

ALICE Brown Bag Lunch Presentation

Performance Tuning of An Unstructured Mesh Solver (PETSc-FUN3D)

**Dinesh K. Kaushik**

Department of Computer Science  
Old Dominion University

*in collaboration with*

**Prof. David. E. Keyes**

Department of Computer Science  
Old Dominion University  
& ICASE, NASA Langley Research Center

**Dr. Barry F. Smith**

MCS Division  
Argonne National Laboratory

November 19, 1998

## **Organization of Presentation**

- Implications of memory hierarchy
- Latency tolerance
- Issues for unstructured grid domain decomposition methods
- Background of FUN3D and PETSc
- Illustrations of general performance issues
- Summary of serial and parallel performance
- Conclusions and future plans

## **The Widening Gap between Memory and CPU**

- Memory latency improvement is about 7% per year.
- CPU improvement is 35% per year until 1986, and 55% per year thereafter.

Figure from Hennessy and Patterson, page 374.

## Some Important Dimensionless Numbers

- Typical number of cycles lost on a cache miss:  
10 – 100
- Typical number of cycles lost on a page fault:  
 $10^6$  –  $10^7$
- Typical transfer rate for eight bytes per floating point operation time:  
10 – 100
- Typical message initiation latency per floating point operation time:  
100 – 1,000

# Implications of Memory Hierarchy

- Arrange for temporal locality
  - *Once an operand is cached on a processor, use it as many times as practical before sending it “down” or “out”*
- Arrange for spatial locality
  - *When an operand needs to be moved “up” or “across”, fill up the slots in the same packet with other operands that will be required soon*
- Don't agonize over flops
  - *Flops are cheap compared with memory transfers, so do some extra work per data transfer-laden “cycle” if it reduces the number of cycles*
- Agonize over (low) bandwidth and (high) latency tolerance

## Latency Tolerance — Architect’s Perspective

In “latency” we include both the startup (size-independent) part of the data access (when remote data is ready) **and** the synchronization cost (when remote data is not ready). There are two classes of latency tolerance strategies (from D. Culler, et al., 1998, Chapter 11):

- Amortize the latency:
  - Block data transfers
- Cover the latency:
  - Precommunication
  - Proceeding past an outstanding communication in the same thread
  - Multithreading

The requirements are excess concurrency in the program (beyond the number of processors being used) and excess capacity in the memory and communication architecture.

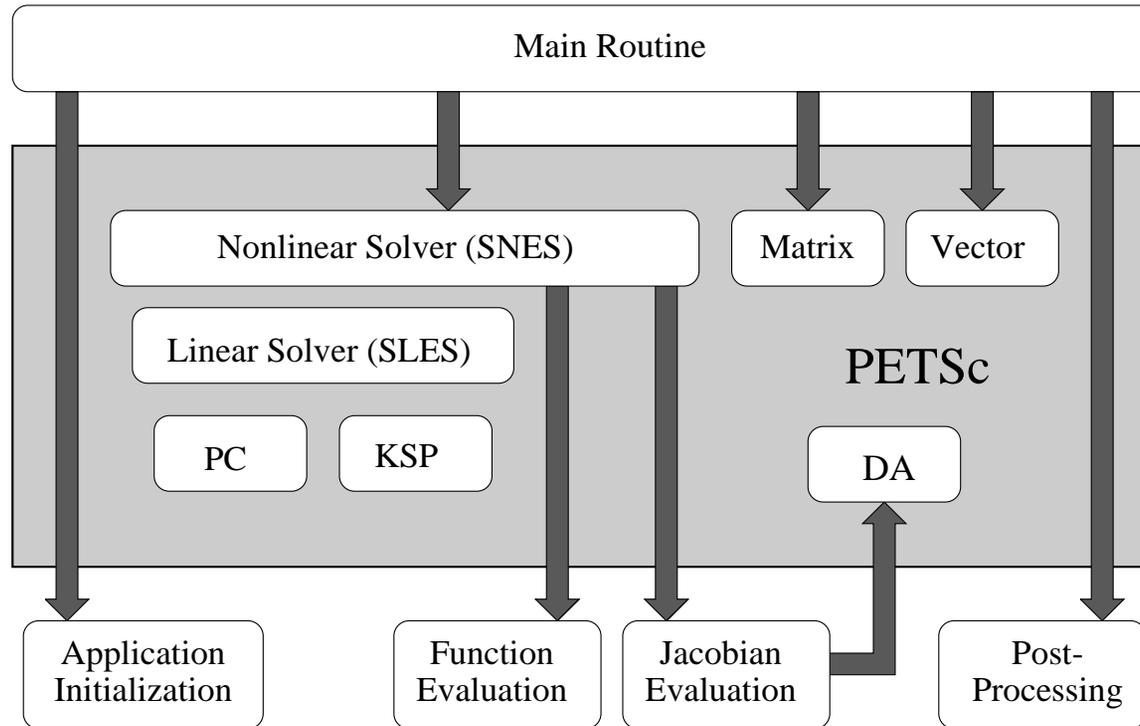
## Description of the Legacy Code - FUN3D

(<http://fmad-www.larc.nasa.gov/~wanderso/Fun/fun.html>)

- FUN3D is a tetrahedral vertex-centered unstructured grid code developed by W. K. Anderson (LaRC) for compressible and incompressible Euler and Navier-Stokes equations
- Parallel experience is with Euler so far, but nothing in the solution algorithms or software changes when viscosity and turbulence are added; only convergence rate will vary with conditioning, as determined by Reynolds number (and mesh)
- FUN3D uses 1st- or 2nd-order Roe for convection and Galerkin for diffusion, and false timestepping with backwards Euler for nonlinear continuation towards steady state
- Solver is Newton-Krylov-Schwarz; timestep is advanced towards infinity by the switched evolution/relaxation (SER) heuristic of Van Leer & Mulder

# Integration with the Library Solver - PETSc

(<http://www.anl.gov/petsc>)



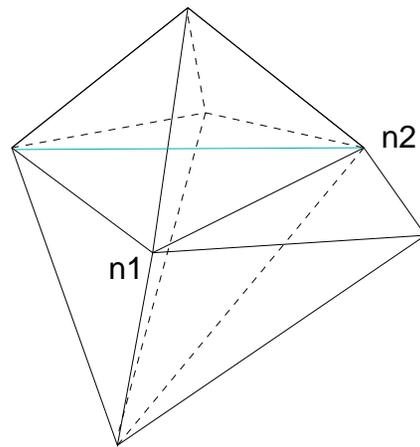
## Pseudo-Transient Newton-Krylov-Schwarz Algorithm

(after Cai, Gropp, Keyes, and Tidriri (1994))

```
for (l = 0; l < n_time; l++) {
  SELECT TIME-STEP
  for (k = 0; k < n_Newton; k++) {
    compute nonlinear residual and Jacobian
    for (j = 0; j < n_Krylov; j++) {
      forall (i = 0; i < n_Precon ; i++) {
        solve subdomain problems concurrently
      } // End of loop over subdomains
      perform Jacobian-vector product
      ENFORCE KRYLOV BASIS CONDITIONS
      update optimal coefficients
      CHECK LINEAR CONVERGENCE
    } // End of linear solver
    perform DAXPY update
    CHECK NONLINEAR CONVERGENCE
  } // End of nonlinear loop
} // End of time-step loop
```

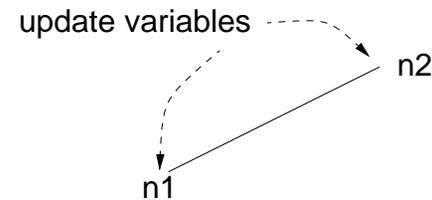
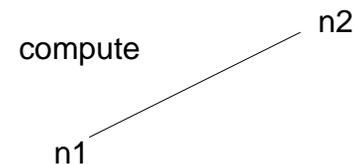
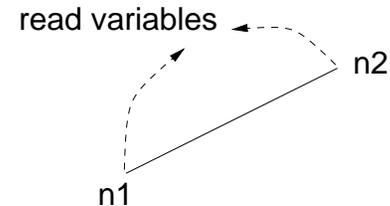
# Edge-based Loops for Flux Computation

- Used inside Newton loop in every residual evaluation
- Used inside Krylov loop in every matrix-vector product

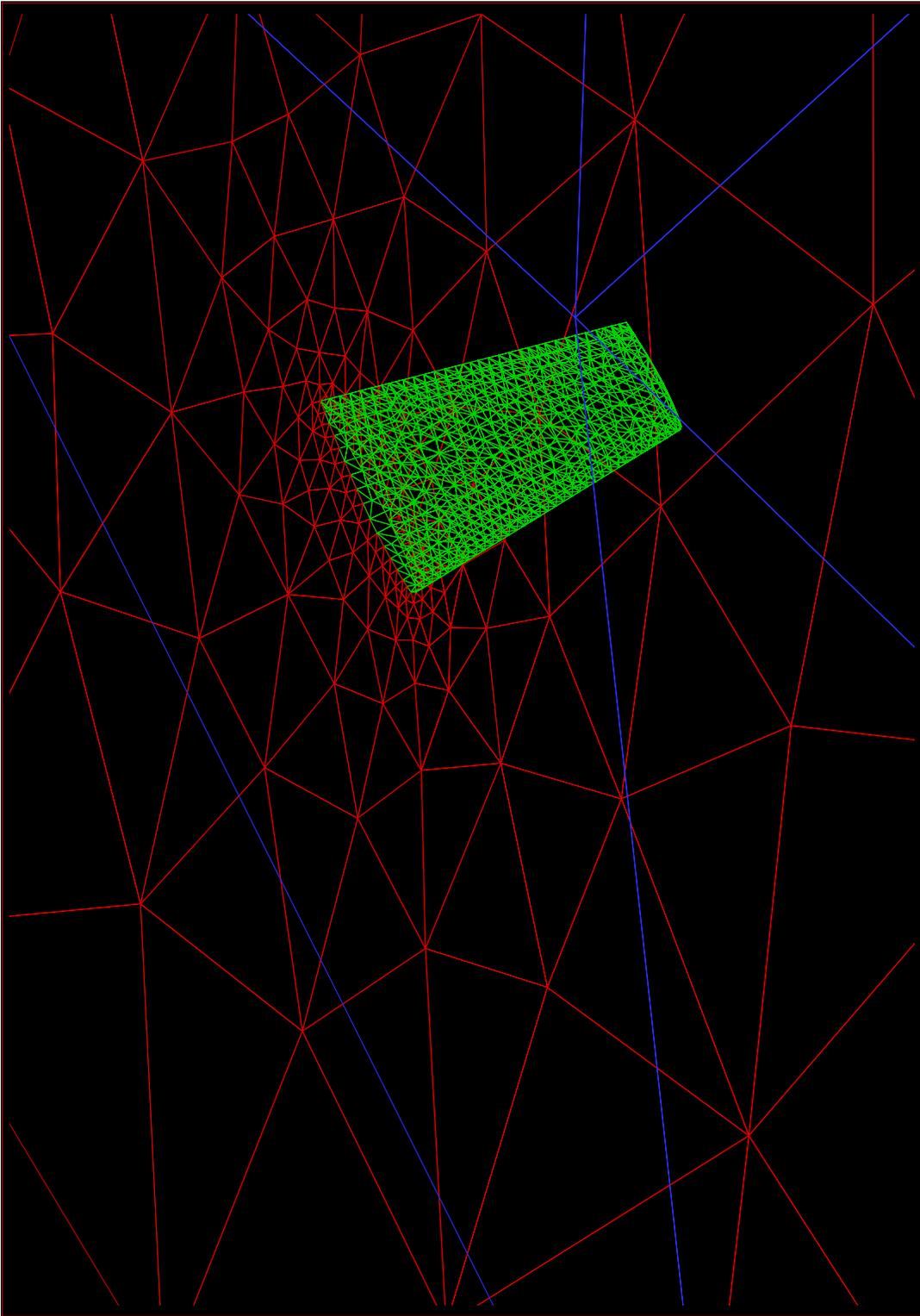


Variables at each node:  
density,  
momentum (  $x,y,z$  ),  
energy,  
pressure

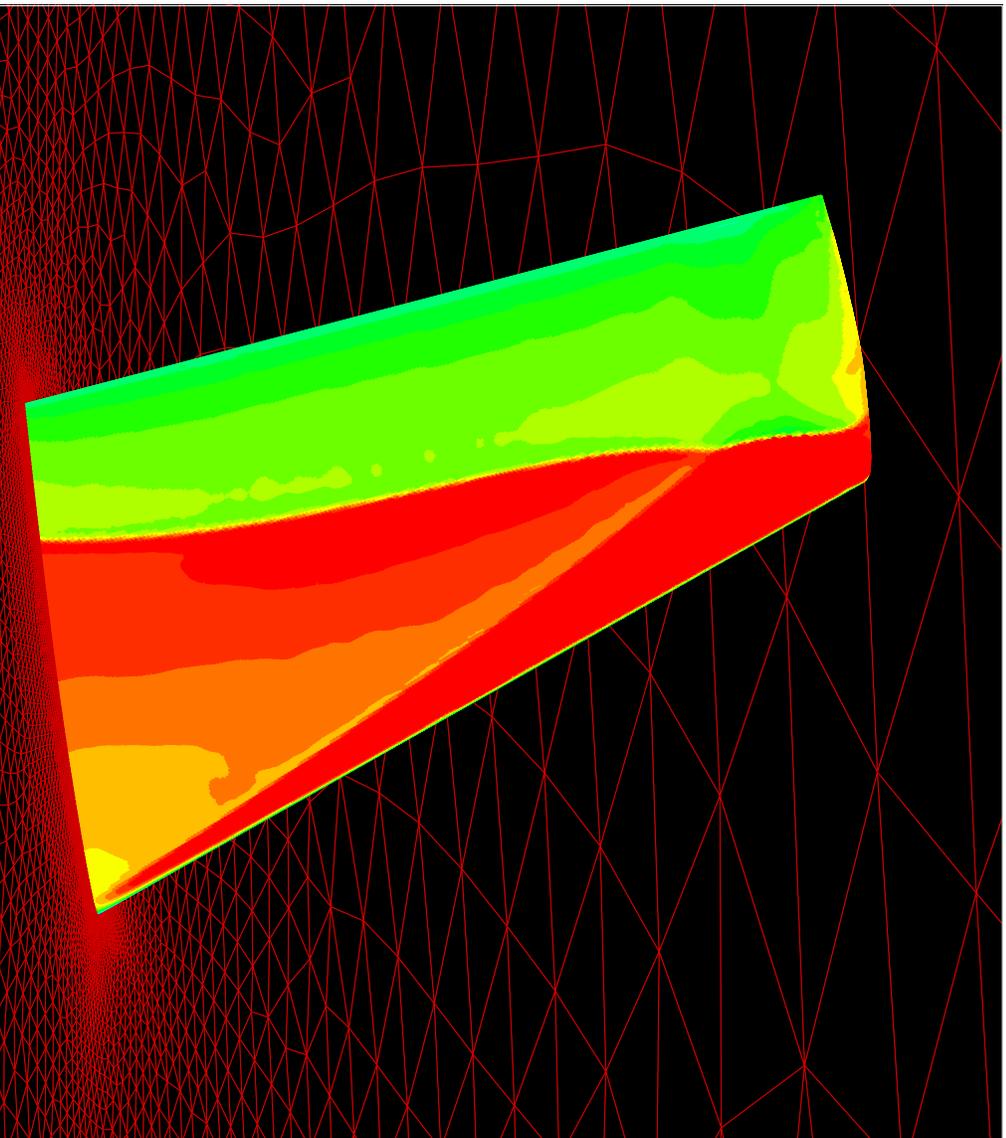
Variables at edge:  
identity of nodes,  
orientation(  $x,y,z$  )



# Surface Visualization of Test Domain (M6 wing)



# Illustrative Solution of “Lambda Shock” Case



## Performance Tuning – Three Fronts

- Algorithmic Tuning
  - Choose “optimal” compromise of large number of nonorthogonal parameters
- Compiler Transformations
  - Free the compiler to do what it does best
- Data Layouts
  - Stay in harmony with the memory hierarchy

## Algorithmic Tuning for NKS Solver

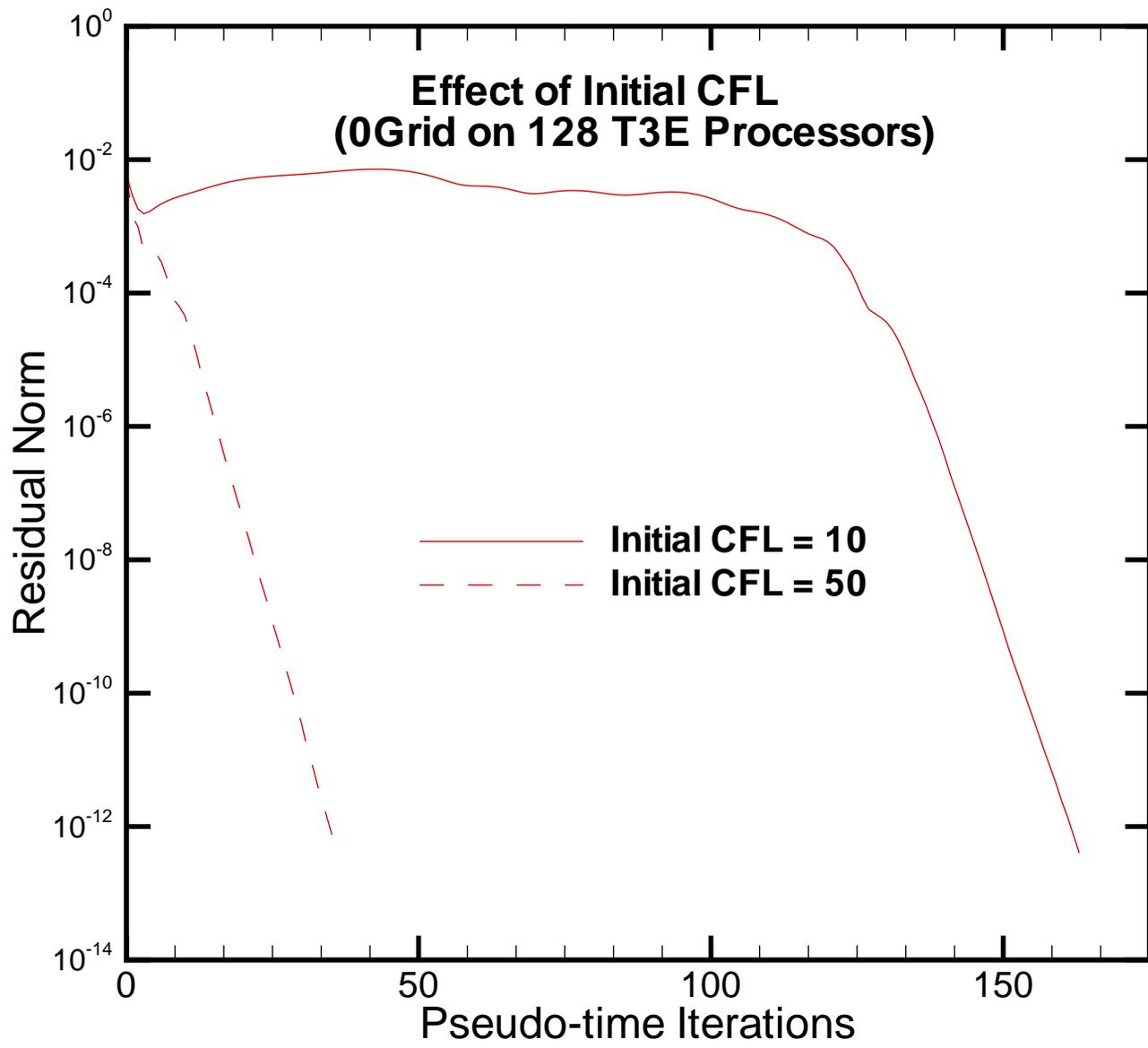
- Continuation parameters: discretization order, initial timestep, timestep evolution
- Newton parameters: convergence tolerance, globalization strategy, Jacobian refresh frequency
- Krylov parameters: convergence tolerance, subspace dimension, restart number, orthogonalization mechanism
- Schwarz parameters: subdomain number, subdomain solver, subdomain overlap, coarse grid usage
- Subproblem parameters: fill level, number of sweeps

## Algorithmic Tuning – Continuation Parameters

- SER heuristic

$$N_{CFL}^\ell = N_{CFL}^0 \left( \frac{\|f(u^0)\|}{\|f(u^{\ell-1})\|} \right)^p$$

- Parameters of Interest
  - Initial **CFL** number
  - Exponent in the Power Law
    - $> 1$  for first-order discretization (1.5)
    - $< 1$  at outset of second-order discretization (0.75)
    - $= 1$  normally
  - Switch over Ratio between FO and SO



## Algorithmic Tuning – Krylov Parameters

- These parameters were chosen after lot of experiments
- Convergence Tolerance
  - a value of 0.01 works well for most of the cases run
- Subspace Dimension
  - depends on the problem dimension
  - typical values range from 10 (for smallest problem) to 60 for the largest problem
- Restart Number
  - dependent on the available memory
  - typical values are 15 to 30

## Optimal Granularity of Decomposition

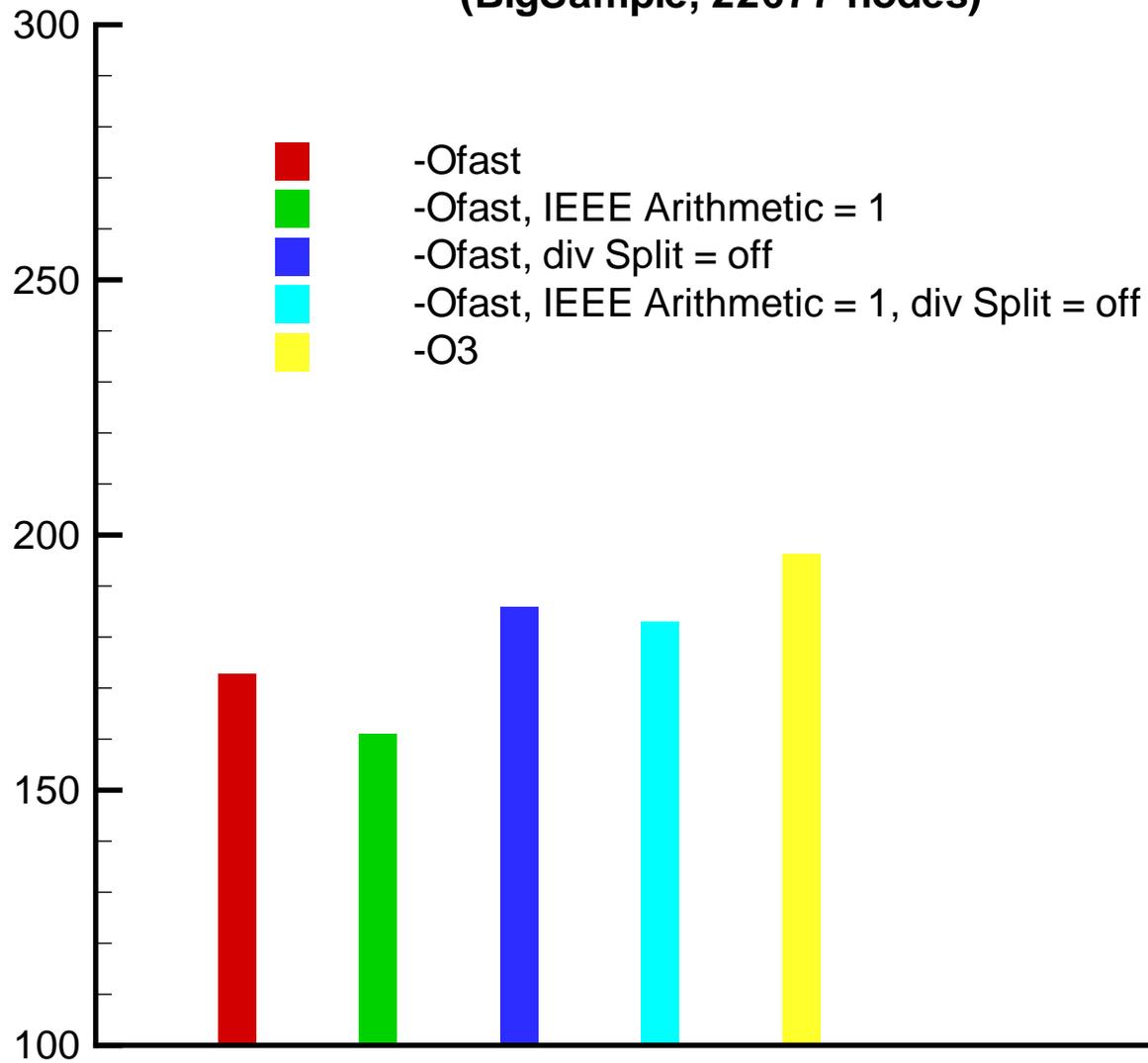
For **cache-based microprocessors**, granularity of domain decomposition iterative methods is determined by three forces:

- **Convergence Rate**  
usually deteriorates with increased granularity
- **Communication Volume**  
increases with increased granularity
- **Size of Local Working Set**  
fits better into successively smaller cache levels with increased granularity

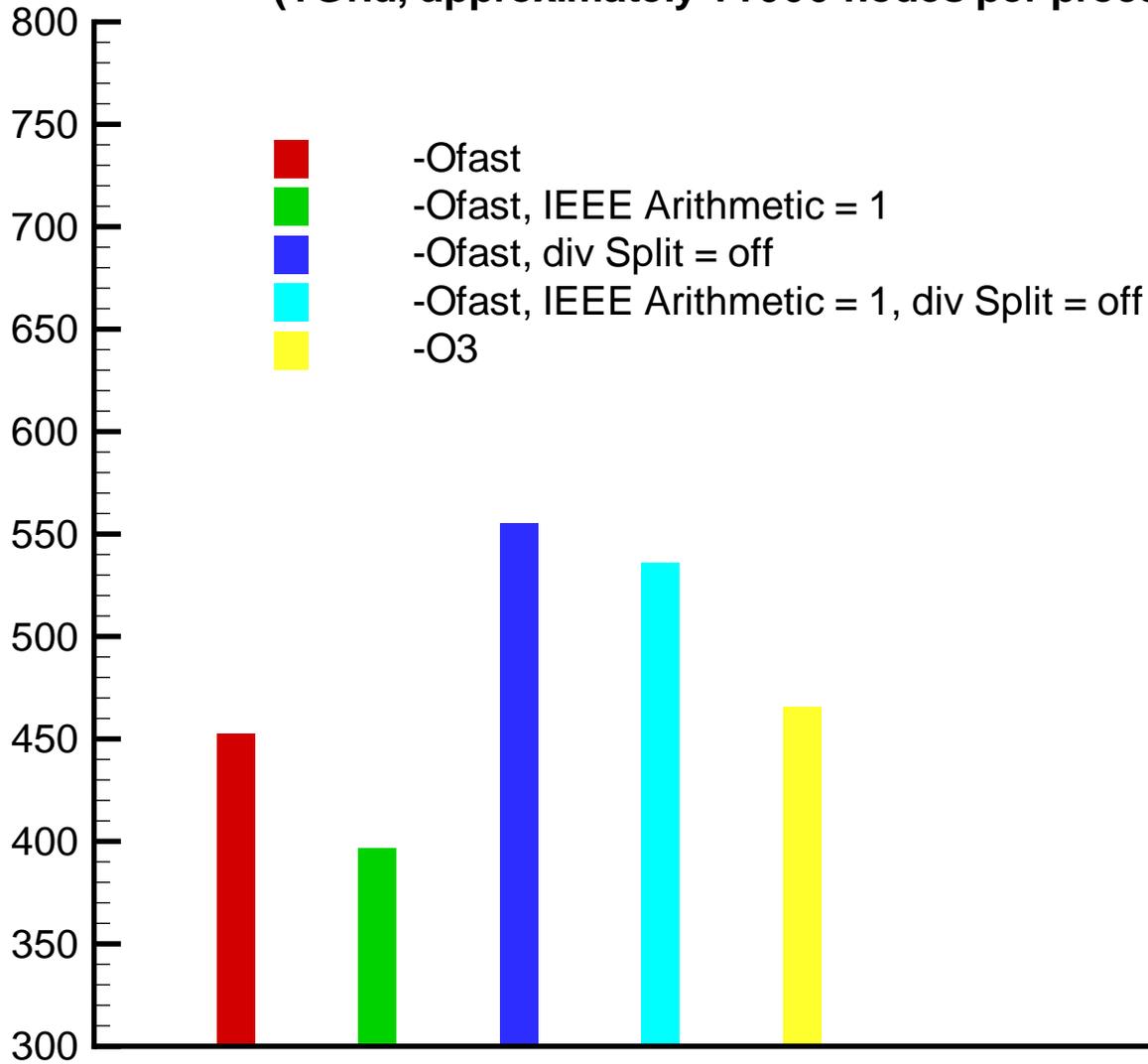
## Compiler Transformations

- Choose the highest level of optimization that give the right result
- Effect of different compiler flags (Origin 2000)
  - **-Ofast** : does aggressive optimization (including O3 optimizations)
  - **-OPT:IEEE\_arithmetic=1** : inhibits optimizations that produce less accurate results than required by ANSI/IEEE 754-1985
  - **-OPT:div\_split=off** : disables the calculation of  $x/y$  as  $x^*(1.0/y)$
  - **-O3** : level 3 optimization

### Execution Time for Sequential Case (BigSample, 22677 nodes)



**Execution Time on 32 Processors  
(1Grid, approximately 11000 nodes per processor)**



## Data Layouts

- Choose data layouts that enhance locality at every level of Memory hierarchy
- Storage/use patterns should follow memory hierarchy
  - **Blocks for Registers**  
block storage format for multicomponent systems – saves CPU cycles
  - **Interlaced Data Structures for Cache**  
choose  
$$u_1, v_1, w_1, p_1, u_2, v_2, w_2, p_2, \dots$$
in place of  
$$u_1, u_2, \dots, v_1, v_2, \dots, w_1, w_2, \dots, p_1, p_2, \dots$$
  - **Subdomains for Distributed Memory**  
“chunky” domain decomposition for optimal surface-to-volume (communication-to-computation) ratio
  - This hierarchy is concerned with different issues than the **algorithmic efficiency** issues associated with hierarchies of grids

## Data Layouts (contd.) – Reorderings

- Edge Reordering
  - sort the nodes at either ends of the edges
  - this effectively transforms an edge based loop into a node based loop
  - enhances temporal locality
- Node Reordering
  - \* Reverse Cuthill McKee (**RCM**)
  - \* Fast Sloan

## Locality Enhancing Strategies in Serial

- Flow over M6 wing with fixed-size grid of 22,677 vertices (90,708 DOFs incompressible; 113,385 DOFs compressible)
- Turn on each optimization one by one to isolate the effect of each
- Five architectures considered: Cray T3E, IBM SP, Origin 2000, Intel Pentium, and Sun Ultra
- Impact of these techniques vary on different architectures — improvement ranges from **2.5 on Pentium** to **7.5 on SP**

## Sequential Performance on IBM SP

IBM P2SC (“thin”), 120MHz, cache: 128KB data and 32 KB instr

Enhancements				Results			
Field	Structural	Edge	Incompressible	Compressible			
Interlacing	Blocking	Reordering	Time/Step	Ratio	Time/Step	Ratio	
			165.7s	—	237.6s	—	
×			62.1s	2.67	85.8s	2.77	
×	×		50.0s	3.31	65.7s	3.62	
		×	43.3s	3.82	67.5s	3.52	
×		×	33.5s	4.95	50.8s	4.68	
×	×	×	22.1s	7.51	32.2s	7.37	

## Sequential Performance on Intel Pentium

Intel Pentium II (NT), 400MHz, cache: 16KB data / 16KB instr / 512KB L2

Enhancements				Results			
Field	Structural	Edge	Incompressible	Compressible			
Interlacing	Blocking	Reordering	Time/Step	Ratio	Time/Step	Ratio	Ratio
			70.3s	—	108.5s	—	
×			44.1s	1.59	70.1s	1.55	
×	×		37.4s	1.88	57.3s	1.89	
		×	43.8s	1.61	72.4s	1.50	
×		×	34.0s	2.07	54.5s	1.99	
×	×	×	27.6s	2.55	43.2s	2.51	

## Sequential Performance on SGI Origin

MIPS R10000, 250MHz, cache: 32KB data / 32KB instr / 4MB L2

Enhancements			Results			
Field	Structural	Edge	Incompressible	Compressible		
Interlacing	Blocking	Reordering	Time/Step	Ratio	Time/Step	Ratio
			83.6s	—	140.0s	—
×			36.1s	2.31	57.5s	2.44
×	×		29.0s	2.88	43.1s	3.25
		×	29.2s	2.86	59.1s	2.37
×		×	23.4s	3.57	35.7s	3.92
×	×	×	16.9s	4.96	24.5s	5.71

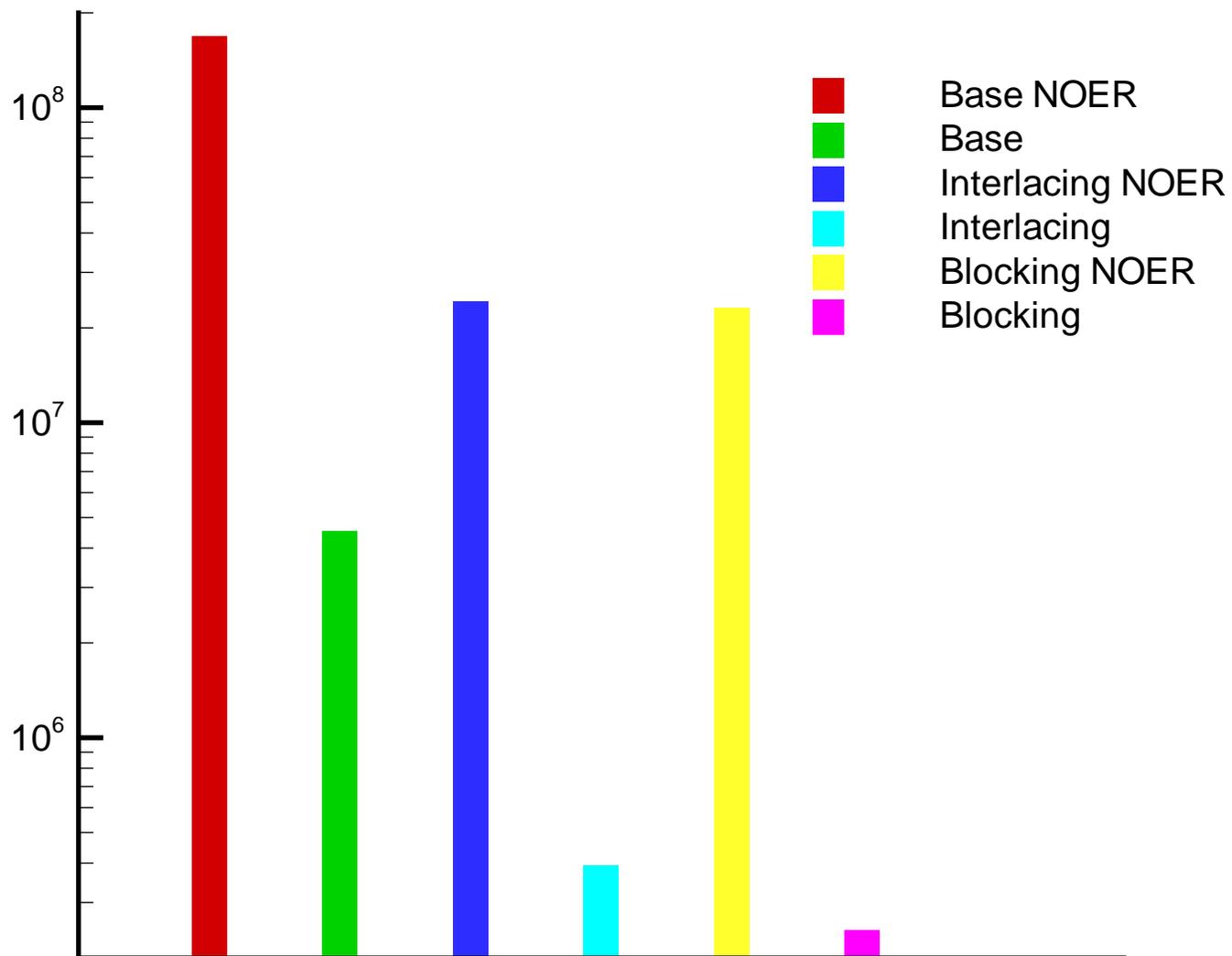
## **Performance Monitoring – Hardware Counters**

- Hardware counters available on almost all modern architectures
- Each vendor provides own interface performance monitoring
- At least two independent efforts to provide a portable user interface
  - PCL — The Performance Counter Library from Central Institute for Applied Mathematics, Research Centre Juelich, Germany
  - RABBIT — A Performance Counters Library for Intel Processors and Linux from Ames Laboratory
- PerfAPI — Performance Data Standard and API project is directed towards a possible standard

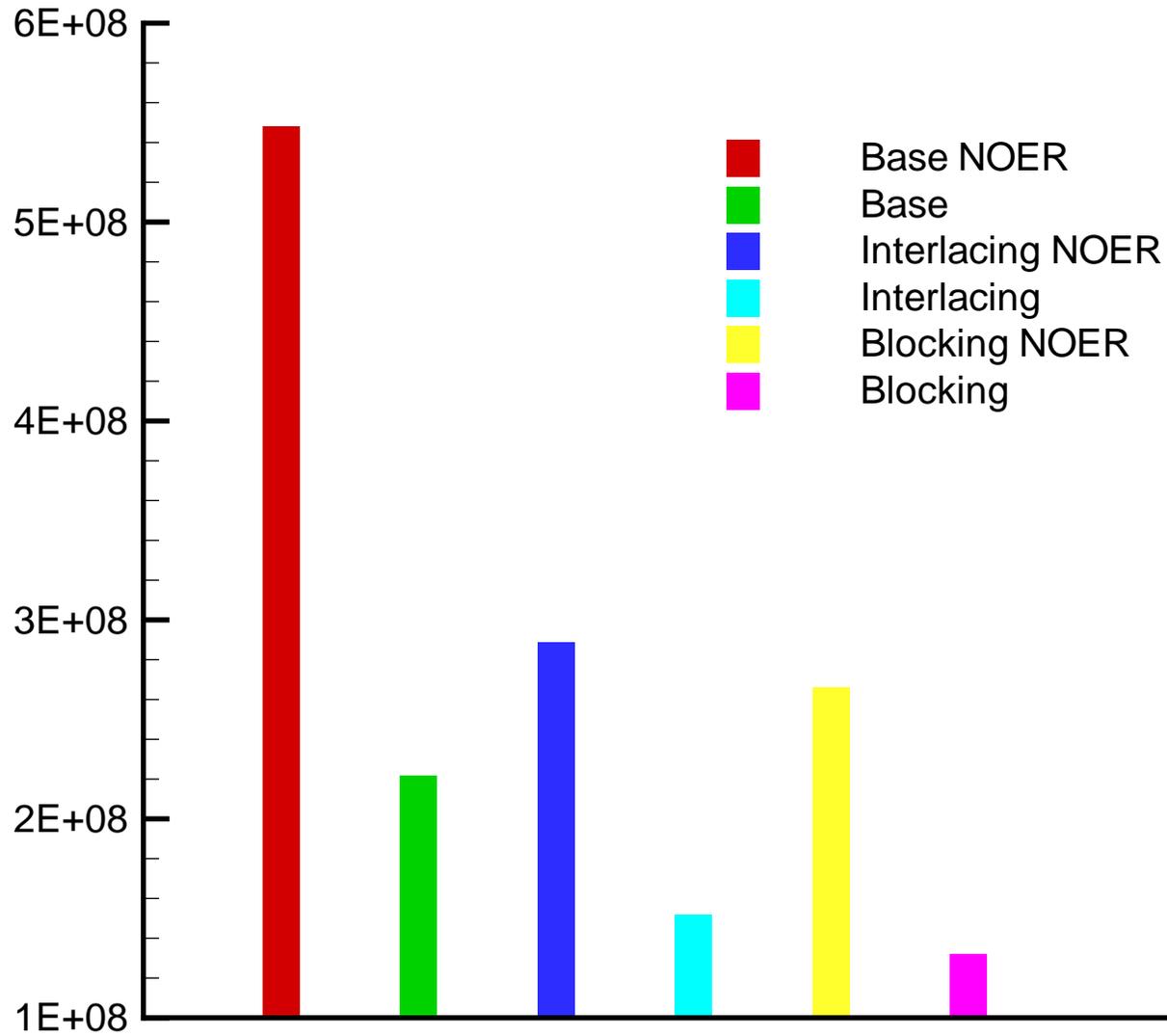
## Hardware Profiling on SGI Origin

- TLB Misses
- Primary Cache Misses
- Secondary Cache Misses
- Graduated Loads and Stores Per Floating Point Instruction

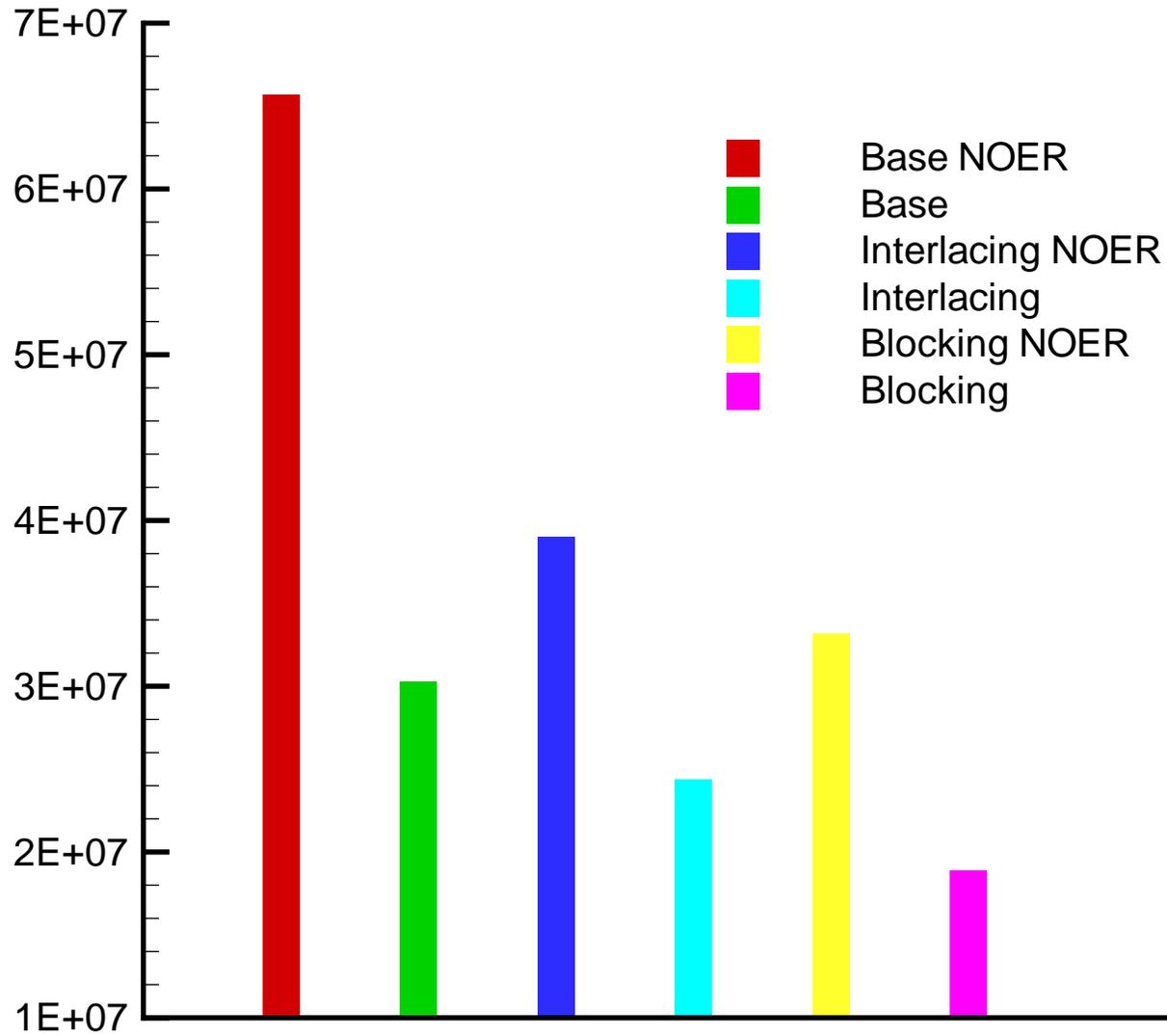
# TLB Misses



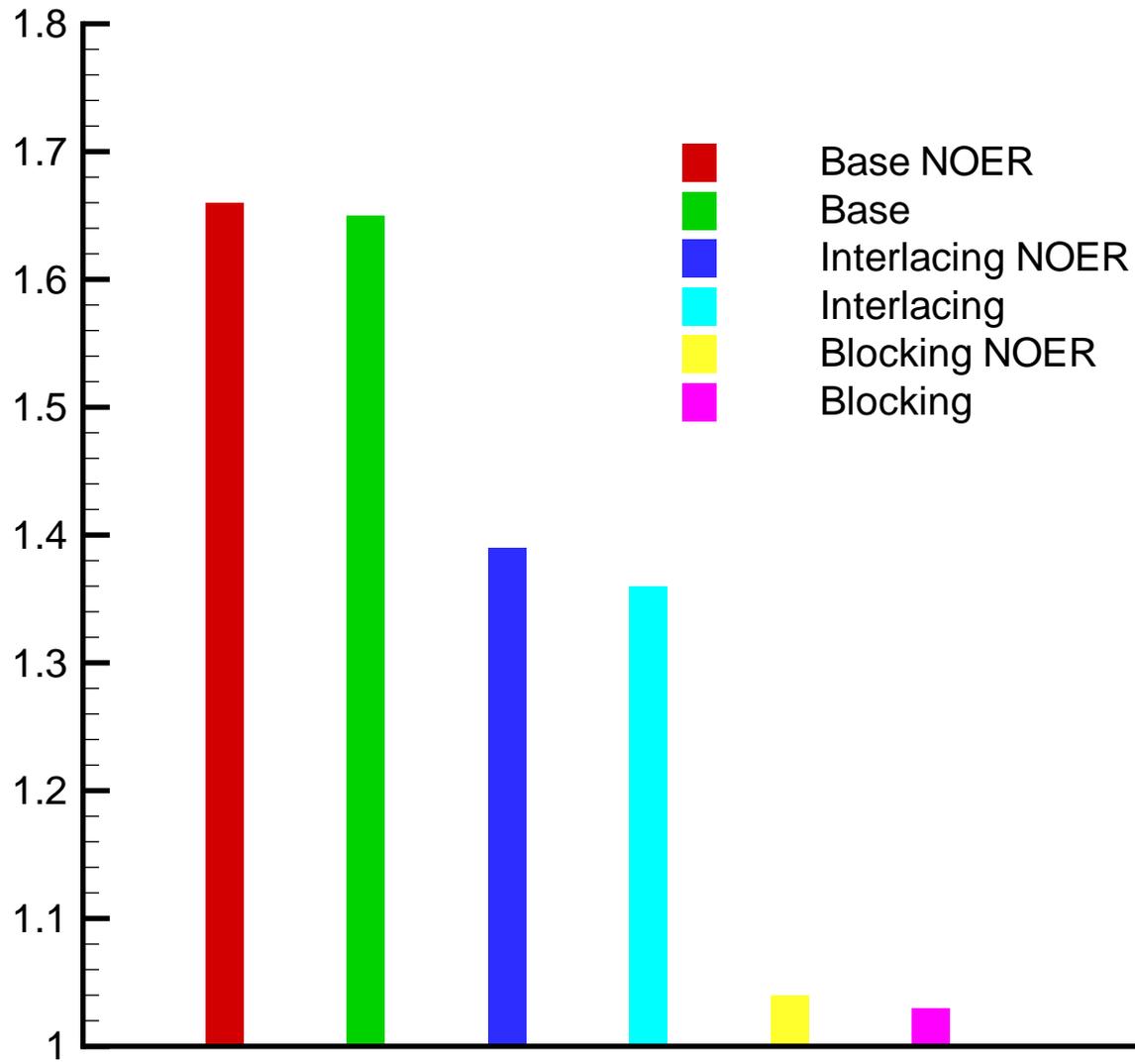
### Primary Cache Misses



## Secondary Cache Misses



### Graduated Loads and Stores / Floating Point Instruction



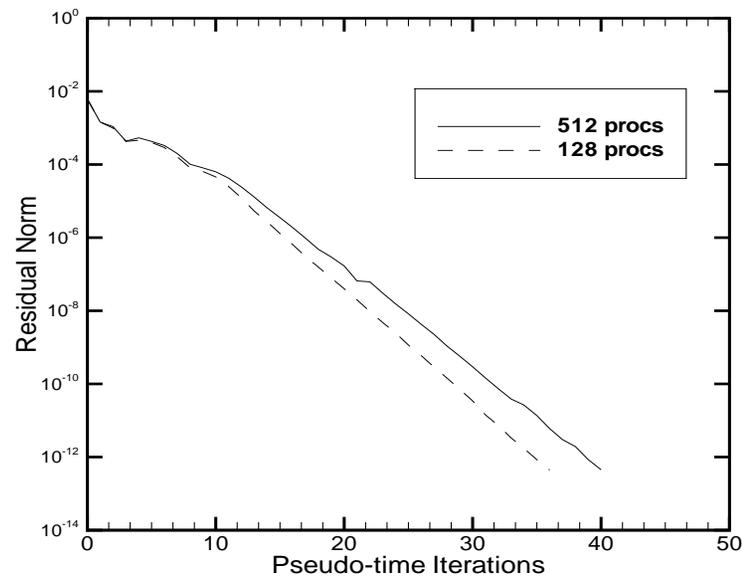
# Parallel Performance of Incompressible Solver on Cray T3E

FUN3D-PETSc ONERA M6 Wing Test Case, 2nd-order Roe Scheme, 1-layer Halo

Tetrahedral grid of 2,761,774 vertices (11,047,096 unknowns)

on T3E-900 (450 MHz) at NERSC

procs	its	time	speedup	Efficiency			Communication		sustained Mflop/s per proc.	sustained total Gflop/s
				$\eta_{alg}$	$\eta_{impl}$	$\eta_{overall}$	inner prod.	halo exch.		
128	37	2,811.20s	1.00	1.00	1.00	1.00	6%	3%	71.5	9.1
256	38	1,495.24s	1.88	0.97	0.96	0.94	8%	3%	69.7	17.8
512	41	833.75s	3.37	0.90	0.93	0.84	9%	4%	68.3	35.0



# Parallel Performance of Compressible Solver on Cray T3E, IBM SP, and SGI Origin

Transonic flow over M6 wing; fixed-size grid of 357,900 vertices

No. Procs.	Cray T3E			IBM SP			SGI Origin		
	Steps	Time	Eff.	Steps	Time	Eff.	Steps	Time	Eff.
16	55	2406s	—	55	1920s	—	55	1616s	—
32	57	1331s	.90	57	1100s	.87	56	862s	.94
48	57	912s	.88	57	771s	.83	56	618s	.87
64	57	700s	.86	56	587s	.82	57	493s	.82
80	57	577s	.83	59	548s	.70	57	420s	.77

# Comparison of Euler Flow Regimes over M6 Wing on SGI Origin 2000

Fixed Size Scaling: 357,900 vertices

(1,431,600 DOFs incompressible, 1,789,500 DOFs compressible)

No.	Steps	Time per Step	Per-Step Speedup	Impl. Eff.	FcnEval MHop/s	JacEval MHop/s
Incompressible (Mach 0) (4 × 4 blocks)						
16	19	41.6s	—	—	2,630	359
32	19	20.3s	2.05	1.02	5,366	736
48	21	14.1s	2.95	0.98	7,938	1,080
64	21	11.2s	3.71	0.93	10,545	1,398
80	21	10.1s	4.13	0.83	11,661	1,592
Subsonic (Mach 0.30) (5 × 5 blocks)						
16	17	55.4s	—	—	2,002	2,698
32	19	29.8s	1.86	0.93	3,921	5,214
48	19	20.5s	2.71	0.90	5,879	7,770
64	20	14.3s	3.88	0.97	8,180	10,743
80	20	12.7s	4.36	0.87	9,452	12,485

# Comparison of Euler Flow Regimes over M6 Wing on SGI Origin 2000

Fixed Size Scaling: 357,900 vertices  
(1,789,500 DOFs compressible)

No. Procs.	Steps	Time per Step	Per-Step Speedup	Impl. Eff.	FcnEval MHop/s	JacEval MHop/s
Transonic (Mach 0.84) (5 × 5 blocks)						
16	55	29.4s	—	—	2,009	2,736
32	56	15.4s	1.91	0.95	4,145	5,437
48	56	11.0s	2.66	0.89	5,942	7,961
64	57	8.7s	3.39	0.85	8,103	10,531
80	57	7.4s	3.99	0.80	9,856	12,774
Supersonic (Mach 1.20) (5 × 5 blocks)						
16	80	19.2s	—	—	2,025	2,679
32	81	10.6s	1.81	0.90	3,906	5,275
48	81	7.1s	2.72	0.91	6,140	7,961
64	82	5.8s	3.31	0.83	7,957	10,398
80	80	4.6s	4.20	0.84	9,940	12,889

## Conclusions

- The near-scalable algorithms for general purpose PDE simulations that we use today can in theory be adapted to an architectural climate of diverging rates of computation and memory access, requiring increased concurrency with concentrated locality.
- But, in practice, we must simultaneously improve algorithmic tolerance to the memory latency of the architecture.

## Future Directions

- Architecture-oriented
  - correlate hardware counter measurements with data structure organization and refine cache strategies in a quantitative way
- Programming model-oriented
  - examine appropriate role of multi-threading within a subdomain in a hybrid DSM/SMP programming style
- Application-oriented
  - examine the relative advantages of structured and unstructured grids from a performance perspective (partitioning and ordering flexibility versus representation efficiency)

## References

- *On the Interaction of Architecture and Algorithm in the Domain-Based Parallelization of an Unstructured Grid Incompressible Flow Code* (with Keyes and Smith), 1998, in “Proc. of the 10th Intl. Conf. on Domain Decomposition Methods”, J. Mandel et al., eds., AMS, pp. 311–319.
  - cache-aware focus
- *Newton-Krylov-Schwarz Methods for Aerodynamics Problems: Compressible and Incompressible Flows on Unstructured Grids* (with Keyes and Smith), 1998, submitted to “Proc. of the 11th Intl. Conf. on Domain Decomposition Methods”, C.-H. Lai et al., eds.
  - multi-platform comparisons focus
- *Prospects for CFD on Petaflops Systems* (with Keyes and Smith), 1998, to appear in “CFD Review”, M. Hafez et al., eds., Wiley.
  - parallel scalability focus
- all these can be downloaded from
  - <http://www.cs.odu.edu/~kaushik/papers.html>
  - <http://www.cs.odu.edu/~keyes/keyes.html>