North American Response Center

HP 3000 APPLICATION NOTE #30

DISC CACHE

HEWLETT PACKARD

JUNE 15, 1987
Document P/N 5958-5824R2725

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HP 3000 APPLICATION NOTES *are published by the North American Response Center twice a month and distributed with the Software Status Bulletin. These notes address topics, where the volume of calls received at the Center indicates a need for addition to or consolidation of information available through HP support services. You may obtain previous notes (single copies only, please) by returning the attached Reader Comment Sheet listing their numbers.*

Note #	*Published*	*Topic*
4	4/15/86	*HP 3000 Printer configuration Guide - Revised*
5	5/01/86	*MPE System Logfile Record Formats*
6	5/15/86	*HP 3000 Stack Operation*
7	6/01/86	*COBOL II/3000 Programs: Tracing Illegal Data*
8	6/15/86	*KSAM Topics: COBOL's Index I/O; File Data Integrity*
9	7/01/86	*Port Failures, Terminal Hangs, TERMDSM*
10	7/15/86	*Serial Printers - Configuration, Cabling, Muxes*
11	8/01/86	*System Configuration or System Table Related Errors*
12	8/15/86	*Pascal/3000 - Using Dynamic Variables*
13	9/01/86	*Terminal Types for HP 3000 HPIB Computers - Revised*
14	9/15/86	*Laser Printers - A Software and Hardware Overview*
15	10/01/86	*FORTRAN Language Considerations - A Guide to Common Problems*
16	10/15/86	*IMAGE: Updating to TurboIMAGE & Improving Data Base Loads*
17	11/01/86	*Optimizing VPLUS Utilization*
18	11/15/86	*The Case of the Suspect Track for 792X Disc Drives*
19	12/01/86	*Stack Overflows: Causes & Cures for COBOL II Programs*
20	1/01/87	*Output Spooling*
21	1/15/87	*COBOLII and MPE Intrinsics*
22	2/15/87	*Asynchronous Modems*
23	3/01/87	*VFC Files*
24	3/15/87	*Private Volumes*
25	4/01/87	*TurboIMAGE: Transaction Logging*
26	4/15/87	*HP 2680A, 2688A Error Trailers*
27	5/01/87	*HPTrend: An Installation and Problem Solving Guide*
28	5/15/87	*The Startup State Configurator*
29	6/01/87	*A Programmers Guide to VPLUS/3000*

DISC CACHE

INTRODUCTION

We receive many questions on the subject of DISC CACHING. What is disc caching? How does it work? What TYPES of caching are available? This note will address these questions.

Hewlett Packard offers two types of disc caching: "XP" or hardware onboard memory built into the disc controller, and MPE disc caching handled by software. We will first disucss MPE (disc) cache.

MPE CACHE

What is MPE CACHE?

MPE (disc) CACHE is an optional MPE subsystem that utilizes excess memory and CPU time to keep portions of frequently referenced disc "domains" in memory. MPE CACHE manages the transfer of the disc domains between main memory and disc storage. No portion of main memory is permanently dedicated to disc cache, it is shared with other types of MPE segments and treated the same by the memory manager. MPE CACHE is a software process.

MPE CACHE was implemented to make data available to the CPU in a more expedient manner. Since memory access is considerably faster than Disc I/O, MPE cache is used to store large chunks of data (in main memory) in hopes that it would be needed in the near future. Thus reducing slow disc seeks and data transfer overhead, and less CPU idle time waiting around for disc I/O. MPE speeds up data access to the CPU.

What is a CACHE DOMAIN?

A cache domain is a copy of a portion of disc, residing in main memory. A disc domain is considered "cached" when it is in memory and "mapped" when there is I/O pending against it. A disc is never mapped in more than one place in the cached domains; the cache domains never overlap. They are treated exactly like data segments by the memory manager therefore, the amount of memory which is cache domain is variable.

How does MPE CACHE work?

When a process generates a read request from the disc, the list of currently cached disc domains of the specified device is searched. If the requested disc domain is present in main memory, the read operation is completed in 1/20 of the normal time. This process will continue executing without an interruption. If the cache domain is not found in memory, then a physical disc request will be initiated.

When a process requests a write and there is currently a write pending against the specified disc domain, the request is queued until the pending write is posted to disc. If a process requests a write to disc and it is not in cache (it is in main memory) an available region of memory is obtained, and is used to map the corresponding disc image. Once this data move takes place a post to the disc is initiated. Only the portion of the cached disc image that is modified is posted. After the move to the disc image is performed and the post to disc is initiated, the writing process is allowed to continue running without having to wait for the physical post update to complete.

With these mechanisms and strategies, MPE disc caching reduces the traffic between main memory and disc storage, and reduces delays to read or write disc information at the cost of additional memory and CPU overhead.

How do I use MPE disc CACHING commands?

MPE CACHE is a software process which can be optionally installed on your system. Disc CACHING is transparent to system applications. Once the software is installed the only thing needed to make it work is the use of the following MPE commands:

To turn on disc CACHING use the command:

:STARTCACHE (ldev)

To turn off disc CACHING use the command:

:STOPCACHE (ldev)

The cached domains in memory are managed by the system just as any other memory structure is managed. There is no special memory area reserved for CACHING, the amount of memory used is a function of the total memory demands on the system at any given time.

The :SHOWCACHE command gives a lot of very useful information. For example:

```
:SHOWCACHE


DISC     CACHE      READ  WRITE          PROCESS                  % OF    CACHE
LDEV     REQUESTS   HIT%  HIT%   READ%   STOPS      K-BYTES       MEMORY  DOMAINS
-------------------------------------------------------------------------------
1        1732764    94    71     82      129893     3822          23      1113
2        595748     76    93     78      144770     12823         7       386
3        606932     75    92     81      136951     14861         8       455
4        619519     86    78     73      80249      85            0       97
5        356516     76    92     63      62193      1653          10      553

TOTAL    35611479   86    82     78      554056     8244          50      2604

67% of user I/Os eliminated.
Data overhead is 674k bytes.
Sequential fetch quantum is 96 sectors.
Random fetch quatum is 16 sectors.
Block on Write = NO
```

WHERE:

CACHE REQUESTS - The number of non-memory management (I/O) calls that were made to disc.

READ HIT % - The percentage of the disc read requests that were found in the cached domains resident in memory. From a performance aspect, there is no such thing as this number being too high. However, giving the system more memory may not cause this percentage to increase if an adequate amount of memory is already available and/or your applications mix has only limited locality.

WRITE HIT % - The percentage of write requests that were satisfied by having the desired object already in memory. Write hits are undesirable since they represent updates to existing records or writes to areas previously initialized. This situation causes more work for cache management and causes requests for other records in the cached block to queue up until the write has completed. Things that do not cause "write hits" are additions to the end of files. for example, writing a sort output file or a logging file.

READ % - Read requests as a percentage of total requests. This figure multiplied by the read hit % gives the percentage of the disc I/O operations that were eliminated by the use of disc CACHING.

PROCESS STOPS - The number of times the requesting process had to stop because the object was not yet in memory. This can occur because the disc record that you want is not in a cached domain and needs to be set up as such or because someone else needed the same domain before you did but it is not yet in memory.

MEMORY USAGE - The amount of memory that is being used for cached domains. There is no "right" amount of memory since more is (almost) always better but not necessarily economically justifiable.

DATA OVERHEAD - The amount of overhead involved in the management of CACHING. This includes not only the structures that keep track of the cached domains but also the region/subregion header and trailer overhead attributable to those domains.

SEQUENTIAL FETCH QUANTUM - The approximate size of the block retrieved for sequential read requests. This can be manipulated by using the :CACHECONTROL command.

RANDOM FETCH QUATUM - The approximate size of the block retrieved for random read requests. This can be manipulated using the :CACHECONTROL command.

BLOCK ON WRITE - This is a mechanism to set block on write globally across all discs and processes. This can be manipulated using the :CACHECONTROL command.

HARDWARE CACHE

What is HARDWARE CACHE?

HARDWARE CACHE (CONTROLLER CACHE or XP CACHE) is currently available on the 7933/35/37XP disc drives. The following discussion pertains to the 7933/35XP drives *only*. XP is a higher performance version of the 7933/35 H disc drives that incorporates 1 Megabyte (4096 bytes) of memory on the controller. The added memory is used to contain the last 255 active areas of the disc so that future accesses to these same areas will be immediately available to the CPU without a physical read actually occurring. The object of this XP ("CONTROLLER CACHE" MEMORY) is to decrease the overall

disc response time and increase system throughput. The amount of system performance improvement will vary depending upon individual system configuration and program applications.

How does HARDWARE CACHE work?

When the CPU requests a read, XP CACHE memory is searched. A read hit occurs when the entire request is already in memory, and the data is transferred to the CPU immediately. Otherwise, a physical read from the disc is initiated. Write caching is not implemented on the 7933/35XP drives.

How do I use HARDWARE CACHE?

By default, XP cache is always enabled. It is a part of the firmware within the disc drive. If, however, the firmware detects a problem with XP cache, it will disable itself. This benefit allows the disc drive to remain online.

CONCLUSION

MPE or HARDWARE CACHE will, in most cases, improve system performance. However the determination of which TYPE of CACHING to use can be confusing. With HARDWARE CACHING the disc drive does a search through its own memory to satisfy the disc request. The advantage of HARDWARE CACHING is that CPU resources are not impaired by waiting for disc I/O. In both cases data transfer is optimized if the request is satisfied from CACHE. The determination as to which type of cache to use is beyond the scope of this article. It requires the use of advanced performance tools, and performance consulting. Either form of CACHING can actually degrade performance. In the case of MPE CACHE, if there is not enough memory for CACHING to use, or CPU utilization is very high, then the overhead to use MPE CACHE is relatively costly. In this case more I/O throughput is actually achieved with CACHING turned *off*. In the case of XP CACHE, if a large percentage of disc requests are not satisfied by XP CACHE the cost of always searching cache, added to the physical I/O (s) performed, will cause system performance degradation. However, with proper implementation and selection of cache type, system performance will be improved.

READER COMMENT SHEET

North American Response Centers
HP 3000 Application Note #30: DISC CACHE
RC Questions & Answers (JUNE 15, 1987)

We welcome your evaluation of this Application Note and attached RC Questions & Answers Sheet. Your comments and suggestions help us to improve our publications. Please explain your answers under Comments, below, and use additional pages if necessary.

	AppNote	RC Q&A
Is this publication applicable to your site?	☐ Yes ☐ No	☐ Yes ☐ No
Are the concepts and wording easy to understand?	☐ Yes ☐ No	☐ Yes ☐ No
Would you like to see additional Notes on this subject?	☐ Yes ☐ No	☐ Yes ☐ No

Comments and/or suggestions for future Application Notes:

This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

FROM: **Date** _____

Name _____

Company _____

Address _____

_____

_____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1710, SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Application Note / RC Q & A Comments
Hewlett-Packard Western Response Center
3300 Scott Boulevard
Santa Clara, CA 95054

FOLD FOLD

HEWLETT-PACKARD RESPONSE CENTER QUESTIONS & ANSWERS

HP 3000 Questions Commonly Received by the North American Response Centers

Q. What are the maximum record lengths that the HP3000-to-Non-HP batch communications subsystems (RJE, MRJE, SNA/NRJE) can transmit to and receive from the host or peer system?

A. The following table describes the maximum supported record lengths for the three communications subsystems:

Subsystem	Type Communication	Max. Rec. Lengths Supported (in Bytes) <Transmit>	<Receive> Print (1)	<Receive> Punch
RJE	To PEER	256	256	256
	To HOST	80	256	256
MRJE	To HOST	80	255	80
SNA/NRJE-I (2)	To HOST	80	133	80
SNA/NRJE-II (2)	To HOST	252 (3)	255	255
SNA/NRJE-II (2)	To HOST	248 (4)	255	255
SNA/NRJE-II (2)	To HOST	128 (5)	255	255

Notes:
(1) - Including Carriage Control Byte
(2) - NRJE versions prior to A.52.00 are considered NRJE-I. Versions A.52.00 and after are considered NRJE-II.
(3) - NRJE, JES3 using a 512 byte RU.
(4) - JES2, JES3 using a 256 byte RU.
(5) - VSE/POWER

Additionally, the maximum record lengths selected must match on the HP3000 and the host or peer side. That is, coordination between the HP3000 site and the host or peer site is required to insure that the transmitter's and the receiver's configured record lengths match exactly, device for device.

To transmit or receive records longer than the supported maximum record lengths requires user written "cut and paste" applications which split longer records on one system to a transmittable size and reassemble them on the receiving system.

Q. I have created a forms file and REFSPEC file in a test account. Now I want to FCOPY them to a production account. When I do this, REFSPEC will not let me change the name of the associated forms file. How can I do this?

more...

A. Copying Form files and Reformat files from one group/account to another can be accomplished with doing the following steps exactly in the order they are written below.

1. :FCOPY the files to a new account. (account #2).
2. :RENAME the original form file in the old account (account #1).
3. Recompile the form file in the account #2.
4. ;RUN REFSPEC
5. Form file menu will appear with old form file from account #1 fully qualified.
6. Change this to form file from account #2 fully qualified.
7. Recompile reformat file
8. :RENAME the original form file in account #1 back if desired.

The rename in steps two and eight can be avoided by using the MPE command :RENAME instead of :FCOPY, (if possible) in step one. If the form file in account #1 is left with the same name as the form file in account #2, the following error message will occur:

INTERNAL ERROR IN WRITING SOURCE REC TO REFORMAT FILE (FSERR 40)

Q. I have received the messages ("DCU LOGGING IN PROGRESS" and "DCU LOGGING COMPLETE") on my console. What do these messages mean?

A. DCU logging is, in effect, "hardware logging". Any type of hardware interrupt detected by the DCU, (starts, halts, loads, dumps and power failures) is logged to a log file register on the DCU board. The DCU log will be dumped to a MPE logfile (type 47) at a predetermined time interval or log size. The time interval can be set by using the DCU log command:

C>LD <dump interval> [,<min log size>]

The system defaults call for the DCU logs to be interrogated every 12 hours and must have 22 entries before MPE can request a download of the system logs. When the DCU log file is transferring to the MPE log the message "DCU LOGGING IN PROGRESS" appears. After it finishes the procedure, the console message "DCU LOGGING COMPLETE" is sent, and the hardware log file register is emptied.