systems provide the capability for code developers and support staff to maintain their software without the need for system administrators or any special administrative privileges.

## 4.8. Best Practices

1. *Shall publish "best practices" for performance.*

The ACES team is in the process of developing a "best practices" document for Cielo. This document will be located on the ACES policies Web page `http://aces.sandia.gov/policies` as well as in the `/usr/local/docs` directory. In the interim, users may benefit from various recommendations that appear throughout the ACES Web pages and in documents located in the `/usr/local/docs` directory.

# 5. Problem Setup

1. *Shall provide resources and connectivity to users for the generation of meshes/netlists/schematics, material databases, and necessary data storage for application data bases.*
2. *Shall provide real-time information to support smart domain decomposition to allow near optimal mapping onto the machine, taking into account nonhomogeneous system resources.*

Problem setup includes generating meshes, domain decomposition, materials databases and constitutive models, model/assembly management, visualization of meshes, and archiving of solid models. The user environment provides system resources and network connectivity for the generation of meshes/netlists/ schematics, domain decomposition, material databases, and application databases.

Application setup tools (e.g., Cubit, GMV) are provided and supported by specific project teams. These applications and libraries will likely be found in `/usr/projects/` or `/usr/projects/packages`.

## 5.1. Mesh generation

Many of the SNL analysis/simulation codes read and write one or more files per process, so there is a requirement that the Problem Solving Environment support mesh generation tools (Cubit) and engineering analysis utilities (SEACAS). The SEACAS tools are a collection of structural and thermal codes, data format and I/O libraries, and utilities used by analysts at SNL. The system includes pre- and postprocessing codes, analysis codes, database translation codes, support libraries, UNIX shell scripts, and an installation system. The support tools for data pre- and postprocessing and data manipulation include Algebra, Aprepro, Blot, Gen3D, Grepos, Grope, Mapvar, nem_slice, nem_spread, and epu (nem_join), to name a few. Support libraries include Chaco, ExodusII data model, and NetCDF I/O library.

The CUBIT mesh generation environment is a two- and three-dimensional finite element mesh generation tool for the generation of quadrilateral and hexahedral elements. It is a solid-modeler based preprocessor that meshes volume and surface solid models for finite element analysis**.** Mesh output from CUBIT is ordinarily fed to SEACAS domain decomposition tools for task assignment to individual processors.

Cielo does not provide large-memory system resources. Users who need large-memory platforms for mesh generation, domain decomposition, or any other similar purpose will need to perform those activities on other platforms and transfer the resulting data.

## 5.2. Domain Decomposition

Prior to running a simulation on a parallel computer, the simulation must first be decomposed into tasks assigned to different processors. Efficient use of the machine requires that each processor have approximately the same amount of work to accomplish and that the quantity of interprocessor communication remains small. Finding an optimal decomposition is very difficult, but due to its practical importance, a great deal of effort has been devoted at SNL to developing heuristics for this problem. A variety of algorithms for graph partitioning have been researched, developed, and implemented, and they include Chaco, Zoltan, and Metis. SNL specific codes using these libraries include `nem_slice`, `nem_spread`, and `yada`.

# 6. Running the Application to Solve the Problem

Cielo is designated as a capability resource. Allocation and scheduling of the stockpile stewardship workload on Cielo is defined by the Capability Compute System Scheduling Governance Model through the CCC process.

In 3Q06, the tri-Labs agreed upon and accepted a common batch scheduling system, Moab, from Adaptive Computing, Inc. This scheduler fulfills the goal of a common resource manager across the tri-Labs. The Moab license contract grants ASC use of Moab software, which provides workload management, system accounting, capacity planning, automated failure recovery, virtualization, and a host of other capabilities in cluster, grid, and utility computing environments. In addition, the contract also includes collaborative research and development, and consulting.

The Moab solution adds significant manageability and optimization to HPC resources, while providing a common interface to users throughout the tri-Lab complex (LLNL, SNL, and LANL).

As with the application development environment, the ASC Cielo application run-time environment is consistent with other LANL production systems. Users will find the expected file systems, environment variables, shells, compilers, tools, utilities, software and documentation, all discussed previously in this document. Similarly, Cielo systems employ Moab to manage batch and interactive jobs.

## 6.1. Submitting the Job (Local and Remote)

1. *Shall provide users with a common queue syntax and semantics across the tri-Lab. (i.e., "large queue" means the same across the tri-Labs) (Note: To the extent that a common queue syntax and semantics for the tri-Labs exist and are documented.)*
2. *Shall provide a common user interface to access different underlying resource management system resource information. (Note: To the extent that a common user interface for the tri-Lab exists and is documented. To achieve tri-Lab common interface – may require change to your scripts.)*
3. *Shall provide the user the ability to specify the approximate requirements of their job. (e.g., the ability to specify the resources that are needed for the job.)*
4. *Shall provide the user with a sorted list of resources available to handle their job. (e.g., How many processors are available, In which queue, Estimate of time to start-up, Estimate of run time)*
5. *Shall make the remote job submittal process the same across the complex. (Note: To the extent that a common job submittal process exists and is documented.)*
6. *Shall provide resource management mechanisms that support accurate and reasonable time limit and node allocation requirements.*
7. *Shall support dependent jobs.*
8. *Shall provide automatic job submission e.g., for regression testing. (e.g., cron)*
9. *Shall provide job schedulers that place processes so as to optimize use of the platform topology or specialized resources. (e.g., for compilation, I/O, visualization, etc)*
10. *Need quick turnaround for small regression tests.*

Moab is the job scheduler on Cielo. LANL Cielo users will find Moab to be similar to the system they are familiar with on LANL systems. More information on Moab, including current usage and policies, can be found on the ACES Web pages and at http://www.adaptivecomputing.com/resources/docs/mwm/moabusers.php.

Parallel job launch on Cielo differs from a conventional Linux cluster because the nodes involved must be allocated from different pools. Every Moab job requires at least one service node. A service node will be designated by TORQUE to run the job script process (`aprun`), which launches executables onto compute nodes and manages the parallel job.

ALPS is the allocator provided by Cray for the compute nodes. It works in concert with Moab and TORQUE to provide batch job submission and job script launch facilities. Commands within ALPS for running (`aprun`), status enquiry (`apstat`) and termination (`apkill`) of jobs are basic commands with which a user should become familiar.

Visualization nodes (compute and service) are allocated through Moab by using a special Moab class (queue). Both batch and interactive viz jobs are supported.

The job submission process begins with a user-constructed job script that contains keywords and parameters, shell script commands and, optionally, comments. The user job script is then "submitted" to the Moab system by a simple Moab command called `msub`, usually entered on the shell command prompt after logging into one of the system's external login nodes. Also, `msub` commands may be issued from within shell scripts or cron jobs.

Parameters associated with the job, such as how many nodes/processors to use, how long to run, where to send output, are specified by the user in the job command script (or as arguments to the `msub` command) and enforced by Moab. In routine usage, Moab will not allocate more nodes or a greater run-time than permitted by the defined Moab QOS (Quality of Service). QOSs are established by ACES management in conjunction with ASC program managers and users. QOS limits are subject to change and are documented within the ACES Web pages. They may also be displayed with a simple command after logging in. For example "`mdiag  –q`" will display the QOS information for Cielo.

A fair-share model is used for Cielo. Scheduling is dynamic and may change at any time before a job actually starts to run, depending upon the priorities of any other jobs that may be submitted before execution begins. For this reason, it is difficult, if not impossible, to predict exactly when a normal user job will run. Users are provided with a limited amount of control in determining when a job should run, however. They can specify that a job should not run until another specified job has completed (dependency), or specify a date/time before which the job should not run.

Job turnaround time is a function of several interrelated factors, including job priority, resources requested (number of nodes, length of time), and system workload. Furthermore, because Cielo is a capability platform, large jobs are the norm. In practice however, small short jobs are often able to be "backfilled" by Moab into suitable node/time slots for improved turnaround.

Jobs submitted using the Standby allocations are free to run unless they are preempted by a CCC calculation. Some opportunity will be given to a Standby job to checkpoint itself; however, the intent is to minimize the impact on CCC teams seeking immediate access to capability cycles. Utilization by Standby jobs will not be counted as part of any approved CCC.

## 6.2. Monitoring Job Status

1. *Shall allow users to examine the contents of files being used by an executing job. (with STDIN and STDOUT caveats)*
2. *Shall be able to attach to a running job to control, steer and monitor it. (e.g., run proxy.) (Note: Users will be able to attach to a running job to steer and monitor with TotalView.)*
3. *Shall be able to monitor memory and cpu use for load balancing purposes.*

Users have several ways to monitor job status. Moab provides the `checkjob` command. By default `checkjob` displays information on a specific job. This information includes submit time, wallclock time, and nodes used. The `showq` command can also be run to see the eligibility of submitted jobs. These commands are issued interactively. Cray provides the `xtnodestat` command to see the active jobs on the mesh.

By default, Moab handles `stderr` and `stdout` by writing each to a unique file, the name of which is typically a concatenation of the job script name and the Moab assigned job id number. However, users can select their own file names and also have both `stdout` and `stderr` go to the same output file. Moab output files are created while the job runs, and are readable immediately by the user.

## 6.3. Stopping the Job

1. *Shall provide a well-defined signal interface to be sent to all user processes prior to the system going down. The signal shall be sent out with sufficient notice to allow graceful shutdown and cleanup.*
2. *Shall make it possible for the user to inform the resource management system of the appropriate time delay between signal issue and process termination for their application.*
3. *Shall allow a user to initiate a kill of a job causing an immediate termination, to include an optional flushing of IO buffers - this capability needs to work in the presence of resource managers. (e.g., the signal propagates through the various layers of daemons to the user processes).*

Moab permits users to terminate queued and running jobs with the “`mjobctl -c`” command. For a queued job, the effect of the “`mjobctl –c`” command is to remove the job from the waiting queue; for a running job, “`mjobctl –c`” will terminate the job first and then remove it from the running queue. Users may also put a nonrunning queued job on hold for an indefinite period by using the “`mjobctl –m`” command.

## 6.4. Interactive Use

1. *Shall provide some portion of the system to be available to support timely interactive use.*
2. *Shall provide dedicated resource allocation for interactive use. (e.g., for computational steering or debugging)*

Users can submit interactive jobs by using the Moab command “`msub –I`”. This allows a user to access one or more compute nodes and debug interactively instead of launching a batch job and waiting for the results.

On Cielo, when an interactive job is launched, the user is logged in to a service node via ssh. From this node, the user can launch parallel job, with sizes up to the maximum number of processors requested in the interactive job, via `aprun`. Note that in this case, the user is not logged into a compute node, but into a service node that manages the compute node allocation for her job.

## 6.5. Adapting the Job for Expected System Reliability

1. *Shall provide a history of machine reliability to assist users in determining frequency of restart dumps.*
2. *Shall provide users with information about planned system down time.*
3. *Shall provide procedures for "automatic" job restart after system interruption.*

For scheduled maintenance and designated system time (DST), Cielo users are notified in a variety of ways as discussed earlier. This allows users to plan how they might submit and monitor jobs to avoid unnecessary impacts from planned system unavailability. Unplanned and emergency downtimes are similarly announced with as much advance notice as possible. Nevertheless, experienced users who are concerned about losing results and/or progress for longer running jobs routinely checkpoint their applications for later restart in the event of unexpected service outages. Building checkpoint/restart capability into an application is the responsibility of the developer, and differs because of this.

In addition to checkpointing, users are encouraged to backup essential data, as many file systems, including the large parallel file systems, are not backed up. Unexpected network, disk hardware or other problems have been known to result in permanent loss of user data, something which is easy to prevent by ensuring data is backed up to the archival system or copied to a file system that is routinely backed up

## 6.6. System Reliability

1. *Shall provide tools and systems to provide a reliable environment for users. (e.g., lock out unreliable nodes or nodes with other problems)*
2. *Shall provide the capability to transparently migrate user processes if a node goes down.*

See the discussion above in Section 6.5. Additionally, system administrators are provided with tools, from vendors and locally developed, to help maintain a robust and reliable computing system. 24x7 operations support also helps ensure prompt trouble response and system reliability.

# 7. Processing Simulation Output

1. *Shall provide specialized resources (i.e., software and platforms) for the manipulation (e.g., combining) of visualization data - may require more memory than compute nodes, etc.*

In support of the diverse capability users within the tri-Lab community, Cielo supports several applications and their associated usage models for the analysis and postprocessing of simulation results. The postprocessing applications of interest include EnSight, ParaView, and VisIt. Each of these has undergone extensive development throughout the course of the ASC program, and is widely used within the computational science community. Additionally, each is capable of scalable parallel operations, which is requisite for efficient analysis of data at the scale produced by petascale-class systems.

Final processing of simulation results depends heavily on all aspects of system operation and performance. The interactive nature of most postprocessing activities, coupled with the needed integration of many HPC components during a postprocessing session, necessitates stability and overall usability of the entire platform. Visualization uses all of the Cielo and network infrastructure. The most relevant portions include the resource allocation system, large-memory viz nodes, the parallel file system, the system interconnect, the tri-Lab WAN (DisCom), and the data archival system.

## 7.1. Specific Visualization Models

The tri-Labs employ three overarching models for effective visualization of petascale datasets: image transfer, geometry transfer, and data transfer. Cielo fully supports each model. In fact, all three models are supported by the three software packages. It is important to understand these models, as they ultimately dictate the underlying requirements and mechanisms for post-processing and analysis on Cielo. The two primary components of distributed visualization, pertinent to each model, are the I/O-intensive geometry extraction process, and the compute-intensive graphics rendering process.

### 7.1.1. Image Transfer Model

The *image transfer model* retains all geometry extraction and rendering operations on the hosting compute platform (Cielo). All images are subsequently generated compute-side through distributed software rendering. The resultant images are then transferred, via network, to an interactive client operating on a remote platform with direct video, and keyboard and mouse control.

Beneficially, this model requires no movement of simulation results, and requires no special compute hardware at the client side. However, significant portions of the compute platform must be devoted to the rendering process, and issues relating to the bandwidth and latency of the server-to-client network connection can negatively impact interactive data analysis.

The image transfer model is usually employed by SNL, using ParaView software, and by LLNL, using VisIt software.

### 7.1.2. Geometry Transfer Model

The *geometry transfer model* separates the data extraction component from the graphics rendering component by moving the rendering process to the client-side workstation (or to a purpose-built visualization platform).

Client-side rendering allows the analysis application to make extensive use of special purpose rendering hardware to significantly accelerate the rendering process. Thus, fewer compute resources are required on the hosting compute platform for simulation postprocessing. This model allows for adequate interactive analysis, while requiring no movement of simulation results. However, it is important that users have a solid understanding of the underlying graphics mechanisms, so that server-to-client transfer of unreasonably large geometries is avoided.

The geometry transfer model is the usual model for LANL, using EnSight software.

### 7.1.3. Data Transfer Model

The *data transfer model* moves all postprocessing operations off of the hosting compute platform, by transferring all simulation data, in their entirety, to a purpose-built analysis and visualization platform.

This model allows the hosting compute platform to remain entirely devoted to simulation, but will become prohibitive as the growth of simulation results is fast outpacing WAN data transfer performance. Any such transfer must make use of the DisCom WAN and the various data transfer tools available on Cielo. For further details, refer to Section 3.

The image transfer model is deprecated, and should not be used as a usual means of visualizing data. Large-scale data movement may not in fact be supported in the petascale environment.

## 7.2. Resource Allocation

### 7.2.1. Dedicated Visualization Resources

Both the image and geometry transfer models require an allocation of compute nodes on Cielo. Cielo provides dedicated, full-up nodes for analysis activities. Access to these nodes is acquired via the Moab scheduling and allocation system, by specifying the appropriate class (or queue) during job submission. Instructions on using dedicated visualization resources, as well as CCM for EnSight, can be found on the ACES Web pages.

### 7.2.2. General Compute Resources

Both the image and geometry transfer models require an allocation of compute nodes on Cielo. All compute nodes on Cielo are intended to be usable for either simulation or analysis activities. The allocation of full-up compute nodes for analysis purposes is provided by the Moab scheduling and allocation system. Users may specify the appropriate queue during job submission, which is specially configured to provide immediate access to compute resources for interactive analysis sessions.

## 7.3. Analysis

1. *Shall document what viz tools are available.*
2. *Shall establish policies for utilization of compute platform visualization nodes.*
3. *Shall provide the ability to visualize the data in place without the need to move data to another system.*

4. *Shall provide computing (hardware) resources to be used for data extraction and visualization as part of an interactive visualization process that does not require the movement of data.*
5. *Shall provide a full featured scalable parallel visualization tool. This requirement currently met by Ensight and VisIt.*
6. *Shall provide data analysis capabilities. IDL is the preferred tool due to a link to experimentalists' data.*
7. *Shall provide visualization from the desktop.*

Cielo supports the operation of EnSight, ParaView, and VisIt for scalable, parallel results analysis. Each of these tools provides a modern suite of graphical representation methods for scientific data analysis, including surface extraction, contour generation, and volume rendering. Each is also capable of providing an interactive analysis session, without movement of simulation data, using a client/server architecture. The particular details involved in configuring and using each of these can be found in the plethora of public and commercially available documentation, with information and links available on the ACES Web pages. In each case, it is a requirement that the server component of the analysis tool be operated on one or more Cielo compute nodes, which have been acquired via the resource allocation system. Analysis activities on the external login nodes, FTAs, and other noncompute hardware are not supported.

Cielo additionally supports the operation of Interactive Data Language (IDL) for analysis activities. Details regarding operation and use of IDL are left to the user.

## 7.4. **Sharing Results**

1. *Shall provide a command to transfer a file to another user. Must be a standard interface and a standard location. (e.g., "give" command)*

The LANL `give` tool is available both on Cielo and on LANL HPSS for quick transfer of files to other users.

## 7.5. **Archiving Results**

Simulation and analysis results may be archived to LANL HPSS, or transferred to SNL or LLNL for subsequent storage. Refer to Section 3 for further details.

# 8. ACES Coordinated Operational Support

User support for Cielo accommodates the needs of both local and remote users, and is provided jointly by LANL and SNL. This includes integrated telephone support, timely dissemination of operational information (e.g., scheduled and unscheduled machine or WAN downtime), training and documentation that meets the needs of local and remote users, as well as the availability of common ACES project Web pages with consistent format and information.

## 8.1. User Support

1. *Shall publish and maintain phone numbers, email addresses, and hours of support hotline and help desk.*
2. *Shall provide contact information for reporting problems outside working hours.*
3. *Shall provide adequate local support staff that is available during tri-lab working hours to answer user questions either in person or via the telephone (single access number/site.)*
4. *Shall have the capability for open and secure problem reporting and tracking for each system.*
5. *In the ACE, the local user support team shall act as a clearinghouse for local user support related to both local and remote platform usage.*

User support for Cielo is handled by the ACES helpdesk, provided jointly by LANL and SNL HPC personnel. Users may access support any of four ways: via the Web, electronic mail, telephone, and walk-in. The ACES Web pages provide the necessary contact information and instructions on how to obtain support via all of these methods, such as email addresses, phone numbers, support hours, and locations.

Web support includes all of the resources described in Section 1.2. It also includes the ability to enter a “trouble ticket” at any time, by means of the ACES Web-based tracking system. Information about the trouble ticket system can be found at the ACES Web pages. The trouble ticket system is available for problems in both the unclassified and classified networks.

Telephone assistance can be accessed via a toll-free number, 877-378-1110. LANL users can also call the local number 665-4444 Option 3. User assistance is available from 8:00 AM to 5:00 PM Mountain Time, Monday through Friday. Users may also visit the Sandia or Los Alamos HPC helpdesk personnel during regular business hours; some users may require escort to helpdesk locations, particularly at LANL. All requests are addressed by a team of HPC consultants or account specialists. Additional services, such as account requests, training, sample codes, code porting, application tutorials, can be found at the ACES Web page.

For after-hours and weekend support issues, onsite operators, system administrators and network staff provide basic on-call user support services.

## 8.2. Trouble shooting (in-depth consulting)

1. *Shall provide experts to be available to resolve complex system problems in areas such as IO, compilers, visualization, parallel programming, performance, debugging, platform specific issues, etc.*
2. *Shall provide enhanced system support for early machine problems and milepost calculations.*

The ACES helpdesk provides user support for Cielo. However, difficult and complex problems that require expert-level troubleshooting arise on a regular basis, requiring one-on-one assistance from time to time. For these type of support issues, LANL and SNL have available a number of other HPC and vendor-supplied staff who are experts in various areas, such as compilers, tools, I/O, visualization, debuggers, hardware issues, networks, batch systems. These same experts are also typically involved in supporting early systems and milepost work.

In most of these cases, the relevant expert is assigned a related ticket and communicates a solution back to the user through the ACES helpdesk. In some cases, the expert may work directly with the user and/or the problem until it can be resolved or a workaround solution found. If necessary, the expert will work with vendors or other relevant third parties in an attempt to facilitate their ownership and resolution of the problem. Because most in-depth consulting originates from a trouble ticket, it can be tracked by the usual means through the ACES trouble ticket system.