Parallelism: The Future of Computing?

Winston A. Sumalia
CS 252 [EE298]
University of the Philippines

Abstract—**The advent of parallelism is being widely embraced by avant-garde computing groups coming from the industry and the academe. Various universities had been integrating parallel computing courses into its academic program, starting from the undergraduate degree towards the graduate level.**

Parallel computing per se is a broad topic, with many kinds of approaches to it. However, the main objective of this paper is to break down parallelism into some of its implementations and eventually coming up with an evaluation coming from industry and academic sources.

This paper gives an overview on why parallel computing is used, comparisons between its memory architectures and programming models, and addresses some issues coming from but not limited to the 1989 paper entitled "Parallel Computing – One Opportunity Four Challenges" [1]. That after 17 years, programmability within the parallel architecture remains an issue especially for the industry.

This is a midterm research paper for the UC-Berkeley CS252, a graduate computer architecture class, which deals greatly with parallel processing education.

I. INTRODUCTION

Exponential developments with respect to chip technology have been consistent for the past few decades. Computer performance has been driven largely by decreasing the size of chips while increasing the number of transistors they contain. In accordance with Moore's law, this has caused chip speeds to rise and prices to drop. This ongoing trend has driven much of the computing industry for years.

Traditionally, software had been written in a serial/sequential fashion. A problem is broken into a discrete series of instructions. Instructions are executed one after another and only one instruction may execute at any moment in time. Improvement in computer performance was implemented through clock rate ramping in order to provide faster execution of the instructions; somewhat equivalent to having better instruction throughput.

However, transistors can't shrink forever. Increasing clock speeds causes transistors to switch faster and thus generate more heat and consume more power. Even now, as transistor components grow thinner, chip manufacturers have struggled to cap power usage and heat generation, two critical problems. Current transistor technology limits the ability to continue making single processor cores more powerful.

The transition from serial to parallel computing is one in a way to extend Moore's law into getting more performance out of a piece of silicon.

II. PARALLEL COMPUTING

Parallel computer architecture had been a classic research topic for the past few decades. With the sudden sea change in computer trends, parallel computing now becomes one of the promising and interesting research thrusts geared into improving computational capability.

Parallel computing is basically a development from serial computing. [2]Parallel computing, in its simplest sense, is the simultaneous use of multiple compute resources to solve a computational problem. It is usually used in running multiple CPUs, enabling a problem broken down to discrete parts that it would be solved concurrently. Each part is further broken down to a series of instructions, and those instructions execute simultaneously on the different CPUs.

Why use parallel computing? Aside from the limitations set for the silicon process, parallelism is aimed at saving time and performance. Being able to perform multiple tasks at any moment clearly makes a difference. In the case of parallel supercomputers, the use of multiple cost-efficient processors to work on the same task is better than relying on a single yet expensive supercomputer.

The Fig. 1 clearly states that parallel computing is still in its research and development state (50% + 24%). Meanwhile, the industry (17%) is notably close in adapting it towards its commercialization, being evident in recent computer products brought about by major companies such as Intel and AMD.

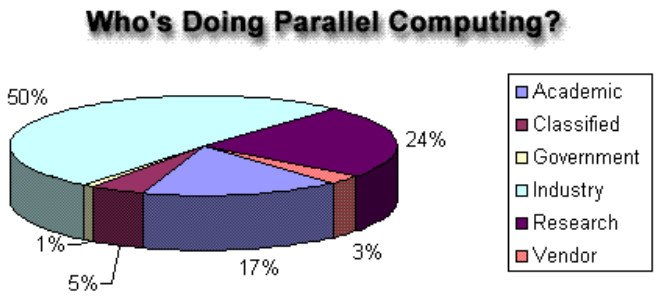

Fig. 1 Parallel Computing User Distribution [2]

Despite its growing popularity, there have always been debates regarding the different approaches towards optimal parallelism, which deals in both hardware and software design.

III. PARALLEL COMPUTER MEMORY ARCHITECTURES

The way processors communicate is dependent upon memory architecture, which in turn would affect the way one would write a parallel program.

2 primary approaches in computer memory architecture involve the shared and the distributed memory model. In a

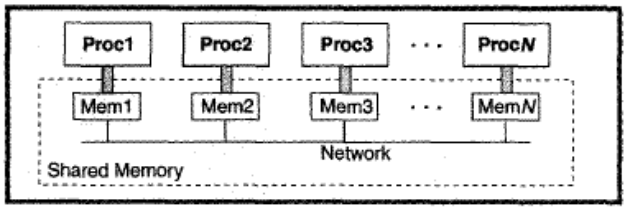

Fig. 2 Distributed shared memory: Each processor sees a shared address space, denoted by the dashed outline, rather than a collection of distributed address spaces.

shared memory, the use of a global physical memory equally shared among processors allows simple data to share through a uniform read and write mechanism into the memory. Problems arising with this model are usually due to increased contention and longer latencies, thus limits its performance and scalability. Examples of this are the Cray C-90 and Cray Y-MP.

Distributed memory architectures, on the other hand, have local memory modules interconnected through a network. Their scaleable nature makes it possible for higher computing power. However, problem lies on the communication between nodes leading to complex software problems. Examples in the industry are the IBM SP1 and SP2 and the Intel Paragon.

A relatively new concept involves the combining of the advantages of the two, the Distributed Shared Memory (DSM). DSM allows processes to assume globally shared virtual memory. [3]In Fig. 2, the distributed shared memory system consists of ***N*** networked workstations, each with its own memory. The distributed shared memory system provides the concept of a globally shared memory. Each processor can access any data items without the programmer having to worry about where the data is or how to obtain its value. Therefore the programmer does not have to specify inter processor communication which would be very complex especially for sophisticated parallelization strategies. The largest and fastest computers today employ the shared and distributed memory architecture.

IV. PARALLEL PROGRAMMING MODELS

There are various ways in programming parallel computers, the first two being the more popular models:

Message Passing - the user makes calls to libraries to explicitly share information between processors.

Data Parallel - data partitioning determines parallelism

Shared Memory - multiple processes sharing common memory space.

Remote Memory Operation - set of processes in which a process can access the memory of another process without its participation.

Threads - a single process having multiple (concurrent) execution paths.

Combined Models - composed of two or more of the above.

These models are machine/architecture independent; any of the models can be implemented on any hardware given appropriate operating system support. An effective implementation is one which closely matches its target hardware and provides the user ease in programming.

V. THE FOUR CHALLENGES

In Gaudiot's[1] paper published in 1989, parallel computing had been facing four challenges towards its implementation. 17 years later, with the due recognition of parallel computing in today's systems, the four challenges are still impending.

(a) Programmability

- This is the foremost challenge besetting parallel computing ever since. As intrinisically discussed in previous sections, there have been many approaches in hardware designs to have an ideal parallel computer, but the problem lies on how the programmer may be able to formulate his own program with the hardware.

One major bottleneck for parallel programming is that most people still have less experience in parallel computing. People developing parallel systems software are similarly behind on their learning curves. Often they reuse material developed for serial systems, even when it causes performance problems.

(b) Communication Network Design

- For distributed memory systems, message passing communication plays a vital role in linking all the distributed units by using different protocols. Hardware considerations limit connectivity among systems. More or less, we are still using the same communication systems in our devices. New technologies in interconnection such as that of fiber optic communications have given a slight progress but still are not yet widely used in the industry. The industry's current focus lies on multi-core chips, which has the advantage of having better signal communications due to shorter traveling paths.

(c) Reliable Operation

- Parallel computing, being in the limelight of academic computing research and development, has through all these years been studying different fault-tolerant algorithms that could help in the improvement of reliability for computers. The availability of the more developed industry standard EDA tools help in the automation of the design processes especially on larger machines, even on small chips.

(d) Performance Evaluation and Benchmarking

- There are already many benchmarking standards that can be used to test various systems. There are already some companies dedicated in creating such benchmarking products in which they aim to establish a comprehensive set of parallel benchmarks that would be generally accepted by both users and the vendors of the systems.

VI. CONCLUSION

With the wall blocking the exponential path of Moore's law, parallel computing so far is the only viable option to extend it further till exhaustion. Some other options may be available but they are not as tried and tested such as that of parallel computing, which already had been present in the computing era for decades. For now, parallel computing is undergoing a major transition. What parallel computing needs is the support of the software industry for as to make parallel computers useful. Due to the everyday demand of higher computational capability, there will be plenty of applications that can take advantage of parallelism, like that of real-time monitoring programs such as the weather forecasting system.

As programmability has been the major problem despite the ongoing popularity of parallel computing, various universities had been stepping up in improving parallel processing education in order to achieve improvement in the programmer level. The continued exponential throughput advances of new processors should also be complemented by applications that can take advantage of parallelism.

As for the current Intel vs. AMD competition, coming from clock rate ramping, the trend now leads onto whoever gives the best performance out of multicore processors. The commercial releases may have impressive results from commercial benchmarks, but the significance of it in everyday use is not yet well established due to the lack of programs that take advantage of parallel technology.

One reason why independent software vendors still refuse to program for parallel architectures is that there is still no industry standard for it. As enumerated in some sections of this paper, there are about countless ways on how to implement a parallel computer architecture. Independent software vendors would likely adopt parallelism once parallel programming interfaces become industry standard.

From this, we are coming to realize that parallel computing becomes a software problem. The industry needs software and standards to make parallel programs easier to write. Once the software issue had been properly addressed, the whole parallel computing industry would grow and would result to the ideal setup of higher computational capability among computers.

REFERENCES

[1] J.L. Gaudiot, "Parallel Computing – One Opportunity, Four Challenges," IEEE, 1989

[2] Introduction to Parallel Computing, "http://www.llnl.gov/computing/tutorials/parallel_comp/"

[3] M.A. Ali and I.A.K. Khel, "Parallel Computing for Less", *IEEE Potentials*, pp.33-35 April/May 1999

[4] T.K. Hazra, "Parallel Computing", *IEEE Potentials*, pp.17-20 August/September 1995

[5] E. Hagersten and G. Papadopoulos, "Parallel Computing in the Commercial Workspace", *Proceedings of the IEEE*, Vol.87, No.3, March 1999

[6] http://www.crpc.rice.edu/newsletters/oct94/director.html

[7] Herb Sutter, "The Free Lunch is Over: A Fundamental Turn Toward Concurrency in Software", *Dr. Dobb's Journal*, March 2005

[8] http://www.eecs.umich.edu/~qstout/parallel.html

[9] http://computer-engineering.science-tips.org/parallel-computing/fundamentals/on-parallel-computing.html

[10] http://www.mhpcc.edu/training/workshop/parallel_intro/MAIN.html#memory%20architectures

[11] Silva and Buyya, "Parallel Programming Models and Paradigms", *Monash University, Melbourne, Australia*