Commerical Systems Division

HP3000 PERFORMANCE NOTE #1

Optimizing
Program Performance
on MPE XL

HEWLETT PACKARD

September 15, 1989
Document P/N 5960-0602

COMMERCIAL SYSTEMS DIVISION PERFORMANCE NOTES

HP CSY PERFORMANCE NOTES are published by the Commercial Systems Division and are distributed with the Software Status Bulletin.

This article is the first in a series of three "PERFORMANCE NOTES" which comprise a "Pilot" program. After reading the article, please take time to jot down any suggestions or ideas you may have. A postage-paid "Reader Comment Sheet" is provided for your convenience. Use it to suggest improvements or topics that you would like to see addressed in future PERFORMANCE NOTES. At the conclusion of the Pilot program, all "Reader Comment Sheets" received will be evaluated to determine whether or not a need exists for this type of information, so please cast your vote. PERFORMANCE NOTES will either be made a permanent feature of the Software Support Bulletin (SSB) or discontinued. The plan is to eventually include all PERFORMANCE NOTES on LaserROM and in the SupportLINE database.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. Permission to copy all or part of this document is granted provided that the copies are not made or distributed for commercial advantage; that this copyright notice, and the title of the publication and its date appear; and that notice is given that copying is by permission of Hewlett-Packard Company. To copy otherwise, or to republish, requires prior written consent of Hewlett-Packard Company.

Copyright © 1989 by HEWLETT-PACKARD COMPANY

OPTIMIZING PROGRAM PERFORMANCE ON MPE XL

HOW TO ATTAIN OPTIMAL PERFORMANCE
in order of priority

(1) CONVERT PROGRAM AND LIBRARY FILES TO NATIVE MODE

(2) TRANSLATE PROGRAM AND LIBRARY FILES USING THE OCT UTILITY

READ ON FOR MORE DETAILS...

INTRODUCTION.

Generally, optimal program performance is achieved when running in native mode on a 900 Series. Performance costs are most expensive when running a compatibility mode program file in emulated mode. If factors such as application size, availability of source code, or lack of time prevent immediate recompilation to native mode, the compatibility mode program files should be translated via a utility called the Object Code Translator (OCT).

NATIVE MODE.

Native Mode (NM) is the natural operating mode of the 900 Series HP3000 computer. Programs that run in native mode are those compiled on a 900 Series system using one of the native mode compilers (e.g., COBOLII/XL). Native Mode programs use MPE XL instructions directly thereby taking full advantage of features and functionality inherent to the new architecture. No intermediate translation or emulator overhead is incurred. THUS, PURE NATIVE MODE PROGRAMS YIELD MAXIMUM PERFORMANCE ON MPE XL.

COMPATIBILITY MODE.

There are two types of compatibility mode (CM) programs that can be run on a MPE XL-based system. The first type, emulated, includes both MPE V/E program files that have been :RESTOREd from a V/E-based machine as well as program files that have been created directly on a 900 Series system using one of the compatibility mode compilers. For example, the COBOLII/V compiler on the MPE V/E-based system is exactly the same as the COBOLII/V compiler on the MPE XL-based system. Thus, these identical compilers will produce identical program files that when executed will have identical performance.

The second type of compatibility mode programs is "translated". Emulated programs become translated when they are run through the Object Code Translator (OCT) utility. OCT will append the sequence of

MPE XL instructions to the program or SL file for future usage. This eliminates the need for fetching and decoding instructions each time the program is run. The translated program file will consume substantially more disc space than the original file. However, additional disc space consumed will usually be more than offset by performance gains derived. For example, an emulated pascal program which was computational in nature took 32.03 CPU seconds to run on MPE XL. After being translated, elapsed time was reduced to 18.99 CPU seconds for a 40% performance gain. Translating a program file, however, is only an interim method of improving performance. A translated program still operates in compatibility mode. To attain optimal performance, the program file must be migrated to native mode. IF CONVERTING TO NATIVE MODE IS NOT POSSIBLE, TRANSLATING THE CODE VIA OCT IS THE NEXT BEST OPTION.

WHAT HAPPENS AT RUN TIME?

At run time, the MPE XL Loader will check the file code to determine if the program file is a native mode or compatibility mode program file. If it is a native mode program file, the program will be launched. If the MPE XL Loader determines that the program file is a compatibility mode program file, it will verify whether or not the file contains code produced by the Object Code Translator. If it does, the program will be launched. If not, the MPE XL Loader will call the Emulator. At run time, the Emulator will interpret each MPE V/E machine instruction calling a functionally equivalent sequence of HP Precision Architecture machine instructions.

MIXED MODE PROGRAMS AND SWITCHING.

A program can run partly in native mode and partly in compatibility mode. This is referred to as a "mixed-mode" program. MPE XL has a Switch subsystem which resolves differences between native mode and compatibility mode access. MPE XL transparently switches between modes for system routines. For example, if a user-written compatibility mode program accesses TurboIMAGE, a database subsystem written in native mode, the MPE XL operating system will handle all switches between the two modes. If, however, the user-written program accesses a user-written subprogram in a native mode library, the end user must write their own switching routines. Refer to the "Switch Programming User's Guide" for details.

Native mode to compatibility mode switches are usually more expensive in terms of resource expenditures than are compatibility mode to native mode switches. When switching from native to compatibility mode, all parameters must be copied to compatibility mode data structures, thus resulting in the additional overhead. Compatibility mode programs are still restricted to a 16 bit address range; therefore, items passed by reference may have to be copied to areas within the address range of the compatibility mode program. When a native mode procedure passes a large string to a compatibility mode procedure, the string is copied to a 16 bit address range. This is done for addressability enabling compatibility mode code to address the data. When a compatibility mode procedure calls a native mode procedure, this is not an issue, since native mode code can access the full address range.

For example, a mixed-mode application may consist mainly of user programs created with FORTRAN77/XL (native mode). In addition, there are a few program files that were created with the compatibility mode compiler FORTRAN77/V. All subroutines for this application have been written in HP PASCAL/XL and are contained in a native mode library. In this case, migrating those remaining few FORTRAN77/V program files to native mode would net a substantial performance increase by eliminating mode switching.

The operating system contains many program files and libraries which are often referred to as "system code". Those programs and libraries that are created by end users are frequently referred to as "user code". The term "program" is inclusive and can refer to either or both. Typically, a program created by

an end user will spend 80 to 95% of its execution time (i.e. elapsed time or "CPU time") in system code and only 5 to 20% of its CPU time directly in the user written code. Examples of system code include: directory structures being updated, various subsystems being accessed (e.g., KSAM, message files, TurboIMAGE), and file intrinsics being executed.

Performance analysis, therefore, becomes more complex when assessing mixed mode applications. BEGIN BY IDENTIFYING EACH PROGRAM AND LIBRARY CONTAINED IN THE APPLICATION. Does it execute in compatibility mode, native mode, or a combination of both? Have all compatibility mode libraries and programs been translated? Which compatibility mode libraries and programs should be migrated to native mode first? THOSE PROGRAMS CONSUMING THE MOST CPU TIME AND SPENDING THE LARGEST PERCENTAGE OF TIME IN USER CODE SHOULD BE MIGRATED FIRST. The goal is a 100% native mode application for optimal performance. If a program spends only minimal time in user code, it is a good idea to migrate that program anyway to eliminate switching overhead.

MONITORING APPLICATION PROGRAMS.

One way to identify the most heavily used programs is by monitoring the 900 Series system during a significant period of the business day. Two HP tools can be utilized to view performance of executing applications. HP GLANCE/XL can be used to make an initial assessment of executing programs and processes. HP GLANCE/XL provides global, job-session, and process level data on CPU, memory, and I/O activity. HP LaserRX reveals the detailed information needed for analyzing application programs, including data on compatibility mode and native mode switching. HP LaserRX operates with very low system overhead while capturing global, program, and process level data. One feature allows the user to pre-define an application in terms of groups of programs. This facilitates performance information management through selective reporting on key programs.

CONCLUSION

Through analysis of application programs, performance improvements can be gained. Programs running in native mode exploit the higher performance and expanded addressing capabilities of the HP Precision Architecture. THE GOAL IS TO HAVE ALL APPLICATIONS EXECUTE IN NATIVE MODE. Establish a strategy and appropriate timetable to migrate any remaining compatibility mode programs to native mode. In the interim, be sure to at least translate those remaining compatibility mode program files with the Object Code Translator (OCT). Optimization of application programs results in improved system performance.