CHAOTIC LINEAR SYSTEM SOLVERS
IN A VARIABLE-GRAIN DATA-DRIVEN
MULTIPROCESSOR SYSTEM*

Jean-Luc Gaudiot and Chih-Ming Lin

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-0781

Abstract

Linear systems are important problems in many scientific applications. While asynchronous methods are effective solutions to linear systems, they are difficult to realize due to the chaotic behavior of the algorithms. In this paper, we investigate the implementation as well as the performance of an asynchronous method, namely chaotic relaxation, in our Variable-grain Tagged-Token Data-flow (VTD) System. We compare asynchronous methods with synchronous methods executed on both the fine-grain and the coarse-grain execution models. New high-level data-flow language constructs are introduced in order to express asynchronous operations. A new firing rule that deviates from the single assignment rule of functional languages is proposed to support the implementation of asynchronous computations in the VTD system. In addition to the conventional speedup measure, we then define new performance measurements, called Growth Factor, Scalability Factor, and Robustness to characterize the system performance from the machine and application viewpoints. Simulation results indicate that asynchronous methods are more efficient than synchronous methods and that the coarse-grain execution mode is more efficient than the fine-grain execution mode in our VTD system.

1 Introduction

Linear systems play an important role in many applications such as PDE solvers. Generally, linear systems can be solved by direct or iterative methods. Iterative methods can further be classified as synchronous [9] or asynchronous [3]. While synchronous methods are easy to implement, they do not yield acceptable levels of performance for complex problems, mainly because of the synchronization necessary among the various processes. On the other hand, asynchronous approaches have been found by many researchers [3, 5] to efficiently exploit runtime parallelism. In an asynchronous approach, communication between processes is achieved by reading the dynamically updated variables while each

* This material is based upon work supported in part by the U.S. Department of Energy, Department of Energy Research, under Grant No. DE-FG03-87ER25043.
process continues its execution to update shared variables. Therefore, the chaotic behavior of data in an asynchronous algorithm is very complex. However, while an asynchronous method can be effective in parallel machines and can deliver high performance, it is difficult to implement due to the chaotic behavior of the method itself. From the software perspective, language constructs must be defined to specify the asynchronous method, thereby parallelizing the algorithm. From the hardware point of view, special architecture schemes dedicated to the algorithm need to be developed.

The data-flow principles of execution [2] offer the programmability needed to synchronize at runtime the many parallel processes on a large scale multiprocessor. Instead of relying on the conventional central program counter, the availability of data renders an instruction executable. Asynchronous algorithms have been implemented in data-driven systems, more precisely in micro-actor-based data-driven systems [5]. Although the micro approach to asynchronous methods correspond well to the simplicity of data-driven principles, it yields much overhead to respect the functionality of execution.

In this paper, we will first introduce special high-level data-flow language constructs (Async-Repeat and Async-For) to describe the chaotic behavior in asynchronous algorithms. The scheme to form coarse-grain (macro-actor) data-flow graphs and a specific firing rule in the Matching Store with Locks of processors will also be introduced in order to correctly execute the computations of the asynchronous algorithms. In this paper, we are also interested in measuring and comparing the performance of algorithms as well as our VTD system: First, to evaluate the performance of the architecture, the conventional "Speedup" measurement will be taken to depict the trend of the performance with larger machine configurations. Second, to estimate the growth of parallelism within an algorithm when the algorithm's complexity has been increased, a new measurement, called "Growth Factor", will be defined to show how suitable an algorithm is for multiprocessor systems. Third, to measure the efficiency of parallel systems in the execution of parallel algorithms, we will introduce another new measurement, called "Scalability Factor", to demonstrate the scalability property of the systems. Finally, we will define "Robustness" to indicate the potential performance of the systems.

We shall start our discussion in section 2 by giving a brief introduction to the data-flow principles of execution as well as to asynchronous methods for solving linear systems. In section 3, the Jacobi method and the chaotic relaxation method are described in a High-level data-flow language along with the new languages constructs. The VTD System and the new firing rule for chaotic relaxation and the new performance measurements will be described in section 4. Section 5 will present the results of a deterministic simulation on the system and concluding remarks will be made in section 6.

2 Data-flow Principles and Iterative Solutions for Linear Systems

In this section, we first introduce the data-flow principles of execution and review the essentials of the synchronous and asynchronous linear system solvers which will be evaluated on our VTD system.