Performance Experience with the p690

Patrick H. Worley
Trey White
Oak Ridge National Laboratory

SCICOMP6
Berkeley, CA
August 20, 2002

Acknowledgements

- Research sponsored by the Atmospheric and Climate Research Division and the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.
- These slides have been authored by a contractor of the U.S. Government under contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes
- Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the United States Department of Energy under Contract No. DE-AC05-00OR22725.

Talk topics

- What performance you should expect
- Performance quirks and bottlenecks
- Performance optimization tips

Performance data repository:
`http://www.csm.ornl.gov/evaluation`

Caveats

- Results here are for a single 32-way SMP node.
- Performance characteristics are complicated, and many issues are not well understood yet.
- Performance characteristics are still changing, due to continued evolution of OS and compilers and libraries.
  - A good thing - performance continues to improve.
  - A problem - advice on how to optimize performance continues to change.

Take nothing for granted. Check back with us (and others) for latest evaluation results periodically. Verify what we tell you with your own codes. Tell us what you find out.

Cheetah configuration

27 Turbo p690 Nodes

- 32 1.3-GHz processors per node
- 8 nodes partitioned into 4 Affinity LPARS, each with 8 processors
- 19 single-image nodes, with memory affinity
- Newer OS kernel, with thread-scheduling affinity
- L3s set to "technical" mode, 512B cache lines
- Dual-plane Colony interconnect through Corsair-PCI-RIO-GX

What performance should you expect?

Looking at…

- Single node performance for important applications
- Performance compared with runs on other similar platforms
- Scaling studies to identify performance bottlenecks

We see…

- 2-3x faster than Eagle
  - IBM SP, 375 MHz Power3, Winterhawk II 4-way SMP node
- Faster than other IBM SP, AlphaServer SC, and Cray T3E-900 systems

…on the same number of processors, up to 32 processors.

Applications

- **AORSA3D (Fusion code)**
- **EVH1 (Astrophysics code)**
- **PCTM (Climate code)**

Other platforms

- **AlphaServer SC at PSC**
  - **750 ES45 4-way SMP nodes (1GHz Alpha EV68)**
  - **Quadrics interconnect with two interconnect interfaces per node**
- **AlphaServer SC at ORNL**
  - **64 ES40 4-way SMP nodes (667MHz Alpha EV67**
  - **Quadrics interconnect with one interconnect interface per node.**
- **IBM SP at the NERSC**
  - **184 Nighthawk II 16-way SMP nodes (375MHz POWER3-II)**
  - **SP Switch2 with two switch interfaces per node**
- **IBM SP at ORNL**
  - **176 Winterhawk II 4-way SMP nodes (375MHz POWER3-II)**
  - **SP Switch with one switch interface per node.**
- **Cray Research T3E-900 at NERSC**
  - **644 Alpha 21164 (EV5) processors running at 450 MHz**
  - **3-D torus interconnect**

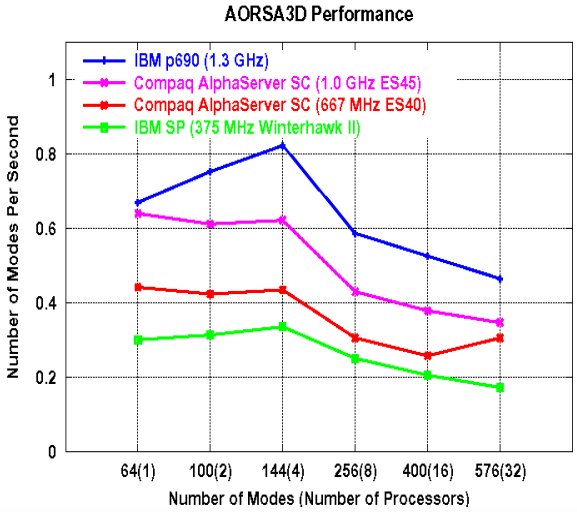
AORSA3D

- Solves for the wave field and heating in a 3-D stellerator plasma heated by radio-frequency waves using an all-orders spectral algorithm
- Important application in the "Numerical Computation of Wave-Plasma Interactions in Multi-dimensional Systems" DOE SciDAC project
- MPI code using ScaLAPACK to solve linear systems from the spectral discretization

AORSA3D performance

Scale-up mode:
Number of Fourier modes increased as the number of processors is increased, keeping the memory size per process approximately constant.

Comparison with Power3:
- **1 P: 2.2x**
- **32P: 2.7x**



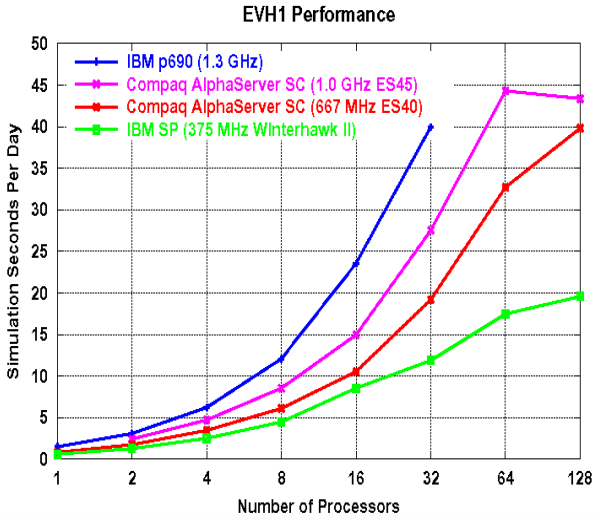
Enhanced Virginia Hydrodynamics #1 (EVH1)

- Important application kernel in the "TeraScale Simulations of Neutrino-Driven Supernovae and Their Nucleosynthesis" SciDAC project
- MPI code that uses second-order operator splitting and 1D Lagrangian hydrodynamics in each coordinate direction
- Uses **`MPI_Alltoallv`** to remap domain decomposition between solves in each coordinate direction

EVH1 performance

Fixed-size problem: Sedov-Taylor blast-wave solution in a 2D spherical geometry using a 128x128 grid

Comparison with Power3:
- **1 P: 2.5x**
- **32P: 3.4x**



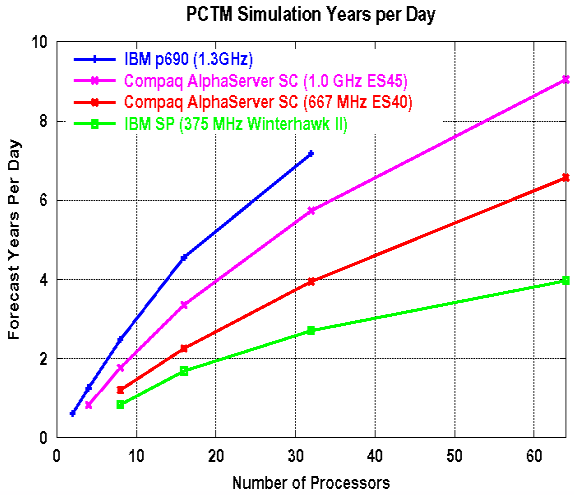
Parallel Climate Transitional Model (PCTM)

- Next generation of the Parallel Climate Model
- Atmosphere, ocean, land-surface, and sea-ice component models, and a coupler to exchange fluxes between the component models
  - Community Climate Model, developed at NCAR
  - POP (Parallel Ocean Program), developed at LANL, the National Physical Laboratory (NPL), and NCAR
  - Components run one after another, sharing the same processors
- MPI only
- Each component uses a different parallelization strategy, and has significantly different performance characteristics

PCTM performance

Fixed-size problem:
Atmosphere: 128x64x18
Ocean: 388x288x40

Comparison with Power3:
- **8 P: 3.0x**
- **32P: 2.7x**



What determines performance?

Looking at…

- Single node performance for representative kernels
- Experiments designed to look at memory contention, compiler optimization, math-library performance, and parallel overheads
- Performance compared with runs on other similar platforms

We see…

- p690 performance is sensitive to memory footprint and access patterns
- Compiler optimizations and math libraries can make a big difference in performance

Kernel codes

- CCM/MP-2D: parallel atmospheric model benchmark
- PSTSWM: serial dynamical-core benchmark
- PCRM: serial physics “core” benchmark
- COMMTEST: interprocessor communication benchmark suite

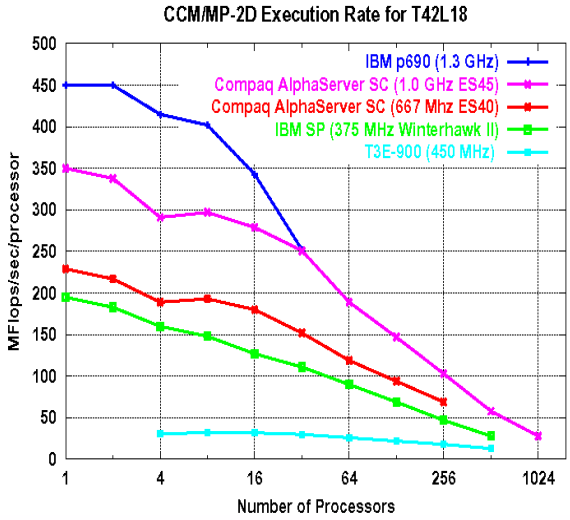
CCM/MP-2D

- Massively parallel implementation of version 3.6.6 of the Community Climate Model (CCM)
- Developed to determine how best to parallelize CCM
  - Results being used in the parallelization of the Community Atmospheric Model (CAM), successor to CCM
- Now used for benchmarking parallel systems
- MPI only
- Numerous parallel-algorithm options are supported
- Tuned specifically for each problem size, processor count, and system when used for benchmarking

CCM/MP-2D performance

Fixed-size problem:
T42L18 (128x64x18)

Comparison with Power3:
- **1 P: 2.3x**
- **32P: 2.3x**



CCM/MP-2D parallel overhead

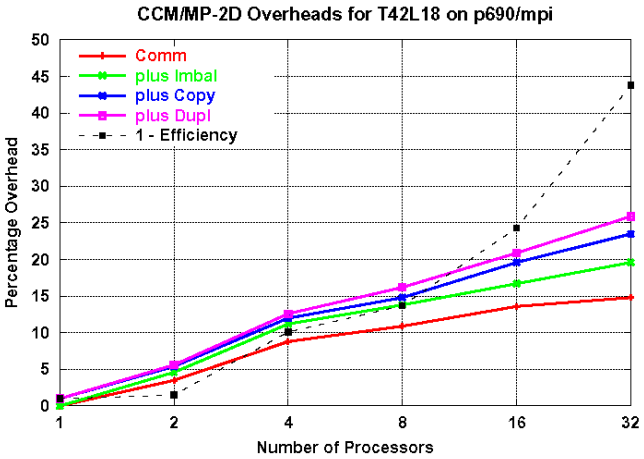
Comm: interprocessor communication costs

Dupl: duplicate computation

Copy: array copies in support of communication

Imbal: load imbalance

1 - Efficiency: changes in computational rate as the problem granularity changes?



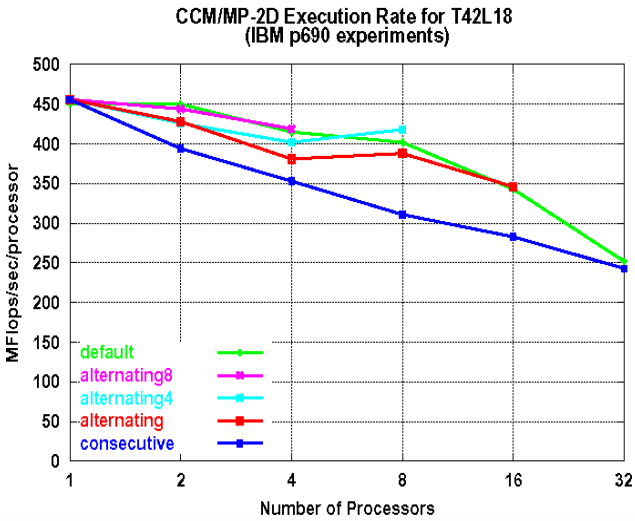
Overhead Time Diagnosis on ORNL IBM p690

CCM/MP-2D performance with binding

Processes are bound to:

- consecutive processors (**consecutive**)
- every other processor (**alternating**)
- every fourth processor (**alternating4**)
- every eighth processor (**alternating8**)

and with runs without explicit binding (**default**)



CCM/MP-2D imbalance and "other"

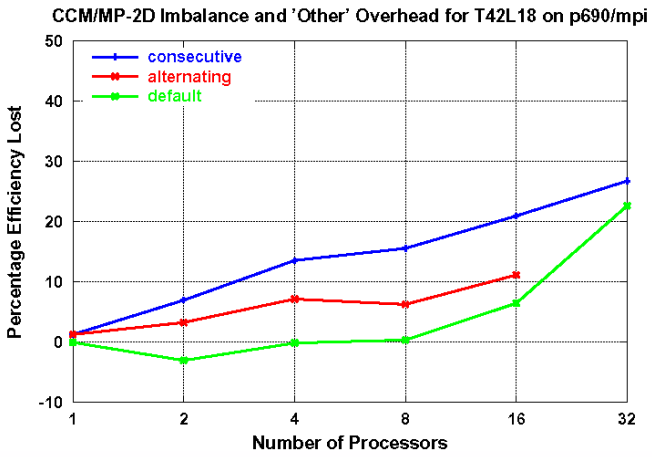
Load imbalance plus unidentified source of performance loss for consecutive, alternating, and default processor bindings.

Overhead categories are nearly identical for all.

Smooth increase in performance loss for consecutive binding

Dramatic increase for the default.

Memory performance issue?



Overhead Time Diagnosis on ORNL IBM p690

Parallel Spectral Transform Shallow Water Model (PSTSWM)

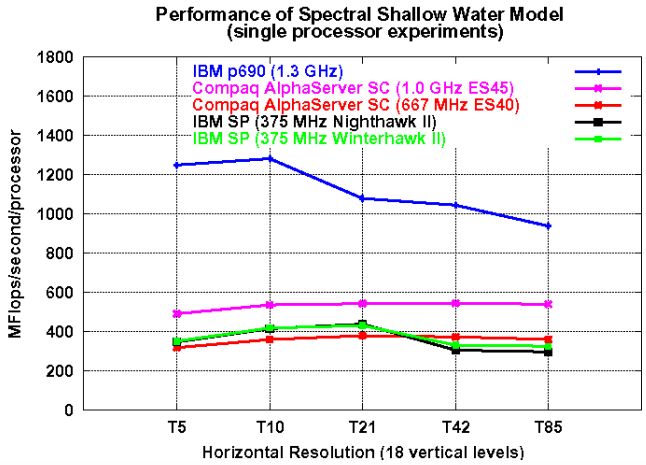
- Important computational kernel in spectral global atmospheric models
- 99% of floating-point operations are multiply or add
  - Runs well on systems optimized for these operations
- Exhibits little reuse of operands
  - Exercises memory subsystem as problem size is scaled
- Can be used to evaluate memory contention
- Following experiments compare performance using one processor and serial benchmark on multiple processors simultaneously
- Performance measured for a range of horizontal resolutions, all with 18 vertical levels

Serial PSTSWM performance

Horizontal Resolutions (all x18)

T5:	8x16
T10:	16x32
T21:	32x64
T42:	64x128
T85:	128x256

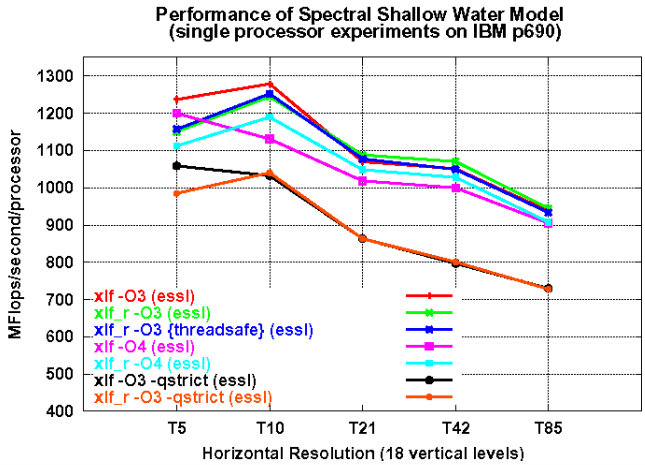
Comparison with Power3: 2.5-3.5x



Platform comparisons - 1 processor per SMP node

PSTSWM compiler comparisons

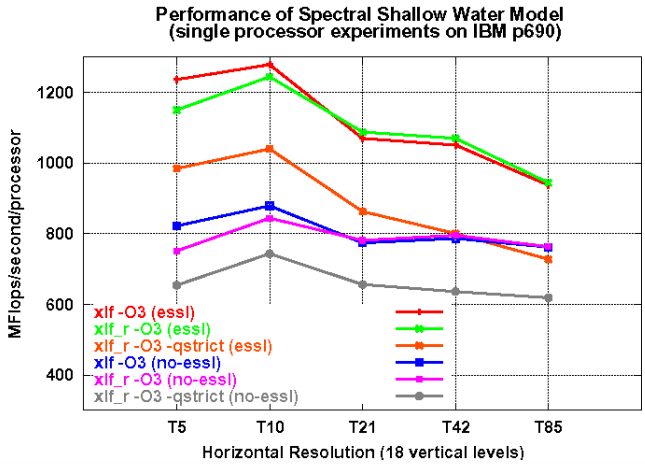
- Non-re-entrant better for small problems
- `-O3` better than `-O4`
- `-qstrict` can have a large negative impact



Compiler optimization comparisons - 1 processor per SMP node

PSTSWM with/without ESSL

- **Very sensitive to the performance of FFT**
- **ESSL FFT performance is much better than that of Fortran**



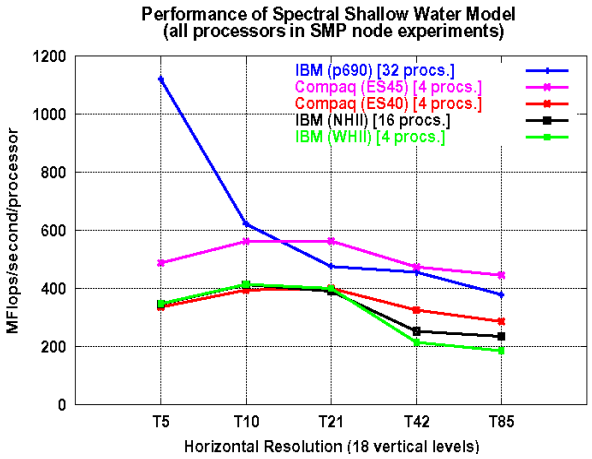
ESSL performance evaluation - 1 processor per SMP node

PSTSWM with 32 serial jobs

Per processor performance when solving same problem simultaneously on all processors in SMP node (using -O3 on p690)

Comparison with Power3: 1.2-3.2x

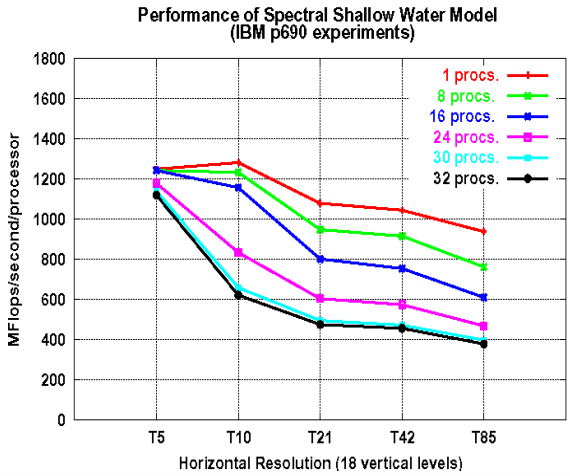
Contention decreases performance significantly when using all processors in the p690 SMP node



Platform comparisons - all processors per SMP node

PSTSWM sensitivity to contention

- Performance degradation is a "smooth" function of number of processes

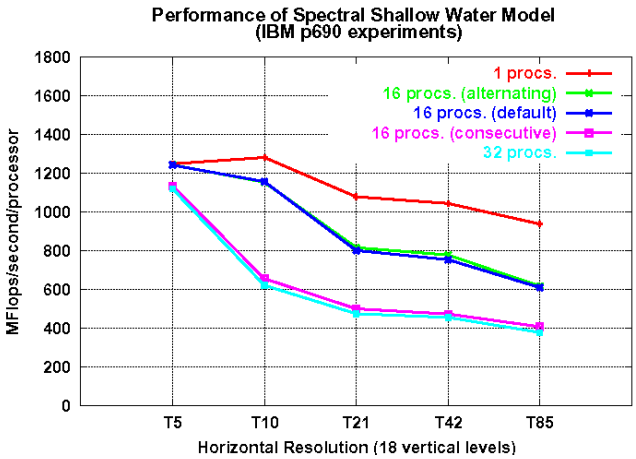


Process scaling on IBM p690

PSTSWM with processor binding

Performance for consecutive, alternating, and default binding for 16 processes:

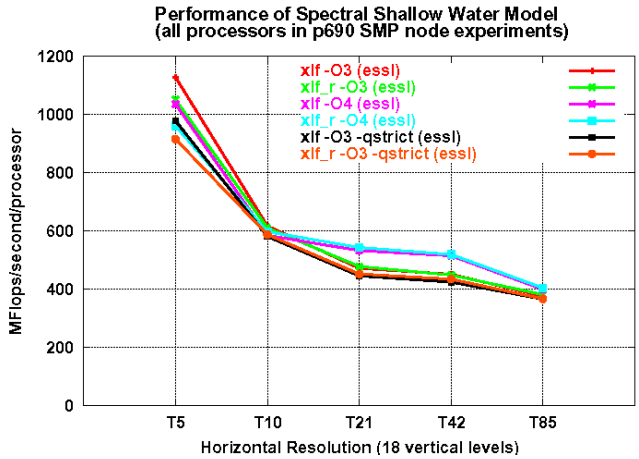
- **16 processes bound to first 16 processors same as 32 processes**
- **Performance degradation solely from memory contention?**



Process binding comparisons - 16 processes in SMP node

PSTSWM compiler comparisons

- T10 and larger, only significant difference is `-O4` or not
- `-O4` worse for tiny problems
- `-qstrict` does not hurt `-O3` performance
- For T85, all options are equivalent



Compiler optimization comparisons - all processors in SMP node

Parallel Column Radiation Model (PCRM)

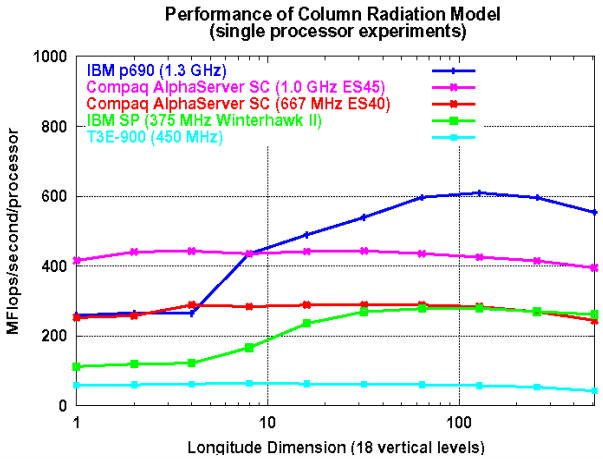
- Slightly modified column radiation model from CCM 3.6.6
- Important computational kernel in atmospheric models
- Differs significantly from PSTSWM
  - Over 6% of the floating-point operations are divide or square root
  - Much better operand reuse
  - Vastly different demands on processor and compiler
- Fixed problem size of 512 columns, each with 18 vertical levels
  - Lengths of longitude and latitude dimensions are varied, keeping product at 512

Serial PCRM performance

Curves show performance using optimal compiler options for each platform.

Performance curve similar to that of Power3

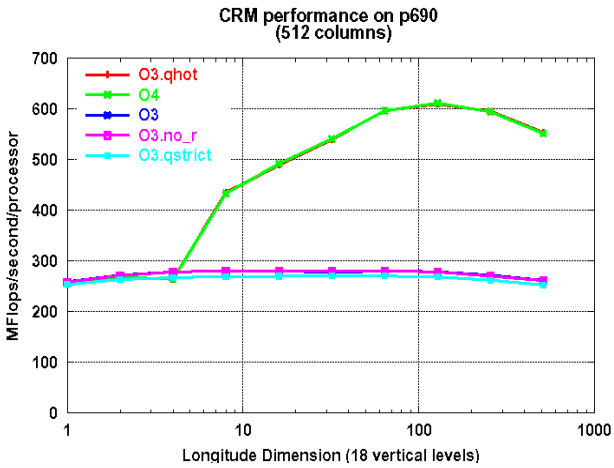
Comparison with Power3: ~2.3x



Platform comparisons - 1 processor per SMP node

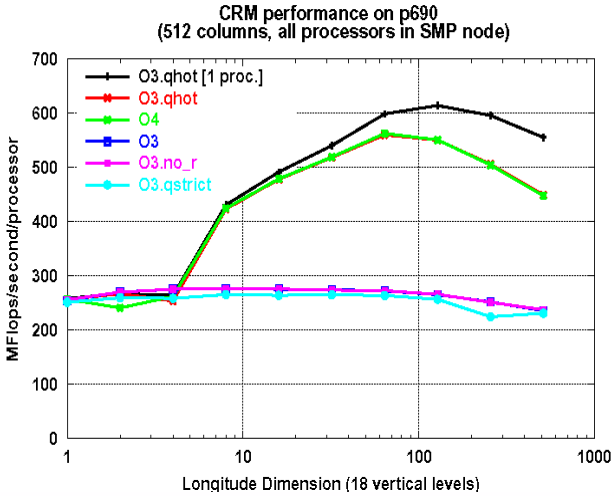
PCRM compiler comparisons

- `-O3 -qhot` and `-O4` have same performance signatures
- Other options have similar performance signatures
- Vector intrinsics improve performance significantly
- Vector intrinsics not allowed by `-qstrict`



PCRM compiler comparisons, 32 processes

- Qualitatively the same as for single process
- Contention impacts performance for large longitude dimensions, but only 10% worse



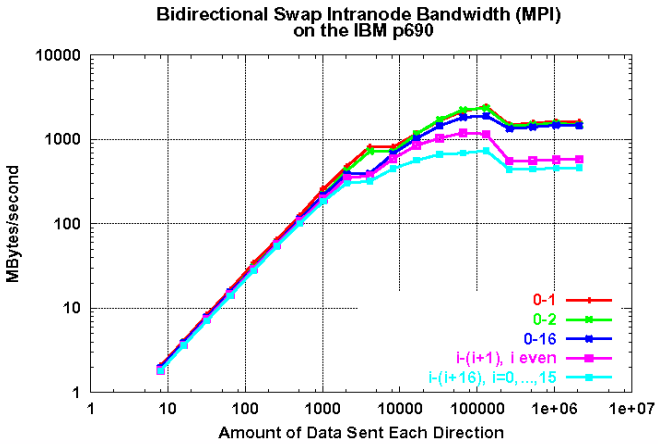
COMMTEST

- Suite that measures performance of MPI
- Evaluates impact of communication protocol, packet size, and total message length
- Bidirectional "swap" bandwidth when one or more processors are communicating within a p690 node
- All within one node
- **`MP_SHARED_MEMORY=yes`**

Performance of swap, log-log

Between processors:

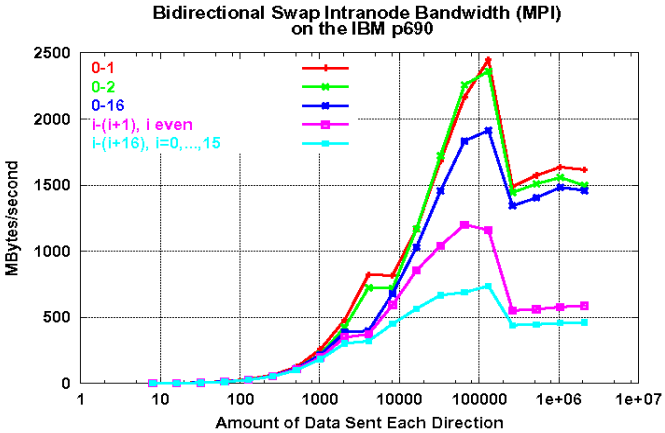
- **0 and 1**
- **0 and 2**
- **0 and 16**
- **0-1; 2-3; …; 30-31, simultaneously**
- **0-16; 1-17; …; 15-31 simultaneously**



Log-Log plot

Performance of swap, log-linear

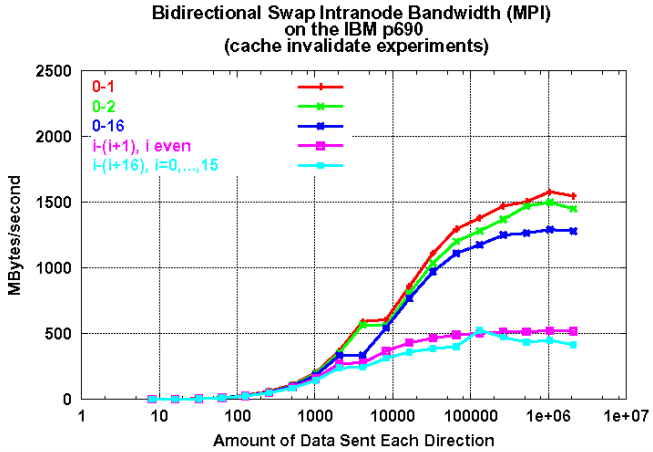
- **Distance between processors has performance impact for certain message lengths**
- **Contention decreases bandwidth per swap 3x for large messages**



Log-Linear plot

Performance of swap, invalidated L2 cache

- Distance between processors still matters
- Contention still matters
- Similar performance for large messages as when not invalidating the cache



Log-Linear plot

Large pages?

Looking at…

- Performance of PSTSWM, with and without large pages
- Performance of CCM/MP-2D, with and without large pages
- Performance of PCM, with and without large pages

We see…

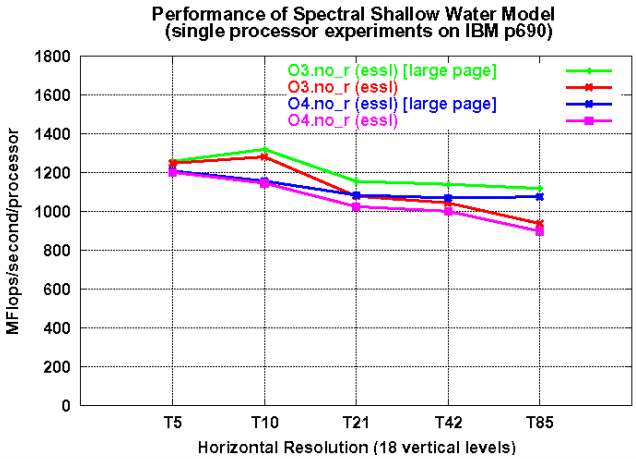
- Large pages improve PSTSWM performance for certain problem sizes, but not when memory-bound
- Large pages have no impact on CCM/MP-2D performance
- PCM is slightly faster (3%) when ***not*** using large pages

Large pages are not a cure-all for memory performance problems.

PSTSWM page-size comparisons, serial

Large pages improve performance 20% for largest problem size.

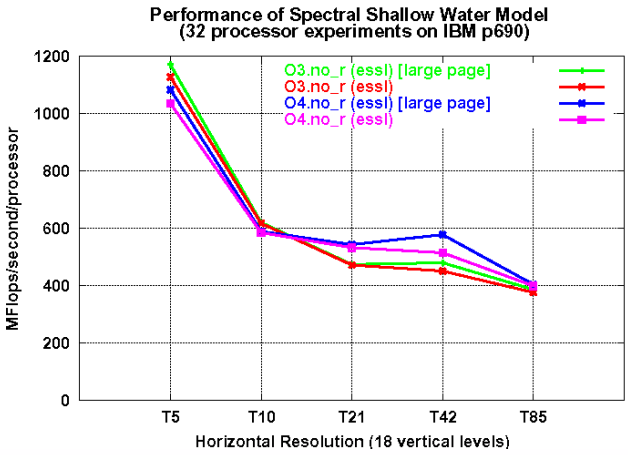
Large pages never decrease performance.



Large page comparisons - 1 processor per SMP node

PSTSWM page-size comparisons, 32 processes

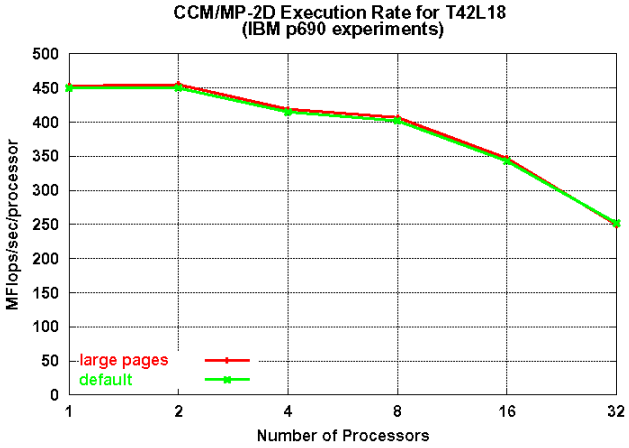
- Large pages are 15% better for one problem size
- Little impact otherwise
- No apparent benefit when bottleneck is L3 or main-memory bandwidth



Large page comparisons - all processors in SMP node

CCM/MP-2D page-size comparisons

No significant performance difference



Multiple jobs on the same node?

Looking at…

- Performance when filling a node with multiple small parallel jobs, assigning one process per processor
- Performance for different binding policies

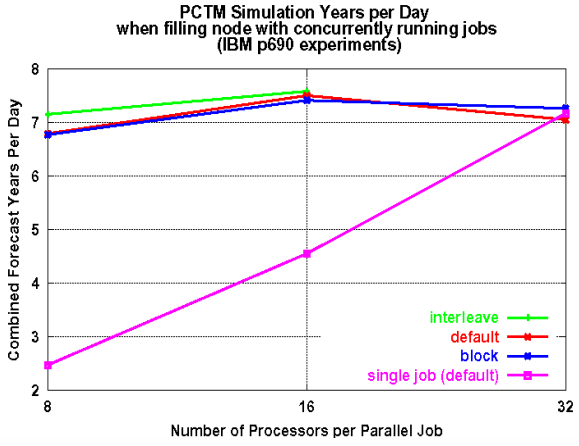
We see…

- Parallel efficiency of a whole-node, fixed-problem-size job less than that of a job using fewer processors
- Multiple jobs each using a small number of processors but solving the same fixed-size problem require more memory than a contend for memory bandwidth
- Node sharing is not necessarily bad (or a good) idea
- Performance is application and problem-size dependent

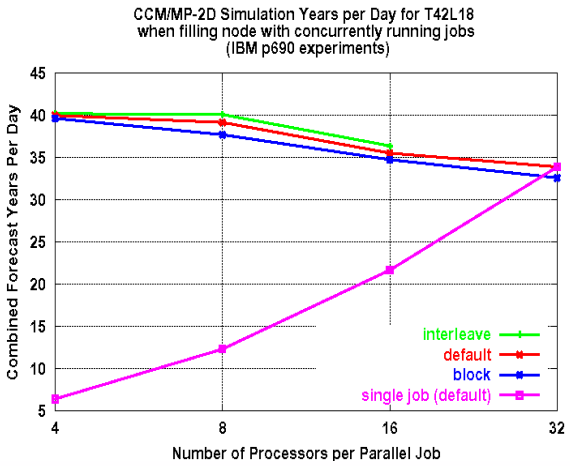
PCTM node-sharing performance

Binding policies:
- **Assign process 0 for each job to consecutive processors, then assign process 1, *etc*. (interleave)**
- **Assigning *P* consecutive processors to each *P*-way parallel job (block)**
- **Letting the system make assignments (default)**

Optimal performance:
- **Two 16-way jobs**
- **Optimum not very strong**
- **Binding policy unimportant**



CCM/MP-2D node-sharing performance

Binding policies:
- **Assign process 0 for each job to consecutive processors, then assign process 1, *etc*. (interleave)**
- **Assigning *P* consecutive processors to each *P*-way parallel job (block)**
- **Letting the system make assignments (default)**

Optimal performance:
- **Eight 4-way jobs**
- **Binding policy unimportant**