IBM Informix IDS

RTO_SERVER_RESTART and nonblocking checkpoints

Tuning the IDS server to take advantage of nonblocking checkpoints and improving fast recovery performance

ON DEMAND BUSINESS™

Agenda

- **What are checkpoints and why do we do them?**
- **Tuning checkpoint performance for 7.x, 9.x and 10.x**
- **Tuning checkpoint performance for 11.x**
- **Maintaining a recovery time objective**
- **New onstat options**

What are checkpoints and why do we do them?

- **Create a consistency point to start fast recovery from in the event of an unexpected failure**
- **Create a consistency point to perform some function… like taking a backup of the database**
- **A checkpoint is a point in time where cached data (bufferpool) is flushed to disk**

When do checkpoints get triggered?

- **Administration events**
  - Database backup, adding a DBSpace, users (onmode –c)
- **Physical Log 75% full**
- **1 Checkpoint in the logical log**
- **Long transactions**
- **Maintain Recovery Time Objective (RTO) policy using CKPTINTVL**
- **HDR Secondary requires checkpoint**

Tuning checkpoint performance for 7.x servers

- **How to reduce transaction blocking…**
  - Aggressive LRU flushing
    - More LRUs
    - More Cleaners
    - Low LRU min and max settings (< 1%)
  - onmode –B just prior to checkpoint
  - Improve I/O subsystem

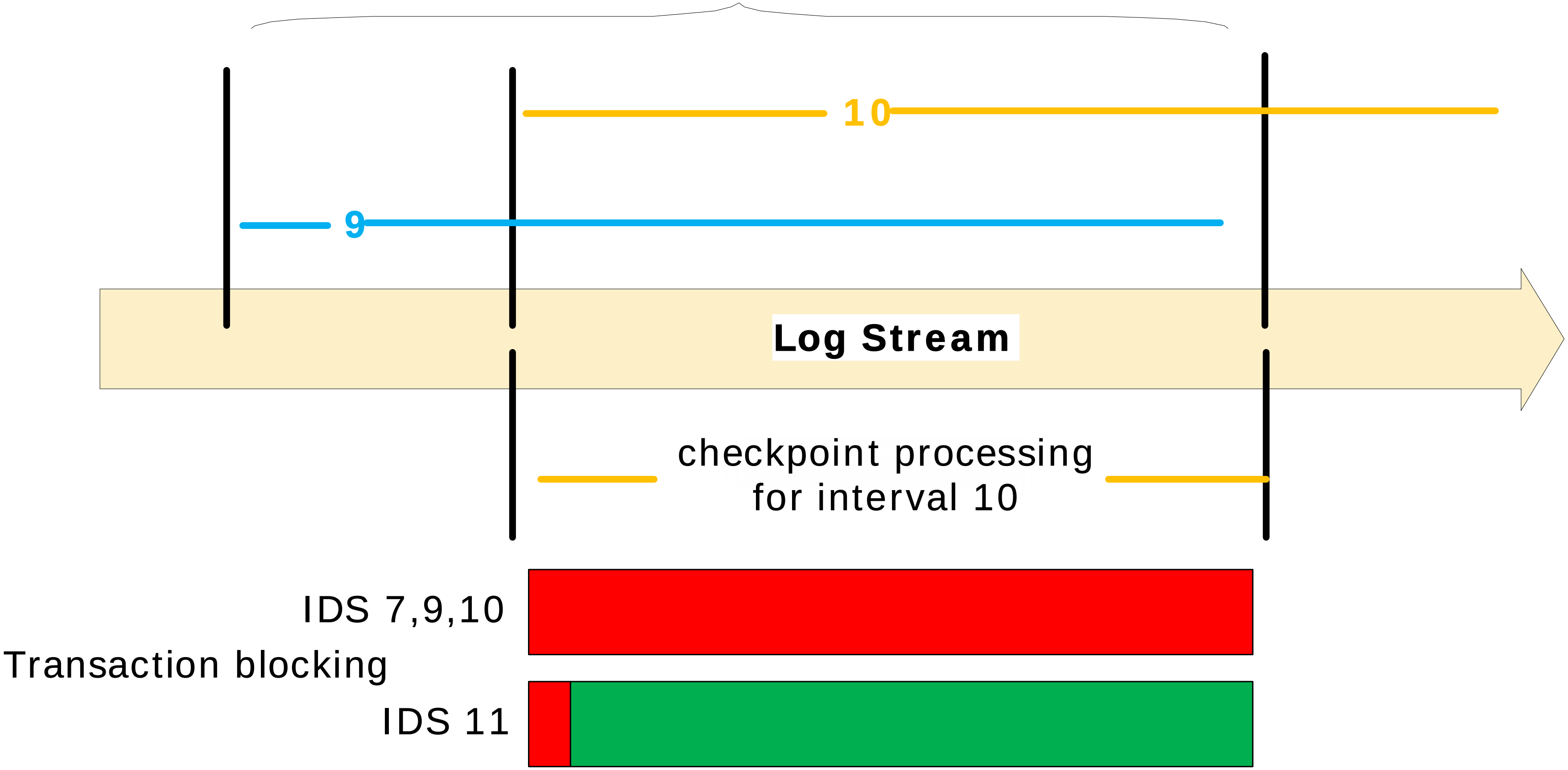
Tuning checkpoint performance for 9.x & 10.x servers

- **Fuzzy checkpoint alleviates some of the problem but…**
  - Unpredictable checkpoint performance
  - Unpredictable fast recovery times
  - Same techniques as 7.x

Nonblocking Checkpoints

- **No transaction blocking during the flush of the bufferpools**
- **Fuzzy checkpoints removed**
- **Nonblocking checkpoints are triggered by …**
  - Physical log 75% full
  - Logical log full
  - CKPTINTVL
  - Initial boot checkpoint
  - A few other miscellaneous cases
- **All others are transaction blocking, like…**
  - Adding a DBSpace
  - Doing a database backup

If IDS would experience and unexpected outage, fast recovery would restart at checkpoint interval 9 until checkpoint processing of interval 10 completed.



When will nonblocking checkpoints block?

During checkpoint processing (disk flush), transactions will continue to consume physical and logical log resources

- **Transactions will block to…**
  - Avoid physical log overflow
  - Avoid logical log overlap
- **To avoid transaction blocking…**
  - Turn on automatic checkpoints (AUTO_CKPTS)
  - Increase the resource (physical or logical log) to allow more time to flush the bufferpool
  - Make LRU flushing more aggressive
  - Increase I/O performance
    - More AIO VPs and cleaners
    - Improve performance of I/O subsystem

Tuning checkpoint performance for 11.x servers

- **Upgrades should just start just using Cheetah**
- **Relax LRU flushing**
  - Can dramatically improve performance
    - TPCC testing saw over 1000% performance improvement in 100% cached scenarios
    - Feeling brave… try lru_min=70, lru_max=80
    - Conservative… try lru_min=30, lru_max=40
- **Don’t be scared of long checkpoints!**
  - Its not how long the checkpoint takes, its how long transactions are blocked
- **Use onstat –g ckp and performance advisories**

New ONCONFIG parameters

- **AUTO_CKPTS**
  Trigger checkpoints sooner to avoid transaction blocking
- **AUTO_LRU_TUNING**
  Make LRU flushing more aggressive
  - Hot page is replaced, 1% more aggressive
  - Foreground write, 5% more aggressive
  - Time to flush bufferpool > RTO_SERVER_RESTART, 10% more aggressive
- **AUTO_AIOVPS**
  Monitor AIO VPs and add more when I/O requests suggest more AIO VPs would be beneficial

New ONCONFIG parameters

- **RTO_SERVER_RESTART**
  allows users to specify a target amount of time the server is allowed for fast recovery
  - RTO_SERVER_RESTART=0
    - Use CKPTINTVL to trigger checkpoints
  - 60 to 1800 seconds (1 – 15 minutes)
  - Server will fine tune with each fast recovery to improve predictability

How does RTO_SERVER_RESTART work?

- **Estimate/Calculate the speed of fast recovery**
  - Server boot time
  - Physical log recovery (RAS_PLOG_SPEED)
  - Logical log recovery (RAS_LLOG_SPEED)
  - Assume all updates fit into bufferpools
- **Monitor physical and logical log usage to trigger a checkpoint when the estimate of recovery would exceed policy**

Tuning for RTO_SERVER_RESTART

- **More physical log**
  - RTO_SERVER_RESTART uses more physical log resources
- **Everything fits into memory**
  - Bufferpool should be big enough to handle all pages updated during fast recovery
  - Physical log seeds bufferpools with all the pages that will get updated during fast recovery
  - Avoid I/O to improve predictability
    - Doing I/O won’t make fast recovery fail, just unpredictable/slower

ONCONFIG file defaults changes

- **ONCONFIG changes**
  - Default PHYSBUFF
    - 128Kb / 512Kb when RTO_SERVER_RESTART enabled
  - Default LOGBUFF
    - 64Kb
  - When server is configured with resources smaller than recommended (default), a performance warning message is sent to the message log

onmode commands

- **AUTO_CKPTS**
  - onmode –wm AUTO_CKPTS=1 … turn automatic checkpoints on
  - onmode –wm AUTO_CKPTS=0 … turn automatic checkpoints off
- **AUTO_AIOVPS**
  - onmode –wm AUTO_AIOVPS=1 … turn automatic aio vp tuning on
  - onmode –wm AUTO_AIOVPS=0 … turn automatic aio vp tuning off
- **AUTO_LRU_TUNING**
  - onmode –wm AUTO_LRU_TUNING=1 … turn automatic lru tuning on for all bufferpools
  - onmode –wm AUTO_LRU_TUNING=1,min=40,max=50 … turn automatic lru tuning on, set lru min and max for all bufferpools
  - onmode –wm AUTO_LRU_TUNING=0 … turn automatic lru tuning off
  - Does not support –wf option!
- **RTO_SERVER_RESTART**
  - onmode –wm RTO_SERVER_RESTART=60 … turn automatic fast recovery tuning on and set fast recovery time to 60 seconds
  - onmode –wm RTO_SERVER_RESTART=0 … turn automatic fast recovery tuning off

Changing physical log

- **Can now change physical log size and/or location on the fly**
  - No server reboot!
  - Changing ONCONFIG file to change physical log no longer supported

Performance Advisory

- **New messages to message log to suggest performance changes**

```
Performance advisory: The physical log is too small to
  accommodate the time it takes to flush the bufferpool.

Results: Transactions may block during checkpoints.

Action: Increase the size of the physical to at least 123000 Kb.
```

Onstat –g ckp

```
Auto Checkpoins=On   RTO_SERVER_RESTART=60 seconds   Estimated recovery time 7 seconds

                                                       Critical Sections                                Physical Log      Logical Log
          Clock                          Total   Flush  Block  #       Ckpt   Wait   Long   # Dirty  Dskflu  Total      Avg    Total      Avg
Interval  Time      Trigger   LSN         Time    Time   Time   Waits   Time   Time   Time   Buffers  /Sec    pages      /Sec   Pages      /Sec
1         18:41:36  Startup   1:f8        0.0     0.0    0.0    0       0.0    0.0    0.0    4        4       3          0      1          0
2         18:41:49  Admin     1:11c12cc   0.3     0.2    0.0    1       0,0    0.0    0.0    2884     2884    1966       162    4549       379
3         18:42:21  Llog      8:188       2.3     2.0    2.0    1       0.0    2.0    2.0    14438    7388    318        10     65442      2181
4         18:42:44 *User      10:19c018   0.0     0.0    0.0    1       0.0    0.0    0.0    39       39      536        21     20412      816
5         18:46:21  RTO       12:188      54.8    54.2   0.0    30      0.6    0.4    0.6    68232    1259    210757 1033       150118     735

Max Plog    Max Llog    Max Dskflush    Avg Dskflush    Avg Dirty    Blocked
pages/sec   pages/sec   Time            pages/sec       pages/sec    Time
8796        6581        54              43975           2314         0
```

SYSMASTER tables

- **syscheckpoint**
  - Keeps history on checkpoints
- **sysckptinfo**
  - Keeps info on automatic checkpoints

Monitoring I/O activity

- **onstat –g iof**

```
AIO lobal files:
gfd   pathname                          bytes read   page reads   bytes write   page writes   io/s
3      /dev/sdb5                        317440       155          18432          9             570.8
       op type          count           avg.time
       seeks            0               N/A
       reads             0               N/A
       writes           0               N/A
       kaio reads       27              0.0023
       kaio writes       9               0.0003

4      /work/chunk                      4147200        2025          177547264   86693        617.4
       op type          count           avg. time
       seeks            0               N/A
       reads             2025            0.0001
       writes           1369             0.0040
       kaio reads       0                N/A
       kaio writes       0               N/A
```

Additional Information

http://www.ibm.com/developerworks/db2/library/techarticle/dm-0703lashley/index.html