



# SAN Stories - IO Performance

*Anjo Kolk*





## The source of the I/O problem ☺

---





## Storage Area Network (SAN)

---

“A storage area network is one or more devices communicating via a serial SCSI protocol (such as FC or iSCSI).”

Using SANs and NAS, W. Preston, O'Reilly



## Network Attached Storage (NAS)

---

“Network attached storage (NAS) is a computer (or device) dedicated to sharing files via NFS, CIFS, or DAFS.”

Using SANs and NAS, W. Preston, O'Reilly



## Comparing SAN and NAS

---

	SAN	NAS
Protocol	Serial SCSI-3	NFS/CIFS
Shares	Raw Disk	File Systems
Example	/dev/dsk/c0t0d0s2	\\filer\C\directory\file /nfsmount/directory
Allows	Different servers can access same disk	Different users can access same file or fs
Replaces	Locally attached disks	Replaces UNIX NFS and NT CIFS servers



## “How about those SANs?”

---

- ▶ SANs are expensive
- ▶ SANs are complex
- ▶ SANs are getting more popular
  - Storage requirements are growing (exponentially)
- ▶ SANs are black boxes
- ▶ SANs need expensive HBAs
- ▶ SANs require (large) caches
  - How will multiple systems share the same SAN cache?
- ▶ SANs are not standardized



## Conversation between DBA and SAN Administrator

---

- ▶ DBA
  - “I need storage for a new database”
- ▶ SAN Admin
  - “How big is your database?”
- ▶ DBA
  - “Well around 200 Gigabyte ....”
- ▶ SAN Admin
  - “Ok, give me a couple of minutes to create a new file system for you”
  - Or: “There is room on filesystem xyz, just create a new directory for your database”



## Conversations

---

- ▶ Do you recognize this conversations or have you experienced almost similar conversations?
- ▶ What is wrong in this exchange?
  - The SAN Admin is concerned with STORAGE capacity (“How many Gigabytes”)
  - He/She is not concerned with the number IO Per Seconds (“How many IOPS”)



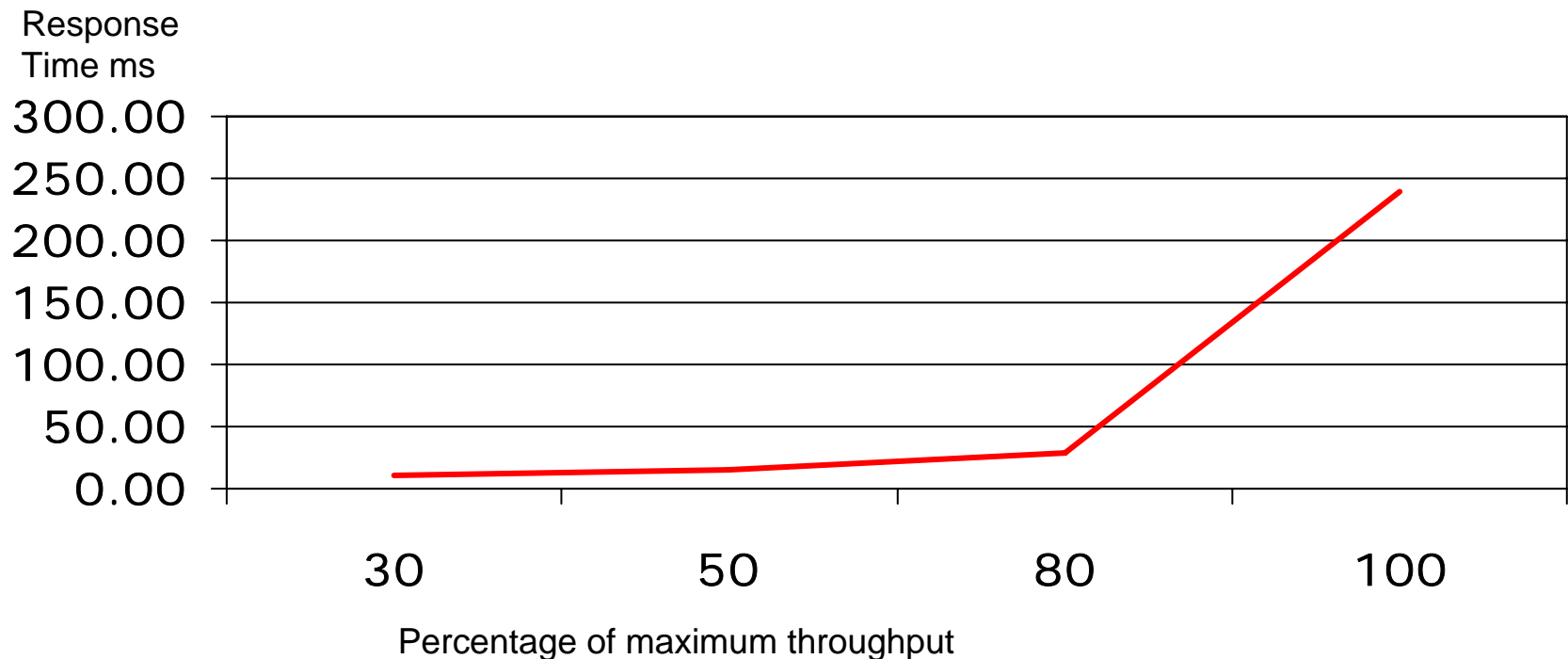
## Conversations

---

- ▶ The 200 GB could fit on a single large disk
  - Enough STORAGE
- ▶ But most likely not enough IOPS capacity
  - Depending on device utilization the IO time will go up or down. Actually it behave as the so-called “Hockey Stick Curve”



## IOPS versus Response Time



- ▶ Low response time **user-desirable** but leads to low throughput that is **system-UNdesirable** (low device utilization)

Source: Dave Patterson



## How to recognize a good SAN administrator

---

- ▶ “How many IOPS will the database need to do during peak times?”
- ▶ “What is maximum response time for the most critical transaction?”
- ▶ “How many IOPS will that transaction do?”
- ▶ “What IO response time do you need?”
  - This assumes that the DBA should know 😊
  - BTW, what is a good response time?



## What is an PIO?

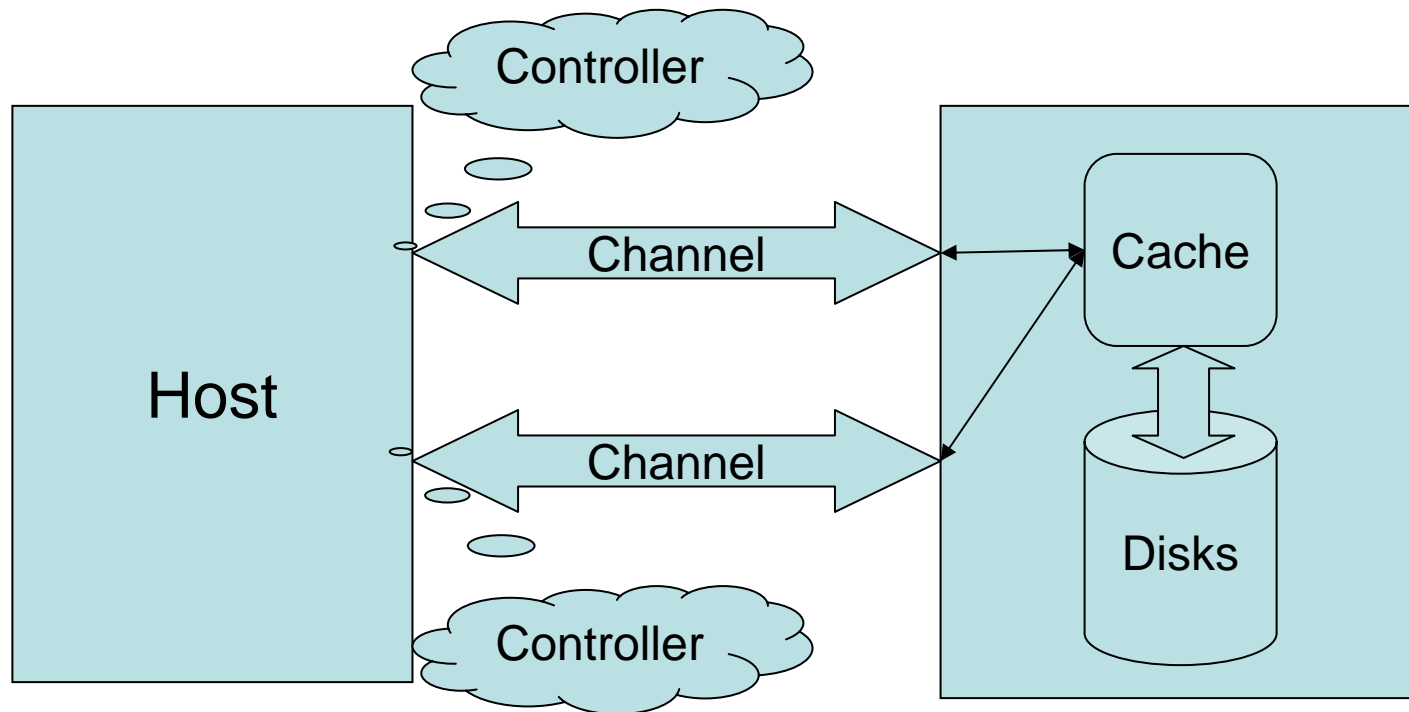
---

- ▶ PIO
  - Physical I/O
- ▶ If a buffer can't be found in the Oracle Buffer Cache, Oracle thinks it is Physical I/O
- ▶ Even if the buffer can be found in the File System Buffer Cache or the SAN Cache



## Oracle and Disk Arrays/SANs

---





## PIO and SANs

---

- ▶ Physical I/Os on SANs limited by
  - Controllers, not by Channels
  - Number of disks, not amount of SAN Cache
- ▶ In order to use the SAN cache, you need to use the controllers



## The biggest mistake!

---

“IO subsystems are sized by storage requirements and not by IOPS”

**Actually: Max(disks for storage, disks for IOPS)**



## SAN has a big impact on performance

---

- ▶ Following example is from Hennessy and Patterson
  - Ultra3 SCSI controller 0.3 msec overhead, 160 MB/sec
  - Consider Oracle I/O
    - Single block reads (4K, 8K, 16K)
    - Multiblock reads (32K, 64K, 128)
    - Single block writes (4K, 8K, 16K)
    - Multiblock writes (32K, 64K, 128K)



## How many IOPS can we do?

---

- ▶ How many I/O can we do?
  - Transfer time
    - 4K / 160MB = 0.025 msec
    - 8K / 160MB = 0.05 msec
    - 16K / 160MB = 0.1 msec
    - 32K / 160MB = 0.2 msec
    - 64K / 160MB = 0.4 msec
  - Controller Overhead + Transfer Time
    - 64K  $1/(0.3 + 0.4)\text{ms} = 1428$  IOPS
    - 32K  $1/(0.3 + 0.2)\text{ms} = 2000$  IOPS
    - 16K  $1/(0.3 + 0.1)\text{ms} = 2500$  IOPS
    - 8K  $1/(0.3 + 0.05)\text{ms} = 2857$  IOPS
    - 4K  $1/(0.3 + 0.025)\text{ms} = 3076$  IOPS



## How many IOPS can we do?

---

- ▶ The number of IOPS per controller varies depending on the I/O block size, between 1400 to 3100.
- ▶ This means that the SCSI bus is only utilized for
  - $1400 * 64K = 87.5 \text{ MB} / 160 \text{ MB} = 54.68\%$
  - $3100 * 4K = 12.1 \text{ MB} / 160 \text{ MB} = 7.56 \%$
  - (Ignoring SCSI protocol overhead)



## How many I/O can a disk do?

---

- ▶ Average Time for a disk I/O:
  - 15000 RPM
  - 5 ms average seek time
  - 40 MB/sec transfer rate
- ▶ I/O time is
  - $5 \text{ ms} + (\text{rotational delay})/15000 + (\text{blocksize}/(40 \text{ MB/sec}))$
  - $5 \text{ ms} + 0.5 / 15000 + 32\text{K}/40\text{MB} = 7.8 \text{ msec}$
  - $1 / 7.8 \text{ ms} = 128 \text{ IOPS}$
- ▶ Example from page 745, Computer Architecture: A Quantitative Approach, 3<sup>rd</sup> Edition



## Different block sizes

---

- ▶ 4K (1/7.1 ms = 140 IOPS)
  - $5\text{ms} + (0.5/15000\text{RPM}) + 4\text{K}/40\text{MB} = 5 + 2 + 0.1 = 7.1$
- ▶ 8k (1/7.2 ms = 139 IOPS)
  - $5\text{ms} + (0.5/15000\text{RPM}) + 8\text{K}/40\text{MB} = 5 + 2 + 0.2 = 7.2$
- ▶ 16K (1/7.4 ms = 135 IOPS)
  - $5\text{ms} + (0.5/15000\text{RPM}) + 16\text{K}/40\text{MB} = 5 + 2 + 0.4 = 7.4$
- ▶ 32K (1/7.8 ms = 128 IOPS)
  - $5\text{ms} + (0.5/15000\text{RPM}) + 32\text{K}/40\text{MB} = 5 + 2 + 0.8 = 7.8$
- ▶ 64K (1/8.6 ms = 116 IOPS)
  - $5\text{ms} + (0.5/15000\text{RPM}) + 64\text{K}/40\text{MB} = 5 + 2 + 1.6 = 8.6$



## Optimizing I/O Response Times

---

- ▶ M/M/1
  - Server utilization = arrival rate x Time<sub>Server</sub>
  - Time<sub>System</sub> = Time<sub>Server</sub> + Time<sub>Queue</sub>
  
- ▶ 8K blocksize (135 IOPS, 7.2 ms)
  - 135 => 240.0 ms
  - 105 => 29.5 ms
  - 75 => 15.7 ms
  - 45 => 10.6 ms



## Optimizing Response Time

---

- ▶ 64K blocksize (116 IOPS, 8.6 ms)
  - 135 => Not Possible
  - 105 => 88.6 ms
  - 75 => 24.6 ms
  - 45 => 14.6 ms



## Some Conclusions

---

- ▶ Seek Time still biggest component
  - 5 ms out of 7.1 to 8.6 (58 to 70 percent)
  - Disks should not be fully utilized, but may be only for 60 percent of the Maximum IOPS
    - This is of course application and end user dependent.
  - Buy more disks for IOPS not for storage
  - Like with all critical resources don't aim for 100 percent utilization!
    - IOPS
    - Storage



## Inside versus Outside Disk

---

- ▶ Interesting tests on Redhat AS 3.0 with firewire external disk (Maxtor, 200GB)
  - 3 partitions created with fdisk (sda1=2GB,sda2=196GB, sda3=2GB)
  - Randomly read 1000 blocks from partition in 2 separate processes
    - Test 1: process 1 => sda1, process 2 => sda1
    - Test 2: process 1 => sda3, process 2 => sda3
    - Test 3: process 1 => sda1, process 2 => sda3



## Results

---

- ▶ Test 1
  - 11.5 seconds
  - 2000 reads
  - 5.75 msec per I/O
- ▶ Test 2
  - 17.5 seconds
  - 2000 reads
  - 8.75 msec per I/O (50 percent slower)
- ▶ Test 3
  - 32.5 seconds
  - 2000 reads
  - 16.25 msec per I/O (180 percent slower)



## Some Interesting Facts

---

- ▶ 1/3 of the outer tracks contain around 50 percent of the data
- ▶ The outside of the disk is around 40-50 percent faster than the inside



## Interesting tools for the DBA

---

- ▶ Find the file with highest read time
  - Select file#, maxiortim from v\$filestat order by 2
  - Select file#, maxiortim from v\$tempstat order by 2
  - The timings are in centi seconds (1/100 second)



## Interesting tools for the DBA

---

- ▶ Reset this max stat with the command (as sys)
  - Execute `dbms_system.kcfrms;`
  - Will reset the
    - V\$filestat
    - V\$tempstat
    - V\$session\_event
      - Db file sequential read
      - Db file scattered read
      - (and other I/O related events)



## The source of the problem (highlighted)

---



The disk arm  
is the real  
problem !!



## References

---

- ▶ Computer Architecture: A Quantitative Approach, John L. Hennessy & David A. Patterson, ISBN 1558607242
- ▶ James Morle, Scalabilities, <http://www.scaleabilities.co.uk>
- ▶ Oak Table (<http://www.oaktable.net>)
- ▶ Jonathan Lewis (<http://jlcomp.demon.co.uk>)
- ▶ OraPerf.com (<http://www.oraperf.com>)
- ▶ BAARF (<http://www.baarf.com>)