High-Performance Computing
An Applications Perspective

sunil.d.sherlekar@intel.com

REACH-IIT Kanpur

10th Oct. 2010

HPC in Science & Technology

- “Theory” and “Experiment” have been two traditional pillars of scientific research; “Simulation” has been added as the third pillar
- To do for other engineering what VLSI has achieved
- Abstraction layers missing: hence HPC
- Non-linearity & quantum mechanics: hence HPC

Some HPC Applications

- Fluid dynamics:
  - Aerospace, Automobile, chemical & power plants
- Structural mechanics:
  - Crash simulation, armour design
- Signal processing:
  - Oil exploration, astronomy, multimedia data mining
- Quantum mechanics, MD
  - Nanotechnology, computational biology

The Computing Crisis: Opportunity Without A Choice!

- Processors not getting any faster
- Hence proliferation of multicores
- Computation & communication: speed mismatch
  - Processor, server and cluster levels
- What do you with so many cores anyway?
  - There is a potential market: opportunity
  - Need to program all these cores: challenge
- Mainstreaming/Democratisation of HPC?

Abstractions

- Important for handling complexity
  - Limitation of the human mind
  - Lemmas leading to theorems
  - The Clocking Abstraction: Synchronous Ckts.
  - Decoupling of concerns
- Need to break them for efficiency
- Breaking needs to be creative & selective to avoid chaos
- May lead to a new (better) abstraction

Abstraction: Examples

- Physics & Chemistry:
  - QM, Classical MD, Continuum
- The Computing Stack:
  - Polygons, transistors, gates, ALU/Register, ISP
- The Communications Stack:
  - PHY, MAC, ...
- Memory Hierarchy:
  - Cache, Virtual Memory
- Management:
  - Organisation Structure, Hierarchy

The Unravelling of Abstractions

- Layout design
  - Diffraction: rectangles get fuzzy
- Circuit design
  - All transistors are not the same
- Logic design
  - Wire delay dominates switching delay
- Processor design
  - Clocking hits walls: skew and power
- Software
  - Need abstract model of processor, memory, interconnect, ...

Computing: Correctness & Performance

- Software abstractions mainly address correctness
  - True for both sequential & parallel programming
- Efficiency issues: optimising compilers
  - Largely confined to sequential programs
- Algorithms people focussed on order-of-magnitude analysis
  - Makes the Turing machine model suitable
- Need exact analysis a la Knuth

Computing: Correctness & Performance

- Need a “realistic” abstraction of hardware
  - Must meet the need of problem abstraction
- Processor: need to model the pipeline
  - MAC throughput different from latency
- Memory: need to model access timing
  - Access pattern dependent: again throughput v/s latency
  - RAM is not exactly “random” access
  - Cache transparency a performance problem
  - Cache coherence is yet another problem

Computing: Correctness & Performance

- Need to model interprocessor communication
  - Between blades on a cluster
  - Between dies on a blade
  - Between cores on a die
- Physical interconnect delays
- Interconnect topologies
- Interconnect protocols: retransmissions etc.
  - Now even on a chip!

The Productivity Challenge

- Programmers need to learn the “new” performance-aware models
- Programs must match these models
  - Not Turing Machine equivalents
  - Not even idealised parallel computing machines
- Phase I:
  - Write parallel programs for idealised parallel machines
  - Map programs onto real architectures: “machine-dependent” optimisation
  - Possible to (at least partially) automate mapping
- Phase II: Develop algorithms for real models

The Applications Perspective

- Ideal programming paradigms are application-domain dependent
- Most architecture-mapping problems best solved by exploiting application characteristics
- Characterisation by 13 dwarfs (earlier 7 dwarfs)

- “Black Magic” must graduate to methodology
  - Education & training is the key
- Methodology should graduate to automation
  - Libraries
  - Parallelising compilers: for real machine models

A Made-For-India Opportunity

- Dropping hardware costs: potentially exploding market size
- Large workforce adept at programming
- Challenge: Education for the “new” world of *performance-driven* parallel computation
- A case for a whole program in “Computational Science”
  - A great opportunity for “IISER-Class” science graduates

Thank You!

Questions?