Approaches for Software Performance Modelling, Cloud Computing and Openstack

Deepali Gill
Dept. of Computer Science & Engineering
Amity University
Noida, India

Hari Mohan Pandey
Dept. of Computer Science & Engineering
Amity University
Noida, India

ABSTRACT

Performance modelling is simulation of the user inputs and then altering the systems' configuration in order to monitor performance of software. Performance modelling rarely incorporated but gradually we have realized its significance. There are various approaches for modelling and each has its own benefits and hindrances. Performance modelling when applied on the cloud is again a very challenging task. For a good model for specific software we should have a good knowledge about the software working. This paper includes all the important performance modelling approaches. Along with this, we also have an overview of the cloud and Openstack environment.

General Terms

Cloud computing

Keywords

Cloud Computing, OpenStack, Performance Modelling.

1. INTRODUCTION

Performance modelling is indeed of prime importance. It is significant as errors that usually occur after the deployment of the product are not because of any functional error or we can say fault but because of the complaints in the non-functional parameters like not responding, crashing because of the heavy load [1]. Though it's one of the parameters that is quite neglected but problems related to this could be severe that could actually lead to major changes in the entire architecture [2]. Performance testing is a sub part of the System Performance Engineering which checks the application for the various features. Modelling is performed to track various factors that would contribute in the growth and better performance with sole goal to ardently predict if there could be any breach of Service Level Agreement [3].

Performance Modelling is key role in building trust and confidence with the customers in IT company [4].A large amount of performance test in order to design the model after every build of an application. It is basically checking all the performance parameters with respect to the software or hardware configuration. IT is performed after the load testing and functional testing [5]. When we talk about the load and functional testing, the results in the form of the Boolean value either they pass or fail. On contrary, in the performance tests the results is in the form of the quantitative value like response times is this much and the maximum number of the virtual machines that are running, the hardware utilization is this much. Now, if we see in detail, the performance testing includes loads of tests to be conducted and these tests are conducted against and again to check their accurate validity. And as they are performed manually, hence making it more complicated. Tests are executes iteratively many a times to check the exact values. It is not doubt time consuming like we have to repeat entire sets of tests only because of one bug to be evaluated.

Automates modelling has also gain good popularity like WebLoad and LoadRunners, these tools are very useful but then at the end of the final release is we compare, and infer that the conventional no automated mechanism is more trusted to test efficiently.

The paper comprises of 9 sections. Initially starting with the overview of the performance modeling and moving forward with the different approaches for the same. In the later sections, we have described the concept of cloud computing and OpenStack.

2. PERFORMANCE MODELING APPROACHES

The modelling could be achieved by any of the three mechanisms namely - analytically, measurement based or by simulating. The measurement based is very tedious and it is applicable once the product or application is fully developed and deployed. Thus, now the latter two techniques are more popular, specifically the simulation based. Where analytical techniques use the formulas and theoretical models comprising of various techniques like Markov chains, queuing networks etc.simulation is the emulation process of the product itself. But the hindrance in the case of the analytical and simulation technique is that, it requires a very good knowledge and understanding of the software that has to be tested. So, rather than going into evaluating mathematical equations or software for simulations, professionals opt for the measurement based models. Taking into account the simplicity, it is used more as compared to the other two [1].

Simulation is the latest approach that enable the analysis of the software early in the development cycle and with liberty to perform it on a much larger scale, at average costs and plus within the strict time limits [6].

3. MEASUREMENT BASED MODELLING

Measurement is the most apt choice for the performance modelling among the most of the approaches used to do the same. But the mechanism is tedious, as it requires application or execution of large number of performance tests. And it's not only time consuming but as it is done manually the probability of the error is also more. Hence, we need a systematic approach to use it.

Few of the challenges that we encounter are:-

a) Large number of tests that have to be performed.
b) Time limitation in performing the entire set of test cases.
c) Manual processing increase chances of making an error.
d) Repetition of the whole process if any bug is found.
Now most of the problems could solve if automation is added into it.

The Modelling process includes listing of various tests that have to be performed. It should encompass all kinds of areas in order to develop an accurate model. Then we comes the filtering out the tests which helps in taking out the weeds and helps us to focus on the main tests and thus reducing the redundancy. Setting the environment for performance testing is quite tedious. It mainly includes installation of the various tools and executing them in different operating systems. After that we execute the tests, in this it mainly comprises of 3 activities that are setting up, running the tests and shutdown test. Next thing is the readying the process to switch to next tests when one ends, and then after all the tests are performed the stage comes of making the apt comparison. And finally the modelling is performed.

Mechanism followed:-

1. Test itemization
The foremost step towards the modeling is the listing out the entire possible test that should be under the performance testing and would also satisfy the expectations of the customers too.

Hence the entire process of enumeration could be broken down into various stages:

a) Listing out the functional operations.

b) Specifying the relation between the functional operations and the corresponding issues. As we already discussed the issue that multiple scenarios could include various repetitive steps, so we could simply group the operations together and that could be considered as a single operation.

c) Next most important thing to keep the work systematic is assigning priority to each of the scenario that is identified. For instance, the parts of the system that are upgraded or changed recently should be given more preference as compared to the old ones. This is one way of actually assigning there could be various other parameters on which the priorities can b allocated.

d) As we know grouping and classification are 2 different things, so we have grouped the tests but we have to categories on the other basis, for better understanding we can take examples: - Ranging the tests in the form of

(i) Number of operations in an hour (Frequency check).

(ii)Numbers of the process functioning in parallel (concurrency) are quite many.

2. Test Minimization
Filtering out the tests from the complete test space is a tedious task as we have to read through the tests that we have listed. This task can be succeeded in two ways either manually or by tools. Both the methods have their own positive points. For specification the manual work we state this method as the Static method and the latter one as Dynamic Method.

If we talk about the static part we just have to rational about the existence of each and every test case. For example if we realize that a test scenario is quite similar to any other we could just eliminate one.

3. Setting up the Environment
The next step is for the installation of the requisite tools for the load and performance testing. We could have automated scripts for this whole process and thus simplifying the task.

4. Test Execution
This phase of the method is the most tedious among all the processes so far, so it's advisable to divide it into various sub phases and these are as follows:-

a) Test Organization
Now when it comes to testing a fully fledged system it is mandatory that each and every part of the system is tested fully. And not only have the functioning parted but also the configurations of the computing server, the database and the load generator etc. It is proposed so as variant configurations may lead to a drastic change in the performance output. The only hindrance in this part is that we can evaluate the exact results until the analysis is performed thus validation plays an eminent part.

b) Test Running
Automation has ease up this part of the process. There have been evolution of wide variety of automated tools that save all the scripts and they develop test cases by running various instances of the scripts for a real life user's experience. We have tools like LoadRunner, WebRunner and many more.
Same as this automation of the test execution we have the systematic automation of the overall framework. It basically comprises of 3 stages:-

A) Get Ready ('Warm Up') - The total load gets gradually increased.
B) Consistent Phase ('Steady State')- The former process gives way to the this stage in which we have a well defined state of the software and during this stage all the metrics for the measurement of the performance are on average value.
C) Shut-down ('Cool Down') - Everything heads toward finishing point.

b) Test Termination
This phase includes the shutdown of the process of the software that is being tested. This decision is optional whether to continue with running process or end it.

7. Test Transition
The above process states the well defined procedure how to go from one test to another of performance. The conventional way to perform this task it to start or load the next test as soon as the former test gets completed but if we practically think there could be a load of the previous test could be still be little bit left, so for overcoming this process we could add a delay in between the 2 tests to solve this problem.

The accumulation of data could be harmful to the performance to the system. Thus achieving the data is advisable and it conserving lot of wasted space to save and handle all the present data.

8. Test Analysis
The loads of tests that are performed have to be validated as to see the absence of errors. If this process is taken as to be

performed manually it probably is turned out as very lengthy n hectic. But now as automation has become an inevitable necessity so this process is not only non manual but also performs rest of the process like reducing of the tests and modelling. The whole process could also be divided into sub phases.

a) Validation of the tests
Several problems can come up like 'performance counters' can be violated due to many software or hardware situations, for instance of we want to talk about the software anomaly; the automated upgrades and downloads could interfere with the normal functioning and load of the software and could lead to sudden alterations and inconsistent values.

b) Metric Analysis
In this the real comparison is performed as we concentrate on the metric values and then all these are saved on the form of the database for the final step of the modelling.

4. ANALYTICAL APPROACH

As it is observed performance modelling is not so very popular in the present environment of the IT Company. But we cannot deny the fact that it is quite useful when we want to address issues like capacity planning, analysis for future or prediction, extrapolation and also for much problematic analysis.
Modelling does need knowledge about mathematics. Though various tools are also available but still the customers want to rely on personalized performance predictions in order to measure capacity, scalability and other parameters. Before we go in details about the process to build models that is being used in IT systems, we should have knowledge about quantities that normally play a major part in building the models. Quantities like response time, throughput, storage space and others are few of the random quantities that we should know to characterize. Some of the very simple and basic laws like 'Little's law' and 'Bottle neck law' are very popular [7].

a) Markov Chains
Apart from the above discussed laws, we have been another mechanism for doing the reliability and performance analysis. It's very apt when we have to express it in the terms of state space. It helps in good understandability of the system apart from the rough calculations that are performed. Though we have to make lot of assumptions and matrices is involved because of which many IT analyst step back, but still now we have various tools to reduce the amount of elements in the state space.

b) Petri-Nets
When the state space grows, the Markov chains are inefficient to devise, thus, then we can go with another option that is Petri-Nets. For simplification, we can say that of Markov chains are 'State-Transition Diagrams' then the Petri-nets are like flowcharts. This mechanism has become gradually very popular.

c) Queuing Networks
The best option for performance modelling; now days are the queuing networks. The whole system is represented in the form of networks of queues that could be later analyzed and results could be inferred for each of the work .

5. SIMULATION APPROACH

Now that we have discussed both analytical and measurement based approaches, till now we have inferred that these mechanisms are very efficient in getting the results if the system is in steady state. The analytical presentation is useful as any kind of problem could be easily solved by the mathematical formulation that has been deduced but if we talk about the simulation approach it concerns with algorithm cogitation for showing the performance of the software.

Even if we have changed our approach but still we will be having the random variables the stochastic process has the main concern about the time and the response to the system from one time instance to another. The whole process would revolve around the "sample paths". The sample paths can be defined as the values variables in a given time would move in a given state space this path is sample paths. Every execution of the simulation would develop a unique path [10].

Reasons to switch to the Simulation from analytical approach

(i) Level of abstraction:-
Analytical has bend towards idealism in the process; the options for making assumptions are always present and are taken. But it gives rise to unrealistic behavior as in practicality we know that every system or software is bound of various kinds of exceptions and we have to keep in mind all of them in order to produce accurate results. Even though for consistency if we assume that the interval times are exponentially decreasing or increasing , we know that could be not the case in reality as with change in situation the behavior of the software could also change. In case of simulation, there are umpteen cases of random levels that could occur.

(ii) Transient Analysis:-
One of the key factors that we see when analyzing through analytical approach is the steady behavior. Some of the software is very rarely have a steady state. As discussed we have a warm up state and then the software goes into a steady state but analytical approach fails to show the details of the state of the software when it's moving from warm up state to the steady state. Analytics ignore this scenario as it's beyond their limits. Though few of the mathematical models like Markov chains have the ability to work on the transient state too but, then the effort in evaluating such expressions is very complex and full of effort task.

(iii) Size of State Space:-
In case of the analytical modelling includes the construction and storage of the entire state space. For instance, in a Markov process the evaluation may have A X a matrix size with elements and as the process continues and with time it may be that A becomes too large such that it's unfeasible to manage it. Contrary to this, simulation generates the state-space "on the fly" that is it is just winged up [10].

Our main focus is on the discrete events and this comprises of various number of items and objects that interact to make the behavior of the system or software. Simulations are meant to be very complex programs that could be developed in language and each having their own leverage to choose one them. Some languages also come with built in features for simulation and that could be used at routine intervals for modelling.

Few of features that are possessed by all the simulations tools are listed below.

(i) Event Scheduler:-
As the whole is monitoring the behavior of the events, where events are the change in the state of the software or system. Thus, to keep track of these events we have tracker called "Event Scheduler". It has the responsibility to control and manage the events. To assemble them usually a linked list is taken. It enables the manipulation of the events in different ways like scheduling an event E a time T or keeping an event E in wait or hold state for particular time interval.

(ii) Time simulation:-
The simulated time is one of the most important time in the simulating a model. Though the event scheduler has the duty to schedule the time and this approach is known as the "event-driven time management".

(ii) Event and Random number generators:-
As both of these should be generated in time and good accuracy. The simulation has in-built tools for their generation. Random number has the sequence of values between O and 1.

(iii) Report Generator:-
The large chunks of the calculations and statistics so come in the output are also evaluated automatically and report is also generated at the end of the finish of the run.

(iv) Dynamic Memory management:-
As there are numerous entities in different activities during the execution of the simulation modelling, we are dealing both new items and also with the outdated ones. Most of the simulation models have in built feature that have automated collection of the garbage values to remove rarely used items.

6. CLOUD COMPUTING

Cloud computing has all the hype right now but what does it really mean? Cloud computing is pretty big and it's going to be way bigger with every coming day. There are basically three layers in the cloud computing, company's use them differently based what they offer. At the bottom there is the foundation the Infrastructure, this is where the people start and more people begin to build. This is the layer where cloud hosting lives. Let's discuss about hosting in the cloud which is an older concept that due to technology advances is finally starting to work really well. Let's say we have a website and the website has lots of short conversations among the members and gradually your company gets popular and as the customers are more so does is the demands and you may be in trouble as the company cannot keep up with load of demand that is incoming. A few years ago if faced this kind of problem, you would have put your website on a computer or server somewhere and as the success kicked in you had to buy new servers and set them up. Many companies are doing it now but it cost lot of money and time. This is called Hosting, and to maintain it lot of time and money goes in. Moreover you have to pay when you are using it and also when you are not using it.
Now you have a better option "cloud computing " now we instead of putting it on the your server you could put it on the cloud server and the when people start visiting your site if u need more computing power as you need it from the cloud and your customers remain happy without noticing any difference. And similarly when you want to decrease the computing power you could release it easily from the cloud just as easily. Billing is easy too, it very similar to the bills of the water and electricity or getting in a taxi, meter runs when you ride and stops as you done with the ride. So the motive is to understand that how we should buy whole taxi when we could hire it for some time and pay the rent.

7. OPENSTACK

Openstack [13] is open-source software for managing cloud [13]. It is a product of Rack space and NASA [12]. We can also refer it as the operating system for cloud. Openstack is the open source version of the Eucalyptus or Open Nebula [11].For getting the hold of Openstack we should get started with what IaaS look like. We got storage which is build to be with no of raid and other technologies, networking build using VSS (Visual Source Safe), multi-link and high availability ability. Manage the Virtual Machines and manage clouds and tenants, customers and multiple group of virtual machines is handled by compute part. In the past we have built extremely available infrastructure whether at network layer or storage layer.

This method is traditional but they were not so flexible but only to run them efficiently. So, what if you have less knowledge of infrastructure and good efficiency not to evaluate something differently. And proprietary closed and not easy to get access to.

Now when we compare the cloud to OpenStack. It has couple of very different properties. OpenStack is fully Open source and it's completely community based so there is no need to collaborate with other enterprises and other vendors companies completely open.

The second thing is that as being built on applications that may have been legacy applications, its defined for new type of applications like web based applications, cloud applications, the applications that expect to scale dynamically, scale up and scale down, how to deal with an infra that may fail from time to time like design fail, so we will be not discussing about the legacy applications here.

Openstack is would mainly contain components, the first is the “glance” it is the code name of the image repository, the thing that will keep the track of your bills, all the things you need to create to pull together the applications. The second piece is Nova, it focuses on the compute layer, how do I get a commodity or any x86 platform to be able to boot or load images and to do compute function and third thing is Swift, it's the project name for storage. The storage piece instead of being file based or file based in IaaS it's got to be object-based so again it's so to its HTTP based outs and gets. The applications are getting highly intelligent that is being provided by the OpenStack, though it's less available.

8. NETWORKING IN OPENSTACK

So if we see the C version of the Openstack that is the Cactus it defines the network part having Layer 2. Hence, the network is very simple and flat. Thus we could say it's a NetStack that we will be talking about. In today's OpenStack environment it focuses on two types of the areas; one is Layer 2 and also focused on L3-L7.

So when we talk about the NetStack we have to see 2 things
(i) Quantum - tends to be bucket that holds all the services that focus on L2.

(ii) Dona is that is concerned with L2-L7 services, types of the containers.

In quautmn space they are really looking at sort of providing greater formality and definition around those more simplified layer 2 services, applications. So they just focus on the address space management with IP addresses, MAC addresses dealing with addresses moving. The second thing they trying to do are to connect to VM and the attributes of the VM with the network. Hence more simplified layer 2 service called Quantumn and we have the ability for applications to basically call that within OpenStack.

Separating that from other areas, Cisco is primarily working on "Donanbe" focusing on the containers. Now containers here means the idea that there are more than basic services that the network can provide like load balancing services, security services, firewall services and application acceleration, IDS things that we traditionally sold in the network but in automated way So we need not go in the depth of these but we have to see the container in the terms of an Application Center so that I need to be highly secure or highly available or have greater bandwidth than many other applications. However it can be described in any way but it we want to provide them in a programmed way, an abstract programmable way of all the services through which we can request certain kind of layer 2 through other layers so it can be VLAN, Load balancing, security, whatever be it, a programmatic way to do that and the way there defining that is called Network containers. And also, ultimately manage these containers, how to create them, control them and so on.

9. CONCLUSION

In this paper, we have discussed about the performance modeling in detail. All the three approaches of performance modeling have been described with the reasons and situations of selection for each of them. We presented an overview of the cloud and OpenStack. And we infer that, each of the approaches has its own significance and thus for most appropriate designing of the performance modeling of an application on OpenStack each of these should be incorporated. OpenStack is an extremely vast project, so opting for any one of these could result in leaving the result to be incomplete i.e. ample of scope is present, which can be explored.

10. ACKNOWLEDGMENTS

I am grateful to my mentor who guided me at every step and the institution to encourage and avail all the resources I needed in order to complete this paper.

11. REFERENCES

[1] Thakkar, Dharmesh, et al. "A framework for measurement based performance modeling." Proceedings of the 7th international workshop on Software and performance. ACM, 2008.

[2] Ajitha, S., TV Suresh Kumar, and K. Rajanikanth. "Framework for Multi-Agent Systems Performance Prediction Process Model: MASP 3." Our Indexing at International Level 4852: 774.

[3] “Wanted: Performance Management That Measures Up, and OpenStack Delivers." The Path to Open Hybrid Cloud. N.p., n.d. Web. 24 Jan. 2015.

[4] Mania, Daniela, and John Murphy. "Framework for predicting the performance of component-based systems." Proceedings of the 10th IEEE International Conference on Software, Telecommunications and Computer Networks, Croatia-Italy. Vol. 181. 2002.

[5] "Performance Engineering Fundamentals | Performance Modeling 101."Practical Performance Analyst. N.p., 30 Sep. 2013. Web. 23 Jan. 2015.

[6] Altevogt, Peter, Wolfgang Denzel, and Tibor Kiss. "The IBM Performance Simulation Framework for Cloud." IBM Research Technical Paper http://domino. Research. IBM. com/library/cyberdig. nsf/papers.

[7] "Modelling." Software Performance Engineering. Rajesh Mansharamani, n.d. Web. 04 Feb. 2015.

[8] Wagner, Mark, Senior Principal Engineer Red Hat, and 201 June 12. Red Hat OpenStack Performance & Scale (n.d.): n. page. Web. 11 Feb. 2015.

[9] "Performance Modelling Services." - Testing Services. N.p., n.d. Web. 22 Jan. 2015.

[10] Jane Hillston. 13 Simulation Models—Introduction and Motivation (n.d.): n. pag. 20 Oct. 2014. Web. 29 Jan. 2015.

[11] Oliveira, Rui, et al. "Automatic Elasticity in OpenStack." SDMCMM'12: Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management, with ACM/IFIP/USENIX ACM International Middleware Conference. 2012.

[12] Wen, Xiaolong, et al. "Comparison of open-source cloud management platforms: OpenStack and OpenNebula." Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on. IEEE, 2012.

[13] "OpenStack Has a True Community around It."*Home » OpenStack Open Source Cloud Computing Software*. N.p., n.d. Web. 18 Jan. 2015.