Costs and Benefits of Multithreading with Off-the-Shelf RISC Processors

Olivier C. Maquelin\textsuperscript{1}, Herbert H.J. Hum\textsuperscript{2}, Guang R. Gao\textsuperscript{1}

\textsuperscript{1} McGill University, School of Computer Science
3480 University St., Montréal, Canada, H3A 2A7
\textsuperscript{2} Concordia University, Dept. of Electrical and Computer Engineering
1455 de Maisonneuve W., Montréal, Canada, H3G 1M8

Abstract. Multithreaded architectures have been proposed for future multiprocessor systems due to their ability to cope with network and synchronization latencies. Some of these architectures depart significantly from current RISC processor designs, while others retain most of the RISC core unchanged. However, in light of the very low cost and excellent performance of off-the-shelf microprocessors it seems important to determine whether it is possible to build efficient multithreaded machines based on unmodified RISC processors, or if such an approach faces inherent limitations. This paper describes the costs and benefits of running multithreaded programs on the EARTH-MANNA system, which uses two Intel i860 XP microprocessors per node.

1 Introduction

Multithreaded architectures [1, 2, 12] have been promoted as potential processing nodes for future parallel systems due to their inherent ability to tolerate network and synchronization latencies. These delays are hidden by letting the processing unit switch to a different thread of execution instead of idling until the operation has completed. Due to the additional synchronization overhead when taking advantage of parallelism at a finer level, many architects question whether multithreading support can be made transparent to sequentially executing code and still be useful. However, preliminary results gained with the EARTH-MANNA system show that multithreading can indeed be useful, even on machines built with conventional RISC microprocessors. Even though the Intel i860 XP processor used in EARTH-MANNA was not designed for multithreading, benchmark results show that good speedups can be achieved, even compared with an efficient sequential implementation. Moreover, a detailed analysis of the multithreading overheads shows that they could be reduced significantly without having to switch to a custom processor design.

1.1 The EARTH-MANNA system

The results discussed in this paper were gained with our implementation of the EARTH (Efficient Architecture for Running THreads) model [4, 6, 5] on top of
the MANNA (Massively parallel Architecture for Non-numerical and Numerical Applications) multiprocessor [3] developed at GMD-FIRST in Berlin, Germany. Each node of a MANNA machine consists of two Intel i860XP RISC CPUs, clocked at 50 MHz, 32 MB of dynamic RAM and a bidirectional network interface capable of transferring 50 MB/s in each direction. This dual-processor design is similar to the EARTH model (see Fig. 1), which separates the processing node into an Execution Unit (EU) and a Synchronization Unit (SU).

![Diagram of EARTH architecture](image)

Fig. 1. The EARTH architecture

As demonstrated in [5], it is possible to implement multithreading support for such a machine without a major impact on performance. Performance of the parallelized code on a single node can be close to that of the sequential code because substantial portions of the code can often be executed in the normal sequential way. As shown in Sect. 3.3, performance gains can also be achieved by taking advantage of the SU to off-load data transfers from the main CPU. It is also interesting to note that for the EARTH-MANNA system the cost of saving and restoring registers is only a relatively small fraction of the total context switch costs (see Sect. 2.4). This means that better hardware support for multithreading should focus primarily on reducing the remaining costs, which are mostly due to communication between the EU and SU.

1.2 Synopsis

The next section discusses the overall performance of the EARTH-MANNA system and the costs associated with the multithreading support, such as the overhead to issue a split-phase transaction and the context switching costs. Then, Sect. 3 gives some insights into the relative performance of single-processor vs. dual-processor node designs. The costs for all internal operations involved in a remote memory access are shown for both cases, and finally some experimental results showing the benefit of a second processor are discussed.