Optimization and Scalability
on Ranger and Lonestar

Drew Dolgert
model

parallelism, scalability

algorithm

performance libraries

implementation

compiler options

compilation

diagnostics, tuning

runtime environment
# Libraries

<table>
<thead>
<tr>
<th>Performance</th>
<th>Math Libs</th>
<th>Method Libs</th>
<th>Applications</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>gprof</td>
<td>fftw</td>
<td>petsc</td>
<td>Amber</td>
<td>netcdf</td>
</tr>
<tr>
<td>tau</td>
<td>GotoBLAS</td>
<td>scalapack</td>
<td>NAMD</td>
<td>hdf5</td>
</tr>
<tr>
<td>papi</td>
<td>Metis/parmetis</td>
<td></td>
<td>charm++</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKL 10.0</td>
<td></td>
<td>Gamess</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gnu Scientific Library</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exercise Libraries

2.2 Compare libraries and hand-written code
Model of an Algorithm’s Environment

Network → Algorithm → Disk
Model of an Algorithm’s Environment
Model of an Algorithm's Environment

Algorithm

Network

Disk

Memory

Bus
Memory: The Long Pole in the Tent

Relative Memory Bandwidths

- Functional Units
  - Registers
  - L1 Cache ~50 GB/s, ~5 CP
  - L2 Cache ~25 GB/s, ~15 CP
- Processor
- L3 Cache Off Die ~12 GB/s
- Local Memory ~8 GB/s, ~300 CP

Relative Memory Sizes
- L1 Cache 16/32 KB
- L2 Cache 1 MB
- Memory 1 GB
numactl
Use your chip – vectorize.

```c
for (i=0; i<N; i++) {
    y[xldx] = sqrt(psi[i][j]);
}
```

cat /etc/procinfo
Let the compiler roam.

Interprocedural Optimization - -ipo
But watch it.

-g  -O1  -O2  -O3
Optimization Exercise 3.2
This code runs well.

It has to run this many times.
SU \approx \text{cpu-hour}
time = _ + _ + _
Parallel Random Access Model of the Machine

- Multiple processors.
- Single shared memory.
- Every processor accesses memory in unit time.
LogP Model of Machine

• Latency of communication medium
• Overhead of sending and receiving a message
• Gap between two send / receive operations
• Processing units, the number of them.
Bulk Synchronous Parallel Model for Computation

Compute

Communicate

Synchronize

What year?
Exercise

Focus:
• How much work to do.
• When and how much to communicate.
• Structure of communication.

4.1 Analyze a Parallel System
It is seen that for the chosen grid resolution the LES code exhibits linear scalability up to 128 processors and reasonable scalability up to 256 processors.
Logarithms for Regimes

Wall Time (10^{-6}s)

Core Count

Logarithms for Regimes
Efficiency

How much an N-way parallel job does

How much an N serial jobs do
Efficiency

Time for 1-way job/N

Time for N-way job
Highly arbitrary
low water mark

Efficiency vs. Core Count graph.
1.4 Allocation
MadLib
We expect improvements in scalability with increasing problem size.
Communication versus Computation
Weak Scaling

Performance for Argon

Number of Processors

timesteps/s

0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80
Communication Pattern in Time
Amount sent / bandwidth
Number of messages $\times$
latency

What is the time per iteration?
time = \frac{W}{N} + \frac{N}{bw} + 2 \times \text{latency} \\
O(f(N))
It’s LOG!

![Graph showing performance of different systems over days and steps.](chart.png)

- JAC/DHFR (24k atoms): 32 ns/day, 2.7 ms/step
- ApoA1 (92k atoms): 15 ns/day, 5.6 ms/step

It’s LOG!
Section 4.2 Excel Demo of Fluent
Section 4.3 Measure Strong Scaling
2.3 μs
All-to-All Communication

How would you do it? Should you write this yourself?
Right Resources Match Computational Models to Program Models

- All more complex than what we have described.
- Include RAM, flash, disk, tape, WAN.
Profiling and Presents
Speeds and Feeds at Scales