

Climate Computing: The State of Play

Symposium on HPC and Data-Intensive Applications in Earth
Sciences

ICTP, Trieste ITALY

Keynote talk

V. Balaji

with contributions from many

NOAA/GFDL and Princeton University

13 November 2014

Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

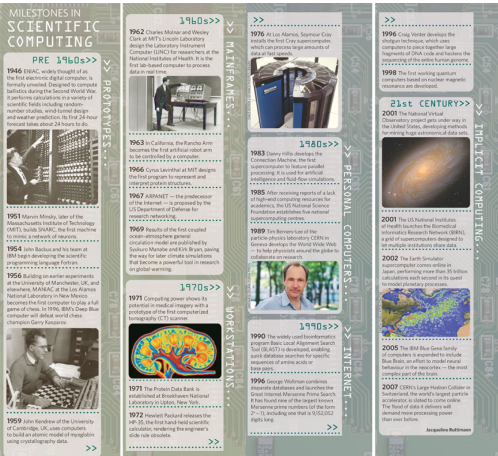
Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

GFDL and Princeton University

"Milestones in Scientific Computing", from Nature (23 March 2006)

- 1946 "ENIAC, ... the first electronic digital computer"
- 1969 "results from first coupled ocean-atmosphere general circulation model are published by Syukuro Manabe and Kirk Bryan, paving the way for later climate simulations that become a powerful tool in research on global warming...."
- 1972 "... the first hand-held scientific calculator"
- 1989 "Tim Berners-Lee ... develops the World Wide Web"



MPP, ShmemPP: Parallel Processing for Sceptics

V. Balaji
SGI/GFDL

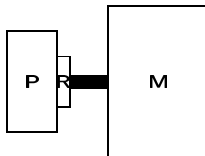
29 July 1998

The von Neumann model of computing

Sequential computing

The von Neumann model of computing conceptualizes the computer as consisting of a memory where instructions and data are stored, and a processing unit where the computation takes place. At each turn, we fetch an operator and its operands from memory, perform the computation, and write the results back to memory.

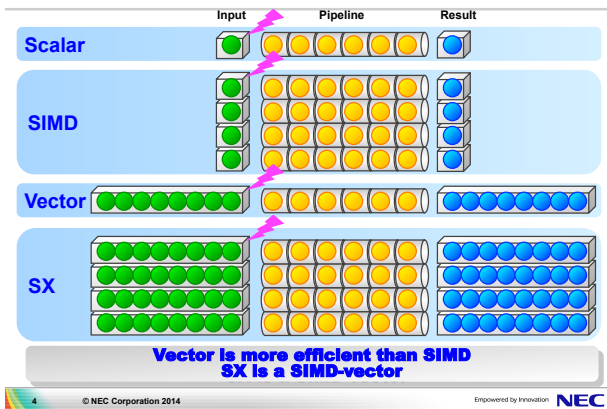
$a = b + c$



3

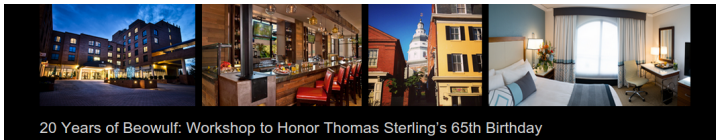
Scalars, vectors, parallel, parallel vectors

SIMD = Vector ?



Courtesy Rudi Fischer, NEC.

Commodity clusters: Beowulf



About
Call For Papers
Registration
Program Committee
Venue
Hotel
Directions
Agenda
Contact

About the Workshop

This workshop will mark the 20th anniversary of the introduction of commodity (AKA Beowulf) clusters, an architectural approach to creating parallel computers using mostly or entirely commodity components and open source system software. The initial target of the Beowulf cluster project was inexpensive, small to moderate parallel computing platforms; the Beowulf approach was extremely successful and adopted worldwide by teams ranging from high-school students to senior scientists. The Beowulf approach is now the basis of most of the world's most powerful computers as well.

The workshop will also celebrate the 65th birthday of Thomas Sterling, who has made major contributions over his career (so far), including playing a key role in conceiving and implementing commodity cluster computing (aka Beowulf), HPC architecture, run time systems, and exascale systems.

Dates

October 13-14, 2014



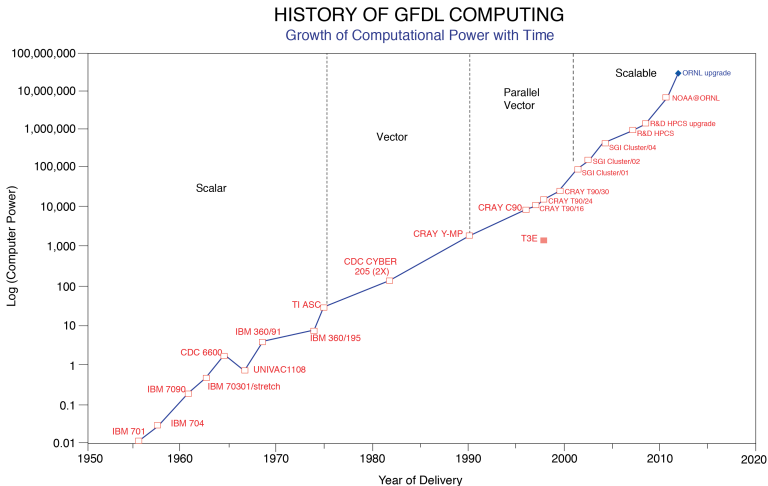
<http://crest.iu.edu/beowulf14/>

MPP, ShmemPP: Parallel Processing for Sceptics

V. Balaji
SGI/GFDL

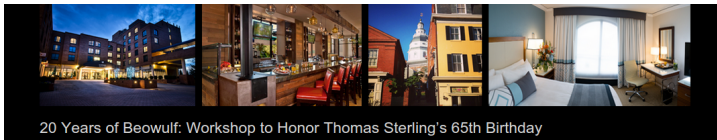
29 July 1998

History of GFDL Computing



Courtesy Brian Gross, NOAA/GFDL.

Commodity clusters: Beowulf



About
Call For Papers
Registration
Program Committee
Venue
Hotel
Directions
Agenda
Contact

About the Workshop

This workshop will mark the 20th anniversary of the introduction of commodity (AKA Beowulf) clusters, an architectural approach to creating parallel computers using mostly or entirely commodity components and open source system software. The initial target of the Beowulf cluster project was inexpensive, small to moderate parallel computing platforms; the Beowulf approach was extremely successful and adopted worldwide by teams ranging from high-school students to senior scientists. The Beowulf approach is now the basis of most of the world's most powerful computers as well.

The workshop will also celebrate the 65th birthday of Thomas Sterling, who has made major contributions over his career (so far), including playing a key role in conceiving and implementing commodity cluster computing (aka Beowulf), HPC architecture, run time systems, and exascale systems.

Dates

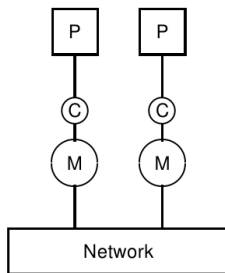
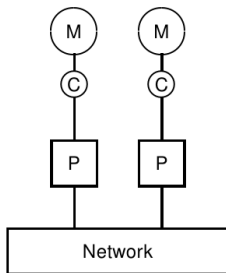
October 13-14, 2014



<http://crest.iu.edu/beowulf14/>

The commodity cluster era

- Speedups from Moore's Law: transistor density doubles every 18 months.
- Dennard scaling: power density is constant as fab shrinks.
- Moore's Law and Dennard scaling have both reached end of life!
- Networks and I/O still ripe for improvement.



Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile**
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

Climate modeling, a computational profile

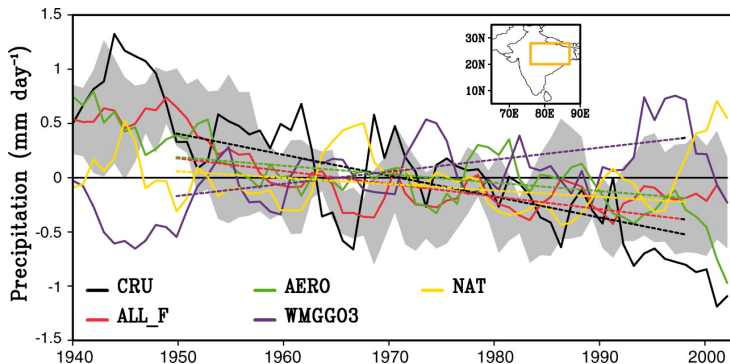
- Intrinsic variability at all timescales from minutes to millennia; all space scales from microbes to megacontinents.
- physics components have predictable data dependencies associated with grids;
- algorithms generally possess weak scalability;
- Adding processes and components improves scientific understanding;
- ... this complexity implies lots of diagnostic I/O;
- New physics and higher process fidelity at higher resolution;
- thus, coupled multi-scale multi-physics modeling;
- Ensemble methods to sample uncertainty (ICEs, PPEs, MMEs...)

In sum, climate modeling requires **long-term integrations** of **weakly-scaling, I/O and memory-bound** models of enormous complexity.

Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty**
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

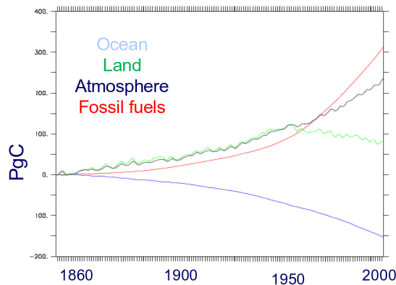
Aerosol indirect effects weaken South Asian monsoon



Cloud-aerosol feedbacks induce a weakening of the Indian monsoon (Figure courtesy Bollasina et al., **Science** 2011).

Carbon sources and sinks

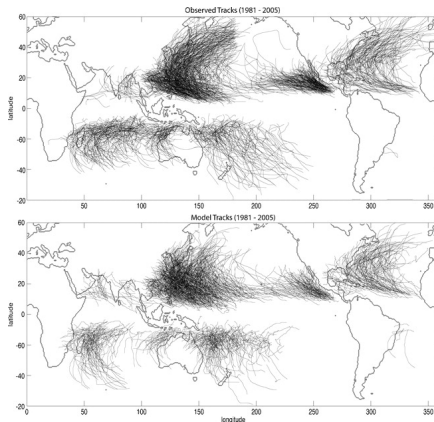
Cumulative Carbon Release into Atmosphere



- Land carbon fluxes dominant before 1960; then trend changes sign.
- Fossil fuels dominant contemporary source.
- Ocean uptake scales with $p\text{CO}_2$.

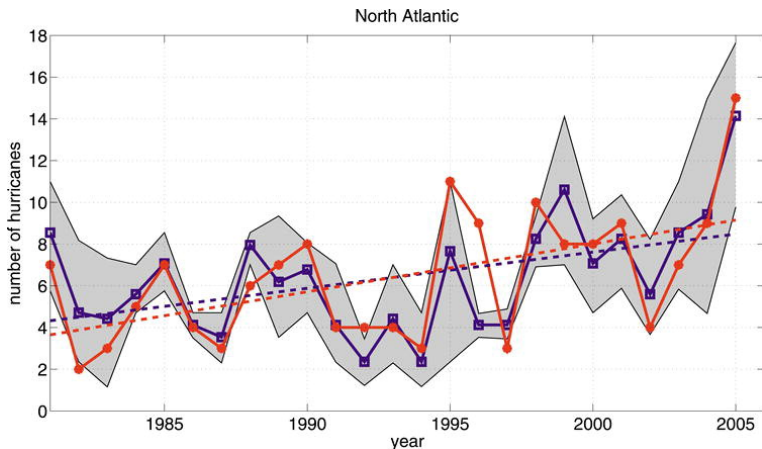
Figure courtesy Ron Stouffer, NOAA/GFDL; pre-publication.

Hurricane statistics from global high-resolution atmosphere models



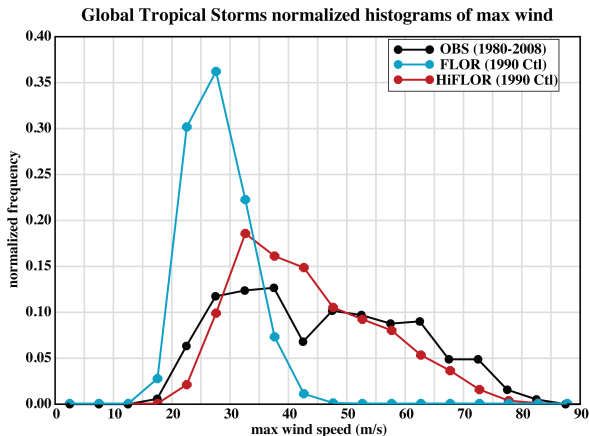
Observed and modeled hurricane tracks from 1981-2005 in a global 50 km (C180) atmospheric model forced by observed SSTs. (Figure 3 from Zhao and Held 2009).

Interannual variability of hurricane frequency



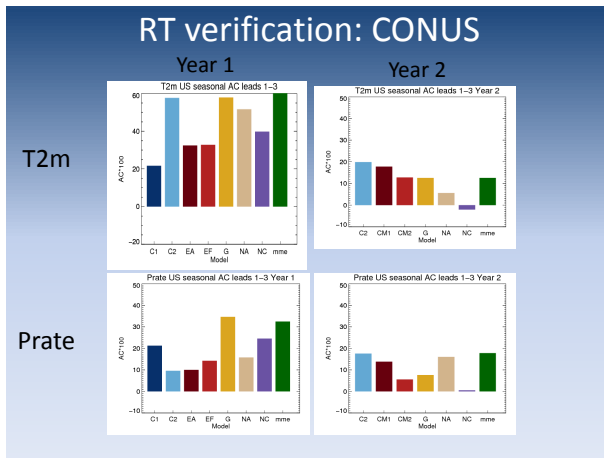
Interannual variability of W. Atlantic hurricane number from 1981-2005 in the C180 runs. (Figure 7 from Zhao and Held 2009).

"TC-permitting" models get better with resolution



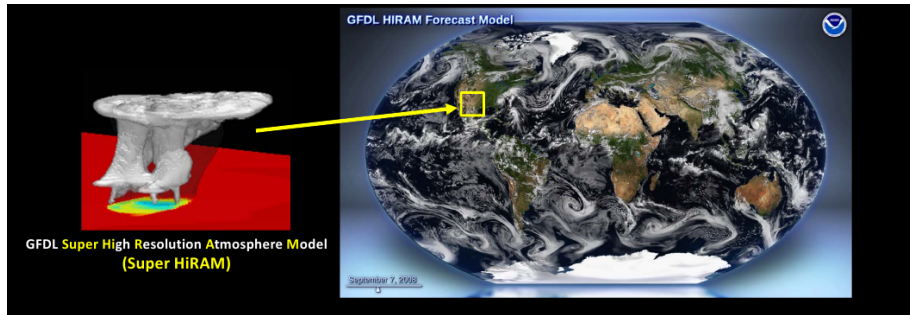
Intensity distribution improves with resolution. Figure courtesy Gabe Vecchi.

"TC-permitting" model FLOR is now used in the NMME



Seasonal forecasting product used in NMME and SPECS. Figure courtesy Gabe Vecchi.

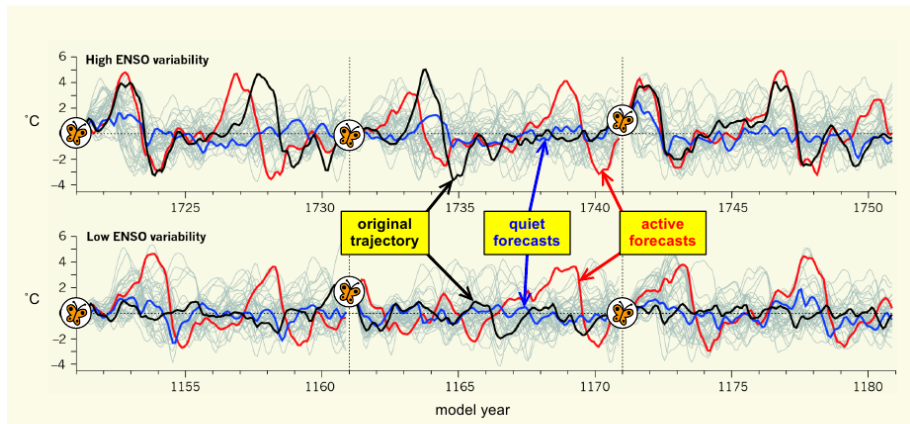
Towards global cloud-resolving models



Variable-resolution grid in the FV3 model, courtesy S-J Lin.

ENSO modulation: is it decadalally predictable?

“Perfect-model” forecasts of NINO3 SSTA, for extreme-ENSO epochs simulated by CM2.1



(External forcings held fixed at 1860 values.)

Wittenberg et al. (*J. Climate*, 2014)

Effects of the proverbial “flap of a butterfly’s wing”...

Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale**
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

The hardware jungle

Upcoming hardware roadmap looks daunting! GPUs, MICs, DSPs, and many other TLAs...

- Intel straight line: IvyBridge/SandyBridge, Haswell/Broadwell: “traditional” systems with threading and vectors.
- Intel knight’s move: Knights Corner, Knights Landing: MICs, thread/vector again, wider in thread space.
- Hosted dual-socket systems with GPUs: SIMD co-processors.
- BG/Q: CPU only with hardware threads, thread and vector instructions. No followon planned.
- ARM-based systems coming. (e.g with DSPs).
- FPGAs? some inroads in finance.
- Specialized processors: Anton for molecular dynamics, GRAPE for astrophysics.

The software zoo

Exascale using nanosecond clocks implies billion-way concurrency!
It is unlikely that we will program codes with $10^6 - 10^9$ MPI ranks: it will be MPI+X. Solve for X . . .

- CUDA and CUDA-Fortran: proprietary for NVIDIA GPUs. Invasive and pervasive.
- OpenCL: proposed standard, not much penetration.
- ACC from Portland Group, now a new standard OpenACC.
- Potential OpenMP/OpenACC merging...?
- PGAS languages: Co-Array Fortran, UPC, a host of proprietary languages.

GFDL between jungle and zoo

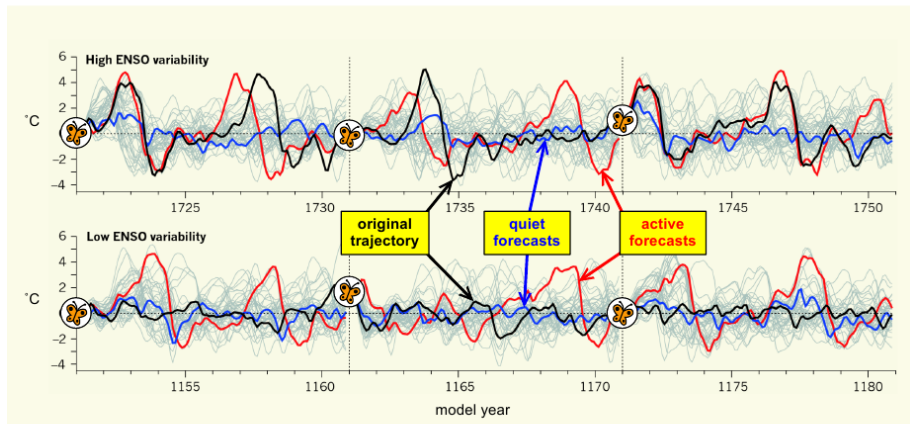
GFDL is taking a conservative approach:

- it looks like it will be a mix of MPI, threads, and vectors.
- Developing a three-level abstraction for parallelism: **components**, **domains**, **blocks**. Kernels work on blocks and must have vectorizing inner loops.
- **Recommendation: sit tight, make sure MPI+OpenMP works well, write vector-friendly loops, reduce memory footprint, offload I/O.**
- Other concerns:
 - Irreproducible computation
 - Tools for analyzing performance.
 - Debugging at scale.

Recent experience on Titan, Stampede and Mira reaffirm this approach.

ENSO modulation: is it decadalally predictable?

“Perfect-model” forecasts of NINO3 SSTA, for extreme-ENSO epochs simulated by CM2.1

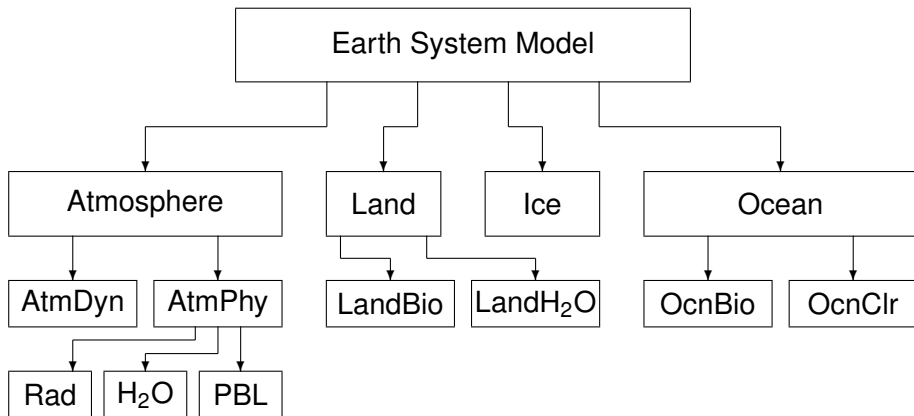


(External forcings held fixed at 1860 values.)

Wittenberg et al. (*J. Climate*, 2014)

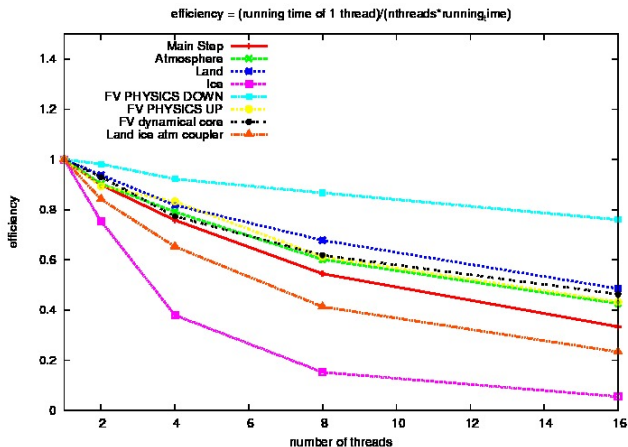
Effects of the proverbial “flap of a butterfly’s wing”...

Earth System Model Architecture



Complexity implies many different instruction sequences; no hotspots.

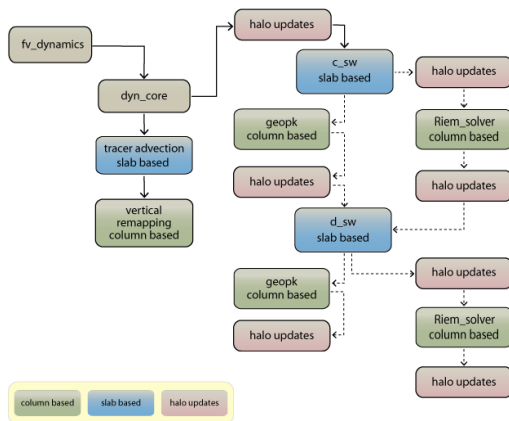
Most of FMS is now threaded/vectorized



CM4 on up to 16 threads on gaea. (Figure courtesy Zhi Liang, 16 Sep 2014)

Vectors (AVX) need to be used with care.

Analysis of dycore architecture for GPU/MIC



Study of code for MPI, threads, vectors. (Chris Kerr, Zhi, Kareem Sorathia (NASA), Duane Rosenberg (ORNL), Eric Dolven (Cray)...)

Blocking the dycore for GPU/MIC

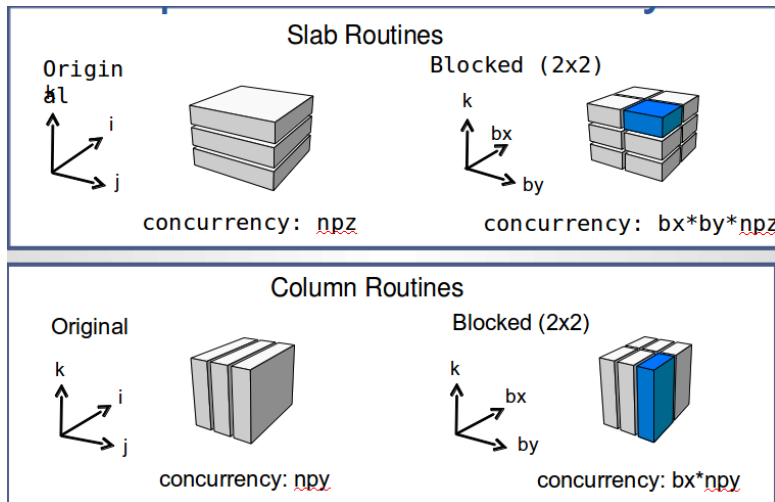


Figure courtesy Kareem Sorathia (NASA). Inner loops on i are retained for vectorization.

Performance summary: Xeon-SNB vs Xeon-Phi

Phi “speedup” over SNB:

- Overall: 0.73
- Communication: 0.34
- All Computation: 0.86
- Top 4: 0.996

Coding issues:

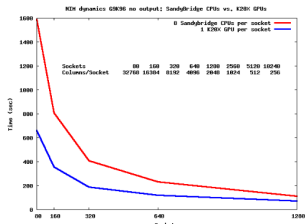
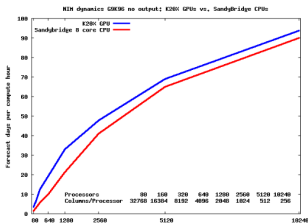
- Vector performance very hard to achieve, even with padding halos for alignment.
- Loop unrolling/stripmining/etc needs to be done by hand.
- Better performance analysis tools needed.

Courtesy Kareem Sorathia, NASA.

Results from NIM icosahedral dycore: SNB vs GPU

NIM Dynamics: GPU versus Intel-SB

- Single source code optimized for CPU, MIC & GPU
 - OpenMP directives for CPU & MIC
 - OpenACC, F2C-ACC for NVIDIA GPU
- 15 KM model, 96 levels, single-precision
 - Strong scaling: 80 - 10240 GPUs
 - GPU 2-3x faster than CPU socket for 8192 columns



Courtesy Mark Govett, NOAA/ESRL.

OpenACC

```
!$acc parallel num_gangs(ihe-ips+1) vector_length(64)
!$acc loop gang
    do ipn=ips,ihe
!$acc loop vector
    do k=1,nvl
        flxhi(k) = vnorm(k,edg,ipn)*dp_edg(k,edg,ipn)
```

Can merge gang and vector on same axis:

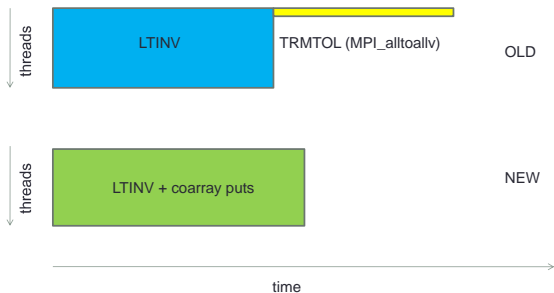
```
do k = kts,kte
!$acc loop gang vector
    do i = its,ite
        za(i,k) = 0.5*(zq(i,k)+zq(i,k+1))
```

Courtesy Mark Govett, NOAA/ESRL.

ECMWF uses PGAS (Co-Array Fortran)

iCAS2013, Annecy

Overlap Legendre transforms with associated transpositions



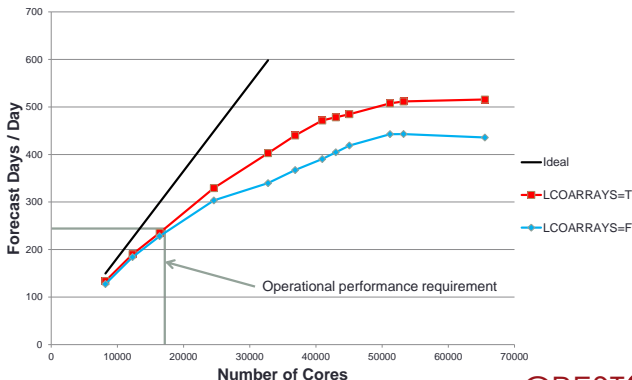
Co-array assignments become one-sided puts from within threaded regions.

Courtesy George Mozdzyński, ECMWF.

CAF results using Cray compiler CCE

iCAS2013, Annecy

T2047L137 model performance on HECToR (CRAY XE6)
RAPS12 IFS (CY37R3), cce=8.0.6 -hflex_mp=intolerant

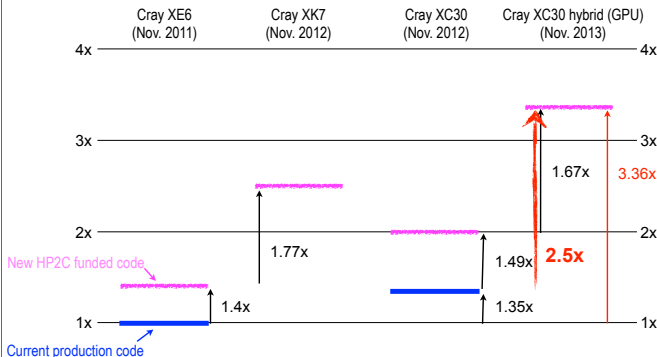


Courtesy George Mozdzynski, ECMWF.

COSMO: NWP production code using GPUs

ETH zürich

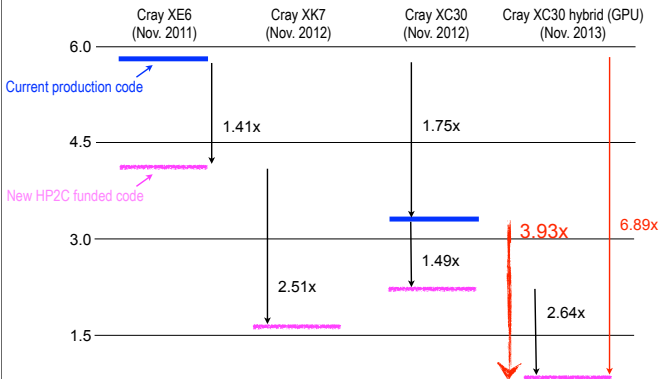
Speedup of the full COSMO-2 production problem (applies to apples with 33h forecast of Meteo Swiss)



COSMO: energy to solution

ETH zürich

Energy to solution (kWh / ensemble member)



Summary of results in the jungle and zoo

- Billion-way concurrency still a daunting challenge for everyone: no magic bullets anywhere to be found. ECMWF's PGAS approach is interesting, and there is at least one production GPU model.
- GPU/MIC based systems show nominal ~ 10 increase in flops/socket, but actual performance about 1-2X (thus percent of peak drops from $\sim 10\%$ to $\sim 1\%$)
- Software investment paid back in power savings (Schulthess).
- More computational intensity needs to be found: to fit 10^{18} op/s within a 1 MW power budget
 - an operation should be 1 pJ: data movement is ~ 10 pJ to main memory; ~ 100 pJ on network!
- DARPA: commodity improvements will slow to a trickle within 10 years: go back to specialized computing?
- DOE: double investment in exascale.

GFDL between jungle and zoo

GFDL is taking a conservative approach:

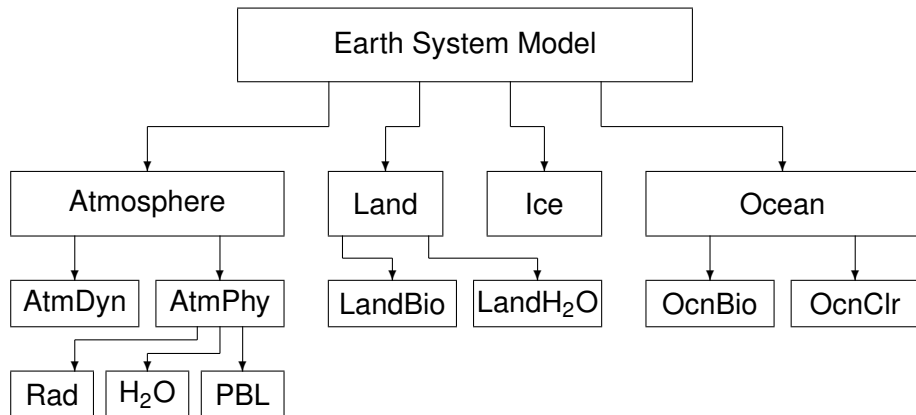
- it looks like it will be a mix of MPI, threads, and vectors.
- Developing a three-level abstraction for parallelism: **components**, **domains**, **blocks**. Kernels work on blocks and must have vectorizing inner loops.
- **Recommendation: sit tight, make sure MPI+OpenMP works well, write vector-friendly loops, reduce memory footprint, offload I/O.**
- Other concerns:
 - Irreproducible computation
 - Tools for analyzing performance.
 - Debugging at scale.

Recent experience on Titan, Stampede and Mira reaffirm this approach.

Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability**
- 6 Comparing real performance across models and machines

Earth System Model Architecture



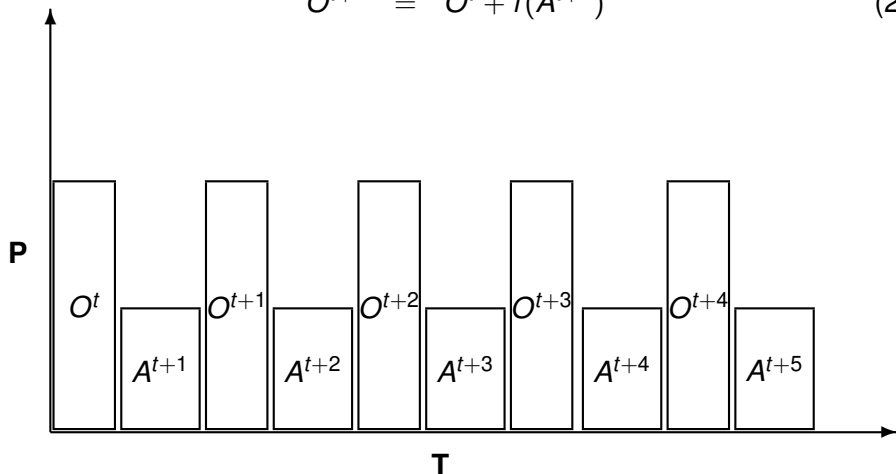
Extending component parallelism to $\mathcal{O}(10)$ requires a different physical architecture!

Serial coupling

Uses a forward-backward timestep for coupling.

$$A^{t+1} = A^t + f(O^t) \quad (1)$$

$$O^{t+1} = O^t + f(A^{t+1}) \quad (2)$$

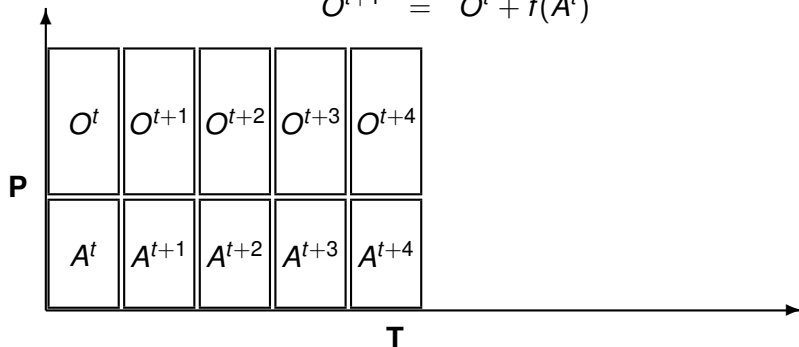


Concurrent coupling

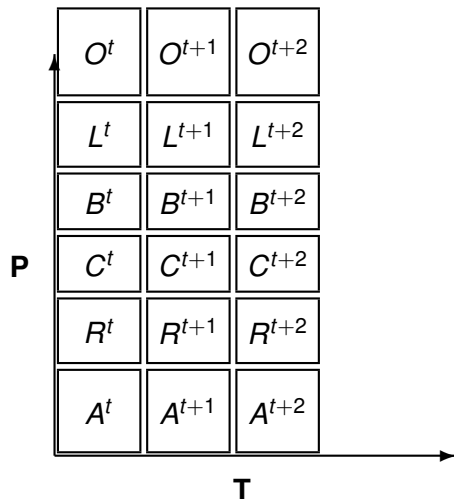
This uses a forward-only timestep for coupling. While formally this is unconditionally unstable, the system is strongly damped*. Answers vary with respect to serial coupling, as the ocean is now forced by atmospheric state from Δt ago.

$$A^{t+1} = A^t + f(O^t) \quad (3)$$

$$O^{t+1} = O^t + f(A^t) \quad (4)$$

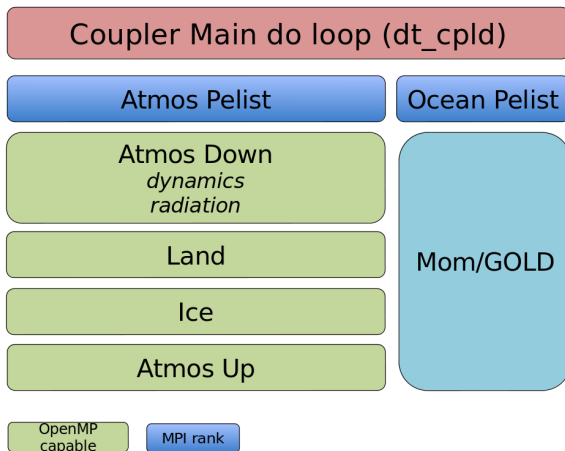


Massively concurrent coupling



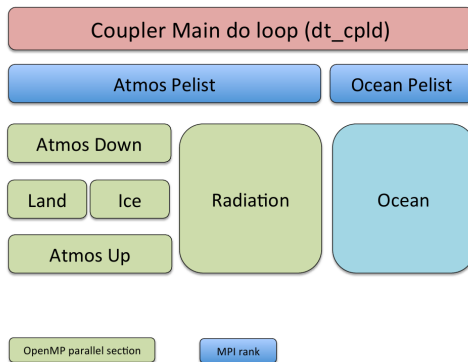
Components such as radiation, PBL, ocean biogeochemistry, each could run with its own grid, timestep, decomposition, even hardware. Coupler mediates state exchange.

Traditional coupling sequence



Radiation timestep much longer than physics timestep.
(Figure courtesy Rusty Benson, NOAA/GFDL).

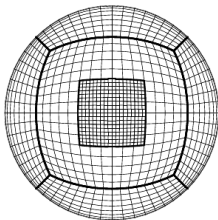
Concurrent radiation coupling sequence



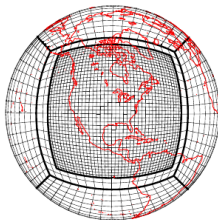
Physics and radiation share memory. Radiation executes on physics timestep from **lagged** state. Threads can be dynamically reassigned between components. This model has completed AMIP runs and further analysis is underway.

(Figure courtesy Rusty Benson, NOAA/GFDL).

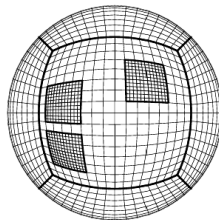
Nested grids



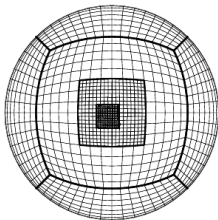
3:1 nested grid



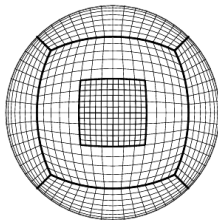
Large nest for RCMs



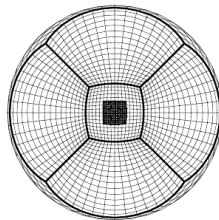
Multiple nests



Telescoping nests



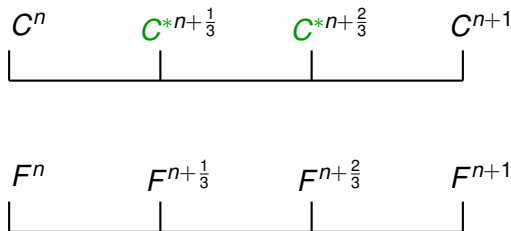
2:1 nested grid



Nest in stretched grid

Concurrent two-way nesting

Typical nesting protocols force serialization between fine and coarse grid timestepping, since the C^* are estimated by interpolating between C^n and C^{n+1} .



We enable concurrency by instead estimating the C^* by **extrapolation** from C^{n-1} and C^n , with an overhead of less than 10%. (See Harris and Lin 2012 for details.)

Outline

- 1 Some historical perspective
- 2 Climate modeling: a computational profile
- 3 Scientific drivers: complexity, resolution, uncertainty
- 4 Towards exascale
- 5 Adapting ESM architecture for scalability
- 6 Comparing real performance across models and machines

Multi-model ensembles for climate projection

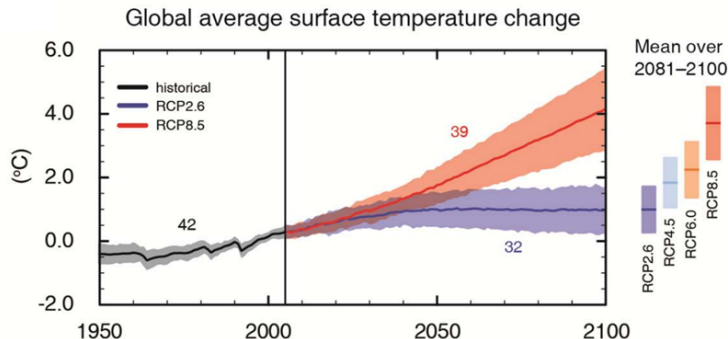
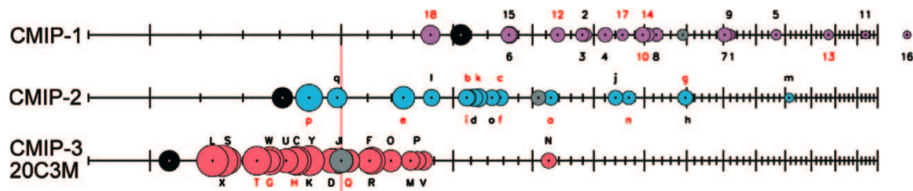


Figure SPM.7 from the IPCC AR5 Report. 20th century warming cannot be explained without greenhouse gas forcings.

Multi-model ensembles to overcome “structural uncertainty”



Reichler and Kim (2008), Fig. 1: compare models' ability to simulate 20th century climate, over 3 generations of models.

- Models are getting better over time.
- The **ensemble average** is better than any individual model.
- Improvements in understanding percolate quickly across the community.

NRC Report on “Advancing Climate Modeling”

The 2012 NRC Report “A National Strategy for Advancing Climate Modeling” (Google for URL...) made several recommendations:

- **Structural uncertainty**: key issue to be addressed with common modeling experiments: maintain model diversity while using common infrastructure to narrow the points of difference.
- **Global data infrastructure** as critical infrastructure for climate science: data interoperability, common software requirements.
- “Nurture” at least one unified **weather-climate** effort: NWP methods to address climate model biases; climate runs to address drift and conservation in weather models.
- **Forum** to promote shared infrastructure: identify key scientific challenges, design common experiments, set standards for data interoperability and shared software.

Real model performance: some considerations

- Productions runs may be configured for **capability** (minimizing time to solution or **SYPD**) or **capacity** (minimizing allocation or **CHSY**).
- Computing resources can be applied to resolution or **complexity**: what is a good measure of model complexity?
- ESM architecture governs **component concurrency**: need to measure **load balance** and **coupler cost**.
- Codes are **memory-bound**: locate **bloat** (memory copies by user or compiler).
- Models configured for scientific analysis bear a significant **I/O load** (can interfere with optimization of computational kernels). **Data intensity** (GB/CH) is a useful measure for designing system architecture.
- **Actual SYPD** tells you if you need to devote resources to system and workflow issues rather than optimizing code.

Analysis of several GFDL models

- Measure overall computation cost for capability (**S**peed) or capacity (**T**hroughput) configurations.
- Measure complexity as number of prognostic variables in the model. (There may be better measures based on cluster coefficients, etc.)
- Measure coupler cost and load imbalance separately.
- Measure memory bloat as actual memory (resident set size) compared to ideal memory (number of variables \times data domain size).
- Measure I/O load by rerunning model with diagnostics off. (input files and restart files are considered an unavoidable cost and aren't counted here.)
- Measure actual SYPD for a complete run (from when you typed **run** to when the last history file was archived).

Land and Ice components are ignored in this analysis.

Analysis of GFDL models: results

Model	Resolution	Cmplx.	SYPD	CHSY	Coupler	Load Imb.	I/O	MBloat	ASYPD
CM2.6 S	A0.5L32 O0.1L50	18	2.2	212,465	5.71%	20%		12%	1.6
CM2.6 T	A0.5L32 O0.1L50	18	1.1	177,793	1.29%	60%	24%	12%	0.4
CM2.5 T	A0.5L32 O0.25L50	18	10.9	14,327	17%	0%			6.1
FLOR T	A0.5L32 O1L50	18	17.9	5,844	0%	57%	5.1%	31%	12.8
CM3 T	A2L48 O1L50	124	7.7	2,974	0.5%	41%	14.76%	3%	4.9
ESM2G S	A2L24 O1L50	63	36.5	279	8.91%	1%		34%	25.2
ESM2G T	A2L24 O1L50	63	26.4	235	2.63%	22%		34%	11.4

- More details are available (layout on MPI/thread, aggregate I/O per CH or SD, platform, optimization, cost per component...)
- Is this a basis for a cross-model comparison of performance (CPMIP, anyone?) for a common understanding of the roadblocks to performance?

Preliminary cross-model comparisons

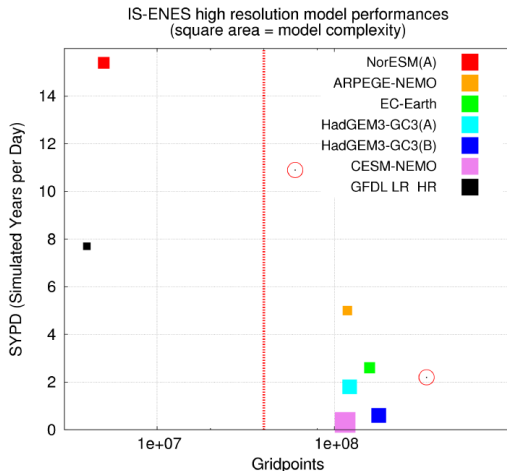


Figure courtesy Eric Maisonnave, Joachim Biercamp, Giovanni Aloisio and others on the ISENES2 team.

Preliminary cross-model comparisons: layout

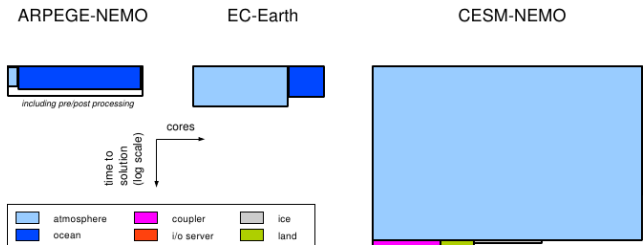
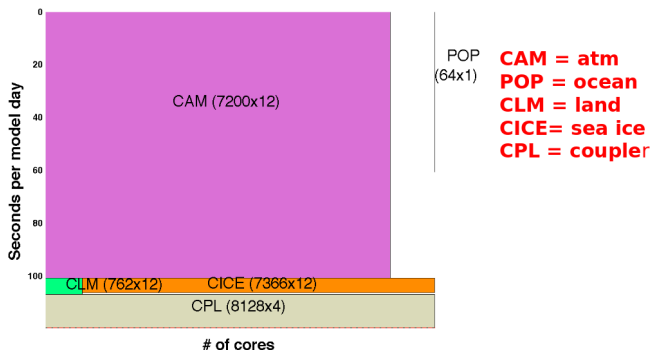


Figure 2: Parallelism and execution time (inverse of SYPD)
for three of the participating ESMs

Figure courtesy Eric Maisonnave, Joachim Biercamp, Giovanni Aloisio and others on the ISENES2 team.

CESM component-wise layout and simulation rate on MIRA



Courtesy John Dennis and Rich Loft, NCAR. 0.25° atmosphere, 1° ocean on 32k cores of Mira at ~ 2 SYPD.

Conclusions

- The commodity computing era has taken us from the von Neumann model to the “**sea of functional units**” (Kathy Yelick’s phrase). Not easy to understand, predict or program performance.
- The “**free lunch**” decade encouraged us to indulge in very abstract programming, and now they’ve come to take away your plates.
- The “**component**” abstraction still may let us extract some benefits out of the machines of this era:
 - sharing of the wide thread space.
 - distribute components among heterogeneous hardware?
- Can we approach models as experimental **biological** systems? (single organism or “**cell line**” not exactly reproducible; only the ensemble is.)
- Radically new computing paradigms – neuromorphic, biological, quantum – are several acquisition cycles away.
- The NWP and climate communities are all in the same boat: greater cooperation is advisable (NRC Report 2012).