Parallelism Across the Curriculum

John E. Howland
Department of Computer Science
Trinity University
One Trinity Place
San Antonio, Texas 78212-7200
Voice: (210) 999-7364
Fax: (210) 999-7477
E-mail: jhowland@Trinity.Edu
Web: http://www.cs.trinity.edu/j̃howland/

August 3, 2006

Abstract

A sea change within the computing discipline occurred last year which is not widely recognized nor fully understood. Individual processor chips are not going to be faster in the future. The implications of this fact are discussed and changes to the undergraduate computing curriculum are proposed. [1]

Subject Areas: Computer Science Education, Computer Science Curriculum.
Keywords: Parallelism, Parallel Algorithms.

1 Introduction

As Bob Dylan said,

"...the times they are a-changin' ..." [2]

From the very beginning, the computing discipline has experienced change at an astonishing pace. Computing in academe has grown accustomed to rapid paced change and is used to frequent, periodic curriculum review. However, an announcement last fall caught many within our field by surprise.

[1] This paper was published in the Journal of Computing Sciences in Colleges, Volume 21, Number 4, April 2006, Pages 134-138. Copyright ©2006 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

On Friday, October 15, 2004, Intel Corporation announced [5] to the business community that “Intel Corp.is scrapping plans to hit a high-performance milestone for its flagship microprocessor, the latest in a series of course changes and miscues by the big chip maker.” The company said that it would not be able to meet its goal of releasing a four giga-hertz Pentium 4 processor chip. Moreover, future chips in the Pentium 4 line were likely to be a little slower than the 3.8 giga-hertz chip available at the time.

It wasn’t that Intel did not want their processors to run at higher clock rates, but rather, it was impossible for them to deliver chips which would reliably achieve frequencies at or above the four giga-hertz rate.

1.1 Moore’s Law

In the April 19, 1965 issue of *Electronics* [3], electrical engineer Gordon Moore noted that the number of circuit elements inside chips was doubling about every year. Ten years later, Moore observed, because of the increasing complexity of chips, that the pace would slow to doubling the number of transistors in a chip about every 24 months. Moore’s law is not a law of physics, but reflects, more than anything else, the competitive forces within the chip manufacturing industry. Moore’s law continues to hold and Moore now predicts that the number of transistors in processor chips will continue to double about every 18 months for the next twenty years.

Until recently, the increase in speed of processor chips tracked the rate of increase of transistors. However, Intel’s announcement indicates that even though Moore’s law continues to hold, speed increases will not continue due to heat dissipation problems and current leakage problems as the internal components of a chip get smaller.

Intel, as well as other processor chip manufacturers such as AMD, IBM, Freescale, etc., have decided to use the results of Moore’s law by producing processors which contain multiple CPU units which share a common memory system. This means that in order to produce software designs which run faster, those designs must employ the techniques of parallel computing.

1.2 Changes to Computer Science Education

Moore’s law has provided an environment which allows system design to assume that more powerful processors will soon be available to properly execute bloated designs. Computer science education has contributed to this trend often by teaching techniques which are not optimal and students learn to rely on ever faster processors to run their non-optimal algorithms and designs.

This author told the 2004 graduating computer science class that they could no longer rely on faster processors in their future designs even though that possibility existed when they began their computer science educations in 2000. They were challenged to become more expert in parallel computing techniques so that their algorithms and designs could take advantage of parallel processing chips.

2 Curriculum Changes

Intel's announcement provoked discussion within our computer science department about the kind of changes needed within our curriculum to properly prepare our graduates to be productive in an increasingly parallel future. Input was also received from the department's external advisory board concerning the effects processor speed limits would impose on the computing industry. The idea of parallelism across the undergraduate computer science curriculum emerged from these discussions. It should be noted that Computing Curricula 2001 Computer Science [1] does not address the subject matter of parallel computing being distributed across the undergraduate curriculum.

Two faculty members of our department have research specialties in parallel computing. As a result, our department already offers two elective courses (Parallel Processing and Advanced Topics in Parallel Computing) which may be taken by junior and senior computer science majors.

2.1 Parallelism Across the Curriculum

Our department is beginning to experiment with several short modules within several standard courses in our undergraduate curriculum. Many of the modules are short topics with demonstrations which may be covered in one or two lecture periods.

- CS 1

  Simple Threads model to illustrate two tasks computed in parallel using pthreads.

- CS 2

  Several threading examples of removing device latencies in common algorithms using the Java thread library.

- Data Structures

  Array data structures for parallel algorithms.

- Discrete Structures

  Mathematical models for synchronizing parallel tasks.

- Functional Languages

  Church-Rosser property of independence of order of evaluation.

- Upper Division Elective Courses

  - Computer Graphics

    Parallel processing within modern raster display processors. Techniques of parallel image rendering.

  - Computer Networks

    Multiple network interfaces to increase network transfers. Multiple daemons for handling client-server requests such as serving web pages.

  - Database Systems
    Multiple daemons for handling client-server requests in database applications.
  - Simulation Theory
    Application of parallel processing in scientific simulations.
  - Artificial Intelligence
    Parallel algorithms for evaluating decision nets.

  Since two of our computer science faculty have research specialties in parallel computing we have two upper-division courses in parallel computing. These courses provide in-depth treatment of current topics and trends in parallel computing and parallel languages.

- Parallel Processing Fundamentals

- Advanced Topics in Parallel Processing

3 Laboratory Equipment

Parallel hardware is necessary to provide demonstration of concepts and provide laboratory experience. Fortunately, it is now relatively inexpensive to acquire such equipment as desktop systems with dual core or dual processor systems are now common. We have found that a few such machines running some dialect of Unix (Linux or OS X) are sufficient to provide laboratory experience in all of the above course modules and courses. It is also possible to use lab computers or classroom computers running Linux to provide laboratory experience with parallel clusters using the mpi libraries. An NSF-CCLI grant proposal is being prepared to fund a cluster of dual processor machines which will be used to support laboratory experiences for our parallelism across the curriculum initiative.

4 Conclusions

The computing industry has an insatiable desire for increased performance. “Intel Corp. [6] has laid out plans to exploit two trends – parallel computing and a wireless technology known as *WiMax* – that the chip maker believes will drive electronics demand for years.” “Intel Corp. [4] vowed to deliver a tenfold boost to computer performance in the next three years by putting the equivalent of multiple electronic brains on each chip.” With these announcements to the business world, it is clear that parallel computing will be central to the future of computing. This future poses a challenge [7] to software makers and computer science education.

The motivation for this paper is to share preliminary ideas and experience of introducing parallel computing through out the undergraduate computer science curriculum with the hope of provoking discussion and interchange concerning changes within the computing discipline which will likely define the future of computing. Our initial results indicate that it is possible to introduce parallel processing concepts at a very early point in the curriculum and continue to expand and develop these ideas through several intermediate courses, culminating, depending on student interest, in junior and senior level courses in parallel computing.

References

[1] *Computing Curricula 2001 Computer Science, Final Report*, The Joint Task Force on Computing Curricula, IEEE Computer Society and Association for Computing Machinery, IEEE Computer Society, December 2001.

[2] Bob Dylan, *The Times They Are A-Changin*, CBS Records, February 10, 1964.

[3] Moore, Gordon, "Electronics", April 19, 1965.

[4] The Wall Street Journal, "Intel Aims for a Tenfold Boost In Chip Performance by 2008", Page B8, Column 1, December 8, 2004.

[5] The Wall Street Journal, "Intel Gives Up on Speed Milestone", Page B8, Column 1, October 15, 2004.

[6] The Wall Street Journal, "Intel Plans to Exploit Chip Trends", Page B3, September 8, 2004.

[7] The Wall Street Journal, "New Chips Pose a Challenge to Software Makers", Page B3, April 14, 2005.