Agent-Based Approaches in Cloud Computing: A Survey

In partial fulfillment of the requirements for course CMPT 835

Ahmed Abdel Moamen
Computer Science Department
University of Saskatchewan
Saskatoon, Canada
ama883@mail.usask.ca

Abstract—Several large-scale computational problems can be effectively solved in distributed environments which are based on the agent paradigm. These computation tasks are decomposed into sub-tasks which are assigned to the computational agents. Combining the concepts of the cloud (as a way of providing the illusion of infinite computing resources available on demand) and the agent paradigm leads to a practically zero-cost and highly scalable solution, which may be called an Augmented Cloud. The scalability of the classical cloud is limited by the number of nodes in the data center on which the cloud is deployed. Agent-based approach can augment the classical cloud by a significant increase of its scalability. This paper presents a comprehensive survey on existing research done in using agent-based approaches in cloud computing. Finally, important directions for future work are highlighted.

Index Terms—Multi-Agent Systems, Augmented Cloud, Cloud Computing.

I. INTRODUCTION

Cloud Computing (CC) has recently emerged as a viable alternative to the acquisition and management of physical or software resources [1]. Cloud computing is defined in [2] as “the aggregation of computing as a utility and software as a service where the applications are delivered as services over the Internet and the hardware and systems software in data centers provide those services”. CC adopts the concept of virtualization, service oriented architecture, autonomic, and utility computing. The cloud has more advantages and easy to implement with any business logics. Cloud delivers services from different data sources and servers located on different geographical location but the user gets single point of view from the cloud service [3].

Cloud computing service providers deliver their resources to consumers as a service (i.e., software, platform and infrastructure) [1]. Those services are based on the demands of the consumers, and the provider offers the services to consumers through virtualized resources [2]. For Infrastructure-as-a-Service (IaaS), the consumer requests are delivered in the form of Virtual Machines (VMs) that are fully satisfying the request in terms of the amount of resources. The resources of the providers are usually hosted by a data center. The data center is a set of physical machines that are interconnected, virtualized, and geographically distributed [4].

A major class of applications that demonstrated the usefulness of cloud infrastructures is data-intensive applications such as data analytics (e.g., map/reduce framework [5]). These applications consisting of a very large number of tasks that need to process a large dataset to extract meaningful information. As each individual task usually runs on a separate virtual machine (VM) in the cloud [6]. The usage of autonomous agents to execute the small tasks leads to highly scalable solution while maintaining the cloud specific flexibility. An agent is defined in [7] as “a software entity which functions continuously and autonomously in a particular environment which able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment”.

The agents in cloud environment can be running independently, not necessarily to be connected to a certain central processing unit (i.e., self-defending and self-acting). Hence, we can term them autonomous. A multiagent system consists of a number of agents, which interact with one another [8]. To successfully interact, agents require the ability to cooperate, coordinate, and negotiate with each other. First, cooperation is the process when several agents work together and draw on the broad collection of their knowledge and capabilities.
to achieve a common goal. Second, coordination is the process of achieving the state in which actions of agents fit in well with each other. Third, negotiation is a process by which a group of agents communicate with one another to try to come to a mutually acceptable agreement on some matter [9]. The integration of cloud computing and agents is called Augmented Cloud that can provide flexible and scalable services for users taking full advantages of cloud computing.

This paper presents a comprehensive survey about the usage of agent-based approaches in CC. We present a classification of the key research issues in this area of research along with discussing the proposed approaches to tackle these issues. In this paper, we try to answer some key questions about agent-based approaches in CC such as: How does agent-based approaches augment the cloud services? How agents can span the cloud (i.e., computing infrastructure) beyond the data center borders by utilizing distributed machines as computational workers? And at what cost?

The rest of the paper is organized as follows. Section II presents a brief overview on cloud computing. Section III presents a simple categorization and classification of the current research challenges in agent-based approaches in cloud computing, along with a survey of the existing research approaches proposed in literature. The open issues related to implementation of agent-based approaches in cloud computing which need to be addressed are discussed in Section IV. Section V concludes the paper.

II. OVERVIEW ON CLOUD COMPUTING

Cloud computing is a developing computing technology that uses the internet and multiple remote servers to maintain data and software applications. Cloud computing allows users to use robust software applications without installing them on a local computer [1]. This section reviews the basic design principles of CC that includes the deployment models and the new concept of augmented cloud.

A. Cloud Computing Architecture

Cloud computing providers offer their services according to several fundamental models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models (Figure 1).

- **Infrastructure as a Service (IaaS).** This is the foundation of all cloud services in which cloud vendors provide the basic infrastructure that includes virtual platforms, raw storage, firewalls and networking services. Instead of buying resources, users have to pay for their usage. The users have right to control the storage, operating systems, deployed applications and limited control of selected networking components [1].
- **Platform as a Service (PaaS).** This model sits directly above IaaS on the stack. At this level, cloud providers abstract everything up to operating system and middleware. This simply means that the cloud provides software and tools for building application to customer. It offers developers a service that provides a complete software development to testing and maintenance [1].
- **Software as a Service (SaaS).** The cloud provides software to the users. Users do not need to install any application on their PCs to access the software services provided by the cloud. The minimum requirement for access all the cloud services are internet connection with a PC (e.g., Google Calendar) [1].

B. Cloud Computing Design Principles

An architectural principle is a guideline or paradigm driving architectural decisions across the entire architecture process on a more abstract level [1]. In this subsection, we list some of these principles that motivate industrial and academic researchers to develop effective and efficient models for cloud computing.

- **Virtualization.** Users get benefit of many attractive applications without a large amount of capital investment in infrastructure and services. In other words, all hardware devices and running services are hidden (i.e., virtualized) from the end user who can access the desired services through a browser. Virtual Machines...
(VMs) (installed in cloud servers) can also be used to improve user’s ability to access cloud resources so that each user feels that there is a dedicated machine to support their applications [1].

- **Usability and scalability.** Cloud computing by its intrinsic design principles provides usable and scalable services to end users. CC provides a safe way to store users’ data on the cloud, without requiring them to worry about the issues such as software updates, virus attacks and data loss [3].

- **Interoperability Principle.** Data portability and seamless use of interoperable applications are key consideration for all cloud users [10].

- **Lightweight Principle.** Cloud computing systems should be built to support lightweight service management policies, processes, and technologies [1].

- **Genericity Principle.** Define service templates and manage service instances generically along their life cycle, across I/P/S/aaS [3].

- **Security.** Cloud users need assurance that security risks associated with storing their data and running their applications on cloud systems are understood and appropriately managed [11].

- **Protecting intellectual property rights.** Providers of cloud computing technologies and services, as with other highly innovative technologies, rely on patents, copyrights and other forms of intellectual property protection [3].

### C. Deployment Models of Cloud Computing

Deployment models in CC can be categorized into four models, namely private cloud, public cloud, hybrid cloud and community cloud. Each model has specific characteristics that supports the needs of the services and users of the clouds in different ways.

- **Private Cloud.** Private clouds are owned and operated by a single user. Private clouds utilize the same technology as public clouds and it is mainly built to enable an individual company to maximize the use of its computing resources and be more responsive to company needs [11].

- **Public Cloud.** Public clouds are owned and operated by third parties and located in data centers that operate outside of the user location. Multiple companies share the cloud resources; each cloud user is assigned own virtual computing capabilities based on a common set of physical resources [11].

- **Hybrid Cloud.** Hybrid clouds are combinations of multiple clouds that are both public and private. These clouds are created by individual customers to meet their precise needs [11].

- **Community Cloud.** Community cloud infrastructure is shared by a number of organizations and supports a specific community that shares the same concerns (e.g., mission, security requirements, government and education). It may either be managed by the organizations or a third party and may exist on premise or off premise [11].

### D. Augmented Cloud

From the cloud computing service models [11] point of view, the Augmented Cloud can be seen as a kind of Infrastructure as a Service (IaaS) in which part/whole computations are executed by autonomous agents. This level of abstraction is too low when taking into account both the development and deployment of (multi)agent-based systems [12]. Augmented Cloud can be considered as the second layer of the cloud computing reference architecture which corresponds to the Platform as a Service (PaaS) model. In this context it can be named Agent Platform as a Service (AgPaaS) to emphasize the agent orientation of the platform [11].

### III. AGENT-BASED CLOUD COMPUTING

Some of the essential characteristics of cloud computing include resource pooling and resource sharing. In clouds, computing resources are pooled to serve multiple consumers, and data are shared by a broad group of cross-enterprise and cross-platform users [1]. Resource pooling and sharing involve (i) combining resources through cooperation among cloud providers; (ii) mapping, scheduling, and coordination of shared resources; and (iii) establishment of contracts between providers and consumers. In agent-based cloud computing, cooperation, negotiation, and coordination protocols of agents are adopted to automate the activities of resource pooling and sharing in clouds [12].

Agent-based cloud computing is concerned with the design and development of software agents for supporting cloud service discovery, service negotiation, and service composition [12], [13]. Sim [14] proposed that software agents are appropriate tools for autonomously managing cloud resources. Whereas consumers need to make decisions to select suitable providers and negotiate with providers to achieve ideal service contracts. Providers need to make decisions for selecting appropriate requests to accept and execute depending on the availability of resources. Since agents are capable
of making decisions when carrying out tasks on behalf of their users, and interacting with other agents through negotiation, cooperation, and coordination. All the above-mentioned challenges provide the motivations for adopting autonomous agents to allocate resources amid dynamically changing resource demands.

Figure 2 illustrates a classification of the existing research areas in Agent-Based Cloud Computing. We classify the research approaches presented for addressing the issues of Agent-Based Cloud Computing into five main categories, namely: cloud crawlers, agent-based cloud commerce, concurrent cloud resource negotiation, grid resource negotiation, and agent-based security approaches. The following subsections will compare and discuss existing approaches from these perspectives.

A. Cloud Crawlers

A crawler is a program that visits web sites and reads their pages and other information in order to create entries for a search engine index. In the cloud, many parallel threads of the cloud crawler are deployed to gather information about cloud service providers [12]. The architecture of a cloud crawler is shown in Figure 3.

It consists of a crawling agent, a URL filter softbot (a software agent), and a database agent. The crawling agent traverses the WWW to extract webpages that are relevant to cloud computing services. Starting from a root URL that is predefined by a user, the crawler agent traverses websites by following hyperlinks. As the crawling agent visits a website, its fetcher module downloads a copy of the webpage, then examines the contents and extracts the link data (consisting of URLs and hyperlinks) and the contents of the webpage by parsing the downloaded document (stripping away the html tags) [12].

The fetcher module stores the extracted contents in a shared content memory and updates the link data in the crawling agent’s link database. Using an ontology of cloud concepts, the URL filter softbot examines the contents of webpages stored in the shared content memory and determines if the contents are relevant to cloud services. The URL filter softbot scans the texts in each document that are stored in the shared content database to search for cloud concepts in the ontology. A document that contains more cloud concepts receives a higher score. The database agent examines relevant documents in the shared content memory by extracting the name of the service provider, service type, price, and technical specifications such as CPU speed and RAM capacity. All such information together with the URL of the webpage will be stored in the database [12].

Sim [12] proposed a novel agent-based search engine for cloud service called Cloudle. The author showed that the agent-based negotiation mechanisms can be effectively adopted for supporting cloud service negotiation and cloud commerce. He also showed that agent-based cooperative problem solving techniques can be effec-
tively adopted for automating cloud service composition. Cloudle consists of (i) a service discovery agent that consults a cloud ontology for determining the similarities between providers’ service specifications and consumers’ service requirements; and (ii) multiple cloud crawlers for building its database of services. Cloudle supports three types of reasoning: similarity reasoning, compatibility reasoning, and numerical reasoning.

B. Agent-Based Cloud Commerce

A cloud computing system consists of a collection of interconnected and virtualized computers dynamically provisioned as one or more unified computing resource(s) through negotiation of service-level agreements (SLAs) between providers and consumers [1]. The challenge in cloud service negotiation is to establish SLAs between consumers and brokers, and between brokers and service providers. Whereas e-commerce negotiation mechanisms involve two types of participants (buyers and sellers) in only one market and participants are not allowed to break contracts [13].

The problem of developing a complex negotiation mechanism for cloud commerce is much more complex because a complex cloud negotiation mechanism specifies parallel negotiation activities among three types of participants (consumers, brokers and providers) in multiple interrelated markets. In the same time participants are allowed to breaking contracts by paying penalty fees [12]. In a cloud business model, consumers pay service providers for consumption of computing capabilities. A market-oriented approach for managing cloud resources is necessary for regulating the supply and demand through flexible and dynamic pricing [1].

Sim [15] described a market model for trading cloud resources. The author proposed an agent-based testbed for simulating cloud commerce as shown in Figure 4. The figure shows the design of an agent-based testbed for simulating cloud commerce. The model consists of resource/service providers, consumers, and brokers. Brokers purchase resource capacities from providers. They compose multiple resources from different providers into bundled services, then sublease the unified services to consumers. Each broker accepts requests from multiple consumers and each consumer can also submit its service requests to many brokers. Consumers, brokers, and providers are bound to service contracts through SLAs that specify the details of the service to be provided agreed upon by all cloud participants, and the penalties for violating the expectations.

Figure 5 shows a cloud negotiation mechanism for facilitating the negotiation activities (i) between consumer agents and broker agents; and (ii) between broker agents and provider agents. Each broker agent can accept requests from many consumer agents, and each consumer agent can also submit its requirements and requests to many broker agents. So, there is a need for a many-to-many negotiation model be adopted for negotiation between consumer agents and broker agents. Since a cloud service may be dynamically composed using multiple types of cloud resources, each broker agent can potentially negotiate in multiple types of cloud resource markets with multiple groups of cloud providers that provide different types of cloud resources. Hence, a concurrent one-to-many negotiation mechanism is adopted to facilitate concurrent negotiation activities between broker agents and different groups of provider agents [15].

C. Concurrent Cloud Resource Negotiation

Sim et al. [16] proposed a concurrent negotiation protocol for concurrent negotiation of multiple service-
level agreements (SLAs). Figure 6 shows a concurrent negotiation mechanism of a broker agent for establishing multiple SLAs for a collection of cloud resources. It consists of a coordinator which coordinates the parallel negotiation activities for acquiring \( n \) different types of cloud resources in \( n \) different cloud resource markets. In each cloud resource market, a broker agent establishes an SLA by negotiating simultaneously with multiple provider agents for one type of cloud resource. Furthermore, both broker and provider agents can be freed from a contract (i.e., an agent can decommit a contract) by paying penalty fees to their opponents.

The reasons for allowing decommitments are as follows: (i) if a broker agent cannot acquire all its required resources before its deadline, it can release those resources acquired so that providers can assign them to other broker agents; and (ii) it allows a broker agent that has already reached an intermediate contract for a resource to continue to search for better deals before the entire concurrent negotiation terminates. In negotiating for one type of cloud resource in a resource market, there is a commitment manager that manages both commitments and decommitments of contracts.

D. Grid Resource Negotiation

Since Grid Computing systems involve large-scale resource sharing, resource management is essential to their operations. However, providing efficient resource allocation mechanisms in the Grid is a complex undertaking due to its scale and the fact that resource owners and consumers may have different goals, policies, and preferences [17]. So, there is a persistent need for Grid negotiation mechanisms to solve these problems.

Negotiation is a form of decision making with two or more actively involved agents who cannot make decisions independently, and therefore must make concessions to achieve a compromise [18]. On the other hand, an auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market agents [19]. While auction-like protocols play a major role when determination of value is the primary concern, negotiation-like protocols may be more appropriate when participants are not only concerned with determining value, but also other factors (e.g., inter-business relationships and success rates). In situations involving inter-business relationships, an integrative negotiation mechanism (which finds solutions to reconcile the interests of all agents) may be more appropriate than auctions, because auctions focus on determining the value of objects of unknown value while negotiations are about cooperating to create value [17].

Lang [20] proposed a multiple-attribute negotiation mechanism for managing the resource usage in a computational Grid using a Grid carrier agent (GCA). The goal of mechanism is to implement the intermediary function of-matching suppliers’ service capabilities and resource consumers’ demand profiles. The authors proposed agents that autonomously negotiate multiple-attribute Grid service contracts. The negotiation protocol consists of (i) a distributive negotiation phase; and (ii) an integrative negotiation phase, in which agents attempt to find joint gains while trying to maintain the utility distribution outcomes from the distributive negotiation phase.

Ghosh et al. [21] proposed a fair pricing strategy and an optimal static job allocation scheme to solve the issue of load balancing in a mobile computational Grid. In the proposed model, a mobile Grid computing system consists of mobile devices that are sellers of resources, and Wireless Access Point (WAP) servers that negotiate with mobile devices to purchase resources for providing services to a community of Grid resource consumers. The negotiation between a WAP server and a mobile device is modeled as a two-player non-cooperative game of incomplete information. If there are \( n \) mobile devices under a single WAP server, the WAP server will compose a price per unit resource vector \((p_1, p_2, ..., p_n)\) by playing \( n \) such games with all \( n \) corresponding mobile devices.

E. Agent-based security approaches

Security in cloud computing is arguably the most relevant concern that may limit a wider adoption of this promising technology. Many factors have an impact on cloud security, but the multi-tenant nature of the cloud brings the newest and more challenging problems to cloud settings. IaaS clouds inherit many security concerns associated to the technologies they use. However, the large amounts of users sharing resources creates new security concerns. The key for enabling cloud operators to provide seemingly unlimited scaling potential is resource sharing. This enables maximum utilization of the available assets, but introduces new multi-tenancy concerns [11].

The resources shared are not only the physical machines, but it also the network (i.e., conventional links and even the network interface on the machine). Virtualization eases computational resources sharing and it is broadly used in current clouds (e.g. Amazon employs a tuned version of Xen [22]). However, virtualization
proposes a series of threats and vulnerabilities due to the transformation from a dedicated infrastructure into a multi-tenant scenario where machines and networks are totally shared [23].

Doelitzscher et al. [24] proposed an incident detection system for cloud computing named Security Audit as a Service (SAaaS), which is built upon autonomous agents collecting data directly at the source, analyse and aggregate information and distribute it considering the underlying business process. This data interpretation is achieved using a Security Service Level Agreements (SSLA) policy modeling engine that allows to define monitoring events and consider business process flows. The usage of autonomous agents enables a behavior anomaly detection of cloud components while maintaining the cloud specific flexibility. The proposed system respects the following cloud specific attributes: (i) a high number and complexity of distributed systems; (ii) a changing infrastructure; and (iii) an interpretation of the cloud activation in respect to business processes.

SAaaS agents are enabled to be aware of underlying business flows. Therefore several simple agents are logically belonging together forming an agent group where every agent knows about the other agent. While a single agent monitor simple events (e.g., user login on a VM) and share them with the agent group complex events can be created. Combining events from system agents (VM agent and host agent - see Figure 7) and infrastructure monitoring agents (network agent, firewall agent) incident detection is not limited to either host or network based sensors which is especially important for the characteristics of cloud environments.

Raj et al. [22] introduced a virtualization service implemented as Xen Virtual Machine (VM) extensions, which provides Role Based Access Control (RBAC) based on a trust value of a VM. This trust is based upon a VM’s attributes such as the number of open network connections. Access to different cloud services (e.g., file access) is given on a VM’s trust value. The presented implementation methods are following the same idea as the SAaaS architecture [24]: trust generation via behavioral monitoring to build a normal cloud usage profile.

IV. OPEN RESEARCH PROBLEMS

There are several open issues related to the implementation of Augmented Cloud Computing that need to be addressed. In this section, we present some of these issues and the possible research directions in the evolution of this approach.
A. Agent-Based Computation Platform

There is a persistent need for an agent-based computation platform to support building a wide range of agent-based optimization and simulation systems. These systems should utilize various meta-heuristics such as evolutionary algorithms [9]. A computation task to be executed on the platform is defined by providing a computation description file, which includes: the computation decomposition details (e.g., types of agents, their structure, types of operations that specify algorithms), problem-dependent parameters, and the problem stop condition.

Based on the description file, on the platform startup, the computation context is built from the necessary components. Next, all the required agents and their environments are created, configured and distributed among nodes. Finally, the computation can be started and is performed until it reaches the stop condition. During the execution time some of the computational agents are attached to the platform monitors. The monitors (type of agents) are responsible for collecting problem-dependent data, used to visualize the current state and the final results of the computation [12].

B. Agent Platform as a Service

From the cloud computing service models [11] point of view the Augmented Cloud can be seen as a kind of Infrastructure as a Service (IaaS), in which (at least some of) the computational resources are web-browsers based. What is needed is a set of services/libraries forming a highly scalable platform dedicated for the target domain—agent-based systems. It can be considered as the second layer of the Cloud Computing Reference Architecture [25] which corresponds to the NIST Platform as a Service (PaaS) [11]. In this context, it can be named Agent Platform as a Service (AgPaaS) to emphasize the agent-orientation of the platform.

C. Complex Negotiation for Cloud Economy

In a cloud computing environment, the dynamic configuration of a personalized collection of resources often requires cloud participants (consumers, brokers, and providers) to establish service-level agreements (SLAs) through negotiation. However, to date, state-of-the-art approaches in cloud computing provides limited or no support for dynamic SLAs negotiation. So, there is a need for a complex cloud negotiation mechanism that
supports negotiation activities in interrelated markets such as: (i) provide a cloud service market between consumer agents and broker agents, and multiple cloud resource markets between broker agents and provider agents; (ii) specify the negotiation protocols and strategies of consumer and broker agents in a cloud service market; and (iii) present the design of the contracting and coordination algorithms for the concurrent negotiation activities between broker and provider agents in multiple cloud resource markets [13].

The complex cloud negotiation mechanism should be designed to support complex negotiation activities in interrelated markets in which the negotiation outcomes between broker and provider agents in a cloud resource market can potentially influence the negotiation outcomes of broker and consumer agents in a cloud service market [8].

V. CONCLUSION

In this paper, we have given an overview of agent-based approaches in cloud computing. We have discussed the motivations for combining the concepts of the cloud and the agent paradigm to provide highly scalable solutions. We also discussed the essential design considerations for this promising technology such cloud crawlers, agent-based cloud commerce, concurrent cloud resource negotiation, and agent-based security approaches. We have presented a taxonomy of issues found in this area, and the approaches in which these issues have been tackled. The new computing paradigm Augmented Cloud is also highlighted and discussed. The Augmented cloud can be seen as a kind of Infrastructure as a Service (IaaS), in which the computational resources are web-browsers based. Finally, we present some open research problems in agent-based approaches in cloud computing along with the possible research directions in the evolution of it.

REFERENCES


