CS 550:

Advanced Operating Systems

Introduction to Distributed Systems

Part 3

Ioan Raicu
Computer Science Department
Illinois Institute of Technology

CS 550
Advanced Operating Systems
January 20th, 2011

Key Characteristics of Distributed Systems

- Support for resource sharing
- Openness
- Concurrency
- Scalability
- Fault tolerance (reliability)
- Transparence

Resource Sharing

- Share hardware,software,data and information
- Hardware devices
  - Printers,disks,memory,…
- Software sharing
  - Compilers,libraries,toolkits,…
- Data
  - Databases,files,…

Openness

- Definition?
- Hardware extensions
  - Adding peripherals,memory,communication interfaces…
- Software extensions
  - Operating systems features
  - Communication protocols

Concurrency

- In a single system several processes are interleaved
- In distributed systems: there are many systems with one or more processors
  - Many users simultaneously invoke commands or applications
  - Many servers processes run concurrently, each responding to different client request

Scalability

- Scale of system
  – Few PCs servers ->dept level systems->local area networks->internetworked systems->wide are network…
  – Ideally, system and application software should not change as systems scales
- Scalability depends on all aspects
  – Hardware
  – Software
  – networks

Fault Tolerance

- Definition?
- Two approaches:
  - Hardware redundancy
  - Software recovery
- In distributed systems:
  - Servers can be replicated
  - Databases may be replicated
  - Software recovery involves the design so that state of permanent data can be recovered

Transparency in a Distributed System

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Means that users do not know whether a replica or a master provides a service.
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Pitfalls When Developing Distributed Systems

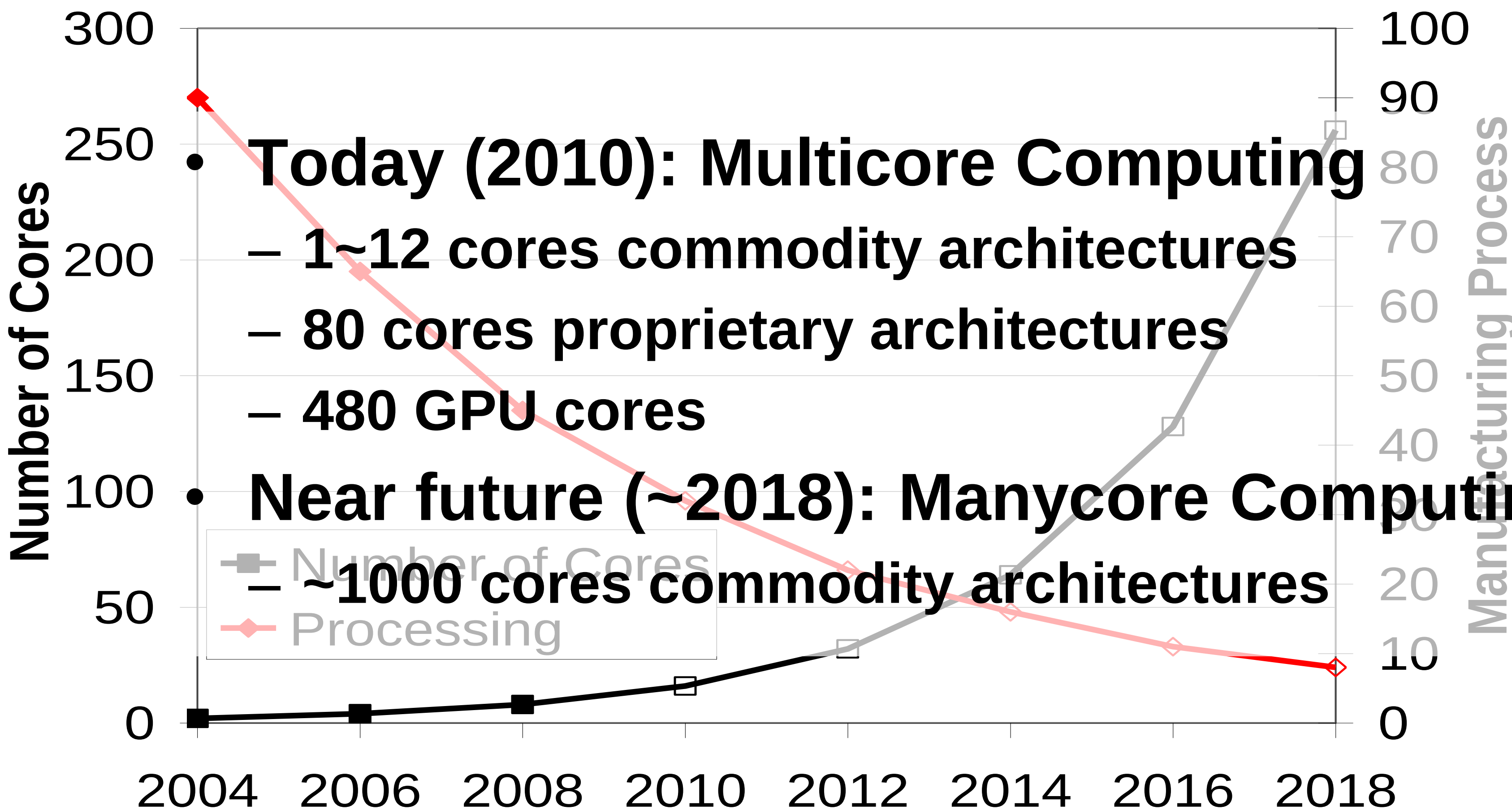
- False assumptions made by first time developer:
  – The network is reliable.
  – The network is secure.
  – The network is homogeneous.
  – The topology does not change.
  – Latency is zero.
  – Bandwidth is infinite.
  – Transport cost is zero.
  – There is one administrator.

Famous Quotes

The advent of computation can be compared, in terms of the breadth and depth of its impact on research and scholarship, to the invention of writing and the development of modern mathematics.

Ian Foster, 2006

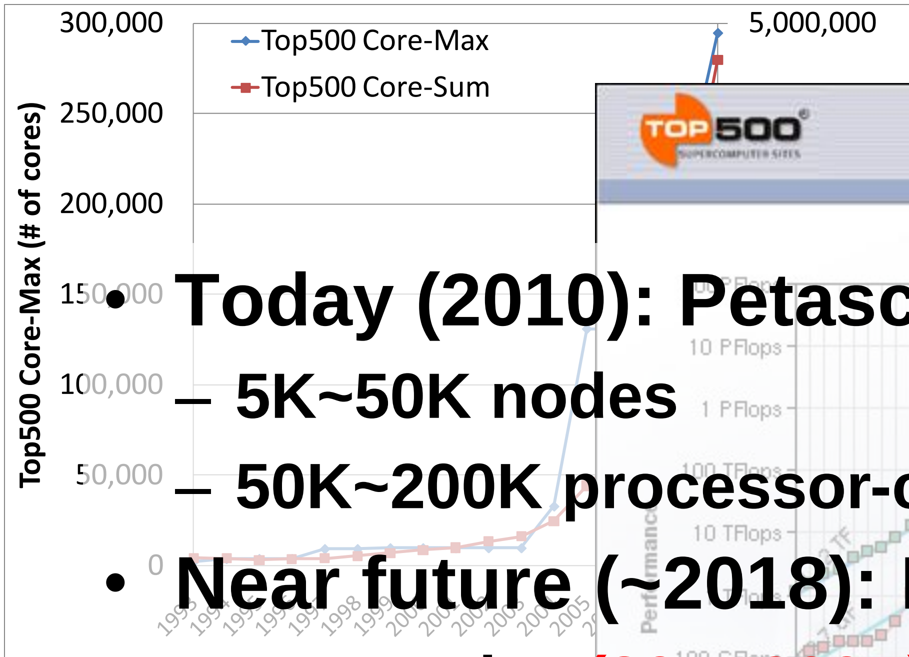
Manycore Computing

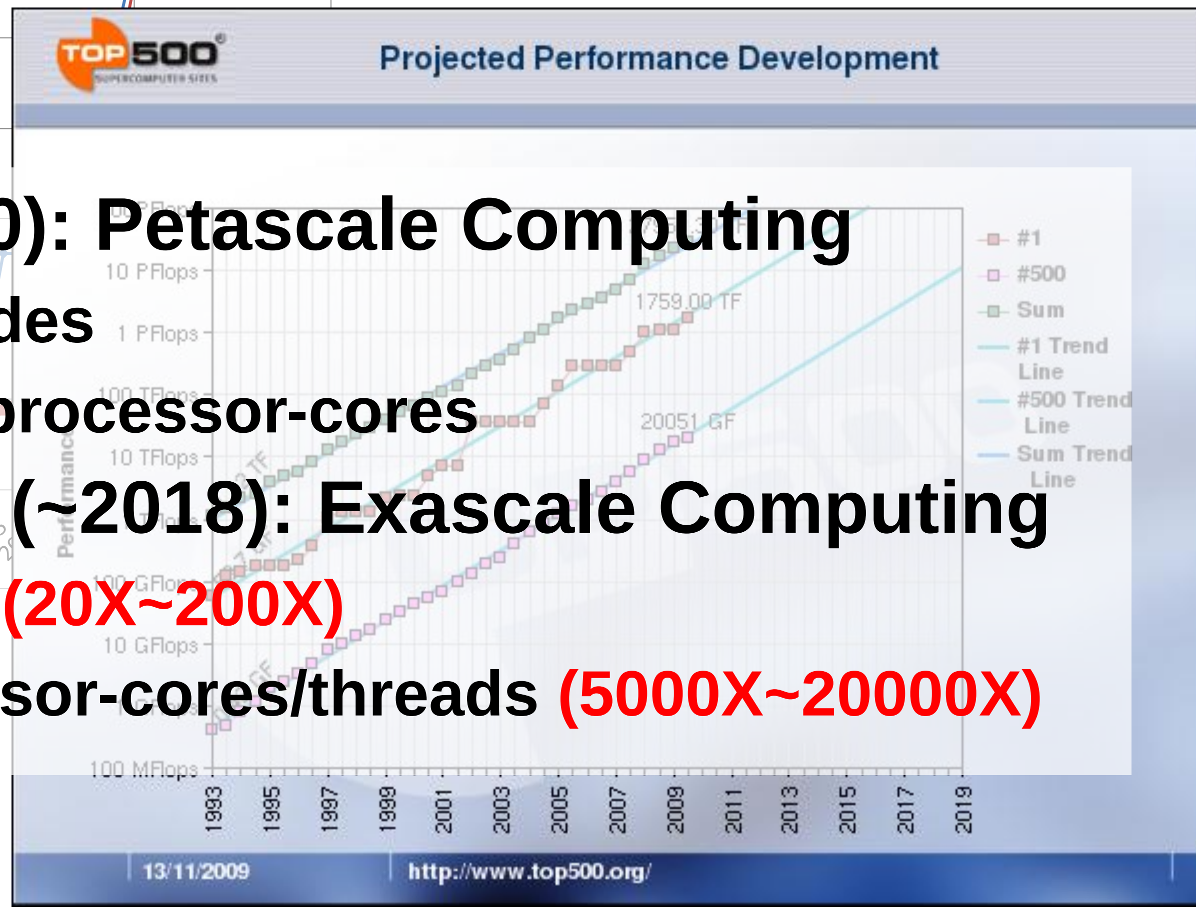


- **Today (2010): Multicore Computing**
  - **1~12 cores commodity architectures**
  - **80 cores proprietary architectures**
  - **480 GPU cores**
- **Near future (~2018): Manycore Computing**
  - **~1000 cores commodity architectures**

Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007

Exascale Computing





- **Today (2010): Petascale Computing**
  - **5K~50K nodes**
  - **50K~200K processor-cores**
- **Near future (~2018): Exascale Computing**
  - **~1M nodes (20X~200X)**
  - **~1B processor-cores/threads (5000X~20000X)**

Top500 Projected Development, http://www.top500.org/lists/2009/11/performance_development

Cloud Computing

- Relatively new paradigm… 3 years old
- Amazon in 2009
  - 40K servers split over 6 zones
    - 320K-cores, 320K disks
    - $100M costs + $12M/year in energy costs
    - Revenues about $250M/year
- Amazon in 2018
  - Will likely look similar to exascale computing
    - 100K~1M nodes, ~1B-cores, ~1M disks
    - $100M~$200M costs + $10M~$20M/year in energy
    - Revenues 100X~1000X of what they are today

Common Challenges

- Power efficiency
  - Will limit the number of cores on a chip (Manycore)
  - Will limit the number of nodes in cluster (Exascale and Cloud)
  - Will dictate a significant part of the cost of ownership
- Programming models/languages
  - Automatic parallelization
  - Threads, MPI, workflow systems, etc
  - Functional, imperative
  - Languages vs. Middlewares

Common Challenges

- Bottlenecks in scarce resources
  - Storage (Exascale and Clouds)
  - Memory (Manycore)
- Reliability
  - How to keep systems operational in face of failures
  - Checkpointing (Exascale)
  - Node-level replication enabled by virtualization (Exascale and Clouds)
  - Hardware redundancy and hardware error correction (Manycore)

My Research: Current Projects

- Falkon: a Fast and Light-Weight Task Execution Framework
  - http://dev.globus.org/wiki/Incubator/Falkon
- Swift: a Parallel Programming System
  - http://www.ci.uchicago.edu/swift/
- FusionFS: Fusion Distributed File System
  - More info soon at: http://datasys.cs.iit.edu/
- D^3: Direct Distributed Data-Structure
  - More info soon at: http://datasys.cs.iit.edu/
- Interested, email me at iraicu@cs.iit.edu

Questions

?