The Information Gathering Module of the WebCom-G Operating System

John P.Morrison, Brian Clayton, Adarsh Patil, and Sunil John

Centre for Unified Computing,
Dept. Computer Science,
National University of Ireland,
Cork,
Ireland.
e-mail: {j.morrison,b.clayton,a.patil,s.john}@cs.ucc.ie
Telephone: +353 21 4902795
Fax: +353 21 4274390

Abstract. The WebCom-G operating system will be designed to act as a sophisticated Grid middleware that will hide middleware and architectural detail from the application programmer. A central goal of WebCom-G is to provide seamless interoperability with existing core grid middlewares. This paper sketches the main components of the WebCom-G OS and presents its information gathering module in detail. This module is pivotal for the implementation of Fault Tolerance, Load Balancing, Quality of Service and hence is vital for the development of a realistic economic model.

1 Introduction

In traditional distributed computing systems, large tasks are executed by distributing sub-tasks to cooperating machines. Task dependent libraries and binaries are typically provided and managed by the user, thus making the process time consuming and error prone. Current research is focused on exploiting the computing resources of geographically distributed volunteers using the Internet and user-level middlewares such as Seti@Home and Distributed.Net, while other approaches exploit core level middleware technologies such as Globus [1] and Legion [2].

The WebCom [3] metacomputer is an application execution environment operating across the Internet or on intranets. Applications are specified as Condensed Graphs [4], in a manner which is independent of the execution architecture, and this platform independence facilitates computation in heterogeneous environments.

In addition to its expressive programming model, WebCom automatically handles task synchronisation, load balancing, fault tolerance, and task allocation without the need for these decisions to be propogated to the application developer. These characteristics, together with the ability of the CG model to mix evaluation strategies to match the characteristics of geographically dispersed

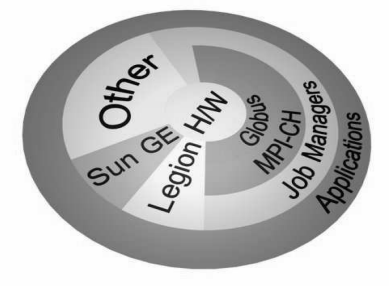

Fig. 1. An overview of the various grid middlewares

facilities and overall problem-solving environment, make WebCom a promising grid middleware candidate.

The WebCom-G Operating System is proposed as a Grid Operating System. It is modular and constructed around a WebCom kernel, offering a rich suite of features to enable the Grid. It will utilise the tested benefits of the WebCom metacomputer and will leverage existing grid technologies such as Globus and MPI. The aim of the WebCom-G OS is to hide the low level details from the programmer while providing the benefits of distributed computing.

Resources in a heterogeneous environment like the Grid join and leave at times of their choosing. Gathering information and analyzing resources is important for resource management, scheduling, load balancing and fault tolerance. To build strict models (process management models, programming models, service oriented models) for quality of service, a proper Grid Operating System is needed. As with most operating systems there will be various components to provide the required functionalty.

This paper describes how WebCom-G retrieves resource information from a heterogeneous environment and how it will use this information in conjunction with its resource management techniques to provide a home for different models which promise Quality of Service. Different methods for gathering and analyzing resource information to improve the Quality of Service requirement are discussed. A statistics analyser is introduced, which forms the basis for evaluating the costs of executing jobs on the grid. Finally, a comparison with Globus approach to static and dynamic information gathering is made.

2 WebCom-G Operating System

The WebCom-G Operating System is designed to operate between the system hardware and any installed grid middleware. It will interoperate with Globus (Versions 2.4 & 3) and with other middlewares such as PVM[8], MPI [7] and will work with standalone products such as Distributed.Net [6] and Seti@Home [5] clients.

The WebCom-G Operating System, illustrated in figure 3, is designed to be modular. This design allows WebCom-G to be used in a number of different

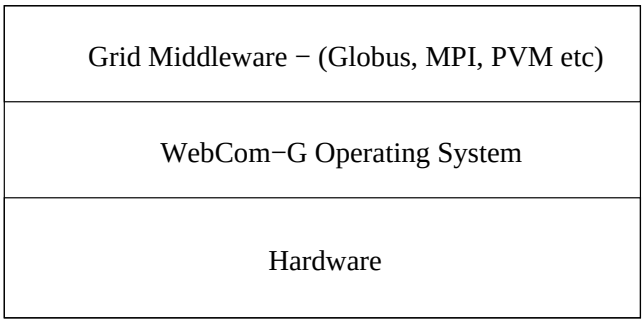

Fig. 2. WebCom-G extends the operating system functionality of the native hardware, interfacing with traditional Grid middleware.

contexts – e.g., where WebCom-G is the only grid middleware installed, or to co-exist with say Globus, or MPI or both. If the system is configured to consist of multiple middlewares, each with its own information provider service, WebCom-G will automatically choose between them based on specific requirements of the application. However, the decision can be overridden by the programmer. WebCom-G will be able to treat different middlewares and users as Vitrual Organisations, giving it control over task priorities.

3 Components of the WebCom-G OS

Economy Status Analyser
The Economy Status Analyzer is a plug-in which gets the overall resource status of the machines comprising the grid from either the stats daemon or the information manager in a WebCom-G system or from the WIS if other information gathering middlewares (such as GRIS & GIIS & Ganglia in case of Globus) are present. This module will use various algorithms to evaluate independent cost and total cost of utilization. Scheduling compute intensive, time critical and data intensive jobs depends on accurate and timely updates of resources. Based on this inforamtion, the ESA calculates the cost of executing the job. The quality of this service depends critically on the reported status information.

Stats Daemon
The Stats Daemon is a Linux daemon or Windows service, and uses standard system calls to retrieve system information - the Linux `sysinfo` kernel command or standard MFC function calls on Windows. It logs system usage to the hard drive, using one of several strategies e.g., fifo fixed filesize logging.

WebCom-G Information Module
The WebCom-G Information System (WIS) is the information gathering module within the WebCom-G OS. It consists of three parts, a low-level stats daemon to run directly on the hardware, a higher-level Information Manager, which can be aware of multiple hosts, and the WIS proper, which is capable of communicating

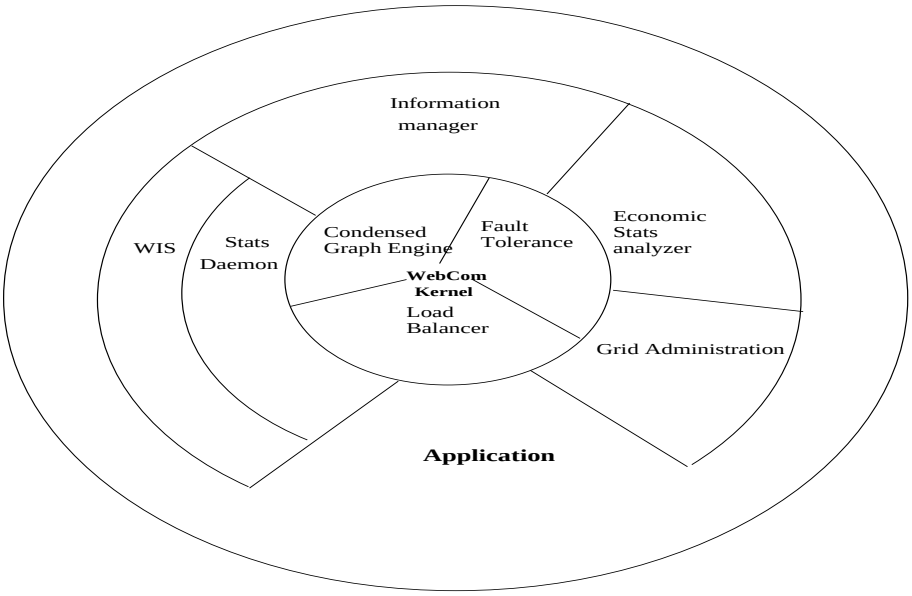

Fig. 3. WebCom-G OS architecture

with a single stats daemon, with an Information Manager or with Globus via the GIIS.

Grid Administration

The Grid Administration tool will allow administration by user or middleware. It will be possible to dynamically renice processes when unfair CPU allocation occurs, or to give priorities to particular jobs. Processes launched across the grid will be monitored and recorded, allowing the cluster manager to charge accordingly, or to prove quality of service or even to renice the processes on demand (allowing the user to purchase more CPU time or a higher priority).

4 WIS – The WebCom-G Information System

The WebCom-G Information Gathering Module consists of three components: (1) The Stats Daemon (2) The Information Manager and (3) the WIS

The Stats Daemon sits on the hardware and reports to the Information Manager. The WIS sits above the Information Manager and so the WIS can choose to communicate with either the Stats Daemon directly (if it's only talking to a single machine) or it can talk to an Information Manager (which is a promoted stats daemon and so would have information about more than one machine) or to a third party information service provider such as a Globus GIIS. (see figure 3)

The Stats Daemon The Stats Daemon is an independent entity residing on the hardware, which gathers system information. It records information on the

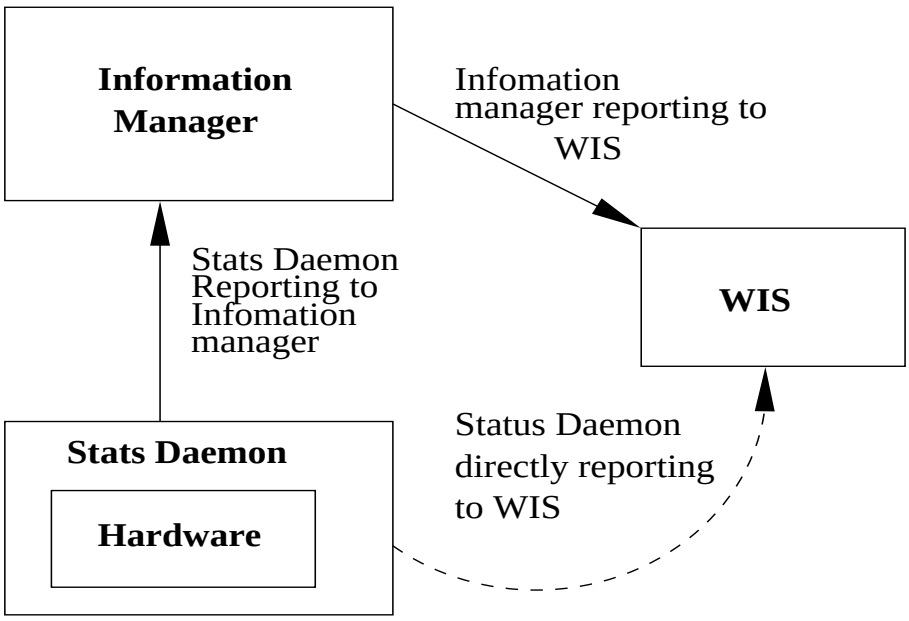

Fig. 4. Architecture of the WebCom-G Information Gathering Module

CPU load, the RAM (total+used), Pagefiles (total+used), Hard Drives (total+used), Architecture, Processes running, Users, Networking and Kernel version. The Stats Daemon records system information to log files stored on the localhost and provides them over TCP/IP in XML or raw binary format.

The Information Manager The Information Manager module provides detailed runtime and system information about a number of machines. It communicates with one or more Stats Daemons, gathering stats on each machine. It then sorts and partitions the data as required - e.g., giving an average machine load over a cluster for the Grid Administration module. It is possible to partition the information returned by middleware, person or uid and by machines or sub-clusters. This information can then be used by other WebCom-G modules, such as the Load Balancer or by the ESA. Retrieval of information at a given time period (e.g., given a time scale) is essential to determine the nature and health of connected resources and to execute jobs. The system of collection of data at fixed times may not be suitable for heterogeneous environments, which consist of dynamically changing resources. However, this method can be suitable for homogeneous nodes. It is also advantageous where the network load is high, since this model does not use any of the Globus services. Moreover, this information enables the WebCom Scheduler to execute a (less critical) job request on a more suitable resource.

WIS The WIS is the highest-level component of the WebCom-G Information Gathering Module. It will be able to query Globus to retrieve information (see

figure 4) supplied by Globus through the GIIS and Ganglia (i.e. CPU loads, memory usage etc). It will do this through one of two methods - either by building a new information provider to work with the GRIS or through the use of Commodity Grid kits (e.g., Java Cog Kit), which allow access to the information via the Globus Frame work.

The WIS provides a Java based GUI for Unix and an MFC based one for Windows.

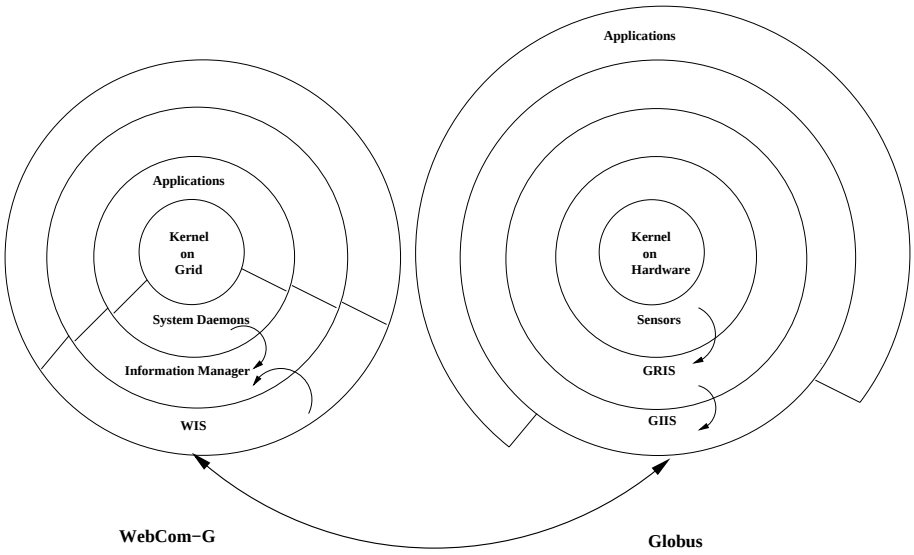

Fig. 5. WIS interopability with Globus

Each WebCom-G enabled machine will have the WebCom-G stats daemon running. The daemon will provide similar functionality to Globus, but will not rely on a single centralised stats server (GRIS, GIIS). Instead, it will use the client promotion feature of WebCom. This allows any client to be promoted to a master. Any Stats Daemon can be promoted to become an Information Manager. This Information Manager can then retrieve information about the cluster in which it is situated. This will eliminate bottlenecks and provide multiple entry points for clusters. Also the user can specify a "sub-cluster" within in a cluster and only retrieve information from and interact with those listed machines.

5 The Globus Approach

The Grid is a large distributed computing system formed by tightly or loosely coupled computers following a common set of rules for sharing, authentication and job execution thus enabling the creation of virtual organizations within the

Grid environment. The Grid is build up of many virtual organizations; resource information about individual systems building the Grid plays an important role in load balancing, resource allocation & re-allocation and execution of jobs within them.

In the case of tightly coupled computers such as Clusters of Workstations there is central information such as the Network Information Service, which stores information about all the individual nodes forming the cluster. In the case of the Grid, information is hosted through information services. Information is the most crucial part of any Grid or meta-computing system. Information about the Grid system can be acquired through information services hosted at different levels in the Grid systems.

Globus is a collection of services toolkit (reference1) used to build computational Grids. The Grid built using Globus toolkit is formed of three supporting layers (reference 2) (1) Information service (2) Data Management services (3) Management services all under one security service. Information services provide information about Grid resources using utilities such as MDS, GRIS, and GIIS.

MDS, at the top level pinnacle provides information directory services for a Grid built using the Globus toolkit. It can be queried to discover the properties of the machines, (architectures, networks, processor availability, bandwidth and disk space). GRIS is a standard information service, which runs on all resources; it interfaces with LDAP and provides information about the resources. GRIS, the core information provider for MDS provides the resource information of the local system which includes platform type and instruction set architecture, OS version and type, CPU information, Memory, Network interface information and file system summary. Each compute node on the Globus grid runs a GRIS acting as white pages. The GIIS acts as a caching service for searching. Resources register with a GIIS, which in turn publishes the information when requested by the client. MDS uses an LDAP server and (via LDAP protocol and schema) interface for querying, generating, publishing, storing, searching and displaying of such middleware information. Thus the MDS provides tools necessary to build an LDAP based information tree for computational grids. GIIS provides a level of combining individual GRIS services to provide a single system image forming aggregate index service. Thus the GIIS forms yellow pages providing collective indexing and searching function of all the computational resources available in a Grid environment and across multiple virtual organizations forming the Grid.

6 Comparing WIS and Globus

The application programmer must explicitly use the services provided by the Globus Toolkit to schedule his job. This must be done this at compile time - where the application is coded. Within the WebCom-G system, control is handed over to the WebCom-G kernel, which will automatically seek out necessary information from available services at runtime and will schedule the job accordingly. The application programmer does not need to specify how this is to be done, nor does he need to know how to do it.

Globus and WIS differ in 'lookup initiation' and flow of control (see figure 5). In Globus the application programmer specifies the services to be invoked whereas in WIS the kernel will initiate lookups and choose the service to be invoked.

Globus relies on a single server - the GIIS to coordinate information gathering. This can lead to bottlenecks and provides a single point-of-failure. The WIS approach of promoting Stats Daemons to Information Managers avoids the problem of single point-of-failures.

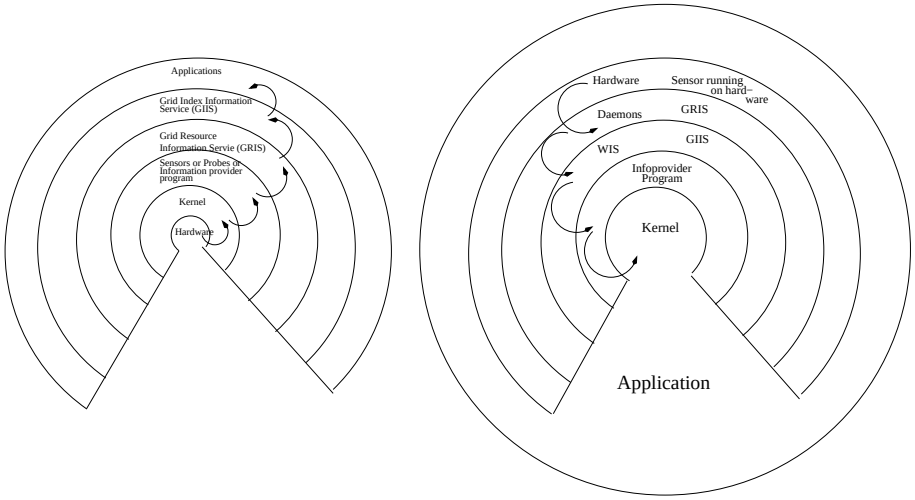

Fig. 6. Flow of control and lookup initiation in WebCom-G and Globus

7 Conclusion & Future Work

An information gathering module for the WebCom-G OS was introduced. This module provides a hierarchy of customizable information, ranging from low level machine details, to details of clusters, middlewares, users and Virtual Organizations. It contrasts with traditional information gathering systems in its flexibility and in the range of information that it provides.

Whereas the traditional mechanism for information gathering in Grid application requires embedding calls to the information service provided within the application code, thus making that code dependent on specific underlying middleware, the WebCom-G approach is to dynamically invoke the services of the information provider when scheduling tasks for execution. Since these invocation are external to the application code, the portability of the code is preserved and the complexity of the application is reduced.The information gathering module of the WebCom-G OS is specifically constructed to support this mode of operation.

The information gathering module of WebCom-G is a core component upon which other modules will be built. The quality of these modules will directly depend on the quality of information provided by the information gathering modules.

References

1. I. Foster, C. Kesselman Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputing
2. Brian S. White Michael Walker Marty Humphrey Andrew S. Grimshaw LegionFS: A Secure and Scalable File System Supporting Cross-Domain High-Performance Applications Presented at Supercomputing 2001, Denver, Colorado, November, 2001
3. John Morrison, James Kennedy and David Power, WebCom: A Web Volunteer based Computer, In The Journal of Supercomputing, 18, 47-61, 2001
4. John P. Morrison Condensed Graphs: Unifying Availability-Driven, Coercion-Driven and Control-Driven Computing
5. URL:http://setiathome.ssl.berkeley.edu/
6. URL:http://www.distributed.net
7. URL:http://www-unix.mcs.anl.gov/mpi/
8. URL:http://www.csm.ornl.gov/pvm/pvm_home.html