A Unified Architecture for Optimal Resource Utilization in a Heterogeneous Cloud Environment

L. Arockiam and A. Stanislas

Abstract—Cloud computing is the latest effort in the field of computer technology in delivering computing resources as a service. It is a paradigm shift taken from computing as a product that is to be purchased to computing as a service that is delivered to consumers as 'pay-per-use' over the internet on clouds. The biggest challenge that the present business world faces in cloud computing is the lack of a single and standard architectural method that can meet the requirements of an enterprise cloud approach. In order to address this challenge, we present an architectural framework with an algorithm which enhances scalability in a unified cloud environment. It enables the service providers to manage and to allocate the resources according to the demand of the users. This paper provides a better understanding of the architectural designs for scalability in cloud computing, allocates the resources optimally to reduce the cost and maintains customer-provider relationship.

Index Terms—Scalability, unified architecture, cloud broker, service-allocator, service-provider.

I. INTRODUCTION

Down the years in the last century, Leonard Kleinrock [1], one of the chief scientists of the original Advanced Research Projects Agency Networks (ARPANET) project which was the seed of the Internet, had the vision and predicted the future computing with these words, "As of now, computer networks are still in their infancy, but as they grow up, and become sophisticated, we will probably see the spread of 'computer utilities' which like present electric and telephone utilities, will serve individual homes and offices across the country". His prediction has come into reality in the form of cloud computing after a century. Cloud computing is the latest incarnation of a general purpose public computing utility [2].

The ultimate goal of cloud computing is to provide on-demand computing services [3] with high reliability [4], scalability [5]-[7], availability, Quality of Service (QoS) [8], and with minimum cost in a federated [9] computing environments. It is important to note that the concept of cloud computing is not new, but it represents the next evolutionary step of several initiates carried out in the last few years, including distributed computing [10], grid computing [6], [11], [12], utility computing [6], [12]-[14], virtualization [6], [7], [15]-[17], and server clusters [15], [18].

Cloud computing is considered to be the business-oriented evolution of Grid computing which was focused on research

and collaboration. It provides a paradigm shift of business and IT infrastructure, where computing power, data storage, and services are outsourced to the third parties and made available as commodities to enterprises and customers. In this paper, we would like to consider important architectural requirements for cloud computing systems from the point of view of enterprise users, so as to enable the service providers to manage and to allocate the resources according to the demand of the users.

Cloud computing must evolve as an open environment, minimizing vendor lock-in and increasing customer choice. However, though cloud computing offers huge opportunities to the IT industry, the development of cloud computing technology is currently at its infancy, with many issues still to be addressed. Cloud providers must work together to ensure that the challenges to cloud adoption are addressed through open collaboration and the appropriate use of standards. Any community effort around the open cloud should be driven by customer needs, not merely the technical needs of cloud providers, and should be tested or verified against real customer requirements. So, the success of next-generation cloud computing infrastructures will depend on how efficiently/ effectively these infrastructures will be able to instantiate and dynamically maintain computing platforms, constructed out of cloud resources and services, which meet arbitrarily varying resources and service requirements of cloud customer applications [19].

The rest of the paper is organized in the following manner. Different works related to the proposed system by a number of authors are presented in Section II and Section III presents the centralized system architecture with the descriptions of its preliminary concepts. The Section IV gives the detailed account of the MMS system with a Logical Flow Diagram (LFD) and a searching algorithm and points out the advantages of the proposed system. Finally, Section V concludes with the suggestions indicating the research areas for the future work.

II. MOTIVATIONAL SCENARIO

Rochwerger *et al.* [2] identified a number of inherent deficiencies in the current cloud computing systems, such as, limited scalability of single provider of clouds, lack of interoperability among the cloud providers and lack of built-in business service management support. Therefore, they proposed an architecture called 'RESERVOIR' to address the needs of enterprise-grade solutions, such as support for infrastructure level SLAs. This was an extensible and modular architecture which supported the Business Service Management (BSM) and Federation of Clouds. This architecture allowed the service providers of cloud

infrastructure to dynamically partner with each other to create seemingly infinite pool of IT resources. But it fully preserved the autonomy of technological and business management decisions. It leveraged and extracted the advantages of virtualization and embedded autonomous management into the infrastructure. The vision of this approach was to enable on-demand delivery of IT services at competitive cost. In their analysis, they used the SAP system to validate this RESERVOIR model.

Rajkumar Buyya *et al.* [20] provided architecture for creating cloud with Market-Oriented Resource Allocation by leveraging technologies like Virtual Machines (VMs). They also gave insights on market-based resource management strategies that encompassed both customer-driven service management and computational risk management. This was helpful to sustain the Service Level Agreement (SLA)-oriented resource allocation. Then, they presented some cloud platforms which were developed in industries and the market-oriented resource allocation of clouds as realized in Aneka enterprise cloud technology. They also discussed about interconnecting clouds for dynamically creating global cloud exchanges and markets. They described a meta-negotiation infrastructure to establish the global cloud exchanges.

Cloud computing was making its way to unlimited computing power with high availability. But the cloud computing service-providers seemed to be reserved to very large companies that could achieve having huge data center. So, the small companies and organizations were forced to play the role of cloud users. In order to overcome this issue, Boris Mejias *et al.* [21] proposed architecture to organize a set of mini clouds provided by different institutions of similar services (eg. SaaS) that would form a large cloud which would appear to its users as a single cloud unit. This architecture required a self managing behavior in order to deal with the complexity of matching cloud user's request.

Lijun Mei *et al.* [22] presented a comparative study of cloud computing with service computing and pervasive computing in order to identify the research issues. Their comparison was based on the classic model of computer architecture. To improve the business responsibility to the customer, they proposed architecture called 'Service-Oriented Computing' which could be implemented in cloud environment. In this architecture, there were services which published themselves, discovered their peer services and bound to the latter services to form service compositions using standard protocols. There were also service registries in a service-oriented network. The service provider registered itself with one or more service registries. A service consumer first discovered the service from a registry and then bound to the service.

Though the computing providers have setup several data centres in different geographical locations, it did not support mechanism and policies for dynamically coordinating load distribution among different cloud-based data centres in terms of achieving QoS. To overcome such situations, Rajkumar Buyya *et al.* [9] suggested that creating a federated cloud computing environment (InterCloud) would facilitate just-in-time and scalable provisioning of application services. So, in order to handle sudden variations in service demands, they created this computing environment that supported

dynamic expansion or contraction of capabilities like VMs, services, storage and database. The architecture of the proposed model is given in the following Fig. 1.
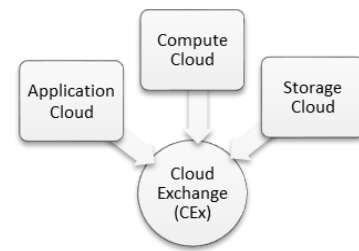

Fig. 1. InterCloud model.

The biggest challenge that the present business world faces in cloud computing is the lack of a single and standard architectural method that can meet the requirements of an enterprise cloud approach. So, Bhaskar Prasad Rimal *et al.* [7] proposed a model which fulfills the architectural requirements for cloud computing systems, and especially creates a new horizon of opportunity for enterprises. To avoid heavy upfront investments, it introduces new operating and business models that allow customers to pay for the resources they effectively use. Their proposed model consists of three layered architectural requirements namely; provider requirements, enterprise requirements and user requirements. They have taken all the important factors (SLA, QoS, Scalability, Security, Interoperability, VMs, Fault tolerance etc.) into account in their architecture. They also discuss and make a comparative analysis of the common requirements.

In the present vision of computing, there are various IT paradigms which are ready to deliver computing as utility like any other utilities. Having studied and knowing the growing demand for computing, Rajkumar Buyya *et al.* [23] defined architecture for creating market-oriented clouds and computing atmosphere by leveraging technologies such as virtual machines. This architecture has the provision of market-based resource management strategies for SLA-oriented resource allocation. For their proposed architecture, they introduce a new toolkit called 'CloudBus'. This toolkit has various components like, Aneka – a software platform for developing and deploying cloud computing applications, Broker – a middleware for scheduling distributed applications, Workflow Engine – a middleware that handles dependant tasks, Market Maker which matches user's requirement, InterGrid that links with various grids, MetaCDN – a system that intelligently places user's content into storage cloud and CloudSim – a simulation toolkit that helps user's model.

III. PROPOSED SYSTEM

Cloud computing is the attraction for the present IT resource business owners as it eliminates the requirements for users to plan ahead for provisioning. It allows enterprises to start from the small firms and increase resources only when there is a rise in service demand. The biggest challenge that the present business world faces in cloud computing is the lack of a single and standard architectural method that can meet the requirements of an enterprise cloud approach. In order to address this challenge, we present a central architectural framework with an algorithm. This model

would enhance the scalability in a heterogeneous cloud environment. It enables the service provider to manage and allocate the resources according to the demand of the users. The general architecture of the proposed model is given in the following Fig. 2.
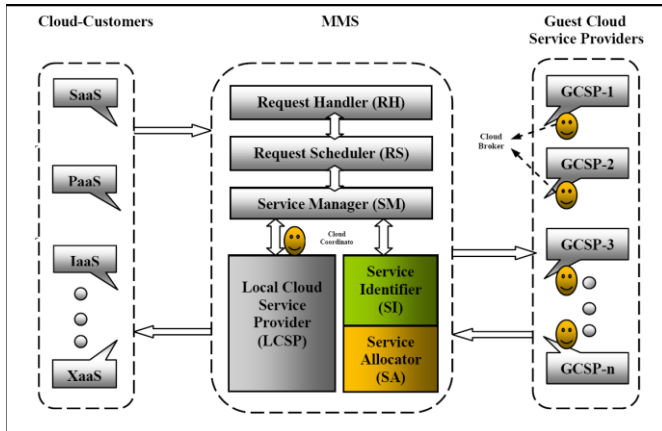

Fig. 2. Centralized heterogeneous cloud model.

As shown in Fig. 1, the cloud customers make their requests to the proposed system called Match Making Service (MMS). The MMS system has different components having their responsibilities to find the required services from the appropriate service providers. The responsibilities of each component are given in detail in the followings.

A. Customer/User (CCU)

In general, the customers are the thin clients who use services in the cloud computing environment. They interact with the cloud management systems through an automatic system such as broker or scheduler who act on customer's behalf to submit service requests from anywhere in the world to the cloud to be processed. To start a service session in such an environment, each user makes a service check-in request to the CSP a billing transaction. To end the service session, the user can check-out request to the CSP with a billing transaction.

B. Request Handler (RH)

When a service request is submitted by the customer, the service Request Handler (RH) interprets the submitted requests for the QoS requirements before determining whether to accept or to reject the request. This process ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources available. It also needs the latest status information regarding resource availability from VM monitoring mechanism and workload processing from service monitoring mechanism in order to make resource allocation decisions effectively. Then, it assigns requests to SM to allocate the resources.

C. Request Scheduler (RS)

Once the request is accepted by the Request Handler (RH), the main work of the Request Scheduler (RS) component is to schedule the requests to SM based on the information provided by the CSP about the availability of the services. The scheduler waits for the response from the cloud service provider informing that the new virtual machines are ready for use.

D. Service Manager (SM)

The responsibility of this component, Service Manager (SM) is to discover the appropriate service provider, select the services needed for the customer, coordinate with the service provider through Service Level Agreement (SLA), and to monitor the services. When the request is sent from the Request Scheduler (RS) to the SM, the main function of this component is to automate this process of deploying the necessary services. Once the user's request is accepted, the SM consults locally with the Local Cloud Service Provider (LCSP) regarding the availability of the services (SaaS, PaaS, IaaS) required to satisfy the request. If the service is available in LCSP, then the feedback about the utilization status is submitted, the market policies and accountability are enquired and the service is provided.

It has also to take care of the reliability of the service, cost and the SLA violation reporting. Its highest level of abstraction is to interact with the service providers to receive their service manifests, negotiate the pricing, and handle the accounting and billing. SM is also responsible for adjusting its capacity through monitoring the deployed services. The account and billing of the services are done based on the information provided by the service manager (SM). There can be two modes of payment in CSP, namely pre-paid and post-paid. Both pre-paid and post-paid modes are based on the resource utilization information provided by the SM.

E. Cloud Coordinator (CC)

The cloud coordinator service is responsible for the management of domain specific enterprise clouds and their membership to overall federation driven by market-based trading and negotiation protocols. It provides a programming management and deployment environment for applications in a federation of clouds. The cloud coordinator exports the services of a cloud to the federation by implementing basic functionalities for resource management such as scheduling, allocation, discovery or identifier.

F. Cloud Broker (CB)

Cloud brokers acting on behalf of the users or customer identify the suitable cloud service providers through the cloud exchange and negotiate with the cloud coordinators for an allocation of resources that meets the QoS needed of users. The brokers have the ability to provision computing and storage service in cloud resources. It also provides QoS parameters in its service description for application regarding a mix of public and private cloud resources.

G. Service Identifier (SI)

In order to dynamically perform scheduling resource allocation, and VM migration to meet SLAs in a federated network, it is mandatory that up-to-date information related to cloud's availability; pricing and SLA rules are made available to the outside domains through cloud broker. It helps updating the resource status metrics including utilization, heat dissipation, power consumption based on feedback given by the sensor component. It facilitates the market and policy engine in periodically publishing the pricing policies. It aids the scheduling and allocation

component in dynamically discovering the clouds that offer better optimization for SLA constraints such as deadline and budget limits. It also helps the virtualization component in determining load and power consumption. Such information aids the virtualization component in performing load-balancing through dynamic VM migration.

H. Cloud Service Provider (CSP)

In current utility computing environments, especially in cloud computing, it is a vital task to identify the right Cloud Service Provider (CSP) who has the resources that can meet the costumer's requirement efficiently and optimally. Such computing environments contain a large collection of different types of resources, which are distributed worldwide. These resources are owned and operated by various providers with heterogeneous administrative policies. Some providers have only a particular type of computing service like Microsoft provides SaaS and PaaS, Salesforce.com provides only SaaS and IBM provides IaaS and so forth. Hence, the difficulty arises when a customer is in need of multiple types of services. He has to approach various CSPs for his requirements which is a tedious task. In the proposed centralized architecture, there are two types of providers, namely, Local Cloud Service Providers (LCSP), and Guest Cloud Service Provider (GCSP). When a customer makes a request for service, it first contacts the LCSP, and then it searches among the GCSPs in parallel for the required service. So, every CSP can be a LCSP and at the same time be a GCSP for the other providers. Hence, a CSP is a customer when he has to depend on other providers for the required services. So, there is a sense of interdependency among the cloud service providers. Resources or services can join and leave computing environment at any time. Therefore, their status changes dynamically and unpredictably. Solution for service provider discovery problems must efficiently deal with scalability, dynamic changes, heterogeneity and autonomous administration.

I. Service Allocator (SA)

The Service Allocator (SA) acts as the interface between the cloud computing infrastructure and the external user or broker. Here, the infrastructure will be the resources available with the LCSPs and GCSPs. The information about the resources available and the resources provisioned will be instantly updated in the service repository through the service manager. So, SA has to optimally allocate the resources to the customer according to the demand.

IV. MMS ARCHITECTURE

The proposed MMS system is a single standard architecture that can meet the requirements of the customers in a heterogeneous cloud environment. The unique characteristic of this system is to provision the resources through a single door even though there are a number of service providers with different IT resources. It makes the customers easy to approach one provider for any type of needs. It is the duty of the broker to identify the appropriate service provider for the required services and to allocate the resources. The simple architectural model is designed in the following Fig. 3 to explain the MMS system.
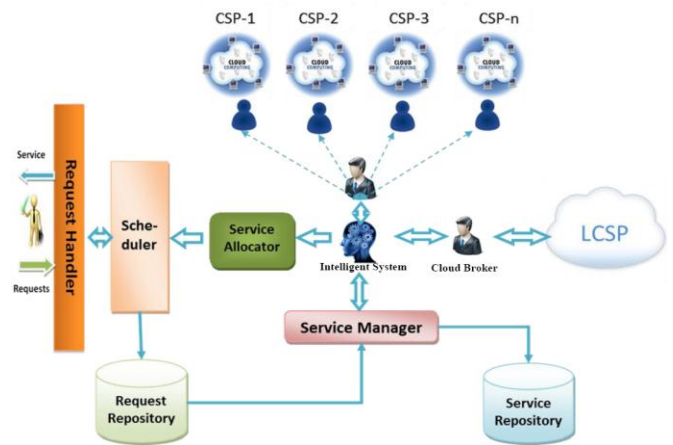

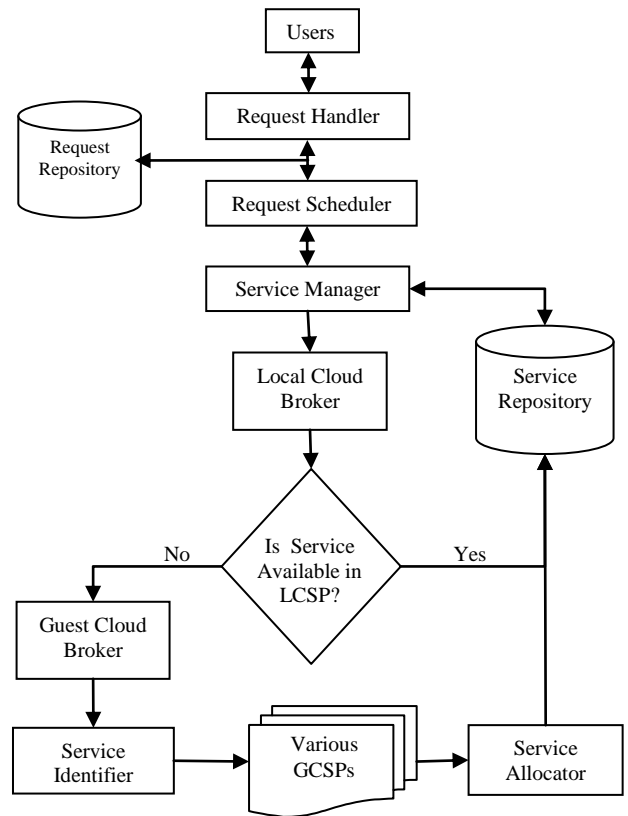Fig. 3. Match Making Service (MMS) architecture.

Fig. 4. Logical flow diagram of MMS.

As shown in the Fig. 3, the request submission and the service provision are carried out through a single entry point. All the requests are deposited in the request repository for further future prediction. The submitted requests are scheduled through the service manger to the local cloud broker who identifies the appropriate service provider. If the required service is available with the Local Cloud Service Provider (LCSP), the broker provisions the resources through the allocator after keeping the information in the service repository. In the case of non-availability of the resources with the LCSP, the broker searches for the resources among the Guest Cloud Service Providers (GCSP) and allocates the service through the service allocator. The information of every service allocated by the service manager is stored in the Service Repository before it is allocated. The unique feature of this system is that every provider may act as a LCSP at times when the service required is available with them and GCSP at other times when the required service is not

available with the LCSP. The Logical Flow Diagram (LFD) gives the detailed explanation of the functions of MMS system in Fig. 4.

As shown in the Logical Flow Diagram, the customer sends the service requests to the request handler. The request handler stores all the requests in the request repository and interprets the submitted requests for the QoS and accepts the request based on the information about the availability of the resources. Once the request is accepted, it is sent to the request scheduler to schedule it to the service manager. The service manager discovers the suitable service provider for the requests and negotiates for further formalities like SLA, QoS, accounting, security and billing and so forth. First, it checks the local cloud service provider for requested service through the local cloud broker and when it is not available in the local cloud, a parallel search is made amongst various guest cloud service providers through guest cloud broker. The identified service, after having stored its information in the service repository is allocated to the customer through service manager. The service manager keeps track of all information about the service rendered and released based on the data stored in the service repository. The data available in the service repository is used to check both the availability of the resources particularly the virtual machines and the released virtual machines.

A. Algorithm

```
// Declarations
N ← number of Requests
S ← Services
SR ← Service Request
RH ← Request Handler
RR ← Request Repository
RS ← Request Scheduler
SM ← Service Manager
SR ← Service Repository
LCSP ← Local Cloud Service Provider
GCSP ← Guest Cloud Service Provider
SA ← Service Allocator
SI ← Service Identifier
_____

Step-1: Submit SR in RH
Step-2: Information of SR stored in RR
Step-3: SR scheduled to SM
Step-4: S availability checked
        If S in LCSP, information stored and S is provided
        Else
        SI Searches CSP1 to CSPn and fetches S among GCSP and
            allocates
Step-5: SM monitors and Manages Services Deployed
Step-6: End.
```

B. Advantages of the Proposed System

1) Optimized Resource Utilization: Every Cloud Service Provider (CSP) intends to utilize the resources to the maximum possible. The business growth of the CSPs will be hindered if resources are kept idle or underutilized. Since the proposed system has the network of various CSPs with different types of resources like SaaS, PaaS, IaaS and so on, the resources can be shared among the CSPs in order to meet the demand of the customers. When the resources are shared at the instant request, the utilization rate of the resources will be increased. For example, every provider will have a threshold value say, 80% for the maximum utilization of the

resources. When the threshold point is reached, the system gives an alert in order that the resources have to be physically added. The MMS system with a number of CSPs with different resources would enhance the optimized resource utilization which would enable to increase the threshold value up to 90%.

2) Reduced Cost for the Customer: The optimized utilization of the resources would benefit the customers. The providers can reduce their capital expenditures and use operational expenditures for increasing their computing capabilities. When the computing capabilities are increased and the capital expenditure is reduced the cost of the resources and services would automatically come down. This would certainly benefit the customers.

3) Increased Profit for the Service Providers: In the MMS system, the CSPs are networked together to meet the requirement of the need of the time. Since the resources are used optimally, the management cost, infrastructure cost and monitoring cost would come down and thus the profit for the providers will be in rise. Also, the service providers will act at times as providers when they have resource with them and act as the customer when they borrow services from other providers. When they borrow services from other providers, both the lender and the provider make profit.

4) Better Network among the CSPs: The proposed system encourages the CSPs to establish a network among them in order to provision the resources efficiently. Every CSP does not need to have all types of resources. Instead, each provider can have different type resources and can share among them according to the demand of the customers. For example, a provider CSP-1 has only IaaS and CSP-2 has only PaaS. When there is a service request for IaasS through CSP-2, the provider CSP-1 can provision the service to CSP-2 and vice versa. So, as there is a number of CSPs in the network with various services, there can be a better network among the CSPs for efficient provisioning of the service.

5) Better Quality of Service (QoS): The MMS system ensures the customers with the Quality of Service as it reduces the response time, increases the reliability, guarantees the availability, and reduces the cost of service.

6) Enhanced Customer Satisfaction Level: The ultimate goal of any service provider is to increase the customer satisfaction level. The customers do not have to approach many providers for different services as the services are made available at one desk through Match Making Service. The MMS enables every CSP in the network to be provider at times and user other times. That way, any small user or provider can be the part of this MMS system.

7) Highly Scalable Nature: The key characteristic of cloud computing is elasticity. When the resources are available with various CSPs, scaling is highly possible without going for adding extra physical resources.

V. CONCLUSION AND FUTURE WORKS

The aim of this paper was to provide a novel mechanism wherein all the Cloud Service Providers (CSP) could share their resources in order to utilize them efficiently. Automating this process of negotiating service requests from various CSPs for different services in cloud environment needs a central architecture. In a heterogeneous environment

such as cloud, the resources have to be optimally utilized in order to increase the profit for the providers, to reduce the cost for customers, and thus to increase the customer satisfaction level and a better Quality of Service (QoS). In the proposed MMS system, a central architecture for enhancing scalability in heterogeneous cloud environment is presented. A searching algorithm is proposed for efficient allocation of the resources so as to maximize the profit of the providers and to minimize the cost for the customer. This MMS system also attempts to establish better network among the CSPs and gives better QoS through optimized resource allocation. This algorithm enables fast response time in comparison with switching over to different providers. In addition, this approach is highly scalable and the deployment can be accomplished in a reasonable time.

However, the experiments and implementation process could not be carried out as there is no federated system existing in the present scenario. For our future work, we plan to carry out the implementation of the MMS system to increase the threshold value so as to enhance optimized usage of the resources and to increase the customer satisfaction level. In the proposed system, there are still factors that can be looked into. For example, SLA between the providers, billing and accounting of the CSPs for each others' service provided could be still future tasks for a complete architecture.

REFERENCES

[1] L. Kleinrock, "A vision for the internet," *ST Journal of Research*, vol. 2, no. 1, pp. 2-5, 2005.
[2] B. Rochwarger, D. Breightgand, E. Levy, A. Galis, and A. Nagin, "The RESERVOIR model and architecture for open federated cloud computing," *A Project Paper*, 2010.
[3] L. Wu, S. K. Garg, and R. Buyya, "SLA – Based admission control for a software-as-a-service provider in cloud computing environments," *Journal of Computer and System Sciences*, vol. 78, pp. 1280-1299, 2012.
[4] A. V. Dastjerdi and R. Buyya, "An autonomous reliability – aware negotiation strategy for cloud computing environments," in *Proc. the Twelfth IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, IEEE Computer Society, pp. 284-291, 2012.
[5] L. M. Vaguero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45-52, 2011.
[6] Q. Zhang, L. Cheng, and R. Boataba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1 Springer, pp. 7-18, 2010.
[7] B. P. Rimal, A. Jukan, D. Katsaros, Y. Goeleven, and G. Yves, "Architectural requirements for cloud computing systems: An enterprise cloud approach," *Springer Science-Business Media*, pp. 3-26, 2010.
[8] V. C. Emeakaroha, M. A. S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. F. De Rose, "Towards autonomic detection of SLA violations in cloud infrastructures," *Future Generation Computer Systems*, Elsvier, pp. 1-13, 2011.
[9] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-Oriented federation of cloud computing environments for scaling of application services," Springer-Verlag Berlin Heidelberg, C. H. Hse *et al.*, Eds., pp. 13-31, 2010.
[10] V. K. Garg, *Elements of Distributed Computing*, Wiley-IEEE Press, , 2002.
[11] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of High Performance Computer Application*, vol. 15, no. 3, pp. 200-222, 2001.

[12] K. W. Park and J. Chung, "THEMIS: A mutually variable billing system for the cloud computing environment," *IEEE Transactions on Service Computing*, pp. 1-14, 2012.
[13] L. Wu and R. Buyya, "Service Level Agreement (SLA) in utility computing systems," *IGI Global: Information Science Reference (ISR)*, pp. 1-25, 2012.
[14] J. W. Ross and G. Westerman, "Preparing for utility computing: The role of IT architecture and relationship management," *IBM Systems,* vol. 43, no. 1, 2004.
[15] H. M. Kyi and T. T. Naing, "Stochastic markov model approach for efficient virtual machines scheduling on private cloud," in *Proc. the International Journal on Cloud Computing Services and Architecture (IJCCSA)*, vol. 1 no. 3, pp. 1-13, 2011.
[16] R. Buyya, S. K. Garg, and R. N. Calhairos, "SLA-Oriented resource provisioning for cloud computing: challenges, architecture, and solutions," in *Proc. the IEEE International Conference on Cloud and Service Computing*, pp. 1-10, 2011.
[17] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Werfield, "Xen and the art of virtualization," in *Proc. Nineteenth ACM Symposium on Operating Systems Principles*, 2003.
[18] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier, "Cluster-based scalable network services," in *Proc. the Sixteenth ACM Symposium on Operating Systems Principles*, 1997.
[19] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, and E. Turruni, "QoS – Aware clouds," in *Proc. the IEEE Third International Conference on Cloud Computing, IEEE Computer Society*, pp. 321-328, 2010.
[20] R. Buyya, C. Shin, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype and reality for delivering as the fifth utility," *Future Generation Computer System*, vol. 25, pp. 599-616, 2008.
[21] B. Mejias and P. V. Roy, "From mini clouds to cloud computing," *A SELFMAN Project Paper*, pp. 1-5, 2010.
[22] L. Mei, W. K. Chan, and T. H. Tse, "A tale of clouds: Paradigm comparisons and some thoughts on research issues," in *Proc. the IEEE Asia-Pacific Service Computing Conference,* pp. 464-469, 2008.
[23] R. Buyya, S. Pandey, and C. Vecchiola, "CloudBus toolkit for market-oriented cloud computing," *Springer-Veerlag Heidelerg*, M. G. Jaatun, G. Zhao, C. Rong, Eds. Cloudcom, LNCS 5931, pp. 24-44, 2009.

L. Arockiam is working as an associate professor in the Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 25 years of experience in teaching and 16 years of experience in research. He has published 1 research articles in the international/national conferences and journals. He has also presented 2 research articles in the Software Measurement European Forum in Rome and 1 article in international conference held at Bali, Indonesia. He received "Best Academic Researcher of 2012" award from ASDF Techno Forum Group, Pondicherry. He has chaired many technical sessions and delivered invited talks in national and international conferences. He has authored a book on "Success through Soft Skills" and "Research in Nutshell". His research interests are software measurement, cognitive aspects in programming, data mining, mobile networks, and cloud computing.

A. Stanislas is doing his Ph.D. in computer science in St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. Prior he received his MCA at Loyala College, Chennai. He worked as a lecturer in St. Xavier's College, Dumka, Jharkhand. He has attended many national and international conferences and workshops, presented papers and published a few papers. He has also delivered a few guest lectures on green and cloud computing in seminars. His research area includes networking, green computing and cloud computing.