FILESYSTEM DIAGNOSTIC FOR STORNEXT DLC

Alain Renaud
Sustaining engineering

February 2013

Filesystem Diagnostic for StorNext DLC.

- Filesystem Diagnostic tools.
  - iozone
  - cvdb –x
  - cvdb –bv
- References.

Filesystem Diagnostic tools.

- Once we confirmed that we do not have any network problem we can then start to test the filesystem performance.
- It is very useful to know the type of IO the customer is expecting. This will allow us to tune the IO test to represent as close as possible the customer performance.
- Like for the networking test we need to make sure we test the read and the write path.
- There are multiple tools to test IO performance: lmdd, iozone, SIO.
- We are going to see some example using iozone.

## iozone

- iozone is a very powerful tool to analyze all kind of filesystem trafic. Here are some example.
  - # iozone -r 1m -s 5g -i 0 -i 1 -e -c -+n -t 1
    - Create a 5gig(-s5g) file using read and write of 1 meg(-r1m).
    - Will do a write test (-i0) to create the file then a read test (-i1).
    - Include flush/fsync/close is perf calculation (-e –c).
    - No re-read and re-write (-+n).
    - Have one thread (-t 1).
  - Depending on the customer application you can change the following flag.
    - If you know the size of the read and write changing the –r to match the application is a very good idea.
    - The record size (-r) will decide if we use the buffer cache or not. record size equal or smaller then 1Meg will use buffer cache by default.
    - Removing the '-+n' can be useful to catch buffer cache issues. re-read should be faster if it goes via the cache.

iozone (cont)

```
# iozone -r 1m -s 5g -i 0 -i 1 -e -c -+n -t 1
 Iozone: Performance Test of File I/O
                Version $Revision: 3.279 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                     Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                     Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                     Randy Dunlap, Mark Montague, Dan Million,
                     Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy,
                     Erik Habbinga, Kris Strecker, Walter Wong.

        Run began: Thu Feb  7 13:55:16 2013
        Record Size 1024 KB
        File size set to 5242880 KB
        Include fsync in write timing
        Include close in write timing
        No retest option selected
        Command line used: /usr/bin/iozone -r 1m -s 5g -i 0 -i 1 -e -c -+n -t 1
        Output is in Kbytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 Kbytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
        Throughput test with 1 process
        Each process writes a 5242880 Kbyte file in 1024 Kbyte records
```

iozone (cont)

```
        Children see throughput for  1 initial writers  =  200134.89 KB/sec
        Parent sees throughput for  1 initial writers   =  200131.54 KB/sec
        Min throughput per process                      =  200134.89 KB/sec
        Max throughput per process                      =  200134.89 KB/sec
        Avg throughput per process                      =  200134.89 KB/sec
        Min xfer                                        = 5242880.00 KB

        Children see throughput for  1 readers          =  248509.91 KB/sec
        Parent sees throughput for  1 readers           =  248503.45 KB/sec
        Min throughput per process                      =  248509.91 KB/sec
        Max throughput per process                      =  248509.91 KB/sec
        Avg throughput per process                      =  248509.91 KB/sec
        Min xfer                                        = 5242880.00 KB
```

cvdb –x

- When doing filsystem IO testing you can use the command the ‘cvdb –x’ on the client and/or the gateway. The report will give glance at the speed and balancing.
- The view from the client show the current IO speed to each proxy server and total amount of data transfer to each server.

cvdb –x (cont)

```
# cvdb -x
 Proxy # Listen/Backlog Sockets: 0
 Proxy # Active Sockets:         2
 Proxy # Server Address Entries: 2
 Proxy # Active Servers:         2

 Proxy: Connection # 1 (Client) for <vsop02a>
   Server System ID  10.65.179.235
   Server    IP Addr  192.168.16.235 (57881)
   Client    IP Addr  192.168.16.247 (46591)
     Read  Bytes/Sec 0
     Write Bytes/Sec 83388006 (79MB + 537KB + 614 bytes)
  FS Read  Bytes/Sec 0
  FS Write Bytes/Sec 170295296 (162MB + 416KB)
     Queued I/O      256.0 K
     Reads 540673 (528KB + 1 bytes)
     Writes 583169 (569KB + 513 bytes)
     Read  Bytes 35433545728 (33GB + 64KB)
     Write Bytes 38218563584 (35GB + 608MB + 64KB)

 Proxy: Connection # 2 (Client) for <vsop02a>
   Server System ID  10.65.178.241
   Server    IP Addr  192.168.16.241 (57786)
   Client    IP Addr  192.168.16.247 (51819)
     Read  Bytes/Sec 0
     Write Bytes/Sec 86907288 (82MB + 902KB + 408 bytes)
  FS Read  Bytes/Sec 0
  FS Write Bytes/Sec 170295296 (162MB + 416KB)
     Queued I/O      63.5 M
     Reads 0
     Writes 14877 (14KB + 541 bytes)
     Read  Bytes 0
     Write Bytes 974979072 (929MB + 832KB)
```

cvdb -bv

- To get buffer cache statistic you can always use the command ‘cvdb –bv’ This command will report on the buffer cache usage on the DLC client.
- A high cache hit indicate a good usage of the buffer cache.
- If your IO record are bigger then 1Meg you will bypass the cache by default unless you change the parameter on the client mount.

```
# cvdb -bv
 Buffer Cache Stats for 1 buffer cache:

 65536 cachebufsize used by 1 mount. 1024 bufs consumes 64 MBs.
 cache_hits 842209 (30.71%) cache_misses 1900574 (69.29%)
 No buffer write throttling detected 0 dirty_checks.
 cap 1024 buffercachewant 0 bufhashsize 509
 bcdirtycnt 0 dirty_ndone 0 flusheractive 0
 deferredflush 0 dirtywaiters 0
 rsvd max 73793536 non-zero rsvd min 6684672
 successful rsvd requests 1937 failed rsvd requests 0
 Mount Info Cache Stats for 1 mount max is 24:

 "/stornext/vsop02a": 65536 cachebufsize 32 MB rsvd_req
 "/stornext/vsop02a" 0 buffer dirty cnt 0 files dirty 435 max files dirty.
```

References

- http://www.iozone.org/
- http://sourceforge.net/projects/lmbench/

BE CERTAIN