Parallel Computing

CS 451/785

thulasir@cs.umanitoba.ca

Introduction and Overview

What's parallel computing

- Is "fast" fast enough??
- Today's workstations are 100 times faster than a decade ago.
- But, computational scientists and engineers need more speed.
  - Make greater simplifications to problems
  - But, wait 5 hours, days and weeks to finish running

One way

- Wait and let the CPU's get faster
- In 5 years, single CPU will be 10 times faster (Moore's law)
- If you can afford to wait 5 years, you must not be in such a hurry!!
- Parallel Computing—proven way to get higher performance

What's Parallel Computing

- Parallel Computing is the use of a parallel computer to reduce the time needed to solve a single computational problem.

What's a parallel computer?

- Multiple processor system supporting
- parallel programming
- 2 categories of parallel computers:
  - Multi-computers :
    - A parallel computer constructed out of multiple computers and a interconnection network
    - Interact through message passing
  - Centralized multiprocessors
    - SMP (Symmetric Multiprocessor)
    - Highly integrated system in which all CPU's share access to a common global memory
    - Shared memory supports communication and synchronization among processors

What's parallel programming?

- Parallel Programming is programming in a language that allows you to explicitly indicate how different portions of the the computation may be executed concurrently by different processors.
  - MPI
  - PVM
  - OpenMP
  - Solaris p-threads
  - HPF (High performance Fortran)

Is parallel programming really necessary??

Programming Parallel Computers

Extend an existing compiler to translate sequential programs into parallel programs

- Lot of research gone into development of compiler technology– Fortran and C
  - Programs translated into codes that can be executed efficiently on parallel computers with large number of processors
    - **Parallelizing compilers**
  - Exploit the parallelism in existing sequential languages
    - Existed for 2 decades
  - Very difficult problem
    - More work done in Fortran
- Companies provided compilers that translate Fortran 77 code intoparallel programs for message passing and shared memory machines
  - Experimental Parallelizing compilers developed (eg. Earth-C compiler at McGill Univ led Laurie Hendren and Guang R. Gao, TERA (now Cray inc.; led by Burton Smith))
    - Very difficult
    - Infancy in commercial systems
- Problem: some parallelism may be lost if compiler not smart enough

Add a new parallel language layer on top of an existing sequential language

- Lower layer:
  - Core of the computation
  - Handled by the sequential language
- Upper layer:
  - Partitions, distributes tasks to processors
- Examples:
  - CODE (Computationally Oriented Display Environments)
  - HENCE(Heterogeneous Network Computing Environment)
- Problem: learn a new Parallel programming system

Create a parallel language

- Start from scratch
- Example: Occam(transputer)
  - Automatic communication and synchronization
  - Could make different topologies
- Add parallel constructs
  - High Performance Fortran (HPF)
  - Fortran 90
- Problem:
  - Portability compromised

Extend an existing language with new operations that allow users to express parallelism

- What is needed?
  - To create and terminate processes
  - Synchronize
  - Communicate with each other
- Easiest, quickest, least expensive
  - MPI and OpenMP
- Subroutine library
- Can use existing compilers
- Portable
- Problem: Lack of compiler support
  - May have debugging problems

quotes

- "Suddenly, even simple tasks, programmed by experienced programmers who were dedicated to the idea of making parallel programming practical reality, seemed to lead inevitably to upsetting, unpredictable, and totally mystifying bugs".(Robert B. Babb II)
- "The behavior of even quite short parallel programs can be astonishingly complex. The fact that a program functions correctly once, or even one hundred times, with some particular set of inputs, is no guarantee thnat it will not fail tomorrow with the same inputs". (James R. McGraw and Timothy S. Axelrod)

Why program in MPI and OpenMP

- MPI (Message Passing Interface)
  - Standard specification for message passing libraries
  - Run on network of workstations (NOW) or parallel computers built out of commodity components, PCs and switches (Beowulf)
  - reusable

- OpenMP
  - SMP's increasing
  - CPUs have a shared address space
  - Between SMPs, MPI better way of communication
  - Among SMPs, OpenMP better way of communication

Modern Scientific Method

- Classical science:
  - Observation
    - Leads to hypothesis
  - Theory
    - To explain the phenomenon
  - Physical experimentation
    - To test the theory
    - Results → refine or reject theory
    - Again observation is centre stage

Modern scientific method

- Contemporary science
  - Observation
  - Theory
  - Experimentation
  - Numerical simulation
    - Cannot use physical experiments alone because too expensive or time-consuming or unethical or impossible to perform
    - Revise theory and/or make more observations

Grand Challenge problems

- Problems that can not be solved in a reasonable amount of time with today's computers.
- Fundamental problem in science and engineering that has a broad economic and social impact and whose solution could be advanced by applying high performance computing techniques and resources.
- Federal agencies: National Science foundation, Department of Energy, NASA, Dept. of defense provide powerful resources

Grand Challenge problems

- Quantum chemistry, statistical mechanics and relativistic physics
- Cosmology
  - Study of universe, evolution and structure
  - Observations from Hubble space telescope and digital sky survey
  - Now, cosmologists simulate their theories→ large memories available
  - Make comparisons of theory with observations and check their compatibility
- Astrophysics
- Computational fluid dynamics and turbulence
  - NASA (Ames)

Grand Challenge problems

- Material design and superconductivity
- Biology, pharmacology, genome sequencing, genetic engineering, protein folding, enzyme activity and cell modeling
  - Hot topic
  - Also called computational biology
- Medicine, modeling of human organs and bones
  - Image processing, 3D CT scans
- Global weather and environmental modeling
  - Advanced Regional Prediction System developed by NSF and technology center for analysis and prediction of storms
  - www.caps.ou.edu

Problem

- CPU performance has exceeded but application performance not matched the scaling of peak speed
- Applications must scale automatically as the number of processors increases.
- Exploit parallelism
- Burden on the application developer

Center for Research on Parallel Computation (CRPC)

- Developed in 1989
  goal : make parallel programming easy
- National Science and Foundation (NSF) Science and technology center
- Research on software and algorithms
- Published their work in top journals
- Rice University, Caltech, ANL (Argonne National Lab), LANL (Los Alamos National Lab), Syracuse Univ, Univ. of Tenessee-Knowxville, Univ. of Texas-Austin

Parallel Programming task

- Besides parallelism, that is needed to determine how to partition the computational tasks,
  - Synchronization and Data movement is very important.
    - For correct answers and high performance
  - And bandwidth from main memory to single memory
- Machine independent –portability
  - Difficult, price of portability is performance
  - Eg. Ocean modeling (BCS code) porting from vector machines to CM2 and CM5
  - Now, satellite data key feature in verification and validation of global ocean models.
- Tasks are complex
  - Programming languages??
  - HPF and MPI – MPI till stands, HPF disappeared
    - Community liking

Parallel Computing Hardware

- US govt. played a key role in
  - Development and
  - Use of high performance computers
- WW II – US army paid for the construction of the ENIAC to speed up the calculation of artillery tables
- 30 years after WW II, US govt. used high performance computers to design nuclear weapons, break codes and perform other national security-related applications
- In 50 years
  - Rapid changes
  - Turnover of vendors, architectures, technologies and system usage
  - Despite all these changes, Moore's law follows

Supercomputers

- Most powerful computers that can be built
- Term widespread with introduction of Cray-1 supercomputer in 1976.
  - A single CPU computer with high performance pipelined vector processor connected to a high performance memory system.
  - Today, supercomputer means 1000s of CPUs
  - Cost $10 million or more– which meant only govt. research labs could use them
  - Late 1970s: supercomputers appeared outside of govt. labs—
    - Petroleum companies– to help them look for oil
    - Automobile manufacturers-improve fuel efficiency, and safety of products

Supercomputers

- 10 years later, business enterprises use these machines—why?
  - Quicker computations lead to competitive advantage
  - More rapid crash simulations reduces the time an automaker needs to design a new car
  - Faster drug design can increase number of patents held by pharmaceutical companies
- Computing speeds increased:
  - ENIAC– 350 multiplications per sec
  - Now, trillions of flops/sec, billions time faster
    - Processor million times faster than ENIAC
    - HIGHER CLOCK RATES
    - GREATER SYSTEM CONCURRENY—work simultaneously on multiple instructions/operations (superscalar machines)
  - High performance microprocessors—Intel Pentium 4 CPU

History

- 1970s:
  - Vector computers
- 1980s:
  - Charles Seitz and Geoffrey Fox at California institute of technology (Caltech)-- 1981
  - Developed Cosmic Cube– 64 Intel 8086 microprocessors
  - Microprocessor based parallel computing
  - Operational in 1983
  - 5 to 10 megaflops (5 to 10 times faster than DEC VAX at that time)
  - Also the value of its parts were less than half the price of the VAX
  - Intel provided much of h/w
  - Story: John Palmer of Intel went to Caltech, was so impressed, quit his job, started his own parallel computing company nCUBE
  - Then at Intel came the new Intel parallel computer division, called Intel Scientific Supercomputing
  - Integration of vector computers into conventional computing environments

History

- Late 1980s:
  - Commercial parallel computers such as BBN (Bolt, Beranek and Newman) – butterfly machine
  - Around the world were selling parallel computers– Parsytec (Germany), C-DAC(India,active, 1991), Myrias (Canada,1987) IBM, NEC, KSR, Sun, SGI
  - Connection machine by Thin king Machines corporation – 1986
  - Also, distributed memory machines came into picture
    - To overcome h/w limitations of SMPs
    - RISC architecture
    - Off the shelf processors

History

- Early 1990s:
- Massively parallel processors
- Top500 list began
  - Statistics of parallel computers
  - How to rank these
    - Used the best LINPACK benchmarks
  - Updated twice a year since 1993
  - SIMD system: multiprocessor systems—156 MPPs (MASPAR)

History

- 1990:
  - mid 1990s:More government labs, HPC programs
  - Research labs were buying parallel systems
  - Industry still in vector computers
    - Distributed machines not mature or stable
    - Industries went bankrupt, taken over.
    - Declined from Top500 list
  - However, SGI, SUN,Hp, DEC (leading industries) had parallel computers
  - IBM-SP2
    - Numerically intensive applications
    - Last half of 1995→ sell to commercial market place : dedicated database system
  - Compared to PC, very expensive → do it yourself attitude

History

- Summer of 1994: NASA Goddard Space Flight Center→ Thomas Sterling and Don Becker built a parallel computer entirely out of commodity h/w and freely available s/w
- Beowulf
  - Ethernet connection
  - Linux OS
  - GNU compilers
  - MPI
  - Supercomputing conference → 1996 $50,000, 1 billion flops (1 Gigaflops)
  - Supercomputing 1997→ 140 node cluster running an n-body simulation at greater than 10 gigaflops

History

- Beowulf:
  - COTS (Commodity off-the shelf) components
  - Unlike commercial systems, commodity clusters are not balanced between compute and communication speed (network quite slow compared to processor speed)
  - Computations dominated by computations, clusters give better performance.
  - Popular platform for academic institutions.

Advanced Strategic Computing Initiative (ASCI)

- US ambitious plan:
  - Series of 5 supercomputers (each $100 million):
    - Plans to maintain its stockpile of nuclear weapons (not create new ones)
    - Numerical simulations to guarantee safety, reliability and performance of the nuclear stockpile.
    - ASCI is trying to develop these fast supercomputers.

ASCI

- ASCI Red
  - 1997
  - Sandia Lab
  - 9000 intel Pentium II Xeon CPUs
  - 1 Teraflops (1 trillion operations per second)
  - Intel dropped out of supercomputer business after this.

ASCI

- ASCI Blue Pacific
  - 1998 from IBM
  - Lawrence Livermore National Lab (LLNL)
  - 5,856 PowerPC CPUs
  - 3 teraflops
- ASCI White
  - 2000 from IBM
  - LLNL (Loas Alamos National Laboratory)
  - SMP based multicomputer
  - 512 nodes; each node with 16 PowerPC CPUs
  - 8192 CPUs → 10 Teraflops

- 2005 :
  - 100 terflops
- Petaflop machines→ 2009
  - SUN, CRAY, IBM

New Parallel Supercomputer

- Japanese Earth Simulator
- Fastest supercomputer
- Developed by NEC
- 35Tflops/s
- 640 supercomputers connected by high speed network (12.3 Gbytes)
- Each supercomputer contains 8 vector processors

So,

- Parallel computers are here to stay!!

Parallel Computing Thesaurus

- *Distributed Computing*: Use of a network of computers to solve a problem. Computers may be heterogeneous, multiuse, usually individual tasks assigned to single processors
- *Heterogeneous Computing*: Use of more than one type of hardware/software system to solve a job
- *Symmetric Multiprocessing (SMP):* Multiple processors sharing a single address space and access to all resources

Thesaurus – Contd.

- *Cluster Computing*: Hierarchical combination of commodity units, such as SMPs, to build larger systems
- *Super Computing:* Use of the fastest, biggest machines to solve big problems. Historically vector computers, but now the norm is parallel or parallel/vector
- *High Performance Computing*: Solving problems via supercomputers + fast networks + visualizations
- *Pipelining:* Breaking a task into steps performed by different unit, with multiple inputs streaming through. Next input starts in unit when previous input done with unit but not necessarily done with task

Thesaurus – Contd.

- *Vector Computing*: Use of vector processors, where operation such as multiply broken into several steps, and is applied to a stream of operands ("vectors"). Most common special case of pipelining
- *Systolic*: Similar to pipelining, but units often not arranged linearly, steps are typically performed in lock-step fashion. Often used in special-purpose hardware such as image or signal processors