SANDIA REPORT

SAND99–1143
Unlimited Release
Printed May 1999

DISCOM^2: The Distance Computing SP2 Pilot FY98 Report

Thomas J. Pratt, Thomas D. Tarman, Martha J. Ernest, John P. Noe, Walter H. VanDevender, Sue P. Goudy, Rupert K. Byers, Judy Beiriger, David P. 'Dave' Wiltzius, and David N. Shirley

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 under Contract DE-AC04-94AL85000.

Sandia National Laboratories

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

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

Prices available from (703) 605-6000
Web site: http://www.ntis.gov/ordering.htm

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

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND99–1143
Unlimited Release
Printed May 1999

DISCOM^2: The Distance Computing SP2 Pilot FY98 Report

Thomas J. Pratt, Thomas D. Tarman, and Martha J. Ernest
Advanced Network Integration

John P. Noe, Walter H. VanDevender, Sue P. Goudy, Rupert K. Byers
Scientific Computing Systems

Judy Beiriger
Decision Support System Architecture

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0806

David P. 'Dave' Wiltzius
Lawrence Livermore National Laboratory
P.O. Box 808 / L-63
Livermore, CA 94550

David N. Shirley
ABBA Technologies, Inc.
2403 San Mateo NE, Suite P-16
Albuquerque, New Mexico 87110

Abstract

The items discussed in this report reflect the work in progress during FY98.

As a way to bootstrap the DISCOM^2 Distance Computing Program the SP2 Pilot Project was launched in March 1998. The Pilot was directed towards creating an environment to allow Sandia users to run their applications on the Accelerated Strategic Computing Initiative's (ASCI) Blue Pacific computation platform, the unclassified IBM SP2 platform at Lawrence Livermore National Laboratory (LLNL). The DISCOM^2 Pilot leverages the ASCI PSE (Problem Solving Environment) efforts in networking and services to baseline the performance of the current system. Efforts in the following areas of the pilot are documented: applications, services, networking, visualization, and the system model. It details not only the running of two Sandia codes CTH and COYOTE on the Blue Pacific platform, but also the building of the Sandia National Laboratories (SNL) proxy environment of the RS6000 platforms to support the Sandia users.

CONTENTS

List Of Figures

List Of Tables

1. Introduction

The SP2 pilot was designed to initiate the activities surrounding the distance computing portion of the DISCOM^2 program. This report discusses the activities of the pilot for the period of March 1998 to October 1998. The focus of the pilot was to build a computing environment that connected SNL applications and infrastructure to LLNL's ASCI (Accelerated Strategic Computing Initiative) computing platform Blue Pacific. For this initial pilot the scope was to investigate how MPICTH and COYOTE, two SNL applications would be run on the Blue Pacific architecture at LLNL. Another objective was to explore the services needed to maintain a pre-eminent high performance computing capability without being local to the high-end computational platform. A small IBM RS6000 based environment was created at SNL, as a proxy to the large machine at LLNL in order to begin the understanding of how closely the sites could be coupled/linked.

Network development and testing concentrated on the end to end performance issues of distance computing. Key elements of the networks within SNL and LLNL as well as the network interface cards (NICs) and protocol stacks on the high-end platforms were characterized. Tuning parameters and bottlenecks were identified at different layers.

Visualization plays a major role in the simulation and modeling environment of the typical user. As such, this area was also investigated to begin the analysis of remote vs local visualization services. Several approaches are described within the report.

In addition, system model work began this past year. The initial efforts were to document current and future user requirements/needs. User interviews started this process and initial use cases for the model were developed.

The experiences gained in the areas above and their future directions are expounded upon within the report. Basically, what has been done to further understand the environment that will be required to allow individuals as representatives of institutions to have access to ASCI machines has begun with the SP2 Pilot and will continue with the next evolution of the Pilot initiative.

2. LaPorte Servers

The proxy systems at Sandia are called LaPorte1 and LaPorte2. Each of these systems is an IBM RS/6000 Model F50 with two PC604e processors and 256MB of memory. (Essentially half of a node of the TR (Technology Refresh) system at LLNL.) AIX 4.2.1 is installed. Each system has an Ethernet connection (10 Mbits/sec) and an ATM connection (155 Mbits/sec) into the ASCI Red unclassified network. Each system has local disk storage of 9.1 Gbytes.

The standard IBM compilers (version 5) are installed. The Parallel Operating Environment (POE) and the MPI (Message Passing Interface) libraries are installed. The Load Leveler product is available but is not being using. DPCS, the batch system at LLNL is not currently installed but should be soon. These and other user environment tools are listed in table 2.1.

DCE/DFS clients are active on the system and access to the DFS file space is available. Local Sandia customers use these systems for compilation and editing. Access to the systems is via "ssh" and not telnet. Accounts on this system are restricted to the DISCOM^2 project personnel.

A process has not been established to track the ASCI Blue Pacific software system updates. The initial system installation was prepared from the list of modules loaded on Blue Pacific and obtained with the AIX command "lslpp –l".

The total cost for these systems was approximately $70,000 including hardware and software.

Table 2.1 List of user environment tools available on LaPorte

	Compilers
gcc	GNU project C compiler
g++	GNU project C++ compiler
xlf	IBM's standard AIX Fortran 77 compiler
xlf_r	IBM's SMP Fortran compiler; for use with threads
xlf90	IBM's standard AIX Fortran 90 compiler
xlf90_r	IBM's SMP Fortran 90 compiler; for use with threads
xlc	IBM's standard AIX C compiler
xlc_r	IBM's SMP AIX C compiler; for use with threads
xlC	IBM's standard AIX C++ compiler
xlC_r	IBM's SMP AIX C++ compiler; for use with threads

mpxlf	IBM's Fortran 77 compiler for parallel jobs using IBM's MPI
mpxlf_r	IBM's SMP Fortran 77 compiler for parallel jobs using IBM's MPI and threads
mpcc	IBM's C compiler for parallel jobs using IBM's MPI
mpCC	IBM's C++ compiler for parallel jobs using IBM's MPI
	Debuggers
xpdbx	IBM's Parallel Debugger
	Editors
emacs	The GNU emacs editor
ex	Edits lines interactively with screen display
sed	Provides a stream editor
vi	Edits files with a full-screen display
	Loaders
ld	Links object files
xlf	Links Fortran object files
xlc	Links C object files
xlc	Links C++ object files
	Message Passing
MPI	IBM's MPI library
	Parallel Operating Environment
poe	Submit a POE job. POE environment variables
poekill	Kill a running POE job
	Performance Analysis Tools
xprofiler	Must use the –pg flag on your compile and load lines
vt	IBM's monitoring tool. Provides visual monitoring of the message behavior
	X11
X11R5	The standard X11 package with IBM modifications

3. Experiences with MPICTH on Blue Pacific

3.1 Introduction

The experiences of an application user in developing and running the CTH code on a new machine are documented. This is a snapshot in time of the issues and problems experienced in finding out what it takes to use a new platform. A great deal of knowledge was gained and many lessons were learned. The Blue Pacific configuration is dynamic, current information can be obtained from web pages or calling the LLNL hotline.

CTH is an Eulerian, explicit finite-difference code which utilizes a regular grid for problem specification and solution. Most of the code is written in Fortran77, with extensions for dynamic memory allocation. The MPI variant of the parallel CTH implementation was chosen because this version has been tested on other SP systems; PVM can also be used on IBM systems. It was believed that this would minimize the time and effort required to get CTH running on Blue Pacific.

The motivation for selection of CTH as one of the codes in the DISCOM^2 Pilot Project is threefold. For a given problem CTH memory requirements are clearly defined. For suitably sized problems solved in parallel, CTH computation dominates communication. For all parallel decompositions of CTH problem space, the communication pattern is regular. Because of the long history of CTH development at Sandia these properties are well understood. The interested reader can obtain further information from "Multi-Processing CTH: Porting Legacy FORTRAN Code to MP Hardware", by Bell, Elrick, and Hertel.

The regularity of the CTH problem grid allows the user to predict memory usage for the problem one wishes to solve. From that information and the known availability of memory on a system, the user can request an appropriate number of processors to optimize the fit of the problem in memory. It is also possible for a user to tailor memory usage by changing the mesh size for a particular problem. This property proved to be useful for determining the amount of effective memory per node on Blue Pacific nodes. (See the section on Scaling Tests.)

The parallel implementation of CTH was initially developed on the Intel Paragon and was adapted for the Intel Teraflops computer; it has also been run on Origin 200, Cray T3E, and DEC 8400. The execution characteristics of CTH have been studied extensively in the Teraflops environment. Thus it is well-known that computation time strongly dominates communication time, provided that each node in a parallel task has sufficient work to do. Scaling tests have been designed to discover the optimal problem size per processing element and a set of these benchmarks was run on Blue Pacific.

Communication in MPICTH is regular in both time and space. During each computational cycle each processing element (PE) exchanges information with its nearest neighbors. There are approximately seventy-two large messages (between 0.5 and 1.0 megabytes) per cycle. The problem domain is decomposed so that each processing element in a logically rectangular array has a well-defined part of the domain to work. The code attempts to form nearly cubical subdomains so that the surface area to volume ratio of subdomains will be small, thereby minimizing communication overhead. During each cycle, processing elements exchange values which have been calculated at processor boundaries. Computation then proceeds with the updated values on each PE. For a given problem the amount of the exchanged information is the same at each time step.

In the following sections the maintenance of MPICTH executables, the lessons learned while running the scaling tests, and the resources required for solution of a moderately sized problem are discussed.

3.2 Environments for Code Configuration Management

When work began on the DISCOM^2 Pilot Project in May, the only option for construction of MPICTH was to copy the source code from its Sandia repository to Blue Pacific for compilation. The current CTH distribution uses make files and the compilation process went very smoothly. Within an hour after the files were transferred, the serial and parallel CTH executables were built. The serial version of CTH and the parallel version of MPICTH were tested in single processor mode. At that time, it was impossible to test multiprocessor MPICTH because the four interactive nodes in the debug partition were in use. Since then, the interactive partition has acquired four additional nodes and is much more usable for quick turnaround tests. The next task was to learn to use all four CPUs on a node. This effort will be described in the section on Scaling Tests.

In September, MPICTH was rebuilt to incorporate the visualization library. At that time, an alternative compilation environment, a pair of IBM F50 workstations, had become available at Sandia. The environment of the F50 matched that of Blue Pacific as closely as possible, so the executables produced by both compiling systems were compared. For the comparison, the same sources and makefiles, with local copies of the files on each machine were used.

Again the process on Blue Pacific went smoothly; however, this time the compilation took several hours instead of minutes. A probable reason for the discrepancy is that there were many active login sessions on the node where the make command was invoked. By contrast, the F50 compilation completed within an hour. Note that the same source code and compiler optimizations were used on Blue Pacific and on the F50.

The executables produced on Blue Pacific and on the F50 were indistinguishable. In fact, the current executable on Blue is the MPICTH, which was compiled and linked on the F50. This means that our export-controlled source code can be kept in a repository which

is local to Sandia. It is natural to ask if our executables can be kept at Sandia, say in DFS space, and loaded from this file space at run time.

3.3 Scaling Tests

Although the production of MPICTH from source code was easily accomplished, the actual use of the code for parallel solution of problems was not straightforward. There were several obstacles to successful completion of the scaling benchmark: learning to use all four CPUs on a node, using the batch system optimally, utilizing the parallel file systems, and discovering the amount of effective memory available on a node. The successful resolution of these issues is described first. In the latter part of this section the explanation of the scaling benchmark and timing data are described. Lastly, an outline for further work on scaling issues is presented.

Blue Pacific is a different kind of machine, and required time to learn how to use it. Fortunately, the folks at Livermore have provided several paths for getting help. There is an abundance of on-line documentation for the user who wants to be relatively independent. There are Web pages based at http://www.llnl.gov/asci/platforms/bluepac/, man pages on the batch system (psub and related commands), and a directory (blue:/usr/local/docs) with many text files and user manuals in PostScript or Acrobat format. Livermore Computing runs a telephone/email hotline service, and the staff is friendly and helpful. The service seems to be slanted more toward providing help over the telephone. (That is to say, telephone response was found to be better than email response.)

New users should scan the news items, which are presented as a list upon login. The most useful items were those shown in Table 3.1.

do.and.dont.list	**Mon. Apr. 13 11:57:03 PDT 1998**
paging.space	**Thu. Jun. 11 14:19:58 PDT 1998**
large.memory	**Thu. May 14 14:38:49 PDT 1998**
job.node.status	**Wed May 13 20:33:09 PDT 1998**
parallel.io	**Wed May 13 10:49:01 PDT 1998**
Hack_Update	**Wed May 6 14:53:58 PDT 1998**
info.messages	**Mon. May 4 13:32:10 PDT 1998**
job.limits	**Wed Apr. 29 12:36:18 PDT 1998**
piofs.user.dir	**Wed Apr. 29 12:27:41 PDT 1998**
documentation	**Mon. Apr. 13 09:26:58 PDT 1998**
README	**Fri Apr. 10 11:29:58 PDT 1998**
one_job_per_node	**Tue Mar 31 12:42:07 PST 1998**

Table 3.1 Recommended Reading List

The current configuration of Blue Pacific has a combined MPP and SMP architecture consisting of SMP nodes where each node has four CPUs (332-MHz PowerPC 604e) and 512 megabytes of memory. Utilization of all CPUs on a node is not automatic, and the user must learn how the trick is done. There is one document which was found to be particularly useful: /usr/local/docs/TR_Hack.doc . This file summarizes some of the configuration changes in migrating from the previous to the current system, and discusses changes required to use the SMP architecture of multiple CPUs per node. The following passage, which explains how to use Blue Pacific compute nodes effectively, is taken directly from the TR Hack document.

> **The TR systems will continue to run in "dedicated-node" mode; that is, there will not be more than one user's job using the multiple CPUs within a given node at any given time. Therefore, a job's ability to use most or all of the CPUs within a node is important to the efficient use of the TR systems.**
>
> **While applications running on the ID systems probably have been coded for parallelism across nodes, for example using MPI, these applications may not have an equivalent version to take advantage of parallelism within each SMP node.**
>
> **One technique for parallelism within a node would be compiler-assisted parallelism. On the IBM SPs, the Fortran compiler, xlf_r, supports a subset of OpenMP SMP directives. The xlf_r -qsmp option can be used to attempt automatic parallelization of a code, and the -qsmp=noauto option can be used if you are inserting directives into your code to direct the parallelization.**
>
> **Another approach would be to use POSIX Threads (PThreads) to code for parallelism within a node. To use the Pthreads library, you need to compile your code using the Fortran xlf_r or the C xlc_r re-entrant compilers. Threads of execution could be created by a process running on a node to compute on the data resident on that node, and thread synchronization routines would be used to ensure orderly access to the shared data. These threads execute on the CPUs of the node and all use the same process space. MPI communications continue to be used to pass data between nodes. Programming with Pthreads involves more effort, but also provides the potential for greatest efficiency in the use of machine resources and performance gains.**
>
> **These approaches to achieving parallelism within a node, especially the second one, can involve considerable effort. Given that most codes running on the ID system likely use MPI across nodes, it might seem more straightforward to just try to use MPI for communication**

within a node as well. However, the current versions of the LoadLeveler and POE software do not support the start up of multiple processes in user space communication mode on a node, and the software release that will support this is not expected to be available until mid 1998. Currently, only multiple IP processes can be run on a single node.
We suspect that most applications running on the ID systems have not been coded to take advantage of parallelism within a node. If these applications now run on the TR system as they have been run on the ID systems, the net effect might be that three out of every four CPUs on the TR nodes would be idle. To avoid this situation, a set of modifications have been implemented to create an interim strategy whereby each of the four CPUs on a TR node can be used as four "effective nodes." This set of modifications, which you may have heard affectionately referred to as "The Hack", supports the use of an optional -g option (geometry) with PSUB to specify at batch job submission how the job's MPI tasks should be distributed to the nodes allocated for a job. For interactive jobs, the optional -g option can be used by executing POEFE (a POE front end) instead of POE.

Jobs submitted using the -g option where the geometry specified will result in more than one task being assigned to any allocated node will not be able to use the switch in user space mode, and the environment variables will be automatically set to use the switch in IP mode. This set of modifications includes changes to DPCS and LoadLeveler and the introduction of the POE front end, POEFE.

By following examples in the TR Hack document, all four CPUs on an interactive compute node were quickly usable. In order to get MPICTH running quickly, with a minimum of effort, the IBM SP2 switch in IP mode was used. Without doubt this choice impacts performance; however, it was very easy to accomplish by using the geometry option for POEFE. It seems unlikely that MPICTH will be rewritten to use Posix Threads in the near future but a version with OpenMP directives does exist. Future work will include an attempt to compile this version and compare timing data with that of MPICTH running in IP mode.

To run the set of scaling benchmarks the batch system DPCS was chosen to run all the tests, even those which required only a few processing elements. The structure of scripts and the submission command are very similar to the corresponding elements of NQS and the TR Hack document explains how to use the geometry option in a batch script. Thus, it was fairly easy to use multiple CPUs on a node in the batch pool - at least for a problem with small memory requirements.

Although the scaling benchmarks will be described in detail later, it is relevant to state now that the initial input decks had been run successfully when Blue Pacific was in its original configuration (one CPU per node). These input decks were designed to use most of the nominally available 128 megabytes of memory per CPU. The single CPU test ran to completion in about ten minutes, and the 2-CPU test completed in about twelve minutes. It was surprising when the 4-CPU test did not even make a decent start in half an hour!

The following describes the batch time limits and interaction with the Livermore hotline lc-hotline@llnl.gov . The following table was presented in the news item job.limits:

Pbatch pool***	Max wall* clock time	Maximum* num. Nodes	Num. Active Jobs* (any state)**
Days (8AM-5PM)	2.0 hours	64 (256 CPUs)	2
Night (5PM-8AM)	8.0 hours	128 (512 CPUs)	2
Weekends (5PM Fri-8AM Mon)	12.0 hours	154 (616 CPUs)	2

Table 3.2 Batch Queue Node Time Limits

Now why did a 4-CPU job get only 30 minutes when the limit is two hours? The response from the hotline was prompt and succinct: the default time limit is half an hour, and the maximum limit can be obtained only by asking for it at job submission time. The response also included the correct psub option (-tM). This information was really helpful because the man page for psub lists two other options for dealing with time limits, at least one of which has no ascertainable effect. Armed with this the 4-CPU test was resubmitted—but it did not complete in the requested two hours!

The 4-CPU job was resubmitted and monitored using the jr command. The output from that command eventually led to the speculation that the test was getting very little CPU time because of oversubscribed memory (always anathema in a virtual memory system). At this point, two separate courses of action were taken. Because part of the project was to interact with (and evaluate) the hotline support, a lengthy exchange of email messages with that entity was initiated. And, to test a hypothesis, the problem size for the set of scaling benchmark tests was drastically reduced. All went well until the use of 128 CPUs was attempted - and so the next obstacle was observed, the parallel file system(s).

Based on instructions in the news item do.and.dont.list , CTH environment variables were established so that the large output files would be written to one of the parallel file systems. In the beginning of running MPICTH on Blue Pacific the only such file system type was PIOFS. The difficulty here was that for a 128-CPU job there would be a random small collection of zero-length restart files produced in the "gen" phase of the job. This is not normal behavior, and the "cth" phase of the job would terminate with ugly messages.

Fortunately, hardware developments resolved this issue. When the GPFS file system was made available for general use, the CTH environment variables were changed so that large output files would be written to the new type of parallel file system and the zero-length files no longer occurred. Note that MPICTH is using Fortran I/O to create and write these files. To use the GPFS efficiently with MPI requires code modifications to call library routines for parallel file I/O. There is a PostScript document (blue:/usr/local/docs/mpiio_gpfs_user.ps) which describes the necessary changes.

As stated previously, in order to get the scaling benchmarks running on more than four processors the per-CPU memory requirement was reduced. In fact, by using hints from the news item paging.space and output from the jr command, "good" problem size was discovered to be about one-third of the nominally available physical memory. Coupled with the MPI limitation of 512 tasks, this means that effective problem size for MPICTH on Blue Pacific is substantially smaller than on the ASCI Red machine. One potential user of MPICTH on Blue Pacific, when told of these results, exclaimed that the machine is useless in its current configuration. Since this was clearly not an acceptable conclusion for the DISCOM^2 Pilot, a more reasonable level of memory utilization needed to be discovered.

Using the hotline to resolve these problems was not entirely successful: it took much too long (over a month) to get queries about performance referred to someone who could ask the pertinent questions to clarify results and actually help resolve the issue. Nonetheless, through self-learning and a Blue Pacific system administrator, the missing pieces of the puzzle were discovered.

From a news item announcing an increase in paging space, it was discovered that the OS uses about 60 megabytes of "pinned" memory, that is, memory that is always required to be resident. After the determination that system daemons on batch pool nodes were using a high amount of paging space, it was increased to 1024 megabytes to provide an acceptable amount for user processes. The increase in available paging space should permit user jobs to allocate 512 megabytes (all of the physical memory on a node) with the following caveat: a job which allocates more than 452 (512-60) MB of memory is guaranteed to do some swapping. Now based on the pattern of memory access for a particular job, this paging will affect performance to some degree. For MPICTH, the performance hit was devastating. Moreover, the degradation for MPICTH occurred at considerably less than 452 MB memory allocation.

Recalling that the 2-CPU scaling test was able to run using the original memory allocation (about 120 MB per CPU), output from system monitoring utilities was examined. Table 3.3 shows the display from the jr command enabled the formulation of a question which ultimately led to better understanding of actual memory use on Blue Pacific compute nodes, for all involved including the system administrator.

----------Process Information----------+----------Node Information----------

Node Name	Process Name	CPU Time	Virtual Size	Phy Mem Tot/Used	Paging Tot/Used	/var/tmp Tot/Used	Load Avgs
blue126	Mpicth	00:02:46	118.5M	512M/9%	1024M/46%	1000M/%	1.99/1.07/0.70
	Mpicth	00:02:52	121.3M				

Table 3.3 System Monitoring Utilities Output

Now MPICTH is using 121.3+118.5=239.8 megabytes. Add the amount for "pinned" memory, 239.8+60.0=299.8 MB. The total physical memory in use is 93 percent of 512 MB, or 476.2 MB. What happened to the other 476.2-299.8=176.4 MB of memory? In this case, where memory is not oversubscribed, the system daemons which run on the compute nodes are still resident and occupy about 144 MB, as the system administrator determined by the output of two system utilities, svmon and realmem. The administrator also indicated that the parallel file system GPFS requires an additional 31 MB of "pinned" memory. So that accounts for the memory which is not in use by MPICTH, at least for the case where memory is not oversubscribed.

Given the fact the users of MPICTH want to push the limits of available memory, it is relevant to inquire what happens when a user code oversubscribes memory. As described above, system daemons use about 144 MB of memory. Pinned memory is only significant when the user starts approaching the limit of real memory. Daemons will be swapped out as needed to make room for the code, with the exception of pinned memory. Pinned memory is never available for paging. It was conjectured that user code should be able to allocate about 420 megabytes of memory before paging of user code occurs. Up to that point the user code should behave reasonably well.

There are at least three utilities available to the user which indicate memory usage: "jr", "ps acux", and "ps -eo command,vsz". All these utilities report different numbers for a job's memory usage, but each is useful in its own way. Fortunately, MPICTH logs the size of its memory request so a hard number was available for comparison. The jr virtual size report seems to be the least accurate. The output from "ps -eo" is the closest to the memory allocation printed in the MPICTH log file. The SZ column in the "ps acux" output includes memory that is preallocated by Fortran as well as the additional memory which is dynamically allocated by MPICTH. In this sense, "ps acux" is the most accurate; it is also the most useful because the RSS column reports the resident set size, indicating the degree to which user data has been paged out.

Using the virtual size reported by "ps -eo", it was discovered that allocation beyond the 420 megabyte mark suffered very noticeable degradation in performance. During the experiments with varying memory allocated by MPICTH, it was learned about the startup phase of MPICTH. Those details are omitted from this report. It is sufficient to note that a small amount of paging (in the range of 2-3 percent) does not seem to cause much deterioration in performance. Future work should include a careful study of the paging/performance issue.

Having resolved the issues of CPU utilization in batch mode, file I/O by many nodes, and allocation of memory to reduce the effects of paging, the scaling benchmark was ready to begin. The problem which is solved in the set of scaling tests is called the "Two Gas Problem". Physically it is uninteresting, but it exhibits qualities which make it suitable for the purpose of timing parallel execution. The computational load is balanced across the entire mesh. The problem can be easily altered to run on a larger number of processors with only a few lines in the input deck that must be changed. Also, once a

certain threshold of processing elements has been reached, the work per PE can be held constant as the problem size increases.

A CTH log file contains a wealth of information about the execution status. One interesting statistic is the "grind time", which is given in the log as the number of zone-cycles per hour of cpu time and as the number of cpu seconds required to perform all the calculations for a zone-cycle. Without explaining the term "zone-cycle" let us simply acknowledge that the grind time can be used as a measure of the amount of work done in a fixed amount of cpu time. A decreasing grind time means improved performance. The log file also contains information about start and finish times for I/O. Table 3.4 summarizes the grind times and elapsed times for the scaling benchmark.

number of PEs	zone-cycles/ cpu hour	cpu secs/ zone-cycle	cpu seconds	Actual Seconds	problem size
1	1.0519E+08	3.4225E-05	419.010	422	3.73E+05
2	1.9071E+08	1.8877E-05	461.480	480	7.47E+05
4	3.3003E+08	1.0908E-05	533.200	738	1.49E+06
8	6.2303E+08	5.7782E-06	565.870	1032	2.99E+06
16	1.2811E+09	2.8102E-06	550.680	1039	5.97E+06
32	2.5981E+09	1.3856E-06	542.260	1032	1.19E+07
64	5.4244E+09	6.6366E-07	519.260	864	2.39E+07
128	1.0842E+10	3.3203E-07	521.640	1513	4.78E+07
256	2.1624E+10	1.6648E-07	521.320	1272	9.56E+07

Table 3.4 Two Gas Problem on 72x72x72 kernel grid per PE

Examination of the table shows that this problem exhibits linear scaling above 64 PEs. This is particularly obvious in the column which shows elapsed cpu time. This result compares well with the performance of CTH on other parallel machines, notably the Intel Teraflops computer. The alert reader will notice that the elapsed actual seconds do not exhibit linear scaling.

What explanation can be given for the anomalous real times of execution? First of all, these times are for a single set of runs. All except the 256-processor run were made on the same day, between the hours of 4 p.m. and 8 p.m. The variation in the execution times indicates that further tests are needed. Several runs of the 32-CPU problem were made and a distribution of elapsed times was observed. Based on the outcome of those runs, the tests were repeated for each number of PEs to obtain minimum, maximum, and average elapsed real times. Any system factors which might illuminate the results were noted, such as overall load and distribution of processor sets for each run. Another interesting system characteristic is activity on the various file systems where output and log data are kept.

3.4 Execution of a Real Problem

Timing studies are interesting; however, the proof of a machine lies in its capacity for solving actual problems. The problem selected for this phase of the DISCOM^2 Pilot is one which has been run previously on the Intel Teraflops computer and also on the DEC 8400 cluster at Sandia. Thus, qualitative measures of the performance of parallel CTH in a batch environment on a massively parallel machine and also in an interactive environment on a moderately parallel machine were available. Only subjective comparisons will be made about performance on this problem.

To describe briefly the question of interest, a tungsten rod of a certain length is moving at high velocity through water. The amount of damage sustained by the rod during its passage was to be ascertained. This experiment can be set up to produce files which are suitable for visualization processing. In fact, it was the desire to visualize the rod disintegration which prompted the rebuild of MPICTH to incorporate that capability.

Blue Pacific and ASCI Red limit batch processing in similar ways, that is, shorter batch time limits during the day and longer limits at nights and on weekends. Thus, running this problem in batch mode was equally frustrating on both these machines: the job had to be restarted multiple times in order to run to completion in day queues. Using 32 CPUs on Blue Pacific, the weekend queue may have allowed enough time to complete this job in a single batch submission. Because of the multiple restarts, the only Blue Pacific timing which makes sense is the one that used 128 CPUs, and that run took about seven hours to accumulate 7158 CPU seconds. By contrast, on the Sandia DEC 8400 cluster, 32 CPUs completed this problem in twelve hours.

During the course of making the batch runs for the tungsten rod problem, the I/O patterns of MPICTH were summarized. This analysis is included here because of its relevance to those who must plan for file transfer and storage. Note that actual file sizes vary according to the problem definition. It is emphasized that this description is for the particular tungsten rod input deck used for the DISCOM^2 Pilot project.

There are three types of file output produced by MPICTH: log files, restart files, and plot/visualization files. Each type has characteristic I/O patterns and uses which may determine the file system where it should be written. Log files are typically much smaller and output occurs much more frequently, and in smaller bursts, than restart files.

The logs produced by MPICTH include: progress reports to the screen, an ASCII file with information about the problem and its solution, and a binary history file which records values of interest ordered with respect to time. The batch system writes a file which captures output that is normally sent to the screen. Screen output is produced only by logical processor 0, and occurs whenever a significant event (such as a restart dump) takes place. Log files receive much more frequent output and only logical processor 0 produces output to these files. The NFS-mounted file system was used for these files since only one processor accesses each of them.

The ASCII file contains a record of certain key values for every problem cycle, independent of user control. Logical processor 0 writes out only data values which exist on that PE. As an example of the amount of data written to the ASCII log file, a 32-CPU run of tungsten rod problem produced about one thousand 132-column lines per hour. The history file records data based on user request; frequency of output to this file is under user control. Although only logical PE 0 writes to the history file, information is collected from all the processors; thus history file I/O must wait for communication which has the character of a reduction operation. For the same 32-CPU run using the tungsten rod input deck, the history file received about 170 Mbytes per hour.

There are some rather large files created for restart output. Restart dumps are of two types, but both types are designed to hold enough information about solution status so that they can be used to restart the job. The frequency with which these files are written is part of the input supplied by the user. Backup restart files are intended to provide the capability of restart in case of abnormal termination of MPICTH. Normal restart files can be plotted by the CTHPLT utility. For both types of restart file, the size depends only on problem size: the mesh on which the problem is defined and the number of variables to be solved for. Each PE writes a restart file, and individual file sizes depend on activity within the problem region assigned to that PE.

Visualization files and their close relatives, the plot files, are also problem size dependent. These files can grow to enormous size, depending on what variables the user wants to view and on the frequency with which the user plots these values. As with restart files, each PE writes a visualization file with its own data. Both restart and visualization files receive large bursts of output and are not touched between these bursts. The size of a burst can be several megabytes. Since every PE writes restart files and visualization files, the GPFS parallel file system was used for these types of files. There is an I/O library for MPI codes which use the GPFS; however, on Blue Pacific MPICTH relies on Fortran I/O and does not make use of this library.

3.5 Summary and Conclusions

Running a code on Blue Pacific takes some adjustment on the part of the user, but help is readily available in the form of on-line documents and the Livermore Computing Hotline.

The current configuration of the compute nodes does not provide enough memory to accommodate the large jobs which are prevalent on ASCI Red. However, the memory is scheduled for upgrade to one gigabyte per node, and that increase should improve the computing capability for MPICTH.

Files produced by MPICTH exhibit large variability in size and frequency of the write operation. Further statistical information should be gathered before attempting to resolve

questions about the "best" file system for the various files generated and used by MPICTH.

4. Experiences with Coyote on Blue Pacific

Coyote was the first application to demonstrate the ability to build and test an executable on the (4 PE) F50 cluster, LaPorte, at SNL in Albuquerque, New Mexico (SNL-NM) and then run that SNL production code on the Blue Pacific platform at LLNL in Livermore, California. The production problem that was selected and run was the "4277 Header Subassembly Simulation" for the neutron generator furnace. The problem ran on 256 PEs on Blue Pacific TR Platform. For this problem, the aggregate size of the input files was 15 Mbytes. After the run, the results of this simulation were brought back to SNL-NM and visualized on SNL's large visualization server, Tesla using the MUSTAFA tool. The size of the file that was shipped to the visualization machine was 43 Mbytes.

The breakdown of activities will be discussed. The Coyote executable was built on one of the local F50 LaPorte machines in Albuquerque. Then the executable image and the input data (total about 20 Mbytes) were moved to Blue Pacific using FTP.

Using ssh, the loadbal script was run interactively to execute the CHACO load balance program, and nem_spread to create exodus files for input to Coyote. This activity took about 20 minutes. Files were created for a 256 PE run.

A script was then submitted to run the neutron generator furnace problem over the weekend. The code was run for the maximum allowed amount of time (12 hours) on 256 PEs. During this time, 24 steps were taken by the code. This compares with a run time of about 25 minutes on 64 PEs of the Janus, the ASCI Red Platform, to run 20 steps. The gross difference in execution speed appears to be caused by poor efficiency in the communication phase using MPI on Blue Pacific. The reason for this is unclear at the present time, but will be investigated further.

Following execution of Coyote, the 256 exodus output files were interactively combined using nem_join on Blue Pacific to produce a 43 Megabyte graphics output file in about 10 minutes. This was moved to the visualization server (Tesla) in Albuquerque, and visualized using MUSTAFA.

Overall, the amount of time spent moving data between sites was a small part of the total. The logistics will be simplified somewhat when Coyote is migrated from Exodus to PDS file formats. Then there will no longer be a separate file for each PE. This conversion is in progress. Using DFS rather than FTP would simplify things quite a lot as well, particularly if Albuquerque based files can be used directly by the running application in Livermore. The I/O requirements for this code are relatively low, so this should be practical with minimal performance impact. The upgrade to PDS should be completed before this is attempted due both to efficiency and logistic reasons.

5. Pilot Services

5.1 What was learned

The pilot showed that a "proxy" environment of the remote resource could be established cheaply in the local network. The proxy environment could also be made to emulate the software environment of the larger system. For example, SNL's two IBM RS/6000 F50 systems supported the same compilers and MPI libraries as the Blue Pacific SP2 systems.

These local resources could be used to port code and run small debug problems prior to interacting with the larger remote system. This will reduce interactive load on the remote resource and allow some level of local prioritization and experimentation. Local systems would also provide tighter controls on source code. It was established that binaries built locally could run properly at the remote site on the larger system.

The system administrators of the local resource could provide a first level of support to the customers, effectively reducing the occurrence of questions to the remote resource help desk.

Integration of the AIX DFS client into the SNL DCE/DFS environment was demonstrated. Several code groups intend to use DFS as the repository for their source trees and as a distribution mechanism for binaries. Also, the LoadLeveler software package was installed. This batch control system can be configured to submit jobs to the large remote resource. Consistency within the environments may require additional software for this task.

During the pilot IBM ATM interfaces and software drivers presented some implementation challenges. Also, ordering software products from IBM was found to be a difficult task.

5.2 What's next

In FY99 the local proxy system and the large remote system will be configured so that batch jobs can be submitted from Sandia to the Blue Pacific system. As well as explore further integration of local resource into the network of the remote resource

There will continue to be a push for the completion of cross cell trust in the DCE environment. This will allow for the access of input files for a Blue Pacific system run from the Sandia DFS name space. And allow output files from a Blue Pacific system run to be delivered to the Sandia DFS name space. In addition MPI-IO techniques to conduct I/O activity locally while the simulation runs remotely will be investigated.

Visualization testing will continue—simulation results will be visualized at both the remote site (LLNL) and local site (SNL-NM) and comparisons of qualitative and quantitative behavior will be done.

Additionally, test runs on Blue Pacific will be expanded to include more code groups.

6. Security Architecture

A common, or at least compatible, security architecture is critical to the success of distance computing, especially where inconsistencies among the participating sites' security policies can lead to poor end-to-end user performance. The focus of security architecture activities during this pilot was to become familiar with the internal computer security policies at LLNL, SNL and LANL, with special emphasis on where inconsistencies could lead to problems. Fundamentally, each of the participating sites is properly concerned with computer security in general and Need-to-Know protections for sensitive data in particular.

The DCE security model that has been adopted by the ASCI program is making great progress in addressing potential security inconsistencies. DISCOM^2 should follow the ASCI lead in this area and leverage its activities. In addition, the SECURENET community has been in operation for three years and is in the process of refining its security requirements based upon what it has learned during this period. Leveraging DISCOM^2 against SECURENET is appropriate whenever DISCOM^2 includes classified computing resources.

7. Blue Pacific and LLNL Interactions

7.1 FY98 LLNL DISCOM² Distance Computing Accomplishments

A collaborative effort between LLNL and SNL-NM investigated wavelet compression. The LLNL effort focused on lossless compression applied to images and data. The results of this effort advanced LLNL's conceptual and mathematical understanding of wavelet compression. A survey was started of existing development efforts, including the identification of several commercial and public domain software packages supporting the exploration of wavelet compression techniques. This effort is expected to continue in FY99 with some demonstrable deliverables.

A contract with Bellcore was recently signed that will be liened against FY98 funds. This contract will contribute to many aspects of Distance Computing, but particularly the modeling effort.

Network management efforts provided Web-accessible traffic information on LLNL routers and the ESNET router. There is also information on router CPU, memory, and bandwidth utilization.

Network support was provided to monitor tests and otherwise enable and participate in network performance efforts.

Consultant services were enhanced to support remote users. This included phone support and enhanced Web pages for the SNL DISCOM² staff and the users.

System administration support was provided to discuss planning of the Pilot and to contribute to the Pilot efforts. System administration personnel were also involved with resolving some of the issues presented by the SNL users. The system administration and network staff also participated in the remote visualization demonstration.

7.2 Proposal For The Use Of LLNL's Development SP2 Platform

The DISCOM team submitted a few proposals to LLNL Blue Pacific development team to use their development platform, Baby, for some experiments. Baby, a small SP2 platform is the development platform for the Blue Pacific system managers. This machine is heavily utilized by the LLNL staff. Because the DISCOM team was unable to secure the resource for our experiments, we performed the initial deployment of security and file services on the SP2 proxy machine. Interoperability tests of the service required that the DISCOM team become users of the large resource to verify that services were operational. The DISCOM team would still like to get access to the development platform to do the cross-site testing and for testing that the proxies are physically unable to do now.

The proposed tests follow.
Cross-Site MPI Tests: A series of MPI transport experiments between the SP2 development machine and the two DISCOM^2 Pilot LaPorte Servers. These experiments will determine how an application could make use of direct MPI interaction across the WAN. The experiment will determine the level of performance needed for effective application usage and to determine the bottlenecks in the current system.

Cross-Site Security Tests: A series of tests to demonstrate the feasibility and advantages of the proposed changes to the current security architecture. This includes WAN as a wire, joint addressing space and transparent cross-cell trust.

SP2 Router Performance Tests: To test the capability of the SP2 router. These experiments will determine the performance of the Ascend router and the system performance of the SP2s with the router in the communication system.

8. Remote Visualization

8.1 Introduction

Wide area bandwidth is relatively expensive compared to a similar amount of bandwidth in a computer room. Therefore, it is less expensive to move files from a supercomputer to local storage than to move files to a remote file server. This has interesting implications for remote users who want to run simulation jobs and post-process and visualize the results. Assuming that the output files from a simulation run are stored locally, it is best that the visualization server that supports remote users' visualization sessions be located at the local site as well. This is particularly true if the user wants to take a quick look at the simulation results to determine if the job is running correctly, and if any input parameters need changing. However, if the visualization server is located at the local site, then some mechanism must be provided to allow visualization output to be displayed at the remote user's console. This paper describes one such approach that was implemented in the FY 98 Distance Computing Pilot, with the local site being LLNL, and the remote site being SNL. This report describes the approach in detail, the requirements it addresses, and performance results that were observed.

8.2 The Problem of Remote Visualization

As part of the U.S. Department of Energy's (DOE) Accelerated Strategic Computing Initiative (ASCI), increasingly powerful supercomputers will be delivered to each of the DOE weapons laboratories over the next several years. As these machines are delivered at each site, users at the other sites will want to remotely use these resources to perform increasingly detailed simulations. Therefore, the DOE has initiated the DISCOM^2 program, which is chartered to deliver solutions that allow institutions (and its users) to access and use computational resources that are located at other laboratories.

One of the more difficult problems in DISCOM^2 is remote visualization, where a remote analyst uses visualization resources located at the same site as the supercomputer that ran the simulation. (Note: in this paper, the term "local" refers to users and resources that are located at the same site as the supercomputer that runs the simulation job, and the term "remote" refers to users and resources that are located at other sites.) Remote visualization is often required because simulation jobs typically generate more than 1 GB (Gigabyte) of output data that can be transferred relatively quickly to local storage, but can take a long time to transfer over a wide area network to a remote user. However, with the data stored locally, careful design of the remote visualization system is required to ensure that the remote user's interaction with the system is not overly degraded (in terms of delay, image resolution, complexity of operations, etc.) from the case where he/she is interacting with the data locally.

This report describes a system that was designed to help solve this problem. This system places the visualization server at the local site with the simulation supercomputer, and

uses video compression equipment to provide a "remote video head", i.e., a video stream that displays the visualization images rendered on the local visualization server to the remote user. As this report describes, this system is well-suited for the analyst who wishes to take a quick look at the result of the simulation run to determine whether it ran correctly, and is worth a closer look.

This report is organized as follows: Section 8.3 describes an number of alternative architectures for remote visualization, Section 8.4 describes the general approach for the "remote video head" option for remote visualization. Specific network architectures and test results for ATM-based and IP-based remote video head configurations are described in Sections 8.5 and 8.6. Finally, Section 8.7 provides some concluding remarks and directions for future work.

8.3 Remote Visualization Alternatives

8.3.1 Requirements

The primary goal of remote visualization is to provide a mechanism that hides the fact to the user that he is interacting with machines and/or data that are physically distant from his office. Concealing this fact is a difficult task for the system designer because increasing distance results in increasing end-to-end packet transfer delay, which may result in noticeable delays to the end user. Furthermore, wide area network bandwidth is lower than LAN bandwidth (due to cost), which leads to increased delay. Finally, WAN connections are often highly aggregated (shared) among many users who have many other reasons for using WAN-connected resources (and may not be willing to relinquish their "fair share" of bandwidth for a single user's desire to perform remote visualization). Therefore, the secondary goal of such a system is to be a "good citizen" with respect to other users of the networks that are used by the system.

The two broad goals stated above can be broken-down into the following specific requirements:

Req. 1 - Low Delay between Simulation Completion and Start of Visualization
This is an issue of productivity for the analyst–time spent transferring files and doing other functions after the simulation run is completed, but prior to the start of output interpretation is time spent away from the task. By reducing this time, more simulation runs can be attempted in a given period of time, leading to more efficient use of computing and human resources, and faster determination of results.

Req. 2 - Low End-to-End Delay
Visualization, whether performed remotely or locally, is a highly interactive task because the user is frequently manipulating objects to view them from different perspectives. As with other interactive tasks (e.g., telephony conversations, video conferencing, distributed

game playing, etc.), round-trip delays in excess of 250 milliseconds can become bothersome to the user.

Req. 3 - Low Bandwidth Requirements
As stated earlier, the goal of the DISCOM^2 project is to provide institutions remote access to computing resources. However, institutional connections to wide area networks are often used for more purposes than for distance computing tasks such as remote visualization. Furthermore, these connections may be used by a large number of end users. As a result, remote visualization users must not generate excessive amounts of traffic, otherwise, all users' data flows and applications (including the remote visualization user's) will suffer excessive performance degradation.

Req. 4 - Good Image Resolution
The visualization analyst is often highly trained, and subtle details in the visualization output can lead to significant insights. If the remote visualization solution results in excessive loss of image resolution, then such details may be missed.

Req. 5 - Minimal Impact on Existing Visualization Applications
A variety of tools are being used by analysts today. Paying a myriad of visualization software vendors to make substantial modifications to their products to support remote visualization scenarios is expensive, and not practical.

Req. 6 - Minimal Impact on Existing Visualization Resources
To be cost-effective, a remote visualization solution would ideally involve no added hardware, and no modifications to existing resources. If additional hardware is needed, it should be commercially-available at a low cost, and easily integrated with existing visualization hardware and networks.

Req. 7 - Ability to Access Locally-Stored Data
Throughout this report, it is assumed that the available bandwidth for the movement of simulation data is greatest between resources that are collocated. This assumption implies a preference to move simulation output from the supercomputer to some local data storage server, which must be subsequently accessed by visualization software for analysis.

8.3.2 Architectural Alternatives

In a previous report, a basic taxonomy of alternatives for remote visualization alternatives in a pilot distance computing system was presented [1]. At the topmost level of this taxonomy, remote visualization alternatives are classified into two alternatives – locate the visualization server at the local site, near the simulation machine, and locate the visualization server at the remote site, near the user. When the visualization server is located at the remote site, the user is assumed to have direct access to the server's display console, and the simulation output files must somehow be made accessible to the

visualization server. In this case, implementation options using FTP or networked file systems (e.g., NFS, DFS, AFS, etc.) can be considered. When the visualization server is located at the local site, the server is assumed to have direct access to the files, and the display console (or frame buffer) must be made available to the remote user. In this case, remote video head (e.g., transport of bitmaps through the X Windows protocol, SGI's networked dual-headed software daemon (NDSD), or networked video streams) or remote frame buffer techniques (e.g., SGI's graphic rendering service (GLR) or direct frame buffer transport) must be employed.

Although a thorough consideration of alternatives to remote visualization is a useful exercise, the purpose of the previous report[1] was to describe a solution that could be implemented quickly for short-term delivery of the DISCOM^2 pilot. The technique that was advocated in that report used a visualization server based at the local site, and the use of networked video distribution to provide a remote video head to a remote user. The following section describes this architecture in more detail.

8.4 Remote Visualization using Networked Video Distribution

As stated earlier, when the visualization service is collocated with the simulation machine, a mechanism is required to implement either a remote video head or a remote frame buffer capability to provide the visualization output to the remote user. The mechanism that was selected for the DISCOM^2 pilot was based on the remote video head concept. This approach was chosen because it was immediately available using off the shelf components, and because it best addressed the requirements stated earlier. Specifically, this approach addressed these requirements as follows:

Req. 1 - Low Delay between Simulation Completion and Start of Visualization
Since this mechanism used data that is stored locally to both the simulation machine and the visualization server, no delay is incurred in transferring this data over a wide-area link that is presumed to be slower, and shared with more users.

Req. 2 - Low End-to-End Delay
The remote video head approach that was chosen used a low-delay video compression algorithm which, when added to the round trip ping delay of the wide area network, incurred a delay of less than 100 ms.

Req. 3 - Low Bandwidth Requirements
The video compression algorithm and hardware that was selected for this approach implements a video stream in under 6 Mbps (Mega bits per second), which is 1/22 of the available wide area link capacity of 132 Mbps.

Req. 4 - Good Image Resolution
Although video compression was used in this implementation, good video quality (near broadcast quality video) is achieved at 6 Mbps. Initial feedback from prospective users

provided a similar impression of video quality. Although, there are some users who require higher fidelity, there are many who see a need for a video quality capability.

Req. 5 - Minimal Impact on Existing Visualization Applications
The approach that was selected for the DISCOM^2 Pilot required the visualization application to be modified to separate user controls (which were carried from client to server using the standard X Windows protocol) from visualization output (which was carried from server to client using compressed network video).

Req. 6 - Minimal Impact on Existing Visualization Resources
Additional hardware is required to implement the DISCOM^2 Pilot solution, but the hardware is commercially available and easily integrated with the existing network and visualization machines. The cost of the additional hardware to implement this approach is less than $20K.

Req. 7 - Ability to Access Locally-Stored Data
The DISCOM^2 Pilot solution used simulation output data that was locally stored.

8.4.1 Visualization Network Architecture

The generic architecture for the remote video head solution using compressed networked video is shown in Figure 8.1.

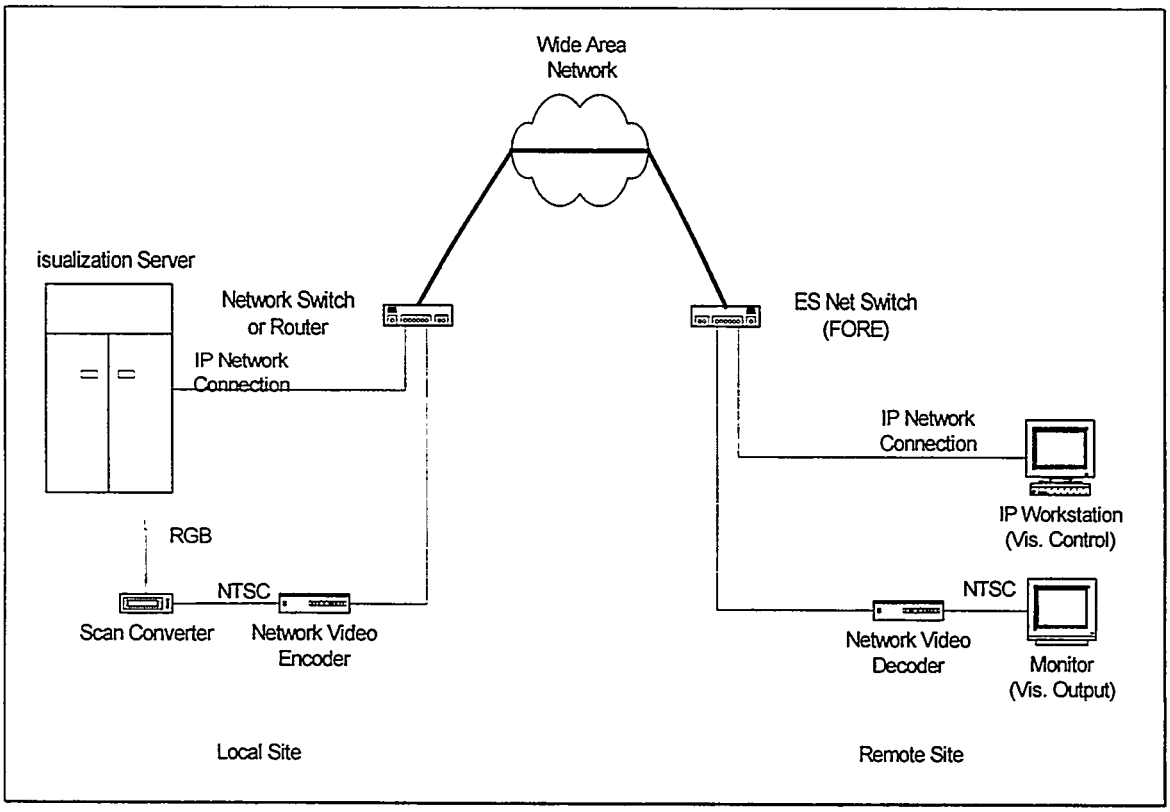

Figure 8.8.1 General Network Architecture

In this implementation, the remote video head is provided by first enabling the visualization server's "video pipeline" to take the server's frame buffer and display it on an available RGB output. Next, this RGB output is converted to a composite, NTSC-compatible signal that is suitable for capture, compression, and network transmission by the network video encoder. Once converted to the proper network format (e.g., IP frames and/or ATM cells), the network video is switched and routed over the wide area network to the remote site. When the frames/cells arrive at the network video decoder, the information is de-compressed, converted to NTSC composite signal, and displayed on a television monitor. Alternatively, the NTSC signal can be captured by a video capture board, and displayed on an RGB monitor.

The remote user interacts with the visualization software running on the local visualization server using the standard X Windows protocol over an IP path.

8.4.2 Issues

In this architecture, resolution loss occurs in four processes:

1. Digital to analog conversion from the frame buffer to the RGB output,
2. Down-sampling losses from the frame buffer to the RGB output,
3. Scan conversion loss, and
4. Compression loss

The first opportunity for resolution loss occurs when the digital frame buffer contents are converted to analog RGB signals. Although the amount of loss in this step is very low, the analog connection between the visualization server and the scan converter introduces a potential for analog noised to be introduced, particularly if the cable is very long. Ideally, a low-loss remote visualization system would transfer the contents of the frame buffer directly to the remote user. This type of solution, called a "remote frame buffer" capability, is currently implemented in SGI's GLR graphic rendering service, and other implementations are also being studied.

The second resolution loss opportunity occurs when the entire frame buffer (typically 1280 x 1024 pixels) is converted to the 640 x 512 resolution of NTSC. When this "down sampling" occurs, many pixels are discarded, which directly leads to resolution loss. This effect can be mitigated by taking, for example, a 1280 x 1024 RGB signal, breaking it into four 640 x 512 quadrants, independently transmitting the four video quadrants, and re-combining at the remote user's end. This approach has been prototyped in the lab, and shows much promise.

The third type of resolution loss occurs when the 640 x 512 RGB signal is scan-converted to NTSC. The degree of loss is partly fixed due to the conversion from progressive scan

to interlaced video, and partly variable, depending on the quality of the scan converter that is being used. Currently, a variety of commercially available scan converters are being evaluated to determine the degree of loss versus cost. In addition, techniques for directly compressing an RGB signal are also being studied.

Finally, the compression algorithm that is used by the video encoders and decoders introduces significant resolution loss. The degree of resolution loss for a given bandwidth target is dependent on the compression algorithm, and currently, a number of compression algorithms are being studied to determine which one provides the best bandwidth vs. video quality tradeoff. Algorithms currently under study include Motion JPEG, MPEG, and wavelet compression (which will be implemented in the upcoming JPEG-2000 standard).

8.5 Performance Observations and Considerations–Video over ATM

The demonstration system that was integrated for the FY98 DISCOM^2 pilot used ATM video encoders, and the standard scan-converter provided with the SGI Onyx2 system. This section provides a summary of tests that were performed on this system and an identically configured laboratory system. Additional details on video over ATM testing can be found in [2].

8.5.1 Test Setup

The test setup and DISCOM^2 pilot architecture for the remote visualization system using video over ATM is shown in Figure 8.2.

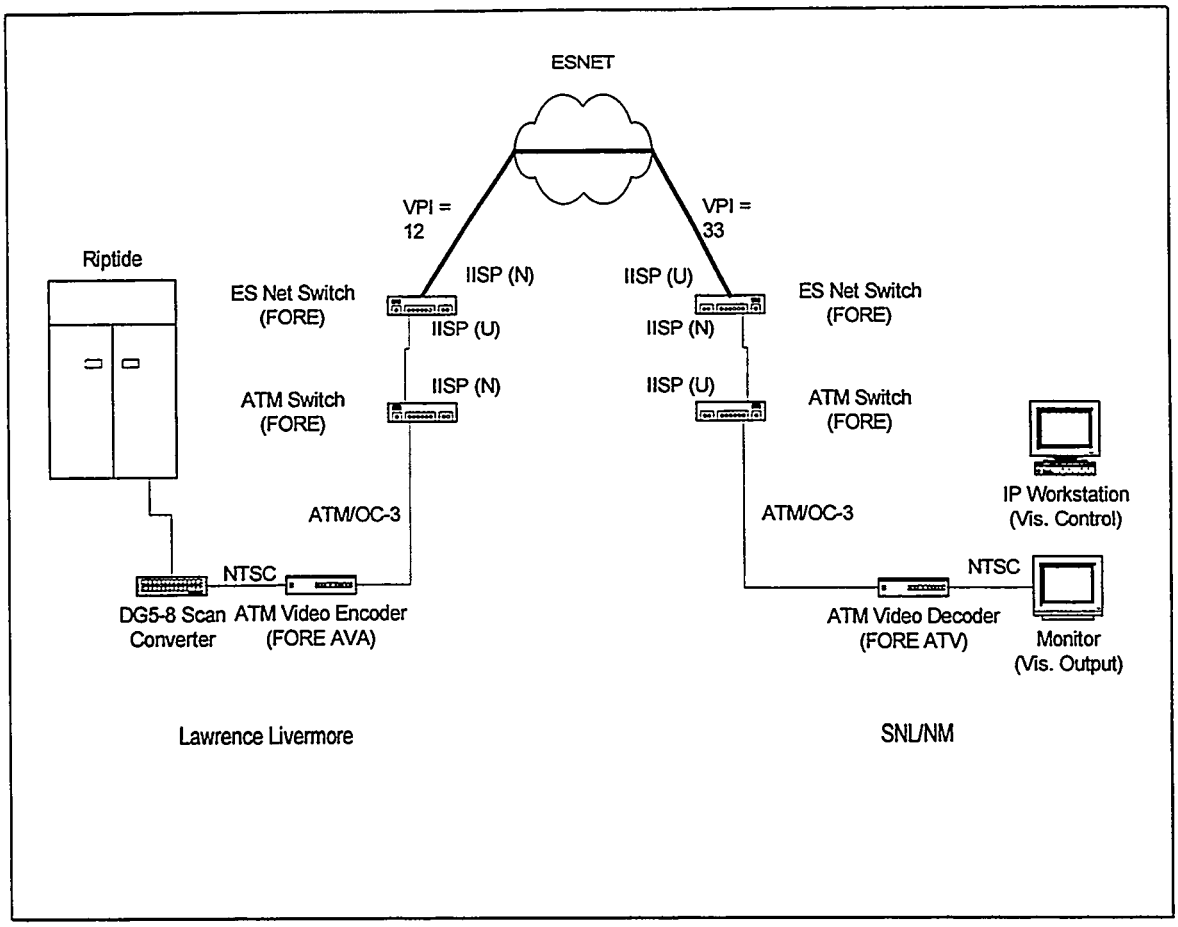

Figure 8.2 Remote Visualization Architecture and ATM Test Setup

As stated earlier, the primary characteristics of this system is its use of the built-in scan converter on the LLNL SGI Onyx2 machine, and its use of the Fore Systems ATM video compression/decompression equipment. ESNET provided wide area network connectivity via switched virtual circuit tunneling through the ESNET PVC mesh. The network configuration that was used for laboratory testing was practically identical, except for the use of two DS-3 ATM ports on one of the switches to allow virtual circuits to be routed through a DS-3 delay/error simulator.

8.5.2 Performance Results

A variety of performance results from this system are described in detail in [2], and are summarized here. Four aspects of performance were studied: forward (video) bandwidth requirements, backward (X windows control) bandwidth requirements, degradation of interactivity as delay increases, and sensitivity of video to transmission bit errors.

The most interesting performance results were obtained when measurements of the video quality and bandwidth requirements were taken when the level of compression was varied. The results of these measurements are summarized in Table 8.1.

Q Factor	Forward Bandwidth (Mbit/sec)	JPEG Compression Ratio	Perceived Image Quality
Variable	5.600	42:1	good
16	5.600	42:1	good
20	4.800	49:1	good
32	4.400	54:1	good
64	4.000	59:1	good
128	3.200	74:1	minor "blockiness"
256	2.800	84:1	poor image
512	2.400	98:1	extremely poor image

Table 8.8.1 Bandwidth and Video Quality vs. Q Factor

The independent parameter in these tests is the "Q Factor", which is a JPEG compression parameter that is used to adjust the compression ratio. As indicated in Table 8.1 as the JPEG Q factor increases, the amount of compression increases, and the bandwidth requirement and video quality decreases. In addition, the Fore Systems video compression/decompression units support a "Variable Q Factor" setting. This setting allows one to specify the maximum amount of bandwidth consumed by the video stream, and the compression/decompression units adjust the compression Q factor so that the output video rate stays within the bandwidth target. However, since video quality is an important requirement for remote visualization, the highest-quality setting (Q factor = 16) was used in subsequent tests and demonstrations. Figure 8.3 shows the relative quality of uncompressed and compressed visualization images for a Q factor of 16.

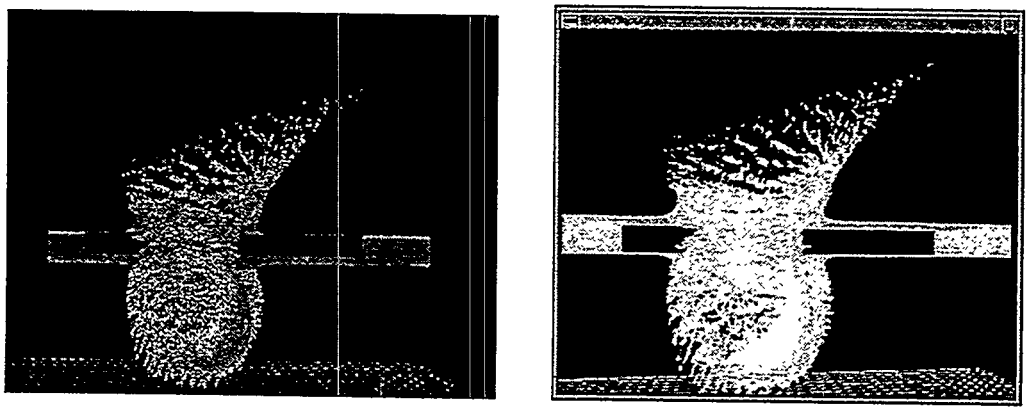

Figure 8.3 Quality of Uncompressed (left) and Compressed Images

In addition to the forward bandwidth tests, the reverse bandwidth that is used by the X windows controls was also measured. These measurements showed a required reverse bandwidth of 40 Kbps, which indicates that any IP connection (even dial-up!) is sufficient to handle the visualization control traffic.

Since this system will be used over a wide area network, system performance under added delay and bit error conditions is a special consideration. With the delay sensitivity and bit error tests that were conducted in the laboratory, the video and X windows virtual circuits were routed through the DS-3 delay/error simulator. In the delay tests, the delay simulator was adjusted between 0 and 80 milliseconds, and the sum of this delay and the compression/decompression delays were evaluated. The results of the delay tests show that for the maximum value of simulated delay (80 ms), the system delay did not pose any problems in interacting with visualized objects. Likewise, the bit error tests were conducted by adjusting the error simulator between error rates of 0 and 10^{-8} bit errors per second. The worst-case test of 10^{-8} showed minor glitches in the video output once every ~15 seconds.

8.5.3 Discussion

The tests described above show generally acceptable performance in terms of video quality, bandwidth requirements, and robustness in high-delay, high-error rate environments. However, additional tests (described in [2]) show that video performance degrades spectacularly when network congestion occurs. This is true because video decompression is sensitive to errors and losses in the compressed video stream, and this is particularly true for packet and cell losses. Since network video over ATM uses a non-assured delivery protocol (specifically, Unspecified Bit Rate (UBR) AAL 5 connections), network congestion and associated cell loss is not corrected by a higher layer protocol. In order to increase the probability of successful video transmission over congested links, ATM video cells must be given higher priority over other ATM virtual circuits.

8.6 Performance Observations and Considerations–Video over IP

Although ATM is better suited for the transport of real-time video, video can also be transported over IP networks. Video over IP is often desired because it provides media-independence and more flexibility. For this reason, there is some interest in implementing this system over arbitrary IP networks. This section describes a similar set of tests that were performed to determine the suitability of Motion JPEG compression over IP.

8.6.1 Test Setup and Procedures

The test setup for remote visualization over IP is shown in Figure 8.4.

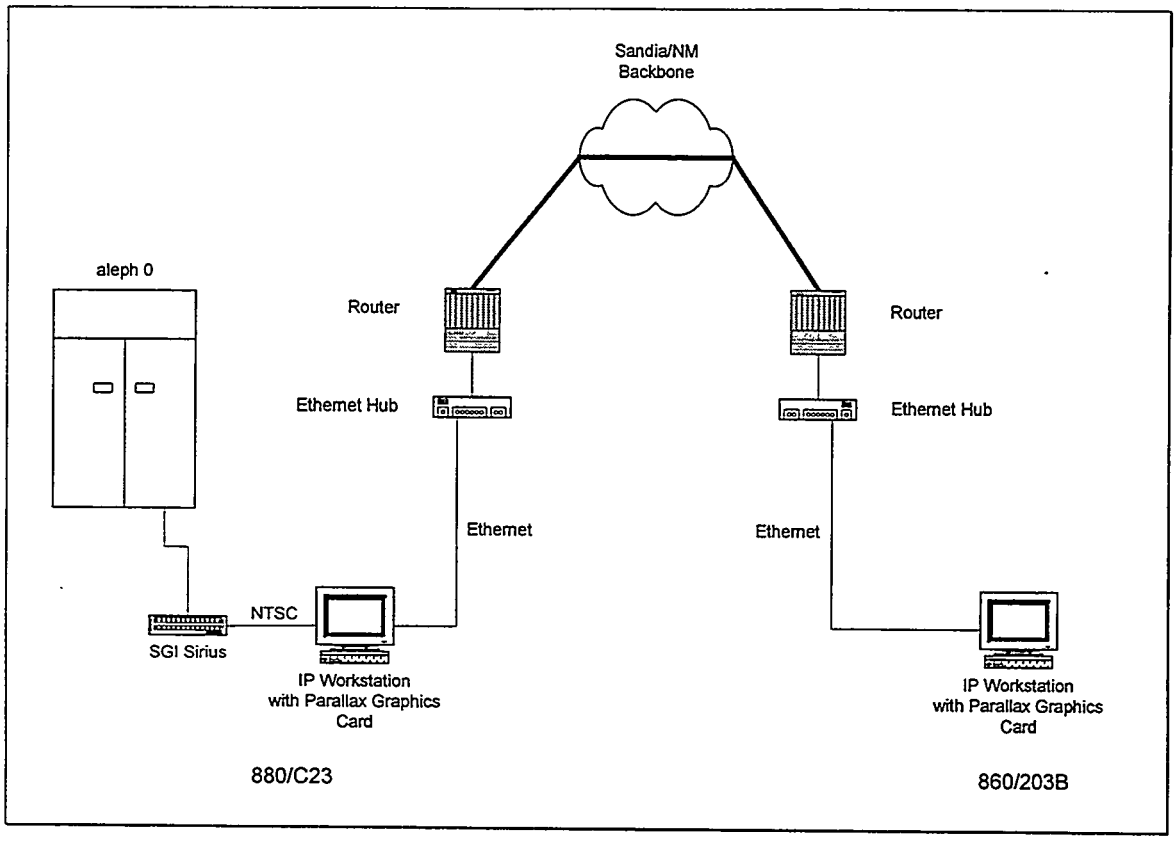

Figure 8.4 Remote Visualization over IP - Test Setup

In this test setup, the RGB to NTSC conversion is performed as in the case for ATM video. However, the NTSC video is captured and compressed using a Sun Ultra 1 workstation with a Parallax Graphics XVideo interface card running the "tcomp" video distribution program provided in the Parallax driver distribution. Once the video is compressed, it is packed into IP packets and routed across Sandia's internal network to the remote user. At the user's end, the IP packets are de-compressed using another Parallax Graphics board and displayed on the user's workstation along with the X Windows visualization controls.

8.6.2 Performance Results

The performance results that were observed with the video over IP implementation were similar to those observed for video over ATM. The bandwidth for a Motion JPEG Q factor of 16 was measured at 4.5 Mbps using the same visualization data. However, since software is involved in the video-to-IP interface, some amount of CPU overhead is incurred. The CPU overhead in this test was measured at 10 % on the sending side, and 20 % on the receiving side. Both machines used in this test were SPARC Ultra 1 with 170 MHz processors, and running Solaris 2.5.1.

8.6.3 Discussion

This appears to be a viable solution for remote visualization over IP networks. Although expensive workstations and interface cards are required to implement this option, the workstations may be used for purposes in addition to video capture and display (e.g., as the user's control workstation for interacting with the application running on the visualization server). Unfortunately, as this article was being written, Parallax Graphics announced that it will be closing its sales, engineering, technical support, and manufacturing operations [3]. Therefore, other sources will need to be identified to implement this approach.

As with the video over ATM approach, prioritization of video over IP packets is also required. This is true because video over IP is implemented using UDP, a non-assured packet delivery protocol. Recent efforts in the Internet Engineering Task Force's (IETF) Differentiated Services (diffserv) and Integrated Services (intserv) working groups have progressed IP standards development in the implementation of IP packet prioritization.

8.7 Future Directions

The FY 98 DISCOM^2 remote visualization efforts have focused on the implementation and evaluation of a remote visualization solution based on network video transport technologies. Although this was the focus of the work, it is certainly not the only mechanism that can implement the remote visualization service. Other options were briefly stated in this report, and in [1]. The DISCOM^2 work in FY99 should be structured so that a more thorough evaluation of other mechanisms is performed.

Once these mechanisms are identified, the scenarios for their use should be determined so that the requirements listed in this report are well-balanced for each usage scenario. For example, in a scenario where a user has run a simulation job, and wants to take a "quick look" at the visualization output, using the networked video solution described in this report may be optimal. However, if the remote user wishes to carefully examine the graphical output of the simulation, then some other remote visualization approach should be considered. Certainly, other applications scenarios exist as well, and these should be enumerated in the FY 99 activities.

8.7.1 Coordination with ASCI Visualization Efforts

In developing a list of application scenarios and implementation options, it is important to coordinate the DISCOM^2 activities with those in related programs, and ASCI in particular. The ASCI developers have spent much time thinking about this problem, and likely have additional implementation insights and usage scenarios that must be accounted for in the DISCOM^2 study.

8.7.2 Evaluation of Non-Compressed Approaches

In his analysis of remote visualization scenarios, Dave Wiltzius of LLNL describes a number of alternatives for transporting post-processing flows across the WAN [4]. These

options are practical when the available intersite, single session bandwidth exceeds the SONET OC12c (622 Mbs) rate. The most promising alternatives is to implement the entire visualization pipeline at the local site, and transfer frame datasets (screen bitmaps) to the remote user in real time. This option induces the least amount of delay, and allows for remote interaction with the visualized images. Although this option requires a lot of dedicated bandwidth, it may be appropriate under certain usage scenarios (most importantly, the single user scenario).

In addition, Wiltzius also describes other data transfer options that use global disks and/or mass storage devices at the remote location. These devices can be made available to local resources via DFS to allow transfer during simulation or post-processing. The bandwidth that is required for this approach varies, depending on the amount of data that is produced, and the length of the simulation run [4]. However, in certain scenarios, using DFS to transfer simulation data to the remote user for post processing may be the right choice. The tradeoff will require further study.

8.7.3 Improvements on Networked Video Approach

There are a number of technical areas that can benefit from additional study to determine how the remote visualization approach described here can be improved. As stated earlier, there are a number of opportunities for resolution loss in this system. Since this falls short on the image resolution requirement, this area should receive the most attention.

The ASCI program has developed a compression research program that is looking at advanced topics in video and ATM cell compression. Advanced video compression techniques that deserve further consideration include wavelet compression and the JPEG-2000 standard (which uses wavelet compression), variable compression within video frames (to give certain regions higher resolution over other regions). In addition, techniques to compress the contents of ATM cells in a lossless manner should be considered as well (to provide the ATM equivalent of a compressed dial-up connection).

In addition, alternative system design techniques should be considered to remove additional losses in a video distribution system. Examples of efforts in this area include the compression and transport of individual quadrants in a 1280 x 1024 display, the use of digital video standards to remove analog to digital conversion losses, the evaluation of high-quality scan converters, and the implementation of a remote frame buffer option. This last option requires the development of a board to capture, compress, and transfer frame buffer contents directly to a remote user, rather than using a number of conversion steps.

8.8 Summary/Conclusions

Assuming that more bandwidth exists for a given cost between local resources (e.g., supercomputers, file storage and visualization servers) than over the wide area network, then the task of transferring large output files across the wide area to remote user

becomes prohibitively expensive (in terms of dollars and/or time). Therefore, it is more cost effective to leave large output files at the local site, and implement some sort of mechanism for remote users to post-process and visualize the output.

If the output files remain at the local site, then there are two choices for implementing remote visualization: locate the visualization server at the remote site, and provide remote access to the local files (e.g., using DFS), or locate the visualization server at the local site, and provide a remote video display capability to the remote user. The first option is difficult for remote visualization because of the assumption that wide area bandwidth is expensive, and hence, restricted. Transporting raw visualization files to a visualization server using DFS will result in large delays while these files are read through the restricted network "pipe".

Displaying visualization output from a local visualization server to a remote user is also problematic because full-rate motion video requires 250 – 1 Gbps of bandwidth, which is expensive to obtain. However, video compression can help alleviate this problem. The tradeoff, however, is loss of output resolution. Nevertheless, this sort of solution provides the user with a fully interactive remote visualization environment, and provides sufficient resolution to allow the analyst to make rough decisions about the results of the simulation run (e.g., "Did the simulation run correctly?").

The remote visualization approach described in this document, and implemented in the DISCOM^2 SP2 Pilot uses the remote video display option. This option was implemented using video compression equipment for ATM networks. In addition, an option that implements compressed video transport over IP networks was also tested. Both systems showed very good interactivity, acceptable image quality, and low bandwidth requirements (under 6 Mbps).

8.9 References

1. T. D. Tarman, *Remote Visualization in the DisCom SP2 Pilot*, Internal Project Report, July, 1998.
2. L. Stans, editor, *Lockheed Martin Management Data Systems (M&DS) Sandia National Laboratories Technical Report, IWTA #RRM066053*, Sandia Technical Report, September, 1998.
3. See http://www.parallax.com
4. D. Wiltzius, *Modeling and Analyzing Visualization Post-Processing over Distance* (Draft), May 21, 1997.

8.10 Acknowledgements

The author wishes to extend special thanks to Rich Fischer of LLNL for his assistance with the use of the Riptide SGI Onyx2 system, assistance during the system setup, and for performing any necessary configuration steps prior to demonstrations. Networking

coordination on the LLNL side was provided by Dave Wiltzius and Joe Slavec, and design advice and assistance on the Sandia side was provided by Joe Brenkosh. Advice regarding testing and demonstration configuration procedures was provided by Tom Pratt, and general philosophical and design advice was provided by Lyndon Pierson and the rest of his ASCI compression team.

9. Networking

The DISCOM^2 FY98 Pilot built several networks to serve the Pilot's testbeds. The Pilot tested the performance of network interface card, NICs, and protocol stacks on the different hardware platforms that the ASCI computational, visualization and storage services use. The performance of key network elements within Sandia and Lawrence Livermore internal networks, as well as, the performance of the shared ESNET ATM OC3 service were characterized. This effort led to improvements in the network performance of SECURENET between SNL-NM and LLNL. These improvements were accomplished by using distance sensitive protocol tuning methods. The performance of the unclassified networks between SNL-NM and LLNL and between SNL-NM and LANL was tested. Several discovered performance bottlenecks, key tuning parameters and performance capability in the current end to end network were discovered.

9.1 Building the Networks

In FY98, the DISCOM^2 Pilot built networks to support the LaPorte servers, IBM F50s, the DISCOM^2 users testbed, and the ATM video experiments.

9.1.1 The Server Networks

The LaPorte Servers are two IBM F50s, a product from the RS6000 family. The servers were purchased with an IBM OC3 ATM NIC and a 10baseT Ethernet NIC. Two networks, an ATM and an Ethernet were constructed to support the Servers.

In addition to the Servers' ATM connections, the ATM network consists of a Cisco LS1010 ATM switch and an ATM connected Cisco 7000 Router. The Cisco 7000 router connects into the Sandia's production EON mesh through another Cisco 7000 router that is directly attached to the mesh. The network uses RFC1577 Classical IP over ATM as the ATM protocol and used switched virtual circuits, SVCs. Permanent virtual circuits PVCs were also successfully tested on the NICs. The IBM ATM NIC did experience intermittent problems losing the ability to stay connected to the LS1010's SVC services. This problem is related to a known interoperability problem of the SSCOP protocol between the many ATM NICs and the CISCO LS1010. This interoperability problem hasn't been seen on Fore System ATM NICs connected to LS1010 that run on similar IBM platforms. The IBM NICs will be exchanged with Fore System NICs to see if they are more reliable. Multiple ATM OC3 NICs will be added into the server to test the F50 ability to source and sink multiple physical network connections. Current plans are to fill the available PCI slots until the RS6000 output saturates.

The Ethernet network consists of the servers each having a 10baseT connection on the motherboard, a Cisco Catalyst 5000 with 10/100 Ethernet ports and the same ATM connected router used in the ATM network. There was an initial assumption made that

the server Ethernet port was a 100baseT port. Because the port didn't do port negotiation it was deduced that the port was only a 10baseT. Once it was ascertained that this was the case the effort to test performance on the Ethernet network became uninteresting. An evaluation of 100baseT PCI cards that could be installed in the RS6000 is being considered.

9.1.2 The Testbed Network

The Testbed Network was constructed to be able to move between multiple SNL-NM production networks. The network was built to support both ATM attached hosts and 10/100 baseT attached hosts The network initially was constructed in Sandia's IRN network, which is the location of the majority of SNL unclassified high performance computing users. The machines that are attached to the ATM portion of the testbed network are an SGI Origin2000 with multiple (4) OC3 ATM connection, a SUN ULTRASPARC II with OC12 and OC3 ATM NIC. The testbed also contained two diskless X-window workstations that attached into the testbed network via 100baseT NICs. DISCOM^2 purchased a Cisco Catalyst 5500 to support the testbed. This Catalyst contains both ATM and Ethernet ports. The testbed is connected to SNL production network through the Catalyst. Any production router that the DISCOM^2 Team connects to the Catalyst 5500 supports the routing of the testbed to Sandia's production networks.

9.1.3 The ATM Video Network

The ATM network was constructed to support the ATM video experimentation and demonstrations. The network connects visualization platforms and users at LLNL, SNL-NM and SNL-CA. The video source for the demonstration was the large SGI reality engines that process the high performance computing applications output. The LLNL's SGI platform RIPTIDE served as the LLNL entry point during the experiments. The SNL-NM entry point was the SGI platform ALEPH0 that is a part of SNL's Interactive Video Lab, ILAB. For the series of experiments that included SNL-CA, the California sites were the output point of the network demonstrating remote user access to the system. The ATM video encoders and decoders used in the demonstration were Fore Systems' AVAs and ATVs. These units convert between NTSC video and MJPEG digital video. The units use standard ATM QOS signaling to set up and tear down the needed video services. The units can also employ a form of dynamic compression which along with ATM services can vary the output video quality based on the network load. The ATM circuits that were employed used both an ESNET's provided PVP as well as Sandia's Intersite Network, a private network between SNL-NM and SNL-CA. The highest quality video stream consumed less than 5% of the available ESNET provided OC3 bandwidth. The same service running across the SNL Intersite's DS3 circuit consumed approximately 11% of the total bandwidth and about 25% of the unsubscribe bandwidth. The FY98 demos and experiments employed a single video steam to pass the data. The DISCOM^2 Remote visualization program plans on providing 4 concurrent steams as an initial service offering.

9.2 Network Performance—Machines

The IP performance of all the major Unix workstations with commonly available operating systems have improved greatly in the last two years. Currently these machines can all reach 90 percent utilization of the physical speed of 100baseT or OC3 ATM. SUN UltraSparcs have been tested with OC12 interfaces and Gigabit Ethernet interfaces. The best results seen for an OC12 stream is about 62 Mbytes per seconds. The best Gigabit Ethernet performance seen to date is 45 Mbytes per seconds. The different performance of these two technologies is tied to the different MTU that each of these technologies uses.

Since the workstation network performance is capable of getting near the physical speed of the NIC, it should follow that user performance would be good. In actual implementations, this is not the case. There are many factors that effect the throughput that the protocol will deliver to the user. In most cases, on LAN performance tracks the machine performance closely. The problems show up on off LAN connections. Off LAN introduces many opportunities for the protocols to not perform. The first problem is that Off LAN routing usually reduces the maximum transport unit size. For TCP this is usually referred to as the MSS, Maximum Segment Size. In the default case the MSS can be lowered to a packet size of 576 bytes. This is a large performance hit to the sending and receiving machines. On most local networks the SUBNETARELOCAL flag is set, so this performance penalty is only seen when going to a different Class B addresses. The SUBNETARELOCAL flag trips another large performance hit, which is a mismatched MSS. On networks where the MSS size that the communicating devices used are mismatched, routers are required to fragment packets. This can seriously degrade performance. At SNL there are Ethernet 1500 Byte MTU, ATM LANE 1500 Byte MTU, FDDI 4420 MTU and CLIP 9180 Byte MTU. If SUBNETARELOCAL is set then the machine will use its MTU as the MSS and as a result the MSS are mismatched. A good implementation of RFC1191 can solve this problem. However, using a smaller MSS will cause a computer to use more processing power to handle the same amount of network traffic.

Network latency was the next problem addressed. At 100 megabit rates the largest standard TCP window is passed in 5.24 milliseconds. At OC12 rates it is passed in 970 microseconds. As a general rule, to keep network latency from effecting protocol performance the TCP/IP window size should at least be equal to the number of bytes required to fill 1.5 times the roundtrip latency of the network. To be able to use large capacity communication pipes TCP Big Windows RFC1323 is needed. UDP can also be used to mask latency, but using UPD for performance is risky.

For user applications other parameters outside of network performance can pace the system performance. Things to be aware of are local disk speed and whether your data is being remotely mounted via NFS or DFS.

9.2.1 F50s

The F50s OC3 ATM test revealed that they are capable of passing and receiving 16 megabytes per second. The IBM ATM network interface cards (NIC) exhibit an intermittent problem that results in the ATM services being dropped. Other vender interface cards and 100baseT NICs are currently being investigated.

9.2.2 Blue Pacific

Blue Pacific is connected via an FDDI connection to the WAN. It is capable of providing greater than 12 megabytes per second. Blue Pacific has 4 FDDI connections that terminate in a FDDI switch. Speculation suggests that the machine could source nearly 50 megabytes per second if the network traffic is diverse.

9.2.3 SNL's HPSS

SNL HPSS has 8 OC3 ATM connected data movers. Each mover is capable of passing 11 Megabytes per second. Tape access and the policy of using no more than 4 movers per job limits single user access to near 40 megabytes per second.

9.2.4 SGI Visualization Platform—Tesla

Tesla is a large Orgin2000 platform. It contains 8 OC3 ATM NICs. Each node is capable of passing 16 megabytes per second. The current best speed recorded for a multiple stripe Parallel FTP (PFTP) to Tesla is 80 megabytes per second.

9.2.5 SNL's INTEL Teraflop Machine

This machine referred to as Tflops or Janus contains 8 OC3 ATM NICs per side, classified and unclassified. Six of these NICs are in production use. Each NIC is capable of sourcing 9 megabytes per second of IP traffic. Four of the NICs together support the parallel HPSS and PFTP applications. The best speed recorded off of the four combined NICs is near 30 megabytes per second.

9.3 Network Performance—Network Elements

9.3.1 FASTLANE

Although FY98 goals did not include classified testing the results of the ASCI's PSE efforts on testing of the FASTLANE ATM encryptors is included. The FASTLANE performance testing showed that the FASTLANEs were capable of supporting full OC3 line rate. The FASTLANE was also tested to see that it could recover without operator interaction from a momentary network outage. The DISCOM^2 team will be doing some tests during the FY99 to test the operational robustness of the FASTLANE. These tests will be geared to cell loss cryptographic recovery, the effect of parallel PVPs, PVCs and PSVCs on the FASTLANE, and the performance of the OC12 FASTLANES.

9.3.2 Cisco 7500 & 7000 Routers

Cisco 7500 routers and Cisco 7000 routers play key roles within the laboratories production networks. Cisco's 7500 routers are the routing platform used by ESNET. The router performance tests were done within the confines of the production environment. That means that other traffic may have been present competing with the test traffic. It was determined that the production traffic was sufficiently small as to not play a large role in the measurement attained. In our local network a router running an Access Control List was capable of providing 90 - 100 megabits per second to the CLIP TCP/IP test traffic. Increasing the source traffic into the router didn't cause packet loss. The buffering within the router was adequate to handle the burstiness of 4 IP streams. This indicated to the testers that the limit was the packet handling rate of the router. When testing to LLNL the router was capable of delivering 64 megabits per second to the test traffic. This level of performance was only possible if the input traffic was effectively spread in a way to smooth out the inherent TCP/IP burstiness. The router performance degraded to below 60 megabits per second when traffic was allowed to get to the expected burstiness of four parallel steams. These results indicated that the limitation was based on the ability of the router to buffer data. Further testing revealed that the router would lose packets if the incoming traffic exceeded an aggregate window size of 430,000 bytes. There was a great amount of effort put into isolating the production router as the source of this bottleneck. Different traffic sources and destinations where used to ensure that the router was the actual place where data was being dropped. Also isolated during these tests was an ESNET 7500 router, the entry point at LLNL. Passing data from LANL and SNL-NM, the result was a slight improvement, 63 megabits per second. This testing of the 7500 is less reliable than the 7000 testing because of the lack of information as to the concurrent production traffic information on the ESNET controlled router. However, in both cases the test indicates that the local production network is the first order bottleneck in the current network. It is also of note that the router didn't give a consistent report of the fact that data was being dropped. Some of the test resulted in the routers giving a loss packet indication while at other times it did not.

9.3.3 LS1010

The testing on the LS1010 revealed that the box was capable of providing full OC3. The switch has a 5 gigabit per second backplane and 64 kilocells of data buffering. The switch has a traffic shaping function that helped to identify the router bottleneck. If the local production traffic and remote ASCI traffic is separated this feature could be employed to prevent router buffer overrun. Further testing on the OC12 switch connection performance capabilities are anticipated.

9.3.4 ESNET

ESNET provides an OC3 ATM network into all FY98 DISCOM^2 sites. The network provides for only one Class of Service, Unspecified Bit Rate, or UBR. UBR presents some difficulties in getting to absolute performance characteristics. However, a snapshot of how much ATM UBR traffic ESNET could support was documented. A completely dedicated OC3 ATM circuit can provided 134 megabits per second of sustained user data.

For this test a high level of AAL5 data was inserted into the circuit at the SNL-NM site for a sustained period, lasting from a minute up to several hours. The traffic was logically loopback at SNL-CA. The output of the AAL5 checksum was monitored back at SNL-NM to verify that no cells were lost in transit. This testing indicated that the circuit could successfully passed 100 megabits per second of sustained traffic. During long testing it was discover that the midday was the period of heavy network use during which the loaded circuit would drop cells. These congestion events where short, lasting for less than a minute. Further testing revealed that the path from SNL-CA to SNL-NM could sustain a greater level of traffic without congesting. This showed that the network congestion was occurring on the SNL-NM to SNL-CA path. ESNET revealed that the level of traffic into the Oakland Sprint POP would account for this one way congestion.

The next test performed was the bursty traffic test. In this test several 25,000 cell bursts were successfully passed into the network at full OC3 speed. This level is sufficient to twice fill the WAN latency of ESNET. These tests indicated that the ESNET circuit could support multiple, two or three, high performance TCP applications without overloading the circuit. The available 100 megabits would be shared by the applications. By engineering any single input to the network to limit throughput to 100 megabit would provide further protection from data loss. These tests also indicate that currently running across ESNET will not be invisible to the user performance.

9.4 End to End Performance

For data transfers from SNL to LANL on the unclassified network the users can achieve 7 megabytes per second on off-hour transfers. This performance falls to less than 3 megabytes per second during operational hours. This different in performance is going to be investigated in FY99.

For Data transfers from LLNL to SNL on the unclassified network the users can achieve 4 megabytes per second. This rate is being paced by the router buffer limitation and the latency between the sites.

10. Distributed Resource Management

The DRM effort began in June 1998. DRM must provide the capabilities and services needed to access and manage the high-end and distributed simulation resources dispersed throughout the DP complex, including computing, storage, network, and visualization resources. Other remote resources envisioned include databases and utility executables such as file format conversion routines. The DRM activity collaborates with the Distance Computing Model, Data/Simulation Framework, and ASCI PSE DRM activities.

To achieve its goal, the DRM project is divided into several tasks:

- The Service Model will provide a user perspective of requirements across the complete range of system operations, problem domains, and organizational cultures. The Service Model is being developed from use case analysis of in-depth user interviews. In FY98, the process of interviewing a broad range of users at the three labs was begun. This effort has been working closely with the Distance Computing Model task.
- The Architecture Model will provide the target system design and an evolutionary strategy for phased implementation of a DRM system. The use of off-the-shelf products is preferred, and they will be employed where consistent with the desired system. A preliminary system architecture, based on an initial understanding of user requirements, was presented at the tri-lab DISCOM^2 meeting in July. Based on feedback from the meeting, further discussions with DRM staff at the three labs, and subsequent user interviews, the architecture model was revised.
- The Product Evaluation identifies candidate products for testing and experimentation. Products have been compared to the desired architecture, and evaluated for cost, features, maintainability, and extensibility. Promising products will be installed and further evaluated in a testbed. The product evaluation task is coordinating with the ASCI PSE Request for Information (RFI) process that is exploring DRM products.
- Collaboration with the external supercomputing community provides additional insights and opportunities to leverage existing work. In FY98, DRM staff collaborated with Geoffrey Fox of the Northeast Parallel Architectures Center for product evaluation and mapping to the architecture model, and for HLA/RTI synergy; participated in the Globus retreat; and participated in the Desktop Access to Remote Resources working group, which is working to define open standards for accessing resource management system services.

Plans for FY99 include:

- A DRM service will be deployed in the distance pilots in the short term to provide capability until a final DRM solution can be agreed on and prototyped in the testbed.
- The Service Model task will merge with the Distance Computing Model task. The user interviews will be completed and analyzed to abstract a set of use cases. A draft document will be produced in Q1. An object model based on the use case analysis will be developed. To manage areas of development risk, the model will be simulated to explore design alternatives. DRM simulation activities will focus on resource management services such as distributed queueing and resource allocation policies.

- The Architecture Model will be refined. A draft document will be produced in Q1. It is expected that the Architecture Model will continue to evolve as appropriate to the findings of the Service Model, simulation, product evaluation, testbed, distance pilot, data and simulation frameworks, RFI, and external community activities. A preliminary system design and prototype will be developed.
- The preliminary Product Evaluation will complete and a draft document will be produced in Q1. Promising products will be installed and further evaluated in a tri-lab testbed, and a final Product Evaluation report will be produced. This activity collaborates closely with the ASCI PSE DRM activities.
- Testing and evaluation of DRM services will be conducted in a tri-lab testbed. Testing activities serve three purposes: test prototype resource management services before they are transitioned to the pilot; evaluate candidate products; and evaluate design alternatives to manage areas of development risk. A draft Test Plan for the testbed activities will be produced in Q1. Test activities and results will be documented in subsequent reports.

Collaboration with the external community will continue.

11. DISCOM2 Model

11.1 Introduction

Build it and they will come...
This philosophy may be true in some environments, but not in the computing community. They will come—Only if it meets their needs/requirements, is relatively easy to use, and help is available. The modeling effort has begun to look into the needs/requirements of the users—gathering their perspective on the environment that is to serve them in the future. A system level model is a high level view of the entire computing environment that is essential to meet the users needs/requirements from job definition through job visualization and analysis. The pilot will be used for prototyping technologies and their uses within the computing environment and feeding the lessons learned into the model. The model will be transformed into a simulation model to look closer at the ramifications of specific placement of servers and services and other technologies.

11.2 Why Visual Modeling?

Modeling allows for an abstraction from the complexity of the system to be developed. It is a way to deal effectively with the complexity of a system. A model does not imply a design or a "how to", but rather a "What" needs to be done. This method is a "Big Picture" approach that is above the details of implementation and which includes a system level view of the interaction of components. The model also provides a common language for the users and developers of the system to communicate effectively while documenting their understanding.

Quatrani [1] compares modeling to the world of architecture where blueprints are developed to view the structure to be built. In the Distance Model instead of a blueprint it should be viewed as a roadmap. The results of the pilot will be fed into the model—what the model may suggest is not one path, but a roadmap where multiple paths would be feasible dependent upon the user application, the data generated, the visualization needed, etc. Before a model can be generated the requirements of the system must be gathered and understood.

11.3 User Requirements for Computing

11.3.1 Gathering the requirements

To begin the process of requirements gathering during the initial FY98 model activity, interviews and discussions began with those using the pilot and others familiar with the Teraflop environment. These early interviews/discussions resulted in the development of the Roles Spreadsheet in Figure 11.1. This spreadsheet breaks down the typical developer and analyst user roles within the application life cycle into the actions and characteristics of each stage. The stages of Job Creation, Job Submission, Running, Post

Processing, and Display/Output became the initial use cases that will be discussed later in Section 3.

	Code Development		Analysis	
	Action	Characteristic	Action	Characteristic
Job Creation	Compiling	Content of local environment	Define simulation problem	Domain Knowledge Understand Needs
	Debugging	Short turnaround Interactive	Determine tools	Availability Understand applicable Tool resolution
			Grid Generation	Interactive Preprocessing Access to many local files Control codes Licensing
			Parameter Selection for Input	Manual Goal Oriented
Job Submission	Move Input files	Local resources	Move Input files	Resource selection Local or remote Authorization
	Move binaries	Local resources	Move binaries	Resource selection Local or remote Authorization
	Launch Job	Many small jobs Iterative	Launch Job via job queuing pkgs.	One large job Authorization
Running	Monitoring	Optimization	Steering	Interactive
			Monitoring	Determine correctness
			Staging Data	Interactive Scripts
Post processing	File format conversion	All local Does not need to occur on VIZ server	File format conversion	Selection of formats Selection of VIZ server
			Data Reduction	Remove redundant and/or unimportant data
			Staging Data	Right Format Right Place Right Time
Display/Output	Output display and rendering	Display and person co-located Need historical data Short turnaround Access to post processing data Interactive Text/Plots/Graphics	Extract Information	Co-located with display Interactive Identifying region of interest Contrast/color differentiation Needs historical data Access to post processing data
	Validation	Code correctness	Validation	Model correctness
			Data Reduction	Determine reducibility

Figure 11.1 Developer and Analyst Roles within the Application Life Cycle

As the need for further interviews became apparent for both the Distance and Distributed Computing areas of DISCOM^2, a combined effort was undertaken to formalize the interview process. Appendix A contains the working interview questions that were drafted from this combined effort. These questions have been used to guide numerous interviews in the summer and fall of '98. These interviews included customers at SNL and LLNL. Future interviews will include LANL and Y-12. The interviews will be written up and translated into use cases. Since the use cases define the interaction of the system from the user perspective, the developed use cases will be presented to and reviewed by the user community to ensure completeness and accuracy.

The interviews will help us understand and look at how a user currently uses the system and how they would like to use the system in the future. The information gathered during these interviews and discussions will be used to create the use cases for the Distance and Distributed Model. These use cases in turn will be used to understand the future direction/needs/requirements and plan for them within the computing system via the model.

11.4 Users and the Computing Environment

Users have tended to use the system in the way that works for them. This may or may not be the easiest/most effective/efficient way to use the larger systems. Their use appears to be dependent upon who they talked to when they first needed to use the larger systems. As things have changed, many users have not gone back and revisited how they use the larger systems, they continue in the same usage mode that has always worked for them in the past. Thus, they may not be taking advantage of the full capability of the systems they use. This is why it is necessary to create the roadmaps for the users. But, these maps can not be created without knowing how the users interact with the system and would like to interact with the system. Separating the real needs/requirements from the current usage mode is an interesting and difficult task that must be done.

The system should be modeled after the needs/requirements of the user, and the users should not have to mold their work to fit the computing environment. There must be a balance between what the user needs/requirements and what is technically feasible. The user needs to know the options available within the computing environment—what will work best is dependent upon the application, the amount of data generated, the visualization needs, or a combination of these. Basically the user should know the tradeoffs within the system, such as: DFS vs file transfer—when it is more efficient/effective to use one over the other and why.

11.5 Use Cases

11.5.1 Use Case Modeling Definitions

The following definitions were taken from Visual Modeling with Rational Rose and UML by Terry Quatrani:

Actor – Someone or something external to the system that must interact with the system under development…In the Unified Modeling Language (UML), an actor is represented as a stickman.

Communicates Association – represents communication between an actor and a use case…is represented as a line connecting the related elements. The navigation direction of an association represents who is initiating the communication.

Model – An abstraction that portrays the essentials of a complex problem or structure, making it easier to manipulate

Scenario – An instance of a use case—it is one path through the flow of events for the use case

Use Case Model – The collection of actors, use cases, and use case diagrams for a system

Use Case – Sequence of transactions performed by the system that yields a measurable result of values for a specific actor. Representation of the business processes of the system. The model of a dialogue between an actor and the system…In the UML, a use case is represented as an oval.

Use Case Diagram – A graphical representation of some or all of the actors, use cases, and their interactions.

Visual Modeling – A way of thinking about problems using models organized around real-world ideas.

11.5.2 Current View

The information gathered initially was used to create current and future use case and pictorial views. These views were used to communicate an early interpretation of the discussions and interviews. Although the computing environment is used in many different ways the basic usage flow is depicted in the Current Use Case View-Figure 11.2. The user initiates four of the five use cases in the current view. Thus, the current view is very depended upon the user pushing the overall task through the computing environment.

The user defines the problem, compiles and debugs during Job creation. Then during Job Submission the user FTPs input and binary files, launches the job, and then waits for the job to be released from the queue. Job Computation occurs when the job is released from the queue. This use case includes the basic computation with minimal monitoring capability and could include data staging. Currently the batch submission does not allow interaction between the user and the running code. During an interactive submission there is some interaction, but it is still limited. Job Post Processing begins when the computation completes. The user must move the generated data—this is done manually

file by file or by starting a script to more the files. Data manipulation also takes place at this point in time. The Post Processing Use case is basically getting the output data into the right place, in the right format, at the right time (i.e. the Viz resource is available). And finally during the Job Output/Display Use case the output data is displayed and analyzed. See Appendix B for the Draft use cases from the early discussions/interviews that are associated with the current use case view. These use cases are draft and were used as proof of concept to determine if they could represent the current mode of usage.

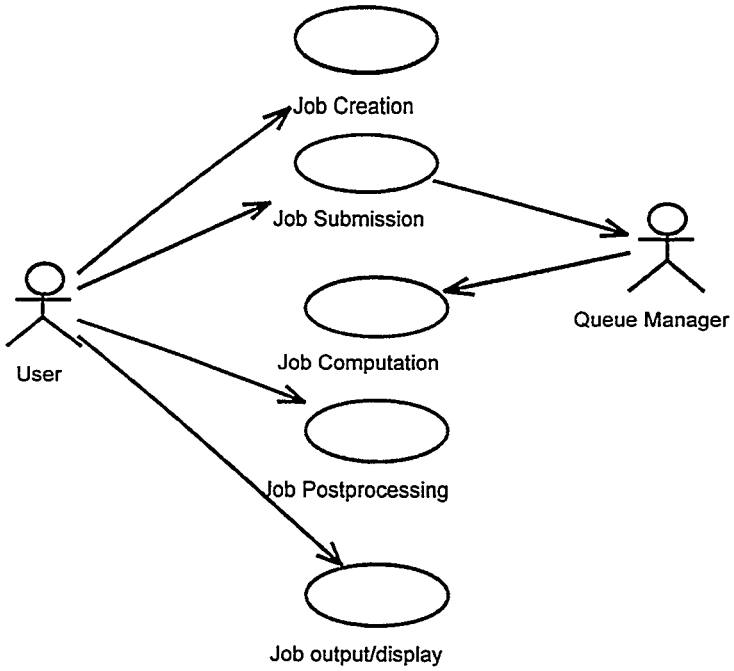

Figure 11.2 Current Use Case View

Figure 11.3 shows the Pictorial View of the Current Use Case View. The user LAN, the Large Local Server LAN and the Very Large Remote Server LAN are not closely coupled. The users must be knowledgeable of the environment to run jobs in this computing environment. Many users rely on scripts to hide the complexity of this environment.

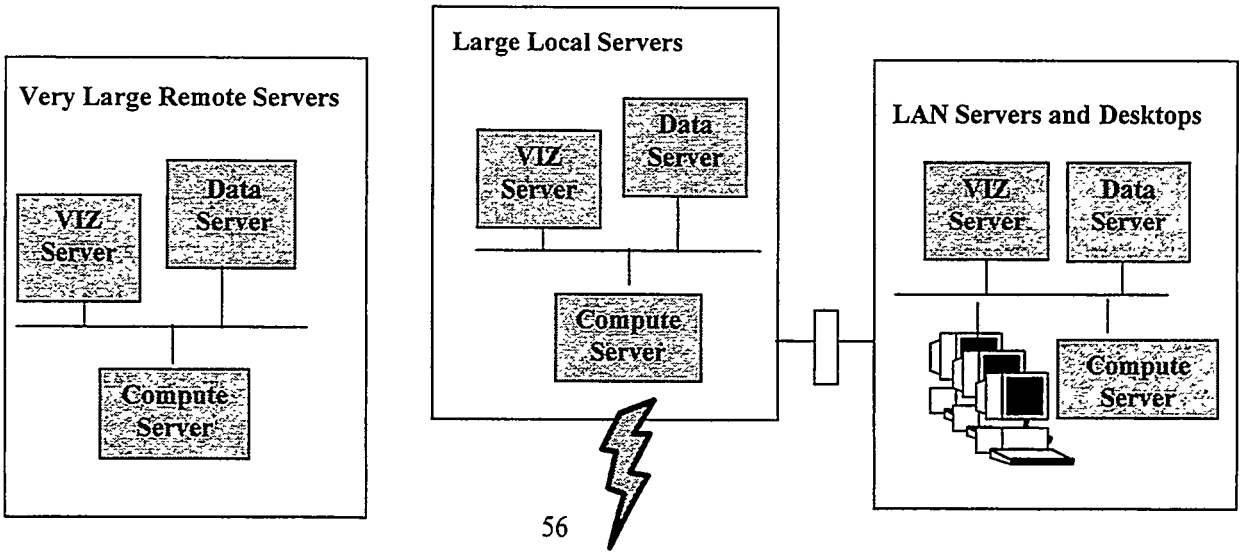

Figure 11.3 Current Pictorial View

11.5.3 Future View

When discussing future computing needs with the users the flow becomes more interactive. The Future Use Case View in Figure 11.4 shows a dramatic change in how the user would like to interact with the system. Instead of the user initiating most of the use cases (as in the current view) most of the use cases are initiated from within the environment. Again there are the basic use cases seen in the current use case view, but information flows back to the user and between the resource and server managers within the computing environment. In addition there would be a user interface, a resource allocation use case, and several resource/server actors. Basically the future view shows the user initialing the Job Creation and Submission use cases, then the resource manager via the Resource Allocation Use case would contact the computation, storage and Viz managers to negotiate and allocate these services/resources for the user. The user would receive updated information from the Job Computation, Job Post Processing, and Job Output/Display use cases, then be allowed to respond as appropriate to these use cases. There would be direct interaction between the service/resource managers to ensure the data files (input or output) would be available when needed.

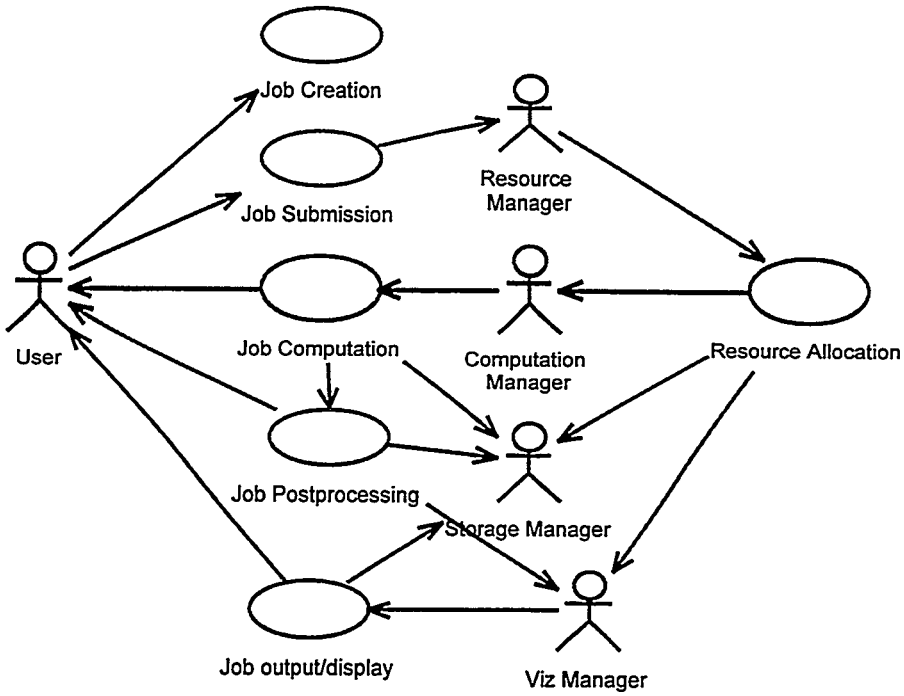

Figure 11.4 Future Use Case View

Within the future environment the users vision includes: The desktop as a window into the computing world. Data is accessible wherever it is needed—there would be no need to stage data. There would be no duplication of data. Resources would be easily accessible. Interactive capability would be available. A job could be submitted via a user interface by specifying the input files and the code to be run. Monitoring and Steering Capabilities would be fully functional. And there would be the ability to both expand and contract a running job due to the increase or decrease in nodes available to the job.

In order to have a computing environment that would fit this vision the environment would need to be more closely coupled than the current environment. Figure 11.5 depicts the Future Pictorial View. This view stretches the fabric of the computing environment and begins to blur the boundaries between the once distinct environments. Other considerations that need to be investigated in this future environment include the user interface, security, messages, status, standards, and local and global policies.

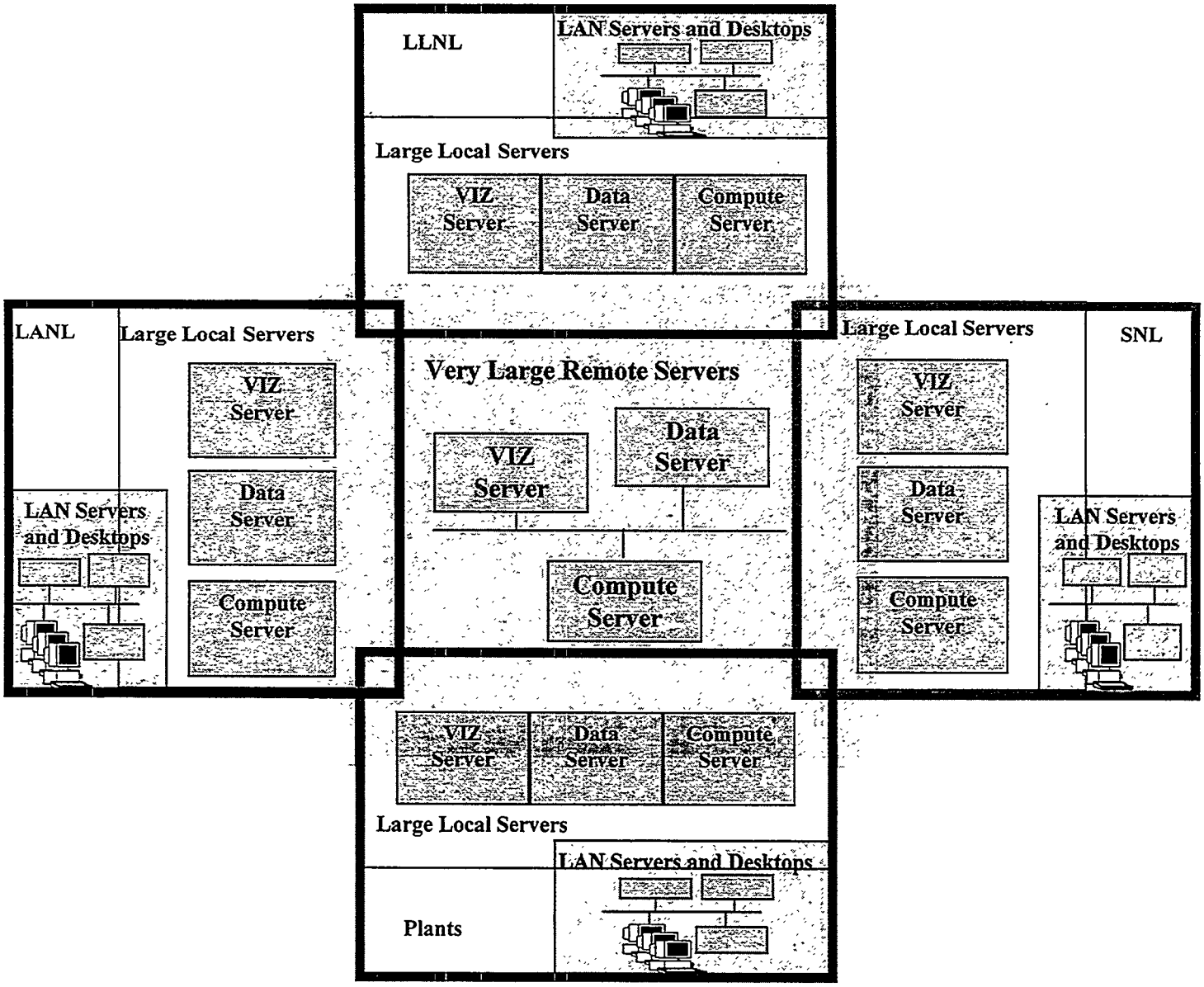

Figure 11.5 Future Pictorial View—Stretching the Fabric

11.6 The Model and the Pilot

One of the things that is not shown/described is the evolution form the current to the future. Understanding what is technically feasible and how it can be applied in the computing environment is essential. The lessons learned in applying technology in the pilots help to determine feasibility and scalability. This information is fed into the model. The lessons learned combined with user requirements, and architectural needs are incorporated into the model and then back into the pilot. This interaction between the model and the pilots can best be depicted by Figure 11.6—presented during the 30 July 98 DISCOM^2 meeting by Mike Vahle. The intent is to try new/different technologies in the pilots then to determine what works and include those in the model. The model is the big picture of the computing environment—The system—and ensures the pilots are representative of that environment.

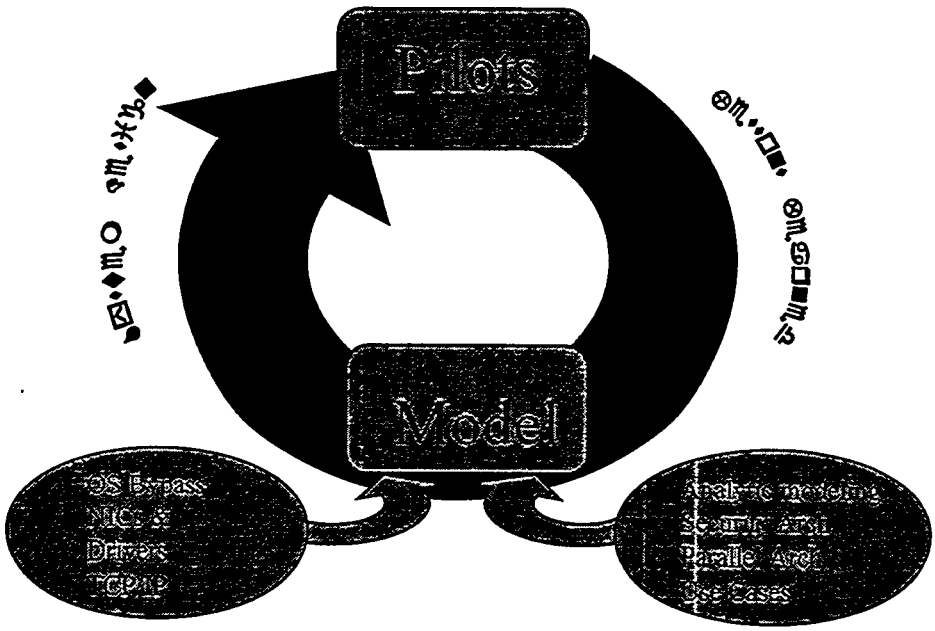

Figure 11.6 Interaction between the Pilot and the Model

11.7 Coordination and Future Directions

11.7.1 Coordination of Essential Services

One of the starting points in further understanding the user community and in working with the other related projects is to determine and understand the essential services needed for a complete computing environment. Table 11.1 lists a draft version of the Essential User Services, while Table 11.2 lists a draft of the Essential Administrative and Network Services for distance and distributed computing.

Initially there was a need identified to prioritize and discuss these services in order to provide a clearer understanding of the tasks and resources required to put the services in place for distance and distributed computing. The discussions began in the fall of 98 and are continuing. These are tri-lab meetings (videoconferences) led by Martha Ernest. The effort was focused to determine the following: Who is working in each service area, the current status and direction, resource requirements for the service (People, tasks, hardware/software/licenses, etc.), dependency of service to other services/tasks/resources, ordering/prioritization of services, and finally identifying the roadblocks and who is waiting on what area.

The meetings have been well received by the tri-lab community and are continuing. These discussions are necessary for the flow of information between projects, and provide the cooperation/coordination that is required for these initiatives to accomplish the shorter-term tasks and balance the longer-term goals.

- **User Authentication**
- **Terminal Access**
- **Parallel Operating Environment**
- **Code Support (libraries)**
- **Interactive Access**
- **Batch Access**
- **Load Leveler, LSF**
- **Code Compilation**
- **Fortran, C, C++**
- **Loaders**
- **Code Debugging**
- **Parallel Debuggers**
- **Editors**
  - **vi, emacs**
- **Compute Cycles (TOPS)**
- **Simulation Monitoring & Steering**
- **Data Servers (move/share/archive)**
  - **FTP/PFTP/RCP, NFS/DFS, Storage/Archive**
- **Output Services**
  - **X, MPEG, JPEG, Taping, Printing**
- **System Status**
  - **Mail, Web, Data file**
- **Visualization Services**
  - **SGI servers, Ensight, AVS/Express, IBM Data Explorer**

Table 11.1 Initial Essential User Services for Distance and Distributed Computing

Administrative Services

- **User Authentication**
- **Kerberos**
- **DCE Authentication**
- **DCE Cross Cell Authentication**
- **Resource Management/Auditing**
- **Distributed Resource Management**
- **LDF, Loadleveler**

Network Services

- **Data Encryption**
- **IPSEC, LINK**
- **Data Transport**
- **Ethernet. IP, ATM**
- **IMP**
- **Firewalling**
- **IP routing**
- **DNS Service**
- **Time Service**
- **ARP**
- **LANE**

- **Resource Management**
- **Auditing, Batch Queuing, Time**
- **Code Compilation and Loading**
- **Compilers, YOD**
- **Data Security**
  - **Policy, SSH, IPSEC**
- **User Notification**
  - **Web, Mail, Data file**

Table 11.2 Initial Essential Administrative and Network Services for Distance and Distributed Computing

11.7.2 Coordination with Distributed Computing Efforts

An effort between the Distance and Distributed computing areas of DISCOM^2 continues with the interviews and further development of the use cases and models. Although the basis of the use cases come from the initial work documented here (Current/Future Use case view) the emphasis will be toward the future and a more complete set of use cases that is representative of the tri lab user community. As this paper is written that work in moving forward and drafts are expected in late '98 or early '99.

11.7.3 Coordination with ASCI Efforts

Coordinated work with the ASCI Network Simulation and Modeling project began in the fall of '98. There is an effort underway to tap into the networks and capture the usage pattern(s) for a specific user. This will characterize ASCI application and how they use the network. In addition, the background characteristics of typical traffic will be collected. The information will be analyzed and later used to ensure accuracy of the model.

11.8 Summary/Conclusions

Efforts that began in FY 98 to understand the user needs/requirements are continuing and will result in complete use case views and generated models in FY 99. The model created from the use cases will be transformed into a network simulation. Input from the pilots, testbeds, and other projects will help to create an accurate representation of the network, the servers, services and their interaction. This will allow the comparisons of different alternatives/solutions—placement of servers, services, and network technologies. Some of these alternatives/solutions will become the roadmaps (mode of usage) for the users of this environment.

Some of the future areas that need to be investigated are discussed in portions of the DISCOM^2: Distance Computing The SP2 Pilot FY98 Final Report. Some of the questions to be answered in the '99 pilots include:
Visualization What capability—remote and local—is required?
DFS on clients and servers—who are the severs and who are the clients?
File migration via HPSS
FTP vs DFS what are the tradeoffs?

Where should the application servers reside?
What capability and how much capability are required of a proxy?
Can the System Area Network be expanded?
Can the computing fabric be extended to overlap each site—how can this be done?
How can new technology be integrated—how do we plan for this?
What are the implications for the model?

11.9 References

1. Terry Quatrani, Visual Modeling with Rational Rose and UML, Addison-Wesley, 1998.

12. Observations

The following are general observations of behavior that is seen at all high performance computing sites. While there are many difficult technical challenges contained in the DISCOM^2 directive, some of the most difficult tasks are associated with the creation of inter-organizational trust and the blending of the computing cultures. All of the pre-eminent computing facilities associated with DISCOM^2 have their own perception of what distance computing is, as well as, what distance computing should be. Bringing the sites to a common understanding and vision is a significant challenge.

The communities that handle the daily operations of the high performance computation centers have a production culture that is highly motivated, but short-term goal orientated. There tends to be little formalization of the operational procedures to plan and implement system changes. This is acknowledged as a problem that needs to be addressed in order to support tri-lab coordination and synchronization of changes. This is essential when systems are shared and consistency within the environment is required.

Application and user priority assignment is an area where the lack of formality can be a source of concern for the remote users. The lack of formality is viewed favorably as a function of a pre-eminent computing facility because it allows for highly flexible resource management. The DISCOM^2 model must accommodate flexible administration of priorities for both local and remote users. To accomplish this the large computing site will need to accommodate the operational flexibility needs of the remote institutions.

In general it is common for organizations to impose formality on a remote user that they don't impose on their own local users. The DISCOM^2 Pilot will continue to champion the degree of formality that is needed to protect remote users from the local user bias that tends to be inherent in some environments. While there have been some successes at addressing these types of problems, more attention is needed to overcome the barriers these differences create. These barriers are at odds with the long-term goals and objectives of distance computing.

The ASCI program has many interlocking efforts. In many cases it is access to the human resources that is elusive when trying to accomplish the goals and objectives. The deployment of the next generation of computing hardware at LLNL created a large workload for their computing staff. This lightened the support needed for the creation of distance computing services. Cooperation and coordination of many people are required for distance computing. This factor has been a pacing element for some activities in DISCOM^2. Weekly meetings began in October to bring many of these players together for the essential services for DISCOM^2.

The FY 98 pilot brought to light many of the needs/requirements and issues of a future computing environment that should appear local from a distant user. This initial pilot

broke the ground and began the foundation for the understanding and prototyping of a future tri-lab computing environment. As the next pilot proceeds there are many groups that are beginning collaborative efforts to move computing forward. These efforts include, but are not limited to services, networks, system models, security, and resource management.

Appendix A: High End User Interviews

This document describes the typical interview team introduction, questions, and logistics.

Introduction

We're performing this analysis for Art Hale for DISCOM^2. Our goals include **understanding**

- Job
- Lifecycle
- How they work (current and future)

This will be used to help us define system models, which assists in **requirement definition** for resource management – both distance and distributed. This requirement definition will be useful for pilot projects (you're not being signed up to participate in a pilot!).

We may need to have follow-on session(s) for verification or fleshing out the model.

Do the interviewee(s) want to remain anonymous. Will need to identify info source down to the code level.

Questions

How

- ☐ 1. How do you work now?
- ☐ 2. What bugs you about your current process?
- ☐ 3. How would you like to work?
- ☐ 4. How does code interact with resources?
- ☐ 5. How does a heterogeneous environment impact you?
- ☐ 6. How interactive do you want to run?

Why

- ☐ 1. Why run on a big machine?

What

- ☐ 1. What is a job? {Compute, visualization, data reduction, others?}
- ☐ 2. What is the job mix? (Probably inferred after a number of interviews.)
- ☐ 3. What's the job flow?
- ☐ 4. What's the big picture with respect to the code's lifecycle?
- ☐ 5. What are the inputs? Outputs?
- ☐ 6. What are the results? Problem trying to solve?
- ☐ 7. What machines are used?
- ☐ 8. What visualization is used?
- ☐ 9. What software packages/libraries are used? Licenses?

- ☐ 10. What CM or release management are you using? What's the specification/pedigree of the job/code, parameters, data?
- ☐ 11. What are your security requirements?

When

- ☐ 1. How often does code get run?
- ☐ 2. Any sequencing/dependency issues?
- ☐ 3. Timing characteristics – real time, other?

Where

- ☐ 1. Where does the code fit in the big picture?
- ☐ 2. Where are the input/output stored? Moved? Archived?
- ☐ 3. Where is the code run?

Who

- ☐ 1. Who is the main POC?
- ☐ 2. Who is involved with the code – roles?
- ☐ 3. Who are the customers (immediate and downstream) and suppliers?
- ☐ 4. Who touches the input/output?
- ☐ 5. Who needs to be informed of certain occurrences in the system? What occurrences?

Logistics

- The meeting should be held at the interviewee's place
- Each session should be less than 1.5 hours
- Use an electronic white board or flip chart to record session notes
- Have at least 2 interviewers
- Provide 3-5 top level questions to interviewee to think about prior to interview – 1st 5 "How" questions

Appendix B: DRAFT Use Cases for DISCOM²

DRAFT
1.0 Use Case: Job Creation
Begins when a code change/enhancement/update is needed
Inputs: Domain Knowledge
1.1 Define the simulation problem
1.2 Determine the tools to be used
1.3 Determine grid to be used or generate the necessary grid
1.4 Select or consider the parameters for input
Ends when job is ready to be submitted on large machine

DRAFT
2.0 Use Case: Job Submission
Begins when job is ready for submission to large resource
Inputs: Input files and binaries
2.1 Determine resource requirements
2.2 Select resources to be used
2.3 Move Input files
2.4 Move binaries
2.5 Launch Job
2.6 Check on Job
Ends when job is released from the queue to run

DRAFT
3.0 Use Case: Job Computation
Begins when the job is selected to run on the resource.
Inputs: binaries, input decks, resource requirements
3.1 Computation of the Job
    Time steps
3.2 Steering the job
3.3 Monitoring the job
3.4 Staging the data
    Local output
    Remote output
Ends when Computation is complete

DRAFT
4.0 Use Case: Job Post processing

Begins when the Job Computation has completed and user is ready to manipulate the data
Inputs: output data from job computation.
4.1 Conversion of data into the proper form (optional)
4.2 Translation of data from one format to another
4.3 Reduction of Data (optional)
4.4 Staging the Data (optional)
Ends when data is in the right format and the right place.

DRAFT
5.0 Use Case: Job output/display
Begins when output data is in the right format, at the right place, and the right time (resource is available)
Inputs: Output data. Available
5.1 Visualize data (optional)
    Fly around image
    Run time step sequence
5.2 Print (optional)
    Image
    Text
    Plot
5.3 Create movie (optional)
5.4 Record images (optional)
Ends when output/display is complete or Viz resource is released

Appendix C: Getting on the ASCI curve

John Naegle
Steve Gossage
Richard Hu
Thomas Pratt
Sandia National Laboratories

According to the ASCI program requirements, a 100 Gigabit Wide Area Network (WAN) will be required in the year 2004 to support the 100 TeraOp computational machine.

The ASCII program has developed the concept of a "network curve" to map a path to reach this goal. The curve outlines the milestones needed to reach this goal on the required schedule. It is clear from the curve that the evolutionary extensions of current technologies, as driven by industry, will not progress fast enough to meet the ASCI requirements. At the "Workshop on End-To-End Data Flow in the ASCI Infrastructure", several research areas and strategies were identified that will enable the accelerated ASCI network curve.

Strategies:
The latency and performance bottlenecks of the Internet Protocol (IP) must be overcome. Although improving the IP implementations can make some gains, the most promising strategy that has been identified for achieving significant performance improvements is the Operating System (OS) bypass mechanism. The concept is to bypass the overhead associated with processing the IP stack from user space, through the OS, and then to the Network Interface Card (NIC). By having the NIC card move the data directly from the user space using Direct Memory Access (DMA) or some other efficient mechanism, the network latency is greatly reduced. Since the amount of processing required to move data is reduced, the network performance is improved as well as freeing more CPU cycles for computational processing. In order to implement this strategy, more intelligent and capable NIC cards must developed. Also, applications will require porting to the new Application Programming Interfaces (APIs) provided by the OS bypass mechanism.

Although faster network interfaces are being developed, none of the projected interface technologies will be fast enough to meet the required ASCI WAN bandwidths with a single pipe in the 2004 time frame. Parallel networking will be required to reach the 2 to 3 orders of magnitude improvement required by the ASCI curve. There are two efforts required to implement parallel networks. The first effort is to build a physical network infrastructure that is capable of providing parallel paths between end hosts. The second effort is to develop the applications and network drivers that are able to take advantage of the parallel network infrastructure.

All of the performance gains achieved through OS Bypass and parallel networking will be wasted if the current strategy of using single gateways and access control lists continues to be the security architecture. These mechanisms will not scale. A new strategy of providing security in high speed WANs will be required. The ASCI program is currently implementing a host based security strategy that will require better security on the hosts (at least a C2 OS) but will remove the security bottlenecks from the high-speed data path.

Partnerships with long distance carriers for providing the WAN bandwidth must be identified and developed. The current commercial cost of purchasing WAN bandwidth is prohibitively expensive. There are several carriers that may be interested in learning, with the ASCI community, how to effectively provide very high bandwidths. A research partnership with such a vendor would be mutually beneficial to the carrier and to the ASCI program.

Known technologies:

There are a few specific technologies that are projected be part of any ASCI WAN.

The WAN bandwidth providers have committed to using Dense Wave Division Multiplexing (DWDM) in order to provide the very high speed bandwidths required by the ASCI program. This dictates that 100 Gigabits/Second connectivity between the ASCI sites will not be provided with a single channel. The quanta of communication that the providers are building towards for the year 2004 is the OC-48 (2.4 Gigabits/Second) rate. Therefore, a 100 Gigabits/Second link would be configured as roughly 40 parallel OC-48 pipes. While the quanta rate may quadruple, optical dispersion and other fundamental physical limitations will probably not allow the quanta to increase beyond OC-192 (9.6 Gigabits/Second) in the 2004 time frame. In order for the ASCI program to utilize the available WAN bandwidth in 2004, mechanisms for effectively using 40 parallel paths will be required.

The MPI API appears to be the leading choice for the application programmers in the ASCI community. Some application programmers are willing to write directly to the physical drivers of the currently available NICs to achieve significant performance improvements. OS Bypass mechanisms, coupled with an efficient MPI implementation, should be able to achieve almost the same performance as applications written directly to the underlying network driver. The obvious benefits of using a common API for all applications on all platforms and networks should solidify MPI as the API of choice for computational applications.

There are several different OS Bypass mechanisms being investigated. The list includes, but is not limited to Scheduled Transfer (ST), Fast Messaging (FM), Myricom's GM, and the Virtual Interface Architecture (VIA) specification. There is not enough data currently available to indicate which technology will be the most effective in the ASCI environment. Since the OS bypass mechanism will be critical to achieving the ASCI curve requirements, much more research must be conducted to understand the issues and

relative merits of these different mechanisms. A coordinated, tri-lab effort to assure that this research is not overly duplicated and is not completely disparate is required. It may be that integrating more than one of the mechanisms in the ASCI environment will be required.

Although OS bypass appears to be the future for very high-speed networking, the TCP/IP protocol will continue to be the workhorse of the WAN networks for several years. It may be that the TCP/IP implementations can be improved sufficiently to meet all but the very highest ASCI networking requirements. Therefore, the ASCI community must continue to research improved TCP/IP implementations and effectively tuning the existing implementations to achieve the maximum performance in the existing networks.

Necessary Research Elements:

An environment where the network strategies and technologies identified in the ASCI curves meeting (as enumerated above) can be investigated is needed. The current computational ASCI environments are focused on computational performance. In order to get on, or ahead of, the ASCI network curve, an environment where the network is the focus of the research is required. At Sandia, this proposed environment would build on existing network infrastructure and LDRD research efforts. It is anticipated that LLNL and LANL would build environments based on their respective research agendas. Common elements of these environments would consist of the following areas:

1. A high speed, parallel, System Area Network (SAN).

The SAN will be a fully switched network, which provides parallel paths. Investigation of a switching technology that will have perfect scheduling and flow control to provide fair, guaranteed delivery of network traffic will be performed. In order to achieve a very aggressive price point, the initial physical interconnects will be copper 1.2 Gigabits/Second links.

2. Seamless conversion of the SAN to fiber, WAN, and parallel (SONET, DWDM) technologies.

Although the initial SAN will be copper, fiber interconnects will be developed in order to extend the SAN between computer rooms and to standard structured wiring systems. The initial speed of the interconnects will be OC-24. Since the WAN links are OC-12 or OC-48, speed matching interfaces between the SAN and the WAN links will be developed. Extending the perfect scheduling and flow control across the WAN will be investigated. Extending the parallel switched network across parallel DWDM WAN links will also be investigated.

3. OS Bypass mechanism with an intelligent NIC implementation

The first OS Bypass mechanism to be investigated in this environment will be an implementation of the VIA specification from Giganet. This implementation will provide a complete VIA software development environment, as well as intelligent PCI NIC cards. This will allow detailed analysis of the interaction of the OS Bypass, system memory bus, PCI bus, application, and network. The performance improvements as well as limitations

and programming environment will be documented. This research will be coordinated with the OS bypass research being conducted at the other laboratories.

4. MPI API on top of VIA to simplify porting ASCI applications to the proposed environment

MPI is a critical technology in the ASCI environment. The focus of this effort will be to achieve high performance as well was maintain the required feature set of MPI when using an OS Bypass rather than the traditional IP stack or native network drivers. This will allow porting some of the important ASCI codes to this environment. Investigating the interaction and the scaling performance of ASCI codes in this environment should give a good indication of how the codes will operate in the production computational environments.

5. Sufficient end hosts to load the SAN and WAN

One of the most difficult issues when designing and implementing network protocols and applications is scaling from the test environment to a full production network size. An application that works well in a small test environment may completely fail when scaled to a large production network. Therefore, this environment will require a sufficient number of end hosts to validate the ASCI network curve. Since the computational performance of the end hosts is not as important as network capability, the cost of the end hosts should be much less than in the ASCI computational environments.

6. Detailed analysis of both parallel benchmark codes, as well as ASCI production codes

Since this environment is focused on network research, the absolute computational ability is not the focus of the analysis. The focus is to validate the ASCI network curve and demonstrate how real ASCI applications can use the provided bandwidth. This analysis will also help indicate where the important issues will be when integrating the lessons learned with the production ASCI computational platforms.

7. Connecting existing ASCI resources to and through this environment

Although the goal of this environment is to achieve the performance requirements of the ASCI network curve, many of these technologies can not be implemented in the existing ASCI computational machines. Therefore, methods for using this environment to provide high speed connectivity between existing ASCI resources will be investigated. As the future ASCI computational machines start to incorporate technologies from this environment, more tightly coupling this development environment with the production systems will be investigated.

8. Parallel application development

The goal of parallel networking is to provide parallel performance without having to modify the applications. That goal is not possible in the current TCP/IP environment. The applications must be modified to be aware of the parallel paths. There is a substantial amount of ASCI research progressing in this area. This research will be leveraged to investigate its applicability to the OS Bypass technologies. It is not clear at this point if the OS Bypass technologies will be able to hide the network parallelism from

the existing applications. Parallel networking must be built into the OS Bypass standards rather than retrofitting as has been required in IP. This area, as well as intermediate steps of providing modified applications that can exploit parallelism, will be investigated.

9. Security architecture

Most of the current security functions are based on IP networking. Many of the future security functions will be host based and DCE oriented. Since a majority of the ASCI applications are classified, whatever technologies are used to get on the ASCI curve must be developed and tested for conformance to classified computing requirements. _

10. TCP/IP research

As identified at the ASCI network curve meeting, the primary limitation of the current TCP/IP performance is due to poor vendor implementations. Since TCP/IP will continue to be used extensively in the ASCI environment, cooperative research with vendors to improve the existing implementations is critical. Cooperative efforts between the tri-labs to tune the TCP/IP protocol for maximum performance through the high-delay WAN is also critical to achieving the goals of the ASCI curve.

The goal of this environment and research effort will be to demonstrate that the ASCI network curve's WAN performance can be achieved. All of these research efforts should be integrated as a tri-lab effort in order to minimize overlap and simplify the final integration of a working environment. In order for the concepts that are developed in this project to be applied to the 100 TeraOps machine in 2004, this proposed timetable is to demonstrate the core networking capabilities by the year 2001. The application development, technology comparison and integration, and WAN resource acquisition will continue until a system that provides location independent access to a 100 TeraOps computational platform from all three laboratories is operational.

Appendix D: May 28 Status Report

Discom Pilot Status Report May 28th, 1998

Networks (Tom Pratt)

LaPorte Servers

The two IBM F50 are in place in the 880 central site. The machine 100 megabit Ethernet interfaces are operational and configured in the NWIS database.
Their names and IP addresses are:
laporte1 132.175.108.31
laporte2 132.175.108.32

The 155 mbit ATM connections are not operational (fixed Jun 3rd) but are configured.
Their names and addresses are:
laporte1-atm 132.175.26.20
laporte2-atm 132.175.26.21

The ATM interfaces aren't doing SVC signaling correctly. A trouble call has been placed.

LaPorte Desktops

EON and IRN LANs requested
LAN Switch ordered, expected arrival date 5/30/98

BLUE Pacific / BABY

Made contact with IBM to see if SP2 router is viable as an internal SAN extention machine(next meeting June 2nd)
75% of SNL Team has access to Blue Pacific
Basic FTP benchmark from IRN to Blue Pacific show .2-3megabit/sec performance.
We are building a network performance suite on the LaPorte servers. When they are built we will transfer them over into user space on Blue Pacific.

No agreement on the use of Baby as a Interim remote test SP2 platform
No agreement as to direct LLNL support of DISCOM Distance configuration experiments

Cross WAN network performance tests

We have been examining the IP fragmentation issues between various the sites' LANs

Remote Video (Tom Tarman)

1. Ordered two FORE switches for ASCI compression/remote viz. work. If ATM-based viz. approach is used for pilot demo, then these switches can be put to use for this purpose as well.

2. Currently evaluating the use of Parallax Graphic video capture/display adapters for IP-based remote viz.

June Goals

1. Evaluation of IP video completed
2. Recommendation for remote viz. completed
3. Start with implementation of remote viz. from LLNL to Albuquerque.

Security Architecture

1. Leveraging activities on SNL NwSA and C-PLANT security architecture.
2. Made second presentation on C-PLANT security.
3. Preparing for 25 node Windows NT C-PLANT to be delivered in July.
4. LaPorte security will be at least same as current ASCI, with NTK enhancements where appropriate.
5. Also leveraging NTK Pilot with Discom2.
6. John Noe and Tom Pratt took a DCE Internals class

Services

The LaPorte systems have been installed and are connected to the Ethernet. Working with IBM to correct a bug in the ATM interfaces (Fixed June 3rd). Have ordered the compilers for the systems. Not yet received.
Will build the ssh/krb5 code for AIX 4.2 on another system and install the binaries this week.

What we had asked for from LLNL was a list of the IBM AIX system installed resources...that is the software modules which are loaded onto the system. These include such things as the OS version and any options loaded, the installed software products and runtime environments, compilers, loaders, debuggers etc. We have some of this available from the web pages but the IBM system provides means to list this information and this output is what I was after in order to check our release levels against theirs. IBM is particularly difficult to work with when trying to align systems to the same revisions.

Sue Goudy, Tom Pratt, and Rupe Byers have obtained LLNL accounts and have logged into the Blue Pacific system.

List of Key Services Created

Key Administrative Services

- User Authentication
  - Kerberos
  - DCE Authentication
  - DCE Cross Cell Authentication
- Resource Management/Auditing
  - Distributed Resource Management
    - LSF
    - Loadleveler
  - Resource Management
    - Auditing
    - Batch Queue
    - Time
  - Code Compilation and Loading
    - Compilers
      - GNU
    - YOD
  - Data Security
    - Policy
    - SSH
    - IPSEC
  - User Notification
    - Mail
    - Web

Key User Services

- User Authentication
  - Terminal Access
    - Telnet
    - SSH
- Parallel Operating Environment
- Code Support (libraries)
- Interactive Access
- Batch Access
  - Load Leveler
  - LSF
- Mesh Generation
- Code Compilation
  - IBM Standard AIX Fortran 77
  - IBM Standard AIX C
  - IBM Standard AIX C++
- Loaders
  - Fortran

- C
- C++
- Code Debugging
  - IBM Parallel Debugger
- Editors
  - vi
  - Emacs
- Compute Cycles (TOPS)
- Simulation Monitoring and Steering
- File Services (moving/sharing/archiving)
  - File transfer FTP/PFTP/RCP
  - File mounting NFS/DFS
  - File Storage/Archive (HPSS)
- Output Services
  - X
  - MPEG
  - JPEG
  - IP Video Streaming
  - Taping
  - Printing
- System Status
  - Mail
  - Web
  - Data file
- Visualization Services
  - SGI Servers
  - Ensight
  - AVS/Express
  - IBM Data Explorer

Key Network Services

- Data Encryption
  - IPSEC
  - LINK
- Firewalling
- IP routing
- DNS Service
- Time Service
- ARP
- LANE
- Data Transport
  - Ethernet
  - IP
  - ATM
- MPI

Distribution:

1 - U.S. Department Of Energy Headquarters
Forrestal Building
David Luginbuhl
Route DP-51, MS 4A-024
1000 Independence Avenue SW
Washington, DC 20585

1 - U.S. Department Of Energy Headquarters
Forrestal Building
Paul Messina
Route DP-51, MS 4A-024
1000 Independence Avenue SW
Washington, DC 20585

1 - University of California
Lawrence Livermore National Laboratory
David P. Wiltzius, L-060
7000 East Avenue
P. O. Box 808
Livermore, CA 94550

1 - University of California
Lawrence Livermore National Laboratory
Teresa M. "Terri" Quinn, L-0601
7000 East Avenue
P. O. Box 808
Livermore, CA 94550

1 - University of California
Lawrence Livermore National Laboratory
Virginia W. 'Jean' Shuler, L-067
7000 East Avenue
P. O. Box 808
Livermore, CA 94550

1 - University of California
Lawrence Livermore National Laboratory
Marcus Miller, L-060
7000 East Avenue
P. O. Box 808
Livermore, CA 94550

1 - Los Alamos National Laboratory
Steve Tenbrink, MS B255
CIC-5: Network Engineering
Los Alamos, New Mexico 87545

1 - Los Alamos National Laboratory
Mitchel W. Sukalski, MS B255
CIC-5: Network Engineering
Los Alamos, New Mexico 87545

1 - Los Alamos National Laboratory
Denny Rice, MS B255
Los Alamos, New Mexico 87545

1 - Los Alamos National Laboratory
Curt Canada, MS B287,
CIC-ACL: CIC Advanced Computing Lab.
Los Alamos, New Mexico 87545

1 - 0318 A. R. Breckenridge, 9215
1 - 0318 P. D Heermann, 9215
1 - 0318 V. P. Holmes, 9215
1 - 0321 A. L. Hale, 9224
1 - 0321 J. A. Ang, 9224
1 - 0321 W. J. Camp, 9204
1 - 0469 J. F. Jones, Jr., 4600
1 - 0806 M. O. Vahle, 4616
1 - 0806 J. P. Brenkosh, 4616
1 - 0806 L. B. Dean, 4616
25 - 0806 M. J. Ernest, 4616
1 - 0806 S. A. Gossage, 4616
1 - 0806 L. G. Martinez, 4616
1 - 0806 J. H. Naegle, 4616
1 - 0806 L. G. Pierson, 4616
10 - 0806 T. J. Pratt, 4616
1 - 0806 J. A. Schutt, 4616
1 - 0806 L. Stans, 4616
1 - 0806 T. D. Tarman, 4616
1 - 0806 C. D. Brown, 4621
1 - 0806 G. D. Machin, 4621
1 - 0807 S. D. Nelson, 4417
1 - 0807 R. M. Cahoon, 4418
1 - 0807 M. L Barnaby, 4418
1 - 0807 R. K. Byers, 4418
1 - 0807 S. P. Goudy, 4418
1 - 0807 J. H. Laros, 4818
1 - 0807 J. P. Noe, 4418
1 - 0807 D. N. Shirley, 4418
1 - 0807 W. D. Vandevender, 4418
1 - 0812 M. R. Sjulin, 4914
1 - 0812 L. F. Tolendino, 4914
1 - 0820 P. Yarrington, 9232
1 - 0826 J. D. Zepper, 9136
1 - 0833 J. H. Biffle, 9103
1 - 0841 P. J. Hommert, 9100
2 - 0899 Technical Library, 4916
1 - 0957 C. S Leishman, 1401
1 - 1110 R. B. Brightwell, 9223
1 - 1110 L. A. Fisk, 9223
1 - 1110 J. P. VanDyke, 9223
1 - 1110 R. E. Riesen, 9224
1 - 1138 J. I. Beiriger, 6532
1 - 1138 A. L. Hodges, 6531
1 - 1138 W. R. Johnson, 6531
1 - 1138 L. J. Ellis, 6531
1 - 1138 B. N. Malm, 6532
1 - 1138 S. K. Chapa, 6533
1 - 9003 D. L. Crawford, 5200
1 - 9011 P. W. Dean, 8903
2 - 9019 Central Technical Files, 8940
2 - 0619 Review & Approval Desk, 00111
For DOE/OSTI