The Current State of High-Resolution Weather Forecasting on Cluster Technology

An Investigation of Code Efficiency on Scalar Technology

Dr. Daniel B. Weber
Center for Analysis and Prediction of Storms
University of Oklahoma
Acknowledgements

- Dr’s Henry Neeman and Kelvin Droegemeier, University of Oklahoma
- Jason Levit, University of Oklahoma
- National Center for Supercomputing Applications (Performance Expedition Workshop, May 2003)
- Pittsburgh Supercomputer Center
- Software Support (Scott Hill)
- PAPI Developers (Kevin London)

ClusterWorld 2003
Note

- I am a meteorologist, not a computer scientist!!! Computers help us do science!
- Scientists need assistance with the terminology and concepts
- A full scale code analysis requires weeks or months of preparation
- The current analysis is the beginning of an in-depth study of the ARPS model running on commodity based clusters
Outline

- Statement of goals
- Introduction and historical perspective of storm-scale prediction on supercomputers
- Application background (ARPS)
- Benchmark/performance summary
- Optimization game plan
- Preliminary results
- Future work

ClusterWorld 2003
Science Goal

- Perform 1-km resolution storm-scale forecasts on a Continental US domain
- Generate ensembles of storm-scale predictions - provide probabilities of specific weather events for public use
- We want to predict severe weather before it happens (warning issuance, etc...)
- The future of numerical weather prediction depends on the efficient use of large clusters (economics are driving HPC)
Optimization Goals

- Need to optimize the model to achieve our science goal
- Keep the code easy to read, important for code maintenance and further development (Meteorologists developed the code)
- The modified code must perform well on both vector and scalar architectures (keep do loops vectorizable)
- Can we achieve 40% efficiency on a single processor?

ClusterWorld 2003
Numerical Weather Prediction

Collect and Process Data

Run Forecast Model on Supercomputer

Create Products

Dissemination to End Users

Make Observations

ClusterWorld 2003
Numerical Weather Prediction

- Break the forecast into grid boxes (finite grid)
- Solve complicated equations within each grid box to account for:
  - wind speed and direction
  - temperature
  - humidity
  - sun heating the ground
  - surface vegetation
  - lakes and oceans
  - clouds, rain, hail, snow
  - terrain
  - turbulence

ClusterWorld 2003
Numerical Weather Prediction

- Over the course of a single forecast, the computer model solves billions of equations.
- Requires the fastest supercomputers in the world -- capable of performing billions to trillions of calculations each second.
- Local cluster (OSCER), photo courtesy of H. Neeman.

ClusterWorld 2003
Weather features are LOCAL!

Rain and Snow
Intense Turbulence
Snow and Freezing Rain
Severe Thunderstorms
Fog
Rain and Snow
Achieving the Goal: First Grids, then 1-km CONUS

ClusterWorld 2003
Why should we care about forecasting the weather?

- Adverse weather impacts the US economy and the lives of you and me... (on the order of billions annually)
- We can reduce losses of both life and property
- Can numerical weather prediction improve storm forecast quality?
- I will present two severe storm predictions

ClusterWorld 2003
Sample Forecasts
Four Years Ago...
Spring 1999: ARPS Forecast Capabilities

- Radar data is key to predicting thunderstorm activity
- “Research Mode” use of radar data, due to problems linking the data to the forecast system
- Our fourth spring thunderstorm season forecast experiment!
High Resolution Grids Require Fine-Scale Observations: NEXRAD Doppler Radar Data
The 3 May 1999 Oklahoma Tornado Outbreak

Copyright 1999 The Daily Oklahoman
Largest Tornado Outbreak In Oklahoma History

- 60+ tornadoes statewide (55 in Norman CWFA)
- Previous state record 26 tornadoes in one day
- First F5 since 1982
- First F5 ever for Oklahoma City

Courtesy Dave Andra, Oklahoma City Area National Weather Service Forecast Office

ClusterWorld 2003
May 3 Tornado Damage
The Forecasts and Warnings

- Severe thunderstorms in 4:30 AM forecast
- Thunderstorm outlooks mention tornadoes
- First warnings issued SW Oklahoma City 4:15 PM
- Short term forecast at 5:40 PM mentions tornadic storms moving into metro by 7:00 PM
- Numerous warnings and detailed statements tracked tornado into and through metro area
- NOTE: No model forecasts of the event!

Courtesy Dave Andra, Oklahoma City Area National Weather Service Forecast Office
May 3, 1999 Forecast Configuration

- Full single-Doppler velocity retrieval using base (Level II) data from the Oklahoma City (KTLX) WSR-88D radar
- Other observations
  - Oklahoma Mesonet, GOES Satellite, Wind Profiler, Surface METARS, MDCRS Commercial Aircraft
- 4-Hour Forecast: 2200 - 0200 UTC
- Cold start (no dynamic data assimilation)
  - Simplest possible configuration
  - More sophisticated have been developed
- Full model physics including radiation, ice microphysics, terrain, surface energy budget
- 3-km spatial resolution

ClusterWorld 2003
CAPS Numerical Forecasts of the May 3 Tornadic Storms

5:00 pm - Model Initialization Time

ARPS Prediction Model
(0 hour forecast)

ClusterWorld 2003

Storm Beyond Velocity Range of NEXRAD

NEXRAD Radar Observations
CAPS Numerical Forecasts of the May 3 Tornadic Storms
5:30 pm - 30 min Forecast

Model Generates the Storm Itself

ARPS Prediction Model
(1/2 hour forecast)

ClusterWorld 2003

NEXRAD Radar Observations
CAPS Numerical Forecasts of the May 3 Tornadic Storms

6:00 pm - 1 hour Forecast

ARPS Prediction Model
(1 hour forecast)

ClusterWorld 2003

NEXRAD Radar Observations
CAPS Numerical Forecasts of the May 3 Tornadic Storms

6:30 pm - 1.5 hour Forecast

ARPS Prediction Model
(1 1/2 hour forecast)

ClusterWorld 2003
Forecasts With and Without NEXRAD Data

2-Hour CAPS Computer Forecast Down to the Scale of Counties

Moore, OK Tornadic Storm

ClusterWorld 2003

NEXRAD Radar Observations
May 8, 2003 Oklahoma City
Tornadic Thunderstorm
Real Time Forecasts Results

ClusterWorld 2003
CAPS Real Time Forecast System

- Web Site: http://www.caps.ou.edu/wx
- Resources: Oklahoma Supercomputing Center for Education and Research (OSCER) 262 Processor Linux Cluster Pentium 4 2.0 GHZ processors
- The prediction system consists of daily forecasts:
  - 27-km Continental US 48 hour forecast initialized at 00Z
  - 9-km Southern Plains 24 hour forecast initialized at 12Z
  - 3-km Southern Plains Storm-Scale 12 hour forecast initialized at 12Z
  - 3-km Southern Plains Storm-Scale 6 hour forecast initialized at 00Z
    the Oklahoma City (KTLX) WSR-88D radar
  - 48-km Continental US 63 hour Ensemble (5) forecasts
    (contributing to the NCEP SREF effort), twice daily

ClusterWorld 2003
CAPS Numerical Forecasts of the May 8, 2003 Tornadic Storms

4:00 pm 9 hour Forecast

ClusterWorld 2003
CAPS Numerical Forecasts of the May 8, 2003 Tornadic Storms

10 hr forecast valid Thu, 8 May 2003, 5 pm CDT (22Z)
Radar, Clouds, MSL Pressure

Thu, 8 May 2003, 5 pm CDT (22Z)
Radar, Clouds, MSL Pressure

5:00 pm 10 hour Forecast    ADAS Radar Analysis
ClusterWorld 2003
CAPS Numerical Forecasts of the May 8, 2003 Tornadic Storms

11 hr forecast valid Thu, 8 May 2003, 6 pm CDT (23Z)
Radar, Clouds, MSL Pressure

Thu, 8 May 2003, 6 pm CDT (23Z)
Radar, Clouds, MSL Pressure

6:00 pm 11 hour Forecast  ADAS Radar Analysis
ClusterWorld 2003
The Problem: Small Scale Weather

- Thunderstorm prediction
  - Predict precipitation, flooding
  - Severe weather include high winds
  - Tornadoes (not yet...need more computer power)

- Winter storm forecasting
  - heavy snow band
  - ice storm intensity and location
Thunderstorm Prediction
(Super Application)

- 1-km storm-scale forecast on a Continental US domain requires:
  - 5500x3600x100 grid points
  - The ARPS contains 3500 computations per grid point per time step
  - A 12 hour forecast requires 7200 time steps
  - The end results needs to be a FORECAST, i.e. available 10 hours in advance

ClusterWorld 2003
Super App:
Thunderstorm Prediction

- Computer resource estimate:
  - 5500x3600x100 grid points x 3500 calc/grid point = 6.9 TFLOPS
  - Current cluster technology (i.e. IA-32 based) 3GHz Pentium 4 provides a peak of 6 GFLOPS/processor
  - Requires 1155 processors assuming a perfect CPU utilization, network, and message passing environment
Super App: Thunderstorm Prediction

- This is within reach of current Cluster implementation (PSC, NCSA, and others) if the NWP code achieves peak performance!!!
- No real world NWP application achieves near peak performance on existing scalar architecture
- How can we achieve better performance?
The earth simulator consists of 5120 vector processes each capable of 8GFLOPS.

The typical vector processor application performance is on the order of 60-80% of peak and thus the “Super App” could be simulated on the Earth Simulator in ENSEMBLE mode (several forecasts with different initial conditions).
Vector - Scalar Architecture Comparison
Vector Architectures

- Fast access to memory
- Streamlined computation units specialized for floating point arithmetic
- Poor integer performance, commonly use scalar architecture chips to perform integer operations.
- Very expensive due to fast memory and highly specialized processors
Scalar Architecture

- Inexpensive and fast sequential processors (Clock rates > 3GHz)
- Memory access is slower than vector architecture due to less expensive and slower chip and bus speeds
- Scalar processors utilize multi-layer cache to minimize memory access latency
- Excellent integer performance
Vector-Scalar Architecture Performance Comparison

- Metrics
  - FLOPS
  - Cost/Flop

- Weather forecast applications achieve 5-15% efficiency on scalar-based cluster platforms

- This efficiency level for applications on scalar architecture is considered “acceptable”

ClusterWorld 2003
In the past, NWP centers used CRAY C90’s (NCEP) and Fujitsu VPP 700 series (ECMWF) vector supercomputers.

NCEP and ECMWF upgraded to scalar architecture based clusters (IBM P690’s).

- Significant tuning is required many man hours (months).

RESULT: Weather applications REQUIRE modified code to run efficiently on scalar technology.

ClusterWorld 2003
Performance Assessment of ARPS
But first...a few words about the code
The Advanced Regional Prediction System (ARPS)

Developed by
The Center for Analysis and Prediction of Storms
at the
University of Oklahoma

1990-Present
ARPS Model

- General features
  - Fully self-contained meso- and storm-scale prediction model
  - Can be run at any resolution (10m to 60km)
  - Fortran 77 with Fortran 90 extensions
    - Fortran 90 version (5.0) released 2003
  - Multiple I/O formats: binary, HDF, NetCDF, GrADS, GRIB, AVS, Vis5D, ASCII, packed binary
  - Extensive in-code documentation and user’s guide (online)
  - Email-based user support system + online FAQ
  - Code available online (http://www.caps.ou.edu/ARPS)
  - Free non-commercial license
  - Designed for all architectures
    - MPI for shared and distributed memory parallel computers
    - UNIX workstations
    - Linux and Windows PCs
ARPS Model

- Dynamics and Numerics
  - Non-hydrostatic and fully compressible
  - Generalized terrain-following vertical coordinate
  - Arakawa C-grid
  - User-defined vertical stretching
  - Polar stereographic, Lambert conformal, Mercator projections
  - 1-D, 2-D, 3-D geometry
  - Split-explicit solution with vertically implicit option
    - 2nd and 4th order quadratically conservative centered differences
    - Zalesak multi-dimensional FCT
    - Multi-dimensional positive definite centered difference (MPDCD)
  - Initialization
    - Horizontally homogeneous (analytic, sounding)
    - 3-D inhomogeneous

ClusterWorld 2003
ARPS Model

- Physics
  - Moist Processes
    - Kuo or Kain-Fritsch cumulus parameterization
    - Kessler, Lin-Tao, Schultz NEM grid-scale microphysics (all highly optimized)
  - Surface and PBL
    - Convective PBL scheme based on TKE formulation
    - Stability-dependent bulk aerodynamic drag for surface heat, momentum, and moisture fluxes
    - Multi-layer diffusive soil model with surface energy budget (multiple soil types in 1 grid cell; API initialization)
    - Full long- and short-wave radiation (NASA code) including cloud interactions, cloud shadowing, and terrain gradient effects
    - 1 km resolution (over US) USDA sfc characteristics data base and pre-processing software; 30 second global terrain database; 3 second for US plus pre-processing software

ClusterWorld 2003
ARPS Components

Data Acquisition & Analysis
- ARPS Data Analysis System (ADAS)
  - Ingest
  - Quality control
  - Objective analysis
  - Archival

Parameter Retrieval and 4DDA
- Single-Doppler Velocity Retrieval (SDVR)
- 4-D Variational Velocity Adjustment
- Data & Thermodynamic Retrieval

Forecast Generation
- ARPS Numerical Model
  - Multi-scale non-hydrostatic prediction model with comprehensive physics

Product Generation and Data Support System
- ARPSPLIT and ARPSVIEW
  - Plots and images
  - Animations
  - Diagnostics and statistics
  - Forecast evaluation

Incoming data
- Lateral boundary conditions from large-scale models
- Gridded first guess
- Mobile Mesonet Rawinsondes
- ACARS
- CLASS
- SAO
- Satellite Profilers
- ASOS/AWOS
- Oklahoma Mesonet
- WSR-88D Wideband

ClusterWorld 2003
Sample Equation Set for ARPS

\[
\begin{align*}
\frac{\partial u}{\partial t} &= -(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z}) - \frac{1}{\rho} \frac{\partial p}{\partial x} + \text{Turb} + \text{Cmix} \\
\frac{\partial v}{\partial t} &= -(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z}) - \frac{1}{\rho} \frac{\partial p}{\partial y} + \text{Turb} + \text{Cmix} \\
\frac{\partial w}{\partial t} &= -(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z}) - \frac{1}{\rho} \frac{\partial p}{\partial z} + \text{Turb} + \text{Cmix} \\
\frac{\partial p}{\partial t} &= -(u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z}) - \rho \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)
\end{align*}
\]

- Plus 7 more equations!
ARPS Benchmarks
Severe thunderstorm simulation (May 20, 1977 in Oklahoma City) using 1-km horizontal resolution

2-hour forecast (the benchmark involved a 10 minute model simulation)

The test applies the same grid size and code but a different part of the domain to each processor

Processors are added to measure the network performance

ARPS Benchmark
ARPS Benchmark

- ARPS benchmarks performed on several computer platforms including:
  - DEC/COMPAQ/HP Alpha
  - IBM Power
  - SGI MIPS
  - CRAY J/C90 and T3x
  - NEC SX-4 and SX-5
  - Intel PII, PIII, P4, Itanium
  - Sun Systems
ARPS Benchmark Results

ARPS Benchmark Timings
19x19x43 3km grid/processor

- Itanium 733MHZ
- CAPS Origin 2000
- Platinum 1proc/node
- Platinum 2proc/node
- NCSA Origin 2000
- PSC ES-45
- PSC ES-40
- IBM WHII Power3
- IBM NHII Power3
- IBM Regatta Power4
- P4-1.6Ghz w/P3 Compile
ARPS Benchmark Cost Analysis

ARPS Dollars/MFLOP
19x19x43 3km grid/processor

$/MFLOPS vs. Processors

ClusterWorld 2003
Non-commodity clusters (COMPAC, IBM, SGI, etc) offer better performance than Beowolf type clusters due to more attention (and costs) to the details (cpu, memory, disk, network).

But in terms of price/performance, the commodity clusters are highly competitive.
ARPS Optimization
ARPS Optimization History

- 13 years of code development and optimization
- A focused effort during 1997-1998 yielded 20-33% improvement on computers ranging from IA-32 to Vector processors (combined loops and saved redundant calculations, etc...)
- Optimization of the ARPS on the SX-5 platform applied loop fusion (do loop limits = array limits)
- Performance increases, in terms of FLOPS, on the order of 600% were observed on a vector machine for small vector length problem sizes (inner loop < processor vector length)
ARPS SX-5 Optimization Results

ARPS SINGLE LOOP SX-5 PERFORMANCE

MFLOPS

NX (inner loop length)

ClusterWorld 2003
Complete System Optimization

- System components
  - Single processor performance
  - Parallel applications (mpi, pvm, hpf)
  - I/O - file system access

- Perform a single processor performance analysis for this study
- Present results in terms of processor peak for a typical loop calculation
Generic Optimization Strategy

- Assess the code, isolate the subroutines requiring the most cycles (apply performance tools e.g. Perfex, Speedshop, PAPI, Apprentice)

- Three main issues
  - memory bound
  - compute bound
  - I/O bound

- Scalar processor performance will be compared to vector processor performance!
Scalar Optimization Strategy

- Key points:
  - Remove excess computations (divides, exponential, etc.)
  - Remove IF’s from within Do Loops
  - Memory access is slower than vector architecture, maximize data reuse
  - Rethink the order/layout of the computational structure of code (a very difficult and time consuming task)
ARPS Scalar Architecture Optimization Strategy

- ARPS 100,000 lines of Fortran code
- Timing statistics are included in the simulation model (subroutines and MPI)
- Identify the largest CPU related subroutines
- Assess the MFLOPS and memory references
- Important: generate vector friendly code!
ARPS Optimization
Case #1

ClusterWorld 2003
Case #1 Optimization Strategy

- Loop Optimization
  - Combine loops further, the result is reduced loads and stores. This is very important on the new scalar processor technology
  - Need to understand the order of execution of the code, this requires a detailed knowledge of the physical processes!!!
  - Code restructuring is a man-power intensive process

- Compute forcing

\[ u_{force} = u_{force} + u \frac{\partial u}{\partial x} \]
Horizontal 4th Order Advection

DO k=2,nz-2 ! compute avgx(u) * difx(u)
DO j=1,ny-1
DO i=1,nx-1
  tem2(i,j,k)=tema*(u(i,j,k,2)+u(i+1,j,k,2))*(u(i+1,j,k,2)-u(i,j,k,2))
END DO's
DO k=2,nz-2 ! compute avg2x(u)*dif2x(u)
DO j=1,ny-1
DO i=2,nx-1
  tem3(i,j,k)=tema*(u(i-1,j,k,2)+u(i+1,j,k,2))*(u(i+1,j,k,2)-u(i-1,j,k,2))
END DO's
DO k=2,nz-2 ! compute 4/3*avgx(tem2)+1/3*avg2x(tem3)
DO j=1,ny-1 ! signs are reversed for force array.
DO i=3,nx-2
  uforce(i,j,k)=uforce(i,j,k)
  : +tema*(tem3(i+2,j,k)+tem3(i-1,j,k))
  : -temb*(tem2(i-1,j,k)+tem2(i,j,k))
END DO's
Horizontal Advection - Modified Version

Three loops are merged into one large loop that reuses data and reduces loads and stores

- DO $k=2,nz-2$
- DO $j=1,ny-1$
- DO $i=3,nx-2$

  $\text{uforce}(i,j,k) = \text{uforce}(i,j,k)$

  $+ \text{tema}*((u(i,j,k,2)+u(i+2,j,k,2))*(u(i+2,j,k,2)-u(i,j,k,2))$

  $+(u(i-2,j,k,2)+u(i,j,k,2))*(u(i,j,k,2)-u(i-2,j,k,2))$)

  $- \text{temb}*((u(i,j,k,2)+u(i+1,j,k,2))*(u(i+1,j,k,2)-u(i,j,k,2))$

  $+(u(i-1,j,k,2)+u(i,j,k,2))*(u(i,j,k,2)-u(i-1,j,k,2))$)

END DO's...
Result - Merged Loops

4th Order East-West Advection Loop Optimization Tests

MFLOPS

- PIII 1GHz
- NCSA Platinum
- NCSA Itanium
- CAPS Origin 2000
- TCS Alpha ES-40

ClusterWorld 2003
Result - Merged Loops

4th Order North-South Advection Loop Optimization Tests

MFLOPS

- PIII 1Ghz
- NCSA Platinum
- NCSA Itanium
- CAPS Origin 2000
- TCS Alpha ES-40

ClusterWorld 2003
Previous ARPS Optimization Benefits

- Optimize loops (ARPS is 95% loops)
  - remove redundant calculations (21% less work)
  - combine work, eliminate calls to operator type subroutines - not good for readability
  - bigger loops are better, less instruction overhead and memory access
  - remove divides!, very slow on any processor
Previous ARPS Optimization Costs

- Larger memory footprint (13 additional 3-d arrays) but memory is more than sufficient with parallel runs (MPI)
- Increased granularity - more work done in upper levels routines
- Code is less modular, more complex loops
ARPS Memory Requirements

- ARPS contains 150 3-D arrays (per processor)
- A typical forecast sub domain has on the order of 23x23x53 grid points (16+MB)
- The majority of the computations are performed on the small time step for thunderstorm simulations
- The small time step requires 42 3-D arrays
- Approximately 44 3-D arrays can fit into the 4mb O2000 cache
- Result: ARPS will not fit into any current or near future cache system...
ARPS Optimization
Does an Alternative Approach Exist?
ARPS Optimization
Case #2
The Case for Tiling
Tiling can be defined as the process to which the original domain of computation is split up into smaller sections that can fit into the top level cache.

The goal of tiling is to reuse data in the L2 (or L3) cache as much as possible prior to computing the next region.

This approach requires the changing of do loop limits to perform calculations on the sub-domain.

The goal is to tune the application to fit the tile region within the cache and achieve enhanced data reuse and application performance.
GOAL: Develop a strategy for modifying the forecast model to achieve better scalar architecture single processor performance.

The selected loop is similar to fourth order computations (advection, computational mixing) and turbulent mixing.

Use PAPI to access the performance counters on my Dell Pentium III laptop (2 hardware counters).
Tiling Test Methodology

- Evaluate L1 and L2 cache instruction and data misses and loads/stores as well as FLOPS and the Translation look aside buffer (TLB) as a function of problem size
- Adjust problem size to assess memory hierarchy behavior and patterns
## Pentium III Coppermine Specs

<table>
<thead>
<tr>
<th>Operation</th>
<th>Throughput</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>1 cycle</td>
<td>3 cycles</td>
</tr>
<tr>
<td>Multiply</td>
<td>1-2 cycles</td>
<td>5 cycles</td>
</tr>
<tr>
<td>Divide</td>
<td>18 cycles</td>
<td>not-pipelined</td>
</tr>
<tr>
<td>L1 access</td>
<td>32 byte line</td>
<td>3 cycles</td>
</tr>
<tr>
<td>L2 access</td>
<td>256 bit bus</td>
<td>7 cycles</td>
</tr>
<tr>
<td>Load</td>
<td>1/cycle</td>
<td>3 cycles</td>
</tr>
<tr>
<td>Store</td>
<td>1/cycle</td>
<td>1 cycle</td>
</tr>
<tr>
<td>Memory bandwidth</td>
<td>680MB/sec</td>
<td></td>
</tr>
</tbody>
</table>

ClusterWorld 2003
Case #2 Code Description

- **I Loop (Fortran 90)**
  Outer loops increment nz, ny, nx... from small to large
  
  Do n = 1, loopnum    ! Loopnum = 80
  Do k = 1, nz
  Do j = 1, ny
  Do i = 3, nx-2
  Pt(i,j,k) = (u(i+2,j,k)+u(i+1,j,k)-u(i,j,k)+u(i-1,j,k)-u(i-2,j,k))*1.3*n
  End Do’s

- 5 point stencil, reusing data in the i-direction
- Nx, ny, nz are varied to adjust the size of the problem
- U and PT are allocated and deallocated for each change in nx, ny, and nz
- Same tests were performed for reusing data in j and k directions

ClusterWorld 2003
Case #2 Code Description

- **J Loop (Fortran 90)**
  
  ```fortran
  Do n = 1,loopnum  ! Loopnum = 80
  Do k = 1,nz
  Do j = 3,ny-2
  Do i = 1,nx
  Pt(i,j,k) = (u(i,j+2,k)+u(i,j+1,k)-u(i,j,k)+u(i,j-1,k)-u(i,j-2,k))*1.3*n
  End Do's
  ```

- **K Loop (Fortran 90)**

  ```fortran
  Do n = 1,loopnum  ! Loopnum = 80
  Do k = 3,nz-2
  Do j = 1,ny
  Do i = 1,nx
  Pt(i,j,k) = (u(i,j,k+2)+u(i,j,k+1)-u(i,j,k)+u(i,j,k-1)-u(i,j,k-2))*1.3*n
  End Do's
  ```
ARPS Optimization
Case #2 FLOPS Results
Pentium III Flops vs Problem Size (data)

Data Size (Kbytes)

Mflops

I Loop Flops
Pentium III Flops vs Problem Size (data)

- Mflops vs Data Size (Kbytes)

Legend:
- J Loop Flops
Pentium III Flops vs Problem Size (data)

Data Size (Kbytes)

Mflops

K Loop Flops
I Loop L1 and L2 Cache Misses

- L1 Cache Misses
- L2 Cache Misses

ClusterWorld 2003
J Loop L1 Cache Load and Store Misses

Data Size (Bytes)

Cache Misses (Occurrences)

L1 Load Misses
L1 Store Misses

ClusterWorld 2003
Performance (FLOP rating) of scalar architecture is linked to the length of the inner most loop, larger inner loop ranges utilized data in the L1 cache more efficiently – similar to VECTOR architecture behavior!!!

I loop performance (flop rating) is reduced due to smaller inner most loop bounds and the resultant minimal use of L1 cached data.

There is an unexplained reduction in I-loop performance when data size is > 4-8 Kbytes and appears to be linked to L1 load and store misses.
Tiling Experiments Summary

- J and K loop performance >40% of peak for problem data sizes < L2 cache
- FLOP rating not dependant of the data size with respect to the L1 cache
- Significant reduction of throughput (factor of 2) for the J and K loops was observed when the data size > L2 cache
- 256KB cache allows 1454 grid points within the ARPS small time step (27x1x53)
Problem size DOES matter! (due to memory hierarchy)
The results from this study clearly indicate that smaller problems run faster on IA-32 architecture due to cache memory hits/misses
The behavior of simple loops can help us devise a strategy for optimizing pieces of the code or the entire application
Future Work

- Investigate the behavior of loop length (can loop fusion assist the compiler in optimizing the loop?)
- Modify the code to include do loop limits as parameters set at run time
- Investigate the possibility of restructuring the order of computation (grouping forcing terms in a more clever fashion to reduce memory references)
Implications to Real Codes

- Tiling may be an effective approach to improve code performance on scalar architecture
- Complex codes, such as the ARPS, may require significant modifications to allow for variable do loop indexing
- The simple loop performance results provide hope that we can obtain a significant improvement in the overall model throughput
- Tiling algorithm must allow for vector techniques via resetting of parameters (such as an input file setting etc...)
- Depending on the application, tiling can be man power intensive
- BUT Scalar Optimization may REQUIRE A FUNDAMENTAL SHIFT IN PORTING CODES TO NEW PLATFORMS!
Application Optimization Needs

- Provide further assistance to users for use in understanding how to improve performance
- Manuals exist for the processor specifications, but not in a user friendly package
- National Centers and Cluster builders must collaborate with chip makers to improve processor understanding
- Single processor performance tuning workshops will help!
Thank you for your attention!

A copy of this presentation can be found at: http://www.caps.ou.edu/~dweber/cw2003dbw.ppt