An Enabling Environment for Distributed Simulation and Visualization

Yao Zheng[1,2], Guanghua Song[1,2], Jifa Zhang[1,2], Jianjun Chen[1,2]
1 College of Computer Science, Zhejiang University, Hangzhou, 310027, P. R. China
2 Center for Engineering and Scientific Computation, Zhejiang University, 310027, P. R. China
E-mail: yao.zheng@zju.edu.cn, ghsong@cs.zju.edu.cn

Abstract

In this paper we present the MASSIVE project that aims to use Grid technology to establish an enabling environment for distributed simulation and visualization of large-scale scientific and engineering research. The project focuses on collaborative numerical simulation and visualization in grid environments. Furthermore, the project is also to develop Grid-enabled capability and products for use by the wider community. The current status of the MASSIVE project is reported and a prototype Grid utilizing resources at the Center for Engineering and Scientific Computation (CESC) in ZheJiang University (ZJU) is discussed. This Grid gets a model geometry generated by CAD systems and transfers it to an SGI Onyx 3900 supercomputer where it is meshed. The meshed geometry is then transferred to a DAWNING PC Cluster where it is used to solve computational fluid dynamics and computational solid mechanics problems and it is finally visualized showing numerical results.

Key words: Grid, MASSIVE, portal, simulation, CFD, CSM

1. Introduction

As a new parallel computing infrastructure, the Grid [1-3] enables flexible sharing of a collection of resources that is maintained as parts of different administrative domains. Providing the basic middleware infrastructure for bootstrapping a sophisticated collaborative environment, the computational grid allows scientists to share resources even if controlled by different domains. Access to these resources is enabled through the creation and use of “virtual organizations”(VOs).

What scientists in the area of engineering and scientific computing mainly need is a grid-enabled portal, by which they are able to deploy applications easily without concern with the complex details of grid technologies. With graphical user interfaces, the portal is an integrated development environment based on grid services. In this paper, we describe a Multidisciplinary ApplicationS-oriented SImulation and Visualization Environment (MASSIVE). It provides a development and deployment platform for geometrical modeling, discretization, scientific computing and visualization. Its advanced services can be accessed easily in a visual manner. Their integration as a part of a workflow process enables the creation of services that can be easily reused by the community. Scientists are then in a position to concentrate on the science, while platform developers can focus on the delivery of services that can be assembled as building blocks to create more elaborate services.

Large-scale CFD (Computational Fluid Dynamics) and CSM (Computational Solid Mechanics) simulations are computationally intensive, and may involve access to resources that are intrinsically distributed. For example, in the case of an organization in which multiple partners from industry and academia are cooperating to design and to build a complex system that requires CFD/CSM simulations, the geometry of the component may be created at one location, a mesh conforming to this geometry may be generated at a second location, and a CFD/CSM simulation based on the mesh may be performed at a third location. Finally, the output from the simulation may be analyzed and visualized at one or more other locations. In such a situation, the partners in the organization are prepared to share data (the geometry, the mesh, and the simulation result), but not the usual proprietary software systems (the mesh generators and problem solvers) and the hardware that generate the data. Thus, it is not feasible to place all the data, computing, and human resources in one geographical location. The Grid, therefore, is an excellent candidate for providing the infrastructure needed to support the organization. It should be noted that it has essentially the same meaning as “virtual organization” [1].

This paper reports on the current status of the MASSIVE project and discusses a prototype Grid integrating resources at the Center for Engineering and Scientific Computation (CESC) in ZheJiang University (ZJU). This Grid is capable of taking a model geometry generated by CAD Systems, transferring it to a SGI Onyx 3900 supercomputer where it is meshed. The meshed geometry is then transferred to a DAWNING PC Cluster where it is used to solve a computational dynamics problem and visualized.

Important aspects of the research include the deployment and use of Grid-based mesh generation and manipulation services, the secure remote execution of computational numerical simulations on the various hardware resources available at the Grid nodes, including an SGI Onyx 3900 supercomputer and a DAWNING PC Cluster in CESC, and collaborative visualization and analysis of both meshes and the simulation results by geographically dispersed participants.

A key issue in the MASSIVE project, and in the use of the Grid in general, is the capability to make effective use of the distributed resources within the constraint of dynamically changing network bandwidth. Hence there is a need to monitor Grid resources, including the network bandwidth and traffic, and to make scheduling decisions to ensure good services in terms of both performance and fault tolerance.

Another issue that the MASSIVE project will be investigating is how a simulation code can be migrated to remote computers and executed in a secure way. The ultimate aim in this aspect is to ensure that the code is protected against unauthorized access throughout the migration and execution. Although cryptographic techniques exist for securely migrating code, there is currently no mechanism known to be effective for protecting it from interference by privileged users on the remote machine. Achieving a truly secure remote execution will probably require changes in computer architecture and/or chip design. Although this is beyond the scope of the MASSIVE project, techniques for mitigating risks within a trust model for the extended enterprise will be investigated.

The rest of this paper is arranged as follows. Section 2 gives an overview of the current prototype MASSIVE Grid. Sections 3 and 4 describe mesh generation and computational problem solving software, respectively. Approaches to collaborative visualization are considered in Section 5. Section 6 discusses main features of a portal through which users access MASSIVE software and hardware resources. As an application example, section 7 presents a structure analysis of a crank processed upon MASSIVE. Finally, Section 8 summarizes what has been achieved so far in the MASSIVE project, and outlines the main topics of future work.

2. An Overview of the MASSIVE Project

The grid created for the MASSIVE project utilizes resources located at the CESC, and other departments of Zhejiang University. We plan to add resources at other institutions of high performance computing. The grid currently uses Globus Toolkit 2.4.3 (GT2) as the middleware to enable the resources at each site to be accessed in a secure and uniform manner.

For the MASSIVE project we decided that resources should be used according to certificates signed by the CESC Certificate Authority (CA). This allows interoperation with other CESC grids, and relieves the MASSIVE project from the time-consuming task running with its own CA. It also simplifies the granting of access to other researchers within the CESC, who may already have these credentials.

The extent of a grid can be defined using several different criteria. One possibility is to define boundaries of a grid based on a trust horizon, where a common set of credentials potentially acceptable to each resource defines the extent of the grid. This follows from the simplicity of accepting a new user or resource from inside this horizon relative to accepting those from outside it.

A better definition of the extent of the MASSIVE Grid is the awareness horizon, created through the existence of a GIIS (Grid Index Information Service) specifically for the MASSIVE project. The Globus GIIS provides a central point at which users can discover the other services available in a Grid. Once a user has been granted access to the Grid, they simply need to be pointed to the GIIS in order to discover resources in the Grid. By limiting a GIIS to contain only MASSIVE resources, searches are simpler since there is no need for a set of additional attributes in the MASSIVE.

By using GT2, the addition of a new user to the MASSIVE Grid requires creation of accounts on each resource and modification of the standard Globus access control file. At present, there is no means of applying to the VO for access to all resources. Instead, the user must apply to each site for access. We expect to move fairly quickly to an Open Grid Service Architecture (OGSA), in which access to services can be enabled as a single operation with lower administrative overhead at each site. This will be possible since OGSA allows us to provide users with access to specific services without creating individual user accounts as required by GT2. The flexibility of modifying the middleware used by the project quickly is one of the main benefits of creating a grid separate from the main CESC Grid. This is possible because a resource can provide services to multiple grids, as long as the various specifications do not conflict.

The MASSIVE project Grid-enables the essential aspects of industrially relevant CFD/CSM using a service-oriented architecture based on emerging standards such as OGSA. Services for mesh generation and geometry pre-processing, the migration and execution of application programs on remote platforms, collaborative visualization, and data analysis form the basis of a globally distributed VO, in which security and performances are key concerns. A typical usage would be the generation of a mesh using a meshing service on an IRIX platform, the solution of a CFD/CSM problem with the meshes previously created on a PC Cluster, and the collaborative visualization of the numerical results with equipment such as a display wall and a BARCO stereo projection system at the CESC.

A client-server architecture based problem-solving environment (VGrid[4]) provides the main interface through which these services are accessed. The VGrid, as a component of MASSIVE, supports the composition of applications from service-based components, the execution and monitoring of such applications on remote resources, and the collaborative visualization, exploration, and analysis of the numerical results. The VGrid maintains and provides access to an archive of the results. In addition, the MASSIVE environment also provides an interface to the meshing service and supports the visualization capability of meshes created.

The meshing service provides access to integrated surface and volume mesh generators, and a mesh generation toolbox for industrial problems, by which partners can add their own tools. The meshing service accepts CAD data in the widely used IGES format. Mesh analysis services include validity checks and mesh quality statistics. Other tools include a surface patch merger, a geometry handler and facilities for geometry repair. The surface mesh generators provide unstructured triangular quadrilateral meshes, and the ability to mesh across boundaries and non-manifold geometries. The volume mesh generators support tetrahedral, hexahedral, prismatic, and pyramidal meshes. Mesh optimization and enhancement facilities will also be provided.

The ability to execute a simulation on a remote resource is important in a number of industrial settings, for example, when a company does not have the necessary computing power. In such a case the simulation code could be installed on the remote machine. However, this may not always be the best model for remote execution. The company may not want to place a valuable or business-sensitive code on a third-party machine for an extended period of time because of the risk of unauthorized access. To minimize such risks, the company may prefer to keep the code on its own machine and migrate it to the remote machine only when it is to be executed, after which it is deleted from the remote machine. This approach may also be appropriate when code is in development undergoing frequent changes. Services will be provided to support the secure migration, remote execution and deletion of the application program, and the secure return of the application results to the user. The transfer of code and data between machines can be handled securely by encryption. However, once the code is decrypted on the remote machine it is vulnerable [5].

The collaborative visualization and interpretation of complex 3-D meshes and scalar and vectors fields will be another major area of the MASSIVE project. To make effective use of visual cues, such as stereo display walls will be used extensively. The collaborating partners, in general, will be globally distributed, use a heterogeneous set of visualization platforms and software environments, and be subject to dynamic bandwidth constraints. Visualization of large scaled scientific data sets using the Grid is an important aspect of the MASSIVE project.

3. Mesh Generation Service

The MASSIVE environment provides a mesh generation service with capabilities similar to those involved in the parallel simulation user environment (PSUE) [6]. PSUE was developed for pre and post-processing for unstructured mesh-based computational simulation. The attractive aspect of the unstructured mesh technology is the ease and speed with which complex geometrical configurations can be treated. Arbitrary computer application software can be integrated into the environment to provide a multi-disciplinary engineering analysis capability within one unified computational framework. The PSUE harnesses state-of-the-art high performance computing and networking to provide a software environment for a diverse range of applications in computational field simulation. It is aimed to facilitate the optimization of design and to help reduce the number of production cycles in manufacturing industry.

Similar to the PSUE as shown in Figure 1, the MASSIVE provides an enhanced capability for complex and multiple problem definition. Developments aim at integrating CAD and mesh generating capabilities, addressing end-user requirements and data formats. This includes the provision of tools to correct invalid geometries, a graphics environment for guidance through the mesh generation process with visual validation of each step, and robust and computationally efficient unstructured mesh generation modules. The environment also provides the functionality for integrating arbitrary engineering software applications and executing analysis modules on computer platforms, including parallel computers.

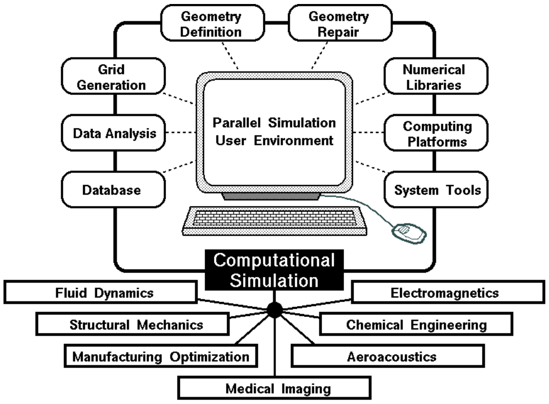


Figure 1. A general schematic of the parallel simulation user environment (PSUE)[6].

4. Computational Problem Solving Service

Due to the compute-intensive nature of large-scale simulations the computational problem solving service used by the MASSIVE is based on parallel solvers developed at CESC for CFD and CSM simulations. The simulations require the use of extremely large meshes and the use of significant computational resources and, in this case, the use of parallel computers becomes essential. The Finite Element Method (FEM) is used by the computational problem solving service in the MASSIVE project.

FEM appears for the first time in the open literature in an article by Clough [7]. Turner et al. [8] used method in nonlinear structural analysis. Computerized nonlinear structural analysis is matured, but has not developed equally in all areas. FEM has various aspects. From a mathematical point of view, it is a scheme to produce an approximate solution, the accuracy of which can be increased as desired. Functional Analysis is essential to understand this point. From a computational point of view, efficiency should be sought in solving actual problems, which result in large degrees of freedom. Sophisticated algorithms need to be implemented in applying FEM, in particular, to non-linear problems. Parallel computing, which fits naturally with FEM, is needed to solve large-scaled problems.

Continuum mechanics is the study of bodies at the macroscopic level, using continuum models in which the microstructure is homogenized by experimental averages. The two traditional areas of applications are solid and fluid mechanics. The former includes structures, which, for obvious reasons, are fabricated with solids. Computational solid mechanics takes an applied sciences approach, whereas computational structural mechanics emphasizes technological applications to the analysis and design of structures. Finally, system identifies mechanical objects, whether natural or artificial, that performs a distinguishable function. Examples of man-made systems are airplanes, building, bridges, engines, cars, microchips, radio telescopes, robots, roller skates and garden sprinklers.

The FEM is viewed as a procedure for obtaining numerical approximations to the solution of Boundary Value Problems (BVPs) posed over a domain. This domain is replaced by the union of disjoint sub-domains called finite elements. In general, the geometry of the domain is only approximated by that of union of sub-domains. The unknown function (or functions) is locally approximated over each element by an interpolation formula expressed in terms of values taken by the function(s), and possibly its derivatives, at a set of node points generally located on the element boundaries. The states of the assumed unknown function(s) determined by unit node values are called shape functions. The union of shape functions "patched" over adjacent elements forms a trial function basis for which the node values represent the generalized coordinates. The trial function space may be inserted into the governing equations and the unknown node values determined by the Ritz method (if the solution extremizes a variational principle) or by the Galerkin, least-squares or other weighted-residual minimization methods if the problem cannot be expressed in a standard variational form.

The FEM computation can be parallelized in the following manner:

1. Divide the global mesh data into multiple parts and transfer them to distributed compute power locations in the MASSIVE Grid;
2. Run the solver program; and
3. Collect result data set and combine them.

5. Visualization

MASSIVE supports the collaborative visualization of meshes and visual steering of CFD/CSM simulations. The visualization will be implemented by our on-going project named ViG (Visualization in Grid). ViG makes use of the DAWNING PC-cluster. The cluster has 9 rendering nodes. Each node has a dual-head FX5200 display card connected to a projector. The projectors are deployed to generate stereoscopic 3 x 3 images on the large display wall.

ViG will provide input to the development of this service and will develop a fully Grid-enabled extension of standard rendering APIs (such as openGL), to allow scientists to run Grid applications through the ViG user interface.

ViG is a problem-solving environment that enables a user to perform unstructured mesh generation in parallel, to execute CFD and CSM solvers on parallel platforms, and to perform the visualization of grid data and visual steering of the solution. ViG allows the user to visualize large geometry and mesh data sets and perform any necessary tasks to define inputs to other modules such as mesh sources and boundary conditions, and perform any mesh quality analysis and solution post-processing operations. In ViG, mesh generation and CFD/CSM computations are done on the SGI supercomputer, and the rendering is done on the PC-cluster. The large volume data sets are stored and traversed on the parallel server with only the geometric primitives that are to be actually rendered being transmitted to the cluster. It is anticipated that two types of collaborative visualization will be supported in MASSIVE. In the first, one participant will be designated as the leader and will be able to navigate the data by selecting the viewpoint, lighting conditions, and other related attributes. In this mode all participants will see the same scene as the leader, and will be able to interact via audio links. In the second approach each participant will be able to navigate the data independently. The idea here is that when a participant discovers something of interest, other participants can subscribe to that participant's view. In effect, this forms a sub-group of participants with a leader, as in the first mode.

6. MASSIVE Portal

User access to all MASSIVE software and hardware resources will be via an application portal, or problem-solving environment. This portal will hide the complexity of the Grid system from the end-user. The main functions of the MASSIVE portal are as follows:

1. To provide an environment for graphically composing applications. Typically this will involve selecting geometry data that is passed to the mesh generator.
2. To provide access to a Grid execution environment to which MASSIVE applications can be submitted. This execution environment is responsible for scheduling the application and returning the results to the portal.
3. To support the collaborative analysis and visualization of results.

The MASSIVE portal is currently being designed, previous named "VGrid", and it is intended to build on Client/Server techniques by making using of GT2.

As the portal of MASSIVE, VGrid is located between grid middleware and engineering applications. It provides an integrated development environment, abstracts basic grid services (such as authorization, authentication, MDS, GRAM, GASS, etc.) provided by grid middleware, and provides engineering applications-oriented high level VGrid services. Using VGrid services by their API interfaces, applications can be developed and deployed more efficiently under the grid computing environment. The client component of VGrid provides a graphical interface for grid-based application development and execution. VGrid consists of system manager, services manager, application development environment, task manager and result manager. Their functionality is itemized as follows.

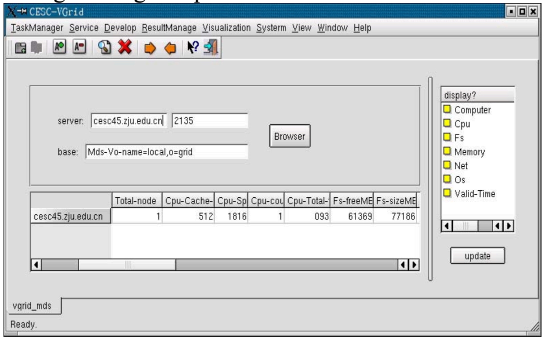
1. The functions of the VGrid system manager include VGrid user management, VGrid runtime environment setup, virtual organization management, security management, and local gatekeeper setup.
2. The functions of the services manager include the supplying, management and upgrading of services modules, such as geometric modeling and visualization.
3. The functions of the application development environment include an auxiliary toolkit for engineering computation program development.
4. The functions of the task manager include task (also named VGrid job) descriptor, task scheduler, task executor and task monitor.
5. The functions of the result manager include collection of resulting data, cleaning of unused files and local, remote, distributed or collaborated visualization of resulting data.

In order to satisfy different requirements, VGrid users are classified into three levels (User A, User B and User C) according to their motivation. User A has some developed application programs, but does not know about grid technologies. He only wants to find out proper resources to execute his programs. User B also has some developed application programs, and has some acquaintance with grid technologies and utilization of the grid-based distributed resources. He wants to choose the proper resources to deploy his program and monitor the processes of submission, execution and result processing. User C wants to develop applications in the grid or to upgrade an existing program to make it fit for the grid environment. He is very familiar with grid technologies, and wants to take full advantage of grid services to develop and deploy his applications in order to use resources more efficiently.

6.1 Resources Discovery

The execution of a computing task needs a lot of resources. The resources are distributed, dynamic and autonomous in MASSIVE Grid. How many resources are available? Which should be allocated to a specified task in available resources? These problems must be answered during task scheduling. To answer these two questions, resources information services provide two interfaces, a hierarchical resources tree (as shown in Figure 2), and a resources list. The content of the tree is the detailed information regarding hosts in virtual organizations, where local hosts are registered. It also shows the architecture of those virtual organizations. In the list, each item contains information of an individual host. The information on the list can be customized for display. Generally it consists of host name, node count, CPU count, speed and free percentage of CPU, free memory size, free secondary storage, network bandwidth and latency.

In the MASSIVE Grid, resources are divided into two classes, computing power and visualization devices. The computing power includes personal computers, PC clusters, workstations and supercomputers. The visualization devices are special distributed display devices, and are used to display high fidelity images and sophisticated graphic results of engineering computation.



server: cesc45.zju.edu.cn 2135

base: Mds-Vo-name=local,o=grid

	Total-node	Cpu-Cache-	Cpu-Sp	Cpu-cou	Cpu-Total-	Fs-freeME	Fs-sizeME
cesc45.zju.edu.cn	1	512	1816	1	093	61369	77186

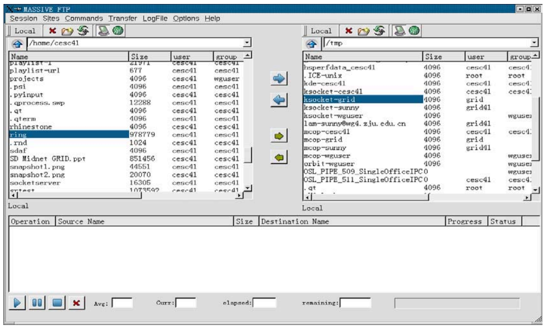
Ready.

Figure 2. The snapshot of the MDS search result

6.2 File Transfer

To satisfy the requirement of CFD/CSM applications, MASSIVE integrates GASS (Global Access to Secondary Storage) service and GridFTP service into visual file transfer service. The user interface of the service is shown in Figure 3. Visual file transfer service supports remote I/O in a grid environment based on GSI and Kerberos security mechanism. Under authentication, it also supports third-party controlled data transfer mode used to transfer large data sets in the distributed communication environment.

Generally, there is a lot of distributed file transfer to perform during parallel computation, such as (1) delivering of executable programs, parameter files, and necessary libraries, and (2) collection of computation results among distributed storage nodes. It is convenient to monitor such process of transfer by making use of the visual file transfer service in the MASSIVE Grid.



/home/cesc41

Name	Size	user	group
playlist-1	21971	cesc41	cesc41
playlist-url	677	cesc41	cesc41
projects	4096	cesc41	wguser
.psi	4096	cesc41	cesc41
.pyinput	4096	cesc41	cesc41
.qprocess.swp	12288	cesc41	cesc41
.qt	4096	cesc41	cesc41
.qterm	4096	cesc41	cesc41
rhinestone	4096	cesc41	cesc41
ring	978779	cesc41	cesc41
.rnd	1024	cesc41	cesc41
sdaf	4096	cesc41	cesc41
SD Midnet GRID.ppt	851456	cesc41	cesc41
snapshot1.png	44551	cesc41	cesc41
snapshot2.png	20070	cesc41	cesc41
socketserver	16305	cesc41	cesc41
srtest	1073592	cesc41	cesc41

/tmp

Name	Size	user	group
hsperfdata_cesc41	4096	cesc41	cesc41
.ICE-unix	4096	root	root
kde-cesc41	4096	cesc41	cesc41
ksocket-cesc41	4096	cesc41	cesc41
ksocket-grid	4096	grid	
ksocket-sunny	4096	grid41	
ksocket-wguser	4096		wguser
lam-sunny@wg4.zju.edu.cn	4096	grid41	
mcop-cesc41	4096	cesc41	cesc41
mcop-grid	4096	grid	
mcop-sunny	4096	grid41	
mcop-wguser	4096		wguser
orbit-wguser	4096		wguser
OSL_PIPE_509_SingleOfficeIPC	0		wguser
OSL_PIPE_511_SingleOfficeIPC	0	cesc41	cesc41
.qt	4096	root	root

Operation	Source Name	Size	Destination Name	Progress	Status

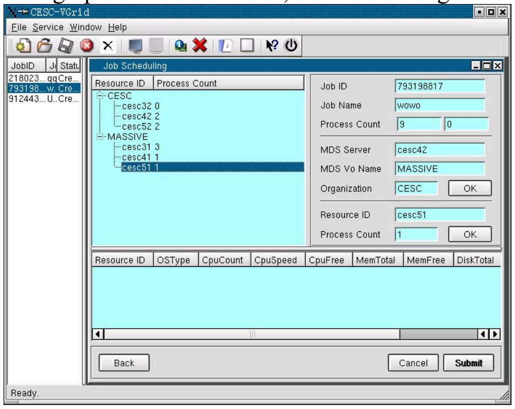
Avg: Curr: elapsed: remaining:

Figure 3. The snapshot of the FTP interface in MASSIVE

6.3 Dynamic Resource Allocation

The MASSIVE portal provides a graphical user interface to submit a new task or to update an existing one. Information for submitting a new task includes executable program, resources requirements and schedule requirements. When the portal accepts the information, an instance of the task is initialized and added to the logical resources allocation queue.

Dynamic resource allocation includes two steps, logical resources allocation and physical resources allocation. During the logical resources allocation, tasks in the queue gain logical resources by FIFO algorithm. A logical resource is a type of virtual resource. It includes resource name, node count, and CPU count. The physical resources are not allocated until the task is scheduled to execute, because resources in the grid are dynamic. The next step of this phase is task scheduling (mapping logical resources to physical resources). A task is added to ASQ (auto schedule queue) or ISQ (interactive schedule queue) according to its attribute of scheduling control. Tasks in ASQ will be scheduled automatically with the deadline and budget constrained cost-time optimization algorithm [9]. Tasks in ISQ should be scheduled manually: that is, the user can select resources manually from the available resources list as required with a graphical user interface, as shown in Figure 4.



JobID	J	Statu
218023...	qq	Cre...
793198...	w.	Cre...
912443...	U.	Cre...

Resource ID	Process Count
CESC	
cesc32	0
cesc42	2
cesc52	2
MASSIVE	
cesc31	3
cesc41	1
cesc51	1

Job ID: 793198817
Job Name: wowo
Process Count: 9 0
MDS Server: cesc42
MDS Vo Name: MASSIVE
Organization: CESC
Resource ID: cesc51
Process Count: 1

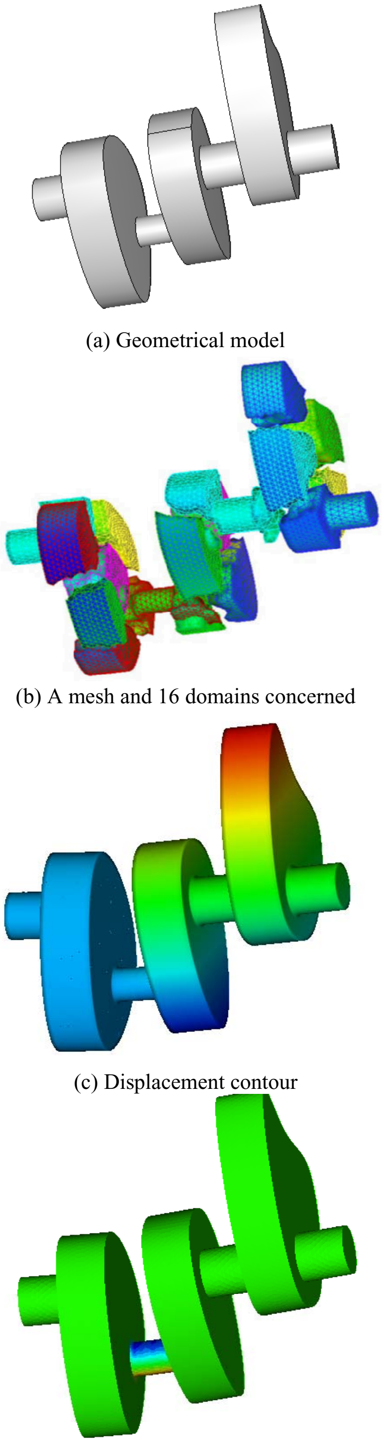
Resource ID	OSType	CpuCount	CpuSpeed	CpuFree	MemTotal	MemFree	DiskTotal

Figure 4. Snapshot of task-scheduler in MASSIVE

7. Applications

As an example, a structure analysis of a crank is processed upon MASSIVE. After geometrical modeling, the crank is decomposed into 16 domains and delivered to different processors for parallel execution, and at last the data is visualized. We conduct this simulation on an SGI Onyx 3900 supercomputer and a Dawning PC cluster with diverse parameters to explore the performance of the system.

Fig. 5 shows the corresponding geometrical model, discretization of the model and the simulation result. The simulation has been carried out with various numbers of elements and those of CPUs on both the supercomputer and the PC cluster. Fig. 6 shows the efficiency for cases with the numbers of elements and CPUs. From the figure we can see that parallel processing is capable of reducing the computing time greatly. However, more CPUs cannot always guarantee higher performance. When too many CPUs are utilized, domain decomposition, data transfer and other communication operations will take lots of time and reduce the performance.



(a) Geometrical model

(b) A mesh and 16 domains concerned

(c) Displacement contour

(d) Deformed shape with stress contour

Fig. 5. Simulation of a crank

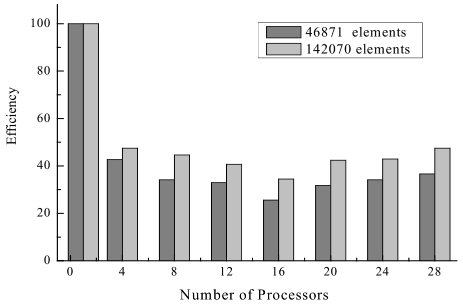


Fig. 6. Efficiency for cases with various numbers of elements and those of CPUs on SGI Onyx3900

8. Summary and Future Work

The MASSIVE project is still in the middle stages of development. However, a simple prototype of the MASSIVE Grid described in this paper has been implemented and demonstrated. This grid can be used to perform distributed CFD/CSM simulations in which geometry data resides at one site, and the mesh generation, CFD/CSM simulation and visualization of output, are performed at other sites. These tasks, as well as the flow of data between them, are controlled by the MASSIVE portal. Future work will focus on the following key areas:

1. Upgrading the Grid. We intend to migrate the MASSIVE Grid swiftly to a GT3-based one. Key tasks in this migration will be representing the mesh generator and CFD/CSM simulation solver(s) as OSGA compliant services that interact via messaging service.
2. Developing the Web-based MASSIVE portal. The current MASSIVE portal is Client/Server architecture.
3. Developing support for collaborative analysis, visualization, and visual steering. The initial effort here will be on post-simulation examination and navigation of the results, but the ability to interactively steer a simulation without interrupting its execution is also of interest.

9. Acknowledgements

The authors wish to thank the National Natural Science Foundation of China for the National Science Fund for Distinguished Young Scholars under grant Number 60225009. We would like to thank the Center for Engineering and Scientific Computation, Zhejiang University, for its computational resources, with which the research project has been carried out.

10. References

[1] Ian Foster, Carl Kesselman, Jeffrey Nick, and Steven Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, *http://www.globus.org/research/papers/ogsa.pdf,* February 2002.

[2] Gregor von Laszewski, Gail Pieper, and Patrick Wagstrom, *Performance Evaluation and Characterization of Parallel and Distributed Computing Tools, Chapter Gestalt of the Grid*, Wiley Book Series on Parallel and Distributed Computing, 2002.

[3] I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *International Journal of Supercomputer Applications*, 15(3), 2001.

[4] Guiyi Wei, Yao Zheng, Jifa Zhang, Guanghua Song, An Engineering Computation Oriented Visual Grid Framework, Proceedings of the Second International Workshop on Grid and Cooperative Computing (GCC2003) (Shanghai, China, 2003), *Grid and Cooperative Computing: Second International Workshop, GCC 2003, Part I, Lecture Notes in Computer Science, Vol. 3032*, (eds. M. Li, et al), Springer-Verlag, Heidelberg, 2004, 51-58.

[5] David W. Walker, Jonathan P. Giddy, Nigel P. Weatherill, Jason W. Jones, Alan Gould, David Rowse, and Michael Turner, GECEM: Grid-Enabled Computational Electromagnetics, *Proceedings of UK e-Science All Hands Meeting*, 2003.

[6] Yao Zheng, Negil. P. Weacherill and Edward A. Turner-Smith, An interactive geometry utility environment for multi-disciplinary computational engineering, International Journal for Numerical Methods in Engineering, 53(2002)1277-1299.

[7] R. W. Clough, The Finite Element Method in Plane Stress Analysis, *Proc. 2nd ASCE Conf. on Electronic Computation*, Pittsburgh, Pa, 1960

[8] M. J. Turner, E. H. Dill, H. C. Martin and R. J. Melosh, Large Deflection Analysis of Complex Structures Subjected to Heating and External loads, *J. Aerospace Sci.*, 27 (1960) 97-107

[9] Haolin Feng, Guanghua Song, Yao Zheng, Jun Xia, A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Dependent Tasks in Grid Computing, Proceedings of the Second International Workshop on Grid and Cooperative Computing (GCC2003) (Shanghai, China, 2003), *Grid and Cooperative Computing: Second International Workshop, GCC 2003, Part II, Lecture Notes in Computer Science, Vol. 3033*, (eds. M. Li, et al), Springer-Verlag, Heidelberg, 2004, 113-120