Overview of Parallel Computing

Timothy H. Kaiser, PH.D.
tkaiser@mines.edu
Introduction

• What is parallel computing?
• Why go parallel?
• The best example of parallel computing
• Some Terminology
Slides and examples at:

http://geco.mines.edu/workshop
What is Parallelism?

- Consider your favorite computational application
- One processor can give me results in $N$ hours
- Why not use $N$ processors -- and get the results in just one hour?

The concept is simple:
Parallelism = applying multiple processors to a single problem
Why do parallel computing?

- Limits of single CPU computing
- Available memory
- Performance

Parallel computing allows:

- Solve problems that don’t fit on a single CPU
- Solve problems that can’t be solved in a reasonable time
Why do parallel computing?

- We can run...
  - Larger problems
  - Faster
  - More cases
  - Run simulations at finer resolutions
  - Model physical phenomena more realistically
Parallel computing is computing by committee

- Parallel computing: the use of multiple computers or processors working together on a common task.
- Each processor works on its section of the problem
- Processors are allowed to exchange information with other processors

Grid of a Problem to be Solved

- Process 0 does work for this region
- Process 1 does work for this region
- Process 2 does work for this region
- Process 3 does work for this region
Simple data parallel program

- Example: integrate 2-D propagation problem

Starting partial differential equation:

\[
\frac{\partial \Psi}{\partial t} = D \frac{\partial^2 \Psi}{\partial x^2} + B \frac{\partial^2 \Psi}{\partial y^2}
\]

Finite Difference Approximation:

\[
\frac{f_i^{n+1} - f_i^n}{\Delta t} = D \frac{f_{i+1,j}^n - 2f_{i,j}^n - f_{i-1,j}^n}{\Delta x^2} + B \frac{f_{i,j+1}^n - 2f_{i,j}^n - f_{i,j-1}^n}{\Delta y^2}
\]
Data Parallel Programming

- One code will run on 2 CPUs
- Program has array of data to be operated on by 2 CPU so array is split into two parts.

```
program.f:
...
if CPU=a then
    low_limit=1
    upper_limit=50
elseif CPU=b then
    low_limit=51
    upper_limit=100
end if
do I = low_limit, upper_limit
    work on A(I)
end do
...
end program
```

```
program.f:
...
low_limit=1
upper_limit=50
do I= low_limit, upper_limit
    work on A(I)
end do
...
end program
```

```
program.f:
...
low_limit=51
upper_limit=100
do I= low_limit, upper_limit
    work on A(I)
end do
...
end program
```
Typical data Parallel Program

- Solving a Partial Differential Equation in 2d
- Distribute the grid to $N$ processors
- Each processor calculates its section of the grid
- Communicate the boundary conditions
Limits of Parallel Computing

- Theoretical upper limits
- Amdahl’s Law
- Practical limits
Theoretical upper limits

- All parallel programs contain:
  - Parallel sections
  - Serial sections
- Serial sections are when work is being duplicated or no useful work is being done, (waiting for others)
- Serial sections limit the parallel effectiveness
- If you have a lot of serial computation then you will not get good speedup
- No serial work “allows” perfect speedup
- Amdahl’s Law states this formally
Amdahl’s Law

- Amdahl’s Law places a strict limit on the speedup that can be realized by using multiple processors.
  - Effect of multiple processors on run time
    \[ t_p = \left( \frac{f_p}{N} + f_s \right) t_s \]
  - Effect of multiple processors on speed up
    \[ S = \frac{t_s}{t_p} = \frac{1}{\frac{f_p}{N} + f_s} \]
  - Where
    - \( F_s = \) serial fraction of code
    - \( F_p = \) parallel fraction of code
    - \( N = \) number of processors
    - Perfect speedup \( t = t_1/n \) or \( S(n) = n \)
Illustration of Amdahl's Law

It takes only a small fraction of serial content in a code to degrade the parallel performance.
Amdahl’s Law provides a theoretical upper limit on parallel speedup assuming that there are no costs for communications.

In reality, communications will result in a further degradation of performance.

fp = 0.99
Some other considerations

- Writing effective parallel application is difficult
- Communication can limit parallel efficiency
- Serial time can dominate
- Load balance is important
- Is it worth your time to rewrite your application
- Do the CPU requirements justify parallelization?
- Will the code be used just once?
Best Example of Parallel Computing

Working on a Committee
(has the same advantages and disadvantages)
<table>
<thead>
<tr>
<th>Committee</th>
<th>Parallel Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Too much for one person</td>
<td>Too much for one processor</td>
</tr>
<tr>
<td>Share work for Factor of N improvement</td>
<td>Share work for Factor of N improvement</td>
</tr>
<tr>
<td>Meeting (no work gets done)</td>
<td>Synchronization  Primitive MPI_BARRIER</td>
</tr>
<tr>
<td>Can have a head</td>
<td>Manager Process</td>
</tr>
<tr>
<td>Every one shares the load evenly (or not)</td>
<td>Every processor shares the load (or not)</td>
</tr>
<tr>
<td>Subcommittee</td>
<td>Processor collection (MPI_Communicator)</td>
</tr>
<tr>
<td>Someone does not do their job on time</td>
<td>Processor collection (MPI_Wait)</td>
</tr>
<tr>
<td>Email or phone call</td>
<td>Point to Point communication (MPI_Send)</td>
</tr>
<tr>
<td>Mailing list</td>
<td>Broadcast (MPI_Bcast)</td>
</tr>
<tr>
<td>Practical size limits</td>
<td>Amdahl’s Law</td>
</tr>
<tr>
<td>Communication and work balance difficult</td>
<td>My Ph.D.</td>
</tr>
</tbody>
</table>
Terms related to algorithms

- Amdahl’s Law (talked about this already)
- Superlinear Speedup
- Efficiency
- Cost
- Scalability
- Problem Size
- Gustafson’s Law
Scalability

Used to indicate a hardware design that allows the system to be increased in size and in doing so to obtain increased performance - could be described as architecture or hardware scalability.

Scalability is also used to indicate that a parallel algorithm can accommodate increased data items with a low and bounded increase in computational steps - could be described as algorithmic scalability.
Other names for Scaling

- **Strong Scaling (Engineering)**
  - For a fixed problem size how does the time to solution vary with the number of processors
- **Weak Scaling**
  - How the time to solution varies with processor count with a fixed problem size per processor
Gustafson’s law (Weak Scaling)

Rather than assume that the problem size is fixed, assume that the parallel execution time is fixed. In increasing the problem size, Gustafson also makes the case that the serial section of the code does not increase as the problem size.

Scaled Speedup Factor

The scaled speedup factor becomes

\[ S_s(n) = \frac{s + np}{s + p} = s + np = n + (1 - n)s \]

called Gustafson’s law.

Example

Suppose a serial section of 5% and 20 processors; the speedup according to the formula is 0.05 + 0.95(20) = 19.05 instead of 10.26 according to Amdahl’s law. (Note, however, the different assumptions.)
Some Classes of machines

Distributed Memory
Processors only have access to their local memory
“talk” to other processors over a network
Some Classes of machines

Uniform Shared Memory (UMA)

All processors have equal access to Memory

Can “talk” via memory
Some Classes of machines

Hybrid
Shared memory nodes connected by a network
Some Classes of machines

More common today
Each node has a collection of multicore chips

Ra has 268 nodes
256 quad core dual socket
12 dual core quad socket
Some Classes of machines

Hybrid Machines

- Add special purpose processors to normal processors
- Not a new concept but, regaining traction
- Example: our Tesla Nvidia node, cuda
Network Topology

• For ultimate performance you may be concerned how you nodes are connected.

• Avoid communications between distant node

• For some machines it might be difficult to control or know the placement of applications
Network Terminology

- **Latency**
  - How long to get between nodes in the network.
- **Bandwidth**
  - How much data can be moved per unit time.
  - Bandwidth is limited by the number of wires and the rate at which each wire can accept data and choke points.
Figure 1.11  Two-dimensional array (mesh).
Tree

Fat tree
the lines get
wider as you
go up

Figure 1.12 Tree structure.
Hypercube

3 dimensional hypercube
Some communications algorithms are hypercube based.
How big would a 7d hypercube be?
Quality depends on what is in the center
Ra’s Topology

Infiniband, DDR, Cisco 7024 IB
Server Switch - 48 Port

Adaptors. Each compute node has one DDR 1-Port HCA

4X DDR=> 16Gbit/sec

140 nanosecond hardware latency

1.26 microsecond at software level
Golden Energy Computing Organization

Front Range High Performance Computing
dedicated to the
Energy Sciences
GECO will put CSM in the HPC Game

- Computational hub for finding new ways to meet the energy needs of our society
- Energy node for Front Range high performance computing

Intended Impacts:
- Advance energy research
- Attract large-scale, multi-group projects
- Foster education in high performance computing
- Promote Front Range high performance computing
GECO Partners

- Colorado School of Mines
- National Renewable Energy Laboratory
- National Center for Atmospheric Research
- National Science Foundation
GECO HPC Hardware/Software: “Ra”

• **Architecture**
  - Dell with Intel quad-core, dual-socket system
  - 2144 processing cores in 268 nodes
    - 256 nodes with 512 Clovertown E5355 (2.67 GHz) (quad core dual socket)
    - 184 with 16 Gbytes & 72 with 32 Gbytes
  - 12 nodes with 48 Xeon 7140M (3.4 GHz) (quad socket dual core) 32 Gbytes each

• **Memory**
  - 5,632 Gbytes ram (5.6 terabytes)
  - 300 terabyte disk
  - 300 terabyte tape back up
  - 16/32 gigabytes RAM per node
  - Library of Congress has 20 terabytes
  - All the films made in the world in 1989 would take up 166 terabytes

• **Performance**
  - 17 teraflop sustained performance
  - like every human on the planet doing 2500 calculations per second
Important Links

- http://geco.mines.edu
  - Main GECO page
  - Contains Links to additional information
- http://geco.mines.edu/guide
  - Users Guide
- http://geco.mines.edu/checkjobs.shtml
  - Ra queues status pages
- http://geco.mines.edu/education.shtml
  - Educational resources
Mio.mines.edu

- New concept in HPC for CSM
- School puts up the money for infrastructure
- Researchers purchase individual nodes
  - They own the nodes
  - Can use other’s when they are not in use
- Started with 4 nodes (Here this week?)
- Have added 12 more = 1.5 Tflop
- [http://inside.mines.edu/mio](http://inside.mines.edu/mio)
Mio Nodes

- Penguin
  - 2 x Dual Intel Xeon X5570 Quad Core 2.93GHz 8MB
    max RAM speed 1333MHz
  - up to 2 x 48GB DDR3-800 REG, ECC (24 x 4GB)
  - 2 x 160GB, SATA, 7200RPM
  - 2 x Intel Xeon Dual Socket Motherboard with
    Integrated Infiniband DDR/CX4 Connections
- Half the size of RA nodes
  - More efficient in power and computation
  - More memory
  - Faster Clock
CUDA1.mines.edu

- NVIDIA Tesla S1070 GPU Computer
- Tech-Fee Money
- 4 Tesla GPUs
- 16GB memory
- 960 processors
- 1U enclosure
- Portland Group
- Cuda
- CULA - Linear algebra
- http://geco.mines.edu/tesla