

# Parallel and Distributed Systems

Instructor: Sandhya Dwarkadas  
TA: Andrew Sexton  
Department of Computer Science  
University of Rochester

1

- What is a parallel computer?
  - “A collection of processing elements that communicate and cooperate to solve large problems fast”
- What is a distributed system?
  - “A collection of independent computers that appear to its users as a single coherent system”

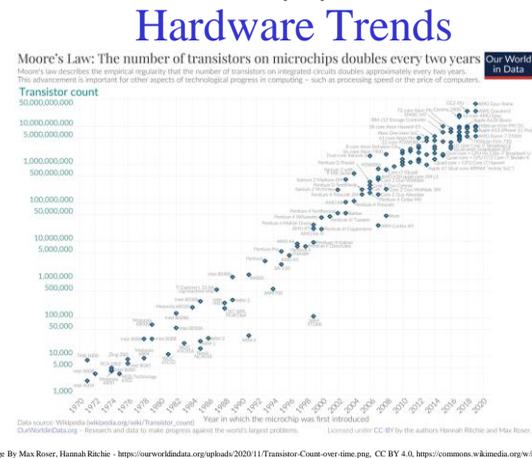
2

## Why Parallel or Distributed Computing?

- Fundamentally limited by the speed of a sequential processor
- Resource sharing
- Information exchange
- Collaboration

3

Moore’s law (attributed to Gordon Moore, Founder of Intel):  
Number of transistors doubles every 2 years



4

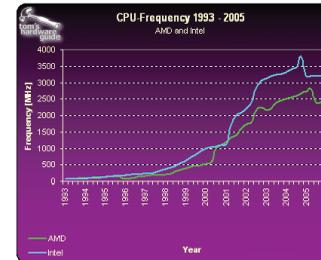
## Leveraging Moore's Law

- More transistors – opportunities for exploiting parallelism
  - Implicit parallelism
    - Pipelining
    - Superscalar
  - Explicit parallelism
    - Streaming and multimedia processor extensions
      - E.g., MMX, AltiVec
    - Very long instruction words (VLIW)

5

## Uniprocessor Limits

The power problem!



[http://www.tomshardware.com/2005/11/21/the\\_mother\\_of\\_all\\_cpu\\_charts\\_2005](http://www.tomshardware.com/2005/11/21/the_mother_of_all_cpu_charts_2005)

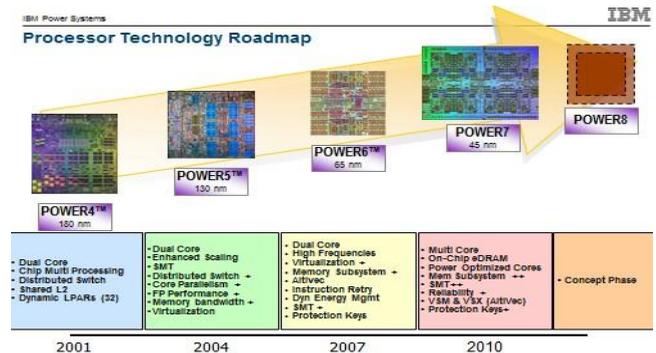
6

## Current Trends

- Problems:
  - Fundamental circuit delay and heat limitations
  - Limited amount of instruction-level parallelism
- Solutions: proliferation of (from Sun, IBM, Intel, Nvidia, ...)
  - Accelerators
  - Multithreading
  - Multicore
  - Multiprocessors

7

## Explicit Parallelism is here to stay!



<http://www.itjungle.com/it/ib050211-story01.html>

8

## Qualcomm's Snapdragon, NVIDIA's Tegra



<http://www.avidia.com/object/tegra.html>



<http://www.gsmarena.com/news.php3?Tag=Qualcomm>

9

## Intel's Many-Integrated-Cores (MIC/Phi)

- 60 cores, 240 threads
- 8 GB memory, 320 GB/s bandwidth
- 1 Teraflop peak double-precision floating point



<http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html>

1 Teraflop

Previously requiring  
2,500 sq. ft!

10

## Nvidia and AMD GPUs



- Nvidia A100 Tensor Core
  - Up to 19.5 Tflops (FP64)
  - Up to 80 GB HBM (High Bandwidth Memory) GPU memory, 2TB/s bandwidth
- AMD Radeon Pro SSG
  - 2 TB of on-board solid state memory!

11

## Top 500 List of Supercomputers ([www.top500.org](http://www.top500.org) – Nov. 2019)

- Top 5 from the list
  - Summit, IBM Power and NVIDIA at Oakridge (ORNL), 2,414,592 cores, 148.6 (200.7 peak) Petaflops/sec on Linpack, 10.1 MW
  - Sierra, IBM Power and NVIDIA at DOE/NNSA/LLNL, 1,572,480 cores, 94.6 (125.7 peak) Pflops/sec, 7.4 MW
  - Sunway TaihuLight, Sunway at NSC in Wuxi China, 10,649,600 cores, 93 (125.4 peak) Pflops/sec, 15.3 MW
  - Tianhe-2A, RIKEN Intel Xeon at NSC in Guangzhou China, 4,981,760 cores, 61.4 (100.6 peak) Pflops/sec, 18.5 MW
  - Frontera, Dell/Xeon at TACC UT Austin, 448,448 cores, 23.5 (38.7 peak) Pflops/sec



12

## Top 500 List of Supercomputers ([www.top500.org](http://www.top500.org) – Nov. 2014)

- Top 5 from the list
  - Tianhe-2, Intel Xeon and Xeon Phi, 3,120,000 cores, 33.8 (54.0 peak) Petaflops/sec on Linpack, 17.8 MW
  - Titan, Cray XK7 at Oakridge (ORNL), AMD Opteron and NVIDIA, 560,640 cores, 17.59 (27 peak) Pflops/sec, 8 MW
  - Sequoia, BlueGene/Q, LLNL, 1,572,864 cores, 17.1 Pflops/sec, 8 MW
  - Fujitsu's K, RIKEN, Japan, SPARC, 705,024 cores, TOFU Interconnect, 10.51 Pflops/sec, 12.6 MW
  - Mira, IBM BlueGene/Q at ANL, 786,432 cores, 8.58 Pflops/sec, 3.9 MW



<https://www.osti.gov/blast>

13

## Top 500 List of Supercomputers ([www.top500.org](http://www.top500.org) – Nov. 2012)

- Top 5 from the list (23 with petaflops performance)
  - Titan, Cray XK7 at Oakridge (ORNL), 560,640 cores, 17.59 Petaflops/sec on Linpack
  - Sequoia, BlueGene/Q, LLNL, 1,572,864 cores, 16.32 Pflops/sec
  - Fujitsu's K, RIKEN, Japan, 705,024 cores, 10.51 Pflops/sec
  - Mira, IBM BlueGene/Q at ANL, 786,432 cores, 8.16 Pflops/sec
  - JUQueen, IBM Blue Gene/Q in Germany, 393,216 cores, 4.14 Pflops/sec



<https://www.osti.gov/blast>

14

## Top 500 List of Supercomputers ([www.top500.org](http://www.top500.org) – Nov. 2008)

- Top 5 from the list
  - IBM Cell-based "Roadrunner", 2008, 129600 cores, 1.059 petaflops/sec
  - Cray XT5 "Jaguar", 2008, 150152 cores
  - SGI "Pleiades", 2008, 51200 cores
  - IBM BlueGene/L, 2007, 212992 cores
  - IBM Blue Gene/P, 2007, 163840 cores



15

## The Green 500 List (Nov. 2019)

- Fujitsu's A64FX prototype (rank 159)
  - 16.876 Gflops/watt
- PEZY's NA-1 (rank 420)
  - 16.256 Gflops/watt
- IBM's AiMOS (rank 24)
  - 15.771 Gflops/watt

16

## Measures of Performance

- Flops/sec
- Flops/watt
- Speedup on N processors: execution time on one processor/execution time on N processors

17

## Why is Parallel Computing Hard?

- Amdahl's Law: Speedup is a function of the fraction  $\alpha$  of the overall execution improved (by a factor of  $k$ ):
  - $T_{new} = (1 - \alpha)T_{old} + (\alpha T_{old})/k$
  - *Overall Speedup* =  $T_{old}/T_{new} = \frac{1}{(1-\alpha) + \frac{\alpha}{k}}$
- Overhead of communication and coordination
- Portability – knowledge of underlying architecture often required

18

## Parallel Systems Issues

- Data sharing – single versus multiple address spaces
- Process coordination – synchronization using locks, messages, ...
- Distributed (non-uniform access – NUMA) versus centralized (uniform access – UMA) memory
- Connectivity – single shared bus versus network with many different topologies
- Fault tolerance/reliability

19

## Shared Memory Parallel Systems

- Multiple processors can access the same memory simultaneously
- Challenges:
  - One processor's cache may contain a copy of data that was just modified by another
    - Requires hardware support for coherence
  - Two processes' operations may be interleaved in unintuitive ways
    - Requires hardware support and guarantees on atomicity and ordering

20

## Multicore Resource Management

- Multicore systems contain many shared resources, e.g., memory bandwidth and cache
- Problem:
  - Management of shared resources for
    - Efficiency, fairness

21

## Distributed Memory Parallel Systems

- Parallel systems that do not share memory
- Software system support for communication (point-to-point, group)
- Data must be explicitly partitioned and transferred when needed
- Dynamic workload management?

22

- What is a parallel computer?
  - “A collection of processing elements that communicate and cooperate to solve large problems fast”
- What is a distributed system?
  - “A collection of independent computers that appear to its users as a single coherent system”

23

## Distributed Systems Issues

- Transparency
  - Access (data representation), location, migration, relocation, replication, concurrency, failure, persistence
- Scalability
  - Size, geographically, administratively
- Reliability/fault tolerance: fail-stop, byzantine (arbitrary) failure models

24

## Distributed Systems Issues

- Problems:
  - Distributed consensus
  - Replication, caching consistency
  - Security and trust

25

## Cluster-Based Servers

- Large Internet data centers (Google, Microsoft, ...)
  - A form of parallel computing
  - Customized parallel programming model (MapReduce)
  - A distributed system
    - Consistency, reliability, scalability, availability
  - Power management, network topology, error monitoring

26

## Course Overview

- Parallel programming
  - models, languages, and algorithms
- Parallel systems design issues
  - E.g., synchronization, communication, coherence, consistency
- Distributed systems design issues
  - E.g., consistency, group communication, fault tolerance

27

## Breakout

Consider the following sequential code initializing two arrays.

```
for (i = 0; i < 100; i++)
```

```
    A[i] = f(x,i);
```

```
for (i = 0; i < 100; i++)
```

```
    B[i] = A[99-i]*f(y,i);
```

How would you parallelize this code in order to maximize performance given unlimited compute and memory resources?

28