PROGRESS SOFTWARE

DB-18: Advanced Performance Statistics

Adam Backman
President
White Star Software
adam@wss.com

Exchange 07

Agenda

- Performance, Performance, Performance
- Logging
- Space Allocation
- Data Utilization
- Memory Stuff
- Latches

Disclaimer

Although I have extensive experience in the field, the opinions in this presentation are just that, opinions. I encourage everyone to test any change prior to moving it into their production environment.

Logging

- The before image file
  - Cluster size
  - Page writers
  - BI Grow
- Logging buffers
- Checkpoints

What is a Checkpoint?

- A synchronization point between memory and disk
  - The initial checkpoint schedules modified buffers (-B) to be written prior to the next checkpoint
  - Subsequent checkpoints first check to make sure all of the previously scheduled buffers have been written and then schedules
- The goal is to have “just zero” buffers to be written at checkpoint.

Checkpoint Efficiency

- The before image cluster size determines the frequency of checkpoints
- The ability APWs to write all of the modified blocks determines the efficiency
- Buffers flushed at checkpoint can be caused by a disk bottleneck as well as improper settings of the cluster size and number of APWs

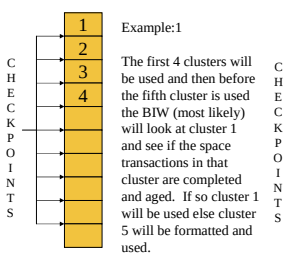
Tuning APWs

- Get the BI cluster size right
- Start with 1 APW
- If you see buffers flushed at checkpoint increasing during the “important” portion of the day then add one more APW
- If adding more APWs does not help look at the disk where the before image file(s) reside(s)

I am using multi-volume BI files why should I pre-grow my bi file?

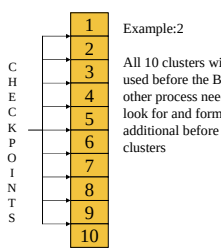
- The BI grow utility formats the bi clusters so the users will avoid this overhead
- The BI file will not need to check to see if it should reuse space until all of the formatted clusters are used
- It should promote more sequential access to the bi file which is good for performance

How the growth of the BI works



Example:1

The first 4 clusters will be used and then before the fifth cluster is used the BIW (most likely) will look at cluster 1 and see if the space transactions in that cluster are completed and aged. If so cluster 1 will be used else cluster 5 will be formatted and used.



Example:2

All 10 clusters will be used before the BIW or other process needs to look for and format additional before image clusters

I/O operations to the logs

- The AI and BI files are buffered but tightly managed to ensure data integrity
- The AI and BI block sizes should be the same
- Block sizes should match or be a multiple of the operating system block size
- The -aibufs parameter should be slightly higher than the -bibifs parameter to avoid unneeded writes to the BI file due to the lack of AI buffers (Theoretical Issue: no hard data)

Space Allocation

- Area basics
- Online DB Extends
- Toss your record manager blocks

Area Basics

- Use 8k block size (4k for NT)
- Use Type II Areas
- Manage records per block properly
- Pre-extend your areas to accommodate growth

Online Data Extent Extend

- New prostrct option addonline
- New areas cannot be added online to databases with after imaging enabled
- Ensure the permissions for the newly created extent(s)/area(s) allow for client access
  (Unix: umask 0)
- Run a verification pass before adding the new extent(s)/area(s)

Toss limits

- What kinds of data blocks are there?
- What are toss limits?
- What are create limits?
- When would I change these limits?
- How would I change these limits?

Types of Data Blocks

- Empty Block - Block in the database but above the high water mark for the area
- Free Block - Unaffiliated block below the high water mark
- RM Block - “Full” data block
- RM Chain Block - Non-”Full” data block
- IX Block - Index block

  Note: IX blocks are never considered “full”

What are Toss and Create Limits

- Toss Limit - The minimum amount of space in a block needed to remain on the RM chain
- Create Limit - The minimum amount of space in a block required to allow the creation of a new record in the block

Why Would You Change the Toss and Create Limits?

- If you see a significant amount of “Move rm front to back” on the Progress Monitor

  R&D -> Activity -> Space Allocation

And

- If you have a long RM chain as seen in chain analysis or VSTs
  - VSTs will only tell you how many RM Chain blocks there are in an area
  - Chain Analysis will tell how much room is in each block

Space Allocation

04/09/07 17:33:08 **Activity: Space Allocation**
04/07/07 03:13 to 04/07/07 09:32 (6 hrs 19 min)

	Total	Per Min	Per Sec	Per Tx
Database extends	0	0	0.00	0.00
Take free block	7051	19	0.32	0.00
Return free block	612	2	0.03	0.00
Alloc rm space	416824	1100	18.33	0.10
Alloc from rm	412141	1087	18.12	0.10
Alloc from free	4683	12	0.20	0.00
Bytes allocated	45254736	119406	1990.10	0.56
rm blocks examined	479314	1265	21.08	0.11
Remove from rm	47849	126	2.10	0.01
Add to rm, front	44815	118	1.97	0.01
Add to rm, back	3030	8	0.13	0.00
Move rm front to back	19319	51	0.85	0.00
Remove locked rm entry	194664	514	8.57	0.04

Determining number of RM Blocks through VSTs

```
FOR EACH _AreaStatus:
  DISPLAY _AreaStatus-AreaName
          _AreaStatus-Rmnum.
END.
```

Note: The BI file and any AI files will return ? as these areas do not contain RM Blocks

Chain Analysis

proutil <dbname> -C chanalys > <outputfile>

Sample Output:

```
RM CHAIN ANALYSIS
--------------------------------
            LIST OF RM CHAIN BLOCKS
                    free     # free
      dbkey        space    slots      hold
    4194304         3685       27         0
       ...
# blocks found in the RM chain.
```


Data Utilization

- Analyzing database analysis
  - Scatter
  - Fragmentation
- Table Statistics
  - What to look for?
  - How to do analysis?
- Index Statistics
  - Block splits and their effect

DB analysis

- Command line tool
  proutil <dbname> -C dbanalys
- Pros:
  Only way to determine DB storage efficiency
  Low impact on the system
  Can be run while the system is up
- Cons:
  No historical storage of information
  No flexibility of output

Database analysis

- Useful tool for determining low level storage information
- Helpful for determining records per block in storage areas
- Help determine when to compress/rebuild indexes

Using DB analysis

- proutil <dbname> -C dbanalys > output_file
- Output contains multiple sections
- We only care about two sections
  - Record block summary
  - Index block summary

Interpreting the data - DB analysis

- Should I do a chain analysis, a index analysis or both?
- How often should I run the utility?
- What does the data really mean?
- How can I use it to take action?

Example: Record Block Summary

RECORD BLOCK SUMMARY FOR AREA "Archived Ratings" : 151

Table	Records	Size	Min	Max	Mean	Count	Factor	Factor
PUB.hsrad	68884747	4.0G	53	68	62	68884747	1.0	2.0
PUB.hsrat	353726306	54.1G	120	210	164	353726306	1.0	1.2
Subtotals:	22611053	58.1G	53	210	147	422611053	1.0	1.4

RECORD BLOCK SUMMARY FOR AREA "Archived Comps" : 152

Table	Records	Size	Min	Max	Mean	Count	Factor	Factor
PUB.hscma	12675	549.7K	43	46	44	12675	1.0	1.5
PUB.hscmd	99680152	6.0G	56	99	64	99680152	1.0	1.5
PUB.hscmp	41122686	7.0G	98	301	183	41122686	1.0	1.5
PUB.hsgft	9217509	793.0M	67	104	90	9217509	1.0	2.4
Subtotals:	50033022	13.8G	43	301	98	150033022	1.0	1.6

Example: Index block summary

INDEX BLOCK SUMMARY FOR AREA "Archived Ratings" : 151

Table	Index	Fields	Levels	Blocks	Size	% Util	Factor
PUB.hsrad							
prradstat	18	3	3	4248	21.6M	5.2	1.7
prrat	19	4	3	205141	1.4G	90.3	1.2
PUB.hsrat							
audit	13	3	4	397168	3.0G	99.9	1.0
cmday	14	3	4	421648	2.0G	62.3	1.8
cmtbl	45	4	4	1080387	4.2G	50.8	2.0
kkcus	15	3	4	842577	4.2G	66.1	1.7
prrat	12	2	4	441148	3.0G	89.0	1.2
sltrn	16	3	4	449431	2.9G	85.3	1.3
sylic	17	4	4	471460	3.2G	88.1	1.2

Memory Stuff

- Client settings
  - -Bt
  - -tmpbsize
  - -Bp
  - -Mm
- Broker settings
  - -B
  - -semsets
  - -Mm

Client side parameters

- -Bt - Buffers for TEMP-TABLES
  - This will allocate blocks for temporary table caching which will reduce/eliminate disk I/O
- -tmpbsize - Block size for TEMP-TABLE buffers
  - This changes the allocation from the default of 1k per -Bt to 8k (If set to 8). This allows I/O operations to be done in larger chunks which is generally more efficient

Client Side Parameters - continued

- -Bp - Private Database Buffers
- Not really private
- Uses existing -B space
- Uses separate LRU chain
- This allows READ ONLY reports to use and reuse the same buffer space rather that evicting other buffers
- The total of all -Bp paramters cannot exceed 25% of -B

Client and Server Side Parameters

- -Mm - Message buffer size.
- Most application benefit from a larger setting of -Mm (1024 is the default)
- Increasing -Mm does not mean that every message sent will be -Mm size. -Mm just regulates the largest message that can be sent
- Both the client and every server that the client is connected to must have the same -Mm

Broker Side Parameters - -B

- -B - Database buffer pool
- Each -B equals 1 database block not 1k
- 90% Buffer hit rate sounds good
- 10 physical reads for every 100 requests
- Each Disk can support 100 I/O operations per second
- At 90% you can support 1000 requests per second per disk
- At 95% you can support 2000

Broker Side Parameters - -semsets

- This increases the number of semaphore sets available to the broker
- The manuals say to use it if your user counts are greater than 1000
- I believe you should use this parameter if your user count is higher than 100
- Low cost potentially high gain

Latches

- What are latches?
- Managing -spin
- TXE latch - Why is this different?

What are latches?

- Latches are used to manage concurrency within the shared memory portion of the database

Now that we have that cleared up let's move on

What are latches?

- Each resource in memory has one or more latches associated with it.
- The latches ensure that one user does not overwrite another user's changes
- Example: I want to add a buffer in the -B and Bob wants to modify the same block. With multiple CPUs this would be possible but the latches will keep the second user from making a change until the first change is complete

-spin

- -spin determines the number of times a process will retry to obtain a resource before relinquishing the CPU
- If you have multiple CPUs you should always set -spin. In version 10 -spin is set to a non-zero number by default but it used to default to 0
- -spin 0 tells the broker to use semaphores rather than latches to maintain concurrency

-spin - continued

- Every theory about -spin is wrong (Use x per CPU theories do not apply here)
- The idea is to avoid “naps”. A nap is done at the end of a spin cycle if the resource in not obtained
- The symptom of high nap counts it high system time on the CPU
- The high system time is caused be the overhead of moving processes on and off the run queue without getting any real work done

TXE - Why is it different?

- Tied to physical activity versus logical activity
- It is also tied to application code more tightly
  - More smaller transactions are generally easier on the TXE latch than fewer large transactions
- Database fragmentation causes higher TXE utilization
- RM Chain manipulation causes TXE utilization

Conclusion

- Try to push the performance bottleneck from the slowest resource to the fastest
- Do not focus solely on the performance numbers. It is very important to understand the impact on the application
- Constant maintenance will help maintain good performance (indexes, scatter, fragmentation)

Conclusion - continued

- Try to reduce the impact of reporting through the use of -Bt and -Bp
- Logging can have a tremendous effect on performance if not setup properly
- Keep an eye on your CPU for signs of poor latch utilization
- After all is said, application efficiency is still the largest contributor to overall system efficiency

Questions?

Adam Backman

adam@wss.com

Thank you for your time!