Parallel Programming

BSW2012, Yozgat

Cem Özdoğan

ozdogan@cankaya.edu.tr http://siber.cankaya.edu.tr

Department of Materials Science and Engineering, Çankaya University,06530 Ankara, Turkey

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.
- High performance may come from

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.
- High performance may come from
  - fast dense circuitry,

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.
- High performance may come from
  - fast dense circuitry,
  - packaging technology,

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.
- High performance may come from
  - fast dense circuitry,
  - packaging technology,
  - parallelism.

Introduction

- Data-intensive applications; transaction processing, information retrieval, data mining and analysis, multimedia services, computational physics/ chemistry/biology and nanotechnology.
- High performance may come from
  - fast dense circuitry,
  - packaging technology,
  - parallelism.
- Parallel processors are computer systems consisting of multiple *processing units* connected via some *interconnection network* plus the software needed to make the processing units work together.

Field I

- *Uniprocessor* – Single processor supercomputers have achieved great speeds and have been pushing hardware technology to the physical limit of chip manufacturing.

Field I

- *Uniprocessor* – Single processor supercomputers have achieved great speeds and have been pushing hardware technology to the physical limit of chip manufacturing.
  - Physical and architectural bounds (Lithography, μm size, destructive quantum effects.

Field I

- *Uniprocessor* – Single processor supercomputers have achieved great speeds and have been pushing hardware technology to the physical limit of chip manufacturing.
  - Physical and architectural bounds (Lithography, μm size, destructive quantum effects.
  - Proposed solutions are maskless lithography process and nanoimprint lithography for the semiconductor).

Field I

- *Uniprocessor* – Single processor supercomputers have achieved great speeds and have been pushing hardware technology to the physical limit of chip manufacturing.
  - Physical and architectural bounds (Lithography, μm size, destructive quantum effects.
  - Proposed solutions are maskless lithography process and nanoimprint lithography for the semiconductor).
  - Uniprocessor systems can achieve to a limited computational power and not capable of delivering solutions to some problems in reasonable time.

- *Multiprocessor* – Multiple processors cooperate to jointly execute a single computational task in order to speed up its execution.

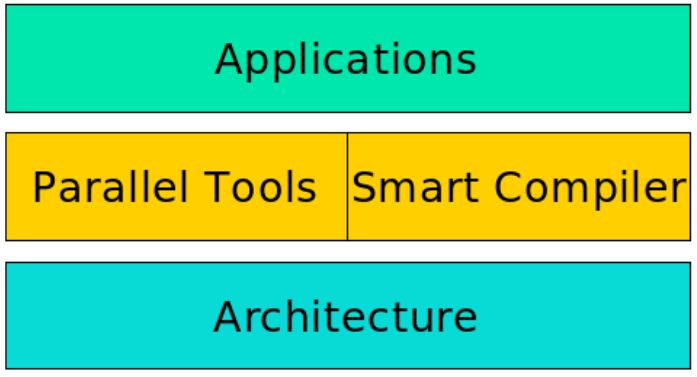

Figure 1: Abstraction Layers

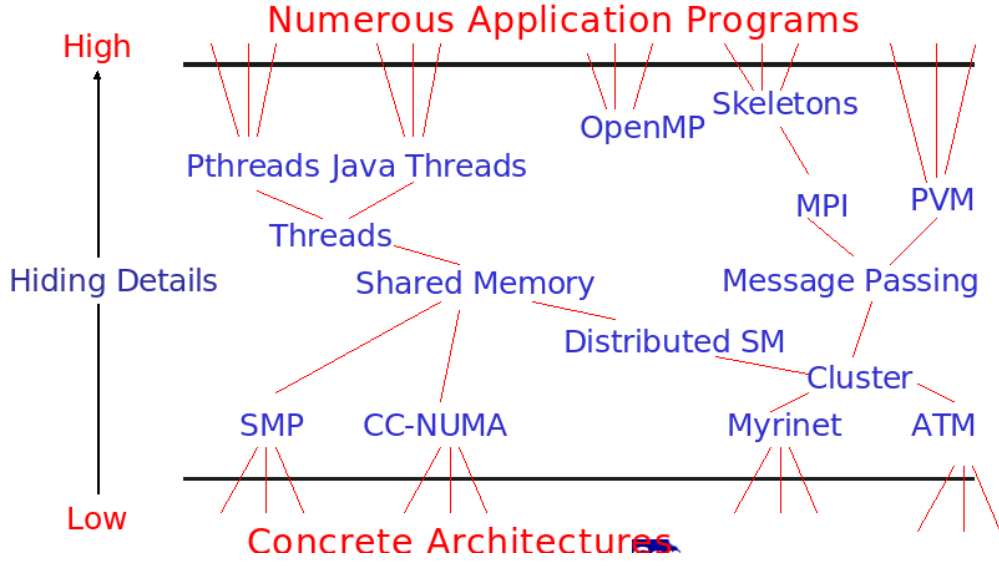

Figure 2: View of the Field

Field III

- New issues arise;

Field III

- New issues arise;
  - Multiple threads of control vs. single thread of control

Field III

- New issues arise;
  - Multiple threads of control vs. single thread of control
  - Partitioning for concurrent execution

Field III

- New issues arise;
  - Multiple threads of control vs. single thread of control
  - Partitioning for concurrent execution
  - Task Scheduling

Field III

- New issues arise;
  - Multiple threads of control vs. single thread of control
  - Partitioning for concurrent execution
  - Task Scheduling
  - Synchronization

Field III

- New issues arise;
  - Multiple threads of control vs. single thread of control
  - Partitioning for concurrent execution
  - Task Scheduling
  - Synchronization
  - Performance

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.

  1. Batch Era

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.
  1. Batch Era
  2. Time-Sharing Era

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.
  1. Batch Era
  2. Time-Sharing Era
  3. Desktop Era

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.
  1. Batch Era
  2. Time-Sharing Era
  3. Desktop Era
  4. Network Era. They can generally be classified into two main categories:

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.
  1. Batch Era
  2. Time-Sharing Era
  3. Desktop Era
  4. Network Era. They can generally be classified into two main categories:

  (a) shared memory,

Four Decades of Computing

- Most computer scientists agree that there have been four distinct paradigms or eras of computing. These are: batch, time-sharing, desktop, and network.
  1. Batch Era
  2. Time-Sharing Era
  3. Desktop Era
  4. Network Era. They can generally be classified into two main categories:
     (a) shared memory,
     (b) distributed memory systems.

Four Decades of Computing

- Distributed memory systems.

Four Decades of Computing

- Distributed memory systems.
  - The number of processors in a single machine ranged from several in a shared memory computer to hundreds of thousands in a massively parallel system.

Four Decades of Computing

- Distributed memory systems.
  - The number of processors in a single machine ranged from several in a shared memory computer to hundreds of thousands in a massively parallel system.
  - Examples of parallel computers during this era include Sequent Symmetry, Intel iPSC, nCUBE, Intel Paragon, Thinking Machines (CM-2, CM-5), MsPar (MP), Fujitsu (VPP500), and others.

Four Decades of Computing

- Distributed memory systems.
  - The number of processors in a single machine ranged from several in a shared memory computer to hundreds of thousands in a massively parallel system.
  - Examples of parallel computers during this era include Sequent Symmetry, Intel iPSC, nCUBE, Intel Paragon, Thinking Machines (CM-2, CM-5), MsPar (MP), Fujitsu (VPP500), and others.
- Current Trends: Clusters, Grids.

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.
- Flynn's classification scheme is based on the notion of a stream of information.

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.
- Flynn's classification scheme is based on the notion of a stream of information.
  - Two types of information flow into a processor:

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.
- Flynn's classification scheme is based on the notion of a stream of information.
  - Two types of information flow into a processor:
    1. **Instruction**. The instruction stream is defined as the sequence of instructions performed by the processing unit.

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.
- Flynn's classification scheme is based on the notion of a stream of information.
  - Two types of information flow into a processor:
    1. **Instruction**. The instruction stream is defined as the sequence of instructions performed by the processing unit.
    2. **Data**. The data stream is defined as the data traffic exchanged between the memory and the processing unit.

Flynn's Taxonomy of Computer Architecture

- The most popular taxonomy of computer architecture was defined by Flynn in 1966.
- Flynn's classification scheme is based on the notion of a stream of information.
  - Two types of information flow into a processor:
    1. **Instruction**. The instruction stream is defined as the sequence of instructions performed by the processing unit.
    2. **Data**. The data stream is defined as the data traffic exchanged between the memory and the processing unit.
- According to Flynn's classification, either of the instruction or data streams can be **single** or **multiple**.

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).
- **Parallel computers are either SIMD or MIMD.**

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).
- **Parallel computers are either SIMD or MIMD.**
- The processing units can communicate and interact with each other using either

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).
- **Parallel computers are either SIMD or MIMD.**
- The processing units can communicate and interact with each other using either
  - shared memory

Flynn's Taxonomy of Computer Architecture

- Computer architecture can be classified into the following four distinct categories:
  1. single instruction single data streams (SISD)
  2. single instruction multiple data streams (SIMD)
  3. multiple instruction single data streams (MISD)
  4. multiple instruction multiple data streams (MIMD).
- **Parallel computers are either SIMD or MIMD.**
- The processing units can communicate and interact with each other using either
  - shared memory
  - or message passing methods.

Two broad categories I

1. **Shared memory**. Processors exchange information through their **central shared memory**.
   - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
2. **Message passing.** Also referred to as distributed memory. Processors exchange information through their **interconnection network**.
   - There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*

Two broad categories I

1. **Shared memory**. Processors exchange information through their **central shared memory**.
   - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
2. **Message passing.** Also referred to as distributed memory. Processors exchange information through their **interconnection network**.
   - There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*
   - This is typically done by a **Send/Receive pair** of commands, which must be written into the application software by a programmer

Two broad categories I

1. **Shared memory**. Processors exchange information through their **central shared memory**.
   - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
2. **Message passing.** Also referred to as distributed memory. Processors exchange information through their **interconnection network**.
   - There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*
   - This is typically done by a **Send/Receive pair** of commands, which must be written into the application software by a programmer

Two broad categories I

1. **Shared memory**. Processors exchange information through their **central shared memory**.
   - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
2. **Message passing.** Also referred to as distributed memory. Processors exchange information through their **interconnection network**.
   - There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*
   - This is typically done by a **Send/Receive pair** of commands, which must be written into the application software by a programmer

Two broad categories I

1. **Shared memory**. Processors exchange information through their **central shared memory**.
   - Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
2. **Message passing.** Also referred to as distributed memory. Processors exchange information through their **interconnection network**.
   - There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*
   - This is typically done by a **Send/Receive pair** of commands, which must be written into the application software by a programmer

Two broad categories II

- Data copying and dealing with consistency issues.

Two broad categories II

- Data copying and dealing with consistency issues.
- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.

Two broad categories II

- Data copying and dealing with consistency issues.
- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems. In such systems,

Two broad categories II

- Data copying and dealing with consistency issues.
- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems. In such systems,
  - memory is physically distributed; for example, the hardware architecture follows the message passing school of design,

Two broad categories II

- Data copying and dealing with consistency issues.
- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems. In such systems,
  - memory is physically distributed; for example, the hardware architecture follows the message passing school of design,
  - but the programming model follows the shared memory school of thought.

Two broad categories II

- Data copying and dealing with consistency issues.
- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems. In such systems,
  - memory is physically distributed; for example, the hardware architecture follows the message passing school of design,
  - but the programming model follows the shared memory school of thought.
  - Thus, the DSM machine is a *hybrid* that takes advantage of both design schools.