Ptak Associates

Informed opinion for savvy IT

HYBRID APPLICATION PERFORMANCE TESTING

Managing the performance of today's mobile, web and cloud applications requires a proactive, multi-faceted approach to performance testing.

This paper is sponsored by Apica Systems

HYBRID APPLICATION PERFORMANCE TESTING

INTRODUCTION AND PRIMARY CHALLENGES

Today, IT finds itself at the crossroad of major game-changing technology shifts: the explosive rise of cloud and mobile computing.

The promise of cloud scalability, flexibility and agility is driving enterprises to move applications out of the office and into the cloud. Simultaneously, mobile computing and social media are transforming and increasing interactions between companies and their customers/end users. Organizations are developing new mobile and social media applications that reach out to engage users as digital extensions of their sales, marketing and customer service efforts. These modern, hybrid applications must be flexible enough to perform reliably across a variety of devices and computing environments, and scalable enough to maintain peak performance under heavy loads.

As IT staff move applications to the cloud and mobile, many discover that capacity testing is not the same as it would be testing applications in traditional environments. Applications cannot always scale up, even when running on scalable cloud infrastructures. This becomes painfully clear when applications which performed adequately in traditional environments "break" when subjected to higher loads and dynamic scalability in the cloud.

Hybrid application performance testing must look seamlessly at end-to-end application performance as it travels across diverse environments, and performance must be optimized for each environment along the application delivery chain. This requires advanced, proactive planning and QA testing throughout the development lifecycle. Just as suspension bridge engineers painstakingly test and build strength and resiliency into bridge designs to handle the weight loads of actual use, modern application developers must also meticulously test and build performance-ready hybrid applications.

This paper examines some of the most important considerations for modern application performance testing and planning.

A MULTI-FACETED PERFORMANCE TESTING APPROACH

The unique nature of modern, hybrid applications running across diverse environments demands a different approach for testing performance.

> ### Performance Tip #1
>
> Performance testing should be a regular, iterative process integrated into the application development lifecycle and built into your quality processes.
>
> This is particularly important because of today's accelerated release schedules, which frequently result in inadequate performance testing as developers rush to meet aggressive business deadlines. This "corner cutting" is alarming when considering that inadequate performance can directly impact business and revenue.

Performance testing must use a multi-faceted approach that goes beyond solely measuring response time. This approach is outlined in the sections below.

Understanding Application Characteristics:

The first step to hybrid application performance testing is understanding the performance characteristics of your application.

Knowing how an application performs under various load levels and in each environment (cloud, mobile, traditional data center, etc.) enables managing a hybrid application's performance proactively, in support of business goals.

> ### Performance Tip #2
>
> Establish performance baselines when developing software, by thoroughly testing how it performs under increasing loads and across both desktop and mobile devices.
>
> Performance baseline metrics are useful to:
>
> - Compare the performance of previous and subsequent versions before release.
> - Create new goals and benchmarks for the development team based on business goals. (For example, hit 5 million active users by July 2015.)

The Load Curve

A load curve is the most important measurement of a load test. Understanding how your application performs along the load curve enables operational IT staff to know when increased resources are required to prevent unacceptable performance degradation and poor end-user experiences.
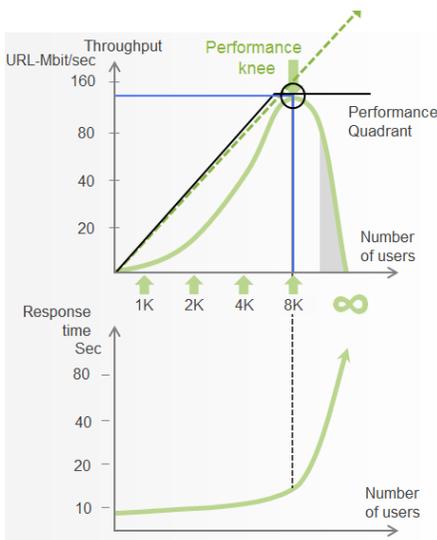
The load curve diagram (graph at the top left) shows application response times versus increasing loads. If throughput doubles at the same rate as the number of users doubles, response times will stay constant. Performance testing helps identify when infrastructure capacity limits begin impacting performance, enabling IT operational staff to proactively avert issues.

The performance "knee" (depicted in the graph at the bottom left) represents the inflection point when throughput reaches its limit and response times begin to spike upward.

Important Performance Metrics

Accurate metrics are necessary to help identify potential bottlenecks and performance issues. Some typical performance metrics to focus on are:

- **Average Response time per web page / scenario**

  Shows how response times increase with increasing loads.
- **Transaction rate – URL /sec**
  Shows the effect of increasing loads on transaction throughput and is useful in identifying when the maximum server/resource capacity is reached.
- **Network rate – Mb/s**
  Shows the effect of increasing loads on network throughput.
- **Error levels**
  Errors could cause performance issues.
- **Infrastructure load – CPU / Server / Session**
  Identifies infrastructure resource bottlenecks or capacity issues.
- **Network and CDN load – CPU Load balances / Session Volumes / CDN metrics**
  Identifies network-related issues, including load balancing, CDN and network loads.

Performance goals should be defined by the metrics above, which provide guidance for the testing process and resulting actions.

Business Goals are a Critical Factor

It's important to note that performance goals are more effective when combined with business goals (such as the number of completed transactions per minute), which are necessary to achieve specific business targets (revenue gains, new customer growth, etc.) Without solid business goals, it is impossible to effectively measure how performance testing and optimization help or hinder the business.

Business metrics typically correlate well with these performance metrics:

- Response time per page or per completed scenario
- Number of page views per minute
- Number of active users

Performance Tip #3

Don't forget to consider the *geographic profiles of load sources*, or the awareness of geographic-related issues that could affect end-user experiences.

For example, your application may run well in the U.S. But if subscribers are increasing rapidly in Japan, are you confident that performance will be reliable for all users, regardless of geographic location?

Business Goals: Example

Example: You are preparing your e-commerce website for the upcoming holiday season.

Your company's business goal is to increase holiday revenue by 25%. The key is determining the most important business metrics necessary to achieve that target (for example, average number of completed transactions/ purchases per hour), as well as the specific performance goals required to reach those goals.

A performance goal, for example, could be a maximum four-second response time on the "Home," "Search for Item," and "Checkout" pages. These performance metrics have a direct influence on your business targets.

Important Considerations when Testing Application Performance

Once you have outlined goals and are ready to start testing, how do you begin? Listed below are suggestions of application performance considerations to include in your testing program:

1. **Consider Specific Performance Characteristics of Each Environment**

   Performance testing modern applications is challenging because of the diverse environments that applications span across (Cloud, Web, mobile, traditional data centers and service providers.) The same performance testing approach does not work equally well for all of these environments, which mean that a unique testing approach should be developed to address each environments' specific performance considerations.

   For example, a web-based application typically has different queries or calls to the server, which is different from what an iOS or Android application would do. A web-based app running in a browser may load images and JavaScripts with GETS and POSTS, as well as HTTP requests. By contrast, a mobile application would typically call Web Services or APIs. Each application-type has unique performance patterns and considerations based on its unique environment and call methods.

2. **Consider the End-User**

   The primary objective for application performance is ensuring the application performs responsively for the end user. Testing should consider the users' browsers and mobile device platforms.

   For instance, modern web applications spend 20 – 50% of load time locally rendering various resources (JavaScript, etc.) that execute after data is transmitted from the server. How is this affecting your end user's experience?

3. **The Performance Baseline**

   First, using a load curve to provide metrics, map a performance baseline for all applications based on individual use-cases. A baseline allows you to identify the impact of new functions and/or infrastructure changes. It also provides the foundation from which you can create automatic alerts for performance degradations.
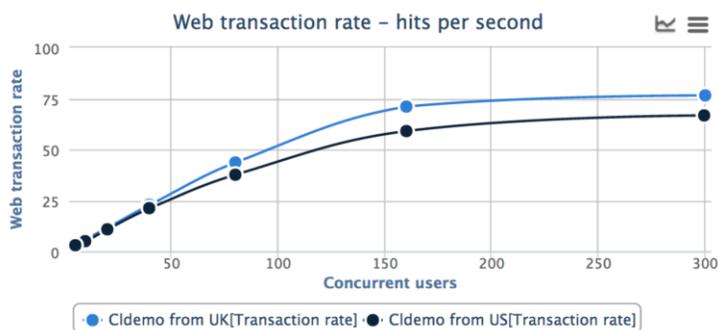
The diagram to the left shows the performance baseline of an application tested from two different locations (note the two load curves of the Web Transaction Rate.) Performance baseline charts are useful for mapping an application's performance at different times throughout

the development lifecycle or from two or more locations.

4. **Know Your Limits: Proactive Cloud Scaling**
    A load test provides the metrics to identify when existing cloud infrastructure will reach its limits, enabling cloud application managers to proactively spin up additional servers *before* performance is degraded. Cloud providers may offer this as a service, called Auto Scaling.

    An Auto Scaling service automatically increases the number of servers in use when user demand increases and removes servers when demand decreases. This maintains reliable performance for variable loads, while limiting costs for unnecessary servers.

5. **Analyze and Test the Application Delivery Chain**
    Managing the performance of hybrid applications is more complex because transactions must travel across multiple environments, with each new environment adding potential performance issues.

    Stress testing a hybrid application assists in identifying potential bottlenecks in the application delivery chain as load levels increase. When heavier transaction loads (stress) are placed on the cloud service, weak links and components either break or begin showing slowdowns. Identifying these performance vulnerabilities and bottlenecks enables IT staff to remediate weak links preemptively before they affect real customer transactions.

> ### The Rise of Mobile
>
> The rise of Mobile creates a plethora of additional performance considerations:
>
> First, there are a wide variety of devices available (mobile phones, tablets, phablets, PCs, etc.*)*
>
> Then each device type also has a variety of models (Apple, Samsung, HP, Dell, etc.) and OS platforms (Windows, MAC OS, iOS, Android, etc.) from which to choose.
>
> Then add in the variety of browsers (Firefox, IE, Chrome, etc.) and/or device-installed applications the end-user may use. The resulting combination of user-chosen options can impact performance.

6. **Know The Impact of Adding New Servers**
    Understand the real impact of adding new servers. Doubling server capacity does not necessarily mean the application can handle 2x the number of users, especially when those users may be accessing the application from various geographic locations and devices. Base your cloud scaling decisions on real data, not assumptions.

7. **Failover Procedures**
    Make sure to test redundancy (i.e. failover procedures) during peak loads and in multi-tenancy environments. This enables staff to "work out the kinks" before you really need it.

8. **Mobile Apps – Analyze API Usage**

Mobile applications that are using a web service API need to be tested using the real access to the API, *not* just the API itself. In other words, it is critical to analyze how a mobile app uses a particular API before testing, in order to establish a baseline understanding of Mobile performance.

Differences such as operating system, device type, etc. can change how the API is being used. In many cases, performance issues may stem from the mobile app not effectively using the API. Proper usage analysis (followed by performance testing) is the only way to determine this.

9. **Internal APM Systems**
   Using internal Application Performance Management (APM) systems to monitor application transaction data is an effective way to track problems in the application's code.

Integrating load testing with APM is the best way to proactively find, fix and validate that code problems are fixed. *Also note that* Integration must be at the http level, so an individual POST can be tracked from the test tool all the way to the APM solution.

> ### Customer: Virgin America
>
> "We recently rebuilt our website… High capacity planning, stress testing our booking purchase flow, and overall performance optimization were critical before the big launch. Apica's team worked hard to ensure our site lived up to its "faster, smoother, and more intuitive" promise. They handled all aspects of testing from start to finish, both helping us pinpoint our greatest bottlenecks and translating test metrics into actionable insight for our development team. Their advice saved us countless hours troubleshooting and helped us avoid any post-launch performance snafus."
>
> Tony Vela
> IT Manager, Virgin America

The Final Word

Managing the performance of today's mobile, web, and cloud applications requires a proactive, multi-faceted approach to performance testing. As applications span across disparate environments, each with their own unique performance issues and characteristics, it is important to know what to test and how to test effectively to ensure optimal performance, especially when load demands increase unexpectedly.

The key to success lies in basing performance decisions on actual data and insights, not guesswork. Understanding the performance characteristics of each type of application environment is crucial, as well as understanding what happens when an application reaches its capacity so outage avoidance measures can be taken preemptively before they impact the business. Modern application performance testing is necessary to provide performance insight based on real data, that can be used to minimize the risk of unpredictable, ill-informed and costly behavior in live application environments.

Determining the best application performance management approach can be confusing because of the sheer number of vendors offering silo-based mobile, web and cloud testing solutions. However, it is important to note that the most critical component of application performance management is not the amount of data produced, but understanding how to use that data to find and fix performance problems while optimizing applications for high scalability, performance and agility.

Ptak Associates

Informed opinion for savvy IT

Apica's approach to application performance focuses on solving customer performance concerns with a combination of superior performance testing data and "strike team" expertise. Apica's Professional Services team uses their best practices, methodologies and know-how to help customers interpret and use performance metrics to find problems, fine-tune and optimize applications.

Companies worldwide use Apica's solutions for actionable end-to-end application performance testing for mobile, browsers, Web, cloud, and traditional data centers. Apica's SaaS performance testing solutions can also be integrated with DevOps teams.

Apica's unique combination of **Synthetic Web Performance Monitoring (WPM) and LoadTesting** gives a complete view of application performance, both on the server and client side. This is especially effective for modern JavaScript and Mobile applications.

Though customers have the option to "do-it-themselves" with Apica's self-service testing platform, one of Apica's key differentiators is the expertise they provide working alongside their customers to diagnose and solve performance concerns.

As one customer stated, "*Apica provided us with automated testing that, when accompanied by their human touch and personal analysis of the data, allowed us to troubleshoot all of our bottlenecks.*"

Learn more at www.apicasystem.com.

Sponsor Information provided by Apica

Publication Date: November 6, 2014

Author: Audrey Rasmussen, Partner and Principal Analyst
**   Ptak Associates LLC**

About Ptak Associates LLC
Our analysts cover a breadth of areas that are ideal to bring you the "Big picture" on new technology trends across the industry. Whether it's Cloud computing, Mobile (BYOD), the Internet of Things, DevOps, Big Data, IT Operational Analytics, Workload Optimized systems or other new trends, Ptak Associates analysts cover these trends with a unique perspective that is both deep and broad.

Our clients include both industry leaders and dynamic newcomers. We help IT organizations understand and prioritize their needs within the context of present and near-future IT trends, enabling them to use IT technology effectively in solving business problems. We help technology vendors refine their strategies, and provide them with both market insight and deliverables that communicate the business values of their products and services. We provide all clients with an understanding of how their competitors are playing in their market space, and deliver actionable recommendations.

www.ptakassociates.com