25845
25844

INTELLIGENT AGENT-BASED GRID COMPUTING: TOWARDS EFFECTIVE HEALTHCARE DELIVERY

Syed Zahid H. Zaidi[1], Fazilah Haron[2], Gian Chand Sodhy[2], Shailendra Singh[1], Chan Huah Yong[2]

[1]*Health Informatics Research Group*
School of Computer Science
Universiti Sains Malaysia, 11800 Penang, Malaysia.
srehan78@yahoo.com, shai_mus@hotmail.com

[2]*Grid Computing Research Group,*
School of Computer Science
Universiti Sains Malaysia, 11800 Penang, Malaysia.
{fazilah, sodhy, hychan}@cs.usm.my

Abstract

The provision of healthcare typically involves a number of individuals, located in a number of different institutions, whose decision and actions need to be coordinated if the care is to be effective and efficient. To facilitate this decision-making and to ensure the coordination process runs smoothly, the use of software support is becoming increasingly widespread. To this end, this paper proposes the notion of an Agent-based Grid Computing Infrastructure (AGCI) which is developed to help manage the healthcare processes in real world settings. The AGCI environment extends the user desktop by providing seamless access to remote computational resources (such as hardware, software, and data) and hides from user the complexity of heterogeneous, distributed, high performance back end systems.

1. Introduction

Healthcare at all levels — local, regional, national and international — is a vast open environment characterized by shared and distributed decision making and management of care requiring the communication and coordination of complex and diverse forms of information between a variety of clinical and other settings. It is the aim of healthcare software systems to operate effectively in this environment to meet information needs of patients and healthcare providers.

Practitioners in healthcare environments, in particular, require that the information is both timely and error-free, such that recommendations or decisions offered by the software systems are secure and trustworthy. On the other hand, disparate computing resources within a software system keeps disciplines stratified, so healthcare researchers often end up wasting time by replicating work and this often results in wastage of resource utilization.

Healthcare services have been using information technologies with the objective to ease and automate a particular job or task. Thus there have been many systems developed for healthcare services to achieve the targeted objective. In addition some of these systems are built to run on a particular operating system only. The problem will only start to arise when there is a need for these heterogeneous systems to communicate and pass information among them.

The solutions that have been thought to solve the heterogeneous systems are 'distributed systems' [1]. Unfortunately, often most of the distributed system solution available is a tightly integrated collection of technologies designed to facilitate distributed computing. Thus, mostly, this form of solution is very much focused for a particular type of technology only. For example, it may only be able to integrate Linux to Windows platforms. In addition to that, most of the distributed systems are in close physical proximity to one another. Thus, to achieve full integration of heterogeneous systems we need to use the grid solution [2]. The main reason for using grid solution is because grid is efficient and optimally utilizes a wide range of heterogeneous, loosely coupled resources in an organization tied to sophisticated workload management capabilities or information virtualization. Moreover, it helps two machines containing different healthcare services and also running on different platforms to communicate and maximize the usage of both resources which are available currently. These resources can be in the form of computing power, hard drive, memory space etc.

2. Related Work

A diversified number of methodologies have been proposed for designing grid- and agent-based healthcare systems. The development of these design tools range from commercially available products to research prototypes. We briefly focus first on a few available, advanced and widely accepted systems and then compare them with our proposed AGCI infrastructure.

2.1. Medicine & Biology Related Grid Systems

CROSS GRID [3] is a prototype system for pre-treatment planning in vascular intervention and surgical procedures. DATAGRID [4] is a testbed that provides the right platform for data mining algorithms, databases, code management, graphical interface tools and

0-7803-8482-2/04/$20.00

facilitate sharing of genomic and medical imaging databases. NEUROGRID [5] provides large-scale processing resources to efficiently diagnose and analyze brain functions. BIRN [6] is another testbed to address biomedical researchers' needs to access and analyze data at a variety of levels of aggregation located at diverse sites throughout the country. EUROGRID (BIOGRID) [7] allows knowledge discovery and access to multiple types of unstructured data, effectively visualized and accessible in a structured data model. GEMSS (Grid Enabled Medical Simulation Services) [8] provides medical practitioners with access to advanced simulation and image processing services for improved pre-operative planning and real-time surgical support for specific time-critical applications. MAMMOGRID [9] is designed to investigate a set of important healthcare applications as well as the potential of the grid to support effective co-working between healthcare professionals throughout Europe. HEALTHGRID [10] gathers grid-related projects with the goals of acquiring and sharing experiences in deploying biomedical applications using existing middlewares as well as promoting the grid concept in the biomedical community. NC BIOGRID [11] provides access to, and coordination of, the computing, data storage, and networking infrastructure required for researchers to take full advantage of the genomics revolution.

We notice that most of the current research on grid-based healthcare revolves around gene and brain functions, surgical procedures and supports, and data access, sharing and analysis. None of the systems provide strategic healthcare services pertinent towards organization efficacy — allowing healthcare practitioners to make better decisions. This services may include: analysis, planning, trending, examination, forecasting, prediction, bench marking and best practices reporting, outcome measurements, what–if scenario analysis, comparisons, organization practices with organization rules, market research, effectiveness on outcomes of treatment, data analysis for organization financing, health surveillance and resource allocations.

2.2. Grid and Agent-based Grid Systems.

Globus [12] focuses on defining a toolkit of low-level services for security, communication, resource location, resource allocation, process management, and data access on the grid. These basic services are then used to implement higher-level services, tools, and programming models. ARMS [13] utilizes performance prediction techniques to provide quantitative data regarding the performance of complex applications running on a local grid resource. At the meta-level, a hierarchy of homogeneous agents is used to provide a scalable and adaptable abstraction of the system architecture. Control of Agent-Based Systems (CoABS) is an open agent grid architecture populated with scalable, deployable, industrial strength agent grid components [14].

The main advantages of agent-based grid computing are reducing user's burden, helpful for novice users (non IT professionals such as doctor, manager, etc.), suitable for distributed environment especially in sharing load, tasks and managing the links, able to learn and make better decisions by itself, helpful for retrieving and monitoring data from online data repositories (such as websites), and also during processing as information can be checked at each node, providing reliable discovery of knowledge.

Formally speaking, a software agent is a software package that carries out tasks for others (human users, processes, workflows or applications) autonomously without being controlled by its user once the tasks have been delegated. From an architectural point of view, the strong advantage of agents comes from its decentralized, distributed and light architecture. An agent is just a small independent program, small piece of code sometimes no more than a Java class of 10 lines rather than millions of lines. This view of small programs working together looks attractive for scheduling bigger processes, for collaborative work, etc. The architecture of an agent paradigm is basically to make available some resources that can be used by the agent.

2.3. Multiagent System Design and Methodology

A number of methods have been proposed for the modeling of intelligent agents in a distributed and heterogeneous environment. The best and widely used approach is to model agents based on BDI (believe, desires and intention) [15]. This approach looks the problem in two perspectives, the external and internal views. The external view breaks the problem into two main components: the agents themselves (agent model) and their collaboration/communication model (Knowledge Query Manipulation Language, or KQML). The internal viewpoint uses three models for each BDI agent class: an agent model for defining relationships between agents, goal model for describing goals and planning and scheduling models to achieve agent goal. In any distributed environment, the agents can be classified into particular roles according to their capability description. Agents may have persistent roles (long term assignment as well as task-specific roles) or short term assignments. This way, we can group the multi agents- based organization into two main models: the agents/roles model (agents' capability and behaviour) and the agents/roles interaction model (KQML). Moreover, the roles can be arranged in class hierarchy and the responsibilities are then assigned to each role along with services to meet those roles. Finally, agents' interaction can be defined down to the level of individual functions and to their associated data.

3. AGCI Methodology — An Overview

We propose the notion of an Agent-based Grid Computing Infrastructure (AGCI), developed to help manage the healthcare processes in real-world settings. Grid computing enables seamless integration of computing systems and clusters, data storage, specialized networks, sophisticated analysis and visualization tools, and steady and reliable source of computing power. But, effective load balancing, task allocation and resource management still remain as some of the challenges. The notion of intelligent agents technology in grid computing has been introduced to address such issues ranging from task allocation, networks and resource management, data repository access and retrieval to result generation for end users.

The design and methodology of AGCI framework is based on most advanced and widely accepted grid frameworks, specifically on Globus [12] and CoABS [14] systems. AGCI uses Globus components concept — resource management, information services, data management — deploying three main types of agents. To realize these three components, the technologies these agents use include Grid Resource Allocation Management (GRAM), Monitoring and Discovery Service (MDS), and Grid File Transfer Protocol (GridFTP). All of these components utilize the Grid Security Infrastructure (GSI) protocol for security at the connection layer by providing single sign-on, authorization and authentication, job submission, resource monitoring, searching and allocation, data movement, and a set of tools for application programming interfaces (APIs).

For application and functional development, AGCI uses CoABS concept which provides both appropriate programming models and a range of services. At the application level, the agent grid is defined as an enabling technology needed for command and control. From a functional point of view, the AGCI grid application-level characteristics suggest that the agent grid knows not only about agents, but also about their computational requirements (e.g., internal behaviors). Hence, the AGCI provides a unified heterogeneous distributed computing environment in which computing resources are seamlessly linked.

4. AGCI Framework

We present the architectural and functional overview of our proposed multi Agent-based Grid Computing Info-Structure (AGCI) for the generation of data-mediated, diagnostic support and strategic healthcare services, as depicted in figure 1. AGCI features a federation of intelligent agents, each one providing a particular service to either other intelligent agents or end-users. The agent-federation is designed to service four functional components: (i) end-user interface; (ii) resources and data management; (iii) link remote data access network; and (iv) information and diagnostic-support strategic healthcare services. Hence it comprises four types of autonomous intelligent agents which are described next.

4.1. Interface Agents

The Interface Agent (IA) has the responsibility of garnering detailed specification and requirement of a data-mediated healthcare service from the user through web-based interface. The specification of the services is very flexible to allow the user to tweak the smallest detail of the service specification. The activation of an interface agent, based on timestamp, dynamically forms an organizational structure, where finite number of agents are available on demand to coordinate with each other. More specifically, the IA takes as input a set of service goals and autonomously translates them to three follow-up task specifications: (i) the tasks that need to be performed; (ii) the data that need to be retrieved; and (iii) the presentation style of the results (i.e. the service). This is achieved via the following modules:

- *Service Specification Module* consists of a list of pre-designed data-mediated services. Each service script guides the user to specify the parameters of the problem.
- *Task Definition Module* defines the follow-up tasks based on the service specification.
- *Communication Module* ensures communication with other agents mainly for assigning the processing tasks to the respective agents.

4.2. Data Collection Agents

The core functionality of the Data Collection Agent (DCA) is to facilitate the on-demand retrieval of 'relevant' data from the multiple healthcare data repositories. DCA is activated by IA which passes a task-specification to DCA. In an autonomous manner, DCA performs the following tasks: (a) establishing protocols for remote data access; (b) data selection and retrieval; and (c) data synthesis. The entire operational functionality of DCA is realized via the following inherent modules:

- *Data Collection Module* contains routines for translating an agent's request to database queries with respect to the data model of the available data repositories.
- *Aggregation Module* defines constraints for heterogeneous data aggregation.
- *Pre-processing Module* processes raw data to get quality data via a variety of pre-processing techniques.
- *Communication Module* is responsible for communication with other agents for co-ordination purposes.

4.3. Information Services Agent

The main task of this agent is two-fold. First, it collates all the results from the different agents, sets up the result-visualization algorithms to produce different perspectives of the results and package the results into turn-key decision-support/strategic service. Secondly, it provides support for collecting information in the grid and for querying this information. This is achieved by deploying the following components:

- *Grid Resource Information Service (GRIS)* is the repository of local resource information derived from information providers. GRIS is able to register its information with a GIIS, but GRIS itself does not receive registration requests. The local information maintained by GRIS is updated when requested, and cached for a period of time known as the time-to-live (TTL). If no request for the information is received by GRIS, the information will time out and be deleted. If a later request for the information is received, GRIS will call the relevant information provider(s) to retrieve the latest information.
- *Grid Index Information Service (GIIS)* is the repository that contains indexes of resource information registered by the GRIS and other GIISs. It can be seen as a grid-wide information server. GIIS has a hierarchical mechanism, like DNS, and each GIIS has its own name. This means client users can specify the name of a GIIS node to search for information.
- *Resource and Information Provider.* The information providers translate the properties and status of local resources to the format defined in the schema and pass it to GRIS. GRIS registers its local information with the GIIS, which also registers with another GIIS, and so on. MDS clients can get the resource information directly from GRIS (for local resources) and/or a GIIS (for grid-wide resources).

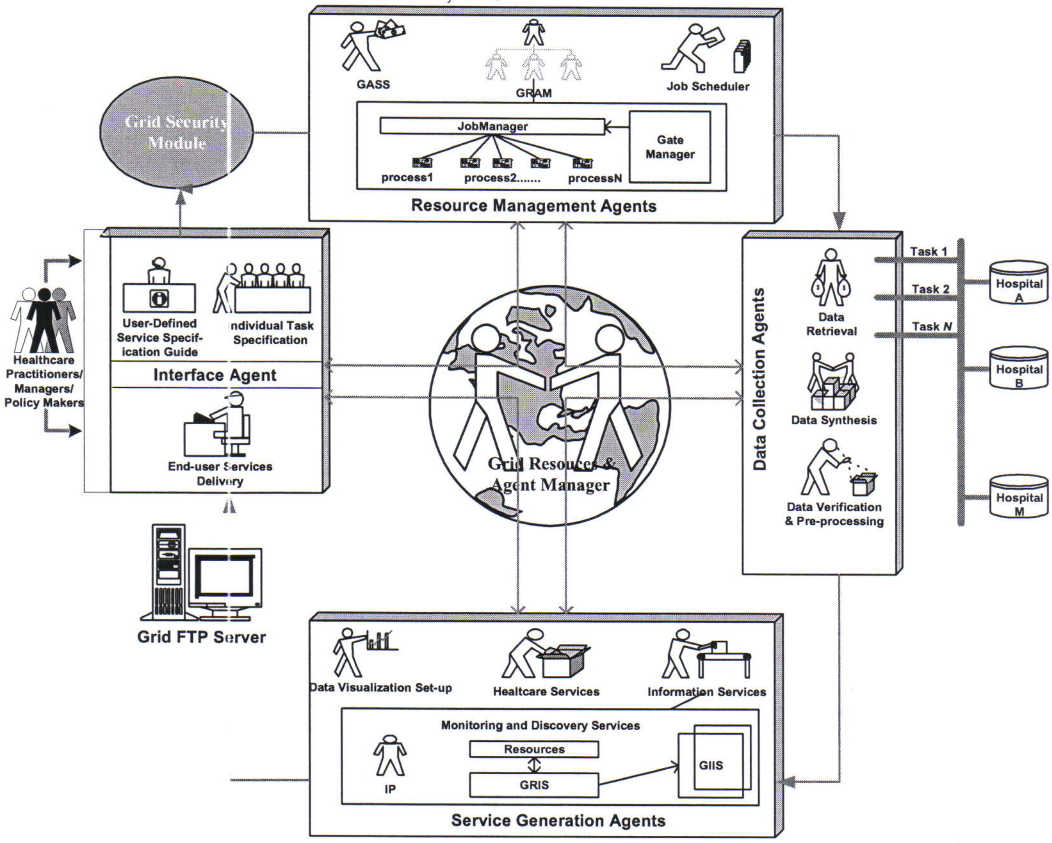

Figure 1: A sketch of an agent community (i.e. AGCI) which deals with grid computing tasks in scenarios to provide effective healthcare delivery.

4.4. Resource Management Agent

The resource management agent provides support for resource allocation, jobs submission, remotely running executable files and receiving results, and managing job status and progress. Our proposed AGCI system offers its own job scheduler, in contrast to Globus system, to find available resources and automatically send jobs to suitable machines. The main building blocks of this agent are discussed as follows:

- *GRAM/GASS.* The primary components of the resource management agents are the Grid Resource Allocation Manager (GRAM) and the Global Access to Secondary Storage (GASS). GRAM is the module that provides the remote execution and status management of the execution. GRAM uses GASS for providing the mechanism to transfer the output file from servers to clients.
- *GateManager and Job Manager.* When a job is submitted by a client, the request is sent to the remote host and handled by the gatemanager located in the remote host. Then the gatemanager creates a job manager to start and monitor the job. When the job is finished, the job manager sends the status information back to the client and terminates. The gatemanager is responsible for secure communication between clients and servers. Among the functions of a job manager are parsing the resource language, breaking down the RSL (resource scripting language) scripts, and allocating job requests to local job scheduler for execution.

4.5. Grid and Agent Manager

The services provided by Grid Manager include security (i.e. authentication, authorization, protection), process and data concepts (i.e. memory, files, databases, shared address space, communication mechanisms, control mechanism) and resource management (i.e. acquisition, allocation, scheduling). Thus, the main building blocks of grid manager include:

- Access Framework — grid access mechanism for message handling and Agent Communication Language (ACL) translation;
- Directory — white and yellow pages manager;
- Logging — message log manager;
- Visualization — grid activity and status manager;
- Brokerage — recruitment and mediation manager;
- Translation — ACL translation to/from KQML and Foundation for Intelligent Physical Agents (FIPA) ACL.

The main functions of Agent Manager are to control coordinating/collaborating/negotiating facilities among the agents, including process management and discovery constraints, besides maintaining the communication protocols of the working system. In the KQML infrastructure or agents paradigm, our agent manager uses match-making services mechanism — where each agent has to register itself to the agent manager and provide services to recognize each other during problem solving activities.

4.6. Grid FTP Server

It provides support to transfer files among machines in the grid and for managing these transfers. This module is also called data management module. GridFTP is a key component for secure and high-performance data transfer, thus, the word GridFTP can be referred to as a protocol, a server, or a set of tools.

4.7. Grid Security Module

All of AGCI agents and deployed components utilize Grid Security Infrastructure (GSI) protocol for security at the connection layer by providing single sign-on, authorization, authentication, job submission, resource allocation, monitoring and searching, data movement, and set of tools for application programming interfaces (APIs).

5. Work Flow of Our Proposed Framework — A Scenario

We will discuss a scenario in order to identify some of the relevant issues pertaining to the functional incorporation of intelligent agents in a grid framework. This scenario pictures a regional healthcare manager/administrator dealing with an infectious-disease epidemic and needs to: (a) forecast the possible geographical impact of the epidemic; (b) analyze hospital admissions due to the epidemic; and (c) plan an effective antibiotic regime (with regards to the sensitivity of different antibiotics to the infectious disease) to treat the infectious bacteria. Considering that a region consists of multiple hospitals, each with its own data repository that is remotely accessible via a private network, the tasks anticipated are as follows:

- The user states complex problems/computations and allocates all resources needed to solve them through web-based graphical user interface.
- Formation of agent organization.
- Agents start communicating with each other over distributed network to request and provide information, find information sources to pre-process, integrate information, and negotiate to remove conflicts in different tasks. All agents work in parallel and share their information through the grid and agent manager modules.
- Establishing a communication channel to enable remote access to data repositories of multiple hospitals. At the technical level, when using an ACL, agents transport messages over the network using a low-level protocols such as SMTP, TCP/IP, HTTP, etc.
- Collection of 'relevant' data to complete each individual task needed. This involves identifying and subsequently retrieving data from the respective data repositories.
- Synthesis of heterogeneous data originating from multiple data repositories.
- Preparation of the data according to the specification of the service packages.
- Execution of the data processing algorithms.
- Generation of healthcare report for the end-user.

In this scenario, it may be noted that the tasks outlined above correspond well with the functional capabilities of intelligent agents, hence there is a case for an agent-based solution for healthcare in a grid environment.

6. Conclusion

In our work, we have managed to leverage intelligent agents for generating data-mediated decision-support or strategic-planning services targeted for healthcare professionals and managers. Although at a preliminary stage, the AGCI infrastructure developed thus far allows us to explore the ways functional collaboration may occur using agents to realize a distributed grid info-structure. We believe that the emerging intelligent agent-based grid framework applied to the healthcare domain can provide interesting opportunities to operationalize the volumes of healthcare data routinely collected within numerous healthcare enterprises. Thus, our proposed AGCI framework is not meant to provide a uniform agent architecture that all components must adhere to, but rather a bridge between agent (and other component) architectures, allowing interoperability across these architectures but not replacing all of the services provided by these infrastructures. We see that grid computing has a big role to play in healthcare services. Grid does not only ease on the communication part but also on the resources utilization. In short the extensive computational power usage which we will get from grid can make the system to work even faster and more efficient.

7. References

[1] Zsolt Nemeth, Mauro Migliardi, Dawid Kurzyniec, and Vaidy Sunderam, "A Comparative Analysis of PVM/MPI and Computational Grids", *Proceedings of the 8th European PVM/MPI Users' Group Meeting*, Santorini/Thera, Greece, September 23- 26 2003, pp. 14-15.

[2] Craig Lee, Cheryl DeMatteis, James Stepanek and Johnson Wang, "Cluster Performance and the Implications for Distributed, Heterogeneous Grid Performance", *Proceedings of the 9th Heterogeneous Computing Workshop*, Cancun, Mexico, May 2000, pp. 253-261.

[3] Koen Holtman, "CMS Data Grid System Overview and Requirements", *Proceedings of Computing In High Energy And Nuclear Physics*, Beijing, September 3–7 2001, pp. 754-757.

[4] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets", *Journal of Network and Computer Applications*, Volume 23, 2001, pp. 187-200.

[5] S. Joseph, "NeuroGrid: Semantically routing queries in peer-to-peer networks", *Proceedings of the International Workshop on Peer-to-Peer Computing (co-located with Networking 2002)*, Pisa, Italy, May 2002.

[6] Mark H. Ellisman, "Biomedical Informatics Research Network to Improve Understanding of Brain Disorders", *Envision Magazine*, October–December 2001.

[7] Piotr Bala, Victor Alessandrini, Denis Girou, Jon MacLaren, John Brooke, Jaroslaw Pytli´nski, Miroslaw Nazaruk, Dietmar Erwin, Daniel Mallmann and Jan-Frode Myklebust, "BioGRID - An European grid for molecular biology", *11th IEEE International Symposium on High Performance Distributed Computing*, Edinburgh, Scotland, UK, 23-26 July 2002, pp. 412.

[8] Jochen Fingberg, "Numerical Simulation for eHealth: Grid-enabled Medical Simulation Services (GEMSS)", *Parallel Computing*, Dresden, Germany, 2-5 September 2003.

[9] Jérôme Declerck, "eDiamond and Mammogrid to develop grid infrastructure for standardized annotated mammograms", *The First European HealthGrid Conference*, 16 January 2003.

[10] Vincent Breton, "Health Grid", *International Symposium on Grid Computing*, Sinica, Taipei, Taiwan, March 8-11, 2003.

[11] J. Charles Kesler, "NC BioGrid Testbed Accomplishments", *MCNC-GCNS Grid Forum*, October 2003.

[12] Ian Foster, Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", *The International Journal of Supercomputer Applications and High Performance Computing*, Volume 11, Number 2, 1997, pp. 115-128.

[13] Junwei Cao, Stephen A. Jarvis and Subhash Saini, "ARMS: An agent-based resource management system for grid computing", *Scientific Programming*, Volume 10, Number 2, 2002, pp. 135-148.

[14] Niranjan Suri, Jeffrey M. Bradshaw, Maggie R Breedy, Paul T. Groth, Gregory A. Hill, Renia Jeffers, and Timothy S. Mitrovich, "An Overview of the NOMADS Mobile Agent System", *Proceedings of European Conference on Object-Oriented Programming*, Sophia Antipolis, France, June 13 2000.

[15] Moises Lejter and Thomas Dean, "A Framework For The Development Of Multiagents Architecture computing", *IEEE*, December Edition 1996, pp.47.