The Lecture Contains:

Parallel Computer Architecture: Today and Tomorrow

- What is computer architecture ?
- Architect’s job
- 58% growth rate
- The computer market
- The applications
- Parallel architecture
- Why parallel arch.?
- Why study it?
- Performance metrics
- Throughput metrics
- Application trends
- Commercial sector
- Desktop market

[From Chapter 1 of Culler, Singh, Gupta]

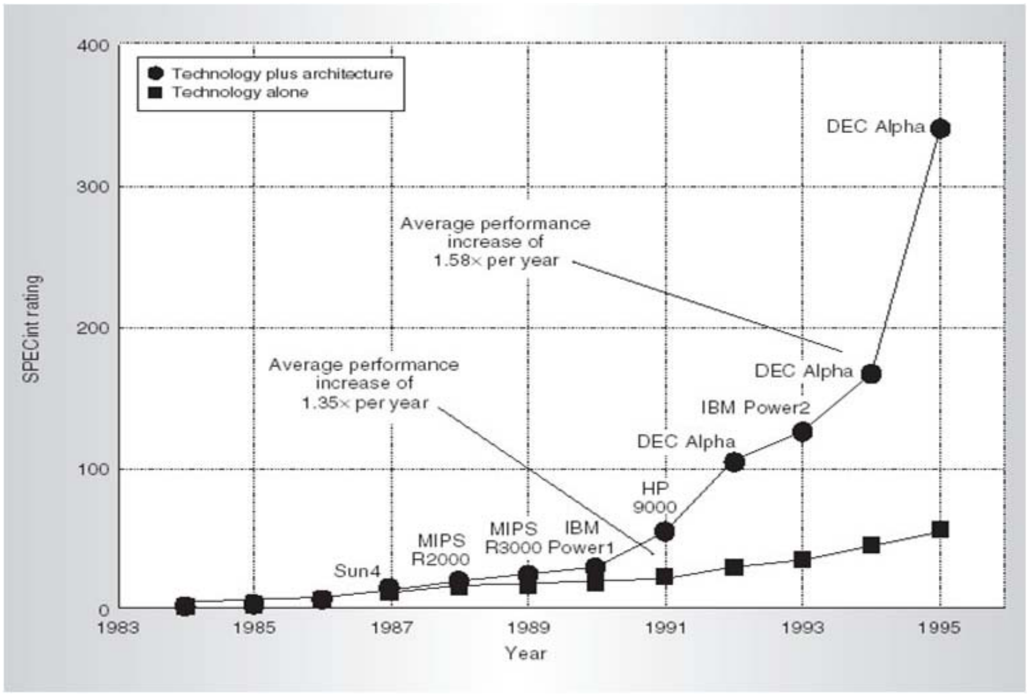
Previous Next

What is computer architecture ?

- Amdahl, Blaauw and Brookes, 1964 (IBM 360 team):
  - The structure of a computer that a machine language programmer must understand to write a correct (timing independent) program for that machine
- Loosely speaking, it is the science of designing computers "leading to glorious failures and some notable successes"

Architect's job

- Design and engineer various parts of a computer system to maximize performance and programmability within the technology limits and cost budget
- Technology limit could mean process/circuit technology in case of microprocessor architecture
- For bigger systems technology limit could mean interconnect technology (how one component talks to another at macro level)



Slightly outdated data

58% growth rate

- Two major architectural reasons
  - Advent of RISC (Reduced Instruction Set Computer) made it easy to implement many aggressive architectural techniques for extracting parallelism
  - Introduction of caches
- Made easy by Moore's law
- Two major impacts
  - Highest performance microprocessors today outperform supercomputers designed less than 10 years ago
  - Microprocessor-based products have dominated all sectors of computing: desktops, workstations, minicomputers are replaced by servers, mainframes are replaced by

multiprocessors, supercomputers are built out of commodity microprocessors (also a cost factor dictated this trend)

Previous Next

The computer market

- Three major sectors
  - Desktop: ranges from low-end PCs to high-end workstations; market trend is very sensitive to price-performance ratio
  - Server: used in large-scale computing or service-oriented market such as heavy-weight scientific computing, databases, web services, etc; reliability, availability and scalability are very important; servers are normally designed for high throughput
  - Embedded: fast growing sector; very price-sensitive; present in most day-to-day appliances such as microwave ovens, washing machines, printers, network switches, palmtops, cell phones, smart cards, game engines; software is usually specialized/tuned for one particular system

The applications

- Very different in three sectors
  - This difference is the main reason for different design styles in these three areas
  - Desktop market demands leading-edge microprocessors, high-performance graphics engines; must offer balanced performance for a wide range of applications; customers are happy to spend a reasonable amount of money for high performance i.e. the metric is price-performance
  - Server market integrates high-end microprocessors into scalable multiprocessors; throughput is very important; could be floating-point or graphics or transaction throughput
  - Embedded market adopts high-end microprocessor techniques paying immense attention to low price and low power; processors are either general purpose (to some extent) or application-specific

Parallel architecture

- Collection of processing elements that co-operate to solve large problems fast
- Design questions that need to be answered
  - How many processing elements (scalability)?
  - How capable is each processor (computing power)?
  - How to address memory (shared or distributed)?
  - How much addressable memory (address bit allocation)?
  - How do the processors communicate (through memory or by messages)?
  - How do the processors avoid data races (synchronization)?
  - How do you answer all these to achieve highest performance within your cost envelope?

Previous Next

Why parallel arch.?

- Parallelism helps
- There are applications that can be parallelized easily
- There are important applications that require enormous amount of computation (10 GFLOPS to 1 TFLOPS)
  - NASA taps SGI, Intel for Supercomputers: 20 512p SGI Altix using Itanium 2 (http://zdnet.com.com/2100-1103_2-5286156.html) [27th July, 2004]
- There are important applications that need to deliver high throughput

Why study it?

- Parallelism is ubiquitous
  - Need to understand the design trade-offs
  - Microprocessors are now multiprocessors (more later)
  - Today a computer architect's primary job is to find out how to efficiently extract parallelism
- Get involved in interesting research projects
  - Make an impact
  - Shape the future development
  - Have fun

Performance metrics

- Need benchmark applications
  - SPLASH (Stanford ParalleL Applications for SHared memory)
  - SPEC (Standard Performance Evaluation Corp.) OMP
  - ScaLAPACK (Scalable Linear Algebra PACKage) for message-passing machines
  - TPC (Transaction Processing Performance Council) for database/transaction processing performance
  - NAS (Numerical Aerodynamic Simulation) for aerophysics applications
    - NPB2 port to MPI for message-passing only
  - PARKBENCH (PARallel Kernels and BENCHmarks) for message-passing only
- Comparing two different parallel computers
  - Execution time is the most reliable metric
  - Sometimes MFLOPS, GFLOPS, TFLOPS are used, but could be misleading
- Evaluating a particular machine
  - Use speedup to gauge scalability of the machine (provided the application itself scales)
  - Speedup(P) = Uniprocessor time/Time on P processors
  - Normally the input data set is kept constant when measuring speedup

Previous Next

Throughput metrics

- Sometimes metrics like jobs/hour may be more important than just the turn-around time of a job
  - This is the case for transaction processing (the biggest commercial application for servers)
  - Needs to serve as many transactions as possible in a given time provided time per transaction is reasonable
  - Transactions are largely independent; so throw in as many hardware threads as possible
  - Known as throughput computing

Application trends

- Equal to or below 1 GFLOPS requirements
  - 2D airfoil, oil reservoir modeling, 3D plasma modeling, 48-hour weather
- Below 100 GFLOPS requirements
  - Chemical dynamics, structural biology, 72-hour weather
- Tomorrow's applications (beyond 100 GFLOPS)
  - Human genome, protein folding, superconductor modeling, quantum chromodynamics, molecular geometry, real-time vision and speech recognition, graphics, CAD, space exploration, global-warming etc.
- Demand for insatiable CPU cycles (need large-scale supercomputers)

Commercial sector

- Slightly different story
  - Transactions per minute (tpm)
- Scale of computers is much smaller
  - 4P machines to maybe 32P servers
- But use of parallelism is tremendous
  - Need to serve as many transaction threads as possible (maximize the number of database users)
  - Need to handle large data footprint and offer massive parallelism (also economics kicks in: should be low-cost)

Desktop market

- Demand to improve throughput for sequential multi-programmed workload
  - I want to run as many simulations as I can and want them to finish before I come back next morning
  - Possibly the biggest application for small-scale multiprocessors (e.g. 2 or 4-way SMPs)
- Even on a uniprocessor machine I would be happy if I could play AOE without affecting the performance of my simulation running in background (simultaneous multi-threading and chip multi-processing; more later)

Previous Next