Performance predictions for the upgrading of a VAX-cluster

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science

Memorandum COSOR 88-36

PERFORMANCE PREDICTIONS FOR THE UPGRADING OF A VAX-CLUSTER

J. van der Wal, R.J. Wijbrands,
A.P. de Grient Dreux, J. Hoogendoorn, R.C. Marcelis

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands

Eindhoven, December 1988
The Netherlands

PERFORMANCE PREDICTIONS FOR THE UPGRADING OF A VAX-CLUSTER

J. van der Wal, R.J. Wijbrands,
A.P. de Grient Dreux, J. Hoogendoorn, R.C. Marcelis
University of Technology, Eindhoven

ABSTRACT

During the last two decades many interesting and useful results have been obtained in the area of queueing networks. It has been shown that the queueing network model is a powerful tool in computer performance analysis.
In this paper we report on some of the difficulties we met in a performance study for the upgrading of the VAX-cluster at the Eindhoven University of Technology. Our conclusion has to be that there are sufficiently many queueing network models and techniques for analyzing them, but that for accurate performance predictions the behaviour of memory contention is not well understood.

1. INTRODUCTION.

At the Eindhoven University of Technology (EUT) an attempt has been made to model and analyse the performance of the local VAX/VMS (or shortly VAX) cluster, consisting of three VAXes sharing background memory. The result is a computer package called VAMP (VAX Analysis and Measurement Package), consisting of the following 5 programs (cf. [2,3]):

1. A program that collects measurements on the system behaviour at intervals of 3 minutes, based on MONITOR, a DEC monitoring program.
2. A program that compresses these data at the end of each day.
3. A program that translates these data into parameters for a queueing network algorithm.
4. A mean-value type algorithm to calculate the performance characteristics.
5. An interface that enables the user to create other VAX clusters and make performance predictions for them.

When the project started the EUT cluster consisted of three VAX-11/750 computers each having 6Mb of main memory. As background memory there were five RA-81 disks with a total access time of 38ms (seek 28ms, latency 8.3ms, transfertime 1.3ms) and two slower RA-60 disks with a total access time of 52ms.

In section 2 we briefly discuss the modelling of the cluster and some aspects which complicated the design of VAMP.

In October 1987 the configuration changed. One of the three VAXes was replaced by a VAX-8530, with a processor speed 5 times as fast as the VAX-11/750. The main memory of the new VAX became 16Mb. Two RA-81 disks were added and there was some reorganization of the disk workload.

Section 3 describes the attempt to use the model to estimate the change in performance. Unfortunately the results are very disappointing. The reason for this failure is discussed in section 4.

2. THE VAX CLUSTER MODELLING.

The actual configuration can be modelled as shown in Figure 1.

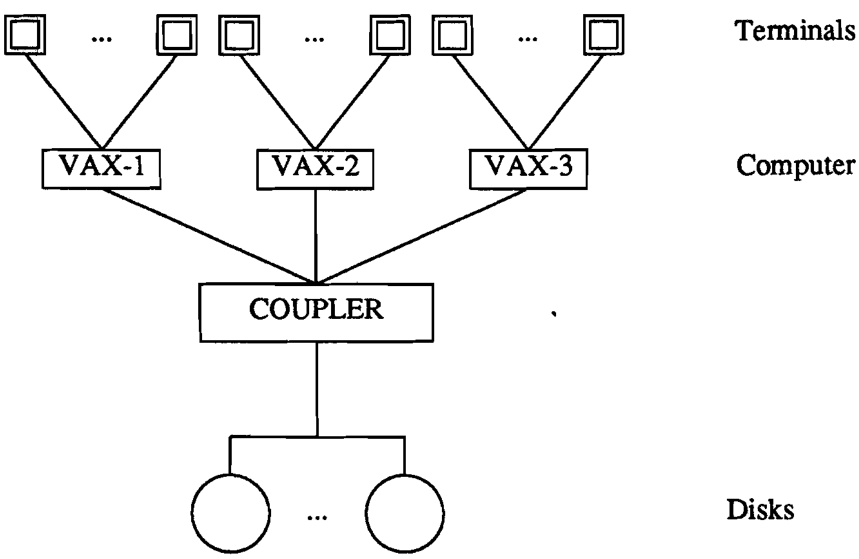

Figure 1: VAX/VMS-cluser at the Eindhoven University of Technology

In the VAX Operating System three classes of processes can be distinguished:

1. Interactive Processes. Programs which every now and then need input from the terminal.
2. Batch Processes. These are programs that run automatically without needing additional input from a terminal.
3. System Processes. These are processes that are created by the other two classes in order to perform a certain task. Once this task has been completed, the system process disappears again.

It seems necessary to distinguish these three classes since the CPUs do so. The system processes receive the highest priority, then come the interactive processes and finally the batch processes. Since system processes are only active when called upon by a user process, we have decided to divide the workload of the system processes over the other two classes according to their CPU utilization. As a result we have to distinguish only six types of processes, interactive and batch for each CPU.

The Round Robin disciplin with time slices of 200ms which is used by the CPUs is modelled as Processor Sharing.

The measurement program based on Monitor produces every 3 minutes a photograph with the total CPU time in tenths of a second and the total number of I/Os since its birth for each process. By comparing two consecutive photographs it can be calculated how much CPU time is used and how many I/Os have been executed for a process in the three minute period.

Further these photographs display the status of each process: is it ready to be processed, waiting for terminal input, waiting for an I/O to be completed, etc.

From these data it is not possible to calculate data like the average think time at the terminal for an interactive user, or the probability that after a CPU visit terminal input is needed. Therefore the model only uses the socalled relative workloads, i.e. the fractions of the time a process spends thinking at the terminal, calculating at the CPU and performing an I/O at a disk.

Also Monitor does not display per process on which disk the I/Os have been performed. It only presents these quantities per CPU. Therefore the assumption has to be made that all processes on a specific CPU have the same behaviour, i.e., the probability that an I/O concerns disk i is the same for all processes.

In order to obtain more accurate data from the measurements, it was decided that it was better to count only the "active" processes, and not all interactive processes. An active process is a process which during the last 3 minute interval used at least one tenth of a second CPU time (the smallest amount Monitor measures) or has performed at least one I/O. This eliminates "sleeping" processes.

From the measurements we obtain parameters for each number of active processes. In particular the measurements show how the number of I/Os increases if the number of active processes is increased. So for the memory contention we use measurements instead of some kind of model. Here a problem appears, how can we predict what will happen if the present configuration changes without having a good idea about the memory consequences. We will come back to this issue in sections 3 and 4.

The measurements are translated into input parameters for a closed queueing network model with three CPUs with processor sharing service disciplin, with six classes of jobs, active and batch for each CPU, a number of disks and a terminal station with infinite server disciplin.

The network model is analysed using an approximate mean-value algorithm, a first order depth improvement of Schweitzer's algorithm (cf. [1,4,5]). Since at the disks the workloads are definitely not exponentially distributed, there the Pollaczek-Khintchin formula for the $M \mid G \mid 1$ queue with a coefficient of variation for the service time of roughly .6 was implemented (cf. [7]).

Measurements have been collected during several months, and as a result the measurements are fairly accurate. Since also the model is very close to the actual situation, the results from the mean-value algorithm must be good as well.

One of the most informative results is the relative response time, i.e., the response time per CPU second depending on the number of active processes. Seen as a function of the number of active processes the relative response time indicates how one approaches saturation. Also the results from the model show which part of the system will become the bottleneck and what results can be expected from an extra disk or a different distribution of data over the disks.

In this case it became clear that the System Disk was quite heavily used, approximately 50% of all disk visits were to this disk.

3. THE PERFORMANCE PREDICTION FOR THE NEW CLUSTER

As said before, in October 1987 the configuration was changed. One of the three VAX 11/750s (called VAX-3 in the cluster) was replaced by a VAX-8530 which is approximately 5 times faster than the 11/750. The new VAX received a 16Mb main memory. Also two RA-81 disks were added and the disk contents were reorganized in particular with respect to the system disk.

The question is, can we make a sufficiently accurate performance prediction for the new cluster based on the measurements collected for the old one. We thought we could.

Concerning the memory contention on the new VAX, we made the assumption that i jobs on VAX-8530 would lead to the same paging behaviour as 6/16 times i jobs on the VAX/11-750. The ratio 6/16 corresponds to the ratio in main memory, 6Mb on the 11/750 and 16Mb on the 8530. This seems to be a reasonable approximation. The paging behaviour for the 11/750 was known from the measurements.

Another problem was that the system manager decided that VAXes 1 and 2 would be dedicated more strictly than before to special groups of users. Since it was unclear what this change would lead to, we ignored it.

A third problem is that a change in user behaviour can be expected. If the processor speed is 5 times as high as before, a number of users will increase the CPU load of their programs. For instance, a user might decide to extend his simulation from 20000 to 100000 events.

Finally it was clear that in a couple of months a serious number of users would migrate from the Burroughs computer (for which the support was to be terminated in the summer of 1988) to the VAX-8530.

We decided not to estimate all these changes as we had unsufficient information to do so. We made performance predictions assuming it was known how many active users there would be in the new cluster. The result of the prediction would then be compared with measurements in the first weeks after the change. The results of this prediction and the results obtained using the measurements are displayed in Table 1 for the average number of active processes in these weeks.

	number of active processes	Measured		Predicated	
		CPU utilization	relative response time	CPU utilization	relative response time
VAX-1	3	0.44	1.73	0.32	1.87
VAX-2	3	0.28	1.80	0.39	1.86
VAX-3	13	0.49	2.90	0.50	5.19

Table 1: Measured versus predicted performance

As we see the results for VAXes 1 and 2 are quite reasonable, the average CPU utilization is predicted well, but the split over the two is not very good. The response times per CPU second are quite good. For the new and faster CPU the prediction for the CPU utilization is good as well, but the response time ratio is completely wrong.

4. ANALYSIS OF THE FAILURE.

So unfortunately there is one big error in the predictions, the response time ratio for VAX-3. Since the CPU utilization is predicted well, the error must lie in the disk workload. Comparing the results for the disk utilization, we see that indeed in reality much less disk I/Os are needed than is assumed in the model used for the prediction.

An explanation one might think of is, that sharing 16Mb with 16 users is easier than sharing 6Mb with 6 users. Since in practise the number of free pages in main memory is usually high, it is unlikely that this explains more than a very small part of the error.

It took us a long time to come up with another, much better explanation. Therefore we had to go back to the way the VAX/VMS Operating System controls the number of pages each process receives. When a process starts, it receives an initial number of pages in main memory. If it turns out that this number is insufficient, the process receives an additional amount. The criterion for the number of pages being insufficient is that the number of page faults per CPU second exceeds some system parameter called PFRATH (Page Fault RATe High). This process continues until some maximum for the number of pages in main memory is reached (cf. [6]). The value for PFRATH had been set on 12 in the old configuration, and it was also set on 12 for the new, 5 times faster, VAX.

This explains a lot. Since the new VAX is 5 times faster, 12 page faults per second for the new VAX corresponds to only 2.4 page faults per second on the old one. So on the new VAX a process much easier receives additional pages, and as a result the number of disk I/Os will be less.

It is not possible to verify whether this is the full explanation, since this would mean that the system manager would have to set the system parameter PFRATH to 60 on the new VAX, while it is almost certain that this would lead to a serious decrease in the performance. Together with the system manager we decided to follow another way to check our assumption about the importance of the value of PFRATH. Instead of increasing the value for VAX-3 the parameter value was decreased for VAX-1. This experiment showed that the I/O rate, and hence the performance, are indeed very sensitive to the value of this system parameter.

5. CONCLUSION

In this paper we presented our experiences with a performance study for the upgrading of the VAX/VMS cluster at the Eindhoven University of Technology.

Our main conclusion is that it is absolutely vital to understand how sensitive the performance of the system, in particular the memory contention, is to the various system parameters, such as in this case the parameter PFRATH.

Another problem we ran into is that it is unclear how users of the system (in this case researchers and students) will react on the performance improvement. In professional organizations in banking and industry these changes are more predictable.

REFERENCES

[1] J.B.M. van Doremalen, J. Wessels and R.J. Wijbrands, Approximate analysis of priority queueing networks. In O.J. Boxma, J.W. Cohen, H.C. Tijms (Eds.), Teletraffic Analysis and Computer Performance Evaluation (North-Holland, Amsterdam), 117-131, 1986.

[2] A.P. de Grient Dreux, Performance Onderzoek naar het TUE VAX-cluster systeem, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1987 (Master thesis, in Dutch)

[3] J. Hoogendoorn, Towards a DSS for performance evaluation of VAX/VMS-clusters, Memorandum COSOR 88-22, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1988.

[4] M. Reiser and S.S. Lavenberg, Mean-Value Analysis of Closed Queueing Networks, J.A.C.M. 27 (1980), 313-322.

[5] P.J. Schweitzer, Approximate Analysis of Multiclass Closed Networks of Queues, Lecture presented at The International Conference on Stochastic Control and Optimization, Amsterdam, 1979.

[6] VAX Software Handbook (Digital Equipment Coorporation, USA), 1982.

[7] R.J. Wijbrands, Queueing Network Models and Performance Analysis of Computer Systems, Doctoral Dissertation, Eindhoven University of Technology, 1988.

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science
PROBABILITY THEORY, STATISTICS, OPERATIONS RESEARCH AND SYSTEMS THEORY
P.O. Box 513
5600 MB Eindhoven - The Netherlands
Secretariate: Dommelbuilding 0.03
Telephone: 040 - 47 3130

List of COSOR-memoranda - 1988

Number	Month	Author	Title
M 88-01	January	F.W. Steutel, B.G. Hansen	Haight's distribution and busy periods.
M 88-02	January	J. ten Vregelaar	On estimating the parameters of a dynamics model from noisy input and output measurement.
M 88-03	January	B.G. Hansen, E. Willekens	The generalized logarithmic series distribution.
M 88-04	January	J. van Geldrop, C. Withagen	A general equilibrium model of international trade with exhaustible natural resource commodities.
M 88-05	February	A.H.W. Geerts	A note on "Families of linear-quadratic problems": continuity properties.
M 88-06	February	Siquan, Zhu	A continuity property of a parametric projection and an iterative process for solving linear variational inequalities.
M 88-07	February	J. Beirlant, E.K.E. Willekens	Rapid variation with remainder and rates of convergence.
M 88-08	April	Jan v. Doremalen, J. Wessels	A recursive aggregation-disaggregation method to approximate large-scale closed queuing networks with multiple job types.

Number	Month	Author	Title
M 88-09	April	J. Hoogendoorn, R.C. Marcelis, A.P. de Grient Dreux, J. v.d. Wal, R.J. Wijbrands	The Vax/VMS Analysis and measurement packet (VAMP): a case study.
M 88-10	April	E. Omey, E. Willekens	Abelian and Tauberian theorems for the Laplace transform of functions in several variables.
M 88-11	April	E. Willekens, S.I. Resnick	Quantifying closeness of distributions of sums and maxima when tails are fat.
M 88-12	May	E.E.M. v. Berkum	Exact paired comparison designs for quadratic models.
M 88-13	May	J. ten Vregelaar	Parameter estimation from noisy observations of inputs and outputs.
M 88-14	May	L. Frijters, T. de Kok, J. Wessels	Lot-sizing and flow production in an MRP-environment.
M 88-15	June	J.M. Soethoudt, H.L. Trentelman	The regular indefinite linear quadratic problem with linear endpoint constraints.
M 88-16	July	J.C. Engwerda	Stabilizability and detectability of discrete-time time-varying systems.
M 88-17	August	A.H.W. Geerts	Continuity properties of one-parameter families of linear-quadratic problems without stability.
M 88-18	September	W.E.J.M. Bens	Design and implementation of a push-pull algorithm for manpower planning.
M 88-19	September	A.J.M. Driessens	Ontwikkeling van een informatie systeem voor het werken met Markov-modellen.
M 88-20	September	W.Z. Venema	Automatic generation of standard operations on data structures.

Number	Month	Author	Title
M 88-21	October	A. Dekkers E. Aarts	Global optimization and simulated annealing.
M 88-22	October	J. Hoogendoorn	Towards a DSS for performance evaluation of VAX/VMS-clusters.
M 88-23	October	R. de Veth	PET, a performance evaluation tool for flexible modeling and analysis of computer systems.
M 88-24	October	J. Thiemann	Stopping a peat-moor fire.
M 88-25	October	H.L. Trentelman J.M. Soethoudt	Convergence properties of indefinite linear quadratic problems with receding horizon.
M 88-26	October	J. van Geldrop Shou Jilin C. Withagen	Existence of general equilibria in economies with natural enhaustible resources and an infinite horizon.
M 88-27	October	A. Geerts M. Hautus	On the output-stabilizable subspace.
M 88-28	October	C. Withagen	Topics in resource economics.
M 88-29	October	P. Schuur	The cellular approach: a new method to speed up simulated annealing for macro placement.
M 88-30	November	W.H.M. Zijm	The use of mathematical methods in production management.
M 88-31	November	Ton Geerts	The Algebraic Riccati Equation and singular optimal control.
M 88-32	November	F.W. Steutel	Counterexamples to Robertson's conjecture.
M 88-33	December	N.P. Dellaert	Multi-item production control for production to order.
M 88-34	December	S.Q. Zhu M.L.J. Hautus	Some remarks on the gap metric.

Number	Month	Author	Title
M 88-35	December	I. Adan J. Wessels W.H.M. Zijm	Queuing analysis in a flexible assembly system with a jobdependent parallell structure
M 88-36	December	J. v.d. Wal R.J. Wijbrands A.P. de Grient Dreux J. Hoogendoorn R.C. Marcelis	Performance predictions for the upgrading of a VAX-cluster.