

.....

**PC&C NAVISION
ON COMPAQ SERVERS
Performance Brief**

COMPAQ
June 1995

NOTICE

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL. THIS INFORMATION IS BRANDED. "AS IS", WITHOUT WARRANTY, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE, TITLE AND AGAINST INFRINGEMENT.

The software described in this guide is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of that agreement.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state, or local requirements. Compaq does not warrant products other than its own strictly as stated in Compaq product warranties.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

© 1995 Compaq Computer Corporation.
All rights reserved. Printed in Denmark.

Compaq, ProLiant, and NetFlex are registered
United States Patent and Trademark Office.

ProSignia VS and ProLiant art trademarks of Compaq Computer Corporation.

PC&C Navision Performance on Compaq Servers

Performance Brief

Final Edition (June 1995)

CONTENTS

PC&C NAVISION ON COMPAQ SERVERS Performance Brief	1
CONTENTS	3
INTRODUCTION	4
PROLIANT AND PROSIGNIA SERVERS.....	5
PURPOSE OF THIS DOCUMENT	5
USE OF THIS DOCUMENT	5
TESTING METHODOLOGY.....	5
BENCHMARK OVERVIEW	6
BENCHMARK SETUP.....	6
THE NAVISION DBMS	7
THE NAVISION BENCHMARK	12
NAVISION BENCHMARK.....	12
NAVISION INDEX.....	13
TUNING EXISTING INSTALLATIONS	15
BENCHMARKS	16
PROSIGNIA VS SCSI	16
SERVER PROCESS FOREGROUND / BACKGROUND	17
CPU / MEMORY SUBSYSTEM.....	18
DISK CONTROLLER	19
RAID 1 VS RAID 5	20
SINGLE CPU VS. SYMMETRIC MULTI-PROCESSING.....	21
DATABASE SIZE IMPACT ON PERFORMANCE	22

INTRODUCTION

The Compaq server line offers the networking industry's leading servers delivering the broadest customer value. Compaq's award-winning servers combine the legendary price:performance leadership with unrivaled innovations in dependability and ease of ownership."

The ProSignia VS is the entry in the Compaq line of servers that delivers true server functionality for the price of a desktop PC, eliminating the need to use desktop PCs as servers.

The Compaq ProSignia 300 is the next step in the server line that delivers true server functionality for the price of a desktop PC. The innovative TriFlex/PCI architecture of the ProSignia 300 uses high-performance processor and I/O technologies to provide users with maximum performance and flexibility. The Compaq ProSignia 300 introduces high-availability features such as Error Correcting Code (ECC) memory to the entry-level server family.

The Compaq ProSignia 500 is the high-performance server that is easy on your budget, with significant headroom for future growth. Built from the ground up as a server, the ProSignia 500 also incorporates TriFlex/PCI architecture and high availability features into the popular ProSignia family of NetWare servers. Integrated SACS disk and network controllers deliver unbeatable performance while providing the best-in-class expandability.

The Compaq ProLiant 1500 is the affordable, mission-critical server. The Compaq ProLiant 1500 has a breakthrough product design that delivers a powerful set of performance and fault management capabilities never seen before on a server in this class. File and print applications are enhanced through features such as Fast-Wide SCSI-2 hard drives and a PCI expansion bus. Database performance is increased with the use of high-performance Pentium processors and optional DualProcessor capability. These features, along with high availability features such as ECC memory and Insight Manager, make the ProLiant 1500 the affordable mission-critical file server.

The next step is to the multiprocessor line with Compaq ProLiant 2000, the high-end server that delivers unmatched system availability in its class. This server delivers more high-availability features and performance scalability for more demanding business-critical applications. And for the ultimate serving of your most critical networks, get the Compaq ProLiant 4000. The Compaq ProLiant 4000 is a new generation superserver that delivers business-critical network capabilities and exceptional performance at an unprecedented price.

PROLIANT AND PROSIGNIA SERVERS

The ProLiant and ProSignia Servers provide user flexibility and increased system performance.

Prosignia VS Prosignia 300	Prosignia 500	ProLiant 1500	ProLiant 2000	ProLiant 4000
True server functionality for the price of a desktop PC.	High-performance server that is easy on your budget, with significant headroom for future growth	Affordable mission-critical server with powerful performance and fault management capabilities.	High-end file and application server for larger business-critical networks	Application server for large departmental / enterprise networks

PURPOSE OF THIS DOCUMENT

The object of this comprehensive test is to evaluate the consequence on performance, when selecting and configuring the different hardware subsystems. It is not the intent to use these numbers to directly compare different hardware vendors performance using Navision, if used, it must be ensured that the tests are performed under the exact same conditions. One of the lessons from the performance test is, that different setup of load, cache etc. can influence the resulting performance index.

USE OF THIS DOCUMENT

The PC&C Navision Performance on Compaq Servers Performance Brief is intended to be used by various audiences including Compaq worldwide field sales personnel, Compaq Authorized Resellers and customers interested in Compaq server performance results.

TESTING METHODOLOGY

The Compaq approach to running benchmark tests is to obtain the best representative results, while ensuring the results can be reproduced. To accomplish this, we set up the following test using the Navision Benchmark Index.

BENCHMARK OVERVIEW

The following benchmark tests were done using a setup described below and PC & C benchmark tool, which gives an absolute number expressed in an index value.

BENCHMARK SETUP

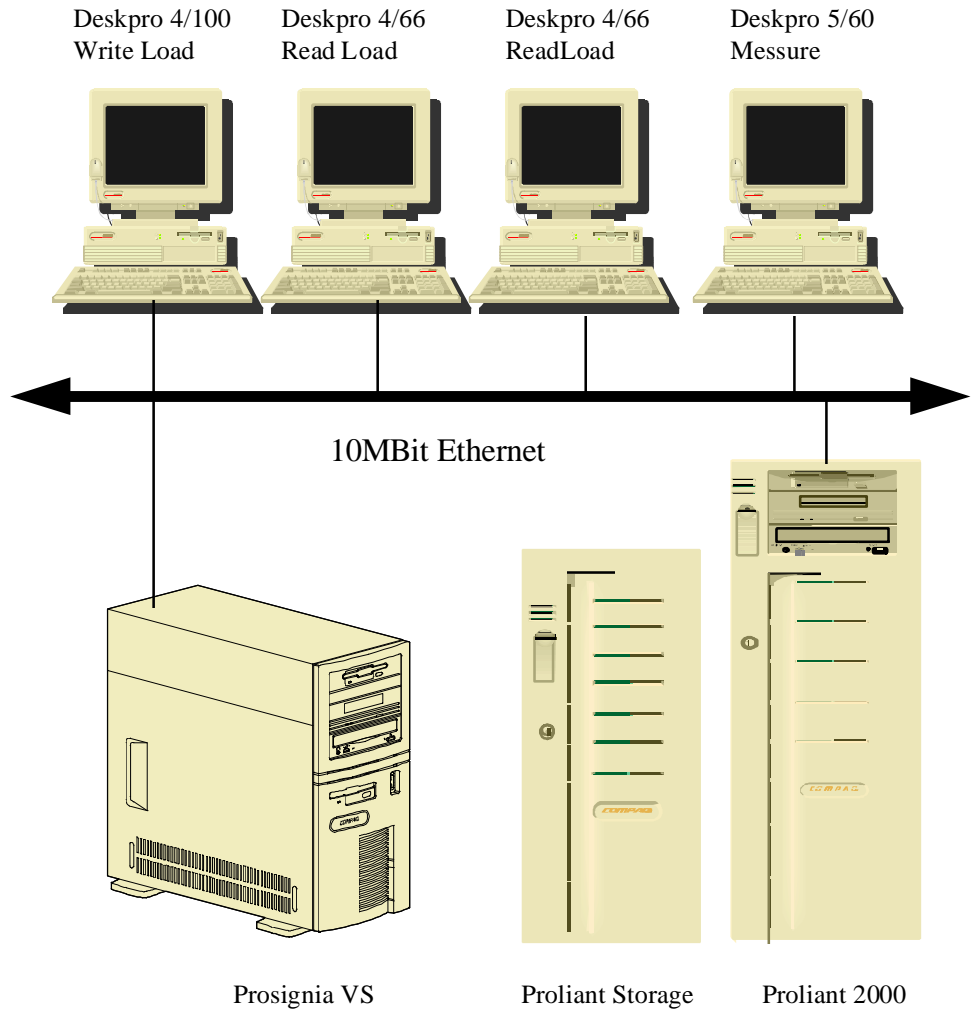


FIG 1. Benchmark hardware setup

The setup used is the standard way PC & C measure performance of the application, with 2 read and one write load, and one station to perform the actual test from. The servers configuration was changed to accommodate the goals of the test, this led to approx 65 - 75 individual test-runs.

The servers was installed using Compaq SmartStart, with OS/2 and LanServer 3.01, the LanServer program was not loaded during the test, only the network support. The parameters (config.sys) and other software settings were not changed from defaults.

The setup include 4 workstations and two test-servers. The systems are configured as follows :

Computer	Function	CPU	RAM	Disk	NIC
Deskpro XE	Read Load	486/66	4 MB	240 MB	TP16
Deskpro 860 CDS	Read Load	486/66	4 MB	240 MB	TP16
Deskpro XE 4/100	Write Load	486/100	4 MB	240 MB	TP16
Deskpro XE 5/60	Measurement	Pentium 60	4 MB	240 MB	TP16
Prosignia VS	Test Object	486/66	32 MB	1-6 1GB	Int. Netflex
Proliant 2000	Test Object	1-2 Pentium66	32 MB	1-12 1GB	Netflex-2

THE NAVISION DBMS

If you are new to the NAVISION database system, this section will introduce you to some of the most important concepts you need to understand in order to work with data in a NAVISION database. The word 'database' is in such common use that we must begin by defining what a database is.

We normally consider a database as a collection of related information, or data. By data we mean some known facts that can be stored and that have an implicit meaning. A typical example is a list which describe the names, addresses and telephone numbers of your customers. These data can be recorded in an indexed address book, or in a disk file using appropriate software. This is a simple example of a collection of related information and hence it is a database.

In this document we will consider a *database* as a collection of related data which is stored in a computer in a way that enables you to retrieve the information you need very quickly.

Physical vs. Logical Database

As a typical database user you are not concerned with where each data item is stored on the hard disk or what size it is; your only concern is that when a reference is made to for example a Name, the correct value is returned.

This is why the NAVISION database system provides you with a conceptual representation of data that does not include to many details of how the data is stored. An abstract data model is used to provide this conceptual representation. This data model uses logical concepts, such as objects, their properties and their relations, which are easier to understand.

This leads us to distinguish between the *Logical* and the *Physical* database. When we speak about the logical database we are only concerned with the structure of the data and the relationship between different bits of information. In this description we do not deal with how these structures and relations are implemented. When we speak about the physical database we only deal with the structure in the logical database and how the search paths between them are implemented.

When we use the term *database* in the following, it should be interpreted as the logical database.

What the user sees as a coherent set of information in the NAVISION database system can be stored on up to 16 physical disk files, but this is transparent to the user.

The Figure illustrates how one logical database physically can be stored on three hard disks d:, e: and f:, but nevertheless they still comprise a single (logical) database.

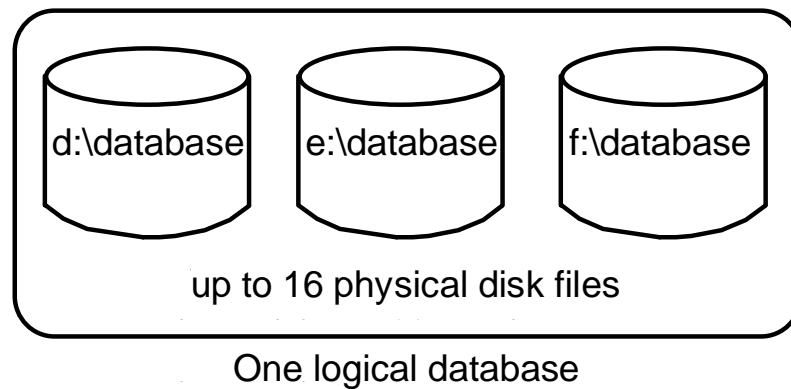


FIG 2. Database parts

The logical database contains *Objects*. These objects can be divided into two main categories:

Object used by a NAVISION application, that is: Batch, Function, Import, Menu, Report and Window objects

Tables

The first category of objects can be seen as basic building blocks providing services needed when we want to insert, modify, delete or show data stored in the other category of objects, that is, the tables. A typical example of the use of a service provided by the first category of objects, is when we use a window object to show a detailed view of the fields in a record. Another typical use of the first category of objects, is when we use a report object to print, for example a survey of a budget.

The DBMS Cache

The DBMS Cache is a buffer storage, which holds copies of portions of the database currently used by the DBMS. As long as the required data is in cache, the data appears as being immediately available. When the required data is not in the cache, it must be fetched from the disk first.

The Cache is transparent to the user/programmer:

Data is fetched into the cache automatically and copied back to the physical disk automatically.

No instruction are required to do these transfers.

The user does not have to know of the cache's existence for the program to work correctly, but the developer has to be aware of the caches existence for the application to work well.

Figure 3 illustrates applications sending requests to the DBMS. When, for example, Application 2 sends a request to read data from the database, the request handler determines if the desired data can be fetched directly from the cache, or if the data must be fetched from a disk.

At the same time another application could be modifying a record in a table in the database. This modification will be written to the DBMS cache, not to the disk. When this application ends the write transaction (commit the changes), the data in the cache modified during the transaction will be written to the disk; the cache is said to be *flushed*.

The DBMS Cache always holds the most recently used data. The cache automatically replaces parts of the cache memory with relevant parts of the database.

The size of the cache has great impact on the performance. Two simple rules apply when determining the size of the cache:

The more memory you assign to your cache, the more efficient the cache will be (Of course there is no reason to assign more memory to the cache than the total size of your database).

The size of the cache should not exceed the amount of physical memory available on your system, as this may cause your operating system to swap the cache memory to disk; this will slow down the overall speed of your NAVISION system tremendously.

Nevertheless, there are upper limits for the DBMS Cache size. The maximum cache size in DOS and OS/2 versions of NAVISION is 6400 KB, while the maximum size in UNIX versions are 32000 KB.

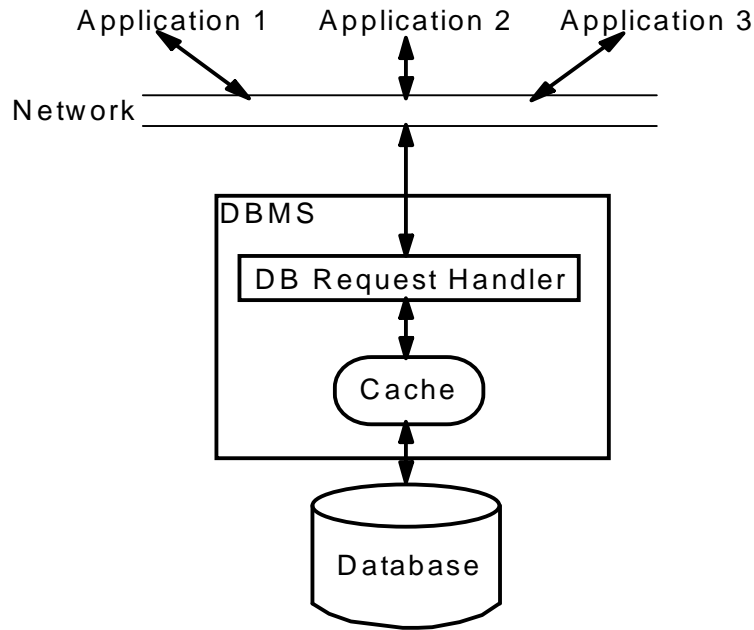


Fig 3. Multibel requests send to DBMS

The Commit Cache

The Commit Cache is a special write buffer storage for the disk(s) in your system. The Commit Cache has been designed to:

Quickly absorb committed transactions from the DBMS. This frees the DBMS for other tasks

Enable asynchronous disk writes

Enable parallel disk writes when using multiple disks

Guarantee that the disk file always is consistent

The Commit Cache is placed between the DBMS and the database, and absorbs committed transactions from the DBMS. When the Commit Cache has received a committed transaction, it takes care of writing the data to the disk(s). In this way the DBMS is free to perform other tasks while data is being written to the disk. The data is said to be written *asynchronously* to the disk, as the time for the disk write is independent of the time when the transaction was committed by the DBMS.

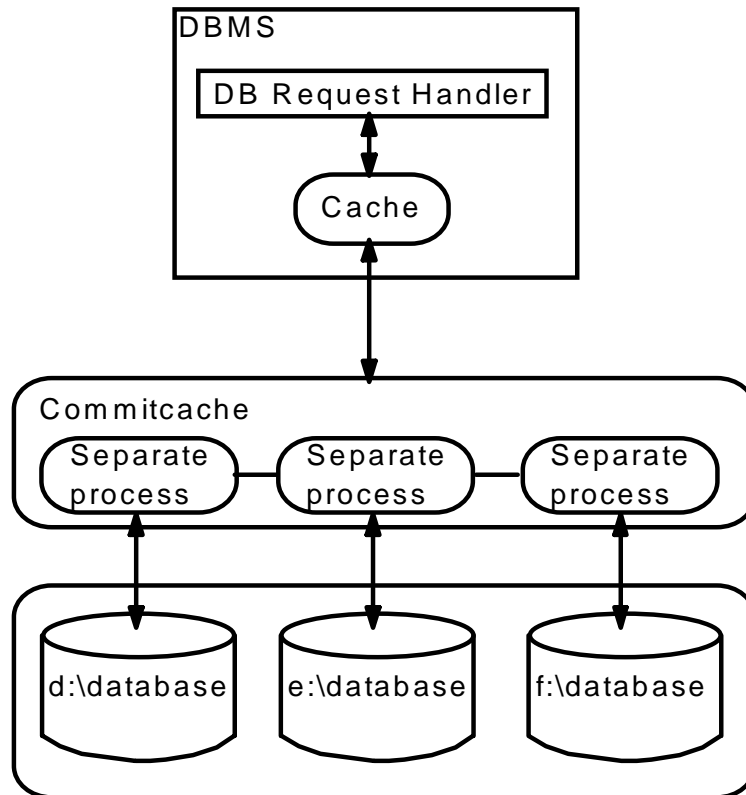


Fig. 4 illustrates a database stored on three physical disks. Each disk is controlled by its own commit cache process. All processes are connected to enable parallel writing.

As described in the section “Physical vs. Logical Database” on page xxx, the logical database can be stored in up to 16 distinct disk files (disks). When using more than one disk, each of these disks are controlled by a separated commit cache process, which are linked to each other in order to enable and control (asynchronous) parallel write operations.

The Commit Cache always guarantees that data is written to the disk in the same sequence as the data is sent to the Commit Cache. This assures that the database file always is consistent. The database file is consistent, even if a power failure should occur during a write operation to the disk.

However, the data which is currently in the Commit Cache memory when a power failure occurs, is lost, and must be re-entered via the NAVISION user interface.

Note: Do not use advanced disk caches with delayed write back (sometimes called lazy write). This may cause your database file to be corrupted. This however, is not the case with the Compaq SMART Controller where the write cache is batterybacked and mirrored.

THE NAVISION BENCHMARK

The NAVISION Index is calculated based on a measurement in a special application. Before starting the measurement the database must be created and initialized. First the database is created with the desired size and number of parts. The NAVISION Index-objects are then entered into the database.

Due to the way the DBMS is working the database must be prepared before the test data is created. When a new database is created (or when switching off the NAVISION server, or importing a full security copy) the free block list will be reset. Meaning that all writing will start on the first database-part then second and so on. To prevent this the free blocks must be mixed by a batch job. (in normal cases will this be done very fast after a server crash (1-2 days depending on the amount of data)). After that can the data be created by a batch job.

Normal size for the testdatabase varies, for some purposes 10 MB is enough. But the normal size is 100 to 500 MB. The time needed for creating the testdata is almost proportional with the size of the database (normally the database are filled 50 % with data).

The following describes the individual items on the NAVISION Index menu.

NAVISION Index			
NAVISION index	Benchmark	Preparation	Records
Reading load	Company data	Mix blocks	Test records
Writing load	Benchmark	Create data	Test series
System test	Network speed	Reinit data	Test setup
Stop load		Mix and create	Mix records
Cancel load			Control load
Results			Design

NAVISION BENCHMARK

Benchmark:

A short measurement based on simple tests between the client and server and some measures of the performance of each part.

Network speed:

The measurement of the network speed.

NAVISION INDEX

Reading load:

Used on a client workstation to generate load, based on readings from the server. This batch job is controlled by the test automatically. More than one read-client can be started at the same time. With the macro "READ.MAC" can the workstation be started automatically with the command:

```
navi2 serverno=1,macro=read
```

Writing load:

Used on a client workstation to generate load, based on writings on the server. This batch job is controlled by the test automatically. More than one write-client can be started at the same time.

With the macro "WRITE.MAC" can the workstation be started automatically with the command:

```
navi2 serverno=1,macro=write
```

System test:

This batch job measures the speed of the NAVISION Index. The index is based on the operations needed for creating and booking an invoice. This Test is run over a time and the NAVISION Index is calculated. The test can also be run with a different setting of the load speed. This is the setting Serie=Yes. When doing this the speed and test time taken from the "Test serie"- file. Before starting the test the "Test setup" window must be entered for initializing some data.

Stop load:

After the batch job "System test" is stopped by "ESQ" can the load-clients be stopped by this batch job.

Cancel load:

When a test serie is ended, and the load-clients are still running, they can be stopped with this batch job. Useful for changing between databases, when the clients have been started with a macro.

Mix blocks:

Before starting a test the free blocks of the database must be mixed. This is done to give a very fast way to get into an unsorted mode when using more than 1 database-part.

All the space in the database is used during this batch job. Sometimes (1 of 10 times) it can fail. When this happens the database size must be increased by 10 Kb, and the batch job restarted with continue=Yes.

Create data:

The testdata can be created with this batch job.

Reinit data:

After running the test for very long time the testdata can be very fragmented. With this batch job they can be restored to the initial situation.

Mix and create:

The batch job for mixing the free blocks and creating data is combined in one. Then you can start it and leave it. (Error situation see "Mix blocks")

TUNING EXISTING INSTALLATIONS

To speed up the performance in an existing installation by splitting the database in two or more parts the following procedure must be used. The special procedure is used due to the way the NAVISION DBMS is working internally.

If a full security copy is entered in a new database, the list of all the free blocks will be reset, and all the data will be placed on the first disks, therefor not using the new disks that was added.

When a security copy of only one company is imported into the database the free block list will be maintained. By doing this with all the companies and mixing the free blocks before that, it is possible to place all data random on the disks. All the necessary steps needed, is described in the procedure below.

First make a copy of the database. To be sure not to destroy vital data, when working with the database and deleting the companies.

In the *copy*, make a security-copy of all the companies(one company in each security-copy).

In the *copy* delete all companies.

In the *copy* make a security-copy of the whole database (after deleting all companies the database will only contain common files and objects).

Create the new database with the desired amount of databaseparts.

Import the security-copy of the whole database. (the security containing only common files and objects).

Import the batch job "BENCH, Mix blocks". This batchjob is a part of the "NAVISION Index" Utility.

Use the batch job "BENCH, Mix blocks" to get all the free blocks mixed.

Import the security copy of all the companies one at a time. If all the companies where imported directly will the free block list be reset and the work done with mixing the blocks was not used.

Now the new database should be ready.

In case of a server crash the free blocks list must be initialised, because the performance will be lower the first days until the free block list is rebuild as random. After extending the database the free blocks must be mixed, because it will be the same situation as after a server crash.

BENCHMARKS

PROSIGNIA VS SCSI

This 486/66 system is our baseline, it is equipped from 1 to 6 Fast SCSI-2 1.05 GB disks. This setup is one used in many of the Navision installations, and it documents the increase in performance that can be obtained by splitting the database over many disks, going from 1 to 6 disks gives an **increased performance of 215%**.

The curve shows how fine the performance scale is, if the database is split over multiple disks. From 1 to 2 disks 100%, 2 to 3 disks 52%, 3 to 4 disks 30.6%, 4 to 5 disks 21% and 10.8% on the last 5 to 6 disks.

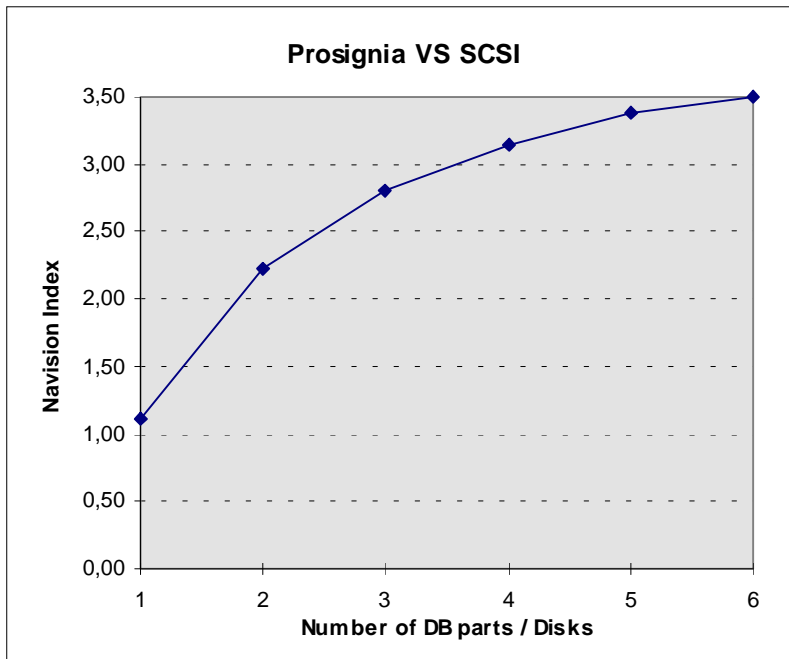


Fig. 5 Prosignia VS SCSI , baseline.

SERVER PROCESS FOREGROUND / BACKGROUND

When the server process is in foreground it gets a better priority than other programs/processes on the system, therefore it performs slightly better when using one database part. Using more databases, Navision starts up one individual process per database, and they get starved for time on the CPU, so we see a lower performance when adding databases / disks.

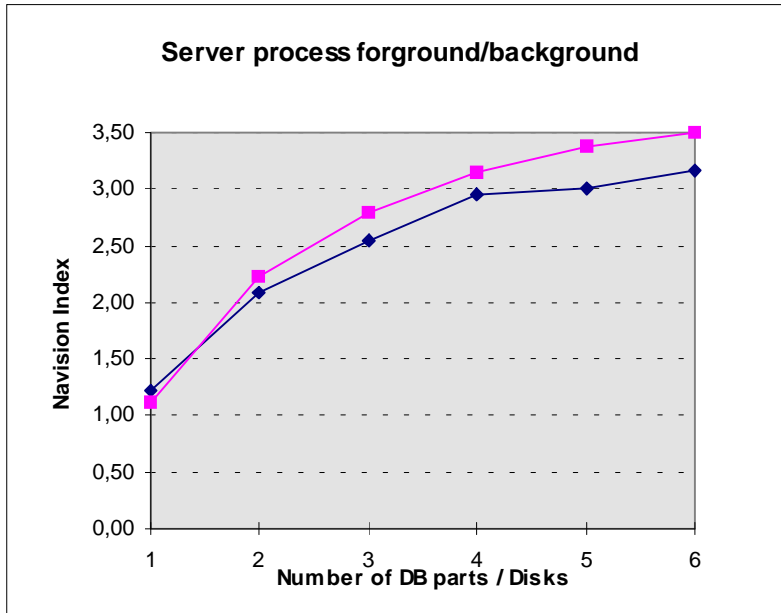


Fig 6. ◆ Server foreground, ■ Server background

CPU / MEMORY SUBSYSTEM

Prosignia VS Fast SCSI-2 vs. ProLiant 2000 Fast SCSI-2

On this test we demonstrate the difference the CPU / Memory system has on performance, when the same disks and disk controllers are used. The only difference is the Prosignias 486/66 / 32bit memory path contra the ProLiant Pentium66 / TriFlex architecture. The Pentium/TriFlex architecture clearly demonstrate its capacity, over the standard CPU / Memory bus implementation by a 62% better performance, and it also expresses itself in its scaling of 305% vs 204% on five disks.

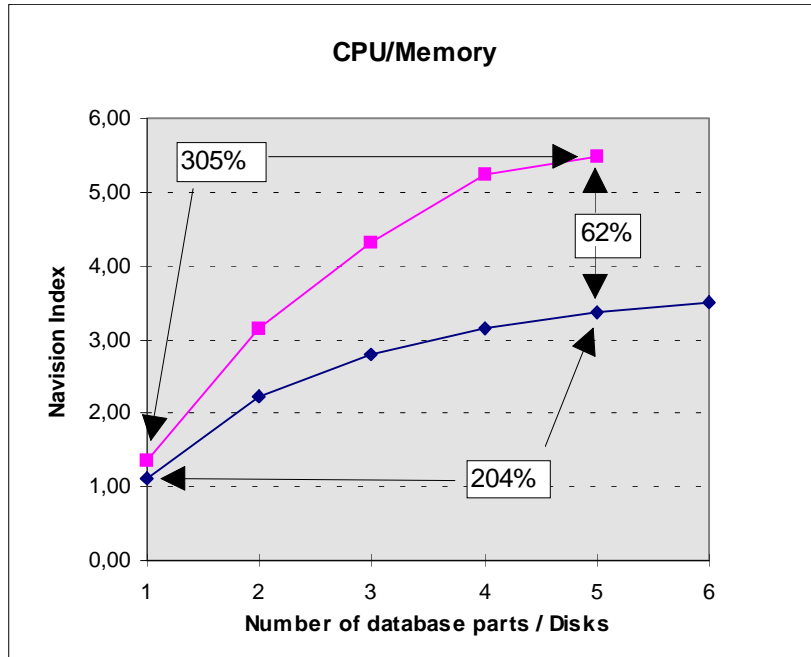


Fig 7. ◆ Prosignia, ■ Proliant

DISK CONTROLLER

Compaq SMART RAID vs. Compaq SCSI-2

In this test we use 6 disks connected to Compaq Fast SCSI-2 controller and compare it to 12 disks connected to Compaq SMART controller, configured as 6 RAID 1 (mirror) drives.

The high performance of the Compaq SMART array controller is evident here in this picture, a small overhead, although it has to write the data twice, The SMART Controller protects against drive failures, and can also use spare drives, that will automatically replace a failed drive without user intervention and will maintain a high data integrity.

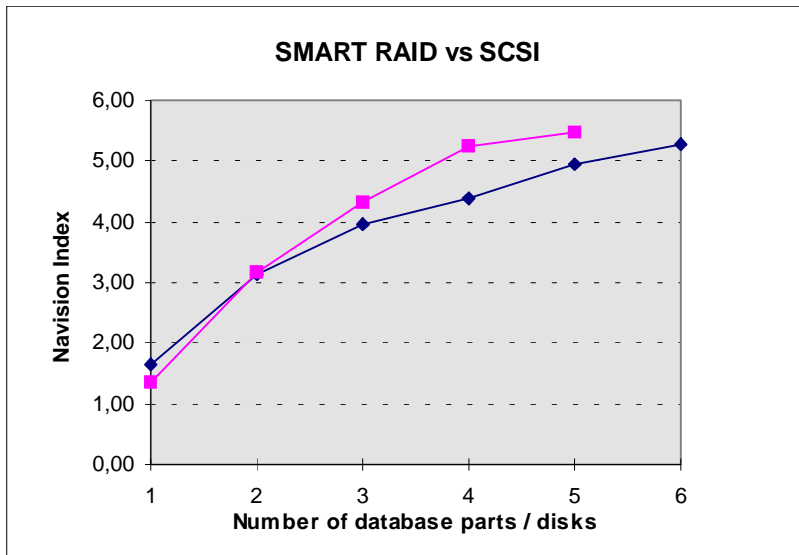


Fig 8. ◆ RAID 1, ■ Fast SCSI-2

RAID 1 VS RAID 5

This is an interesting measurement, the RAID 5 is using 10 disks, and we tried to split the database up in 2,4 and 6 parts on the same drive, because we knew that Navision disk-process should be able to issue multiple request to the single drive this way. But that did not increase the performance, although the controller is capable of handling multiple access, as shown when running the drives as RAID 1's.

So the conclusion is, if we have i.e. 10 drives they should be configured as 5 pairs of mirrored drives, *not* a single RAID 5 drive.

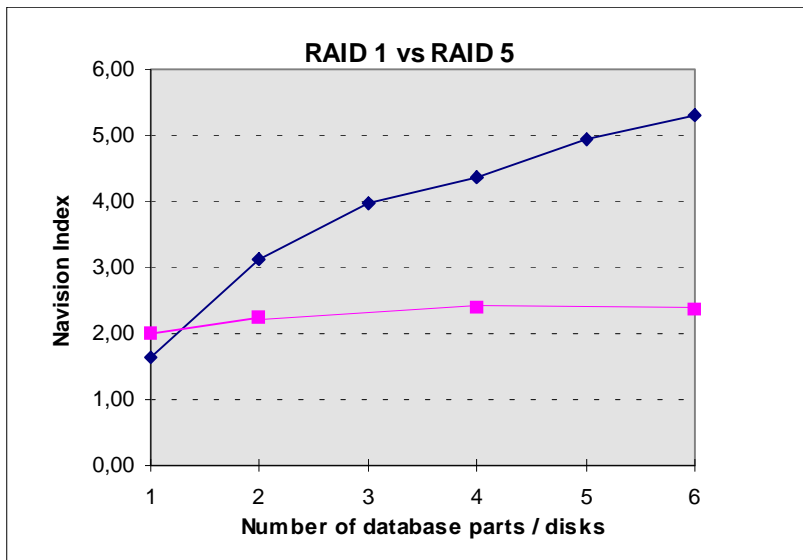


Fig 9. ◆ RAID 1, ■ RAID 5

SINGLE CPU VS. SYMMETRIC MULTI-PROCESSING

The main process of Navision cannot be scaled across multiple CPU's , so the small overhead of multiprocessing is slowing the server down a bit. This test was done on the Proliant 2000 system with RAID 1 disk configuration.

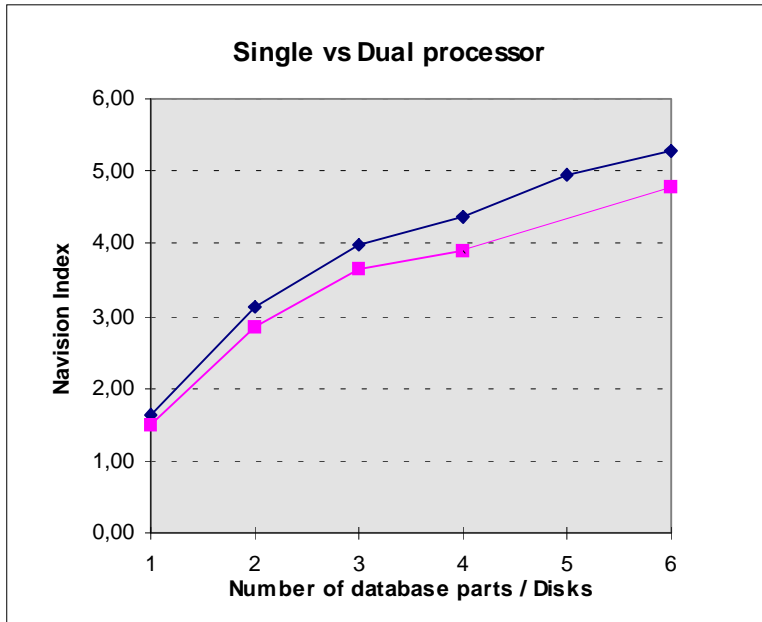


Fig 10. ◆ 1 CPU, ■ 2 CPU

DATABASE SIZE IMPACT ON PERFORMANCE

This test was done on Proliant 2000 RAID 1 with 5 database parts. Only changing the size of the database from 100 MB, to 500MB and 2.5GB all with 50% data.

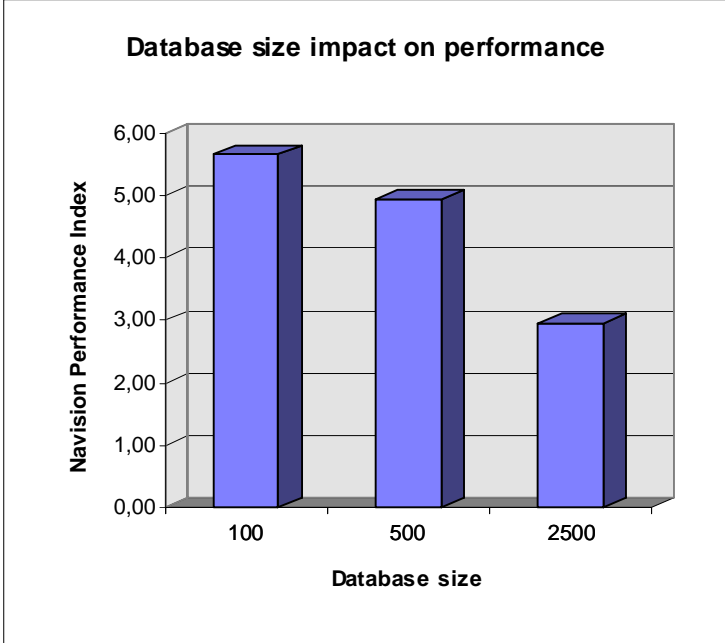


Fig 11.