

Chapter 2

Petascale Computing: Impact on Future NASA Missions

Rupak Biswas

NASA Ames Research Center

Michael Aftosmis

NASA Ames Research Center

Cetin Kiris

NASA Ames Research Center

Bo-Wen Shen

NASA Goddard Space Flight Center

2.1	Introduction	29
2.2	The Columbia Supercomputer	30
2.3	Aerospace Analysis and Design	31
2.4	Propulsion Subsystem Analysis	35
2.5	Hurricane Prediction	39
2.6	Bottlenecks	42
2.7	Summary	44
	References	45

2.1 Introduction

To support its diverse mission-critical requirements, the National Aeronautics and Space Administration (NASA) solves some of the most unique, computationally challenging problems in the world [8]. To facilitate rapid yet accurate solutions for these demanding applications, the U.S. space agency procured a 10,240-CPU supercomputer in October 2004, dubbed Columbia. Housed in the NASA Advanced Supercomputing facility at NASA Ames Research Center, Columbia is comprised of twenty 512-processor nodes (representing three generations of SGI Altix technology: 3700, 3700-BX2, and 4700), with a combined peak processing capability of 63.2 teraflops per second (TFLOPS). However, for many applications, even this high-powered computational workhorse, currently ranked as one of the fastest in the world, does not

have enough computing capacity, memory size, and bandwidth rates needed to meet all of NASA's diverse and demanding future mission requirements.

In this chapter, we consider three important NASA application areas: aerospace analysis and design, propulsion subsystems analysis, and hurricane prediction, as a representative set of these challenges. We show how state-of-the-art methodologies for each application area are currently performing on Columbia, and explore projected achievements with the availability of petascale computing. We conclude by describing some of the architecture and algorithm obstacles that must first be overcome for these applications to take full advantage of such petascale computing capability.

2.2 The Columbia Supercomputer

Columbia is configured as a cluster of 20 SGI Altix nodes, each with 512 Intel Itanium 2 processors and one terabyte (TB) of global shared-access memory, that are interconnected via InfiniBand fabric communications technology. Of these 20 nodes, 12 are model 3700, seven are model 3700-BX2, and one is the newest-generation architecture, a 4700. The 3700-BX2 is a double-density incarnation of the 3700, while the 4700 is a dual-core version of the 3700-BX2. Each node acts as a shared-memory, single-system-image environment running a Linux-based operating system, and utilizes SGI's scalable, shared-memory NUMAflex architecture, which stresses modularity.

Four of Columbia's BX2 nodes are tightly linked to form a 2,048-processor 4 TB shared-memory environment and use NUMalink4 among themselves, which allows access to all data directly and efficiently, without having to move them through I/O or networking bottlenecks. Each processor in the 2,048-CPU subsystem runs at 1.6 GHz, has 9 MB of level-3 cache (the Madison 9M processor), and a peak performance of 6.4 gigaflops per second (GFLOPS). One other BX2 node is equipped with these same processors. (These five BX2 nodes are denoted as BX2b in this chapter.) The remaining fourteen nodes: two BX2 (referred to as BX2a here) and twelve 3700, all have processors running at 1.5 GHz, with 6 MB of level-3 cache, and a peak performance of 6.0 GFLOPS. All nodes have 2 GB of shared memory per processor.

The 4700 node of Columbia is the latest generation in SGI's Altix product line, and consists of 8 racks with a total of 256 dual-core Itanium 2 (Montecito) processors (1.6 GHz, 18 MB of on-chip level-3 cache) and 2 GB of memory per core (1 TB total). Each core also contains 16 KB instruction and data caches. The current configuration uses only one socket per node, leaving the other socket unused (also known as the bandwidth configuration). Detailed performance characteristics of the Columbia supercomputer using micro-benchmarks, compact kernel benchmarks, and full-scale applications can be found in other articles [7, 10].

2.3 Aerospace Analysis and Design

High-fidelity computational fluid dynamics (CFD) tools and techniques are developed and applied to many aerospace analysis and design problems throughout NASA. These include the full Space Shuttle Launch Vehicle (SSLV) configuration and future spacecraft such as the Crew Exploration Vehicle (CEV). One of the more commonly used high-performance aerodynamics simulation packages used on Columbia to assist with aerospace vehicle analysis and design is Cart3D. This software package enables high-fidelity characterization of aerospace vehicle design performance over the entire flight envelope.

Cart3D is a simulation package targeted at conceptual and preliminary design of aerospace vehicles with complex geometry. It solves the Euler equations governing inviscid flow of a compressible fluid on an automatically generated Cartesian mesh surrounding a vehicle. Since the package is inviscid, boundary layers and viscous phenomena are not present in the simulations, facilitating fully automated Cartesian mesh generation in support of inviscid analysis. Moreover, solutions to the governing equations can typically be obtained for about 2–5% of the cost of a full Reynolds-averaged Navier-Stokes (RANS) simulation. The combination of automatic mesh generation and high-quality, yet inexpensive, flow solutions makes the package ideally suited for rapid design studies and exploring what-if scenarios. Cart3D offers a drop-in replacement for less scalable and lower-fidelity engineering methods, and is frequently used to generate entire aerodynamic performance databases for new vehicles [1, 3].

2.3.1 Methodology

Cart3D's solver module uses a second-order cell-centered, finite-volume upwind spatial discretization combined with a multigrid-accelerated Runge-Kutta scheme for advance to steady-state [1]. As shown in Figure 2.1 (a), the package uses adaptively refined, hierarchically generated Cartesian cells to discretize the flow field and resolve the geometry. In the field, cells are simple Cartesian hexahedra, and the solver capitalizes on regularity of the mesh for both speed and accuracy. At the wall, these cells are cut arbitrarily and require more extensive data structures and elaborate mathematics. Nevertheless, the set of cut-cells is lower-dimensional, and the net cost remains low. Automation and insensitivity to geometric complexity are key ingredients in enabling rapid parameter sweeps over a variety of configurations.

Cart3D utilizes domain decomposition to achieve high efficiency on parallel machines [3, 6]. To assess performance of Cart3D's solver module on realistic problems, extensive experiments have been conducted on several large applications. The case considered here is that of the SSLV. A mesh containing approximately 4.7 million cells around this geometry is shown in Figure 2.1 (a) and was built using 14 levels of adaptive subdivision. The grid is painted to

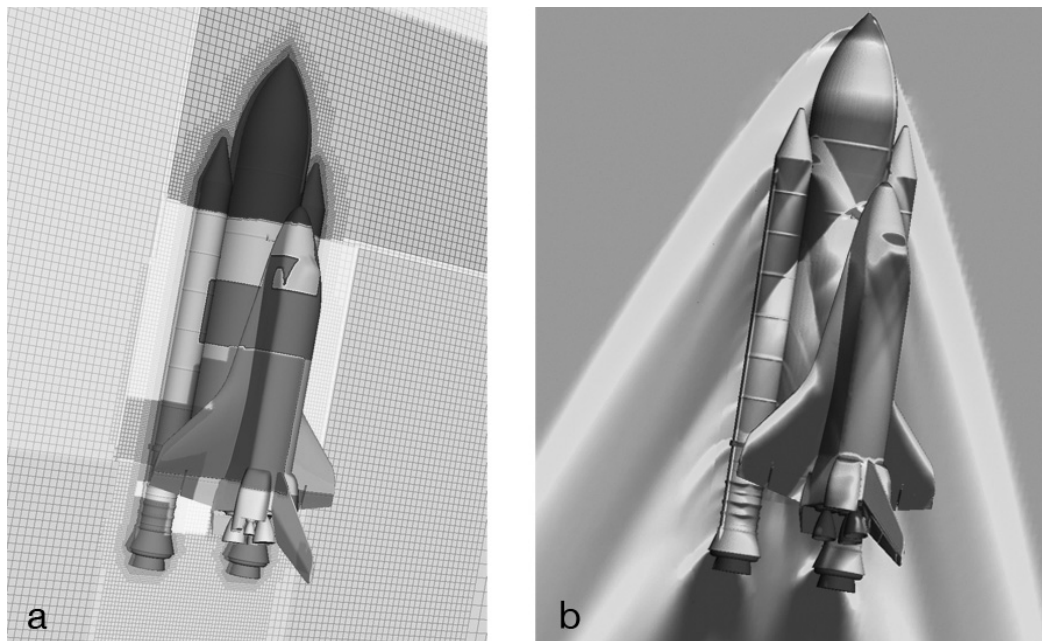


FIGURE 2.1: (See color insert following page 18.) Full SSLV configuration including orbiter, external tank, solid rocket boosters, and fore and aft attach hardware. (a) Cartesian mesh surrounding the SSLV; colors indicate 16-way decomposition using the SFC partitioner. (b) Pressure contours for the case described in the text; the isobars are displayed at 2.6 Mach, 2.09 degrees angle-of-attack, and 0.8 degrees sideslip corresponding to flight conditions approximately 80 seconds after launch.

indicate partitioning into 16 sub-domains using the Peano-Hilbert space-filling curve (SFC) [3]. The partitions are all predominantly rectangular, which is characteristic of sub-domains generated with SFC-based partitioners, indicating favorable compute/communicate ratios.

Cart3D's simulation module solves five partial differential equations for each cell in the domain, giving this example close to 25 million degrees-of-freedom. Figure 2.1 (b) illustrates a typical result from these simulations by showing pressure contours in the discrete solution. Surface triangulation describing the geometry contains about 1.7 million elements. For parallel performance experiments on Columbia, the mesh in Figure 2.1 (a) was refined so that the test case contained approximately 125 million degrees-of-freedom. These investigations included comparisons between the OpenMP and MPI parallel programming paradigms, analyzing the impact of multigrid on scalability, and understanding the effects of the NUMalink4 and InfiniBand communication fabrics.

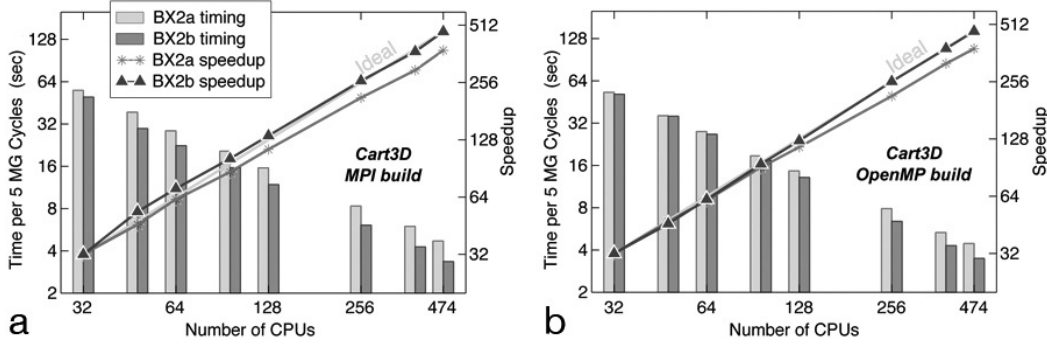


FIGURE 2.2: Comparison of execution time and parallel speedup of the Cart3D solver module on single BX2a and BX2b Columbia nodes using (a) MPI and (b) OpenMP parallelization strategies.

2.3.2 Results

The domain-decomposition parallelization strategy in the Cart3D flow simulation package has previously demonstrated excellent scalability on large numbers of processors with both MPI and OpenMP libraries [6]. This behavior makes it a suitable candidate for comparing performance of Columbia's BX2a and BX2b nodes using varying processor counts on a complete application. Figure 2.2 shows charts of parallel speedup and execution timings for Cart3D using MPI and OpenMP. Line graphs in the figure show parallel speedup between 32 and 474 CPUs; the corresponding execution time for five multigrid cycles is shown via bar charts. This comparison clearly demonstrates excellent scalability for both node types, indicating that this particular example did not exceed the bandwidth capabilities of either system. The bar charts in Figure 2.2 contain wall-clock timing data and show comparisons consistent with the slightly faster clock-speeds in the BX2b nodes.

The memory on each of Columbia's 512-CPU nodes is globally sharable by any process within the node, but cache-coherency is not maintained between nodes. Thus, all multi-node examples with Cart3D are run using the MPI communication back-end. Numerical experiments focus on the effects of increasing the number of multigrid levels in the solution algorithm. These experiments were carried out on four of Columbia's BX2b nodes. Figure 2.3 (a) displays parallel speedup for the system, comparing the baseline four-level multigrid solution algorithm with that on a single grid, using the NUMalink4 interconnect. The plot shows nearly ideal speedup for the single grid runs on the full 2,048-CPU system. The multigrid algorithm requires solution and residual transfer to coarser mesh levels, and therefore places substantially greater demands on interconnect bandwidth. In this case, a slight degradation in performance is apparent above 1,024 processors, but the algorithm still posts parallel speedups of about 1,585 on 2,016 CPUs.

Figure 2.3 (b) shows that the InfiniBand runs do not extend beyond 1,536 processors due to a limitation on the number of connections. Notice that

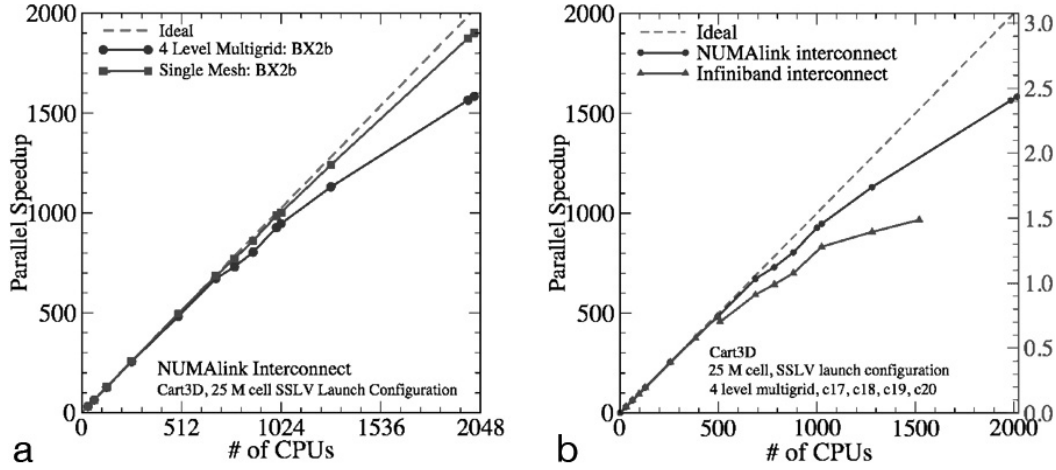


FIGURE 2.3: (a) Parallel speedup of the Cart3D solver module using one and four levels of mesh in the multigrid hierarchy with a NUMalink4 interconnect. (b) Comparison of parallel speedup and TFLOPS with four levels of multigrids using NUMalink4 and InfiniBand.

the InfiniBand performance consistently lags that of the NUMalink4, and that there is an increased penalty as the number of nodes goes up. Using the standard NCSA FLOP counting procedures, net performance of the multigrid solution algorithm is 2.5 TFLOPS on 2,016 CPUs. Additional performance details are provided in [2].

2.3.3 Benefits of petascale computing to NASA

In spite of several decades of continuous improvements in both algorithms and hardware, and despite the widespread acceptance and use of CFD as an indispensable tool in the aerospace vehicle design process, computational methods are still employed in a very limited fashion. In some important flight regimes, current computational methods for aerodynamic analysis are only reliable within a narrow range of flight conditions — where no significant flow separation occurs. This is due, in part, to the extreme accuracy requirements of the aerodynamic design problem, where, for example, changes of less than one percent in the drag coefficient of a flight vehicle can determine commercial success or failure. As a result, computational analyses are currently used in conjunction with experimental methods only over a restricted range of the flight envelope, where they have been essentially calibrated.

Improvements in accuracy achieved by CFD methods for aerodynamic applications will require, among other things, dramatic increases in grid resolution and simulation degrees-of-freedom (over what is generally considered practical in the current environment). Manipulating simulations with such high resolutions demands substantially more potent computing platforms. This drive toward finer scales and higher fidelity is obviously motivated by

a dawning understanding of error analysis in CFD simulations. Recent research in uncertainty analysis and formal error bounds indicates that hard quantification of simulation error is possible, but the analysis can be many times more expensive than the CFD simulation itself. While this field is just opening up for fluid dynamic simulations, it is already clear that certifiably accurate CFD simulations require many times the computing power currently available. Petascale computing would open the door to affordable, routine, error quantification for simulation data.

Once optimal designs can be constructed, they must be validated throughout the entire flight envelope, which includes hundreds of thousands of simulations of both the full range of aerodynamic flight conditions, and parameter spaces of all possible control surface deflections and power settings. Generating this comprehensive database will not only provide all details of vehicle performance, but it will also open the door to new possibilities for engineers. For example, when coupled with a six-degree-of-freedom integrator, the vehicle can be flown through the database by guidance and control (G&C) system designers to explore issues of stability and control. Digital Flight initiatives undertake the complete time-accurate simulation of a maneuvering vehicle and include structural deformation and G&C feedback. Ultimately, the vehicle's suitability for various mission profiles or other trajectories can be evaluated by full end-to-end mission simulations, and optimization studies can consider the full mission profile. This is another area where a petascale computing capability can significantly benefit NASA missions.

2.4 Propulsion Subsystem Analysis

High-fidelity unsteady flow simulation techniques for design and analysis of propulsion systems play a key role in supporting NASA missions, including analysis of the liquid rocket engine flowliner for the Space Shuttle Main Engine (SSME). The INS3D software package [11] is one such code developed to handle computations for unsteady flow through a full-scale, low- and high-pressure rocket pump. Liquid rocket turbopumps operate under severe conditions and at very high rotational speeds. The low-pressure fuel turbopump creates transient flow features such as reverse flows, tip clearance effects, secondary flows, vortex shedding, junction flows, and cavitation effects. The reverse flow originating at the tip of an inducer blade travels upstream and interacts with the bellows cavity. This flow unsteadiness is considered to be one of the major contributors to high-frequency cyclic loading that results in cycle fatigue.

2.4.1 Methodology

To resolve the complex geometry in relative motion, an overset grid approach [9] is employed where the problem domain is decomposed into a number of simple grid components. Connectivity between neighboring grids is established by interpolation at the grid outer boundaries. Addition of new components to the system and simulation of arbitrary relative motion between multiple bodies are achieved by establishing new connectivity without disturbing existing grids.

The computational grid used for the experiments reported in this chapter consisted of 66 million grid points and 267 blocks or zones. Details of the grid system are shown in Figure 2.4. The INS3D code solves the incompressible Navier-Stokes equations for both steady-state and unsteady flows. The numerical solution requires special attention to satisfy the divergence-free constraint on the velocity field. The incompressible formulation does not explicitly yield the pressure field from an equation of state or the continuity equation. One way to avoid the difficulty of the elliptic nature of the equations is to use an artificial compressibility method that introduces a time-derivative of the pressure term into the continuity equation. This transforms the elliptic-parabolic partial differential equations into the hyperbolic-parabolic type. To obtain time-accurate solutions, the equations are iterated to convergence in pseudo-time for each physical time step until divergence of the velocity field has been reduced below a specified tolerance value. The total number of required sub-iterations varies depending on the problem, time step size, and artificial compressibility parameter. Typically, the number ranges from 10–30 sub-iterations. The matrix equation is solved iteratively by using a non-factored Gauss-Seidel-type line-relaxation scheme, which maintains stability and allows a large pseudo-time step to be taken. More detailed information about the application can be found elsewhere [11, 12].

2.4.2 Results

Computations were performed to compare scalability between the multi-level parallelism (MLP) [17] and MPI+OpenMP hybrid (using a point-to-point communication protocol) versions of INS3D on one of Columbia’s BX2b nodes. Both implementations combine coarse- and fine-grain parallelism. Coarse-grain parallelism is achieved through a UNIX fork in MLP, and through explicit message-passing in MPI+OpenMP. Fine-grain parallelism is obtained using OpenMP compiler directives in both versions. The MLP code utilizes a global shared-memory data structure for overset connectivity arrays, while the MPI+OpenMP code uses local copies.

Initial computations using one group and one thread were used to establish the baseline runtime for one physical time step, where 720 such time steps are required to complete one inducer rotation. Figure 2.5 displays the time per iteration (in minutes) versus the number of CPUs, and the speedup factor for

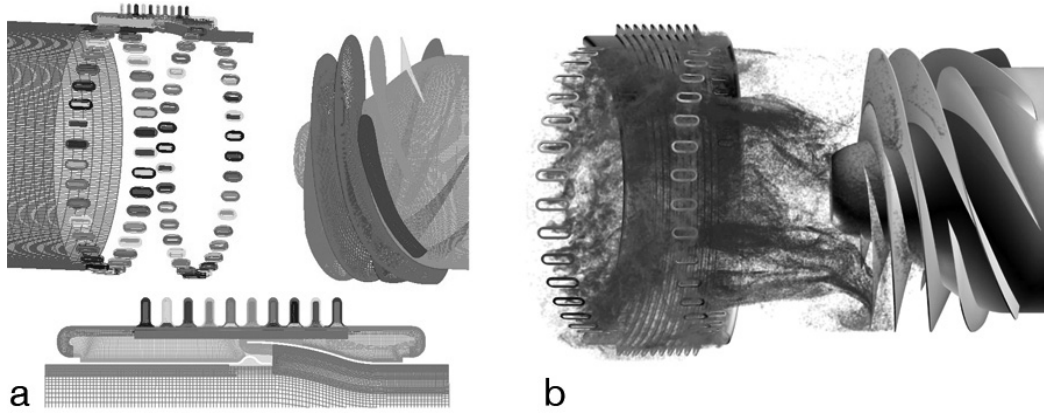


FIGURE 2.4: (See color insert following page 18.) Liquid rocket turbopump for the SSME. (a) Surface grids for the low-pressure fuel pump inducer and flowliner. (b) Instantaneous snapshot of particle traces colored by axial velocity values.

both codes. Here, 36 groups have been chosen to maintain good load balance for both versions. Then, the runtime per physical time step is obtained using various numbers of OpenMP threads (1, 2, 4, 8, and 14). It includes the I/O time required to write the time-accurate solution to disk at each time step. The scalability for a fixed number of MLP and MPI groups and varying OpenMP threads is good, but begins to decay as the number of OpenMP threads becomes large. Further scaling can be accomplished by fixing the number of OpenMP threads and increasing the number of MLP/MPI groups until the load balancing begins to fail. Unlike varying the OpenMP threads, which does not affect the convergence rate of INS3D, varying the number of groups may deteriorate it. This will lead to more iterations even though faster runtime per iteration is achieved. The results show that the MLP and MPI+OpenMP codes perform almost equivalently for one OpenMP thread, but that the latter begins to perform slightly better as the number of threads is increased. This advantage can be attributed to having local copies of the connectivity arrays in the MPI+OpenMP hybrid implementation. Having the MPI+OpenMP version of INS3D as scalable as the MLP code is promising since this implementation is easily portable to other platforms.

We also compare performance of the INS3D MPI+OpenMP code on multiple BX2b nodes against single node results. This includes running the MPI+OpenMP version using two different communication paradigms: master-worker and point-to-point. The runtime per physical time step is recorded using 36 MPI groups, and 1, 4, 8, and 14 OpenMP threads on one, two, and four BX2b nodes. Communication between nodes is achieved using the InfiniBand and NUMalink4 interconnects, denoted as IB and XPM, respectively.

Figure 2.6 (a) contains results using the MPI point-to-point communication paradigm. When comparing performance of using multiple nodes with that of

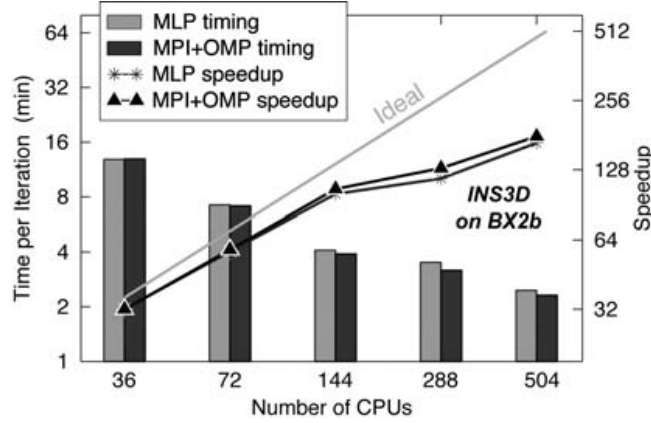


FIGURE 2.5: Comparison of INS3D performance on a BX2b node using two different hybrid programming paradigms.

a single node, we observe that scalability of the multi-node runs with NUMA-link4 is similar to the single-node runs (which also use NUMAlink4 internally). However, when using InfiniBand, execution time per iteration increases by 10–29% on two- and four-node runs. The difference between the two- and four-node runs decreases as the number of CPUs increases. Figure 2.6 (b) displays the results using the master-worker communication paradigm. Note that the time per iteration is much higher using this protocol compared to the point-to-point communication. We also see a significant deterioration in scalability for both single- and multi-node runs. With NUMAlink4, we observe a 5–10% increase in runtime per iteration from one to two nodes, and an 8–16% increase using four nodes. This is because the master resides on one node, and all workers on the other nodes must communicate with the master. However, when using point-to-point communication, many of the messages remain within the node from which they are sent. An additional 14–27% increase in runtime is observed when using InfiniBand instead of NUMAlink4, independent of the communication paradigm.

2.4.3 Benefits of petascale computing to NASA

The benefits of high-fidelity modeling of full-scale multi-component, multi-physics propulsion systems to NASA’s current mission goals are numerous and have the most significant impact in the areas of: crew safety (new safety protocols for the propulsion system); design efficiency (provide the ability to make design changes that can improve the efficiency and reduce the cost of space flight); and technology advancement (in propulsion technology for manned space flights to Mars).

With petascale computing, fidelity of the current propulsion subsystem analysis could be increased to full-scale, multi-component, multi-disciplinary

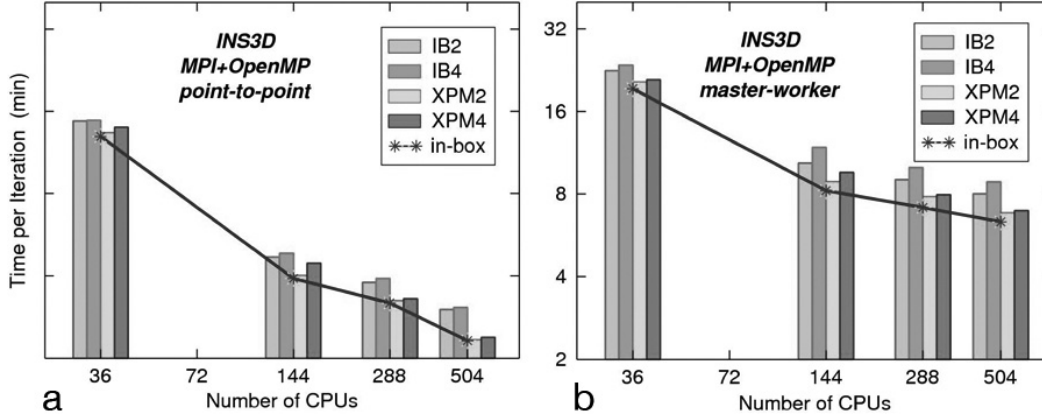


FIGURE 2.6: Performance of INS3D across multiple BX2b nodes via NUMalink4 and InfiniBand using MPI (a) point-to-point, and (b) master-worker communication.

propulsion applications. Multi-disciplinary computations are critical for modeling propulsion systems of new and existing launch vehicles to attain flight rationale. To ensure proper coupling between the fluid, structure, and dynamics codes, the number of computed iterations will dramatically increase, demanding large, parallel computing resources for efficient solution turnaround time. Spacecraft propulsion systems contain multi-component/multi-phase fluids (such as turbulent combustion in solid rocket boosters and cavitating hydrodynamic pumps in the SSME) where phase change cannot be neglected when trying to obtain accurate and reliable results.

2.5 Hurricane Prediction

Accurate hurricane track and intensity predictions help provide early warning to people in the path of a storm, saving both life and property. Over the past several decades, hurricane track forecasts have steadily improved, but progress on intensity forecasts and understanding of hurricane formation/genesis has been slow. Major limiting factors include insufficient model resolutions and uncertainties of cumulus parameterizations (CP). A CP is required to emulate the statistical effects of unresolved cloud motions in coarse resolution simulations, but its validity becomes questionable at high resolutions. Facilitated by Columbia, the ultra-high resolution finite-volume General Circulation Model (fvGCM) [4] has been deployed and run in real-time to study the impacts of increasing resolutions and disabling CPs on hurricane forecasts. The fvGCM code is a unified numerical weather prediction (NWP) and climate model that runs on daily, monthly, decadal, and century timescales,

TABLE 2.1: Changes in fvGCM resolution as a function of time (available computing resources). Note that in the vertical direction, the model could be running with 32, 48, 55, or 64 stretched levels.

Resolution (lat×long)	Grid Points ($y \times x$)	Total 2D Grid Cells	Major Application	Implementation Date
$2^\circ \times 2.5^\circ$	91×144	13,104	Climate	1990s
$1^\circ \times 1.25^\circ$	181×288	52,128	Climate	Jan. 2000
$0.5^\circ \times 0.625^\circ$	361×576	207,936	Climate/Weather	Feb. 2002
$0.25^\circ \times 0.36^\circ$	721×1000	721,000	Weather	July 2004
$0.125^\circ \times 0.125^\circ$	1441×2880	4,150,080	Weather	Mar. 2005
$0.08^\circ \times 0.08^\circ$	2251×4500	11,479,500	Weather	July 2005

and is currently the only operational global NWP model with finite-volume dynamics. While doubling the resolution of such a model requires an 8-16X increase in computational resources, the unprecedented computing capability provided by Columbia enables us to rapidly increase resolutions of fvGCM to 0.25° , 0.125° , and 0.08° , as shown in Table 2.1.

While NASA launches many high-resolution satellites, the mesoscale-resolving fvGCM is one of only a few global models with comparable resolution to satellite data (QuikSCAT, for example), providing a mechanism for direct comparisons between model results and satellite observations. During the active 2004 and 2005 hurricane seasons, the high-resolution fvGCM produced promising forecasts of intense hurricanes such as Frances, Ivan, Jeanne, and Karl in 2004; and Emily, Dennis, Katrina, and Rita in 2005 [4, 14, 15, 16]. To illustrate the capabilities of fvGCM, coupled with the computational power of Columbia, we discuss the numerical forecasts of Hurricanes Katrina and Rita in this chapter.

2.5.1 Methodology

The fvGCM code, resulting from a development effort of more than ten years, has the following three major components: (1) finite-volume dynamics, (2) NCAR Community Climate Model (CCM3) physics, and (3) NCAR Community Land Model (CLM). Dynamical initial conditions and sea surface temperature (SST) were obtained from the global forecast system (GFS) analysis data and one-degree optimum interpolation SST of National Centers for Environmental Prediction.

The unique features of the finite-volume dynamical core [13] include a genuinely conservative flux-form semi-Lagrangian transport algorithm which is Gibbs oscillation-free with the optional monotonicity constraint; a terrain-following Lagrangian control-volume vertical coordinate system; a finite-volume integration method for computing pressure gradients in general terrain following coordinates; and a mass, momentum, and total energy conserving algorithm for remapping the state variables from the Lagrangian control-volume

to an Eulerian terrain-following coordinate. The vorticity-preserving horizontal dynamics enhance the simulation of atmospheric oscillations and vortices. Physical processes such as CP and gravity-wave drag are largely enhanced with emphasis for high-resolution simulations; they are also modified for consistent application with the innovative finite-volume dynamics.

From a computational perspective, a crucial aspect of the fvGCM development is its high computational efficiency on a variety of high-performance supercomputers including distributed-memory, shared-memory, and hybrid architectures. The parallel implementation is hybrid: coarse-grain parallelism with MPI/MLP/SHMEM and fine-grain parallelism with OpenMP. The model's dynamical part has a 1-D MPI/MLP/SHMEM parallelism in the y -direction, and uses OpenMP multi-threading in the z -direction. One of the prominent features in the implementation is the permission of multi-threaded data communications. The physical part inherits 1-D parallelism in the y -direction from the dynamical part, and further applies OpenMP loop-level parallelism in this decomposed latitude. CLM is also implemented with MPI and OpenMP parallelism, and its grid cells are distributed among processors.

All of the aforementioned features make it possible to advance the state-of-the-art of hurricane prediction to a new frontier. To date, Hurricanes Katrina and Rita are the sixth and fourth most intense hurricanes in the Atlantic, respectively. They devastated New Orleans, southwestern Louisiana, and the surrounding Gulf Coast region, resulting in losses in excess of 90 billion U.S. dollars. Here, we limit our discussion to the simulations initialized from 1200 UTC August 25, and 0000 UTC September 21, for Katrina and Rita, and show improvement of the track and intensity forecasts by increasing resolutions to 0.125° and 0.08° , and by disabling CPs.

2.5.2 Results

The impacts of increased computing power and thus enhanced resolution (in this case, to 0.125°), and disabling CPs on the forecasts of Hurricane Katrina have been documented [15]. They simulated comparable track predictions at different resolutions, but better intensity forecasts at finer resolutions. The predicted mean sea level pressures (MSLPs) in the 0.25° , 0.125° , and 0.125° (with no CPs) runs are 951.8, 895.7, and 906.5 hectopascals (hPa) with respect to the observed 902 hPa. Consistent improvement as a result of using a higher resolution was illustrated from the six 5-day forecasts with the 0.125° fvGCM, showing small errors in center pressure of only ± 12 hPa. The notable improvement in Katrina's intensity forecasts was attributed to the sufficient fine resolution used for resolving hurricane near-eye structures. As the hurricane's internal structure has convective-scale variations, it was shown that the 0.125° run with disabled CPs could lead to further improvement on Katrina's intensity and structure (asymmetry).

Earlier forecasts of Hurricane Rita by the National Hurricane Center (represented by the line in Figure 2.7 (a) with the square symbols) had a bias

toward the left side of the best track (the line with circles, also in Figure 2.7 (a)), predicting the storm hitting Houston, Texas. The real-time 0.25° forecast (represented by the line in Figure 2.7 (a) with the diamonds) initialized at 0000 UTC 21 September showed a similar bias. Encouraged by the successful Katrina forecasts, we conducted two experiments at 0.125° and 0.08° resolutions with disabled CPs, and compared the results with the 0.25° model. From Figures 2.7 (b-d), it is clear that a higher resolution run produces a better track forecast, predicting a larger shift in landfall toward the Texas-Louisiana border. Just before making landfall, Rita was still a Category 3 hurricane with an MSLP of 931 hPa at 0000 UTC 24. Looking at Figures 2.7 (b-d) that show the predicted minimal MSLPs (957.8, 945.5, and 936.5 hPa at 0.25° , 0.125° , and 0.08° resolutions, respectively), we can conclude that a higher-resolution run produces a more realistic intensity. Although these results are promising, note that the early rapid intensification of Rita was not fully simulated in either of the above simulations, indicating the importance of further model improvement and better understanding of hurricane (internal) dynamics.

2.5.3 Benefits of petascale computing to NASA

With the availability of petascale computing resources, we could extend our approach from short-range weather/hurricane forecasts to reliable longer-duration seasonal and annual predictions. This would enable us to study hurricane climatology [5] in present-day or global-warming climate, and improve our understanding of interannual variations of hurricanes. Petascale computing could also make the development of a multi-scale, multi-component Earth system model feasible, including a non-hydrostatic cloud-resolving atmospheric model, an eddy-resolving ocean model, and an ultra-high-resolution land model [18]. Furthermore, this model system could be coupled with chemical and biological components. In addition to the model's improvement, an advanced high-resolution data assimilation system is desired to represent initial hurricanes, thereby further improving predictive skill.

2.6 Bottlenecks

Taking full advantage of petascale computing to achieve the research challenges outlined in Sections 2.3–2.5 will first require clearing a number of computational hurdles. Known bottlenecks include parallel scalability issues, development of better numerical algorithms, and the need for dramatically higher bandwidths.

Recent studies on the Columbia supercomputer underline major parallel

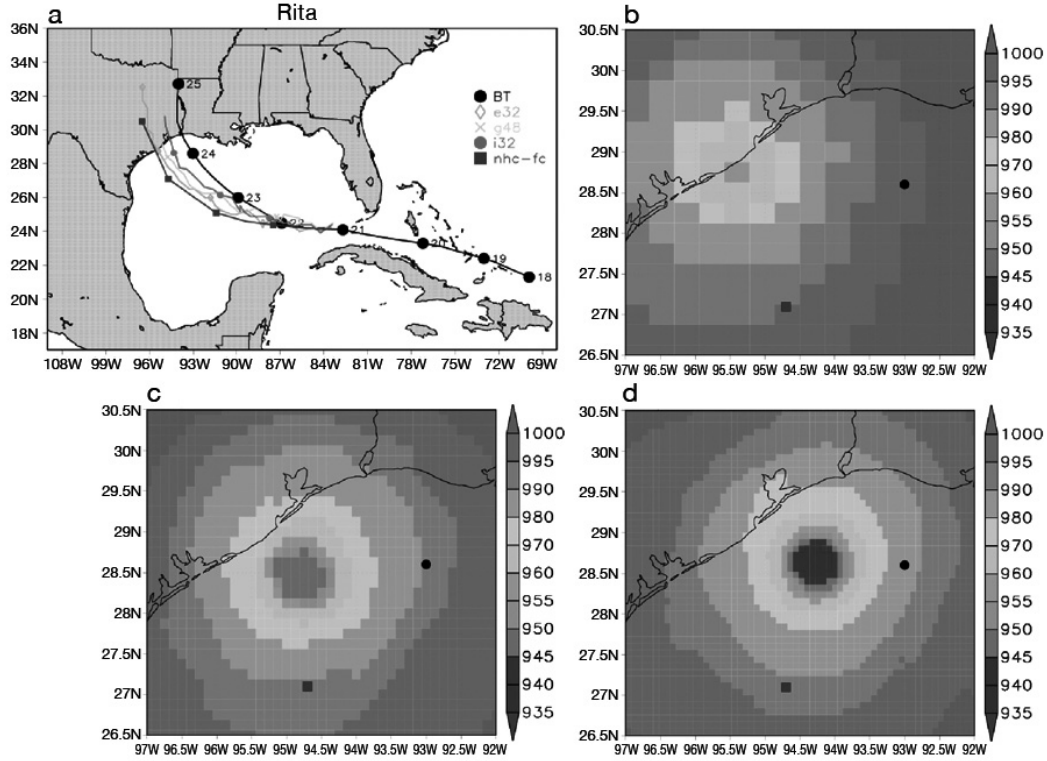


FIGURE 2.7: (See color insert following page 18.) Four-day forecasts of Hurricane Rita initialized at 0000 UTC September 21, 2005. (a) Tracks predicted by fvGCM at 0.25° (line with diamond symbols), 0.125° (line with crosses), and 0.08° (line with circles) resolutions. The lines with hexagons and squares represent the observation and official prediction by the National Hurricane Center (NHC). (b-d) Sea-level pressure (SLP) in hPa within a $4^\circ \times 5^\circ$ box after 72-hour simulations ending at 0000 UTC 24 September at 0.25° , 0.125° , and 0.08° resolutions. Solid circles and squares indicate locations of the observed and official predicted hurricane centers by the NHC, respectively. The observed minimal SLP at the corresponding time is 931 hPa. In a climate model with a typical $2^\circ \times 2.5^\circ$ resolution (latitude \times longitude), a $4^\circ \times 5^\circ$ box has only four grid-points.

scalability issues with several of NASA's current mainline Reynolds-averaged Navier-Stokes solvers when scaling to just a few hundred processors, requiring communication among multiple nodes). While sorting algorithms can be used to minimize internode communication, scaling to tens or hundreds of thousands of processors will require heavy investment in scalable solution techniques to replace NASA's current block tri- and penta-diagonal solvers.

More efficient numerical algorithms are being developed (to handle the increased number of physical time steps) which focus on scalability while increasing accuracy and preserving robustness and convergence. This means computing systems with hundreds of thousands of parallel processors (or cores) are not only desirable, but are required to solve these problems when including all of the relevant physics. In addition, unlike some Earth and space science simulations, current high-fidelity CFD codes are processor speed-bound. Runs utilizing many hundreds of processors rarely use more than a very small fraction of the available memory, and yet still take hours or days to run. As a result, algorithms which trade-off this surplus memory for greater speed are clearly of interest.

Bandwidth to memory is the biggest problem facing CFD solvers today. While we can compensate for latency with more buffer memory, bandwidth comes into play whenever a calculation must be synchronized over large numbers of processors. Systems 10x larger than today's supercomputers will require at least 20x more bandwidth, since current results show insufficient bandwidth for CFD applications on even the best available hardware.

2.7 Summary

High performance computing has always played a major role in meeting the modeling and simulation needs of various NASA missions. With NASA's 63.2 TFLOPS Columbia supercomputer, high-end computing is having an even greater impact within the agency and beyond. Significant cutting-edge science and engineering simulations in the areas of space exploration, shuttle operations, Earth sciences, and aeronautics research, are continuously occurring on Columbia, demonstrating its ability to accelerate NASA's exploration vision. In this chapter, we discussed its role in the areas of aerospace analysis and design, propulsion subsystems analysis, and hurricane prediction, as a representative set of these challenges.

But for many NASA applications, even this current capability is insufficient to meet all of the diverse and demanding future requirements in terms of computing capacity, memory size, and bandwidth rates. A petaflops-scale computing power would greatly alter the types of applications solved and approaches taken as compared with those in use today. We outlined potential

benefits of petascale computing to NASA, and described some of the architecture and algorithm bottlenecks that must be overcome to achieve its full potential.

References

- [1] M.J. Aftosmis, M.J. Berger, and G.D. Adomavicius. A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries. In *Proc. 38th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, Jan. 2000. AIAA-00-0808.
- [2] M.J. Aftosmis, M.J. Berger, R. Biswas, M.J. Djomehri, R. Hood, H. Jin, and C. Kiris. A detailed performance characterization of Columbia using aeronautics benchmarks and applications. In *Proc. 44th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, Jan. 2006. AIAA-06-0084.
- [3] M.J. Aftosmis, M.J. Berger, and S.M. Murman. Applications of space-filling-curves to Cartesian methods in CFD. In *Proc. 42nd AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, Jan. 2004. AIAA-04-1232.
- [4] R. Atlas, O. Reale, B.-W. Shen, S.-J. Lin, J.-D. Chern, W. Putman, T. Lee, K.-S. Yeh, M. Bosilovich, and J. Radakovich. Hurricane forecasting with the high-resolution NASA finite-volume General Circulation Model. *Geophysical Research Letters*, 32:L03801, doi:10.1029/2004GL021513, 2005.
- [5] L. Bengtsson, K.I. Hodges, and M. Esch. Hurricane type vortices in a high-resolution global model: Comparison with observations and re-analyses. *Tellus*, submitted.
- [6] M.J. Berger, M.J. Aftosmis, D.D. Marshall, and S.M. Murman. Performance of a new CFD flow solver using a hybrid programming paradigm. *Journal of Parallel and Distributed Computing*, 65(4):414–423, 2005.
- [7] R. Biswas, M.J. Djomehri, R. Hood, H. Jin, C. Kiris, and S. Saini. An application-based performance characterization of the Columbia super-cluster. In *Proc. SC'05*, Seattle, WA, Nov. 2005.
- [8] R. Biswas, E.L. Tu, and W.R. Van Dalsem. Role of high-end computing in meeting NASA's science and engineering challenges. In H. Deconinck and E. Dick, editors, *Computational Fluid Dynamics 2006*. Springer, to appear.

- [9] P.G. Buning, D.C. Jespersen, T.H. Pulliam, W.M. Chan, J.P. Slotnick, S.E. Krist, and K.J. Renze. Overflow User's Manual, Version 1.8g. Technical report, NASA Langley Research Center, Hampton, VA, 1999.
- [10] R. Hood, R. Biswas, J. Chang, M.J. Djomehri, and H. Jin. Benchmarking the Columbia supercluster. *International Journal of High Performance Computing Applications*, to appear.
- [11] C. Kiris, D. Kwak, and W.M. Chan. Parallel unsteady turbopump simulations for liquid rocket engines. In *Proc. SC2000*, Dallas, TX, Nov. 2000.
- [12] C. Kiris, D. Kwak, and S. Rogers. Incompressible Navier-Stokes solvers in primitive variables and their applications to steady and unsteady flow simulations. In M. Hafez, editor, *Numerical Simulations of Incompressible Flows*, pages 3–24. World Scientific, 2003.
- [13] S.-J. Lin. A vertically Lagrangian finite-volume dynamical core for global models. *Monthly Weather Review*, 132:2293–2307, 2004.
- [14] B.-W. Shen, R. Atlas, J.-D. Chern, O. Reale, S.-J. Lin, T. Lee, and J. Chang. The 0.125-degree finite volume General Circulation Model on the NASA Columbia supercomputer: Preliminary simulations of mesoscale vortices. *Geophysical Research Letters*, 33:L05801, doi:10.1029/2005GL024594, 2006.
- [15] B.-W. Shen, R. Atlas, O. Reale, S.-J. Lin, J.-D. Chern, J. Chang, C. Henze, and J.-L. Li. Hurricane forecasts with a global mesoscale-resolving model: Preliminary results with Hurricane Katrina (2005). *Geophysical Research Letters*, 33:L13813, doi:10.1029/2006GL026143, 2006.
- [16] B.-W. Shen, W.-K. Tao, R. Atlas, T. Lee, O. Reale, J.-D. Chern, S.-J. Lin, J. Chang, C. Henze, and J.-L. Li. Hurricane forecasts with a global mesoscale-resolving model on the NASA Columbia supercomputer. In *Proc. AGU 2006 Fall Meeting*, San Francisco, CA, Dec. 2006.
- [17] J.R. Taft. Achieving 60 Gflops/s on the production CFD code OVERFLOW-MLP. *Parallel Computing*, 27(4):521–536, 2001.
- [18] W.M. Washington. The computational future for climate change research. *Journal of Physics*, pages 317–324, doi:10.1088/1742-6596/16/044, 2005.

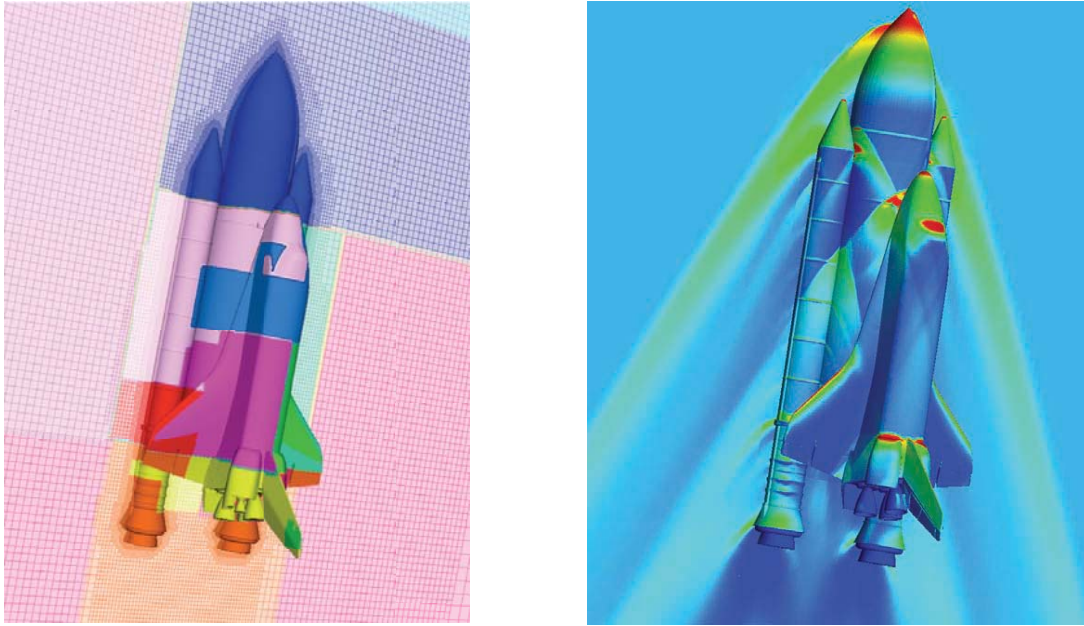


FIGURE 2.1 Full SSLV configuration including orbiter, external tank, solid rocket boosters, and fore and aft attach hardware. (a) Cartesian mesh surrounding the SSLV; colors indicate 16-way decomposition using the SFC partitioner. (b) Pressure contours for the case described in the text; the isobars are displayed at 2.6 Mach, 2.09 degrees angle-of-attack, and 0.8 degrees sideslip corresponding to flight conditions approximately 80 seconds after launch.

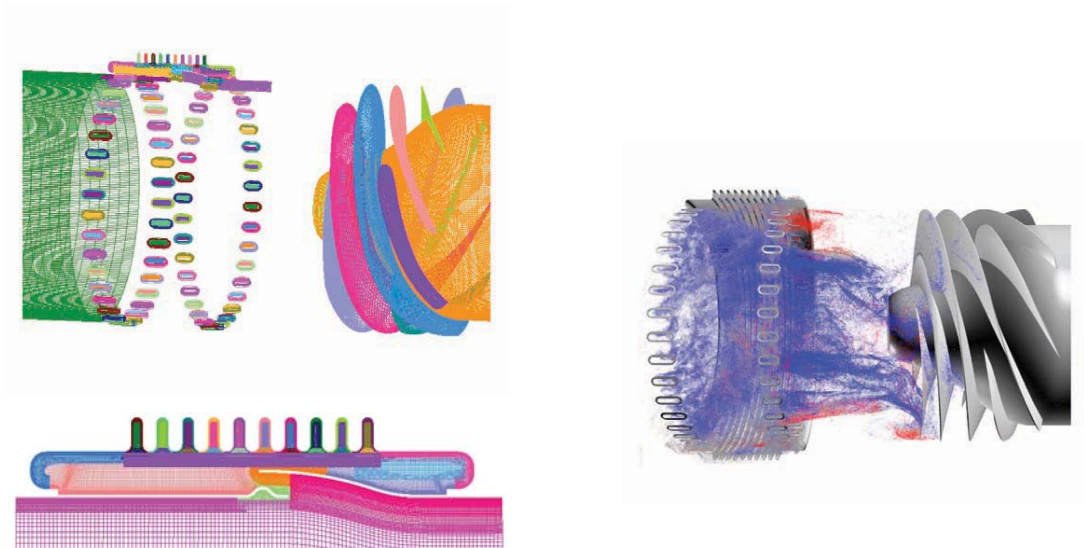


FIGURE 2.4 Liquid rocket turbopump for the SSME. (a) Surface grids for the low-pressure fuel pump inducer and flowliner. (b) Instantaneous snapshot of particle traces colored by axial velocity values.

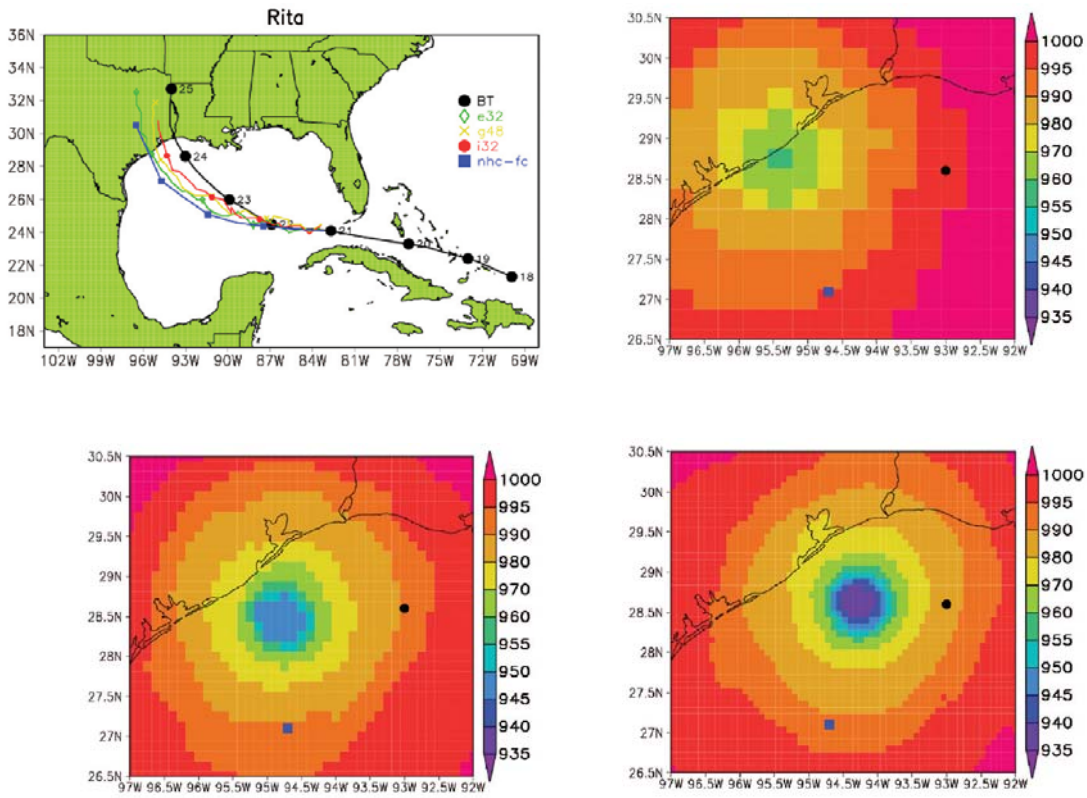


FIGURE 2.7 Four-day forecasts of Hurricane Rita initialized at 0000 UTC September 21, 2005. (a) Tracks predicted by fvGCM at 0.25° (line with diamond symbols), 0.125° (line with crosses), and 0.08° (line with circles) resolutions. The lines with hexagons and squares represent the observation and official prediction by the National Hurricane Center (NHC). (b-d) Sea level pressure (SLP) in hPa within a 4° × 5° box after 72-hour simulations ending at 0000 UTC 24 September at 0.25°, 0.125°, and 0.08° resolutions. Solid circles and squares indicate locations of the observed and official predicted hurricane centers by the NHC, respectively. The observed minimal SLP at the corresponding time is 931 hPa. In a climate model with a typical 2° × 2.5° resolution (latitude × longitude), a 4° × 5° box has only four grid points.

PETASCALE COMPUTING

ALGORITHMS AND APPLICATIONS

"This is an exciting period for HPC and a period which promises unprecedented discoveries 'at scale', which can provide tangible benefits for both science and society. This book provides a glimpse into the challenging work of petascale's first wave of application and algorithm pioneers, and as such, provides an important context for both the present and the future."

—Francine Berman, Director, San Diego Supercomputer Center, La Jolla, California, USA

"This book provides a quick introduction on how the next generation of supercomputers will be used and a look into the future of large-scale scientific computing. The authors present many of the issues and challenges that face computational scientists in the effective use of the fastest computers."

—Jack Dongarra, University of Tennessee, Knoxville, USA

"The collection of articles in this book provides an excellent introduction to the state of the art in high-performance computing. Written by some of the best practitioners in the field and focused on real applications, it clearly illustrates the complex interplay between application characteristics, programming languages and libraries, and machine characteristics. Any person involved in the development of high-performance computing software will benefit from reading this timely book."

—Marc Snir, University of Illinois, Urbana-Champaign, USA

"A milestone book on petascale computing."

—Guojie Li, Chinese Academy of Sciences, Beijing

"There is a need for this book. Petascale systems are arriving in 2008 or so, and there will be a strong demand to demonstrate that these systems are useful. David Bader has collected an impressive list of topics and contributors. The content will be very relevant to the pursuit of effective petascale system use."

—Michael A. Heroux, Sandia National Laboratories, Albuquerque, New Mexico, USA

"A timely textbook for Japan's next-generation supercomputer users in nanophysics, bioscience, and technology."

—Yoshio Oyanagi, Kogakuin University, Japan



Chapman & Hall/CRC

Taylor & Francis Group

an informa business

www.taylorandfrancisgroup.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487

270 Madison Avenue
New York, NY 10016

2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK



www.crcpress.com