HEWLETT-PACKARD

HP 3000 Series 925 and 925LX

Performance Planning Guide

MPE XL A.01.00

Series 925 and 925LX Performance Planning Guide

MPE XL Version A.01.00
HP 3000 Computer Systems

HEWLETT
PACKARD
19111 Pruneridge Avenue
Cupertino, CA 95014
U.S.A.

Table of Contents

Introduction

Objectives

This Performance Planning Guide will help you evaluate the Series 925 or Series 925LX systems for your application needs. The information contained herein will be useful to customers considering an upgrade from their Series 5X systems, as well as potential customers considering an HP 3000 solution for the first time.

This Guide presents benchmark information specific to MPE XL version A.01.00. Performance enhancements are already being tested in the lab for the next major release of MPE XL. Future Guides will present updated performance numbers for this release.

Relative performance of the Series 5X machines to the Series 925 will vary greatly depending upon the characteristics of the application. Applications that take advantage of the new HP Precision Architecture features will perform better than those that do not. This document will provide you with guidelines for evaluating your application's performance potential on the Series 925 machines.

Overview

The first section of this guide introduces the basic features of HP Precision Architecture (HPPA) and how they relate to system performance. An overview of the Series 925 and Series 925LX, and where they fit in the family of HPPA computers, is included.

Key factors that affect Series 925 performance are discussed in detail in the second section. Since performance varies from application to application, this section also discusses the application characteristics that weight the impact of these performance factors.

The third section presents the results of two application benchmarks contrasting the Series 58 with the Series 925. Analyses of the results are included.

In the last section of the Guide, an overview of the performance upgrade options for the Series 925 machines is presented.

HP Precision Architecture

HP Precision Architecture is based on the concept of Reduced Instruction Set Computing (RISC). The basic RISC design features are:

- Simple, efficient instruction sets
- Fixed-length, fixed-format instructions
- Load/Store design
- Hardwired instructions
- Single-cycle operation
- Optimizing compilers

Simple, efficient instruction set

The HPPA instruction set was built from the ground up, using only the most frequently used instructions. These simple instructions are the building blocks for more complex and specific operations. Hardware is not wasted on rarely used complex instructions.

Fixed-length, fixed-format instructions

By using instructions that are fixed-length and fixed-format, the HPPA design facilitates pipelining. Pipelining overlaps instruction processing so that one instruction can begin to execute before the previous one has finished. Pipelining is used effectively at the instruction level when the start and end points of the instruction can be easily evaluated, and when the fixed length of all instructions keeps the pipeline consistently full.

Load/Store design

The Load/Store design of RISC allows only Load and Store instructions to access memory. Load instructions transfer data from memory to CPU registers, and Store instructions transfer data from registers back to memory. This simple control scheme ensures that all computations are performed register-to-register or between a register and a constant located in the instruction. This design, combined with optimizing compilers, ensures that frequently used data is managed within registers to minimize memory access.

Hardwired instructions

Because instructions are simple and the machine uses a Load/Store design, instructions can be hardwired. Simple, hardwired instructions run faster than complex instructions that need to be decoded by microcode.

Single-cycle operation

Simple, hardwired instructions that are efficiently pipelined allow HPPA to execute an instruction on virtually every machine cycle. Maximum efficiency is reached when no machine cycles are wasted. Since machine cycles are shorter with RISC architecture, the result is higher performance.

Optimizing compilers

Optimizing compilers ensure that instructions execute in the most efficient order. They also take advantage of the HPPA delayed-branch capability. In many pipelined systems, when a branch instruction is loaded, the next instruction in the sequence is loaded but not executed. Instead of wasting a machine cycle, HPPA optimizing compilers schedule other instructions to be executed around the branch instruction. The result is more efficient pipelining, and minimized program execution time.

The midrange Series 925 is the low entry point to the HP Precision Architecture design. The Series 925LX provides the same level of performance as the Series 925, but is designed to satisfy smaller configuration needs.

Performance Factors

Performance improvements on the Series 925 can be attributed to several factors in HP Precision Architecture (HPPA) and the MPE XL operating system. The extent to which these factors affect performance depends on the characteristics of the application being run. Since application characteristics vary, overall performance varies.

This section outlines the key factors that affect Series 925 performance and examines how application characteristics reduce or increase the impact these factors have on performance. The benchmark analyses are written in terms of these factors and characteristics.

The key factors that affect the performance of the Series 925 are:

- Native mode code is more efficient than compatibility mode code.
- MPE XL's extended address range eliminates the need for extra data segments.
- Memory mapped I/O allows data to be accessed at memory access speeds.
- MPE XL's larger memory capacity reduces the number of physical I/Os.
- Transaction Management reduces the number of database physical I/Os.
- HPPA's new floating point coprocessor significantly accelerates floating point calculations.

Native mode code is more efficient than compatibility mode code

MPE XL supports a 16-bit MPE V compatible environment, called compatibility mode (CM), concurrent with a native 32-bit environment called native mode (NM). Access to functions in compatibility mode is accomplished by a switch mechanism allowing procedure calls from native mode to procedures in compatibility mode, and vice versa.

Native mode code is more efficient than compatibility mode code since the compatibility mode instructions (non-HPPA) are emulated using one or more HPPA instructions. In many applications, compatibility mode activity will have an adverse effect on performance. Migrating the application to native mode will often reduce the impact of compatibility mode processing on performance. The impact of this factor on overall performance depends on two application characteristics.

First, the percentage of time the application spends executing user code will determine the extent of performance improvement. The CPU time used by an application can be divided into two categories, direct time which is time directly spent processing user code and indirect time which is CPU time spent in system code on behalf of user code. Indirect time includes system services that have been requested by the user and those that are managed by the operating system. Some examples of indirect time are file system management, terminal handling, extra data segment management, and disc caching.

The larger the percentage of time an application spends executing user code on a Series 5X, the greater positive effect this factor will have on performance when the application is migrated to native mode on the Series 925.

The second characteristic to determine the weight of this factor is the frequency with which an application uses compatibility mode system services on MPE XL. Examples of current CM system services include management of KSAM, message, and SPOOL files. Applications that have been migrated to NM will incur additional overhead as a result of switching from one mode to the other when invoking these services.

The less frequently an application switches to these CM system services, the greater the positive impact on performance. Moreover, as CM services are migrated to NM in the future, the performance benefits of migrating these user applications to NM will be greater.

It should be noted that for some applications NM and CM performance varies only slightly. As a result, some customers and independent software vendors choose to maintain their application in CM. We recommend discussing the use of native mode vs. compatibility mode with your systems engineer as there are services available from Hewlett-Packard to help you with these decisions.

MPE XL's extended address range eliminates the need for extra data segments

MPE V applications use extra data segments to access data beyond the 64K byte stack limit. Using extra data segments improves the user's ability to reference data in memory rather than access disc. Managing extra data segments in user code, as well as using the necessary additional system resources, increases the amount of CPU required to access these data structures.

MPE XL's extended address range eliminates the need for extra data segments, since data structures are no longer bound to 64K bytes. The design of MPE XL allows large data structures to be accessed directly without additional system overhead.

This factor translates into significant performance gains in applications characterized by heavy use of extra data segments. The overhead of managing extra data segments is eliminated when the user removes references to extra data segments from their code.

Memory mapped I/O allows data to be accessed at memory access speeds

MPE XL takes advantage of HPPA's large address space and increased memory capacity to manage large volumes of disc file data efficiently in main memory. All open disc files are mapped directly into virtual address space. This allows the data to be accessed at memory speeds and eliminates a substantial number of disc I/O operations. This factor contributes to improved throughput in any application.

MPE XL's larger memory capacity reduces the number of physical I/Os

The large memory configurations possible with HP Precision Architecture enable more data to reside in memory, and thus fewer physical disc I/Os are required. The greater an application's ability to keep data in memory, the greater positive impact this factor will have on performance.

Transaction Management reduces the number of database physical I/Os

Transaction Management collects and buffers database modifications before posting them to disc, thereby decreasing the number of physical writes. The more an application makes use of Transaction Management (e.g. with TurboIMAGE), the greater positive effect this factor will have on performance. Transaction Management also insures the physical integrity of databases.

HPPA's new floating point coprocessor significantly accelerates floating point calculations

The floating point coprocessor works in parallel with the CPU, performing floating point calculations while the CPU performs integer calculations and other functions. This factor has a dramatic positive impact on performance.

Performance Benchmarks

Overview

Operating System Versions

To ensure consistency, all benchmarks were run on the same version of MPE V or MPE XL. The version of MPE V software used for the Series 58 benchmarks was UB-MIT (G.A2.B0). This version of MPE contains an improved memory manager, optimized disc caching and the expanded segmented library.

The Series 925 benchmarks were run on version A.01.00 of MPE XL. Operating system tuning on MPE XL is continuing and performance gains can be expected from the future releases of MPE XL.

Hardware Configurations

MPE V benchmarks were run on an HP 3000 Series 58 with eight megabytes of memory and six HP 7933/35 disc drives. MPE XL benchmarks were run on an HP 3000 Series 925 with 48 megabytes of memory and six HP 7933/35 disc drives.

Benchmark Applications

The benchmarks on the following two pages are examples of traditional applications found on HP 3000 systems today. The results show the Series 925 performing about 60%-70% faster than the Series 58 in interactive and batch environments.

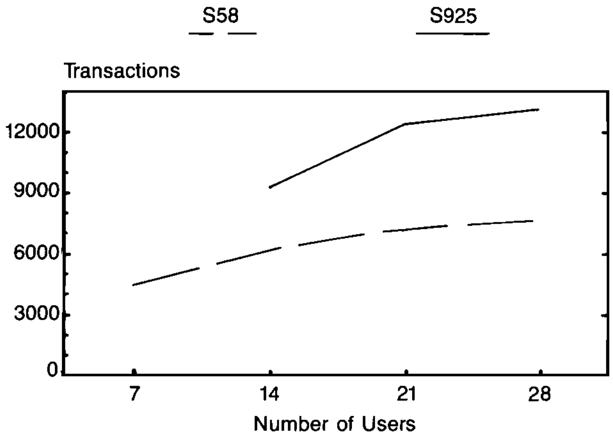
HP Precision Architecture has opened up new application opportunities to gain exceptional performance. For example, applications that perform intensive mathematical computations and CPU-intensive applications such as SQL maximize the benefits of HP Precision Architecture. While no formal benchmarks have been performed to date on the Series 925 with these types of applications, results from benchmarks performed on another MPE XL machine, the Series 950, show these applications realizing dramatic performance gains.

V20 Interactive Benchmark

Description

The V20 benchmark is a database intensive interactive application that performs library management. The data resides in nine TurboIMAGE databases which occupy 89 megabytes of disc storage. It is written primarily in COBOL II, with some PASCAL procedures used to handle file and database access. This application has been migrated to native mode.

There are two types of transactions in this application: search and circulation. Search transactions look up books in the card catalog by a particular author or subject. Circulation transactions perform book check-in and check-out.



Relative throughput: 1.6

Figure 1. V20 Throughput

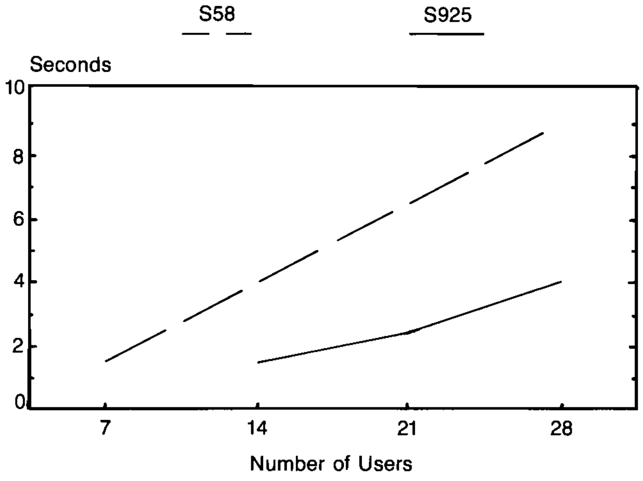


Figure 2. V20 Response Time

Benchmark Results

Two of the factors contributing to the increased performance of this benchmark are MPE XL's larger memory capacity and Transaction Management. A characteristic of this application that enables improved throughput and response time is database size. Much of the database fits in memory at one time, taking advantage of MPE XL's larger memory capacity to reduce disc I/O. Transaction Management further reduces physical I/Os by buffering database writes before they are posted to disc.

The performance gain of this benchmark is limited by the heavy use of message files from a user written logging function. The application forces each write to be posted to disc, thus detracting from the Series 925's ability to reduce physical I/Os through larger memory capacity. The forced posting on the message file degrades performance more at this time than it may in the future because message file processing routines are currently in compatibility mode. In addition, the switching from one mode to another also incurs some overhead.

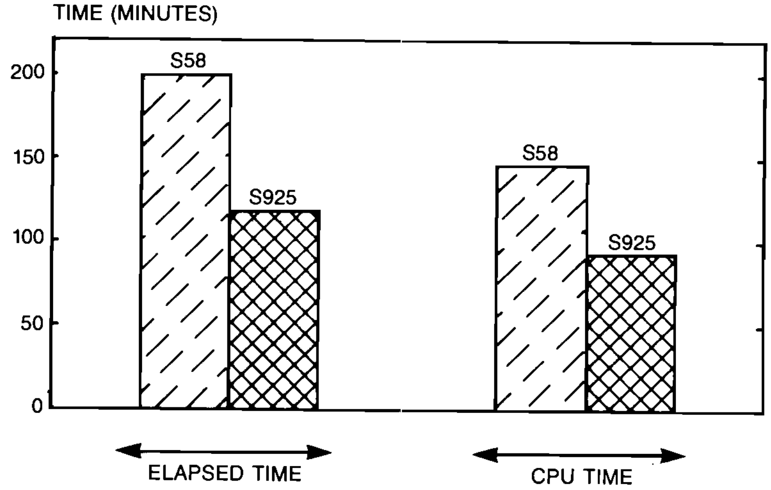


Figure 3. Material Requirements Planning (MRP) Batch Benchmark

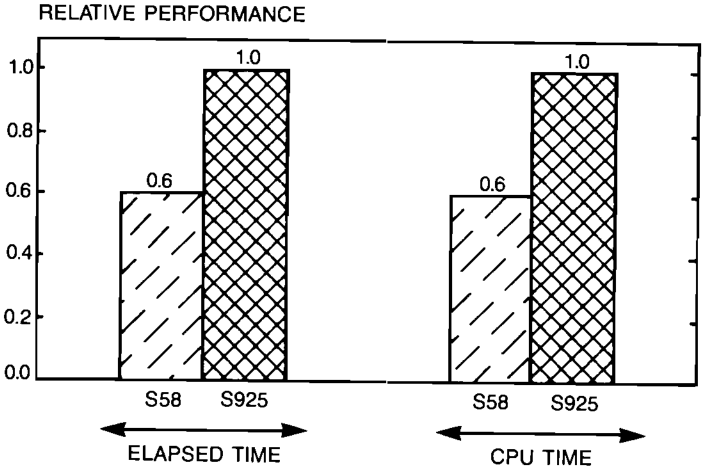


Figure 4. Material Requirements Planning (MRP) Batch Benchmark

Material Requirements Planning (MRP) Benchmark

Description

The MRP benchmark is a batch application that performs Material Requirements Planning functions. The user programs are written in COBOL II and recompiled into native mode, with the exception of a third-party subroutine which is in compatibility mode.

The main program in this test reads and processes a transaction file to perform adds/changes/deletes to a TurboIMAGE database. Transactions are then classified and written to another file for later reporting purposes.

A son process is created to delete completed transactions, and the third-party subroutine is called to read and sort large amounts of data.

Benchmark Results

Performance in this benchmark is limited by frequent switches from native mode to the compatibility mode subroutine.

This is a write intensive application that suffers from a disc I/O bottleneck on the MPE V system. Transaction Management, by reducing physical writes, alleviates the I/O bottleneck and improves performance on the Series 925.

MPE XL Disc Requirements

It should be noted that installations migrating from an MPE V system to an MPE XL system will need to plan for additional disc space. This additional requirement is due to an increase in the size of the Fundamental Operating System (FOS), Transaction Management (MPE XL's new facility to insure physical integrity of databases), and transient disc space (MPE XL's implementation of MPE V's virtual memory).

The amount of increase varies widely depending on the average number of jobs and sessions, the number of volume sets, and the amount of file space currently being used on the MPE V system. Disc space requirements for your installation should be discussed in detail with your account representative.

Upgrade Options

Customers who purchase a Series 925LX can easily field upgrade to the Series 925. While the processor speed does not change, this upgrade provides an additional eight megabytes of memory and enables the system to support the expanded Series 925 configuration capabilities.

The Series 925 can be expanded with peripherals and an additional I/O Bus. To achieve greater performance, the Series 925 can be field upgraded to the Series 935. This upgrade will provide greater expandability, more CPU cache, faster floating point processing, and up to 70–100% increase in performance.

Glossary

BRANCH
A machine instruction that alters the sequence of instructions being executed by the CPU.

Cache
A small, high-speed buffer between memory and the CPU that holds all or part of an executing program and/or data, making the information more readily accessible.

Instruction Set
The set of all possible machine instructions understood by the computer. The instruction set defines the computer's architecture.

LOAD
A machine instruction that tells the CPU to take data from memory and place it in a register.

Locality
The tendency of programs to reference storage in non-uniform, highly localized patterns.

Machine cycle
The period of time required by a computer to perform the most fundamental operation.

Registers
Small, high-speed devices within the Execution Unit of the CPU where information is held temporarily.

Response time
Response time is the time that the user waits for a reply from the computer after entering data on a terminal. Response time is more formally defined as the time from terminal read completion to the prompt for the next terminal read.

STORE
An instruction that tells the CPU to write data from a register into memory.

Switch
MPE XL switches between native mode and compatibility mode transparently. This transparency in operation is made possible by MPE XL's Switch Subsystem which determines whether code is in native mode or compatibility mode, and automatically switches between modes as needed while an application is running.

Throughput
A measure of work accomplished per unit of time. Throughput is typically given in terms of terminal transactions per hour.

Transaction Management
A new feature of MPE XL which provides physical consistency of data through concurrency control, logging, and recovery capabilities. It is used extensively by Turbo-IMAGE software.

Technical information in this document is subject to change without notice.

HEWLETT
PACKARD

Printed in U.S.A. M0588
5951-0677