



Lotusphere2011

IBM Software

BP117 - IBM Lotus Domino Server and Application Performance in the Real World

**Andrew Pollack, President
Northern Collaborative Technologies**

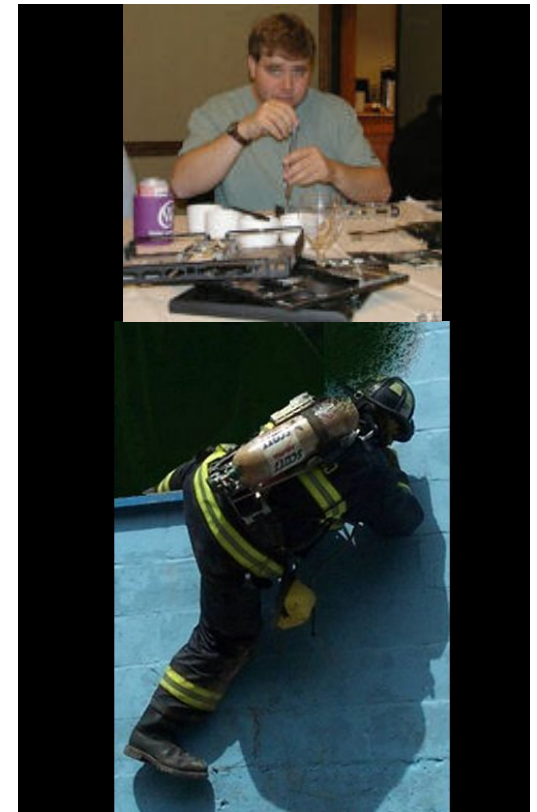
**andrewp@thenorth.com
<http://www.thenorth.com>**





Who Am I?

- Administrator & Developer since version 2.0
- IBM Lotus Beacon Award Winner
- Products
 - NCT Search
 - NCT Compliance Search
 - NCT Simple Sign On
- Services
 - Site Performance Reviews
 - Application Development
 - Administrative Overhaul
 - Security Review & Penetration Testing
- Structural Firefighter
 - Lieutenant of Cumberland, Maine – Engine 1





Please Shut Off All Noisemaking Appliances

- I'm not judging, just shut them off while we're talking
 - Unless they are medically necessary



**You can Perform
Better in the Server
Room!**

Ask your consultant how.....



Key Focus Points

- Performance with a Big Picture approach
- Defining Performance In User Terms
- Key Performance Choke Points
- General Considerations
- Common General Tweaks
- Make Your Web Site Faster!
- Developers, Developers, Developers
- Servers and SANS and VMs – oh My!
- Virtually Perfection
- When good INI settings go Bad!
- Finding Your Own Choke Points
- Summary



**Can Prevent
Performance
Problems**



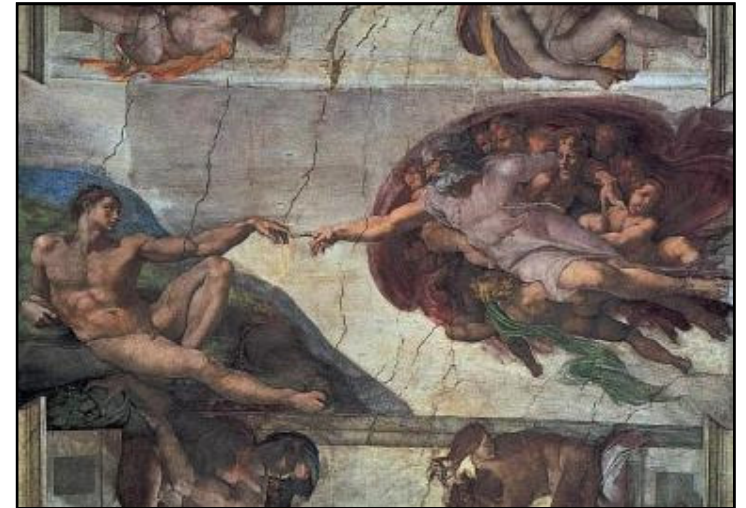
A face lift may make you look better for a while, but it won't cure cancer

PERFORMANCE WITH A BIG PICTURE APPROACH



Big Picture : There Is No Magic

- No Single INI Variable -- #1 Server Fix
- Focus On The Basics!
- No Super Storage Network
- No Ultimate Network Switch
- No Omnipotent Third Party Application
- No Über-Consultant
 - Not Even Me!





Big Picture: Small Issues Stack Up

- Performance Problems Are like snowflakes
 - Individually, they don't matter much at all
 - You notice them only once they stack up
- For example:
 - Poorly Performing Disk I/O
 - + Agents Changing Many Documents
 - + Many Views (or BAD views) to Update
 - == Very Slow System
- These kinds of problems create a feedback loop, which amplifies the problems





Beware the INI change advice – No matter how tempting!

- My Number One Server Crash Response
- INI Changes Come From
 - Well meaning tips
 - Low level tech-support
 - Too much time on public forums
- How to fix most server crashes
 - Clean out ALL non-default INI settings
 - Unless you can specifically document why it's critical
 - Clean out ALL non-shipping Code
 - Get rid of those fix-packs that didn't fix the problem
- Yes, there are some good changes to make to the INI file



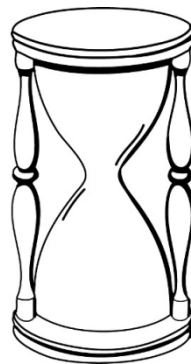
It's not how you feel, its how you look.
Darling, you look marvelous! -- Billy Crystal

DEFINING PERFORMANCE IN USER TERMS



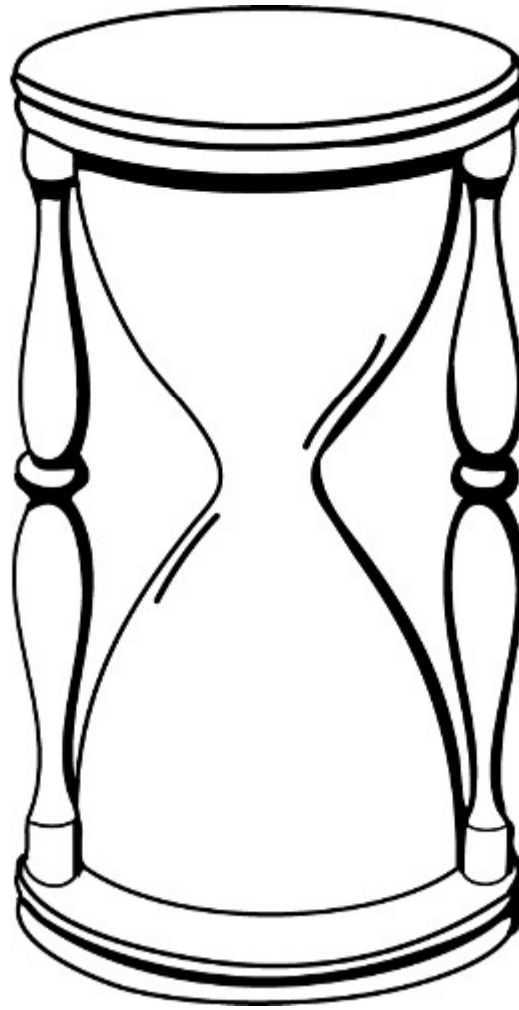
Performance in User Terms

- If the user must wait for something, it will always seem slow – no matter how fast you make it.
- Nothing is worse than an hourglass cursor and a bar slowly moving across the screen
 - Except NOT having the bar





...Except NOT having the bar





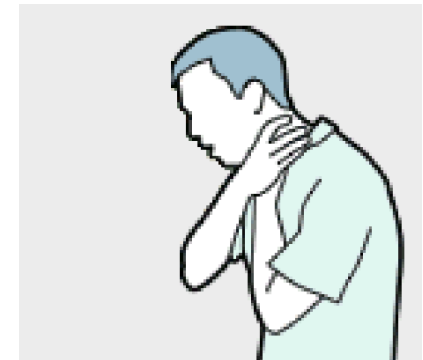
Performance in User Terms: Tips

- Move anything not immediately required by the user to a background process
 - Batch process updates of data that users do not need instantly
- Cache Commonly Referenced Data
 - How often do your common lookups change?
 - Country Names?
 - Escalation Levels?
 - Document Categories?
 - Lookup once when the database is opened, and store the values as environment variables locally
- Don't pop-up modal dialog boxes with no choices!



We're going the wrong way, but we're making excellent time!

KEY PERFORMANCE CHOKE POINTS





Choke Points: The Network

- Bandwidth vs. Latency
 - Bandwidth
 - How big around is the pipe?
 - Latency
 - How long is the pipe from end to end?
- Even light takes several minutes to reach us from the Sun. That's latency
- Latency impacts “Chatty” connections
 - Notes Database Open
 - Multiple Views Lookups
 - AJAX on Web Applications



Where does Latency Come from?

- Ping times larger than 100ms are “high” latency.
- WAN links, Satellite links, Modems, and VPN’s are all prone to latency issues
- Multi-Hop connections across buffered routers and firewalls can introduce latency
- Encryption software can introduce latency



Dealing with High Latency

- Avoid opening and closing many documents
- Avoid DB Lookups by caching common values
 - Example: Use a db open script to write common lookup values to a local environment variable each time the user opens the database
- Use “RunOnServer” to move complex agent work to the server, then read the result from a profile document
- Consider JSON instead of AJAX
- Stop using “NoCache” on your DBLookups



Choke Points: Disk I/O

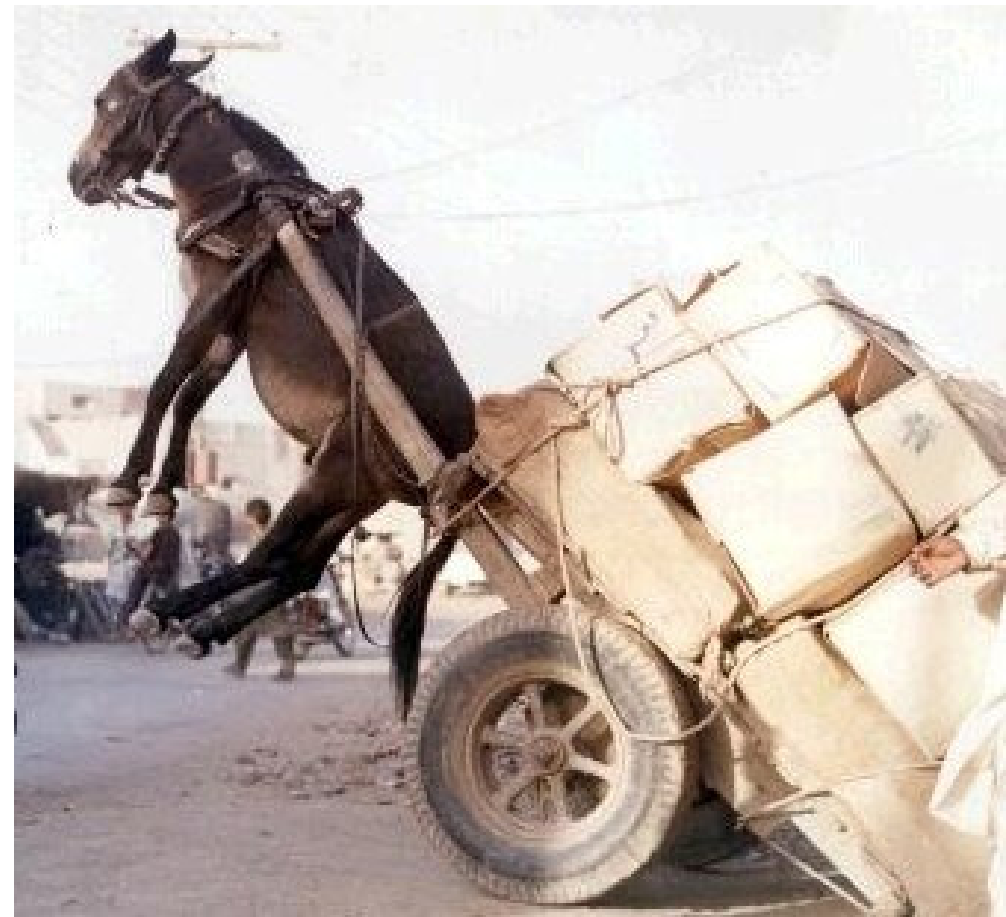
- This is the #1, #2, and #3 Root performance problem on Domino Server
- Nearly any other performance problem is made many times worse if the Disk I/O is overwhelmed
- Most Domino Servers are not well optimized for Disk I/O





Common Sources of Disk Performance Problems

- One “Data” drive is used for too much
 - databases, index rebuilds, temporary files, swap files, and even transaction logging
- Transaction Logging used in conjunction with journaling file systems
- Poor choice of RAID configurations
- Too heavy a reliance on Storage Area Networks





Poor choice of RAID configurations

- RAID is not ALWAYS the best performance choice
- Some Common Types of RAID
 - RAID 0
 - Increase performance, Decrease Reliability (x number of drives)
 - RAID 1
 - Increase Reliability, No Performance Difference
 - RAID 5 (The Most Common) – Uses 3 or more drives
 - Balance of redundancy plus some performance gain
 - RAID 1+0 (aka RAID 10)
 - Two pairs of RAID1 read as a RAID0 (Hybrid)
 - RAID 0+1
 - Two pairs of RAID0 written as RAID1 (Hybrid)



Why RAID isn't ALWAYS the best way?

- Competition for Resources
 - An overall performance gain with RAID 5 of 30% (typically) is spread across all the disk I/O on the server.
- You are smarter than RAID
 - You can put highly intensive resources on specific drives or arrays, balancing the load more effectively
- Multiple RAID Arrays are not always possible
 - Expensive
 - Multiple Drive Bays
 - Power Hungry Drives



Disk I/O: Rethinking RAID

- Most RAID arrays are configured to improve redundancy, not necessarily speed.
- Not all data requires redundancy
 - Loss of some data is very low risk
 - Memory Swap Files
 - Indexing scratch space
 - Temporary files
 - Cache files
- Inexpensive SATA drives can be used for a real performance gain
- Solid State Drives – Possible future use for very high speed, relatively small footprint data, like transaction logging
 - More on SSD....



Too heavy a reliance on Storage Area Networks

- The SAN is not Evil – But it isn't the perfect solution to all problems
 - High Speed, but High Latency
- A SAN is a Compromise
 - Trades the speed and simplicity of local drives for enterprise manageability and flexibility
- Good for Backup Data
- Good for Big, Sequential Files
 - Media Files
 - Installation Kits
 - Archival Data
- Choke point for active database work
 - * More on SANs shortly....



If you do use a SAN

- Where can you compromise?
 - Cheap local drives for low-risk use
 - Memory Swap File
 - Temporary Scratch Space for View Rebuilds
 - Web Server Cache Files
 - Log Files
 - Domino Directory
 - Keep one replica backed up on the SAN
- If you can't compromise at all – that's fine
 - But you'll need more servers



Virtualization and Domino

- Domino runs just fine in VMWARE
 - I use VMWARE esxi for my office & development environment
 - (Some of my best friends are virtual servers)
- Performance issues are VERY similar to SANs
 - Disk I/O is again critical to Domino performance
 - Virtual environments often share disk resources
 - Virtual environments often utilize SANs
- Follow the guidelines for using Domino on a SAN
 - Local, dedicated storage spindles wherever possible



Domino with a SAN

- Consider the benefits of a SAN
 - Highly redundant storage
 - Single backup point
 - Consolidated free space
 - Performance?
 - I have yet to see a SAN that truly outperforms local high speed disks
- Not all Domino Data needs these features
 - Transaction Logs – Consider local RAID if possible
 - Indexing Scratch Space – Use Cheap, Local, Fast Drives
- If you're already clustering Domino, only one of the clustered machines may need to be on the SAN



Disk I/O: WARNING

- Many of the following recommendations balance performance with safety
- You need to assess each as it relates to your overall data strategy





Disk I/O: Use Multiple Drives

- Put your transaction logging files on a separate drive
- Move your view indexing temporary files to another drive
- Consider moving disk-intensive applications to their own drive
- If you must have memory swapping, give it its own drive
- Active Log Files for Web Servers, SMTP, etc.



Disk I/O: Not Everything Needs Its own Drive

- Things that load once and are not re-accessed frequently do not need to be on high performance resources
- The Operating System
- Application Program Files
- Archived Log Files



Disk I/O: Think in terms of “Spindles” or “Data Paths”

- One Disk may have multiple partitions
 - Different partitions are NOT different spindles
 - All the partitions on the same drive, share the same read-write head and are impacted by data access as a single entity.
- Multiple drives in a RAID array don't count
 - A RAID array is treated by the system as a single drive. By definition, data is written across the whole array
- The “Best-Case” is multiple drives on different drive controllers



Solid State Drives (SSD)

- Also known as “FLASH” drives
- Getting more common on Laptops, Netbooks
- Reliability Issues are Largely Resolved
- VERY Fast READ Times
- Write Performance Quickly Degrades
 - This is changing quickly, but still the case for most uses
 - Windows 7 & Windows Server 2008 R2 Support “TRIM”
 - <http://en.wikipedia.org/wiki/TRIM>
- Good for Program Files, Java Libraries
- Bad for NSF Databases, Indexing, Translogs



Disk I/O: Journaling File Systems

- Windows NTFS – And you can't turn it off!
- Linux ext3 file system
- IBM AIX, SUN Solaris, and Apple OSX all make use of Journaling File Systems
- Not all the same – but generally speaking, disk WRITES are doubled
- Transaction Logs on Journaling File Systems are redundant redundant



Journaling: How Safe is Too Safe?

- RAID Configuration
 - Data is written twice (at least)
- Formatted with a JFS
 - Data is written twice
- Using Transaction Logging
 - Data is written twice
- $2 \times 2 \times 2 = 8$ Times the Data Writes
- Now think about that on a pair of clustered servers



Choke Points: System Resources

- These should be obvious
 - More RAM is better – Up to what is supported
 - Depending on the OS, you may need to partition your server to take full advantage
 - Drive Cache – If your OS lets you manage it, you should work to really optimize this
- Most Anti-Virus Software is EVIL when it runs against Domino Databases
 - Make sure your AV is Domino aware!



Faster faster! The lights are turning red...

MAKE YOUR WEBSITE FASTER!



Let the browser cache common items

- Resources that don't change frequently can be cached

- JPG
- PNG
- GIF
- MOV
- MP3
- MSI
- MPG
- ZIP
- EXE

Web Site Rule

Basics | Comments | Administration

Basics

Description:	Cache JPG		
Type of rule:	HTTP response headers		
Incoming URL pattern:	*.jpg*		
HTTP response codes:	200, 206		
Expires header:	<input type="radio"/> Don't add header <input type="radio"/> Add header only if application did not <input checked="" type="radio"/> Always add header (override application's header)		
	<input checked="" type="radio"/> Specify as number of days <input type="radio"/> Specify as date Expires after 30 days		
Custom headers:	Name:	Value:	<input type="checkbox"/> Override
	Name:	Value:	<input type="checkbox"/> Override
	Name:	Value:	<input type="checkbox"/> Override



Developers really LOVE when administrators give them feedback

APPLICATION DESIGN STRATEGIES



Choke Points: Views

- For application performance tuning, views are the first, second, and third place to look
- View indexing is very disk intensive – and can amplify disk I/O shortcomings
- To update a view, a full database scan often needs to happen. That can be very very slow on large databases
- Any view performance problem grows exponentially with the volume of data
 - These problems are often not caught in test



What Kills View Performance

WHEN GOOD VIEWS GO BAD



Bad View Design: Too Much Data

- Switch @Responses to @AllDescendants
 - NO visible difference to users
 - Can reduce view sizes drastically
- Can You Set a CUTOFF date?
 - Form = "Request" & @Modified < [01/01/2008]
 - Hardcode The Date
 - Change it by AGENT, Warning in DB Script if out of date
- How about a CUTOFF data for MOST of the views, with just one or two for "Archival" data?



Switch @Responses to @AllDescendants

- NO visible difference to users
- Can reduce view sizes drastically
 - View #2 is 153 Times the Size of #1 and has the EXACT same content

Manage the views of this database

Use this tool to manage the views of this database.

Selected: Test View Index Count.nsf, 4 MB bytes

The view indexes of this database consume 17 MB of disk space, which is 410% of the entire space used by this database.

View name	Size	Owner	Refresh	Discard	NoteID
Test 1 Using AllDescendants	42,192	Andrew Pollack/thenorth	Automatic	If inactive for 45 da	0x10A
Test 1 Using Responses	6,055,892	Andrew Pollack/thenorth	Automatic	If inactive for 45 da	0x186
Test 2 Using AllDescendants	6,138,836	Andrew Pollack/thenorth	Automatic	If inactive for 45 da	0x18A
Test 2 Using Responses	6,055,892	Andrew Pollack/thenorth	Automatic	If inactive for 45 da	0x18E

Purge



Bad View Design: Too Much Sorting

- EACH Sorted Column can as much as DOUBLE the size of the total view index
- Many views have all the columns sorted
 - Even the ones that end up sorted because of order
- Multiple Column Click-To-Sort Views Can be WORSE than multiple views!
- Many SHARED columns are sorted
 - Developers Assume No Downside



Bad View Design: Too Much Data

- Does the SUBJECT really need to be in every view?
- Can you create one “Master” view with all the data, and several “Index” views with an Action Button to open the master view?



Bad View Design: Using TIME Values

- If your view column (bad) or selection formula (worse) uses @Now, @Today, etc.. You're hurting performance
- Time dependant views are "Always" considered out of date and must be re-indexed for every use
- If you've got one, you've got more. Developers that do this tend to repeat the pattern



Alternatives to Time Specific Views

- Use a FOLDER instead
 - Run a agent to select the right documents for the folder on a periodic basis – Daily for “@Today” or Hourly for @Hour(@Now), etc.
 - This will still cause an update, but only once each time the update happens
- Use Categories
 - Categorize documents based on a stored date value, then use a “show single category” option on the view
- If you MUST use a time specific view, set its update frequency to the absolute least frequent you can
 - It will still update for each user access, however, unlike a folder which is static



Bad View Design: Highly Complex Formulas

- Consider a column formula with 10 steps
- Now consider 100,000 Documents in it
- That column must execute 1 Million steps for each view index rebuild – just in that column
- Many column formulas are much more complex, and serve many times that many documents



Alternatives to Complex view formulas

- Create Hidden Fields on the Document
- At “Save” time, compute the value that would be on the view column in the hidden fields
- Display the value of the hidden field as the view column formula.
- What was a complex formula executing hundreds of thousands of times is now a single field value



Bad View Design: Too Many Views

- Consider a database with 100,000 documents
- Consider that database having 10 views
- Consider each view having 5 columns
- Each time data in the database is updated, every selection formula has to be checked to see if the view is impacted
- Every view has to be updated by the indexer



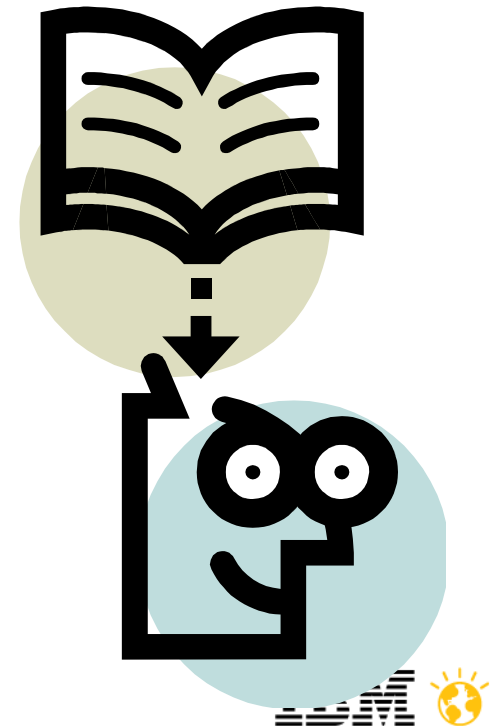
Alternative to Using Many Views

- Embed The View on a Form or Frameset
 - Categorize the view in the same way you would distinguish the different views
 - Use Show Single Category to differentiate the data to the user
 - Compute text values on the form to result in very different data in each category if needed
- Use multi-column hidden views so that the same view can serve multiple lookup needs
 - Make sure your developers coordinate so that duplicate lookup views are not created



Story Time!

- Andrew's View Size story
- 2 GB of View Data reduced by 90%
- without loss of application function





Full Text Search: The Good, The Bad, and the Ugly

- The Good
 - You can use it in agents instead of `db.search`
 - `Db.ftsearch()` has a rich syntax and can be much faster
 - Its lets users find things – of course
- The Bad
 - Usually set to update “immedately”
 - Agents that change many documents can cause a massive amount of disk I/O at the worst possible time
- The Ugly
 - Be careful using it as a way to gather documents in code, as it may not be up to date



Choke Points: @DBLookup

- Why are you using “NoCache”?
 - Cache times are very small, does your data really change on a second by second basis?
- Can be very chatty – a killer on high latency networks, but not as bad for web apps
- Requires more views to be up to date – big performance hit in databases that change a lot
- Many lookups on the same form, to the same place for different values?
 - Use it once to get the UNID, then use @GetDocValue
- Use a profile document, or local environment variables updated in the dbopen script to store commonly looked up data



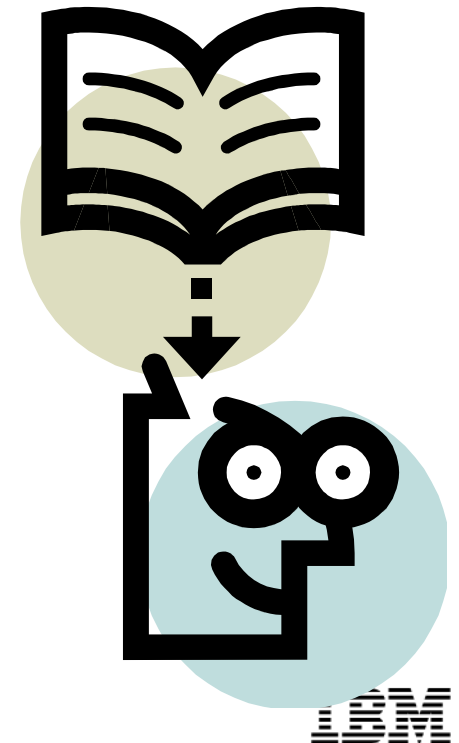
Agents are better with Hash!

- Calculate and Store a HASH value
 - A HASH is a short, nearly unique, value created by applying a mathematical formula to a set of data. For example, you can hash an entire paragraph and get a short string as a hash value. The same source will always produce the same hash, but any change to the source will produce a different hash.
 - You can tell if a document has changed, simply by comparing the HASH value



Story Time!

- Andrew's Magic Hash story
- If you fall asleep, please don't drool on the table
- C'mon, it's a true story!





Other Agent Tricks

- Read View Entries – Not Documents
- Turn off MIME conversion when working with mail documents
 - `NotesSession.ConvertMime=False`
- Run agents periodically, not “Before New Mail Arrives” – that slows down the router



Where to place the blame

FINDING YOUR OWN CHOKE POINTS



Where to look for performance problems

- Look for Disk Performance First
 - Start Simple: Are the drive lights sitting on for long periods of time?
 - Use the operating system's tools
 - Performance monitor & diskmon in Windows, “top” in Linux, etc.
 - Processes like “logasio” which is for transaction logging will show up
- Check for network latency and bandwidth
 - Start Simple: Use Ping to check latency



You really came here looking for cool INI settings like DominoRunFaster=11

INI SETTING TWEAKS



Some NOTES.INI tweaks

- COMMENT NOTES.INI Changes!
- Here's some that I use
 - MailLeaveSessionsOpen=1
 - For busy mail servers, can speed up delivery
 - Update_Fulltext_Thread=1
 - Move full text indexing to its own thread, distinct from the indexer – This is the closest to “runfaster” I have found
 - Ftg_use_sys_memory=1
 - Use memory outside the Domino server
 - HttpQueueMethod=2
 - According to Kerr, this is a must have for busy web server



A few more notes.ini tweaks

- Use These Together:
 - SERVER_NAME_LOOKUP_NO_UPDATE=1
 - Tells the server to use the old index while the new one catches up
 - DEBUG_ENABLE_UPDATE_FIX=8191
 - Fine tunes when the directory indexes get refreshed



And of course... NSF_Buffer_Pool_Size_MB

- NSF_Buffer_Pool_Size_MB=
 - Very powerful, but very complex
 - Check the Lotus Notes Knowledge base
 - Starts at around 300
- Not as critical as it used to be
 - Documentation Says it is now set AUTOMATICALLY for non-partitioned Servers
 - My Testing Says it is also now set AUTOMATICALLY even for partitioned Servers in 8.5.x
- Check your success with this console command
 - `show stat database.database.b*`
 - Don't check too soon after a change, its only valid over time



Notes 8 Client Tweak

- Open this folder:
- {NotesProgramDirectory} \framework \rcp \eclipse \plugins \com.ibm.rcp.j2se.{Version}
 - THIS LOCATION CHANGES in 8.5.1 to
 - {NotesProgramDirectory} \framework \rcp \deploy
- Edit the file: jvm.properties
- Change the line: vmarg.Xmx=-Xmx256m
- So that it reads: vmarg.Xmx=-Xmx512m
- Note: You can set it higher, but aim for about half of your available RAM
- Readers on my blog overwhelmingly report fantastic results with this one



Summary

- Repeat After me:

There is No “RUN_FASTER=1”

I will clean up my NOTES.INI

I will COMMENT my NOTES.INI changes

- Performance Isn't Magic, its Planning
- Save the Disk I/O, Save the Server
- Latency is as critical as Bandwidth
- When in doubt, Blame the developer



Questions?

- Ask now, don't wait for the end and ask quietly at the podium
- The most up to date copy of this presentation will be on my blog site: <http://www.thenorth.com/apblog>
- Andrew Pollack – Northern Collaborative Technologies
 - andrewp@thenorth.com
 - <http://www.TheNorth.com>





Legal Disclaimer

© IBM Corporation 2011. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

All references to any company refer to a fictitious company and are used for illustration purposes only.