Coarse-grained component concurrency in Earth System modeling: parallelizing atmospheric radiative transfer in the GFDL AM3 model using the Flexible Modeling System coupling framework

V. Balaji[1], R. Benson[2], B. Wyman[2], and I. Held[2]

[1]Princeton University, Cooperative Institute of Climate Science
[2]NOAA/Geophysical Fluid Dynamics Laboratory

Correspondence to: V. Balaji (balaji@princeton.edu)

Abstract. Climate models represent a large variety of processes on a variety of time and space scales, a canonical example of multi-physics multi-scale modeling. Current hardware trends, such as Graphical Processing Units (GPUs) and and Many-Integrated Core chips (MICs), are based on, at best, marginal increases in clock speed, coupled with vast increases in concurrency, particularly at the fine grain. Multi-physics codes face particular challenges in achieving fine-grained concurrency, as different physics and dynamics components have different computational profiles, and universal solutions are hard to come by.

We propose here one approach for multi-physics codes. These codes are typically structured as *components* interacting via software frameworks. The component structure of a typical Earth system model consists of a hierarchical and recursive tree of components, each representing a different climate process or dynamical system. This recursive structure generally encompasses a modest level of concurrency at the highest level (e.g atmosphere and ocean on different processor sets) with serial organization underneath.

We propose to extend concurrency much further by running more and more lower- and higher-level components in parallel with each other. Each component can further be parallelized on the fine grain, potentially offering a major increase in scalability of Earth system models.

We present here first results from this approach, called Coarse-grained Component Concurrency, or CCC. Within the GFDL Flexible Modeling System, the atmospheric radiative transfer component has been configured to run in parallel with a composite component consisting of every other atmospheric component, including the atmospheric dynamics and all other atmospheric physics components. We will explore the algorithmic challenges involved in such an approach, and present results from such simulations. Plans to achieve even greater levels of coarse-grained concurrency by extending this approach within other components such as the ocean, will be discussed.

1 Introduction

Climate and weather modeling have historically been among the most computationally demanding domains using high-performance computing. Its history parallels that of modern computing itself, starting with experiments on the ENIAC (Platzman, 1979) and continuing through several changes in supercomputing architecture, including the vector and parallel eras.

The transition from vector to parallel computing was "disruptive", to use a currently popular term. The computing industry itself was transitioning from being primarily responsive to military and scientific needs, to being dominated by a mass market demanding cheap and ubiquitous access to computing. This gradually led to demise of specialized computational machines and the high end of the market also being dominated by clusters built out of mass-market commodity parts ~~.~~ (Ridge et al., 1997; Sterling, 2002) .

The community weathered the challenge well, without significant loss of pace of scientific advance. More narrowly stated, Earth System Models (ESMs) continued to demonstrate continual increases in both resolution and complexity across the vector/parallel transition. For example, the typical resolution of climate models used for the IPCC assessments, and their complexity (the number of feedbacks and phenomena simulated), exhibits a steady increase from the 1990 First Assessment Report (known as FAR) to ~~AR5 (?, see e.g the iconic Figure 1.4 from the Summary for Policymakers) .~~ AR4 (Solomon, 2007, see e.g the iconic *Fi*

A second disruption is upon us in the current era. Current computing technologies are based on increased concurrency of arithmetic and logic, while the speed of ~~arithmetic and logic~~ computation and memory access itself has stalled. This is driven by many technological constraints, not least of which is the energy budget of computing (Cumming et al., 2014; Charles et al., 2015; Kogge et

These massive increases in concurrency pose challenges for high-performance computing (HPC) applications: an era where existing applications would run faster with little or no effort, simply by upgrading to newer hardware, has ended. Substantial recoding and re-architecture of applications is needed. This poses particular challenges to applications such as climate modeling, where we must simulate many interacting subsystems. The state of play of climate computing in the face of these challenges, is surveyed in Balaji (2015) and references therein~~.~~ : for the current generation of technology, the gains to be had seem modest, and the effort of recoding immense. Whether we will continue to demonstrate continued increases in resolution and complexity through this transition remains to be seen.

ESMs[3] are canonical multi-physics codes, with many interacting components, each often built by independent teams of specialists. The coupling of these components, while respecting algorithmic constraints on conservation and numerical accuracy, is a scientific and technological challenge unto itself. Within each component of an ESM, code is parallelized using multiple techniques, including distributed and shared memory parallelism, as well as vector constructs.

Multi-physics codes are particularly challenged by the coming trends in HPC architecture. These codes typically involve many physical-chemical-biological variables (complexity) and associated process representations in code. Computational load is evenly distributed across many components, ~~and there is constant churn of operations and operands on the processor.~~

[3] Note that we are using the term "ESM" generically to denote any level in the hierarchy of complexity of weather and climate models: ranging from single-component models, e.g an atmospheric general circulation model, to models that include coupling with the ocean and land, biospheres, an interactive carbon cycle, and so on. See Figure 2.

each embodying different physics: there are no performance "hotspots". This also means that new operatorss and operands – embodied in physics subroutines and associated variables- are constantly being transferred to and from memory, and locality and reuse hard to achieve. This is particularly unsuited to the novel architectures currently on the horizon. These include graphical processing units (GPUs) which can concurrently process $\mathcal{O}(100)$ data streams following the same instructions sequence, and the Many-Integrated Core (MIC) architecture, which allows many ($\mathcal{O}(100)$) execution threads to access the same memory. These hardware trends have made the cost of data movement prohibitive relative to computing itself, thus strongly favoring codes where both instructions and data have a high rate of reuse and computational intensity (ratio of floating-point operations to memory operations). Algorithms that exhibit *fine-grained concurrency*, where multiple computationally intensive and concurrent data streams follow the same instruction sequence, are best adapted to the emerging architectures of this decade.

The next computational landmark at this juncture is the "exascale", $\mathcal{O}(10^{18})$ operations per second. Given the earlier discussion on the stalling of Moore's Law, the rate of atomic arithmetic operations is still $\mathcal{O}(10^{9})$ per second, thus requiring us to achieve $\mathcal{O}(10^{9})$ concurrency. While continuing to extend physical and dynamical algorithms toward the fine-grained concurrency of the coming era, we believe multi-physics codes must also attempt to share the available concurrency across many physical components, in order to best exploit these new systems. Of the many factors of 10 increase in performance needed to get to the ~~promised land of "exascalecomputing",~~ exascale, we believe at least one can come from component organization. We propose here a major architectural change in the construction of coupled models, such as ESMs. We demonstrate here a possible approach to extending the current rather modest amount of concurrency among ESM components (typically 2-4 top-level realms such as atmosphere, ocean, and land) to a more massive increase in *coarse-grained component concurrency (CCC)*.

In this study, we examine the radiation component (which computes radiative transfer in the atmosphere in response to dynamically evolving concentrations of radiatively active chemical species) of the GFDL Flexible Modeling System (FMS). This is a relatively expensive component of the atmospheric physics, and is usually run at a much coarser timestep than the rest of the atmosphere, purely for expediency rather than any physical justification (temporal subsampling). Other approaches to reducing the computational burden of radiative transfer include subsampling in the spectral domain (Pincus and Stevens, 2009; Bozzo et al., 2014) or in the spatial domain as well as the temporal (Morcrette, 2000; Morcrette et al., 2008) . Some of these methods have been shown to be effective over short timescales (e.g numerical weather prediction and medium-range forecasting) but contribute to model bias over climate timescales. Adaptive methods that tune the subsampling by examining the degree of spatial and temporal correlation in model fields have also been proposed (Manners et al., 2009) .

We focus here on temporal subsampling. This purely expedient choice of timestep has been shown by Pauluis and Emanuel (2004) to be a potential source of instability and bias in radiating atmospheres. Xu and Randall (1995) have also shown that this problem gets considerably worse as the resolution of models increases. A useful way to think about this is that using different timesteps for the radiation component vis-à-vis the rest of the physics creates a discrepancy between the cloud field and the "cloud shadow field" seen by the radiation component, which can lead to numerical issues. Our method permits us to reduce the timestep to match the rest of the atmosphere, *with the same time to solution*, at a modest computational cost in terms of allocated processors. This method does not rule out subsampling along other dimensions (spatial or spectral), which may be

superimposed as well in future developments. The effects of subsampling are not fully understood yet, and further study is needed to understand how results converge as various forms of subsampling are eliminated. That said, subsampling is clearly a matter of expediency and reducing computational expense: there is no case at all to be made that it is in any way numerically or physically superior to the alternative. .

The structure of the paper is as follows. In Section 2 we briefly review current approaches to parallelism in ESMs, particularly in the coupling framework. In Section 3 we describe our approach to coarse-grained concurrency, how it is achieved without increasing data movement. In Section 4 we show results from standard AMIP (The Atmospheric Model Intercomparison Project: Gates, 1992) simulations using the CCC approach. The associated computational results show decreased time to solution for concurrent versus serial approaches in otherwise identical physical formulations, and the ability to run with a much smaller radiation timestep without increasing the time to solution. Finally, in Section 5 we discuss plans and prospects for extending this approach further within FMS, and its potential application on novel architectures.

2 Concurrency in Earth System Models

Weather and climate modeling have always been in the innovative vanguard of computing, dating all the way back to the origins of modern computing in John von Neumann's pioneering studies (Dahan-Dalmedico, 2001). The ability to apply instruction sequences to multiple data streams – concurrency – has long been a cornerstone of performance engineering. The pioneering vector processors of Seymour Cray's era in the late 1970s allowed a data stream to flow through a hardware innovation known as *vector registers*, which allowed the same instruction sequences to apply to each succeeding element in the data stream, known as SIMD (single-instruction multiple-data). Over time, vectors grew to support extremely complex programming sequences, evolving into single-program, multiple-data, or SPMD.

In the 1980s, machines such as the Cray X-MP (the MP stood for multi-processor) were introduced. Here for the first time *parallelism* appears at a very high-level, allowing the concurrent execution of multiple tasks, which were themselves SPMD vector programs. This was the first introduction of coarse-grained concurrency. This led to the development of the MPMD framework: multiple-program, multiple-data. Soon after, distributed computing, consisting of networked clusters of commodity computers, known as symmetric multi-processors (SMPs), began to dominate HPC, owing to the sheer advantage of the volume of the mass market.

To take advantage of distributed computing, new techniques of concurrency began to be developed, such as domain decomposition. Here the globally discretized representation of physical space in a model component is divided into domains and assigned to different processors. Data dependencies between domains are resolved through underlying communication protocols, of which the Message-Passing Interface MPI (Gropp et al., 1998) has become the *de facto* standard. The details of message-passing are often buried inside software frameworks (of which the GFDL Flexible Modeling System, described in Balaji (2012) is an early example), and this convenience led to the rapid adoption of distributed computing across a wide variety of applications. Within the distributed domains, further fine-grained concurrency is achieved between processors sharing

physical memory, with execution threads accessing the same memory locations, using protocols such as OpenMP (Chandra et al., 2001).

Climate computing has achieved widespread success in the distributed computing era. Most ESMs in the world today are MPMD applications using a hybrid MPI-OpenMP programming model. At the highest end, ESMs (or at least, individual components within been run on $\mathcal{O}(10^5)$ distributed processors and $\mathcal{O}(10)$ shared-memory execution threads, which places them among the most successful HPC applications in the world today (Balaji, 2015). Even higher counts are reported on some leadership machines, but these are more demonstrations than production runs for science (e.g Xue et al., 2014) .

2.1 Coupling algorithms in Earth System Models

There are diverse component architectures across Earth System Models (Alexander and Easterbrook, 2015), but they nonetheless share common features for the purposes of discussion of the coupling algorithms. Consider the simplest case, that of two components, called A and O (symbolizing atmosphere and ocean). Each has a dependency on the other at the boundary. When the components execute serially, the call sequence can be schematically represented as:

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

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

where $f()$ and $g()$ nominally represent the feedbacks from the other component, and the superscript represents a discrete timestep. Note that in the second step, O is able to access the updated state at A^{t+1}. This is thus formally equivalent to Euler forward-backward time integration, or Matsuno timestepping, as described in standard textbooks on numerical methods (e.g Durran, 1999).

In a parallel computing framework, now assume the components are executing concurrently. (Figure 1 shows the comparison of serial and concurrent methods in parallel execution.) In this case, O only has access to the *lagged* state A^t.

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

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

Thus, the results will not be identical to the serial case. Furthermore, while we cannot undertake a formal stability analysis without knowing the forms of f and g, this coupling algorithm is akin to the Euler forward method, which unlike Matsuno's method is formally *unconditionally unstable*. Nevertheless, this parallel coupling sequence is widely, perhaps universally, used in today's atmosphere-ocean general circulation models (AOGCMs). This is because for the particular application used here, that of modeling weather and climate, we find that the system as a whole has many physical sources of stability.[4] Radiative processes are themselves a source of damping of thermal instability, and we also note that within each component there are

[4] We are familiar with things that work in theory, but not in practice... this is something that works in practice but not in theory! This is a good example of the opportunistic nature of performance engineering.

internal processes and feedbacks which are often computed using implicit methods, and other methods aimed at reducing instability. This is nonetheless a reason for caution, and in Section 5 we will revisit this issue in the context of future work.

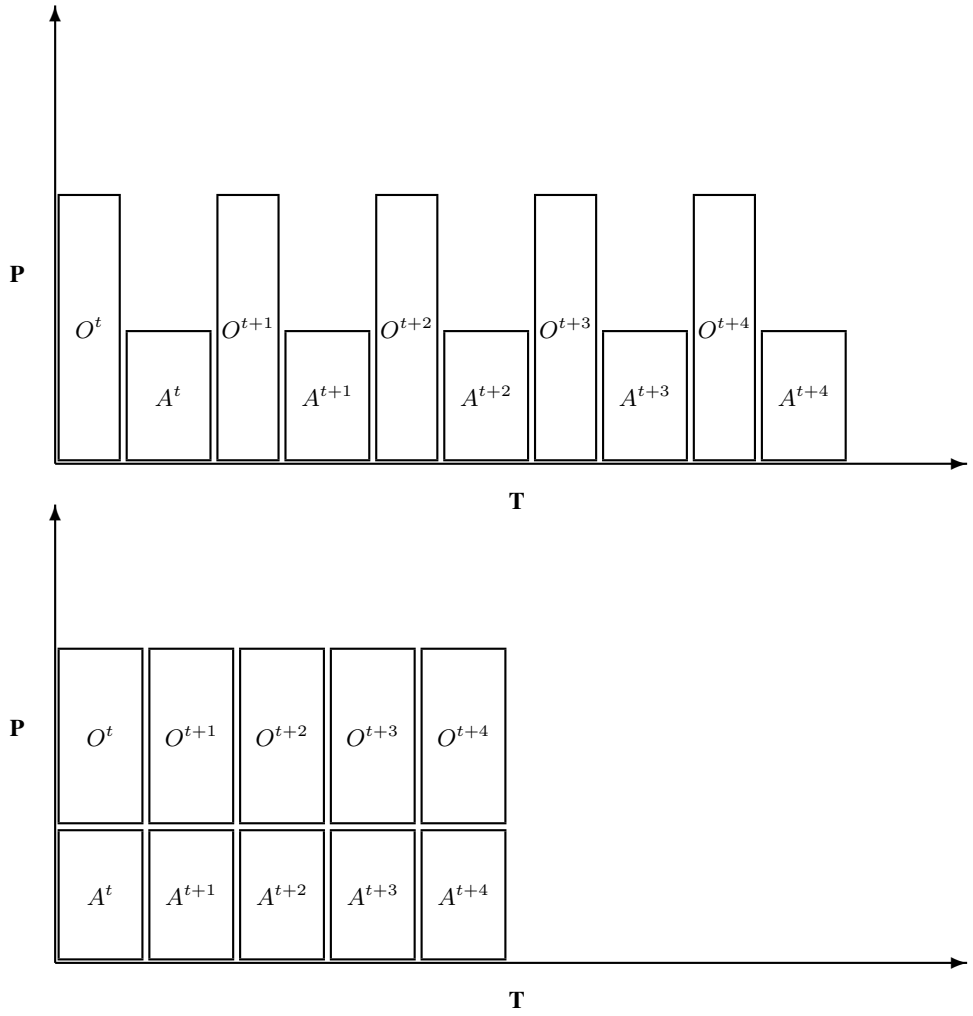

Figure 1. Serial and concurrent coupling sequences, with time on the X-axis and processors on the Y-axis. In the serial case, both components may not scale to the same processor count, leaving some processors idle. Note that in the concurrent coupling sequence below, O^{t+1} only has access to the lagged state A^t.

This discussion has introduced the notions underlying serial and concurrent coupling in the context of two components A and O. An actual ESM has many other components, such as land and sea ice. Components themselves are hierarchically organized. An atmosphere model can be organized into a "dynamics" (solutions of fluid flow at the resolved scale) and "physics" components (subgrid scale flow, and other thermodynamic and physical-chemical processes, including those associated with clouds and subgridscale convection, and the planetary boundary layer). Similarly the land component can be divided into a

hydrology and a biosphere component, and the ocean into dynamics, radiative transfer, biogeochemistry, and marine ecosystems. A notional architecture of an ESM is shown in Figure 2. Different ESMs around the world embody these differently in code; this figure is not intended to define the software structure of all ESMs, which tend to be quite diverse (Alexander and Easterbrook, 2015).

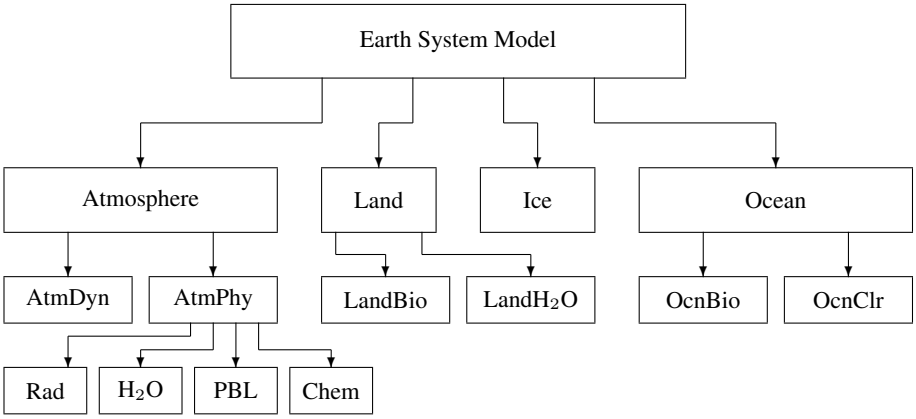

Figure 2. Notional architecture of an Earth System Model, with components embodying different aspects of the climate system, hierarchically organized. Models on a hierarchy of complexity ranging from single-component (e.g atmosphere-only) models to full-scale coupled models with an interactive biosphere, are often constructed out of a palette of components within a single modeling system.

How the notional architecture of Figure 2 gets translated into a parallel coupled ESM code is quite problem-specific. As the science evolves and computing power grows, the boundary of what is resolved and unresolved changes. Also, models grow in sophistication in terms of the number of processes and feedbacks that are included.

For the purposes of this study, we describe the actual code architecture of the GFDL Flexible Modeling System (FMS). The atmosphere and ocean components are set up to run in parallel in distributed memory, communicating on the slow coupling timestep $\Delta t_{\rm cpld}$, on the order of $\Delta t_{\rm cpld}$ = 3600 sec for its flagship application, decadal-centennial climate change. Within the slow coupling loop, the atmosphere communicates on a fast coupling timestep $\Delta t_{\rm atm}$ with a typical value of 1200 sec, set by the constraints of atmospheric numerical and physical stability.

As the land and ocean surfaces have small heat capacity, reacting essentially instantaneously to changes in atmospheric weather, stability requires an implicit coupling cycle. The implicit coupling algorithm requires an down-up sweep through the atmosphere and planetary (land and ocean) surface systems, for reasons detailed in Balaji et al. (2006). The parallel coupling architecture of FMS is shown in Figure 3.

The "atmosphere-up" step is quite lightweight, including adjustments to the atmospheric state imposed by moist physics, and completing the up-sweep of a tridiagonal solver for implicit coupling of temperature and other tracers, as described in Balaji et al. (2006). The bulk of the atmospheric physics computational load resides in the "atmosphere-down" step.

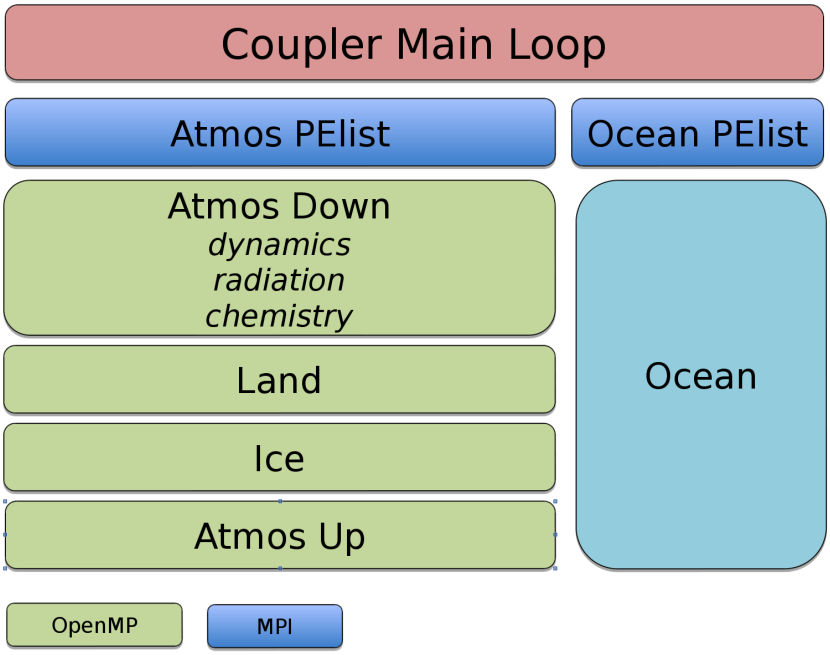

Figure 3. FMS parallel coupling architecture in processor-time space, with PEs across, and time increasing downward. ~~*PElists*~~ Components have different horizontal and vertical extents to indicate the degree of parallelism and time of execution, though these extents are notional and not to be interpreted as drawn to scale. Within a single executable for the entire coupled system, the atmosphere and ocean components run concurrently in distributed memory (MPI, indicated in deep blue). Within the atmosphere stack, ~~and stacked~~ components execute serially, including tight coupling to the land and ice-ocean surface. ~~Components~~ The down-up sequence of implicit coupling is explained in Balaji et al. (2006). These components internally ~~use a hybrid distributed-shared parallel programming model~~shared-memory (OpenMP) coupling, indicated in light green. The ocean component at the present time is MPI-only, indicated in light blue.

The atmospheric radiation component is a particularly expensive component of atmospheric physics, which is why it was chosen as the target for increasing coupling concurrency in FMS. This component is described below.

2.2 The radiation component in FMS

The radiation component in FMS is one of the most expensive components within the atmospheric physics. It consists of shortwave and longwave radiation components. The shortwave component is based on Freidenreich and Ramaswamy (1999), where "line-by-line" (LBL) radiative transfer calculations have been grouped into pseudo-monochromatic bands, and shown in benchmark calculations to provide a similar response to the benchmark LBL results. The calculations have a strong dependency on the evolving (through advection, cloud processes, and chemistry) state of radiatively active species in the atmosphere, including atmospheric water vapor, CO_2, O_3, aerosols, and condensed water fields, in addition to the basic physical state variables. The

longwave radiation components (Schwarzkopf and Ramaswamy, 1999) similarly use approximations for computational efficiency, and also interact strongly with atmospheric ~~chemical species~~ liquid- and gas-phase chemical species, including water vapor and clouds. The species shared between atmospheric physics, chemistry, and radiation are referred to as *tracers*, a term applied to 3D model fields (in atmosphere or ocean) that are advected by the evolving dynamics, and participating in physics and chemistry processes at individual gridpoints.

Despite the simplifying approximations, the radiation component remains prohibitively expensive. As a result, this component is stepped forward at a slower rate than the rest of the atmospheric physics, with ~~= 10800 seconds~~$\Delta t_{\rm rad}$ = 10800 sec, or $9 * \Delta t_{\rm atm}$, as a typical value. The planetary surface albedo, whose time evolution is a function of solar zenith angle only, alone is stepped forward on the atmospheric timestep $\Delta t_{\rm atm}$. This means that at intermediate (non-radiation) atmospheric timesteps, the radiation is responding to a lagged state of atmospheric tracers, which may be as much as $\Delta t_{\rm rad} - \Delta t_{\rm atm}$ (~3~~hours~~ h) behind. ~~The use of the lagged state introduces numerical errors, as described in Pauluis and Emanuel (2004).~~

This timestep discrepancy is vexing, but most climate models around the world make a similar compromise, with a radiation timestep longer than the physics timestep. If the promise of massive increases in concurrency on future architectures is kept, a concurrent radiation component may offer a way forward. Simultaneously, we may be able to decrease the discrepancy between $\Delta t_{\rm rad}$ and $\Delta t_{\rm atm}$, and bring us toward more physical consistency between the radiative and physico-chemical atmospheric states (Pauluis and Emanuel, 2004; Xu and Randall, 1995).

3 Coarse-grained component concurrency

We describe now a method for casting the radiation code in FMS as a concurrent component. Concurrency between atmosphere and ocean component on the slow coupling timestep is achieved using distributed computing techniques, with the components running on separate processor sets or ~~*PElists* ("communicators" in MPI terminology).~~ PElists[5]. Coupling fields are transferred between atmosphere and ocean using the exchange grid (Balaji et al., 2006) and message passing. For the current study exploring the feasibility of concurrent radiation, cross-processor data transfers will impose a daunting cost, because of close time-dependency of radiation on evolving model fields.

We have implemented, instead, components using *shared-memory concurrency*. ~~The atmospheric component is already architected for shared-memory~~ Shared-memory parallelism using OpenMP is already implemented in the atmospheric component, where the columns within the distributed domain decomposition are further organized into *blocks*. Unlike the dynamics, the physics is organized entirely columnwise, and ~~Individual columns (~~individual columns – the k index in an (i, j, k) discretization ~~)~~ – have no cross-dependency in (i, j) and can execute on concurrent fine-grained threads. The blocks apply a certain number of columns per OpenMP thread. In theory, we could have (1,1)-sized blocks and a single column per thread, but the overheads associated with moving in and out of threaded regions of code must be amortized by having enough work per OpenMP thread instance.

[5] A PE or *processing element* is FMS terminology for a unit of distributed memory processing, sometimes called a *core*. A PElist is synonymous with a "communicator" in MPI terminology.

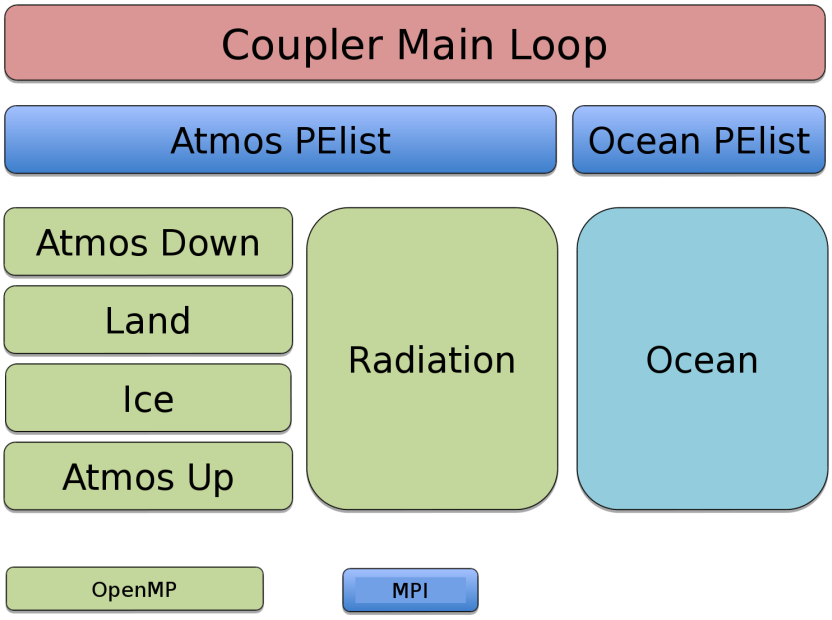

Figure 4. Concurrent radiation architecture. See Figure 3 for comparison, and explanation of legends.

In the concurrent radiation architecture, shown in Figure 4, the available threads in an OpenMP region are divided between the radiation component and the rest of the atmospheric physics and dynamics. This is achieved using nested OpenMP constructs. The atmospheric PElist starts up an OpenMP region with T threads at startup, when the PElist is created. In the nested call these are assigned to atmosphere and radiation (A and R threads, where $T = A + R$; A and R can be dynamically varied in the course of a run). A and R are chosen ~~to offer optimal load balance~~ for load balance (generally arrived at by trial and error) between the radiation code and the rest.

4 Results

4.1 Results from AMIP runs

The model utilized here is based on AM3, the atmosphere-land component of the GFDL CM3 model (Donner et al., 2011), a model with a relatively well-resolved stratosphere, with a horizontal resolution of approximately 100 km and 48 vertical levels. Here the original AM3 has been modified to include an experimental cumulus convection scheme, and a reduced chemistry representation including gas and aqueous-phase sulfate chemistry from prescribed emissions (Zhao et al., 2016). This model is forced using observed sea surface temperatures (SSTs) as a lower boundary condition over a 20-year period 1981-2000. The three experiments described here are:

1. the control run (CONTROL) using serial radiation with a radiative time step $\Delta t_{\rm rad}$ of 3 hours ($\Delta t_{\rm rad} = 9\Delta t_{\rm atm}$ where $\Delta t_{\rm atm}$ = 1200 s is the time scale on which the atmospheric state is updated;

2. serial radiation (SERIAL) using $\Delta t_{\rm rad} = \Delta t_{\rm atm}$ = 1200 s; and

3. concurrent radiation (CONCUR) also using $\Delta t_{\rm rad} = \Delta t_{\rm atm}$ = 1200 s. The difference between the SERIAL and CONCUR experiments shows the impact of concurrent coupling (the radiation sees the lagged atmospheric state), while the CONTROL and SERIAL experiments only differ in the radiative time step. We could of course attempt CONCUR while maintaining $\Delta t_{\rm rad}/\Delta t_{\rm atm} = 9$, but because of the lagged timestep, this is not recommended: the atmospheric and radiative states would be effectively 21600 s, or 6 h, out of synchrony.

All versions of the model utilize what we refer to as a solar interpolator. At the beginning of a radiative time step, the distribution of radiatively-active atmospheric constituents, such as water vapor and clouds, are input into computations of both shortwave and longwave radiative fluxes. When $\Delta t_{\rm rad}$ is longer than $\Delta t_{\rm atm}$ all solar fluxes are rescaled every $\Delta t_{\rm atm}$ by normalizing by the incident solar radiation using the zenith angle appropriate for ~~that~~ that atmospheric time step. Any sensitivity to the $\Delta t_{\rm rad}$ radiation time step is due to the fact that the radiatively-active constituents are held fixed for the duration of that time step and not due to neglected changes in the incoming solar flux ~~.~~ (Morcrette, 2000) .

We show here the effects of changing the radiation timestep on clouds and precipitation in the AMIP simulation. Figure 5 shows the annual mean precipitation bias for the three experiments with respect to the ~~GPCP~~ Global Precipitation Climatology Project (GPCP) v2.2 (Adler et al., 2003) climatology. Panel (b) shows the difference between model and observational precipitation climatology, while (c) and (d) show the model-model differences between the CONTROL run and the SERIAL and CONCUR climatologies. There is very little difference in the climatology of the model runs as the details ~~between these runs are remarkably similar.~~ of the differences between the runs show no glaring differences: all are plausible equivalent climatologies. A further examination of the globally average bias reveals the SERIAL and CONCUR cases are nearly the same, whereas the global bias for the CONTROL is slightly smaller (about 0.025 mm/day). Although this difference is small we feel it is robust and likely related to a sensitivity to the radiative time step discussed below.

Figure 6 shows the radiative energy balance at the top of the atmosphere between the absorbed shortwave and outgoing longwave radiation compared to CERES EBAF edition 2.8 satellite data (Loeb et al., 2009). Again the details of all the experiments are similar but a closer examination of the global mean biases show the SERIAL and CONCUR cases differ from the CONTROL by about +3.1 to 3.6 W/m^2. This magnitude of flux difference has a significant effect on a coupled atmosphere-ocean model. (Compare the change in flux due to a doubling of CO_2 concentrations, holding radiatively active atmospheric constituents fixed, of about 3.5 W/m^2). Nearly all of this difference is in the absorbed shortwave, most of which occurs over the oceans and tropical land areas. The source of this difference is primarily clouds and to a lesser extent water vapor, as determined by examining the clear-sky energy balance, a model diagnostic. The diurnal cycle of clouds and solar radiation appear to be the key factors in determining the sign and size of these responses. The diurnal peak in clouds over the oceans typically occurs close to sunrise, so there is a downward trend in cloudiness on average at the peak in incoming solar radiation.

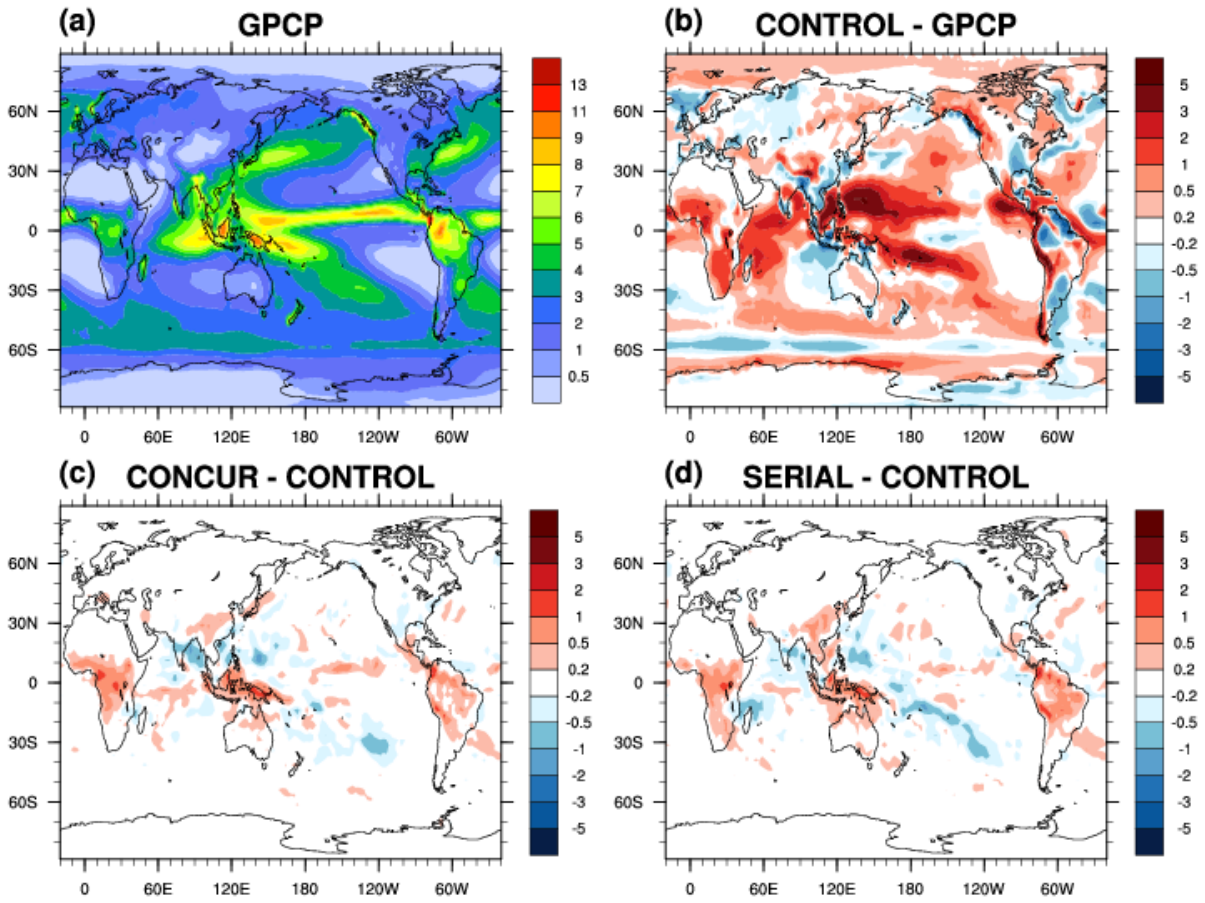

Figure 5. Comparison of model climatologies against GPCP precipitation. (a) shows the GPCP climatology, and (b) ~~,~~ the model climatological biases for the CONTROL run. Panels (c) and (d) show ~~model climatological biases for~~ model-model difference versus CONTROL ~~,~~ for CONCUR and SERIAL runs respectively, plotted on the same color scale as (b).

Therefore the CONTROL case sees more cloudiness over the longer radiative time step, therefore leading to more reflection by clouds and less absorption of shortwave radiation at the surface and in the atmosphere.

A point worth making is that the model was not "retuned" for the configuration where we update radiation calculations in synchrony with the physics – i.e setting $\Delta t_{\mathrm{rad}} = \Delta t_{\mathrm{atm}}$. The discrepancies are not fundamental and small enough that we believe them to be within the margins of the tuning process. To run this model in production in the CONCUR mode, we would undertake modest adjustments of some settings in order to arrive at a top-of-atmosphere radiative flux that is within observational bounds (see e.g Hourdin et al., 2016, for a description of the tuning process) .

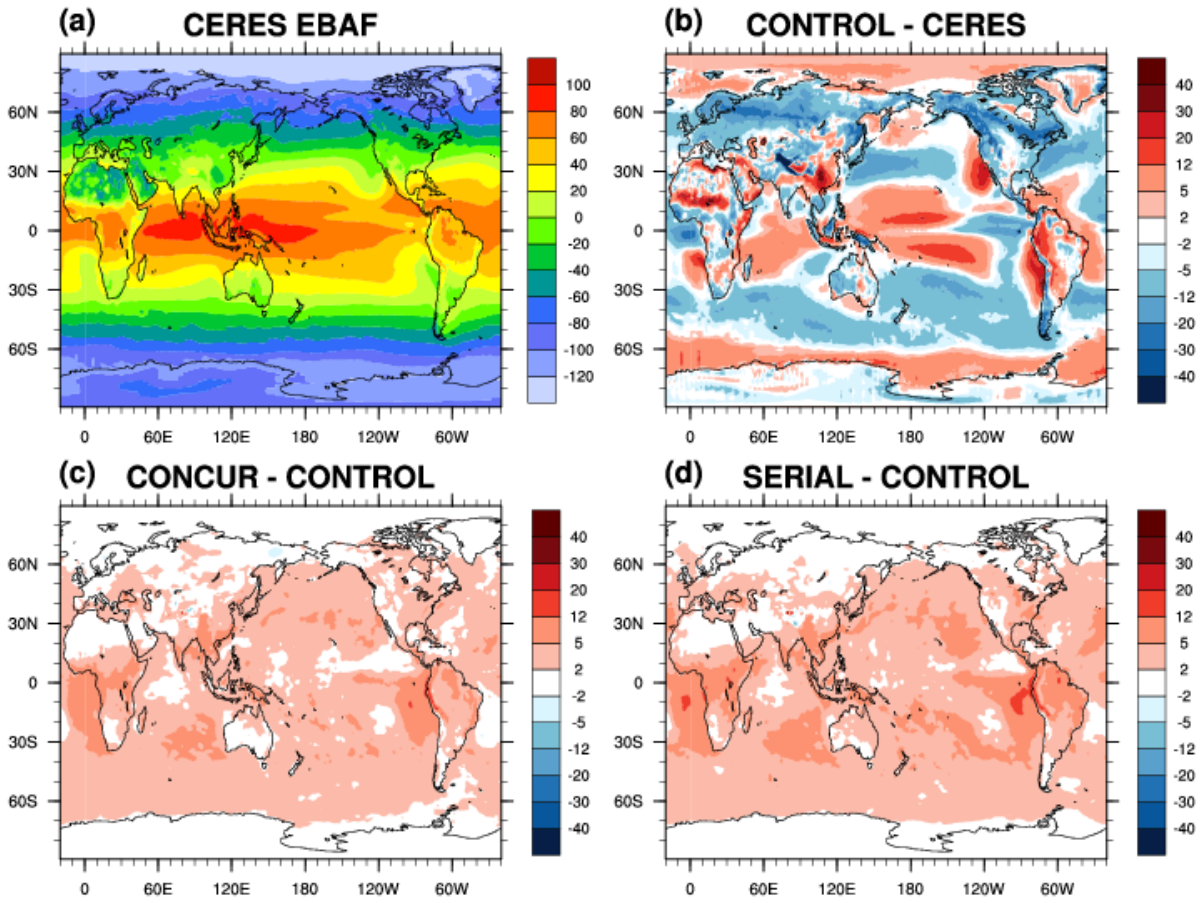

Figure 6. Comparison of model climatologies against CERES EBAF v2.8 climatological top-of-atmosphere radiation budget~~. (a) shows the CERES climatology~~, and (b) ~~,~~ Model climatological biases for the CONTROL run. Panels (c) and (d) show ~~model climatological biases for~~ model-model difference versus CONTROL ~~,~~ for CONCUR and SERIAL runs respectively, plotted on the same color scale as (b).

4.2 Scaling and performance results

Comparisons of the computational performance of the 3 configurations (CONTROL, SERIAL and CONCUR) were performed on the NOAA supercomputer Gaea. Recall that CONTROL is intrinsically computationally less expensive as ~~$\Delta t_{\mathrm{rad}} = 9 * \Delta t_{\mathrm{atm}}$~~ the $\Delta t_{\mathrm{rad}} = 9 * \Delta t_{\mathrm{atm}}$ setting implies that the radiation code is executed very seldom. As the results of Section 4.1 suggest that we should shorten Δt_{rad} if we can, the aim is now to recover the overall model throughput (measured in simulated years per day, or SYPD) of CONTROL using the CONCUR configuration with the shorter timestep $\Delta t_{\mathrm{rad}} = \Delta t_{\mathrm{atm}}$, but at a higher

processor count. The other measure in our comparison is that of the integrated processor-time computational resource request, measured in compute-hours per simulated year (CHSY). These are key measures of computational cost (time to solution, and resource consumption) used at modeling centers around the world (Balaji et al., 2016) .

Initial studies were performed on a machine configuration that used AMD Interlagos processors on Cray's Gemini high-speed interconnect. ~~All runs use~~ For any given PE count, we attempt different partitions between processors and threads (MPI and OpenMP) to arrive at the optimal processor/thread layout for ~~a given~~ that PE count. As Table 1 shows, CONTROL achieved 9.25~~SYPD~~ SYPD on 1728 PEs. The SERIAL configuration shows the relative cost of radiation to the rest of the model, as shortening $\Delta t_{\rm rad}$ from 10800 s to 1200 s, without changing the processor count, substantially raises the total cost of radiation computations within the code, bringing time to solution down to 5.28 SYPD. Running CONCUR on the same processor count increases this time to 5.9 SYPD. Increasing the processor count to ~~2600~~ 2592 brings us back to ~~9~~ 9.1 SYPD. Thus, one can achieve the goal of setting $\Delta t_{\rm rad} = \Delta t_{\rm atm}$ without loss in time to solution (SYPD), at a cost of ~~1.6X in resources (measured in compute-hours per simulated year, or CHSY).~~ 1.52X (the CHSY ratio of the two configurations) in resources. Thus decreasing the radiation timestep 9-fold has raised the computational cost by about 50%, indicating that the original cost of radiation in units of CHSY was about 5%. Models where this fraction is higher will derive an even more substantial benefit from the CCC approach. We believe that as computing architectures express more and more concurrency while clock speeds stall (see Section 5 below), this will be a key enabling technology for maintaining time to solution ~~.~~ while increasing parallelism. While these results are for an atmosphere-only model, they can be readily extended to other components in a more complex model. As noted below in Section 5, we are planning to extend the CCC approach to other components including atmospheric chemistry and ocean biogeochemistry.

Configuration	$\Delta t_{\rm rad}/\Delta t_{\rm atm}$	~~MPI/OMP~~ MPI*OMP	NPES	SYPD	CHSY
CONTROL	9	~~864/2~~ 864*2	1728	9.25	4483
SERIAL	1	~~864/2~~ 864*2	1728	5.28	7854
CONCUR	1	~~432/4~~ 432*4	1728	5.90	7029
CONCUR	1	~~648/4~~ 648*4	2592	9.10	6836

Table 1. Performance results from the various configurations discussed. MPI/OMP shows the assignment of MPI processes and OpenMP threads. In the CONCUR cases 2 threads each are assigned to atmosphere and radiation components. NPES is the total PE count (MPI*OMP). SYPD measures throughput in simulated years per day, and CHSY is the computation cost in processor-hours per simulated year (NPES*24/SYPD).

5 Summary and Conclusions

We are at a critical juncture in the evolution of high-performance computing, another "disruptive" moment. The era of decreasing time to solution at a fixed problem size, with little no effort, is coming to an end. This is due to the ending of the conventional meaning of Moore's Law (Chien and Karamcheti, 2013), and a future where hardware arithmetic and logic speeds stall, and

further increases in computing capacity are in the form of increased concurrency. This comes in the form of heterogeneous computing architectures, where co-processors or accelerators such as Graphical Processing Units (GPUs) provide SIMD concurrency; the other prevalent approach is in the Many-Integrated Core (MIC) architectures, with a vastly increased thread space, an order of magnitude higher than the $\mathcal{O}(10)$ thread parallelism achieved with today's hybrid MPI-OpenMP codes.

It is very likely that radical reimagining of ESM codes will be necessary for the coming novel architectures (Balaji, 2015). ~~Multi-physics~~ That survey of the current "state of play" in climate computing notes that multi-physics codes, which are fundamentally MPMD in nature, are particularly unsuited to these novel architectures. While individual components show some speedup, whole MPMD programs show only modest increases in performance (see e.g Govett et al., 2014; Iacono et al., 2014; Fuhrer et al., 2014; Ford et al., 2014). Other approaches, such as the use of "inexact" computing (Korkmaz et al., 2006; Düben et al., 2014) , are still in very early stages.

We have demonstrated a promising new approach for novel and heterogeneous architectures for MPMD codes such as Earth System models. It takes advantage of the component architecture of ESMs. While concurrency has been achieved at the very highest level of ESM architecture shown in Figure 2, the components are themselves MPMD within a hierarchical component architecture.

In the light of our discussion we propose a precise definition of a component as a unit of concurrency. For the purposes of the ~~CCC~~ coarse-grained concurrency (CCC) approach, a *component* may be defined as *one of many units in a multi-physics model, that is itself SIMD for the most part.* While the word has been loosely used earlier, this study has provided guidance on how we should think about components, and thus, this definition will be followed for the rest of the discussion in this section. Fine-grained parallelism approaches, such as those surveyed in Mittal and Vetter (2015), may be applied within a component as so defined, but are likely to fail above that level. A substantial increase in overall scalability of an ESM may be achieved if several components are run concurrently. We ~~have identified such components in contemporary ESMs.~~ are currently exploring CCC in several other computationally burdensome model components, including atmospheric chemistry and ocean biogeochemistry. It is clear however that this is not an universal solution: given the constraint that concurrent components can only "see" each others' time-lagged state, some components are too tightly coupled to be amenable to the CCC approach.

Furthermore, we have demonstrated a method where multiple components that share a large number of model fields can be run concurrently *in shared memory*. This avoids the necessity of message passing between components that need to synchronize on fine timescales.

We believe the CCC approach will afford very tangible benefits on heterogeneous architectures such as GPUs, and architectures with a wide ($\mathcal{O}$(100-1000)) thread space, such as MICs. In particular:

- Threading within a SIMD component has not been shown to scale beyond a rather modest thread count. By running multiple components within a single thread space, the thread count can be considerably increased.

- Even with additional work in improving the SIMD performance of components, it is clear that some components are better suited to SIMD architectures than others. In a heterogeneous system, with different hardware units, this method may permit different components to be scheduled on the hardware unit to which they are best suited. For example,

we could imagine some embarrassingly parallel components executing on a GPU while another, less suited to that architecture, executes on its host CPU.

There remain caveats to this approach. As shown in the discussion of Equation 3 above, the coupling of concurrent components might be formally unstable. We are exploring more advanced time-coupling algorithms, including 3-time-level schemes such as Adams-Bashforth (see Durran, 1999). Such algorithms have been successfully used within the atmospheric dynamical core of FMS, for two-way nesting. In this approach, the coarse- and fine-mesh components execute concurrently rather than serially as in conventional nesting approaches (Harris and Lin, 2013). We are also exploring a combination of the 3-time-level schemes with time-staggering of components, which no longer suffers from formal instability.

.We conclude that coarse-grained concurrency remains a very promising road to the future of Earth System modeling on novel, massively-concurrent HPC architectures.

6 Source code and data availability

Source code and data, including model output and performance data, associated with this study are freely available upon request.

Acknowledgements. The authors thank Alexandra Jones and Larry Horowitz of NOAA/GFDL for close reading and incisive comments that have greatly improved the quality of the manuscript.

V. Balaji is supported by the Cooperative Institute for Climate Science, Princeton University, under Award NA08OAR4320752 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of Princeton University, the National Oceanic and Atmospheric Administration, or the U.S. Department of Commerce. He is grateful to the Institut Pierre et Simon Laplace (LABEX-LIPSL) for support in 2015 during which drafts of this paper were written.

References

Adler, R. F., Huffman, G. J., Chang, A., Ferraro, R., Xie, P.-P., Janowiak, J., Rudolf, B., Schneider, U., Curtis, S., Bolvin, D., et al.: The Version-2 Global Precipitation Climatology Project (GPCP) monthly precipitation analysis (1979-present), Journal of hydrometeorology, 4, 1147–1167, 2003.

Alexander, K. and Easterbrook, S.: The software architecture of climate models: a graphical comparison of CMIP5 and EMICAR5 configurations, Geosci. Model Dev. Discuss., 8, 351–379, 2015.

Balaji, V.: The Flexible Modeling System, in: Earth System Modelling - Volume 3, edited by Valcke, S., Redler, R., and Budich, R., SpringerBriefs in Earth System Sciences, pp. 33–41, Springer Berlin Heidelberg, 2012.

Balaji, V.: Climate Computing: The State of Play, Comput. Sci. Eng., 17, 9–13, 2015.

Balaji, V., Anderson, J., Held, I., Winton, M., Durachta, J., Malyshev, S., and Stouffer, R. J.: The Exchange Grid: a mechanism for data exchange between Earth System components on independent grids, in: Parallel Computational Fluid Dynamics: Theory and Applications, Proceedings of the 2005 International Conference on Parallel Computational Fluid Dynamics, May 24-27, College Park, MD, USA, edited by Deane, A., Brenner, G., Ecer, A., Emerson, D., McDonough, J., Periaux, J., Satofuka, N., , and Tromeur-Dervout, D., Elsevier, 2006.

Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., Caubel, A., Durachta, J., Foujols, M.-A., Lister, G., Mocavero, S., Underwood, S., , and Wright, G.: CPMIP: Measurements of Real Computational Performance of Earth System Models, Geosci. Model Dev. Discuss., submitted., 2016.

Bozzo, A., Pincus, R., Sandu, I., and Morcrette, J.-J.: Impact of a spectral sampling technique for radiation on ECMWF weather forecasts, Journal of Advances in Modeling Earth Systems, 6, 1288–1300, 2014.

Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., and McDonald, J.: Parallel Programming in OpenMP, Morgan-Kaufmann, Inc., 2001.

Charles, J., Sawyer, W., Dolz, M. F., and Catalán, S.: Evaluating the performance and energy efficiency of the COSMO-ART model system, Computer Science-Research and Development, 30, 177–186, 2015.

Chien, A. A. and Karamcheti, V.: Moore's Law: The First Ending and A New Beginning, Computer, 12, 48–53, 2013.

Cumming, B., Fourestey, G., Fuhrer, O., Gysi, T., Fatica, M., and Schulthess, T. C.: Application centric energy-efficiency study of distributed multi-core and hybrid CPU-GPU systems, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 819–829, IEEE Press, 2014.

Dahan-Dalmedico, A.: History and Epistemology of Models: Meteorology (1946–1963) as a Case Study, Archive for history of exact sciences, 55, 395–422, doi:10.1007/s004070000032, 2001.

Dennis, J. M., Vertenstein, M., Worley, P. H., Mirin, A. A., Craig, A. P., Jacob, R., and Mickelson, S.: Computational performance of ultra-high-resolution capability in the Community Earth System Model, International Journal of High Performance Computing Applications, 26, 5–16, 2012.

Donner, L. J., Wyman, B. L., Hemler, R. S., Horowitz, L. W., Ming, Y., Zhao, M., Golaz, J.-C., Ginoux, P., Lin, S.-J., Schwarzkopf, M. D., Austin, J., Alaka, G., Cooke, W. F., Delworth, T. L., Freidenreich, S. M., Gordon, C. T., Griffies, S. M., Held, I. M., Hurlin, W. J., Klein, S. A., Knutson, T. R., Langenhorst, A. R., Lee, H.-C., Lin, Y., Magi, B. I., Malyshev, S. L., Milly, P. C. D., Naik, V., Nath, M. J., Pincus, R., Ploshay, J. J., Ramaswamy, V., Seman, C. J., Shevliakova, E., Sirutis, J. J., Stern, W. F., Stouffer, R. J., Wilson, R. J., Winton, M., Wittenberg, A. T., and Zeng, F.: The Dynamical Core, Physical Parameterizations, and Basic Simulation Characteristics of the Atmospheric Component AM3 of the GFDL Global Coupled Model CM3, J. Climate, 24, 3484–3519, 2011.

Düben, P. D., Joven, J., Lingamneni, A., McNamara, H., De Micheli, G., Palem, K. V., and Palmer, T.: On the use of inexact, pruned hardware in atmospheric modelling, Phil. Trans. Roy. Soc. London A: Math., Phys. and Engg. Sci., 372, 20130 276, 2014.

Durran, D. R.: Numerical Methods for Wave Equations in Geophysical Flud Dynamics, Springer-Verlag, 1999.

Ford, R., Glover, M., Ham, D., Hobson, M., Maynard, C., Mitchell, L., Mullerworth, S., Pickles, S., Rezny, M., Riley, G., Wood, N., and Ashworth, M.: Towards Performance Portability with GungHo, in: EGU General Assembly Conference Abstracts, vol. 16 of *EGU General Assembly Conference Abstracts*, p. 13243, 2014.

Freidenreich, S. and Ramaswamy, V.: A New Multiple-Band Solar Radiative Parameterization for General Circulation Models, Journal of Geophysical Research: Atmospheres (1984–2012), 104, 31 389–31 409, 1999.

Fuhrer, O., Osuna, C., Lapillonne, X., Gysi, T., Cumming, B., Bianco, M., Arteaga, A., and Schulthess, T. C.: Towards a performance portable, architecture agnostic implementation strategy for weather and climate models, Supercomputing frontiers and innovations, 1, 2014.

Gates, W. L.: AMIP: The Atmospheric Model Intercomparison Project, Bulletin of the American Meteorological Society, 73, 1962–1970, 1992.

Govett, M., Middlecoff, J., and Henderson, T.: Directive-based parallelization of the NIM weather model for GPUs, in: Proceedings of the First Workshop on Accelerator Programming using Directives, pp. 55–61, IEEE Press, 2014.

Gropp, W., Huss-Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., and Snir, M.: MPI: The Complete Reference. The MPI-2 Extensions., vol. 2, MIT Press, 1998.

Harris, L. M. and Lin, S.-J.: A two-way nested global-regional dynamical core on the cubed-sphere grid, Mon. Wea. Rev., 141, 283–306, 2013.

Hourdin, F., Mauriten, T., Getellman, A., Golaz, J.-C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., Rauser, F., Rio, C., Tomassini, L., Watanabe, M., and Williamson, D.: The art and science of climate model tuning, Bull. Amer. Met. Soc., accepted for publication., 2016.

Iacono, M. J., Berthiaume, D., and Michalakes, J.: Enhancing Efficiency Of The RRTMG Radiation Code With GPU And MIC Approaches For Numerical Weather Prediction Models, in: 14th Conf. on Atmospheric Radiation, Boston, MA, Amer. Meteor. Soc., p. 156, 2014.

Kogge, P., Bergman, K., Borkar, S., Campbell, D., Carson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hill, K., et al.: Exascale computing study: Technology challenges in achieving exascale systems, DARPA Information Processing Techniques Office, 705, 2008.

Korkmaz, P., Akgul, B., Chakrapani, L., and Palem, K.: Advocating noise as an agent for ultra low-energy computing: Probabilistic CMOS devices and their characteristics, Japanese Journal of Applied Physics, SSDM Special Issue Part, 1, 3307–3316, 2006.

Loeb, N. G., Wielicki, B. A., Doelling, D. R., Smith, G. L., Keyes, D. F., Kato, S., Manalo-Smith, N., and Wong, T.: Toward optimal closure of the Earth's top-of-atmosphere radiation budget, Journal of Climate, 22, 748–766, 2009.

Manners, J., Thelen, J., Petch, J., Hill, P., and Edwards, J.: Two fast radiative transfer methods to improve the temporal sampling of clouds in numerical weather prediction and climate models, QJR Meteorol. Soc, 135, 457–468, 2009.

Mittal, S. and Vetter, J. S.: A Survey of CPU-GPU Heterogeneous Computing Techniques., ACM Comput. Surv., 2015.

Morcrette, J.-J.: On the effects of the temporal and spatial sampling of radiation fields on the ECMWF forecasts and analyses, Mon. Wea. Rev., 128, 876–887, 2000.

Morcrette, J.-J., Mozdzynski, G., and Leutbecher, M.: A reduced radiation grid for the ECMWF Integrated Forecasting System, Mon. Wea. Rev., 136, 4760–4772, 2008.

Pauluis, O. and Emanuel, K.: Numerical instability resulting from infrequent calculation of radiative heating, Mon. Wea. Rev., 132, 673–686, 2004.

Pincus, R. and Stevens, B.: Monte Carlo spectral integration: A consistent approximation for radiative transfer in large eddy simulations, Journal of Advances in Modeling Earth Systems, 1, 2009.

Platzman, G. W.: The ENIAC Computations of 1950 – Gateway to Numerical Weather Prediction, Bulletin of the American Meteorological Society, 60, 302–312, 1979.

Ridge, D., Becker, D., Merkey, P., and Sterling, T.: Beowulf: harnessing the power of parallelism in a pile-of-PCs, in: Aerospace Conference, 1997. Proceedings., IEEE, vol. 2, pp. 79–91, IEEE, 1997.

Schwarzkopf, M. D. and Ramaswamy, V.: Radiative effects of CH4, N2O, halocarbons and the foreign-broadened H2O continuum: A GCM experiment, Journal of Geophysical Research: Atmospheres (1984–2012), 104, 9467–9488, 1999.

Solomon, S.: Climate change 2007-the physical science basis: Working group I contribution to the fourth assessment report of the IPCC, vol. 4, Cambridge University Press, 2007.

Sterling, T. L.: Beowulf cluster computing with Linux, MIT press, 2002.

Xu, K.-M. and Randall, D. A.: Impact of interactive radiative transfer on the macroscopic behavior of cumulus ensembles. Part I: Radiation parameterization and sensitivity tests, J. Atmos. Sci., 52, 785–799, 1995.

Xue, W., Yang, C., Fu, H., Wang, X., Xu, Y., Gan, L., Lu, Y., and Zhu, X.: Enabling and Scaling a Global Shallow-Water Atmospheric Model on Tianhe-2, in: Parallel and Distributed Processing Symposium, 2014 IEEE 28th International, pp. 745–754, doi:10.1109/IPDPS.2014.82, 2014.

Zhao, M., Golaz, J.-C., Held, I., Ramaswamy, V., Lin, S.-J., Ming, Y., Ginoux, P., Wyman, B., Donner, L., Paynter, D., et al.: Uncertainty in model climate sensitivity traced to representations of cumulus precipitation microphysics, Journal of Climate, 29, 543–560, 2016.

Response to gmd-2016-114-SC1:

1. The main paper must give the model name and version number (or other unique identifier) in the title. If the model development relates to a single model then the model name and the version number must be included in the title of the paper. If the main intention of an article is to make a general (i.e. model independent) statement about the usefulness of a new development, but the usefulness is shown with the help of one specific model, the model name and version number must be stated in the title. The title could have a form such as, "Title outlining amazing generic advance: a case study with Model XXX (version Y)".

   In response, we have modified the title as follows: "Coarse-grained component concurrency in Earth System modeling: Parallelizing atmospheric radiative transfer in the GFDL AM3 model using the Flexible Modeling System coupling framework".

Response to gmd-2016-114-RC1:

This is a well composed manuscript written in a lively style to describe experiments in finer grained component concurrency for at atmospheric model. The novel aspect is the use of component concurrency to improve scalability in an atmospheric code. In general the manuscript is appropriate for GMD, but revisions would strengthen the presentation. The suggested revisions are substantial, but of a nature that the editor can readily adjudicate.

2. I enjoyed the more lyrical style, but in places the style came across as glib, particularly when precision was sacrificed for poetry. Grounding the manuscript better in data and substance would strengthen the exposition greatly.

   We have taken some pains to sharpen the arguments and make them more rigorous at many places through the text, including the specific items highlighted by Reviewer 1 below. We hope we have retained some of the "lyricism" without giving in to "glibness".

3. The figures are substandard and not sufficiently quantitative. In particular differences between different configurations of the same simulation are preferred over differences to the observations, as the former, not the latter, is the point of the manuscript. Also attention to map projections and color scales is required.

   Please see reply to (25) below.

4. The frequent reference to Pauluis and Emanuel was insufficiently discriminating and in places misleading. The reference gives the impression that infrequent coupling to radiation is a substantial source of bias and also model instability. This is not what the manuscript is about. Moreover the Pauluis Emanuel study, while noteworthy, has not been shown to generalize. It might, but the literature is not there. In the grand scheme of things, the trade-off in accuracy and stability of calling radiation more frequently, versus simulating at higher resolution, is not well understood.

   We agree that that there need to be more studies of the effects of temporal subsampling of radiation. However, there is no case to be made that temporal subsampling is superior for *physical* reasons: at best, we can claim that we do not see marked increase in bias or RMS error as an effect of subsampling, and that it improves time to solution. The Morcrette papers we have cited in response to Reviewer 1's other comments (about spectral and spatial subsampling) also only claim expediency and decreased time to solution as a reason. They indicate error increases are tolerable over short timescales and probably increase model bias on climate timescales. Based on these results, we believe our focus on methods to possibly eliminate temporal subsampling are justified. We have modified our discussion of the Pauluis-Emanuel and Xu-Randall studies to indicate the need for further exploration of the effects of temporal subsampling, including solution convergence as Δt_{rad} and Δt_{atm} converge. See page 4, line 4.

5. Some of the hard-core computational issues are insufficiently addressed. In particular, component concurrency probably will affect the hard scaling floor, and the trade-off between communication vs processing in codes like ESMs with low arithmetic intensity. I guess there are also trade-offs that arise because of the need for the concurrency to be in a shared memory implementation. The manuscript would be strengthened if these issues were discussed in a more thorough manner.

While optimal performance for OpenMP threading is generally achieved when the threadpool is constrained to a single socket, scalability of concurrent components is limited by the number of cores in a shared-memory node. This number is slated to be very much larger than today, $\mathcal{O}(100)$ on MIC architectures. See also reply to (35) below.

6. There are too many acronyms, and some seem indiscriminately chosen.

We have gone through the text and noted the following acronyms: GPU, MIC, CCC, ENIAC, ESM, IPCC, FAR, AR5, HPC, FMS, SIMD, SPMD, MPMD, SMP, MPI, GFDL, OpenMP/OMP, AOGCM, LBL, PE, AM3/CM3, SST, AMIP, GPCP, CERES EBAF, NOAA, SYPD, CHSY, NPES, CPU. Most of these are quite well-known and commonly used, and cover standard terminology in the computing and climate literature. Some refer to institutions, models, and observational datasets. We hope none of them look "indiscriminate" in the revised text. All have been defined at first use, thanks to the close reading by reviewers: see (8) below.

7. The manuscript does not distinguish between AGCMs and ESMs, frequently discussing the results in terms of ESMs but then presenting results for the atmospheric GCM alone. Do the results generalize to problems that already have much more concurrency? I guess so but this is a separate point and the manuscript should discriminate between what was done and what is inferred based on what was done.

We have clarified our use of "ESM" as a generic term for models of any level in the complexity hierarchy. We have also more clearly explained that the current implementation is in an atmosphere-only setting, but readily generalized to more components, which we describe in our plans for future work. This includes components invoked only in ESMs defined in the narrower sense, that of models that include an interactive biosphere. See page 2, line 26; page 15, line 16; and the caption to Figure 2.

8. p1l2: Acronyms

Fixed, see page 1, line 2

9. p1l24: Is that true, or did the technology also drive things. it makes it sound like everything was possible and we just chose something, rather than necessity driving development.

It is mostly believed that the principal cause for the decline of vector computing was economics: commodity clusters were able to match performance of custom hardware at a vastly lower price. We have added some text and citations to bolster this argument, see page 2, line 9.

10. p2l5: I am not sure what figure the authors are trying to say here. Actually resolution has not kept pace with computing as far as I can tell, and the reference to the figure does not make sense.

There was an error: it was Fig 1.4 from the 2007 Report, not the 2013 one we cited. It is now fixed, and a link to the figure itself is provided. See page 2, line 15.

11. p2l7: Would help to explain to the reader the phrase "arithmetic and logic" is a memory fetch logic?

Fixed, see page 2, line 17

12. p2l11: “The state of play of climate computing in the face of these challenges” this phrase comes across as a bit of a throw away. Did the Balaji (2015) paper make a point that is important for the present discussion? If so what was it.

    Fixed, see page 2, line 24

13. p2l20: “and there is constant churn of operations” missing an article. . . lyrical, but it gives the idea that the computer is kept busy computing rather than moving information around. Most codes are memory bandwidth limited . . . which you get to shortly. But this intro sentence did not prepare me well.

    Fixed, see page 3, line 3

14. p2l30: Here: “Of the many factors of 10 increase in performance needed to get to the promised land of ’exascale computing’, we believe at least one can come from component organization.” I would prefer precision over poetry.

    Fixed, see page 3, line 15

15. p3l5: Weather centers also run spatially coarse-grained radiation. Also some have proposed a form of coarse graining in the spectral domain, i.e., Monte-Carlo Spectral Integration.

    This is an important oversight, and we have added a short discussion on spatial and spectral subsampling, see page 3, line 28. The main focus of this paper remains the elimination of temporal subsampling at modest computational expense.

16. p4l9: A reference is needed here. My intuition suggests that such high processor counts have only been applied to more traditional Atmosphere, or Ocean or Atmosphere Ocean Problems, but not simulations of the carbon cycle, i.e., the use of ESM here is misleading as earlier it implied biology.

    The reviewer is correct: most of the very high end PE counts are for specialized problems, not full ESMs in climate mode. We have clarified this, and added some references: see page 5, line 7

17. p6l15: Subscript abbreviations are usually written in roman font.

    Fixed, here and everywhere else in the text: see page 7, line 10.

18. Figure 3: I spent some time on this and I am not sure I understood it. The vertical dimension denotes sequence, first to last from top to bottom. The boxes indicated either the legend or a component process? The thickness of the box denotes? A figure should be illustrative, not a riddle. Also a bit more structure might help me understand what atmosphere up is. The meaning of some colors seems to be specific, others decorative?

    Noted. We have reworded the caption to Figure 3 to clarify these points.

19. p7 l2: “interact strongly with atmospheric chemical species and clouds” this could exclude water vapor, why not say, couple strongly to composition.

    Text clarified: see page 9, line 3.

20. p7 l4: By this definition the ocean does not have tracers. Aren’t tracers really just scalar quantities that are transported with the flow subject to source and sink processes.

    Text clarified: see page 9, line 4.

21. p7 [reviewer typo: this is p8] l6: Seconds and hours (line 9) are units, and can be abbreviated.

    Done: see page 9, line 8.

22. p7 [reviewer typo: this is p8] l9: This makes it sound worse than it may be, some codes rescale the radiative heating rates at each timestep by the insolation, or the surface temperature, in a sense linearizing about the state defined every 3 hrs. Something you mention on the next page, but it comes late.

    Agreed, eliminated last sentence of paragraph, as it’s redundant with what come’s next: see page 9, line 11.

23. p7 [reviewer typo: this is p8] l19: Spell out PE . . . Processing Elements? A socket?

    Fixed: see page 9, line 20.

24. p7 l14 [reviewer typo: this is p8 l24]: “Architected”? Okay, it can be used as a verb; but I think designed, or constructed would be better.

    Rewritten: see page 9, line 26.

25. Figure 5: This needs redrafting, first for the colour scale (no rainbows); second to show the common color scales in those panels where it is appropriate. The highly distorted projection should either be motivated or replaced.

    If the objection is to the use of the Mercator projection: we agree that it introduces distortions, particularly areal distortion as you approach the poles. While this is known, this projection is customary and in very widespread use in the literature. We do not believe the use in this instance to be egregious, as the results are in no way compromised by the projection.

    Similarly, we have used a somewhat standard color scheme: the rainbow in panel 1, and a warm/cool two-color palette for the difference plots. This again is quite customary.

    We are somewhat puzzled by this remark as neither the projection nor the color scheme are in any way misleading or distorting the results.

    That said, we have modified Fig. 5 and 6 to show the model-obs. differences in Panel (b), and the (smaller) model-model differences in Panels (c) and (d). We hope this makes the discussion clearer. See page 11, line 19, and captions to Figs. 5 and 6.

26. Figure 5: I would like also to see land temperatures. And differences among the simulations are far more interesting than differences with GPCP

    Again, we are puzzled by the remark and unsure how (or why) to add land temperatures to a global plot of precipitation. No land or ocean cells have been masked on this plot.

    The differences between the simulations are interesting, but cannot be over-interpreted, as in practice we would retune the models to maintain the same top-of-atmosphere net radiative flux. The focus of the paper is on the computational aspects and the coupling algorithm. We expect to visit this issue in greater detail in a future paper describing scientific model results from concurrent-radiation run where $\Delta t_{\mathrm{rad}} = \Delta t_{\mathrm{atm}}$, with appropriate tuning. See also response to (28) below.

27. P10l14: "Remarkably" similar. . . my rule of thumb is that people use the word remarkable when they don't know what of substance to remark.

    Rephrased, see page 11, line 21.

28. p10l30: I would make the tuning point later, as it seems as though the authors are interpreting the differences as fundamental, rather then simply an illustration of compensating biases in a manner that is to be anticipated.

    The tuning discussion comes at the tail-end of Section 4.1, which is the last discussion of physical results. We agree that the differences are not fundamental, and in the initial draft had indicated that they were "within the margins of the tuning process." We have expanded on that to make it clearer, as suggested, see page 13, line 5.

29. p11l5: SYPD has a unit, i.e., yr d
    SUPERSCRIPTNB-1

    SYPD is itself a unit, in fact it is introduced using the phrase "simulated years per day, or SYPD", see page 14, line 3. We believe it would be redundant to attach the unit again. Units are mandatory when the same quantity can be expressed in different units, e.g m and cm. That is not so in this case.

    That said, we have added a discussion and definition of SYPD, see page 14, line 3.

30. Table 1: Maybe spell out the acronyms; for example does the introduction of CHSY really help anything? And if so why not PHSY, processor hours per simulated year. The computational cost of radiation appears small. If increasing the frequency of radiation nine-fold increases the computational cost by 50% this implies that the cost of radiation is about 5% of the total computational cost in the default configuration. Is this correct? If so this is rather small compared to some other models, suggesting that the proposed approach might be even more beneficial for other centers, or offer the possibility of more exact representations of radiative transfer. Here some clear numbers would be useful.

    We do indeed believe the CHSY discussion is useful, as both time to solution (SYPD) *and* resource consumption (CHSY) are taken into account in configuring the parallel layout of a production model. See discussion in Balaji et al. (2016). We

have also chosen to be consistent with that paper in naming the throughput measure CHSY instead of PHSY, which we agree would have also been a valid choice.

In regard the second point, the reviewer is correct, and we have made this point now in the text, see page 14, line 15.

31. p13l13: This paragraph is a bit ungrounded in the manuscript, which does not evaluate MPMD approaches. Certainly the GPU rewrite of COSMO has a factor of 3.6 speed up on a first implementation . . . so there is some room for efficiency gains through reprogramming, also the inexact hardware approaches (Dübben and Palmer) merit mention if this paragraph were to be retained and better grounded in the manuscript.

    It's true that the background material is a bit ungrounded in *this* manuscript, but summarizes the findings of Balaji (2015), which does indeed survey MPMD methods and inexact computing approaches. We have rewritten the paragraph to reflect that these are findings from Balaji (2015). See page 15, line 10.

32. p13l20: Where does the order ten components come from. I think "probably not more than ten" would be more accurate, but in either case if this comes at the end it should also be better ground in the manuscript.

    We have toned down the speculative statements here, see page 15, line 24.

33. p14l22: By data do the authors mean model output? Or the performance data, gleaned from the benchmarks? The use of "data" suggests the latter, but the former should also be addressed.

    Fixed, see page 16, line 13

34. page 2, around line 20: the discussion focuses on performance aspects only. Mention of the implications for energy/power use would also be useful here, as the focus is on extreme-scale systems.

    Fixed, see page 2, line 19

35. page 3, line 12: “how it is achieved without increasing data movement” should, I think, be changed to “and how it is achieved with minimal impact on data movement”. This point is discussed further below [*].

    [*] The following point is my main concern with the paper as it stands. I believe the paper would benefit by being clearer on the use, and limitations, of shared memory threading to implement the concurrent execution of the radiation and the rest of the atmosphere model. It is clear that for a given multi-core processor there will be limits on the number of MPI processes per node and the number of threads each model may use within an MPI process without incurring potentially expensive data movement between caches. The example results given are based on two threads for each model. It would be good to make clear the rationale for this choice. Here are some thoughts on this and suggestions for possible changes which might help achieve this. Currently, the use of threads for executing the radiation in parallel with the rest of the atmosphere is described as not incurring any communication costs. While it is true there will be no MPI communication incurred between cores running the atmosphere and cores running the radiation, I believe there may well be some extra remote data accesses (i.e. cache misses) incurred between cores running the atmosphere and cores running radiation. The magnitude of this effect is, of course, architecture dependent and also depends on the number of threads used for each model and the mapping of the threads to cores. The results presented are for AMD Interlagos processors with two threads used for the atmosphere model and two threads used for the radiation model (within each MPI process). The Interlagos processor chip consists of eight 2-core modules. Two threads executing on the same module share an L2 cache. All 2 core modules share a large L3 cache. So, if one atmosphere and one radiation thread share a module, they can share data in the L2 cache (as well as the L3 cache). If two Atmosphere threads share a module and two radiation threads share a module, they will communicate through the L3 cache, which is more expensive in terms of cycles to access. If threads are on separate processor chips, there will be (even more expensive) data movement within the shared memory node.

    The cache behaviour in either of the above cases is likely to be different to that of a single thread running first the atmosphere and then the radiation. If more than two threads were used for each model, some sharing would have to take place via the L3 cache. Total thread numbers are clearly limited by the core count of a shared memory node. I would suggest to the authors that some clarification of these issues be made. For example: - in Figures 3 and 4, the images depicting MPI and OpenMP could be re-drawn to illustrate the relationship of threads within MPI in each case. In Figure 3, this would simply show multiple threads in an MPI task and multiple MPI tasks. In Figure 4, MPI tasks could be shown with both atmosphere and radiation threads or with ocean threads. For Figure 4, this might be something like: [AARR] [AARR]... [O] [O]... (where [] here represent MPI processes and letters represent threads and their models) - In Section 3 (perhaps?), a brief description of a multi-core processor (like the Interlagos) could be given along with the implications of thread-to-core mapping. This description would help to explain the benefit of using OpenMP to exploit

parallelism between the atmosphere and radiation models (i.e. no MPI communication) and pave the way for a discussion of the potential for sharing data between caches in the specific configurations presented in the results section.

The concurrency software design was formulated to ensure the radiation component was truly independent from the remaining atmosphere components with a single data synchronization point (copy) at the end of each time step. The data copying, itself implemented with OpenMP threading, can be completely removed via index flipping, but this might result in non-local accesses or extra cache invalidations - so a simple data copy is preferable. Additionally, each thread immediately calls a subroutine (either atmosphere or radiation) and is working with either explicitly blocked data structures or pseudo-blocked data created via copy-in variables. So not only are the atmospheric and radiation components completely independent, but within each component the data for each thread is isolated as well. If a radical paradigm shift in computing occurs, this forethought in the design allows us to quickly re-cast the radiation threads as separate MPI processes with a needed MPI communication synchronization point replacing the current data copy. See also (5) above.

36. page 5, Figure 1: The role of A

SUBSCRIPTNBt should be depicted in the top figure (to be consistent with equations 1 and 2), I feel. Also, in many models, the atmosphere is executed on more processors than the ocean (because it scales better). Is this diagram consistent in this respect with the FMS model being described? Also, the bottom figure in Figure 1 implies that the ocean is executed on fewer processors in the concurrent set up. Is the intention to simply show a deployment utilising the same number of processors in total? If so, that should be made clear in the caption and text.

The relative balance of PE count between atmosphere and ocean depends on many factors, and within the FMS system, we have examples of both. However, in Figure 1 this is mostly schematic, indicating that some PEs may idle in serial processing.

We have adjusted the numbering of the component timesteps to be consistent with Eqs. 1-4.

37. page 9, line 2: “chosen to offer optimal load balance” could be extended to “chosen to offer optimal load balance and data sharing”, for example. [I generally have a concern over the use of the work “optimal”, which has a formal sense of “provably best”. The word “good” might be better unless the load balance is provably optimal?]

Fixed, see page 10, line 6

38. page 11, line 9: The above arguments are tied up with the statement that “All runs use the optimal processor/thread layout for a given PE count”. Some explanation about what this layout is and how it was chosen could be added.

Done, see page 14, line 6

39. page 1, line 4: I suggest changing “based on marginal increases in clock speed” to, for example, “based on, at best, marginal increases in clock speed” since it is likely clock speeds may decrease in future in some systems.

Done, see page 1, line 3

40. page 1, line 14: Define the acronym CCC here.

    Done, see page 1, line 16

41. page 1, line 15: is a little ambiguous about what is running in parallel ("and all other atmospheric physics components". I would suggest making it clear that there are only two concurrent components (i.e. not all "other atmospheric components" are executed in parallel with each other!

    Done, see page 1, line 18

42. page 2, line 3: perhaps provide a reference to the IPPC assessments.

    See page 2, line 15.

43. Section 2: page 4, line 4: needs a closing bracket after "example".

    Done, see page 4, line 31

44. page 8, line 24: "Individual" should be "individual".

    Done, see page 9, line 28

45. page 9, Figure 4: In this figure, the Land and Ice models are shown as executing concurrently but this is not mentioned in the text. This should be explained (or made consistent with Figure 3).

    Fixed. It should indeed have been consistent with Fig. 3. See new Fig. 4.

46. page 10, line 9: "that that" should be "that".

    Done, see page 11, line 12

47. page 10, line 11: This sentence would benefit from having a reference added.

    Done, see page 11, line 14

48. page 10, line 13: Expand the acronym GPCP here as a definition.

    Done, see page 11, line 17

49. page 10, lines 15-17: The point here is, I think, that this result is counter intuitive. If that is correct, it would be worth stating.

    As noted above in (25) and (28), the differences between the runs are small and within the bounds of the tuning process. Differences should not be over-interpreted as they are likely to vanish upon tuning. See page 13, line 5 also.

50. page 11, line 3: "less expensive as..." should be "less expensive as the...".

    Done, see page 13, line 4

51. page 12, line 4: the figures for processor count and SYPD given in this line are rounded versions of those in Table 1. Those in the previous sentence are not rounded. Please use the precise figures for consistency.

    Done, see page 14, line 19

52. page 13, line 18: It would be worth giving the definition of CCC again here to remind the reader.

    Done, see page 15, line 16