Cloud Computing Efforts for the Weather Research and Forecasting Model

Jordan G. Powers,[a] Kelly K. Werner,[a] David O. Gill,[a] Yuh-Lang Lin,[b] and Russ S. Schumacher[c]

[a] National Center for Atmospheric Research, Boulder, CO

[b] North Carolina Agricultural and Technical State University, Greensboro, NC

[c] Colorado State University, Ft. Collins, CO

Corresponding author: Jordan G. Powers, powers@ucar.edu

Early Online Release: This preliminary version has been accepted for publication in *Bulletin of the American Meteorological Society*, may be fully cited, and has been assigned DOI 10.1175/BAMS-D-20-0219.1. The final typeset copyedited article will replace the EOR at the above DOI when it is published.

© 2021 American Meteorological Society

Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

ABSTRACT

The Weather Research and Forecasting (WRF) Model is a numerical weather prediction model supported by the National Center for Atmospheric Research (NCAR) to a worldwide community of users. In recognition of the growing use of cloud computing, NCAR is now supporting the model in cloud environments. Specifically, NCAR has established WRF setups with select cloud service providers and produced documentation and tutorials on running WRF in the cloud. Described here are considerations in WRF cloud use and the supported resources, which include cloud setups for the WRF system and a cloud-based tool for model code testing.

CAPSULE

The popular Weather Research and Forecasting (WRF) Model for atmospheric simulation now has supported capabilities for utilizing cloud computing environments.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.

Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

1. Introduction

Cloud computing is the use of remote computer systems via the internet, and in the context of numerical weather prediction (NWP) encompasses the generation of atmospheric simulations. Cloud computing has exploded over the past decade, with the market served by big enterprises with broad portfolios such as Amazon, Google, and Microsoft, as well as a host of newer, cloud-focused firms such as Scala Computing, Rescale, and Penguin Computing. The growing cloud demand includes the running of compute-intensive Earth-system models, such as those for weather, air chemistry, climate, and ocean circulation (see, e.g., Chen et al. 2017; Zhuang et al. 2019; Coffrin et al. 2019). In addition, the cloud availability of datasets useful for atmospheric modeling is increasing, in settings such as NOAA's Big Data Program (Ansari et al. 2018), supported by the cloud service providers (CSPs) Amazon Web Services, Google Cloud Platform, and Microsoft Azure.

The Weather Research and Forecasting (WRF) Model (Skamarock et al. 2019; Powers et al. 2017) is one such application increasingly run in the cloud. This system has been built for both meteorological research and real-time forecasting and could be considered the world's most popular NWP model (Powers et al. 2017).[1] The National Center for Atmospheric Research (NCAR) supports WRF to a worldwide community consisting of users in universities, research labs, operational centers, and commerce. The WRF program provides user assistance, developer guidance, tutorials, workshops, and code releases.

[1] The cumulative number of WRF user registrations is over 54,000, representing over 162 countries, and the interest level in the model is reflected in user registrations recently averaging over 4,000 annually.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

In light of the increasing reach of cloud computing, the meteorological community's push to run WRF in the cloud, and NCAR's responsibility for model support, the WRF effort has assembled resources for model users and developers to exploit cloud environments. The purpose of this article is to present these capabilities, which we refer to as "Cloud WRF". Detailed below, the materials consist of WRF system cloud setups, an online Cloud WRF tutorial, and a cloud-based capability for testing WRF code.

We note that there have been a number of publications exploring WRF's operation and performance in the cloud (Molthan 2015; McKenna 2016; Siuta et al. 2016; Duran-Limon et al. 2016; Goga et al. 2018). A basic finding is that the cloud can be effective, reliable, and affordable for running the system (e.g., Chui et al. 2019). Thus, as the viability of WRF in cloud compute environments has been established, our focus is on describing the cloud resources for WRF use and development that NCAR has positioned for the community.

2. Cloud Considerations with WRF

Before describing the Cloud WRF components, we summarize considerations for users contemplating running WRF in the cloud. Cloud computing can present a new environment and new issues to atmospheric modelers, with cost foremost among these.

In terms of compute settings, WRF can operate on a range of UNIX/LINUX platforms from laptops to massively-parallel, high-performance computers (HPCs). Whatever the platform, the compute requirements for a WRF job (e.g., processor and memory requirements) are functions of the model configuration (e.g., grid spacings and domain dimensions) and production timing needs. In the cloud setting, grid configurations, simulation time constraints, and the true costs

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

of local computing all factor into whether cloud computing offers pricing or performance superiority to traditional, on-premise computing.

Compute advantages of the cloud are: the availability of powerful, flexible resources without responsibility for the systems; extensible data storage; updated hardware, software, and workflow tools; accessibility; and customer support. For any entity, computing systems are capital acquisitions that depreciate, while presenting maintenance and management costs. In contrast, the cloud offers users compute resources without direct expenditures for hardware purchase, system upkeep, and persistent staffing. Of course, CSPs see such costs and ultimately impose them on users at some level. Thus, there is a point at which users' cloud computing outlays— that implicitly have these cost elements— will surpass the costs that accurately reflect their access to and support of on-premise computing systems. However, users pay for resources on the cloud only as they need and consume them.

The cloud also reflects a competitive, agile marketplace, which can benefit users in ways institutional facilities might not. CSPs update their hardware and software environments and their development and workflow tools continuously. Their pairing of the latest architectures with support capabilities can optimize compute performance for an individual's application, increasing a user's productivity. Furthermore, CSP customer service can provide users the levels of tailored assistance needed without long-term investment in system administration.

3. The WRF Model and Cloud Computing

a) WRF Background and Model Support

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

The WRF modeling system has proven to be an adaptable platform and has been tailored for applications such as atmospheric chemistry (WRF-Chem; Grell et al 2005; Fast et al. 2006), wildland fire (WRF-Fire; Coen et al. 2013), and hydrological processes (WRF-Hydro; Gochis et al. 2015). NCAR's Mesoscale and Microscale Meteorology Laboratory (MMM) runs the WRF user support program, having foci of user help, system tutorials, and code oversight. MMM manages the WRF codeset and assists developers in integrating their contributions. The WRF repository is maintained with the software version control system Git (Chacon and Straub 2014) and is hosted on GitHub.[2] WRF is a community model, and code contribution is open to all; however, developers are required to conduct testing on their contributions to ensure proper builds, bit-for-bit parallel reproducibility, and codeset integrity.

Cloud capabilities are facilitating these WRF community support functions. The cloud serves as a shared environment for troubleshooting user problems, and cloud accessibility and resources are providing a better environment for WRF training. In addition, for model maintenance and development, the cloud has addressed a previous bottleneck in code testing. For this, a new cloud-based tool for conducting WRF code regression tests (Sec. 4d) now handles effectively the volume of jobs in the multiplex testing workflow.

b) Cloud Computing

1) Cost Considerations

The cloud can serve processing needs while avoiding certain costs and responsibilities attending on-premise systems. The strategy, however, is not free: it is simply a pay-as-you-go

[2] The public WRF repository may be found at: https://github.com/wrf-model/WRF.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

approach, the cost-effectiveness of which will vary for each user. For example, most in academia and government have access to on-premise compute resources, making cloud computing a new expense whose justifiability may not be immediately apparent. Nonetheless, the cloud may offer options and capabilities that such "free" computing does not provide, such as more compute power or fewer scheduling constraints. And, for users who do pay for on-premise computing, there are aspects of the cloud that can make it the better-priced option: they only pay for the resource amounts used, such as those for compute time and data storage/transfer; they avoid support and depreciation costs of their own physical assets, whether used or idle; and they have access to the latest in hardware, software, and operating environments.

The charges one can expect for using WRF in the cloud mainly come from computing resource usage and data resource usage. The computing cost is based on the extent and duration of the hardware engaged for a job, and the cost is modulated by variations in core processing and node interconnect speeds for one's virtual machine. As an example of performance sensitivity to platform type, Chen et al. (2017) showed that in a comparison with that of an on-premise HPC, cloud operation of the Community Earth System Model (CESM; Hurrell et al. 2013) was marked by performance ceilings for certain core counts, due to the lesser bandwidth of the cloud system's interconnect. This is one example illustrating that a user's best answer to the compute cost-effectiveness of cloud vs. on-premise resources may need to come from system trials of their specific application.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

It is important to recognize that virtually all aspects of cloud computing activity can be charging points: storage, access, data egress, compute cycles, and even idle time. A virtual machine accrues charges for all of the time it is engaged. Thus, if a job is initiated and is either not progressing or is not terminated when completed, charging continues. Depending on the size of the virtual machine, costs for such unintended use can run in the thousands of dollars over a few days. Thus, both novice and experienced cloud users must be vigilant.

Lastly, rates for data occupancy vs. data transfer vary among CSPs. Some may present lower billing rates for data occupancy, but impose higher ones for transferring data from their space. One tactic to address this is to analyze voluminous model output in-situ in the cloud, offloading only results or derived products.

2) Atmospheric Model Cloud Computing Experiences

To date, the literature on cloud computing for atmospheric modeling has concentrated on cloud use for real-time systems, with WRF a recurring example. Molthan et al. (2015) debuted details of running a WRF forecasting system on Amazon Web Services (AWS), finding the cloud an attractive compute option. Siuta et al. (2016) ran an operational WRF system on the Google Cloud Platform, concluding it an economically-viable replacement for their on-premise system. McKenna (2016) ported a coupled Earth modeling system to the AWS cloud for regional real-time prediction. This system linked WRF to the ROMS (Regional Ocean Modeling System) ocean model (Shchpetkin and McWilliams 2005) and the SWAN (Simulating Waves Nearshore) wave model (Booij et al. 1999). For this application, the cloud increased real-time robustness and efficiency and improved their development workflow.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Similar advantages were noted by Chen et al. (2017) in running the climate model CESM. They found cloud implementation to be cost-effective and to scale well with increasing core counts, with ultimate performance comparable to that of a tested HPC. They cautioned, however, that for multi-node virtual machines one's model parallelization configuration should be analyzed to confirm optimization of the setup applied. On that issue, Zhuang et al. (2020) investigated cloud jobs using up to 1152 processors for running the NASA GEOS-Chem air chemistry model (Bey et al. 2001). They found compute performance and cost-effectiveness for implementations on that compute scale to be comparable to running on HPCs, but recognized that cost-effectiveness must ultimately be determined on a user-specific basis, being a function of the user's priorities (e.g., time to run completion).

Chui et al. (2019) explored the sensitivity of the costs of running WRF to two factors: data egress and job prioritization. Regarding the former, they noted that compressing WRF output to decrease the volume of data offloaded can significantly reduce transfer charges. Regarding job prioritization, they tested cloud options for "preemptible" resources offering lower price points. In this mode, one's virtual machine resources can be taken over by jobs with higher priority. Because preemption terminates one's job, the option has obvious disadvantages. Addressing this, however, Chui et al. invoked the WRF restart capability to enable job resumption when resources re-emerged. Thus, their simulations could survive occasional interruptions in the preemptible queues. While this approach is only possible for time-insensitive workflows, many research applications could fit the bill.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Another cost-reduction approach is to link cloud resource use to rate thresholds and exploit spot instance pricing. This strategy is based upon compute charging by a CSP varying with its current load: CSPs may offer a temporary "spot" price lower than the normal "on-demand" price, i.e., the price charged for the fulfillment of a compute order immediately on request. While spot-thresholded jobs can be cheaper, they are on standby until the current spot price drops to the user's level. Furthermore, they may be subject to resource preemption. Nonetheless, the spot approach may return lower-cost jobs for those able to wait and tolerate interruptions (see, e.g., Coffrin et al. 2019; Zhuang et al. 2020).

In summary, explorations like those of Chen et al. (2017) and Chui et al. (2019) show that a general conclusion cannot be made as to whether for WRF cloud computing is consistently better than on-premise computing. Importantly, however, they do show that the flexibility in the WRF system for structuring simulations makes finding a competitive cloud solution likely.

4. Cloud WRF Capabilities

a. Basic Cloud Use and Supported WRF Setups

To prepare for cloud use, the first step is to engage a CSP and establish an account. This is the user's responsibility, even for the WRF materials described here. The next step is to set up one's job environment. Compared to WRF on-premise operation, running Cloud WRF has extra setup details. Users must choose a machine type and the type of "instance", which is a single setup of a cloud virtual machine and its environment for an application.[3] The user must also create a public "key"— an encrypted credential— to provide secure shell access to the instance.

[3] Machine configurations encompass the operating system and compute platform class, and the environment setup encompasses the compute node count, storage devices, and software stack.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

NCAR-installed Cloud WRF setups are currently available on two CSPs: Amazon Web Services (AWS) and Scala Computing. WRF has been ported to these platforms with its supporting environment.[4] We stress that while NCAR has positioned Cloud WRF setups in these environments, the CSPs charge for use of their resources, and paying for an account from these or other providers is the user's responsibility.

The Cloud WRF materials consist of system code and static input data. The supported environments are built with GNU Fortran (GFortran) compilers, which are free to the public and may be distributed under the GNU General Public License. Because NCAR cannot distribute proprietary software, if such a compiler, such as one of Intel or NVIDIA, is desired, users must upload their personal or institutional license to the CSP environment or otherwise acquire the package.[5] In the set-up cloud environments, all required libraries are installed, as is a version of the GNU compiler. While the NCAR materials describe the procedures for building the libraries and WRF code, users may also use pre-configured environments, with bundled WRF binaries. For reference, Fig. 1 presents a diagram of the components in Cloud WRF. WRF and WPS (WRF Preprocessing System) are available with the supported CSPs for the latest major version release, as well as for a number of older ones. In the AWS environment, users can also run the WRF Data Assimilation (WRFDA) system. NCAR's WRF support group can

[4] For information on AWS and Scala, see either https://aws.amazon.com or https://scalacomputing.com. For documentation on Cloud WRF, see links under the main WRF users' page: https://www2.mmm.ucar.edu/wrf/users.

[5] As of this writing, Intel offers for free download its *oneAPI* toolkit that is a package including compilers and other products. NVIDIA offers the NVIDIA HPC SDK package: https://developer.nvidia.com/hpc-sdk.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

address user inquiries regarding Cloud WRF materials in the established AWS and Scala environments.

b. Using Cloud WRF on AWS

The Cloud WRF setup on AWS is maintained on the AWS Elastic Compute Cloud (EC2) and packaged in the form of Amazon Machine Images (AMIs). These are configured with installed WPS and WRF code on instances running the Amazon Linux AMI 64-bit operating system. Images allow users to save and share their setups, making the remote workspaces and workflows function like those on traditional computers. The Cloud WRF images are available from a given AWS regional endpoint, the US East/Northern Virginia location, but users can copy them to another AWS region to work in if desired.

For input atmospheric data, AWS provides access to real-time output from NCEP's Global Forecasting System (GFS) (NOAA 2003; Harris et al 2020) that can be used for WRF initialization and boundary conditions. However, for simulating historical cases, users should expect to have to obtain the background inputs themselves.

The NCAR Command Language (NCL) and Read Interpolate Plot (RIP) postprocessing/graphics tools are included in the AWS image. For model output visualization, the NetCDF "ncview" capability for NetCDF-formatted files and the X11 window system are installed. These tools eliminate the need for users to transfer volumes of WRF output to their local systems in order to generate and view imagery, as data egress is an important cost

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

consideration. Specifics on the AWS WRF environment and running executables are described in the packaged instance, as well as in the online model tutorial.

c. Using Cloud WRF on Scala Computing

Instead of maintaining hardware itself, Scala Computing serves clients through accessing the compute infrastructures of other CSPs. The Scala interface submits jobs to the provider determined optimal at the time, reflecting price and compute request. Users manage their own "projects", which are individual environments configured for their job type, and, through a set of commands from their local environments, users declare job specifications. Scala provides configured WRF environments, including installations of the compilers, libraries, WRF and WPS binaries, and static input data. Users running WRF only need to modify their namelists and scheduler scripts and to import meteorological data for each run. This setup is good for users repeating consistently-configured simulations, such as in a real-time WRF forecasting system.

For the Cloud WRF setup, the Scala Compute Platform provides a development environment currently coupled with an AWS cluster, using a CentOS instance. Scala provides NFS file systems for facilitating simulations and data storage which are mounted on a head node and accessed for the cluster's instances when a job is submitted. The Scala environment offers sample scripts for submitting WRF jobs, using a Slurm scheduler. Users define their cluster in terms of number of cores, amount of memory, and instance type. For visualization purposes, NCL is included, and the ncview and X11 utilities are installed for quick viewing of model output.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

d. Cloud-based WRF Code Testing Capability

WRF has grown over the years through code contributions from developers around the world (Tab. 1), with MMM overseeing the code testing and integration process. As Tab. 1 shows, recent years have seen a transition from the paradigm of WRF support group members shepherding code into the repository to one of external contributors acting independently. The process of preparation and implementation of new code by such contributors was being hampered by the NCAR community supercomputer's inability to handle the job load for regression testing of the WRF submissions. That framework executes tests to ensure that all model code compiles, that code changes and additions do not break other model elements, and that numerical results are bit-reproducible in both serial and parallel execution. The issues with running the testing framework on the HPC were that not only was the multitude of small test jobs launched by the framework incompatible with the HPC, and in particular its scheduler constraints, but also that to users without accounts on the NCAR machine, running the regression package was tough due to script complexity and lack of access to necessary data. The cloud, however, has provided an alternative, efficient solution.

The WRF support team now maintains a cloud-based utility for running automatic code tests. This uses the continuous integration software Jenkins[6] and employs Docker containers for a standardized environment that includes the directory structure, initial data, namelist options, run scripts, validation scripts, built libraries, and a compiler. The testing utility runs automatically for each proposed modification submitted via a GitHub pull request (PR) to the WRF

[6] https://www.jenkins.io/

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

repository, with the tests commenced upon the PR submission. The testing puts the source code through approximately 50 separate builds with approximately 200 short simulations spread across them, utilizing 20 cloud instances running the containers, and reporting results within 30 minutes. Exploiting cloud resource flexibility, this automated, reliable, and quick regression testing capability has eliminated the previous bottleneck caused by an HPC that was both inaccessible to most external contributors and was not designed to support the testing necessary for a continuous integration workflow.

The cloud testing capability can support a more distributed network of external code contributors. For example, in the preparation of the most recent WRF major release, more than 80 separate pull requests from external contributors were received, amounting to over 500 sets of regression tests. This shift in the open development for WRF enabled by cloud computing has significantly modified the release schedule. No longer are there periods where the repository is frozen to contributions. And, the period blocked out for testing of the release's tentative code has been greatly reduced as contributors now do the compatibility testing in advance, made possible by the accessibility of the testing harness. Furthermore, contributors no longer must rely on the availability of WRF support personnel to shepherd code inputs. In summary, due to the new cloud code testing capability, the WRF release workflow enables more contributors, can absorb more new developments, requires less staff time, and yields a more robust release.

e. WRF Computational Performance

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

To give an idea of cloud vs. HPC performance for WRF and to illustrate how high levels of cloud resources can be successfully applied for the model, we have run benchmarks using both AWS hardware and the HPC managed by NCAR for the geosciences community, named "Cheyenne". Our benchmarking[7] uses WRF Version 4.2 (Skamarock et al. 2019) configured with a single domain of 1500x1500 horizontal grid points and 50 vertical levels. The tests use increasing counts of compute cores on the HPC Cheyenne and a cluster of AWS nodes of designation "c5n.18xlarge".[8] Both machines have 36 cores/node, and processes are single-threaded for each core. WRF was built on both platforms with both GNU and Intel compilers invoking distributed-memory parallelism.

We present timing comparisons of the WRF benchmark for the two compilers for model integration timesteps only, as well as a benchmark for timing the output of history files. For both benchmarks, we obtained robust statistics using short simulations. The computational benchmarks were twenty timesteps long, and the I/O benchmarks were four time steps long.

First, Fig. 2 presents the computational timing results with the ratio of averages of elapsed wallclock times per WRF model timestep: this is a ratio defined as the time reported by the AWS cluster to the time reported by the HPC. Here the timing calculations are done for three variants of integration timesteps: the time for a model step with no radiation computation[9], the

[7] The WRF benchmark input data, validation data, configuration files, and validation script are available from https://www2.mmm.ucar.edu/wrf/src/benchmark_large.tar.gz.
[8] These instances ran the Intel Xeon Platinum 8000 series (Skylake-SP) processor with clock speed of up to 3.5 GHz. The c5n instances have up to 100 Gbps of network bandwidth and support AWS's Elastic Fabric Adapter (EFA) inter-node communication network interface, used for these tests.
[9] The radiation scheme is called periodically, and when called it entails more computation per model time step. In these benchmarks, the radiation scheme was called for every three minutes of forecast time.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

time for a model step with radiation computation, and the time for an average model step weighting the frequencies of the two.

For these non-I/O results, one test reflects WRF built with a GNU compiler (Figs. 2a,c), which is bundled in the packaged Cloud WRF materials, and the other uses WRF built with an Intel compiler (Figs. 2b,d). The latter would more typically be the choice for an HPC user, due to the Intel executable's better computational performance for WRF. One sees that the timings for both the radiation and non-radiation steps exhibit flat behavior for increasing processor counts, to 3600 cores for the GNU build and 1000 cores for the Intel (Figs. 2a,b). In this regime, the time ratios rely largely on the relative capabilities of the machines' chip performance and the volume of computation vs. communication. Since the GNU WRF executable is slower than the Intel executable, the fixed costs of communications are relatively smaller for the GNU runs. In addition, based on timing comparisons (not shown), the GNU build scales better than the Intel build, albeit due to the slower speed of the GNU executable. As the radiation timesteps have significantly more column-wise (i.e., non-communicated) computations, the radiation timestep curve (red) remains flatter for a greater core range than the non-radiation curve (blue), and this is the case for both compilers (Figs. 2a,b). With the number of processors increasing, the amount of computational work per process is reduced, meaning the nearly fixed cost for communication becomes more important for increasing core counts. This condition is delayed for the radiation steps and for the GNU-built executable.

For non-radiation steps and for the Intel WRF executable, the time taken by communication begins to exert its influence earlier with increasing core counts (Figs. 2c,d), as the

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

computational workload per processor is reduced and as the disparity of the interconnects of the AWS virtual machine and the NCAR HPC comes into play. Considering only computational efficiency (i.e., excluding I/O), the solution crossover point for this WRF benchmark is at about 7200 processors for Intel, and greater than 7200 processors for GNU (Figs. 2c,d). Thus, with this single-domain WRF benchmark case, the AWS cloud platform provides a faster time to solution for Intel through 7200 processors and GNU through 3600 processors. This corresponds to approximately 300 and 600 horizontal grid cells per MPI task, respectively, for Intel and GNU.

Figure 3 shows the comparisons of the times for outputting a non-compressed WRF history file on each machine during I/O timesteps (Intel build only). Here, the serial NetCDF4 library was used to output the data in each of the four 11-GB history files. It is seen that throughout the entire range of processor counts, the NCAR HPC outputs data to disk faster than the AWS machine. In the default output mode used here, all data are communicated to a single process for output, and as the number of processes increases, the total amount of time to output the data increases.

While the output timings in Fig. 3 reflect this single-file outputting approach, another approach available in WRF is to have each MPI process write its computational region's output to its own file, with such separate files later combined. This reduces output elapsed times, as each process writes a much smaller file, and the MPI processes avoid communicating each core's portion of the domain to another process for writing. Illustrating the timing differences for the two approaches, Tab. 2 lists the output times for the larger core count runs on each system for

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

single-file vs. split-file outputting. As expected, the split-file approach is faster, and the NCAR HPC shows greater output speed than the AWS platform.

5. Cloud WRF Applications

a) Cloud Support of WRF Tutorials

The WRF support group conducts two modeling system tutorials annually at the NCAR facility and typically delivers at least one abroad each year. The tutorials are time-consuming for the team with preparation of the compute environment, as practice materials must be installed and tested on an array of classroom machines. In addition, for venues abroad, the setup work involves more time and uncertainty due to obstacles encountered in configuring the unfamiliar hardware under greater security restrictions.

Reducing the time, cost, and risk with reliance on local computing, MMM has moved to the cloud for WRF tutorial compute needs. This has simplified tutorial management by providing globally-accessible compute environments enabling efficient setup. Machines no longer have to be individually prepared on-site, tutorial materials can be updated centrally at any time, and it is easier to maintain the practice environment. And crucially, WRF trainees have found the instructional cloud settings understandable and user-friendly. The sidebar presents examples of positive feedback on the use of Cloud WRF in training.

The cloud approach also helps those taking the online WRF tutorial, which otherwise requires users do the exercises on their own diverse hardware. That non-uniformity can present difficulties in installing or using necessary background elements such as libraries and

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

compilers, for example. New users that had undertaken the online tutorial were often unprepared to set up the complex environment required to build and run the WRF system. The fixed, accessible WRF cloud environment, however, removes these barriers, reducing new user frustration and accelerating learning.

b) University Classroom Use

Specialized tutorials on Cloud WRF are now given by the WRF support team. These have been delivered at NCAR, as well as at its partners North Carolina State Agricultural and Technical University (NCAT) and Colorado State University (CSU). The tutorials were attended by faculty and students, ranging from those new to WRF to those experienced with the modeling system, and they included WRF and cloud computing presentations followed by hands-on exercises via the AWS environment.

NCAT’s Cloud WRF tutorial students found the installation of WRF in AWS easy to use and noted the importance of flexibility in accessible compute power for their research needs. Those new to WRF benefitted from the introduction to the model and readily being able to work with it in the configured cloud environment, while the experienced WRF users saw how cloud computing could be tailored to their modeling projects. Overall, participants felt the cloud could become the platform of choice for WRF simulations and weather data analysis.

CSU's use of Cloud WRF was in a graduate-level mesoscale meteorology class that included an exercise on modeling convective storms. Afterward, students had a class lab assignment to use Cloud WRF to reproduce results from a study in the literature and then to design and run their

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

own experiment. Feedback was positive, in particular in the citing of new understandings of cloud differences from other computing environments and of the potential for the application of the cloud for their model use. The students found that configuring and running WRF remotely was straightforward and easy. Challenges reported were in analyzing and visualizing model output in the cloud and in transferring output to local computers, which are issues attending computing on any remote HPC system.

For exploration of the potential for, not only running WRF in the cloud, but for cloud computing in general, some CSPs offer credits to educational institutions for trial of their systems. As we wish to emphasize, hands-on trial is the way to determine the utility and cost-effectiveness of the cloud for one's research or teaching, and CSP educational credit offers can allow university personnel a way to get direct and free cloud experience. Moreover, to enable potential cloud users to get an idea of costs, CSPs provide online pricing calculators, and examples may be found on the AWS, Microsoft Azure, and Google Cloud Platform web sites.

6. Summary

NCAR has undertaken a Cloud WRF effort to advance the WRF system and serve the model's user community via the new paradigm of cloud computing. With the setups and tools created, the cloud provides accessible and flexible environments for model use, development, and instruction. For those wanting to apply WRF and lacking the resources to acquire and maintain their own compute hardware, the cloud and materials provided can be a viable solution.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

The primary supported Cloud WRF tools are model setups and documentation for running on the cloud service providers engaged. Accessing, configuring, and operating in their distinct workspaces differs, and through trial users can determine the CSP that is better for their workflows. The provided materials are the WRF source code, compiled model binaries, static input data, libraries, and postprocessors. Step-by-step instructions guide users through establishing entry, invoking instances, configuring virtual machines, creating images, transferring files, and running the WRF modeling system components.

To illustrate how WRF in the cloud can scale to large-machine configurations and to give an idea of cloud/HPC compute performance differences, we conducted benchmark runs of WRF configurations both on the community HPC maintained by NCAR and on an AWS virtual machine. The tests also assessed the wallclock time required for I/O. Considering only computational efficiency (i.e., excluding I/O) with two different compiler builds, the cloud platform provided a faster time to solution for machine configurations using up to 7200 processors with Intel and 3600 processors with GNU, with the HPC faster beyond those respective counts. In the analysis of I/O timing, it is found that the NCAR HPC outputs data to disk faster than the compared AWS virtual machine regardless of processor count. These test examples, however, do not speak to the variable cost dimensions of on-premise v. cloud computing. Those factors make it a responsibility of a given user to assess their application needs, production demands, and compute capital in performing a relevant cost-benefit analysis.

NCAR has also created a cloud-based WRF code testing capability to better support contributors making submissions to the WRF repository and to streamline the code

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

implementation path. With this, when contributors submit pull requests, the cloud utility automatically conducts the necessary WRF regression testing suite. This tool has simplified, strengthened, and accelerated the code integration process for WRF.

The Cloud WRF materials are also assisting atmospheric model training and meteorological education. They now support the regular WRF tutorials delivered by NCAR, and they provide new means for professors to enlist WRF in university curricula and research. Partner universities in this effort have successfully engaged their students in learning the system and have been enthusiastic in pursuing cloud applications.

Cloud computing capabilities are growing, and the cloud can offer advantages over traditional, on-premise computing: no capital investment and facility support costs; flexible, cutting-edge compute power; and elastic storage, to name a few. However, cloud computing is not free, and most users may not be accustomed to the direct, multifaceted costs of their compute usage. Ultimately, for running any Earth system model, there is no universal answer as to whether cloud or traditional computing is better for a given user: it depends on the particulars of the user's needs, resources, and priorities.

Documentation on using WRF in the supported CSP environments may be found on the WRF users' page.[10] The cloud and these new capabilities are meeting needs of the WRF user and developer communities, as well as advancing the support of the modeling system itself.

[10] This may be found under the main WRF users' page: https://www2.mmm.ucar.edu/wrf/users. Information is updated under the "User Support" tab under subheading "WRF Cloud Computing Info".

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Acknowledgments

The authors thank Amazon Web Services and Scala Computing for their support in both resources and expertise in the creation, support, and use of Cloud WRF components. The authors also wish to express their appreciation of the reviewers for their input that improved the manuscript. We acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. This work was supported by the National Science Foundation, Office of Advanced Cyberinfrastructure, and its Cyberinfrastructure for Sustained Scientific Innovation program under NSF Award 1835511.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

References

Ansari, S., and Coauthors, 2018: Unlocking the potential of NEXRAD data through NOAA's Big Data Partnership. *Bull. Amer. Meteor. Soc.*, **99**, 189–204, https://doi-org.cuucar.idm.oclc.org/10.1175/BAMS-D-16-0021.1.

Bey, I., D. J. Jacob, R. M. Yantosca, J. A. Logan, B. Field, A. M. Fiore, Q. Li, H. Liu, L. J. Mickley, and M. Schultz, 2001: Global modeling of tropospheric chemistry with assimilated meteorology: Model description and evaluation. *J. Geophys. Res.*, **106**, 23,073-23,096, 2001.

Booij, N., R.C. Ris, and L.H. Holthuijsen, 1999: A third-generation wave model for coastal regions: Part I— Model description and validation. *J. Geophys. Res.*, **C4, 104**, 7649–7666.

Chacon, S., and B. Straub, 2014: Pro Git. Apress, 458 pp. doi: https://doi.org/10.1007/978-1-4842-0076-6

Chen, X., and Co-Authors, 2017: Running climate model on a commercial cloud computing environment: A case study using Community Earth System Model (CESM) on Amazon AWS. *Computers and Geosciences*, **98**, January 2017, 21-25. https://doi.org/10.1016/j.cageo.2016.09.014

Chui, T.C., D. Siuta, G. West, H. Modzelewski, R. Schigas, and R. Stull, 2019: On producing reliable and affordable numerical weather forecasts on public cloud-computing infrastructure. *J. Atmos. Oceanic Technol.*, **36**, 491–509, https://doi.org/10.1175/JTECH-D-18-0142.1

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Coen, J.L., M. Cameron, J. Michalakes, E.G. Patton, P.J. Riggan, and K.M. Yedinak, 2013: WRF-Fire: Coupled Weather–Wildland Fire Modeling with the Weather Research and Forecasting Model. *J. Appl. Meteor. Climatol.*, **52**, 16–38. doi: http://dx.doi.org/10.1175/JAMC-D-12-023.1.

Coffrin, C., and Coauthors, 2019: The ISTI rapid response on exploring cloud computing 2018 (August), 72 pp. http://arxiv.org/abs/1901.01331

Duran-Limon, H. A., J. Flores-Contreras, N. Parlavantzas, M. Zhao, and A. Meulenert-Pena, 2016:. Efficient execution of the WRF model and other HPC applications in the cloud. *Earth Sci. Informatics*, **9**, 365–382, https://doi.org/10.1007/s12145-016-0253-7.

Fast, J.D., W.I. Gustafson, E.G. Chapman, R.C. Easter, J. Rishel, R.A. Zaveri, G.A. Grell, and M. Barth, 2006: Evolution of ozone, particulates, and aerosol direct forcing in an urban area using a new fully-coupled meteorology, chemistry, and aerosol model. *J. Geophys. Res.*, **111**, D21305. doi:10.1029/2005JD006721.

Gochis, D.J., W. Yu, D.N. Yates, 2015: The WRF-Hydro Model Technical Description and User's Guide, Version 3.0. NCAR Tech. Document. 120 pp. Available online at: http://www.ral.ucar.edu/projects/wrf_hydro/.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Goga, K., A. Parodi, P. Ruiu, and O. Terzo, 2018: Performance analysis of WRF simulations in a public cloud and HPC environment. *Complex, Intelligent, and Software Intensive Systems*, L. Barolli and O. Terzo, Eds., pp. 384–396. Springer International, 517 pp.

Grell, G.A., S.E. Peckham, R. Schmitz, S.A. McKeen, G. Frost, W.C. Skamarock, and B. Eder, 2005: Fully-coupled "online" chemistry within the WRF model. *Atmos. Environ.*, **39**, 6957–6976.

Harris, L., L. Zhou, X. Chen, and J.-H. Chen, 2020: *The GFDL Finite-Volume Cubed-Sphere Dynamical Core: Release 201912*, Princeton, NJ: NOAA Technical Memorandum OAR GFDL, 2020-001, doi:10.25923/7h88-c53410pp.

Hurrell, J. W., Holland, M. M., Gent, P. R., Ghan, S., Kay, J. E.,Kushner, P. J., Lamarque, J.-F., Large, W. G., Lawrence, D.,Lindsay, K., Lipscomb, W. H., Long, M. C., Mahowald, N.,Marsh, D. R., Neale, R. B., Rasch, P., Vavrus, S., Vertenstein, M.,Bader, D., Collins, W. D., Hack, J. J., Kiehl, J., and Marshall, S, 2013: The Community Earth System Model: a framework for collaborative research. *Bull. Amer. Meteor. Soc.*, **94**, 1339–1360, doi:10.1175/BAMS-D-12-00121.1.

McKenna, B., 2016: Dubai operational forecasting system in Amazon cloud. *Cloud Computing in Ocean and Atmospheric Sciences*, T. C. Vance et al., Eds., Academic Press, 325–345, doi:10.1016/B978-0-12-803192-6.00016-5.

Molthan, A.L., J.L. Case, J. Venner, R. Schroeder, M.R. Checci, B.T. Zavodsky, A. Limaye, and R.G. O'Brien, 2015: Clouds in the cloud: Weather forecast and applications within cloud

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

computing environments. *Bull. Amer. Meteor. Soc.*, **96**, 1369–1379, doi:10.1175/BAMS–D–10–00013.1.

NOAA Environmental Modeling Center, 2003: The GFS Atmospheric Model. NCEP Office Note 442. 14 pp. [Available from U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Centers for Environmental Prediction, 4700 Silver Hill Rd., Mail Stop 9910, Washington, DC, 20233.]

Powers, J. G., and Coauthors, 2017: The Weather Research and Forecasting (WRF) Model: Overview, system efforts, and future directions. *Bull. Amer. Meteor. Soc.*, **98**, 1717–1737, doi: 10.1175/BAMS–D–15–00308.1.

Shchepetkin, A.F., McWilliams, J.C., 2005. The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, **9(4)**, 347–404.

Siuta, D., G. West, H. Modzelewski, R. Schigas, and R. Stull, 2016: Viability of cloud computing for real-time numerical weather prediction. *Wea. Forecasting*, **31**, 1985–1996, doi: 10.1175/WAF-D-16-0075.1.

Skamarock, W.C., J.B. Klemp, J. Dudhia, D.O. Gill, Z. Liu, J. Berner, W. Wang, J.G. Powers, D.M. Barker, and X.-Y. Huang, 2019: A description of the Advanced Research WRF Model Version 4. NCAR Tech. Note, NCAR/TN-5565+STR, 162 pp. doi: http://dx.doi.org/10.5065/1dfh-6p97.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Zhuang, J., D.J. Jacob, J.F. Gaya, R.M. Yantosca, E.W. Lundgren, M.P. Sulprizio, and S.D. Eastham, 2019: Enabling immediate access to Earth science models through cloud computing: Application to the GEOS-Chem Model. *Bull. Amer. Meteor. Soc.*, **100**, 1943–1960, https://doi.org/10.1175/BAMS-D-18-0243.1.

Zhuang, J., D.J. Jacob, H. Lin, E.W. Lundgren, R.M. Yantosca, J.F. Gaya, M.P. Sulprizio, and S.D. Eastham, 2020: Enabling high- performance cloud computing for Earth science modeling on over a thousand cores: Application to the GEOS- Chem atmospheric chemistry model. *J. Adv. in Modeling Earth Systems*, **12**, e2020MS002064, https://doi.org/10.1029/2020MS002064.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.

Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Sidebar: WRF Tutorial Use of Cloud Computing

Instructional tutorials on the WRF Model have turned to cloud computing, using an AWS environment, for support of practical training on running the system. This instruction involves students configuring and executing WRF simulations using the cloud setup. Feedback from tutorial students on Cloud WRF has been positive, and the quotations below are from post-tutorial surveys. The examples note the cloud's practicality and ease of use for WRF, with learning and model operation facilitated. The chart shows ratings of Cloud WRF used for the tutorial's practice sessions on a scale from 1 to 5 (best) based on surveys following four tutorials. 92% of the 96 respondents rated the experience 4 or 5.

"Best training environment I have experienced. Everything just worked fine."

"It works great and likely very similar to how most people would use WRF in a practical environment."

"I think this is the best way to administer the tutorial— a reason being is that people always cite issues with trying to build the code on their respective platforms/laptops."

"This was actually really nice to practice with since some institutions are looking into cloud-based solutions."

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

"I had no complaints. Everything was easy and accessible. I would happily run the practice in the cloud again."

"Using the cloud to run WRF was a great idea since my computer cannot handle the load in a decent time frame, nor the storage for the output files. This also helped to solve dependency conflicts as the environment was already setup and ready to go."

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Tables

	Period Ending April 2016 V3.8 Release	Period Ending April 2017 V3.9 Release	Period Ending June 2018 V4.0 Release	Period Ending April 2019 V4.1 Release	Period Ending April 2020 V4.2 Release
Core Contributors	11	9	10	8	7
Non-Core Contributors	1	10	14	16	34
Number of PRs by External Contributors	1	38	48	37	55

Tab. 1: Comparison of the recent WRF releases, showing the contributions accumulated during the previous year by the WRF support team members (core contributors) and by external developers (non-core contributors). The jump between WRF releases 3.8 and 3.9 represents the move from the Subversion code management system to that of Git and GitHub. The next big increase in external contributions, from the WRF V4.1 to V4.2, reflects the availability of the automated cloud testing system. The number of pull requests (PRs) to the WRF repository by external users has steadily increased.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Nodes	**Cores (MPI Processes)**	**NCAR HPC—Single file (sec)**	**NCAR HPC—Split file (sec)**	**AWS—Single file (sec)**	**AWS—Split file (sec)**
32	**1152**	**40.5 ± 0.4**	**0.22 ± 0.03**	**72.4 ± 3.7**	**10.8 ± 0.2**
64	**2304**	**46.4 ± 0.2**	**0.18 ± 0.03**	**82.4 ± 5.9**	**8.9 ± 0.4**
100	**3600**	**49.5 ± 0.1**	**0.19 ± 0.09**	**78.8 ± 4.6**	**8.7 ± 0.5**

Tab. 2: Amount of time (sec, ± standard deviation) to output each of the four WRF 1500x1500 benchmark history time periods, where the aggregate of each time period is approximately 11 GB using uncompressed NetCDF4. The "single file" option is the standard run-time configuration, and the "split file" option is for each MPI process outputting the portion of the file resident in that process's memory. Timings are provided for the larger core counts conducted with the WRF benchmark case for the NCAR HPC Cheyenne and the AWS c5n.18xlarge platforms, each with 36 cores/node.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

Figures

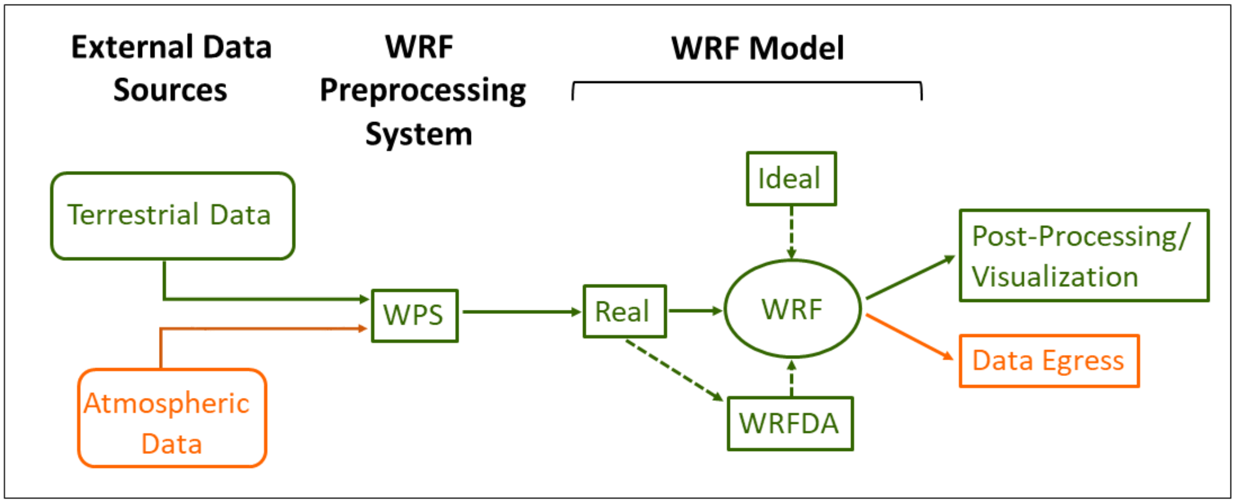
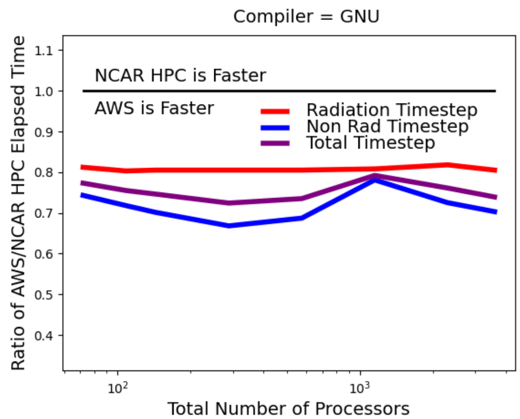
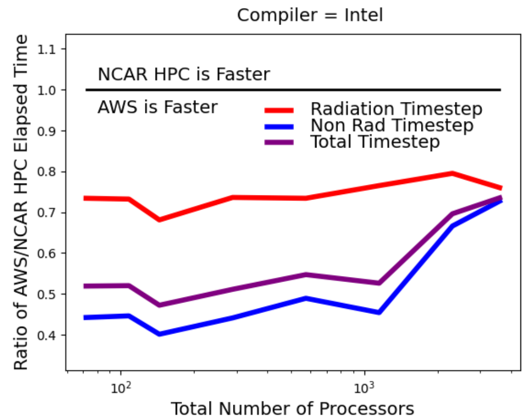


Fig. 1: WRF system components currently in available Cloud WRF in flow chart of model simulations. WPS= WRF Preprocessing System, REAL= program Real, WRFDA= WRF data assimilation system, Ideal= program Ideal. Dashed lines denote optional paths/approaches for model simulations: performing idealized simulations or reanalyzing a real-data first-guess field with observations using WRFDA. Components in green are in the cloud AWS and Scala environments. Elements in orange are up to the user to provide/arrange.

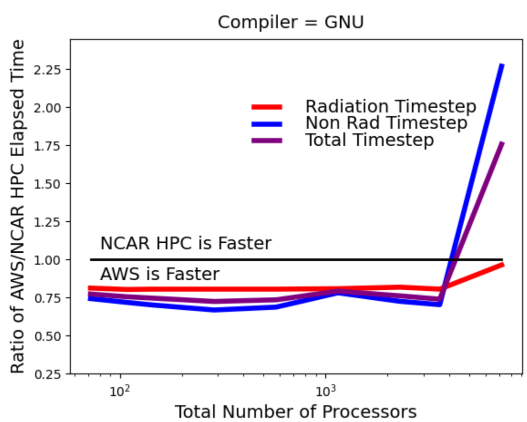
Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC



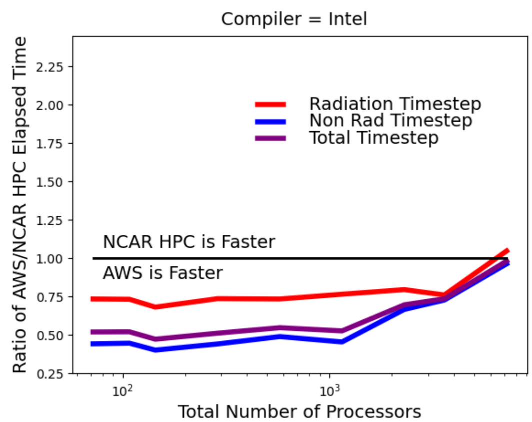
(a)



(b)



(c)



(d)

Fig. 2: Timing results of the WRF benchmark runs on AWS and NCAR HPC (high performance computer; “Cheyenne”) hardware, without I/O time included, for two compilations of WRF, one using a GNU compiler and one using an Intel compiler. The relative performance of the two environments is expressed as the ratio of wallclock seconds per timestep of the AWS platform to the NCAR platform (AWS/NCAR), with timestep averaging over 19 steps. The benchmark is run with increasing numbers of processors on both platforms, with the curves based on the following core counts: 72, 108, 144, 288, 576, 1152, 2304, 3600, and 7200. The results to 3600 cores are shown in (a) and (b) as separate panels for clarity across this range. Ratio values less than 1 mean that the wallclock time for each WRF model time step on the AWS platform is less than that for each one on the NCAR HPC (i.e., less wallclock time per

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

WRF model time step), at the shown fraction; for these values AWS's time-to-solution pace is faster. Conversely, for timing ratio values greater than 1 the HPC's time-to-solution is faster. Red curve shows relative performance for radiation time steps, blue curve for non-radiation time steps, and purple curve for a weighted average of radiation and non-radiation time steps. For the averaged results (purple), the point beyond which the ratio of AWS/HPC exceeds 1 occurs at 3600 cores for GNU and 7200 cores for Intel. (a) GNU compiler, to 3600 cores. (b) Intel compiler, to 3600 cores. (c) GNU compiler, to 7200 cores. (d) Intel compiler, to 7200 cores.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

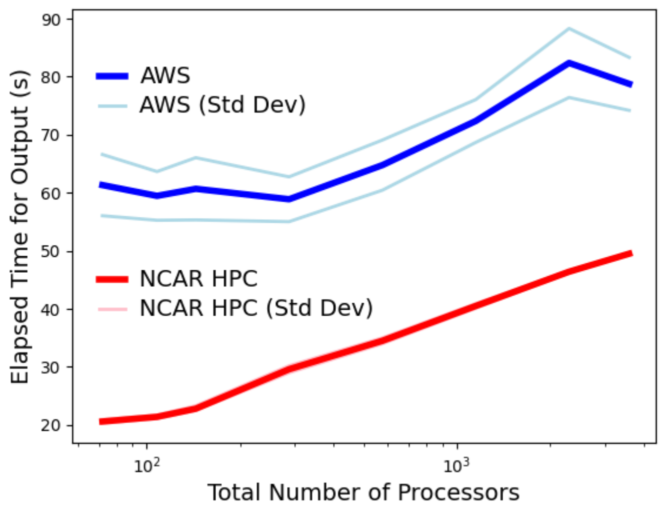


Fig. 3: Timing results of the WRF benchmark runs on AWS and NCAR HPC (high performance computer; “Cheyenne”) hardware, with only the I/O time included, which here is output only. The benchmark is run on increasing numbers of processors, with the curves based on the following core counts: 72, 108, 144, 288, 576, 1152, 2304, and 3600. A total of four time periods were output, and the average value is plotted. The serial NetCDF library outputs the 11 GB data in the classic format with WRF option io_form_history=2. Average value= thick lines; standard deviation= thin lines reported. The variability of the output timings on the NCAR HPC machine is too small to be seen on this plot (e.g., typically 0.5 s), and thus the standard deviation lines are not distinct for the NCAR HPC curve. For this test, the Intel compiler build of the WRF model is used.

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC

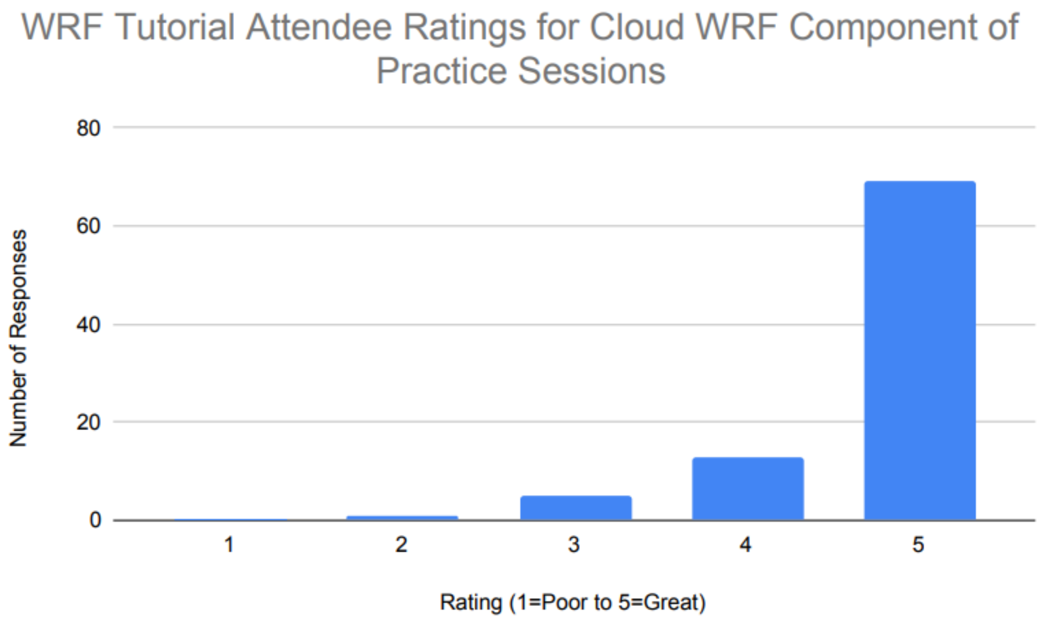


Fig. SB1: Results of surveys of WRF tutorial students rating the Cloud WRF component of the tutorial practice sessions. Practice sessions are those in which the students configure and run WRF simulations, here using a cloud compute environment. Number of respondents 96, over four tutorials. Scale from 1 (poor) to 5 (great).

Accepted for publication in *Bulletin of the American Meteorological Society*. DOI 10.1175/BAMS-D-20-0219.1.
Unauthenticated | Downloaded 04/15/21 12:13 AM UTC