Publication proof

Cloud Computing in Bioinformatics

Javier Bajo, Carolina Zato, Fernando de la Prieta, Ana de Luis, and Dante Tapia

Abstract. Cloud Computing presents a new approach to allow the development of dynamic, distributed and highly scalable software. For this purpose, Cloud Computing offers services, software and computing infrastructure independently through the network. To achieve a system that supports these characteristics, Service-Oriented Architectures (SOA) and agent frameworks exist which provide tools for developing distributed and multi-agent systems that can be used for the establishment of Cloud Computing environments. This paper presents a CISM@ (Cloud computing Integrated into Service-oriented Multi-Agent) architecture set on top of the platforms and frameworks by adding new layers for integrating a SOA and Cloud Computing approach and facilitating the distribution and management of functionalities. CISM@ has been applied to the real case study consisting of the analysis of microarray data and has allowed the efficient management of the allocation of resources to the different system agents.

Keywords: Cloud Computing, SOA, Bioinformatics, Microarray, Multi-Agent Architecture.

1 Introduction

One of the recent developments in terms of Web Architectures referred to is denoted by the term Cloud Computing [17]. This new architectural concept offers different advantages with respect to preceding architectures as it has the capacity to offer the same level of traditional services, from complete software packages to infrastructural hardware. They are dynamic, distributed and scalable systems that provide different services on demand. These types of software usually require the creation of increasingly complex and flexible applications, so there is a trend toward reusing resources and sharing compatible platforms or architectures. In some cases, applications require similar functionalities already implemented into other systems, which are not always compatible. Microarray Data Analysis [0][1] consists of the expression study of different levels of expression in different genes. For this, statistical techniques which are widely used in various fields are carried out so that the functionality is largely reusable.

Javier Bajo, Carolina Zato, Fernando de la Prieta, Ana de Luis, and Dante Tapia
Departamento de Informática y Automática, Universidad de Salamanca,
Plaza de la Merced, s/n, 37008, Salamanca, Spain
e-mail: {jbajope,carol_zato,fer,adeluis,dantetapia}@usal.es

A.P. de Leon F. de Carvalho et al. (Eds.): Distrib. Computing & Artif. Intell., AISC 79, pp. 147–155.
springerlink.com

It is necessary to develop innovative solutions that integrate different approaches in order to create flexible and adaptable systems, especially for achieving higher levels of reutilization of developed algorithms with independence from the architecture used. It is therefore necessary to develop new functional architectures capable of providing adaptable and compatible frameworks and allowing access to services and applications regardless of time and location restrictions. There are Service-Oriented Architectures (SOA) and agent frameworks [3] [4] [5], which provide tools for developing distributed systems and multi-agent systems [6] [7] [8] that can be used for the establishment of cloud computing environments. However, these tools do not solve the development requirements of these systems by themselves.

The main purpose of this research is to design CISM@ (Cloud Computing Integrated on Service-oriented Multi-Agent) architecture with several features capable of being executed in dynamic and distributed environments to provide interoperability in a standard framework. These features can be implemented in devices with limited storage and processing capabilities. CISM@ architecture is set on top of the platforms and frameworks by adding new layers for integrating a SOA and Cloud Computing approach and facilitating the distribution and management of functionalities. A distributed agent-based architecture provides more flexible ways to move functions to where actions are needed. Additionally, the programming effort is reduced because it is only necessary to specify global objectives so that agents cooperate in solving problems and reaching specific goals, thus giving the systems the ability to generate knowledge and experience.

CISM@ has been applied in the analysis of microarray data. CISM@ integrates intelligent agents with a service-oriented philosophy that allows analysis of microarray data through the integration of different services in CISM@ and for the expression analysis of different genetic characteristics. An expression analysis consists of three stages: normalization and filtering; clustering and classification; and extraction of knowledge. The context that provides a Cloud Computing-based architecture is ideal for the treatment and analysis of bioinformatics data. It allows the exchange of great quantities of data, covers the required computational needs at the execution time of different algorithms on the aforementioned data and provides an adequate software environment for the display and study of the results obtained.

In the next section, the problem of microarray analysis is briefly explained. The specific characteristics and the agent-based architecture will be described in section 3. Finally, section 4 will present the results and the conclusions obtained.

2 Microarray Analysis

The use of microarrays, and more specifically expression arrays, enables the analysis of different sequences of oligonucleotides [0][1][0]. Microarrays have become an essential tool in genomic research, making it possible to investigate global gene expression in all aspects of human disease. [8]. Microarray technology is based on a database of gene fragments called ESTs (Expressed Sequence Tags), which are used to measure target abundance using the scanned fluorescence intensities from

tagged molecules hybridized to ESTs [10]. Specifically, the HG U133 plus 2.0 [9] are chips used for expression analysis. These chips analyze the expression level of over 47,000 transcripts and variants, including 38,500 well-characterized human genes. It is comprised of more than 54,000 probe sets and 1,300,000 distinct oligonucleotide features. The HG U133 plus 2.0 provides multiple, independent measurements for each transcript.

Simply put a microarray is an array of probes that contains genetic material with a predetermined sequence. These sequences are hybridized with the genetic material of patients, thus allowing the detection of genetic mutations through the analysis of the presence or absence of certain sequences of genetic material. The analysis of expression arrays is called expression analysis. An expression analysis basically consists of three stages: normalization and filtering; clustering and classification; and extraction of knowledge. These stages are carried out from the luminescence values found in the probes.

3 CISM@ Architecture

CISM@ is a novel architecture which integrates a Cloud Computing approach with SOA and intelligent agents for building a system that needs be dynamic, flexible, robust, adaptable to changes in context, scalable and easy to use and maintain. The architecture proposes a new and easier method to develop distributed intelligent systems, where cloud services can communicate in a distributed way with intelligent agents, even from mobile devices, independent of time and location restrictions. The functionalities of the systems are not integrated into the structure of the agents; they are modeled as distributed services and applications that are invoked by the agents acting as controllers and coordinators. Another important functionality is that, thanks to the agents' capabilities, the systems developed can make use of reasoning mechanisms to handle cloud services according to context characteristics, which can change dynamically over time.

CISM@ is based on agents because of their characteristics (autonomy, reasoning, reactivity, pro-activity, mobility and organization), which allow them to cover several needs for highly dynamic environments, especially ubiquitous communication and computing and adaptable interfaces. CISM@ combines a cloud computing approach built on top of Web Services and intelligent agents to obtain an innovative architecture, facilitating high levels of human-system-environment interaction. It also provides an advanced flexibility and customization to easily add, modify or remove services on demand. The main goal in CISM@ is not only to distribute services, but also to promote a new way of developing highly dynamic systems focusing on ubiquity and simplicity. It provides the systems with a higher ability to recover from errors and a better flexibility to change their behavior at execution time.

CISM@ is set on top of existing agent frameworks by adding new layers to integrate a cloud computing approach and facilitate the provision and management of services at two different levels, Software as a Service (Seas) and Platform as a Service (Peas) [12]. Therefore, the CISM@ framework has been modeled following the Cloud Computing model based on SOA, but has added the applications

block which represents the interaction with users. These blocks provide all the functionalities of the architecture. CISM@ adds new features to common agent frameworks, such as OAA, RETSINA and JADE and improves the services provided by these previous architectures. These aforementioned architectures have limited communication abilities.

As can be seen in Figure 1, CISM@ defines four basic blocks:

1. **PaaS (Platform as a Service).** This involves all the custom applications that can be used to take advantage of the system functionalities. Applications are dynamic and adaptable to context, reacting differently according to the particular situation. They can be executed locally or remotely, even on mobile devices with limited processing capabilities, because computing tasks are largely delegated to the agents and services.
2. **Agent Platform.** This is the core of CISM@. The set of agents contains agents predefined by the CISM@ architecture and virtual organisation for massive data analysis. The virtual organisation of the agents is established in function of the case studies, so that for the case of microarray data analysis an organisation that simulates the behaviour of laboratory personnel is generated.
3. **SaaS (Software as a Service).** These represent the activities that the architecture offers. Services are designed to be invoked locally or remotely and they can be organized as local services, Web Services, Cloud services, or even as individual stand-alone services. Services can make use of other services to provide the functionalities that users require. CISM@ has a flexible and scalable directory of services, so they can be invoked, modified, added, or eliminated dynamically and on demand. As well as the Agent Platform it includes the services of the specific case study.
4. **Communication Protocol.** This allows applications and services to communicate directly with the agent platform. The protocol is completely open and independent of any programming language, facilitating ubiquitous communication capabilities. This protocol is based on SOAP specification to capture all messages between the platform and the services and applications [13]. All external communications follow the same protocol, while the communication amongst agents in the platform follows the FIPA Agent Communication Language (ACL) specification.

One of the advantages of CISMA@ is that the users can access the system through distributed applications, which run on different types of devices and interfaces. The agents in the platform handle all requests and responses. The agents analyze all requests and invoke the specified services either locally or remotely. Services process the requests and execute the specified tasks. Then, the services send back a response with the result of the specific task.

The Web Services Architecture model uses an external directory, known as UDDI (Universal Description, Discovery and Integration), to list all available services. Each service must send a WSDL (Web Services Description Language) file to the UDDI to be added to the directory. Applications consult the UDDI to find a specific service. These services are grouped in accordance with their functionality, to facilitate their selection. However, CISM@ does not include a service discovery

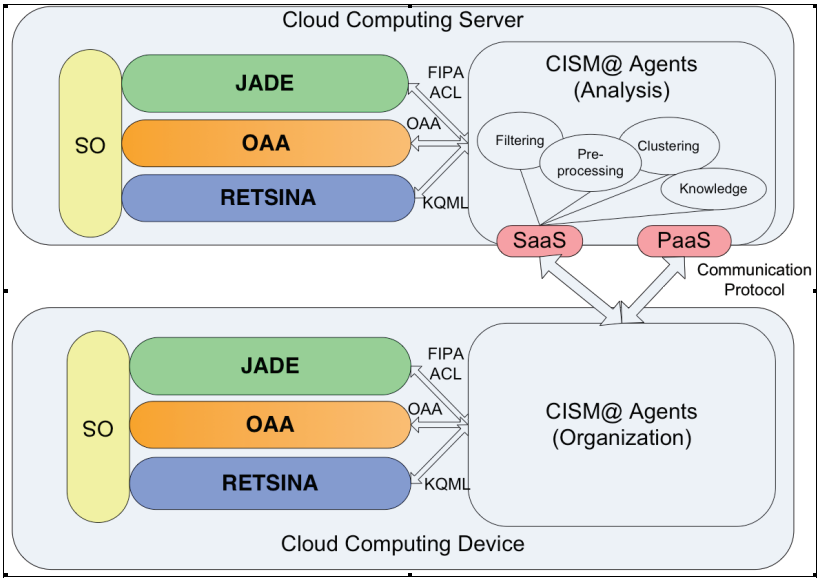

Fig. 1 Cism@ Architecture

mechanism, so applications must use only the services listed in the platform. In addition, all communication is handled by the platform, so there is no way for direct interaction between applications and services. Moreover, the platform makes use of deliberative agents to select the optimal option to perform a task, so users do not need to find and specify the service to be invoked by the application. These features have been introduced in CISM@ to create a secure communication between applications and services.

CISM@ is a modular multi-agent architecture, where services and applications are managed and controlled by deliberative BDI agents. There are different kinds of agents in the architecture, each one with specific roles, capabilities and characteristics. This fact facilitates the flexibility of the architecture when incorporating new agents. However, there are pre-defined agents, which provide the basic functionalities of the architecture.

The pre-defined CISM@ agents are described next:

1. **PaaS Agent.** This agent is responsible for all communications between applications and the agent platform. It manages incoming requests from the applications to be processed by services. It also manages responses from services (via the platform) to applications.
2. **SaaS Agent.** This agent is responsible for all communications between services and the agent platform. The functionalities are similar to PaaS Agent but the other way around. All messages are sent to the Security Agent for their structure and syntax to be analyzed. This agent also periodically checks the status of all services to know if they are idle, busy, or crashed. The analysis of messages is carried out through the use of previously implemented services.

3. **ServiceDir Agent.** This agent manages the list of services that can be used by the system. For security reasons, the list of services is static and can only be modified manually; however, services can be added, erased or modified dynamically.
4. **Control Agent.** This agent supervises the correct functioning of other agents in the system.
5. **Security Agent.** This agent analyzes the structure and syntax of all incoming and outgoing XML messages.
6. **Manager Agent.** The Manager Agent decides which service must be called by taking into account the QoS and users' preferences. Users can explicitly invoke a service, or can let the Manager Agent decide which service is best suited to accomplishing the requested task.
7. **Interface Agent.** This kind of agent was designed to be embedded in users' applications. Interface agents communicate directly with the agents in CISM@. These agents must be simple enough to allow them to be executed on mobile devices, such as cell phones or PDAs. All high demand processes must be delegated to services.

CISM@ is an open architecture that allows developers to modify the structure of the aforementioned agents. Developers can add new agent types or extend the existing ones to conform to their projects needs. However, most of the agents' functionalities should be modeled as services, releasing them from tasks that could be performed by services.

4 CISM@ Architecture in Expression Analysis

CISM@ Architecture has been adapted to the analysis of microarray expression, since it has been necessary to include agents that simulate the behaviour of a laboratory and the necessary services in the Agent Platform in order to carry out analysis.

As well as the predefined agents, the Agent Platform includes agents that simulate the roles associated with the case study. Figure 1 shows two types of agent layers:

- **Organization.** The organization agents run on the user devices or on servers. The agents installed on the user devices create a bridge between the devices and the system agents which perform data analysis. The agents installed on servers will be responsible for conducting the analysis of information following the CBP-BDI [16] reasoning model. The agents from the organizational layer should be initially configured for the different types of analysis that will be performed. Because these analyses vary according to the available information and the search results, it is imperative to establish a previous workflow configuration at the analysis layer.
- **Analysis.** The agents in the analysis layer are responsible for selecting the configuration and the flow of services best suited to the problem that needs to be solved. They communicate with Web services to generate results. The agents of this layer follow the CBP-BDI [16] reasoning model. The workflow

and configuration of the services to be used are selected with a Bayesian network and graphs, using information that corresponds to previously executed plans. The agents at this layer are highly adaptable to the case study to which they are applied. Specifically, the microarray case study includes the required agents to carry out expression analysis.

On the other hand, the services necessary to carry out expression analysis must be implemented within SaaS (Software as a Service) [0][1]. These services are those used by agents from the analysis layer to carry out data analysis.

- **Pre-processing Service.** This service implements the RMA (Robust Multi-array Average) algorithm which is frequently used for pre-processing Affymetrix microarray data.
- **Filtering Service.** The filtering service eliminates variables that do not allow classification of patients by reducing the dimensionality of the data. Three services are used for filtering: Variability, Uniform Distribution and Correlations.
- **Clustering Service.** It addresses both clustering and association of a new individual to the most appropriate group. The services included in this layer are: the ESOINN neural network. Additional services for clustering in this layer are the Partition around medoids (PAM) and demdrograms.
- **Knowledge Extraction.** The knowledge extraction technique applied has been the CART (Classification and Regression Tree) [13] algorithm and C 4.5 [14].

As shown in Figure 1, the agents from the different layers interact to generate the plan for the final analysis of data. The different system agents are distributed according to the layers and the connections that each type of agent can make with the other types of system agents and services. For example, in order to carry out its task, the Diagnosis agent at the organizational layer uses a specific sequence to select agents from the analysis layer. In turn, the analysis layer agents select the services that are necessary to carry out the data study: the filtering agent at the analysis layer selects, from the services and workflow available, those that are most suitable for the data.

The agents at the organizational layer are CBP-BDI agents with the ability to generate plans automatically based on previously existing plans in the system. Each of the CBP-BDI agents handles its own case memory in which it stores past experiences related to the specific tasks assigned to the agent. As a result, each CBP-BDI agent manages its own case memory, which is updated each time a global plan is carried out.

5 Conclusions

CISM@ facilitates the development of dynamic and intelligent multi-agent systems. Its model is based on a Cloud Computing approach where functionalities are implemented using Web Services. The architecture proposes an alternative where agents act as controllers and coordinators. CISM@ takes advantage of the agents' characteristics to provide a robust, flexible, modular and adaptable solution that can cover most requirements of a wide diversity of distributed systems. All

functionalities, including those of the agents, are modelled as distributed services and applications allowing the decoupling of functionality of agents, which contributes better system integrity with regard to failure of the agents. The decoupling of functionality also gives the system greater reutilization and adaptation to new information processing.

One of the objectives of the research activity was testing the application of Cloud Computing and Cloud services to systems and platforms oriented to the analysis of large volumes of information. The architecture has enabled the quick and efficient integration of a case study and made the inclusion of new case studies possible with a simple rearrangement of the Agent Platform, based on the needs of the problem and the definition of new services where necessary.

As a conclusion we can say that although CISM@ is still under development, preliminary results demonstrate that it is adequate for building complex systems and taking advantage of composite services. However, services can be any functionality (mechanisms, algorithms, routines, etc.) designed and deployed by developers. CISM@ has laid the groundwork to boost and optimize the development of future projects and systems that combine the flexibility of a Cloud Computing approach with the intelligence provided by agents. CISM@ makes it easier for developers to integrate independent services and applications because they are not restricted to programming languages supported by the agent frameworks used (e.g. JADE, OAA, RETSINA). The distributed approach of CISM@ optimizes usability and performance because it can obtain lighter agents by modelling the systems' functionalities as independent services and applications outside of the agents' structure, thus these may be used in other developments.

Acknowledgments. This research has been partially supported by the project PET2008_0036.

References

[1] Lina, K.S., Chien, C.F.: Cluster analysis of genome-wide expression data for feature extraction. Expert Systems with Applications 36(2-2), 3327–3335 (2009)

[2] Stadlera, Z.K., Come, S.E.: Review of gene-expression profiling and its clinical use in breast cancer. Critical Reviews in Oncology/Hematology 69(1), 1–11 (2009)

[3] Maamar, Z., Kouadri, S., Yahyaoui, H.: Toward an Agent-Based and Context-Oriented Approach for Web Services Composition. IEEE Transactions on Knowledge and Data Engineering 17(5), 686–697 (2005)

[4] Buhler, P., Vidal, J.M.: Integrating Agent Services into BPEL4WS Defined Workflows. In: Proceedings of the 4th International Workshop on Web-Oriented Software Technologies, pp. 244–251 (2004)

[5] Fuentes-Fernández, R., García-Magariño, I., Gómez-Sanz, J.J., Pavón, J.: Integration of Web Services in an Agent-Oriented Methodology. International Transactions on Systems Science and Applications 3, 145–161 (2007)

[6] Martin, D.L., Chever, A.J., Moran, D.B.: The Open Agent Architecture: A framework for Building Distributed Software Systems. Applied Artificial Intelligence 13, 91–128 (1999)

[7] Sycara, K., Paolucci, M., Van Velsen, M., Giampapa, J.: The RETSINA MAS Infrastructure. Autonomous Agents and Multi-Agent Systems 7, 29–48 (1999)

[8] Fellifemine, F., Poggi, A., Rimassa, G.: JADE–A FIPA-compliant Agent Framework. In: Proceedings of PAAM, pp. 97–108 (1999)
[9] Quackenbush, J.: Computational analysis of microarray data. Nature Review Genetics 2(6), 418–427 (2001)
[10] Affymetrix. GeneChip® Human Genome U133 Arrays, http://www.affymetrix.com/support/technical/datasheets/hgu133arrays_datasheet.pdf
[11] Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R., Lockhart, D.H.: High density synthetic oligonucleotide arrays. Nature Genetics 21(1), 20–24 (1999)
[12] Foste, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Grid Computing Environments Workshop, GCE 2008 (2008), doi:10.1109/GCE.2008.4738445
[13] Cerami, E.: Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly Media, Inc., Sebastopol (2002)
[14] Breiman, L., Friedman, J., Olshen, A., Stone, C.: Classification and regression trees. Wadsworth International Group, Belmont (1984)
[15] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
[16] Glez-Bedia, M., Corchado, J.M.: A planning strategy based on variational calculus for deliberative agents. Computing and Information Systems Journal 10(1), 2–14 (2002)
[17] Vaquero, L.M., Rodero-Merino, L., Cáceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review 39(1), 50–55 (2008)