Geosci. Model Dev., 9, 3605–3616, 2016
www.geosci-model-dev.net/9/3605/2016/
doi:10.5194/gmd-9-3605-2016

Geoscientific Model Development
Open Access
EGU

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], **Rusty Benson**[2], **Bruce Wyman**[2], **and Isaac Held**[2]

[1]Princeton University, Cooperative Institute of Climate Science, Princeton NJ, USA
[2]National Oceanic and Atmospheric Administration/Geophysical Fluid Dynamics Laboratory (NOAA/GFDL), Princeton NJ, USA

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

Received: 6 May 2016 – Published in Geosci. Model Dev. Discuss.: 24 May 2016
Revised: 7 September 2016 – Accepted: 13 September 2016 – Published: 11 October 2016

Abstract. Climate models represent a large variety of processes on a variety of timescales and space scales, a canonical example of multi-physics multi-scale modeling. Current hardware trends, such as Graphical Processing Units (GPUs) and Many Integrated Core (MIC) chips, 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 the scalability of Earth system models.

We present here first results from this approach, called coarse-grained component concurrency, or CCC. Within the Geophysical Fluid Dynamics Laboratory (GFDL) Flexible Modeling System (FMS), 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 (HPC). 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 the 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 Intergovernmental Panel on Climate Change (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 the Fourth Assessment Report (AR4) (Solomon, 2007, see e.g., the iconic Figs. 1.2[1] and 1.4[2] from the summary for policymakers).

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 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 al., 2008). These massive increases in concurrency pose challenges for 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, each embodying different physics: there are no performance hotspots. This also means that fresh operators and operands – embodied in physics subroutines and associated variables – are constantly being transferred to and from memory with each context switch, and locality and reuse are 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 for $\mathcal{O}(100)$ 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 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 Geophysical Fluid Dynamics Laboratory (GFDL) Flexible Modeling System (FMS). This is a relatively expensive component of the atmospheric physics, and is usually run at a much coarser time step than the rest of the atmosphere (temporal subsampling), purely for expediency rather than any physical justification. 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 as well as the temporal domain (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

[1] https://www.ipcc.ch/publications_and_data/ar4/wg1/en/figure-1-2.html

[2] https://www.ipcc.ch/publications_and_data/ar4/wg1/en/figure-1-4.html

[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 Fig. 2.

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 time step has been shown by Pauluis and Emanuel (2004) to be a potential source of instability and bias in radiating atmospheres. Xu and Randall (1995) also showed that this problem gets considerably worse as the resolution of models increases. A useful way to think about this is that using different time steps 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 time step 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 Sect. 2 we briefly review current approaches to parallelism in ESMs, particularly in the coupling framework. In Sect. 3 we describe our approach to coarse-grained concurrency, how it is achieved without increasing data movement. In Sect. 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 vs. serial approaches in otherwise identical physical formulations, and the ability to run with a much smaller radiation time step without increasing the time to solution. Finally, in Sect. 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 has 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 for 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 for 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 (multiple program, multiple data) framework. Soon after, distributed computing, consisting of networked clusters of commodity computers, known as symmetric multi-processors, 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 FMS, 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 them; see e.g., Dennis et al., 2012, S.-J. Lin and C. Kerr, personal communication, 2013) have 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

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

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

where $f()$ and $g()$ nominally represent the feedbacks from the other component, and the superscript represents a discrete time step. 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 time stepping, 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.

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

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

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 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 Sect. 5 we will revisit this issue in the context of future work.

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 (sub-grid-scale flow, and other thermodynamic and physical–chemical processes, including those associated with clouds and sub-grid-scale 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 Fig. 2. Different

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

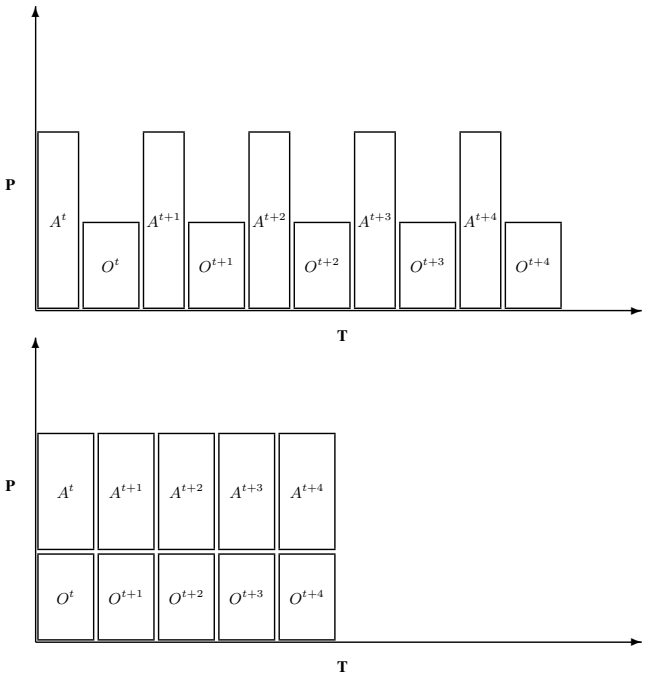

Figure 1. Serial and concurrent coupling sequences, with time on the x axis and processors on the y axis. In the serial case, both components may not scale to the same processor count, leaving some processors idle. Note that in the concurrent coupling sequence below, O^{t+1} only has access to the lagged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

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).

How the notional architecture of Fig. 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 FMS. The atmosphere and ocean components are set up to run in parallel in distributed memory, communicating on the slow coupling time step Δt_{cpld}, on the order of $\Delta t_{\mathrm{cpld}} = 3600\,\mathrm{s}$ for its flagship application, decadal–centennial climate change. Within the slow coupling loop, the atmosphere communicates on a fast coupling time step Δt_{atm} with a typical value of 1200 s, 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 a 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 Fig. 3.

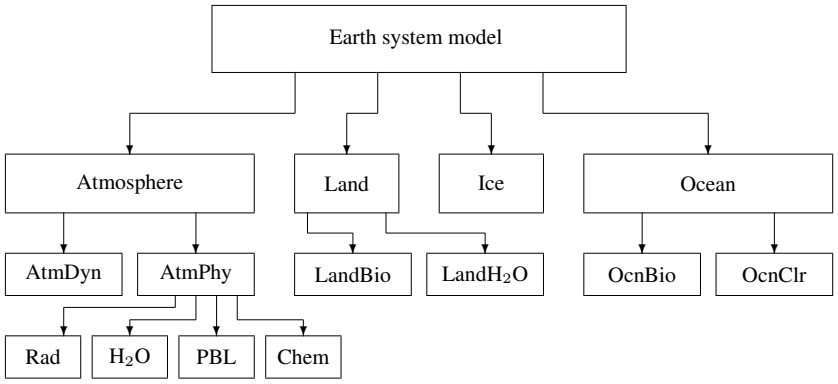

Figure 2. Notional architecture of an Earth system model, with components embodying different aspects of the climate system, hierarchically organized. Models on a hierarchy of complexity ranging from single-component (e.g., atmosphere-only) models to full-scale coupled models with an interactive biosphere,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

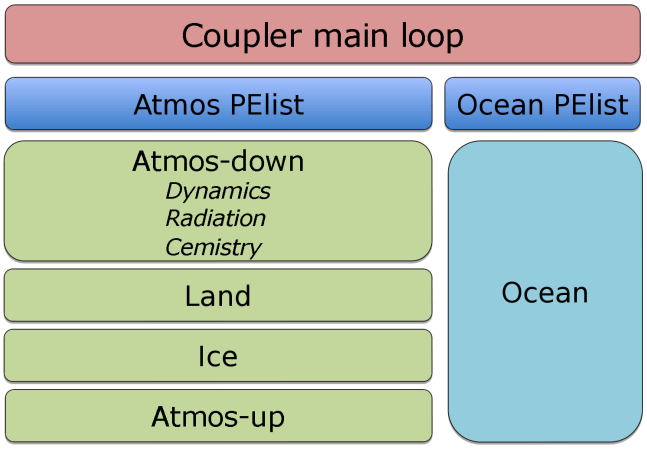

Figure 3. FMS parallel coupling architecture in processor–time space, with processors across, and time increasing downward. Components have different horizontal and vertical extents to indicate the degree of parallelism and time of execution, though these extents are notional and not to be interpreted 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, components execute serially, including tight coupling to the land and ice–ocean surface. The down-up sequence of implicit coupling is explained in Balaji et al. (2006). These components internally use shared-memory (OpenMP) coupling, indicated in light green. The ocean component at the present time is MPI only, indicated in light blue.

The "Atmos-up" step is quite lightweight, including adjustments to the atmospheric state imposed by moist physics, and completing the upsweep 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 "Atmos-down" step.

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 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 3-D model fields (in atmosphere or ocean) that are advected by the evolving dynamics, and participating in physics and chemistry processes at individual grid points.

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 $\Delta t_{\mathrm{rad}} = 10\,800$ s, or $9 \times \Delta t_{\mathrm{atm}}$, 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 time step Δt_{atm}. This means that at intermediate (non-radiation) atmospheric time steps, the radiation is responding to a lagged state of at-

mospheric tracers, which may be as much as $\Delta t_{\rm rad} - \Delta t_{\rm atm}$ (~ 3 h) behind.

This time step discrepancy is vexing, but most climate models around the world make a similar compromise, with a radiation time step longer than the physics time step. 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

Before we begin describing a method for casting the radiation code in FMS as a concurrent component, we need to describe the current methodology shown in Fig. 3. Concurrency between atmosphere and ocean component on the slow coupling time step is achieved using distributed computing techniques, with the components running on separate processor sets or PElists. In FMS terminology, a PE or processing element is a unit of hardware supporting a single execution thread, sometimes called a core. A PElist is synonymous with a communicator in MPI terminology, and lists the PEs assigned in distributed memory processing. Each PE in a PElist can spawn multiple shared-memory execution threads. These threads are assigned to other PEs to avoid contention. Coupling fields are transferred between atmosphere and ocean using the exchange grid (Balaji et al., 2006) and message passing. Within the atmosphere component, shared-memory parallelism using OpenMP is already implemented. For the dynamics phase, the OpenMP acts on individual loops, some of which may contain calls to subroutines or comprised of large programmatic constructs. These include regions where concurrency is on slabs (horizontally tightly coupled) and others organized in columns (vertically tightly coupled).

Unlike the dynamics, the physics is organized entirely columnwise, and 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 arrays here can be organized into groups of vertical columns, or blocks, that can be scheduled onto the same OpenMP threads at a high (coarse) level – meaning a single thread will persistently see the block (thus assuring thread data affinity) through the complete Atmos-down phase (sans dynamics), and again through the Atmos-up phase.

We now come to the reorganization of components for CCC in the current study. Understanding that the radiation is but one phase of the physics, which is already utilizing blocks and OpenMP threads, it makes sense to extend the concept and have the radiation run concurrently in a separate group of OpenMP threads. In the concurrent radiation architecture, shown in Fig. 4, the decision was made to utilize nested

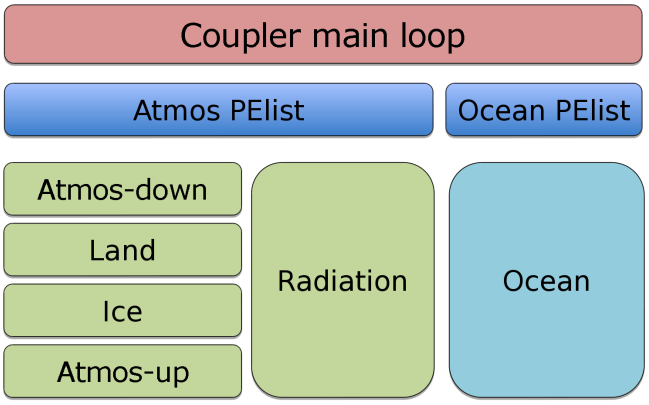

Figure 4. Concurrent radiation architecture. See Fig. 3 for comparison, and explanation of legends.

OpenMP, instead of a flat thread pool. Each MPI rank assigned to the atmosphere PElist starts up an OpenMP region with two threads, one to act as the master for the radiation and the other to drive the non-radiation components. These master threads are able to utilize the nested OpenMP constructs, to start up a number of atmosphere threads (A) and radiation threads (R), where the total numbers of threads $T = A + R$. For a given value of T, A and R can be dynamically adjusted during the course of a run to achieve optimal load balance. Because the radiation and atmosphere concurrency occur at a high level, the memory space is unique to each component and furthermore, the memory space is unique to each block. This separation of memory spaces ensures there are no performance deficiencies due to cache coherency effects (false sharing or cache invalidations). A single data synchronization point (copy) at the end of each atmospheric time step ensures that the atmospheric and radiation components remain completely independent.

In the limit, we could create blocks containing a single column, so that A and R both equal the number of columns in the domain, and $T = 2A$. 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. Current processors rely on having a moderate number of data elements to achieve best performance by hiding the various latencies for each instruction, including time spent waiting for operands to be loaded to memory registers. Empirically, on current technology, we have found optimal results using blocks of $\mathcal{O}(32)$ columns.

4 Results

4.1 Results from AMIP runs

The model utilized here is based on AM3, the atmosphere–land component of the GFDL Climate Model v3 (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 the following:

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

2. Serial radiation (SERIAL) uses $\Delta t_{\rm rad} = \Delta t_{\rm atm} = 1200$ s.

3. Concurrent radiation (CONCUR) also uses $\Delta t_{\rm rad} = \Delta t_{\rm atm} = 1200$ 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 time step, this is not recommended; the atmospheric and radiative states would be effectively 21 600 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 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 time step on precipitation and top-of-atmosphere radiative fluxes in the AMIP simulation. Figure 5 shows the annual mean precipitation bias for the three experiments with respect to the Global Precipitation Climatology Project (GPCP) v2.2 (Adler et al., 2003) climatology. Panel (b) shows the difference between CONTROL and observational precipitation climatology, while (c) and (d) show the model–model differences between the CONTROL run and the SERIAL and CONCUR climatologies. The difference in the pattern of annual mean precipitation due to the change in time step (Fig. 5c and d) is negligible compared to the difference between the model and observations. Panels (c) and (d) are similar to first approximation, indicating that the difference in precipitation pattern due to the choice of serial vs. concurrent integration is smaller than the difference due to the radiation time step and even less significant as compared to the difference between model and observations.

The more significant difference in the simulation due to the change in radiation time step is in the energy balance at the top of the atmosphere between absorbed shortwave and outgoing longwave radiation. The simulated annual mean pattern of this net energy flux is displayed in Fig. 6 in analogous format to Fig. 5. The energy flux data are CERES EBAF edition 2.8 satellite data (Loeb et al., 2009). Neither the radiation time step nor the choice of serial vs. concurrent integration modifies the geographical pattern of model bias significantly, but they do alter the simulation with a fairly spatially uniform offset. Again the details of all the experiments are similar but a closer examination of the global mean biases show the SERIAL and CONCUR cases (Fig. 6c and d) differ from the CONTROL by about $+3.1$ to $+3.6\,{\rm W\,m^{-2}}$. This magnitude of flux difference would have 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 $3.5\,{\rm W\,m^{-2}}$.) 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. 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.

To be a viable climate model, the global mean top-of-atmosphere energy balance has to be small, less than $1\,{\rm W\,m^{-2}}$, to avoid unrealistic climate drift. This global energy balance is tuned in all models, as our ability to simulate the Earth's cloud field from first principles is inadequate to generate a radiation field with the required fidelity. Parameters in the model that control the cloud simulation and that are not strongly constrained by observations are used in this tuning process (see e.g., Hourdin et al., 2016, for a description of the tuning process). The model simulations displayed here have not been re-tuned so as to isolate the effects of the radiation time step and coupling strategy. In our experience a retuning of 3–4 $\rm W\,m^{-2}$ is viable, but large enough that the resulting changes in other aspects of the model simulation can be non-negligible, emphasizing the importance of algorithms that make the reduction in the radiative time step less onerous computationally. However, the difference between serial and concurrent coupling of $0.5\,{\rm W\,m^{-2}}$ is well within the range in which retuning has a marginal impact on other aspects of the simulation, encouraging examination of the performance of the concurrent option.

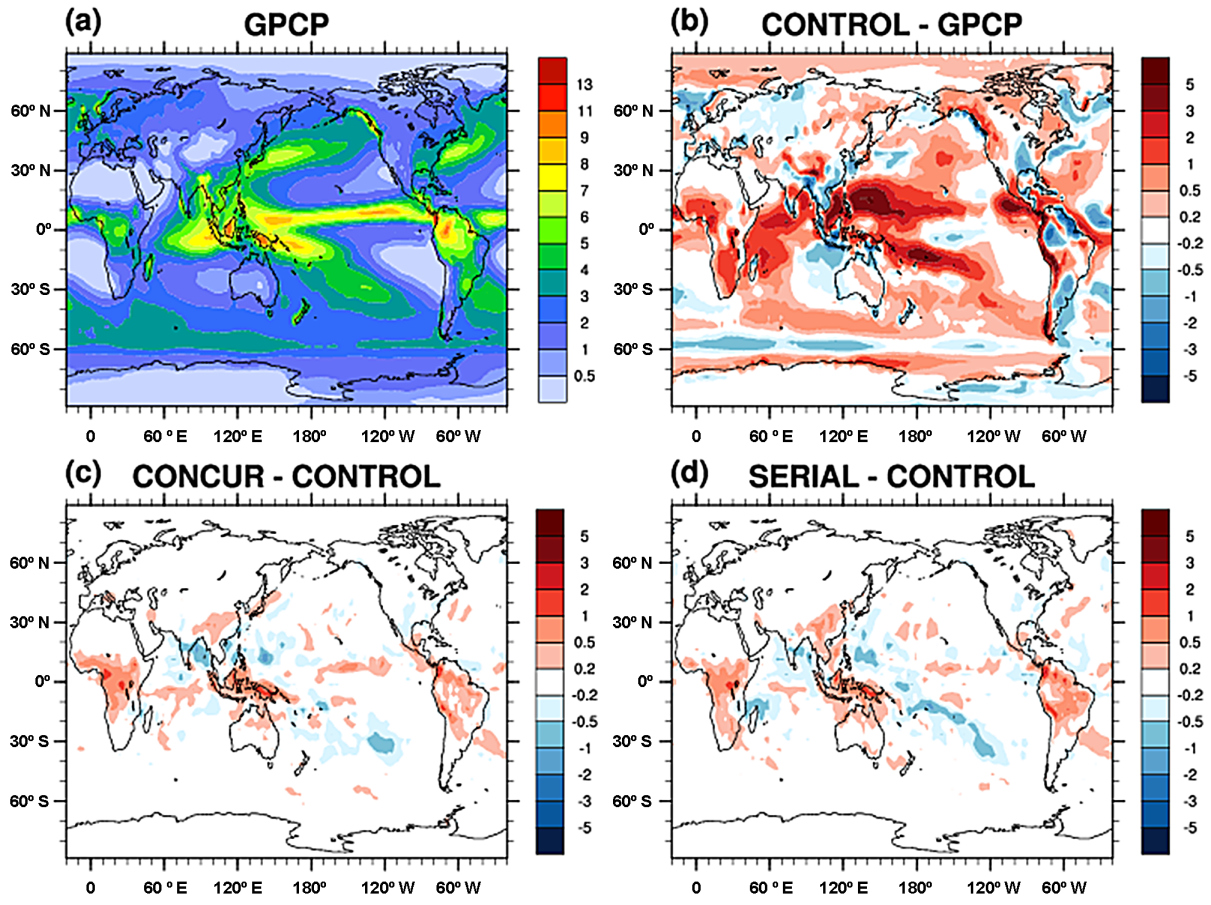

Figure 5. Comparison of model climatologies against GPCP precipitation. **(a)** shows the GPCP climatology, and **(b)** the model climatological biases for the CONTROL run. Panels **(c)** and **(d)** show model–model difference vs. CONTROL for CONCUR and SERIAL runs plotted on the same color scale as **(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 the $\Delta t_{\mathrm{rad}} = 9 \times \Delta t_{\mathrm{atm}}$ setting implies that the radiation code is executed very seldom. As the results of Sect. 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 time step $\Delta t_{\mathrm{rad}} = \Delta t_{\mathrm{atm}}$, 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. 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 that PE count. As Table 1 shows, CONTROL achieved 9.25 SYPD on 1728 PEs. The SERIAL configuration shows the relative cost of radiation to the rest of the model, as shortening Δt_{rad} from 10 800 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 2592 brings us back to 9.1 SYPD. Thus, one can achieve the goal of setting $\Delta t_{\mathrm{rad}} = \Delta t_{\mathrm{atm}}$ without loss in time to solution (SYPD), at a 52 % increased cost in resources (i.e. the CHSY ratio of the two configurations is 1.52). Thus, decreasing the radiation time step 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 Sect. 5 below), this

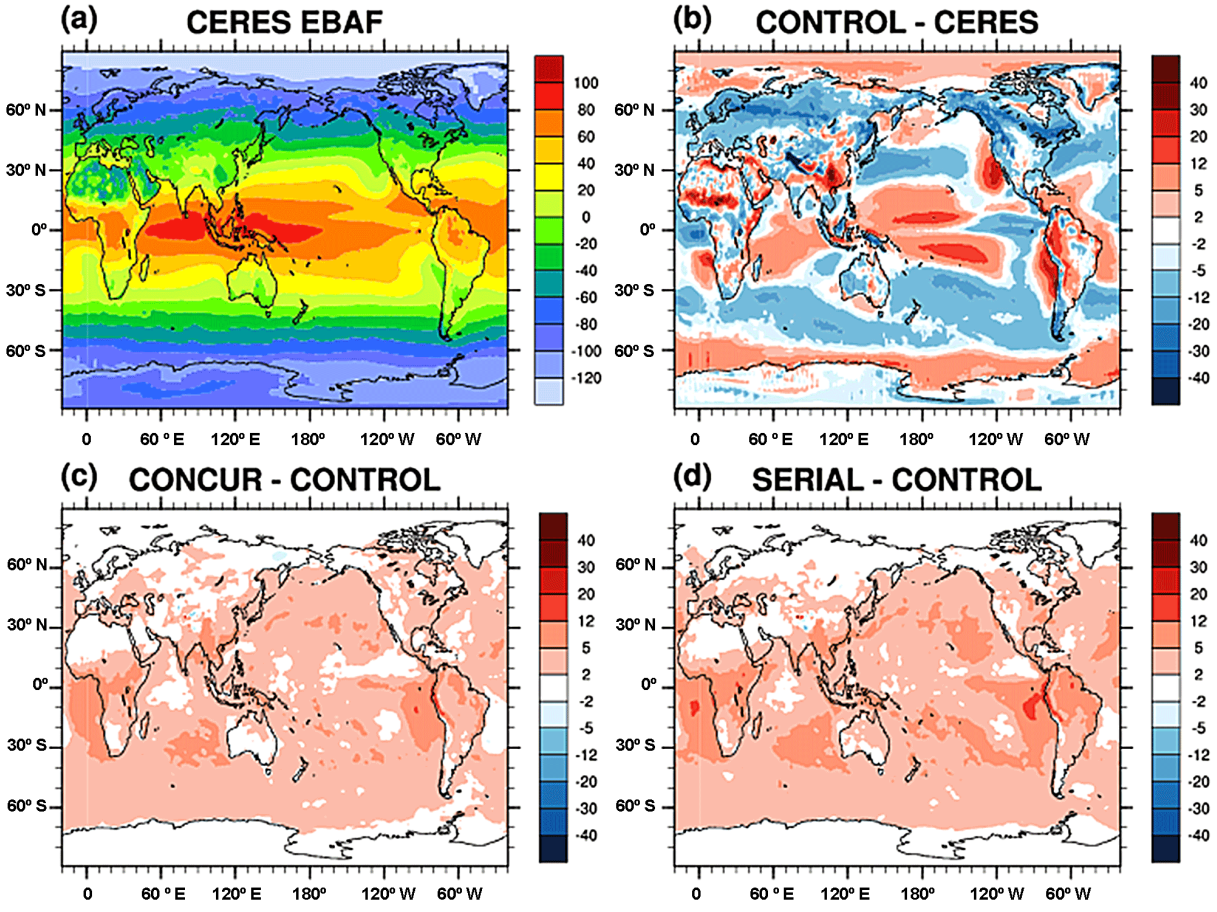

Figure 6. Comparison of model climatologies against CERES EBAF v2.8 climatological top-of-atmosphere radiation budget shown in **(a)**. Panel **(b)** shows model climatological biases for the CONTROL run. Panels **(c)** and **(d)** show model–model difference vs. CONTROL for CONCUR and SERIAL runs plotted on the same color scale as **(b)**.

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 Sect. 5, we are planning to extend the CCC approach to other components including atmospheric chemistry and ocean biogeochemistry.

5 Summary and conclusions

We are at a critical juncture in the evolution of high-performance computing (HPC), another disruptive moment. The era of decreasing time to solution at a fixed problem size, with little or 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 re-imagining of ESM codes will be necessary for the coming novel architectures (Balaji, 2015). 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

Table 1. Performance results from the various configurations discussed. MPI × OMP shows the assignment of MPI processes and OpenMP threads. In the CONCUR cases two 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).

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

at the very highest level of ESM architecture shown in Fig. 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 coarse-grained concurrency (CCC) approach, a *component* may be defined as *one of many units in a multiphysics model, which 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 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.

- 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 (central processing unit).

There remain caveats to this approach. As shown in the discussion of Eq. (3) above, the coupling of concurrent components might be formally unstable. We are exploring more advanced time-coupling algorithms, including three-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 three-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, Topical Editor Sophie Valcke of CERFACS, and two anonymous reviewers, 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.

Edited by: S. Valcke
Reviewed by: two anonymous referees

References

Adler, R. F., Huffman, G. J., Chang, A., Ferraro, R., Xie, P.-P., Janowiak, J., Rudolf, B., Schneider, U., Curtis, S., Bolvin, D., Gruber, A., Susskind, J., Arkin, P., and Nelkin, E.: The Version-2 Global Precipitation Climatology Project (GPCP) monthly precipitation analysis (1979–present), J. Hydrometeorol., 4, 1147–1167, 2003.

Alexander, K. and Easterbrook, S. M.: The software architecture of climate models: a graphical comparison of CMIP5 and EMICAR5 configurations, Geosci. Model Dev., 8, 1221–1232, doi:10.5194/gmd-8-1221-2015, 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, Springer Berlin Heidelberg, 33–41, 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, 24–27 May, 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. N., 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., doi:10.5194/gmd-2016-197, in review, 2016.

Bozzo, A., Pincus, R., Sandu, I., and Morcrette, J.-J.: Impact of a spectral sampling technique for radiation on ECMWF weather forecasts, J. Adv. Model. Earth Syst., 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, IEEE Press, 819–829, 2014.

Dahan-Dalmedico, A.: History and Epistemology of Models: Meteorology (1946–1963) as a Case Study, Arch. Hist. Exact Sci., 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, Int. J. High Perform. C., 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, Philos. T. Roy. Soc. A., 372, doi:10.1098/rsta.2013.0276, 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, EGU General Assembly, Vienna, p. 13243, 2014.

Freidenreich, S. and Ramaswamy, V.: A New Multiple-Band Solar Radiative Parameterization for General Circulation Models, J. Geophys. Res.-Atmos., 104, 31389–31409, 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, 45–62, 2014.

Gates, W. L.: AMIP: The Atmospheric Model Intercomparison Project, B. Am. Meteorol. Soc., 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 Directive, IEEE Press, 55–61, 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. Weather 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, B. Am. Meteorol. Soc., 97, online first, doi:10.1175/BAMS-D-15-00135.1, 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., Hiller, J., Karp, S., Keckler, S., Klein, D., Lucas, R., Richards, M., Scarpelli, A., Scott, S., Snavely, A., Sterling, T., Williams, R. S., and Yelick, K.: Exascale computing study: Technology challenges in achieving exascale systems, Tech. Rep. 15, DARPA Information Processing Techniques Office, 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, Jpn. J. Appl. Phys., 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, J. 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, Q. J. Roy. Meteor. 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. Weather Rev., 128, 876–887, 2000.

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

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

Pincus, R. and Stevens, B.: Monte Carlo spectral integration: A consistent approximation for radiative transfer in large eddy simulations, J. Adv. Model. Earth Syst., 1, online only, doi:10.3894/JAMES.2009.1.1, 2009.

Platzman, G. W.: The ENIAC Computations of 1950 – Gateway to Numerical Weather Prediction, B. Am. Meteorol. Soc., 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, vol. 2, 79–91, 1997.

Schwarzkopf, M. D. and Ramaswamy, V.: Radiative effects of CH_4, N_2O, halocarbons and the foreign-broadened H_2O continuum: A GCM experiment, J. Geophys. Res.-Atmos., 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, 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., and Guo, H.: Uncertainty in model climate sensitivity traced to representations of cumulus precipitation microphysics, J. Climate, 29, 543–560, 2016.