

Responsive Systems – Performance Software that Works!!

DB2 Buffer Pool Tuning Getting the Benefits & System Performance Metrics The Big Picture

Joel Goldstein
Responsive Systems



© Responsive Systems 2014 www.responsivesystems.com

DB2 is a complex system, with a major impact upon your processing environment. There are substantial performance and instrumentation changes in versions 9, 10, 11, that must be used to measure, evaluate, and quantify the performance of the DB2 system and applications. This presentation addresses the origins and specific data available for analyzing, tuning, and tracking the performance of the system and applications, and the inter-dependencies and relationships between the various data elements. Our performance world has changed a lot since this topic was originally presented a dozen years ago. The presentation assumes the attendee has a good understanding of DB2, and does not define basic DB2 terminology.

Responsive Systems – Performance Software that Works!!

The foundations of performance

- **Processor**
 - CPU speed
 - Number of engines
 - Depending upon your workload, more MIPS with fewer engines may degrade performance

Basics haven't changed
- **Memory** – more, more, more... Terabyte is coming...
- **I/O**
 - Elapsed time – faster devices, more cache, what's good?
 - Even 1 Ms can be painful with a **high rate/second**
 - CPU cost of I/O

Performance is dependant upon application design & coding

The basics of performance have not changed within the last 50 years.

Responsive Systems – Performance Software that Works!!

Analysis Data

- System Data

- z/OS data
 - RMF data, from SMF
 - CPU, memory, paging, components of dasd response, etc
- DB2 performance data
 - System statistics – *one minute intervals* - summarize
 - Application accounting – large volume
 - Performance traces
 - SMF, GTF, IFI, various vendor products
 - Displays – Lstats, etc
 - 199 Records - **better than nothing, but limited value**
 - » Only objects > 1 IO/Sec

Data from multiple sources must be used for tuning. If you don't know what data you need, how to get, and how to analyze it, your tuning effectiveness will....

be less than management wants/needs. Producing Stats records every 60 seconds is only 1440 records, not a problem. This lets you find problems with a low level of granularity, but start from a higher level of initial analysis like 15 or 30 minutes.

While using anything is vastly better than not tuning, it's important to understand the limitations of data. While Lstat displays, and 199 object usage records will provide some useful information, tuning from there is simply a guess. There is no validation without changing your system. Can you afford guesswork on your production system?

Making a pool larger and monitor performance won't hurt if you don't impact system paging. Move an object to the wrong pool, and you're in big trouble.

Responsive Systems – Performance Software that Works!!

Data, Data, Everywhere...

- Never as we want it....
 - Too much - can you find the tree in the forest?
 - Too little *Slice/dice, analyze*
 - Wrong intervals or duration
 - Too much of short duration, we can summarize
 - Too little, or too long a duration, and it's useless
- Hindsight is always perfect, **if** you knew what you wanted/needed
 - People frequently send data that isn't useful...
 - DB2L - do you subscribe?
 - LinkedIn.com
 - Facebook

We need the ability, and tools to slice/dice data many different ways.
Putting data into Excel can open a whole new world of analysis.

Responsive Systems – Performance Software that Works!!

Tuning Goals

- Reduce processing costs

- Save **\$\$ millions per year with some tuning**
 - Reduce transaction/business function/job CPU processing cost
 - Improve user productivity by reducing elapsed times
 - Avoid or delay processor upgrades
 - Software upgrade costs are often much more than hardware
- **Need History data**
 - How does today's pain compare to last week/month?
 - Points of reference
 - Trends...
 - How similar is the workload?

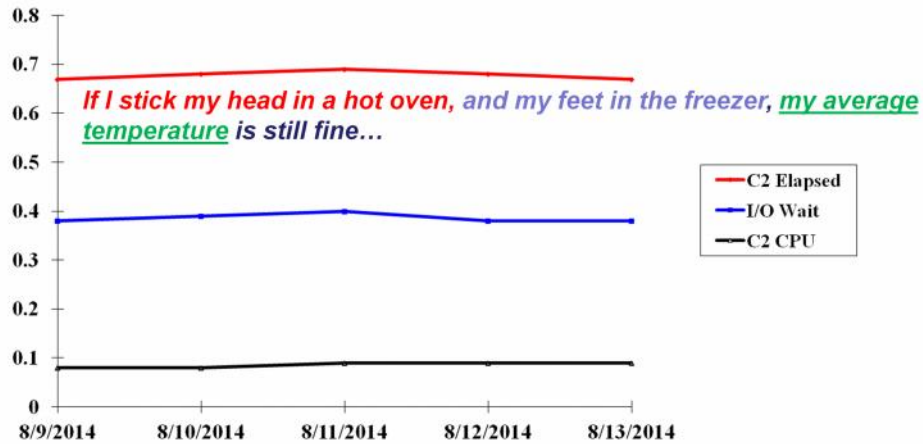


Large companies, worldwide, are wasting millions per year by ignoring tuning opportunities.

History data lets you see if this is a new problem, or an old issue that nobody complained about before. Has it always been this bad, or has it gotten worse over time?

Responsive Systems – Performance Software that Works!!

Long measurement interval
Online transaction

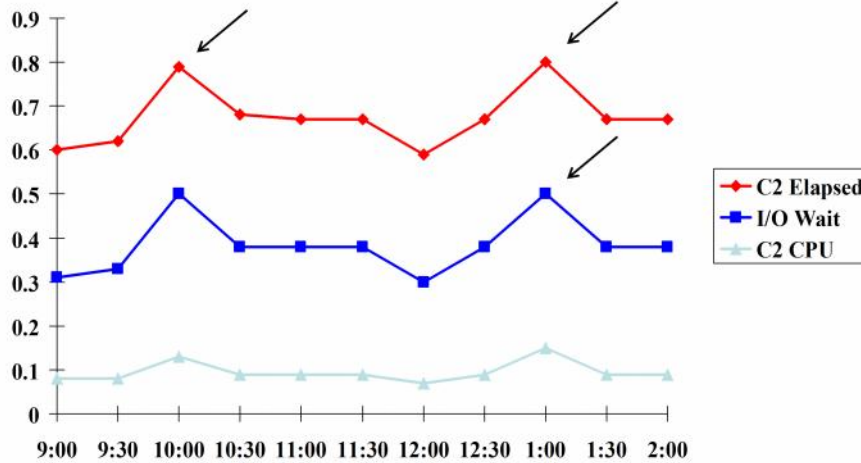


Long measurement intervals hide performance problems by showing averages over a long period. An interval spanning a hour or two during peak load periods would be valid in most cases to get average response times. However, when periods of low system activity are factored into the data, it becomes meaningless – does not represent the times you should be most concerned with – the peak periods.

Responsive Systems – Performance Software that Works!!

Shorter measurement interval

Online transaction



Shorter periods show the peaks and valleys. From here you should go to 5 minute periods between 9:45 – 10:15, and 12:45 – 13:15.

If necessary, home into a couple of 1 minute periods to isolate the worst (highest) points.

Then look into detailed application and system performance data to determine the cause of the response time spikes.

What can be done to reduce the I/O wait time issues?

Responsive Systems – Performance Software that Works!!

Data relationships, and %, are vital

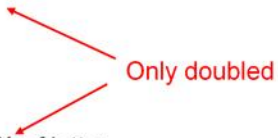
- Sometimes the raw numbers get our attention
 - Transaction elapsed times of 5, 10, 20 seconds
- Sometimes percentages are more meaningful
 - Transaction elapsed .1 secs, .08 I/O wait
 - *I/O wait is 80% of the elapsed time*
- What other factors or events might have impacted the numbers you are looking at?
 - Elapsed times are usually < 1 sec, but spike to 10 secs for some periods
 - Who does what, to whom...?? *z/OS, DB2, or Application?*

Eyeball method – what looks big or out of proportion to other data?

Responsive Systems – Performance Software that Works!!

Data relationships, and %, are vital

- Let's put it into terms of postage stamps

- In the 40's
 - Letter was .03
 - Postcard was .01
 - Letter is 3x postcard, or postcard is **33%** of letter cost
 - A few yrs ago, compared to the 40's
 - Letter is .42 14x, 1300% increase
 - Postcard is .27 27x, 2600% increase
 - Letter is 1.5x postcard, or postcard is **64%** of letter
- 

- **Opportunities....**

- Come in different sizes, different scopes, and different paybacks – it all depends upon the data you use
- *Lies, darn lies, and statistics.....*

We can play many games with data, depending upon what we want to achieve, or prove.

Lies, darn lies, and statistics.....

Statistics can prove, or disprove anything, depending upon the data sample.

Responsive Systems – Performance Software that Works!!

The biggest CPU payback

- In your applications
 - Indexing
 - SQL coding
 - Design

CPU cost is proportional to the number of pages you process

- Where's the payback?
 - You can usually find it easily
 - If you look in the right places
- Document your savings
 - blow your own horn, let people know, but don't be obnoxious about it

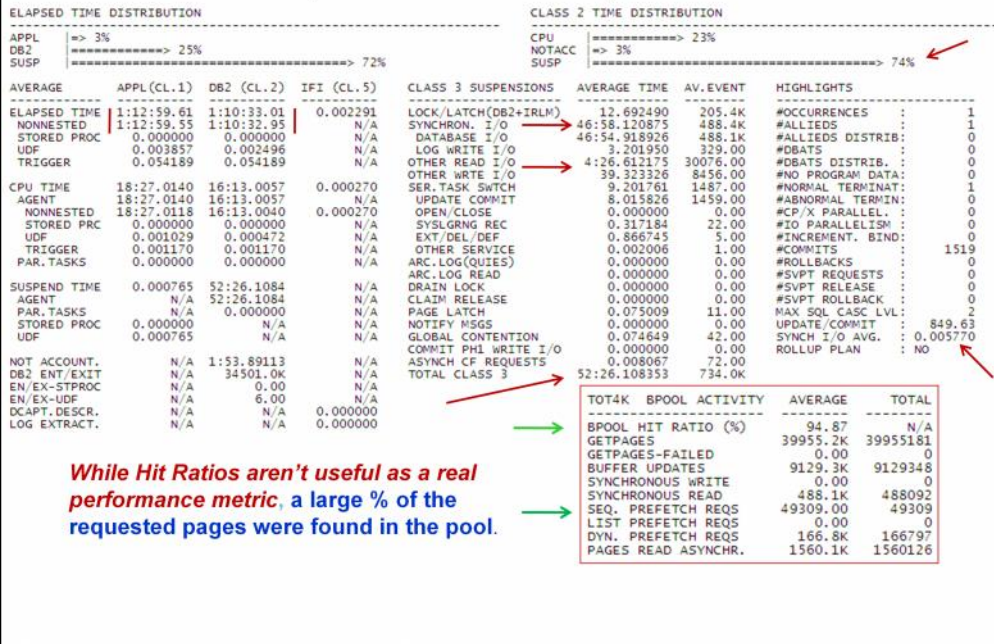


Finding and fixing one object with heavy scan activity, continually, will pay for your salary many times over.

We have several clients over the years that have cancelled their processor upgrades after some tuning...

Responsive Systems – Performance Software that Works!!

Where is the performance problem?



Application accounting report for a batch job. Job is running too long, DBA said they had a buffer pool tuning problem, and the systems people would not tune the pools.

While pool tuning might reduce overall I/O and thus the elapsed time, the underlying issues are:

- Death by synch I/O – possibly sort the input into the key order of the data
- The cache on the DASD controller is too small
- The synch I/O elapsed time is poor, that relates to b.

Speaking with the sysprogs, and analyzing system performance data, pools cannot be increased without causing the system to page.

They know they have a DASD performance issue, the cache is too small, and DASD upgrades are a few months in the future.

Responsive Systems – Performance Software that Works!!

Made tuning changes

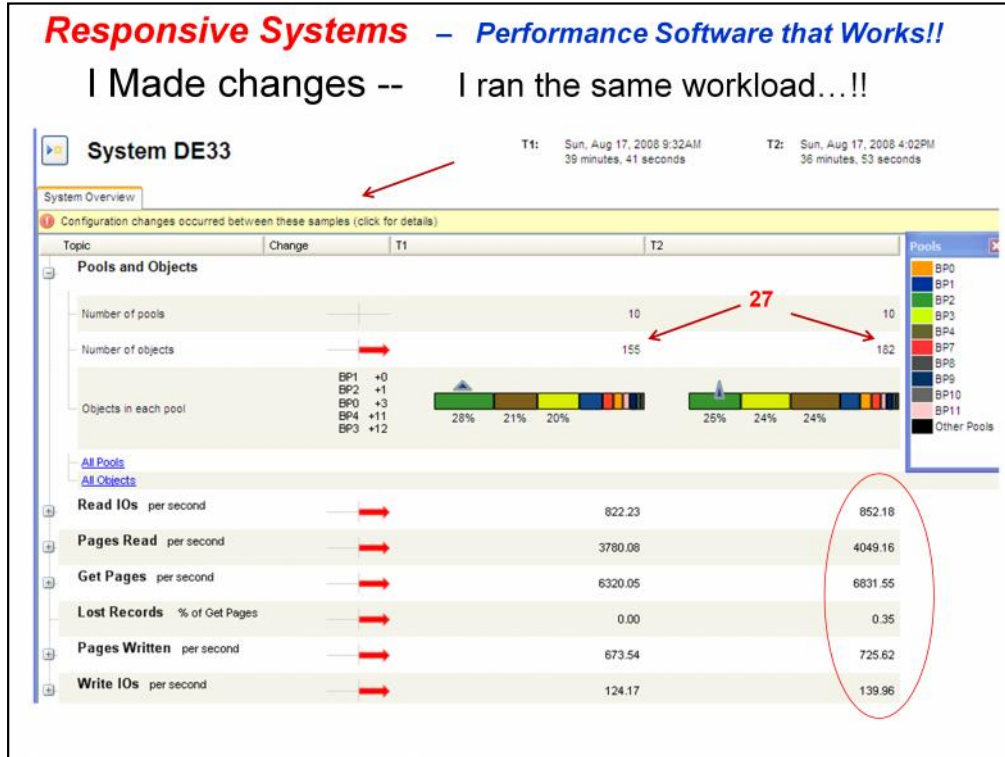
is performance better now?

- Workload comparisons must be reasonably similar
 - Quite common to have significant differences of object usage, objects accessed, type of access
 - Same timeframe?
 - Length of comparison interval
 - Is your workload infinitely variable?
 - Very difficult to measure
 - Object activity inter-relationships
 - Dynamic queries?
 - Batch jobs all day?

Need multiple measurement points, both before and after



One of the most difficult problems when comparing performance, is having workloads that are reasonably similar. Unless you have a stand-alone test system, with a specific driven workload, you will never achieve an exact comparison. So the key for comparisons is – “reasonably similar”.



Client made pool tuning changes to measure the improvement of a batch workload. Adamant that they ran the exact same workload, and complained that the I/O rate increased.

However 27 additional (new) objects were accessed during the second performance measurement period. The Getpage rate was 10% higher, number of pages written, and write I/O was higher.

The workload in the overall system was quite different.

Responsive Systems – Performance Software that Works!!

Pool performance analysis – *before* 1a

The screenshot shows the Buffer Pool Tool for DB2 - BP2 interface. It includes a menu bar with options like Report Info, Graphic Summary, Pool Info, Object Info, Expert Tuning, Scan Cost, I/O Cost, Sim Graph Analysis, Sim Cluster Analysis, and Memory. On the left, there are input fields for Date (2008-09-03), Time (17:29:03), Elapsed Time (00:44:16), System (T104), Sub System (DE34), DB2 Version (8.1), and DS Group (DE30). The main area contains a table of performance metrics for various pools (BP0 to BP11 and BP8K0). A red arrow points to the 'Total 4K Buffers 250,000' field at the bottom left. Summary statistics are shown at the bottom right.

Pool	RI/O/Sec	Get Pages	Updates	Hit Ratio	I/O	WIO/Sec	Pages/Write	Write I/Os	Pages
BP0	0.01	5873	1100	98.4	68	0.02	11.11	44	
BP1	239.56	5561120	2858986	14.9	749341	42.57	11.38	113072	
BP2	859.45	41060193	6391853	87.7	2611449	123.78	6.21	328759	
BP3	0.07	44216	11797	99.3	383	0.08	5.38	202	
BP4	0.00	80132	2	100	12	0.00	1.75	4	
BP6	49.19	4240701	0	88	130643	0.00	0.00	0	
BP7	0.00	131489	87659	100	0	0.00	0.00	0	
BP8	389.91	8157446	4421953	10.5	1090852	20.80	30.93	55257	
BP9	965.14	60904853	15400381	93.9	2668043	38.64	26.90	102622	
BP10	313.05	5203326	2239280	1	877685	17.40	22.44	46221	
BP11	1,226.28	40559291	10964901	88.4	3375041	44.44	28.57	118045	
BP8K0	0.00	7	0	100	0	0.00	0.00	0	

Summary Statistics:

- Total 4K Buffers: 250,000
- Total Read/Write I/O: 11,501,517
- Overall Sys Hit Ratio: 81.23
- Total Updates: 42,377,892
- Total Get Pages: 165,948,647
- Total I/Os per second: 4,330.39
- Pages per write: 15.98

Before tuning

The following performance data is after moving to a larger, faster machine, with more memory – and other workloads.

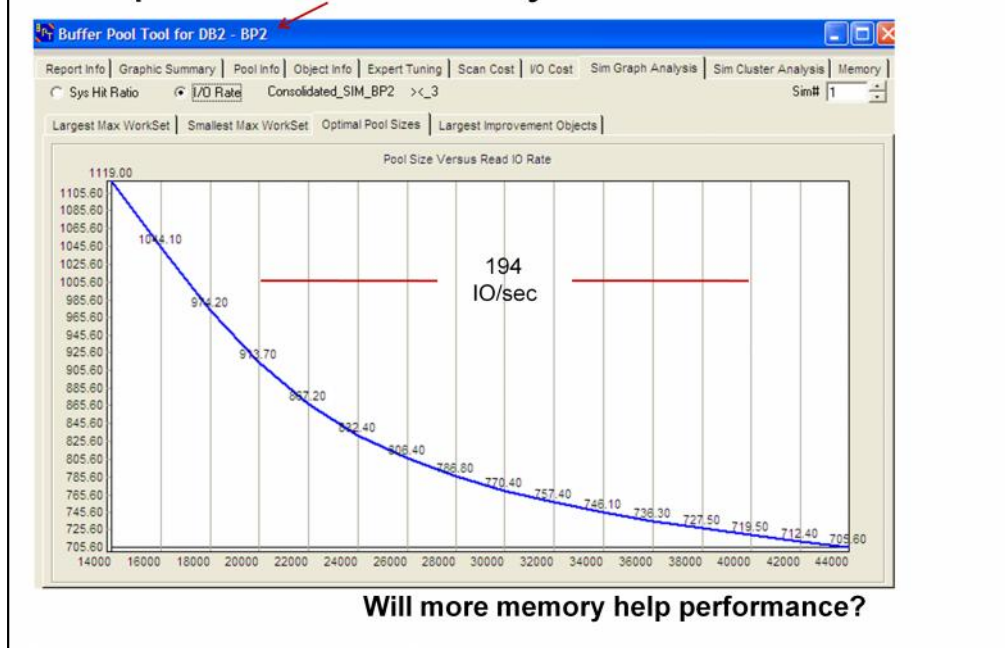
High overall I/O rate, spread across several pools

Before tuning, making some pools larger

The above and following slides titled “before” look at the current performance, and the “predicted” improvements of pool size increases

Responsive Systems – Performance Software that Works!!

Pool performance analysis – before 1b

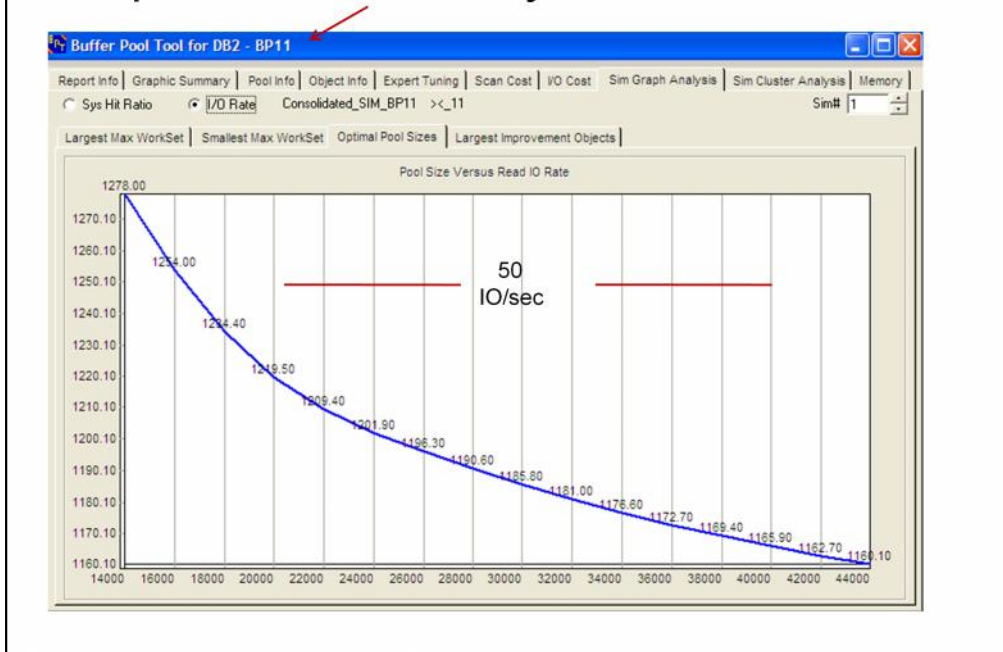


Predicted saving by doubling BP2 from 20,000 to 40,000 buffers, a bit less than 200 I/O second

Before tuning, making some pools larger....

Responsive Systems – Performance Software that Works!!

Pool performance analysis – before 1c



Predicted saving by doubling BP11 from 20,000 to 40,000 buffers, about 50 I/O sec, 1165 at 40,000.

Before tuning, making some pools larger....

Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1a

The screenshot shows the Buffer Pool Tool for DB2 - BP2 interface. It includes a menu bar with options like Report Info, Graphic Summary, Pool Info, Object Info, Expert Tuning, Scan Cost, I/O Cost, Sim Graph Analysis, Sim Cluster Analysis, and Memory. On the left, there are input fields for Date (2008-09-04), Time (20:56:21), and Elapsed Time (00:49:41). Below these are System Info fields for System (T104), Sub System (DE34), DB2 Version (8.1), and DS Group (DE30). The main area contains a table of performance metrics for various buffer pools (BP0 to BP9). A large text box in the center states "Added 60,000 buffers, saved 700+ IO/Sec". At the bottom, summary statistics are shown: Total 4K Buffers 310,000, Total Read/Write IO 11,413,991, Overall Sys Hit Ratio 80.66, Total Updates 40,246,859, Total Get Pages 168,129,351, Total I/Os per second 3,828.91, and Pages per write 14.45. Red arrows point to the "Total 4K Buffers" and "Total I/Os per second" values. Below the screenshot, the text "Saved 737 IO/Sec..." is displayed in red.

Pool (A)	RIQ/Sec	Get Pages	Updates	Hit Ratio	I/O	WIO/Sec	Pages/Write	Write I/Os	Page
BP0	0.07	8566	1550	91.1	268	0.02	11.04	52	
BP1	266.15	5840459	2996627	8.3	950535	52.72	8.74	157149	
BP10	312.74	5176445	2095127	-2.7	1021357	29.88	11.57	89087	
BP11	1,027.07	39796458	8237241	88.2	3240602	60.02	18.23	178916	
BP2	737.96	43249062	7180492	87.5	2448795	83.51	8.09	248929	
BP3	6.14	115985	34732	79.9	26709	2.82	2.64	8410	
BP4	0.01	737306	160	100	49	0.00	1.50	12	
BP6	49.57	4709311	0	84.3	147768	0.00	0.00	0	
BP7	0.00	374153	249432	100	0	0.00	0.00	0	
BP8	335.82	8222158	4387503	12.5	1055921	18.40	31.16	54856	
BP9	812.57	59899448	15063995	93.7	2521997	33.46	26.90	99740	

Added 60,000 buffers, saved 700+ IO/Sec

Total 4K Buffers **310,000** Total Read/Write IO **11,413,991** Total Get Pages **168,129,351**
 Overall Sys Hit Ratio **80.66** Total I/Os per second **3,828.91**
 Total Updates **40,246,859** Pages per write **14.45**

Saved 737 IO/Sec...

After the pools were enlarged.... After base pool tuning.

High overall I/O rate, spread across several pools

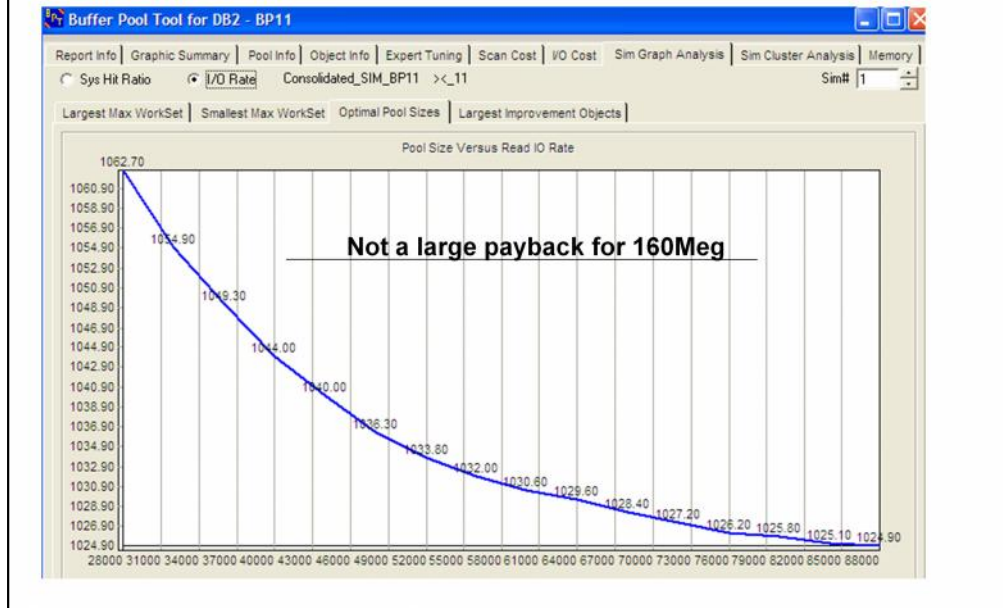
The I/O rate dropped for both BP2 and BP9 that were increased.

We predicted 720 I/O second at 40,000 buffers, we got 737.

There are still variations in the workload.

Responsive Systems – Performance Software that Works!!

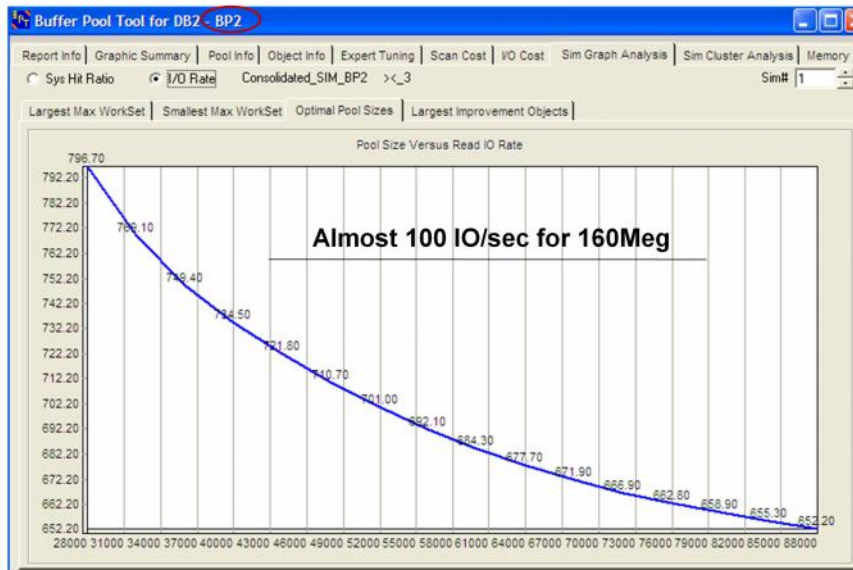
Pool performance analysis – after 1b



Additional doubling BP11 from 40,000 to 80,000 buffers will save about 20 I/O second. Not a large payback, but later slide will imply that we might do better because more pages are found in the DASD cache..

Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1c

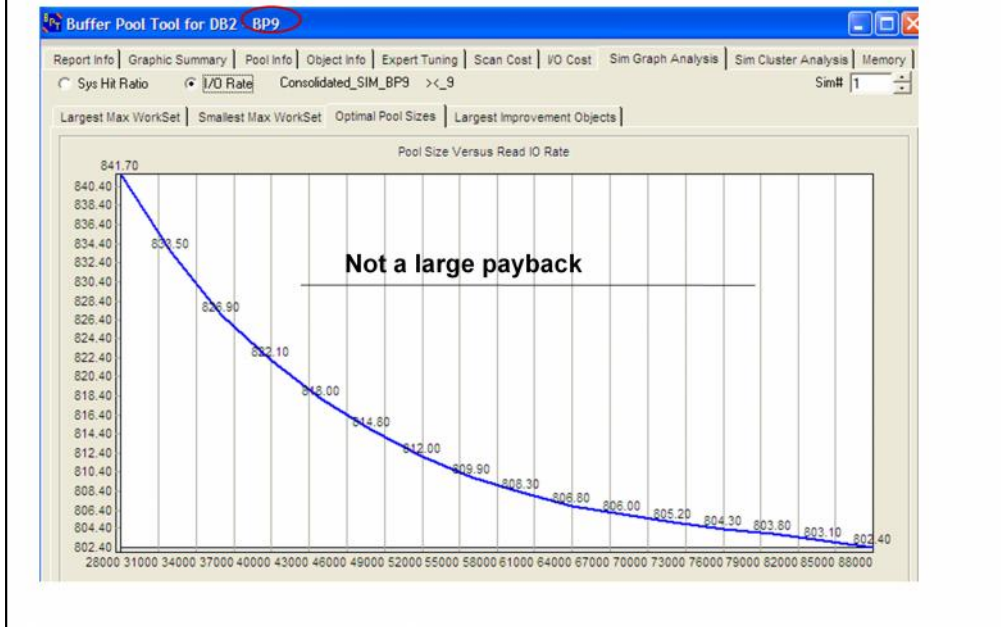


Is there a knee to this curve where the I/O drops dramatically?

Doubling BP2 from 40,000 to 80,000 saves almost 100 I/O second. There might be a bit more to be gained, but the curve is flattening rapidly.

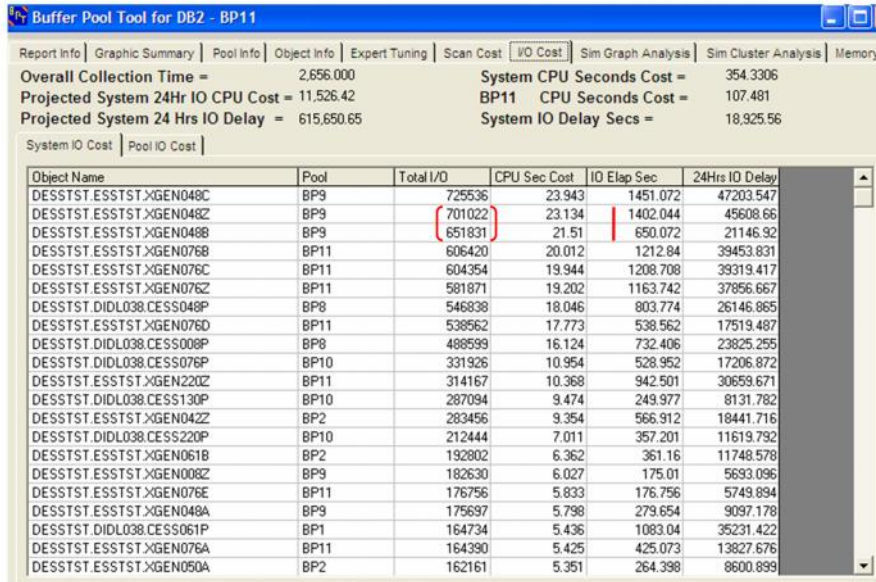
Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1d



Doubling BP9 from 40,000 to 80,000 saves about 20 I/O second. Not a large payback, and the curve flattens as we give it more memory.

Responsive Systems – Performance Software that Works!!
Pool performance analysis – after 1e



Object Name	Pool	Total I/O	CPU Sec Cost	IO Elap Sec	24Hrs IO Delay
DE SSTST.ESSTST.XGEN048C	BP9	725536	23.943	1451.072	47203.547
DE SSTST.ESSTST.XGEN048Z	BP9	701022	23.134	1402.044	45608.66
DE SSTST.ESSTST.XGEN048B	BP9	651831	21.51	650.072	21146.92
DE SSTST.ESSTST.XGEN076B	BP11	606420	20.012	1212.84	39453.831
DE SSTST.ESSTST.XGEN076C	BP11	604354	19.944	1208.708	39319.417
DE SSTST.ESSTST.XGEN076Z	BP11	581871	19.202	1163.742	37856.667
DE SSTST.DIDL038.CESS048P	BP8	546838	18.046	803.774	26146.885
DE SSTST.ESSTST.XGEN076D	BP11	538562	17.773	538.562	17519.487
DE SSTST.DIDL038.CESS008P	BP8	488599	16.124	732.406	23825.255
DE SSTST.DIDL038.CESS076P	BP10	331926	10.954	528.952	17206.872
DE SSTST.ESSTST.XGEN220Z	BP11	314167	10.368	942.501	30659.671
DE SSTST.DIDL038.CESS130P	BP10	287094	9.474	249.977	8131.782
DE SSTST.ESSTST.XGEN042Z	BP2	283456	9.354	566.912	18441.716
DE SSTST.DIDL038.CESS220P	BP10	212444	7.011	357.201	11619.792
DE SSTST.ESSTST.XGEN061B	BP2	192802	6.362	361.16	11748.578
DE SSTST.ESSTST.XGEN008Z	BP9	182630	6.027	175.01	5693.096
DE SSTST.ESSTST.XGEN076E	BP11	176756	5.833	176.756	5749.894
DE SSTST.ESSTST.XGEN049A	BP9	175697	5.798	279.654	9097.178
DE SSTST.DIDL038.CESS061P	BP1	164734	5.436	1083.04	35231.422
DE SSTST.ESSTST.XGEN076A	BP11	164390	5.425	425.073	13827.676
DE SSTST.ESSTST.XGEN050A	BP2	162161	5.351	264.398	8600.899

Why the huge elapsed time difference 48Z to 48B?

This illustrates the objects with the highest I/O rates across the system, shows how many CPU seconds of CPU are caused by the I/O, and the application elapsed time effects of the I/O. The difference in IO is less than 10%, but more than twice the IO Elapsed seconds. The underlying reason for the huge difference between the IO elapsed times, is because of the DASD response times, as you will see on the next slide.

Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1e *aha!*

App Hit Ratio 93.2 System Hit Ratio 93.2 Read IO Rate/sec 263.94 Pages / Write 26.77 Reads For Seqpr 0 Writes Sync 0				Pages Read Sync 701022 Pages Read Seqpr 0 Pages Read Listpr 0 Pages Read Dynpr 0 Reads For Dynpr 0 Writes Asynch 25,272				Total Get Pages 10259741 Get Page Rand 10259741 Get Page Seq 0 Get Page RidList 0 Reads For Listpr 0 Updates 2,564,952				App Hit Ratio 93.7 System Hit Ratio 93.2 Read IO Rate/sec 245.42 Pages / Write 27.14 Reads For Seqpr 0 Writes Sync 0				Pages Read Sync 650072 Pages Read Seqpr 0 Pages Read Listpr 0 Pages Read Dynpr 55384 Reads For Dynpr 1,759 Writes Asynch 25,778				Total Get Pages 10355058 Get Page Rand 10355058 Get Page Seq 0 Get Page RidList 0 Reads For Listpr 0 Updates 2,564,998			
Avg Synth IO (ms) 2	Avg SP IO (Seq Pref) 0	Avg SP IO (Dyn Pref) 0	Avg SP IO (List Pref) 0	Avg Synth IO (ms) 1	Avg SP IO (Seq Pref) 0	Avg SP IO (Dyn Pref) 3	Avg SP IO (List Pref) 0																
Pages Written 676,577	Avg Sync Wrt 0	Avg Asynch Wrt 4	Pages Written 699,695	Avg Sync Wrt 0	Avg Asynch Wrt 3																		

XGEN048Z
XGEN048B

Why the huge elapsed time difference 48Z to 48B?

Both good DASD performance
Compare GP, I/O rate, Synch I/O Elapsed

Hit % & IO/Sec pretty close....

48B has a higher number of GP, a lower I/O rate, and lower Sync I/O times.

48B found more pages in the pool, and more pages in the DASD cache.

Synch IO is noticeably lower

Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1f

Buffer Pool Info

Name: BP11

Objects: 10

Buffers: 40000

Pages Freed: Y

Pool Mgt: LRU

Threshold

VPSEGT: 80

DwGWT: 30

VDwGWT: 5

Buffer Pool - BP11

Type of GetPage Access

Buffer Pool - BP11

Highest I/O Rates/Sec

Buffer Pool Info

Name: BP11

Objects: 10

Buffers: 40000

Pages Freed: Y

Pool Mgt: LRU

Threshold

VPSEGT: 80

DwGWT: 30

VDwGWT: 5

App Hit Ratio	88.8	Pages Read Sync	574222	Total Get Pages	5114518
System Hit Ratio	88.8	Pages Read Seqpr	0	Get Page Rand	5114518
Read IO Rate/Sec	192.63	Pages Read Listpr	0	Get Page Seq	0
Pages / Write	29.58	Pages Read Dynpr	0	Get Page RidList	0
Reads For Seqpr	0	Reads For Dynpr	0	Reads For Listpr	0
Writes Sync	0	Writes Asynch	18,422	Updates	1,278,739
Avg Synch IO (ms)	2	Avg SP IO (Seq Pref)	0	Avg SP IO (Dyn Pref)	0
Pages Written	544,859	Avg Sync Wrt	0	Avg Asynch Wrt	4

Batch job cycle, death by random I/O

If we have high hits in DASD cache, should we make pool larger?

Maybe, maybe not

BP11 is 100% random access, we can see the top 10 I/O objects in the pool, and then see the performance characteristics of each object. Overall avg. synch I/O times of 2 Ms, is very good, means almost all pages are found in the DASD cache. That also tells us that making the pool larger will find more pages in the BP, and reduce the I/O rate, and save CPU cycles.

Responsive Systems – Performance Software that Works!!

Pool performance analysis – after 1g

The image displays three screenshots of the Buffer Pool Tool for DB2, showing cluster performance analysis for BP2, BP9, and BP11. Each screenshot includes a 'Clusters' table and an 'Object Data' table. Red arrows point to specific data points in the tables. To the right of the screenshots are labels: Ramos, Samos, Groups, and WKSET.

BP2 Cluster Info:

Object	Smallest Ma	Largest Max
1	4734	6487
2	2729	4244
3	1049	2216
4	1	945

BP9 Cluster Info:

Object	Smallest Ma	Largest Max
1	7606	11793
2	15	3897

BP11 Cluster Info:

Object	Smallest Ma	Largest Max
1	6707	9420
2	4693	5221
3	13	2824

Working set size is often a crucial factor used for splitting objects into pools.

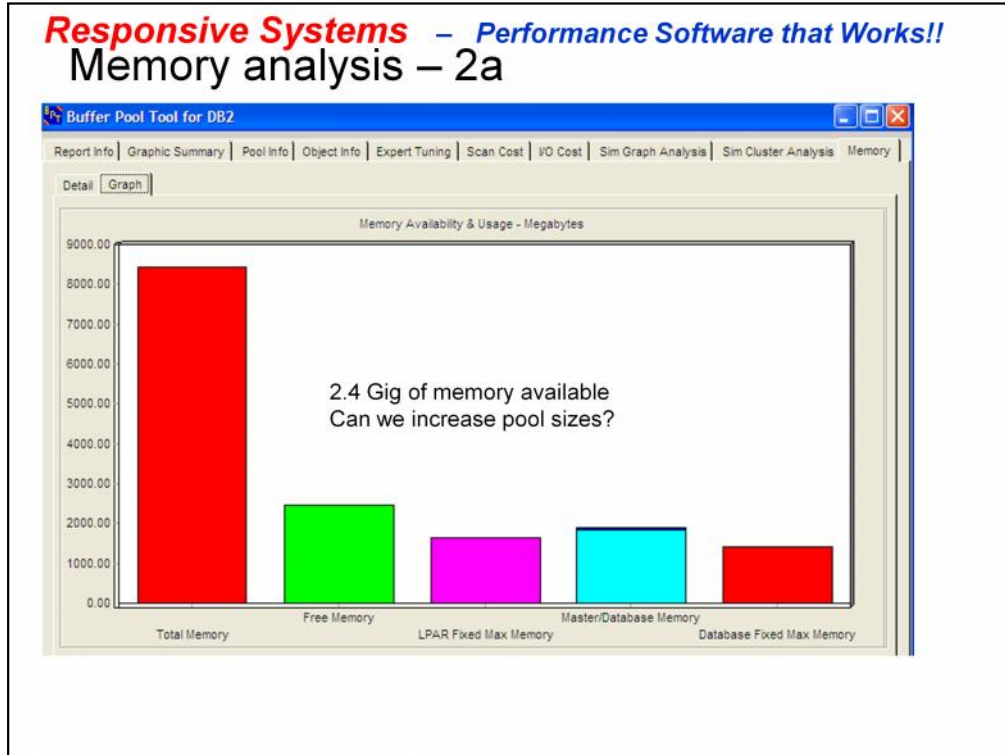
The general methodology, is Ramos/Samos – (randomly accessed, mostly), and (sequentially accessed mostly); then within those criteria, very large working set objects from the rest. The goal is to separate objects that monopolize a pool.

The first obstacle we encounter in this overall workload, is that it's almost all random.

Based on the working set sizes in each pool, there probably isn't much opportunity for gain by splitting out objects.

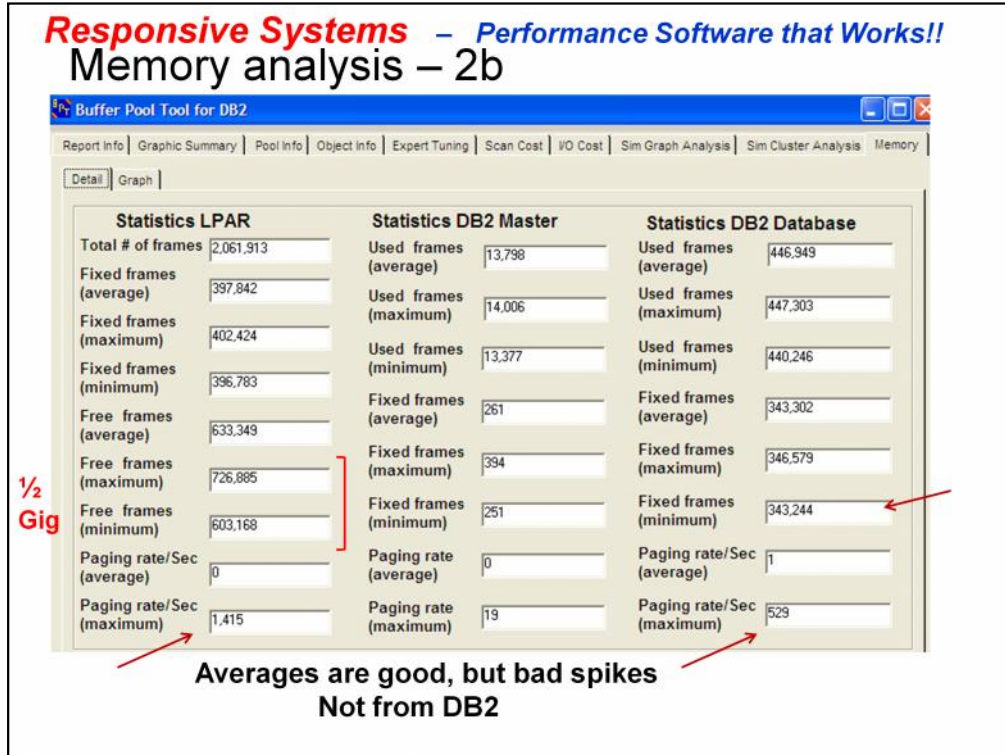
BP9 is a maybe, based on 4 objects in each cluster.

Responsive Systems – Performance Software that Works!! Memory analysis – 2a



There is 2 Gig of memory available on the LPAR, can we increase any pools?

Responsive Systems – Performance Software that Works!! Memory analysis – 2b



Even with 2.4 Gig available, the system has shown paging activity over the monitored interval. Pool increases might hurt overall performance.

A frame is 4K.

Responsive Systems – Performance Software that Works!!

Pool Pagefix CPU savings

- Expect about 8% CPU saving of IO cost
 - It's the cost of fixing and releasing the page in memory that causes the CPU cost
- It will vary depending upon your workload
 - A synch read is 1 page
 - A prefetch may actually read anywhere from 1 – 64 pages
 - A write *has to write* every page, but may have multiple IOs
- We don't have a lot of control over write activity - Pages/Write
- Pagefix the pools with high IO rates/second

Check memory availability first..

The number of pages actually read or written will determine the saving.

If a prefetch “can” read 64 pages.... If the pool is pagefixed, an actual read of 64 pages, will “save” more than a prefetch that only has to read 10...

Responsive Systems – *Performance Software that Works!!*

Pool Pagefix CPU savings

- Additional consideration
 - Size of the pool(s)
- Two pools with 500+ IO/Sec
 - One has 50,000 buffers
 - The other has 150,000 buffers
- Which would you Pagefix?
 - Biggest bang for the buck...?
 - How much avg **free** memory do you have?

Responsive Systems – Performance Software that Works!!
Pool Pagefix CPU savings

DATE	SRB CPU/TM	Prefetch Read I/Os	Total Sync Read I/O's	IOCOUNT	CPU Seconds	Prefetch CPU Cost per IO CPU Seconds	% Saving
	DBM1				DBM1		
28-Apr-08	0:54:04.31	91,066,400	41,435,109	132,501,509	3244	0.000035622	Base
29-Apr-08	0:57:23.67	98,631,004	41,811,306	140,442,310	3443	0.000034908	
Pools Pagefixed							
1-May-08	0:56:42.66	106,118,184	43,969,479	150,087,663	3402	0.000032059	10.00%
2-May-08	0:56:27.99	97,573,050	45,343,642	142,916,692			
7-May-08	0:53:48.67	90,425,961	42,672,240	133,098,201	3228	0.000035698	-0.21%
8-May-08	0:57:34.25	92,418,870	44,804,753	137,223,623			
12-May-08	1:02:57.66	134,339,741	47,452,124	181,791,865	3677	0.000027371	23.16%
13-May-08	1:09:30.06	157,009,921	47,800,608	204,810,529	4080	0.000025986	27.05%
15-May-08	1:01:08.25	128,107,975	44,278,455	172,386,430			
16-May-08	1:05:45.95	129,716,219	45,207,337	174,923,556	3945	0.000030413	14.63%
19-May-08	0:58:54.02	121,134,552	37,964,597	159,099,149			
20-May-08	1:05:05.63	130,500,306	44,076,187	174,576,493	3905	0.000029923	16.00%
21-May-08	1:02:24.31	123,246,902	41,336,231	164,583,133			
9-Jun-08	0:48:25.30	89,988,804	37,248,998	127,237,802	2905	0.000032282	9.38%
10-Jun-08	0:49:14.31	93,629,950	36,642,766	130,272,716	2954	0.000031550	11.43%
11-Jun-08	0:48:18.86	92,112,926	37,892,212	130,005,138	2898	0.000031461	11.87%
12-Jun-08	0:51:20.76	95,807,846	39,655,823	135,463,669	3080	0.000032148	9.75%

Prefetch is not the only user of SRB CPU in DBM1, is is the predominant cause.
Synch IO is charged to the application TCB

We normally estimate about 8% at the application level.

Responsive Systems – Performance Software that Works!!

LRU vs. FIFO Pool Management

- LRU – Least Recently Used
 - Queues have to be maintained for every **getpage**
 - FIFO – First In First Out
 - Eliminating queue management will save CPU
 - ***if*** it doesn't cause the IO rate to increase... Latch 14 overhead reduction
 - Can help for:
 - Objects that are pool resident – fit into the pool
 - Objects *very large & very random*, almost always an IO..
 - Does not occur too often, based on the system data I have seen
- One company switched **back** to LRU from FIFO, and saved 6% CPU

The number of pages actually read or written will determine the saving.

If a prefetch “can” read 64 pages.... If the pool is pagefixed, an actual read of 64 pages, will “save” more than a prefetch that only has to read 10...

The key to using FIFO is either:

- a. Objects have no IO
- b. Almost totally random, and rarely find a page in the pool – always do an IO

Responsive Systems – Performance Software that Works!!
Buffer Pool Data - statistics

BP1	GENERAL	QUANTITY	/SECOND	BP1	READ OPERATIONS	QUANTITY	/SECOND
	CURRENT ACTIVE BUFFERS	3014.44	N/A		BPOOL HIT RATIO (%)	13.07	
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00		GETPAGE REQUEST	7270.9K	2424.49
	NUMBER OF DATASET OPENS	0.00	0.00		GETPAGE REQUEST-SEQUENTIAL	3.00	0.00
	BUFFERS ALLOCATED - VPOOL	40000.00	N/A		GETPAGE REQUEST-RANDOM	7270.9K	2424.49
	DFHSM MIGRATED DATASET	0.00	0.00		SYNCHRONOUS READS	624.5K	208.24
	DFHSM RECALL TIMEOUTS	0.00	0.00		SYNCHRON. READS-SEQUENTIAL	1.00	0.00
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00		SYNCHRON. READS-RANDOM	624.5K	208.24
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00		GETPAGE PER SYN.READ-RANDOM	11.64	
	CONCUR.PREF.I/O STREAMS-HWM	0.00	N/A		SEQUENTIAL PREFETCH REQUEST	2.00	0.00
	PREF.I/O STREAMS REDUCTION	0.00	0.00		SEQUENTIAL PREFETCH READS	2.00	0.00
	PARALLEL QUERY REQUESTS	0.00	0.00		PAGES READ VIA SEQ.PREFETCH	61.00	0.02
	PARALL.QUERY REQ.REDUCTION	0.00	0.00		S.PRF.PAGES READ/S.PRF.READ	30.50	
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00		LIST PREFETCH REQUESTS	0.00	0.00
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00		LIST PREFETCH READS	0.00	0.00
	BP1 WRITE OPERATIONS	QUANTITY	/SECOND		PAGES READ VIA LIST PREFETCH	0.00	0.00
	BUFFER UPDATES	8674.5K	2892.52		L.PRF.PAGES READ/L.PRF.READ	N/C	
	PAGES WRITTEN	1662.7K	554.43		DYNAMIC PREFETCH REQUESTED	361.3K	120.47
	BUFF.UPDATES/PAGES WRITTEN	5.22			DYNAMIC PREFETCH READS	233.3K	77.80
	SYNCHRONOUS WRITES	1127.00	0.38		PAGES READ VIA DYN.PREFETCH	5696.0K	1899.35
	ASYNCHRONOUS WRITES	147.1K	49.06		D.PRF.PAGES READ/D.PRF.READ	24.41	
	PAGES WRITTEN PER WRITE I/O	11.22			PREF.DISABLED-NO BUFFER	0.00	0.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00		PREF.DISABLED-NO READ ENG	0.00	0.00
	VERTI.DEF.WRITE THRESHOLD	1593.00	0.53		PAGE-INS REQUIRED FOR READ	1270.00	0.40
	DM THRESHOLD	0.00	0.00				
	WRITE ENGINE NOT AVAILABLE	0.00	0.00				
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00				

Paging indicates that the overall system memory requirements are too high. Paging impacts the entire LPAR, not only DB2.

There are Synch writes, but the DM threshold has not been reached, so this isn't a problem.

Frequently Mis-Understood

Current Buffers Active

I have 150,000 buffers in the pool, but I'm only using 1763 of them.

Should I make the pool smaller?

The naming of this field is misleading. It is the number of buffers that are NOT available – pages that have been updated but not written yet, and pages on a read or write queue.

Responsive Systems – Performance Software that Works!!

Buffer Pool Performance Metrics

- Hit Ratio?
 - $(\text{Getpages} - \text{RIO}) / \text{Getpages}$
 - Mostly useless, only relevant from an application delay perspective
- System Hit Ratio
 - $(\text{Getpages} - \text{Sum of Pages Read}) / \text{Getpages}$
 - Interesting, historical
 - Can't convert it to a meaningful metric like CPU or elapsed
 - Dynamic prefetch will reduce it, can be negative
- Residency Time – useless for mixed pools, random & SP, and many objects – maybe for isolated object performance
- Miss Ratio? Re-read ratio? *Simply obfuscations...*
- IO Rate/Second – easily converts to CPU cost, and application delay relationships

It has been proven of dozens of performance studies that the IO rate/second is the only metric that can be converted into measurements that really matter to clients – CPU costs, and specific application delay factors.

Responsive Systems – Performance Software that Works!!

Negative System Hit Ratio

[Table Space] EBBPEPPE.GEBP.EBPPE001[0]

Collection

Date.....06-01-2012 DB2 SS.....DSP1 DB2 Ver.....9.1 New Function.....#NA# Get Pages.....67,989,641
 Time.....02:09:18 z/OS.....GEB DS Group....DSNP Lost Records.....744,411 Elapsed Time.....00:15:00

Attributes

Object Type.....Table Space Number of Partitions.....16
 Statistics Pool.....BP1 Current Partition.....0

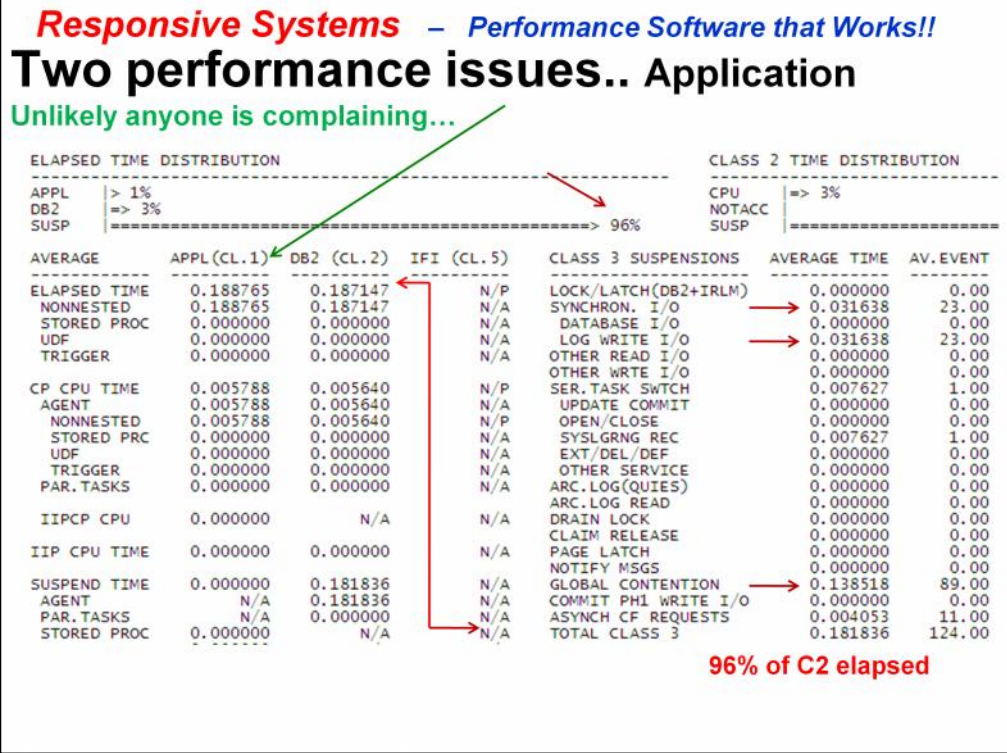
Get Pages.....405,898	451.0 /Sec	Pages Read.....572,517	636.1 /Sec
Get Pages - Random.....405,894	100.0 %	Pages Read - Synch.....215,208	37.6 %
Get Pages - Seq.....4	0.0 %	Pages Read - SPref.....90	0.0 %
Get Pages - RID List.....0	0.0 %	Pages Read - DPref.....357,219	62.4 %
		Pages Read - LPref.....0	0.0 %
		Pages Read per Seq I/O.....25.2	
Get Pages - Random Misses.....130,476	32.1 %	Read I/Os.....229,398	254.9 /Sec
Get Pages - Other Misses.....3	0.0 %	Read I/O - Synch.....215,208	93.8 %
Get Pages - Hits.....275,419	67.9 %	Read I/O - SPref.....4	0.0 %
Get Pages - No Read.....0	0.0 %	Read I/O - DPref.....14,186	6.2 %
System Hit Ratio.....-41.0 %		Read I/O - LPref.....0	0.0 %
Group BP Hits.....0	0.0 %	Synch I/O Delay - Avg.....13 mSec	3618 Max
Page Residency - Avg Seconds.....0		SPref I/O Delay - Avg.....10 mSec	21 Max
		DPref I/O Delay - Avg.....6 mSec	679 Max
		LPref I/O Delay - Avg.....0 mSec	0 Max

Dynamic prefetch is reading in thousands of pages that are never accessed by the application.

i.e. no getpage request issued for them.

So we see negative hit ratios at object levels and, of course, at the overall pool level.

Pool ratio can still be positive, although some objects are reducing the overall pool hit ratio.



Logging delays, and cross-system contention with another member in the data sharing group.

While the elapsed time is good, 96% is wait time. This is one of little item often overlooked during performance analysis.

Because the elapsed time is good, nobody looks for the delay problems.

Synch IO avg isn't bad, but on the high side for todays IO subsystems.

Responsive Systems – Performance Software that Works!!

Performance problem, emailed

Yesterday, we had a major incident in Production when we adjusted the Bufferpool sizes. At first, all went well. We resized one BP, waited for 15 minutes, no problem, resized the next and so on. After the fifth, all of a sudden, the whole system went nearly dead. Only with a high privileged user, we could log-on and undo the sizing. As soon as one of the Bufferpools was at its original size, within a few seconds, the system recovered and all works fine since then.

We detected no paging activity, just CPU at 100%, which means that more than 3000 MIPS were in use while we usually have peaks up to 2200 MIPS and an average usage of much less. We have assigned 42 GB of RAM to the LPAR, only 50% of it is really used and the resizing of the Bufferpools should have been from about 10 GB to 15 GB.

I had no idea why DB2 could be able to use 6 processors with highest priority at 100%.....

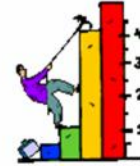
So, DB2 just ate your system. However, with the memory usage mentioned, and NO paging activity, it's unlikely that the pool increases caused the problem.

The fact that it went away after pools were decreased, is probably incidental luck.

Responsive Systems – Performance Software that Works!!

Performance problem, emailed

- We asked some basic questions, and asked for a DB2 Statistics report
- The next data was an RMF report
 - No paging activity, what-so-ever, nothing
 - All 6 engines very busy
 - Several devices with heavy I/O volume
 - Not paging or swap volumes
- Took them a week to get and send a DB2 Statistics report
 - So how important was problem resolution?
 - We see this same issue weekly on DB2L....



The only time I ever saw DB2 completely eat a machine was during a huge backout, and that was more than a decade ago..

Responsive Systems – Performance Software that Works!!

Performance problem, emailed

SUBSYSTEM SERVICES	QUANTITY	LOG ACTIVITY	QUANTITY	/SECOND
IDENTIFY	6189.00	READS SATISFIED-OUTPUT BUFF	67691.3K	817.53
CREATE THREAD	1464.6K	READS SATISFIED-OUTP.BUF(%)	38.30	
SIGNON	1455.4K	READS SATISFIED-ACTIVE LOG	109.1M	1317.22
TERMINATE	1473.9K	READS SATISFIED-ACTV.LOG(%)	61.70	
ROLLBACK	23038.00	READS SATISFIED-ARCHIVE LOG	0.00	0.00
		READS SATISFIED-ARCH.LOG(%)	0.00	
COMMIT PHASE 1	1452.5K			
COMMIT PHASE 2	322.1K	TAPE VOLUME CONTENTION WAIT	0.00	0.00
READ ONLY COMMIT	1232.0K	READ DELAYED-UNAVAIL.RESOUR	0.00	0.00
UNITS OF RECOVERY INDOUBT	0.00	ARCHIVE LOG READ ALLOCATION	0.00	0.00
UNITS OF REC.INDBT RESOLVED	0.00	ARCHIVE LOG WRITE ALLOCAT.	304.00	0.00
SYNCHS(SINGLE PHASE COMMIT)	367.4K	CONTR.INTERV.OFFLOADED-ARCH	9138.2K	110.37
QUEUED AT CREATE THREAD	0.00	LOOK-AHEAD MOUNT ATTEMPTED	0.00	0.00
SUBSYSTEM ALLIED MEMORY EOT	2251.00	LOOK-AHEAD MOUNT SUCCESSFUL	0.00	0.00
SUBSYSTEM ALLIED MEMORY EOM	0.00			
SYSTEM EVENT CHECKPOINT	381.00	UNAVAILABLE OUTPUT LOG BUFF	11.00	0.00
		OUTPUT LOG BUFFER PAGED IN	0.00	0.00
HIGH WATER MARK IDBACK	73.00	LOG RECORDS CREATED	156.7M	1892.84
HIGH WATER MARK IDFORE	28.00	LOG CI CREATED	9143.0K	110.42
HIGH WATER MARK CTHREAD	130.00	LOG WRITE I/O REQ (LOG1&2)	4285.9K	51.76
		LOG CI WRITTEN (LOG1&2)	20273.8K	244.85
		LOG RATE FOR 1 LOG (MB)	N/A	0.48
		LOG WRITE SUSPENDED	1321.9K	15.96

**What happens during a backout?
Synch IO backout, logging increases
50%..., checkpoints, filling logs, archiving,
etc**

A lot of rollbacks, and a huge number of reads from the Active Log. The read rate for the active logs is 1317/Second. That's a very high I/O rate. Note the number of checkpoints, archive log allocations and Cis

Responsive Systems – Performance Software that Works!!

Buffer Pool Part1

BP1	GENERAL	QUANTITY	/SECOND	/THREAD	/COMMIT	BP1	READ OPERATIONS	QUANTITY	/SECOND
	CURRENT ACTIVE BUFFERS	1037.85	N/A	N/A	N/A		BPOOL HIT RATIO (%)	53.81	
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	0.00	0.00		GETPAGE REQUEST	447.0M	5398.69
	NUMBER OF DATASET OPENS	3217.00	0.04	0.00	0.00		GETPAGE REQUEST-SEQUENTIAL	110.8M	1337.65
	BUFFERS ALLOCATED - VPOOL	232.3K	N/A	N/A	N/A		GETPAGE REQUEST-RANDOM	336.3M	4061.04
	DFHSM MIGRATED DATASET	0.00	0.00	0.00	0.00		SYNCHRONOUS READS	20658.6K	249.50
	DFHSM RECALL TIMEOUTS	0.00	0.00	0.00	0.00		SYNCHRON. READS-SEQUENTIAL	4244.1K	51.26
	VPOOL EXPANS. OR CONTRACT.	2.00	0.00	0.00	0.00		SYNCHRON. READS-RANDOM	16414.5K	198.24
	VPOOL OR HPOOL EXP. FAILURE	0.00	0.00	0.00	0.00		GETPAGE PER SYN.READ-RANDOM	20.49	
	CONCUR.PREF.I/O STREAMS-HWM	0.00	N/A	N/A	N/A		SEQUENTIAL PREFETCH REQUEST	2438.7K	29.45
	PREF.I/O STREAMS REDUCTION	0.00	0.00	0.00	0.00		SEQUENTIAL PREFETCH READS	1492.0K	18.02
	PARALLEL QUERY REQUESTS	0.00	0.00	0.00	0.00		PAGES READ VIA SEQ.PREFETCH	46269.7K	558.81
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	0.00	0.00		S.PRF.PAGES READ/S.PRF.READ	31.01	
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00	0.00	0.00		LIST PREFETCH REQUESTS	2793.2K	33.73
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00	0.00	0.00		LIST PREFETCH READS	290.5K	3.51
							PAGES READ VIA LIST PREFETCH	5613.4K	67.79
							L.PRF.PAGES READ/L.PRF.READ	19.33	
							DYNAMIC PREFETCH REQUESTED	20125.4K	243.06
							DYNAMIC PREFETCH READS	8848.1K	106.86
							PAGES READ VIA DYN.PREFETCH	133.9M	1617.50
							D.PRF.PAGES READ/D.PRF.READ	15.14	
							PREF.DISABLED-NO BUFFER	0.00	0.00
							PREF.DISABLED-NO READ ENG	25475.00	0.31
							PAGE-INS REQUIRED FOR READ	0.00	0.00

What are the problems in this data? What happens when you hit this threshold?

Dataset opens, Prefetch Disabled-No Read Engine.

Responsive Systems – Performance Software that Works!!

Buffer Pool Part2

BP1	WRITE OPERATIONS	QUANTITY	/SECOND	

	BUFFER UPDATES	57165.1K	690.40	
	PAGES WRITTEN	7549.6K	91.18	
	BUFF.UPDATES/PAGES WRITTEN	7.57		
	SYNCHRONOUS WRITES	17486.00	0.21	
	ASYNCHRONOUS WRITES	2356.3K	28.46	
	PAGES WRITTEN PER WRITE I/O	3.18		Low, but normal...
	HORIZ.DEF.WRITE THRESHOLD	25819.00	0.31	← Reverse this relationship
	VERTI.DEF.WRITE THRESHOLD	5658.00	0.07	←
	DM THRESHOLD	0.00	0.00	
	WRITE ENGINE NOT AVAILABLE	47.00	0.00	← Fix this problem...
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	

Probable cause of the problems here?

Write Engine Not Available. DWQT is high, VDWQT is low – we should see the opposite. Pages/Write is low single digit, set VDWQT=(0,140)

Responsive Systems – Performance Software that Works!!

Buffer Pool Tool for DB2 - BPO

Report Info | Graphic Summary | Pool Info | Object Info | Expert Tuning | Scan Cost | I/O Cost | Sim Graph Analysis | Sim Cluster Analysis | Memory

Collection

Date: 2009-09-23
Time: 09:30:55
Elapsed Time: 00:30:00

System Info

System: IPD4
Sub System: DB2P
DB2 Version: 8.1
DS Group: *NA*

Pool	RIO/Sec	Get Pages	Updates	Hit Ratio	I/O	WIO/Sec	Pages/Write	Write I/Os	Pages/Writer	Avg Pg Res	Sec F
BP0	0.03	6370	44	97.9	97	0.02	1.11	38	42		1762
BP3	385.30	23506149	83150	95.6	713183	10.91	1.41	19645	27616		1720
BP4	497.15	12967873	119714	74.3	918688	13.23	1.78	23810	42460		1337
BP5	133.00	13697098	1967150	87.2	356010	64.78	22.17	116611	2585363		1569
BP7	1,350.47	22903242	670556	57.6	2514277	46.35	2.19	83430	182314		1035
BP8	10.57	1313428	187	92.1	19104	0.04	1.79	77	138		1657
BP9	294.77	11405385	140453	88.7	544669	7.82	1.67	14078	23468		1596
BP10	241.02	5405145	5445	38	437243	1.89	1.29	3404	4386		683
BP11	756.41	33689287	665582	93	1483591	67.81	1.66	122052	202024		1674
BP16	110.27	2036555	1679	30.7	199440	0.53	1.49	948	1415		552
BP19	87.71	5129001	15979	59.9	160805	1.62	1.82	2920	5313		1078
BP21	4.01	992059	326	95.9	7412	0.10	1.32	186	245		1727
BP22	51.33	3369605	670082	97.2	283659	106.26	2.06	191274	393212		1750
BP2K	0.00	14897	7189	100	30	0.02	2.23	30	67		1800
BP2K1	0.00	6142	5958	100	0	0.00	0.00	0	0		1800
BP2K2	0.00	65790	29188	100	10022	5.57	1.90	10017	19034		1799
BP8K0	8.29	56904	0	41.1	14922	0.00	0.00	0	0		0
BP16K0	0.00	84	0	100	0	0.00	0.00	0	0		1799

Total 4K Buffers: 260,400

Total Read/Write I/O: 7,663,152

Overall Sys Hit Ratio: 80.50

Total Updates: 14,382,682

Total Get Pages: 136,564,014

Total I/Os per second: 4,257.31

Pages per write: 5.93

Create File | Performance Wizard | Print | < Back | Finish

When this is less than 10.0, set vdwt=(0,40) or (0,80) or (0,128)
depending upon number of objects in the pool... unavailable pool pages....

When a pool has a lot of objects, hundreds or thousands, lower is better. Think about the number of pages that might be updated in the pool, across all the objects, and have to be written at checkpoint.

Responsive Systems – Performance Software that Works!!

Buffer Pool DB2 V9

BF20	GENERAL	QUANTITY	/SECOND	/THREAD	/COMMIT	BF20	READ OPERATIONS	QUANTITY	/SECOND
	CURRENT ACTIVE BUFFERS	100.44	N/A	N/A	N/A		BPOOL HIT RATIO (%)	59.14	
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	0.00	0.00		GETPAGE REQUEST	234.1M	2715.21
	NUMBER OF DATASET OPENS	19795.00	0.23	0.01	0.00		GETPAGE REQUEST-SEQUENTIAL	104.6M	1213.33
	BUFFERS ALLOCATED - VPOOL	90000.00	N/A	N/A	N/A		GETPAGE REQUEST-RANDOM	129.5M	1501.88
	DFHSM MIGRATED DATASET	0.00	0.00	0.00	0.00		SYNCHRONOUS READS	10878.4K	126.17
	DFHSM RECALL TIMEOUTS	0.00	0.00	0.00	0.00		SYNCHRON. READS-SEQUENTIAL	29762.00	0.35
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00	0.00	0.00		SYNCHRON. READS-RANDOM	10848.6K	125.82
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	0.00	0.00		GETPAGE PER SYN.READ-RANDOM	11.94	
	CONCUR.PREF.I/O STREAMS-HWM	251.00	N/A	N/A	N/A		SEQUENTIAL PREFETCH REQUEST	842.0K	9.77
	PREF.I/O STREAMS REDUCTION	0.00	0.00	0.00	0.00		SEQUENTIAL PREFETCH READS	788.9K	9.15
	PARALLEL QUERY REQUESTS	88.00	0.00	0.00	0.00		PAGES READ VIA SEQ.PREFETCH	50206.6K	582.31
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	0.00	0.00		S.PRF.PAGES READ/S.PRF.READ	63.64	
	PREF_QUANT.REDUCED TO 1/2	0.00	0.00	0.00	0.00		LIST PREFETCH REQUESTS	370.6K	4.30
	PREF_QUANT.REDUCED TO 1/4	0.00	0.00	0.00	0.00		LIST PREFETCH READS	41866.00	0.49
							PAGES READ VIA LIST PREFETCH	1276.8K	14.81
							L.PRF.PAGES READ/L.PRF.READ	30.50	
							DYNAMIC PREFETCH REQUESTED	2916.9K	33.83
							DYNAMIC PREFETCH READS	1139.7K	13.22
							PAGES READ VIA DYN.PREFETCH	33286.0K	386.06
							D.PRF.PAGES READ/D.PRF.READ	29.21	
							PREF.DISABLED-NO BUFFER	0.00	0.00
							PREF.DISABLED-NO READ ENG	0.00	0.00
							PAGE-INS REQUIRED FOR READ	0.00	0.00

What important piece of information is missing ??

Note Pages for Prefetch read...

Time -- elapsed time for the statistics data !!!

Responsive Systems – Performance Software that Works!!

Buffer Pool DB2 V9

BP20	WRITE OPERATIONS	QUANTITY	/SECOND
-----	-----	-----	-----
	BUFFER UPDATES	18540.8K	215.04
	PAGES WRITTEN	1014.4K	11.77
	BUFF.UPDATES/PAGES WRITTEN	18.28	
	SYNCHRONOUS WRITES	8935.00	0.10
	ASYNCHRONOUS WRITES	218.9K	2.54
	PAGES WRITTEN PER WRITE I/O	4.45	
	HORIZ.DEF.WRITE THRESHOLD	227.00	0.00
	VERTI.DEF.WRITE THRESHOLD	184.4K	2.14
	DM THRESHOLD	0.00	0.00
	WRITE ENGINE NOT AVAILABLE	0.00	0.00
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00

This is what we like to see as a relationship... mostly VDWQT triggering

High numbers for VDWQT vs. DWQT

Almost all writes are triggered by VDWQT.

Responsive Systems – Performance Software that Works!!

Large DB2 Financial systems...

Total 4K Buffers 3,600,000	Total Read/Write IO	11,088,878	Total Get Pages	60,806,475
	Overall Sys Hit Ratio	92.03	Total IOs per second	13,107.42
	Total Updates	9,882,351	Pages per write	1.39

10 minutes, client A
365 million getpages/hour
13,107 IO/second
15 Gig of buffer pools allocated
Memory is already stretched a bit, with some small paging spikes

The big problem is on the next slide

Total 4K Buffers 4,642,000	Total Read/Write IO	5,818,560	Total Get Pages	224,178,137
	Overall Sys Hit Ratio	90.65	Total IOs per second	3,055.97
	Total Updates	6,318,543	Pages per write	2.27

30 minutes, client B
448 million getpages/hour
19 Gig of buffer pools allocated

Which sounds bigger – 500 million, or ½ billion?

Systems are constantly getting larger, and have amazing throughput levels.

Responsive Systems – Performance Software that Works!!

Client A problem...

App Hit Ratio	85	Pages Read Sync	96795	Total Get Pages	649174
System Hit Ratio	85.1	Pages Read Seqpr	0	Get Page Rand	649174
Read IO Rate/sec	114.41	Pages Read Listpr	0	Get Page Seq	0
Pages / Write	1.32	Pages Read Dynpr	0	Get Page RidList	0
Reads For Seqpr	0	Reads For Dynpr	0	Reads For Listpr	0
Writes Sync	2.622	Writes Asynch	252,217	Updates	105,398
Avg Synchron IO (ms)	10	Avg SP IO (Seq Pref)	0	Avg SP IO (Dyn Pref)	0
Avg Synchron IO (ms)	10	Avg SP IO (List Pref)	0	Avg SP IO (List Pref)	0
Pages Written	337,264	Avg Synchron Wrt	10	Avg Asynchron Wrt	15

DASD problem

Is PAV on and set correctly?

DASD Cache size?

One of the most heavily hit objects in the system

The client's huge performance problem is poor DASD performance.
10 Ms Synchron IO times, with a very high IO rate/second
Synchron write times are also poor.

DASD cache is severely under sized for the workload.

Responsive Systems – Performance Software that Works!!

Drilling down to performance problems... 1

BPT Statistics & Simulation Integrated A



Responsive Systems – Performance Software that Works!!

Drilling down to performance problems... 2

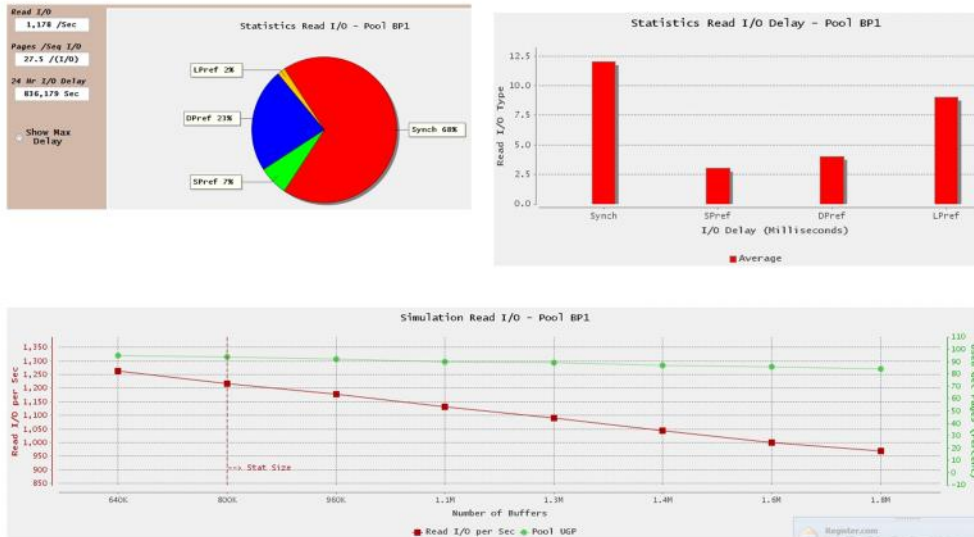
-----Pool Attributes-----					Sim. Reports	--Activity--		-----Read I/O-----	
Name	Size	PGFIX	PGSTEAL	VPSEQT	Valid [NV]	Get Page /Sec	I/O /Sec	Pages /Seq-I/O	
BP0	15,000	No	LRU	25 %	0 [1]	198 /Sec	1 /Sec	12.4 / I/O	
BP1	800,000	Yes	LRU	10 %	1 [0]	20,660 /Sec	1,178 /Sec	26.1 / I/O	
BP2	900,000	Yes	LRU	20 %	1 [0]	34,649 /Sec	545 /Sec	24.7 / I/O	
BP3	300,000	No	LRU	85 %	1 [0]	2,502 /Sec	9 /Sec	7.4 / I/O	
BP4	300,000	No	LRU	50 %	0 [1]	833 /Sec	3 /Sec	28.3 / I/O	
BP5	500,000	Yes	LRU	40 %	1 [0]	13,299 /Sec	365 /Sec	37.9 / I/O	
BP6	800,000	Yes	LRU	20 %	0 [1]	706 /Sec	31 /Sec	26.2 / I/O	
BP7	900,000	No	LRU	20 %	0 [1]	548 /Sec	2 /Sec	7.7 / I/O	
BP8	10,000	No	LRU	20 %	0 [1]	3 /Sec	1 /Sec	19.2 / I/O	
BP9	700,000	No	LRU	65 %	0 [1]	570 /Sec	0 /Sec	28.2 / I/O	
BP10	10,000	No	LRU	60 %	0 [1]	0 /Sec	0 /Sec	0.0 / I/O	
BP11	10,000	No	LRU	10 %	0 [1]	23 /Sec	0 /Sec	0.0 / I/O	
BP8K0	1,000	No	LRU	80 %	0 [1]	3 /Sec	1 /Sec	13.2 / I/O	
BP16K0	80,000	No	LRU	10 %	1 [0]	241 /Sec	21 /Sec	3.1 / I/O	
BP32K	62,000	No	LRU	40 %	1 [0]	1,295 /Sec	164 /Sec	6.5 / I/O	

**The client’s huge performance problem is poor DASD performance.
 10 Ms Synchron IO times, with a very high IO rate/second
 Synchron write times are also poor.**

DASD cache is severely under sized for the workload.

Responsive Systems – Performance Software that Works!!

Drilling down to performance problems... 3

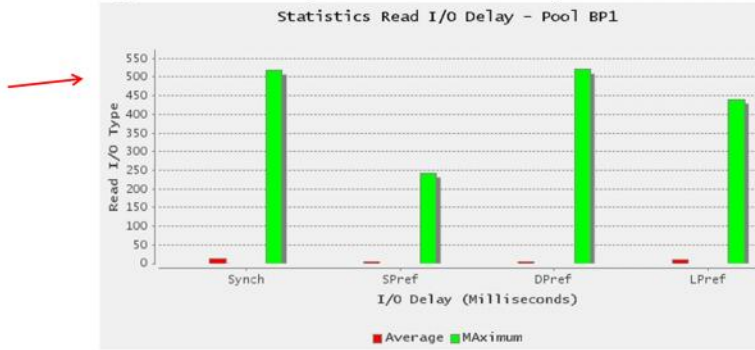


→ Integrated picture of a pool, IO Performance, and benefit of more memory

Here we can see the reduced IO rate/second if the pool size is increased.

Responsive Systems – Performance Software that Works!!

Drilling down to performance problems... 4



Not unusual to see sporadic "huge" IO waits.... As a max value

These huge Max IO waits are not unusual -- I see them in many systems.

You get a dasd controller cache miss, and your data is striped across several spinning disks in a Raid array....

Responsive Systems – Performance Software that Works!!

Drilling down to performance problems... 5

Pool Name	Activity /Sec	Get Page	Seq GP	Total	Head I/O /Sec	[24 Hour Delay]	in Seconds	Synch	[Delay]	SPref	SPref	LPref	Write I/O /Sec	Asynch	Synch	Working Set	Avg	Max	Delta	Type	NoPart	PartNo	Obj Name
BP1	451.0	0.0	254.6	[276,753]	239.1	[268,579]	0.0	15.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[1]	KB000000.GP0P.EBP00001	
BP1	32.2	0.0	23.2	[26,043]	23.1	[26,000]	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[16]	KB000000.GP0P.EBP00001	
BP1	48.4	0.0	20.4	[22,473]	19.2	[21,614]	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[11]	KB000000.GP0P.EBP00001	
BP1	31.7	0.0	19.8	[20,256]	19.3	[20,043]	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[1]	KB000000.GP0P.EBP00001	
BP1	27.4	0.0	19.2	[21,541]	19.2	[21,546]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[14]	KB000000.GP0P.EBP00001	
BP1	42.0	0.0	18.8	[20,736]	17.2	[19,310]	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[4]	KB000000.GP0P.EBP00001	
BP1	36.4	0.0	17.5	[19,278]	15.9	[17,875]	0.0	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[5]	KB000000.GP0P.EBP00001	
BP1	30.7	0.0	17.4	[18,808]	16.2	[18,159]	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[12]	KB000000.GP0P.EBP00001	
BP1	35.7	0.0	17.0	[18,001]	15.4	[17,294]	0.0	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[10]	KB000000.GP0P.EBP00001	
BP1	36.4	0.0	16.5	[16,294]	14.9	[15,489]	0.0	1.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[3]	KB000000.GP0P.EBP00001	
BP1	21.7	0.0	15.7	[19,033]	15.7	[19,021]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[15]	KB000000.GP0P.EBP00001	
BP1	31.4	0.0	15.7	[16,426]	14.1	[15,807]	0.0	1.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[6]	KB000000.GP0P.EBP00001	
BP1	140.4	0.0	120.5	[116,092]	92.3	[103,633]	0.0	28.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00008	
BP1	366.8	0.0	115.0	[91,707]	69.5	[72,066]	1.0	44.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00008	
BP1	451.7	0.0	82.2	[47,423]	31.8	[30,186]	0.0	30.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00008	
BP1	672.9	392.0	59.3	[35,886]	22.3	[23,075]	12.1	24.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00008	
BP1	143.0	0.0	49.2	[44,599]	33.8	[37,914]	0.0	15.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	16	[+]	KB000000.GP0P.EBP00002	
BP1	712.1	0.0	47.9	[33,226]	12.7	[12,076]	0.0	17.9	17.2	0.8	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00078	
BP1	447.0	166.0	47.7	[38,038]	36.9	[35,037]	8.9	1.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	TS	0		KB000000.GP0P.EBP00042	

Drilling into the pool, and sorting the objects by "IO Delay", then expanding the object to see the partitions in use, and their performance

Pool Size	GPU %	Elapsed Time	Pages Read /Sec	Read I/O /Sec	wSet	Avg %	Max %	Delta %
640,000	96 %	00:14:20	736.7 /Sec	260.5 /Sec	8 %	12 %	4 %	
800,000	95 %	00:14:11	722.4 /Sec	253.8 /Sec	8 %	12 %	4 %	
960,000	94 %	00:14:01	715.8 /Sec	247.7 /Sec	8 %	12 %	4 %	
1,120,000	93 %	00:13:49	707.6 /Sec	241.1 /Sec	8 %	12 %	4 %	
1,280,000	91 %	00:13:37	698.4 /Sec	234.4 /Sec	9 %	12 %	3 %	
1,440,000	90 %	00:13:25	687.8 /Sec	227.5 /Sec	9 %	12 %	3 %	
1,600,000	88 %	00:13:13	677.1 /Sec	221.0 /Sec	9 %	12 %	3 %	
1,760,000	86 %	00:12:58	669.3 /Sec	214.6 /Sec	9 %	12 %	4 %	

Working set size represented as a % of the pool size – so it grows as the pool increases, thus the IO rate/sec decreases

Responsive Systems – Performance Software that Works!!

Summary... As we started off

- Data is rarely as we want it...., or **when** we want it
 - Sometimes too much, sometimes too little
 - Collect it all, save it all
 - **Roll it up**, but beware of long averages...
 - Too much of short duration, we can summarize
 - Too little, or too long a duration, and it's useless
- Hindsight is always perfect, **if** you know what you want/need
 - *You must know where you have been, and where you are, to have a path into the future.*
 - *Is this a new performance problem, or was it poor for the last 6 months and somebody finally complained today?*

We need the ability, and tools get data, and to slice/dice data many different ways.
There are never enough tools.

HISTORY data.... Shows you where you were...

Responsive Systems – Performance Software that Works!!

Summary...

- Tuning will **save your company a lot of money**
 - It will justify your job forever (or until management changes...)
 - Reducing processor busy rates saves **Energy**
 - *Scott Hayes from DBI -- measurements*
- The proper tools make it possible for you to do your job effectively
 - *Time is money*
 - Increased throughput improves productivity
 - Both for internal and external clients
 - **This is money, and keeps your company in business**

You must know which tools you need, which tools are available – and the different pieces of performance perspective available from each tool.

Some tools may be redundant with others, may not add much, and consume CPU cycles for little benefit.

Responsive Systems – Performance Software that Works!!

Summary...

- Be able to predict the effect of pool changes on your IO rate/second – because the purpose of tuning is to reduce the IO....
 - Pool Size
 - Thresholds
 - Object placement
 - Moving objects into *other* or **NEW** pools
 - Proper object grouping is the key to performance
 - 3 or 4 monster pools won't cut the **mustard**
 - You don't need 50 pools either....

You must know which tools you need, which tools are available – and the different pieces of performance perspective available from each tool.

Responsive Systems – Performance Software that Works!!

Summary...

- Youth learns

I had someone who tried to add x'01' to x'6F' and got x'6G'..

- Age understands

- Depth of experience goes a long way
- IMO -- Mainframe knowledge is decreasing rapidly

- Nobody knows everything

*Be curious,
opportunity is
everywhere*

- Nobody has seen everything

- A day without learning, is a wasted day

Finding something new, the cause of a problem, that sudden flash of insight – is a wonderful thing!!

Responsive Systems – *Performance Software that Works!!*

I WENT TO A BOOKSTORE AND ASKED THE SALESWOMAN,
"WHERE'S THE SELF- HELP SECTION?"

SHE SAID IF SHE TOLD ME, IT WOULD DEFEAT THE PURPOSE.

Thank you for coming

joel@responsivesystems.com

www.responsivesystems.com

732-972-1261

Responsive Systems – Performance Software that Works!!

Joel Goldstein
Responsive Systems
joel@responsivesystems.com

Thank you for attending today !

The Performance
Wizard is here, he
lives in the Buffer
Pool Tool !!!

