Architectures, Compilers and Applications in Multiprocessors
TIC2003-06623

Emilio López Zapata *
Depto. de Arquitectura de Computadores
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Abstract

The high performance experienced by current computers is the result of combining advances in fabrication technologies (microelectronics, optics and magnetic), in architecture (new design techniques) and in compilers. All of these allows to adapt different components, with different speeds, between themselves with minimum performance degradation of the fastest components. We are all convinced that this overall performance will increase in the coming years. However new computer applications show up in the horizon that demand for higher computational power, higher storage capacity and faster communications.

In this project we will explore in depth the new multiprocessor systems (clusters) and Grid computing, considering preferentially the important problems around mass storage and data management and processing, typical of multimedia applications. We will approach this subject from three main research lines: 1) Automatic optimization (parallelism, locality) of irregular and dynamic applications, 2) mass storage, multimedia, graphics and video processing architectures, and 3) emerging applications, in fields like audiovisual information, bioinformatics and numerical simulation.

Keywords: Supercomputing, Parallel Computing, Optimizing Compilers, Advanced Architectures, Data Locality Exploitation, Parallelism Exploitation, Storage Architectures, Multimedia, Graphics Processing, Video Processing, Bioinformatics, Parallel Numerical Algorithms.

1 Project Goals

This report is organized according to the three main research lines that were defined in the initial project proposal (and specified at the end of the abstract of this document). The specific goals of each research line are summarized in this section. It is important to note that this project is the natural continuation of the research accomplished in previous projects, that combined basic and applied research.

*Email: ezapata@ac.uma.es
1.1 Automatic optimization (parallelism, locality)

This line of the project is focused in the design and development of new techniques to optimize large and complex scientific and engineering codes (irregular and dynamic, or pointer-based, codes).

a) Automatic optimization/parallelization of irregular and dynamic codes. Current compilers usually fail when optimizing dynamic codes, that include pointer-based, complex data structures and use dynamic memory allocation. Such compilers are unable to extract from the code the information needed to exploit important performance aspects, like parallelism and locality. Many of the emerging applications that we study in the third research line, for instance, need from a deep analysis of the data structures and memory references in order to optimize them. So data analysis, that includes data organization and memory reference pattern, is a key step in the optimization of dynamic codes. There has been much research in this topic, allowing some effective compiler optimizations. However, these results are not enough to tackle more aggressive optimizations. In previous years we have developed new techniques based on shape analysis in order to capture the main characteristics of pointer-based data structures. This module is focused in advance such techniques and propose new ones to allow effective compiler phases, like interprocedural analysis and data dependence analysis, for dynamic codes. In addition, these new techniques will be used to develop new strategies for exploiting parallelism and locality.

1.2 Architectures

This line is focused in the design of new architectures for specific applications.

a) Distributed system for video on demand (VoD). A CDN (Content Distribution Network) is an economically viable solution for a VoD system. A CDN consists of a distributed system of servers interconnected through a specific network. CDNs present diverse difficult problems, like content replication, mass storage consumption, network traffic, and so on. This module proposes to develop a distributed VoD system where the video archives are partially replicated based on their popularity index. In addition different multicast techniques will be evaluated.

b) Mass storage architectures. The fast development of digital multimedia contents during the last years have entailed strong requirements on mass storage. Robust, scalable mass storage systems are unavoidable in this scenario. Typically, a scalable storage system store a small amount of the whole multimedia contents in the secondary level (hard disks), usually the most recently/frequently used data. The rest of the data is stored in a tertiary level (sequential magnetic tapes), of large capacity. This module is focused in the design of a scalable, massive storage architecture model, able to cover the needs of the current and future storage-demanding applications (multimedia, bioinformatics, CFD, astronomy ...).

c) Arithmetic and multimedia processors. Arithmetic hardware for evaluating elementary functions usually works with input arguments within a specific value interval. So when evaluating such functions, a rank reduction of input arguments are required before using the arithmetic hardware. This rank reduction is usually made by means of a series of multiplications and divisions. However, when the input values are very large, the precision of resulting reduced values may be compromised. This module proposes to design a specific hardware to accomplish
the argument rank reduction in a reliable means. On the other hand, hardware for multimedia extensions attached to processors has resulted in new possibilities to be developed. In this module we also study the design of arithmetic elementary functions using the hardware for multimedia extensions.

d) MPEG-4 codec architectures. MPEG-4 is a recent standarized video compressed format. Low profiles of this standard can be covered by current processors. However, medium and high profiles cannot be currently fully developed due to the great computational power required. This module is focused in the design of specific architectures for medium and high profile MPEG-4 codification/decodification.

1.3 Emerging applications

This line is focused in the study of a selection of applications that demand large computational resources.

a) Audiovisual information systems. Storage and management of audiovisual information are experiencing great changes in the last few years, in the context of digitalization of video/audio contents. In fact, recent standards, like MPEG-4, proposes new efficient methods for encoding multimedia data, including content-descriptive elements. In this context, it is clearly needed a common model to describe audiovisual information in a way useful for applications in diverse areas (multimedia, biology, astronomy ...). This module is focused in the design of a series of analysis techniques for processing audiovisual information, like those needed for the detection of key frames in a compressed video (storyboard), based on some characteristics. Or those needed to detect and extract objects in compressed video.

b) Genomics and proteomics. Recent advances in genomics entail an increasingly important role for bioinformatics research, in contrast to applied research in the laboratory. In fact, at present the huge volume of data generated in laboratories requires the development of new and efficient computational solutions for data storage and classification. This module proposes the design and implementation of software architectures for storage, management and analysis of biological data.

c) Simulation of flexible objects, and other scientific/engineering problems. The simulation of flexible objects is an essential tool for realistic computer animations of virtual humans and dynamic scenarios. New technologies, like digital TV and multimedia devices, require the development of applications for cheap, (near) real-time virtual reality simulations. This module tackles this problem from three approaches: the mathematical model, the solution of the equation systems, and the optimization techniques for high-performance computation. In this module we also include research about the design of efficient solutions to other scientific and engineering problems, using mainly parallel computation.

2 Level of Success Reached in the Project

At the time of writing this report, the project has consumed 20 months out of its whole three-year period. At present, a large part of the proposed goals have already been accomplished. In this section we summarizes the main research achievements reached until now.
2.1 Automatic optimization (parallelism, locality)

a) Automatic optimization/parallelization of irregular and dynamic codes. We have proposed a methodology to develop efficient parallelization techniques for irregular and dynamic applications [5, 39, 40]. Based on this methodology, a framework using shape analysis was designed to capture properties of pointer-based dynamic data structures [8]. This framework was partially used to propose new strategies to exploit locality and parallelism for such codes [16, 33]. A key data analysis component for such optimizations is an effective data dependence analysis for pointer-based data structures [9, 32, 51]. We have also developed a tool for performance analysis of the cache [11], new techniques for efficient parallelization of irregular reductions [15, 17], and a cache model to design efficient locality-based loop transformations [18, 19]. Finally, we have experimented with a novel approach for exploiting the graphical resources (GPU) of a commodity PC for different applications, like fast retrieval of indexed data [57] and fast execution of irregular computations [58].

2.2 Architectures

a) Distributed system for video on demand (VoD). We have proposed hybrid multicast algorithms inspired by the Maximum Factored Queue Length (MFQL) batching scheme used to decide which video queue will be serviced with a multicast session [13]. We have also designed a new threshold-based patching scheme to control partial stream transmission during an ongoing multicast session [14].

b) Mass storage architectures. New storage technologies, like SAN and iSCSI, are essential for the deployment of audiovisual information systems. Apart from the high storage capacity, the access performance to huge archives is a critical issue. The PhD research of one component of the group tackles this problem. To date, we are developing in collaboration with Tedial an automatic managing system for distributed mass storage in the context of professional TV.

c) Arithmetic and multimedia processors. We have used the CORDIC algorithm to develop different solutions in the arithmetic field: an improvement of the pipelined CORDIC with linear approximation to rotation [4], a CORDIC processor for variable-precision coordinates, that permits a reliable accurate evaluation of elementary functions [27], and a new range reduction technique optimized for CORDIC [26]. We have studied on-line arithmetic for digital signal processing: a design of a comparison operation for signed digit representations [24], a proposal of an on-line full-adder to deal with multioperands [62], and an VLSI architecture for the integer-to-integer wavelet transform [25]. Finally, we have analyzed how to use multimedia extensions to compute elementary functions based on polynomial approximations [6].

d) MPEG-4 codec architectures. We have proposed a FPGA design for fast computing of the minimum sum of absolute differences (SAD), used in block based motion estimation algorithms in MPEG [35, 34, 23].

2.3 Emerging applications

a) Audiovisual information systems. We have developed a video content analysis and metadata organizational system for research videos arising from biological microscopy of living cells [42, 43, 44]. We have also designed solutions to the temporal segmentation of MPEG compressed
video sequences, based on different video characteristics, like edges, luminance and gradual transitions [49, 50]. We have proposed new methods for image processing, like a fully automatic algorithm for extraction of vessels in noisy medical images [59, 60], and a time-frequency method for texture segmentation and texture inhomogeneities detection [22].

b) Genomics and proteomics. We have developed an architecture for XML-based mediator systems and a framework for helping system developers in the construction of mediator-services to integrate heterogeneous data sources in the biological field [11, 2, 3]. We have proposed a mathematical model to extend the dynamic range of gene expression data measured by laser scanners [12]. We have also studied applications of knowledge discovery technology (data mining) to bioinformatics. Basically, we combine exhaustive database searching and automatic pattern recognition strategies with discovering of association rules [41, 7]. Two surveys related to bioinformatics were also accomplished in this period. One is about computational strategies to parallelize most of the typical applications in bioinformatics, specially those based in metaheuristics approaches [55]. A second one studies grid technologies and architectures, and how they are being used in biological sciences [52]. Related to this, a new adaptive task scheduling model was designed for computational grid environments [20, 21], as well as a new component (service) of a computational grid, AnaGram, was developed for protein function assignment [38].

c) Simulation of flexible objects, and other scientific/engineering problems. We have developed efficient parallel solutions to irregular reductions in the context of physically-based fabric simulators. Such operations appear when evaluating forces, a very time consuming task [45]. In the context of simulation, we have proposed a three-level (optical, energetics and physiological) model for simulating the interaction between laser irradiation and human tissues [61, 47, 10]. We have also analyzed different parallel implementations of the (preconditioned) conjugate gradient method for band systems, for several architectures and programming paradigms [46, 36]. From our experience in parallelizing complex applications, we have proposed an uniform interface for the design of efficient hybrid (shared-memory and message-passing) parallel codes [37].

3 Result Indicators

a) Publications
It should be highlighted that this research has already produced an important amount of results that have been published in the most prestigious journals and conference proceedings of the field (many from IEEE and ACM). In summary, during this period, the project produced: 19 publications in journals listed in JCR, 31 papers in international conferences, 10 papers in national conferences, and 2 book chapters.

b) PhD Thesis
In addition, the ongoing research of PhD candidates working in the project should produce three more PhD thesis by the end of the project (december 2006).

c) Cooperation with other groups
An important part of the research described in section 2 is being developed in cooperation with
several prestigious, world-level research groups: Prof. David Padua’s group, from the Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, USA (section 2.1.a), Prof. Joel Saltz’s group, from the Biomedical Informatics Dept., The Ohio State Univ., USA (section 2.1.a), Prof. Tomas Lang, from the Dept. of Electrical and Computing Engineering, Univ. of California at Irvine, USA (sections 2.2.c and 2.2.d), and Dr. José M. Carazo’s group, from the CNB (Centro Nacional de Biotecnología) (section 2.3.b).

d) Participation in European projects
The group is participating in the European Network of Excellence on High-Performance Embedded Architectures and Compilers (HiPEAC), Project No.: IST-004408.
The group is also participating in the European Project ”Express-Fingerprints, Expression Profiles as Fingerprints for the Safety Evaluation of New Strains Including GMOs Used in Bioprocessed Food”, Project QLRT-2000-01473.

e) Technology transfer activities
Members of the research group have a permanent collaboration with Tedial company, being in charge of its R&D and transferring technologies appropriate for the analysis and automatic indexing of video, audio and images, as well as the design of high performance processing and mass storage architectures. A similar level of collaboration is being carried out with Integromics company, specialized in the design of bioinformatics applications for genomics and proteomics.

References


[34] J. Olivares, J. Hormigo, I. Benavides and J. Villalba Alternativas al Cálculo de la Estimación de Movimiento en MPEG, XV Jornadas de Paralelismo, Almería, Spain, 15-17 September 2004, pp. 84-89.


Applications meant to be run in multiprocessor, multi-core environments could end up with serious, hard-to-find performance issues if these considerations are not addressed during design. This article outlines some of the key considerations for designing software for multi-core, multiprocessor environments. Impediments to software scalability on chip multithreaded, multi-core, multiprocessor architectures. Applications are expected to scale and perform better on multi-core, multiprocessor environments. Although, compilers and runtime environments could aid in optimizing to a certain level, you cannot depend on it to address all the scalability issues. The spread of multiprocessor architectures will have a pervasive effect on how we develop software. Until recently, advances in technology meant advances in clock speed, so software would effectively "speed up" by itself over time. Now, however, this free ride is over. Multiprocessor programming is challenging because modern computer systems are inherently asynchronous: activities can be halted or delayed without warning by interrupts, preemption, cache misses, failures, and other events. These delays are inherently unpredictable, and can vary enormously in scale: a cache miss might delay a processor for fewer than ten instructions, a page fault for a few million instructions, and operating system preemption for hundreds of millions of instructions. Synthesis of Application Speciﬁc Multiprocessor Architectures for Process Networks. Basant Kumar Dwivedi, Anshul Kumar and M. Balakrishnan. Dept. of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India. {basant, anshul, mbala}@cse.iitd.ernet.in. Abstract. In this paper, we address the problem of synthesis of application speciﬁc multiprocessor SoC architectures for process networks of streaming applications. An application is modeled as Kahn Process Network (KPN) which makes the parallelism present in the application explicit. The synthesis process involves...
Abstract models are (mostly) OK to understand algorithm correctness and progress. To understand how concurrent algorithms actually perform, you need to understand something about multiprocessor architectures. We look at how multiprocessor hardware architecture affects the design of efficient concurrent data structures and algorithms. We identify basic components, describe what they do, how they interact, and why some activities that appear fast and simple may sometimes be slow and complex. Multiprocessors present a challenge: systems are being used successfully today to improve performance in systems running multiple programs concurrently. In addition, multiprocessor systems have shown the ability to improve single-program performance significantly for certain applications containing easily parallelized loops. The extraction of coarse-grained parallelism from a software description and, indeed, the study of languages used to describe parallel software are a flourishing area of research.

With the introduction of the single-chip multiprocessor (Olukotun et al., 1996), the dividing line between research Multiprocessors, Multicomputers, and Clusters. In this and following lectures, we shall investigate a number of strategies for parallel computing, including a review of SIMD architectures but focusing on MIMD. The two main classes of SIMD are vector processors and array processors. We have already discussed each, but will mention them again just to be complete. There are two main classes of MIMD architectures (Ref. Note that each of the SIMD and MIMD architectures call for multiple independent processors. The main difference lies in the instruction stream. SIMD architectures comprise a number of processors, each executing the same set of instructions (often in lock step). MIMD architectures comprise a number of processors, each executing its own program.