Application of Grid techniques in the CFD field

Xiaobo Yang[*] and Mark Hayes[†]

Cambridge eScience Centre, University of Cambridge
Wilberforce Road, Cambridge CB3 0WA, UK
Email: {xy216,mah1002}@cam.ac.uk

Abstract

Besides the widely used Reynolds-averaged Navier-Stokes (RANS) solver, Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) are becoming more and more practical in today's Computational Fluid Dynamics (CFD) studies. All these techniques are in constant pursuit of ever larger computing resources in order to produce acceptable and reasonable results. A new technique, the Grid technique, offers the possibility of aggregating the capabilities of geographically distributed computing resources. In this paper, a brief description of the Grid technique is presented followed by the current progress in the study of its application to CFD between the Cambridge eScience Centre and the CFD group at the Cambridge University Engineering Department.

Introduction

Grid is a technique developed in middle 1990s based on distributed computing. Although the concept of Grid itself is still under development, a definition given by the Globus group[1] reads: "The Grid refers to an infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organisations." The metacomputer could be used here to denote a networked virtual supercomputer, which is constructed dynamically from geographically distributed resources linked by high-speed networks. Such a metacomputer integrates distributed devices including supercomputers, large databases, archival storage devices, advanced visualisation devices, etc. Therefore it is suitable for large-scale computing, for instance, computational fluid dynamics (CFD) area.

Foster *et al.*[2,3] described the Grid technique in details in their publications. Foster and Kesselman[2] answered the following six basic questions and constructed the blueprint for the Grid.

- Why do we need computational grids?

- What types of applications will grids be used for?

- Who will use grids?

- How will grids be used?

- What is involved in building a grid?

- What problems must be solved to enable grid development?

---

[*]Scientific Programmer
[†]Technical Director

Virtual organisation (VO) was introduced by Foster *et al.*.[3] In a real grid environment, resources sharing is inevitable. An example is that Cambridge eScience Centre and the CFD group at the Cambridge University Engineering Department both share part of their own resources including computing facilities and research staffs to set up a VO called "Cambridge CFD" (Fig. 1). In such a VO, resourcs sharing is dynamic but with multi real organisations. VO is scalable so that it could be huge like several computing centers around the world or as small as two organisations. In a VO, usually people work for the same object such as large scale CFD study.
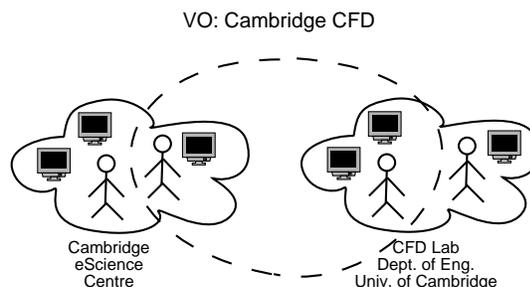

Fig. 1   VO: Cambridge CFD

For CFD, the Grid technique brings us two exciting ways solving high computing resources demanding problems. (1) Distributed supercomputing is now possible. On one hand, very large problems can be solved. On the other hand, more complex and advanced algrithms, models can now be implemented and higher spatial and temporal resolution can now be achieved. LES and DNS become more practical with such supercomputers. This gives scientests more detailed information about flow field and helps to set up fundermental CFD study in fluid dynamics, for example, mechanism of turbulence. (2) Grid resources

can also be used to solve large numbers of small tasks. By modifying one or some parameters like Mach number, angle of attach, iteration steps, CFD researchers can put all the jobs in a pool in which these jobs will be managed automatically.

In this paper, the Globus Toolkit (GT)[1,4] and Condor[5] are introduced. The Globus Toolkit makes it possible to develop grid applications while Condor is used to manage compute-intensive jobs. Condor can be used to build grid computing environment managed by the Globus Toolkit. The application of the Grid technique in CFD is followed. In this section, an introduction of the Information Power Grid (IPG) built in NASA is given. IPG is used to provide ubiquitous and uniform access to a wide range of computational, communication, data analysis, and storage resources across the United States. Current progress in the study of the application of the Grid technique in CFD between the Cambridge eScience Centre and the CFD group at the Cambridge University Engineering Department is then described with conclusions followed.

Globus Toolkit

Introduction of the Globus Toolkit

The Globus Toolkit[1,4] provides a set of services and software libraries to support Grids and Grid applications. The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. Recently a new version GT 3.0[1] has been released which is based on web service and Extensible Markup Language (XML)[6] techniques. In this paper, only the current dominating release GT 2.x will be discussed. This version is now widely used in many projects around the world including several in Cambridge eScience Centre.[7]

The Globus Toolkit can be treated as a "bag of services". As described in Fig. 2, GT 2.x is constructed by three components based on the Grid Security Infrastructure (GSI) security protocol at the connection layer.
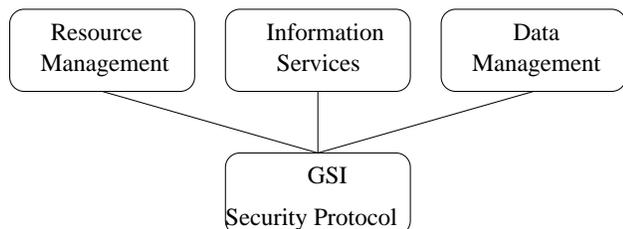
Fig. 2   Composition of GT 2.x

Security is always a top factor people consider nowadays. GT 2.x uses GSI which realises secure authentication and communication over an open network. During the development of GSI, extensions including single sign-on and delegation have been developed above exsiting public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol. Above GSI, three components have been included in the Globus Toolkit.

(1) Resource management is a high-level resource manager on top of local resource allocation services. The Globus Resource Allocation Manager (GRAM) is the key part which provides a standardised interface to all local resource management tools such as Local Sharing Facility (LSF) and Condor. Since general applications usually do not have the possibility to predict where and when resources are available, it is important that resource management involves scheduling the resources and performing preliminary work for executing jobs.

(2) The Globus Metacomputing Directory Service (MDS) is based on LDAP protocol. It provides a uniform querying system for obtaining real-time information such as CPU type, model, memory size, load, etc. of computational grids which may involve many organisations.

(3) GridFTP provides high-performance, secure, and reliable data transfer which is optimised for high-bandwidth wide-area networks (WAN). The GridFTP protocol is based on the widely used Internet file transfer protocol (FTP) with additional features implemented to improve the performance. For example, multiple data channels are designed for parallel transfer.

Grid based parallel computing

Parallel computing is today a commonly used tool for CFD applications to provide higher computing ability. MPICH-G2[8,9] is a grid-enabled implementation of the message passing interface (MPI) v1.1 standard. It enables users to couple multiple computers, may be in different architectures, to run MPI applications. MPICH-G2 can automatically convert data in messages transferred between computers of different architectures and support multiprotocol communication by automatically selecting the transmission control protocol (TCP) for intermachine messaging and possible vendor-supplied MPI for intramachine messaging.

Mahinthakumar *et al.*[10] presented a metacomputing application of multivariate, nonhierarchical statistical clustering to geographic environmental data from the 48 conterminous United States in order to produce maps of regions of ecological similarity. With the application of MPICH-G, finer mapps

were generated due to the big advantage given by the grid environment that makes greater number of computing resources. MPICH-G is a Globus-enabled device for the public domain implementation of MPI. It is the previous version of MPICH-G2. Both versions use many Globus services like job startup, security and data conversion. Difference between the two version can be found on MPICH-G2 webpage.[8] Mahinthakumar *et al.* also pointed out the communication performance is several orders of magnitude higher within a tightly coulpled parallel architecture compared to regular ethernet, and processor and communication performance can be different from one architecture to the other.

Condor

Usually in the CFD community, a High Performance Computing (HPC) environment is used to deliver a large amount of compute power over a short period of time, for example, large shared memory multiprocessor machines or dedicated linux clusters. By contrast, High-Throughout Computing (HTC) provides a computing environment that delivers large amounts of computational power over a long period of time. In this way, the computing power available is not concerned with operations per second, but operations per month or per year. We are more interested in how many jobs can finish over a long period of time rather than how quickly an individual job can finish. The key to HTC is to efficiently harness the use of all available resources. As mentioned before, Grid resources can also be used to solve large numbers of small tasks. Here the Condor project[5] is introduced.

The Condor project produces a software system that creates a HTC environment. When a job is submitted, Condor finds an available machine on the network and begins running the job on that machine. Condor has the capability to detect that a machine running a job is no longer available, for instance, when the owner of the machine starts his own work. As personal desktops and PCs becoming faster and cheaper, the use of Condor becomes more attractive on networks with large numbers of PCs and workstations. Nowadays in universities and companies, nearly everyone has his or her own personal computer. It is very convenient for user but usually such a PC is not efficiently used. These machines are normally idle when its owner is out for lunch, having a meeting, or at home. The benefit of Condor is that *"Condor takes this wasted computation time and puts it to good use"*. This makes Condor attractive for CFD people. Since CFD is always CPU tensive, it would be ideal to harness those "wasted" computing resources. For instance, a computer lab in a department of an university is usually free during night and holidays, these computers can be put in a Condor pool to run

CFD jobs. They may not provide powerful computing ability, but after one week, or even several months, obviously fruitful results will finally be achieved.

Condor is also suitable for submitting large amount of jobs at the same time. For example, under the same flow conditions angle of attack may be interesting. With many different angles of attack, these jobs can be send to a Condor pool simultaneously rather than submitting many times in usual way. Condor also provides the possibility to create checkpoint, which makes it possible for jobs migration. This is extremely important for long-time running jobs.

With the emerging of Grid technique, Condor-G[11,12] has also been developed. Condor-G can be recognised as the job manager part of Condor. Condor-G uses the Globus Toolkit to start jobs on remote machines, which allows the user to treat the Grid as an entirely local resource.

Grid and CFD
### NASA's Information Power Grid

A typical example of the application of the Grid technique in CFD is NASA's Information Power Grid (IPG).[13] The IPG is intended to link heterogeneous and geographically distributed computing resources of NASA and its IPG partners to create a scalable, adapative and transparent computational environment. In the IPG environment, extremely large applications are possible to be solved which leading to detailed understanding of the fluid dynamics. The IPG is based on the Globus Toolkit and MPICH-G. It was designed to support large-scale numerical simulations and very data- and compute-intensive visualisations for aerospace research and engineering.

Barnard *et al.*[14] reported testing the OVERFLOW code in the IPG environment on three geographically-distributed supernodes including two SGI Origin2000 systems located at NASA Ames Research Centre and one at Argonne National Laboratory. Barnard *et al.* found that the main technical challenge in implementing scientific applications on the IPG lies in accommodating the sizable and variable latencies as well as the reduced bandwidths incurred in distributed computations on geographically-separated machines. In order to deal with this problem, the OVERFLOW code which was designed for tightly-coupled parallel machines has been modified as shown in Fig. 3. In the original parallel code, at the beginning of timestep N+1 all communicating processors first exchange boundary values using values of timestep N, then the interior solution domain are updated independatently. In the modified code, boundary values are also exchanged at the beginning of timestep N+1, but these values will be used for timestep N+2. While

the boundary values for this timestep are calculated before according to data of timestep N-1. This makes the code latency tolerable by performing an extra timestep update.
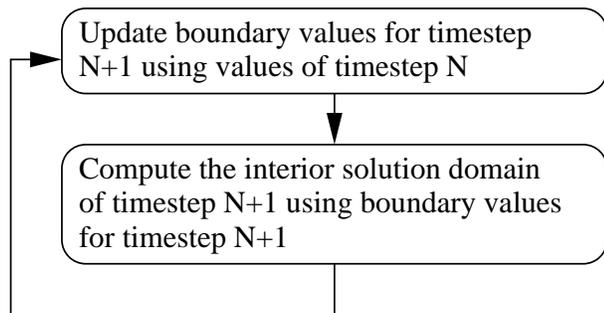
Original OVERFLOW code
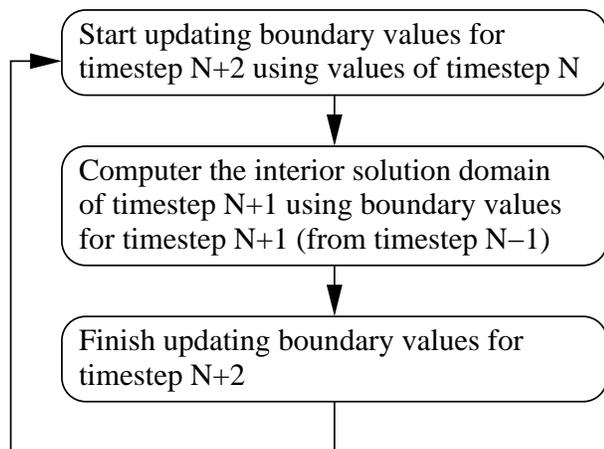
Latency tolerable OVERFLOW code

Fig. 3 OVERFLOW code modification for latency tolerance

Another factor considered is load balancing. Experiments have also been performed to improve the load balancing part in the OVERFLOW code with consideration of minimising communication and the effect of reduced bandwidth.

With above careful treatment of the two key issues in the OVERFLOW code, one of the test cases, NASA's X-38 Crew Return Vehicle (CRV), is given by Barnard et al.[14] using 2.5 million points. The flow conditions are at Mach number of 1.5 and a 15-degree angle of attack. Barnard et al. found that the latency tolerable method applied to the bounday conditions results negaligible difference in converfence in this test. With careful treatment, distributed computing was proven to be sucessful over WAN based on IPG using Globus. Detailed CFD results were reported by Meakin.[15]

Grid and CFD in Cambridge

A joint project between the Cambridge eScience Centre and the CFD group at the Cambridge University Engineering Department has been set up to develop a distributed problem-solving environment that will enable CFD studies involving turbulent and reacting flows in complex industrial geometries. In order to realise this objective, the Globus Toolkit and Condor have been set up for test. Currently, the Globus Toolkit is in everyday use at the Cambridge eScience Centre as part of the UK eScience Grid. At the same time, a web portal has been developed for user friendly access to the Grid environment.

Cambridge CFD Grid

...Mark, here I think its time to talk about the Cambridge CFD grid with your figure...

SENGA,[16] a parallel combustion DNS code, has been developed at the Cambridge CFD laboratory to study the effects of a turbulent flame kernel, in which there exists a strong coupling of turbulence, chemical kinetics and heat release. In SENGA, high accuracy numerical methods have been applied which are 10th-order explicit in space and a third-order explicit Runge Kutta method in time. Experimental work has been performed by submitting several simulations with different parameters at the same time to a Condor pool at the Cambridge CFD laboratory, which verifies the possibility of applying the Condor technique in CFD field.

Fig. 4 and Fig. 5 give an example on 3D parallel flame kernel growth in a turbulent environment on a grid of 128^3 points. A turbulent field is first numerically generated, followed with a laminar flame inserted into the turbulent field. Once the laminar flame is established, it begins to interact with the surrounding turbulence and becomes to propagate spherically outwards. The Reynolds number of this case is 30.0 and the time step is 1.0×10^{-6}. Such a small time step makes it possible to capture the chemical scale, which is below the Kolmogorov scale. 14,000 time steps have been performed in this simulation. In both Fig. 4 ($C = 0.5$) and Fig. 5 ($C = 0.8$), from (a) to (d) the kernel flame grows gradually. The difference between them is that they are at different positions. "C" is the distance between the unburned area ($C = 0$) and the fully burned area ($C = 1$). Fig. 4 ($C = 0.5$) descibes flame towards the front of the flame, while Fig. 5 ($C = 0.8$) descibes flame towards the back of the flame. Stronger flame is observed in Fig. 4 comparing with Fig. 5 which indicates the combustion process.
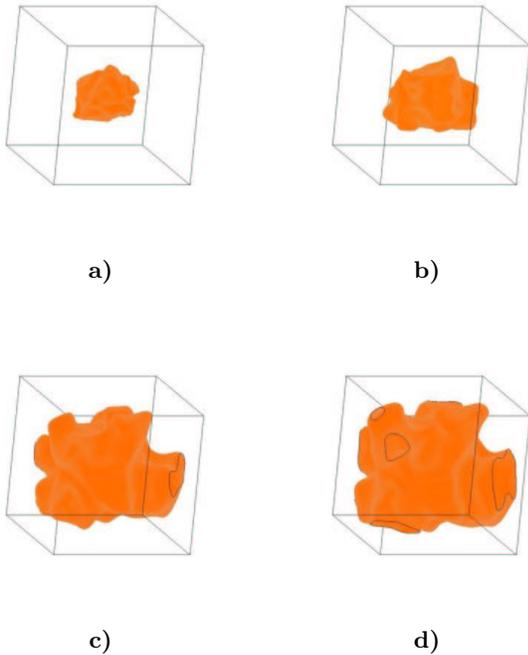
a)   b)

c)   d)

Fig. 4   **3D parallel flame kernel growth** $- C = 0.5$

a)   b)

c)   d)

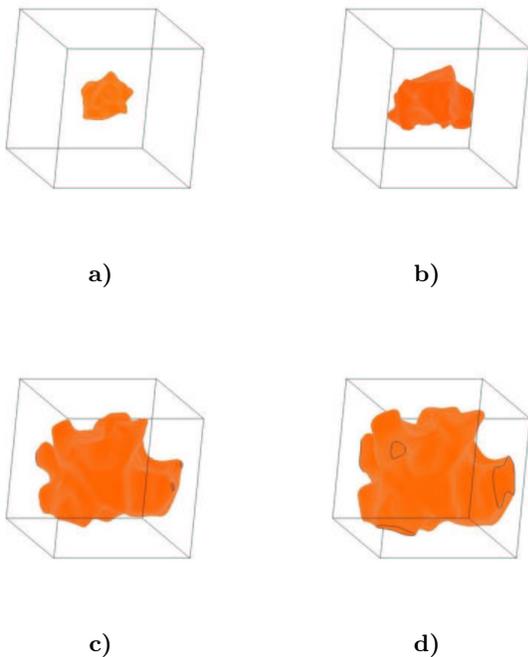**Fig. 5**   **3D parallel flame kernel growth** $- C = 0.8$

Portal Development

A web portal, Cambridge CFD Web Portal (Cam-CFDWP), has been developed for this particular project based on the Grid Portal Toolkit (Grid-port).[17] As Globus is currently based on a rather low level set of command line tools, a user friendly front end has been developed (a "portal") to guide the user through running the CFD code, SENGA. With web browser being ubiquitous nowadays, a web interface is a natural choice for portal development.

Gridport, developed at the San Diego Super-computing Centre, is a collection of technologies designed to aid in the development of science portals on computational grids: user portals, applications interfaces, and education portals. Based on Gridport 2.2, CamCFDWP has been developed which provides the following capabilities through a web browser:

- Login/logout through MyProxy delegation

- Remote job submission either interactively or in batch mode

- A batch job manager

- File transfer including upload, download and third party transfer

Fig. 6 shows the architecture of CamEMSWP. The portal web server plays a key role. First, all requests from users are sent to it. Second, all computing resources are obtained through it. Also for status validation, it should communicate with the MyProxy[18] server. The MyProxy server is introduced here for security reason. The Globus Toolkit uses user's private key each time when generating a proxy for him. With the MyProxy server, a user first creates a MyProxy credential using his private key by contacting to the MyProxy server directly, normally through a workstation with his keys stored. Such a MyProxy credential normally lasts for one week or so. Now from a web browser an end user is able to login the portal. When a user tries to login, the web portal server will get a temporal proxy from the MyProxy server, thus the status of this user will be validated.
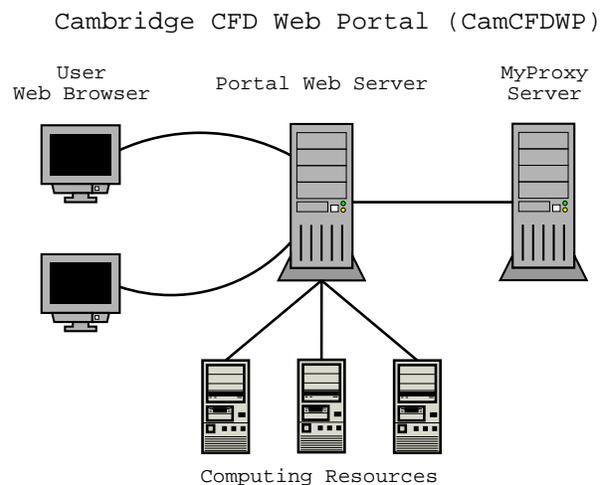
Cambridge CFD Web Portal (CamCFDWP)

Fig. 6   **Architecture of Cambridge CFD Web Portal (CamCFDWP)**

A successful login window shown in Fig.7. Once the

user logs in, it is time for him to submit jobs to remote computing resources, manage these jobs, transfer data, etc.
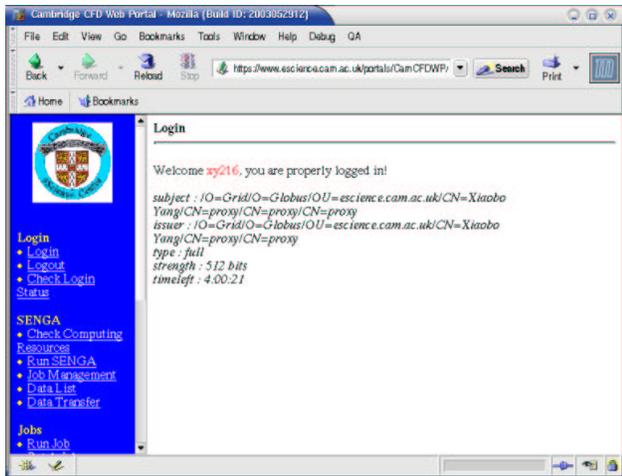

Fig. 7 Successful login Cambridge CFD Portal (CamCFDWP)

During the development of CamCFDWP, XML[6] technique has been introduced to this CFD project. Without too much modification of the legacy CFD code (SENGA) written in FORTRAN, a user inputs parameters through a web form (Fig. 8) in Cam-CFDWP. These parameters will first be saved as an XML file, which will then be validated against a schema designed for SENGA. Inside the schema, all the input parameters are described as precisely as possible so that they can be set up correctly for SENGA. Xerces-C++[19] is used in this study to validate the XML file against the schema. If the validation is successful, a plain text file with all input parameters will then be created and transferred to SENGA. Inside the XML file, extra information such as creator and date are also saved for future query.

When the numerical simulation finishes (on remote machines), all output data will be transferred to a file server. In this stage, location of these files will again be recorded in XML format. Thus for each calculation, input parameters, location of output data with some extra information, e.g., creator, date, are all recorded in XML file. A native XML database will then be introduced in order to manage these XML files. Apache Xindice[20] is currently under test in Cambridge eScience Centre. Xindice is a native XML database which has the ability to manage a lot of small XML files rather than large files. At present, Xindice uses XPath for its query language. According to our test, it has the ability to query an element in an XML database and returns elements only or whole XML files. For example, a user may be interested in querying all data created by user "abc", or all simulations done with the Reynolds number equals "30.0". Such
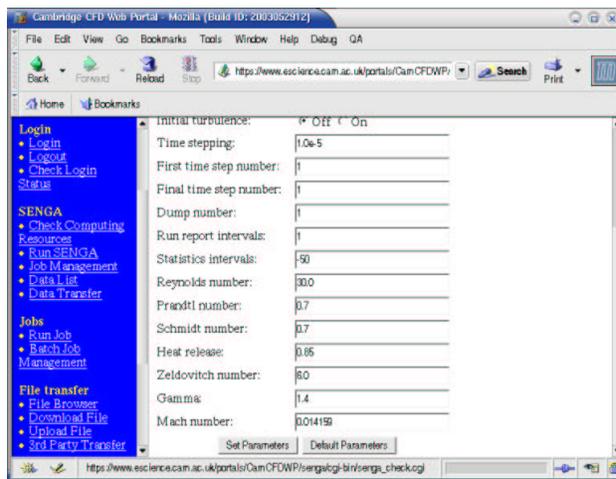

Fig. 8 Set parameters for CFD code SENGA within Cambridge CFD Web Portal (Cam-CFDWP)

information is in great interest with CFD people. The implementation of Xindice in our CamCFDWP is now under-development.

Conclusions

In this paper, a brief introduction of the Grid technique is given. With the Globus Toolkit, grid applications can be constructed which harnesses geographically distributed computing resources. CFD is an area in which the Grid technique can be applied to realise large-scale RANS/LES/DNS calculations. A description of the Cambridge CFD Grid has been given with some primary experimental results demonstrating the promising technique Grid. A web portal CamCFDWP has been developed to provide a user friendly grid manipulate environment.

References

[1]http://www.globus.org/.

[2]Foster, I. and Kesselman, C., "Computational Grids", Chapter 2 of "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufman, San Francisco, Calif, 1999.

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

[4]Foster, I. and Kesselman, C., "Globus: A Metacomputing Infrastructure Toolkit", *Int. J. Supercomputer Applications*, 11(2):115-128, 1997.

[5]http://www.cs.wisc.edu/condor/.

[6]http://www.w3.org/XML/.

[7]http://www.escience.cam.ac.uk/centre.html.

[8]http://www.hpclab.niu.edu/mpi/.

[9] Karonis, N., Toonen, B. and Foster, I., "MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface", *Journal of Parallel and Distributed Computing*, to appear 2003.

[10] Mahinthakumar, G., Hoffman, F.M., Hargrove, W.W. and Karonis, N.T., "Multivariate Geographic Clustering in a Metacomputing Environment Using Globus", *Proceedings of the ACM/IEEE SC99 Conference*, Portland, OR, 13-15 November 1999.

[11] http://www.ipg.nasa.gov/.

[12] Frey, J., Tannenbaum, T., Foster, I., Livny, M. and Tuecke, S., "Condor-G: A Computation Management Agent for Multi-Institutional Grids", *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10)*, San Francisco, California, 7-9 August 2001.

[13] http://www.ipg.nasa.gov/.

[14] Barnard, S., Biswas, R., Saini, S., Van der Wijngaart, R., Yarrow, M., Zechter, L., Foster, I. and Larsson, O., "Large-Scale Distributed Computational Fluid Dynamics on the Information Power Grid Using Globus", *Proc. of Frontiers'99*, 1999.

[15] Meakin, R., "On Adaptive Refinement and Overset Structured Grids", *13th AIAA Computational Fluid Dynamics Conf.*, AIAA-97-1858, 1997.

[16] Jenkins, K. and Cant, R.S., "Direct Numerical Simulation of Turbulent Flame Kernels", *Recent Advances in DNS and LES, eds. Knight, D. and Sakell, L.*, pp. 191-202, Kluwer Academic Publishers, New York, 1999.

[17] https://gridport.npaci.edu/.

[18] http://grid.ncsa.uiuc.edu/myproxy/.

[19] http://xml.apache.org/xerces-c/index.html.

[20] http://xml.apache.org/xindice/.