

Multicore Programming Challenges

Michael Perrone^{*}

IBM, TJ Watson Research Lab
Yorktown Heights, NY, USA
mpp@us.ibm.com

Abstract. The computer industry is facing fundamental challenges that are driving a major change in the design of computer processors. Due to restrictions imposed by quantum physics, one historical path to higher computer processor performance - by increased clock frequency - has come to an end. Increasing clock frequency now leads to power consumption costs that are too high to justify. As a result, we have seen in recent years that the processor frequencies have peaked and are receding from their high point. At the same time, competitive market conditions are giving business advantage to those companies that can field new streaming applications, handle larger data sets, and update their models to market conditions faster. The desire for newer, faster and larger is driving continued demand for higher computer performance.

The industry's response to address these challenges has been to embrace "multicore" technology by designing processors that have multiple processing cores on each silicon chip. Increasing the number of cores per chip has enable processor peak performance to double with each doubling of the number of cores. With performance doubling occurring at approximately constant clock frequency so that energy costs can be controlled, multicore technology is poised to deliver the performance users need for their next generation applications while at the same time reducing total cost of ownership per FLOP.

The multicore solution to the clock frequency problem comes at a cost: Performance scaling on multicore is generally sub-linear and frequently decreases beyond some number of cores. For a variety of technical reasons, off-chip bandwidth is not increasing as fast as the number of cores per chip which is making memory and communication bottlenecks the main barrier to improved performance. What these bottlenecks mean to multicore users is that precise and flexible control of data flows will be crucial to achieving high performance. Simple mappings of their existing algorithms to multicore will not result in the naive performance scaling one might expect from increasing the number of cores per chip. Algorithmic changes, in many cases major, will have to be made to get value out of multicore. Multicore users will have to re-think and in many cases re-write their applications if they want to achieve high performance. Multicore forces each programmer to become a parallel programmer; to think of their chips as clusters; and to deal with the issues of communication, synchronization, data transfer and nondeterminism as integral elements

^{*} Invited Speaker.

of their algorithms. And for those already familiar with parallel programming, multicore processors add a new level of parallelism and additional layers of complexity.

This talk will highlight some of the challenges that need to be overcome in order to get better performance scaling on multicore, and will suggest some solutions.