SHARE Live!: z/OS 2.2 Unix Systems Services Latest Status and New Features

Wednesday, August 12, 2015:

11:15 AM - 12:15 PM,

Dolphin, Oceanic 4

Vivian W Morabito

AGENDA

- AMODE 64: LFS (Logical File System) & zFS

- Improved Displays and Query commands

- New zFS fsinfo command

- Health Check Updates

- Miscellaneous Enhancements

LFS updated in V2R2 to run AMODE 64

– Support for large workloads… some customers are near the limit.

– z/OS V2R1 current max threads: ~32K (in kernel at any one time)

- Primarily due to 36K bytes of kernel private storage that is required for each thread's stack

– z/OS V2R2 increased max threads: ~ 500K

- Theoretical limit, will greatly depend on what type of services are being issued, and the storage requirements of those services.

– Supports both AMODE 31 and AMODE 64 PFS-es

Kernel / LFS stacks above the bar

- Kernel storage can be (optionally) moved above the bar to alleviate potential storage constraints

- New BPXPRM parmlib option

  - KERNELSTACKS(BELOW | ABOVE)

  - BELOW is default

  - Only installations that require larger thread capacity need set the ABOVE option

  - Effective on the next IPL

  - Affects all UNIX System Services kernel processes and threads

Kernel / LFS stacks above the bar…

- KERNELSTACKS(ABOVE)

  - Each thread obtains 512K byte stack above the bar which persists until the thread is terminated (undubbed)

  - 2K work area of below the bar available to each thread

    - used to call MVS services which do not support above the bar storage

  - D OMVS,O   displays KERNELSTACKS setting

zFS V2R2: AMODE 64

- zFS converted to 64 bit addressing in V2R2, which provides improved performance and scalability

  - Eliminated possible constraint on vnodes

  - Allows for large caches, large trace table size

zFS V2R2 optionally runs in OMVS address space

- Running zFS in its own ("colony") address space remains the default
  - *No installation or configuration changes required.*

- Running in OMVS AS provides performance improvement for file and directory operations since OMVS can call zFS more efficiently
  - *Performance benefit up to 10%, depending on workload.*

- CANCEL ZFS command is not available when running in OMVS address space

zFS V2R2: AMODE 64 …

- To run **in the OMVS address space** omit ASNAME parm from the FILESYSTYPE statement in BPXPRMxx

   FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM) PARM('PRM=(xx,xx,…)')

  - *The IOEPRMxx parmlib method must be used if zFS is running in the OMVS address space*

- To run **in a colony AS** (same as in V2R1 and prior):

   FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM) ASNAME(ZFS) PARM('PRM=(xx,xx,…)')

zFS V2R2 – cache changes

- In z/OS V2R2, zFS caches were moved above the 2 G addressing bar. This move allows for the use of very large zFS caches which benefits performance.

- In V2R2 cache sizes ranges are as follows:
  - *vnode_cache_size*   1000 to 10000000
  - *meta_cache_size*   1 M to 64 G
  - *token_cache_size*   20480 to 20000000
  - *trace_table_size*   1 M to 65535 M (partial ranges can be printed)
  - *xcf_trace_table_size*   1 M to 65535 M

- No separate metaback_cache in V2R2.
  - *If metaback_cache_size specified it will be combined with meta_cache_size*

DISPLAY OMVS updates

- D OMVS,STORAGE with KERNELSTACKS(ABOVE) shows THREADS instead of STACK CELLS

- If no processes have more than 50 threads you will get

     NO PROCESSES FOUND USING 50 OR MORE THREADS

   …rather than list of threads

- Display unchanged for KERNELSTACKS(BELOW)

D OMVS,STORAGE

- New option:

  PROCESSES=xx|P=xx

- xx is the number of processes to be shown in the display.
- If not specified, default is 10

- For example: D OMVS,ST,P=14

MODIFY OMVS command

- Changed to support PFS-es running in the OMVS AS

- **MODIFY OMVS,PFS=ZFS,<cmd>**

  - Required if zFS is running in the OMVS address space

  - Will also work if zFS is running in a colony address space

  - For example: MODIFY OMVS,PFS=ZFS,QUERY,LEVEL

- **MODIFY ZFS,<cmd>**

  - This is the existing, unchanged MODIFY command

  - Can still be used when running in a colony address space

zFS V2R2 – new fsinfo utility

- **zfsadm fsinfo** command
  - displays detailed information for one or more file systems.
  - File systems can be specified with a specific name, or in a group by using a common prefix or common suffix.
  - They can also be selected by specifying common attributes.

- This information is also available via:
  - new **File System Information API**
  - new **FSINFO modify command**.

zFS V2R2 – new fsinfo utility…

- **zfsadm fsinfo [-aggregate *name* | -path *path_name* | -all]**
  [{-basic |-owner | -full |-reset}]
  [-select *criteria* | -exceptions]
  [-sort *sort_name*][-level][-help]

- API

    - FSINFO_REQUEST (0x40000013)
    - opcode 153 (query), opcode (154) reset

  modify *procname*,fsinfo[,{*aggrname* | all}
  [,{full | basic | owner | reset}
  [,{select=criteria | exceptions}] [,sort=sort_name]]]

zFS V2R2 – new fsinfo utility…

- Provides a quick, easy method to display filesystem info. For example:

- zfsadm fsinfo –all
  - Displays basic information about all mounted / attached filesystems

- zfsadm fsinfo –select Q
  - Displays basic information about quiesced filesystems

- zfsadm fsinfo –select RS –full
  - Displays full info** about filesystems mounted RWSHARE

    **information that is maintained by the system owning each specified file system AND information that is locally maintained by each system in the sysplex that has each specified file system locally mounted.

fsinfo utility can select filesystems by name, mount point, or any selection criteria(s)

Criteria	Apply to aggregates that…
CE	Had XCF communication failures
DA	Are marked damaged by the zFS Salvager
DI	Are disabled for reading and writing
GD	Are disabled for dynamic grow
GF	Had failed dynamic grow attempts
IE	Had disk IO errors
L	Have less than 1MB of free space
NS	Are mounted NORWSHARE
QV	Contain extended (v5) directories that are using overflow pages
Q	Are currently quiesced
RQ	Had application activity
RO	Are mounted read-only
RW	Are mounted read/write
RS	Are mounted RWSHARE
SE	Have returned ENOSPC errors to applications
TH	Have sysplex thrashing objects in them.
V4	Shows aggregates that are version 1.4
V5	Shows aggregates that are version 1.5
V5D	Disabled for conversion to V1.5
WR	Had application write activity

zfsadm fsinfo -exception

- Quick, easy way to find out if there are filesystems with "exceptions"
    - Quiesced
    - Disabled
    - Had grow failures
    - Low on space
    - Failed to convert a directory to v5
    - Damaged
    - Experienced XCF communication failure
    - Ran out of space

7/29/2015

18

zFS V2R2 – fsinfo sample output

```
File System Name: PLEX.ZFS.SMALL2

    *** Owner information ***
    Owner:                DCEIMGHQ           Converttov5:          OFF,n/a
    Size:                 40320K             Free 8K Blocks:       905
    Free 1K Fragments:    7                  Log File Size:        32800K
    Bitmap Size:          8K                 Anode Table Size:     80K
    File System Objects: 28                  Version:              1.5
    Overflow Pages:       0                  Overflow HighWater:   0
    Thrashing Objects:    0                  Thrashing Resolution: 0
    Token Revocations:    37                 Revocation Wait Time: 85.141
    Devno:                46                 Space Monitoring:     90,5
    Quiescing System:     n/a                Quiescing Job Name:   n/a
    Quiescor ASID:        n/a                File System Grow:     OFF,0
    Status:               RW,RS,GD,SE
    Audit Fid:            C3C6C3F0 F0F001F9 0000

    File System Creation Time: Nov 14 00:07:36 2014
    Time of Ownership:         Dec  9 11:10:36 2014
    Statistics Reset Time:     Dec  9 11:10:35 2014
    Quiesce Time:              n/a
    Last Grow Time:            n/a

    Connected Clients:    n/a


Legend: RW=Read-write, GD=AGGRGROW disabled, RS=Mounted RWSHARE
        SE=Space errors reported
```

zFS V2R2 – fsinfo full output (cont'd)

```
*** local data from system DCEIMGHQ (owner: DCEIMGHQ) ***
   Vnodes:                  410          LFS Held Vnodes:             30
   Open Objects:            2            Tokens:                      29
   User Cache 4K Pages:     13           Metadata Cache 8K Pages:     35
   Application Reads:       104355143    Avg. Read Resp. Time:        0.151
   Application Writes:      39105328     Avg. Writes Resp. Time:      0.641
   Read XCF Calls:          0            Avg. Rd XCF Resp. Time:      0.000
   Write XCF Calls:         0            Avg. Wr XCF Resp. Time:      0.000
   ENOSPC Errors:           15920695     Disk IO Errors:              0
   XCF Comm. Failures:      0            Canceled Operations:         0

   DDNAME:                  SYS00008
   Mount Time:              Dec  9 11:10:35 2014


VOLSER PAV    Reads       KBytes     Writes     KBytes       Waits
Average
------ ---  ---------  ----------  --------  ----------  ---------- ----
-----
CFC000  1    821059     6997808     537165    14633376     974568
2.805
------ ---  ---------  ----------  --------  ----------  ---------- ----
-----
TOTALS       821059     6997808     537165    14633376     974568
2.805
```

zFS V2R2 Performance Query Improvements

- zFS performance counters were changed from 4 bytes to 8 bytes

  - Allows for monitoring of zFS performance over longer periods of time before the counters wrap.

  - Improved counters available via:

    - zFS Statistics APIs (version 2)

    - zFS modify queries

    - zfsadm query command reports.

  - Version 1 zFS Statistic APIs will continue to be supported, returning the low words of the counters.

zFS V2R2 Performance Query Improvements…

- New API queries and zfsadm queries added for:

- stkm (server token management statistics),
  - STATOP_STKM (252)
  - zfsadm query -stkm

- ctkc (sysplex client operations),
  - STATOP_CTKC (253)
  - zfsadm query -ctkc

- svi (sysplex owner operations statistics)
  - STATOP_SVI (254)
  - zfsadm query -svi

zFS V2R2 Storage Report Improvements

- Storage reports now contain information about storage usage above the 2G bar.

  - STATOP_STORAGE_ABOVE (255)

  - modify zfs,query,storage

  - zfsadm query -storage

Health Checks

- USS_KERNEL_STACKS_THRESHOLD

    – existing health check

    – will be disabled when KERNELSTACKS(ABOVE) is specified in parmlib, and generates message BPXH075I

```
BPXH075I KERNELSTACKS(ABOVE) is specified in
parmlib members during IPL
USS_KERNEL_STACKS_THRESHOLD
is not valid in the current environment
```

Health Checks…

- **USS_KERNEL_PVTSTG_THRESHOLD**

- Existing check (V2R1)

- Monitors the current usage of z/OS UNIX System Services private storage

  - KERNELSTACKS…

    - BELOW: private storage is a portion (~20%) of kernel region size

    - ABOVE: private storage is total number of bytes of below the bar private storage that can be allocated

- Change in V2R2: Override PARMs modified to allow varying severity levels in the warning

  - High, Medium, Low, and / or None

Health Checks….

- **USS_KERNEL_PVTSTG_THRESHOLD** (cont'd)

- Was: PARM('PVTSTG(85%)'),
- Now: PARM('PVTSTG_HIGH(90%), PVTSTG_MED(85%), PVTSTG_LOW(80%), PVTSTG_NONE(xx%)'),

- Thresholds indicate where the exception messages go…. For example:
  - PVTSTG_HIGH(xx%):   system console, highlighted in red + syslog + sdsf HC queue
  - PVTSTG_NONE(xx%):   only syslog + sdsf HC queue

- One or more threshold level may be specified. If a threshold for a given severity is not specified, no exception message will be issued for that level.

- If PVTSTG is specified along with any PVTSTG_xxx value the PVTSTG value is ignored

Health Checks…

- **USS_KERNEL_RESOURCES_THRESHOLD**

- New in V2R2, only valid when running KERNELSTACKS(ABOVE)

- Monitors the current usage of z/OS UNIX System Services kernel resources

- Multiple, override-able threshold levels may be specified

  - KRES_HIGH(x%), KRES_MED(x%), KRES_LOW(x%), KRES_NONE(x%)

- Default is: KRES_HIGH(90%), KRES_MED(85%), KRES_LOW(80%)

Health Checks

- ZOSMIGV1R13_ZFS_FILESYS and ZOSMIGREC_ZFS_RM_MULTIFS were removed.

- Existing ZFS_VERIFY_CACHESIZE was updated to reflect V2R2 cache changes

- New healthcheck ZFS_CACHE_REMOVAL
  - tran_cache_size & client_cache_size are obsolete (ignored)
  - metaback_cache_size, obsolete …if specified, is combined with meta_cache_size.
    - *IBM recommends update to update your configuration to use only meta_cache_size.*

Improved zFS logging

- zFS is using a better performing method for handling the writing of records to the zFS aggregate log.

- The new logging method returns different statistics in:

    - zfsadm query -logcache

    - MODIFY ZFS,QUERY,LOG performance report.

    - Statistics Log Cache Information API

Miscellaneous zFS V2R2 updates

- New IOEFSPRM configuration variables

  - **modify_cmd_threads**: specifies the current number of threads that are defined to handle zFS modify commands.

  - **quiesceinfo_message_delay:** specifies the minimum number of seconds to delay issuing the IOEZ00830E

- Toleration APAR OA46026 must be installed and active on z/OS V1R13 and z/OS V2R1 to bring a z/OS V2R2 member into the sysplex

ptrace register vector support

- Enables debuggers (such as dbx) to read and write vector (SIMD) registers

- Previous zSeries CPUs did not have vector registers, therefore the ptrace command did not have vector support.

- The new z13 Processor includes vector (Single Instruction, Multiple Data or SIMD) support which allows increased parallelism with instructions that process several operands in a single instruction

w_getpsent() security option

- w_getpsent() gets information about the status of any process that the calling process has access to.

- V2R2 update will allow w_getpsent() to use thread level security.

- This fixes a problem where the z/OS CIM server was returning process level rather than thread level information

- New env var _BPXK_GPSENT_SECURITY with 2 settings:

  - THREAD: uses thread level ACEE if one is present

  - PROCESS: (default) – uses process level security

SMT Support

- Support for new z13 SMT (Simultaneous Multi-threading support)

- A new field was added to the BPXZODMV mapping macro used by the BPXEKDA macro, used by RMF

- Will allow RMF to correctly report z13 general purpose and zIIP CPU times