

# Putting Critical Applications in the Public Cloud

The Very Latest Best Practices & Methodologies

**Business White Paper**

June, 2011

## Introduction

Many organizations are beginning to realize that there are significant advantages to moving their business-critical applications to the public cloud. With competitive pressures becoming stronger all the time, the heat is on to make development cycles more agile and applications more dynamic.

One of the most famous of these companies is Netflix, which is currently deploying its business-critical applications in the public cloud and running thousands of nodes on Amazon EC2. Netflix realized they could not innovate rapidly if they stayed in their data center; they needed the public cloud in order to drive explosive growth. Many companies have come to the same conclusion and are rushing to join Netflix in the cloud.

But you might wonder: what are the ways to avoid the pitfalls of moving apps in the cloud in the first place, ensuring that the migration can be done with minimal risk? And once the migration is accomplished, what's the best way to maximize the cloud's much-touted benefits?

The following white paper is intended for developers, operations, and architects who are struggling with these questions. It offers best practices and tips for ensuring that the migration doesn't degrade application performance, as well as how to make those applications perform at the highest level possible through *self-tuning*.

### MIGRATING WITHOUT FEAR: INTRODUCING THE RATU

The data center is warm. The data center is cozy. It's familiar. And you are no doubt aware, transitioning distributed applications to the fast-changing environment of the cloud is a complex and risky process. How do you move out of the safety of your data center, where you have successfully run for years (if not decades), without sacrificing the performance of your applications?

There is no easy way to predict your application's performance on cloud resources. With such technical names as "small," "medium," and "large," how can you even begin to estimate capacity needs for your application? The cloud is a mysterious place with computing resources that bear no resemblance to the systems in the data center, and a successful transition will require arduous analysis.

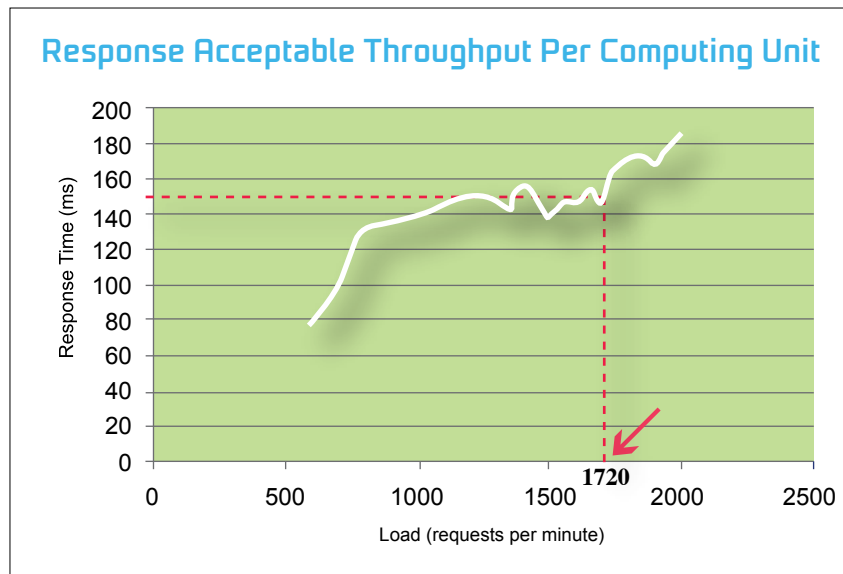
But if you use Analytics on top of your rich Application Performance Management (APM) data, you can do all this in a few straightforward steps. By comparing performance data from your current application to that from your test environment in the cloud, you can accurately estimate your cloud capacity requirements.

There are some key prerequisites before moving forward. First, you need access to performance data from your APM system (you do have an Application Performance Management system, don't you?). Second, you need to test your app with simulated load on the cloud environment that you intend to use. Most importantly, your APM system must run on both old and new environments and record the performance characteristics.

Our goal is to compare the capacity of a machine in our data center against the capacity of computing nodes in the cloud to discover just how many nodes we'll need to match the performance of our data center. To do this, we compare the throughput of a single application running in our data center against the throughput for the new cloud environment.

The unit of measurement to use for this is RATU — the Response Acceptable Throughput per computing Unit. Let me explain.

The graph below shows measured Response-Times at varying levels of load (Throughput). Once we decide what Response-Time is currently acceptable for our application, we can find the largest load that meets this limit within, say, 2 standard deviations. Now simply divide this throughput number by the number of machines the application is running on. (Note: for simplicity I'm assuming the hardware is uniform, but if it's not then you can compute a weighted average.)



Now, as the application isn't just one atomic unit, you'll need to measure the throughput for each high-level transaction. You need to perform this calculation across all your Business Transactions to get an accurate result. It is important to examine individual Business Transactions to ensure every business facing part of your application has been accounted for. Calculate the RATU for each of your relevant Business Transactions and take the average. You can customize this process a bit by excluding some transactions or even by doing a weighted average based on the importance or volume of the Business Transaction.

Now run the same analysis (calculating the RATU) on your test environment in the cloud.

To get accurate numbers, you'll want to take a measurement over a long enough time to cover at least one full workload cycle, if not more. As a rule of thumb I'd recommend a month of data, as that should cover most variations in the application's lifecycle.

You should now have 2 numbers: RATU for the old data center environment and RATU for our cloud. Simply divide the first by the second to see what percentage more cloud units you'll need to achieve the same performance as your current environment.

Let's try an example using Amazon's elastic cloud (EC2). Say we are looking at an "Asset Tracking" application. We have 100 large computers in our data center running this application. We also ran a test in the cloud of this same application, on 20 "large" size ec2 nodes. We've run analysis on one month of data and found that our data center RATU is 10,000, while our RATU in the cloud is 1,000. Therefore:  $10,000 \text{ divided by } 1,000 = 10$ .

This means we need 10 times more cloud units than data center hardware units we currently use to achieve satisfactory throughput. Since we were testing on 100 datacenter machines, we now know that we need 1000 "large" nodes if we want to achieve the same throughput when we migrate to the cloud. (Make sure you have enough money for all those nodes before you start!)

You may be saying to yourself, wow this would be great, if only my APM system actually had analytics. And that's the key: your APM solution needs to be built from the ground up for the challenges of cloud environments. If it's not, it won't be able to accomplish what you need—and your cloud initiative will be stuck in neutral.

## MAXIMIZING THE CLOUD: THE POWER OF THE SELF-TUNING APP

So now you've successfully moved your applications to the cloud. But how do you know that you're leveraging all of the cloud's benefits?

You probably decided to make the trek in the first place in order to take advantage of the dynamic nature of the cloud with the ability (and agility) to quickly scale applications. Your application's load probably changes all day, all week, and all year. Now

your application can utilize more or less resources based on the changes in load. Just ask the cloud for as much computing resources that you need at any given time, and unlike at data centers, the resources are available at the push of a button.

Or, at least, so the marketing video tells you. But in the real world, no one can find that magic button to push. Instead, scaling in the cloud involves pushing many buttons, running many scripts, configuring various software, and then fixing whatever didn't quite work. Oh, and of course even that is the easy part, compared to actually knowing when to scale, how much to scale and even what parts of your application to scale. And this repeats all day, every day, at least until everyone gets discouraged.

So is it not true that you can gain the scaling benefits of the cloud, contrary to what the hype would have you believe? No, the benefits of cloud scaling are real—but in order to get them, you need to make your application self-tuning.

For the application owner—the individual in charge of uptime and performance of mission-critical applications—the self-tuning application is the true holy grail of cloud-based performance. Luckily, the technology is finally available and I'll show you how to use it. But first, let's take a quick look at the evolution of scaling in the cloud.

The first generation of technology wasn't really technology; it was manual. Early adapters of the cloud would observe performance of their application and make decisions to add more nodes. Then Ops staff would do the same work that they had done in the data center, installing and configuring software on the new nodes, changing load balancing scripts to include the new nodes, and so on. The only benefit here from the cloud was that you saved time and money by not having to buy hardware for your data center. The data center no longer throttled you, but your staff did, as you could only scale as fast as your team could finish the manual work. The difficulty and time required to do the manual work limited how often you could scale, so while you might be able to scale every few days once you recovered from the previous push, you could not quickly react to sudden spikes in load, or even to daily fluctuations in your application's usage pattern.

Recently, the second generation emerged. Both startups and cloud providers began offering software to scale applications automatically based on performance. Unfortunately, these lacked some key elements like: a) the ability to collect complete business relevant performance data, and b) the ability to make decisions based on processing that large amount of data. And so these systems can only measure simple metrics like CPU, memory or heartbeats, and then make haphazard decisions based on them. But how well is your application really performing now? Is it meeting key performance based business SLAs? What effects have past scaling attempts had on your application? These simple metrics didn't shed much light on this. The tools started scaling your application without really understanding it.

Sounds a bit scary doesn't it? It should, but many continue to learn the hard way. I recently worked with a large Internet-based company who used such auto scaling technology for the first time. Because the tool was measuring metrics that gave false positives, it went crazy and began to crank out new nodes every minute, greatly stressing the application environment for days while teams investigated the problem.

With all that behind us, we are ready for the third generation of scaling technology, the self-tuning app. Three technologies must work together to make this possible.

First, Cloud-Ready Business Transaction Monitoring. Your monitoring system must be focused on business transactions so that it can be cloud aware. In doing so, it can automatically discover and understand the distributed nature of your application in the cloud. Static monitoring systems will not work if you really have a dynamic application that is constantly changing and taking advantage of the various cloud resources. And the system must also be business aware; distributed business transactions are the key unit of measure when looking at your application. You are not interested in various low level CPU metrics; you are interested in the business performance of your application, as that is the ultimate measurement of success.

Second, Analytics on top of this rich monitoring data. A good monitoring system such as I've described above will collect large amounts of data, which means you'll need an analytics system which can help in performing long range analysis and trending to understand the behavior of your application and understand the impacts that scaling has on your application. You can then gain key insights and detect patterns that help you predict when scaling will be required. As a result, you can track your scaling history to better optimize each scaling operation.

Third, Cloud Orchestration. You'll need a system which can take all the information you have and actually perform the scaling operations. It should work with a wide array of cloud providers and be extensible for your custom needs. It must also be closely integrated to your monitoring and analytics system so each is aware of each other's actions. Your system knows that a threshold has been reached and initiates a scaling operation, the cloud orchestrator is fed all data from the monitoring system so it has the latest information about the various parts of your application, and then the system knows when the scaling operation took place and can measure the results from the operation and make further decisions about when to scale again.

As you see, these three systems work tightly together to study your application, scale it, and study some more. Taken together, they can be considered to be a true Application Performance Management system—which needs to consist of Cloud-Ready Business Transaction-centric Monitoring, Analytics, and Orchestration. This powerful APM engine improves the efficiency and effectiveness of the scaling, allowing your application to perform optimally in the cloud without much human intervention. And now you have a self-tuning application!

The cloud really does offer amazing opportunities, as long as you have the technology to harness that power. Key among them is the self-tuning app.

## CONCLUSION

Companies recognize that they can get amazing agility by moving their apps to the public cloud. They want to take advantage of its dynamic nature of the cloud and gain the ability to scale easily.

Once a company decides to make the move, it's important to employ best practices to ensure that application performance doesn't degrade inside the new environment. This can be done by carefully monitoring application performance before the migration as well as after the migration, using a system known as RATU — the Response Acceptable Throughput per computing Unit.

Once the migration has been accomplished, those apps should take advantage of their new environment by employing an Application Performance Management (APM) technology and becoming self-tuning, using a combination of business transaction monitoring, analytics, and cloud orchestration. Doing so will remove a great deal of burden off the Operations team and free up time that can be used for innovation and technological differentiation.

In regards to both the migration itself as well as putting the processes in place for a self-tuning app, the company will need to have the right APM solution to ensure strong application performance in the cloud environment.



### ABOUT THE AUTHOR

Boris Livshutz works in the Server Technologies group of AppDynamics, helping customers monitor and manage application performance management in the cloud—with many deployed in public clouds with well over two thousand Java Virtual Machines.

### ABOUT APPDYNAMICS

AppDynamics is the leading provider of application performance management for modern application architectures in both the cloud and the data center, delivering solutions for highly distributed, dynamic, and agile environments. Companies such as Priceline, TiVo, and ZipRealty use AppDynamics to monitor, troubleshoot, diagnose, and scale production applications, and over 30,000 people have downloaded AppDynamics Lite at [appdynamics.com/free](http://appdynamics.com/free). The company was recognized as a Gartner Cool Vendor in IT Operations Management. Try our free java performance solution at [www.appdynamics.com/free](http://www.appdynamics.com/free) or find out more at [www.appdynamics.com](http://www.appdynamics.com).



### AppDynamics

303 Second Street, Suite 450 | San Francisco, CA 94107  
[www.appdynamics.com](http://www.appdynamics.com)