SANDIA REPORT

SAND2006-2729
Unlimited Release
Printed November 2006

ASC Computational Environment (ACE) Requirements Version 8.0 Final Report

Alex Larzelere and Judy E. Sturtevant

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2006-2729
Unlimited Release
Printed November 2006

ASC Computational Environment (ACE) Requirements Version 8.0 Final Report

Alex Larzelere (ed.)
Exagrid Engineering
6511 Wynema Ct., Suite 180
Alexandria, VA 22315
and
Judy E. Sturtevant (ed.)
Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185-0807

With Input and Review from
Tri-Lab S&CS Program Element Management Team

Abstract

A decision was made early in the Tri-Lab Usage Model process, that the collection of the user requirements be separated from the document describing capabilities of the user environment. The purpose in developing the requirements as a separate document was to allow the requirements to take on a higher-level view of user requirements for ASC platforms in general. In other words, a separate ASC user requirement document could capture requirements in a way that was not focused on “how” the requirements would be fulfilled.

The intent of doing this was to create a set of user requirements that were not linked to any particular computational platform. The idea was that user requirements would endure from one ASC platform user environment to another. The hope was that capturing the requirements in this way would assist in creating stable user environments even though the particular platforms would be evolving and changing. In order to clearly make the separation, the Tri-lab S&CS program decided to create a new title for the requirements. The user requirements became known as the ASC Computational Environment (ACE) Requirements.

Acknowledgements

We thank the ASC S&CS Program Element management team for providing the organizational framework and managerial support to ensure the completion of ACE Requirements Version 8.0 and this document.

Contributors

Alex Larzelere (Exagrid Engineering for SNL)

Jim Handrock (SNL)

Ann Hodges (SNL)

Karen Jefferson (SNL)

Dino Pavlakos (SNL)

Judy E. Sturtevant (SNL)

Jeff Brown (LANL)

Brett Kettering (LANL)

Randal Rheinheimer (LANL)

Bob Tomlinson (LANL)

Cheryl Wampler (LANL)

Brian Carnes (LLNL)

Steve Louis (LLNL)

Mary Zosel (LLNL)

Thuc Hoang (NNSA/HQ)

Contents

ASC Computational Environment (ACE) Requirements
Version 8.0
Final Report

With Input and Review from
Tri-Lab S&CS Program Element Management Team

Background

In preparation for the delivery and installation of the ASC Red Storm system, Sandia National Laboratories (SNL) embarked on documentation of the Red Storm Usage Model. Previous Usage Models were prepared by Lawrence Livermore National Laboratory (LLNL) for ASC White and Los Alamos National Laboratory (LANL) for ASC Q.

The ASC Q Usage Model was developed as a large spreadsheet that provided basic requirements for the ASC Q system and description of the features of the Q platform environment that met those requirements. The requirements for the ASC Q Usage Model were collected during a series of user meetings at each of the three laboratories. At those meetings, users were asked to consider their workflow, as they used the ASC platforms, and their requirements for the platform environment. This became a matrix with about 160 requirements as the rows. The columns in the matrix were the descriptions of how the Q platform environment met those requirements.

An important step in the planning and deployment of Red Storm was documentation of the Red Storm Usage Model Version 1. That Usage Model would include a mapping of Red Storm capabilities to the ASC Computational Environment (ACE) Requirements. As part of this process, it was determined that further refinement of the ACE Requirements was needed. Therefore, ACE Requirements Version 7s were updated to ACE Requirements Version 8.0.

Introduction

A decision was made early in the Tri-Lab Usage Model process, that the collection of the user requirements be separated from the document describing capabilities of the user environment. The purpose in developing the requirements as a separate document was to allow the requirements to take on a higher-level view of user requirements for ASC

platforms in general. In other words, a separate ASC user requirement document could capture requirements in a way that was not focused on “how” the requirements would be fulfilled.

The intent of doing this was to create a set of user requirements that were not linked to any particular computational platform. The idea was that user requirements would endure from one ASC platform user environment to another. The hope was that capturing the requirements in this way would assist in creating stable user environments even though the particular platforms would be evolving and changing. In order to clearly make the separation, the Tri-lab S&CS program decided to create a new title for the requirements. The user requirements became known as the ASC Computational Environment (ACE) Requirements.

ACE Requirements Scope

An important part of the refinement of the ACE Requirements was the reaffirmation of the scope for the requirements. The basic workflow organization from the White and Q Usage Models was left mostly intact, with some adjustments made to the data migration, problem set-up and post processing elements of the environment. The following are the major sections of the ACE Requirements:

1. Getting started (learning about the system, gaining access, etc.)
2. Setting up the work environment
3. 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

Figure 1 provides a graphical view of the scope.

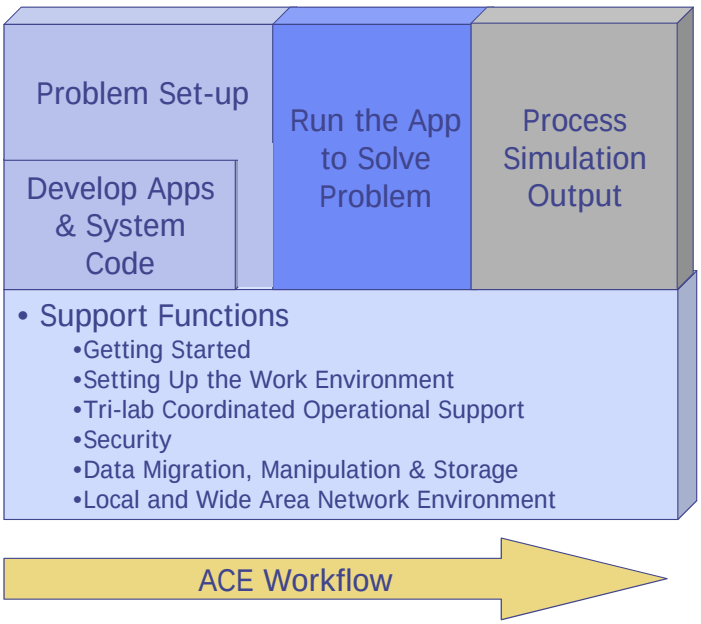

Figure 1. Graphical View of Scope

Refinement Methodology

The refinement of the ACE requirements took place over 2004 and 2005. The focus of this effort started with the requirements, as stated in the ASC Q Usage Model. The Q Usage Model requirements were edited to provide clarifications. In addition, some of the requirements were reworded to create firm expectations. Then a series of Tri-lab S&CS meetings were held to approve the edits and to add and delete requirements. After that, each laboratory team met with their users to validate the ACE requirements and to create "Levels of Need." S&CS recognized that the Levels of Need for each requirement would vary from laboratory to laboratory and decided that this be reflected in the finalized set. The refinement process ended with a Tri-lab meeting that captured the data from the individual laboratory surveys.

ACE Requirements

The ACE Requirements Version 8.0 are included as Appendix 1. The requirements were developed using Excel and are available as spreadsheets. In addition, the ACE requirements are also kept in a DOORS (Dynamic Object Oriented Requirement Systems) database. The use of DOORS can assist with the tracking of changes to the requirements and analysis of how the requirements are being implemented. The Red Storm Usage Models Version 1 and Version 2 include a mapping of the Usage Model capabilities to the ACE Requirements Version 8.0.

Conclusions

The purpose of separating the ACE requirements from the Usage Model was to create an enduring set of expectations from the ASC platform environment users. The idea is that, while there will be continued changes to the environments and the platforms themselves, the user requirements will not change at the same pace. That does not mean that the

requirements will not change at all. As users' workflows change and adapt to new mission demands and technologies, changes to the requirements are to be expected. Therefore, it is important that the ACE Requirements be viewed as a "living document" and be revisited on a periodic basis.

Appendix 1 – ACE Requirements

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
	1	**Getting started (learning about the system, gaining access, etc.)**					
	1.1	Platform Based Environment					
1			Shall provide machine policy information. This will include but not be limited to limitations on interactive use.	4	4	4	
2			Shall clearly specify platform security policies these shall include but not be limited to export control policies.	5	5	4	
3			Verify the environment by running a set of application based regression tests to exercises changes to the operating system, compilers, libraries, etc.		3	5	
	1.2	Learning about the system					
4	1.2.1	Web-based system information	Shall provide web-based system information.	4	5	4	This is an area that the labs need to do something about, if they are serious.
5			The web-based information system shall include open, secure, non-password protected pages that are available to the tri-labs.	4	4	4	
6			The web-based information system shall include links to platform-specific web pages.	4	4	4	
7			Shall provide current information about the system that is both open and secure.	4	4	4	
8			Shall provide web pages that are searchable.	4	4	4	
9			Shall provide a single, well-known point of entry for system information. This point of entry shall provide links to policy information.	4	4	4	
10			Shall provide system information in a way to allow increasingly more detail as topic areas are explored.		2	4	
11			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.	4	3	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
12			The ACE system information shall cover utilities for moving data between machines and within a platform group.	4	4	4	
13			System information shall be kept current and shall include a last modification date and email address for the personnel responsible for the page content.	4	3	4	
14			Shall provide a user feedback system.	3	2	3	
15			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.)*	2	2	4	
16			Shall maintain list of FAQs.	4	3	3	
17			Shall provide access to current system configuration information.	4	4	4	
18	1.2.2	On-line system information (on the machine)	Shall provide a current set of man pages for each platform to include locally developed software.	4	4	4	
19	1.2.3	Written system documentation	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.	4	3	4	
	1.2.4	Training					
20			Shall provide vendor and locally developed training information.	4	4	4	
21			Shall provide periodic tailored training for remote users at their sites. Targeted to specific groups like end-users and developers.	3	3	4	
22			Shall provide training materials that are electronically accessible.		5	5	
23			Shall provide trainer contact information.	3	2	3	
	1.3	Gaining access to the system (general)					
	1.3.1	Account and password management (authorization)					
24			Shall provide the ability to request accounts on-line. This shall include instructions and criteria for getting accounts.	5	5	4	
25			Shall provide information about tri-lab access policies and procedures. This shall cover all ASC computing resources with links to platform specific web pages.	4	3	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
26			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	3	4	
27			Shall provide guidance to users concerning which platform they should consider applying for an account based on the machine maturity and how their intended usage.	4	4	4	
28			Shall provide clear guidance on the authorization approval criteria.	4	5	5	
29			Shall provide a tri-lab service to process account requests, with user transparent links to site-local authorization mechanisms.	4	4	4	
	1.3.2	Gaining access to the machine (authentication)					
30			Shall minimize the number of authentications users are required to make while meeting security requirements.	4	4	4	
31			Shall support cross-laboratory honoring of credentials.	5	5	4	
32			Automated authentication support for batch jobs (e.g. cron), for nightly builds, regression testing, etc. (including documented solutions)			4	
33			Provide the ability for multiple users to have access control of a common running application.		5	4	
	1.3.3	System availability/scheduling information					
34			Shall provide real-time information via the web about system availability and system loading.	4	3	4	
35			Shall provide an estimated uptime when the system is unavailable.	4	4	4	
36			Shall provide the capability for users to determine the size of a job that can be run.	5	4	4	
37			Shall be able to determine the state of the current queue and job limits.	5	5	5	
38			Shall support use of lightweight scripts access to real-time availability information. *(e.g. Perl, CSH)*	5	3	3	
37			Shall provide users with current maintenance and upgrade schedules.	4	4	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
38			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.	5	4	4	
39			Shall limit access to secure job status information.	4		3	
40			A utility that returns the set of queues (authorized for user) that meet resource requirements (nodes, memory, etc.)		3	3	
41			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.		4	5	
42			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.	4	3	3	
	2	**Setting up the work environment**					
	2.1	File system standards & documentation					
43			Shall use standardized path names for the location of applications and project areas. *(e.g. /projects/<project-name>/)*	5	3	4	This is an area that the labs need to do something about, if they are serious.
44			Shall use configurations for home directories, scratch areas and archival storage that are available and consistent across platforms.	5	4	4	This is an area that the labs need to do something about, if they are serious.
45			Shall provide current documentation on configurations of home directories, scratch space and parallel file systems.	4	5	4	
46			File names should implement similar policies. (e.g. scratch, netscratch, vizscratch.)		3	4	This is an area that the labs need to do something about, if they are serious.
47			Shall provide appropriate information for all file systems. This shall include but not be limited to back-up policy, quota and purge policy.	4	5	4	
48			Shall provide tri-lab access to common disk space to enable a remote build capability.		2	4	
49			At least one file system shall be provided backup for source code.		4	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
	2.2	Paths, environment variables, modules, etc.					
50			Shall provide a documented and maintained skeleton template cshrc file and others that provides the basics for setting up the user environment.	4	4	4	
51			Shall allow users to change the login shell without requiring sys admin intervention.	3	4	4	
52			Shall provide home directories that are portable across multiple architectures	5	4	4	
53			Include a standard architecture-specific environment initialization capability. (e.g. cshrc.aix, .cshrc.tru64, etc.)	3	4	4	
54			Shall have a standard module capability on all platforms to manage software dependencies.	5	3	2	This is an area that the labs need to do something about, if they are serious.
55			Shall standardize the system environment variable across platforms (to the degree possible) to enable portable user scripts.	4	3	4	This is an area that the labs need to do something about, if they are serious.
56			Shall provide paths to common software that are standardized across platforms.	5	4	4	
57			Provide other initialization scripts (termcap, emacs, vim)		3	3	
	2.3	Setting up user groups					
58			Shall provide documented processes that allow tracking additions and deletions to groups.		3	3	
59			Shall provide a service for group owners to set up and manage groups.		4	4	
60			Shall provide a hierarchical group capability. (Used to expand the number of groups that a user can belong to.)	4	4	3	
61			Shall provide a user level mechanism to change the default group.		4	4	
62			Shall provide the capability to set up groups for export control purposes.		4	4	
63			Shall have a single gid space across the complex to enable imported tar files to map to the correct group.	3	3	3	This is an area that the labs need to do something about, if they are serious. Check with ICSI

Req #		Section	Tri-Lab ACE Requirements		Level of Need			Comments
					LANL	LLNL	SNL	
	3	**Data Migration**						
	3.1	Tools for Transferring Files						
64			Shall provide the infrastructure to support high speed remote bulk data transfer that can be executed unattended and is resilient		5	5	4	
65			Shall provide tools for high performance and scalable data transfer that can be initiated from either side. (currently pftp)		5	5	5	
66			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	4	5	5	
				Local to remote	4	4	5	
				Remote to local	4	4	5	
				Remote to remote	3	3	5	
67			Shall provide local files systems that are accessible over the WAN.		4		4	
69	3.2	Staging Data to the Machine (source code, input decks, data, etc.)						
70			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.)*		4		4	
71			Shall provide a location where libraries and tools associated with an application can be maintained.		5	5	4	
	4	**Application and system code development**						
	4.1	Gaining access to a machine for code development						
72			Provide resources configured to facilitate code development (system and application.)			4	4	
	4.2	Peculiarities of the system						
73			Shall document deviations from POSIX and ANSI/ISO standards.		4	4	3	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
74			Shall document deviations from compliance with language and run-time standards. *(e.g. MPI, I/O, Cray pointers etc.)*	4	4	4	
75			Shall publish deviations from IEEE arithmetic.	3	3	3	
76			The ACE system information shall provide current documentation for compilers, libraries and development tools.	5	4	4	
77			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		3	4	
78			Shall provide a "quick start" guide for users of new platforms.		5	5	
	4.3	Configuration Control					
79			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.		3	4	
80			Shall provide access to source repository (e.g. SourceForge, Razor, CVS, etc.)		3	4	
	4.4	Parallel programming models and run-time systems					
81			Shall provide the ASC standard compliant MPI library (currently 1.2) optimized for both on-box and across-box messaging.	5	5	5	
82			Shall provide POSIX compliant thread package.	4	5	2	
83			Shall provide support for hybrid OpenMP/MPI parallel programs	2	4	2	
84			Shall provide documentation with guidelines for running OpenMP and MPI programs (especially for threadsafe code.) *(e.g. how to tune buffer sizes, etc.)*	4	5	5	
85			Shall make it clear to users which MPI library is loaded, especially when wrappered compiler commands are used.	4	4	5	
86			Shall provide support for locally developed libraries that can be	5		3	

Req #		Section	Tri-Lab ACE Requirements		Level of Need			Comments
					LANL	LLNL	SNL	
			centrally accessed . *(e.g. LA-MPI, UPS, HYPRE, etc.)* .					
87			Need a mechanism to force a specific version of MPI to be linked into the code			5	5	
88			Need a standard interrupt propagation mechanism to tasks associated with a parallel application.			4	4	
89			Consistent integration with the batch system, and access to resource availability in order to build the MPI run command				3	
90			Provide mechanisms in MPI for enhanced reliability – e.g. survive NIC failures				5	
	4.5	Addressing IO: effective use of the file systems, serial and parallel IO libraries						
91			Need an API to determine whether a file system is local or global, and serial vs. parallel			2	2	
92			Shall provide guidance for code developers to make effective use of serial and parallel file systems, including of the main file systems primarily to include but not limited to large parallel file system-topology.		5	5	5	
93			Shall provide platform specific single global parallel file system visible from all nodes with homogeneous node IO performance to the file system.		5	5	4	
94			Shall provide a single parallel file system that spans platforms		5	4	4	
95			Shall provide links to information about the system parallel file system.		4	4	4	
96			Shall provide optimized and standard compliant parallel I/O libraries:	MPI IO	5	5	4	
				HDF5	4	4	4	
				pNetCDF	4	3	4	
				UDM		2		
97			Shall provide IO libraries that are thread safe with scalable performance.		5	5	4	
98			Shall allow for the number of concurrent open files (including system files) equal to approximately ten times the number of processors in use by the job.		2	4	4	

Req #		Section	Tri-Lab ACE Requirements		Level of Need			Comments
					LANL	LLNL	SNL	
	4.6	Third party libraries and utilities						
99			Shall manage changes to center supported third party libraries via a change control procedure.		5	4	4	
100			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.		4	3	4	
101			Shall provide the capability for developers and support personnel to maintain products without requiring sys admin privileges.		4	5	5	
102			Shall provide system documentation that includes a POC for each center supported third party product on the system.		3	3	4	
	4.6.1	Math libraries (including solvers)						
103			Shall provide support for standard sparse matrix operations		3	4	3	
104			Shall provide support for parallel sparse linear solvers		4	4	2	
105			Shall provide optimized BLAS and LAPACK.		3	4	5	
106			Shall provide system documentation that point to what solvers are installed and the support POC.		4	3	3	
	4.6.2	Networking and other libraries						
107			Shall support libraries that are callable from C++. (Note: *If important please supply which libraries.)*			?	4	
108			Shall support secure interprocess communications such as sockets, HTTPS, and/or ssh.		5	5	3	
109			Shall support the deployment of client/server socket-based parallel tools (including visualization) on platforms.		4	4	3	
	4.7	Compilation						
110			Shall provide user recommendations for recommended compiler options.		3	4	4	
111			Shall provide the latest 64-bit versions of the gnu utilities. (*e.g. make, g++, gawk, gdb, gawk, etc.)*		5	5	5	
112			Shall provide parallel make and compilation in build.		5	5	5	
113			Shall provide standard compliant compilers.	C	5	5	5	
				C++	5	5	5	
				Fortran9X	5	5	5	

Req #		Section	Tri-Lab ACE Requirements		Level of Need			Comments
					LANL	LLNL	SNL	
114			Shall provide compiler support for F77 codes including support for Cray pointers.		5	5	5	
115			Shall provide information about different compiler versions resident on the system and guidance to the user as to which one to use.		4	4	4	
116			Shall provide compiler support for OpenMP.	C & OpenMP	1	4	2	
				C++ & OpenMP	1	4	2	
				Fortran9X & OpenMP	1	4	2	
117			Shall provide JAVA support via standard JDK toolkit.		3	4	2	
118			Shall provide common scripting languages. (e.g. perl, python,Tcl/Tk, sh, csh, tcsh, ksh, and expect))		5	5	4	
119			Shall provide support for dynamic/shared libraries.			4	3	
	4.8	Debugging and correctness testing						
120			Shall provide the ASC standard parallel debugger. (currently TotalView)		5	5	5	
121			Shall provide a usable debugging capability for jobs that span across 1/3 of the system.		5	3	5	
122			Shall provide common tools for parallel aware memory checking including user callable routines to track memory use and availability. (e.g. Valgrind type functionality)		5	5	5	
123			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.)		4	4	4	
124			Shall provide tools for static analysis of source. *(e.g. flint)*		4	3	3	
125			Shall provide tools that measure testing code coverage for all supported standard languages.		5	3	3	
	4.9	Performance measurement, analysis and tuning						
126			Shall provide a message tracing tool that is usable for jobs that span across 1/3 of the system. *(e.g. Vampir)*		4	4	4	
127			Shall provide a message profiling interface *(e.g. mpiP)*		4	4	4	
128			Shall provide loop level application profiling tools.		5	4	4	
129			Shall provide tools to analyze cache performance.		5	4	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
130			Shall provide user level access to hardware performance counters, preferably via the PAPI API.	4	5	4	
131			Shall provide lightweight high-resolution timers.	5	5	4	
132			Shall make it clear which tools can be used together. *(e.g. compiler and memory tool, or debugger and mpi)*	4	4	4	
133			Shall provide tools to access thread performance.		3	3	
134			Shall provide tools to characterize the IO performance of a code.	4	4	4	
135	4.1	Best Practices	Shall publish “best practices” for performance.	4	3	4	
136	5	**Problem setup**	Shall provide users with resources and connectivity to generate meshes/ netlists/ schematics, material databases, and necessary data storage for application data bases.		4	5	
137	5.2	Domain decomposition	Shall provide real-time information to support smart domain decomposition to allow near optimal mapping onto the machine, taking into account non-homogeneous system resources.	4	2	5	
	6	**Running the application to solve the problem**					
	6.1	Submitting the job (local and remote)					
138			Shall provide users with a common queue syntax and semantics across the tri-lab. *(i.e. “large queue” means the same across the tri-lab)*	2	3	4	This is an area that the labs need to do something about, if they are serious.
139			Shall provide a common user interface to access different underlying resource management system resource information. *(Note to achieve tri-lab common interface – may require change to your scripts.)*	3	3	4	This is an area that the labs need to do something about, if they are serious.
140			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.)	3	2	3	
141			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)	4	2	3	
142			Shall make the remote job submittal process the same across the complex.	3	3	4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
143			Shall provide resource management mechanisms that support accurate and reasonable time limit and node allocation requirements.	4	4	4	
144			Shall support dependent jobs.	5	5	4	
145			Shall provide automatic job submission e.g. for regression testing. *(e.g. cron)*	5	4	5	
146			Shall publish a "good neighbor" policy to advertise the policy for submitting jobs to the queues.	2	4	4	
147			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)*	3	3	4	
148			Shall provide resources for quick turnaround for small regression tests		3	5	
	6.2	Monitoring the job					
149			Shall allow users to examine the contents of files being used by an executing job.	5	5	5	
150			Shall be able to attach to a running job to control, steer and monitor it. (e.g. run proxy.)	4	4	4	
151			Shall be able to monitor memory and cpu use for load balancing purposes.	4	4	4	
	6.3	Stopping the job					
152			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.	5	5	4	
153			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	3	4	
154			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).*	4	4	5	
	6.4	Interactive use					
155			Shall provide some portion of the system to be available to support timely interactive use.	5	4	5	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
156			Shall provide dedicated resource allocation for interactive use. (e.g. for computational steering or debugging)	5	4	5	
	6.5	Adapting the job for expected system reliability					
157			Shall provide a history of machine reliability to assist users in determining frequency of restart dumps.	3	3	4	
158			Shall provide users with information about planned system down time.	4	5	4	
159			Shall provide the capability to transparently migrate user processes if a node goes down.	3	2	5	
	6.6	System reliability					
160			Shall provide tools and systems to provide a reliable environment for users. (e.g. lock out unreliable or nodes with other problems)		5	5	
161			Shall provide procedures for “automatic” job restart after system interruption.	3	3	4	
	7	**Processing simulation output**					
162	7.1	Prepare data for analysis	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..	5	4	5	
	7.2	Analyze the results					
	7.2.1	Visualization					
163			Shall document what viz tools are available.	4	4	4	
164			Shall establish policies that are established for utilization of compute platform visualization nodes.	4	4	4	
165			Shall provide the ability to visualize the data in place without the need to move data to another system.	4	4	4	
166			Shall provide computing (hardware) resources to be used for data extraction and visualization as part of an interactive visualization process does not require the movement of data.	4	4	4	
167			Shall provide a full featured scalable parallel visualization tool. This requirement currently met by Ensight and VIzIt.	5	4	4	
168			Shall provide data analysis capabilities. IDL is the preferred tool due to a link to experimentalists’ data.		1		

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
169			Shall provide visualization from the desktop.	5	5	5	
	7.3	Hand off the results to another user.					
170			Shall provide a command to transfer a file to another user. Must be a standard interface and a standard location. (e.g. “give” command)	5	5	5	
	7.4	Archive the results of the simulation					
171			The archive implementation shall provide high reliability. Provide disaster recovery mechanisms for user identified files to attain even higher reliability.	5	5	5	
172			The archive implementation shall provide high availability. Provide failover for archive write.	5	5	5	
173			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)		3	4	Users are looking for an integration of SourceForge and SimTracker with the data management tools.
174			The archive implementation shall provide data security. To include access control lists, portable across HPSS platforms/sites implementations.	5	4	5	
175			The archive implementation shall provide the capacity sufficient to track user needs.	5	5	5	
176			The archive implementations shall provide common functionalities.	5	4	4	
177			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)*	5	4	4	10% of FS BW for large files for write 2% of FS BW for <1mb files (e.g. via htar) for write Read issue is latency (metric?) Performance metrics mean: delivered to the user
178			Shall provide a common (across systems and tri-lab) user interface to the archive.	3	4	4	
179			Shall provide utilities to consolidate many small files into a single file. *(currently htar)*	4	4	4	
180			Shall provide a persistent archiving capability with seamless transition across archival systems.	5		4	

Req #		Section	Tri-Lab ACE Requirements	Level of Need			Comments
				LANL	LLNL	SNL	
	8	**Tri-lab coordinated operational support**					
	8.1	User support					
181			Shall publish and maintain phone numbers, email addresses, and hours of support hotline and help desk.	5	5	5	
182			Shall provide contact information for reporting problems outside working hours.	4	5	5	
183			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.)	5	5	5	Working hours for the tri-lab community, which encompasses PST and MST, can be defined as 8 AM MST to 6PM MST.
184			Shall have the capability for open and secure problem reporting and tracking for each system.	5	5	5	
185			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.	3	4	4	
	8.2	Trouble shooting (in depth consulting)				4	
186			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.	5	4	4	
187			Shall provide enhanced system support for early machine problems and milepost calculations.		4	4	.

DISTRIBUTION:

5	Alex Larzelere Exagrid Engineering 6511 Wynema Ct, Suite 180 Alexandria, VA 22315
1	Los Alamos National Laboratory Attn: Jeff Brown, F465 PO Box 1663 Los Alamos, NM 87545
1	Los Alamos National Laboratory Attn: Brett Kettering, B294 PO Box 1663 Los Alamos, NM 87545
1	Los Alamos National Laboratory Attn: Randal Rheinheimer, B294 PO Box 1663 Los Alamos, NM 87545
1	Los Alamos National Laboratory Attn: Bob Tomlinson, B260 PO Box 1663 Los Alamos, NM 87545
1	Los Alamos National Laboratory Attn: Cheryl Wampler, F644 PO Box 1663 Los Alamos, NM 87545
1	Lawrence Livermore National Laboratory Attn: Brian Carnes, L-067 7000 East Avenue Livermore, CA 94550
1	Lawrence Livermore National Laboratory Attn: Steve Louis, L-554 7000 East Avenue Livermore, CA 94550
1	Lawrence Livermore National Laboratory Attn: Jean Shuler, L-554 7000 East Avenue Livermore, CA 94550
1	Lawrence Livermore National Laboratory Attn: Mary Zosel, L-555 7000 East Avenue Livermore, CA 94550

1	Thuc Huong Department of Energy National Nuclear Security Administration Office of Advanced Simulation and Computing, NA-114 1000 Independence Ave SW Washington, DC 20585	
1	0139	Robert K. Thomas, 1904
1	0139	Paul Yarrington, 1910
1	0431	Ann Hodges, 0232
5	0807	Judy E. Sturtevant, 4328
1	0807	John P. Noe, 4328
1	0822	Dino Pavlakos, 9326
1	0823	John Zepper, 4320
1	1319	James A. Ang, 1422
1	1322	Sudip Dosanjh, 1420
1	9151	Howard Hirano, 8960
1	9152	Jerry Friesen, 8963
1	9153	James L. Handrock, 8810
1	9159	Mike Hardwick, 8964
1	9201	Karen Jefferson, 8116
2	9018	Central Technical Files, 8944
2	0899	Technical Library, 4536