IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

QUALITY-OF-SERVICE WITH LOAD BALANCING IN CLOUD COMPUTING

[1]Author: Imtiyaz Khan, research scholar department of Computer science & Engineering at Sri Satya Sai University of Technology & Medical Sciences, Sehore-MP.

[2]Author: Dr. Pankaj Kawadkar, Professor at department of Computer science & Engineering at Sri Satya Sai University of Technology & Medical Sciences, Sehore-MP.

Abstract: Because of the added complexity of dealing with quality of service (QoS) needs in a highly virtualized dynamic environment, the cloud computing paradigm brings new problems to performance management of both applications and infrastructure. This is especially true.This is true for either SaaS apps in public clouds that are being launched into production with strict security requirements, or for SaaS applications in private clouds that are being deployed into production with strict security SLAs or important internal applications developed on private cloud PaaS/IaaS infrastructures.As a result, performance testing activities are critical for lowering the risks associated with software development. Managing significant changes in user and transaction workloads or deploying to production. Enterprise apps have been migrating to the cloud in large numbers in recent years. Managing QoS, the difficulty of assigning resources to the application in order to provide a service level in terms of performance, availability, and dependability, is one of the difficulties faced by cloud applications. QoS modelling approaches applicable for cloud systems are summarised in this paper, which aims to help researchers in this area. We also discuss and characterise their early use in different cloud QoS management decision-making concerns.

Keywords: virtualization, infrastructure, Service Level Agreement, Quality of Service.

1 Introduction

Performance testing is often a time-consuming and costly task, but it helps to reduce the risks of going live. Performance testing in cloud settings is more difficult.As a result, the costs may rise. However, because the dangers of a conventional setting remain,Performance testing must be properly conceived and organised if they are to exist. In cases when aWhen it comes to private clouds, responsibility is usually split amongst internal applications.owners, as well as the provider of internal cloud infrastructure. Explicit SLAs are rarely used in this situation defined. Nonetheless, when working with mission-critical applications and applications that require a high level of security,heavy loads, cloud-specific performance testing should be planned together.as well as the execution.

When SaaS applications are delivered to customers with SLAs on public clouds, cloud-specific performance testing must be included alongside other standard testing tasks. Because the cloud service provider owns the application and is responsible for ensuring that SLAs are satisfied, the provider must arrange the activities in this situation.A cloud environment, in general, consists of a cloud management platform and a managed platform. It's a good idea to think about unique performance testing scenarios for each of these:

- The cloud management platform is being tested.
- On-premises and cloud-based application performance testing

The on-demand capacity management paradigm has gained traction in recent years due to its technological and economic benefits [1]. A large number of cloud service companies are now offering a variety of products and services to consumers, including Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [2]. When it comes to business data centres, private and hybrid cloud architectures are becoming more common. However, despite the fact that the cloud has made capacity provisioning easier, it has also brought new QoS management issues to light. Application and platform/infrastructure quality of service relates to how well an application performs, is reliable, and is always available (QoS). Both cloud customers and cloud providers rely on QoS to ensure that promised quality features are delivered and to keep costs in check. With service level agreements (SLAs) that set quality-of-service objectives and consequences for SLA breaches, finding the right compromise is a challenging process. [3] In spite of the growing interest in quality of service (QoS) features since the advent of cloud computing, the fluctuating performance and resource isolation tactics used by cloud platforms have made QoS research, prediction, and assurance more difficult. Due to the high degree of programmability available in cloud hardware and software resources, some academics have started investigating approaches for automating QoS management [4]. Cloud computing QoS modelling tools and their initial application to cloud resource management are discussed in this study in order to aid in these efforts.

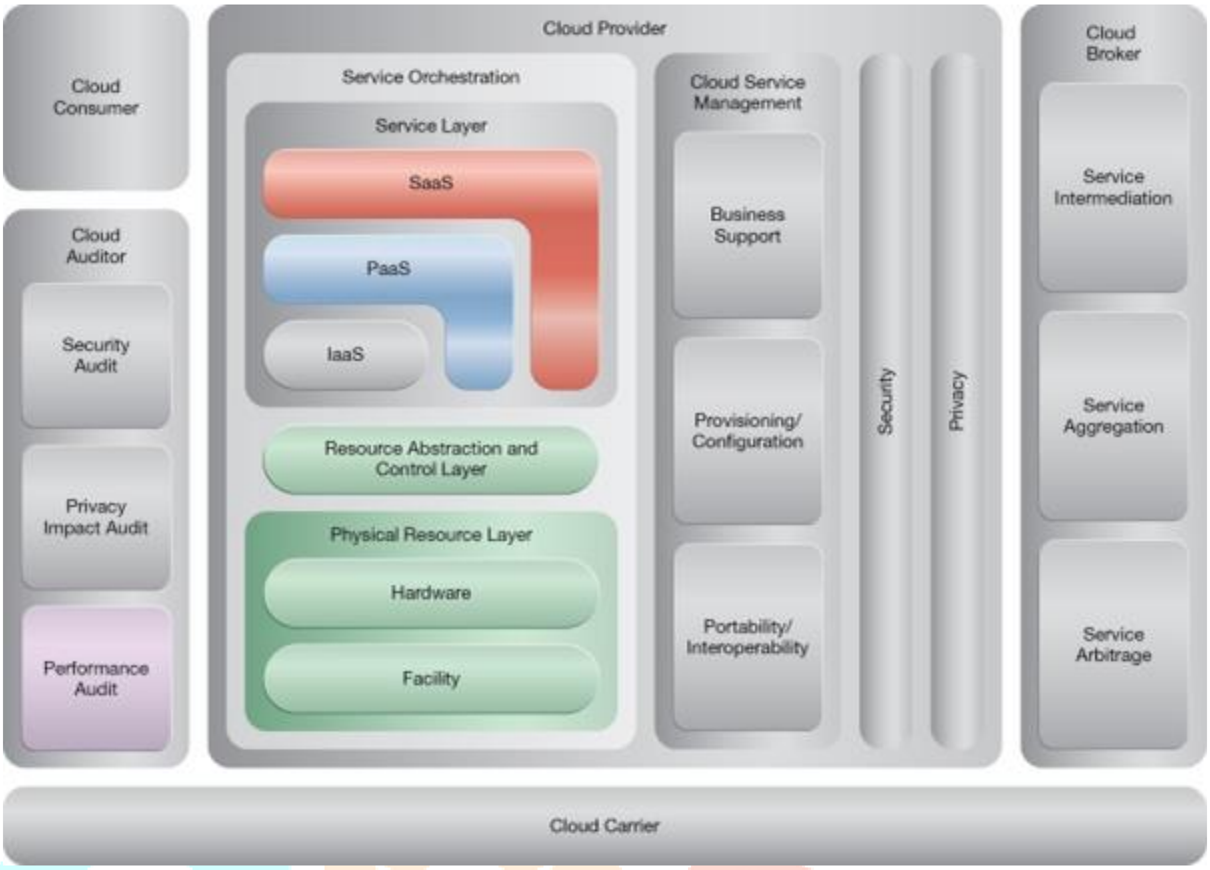

Fig 1:architecture of cloud computing platform framework

Scope: When it comes to cloud computing, various technical advancements have been made, including virtualization, web services, and enterprise application service level agreements (SLA) management. To deal with this level of technical uncertainty, cloud system characterization necessitates the use of a variety of modelling methods. However, the literature on QoS modelling is voluminous, making it challenging to get a complete grasp of the methodologies available and their present state.

Methodology: Our objective is to give an overview of early research efforts in cloud QoS modelling, categorising contributions based on their relevance for pertinent issues as well as approaches used. Our approach is as follows:instead of covering particular technical issues or offering new ideas, tries to cover as many works as possible.to introduce readers to modelling approaches We concentrate on contemporary modelling research released after 2006 in particular.concentrating on quality of service in cloud systems We also talk aboutSeveral strategies created originally for modelling andenterprise data centres with dynamic managementIn the cloud, they've been used in a number of ways. In addition, the review takes into account QoS modelling methodologies forMulti-tier apps, for example, are interactive cloud services. Those who specialise in batch applications, such as those built on the MapReduce framework.

Predicting or foreseeing the pace at which requests will arrive and the resources (e.g., CPU use) that applications will need on an infrastructure or platform is known as workload modelling, a strategy. In reaction to such workloads, the Quality of Service (QoS) displayed We assess cloud measurement research to aid in the definition of such characteristics for a certain cloud. Discussions of workload characteristics and inference methodologies for a Quality of Service (QoS) research will next be given.

- The purpose of system modelling is to evaluate a system's performance.

The value of some assets is forecasted using models. QoS measurements include response time, dependability, and availability. We look at formalisms and tools.used in these evaluations, as well as their present applications. QoS models are frequently used in system management to solve decision-making problems. Simple heuristics to nonlinear programming and meta-heuristics are all used to make better conclusions. In Section 4, we look at work on capacity allocation and load balancing decision-making balance, as well as admissions control, which includes researchworks that give managerial solutions fora cloud infrastructure (provided by the cloud service provider)Techniques for resource management (from a standpoint) andthe infrastructure user (for example, a service provider)seeking to reduce operational costs as much as possible, whileensuring a high degree of QoS for end users).Modeling cloud workloads

To have high prediction capability for QoS models, correct workload models must be defined. We look at workload characterisation studies and related modelling methodologies in this article.

2 CHARACTERIZATION OF THE WORKLOAD

2.1 Environment in which the application will be deployed.

Several studies have used benchmarking to try to characterise the QoS provided by cloud deployment settings. In QoS modelling, statistical characterizations of empirical data may be used to assess hazards without the requirement for an ad-hoc measurement investigation. QoS model components, such as network bandwidth variance, VM starting delays, and start failure rates, can only be accurately modelled if they are based on real-world data. Performance differences for several kinds of virtual machine instances have been documented [5-7]. The main sources of this unpredictability, which may also be observed within the operating system, are hardware heterogeneity and VM interaction. Other studies describe the heterogeneity in VM starting times [7,8], which is linked to the size of the operating system image [8]. High-performance contention in CPU-bound applications [9] and network performance overheads [10] have been discovered in various Amazon EC2 research. A few public and private sector characterization studiesPrivate PaaS hosting alternatives, as well as cloud database comparisons, were also mentioned in the literature [11,12][13-16] and storage services A comparison of different providers is also offered based on a broad range of parameters.Techniques of Regression A popular workload inference method includes calculating just the mean demand exerted on the resource by a certain type of request [26-28].[26] introduces a common model calibration approach. Performance model predictions are compared with experimental data in order to determine whether the model is accurate (e.g., reaction time, throughput, and resource consumption).

2.2 Queueing Systems are a type of queueing system.

In system modelling, queuing theory is frequently used to describe hardware or software.a scarcity of resources There are several analytical formulae available, for example. One way to describe the characteristics of single queueing systems is to use mean waiting times or waiting buffer occupancy probability. Analytic queueing is a feature of cloud computing.In optimization programmes, formulae are often used.where they're put through their paces in a variety of what-if scenarios. Queues are common in analytical calculations. Servers with exponential service arrival times (M/K/1), queues, and one server or k servers. All three shifts (M/G/1) have the same amount of service time. On a first-come, first-served (FCFS) basis, or on a processor-sharing basis, scheduling is generally thought of as (PS). The M/G/1 PS A queue, for example, is a popular abstraction for a CPU. A number of cloud investigations [47,48] have used it.

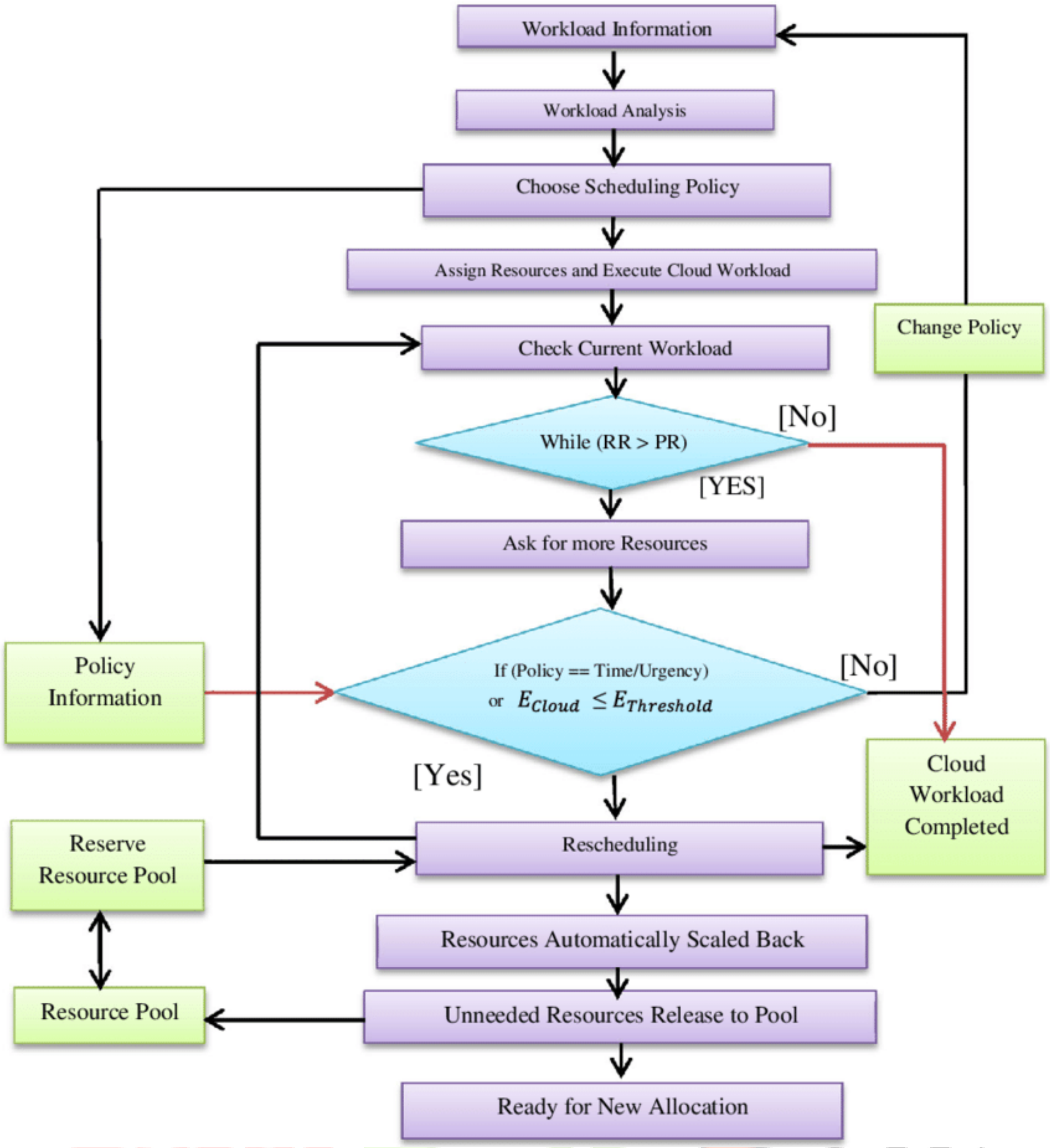

Fig 2: Flowchart of cloud workload management framework

Cloud-based app workloads When attempting to describe the cloud deployment environment in earlier works, users often run into the added challenge of attempting to describe the workloads that a cloud application executes.

Methods such as blackbox forecasting and trend analysis are often used to estimate the volume of online traffic at different points in time. For over two decades, web servers have relied on time series forecasting. [18] Cloud application modelling is already using autoregressive models for auto-scaling. Wavelet-based procedures, regression analysis, filtering, Fourier analysis, and kernel-based methods are some of the more prevalent techniques. Conclusions regarding workload Before most QoS models for business applications can be parameterized, they need to be able to assess resource needs. Deep monitoring overheads and the difficulty of tracking individual request execution routes are typically cited as justifications for inference [25]. Over the past two decades, many research have looked at the topic of measuring the resource demand exerted on physical resources by an application, such as CPU needs, using indirect measures. Because of the scarcity of available data, cloud service providers and customers may utilise inference approaches to estimate the workload profile of individual virtual machines (VMs) operating on their systems.

2.3 Workloads for cloud applications.

When attempting to describe the cloud deployment environment in earlier works, users often run into the added challenge of attempting to describe the workloads that a cloud application executes. Web traffic volumes at certain times and sizes are often projected using black box forecasting and trend analysis approaches. There has been a lot of research done on time series forecasting.For over two decades, it has been utilised for web servers. In example, autoregressive models are frequently utilised in software and are already employed in cloud applications. For example, modelling for auto-scaling [18]. Other typicalWavelet-based approaches and regression are examples of methodologies.filtering, Fourier analysis, and kernel-based analysismethodsrecent papers in workload modelling that are important to cloud computing. Hidden Markov Models are used by Khan et al. [20] to capture and forecast temporal correlations between workloads on multiple cloud computing clusters. The authors of this study offer a method to define and anticipate workloads in cloud systems so that cloud resources may be provisioned efficiently. To discover servers with comparable workload patterns, the authors devise a co-clustering technique. The trend is discovered by comparing the performance of programmes running on various servers. By detecting the temporal links between various clusters and making use of this knowledge, they use hidden Markov models to predict the future.

2.4 Inferences about workload

Most QoS models for corporate applications require the ability to measure resource demands before they can be parameterized. It is common to find grounds for inference in the high costs of monitoring and the difficulties in tracking individual request execution pathways. [25]. Several studies have looked at the difficulty of detecting an application's resource demand on physical resources, such as CPU demands, using indirect measurements throughout the last two decades. Because of the scarcity of available data, cloud service providers and customers may utilise inference approaches to estimate the workload profile of individual virtual machines (VMs) operating on their systems.

3 MODELS OF SYSTEMS

Unbiased when it comes to the logic that drives a cloud system ExplicitThis logic, or a portion of it, can be modelled for QoS prediction.Assist in increasing the efficiency of QoS management.To represent QoS in a network, there are several types of models that may be employed.cloud computing systems We'll go over queueing models briefly here.Petri nets and other specific formalisms are used to assess dependability. Stochastic activity networks, stochastic process algebras, models evaluated using stochastic reward networks [44], and models assessed utilising Checking probabilistic models are some other categories that may be used. [45]. An evaluation of the two Here you may learn about the benefits and drawbacks of certain typical stochastic formalisms. It is possible to find [46].

3.1 Models of performance

Queueing systems, queueing networks, and layered queueing networks are examples of performance models (LQN). In contrast to queueing systems, queuing networks may represent the interactions of several resources and/or application components.

Connection pools, admission controls, and synchronous request calls all use LQNs to better specify essential interactions between application processes. Modeling these qualities typically needs a deep knowledge of the behaviour of the application. While certain queueing systems and networks have closed-form solutions, numerical methodologies are employed to solve other models, including LQNs.

3.2 Models of dependability

Most often used formalisms for dependable research include Petri nets, Reliability Block Diagrams (RBD), and Fault Trees. It is possible to utilise Petri nets to describe generic interactions between system components, such as the synchronisation of event firing times. They're often used in employee evaluations, too.

The goal of RBDs and Fault Trees is to derive overall system reliability from system component reliability. One or more of the components may fail, and this might lead to the failure of other components as well. A sort of petri dish is a petri dish. Since its inception in the 1970s, Petri nets have been well recognised for their capacity to improve computer speed and dependability. Stochastic transitions have been added to Petri nets.

3.3 Models of black-box service

Aside from web service composition optimization, service models are increasingly being employed in a broader range of contexts [76].IaaS is also significant in the definition of SaaS apps.cloud-based business processes and resource orchestrationexecution. The concept underlying the approaches discussed in this article. We'll characterise a service in this part by looking at how it responds. as a result of a high number of inquires) is assumed to have no more information on its internal characteristics. Blackbox service models based on deterministic or average execution times [77-81] are non-parametric. A number of other books, on the other hand, include standard deviations or finite ranges in their descriptions.

3.4 Models for simulation

For cloud system simulation, a variety of simulation tools is available. This toolkit, CLOUDSIM [93], has been used in many projects, which enables users to build a simulation model that incorporates virtualized cloud. Resources, which may be spread over many data centres,hybrid deployments, for example. CLOUDANALYST is a CLOUDSIM addon that allows you to model geographically dispersed workloads.apps running on several virtualized data centresCLOUDSIM [95] is enhanced by EMUSIM [95].an emulation phase that makes use of the Automated Emulation Framework[96] Framework (AEF). Emulation is a technique for learning.gathering profiling information from the application's behaviourCLOUDSIM then uses this information as input.It calculates the quality of service for a specific cloud deployment.[84,85] There is a lot of variation in the execution times. ParametricInstead, service models assume exponential or Markovian distributions.To capture heavytailed execution times, use distributions [86,87] and Pareto distributions.

3.5 Allocation of infrastructure-user capacity

As the user determines the number of virtual machines or application containers running in the system, capacity allocation takes place from their perspective. In IaaS and PaaS environments, this occurs. Users in this circumstance tend to be software vendors looking to boost their income by offering higher-quality services. The next step is to figure out how many VMs or containers are needed to achieve the specified QoS while keeping costs and performance in balance. Auto-scaling rules are often used by users to allocate capacity. Mao and Humphrey [111] create an auto-scaling system to ensure that all activities are completed on time. Workload is taken into consideration in the solution.

3.6 Load balancing between infrastructure providers

The use of request load-balancing is becoming increasingly popular in cloud services. According to a load dispatching policy, a load balancer sends user requests to servers. Several policies control the decision-making process. Besides the quantity of data they collect, Research employs a variety of methods to analyse it. Policies that are easy to implement or understand have been the focus. therefore reducing costs or providing some degree of confidence of optimality, as shown by analytic models Both centralised and decentralised load balancing strategies have been studied in the literature.Providers [122] provides a decentralised method among centralised alternatives. SLAs are explicitly taken into account in an offline optimization problem for regional load balancing among data centres. in addition to fluctuating power prices. This is accompanied bya web-based algorithm to deal with the volatility of power prices The proposed algorithm is put to the test.

3.7 Load balancing between infrastructure and users

In the case study described above, the load balancer is set up and managed in an open and transparent manner. the cloud service provider. In certain cases, the user has the ability to choose. A cloud application must have its own load balancer deployed. This might be useful for tackling capacity issues together, for example.Load balance and allocation[47], for example, analyses a multi-IaaS service centre joint optimization problem. A non-linear model is one that does not follow a straight line.for the distribution of capacity and load redirection of numerous serversDecomposition is presented and used to solve request classes. An oracle having comprehensive understanding of the circumstance, as opposed to a collection of literature-based heuristics. The suggested strategy is effective, as shown by future load. without penalising SLAs and relying on heuristics It has the capacity to create new things.

3.8 Control of entrance by infrastructure providers

Admission control is a load balancing system that denies requests during periods of high workload.Defend the quality of service (QoS). There has been a significant amount of effort put in.Throughout the previous ten years for optimal online admission controlmulti-tier apps and servers The underlying concept is to anticipate the value of a certain QoS measure and whether or not such a value is possible.. When the number of new sessions hits a specific level, the admission controller suspends all new sessions in favour of meeting the needs of those who have already registered. Selected from sessions that have previously been approved In the field of cloud computing, many publications on admission control have been published. Establish a resource provisioning, virtual machine provisioning, and cloud computing analytical model in IaaS. pool administration and deployment Task rejection probability, service delay, and steady-state server pool distribution are all predicted by this model.

3.9 Control of access to the infrastructure by users

The admission control strategy is used as an extreme overload mechanism when new resources are obtained after a lengthy wait. If, for example, during a cloud burst, public cloud services are not immediately accessible (i.e., when part of the application traffic is redirected from a private to a public data centre to cope with a traffic intensity that exceeds the capacity of the private infrastructure). To keep the application's quality of service (QoS) as high as possible for present users, one may choose to reject new requests (or at least a portion of them, such as gold customers).

CONCLUSION

As a standard operating paradigm for corporate applications, cloud computing has grown from a cutting-edge solution. There are many different technologies utilised in cloud systems, making it difficult for a service provider to analyse their quality of service and establish service-level guarantees. We looked at existing workload and system modelling methodologies, as well as early cloud QoS management apps.

Although prominent in the software performance engineering field, the number of papers that use white-box system modelling methodologies to QoS management is fairly restricted. This basically creates a split between the knowledge that an application's designers may make accessible for it and the strategies utilised to manage it. QoS management may be improved by having a better grasp of application internals, according to a research question. . Accessible data, QoS model complexity, decision-making computational cost, and prediction accuracy are all trade-offs. The scientific community should take a closer look at this trade-off. In QoS management research, gray-box models with an emphasis on resource consumption modelling are becoming more prominent. Performance, on the other hand, is typically stated in a basic fashion and is tied to the average resource requirements of the applications. It has arisen as a key concern in today's cloud offerings because of the cloud measurement studies discussed in Section 2.1, requiring the building of more complete models that can not only capture average CPU needs but also variability. White-box and gray-box models are less widely recognised in comparison to black-box models (for example, Quality of Service (QoS) in online services).

REFERENCES

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2010) A view of cloud computing. Commun ACM 53(4):50–58
2. 2. Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Internet ServAppl 1(1):7–18
3. Ardagna D, Panicucci B, Trubian M, Zhang L (2012) Energy-aware autonomic resource allocation in multitier virtualized environments. IEEE Trans ServComput 5(1):2–19
4. 4. Petcu D, 0 Macariu G, Panica S, Craciun C (2013) Portable cloud applications - from theory to practice. Future Generation ComputSyst 29(6):1417–1430
5. Farley B, Juels A, Varadarajan V, Ristenpart T, Bowers KD, Swift MM (2012) More for your money: Exploiting performance heterogeneity in public clouds. In: Proceedings of the 2012 Third ACM Symposium on Cloud Computing, SoCC ’12, San Jose, CA, USA, pp 1–14
6. Ou Z, Zhuang H, Nurminen JK, Ylä-Jääski A, Hui P (2012) Exploiting hardware heterogeneity within the same instance type of amazon ec2. In: Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Ccomputing, HotCloud’12, Boston, MA, USA, pp 4–4
7. Schad J, Dittrich J, Quiané-Ruiz J-A (2010) Runtime measurements in the cloud: Observing, analyzing, and reducing variance. Proc VLDB Endowment 3(1–2):460–471
8. Mao M, Humphrey M (2012) A performance study on the VM startup time in the cloud. In: Proceedinngs of the 2012 IEEE Fifth International Conference on Cloud Computing, CLOUD ’12, Honolulu, HI, USA, pp 423–430
9. Xu Y, Musgrave Z, Noble B, Bailey M (2013) Bobtail: Avoiding long tails in the cloud. In: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, NSDI ’13, Lombard, IL, USA, pp 329–342
10. Wang G, Ng TSE (2010) The impact of virtualization on network performance of amazon ec2 data center. In: Proceedings of the 29th Conference on Information Communications, INFOCOM’10, San Diego, CA, USA, pp 1163–1171

11. Hill Z, Li J, Mao M, Ruiz-Alvarez A, Humphrey M (2011) Early observations on the performance of Windows Azure. Sci Program 19(2–3):121–132
12. Li Z, O'Brien L, Ranjan R, Zhang M (2013) Early observations on performance of Google compute engine for scientific computing. In: Proceedings of the 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, volume 1 of CloudCom 2013, Bristol, United Kingdom, pp 1–8
13. Drago I, Mellia M, Munafo MM, Sperotto A, Sadre R, Pras A (2012) Inside Dropbox: understanding personal cloud storage services. In: Proceedings of the 2012 ACM Conference on Internet Measurement Conference, IMC
14. Kossmann D, Kraska T, Loesing S (2010) An evaluation of alternative architectures for transaction processing in the cloud. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, Indianapolis, IN, USA, pp 579–590

15. Wada H, Fekete A, Zhao L, Lee K, Liu A (2011) Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective. In: Proceedings of the 5th Biennial Conference on Innovative Data Systems Research, CIDR 2011, Asilomar, CA, USA, pp 134–143

16. Liu S, Huang X, Fu H, Yang G (2013) Understanding data characteristics and access patterns in a cloud storage system. In: Proceedings of the 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2013, Delft, Nederlands, pp 327–334

17. Li A, Yang X, Kandula S, Zhang M (2010) Cloudcmp: comparing public cloud providers. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, pp 1–14

18. Roy N, Dubey A, Gokhale A (2011) Efficientautoscaling in the cloud using predictive models for workload forecasting. In: Proceedings of the 2011 IEEE International Conference on Cloud Computing, CLOUD '11, Washington, DC, USA, pp 500–507

19. Gasquet C, Witomski P (1999) Fourier analysis and applications: filtering, numerical computation, wavelets, volume 30 of Texts in applied mathematics. Springer, New York, USA

20. Khan A, Yan X, Shu T, Anerousis N (2012) Workload characterization and prediction in the cloud: A multiple time series approach. In: Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012, Maui, HI, USA, pp 1287–1294

21. Di S, Kondo D, Walfredo C (2012) Host load prediction in a google compute cloud with a Bayesian model. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC12, Salt Lake City, Utah,USA, pp 1–11

22. Gmach D, Rolia J, Cherkasova L, Kemper A (2007) Workload analysis and demand prediction of enterprise data center applications. In: Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization, IISWC '07, Boston, MA, USA, pp 171–180

23. Zhu Q, Tung T (2012) A performance interference model for managing consolidated workloads in QoS-aware clouds. In: Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, CLOUD '12, Honolulu, HI, USA, pp 170–179

24. Hoffmann GA, Trivedi KS, Malek M (2007) A best practice guide to resource forecasting for computing systems. IEEE Trans Reliability 56(4):615–628

25. Anandkumar A, Bisdikian C, Agrawal D (2008) Tracking in a spaghetti bowl: Monitoring transactions using footprints. In: Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. ACM Press, Annapolis, Maryland, USA, pp 133–144

26. Menascé D, Almeida V, Dowdy L (1994) Capacity planning and performance modeling: from mainframes to client-server systems. Prentice-Hall, Inc. NJ, USA

27. Rolia J, Vetland V (1995) Parameter estimation for performance models of distributed application systems. In: In Proc. of CASCON. IBM Press, Toronto, Ontario, Canada, p 54

28. Rolia J, Vetland V (1998) Correlating resource demand information with ARM data for application services. In: Proceedings of the 1st international workshop on Software and performance. ACM, Santa Fe, New Mexico, USA, pp 219–230

29. Liu Y, Gorton I, Fekete A (2005) Design-level performance prediction of component-based applications. IEEE Trans SoftwEng 31(11):928–941

30. Sutton CA, Jordan MI (2008) Probabilistic inference in queueing networks. In: Proceedings of the 3rd conference on Tackling computer systems problems with machine learning techniques. USENIX Association, Berkeley, CA, US, p 6

31. Sutton CA, Jordan MI (2010) Inference and learning in networks of queues. In: International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, pp 796–803

32. Zhang Q, Cherkasova L, Smirni E (2007) A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In: Proc. of the 4th ICAC Conference, Jacksonville, Florida, USA, pp 27–27

33. Liu Z, Wynter L, Xia C, Zhang F (2006) Parameter inference of queueing models for it systems using end-to-end measurements. Perform Eval 63(1):36–60

34. Casale G, Cremonesi P, Turrin R (2008) Robust workload estimation in queueing network performance models. In Proc. of Euromicro PDP:183–187

35. Kalbasi A, Krishnamurthy D, Rolia J, Dawson S (2012) DEC: Service demand estimation with confidence. IEEE Trans SoftwEng 38(3):561–578

36. Kalbasi A, Krishnamurthy D, Rolia J, Richter M (2011) MODE: Mix driven on-line resource demand estimation. In: Proceedings of the 7th International Conference on Network and Services Management. International Federation for Information Processing, pp 1–9

37. Pacifici G, Segmuller W, Spreitzer M, Tantawi A (2008) CPU demand for web serving: Measurement analysis and dynamic estimation. Perform Eval 65(6):531–553

38. Cremonesi P, Sansottera A (2012) Indirect estimation of service demands in the presence of structural changes. In: Proceedings of Quantitative Evaluation of Systems (QEST). IEEE, London, UK, pp 249–259

39. Cremonesi P, Dhyani K, Sansottera A (2010) Service time estimation with a refinement enhanced hybrid clustering algorithm. In: Analytical and Stochastic Modeling Techniques and Applications. Springer, pp 291–305

40. Sharma AB, Bhagwan R, Choudhury M, Golubchik L, Govindan R, Voelker GM (2008) Automatic request categorization in internet services. ACM SIGMETRICS Perform Eval Rev 36(2):16–25

41. Wu X, Woodside M (2008) A calibration framework for capturing and calibrating software performance models. In: Computer Performance Engineering. Springer, pp 32–47

42. Zheng T, Woodside CM, Litoiu M (2008) Performance model estimation and tracking using optimal filters. IEEE Trans SoftwEng 34(3):391–406

43. Desnoyers P, Wood T, Shenoy PJ, Singh R, Patil S, Vin HM (2012) Modellus: Automated modeling of complex internet data center applications. TWEB 6(2):8

44. Longo F, Ghosh R, Naik VK, Trivedi KS (2011) A scalable availability model for Infrastructure-as-a-Service cloud. In: Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems Networks, DSN 2011, Hong Kong, China, pp 335–346

45. Calinescu R, Ghezzi C, Kwiatkowska MZ, Mirandola R (2012) Self-adaptive software needs quantitative verification at runtime. Commun ACM 55(9):69–77

46. Chung M-Y, Ciardo G, Donatelli S, He N, Plateau B, Stewart W, Sulaiman E, Yu J (2004) Comparison of structural formalisms for modeling large markov models. In: Parallel and Distributed Processing Symposium, 2004 Proceedings. 18th International, Santa Fe, New Mexico, USA, p 196

47. Ardagna D, Casolari S, Colajanni M, Panicucci B (2012) Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. J Parallel Distributed Comput 72(6):796–808

48. Xiong P, Wang Z, Malkowski S, Wang Q, Jayasinghe D, Pu C (2011) Economical and robust provisioning of n-tier cloud workloads: A multi-level control approach. In: Proceedings of the 31st IEEE International Conference on Distributed Computing Systems (ICDCS), Minneapolis, Minnesota, USA, pp 571–580

49. Almeida J, Almeida V, Ardagna D, Cunha I, Francalanci C, Trubian M (2010) Joint admission control and resource allocation in virtualized servers. J Parallel Distributed Comput 70(4):344–362

50. Goudarzi H, Pedram M (2013) Geographical load balancing for online service applications in distributed datacenters. In: Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing, CLOUD '13, Santa Clara, CA, USA, pp 351–358

51. Zhang Q, Zhu Q, Zhani MF, Boutaba R (2012) Dynamic service placement in geographically distributed clouds. In: Proceedings of the 2012 IEEE 32Nd International Conference on Distributed Computing Systems, ICDCS '12, Macau, China, pp 526–535

52. Ellens W, Zivkovic M, Akkerboom J, Litjens R, van den Berg H (2012) Performance of cloud computing centers with multiple priority classes. In: Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing, CLOUD '12, Honolulu, HI, USA, pp 245–252

53. Kusic D, Kandasamy N (2006) Risk-aware limited lookahead control for dynamic resource provisioning in enterprise computing systems. In: Proceedings of the 2006 IEEE International Conference on Autonomic Computing, ICAC '06, Dublin, Ireland, pp 74–83

54. Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2009) Power and performance management of virtualized computing environments via lookahead control. Cluster Compute 12(1):1–15