Grid Solutions for Biological and Physical Cross-Site Simulations on the TeraGrid

S. Dong[1], N.T. Karonis[2,3] and G.E. Karniadakis[1]

[1] Division of Applied Mathematics
Brown University
Providence, RI 02863
{sdong,gk}@dam.brown.edu

[2] Department of Computer Science
Northern Illinois University
DeKalb, IL 60115
karonis@niu.edu

[3] Mathematics and Computer Science Division,
Argonne National Laboratory
Argonne, IL 60439

Abstract

Computational grids and grid middleware offer unprecedented computational power and storage capacity, and thus, have opened the possibility of solving problems that were previously not possible on even the largest single computational resources. These opportunities notwithstanding, the development of grid applications that run efficiently remains a challenge due to the inherent heterogeneity of networks and system architectures inherent in such environments. We present grid solutions to two grand challenge problems in computational mechanics. To study the scalability of our solutions we implemented both as MPI applications and ran them on the TeraGrid using NEKTAR and MPICH-G2. We present the results of our study which demonstrate near linear scalability in both applications when run across multiple TeraGrid sites and at a scale of hundreds or processors.

1. Grid Computing

The National Science Foundation's TeraGrid (TG) (http://www.teragrid.org) integrates the most powerful open resources in the US, which at present amount to about 50 teraflops in processing power and 1.5 petabytes of online storage connected with 40 Gb/s network. Unlike conventional supercomputers, it offers the opportunity for potentially unlimited scalability. The key question that computational scientists are faced with, however, is how to adapt their application to such complex and heterogeneous network effectively. We are, indeed, at a crossroads in parallel scientific computing, similar to what computational scientists went through about fifteen years ago. The emergence of parallel software, (e.g., MPI and OpenMP), and also of domain decomposition algorithms and corresponding freeware, (e.g., METIS) [14], made parallel computing available to the wider scientific community and allowed first-principles simulations of turbulence at very fine scales, of blood flow in the human heart [15], and of global climate at just a few km-level resolution.

On the other hand, simulations designed to capture detailed physicochemical, mechanical or biological processes have demonstrated quite different characteristics [2, 4, 5, 17, 18]. Some applications are computation intensive, requiring extremely powerful computing systems. Others are data intensive [1, 3, 16], necessitating creation or mining multi-terabyte data archives to extract scientific insight. Large-scale biological and physical simulations are extremely computation intensive, and are usually characterized by tightly-coupled computations and communications. To efficiently and effectively harness the power of grid computing, it is necessary to design and adapt applications to exploit ensembles of supercomputers and match application requirements and characteristics with grid resources.

The challenges in the development of such grid-enabled applications lie primarily in the high degree of system heterogeneity and dynamic behavior in architecture and performance of the Grid environment. For example, a grid may have a highly heterogeneous and unbalanced communication network, whose bandwidth and latency characteristics may vary widely over time and space. The computers in grid environments may also have radically different operating systems and utilities.

The Grid technology [9] represented predominantly by Globus-family services has largely overcome the difficulties in the management of such heterogeneous environment. With the uniform mechanisms for user authentication, accounting, resource access and data transfer provided by these services, it becomes possible for users and applications to discover and use disparate resources in coordinated ways. In particular, the emergence of scientific application-oriented grid middleware such as MPICH-G2 [12] has significantly relieved computational scientists from low-level details of communication handling, network topology, resource allocation and management on the grid. Nevertheless, how to devise efficient algorithms for biological and physical applications to take advantage of the potentially unlimited scalability offered by the TeraGrid remains an enormously challenging problem.

2. Motivation and Objectives

The present study is motivated by two grand-challenge problems in biological and physical sciences that are infeasible to solve with conventional supercomputers. The first problem is simulation of blood flow in the *entire* human arterial tree while the second one is direct numerical simulation (DNS) of "drag crisis" of turbulent

1-4244-0054-6/06/$20.00

flow past bluff bodies. Both problems are very significant from both fundamental and applications standpoints, and their resolution will have profound scientific and direct societal impacts.

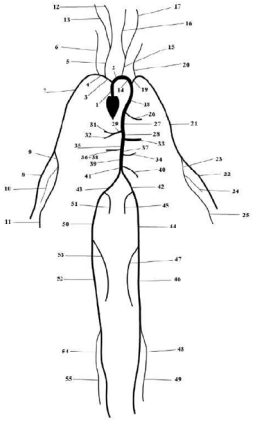

Figure 1: Sketch of the arterial tree containing the largest 55 arteries in human body.

The human arterial tree simulation problem originates from the widely accepted causal relationship between blood flow and the formation of arterial disease such as atherosclerotic plaques. These disease conditions are observed to preferentially develop in separated and recirculating flow regions such as arterial branches and bifurcations. Interactions of blood flow in human arterial system can occur between different scales, or at similar scales in different regions of the vascular system. At the largest scale, the human arterial system is coupled through the wave-like nature of the pulse information traveling from the heart into elastic arteries. Surgical interventions, such as bypass grafts, leading to a blockage of the system alter the wave reflections, which in turn can modify the flow waveforms at seemingly remote locations. Subsequently, the modification of a local waveform can lead to the onset of undesirable wall stresses, possibly starting another pathological event.

The challenge of modeling these types of interactions lies in the high demand for supercomputing to model the three-dimensional unsteady fluid dynamics within sites of interest such as arterial branches. Our goal is to simulate the blood flow in the *entire* arterial tree, which is different from previous investigations on individual arteries, see for example [20, 21, 22, 23]. What makes this type of application amenable to grid computing is that the waveform coupling between the sites of interest can be reasonably modeled by a reduced set of one-dimensional equations, which capture the cross-sectional area and sectional velocity properties [19]. One can therefore simulate the entire arterial tree using a hybrid approach based on a reduced set of one-dimensional equations for the overall system and detailed 3D Navier-Stokes equations at arterial branches and bifurcations. To capture the flow dynamics in an artery bifurcation reasonably well, the grid resolution typically requires a mesh of 70,000 to 200,000 finite elements of high-order; here we use spectral elements with a spectral polynomial order of 10 to 12 on each element [11]. The human arterial tree model in Figure 1(a) contains the largest 55 arteries in the human body with 27 artery bifurcations. The inclusion of all 27 artery bifurcations in the simulation with the above grid resolutions requires a total memory of 3 to 7 terabytes, which is beyond the current capacity of any single supercomputing site available to the open research community in the US. The collective computational resources of the TeraGrid, enabled by MPICH-G2 and Globus-family grid services, makes simulations at these resolutions possible.

The second problem, DNS of "drag crisis" (sudden drop of drag force around Reynolds number Re=300,000) in turbulent bluff-body flows is a fundamental grand challenge problem in fluid dynamics. The need to resolve all the energetic scales in DNS, down to the Kolmogorov scale, dictates that the number of grid points should be on the order of $\mathrm{Re}^{9/4}$, about a trillion grid points at drag crisis conditions. Concentration of turbulence in the bluff-body wake and non-uniform meshing will effectively reduce the required number of grid points to a few billion. The appropriate mesh will consist of about 512 to 768 Fourier modes along the cylinder axis and 50,000 to 80,000 spectral elements in non-homogeneous planes, with a spectral polynomial order 6 to 10 on each element. A monolithic simulation with such resolutions requires over 4 terabytes of memory, exceeding the current capacity of any NSF open supercomputer. Like the human arterial tree simulation, the TeraGrid enabled by Globus and MPICH-G2 becomes a viable choice for carrying out such a grand-challenge simulation.

These extremely large biological and physical simulations are only feasible with computing power similar to the aggregate computing power of the TeraGrid and both share a common characteristic: The solution process requires tightly coupled communications among different TeraGrid sites. This is in sharp contrast to other application scenarios of the grid, for example so called "functional pipelines", in which a monolithic application runs on one grid site while the data produced by the application is moved to another site for visualization or post-processing (e.g. the TeraGyroid project, http://www.realitygrid.org/TeraGyroid.html). At issue here is the scalability of an application involving multiple TeraGrid sites and the slow-down factor of multi-site runs compared to single-site performance under otherwise identical conditions.

To investigate these issues and the feasibility of cross-site runs on the TeraGrid, we consider a *scaled-down* setting of the "drag crisis" problem – simulation of turbulent flow past a circular cylinder at lower Reynolds numbers (Re=3,900 and 10,000) – as a prototype problem

and a simulation of the human arterial system. Through a series of single-site and multi-site experiments on the TeraGrid, we study the scaling up to hundreds of processes of cross-site computations and the slow-down ratio of cross-site runs compared to single-site performance. The objective of this paper is to investigate issues of cross-site computations on the TeraGrid: computation algorithms, feasibility and scalability.

3. Spectral Element Code NEKTAR and MPICH-G2

A high-order CFD code NEKTAR is employed in current computations. It employs a spectral/hp element method [11] to discretize in space and a semi-implicit scheme in time. The mesh consists of structured or unstructured grids or a combination of both, similar to those employed in standard finite element and finite volume methods. Flow variables are represented in terms of Jacobi polynomial expansions. This provides multi-resolution, a way of hierarchically refining the numerical solution by increasing the order of the expansion (p-refinement) within every element without the need to regenerate the mesh, thus avoiding a significant overhead cost.

We employ MPICH-G2 [12] which is a Globus-based MPI library that extends the MPICH implementation of MPI to use services provided by the Globus Toolkit (http://www.globus.org/). The library exploits MPI constructs for performance management and for application-level adaptation to the network topology. During the computation, MPICH-G2 selects the most efficient communication method possible between two processes, using vendor-supplied MPI if available, or otherwise Globus communication for TCP. MPICH-G2 uses information in the Globus Resource Specification Language (RSL) script to create multilevel clustering of the processes based on the underlying network topology, and stores this information as attributes in the MPI communicators for applications to.

4. Multi-Site Computation Algorithms

Here we describe how we used NEKTAR and MPICH-G2 to develop algorithms that are suitable for execution on grids like the TeraGrid to solve the entire human arterial tree and turbulence bluff-body problems described in the previous section.

4.1 Human Artery Tree Simulation

In the hybrid approach for human arterial tree simulations, we employ a system of reduced conservation equations [19] to capture the wave-like interaction of the blood flow. The reduced system is analogous to the shallow-water equations of hydrodynamics or the one-dimensional inviscid equations of gas dynamics. Within a human arterial tree as shown in Figure 1(a) the solution to these equations captures the wave propagation as flow is ejected from the heart and the subsequent reflection of waves at arterial branches and the peripheral vasculature. Detailed information is required at the sites of interest such as the arterial branches that are susceptible to disease onset. Within these branches the 3D unsteady Navier-Stokes equations will be solved. Spectral/hp element algorithms have been developed and validated to solve both the reduced equations and the full Navier-Stokes equations using the code NEKTAR [19].

Therefore, the blood flow simulation in the human arterial network consists of an overall 1D simulation through the full arterial tree and detailed 3D simulations on a number of selected or all the main artery bifurcations. The overall simulation of the arterial tree is loosely coupled. Coupling between different parts of the artery network is through 1D variables only. We next present two multi-site computation paradigms for the artery tree simulation on the TeraGrid: (1) the coordinated workflow model, and (2) the distributed model for the full network.

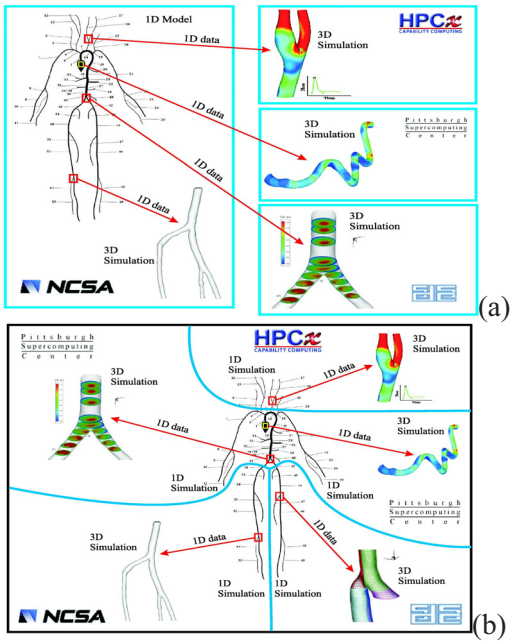

Figure 2: (in color) Artery simulation paradigms on the TeraGrid: (a) Coordinated workflow model: 1D model computed on the master site; Detailed 3D bifurcation simulations are conducted on different TeraGrid sites. 1D results feed 3D simulations as inflow conditions. (b) Distributed model: 1D model is computed across multiple sites; 3D artery bifurcation simulations are performed at the same TeraGrid site as the 1D computation containing that particular bifurcation.

Coordinated Workflow Model: In this approach the 1D simulation of the full arterial tree is conducted on a

single TeraGrid site (master site). This is feasible owing to the limited computational resources involved in the 1D model. Detailed full 3D simulations at selected artery bifurcations are conducted on different TG sites, including the master site itself (see Figure 2(a)). At each time step, the 1D results feed the full 3D simulations with appropriate inflow conditions. Therefore, cross-site communication involves only broadcasts of 1D data from the master site to the other TeraGrid sites. The full 3D simulation of each selected bifurcation is conducted within a TeraGrid site using NEKTAR/MPICH-G2 through domain decomposition.

Processors participating in the entire simulation are partitioned into groups responsible for 1D model computations and for different 3D simulations of artery bifurcations. Three types of process groups are involved in the overall simulation: (1) a group responsible for 1D model computations (1D group); (2) groups responsible for 3D simulations with each corresponding to a selected artery bifurcation (bifurcation group); (3) groups responsible for cross-site boundary condition communications from 1D model to 3D simulations (BC group). Each BC group involves processors of a bifurcation group that contain elements with inflow boundary conditions and a processor in the 1D group that holds the 1D result for that particular bifurcation. Cross-site communications involve only processors in the BC groups, while those involving the 1D group and the bifurcation groups are all intra-site communications. We used the network topology information made available by MPICH-G2 [12] to dynamically create process groups (i.e., MPI communicators) to ensure that all processors of the master group or a bifurcation group come from the same TeraGrid site.

The one-way coupling nature (3D simulations depend on the 1D data, but not the other way) and the computation speed difference between 1D and 3D simulations (1D simulation is significantly faster than 3D simulations) is exploited to overlap cross-site communications with in-site computations. Specifically, in a cross-site communication the master site sends a collection of M time steps of 1D data to 3D simulations using MPI non-blocking calls, M being a tunable parameter depending on the relative speed of 1D and 3D simulations. Therefore, for 3D simulations the in-site computation of current M time steps overlaps with the cross-site receive of the next M steps of inflow data. For 1D simulation the in-site computation of the current M time steps overlaps with the cross-site "send" of the previous M steps of 1D data. The 3D simulations lag behind the 1D simulation by about M time steps.

Distributed Model: In this approach we partition the 1D human arterial tree into different blocks/domains. Each block contains several bifurcations, and is computed on a different TeraGrid site (see Figure 2(b)). A full 3D simulation of an artery bifurcation is computed on the same site as the 1D block containing that particular bifurcation. Therefore, the communications between the 1D and 3D computations are of intra-site type. Cross-site communications are necessary only when solving the overall 1D model. Because an explicit time integration scheme is employed in solving the 1D model and different 1D domains/blocks are coupled through the flux on the domain boundaries only, at each time step only one cross-site communication of the 1D boundary flux between adjacent blocks is involved, regardless of the number of full 3D simulations.

Processors involved in the entire computation are partitioned into groups responsible for the 1D and 3D computations of different bifurcations. There are also three types of process groups involved: 1D Group, bifurcation group and BC group, similar to those in the coordinated workflow model. However, communications involving bifurcation groups and BC groups are now all of intra-site type, while those involving the 1D group are cross-site communications.

4.2 Turbulence Bluff-Body Simulations

Here we consider 3D turbulent flow past bluff bodies such as a circular cylinder. The numerical algorithm we employ features a Fourier spectral expansion in the homogeneous direction (along the cylinder axis) and a spectral element discretization in streamwise-crossflow planes, together with a stiffly-stable pressure-correction type scheme for time integration with a third-order accuracy [11]. To address the enormous computational challenges in extreme conditions such as those encountered in drag crisis simulations, we next present two multi-site parallel algorithms on the TeraGrid based on different data distribution strategies to minimize the number of cross-site communications and to overlap cross-site communications with in-site computations/communications. Both algorithms are designed on top of a two-level parallelization strategy [6, 7].

Fourier Modal-Based Algorithm: The main idea of the first algorithm is to distribute different groups of Fourier modes onto different TeraGrid sites. This is based on the observation that different Fourier modes of a physical variable (three velocity components and the pressure) are de-coupled except in evaluating the FFT when computing nonlinear terms in Navier-Stokes equations. At each site we compute a sub-set of the Fourier modes for all physical variables. As a result, solutions of any physical variable on different sites are largely independent. Coupling among different sites (hence cross-site communication) only occurs in the transposition of distributed matrices, an all-to-all type communication, when evaluating the FFT in the non-linear term calculation.

Special care is taken in the application to minimize the cross-site latency impact and improve the cross-site bandwidth utilization. The network topology information provided by MPICH-G2 is used to enforce the data distribution strategy, to ensure that in the two-level parallelization [6] computations within non-homogeneous planes involve processors from the same site only, and to create MPI communicators based on machine boundaries thus avoiding costly TCP polling for communications involving "`MPI_ANY_SOURCE`" [12] on those communicators. We also agglomerate the data of different physical variables such that a single cross-site matrix transposition is performed instead of several separate transpositions for different variables. Therefore, only two cross-site communications (one forward transform and one backward transform) are needed when computing the non-linear terms. Compared to the usual approach that performs FFTs of different physical variables separately, the data agglomeration minimizes the number of the cross-site communications and increases the size of each message, and therefore reduces the latency effect. The larger message size also improves the cross-site bandwidth utilization.

Physical Variable-Based Algorithm: The main idea of the second algorithm is to compute different physical variables on different TeraGrid sites. The purpose is to take advantage of the coupling characteristics among different physical variables in Navier-Stokes equations. Computations of different velocity components are independent except for their inter-dependence in the non-linear term. A mutual dependence exists between the velocity and pressure: (1) Computation of pressure depends on the velocity divergence, and the non-linear term and velocity gradients on boundaries; (2) Computation of velocity depends on the pressure gradient. We assume three TeraGrid sites for simplicity. All the Fourier modes of a velocity component, together with a third of the pressure Fourier modes, are computed on a different site with this algorithm.

Three cross-site communications are involved in the computation. The first one occurs prior to the non-linear solve. Here the nonlinear solve, and the pressure and velocity solves in subsequent discussions, refer to a three-step time integration scheme [11] and implemented in NEKTAR. Each site needs to communicate its own velocity component to and receive other velocity components from other sites for nonlinear term calculation. This is implemented with a cyclic shift between sites involving non-blocking communications. With the two-level parallelization in NEKTAR a processor at one site only communicates with the corresponding processors at the other two sites. Therefore, the cross-site communication involves only three processors while different processors at the same site participate in parallel independent communications. The second cross-site communication, a SUM reduction for velocity divergence and pressure boundary conditions, occurs prior to the pressure solve. The third cross-site communication takes place prior to the velocity solve for distributing the pressure gradient data (of a third of pressure Fourier modes) to other sites and receive the pressure gradient component that it computes from the other sites. This is implemented with a cyclic shift, and is overlapped with in-site computations using non-blocking communications.

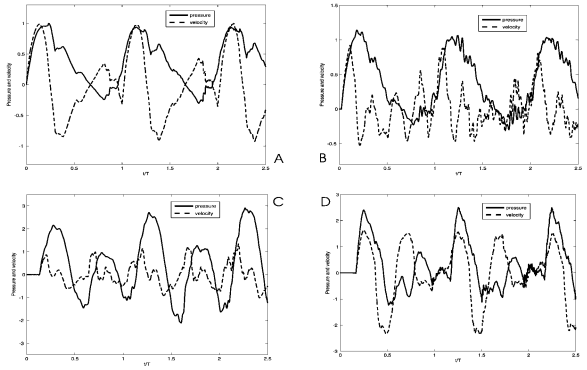

Figure 3: Time histories of velocity (dashed line) and pressure (solid line) waveforms in four arteries: (A) Ascending Arota (artery 1), (B) Right Carotid (artery 5), (C) Right Ulnar (artery 9), and (D) Right Femoral (artery 52). Inflow condition from the heart is modeled as a half sinusoidal wave. Pressure and velocity values in all arteries are normalized by their respective maxima in artery 1.

5. Simulation Results

Before discussing the performance issues in single-site and cross-site computations, we first demonstrate some simulation results of the human artery tree problem and turbulent bluff-body flows obtained on the TeraGrid clusters. Figure 3 shows the time histories of velocity and pressure in four arteries (ascending arota, right arotid, right ulnar and right femoral) of the human artery tree model in Figure 1(a). The inflow from the heart is modeled as a pulsatile half sinusoidal wave. The results show that the pressure and velocity waveforms differ significantly from one region to another in the human arterial network.

In Figure 4 we compare the statistical characteristics in the turbulent wake of a circular cylinder at Reynolds number Re=10,000 between our three-dimensional DNS and PIV (particle-image-velocimetry) experiments [8]. This Reynolds number is the highest one DNS has achieved for this flow so far. The figure shows a comparison of the normalized streamwise *rms* velocity fluctuation u'/U_0 between the experiment (left plot) and the simulation (right plot). Experimental results and DNS results are plotted on identical contour levels, with a

minimum *rms* value $u'/U_0 = 0.1$ and an incremental value of 0.025 between contour lines. The distribution patterns show strong fluctuations in the separating shear layers, and two maxima associated with the vortex formation. The downstream locations of the rms maxima are essentially the same from both experiment and simulation (at *x=1.13*), and the respective peak values are also the same.

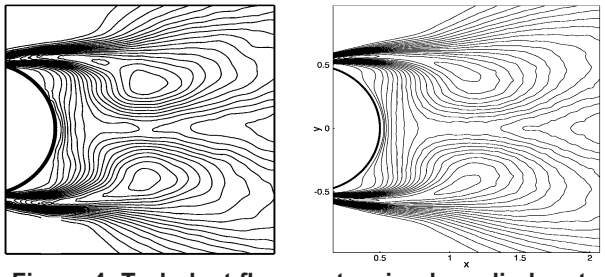

Figure 4: Turbulent flow past a circular cylinder at Reynolds number Re=10,000: Comparison of streamwise rms velocity between PIV experiment (left) and present DNS (right). Contours are plotted on the same levels for experiment and DNS: minimum 0.1 and incremental value is 0.025 between contour lines.

6. Performance Results

In this section we report performance results of a series of controlled experiments of NEKTAR in conjunction with MPICH-G2 on the TeraGrid machines with a turbulent flow past a circular cylinder and the human arterial system. We have conducted single-site and cross-site runs. The NCSA, SDSC, and ANL TeraGrid clusters have Intel IA-64 processors (Itanium-2, 1.5GHz) while those at PSC have Compaq Alpha processors (Alpha EV68, 1GHz).

6.1 Single-Site Performance for Turbulent Flow

To demonstrate the scalability of NEKTAR, in Figure 5(a) we plot the parallel speedup with respect to the number of processors on the PSC TeraGrid cluster for a fixed problem size with 300 million degrees of freedom. The parallel efficiency exceeds 95% on 1024 processors. The test problem here is the turbulent cylinder flow at Reynolds number Re=10,000 based on the free-stream velocity and cylinder diameter. A spectral element mesh with 9272 triangular elements is employed in non-homogeneous planes, and the number of Fourier planes in the spanwise direction is 256 in this test.

The scalability for a fixed workload per processor is another important measure. In this set of tests, as the problem size increases the number of processors is increased in proportion such that the workload on each processor remains unchanged. The test problem is still the turbulent cylinder flow at Re=10,000, and the same grid resolution in non-homogeneous plane is used. We increase the problem size by doubling the number of Fourier planes in spanwise direction. In the test, the number of Fourier planes increases from 8 to 128, and the number of processors is increased proportionally from 32 to 512 to keep the workload per processor constant. Figure 5(b) shows the wall-time per step (in seconds) as a function of the number of processors from the test. The ideal result would be constant wall-time per step for any number of processors (flat curve). Only a slight increase in wall-time is observed as the number of processors increases 32 to 512, indicating a good scalability. Native (vendor) MPI libraries are employed in the above tests.

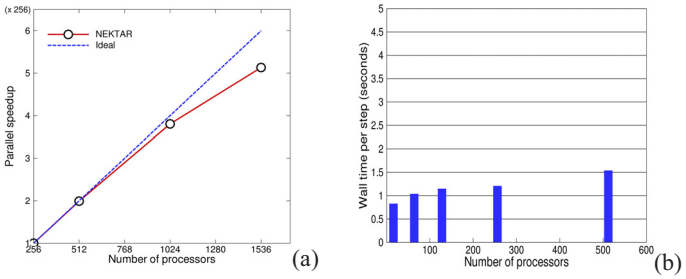

Figure 5: PSC TeraGrid cluster: (a) Speedup vs. CPU for a fixed problem size. (b) Wall time/step vs. CPU for a fixed workload per processor. Speedup is calculated based on the wall-time on 256 CPUs.

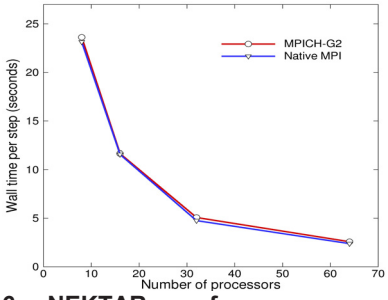

Figure 6. NEKTAR performance comparison between MPICH-G2 and Native MPI on SDSC TeraGrid cluster (single-site): Wall clock time per step versus the number of processors for a fixed problem size.

Because cross-site communications are based on MPICH-G2 library while non-grid enabled applications usually employ native MPI implementations, it is important to quantify the performance differences between MPICH-G2 and native MPI implementation in single-site environment. MPICH-G2 hides from the application the low-level details of operations such as communication channel selection (vendor or TCP), data conversion, resource allocation and computation management, which may induce a performance penalty. Measures have been taken in MPICH-G2 to minimize the overhead cost [12], for example, by eliminating memory copies and unnecessary message polling. Figure 6 shows a performance comparison of NEKTAR compiled with

MPICH-G2 and with native MPI on the TeraGrid cluster at SDSC. We plot the wall-time per step as a function of the total number of processors for a fixed problem size. The test problem is the turbulent cylinder flow at Reynolds number Re=3,900. We employ a spectral element mesh with 902 triangular elements in non-homogeneous planes, and 128 planes in the spanwise direction in the tests. The polynomial order is 8 on all elements. MPICH-G2 demonstrates a performance virtually identical to the native MPI, indicating a negligible overhead cost.

6.2 Cross-Site Performance

A series of cross-site computations are performed with NEKTAR and MPICH-G2 between the TeraGrid machines at SDSC and NCSA for up to 256 processors employing the Fourier modal-based cross-site algorithm (see section 4.2). We next investigate the scaling in cross-site runs for cases with a fixed problem size and with a fixed workload per processor.

The test problem is the turbulent flow past a cylinder at Reynolds number Re=3,900. A spectral element mesh with 902 triangular elements is employed in non-homogeneous planes (with a spectral element order of 8 on all elements). The number of Fourier planes in the spanwise direction varies from 16 to 128. We first investigate the scaling for a fixed problem size with 128 Fourier planes in the spanwise direction. Figure 8 shows the wall clock time per step (in seconds) as a function of the total number of processors for cross-site runs between NCSA and SDSC TeraGrid machines, together with results for single-site runs on the NCSA TeraGrid machine under identical configurations. The total number of processors varies from 16 to 256. For cross-site runs, in each case half of the processors are from the NCSA TeraGrid machine and the other half are from the SDSC TeraGrid machine. For example, in a 256-CPU cross-site run 128 processors are from both NCSA and SDSC. MPICH-G2 is employed in both single-site and cross-site runs, and at least three independent runs are performed for each case. It is observed that in single-site runs the wall-time shows essentially a linear relationship with respect to the number of processors, indicative of a near-linear speedup. In cross-site runs, the wall-time decreases significantly with increasing number of processors. The wall time-CPU curve shows a dramatic decrease, nearly an order or magnitude, as the number of processors increases from 32 to 64. To check if this performance jump results from inadvertent factors, we have taken special care to ensure that we have obtained identical, correct computation results in all the test cases, including those on 32 and 64 processors, and have conducted a number of independent runs for each case (at least three for smaller CPU counts, at least five for larger CPU counts). The tests were conducted at special reserved time for both machines, with exclusive access to about a third of the TeraGrid machine at NCSA and the whole TeraGrid machine at SDSC. The performance results are repeatable, with only slight variation in exact values (Figure 8 shows the mean values). We are convinced that these are not spurious data points. Although the exact reason for this performance jump is not totally clear at this point, we suspect that it is related to the communication characteristics of the network connecting NCSA and SDSC. As expected, a cross-site run is slower than the corresponding single-site run on the same total number of processors. The slow-down ratio, however, decreases dramatically as the number of processors increases. Beyond 32 processors the slow-down ratio of the cross-site runs ranges from 1.5 to 2.0.

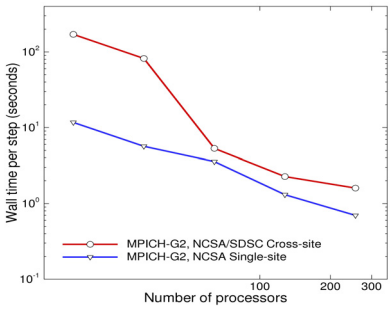

Figure 8: Benchmarking of NEKTAR/MPICH-G2 for a fixed problem size (Turbulent bluff-body flow): NCSA/SDSC cross-site runs and NCSA single-site runs. In NCSA/SDSC cross-site runs half processors are from NCSA and the other half are from SDSC. Shown is the wall time/step as a function of CPUs for a fixed problem size.

We examine next the scaling for a fixed workload per processor. The problem size is varied by changing the number of Fourier planes in spanwise direction. In this set of tests we start with 8 Fourier planes in the spanwise direction, and double the number of Fourier planes each time until it reaches 128. Correspondingly, we increase the total number of processors proportionally, from 8 to 128 processors, such that the workload on each processor remains unchanged. In Figure 9 we plot the wall-time per step (in seconds) as a function of the total number of processors, or equivalently the problem size, for cross-site runs between NCSA-SDSC TeraGrid machines, as well as results for single-site runs on NCSA TeraGrid machine only under identical configurations. In cross-site runs, again half of the processors are from NCSA and the other half from SDSC, and MPICH-G2 is employed in both cross-site and single-site runs. Ideally, a constant wall-time would be observed for all cases. In single-site runs the wall-time increases very slightly as the number of processors increases from 8 to 128, indicating an excellent scalability. In cross-site runs, we observe a larger increase in wall-time as the number of processors increases from 8 to 32. Again a dramatic decrease in wall-time is observed as the number of processors increases from 32 to 64.

Compared to single-site runs on the same number of processors, the slow-down ratio of cross-site runs decreases significantly beyond 32 processors.

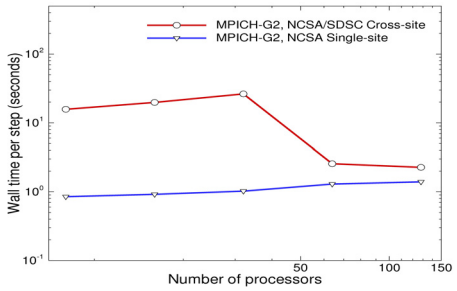

Figure 9: Benchmarking of NEKTAR/MPICH-G2 for a fixed workload per processor (turbulent bluff-body flow): cross-site (NCSA-SDSC) and single-site (NCSA only) runs. As the number of processors increases, the problem size increases proportionally such that the workload per processor remains unchanged. Shown is wall time/step as a function of CPUs. In NCSA/SDSC cross-site runs half processors are from NCSA and the other half are from SDSC.

We have also examined the influence of processor configuration on each TeraGrid site on the performance of cross-site runs. Table 1 lists the wall-time per step on a total of 256 processors in cross-site runs between NCSA and SDSC with a fixed problem size for turbulent cylinder flow at Re=10,000. Several different configurations are tested with different number of processors from each site. The wall-timing for different configurations is essentially the same, and no significant influence of processor configuration on the performance is observed.

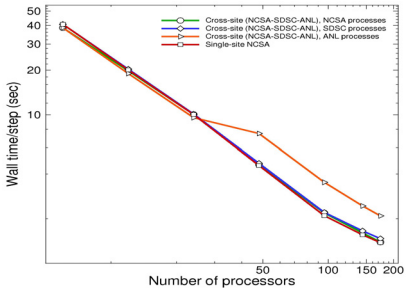

Figure 10: Benchmarking of NEKTAR/MPICH-G2 for a fixed problem size in (human arterial tree simulation, problem size: a total of 3 arteries) in cross-site (NCSA-SDSC-ANL) and single-site runs (NCSA only). In cross-site runs, one third of CPUs are from each site. Shown is wall time/step as a function of total number of CPUs.

We also studied the scalability of our coordinated workflow model for human arterial system simulation. In Figures 10 and 11 we see the scalability of a fixed-size problem run up to nearly 400 processes in Figure 11 and nearly 200 processes in Figure 10. Two of the lines in Figure 11 represent single-site runs at NCSA and PSC and the remaining three lines represent the same fixed-size problem spread evenly across the three TeraGrid sites NCSA, SDSC, and PSC as reported by processes running at each of the three sites, respectively. Likewise, Figure 10 depicts one line representing a single site run at NCSA and three lines solving the same fixed-size problem spread evenly across NCSA, SDSC, and ANL. Since ANL TeraGrid cluster has a limited number of processors we had to limit the total number of processes in Figure 10's experiments so as to keep the distribution of processes even across the three TeraGrid sites. As these figures show, there is no appreciable difference in scalability as we move our application from a single site to up to three sites across the grid. This demonstrates that our coordinated workflow solution to the human arterial simulation problem effectively tolerates the heterogeneous network characteristics found in computational grids, specifically the significant differences in inter-cluster vs. intra-cluster latency and bandwidth, that is often the demise for grid applications. Figure 12 shows the performance for cross-site simulation using four TeraGrid sites (NCSA, PSC, SDSC, TACC) for a fixed problem size for up to about 1500 processors, again demonstrating a near linear speedup.

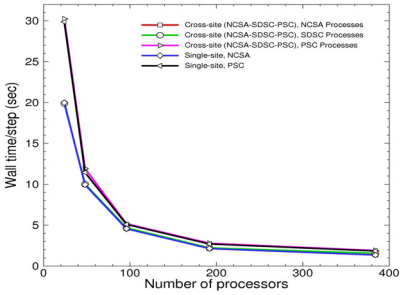

Figure 11: Benchmarking of NEKTAR/MPICH-G2 for a fixed problem size (human arterial tree simulation, problem size: a total of 6 arteries) in cross-site (NCSA-SDSC-PSC) and single-site runs (NCSA only, and PSC only). In cross-site runs, one third of CPUs are from each site. Shown is wall time/step as a function of total number of CPUs.

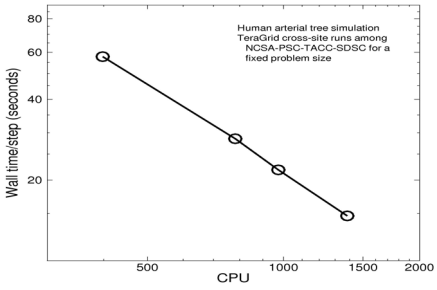

Figure 12: Benchmarking of NEKTAR/MPICH-G2 for a fixed problem size (human arterial tree simulation) in cross-site (NCSA-SDSC-PSC-TACC)

runs. Shown is wall time/step as a function of total number of CPUs.

Table 1. Effect of processor configurations on the performance of cross-site runs for flow past a cylinder at Reynolds number Re=10,000.

Cross-site run	CPUs from NCSA	CPUs from SDSC	time/step (sec)
Total 256 CPUs	128	128	16.31
	144	112	16.47
	160	96	15.94

Finally, we studied the scalability of our application as it used the TeraGrid to solve problems of increasing size. Figure 13 depicts to execution cycles of our human arterial application, one that was started at SDSC and a second that was started at NCSA. Each line starts with a problem solved at a single site. As we add more sites we add the same number of processors at each and we increase the problem size by a factor equal to the original problem size (i.e., we double the problem size when running at two sites and triple the problem on three sites). We observe essentially linear scalability as we add increase the problem size proportionally with the number of sites and compute power once again demonstrating our application's ability to effectively scale in a heterogeneous grid environment.

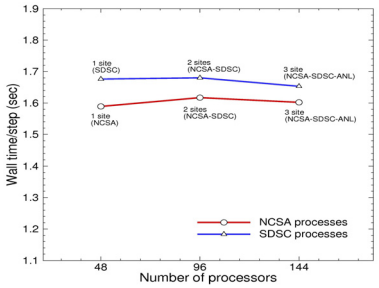

Figure 13: Benchmarking of NEKTAR/MPICH-G2 for a fixed workload per TeraGrid site (human arterial tree simulation). As the number of arteries increases in simulations, the number of TeraGrid sites increases proportionally such that the workload per site remains unchanged.

7. Summary

We have presented experimental results of high-order spectral element code NEKTAR in conjunction with MPICH-G2 in a series of single-site and cross-site runs on the TeraGrid machines at NCSA, SDSC, PSC, and ANL. NEKTAR is representative of large-scale scientific applications characterized by strongly coupled computations and communications. Our interest has been the application scenarios of the grid in which the solution process requires tightly coupled communications among different TeraGrid sites, as would be required by extremely large biological and physical simulations. Test results demonstrate excellent scalability of cross-site computations. Compared to single-site performance on the same number of processors, the slow-down ratio of cross-site runs decreases significantly as the number of processors increases. For the test cases considered, the slow-down ratio ranges from 1.5 to 2.0 as the number of processors increases above 32. TeraGrid cross-site computations are performed with a Fourier-modal based algorithm, which is characterized by a stressful all-to-all type cross-site communication for the transposition of distributed matrices. For other applications characterized by less stressful communication patterns such as the human artery tree simulation we achieved near-linear scalability when running fixed-size problems and problems of increasing size for multi-site runs on the TeraGrid. Our findings confirm that our solutions to these two grand challenge problems do, in fact, scale on computational grids and can effectively tolerate the heterogeneity inherent in computational grids that plague most applications also covetous of grids' computational power and memory capacity. Hoekstra and Sloot [24] provide an instructive framework for understanding the performance of parallel applications in a homogeneous Grid environment. We are further analyzing the results although the non-homogeneity of the TeraGrid presents challenges for measuring the parameters.

Cross-site performance can be further boosted, to even possibly match single-site performance, with a number of techniques, for example, through multithreading to truly overlap cross-site communications with in-site computations, and through UDP based messaging to improve inter-site communication bandwidth utilization [10, 13]. These techniques are currently being incorporated into the next-generation implementation of MPICH-G2.

Acknowledgements

This work was supported by NSF, ONR and DARPA. Computer time was provided by the TeraGrid through NCSA, SDSC and PSC. The help from Spencer Sherwin, Alex Yakhot, Leopold Grinberg is greatly appreciated. We would also like to thank John Towns (NCSA), Rob Pennington (NCSA), David O'Neal (PSC), Donald Frederick (SDSC), and the TeraGrid support team for their assistance.

References

1. B. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, A. Shoshani, B. Drach and D. Williams. High-performance remote access to climate simulation data: A challenge

problem for data grid technologies. *Supercomputing 2001 (SC01)*, Denver, November 2001.

2. G. Allen, T. Dramlitsch, I. Foster, N.T. Karonis, M. Ripeanu, E. Seidel and B. Toonen. Supporting efficient execution in heterogeneous distributed computing environments with Cactus and Globus. *Proceedings of Supercomputing 2001 (SC01)*, Denver, November 2001.
3. M.S. Allen and R. Wolski. The livny and Plank-Beck problems: studies in data movement on the computational grid. *Proceedings of Supercomputing 2003 (SC03)*, November 2003.
4. W. Benger, I. Foster, J. Novotny, E. Seidel, J. Shalf, W. Smith, P. Walker. Numerical relativity in a distributed environment. *Ninth SIAM Conference on Parallel Processing to Scientific Computing*, April, 1999.
5. S. Brunett, D. Davis, T. Gottschalk, P. Messina and C. Kesselman. Implementing distributed synthetic forces simulations in metacomputing environments. *Proceedings of Heterogeneous Computing Workshop*, March 1998.
6. S. Dong and G.E. Karniadakis. Multilevel parallelization models in CFD. *Journal of Aerospace Computing, Information, Communication*. 1, 256-268, 2004a.
7. S. Dong and G.E. Karniadakis. Dual-level parallelism for high-order CFD problems. *Parallel Computing*, 30, 1-20, 2004b.
8. A. Ekmekci. Flow structure from a circular cylinder with defined surface perturbations. Ph.D. Dissertation, Department of Mechanical Engineering and Mechanics, Lehigh University, 2005.
9. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a future computing infrastructure*. Morgan Kaufmann Publishers, 1999.
10. E. He, J. Leigh, O. Yu and T.A. Defanti. Reliable blast UDP: Predictable high performance bulk data transfer. Presented in *IEEE Cluster Computing*, 2002.
11. G.E. Karniadakis and S.J. Sherwin. *Spectral/hp element methods for CFD*. Oxford University Press, 1999.
12. N.T. Karonis, B. Toonen and I. Foster. MPICH-G2: A Grid-enabled implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing*, 63, 551-563, 2003a.
13. N. Karonis, M.E. Papka, J. Binns, J. Bresnahan, J. Insley, D. Jones and J. Link. High-resolution remote rendering of large datasets in a collaborative environment. *Future Generation Computer Systems*, 19, 909-917, 2003b.
14. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing, 20, 359-392, 1998.
15. S.J. Kovacs, D.M. McQueen and C.S. Peskin. Modelling cardiac fluid dynamics and diastolic function. *Philosophical Transactions of the Reyal Society of London Series A –Mathematical Physics and Engineering Science*, 359, 1299-1314, 2001.
16. G. von Laszewski, J.A. Insley, I. Foster, J. Bresnahan, C. Kesselman, M. Su, M. Thiebaux, M.L. Rivers, S. Wang, B. Tieman and I. McNulty. Real-time analysis, visualization and steering of microtomography experiments at photon sources. *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, April, 1999.
17. M. Ripeanu, A. Iamnitchi and I. Foster. Performance predictions for a numerical relativity package. *International Journal of Supercomputing Applications*, vol 15(4), 2001.
18. M. Russel, G. Allen, G. Daues, I. Foster, E. Seidel, J. Novotny, J. Shalf and G. von Laszewski. The astrophysics simulation collaboratory: A science portal enabling community software development. *Cluster Computing*, 5(3), 297-304, 2002.
19. S.J. Sherwin, L. Formaggia, J. Peiro and V. Franke. Computational modeling of 1D blood flow with variable mechanical properties in the human arterial system. *International Journal for Numerical Methods in Fluids*, 43, 673-700, 2003.
20. A. Tirado-Ramos, P.M.A. Sloot, A.G. Hoekstra and M. Bubak. An integrative approach to high-performance bio-medical problem solving environment on the Grid. *Parallel Computing*, 30, 1037-1055, 2004.
21. E.V. Zudilova, P.M.A. Sloot and R.G. Belleman. A multi-modal interface for an interactive simulated vascular reconstruction system. *Fourth IEEE ICMI'02 International Conference on Multimodal Interfaces*, Pittsburgh, PA, Oct. 2002.
22. A.M.M. Artoli, A.G. Hoekstra and P.M.A. Sloot. Simulation of a systolic cycle in a realistic artery with the Lattice Boltzman BKG method. *International Journal of Modern Physics* B, 17, 95-98, 2003.
23. A.M.M. Artoli, A.G. Hoekstra and P.M.A. Sloot. Accelerated Lattice BGK method for unsteady simulations through Mach number annealing. *International Journal of Modern Physics C*, 14, 835-847, 2003.
24. A.G. Hoekstra and P.M.A. Sloot. Introducing the grid speedup Γ: A scalability metric for parallel applications on the grid. *Lecture Notes in Computer Science*, 3470, 245-254, 2005.