z/TPF V1.1

TPF Users Group Spring 2008

Title: z/TPF File System Review

Subtitle: Our Greatest Hits

Name: Stephen Record
Venue: DBDF Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Topics

- **z/TPF file systems**
- **Commands for manipulating file systems**
- **Caching**
- **File attributes**
- **File system check utility**
- **Comparison of file system characteristics**
- **Tips for optimizing file system throughput**
- **Backup and restore**
- **File system security**
- **Other new and changed commands**
- **Useful ZFILE commands you may have missed**

z/TPF File Systems

- **Four conventional file systems**
  - TFS z/TPF Collection Support File System
  - MFS Memory File System
  - FFS Fixed-File File System
  - PFS Pool File System
  - See http://ibm.com/tpf/tpfug/tgf05/tgf05g.pdf for details
- **Two pseudo-file systems**
  - PROCFS Process Pseudo-File System
  - SYSFS System Pseudo-File System
  - See http://ibm.com/tpf/tpfug/tgs07/tgs07i.pdf for details

Commands for Manipulating File Systems

- **ZAVFS BUILD**
  - Create or reinitialize an FFS or PFS file system
- **ZFILE mount**
  - Mount a file system or change the attributes of a previously mounted file system
  - Record the mounting of the file system in a TPF record called the MTAB
    - File system will be automatically remounted after an IPL
  - Mounting a new MFS first creates the MFS instance
- **ZFILE mtab**
  - Display or remove entries from the MTAB
- **ZFILE umount**
  - Dismount a file system
  - Remove it from the MTAB
  - Dismounting an MFS also destroys the MFS instance

TFS Inode and Directory Caching

- **Separate logical record caches hold recently referenced directory entries and inodes**
  - TFS_FS_DIR for directory entries
  - TFS_FS_INODE for inodes
- **Cache sizes controlled by ZFINT DC and ZFINT IC commands, respectively**
- **Commands ZFINT DISPLAY, ZFINT USAGE, and ZCACH DISPLAY give helpful information for tuning the cache sizes**

FFS and PFS Record Buffering

- **Record buffering supports the reading/writing of file system records from/to a record buffer area shared by all processes and file systems on the processor.**
- **The record buffer is managed using a logical record cache named TPF_RECBUF.**
- **The use of the record buffering is controlled at the file level through an attribute called a File Service Level.**
- **Each File Service Level defines a set of parameters**
  - How much of the record buffer, if any, may be occupied by records from the file
  - Whether writes are synchronous or not (writes are always synchronous in 1052 state)
  - How many changed records from the file may be buffered, if any, and for how long.
- **File Service Level parameters may be displayed or modified by using the ZDVFS and ZAVFS commands' SERVICE option.**

File Attributes

- **Attributes are keyword=value pairs.**
- **Pre-defined system file attributes may be set or queried.**
  - The system attributes supported vary by file system type
    - File service level (FFS and PFS only)
    - Record IDs to assign
      - Data records (FFS, PFS, and TFS)
      - Object control records (TFS only)
      - Index records (TFS only)
      - Directory records (TFS only)
    - TPFCS DDNAME to use (TFS only)
- **Arbitrary user file attributes may be defined for any file and subsequently set, queried, or deleted.**
- **Attribute manipulations are performed with the ZFILE attr command or through the file system API.**

File System Check Utility

- **Scandisk-like function with fix capability for all file systems (TFS, MFS, FFS, PFS)**
- **Invoked via the ZFILE fsck command**
  - Available also on TPF4.1 as APAR PJ30310 (PUT 20)
- **Ability to check and optionally correct a file system while in use, without requiring a re-initialization of the file system or an IPL**
- **Actual checks and fixes performed are specific to the type of file system under analysis**
- **Typical checks performed:**
  - Scan for lost inodes (files or directories)
  - Scan for dangling directory entries
  - Scan for bad inode data

Comparison of File System Characteristics

	TFS	PFS	FFS	MFS
Data store	Pools	Pools	Fixed file	Heap
Data persistence	Yes	Yes	Yes	No
Processor shared	Yes	R/O	R/O	No
Subsystem shared	No	No	No	No
Mountable	n/a	Yes	Yes	Yes
Caching	Some	Yes	Yes	n/a
TPF 4.1 interoperability	Yes	No	No	No
File attribute support	Yes (z)	Yes	Yes	Yes
Byte range locking	No	Yes	Yes	Yes
ZFILE FSCK support	Yes	Yes	Yes	Yes
Relative throughput	Slowest	Fast	Fast	Fastest

Tips for Optimizing File System Throughput

- **Use MFS, FFS, or PFS where appropriate**
- **Mount an MFS on /tmp**
- **Make judicious use of file service level definitions for FFS and PFS files**
- **Use TFS inode and directory caching**

Backup and Restore

- **ZFILE tar**
  - Standard UNIX command to manipulate tape archives
  - Support headers in either EBCDIC or ASCII
    - Use ZFILE dd command to convert text files as needed
- **ZFILE pax**
  - POSIX command to provide portable archive interchange (x)
  - Support EBCDIC headers only
- **ZFILE dd**
  - Options to simplify conversion of character set and line end sequence for files being moved between z/TPF and Linux or UNIX systems
- **ZFILE df**
  - Standard UNIX command to display disk usage per file system
- **ZFILE du**
  - Standard UNIX command to display disk usage per file

File System Security

- **Control who can manipulate files in the file system**
- **Control who can execute specific operator commands**
- **Based on POSIX file access permissions**
- **Administrator commands**
  - ZOVFS Manage file system users and groups
  - ZFILE prot Control the usage of commands
- **User command**
  - ZPVFS Gain access to the file system
- **APIs for access to the command authorization process**
- **User exit for password encryption**
- **TOS support for password obfuscation**
- **See http://ibm.com/tpf/tpfug/tgf06/tgf06e.pdf for details**

Other Recently Added and Changed Commands

- **ZDSMG DEFINE**
  - Define of a DDNAME which refers to a file system file
- **ZFILE dspsys**
  - Display all records from a individual file or an entire file system
  - Display the contents of the record buffer
- **ZFILE view**
  - Display allocated record addresses and record IDs for an individual file or an entire file system
- **ZCACH DUMP**
  - Dump the contents of the specified logical record cache
- **ZFILE shmm**
  - Manage POSIX shared memory

Useful ZFILE Commands You May Have Missed

- **ZFILE cat**
  - Function: concatenate input sources and write to standard output
  - Most frequent use: display a file on the console
- **ZFILE echo**
  - Function: write data to standard output
  - Frequent uses: display an environment variable; create a short text file
- **ZFILE sed**
  - Function: non-interactive general purpose stream editor
  - Frequent uses: not many – arcane and complex, but very powerful
- **ZFILE tr**
  - Function: copy standard input to standard output with character translation
- **ZFILE xargs**
  - Function: construct an argument list from standard input and run a command
- **… and many of the other familiar POSIX shell commands, too**
  - cd, chmod, chown, cp, export, find, grep, head, hex, kill, ln, ls, mkdir, mkfifo, mknod, ps, pwd, rm, rmdir, tail, tee, touch, unset

Trademarks

- IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.

- Notes
- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.