Coarse-grained component concurrency in Earth System modeling

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 GPUs and MICs, are based on 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. Within the GFDL Flexible Modeling System, the atmospheric radiative transfer component has been configured to run in parallel with 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 the high end of the market also being dominated by clusters built out of mass-market commodity parts.

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 (Stocker et al., 2013, see e.g the iconic Figure 1.4 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 arithmetic and logic itself has stalled. 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. Whether we will continue to demonstrate continued increases in resolution and complexity through this transition remains to be seen.

ESMs 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. 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.

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 "exascale computing", 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. This purely expedient choice of timestep

5 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

10 modest computational cost in terms of allocated processors.

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

15 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

20 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,

25 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

30 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 va-

5   riety 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 run on $\mathcal{O}(10^5)$ distributed

10   processors and $\mathcal{O}(10)$ shared-memory execution threads, which places them among the most successful HPC applications in the world today (Balaji, 2015).

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

15   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:

$$
\begin{aligned}
A^{t+1} &= A^t + f(A^t, O^t) && (1) \\
O^{t+1} &= O^t + g(A^{t+1}, O^t) && (2)
\end{aligned}
$$

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

20   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.

$$
\begin{aligned}
25 \quad A^{t+1} &= A^t + f(A^t, O^t) && (3) \\
O^{t+1} &= O^t + g(A^t, O^t) && (4)
\end{aligned}
$$

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

30   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.[1] 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 Section 5 we will revisit this issue in the context of future work.
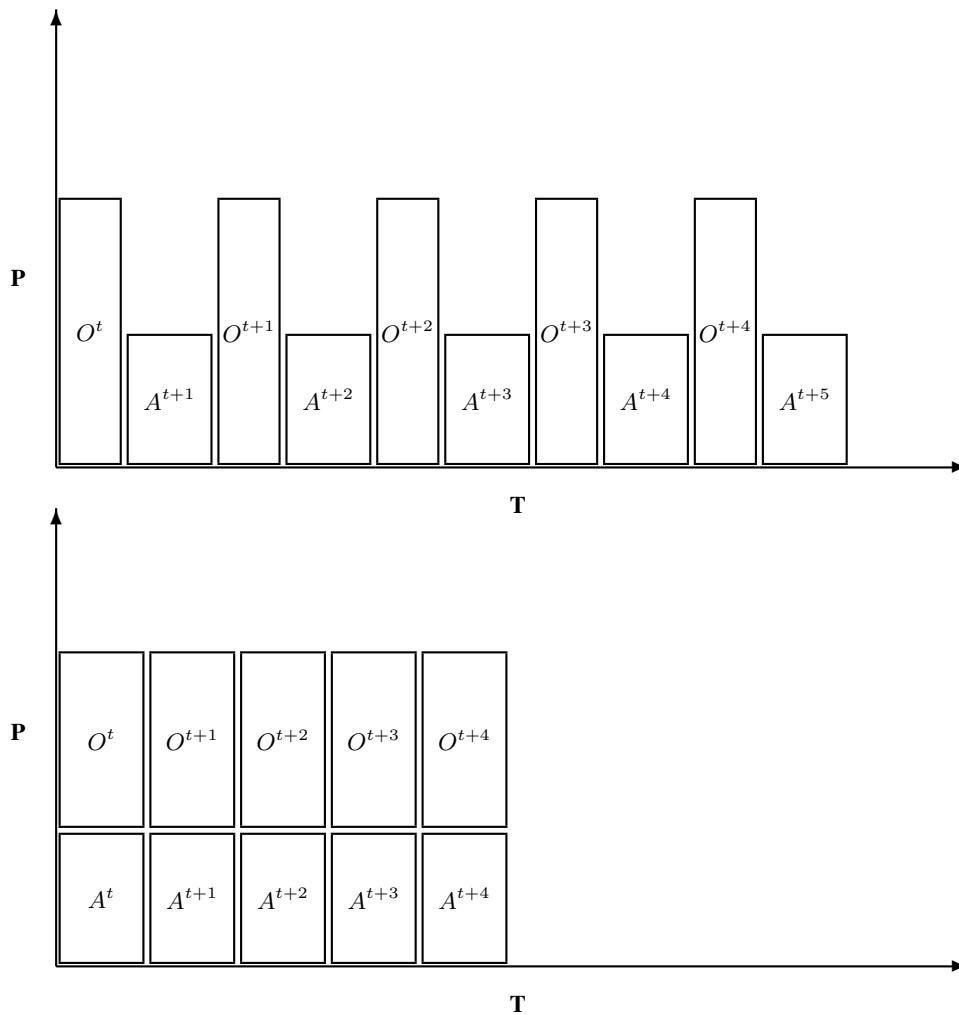
Figure 1. Serial and concurrent coupling sequences, with time on the X-axis and processors on the Y-axis. In the serial case, both components may not scale to the same processor count, leaving some processors idle. Note that in the concurrent coupling sequence below, O^{t+1} only has access to the lagged state A^t.

---

[1] 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.

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

5 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).
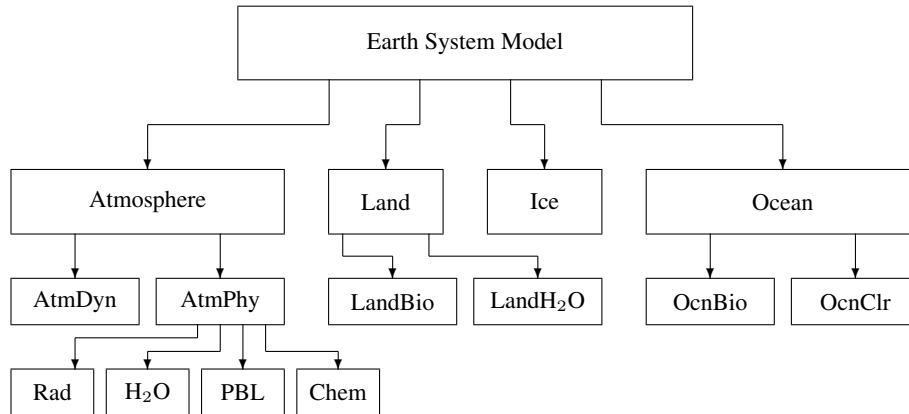
Figure 2. Notional architecture of an Earth System Model, with components embodying different aspects of the climate system, hierarchically organized.

10 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

15 timestep Δt_{cpld}, on the order of Δt_{cpld} = 3600 sec for its flagship application, decadal-centennial climate change. Within the slow coupling loop, the atmosphere communicates on a fast coupling timestep Δt_{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.
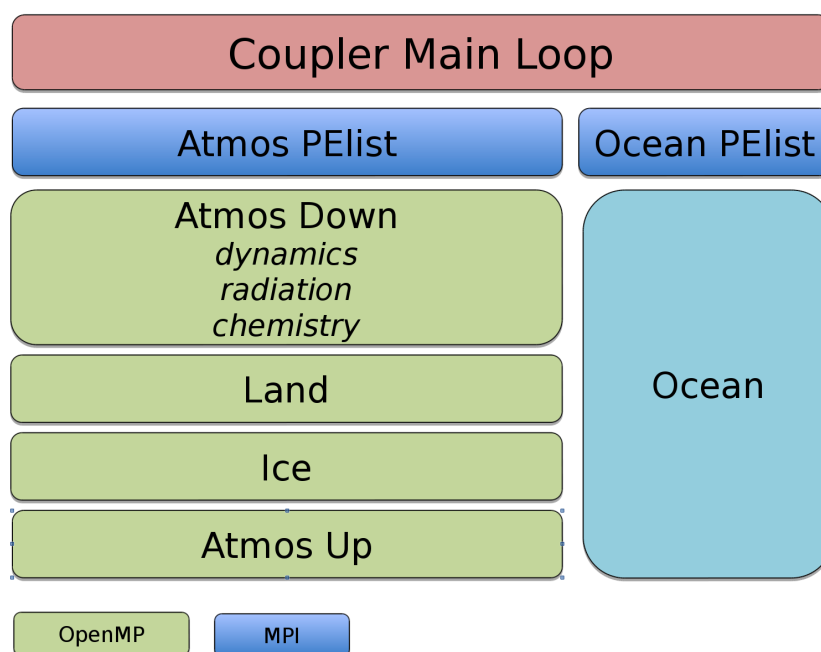
Figure 3. FMS parallel coupling architecture. *PElists* run concurrently in distributed memory, and stacked components execute serially. Components internally use a hybrid distributed-shared parallel programming model.

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.

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 short-wave 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 and clouds. The species shared between atmospheric physics, chemistry, and radiation are referred to as *tracers*, 3D model fields that are advected by the evolving dynamics, and participating in physics and chemistry processes at individual gridpoints.

5   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 $\Delta t_{\mathrm{rad}} = 10800$ seconds, or $9 * \Delta t_{\mathrm{atm}}$, 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 timestep Δt_{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 $\Delta t_{rad} - \Delta t_{atm}$ (~3 hours) behind. The

10   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 Δt_{rad} and Δt_{atm}, and bring us toward more physical consistency between the radiative and physico-chemical atmospheric

15   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). Coupling fields are transferred between

20   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 parallelism using OpenMP, where the columns within the distributed domain decomposition

25   are further organized into *blocks*. 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 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.

30   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
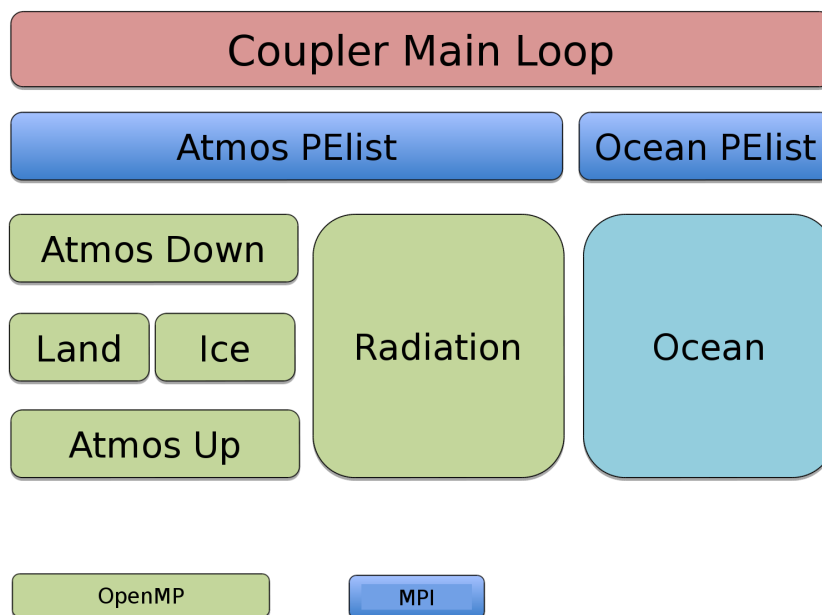
Figure 4. Concurrent radiation architecture.

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 between the radiation code and the rest.

4 Results

4.1 Results from AMIP runs

5 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
10 three experiments described here are:

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

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

3. concurrent radiation (CONCUR) also using $\Delta t_{rad} = \Delta t_{atm}$ = 1200 s. The difference between the SERIAL and CON-CUR 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 $\Delta t_{rad}/\Delta t_{atm} = 9$, 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 Δt_{rad} is longer than Δt_{atm} all solar fluxes are rescaled every Δt_{atm} by normalizing by the incident solar radiation using the zenith angle appropriate for that that atmospheric time step. Any sensitivity to the Δt_{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.

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 v2.2 (Adler et al., 2003) climatology. There is very little difference in the climatology of the model runs as the details between these runs are remarkably similar. 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. 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 $\Delta t_{rad} = \Delta t_{atm}$. The discrepancies are small enough that we believe them to be within the margins of the tuning process.
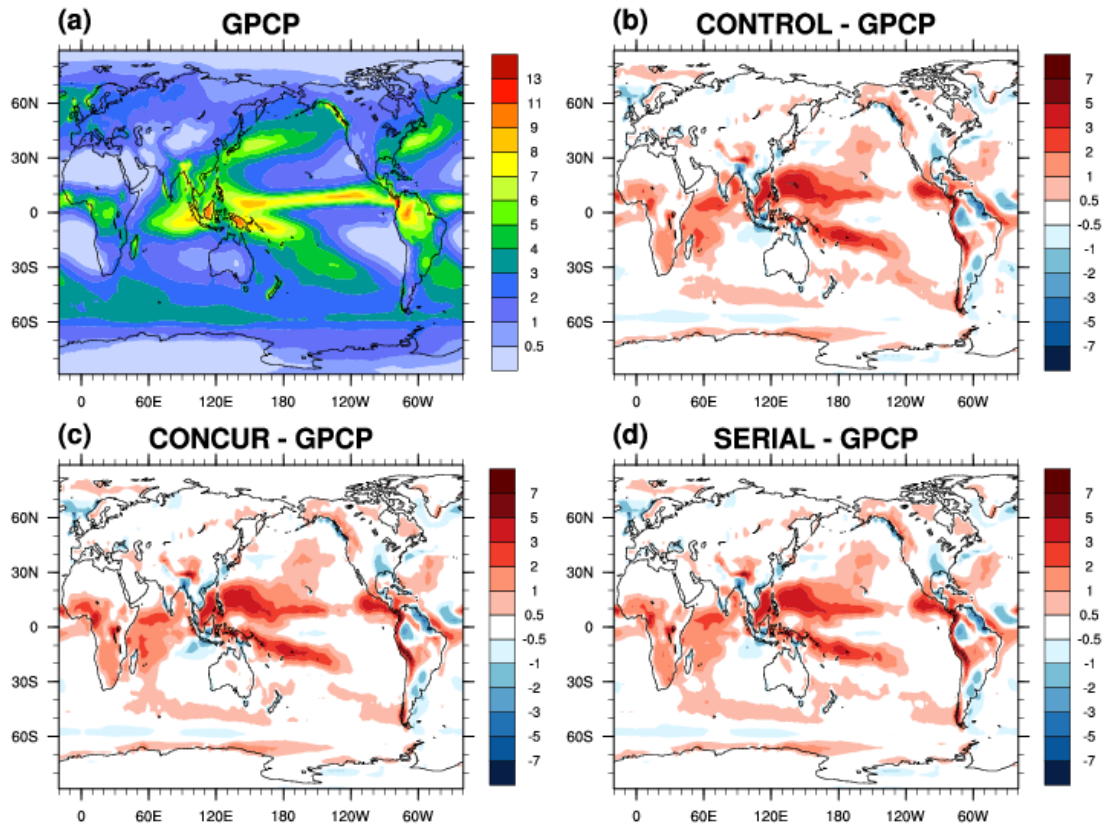
Figure 5. Comparison of model climatologies against GPCP precipitation. (a) shows the GPCP climatology, and (b), (c) and (d) show model climatological biases for CONTROL, CONCUR and SERIAL runs respectively.

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 $\Delta t_{rad} = 9 * \Delta t_{atm}$ 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 $\Delta t_{rad} = \Delta t_{atm}$, but at a higher processor count.

Initial studies were performed on a machine configuration that used AMD Interlagos processors on Cray's Gemini high-speed interconnect. All runs use the optimal processor/thread layout for a given PE count. As Table 1 shows, CONTROL
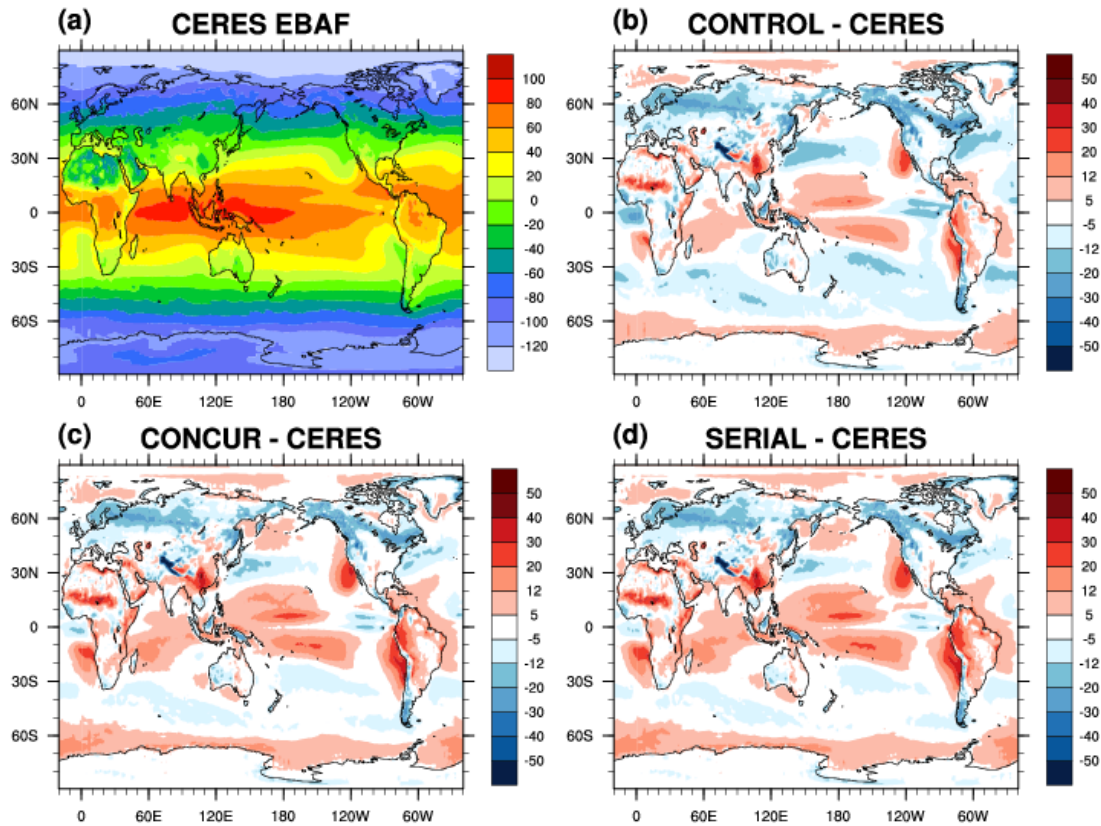
Figure 6. Comparison of model climatologies against CERES EBAF v2.8 climatological top-of-atmosphere radiation budget. (a) shows the CERES climatology, and (b), (c) and (d) show model climatological biases for CONTROL, CONCUR and SERIAL runs respectively.

achieved 9.25 SYPD. The SERIAL configuration shows the relative cost of radiation to the rest of the model, as shortening Δt_{rad} from 10800 s to 1200 s 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 brings us back to 9 SYPD. Thus, one can achieve the goal of setting $\Delta t_{rad} = \Delta t_{\mathrm{atm}}$ without loss in time to solution (SYPD), at a cost of 1.6X in resources (measured in compute-hours per simulated year, or CHSY). 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.

Configuration	$\Delta t_{rad}/\Delta t_{atm}$	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

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 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 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).

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 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 $\mathcal{O}(10)$ such components in contemporary ESMs.

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 architec-
5 tures with a wide ($\mathcal{O}(100\text{-}1000)$) 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
10 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 compo-
nents might be formally unstable. We are exploring more advanced time-coupling algorithms, including 3-time-level schemes
15 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
20 novel, massively-concurrent HPC architectures.

6 Source code and data availability

Source code and 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.
25   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 recommen-
dations 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.

5 Alexander, K. and Easterbrook, S.: The software architecture of climate models: a graphical comparison of CMIP5 and EMICAR5 configurations, Geoscientific Model Development Discussions, 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.

10 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.

Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., and McDonald, J.: Parallel Programming in OpenMP, Morgan-Kaufmann, Inc.,
15 2001.

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

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.

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.,
20 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.

25 Durran, D. R.: Numerical Methods for Wave Equations in Geophysical Fluid 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
30 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,
35 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, Monthly Weather Review, 141, 283–306, 2013.

5 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.

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.

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

10 Pauluis, O. and Emanuel, K.: Numerical instability resulting from infrequent calculation of radiative heating, Monthly weather review, 132, 673–686, 2004.

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

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

Stocker, T., Qin, D., Plattner, G., Tignor, M., Allen, S., Boschung, J., Nauels, A., Xia, Y., Bex, B., and Midgley, B.: IPCC, 2013: climate change 2013: the physical science basis. Contribution of working group I to the fifth assessment report of the intergovernmental panel on climate change, 2013.

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

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.