Petri Net Performance Models of Parallel Systems - Methodology and Case Study*

H. Wabnig, G. Haring

University of Vienna
Institute of Applied Computer Science and Information Systems
Department of Advanced Computer Engineering
Lenaugasse 2/8, A-1080 Vienna, Austria

Tel: +43 1 408 63 66 10, Fax: +43 1 408 04 50
e-mail: wabnig@ani.univie.ac.at, haring@ani.univie.ac.at

Abstract. In the PAPS - parallel program performance prediction toolset - parallel systems are specified by the structure of the parallel program, the multiprocessor hardware, and the mapping of the program elements to processor nodes. The task scheduling strategy and the communication network behaviour is described in terms of timed Petri nets. Examples of Petri nets, reflecting different task scheduling strategies, are presented. A detailed Petri net performance model for the Virtual Channel Router (VCR) which is a software implementation of a packet switching communication kernel built upon a store & forward communication network is elaborated and validated. Resource parameters for an actual multiprocessor computer system running the VCR communication software are determined. A case study shows the applicability and accuracy of the presented Petri net performance models for VCR based packet switching communication networks.

Keywords: Performance evaluation, simulation, parallel processing.

1 Introduction

The PAPS - parallel program performance prediction toolset [34] contains in its current version an initial set of Petri net based performance prediction tools. The toolset is aimed to support the performance engineering activities in the early stages of parallel program development as they are defined within the context of CAPSE (Computer Aided Parallel Software Engineering) [5]. CAPSE is an integrated parallel software development environment, consisting of a well designed set of tools to support the whole parallel software development lifecycle. These tools are currently under development at the Universities of Vienna and Linz.

* This work was funded by the Austrian Science Foundation (research grant S05303-PHY).
Other integrated development environments for parallel architectures are PIE from Carnegie Mellon University [31], Poker from the University of Washington [33], PPSE from the Oregon Advanced Computing Institute [22], FAUST from the University of Illinois [12], TOPSYS from the University of Munich [7], IMSE [27], Tangram from UCLA Computer Science Department [11] and Paragon from Cornell University [1]. A good overview on performance tools rather than complete environments has been worked out by Nichols [21], a summary of existing design tools by Oman [23] and a summary of parallel processing tools by Harrison [24].

At a certain level of abstraction the logical structure of a parallel program, the architecture and the mapping of the parallel program onto the architecture are the most critical aspects, influencing the performance of parallel programs, which are being executed on a parallel hardware. To keep the specification of these three parts as easy and flexible as possible the PRM (program - resource - mapping) methodology is applied. The PRM methodology has been designed according to the PRM-net models elaborated in [4]. The main idea of the PRM methodology is the separation of the three input specifications. Due to the separation it should be possible to vary one of them without having to adapt the other specifications. That should make it possible to compare the performance of programs on different hardware architectures or with different mappings very quickly and efficiently. In distributed memory MIMD multiprocessor systems the communication structure of the parallel program is an important performance factor. Another performance criteria are precedence relations between blocks of computation. Hence, these aspects have to be described in the program specification. Directed task graphs [10] are the basis for the program specification in PAPS. Task graph nodes are used to represent computational demands and arcs represent logical data flow between nodes. The hardware is described by a directed processor graph. Nodes in the processor graph represent processing elements and directed arcs communication links that allow to transfer messages from one processor to another processor. The task graph nodes have to be mapped onto processor nodes.

The PAPS toolset is dedicated to MIMD computers with distributed memory. That means that accesses to data not stored in local memory have to be implemented by passing messages between processors. Petri net performance models can be automatically generated by the PAPS toolset for various communication network types and task scheduling strategies. Petri net simulation is used to evaluate the performance models. During the simulation of the Petri net, performance relevant events, e.g. begin and end time of the execution of a task graph node, are stored in a tracefile in PICL syntax (Portable Instrumented Communication Library [8] [9]). The generated tracefiles can be visualized by the well known visualization tool ParaGraph [13] [14]. ParaGraph has a multitude of different graphical displays giving insight into the execution of parallel programs. Because ParaGraph does not support multiple processes running on

---

2 PICL is a portable communication library, which has certain features to collect event and statistic trace information.