COMPUTATIONAL ISSUES

TOM ROSMOND

NRL, Monterey (SAIC)

JCSDA Summer Colloquium on Data Assimilation

Stevenson, Washington

July 7 – 17, 2009

COMPUTATIONAL ISSUES

Outline

- *Why is this a topic of this course?*
  - *Like it or not, we all spend a LOT of time struggling with these issues*
  - *Better understanding, and facility at dealing with the issues, will pay off in more scientific productivity*
- *Historical Overview*
  - *Analysis ⇨ Data Assimilation*
  - *Forecast model ⇨ Data Assimilation*
  - *Conventional Data ⇨ Satellite Data*
- *Computational Environments*
  - *Mainframe computers*
    Vector supercomputers
    Massively parallel computers
    Cluster supercomputers & Workstations
- *Programming*
  - *Languages*
  - *Relationship to computational environments*

COMPUTATIONAL ISSUES

- *Today 'Data Assimilation' has replaced 'Objective Analysis'*
  - *Update cycle*
  - *Data quality control*
  - *Initialization*

- *NWP computational costs (Before late 1970's)*
  - *Objective analysis relatively inexpensive*
  - *Forecast model(s) dominated*

- *NWP computational costs (1980's & 1990's)*
  - *Data volumes increased dramatically*
  - *Model & Data assimilation costs roughly equivalent*

- *NWP computational costs (Today)*
  - *Data volumes continue to increase*
  - *Data assimilation costs often exceed model costs*
    - *4Dvar with multiple outer loops*
    - *Ensemble based DA*
    - *Non-linear observation operators (radiance assimilation)*

COMPUTATIONAL ISSUES

Current computational challenges

- *Massive increases in data volume, e.g. NPOESS*
- *Ensemble based covariances (ETKF)*
- *Marriage of 4Dvar and ensemble methods?*
- *Non-linear observation operators*
  - *Radiance assimilation*
  - *Radar data for mesoscale assimilation*
- *Heterogeneous nature of DA*
  - *Significant serial processing*
  - *Parallelism at script level?*
- *Data assimilation for climate monitoring*

COMPUTATIONAL ISSUES

Other challenges

- *Distinction between DA 'people' and 'modelers' blurring*
  - *TLM & Adjoint models in 4Dvar*
  - *Ensemble models for covariance calculation*
- *Scientific computing no longer dominant*
  - *Vendor support waning*
  - *Often multiple 'points of contact' for problems*

COMPUTATIONAL ISSUES

Computing environments: 1960’s – 1970’s

- *IBM, CDC, DEC, etc*
- *Mainframe computers, proprietary hardware*
- *Proprietary operating systems*
- *No standard binary formats*
- *Little attention paid to standards*
- *Code portability almost non-existent*
- *Users became vendor ‘shops’*

COMPUTATIONAL ISSUES

Computing environments: 1980's – mid 1990's

- 'Golden Age' of scientific computing
- Scientific computing was king
- Vector supercomputers, proprietary hardware
- Price/performance : supercomputer cheapest
- Cray, CDC (ETA), Fujitsu, NEC, IBM
- Excellent vendor support (single point of contact)
- Cray became defacto standard (UNICOS, CF77)
- First appearance of capable desktop WS's and PC's

COMPUTATIONAL ISSUES

Computing environments: mid 1990's - today

- *Appearance of massively parallel systems*
- *Commodity based hardware*
- *Open source software environments (Linux, GNU)*
- *Scientific computing becoming niche market*
- *Vendor support waning*
- *Computing environments a collection of 3rd party components*
- *Greater emphasis on standards: data and code*
- *Portability of DA systems a priority*
- *Sharing of development efforts essential*

COMPUTATIONAL ISSUES

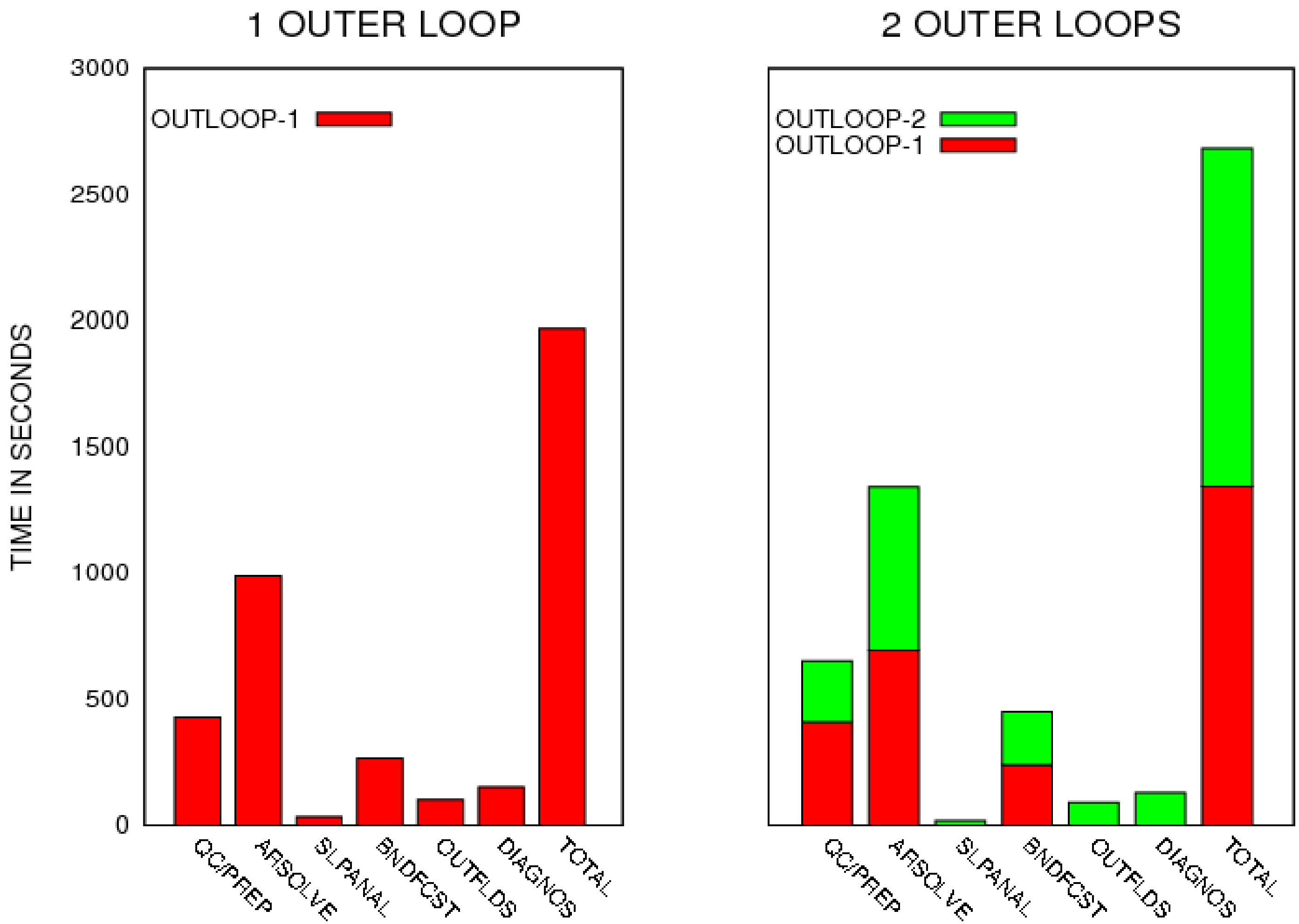
Challenges

- DA is by nature a heterogeneous computational problem
  - Observation data ingest and organization
  - Observation data quality control/selection
  - Background forecast: NWP model
  - Cost function minimization (3Dvar/4Dvar)
  - Ensemble prediction (ensemble DA)

- Parallelism also heterogeneous
  - Source code
  - Script level
  - An important contribution to complexity of DA systems
  - SMS (developed by ECMWF, licensed to other sites)

COMPUTATIONAL ISSUES

NAVDAS-AR Components



COMPUTATIONAL ISSUES

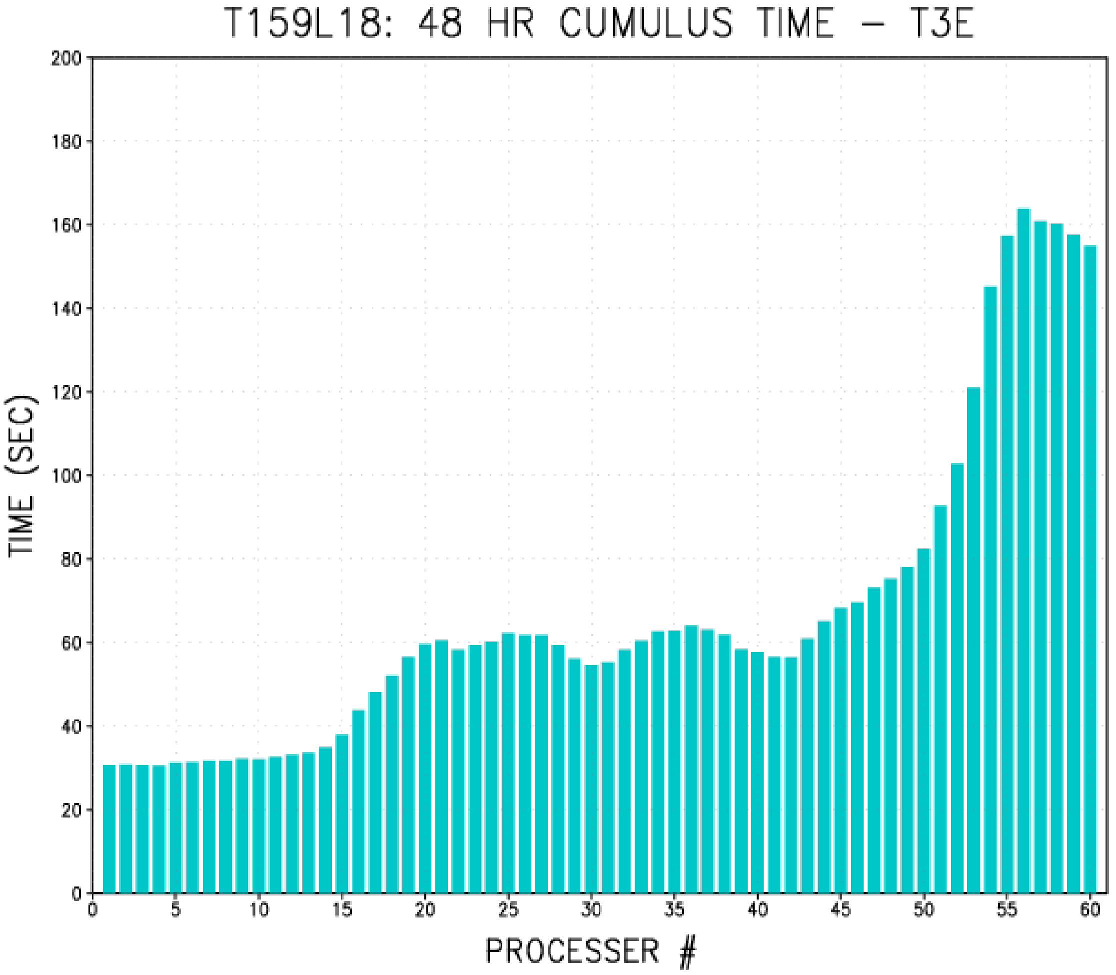
MUST always think parallel

- *Programming models*
  - *OpenMP*
  - *Message Passing (MPI)*
  - *Hybrids*
  - *Co-array Fortran*
  - *High-Performance Fortran (HPF)*

- *Parallel Performance (How well does it scale)*
  - *Amdahl's Law*
  - *Communication fabric (network)*
  - *Latency dominates over bandwidth in limit*
  - *But: For our problems, load imbalance limiting factor*

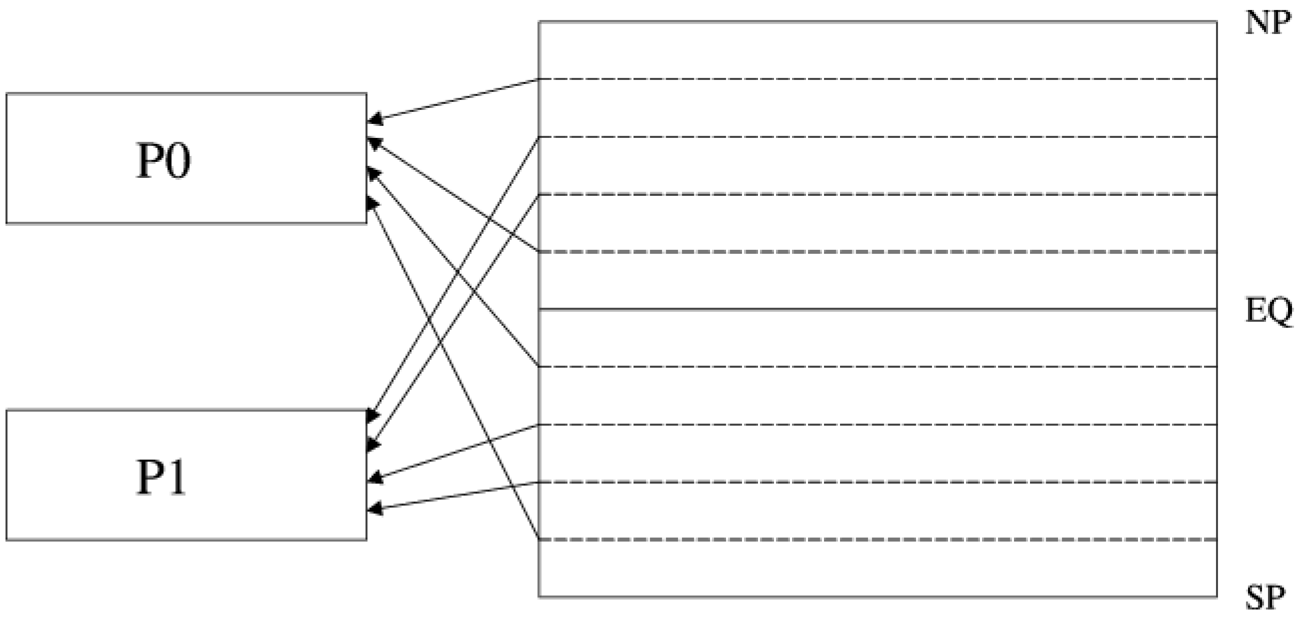
COMPUTATIONAL ISSUES

Load Balancing: no shuffle



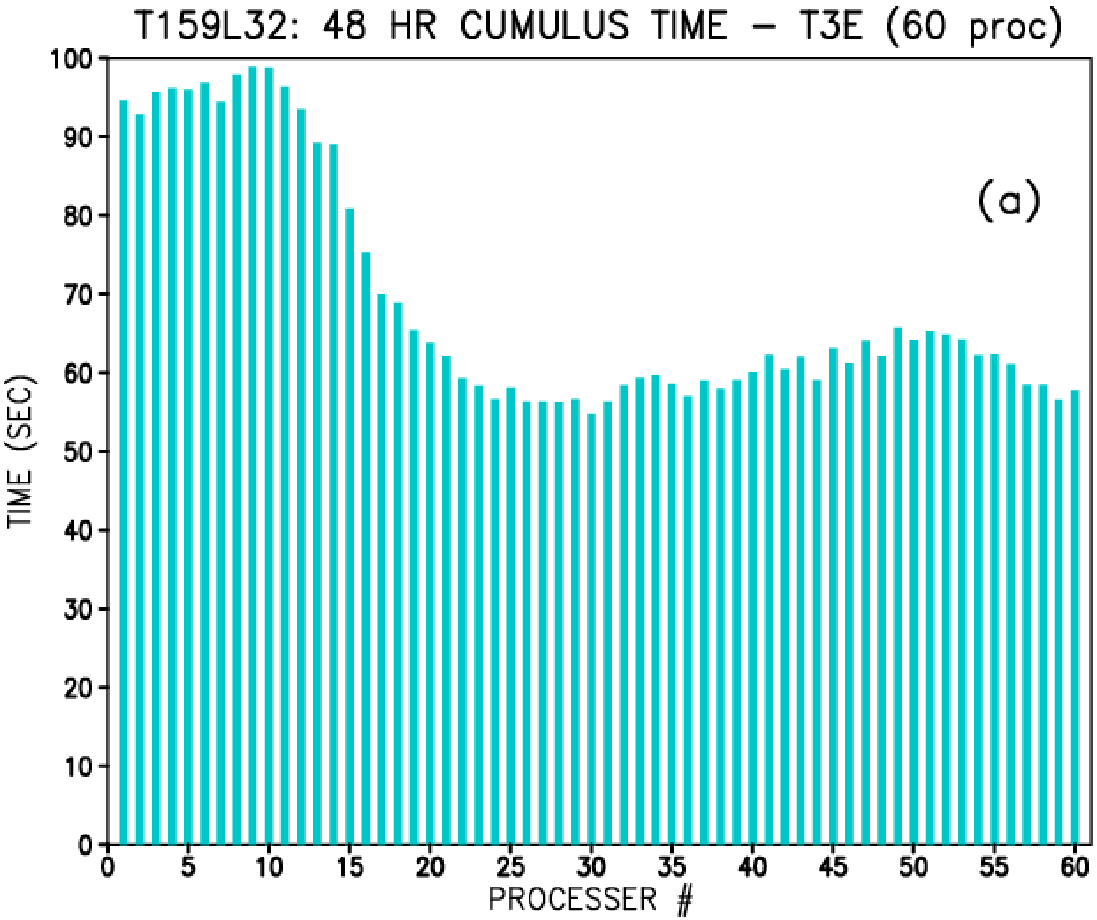
COMPUTATIONAL ISSUES

Load balance + spectral transform "shuffle"



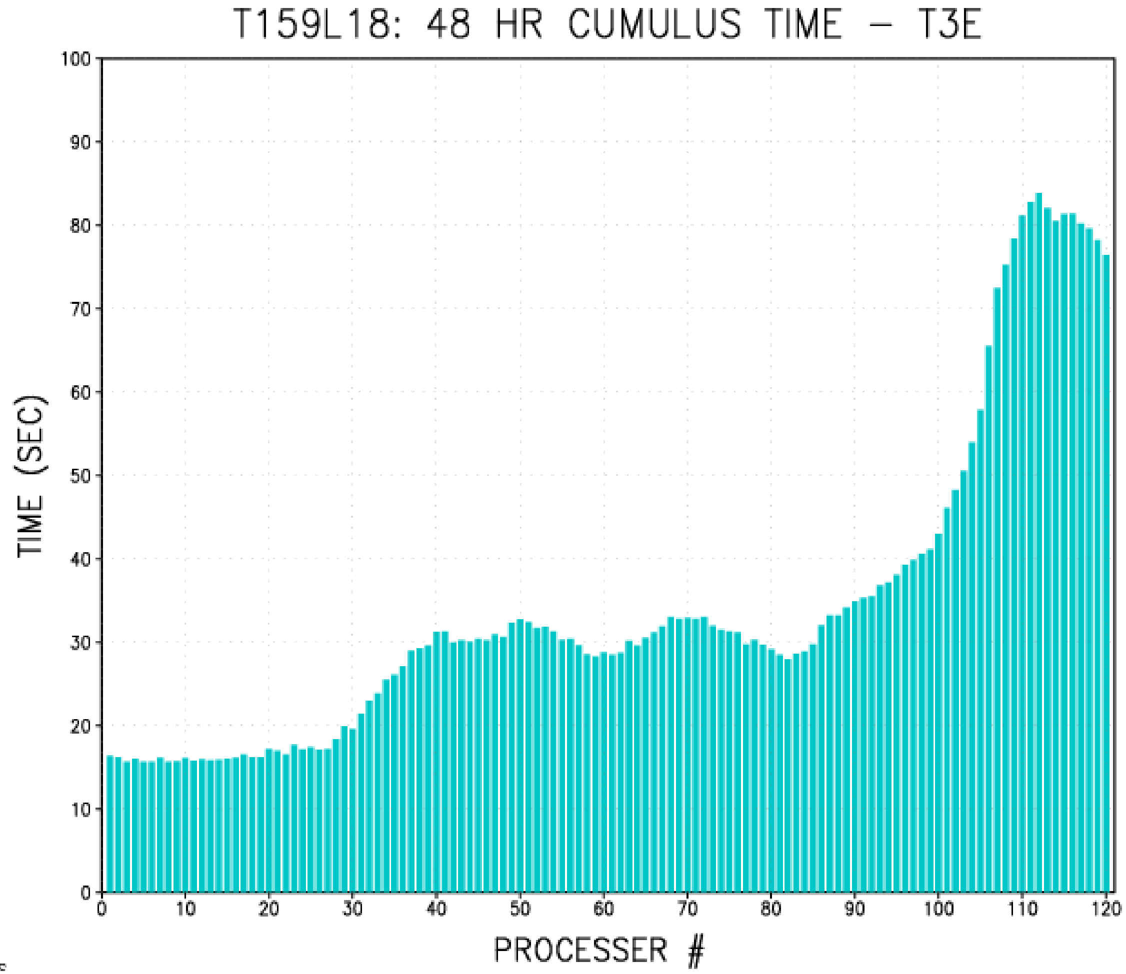
COMPUTATIONAL ISSUES

Load Balancing: with shuffle



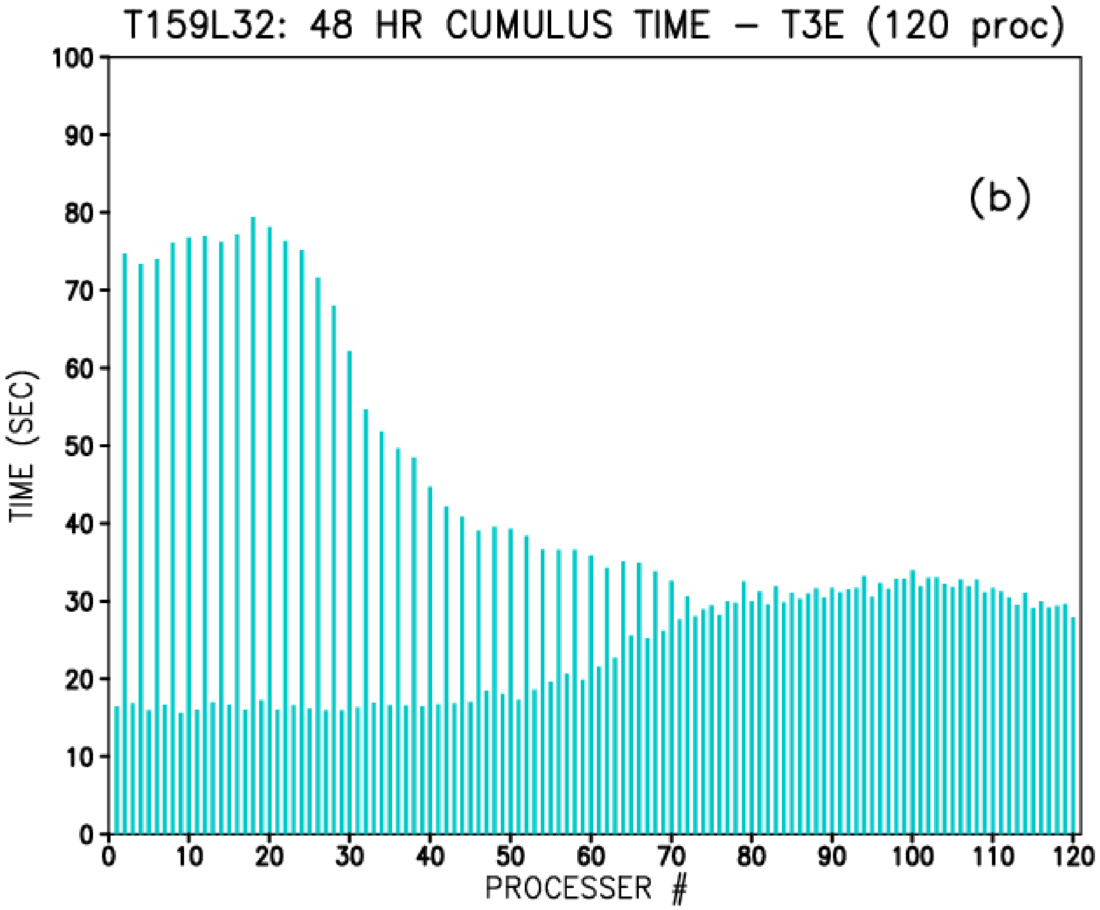
COMPUTATIONAL ISSUES

Load Balancing: no shuffle



COMPUTATIONAL ISSUES

Load Balancing: with shuffle



COMPUTATIONAL ISSUES

OpenMP

- Origin was ‘multi-tasking’ on Cray parallel-vector systems
- Relatively easy to implement in existing codes
- Supported in Fortran and C/C++
- ‘Best’ solution for modest parallelism
- Scalability for large processor problems limited
- Only relevant for shared memory systems (not clusters)
- Support must be built into compiler
- ‘On-node’ part of hybrid programming model

COMPUTATIONAL ISSUES

Message Passing (MPI)

- *Currently dominates large parallel applications*
- *Supported in Fortran and C/C++*
- *External library, not compiler dependent*
- *Many open source implementations (OpenMPI, MPICH)*
- *Works in both shared and distributed memory environments*
- *2-sided message passing (send-recv)*
- *1-sided message passing (put-get) (shmem)*
- *MPI programming is 'hard'*

COMPUTATIONAL ISSUES

Hybrid programming models

- MPI + OpenMP
- OpenMP on ‘nodes’
- MPI between ‘nodes’
- Attractive idea, but is it worth it?
- To date, little evidence it is, but experience limited
- Should help load imbalance problems
- Limiting case of full MPI or full OpenMP in single code.

COMPUTATIONAL ISSUES

Co-array Fortran

- *Effort to make parallel programming easier*
- *Attractive concept, but support limited (Cray)*
- *Adds processor indices to Fortran arrays (co-arrays)*

  e.g.   :   x(i,j)[l,k]
- *Scheduled to be part of next Fortran standard*

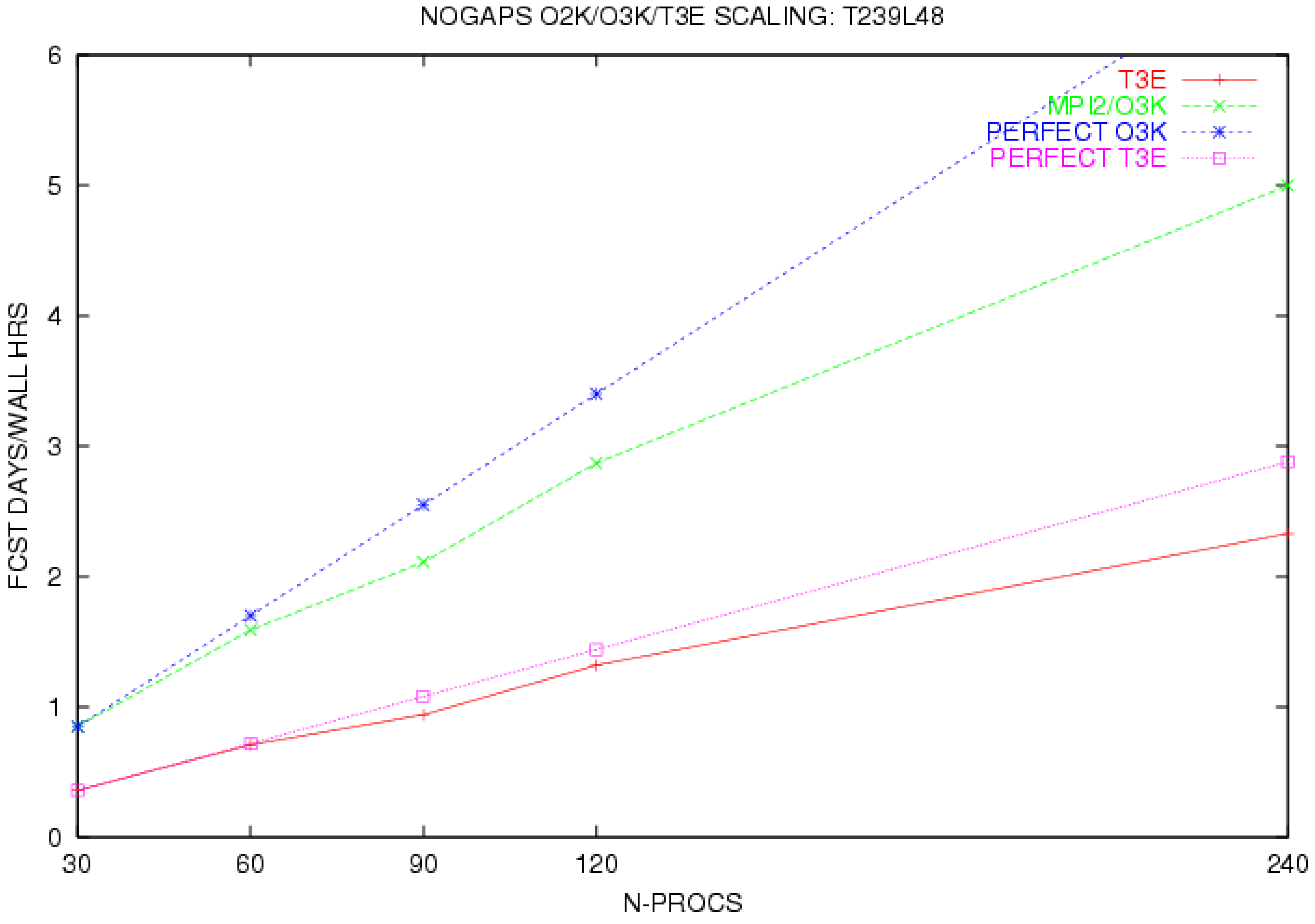
COMPUTATIONAL ISSUES

High-performance Fortran (HPF)

- Another effort to make parallel programming easier
- Has been around several years
- Supported by a few vendors (PGI)
- Performance is hardly high (to say the least)
- A footnote in history?

COMPUTATIONAL ISSUES

Scalability 1990's



COMPUTATIONAL ISSUES

More challenges

- *Many 'supercomputers' (clusters) use same hardware and software as desktops*

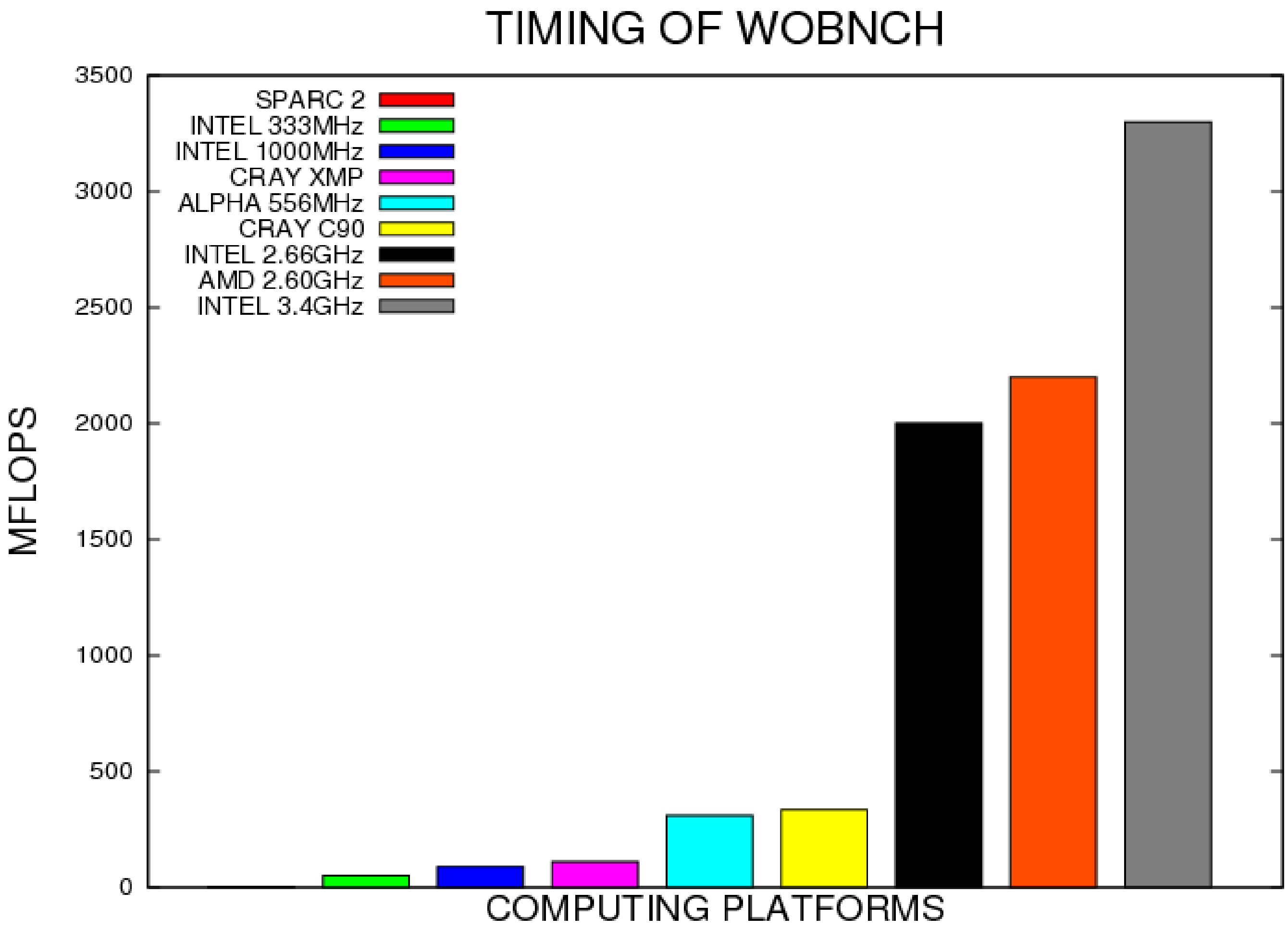
  - *processors*
  - *motherboards*
  - *mass storage*
  - *Linux*

- *Price/performance ratio has seemingly improved dramatically because of this*

  - *A Cray C90 equivalent is about $1000*
  - *1 Tbyte HD (> $100) is ~ equivalent to the disk storage of all operational NWP centers 25 years ago*

COMPUTATIONAL ISSUES

Evolution of processor power: ~ 20years



COMPUTATIONAL ISSUES

More challenges

- *Current trend of multi-core processors*
  - *4, 8 cores now common*
  - *multiple processors on single MB*

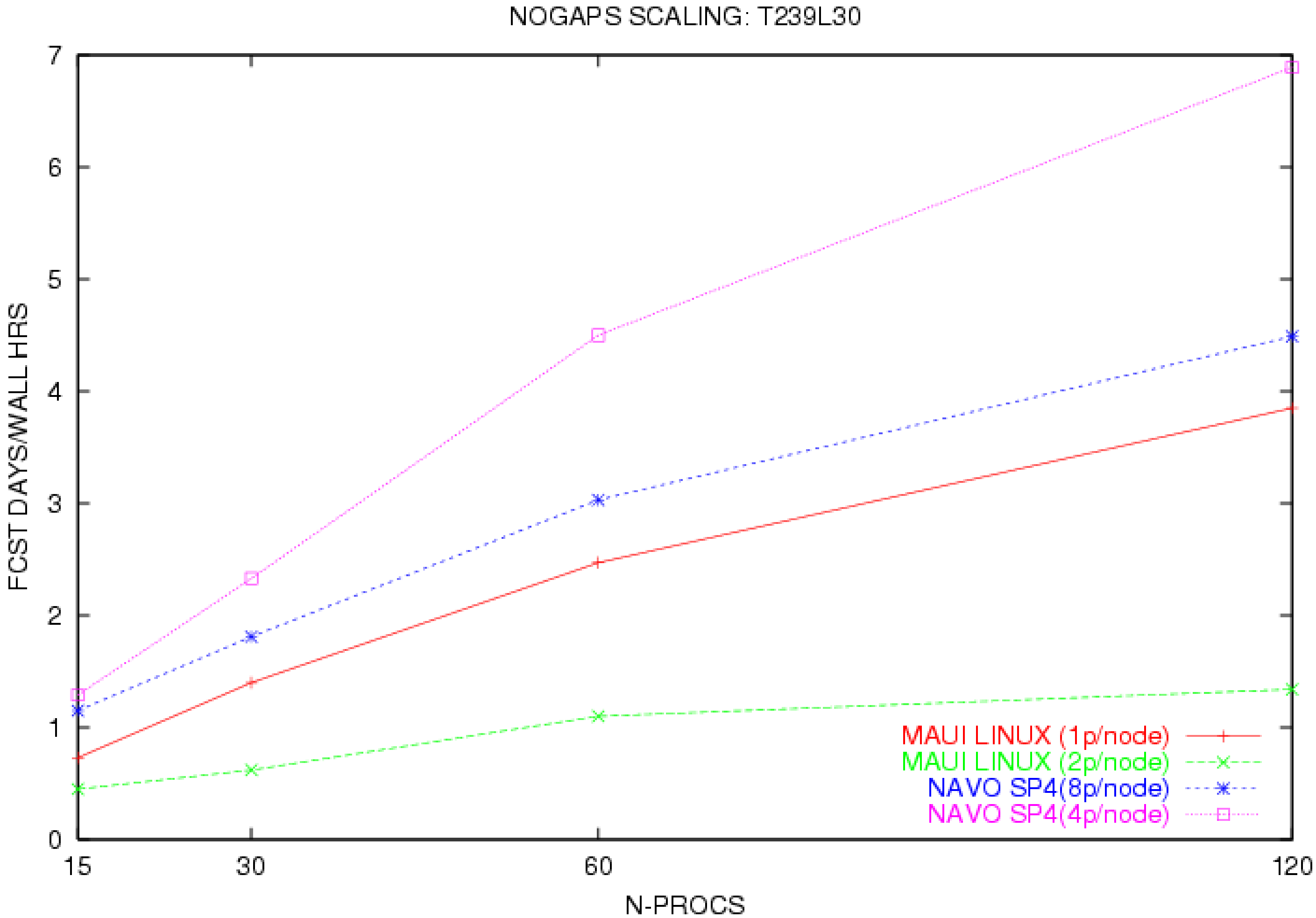
- *Problem: Cores increasing, but system bandwidth (bus speed) isn't keeping pace*
  - *Terrible imbalance between processor speed and system bandwidth/latency*

- *Everything we really want to do depends on this*
  - *Memory access*
  - *IO*
  - *Inter-processor communication (MPI)*

- *Sandia report: disappointing performance and scalability of real applications on multi-core systems.*

COMPUTATIONAL ISSUES

Impact of processor/node utilization



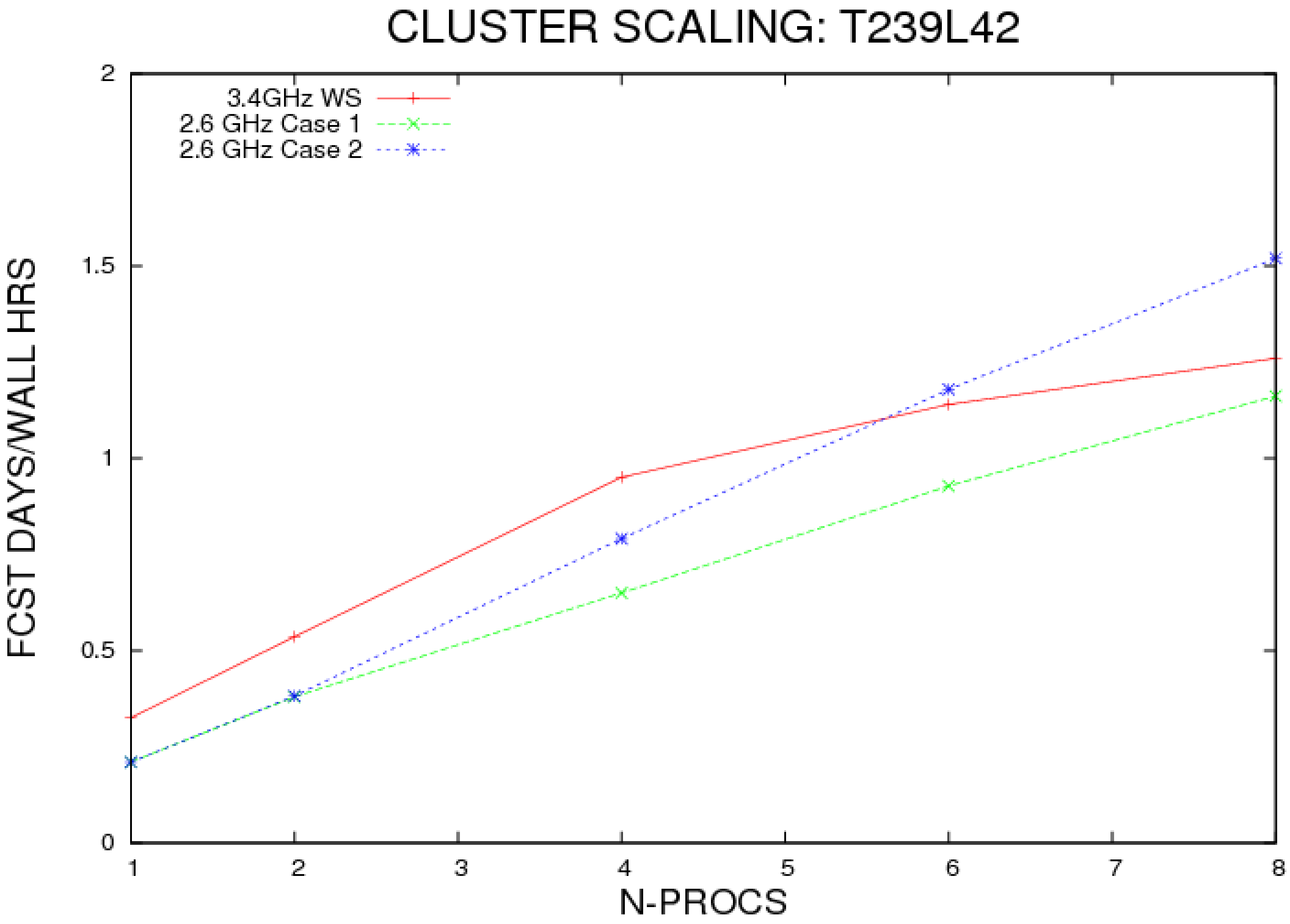
COMPUTATIONAL ISSUES

Why is this happening

- *It is easy (and cheap) to put more cores on a MB*
- *Marketing: appeals to video game industry*
- *Everything about the system bandwidth problem COSTS*
- *One of the byproducts of scientific computing de-emphasis*
- *Result:*
  - *Our applications don't scale as well as a few years ago*
  - *'Percentage of Peak' performance is degrading*

COMPUTATIONAL ISSUES

Impact of increasing processor/node ratios



COMPUTATIONAL ISSUES

Can we do anything about it?

• Given a choice, avoid extreme multi-core platforms
  - A multi-blade system connected with NIC's (e.g. Myrinet, Infiniband) will perform better than the equivalent multi-core system

• Realize there is no free lunch; if you really need a 'supercomputer', it will require an fast internal network and other expensive components

• Fortunately, often we don't need extreme scalability
  - For research, we just want a job finished by morning
  - In operational environments, total system throughput is often first priority, and clusters are ideal for this.

COMPUTATIONAL ISSUES

The future: petascale problems?

•Scalability is the limiting factor: problems must be HUGE
- extreme resolution (Atmosphere/Ocean models)
- very large ensembles (Covariance calculation)
- embarrassingly parallel as possible

•Very limited applications

• But: climate prediction is really a statistical problem
so may be our best application

• Unfortunately, DA is not a good candidate
- heterogeneous
- communication/IO intensive

COMPUTATIONAL ISSUES

Programming Languages

- Fortran
  - F77
  - F90/95

C/C++

- Convergence of languages?
  - Fortran standard becoming more object oriented
  - Expertise in Fortran hard to find
  - C++ language of choice for video games
  - But, investment in Fortran code immense

- Script languages
  - KSH
  - BASH (Bourne)
  - TCSH (C-shell)
  - Perl, Python, etc

COMPUTATIONAL ISSUES

Fortran, the original scientific language

- *Historically, Fortran is a language that allowed a programmer to get 'close to the hardware'*

- *Recent trends in Fortran standard (F90/95)*
  - *e.g. object oriented properties, are designed to hide hardware*
  - *Many features of questionable value for scientific computing*
  - *Ambiguities in standard can make use of 'exotic' features problematic*

- *Modern hardware with hierarchical memory systems is very difficult to manage*

- *Convergence with C/C++ probably inevitable*
  I won't have to worry about it, but you might
  Investment in Fortran software will be big obstacle

COMPUTATIONAL ISSUES

Writing parallel code

- *How many of you have written a parallel (especially MPI) code?*
- *If possible, start with a working serial, even 'toy' version*
- *Adhere to standards*
- *Make judicious use of F90/95 features, i.e. more F77 'like'*
    - *avoid 'exotic' features (structures, reshape, etc)*
    - *use dynamic memory, modules (critical for MPI applications)*
- *Use 'big endian' option on PC hardware*
- *Direct access IO produces files that are infinitely portable*
- *Remember, software lives forever!*

COMPUTATIONAL ISSUES

Fortran standard questions

- *Character data declaration: what is standard?*

```
Character(len=5) char
Character(5)  char
Character*5  char
```

- *Namelist input list termination: what is standard?*

```
var,
&end

var,
 $end

var
 /
```