IBM ECM
Enterprise Content Management

IBM

IBM Enterprise Content Management Performance Methodology

Performance planning, testing, tuning, monitoring, and troubleshooting

http://www.ibm.com/support/docview.wss?uid=swg27045462

November 2015

Disclaimer

This document is provided “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranty of merchantability or fitness for a particular purpose. This document is intended for informational purposes only. It could include technical inaccuracies or typographical errors. The information herein and any conclusions drawn from it are subject to change without notice.

Many factors contribute to ECM system performance and IBM does not guarantee comparable results. The actual performance in customer environments with production workloads will depend on the unique circumstances of each customer's hardware and software configurations, data model, and workload, and many factors including other applications running on the systems and configuration of the storage or network.

IBM ECM Performance Methodology

Performance planning, testing, tuning, monitoring, and troubleshooting

- This presentation provides an overview of the IBM ECM Performance Methodology for building and maintaining high performance and scalability in complex real-world ECM deployments
- The IBM ECM Performance Methodology includes

  1. Planning for performance
  2. Pre-production performance testing
  3. Performance tuning
  4. Monitoring and maintaining performance
  5. Performance troubleshooting

- In this presentation IBM Content Navigator and IBM Case Manager are used as models to illustrate the concepts, but the Methodology can be applied to any Enterprise Content Management system

Planning for performance

- Start with an initial sizing and capacity planning estimate
  - Your IBM account representative and ECM Lab Services have tools to assist with sizing and capacity planning based on your unique solution and workload
- Design your solution with performance in mind
  - It’s better to design in performance from the start
- Model and test your own expected workload and system configuration
  - Test the full application stack (for example, using Rational Performance Tester, JMeter, or similar tools)
  - Test both single user UI performance and under load
- Set performance objectives and plan for an initial tuning period
- Create a “performance profile” as a baseline for production monitoring and capacity planning
  - Including the workload operation mix, resource utilizations and system vitals (CPU, memory/JVM/GC, network, disk, FileNet System Dashboard), response times, and throughput
  - All elements should be collected and reviewed periodically over time from production and modeled in the test environment (for example: daily, weekly, or monthly, depending on your requirements, with longer-term reviews quarterly and yearly)
  - This also creates a baseline for comparison when upgrading fix packs or to a new release, or when bringing on new workloads

IBM Content Navigator solution design for performance

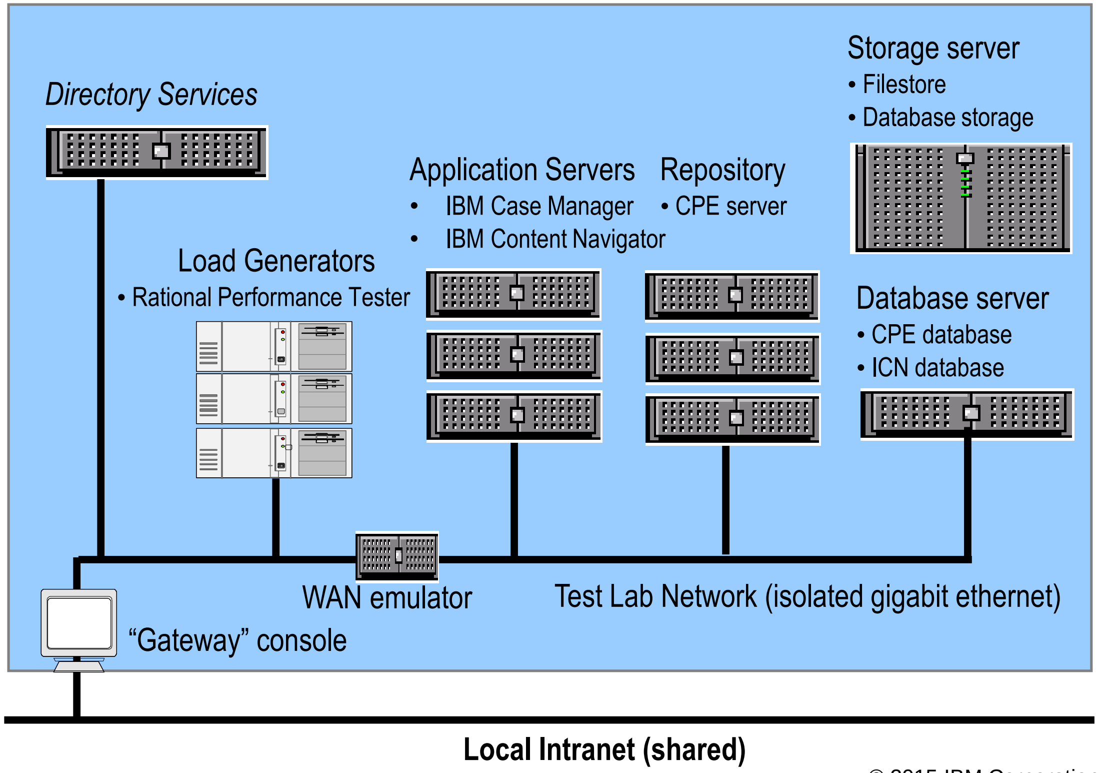
- Optimize logon times
  - Create a good default for the home page after logon. For example, avoid Browse if you have a large number of root-level folders, Search Templates, Favorites, or number of object stores. Understand the performance impact of custom plugins.
  - You can have different desktops for different business cases
  - Optimize LDAP lookup times:
    - The number of LDAP groups a user belongs to can affect logon time. Avoid very large (hundreds)
    - If LDAP servers are not local, optimize their WAN performance
    - Custom group filters can improve logon times for users who belong to large numbers of LDAP groups
- Optimize use of external data services
  - ICN EDS REST protocol: best performance, not chatty, and is meant to handle 80% of typical use cases
  - Plugins provide full power and capability, but give you full responsibility for performance
    - Cleanup to avoid memory leaks
    - Understand the performance impact on both server and client sides
    - Provide monitoring and tracing for serviceability
- For integration with custom applications, optimize the access path
  - An example of a non-optimal integration: another application generates a URL to launch ICN in a browser window. Each time a user clicks on the URL from the other application ICN is launched and the user is treated as a new user with a new session with a new logon.
  - To improve this, build the integration in a such a way where the user is authenticated once and on all subsequent request from the same user pass the sessionId on the URL so that the same session is reused.
- Understand the performance considerations for key features
  - Download as PDF or use of the PDF Conversion Viewer
  - Preview documents or use of the HTML Conversion Viewer
  - Thumbnail rendering (Content Navigator thumbnails vs FileNet CM thumbnail sweeps)
  - Understand the performance differences between the IBM Daeja ViewONE Virtual and Pro Viewers, and how viewing performance is affected by document size, type, and complexity
  - Viewing AFP or Line data with AFP2PDF or Line2PDF conversion
  - Sync, if users are treating it as a backup for their C:/

IBM Case Manager solution design for performance

- Role-based custom views can improve performance by avoiding the retrieval and presentation of unneeded properties
- Consider the performance impact of external services and overly-complex visual presentations
  - IBM Case Manager can also use ICN EDS REST for better performance with external data sources
  - Combine and/or parallelize multiple calls when possible
  - Maximize HTTP object caching and sharing (including for external web services) and compress large payloads
  - Defer processing when possible (for example, multiple pages instead of a single large form)
  - Set reasonable result set size limits
- Organize properties views to optimize performance using lazy loading
  - If a view has a large number of properties, organize them into tab containers and collapsed titled layout containers to minimize the number of visible properties to load during the initial loading and to optimize the perceived loading time of view
  - Find a balance between having too many properties into any container, and too many containers with small numbers of properties
  - Visible properties at the top of the view are loaded first, followed by visible properties further down the view
  - Properties in collapsed titled layout or alternate tabs are "lazy-loaded" after all the visible properties are loaded
- Other recommendations
  - Avoid creating one large case task that encompasses the entire business process. Separate into smaller meaningful tasks as either required or optional
  - Reuse case types, document types, and properties to avoid creating many essentially duplicate elements
  - Use “white-box profiling” to optimize the solution design for WAN deployments (see the “More information” slide)
  - Use up-to-date browsers for best performance

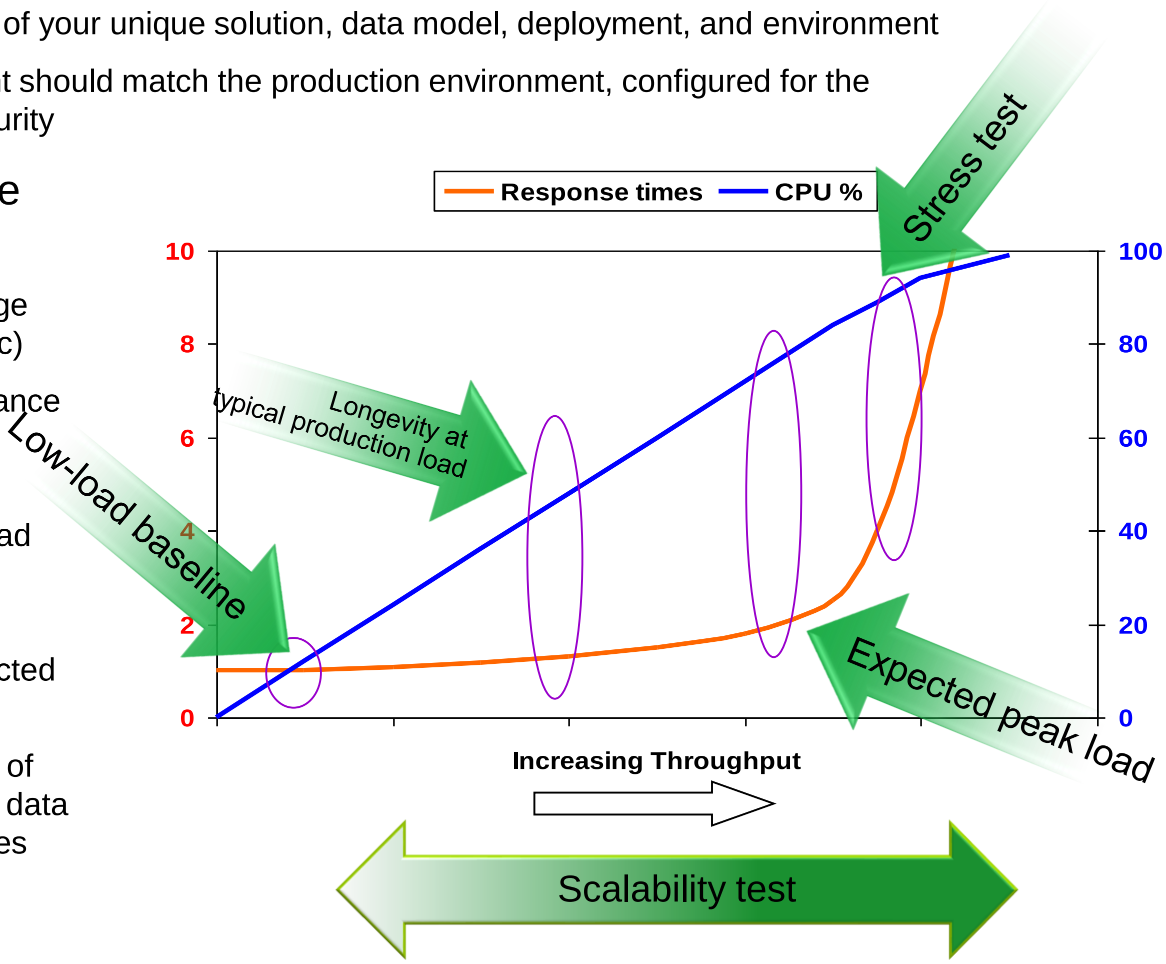
Performance testing – summary

- IBM takes the performance testing of our ECM software very seriously
  - Every release is held to very strict performance requirements. No regressions from previous releases are allowed, and the development of new features is not allowed to impact performance for customers that don't use them
  - Testing is done in an isolated-network non-shared performance test lab to ensure that results are fully reproducible and highly accurate
  - We're not a "benchmarking center". We're not looking for the "best numbers", but demonstrating that our software is built to perform and scale in real-world environments
- Performance in customer environments will naturally vary
  - Environmental factors, such as network quality and latency, storage latency and bandwidth, deployment patterns, etc
  - Solution factors, the unique differences in your data model, solution design, and documents, etc
- Customer pre-production testing builds on the proven underlying ECM software performance
  - Evaluate and tune the performance of your unique solution, data model, deployment, and environment
  - Establish the initial Performance Profile to support ongoing production monitoring, tuning, and troubleshooting



Performance testing – details

- Thorough pre-production testing increases confidence and reduces risk for the production system
  - Set realistic performance objectives and plan for an initial tuning period
  - Evaluate and tune the performance of your unique solution, data model, deployment, and environment
  - The pre-production test environment should match the production environment, configured for the production profile with elevated security
- Evaluate end-user performance under realistic conditions
  - Model the expected production usage (operations, data, load, topology, etc)
  - First demonstrate baseline performance at low load
  - Then move on to longevity and scalability under load, with a workload that models expected user activity, gradually ramping up
  - Stress test at loads above the expected peak load
  - Pre-populate with a background set of documents, cases, etc with realistic data volumes to make sure that databases and storage are properly tuned
  - Use load testing to prepare for production monitoring



Performance testing – details

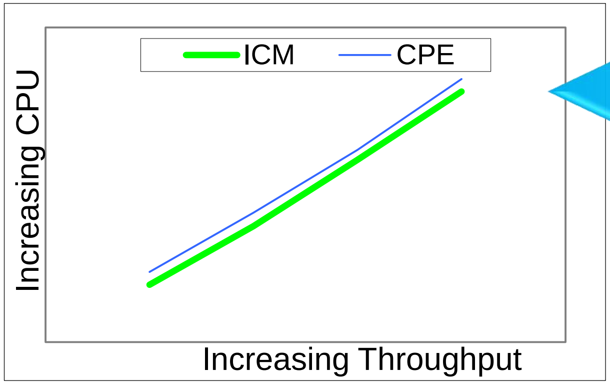
Two key concepts in performance testing

1. Scalability: vertical and horizontal
2. Cold, warm, and hot end user access

Cold = 1st time user
- Browser cache is empty, all resources must be retrieved, worst performance

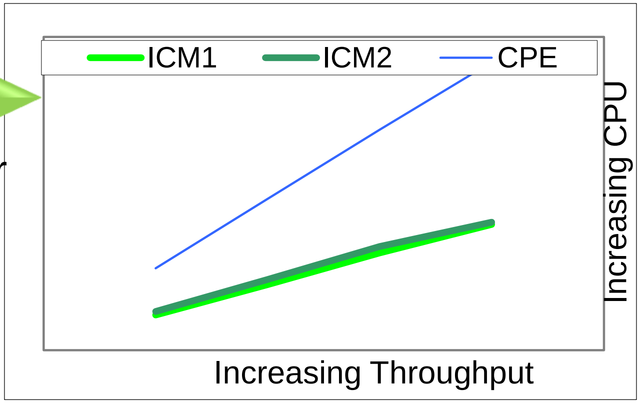
Warm = 1st time today
- Most resources cached, but all session-specific data must be retrieved

Hot = middle of workday
- Everything cacheable is cached, best possible performance



- **Vertical scalability**
- A linear match between throughput and CPU usage as load increases

- **Horizontal scalability**
- An evenly balanced cluster under increasing load



Performance testing – details

- Strategy for creating test scripts
  - Use a UI testing tool such as IBM Rational Functional Tester (RFT), open-source Selenium, etc
  - Use a load testing tool such as IBM Rational Performance Tester (RPT), open-source Apache JMeter, etc
  - Understand the difference between Cold, Warm, and Hot end user accesses
    - Important to ensure the load test is a valid measure of the end users' actual experience
    - Important for evaluating the impact of WAN deployments
  - Create role-based scripts for each of the most important end-user roles, so that each VU (virtual user) emulates one specific type of end user
    - If the solution has inboxes for each user or user role, that makes driving role-based scripts more convenient. Then at test runtime, you can launch VUs for the role-based scripts in proportion to the numbers of actual end users with those roles
    - Balance the numbers of VUs and adjust script "think times" so the overall workmix models the expected system use
  - An alternative to role-based scripts is to have each VU follow a single document or case through its full lifecycle, switching roles as needed for each of the various users who interact with it
    - The potential problem with this approach is that if the script needs to repeatedly log out of one role to log into the next role for users interacting the case, then the proportion of logout/login operations in the overall workmix can easily become unrealistically large, distorting the results unless this actually matches the expected production use

- Strategy for prepopulating the test repository
  - For testing, prepopulate a "background" set of documents and cases so the repository is large enough that (for example) database indexes will be exercised in a realistic production way
    - When database tables are too small, the database query plan optimizer may choose table scans instead of using indexes as being more efficient, which is not usually the case for production environments. These table scans can lead to concurrency issues (eg, lock waits and deadlocks) during testing that would not occur with the full-size production database
  - Populate objects in a range of valid states, in proportion to their expected occurrence in production
    - If the role-based RPT scripts are designed so that as a set they generate and consume objects in a realistic way, then they will naturally keep the system in an overall valid state where no individual script will run out of work to do -- eg, having an empty inbox (unless that is actually an expected production situation)
  - An alternative population approach is to build the pre-population so that there are already enough cases in the needed states sufficient for an entire test run
  - Whenever possible, use real documents with the range of types, sizes, and complexity you will see in production

Performance tuning

- The tuning process is similar to the performance troubleshooting process
  1. Understand your success criteria
  2. Test with a real workload modeled on expected production usage, and capture detailed metrics to identify bottlenecks
  3. Review the performance tuning resources to identify appropriate tunings specific to the bottlenecks actually found
  4. Iteratively implement the solution – change one thing at a time and retest, only keeping tunings that help
  5. Stop when you are successful

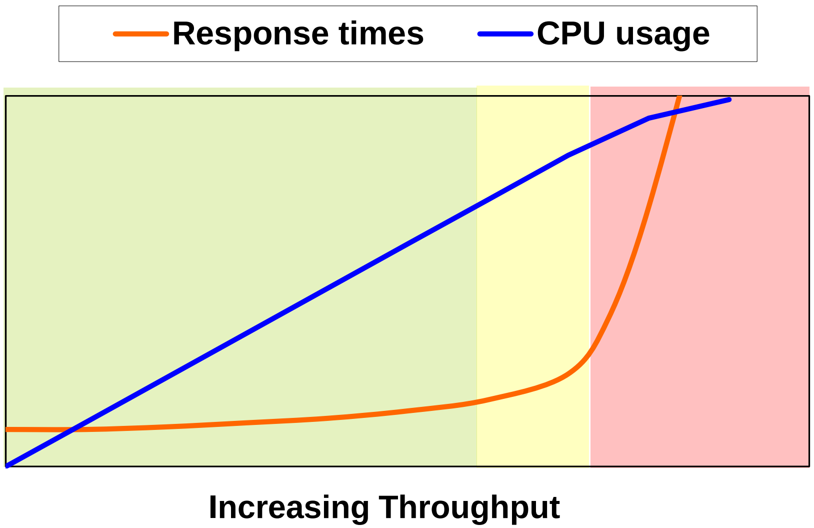
- Performance tuning can touch all components in the system
  - Operating systems, databases, application servers, directory servers
  - The Knowledge Center has tuning recommendations for all these areas

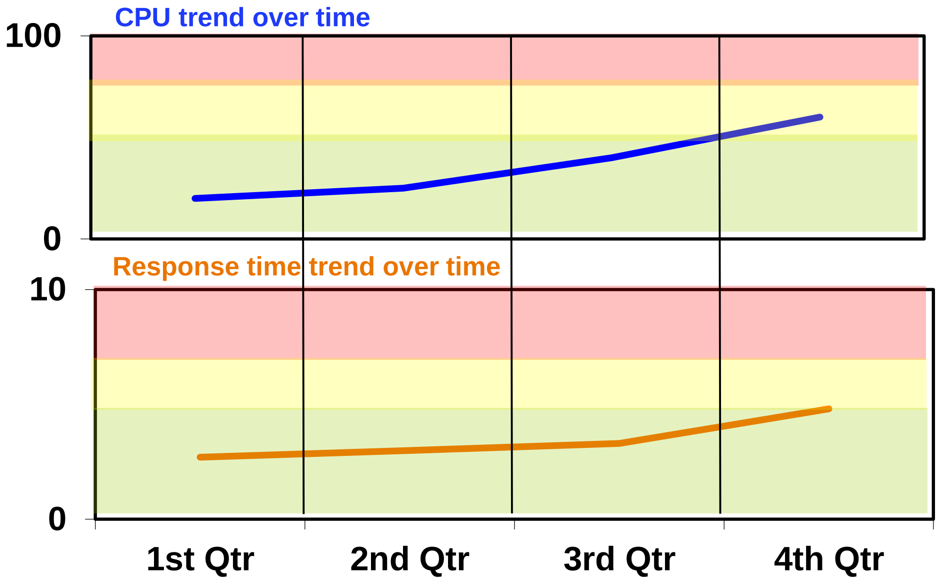
- Common aspects to tune for IBM Case Manager and IBM Content Navigator
  - Adjusting JVM heap sizes
  - Adjusting cache sizes and refresh/timeout intervals
  - Tuning database connection pools, web container thread pools, and the ORB
  - Creating workload-specific database indexes
  - Tuning transmission protocols
  - Database compression
  - Other tunings for the underlying FileNet Content Platform Engine repository

Monitoring and maintaining performance - summary

- Maintain an ongoing “performance profile” of the workload metrics and resource utilizations on your ECM servers
  - Create production procedures to monitor overall usage patterns, to identify when peak resource usage and/or overall usage patterns change substantially

- The Green, Yellow, Red Zone Approach
  - Green: performing within acceptable boundaries
  - Yellow: increased resource utilization or response times, approaching or on a trend to approach acceptable limits
  - Red: system performance actually degraded outside acceptable range





Monitoring and maintaining performance - details

- Create production procedures to monitor overall usage patterns, to identify when peak resource usage and/or overall usage patterns change substantially
  - Compare against the initial baseline in order to quantify any performance changes
  - Continue to monitor over time to observe trends **before** they become a problem
  - Perform more detailed performance analysis for appropriate areas, and tune the system configuration as required

The Green, Yellow, Red Zone Approach

- A very useful way to think of system performance is to picture your overall system as being in a “zone”. A ‘green zone’ implies that the system is performing within acceptable boundaries and is not in any danger of imminent problems due to over-utilization of system resources
- The system enters the ‘yellow zone’ when one of the monitors shows that resource usage is abnormal in some way. For example, suppose a Windows server has 4GB of real memory and approximately 2GB free memory when it is in its “green zone”. If the free memory were to drop to 1GB, then you could say that they system has gone into the “yellow zone”. The server is still performing fine and there is plenty of available memory. However, this sudden abnormal usage of memory warrants investigation by the system administrator and further monitoring or tuning activities need to occur
- The system enter the ‘red zone’ when system performance degrades because of unavailable system resources. In the previous example, it is possible that library server searches were taking more and more memory because of the way the metadata tables were indexed and because of their growing size. If the system administrator didn’t take action when the server was in the ‘yellow zone’, eventually the amount of free memory would fall to the point where excessive paging was occurring and slowing down system response time. Now the system has gone from ‘yellow’ to ‘red’ and users are suffering as a result

- In order to know what a system’s “green zone” is requires tuning the system, comparing throughput with expected results using the sizing tool, and then creating a baseline profile using the monitoring tools described previously. There isn’t a mathematical formula or set tolerance level for determining when the system has gone “yellow”. Any unusual departure from the “green zone” (but before performance is impacted) means the system has gone “yellow”

- A system administrator should intuitively know when the system has gone into the “red zone”. This is usually accompanied by their phone ringing as users call to complain about system performance. But there are more empirical measurements that an administrator can look for to show that the system is in imminent danger

Performance troubleshooting - summary

- When performance issues arise, use a disciplined and organized approach to solve the problem

1. Clarify and document the problem and success criteria
2. Capture detailed metrics to isolate the bottleneck
3. Identify appropriate potential tunings
4. Iteratively implement the solution, documenting results
5. Stop when you are successful

Performance troubleshooting – details

- The key to successfully and efficiently resolving performance problems in an ECM system is to have an organized, disciplined process that stays focused on identifying and relieving the system's performance bottlenecks. Without an organized process focused on identifying bottlenecks it is very easy to spend a lot of time and energy tweaking various parameters, ending up with a system in a worse state than before and no good record of what was changed and why.
- To be successful, you should use what we call Data-Driven Performance Analysis: collect the right data and let the facts lead you to the real bottleneck. Then you will be able to focus on tunings that will address that bottleneck and actually improve the performance of your ECM system.
- But before even beginning the full performance troubleshooting process, first ensure that the database statistics for the Content Platform Engine database are up to date. The combination of IBM Case Manager or IBM Content Navigator with FileNet Content Manager is a large-scale database application, and so depends on accurate database statistics for optimizing all operations. Updating database statistics on the CPE database is quick and easy to do, and often turns out to be the only "fix" needed.
- If you have been collecting performance profiles in your production monitoring you will find you likely already have most of the information needed. If not, begin collecting them now to help resolve your current performance problem and you will be better equipped to effectively handle any future performance problems. If at the end of the troubleshooting process you find your success criteria can't be met, you will have a full description of the problem and detailed traces for IBM Support or Services.

Performance troubleshooting – details

1. Clarify and document the problem and success criteria

- It is very important to properly clarify and document the performance problem, to make sure that the troubleshooting stays focused on the real problem, and on clearly specifying the success criteria, so you will know when you are done. If you have them, use the system/workload descriptions and performance objectives from your performance planning phase. If you don't, create them now!

Checklist

- □ Is this a single-user performance or multi-user scalability problem? That is, does the problem occur when the system is lightly loaded, or only when the server is under load from many users?
- □ Is this a regression, or were the performance objectives never met?
- □ Are all operations affected or just specific operations?
- □ Is the problem worse at different times of day?
- □ Does the problem affect end-users, batch operations, or both?
- □ Can the problem scenario be simplified?
- □ Are there functional failures associated with the performance problem?
- □ Define specific success criteria: how will you know when the problem is solved?

Performance troubleshooting – details

2. Capture detailed metrics to isolate the bottleneck

- The goal here is not to immediately solve the problem, but to narrow down the problem and identify the bottleneck server and/or ECM component. Use the tools and baseline data from your performance profiles. Don’t make changes yet!
- Start with the high-level ECM component performance traces and server machine resource utilization metrics to isolate the machine, ECM component, and specific operations that have the performance problem. Once those have been identified, a second pass of traces may be needed to gain more detailed diagnostic data.
- Common tools for collecting resource utilizations are NMON for AIX or Linux, and PerfMon for Windows
- The IBM System Dashboard for Enterprise Content Management is a good tool for collecting performance data for both troubleshooting and monitoring trend analysis

Checklist

□ Document server configuration and parameter information

□ Document all system and ECM tuning parameters changed from out-of-box defaults

□ Capture ECM component performance traces (IBM Case Manager, Content Platform Engine, database)

□ Capture workload metrics: operation mix, response times, throughput

□ Capture server resource utilization metrics: CPU, network, memory, disk

□ Capture other performance traces, as needed: DB2 snapshots, WebSphere JVM verbosegc, operating system tools, FileNet System Dashboard

Performance troubleshooting – details

3. Review the performance tuning resources

- Now that you have accurately characterized the problem and isolated the bottleneck system component, use the performance tuning topics in the IBM Case Manager, FileNet, DB2, and WebSphere Knowledge Centers to identify potential tunings for the problem areas identified. Search the Support site for applicable Technotes.
- Identify any system resource bottlenecks that can be relieved with appropriate tuning or increased capacity (see the next slide).

4. Iteratively implement the solution

- Don't lose focus and apply a whole set of tunings all at once. If applying one tuning makes things worse, back it out and try another.
- Some tunings can have side effects, so capture a full set of metrics to make sure that improvements in one area are not accompanied by degradations in other areas.

5. Stop when you are successful

- Once your success criteria are met, stop the tuning process – don't tune just for the sake of it.
- If the success criteria can't be met, you will have a full description of the problem and detailed traces for IBM Support or Services.

Key system metrics for performance troubleshooting

CPU utilization and saturation

- Key metrics:
  - – Utilization % (system, user, wait)
  - – Processor Queue Length – shorter is better, showing minimal wait time
  - – Context Switches (switching between threads), number of threads and Process

- What to look for:
  - – Overall utilization above 60-80% along with high processor queue length
  - – Excessive context switching and abnormally high number of threads and/or processes

- What to do:
  - – Isolate the program that is spawning excessive threads/processes
  - – Excessive context switching may indicate other resource bottlenecks
  - – Allocate more or faster CPUs

Key system metrics for performance troubleshooting

Memory utilization and saturation

- Key metrics:
  - – Available memory
  - – Page Writes/s, Reads/s
  - – JVM GC intervals, duration

- What to look for:
  - – Not enough memory available
  - – Hard page faults resulting in hard disk reads
  - – Excessive GC (> 5% of the time, >1 sec pauses)

- What to do:
  - – Adjust JVM heap size or number of JVMs
  - – Adjust session cache size or timeout
  - – Increase amount of physical memory
  - – Stop unnecessary background processes

Key system metrics for performance troubleshooting

Disk utilization and saturation

- Key metrics:
  - IOPS and disk throughput (read and write)
  - Disk Queue Length (Read and Write Queue)
  - CPU I/O Wait (UNIX)

- What to look for:
  - Windows: High disk queue length (>2 per spindle)
  - UNIX: High I/O Wait (>20%) along with high disk usage
  - High latency (>10ms) for SAN storage

- What to do:
  - If possible, tune application to use more physical memory as cache
  - Increase SAN bandwidth and parallelism, and consider using RAID 10

Key system metrics for performance troubleshooting

Network utilization and saturation

- Key metrics:
  - NetIOPS and throughput (read and write)
  - Network errors (dropped or lost packets)
  - WAN latency

- What to look for:
  - Saturating bandwidth (>30% capacity), look for “chatty” protocols
  - Errors possibly due to misconfigured network cards
  - High latency

- What to do:
  - Reduce latency
  - Increase link speed (network bandwidth)
  - Match duplex settings between systems
  - Look for “chatty” protocols

More information

- IBM ECM Performance Optimization Services
  - Solution design, system health check, performance tuning, and capacity planning services from the ECM Lab Services team

- ECM performance-related information resources
  - The ECM Performance and Scalability Library
    - `http://www.ibm.com/support/docview.wss?uid=swg21970857`
    - Your “one stop shop” for performance studies and whitepapers for the Enterprise Content Management products, and resources for optimizing, tuning, and troubleshooting performance

  - IBM Content Navigator 2.0.3 Knowledge Center
    - `http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3/contentnavigator_2.0.3.htm`
    - Planning, installing, and configuring IBM Content Navigator > Performance tuning for IBM Content Navigator
  - IBM FileNet P8 Platform 5.2.1 Knowledge Center
    - `http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm`
    - Administering > Performance tuning IBM FileNet P8 components
  - IBM Case Manager 5.2.1 Knowledge Center
    - `http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.1/com.ibm.casemgmttoc.doc/casemanager_5.2.1.htm`
    - Administering your case management solution > Monitoring system performance
    - Administering your case management solution > Tuning IBM Case Manager

More information

- A few examples of what you can find in the ECM Performance and Scalability Library

ECM performance and scalability studies

- IBM FileNet Content Manager 5.2 High Volume Scalability
  - A case study of a IBM Content Navigator / FileNet CM system with over 5 Billion documents running a workload of over 120K transactions per minute)
  - `http://www.ibm.com/support/docview.wss?uid=swg27043875`
- IBM FileNet CM 5.2.1 Content Engine Bulk Import Tool (CEBIT) Performance, Tuning, and Best Practices
  - `http://www.ibm.com/support/docview.wss?uid=swg27046510`

ECM performance monitoring, tuning, and troubleshooting resources

- Redbook: IBM FileNet Content Manager Implementation Best Practices and Recommendations
  - `http://www.redbooks.ibm.com/abstracts/sg24757.html`
- IBM System Dashboard for Enterprise Content Management
  - `http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC19-3084-02`
- Indexing for IBM FileNet P8 Content Engine Searches
  - `http://www.ibm.com/support/docview.wss?uid=swg21502886`
- How much memory do I need on my IBM Content Navigator server?
  - `http://www.ibm.com/support/docview.wss?uid=swg21679302`
- White-box Profiling for Optimizing WAN Performance of IBM Case Manager 5.2
  - `http://www.ibm.com/support/docview.wss?uid=swg27040542`

Other supporting performance resources

- IBM WebSphere Application Server Performance
  - The centralized WebSphere site with many helpful performance reports, articles, tools, and downloads
  - `http://www.ibm.com/software/webservers/appserv/was/performance.html`
- developerWorks community “DB2 for LUW Best Practices”
  - Whitepaper “Tuning and Monitoring Database System Performance” (one example whitepaper)
  - `http://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Wc9a068d7f6a6_4434_aece_0d297ea80ab1/page/Tuning%20and%20Monitoring%20Database%20System%20Performance`

End notes

Author: Dave Royer, ECM Performance Engineering Architect

Thanks to these contributors who helped improve this presentation

- Zhi Zhang, Zai Ming Lao, Chao Shen, Adrian Hermosillo, Michael Bordash, Nhan Hoang, and Ruairi Pidgeon from the ECM SVT/Performance team
- Kevin Trinh, the ECM SVT/HA Architect
- Brent Taylor, Steven Hsieh, and Brett Morris from the ECM Development team
- Ted Dullanty, Andrew Im, Danny Palar, Joe Krenek, Tom Garda, and Will Kilpatrick from the ECM Support team