Multi-core Processors – A Necessity

Vikash Rai, Tapish Chauhan
Information Technology
Dronacharya College Of Engineering

Abstract- As personal computers have become more prevalent and more applications have been designed for them, the end-user has seen the need for a faster, more capable system to keep up. Speedup has been achieved by increasing clock speeds and, more recently, adding multiple processing cores to the same chip. Although chip speed has increased exponentially over the years, that time is ending and manufacturers have shifted toward multicore processing. However, by increasing the number of cores on a single chip challenges arise with memory and cache coherence as well as communication between the cores. Coherence protocols and interconnection networks have resolved some issues, but until programmers learn to write parallel applications, the full benefit and efficiency of multicore processors will not be attained.

I. INTRODUCTION

Parallel processors have had a long history going back at leastto the Solomon computer of the mid-1960s. The difficulty ofprogramming them meant they have been primarily employedby scientists and engineers who understood the applicationdomain and had the resources and skill to program them.
Along the way, a surprising number of companies created parallelmachines. They were largely unsuccessful since their difficultyof use limited their customer base, although, there wereexceptions: the Cray vector machines are perhaps the bestexample. However, these Cray machines also had a very fast scalar processor that could be easily programmed in a conventional manner and the vector programming paradigm was notas daunting as creating general parallel programs. Recentlythe evolutionof parallel machines has changed dramatically. For the first time, major chip manufacturers—companieswhose primary business is fabricating and selling microprocessors—have turned to offering parallel machines, or single chipmulticore microprocessors as they have been styled. There are a number of reasonsbehind this, but the leading oneis to continue the raw performance growth that customers havecome to expect from Moore's law scaling without being overwhelmedby thegrowth in power consumption.

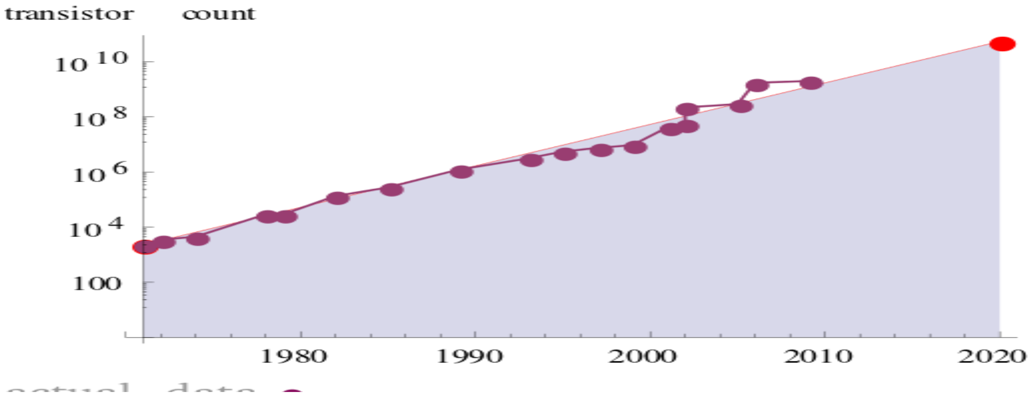
II. THE NEED OF MULTIPROSSESING

Mobile devices perform a wide variety of tasks such as Web browsing, video playback, mobile
Gaming, SMS text messaging, and location-based services. Due to the growth in the availability
of high speed mobile and Wi-Fi networks, mobile devices will also be used for various
performance-intensive tasks that were previously handled by traditional PCs. The next
generation of smartphones (called "Super phones") and tablets will be used for a wide variety of
tasks such as playback of high definition 1080p videos, Adobe Flash-based online gaming,
Flash-based streaming high definition videos, visually rich gaming, video editing, simultaneous
HD video downloads, encode and uploads, and real-time HD video conferencing.
The current generation of mobile processors is not designed to deal with this tidal wave of high
Performance use cases. The quality of experience on devices based on single core CPUs
Rapidly degrades when users run several applications concurrently, or run performanceintensive applications such as games, video conferencing, video editing, and more. In order to
Improve CPU performance, engineers employ several techniques, such as using faster and
Smaller semiconductor processes, increasing core operating frequency and voltage, using larger cores, and using larger on-die caches.

III. MOORE'S LAW

One of the guiding principles of computer architecture is known as Moore's Law. In 1965 Gordon Moore stated that the numbers of transistors on a chip will roughly double each year (he later refined this, in

1975, to every two years). What is often quoted as Moore's Law is Dave House's revision that computer performances will double every 18 months.



Multicore Challenges

Having multiple cores on a single chip gives rise to some problems and challenges. Power and temperature management are two concerns that can increase exponentially with the addition of multiple cores. Memory/cache coherence is another challenge, since all designs discussed above have distributed L1 and in some cases L2 caches which must be coordinated. And finally, using a multicore processor to its full potential is another issue. If programmers don't write applications that take advantage of multiple cores there is no gain, and in some cases there is a loss of performance. Application need to be written so that different parts can be run concurrently (without any ties to another part of the application that is being run simultaneously).

IV. CACHE COHERENCE

Cache coherence is a concern in a multicore environment because of distributed L1 and L2 cache. Since each core has its own cache, the copy of the data in that cache may not always be the most up-to-date version. For example, imagine a dual-core processor where each core brought a block of memory into its private cache. One core writes a value to a specific location; when the second core attempts to read that value from its cache it won't have the updated copy unless its cache entry is invalidated and a cache miss occurs. This cache miss forces the second core's cache entry to be updated. If this coherence policy wasn't in place garbage data would be read and invalid results would be produced, possibly crashing the program or the entire computer.

V. MULTITHREADING

The last, and most important, issue is using multi-threading or other parallel processing techniques to get the most performance out of the multicore processor. "With the possible exception of Java, there are no widely used commercial development languages with [multithreaded] ex-tensions."

VI. CONCLUSION

Before multicore processors the performance increase from generation to generation was easy to see, an increase in frequency. This model broke when the high frequencies caused processors to run at speeds that caused increased power consumption and heat dissipation at detrimental levels.Adding multiple cores within a processor gave the solution of running at lower frequencies, but added interesting new problems.Multicore processors are architected to adhere to reasonable power consumption, heat dissipation, and cache coherence protocols. However, many issues remain unsolved.In order to use a multicore processor at full capacity the applications run on thesystem must be multithreaded. There are relatively few applications written with anylevel of parallelism. Thememorysystems and interconnection networks also need improvement. And finally it is still unclear whether homogenous or heterogeneous cores are more efficient.

REFERENCES

[1] W. Knight, "Two Heads Are Better Than One", IEEE Review

[2] R. Merritt, "CPU Designers Debate Multi-core Future", EETimes Online http://www.eetimes.com/showArticle.jhtml?articleID=206105179

[3] P. Frost Gorder, “Multicore Processors for Science and Engineering”, IEEE CS
[4] D. Geer, “Chip Makers Turn to Multicore Processors”, Computer, IEEE Computer
Society.
[5] L. Peng et al, “Memory Performance and Scalability of Intel’s and AMD‟s Dual-Core
Processors: A Case Study”, IEEE.
[6] D. Pham et al, “The Design and Implementation of a First-Generation CELL Processor”,
ISSCC
[7] P. Hofstee and M. Day, “Hardware and Software Architecture for the CELL Processor”,
CODES+ISSS.
[8] J. Kahle, “The Cell Processor Architecture”, MICRO-38 Keynote.
[9] D. Stasiak et al, “Cell Processor Low-Power Design Methodology”, IEEE MICRO.
[10] D. Pham et al, “Overview of the Architecture, Circuit Design, and Physical
Implementation of a First-Generation Cell Processor”, IEEE Journal of Solid-State
Circuits.