APPLICATION OF GRID COMPUTING IN AIRCRAFT DESIGN

Tian Zhimin[*,] Lin Qi[**] Yang Guangwen[*]**
*** Department of Computer Science & Technology, Tsinghua University**
**** Aviation Industry Development Research Center of China**

Keywords: *Grid Computing, Aircraft Design, CFD Workflow, Model*

Abstract

Computational Grids are motivated by the desire to provide transparent access to resources among many organizations to solve large scale scientific and engineering problems. In grids, the resource management systems provide users consistent, pervasive, inexpensive and secure access to high-end computing capabilities. Therefore, grid computing is a promising resort to integrate massive computing powers into online virtual environment so that the application will benefit from grid computing as it enables collaboration and the use of distributed computing resources. In this paper, we use a grid computing platform consisting of several heterogeneous nodes with different OS and hardware configurations. We also proposed a large scale virtual environment for the designers to aircraft design online based on this grid computing platform.

1 Introduction

An important factor to the success of portals like Google[1], Yahoo[2] and MSN[3] is that often involve highly personalized solutions or even, in the case of Google, an ad hoc infrastructure. Both IBM[4] and Google have taken the latter as starting point to create the new computing paradigm, grid computing[5]. The birth of cloud computing is very recent, but the technology's origins can be traced to the evolution of grid computing technologies and in particular, the accomplished business way from the main search engines, which were also the first to propose grid services on the market [6][7].

These search & retrieval systems first started to appear in the 1990s. Considering that grid computing emerged just a few years later, very few search engines would have been able to use it as their native technology. Indeed, had they opted for grid computing, they would probably not have met with the same success. Furthermore grid technology continued to evolve and mature in the early 2000s to become the well-established technology as it is today. For this reason the major search engines are based on a cluster-based computing paradigm. At the time this was the only suitable technology for the parallel computing. Clearly cluster and grid computing differ substantially: grid computing can be regarded to a certain extent, as the evolution of the cluster. Without going into detail, the main difference between grid computing technology and cluster computing lies basically in its being able to manage geographically distributed computational resources, supporting different hardware configurations and many versions of operating systems. Consequently, the major internet search engines also had to gradually adapt to these developments, changing in part their architecture and organization so as to fully exploit the potential of grid computing. The technology offered greater productivity and security, better QoS and an optimal system throughput. New and highly innovative services were also offered to the end customer, then included in the most important Web 2.0 services. Those companies that were able to implement grid technologies, achieved better results. This was one of the reasons that led to the disappearance of small search engines. Users decided to remain faithful to the few, main companies which, apart from providing text search services in the web pages, proposed various accessory services such as on line shopping and image searches.

The large computing and storage capacity offered by grid technology, led to the development of another service category, at the time not identifiable as belonging to a single category, for example web-mail services, web-office automation, web folders for images and so on. These services were defined as services later to the common features identification, and subsequently was the cloud computing paradigm defined as a hybrid model of exploiting the resources provided by computer networks.

On the one hand cloud computing is described as a sub set of grid computing concerned with the use of special shared computing resources. For this reason it is described as a hybrid model exploiting computer networks resources, chiefly Internet, enhancing the features of the client/server scheme. On the other hand, by delocalizing hardware and software resources cloud computing changes the way the user works as he/she has to interact with the "clouds" on-line, instead of in the traditional stand-alone mode. Gartner maintains that cloud computing is one of the 10 strategic technologies for 2008 and many companies will compete in this IT sector.

In the aviation industry grid computing model is exploited step by step. During the tenth five-year plan, we have already create grid computing platform for aviation industry that implemented share of license of expensive software and representative scientific computing, such as CFD, etc. However, we only resolve the problem of lack of license at the peak of application, regardless of priority. For example, the owner of the hardware and software resources may have preferential right. They may seize/order the resources that have being used. In this work, we will resolve the problem by

2 Grid Computing Model in Aviation Industry

A key differentiating element of a successful information technology (IT) is its ability to become a true, valuable, and economical contributor to cyberinfrastructure[8]. Grid computing embraces cyberinfrastructure and builds upon decades of research in virtualization, distributed computing, utility computing, and more recently networking, web and software services. It implies a service oriented architecture, reduced information technology overhead for the end user, greater flexibility, reduced total cost of ownership, on-demand services and so on[9][10].

2.1 General Framework

The framework of Grid Computing Platform of Aviation Industry (GCPAI) includes physical resource layer, grid middleware layer, Grid Middleware Layer and Application Service Layer. Meanwhile, the framework efficiently provides a mechanism of monitoring management and security.

When a computational grid is to be introduced into a platform for providing services, some initial operations are required before being able to offer its own resources. In the physical resource layer a virtualized environment needs to be set up in which to create the platform. One can dynamically choose percentage of the available grid resources by conserving the system's heterogeneity features. At the same time, you can create a grid instance which is comprised of some sets of compatible homogeneous subsystems. In the second layer, computational nodes management complexity is kept to grid middleware used by the grid. This level can certainly be considered middleware: in a grid context, it comes between the grid middleware and the higher application levels. If an instance is selected then the management of its virtual subsystems is entrusted to the grid middleware, which must be present even if it is chosen to manage the grid resources.

The physical resource layer integrates high performance computing infrastructure, floating license, all sorts of data that every factory and institute has provided. In the grid middleware layer, the platform may implement share of hardware based on virtualization technology. The grid middleware layer, as the core of the GCPAI, encapsulates the services such as computing resources scheduling service, data service, global floating license management service, charging service. The layer supports to create the typical application of the whole industry. In application layer, it is conceived to construct share of UG/CATIA, CFD computing, DMU(Digital Mock-up) service and so on. The application is divided into small work units, and each unit can be executed in any computational node. The framework also has a distributed file system, which stores data on the various nodes. Once the grid services supply platform has been created, it will provide access to the grid, to those factories/institutes requesting it. Every guest may decide whether to offer services or introduce its services in a general catalogue within a default grid and in the same common administrative domain. The main features of the platform include high system reliability and transparent management of information shifting. One possible configuration of the hierarchical abstraction layers involved in this case-study is shown in the Figure 1.

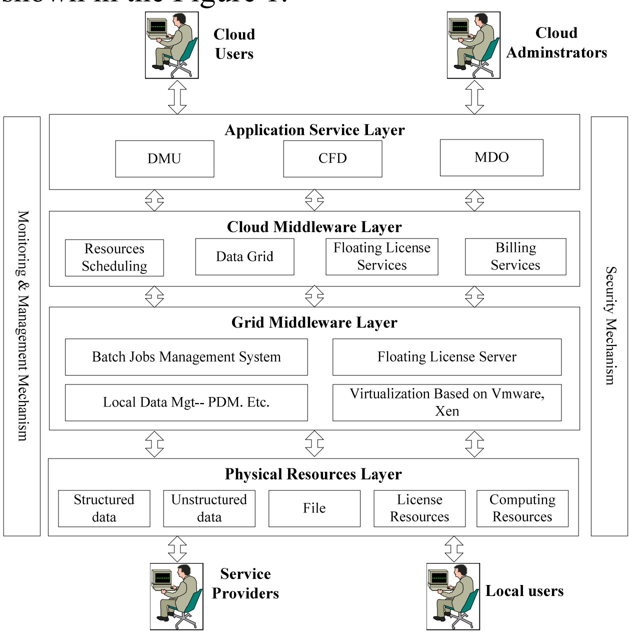

Fig. 1. The framework of Grid Computing Platform of Aviation Industry

The mechanism of monitoring management is responsible for running of the whole grid computing environment. In defense industry information security is of vital importance. Intellectual property protection is crucial among the factories/institutes. Information security strategy is proposed to ensure confidentiality of all of the resources in GCPAI. According to responsibility and right, we divide the users into four categories: local user, grid user, resources provider, and grid administrators. The local users directly use the cluster and data of the companies owned, while they don't use the resources of other companies. The grid users, as aircraft designers over intranet, use UG/CATIA floating license to design or manufacture the aircraft. The resources providers are the administrators of the factories/institutes. They are responsible for deploying their hardware and software resources on the grid. The grid administrators monitor and configure the whole grid platform.

2.2 Computing Workflow of CFD

The Computational Fluid Dynamics(CFD) in aircraft design mainly includes three processes: preprocessing, flow field solving and postprocessing. After the prototype of an aircraft is finished, designers need to harness CADfix and other geometric modeling tools to process appropriately to meet CFD's computing requirements, then using mesh generation software such as ICEM CFD, Gridgen to generate a mesh, define boundary conditions, organize computing input reference files, all the aboves are the main work of CFD pre-processing. The mesh generation job can be done in the local graphic workstation, or submit it to GCPAI when the grid node is fairly large enough, or can be done remotely by the super graphic server in the AVIC High Performance Computing Center via GCPAI. When the computing grid data file, boundary condition information file and computing reference input file are prepared and organized, the computing task can be submitted to GCPAI via CFD application portal, GCPAI will schedule the resources and decompose the tasks automatically, and construct data distribution association diagram needed by achieving task list, calculate the degree of undirected graph, then shape a realistic parallel scheduling plan, and allow those data package with higher priority can be stored and processed first in data grid, meanwhile, those data package with lower priority can be processed when system is in low load. After batch calculation task is finished, system will store the computing result in the data grid, and the system will notify the user where it stores. The calculation task can be finished in local computing resources as well, and the calculation result can be exported to data grid via CFD application portal. In the post-processing phase, user can download the computing result back to local, or using GCPAI directly to retrieve data and conduct visualization analysis work, which can be finished by the super graphic server inside AVIC High Performance Computing Center interactively.

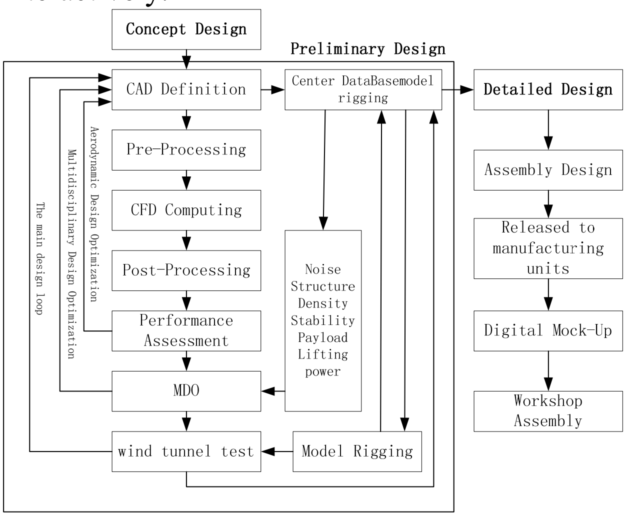

Fig. 2. The CFD Model for Aircraft Design

In the flow field solving phase, the system will break the input data as so to split a big analysis job into several fairly small jobs. After the split, each small job will have its own input and output data. Job scheduling software will submit the split jobs to the computing resource pool in the grid, and conduct each analysis separately in each node, after the job is done, the analysis result will be merged by the postprocessing, and store the result in PDM data base. The analysis process in the grid is illustrated in the following figure.

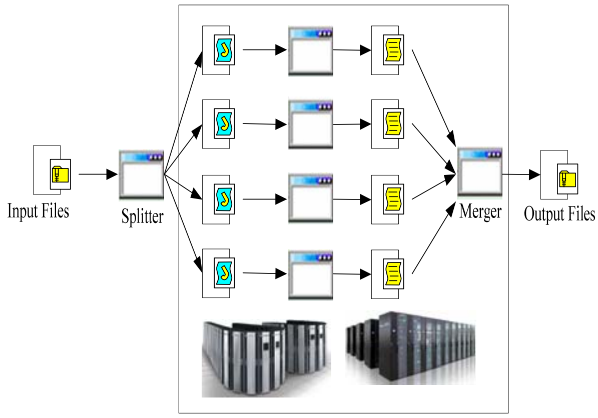

Fig. 3. The process of Flow Field Solving

2.3 Virtualization

Virtualization is another very useful concept. It allows abstraction and isolation of lower-level functionalities and underlying hardware. This enables portability of higher-level functions and sharing and/or aggregation of the physical resources.

The virtualization concept has been around in some form since 1960s. Since then, the concept has matured considerably and it has been applied to all aspects of computing – memory, storage, processors, software, networks, as well as services that IT offers. It is the combination of the growing needs and the recent advances in the IT architectures and solutions that is now bringing the virtualization to the true commodity level. Virtualization, through its economy of scale, and its ability to offer very advanced and complex IT services at a reasonable cost, is poised to become, along with wireless and highly distributed and pervasive computing devices, such as sensors and personal cell-based access devices, the driving technology behind the next wave in IT growth. Not surprisingly there are dozens of virtualization products, and a number of small and large companies that make them. Some examples in the operating systems and software applications space are VMware[11], Xen[12] - an open source Linux-based product developed by XenSource, and Microsoft virtualization products, to mention a few. Major IT players have also shown a renewed interest in the technology. Classical storage players such as EMC, NetApp, IBM and Hitachi have not been standing still either. In addition, the network virtualization market is teeming with activity. In GCPAI, we make use of vmware and Xen technology to virtualize the hardware of the factories/institutes. By the method, we provide all sorts of services to upper level.

2.4 Users

The most important Grid entity, and the principal quality driver and constraining influence is, of course, the user. The value of a solution depends very much on the view it has of its end-user requirements and user categories. Figure 1 illustrates four broad sets of user categories: grid administrators, service providers of different component services and underlying applications, technology and domain personnel that integrates basic services into composite services and their orchestrations (workflows) and delivers those to grid users, and finally users of simple and composite services[13]. User categories also include domain specific groups, such as designers, experimenter, and so on. Functional and usability requirements derive, in most part, directly from the user profiles. The user categories appropriate in the GCPAI would be expected to:

a. Support large numbers of users that range from very naive to very sophisticated (millions of researchers and designers).

b. Support construction and delivery of experimental data and design drawings for these users. For that, the system needs to provide support and tools for the grid users.

c. Generate adequate content diversity, quality, and range. This may require many workloads of services providers.

d. Be reliable and cost-effective to operate and maintain. The effort to maintain the system should be relatively small, although introduction of new paradigms and solutions may require a considerable start-up development effort.

2.4.1. Grid administrators

Grid administrators are responsible for development and maintenance of the Grid framework. They develop and integrate system hardware, storage, networks, interfaces, administration and management software, communications and scheduling algorithms, services authoring tools, workflow generation and resource access algorithms and software, and so on. They must be experts in specialized areas such as networks, computational hardware, storage, low-level middleware, operating systems imaging, and similar. In addition to innovation and development of new grid functionalities, they also are responsible for keeping the complexity of the framework away from the higher-level users through judicious abstraction, layering and middleware. One of the lessons learned from , for example, grid computing efforts is that the complexity of the underlying infrastructure and middleware can be daunting, and if exposed can impact wider adoption of a solution.

2.4.2. Service providers

Service providers are developers of services that may be used directly, or may be integrated into more complex service aggregates and workflows by service provisioning and integration experts. It incorporates: a) any base-line operating system, and if virtualization is needed for scalability, a hypervisor layer, b) any desired middleware or application that runs on that operating system, and c) any grid user access solution that is appropriate (e.g., ssh, web, etc.). The services of share of software or computing can be loaded into an operating system/application virtual environment of choice. When a grid user has the right to create a grid instance, that user usually starts with a job delivery by UG or CATIA Plugin and extends it with his/her applications. Similarly, when an service provider constructs composite services, the user extends service capabilities of GCPAI. An service provider can program an instance for sole use of one or more hardware units, if that is desired, or for sharing of the resources with other users. Scalability is achieved through a combination of multi-user service hosting, application virtualization, and both time and CPU multiplexing and load balancing.

The service providers must be component experts and must have good understanding of the needs of the grid user. Some of the functionalities that a grid framework must provide for them are service creation tools, service management tools, service brokers, service registration and discovery tools, security tools, provenance collection tools, grid component aggregations tools, resource mapping tools, license management tools, fault-tolerance and fail-over mechanisms, and so on[14].

It is important to note that the service providers, and thus the tools and interfaces must be appliances: easy-to-learn and easy-to-use and they must allow the service providers to concentrate on the service development rather than struggle with the grid infrastructure intricacies.

2.4.3. Grid Users

The grid users of services are the most important users. They require appropriately reliable and timely service delivery, easy-to-use interfaces, collaborative support, information about their services, etc.. The distribution of services, across the network and across resources, will depend on the task complexity, desired schedules and resource constraints. However, at any point in time, users' work must be secure and protected from data losses and unauthorized access. As an example the resource needs of industry grid users may range from license servers that may deliver application appropriate to the sharing software domain, , to one or more servers supporting different functions, to groups of coupled servers, e.g., an Apache server, a database server, and a workflow management server all working together to support a particular task, to research clusters, and high-performance computing clusters. The duration of resource ownership by the end-users, supporting snatch or booking model, may range from a few hours, to several weeks.

2.5 Access Control

In aircraft collaborative design based on grid computing access control mechanisms include four basic elements: the user, the role, permission, and the session. Users may get a group of access rights by activating a subset of roles in a session. So they can execute permission operations on objects related, and any non-explicitly granted permission is prohibited. On this basis, the introduction of the role of hierarchy, there may be overlap between the role of competence, belonging to different roles of users may need to perform some of the same operation. It defines a number of roles: in addition to its own authority, but also inherits the rights of other roles. Role of the level of an organization or department can reflect the responsibilities and power relations. In the grid, collaborative design also considered binding mechanisms, such as in an organization checker and designer can not be the same person (called the separation of duties). By introducing the set of constraints we consider if the various operations may be acceptable, and only acceptable action was allowed. Constraint mechanisms play a important part in access control system. It can be used to implement a higher level a powerful mechanism for organizational strategy. Constraint rules are: least privilege, mutually exclusive roles, cardinality constraints, prerequisites, time frequency constraints, such as run-time mutually exclusive. Under the constraint model and the level model to study the complex model to build security strategies and mechanisms in collaborative design.

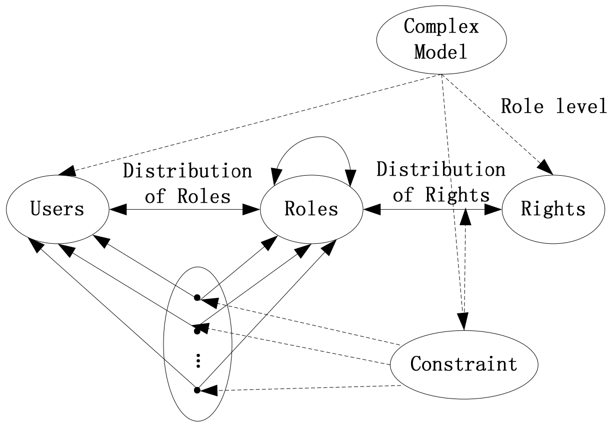

Fig. 4. Access Control Model for GCPAI

3 The Simplified Architecture of GCPAI

The aims of the GCPAI range from license share of expensive software to resources scheduling, to management and share of industry experimental data and design drawings, to run of the platform, and to mechanism of security and monitoring. The platform has efficiently aggregated the resources, including hardware and software, of CAE(Chinese Aeronautical Establishment), ACTRI(Aviation Computing Technology Research Institute), FAI(The First Aircraft Institute of AVIC), and HONGDU(HONGDU Aviation Industry Group LTD). The GCPAI takes research and manufacturing of L15 trainer aircraft for example to efficiently support the CFD, DMU and MDO(Multidisciplinary Design Optimization). The figure 5 illustrates the case.

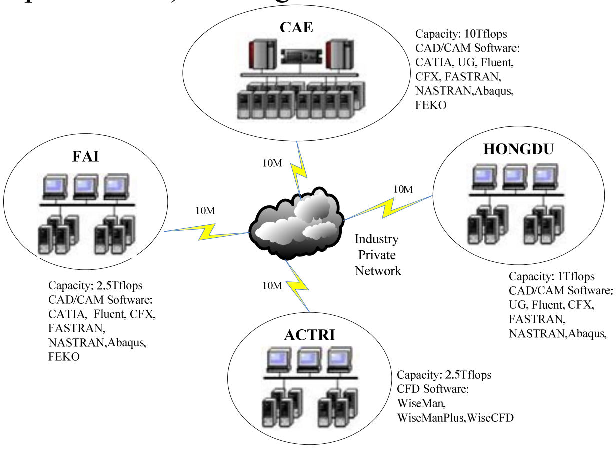

Fig. 5. The Simplified Architecture of GCPAI

4 Conclusions

Grid computing builds on decades of research in virtualization, distributed computing, utility computing, and more recently network, web and software services. It implies a service oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership, on demand services and many other things. This paper discussed the concept of grid computing, issues it tries to address, related research topics, and a grid implementation in aviation industry. Our experience technology is excellent and we are in the process of addition functionalities and features that will make it even more suitable for grid framework construction.

References

[1] Google App Engine, http://code.google.com/appengine/.

[2] HP, Intel and Yahoo Cloud Computing Test Bed, http://labs.yahoo.com/Cloud_Computing.

[3] Microsoft Cloud Computing Tools, http://msdn.microsoft.com/en-us/vstudio/cc972640.aspx

[4] IBM Blue Cloud, http://www-03.ibm.com/press/us/en/pressrelease/22613.wss.

[5] Wikipedia, "Cloud Computing," http://en.wikipedia.org/wiki/Cloud_computing, May 2009

[6] Amazon Web Services, http://aws.amazon.com,http://aws.amazon.com.

[7] M. Palankar, A. Onibokun, et al., " Amazon S3 for Science Grids: a Viable Solution," in 4 th USENIX Symposium on Networked Systems Design & Implementation (NSDI'07), 2007.

[8] Vouk, M.A. Cloud computing--Issues, research and implementations. Information Technology Interfaces, 2008. PP.31-40.

[9] A. Ramakrishnan, G. Singh, et al., Scheduling Data-Intensive Workflows onto Storage-Constrained Distributed Resources, in CCGrid 2007.

[10] H. Liu and D. Orban. Gridbatch: Cloud computing for large-scale data-intensive batch applications. In CCGRID, pages 295–305. IEEE Computer Society, 2008.

[11] VMWare, The Open Virtual Machine Format - Whitepaper for OVF Specification, v0.9, 2007, http://www.vmware.com/pdf/ovf_whitepaper_specification.pdf

[12] Virtualization Hypervisor, http://www.xen.org/.

[13] K. Lee, N.W. Paton, R. Sakellariou, and A.A.A. Fernandes. Utility Driven Adaptive Workflow Execution. In Proc. 9th CCGrid. IEEE Press, 2009.

[14] Mladen Vouk, Sam Averitt, Michael Bugaev, Andy Kurth, Aaron Peeler, Andy Rindos*, Henry Shaffer, Eric Sills, Sarah Stein, Josh Thompson "'Powered by VCL' – Using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing." Proceedings of the 2nd International Conference on Virtual Computing (ICVCI), 15-16 May, 2008, RTP, NC, pp 1-10

Copyright Statement

proceedings or as individual off-prints from the proceedings.