Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Journal of Systems Architecture 48 (2003) 221–235

JOURNAL OF SYSTEMS ARCHITECTURE

www.elsevier.com/locate/sysarc

# PET, a software monitoring toolkit for performance analysis of parallel embedded applications

J. García *, J. Entrialgo, D.F. García, J.L. Díaz, F.J. Suárez

*Depto. de Informatica, Area de Arquit. Technol., University of Oviedo, Campus de Viesques s/n, Gijón 33204, Spain*

Received 1 August 2001; accepted 9 October 2001

**Abstract**

Since the late 1980s a great deal of research has been dedicated to the development of software monitoring and visualization toolkits for parallel systems. These toolkits have traditionally been oriented to measuring and analyzing parallel scientific applications. However, nowadays, other types of parallel applications, using signal-processing or image-processing techniques, are becoming increasingly importance in the field of embedded computing. Such applications are executed on special parallel computers, normally known as *embedded multiprocessors*, and they exhibit structural and behavioral characteristics very different from scientific applications. Because of this, monitoring tools with specific characteristics are required for measuring and analyzing parallel embedded applications.

In this paper we present performance execution tracer (PET), a monitoring and visualization toolkit specifically developed to measure and analyze this type of application. Special emphasis is placed on explaining how PET deals with the particular characteristics of these applications. In addition, in order to demonstrate the capabilities of PET, the measurement and analysis of a real application using this tool are described.

*Keywords:* Performance measurement; Performance visualization; Software monitoring tools; Embedded multiprocessors; Parallel embedded applications

## 1. Introduction

In parallel computing the monitoring of application behavior is considered indispensable. Early monitoring approaches were developed in the late 1980s [5,23,27]. Since then, a great deal of research has been dedicated to the development of software monitoring and visualization toolkits for parallel systems. Examples of these toolkits are: the pair of tools PICL/Paragraph [11,12] (for measurement and visualization respectively), the event monitoring utility for hypercube systems [17], the Pablo performance analysis environment [21], the Annai parallel programming environment [25], the Paradyn performance measurement tool [19], and the AIMS toolkit [26]. All these toolkits have been specifically designed to measure and analyze *scientific applications* executed on supercomputers, such as the IBM SP2 or the CRAY T3E.

* Corresponding author. Tel.: +34-985-18-22-26; fax: +34-985-18-19-86/22-28.

*E-mail address:* javier@atc.uniovi.es (J. García).

1383-7621/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S1383-7621(03)00004-3

Nowadays, other types of parallel applications are gaining importance in the field of embedded computing. These applications are being employed in a wide number of areas, such as real-time signal processing (radar, sonar, detection), image processing, vision systems, medical electronics, and so forth. In this paper, we refer to these applications as parallel embedded applications (PEAs) and to the parallel computers used to execute them as embedded multiprocessors (EMs).

EMs are built on boards for industrial buses such as the VME or the CompactPCI. Normally, these multiprocessor boards are made up of a small number of CPUs (frequently no more than a dozen) and are connected via the industrial bus to a host computer, which supplies filesystem service and other I/O capabilities to the multiprocessor. EMs can be based on different types of CPUs. In the late 1980s and early 1990's many EMs were developed using transputers [13]. Nowadays, RISC processors (PowerPCs) and DSPs (C6x, from Texas Instruments, and SHARC, from Analog Devices) are dominating the EM market.

PEAs exhibit structural and behavioral characteristics very different from scientific applications. For example, scientific applications are made up of only one program (as they are developed using the SPMD or the data-parallel programming models), while PEAs are developed using multiple concurrent programs. In addition, the software development systems (that is compilers, linkers, configuration tools, parallelization tools, etc.) used in the development of embedded and scientific applications also present significant differences. Therefore, the monitoring tools devoted to monitoring and analyzing PEAs will require special implementations, which take into account the particular characteristics of these applications and their development systems.

Examples of monitoring tools developed for PEAs are described in [2,3]. However, the amount of research carried out in the area of monitoring tools for PEAs is significantly smaller than that developed in the field of monitoring tools for scientific applications. This has been the main motivation for the development of performance execution tracer (PET), a monitoring and visualization toolkit specifically developed to measure and analyze PEAs.

The rest of this paper is organized as follows. In Section 2 a typical programming model for PEAs, as well as the basic characteristics of the development systems used in the implementation of these applications, are described. In Section 3 the evolution of the PET toolkit as well as the novel characteristics introduced in its current version are described. Section 4 provides a detailed description of the PET monitoring tool, as well as the instrumentation process of PEAs using the PET tools. The main characteristics of the PET visualization tool are described in Section 5. Section 6 shows the utilization of PET in the analysis of a real application. And the paper finishes with the concluding remarks provided in Section 7.

## 2. Parallel embedded applications: programming model and development systems

It is of primary importance that the measurements taken by any monitoring tool are appropriate to explain the performance behavior of the measured applications. However, different types of measurements may be required depending on the kind of application to be measured and analyzed. Thus, the measured information necessary to analyze PEAs is significantly different from that required by other types of parallel applications. The most appropriate measured information to analyze an application depends on the behavioral characteristics of that application, which in turn, depend on the programming model used in its development. So to determine the measured information necessary to analyze PEAs, the programming model used in the development of these applications must be precisely defined. The main characteristics of this programming model are enumerated in Section 2.1.

In addition to defining precisely the information that must be measured by monitoring tools, these must also be designed to integrate well into the development systems used for application development. In fact, a monitoring tool can be considered as a component (like a compiler, linker, etc.) of a development system. So in the design of a