IBM Education Assistance (IEA) for z/OS V2R3

Line Item: zFS Encrypt/Compress   zFS Shrink
zFS SMF support   zFS Online Salvage
zFS Dynamic Change Aggregate Attributes
zFS Implicit Format

Element/Component: Distributed File Service zFS

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None.

Session Objectives

- Encrypt/Compress Aggregate
- Shrink Aggregate
- SMF Support
- Dynamic Change Aggregate Attributes
- Online Salvage
- Implicit Format
- Other Misc. updates

Overview – Encrypt/Compress

- Problem Statement / Need Addressed
  - All modern POSIX file systems have facilities to allow encryption of file system data for security and most file systems allow data compression to save disk space. zFS needs to be functionally competitive with the other file systems.
- Solution
  - With the underlying Media Manager support, allow encryption of zFS file systems.
  - Allow compression of zFS file system data.
  - Provide Encrypt/Compress monitoring tools.
- Benefit / Value
  - Data security and disk space savings
  - Performance improvement(with the new EDCFIXED option)

Encryption Invocation

- New function utilize DFSMS Encryption for VSAM datasets
  - Allow zFS to encrypt file data, ACLs and symbolic link contents. V5 directories are not encrypted (only fragmented v4 directories can be encrypted)
  - zFS uses a page fixed auxiliary buffer pool for use by DFSMS for encryption, **new** IOEFSPRM option **EDC_BUFFER_POOL**
    - Default is 32M. Value range 1M - 1G
  - 2 ways to encrypt a zFS file system
    - During DEFINE time for a new file system
    - Convert an existing file system to be encrypted. Any relevant objects after conversion will be encrypted.
  - File system must be in SMS class that is **eligible** for encryption
    - **Key label** – can be specified when defining a new file system or converting an existing file system to be encrypted.
    - If using IDCAMS command DEFINE CLUSTER to define an aggregate you want to encrypt, recommend to use **ZFS** keyword instead of LINEAR.
  - encryption is only allowed when all systems are at z/OS V2R3

Usage & Invocation

- Introduced new **long-running** admin commands:
    - zfsadm encrypt/decrypt
    - zfsadm compress/decompress
    - zfsadm shrink
    - zfsadm salvage (online salvage)
- Requires available foreground thread
  - New IOEFSPRM option **long_cmd_threads**
    - **long_cmd_threads**=*foreground, background* (value can be 1-64, default is 8,24)
- Use FSINFO command for progress monitoring
- Can be canceled

Compression Invocation

- New function utilizes zEDC Authorized Services (introduced in z/OS 2.1) to
  - Allow zFS to compress file data (only files are compressed, not directories)
  - zFS uses a page fixed auxiliary buffer pool for use by the zEDC services for compression, again the **new** IOEFSPRM option **EDC_BUFFER_POOL**
  - Any file greater than 8K can be compressed and if compression saves at least one disk block, zFS will store that data in a compressed format by recording compression information in the metadata for a file.
  - zFS allows the user to permanently fix the user file cache to eliminate data movement to/from auxiliary buffer pool, provides significant performance benefit, new external → **EDCFIXED** option of IOEFSPRM **user_cache_size** parameter. e.g. user_cache_size=64M,**edcfixed**
- New health check **ZFS_VERIFY_COMPRESSION_HEALTH**
  - Determines whether the zFS user cache pages are all zEDC fixed when there are compressed aggregates. The EDCFIXED attribute of the USER_CACHE_SIZE option can be used to request that the user cache be zEDC fixed. Note that due to zEDC contraints, it may not be possible to zEDC fix all user cache pages.
- Compression is only allowed when all systems are at z/OS V2R3

Usage & Invocation

- **zfsadm encrypt/decrypt** – new command to encrypt existing file system data or decrypt it, while the file system is mounted R/W. Syntax:

```
zfsadm encrypt -aggregate aggr_name [-cancel | -keylabel  key]
zfsadm decrypt -aggregate aggr_name [-cancel]
```

- **zfsadm compress/decompress** – new command to compress existing file system data or de-compress it, while the file system is mounted R/W. Syntax:

```
zfsadm     compress -aggregate aggr_name  [-cancel]
zfsadm decompress -aggregate aggr_name  [-cancel]
```

-cancel : Cancel an in-progress encryption/compression for the specified aggregate.

- **keylabel** *key* : Define a valid key label to associate with the specified aggregate.

- **New encryption API** via ZFSCALL_AGGR(0x40000005) with opcode:
  - 262 for encrypt and 263 for decrypt
- **New compression API** via ZFSCALL_AGGR(0x40000005) with opcode:
  - 264 for compress and 265 for decompress

Usage & Invocation

- **zfsadm define**– new option **-keylabel *key*** to define a key label used for encryption.
- **Format cmds**– new option that allows the user to specify whether new file system should be encrypted or compressed.
  - **zfsadm format**: **-encrypt/-noencrypt/-compress/-nocopmpress**
  - **IOEAGFMT :** **-encrypt/-noencrypt/-compress/-nocopmpress**
  - **IOEFSUTL format: -encrypt/-noencrypt/-compress/-nocopmpress**
  - **Format API** - ZFSCALL_AGGR(0x40000005) with opcode 134
    - new fields **af_encrypt** and **af_compress** in AGGR_FORMAT structure
- IOEFSPRM option **FORMAT_ENCRYPTION** & **FORMAT_COMPRESSION**
  - New default format option for new file systems (the default is OFF, meaning new file systems not encrypted, not compressed)
- **zfsadm query –compression** – shows global statistics for compression effectiveness and zEDC performance.
- **New query API – Statistics Compression Information API** – ZFSCALL_STATS(0x40000007) with opcode(256)

Usage & Invocation

- **zfsadm fileinfo** – shows whether the file is encrypted/compressed or not.

```
...
mtime       Nov  2 11:18:35 2015     atime        Nov  2 11:18:35 2015
ctime       Nov  2 11:18:35 2015     create time  Nov  2 11:18:35 2015
reftime     none
encrypted                            compressed 4762K saved
```

- **zfsadm fsinfo** – shows if a file system is encrypted/compressed.

```
PLEX.ZFS.AGGR                           DCEIMGNJ RW,RS,EN,NC

Legend: RW=Read-write, RS=Mounted RWSHARE, EN=Encrypted, NC=Not compressed
```

Compression Usage

- Files broken into 64K segments
- Performance
  - Reading and writing data to/from disk is faster, often significantly, with compression because the zEDC cards are typically faster than the disks.
  - Use **zfsadm query -compression** to show speed of zEDC calls

```
Compression calls:        246428  Avg. call time:              0.177
 -KB input              13190960    KB output               1971456
Decompression calls:      509140  Avg. call time:              0.154
  KB input               4073128    KB output              21406072
```

- Recommendation
  - Almost all file systems will benefit from compression, only exception being file systems that contain large files heavily updated in random mode.
  - **EDCFIXED** option of the user file cache provides significant CPU savings.
- Disk savings
  - Data dependent
  - Can pair with the new Shrink command to free DASD space

Encrypt/compress- File System Conversion Usage

- **Conversion Command for R/W Mounted File Systems**
  - Runs on background zFS tasks in parallel with application access
  - Conversion will check for contention with user application access and yield to them to prevent “hanging applications”.
  - Conversion can be cancelled
  - **zFS remembers progress of conversion on disk, one can resume a cancelled conversion or reverse-course**
- **FSINFO**
  - Will show progress of the conversion

```
 Status:                    RW,RS,Q,EI,NC
 AuditFid:                  0000000 000000 0000
 Encrypt Progress:          running, 35% started at Apr 27 11:17:12 2016 task 783E90
Legend: RW=Read-write, RS=Mounted RWSHARE, EI=Partially encrypted
      NC=Not compressed
```

- **FILEINFO**
  - Will show whether a particular file is compressed or encrypted, or if its in the process of conversion.

```
mtime       Nov  2 11:18:35 2015      atime         Nov  2 11:18:35 2015
ctime       Nov  2 11:18:35 2015      create time   Nov  2 11:18:35 2015
reftime     none
not encrypted                         converting to compressed
```

- **Recommendation**:
  - Either compress and encrypt, or compress before encryption to reduce amount of data that needs converting because:
  - Conversion could be a long process for large file systems, all files have to be processed, and for encryption, all ACLs and all anode table pages.

Usage & Invocation - Misc.

- New **zfsadm** **-trace** ***file_name*** option for all zfsadm commands
  - Specify the file name where zFS trace data would be written into.
  - Can be either UNIX file or MVS data set (sequential or PDS member)

Interactions & Dependencies

- Software Dependencies
  - None.
- Hardware Dependencies
  - Compression requires zEDC hardware
- Exploiters
  - Any users of zFS file systems in V2R3

Migration & Coexistence Considerations

- Toleration APAR **OA51692** must be installed and active on all z/OS V2R1 and z/OS V2R2 systems prior to introducing z/OS V2R3

Installation

- Changes in zFS installation:
  - Install toleration APAR **OA51692** on all prior release systems

Overview – Shrink

- Problem Statement / Need Addressed
  - Need to be able to reduce the size of zFS aggregates by releasing unused space.
- Solution
  - New zfsadm shrink command
    - Allows shrinking of **existing** zFS aggregates
    - Uses VSAM PARTREL service to release unused DASD space
  - Shrink aggregates while they are **mounted**
    - Users and applications can access or update files or directories during most of the shrink operation
  - IOEZ00881I action message displayed during the shrink operation
- Benefit / Value
  - Unused DASD space in zFS aggregates can be released and used for other purposes.

Usage and Invocation

- New zfsadm shrink command syntax:
  - zfsadm **shrink** **-aggregate** *aggr_name* { **-cancel** | **-size** *newsize* **[-noai]** ***}***
    - *Newsize* is desired new size of aggregate, specified in 1K units
      - If user activity requires more space during a shrink, zFS will actively increase *new size* as needed
    - **-noai** will not actively increase *newsize* if needed
    - Use **-cancel** to cancel a shrink operation
- **New Aggregate Shrink API** via ZFSCALL_AGGR(0x40000005) with opcode 266
- **FSINFO** – shows progress of a shrinking file system.

```
PLEX.ZFS.AGGR                                   DCEIMGNJ RW,RS,SH
Shrink Progress         AN,35% started at Apr 27 11:17:12 2017
Task 735620

Legend: RW=Read-write, RS=Mounted RWSHARE, SH=Shrinking
        AN=Anode table scan (shrink step 4 of 6).
```

Shrink Usage

- File system must be mounted R/W
- Shrinking requires a long-running foreground thread
  - Every object in the file system must be scanned to determine if it has a block above the new size
    - Blocks above the new size line are copied below the new size using background threads for parallel block copying
- Shrink will check for contention with user application access and yield to them to prevent “hanging applications”
- Free space above new size is released via VSAM PARTREL service
- File system size will not be changed if shrink operation canceled

Migration & Coexistence Considerations

- Toleration APAR **OA51692** must be installed and active on all z/OS V2R1 and z/OS V2R2 systems prior to introducing z/OS V2R3, or shrinking any zFS aggregates.

Installation

- Toleration APAR **OA51692** must be installed and active on all z/OS V2R1 and z/OS V2R2 systems prior to introducing z/OS V2R3

Overview – SMF Support

- Problem Statement / Need Addressed
  - zFS has never taken advantage of the system management facility(SMF) as other components of z/OS have.
- Solution
  - Monitor Important Events
    - Allow for monitoring of important events such as dynamic grow, the determination that the file system has moved, disablement of an aggregate, etc.
  - Store General Performance Information
    - zFS will also provide support to store its general performance indicators(information that is shown in various F,ZFS,QUERY commands). This will allow the user to look back in time at performance of zFS on the system.
  - Store File System Performance Information
    - zFS will allow the user to store per-file system information performance indicators. This is the information typically found in the -local option of the FSINFO command.
- Benefit / Value
  - Record important events and usage statistics which can be used for accounting purposes or problem determination.

Usage & Invocation

- Records for zFS are of type **92** from subtype **50-59**.
  - 50 – File system event
  - 51 – zFS call rates and response times
  - 52 – zFS user cache statistics
  - 53 – zFS meta cache statistics
  - 54 – zFS locking statistics
  - 55 – zFS IO statistics
  - 56 – zFS token manager statistics
  - 57 – zFS storage usage summary
  - 58 – zFS sysplex transmit statistics
  - 59 – zFS per-file system statistics

  See *MVS System Management Facilities (SMF)* for more information.
- Enable recording of specific subtypes via SMFPRMxx. See *MVS System Management Facilities (SMF)* for more information.
- Enable new IOEFSPRM option **smf_recording**
  - **on | off | on,*intvl*** (where ***intvl*** is number of mins from 1-60)
  - ***Intvl*** is only applicable for subtype 51-59. Subtype 50 is recorded when the event occurs.
- New **zfsadm config** and **zfsadm configquery** option **-smf_recording**

Installation

- Enable SMF and create a SMFPRMxx parm file to specify which record types and subtypes are to be recorded. See *MVS System Management Facilities (SMF)* for information on this.

Overview – zFS Dynamic Change Aggregate Attributes

- Problem Statement / Need Addressed
  - **Aggrfull , aggrgrow**, sysplex sharemode (**rwshare|norwshare**) are attributes which are assigned to an aggregate at mount time. Once assigned to an aggregate, these attributes cannot be changed without unmounting and remounting the aggregate.
- Solution
  - Provide a zfsadm command / pfsctl to change these attributes while the aggregate is still mounted.
- Benefit / Value
  - For **aggrfull** and **aggrgrow**, the attributes can be changed without loss of access to the file system.
  - For sysplex sharemode, an aggregate will actually be unmounted and mounted again in the same rw mode.

Usage & Invocation

- **zfsadm chaggr -aggre** PLEX.TEMP **-aggrgrow** on
- **zfsadm chaggr -aggre** PLEX.TEMP **-aggrgrow** off
- **zfsadm chaggr -aggre** PLEX.TEMP **-aggrfull** off
- **zfsadm chaggr -aggre** PLEX.TEMP **-aggrfull** 90,5
- **zfsadm chaggr -aggre** PLEX.TEMP **-rwshare**
- **zfsadm chaggr -aggre** PLEX.TEMP -norwshare
- Change Aggregate Attributes API - ZFSCALL_AGGR(0x40000005) with opcode 160

Only one attribute can be changed per **zfsadm chaggr** command.

The **zfsadm chaggr** command can only be used when all members are at z/OS V2R3.

zfsadm fsinfo can be used to view the results.

Overview – zFS Online Salvage

- Problem Statement / Need Addressed
  - current salvage utility requires that an aggregate be unmounted when an unmount may not be convenient
- Solution
  - Provide a zfsadm command / pfsctl to salvage an aggregate while it is still mounted
- Benefit / Value
  - Aggregate can be salvaged without loss of access to the file system.
  - Will allow zFS to attempt to fix a disabled aggregate which may be corrupted.

Usage & Invocation

- **zfsadm salvage -aggre** PLEX.TEMP
- **zfsadm salvage -aggre** PLEX.TEMP **-verifyonly**
- **zfsadm salvage -aggre** PLEX.TEMP **-cancel**
- SALVAGE API - ZFSCALL_AGGR(0x40000005) with opcode 155
- **zfsadm fsinfo** – shows progress of a salvaging file system

```
PLEX.TEMP                               DCEIMGNJ RW,RS,SH
Salvage Verify:      LC started at Apr 27 11:17:12 2017 Task 735620

Legend: RW=Read-write, RS=Mounted RWSHARE, SL=Salvaging
        LC=Link count scan (salvage step 6 of 10)
```

Usage & Invocation

Salvage can be a very long running operation. Use the -**cancel** option when you want to end a long running salvage command. The use of the salvage utility on an unmounted aggregate is the the preferred method for performing a salvage.

The **zfsadm salvage** command can only be used when all members are at z/OS V2R3.

Overview – zFS Implicit Format

- Problem Statement / Need Addressed
  - creating a zFS aggregate is a two step process. A VSAM LDS is created and then **zfsadm format** needs to run to ensure critical zFS information is written. A one step process would be more convenient.
- Solution
  - zFS will now format a VSAM LDS at mount using the default option values for the **zfsadm format** options.
- Benefit / Value
  - A convenience is provided to those customers who have no need to specify **zfsadm format** options such as **-size**, **-logsize**, **-owner**, **-group**, **-perms**.

Usage & Invocation

- **zfsadm define** -**aggre** PLEX.TEMP -**cyl** 10 1

  mount -**f** PLEX.TEMP mp1

  The above scenario will only work with a **zfsadm define** command and mount command both performed on z/OS V2R3. The Define Aggregate API - ZFSCALL_AGGR(0x40000005) with opcode 139, can also be used in place of the **zfsadm define** command on z/OS V2R3.

  The VSAM LDS must not contain any data.

  See the new zFS keyword for the IDCAMS command for an alternative to using zfsadm define.

  A new IOEFSPRM option, **FORMAT_PERMS**, will allow an override of the default value for the -**perms** option for all **zfsadm format** commands. When not specified, the default for -**perms** is o755.

Interactions & Dependencies

- Software Dependencies
  - **IDCAMS** support for the new **zFS** keyword
- Hardware Dependencies
  - None.
- Exploiters
  - None.

Overview – Misc. Changes

- Problem Statement / Need Addressed
  - Version 1.5 aggregates are preferred because they support a larger architected size and extended (V5) directories. The existence of z/OS V1R3 in a sysplex prevented many customers from moving to version 1.5 aggregates. With z/OS V2R3, z/OS V1R13 is no longer supported in a sysplex. Customers must curently take explicit action to create or convert aggregates to version 1.5.
- Solution
  - zFS will change the defaults of some IOEFSPRM options to facilitate the creation and conversion to version 1.5 aggregates and to take other action which customers have requested.
- Benefit / Value
  - A customer already taking the defaults will not need to change their IOEFSPRM options to get the preferred results.

Usage & Invocation

These are the IOEFSPRM changes

- **CHANGE_AGGRVERSION_ON_MOUNT** (default changed to On)

  - aggregate converted to version 1.5 at mount

  - directories are not converted

- **FORMAT_AGGRVERION** (default changed to 5)

  - version 1.5 aggregate created unless -version4 is used

- **ROMOUNT_RECOVERY** (default changed to ON)

  - RO mounts needing recovery will not fail

- **HONOR_SYSLIST** (deprecated, ignored)

  - on z/OS V2R3, the USS automove options are always consulted when moving an aggregate or when accepting zFS ownership of an aggregate. It may still be the case that the USS and zFS will not match.

Migration & Coexistence Considerations

- IOEFSPRM option **format_aggrversion**: default from 4 to 5
  - Specify **format_aggrversion**=4 if no version 1.5 aggregate is desired
  - If do not want a particular aggregate to be formatted as a version 1.5, use **zfsadm format** with -**version4** option

IOEFSPRM option **change_aggrversion_on_mount**: default from off to on

- Specify **change_aggrversion_on_mount**=off if no version 1.5 aggregate is desired

IOEFSPRM option **romount_recovery**: default from off to on

- Specify **romount_recovery**=off if prefer RO mount failure after running logfile recovery
- Use **zfsadm config -romount_recovery** off for dynamic update

IOEFSPRM option **honor_syslist**: default from off to ignored

- Specify **honor_syslist**=on on all members at prior z/OS releases if the zFS owner needs to be confined to the same subset as the USS owner.

Overview – FSINFO Changes

- Problem Statement / Need Addressed
  - In z/OS 2.3, aggregates can now be compressed or encrypted. In addition, long running commands like shrink and online salvage have been added. Customers need a way to find compressed or encrypted aggregates or a way to see the progress of a long running command.
  -
- Solution
  - FSINFO command is modified to support the selection of compressed or encrypted aggregates and the selection of aggregates which are in the middle of a shrink or online salvage.
- Benefit / Value
  - Customers will have a way to query aggregates which are compressed or encrypted. Customers will have a way to monitor shrinking or salvaging aggregates

Usage & Invocation

New **selection** criteria for **zfsadm fsinfo** and the **MODIFY FSINFO** commands:

- **CO**
  select aggregates which are compressed or partially compressed
- **EN**
  select aggregates which are encrypted or partially encrypted
- **EP**
  select aggregates which are partially compressed or partially encrypted
- **NC**
  select aggregates which are not compressed or partially decompressed
- **NE**
  select aggregates which are not encrypted or partially decrypted
- **SH**
  select aggregates which are in the middle of a shrink
- **SL**
  select aggregates which are in the middle of a salvage

Session Summary

- Encrypt/Compress Aggregate
- Shrink Aggregate
- SMF Support
- Dynamic Change Aggregate Attributes
- Online Salvage
- Implicit Format

Appendix

- Publication references
  - z/OS Distributed File Service zSeries File System Administration
  - z/OS Distributed File Service Messages and Codes
  - z/OS DFSMSdfp Storage Administration
  - z/OS UNIX System Services Command Reference
  - z/OS MVS System Management Facilities (SMF)