Tackling Tough Performance Problems: Part II

The summer's newsletter is the second of two parts on tackling performance problems. Part 1 highlighted some of the performance problems that arise in businesses and the impact that business and management bottlenecks have on IT. Strategies for working on these problems were reviewed. Part two covers technical performance concerns such as the theory of constraints, performance testing strategies and performance tooling.

Quick WebSphere Application Server (WAS) Update

Why Upgrade to WAS V6.1

WAS V6.1 has enhancements that has taken away key pain points. The new additions that I think are important for mission critical enterprise systems are:

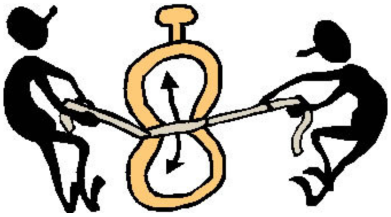
1. **Integrate voice conversations, television and gaming engines into new or existing applications via new SIP technology.** WAS V6.1 enables these benefits via Session Initiation Protocol (a.k.a., SIP, SR-116) which holds the key to the "Quadruple Play" – voice, video, data and mobility convergence. For example, two users accessing the same server or different servers can interact. The key is both servers can be located via standard Internet DNS. A voice conversation can be set up by allowing the servers to broker the connection. SIP sessions can interact with the other major protocols - HTTP and portlet based components.
2. **Portlets can now be run out of WAS without WebSphere Portal Server.** The intention is to allow JSR 168 compliant portlets to be run out of the WAS web container. Note that there is a significant benefit to WPS with content management, search and personalization which is absent in WAS V6.1.
3. **Multiple LDAP servers can easily be configured directly in WAS.** In previous versions of WAS, only one LDAP server could be configured directly in WAS. Therefore, companies that needed information from multiple LDAP servers had to write code to access all of them using the custom registry interface (a.k.a., CUR). Often, the CUR version produced by the application group did not run well and turned into a performance bottleneck. For higher availability configurations redundant servers can be listed.
4. **Key Management and SSL configuration is vastly simplified – security productivity takes off.** This means far less mis-configuration of keys and less time spent on key management.
5. **Increased productivity and security at install versus custom post-install product hardening - install as Non-Root Users.** Most enterprises that demand high security such as banks require the base product to be run as a non-root user. Running as a root user is a very low security solution, which most financial institutions don't want to do - based on either good business practice or for regulatory reasons (a.k.a., Sarbanes-Oxley). Previous versions required the installation as root than lengthy procedures to change WAS files, directories and executables so they ran as non-root. That goes away in WAS V6.1.

6. **Web services extension for notification/events (WS-N) and tighter business level transactional capacity (WS-BA).**
7. **Richer library of pre-built JSF widgets saves development time.**

Destroying Applications Performance Bottlenecks

A bottleneck is a well-known term in operations management and is defined as a resource that cannot meet demand. In the case of applications, this is typically response time as seen by users. All kinds of systems experience bottlenecks including production of things such as clothes, creation of hamburgers off a McDonald's assembly line or in the case at hand, IT systems. For IT systems, specifically, the key to destroying bottlenecks is to consider performance analysis throughout the development lifecycle and into production. First, understand the Theory of Constraints, which is relevant for all systems. Second, understand that you need to minimize bottlenecks to meet SLA requirements rather than in absolute terms.

Theory of constraints affects all systems



The first step in performance analysis is to understand Goldratt's Theory of Constraints:

1. **Identify the system constraints –** no improvement is possible unless the constraint is identified.
2. **Decide how to exploit the system constraints** – make constraints as effective as possible.
3. **Subordinate everything else to that decision** – align other resources to support the constraint even if it reduces the efficiency of the non-constrained resource.
4. **Evaluate the system constraints** – if output is less than demand, then acquire more of the resources needed to satisfy the constraint.
5. **Continuous improvement** – other constraints surface as the current constraint is resolved. If demand (SLA levels) has not been met, then you must go back to step 1 and continue the process.

Key things to remember:

1. **A second lost at a bottleneck is a second lost in the system.**
2. **A second saved at a non-bottleneck is a mirage.**
3. **Place quality control points in applications before suspected bottlenecks (constrained resources) – typically, that's data validation checks.** Delays at bottlenecks translate to delays in the entire system. Bottlenecks should not receive defective work in progress – in IT applications that would be a user request. There should be quality control inspections prior to bottlenecks to ensure that the bottleneck works only on good products. In IT applications, that means that data validation checks should be coordinated across several layers of the IT architecture and potential quality control points include clients, services and database access.

Since database connections and resource are typically the most costly resources in an application solution, quality control points in the application should be placed before accessing this layer.

Development to Production – Driving out Bottlenecks

To meet quality levels required of most mission critical applications, several levels of testing should be done including functional (component and application level), concurrency, longevity and system. The followings are key considerations:

1. **The faster the feedback in correcting bottlenecks, the faster the staff learns and eradicates potential bad practices.** With several levels of performance testing each with its own goals, the initial levels of testing can give fast valuable feedback on what works and what doesn't.
2. **Different anomalies arise during different levels of testing and if only one level of testing is completed, potential performance defects will likely be missed**.
    a. Component testing will help developers detect memory leak and performance issues related to one component in isolation. However, the application must remain robust while many components and users are active. Increased component level testing by operators has been very successful at Japanese automobile manufacturers. Japan's major productivity gains are the result of social changes brought about by Statistical Quality Control (a.k.a., SQC). Japan deploys more operators in direct production work than Ford or GM. SQC almost always put in place more operators offset by a SHARP drop in the number of operators such as quality control inspector typically positioned at final production. They found that both the number of inspectors and amount of rework went down.
    b. Application level functional testing stresses the application architecture and algorithms in ways typical of a real user. Unfortunately, the architecture and algorithms may not be optimally structured to maximize performance and memory usage.
    c. Concurrent loads help test for synchronization, deadlock and other timing issues that may arise. This could negatively impact performance and resource utilization.
    d. Longevity tests help indicates where time dependent defects may arise. Typically, slow memory leaks show themselves with this type of testing.
    e. System testing helps discover performance defects related to major component introduction stressed to production SLA levels.
3. **As more people and resources get involved at each stage of testing, the costs go up.** Therefore, there is a large incentive at each stage to catch issues relevant to that stages required outcome. Testing is done in stages so that the complexity and amount of issues can be reduced within each organizational group. Development focuses on applications and software architecture. Later stages of the delivery "pipeline" focuses on physical architecture. Each stage depends on the last, expecting that only integration gottcha's arising out of movement from one environment to the next.

4. **For mission critical applications, the degree of rigor in the planning and testing of an application should be high.** Applications are never 100% bug free. In a commercial environment, there is always cost benefit trade-offs. It is never a good idea to spend $10 to make $5. Activity Based Costing (ABC) should be use to determine the limits to rigor. In addition, actual usage patterns are hard to test in new products, therefore, sometimes all the additional testing won't help – it's past the organizations "point of knowledge". At that point, you have to run a pilot and find out through experimentation.

How should we select performance tooling?

IT staff comprised of architects and non-business trained personnel often judge the tools on technical superiority and will often include major decision points on the tools price. Total cost is not the price. Costs include:

Total Cost = Monetary Costs + Time Costs + Energy Costs + Psychic Costs

Therefore, evaluation of any product should include

Total Evaluation of Sourcing = Business + Technology considerations

1. **For mission critical applications** that often provide competitive advantages as well as enhance company brand, the price of the components that are a small percent of the overall solution cost, are not that relevant to the decision process.
2. **Labour Cost = Labour Rate X productivity factor.** Labour rate is what companies are billed on paper. Many decision processes in enterprises assume Labour Cost = Labour rate. Productivity varies widely for workers and typically has the highest correlation to intelligence and training level – the higher the better. The right tools and environment are also important.
3. **All business research indicates that speed of delivery is highly correlated to competitive advantages and profitability**. Small gains in productivity in critical situations can mean the difference between being 1st in the industry and 10th. Golfers like Tiger Woods and Phil Michelson know this well.

WAS Performance Tooling - Development

The performance tooling for development environments is somewhat less restrictive compared to target production environments. This is because of the development world is either on Windows or Linux which almost every tooling vendor supports. The main engine for software engineering of IBM platform solutions is RSA/RAD (Rational Software Architect/Rational Application Developer) for design and code development. Other tools in the Rational line such as ClearQuest and ClearCase help with builds, versioning and defect tracking. There are several options for performance tuning including JProfiler by E-J Technologies, JProbe by Quest and the built-in performance profiler in RAD/RSA.

1. **JProbe –** this tool was first targeted at BEA's Weblogic tool and IBM paid Quest (formally owned by KLGroup) to port it to IBM WebSphere Application Server for V3.51. There are two modes, one for examining execution time and CPU utilization and the other for memory analysis. Although you have to restart the JVM for each mode, this is not likely a developer performance issue because the strategy is to look at memory anomalies first then execution times second – not both. The reason for this is high memory usage often impacts CPU utilization with swapping to disk, paging and garbage collection.
2. **JProfiler –** This tool has capacities to examine both execution time and memory usage in all one instantiation. It also analyzes threads for potential deadlocks.
3. **RSA/RAD Profiling –** This tool has basic features that capture execution time and memory utilization down to the method level. The data capture of the program execution allows reverse engineering via the creation of sequence diagrams.

WAS Performance Tooling - Production

In the past, only a few systems were involved in comprising an application. Today, applications are cobbled together across several different systems and layers of applications and services. This makes problem determination for production staff more difficult today than ever. Therefore, tools used in production environments for problem determination and performance bottleneck detection have to take into consideration additional concerns including:

1. **Platforms –** Most organizations have a mixture of operating systems. This includes AIX, Solaris, Linux on System z and others.
2. **A master tool or many single component tools -** Will you introduce point products that address performance on individual components and rely on staff to manually coalesce all the information or select a tool that does the consolidation for you.

3. **Transactions or Statistics –** Are tools that can track aggregates, averages and statistics or only individual transactions sufficient.

Top tools available including CA's Introscope, IBM Tivoli's ITCAM, Mercury Interactive's Application Management (AP) and Quests Foglight and Performasure. Most of the tools will capture data down to the method level but not the lines of code that may be causing the highest resource utilization. At the method level, the problem could be the application or it could be long synchronous blocking on calls to other components in the enterprise such as LDAP or databases, to name a few. Below is a short synopsis of each of the top tools along with a table listing the capacities of each tool.

1. **IBM ITCAM – IBM Tivoli Composite Application Manager for WebSphere V6.0.** ITCAM has two major points of difference: 1) Built in three levels of progressively deeper information gathering that can be configured dynamically and 2) It's the only tool that collects CPU utilization data. ITCAM has three levels of information – level 1 – basic system information, level 2 for problem determination and level 3 for performance analysis. Each level of information gathering gets more intrusive and consumes more CPU resources. ITCAM is the only tool in the group that can capture CPU utilization down to the method level. This is important because you can look at execution time and CPU time and if both are about equal, it's a safe bet the method is the result of the high resource utilization. What it doesn't have and would be valuable is the ability to statistically analyze hot spots so you have an idea whether the hot spot is consistent or sporadic. Sporadic hotspots with very low occurrences, at say 1 in 10,000 requests, may not be a top priority. It also doesn't support WebSphere Portal Server on z/Series as of the time of writing this article. Lastly, it covers SQL and Lock analysis and can even inventory your version such as JVM's and WAS binaries for discrepancies.

   http://www-306.ibm.com/software/tivoli/products/composite-application-mgr-websphere/

2. **Mercury Interactive's Application Management combined with its Diagnostic Probe (a.k.a., AP (formally Topaz)) and Sitescope –** MI's key advantage is collection of component data that can be combined with LoadRunner statistics to provide a consolidated analysis. This is a real advantage because the integrated analysis makes it much easier for staff to synthesis solutions for tough problems. AP instruments the JVM while Sitescope extracts JVM PMI data. The limitation on AP is it doesn't obtain CPU utilization data tools.

3. **Quest Foglight is an operational tool and PerformaSure is a diagnostic tool –** The Foglight tool is used for broad black box analysis of applications collecting responses times and statistics that help administrators and management determine whether they are meeting SLA agreements. Once metrics are found to be out of tolerance, PerformaSure helps production and development groups find out why. According to Quest Foglight's available information, it has not been officially certified for WAS V6, nor is officially ready for anything on System z (formally zSeries).

4. **CA Introscope (formerly Wily) –** Introscope's major point of difference is that it is has a large market share in the applications operation and diagnostics space. It can consolidate data

collected from Web Servers, WAS, WMQ, CICS transactions and WMB adapters such as SAP and PeopleSoft. Currently, the only database it has an agent for is Oracle that allows collection of deep database statistics. It still can gather SQL JDBC information for statement analysis like most of the other tools.

2006, the year of the Satellite – running, navigating and radio listening

Only small portions of the population have bought into time saving satellite based devices. Here is why this is a disruptive technology:

1. **Many executives run to guard against stress and to stay in top form**. Runners know exactly their distance travelled and path taken when using a GPS watch. The Garmin Forerunner 305 will track your exact distance, give you a rough estimate of calories burned, time, pace and heart rate. You can download the data to a computer and use the software to track trends. I have records in my watch for the last 1000 miles of running. Also, on a run in the woods, I lost track of how to get back. The map feature shows the path back to the start.

2. **Car rental GPS navigation systems will save you time and money and maybe your life.** For an additional $10/day, HERTZ will rent you a Neverlost system. I have been using it across North America and it is very accurate. It will save you time in getting to and from the airport, to customer sites and possibly from driving accidentally into a high crime areas. Tall buildings may block reception so you may get cut short on directions.

3. **Portable satellite radios will entertain as well as inform you, anywhere in North America.** Both XM and Sirius have good entertainment stations such as classical, jazz, instrumental and rock music, etc. Both have CNN and financial channels which help me stay in touch. The major disadvantage is the poor reception in some areas where satellite signal is not available.

Dimension	Quest Foglight 4.2	Quest PerformaSure 4.3.5	IBM ITCAM For WAS V6	Mercury Interactive Diagnostics Probe	CA Introscope 7.0 7.0
Purpose	Operations High Diagnostics	Deep Diagnostics	Operations Deep Diagnostics	Deep Diagnostics	Operations Diagnostics
Integrating Products	Spotlight Central PerformaSure Big BrotherTM	Foglight Spotlight JProbe	SNMP compliant tools	LoadRunner – QA Application Management Foundation (formerly Topaz) – Prod Business Availability Center	SNMP compliant tools and specifically: LoadRunner – QA HP Overview Tivoli Enterprise Console
Key Point of Differences	Tools used from development to production that integrate. Tools specialist	Tools used from development to production that integrate. Tools specialist	Three Levels of Intrusiveness – changeable dynamically. IBM has the strongest financial stability amongst all the competitors in this space.	Has the best integration with LoadRunner for diagnosing and duplicating problems in a controlled QA environment. Load runner is the defacto standard for load testing with 80% market share.	Top selling tool in the application problem determination segment means it will be around for a while Fully customizable dashboards makes very flexible for advanced users. Monitoring on/off dynamically but only with JDK 1.5 or higher
Disadvantage	No official support for System z or OS/400	No official support for System z or OS/400 No method level CPU utilization which is especially important of System z where charges occur based on CPU utilization.	No RDMS data collectors to tie in with SQL with database performance statistics	No official support for System z or OS/400	No CPU utilization counters CA will be around for a long time but has recently had financial reporting issues with SEC in the US.
Platforms – WAS V6					
Mutliplatforms	✓	✓	✓	✓	✓
System z – z/OS	X	X	✓	X	✓
System z - Linux	X	X	✓	X	X

Dimension	Quest Foglight 4.2	Quest PerformaSure 4.3.5	IBM ITCAM For WAS V6	Mercury Interactive Diagnostics Probe	CA Introscope 7.0 7.0
OS/400	X	X	✓	X	✓
Collectors – External Components - Retrieve statistics					
Operating System	✓	✓	✓	✓	✓
Network	X	✓	X	X	X
CICS TS	X	X	✓	X	✓/CTG
IMS	X	X	✓	X	X
DB2 Agent	✓	X	X	✓	X
Oracle Agent	✓	X	X	✓	✓
Sybase Agent	✓	X	X	✓	X
MS SQL Server Agent	✓	X	X	✓	X
Web Servers	✓	✓	X	✓	✓
WMQ V6.0	X	X	✓	X	✓
WMB V6.0	X	X	✓	X	✓/Adapters only
Collection Level – WAS V5/V6 J2EE					
Method Level	X	✓	✓	✓	✓
Method Level Execution Time	X	✓	✓	✓	✓
Method Level CPU time	X	X	✓	X	X
Exception Anomalies	X	X	✓/Uncaught Exception only	✓	X
Synchronization & Locks	X	X	✓	✓	X
Memory Analysis	X	✓	✓	✓	✓
SQL/JDBC Analysis	X	✓	✓	✓	✓
Portal Specifics	X	X	X	X	✓

Table based on latest freely available vendor documentation and Toronix experience.