Lecture 36:
Parallel Programming Systems and Models

Ioan Raicu
Department of Electrical Engineering & Computer Science
Northwestern University

EECS 211
Fundamentals of Computer Programming II
May 28th, 2010
• Moore’s Law
  – The number of transistors that can be placed inexpensively on an integrated circuit will double approximately every 18 months.
  – Self-fulfilling prophecy
    • Computer architect goal
    • Software developer assumption
• Impediments to Moore’s Law
  – Theoretical Limit
  – What to do with all that die space
  – Design complexity
  – How do you meet the expected performance increase?
• von Neumann model
  – Execute a stream of instructions (machine code)
  – Instructions can specify
    • Arithmetic operations
    • Data addresses
    • Next instruction to execute
  – Complexity
    • Track billions of data locations and millions of instructions
    • Manage with:
      – Modular design
      – High-level programming languages
• Parallelism
  – Continue to increase performance via parallelism.
• From a software point-of-view, need to solve demanding problems
  – Engineering Simulations
  – Scientific Applications
  – Commercial Applications

• Need the performance, resource gains afforded by parallelism
• Engineering Simulations
  – Aerodynamics
  – Engine efficiency
Introduction to Parallel Computing

- Scientific Applications
  - Bioinformatics
  - Thermonuclear processes
  - Weather modeling
Introduction to Parallel Computing

- Commercial Applications
  - Financial transaction processing
  - Data mining
  - Web Indexing
• Unfortunately, greatly increases coding complexity
  – Coordinating concurrent tasks
  – Parallelizing algorithms
  – Lack of standard environments and support
• The challenge
  – Provide the abstractions, programming paradigms, and algorithms needed to effectively design, implement, and maintain applications that exploit the parallelism provided by the underlying hardware in order to solve modern problems.