THE UNIVERSITY OF WARWICK

Original citation:
Kerbyson, D. J., Harper, J. S., Craig, A. and Nudd, G. R. (1996) PACE : A toolset to investigate and predict performance in parallel systems. In: European Parallel Tools Meeting, ONERA, Paris

Permanent WRAP url:
http://wrap.warwick.ac.uk/60980

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

A note on versions:
The version presented in WRAP is the published version or, version of record, and may be cited as it appears here.For more information, please contact the WRAP Team at: publications@warwick.ac.uk

warwickpublicationswrap

highlight your research

http://wrap.warwick.ac.uk/

PACE: A Toolset to Investigate and Predict Performance in Parallel Systems

D.J. Kerbyson, J. Harper, A. Craig, G.R. Nudd

High Performance Systems Group,
University of Warwick, Coventry, CV4 7AL U.K.
email: {djke, grn}@dcs.warwick.ac.uk

Abstract

A toolset, PACE (PerforAnCE analysis environment) for predicting and analysing performance issues in parallel systems is presented here. In this toolset expert knowledge about performance evaluation techniques is not required as a prerequisite for its use. Instead, an intuitive approach is taken by describing the application under study in way that is accessible to the user. A modular approach is taken which permits separate descriptions of sub-tasks, parallelisation strategies, and platform descriptions to take place. PACE has been used to predict performance for number of applications on a number of different parallel platforms. Results to date have indicated that the predictions typically within 5-7% of actual measurements.

1. Introduction

A number of performance tools have been developed to assist the system developer, the application programmer, and the tuning expert to select the most efficient use of hardware, and parallelisation strategy [1-5]. However, many of the performance tools require a user to hold knowledge of the performance related issues. The purpose of the PACE toolset (PerformAnCE analysis environment) is to provide a performance prediction capability in which a user may focus his/her effort on aspects of a performance study that do not require performance related speciality.

PACE utilises an underlying characterisation approach whose objective, using individual parallel system characteristics, is to provide analysis of application performance in a predictive sense without the need to directly utilise the target platform(s). Performance analysis and prediction can be carried out resulting in accuracies of 90% or greater [6]. The development of this approach, its analysis, and its application has been undertaken at the University of Warwick, partly within the ESPRIT PEPS project [7,8].

The characterisation of a computer system aims to provide an analytic means by which to describe the operational complexity using information about the target platform, and the application. Such characteristics can provide information concerning resource utilisation and performance of the entire system as well predicting possible hotspots or bottlenecks under various workloads before they occur in practice.

Characterisation is aimed at end-users whose primary background is in the development and implementation of application codes, and it does not require specialist information regarding computational systems, parallelisation, or performance related knowledge. Nevertheless full performance data can be made available within PACE including:

- Elapsed Time: the predicted time taken in performing the application given a set of application and system parameters.
- Scalability: how an application's performance changes by increasing problem parameters, and/or system size (processors).

- Resource Usage: the identification of the components of the system being used, and those which result in performance bottlenecks [9].

- Sizing: the identification of a problem size which can be handled given the constraints on processing time and system resources.

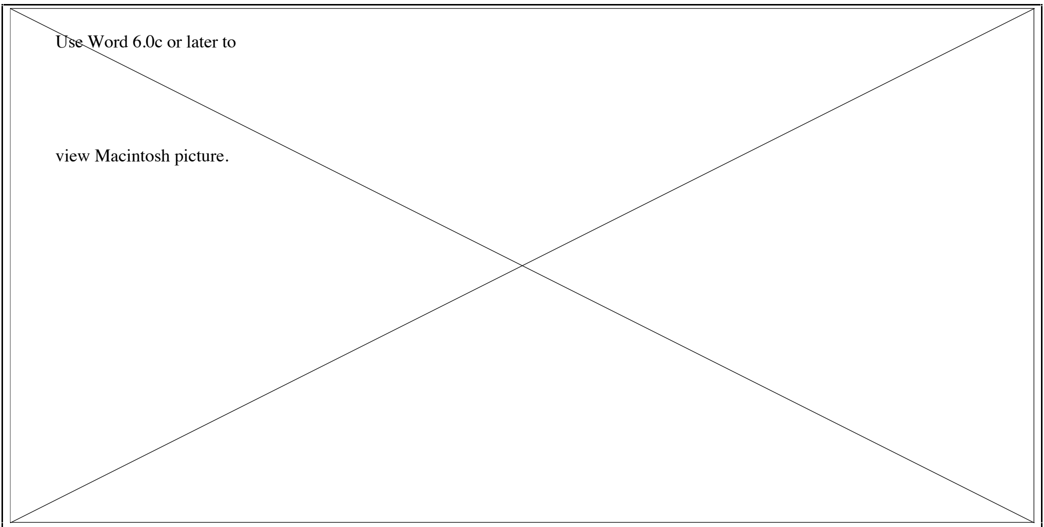

Figure 1 - Concept of the Performance prediction system.

The characterisation approach gives access to this performance on high performance systems prior to the purchase of a system allowing for cross-platform comparisons without costly implementations. Characterisation may also be used for performance tuning, to explore parallelisation alternatives, and to aid the implementation procedure.

2. The Warwick Approach

The characterisation developed by Warwick uses a *layered* approach. It forms a framework consisting of three separate layers, an application layer, a hardware layer, and a parallel template layer as shown in Figure 1.

The layered approach separates out the hardware and software systems through the use of a parallelisation template. This modular approach leads to readily re-useable models which can be interchanged in experimentation. For instance the performance predictions across parallelisation techniques can be compared for a particular application on a particular hardware. The layers used are:

- an application layer, which describes an application in terms of a sequence of sub-tasks each representing a self-contained part of the program (e.g. a matrix operation). These are described using a software execution graph notation containing procedural and resource usage information.
- a parallel template layer, that describes the parallelisation to be used by the application through a task graph notation. This identifies the computations being performed along with necessary communications when the whole task is to be partitioned for parallel execution.
- a hardware layer, containing characteristic performance models for main system components including: inter-processor communication, Input/Output, synchronisation, and computation. Each model is specific to a given hardware platform.

The information required, from an application developers point of view, consists of descriptions of the application and the parallelisation using the first two layers in the framework. To aid the user interaction, the main components of PACE are accessible through a prototype graphical user interface implemented in JAVA.

3. PACE Toolset Components

The PACE toolset includes a range of components that assists a user to create models, visualise results, use pre-defined models from a library, and use information derived from existing application codes. The components of PACE are shown in Figure 2, and are described in more detail below.

- *Evaluation Engine*: Evaluates the current performance model, producing predictions of time, scaling, and resource usage.
- *Workbench*: provides a user-friendly interface to the components of PACE.
- *Source Code Analyser* assists the user in converting sequential source code into the CHIP^{3}S performance language. The user directs this operation by specifying which code are associated with which sub-task elements. Currently this component enables C source to be input, using both parsing and profiling information.
- *Object browser*: assists the user to scan pre-defined model libraries of application kernels, parallelisation strategies (parallel templates), and hardware models. The user may also define new library models.
- *Object Editor*: assists the user to enter and review individual objects contained within the performance model.
- *Parametric visualisation*: enables application and/or system parameters to be varied, and provides a means in which the results can be visualised. Currently supports single and dual parameter manipulation.
- *Trace visualisation*: enables the visualisation of a single prediction scenario. It provides time-space diagrams illustrating computation, communication and idle stages of processors. Currently, this analysis is provided by a trace data file link to the ParaGraph [1] parallel monitoring system.

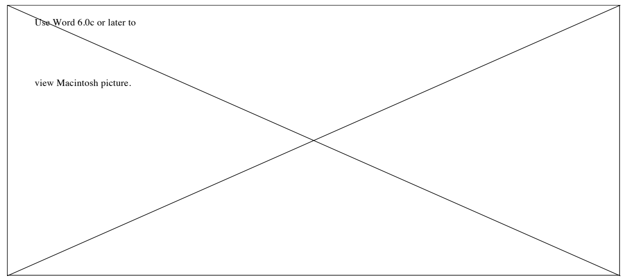

Figure 2 - Overview of the PACE Toolset

The evaluation and performance analysis using characterisation within PACE is performed by the compilation of the underlying CHIP^{3}S performance language into an executable (via a C program representation). This enables a single performance prediction (given a set of application / system parameter values) to be obtained quickly.

The necessary description required from an application can be provided through the use of

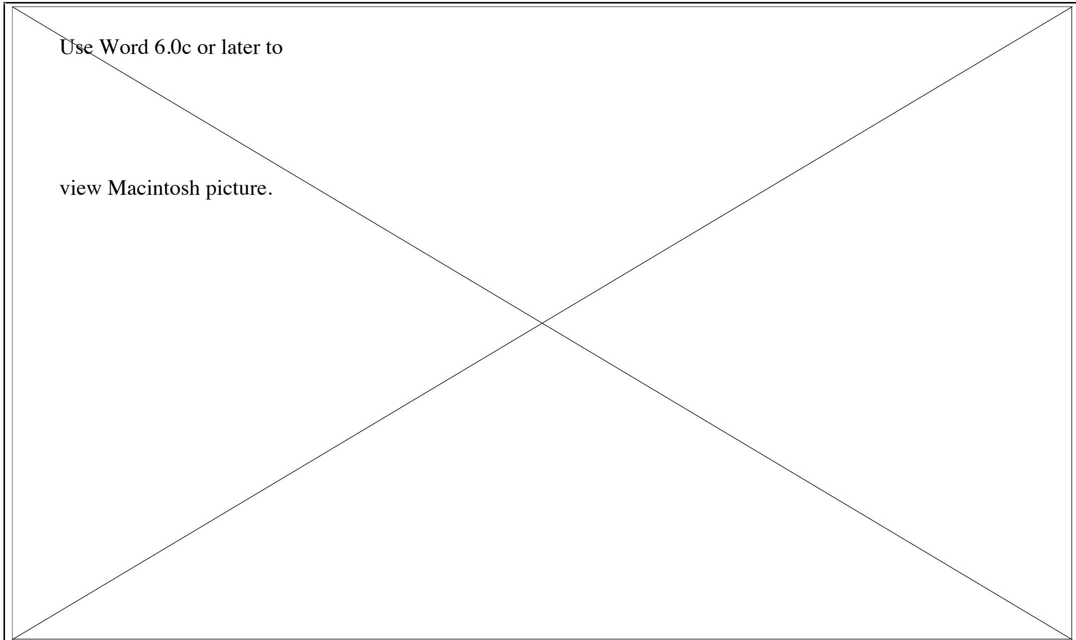

Figure 3 - Orgainisation of the performance objects for an application within PACE.

Software Execution Graphs (control flow), and in terms of resources which can be specified through either component timing (specific to a platform), or Language characterisation (can be used for cross-platform comparisons). The automation of describing the resources directly from application source code is available in the source code analyser, and currently handles sequential C source code. Further manipulation of this output is required to allow the performance session to take place.

4. PACE library components

Within a characterisation performance study, the application is considered to be a sequence of sub-tasks (e.g. a sequence of computational cores). Each sub-task has its own parallelisation description, and can be mapped to a specific hardware platform. This structure can be visualised as shown in a PACE screen-dump PACE, Figure 3.

The autonomy of the layers permit the development of re-useable Characterisation models. One of the goals of the layered framework is the creation of libraries for each of: hardware platforms, parallel templates, and sub-task application models. In a characterisation performance study, it is possible to use pre-defined components, or to develop new ones if appropriate ones are not available.

A set of libraries is being established for the sub-task computations, parallelisation templates, and the hardware platforms. A number of computational kernels, and application codes [10-12] have been characterised, resulting in the availability of many library entries.

The library of hardware platform information, assembled to date by Warwick, includes: workstation (SPARC) network clusters, a Parsytec SuperCluster, and a Telmat Multinode [10-12]. The use of different message passing interfaces (MPI’s) is also possible. A view of some of the characteristic models for a hardware platform is portrayed in Figure 4. Each model is formed from an extensive analysis of measurements taken on the platform [13]. A suite of benchmark programs has been implemented which can be used to semi-automatically provide the necessary characterisation information for a hardware platform.

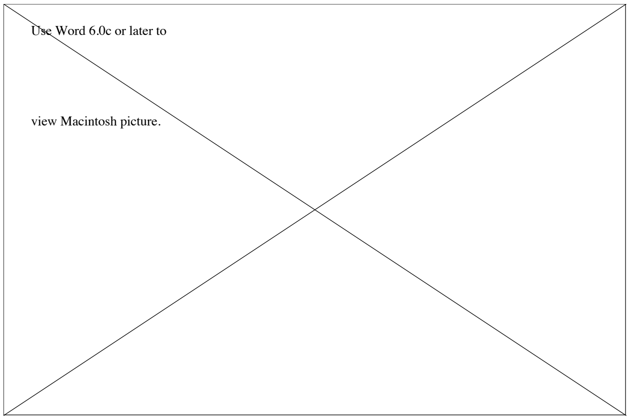

Figure 4 - Example characteristic models.

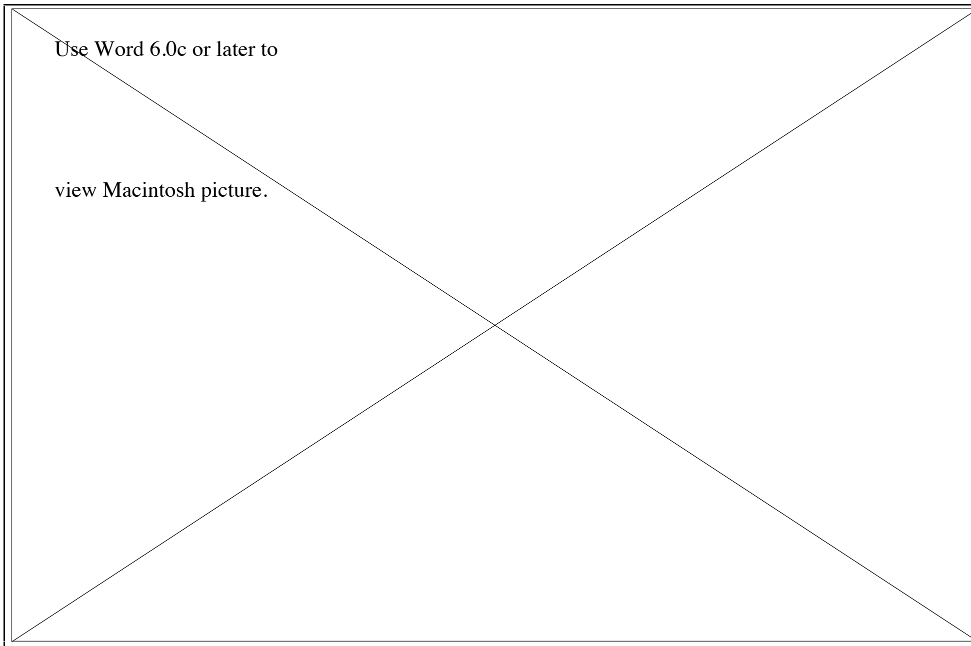

Figure 5 - Characterisation of a Financial Option Application showing Rate of calculation on various problem / system sizes.

5. Typical PACE Analysis

The PACE characterisation techniques have been used to provide insight into the performance of several of the applications, along with their scaling behaviour. An example is shown in Figure 5 for a Financial Option calculation (using Microsoft Excel plot formatting). The rate at which the calculation can be performed is shown on various processor sizes and problem sizes. (Note that the grid points in this figure have been fully predicted from characterisation, and the contours represent interpolated values from these points). Further information can be supplied such as the use of communication channels, and system bottlenecks.

The characterisation techniques within PACE have also been used to analyse the performance on several applications on different target platforms. This work has been performed within a number of testcases in the PEPS project by both internal partners (Warwick, Thomson), and also by external collaborators (DRA - Malvern, NA Software - Liverpool, Parallel Applications Centre - Southampton, and the Financial Options Research Centre - Warwick). A summary of the applications considered, target platforms, and resulting error margins observed from the characterisation approach is listed in Table 1.

The error margins show that the characterisation approach produces performance results which are accurate when compared with actual machine run-times. In a comparison performed within the PEPS project [6], the characterisation predictions were compared with the QNAP2/Modline Queuing network simulation package by Simulog [5]. The comparison showed that the error-margins observed for characterisation smaller in most instances.

6. Summary

Application Area	Target Platform	System Size	Mean Error
Financial Option Pricing [11]	Parsytec SuperCluster Telmat Multinode	128 16	6% 9%
Computational Chemistry [10]	Parsytec SuperCluster	128	5%
Route Optimisation [6]	Sun Workstation Cluster	5	5%
Image Processing [12]	Sun Workstation Cluster	6	7%

Table 1 - Characterisation application summary.

In this paper, the characterisation approach being developed at Warwick for performance analysis and prediction of parallel systems has been illustrated. The techniques are aimed at users with no specific performance related knowledge, and are accessible through the PACE toolset. The approach has demonstrated itself to be of importance in the performance analysis of parallel systems. It is able to undertake predictive performance analysis resulting in information on: time prediction, resource usage, scalability, and sizing. The approach has been used to investigate a number of application codes with resulting predictions typically within 5-7% of actual measurements.

The evaluation of a performance model within PACE enables a performance prediction result to be obtained in minimal time, typically taking less than 1 second per prediction. The automation of describing the resources directly from sequential application source code is currently available. These two features result in a flexible performance analysis environment which can be used by application developers to aid the use of parallel systems.

References

[1] "The Visual Display of Parallel Performance Data", M.T. Heath, A.D. Malony, D.T. Rover, IEEE Computer, 28(11), 1995, pp. 21-28.

[2] "Estimating and Optimizing Performance for Parallel Programs", T. Fahringer, IEEE Computer, 28(11), 1995, pp. 47-56.

[3] "Automated Scalability Analysis of Message-Passing Parallel Programs", S.R. Sarukkai, P. Mehra, R.J. Block, IEEE Parallel & Distributed Technology, 3(4), 1995, pp. 21-32.

[4] "PAWS: A Performance Evaluation Tool for Parallel Computing Systems", D. Pease, A. Ghafoor, I. Ahmed, L.D. Andrews, K. Foudil-Bey, and E.T. Karpinski, IEEE Computer, 24(1), 1991, pp. 18-29.

[5] Open Modline User's Guide Version 1.3, Simulog, France.

[6] "Validation of PEPS Tools", D. Bilsby, M. Hamer, J. Harp, H. Webber, PEPS Project Report D8.3.4, DRA Malvern, UK.

[7] "An Overview of the CHIP^{3}S Performance Toolset for Parallel Systems", E. Papaefstathiou, D.J. Kerbyson, G.R. Nudd, T.J. Atherton, in Proc. of ISCA Int. Conf. on Parallel and Distributed Computing Systems, Orlando, September 1995, pp. 527-533.

[8] "A Mathematica pre-processor for performance analysis of parallel systems", E. Papaefstathiou, D.J. Kerbyson, G.R. Nudd, T.J. Atherton, in Mathematics with Vision, V. Keranen and P. Mitac (Eds), Computational Science Publications, 1995, pp. 283-290.

[9] "Characterisation based bottleneck analysis of parallel systems", M.J. Zemerly, J. Papay, G.R. Nudd, Supercomputer, ASFRA, 11(4), 1995, pp. 89-101.

[10] "Self Consistent Field Computational Chemistry - PEPS Testcase 2", T.J. Atherton, D.J. Kerbyson, J. Papay, G.R. Nudd, PEPS Project Report D8.3.2, University of Warwick, Coventry, March 1996.

[11] "Financial Option Pricing - PEPS Testcase 3", D.J. Kerbyson, J. Zemerly, J. Papay, G.R. Nudd, PEPS Project Report D8.3.3, University of Warwick, Coventry, April 1996.

[12] "Image Processing - PEPS Testcase 5", Peps Project Report D8.3.5, NPL, London, April 1996.

[13] "Evaluation of Operating Systems", D.J. Kerbyson, G.R. Nudd, Y. Lavaux, G. Cristau, PEPS Project Report D5.5, University of Warwick, Coventry, September 1995.