Percentage of peak

- A rule of thumb that often applies
  - A contemporary processor, for a spectrum of applications, delivers (i.e., sustains) 10% of peak performance
- There are exceptions to this rule, in both directions
- Why such low efficiency?
- There are two primary reasons behind the disappointing percentage of peak
  - IPC (in)efficiency
  - Memory (in)efficiency
IPC

- Today the theoretical IPC (instructions per cycle) is 4 in most contemporary RISC processors (6 in Itanium)
- Detailed analysis for a spectrum of applications indicates that the average IPC is 1.2–1.4
- We are leaving ~75% of the possible performance on the table...

Why Fast Machines Run Slow

- Latency
  - Waiting for access to memory or other parts of the system
- Overhead
  - Extra work that has to be done to manage program concurrency and parallel resources the real work you want to perform
- Starvation
  - Not enough work to do due to insufficient parallelism or poor load balancing among distributed resources
- Contention
  - Delays due to fighting over what task gets to use a shared resource next. Network bandwidth is a major constraint.
Latency in a Single System

<table>
<thead>
<tr>
<th>Year</th>
<th>Memory Access Time</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>1999</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>2001</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>2003</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>2006</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>2009</td>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>

**X-Axis:** CPU Clock Period (ns)  
**Ratio:** Memory System Access Time

**THE WALL**

Memory hierarchy

- **Typical latencies for today’s technology**

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Processor clocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>1</td>
</tr>
<tr>
<td>L1 cache</td>
<td>2-3</td>
</tr>
<tr>
<td>L2 cache</td>
<td>6-12</td>
</tr>
<tr>
<td>L3 cache</td>
<td>14-40</td>
</tr>
<tr>
<td>Near memory</td>
<td>100-300</td>
</tr>
<tr>
<td>Far memory</td>
<td>300-900</td>
</tr>
<tr>
<td>Remote memory</td>
<td>$O(10^3)$</td>
</tr>
<tr>
<td>Message-passing</td>
<td>$O(10^3)$-$O(10^4)$</td>
</tr>
</tbody>
</table>
Memory Hierarchy

- Most programs have a high degree of locality in their accesses
  - spatial locality: accessing things nearby previous accesses
  - temporal locality: reusing an item that was previously accessed
- Memory hierarchy tries to exploit locality

![Memory Hierarchy Diagram]

<table>
<thead>
<tr>
<th>Speed</th>
<th>1ns</th>
<th>10ns</th>
<th>100ns</th>
<th>10ms</th>
<th>10sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>B</td>
<td>KB</td>
<td>MB</td>
<td>GB</td>
<td>TB</td>
</tr>
</tbody>
</table>

Memory bandwidth

- To provide bandwidth to the processor, the bus either needs to be faster or wider
- Busses are limited to perhaps 400-800 MHz
- Links are faster
  - Single-ended 0.5–1 GT/s
  - Differential: 2.5–5.0 (future) GT/s
  - Increased link frequencies increase error rates requiring coding and redundancy thus increasing power and die size and not helping bandwidth
- Making things wider requires pin-out (Si real estate) and power
  - Both power and pin-out are serious issue
Processor-DRAM Memory Gap

"Moore's Law"

μProc
60%/yr.
(2X/1.5yr)

Processor-Memory Performance Gap:
(grows 50% / year)

DRAM
9%/yr.
(2X/10 yrs)

Processor in Memory (PIM)

- **PIM merges logic with memory**
  - Wide ALUs next to the row buffer
  - Optimized for memory throughput, not ALU utilization

- **PIM has the potential of riding Moore's law while**
  - greatly increasing effective memory bandwidth,
  - providing many more concurrent execution threads,
  - reducing latency,
  - reducing power, and
  - increasing overall system efficiency

- **It may also simplify programming and system design**
Internet – 4th Revolution in Telecommunications

- Telephone, Radio, Television
- Growth in Internet outstrips the others
- Exponential growth since 1985
- Traffic doubles every 100 days

Peer to Peer Computing

- Peer-to-peer is a style of networking in which a group of computers communicate directly with each other.
- Wireless communication
- Home computer in the utility room, next to the water heater and furnace.
- Web tablets
- BitTorrent
- Imbedded computers in things all tied together.
  - Books, furniture, milk cartons, etc
- Smart Appliances
  - Refrigerator, scale, etc
Internet On Everything

SETI@home: Global Distributed Computing

- Running on 500,000 PCs, ~1000 CPU Years per Day
  - 485,821 CPU Years so far
- Sophisticated Data & Signal Processing Analysis
- Distributes Datasets from Arecibo Radio Telescope
SETI@home

- Use thousands of Internet-connected PCs to help in the search for extraterrestrial intelligence.
- When their computer is idle or being wasted this software will download a 300 kilobyte chunk of data for analysis. Performs about 3 Tflops for each client in 15 hours.
- The results of this analysis are sent back to the SETI team, combined with thousands of other participants.

Largest distributed computation project in existence
- Averaging 40 Tflop/s
- Today a number of companies trying this for profit.

Google

- Google query attributes
  - 150M queries/day (2000/second)
  - 100 countries
  - 8.0B documents in the index
- Data centers
  - 100,000 Linux systems in data centers around the world
    - 15 TFlop/s and 1000 TB total capability
    - 40-80 1U/2U servers/cabinet
    - 100 MB Ethernet switches/cabinet with gigabit Ethernet uplink
  - growth from 4,000 systems (June 2000)
    - 18M queries then
- Performance and operation
  - simple reissue of failed commands to new servers
  - no performance debugging
    - problems are not reproducible

Source: Monika Henzinger, Google & Cleve Moler

Eigenvalue problem; \( Ax = \lambda x \)

\( n=8 \times 10^9 \)

(see: MathWorks Cleve's Corner)

The matrix is the transition probability matrix of the Markov chain; \( Ax = x \)
Next Generation Web

- To treat CPU cycles and software like commodities.
- Enable the coordinated use of geographically distributed resources – in the absence of central control and existing trust relationships.
- Computing power is produced much like utilities such as power and water are produced for consumers.
- Users will have access to “power” on demand
- This is one of our efforts at UT.

• Why Parallel Computing

- Desire to solve bigger, more realistic applications problems.
- Fundamental limits are being approached.
- More cost effective solution
Principles of Parallel Computing

- Parallelism and Amdahl’s Law
- Granularity
- Locality
- Load balance
- Coordination and synchronization
- Performance modeling

All of these things makes parallel programming even harder than sequential programming.

“Automatic” Parallelism in Modern Machines

- Bit level parallelism
  - within floating point operations, etc.
- Instruction level parallelism (ILP)
  - multiple instructions execute per clock cycle
- Memory system parallelism
  - overlap of memory operations with computation
- OS parallelism
  - multiple jobs run in parallel on commodity SMPs

Limits to all of these -- for very high performance, need user to identify, schedule and coordinate parallel tasks
Finding Enough Parallelism

- **Suppose only part of an application seems parallel**
- **Amdahl’s law**
  - let $f_s$ be the fraction of work done sequentially, $(1 - f_s)$ is fraction parallelizable
  - $N$ = number of processors

- Even if the parallel part speeds up perfectly may be limited by the sequential part

Amdahl’s Law

Amdahl’s Law places a strict limit on the speedup that can be realized by using multiple processors. Two equivalent expressions for Amdahl’s Law are given below:

\[
\frac{t_N}{t_1} = \frac{f_p/N + f_s}{f_p/N + f_s} \quad \text{Effect of multiple processors on run time}
\]

\[
S = \frac{1}{f_s + f_p/N} \quad \text{Effect of multiple processors on speedup}
\]

Where:
- $f_s$ = serial fraction of code
- $f_p = $ parallel fraction of code = $1 - f_s$
- $N = $ number of processors
It takes only a small fraction of serial content in a code to degrade the parallel performance. It is essential to determine the scaling behavior of your code before doing production runs using large numbers of processors.

**Illustration of Amdahl’s Law**

<table>
<thead>
<tr>
<th>Speedup</th>
<th>Number of Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Overhead of Parallelism**

- **Given enough parallel work, this is the biggest barrier to getting desired speedup**
- **Parallelism overheads include:**
  - cost of starting a thread or process
  - cost of communicating shared data
  - cost of synchronizing
  - extra (redundant) computation
- **Each of these can be in the range of milliseconds (=millions of flops) on some systems**
- **Tradeoff:** Algorithm needs sufficiently large units of work to run fast in parallel (i.e. large granularity), but not so large that there is not enough parallel work
Locality and Parallelism

- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast on average
- Parallel processors, collectively, have large, fast
  - the slow accesses to “remote” data we call “communication”
- Algorithm should do most work on local data

Load Imbalance

- Load imbalance is the time that some processors in the system are idle due to
  - insufficient parallelism (during that phase)
  - unequal size tasks
- Examples of the latter
  - adapting to “interesting parts of a domain”
  - tree-structured computations
  - fundamentally unstructured problems
- Algorithm needs to balance load
What is Ahead?

- Greater instruction level parallelism?
- Bigger caches?
- Multiple processors per chip?
- Complete systems on a chip? (Portable Systems)

- High performance LAN, Interface, and Interconnect

Directions

- **Move toward shared memory**
  - SMPs and Distributed Shared Memory
  - Shared address space w/deep memory hierarchy
- **Clustering of shared memory machines for scalability**
- **Efficiency of message passing and data parallel programming**
  - Helped by standards efforts such as MPI and OpenMP
High Performance Computers

♦ ~ 20 years ago
  ➢ 1x10^6 Floating Point Ops/sec (Mflop/s)
    » Scalar based

♦ ~ 10 years ago
  ➢ 1x10^9 Floating Point Ops/sec (Gflop/s)
    » Vector & Shared memory computing, bandwidth aware
    » Block partitioned, latency tolerant

♦ ~ Today
  ➢ 1x10^12 Floating Point Ops/sec (Tflop/s)
    » Highly parallel, distributed processing, message passing, network based
    » data decomposition, communication/computation

♦ ~ 1 year away
  ➢ 1x10^15 Floating Point Ops/sec (Pflop/s)
    » Many more levels MH, combination/grid&HPC
    » More adaptive, LT and bandwidth aware, fault tolerant,
      extended precision, attention to SMP nodes

Top 500 Computers

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

\[ Ax = b, \text{ dense problem} \]

Updated twice a year
SC‘xy in the States in November
Meeting in Germany in June
What is a Supercomputer?

A supercomputer is a hardware and software system that provides close to the maximum performance that can currently be achieved.

Over the last 14 years the range for the Top500 has increased greater than Moore’s Law.

1993:
- #1 = 59.7 GFlop/s
- #500 = 422 MFlop/s

2007:
- #1 = 478 TFlop/s
- #500 = 5.9 TFlop/s

Why do we need them? Almost all of the technical areas that are important to the well-being of humanity use supercomputing in fundamental and essential ways.

Computational fluid dynamics, protein folding, climate modeling, national security, in particular for cryptanalysis and for simulating nuclear weapons to name a few.

Architecture/Systems Continuum

- **Custom processor** with custom interconnect
  - Cray X1
  - NEC SX-8
  - IBM Regatta
  - IBM Blue Gene/L
- **Commodity processor** with custom interconnect
  - SGI Altix
  - Intel Itanium 2
  - Cray XT3, XD1
  - AMD Opteron
- **Commodity processor** with commodity interconnect
  - Clusters
    - Pentium, Itanium, Opteron, Alpha
    - GigE, Infiniband, Myrinet, Quadrics
  - NEC TX7
  - IBM eServer
  - Dawning

Tightly Coupled

Loosely Coupled

Hybrid

Custom

Commod
IBM BlueGene/L #1 212,992 Cores
Total of 26 systems all in the Top176
2.6 MWatts (2600 homes) (104 racks, 104x32x32) (32 Node boards, 8x8x16) 2048 processors
70,000 ops/s/person
212992 procs
2.9/5.7 TF/s
32 TB DDR
180/360 TF/s
Full system total of 131,072 processor cores

The compute node ASICs include all networking and processor functionality. Each compute ASIC includes two 32-bit superscalar PowerPC 440 embedded cores (note that L1 cache coherence is not maintained between these cores). (20.7K sec about 5.7 hours; n=2.5M)

“Fastest Computer”
BG/L 700 MHz 213K proc
104 racks
Peak: 596 Tflop/s
Linpack: 498 Tflop/s
84% of peak
DOE NNSA

### LLNL

- **IBM BG/L**
  - Power PC
  - Cores: 212,992
  - Peak: 596 TF
  - Memory: 73.7 TB

- **IBM Purple**
  - Power 5
  - Cores: 12,208
  - Peak: 92.8 TF
  - Memory: 48.8 TB

### SNL

- **Red Storm Cray**
  - AMD Dual Core
  - Cores: 27,200
  - Peak: 127.5 TF
  - Memory: 40 TB

- **RoadRunner IBM**
  - AMD Dual Core
  - Cores: 18,252
  - Peak: 81.1 TF
  - Memory: 27.6 TB

### LANL

- **Thunderbird Dell**
  - Intel Xeon
  - Cores: 9,024
  - Peak: 53 TF
  - Memory: 6 TB

LANL Roadrunner

A Petascale System in 2008

- "Connected Unit" cluster
  - 192 Opteron nodes
    - 180 w/ 2 dual-Cell blades
      - connected w/ 4 PCIe x8 links
  - ≈ 13,000 Cell HPC chips
  - ≈ 1.33 PetaFlop/s (from Cell)
  - ≈ 7,000 dual-core Opterons

- ~18 clusters
  - 2nd stage InfiniBand 4x DDR interconnect
    - (18 sets of 12 links to 8 switches)
  - 2nd stage InfiniBand interconnect (8 switches)

Based on the 100 Gflop/s (DP) Cell chip

Approval by DOE 12/07
First CU being built today
Expect a May Pflop/s run
Full system to LANL in December 2008
DOE OS

ORNL
- Jaguar Cray XT
  - AMD Dual Core
  - Cores: 11,706
  - Peak: 119.4 TF
  - Upgrading 250 TF
  - Memory: 46 TB
- Phoenix Cray X1
  - Cray Vector
  - Cores: 1,024
  - Peak: 18.3 TF
  - Memory: 2 TB

LBNL
- Franklin Cray XT
  - AMD Dual Core
  - Cores: 19,320
  - Peak: 100.4 TF
  - Memory: 39 TB
- Bassi IBM
  - PowerPC
  - Cores: 976
  - Peak: 7.4 TF
  - Memory: 3.5 TB
- Seaborg IBM
  - Power3
  - Cores: 976
  - Peak: 9.9 TF
  - Memory: 7.3 TB

ANL
- BG/P IBM
  - PowerPC
  - Cores: 131,072
  - Peak: 111 TF
  - Memory: 65.5 TB

ORNL NCCS Roadmap for Leadership Computing

Mission: Deploy and operate the computational resources needed to tackle global challenges

- Future Energy
- Understanding the universe
- Nanoscale materials
- Climate Change
- Computational Biology

Vision: Maximize scientific productivity and progress on the largest scale computational problems

- Providing world class computational resources and specialized services
- Providing a stable hardware/software path of increasing scale to maximize productive applications development
- Work with users to scale applications to take advantage of systems

Future leadership class sustained PF system for science

Cray XT4: 119 TF
Cray X1E: 18 TF
Cray XT4: 250 TF
Cray X1E: 18 TF
Cray HPCS-0: 1 PF

FY2007
FY2008
FY2009
FY2011
1000 TF Cray “Baker” system in 2008

System configuration
- 1 PF peak
- ~24,000 quad-core processors
- ~50 kW per cabinet
- ~7 MW power

1 PF Cray system in 2008

Used by permission: Cray, Inc.

Nanoscience Roadmap

Expected outcomes
5 years
Realistic simulation of self-assembly and single-molecule electron transport
Finite-temperature properties of nanoparticles/quantum corrals

10 years
Multiscale modeling of molecular electronic devices
Computation-guided search for new materials/nanostructures
Biology Roadmap

Expected outcome

5 years
- Metabolic flux modeling for hydrogen and carbon fixation pathways
- Constrained flexible docking simulations of interacting proteins

10 years
- Multiscale stochastic simulations of microbial metabolic, regulatory, and protein interaction networks
- Dynamic simulations of complex molecular machines

NCCS Data Analytics & Visualization

- Today: Dedicated cluster for data analysis and visualization
- Goal: Port the analysis and visualization tools to run directly on the large systems to avoid having to move data
**Fusion Roadmap**

**Expected Outcomes**

**5 Years**
- Full-torus, electromagnetic simulation of turbulent transport with kinetic electrons for simulation times approaching transport time-scale
- Develop understanding of internal reconnection events in extended MHD, with assessment of RF heating and current drive techniques for mitigation

**10 years**
- Develop quantitative, predictive understanding of disruption events in large tokamaks
- Begin integrated simulation of burning plasma devices – multi-physics predictions for ITER

**Climate Roadmap**

**Expected Outcomes**

**5 years**
- Fully coupled carbon-climate simulation
- Fully coupled sulfur-atmospheric chemistry simulation

**10 years**
- Cloud-resolving 30-km spatial resolution atmosphere climate simulation
- Fully coupled, physics, chemistry, biology earth system model
Top500 by Usage

- Industry: 287, 57%
- Research: 8, 2%
- Academic: 86, 17%
- Government: 101, 20%
- Vendor: 15, 3%
- Vendor: 3, 1%
- Classified: 0%

Chips Used in Each of the 500 Systems

- 72% Intel
- 12% IBM
- 16% AMD
Interconnects / Systems

Interconnect Analysis

- Others
- Cray Interconnect
- SP Switch
- Crossbar
- Quadrics
- Infiniband (121)
- Myrinet (18)
- Gigabit Ethernet (270)

GigE = Infiniband + Myrinet
Cores per System – November 2007

Top500 Systems November 2007
Computer Classes / Systems

Countries / Performance (Nov 2007)

<table>
<thead>
<tr>
<th>Country</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>60%</td>
</tr>
<tr>
<td>Germany</td>
<td>7.7%</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>7.4%</td>
</tr>
<tr>
<td>China</td>
<td>4.2%</td>
</tr>
<tr>
<td>France</td>
<td>3.2%</td>
</tr>
<tr>
<td>Taiwan</td>
<td>3.2%</td>
</tr>
<tr>
<td>India</td>
<td>2.8%</td>
</tr>
<tr>
<td>Sweden</td>
<td>2.7%</td>
</tr>
<tr>
<td>Spain</td>
<td>2.5%</td>
</tr>
<tr>
<td>Others</td>
<td>60%</td>
</tr>
</tbody>
</table>

30th List / November 2007 www.top500.org page 53

30th List / November 2007 www.top500.org page 53

30th List / November 2007 www.top500.org page 53

30th List / November 2007 www.top500.org page 53
Environmental Burden of PC CPUs

- Total power consumption of CPUs in world’s PCs:
  - 1992: 160 MWatts (87M CPUs)
  - 2001: 9,000 MWatts (500M CPUs)

- That’s 4 Hoover Dams!

Power Consumption of World’s CPUs
Power is an Industry Wide Problem

Google facilities
- leveraging hydroelectric power
- old aluminum plants
- >500,000 servers worldwide

New Google Plant in The Dulles, Oregon, from NYT, June 14, 2006

Gflop/KWatt in the Top 20
And Now We Want Petascale …

- What is a conventional petascale machine?
  - Many high-speed bullet trains … a significant start to a conventional power plant.

Top Three Reasons for “Eliminating” Global Climate Warming in the Machine Room

3. HPC Contributes to Global Climate Warming
   - “I worry that we, as HPC experts in global climate modeling, are contributing to the very thing that we are trying to avoid: the generation of greenhouse gases.”

2. Electrical Power Costs $$$.
   - Japanese Earth Simulator
     » Power & Cooling: 12 MW/year → $9.6 million/year?
   - Lawrence Livermore National Laboratory
     » Power & Cooling of HPC: $14 million/year
     » Power-up ASC Purple → “Panic” call from local electrical company.

1. Reliability & Availability Impact Productivity
     » 50,538 MW: A load not expected to be reached until 2010!
Reliability & Availability of HPC

<table>
<thead>
<tr>
<th>Systems</th>
<th>CPUs</th>
<th>Reliability &amp; Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCI Q</td>
<td>8,192</td>
<td>MTBI: 6.5 hrs, 114 unplanned outages/month. HW outage sources: storage, CPU, memory.</td>
</tr>
<tr>
<td>NERSC Seaborg</td>
<td>6,656</td>
<td>MTBI: 14 days, MTTR: 3.3 hrs. SW is the main outage source. Availability: 98.74%.</td>
</tr>
<tr>
<td>PSC Lemieux</td>
<td>3,016</td>
<td>MTBI: 9.7 hrs. Availability: 98.33%.</td>
</tr>
<tr>
<td>Google</td>
<td>~15,000</td>
<td>20 reboots/day, 2-3% machines replaced/year. HW outage sources: storage, memory. Availability ~100%.</td>
</tr>
</tbody>
</table>

MTBI: mean time between interrupts; MTBF: mean time between failures; MTTR: mean time to restore

Source: Daniel A. Reed, RENCI

Top500 Conclusions

- **Microprocessor based supercomputers have brought a major change in accessibility and affordability.**
- **MPPs continue to account of more than half of all installed high-performance computers worldwide.**
With All the Hype on the PS3
We Became Interested

- The PlayStation 3’s CPU based on a “Cell” processor
- Each Cell contains a Power PC processor and 8 SPEs. (SPE is processing unit, SPE: SPU + DMA engine)
  - An SPE is a self contained vector processor which acts independently from the others.
    - 4 way SIMD floating point units capable of a total of 25.6 Gflop/s @ 3.2 GHZ
  - 204.8 Gflop/s peak!
  - The catch is that this is for 32 bit floating point; (Single Precision SP)
  - And 64 bit floating point runs at 14.6 Gflop/s total for all 8 SPEs!!
    - Divide SP peak by 14; factor of 2 because of DP and 7 because of latency issues

Increasing CPU Performance:
A Delicate Balancing Act

- Lower Voltage
- Increase Clock Rate & Transistor Density

We have seen increasing number of gates on a chip and increasing clock speed.

Heat becoming an unmanageable problem, Intel Processors > 100 Watts

We will not see the dramatic increases in clock speeds in the future.

However, the number of gates on a chip will continue to increase.

Intel Yonah will double the processing power on a per watt basis.
Intel Prediction of Microprocessor Frequency (ca. 2001)

Adopted from a presentation by S. Borkar, Intel

Intel Prediction of Microprocessor Power Consumption (ca. 2001)

Adopted from a presentation by S. Borkar, Intel
Moore’s Law for Power (P $\propto V^2 f$)

- Chip Maximum Power in watts/cm$^2$
- Not too long to reach Nuclear Reactor
- Surpassed Heating Plate
- Pentium - 14 watts
- Pentium II - 35 watts
- Pentium III - 35 watts
- Pentium IV - 75 watts
- Itanium - 130 watts


Change Is Coming

No Free Lunch For Traditional Software

(Without highly concurrent software it won’t get any faster!)
What is Multicore?

- Single chip
- Multiple distinct processing engine
- Multiple, independent threads of control (or program counters MIMD)

Integration is Efficient

- Discrete chips
  - Bandwidth 2GBps
  - Latency 60 ns

- Multicore
  - Bandwidth > 20 GBps
  - Latency < 3ns

Parallelism and interconnect efficiency enables harnessing the power of n. n cores can yield an n-fold increase in performance
Novel Opportunities in Multicores

- Don’t have to contend with uniprocessors
- Not your same old multiprocessor problem
  - How does going from Multiprocessors to Multicores impact programs?
  - What changed?
  - Where is the Impact?
    » Communication Bandwidth
    » Communication Latency

Communication Bandwidth

- How much data can be communicated between two cores?
- What changed?
  - Number of Wires
  - Clock rate
  - Multiplexing
- Impact on programming model?
  - Massive data exchange is possible
  - Data movement is not the bottleneck
    → processor affinity not that important
Communication Latency

- How long does it take for a round trip communication?
- What changed?
  - Length of wire
  - Pipeline stages
- Impact on programming model?
  - Ultra-fast synchronization
  - Can run real-time apps on multiple cores

50X

~200 Cycles ~4 cycles

80 Core
Intel's 80 Core chip
- 1 Tflop/s
- 62 Watts
- 1.2 TB/s internal BW

Recipe Prototype May Herald New Age of Processing

By JOHN MARKOFF
San Francisco, Feb. 11 — Intel will demonstrate on
Monday an experimental computer chip with 80 separate
processing engines, or cores, that company executives say
provides a model for commercial chips that will be used widely
in standard desktop, laptop and server computers within five
years.

The new processor, which the company first described as a Teraflop Chip at a
conference last year, will be detailed in a technical paper to be presented on the opening day of the

While the chip is not compatible with Intel’s current
chips, the company said it had already begun design
work on a commercial version that would essentially
have dozens or even hundreds of Intel-compatible
microprocessors laid out in a tiled pattern on a
single chip.
Multicore, FPGA, Heterogeneity

What about the potential of FPGA or hybrid core processors?

Major Changes to Software

- **Must rethink the design of our software**
  - Another disruptive technology
    - Similar to what happened with cluster computing and message passing
  - Rethink and rewrite the applications, algorithms, and software
- **Numerical libraries for example will change**
  - For example, both LAPACK and ScaLAPACK will undergo major changes to accommodate this
Interconnect Options

Top500 Conclusions

- Microprocessor based supercomputers have brought a major change in accessibility and affordability.
- MPPs continue to account for more than half of all installed high-performance computers worldwide.
Distributed and Parallel Systems

- Distributed systems
  - hetero-
  - geneous
  - SET@home
  - Entropia
  - Grid Computing
  - Beowulf
  - Berkeley NOW
  - NW/Caltech
  - Parallel Dist. Comp.
  - ASCI Trifors

- Massively parallel systems
  - homogeneous

- Gather (unused) resources
- Steal cycles
- System SW manages resources
- System SW adds value
- 10% - 20% overhead is OK
- Resources drive applications
- Time to completion is not critical
- Time-shared
- Bounded set of resources
- Apps grow to consume all cycles
- Application manages resources
- System SW gets in the way
- 5% overhead is maximum
- Apps drive purchase of equipment
- Real-time constraints
- Space-shared

Virtual Environments

Do they make any sense?
This advance took place over a span of about 36 years, or 24 doubling times for Moore’s Law.

- $2^{24} = 16$ million ⇒ the same as the factor from algorithms alone!
Algorithms and Moore’s Law

- This advance took place over a span of about 36 years, or 24 doubling times for Moore's Law
- \(2^{24} = 16\) million \(\Rightarrow\) the same as the factor from algorithms alone!

Different Architectures

- **Parallel computing**: single systems with many processors working on same problem
- **Distributed computing**: many systems loosely coupled to work on related problems
- **Grid Computing**: many systems tightly coupled by software, perhaps geographically distributed, to work together on single problems or on related problems
Types of Parallel Computers

- The simplest and most useful way to classify modern parallel computers is by their memory model:
  - shared memory
  - distributed memory

Shared vs. Distributed Memory

**Shared memory** - single address space. All processors have access to a pool of shared memory. (Ex: SGI Origin, Sun E10000)

**Distributed memory** - each processor has its own local memory. Must do message passing to exchange data between processors. (Ex: CRAY T3E, IBM SP, clusters)
Shared Memory: UMA vs. NUMA

Uniform memory access (UMA): Each processor has uniform access to memory. Also known as symmetric multiprocessors (Sun E10000)

Non-uniform memory access (NUMA): Time for memory access depends on location of data. Local access is faster than non-local access. Easier to scale than SMPs (SGI Origin)

Distributed Memory: MPPs vs. Clusters

- Processors-memory nodes are connected by some type of interconnect network
  - Massively Parallel Processor (MPP): tightly integrated, single system image.
  - Cluster: individual computers connected by s/w
Processors, Memory, & Networks

- Both shared and distributed memory systems have:
  1. **processors**: now generally commodity processors
  2. **memory**: now generally commodity DRAM
  3. **network/interconnect**: between the processors and memory (bus, crossbar, fat tree, torus, hypercube, etc.)

Interconnect-Related Terms

- **Latency**: How long does it take to start sending a "message"? Measured in microseconds.
  (Also in processors: How long does it take to output results of some operations, such as floating point add, divide etc., which are pipelined?)

- **Bandwidth**: What data rate can be sustained once the message is started? Measured in Mbytes/sec.
Interconnect-Related Terms

**Topology**: the manner in which the nodes are connected.

- Best choice would be a fully connected network (every processor to every other). Unfeasible for cost and scaling reasons.
- Instead, processors are arranged in some variation of a grid, torus, or hypercube.

![Diagrams of 3-d hypercube, 2-d mesh, and 2-d torus]

Standard Uniprocessor Memory Hierarchy

- **Intel Pentium 4 2 GHz processor**
- **P7 Prescott 478**
  - 8 Kbytes of 4 way assoc. L1 instruction cache with 32 byte lines.
  - 8 Kbytes of 4 way assoc. L1 data cache with 32 byte lines.
  - 256 Kbytes of 8 way assoc. L2 cache 32 byte lines.
  - 400 MB/s bus speed
  - SSE2 provide peak of 4 Gflop/s

Each flop requires 3 words of data
At 4 Gflop/s needs 12 GW/s bw
Bus has only .5GW/s
So if driven from memory
24 times off of peak rate!!
**Locality and Parallelism**

- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast on average
- Parallel processors, collectively, have large, fast $\pi$
  - the slow accesses to "remote" data we call "communication"
- Algorithm should do most work on local data
Amdahl’s Law

Amdahl’s Law places a strict limit on the speedup that can be realized by using multiple processors. Two equivalent expressions for Amdahl’s Law are given below:

\[ t_N = (f_p/N + f_s) t_1 \quad \text{Effect of multiple processors on run time} \]

\[ S = 1/(f_s + f_p/N) \quad \text{Effect of multiple processors on speedup} \]

Where:
- \( f_s \) = serial fraction of code
- \( f_p \) = parallel fraction of code = 1 - \( f_s \)
- \( N \) = number of processors

---

### Amdahl’s Law - Theoretical Maximum

**Speedup of parallel execution**

- \[ \text{Speedup} = 1/(P/N + S) \]
  - \( P \) (parallel code fraction) \( S \) (serial code fraction) \( N \) (processors)

**Example: Image processing**

- 30 minutes of preparation (serial)
- One minute to scan a region
- 30 minutes of cleanup (serial)

<table>
<thead>
<tr>
<th>Number of processors</th>
<th>Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30 + 300 + 30 = 360</td>
<td>1.0X</td>
</tr>
<tr>
<td>2</td>
<td>30 + 150 + 30 = 210</td>
<td>1.7X</td>
</tr>
<tr>
<td>10</td>
<td>30 + 30 + 30 = 90</td>
<td>4.0X</td>
</tr>
<tr>
<td>100</td>
<td>30 + 3 + 30 = 63</td>
<td>5.7X</td>
</tr>
<tr>
<td>Infinite</td>
<td>30 + 0 + 30 = 60</td>
<td>6.0X</td>
</tr>
</tbody>
</table>

- Speedup is restricted by serial portion. And, speedup increases with greater number of cores!
It takes only a small fraction of serial content in a code to degrade the parallel performance. It is essential to determine the scaling behavior of your code before doing production runs using large numbers of processors.

Amdahl’s Law provides a theoretical upper limit on parallel speedup assuming that there are no costs for communications. In reality, communications (and I/O) will result in a further degradation of performance.
Overhead of Parallelism

- Given enough parallel work, this is the biggest barrier to getting desired speedup
- Parallelism overheads include:
  - cost of starting a thread or process
  - cost of communicating shared data
  - cost of synchronizing
  - extra (redundant) computation
- Each of these can be in the range of milliseconds (=millions of flops) on some systems
- Tradeoff: Algorithm needs sufficiently large units of work to run fast in parallel (i.e. large granularity), but not so large that there is not enough parallel work

Load Imbalance

- Load imbalance is the time that some processors in the system are idle due to
  - insufficient parallelism (during that phase)
  - unequal size tasks
- Examples of the latter
  - adapting to "interesting parts of a domain"
  - tree-structured computations
  - fundamentally unstructured problems
- Algorithm needs to balance load
What is Ahead?

- Greater instruction level parallelism?
- Bigger caches?
- Multiple processors per chip?
- Complete systems on a chip? (Portable Systems)

- High performance LAN, Interface, and Interconnect

Directions

- Move toward shared memory
  - SMPs and Distributed Shared Memory
  - Shared address space with deep memory hierarchy
- Clustering of shared memory machines for scalability
- Efficiency of message passing and data parallel programming
  - Helped by standards efforts such as MPI and HPF