CMSC 180

August 12 2015

Parallelism - Speed-up to beat a deadline.

Motivating Parallelism

> The role of parallelism in accelerating computing speeds has been recognized for several decades.

> Its role in providding multiplicity of datapaths and increased access to storage elements has been significant in commercial applications.

> The scalable performance and lower cost of parallel platforms is reflected in the wide variety of applications.

> Developing parallel hardware and software has traditionally been time and effort extensive.

> If one is to view this in the context of rapidly improving uniprocessor speeds, one is tempted to question the need for parallel compunting.

> There are som unmistakable trends in hardware design, which indicate that uniprocessor (or implicitly parallel) architectures may not be able to sustain the rate of *realizable* performance increments in the future. (there are inherently parallelized processes and inherently serializeable processes)

> This is the result of a number of fundamental physical and computational limitations.

> The emergence of standardized parallel programming environments, libraries, and hardware have significantly reduced time to (parallel) solution.

The Computational Power Argument

> Moore's Law:

There is no room left to sqeeze anything out by being clever. Going forward from here, we have to depend on the two size factors - bigger dies and finer dimensions.

> If one is to buy into Moore's law, the question still remains - how does one translate transistors into useful OPS (operations per second)

> The logical recourse is to rely on parallellism, both implicit and explicit.

> Most serial (or seemingly serial) processors rely extensively on implicit parallelism.

> We focus in this class, for the most part, on explicit parallelism.

The Memory/Disk Speed Argument

> While clocl rates of high-end processors have increased at roughly 40% per year over the past decade, DRAM access times have only improved at the rate of roughly 10% per year over this interval.

> This mismatch in speeds causes significant performance bottlenecks

> Parallel platforms provide increased bandwidth to the memory system.

> Parallel platforms also provide higher aggregate caches

> Principles of locality of data reference and bulk access, which guide parallel algorithm design also apply to memory optimization.

> Some of the fastest growing applications of parallel computing utilize not their raw computational speed, rather their ability to pump data to memory and disk faster.

The Data Communication Argument

> As the network evolves, the vision of the internet as one large computing platform has emerged.

> This view is exploited bu applications such as SETI@home and Folding@home

> In many other applications (typically databases and data mining) the volume of data is such that they cannot be moved.

> Any analyses on the data must be performed over the network using parallel techniques.

Scope of Parallel Computing Applications

> Parallelism finds applications in very diverse application domains for didfferent motivating reasons.

> These range from improved application performance to cost considerations.

Applications in Engineering and Design

> Design of airfoils, inernal combustion engines, high-speed circuits, and structures.

> Design and simulation of micro- and nano-scale systems.

> Process optimization, operations research.

Scientific Applications

> Functional and structural characterization of genes and proteins.

> Advances in computational physics and chemistry have explored new materials, understanding of chemical pathways, and more efficient processes.

> Applications in astrophysics have explored the evolution of galaxies, thermonuclear processes, and the analysis of extremely large datasets from telescopes.

> Weather modeling, mineral prospecting, flood prediction, etc., are other important applications.

> Bioinformatics and astrophysics also present some of the most challenging problems with respect to analyzing extremely large datasets.

*Morton Ordering, Snake Ordering, Row-Major, Column-Major

Application in Compuer Systems

> Network intrustion detectoin, cryptography, multiparty computations are some of the core usersof parallel computing techniques.

> Embedded systems increasingly rely on distributed computing.

Organization and Contents of this Course

> Fundamentals - Parallel platforms, principles of algorithm design, group comm primitives, and analytical modeling techniques.

> Parallel Programming - This part of the class deals with programming using message passing libraries and threads.

> Parallel Algorithms - This part of the class covers basic algorithms for matrix computations, graphs, sorting, discrete optimization, and dynamic programming.