

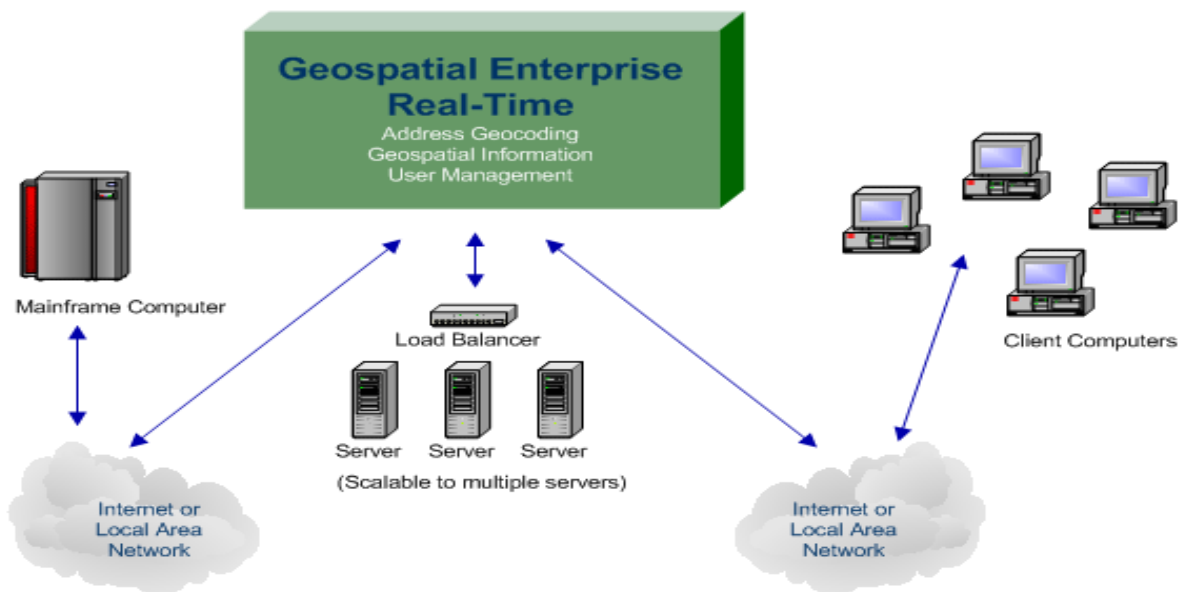
# Performance Management for Web 2.0 Solutions

## Executive Summary

Before we proceed to discuss Performance Management in the world of Web 2.0 solutions, let us briefly examine where the core performance management disciplines evolved from. By doing so we can see that the fundamentals of Performance Management have not changed but in fact are required more than ever.

I do not want to highlight my age, neither promote myself as a mainframe bigot. But the mainframe did establish many mature software engineering disciplines, such as release, change, configuration, environment, problem and performance management. All of these disciplines fed into mature processes such as the Unified Process, which are still used today for the development of new software platforms.

When talking specifically about performance management, in the days when mainframes were the sole core delivery source of customer data, there was a strong performance management discipline. This discipline provided assurance and risk mitigation to the owners of these systems that the customers would not receive any negative experiences when interacting with these systems such as bad response times or system outages. Using the IBM MVS mainframe as an example, there were such tools as OMEGAMON for monitoring queues, cpu, disk and memory, RMFMON and SMF for performance sampling response times, SAS for statistical analysis and TPNS for load simulation. Proactive processes were in place to ensure that the IT departments were aware of any potential performance impacts prior to the customer finding them. Large financial institutions were the major supporters of performance management in this area as they understood system outages could lead to large monetary losses. Today, mainframes continue to be supported by a strict performance management regime.



As the move was made into developing client/server solutions, and progressed to web based solutions, these disciplines, specifically Performance Management, seemed to become temporarily forgotten or maybe just received less priority. This was short lived after a large number of massive failures and subsequent monetary losses were encountered as a result of web site failures caused by halts and excessive response times. Soon the once forgotten discipline of performance testing was once again included

as a core phase of the software development life cycle. Once again test managers, project managers and even senior executive management members were talking about performance testing activities, results, success or failure of KPIs or Service Level Agreements (SLAs) prior to deployment. However, a lot of companies, primarily due to costs involved in performance testing activities, still omitted these activities and suffered the consequences. Although, even those companies who performance tested their applications prior to deployment suffered later as they neglected to encompass the whole performance management package, that used to protect the mainframe from system failures. For performance testing, although a major contributor to the Performance Management solution, is only part of an overall holistic performance management solution.

As there is more hype and talk about Web 2.0 solutions, there is even more necessity for IT departments to deliver a complete Performance Management package. The end user experience is now more complicated to gauge using the old Load Testing 1.0 philosophy of neglecting those components past an external firewall, even the client performance itself had been previously ignored. We will explain in the next section why performance risk areas in Web 2.0 solutions are more prevalent .

This whitepaper will explain to the reader what is required to deliver a complete holistic approach when mitigating performance risks. It will explain what should be incorporated in a Performance Management package and also highlight why companies should ensure they encompass the complete set of Performance Management disciplines, rather than a small subset of them when preparing to embark on any new software development project.

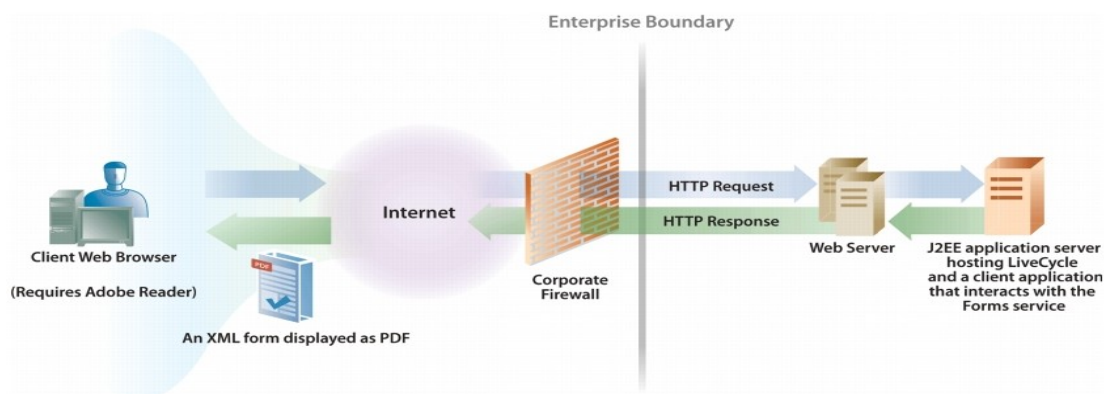
## **Web 2.0 Performance Risk Areas**

Before we get into the detail of what performance management is lets just stop here for a minute and discuss at a high level some performance risk areas within Web 2.0 solutions.

Please refer to the many whitepapers on the internet that will provide a more in depth explanation of what is meant by Web 2.0 solutions. For our purpose here, we will provide a brief description of what is meant by it, highlighting why this has more performance impacts if not properly managed.

## **Client Rendering**

Web 2.0 solutions now deliver more functionally rich capabilities to the web browser. Technologies such as Flex from Macromedia, as an example, take the processing responsibilities away from the server and place them on the client.



Ajax controls and more client powered java scripts, empower the client browser, but now

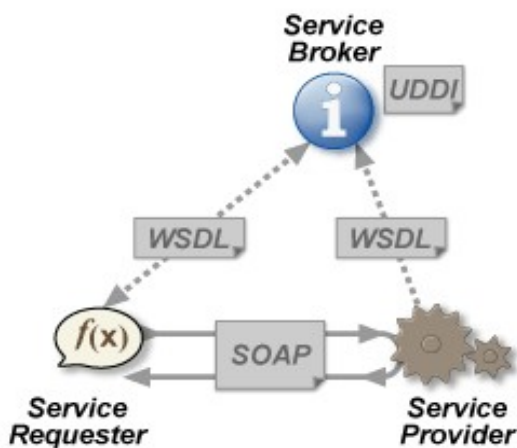
place more importance on the performance measurement of what we call client rendering. Client rendering is the process that the web browser takes to present the complete refreshed response to the end user.

Client rendering could just be the populating of an Ajax drop down list control, or it could be the time take to completely draw that all important Sales graph. Performance testing for Web 1.0 solutions has normally focussed on server performance, although this is still important we need to ensure that we now measure response times on the client as well. A complete end to end user response needs to be measured when the applications servers are not under load, and then when they are under load, to determine whether or not the the response time is acceptable.

## Web Services

More Web 2.0 solutions are utilising the flexibility of using external web services as defined by the World Wide Web Consortium (W3C).

There are many obvious benefits in utilising external web services, specifically in saving development time and costs.



However, your application now has a dependency on performance of an external site. How do you ensure that this performance is adequate?

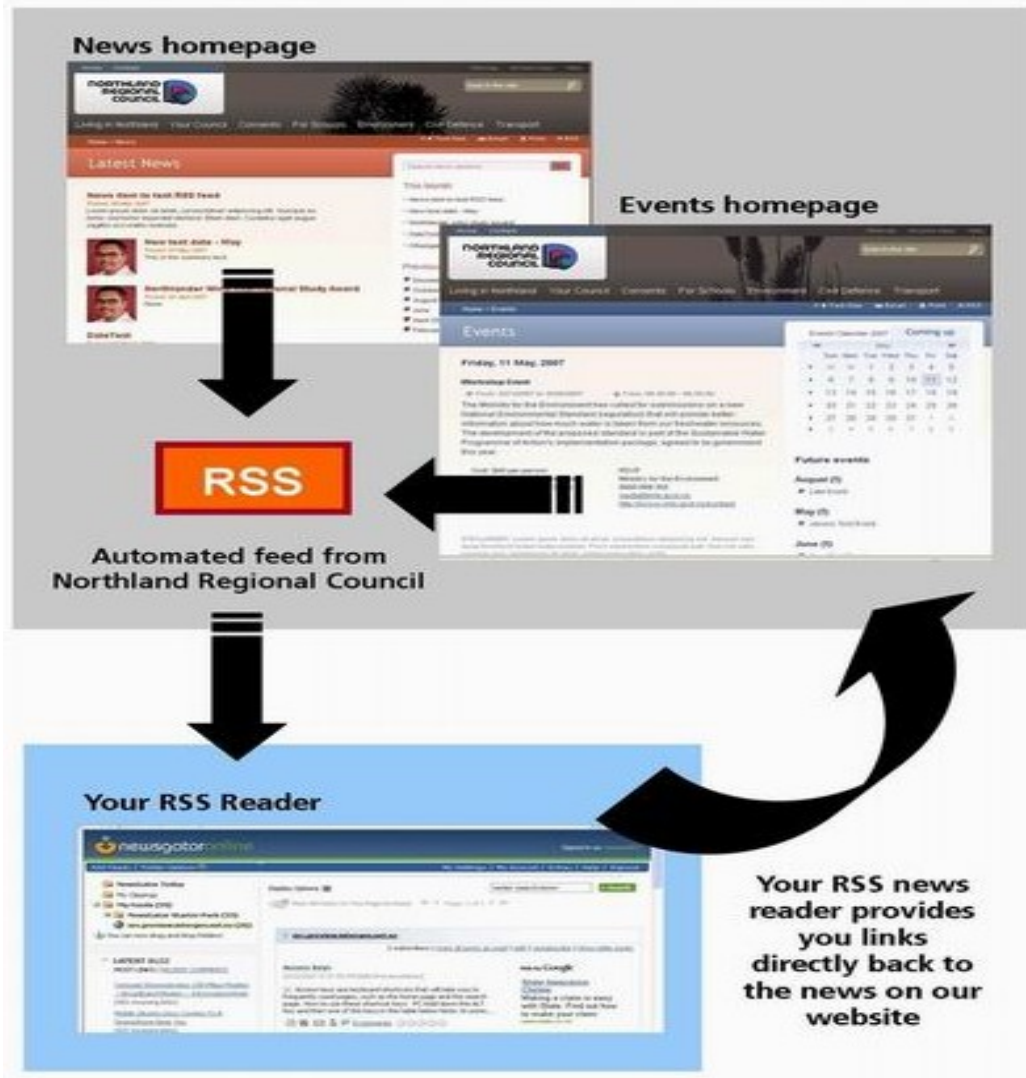
## Geographical Locations

As we know, the benefit of the internet is that users of web site applications can be geographically dispersed. However the downside of geographically dispersed users is determining how to measure the response times of a new web site under test, when so many variables are involved.

With the development of Web 2.0 solutions, the client becomes more functionally rich, and more resource intensive. The complete end to end user experience for every user, no matter their geographical location, now has to be managed by a performance management solution.

## RSS Feeds

There has been a rapid increase in the amount of web sites using RSS (Rich Site Summary) feeds. Basically the RSS feed provides a format to deliver regularly changing content to your website such as news, weather and stock prices as an example.



Again, we now have external dependencies on performance. Also the feeds into the web site will have an unknown performance impact to the other processes running on the clients computer, which will impact the users experience with the web site.

These are some areas with Web 2.0 solutions that magnify the necessity to performance manage your web site. Together with the existing issues surrounding web 1.0 solutions, there is now more than ever a need to encompass the performance management regime.

### ***What do we mean by Performance Management?***

Now we have highlighted some of the issues associated with the deployment of new web site applications specifically Web 2.0 solutions, lets now elaborate on performance management.

When it comes to discussing performance most IT practitioners are now aware of what we refer to as performance testing, although a lot still do not understand how this is done in a properly focussed approach, but that is another white paper in itself. However, as we

mentioned previously, performance testing is one aspect of performance management.

A complete performance management solution, will performance test in a controlled approach, in an appropriate environment, before live deployment. It will then continue to performance regression test subsequent releases, to ensure no performance degradation has occurred between releases. A capacity planning discipline should be adopted for the live site, where statistics and models are reviewed on a regular basis. The user end experience should be proactively monitored using an appropriate dashboard. Lastly all server metrics should be monitored and fed into a viewable dashboard realtime.

That in a nut shell is what we mean when we talk about Performance Management. Now lets discuss some of these aspects in further detail.

## **Performance Testing**

This deserves a white paper of its own.

The most important ingredients into successfully performance testing is focussing on the high volume and business critical functionality of the web site under test. With the emphasis on end user response times, it is now necessary to not only performance test the appropriate servers, but to include the client rendering response time measurement in overall end to end response time measurement.

By utilising some open source alternatives to reduce costs, such as JMeter or OpenSTA, web traffic can be successfully emulated in varying loads on the web, file, application and database servers, by mimicking the browser incoming traffic. Whilst the servers are being loaded in this manner, geographically dispersed real browsers can emulate real user response times from varying locations. This can be achieved via cloud computing and such tools as BrowserMob.

For the test environment and load generators, the Amazon Elastic Cloud Computing (EC2) facilities can be leveraged to again massively reduce the associated costs.

Throughout the various performance tests, server metrics and user end response times should be collected and examined. Refer to the monitoring section discussed further on.

To ensure no performance degradation is encountered between code releases or any other changes, rapid performance tests should be conducted prior to any changes being deployed to production. Technical impact assessments can be conducted to determine what specific performance tests should be executed.

Sometimes it is forgotten that only one bad line of code, or one incorrect value of a configuration file can cause serious application performance problems, and can be a pest to find in production.

## **Capacity Planning**

A proper capacity planning discipline should be adopted for any new website. This can be as simple as a projection model which can determine when links need upgrading, or new hard disk or memory will be required.

A projection model can be prepared based on ballpark projection figures, either sourced from historical data or from business projections. By regularly reviewing real statistics from the usage of the site by examining web logs and other sources, the model can be updated to ensure more realistic projections. Figures from the server monitoring solution, discussed later can feed into the model as well.

Capacity Planning is a logical exercise that when done correctly can save your web site a

lot of downtime.

## Proactive Performance Monitoring

The performance monitoring solution is the glue that holds the performance management solution together. It is vital that you proactively monitor the performance of both your production and test environments.



By proactively monitoring performance we mean consistently collecting server metrics and user end response times from all your environments, and reviewing these on a regular basis. Better still, setting up various thresholds and automatically receiving alerts once these thresholds are breached. This approach is in contrast to the passive monitoring approach, where performance metrics are only examined once a problem has been reported by an end user.

The key to performance monitoring is not only monitoring all the vital server metrics, but also monitoring the critical business processes from a users perspective. These can be done by automating these processes and running them on a regular basis.

## Conclusion

By adopting the three key performance management areas of performance testing, capacity planning and proactive performance monitoring, you will be well on the way to guarding your website from costly outages. You can also safely support and encourage any business campaign or promotion that leads to increased traffic to your web site.

## About the author

Nick Singh is an Australian based Performance Testing practitioner. He is a founder of jkvine.com, a company that prides itself on focussed and cost effective performance management solutions. Nick has 28 years experience in IT of which 23 years have been dedicated to his passion for performance testing.

## ***About JKVine***

JKVine is a company based in Melbourne, Australia that specialises in performance management solutions. It prides itself on expertise in performance management and has expert knowledge in all toolsets both commercial and open source. This allows the company not only to provide quality performance management solutions, but cost effective solutions as well.

For further information on how JKVine can help your company, please contact JKVine

[www.jkvine.com](http://www.jkvine.com)