

**The 6th International Conference on
Linux Clusters
The HPC Revolution 2005
April 26-28, 2005
Chapel Hill, NC**

**PERFORMANCE METRICS FOR
OCEAN AND AIR QUALITY MODELS
ON COMMODITY LINUX PLATFORMS
George Delic, Ph.D.**

© Copyright, HiPERiSM Consulting, LLC,

<http://www.hiperism.com>



**H
I
P
E
R
I
S
M**

HiPERiSM Consulting, LLC.



George Delic , Ph.D.
HiPERiSM Consulting, LLC
(919)484-9803
P.O. Box 569,
Chapel Hill, NC 27514
george@hiperism.com
<http://www.hiperism.com>

© Copyright, HiPERiSM Consulting, LLC,

<http://www.hiperism.com>



**H
I
P
E
R
I
S
M**

Overview

1. Introduction
2. Choice of Hardware, Operating System, & Compilers
3. Choice of Benchmarks
4. Overview of Benchmark Results
5. Performance Inhibition
6. I/O Performance in Air Quality Models
7. Conclusions
8. Outlook



1.0 Introduction

- **Motivation**
 - AQM's are migrating to COTS hardware
 - Linux is preferred
 - Rich choice of compilers and tools is available
 - Need to learn about portability issues
 - Reliable performance metrics are required
- **What is known about compilers for COTS?**
 - Need a requirements analysis of differences in
 - ✓ Performance
 - ✓ Numerical accuracy & stability
 - ✓ Portability issues



2.0 Choice of Hardware, Operating System, and Compilers



- Hardware
 - Intel Pentium 4 Xeon (3 GHz, dual processor) with SSE2 extensions and 1MB L3 cache
 - Intel Pentium 4 Xeon 64EMT (3.4 GHz, dual processor, no L3 cache)
 - Linux 2.4.20, 2.4.27, 2.6.9 kernels
- Fortran compilers for IA-32 Linux
 - Intel 8.1
 - Portland CDK 5.2
- Fortran compilers for Intel 64EMT Linux
 - Intel 8.1

2.0 Choice of Compilers (cont.)



Compiler & version	Selected switches	Key
Intel 8.1.023	ifort -tpp7 -O0 -Ob0 -unroll0 -FI ifort -tpp7 -O3 -Ob2 -prefetch- -FI ifort -tpp7 -xW -O3 -Ob0 -prefetch- -FI	noopt opt sse
Portland 5.2-2	pgf90 -O0 -tp p7 pgf90 -O2 -tp p7 pgf90 -fast -Mvect -tp p7 pgf90 -fast -Mvect=sse -tp p7	noopt opt vect sse



2.0 Choice of Hardware, (cont.)

- PAPI performance event counters
 - Intel Pentium 4 Xeon has 4 HW counters
 - 26 PAPI events require 17 executions for all data
 - Intel Pentium 4 Xeon 64EMT has ? counters
 - Download from <http://icl.cs.utk.edu/papi>.
 - Kernel patch required for PAPI interface
 - Kernels used: 2.4.20, 2.4.27, 2.6.9



PAPI Events for P4 Xeon

Category	Name	Description
Floating Point Ops	PAPI_FP_INS	Floating point instructions
	PAPI_FP_OPS	Floating point operations
Instruction Counting	PAPI_TOT_CYC	Total cycles
	PAPI_TOT_IIS	Instructions issued
	PAPI_TOT_INS	Instructions completed
	PAPI_VEC_INS	Vector/SIMD instructions
Conditional Branching	PAPI_BR_INS	Branch instructions
	PAPI_BR_MSP	Conditional branch instructions mispredicted
	PAPI_BR_NTK	Conditional branch instructions not taken
	PAPI_BR_PRC	Conditional branch instructions correctly predicted

PAPI Events for P4 Xeon (cont.)



H I P E R I S M

Category	Name	Description
Data Access	PAPI_LD_INS	Load instructions
	PAPI_LST_INS	Load/store instructions completed
	PAPI_RES_STL	Cycles stalled on any resource
TLB Operations	PAPI_TLB_DM	Data translation lookaside buffer misses
	PAPI_TLB_IM	Instruction translation lookaside buffer misses
	PAPI_TLB_TL	Total translation lookaside buffer misses

© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

PAPI Events for P4 Xeon (cont.)



H I P E R I S M

Category	Name	Description
Cache Access	PAPI_L1_DCM	L1 data cache misses
	PAPI_L1_LDM	L1 load misses
	PAPI_L2_DCR	L2 data cache reads
	PAPI_L2_LDM	L2 load misses
	PAPI_L2_STM	L2 store misses
	PAPI_L2_TCM	L2 total cache misses
	PAPI_L3_DCR	L3 data cache reads
	PAPI_L3_LDM	L3 load misses

© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

3.0 Choice of Benchmarks



3.1 SOM

- Serial, MPI and MPI + OpenMP versions

3.2 POM

- Serial and MPI versions

3.3 ISCST3

- Serial version only

3.4 AERMOD

- Serial version only (good parallel potential)

NOTE: Only serial version results are shown here

3.0 Choice of Benchmarks



3.1 Stommel Ocean Model (SOM)

- 2-D finite difference scheme with 5-point stencil
- Compute kernel is a Jacobi iteration
- Double nested loop on $N \times N$ grid
- Fixed problem size at $N = 8000$
- Two versions to avoid vector recurrence:
 - mc - memory copy
 - nomc – no memory copy

3.0 Choice of Benchmarks (cont.)



3.2 Princeton Ocean Model (POM)

- Example of “real-world” code that is numerically unstable with sp arithmetic!
 - 500+ vectorizable loops to exercise compilers
 - 9 procedures account for 85% of CPU time
 - 2-Day simulation for four (i,j,k) grids:
 - Grid 1: 100 x 40 x 15 Scaling = 1
 - Grid 2: 128 x 128 x 16 Scaling = 4.4
 - Grid 3: 256 x 256 x 16 Scaling = 17.5
 - Grid 4: 512 x 512 x 16 Scaling = 69.9
- (Note: Grid 4 on 64 EMT only)*

3.0 Choice of Benchmarks (cont.)



3.3 ISCST3

- Industrial Source Complex Short Term Model is the U.S. EPA’s current regulatory model
- Available from <http://home.pes.com/iscst3.htm>.
- Developed on 286 PC platform (1989-1992)
- Legacy Fortran 77 partially converted to F90
- Small memory and I/O bound footprint

3.0 Choice of Benchmarks (cont.)

3.4 AERMOD

- Proposed replacement for ISCST3
- Also developed on PC platform
- Available from U.S. EPA, Technology Transfer Network, Support Center for Regulatory Air Models <http://www.epa.gov/scram001/>.



4.0 Overview of Benchmark Results

SOM & POM results shown here

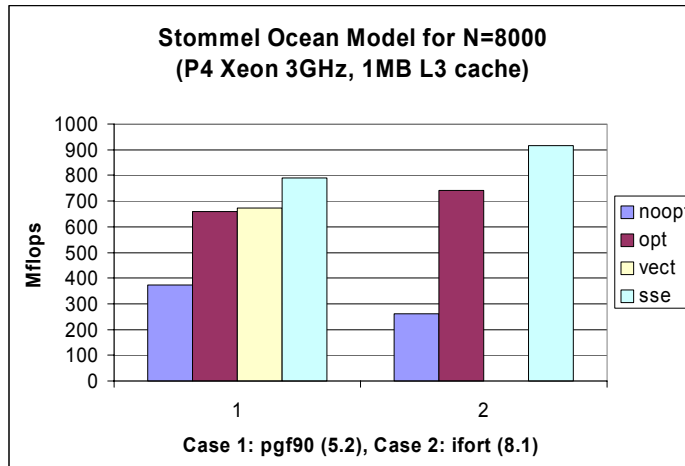
- Mflops
- CPI: Cycles per instruction
- Wall clock time
- Vector instructions

Compilers used

- Intel & Portland compilers (SOM)
- Portland compiler (POM)



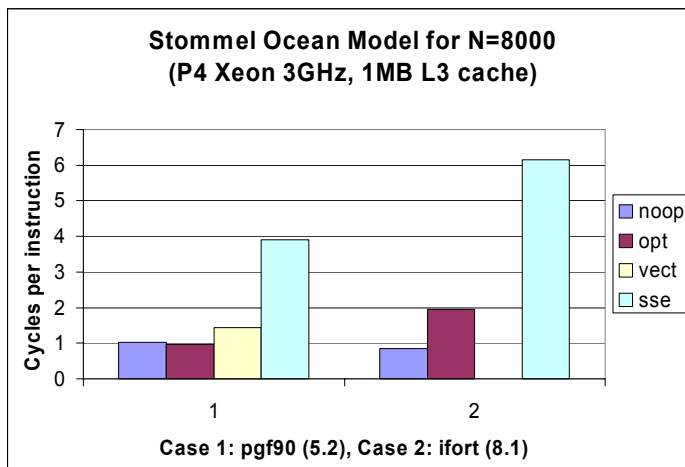
4.0 Overview of Benchmark Results: SOM Mflops



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

4.0 Overview of Benchmark Results: SOM CPI



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

4.0 Overview of Benchmark Results: POM seconds



HIPERISM

GRID	pgf90 noopt	pgf90 opt	pgf90 vect	pgf90 sse
1	190.1	149.6	168.6	145.1
2	2724.0	1566.0	1527.0	1191.8
3	13199.9	7567.9	7603.5	5985.6

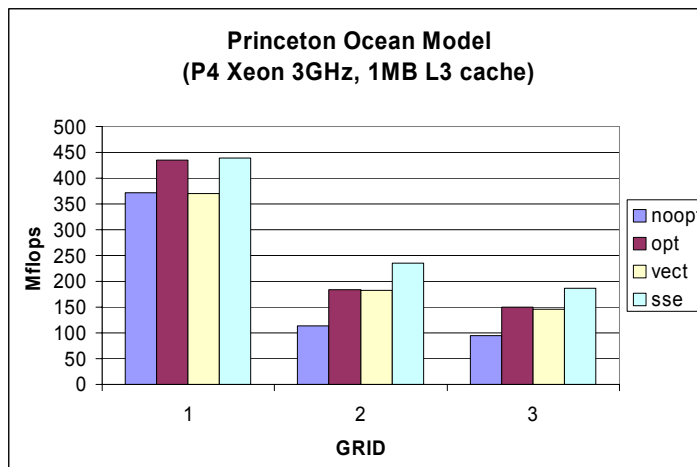
© Copyright, HIPERISM Consulting, LLC,

<http://www.hiperism.com>

4.0 Overview of Benchmark Results: POM Mflops



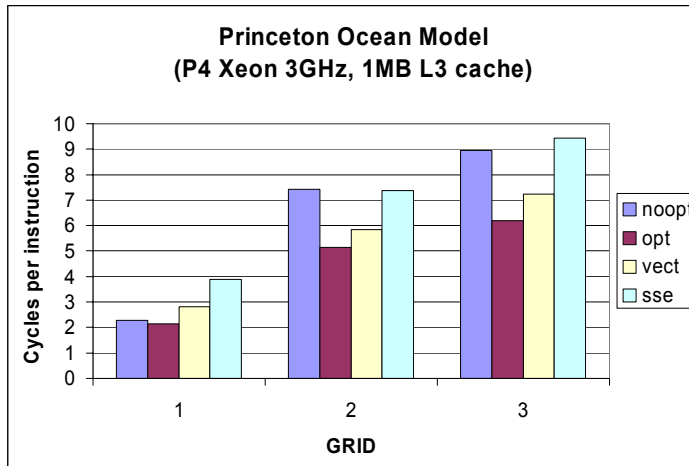
HIPERISM



© Copyright, HIPERISM Consulting, LLC,

<http://www.hiperism.com>

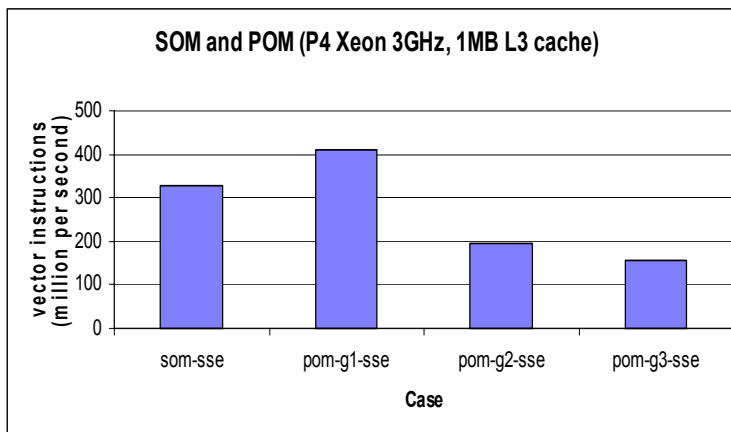
4.0 Overview of Benchmark Results: POM CPI



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

4.0 Overview of Benchmark Results: SOM & POM vector instructions



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>



Summary for SOM & POM

➤ SOM

- ❑ Increasing performance with higher optimization
- ❑ 0.9 Gflops (Intel) and 0.8 Gflops (Portland)
- ❑ With SSE CPI is 6 (Intel) and 3.7 (Portland)
- ❑ CPI has positive correlation with Mflops

➤ POM

- ❑ Increasing performance with higher optimization
- ❑ Performance decreases with problem size
 - 0.44 Gflops for Grid 1
 - 0.23 Gflops for Grid 2 ← WHAT IS THE PROBLEM?
 - 0.19 Gflops for Grid 3 ← WHAT IS THE PROBLEM?
- ❑ With SSE CPI
 - 3.3 for Grid 1
 - 7.2 for Grid 2
 - 9.2 for Grid 3
 - Negative correlation with Mflops

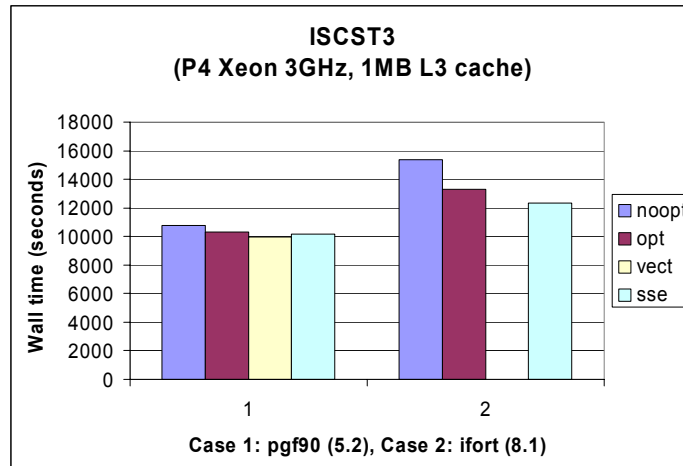


4.0 Overview of Benchmark Results (cont.)

ISCST3 and AERMOD results shown:

- Mflops
- Wall clock time
- Intel & Portland compilers

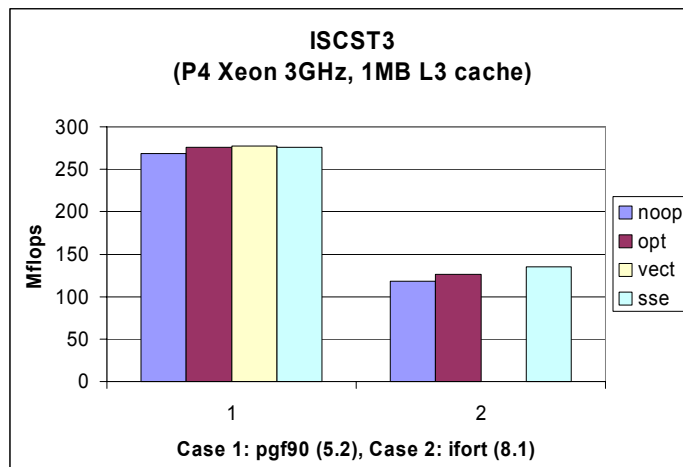
4.0 Overview of Benchmark Results: ISCST3 seconds



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

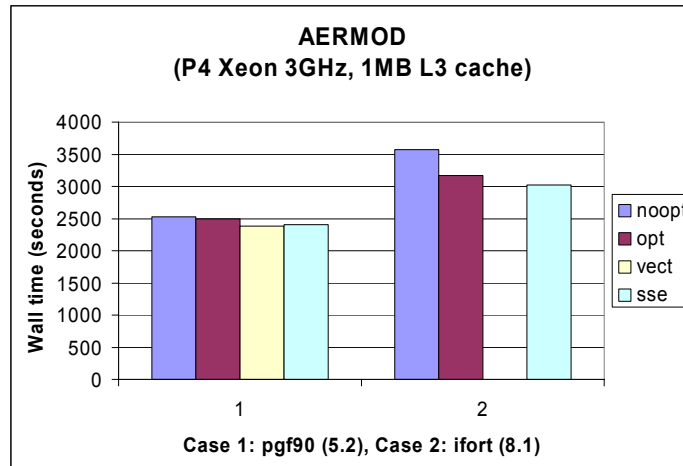
4.0 Overview of Benchmark Results: ISCST3 Mflops



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

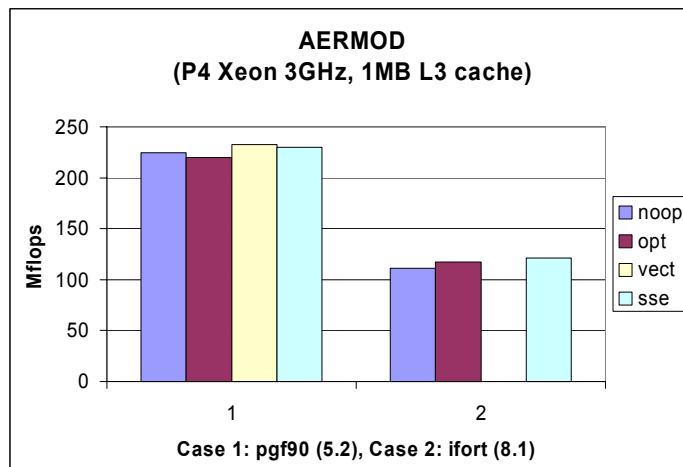
4.0 Overview of Benchmark Results: AERMOD seconds



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

4.0 Overview of Benchmark Results: AERMOD Mflops



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>

Summary for ISCST3 and AERMOD

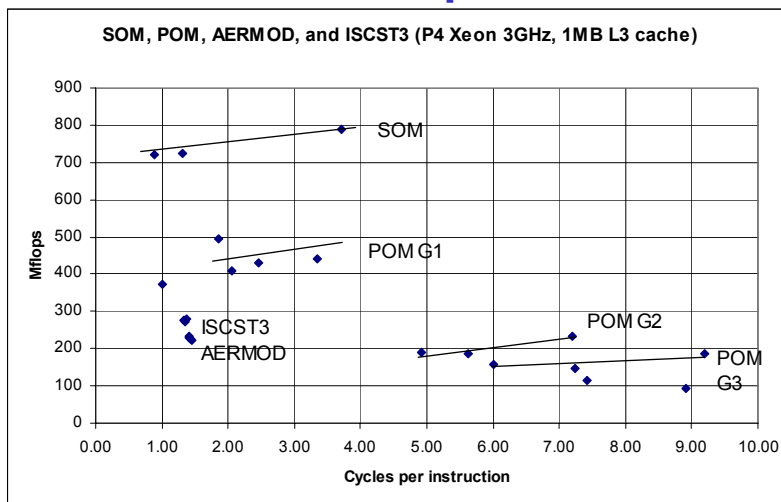


➤ ISCST3 and AERMOD

- ❑ Intel compiler gives a lower performance than the Portland compiler
- ❑ No performance gain with higher optimization
 - 0.28 Gflops (ISCST3)
 - 0.23 Gflops (AERMOD)
- ❑ CPI
 - 1.35 (ISCST3)
 - 1.41 (AERMOD)
 - No correlation with Mflops

WHAT IS THE PROBLEM?

4.0 Overview of Benchmark Results: Mflops vs CPI



SOM/POM vs ISCST3/AERMOD

Four distinct groupings of performance

- High
 - ❑ SOM
- Intermediate
 - ❑ POM with Grid 1
- Low
 - ❑ POM with Grid 2 and 3
 - ❑ ISCST3 and AERMOD

Use PAPI to identify the performance discriminators



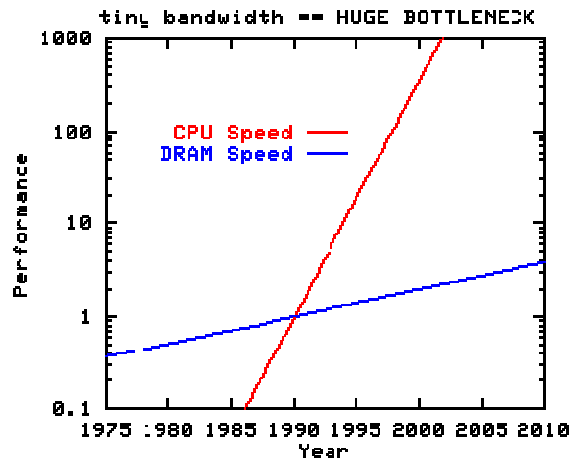
What is the problem?

For:

- SOM – no problem 😊
(just a redundant memory copy)
- POM – memory issues
- ISCST3 and AERMOD –
 - No vector instructions
 - Control transfer instructions
 - memory issues



The memory bandwidth problem (STREAM logo used with permission)



© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>



H I P E R I S M

5.0 Performance Inhibition

More PAPI results with 26 events for

- SOM
- POM
- AERMOD (11 hours for all PAPI events)
- ISCST3 pending (8+ days for all PAPI events!)

Compiler

- pgf90

© Copyright, HIPERiSM Consulting, LLC,

<http://www.hiperism.com>



H I P E R I S M



5.0 Performance Inhibition (cont.)

5.1 Memory access and stalled cycles

- Load/store instructions
- Cycles stalled on any resource

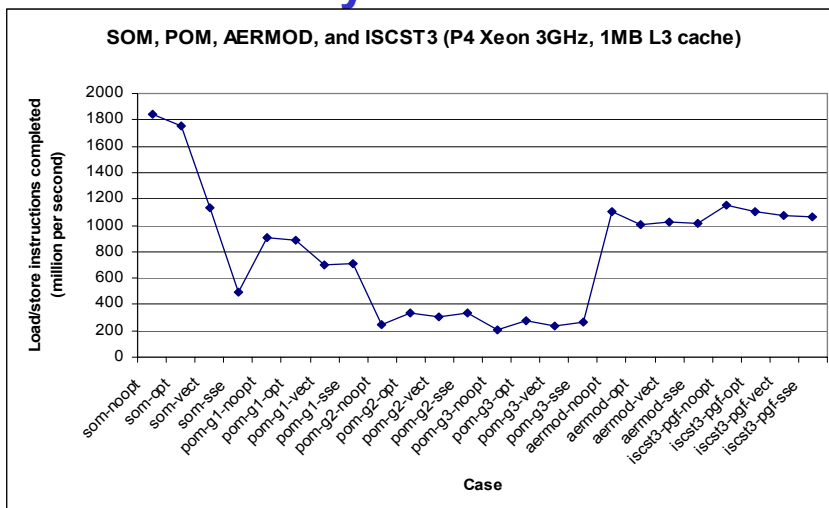
5.2 Cache events

- L1, L2, and L3 cache misses

5.3 Table Lookaside Buffer (TLB) and branching events

- Instruction TLB misses
- Branch instructions

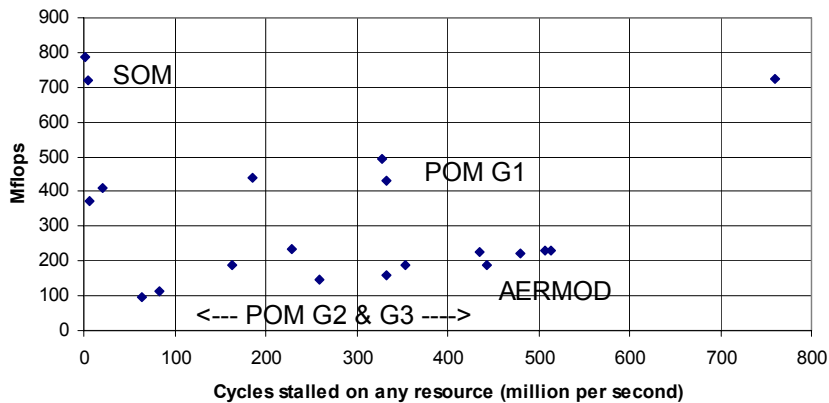
5.1 Performance Inhibition: memory access rates



5.1 Performance Inhibition: Mflops versus stalled cycles



SOM, POM, and AERMOD (P4 Xeon 3GHz, 1MB L3 cache)



5.0 Performance Inhibition



5.1 Memory access and stalled cycles

- High memory access rates can occur for low or high Mflops performance
- Memory access rates for ISCST3 and AERMOD is 2 to 4 times larger than for good vector code
- Highest stalled cycle rates are for ISCST3 and AERMOD

5.0 Performance Inhibition (cont.)



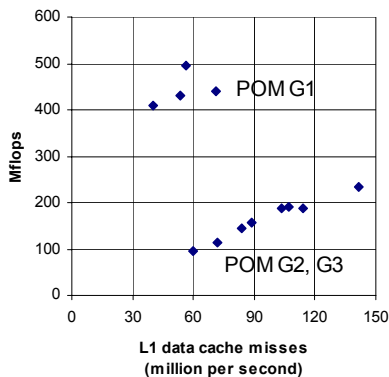
5.2 Cache events

- Data cache misses are not significant for ISCST3 and AERMOD
- Focus on POM performance problem with increasing grid size
- Inspected the data cache rates for L1, L2, and L3 cache

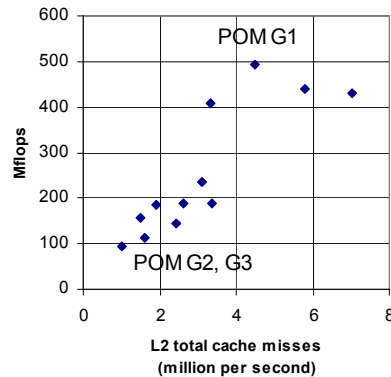
5.2 Performance Inhibition: Mflops & L1 and L2 data cache misses



(a) POM GRID 1, GRID 2, and GRID 3
(P4 Xeon 3GHz, 1MB L3 cache)



(a) POM GRID 1, GRID 2, and GRID 3
(P4 Xeon 3GHz, 1MB L3 cache)



5.0 Performance Inhibition: Lessons learned about POM



POM results

- The L1 data cache is a bottle-neck that differentiates performance as problem size increases
 - ❑ Grid 1 is at lower miss rate threshold
 - ❑ Grids 2 and 3 are at a higher miss rate
- L2 data cache miss rates are an order of magnitude smaller and do not differentiate performance for different Grids.

5.0 Performance Inhibition (cont.)



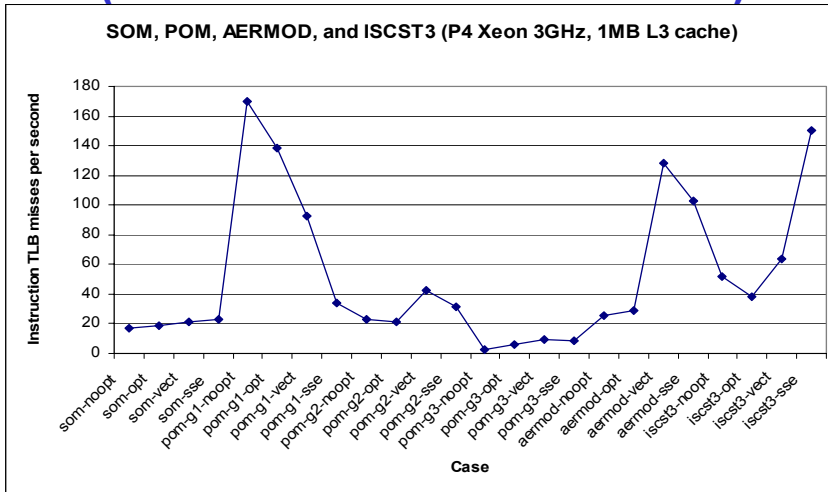
5.3 Table Lookaside Buffer and Branching Events

- For SOM, POM, ISCST3, and AERMOD:
 - ❑ Instruction TLB miss rates
 - ❑ Branch instruction rates

5.0 Performance Inhibition: Instruction TLB misses (AERMOD & ISCST3 x 1/100)



HIPERISM



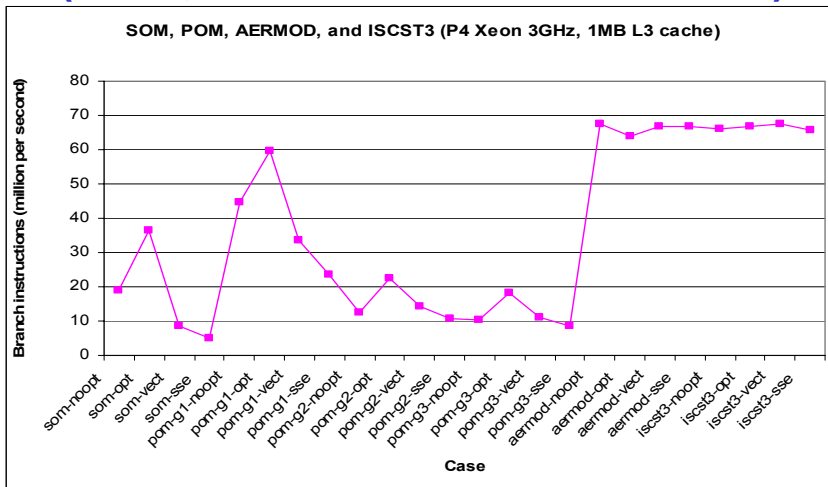
© Copyright, HIPERISM Consulting, LLC,

<http://www.hiperism.com>

5.0 Performance Inhibition: Branch Instructions (SOM, AERMOD & ISCST3 x 1/5)



HIPERISM



© Copyright, HIPERISM Consulting, LLC,

<http://www.hiperism.com>

5.0 Performance Inhibition: Lessons learned for ISCST3 and AERMOD



H I P E R I S M

Compared to SOM or POM

- Instruction TLB miss rates are 85 to 100 times larger
- Branch instruction rates are in the range 10 to 50 time larger

5.0 Performance Inhibition: Summary of lessons learned



H I P E R I S M

Memory usage is key to performance

- L1 cache is a critical bottle-neck that differentiates performance as problem size increases for good vector code because of data cache miss rates
- TLB cache is a critical bottle-neck for serial code because of high instruction TLB miss rates

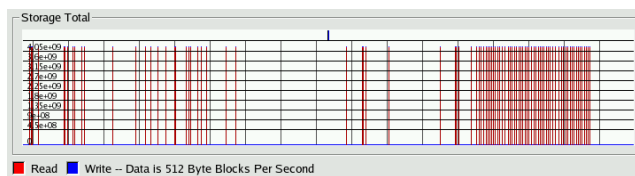
6.0 I/O Performance in Air Quality Models



6.1 I/O Performance for ISCST3

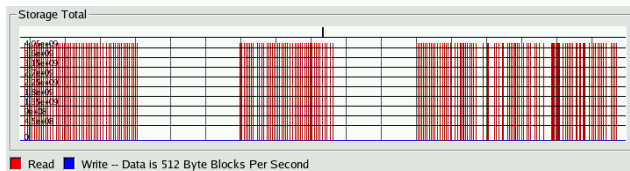
- Inspected with Complete System Performance Monitor (CSPM) snapshots
- Results are for *all* system activity
- Observed:
 - ❑ Lengthy interval at start-up
 - ❑ Cyclical pattern with alternating intervals of sparse and dense read/write activity during computation

CSPM I/O snapshot: ISCST3



55.5 min

67 min



63.5 min

75 min



7.0 Conclusions

- **PAPI** is critical in assessing performance ranges
- **PAPI Events** show that memory usage is the principal differentiator for COTS (not clock speed).
- **SOM & POM** showed that legacy vector codes receive performance boosts from optimizations and SSE on COTS hardware with current compilers
- **AERMOD**: has serious problems on COTS and does not benefit from the performance potential available
 - ❑ **Consequences for AERMOD: re-design of code to reach potential performance limits is advisable.**



8.0 Outlook

- **Hardware**: COTS is delivering good performance on legacy vector code and the outlook is good for such code.
- **Linux**: Operating System is sufficiently reliable.
- **Programming Environment**: rich in compiler and tools technology (e.g. PAPI) for code developers.
- **Consequences for AQM**: the outlook for hardware, Linux, and programming environment requires **careful on-going re-design of code to reach potential performance limits of future COTS technology.**

HiPERiSM's URL

<http://www.hiperism.com>

Technical Reports pages

© Copyright, HiPERiSM Consulting, LLC,

<http://www.hiperism.com>



**H
i
P
E
R
i
S
M**