Parallelism and Cloud Computing

Kai Shen

1

Parallel Computing

- Parallel computing: Process sub-tasks simultaneously so that work can be completed faster.
  - For instance, divide the work of matrix multiplication into subtasks to run in parallel
  - Any other applications you can think of?

2

Why Parallel Computing?

The speed of uniprocessor will catch up with the need of computing?

- Moore's law (1965, with later revisions) – circuit complexity doubles every one year (18 months, 2 years)
- Some argue that it is a self-fulfilling prophecy
- Moore said in 2005:
  "In terms of size [of transistors] you can see that we're approaching the size of atoms which is a fundamental barrier, …"

3

Why Parallel Computing?

Not always desirable to use the fastest hardware.

Example:

- A machine with four 3GHz processor cores, 2GB memory, and hard drives consumes about 250 Watts
- A machine with a 500MHz CPU, 128MB memory, and a CompactFlash drive consumes about 5 Watts
- It may be more energy-efficient to use multiple low-power machines in parallel

4

Why Parallel Computing?

Multiprocessors are increasingly ubiquitous

- Commodity processors contain multiple computing cores because they are cost-effective to produce, and power-efficient to run
- Traditionally sequential software (e.g., desktop applications) should take advantage of it

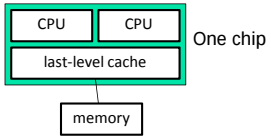
5

Multicore Architecture

- Last-level cache becomes a significant part of the chip

⇨ Multicore: add another processor core on the chip (sharing a single last-level cache)

- Low cost (manufacturing, power)



- Additional benefit: faster processor-to-processor sharing

6

Arguments against Parallel Computing?

- Parallel programming is too difficult
- Performance isn't important

7

Parallelism and Dependency

- Computation-intensive application that contains tasks that can run in parallel
- Dependencies limit available parallelism/concurrency
  - Dependency between two operations – one has to wait for another to complete
  - Examples of dependencies?
- Amdahl's law:

$$\text{Normalized runtime} = 1 - \text{fraction\_enhanced} + \frac{\text{fraction\_enhanced}}{\text{speedup\_enhanced}}$$

8

Load Balance

- Utilizations of parallel processors
  - What happens if most work is assigned to one processor?
- Require load balancing
  - Parallel processors are typically symmetric

9

Simulation of Ocean Currents

- Entity in ocean floor: a unit body of water
- Variables: velocity, direction, …
- Simulation over time
  - Velocity/direction of water body depends on the velocity/direction of nearby water bodies in the last time unit
- Parallelism/concurrency
- Dependencies
- Load balance

10

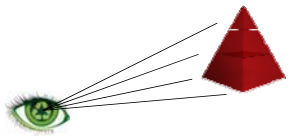
Evolution of Galaxies

- Entity in space: a star
- Variables: velocity, direction, location, …
- Simulation over time
  - Velocity/direction/location of a star depends on the mass and location of other stars in the last time unit
- N-body simulation
- Parallelism/concurrency
- Dependencies
- Load balance

11

Ray Tracing

- Ray tracing



- Parallelism/concurrency
- Dependencies
- Load balance

12

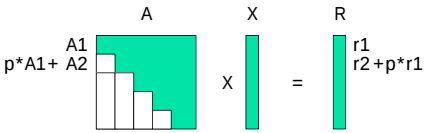
Google PageRank

- To calculate the probability of arriving at a page after clicking a link at random from another page
- Page A is linked by pages B and C, then:
  $PR(A) = PR(B)/L(B) + PR(C)/L(C)$
- Add a damping factor:
  $PR(A) = (1-d)/N + d[PR(B)/L(B) + PR(C)/L(C)]$
- We arrive at a large linear system of equations

13

Parallel Programming Example: Gaussian Elimination

- Solving a system of linear equations
- Reduce an equation matrix into an equivalent upper-diagonal matrix



- Parallelism/concurrency
- Dependencies
- Load balance

14

Parallel Programming

- Challenges:
  - Control dependencies
  - Data sharing
- Parallel algorithms:
  - Identification of parallelisms in applications
  - Design control flow and data sharing mechanism
- Parallel programming:
  - Shared memory
  - Distributed memory

15

Shared Memory Parallel Systems

- In shared memory parallel computers, multiple processors can access the memory simultaneously
  - Examples (large and small)
- Problems:
  - One processor's cache may contain a copy of the data that was just modified by another processor
    ⇨ hardware support for cache coherence
  - Two processes' (semantically) atomic operations may be interleaved
    ⇨ system support for mutual exclusion and synchronization

16

Distributed Memory Parallel Systems

- Parallel systems that do not share memory
  - Examples
- Less requirement on the system support
  - Little or no hardware support
  - Software system support for communications (point-to-point, group)
- More trouble with writing applications
  - Data must be explicitly partitioned and transferred when needed
  - Hard to do dynamic workload management

17

Embarrassingly Parallel Applications

- Require large computing resources ⇨ candidates for parallel computing
- Easily partitioned to fine-grained tasks without inter-task dependencies ⇨ trivial to parallelize
- SETI@home, brute-force password cracking
  - Task: data for processing, passwords to try
- Internet servers
  - Task: request
  - Additional issues: interactivity/QoS/fairness to individual users; 24x7 availability
- Large data center data processing
  - Count the number of web pages containing "#" in Google's databases (mapreduce)

18

What is Cloud Computing?

"The interesting thing about Cloud Computing is that we've redefined Cloud Computing to include everything that we already do. . . . I don't understand what we would do differently in the light of Cloud Computing other than change the wording of some of our ads."

- Larry Ellison, quoted in the Wall Street Journal, September 26, 2008

"A lot of people are jumping on the [cloud] bandwagon, but I have not heard two people say the same thing about it."

- Andy Isherwood, quoted in ZDnet News, December 11, 2008

19

Resource Provisioning Problem

- Provisioning of computing/storage resources for Internet applications
- Challenges:
  - Difficulty to predict load of Internet applications
    - Curse of hugely successful Internet services, DoS attacks
  - Cost of dynamic capacity expansion
  - High upfront cost
    - I want to try some idea and see if people are interested
  - Cost of excess capacity (power)

20

Computing As A Utility

- Computing as a utility (like electricity)
  - Illusion of infinite resources on demand
  - No steep upfront cost, pay as needed (very short-term possible)
- Grid computing
  - High-performance computing applications
- Cloud computing
  - Compute-intensive jobs as well as network services, computing as well as storage

21

Key Idea

- Pooling resources together, managed centrally (in a data center); many computing jobs and network services share the resource pool
- Benefits:
  - While one application's resource needs may vary dramatically, the sum of many independent applications' resource needs varies much less
  - More resource utilization efficiency through sharing
    - Space sharing as well as time sharing
  - Upfront setup cost is amortized over many applications
    - Cost is proportional to use
  - Commoditizing support for availability, data durability, …

22