

## Chapter 2

# Multi-Threaded Architectures: Evolution, Costs, Opportunities

**Ivan Girotto**

*National University of Ireland Galway, Republic of Ireland*

**Robert M. Farber**

*Pacific Northwest National Laboratory, USA*

### ABSTRACT

*This chapter focuses on the technical/commercial dynamics of multi-threaded hardware architecture development, including a cost/benefit account of current and future developments, and the implications for scientific practice.*

### INTRODUCTION

Recently, the computer industry has moved *en masse* to parallel architectures. Computing technologies are commonly based on multi- and many-core systems with tens to hundreds of concurrent hardware processing elements on workstations up to many thousands per data servers or super-computer node.

This powerful computing capacity represents an extraordinary opportunity to speed up both current and further software application. Current parallel hardware from commodity to the newest leadership class supercomputers can provide from

one-order to seven-orders of magnitude increases in performance over single-core processors. Affordable general-purpose graphic processors technology, sold at price points ranging from a hundred to a couple thousand dollars per board, demonstrates performance improvements ranging from one-order to three-orders of magnitude on a wide-range of applications in the scientific literature.

This mass adoption profoundly affects every aspect of computation-based projects (be they new or based on legacy software) including investment, planning, development, procurement, and deployment.

Current software development tools demonstrate that it is possible to program these massively

DOI: 10.4018/978-1-61350-116-0.ch002

parallel systems on a wide-spectrum of problems in high-level languages to gain outstanding performance. However, the gap between hardware and software trajectories tends to grow. Good times were when software application performance was directly related to new hardware generation with higher clock frequency. For assessing how well utilized the processors are as more parallel hardware becomes available efficiency, is a key metric that defines the performance of both the algorithms and software implementation.

The trend to massive parallelism appears inescapable and massive threaded software is a clear requirement to achieve high performance. Consumers, developers, scientists, managers and organizations need to understand that single-threaded (serial) or poorly scaling multi-threaded software will cause application performance to plateau at or near current levels on both current *and* future hardware. This has important implications for computation-dependent projects because it defines the limits of the computing capacity and impacts on both product and project competitiveness relative to other computational-based approaches. However, the cost associated with re-engineering software (and potentially re-design of algorithms) to capitalize on new parallel architectures must be considered along with applications scalability and lifespan. In general, owners of legacy software are more likely to require new software and/or software development because a significant amount of existing commercial and scientific software was developed for single-threaded processors – much of it prior to the general availability of multi-core hardware.

HPC can act as a conduit for disruptive new technologies by acting as an early adopter and proving ground for hardware and software models that radically transform both consumer and business computing market spaces. The innovations that improve performance are not always expected and while they reduce cost, improve performance and create new opportunities, they can also damage existing markets and deprecate

current applications and software. This is not a random process, but rather is part of an evolutionary process driven by competition; limited by the inefficiencies of electrical components and manufacturing processes; and advanced through scientific and design ingenuity.

This chapter will explore the disruptive nature of current innovations in massively threaded architectures, beginning with the demise of faster clock speeds and how that caused a renaissance in massive parallelism. It considers important architectures including GPGPU (general-purpose graphic processors units); specialized architectures such as the Cray XMT; extreme scale computing; and hybrid systems, examining these in terms of cost impacts on budgets and also on power and performance. Finally, future opportunities will be discussed.

## Background

Commodity multi-core laptops and workstations are now common and many-core processors (chips with tens to hundreds of processing cores) are starting to be delivered at high-end price points. Manufacturers such as NVIDIA and ATI have been shipping teraflop-capable ( $10^{12}$  floating-point operations per second) GPGPUs, at commodity price-points that most students and scientists can afford. As an outcome of this technology shift, today's multi-GPUs are generally plugged onto multi-cores computer boards and installed on commodity workstations located on the student's or senior researcher's desks, as well as on very large supercomputers. As of the November 2010 TOP500 list, the first and third fastest supercomputers in the world (China's Nebulae and Tehane-1 hybrid CPU/GPU supercomputers) provide examples of this developing trend.

Specialized architectures such as the Cray XMT have been designed to provide a massively parallel environment to accelerate irregular, memory-access dominated problems. This is in contrast to GPGPU and multi-core architectures

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/multi-threaded-architectures/60354](http://www.igi-global.com/chapter/multi-threaded-architectures/60354)

## Related Content

---

### Women in Computer Science in Afghanistan

Eva Maria Hoffmann (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 1840-1854).

[www.irma-international.org/chapter/women-computer-science-afghanistan/62548](http://www.irma-international.org/chapter/women-computer-science-afghanistan/62548)

### Digital Home: A Case Study Approach to Teaching Software Engineering Concepts

Salamah Salamah, Massood Towhidnejad and Thomas Hilburn (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 1284-1299).

[www.irma-international.org/chapter/digital-home/192923](http://www.irma-international.org/chapter/digital-home/192923)

### Managing Risk in Cloud Computing

Lawan A. Mohammed (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 1318-1333).

[www.irma-international.org/chapter/managing-risk-in-cloud-computing/203562](http://www.irma-international.org/chapter/managing-risk-in-cloud-computing/203562)

### Streaming Coded Video in P2P Networks

Muhammad Salman Raheel and Raad Raad (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming* (pp. 1304-1339).

[www.irma-international.org/chapter/streaming-coded-video-in-p2p-networks/261080](http://www.irma-international.org/chapter/streaming-coded-video-in-p2p-networks/261080)

### Fuzzy Linear Multi-Objective Stochastic Programming Models

(2019). *Multi-Objective Stochastic Programming in Fuzzy Environments* (pp. 78-127).

[www.irma-international.org/chapter/fuzzy-linear-multi-objective-stochastic-programming-models/223803](http://www.irma-international.org/chapter/fuzzy-linear-multi-objective-stochastic-programming-models/223803)