z/OS

Distributed File Service zSeries File System (zFS) APAR OA38697 - Documentation updates to enhance information on enlarging aggregate

Contents

Chapter 1. Overview of support

APAR OA38697 applies to Distributed File Service zSeries File System (zFS). It provides documentation updates to enhance information about changing the size of a compatibility mode aggregate.

APAR OA38697 applies to the following z/OS® releases. In addition, this document supplements *z/OS Distributed File Service zFS Administration* for each respective release.

z/OS release	**Publication level**
z/OS Version 1 Release 12	SC24-5989-11
z/OS Version 1 Release 13	SC24-5989-12

Trademarks

IBM® and the IBM logo are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtm.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Chapter 2. Increasing the size of a compatibility mode aggregate

If your zFS file system runs out of space, you have several options to increase its size:

- You can grow the aggregate. For more information, see “Growing a compatibility mode aggregate”.
- If you cannot grow the aggregate (because, for example, there is no more room on the volume), you can add a volume to the aggregate. For more information, see “Adding a volume to a compatibility mode aggregate.”
- If you cannot grow the aggregate and you cannot add a volume (because, for example, you do not have any more volumes available), you can copy the aggregate into a larger VSAM linear data set. There are two ways to copy the data:
  - You can copy each file and directory of the zFS aggregate to a larger data set as described in “Copying each file and directory of the aggregate to a larger data set.”
  - You can copy the physical blocks of the zFS aggregate to a larger data set; this procedure is described in “Copying the physical blocks of the aggregate to a larger data set” on page 5.

Copying each file and directory of the aggregate to a larger data set

One method to increase the size of a zFS aggregate is to copy each file and directory of the aggregate to a larger data set. Figure 1 on page 4 shows an example of this approach.

```
//SUIMGVMB JOB ,'EXPAND AGGR WITH PAX',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//* Make sure you have no line numbers in this JCL
//DEFINE   EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=H
//SYSIN    DD     *
     DEFINE CLUSTER (NAME(PLEX.NEW.AGGR002.LDS0002) -
            LINEAR CYL(100 5) SHAREOPTIONS(3) -
            VOLUMES(CFC000 CFC001))
/*
//FORMAT   EXEC   PGM=IOEAGFMT,REGION=0M,
//* On the next line, aggregate and compat must be lower case
// PARM=('-aggregate PLEX.NEW.AGGR002.LDS0002 -compat')
//SYSPRINT DD     SYSOUT=H
//*******************************************************************
//**                                                               **
//**  note - use a + sign at the end of each line to indicate there**
//**         is another line to be processed.                      **
//**         use a ; at the end of each COMMAND                    **
//**                                                               **
//**         a single command can span multiple lines if each line **
//**         ends in a +.  when you have reached the end of the    **
//**         command, terminate the command with a ;               **
//**                                                               **
//*******************************************************************
//PAX1     EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//SYSEXEC  DD  DSN=SYS1.SBPXEXEC,DISP=SHR
//SYSTSIN  DD  *
 OSHELL /usr/sbin/mount -t ZFS -f PLEX.OLD.AGGR002.LDS0002          +
  /service2                                                        ; +
   /usr/sbin/mount -t ZFS -f PLEX.NEW.AGGR002.LDS0002 /service3    ; +
   cd /service2                                                    ; +
   pax -rwvCMX -p eW . /service3                                   ;
/*
//* The result of these next two steps should show that
//* More free space is available in the new file system
//AGGRINF1 EXEC PGM=IOEZADM,REGION=0M,
// PARM=('aggrinfo PLEX.OLD.AGGR002.LDS0002 -long')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
/*
//AGGRINF2 EXEC PGM=IOEZADM,REGION=0M,
// PARM=('aggrinfo PLEX.NEW.AGGR002.LDS0002 -long')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
/*
```

Figure 1. Sample job to copy each file and directory of an aggregate to a larger data set

This approach uses the **pax** command to copy the individual files and directories into an already formatted and empty zFS file system. Both file systems must be mounted. **pax** uses the z/OS UNIX file and directory APIs to read and write each individual file and directory of the hierarchy of the file system. (It does not copy lower mounted file systems because of the **-X** and **-M** options.) You can use the ISHELL command or the **automount** command with the **allocany** or **allocuser** keyword to create the new larger aggregate to copy into with **pax**, because they format the aggregate.

If you are running this job on a system running z/OS V1R13 or later, and the file system was written to using a prior release of z/OS, zFS might use more DASD space for the same data than it did on the prior release. The increase in DASD space can occur for small files (1 KB in size or less) because beginning with z/OS VR13 zFS does not store data in 1-KB fragments; instead, it stores data in 8-KB blocks. For example, if the file system contained 1000 files that are 1 KB in size, on z/OS V1R13 zFS could use a maximum of 10 cylinders more than on previous releases. You can determine how many files are in the file system that are 1 KB or less using the following z/OS UNIX command:

```
find mountpoint  -size -3 -type f -xdev | wc -l
```

After you successfully copy the data, when you are comfortable with the new, larger aggregate, you can delete the old aggregate.

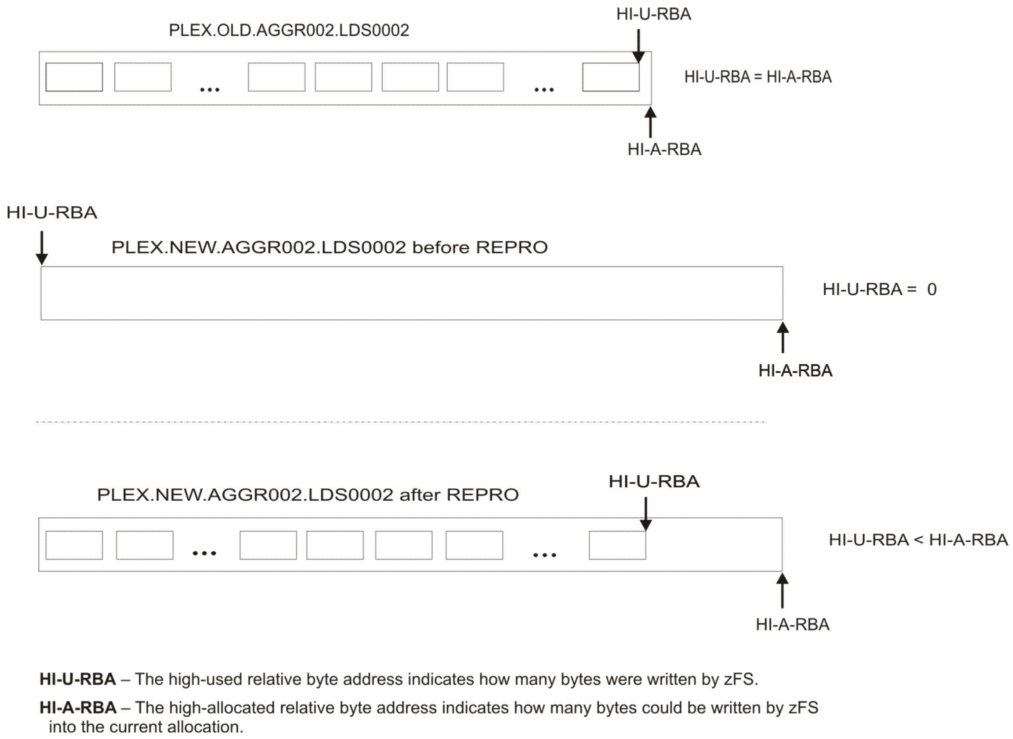
Copying the physical blocks of the aggregate to a larger data set

Another method to increase the size of a zFS aggregate is to copy the physical blocks of the aggregate to a larger data set using the DFSMS REPRO command. This approach is normally faster than using the **pax** command. Do not format the target zFS data set before the REPRO. Figure 2 on page 6 shows an example of this approach.

```
//SUIMGVMB JOB ,'EXPAND AGGR WITH REPRO',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//DEFINE   EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=H
//SYSIN    DD     *
     DEFINE CLUSTER (NAME(PLEX.NEW.AGGR002.LDS0002) -
            LINEAR CYL(100 5) SHAREOPTIONS(3) -
            VOLUMES(CFC000 CFC001))
/*
//LCAT1    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//* This step should show a HI-U-RBA of 0
//* for PLEX.NEW.AGGR002.LDS002
//SYSIN    DD  *
          LISTCAT ENTRIES(PLEX.OLD.AGGR002.LDS0002) -
                  ALL
          LISTCAT ENTRIES(PLEX.NEW.AGGR002.LDS0002) -
                  ALL
/*
//REPRO1   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=H
//* The next line guarantees that the file system is not mounted
//IN1   DD DSN=PLEX.OLD.AGGR002.LDS0002,DISP=OLD
//SYSIN DD *
  REPRO -
     INFILE(IN1) -
     OUTDATASET(PLEX.NEW.AGGR002.LDS0002)
/*
//LCAT2    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//* This step should show the HI-U-RBA of
//* PLEX.NEW.AGGR002.LDS002 equal to the HI-U-RBA
//* of PLEX.OLD.AGGR002.LDS002
//SYSIN    DD  *
          LISTCAT ENTRIES(PLEX.OLD.AGGR002.LDS0002) -
                  ALL
          LISTCAT ENTRIES(PLEX.NEW.AGGR002.LDS0002) -
                  ALL
/*
```

Figure 2. Sample job to copy the physical blocks of an aggregate to a larger data set

Figure 3 on page 7 shows a zFS file system (PLEX.OLD.AGGR002.LDS0002) that is full and a newly-defined zFS data set (PLEX.NEW.AGGR002.LDS0002 before the REPRO) that is larger. PLEX.NEW.AGGR002.LDS0002 has a larger HI-A-RBA than PLEX.OLD.AGGR002.LDS0002. When the blocks from PLEX.OLD.AGGR002.LDS0002 are copied into PLEX.NEW.AGGR002.LDS0002 using REPRO, the result is PLEX.NEW.AGGR002.LDS0002 after REPRO. There is now room to add data to PLEX.NEW.AGGR002.LDS0002.



HI-U-RBA – The high-used relative byte address indicates how many bytes were written by zFS.

HI-A-RBA – The high-allocated relative byte address indicates how many bytes could be written by zFS into the current allocation.

Figure 3. Copying blocks from a full zFS data set into a larger data set

With this approach, the new VSAM linear data set must not be formatted as an empty zFS file system before the REPRO. (If the new data set was formatted, the REPRO would copy blocks to the end of the primary allocation, not the beginning. The data blocks being copied contain all the file system data and the file system information, so formatting is not necessary.) Neither file system needs to be mounted. REPRO uses native VSAM calls to read and write the blocks.

Follow these guidelines:

- When using the REPRO command, do not use the z/OS UNIX **ishell** command or the z/OS UNIX **automount** command with the allocany or allocuser keyword, because those commands will automatically format the aggregate.
- Do not use this approach to copy an HFS file system to a zFS file system because you will be copying the physical blocks of the file system (not the individual files) and the internal format of HFS file systems is different than the internal format of zFS file systems.

Notice that the ZFS attribute is not set in the LISTCAT output for the target data set (PLEX.NEW.AGGR002.LDS0002). It will be set the first time the zFS file system is mounted.

Now the new aggregate can grow into the available space in the allocated portion of the data set or even extend to additional extents if there is space on the volume.

After you successfully copy the data, when you are comfortable with the new, larger aggregate, you can delete the old aggregate.

Chapter 3. Decreasing the size of a compatibility mode aggregate

You can decrease the size of a zFS aggregate using the **pax** command. Figure 4 shows a sample job.

```
//SUIMGVMB JOB ,'SHRINK AGGR WITH PAX',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//* Make sure you have no line numbers in this JCL
//DEFINE   EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=H
//SYSIN    DD     *
     DEFINE CLUSTER (NAME(PLEX.NEW.AGGR002.LDS0002) -
            LINEAR CYL(25 5) SHAREOPTIONS(3) -
            VOLUMES(CFC000 CFC001))
/*
//FORMAT   EXEC   PGM=IOEAGFMT,REGION=0M,
//* On the next line, aggregate and compat must be lower case
// PARM=('-aggregate PLEX.NEW.AGGR002.LDS0002 -compat')
//SYSPRINT DD     SYSOUT=H
//*******************************************************************
//**                                                               **
//**  note - use a + sign at the end of each line to indicate there**
//**         is another line to be processed.                      **
//**         use a ; at the end of each COMMAND                    **
//**                                                               **
//**         a single command can span multiple lines if each line **
//**         ends in a +.  when you have reached the end of the    **
//**         command, terminate the command with a ;               **
//**                                                               **
//*******************************************************************
//PAX1     EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//SYSEXEC  DD  DSN=SYS1.SBPXEXEC,DISP=SHR
//SYSTSIN  DD  *
 OSHELL /usr/sbin/mount -t ZFS -f PLEX.OLD.AGGR002.LDS0002          +
  /service2                                                        ; +
   /usr/sbin/mount -t ZFS -f PLEX.NEW.AGGR002.LDS0002 /service3    ; +
   cd /service2                                                    ; +
   pax -rwvCMX -p eW . /service3                                   ;
//* The result of these next two steps should show that
//* the new file system is smaller
//AGGRINF1 EXEC PGM=IOEZADM,REGION=0M,
// PARM=('aggrinfo PLEX.OLD.AGGR002.LDS0002 -long')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
/*
//AGGRINF2 EXEC PGM=IOEZADM,REGION=0M,
// PARM=('aggrinfo PLEX.NEW.AGGR002.LDS0002 -long')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
/*
```

Figure 4. Sample job to decrease the size of an aggregate

This approach uses the **pax** command to copy the individual files and directories into an already formatted and empty zFS file system. Both file systems must be mounted. **pax** uses the z/OS UNIX file and directory APIs to read and write each individual file and directory of the hierarchy of the file system. (It does not copy lower mounted file systems because of the **-X** and **-M** options.) You can use the ISHELL command or the **automount** command with the **allocany** or **allocuser** keyword to create the new smaller aggregate to copy into, because they format the aggregate.

After you successfully copy the data, when you are comfortable with the new, smaller aggregate, you can delete the old aggregate.

Product Number: 5650-ZOS

Printed in USA

ZSL03178-USEN-01