What is Parallel Computer?

A parallel computer is a collection of processing elements that cooperate to solve large problems fast.

Some broad issues:
- Resource Allocation:
  - how large a collection?
  - how powerful are the elements?
- Data access, Communication and Synchronization
  - how do the elements cooperate and communicate?
  - how are data transmitted between processors?
  - what are the abstractions and primitives for cooperation?
- Performance and Scalability
  - how does it all translate into performance?
  - how does it scale?
Why Parallel Computing

Application demands:
- Computer simulation becomes the third pillar, complementing the activities of theory and experimentation
- General-purpose computing: Video, Graphics, CAD, Database, etc
- Our insatiable need for computing cycles in challenge applications

Technology Trends
- Number of transistors on chip growing rapidly
- Clock rates expected to go up only slowly

Architecture Trends
- Instruction-level parallelism valuable but limited
- Coarser-level parallelism, as in MPs, the most viable approach

Economics

Simulation: The Third Pillar of Science

Traditional scientific and engineering paradigm:
1) Do theory or paper design.
2) Perform experiments or build system.

Limitations:
- Too difficult -- build large wind tunnels.
- Too expensive -- build a throw-away passenger jet.
- Too slow -- wait for climate or galactic evolution.
- Too dangerous -- weapons, drug design, climate experimentation.

Computational science paradigm:
3) Use high performance computer systems to simulate the phenomenon
   - Base on known physical laws and efficient numerical methods.
Challenging Computation Examples

Science
- Global climate modeling
- Astrophysical modeling
- Biology: genomics; protein folding; drug design
- Computational Chemistry
- Computational Material Sciences and Nanosciences

Engineering
- Crash simulation
- Semiconductor design
- Earthquake and structural modeling
- Computation fluid dynamics (airplane design)
- Combustion (engine design)

Business
- Financial and economic modeling
- Transaction processing, web services and search engines

Defense
- Nuclear weapons -- test by simulation
- Cryptography

Units of Measure in HPC

High Performance Computing (HPC) units are:
- Flop/s: floating point operations
- Bytes: size of data

Typical sizes are millions, billions, trillions…

Mega
- Mflop/s = $10^6$ flop/sec
- Mbyte = $10^6$ byte
  (also $2^{20} = 1048576$)

Giga
- Gflop/s = $10^9$ flop/sec
- Gbyte = $10^9$ byte
  (also $2^{30} = 1073741824$)

Tera
- Tflop/s = $10^{12}$ flop/sec
- Tbyte = $10^{12}$ byte
  (also $2^{40} = 1099511627776$)

Peta
- Pflop/s = $10^{15}$ flop/sec
- Pbyte = $10^{15}$ byte
  (also $2^{50} = 1125899906842624$)
Global Climate Modeling Problem

Problem is to compute:
\[ f(\text{latitude}, \text{longitude}, \text{elevation}, \text{time}) \rightarrow \]
- temperature, pressure, humidity, wind velocity

Approach:
- Discretize the domain, e.g., a measurement point every 1km
- Devise an algorithm to predict weather at time t+1 given t

- Uses:
  - Predict major events, e.g., El Nino
  - Use in setting air emissions standards

Numerical Climate Modeling at NASA

- Weather forecasting over US landmass: 3000 x 3000 x 11 miles
- Assuming 0.1 cubic element ---> $10^{11}$ elements
- Assuming 2 day prediction @ 30 min ---> 100 steps in time scale
- Computation: Partial Difference Equation and Finite Element Scheme
- Single element computation takes 100 flops
- Total number of flops: $10^{11} \times 100 \times 100 = 10^{15}$ (peta-flops)
- Current uniprocessor power: $10^9$ flops/sec (giga-flops)
- It takes $10^6$ seconds or 280 hours. (Forecast nine days late!)
- 1000 processors at 10% efficiency ---> around 3 hours
- State of the art models require integration of atmosphere, ocean, sea-ice, land models, plus possibly carbon cycle, geochemistry and more; Current models are coarser than this
Commercial Computing

Servers rely more on parallelism for high end

- Database and Web servers for online transaction processing
- Decision support
- Data mining and data warehousing
- Financial modeling

- Scale not so large, but use much more wide-spread
- Computational power determines scale of business that can be handled. E.g. AOL story in 1997

© C. Xu 2004
Real World Example: Inktomi

200+ dual processor machines (as of Year 2001)  
Data partitioned across nodes

More examples

Google
- 200 million searches per day (as of March 2004)
- 6,000 Linux/Intel PCs and 12,000 disks

Hotmail (as of 2000)
- 60 web servers
- 30 data servers
  - data partitioned across them
  - the same person always goes to the same server

CNN
- 20 web servers
- dynamically partitioned to adapt to traffic pattern


**Why Parallel Computer?**

---

**Tunnel Vision by Experts**

“I think there is a world market for maybe five computers.”
– Thomas Watson, chairman of IBM, 1943.

“There is no reason for any individual to have a computer in their home”

“640K [of memory] ought to be enough for anybody.”
– Bill Gates, chairman of Microsoft, 1981.

Slide source: Warfield et al.

© C. Xu 2004
Technology Trends: μ-processor Capacity

2X transistors/Chip Every 1.5 years
Called “Moore’s Law”

Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Microprocessors have become smaller, denser, and more powerful.

Slide source: Jack Dongarra

Technology Trends: Transistor Count

- 40% per year, more functions can be performed by a CPU
- Similar story for storage:
  capacity increased by 1000x over 20 years, speed only 2x

© C. Xu 2004
Technology Trends: Clock Rate

- 30% per year --> today's PC is yesterday's Supercomputer

Technology Trends

The natural building block for multiprocessors is now also about the fastest!
How fast can a serial computer be?

Consider the 1 Tflop sequential machine
- data must travel some distance, r, to get from memory to CPU
- to get 1 data element per cycle, this means \(10^{12}\) times per second at the speed of light, \(c = 3e8\) m/s
- so \(r < c/10^{12} = .3\) mm

Now put 1 TB of storage in a .3 mm\(^2\) area
- each word occupies \(\sim 3\) Angstroms\(^2\), the size of a small atom

Scaling microprocessors

What happens when feature size shrinks by a factor of x?

Clock rate goes up by \(x\)
- actually a little less

Transistors per unit area goes up by \(x^2\)

Die size also tends to increase
- typically another factor of \(\sim x\)

Raw computing power of the chip goes up by \(\sim x^4\!\)
- of which \(x^3\) is devoted either to parallelism or locality
Role of Architecture

Clock rate increases 30% per year, while the overall CPU performance increases 50% to 100% per year

Where is the rest coming from??

Architectural Trends

Greatest trend is increase in parallelism

- **Up to 1985**: bit level parallelism: 4-bit -> 8 bit -> 16-bit
  - slows after 32 bit
  - adoption of 64-bit now under way
- **Mid 80s to mid 90s**: instruction level parallelism
  - pipelining and simple instruction sets (RISC)
  - on-chip caches and functional units => **superscalar** execution
- **Next step**:
  - Memory system parallelism: overlap of mem operation with computation
  - thread level parallelism
Number of transistors per processor chip

Instruction-Level Parallelism

Thread-Level Parallelism?

Bit-Level Parallelism

Year


Number of transistors

100,000,000

10,000,000

1,000,000

100,000

10,000

1,000

100

10

1

ILP Ideal Potential

Fraction of total cycles (%)

Speedup

Instructions issued per cycle

Pentium Pro: 3 instructions,
PowerPC 604: 4 instructions

© C. Xu 2004
Why Parallel Computing: Economics

- Commodity means CHEAP
- Multiprocessors being pushed by software vendors (e.g. database) as well as hardware vendors
- Standardization by Intel makes small, bus-based SMPs commodity
- Desktop: few smaller processors versus one larger one?
  - Large scale multiprocessors replacing vector supercomputers
  - Cluster Computing
  - Multiprocessor on a chip !!
Summary: Why Parallel Computing

Application demands:
- Computer simulation becomes the third pillar, complementing the activities of theory and experimentation
- General-purpose computing: Video, Graphics, CAD, Database, etc
- Our insatiable need for computing cycles in challenge applications

Technology Trends
- Number of transistors on chip growing rapidly
- Clock rates expected to go up only slowly

Architecture Trends
- Instruction-level parallelism valuable but limited
- Coarser-level parallelism, as in MPUs, the most viable approach

Economics