# GEOS Atmospheric Model: Challenges at Exascale

William M. Putman and Max J Suárez

January 31, 2017

## 1 Introduction

The Goddard Earth Observing System (GEOS) model at NASA's Global Modeling and Assimilation Office (GMAO) is used to simulate the multi-scale variability of the Earth's weather and climate, and is used primarily to assimilate conventional and satellite-based observations for weather forecasting and reanalysis. In addition, assimilations coupled to an ocean model are used for longer-term forecasting (e.g., El Niño) on seasonal to interannual times-scales.

The GMAO's research activities, including system development, focus on numerous time and space scales, as detailed on the GMAO website, where they are tabbed under five major themes: *Weather Analysis and Prediction*, *Seasonal-Decadal Analysis and Prediction*, *Reanalysis*, and *Observing System Science*. A brief description of the GEOS systems can also be found at the GMAO website.

GEOS executes as a collection of earth system components connected through the Earth System Modeling Framework (ESMF). The ESMF layer is supplemented with the MAPL software toolkit developed at the GMAO, which facilitates the organization of the computational components into a hierarchical architecture. GEOS systems run in parallel using a horizontal decomposition of the Earth's sphere into processing elements (PEs). Communication between PEs is primarily through a message passing framework, using the message passing interface (MPI), and through explicit use of node-level shared memory access via the SHMEM (Symmetric Hierarchical Memory access) protocol.

Production GEOS weather prediction systems currently run at 12.5km horizontal resolution with 72 vertical levels decomposed into PEs associated with 5,400 MPI processes. Research GEOS systems run at resolutions as fine as 1.5 km globally using as many as 30,000 MPI processes. Looking forward, these systems can be expected to see a 2x increase in horizontal resolution every two to three years, as well as less frequent increases in vertical resolution. Coupling these resolution changes with increases in complexity, the computational demands on

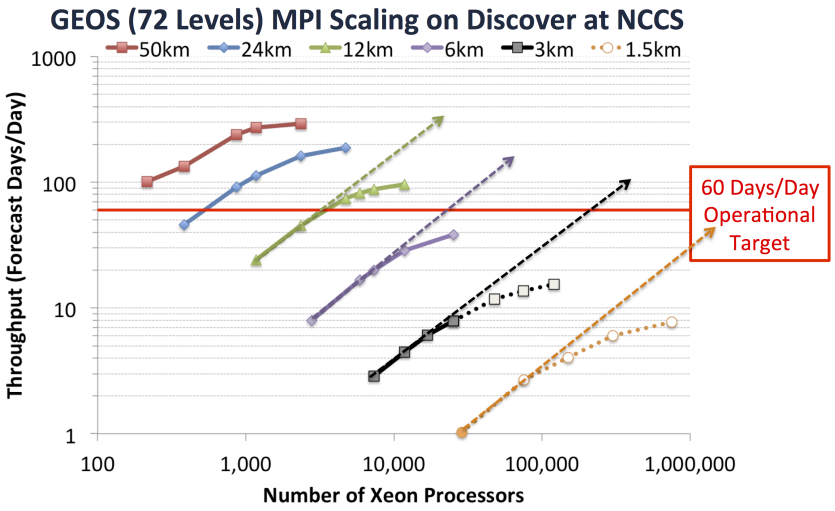


Figure 1: Scaling curves for the GEOS forecast model with increasing horizontal resolution. Solid curves and filled markers represent benchmark numbers obtained on the Discover supercomputer at the NASA Center for Climate Simulation (NCCS). Open markers and dashed curves represent extrapolated benchmarks based on coarser resolution scaling behavior. The dashed lines with arrows represent optimal scaling curves.

the GEOS production and research systems should easily increase 100-fold over the next five years.

Currently, our 12.5 km weather prediction system narrowly meets the time-to-solution demands of a near-real-time production system. Work is now in progress to take advantage of a hybrid MPI-OpenMP parallelism strategy, in an attempt to achieve a modest two-fold speed-up to accommodate an immediate demand due to increased scientific complexity and an increase in vertical resolution. Pursuing demands that require a 10- to 100-fold increases or more, however, would require a detailed exploration of the computational profile of GEOS, as well as targeted solutions using more advanced high-performance computing technologies. Increased computing demands of 100-fold will be required within 5 years based on anticipated changes in the GEOS production systems, increases of 1000-fold can be anticipated over the next 10 years (Table 1).

## 2 Current profiling efforts

The GEOS modeling systems regularly undergo component-level profiling at scale, using both internal timers and profiling tools. Community tools pro-

| Year | Horizontal Resolution (km) | Vertical Resolution (Layers) | Times Step (seconds) | Cores | Scaling Factor |
|---|---|---|---|---|---|
| 2017 | 12.50 | 72 | 450 | 5,400 | 1 |
| 2018 | 12.50 | 132 | 225 | 19,800 | 4 |
| 2019 | 9.00 | 132 | 120 | 148,500 | 28 |
| 2020 | 9.00 | 132 | 120 | 148,500 | 28 |
| 2021 | 9.00 | 210 | 120 | 236,250 | 44 |
| 2022 | 9.00 | 210 | 120 | 236,250 | 44 |
| 2023 | 6.00 | 210 | 60 | 1,890,000 | 350 |
| 2024 | 6.00 | 210 | 60 | 1,890,000 | 350 |
| 2025 | 6.00 | 300 | 60 | 2,700,000 | 500 |
| 2026 | 6.00 | 300 | 60 | 2,700,000 | 500 |
| 2027 | 3.00 | 300 | 30 | 21,600,000 | 4,000 |

Table 1: An estimated scaling in compute capability required by the GEOS operational assimilation and forecast system given anticipated horizontal and vertical resolution increases and changes in model timestep. The scaling estimated begins with the current production resolution of the GEOS system in 2017.

vide some limited capability to provide memory and computational profiling at a finer-grain, but rarely at scale, and even then with limited capability. At present, the GMAO is working on implementing a routine-level internal profiling tool that uses common process-level information stored in internal Linux files during model's execution. This tool should provide detailed routine-level profiling information for the GEOS modeling systems executing at scale.

Component-level profiling tools work well for the atmosphere and ocean modeling components of the GEOS system; but they cannot profile the full data assimilation system (DAS), which runs as a collection of separate executables (submitted in separate batch jobs) synchronized by task managers. Currently, the GMAO makes use of log files written throughout the DAS execution to piece together a rough profile of the DAS execution. A detailed profiling tool for the full DAS system needs further development.

Figure 2 shows a component-level profile of the atmospheric model at 12.5 km resolution. At this resolution, the finite-volume cubed-sphere dynamical core (FV3) accounts for about 50% of the computing profile of the GEOS atmospheric component. FV3 also drives the underlying MPI decomposition for GEOS through the Flexible Modeling System (FMS). FV3 has been developed at NOAA's Geophysics Fluid Dynamics Laboratory (GFDL) in close collaboration with the GMAO since the late 2000s. FV3 uses a hybrid MPI-OpenMP parallelization approach to achieve suitable scalability on existing Intel Xeon based computing clusters. FV3 has proven to be a highly scalable dynamical

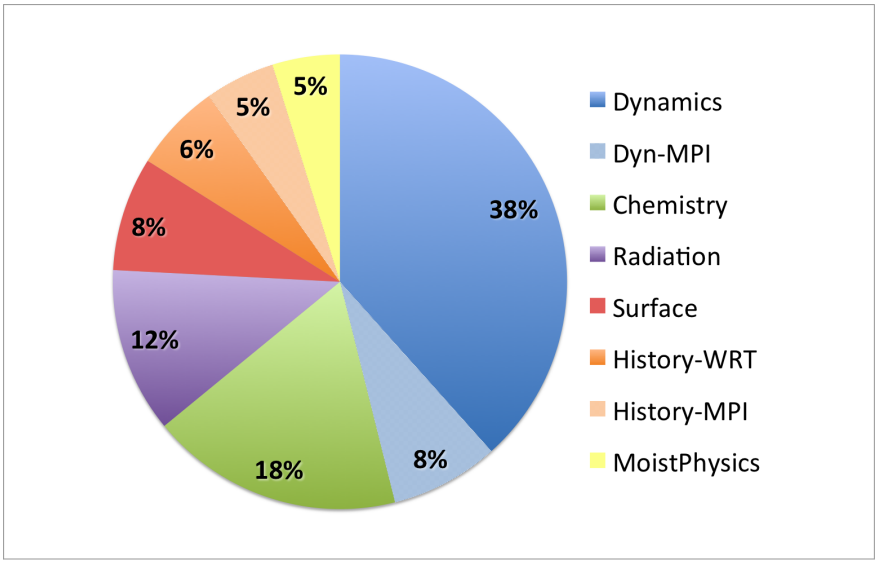


Figure 2: The computing profile of the 12.5km GEOS production atmospheric forecast system on 5,400 Intel Xeon-Haswell cores. The total execution time for a 10-day forecast in this configuration is 3 hours and 15 minutes.

core suitable for efficient application in numerical weather prediction at scales as fine as 3km globally.

The Goddard chemistry aerosol radiation and transport model (GOCART) accounts for another 18% of the GEOS profile (this does not include transport of the species, which is accounted for in the dynamics time).

The Chou-Suarez longwave and shortwave radiation components consumes 12% of the total profile. Under current plans the Chou-Suarez radiation parameterizations soon will be replaced by optimized versions of the Rapid Radiative Transfer Model for GCMs (RRTMG) developed at AER (Iacono et al. 2008, Mlawer et al. 1997, Iacono et al., 2000, Clough et al., 2005). The remaining 24% of the profile is consumed by the land surface component (8%), moist processes (5%) and the communication and writing associated with output history files throughout execution (11%).

The profile of the GEOS seasonal forecast system at 50km horizontal resolution for both the atmosphere and ocean components reveals the weight of ocean and moist physics at coarser resolution. Here the moist physics now consumes 26% of the profile while the radiation and chemistry parameterizations drops to just 9% and 3% respectively. The MOM ocean model and FV3 dynamics share nearly equally the remaining 51% of this GEOS configuration.

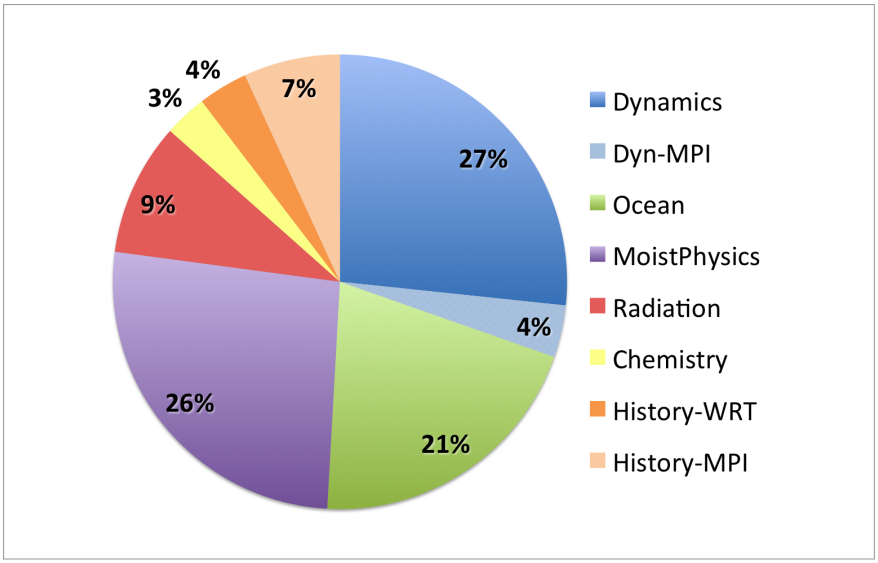


Figure 3: The computing profile of the 50km atmosphere 50km ocean GEOS production seasonal forecast system on 100 Intel Xeon Haswell cores.

# 3 Exploring Emerging Architectures and Parallelism

The GEOS atmospheric model currently has the capability of executing in a hybrid mode on GPU co-processors. This has been accomplished at a component level throughout GEOS using direct CUDA interfaces and some directive based implementations in the FV3 dynamical core. Having pursued this GPU implementation at a component level requires the transfer of data from CPU to GPU at the interface of each component. At this time, this memory transfer limits the effectiveness of the GPU implementation of GEOS, preventing it from being suitable for a production application. A very limited analysis of Intel's Many Integrated Core (MIC) architecture has seen limited success. Further exploration of GPU and MIC at the GMAO remains on hold awaiting developments in these architectures.

# 4 Targeted Optimizations and Developments

## 4.1 Near-term:

-Identify major bottlenecks to the throughput of the production GEOS configuration and streamline components to support a 1.5 to 2x improvement in throughput. This can be accomplished with the current profiling tools and an extensive evaluation of inefficiencies in components like Surface and GOCART. Furthermore, the FV3 dynamical core supports 16-bit precision which could provide as much as a 40 percent improvement in the overall throughput of the FV3 dynamics and tracer advection.

-To improve the scalability of the FV3 dynamics it is necessary to expand the horizontal domain by decreasing the number of MPI processes. This requires using the freed processors to run Open-MP threads decomposing over the vertical. This would then require running the physics on a different number of processor than the dynamics, or extending the hybrid approach to the GEOS physics components. In the physics, this could not be done through vertical decomposition and would require implementing loop-level parallelism via OpenMP over the long horizontal dimensions (I,J). This hybrid capability would extend the scaling curve of the GEOS system by avoiding the decrease in compute to communication ratio inherent in the dynamics as the number of MPI processes grows reducing the size of the compute domain on each processor.

-There is the potential for avoiding implementing OpenMP directives throughout the physics by redistributing the MPI decomposition between the dynamics and physics. The cost of this redistribution will be evaluated to understand the impacts it may have on performance.

-A Co-model approach can be explored to allow certain components like the chemistry and radiation to execute in parallel with the dynamics on a separate set of nodes. This may provide some limited increase in scalability and overall throughput for GEOS, but with limited gains of 2x at best. The increased complexity of coupling with this approach may limit its usefulness.

## 4.2 Midterm:

-The development of a detailed internal profiling tool at the subroutine level will aid in identifying bottlenecks throughout the GEOS forecast model. The plan here is to use frequent access to Linux system files on a process level to engage hooks at the subroutine level throughout the GEOS system to measure the number of calls, execution time, floating point operations and other computational metrics and identify performance bottlenecks throughout GEOS.

-Development of a detailed profiling tool for the full DAS. The current structure of the DAS workflow makes extracting detailed profile information rather complex. System timers exist throughout the DAS but in separate timing files with complex dependencies due to the multiple executable and multiple batch jobs run throughout the workflow.

### 4.3 Long-term:

-An optimized workflow for the execution of the DAS. Once sufficient workflow profiles exist inefficiencies in the configuration will be identified and workflow optimization will be explored.

-Memory management for ensemble members to avoid costly reads throughout the workflow. The DAS system in an ensemble variational configuration requires the running of 32 or more ensemble members that write state variables out to disk, these states are then promptly read in by the central DAS component. This pattern of disk access will not scale well as the resolutions and number of ensemble members grows. A more efficient use of memory within the DAS workflow can be designed to avoid this frequent access to disk, but will require a substantial restructuring of the workflow.