NZOUG
Empower. Elevate. Excite.
2014
The University of Auckland, Auckland
New Zealand, 19 to 21 November

Architecting for Flash

Nathan King
Director Systems Engineering - ANZ
nking@vmem.com

What We're Going to Cover

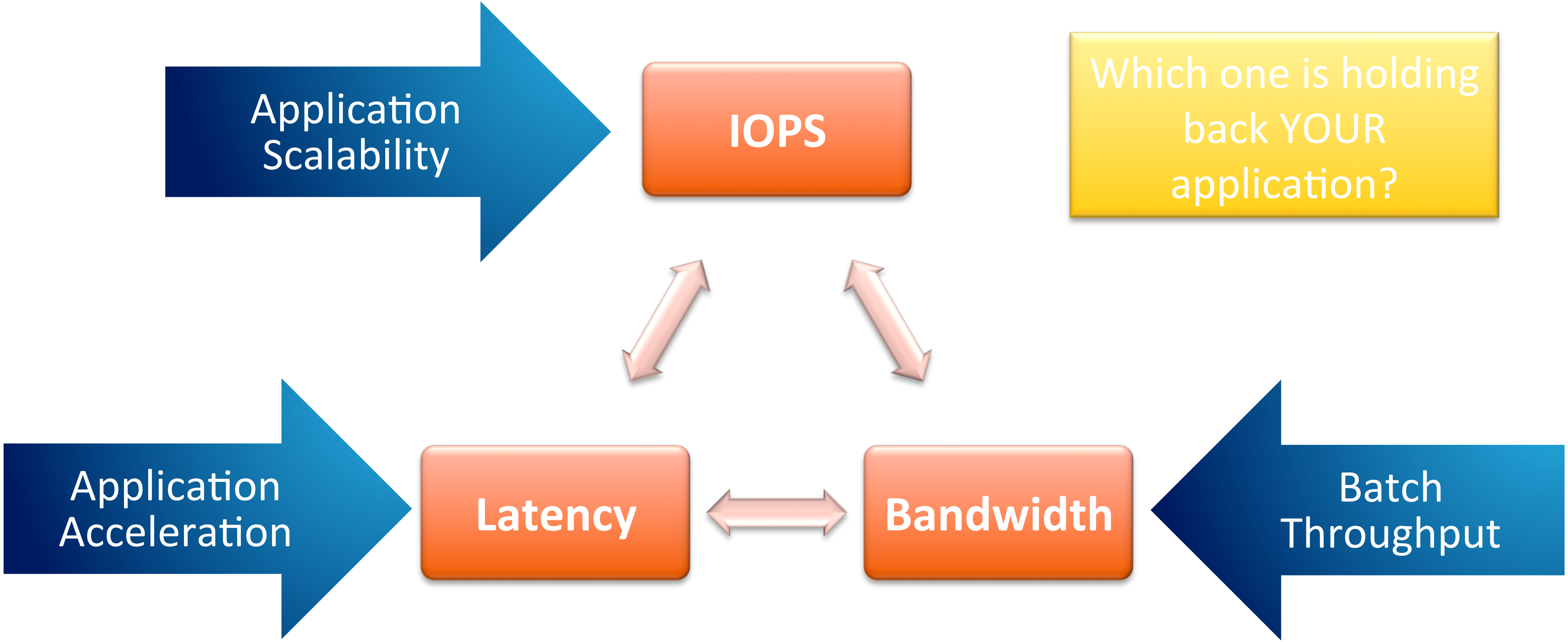
- How IO performance often dictates application performance
- How to identify latency
- The impact of latency
- How to analyze your own DB performance
- See how real customers realized 10x or better application performance just by migrating to Violin All Flash Arrays

The Technology Language Barrier

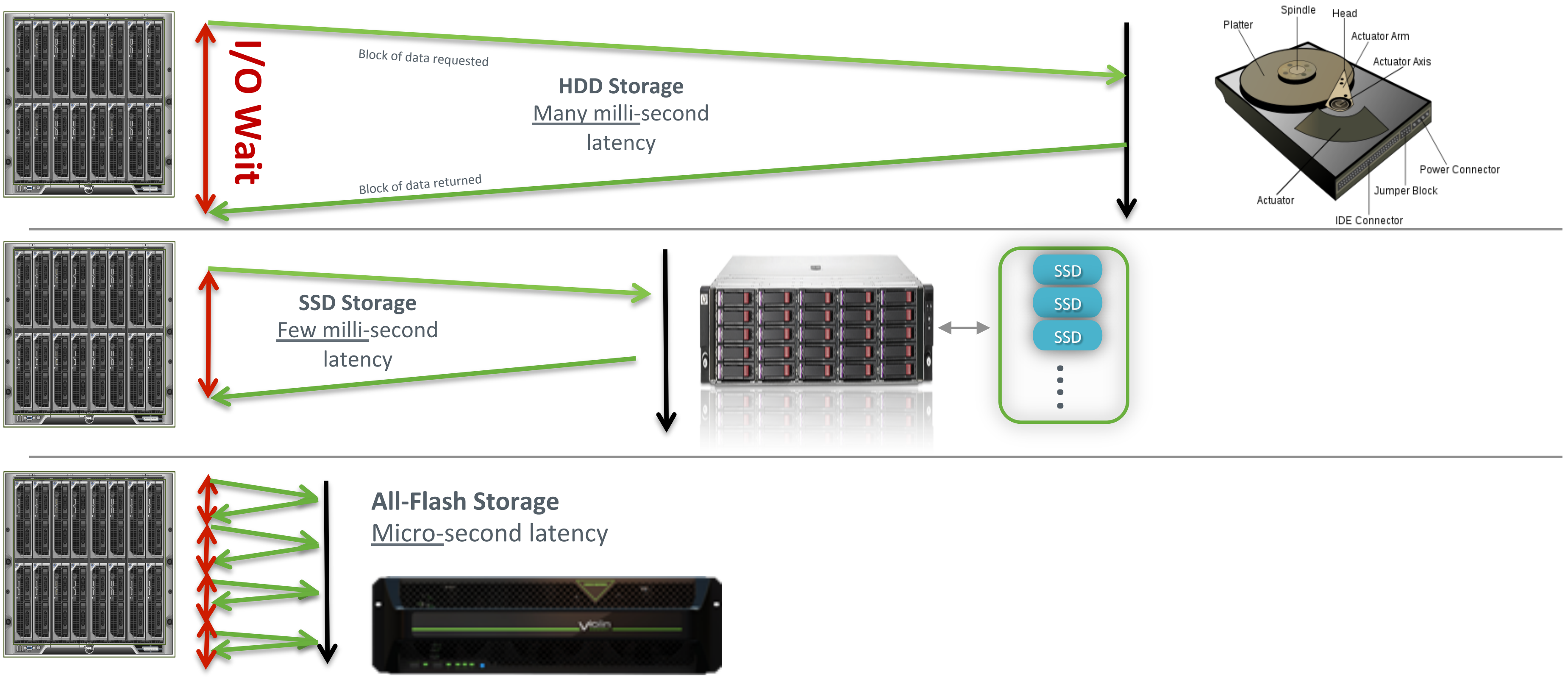
- In the I.T. industry each area of specialism has its own terminology
  - Consider a simple performance issue:

Application	User says "system is slow"
Database	Database Administrator says "wait time is high"
Server	OS Admin says system has "high IOWAIT"
Storage	Storage Admin says "latency is high"

Translating “Storage” Into “Application”



Visualizing latency - I/O Wait Infrastructure



HDD Latency

- 15,000 rotations per minute
- 250 rotations per second
- 1 rotation = 4 milliseconds
- Ave latency = 2 milliseconds
- Add time for head movement
- Ave Seek Time = 3.4ms reads
- Ave Seek Time = 3.9ms writes

15k RPM SAS Drive

HDD IOPS Capacity

15k RPM SAS Drive

- Max 200 IOPS per spindle
- To service a database with a requirement for 100k IOPS:
- = 500 disks
- Consider operational costs:
- Power, cooling, real estate, etc

HDD Bandwidth

15k RPM SAS Drive

- Good at <u>sequential I/O</u>
- Around 200 MB/sec for 2MB blocks

- Bad at <u>random I/O</u>
- Around 1 MB/sec for 4k blocks
- Due to overhead of seek time

What you end up with

How do you build an all flash array?

The Violin Advantage

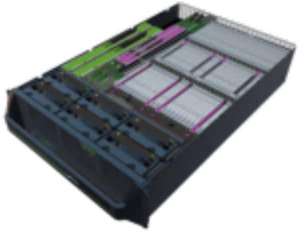
Technological innovation at every layer from Hardware to Software

- Intellectual Property (IP) aggregation resulting in a fundamentally unique solution

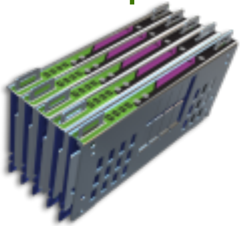
Deep software and hardware integration

- Toshiba partnership
- Violin Switched memory architecture
- VMOS - Violin Memory Operating System optimized for flash
- vRAID - Flash optimized RAID

Violin 6000/7000



vRAID Group



Violin Intelligent Memory Module (VIMM)

Toshiba Flash

But you can now have

Seeing the stats – OS Level

```
[root@colo-fc1-cn4 ~]# iostat -xtdm 10 /dev/dm-2 /dev/dm-3 /dev/dm-4 /dev/dm-5 /dev/dm-6 /dev/dm-7 /dev/dm-8 /dev/dm-9
Linux 2.6.32-279.el6.x86_64 (colo-fc1-cn4.eng.vmem.int)        09/22/2014      _x86_64_        (24 CPU)

09/22/2014 12:54:17 PM
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
dm-2              0.00     0.00    4.01    0.95     0.06     0.01    27.09     0.00    0.61   0.13   0.06
dm-3              0.00     0.00    4.05    0.94     0.06     0.01    32.44     0.00    0.59   0.22   0.11
dm-4              0.00     0.00    2.00    0.56     0.03     0.01    32.95     0.00    0.78   0.08   0.02
dm-5              0.00     0.00    2.55    0.56     0.04     0.01    32.92     0.00    0.73   0.14   0.04
dm-6              0.00     0.00    2.07    0.57     0.03     0.01    33.04     0.00    0.78   0.09   0.02
dm-7              0.00     0.00    3.49    0.94     0.06     0.01    32.46     0.00    0.62   0.21   0.09
dm-8              0.00     0.00    2.35    1.26     0.04     0.02    32.74     0.00    0.66   0.12   0.04
dm-9              0.00     0.00    2.33    0.57     0.04     0.01    32.93     0.00    0.73   0.11   0.03

09/22/2014 12:54:27 PM
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
dm-2              0.00     0.00 16612.00 4555.50   129.78    48.33    17.23    11.00    0.52   0.05  99.07
dm-3              0.00     0.00 17311.70 4746.70   135.26    50.05    17.20    11.49    0.52   0.04  99.08
dm-4              0.00     0.00 16915.00 4535.20   132.15    48.17    17.22    11.11    0.52   0.05  98.97
dm-5              0.00     0.00 16574.20 4553.20   129.49    48.58    17.26    10.91    0.52   0.05  99.00
dm-6              0.00     0.00 16812.40 4593.80   131.35    48.64    17.22    11.12    0.52   0.05  99.01
dm-7              0.00     0.00 17042.80 4619.80   133.16    48.82    17.20    11.30    0.52   0.05  99.11
dm-8              0.00     0.00 16958.80 4641.70   132.49    49.72    17.28    11.25    0.52   0.05  98.88
dm-9              0.00     0.00 16881.10 4590.60   131.88    48.84    17.24    11.16    0.52   0.05  99.21

09/22/2014 12:54:37 PM
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  svctm  %util
dm-2              0.00     0.00 16383.80 4694.60   128.00    49.52    17.25    10.78    0.51   0.05  99.18
dm-3              0.00     0.00 16957.40 4776.20   132.50    50.25    17.22    11.20    0.52   0.05  99.28
dm-4              0.00     0.00 16555.00 4615.10   129.34    49.29    17.28    10.81    0.51   0.05  99.07
dm-5              0.00     0.00 16292.90 4683.70   127.29    49.56    17.27    10.74    0.51   0.05  99.24
dm-6              0.00     0.00 16395.90 4618.00   128.09    49.40    17.30    10.72    0.51   0.05  99.15
dm-7              0.00     0.00 16586.60 4689.40   129.60    49.26    17.22    10.93    0.51   0.05  99.00
dm-8              0.00     0.00 16724.50 4719.40   130.66    49.35    17.19    11.02    0.51   0.05  99.19
dm-9              0.00     0.00 16451.00 4703.20   128.52    49.60    17.24    10.81    0.51   0.05  99.08
```

Actual IO time

IO queue time

IOPS

Bandwidth

Seeing Latency – AWR Detail #1

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
db file sequential read	6,756,848	36,656	5	83.53	User I/O
DB CPU		4,349		9.91	
log file sync	685,997	2,451	4	5.58	Commit
library cache: mutex X	14,875	247	17	0.56	Concurrency
read by other session	38,699	175	5	0.40	User I/O

Foreground = Database process waits for this to complete before moving on!

Even with decent latency (for spinning drive storage) of 5ms, IO waits account for 90% of this database's time.

Seeing Latency – AWR Detail #2

Foreground Wait Events

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% DB time
db file sequential read	6,756,848	0	36,656	5	9.73	83.53
log file sync	685,997	0	2,451	4	0.99	5.58
library cache: mutex X	14,875	0	247	17	0.02	0.56
read by other session	38,699	0	175	5	0.06	0.40
control file sequential read	350,149	0	147	0	0.50	0.33
direct path write temp	75,690	0	132	2	0.11	0.30
gc current grant 2-way	398,586	0	94	0	0.57	0.21
gc current block 2-way	235,159	0	92	0	0.34	0.21
enq: TX - index contention	8,587	0	67	8	0.01	0.15
gc cr grant 2-way	265,092	0	64	0	0.38	0.15
SQL*Net message to client	13,028,285	0	37	0	18.76	0.08

Foreground Wait Events listing provides more detail and a longer list.

Look for those comprising the bulk of DB time.

Seeing Latency – AWR Detail #3

Wait Event Histogram

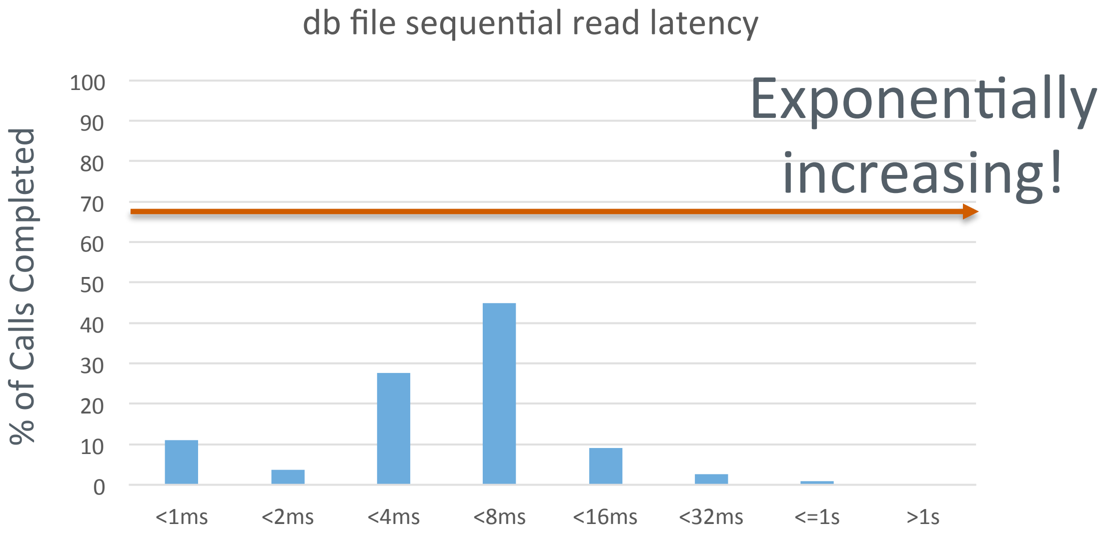
		% of Waits							
Event	**Total Waits**	**<1ms**	**<2ms**	**<4ms**	**<8ms**	**<16ms**	**<32ms**	**<=1s**	**>1s**
db file sequential read	6755.7K	11.1	3.6	27.7	45.0	9.1	2.7	.8	

Pictures are helpful:

Notice the hump at the 8ms bucket?

What is that?

How does that affect my application's performance?



Seeing Latency – AWR Detail #4

Tablespace IO Stats

Tablespace	Reads	Av Reads/s	Av Rd(ms)	Av Blks/Rd	Writes	Av Writes/s	Buffer Waits	Av Buf Wt(ms)
INDEX_1	1,863,306	517	6.09	1.00	1,779,678	494	8	2.50
DATA_1	1,327,584	368	5.86	1.00	1,296,604	360	114	4.30
INDEX_2	598,728	166	4.16	1.00	390,408	108	24,370	2.95
DATA_2	217,744	60	5.87	1.00	322,080	89	1,743	4.04
DATA_3	443,353	123	6.04	1.00	80,714	22	35	5.14
DATA_4	195,901	54	5.79	1.00	312,065	87	435	3.66
UNDOTBS1	373,967	104	1.68	1.00	25,325	7	501	0.52

Reports latency per-tablespace.

Be careful with Direct Path IO though

http://flashdba.com/2014/09/03/oracle-parallelism-and-direct-path-reads-on-flash/

Latency's Impact – Top SQLs #1

SQL ordered by Elapsed Time

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
8,148.83	448,972	0.02	18.57	2.33	98.13	grxdzpwbyxakm	…	select …
7,262.71	1,167,258	0.01	16.55	3.14	96.78	48shscy5ncbh5	…	delete ...
5,666.32	728,259	0.01	12.91	3.26	96.09	1kv8d31rc3bxb	…	delete ...
2,580.17	448,957	0.01	5.88	2.66	97.85	2yk56qjwsxxcb	…	select ...
2,545.43	574,924	0.00	5.80	6.32	90.75	5cvk2t6ap0wbj	…	delete ...
2,423.67	574,928	0.00	5.52	8.25	86.63	bvsdfqb1fwp93	…	delete ...
1,856.30	33,451	0.06	4.23	4.72	92.22	5yb6pmjjf0axc	…	select ...
1,566.86	2,206,990	0.00	3.57	9.11	76.01	bs1401t3pnpqy	…	update ...
1,543.37	533,482	0.00	3.52	2.94	97.11	0ynxtz71bntdr	…	select ...
1,348.64	127,032	0.01	3.07	2.19	97.78	3g6452kc8nwcf	…	select ...
1,071.36	463,794	0.00	2.44	7.81	89.83	dgagw2nht0rr3	…	DELETE ...

Even with very fast executions (10-20ms elapsed averages), IO wait still makes up 90%+ elapsed time for most SQLs.

Latency's Impact – Top SQLs #2

SQL ordered by Reads

Physical Reads	Executions	Reads per Exec	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
1,531,551	448,972	3.41	22.55	8,148.83	2.33	98.13	grxdzpwbyxakm	…	select ...
1,218,922	1,167,258	1.04	17.95	7,262.71	3.14	96.78	48shscy5ncbh5	…	delete ...
885,670	728,259	1.22	13.04	5,666.32	3.26	96.09	1kv8d31rc3bxb	…	delete ...
455,183	448,957	1.01	6.70	2,580.17	2.66	97.85	2yk56qjwsxxcb	…	select ...
442,767	574,924	0.77	6.52	2,545.43	6.32	90.75	5cvk2t6ap0wbj	…	delete ...
388,201	574,928	0.68	5.72	2,423.67	8.25	86.63	bvsdfqb1fwp93	…	delete ...
341,841	533,482	0.64	5.03	1,543.37	2.94	97.11	0ynxtz71bntdr	…	select ...
296,135	33,451	8.85	4.36	1,856.30	4.72	92.22	5yb6pmjjf0axc	…	select ...
283,647	2,206,990	0.13	4.18	1,566.86	9.11	76.01	bs1401t3pnpqy	…	update ...
226,720	127,032	1.78	3.34	1,348.64	2.19	97.78	3g6452kc8nwcf	…	select ...
174,085	463,794	0.38	2.56	1,071.36	7.81	89.83	dgagw2nht0rr3	…	DELETE ...

Very few reads per execution means the buffer cache is doing its job, but each physical IO's latency is critical!

Latency's Impact – Top SQLs #3

Notice the correlation?

SQL ordered by Elapsed Ti

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%CP
8,148.83	448,972	0.02	2.3
7,262.71	1,167,258	0.01	3.1
5,666.32	728,259	0.01	3.2
2,580.17	448,957	0.01	2.6
2,545.43	574,924	0.00	6.3
2,423.67	574,928	0.00	8.2
1,856.30	33,451	0.06	4.7
1,566.86	2,206,990	0.00	9.1
1,543.37	533,482	0.00	2.9
1,348.64	127,032	0.01	2.1
1,071.36	463,794	0.00	7.8

SQL Id	SQL Id
grxdzpwbyxakm	grxdzpwbyxakm
48shscy5ncbh5	48shscy5ncbh5
1kv8d31rc3bxb	1kv8d31rc3bxb
2yk56qjwsxxcb	2yk56qjwsxxcb
5cvk2t6ap0wbj	5cvk2t6ap0wbj
bvsdfqb1fwp93	bvsdfqb1fwp93
5yb6pmjjf0axc	5yb6pmjjf0axc
bs1401t3pnpqy	bs1401t3pnpqy
0ynxtz71bntdr	0ynxtz71bntdr
3g6452kc8nwcf	3g6452kc8nwcf
dgagw2nht0rr3	dgagw2nht0rr3

eads

SQL Module	SQL Text
…	select ...
…	delete ...
…	delete ...
…	select ...
…	delete ...
…	delete ...
…	select ...
…	select ...
…	update ...
…	select ...
…	DELETE ...

Latency's Impact – SQL Elapsed Time

Oracle provides means to determine per-SQL IO latency!

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
8,148.83	448,972	0.02	18.57	2.33	98.13	grxdzpwbyxakm	…	select …

Physical Reads	Executions	Reads per Exec	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
1,531,551	448,972	3.41	22.55	8,148.83	2.33	98.13	grxdzpwbyxakm	…	select ...

8,148.83 seconds / 448,972 executions = **18.1ms** (time for each execution)

8,148.83 seconds x 98.13% IO = 7,996 seconds total IO time

7,996 seconds / 1,531,551 reads = **5.2ms per read**

3.41 reads per execution x 5.2ms per read = **17.7ms** (IO time) for each execution

IO time is 17.7/18.1 = 97.8% of each execution!

Calculating the Impact

(elapsed time) / (number of executions) = time for each execution
(reads per execution) x (latency) = IO time
(time for each execution) - (IO time) = CPU time

From the previous slide:

3.41 reads per execution x 5.2ms per read = 17.7ms IO time for each execution
18.1ms total exec time – 17.7ms IO time = 0.4ms CPU time — IO time is 97.8% of each execution

What happens when average latency is 0.25ms instead of 5.2ms?

3.41 reads per execution x 0.25ms per read = 0.85ms IO time for each execution
0.85ms + 0.4ms = 1.25ms new execution time — IO time is now 68% of each execution

93% reduction in elapsed time (1.25ms vs 18.1ms)

95% reduction in IO time (0.85ms vs 17.7ms)

Latency's Impact – Real Application Performance

Customer Provisioning System PoC Results – Major TV Provider

SQL ID	QA Average Elapsed (ms)	QA/Violin Average Elapsed (ms)	Violin Improvement (%)
3g6452kc8nwcf	121	1.4	**99**
bvsdfqb1fwp93	66	1.1	**98**
gh53bxwudpkxj	41	0.9	**98**
5cvk2t6ap0wbj	47	0.9	**98**
a88cym5upqyx2	43	2.3	**95**
2984pswc6t9am	20	0.4	**98**
dgagw2nht0rr3	30	0.4	**99**
23r31j5d28pav	33	1.0	**97**
grxdzpwbyxakm	90	1.6	**98**
7wyu0nwrz2rfv	21	0.3	**99**

Violin made QA cycles nearly 100x faster!

Latency's Impact – Real Application Performance

SAP Application PoC Results - Bay-area HealthCare Provider

Performance Analysis – All Databases

Individual SQL Performance Comparison

Database	SQL ID	Average Production Time (ms)	Average Violin Time (ms)	Violin % Improvement
SAP BI	3yc4u3pauxgtb	22	1.2	95
	d4wpvpm8t5wj1	10	0.4	96
	1ymdcvbdvj013	3.1	0.1	97
SAP CRM	9akdwpxpqwvty	9.7	0.2	98
	2ycd5bw15vjx3	8.7	0.4	96
	3chjta6bw8pbj	3.9	0.3	92
SAP PROD	41asphypkqb1u	34	0.3	99
	1mn0h8pkz6vyb	3.8	0.1	97
	310cgb2jj0z7r	15	0.2	99

Violin made this 12x-100x faster as well!

Latency's Impact – Real Application Performance

Siemens Application – Auto Manufacturer

Performance Analysis – Top 5 Elapsed SQLs

Nightly BOM Build Operation – Top SQL Performance Comparison

SQL ID	Current Exec Time (s)	Violin Exec Time (s)	% Improvement
72ava0fty5avx	14264	3582	75
chqpmv9c05ghq	10961	6881	37
6wsjkct9m05y2	7765	2443	69
bdr138jafbquq	7645	2147	72
b9k1x0z5xrd8x	6564	1311	80
Total	47199	16364	65
Average			67

Violin took over 13 hours of batch processing time down to 4.5 hours!

In summary…

- Get to know your existing storage latencies
- Love your AWRs
- Don’t believe the marketing – prove it in your environment

- Use our **free** AWR analysis service
- Violin Memory Oracle Performance Assessment Service (O-PAS)
    - www.awr.vmem.com
    - Did I mention its free??

- Happy to discuss further, get my card etc…

Violin MEMORY

Business in a Flash.

Thank You

NZOUG

Empower. Elevate. Excite.

2014

The University of Auckland, Auckland
New Zealand, 19 to 21 November

2014