A SURVEY ON THE CAPABILITIES OF CLOUD SIMULATORS

Vikas Lokesh[1], Srivathsan Jayaraman[2], Aishwarya N[3], Alta Soni[4], Dr. H S Guruprasad[5]

[1,2,3,4]UG Student, Dept. of CSE, BMSCE, Bangalore

[5]Professor and Head, Dept. of CSE, BMSCE, Bangalore,

[1]vikaslsetty@gmail.com, [2]srivathsan.jayaraman@gmail.com, [3]aishwarya.nov08@gmail.com

[4]alta93@gmail.com, [5]drhsguru@gmail.com

Abstract - Numerous cloud simulator tools and frameworks have been developed to aid the simulation of cloud environments in order to test any newly proposed algorithm, model or concept without having to incur the cost of deploying the same on an actual cloud infrastructure. These cloud simulation frameworks are documented and well-illustrated with examples by their respective authors and there exists several such survey papers which delineate and differentiate the features of these frameworks. In this paper, however, we cite some of the recent researches taken place in the world of Cloud Computing where some of these cloud simulators were made use of. It can be observed that although most of the cloud simulators and frameworks have similar architectures and functions, they considerably differ when comes to capability and extensibility. It is also observed that few Cloud Computing concepts cannot be satisfactorily simulated by any of these simulators.

Keywords - Cloud Computing, Simulators, CloudSim, iCanCloud, GreenCloud, VM Migration, Federated Date Center

INTRODUCTION

In recent times, cloud computing has moved from its nascent stages and into the main spotlight, with several products today delivered solely via cloud. Be it Software as a Service (SaaS), Infrastructure as a service (IaaS), or Platform as a service (PaaS), cloud computing has exceeded expectations in delivering reliable, fault-tolerant, secure, scalable, and sustainable computational services. These services are offered commercially as public clouds, although personal use clouds called private clouds are also available, and a mixture hybrid cloud is also possible.

As cloud technologies keep evolving to more dynamic and energy efficient solutions, we face the problem of finding an adequate testing environment (called a testbed). Testing cloud applications directly in a datacenter would be a very expensive process that is fraught with risk, and also time consuming in the resource acquisition process. Live testing is also prone to non reproducible errors. However, testing is a vital process for any large scale application, and especially so with cloud applications.

A Cloud simulation environment is a framework against which applications that are deployed with a cloud backing can be tested and debugged. Cloud simulation tools provide an excellent way to test cloud applications. Testing the application in a simulation environment would be an inexpensive way to test all the facets of the application including server load, responsiveness, and failure response Cloud computing has the issue of being a service as opposed to a product, so continuous testing on a live testbed would incur severe costs, and also may not be repeatable. Simulation tools provide a research environment with repeatable results, without any physical server at hand. It also allows for testing different QoS (Quality of Service) bottlenecks within the cloud. These experiments will greatly increase the productivity of any cloud based service. Without these tools, patrons would have to either acquire a live testbed to run experiments, which is an incredibly expensive and non-experimental environment, or use imprecise approximations, which would hurt their bottom line. Some of the popularly known cloud simulators are CloudSim, Cloud Analyst, GreenCloud and ICanCloud.

LITERATURE SURVEY

1. Resource Management

Olivier Beaumont et al [16] deal with assigning a set of clients with demands to a set of servers with capacities and degree constraints. Their goal is to maximize the overall throughput achieved using any cloud based platform, also denoted by the number of tasks that can be processed within one time unit. In this scenario, the degree constraint is related to the maximum number of TCP connections that a server can handle using quality of service, while its capacity is defined as its overall outgoing bandwidth. They delve into the Maximize-Throughput-Bounded-Degree problem, which is defined as maximizing the capacity of the server allocated by the server to the client, under three conditions: where the sum of all server space allocated is less than the total capacity, degree of the server, and server space itself. They finally prove that MTBD is NP-Complete, that no fully online MTBD can achieve a constant ratio, and that it is possible to maintain an optimal solution at the cost of 4 changes per server each time a new node joins or leaves. While standard cloud simulators such as CloudSim have no heterogeneous resource allocation protocols, specialized cloud simulators such as DynamicCloudSim have out-of-box support for these protocols. However, they do not handle this specific instance of the MTBD problem, as described in the paper.

Cloud Platforms deal with myriad resources, in very large scales; whether it is power, for cooling and running the datacenter, or actual physical servers, used to deploy cloud applications, managing these resources is generally a difficult and expensive task. Most approaches to resource management of large cloud platforms are centralized, with one main controller managing all the resources of the cloud. Bernadetta Addis et al [17] propose a distributed hierarchical approach, based on a mixed-integer nonlinear optimization of resource management across multiple timescales. They first present the working environment of a PaaS, and then proceed to formulate optimization problems of resource management at multiple timescales. They give a solution to the three cases presented in the optimization problem, and compares its algorithms effectiveness via rigorous mathematical treatment. Thus, the efficacy of a distributed hierarchical approach to resource management, over a standard centralized approach, is shown. Currently, most cloud simulators use a centralized resource manager. While a centralized resource manager can be effective for small to medium size clouds, we see the need for a distributed resource manager, such as the one proposed above, in large clouds. Simulators such as CloudSim, ICanCloud, etc, are extensible enough to implement this type of provisioning.

Resource selection in a cloud environment is a very important task that deals mainly with quality of service agreements. QoS may be defined through task deadlines, where there is no direct relation to the time period a task uses a resource and the cost of using the resource; through a tasks start time, duration, and dependencies on other tasks, which is generally infeasible to compute dynamically, or through their requested start and finish times. In the latter, we face a problem of a time interval between the task starting, and its requested start time, due mainly to tasks competing with each other for resources. Nikolas D. Doulamis [18] proposes a methodology to deal with the aforementioned problem. He uses a resource assignment algorithm that does not eliminate overlap of task schedules; rather, it minimizes it, so there are no hard constraints upon tasks. They call this a "Soft Interval Scheduling Problem". They move on to presentation of the problem statement, and the algorithm developed (SCS). Finally, they analyse and contrast SCS to several existing algorithms, such as ECT and Maxima IS.

No cloud simulator offers out of box support for this style of resource selection. Most simulators, however, can be extended to implement this type of resource selection. Currently, most resource selection algorithms only seek to eliminate the overlap of task schedules ("Hard Interval Scheduling Problem"). The SCS algorithm can be adopted by cloud simulators to optimize resource selection for tasks.

Virtual Machines are allocated to applications, but sometimes it is difficult to predict the resources required by a certain application. It could be random or dynamically changing due to various reasons. Allocating more than a needed amount of resources or “over-provisioning” is also detrimental to the system since the resources will only be properly used during peak requests and then will stay under-utilized. Various solutions to this problem, such as Live migration algorithm, Gray Box/Black Box, VectorDot and offline bin-packing algorithms, and their shortcomings have been discussed, and as a result, Weijia Song et. al. [27] introduce a new idea, Variable Item Sized Bin Packing algorithm. Each server is a bin and each VM is an item to be packed. While other bin packing algorithms focus on fully packing the bins, VISBP recognizes the fact that running the servers at full utilization is damaging to the system. VISBP also comes out stronger in terms of green computing, scalability of resources and load balancing.

Dynamic resource allocation, green computing and VM migration are all supported by GreenCloud, CloudSim and DynamicCloudSim. They can be extended to include functionality of VISBP algorithm as this algorithm is a new idea and none of these simulators provide out of box support for it at the moment.

Shikharesh Majumdar et al [34] start off by introducing distributed system infrastructures such as Grids and the more recent Clouds that have increasing popularity among users and researchers. Both these infrastructures can lead to significant savings for the system users as they provide the ability to obtain resources on demand to provide flexibility in resource usage and pay-as-you-go opportunities. For utilizing the power of the underlying distributed resource infrastructure, the middleware provides an efficient management of resources used by service providers. Currently, the problems faced are security during resource access, resource heterogeneity, efficient and effective resource allocation and scheduling. The authors describe the solutions to challenges faced with resource management, especially resource allocation and scheduling. Methods for increasing speed of responses for improving user satisfaction and service provider benefits are discussed.

The cloud has numerous components from different vendors which makes the infrastructure complex. The applications in cloud interact amongst themselves or other deployed applications. Application dependencies and complex infrastructure creates an environment that has to tackle careful management with security and privacy issues. A cloud scheduler manages allocation of virtual resources for the physical resources in the infrastructure. However, the current cloud scheduler implementations do not consider the overall user and infrastructure properties resulting in security and privacy concerns. Thus, Imad M Abbadi et al [37] introduce a cloud scheduler with an advantage which provides user requirements and infrastructure properties that have not been considered in commercial schedulers. The prototype is built on OpenStack, which targets on promising the users that their virtual resources are hosted without them understanding details of the complexity of the cloud infrastructure. Finally, an implementation is provided of previous work on cloud trust management in which the scheduler is provided with trust status of the cloud infrastructure as input. The proposed solution can be implemented in CloudSim.

Resource Management is a major problem across cloud networks; it entails several problems such as minimizing cost of operation, equitable access through load balancing, and diverse resource requirements of VMs by different clients. As the infrastructure of a cloud is managed wholly by one operator, who relies on a single resource management substrate, further requirements include the substrate being general, expressive, and fair. Finally, the algorithm managed to allocate resources must also be efficient. Hong Xu [11] proposes "Anchor", which is a resource management framework. This framework decouples policies from mechanisms, which is not found in current commercially available resource management tools; this enables ISPs to freely create any policy they wish, with the assurance that the mechanism will accommodate them. Anchor consists of three parts: a policy manager, a resource monitor, and a matching engine. The paper then uses Anchor to design a fair and expressive mechanism, based on the stable matching framework from general economics. Finally, a rigorous performance evaluation of the framework is performed, using a prototype implementation on a 20 node server cluster, and large scale simulations. Currently, no simulator exists that provides decoupling between mechanisms and protocols. The author proposes Anchor as the first framework to do so. Cloud simulators can possibly implement Anchor in their own frameworks to allow for decoupled mechanisms and protocols.

As an increasingly large number of organisations are moving to the cloud, the reliability of the cloud infrastructure becomes extremely crucial. The complete execution of tasks submitted to the data centre despite server crashes is of utmost importance. A. Zhou et al [2] propose that this can be achieved by the mechanisms of replication and checkpointing. Replication is a mechanism where the same task is allocated to several VMs simultaneously and ensures that at least one replica of the task is executed on time. Since replication obviously incurs redundancy, the mechanism of Checkpointing is employed. The Checkpointing mechanism periodically saves the state of the task as a checkpoint image file. If the server crashes, then another server takes up the same task and resumes execution based on the task's checkpoint image file. The extensibility feature of CloudSim is largely employed to implement this functionality. Modules such as the fat-tree data center network construction modules, failure and repair event trigger modules, checkpoint-based service recovery modules are added onto CloudSim to create a new cloud simulator FTCloudSim.

2. Resource Provisioning

Currently, the cost of hosting a high scale data center is incredibly high. More than half the power and cooling infrastructure cost is committed to the server hardware alone. Current solutions to this problem include virtualization-based consolidation, to combat server sprawl and to provision elastic resources, and statistical multiplexing, which allows the sum of the peak resource demands of each user exceeding the capacity of a datacenter. By leveraging differences of heterogeneous workloads (workloads of different natures, such as web browsing, streaming etc) and performance goals, a co-operative resource provisioning solution is proposed to decrease the peak resource consumption of workloads on data centers. It involves the four main heterogeneous workloads: Parallel Batch jobs, MapReduce jobs, Web Servers, and Search Engines. Jianfeng Zhan et al [13] present a system called "PhoenixCloud", to enable cooperative resource provisioning for heterogeneous workloads. The authors perform a rigorous analysis of PhoenixCloud for this problem.

DynamicCloudSim already has support for heterogeneous resource allocation; however, heterogeneous workloads are currently not directly supported in cloud simulators. Simulators have to be extended with custom logic to include algorithms such as the one described in the paper. DynamicCloudSim is the closest proponent of this.

There is a huge amount of energy consumption by data centers for cooling and power distribution. One solution for this problem is Dynamic capacity provisioning. However, this method does not look at the problem of heterogeneity of workloads and physical machines. Data centers usually consist of heterogeneous machines with varying capacities and energy consumption. Qi Zhang et al [35] first analyze workload traces from Google's production compute clusters. Due to previous drawback, Harmony is designed as a dynamic capacity provisioning (DCP) framework, which considers workload and machines and has a balance between energy savings and scheduling delay. Directly solving DCP is not possible. Therefore, two technical solutions are provided. Finally, Google workload is evaluated with the proposed systems to conclude that it results in energy savings and also significantly improves task scheduling delay. Open source platforms such as Eucalyptus can adopt this mechanism by changing the scheduling policy to weight round-robin first fit and weight round-robin best fit. Most simulators can be extended to include support for a distributed resource manager.

3. Resource Allocation

Business and science has led to large scale computing, storage, and network capabilities in the grid/cloud networks. Grid/cloud enables users to execute tasks on centralized computing and storage facilities instead their local systems, which requires efficient resource scheduling methods for resource allocation in grids/clouds, to improve resource utilization and reduce cost of scheduling. Pan Yi et al [40] emphasize on combined resource allocation in the grid/cloud environment. Also, optical network architecture is used for reserving network bandwidth. A mixed integer linear programming (MILP) model and heuristics are developed with different job scheduling methods to solve the bandwidth guaranteed optimal joint resource scheduling problem. Experiments are carried out with several network topologies to prove that MILP and heuristics work to solve the problem, but MILP is time consuming. Tabu search also provides most optimal resource allocation unlike best-fit method.

No simulator offers out of box support, but simulators such as CloudSim can be extended to include this functionality. All the standard rule based algorithms like FCFS, Round-Robin, Early start time job first (ESTF) and Shortest Job First Scheduling algorithm along with the newly proposed two heuristics- best-fit method and tabu search have been successfully implemented.

In order to enable users to access stored data and applications on an as-needed basis, cloud computing develops a fluid pool of virtual resources across computers, server stacks as well as whole data centers. The target end users are provided with upgraded communication and computation services by shifting from traditional data-center oriented models to distributed clouds extend over a loosely coupled combined substrate of resources. For efficient realization of networked computing environments, networking and computing resources need to be coupled and optimized. Chrysa Papagianni at al [32] initially devise the optimal networked cloud mapping problem as a mixed integer programming (MIP) problem, indicating goals related to cost efficiency of the resource mapping procedure, while conformed to by user requests for QoS-aware virtual resources to provide an integrated resource allocation framework for networked clouds. A method is proposed for the effective mapping of resource requests onto a shared substrate interconnecting various computing resources. The flexible, structured, and comparative performance evaluation is represented in a simulation/emulation environment. The authors conclude their observations by showing a proof-of- concept understanding of their proposed schema which is set up over a resource virtualization platform with networking and computing facilities, the European future Internet test-bed FEDERICA. The paper utilizes CV I - Sim, a simulator that allows for an organised and flexible performance evaluation of the performance and efficiency of the proposed approach.

Virtual Machines are a mechanism used to multiplex virtual resources onto physical hardware, where the machine is mapped onto physical hardware so that several tasks can be run concurrently on a single physical machine. However, deciding the mapping dynamically so that physical machines are minimized and resource demands are met correctly is a policy issue. Zhen Xiao et al [14] propose an automated resource management system that balances overload avoidance (where the demands of the virtual machine should not exceed the capacity of the physical machine) and green computing (where we use a minimal amount of physical machines to satisfy the demands of all virtual machines). They also introduce "skewness", which is a measure of how unevenly a server is utilized. The algorithm works by calculating the skewness of a physical machine, and maximizing hot spot mitigation. When a server is running very hot, VMs are migrated away from the server unto another server. This achieves the green computing aspect of the algorithm. In conclusion, this algorithm proved to be very effective in achieving dynamic resource allocation in accordance to energy saving and resource saving constraints.

VM migration and Energy conscious Datacenter models are implemented in CloudSim. CloudSim, and other simulators such as GreenCloud, can be extended to include the aforementioned automated resource management system; as it were, CloudSim already implements Dynamic Frequency and Voltage Scaling.

4. Resource Procurement

Cloud infrastructure (IaaS) providers are facing a growing problem where a substantial portion of their cloud resources are being left over after the initial direct-sell process. In order to prevent this loss, these left over or spare resources can be auctioned off to the highest bidder. Since users require a bunch of resources rather than a single type of resource, this resource allocation process can prove to be NP-Hard. Paolo Bonacquisto et al [3] propose a combinatorial algorithm to facilitate the allocation of spare resources to the highest bidder in a virtual procurement auction market.

CloudSim is used to simulate this functionality. A new component called *Auctioneer* has been introduced with interacts closely with the Broker to allocate spare resources. The *DataCenter* class is extended to include additional functionality such as reserving resources, estimating bids and implementing the overbooking mechanism.

Currently, most cloud vendors use a "pay as you go" model as a fixed pricing strategy for their resources, without allowing their users to pick a resource procurement model that caters to their own needs. Abhinandan S. Prasad et al [9] propose 3 mechanisms for a QoS bottleneck in resource procurement; specifically, it allows for dynamic pricing by the cloud vendors, according to the needs of the user, instead of a fixed pricing model. The first mechanism is C-DSIC (Cloud-Dominant Strategy Incentive Compatible), where the user pays the price as per the next lowest bid in the Vickrey Auction. C-DSIC may be preferred when all cloud vendors distribute their price and QoS with the same probability. The second mechanism is C-BIC (Cloud-Bayesian Incentive Compatible), where each cloud vendor contributes a participation fee, which is used to pay other cloud vendors. C-BIC is preferred for government organizations. The last mechanism discussed is C-OPT (Cloud-Optimal), where the virtual cost of each cloud vendor is used to determine the winner, contrasting with the use of the ratio of cost and QoS, as in C-DSIC and C-BIC.

These mechanisms may be implemented by use of a cloud brokerage service. Parameters such as price and Quality of Service constraints are not present in simulators like Eucalyptus. In CloudSim, while both price and quality of service are supported via utilization models, there exists no support for auction protocols. This paper was tested on a custom framework built by the authors.

5. Resource Sharing

Tian Wenhong et al [24] start off by listing out the various advantages of cloud computing in our fast paced world, such as reduced costs, sharing, hiding complexity etc. But most cloud computing platforms' infrastructure is hidden to anyone who would like to research it. The strong need for platforms that support experimentation for research or learning purpose is met with the help of a Platform-as-a-Service. They go on to propose an architecture for the CRESS platform and the various modules and functions within it. The operating environment is described as having one super scheduling centre (a high performance server) and multiple other data centres (Physical clusters with virtual software). This platform is evaluated and the various applications with respect to networking, cloud storage, elastic web service and simulation as well as benefits with respect to time, cost and customization are laid out.

These authors have chosen CloudSim as their simulation tool. Doing so has encouraged understanding of how cloud computing works, and has provided a way to evaluate the effects and performance of the various scheduling and allocation algorithms present in CloudSim infrastructure.

Currently, many clouds provide services such as storage and computing. Demand for scalable resources has been increasing rapidly as cloud customers are charged only for the services they use. However, a single cloud may not have sufficient resources or idle resources are not fully utilized. Therefore, with increasing demand, collaborative cloud computing (CCC) has been introduced, in which scattered resources belonging to different entities are collectively used to provide services. The issues of resource/reputation management are addressed to guarantee successful deployment of CCC. Procedures used before for resource and reputation management were not efficient. Haiying Shen et al [38] propose an integrated platform called Harmony. Considering the interdependencies between resource/reputation management and for efficient and trustworthy resource sharing, Harmony combines three components-Integrated Multi-Faced Res/Rep Management, Multi-QoS-Oriented Resource selection, Price-Assisted Resource/Reputation Control. These three components enhance the reliability of globally scattered distributed resources in CCC. Verification of the different components show that Harmony performs better than existing resource/reputation management systems in terms of high scalability, balanced load distribution, loyalty awareness, QoS, effectiveness. Federated data center provides basic implementation of cloud computing. The authors have proposed an integrated platform called Harmony. Harmony combines three components that enhance the reliability of globally scattered distributed resources in CCC. For the validation of the proposed approach, a numerical simulation is conducted for critical situations.

In a cloud environment, a cloud provider faces a major problem in provisioning the VMs on demand to clients; as workload spikes are erratic and unpredictable, neither over-provisioning nor guaranteeing a limited number of clients access can solve the issue of rejection of a client due to unavailability of a VM. This paper proposes "Federated Clouds", which allows cloud providers to share their resources when they are not needed and request and obtain extra resources during high-demand periods, which allows them to successfully provide continuous service to all clients, with several proposed schemes for capacity sharing in the federation. Nancy Samaan [10] approaches the problem of sharing unused VMs between cloud providers from a game theoretic standpoint; where each cloud provider is assumed to be a rational agent intent on maximizing its payoff. She then proceeds to show the existence of a Nash Equilibrium in the system, and thus derives schemes using "self-enforceable cloud providers" for sharing based on this environment. Thus, by reducing the problem statement to a dynamic programming problem, she finally uses a recursive formulation to effectively reach the solution. CloudSim already provides an implementation similar to what the author proposes. A class FederatedDataCenter.java exists, which provides a Federated Data Center model for use in simulations. Other simulators can also be extended to include this functionality.

6. Cloud Brokerage

Foued Jrad et al [26] start off by introducing the concept of scientific workflows: the process of breaking up a certain compute-intensive task into multiple portions each being solved by a computing unit. While the workflow technology has been explored deeply on infrastructures such as the Grid Model, the growth in the field of cloud computing has opened up doors for workflows as an answer to issues such as scalable computational resources and storage. Currently, however, workflows have only been migrated to a single cloud. There are quite a few applications which have much larger requirements than what can be accommodated by a single cloud, but there isn't any protocol put in place that will enable these applications to run in a multi cloud environment as of yet. Thus, the authors have taken it upon themselves to build this framework along with a Cloud service broker that chooses the appropriate cloud based on the users requirements and the cloud's capacity. Apart from the framework, they have also developed their own cloud simulation environment to test the workflows on the framework and evaluate its performance. The authors evaluate its workflow execution features by implementing a simulation environment based on the CloudSim toolkit. The simulation environment has CloudSim Intercloud Gateway that provides for a common interface to reach the datacenters or Clouds. Management of large scale workflows is done with the help of WorkflowSim. WorkflowSim is responsible for mapping "abstract workflows to concrete workflows" using a Workflow Mapper. It deals with the organization of tasks and data flow dependencies using Workflow Engine, and also minimizes the quantity of tasks by merging them using the Clustering Engine.

Cloud applications are an amalgamation of services that need to be sifted through carefully to obtain the required services. Many small scale businesses do not have the expertise required to perform these "knowledge-intensive" decisions and thus are not able to obtain all of the benefits that the cloud has to offer, such as reduced expenditure and scalability. An application these days is never made to be permanent and unchangeable, because there are always new services cropping up with cheaper performance costs and higher functionality, which can easily replace the older ones. A "man-in-the-middle" would prove valuable to a company which needs to properly evaluate various offerings of the cloud based on its particular needs. This so called "broker" could also be in charge of application execution. A broker needs to also be able to handle service failures (adaptation) and continuously evolve the applications (optimization). While these are both very important aspects of cloud applications, they are also quite costly to handle by the broker. So, Gein Horn [22] introduces a few platforms and tools that the broker could use to help with the quality assurance of these applications.

Horn's definition of Cloud Brokerage is a brand new idea in the field of cloud computing, and as such, there exist no simulators at present which can handle the broker function proposed here. While Eucalyptus cannot handle QoS and price, CloudSim could be extended to include this functionality into its Broker function.

In the traditional Cloud Computing Architecture, Brokers are responsible for the allocation of cloudlets to designated datacenters and for providing applications to the end user from Clouds. In order to manage Broker Cloud Communication and to reduce the response time of application services, an efficient management infrastructure is required. A system, which works in Broker Cloud Communication Paradigm (BCCP), an Efficient Broker Cloud Management (BCM), has been introduced by Gaurav Raj [31] to find a communication link with the least cost of link uses between broker and cloud. In order to find an optimum route between broker and cloud, an algorithm namely, Optimum Route Cost Finder (ORCF) has been proposed. In order to help in executing Cloudlets over Virtual Machines to select best policy for BCM System, several VM Allocation and VM scheduling policies are examined. In addition, the processing cost and total execution cost on the bases of Hops Count, Bandwidth, Network Delay or Combined Approach are analyzed. The author also provides the analysis of the link capacity by using Bandwidth-Delay Product (BDP). ORCF helps in identifying the total execution task for broker. CloudSim toolkit is used with some modification to analyze the cost optimization in broker cloud communication.

7. Resource Scheduling

Numerous rule-based scheduling algorithms are available to schedule tasks to available computer resources such as FCFS, Round-Robin, Min-Min, Min-Max and Shortest Job First Scheduling. However, these algorithms are seldom used in large-scaled cloud infrastructures due to their intrinsic simplicity. Instead, heuristic algorithms such as the ant colony optimization algorithm are used for better optimization. Recent studies indicate that these heuristic algorithms, when used on their own, fall short too. Chun-Wei Tsai et al [1] propose a revolutionary new Hyper-Heuristic Scheduling algorithm which selects the best heuristic scheduling algorithm from a pool of available algorithms at every invocation. The goal is to combine standard heuristic algorithms to perform Transition, Evaluation and Determination of every job pool.

All the standard rule based algorithms like FCFS, Round-Robin, and Shortest Job First Scheduling algorithm along with the newly proposed Hyper Heuristic algorithm have been successfully implemented using a combination of CloudSim and Hadoop.

Dynamic VM Allocation in Cloud Computing is currently achieved using several contemporary VM load balancing algorithms, including Round Robin Load Balancer, where each VM gets allocated in a circular manner for a fixed quanta of time, Throttled Load Balancer, where each VM is either in a busy state or an ideal state, and the TLB sends the ID of the ideal virtual machine to the data center controller for allocation, or Active Monitoring Load Balancer, where the least loaded VM is identified and selected whenever a new resource request arrives. Bhupendra Panchal et al [15] introduce clustering as another mechanism to dynamic VM allocation. Clustering is the method of grouping similar types of objects into clusters, to minimize dissimilarity. In this algorithm, the VM list is divided into K clusters, which are then allocated to the cluster with the closest centroid. The paper then presents a sample implementation of said algorithm in CloudSim, with the results tabulated and contrasted against regular VM allocation methods. Their paper was implemented in CloudSim.

Cloud computing provides opportunities to form large scale scientific problems. Workflows are often used to model these scientific problems. Scientific workflow has a large amount of data and high requirements which demands a computing environment of high performance to execute in a given amount of time. There has been study on workflow scheduling, but only a few are designed to for cloud environments. However, the existing work fails in terms of QoS requirements, elasticity, heterogeneity of computing resources. Therefore, Alejandra Rodriguez et al [39] present a joint resource provisioning and scheduling strategy for executing scientific workflows on IaaS clouds. Implementing the meta-heuristic optimization algorithm, it is modelled to minimize the total execution cost and meet certain user-defined deadlines. The approach includes IaaS characteristics such as pay-as-you-go model, elasticity, heterogeneity and dynamicity of the resources, performance variation and VM boot time. CloudSim framework and different workflows from scientific areas are used in order to evaluate the performance of experiments conducted. Results prove that their approach is better than previous algorithms. The paper proposes a solution that can be supported by CloudSim. Workflow management is handled by workflowsim.

8. Distributed Computing

With the advent of the internet and various web applications, large sets of data can be constantly collected to improve user satisfaction during application usage. For example various searches that go into a search engine are processed to improve the search results' relevance. The analysis of such large sets of data has been made possible by a processing model known as MapReduce. Since running a private Hadoop cluster is too inaccessible, the other option is running Hadoop/MapReduce on top of a public cloud. One challenge is that it is up to the user to determine the correct amount of virtual nodes for the cluster. With respect to the monetary and time costs involved in the resource allocation optimization for MapReduce, there is a tradeoff between the amount of resources provisioned (cost) and the amount of time taken to process. Cloud RESource Provisioning (CRESP), an idea by Keke Chen et al [23] introduces two new concepts of reducing the monetary cost within limited time and reducing time costs within monetary constraints.

Once the authors set up the cost model of the algorithm they explore the resource allocation with respect to time constraints, cost constraints, and optimal tradeoff with no constraints. They demonstrate with the help of experiments and lay the foundation for future work.

There are two main issues when it comes to optimization resource allocation for MapReduce programs: the monetary cost related to VM allocation and the time cost involved to get the job done. Thus the decision problem discussed above has two parts, and the Resource Time Cost Model proposed here deals with the tradeoff of these two factors: iCan cloud - which supports trade-offs between cost and performance - can be extended to include CRESP provisioning.

Cloud Computing also provides Platform as a service (PaaS) which enables users to run MapReduce applications on virtual machines in the cloud. Due to the large amount of data handled by these applications, most of the recent research involves optimizing disk I/O operations in virtual machines to run these MapReduce applications. Eunji Hwang et al [4], however, propose a cost-effective provisioning policy of virtual machines for MapReduce applications. The proposed VM provisioning algorithms is simulated using the CloudSim Toolkit. Four Data Centres with Ten Hosts each were virtually deployed to aid the simulation.

With the exponential growth of Grid and Distributed Computing, a large number of unreliable hosts have become exceedingly common. All the more, a large number of these hosts are heterogeneous in nature. Bahman Javadi et al [5] introduce a novel way of discovering subsets of hosts with similar statistical properties and which can be modelled with similar probability distributions. This algorithm is simulated using the host availability traces obtained from a real Internet-Distributed System, namely SETI@home. The functionality is implemented using a discrete event-driven simulator. This simulator was developed using the Objective Modular Network Testbed in C++ (OMNeT++) simulation environment, which is an open source, component-based and modular simulation framework

Giacomo McEvoy et al [28] explore the results of a parallel application that has been deployed in a local Cloud, which is still under development, in contrast to deployment in a public cloud. It concentrates less on the mathematical specifics of the application, and more on the architecture and computation-intensive tasks that are being executed in a Grid-based Master-Worker model. Performance results of experiments using Xen paravirtualization on Eucalyptus Cloud Infrastructure have been documented and deemed satisfactory. It then goes on to describe the on-premises cloud, a private cloud that is free of charge for users on the LNCC network, and its advantage over public clouds. They then delve into the Nimbus Context Broker software which quickly deploys virtual clusters in a suitably compatible cloud. After listing out the various issues that this software will have to face, the authors of the paper put forward their proposal. They suggest two solutions for security issues with the Global Containers, and describe an ideal Globus based platform which "deploys Grid services on-demand in the cloud". Here, the parallel application deploys in an on-premises Cloud based on Eucalyptus software infrastructure which is used to provide IaaS. In this Master/Worker model, the Master sets up the Workers over Eucalyptus using the Nimbus Context Broker. Nimbus Context Broker is compatible with Eucalyptus cloud due to its Amazon back end.

The introduction of parallel data processing in the Cloud has insinuated an outburst in the number of Companies offering parallel data processing capabilities in cloud infrastructures (IaaS). However, traditional parallel data processing frameworks like Hadoop have been designed for static, homogenous cluster setups and disregard the heterogeneity of the typical cloud infrastructure. Hence, a new data processing framework called Nephele in introduced as a viable alternative by Daniel Warneke [6]. The author has designed his very own data processing framework for cloud environments called *Nephele.* Nephele consists of a Job Manager, Cloud Controller, VMs known as instances, and Task Managers. The Job Manager is analogous to that of Broker in CloudSim, and is responsible to schedule the tasks and allocate VMs with the help of the Cloud Controller. The Task Manager runs on VMs (also known as instances) and derives I/O from a persistent storage such as those offered by Amazon S3.

9. Social Networks in Cloud Computing

The number of wireless handheld devices has been growing at an increasing rate as there is rapid development of communication technologies. The world has moved to global wireless access from fixed Internet access, providing users with the option of watching multi-media live streaming or sharing personal media via their mobiles, laptops, and tablets. However, a challenge of Quality of service (QoS) is faced in multimedia sharing environments. Guofang Nan et al [41] introduce cloud based Wireless Multimedia Social Network (WMSN) architecture to tackle multimedia sharing and distribution, which is a heterogeneous network that consists of a multimedia cloud and other subnetworks. In the proposed WMSN, desktop users receive multimedia services from a multimedia cloud. The architecture has advantages such as network services cost savings and satisfying demands for bandwidth requirements. Game theoretical approach has been used to solve the problem of sharing bandwidth efficiently between desktop users and mobile users. Also, a cheat proof mechanism is proposed to share bandwidth on noticing the greedy behavior of mobile user. Results show best responses for the above mentioned schemes.

This paper was tested on a custom framework built by the authors. Game theoretical approach and a cheat proof mechanism have been implemented. A simulation scenario is set up for the evaluation of the performance of the proposed algorithm. The same custom framework can be implemented in CloudSim or GreenCloud.

Communication between people has changed with the rapid growth of social networking platforms. The social cloud enables users to share their services, data and resources. Simon Caton et al [36] present a concept of social compute cloud, the difficulties faced with its construction, and the architecture. Seattle is an open source peer-to-peer platform for the implementation of a social compute cloud. Different allocation algorithms are studied with respect to allocation time and economic performance in order to evaluate solutions for various social compute cloud. Applying these algorithms such that resource supply and demand do not match to a batch application model are examined. Finally, the authors show how social networks can be strengthened in cloud computing infrastructure construction and allocation of resources in the presence of user sharing preferences.

No out of box support, but simulators such as GreenCloud or CloudSim can be extended to include this functionality. Seattle is implemented in Python and the clearing house is built on the Django framework.

10. Mobile Cloud Computing

Mobile Cloud Computing has gained a lot of press for its amazing applications with resource-lacking mobile devices, but is still in its nascent stages. The resource constrained nature of mobile devices along with the demand for computation-intensive services and applications is clearly affecting the end users' satisfaction. Mobile Cloud Computing seeks to rectify these problems by applying the premise of cloud computing onto mobile devices: delegate a part of the mobile device's load to the cloud and make use of its resources and processing capabilities, instead of completely running the applications on the device itself. Fangming Liu et al [25] introduce various architectures such as cloudlets, Ad Hoc mobile clouds and an attempt at building a better architecture is made. Some challenges faced are the unpredictable nature of wireless networks, VM migration overhead, privacy and security. The authors delve into computational offloading and capability extending which are the two main uses of MCC and highlight its practical applications in the world of remote healthcare, web applications and Augmented Reality.

The authors research deals with several new resources and devices not commonly found in a cloud environment. As such, DynamicCloudSim supports heterogeneous workloads, while CloudSim and GreenCloud support live VM migration. This new style of client server offloading of tasks for mobiles can possibly be implemented in a new extension framework for the mentioned simulators.

While Mobile Cloud Computing has been a booming industry in today's era, there is the issue of the mobile devices being resource-poor. Rakpong Kaewpuang et al [29] introduce the idea of many mobile devices sharing remote and computing resources amongst themselves to support various kinds of compute-intensive mobile applications. This "pooling" of resources can be better appreciated by implementing a suitable framework for resource provisioning and allocation, which has laid out in detail by the authors. It makes uses of the cooperation or "coalition" of various mobile service providers to produce a robust optimization model (RO model) for resource management. In addition, they describe a means to facilitate revenue sharing between these providers after the revenue has been generated from supporting the application instances. Its performance has been evaluated and the results have been tabulated and explained in detail with help of various revenue graphs.

CloudSim, GreenCloud and other simulators can be extended to include RO model for resource management and revenue sharing. CloudSim already supports the FederatedDataCenter.java class, which uses a similar pooling algorithm to share resources and divide revenues effectively. The proposed algorithm can be implemented in the same simulators.

11. Special Topics

Virtual Machine Migrations

The main benefit of a cloud system lies in its flexibility and scalability; resources can be dynamically provisioned to clients at very short notice. However, optimizing the resource allocation so that each user only has the exact resources they require is a difficult procedure; this is where virtual machines come in.

They allow multiplexing the physical resources into several virtual views, so that several virtual machines can act on a physical machine simultaneously. Mayank Mishra et al [19] explore migrating virtual machines from one physical machine to the other, as a means of dynamic provisioning of resources. They delve into various migration strategies, such as "Suspend and Copy", "Pre-Copy", and "Post-Copy". Their paper then presents the nuts and bolts of VM migration, including criteria for migration, why migration is beneficial, when it is beneficial, and where to migrate a VM. It also discusses heuristics used in the migration process. Finally, it explores Wide-Area Virtual Machine Migration for HotSpot mitigation.

Live VM migration has been a field of study for some time now. Currently, several simulators offer implementations of GreenCloud and CloudSim both support live VM migration. However, several new heuristics and optimizations for VM migration are still in their infancy, and can be implemented by extending a simulator such as CloudSim.

Cloud Computing in Healthcare Applications

Yong Woon Ahn et al [20] highlight the issues faced with real-time healthcare applications with respect to the delay incurred while transmitting huge amounts of data from the sensor based medical systems (client side) to the server. Crucial real-time data is dynamic and its size fluctuates over time, which makes virtual machine allocation difficult. This Auto Scaling Algorithm does the resizing of the VMs on an "on-demand basis". This is done by detecting the "warning signals" on the sensor that might come before a large, abnormal event, and using them as a way of forecasting the amount of VMs to be allocated for the data resulting from the coming abnormal event. This algorithm has been tested out on Amazon EC2 and the results clearly show its success in allocating the correct amount of resources based on the previously mentioned "prediction algorithm" and effectively rectifying the problems faced while using other traditional scaling algorithms used for processing real-time data.

The auto scaling mechanism proposed here can be implemented in CloudSim by extending its original functionality. Also, the size of the data being processed is continually changing (dynamic), and thus the prediction algorithm used for forecasting VM allocation can be implemented in DynamicCloudSim.

Cloud-Based Vehicular Networks

Rong Yu et al [21] propose a cloud based architecture for the interconnection of vehicles. One challenge that vehicular networks face is the constraint on the amount of resources available to each one of them locally. The solution put forward is that all of the computational and storage capabilities be shared by all the vehicles in the vicinity. Hierarchical cloud architecture is introduced, with a central cloud, a roadside cloud and a vehicular cloud. The advantage of this system is that the vehicular and roadside clouds are small, which helps in the quick deployment of services. Some of the applications of vehicular networks such as, real-time navigation and shared storage for video surveillance, are highlighted. Game theoretical approach to resource provisioning, and different virtual machine migration scenarios are explored.

Several vehicle-specific scenarios relating to VM allocation have been described, and these scenarios have been dealt with by using an optimal Resource Reservation Scheme which involves VM live migration. Resource allocation through VM live migration is easily handled by simulators like GreenCloud and CloudSim. iCan cloud can be used for simulation of this newly proposed hierarchical cloud architecture, as any freshly defined components of a new cloud architecture can be added to the iCanCloud repository if and when required.

Agent-Based Cloud Computing

The cloud infrastructure, as we know, consists of several interconnected and virtualized computers dynamically provisioned to users on the basis of a Service Level Agreement (SLA) between consumers and providers. Data intensive applications need an enormous amount of resources and more often than not, these resources come from multiple providers. It therefore behoves the cloud providers to pool their resources together. However, mapping, scheduling and coordination of these shared resources prove to be an arduous task. Kwang Mong Sim [8] proposes a self-learning agent based cloud service engine to help with the initial phase, which is, appropriate resource discovery.

The author develops an agent-based cloud service search engine named *Cloudle* for discovering cloud services. The Cloudle accepts as its inputs functional as well as budgetary requirements from consumers and the underlying algorithm determines different levels of matching between the respective schedules and prices of cloud service providers and consumers. Cloudle's extensive database is aggregated by multiple cloud crawlers which extract relevant webpages from the World Wide Web.

Software-Defined Radio Clouds

Software Defined Radio (SDR) is a type of radio communication where components are typically implemented in software that executes on general purpose hardware such as amplifiers, mixers, filters, modulators/demodulators. SDR combined with cloud computing offers a revolutionary new technology for designing and managing future base stations. This provides a scalable solution for the evolution of wireless communications. Ismael Gomez Miguele et al [7] consider the resource management implications and propose a hierarchical approach for managing the real-time computing constraints of wireless communications systems which run on the Software-Defined Radio cloud.

As per our knowledge, there exists no simulation tool for the SDR Clouds. However, SDR concepts can be successfully simulated using Simulink tool. Simulink, developed in Matlab, is a graphical language tool for modelling, simulation and analysing multi domain systems. It is widely used in control theory and digital signal processing. The authors from Vellore Institute of Technology [42] have successfully developed a model of a SDR using SIMULINK tool to implement the IEEE 802.11 standard and the Bluetooth standard.

vGASA: Adaptive Scheduling Algorithm of Virtualized GPU Resource in Cloud Gaming

While various virtualization strategies have revolutionized the way cloud computing manages resources, technology still has a long way to go in terms of GPU (graphics processing unit) virtualization. The most recent VMware player is quite successful and is able to achieve more than 95 percent native performance, but there still isn't enough research done in the area of GPU resource scheduling in cloud gaming. The default resource scheduling algorithm performs very poorly which is why GPU virtualization is not used expansively. Consequently, Chao Zhang et al [30] introduce vGASA, "an adaptive scheduling algorithm for virtualized GPU resources in cloud gaming." It is a scheduler and implements three scheduling algorithms: SLA-Aware (SA) scheduling is responsible for meeting the SLA requirements for the particular VM, Fair-SLA-Aware(FSA) scheduling maximizes the usage of GPU resources to provide a better experience for the end user, Enhanced-SLA-Aware (ESA) takes care of the balance between performance and the number of users on a machine. The architecture is laid out along with the scheduling model, and the performance is evaluated based on experiments with real games.

Resource management and scheduling are basic tasks that are supported by most simulators like CloudSim and GreenCloud. However, the default scheduling algorithms are not optimal in case of virtualization of GPU resources, which is why CloudSim can be extended to include vGASA algorithm.

CloudSim Estimation of a Simple Particle Swarm Algorithm

Kavita Bhatt et al [12] introduce a simple Particle Swarm Optimization algorithm for cloud networks. This algorithm is implemented using the CloudSim cloud simulation framework. They then seek to use the Particle Swarm Optimization algorithm to search for a specific cloudlet within the environment. Particle Swarm Optimization is effective in this respect due to its features such as good convergence rate, its cheapness, its easy applicability, and simple implementation. The authors use CloudSim itself to implement a PSO algorithm. An inbuilt function to enable PSO for searching for cloudlets can be included in most major cloud simulators easily.

CONCLUSION

In conclusion, we observe that CloudSim is the most commonly used simulator by most of the researchers. However, this framework requires a working knowledge of Java and is thus not the most optimal and easy to use solution that could be deployed. The need for an all-inclusive cloud simulator which seeks to reduce human effort to a minimum with a simple Graphical User Interface (GUI) thus becomes evident.

ACKNOWLEDGEMENTS

The work reported in this paper is supported by the college [BMSCE, Bangalore] through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

REFERENCES:

1. Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, Chu-Sing Yang, "A Hyper-Heuristic Scheduling Algorithm for Cloud", IEEE Transactions on Cloud Computing, Volume 2, Issue 2, April 2014, pp 236-250, DOI: 10.1109/TCC.2014.2315797.
2. Zhou, S. Wang, Z. Zheng, C. Hsu, Michael R. Lyu, F. Yang, "On Cloud Service Reliability Enhancement with Optimal Resource Usage", IEEE Transactions on Cloud Computing, Issue 99, November 2014, pp 1, DOI:10.1109/TCC.2014.2369421.
3. Paolo Bonacquisto, Giuseppe Di Modica, Giuseppe Petralia, Orazio Tomarchio, "A procurement auction market to trade residual Cloud computing capacity", IEEE Transactions on Cloud Computing, Volume PP, Issue 99, November 2014, pp 1, DOI:10.1109/TCC.2014.2369435.
4. Eunji Hwang, Kyong Hoon Kim, "Minimizing Cost of Virtual Machines for Deadline-Constrained MapReduce Applications in the Cloud", ACM/IEEE 13th International Conference on Grid Computing (GRID), September 2012, pp 130-138, DOI: 10.1109/Grid.2012.19.
5. Bahman Javadi, Derrick Kondo, Jean-Marc Vincent, David P. Anderson, "Discovering Statistical Models of Availability in Large Distributed Systems: An Empirical Study of SETI@home", IEEE Transactions Parallel and Distributed Systems, Volume 22, Issue 11, January 2011, pp 1896-1903, DOI: 10.1109/TPDS.2011.50.
6. Daniel Warneke, Odej Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", IEEE Transactions on Parallel and Distributed Systems, Volume 22, Issue 6, February 2011, pp 985-997, DOI:10.1109/TPDS.2011.65
7. Ismael Gomez Miguelez, Vuk Marojevic, Antoni Gelonch Bosch, "Resource Management For Software-Defined Radio Clouds", IEEE Micro, Volume 32, Issue 1, September 2011, pp 44-53. DOI:10.1109/MM.2011.81
8. Kwang Mong Sim, "Agent-Based Cloud Computing", IEEE Transactions Services Computing, Volume 5, Issue 4, October 2011, pp 564-577, DOI: 10.1109/TSC.2011.52.
9. Abhinandan S. Prasad, Shrisha Rao, "A Mechanism Design Approach to Resource Procurement in Cloud Computing", IEEE Transactions on Computers, Volume 63, Issue 1, May 2013, DOI: 10.1109/TC.2013.106.
10. Nancy Samaan, "A Novel Economic Sharing Model in a Federation of Selfish Cloud Providers", IEEE Transactions on Parallel and Distributed Systems, Volume 25, Issue 1, pp 12-21, January 2013, DOI: 10.1109/TPDS.2013.23.
11. Hong Xu, Baochun Li, "Anchor: A Versatile and Efficient Framework for Resource Management in the Cloud", IEEE Transactions on Parallel and Distributed Systems, Volume 24, Issue 6, October 2012, DOI:10.1109/TPDS.2012.308.
12. Kavita Bhatt, Mahesh Bundele, "CloudSim Estimation of a Simple Particle Swarm Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013, ISSN: 2277 128X.
13. Jianfeng Zhan, Lei Wang, Xiaona Li, Weishong Shi, Chuliang Weng, Wenyao Zhang, Xiutao Zang, "Cost-Aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers", IEEE Transactions on Computers, Volume 62, Issue 11, pp 2155-2168, May 2012, DOI: 10.1109/TC.2012.103.
14. Zhen Xiao, Weijia Song, Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", IEEE Transactions on Parallel and Distributed Systems, Volume 24, Issue 6, pp 1107-1117, September 2012, DOI: 10.1109/TPDS.2012.283.

15. Bhupendra Panchal, R. K. Kapoor, “Dynamic VM Allocation Algorithm using Clustering in Cloud Computing”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013, ISSN: 2277 128X.
16. Olivier Beaumont, Lionel Eyraud-Dubois, Christopher Thraves Caro, Hejer Rejeb, “Heterogeneous Resource Allocation under Degree Constraints”, IEEE Transactions on Parallel and Distributed Systems, Volume 24, Issue 5, pp 926-937, June 2012, DOI: 10.1109/TPDS.2012.175.
17. Bernadetta Addis, Danilo Ardagna, Barbara Panicucci, Mark S. Squillante, Li Zhang, “A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms”, IEEE Transactions on Dependable and Secure Computing, Volume 10, Issue 5, pp 253-272, January 2013, DOI: 10.1109/TDSC.2013.4.
18. Nikolaos D. Doulamis, Panagiotis Kokkinos, Emmanouel (Manos) Varvarigos, “Resource Selection for Tasks with Time Requirements Using Spectral Clustering”, IEEE Transactions on Computers, Volume 63, Issue 2, pp 461-474, September 2012, DOI: 10.1109/TC.2012.222.
19. Mayank Mishra, Anwesha Das, Purushottam Kulkarni, Anirudha Sahoo,” Dynamic Resource Management Using Virtual Machine Migrations”, IEEE Communications Magazine, Volume 50, Issue 9, pp 34-40, September 2012, DOI: 10.1109/MCOM.2012.6295709.
20. Yong Woon Ahn, Albert M. K. Cheng, Jinsuk Baek, Minho Jo, Hsiao-Hwa Chen, “An Auto-Scaling Mechanism for Virtual Resources to Support Mobile, Pervasive, Real-Time Healthcare Applications in Cloud Computing”, IEEE Network, Volume 27, Issue 5, pp 62-68, September 2013, DOI: 10.1109/MNET.2013.6616117.
21. Rong Yu, Guangdong, Yan Zhang, Stein Gjessing, Wenlong Xia, Guangdong, Kun Yang, “Toward Cloud-Based Vehicular Networks with Efficient Resource Management”, IEEE Network, Volume 27, Issue 5, pp 48-55, September 2013, DOI: 10.1109/MNET.2013.6616115.
22. Geir Horn, “Cloud Brokerage: A Key Enabler for Widespread Cloud Usage”, Second Nordic Symposium on Cloud Computing & Internet Technologies, pp 6-6, DOI: 10.1145/2513534.2513536.
23. Keke Chen, James Powers, Shumin Guo, Fengguang Tian, “CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds”, IEEE Transactions on Parallel and Distributed Systems, Volume 25, Issue 6, pp 1403-1412, December 2013, DOI: 10.1109/TPDS.2013.297.
24. Tian Wenhong, Sun Xiashuang, Jiang Yaqiu, Wang Haoyan, “CRESS: A Platform of Infrastructure Resource Sharing for Educational Cloud Computing”, IEEE Communications, Volume 10, Issue 9, pp 43-52, September 2013, DOI: 10.1109/CC.2013.6623502.
25. Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, Jie Yu, Huazhong, “Gearing Resource-Poor Mobile Devices With Powerful Clouds: Architectures, Challenges, And Applications”, IEEE Wireless Communications, Volume 20, Issue 3, pp 14-22, June 2013, DOI: 10.1109/MWC.2013.6549279.
26. Foued Jrad, Jie Tao, Achim Streit, “A Broker-based Framework for Multi-Cloud Workflows”, International workshop on Multi-cloud applications and federated clouds, 2013, pp 61-68, DOI: 10.1145/2462326.2462339.
27. Weijia Song, Zhen Xiao, Qi Chen, Haipeng Luo, “Adaptive Resource Provisioning for the Cloud Using Online Bin Packing”, No. 1, pp 1, 2013, DOI: 10.1109/TC.2013.148.
28. Giacomo V. Mc Evoy, Bruno Schulze, Eduardo L.M. Garcia, “Performance and Deployment Evaluation of a Parallel Application in an on-premises Cloud Environment”, 7th International Workshop on Middleware for Grids, Clouds and e-Science, Article No. 2, DOI: 10.1145/1657120.1657122.
29. Rakpong Kaewpuang, Dusit Niyato, Ping Wang, Ekram Hossain, “A Framework for Cooperative Resource Management in Mobile Cloud Computing”, IEEE Journal on Selected Areas in Communications, Volume 31, Issue 12, pp 2685-2700, December 2013, DOI: 10.1109/JSAC.2013.131209.
30. Chao Zhang, Jianguo Yao, Zhengwei Qi, Miao Yu, Haibing Guan, “vGASA: Adaptive Scheduling Algorithm of Virtualized GPU Resource in Cloud Gaming”, IEEE Transactions on Parallel and Distributed Systems, Volume 25, Issue 11, November 2013, pp 3036-3045, DOI: 10.1109/TPDS.2013.288.
31. Gaurav Raj, “An Efficient Broker Cloud Management System”, International Conference on Advances in Computing and Artificial Intelligence, Pages 72-76, DOI: 10.1145/2007052.2007067.
32. Chrysa Papagianni, Aris Leivadeas, Symeon Papavassiliou, Vasilis Maglaris, Cristina Cervello Pastor, Alvaro Monje, “On the Optimal Allocation of Virtual Resources in Cloud Computing Networks”, IEEE Transactions on Computers, Volume 62, Issue 6, February 2013, pp 1060-1071, DOI: 10.1109/TC.2013.31.
33. Wassim Itani, Cesar Ghali, Ramzi Bassil, Ayman Kayssi, Ali Chehab, “BGP-Inspired Autonomic Service Routing for the Cloud”, 27th Annual ACM Symposium on Applied Computing 2012, Pages 406-411, DOI: 10.1145/2245276.2245356.
34. Shikharesh Majumdar, “Resource Management on Clouds and Grids: Challenges and Answers”, 14th Communications and Networking Symposium, Pages 151-152, 2011.
35. Qi Zhang, Mohamed Faten Zhani, Raouf Boutaba, Joseph L. Hellerstein, “Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud”, IEEE Transactions on Cloud Computing, Volume 2, Issue 1, pp 14-28, April 2014, DOI: 10.1109/TCC.2014.2306427.

36. Simon Caton, Christian Haas, Kyle Chard, Kris Bubendorfer, Omer F. Rana, "A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks", IEEE Transactions on Services Computing, Volume 7, Issue 3, January 2014, pp 359-372, DOI: 10.1109/TSC.2014.2303091.
37. Imad M. Abbadi, Anbang Ruan, "Towards Trustworthy Resource Scheduling in Clouds", IEEE Transactions on Information Forensics and Security, Volume 8, Issue 6, pp 973-984, February 2013, DOI: 10.1109/TIFS.2013.2248726.
38. Haiying Shen, Guoxin Liu, "An Efficient and Trustworthy Resource Sharing Platform for Collaborative Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, Volume 25, Issue 4, pp 862-875, April 2013, DOI: 10.1109/TPDS.2013.106.
39. Alejandra Rodriguez, Rajkumar Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds", IEEE Transactions on Cloud Computing, Volume 2, Issue 2, pp 222-235, April 2014, DOI: 10.1109/TCC.2014.2314655.
40. Pan Yi, Hui Ding, Byrav Ramamurthy, "Cost-Optimized Joint Resource Allocation in Grids/Clouds With Multilayer Optical Network Architecture", IEEE/OSA Journal of Optical Communications and Networking, Volume 6, Issue 10, pp 911-924, October 2014, DOI: 10.1364/JOCN.6.000911.
41. Guofang Nan, Zhifei Mao, Minqiang Li, Yan Zhang, Stein Gjessing, Honggang Wang, Mohsen Guizani, "Distributed Resource Allocation in Cloud-Based Wireless Multimedia Social Networks", IEEE Network, Volume 28, Issue 4, pp 74-80, July 2014, DOI: 10.1109/MNET.2014.6863135
42. Shriram K Vasudevan, Sivaraman R, Z.C.Alex, "Software Defined Radio Implementation (With simulation & analysis)", International Journal of Computer Applications, Volume 4, Issue 8, 21-27 August 2010