



## WHITE PAPER

# Best Practices in Enterprise Application Performance Management (by Apsera Tech Inc\*)

*Sampath Prakash, Ph.D.*  
*siprakash@apseratech.com*

### Abstract

It's a summer Friday afternoon and you are just a few clicks away from completing that inescapable weekly corporate time-sheet application on your PC and head towards the beach. You may even want to squeeze in that much delayed expense voucher, if you have few more minutes left. You are barely through two clicks and the dreaded hourglass makes its appearance. The scenario although not very uncommon, is perhaps not serious enough to warrant filing a trouble-ticket (and besides, you do not want to start that slow Remedy application and risk getting stuck in the rush hour traffic). On the other hand if you are a call center agent and it is taking a minute to pull that customer record, you are in bad luck.

Application performance problems are just like common colds- hard to understand why they occur and difficult to find easy cures. Well, common colds at least go away in a few days but application performance issues seem to linger for ever.

I have tried to narrate in simple terms, what are the causes of slow application performance in your enterprise network and what you can do to alleviate the situation by adopting some best practices. If you are the Network Infrastructure Manager, Business Unit Head owning the applications and servers, or the CIO, I believe you can gain some insights on this topic of enterprise application performance management which is becoming as serious an issue as IT security.

If you are an end-user like that guy headed to the beach or that customer who is just trying to make changes to her return flight, you may not be able to do much- but at least learn to empathize with the complexity of issues if you have a chance to read this white paper.

---

## **What causes slow application performance?**

(In this white paper, application performance means response time of a transaction from an end-user perspective. A transaction is defined as a user action for which there is a specific identifiable response from the system, such as a query and an answer- response time signifies the user's perceived wait time. Response time concept could also be extended to a machine-to-machine interaction.)

We are not talking here about that spreadsheet application on your PC or a file-server application within your LAN; we are talking about interactive type of applications where your application and database servers are sitting in a data center or a server complex at the other end of the WAN, a few or several thousand miles away or even just a few hundred miles away from you.

To start with, your PC itself might be slow because you have a low performance CPU or you are low on your disk space or you have opened too many applications and have only 128 Kbytes of RAM or you have a virus or other possible local reasons. The client portion of the application running on your PC may be inefficient and might be slowing you down. You should definitely eliminate your PC or laptop as the source of your problem and you may be even lucky to get a replacement machine if you complain to your manager every other day.

Or similar problems may be happening in your application server or database server. If it is capacity related- CPU or RAM or disk utilization, there are many server management tools such as BMC Patrol, CA Unicenter that can alert you proactively. Hopefully there is some initiative to replace that aging NT server- which probably will take a lot more time than replacing your aging PC or laptop. But the present situation or the remedy is still not too bad.

On the other hand if the application itself is architected poorly resulting in inefficiency- for example, if the database queries are not optimized between the application server and the database server (even when they are co-located), it is a harder problem to fix. It reflects poorly on the architect, designer and the developer of the application. Fortunately there are quality certifications to be earned by development and testing organizations and hopefully a poorly written application is not the real reason for the sluggish application performance.

There is an important factor even experienced application development organizations often fail to take into account during creation or testing of applications- that is, to assume that the user, the application server and the database server are all together in the same LAN and ignore the latencies for data flow among the machines. Enough attention is not paid to keep the "chattiness" between the client and the server to a minimum.

It takes 10 milliseconds for light to travel 3000 kilometers straight and in practice it takes even longer to travel through a fiber-optic link for the same distance. A 10 millisecond one-way latency between a user and the application server may seem harmless, but we have seen applications that are very chatty and may take 2000 round-trips (or "turns" as it is called)

between the user PC and the application server. This will easily add a wait-time of 20 seconds!

The combination of application chattiness and the latency between the client and the application server can be a major cause of sluggish application performance.

There is an additional delay effect which often comes into play in the application performance because of the latency between a client and its application server. Most data applications use TCP protocol as the underlying mechanism for reliable data transfers. TCP sends messages in chunks (determined by the "window" size which is typically 8 Kbytes or 17 Kbytes depending on the operating system and not more than 64 Kbytes) and has to wait for an acknowledgement from the client PC before sending the next chunk of data. On inter-continental long haul networks (where delays are in hundreds of milliseconds, rather than tens of milliseconds), you may end up in a situation of having a large size pipe (such as 45 Mbs or higher), but not being able to quite "fill up" the pipe. Even if you attempt to increase the window size, it may not be feasible to increase beyond a certain size because of other factors such as bit-errors on the communications links.

The next factor contributing to the poor application performance is the well-known bandwidth, in particular, the lowest bandwidth or the "bottleneck" bandwidth in the path between the client and the application server- typically the access bandwidth to a remote office. Fortunately there is great awareness of this issue among consumers and business users- thanks to the explosion of Internet and e-commerce applications among consumers, and the ISPs for touting the phenomenal progress from dial-up to high-speed cable-modem/DSL internet access.

As the amount of data that needs to be transferred becomes larger and larger in any particular application transaction, there is a clear performance benefit in providing more and more band-width- well, till some point where the TCP protocol factors mentioned earlier come into play. In many situations such as the ones involving large amounts of data, bandwidth is the major cause for slow application performance. However, the common tendency to throw in more bandwidth for all application performance issues needs to be clearly avoided.

There are a couple of other important factors that affect end-user performance. The network may be congested and the packets may encounter queuing delays (or even get dropped) at the switches, routers, firewalls, and other devices- which can have a compound effect as you make many round-trips before completing a user initiated transaction. Packets that are lost or discarded will have to be re-transmitted, adding to the delays.

If the quality of transmission links is not high (for example if there is a wireless LAN in the path of transmission) there can be packet errors. A single errored or lost packet will result in the retransmission of the whole TCP message. Sustained packet errors and losses will lead to a downward adjustment of TCP window sizes.

It is clear from the above discussion that the cause of slow application performance is due to the combined effect of so many factors and is not always easy to pinpoint. There are very sophisticated tools in the market place which can help isolate the performance problems to a large extent. But these tools can be expensive and often need assistance of performance experts to use and interpret results correctly.

## **What else is happening in real life that compounds the issue of application performance?**

It may be unfair to say that Network Infrastructure Manager (NIM) and Business Unit Heads (BUH) owning applications are totally unaware of the above causes of poor application performance. Being aware is one thing, being able to do something about it is another. You may have heard of so many large budget IT initiatives or Network initiatives in any company, but do you recall any project whose explicit goal was to “*improve performance of existing applications to end-users*”. An application might have performed wonderfully when it was initially introduced and in a matter of months might degrade in performance. Data centers and Network Infrastructure are living entities hit with frequent changes. I tried to list some key reasons below, but there may be many others:

- ◆ *New data applications* are added regularly into the network- not always in a planned manner. This causes competition for existing resources such as bandwidth and server CPU cycles. Also you may have some unknown applications (that have escaped the normal process) springing up in the network
- ◆ *Voice/video/multimedia/streaming applications* added- Voice over IP traffic does not consume lots of bandwidth, but video and multimedia applications can consume significant chunks of the existing bandwidths. Voice and mission critical applications are given priority in the network transit by implementing QoS. The NIM has to be able to arbitrate conflicting requests from many application owners to give higher priority to their applications and smoothly manage the QoS implementation.
- ◆ *New WAN architectures/technologies* implemented to take advantage of added features and better price points. From a leased line to a Frame Relay network to an IP-enable-Frame Relay network to an MPLS network. Many small companies use site-to-site VPNs based on public Internet. The new networking technologies introduce additional latencies among sites and do not, in general, help application performance. You may discover that your latency to the server site has gone up from 10 milliseconds to 30 milliseconds which may be enough to cause noticeable slowness in application performance.
- ◆ *After 9/11 and natural disasters* like Katrina, many US corporations are consolidating and/or relocating data center. Data center consolidations are also undertaken for various business reasons including cost reductions. Whenever a data center is relocated the local users within the data centers are worst hit in application performance as they move from a LAN like environment to a WAN environment. Those users are stuck with increase in latencies and decrease in bandwidth. Depending on the relative location to the new and old data centers, other sites may also suffer performance degradation.
- ◆ *Business related events such as Mergers and Acquisitions* These events can trigger initiatives such as consolidation of networks, relocation of users, streamlining of applications, and consolidation/relocation of data centers. If not planned well, all these activities can impact performance in a negative manner.

- ◆ *Security related impacts-* Worms and viruses can affect network and server infrastructures and bring things to a grinding halt. But these are temporary events that hopefully will pass quickly.
- ◆ *Limited Budget-* Although bandwidth is getting cheaper and the hardware for servers is also getting cheaper, budget constrains may prevent addition of new “capacities”. It is quite possible that the right Network/Server Management tools may not be in place to add capacities to the network infrastructure in a timely manner. Even if the right tools are deployed, there may be a lack of right people or processes to perform an effective capacity management
- ◆ *Inability port old applications to new OS/servers-* Although there may be available budget to buy a new server hardware to port an old legacy application to new hardware, thereby improving performance, the application itself may not port to the new hardware or operating system.
- ◆ *Lack of communication-* The complexity of application performance implies that there has to be clear and frequent communication (supported by well defined processes) between the NIM and the BUHs owning applications. The lack of communication is evident from finger-pointing episodes which happen quite frequently in an IT organization.

### **What are the best-practices that an enterprise can use to ensure good application performance?**

Application performance has to become as much an important issue as, for example, network or application security. There should be at least a part time position such as “Chief Performance Officer (CPO)” reporting to the CIO. CPO will have the responsibility of ensuring that applications are built according to best practices and also ensuring that there is effective communication among business unit application owners and network/server infrastructure managers. CPO should have direct access to the end-user performance “dashboard” and he or she can be the enforcer of application performance SLAs (Service Level Agreements).

An overall best practice that can be implemented by a CPO is an approach called “*Event Risk Management*”. Any application performance impacting activity (among the many mentioned earlier) is considered an “Event” and a risk analysis is conducted (note that new application introduction itself is considered an event). If there are risks to performance, mitigation steps are explored and implemented. We convinced one major company in the financial industry to adopt this best practice.

Under this umbrella of “Event Risk Management” approach, the following best practices are applicable to the BUH owning the application and the application development organization:

- ◆ **In line with any best practice quality principles, application performance issues are best taken care at the drawing board- during planning, development, testing, and pre-deployment stages.**

- ◆ The application architect has to understand the parameters and causes of poor application performance and there needs to be a general awareness of WAN impact in the development organization.
- ◆ It is very important to visualize to the extent possible how the application is deployed in the field in a WAN environment. It is equally important to visualize how the applications will be used by the end-users. This will force one to better understand the pattern of usage. It will help to realize which transactions are used more frequently, which ones are used moderately and which ones are seldom used- this will help in the next step.
- ◆ Set realistic performance objectives i.e., response time requirements for end-users (or even between two machines as needed) for all the transactions. It may not be necessary to have one-second response time for a transaction that is expected to be used once in a week.
- ◆ Incorporate performance requirements as an integral part of the test plans
- ◆ During the application design, understand which tiers of the application will surely be within a LAN and which tiers get separated by the WAN. Continue as before to perform the optimization for the components within the LAN (flow between the application server and the database server)
- ◆ The part of the application design where components have to communicate over the WAN has to incorporate the expected WAN characteristics (realistic bandwidth and latency) – the application architect has to enforce this discipline onto the application developers and coders.
- ◆ Once the unit testing and integration is completed for testing the logic, perform stress testing (with the usual tools such as Load Runner or Empirix) and choose appropriate servers. At this point, testing is still within a LAN environment
- ◆ Once a stable application code is available, the application has to be tested in a WAN environment over which it is planned to run to determine if the response time objectives are met. This testing can be done either by using WAN emulators or by employing modeling/simulation techniques. To ensure more realistic results, the equivalent servers which are planned to be used in the actual field need to be used during testing or modeling/simulation. The WAN model used for testing should reflect conditions similar to the actual network in terms of bandwidth, utilization of circuits, WAN technology, and packet loss. This stage of testing is usually referred to as WAN Application Readiness Testing (WANART).

The results of the WANART stage can be one of the following:

1. The response time objectives for the application are met in the WAN environment that the application is supposed to operate. No bandwidth upgrades are needed at the branches or in the data center for the projected number of users/usage pattern. The results should be communicated to the NIM (via CPO) for his or her ongoing capacity planning purposes. The application is ready for pilot deployment

2. The response time objectives for the application can not be met in the current WAN environment, but can be met with bandwidth upgrades at the branches and/or in the data center for the projected number of users. In this case the CPO has to initiate a business case with participation from the business unit application owner and the NIM and seek approval from CIO, if needed. After the needed bandwidth upgrades are completed the application can be ready for pilot deployment. If it is determined that the application may benefit from QoS, this is a good point to be in negotiations with the NIM.
3. The response time objectives for the application can not be met in the current WAN environment- even if significant bandwidths are added. Perhaps the application is too chatty to handle the expected latencies. It might be best to redesign/recode the application to reduce the chattiness and go through the WANART step again. If that is not possible, alternative deployment plans with servers close to the clients can be considered. If none of these are possible and the application has to be deployed as such, emerging alternatives such as application accelerators (network based such as NetLi or premises-based such as the ones from Juniper, Cisco, Riverbed or Citrix) may be considered.
  - ◆ Benchmark performance during pilot deployment and fine-tune the bandwidth and QoS requirements with the NIM. Sign a mutual SLA with him or her.
  - ◆ Deploy the application!!! Let the NIM do his or her proactive capacity planning. Keep in constant touch with him or her to communicate any change in requirements on an ongoing basis.

### ***Best Practices for Post-deployment***

Best practices during pre-deployment, if observed, will definitely put you ahead of the pack and your end-users/customers will be satisfied for quite some time. But maintaining the response time SLAs is more of a marathon than a sprint and there are best practices to be followed by the application owner as well as the NIM. I have listed the few important ones:

*For the Business Unit application owner:*

- ◆ Know who the actual users of the application are and where they are located- maintain a spreadsheet database that is current.
- ◆ Track user growth across locations and volume growth at each location
- ◆ Have the response times for your application transactions measured once in a quarter at least and ascertain your objectives/SLAs are being met. What is not measured cannot be managed.
- ◆ Be sure to keep the NM in the loop and engage in proactive communication

The CPO can take the overall responsibility for application performance. Some steps he or she can take are:

- ◆ Compile a list of critical business applications. The Business Continuity and Disaster Recovery (BCDR) document is a good place to start as it will contain a unanimous list of the most critical applications for the enterprise (and do not say that you have not updated your BCDR document!). If there is budget/time constraint, which is usually

the case, a prioritized subset of these applications can be chosen to monitor the performance.

- ◆ Conduct a user survey to understand the general performance of the chosen critical applications. The survey can be qualitative- users rate the response time of the applications they use as Good, Satisfactory, Slow, and Very Slow. They can also indicate if a customer wait is involved if the application is slow.
- ◆ For those applications listed slow or very slow, detailed response time measurements should be undertaken (if the business unit application owner has not already done so).
- ◆ Communicate the results to the business unit head and the NM.

*For the NIM:*

- ◆ Be in constant communication with the Business Unit application owners- remember that they are your customers.
- ◆ Understand that whenever new applications are introduced/allowed it will impact the performance of current applications.
- ◆ Enforce the process for introduction of new applications/servers and PCs/desktops.
- ◆ Ensure that change control process is followed for normal maintenance routines and special situations.
- ◆ For Infrastructure applications such as Exchange/Outlook, Lotus Notes, the same best practices as those for a business unit application needs to be followed.

Many enterprises have been following the following best practices by implementing various suites of tools such as Concord, Netscout, Network Physics, and other tools. But it may be worthwhile to repeat them here:

- ◆ Have the right tools/people/process for monitoring traffic utilization, usage trends, and fair bandwidth usage by various applications. Keep utilization low at the branches and the data centers- focus on 1-minute utilization during the busy hour rather than just the hourly utilizations.
- ◆ Know what are the “other” applications consuming bandwidth in the network. Identify and eliminate unwanted traffic quickly.
- ◆ Keep switches, routers, firewalls, load balancers in good health (CPU utilization, buffer overflows, etc.).
- ◆ Have a sound QoS process/policy and Implement QoS correctly. Lower priority should be given to less critical application traffic.
- ◆ Monitor packet loss and re-transmission rate of packets.
- ◆ Monitor ping delays/network round trips among location pairs.
- ◆ Last but not the least, review the server performance summary reports and ensure that servers are operating in healthy zones.

## **Summary and Some Final Thoughts**

Performance has to be an integral part of the application life-cycle- planning, requirements, development, testing, deployment, and maintenance/upgrade. Unlike security issues, which tend to be acute and are easily noticed, performance issues are chronic in nature and can cause slow erosion in productivity and customer and end-user satisfaction.

However, performance management across the board (to include all possible applications in the enterprise) can be an expensive proposition. But well targeted performance management practices need not be expensive and is the right way to start. Performance objectives have to be tied to business needs- Return on Investment (ROI) for having a fast application in terms of lost revenue, productivity, and customer satisfaction has to be looked at. Here are some final parting thoughts:

- ◆ Follow best practices during the application development and deployment phases and avoid/prevent problems. It costs dollars to measure application performance in a production environment in response to user complaints and it can cost a lot more to fix performance problems.
- ◆ Take an *Event Risk Management* approach and follow well-defined processes to foster effective communication among application owners in the business units and infrastructure managers.

If you did not quite follow best practices and if you are lucky, there can be tactical solutions:

- ◆ Some times there can be easy fixes such as bandwidth increase, QoS enablement, moving servers closer to users, increasing server capacity, and adjusting TCP window sizes.
- ◆ There are somewhat more difficult fixes such as using application accelerators (from Riverbed, Juniper, Cisco, Citrix, and others) and application accelerator services (like those from NetLi). These may work for specific applications or for a category of applications such as Web/browser-based applications.
- ◆ Lastly, I would like to point out that web/browser-based applications have certain advantages over general client/server applications in the area of performance management. Many measurement tools are available for baselining the performance. Acceleration services/devices are more readily available for web/browser-based applications.

### **Acknowledgements**

I would like to express my sincere thanks to John Sikora from AT&T (Network Integration and Consulting Division) for his unwavering commitment to the area of application performance and for patiently educating me through various projects we worked together in the last several years. Thanks to Kwasi Yeboah-Afihene for patiently working with me in helping many customers with their application performance issues.