Lotusphere
Connect2013
Get social. Do business.

BP106 IBM Lotus Domino RunFaster=1

Daniel Nashed | CTO Nash!Com, Germany

About the presenter

- Daniel Nashed
  - Nash!Com - IBM/Lotus Advanced Business Partner/ISV
  - Member of The Penumbra group
    - an international consortium of selected Business Partners pooling their talent and resources
  - focused on Cross-Platform C-API, Domino® Infrastructure, Administration, Integration, Troubleshooting and IBM Notes Traveler
  - Platform Focus: Windows, xLinux, AIX® and Solaris®
  - Author of the Domino Start Script for Linux and Unix
- nsh@nashcom.de
- http://www.nashcom.de

Agenda

- Introduction/Overview
- Experience from the Field
  - Hardware / OS
  - Domino Infrastructure
  - Database Settings
  - Applications
  - Client
- Q & A

Domino Performance only a Notes Team issue?

- No!
- This is a task for
  - People who take care about hardware
  - Operation System Support team
  - Network team
  - SAN Admins, ...
  - **Domino Administrators**
  - **Domino Developers**
  - People who install the Notes Clients and Workstation software (OS …)

  - In most cases we are called by the Notes team and end up speaking with a number of different IT teams within a company

A Holistic Approach to Performance Management

- You need to take care about all levels
  - **Hardware**
    - Disk Drives, Memory, CPU
  - **Operating Systems**
    - Parameters, Subsystems, …
  - **Network**
  - **Virtualization Layer**
  - **Notes Infrastructure**
    - Server, Server settings, …
    - Client, Client settings
  - **Databases & Database Settings**
  - **Applications**
    - Use the right techniques designing & coding your applications
  - **Similar to security, the weakest component will define your performance!**

Hardware

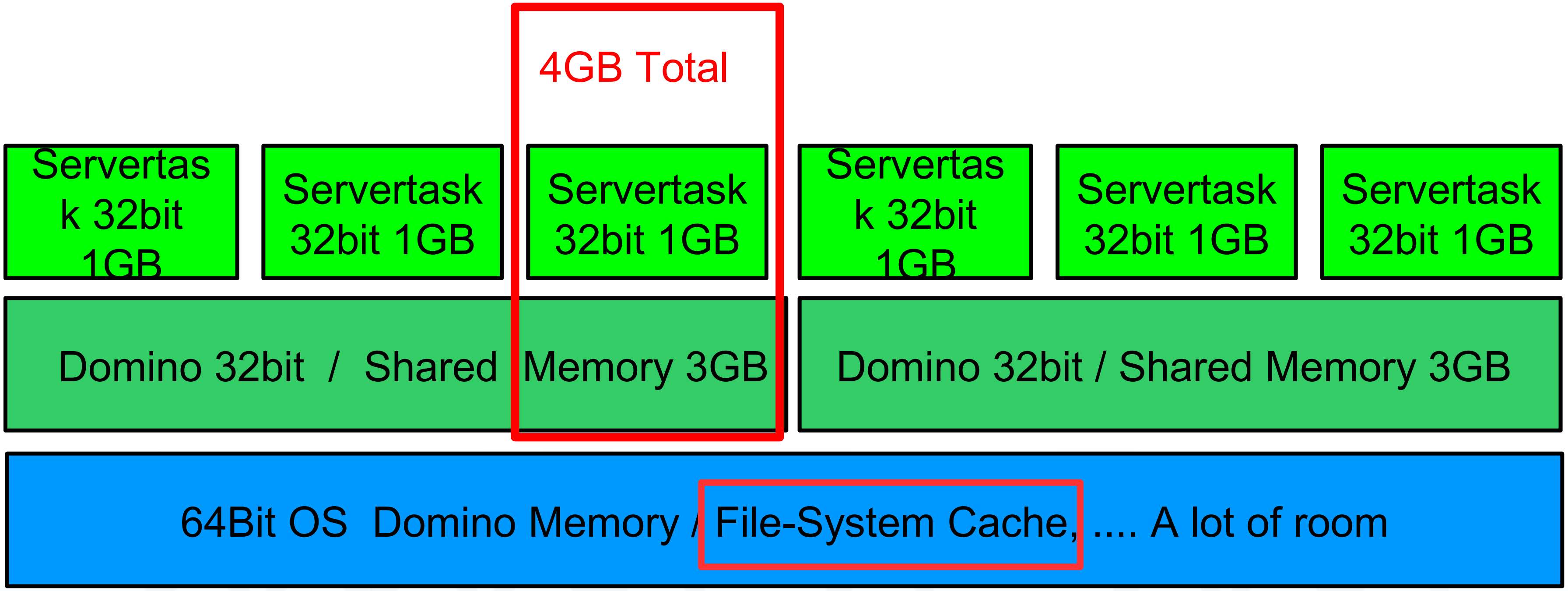
- “Intel®” / x64 Hardware is cost effective
- Usually a single quad core will fit most of your requirements
  - Take care about licensing by CPU & CPU Type!
  - If you are in the CEO model you are save for Notes/Domino
    - Still have to check for add-on software like Tivoli Data Protection
    - In some cases license for data volume fits better than by CPU
- RAM is relatively cheap
  - For larger servers you might want to use at least **16 GB of RAM**
  - Some for server types like partitioned servers or special application servers even **32 GB**
  - Domino 8.5.x (even 64bit) will not use more than 4 GB RAM in most environments
  - But more RAM on a 64Bit OS will be used for file-system cache → reduces read file I/O
- Network
  - A single 1 GB card should be fine
    - You might want a second card for hardware failover and load-balancing
    - Sometimes dedicated backup network makes sense

64Bit Windows and & 64Bit Linux

- You should always use a 64bit OS for **all** type of Domino servers
  - 32bit Windows is limited to 2GB Application Space and 300 MB of File-System Cache
- 64bit Windows and Linux allows a 32bit application to use the full 32bit 4 GB address space
  - You don't necessarily update your Domino Server to native 64bit itself.
    Most benefits come from the 64bit OS in **Domino 8.5.x**
  - This will change with **Domino 9** with 64bit exploitation features!
    Most important internal pools can be increased -- check session **ID107** for details
  - So you might already plan for even more RAM for newer servers
- File-System Cache now uses all free RAM also on Windows
  - Dramatic reduction of Disk Read I/O depending on your configuration and data
  - More data is cached in the OS file-system cache and is not re-read from the disk
- We have seen application servers with 32GB of RAM holding most of the used NSF in memory with almost zero read I/O during normal operations
  - Can be more cost effective than using expensive enterprise level SSD Drives

Domino 32bit on a 64Bit Operating System

- Total Memory per Process is 32Bit = 4 GB
- Router / HTTP uses most local process memory
- NSF Buffer Pool is the biggest Shared Memory block (512 MB)



Transaction Logging

- Recommended for all Domino server types !!!
  - Changes the way databases are locked for concurrent access
  - “Lock-Manager” optimizes performance
  - Changes are written sequentially into translog
  - Asynchronous Log Manager (“logasio”) writes data into databases afterwards
  - Process can continue to run meanwhile

- Without Transaction Logging, databases open at crash time are inconsistent
  - Needs fixup for all open databases which causes load on server and delays the time until Domino Server is completely online after a crash
  - **Data Loss** possible without Transaction Log in crash situations

- Highly Recommend: Domino Backup API aware Backup Solution
  - Or shutdown your Domino Server for backup at night
  - All other backup operations are completely **unsupported**

Disk Configuration

- Different Type of options depending on your strategy: Local Disks, SAN/NAS
  - But still similar configuration even in virtual environments
- Strong Requirement:
  - “Separate Disks” for
    - OS/Swap/Binaries/OS Log etc
    - Domino Data (NSF)
    - Translog (TXN)
    - DAOS (NLO)

  - Optional if needed separate disk for Fulltext Index (FT)
    - New Option since 8.5.3 to store FT outside the data-dir
    - notes.ini **FTBasePath=c:\FTBasePath**

- “**Separate Disk**” can mean different things in different environments
  - **Local Disk** = Separate RAID volume
  - **SAN** = Different Logical Unit (LUN)
  - **NAS** = Has it's own file-system and depends on sizing and capability of the NAS
  - **Vmware ESX** = separate VMDKs – even if they resist in the same disk volume

Disc Configuration

- **Translog** needs fast sequential write
  - **Ensure that write cache is enabled!**
  - Should be separated from “data” but the only key requirement is low latency write
    - Usually a separate RAID1, separate VMDK etc
- **Domino Data**
  - **RAID10** instead of RAID5.
  - High End SAN hardware should be always RAID10
  - Performance mostly depends on
    - Cache, Number of disks, Performance of each disk
  - Domino Data needs fast/small random I/O
  - **Average I/O Rate: 0,5 IOPS per registered user**
- **DAOS**
  - Needs lower performance and has larger sequential I/O requests
  - Could be RAID5 or NAS (even with UNC Path)
    - But with UNC path you need sub-directory on NAS side!!!

OS Level Antivirus

- Disable Antivirus for the following file types
  - ***.nsf, *.ntf, *.box**
  - ***.ncf, *.ndk**
  - ***.TXN**
  - ***.DTF**
  - ***.nlo**
- Not excluding those files could lead to performance issues and even crashes or hangs
- If you need Antivirus scans on Notes databases you should use a Domino aware Antivirus solution
  - Mail Scans on Gateway & Scheduled Scans on Databases
- TIP: Review exclusion lists after <u>every</u> antivirus software (not pattern) update!

Windows Tuning

- There are not many tuning parameters required
  - Most important: Keep fixpacks up to date!
    - In case of a SAN, check for Fibre-channel card best practices and driver updates!
    - In case of NAS/iSCSI, check configuration best practices for network card with your NAS vendor
- Fixed page file (min and max values should be the same)
  - Have separate file-system for paging file
- Disable Paging Executive
  - In order to increase performance, kernel mode drivers and other system components can be configured to that they are not paged to disk.
  - HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management → “DisablePagingExecutive”=dword:00000001

Linux File-System Tuning

- Use your favorite journaled file-system ext3, etx4, Reiser FS, …
  - RHEL 6.x has Ext4 support (by default) / SLES 11 SP2 does not support Ext4 yet
- Disable write of meta information via mount option **-noatime**
  - **Meta information like last accessed time**
- A real Runfaster=1 Parameter:
  - Change the default scheduler from **CFQ** (complete fair queuing) to **NOOP**
  - CFQ tries to optimize disk access by reordering requests
    - But it would be better to send it to a SAN, RAID controller directly
    - Tests have shown that this works better for almost all SAN or local disk configurations
    - Dramatical improvement!
    - See next slides for details
  - Disable per device
    - **echo noop > /sys/block/hda/queue/scheduler**
  - Disable globally via kernel boot parameter
    - Edit **/boot/grub/grub.conf** and enter in kernel line **elevator=noop**.

Linux Performance CFQ vs noop

- Read-Test - 80 thread to read 32000 docs each
  - 80 separate local databases on the server with small documents
- Result:
  - 51 sec with CFQ scheduler
  - 28 sec with noop scheduler
  - 19 sec all data in cache

- Write Test - 80 threads creating 2000 docs each
  - 80 separate local databases on the server
- Result:
  - 132 sec with CFQ
  - 42 sec with noop

- Environment: SLES11 SP2 with local RAID10 disks
- Test-Tool: iostat -x 2 → check the improvement in the “**await**” column

OS Level Performance Measurement

- Domino Platform Statistics are your friend
  - Can be used for longer term monitoring via collect task into statrep.nsf
  - But they are only updated every 60 seconds and collected usually every 10 minutes
  - Not all information is included – for example “await” on Linux (disk queue response time)

- For troubleshooting you should leverage OS level statistics

- Windows: **Perfmon** – “Average Disk Queue Len” < 1
  - **TIP New in Win2008: User Resource Monitor to check response time and I/O**

- Linux: iostat -x – “await” below 10 ms

- Examples next slides

Platform Stats - Disk

- Platform Stats Windows
  - AvgQueueLen should be below 1

```
Platform.LogicalDisk.1.AssignedName = HarddiskVolume1
Platform.LogicalDisk.1.AvgQueueLen = 0,2
Platform.LogicalDisk.1.AvgQueueLen.Avg = 0,8
Platform.LogicalDisk.1.BytesReadPerSec = 0
Platform.LogicalDisk.1.BytesWrittenPerSec = 0
Platform.LogicalDisk.1.PctUtil = 11,00
Platform.LogicalDisk.1.PctUtil.Avg = 50,00
Platform.LogicalDisk.1.ReadsPerSec = 0
Platform.LogicalDisk.1.WritesPerSec = 0
```

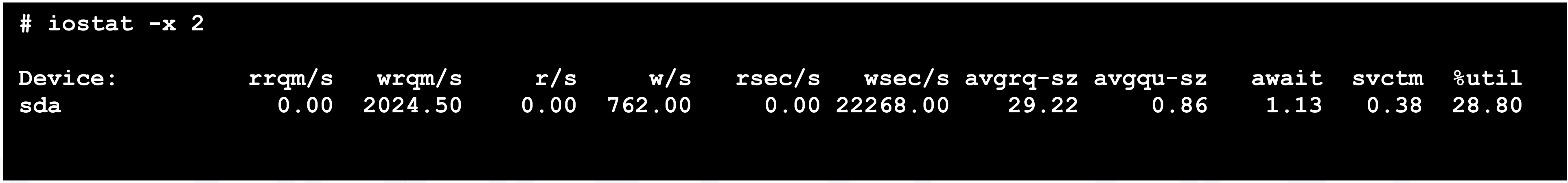
- Platform Stats Linux
  - Service Time is just an indicator – better value would be “await”

```
Platform.LogicalDisk.1.AssignedName = sda
Platform.LogicalDisk.1.AvgQueLen = 11.89
Platform.LogicalDisk.1.AvgQueLen.Avg = 11.89
Platform.LogicalDisk.1.PctUtil = 95.63
Platform.LogicalDisk.1.PctUtil.Avg = 95.63
Platform.LogicalDisk.1.PctUtil.Peak = 95.63
Platform.LogicalDisk.1.ServiceTimeinmsecs = 8.35
Platform.LogicalDisk.1.ServiceTimeinmsecs.Avg = 8.35
```

Linux Native I/O Stats “iostat”

- iostat provides more detailed information about the current I/O statistics
  - Output by device (e.g. sda = first disk)
  - Example shows just one value for one disk
  - TIP: You can filter via grep – e.g. “iostat -x 2 |grep sda”

- **%util** =Disk Utilisation in % → Values above 90% are an indicator for a busy disk
- **r/s** = Disk reads per second
- **w/s** = Disk writes per second
- **svctm** = Disk services time in ms (how fast the device responds)
- **await** = Time the whole request needs (application to disk queue, disk and back)
  - This is the most important statistic and key indicator



```
# iostat -x 2

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00  2024.50    0.00  762.00     0.00 22268.00    29.22     0.86    1.13   0.38  28.80
```

VMware ESX Virtualization

- Generally a good idea for all smaller servers, test servers, HUB, SMTP, Traveler, servers
- For larger servers with more than 500-800 users all components have to play well together and you should plan for sufficient resources
  - Domino can be very resource demanding specially in the I/O area
  - Supported even for larger servers since ESX became a “tier-1 virtualization platform”
- Have separate VMDKs for all parts of the server as mentioned before
  - OS, NSF, Translog, DAOS, ..
  - **Take care where VMDKs are located on the SAN side and who you share your storage and servers with!**
- Plan sufficient Virtual CPUs
- TIP: Start with smaller servers and implement performance monitoring!
  - ESX Servers have embedded statistic modules
  - But you will need access to the ESX Infrastructure Client

Run Current Software!!!

- Run up to date Operating Systems
  - New Operating Systems usually have better performance!
  - **Windows 2008 R2 – SP2!!**
  - **RHEL 6.3**
    - Kernel 2.6.32-x
  - **SLES 11 SP2**
    - Kernel 3.x
    - Already a 3.x kernel. Introduced in SP2
    - There is a random I/O optimization fix in 2.6.33 – not yet in RHEL 6.3
- Run up to date Domino Versions and Clients
  - There are performance fixes in almost any release
  - See some details included in my slides

SLES 11 / RHEL 6.x Performance Issue / RunFaster=1

- SPR# PHEY8RJHXR fixed in 8.5.3 FP2
  - Fixes a timing issue with the CFQ process scheduler introduced in
  - Caused response time issues with transactions – up to 100 ms delay
  - This fix changes semaphore handling for the server thread pool

  - In SLES 11 SP2 use the following in addition to improve performance
    - **echo NO_FAIR_SLEEPERS > /sys/kernel/debug/sched_features**

  - In earlier relases (SLES 11 SP1 + RHEL 6.x) with CFQ you might want to use
    - **echo “0” > /proc/sys/kernel/sched_features**

  - And you need 8.5.3 FP2 or higher to fix the issue
  - This will fix the 100 ms delays completely
  - Performance Test - 30 threads / 100 documents / attachment size 2 MB

	no network	without fix	fix + fair sleeper	fix + no fair sleeper
elapsed time (sec)	56	828	78	68
response time client (ms)	2	40-60	5	3

Network

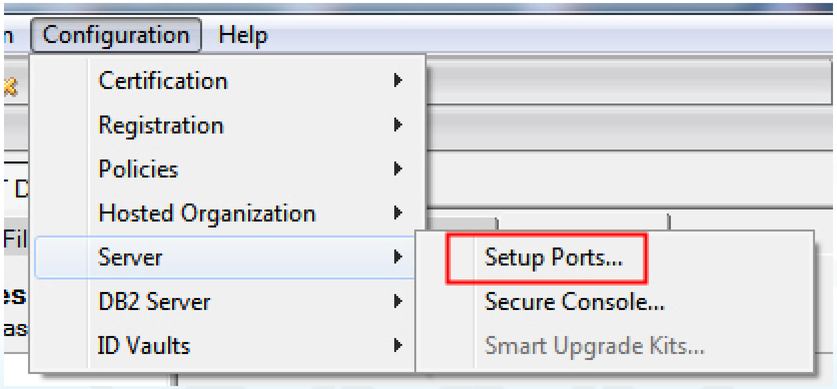
- Different Networks have different latency
  - LAN = around 1 ms
  - WAN in same country depending on network around 50 ms
  - WAN in Asia or other remote locations up to 400 ms

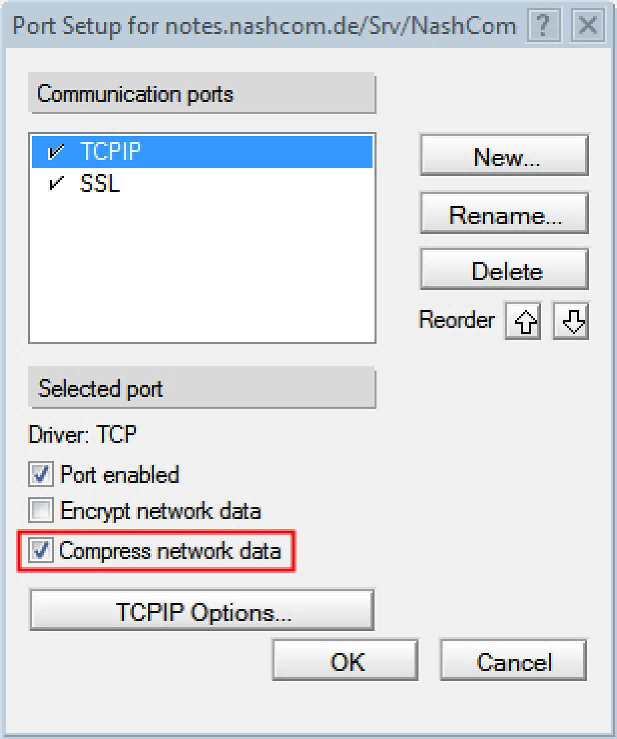
- So the speed alone does not matter if you have many small transactions
  - Network Accelerators can help to some extend specially for re-transmissions etc
  - But can only reduce the latency to a small extend
  - For complex applications already 50 ms make it difficult to work
  - Example: 100 NoteOpen Transactions
    - in LAN = 100 ms
    - in WAN is already 5 seconds

  - Multiple work-arounds:
    - Optimize Application
    - Optimize Network
    - User different access method like Web or Citrix
    - Offload operations to the server (e.g. agent run on server)

Network Optimization

- If you have no active network component that optimizes your traffic or if you use Notes port encryption, enable Notes Port Compression!!
  - This can reduce the traffic by 50% (improves response time but does not reduce network latency)
- Network Compression needs to be enabled on **<u>both</u>** ends! Client **<u>and</u>** Server!
  - Client restart or Notes Server Port restart required (e.g. restart port tcpip)
  - TIP: Can be deployed with Client DesktopPolicy
- Also reduces traffic between Servers!
  - Specially a good option for Traveler Servers




Port Setup for notes.nashcom.de/Srv/NashCom

Communication ports

✓ TCPIP
✓ SSL

Selected port

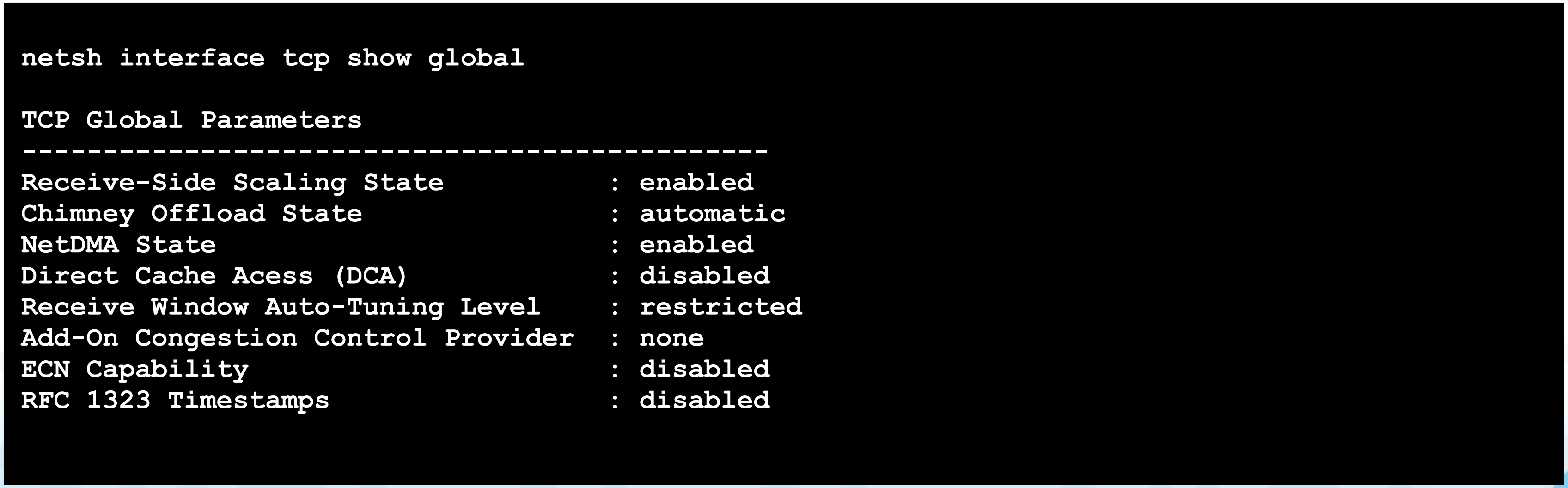
Driver: TCP

☑ Port enabled
☐ Encrypt network data
☑ Compress network data

Windows 7 / Win2008 Network Tuning

- New Networking Stack in Windows 7 (also Visa) and Win2008 Server
- Some optimization can cause issues with Notes/Domino
  - The default for auto tuning options changed between service packs
  - We have seen quite bad performance in some customer situations

- You might want to change the autotuning level from “normal” to “restricted”
  - Set → **netsh interface tcp set global autotuninglevel=restricted**
  - Check → netsh interface tcp show global



```
netsh interface tcp show global

TCP Global Parameters
----------------------------------------------
Receive-Side Scaling State           : enabled
Chimney Offload State                : automatic
NetDMA State                         : enabled
Direct Cache Acess (DCA)             : disabled
Receive Window Auto-Tuning Level     : restricted
Add-On Congestion Control Provider   : none
ECN Capability                       : disabled
RFC 1323 Timestamps                  : disabled
```

Windows 2008 Port Stealth Mode Cluster Failover Issue

- From IBM Technote #1498755

- Windows 2008 has a feature called "Stealth mode" wherein by default it does not respond to network requests on ports for which no process is listening. This feature creates a problem with the Notes client failover when a server is down, since the OS does not respond to the client and thus the client will not failover until it's timeout has been exceeded. Although Domino does support Windows 2008, at the appropriate version level, this is a feature which "breaks" the Notes client failover.

- To run in "Stealth Mode" and get good failover, IBM recommends the installation of Firewall software which disables this "feature" on the Notes NRPC port (1352 by default).

- Optionally, the following link gives directions by Microsoft on how to disable "stealth mode"
  - http://msdn.microsoft.com/en-us/library/ff720058%28v=prot.10%29.aspx

Cluster Failover Load Issue

- Timing/Semaphore issues with the transactions that finds replicas
  - **This issue can cause failover and failback to almost completely hang server transactions for a couple of minutes**
  - That means if one server is down the other server will not handle the load
  - Good news this is fixed and the default has been switch on in 8.5.3

- SPR# SWAS8GGHMC
  - Fix makes cluster failover more efficient in **8.5.3 by default**. This fix was also in 8.5.2 FP3 but you had to set a notes.ini ENABLE_CLUSTER_MATCHES_FAST=1 in the Domino server notes.ini to enable it.
  - There is also a notes.ini to disable that is only valid in 8.5.3 and above which is DISABLE_CLUSTER_MATCHES_FAST=1 (default value being 0).
  - The existing API - ServerGetClusterReplicaMatches resolved the failover replica IDs but caused server outages.
  - Regression in 8.5.2 FP2.

View Rebuild Directory

- By default Domino uses a memory optimized way to rebuild views
- You can specify a directory (e.g. on system disk) for optimized view rebuild
- On Linux you can put temp-files and view rebuild files into tempfs
- **Tempfs is a kind of virtual file-system optimized for temporary files that exist only for a short period of time**
  - Similar to a RAM drive but not reserving memory and self organized
  - Changes are only written to disk if memory is needed by the server
    - It would swap to disk if space is needed
  - tempfs is enabled by default with half the size of the physical memory
    - Located in /dev/shm
- You can also use tempfs for other temporary files

NSF Buffer Pool

- There have been a lot of discussions about NSF Buffer Pool
  - The old rules to have 3/8 of the addressable memory is not helpful any more
  - Forget about parameters like ~~PercentAvailSysResources~~
  - The NSF Buffer Pool aka UBM is used for database I/O
- By default the NSF Buffer Pool is 512 MB for 32bit and 1 GB for 64bit
  - The default works for almost all configurations
  - A cache/buffer is only effective until a certain size
- Check Server Statistics
  - **Database.Database.BufferPool.PercentReadsInBuffer**
- Interpretation: Bad < 90% < PercentReadsInBuffer < 98% < Perfect
  - If you really need to change it use notes.ini **NSF_BUFFER_POOL_SIZE_MB=512**

Database Cache

- Used for caching open databases
  - You should have sufficient entries for all databases than can be open on your server!
  - Number of users / Number of databases whatever is higher
- –Default: NSF_BUFFER Pool size multiplied by 3
  - That's where NSF Buffer Pool comes into play

- Check Server Stats
  - Database.DbCache.HighWaterMark
  - Database.DbCache.CurrentEntries
  - Database.DbCache.MaxEntries
  - **Database.DbCache.OvercrowdingRejections**

- Interpretation
  - Good = HighWaterMark < MaxEntries
  - Good = 0 OvercrowdingRejections

- Tune: e.g. notes.ini **NSF_DbCache_MaxEntries = 3000**

Concurrent Tasks

- Concurrent Threads serving user transactions

- Check Server Stats
  - Server.ConcurrentTasks
  - **Server.ConcurrentTasks.Waiting**

- Interpretation
  - Waiting should be ZERO

- Tune
  - Server_Pool_Tasks = n (e.g. 80)
  - **Server_Max_Concurrent_Trans** = m (e. g. Server_Pool_Tasks * Number of Ports)

  - **notes.ini Server_Pool_Tasks=100**, **Server_Max_Concurrent_Trans=100**

Important Server Settings

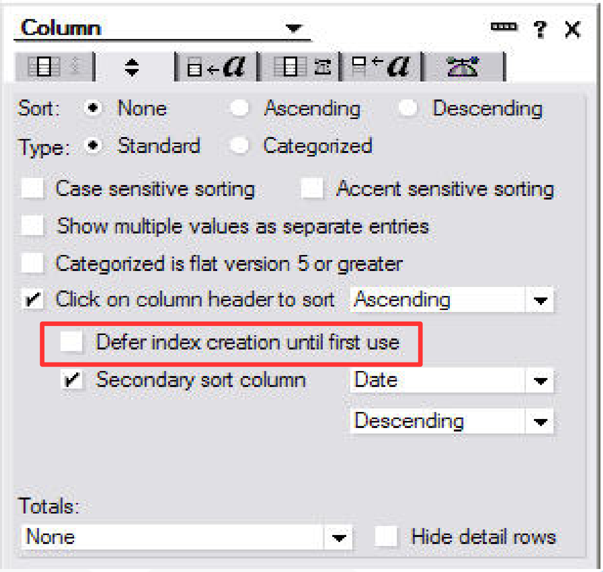
- Increase Shared Memory Maximum Size for Domino 32bit on 64bit OS to 3 GB
  - **ConstrainedSHMSizeMB=3072**
- Default for many pools is too small
  - **EVENT_POOL_SIZE=41943040**
  - **CATALOG_POOL_SIZE_MB=100**
  - **dirman_poolsize_mb=100**
  - **nsf_monitor_pool_size_mb=200**
- Separate Update FT Thread and have FT use system memory
  - **UPDATE_FULLTEXT_THREAD=1**
  - **FTG_USE_SYS_MEMORY=1**
- Router Optimization
  - **RouterMaxConcurrentDeliverySize=1048576**
  - **Disable_BCC_group_expansion=1**

NSF Optimization

- Use the current ODS on Clients and Servers
  - **Create_R85_Databases=1** (of course)
  - TIP: Use client desktop policies to upgrade ODS on Notes Clients!
- Enable Data and Design Compression
  - **Compact -C -v -n**
- Take benefit of DAOS!
  - Reduces disk I/O, disk space, backup time (incremental backup), increases scalability
  - Put DAOS on different disk
- Rebuild Views after moving to new ODS and mailfile-design
  - **Updall -r**
  - Will optimize view collections (only first sort collation will be build)

On demand Collations Best Practices

- Views/Folders often have multiple sort columns with on demand sorting
- Since Domino 8 you can "Defer index creation until first use" under the "Click on column header to sort" in the properties of the column of a view/folder
  - This will ensure that only the primary index of the view/folder is created
  - If the user clicks on another sort index, the index will be automatically build
  - And updated via update task from now on!
- Design needs to have this option enabled
  - Notes 8.x design and higher is already optimized
  - But you need a updall -R to reduce the space
- Tip: Very good hidden in the Designer Help
  - All on-demand sort indices have a life span subject to the view's discard interval. They can also be explicitly removed by issuing this command on the server console:
    load updall <dbname> -r -g



Column

Sort: ● None ○ Ascending ○ Descending

Type: ● Standard ○ Categorized

□ Case sensitive sorting □ Accent sensitive sorting

□ Show multiple values as separate entries

□ Categorized is flat version 5 or greater

☑ Click on column header to sort: Ascending

□ Defer index creation until first use

☑ Secondary sort column: Date

Descending

Totals: None

□ Hide detail rows

Database Performance Tips

- Large numbers of Deletion Stubs can make databases slow
  - It's not only the number of documents
  - Count like normal documents for some operations
  - Use "**Show database dbname.nsf**" to get all live and deleted documents (and design elements!)
- Avoid document update (specially in import agents)
  - But it is still a lot cheaper to update than to delete and add!
  - Delete and add causes a lot of deletion stubs
- Use FT Index with care
  - Can block updaters from indexing other databases
  - Think about having them updated hourly or daily
  - TIP: Separate FT Index Thread → **notes.ini UPDATE_FULLTEXT_THREAD=1**

Search Queries in Agents

- Simple agents often use search queries for document selection
  - If the database has no full-text index an **on-the-fly temporary** FT index is created for every agent invocation!!!

- Avoid on-the-fly fulltext indexing needed by agents with search queries and no FT index on the database
  - Set **notes.ini FT_FLY_INDEX_OFF=1** to disable on-the-fly FT indexing completely
  - Will let agents fail with error which need on-the-fly FT indexing

- The warning message you see in log.nsf for on-the-fly FT index is:
  - **"Warning: Agent is performing full text operations on database '<name>' which is not full text indexed. This is extremely inefficient."**

Database Limits

- Number of documents in a heavy used (updated) database
  - < 50.000 – 60.000 documents
- Almost static database
  - >> 100.000 documents
- Split your application into static config database and databases with normal “data”
- Use archiving to have smaller active and larger static archive databases
  - Sounds simple but is very effective
  - Improves performance and reduces I/O load dramatically
  - Specially true for mail!

Server based Archiving

- Without quotas and even with DAOS, mail grows exponentially
  - Doubles every 12 to 18 month! So maybe just keep 12 to 18 month in your active database?
- Split data into
  - a.) Smaller active, fast changing databases
  - b.) Larger static databases
- Domino 8.5.x Server based Archiving works great
- You could consider using it even if you just archive to the same server
- Use a separate disk for your mail archive
- Use Archiving Policies to implement Archiving
  - Very straight forward and easy to handle
- Most expensive element is the $Inbox folder for a large, busy mail-database!

View Update Performance

- Keep view column formulas and number of columns simple
  - Tip: Calculate parts of complex view formulas in documents
  - Too many categories also make indexing slow
- Never use time-based views (@Now, @Today, ...)
  - View design note has item "**$FormulaTV**" – that's the safest way to spot a time-based view
  - Those views are rebuild **every time** they are opened
  - Workaround: @TextToTime ("Today") and refresh index manually
  - Other idea: Agents putting documents into folders every night or sets fields for selection
- View update performance is linear to number of docs and also number of columns
  - But complex view performance changes exponential
    - 2 times more documents -> takes ~4 times longer
- Reader Fields take much more performance
  - Use small number of reader-fields and entries and keep
  - Proper design shows just a fraction of documents per view

Disable Unread-Marks if possible

- Unreadmarks need to be maintained per user
  - Will be updated when the user opens a database
  - If you have many changes in the database since last open it will take some time!
- But this is not the only reason to disable unread marks...
- Some operations take much longer if unreadmarks are enabled
- One critical issue:
  - Even if you have concurrent web-agents enabled, only one agent will run at a time because unread marks need to sync after each agent run!
- Strongly recommended: Disable Unreadmarks if possible
- TIP: You should disable unreadmarks for all system databases if you don't need it
  - names.nsf, log.nsf, mail.box, admin4.nsf, statrep.nsf

Advanced DB Properties(Last Tab)

- **ON: Don't maintain unread marks**
- OFF: Replicate unread marks: Never, Cluster servers only, All servers
- ON: Optimize document table map – specially when using form = “..”
- **ON: Don't overwrite free space**
- **OFF: Maintain LastAccessed property**

- Maybe ON: Don't support special response hierarchy
  - needed for @AllChildren and @AllDescendants

- **ON: Use LZ1 compression for attachments**
- ON: Allow more fields in database (compact -K / -k)
- ON: Allow soft deletions

- Maybe set to low value: Limit entries in $UpdatedBy / $Revision fields

Profile Documents

- Persistent storage of configuration/preference information
  - Profile-Documents are "sort of" a normal document but access works different
  - Profile-Documents are cached for performance
  - You should not use profiles to store data that permanently changes
    - Specially not for counters etc!!!
- Limit the number of profile documents
  - e.g. have only one per user per application
- profileDoc.userName(0) vs. profileDoc.**GetItemValue**(“userName”)
  - GetItemValue is faster because the code does not know what "userName" is and has too loop thru multiple different kind of elements

Agent Document Processing

- Unprocessed documents in agents can help if used properly
  - **Set documents to processed even you don't work on them!**
  - Use unprocessed documents for performance only
  - You still need a status field else if the agent looses it's run-data (scheduled on a different server or updated) document might be reprocessed

- Example from the field:
  - 100 instances of workflow application with 1.000-30.000 docs,
    Agents where scheduled between 5-30 minutes. Only few changes per hour. Only 20% of docs needed processing.
  - Developer forgot to set other 80% of docs to processed
    - Result → Huge workload on a larger machine. The 80% of documents needed to be checked over and over again

- Extra TIP: If not using “.**UnprocessedDocuments**”, set Agent target to "**none**"
  - The default is “unprocessed documents”

ComputeWithForm Issue

- “ComputeWithForm” does not use cache.ndk
  - It even does not leverage the newer finddesign transaction but does a lookup in the design collection
  - So every ComputeWithForm does read all forms, sub-forms, shared fields which are involved …
  - A lot of overhead. Maybe OK for a single ComputeWithForm but refreshing multiple documents can be painful
- Workaround: Use Run on Server Agents off-loading the work to the server
  - This can be a good approach for other operations that need many transactions

Private on First Use Views/Folders

- You should use Private on First Use Views/Folders with care
- If you have more than 40 design elements with the same name the internal design cache breaks.
  - A fix would require an ODS change – see Technote #1322578, SPR #RSTN7K2EM4
- When the design cache is broken fall-back code is used to find design elements
  - Instead **FINDDESIGN_NOTES** transactions the design collection is opened and searched sequentially until a match is found!
    - For databases with a lot of data this can be quite an overhead
- Workaround: User Single Category Views, Folders/Views for certain groups
- How to find out if the design collection is broken?
  - Free Nash!Com Tool (just send me an email) to check a single database for the design cache flag
  - Alternate way: Check for design collection usage via client_clock tracing
    - Client_Clock=1, debug_outfile=c:\notes.log

Client_Clock Tracing

- How did we find those issues?
- Can be used to track Notes Client/Server Transactions (NRPC)
  - Enable on Client via notes.ini
    - client_clock=1
    - debug_outfile=c:\debug_notes.log (debug text file)
    - debug_console=1 (debug console – don't close during run-time, client will crash!)
- Client_Clock logs
  - Transaction sequence
  - Transaction name
  - Transaction data (ReplicaID, NoteID)
  - Response time (ms)
  - Bytes send, received
- Example:
  - (15-78 [15]) OPEN_NOTE(REPC1256B16:0072BCBE-NT00000E3E,00400020): 0 ms. [52+1454=1506]

Client Clock Annotation

- Manual Annotation is a lot of work
- Use free annotation tool OpenNTF Client_Clock Parser
  - http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/Notes%20RPC%20Parser
- Can help to understand your application foot-print
- See next slides for a reference of most important transactions

Reference - Selected Transaction Types

START_SERVER	Start User Session
OPEN_DB/CLOSE_DB	Open/Close a database
OPEN_NOTE/NIF_OPEN_NOTE	Open a Note
UPDATE_NOTE	Update a Note – there is no close transaction
OPEN_COLLECTION/ CLOSE_COLLECTION	Open/Close a view/folder collection
READ_ENTRIES	Reads data from a view/folder
UPDATE_COLLECTION	Updates a view/folder collection
FIND_BY_KEY	Finds notes in a view/folder collection
FINDDESIGN_NOTES	Finds design notes
SEARCH	Search operation with formula
GET_MODIFIED_NOTES	Find table of modified notes
GET_ALLFOLDERCHANGES_RQST	Get changes in all folders Unread count in mail folders

Reference - Selected Transaction Types

NAME_LOOKUP	Lookup information in Domino Directory
ALLOC_UPDATE_OBJECT	Create or update an object
READ_OBJECT	Read data from an object
WRITE_OBJECT	Write data to an object
READ REPLICATION HISTORY	Read Replication History
DB_INFO_GET	Get database info buffer
GET_NOTE_INFO	Get Note information
DB_MODIFIED_TIME	Get the modified date of DB
DB_REPLINFO_GET	Get Replication info
POLL_DEL_SEQNUM	Get Delivery Sequence Number
DB_GETSET_DEL_SEQNUM	Get or set Delivery Sequence Number
GET_SPECIAL_NOTE_ID	Get a special Note-ID by number

Client Performance Tips

- 64bit Windows has better performance also on client side
  - Consider Win7 64bit!
  - In Win32 the file-system cache is limited to 300 MB
- A word about hardware
  - You should have at least 2 GB of RAM
  - Even better would be 4 GB RAM
  - More current Dual Core CPU at least
    - Have a separate graphics card! No combined CPU/Graphics chips
    - We have seen client startup performance issues due to screen rendering issues with slow performing CPUs & Graphic sub-systems
  - You don't need necessarily a SSD but you should have decent, current hard-disks
    - 7200 RPM vs. 5400 RPM disks, larger cache
    - TIP: Hybrid disks are not really expensive any more

Client Performance Tips

- Startup Performance
  - Every Dot-Release has better performance for the standard client
  - Keep up to date!
- Don't use extmgr_addins from Antivirus Vendors like McAffee and Norton
  - Experience from the field: they can cause stability issues and also cost performance
  - Also see reference https://kc.mcafee.com/corporate/index?page=content&id=KB57589
- If you can disable on demand scanning for Notes programm files
  - A lot of small files for the Ecplise client!
  - Write checks and regular full scans should be fine
- Ensure program directory is not fragmented
  - Tip: Use free Microsoft defrag tool “contig 1.7” after installation on programm directory
    - http://technet.microsoft.com/en-us/sysinternals/bb897428.aspx

TIP - High Resolution Timer

- Getthreadinfo(LSI_THREAD_TICKS)
  - LSI_THREAD_TICKS = 6
  - LSI_THREAD_TICKS_PER_SEC = 7 returns ticks per second (1000)
  - tick = millisecond (different than agent ticks!)
  - Can be used to calculate run-time in milliseconds ;-)
- Code Example
  - Dim lStart,lStop As Long
  - lStart = Getthreadinfo(6)
  - ... do some work ...
  - lStop = Getthreadinfo(6)
  - ms = lStop – lStart
- Create profiling class, create new object in each of your subs
  - timing recorded in constructor/destructor

Linuxfest Returns!

Back for another informative all-inclusive Linux session in 2013

Join Bill Malchisky, Wes Morgan, and guest Daniel Nashed!

When: Thursday, 31 January

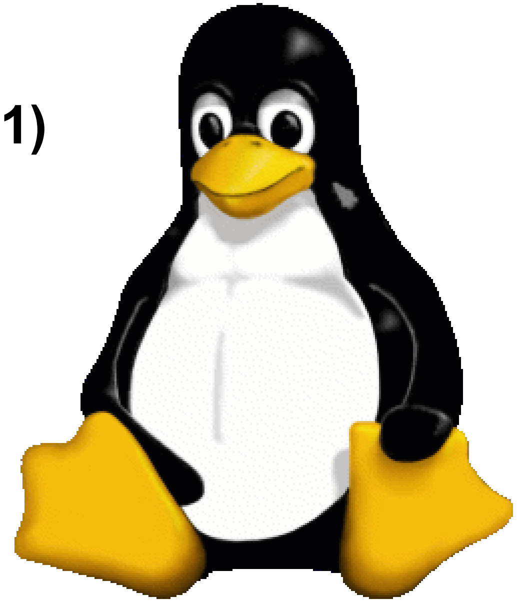
Where: **Dolphin Hotel - Sum Chow's (Next to Picabu, Level 1)**

Time: 12:15 - 1:30 pm

Other: Bring your box lunch!

We're not in the program guide, so mark your calendar, or

See our listing in the ConnectOsphere agenda Notes app



Special thanks to Red Hat for providing our session swag!

Q&A

- Thanks for your attention!
  - Please fill out your evaluations!
- Questions?
  - Now, find me later at the conference or contact me offline
- Contact
  - nsh@nashcom.de
  - http://www.nashcom.de
  - http://blog.nashcom.de
  - +49 172 2141912

Legal disclaimer

IBM Corporation 2013. 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.

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.