[Informix Tuning Basics](http://www.informix-dba.com/p/informix-innovator-c-tuning-basics.html)

# Increase Buffers to Reduce Disk I/O

End Result:

* Buffers increased from default value of 50000
* LRUs increased from default value of 8

1. Determine how much memory you want Informix to allocate

If you are running Innovator-C this maximum is 2 GB, other Informix editions may have different restrictions.

You will also be restricted by the actual amount of memory available on your system. Some memory will be needed for the OS and some will be needed by the other applications running on your system. Identify the total memory footprint you want Informix to have and set the ONCONFIG parameter SHMTOTAL to this value in KB.

informix> vi $INFORMIXDIR/etc/$ONCONFIG

SHMTOTAL 2097152

1. Determine how much of this memory you want Informix to allocate to Buffers.

Informix will cache the data that is on disk in Buffers reducing I/O and improving your performance. Don't be stingy. A decent starting point is 50%, you can adjust this later based on monitoring actual server activity.

Buffer Memory = SHMTOTAL \* Buffer Percent of Memory Buffer Memory = 2097152 KB \* 0.5 = 1048576 KB

1. Calculate the number of 2 KB Buffers

Number of Buffers = 1048576 KB / 2 KB = 524288 Buffers

1. Pick a starting point for number of LRU Queues. I like to use 8322 Buffers per LRU, which gives 63 for 524288 Buffers.

Number of LRU Queues = Number of Buffers / 8322

Number of LRU Queues = 524288 / 8322 = 63.0002 = 63 LRU Queues

1. Change the ONCONFIG BUFFERS parameter and bounce the engine to make change take effect

informix> vi $INFORMIXDIR/etc/$ONCONFIG

BUFFERPOOL size=2k,buffers=524288,lrus=63,lru\_min\_dirty=50,lru\_max\_dirty=60 informix> onmode -ky

informix> oninit -v

<="" a="" style="color: rgb(143, 102, 102); text-decoration: none;">

# Monitor and Tune Buffers

End Result:

* + Determine if too many or not enough buffers are configured

1. Do not attempt to tune buffers until your engine has been running under normal load for some time.

At engine startup the buffer pool is empty and pages are brought into memory over time as SQL requests are processed. Attempting to tune the buffer pool too soon after engine startup will lead to inaccurate results.

1. Determine if too many buffers are configured by using onstat -P | tail get an summary of what is in the buffer pool

|  |  |  |  |
| --- | --- | --- | --- |
| 7340033 4 | 0 | 0 | 4 0 |
| 7340034 3967 | 0 |  | 3967 0 0 |

What is important here is the last 3 lines of the onstat -P output. This is telling me that 55.09% of my buffers contain data pages, 24.59% of my buffers contain index pages and 20.32% of my buffers contain other pages. A large portion of the other pages will be unused buffers.

informix> onstat -P

Totals: 1048576 362651 682521 3404 7958

Percentages: Data 55.09

Btree 24.59

Other 20.32

From this information I see that 20% of my buffer pool is going unused and I can safely reduce the number of buffers by 20%.

1. Determine if not enough buffers are configured by executing onstat -p to get the Read Cache Hit %

The Read Cache Hit % is the percentage of times a page on disk was needed and was already in the buffer pool and is the first %cached column of the onstat -p output.

846929

69665

1667

5239

idx-RA da-RA RA-pgsused lchwaits

ixda-RA 62769

gp\_read gp\_write gp\_rewrt gp\_del gp\_alloc gp\_free gp\_curs

0 0 0 0 0 0 0

ovlock ovuserthread ovbuff usercpu syscpu numckpts flushes

0 0 0 11955.10 906.98 30 16

bufwaits lokwaits lockreqs deadlks dltouts ckpwaits compress seqscans 7280 300 2020820205 0 0 10 449429 208087

commit rollbk

443270 48433 721760 218019

2473713077 116899849 241695271 1638363472 1447480

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line (Prim) -- Up 4 days 21:36:22 -- 1510960 Kbytes

Profile

dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached 363387 10087525 2903812263 99.99 1122317 1294778 14678622 92.35

isamtot open start read write rewrite delete

In the above output my Read Cache Hit % is 99.99%, hard to improve on that. Anything above 95% is great, anything above 90% is good and anything below 90% is a good indication of needing more buffers.

If adding more buffers does not improve your Read Cache then it is possible you have some poorly optimized SQLs that could be using sequential scans which can flood the buffers with a lot of pages that are not reused. If this is the case look at creating necessary indexes to avoid sequential scans or look into using light scans to avoid using the buffer pool for these sequential scans.

You can also utilize the Read Cache Hit % to determine if too many buffers are configured. If you need to reduce the amount of shared memory used by Informix you can most likely reduce the number of buffers if your hit rate is very high. This will result in poorer performance but as long as you keep it above 95% you should be OK.

# Monitor LRU Queues

End Result:

* Determine if enough LRU Queues are configured

1. Calculate Bufwaits Ratio (courtesy of Art Kagel) using bufwaits, pagreads and bufwrits from the onstat -p output

846929

69665

1667

5239

idx-RA da-RA RA-pgsused lchwaits

ixda-RA 62769

gp\_read gp\_write gp\_rewrt gp\_del gp\_alloc gp\_free gp\_curs

0 0 0 0 0 0 0

ovlock ovuserthread ovbuff usercpu syscpu numckpts flushes

0 0 0 11955.10 906.98 30 16

bufwaits lokwaits lockreqs deadlks dltouts ckpwaits compress seqscans 7280 300 2020820205 0 0 10 449429 208087

commit rollbk

443270 48433 721760 218019

2473713077 116899849 241695271 1638363472 1447480

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line (Prim) -- Up 4 days 21:36:22 -- 1510960 Kbytes

Profile

dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached 363387 10087525 2903812263 99.99 1122317 1294778 14678622 92.35

isamtot open start read write rewrite delete

bufwaits = 7280

pagreads = 10087525

bufwrits = 14678622

Bufwaits Ratio (BR) = (bufwaits / (pagreads + bufwrits)) \* 100

Bufwaits Ratio (BR) = (7280 / (10087525 + 14678622)) \* 100 = 0.029%

1. Increase LRU Queues by 10% until BR is less than 7% or the maximum number of LRUs are configured Maximum LRU Queues for 32 bit engines is 128

Maximum LRU Queues for 64 bit engines is 512

# Monitor and Tune Disk Writes

End Result:

* + Maximize Checkpoint writes
  + Minimize Foreground writes

1. Use onstat -F output to determine if any Foreground writes are taking place

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| db957880 | 0 | I | 0 | 0 | 366 | 425136 425380.489 |
| db9580d8 | 1 | I | 0 | 0 | 362 | 424787 424905.846 |

In the above output there have been 1093 Foreground (FG) writes, 98032 LRU writes and 208755 Chunk writes (aka Checkpoint writes).

informix> onstat -F | head

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line (Prim) -- Up 4 days 22:09:56 -- 1510960 Kbytes

Fg Writes 1093

LRU Writes Chunk Writes

98032

208755

address

flusher state data # LRU Chunk Wakeups Idle Tim

1. If Foreground writes are occurring then you need to increase LRU writes to keep some free buffers (aka buffers that do not need to be sync'd to disk at some point) in the LRU queues.

Reduce the LRU Max Dirty and LRU Min Dirty by modifying the BUFFERS ONCONFIG parameter until FG writes no longer occur.

1. If Foregound writes are not occurring then you may be interested in increasing LRU Max Dirty and LRU Min Dirty to reduce the number of LRU writes and increase the number of Chunk writes (which are more efficient) at checkpoint time.

One word of caution. Even though Checkpoint writes are more efficient than LRU writes you must understand what is happening when you maximize Checkpoint writes. When a checkpoint occurs there is a lot of I/O as Informix writes out all of the modified pages to disk. This I/O results in longer run times for SQLs executing during the checkpoint (hey, at least they aren't blocking anymore!) Moving the maximum number of writes to checkpoint time will increase the total number of transactions you can perform because the engine will be more efficient, but the checkpoint times will be longer causing more SQLs to run during the slower period.

You might not run into the problem at all, it is just something to be aware of when configuring writes.

# Initially configure CPUVPs

End Result:

* + CPU Virtual Processors initially configured based on your hardware
  + Assumes all CPU resources can be consumed by Informix. If this is not the case, reduce CPUVPs accordingly.

1. Identify the number of physical CPU cores. Do not count a Hyperthreaded CPU as having 2 cores.
2. Identify the maximum number of CPUVPs you can configure in your Informix Edition
   * Innovator-C: 4 CPUVPs
   * Choice: 8 CPUVPs
   * Growth: 16 CPUVPs
   * Ultimate: No limit
3. Configure MULTIPROCESSOR ONCONFIG parameter 1 core systems: set MULTIPROCESSOR to 0.

Everything else: set MULTIPROCESSOR to 1.

1. Configure SINGLE\_CPU\_VP ONCONFIG parameter

1 and 2 core systems: set SINGLE\_CPU\_VP to 1. Everything else: set SINGLE\_CPU\_VP to 0.

1. Configure number of CPUVPs using VPCLASS ONCONFIG parameter 1 and 2 core systems: Number of CPUVPs = 1

Everything else: Number of CPUVPs = Number of cores - 1. Do not exceed the limits of your Informix Edition.

For an 8 core system running Innovator-C, configure 4 CPUVPs

VPCLASS cpu,num=4,noage

1. Shutdown and restart Informix for changes to take effect.

# Monitor and Tune CPUVPs

End Result:

* + Number of CPUVPs tuned to match Informix workload and CPU resources

1. After Informix has been running under normal load for a few days use onstat -g glo to monitor the efficiency of each CPUVP.

# WARNING: onstat -g glo produces incorrect efficiency in versions prior to 12.10.xC6 if onstat -z is used to clear statistics!

Efficiency is the percentage of time a VP's threads were actually running on a CPU vs the time they where scheduled to run by Informix. A high efficiency means Informix is not waiting for the CPU to become available to do work. The efficiency is contained in the 'Individual virtual processors' output on onstat -g glo.

informix> onstat -g glo

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line -- Up 2 days 01:00:48 -- 1410736 Kbytes

...

Individual virtual processors:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| vp | pid | class | usercpu syscpu total Thread Eff | | | |
| 1 | 7461 | cpu | 8012.55 110.13 8122.68 8291.89 97% | | | |
| 2 | 7464 | adm | 0.06 | 1.21 | 1.27 | 0.00 0% |
| 3 | 7486 | lio | 5.57 | 22.67 | 28.24 | 283.83 9% |
| 4 | 7495 | pio | 0.02 | 0.31 | 0.33 | 8.79 3% |
| 5 | 7497 | aio | 5.78 | 27.54 | 33.32 | 114.46 29% |
| 6 | 7499 | msc | 0.00 | 0.00 | 0.00 | 0.02 0% |
| 7 | 7501 | fifo | 0.00 | 0.00 | 0.00 | 0.00 0% |
| 8 | 7503 | aio | 0.95 | 2.12 | 3.07 | 26.92 11% |
| 9 | 7526 | soc | 0.18 | 0.36 | 0.54 | NA NA |
| 10 | 7538 | aio | 0.05 | 0.52 | 0.57 | 31.09 1% |

11

12

13

14

15

7539

7540

7541

29965

13400

tot

aio 0.05 0.59

aio 0.04 0.14

aio 0.03 0.49

aio 0.03 0.10

aio 0.03 0.34

8025.34 166.52

0.64

0.18

0.52

0.13

0.37

8191.86

27.98

13.48

16.06

15.82

20.00

2%

1%

3%

0%

1%

1. If your efficiency is high you might benefit from additional CPUVPs.

To determine if additional CPUVPs will improve performance you must monitor the CPUVP ready queue. This is where threads are placed when they are ready to be run by a CPUVP. Threads consistently in the ready queue indicate there are not enough CPUVPs to handle the current workload.

You monitor ready queue with the onstat -g rea command. Use onstat -r 1 -g rea to display the ready queue every second. A system that might benefit from more CPUVPs will look like this:

informix> onstat -r 1 -g rea

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line -- Up 2 days 01:12:27 -- 1418928 Kbytes Ready threads:

tid tcb rstcb prty status vp-class name

292 e41290d0 e12f9f30 1 ready 1cpu sqlexec

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line -- Up 2 days 01:12:28 -- 1418928 Kbytes

Ready threads:

tid tcb rstcb prty status vp-class name

290 e4127afe e12f2b40 1 ready 1cpu sqlexec

292 e41290d0 e12f9f30 1 ready 1cpu sqlexec

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line -- Up 2 days 01:12:29 -- 1418928 Kbytes Ready threads:

tid tcb rstcb prty status vp-class name

290 e4127afe e12f2b40 1 ready 1cpu sqlexec

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line -- Up 2 days 01:12:30 -- 1418928 Kbytes Ready threads:

tid tcb rstcb prty status vp-class

294 e412bce3 e123da45 1 ready

290 e4127afe e12f2b40 1 ready

name

1cpu sqlexec

1cpu sqlexec

1. If your efficiency is low you may have too many CPUVPs configured or your CPUs can not handle the load

Reduce the number of CPUVPs, add additional CPUs or move applications that are also running on this server and taking CPU resources from Informix to a different server.

# Initially configure AIOVPs

End Result:

* + AIO Virtual Processors initially configured based on your hardware

1. If Kernel Asynchronous IO is enabled

informix> onstat -g iov

IBM Informix Dynamic Server Version 12.10.UC7IE -- On-Line (Prim) -- Up 5 days 11:11:11 -- 1510960 Kbytes AIO I/O vps:

class/vp/id s io/s totalops dskread dskwrite dskcopy wakeups io/wup errors tempops

fifo 9 0 i 0.0 0

msc 8 0 i 0.2 343147

0 0

0

0

0

0 0.0 0 0

0 343081 1.0 0 343170

aio 7 0 i 62.0 4761188 3052788 102515

aio 10 1 i 7.9

aio 11 2 i 3.0

aio 12 3 i 1.7

aio 13 4 i 1.4

aio 14 5 i 1.2

aio 15 6 i 1.1

604355

227818

130962

107353

95549

87313

487107 38564

166585 30099

98601 26027

81510 24218

72486 22105

0

0

0

0

0

0 4680405 1.0

513321 1.2 0

180650 1.3 0

88725 1.5

69842 1.5

0

0

61630 1.6

65094 21449 0 57238 1.5

0 23338

1036

222

7

5

0 5

0 3

Number of AIOVPs = Number of physical disks containing cooked chunks + 2

1. If Kernel Asynchronous IO is disabled

Number of AIOVPs = Number of physical disks containing chunks + 2

1. Modify the VPCLASS ONCONFIG parameter for AIOVPs and bounce Informix VPCLASS aio,num=5,noage

# Monitor and Tune AIOVPs

End Result:

* + AIO Virtual Processors dynamically tuned to meet your I/O demand
  + Determine if AIOVPs can handle I/O load

1. Enable automatic AIOVP tuning which will dynamically add AIOVPs if they are needed

informix> vi $INFORMIXDIR/etc/$ONCONFIG

AUTO\_AIOVPS 1

1. Use onstat -g iov to determine if I/O requests are backing up

onstat -g iov will show us the number of I/Os performed per AIOVP wakeup. We want to see one or more of the most active AIOVPs with an io/wup of 1.0 or less which tells us on the average number of I/O requests are in the queue each time an AIOVP was awakened to do some I/O work. If there is consistently only 1 I/O request in the queue then our AIOVPs are handling the I/O load.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| aio 16 7 i | 1.1 | 82456 | 61836 | 19979 | 0 | 54071 | 1.5 | | 0 | 3 |
| aio 17 8 i | 1.0 | 77195 | 57215 | 19367 | 0 | 49859 | 1.5 | | 0 | 4 |
| aio 18 9 i | 0.9 | 70738 | 51898 | 18305 | 0 | 45489 | 1.6 | | 0 | 5 |
| pio 6 0 i 0.1 4326 | | | 0 4326 0 | | 4326 1.0 | | 0 | 37156 | | |
| lio 5 0 i 31.3 2405192 | | | 0 2405192 | | 0 2396927 | | 1.0 | 0 19652116 | | |