An AppDynamics Business White Paper
October, 2013

Managing the Performance of Cloud-Based Applications

Taking Advantage of What the Cloud Has to Offer – And Avoiding Common Pitfalls

AppDynamics

Moving your application to the cloud isn't as simple as porting over your code and configurations to someone else's infrastructure – nor should it be. Cloud computing represents a paradigm shift in the world of application architecture from vertical scalability to horizontal scalability. This new paradigm has much to offer organizations that want to build highly scalable and dynamic applications, but it has its dangers, too – if you're not careful and purposeful in how you prepare for the cloud, your application could suffer. In this white paper, we'll discuss how to reap the performance benefits of the cloud and avoid the common pitfalls. But first, let's look at how and why the cloud is so different from anything we've seen before.

Horizontal vs. Vertical Scalability

The biggest fundamental difference between the cloud and your data center is that the cloud is usually run on commodity hardware, rather than the powerful machines you probably use in your data center. This means you have to write an application that's **horizontally scalable** instead of **vertically scalable**. Google probably best described what this means for your application architecture in their blog on highscalability.com:

> A 1,000-fold computer power increase can be had for a 33 times lower cost if you use a failure-prone infrastructure rather than an infrastructure built on highly reliable components. You must build reliability on top of unreliability for this strategy to work.

In other words, it can be cost effective to run an application on cheaper, less reliable commodity servers instead of more expensive and powerful machines. But in order to be successful, the software component needs to be highly scalable – even infinitely scalable – and resistant to failure. These two requirements guide many of the architectural decisions of cloud pioneers like Netflix, and give a good indicator of what's required to be successful in the cloud from a performance standpoint.

Architecting for the Cloud

Very few organizations have the same requirements from their applications that Netflix does. With tens of thousands of nodes in the Amazon EC2, Netflix is undoubtedly a pioneer in cloud architecture. Even though most organizations will never need the scale that Netflix does, these architectural practices and strategies are relevant to anyone building in or migrating to the cloud.

Here are a few of the ways Netflix takes advantage of the cloud, from a presentation by Netflix's Director of Cloud Solutions, Ariel Tseitlin.

Service Oriented Architecture

The easiest way to accomplish horizontal scalability is with a service-oriented architecture. This is already pretty commonplace, but it's especially important in the cloud, where you pay for the resources you consume – as your application scales, you can scale out only the services you need, which is more efficient (and cheaper) than scaling everything across the board. In addition, service-oriented architectures help manage concurrency. The following two diagrams demonstrate this point.

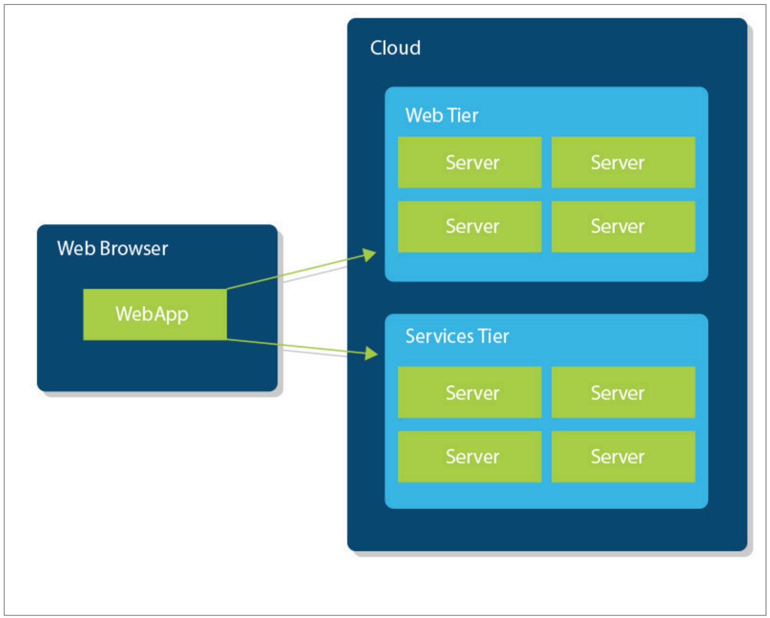

FIG. 1: THE ENVIRONMENT UNDER NORMAL LOAD

Auto-scaling

In the data center, scaling your application is expensive and time-consuming. In the cloud, it's easy and (relatively) cheap. Netflix takes advantage of this, scaling their application up during the evenings when load increases and back down when peak viewing hours are over. Anyone with very dynamic load can take advantage of auto-scaling, which allows you to be cost-effective in the cloud without sacrificing performance.

Planning for Failure

One of the techniques Netflix is most famous for is simulating failure with its Simian Army. While this might not be a feasible approach for everyone, planning for failure is important for any cloud-based application – this is what Google is referring to when it talks about building "reliability on top of unreliability." Your application needs to be able to survive failure at multiple levels – an individual node, a cluster, or perhaps even more.

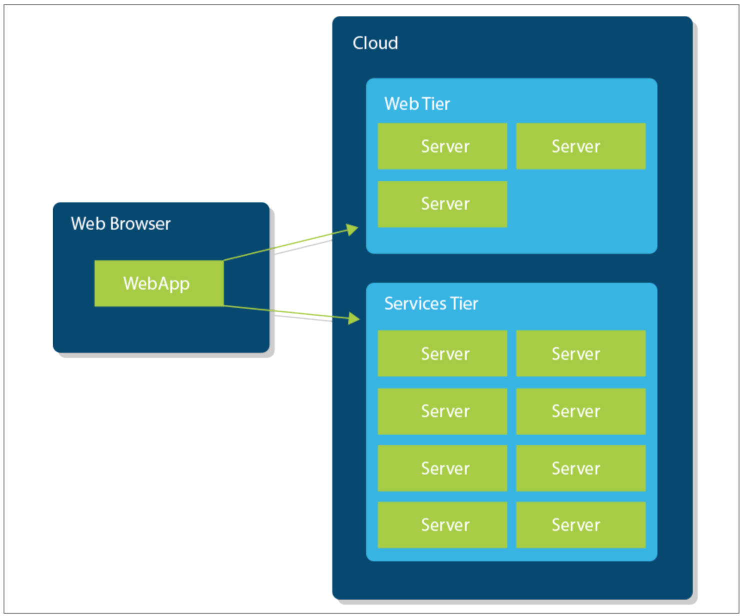

FIG. 2: THE ENVIRONMENT AFTER LOAD CHANGES

Stateless/Cluster Grouping

Server instances can come and go at the drop of a hat, so they must not store any state. Instead, Netflix groups server instances together into "clusters" and considers the behavior of the cluster as a whole.

Asynchronous Communication

Most highly scalable applications use asynchronous communication to ensure their applications never go down even as load increases. Asynchronous communication, often implemented with enterprise service buses (ESBs), allows the application to complete certain tasks whenever it gets around to them, so that the user doesn't have to wait for a response. While this technique is very scalable, it's important to note that it shifts the application paradigm from being always consistent to being eventually consistent, meaning the application may not reflect the most recent changes right away, but it will eventually catch up.

These are just a few of the best practices for building a highly scalable cloud application. In order to take advantage of what the cloud has to offer, you need to rethink the architecture of your entire application. This also means that you need to rethink your approach to managing the performance of your application in order to ensure the availability and performance of your application in the cloud.

Managing Performance in the Cloud

As we saw in the last section, there are several architectural techniques you can use to build a highly scalable, failure-resistant application in the cloud. However, these architectural changes – along with the inherent unreliability of the cloud – introduce some new problems for application performance management. Many organizations rely on logging, profilers and legacy application performance monitoring (APM) solutions to monitor and manage performance in the data center, but these strategies and solutions simply aren't enough when you move into the cloud. Here are a few important considerations for choosing an APM solution that works in the cloud.

Business Transactions

Many monitoring solutions check for server availability and alert users when a server goes down. In the cloud, however, servers can come and go all the time, so alerting on availability will result in a lot of false positives. In addition, many of the server-level metrics that APM tools and server monitoring tools report are no longer as relevant as they were on a vertically scaled system. For example, what does 90% CPU utilization mean to the behavior of your cloud application? Does it mean there is an impending performance problem that needs to be addressed? Or does it mean that more servers need to be added into that tier? This goes for other metrics, too, like physical memory usage, JVM memory usage, thread usage, database connection pool usage, and so on. These are all good indicators of the performance of a single server, but when servers can come and go they're no longer the best approximation of the performance of your application as a whole.

Instead, it's best to understand performance in terms of Business Transactions. A Business Transaction is essentially a user request – for an eCommerce application, "Check out" or "Add to Cart" may be two important Business Transactions. Each Business Transaction includes all of the downstream activities until the end user receives a response (and perhaps more, if your application uses asynchronous communication). For example, an application may define a service that performs request validation, stores data in a database, and then publishes a request to a topic. A JMS listener might receive that message from the topic, make a call to an external service, and then store the data in a Hadoop cluster. All of these activities need to be grouped together into a single Business Transaction so that you can understand how every part of your system affects your end users. Figure 3 shows how this complex Business Transaction may be performed.

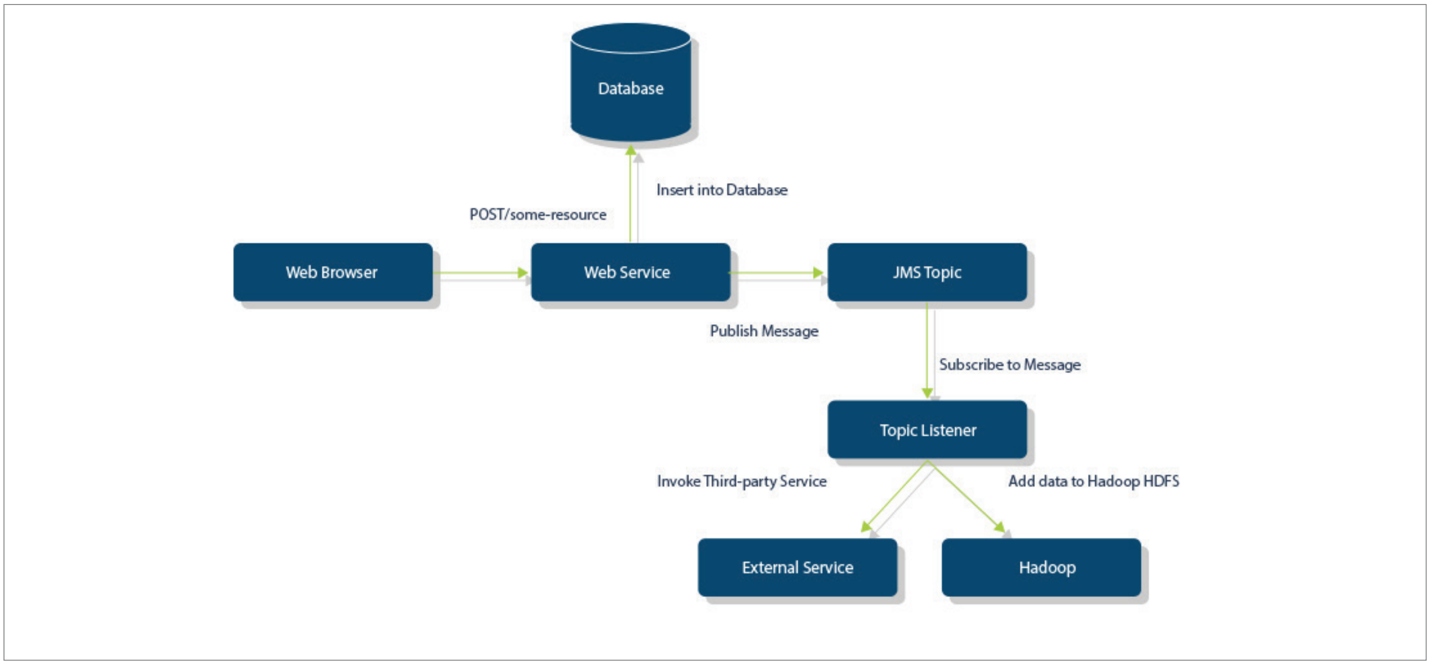

FIG03: SAMPLE COMPLEX BUSINESS TRANSACTION

Tiers

With these various tiers tracked at the Business Transaction level, the next step is to measure performance at the tier level. While it is important to know when a Business Transaction is behaving abnormally, it is just as important to detect performance anomalies at the tier level. If the response time of a Business Transaction, as a whole, is slow by one standard deviation (which is acceptable) but one of its tiers is slower by a factor of three standard deviations, you may have a problem developing, even though it hasn't affected your end users yet. Chances are the tier's problem will evolve into a systemic problem that causes multiple Business Transactions to suffer.

Returning to our example from figure 3, let's say the web service behaves well, but the topic listener is significantly slower than usual. The topic listener has not caused a problem in the Business Transaction itself, but it has slowed down enough to cause concern, so there might be an issue that needs to be addressed. Business Transactions, therefore, need to be evaluated both as a whole and at the tier level in order to identify performance issues before they arise. The only way to effectively monitor the performance of an application in a dynamic environment is to capture metrics at the Business Transaction level and the tier level.

Baselines

One of the most important reasons that many organizations move to the cloud is to be able to scale up and down their applications rapidly as load changes. If the load on your application fluctuates dramatically over the day, week or year, the cloud will allow you to scale your application infrastructure efficiently to meet that load. However, most application monitoring tools are not equipped to handle such dramatic shifts in load or performance. Application monitoring tools that rely on static thresholds for alerting and data collection will create alert storms when load increases and miss potential problems when it decreases. You need to be able to understand what normal performance is for a given time of day, day of the week or time of the year, which is best done by baselining the performance of your application over time.

Baselining your application essentially means collecting data around how your application performs (or how a specific Business Transactions performs) at any given time. Having this data will allow you (or your APM solution) to determine if your application is performing now is normal or if it might indicate a problem. Baselines can be defined on a per-hour basis over a period of time – for example, for the past 30 days, how has Checkout performed from 9:00am to 10:00am? In this configuration, the response time of a specific Business Transaction will be compared to the average response time for that Business Transaction over the past 30 days, between the hours of 9:00am and 10:00am. If the response time is greater than some measurable value, such as two standard deviations, then the monitoring system should raise an alert. Figure 4 attempts to show this graphically.

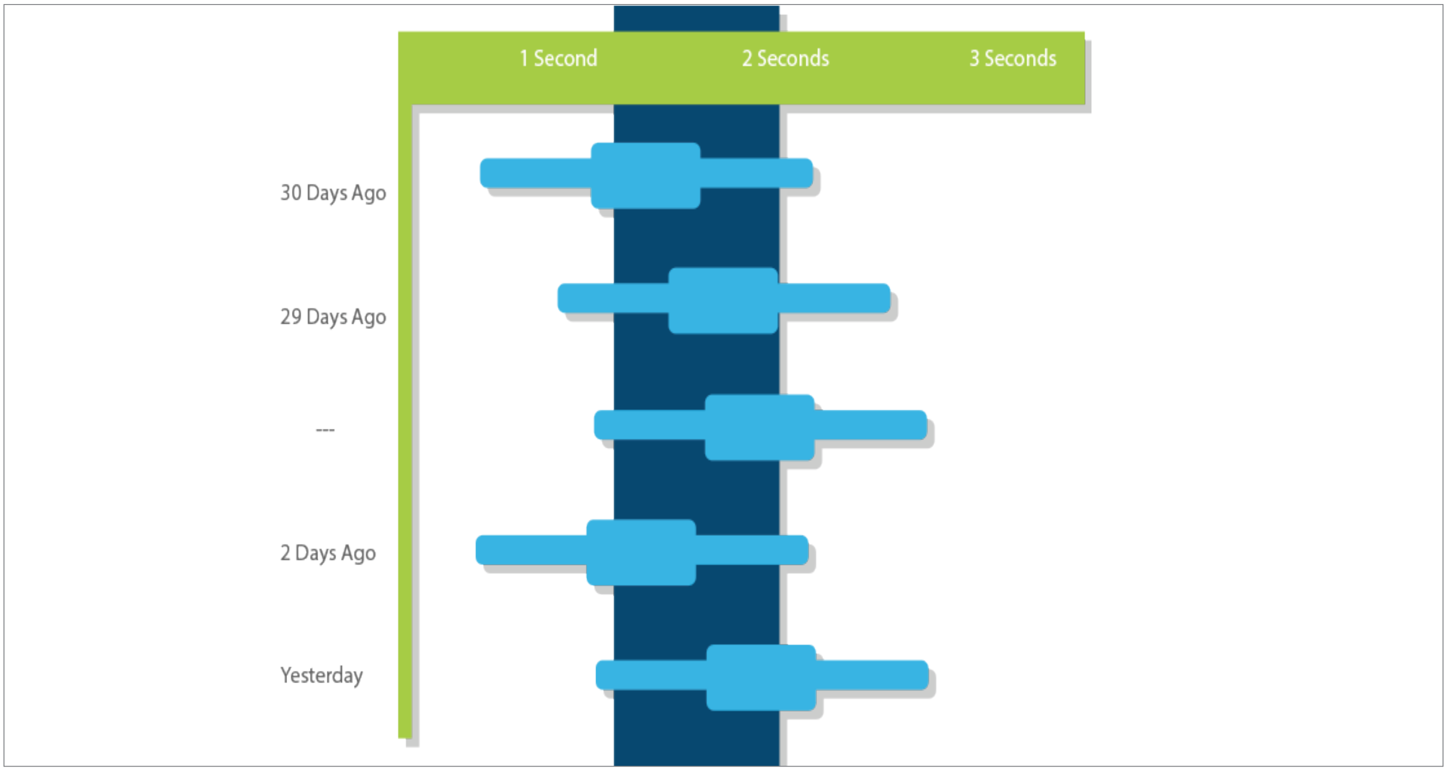

FIG. 4: BASELINING

In figure 4, the average response time for this Business Transaction is about 1.75 seconds, with two standard deviations being between 1.5 seconds and 2 seconds, captured over the past 30 days. All incoming occurrences of this Business Transaction during this hour (9:00am to 10:00am in this example) will be compared to the average of 1.75 seconds, and if the response time exceeds two standard deviations from this normal (2 seconds), then an alert will be raised.

What happens if the behavior of your users differs from day to day or month to month? Your monitoring solution should be configurable enough to handle this. Banking applications probably have spikes in load twice a month when most people get paid, and eCommerce applications are inundated on Black Friday. By baselining the performance of these applications over the year, an APM tool could anticipate this load and expect slower performance during these times. Make sure your APM tool is configurable or intelligent enough that it can understand what's "normal" behavior for your app.

Dynamic Application Mapping

Many monitoring solutions today require manual configuration to instrument and monitor a new server. If new servers are disappearing and appearing all the time, however, this will result in blind spots as you update the tool to reflect the new environment. This will quickly become untenable as your application scales. A cloud-ready monitoring tool must automatically detect and map the application in real time, so you always have an up-to-date idea of what your application looks like. For agent-based monitoring solutions, this can be accomplished by deploying your agent along with your application so that new nodes are automatically instrumented by your APM solution of choice.

Infinite Scalability

Finally, traditional monitoring solutions are designed to manage a relatively small set of servers (up to a couple hundred), not the scale required for a cloud-based application that can grow to tens of thousands of servers. In traditional data centers, each machine is server-grade and the application does not need hundreds of machines at every tier. In a cloud-based application, however, servers are commodity-grade and the number of servers can grow and shrink as load demands. If load grows like Netflix's does then the number of servers can easily top 10,000. Traditional monitoring solutions that instrument every class and method simply can't handle that kind of load – the management server will fall over, or worse, take your application down with it. Monitoring solutions for the cloud must be designed to scale, only collecting the data that's necessary to detect a problem and take action.

Conclusion

The cloud has allowed us to solve new and exciting problems that we couldn't have dreamed of solving 10 years ago, but such a disruptive technology shift requires a new approach in performance management. A performance management solution needs to be highly scalable, resilient to changes in the application topology, and able to understand and baseline the performance of Business Transactions. If your APM solution is capable of addressing all the challenges of cloud-based application monitoring, you'll be ready to take advantage of all the cloud has to offer – without having to worry about compromising the performance or availability of your application.

About AppDynamics

AppDynamics is the next-generation application performance management solution that simplifies the management of complex, business-critical apps. No one can stand slow applications—not IT Ops and Dev teams, not the CIO, and definitely not end users. With AppDynamics, no one has to tolerate slow performing apps ever again.
Visit us at www.appdynamics.com.

Try it for **FREE** at appdynamics.com

AppDynamics

AppDynamics, Inc. 303 Second Street, Suite 450 North Tower, San Francisco, CA 94107 www.appdynamics.com