Monitoring and Tuning IBM Informix IDS Server

Lester Knutsen

Advanced DataTools

Areas to Monitor and Tune

- CPU Usage – How busy are the CPUs?
- Memory Usage – How much memory is being used?
- Disk Usage – What is the disk I/O throughput?
- Network Usage – What is the network utilization?

Unix Tools we will use to Monitor Performance

- SAR – System Activity Recorder
- VMSTAT – CPU and virtual memory statistics
- MPSTAT – Per-CPU statistics
- IOSTAT – Disk I/O throughput statistics
- VXSTAT – Veritas Volume Manager statistics
- PS – Unix processes statistics
- TOP – Top Unix processes statistics
- PSTAT - Top Solaris processes statistics
- NETSTAT – Network statistics

Informix Tools we will use to Monitor Performance

- ONSTAT – Shared memory server statistics
- SYSMASTER DATABASE – Shared memory server statistics
- Server Studio (New 4.0) – Command and control center for Informix Server

SAR – System Activity Recorder

- Setup as a cron job to collect statistics and saves them to a file /var/adm/sa/sa??
- Example Cron setup to collect data every 15 minutes:

  0,15,30,45 * * * * /usr/lib/sa/sa1
- SAR command displays the data collected
- Can also be run in real-time:

  sar 5 5

SAR Reporting Options

-a Report use of file access system routines
-b Report buffer activity
-c Report system calls
-d Report activity for each device (disk or tape drive)
-g Report paging activities
-k Report kernel memory allocation (KMA) activities
-m Report message and semaphore activities
-p Report paging activities
-q Report average queue length
-r Report unused memory pages
-u Report CPU utilization (the default)
-v Report status of process, i-node, file tables
-w Report system swapping and switching activity
-y Report TTY device activity
-A Report all data. Equivalent to -abcdgkmpqruvwy

SAR Collection Options

- -i sec - Select data at intervals as close as possible to sec seconds.
- -s time - Select data later than time in the form hh[:mm]. Default is 08:00.
- -f filename - Use filename as the data source for sar. Default is the current daily data file /var/adm/sa/sadd.
- -o filename - Save samples in file, filename, in binary format.

SAR – Default Output

00:00:00	%usr	%sys	%wio	%idle
07:00:00	27	3	0	70
07:15:02	61	6	0	33
07:30:01	47	4	0	49
07:45:01	28	3	0	70
08:00:00	30	2	0	68
08:15:00	50	3	0	46
08:30:01	56	3	0	41
08:45:00	22	2	0	77

VMSTAT – CPU and Memory

- Options:

  vmstat [-cipsS] [disks] [interval [count]]

- Example:

```
lester@ >vmstat 5 5
 procs     memory            page                    disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr s0 s1 s2 s3   in   sy   cs us sy id
 0 0 0 4350896 573168 0   0  0  0  0  0  0  0  1  1  7 4294967196 0 0 -5 -1 -104
 0 0 0 3749680 370888 106 68 0  0  0  0  0  0  0  1  0  237  835  839  9  1 90
 0 0 0 3748784 369728 3   3  0  0  0  0  0  0  0  1  0  233  368  728 25  0 75
 0 0 0 3748816 369760 1   1  0  0  0  0  0  0  0  0  0  233  287  692 25  0 75
 0 0 0 3748816 369760 0   0  0  0  0  0  0  0  0  1  0  226  278  715  9  2 89
```

MPSTAT – Per-CPU Statistics

- Options:

  mpstat [-p | -P set] [interval [count]]
- Example on a 4 CPU machine:

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
44	1	0	20	304	200	115	10	19	13	0	60	51	0	0	49
45	1	0	16	5	2	240	2	46	10	0	107	5	1	0	94
46	0	0	5	6	1	188	4	38	6	0	82	21	0	0	79
47	0	0	2	9	1	181	7	39	8	0	69	24	0	0	76
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
44	1	0	28	303	200	133	8	22	13	0	69	39	0	0	61
45	0	0	2	11	2	182	7	38	6	0	78	27	0	0	72
46	0	0	4	8	1	191	5	42	7	0	74	9	8	0	83
47	0	0	7	12	1	175	9	38	9	0	77	33	0	0	67

IOSTAT – Disk I/O Statistics

- Options:

  iostat [-cCdDeEImMnpPrstxz] [-l n] [-T u | d] [disk ...] [interval [count]]

- Example:

```
iostat 5 5
   tty         sd0           sd1           sd2           sd3           cpu
 tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv   us sy wt id
   0   33   0   0    1   58   1   11   20   1   10   38   7    2    5  1  0 94
   0   47   0   0    0    0   0    0    3   0    5    5   3    3    2  1  0 96
   0   16   0   0    0    0   0    0    3   0    4    0   0    0    0  0  0 100
   0   16   0   0    0    0   0    0    0   0    3    0   0    0    3  0  0 96
   0   16   0   0    0    0   0    0   28   4    7    0   0    0    1  0  0 98
```

VXSTAT – Veritas Volume Manager Statistics

- Part of Veritas Volume Manager
- To display disk statistics, use the vxstat -d command:

	OPERATIONS		BLOCKS		AVG TIME(ms)	
TYP NAME	READ	WRITE	READ	WRITE	READ	WRITE
dm disk01	40473	174045	455898	951379	29.5	35.4
dm disk02	32668	16873	470337	351351	35.2	102.9
dm disk03	55249	60043	780779	731979	35.3	61.2
dm disk04	11909	13745	114508	128605	25.0	30.7

PS – Unix Processes Statistics

- Key Options:
  - -e List information about every process now running.
  - -f Generate a full listing.
  - -l Generate a long listing.
  - -P Print the number of the processor to which the process or lwp is bound.
  - -t term List only process data associated with term.
  - -u uidlist List only process data whose effective user ID number or login name is given in uidlist.
  - -U uidlist List information for processes whose real user ID numbers or login names are given in uidlist.

PS – Unix Processes Examples

```
lester@atlas >ps -fu informix | more
     UID   PID  PPID  C    STIME TTY      TIME CMD
informix   416     1  0   Apr 17 ?       0:05 oninit -yv
informix   418   417  0   Apr 17 ?       0:05 oninit -yv
informix   428     1  0   Apr 17 ?       0:11 oninit -yv
informix  4085  3984  0 14:45:38 pts/2    0:00 dbaccess
informix  3984  3966  0 14:44:03 pts/2    0:00 bash
informix  3927     1  0 14:23:31 ?       16:21 oninit
informix  3966   874  0 14:37:34 pts/2    0:00 -ksh

lester@atlas >ps -lu informix | more
 F S   UID   PID  PPID  C PRI NI     ADDR    SZ   WCHAN TTY      TIME CMD
 8 S   202   416     1  0  41 20        ?  17648       ? ?       0:05 oninit
 c S   202   418   417  0  41 20        ?  17647       ? ?       0:05 oninit
 8 S   202   428     1  0  40 20        ?  14792       ? ?       0:11 oninit
 8 S   202  4085  3984  0  41 20        ?    654       ? pts/2   0:00 dbaccess
 8 S   202  3984  3966  0  51 20        ?    311       ? pts/2   0:00 bash
 8 S   202  3927     1  0  41 20        ?  17389       ? ?      16:21 oninit
 8 S   202  3966   874  0  51 20        ?    236       ? pts/2   0:00 ksh
```

TOP – Top Unix Processes

```
last pid:  9146;  load averages:  1.76,  1.65,  1.61                  20:16:10
143 processes: 133 sleeping, 3 zombie, 5 stopped, 2 on cpu
CPU states: 87.6% idle,  9.9% user,  2.4% kernel,  0.0% iowait,  0.0% swap
Memory: 12G real, 1122M free, 3899M swap in use, 8K swap free

  PID USERNAME THR PRI NICE   SIZE   RES STATE   TIME    CPU COMMAND
 7928 root       7  58    0   28M   26M sleep   8:36  5.17% dsmc
 2553 informix   5  30  -10 3625M 2896M cpu17 457.3H  1.58% oninit
 2549 informix   5  59  -10 3625M 2952M sleep 502.2H  1.23% oninit
 2551 informix   5  51  -10 3625M 2907M sleep 613.5H  1.19% oninit
 2555 informix   5  51  -10 3625M 2888M sleep 373.4H  0.92% oninit
 2539 informix   5  59  -10 3625M 2959M sleep 496.5H  0.80% oninit
 2550 informix   5  59  -10 3625M 2935M sleep 684.9H  0.70% oninit
 9145 lester     1  50    0 2544K 2120K cpu16   0:01  0.61% top
 2552 informix   5  59  -10 3625M 2906M sleep 528.3H  0.59% oninit
 2554 informix   5  59  -10 3625M 2894M sleep 396.3H  0.52% oninit
 2329 root       1  58    0   13M 3040K sleep 579:30  0.02% jre
 9121 root       1  58    0 5112K 2264K sleep   0:00  0.02% bpsched
14191 root       1  48    0 5176K 2336K sleep   0:13  0.01% bpsched
 9114 lester     1  43    0 1648K 1200K sleep   0:00  0.01% ksh
 9117 root       1  48    0   10M 5808K sleep   0:00  0.01% bprd
```

PSTAT - Top Solaris Processes

```
  PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
 4424 lester   1616K 1424K cpu1    55    0   0:00.00 0.1% prstat/1
 4414 lester   1928K 1264K sleep   41    0   0:00.00 0.1% ksh/1
  853 nobody     43M   26M sleep   58    0   0:00.03 0.1% java/27
 4412 root     1840K 1328K sleep   54    0   0:00.00 0.0% in.telnetd/1
  407 informix  143M 1240K sleep   59  -10   0:00.00 0.0% oninit/1
  405 informix  143M 1576K sleep   59  -10   0:00.00 0.0% oninit/1
  406 informix  143M   12M sleep   59  -10   0:00.07 0.0% oninit/2
  762 root       952K  480K sleep   51    0   0:00.00 0.0% readproctitle/1
  376 root      2352K 1608K sleep   45    0   0:00.00 0.0% caspd/5
  389 root      1656K  792K sleep   31    0   0:00.00 0.0% cimomboot/1
  251 root      3040K 2368K sleep   52    0   0:00.00 0.0% nscd/7
  225 root      3824K 2008K sleep   59    0   0:00.00 0.0% automountd/5
  379 root        50M   19M sleep   58    0   0:00.00 0.0% caspeng/21
 2339 root      1976K 1264K sleep   48    0   0:00.00 0.0% cron/1
  257 root      3160K 1016K sleep   58    0   0:00.00 0.0% lpsched/1
  388 root      1064K  672K sleep   59    0   0:00.00 0.0% utmpd/1
 3527 root      3696K 1960K sleep   58    0   0:00.00 0.0% syslogd/13
  224 root      2224K 1432K sleep   48    0   0:00.00 0.0% inetd/1
  404 informix  143M  129M sleep   59  -10   0:00.12 0.0% oninit/2
  168 root      4608K 2136K sleep   58    0   0:00.03 0.0% skipd/1
   56 root      2232K 1192K sleep   53    0   0:00.00 0.0% syseventd/9
Total: 117 processes, 586 lwps, load averages: 0.02, 0.03, 0.04
```

NETSTAT – Network Statistics

- Options:

```
usage: netstat [-anv] [-f address_family]
       netstat [-g | -p | -s] [-n] [-f address_family] [-P protocol]
       netstat -m
       netstat -i [-I interface] [-an] [-f address_family] [interval]
       netstat -r [-anv] [-f address_family]
       netstat -M [-ns] [-f address_family]
       netstat -D [-I interface] [-f address_family]
```

- Example

```
lester@atlas >netstat -i
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
hme0  1500 atlas.addt.com atlas.addt.com 92751  0     50571  0     0      0
lo0   8232 loopback      localhost       80430  0     80430  0     0      0
```

CPU Monitoring

- Are the CPUs overloaded?
- Factors:
  - Number of CPUs
  - Speed of CPUs (old vs new systems)
  - Number of process needing CPU time.

How Busy are the CPU's?

- Tools to monitor:
  - sar –u
  - vmstat
  - mpstat
  - top, prstat
- Performance Guideline - % CPU busy:
  - < 30 % - Good
  - 30-60% - Fair
  - > 60% - Poor

SAR – Example

00:00:00	%usr	%sys	%wio	%idle	
07:00:00	27	3	0	70	
07:15:02	61	6	0	33	Poor
07:30:01	47	4	0	49	
07:45:01	28	3	0	70	
08:00:00	30	2	0	68	
08:15:00	50	3	0	46	Fair
08:30:01	56	3	0	41	
08:45:00	22	2	0	77	Good

How many process are waiting to run on the CPUs?

- Tools to monitor Load Average:
  - sar –q
  - Uptime
- Performance Guideline – number of waiting processes:
  - < 2 per CPU – Good
  - 2-4 per CPU – Fair
  - > 4 per CPU – Poor

CPU Load Average Example:

```
lester@atlas >uptime
  9:58pm  up 2 day(s),  5:52,  4 users,  load average: 0.03, 0.04, 0.04
```

- Displays run queue over the last 1, 5, and 15 minutes
- On a 4 CPU machine:
  - < 2 x 4 = Good
  - 2-4 x 4 = Fair
  - > 4 x 4 = Poor

How many system calls per CPU per second?

- Tools to monitor System Calls:
  - sar –c
  - vmstat
- Performance Guideline – number of System Calls per CPU (depends on speed of CPU):
  - Fast CPU - > 20,000 poor
  - Medium CPU - > 10,000 poor
  - Slow CPU - > 2,000 poor

System Calls

- Example: sar –c

```
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
00:15:00   20606     935     299    2.52    2.06  844574  871668
00:30:00   20385     844     243    1.22    0.80  588041 1049094
00:45:00   16124    1626     812    2.54    1.74 2986193 3222280
01:00:00   16079    4983    2715    2.90    2.61 1528419  921029
01:15:00    8535    1371     478    3.66    3.55  952043  834463
01:30:00   12853    3361     535    2.80    2.09 2898274  515510
01:45:00   17618    4412     603    1.59    1.37 4723358 4413470
02:00:01   12697    1544     452    0.67    0.46 2231396 1903038
02:15:01   13821    2727     572    1.40    1.22 3362608 3644032
02:30:00   14959    4802     727    1.23    0.94 3710160 2934010
02:45:00   14583    1581     247    0.71    0.62 1567575 1431916
```

Memory Monitoring

- Is memory being over-used or under-used?
- Memory shortage causing swapping to disk.
- Factors:
  - Amount of RAM
  - 32 bit vs 64 bit OS and applications
  - 32 bit Informix IDS limited to:
    - 3.6 GB on Solaris
    - 2 GB on AIX
    - 2 GB on Windows
- One of best Informix IDS performance improvements is adding BUFFERS

How much Memory is Used?

- Tools to monitor
  - top
  - sar –r
  - vmstat
- Performance Guidelines
  - Don’t monitor free memory since a good OS will use all extra memory as file system cache
  - Monitor swap space and paging in/outs

Memory – Key is to Monitor Paging In/Out

- Monitor vmstat:
  - pi - kilobytes paged in
  - po - kilobytes paged out

- Monitor sar –g
  - pgout/s - page-out requests per second.
  - ppgout/s - pages paged-out per second.
- Monitor sar –p
  - pgin/s - page-in requests per second.
  - ppgin/s - pages paged-in per second.

What Processes are Using the Most Memory?

- Tools to monitor – look at the SIZE column:
  - top
  - prstat
  - ps
- Performance Guideline for Informix:
  - BUFFERS - number of shared memory buffers
  - SHMVIRTSIZE - initial virtual shared memory segment size
  - SHMADD - size of new shared memory segments
  - SHMTOTAL – total size of shared memory

Monitoring Disks

- Goal is to balance I/O across all disks
  - Use: sar and iostat
- Find the FAST spot on the disk and locate key chunks there
- Find the optimal disk throughput
  - Use: pfread

  http://www.geocities.com/ahammau/informix/pfread.html

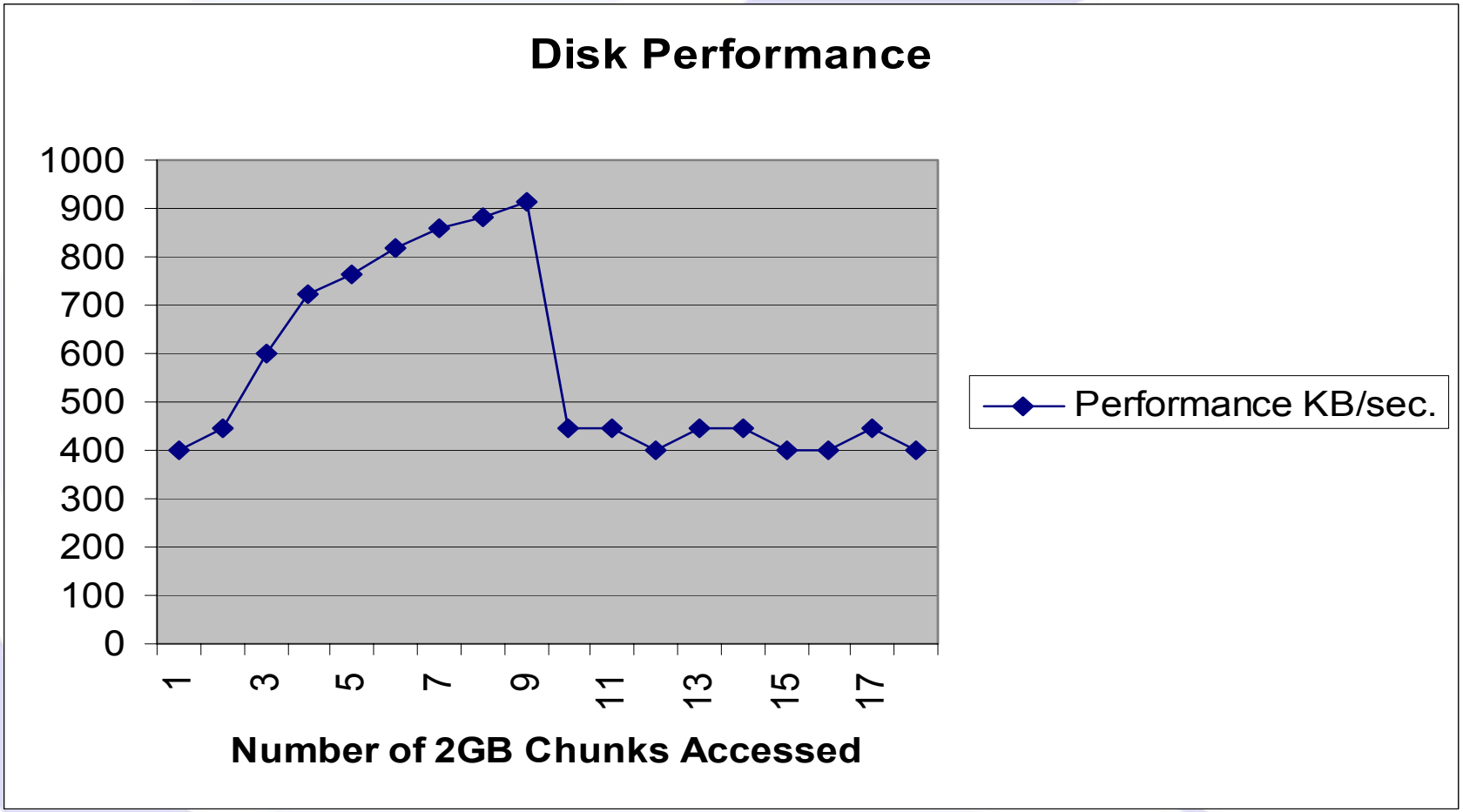
Disk Throughput

- Example pfread – 2 GB chunks on a 72 GB disk

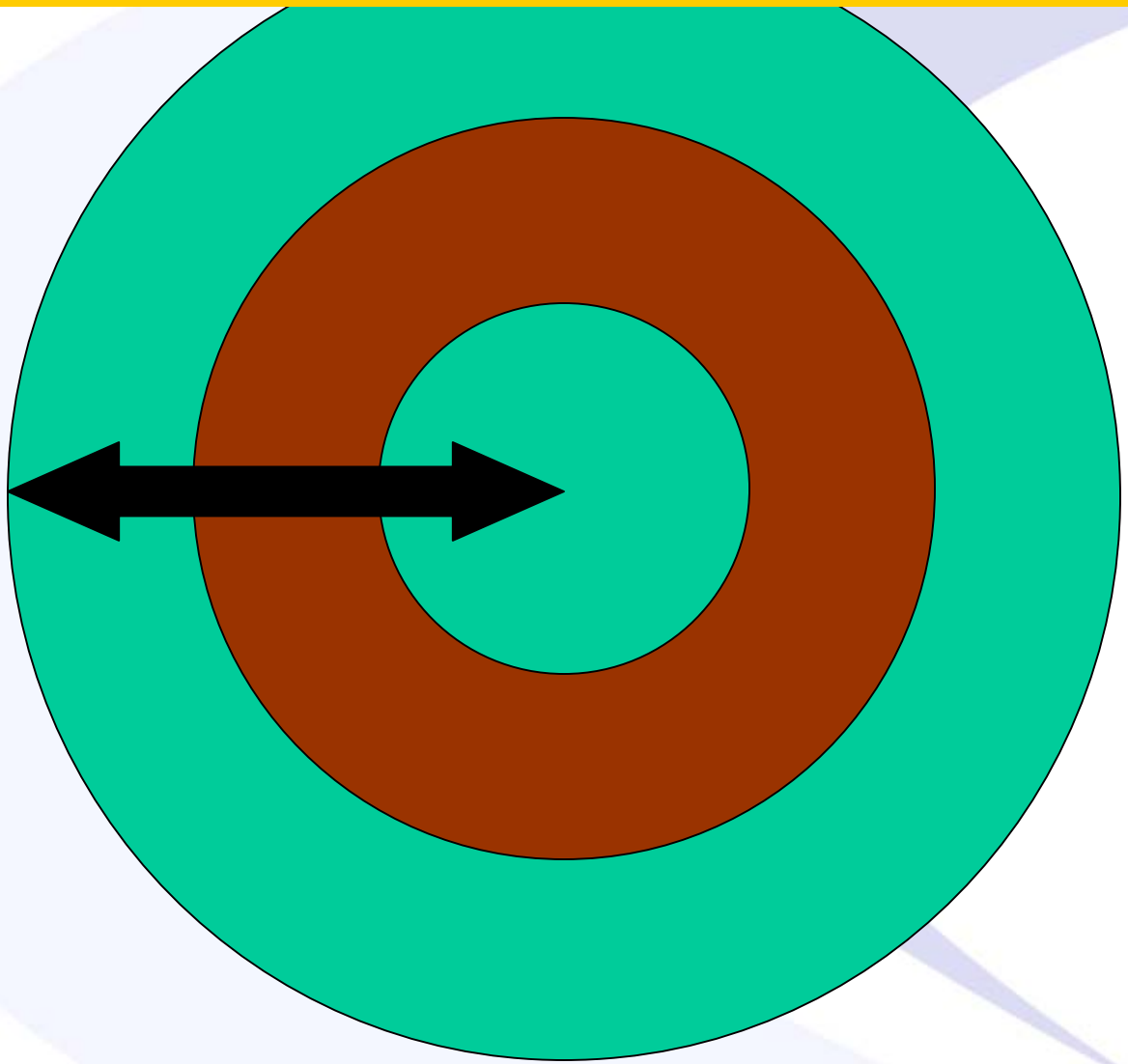
```
pfread.ksh 1 30 /informixchunks/d4chk14
/informixchunks/d4chk14 :       1 concurrent read threads       500 KB/sec.
/informixchunks/d4chk14 :       2 concurrent read threads       500 KB/sec.
/informixchunks/d4chk14 :       3 concurrent read threads       750 KB/sec.
/informixchunks/d4chk14 :       4 concurrent read threads       800 KB/sec.
/informixchunks/d4chk14 :       5 concurrent read threads       1000 KB/sec.
/informixchunks/d4chk14 :       6 concurrent read threads       996 KB/sec.
/informixchunks/d4chk14 :       7 concurrent read threads       1071 KB/sec.
/informixchunks/d4chk14 :       8 concurrent read threads       1082 KB/sec.
/informixchunks/d4chk14 :       9 concurrent read threads       1125 KB/sec.
/informixchunks/d4chk14 :       10 concurrent read threads      500 KB/sec.
/informixchunks/d4chk14 :       11 concurrent read threads      444 KB/sec.
/informixchunks/d4chk14 :       12 concurrent read threads      500 KB/sec.
```

- Best performance is using 9 x 2GB chunks = 18GB of the 72 GB disk

Disk Throughput – 36 GB Disk



Disk Layout - The ***FASTEST*** location on a disk is where the disk arm has to move the least to read or write data

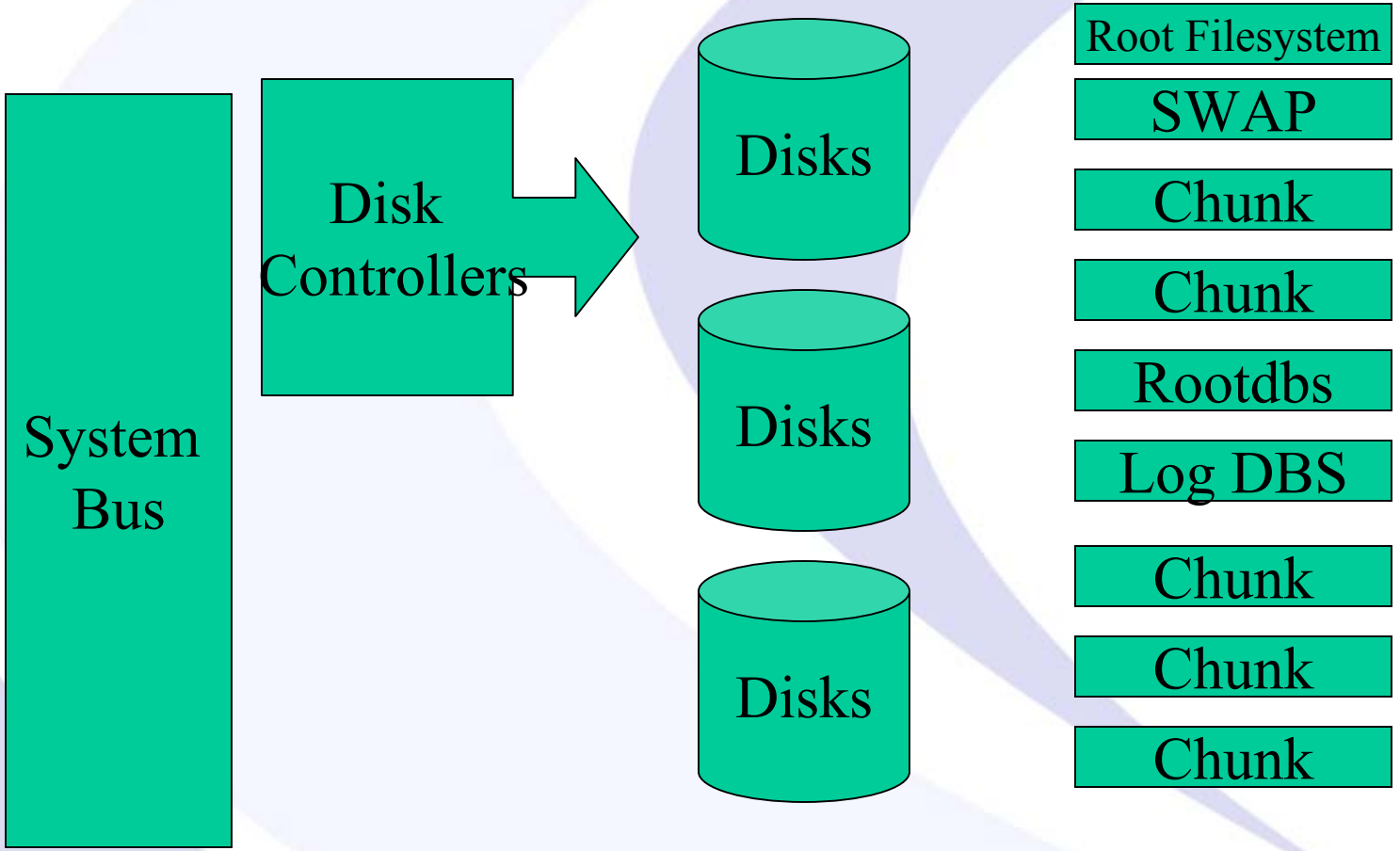


Monitor Disk I/O with SAR

- Report activity for each block device (disk or tape)
  - %busy – portion of time device was busy servicing a transfer request – How busy are your disks?
  - avque – average number of requests outstanding during that time.
  - read/s, write/s, blks/s - number of read/write transfers from or to device, number of bytes transferred in 512-byte units.
  - avwait - average wait time in milliseconds.
  - avserv - average service time in milliseconds.
- Example sar –d

```
00:00:00   device        %busy   avque   r+w/s  blks/s  avwait avserv
00:15:00   nfs1             0     0.0       0       0     0.0   0.0
           sd7             11     0.7      17     225     0.0   40.2
           sd7,a            0     0.0       0       0     0.0   0.0
           sd7,b            0     0.0       0       0     0.0   0.0
           sd7,c            0     0.0       0       0     0.0   0.0
           sd7,d            0     0.0       0       0     0.0   0.0
           sd7,e           11     0.7      17     225     0.0   40.2
```

Map Your Disk Drives



Create a Disk Layout Spreadsheet

- Controller/ Disk Array
- Disk
- Logical Volumes or Slices
- Chunks, Filesystems, etc…
- Tables in Chunks
- Compare results from sar -d and onstat -d

Disk Performance Spreadsheet

Disk Performance					
Disk Layout				Performance	
Controller	Disk	Volume	Chunk/Filesystem	onstat -d	sar -d
c1	disk1	d1v1			
c1	disk1	d1v2			
c1	disk1	d1v3			
c1	disk1	d1v4			
c1	disk2	d2v1			
c1	disk2	d2v2			

Monitoring Network

- How measure real output of network interface?
  - FTP Test – How long does it take to ftp a 2GB file to your destination? KB per second
  - Database server cannot send data out any faster than ftp
- Measure network errors and collusions
  - Netstat –i

Network Errors and Collisions

- Tool to monitor:
  - netstat –i
- Example output:

```
lester@atlas >netstat -i
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
hme0  1500 atlas.addt.com atlas.addt.com 102520 0     51764  0     0      0
lo0   8232 loopback      localhost       101386 0     101386 0     0      0
```

- Performance Guideline – no errors or collisions

Build Your Own Monitoring System

- Provide a baseline of performance information to compare to future problems
- Collect data from:
  - sysmaster
  - sar
- Load into a database for review and analysis
- Save historical data for future comparisons

Data Collection

- Create a cron job to run data collection scripts
  - Hourly/daily
  - Weekly
  - Monthly
- Build a database and load the data

Hourly Data Collection

- From sysmaster:

  lk_sesprof.sql – syssession – User statistics

- From onstat:

  onstat –g mgm – PDQ statistics

- From Unix:

  ps –ef – collect user statistics

  mpstat 5 5 – collect CPU statistics

Daily Data Collection

- From sysmaster:
  - lk_profile.sql – sysprofile – System statistics
  - lk_chkio.sql – syschktab – Chunk I/O
  - lk_dbsfree.sql – sysdbspaces, syschunks – Free space
  - lk_vpprof.sql – sysvplst – VP statistics
  - lk_tabprof.sql – sysptprof – Table I/O statistics
- From sar:
  - sar –u – CPU statistics
  - sar –b – Buffer statistics
  - sar –c – System calls
  - sar –d – Disk I/O statistics
  - sar –q – Run Queue statistics

Weekly/Monthly Data Collection

- From sysmaster:
  - lk_tablayout.sql - sysptnext, outer systabnames
  - lk_tabextent.sql - systabnames, sysptnext
  - lk_chkstatus.sql - sysdbspaces, syschunks
  - lk_idsconfig.sql - sysconfig
- Save configuration for future reference

Coming Soon – Scripts to Collect Data

- Check:

  http://www.advancedatatools.com/TechInfo/InformixInfo.html

Server Studio (New 4.0)

- Performance monitoring (~100 IDS metrics)
- Alerting via email, pager, cell phone
- Real-time graphing of selected performance parameters
- Built-in historical performance database
- Job scheduling (OS commands, shell scripts or SQL scripts)
- Automation of alert event responses
- Graphical tools for analysis of collected performance data
- SQL Capture and captured SQL analysis
- Load Testing (Benchmark Runner)
- Enhanced DB Difference Analyzer
- Database Synchronization Script Generation
- Enhanced SQL Editor
- Stored Procedure Debugger
- Virtual Processor Manager

Server Studio™ JE Sentinel

New

Performance Monitoring, Optimization and Autonomics for IBM IDS

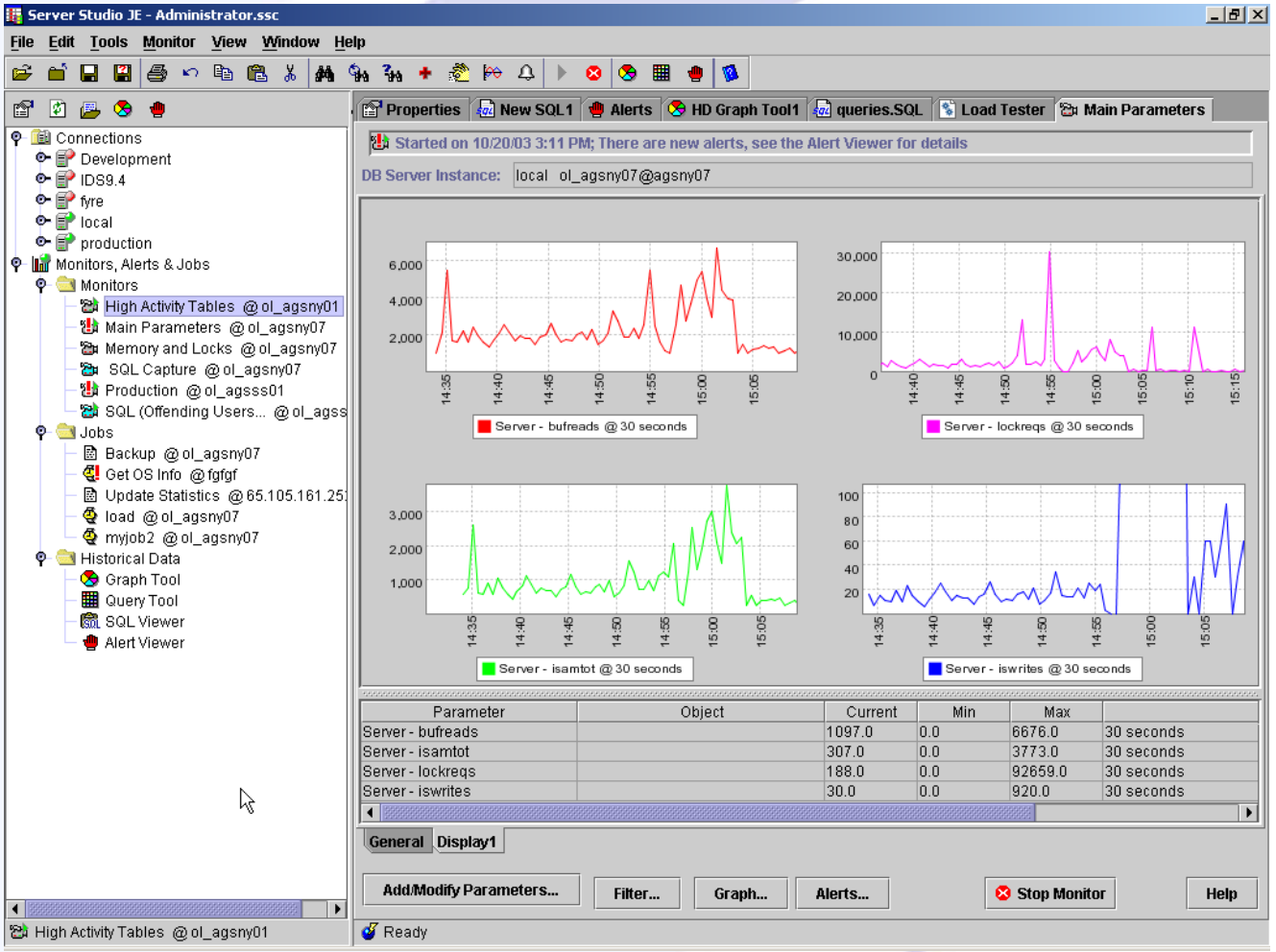
- **Monitor IDS Performance**
  - Real-time user-defined monitors created from a matrix of over 90 IDS performance parameters
  - Custom multi-level Alerts
  - Track SQL scripts from user sessions
- **Time-Series Performance Data Analysis**
  - Collect IDS historical performance matrix in the built-in time-series SQL database repository
  - Perform comprehensive correlation analysis on captured IDS performance parameters
- **Load Simulation for Tuning IDS**
  - Create real life load stress conditions scenarios
  - Run hundreds or even thousands of concurrent load-test virtual "user" sessions
- **IDS Event Management**
  - Autonomic response to IDS events by execution of:
    - *User-defined administration scripts*
    - *OS commands*
    - *SQL scripts*
    - *IDS native utilities*

Monitor and proactively manage your entire IBM IDS infrastructure from a unified graphical console

Server Studio™ JE Sentinel

IDS Real-Time Performance and Event Monitoring

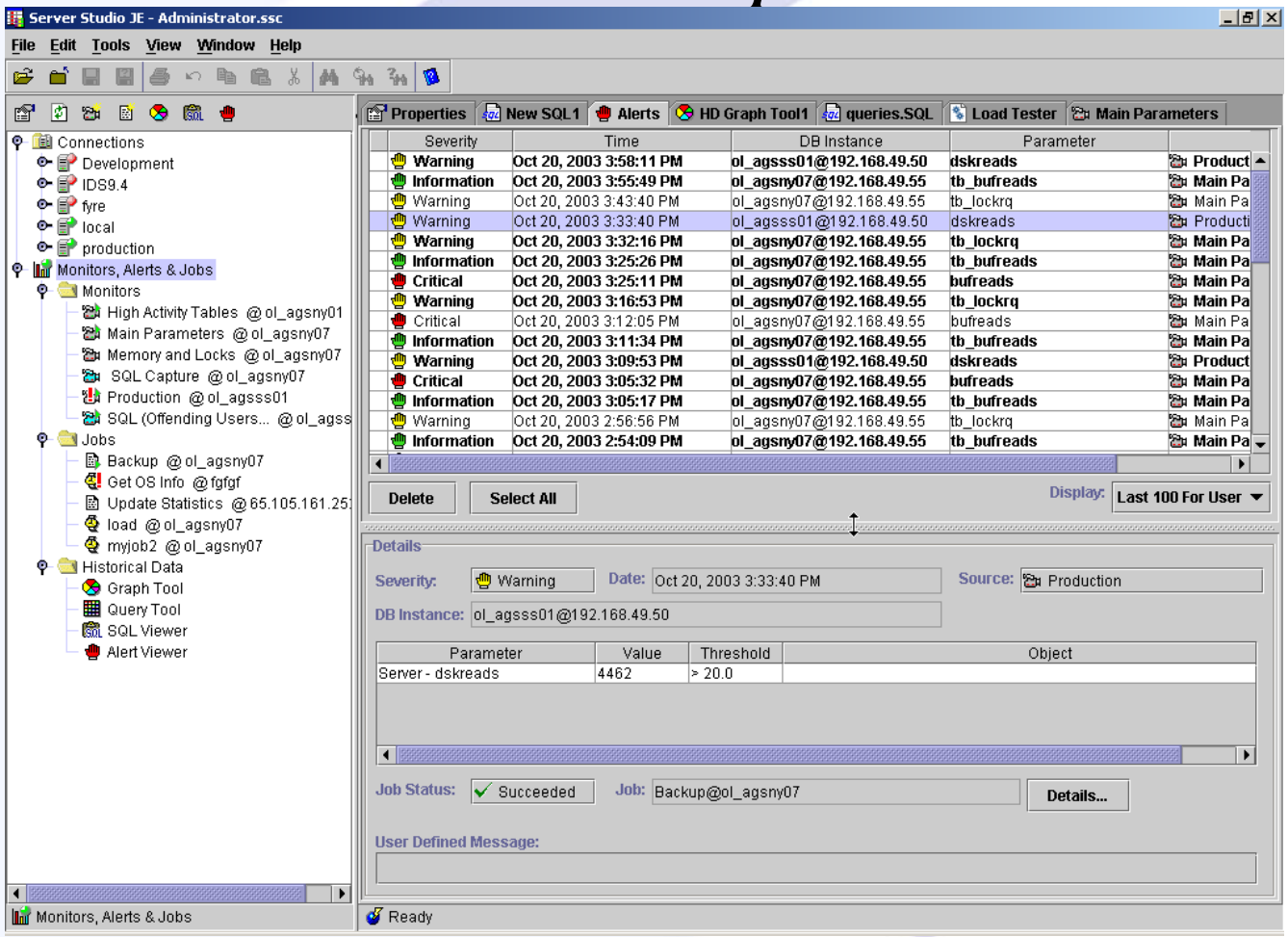
- Define customized performance monitors from over 90 available IDS parameters
- For each IDS instance, real-time monitors can be created at the level of:
  - Server
  - Chunk
  - Dbspace
  - Table
  - Index
  - Session
- Assign multi-level threshold Alerts to each parameter in a monitor



Server Studio™ JE Sentinel

IDS Event Alerts and Autonomic Response

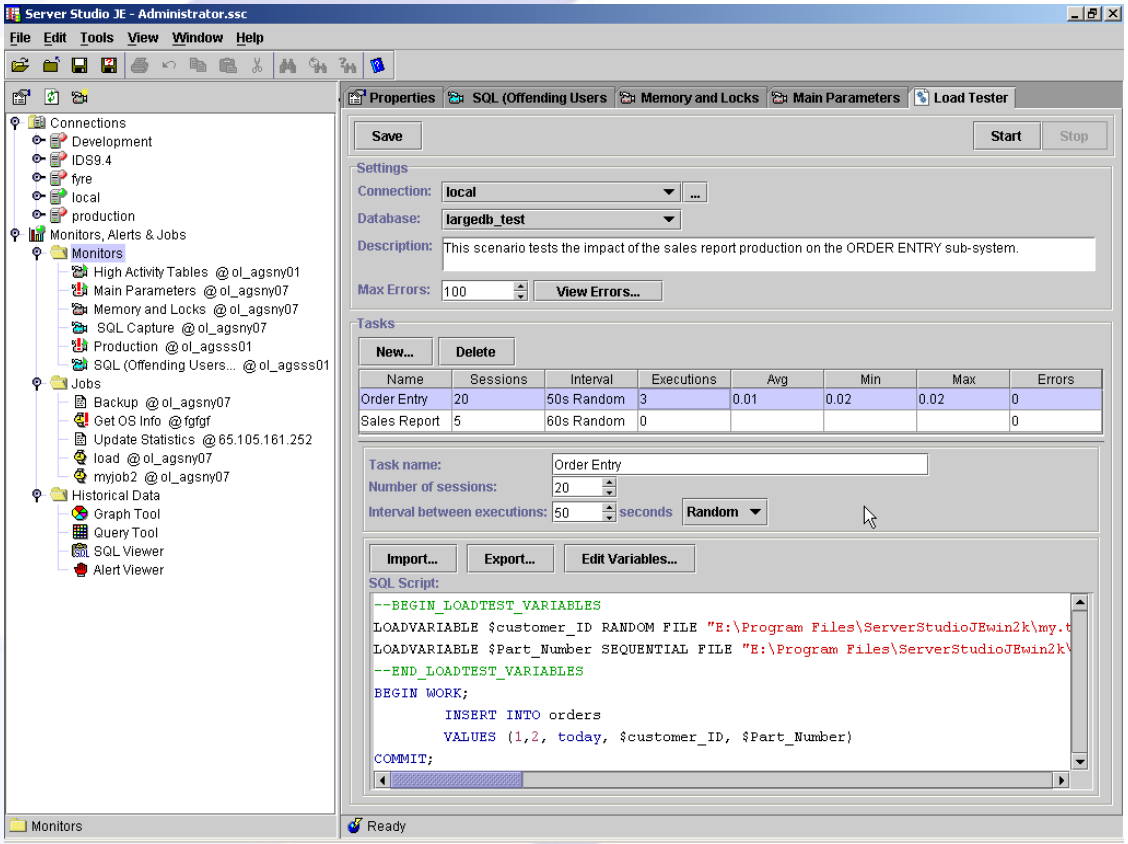
- Customized multi-level Alerts
- Centralized Viewer for Event Alerts
- Notification via email, pager, cell phone, etc.
- Autonomic response to alert events by:
  - User-defined administration scripts
  - OS commands
  - SQL scripts
  - Stored procedures
  - IDS native utilities
- Automation of regular maintenance tasks via scheduled custom-defined jobs.



Server Studio™ JE Sentinel

IDS Performance Simulation under Real Life Load Stresses

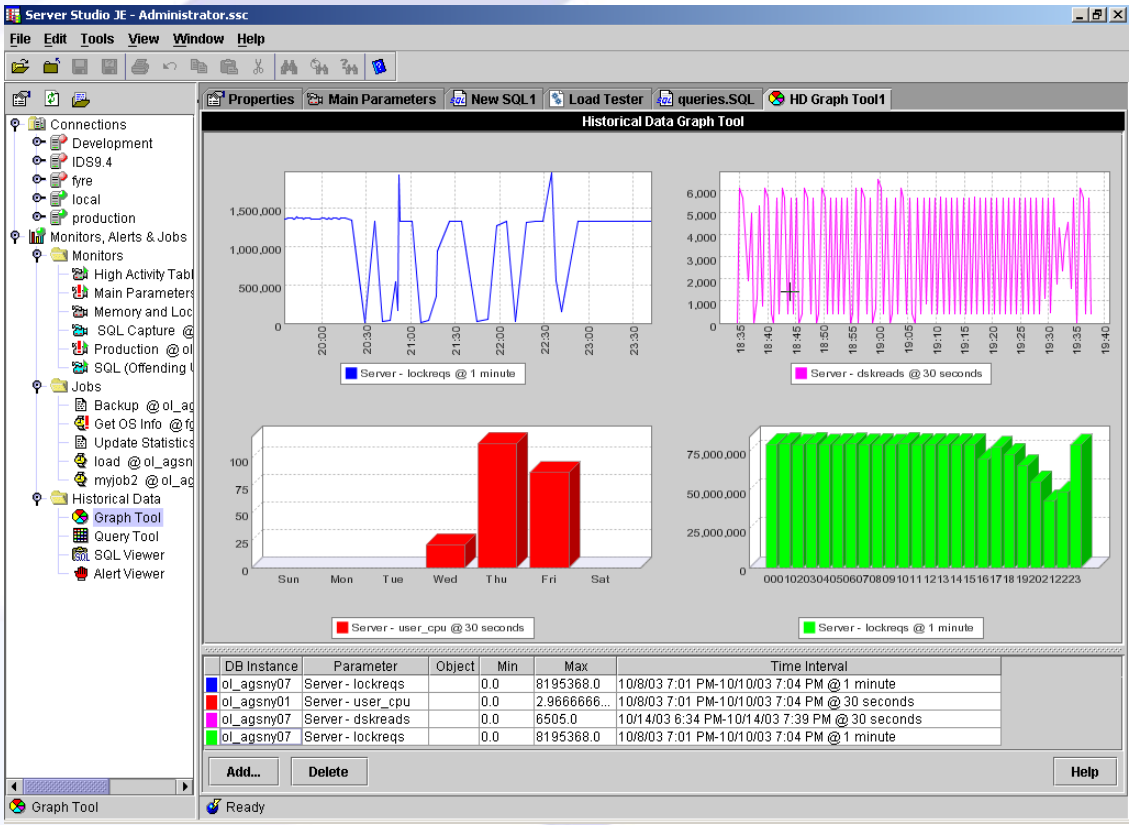
- To optimize your IDS performance, create real life stress conditions scenarios by incorporating into your load tests application-specific:
  - SQL scripts
  - Parametric queries
  - Data
- Run hundreds or even thousands of concurrent load-test virtual "user" sessions, using either the same or varying scenario scripts at random or programmed time intervals
- Automatically measures statistical user sessions' response times
- Initiate the monitors that will collect IDS performance parameters data matrix for the load-test scenario and record it into the built-in time-series repository for subsequent analysis



Server Studio™ JE Sentinel

Time-Series Analysis of IDS Performance Data

- For all IDS instances under monitoring, the time-series repository stores in its own built-in SQL database:
  - performance parameters data
  - Event Alerts history
  - SQL scripts captured from user sessions
  - Result sets captured from autonomic execution of administration scripts or IDS native utilities
- Correlation analysis can be easily performed over any available time interval to spot performance bottlenecks and anomalies
- Run SQL queries against the time-series repository to export complex performance parameters data sets for analysis in external programs



Server Studio™ JE 4.0 Coming Mid-May 2004

- Available at:

  www.ibm.com/software/data/informix/tools/serverstudio

  www.ServerStudio.com

  www.advancedatatools.com

- Advanced DataTools is an AGS Partner and Reseller for Server Studio

Thank You

Lester Knutsen

Email: Lester@advancedatatools.com

Web: http://www.advancedatatools.com

Phone: 703-256-0236 x102