An overview of GFDL's Flexible Modeling System

Advanced School on Earth System Modeling
Indian Institute of Tropical Meteorology, Pune

V. Balaji

NOAA/GFDL and Princeton University

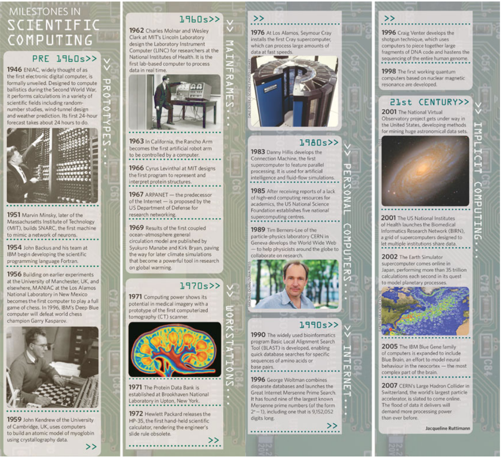
18 July 2016

Outline

Outline

GFDL and Princeton University

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


MILESTONES IN SCIENTIFIC COMPUTING

PRE 1960s>>

1946 ENIAC, widely thought of as the first electronic digital computer, is formally unveiled. Designed to compute ballistics during the Second World War, it performs calculations in a variety of scientific fields including random-number studies, wind-tunnel design and weather prediction. Its first 24-hour forecast takes about 24 hours to do.

1951 Marvin Minsky, later of the Massachusetts Institute of Technology (MIT), builds SNARC, the first machine to mimic a network of neurons.

1954 John Backus and his team at IBM begin developing the scientific programming language Fortran.

1956 Building on earlier experiments at the University of Manchester, UK, and elsewhere, MANIAC at the Los Alamos National Laboratory in New Mexico becomes the first computer to play a full game of chess. In 1996, IBM's Deep Blue computer will defeat world chess champion Garry Kasparov.

1959 John Kendrew of the University of Cambridge, UK, uses computers to build an atomic model of myoglobin using crystallography data.

>> PROTOTYPES...

1960s>>

1962 Charles Molnar and Wesley Clark at MIT's Lincoln Laboratory design the Laboratory Instrument Computer (LINC) for researchers at the National Institutes of Health. It is the first lab-based computer to process data in real time.

1963 In California, the Rancho Arm becomes the first artificial robot arm to be controlled by a computer.

1966 Cyrus Levinthal at MIT designs the first program to represent and interpret protein structures.

1967 ARPANET — the predecessor of the Internet — is proposed by the US Department of Defense for research networking.

1969 Results of the 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.

>> MAINFRAMES...

1970s>>

1971 Computing power shows its potential in medical imagery with a prototype of the first computerized tomography (CT) scanner.

1971 The Protein Data Bank is established at Brookhaven National Laboratory in Upton, New York.

1972 Hewlett Packard releases the HP-35, the first hand-held scientific calculator, rendering the engineer's slide rule obsolete.

1976 At Los Alamos, Seymour Cray installs the first Cray supercomputer, which can process large amounts of data at fast speeds.

>> WORKSTATIONS...

1980s>>

1983 Danny Hillis develops the Connection Machine, the first supercomputer to feature parallel processing. It is used for artificial intelligence and fluid-flow simulations.

1985 After receiving reports of a lack of high-end computing resources for academics, the US National Science Foundation establishes five national supercomputing centres.

1989 Tim Berners-Lee of the particle-physics laboratory CERN in Geneva develops the World Wide Web — to help physicists around the globe to collaborate on research.

>> PERSONAL COMPUTERS...

1990s>>

1990 The widely used bioinformatics program Basic Local Alignment Search Tool (BLAST) is developed, enabling quick database searches for specific sequences of amino acids or base pairs.

1996 George Woltman combines disparate databases and launches the Great Internet Mersenne Prime Search. It has found nine of the largest known Mersenne prime numbers (of the form 2^n-1), including one that is 9,152,052 digits long.

1996 Craig Venter develops the shotgun technique, which uses computers to piece together large fragments of DNA code and hastens the sequencing of the entire human genome.

1998 The first working quantum computers based on nuclear magnetic resonance are developed.

>> INTERNET...

21st CENTURY>>

2001 The National Virtual Observatory project gets under way in the United States, developing methods for mining huge astronomical data sets.

2001 The US National Institutes of Health launches the Biomedical Informatics Research Network (BIRN), a grid of supercomputers designed to let multiple institutions share data.

2002 The Earth Simulator supercomputer comes online in Japan, performing more than 35 trillion calculations each second in its quest to model planetary processes.

2005 The IBM Blue Gene family of computers is expanded to include Blue Brain, an effort to model neural behaviour in the neocortex — the most complex part of the brain.

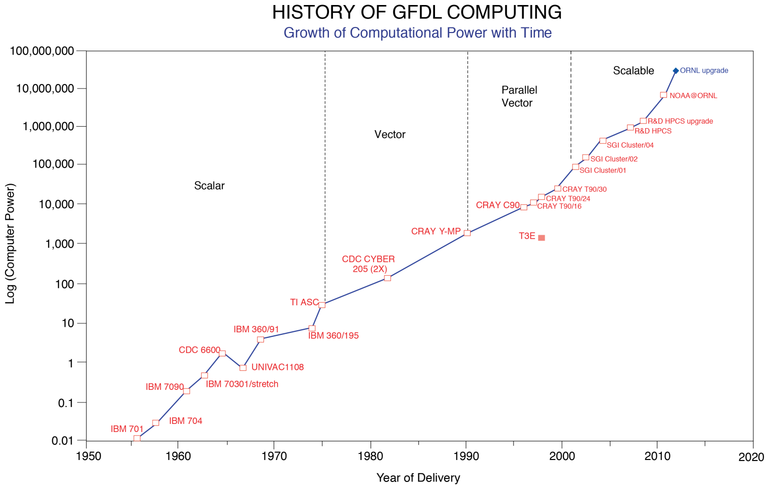
2007 CERN's Large Hadron Collider in Switzerland, the world's largest particle accelerator, is slated to come online. The flood of data it delivers will demand more processing power than ever before.

Jacqueline Ruttimann

>> IMPLICIT COMPUTING...

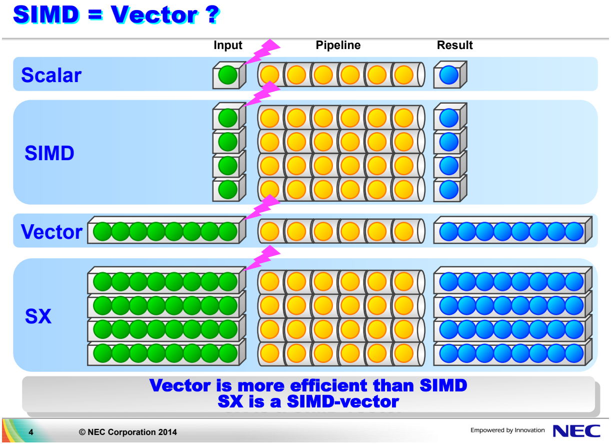
- 1946 "ENIAC, ... the first electronic digital computer"
- 1969 "results from first coupled ocean-atmosphere general circulation model are published by Syukuro Manabe and Kirk Bryan, paving the way for later climate simulations that become a powerful tool in research on global warming...."
- 1972 "...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

History of GFDL Computing



Courtesy Brian Gross, NOAA/GFDL.

Scalars, vectors, parallel, parallel vectors



Courtesy Rudi Fischer, NEC.

Commodity clusters: Beowulf

20 Years of Beowulf: Workshop to Honor Thomas Sterling's 65th Birthday

- 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

sighpc

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

GFDL Strategic Plan: 2012-2016

- Basic climate processes and their representations in models.
- Comprehensive modeling of climate system variability and change.
- Understanding, detection and attribution, and prediction of extreme events.
- Understanding, detection and attribution, and predictability of modes of climate variability.
- Cryospheric amplification of climate change and sea-level rise.
- Understanding the Earth system including biosphere and human activities.
- Climate science, impacts and services.

Google “GFDL Strategic Science Plan”.

Current suite of GFDL models

- CM3: comprehensive tropospheric and stratospheric chemistry, aerosol-cloud feedbacks.
- ESM2M and ESM2G: free-running carbon cycle.
- DECP: decadal prediction models at various resolutions with advanced initialization (ECDA).
- HiRAM: atmospheric models with AM3 physics optimized for tropical storm "permitting" simulations (HiRAM).
- Cloud-resolving models (C2560) with bulk microphysics.
- Under development for CM4: unified ocean core MOM6, simplified aerosol chemistry.
- Performance guidelines for CMIP-class models: 4 models running at 100 years/month using half the available machine.
- Spinup and millennial control runs are capability runs. Note ESMs require very long spinup…

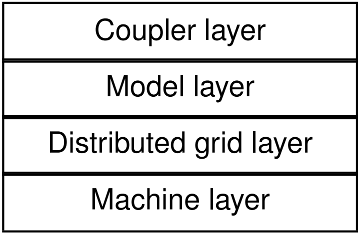
All models built on common framework and run within a single distributed workflow.

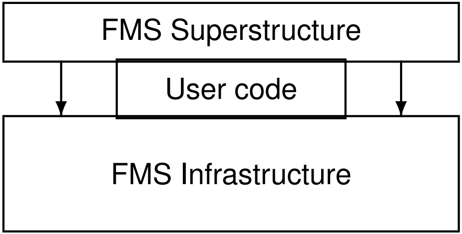
Outline

FMS: Summary

- Scalable high-performance framework on up to $\mathcal{O}(10^5)$ processors.
- Good, stable, dedicated team in Modeling Services.
- Broad acceptance and widespread contributions to a working system: many useful contributions from external users.
- Impressive list of features: mosaics, parallel ensemble capability, experiment database. Equally impressive list of components and options.
- Component list:
  - atmosphere dycore: FV-CS, FV-LL, BGRID, SPECTRAL, ZETAC.
  - atmospheric physics and chemistry: AM2, AM2.1 (HiRAM), AM2.1 (Lin), AM3, simple, dry. Simple-Chem
  - ocean: MOM6, GOLD, MOM5, MOM4p1, MOM4p0, mixed-layer.
  - land: LAD/LM2, SHE/LM3v, LAD2/LM3, river.
  - ocean BGC: TOPAZ, COBALT, BLING.
  - ice: SIS, SIS2.

FMS is in its second decade of active use

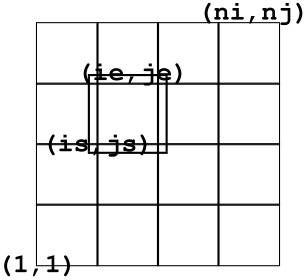




- Flexible Modeling System effort began in 1998, when GFDL first moved on to distributed memory machines
- Provided simplified interface to parallelism and I/O: **`mpp`**. Abstract types for “distributed grid” and “fields”.
- Diagnostic output, data override, time manager.
- Component-based design, abstract types for component state vectors, exchange grid.
- “Sandwich” model influential in community.

User interface to communication kernels

- **mpp_define_domains()**
- **mpp_update_domains()**




```
type(domain2D) ::  domain
call mpp_define_domains( (/1,ni,1,nj/), domain, &
   xhalo=2, yhalo=2 )
call mpp_update_domains( f, domain )
!perform computations on f
```

(1)

Checksums

Floating-point arithmetic is not *associative*: $a + (b + c)$ need not equal $(a + b) + c$. The reason is that FP numbers are stored with an *exponent* and a *mantissa*: the mantissa can only hold a finite number of decimal places (about 7 for 4-byte **`float`**, 15 for 8-byte **`double`**).

- So if **`a=1, b=5e-16`**, **`a+b=1.000...0000d0`**
- If **`a=1, c=1e-15`**, **`a+c=1.000...0001d0`**
- But **`b+b=1e-15`**
- So **`a+(b+b)=1.000...0001d0`**
- But **`(a+b)+b=1.000...0000d0`**

Since climate/weather are chaotic system, this small "order of operations" difference can lead to different solutions.

Checksums (**`mpp_chksum`**) are a method we use to ensure that a small code change does not lead to differences in FP answers.

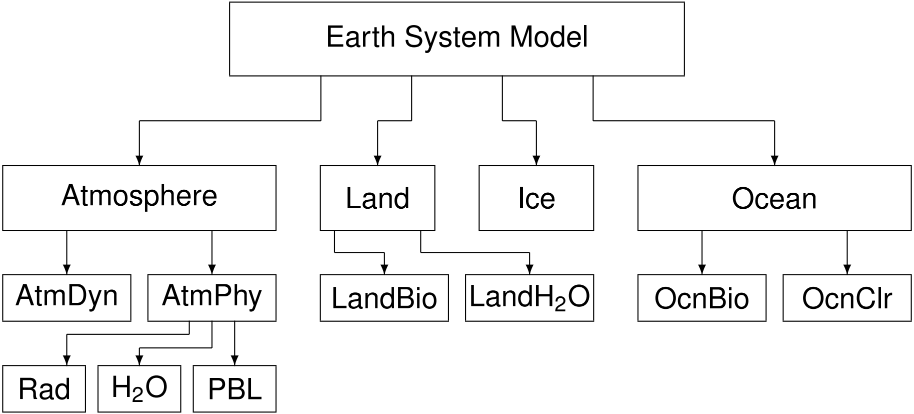
Timers

You can time any code section using *timing calipers*:

```
id = mpp_clock_id( 'Ocean model' )
...
call mpp_clock_begin(id)
call ocean_model
call mpp_clock_end(id)
...
call mpp_exit()
```
(2)

At the end you will get timing statistics for that code section across all PEs (min, max, avg, std: the standard deviation is a measure of load imbalance).

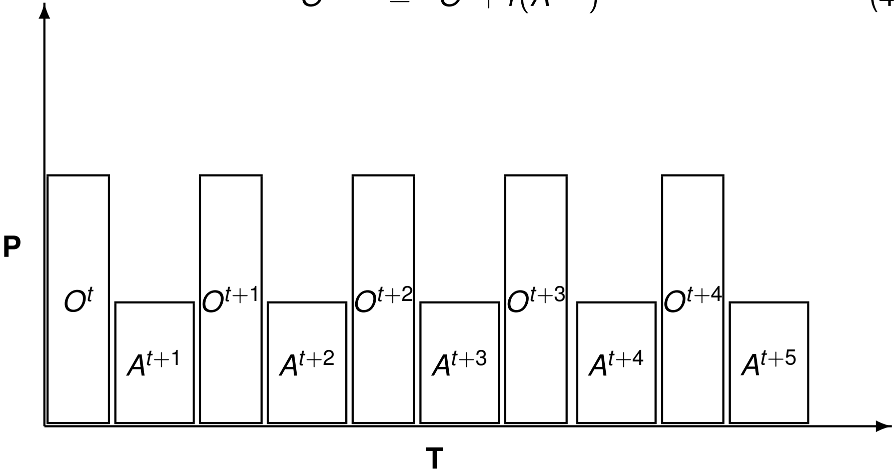
FMS Earth System Model Architecture



Serial coupling

Uses a forward-backward timestep for coupling.

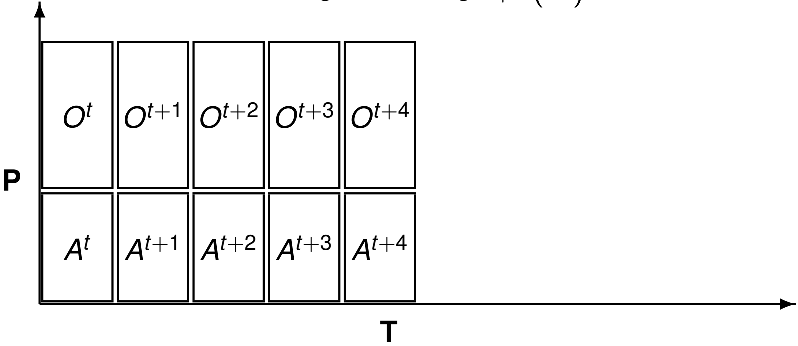
$$A^{t+1} = A^t + f(O^t) \quad (3)$$
$$O^{t+1} = O^t + f(A^{t+1}) \quad (4)$$



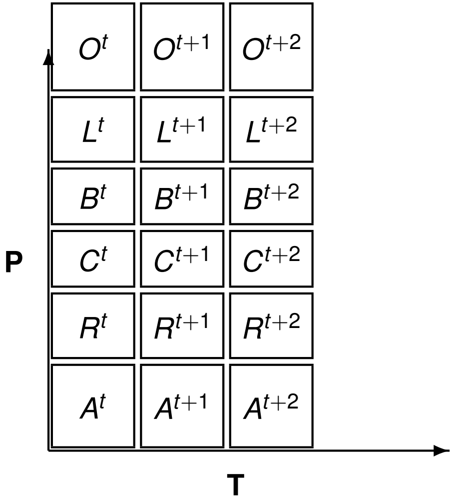
Concurrent coupling

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.

$$
\begin{aligned}
A^{t+1} &= A^t + f(O^t) \quad (5)\\
O^{t+1} &= O^t + f(A^t) \quad (6)
\end{aligned}
$$



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.

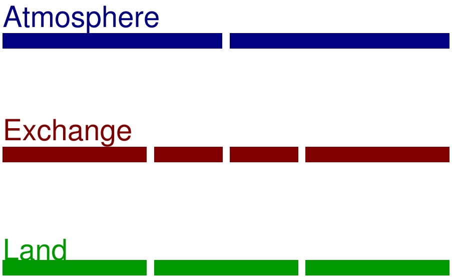
Implicit coupling and the exchange grid

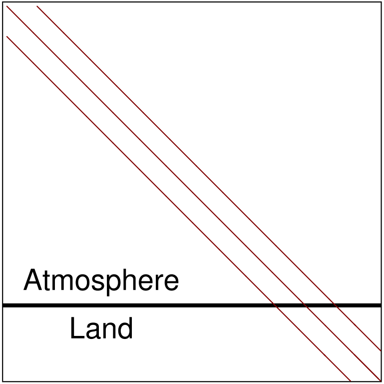
Fluxes at the surface often need to be treated using an implicit timestep. (e.g temperature flux in near-surface layers that can have vanishingly small heat capacity.)

$$
\begin{aligned}
\frac{\partial T}{\partial t} &= K\frac{\partial^2 T}{\partial z^2} && (7)\\
\frac{T_k^{n+1} - T_k^n}{\Delta t} &= K\frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} - 2T_k^{n+1}}{\Delta z^2} && (8)\\
\mathbf{A}\mathbf{T}^{n+1} &= \mathbf{T}^n && (9)
\end{aligned}
$$

Implicit coupling and the exchange grid

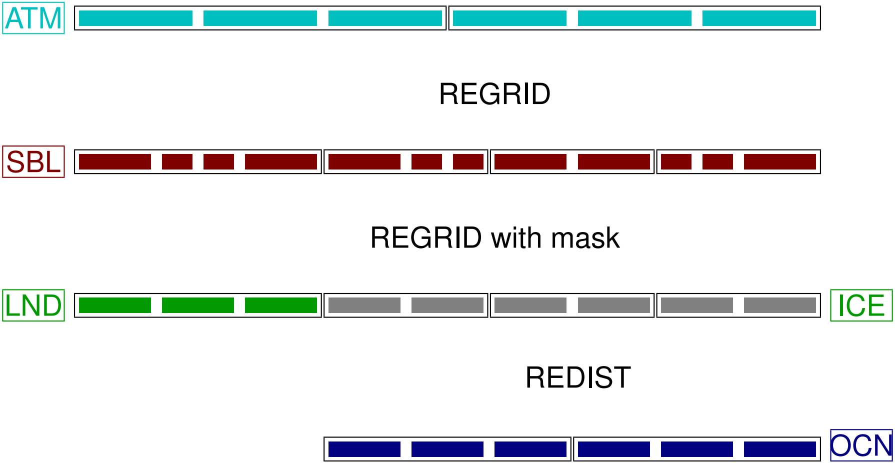
Tridiagonal solver in Eq. 9 across multiple components and grids.



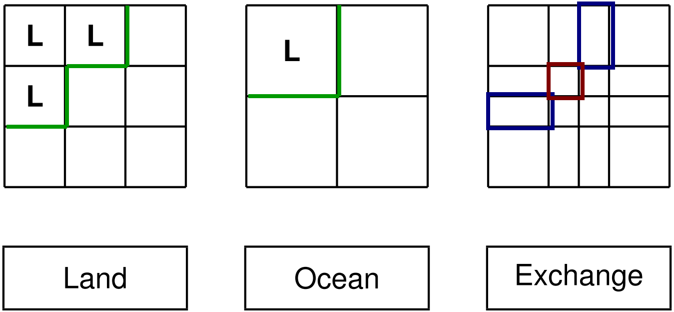


Coupled architecture with SBL on exchange grid

Parallelism in the FMS coupler



The mask problem



An issue arises when grids of two independent components (e.g land and sea) share a boundary. The boundary is defined by a **mask** (e.g land-sea mask) but the mask is discretized independently on the two grids. However, exchange grid cells need to be uniquely assigned to a single component. This means that some cells get **clipped** on one or the other grid. In FMS, by convention, we choose to clip the land grid.

Gaea

The NOAA Climate Modeling and Research System *Gaea*. Extended in 2013 to include GPU capabilities.

FRE: the FMS Runtime Environment

- Operational since 2003, designed to provide an environment for integrated testing and production.
- Rigorous standardized test procedure for evaluating new code and new model assemblies and configurations.
- Integrated existing post-processing structure.
- Captures complete configuration from source assembly to compilation to running, post-processing and analysis.
- Simulation database provides retrieval of model output, model analysis, and now model state and configuration information.
- Again influential in community, with "curators" being prototyped at various sites.

The FMS user interface: FRE

Comprehensive website for all information and documentation:
`http://www.gfdl.noaa.gov/~fms`

- Source code maintenance under CVS; browse over the net using webCVS.
- Model configuration, launching and regression testing encapsulated in XML;
- Relational database for archived model results;
- Standard and custom diagnostic suites;

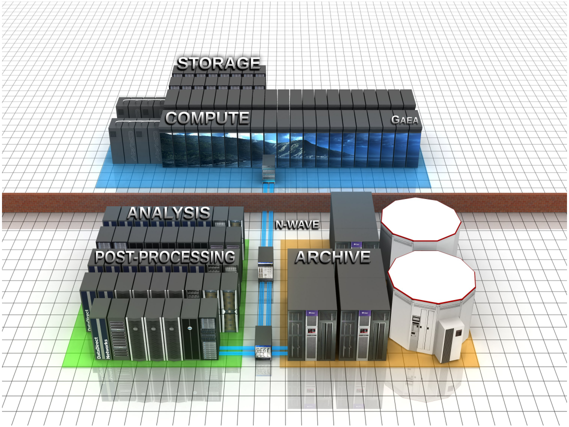
The FMS Runtime Environment (FRE) describes all the steps for configuring and running a model jobstream; archiving, postprocessing and analysis of model results.
`fremake, frerun, frepp, frecheck, ...`
The Regression Test Suite (RTS) is a set of tests that are run continuously on a set of FMS models to maintain and verify code integrity.

Elements of FRE

- **fremake** Checkout an appropriate subset of the FMS source code for an experiment and create an executable;
- **frerun** run an experiment in multiple *segments*; resubmit if necessary;
- **frestatus** check the status of an experiment that is underway;
- **frelist** list available experiments;
- **frepriority** switch a job sequence between queues;
- **frecheck** run RTS checks for bitwise accuracy;
- **frepp** FRE post-processing: create time series, time averages, and plots;
- **frescrub** remove intermediate and redundant files;
- **freppcheck** RTS checks on history and post-processing files.
- **fredb** enter experiments into Curator DB.

URL: `http://www.gfdl.noaa.gov/fms/fre`

Gaea and GFDL



FRE and other elements in the GFDL modeling environment manage the complex scheduling of jobs across a distributed computing resource.

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.
- Code generation:
  - Domain-specific languages (DSLs): e.g STELLA.
  - Source-to-source translators.

GFDL between jungle and zoo

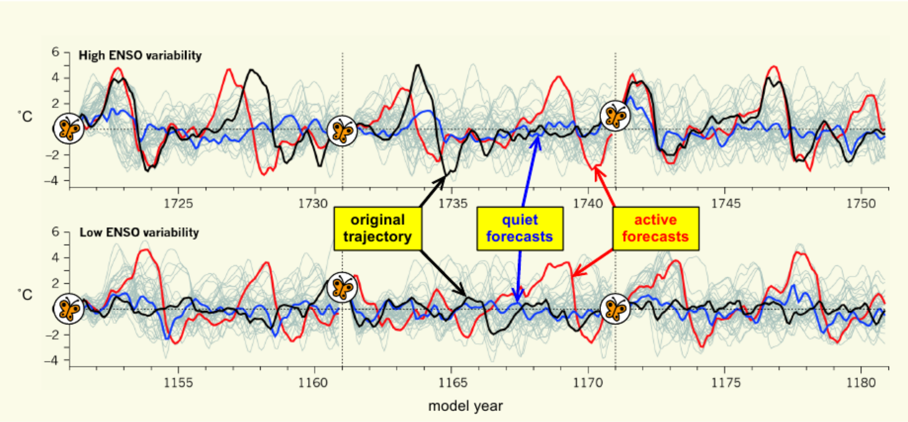
GFDL is taking a conservative approach:

- 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 decadally predictable?

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

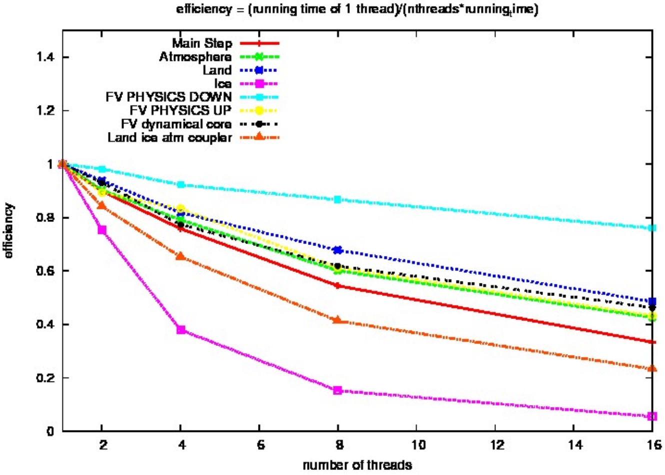


(External forcings **held fixed** at 1860 values.)

Wittenberg et al. (J. Climate, 2014)

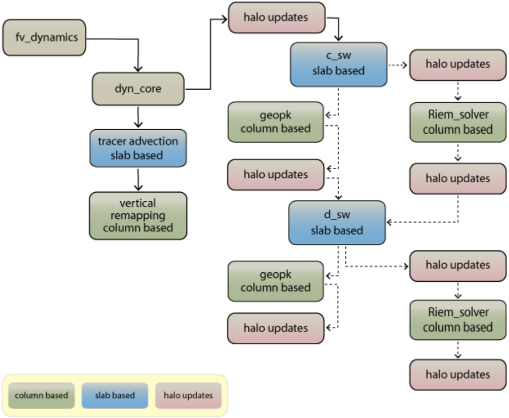
Effects of the proverbial “flap of a butterfly’s wing”...

Most of FMS is now threaded



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

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

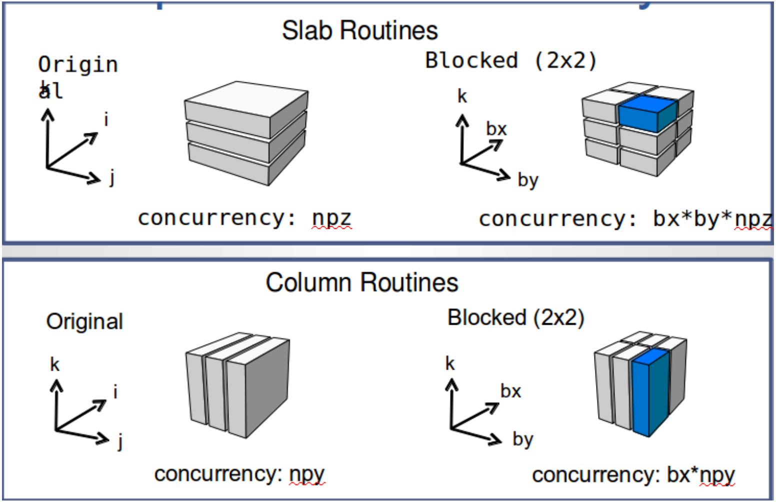


Figure courtesy Kareem Sorathia (NASA). Inner loops on **`i`** are retained for vectorization.

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:

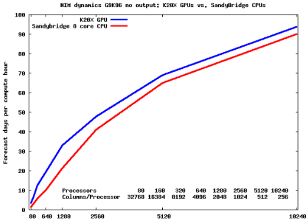
- Vector performance very hard to achieve, even with padding halos for alignment.
- Loop unrolling/stripmining/etc needs to be done by hand.
- Better performance analysis tools needed.

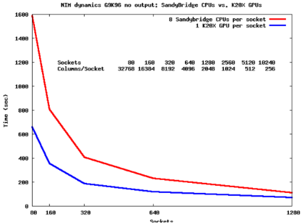
Courtesy Kareem Sorathia, NASA.

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

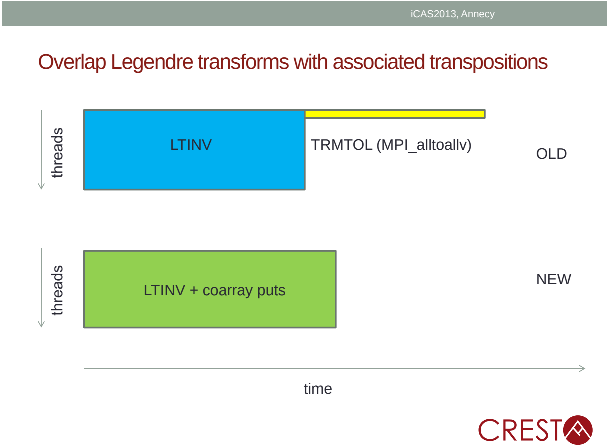
```
!$acc parallel num_gangs(ihe-ips+1) vector_length(64)
!$acc loop gang
    do ipn=ips,ihe
!$acc loop vector
      do k=1,nvl
        flxhi(k) = vnorm(k,edg,ipn)*dp_edg(k,edg,ipn)
```

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))
```

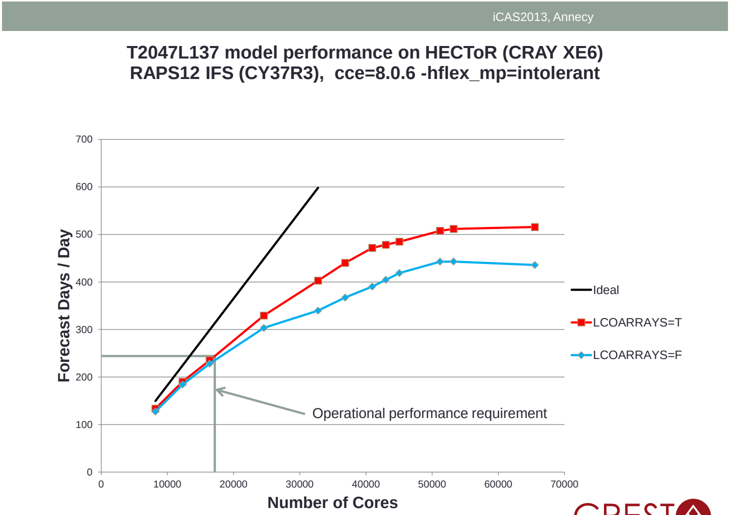
Courtesy Mark Govett, NOAA/ESRL.

ECMWF uses PGAS (Co-Array Fortran)



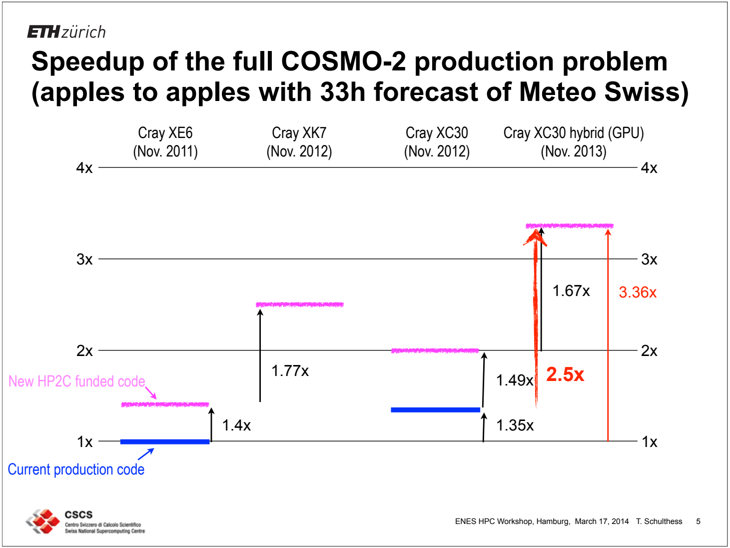
Co-array assignments become one-sided puts from within threaded regions.
Courtesy George Mozdzynski, ECMWF.

CAF results using Cray compiler CCE

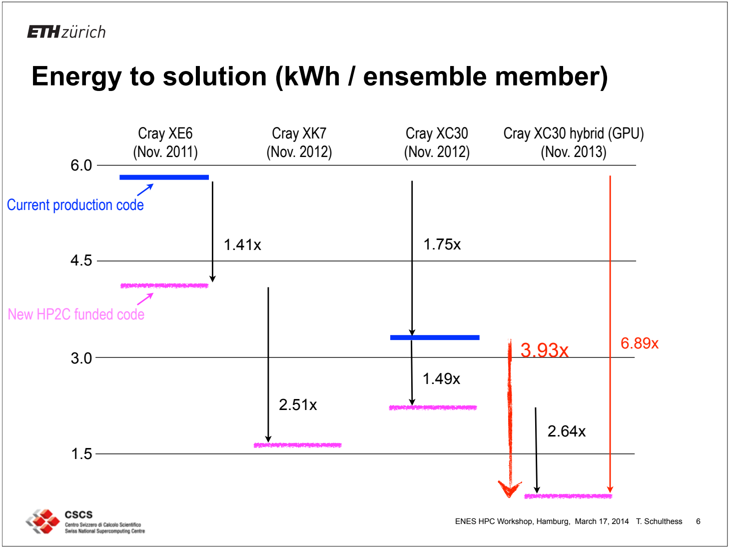


Courtesy George Mozdzynski, ECMWF.

COSMO: NWP production code using GPUs



COSMO: energy to solution

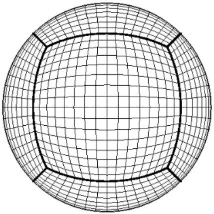


Summary of results in the jungle and zoo

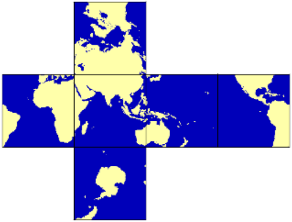
- Billion-way concurrency still a daunting challenge for everyone: no magic bullets anywhere to be found. ECMWF's PGAS approach is unique and interesting, at least one production GPU model.
- GPU/MIC based systems show nominal $\sim$10 increase in flops/socket, but actual performance about 1-2X (thus percent of peak drops from $\sim$10% to $\sim$1%)
- Software investment paid back in power savings (Schulthess).
- More threading 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 $\sim$10 pJ to main memory; $\sim$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.

Outline

Cubed-sphere, Gnomonic Projection



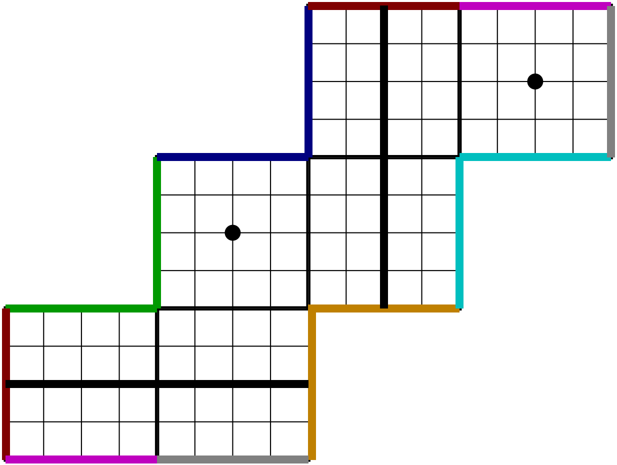
- True equal distance at the 12 edges of the cube
- All coordinate lines are great circles
- Coordinates are continuous at the edges; but derivatives are discontinuous



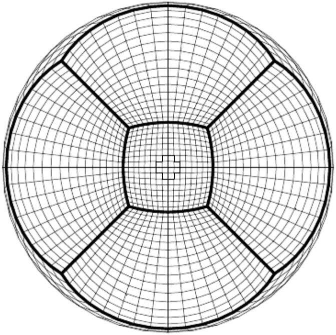
Putman and Lin, *J. Comp. Phys.* 2007.

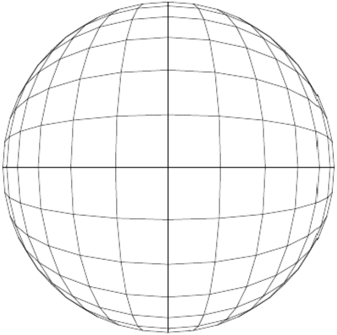
FMS index space representation of the cubed sphere

- Orientation changes (e.g $u \to -v, v \to u$)
- This is a C4 grid (C48 $\sim$ 200 km resolution; C2880 $\sim$ 3 km resolution)
- Typical pace of a coupled model: 10 y/d at C48; 3 y/d at C180.



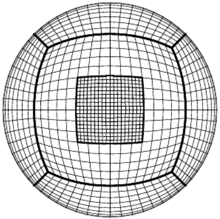
Stretched grids



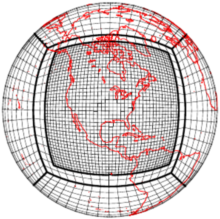


- Opposing face gets very coarse
- Discontinuities in slope
- Scale-aware parameterizations required

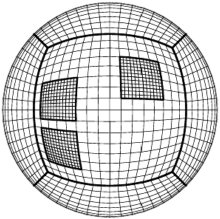
Nested grids



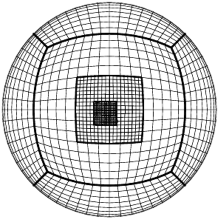
3:1 nested grid



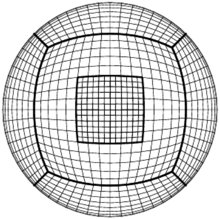
Large nest for RCMs



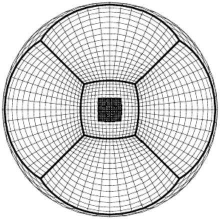
Multiple nests



Telescoping nests

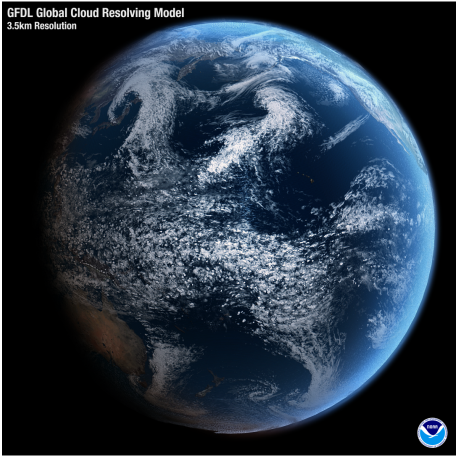


2:1 nested grid



Nest in stretched grid

C2560: 3.5 km resolution global cloud-resolving model



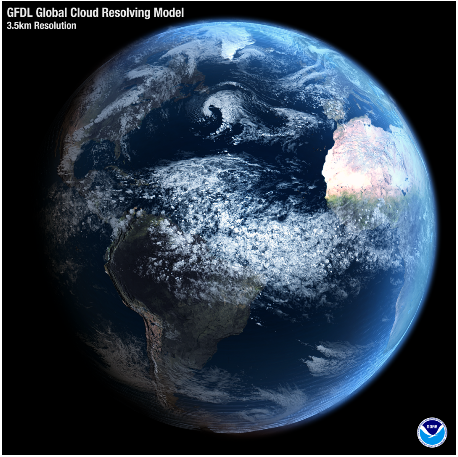
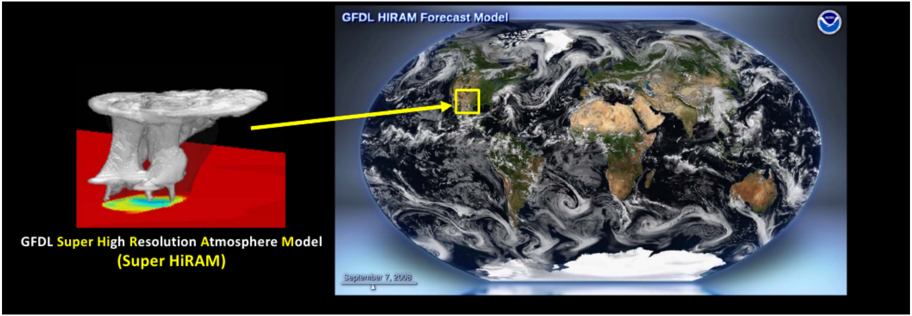


Figure courtesy S-J Lin and Chris Kerr, NOAA/GFDL.

Towards global cloud-resolving models



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

The NGGPS Effort

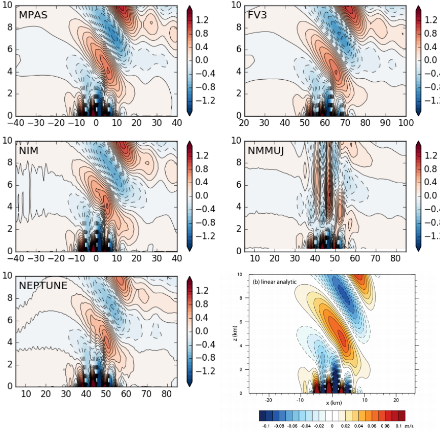
- NGGPS: Next-Generation Global Prediction System
- HIWPP: High-Intensity Weather Prediction Program

NGGPS and HIWPP launched a program to select a dynamical core for the next-generation forecast model (target: 3 km non-hydrostatic in 10 years). Selected dycores will undergo a substantial re-engineering effort for novel architectures.

- Scaling tests
- Idealized baroclinic wave test with embedded fronts (DCMIP 4.1)
- non-hydrostatic orographic mountain waves on reduced-radius sphere, no rotation
- idealized supercell thunderstorm on reduced-radius sphere, no rotation

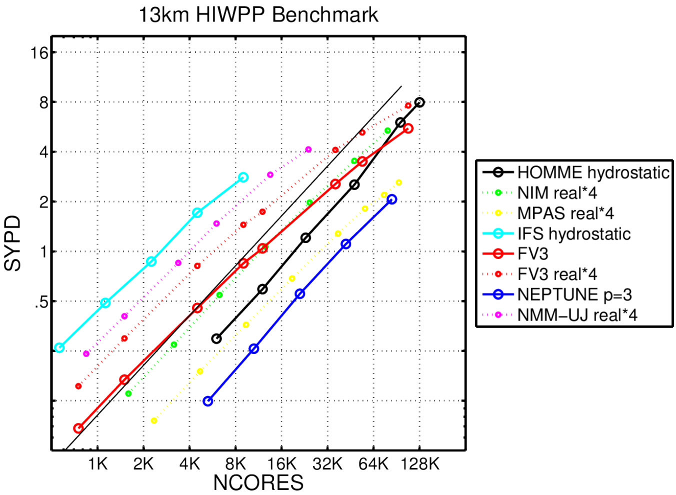
`http://www.nws.noaa.gov/ost/nggps/dycoretesting.html`

NGGPS Mountain Wave test case



`http://www.nws.noaa.gov/ost/nggps/dycoretesting.html`

NGGPS Scaling Study

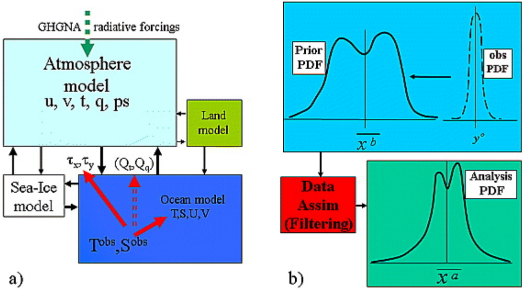


http://www.nws.noaa.gov/ost/nggps/dycoretesting.html

Outline

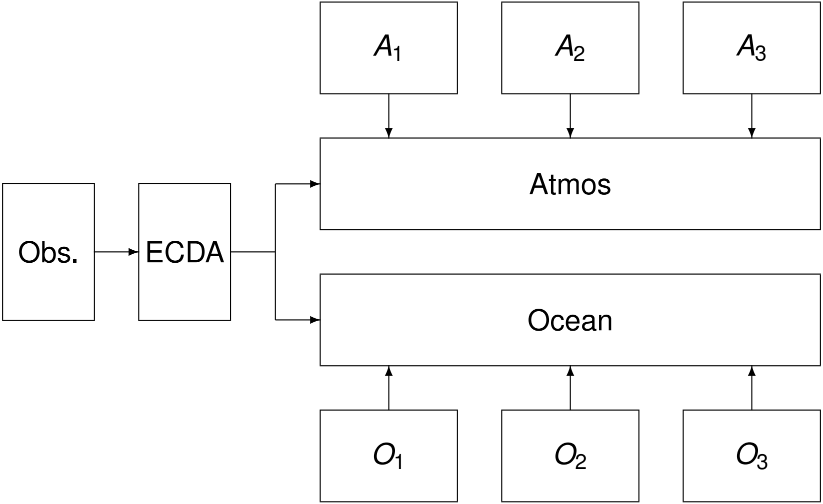
Data assimilation

Zhang - 2008JC005261



Data assimilation uses ensembles to find likely model trajectory taking into account model error and observational error. (Figure courtesy Zhang et al 2008).

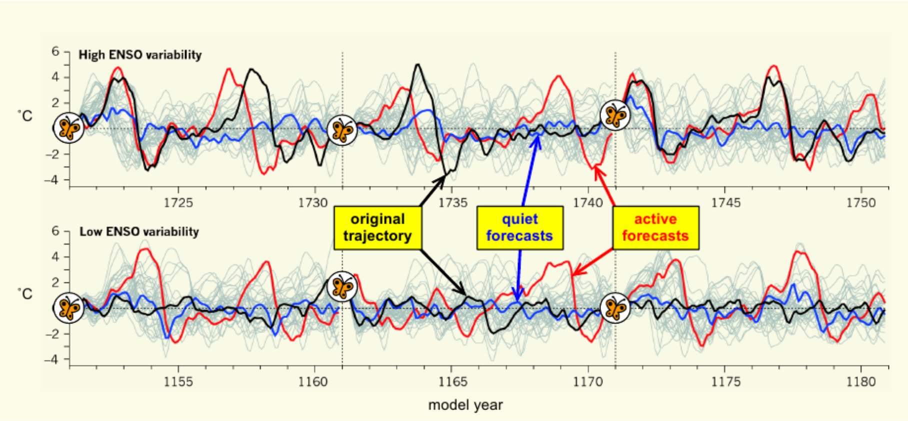
Ensemble Coupled Data Assimilation (ECDA)



Components (“instances”) execute in parallel.

ENSO modulation: is it decadally predictable?

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

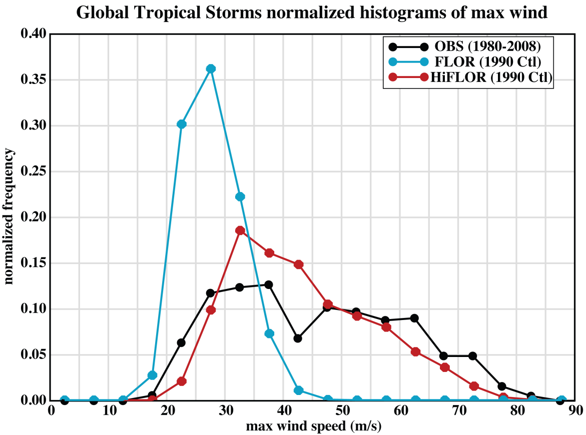


(External forcings **held fixed** at 1860 values.)

Wittenberg et al. (J. Climate, 2014)

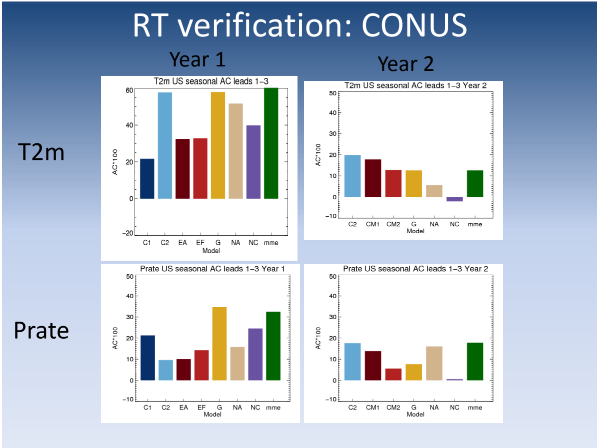
Effects of the proverbial "flap of a butterfly's wing"...

"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.

GFDL formed a new MDT for 2013-2016

In the 2013-2016 time frame, design and develop GFDL's best attempt at a climate model suitable for

- projection of climate change up to several hundred years into the future,
- attribution of climate change over the past century,
- prediction on seasonal to decadal time scales

keeping in mind the needs for improved regional climate information and assessments of diverse climate impacts.
The model will be capable of running from emissions in regard to both the carbon cycle and aerosols.
Courtesy Isaac Held, MDT Lead.

Target model configurations for CM4/ESM4

- 50 km atmosphere (C192) and 0.25° ocean (MOM6)

Determined by

- Lab experience regarding resources needed to develop and utilize a model for centennial-scale climate projections: at least 3-5 years/day throughput on no more than 1/8 of computational resource
- the GAEA computational resource

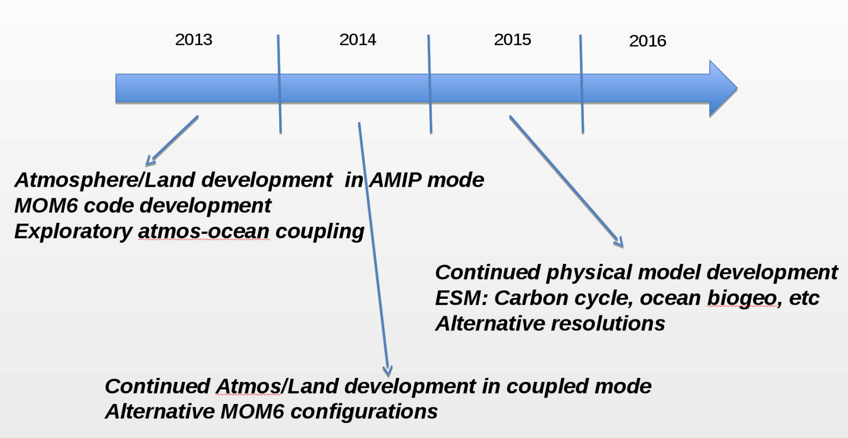
Increases in hardware resources and significant software development would allow us to redefine this trunk model towards higher resolution and/or greater comprehensiveness, e.g. full eddy-resolving ocean resolution; more complete stratosphere/troposphere chemistry module Courtesy Isaac Held.

Scientific and software challenges

- Oceanic mesoscale eddies Can we make a 0.25° degree model look like an eddy-resolving model?
- Aerosol/cloud interactions How do we best combine bottom-up (process-oriented) perspective and top-down constraints provided by 20th century observations?
- Atmospheric boundary layer/low cloud feedbacks Are we in a position to incorporate a dramatically new type of boundary layer/shallow convection module similar to CLUBB?
- Software Can we find more concurrency to improve wall clock performance so that we can increase complexity/resolution relevant to MDT goals

Courtesy Isaac Held.

CM4 Timeline

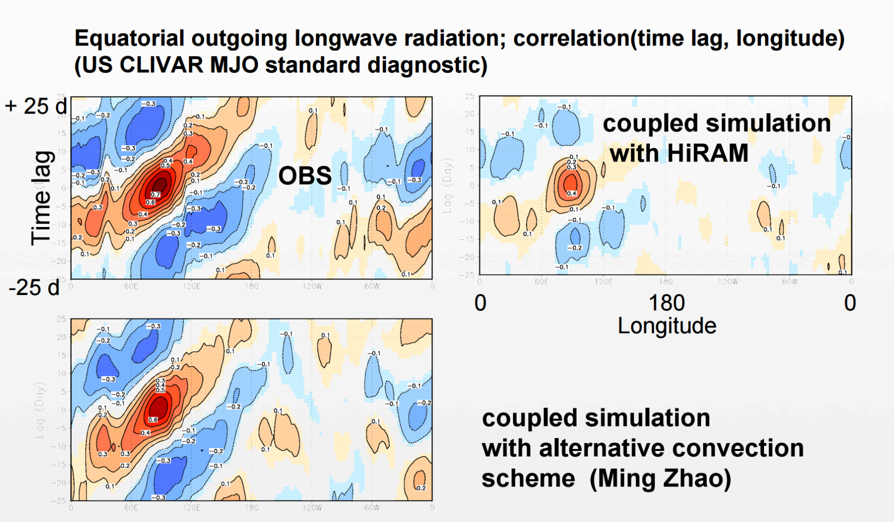


Courtesy Isaac Held.

Current CM4 Status

- About 6 months behind timeline!
- Testing C96L48 (full-chem, $\sim$100 species) and C192L32 (fast-chem, $\sim$20 species) atmospheric models (also C96L32 for rapid test cycle
- boundary layer and convection schemes, gravity-wave drag, etc still under testing and tuning
- COBALT ($\sim$30 species) adds about 50% to the cost of the ocean model
- COBALT timestep decoupled from main loop (much faster!)
- alternate 1° version of MOM6 being developed for predictability research, 0.5° for fast climate model.
- $\sim$5-10,000 core-hours per simulated year (CHSY) target models for CMIP6

New metrics for evaluation of models: MJO



New metrics for evaluation of models: TC climatology

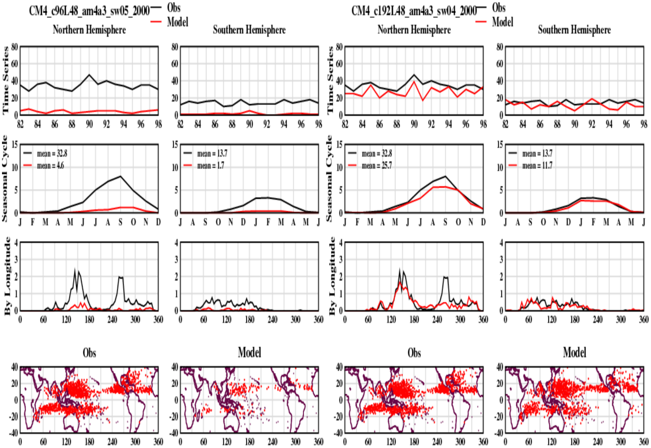
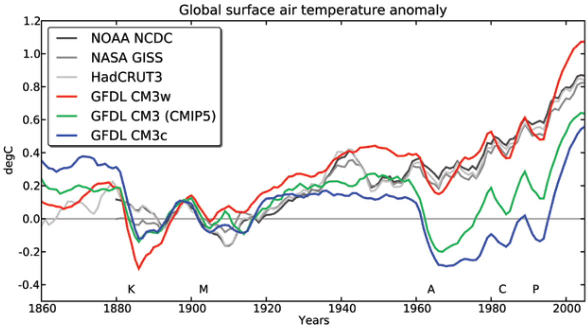


Figure courtesy John Dunne.

Model tuning: Process fidelity vs model bias

Tuning reduces model bias without violating process fidelity (but poses a problem for validation).

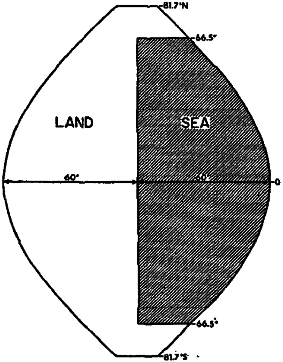


From Golaz et al 2012.

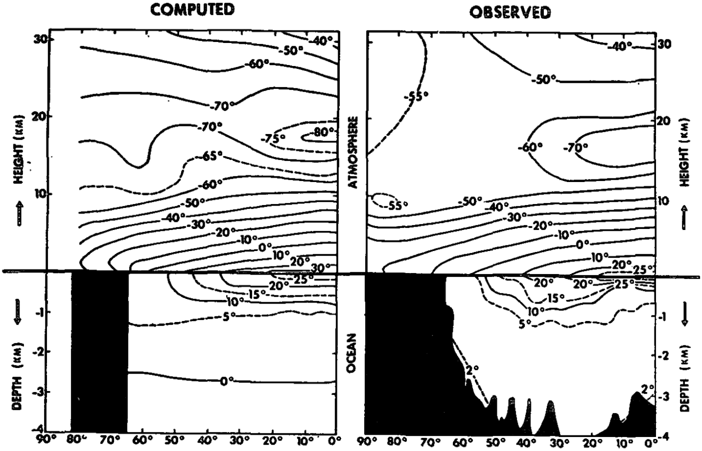
Outline

Manabe and Bryan (1969)

- Recognized as a “milestone in scientific computing”, Nature (2006).
- Sector model of 120°
- 1 atmospheric year coupled to 100 ocean years
- 1200h for 1 simulated year (0.02 SYPD) on Univac 1108



Manabe and Bryan 1969



First description of tuning and equilibration of coupled models.

Manabe and Wetherald 1975

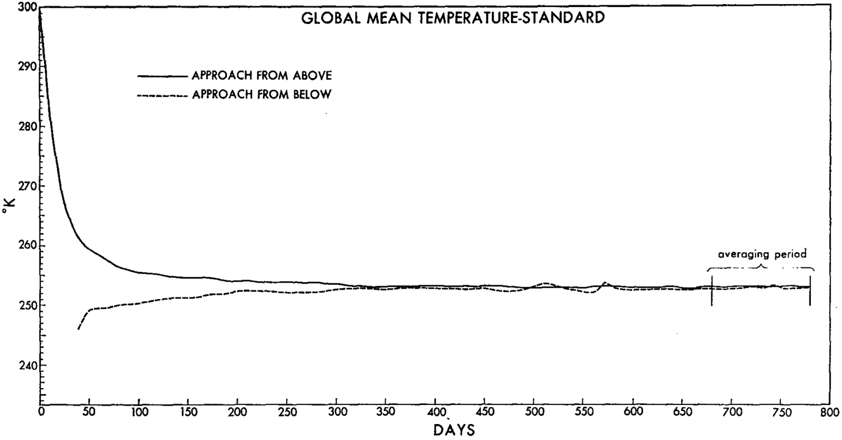


Fig 3 from Manabe and Wetherald (1975), a foundational document of climate modeling. “Equilibrium climate sensitivity” (ECS) introduced.

Atmospheric response to doubled CO_2

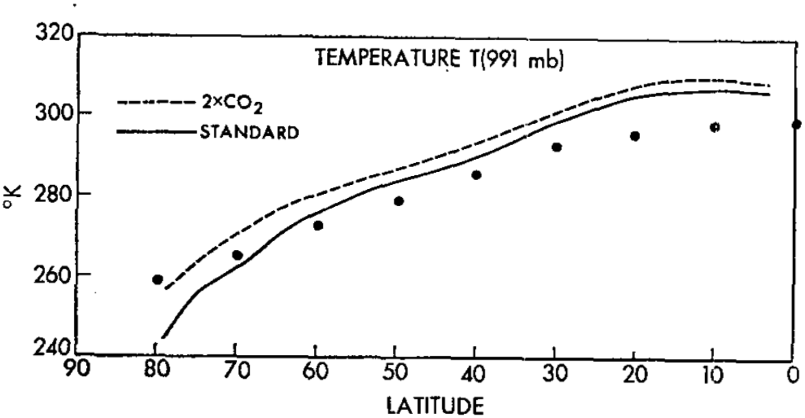


Fig 5 from Manabe and Wetherald (1975), equilibrium response to doubled CO_2.

Multi-model ensembles for climate projection

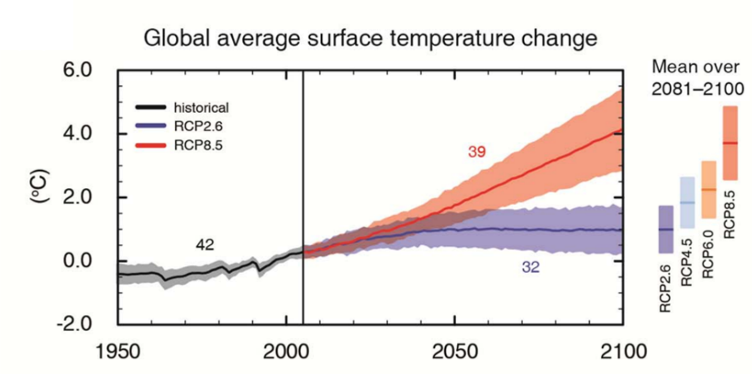
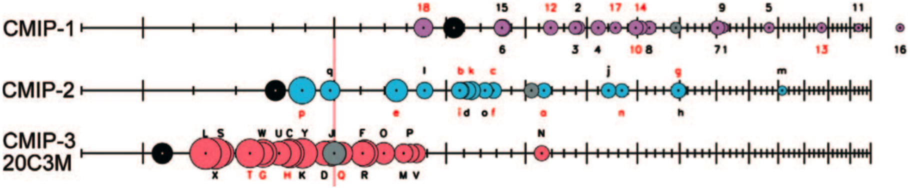


Figure SPM.7 from the IPCC AR5 Report. 20th century warming cannot be explained without greenhouse gas forcings.

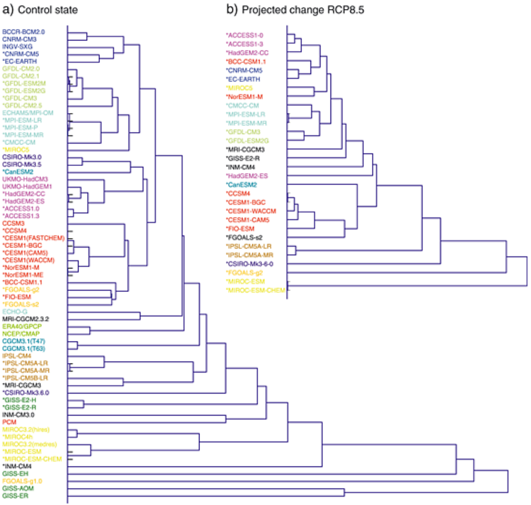
Multi-model ensembles to overcome "structural uncertainty"



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.

Genealogy of climate models



There is a close link between “genetic distance” and “phenotypic distance” across climate models (Fig. 1 from Knutti et al, GRL, 2013).

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.

Outline

Summary

- GFDL Strategic Plan: process studies; development of comprehensive models; climate extremes; experimental prediction; downstream science.
- Experimental seasonal to decadal prediction, including high-resolution fully coupled ensemble Kalman filter for data assimilation
- Continued development of extremely high-resolution atmosphere models using state of the art dynamical core
- Unification of ocean model development through MOM5 and MOM6 (incorporates capabilities from GOLD model into MOM, incorporates results of Climate Process Teams)
- Development of next generation climate model(s) CM4: convergence of multiple model branches into a few "trunk" models, through a Model Development Team led by Isaac Held.
- Increased integration of NOAA modeling across climate research and extended-range forecasting.

Challenges for the next generation of FMS

- Can we have high-level programming models or frameworks to take advantages of new approaches to parallelism? What are the right abstractions?
- Can component-level parallelism via framework superstructure be pushed to $\mathcal{O}(10)$?
- Can we approach models as experimental biological systems? (single organism or "cell line" not exactly reproducible; only the ensemble is.)
- How do we analyze and understand performance on a "sea of functional units" (Kathy Yelick's phrase)?
- How do we support big and diverse user base (including NCEP, IITM, INPE/CPTEC, ...)?