SAND--97-0463C
CONF-9706112--1

ASCI Red - Experiences and Lessons Learned with a Massively Parallel TeraFLOP Supercomputer (U)

Mark A. Christon, David A. Crawford, Eugene S. Hertel,
James S. Peery, and Allen C. Robinson

Computational Physics R&D Department
Sandia National Laboratories
Albuquerque, New Mexico 87185-0819
E-mail: machris@sandia.gov

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Abstract

The Accelerated Strategic Computing Initiative (ASCI) is focused upon advancing three-dimensional, full-physics calculations to the point where "full-system" simulation may be applied to virtual testing. The ASCI program involves Sandia, Los Alamos and Lawrence Livermore National Laboratories. At Sandia National Laboratories, ASCI applications include large deformation transient dynamics, shock propagation, electromechanics, and abnormal thermal environments. In order to resolve important physical phenomena in these problems, it is estimated that meshes ranging from 10^6 to 10^9 grid points will be required. The ASCI program is relying on the use of massively parallel supercomputers initially capable of delivering over 1 TFLOPs to perform such demanding computations. The ASCI "Red" machine at Sandia National Laboratories consists of over 4500 computational nodes (over 9000 processors) with a peak computational rate of 1.8 TFLOPs, 567 GBytes of memory, and 2 TBytes of disk storage. Regardless of the peak FLOP rate, there are many issues surrounding the use of massively parallel supercomputers in a "production" environment. These issues include parallel I/O, mesh generation, visualization, archival storage, high-bandwidth networking and the development of parallel algorithms. In order to illustrate these issues and their solution with respect to ASCI Red, demonstration calculations of time-dependent buoyancy-dominated plumes, electromechanics, and shock propagation will be presented. The applications issues and lessons learned to-date on the ASCI Red machine will be discussed.

1 Introduction

The Accelerated Strategic Computing Initiative (ASCI) is a multi-laboratory program that involves Sandia, Los Alamos and Lawrence Livermore National

Laboratories and is sponsored by the United States (U.S.) Department of Energy. The ASCI program is focused upon the creation of advanced simulation capabilities that are essential to maintaining the safety and reliability of the U.S. nuclear stockpile. ASCI is intended to advance three-dimensional, full-physics codes to the point where "full-system" simulation may be applied to virtual testing and prototyping. In order to achieve a full-physics simulation capability, the ASCI program is relying upon massively parallel supercomputers initially capable of a peak computational rate of over 1 TFLOPs (10^{12} floating point operations per second). Although the program is stimulating the U.S. computer manufacturing industry to develop more powerful high-end supercomputers, it is also creating a computational infrastructure and an operational environment that makes "TeraScale" capabilities usable.

The ASCI program was initiated with the procurement of the "Red" machine. Figure 1 shows an artist's rendering of the complete machine and a photograph of the machine as it is currently installed. The MP-Linpack record of 1.05 TFLOPs was achieved on the "three-row" configuration in December, 1996. In addition to the ASCI Red machine, two 0.1 TFLOPs systems have been delivered to Los Alamos and Lawrence Livermore National Laboratories as a prelude to the two "Blue", 3 TFLOPs computers that will be sited at these laboratories. The ASCI Red machine is a distributed memory machine, while the ASCI Blue machines are based upon clusters of symmetric multiprocessors.

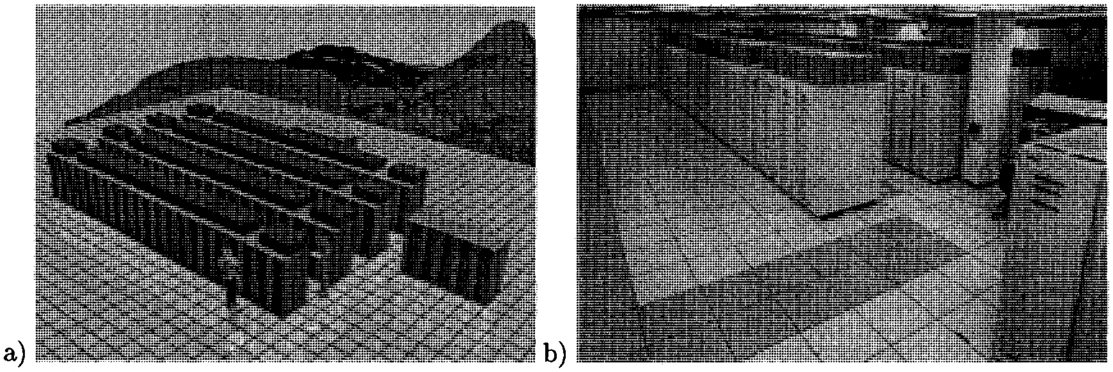

Figure 1: a) Artist's rendering of the ASCI RED machine with the Sandia mountains as a background, b) the ASCI Red machine as of March 31, 1997.

At Sandia National Laboratories, ASCI applications include, but are not limited to electromechanics, large deformation transient dynamics, shock physics, and the simulation of abnormal thermal environments. In order to resolve both important physical phenomena and geometrical features in these problems, it is estimated that mesh resolution ranging from 10^6 to 10^9 grid points will be required for ASCI problems. In addition to the need for high resolution meshes,

there is a concomitant requirement for adequate temporal resolution that ultimately demands scalable, parallel applications.

In order to meet the spatial and temporal resolution requirements, there are many algorithm-to-architecture issues that have to be addressed to achieve parallel scalability. In addition, there are infrastructure issues surrounding the use of massively parallel supercomputers in a production setting. These issues include parallel I/O, mesh generation, visualization, archival storage, and high-bandwidth networking. Before embarking on a discussion of these issues, the ASCI Red architecture, networking, archival storage and user environment are presented. Next, several calculations showing incompressible flow, electromechanics, and shock propagation are presented. In the context of these calculations, the lessons learned to-date with respect to scalable parallel applications, parallel I/O, and visualization on ASCI Red will be discussed.

2 The ASCI Red machine

This section describes the ASCI Red architecture, networking, archival storage and user environment [URL1]. The fully configured ASCI Red machine will consist of 4536 computational nodes with 567 GBytes of memory (64 MBytes/processor). In its final configuration, the Red machine will consist of 85 cabinets in four rows that occupy 1600 square feet as shown in Figure 1a. As of March, 1997, a 1.4 peak TFLOPs machine with 3536 processors and 442 GBytes of memory has been installed at Sandia (Figure 1b). The system configuration is summarized below in Table 1.

The ASCI Red compute nodes are comprised of dual 200 MHz Intel Pentium Pro processors with a total of 128 MBytes of memory. Two compute nodes, i.e., four processors, reside on a "Kestrel" board along with the VLSI Network Interface Component (NIC). With the exception of the NIC, all of the components on the Kestrel board are commodity parts. The inter-processor communication network is configured in a two dimensional, 2-plane, mesh topology with a bi-directional bandwidth of 800 MBytes/second. The cross-sectional bandwidth of the inter-processor network, 51 GBytes/second, is a measure of the total potential message traffic capacity between all nodes in the system.

The ASCI Red Network and Archival Storage

Interactive access to the ASCI Red machine is provided via service nodes that consist of two Pentium Pro processors that share 256 MBytes of memory. Like the compute nodes, there are two service nodes per Kestrel board. In contrast to the compute and service nodes, the I/O nodes consist of two Pentium Pro processors that share 256 MBytes of memory on an "Eagle" board. The network nodes also reside on Eagle boards and provide the Asynchronous Transfer Mode

ASCI Red System Feature	Characteristic
System peak FLOP rate	1.8 TFLOPs
Number of compute nodes	4536
Number of service nodes	32
Number of I/O nodes	36
Number of system nodes	2
Number of network nodes	6
Total number of Pentium Pro processors	9224
System RAM Memory	567 GBytes
Processor L1 instruction / data cache	8 KBytes / 8 KBytes
Processor to L1 bandwidth	6.4 GBytes/second
Processor L2 integrated cache	256 KBytes
Processor to L2 bandwidth	1.6 GBytes/second
PCI bus clock rate	33 MHz
Logical mesh topology (2 planes)	38 × 32 × 2
Bi-directional network bandwidth	800 MBytes/second
Cross-sectional network bandwidth	51 GBytes/second
RAID I/O bandwidth (each subsystem)	1 GBytes/second
RAID storage (total)	2 TBytes
System footprint	1600 ft^2
Number of cabinets	85
Power consumption	800 KWatts

Table 1: Summary of ASCI Red system features.

(ATM) network interface. The ATM connections are currently planned to be OC-12 with a bandwidth of 620 MBytes/second. Figure 2 illustrates the unclassified network configuration for the ASCI Red machine, its application and visualization servers, and the archival storage system (HPSS).

Archival storage for the Red machine is provided by the Scalable Mass Storage System (SMSS) hardware that is currently configured with more than 27 TBytes of disk and uncompressed, high speed, automated tape storage. The SMSS uses the 10 GByte IBM 3590 tape and 3494 library technology for its tape storage. The tape storage system uses eight drives that can transfer data at 10 MBytes/second. With advanced hardware compression capabilities in the 3590 technology, the usable capacity is anticipated to be greater than 50 TBytes (2 - 3 times the uncompressed capacity).

The High Performance Storage System (HPSS) software provides access to the SMSS hardware. The HPSS is a distributed, hierarchical storage management software system that scales, in terms of performance and capacity, through modular software components. All SMSS systems are connected to both Ether-

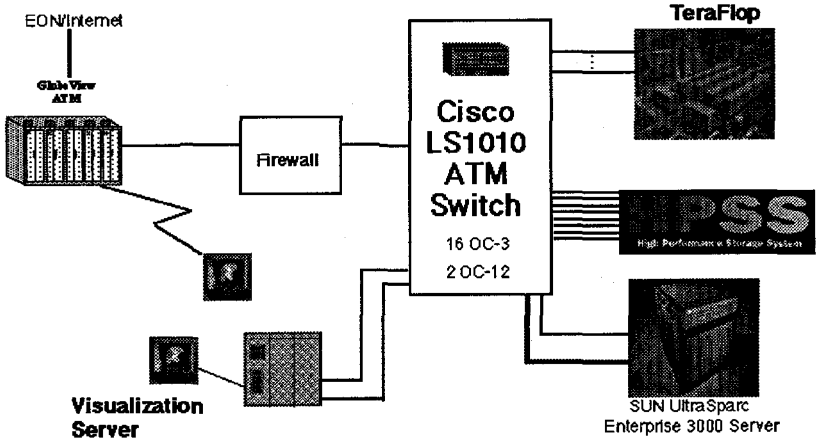

Figure 2: Unclassified network configuration for ASCI Red.

net and ATM networks in order to permit control communication among HPSS components on the SMSS to be routed through the Ethernet network while data transmission to client systems occurs via OC-3 ATM connections. Each "data mover" is capable of transferring TCP/IP network traffic at 11 MBytes/second. By using the software striping capabilities of the HPSS and writing data over six tape volumes concurrently, sustained data transfer rates greater than 50 Mbytes/second have been achieved. By default "large" files are written directly to tape while smaller files are written to disk and then migrated to tape.

The ASCI Red User Environment

The ASCI Red machine consists of a center section of compute nodes that can be switched between a classified and an unclassified Local Area Network (LAN). In addition, each end of the the Red machine owns a portion of the service, I/O, and system nodes, as well as, a significant number of computational nodes. Each end of the ASCI Red machine constitutes a significant parallel computer in its own right, with a peak computational rate of approximately 370 GFLOPs. On each LAN there is a four-processor UltraSparc application server with 250 GBytes of file storage, and an archival storage system (SMSS/HPSS).

There are two operating systems on the ASCI Red machine. The "TFLOPs Operating System" is a distributed OSF UNIX that runs on the service and I/O nodes. The TFLOPs Operating System is used for boot and configuration support, system administration, user logins, user commands and services, and development tools. The Cougar operating system, a descendant of the SUNMOS [19] and PUMA [25, 28] operating systems, runs on the compute nodes. Cougar is a very efficient, high-performance operating system providing program loading, memory management, message-passing, signal and exit handling, and run-time support for applications languages. Cougar is also very small, occupying less that 500 KBytes of memory (RAM).

Although ASCI Red is a distributed-memory, Multiple-Instruction Multiple-Data (MIMD) machine, it supports both explicit and implicit parallel programming models. In the explicit parallel model, data structures are manually decomposed into sub-units and distributed among the computational nodes of the machine. The code executes on each compute node, and messages are passed between the nodes using a message-passing protocol, i.e., MPI [9] or NX [16, 20, 21], to coordinate the calculation. MPI and NX are layered on top of portals [3], a Cougar communications protocol that delivers optimal-performance for both message-passing protocols. In addition, to MPI and NX, the one-sided communication interface from MPI 1.2 is supported.

In contrast to the explicit parallel programming model, the implicit parallel model typically requires source level directives to inform the compiler about the data structures that will be distributed to the compute nodes. The compiler manages the data distribution and parallel processing. Implicit parallel processing will be supported via the HPF programming language [13] on the Red machine.

Currently, C, C++ (with standard templates), Fortran 77, Fortran 90, and HPF are supported on the compute nodes of ASCI Red. The C, C++, Fortran 77, and HPF compilers are native compilers developed by Portland Group International. However, the Fortran 90 compiler is the familiar NAG F90 compiler. An interactive, parallel debugger is available with both a character interface and an X-window interface.

3 Parallel Applications

There are a large number of parallel applications already being exercised on the ASCI Red machine. The Sandia applications discussed below have all successfully used an explicit domain-decomposition message-passing model to achieve scalable, parallel performance.

Electromechanical Applications

ALEGRA [URL2] is a parallel, Arbitrary Lagrangian-Eulerian (ALE) code, written in a mixture of C++, C, and Fortran 77, that provides a general framework for implementing coupled, multi-physics solution algorithms. The code is capable of treating 1-D, 2-D and 3-D geometries and makes use of advanced second-order accurate, unstructured grid algorithms for advection and interface reconstruction [27].

The simulation of coupled, electromechanical phenomena is of the utmost importance in understanding mission critical applications at Sandia National Laboratories. Both quasi-static magnetic and electric field approximations of Maxwell's equations are of interest. Parallel, preconditioned iterative solvers [15] are used for the solution of electric and magnetic field equations. The coupling between the electric and magnetic field equations and shock physics (or solid dynamics) is achieved at the continuum level. It is also necessary to couple the continuum models with lumped element circuit models to account for external electrical components.

The quasi-static magnetic field approximation and the continuum equations of mechanics describe magnetohydrodynamic (MHD) phenomena. Figure 3a illustrates how an azimuthal magnetic field arising from axial currents drive a cylindrical implosion. The quasi-static electric field approximation coupled with continuum mechanics allows the simulation of ferroelectric power supplies. An example of this type of computation is shown in Figure 3b where explosive energy provides the shock compression of X-cut quartz to drive an electrical signal.

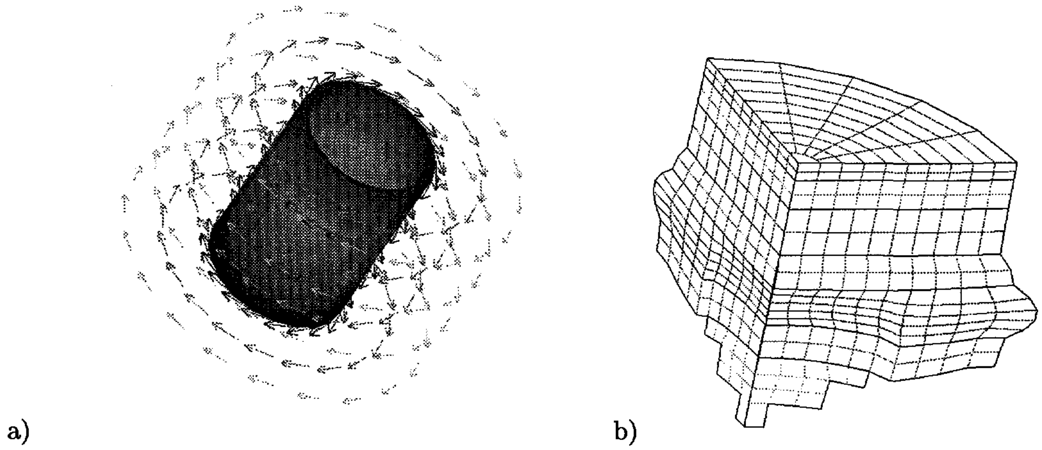

Figure 3: a) Z-Pinch simulation [URL2], b) Shock compression of X-cut quartz.

Shock Physics

CTH [URL3] is an Eulerian computational shock physics code capable of treating 1-D, 2-D and 3-D geometries. CTH uses a logically structured grid and includes the capability to simulate both multi-phase and mixed-phase materials, elastic-viscoplastic solids, and fracture. At the time of this writing, CTH is used to perform computations with grid sizes ranging from 10^7 - 10^9 cells on the ASCI Red machine. A representative CTH calculation is described here.

In 1980, Luis Alvarez, et al. [1] demonstrated that an asteroid, about 10 km in diameter, crashed into the Earth 65 million years ago. Since 1980, numerous investigations have demonstrated that impact events of this magnitude happen on average every 100 million years. These events have caused extreme stress to the Earth's climate and resulted in the extinction of many species, including dinosaurs. This has prompted the question whether events of the magnitude of the recent Comet Shoemaker-Levy 9 impact with Jupiter could cause similar stress on Earth's climate.

A simulation using CTH was performed on the ASCI Red machine to answer this question. The calculation, consisting of 54 million grid points, ran for 48 hours on 1500 nodes of the ASCI Red machine (about one-third of its final configuration). The simulation involves a 1 km diameter comet (weighing about 1 billion tons) traveling 60 km/s and impacting Earth's atmosphere at a 45^o angle (the most probable angle of impact). As shown in Figure 4a, the comet produces a strong luminescent bow shock in the atmosphere as it proceeds downwards. After 0.7 seconds, the badly deformed comet hits the water surface (Figure 4b) forming a large transient cavity in the ocean (Figure 4c). The comet and large quantities of water are vaporized by the tremendous energy of the impact (equivalent to 300×10^9 tons of TNT) and ejected into sub-orbital ballistic trajectories (Figure 4d).

The simulated impact scenario occurs approximately once every $300,000$ years. Information from simulations like this, specifically, estimates of the amount of water and other materials injected into the stratosphere, will be used in global climate simulations to estimate the stress on Earth's biosphere. In the near future, simulations such as this will be able to determine the risk that present day species, including humans, face from asteroid and comet impacts.

Incompressible Fluid Dynamics

GILA is a computational fluid dynamics code for performing transient, viscous, incompressible flow calculations that require high resolution, unstructured meshes. GILA includes a collection of algorithms for performing both explicit [4] and semi-implicit [5] time integration. The explicit time integration algorithms uses constant-gradient, stabilized, linear finite elements technology. The semi-implicit family of algorithms for the Navier-Stokes equations are founded upon a second-order projection that provides very high phase accuracy for advective transport, while maintaining a legitimate segregation of the velocity and pressure. Both the explicit and semi-implicit algorithms rely upon very fast iterative solvers for the pressure Poisson equation.

The pressure solvers in GILA include scalable, parallel iterative solvers based upon sub-domain preconditioners (SSOR and Cholesky sub-domain preconditioners), and alternatively, a parallel saddle-point solver tailored for the incompressible Navier-Stokes equations. GILA also includes a large eddy simulation

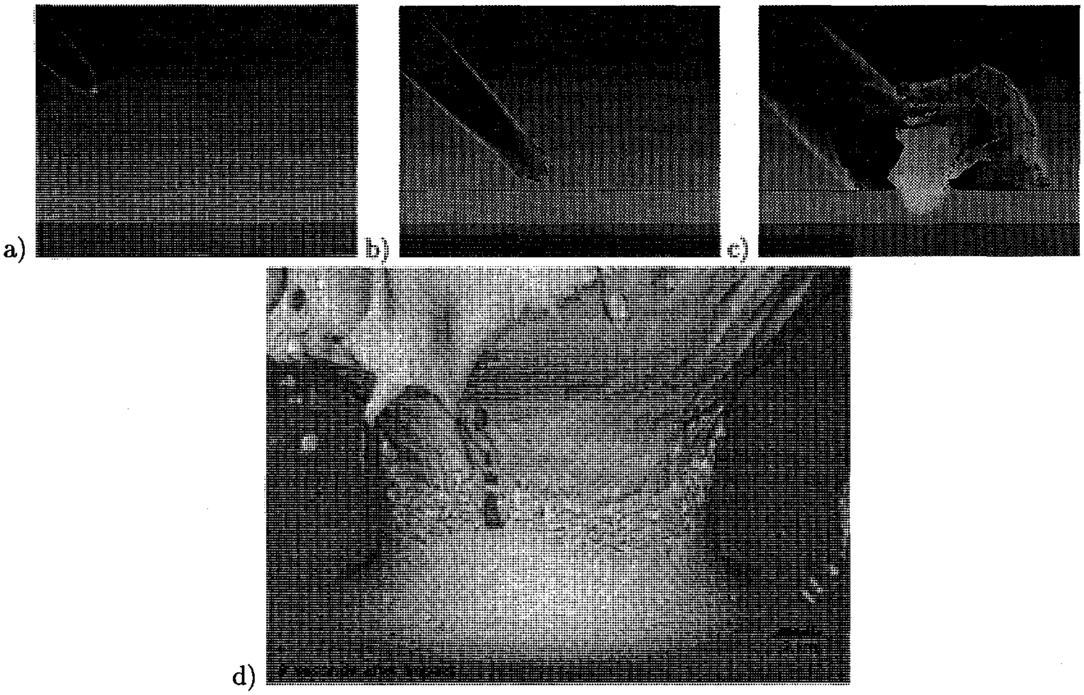

Figure 4: a) A 1 km diameter comet enters Earth's atmosphere, b) The deformed comet just prior to impact with the ocean, c) The transient cavity filled with high pressure steam explodes into the stratosphere, d) Comet and water vapor is ejected onto globe-straddling ballistic trajectories [URL3].

(LES) model that has been exercised on exterior flow problems with Reynolds numbers as high as 10^5. GILA has been exercised on over 800 processors of the ASCI Red machine with parallel efficiencies ranging from about 75% to 95% depending upon the problem and quality of the mesh decomposition.

Figure 5 shows several snapshots from a GILA simulation of a time-dependent, buoyancy dominated plume. In this calculation, the Froude number is 2.2×10^{-3}, and the Reynolds number is 555 based upon the 0.2 m plume diameter. This simulation used over 10^6 elements partitioned across 512 compute nodes on ASCI Red. The measured puffing frequency was approximately 3.3 Hz corresponding precisely to the empirical frequency, $f = 1.5/\sqrt{D}$, where D is the plume base diameter. This type of plume computation can provide critical information about the atmospheric dispersal of pollutants.

MPSalsa [URL4] is a three dimensional, massively parallel, chemically reacting flow code that has been developed for the steady and transient simulation of incompressible and low Mach number compressible fluid flow, heat transfer, and

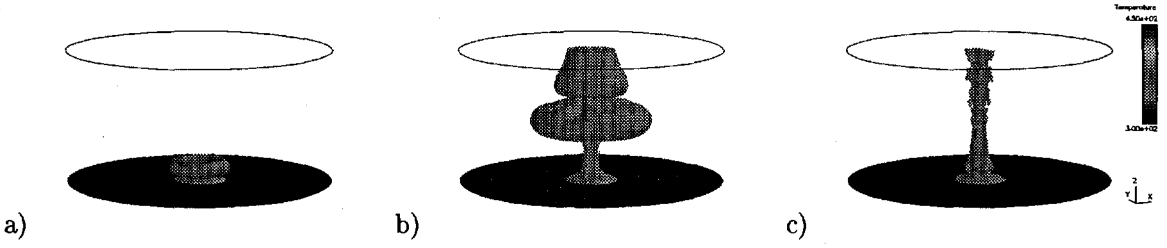

Figure 5: Temperature iso-surfaces for time-dependent, buoyant plume. a) The plume at 0.5 seconds, b) 1.5 seconds, and c) 4.75 seconds.

mass transfer with bulk fluid and surface chemical reaction kinetics. MPSalsa uses a Galerkin least-squares finite element formulation for the governing Navier-Stokes and transport reaction equations [24]. The numerical solution methods are based on efficient and robust iterative solution methods using a fully-coupled inexact Newton method with preconditioned Krylov solvers [15]. Current numerical research is focused on adaptive mesh techniques and multilevel preconditioners [URL5] that are being developed to increase solution accuracy and the underlying convergence rate of the iterative solvers. Current physics development is focused upon the inclusion of turbulence models with chemical reactions, enclosure and participating media radiation, plasma chemistry and multi-physics capabilities.

MPSalsa has been used to simulate the chemical vapor deposition of a number of advanced materials including, silicon carbide (SiC), silicon nitride (SiN3) and gallium arsenide (GaAs). The reaction mechanisms have included up to 20 gas phase and 10 surface phase chemical species undergoing over 60 chemical reactions [23]. Figure 6a shows steady-state streamlines and GaMe3 contours in a horizontal CVD reactor with a tilted susceptor and rotating substrate. Figure 6b shows an instantaneous constant temperature surface produced from a hydrodynamic instability in a rotating disk CVD reactor. This is a two-fold spatio-temporal instability caused by the interaction of rotational and buoyancy forces. MPSalsa has been exercised on over 1000 compute nodes of the ASCI Red machine using meshes with over 1 million elements for CVD , a ion source tube and a number of nuclear nonproliferation simulations.

Parallel Visualization

In order to enable the interrogation and rapid assimilation of the massive data sets being produced by simulations performed on the ASCI Red machine, researchers at Sandia are working to implement scalable, parallel, visualization tools [URL6]. Although other distributed strategies for visualizing large data sets are also being considered, several parallel tools are currently being imple-

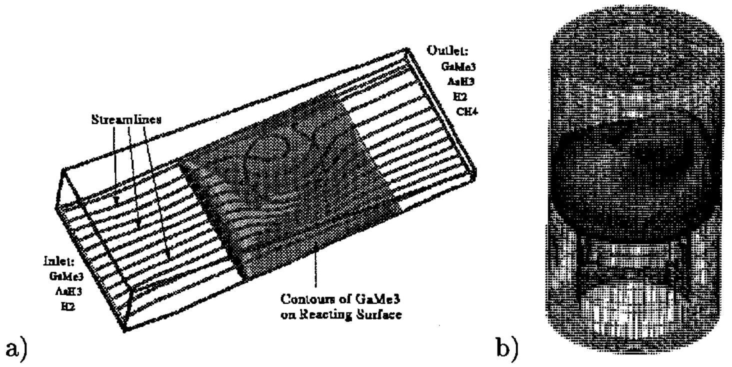

Figure 6: MPSalsa simulations [URL4] showing a) steady-state streamlines and GaMe3 contours in a horizontal tilt CVD reactor, b) flow instability in a rotating disk CVD reactor.

mented directly on the ASCI Red machine to enable in-situ visualization of machine capacity data sets thereby avoiding the need to move the data prior to visualization.

The tools being migrated to the Red machine include an isosurface generator with polygon-decimation and a parallel volume rendering system for structured (e.g., rectilinear) volume data sets. The parallel volume rendering system has been used to successfully render volumes as large as 5×10^8 voxels at pseudo-interactive rates (i.e., a frame every 2 - 5 seconds) on Sandia's Intel Paragon.

Figure 7 illustrates output from the the parallel volume rendering tool. The parallel volume renderer has been integrated with a front-end preview renderer which operates on a low resolution copy of the data – the parallel renderer is invoked on demand to render the full resolution data from desired viewpoints. This paradigm of being able to process data at multiple resolutions while ultimately being able to process the full resolution data using high-end parallel machines is very attractive, and future research is expected to exploit this capability. Other parallel visualization tools intended for the ASCI Red machine include a polygon and volume renderer for unstructured grids.

4 Applications Issues and Lessons Learned

This section presents an overview of the simulation-based analysis cycle and identifies the serial processes in the cycle. The approaches being taken to resolve the serial bottlenecks in order to perform parallel calculations with problem sizes ranging from 10^6 to 10^9 grid points are discussed.

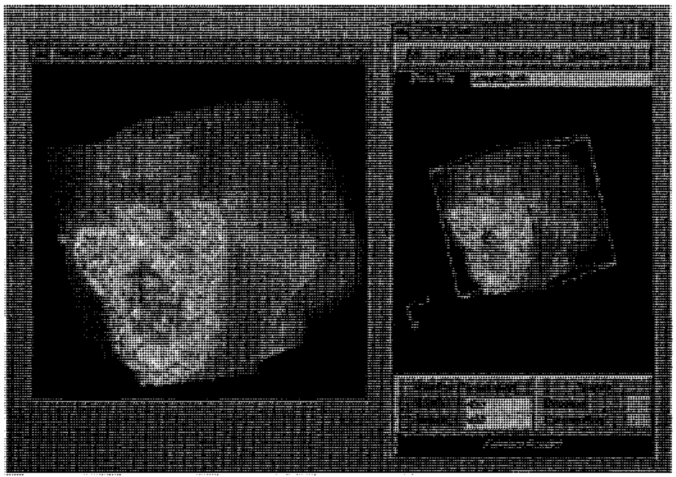

Figure 7: Parallel volume rendering tool [URL6] showing a low-resolution rendering (right panel), and the high-resolution parallel rendering (left panel). The volume rendering illustrates the the protein concentration in a human cell.

The Analysis Cycle

The traditional analysis process, shown in Figure 8, consists of a cycle in which the analyst probes the sensitivity of a simulated physical process to the problem parameters and grid resolution. To date, the analysis cycle has relied upon balanced, general purpose computers. Although massively parallel computers are being touted as general purpose supercomputers, the ratio of the maximum to minimum delivered FLOP rates, i.e., the specialty ratio [14], is generally quite high for massively parallel supercomputers. Therefore, obtaining a large fraction of the peak FLOP rate on a massively parallel supercomputer requires careful consideration of hierarchical memory access, e.g., programming for cache re-use (see Dowd [6]), thread safety for compute nodes consisting of symmetric multiple processors, as well as the appropriate use of domain-based parallelism. Achieving a balance between compute power, I/O bandwidth, disk and memory capacity has proven difficult with massively parallel machines. Furthermore, mesh generation, domain decomposition, and visualization are still primarily serial processes. Thus, simply scaling the analysis model for massively parallel supercomputers is not practical.

Mesh Generation and Domain Decomposition

Unstructured grid applications rely on separate codes to perform the "automatic" mesh generation and subsequent domain decomposition. The mesh gen-

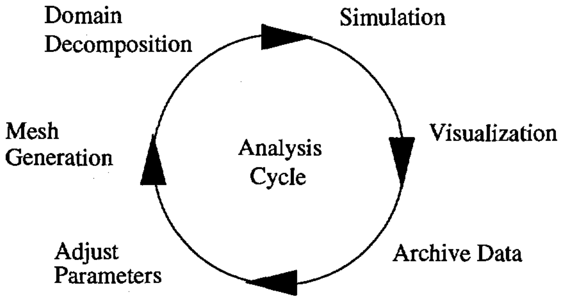

Figure 8: The components of the traditional analysis cycle.

erators at Sandia National Laboratories rely upon algebraic and advancing-front methods such as paving [2], and plastering [26]. These algorithms have remained predominantly serial, and have not scaled to grid sizes beyond $O(10^6)$ elements. Domain decomposition for unstructured grids relies heavily upon methods such as recursive spectral bisection [7, 11, 12, 22], and is typically performed in serial on workstations. Computational experiments with CHACO [URL7] have indicated that meshes up to $O(10^7)$ elements can be decomposed on workstations, although parallel graph partitioning codes such as METIS [17, 18] are becoming available. In contrast, for structured-grid codes, such as CTH, the mesh generation and domain-decomposition steps are simplified by the logically regular mesh. This has permitted CTH to scale up to mesh sizes with $O(10^9)$ grid points.

In order to overcome the mesh generation and decomposition problem, adaptivity is being explored as a way to use relatively coarse grids that can be easily generated and decomposed in serial on workstations. This approach uses a mesh that adequately captures the geometry, and can be rapidly decomposed. The sub-domain meshes are then refined in parallel on each processor in order to achieve the desired grid resolution. This approach also avoids the need to store and manage the data files associated with a high resolution mesh.

The Data Explosion - I/O Limitations

It is anticipated that a transient, unstructured grid calculation with 10^8 - 10^9 grid points will generate $O(10)$ TBytes of data. Thus, the ability to generate data far exceeds the current $O(1)$ TBytes disk capacity of the Red machine. Unfortunately, the problem of finite disk resources is compounded by limited parallelism in the I/O subsystem. The ASCI Red machine provides 18 I/O nodes to service the I/O requests of approximately 4500 compute nodes. From a programming point of view, this demands I/O "staging" whereby only a group of processors can perform I/O while the remaining processors are idle.

In order to obtain 1 GBytes/second I/O bandwidth on the Red machine, the parallel file system requires large blocks of data to be written. Figure 9 shows the results of a single-node benchmark that compares several methods of writing to the parallel file system. The *cwrite* and *fwrite* functions provide the best performance yielding a bandwidth of approximately 20 MBytes/second. Unfortunately, the high-level libraries, NetCDF and HDF do not provide acceptable levels of performance. Note that the results shown in Figure 9 were computed using an early release of the system software, and the peak I/O bandwidth from a single node is expected to be approximately 55 to 60 MBytes/second.

From Figure 9, the number of words required to achieve a reasonable output bandwidth ranges from 10^4 to 10^5. This effectively prescribes the degree of granularity required for the staged I/O and is driving applications towards the use of "collective" I/O. Efforts are already underway to develop a high performance I/O library that permits parallel collective I/O operations in order to achieve the bandwidth of the system. In addition, an application interface to the HPSS is being developed to permit directed data migration from a running simulation.

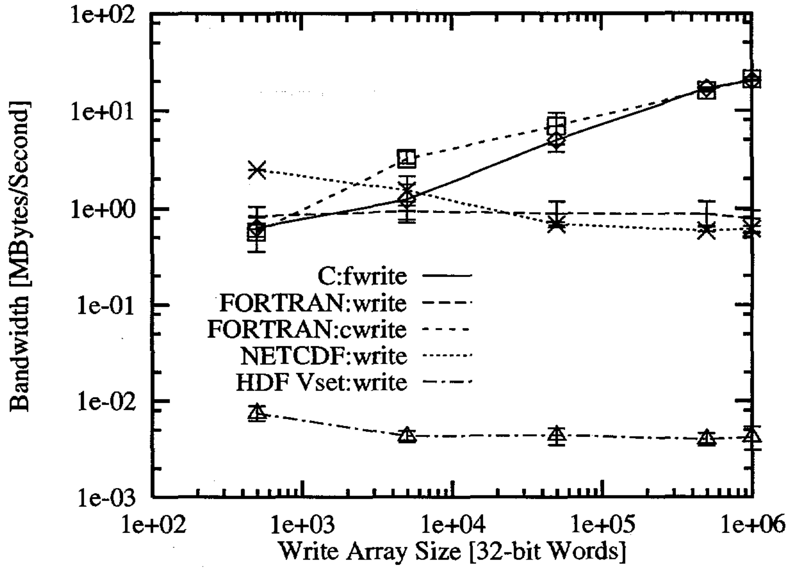

Figure 9: Single-node I/O bandwidth using "work-week 2" software.

Scientific Visualization

The scientific visualization tools available today are serial, and for the most part incapable of treating problems with 10^8 to 10^9 grid points – particularly for time-dependent problems. This issue is complicated by the fact that during the interrogation of simulation results, the visualization study may motivate the user to look at many different derived variables, e.g., effective stress, vorticity, etc. The need for data abstraction demands that sufficient primitive data be available to compute the necessary results.

In order to solve the the visualization bottleneck, there have been several proposed modifications to the current analysis paradigm. The first involves "gutting" the data and retaining only the external surface and its data for rendering. While this is adequate for some solid mechanics problems, this approach clearly falls in the category of ways to say nothing with scientific visualization (see Globus [8]).

The second approach, "co-processing", as proposed by Haimes [10] suggests that visualization be done "on-the-fly" while the simulation is running. This approach requires the user to select the data or derived variable to be visualized at the time the simulation is run. Unfortunately, efforts to implement interactive visualization codes on ASCI Red have been hampered by the lack of standard UNIX networking protocols from the compute nodes to external processes.

A variation of the co-processing model proposes using a light-weight, parallel, polygonal-based renderer that is called during the simulation to generate images with multiple views, derived quantities and visualization methods. In this model, images are written to disk for perusal at a later time with a "flip book". Ultimately, machine capacity simulations may require this type of on-processor rendering during simulation since 1 TByte of disk translates to $O(10^5)$ 24-bit, uncompressed images at 1024×768 resolution - a far more efficient use of disk space. This is one example of a growing attitude toward treating the simulation as a "virtual experiment". In this model, a computational experiment is instrumented very much like a physical experiment and relies upon virtual instrumentation, e.g., virtual transducers and imaging algorithms to measure the response of important physical variables and fields. The amount of data available is limited by finite bandwidth and storage capacity, and the output must be heavily optimized to provide the most useful information.

5 Summary

The ASCI Red machine at Sandia National Laboratories promises to provide a significant resource for "virtual experimentation" and design development. Many parallel codes have been developed and the use of parallel computing for production simulation of a wide variety of physics is becoming the norm. New infrastructure challenges arise with the ASCI Red machine and these challenges are being met with innovative algorithmic approaches and methods for treating simulation results.

Acknowledgments

The authors wish to acknowledge Dan Carroll, Pang Chen, Karen Devine, Marlin Kipp, Rena Haynes, Constantine Pavlakos, John Shadid, and Walt Vandevender for their contributions to the manuscript. This work was supported by the United States Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-ACO4-94AL85000.

References

[1] L. W. Alvarez, W. Alvarez, F. Asaro, and H. V. Michel, *Extraterrestrial cause of the cretaceous/tertiary extinction: Experimental results and theoretical interpretation*, Science, 208 (1980), pp. 1095–1108.

[2] T. D. Blacker and M. B. Stephenson, *Paving: a new approach to automated quadrilateral mesh generation*, Tech. Rep. SAND90-0249, Sandia National Laboratories, 1990.

[3] R. Brightwell and L. Shuler, *Design and implementation of mpi on puma portals*, in Second MPI Developer's Conference, July 1996, pp. 18 – 25.

[4] M. A. Christon, *A domain-decomposition message-passing approach to transient viscous incompressible flow using explicit time integration*, to appear in Computer Methods in Applied Mechanics and Engineering, (1996).

[5] ——, *Domain-based parallelism and the projection algorithm for transient, viscous incompressible flow*, in preparation for Computer Methods in Applied Mechanics and Engineering, (1997).

[6] K. Dowd, *High Performance Computing*, O'Reilly and Associates, Inc., Sebastapol, California, 1993. pp. 5-6.

[7] C. Farhat and M. Lesoinne, *Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics*, International Journal for Numerical Methods in Engineering, 36 (1993), pp. 745–764.

[8] A. Globus and E. Raible, *13 ways to say nothing with scientific visualization*, Tech. Rep. RNR-92-006, NASA Ames Research Center, February 1992.

[9] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI - Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, Cambridge, Massachusetts, 1994.

[10] R. HAIMES, *pv3: A distributed system for large-scale unsteady cfd visualization*, Tech. Rep. 94-0321, AIAA, January 1994.

[11] B. HENDRICKSON AND R. LELAND, *An improved spectral graph partitioning algorithm for mapping parallel computations*, Tech. Rep. SAND92-1460, Sandia National Laboratories Report, September 1992.

[12] ——, *A multilevel algorithm for partitioning graphs*, Tech. Rep. SAND93-1301, Sandia National Laboratories Report, October 1993.

[13] HIGH PERFORMANCE FORTRAN FORUM, *High Performance Fortran Language Specification, Version 1.1*, November 1994. (http://www.crpc.rice.edu/HPFF/hpf1/index.html).

[14] R. W. HOCKNEY, *The Science of Computer Benchmarking*, SIAM, Philadelphia, Pennsylvania, 1996. pp. 5-6.

[15] S. A. HUTCHINSON, J. N. SHADID, AND R. S. TUMINARO, *Aztec user's guide*, Tech. Rep. SAND95-1559, Sandia National Laboratories, October 1995.

[16] INTEL CORPORATION, *Paragon Fortran System Calls - Reference Manual*, no. 312488-BETA, March 1994.

[17] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning*, Tech. Rep. TR95-035, University of Minnesota, Department of Computer Science, 1995.

[18] ——, *Parallel multilevel k-way partitioning scheme for irregular graphs*, tech. rep., University of Minnesota, Department of Computer Science, 1996.

[19] A. B. MACCABE, K. S. MCCURLEY, R. RIESEN, AND S. R. WHEAT, *SUNMOS for the Intel Paragon: A brief user's guide*, in Proceedings of the Intel Supercomputer Users' Group. 1994 Annual North America Users' Conference, June 1994, pp. 245–251.

[20] A. B. MACCABE AND S. R. WHEAT, *Message passing in puma*, Tech. Rep. SAND93-0935, Sandia National Laboratories, October 1993.

[21] K. S. MCCURLEY, *Intel nx compatibility under sunmos*, Tech. Rep. SAND93-2618, Sandia National Laboratories, June 1995.

[22] A. POTHEN, H. SIMON, AND K. LIOU, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

[23] J. SHADID, S. HUTCHINSON, G. HENNIGAN, H. MOFFAT, K. DEVINE, AND A. G. SALINGER, *Efficient parallel computation of unstructured finite element reacting flow solutions*, to appear in Parallel Computing, (1997).

[24] J. N. Shadid, H. K. Moffat, S. A. Hutchinson, G. L. Hennigan, K. D. Devine, and A. G. Salinger, *Mpsalsa: A finite element computer program for reacting flow problems; part 1 - theoretical development*, Tech. Rep. SAND95-2752, Sandia National Laboratories, May 1996.

[25] L. Shuler, R. Riesen, C. Jong, D. van Dresser, A. B. Maccabe, L. A. Fisk, and T. M. Stallcup, *The Puma operating system for massively parallel computers*, in Proceedings of the Intel Supercomputer Users' Group. 1995 Annual North America Users' Conference, June 1995.

[26] M. B. Stephenson, S. A. Canann, and T. D. Blacker, *Plastering: a new approach to automated, 3-d hexahedral mesh generation*, Tech. Rep. SAND92-2192, Sandia National Laboratories, 1992.

[27] R. M. Summers, J. S. Peery, K. W. Wong, E. S. Hertel, T. G. Trucano, and L. C. Chhabildas, *Recent progress in alegra development and application to ballistic impacts*, to be published International Journal of Impact Engineering, (1997).

[28] S. R. Wheat, A. B. Maccabe, R. Riesen, D. W. van Dresser, and T. M. Stallcup, *PUMA: An operating system for massively parallel systems*, in Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, IEEE Computer Society Press, 1994, pp. 56–65.

URL References

[URL1] TERAFLOP/ASCI RED documentation,
`http://www.acl.lanl.gov/~pfay/teraflop/`

[URL2] ALEGRA - A Three-Dimensional, Multi-Material, Arbitrary-Lagrangian-Eulerian Code for Solid Dynamics, `http://www.sandia.gov/1431/ALEGRAw.html`

[URL3] CTH 3D Eulerian Shock Code, `http://www.sandia.gov/1431/CTHw.html`

[URL4] Massively Parallel Numerical Methods for Advanced Simulation of Chemically Reacting Flows, `http://www.cs.sandia.gov/CRF/mpsalsa.html`

[URL5] Aztec: A Massively Parallel Iterative Solver Package for Solving Linear Systems, `http://www.cs.sandia.gov/CRF/aztec1.html`

[URL6] Sandia's Data Visualization and Exploration,
`http://www.cs.sandia.gov/VIS/`

[URL7] Algorithms and Software for Partitioning Meshes,
`http://www.cs.sandia.gov/CRF/chac.html`