Detecting workload hotspots and dynamic provisioning of virtual Machines in Clouds

Narsimha Reddy CH
IBM India Software Labs
IBM India Pvt. Ltd.
Hyderabad, India
narsimha.reddy@in.ibm.com

***Abstract*—One of the primary goals of Cloud Computing is to provide reliable QoS. The users of the cloud applications may access their applications from any Region. The cloud infrastructure must be Elastic enough to improve the QoS requirements. In order to provide reliable QoS, the cloud infrastructure must be able to detect the potential workload hotspots for various cloud applications across Regions and take appropriate measures. This paper presents an approach to detect workload hotspots using application access pattern based method in the cloud. This paper also presents how the existing VDN based Virtual Machine provisioning approach [1] can be used to provision new Virtual Appliances at the detected hotspots dynamically and efficiently at the potential hotspots to improve the QoS.**

***Index Terms*—Cloud, Deprovision, Dynamic Provisioning, Elastic, Global Load Balancing, Image Distribution Network, Latency, Load, Migration Region, Performance, Provision, QoS, Remote Region, Response Time, Throughput, VDN, Virtual Machine, Virtual Appliance, Work Load Pattern.**

I. INTRODUCTION

A cloud can contain multiple Regions and each Region contains a data center. Regions could be geographically separated entities (can be across continents). In clouds, generally a component called ***Cloud Infrastructure Controller*** handles the management (provisioning/deprovisioning) of Virtual Machines or Appliances upon user requests. The applications running on the Virtual Infrastructure can be accessed by customers at any time from any geographic Region. Generally, the QoS can be defined in terms of Service Level Agreements (SLA) that describes such characteristics as Minimal Throughput, Maximal Response Time or Latency delivered by the Cloud. The response time and latency are very important factors for measuring the performance of applications in the Clouds. A study by Microsoft Research Center based on admission control heuristics reported that Round Trip Time (RTT) is the main and basic network measurement metric for judging the real performance of host and server (Gunawarden and Massoulie, 2006). Seshan et al. (1997) reported that users often make decisions on the basis of Response Time from hosts.

In this paper, we present methods to improve the QoS by detecting potential workload hotspots. Also, we present the existing VDN based VM Provisioning method to dynamically provisioning of VMs at potential hotspots.

II. PRIOR SOLUTIONS

For improving QoS, generally the clouds will deploy applications (appliances) in multiple Regions (or Data centers) beforehand. The Global Load Balancing Agents will route the traffic to the nearest Region depending on the user proximity or load or various dynamic network or application parameters. With in a Region, the Local Load Balancing Agents will provision additional appliances based on current load i.e. if the load on a particular application increases a threshold value, then the Local Load Balancing Agent may provision additional Virtual appliance(s) in that Region and distributes load across them. While this method may not improve the response time for users from Remote Regions due factors like large RTT (Round Trip Time), network congestion etc.

III. PROCEDURE FOR DETECTING WORKLOAD HOTSPOTS

The Global Load Balancing Agents will monitor load and response time for applications across Regions. The proposed method defines a few terms w.r.t an application,

- **Application's Hosting Region** - This is a Region where the application is originally hosted.
- **Application's Remote Region** - This is a Region in which no application instance is running. The user requests for application originating from Remote Regions will be routed to a nearest Hosting Region by Global Load Balancing Agents.
- **Request Density per Application per Remote Region** (RDAR) - This factor represents the no. of requests from a Remote Region per application per unit time.
- **Global Load Balancing Agent**: This will load balance across regions.
- **Local Load Balancing Agent**: This will take care of load balancing with in a region.

978-1-4673-4422-7/12/$31.00 ©2012 IEEE

The Global Load Balancing Agents will do a book keeping of the requests for each application from each Remote Region using the factor "Request Density per Application per Region (RDAR)" (defined above). If the Request Density for an application from a Remote Region(s) is more than a threshold, it can cause the increase in Response Time (violating SLA requirements). The Remote Region may require one or more instances of that Application to be provisioned in the Remote Region.

This paper tries to address the problem by presenting two methods:

1. Analysis of Remote Server Performance Impact [3]
2. A workload pattern based method to detect potential Remote Regions in which new application instances to be provisioned.
3. Describes how to optimally provision the new Virtual Appliances using VDN [1].

The subsequent sections of this paper will explain how this approach will improve the overall response time and latency for applications running in geographically distributed large scale clouds with supporting experimental results.

A. ANALYSIS OF REMOTE SERVER PERFORMANCE IMPACT [3]

Generally, the performance of remote applications are measured according to response time, latency, reliability , availability, delay and jitter that are affected under loaded and congested situations[3]. A research conducted [3] reveals that latency of Remote Applications depends upon various factors like RTT (Round Trip Time), load, congestion, and hop counts. The experiments also revealed that, in case of network congestion, path is changed, which may result more number of hop counts penalty that caused much increase in latency/RTT. Similarly, in case of high load, RTT is increased, which may lead to congestion and as a result application's response time is increased.

The research [3] conducted a real time experiment on Yahoo, Gmail and Hotmail servers to measure the response time under normal, average and heavy network loads (see below graphs). It is observed that the response time increased drastically under heavy loads.

The relationship between the RTT and Hop Count is depicted in Fig. 1.

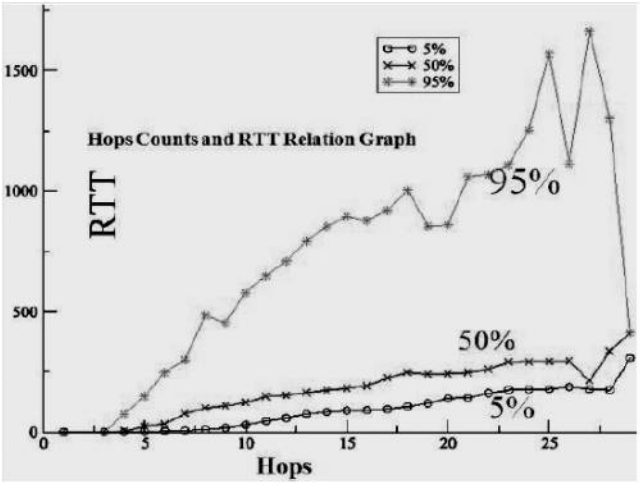

Fig 1. Relationship between the RTT and Hop Count

	Normal (5%)	Average (50 %)	Maximum(95%)
Yahoo	0.44ms	0.50ms	**0.77ms**
Gmail	0.36ms	**1.25ms**	0.81ms
Hotmail	1.09ms	2.39ms	**4.71ms**

Using rounded values for Hotmail server, the RTT values will be.

Normal RTT 1ms
Max RTT 5ms

Per second :
Max trips under Normal Load = 1000
Max trips under Max Load = 200

Say, user burst happens for 20 mins at the rate of 500 reqs/sec
Total request over the burst = 20 x 60 x 500 = 600,000 reqs.

Under normal loads this will be taken care of by the default deployment since the available servers allows 1000 trips per sec and it will be completed during the user burst itself at only 50% loading.

However, under Max load / Max latency, the available capacity is only of 200 trips per sec giving an overhead of 300 requests per second, hence the completion time will increase to 250%.

The above analysis based on experimental results [3] proves that under loaded/congested conditions of the network the response time increases drastically.

1

IV. AN ACCESS PATTERN BASED METHOD TO DETECT APPLICATION WORKLOAD HOTSPOTS

To improve the Response Time for applications, this method proposes a method to detect potential application workload hotspots by studying the application access patterns.

A workload access pattern can be defined as:

(Application Instance) : (Hosting Region) : (Avg no. of requests arriving per unit time from a Remote Region) : (Avg. response time or latency for a request for each Remote Region)

For example, let there be 5 regions A, B, C, D & E. Consider a Web Application which is initially hosted in one particular region A, say.

The usage patterns for regions B, C, D & E can be monitored by global load balancing agent,

Pattern for Region B:
(Web App Inst 1):(A):(1000 Req/Sec):(1 ms)

Pattern for Region C:
(Web App Inst 1):(A):(5000 Req/Sec):(4 ms) <--- more requests and response time

Pattern for Region D:
(Web App Inst 1):(A):(100 Req/Sec):(1 ms)

Pattern for Region E:
(Web App Inst 1):(A):(700 Req/Sec):(.8 ms)

From the above patterns, it is obvious that for Region C, it is required to migrate a virtual appliance (containing instance of Web Application) and needs to provisioned there.

The workload patterns for various applications will be monitored by Global Load Balancing Agents. The algorithm will analyze the workload patterns on a continuous basis. Based on the analysis, the method would recommend for potential new Remote Regions where the Virtual Appliances needs to be provisioned. Based on the recommendations, the Virtual Appliance images will be transmitted to new regions and provisioned dynamically.

V. Virtual Appliance Mobility and Provisioning through VDN

The deployment of Virtual Appliances can be performed by using VDN (Virtual Machine Image Distribution Network). Evaluation shows that VDN achieves as much as 30–80x speed up for large VM images under heavy traffic [1].

A study by IBM Watson Research and others **[3]** revealed that, compared to using the centralized image server, VDN (**V**irtual machine image **D**istribution **N**etwork) can achieve as much as 30–80x speed up, especially for those large VM images or when the traffic load is heavy. The speed up is achieved at a low maintenance cost. The table below depicts the provisioning time for baseline (each host only fetches VM image chunks from the central image server) and VDN images.

	128 MB	512 MB	2 GB	4 GB	8 GB
Baseline(s)	6.9	46.6	522.4	3620	29489
VDN(s)	0.6	3.5	17.2	80.1	433
Speedup	10.5×	12.3×	29.3×	44.2×	67.1×

TABLE IV
PROVISION TIME FOR VM IMAGES WITH VARIOUS SIZES

Fig. 2. Provisioning time for images with various sizes

Based on the above research results, assuming an 8GB image takes around 8 mins to come up, the max loading scenario from above will work out as (assuming limits - need to work out the difference equation):

Overhead during 8 min startup = 8 x 60 x 300 = 2400 x 60

After 8 mins. the combined capacity is 1000 local (possibly better) and 200 remote = 1200 total

So the service time assuming high priority associated with current requests we have 700 extra serviceable trips available. At this rate the backlog will be cleared out in 4 mins from local VM deployment; i.e. 12 mins from start of cloud burst Beyond the 12 mins the local VM will be sufficient to service the burst and there will no backlog, since the network loading will have been rendered inconsequential.

A. Special Case: A Study of Web Server [4]

A research conducted by IBM Watson Research Center [4] revealed that, the web server performance is a critical issue for sites which service a high volume of requests. The research examined the performance of a Web server under high CPU loads. The results shown that the performance is significantly affected by the percentage of requests for dynamic HTML pages (fcgi); dynamic HTML pages adversely affect server performance. The below graph [4] shows that the Maximum throughput which can be sustained by different types of networks.

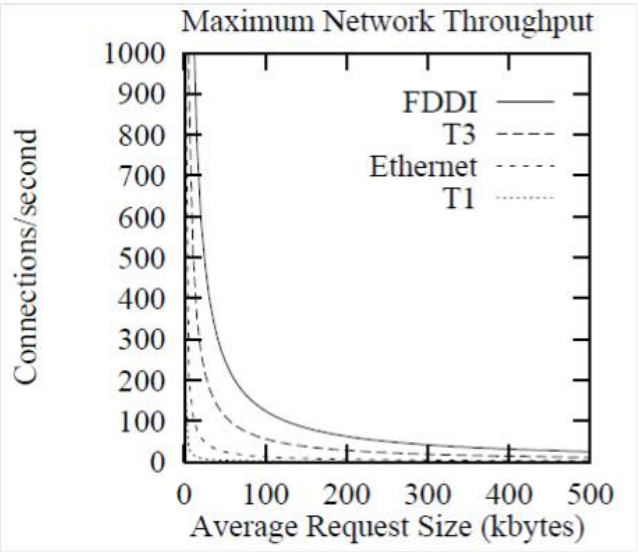

Fig. 3. Maximum throughput sustained by different networks

The below graph shows how the Average latency increases based on the proportion of Dynamic Pages.

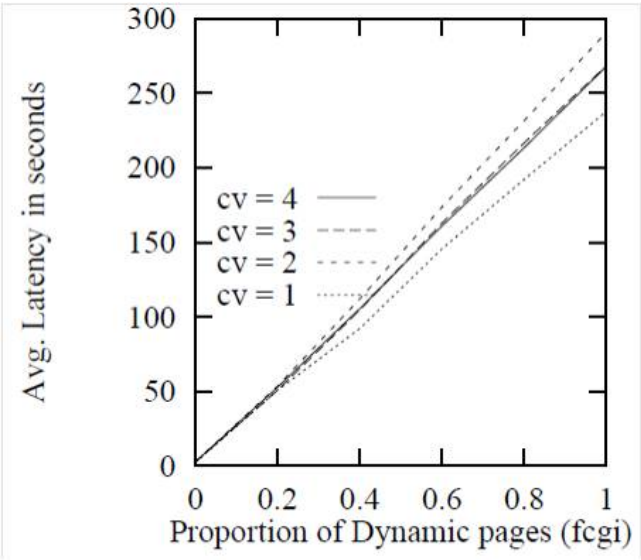

Fig. 4. Average latency vs. proportion of dynamic pages

By combining the results of two papers [2] and [4], consider the web server is located in a Remote Region which has more network latency and hence response time. The total delay will be even more under average workloads,

Total Latency = Network Latency to Remote Region+ Server Latency

This will adversely affect the response time of Remote applications. In such situations, the proposed solution to detect hotspots and dynamic provisioning of Virtual Appliances will be useful. Basically, the web server can be dynamically provisioned in Remote Regions for load balancing purposes. This will reduce the load on the network and also application which improves the response time for end user.

VI. Advantages

This solution can be used in situations where there are varying loads on application servers during different periods of the year. Dynamically provisioning applications on demand in remote Regions will serve better in such cases. Cloud users can subscribe to this service and they can carry their applications/data to any region of the Cloud. This is something like virtually carrying their applications across cloud where ever they want.

VII. Conclusion

Not all customers can afford to the existing load balancing solutions where the application needs to be deployed in multiple data centers before hand, because, it may cost more.

The study [2] reveals that, under high network load situations the performance of Remote Applications drastically reduces. The study [4] reveals that when the percentage of dynamic content is high, the latency of the server increases adversely. Also, if the RTT is increased, then also the response time or latency increases. In such situations, the Global Load Balancer can identify the potential Remote Regions to provision the new Virtual Appliance and provision them on the fly using VDN. Later, when there is reduced load and RTT, then the virtual machine can be shut down/hibernated or deleted in respective Regions. Dynamic provisioning/deprovisioning of Virtual Machines at different regions dynamically is a potential new approach to improve QoS which will give flexibility to Cloud Service Providers to optimally provision Virtual Machines across their Cloud.

VIII. References

[1] VDN: Virtual Machine Image Distribution Network for Cloud Data Centers Chunyi Peng, Computer Science Department, University of California, Los Angeles chunyip@cs.ucla.edu
Minkyong Kim, Zhe Zhang, Hui Lei
IBM Watson Research
Hawthorne, NY, USA
fminkyong, zhezhang, hleig@us.ibm.com

[2] Amazon's EC2 provides load balancing with in a Region and does not have the capability to provision Virtual Appliances on Demand based on request density per Region.

[3] An experimental performance evaluation of different remote servers to analyze the effect of divergent load and congestion
Ijaz Ali Shoukat* and Mohsin Iftikhar
Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, KSA.

[4] An Analysis of Web Server Performance
Arun Iyengar, Ed MacNair and Thao Nguyen
IBM Research Division
T.J. Watson Research Center
P.O.Box 704
Yorktown Heights, NY 10598
Accepted 08, June 2011