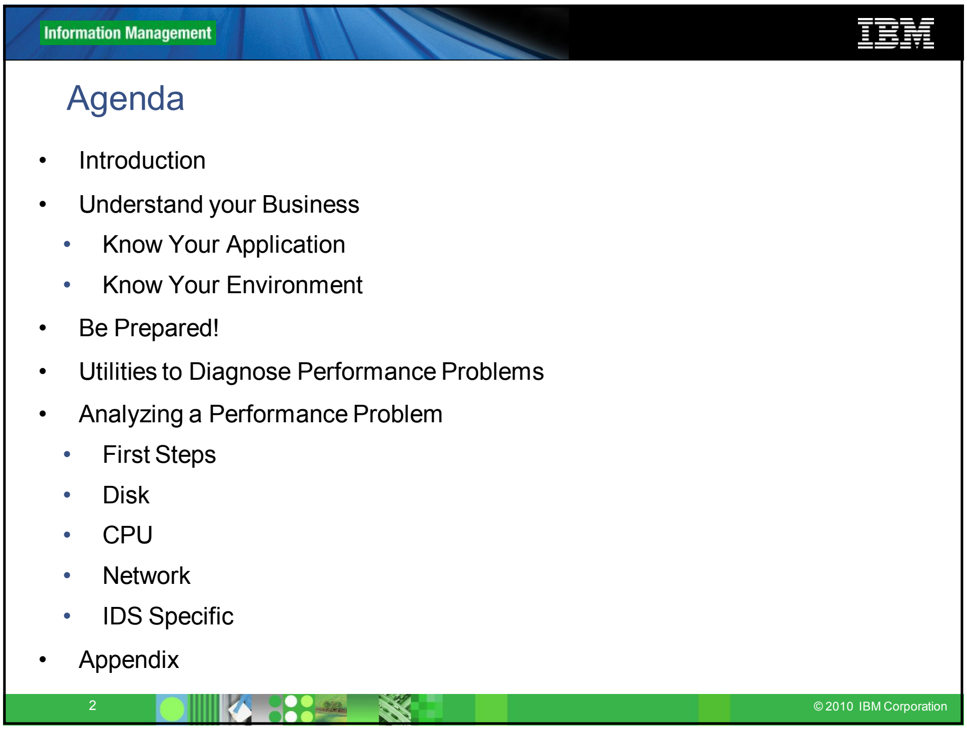
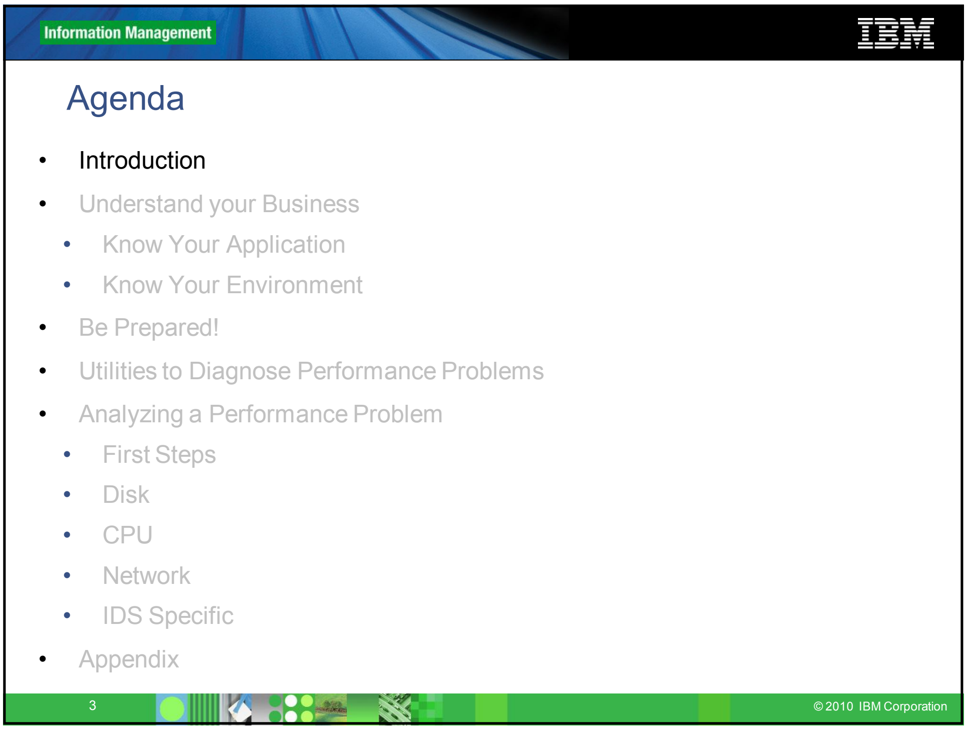
Informix Dynamic Server 11.5
IDS Performance Tuning & Troubleshooting Tools and Techniques

Paul Chang
paulyc@tw.ibm.com



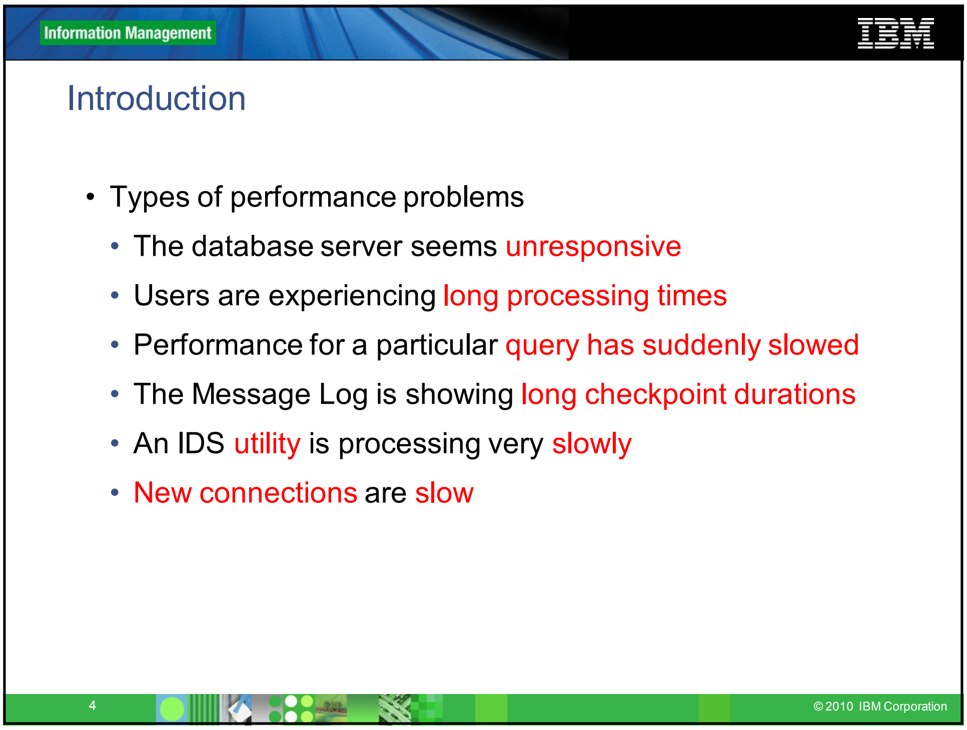
Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



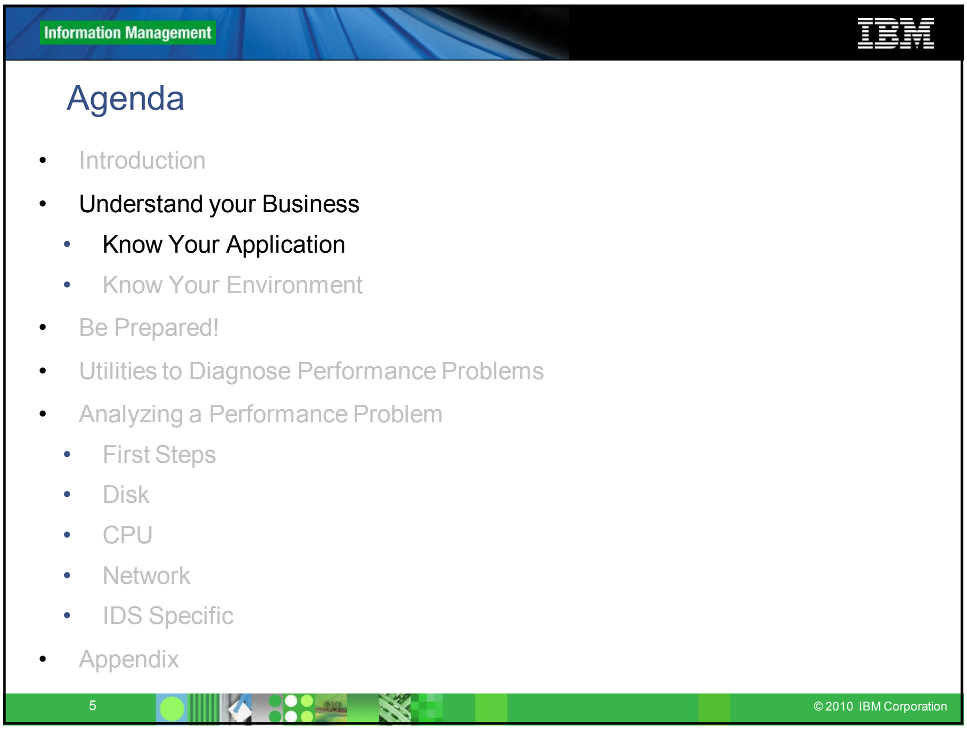
Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



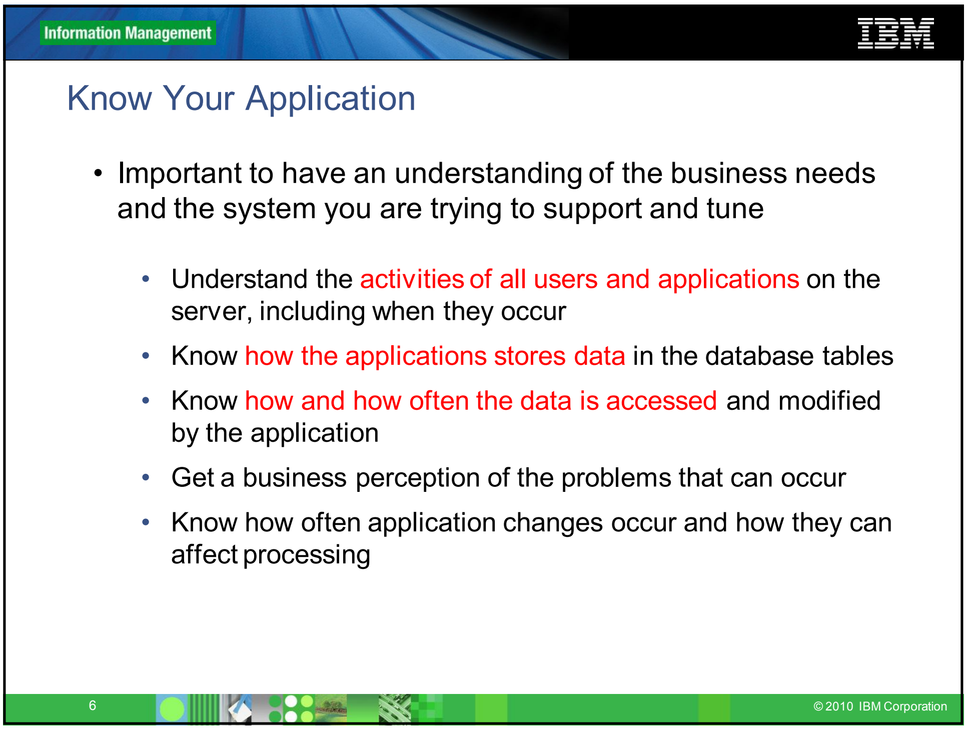
Introduction

- Types of performance problems
  - The database server seems unresponsive
  - Users are experiencing long processing times
  - Performance for a particular query has suddenly slowed
  - The Message Log is showing long checkpoint durations
  - An IDS utility is processing very slowly
  - New connections are slow



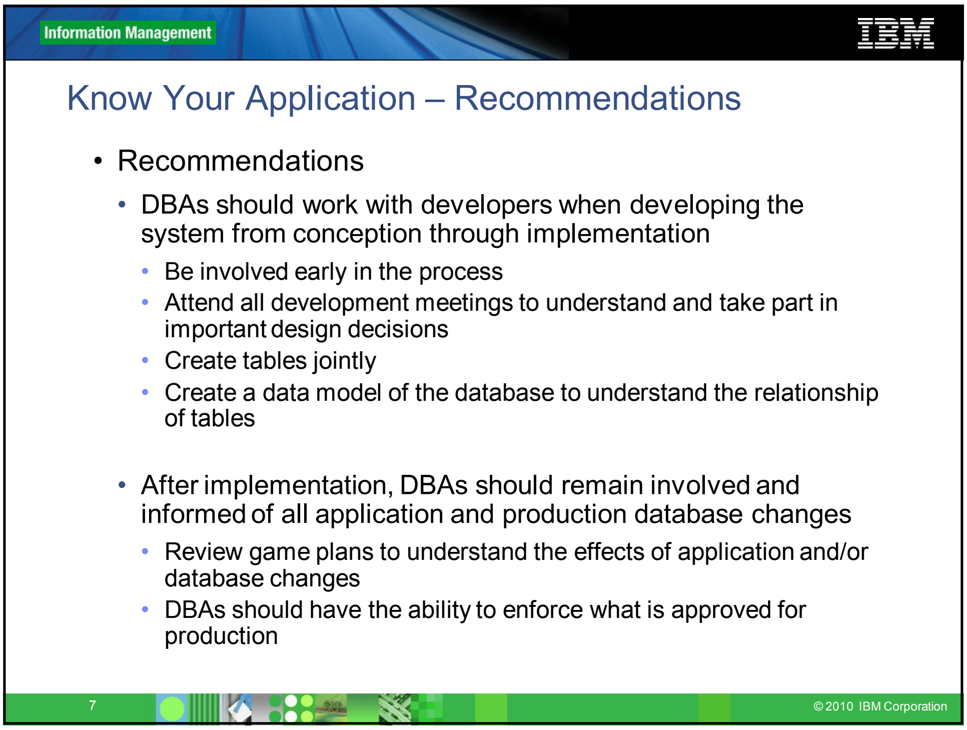
Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



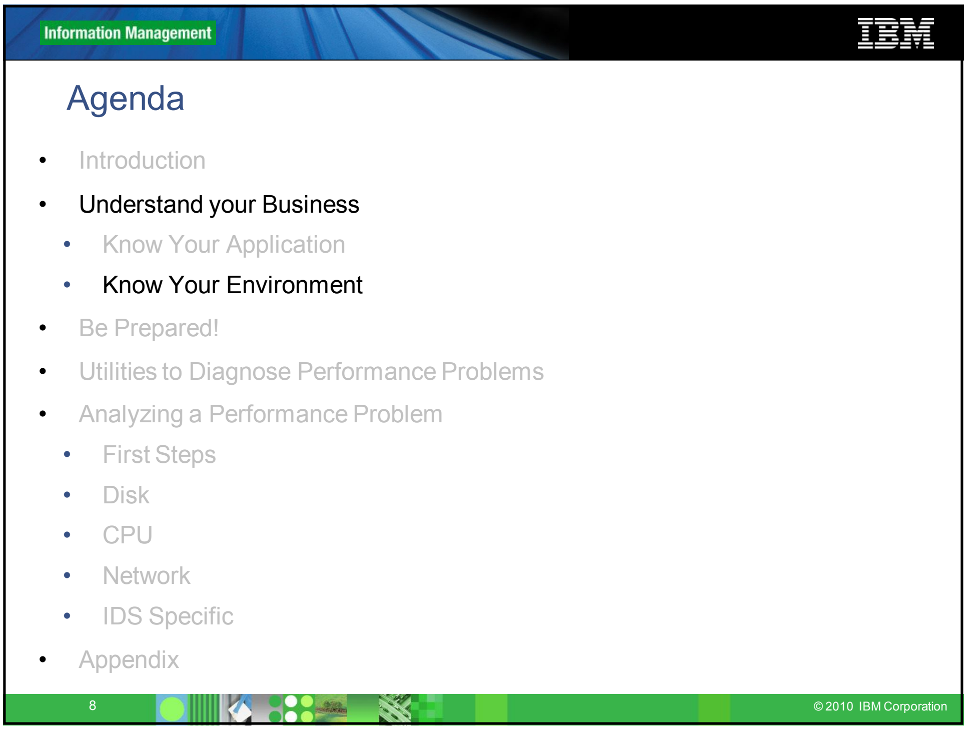
Know Your Application

- Important to have an understanding of the business needs and the system you are trying to support and tune
  - Understand the activities of all users and applications on the server, including when they occur
  - Know how the applications stores data in the database tables
  - Know how and how often the data is accessed and modified by the application
  - Get a business perception of the problems that can occur
  - Know how often application changes occur and how they can affect processing



Know Your Application – Recommendations

- Recommendations
  - DBAs should work with developers when developing the system from conception through implementation
    - Be involved early in the process
    - Attend all development meetings to understand and take part in important design decisions
    - Create tables jointly
    - Create a data model of the database to understand the relationship of tables
  - After implementation, DBAs should remain involved and informed of all application and production database changes
    - Review game plans to understand the effects of application and/or database changes
    - DBAs should have the ability to enforce what is approved for production



Agenda

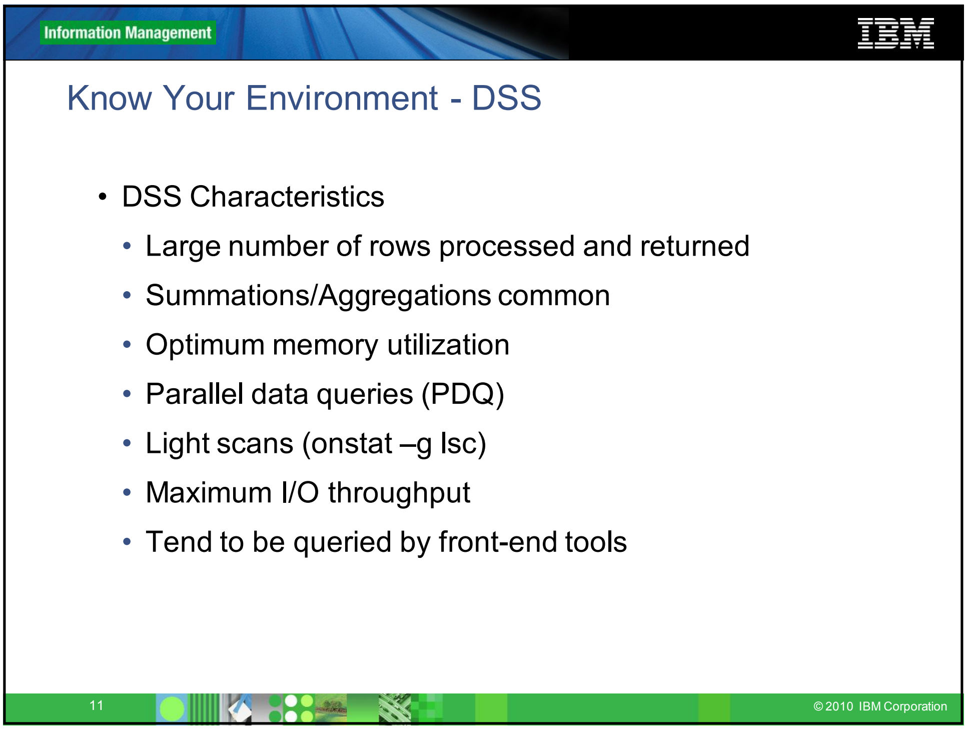
- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix

Know Your Environment - OLTP

- OLTP Characteristics
  - Small number of rows returned
  - Quick response times of queries
  - High read and write buffer cache rates
  - Short checkpoint durations
  - Maximum I/O throughput
    - Eliminating I/O bottlenecks
    - Optimizing index utilization
    - Optimizing fragmentation strategy
  - Known / limited “windows” for periodic tasks
  - Short “fast recovery” times

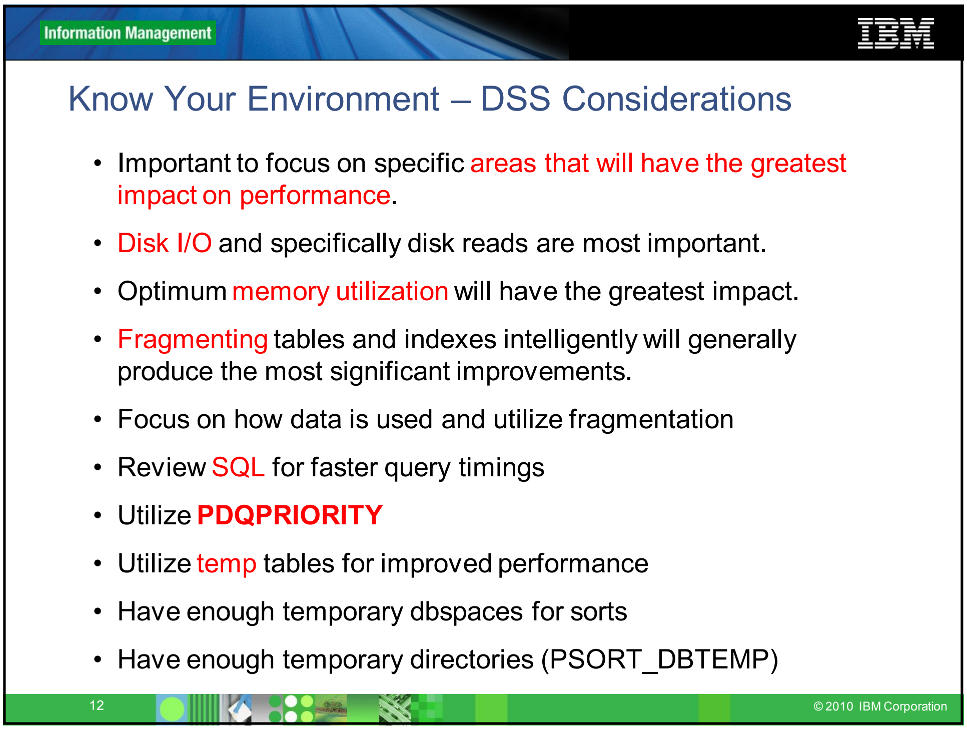
Know Your Environment – OLTP Considerations

- Important to spend time in areas that will have the greatest impact on performance.
- Important to realize that performance tuning is iterative in nature, and doing too many changes at one time can present a challenge.
- Benchmarking is key to knowing what is acceptable performance.
- Focus on index reads, quick return of small amounts of data
- Review how often SQL statements are executed, use SQL Statement Cache
- Understand how the data is searched to then create the correct indexes
- Review application logic if possible



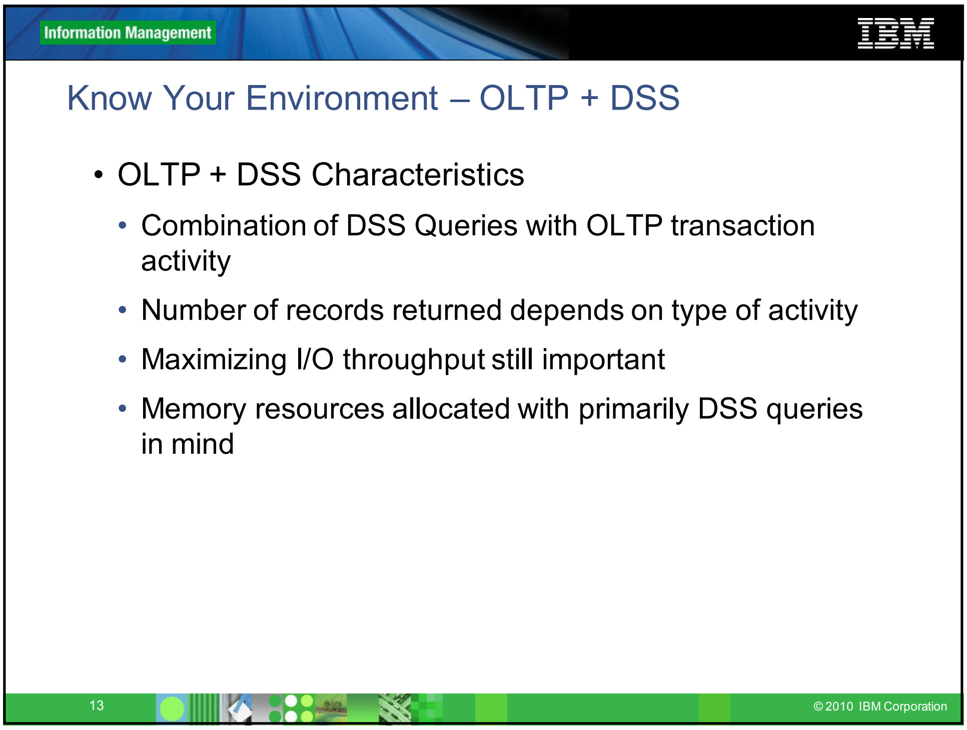
Know Your Environment - DSS

- DSS Characteristics
  - Large number of rows processed and returned
  - Summations/Aggregations common
  - Optimum memory utilization
  - Parallel data queries (PDQ)
  - Light scans (onstat –g lsc)
  - Maximum I/O throughput
  - Tend to be queried by front-end tools



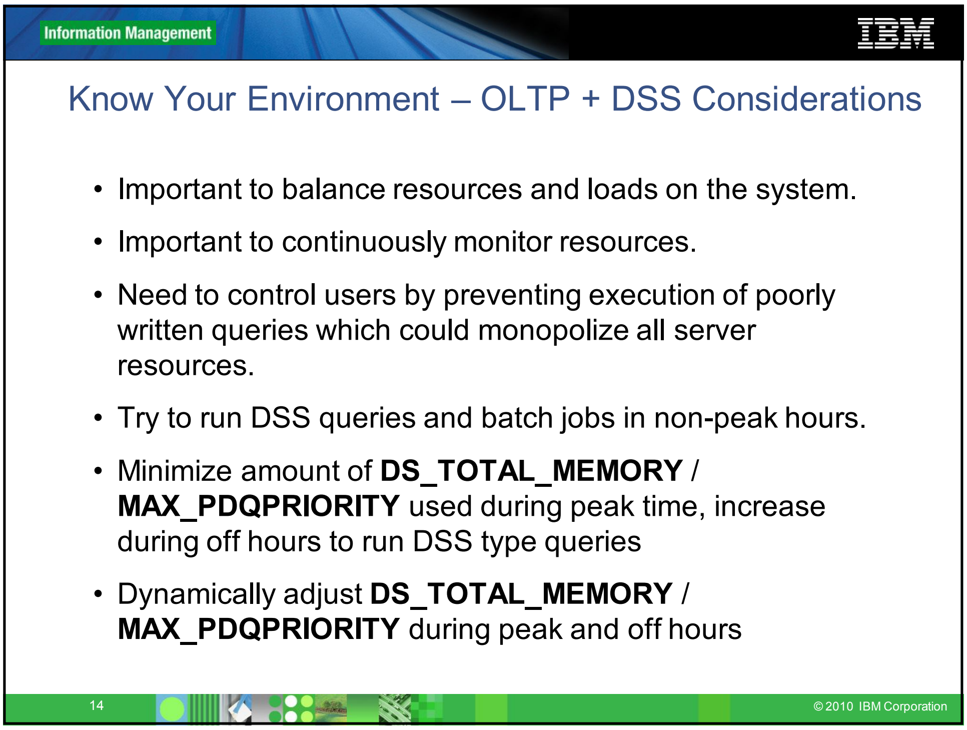
Know Your Environment – DSS Considerations

- Important to focus on specific areas that will have the greatest impact on performance.
- Disk I/O and specifically disk reads are most important.
- Optimum memory utilization will have the greatest impact.
- Fragmenting tables and indexes intelligently will generally produce the most significant improvements.
- Focus on how data is used and utilize fragmentation
- Review SQL for faster query timings
- Utilize **PDQPRIORITY**
- Utilize temp tables for improved performance
- Have enough temporary dbspaces for sorts
- Have enough temporary directories (PSORT_DBTEMP)



Know Your Environment – OLTP + DSS

- OLTP + DSS Characteristics
  - Combination of DSS Queries with OLTP transaction activity
  - Number of records returned depends on type of activity
  - Maximizing I/O throughput still important
  - Memory resources allocated with primarily DSS queries in mind



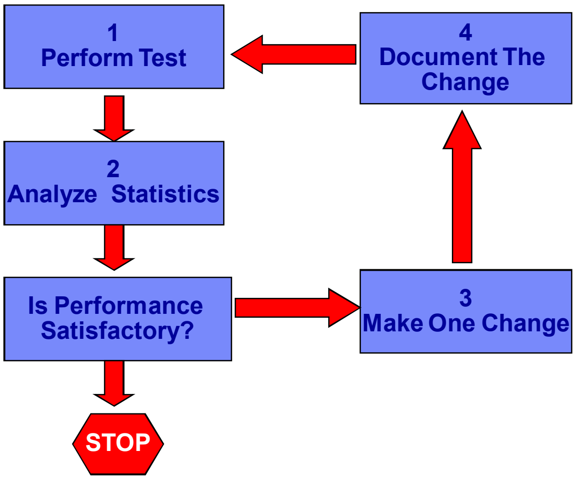
Know Your Environment – OLTP + DSS Considerations

- Important to balance resources and loads on the system.
- Important to continuously monitor resources.
- Need to control users by preventing execution of poorly written queries which could monopolize all server resources.
- Try to run DSS queries and batch jobs in non-peak hours.
- Minimize amount of **DS_TOTAL_MEMORY** / **MAX_PDQPRIORITY** used during peak time, increase during off hours to run DSS type queries
- Dynamically adjust **DS_TOTAL_MEMORY** / **MAX_PDQPRIORITY** during peak and off hours

Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix

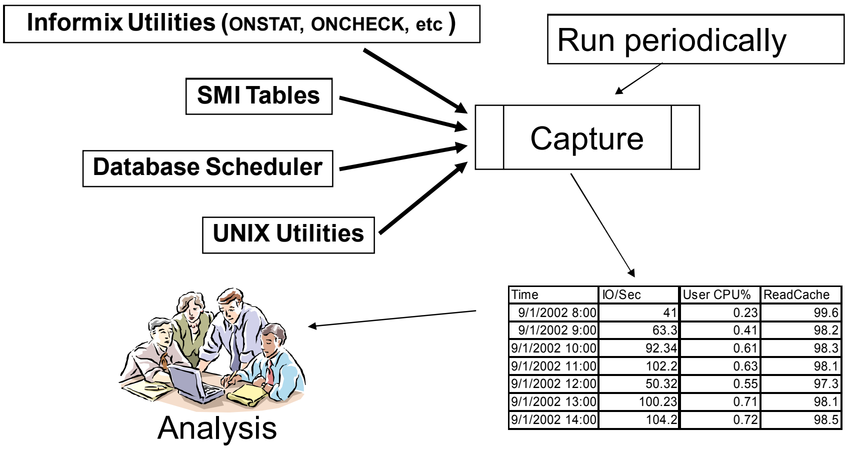
Rules of Performance Tuning



Be Prepared!

- Understand how the system is supposed to work!
- Properly diagnosing a performance problem will require data on system behavior
- You need to have a baseline to compare that with.
  - Having relevant baseline data available makes problem diagnosis easier and faster
  - Also useful for trending analysis as the system grows
- Have the right configuration and profile data available which reflects how things are *when all is well*
  - Being able to compare that with the system configuration and/or behavior when a problem occurs will make our lives much easier.

Create a Base Line BEFORE Problems Develop!



Informix Utilities (ONSTAT, ONCHECK, etc)

SMI Tables

Database Scheduler

UNIX Utilities

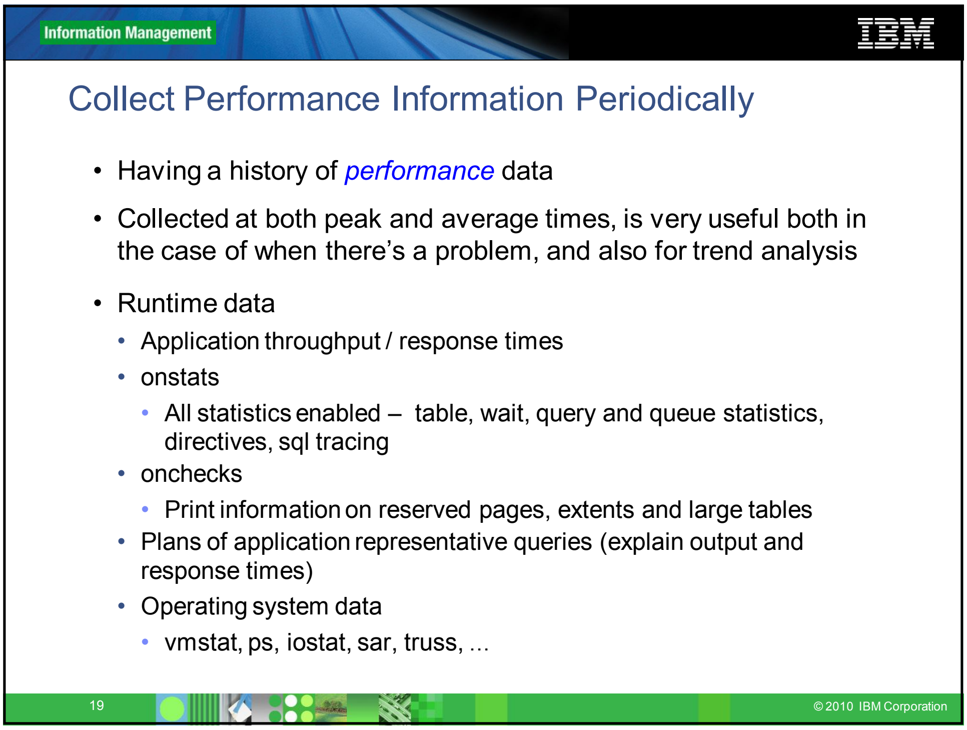
Run periodically

Capture

Time	IO/Sec	User CPU%	ReadCache
9/1/2002 8:00	41	0.23	99.6
9/1/2002 9:00	63.3	0.41	98.2
9/1/2002 10:00	92.34	0.61	98.3
9/1/2002 11:00	102.2	0.63	98.1
9/1/2002 12:00	50.32	0.55	97.3
9/1/2002 13:00	100.23	0.71	98.1
9/1/2002 14:00	104.2	0.72	98.5

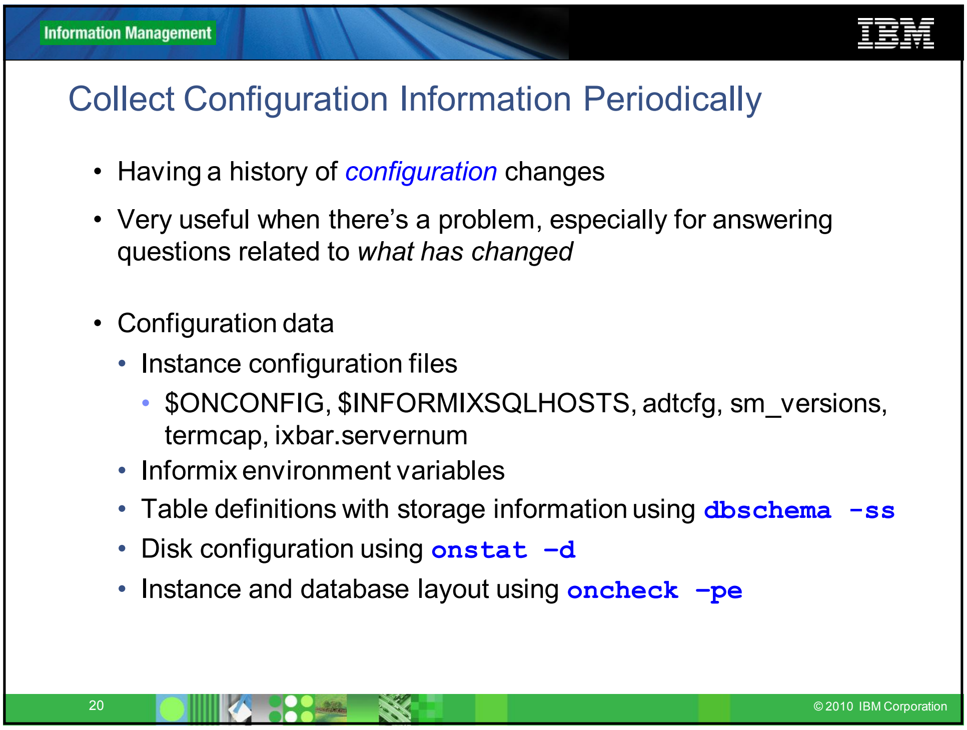
Analysis

Try to have several weeks of data before you need it so that trends and cycles can be understood.



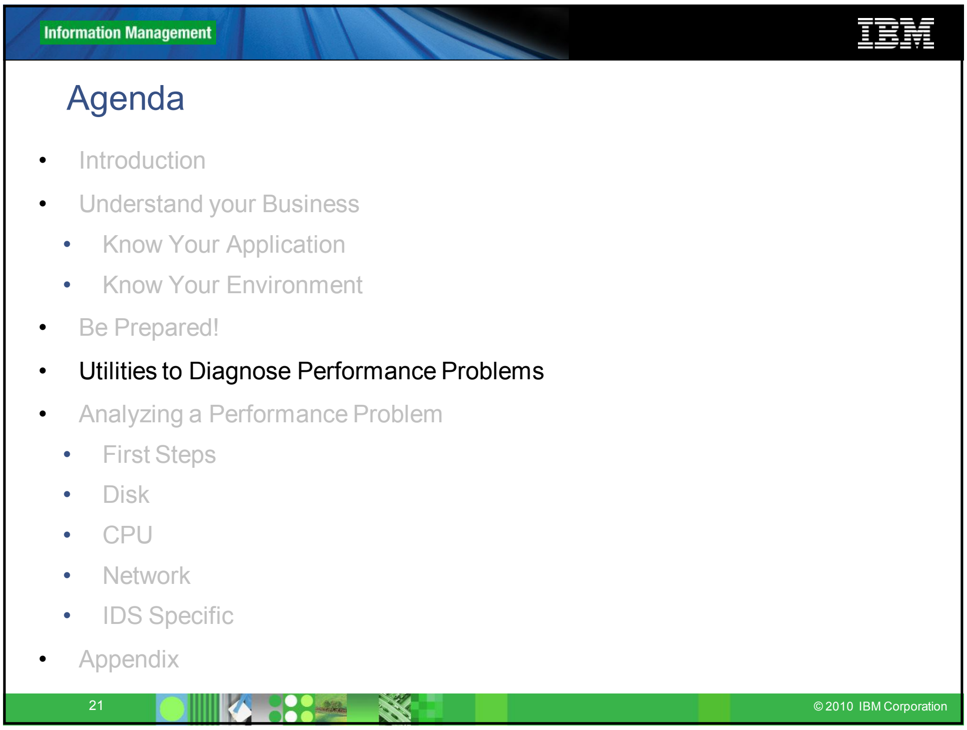
Collect Performance Information Periodically

- Having a history of *performance* data
- Collected at both peak and average times, is very useful both in the case of when there’s a problem, and also for trend analysis
- Runtime data
  - Application throughput / response times
  - onstats
    - All statistics enabled – table, wait, query and queue statistics, directives, sql tracing
  - onchecks
    - Print information on reserved pages, extents and large tables
  - Plans of application representative queries (explain output and response times)
  - Operating system data
    - vmstat, ps, iostat, sar, truss, ...



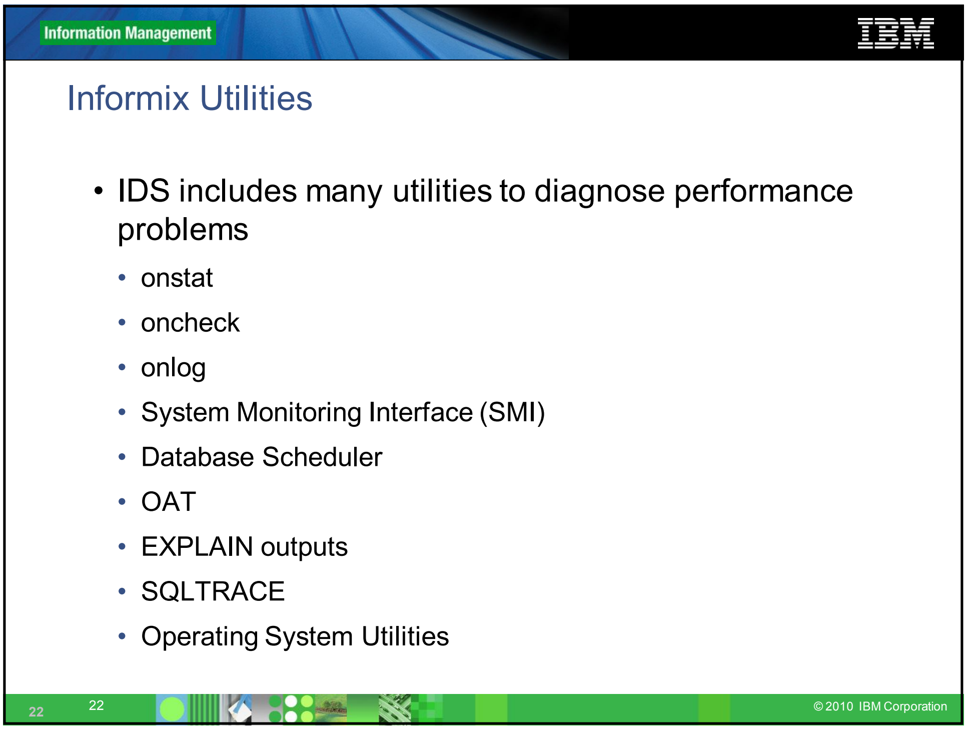
Collect Configuration Information Periodically

- Having a history of *configuration* changes
- Very useful when there’s a problem, especially for answering questions related to *what has changed*
- Configuration data
  - Instance configuration files
    - $ONCONFIG, $INFORMIXSQLHOSTS, adtcfg, sm_versions, termcap, ixbar.servernum
  - Informix environment variables
  - Table definitions with storage information using **`dbschema -ss`**
  - Disk configuration using **`onstat -d`**
  - Instance and database layout using **`oncheck -pe`**



Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



Informix Utilities

- IDS includes many utilities to diagnose performance problems
  - onstat
  - oncheck
  - onlog
  - System Monitoring Interface (SMI)
  - Database Scheduler
  - OAT
  - EXPLAIN outputs
  - SQLTRACE
  - Operating System Utilities

Operating System Utilities

- *ipcs* (information about IDS shared memory)
  - Prints information on active inter-process communication facilities
  - Shared memory segments
  - Semaphores
  - Message queue
- *sar*
  - Monitor cpu utilization
  - Reporting on disk activity
  - Reporting on memory allocation
- *ps*
  - Process state
  - Priority of the process
  - Memory utilization
- *vmstat*
  - Paging/page outs
  - Swapping
  - Page scan rates
  - Free memory
- *time* or *timex*
  - Time the running of a process in real time, user time and system CPU time
- *iostat*
  - Provides data about the I/O device usage of the system
  - Measures and reports
    - disk throughput
    - how many Kbytes are being transferred per second
    - number of seeks per second
    - percentage of disk utilization
    - Number of physical reads or writes per second

Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix

Analyzing a performance problem: Characterization

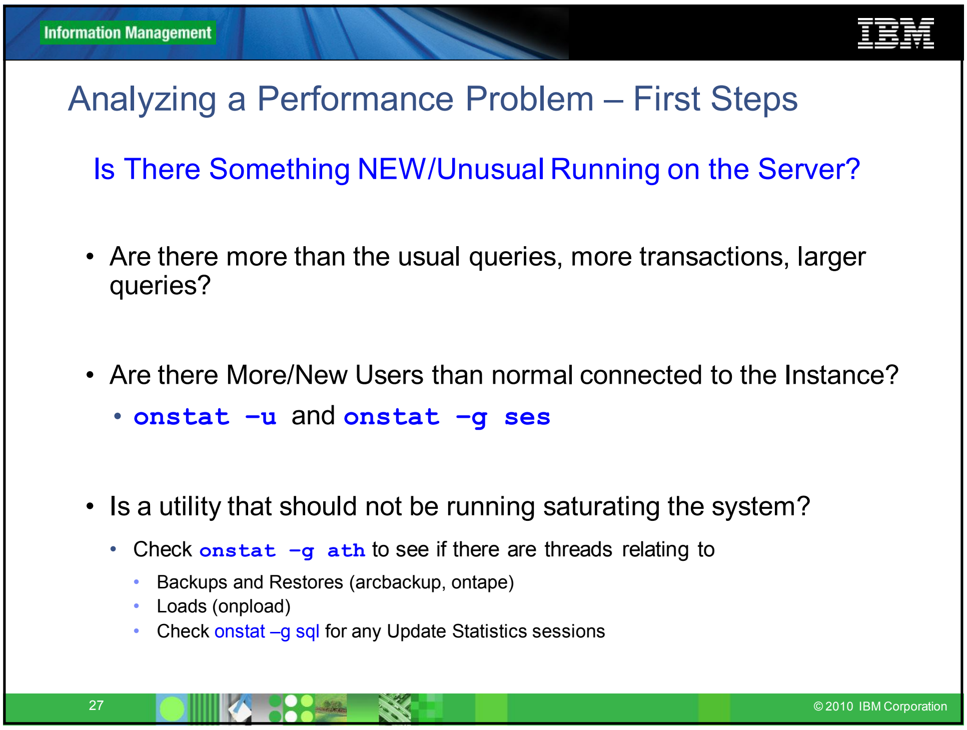
- Types of performance problems
  - The database server seems unresponsive
  - Users are experiencing long processing times
  - Performance for a particular query has suddenly slowed
  - The Message Log is showing long checkpoint durations
  - An IDS utility is processing very slowly
  - New connections are slow

Analyzing a Performance Problem – First Steps

- The most basic question that should be asked first is:

  Has anything been changed on the server?

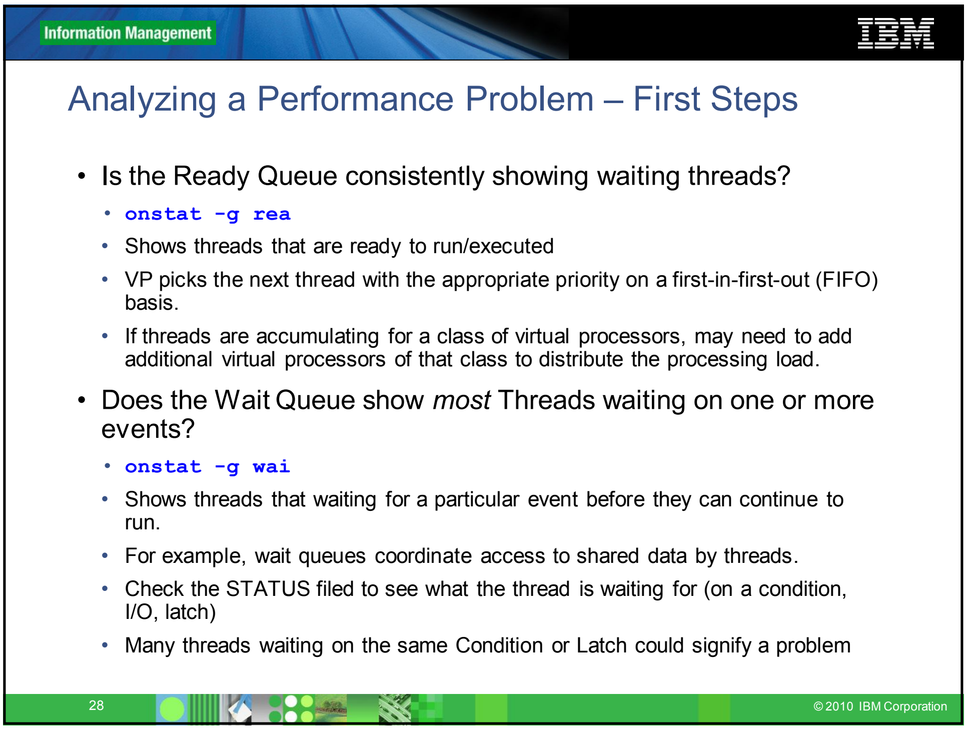
- Often the trigger of a performance issue is a change in the requirements or on the load of the system:
  - New database applications?
  - Other non-database loads on the system?
  - More users?
  - More data?
  - Software/Database configuration changes?
    - Database configuration, environment variables, schema changes, etc.
  - Hardware configuration changes?
- If you can identify the change at this stage, can it / should it be undone or is this something you have to adapt
- If it was unintentional, or not critical, can it / should it be undone or is this something you have to live with?



Analyzing a Performance Problem – First Steps

Is There Something NEW/Unusual Running on the Server?

- Are there more than the usual queries, more transactions, larger queries?
- Are there More/New Users than normal connected to the Instance?
  - `onstat –u` and `onstat –g ses`
- Is a utility that should not be running saturating the system?
  - Check `onstat –g ath` to see if there are threads relating to
    - Backups and Restores (arcbackup, ontape)
    - Loads (onpload)
    - Check onstat –g sql for any Update Statistics sessions



Analyzing a Performance Problem – First Steps

- Is the Ready Queue consistently showing waiting threads?
  - `onstat -g rea`
  - Shows threads that are ready to run/executed
  - VP picks the next thread with the appropriate priority on a first-in-first-out (FIFO) basis.
  - If threads are accumulating for a class of virtual processors, may need to add additional virtual processors of that class to distribute the processing load.
- Does the Wait Queue show *most* Threads waiting on one or more events?
  - `onstat -g wai`
  - Shows threads that waiting for a particular event before they can continue to run.
  - For example, wait queues coordinate access to shared data by threads.
  - Check the STATUS filed to see what the thread is waiting for (on a condition, I/O, latch)
  - Many threads waiting on the same Condition or Latch could signify a problem

Analyzing a Performance Problem – First Steps

- What are the values of some key server statistics?
  - **onstat -p**

```
database sysmaster;
select  name, value from sysprofile
where name in
( "iscommits", "isrollbacks",
  "ovlock", "ovuser", “ovbuff”
  "latchwts", "buffwts", "lockwts", "ckptwts",
  "deadlks",  "lktouts",
  "fgwrites", "lruwrites", "chunkwrites" )
```

fgwrites	Foreground write - caused when a session needs to have a page from disk placed into the buffer pool, but there are no clean/free buffers available.
lruwrites	LRU writes - background writes that typically occur when the percentage of dirty buffers exceeds the percent that is specified for lru_max_dirty in the BUFFERPOOL configuration parameter.
chunkwrites	Chunk writes - commonly performed by page-cleaner threads during a checkpoint

iscommit	The number of times OnLine performed a commit
isrollbk	The number of times a transaction was rolled back
ovlock	Increments when the value of LOCKS is exceeded .
ovuser	Increments when the maximum number of userthreads is exceeded relative to the setting of NETTYPE.
ovbuff	The number of times a request was made for a buffer in the buffer pool but none was available.
latchkwts	Increments when a userthread must wait to acquire a latch
buffwts	Increments when a userthread must wait to acquire a buffer
lockwts	Increments when a userthread must wait to acquire a lock
ckptwts	The number of times a thread has had to wait for a checkpoint to complete before continuing.
deadlks	The number of potential deadlocks situations that have been encountered
lktouts	The number of times users experienced Lock Timeouts

Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix

Is There a Table That is Very Heavily Utilized?

- What tables have the most I/O?
  - `onstat -P`

```
database sysmaster;
select
           dbsname,
           tabname,
           DBINFO ( 'dbspace', partnum ),
           lockreqs, lockwts, deadlks, lktouts,
           isreads, iswrites, isrewrites, isdeletes,
           bufreads, bufwrites, pagreads,
           pagwrites
from       sysptprof
order by isreads desc;

-- change this sort to whatever you need to monitor.
```

```
database sysmaster;
select     tabname[1,8] table,
           sum(isreads) reads,
           sum(iswrites) writes,
           sum(isrewrites) updates,
           sum(isdeletes) deletes
from sysptprof
where tabname not like "sys%"
group by tabname
order by reads desc,
          writes desc,
          updates desc,
          deletes desc;
```

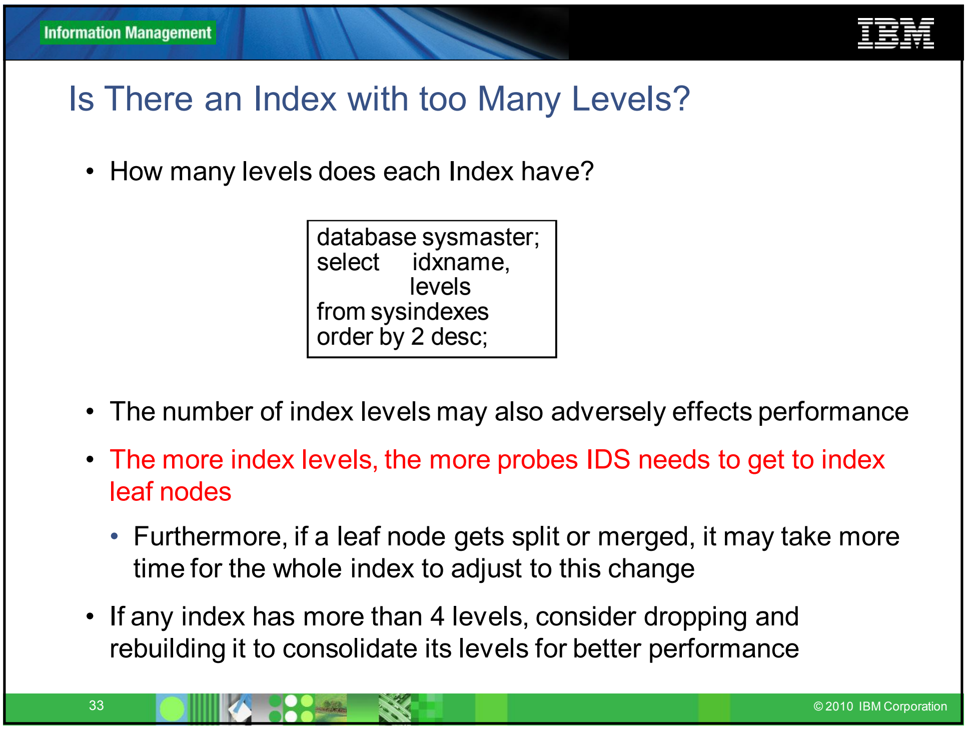
- Heavily utilized tables can cause disk contention
- Is the data for the table fragmented on disk?
  - `oncheck -pt`

Is There a Table With Too Many Extents?

- How many extents does each database table have?
  - `oncheck -pe`

```
database sysmaster;
select
           dbsname,
           tabname,
           count(*) num_of_extents,
           sum( pe_size ) total_size
from       systabnames, sysptnext
where     partnum = pe_partnum
group by 1, 2
order by 3 desc, 4 desc;
```

- An abnormal (> 30) number of extents can slow queries
  - As a result of excessive disk head movement to find and go to the many extents
  - Consider rebuilding the table

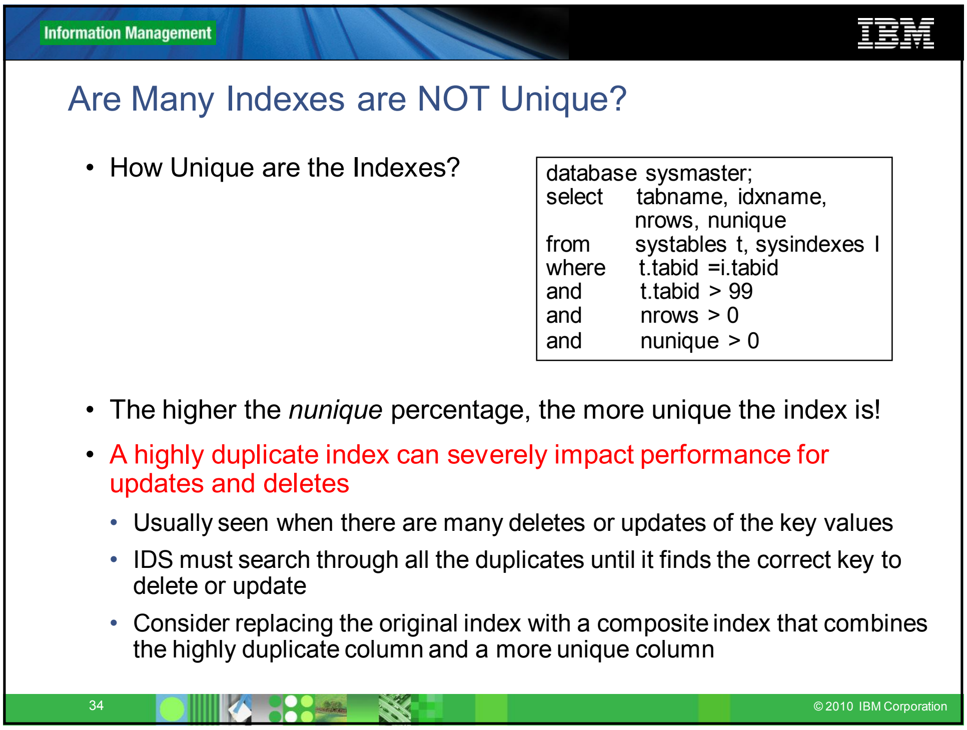


Is There an Index with too Many Levels?

- How many levels does each Index have?

```
database sysmaster;
select     idxname,
           levels
from sysindexes
order by 2 desc;
```

- The number of index levels may also adversely effects performance
- The more index levels, the more probes IDS needs to get to index leaf nodes
  - Furthermore, if a leaf node gets split or merged, it may take more time for the whole index to adjust to this change
- If any index has more than 4 levels, consider dropping and rebuilding it to consolidate its levels for better performance



Are Many Indexes are NOT Unique?

- How Unique are the Indexes?

```
database sysmaster;
select     tabname, idxname,
           nrows, nunique
from       systables t, sysindexes I
where      t.tabid =i.tabid
and        t.tabid > 99
and        nrows > 0
and        nunique > 0
```

- The higher the *nunique* percentage, the more unique the index is!
- A highly duplicate index can severely impact performance for updates and deletes
  - Usually seen when there are many deletes or updates of the key values
  - IDS must search through all the duplicates until it finds the correct key to delete or update
  - Consider replacing the original index with a composite index that combines the highly duplicate column and a more unique column

Are there too many Sequential Scans?

- How many sequential scans are being performed against each Table?
  - `onstat -g ppf`
  - `onstat -p`

```
database sysmaster;
select     dbsname,
           tabname,
           sum(seqscans) total_scans
from       sysptprof
where      seqscans > 0   AND
           dbsname not like "sys%"
group by 1, 2
order by 3 desc
```

- Missing indexes or under-utilized asynchronous data reads can severely affect performance
- High buffer waits may also indicate the difference between RA_PAGES and RA_THRESHOLD is too small
  ReadAheadpct = ((RA-pgsused / (ixda-RA + idx-RA + da-RA)) *100)

Is there excessive I/O one or more Chunks?

- What Chunks have the most I/O?
  - `onstat -D`
  - `onstat -g iof`
- If the data is not spread evenly across dbspaces and chunks, each on different devices, performance can suffer due to I/O waits

```
database sysmaster;
select     name
           dbspace,
           chknum, " Primary " chktype,
           fname[15,25] path_name,
           reads, writes,
           pagesread, pageswritten
from       syschktab c, sysdbstab d
where      c.dbsnum = d.dbsnum
union all
select
           name dbspace,
           chknum, "Mirror" chktype,
           fname[15,25] path_name,
           reads, writes,
           pagesread, pageswritten
from       sysmchktab c, sysdbstab d
where      c.dbsnum = d.dbsnum
order by 1,2,3;
```

Are there excessive Disk Page Reads?

- What percent of I/O is from buffers?
  - **onstat -p**

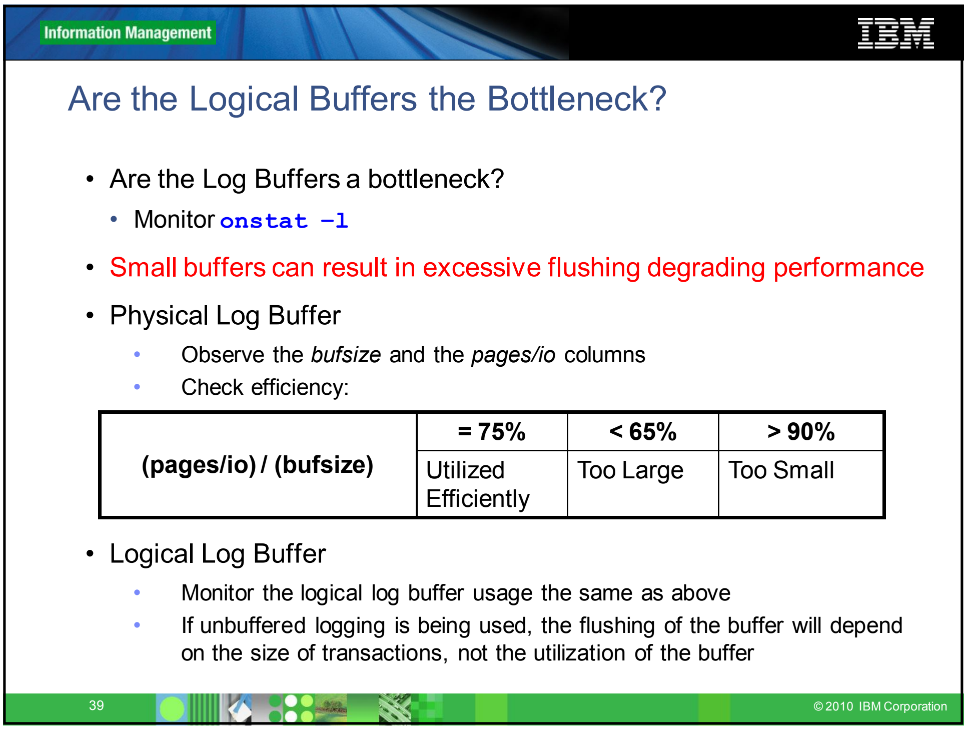
```
databse sysmaster;
-- Get % read cached
select     dr.value dskreads, br.value bufreads,
           round ((( 1 - ( dr.value / br.value )) *100 ), 2) cached
from       sysprofile dr, sysprofile br
where      dr.name = "dskreads“  AND
           br.name = "bufreads";

-- Get % write cached
select     dw.value dskwrites, bw.value bufwrites,
           round ((( 1 - ( dw.value / bw.value )) *100 ), 2) cached
from       sysprofile dw, sysprofile bw
where      dw.name = "dskwrites“  AND
           bw.name = "bufwrites"
```

- In OLTP environments, the Read Cache % should be over 95% and the Write Cache % over 80%

Are the Logical Logs the Bottleneck?

- Is anything using the same disks as the Logs?
  - Run **oncheck -pe** to see what else is in the same dbspace as the logs
  - Check if *iostat* shows > 80 IO/s on a single disk where the logs reside
- Are there an abnormally higher number of transactions?
  - Check commit frequency
    - Monitor using **onstat -l** –(*numwrits* value)
  - Buffered Logging will also cause more log buffer flushes
    - Consider changing to Unbuffered Logging
  - Application snapshot to find out who is committing so frequently
    - Monitor using **onstat -g tpf** (*isct* value)
- Logs can be very performance-sensitive in an OLTP environments
  - A good place to use your best hardware
  - Consider moving logical logs to their own dbspace, faster disks, RAID parallelization with small (e.g. 8k) stripe size, and/or fast controller with write caching

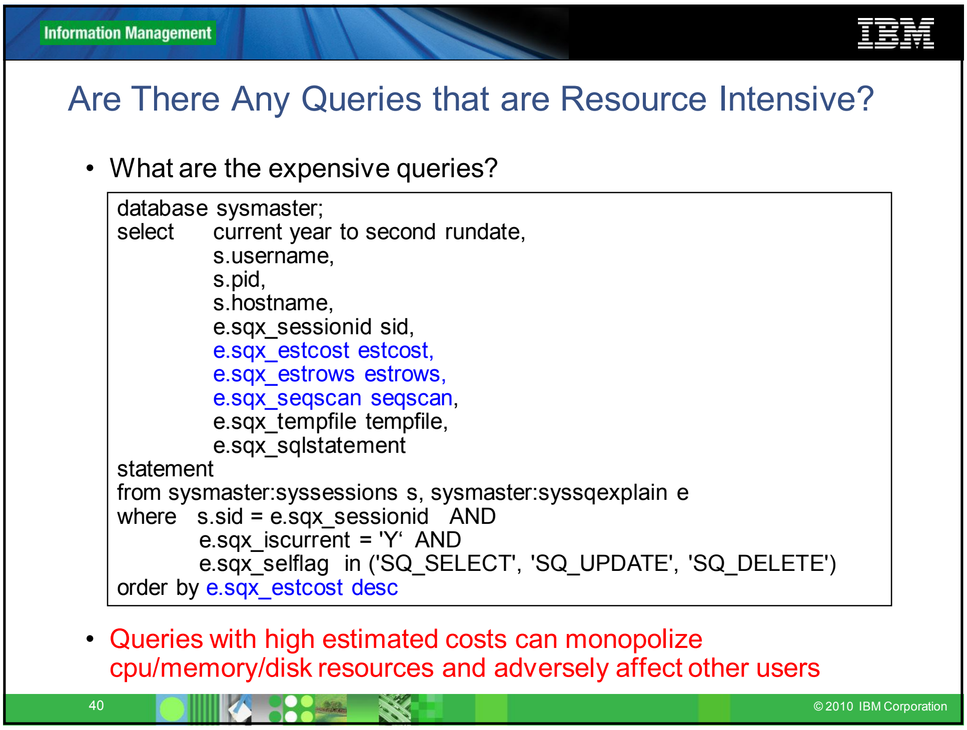


Are the Logical Buffers the Bottleneck?

- Are the Log Buffers a bottleneck?
  - Monitor `onstat -l`
- Small buffers can result in excessive flushing degrading performance
- Physical Log Buffer
  - Observe the *bufsize* and the *pages/io* columns
  - Check efficiency:

(pages/io) / (bufsize)	**= 75%**	**< 65%**	**> 90%**
	Utilized Efficiently	Too Large	Too Small

- Logical Log Buffer
  - Monitor the logical log buffer usage the same as above
  - If unbuffered logging is being used, the flushing of the buffer will depend on the size of transactions, not the utilization of the buffer

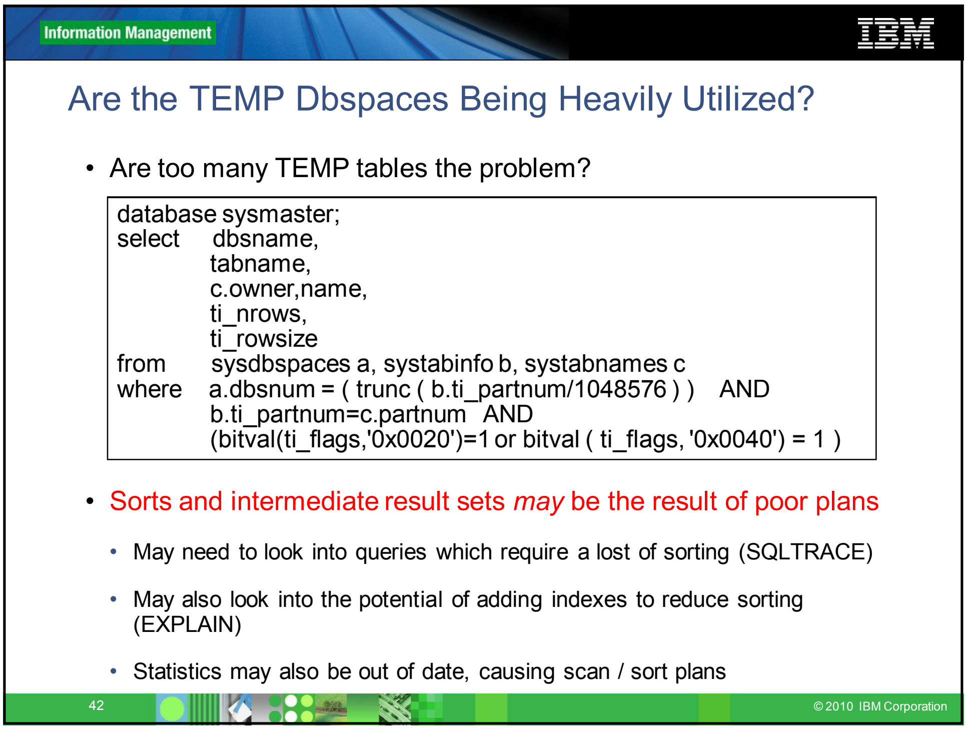


Are There Any Queries that are Resource Intensive?

- What are the expensive queries?

```
database sysmaster;
select     current year to second rundate,
           s.username,
           s.pid,
           s.hostname,
           e.sqx_sessionid sid,
           e.sqx_estcost estcost,
           e.sqx_estrows estrows,
           e.sqx_seqscan seqscan,
           e.sqx_tempfile tempfile,
           e.sqx_sqlstatement
statement
from sysmaster:syssessions s, sysmaster:syssqexplain e
where   s.sid = e.sqx_sessionid   AND
          e.sqx_iscurrent = 'Y'  AND
          e.sqx_selflag  in ('SQ_SELECT', 'SQ_UPDATE', 'SQ_DELETE')
order by e.sqx_estcost desc
```

- Queries with high estimated costs can monopolize cpu/memory/disk resources and adversely affect other users

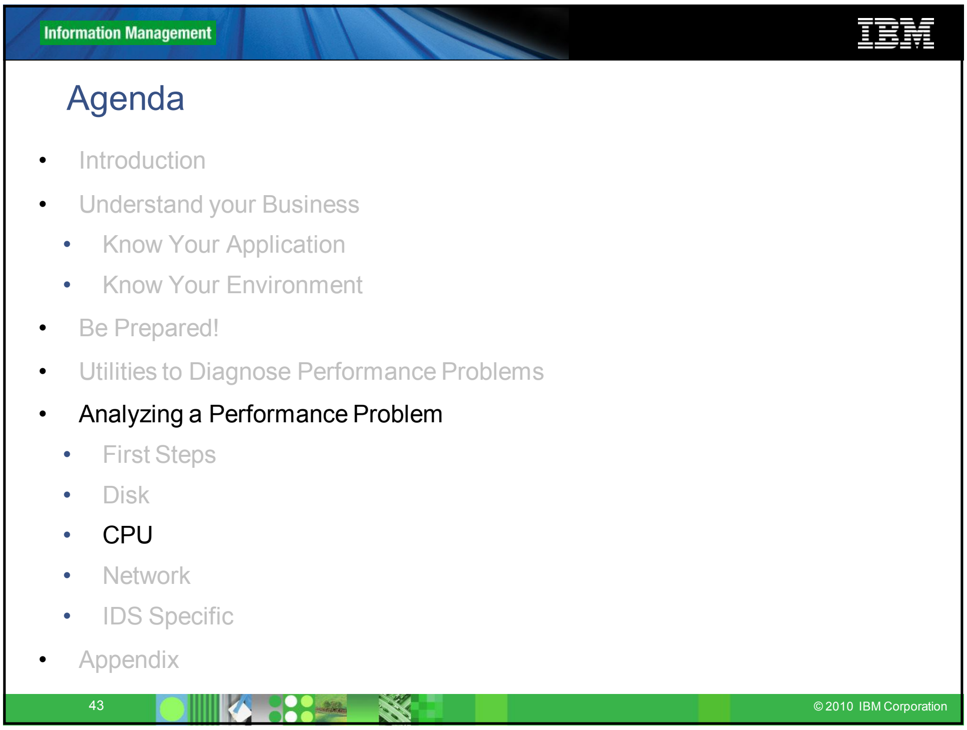


Are the TEMP Dbspaces Being Heavily Utilized?

- Are too many TEMP tables the problem?

```
database sysmaster;
select     dbsname,
           tabname,
           c.owner,name,
           ti_nrows,
           ti_rowsize
from       sysdbspaces a, systabinfo b, systabnames c
where      a.dbsnum = ( trunc ( b.ti_partnum/1048576 ) )    AND
           b.ti_partnum=c.partnum   AND
           (bitval(ti_flags,'0x0020')=1 or bitval ( ti_flags, '0x0040') = 1 )
```

- Sorts and intermediate result sets *may* be the result of poor plans
  - May need to look into queries which require a lost of sorting (SQLTRACE)
  - May also look into the potential of adding indexes to reduce sorting (EXPLAIN)
  - Statistics may also be out of date, causing scan / sort plans



Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix

Is There a THREAD Doing a Lot of Work?

- What THREADs are consuming the most Virtual CPU Time?

onstat –g ath

```
 tid  name            vp      Last Run          CPU Time   #scheds   status
 *2   lio vp 0        3lio    06/27 13:26:39     28.6397      3749   IO Idle
 *3   pio vp 0        4pio    06/27 13:25:09      5.0609       517   IO Idle
 *4   aio vp 0        5aio    06/27 13:29:23     31.1610    112645   IO Idle
 *5   msc vp 0        6msc    06/27 13:27:57      0.1137        50   IO Idle
 *6   aio vp 1        7aio    06/27 13:29:23     19.1152      5524   IO Idle
  7   main_loop()     1cpu    06/27 13:31:55      7.1407    678090   sleeping secs: 1
 *8   sm_poll         1cpu    06/27 13:31:55  67245.0333    940398   running
  9   sm_listen       1cpu    06/27 13:27:57      0.0057        32   sleeping forever
 10   sm_discon       1cpu    06/27 13:31:55      2.5516    676641   sleeping secs: 1
 11   flush_sub(0)    1cpu    06/27 13:31:55      1.7716    677707   sleeping secs: 1
*12   aio vp 2        8aio    06/27 13:29:23     21.7697       727   IO Idle
*13   aio vp 3        9aio    06/27 13:25:09     23.7650       677   IO Idle
*14   aio vp 4       10aio    06/27 13:25:09     18.0777      1118   IO Idle
 16   aslogflush      1cpu    06/27 13:31:55      2.0833    676638   sleeping secs: 1
 17   btscanner_0     1cpu    06/27 13:31:35      1.7299     22352   sleeping secs: 31
*18   onmode_mon      1cpu    06/27 13:31:55      2.9390    676641   sleeping secs: 1
*40   dbScheduler     1cpu    06/27 13:29:23      1.5202      3444   sleeping secs: 148
 71   sqlexec         2cpu    06/27 14:24:22  54277.9907      2655   running
 71   sqlexec         2cpu    06/27 14:24:22    327.3421      2655   IO Wait
```

- Also check the STATUS of the thread to see what it is doing
- If STATUS = running, find the session ID using onstat –u and then use onstat –g <sesid> to find detailed information on the session's activities

Which Virtual Processor is Consuming the Most CPU Cycles?

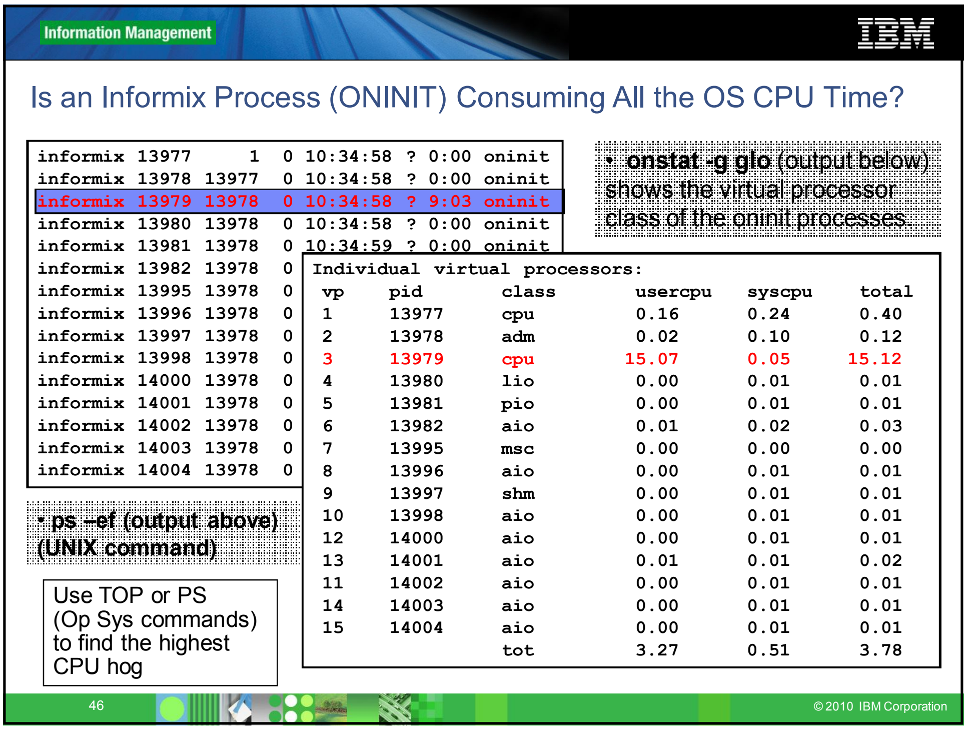
- Is a VP other than the CPU VP consuming the most CPU cycles?

```
database sysmaster;
select   vpid, pid, txt[1,5] class,
         round( usecs_user, 2) usercpu,
         round( usecs_sys, 2) syscpu
from     sysvplst a, flags_text b
where    a.class = b.flags  AND
         b.tabname = "sysvplst"
```

```
vpid pid class   usercpu syscpu
1    295 cpu     503.26  45.22
2    296 adm     0.45    0.72
3    297 lio     1.57    7.83
4    298 pio     0.35    0.21
5    299 aio     7.30    56.16
6    300 msc     0.04    0.64
7    301 aio     3.26    23.75
8    302 tli     0.17    0.54
9    305 pio     0.77    1.02
```

- Inadequate number of VPs to support a workload will cause performance issues
  - **onstat -g glo**
    - Effective CPU Utilization - ***Eff*** column
  - **onstat -g ioq**
    - Maximum queue length - ***maxlen*** column
  - Monitor to see if adding more VPS of a particular class would help

VPCLASS
(CPU, JVP, TLI, SHM, SOC, STR)



Is an Informix Process (ONINIT) Consuming All the OS CPU Time?

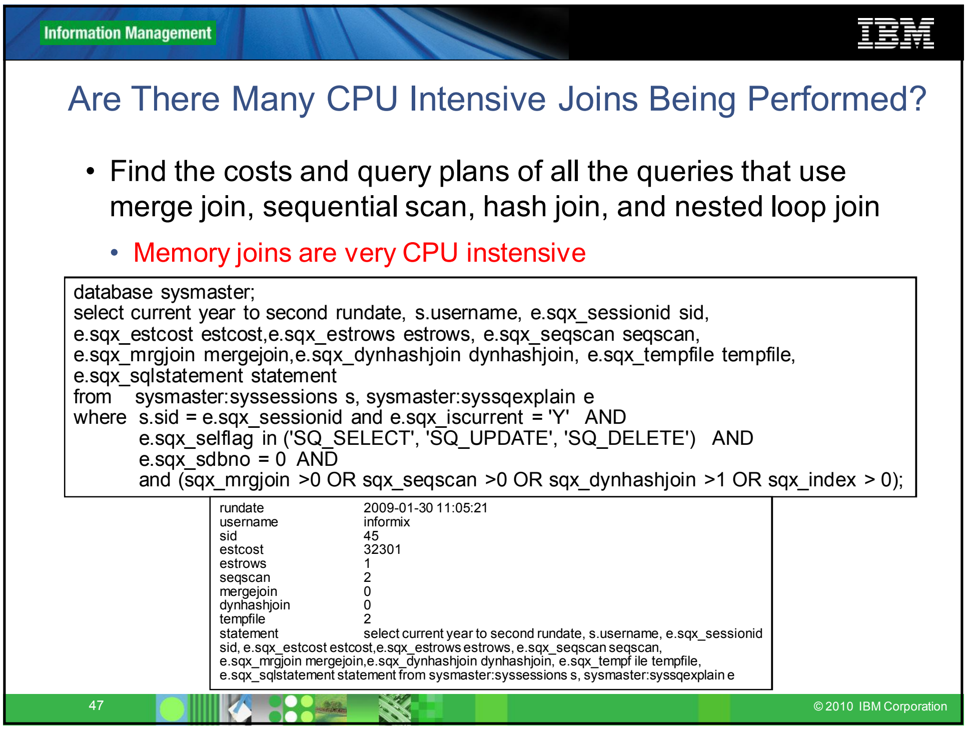
```
informix 13977     1  0 10:34:58 ? 0:00 oninit
informix 13978 13977  0 10:34:58 ? 0:00 oninit
informix 13979 13978  0 10:34:58 ? 9:03 oninit
informix 13980 13978  0 10:34:58 ? 0:00 oninit
informix 13981 13978  0 10:34:59 ? 0:00 oninit
informix 13982 13978  0
informix 13995 13978  0
informix 13996 13978  0
informix 13997 13978  0
informix 13998 13978  0
informix 14000 13978  0
informix 14001 13978  0
informix 14002 13978  0
informix 14003 13978  0
informix 14004 13978  0
```

- **ps –ef (output above) (UNIX command)**

Use TOP or PS (Op Sys commands) to find the highest CPU hog

- **onstat -g glo** (output below) shows the virtual processor class of the oninit processes.

```
Individual virtual processors:
 vp    pid      class       usercpu   syscpu    total
 1     13977    cpu          0.16     0.24      0.40
 2     13978    adm          0.02     0.10      0.12
 3     13979    cpu         15.07     0.05     15.12
 4     13980    lio          0.00     0.01      0.01
 5     13981    pio          0.00     0.01      0.01
 6     13982    aio          0.01     0.02      0.03
 7     13995    msc          0.00     0.00      0.00
 8     13996    aio          0.00     0.01      0.01
 9     13997    shm          0.00     0.01      0.01
 10    13998    aio          0.00     0.01      0.01
 12    14000    aio          0.00     0.01      0.01
 13    14001    aio          0.01     0.01      0.02
 11    14002    aio          0.00     0.01      0.01
 14    14003    aio          0.00     0.01      0.01
 15    14004    aio          0.00     0.01      0.01
                tot          3.27     0.51      3.78
```

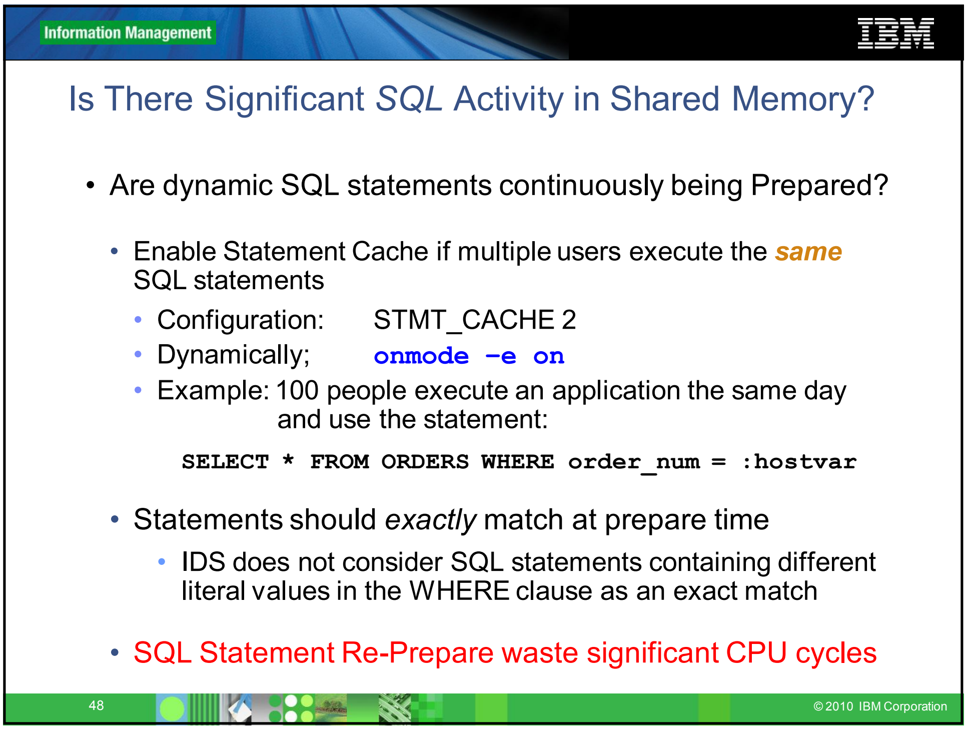


Are There Many CPU Intensive Joins Being Performed?

- Find the costs and query plans of all the queries that use merge join, sequential scan, hash join, and nested loop join
  - Memory joins are very CPU instensive

```
database sysmaster;
select current year to second rundate, s.username, e.sqx_sessionid sid,
e.sqx_estcost estcost,e.sqx_estrows estrows, e.sqx_seqscan seqscan,
e.sqx_mrgjoin mergejoin,e.sqx_dynhashjoin dynhashjoin, e.sqx_tempfile tempfile,
e.sqx_sqlstatement statement
from    sysmaster:syssessions s, sysmaster:syssqexplain e
where  s.sid = e.sqx_sessionid and e.sqx_iscurrent = 'Y'   AND
         e.sqx_selflag in ('SQ_SELECT', 'SQ_UPDATE', 'SQ_DELETE')   AND
         e.sqx_sdbno = 0  AND
         and (sqx_mrgjoin >0 OR sqx_seqscan >0 OR sqx_dynhashjoin >1 OR sqx_index > 0);
```

```
rundate          2009-01-30 11:05:21
username         informix
sid              45
estcost          32301
estrows          1
seqscan          2
mergejoin        0
dynhashjoin      0
tempfile         2
statement        select current year to second rundate, s.username, e.sqx_sessionid
sid, e.sqx_estcost estcost,e.sqx_estrows estrows, e.sqx_seqscan seqscan,
e.sqx_mrgjoin mergejoin,e.sqx_dynhashjoin dynhashjoin, e.sqx_tempf ile tempfile,
e.sqx_sqlstatement statement from sysmaster:syssessions s, sysmaster:syssqexplain e
```

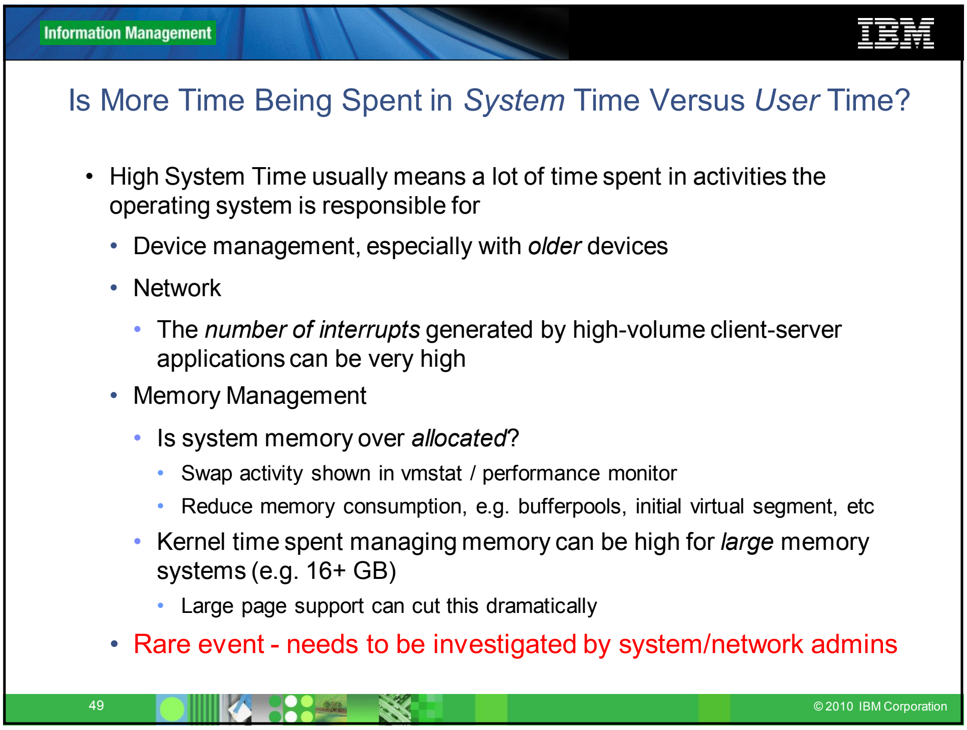


Is There Significant *SQL* Activity in Shared Memory?

- Are dynamic SQL statements continuously being Prepared?
  - Enable Statement Cache if multiple users execute the ***same*** SQL statements
    - Configuration: STMT_CACHE 2
    - Dynamically; `onmode -e on`
    - Example: 100 people execute an application the same day and use the statement:

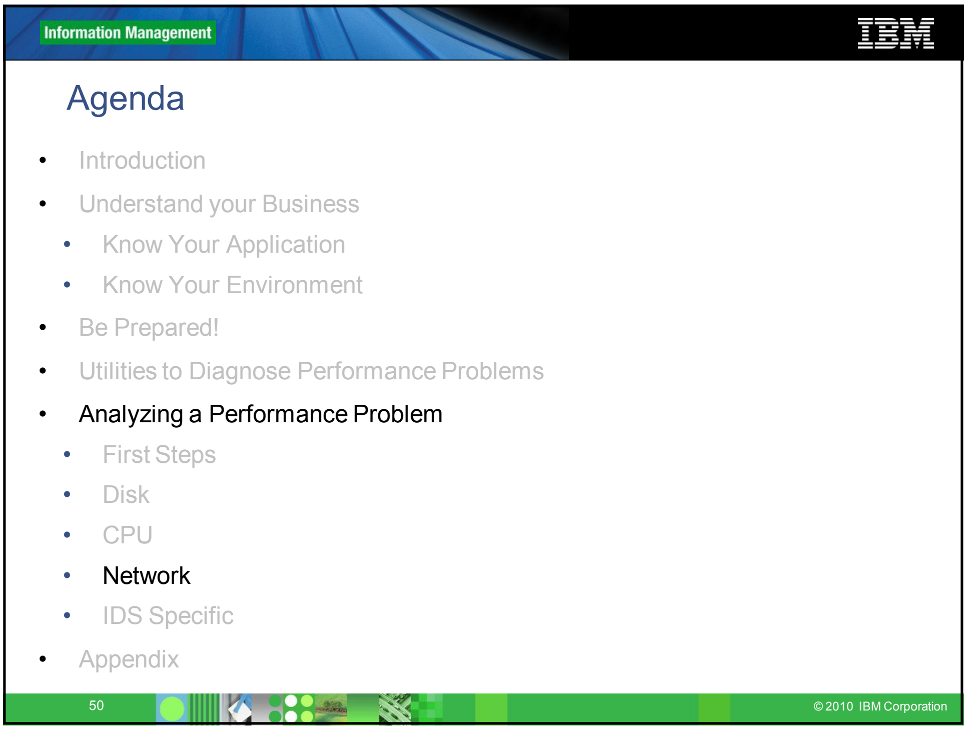
```
SELECT * FROM ORDERS WHERE order_num = :hostvar
```

  - Statements should *exactly* match at prepare time
    - IDS does not consider SQL statements containing different literal values in the WHERE clause as an exact match
  - SQL Statement Re-Prepare waste significant CPU cycles



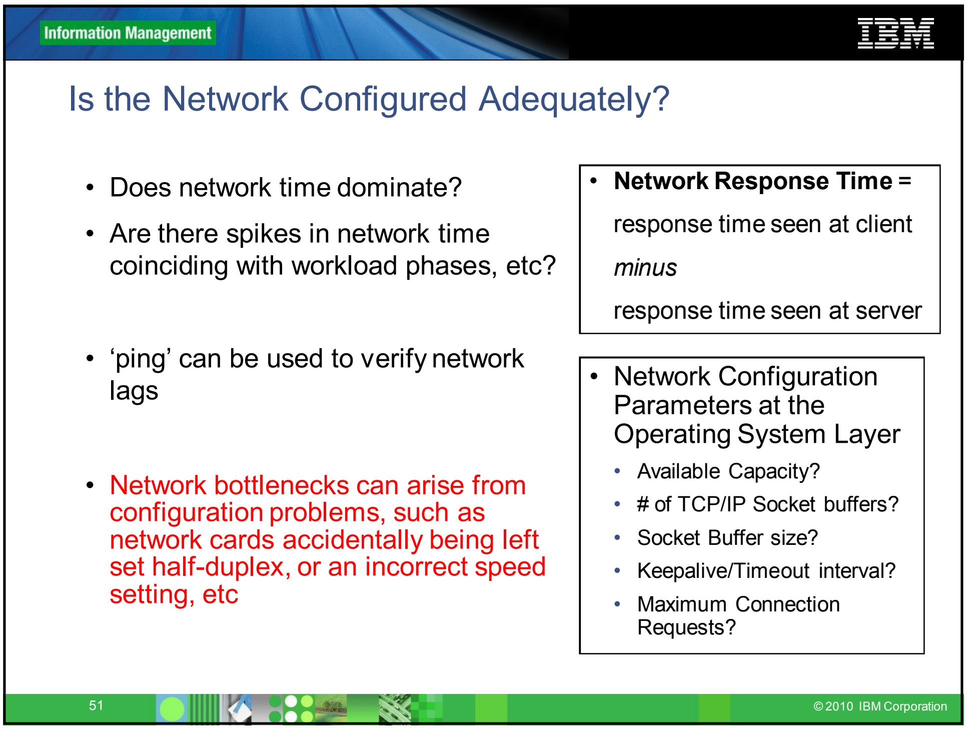
Is More Time Being Spent in *System* Time Versus *User* Time?

- High System Time usually means a lot of time spent in activities the operating system is responsible for
  - Device management, especially with *older* devices
  - Network
    - The *number of interrupts* generated by high-volume client-server applications can be very high
  - Memory Management
    - Is system memory over *allocated*?
      - Swap activity shown in vmstat / performance monitor
      - Reduce memory consumption, e.g. bufferpools, initial virtual segment, etc
    - Kernel time spent managing memory can be high for *large* memory systems (e.g. 16+ GB)
      - Large page support can cut this dramatically
  - Rare event - needs to be investigated by system/network admins



Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



Is the Network Configured Adequately?

- Does network time dominate?
- Are there spikes in network time coinciding with workload phases, etc?
- ‘ping’ can be used to verify network lags
- Network bottlenecks can arise from configuration problems, such as network cards accidentally being left set half-duplex, or an incorrect speed setting, etc

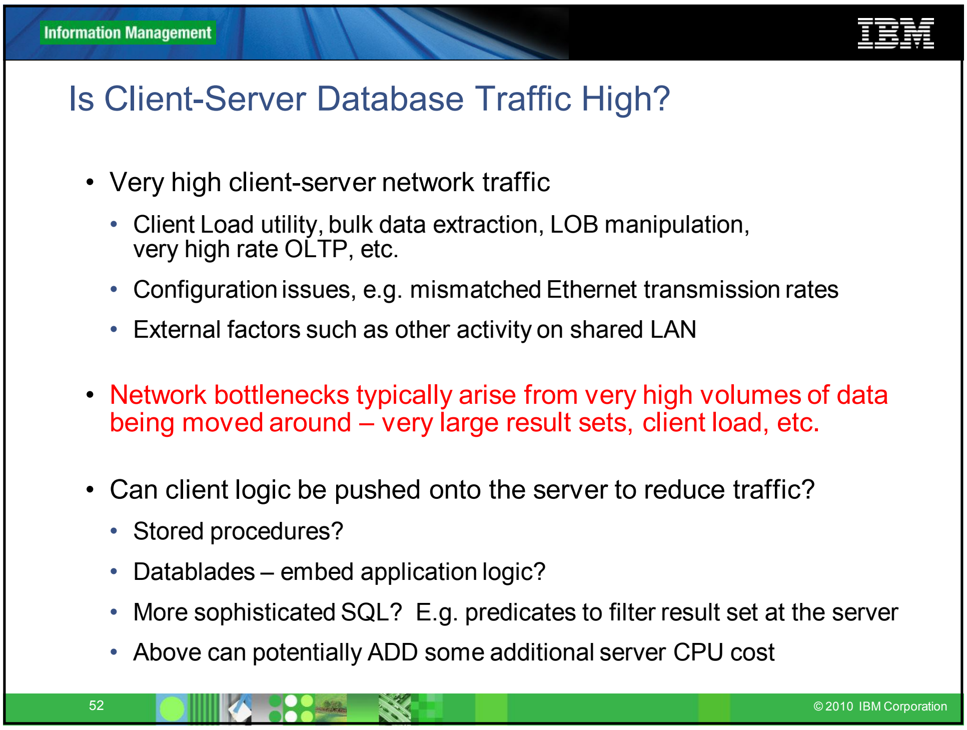
- **Network Response Time =**

  response time seen at client

  minus

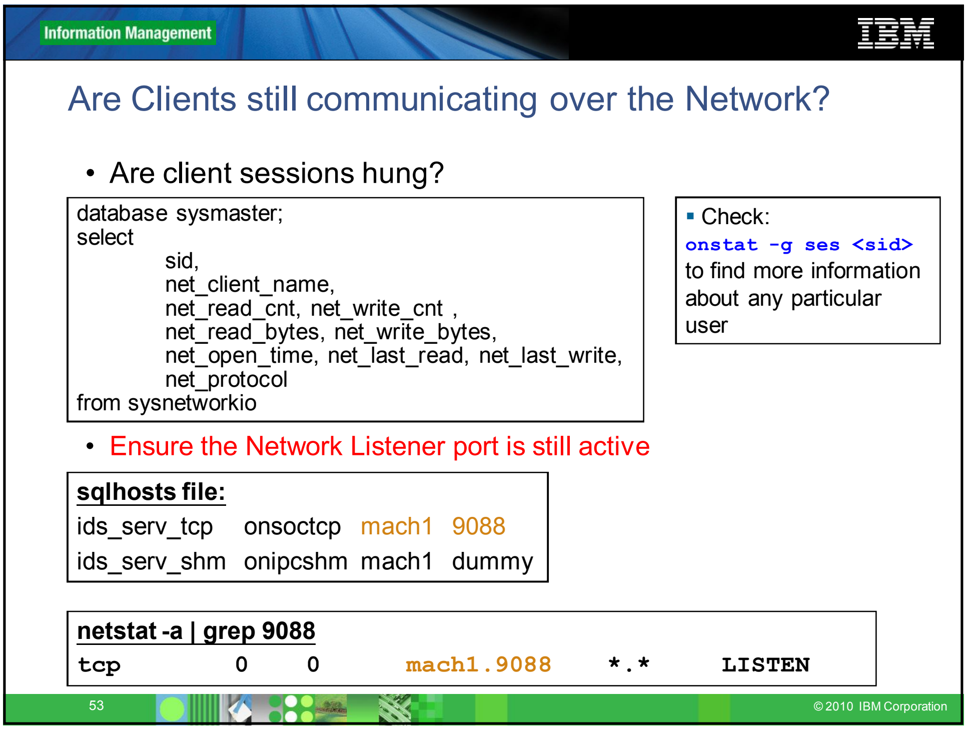
  response time seen at server

- Network Configuration Parameters at the Operating System Layer
  - Available Capacity?
  - # of TCP/IP Socket buffers?
  - Socket Buffer size?
  - Keepalive/Timeout interval?
  - Maximum Connection Requests?



Is Client-Server Database Traffic High?

- Very high client-server network traffic
  - Client Load utility, bulk data extraction, LOB manipulation, very high rate OLTP, etc.
  - Configuration issues, e.g. mismatched Ethernet transmission rates
  - External factors such as other activity on shared LAN
- Network bottlenecks typically arise from very high volumes of data being moved around – very large result sets, client load, etc.
- Can client logic be pushed onto the server to reduce traffic?
  - Stored procedures?
  - Datablades – embed application logic?
  - More sophisticated SQL? E.g. predicates to filter result set at the server
  - Above can potentially ADD some additional server CPU cost



Are Clients still communicating over the Network?

- Are client sessions hung?

```
database sysmaster;
select
        sid,
        net_client_name,
        net_read_cnt, net_write_cnt ,
        net_read_bytes, net_write_bytes,
        net_open_time, net_last_read, net_last_write,
        net_protocol
from sysnetworkio
```

- Check:
  `onstat -g ses <sid>`
  to find more information about any particular user

- Ensure the Network Listener port is still active

sqlhosts file:

```
ids_serv_tcp   onsoctcp  mach1  9088
ids_serv_shm   onipcshm  mach1  dummy
```

netstat -a | grep 9088

```
tcp           0      0          mach1.9088      *.*       LISTEN
```

Are the Network Buffers Under-Configured?

- Have the network buffer queues been exceeded?
  - `onstat -g ntu / onstat -g ntm`

Number of times the free buffer threshold was exceeded

```
global network information:
#netscb connects    read   write  q-limits    q-exceed   alloc/max
  6/  7     58        622   2626  2818/   10  10/  0      0/  0
Individual thread network information (basic):
  netscb type    thread name      sid    fd poll     reads    writes q-nrm q-
exp
 3fc97bc soctcp sqlexec        5269    66    5        38        38  0/ 1  0/
0
```

- Sizing network buffers to accommodate a typical request can improve CPU utilization by eliminating the need to break up a request into multiple messages
- Instead of IDS dynamically satisfying requests network buffer request using the global buffer pool, possible to configure the free buffer thresholds and the size of each buffer ➔

- NETTYPE configuration parameter
- IFX_NETBUF_PVTPOOL_SIZE environment variable
- IFX_NETBUF_SIZE environment variable, and b (client buffer size) option in the sqlhosts file or registry

Is There a High Number of Database Connection Requests?

- Are there a significant number of new connections in a short period?
  - `onstat -g ntu`

Example: 110 connections in 19 seconds

```
IBM Informix Dynamic Server Version 9.40.FC6W1  -- On-Line -- Up 2 days 15:55:26
c000000059d931c8 soctcp soctcplst     21    6   10   533103       0  0/ 0  0/ 0  0/ 0
IBM Informix Dynamic Server Version 9.40.FC6W1  -- On-Line -- Up 2 days 15:55:45
c000000059d931c8 soctcp soctcplst     21    6   10   533213       0  0/ 0  0/ 0  0/ 0
```

- For systems that can have 200 or more concurrent network users, better performance might result from adding more poll threads
- Improve connection throughput for a given interface/protocol combination
  - Allocate additional listener threads (add a new **DBSERVERALIAS**)
  - Add another network-interface card

Is User Authentication the Bottleneck?

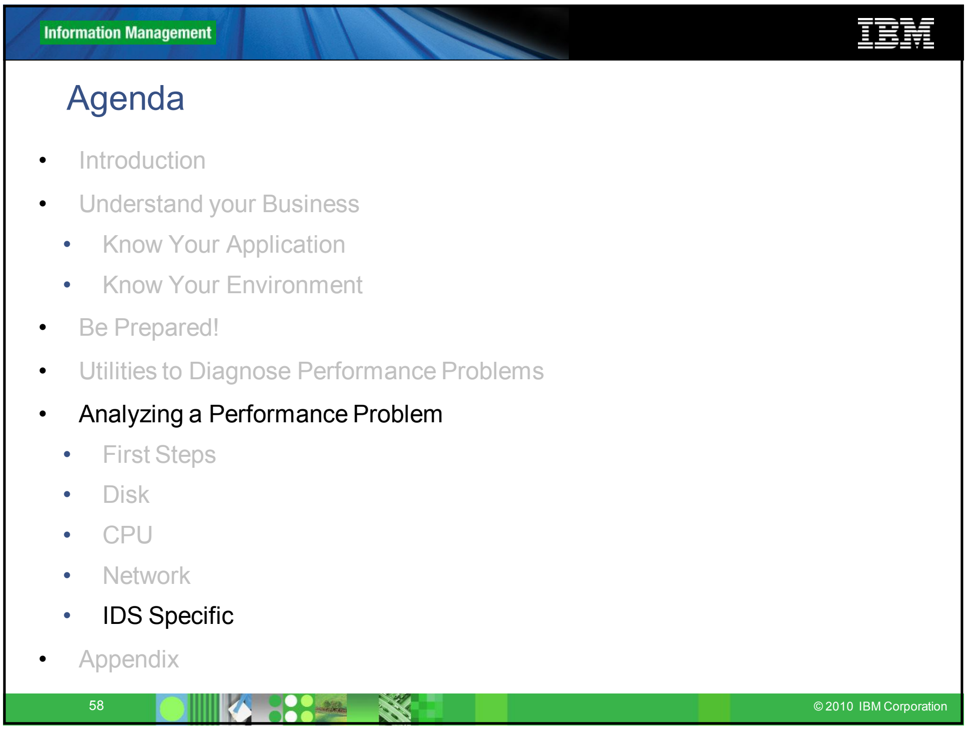
- Are MSC VPs unable to keep up with new connection request?
  - The MSC VP handles network authentication (e.g. gethostbyname)
- Check `onstat -g glo` (*Eff* column)
  - If at or close to 100% effective, try adding another MSC VP
- Beginning with IDS 10, can have multiple MSC VP’s
  - Format for VPCLASS in the ONCONFIG
    - `VPCLASS msc,<#>,noage`
  - Add another MSC VP dynamically
    - `onmode -p +1 msc`

Are Network Connections Timing Out or Rejected?

- Monitor using

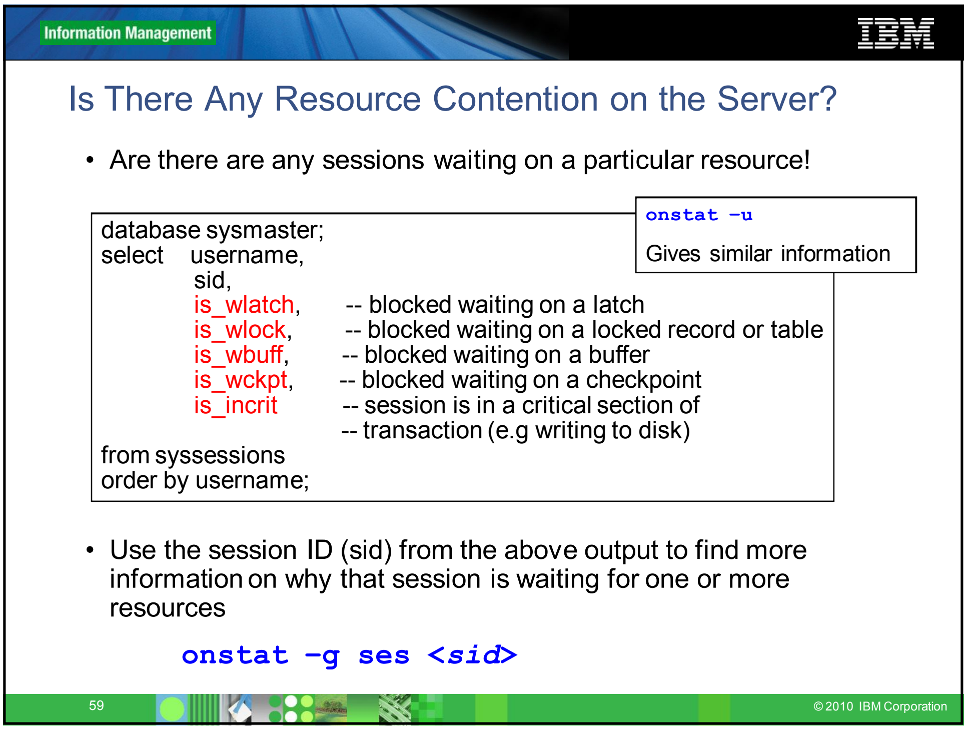
  `onstat -g ntd`
- Check the number of *accepted* vs. *rejected* connections
- If there are a large number of rejections then either:
  - The user table has overflowed (`onstat -p: ovuserthreads`)
  - The network is timing out on the connection
- Consult your System Administrator

```
global network information:
  #netscb connects     read    write   q-free  q-limits  q-exceed
   5/  7       54    9938    41616    1/   1   10/  10     0/   0
Client Type    Calls  Accepted  Rejected      Read      Write
sqlexec          yes        35         7       714        747
srvinfx          yes         0         0         0          0
onspace          yes         0         0         0          0
onlog            yes         0         0         0          0
onparam          yes         0         0         0          0
oncheck          yes        18         0      9107      40806
onload           yes         0         0         0          0
onunload         yes         0         0         0          0
onmonitor        yes         1         0        63         63
dr_accept        yes         0         0         0          0
ontape           yes         0         0         0          0
srvstat          yes         0         0         0          0
asfecho          yes         0         0         0          0
listener         yes         0         0        54          0
crsamexec        yes         0         0         0          0
safe             yes         0         0         0          0
Totals                      54         7      9938      41616
```



Agenda

- Introduction
- Understand your Business
  - Know Your Application
  - Know Your Environment
- Be Prepared!
- Utilities to Diagnose Performance Problems
- Analyzing a Performance Problem
  - First Steps
  - Disk
  - CPU
  - Network
  - IDS Specific
- Appendix



Is There Any Resource Contention on the Server?

- Are there are any sessions waiting on a particular resource!

```
onstat –u

Gives similar information
```

```
database sysmaster;
select    username,
          sid,
          is_wlatch,       -- blocked waiting on a latch
          is_wlock,        -- blocked waiting on a locked record or table
          is_wbuff,        -- blocked waiting on a buffer
          is_wckpt,        -- blocked waiting on a checkpoint
          is_incrit        -- session is in a critical section of
                           -- transaction (e.g writing to disk)
from syssessions
order by username;
```

- Use the session ID (sid) from the above output to find more information on why that session is waiting for one or more resources

```
onstat –g ses <sid>
```

Are there any Blocking Locks?

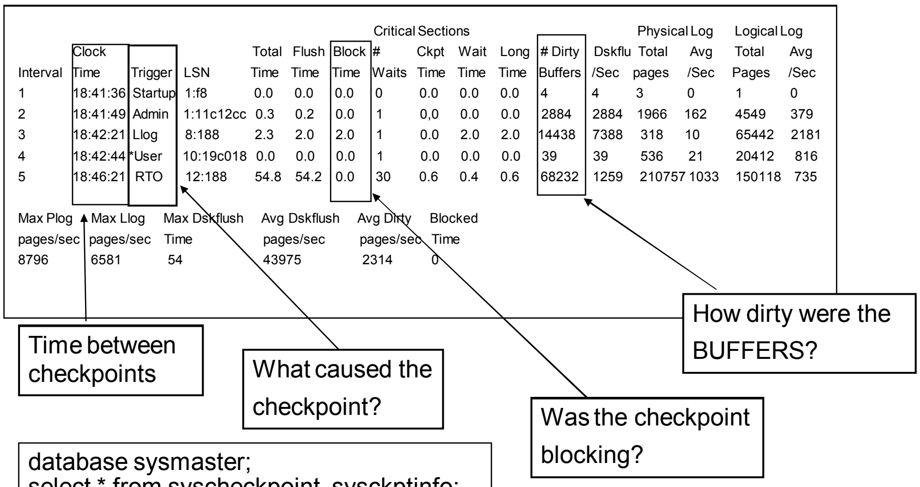
- Find which Locks are blocking other users?
  - Including who those other users are
- Similar to **onstat -k**
- Use the *sessionid* from the results of this query to find what the User, who is holding the lock(s) that other sessions are waiting for, is doing

  onstat -g ses <sessionid>

```
database sysmaster;
select       t2.dbsname database,
             t5.sid SessionID,
             t8.txt type,
             t2.tabname table,
             t4.sid lock_sess,
             t5.username lock_user,
             t6.sid wait_sess,
             t7.username wait_user,
             t1.rowidr row_id,
             t1.keynum
from         sysmaster:'informix'.syslcktab t1,
             sysmaster:'informix'.systabnames t2,
             sysmaster:'informix'.systxptab  t3,
             sysmaster:'informix'.sysrstcb t4,
             sysmaster:'informix'.sysscblst t5,
             sysmaster:'informix'.sysrstcb t6,
             sysmaster:'informix'.sysscblst t7,
             sysmaster:'informix'.flags_text t8
where        t1.owner = t3.address
  AND        t1.partnum = t2.partnum
  AND        t3.owner = t4.address
  AND        t1.wtlist =    t6.address
  AND        t4.sid = t5.sid
  AND        t8.tabname = 'syslcktab'
  AND        t8.flags    = t1.type
  AND        t6.sid = t7.sid
```

Are There Any Long or Blocking Checkpoints?

onstat –g ckp



```
Auto Checkpoins=On   RTO_SERVER_RESTART=60 seconds   Estimated recovery time 7 seconds

                                                            Critical Sections                    Physical Log       Logical Log
            Clock                         Total   Flush   Block  #       Ckpt   Wait   Long   # Dirty  Dskflu  Total    Avg     Total    Avg
Interval    Time      Trigger   LSN       Time    Time    Time   Waits   Time   Time   Time   Buffers  /Sec    pages    /Sec    Pages    /Sec
1           18:41:36  Startup   1:f8      0.0     0.0     0.0    0       0.0    0.0    0.0    4        4       3        0       1        0
2           18:41:49  Admin     1:11c12cc 0.3     0.2     0.0    1       0,0    0.0    0.0    2884     2884    1966     162     4549     379
3           18:42:21  Llog      8:188     2.3     2.0     2.0    1       0.0    2.0    2.0    14438    7388    318      10      65442    2181
4           18:42:44 *User      10:19c018 0.0     0.0     0.0    1       0.0    0.0    0.0    39       39      536      21      20412    816
5           18:46:21  RTO       12:188    54.8    54.2    0.0    30      0.6    0.4    0.6    68232    1259    210757 1033     150118   735

Max Plog   Max Llog   Max Dskflush  Avg Dskflush  Avg Dirty  Blocked
pages/sec  pages/sec  Time          pages/sec     pages/sec  Time
8796       6581       54            43975         2314       0
```

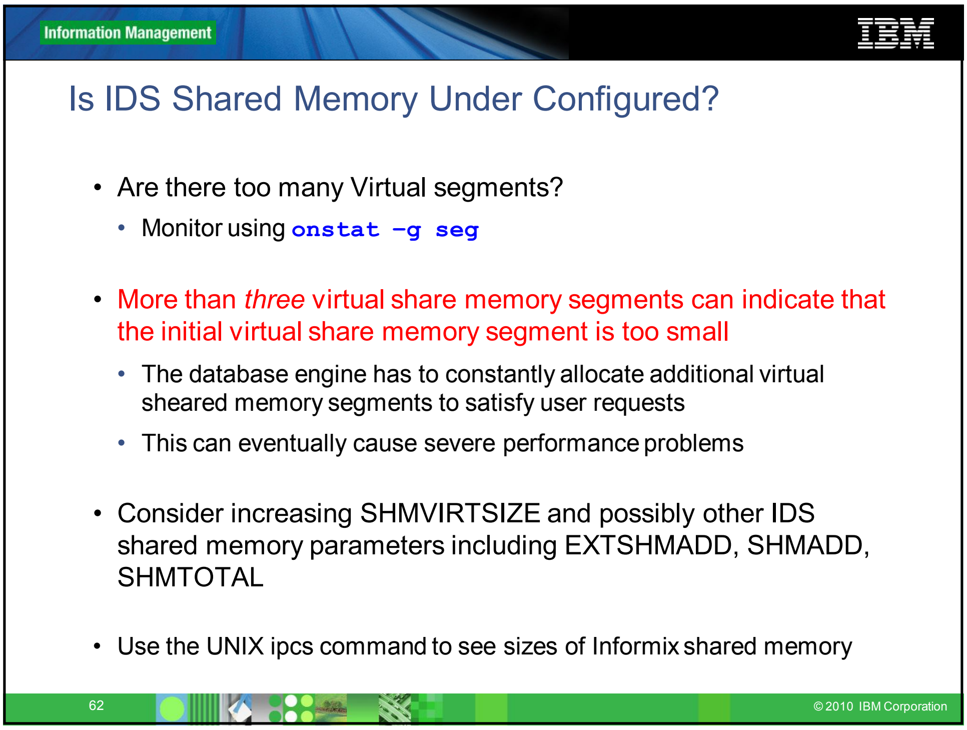
Time between checkpoints

What caused the checkpoint?

Was the checkpoint blocking?

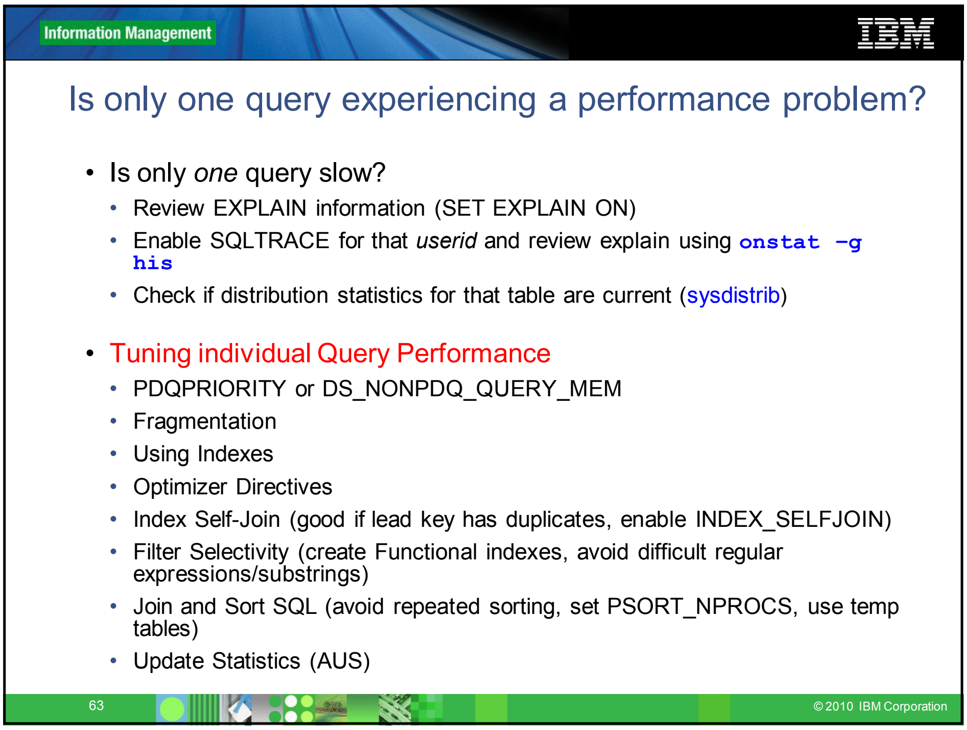
How dirty were the BUFFERS?

database sysmaster;
select * from syscheckpoint, sysckptinfo;



Is IDS Shared Memory Under Configured?

- Are there too many Virtual segments?
  - Monitor using `onstat -g seg`
- More than *three* virtual share memory segments can indicate that the initial virtual share memory segment is too small
  - The database engine has to constantly allocate additional virtual sheared memory segments to satisfy user requests
  - This can eventually cause severe performance problems
- Consider increasing SHMVIRTSIZE and possibly other IDS shared memory parameters including EXTSHMADD, SHMADD, SHMTOTAL
- Use the UNIX ipcs command to see sizes of Informix shared memory



Is only one query experiencing a performance problem?

- Is only *one* query slow?
  - Review EXPLAIN information (SET EXPLAIN ON)
  - Enable SQLTRACE for that *userid* and review explain using `onstat -g his`
  - Check if distribution statistics for that table are current (sysdistrib)
- Tuning individual Query Performance
  - PDQPRIORITY or DS_NONPDQ_QUERY_MEM
  - Fragmentation
  - Using Indexes
  - Optimizer Directives
  - Index Self-Join (good if lead key has duplicates, enable INDEX_SELFJOIN)
  - Filter Selectivity (create Functional indexes, avoid difficult regular expressions/substrings)
  - Join and Sort SQL (avoid repeated sorting, set PSORT_NPROCS, use temp tables)
  - Update Statistics (AUS)

Are Many Queries Experiencing Performance Problems?

- Are most or All Queries slow?
  - *First*, eliminate external factors – disk, network, cpu, memory
  - Enable SQLTRACE globally and review explains using `onstat -g his`
  - Are Statistics out of date?
    - Check when Update Statistics was last run? (*constructed* column)
  - Enable EXPLAIN_STAT configuration parameter to monitor statistics on queries
    - Use SET EXPLAIN or onmode –Y <*sessionid*> to create Query Statistics
  - Consider using External Optimizer Directives
    - Allows DBA to modify the query plan without changing the application
    - Directives not required to be in the application
    - To create external directives use the `SAVE EXTERNAL DIRECTIVE` statement

Is a Database Utility Experiencing Performance Issues?

- Update Statistics?
  - Ensure PDQ (> 0) to to allocate more memory to it
  - DBUPSPACE – is it constraining?
- Index Build?
  - Ensure PDQPRIORITY > 0
  - Set DBSPACETEMP to allocate multiple Temp spaces for sorting, temp tables, etc
  - Set PSORT_NPROCS to perform sorting in parallel
- Backup or Restore?
  - Both ontape and onbar use archive transport buffers in the IDS virtual segments
  - For ontape, the number of transport buffers are fixed
  - For onbar, the number of transport buffers are configurable
    - **BAR_MAX_BACKUP**
    - **BAR_NB_XPORT_COUNT**
    - **BAR_XFER_BUF_SIZE** (set to max allowed)
    - Monitor using `onstat -g stq`

Questions