On Producing Reliable and Affordable Numerical Weather Forecasts on Public Cloud-Computing Infrastructure

TIMOTHY C. Y. CHUI

The University of British Columbia, Vancouver, British Columbia, Canada

DAVID SIUTA

Northern Vermont University, Lyndonville, Vermont

GREGORY WEST, HENRYK MODZELEWSKI, ROLAND SCHIGAS, AND ROLAND STULL

The University of British Columbia, Vancouver, British Columbia, Canada

(Manuscript received 9 August 2018, in final form 20 January 2019)

ABSTRACT

Cloud-computing resources are increasingly used in atmospheric research and real-time weather forecasting. The aim of this study is to explore new ways to reduce cloud-computing costs for real-time numerical weather prediction (NWP). One way is to compress output files to reduce data egress costs. File compression techniques can reduce data egress costs by over 50%. Data egress costs can be further minimized by postprocessing in the cloud and then exporting the smaller resulting files while discarding the bulk of the raw NWP output. Another way to reduce costs is to use preemptible resources, which are virtual machines (VMs) on the Google Cloud Platform (GCP) that clients can use at an 80% discount (compared to nonpreemptible VMs), but which can be turned off by the GCP without warning. By leveraging the restart functionality in the Weather Research and Forecasting (WRF) Model, preemptible resources can be used to save 60%–70% in weather simulation costs without compromising output reliability. The potential cost savings are demonstrated in forecasts over the Canadian Arctic and in a case study of NWP runs for the West African monsoon (WAM) of 2017. The choice in model physics, VM specification, and use of the aforementioned cost-saving measures enable simulation costs to be low enough such that the cloud can be a viable platform for running short-range ensemble forecasts when compared to the cost of purchasing new computer hardware.

1. Introduction: Background and motivation

The field of numerical weather prediction (NWP) has made progress across all aspects of the forecasting process within the past century (Bauer et al. 2015). Though much of the progress has been a result of research conducted at national and international organizations like the National Centers for Environmental Prediction (NCEP) and the European Centre for Medium-Range Weather Forecasts (ECMWF), developments have also been made throughout academia and in private industry. Community models such as the Weather Research and

Forecasting (WRF; Skamarock et al. 2008) Model are used by academic groups for making some of these advances (Mass and Kuo 1998; Gritter and Mass 2002; Thomas et al. 2002; Colle and Zeng 2004; McCollor and Stull 2008). Academic groups often lack the large budget necessary to purchase and maintain the computer hardware and dedicated information technology (IT) staff at the scale needed for some NWP studies. For example, the bare-metal costs of a small (448 core) high-performance computing (HPC) cluster similar to the one currently used at The University of British Columbia (UBC) for NWP research would cost \$143,000–\$226,000 (U.S. dollars) amortized over a life-span of 3–5 years (Siuta et al. 2016, hereinafter *SWMSS*).

Many WRF users instead rely on communal computing resources for their research or teaching needs (Niang 2011; Powers et al. 2017). Examples include

Denotes content that is immediately available upon publication as open access.

Corresponding author: Timothy C. Y. Chui, tchui@eoas.ubc.ca

DOI: 10.1175/JTECH-D-18-0142.1

© 2019 American Meteorological Society. For information regarding reuse of this content and general copyright information, consult the [AMS Copyright Policy](#) (www.ametsoc.org/PUBSReuseLicenses).

NCAR's Cheyenne and Compute Canada's Westgrid systems. Often such resources come with allocation time limits and long queue times. To avoid these problems, public cloud-computing infrastructure can provide a cost-effective, on-demand solution for atmospheric science research and real-time forecasting.

"Public cloud infrastructure" refers to virtualized products and services provisioned by a commercial service provider for use by the general public for a fee (Voorsluys et al. 2011). Examples of commercial cloud providers include the Google Cloud Platform (GCP), Amazon Web Services (AWS), and Microsoft Azure. Products and services include hosted web servers, machine-learning platforms, and application development tools. Infrastructure as a service (IaaS) refers to the sector of cloud computing where raw computing resources can be purchased in the form of virtual machines (VMs) and storage (Chang et al. 2010; Yuan 2016). By leveraging cloud infrastructure, researchers and forecasters can pay small monthly fees to perform computations on clusters managed by commercial providers, without having to pay large up-front costs to purchase computer clusters or hire staff to maintain them.

The use of cloud services for weather and climate research has grown in popularity over the past several years (Vance 2016; Powers et al. 2017). Molthan et al. (2015) showed that running NWP models on AWS was possible, and discussed the potential that cloud-based solutions have on forecast capacity building in developing nations. McKenna (2016) used AWS to implement a real-time forecasting system for Dubai. SWMSS experimented with different compilers, VM sizes, and total compute core counts, and showed that the costs of a forecast system on the GCP can be comparable to or cheaper than the cost of an equivalent on-premise HPC cluster. Aside from NWP, public cloud infrastructure has been used for data dissemination in Unidata's Thematic Real-Time Environmental Distributed Data Services (THREDDS; Ramamurthy 2016); and for visualization in the Climate Engine (Huntington et al. 2017).

Unfortunately, the costs cited in Molthan et al. (2015) and SWMSS may still be too high for many research groups. Molthan et al. (2015) reported a cost of \$40–\$75 for a 48-h simulation with domains (a 12-km parent and two 4-km nests) over the Gulf of Mexico. The costs of the Molthan et al. (2015) simulation would extrapolate to \$36,000–\$68,000 yr⁻¹ when the same setup is used to produce a daily 5-day forecast. SWMSS tested a similarly sized simulation over the Canadian Arctic using the GCP, and found runtime limitations were a function of horizontal and vertical scaling of cloud-computer

resources and the compiler used. Horizontal scaling refers to varying the number of nodes [compute virtual machines (CVMs)] in a cluster, while vertical scaling refers to varying the number of cores [virtual central processing units (vCPUs)] per node. For the case study in SWMSS, the cost-optimized configuration was eight CVMs with eight vCPUs each, running Intel-compiled WRF, resulting in a \$5,500 annual cost. Of the total cost, 35% was due to data egress, which is the fee charged by cloud providers for the transfer of data out of the cloud. SWMSS mentioned the high cost of data egress as an ongoing problem in cloud-based simulations, but did not suggest potential solutions. They also addressed the availability of preemptible resources on the GCP, which are VMs provided at an 80% discount (compared to regular, nonpreemptible VMs) that may be shut down and reallocated without prior notice by Google if demand rises. Their volatility is undesirable for real-time forecasting based on the setup in SWMSS, but the use of preemptible resources could result in substantial savings if they could be leveraged without sacrificing forecast-production reliability.

As mentioned by Molthan et al. (2015), the use of public cloud infrastructure for NWP can greatly enhance the forecasting capabilities of developing nations. Initiatives such as the National Aeronautics and Space Administration (NASA) SERVIR Project (NASA 2018) and the World Meteorological Organization (WMO) Severe Weather Forecasting Demonstration Project (SWFDP; WMO 2016) were set up to disseminate data and model output for local meteorologists in these nations. SERVIR in particular was supported by regional modeling activities conducted using public and private cloud-computing resources (Molthan et al. 2015).

Recently, the SWFDP was expanded to include West Africa and the Sahel in 2015, to enhance local capabilities in using NWP guidance for severe weather forecasting (MeteoWorld 2017). Severe weather in West Africa and the Sahel is often associated with convective activity stemming from the West African monsoon (WAM; Fink et al. 2011). Being able to properly represent convective activity and the WAM in NWP and regional climate models can have positive ramifications for both predicting the spread of disease (Pandya et al. 2015) and agricultural management (Smith 2015; Huntington et al. 2017).

Recent studies to improve weather and climate prediction in West Africa involve exploring the effect of physics parameterizations on regional forecasts (Klein et al. 2015; Heinzeller et al. 2016), increasing spatial and temporal resolutions (Park 2014; Prein et al. 2015; Vellinga et al. 2016), and using appropriate forecast

verification techniques (Graham 2014; James et al. 2017). However, the problem of computing resources and computational cost remains. Many countries in Africa do not have registered WRF users or do not run an operational mesoscale model, which may indicate a lack of computational infrastructure to support running NWP models locally (Dudhia 2014; Powers et al. 2017). These concerns could be addressed by running mesoscale models in an affordable way on cloud infrastructure.

The purpose of this study is to explore new methods (section 2) to reduce cloud-related costs for NWP. In this study, we address the following research questions:

- How can WRF simulation times be reduced? To answer this, we experiment with several WRF versions and compilers (section 3a) and different CVM sizes (section 3b) for a Canadian Arctic domain, expanding on tests conducted in SWMSS.
- How can data egress costs be reduced? To address this, we explore different file compression options as would affect output file sizes, and discuss how cloud-based postprocessing could greatly reduce data egress costs (section 3c).
- How can the WRF restart functionality be used to offset the volatility of the much cheaper preemptible CVMs to further reduce compute-related costs for real-time weather forecasting (section 3d)?

We then apply those cost-saving methods and findings to a precipitation case-study simulation over West Africa (section 4), and show that the costs for WRF runs in this region are comparable to those of the similarly sized Arctic domain. We experiment with two different physics suites to show how physics research could be conducted on the cloud, and discuss how simulation timing and memory requirements could be met on demand. A summary is given in section 5.

2. Methods

a. Google Cloud Platform workflow

The basic structure of our cloud-based forecasting system is shown in Fig. 1. The cloud setup consists of a head VM (HVM), which controls forecast runs and maintains the system; and CVMs, which behave like traditional parallel compute nodes when distributing WRF runs using a message passing interface (MPI) implementation. GCP provides an application programming interface (API) to allow for command-line interaction with VMs. A small, inexpensive management VM can be programmed to use the API to automatically turn on and off the HVM, and the HVM to turn on and off CVMs to minimize costs via scripted

FIG. 1. Diagram of the UBC forecasting system on the GCP, adapted from SWMSS. Arrows represent communication and data transfer pathways. The management virtual machine (Management VM) turns on the head virtual machine (Head VM) at the beginning of a forecast cycle, and initial and lateral boundary-condition files are downloaded from NCEP. The head VM is then responsible for turning on the compute virtual machines (Compute VMs) needed to run a forecast. Input and output fields are stored on the attached network storage disk, and required files are egressed to servers at UBC.

processes. The HVM and CVMs all share a common 1-TB network disk via network file system (NFS) mounting for the short-term storage of preprocessed input files and output results. Initial and lateral boundary condition (IBC) files for WRF are downloaded to the HVM from external data centers, such as NCEP. The HVM is responsible for preprocessing IBCs in the WRF Pre-processing System (WPS) before allocating CVMs and initiating WRF runs. After a run is complete, the required output files are transferred out of the cloud to local servers at UBC for archiving, and finally the CVMs and HVM are automatically turned off to save money. Only the small management VM remains on all the time.

All VMs in this study use Intel Haswell processors, which are also available on AWS and Azure instances. Though the GCP provides other VMs with Intel Broadwell and Intel Skylake processors, preliminary testing showed no significant timing advantage using the default settings over Haswell processors. In particular, the marginal performance improvement with Skylake architecture (under 10 min for a 6-h simulation) does not warrant its premium cost for our forecasting purposes.

We use the same vendor (GCP) and the cloud-cluster setup as used in SWMSS so that comparisons can be made. Preemptible CVMs on the GCP offer an 80% discount over nonpreemptible CVMs (<https://cloud.google.com/compute/pricing>). A similar set of preemptible resources on AWS, called Spot Instances, are discounted up to 90% from regular VMs, although these instances require prebidding to use, and their availability and cost can fluctuate depending on bid

prices (<https://aws.amazon.com/ec2/pricing>). Azure has low-priority VMs that work similarly to preemptible resources on the GCP, and have a similar discount of 80%, although the equivalent cost is more expensive than on the GCP at the time of writing (<https://azure.microsoft.com/en-ca/pricing/details/batch>).

To maintain production reliability while leveraging discounted preemptible resources, one can take advantage of the WRF restart functionality. WRF enables runs to periodically save “restart files,” allowing forecast runs to be restarted from checkpoints. If any single CVM is preempted, the MPI communications between the CVMs break, causing the whole run to fail. Restart files allow a simulation to be restarted in the event of a preemption, without having to start from the beginning of the simulation. The simulation is instead scripted to automatically restart from the point at which the last restart file is written.

We use version 3.9.1.1 of the Advanced Research WRF (ARW) dynamical core of the WRF Model for our tests, which supports restart files (Skamarock et al. 2008). Though SWMSS used version 3.7.1, we decided to use the latest stable release of WRF, version 3, for our tests. Section 3a discusses the differences in run times for the two different WRF versions and compilers for a common case study.

Writing WRF restart files requires additional input/output (I/O) time, which lengthens the overall time of a model run, and consequently the uptime costs of the CVMs. Hence, users must consider an optimal output frequency of restart files, such that restart files are produced frequently enough to minimize setback after a preemption, but not too often that the overall length of the simulation becomes unreasonably long and expensive.

The general workflow for performing a WRF simulation on the GCP using preemptible resources is as follows:

- 1) Download the required IBC files, and preprocess them on the HVM to initialize WRF.
- 2) Spin up (assign and turn on) the preemptible CVMs.
- 3) Run WRF using an MPI implementation across those CVMs. Output the restart files at a set interval, specified by the “restart_interval” variable in the WRF namelist.input file. The “restart” variable is set to “.false.” to indicate that the simulation is a new run.
- 4) If a CVM is preempted, the MPI process fails and will output an appropriate exit code that is automatically detected. Scripts then update the WRF namelist.input file to specify that the run will continue from the latest restart files by editing the appropriate simulation start variables (“start_year,” “start_month,” etc.). The simulation length variables (“run_days,” “run_hours,” etc.) are also reduced by subtracting the restart times (updated “start_year,” “start_month,” etc.) from the simulation end times (“end_year,” “end_month,” etc.). The restart variable is set to “.true.” to indicate that the simulation is starting from restart files.
- 5) Automatically reboot the CVMs, and rerun WRF with the updated namelist.input. If a preemption occurs again, automatically redo step 4.
- 6) Optionally, compress the raw WRF output files and egress them using a file transfer utility to a local disk outside the cloud. Otherwise, perform postprocessing on the HVM, and delete the raw WRF output files once they are no longer needed.

The compression utility we use is “nccopy” from netCDF library, version 4.5.0-rc1 (Unidata 2017). The resulting output after compression is still a usable gridded file that can be accessed, and if required, decompressed to reproduce the original file without compromising the variable fields. The extent to which a file is compressed is defined as

$$\text{Compression factor} = 1 - \frac{\text{Compressed file size}}{\text{Original file size}}.$$

The time it takes to compress a file could vary depending on the options used. Available flags/options include compression level, ranging from “-d 1” being the fastest, to “-d 9” being the slowest; memory shuffling with “-s,” which places bytes contiguously in storage; and dimension conversion with “-u,” which applies a capped dimension size to variables with unlimited size. Shuffling assists with compression, while dimension conversion speeds up data access.

b. Model domains and experimental setups

This study consists of two separate model domain configurations, each with 41 terrain-following (eta) levels, and a model top at 50 hPa:

- 1) A two-way nested $\Delta x = 36\text{--}12\text{--}4\text{-km}$ domain over the Canadian Arctic, identical to SWMSS (Fig. 2a). This is used to assess cost reduction via preemptible CVMs and data compression. The 36-km domain (D01) contains $200 (\text{west--east}) \times 108 (\text{south--north})$ grid points; the nested 12-km domain (D02) contains 346×250 grid points; and the nested 4-km domain (D03) contains 448×307 grid points.
- 2) A two-way nested $36\text{--}12\text{--}4\text{-km}$ domain over West Africa and the Sahel, to demonstrate its cost and utility during the WAM season of 2017 (Fig. 2b).

FIG. 2. Domains used for this study, with the nest areas plotted for (a) the Canadian Arctic domain and (b) the West African domain. The domains are labeled as D01 (36 km; 200 west–east $\times$ 108 south–north grid points), D02 (12 km; 346 $\times$ 250), and D03 (4 km; 448 $\times$ 307). Selected countries are labeled as follows: SE = Senegal; TG = The Gambia; GB = Guinea-Bissau; SL = Sierra Leone; GU = Guinea; LI = Liberia; IC = Ivory Coast; MA = Mali; BF = Burkina Faso; GH = Ghana; TO = Togo; BE = Benin; NR = Niger; NA = Nigeria.

The grids in this domain have the same number of points as the Arctic domain.

The West African case-study domain provides a fair computational-cost comparison with the Arctic domain due to its equivalent size. Therefore, this set of tests is designed to evaluate the impact of changing regions and physics on simulation cost.

The simulation lengths and physics schemes used are summarized in Table 1. All simulations are initialized with the NCEP 0.5° Global Forecast System (GFS), to match that used in SWMSS.

Details pertinent to the Arctic and West African simulations are discussed in the following subsections. A summary of the tested WRF and CVM configurations are detailed in Table 2.

1) CANADIAN ARCTIC DOMAIN

All Arctic domain simulations use 64 vCPUs of the n1-highcpu machine type (0.9 GB of memory per vCPU), to match with the most cost-efficient vCPU setup outlined in SWMSS. The physics schemes used also match those used in SWMSS, and are

collectively referred to as the S2016 suite (Table 1, left column). Timing information is parsed from WRF rsl.error.0000 files. The Arctic domain tests involve four parts:

- 1) Comparisons between WRF-ARW versions 3.9.1.1 and 3.7.1 are made, using a benchmark simulation initialized at 1200 UTC 25 September 2015, to match the simulation used in SWMSS. SWMSS used the GNU's not Unix (GNU) MPICH, version 3.1, and Intel MPI, version 5.1.1, implementations for their distributed runs, and showed that the Intel-compiled WRF was substantially faster than the GNU-compiled WRF for the same number of cores, in agreement with a previous benchmarking study (HPC Advisory Council 2015). However, the Intel compiler library used was a trial version, and had since expired. A full license for the complete 2018 compiler library would cost \$3,000 yr⁻¹ (<https://software.intel.com/en-us/parallel-studio-xe>). Recently, a community edition of the PGI compiler library was made available to all users (<https://www.pgroup.com/products/community.htm>). We compiled WRF-ARW,

TABLE 1. WRF simulation lengths and physics schemes for the Arctic and WAM tests. The Arctic test runs use the same physics schemes used in SWMSS, and these schemes are collectively referred to as the “S2016” suite. The WAM case study has runs made with both the S2016 suite and the C-P suite. D01–D03 refer to the nested grids with horizontal grid spacings of 36, 12, and 4 km, respectively.

	Physics suite	
	S2016	C-P
Simulation length (h)	D01: 156; D02: 108; D03: 60	D01: 156; D02: 108; D03: 60
Planetary boundary layer	YSU (Hong et al. 2006)	MYNN2 (Nakanishi and Niino 2009)
Surface layer	MM5 (Zhang and Anthes 1982)	MYNN (Nakanishi and Niino 2009)
Microphysics	WSM5 (Hong et al. 2004)	Thompson (Thompson et al. 2008)
Convection	Kain–Fritsch (Kain 2004)	Grell–Freitas (Grell and Freitas 2014)
Land surface	Noah (Ek 2003)	Noah (Ek 2003)
Longwave radiation	RRTM (Mlawer et al. 1997)	RRTMG (Iacono et al. 2008)
Shortwave radiation	Dudhia (Dudhia 1989)	RRTMG (Iacono et al. 2008)

TABLE 2. Test configurations for the Arctic and West African domains. The initializations and physics suites used are shown for each test. Machine configurations are specified by compute virtual machine (CVM) type (n1-highcpu, n1-standard) and the number of virtual central processing units (vCPU) in each CVM, totaling 64 vCPUs (16 CVM $\times$ 4 vCPU, 8 CVM $\times$ 8 vCPU, etc.). The WAM-1 test uses the n1-highcpu machine type for the S2016 physics suite, and the n1-standard type for the more memory intensive C-P physics suite.

Test	Initializations	Physics suite	Machine configuration
Arctic-1 (see section 3a)	25 Sep 2015	S2016	n1-highcpu 0.9 GB/vCPU 8 $\times$ 8
Arctic-2 (see sections 3b and 3c)	28 Sep 2017	S2016	n1-highcpu 0.9 GB/vCPU 8 $\times$ 8
Arctic-3 (see sections 3b and 3d)	29 Sep–6 Oct 2017	S2016	n1-highcpu 0.9 GB/vCPU 16 $\times$ 4, 8 $\times$ 8, 4 $\times$ 16, 2 $\times$ 32, 1 $\times$ 64
WAM-1 (see section 4)	17 Aug 2017	S2016	n1-highcpu 0.9 GB/vCPU 16 $\times$ 4
		C-P	n1-standard 3.75 GB/vCPU 16 $\times$ 4

versions 3.7.1 and 3.9.1.1, with PGI pgcc and pgfortran, version 17.4, using the OpenMPI, version 1.10.2, implementation for MPI. The simulations are conducted using nonpreemptible 8-vCPU instances, with hourly output of meteorological fields and no output of restart files. The results for this test are discussed in [section 3a](#) ([Table 2](#); Arctic-1), which discusses speed and cost differences between the two WRF versions, and between three compilers.

- 2) Vertical scaling is tested to try to minimize simulation costs for a WRF run, by varying the number of vCPUs per CVM while keeping the total vCPU count at 64. Two sets of simulations are conducted, one with a static time step ($\Delta t = 216$ s for the $\Delta x = 36$ -km domain), and the other with an adaptive time step ([Hutchinson 2007](#)). All simulations are initialized at 0000 UTC 28 September 2017. I/O frequency is increased to determine whether it would have a larger impact on the total simulation time than the NWP computation time. The output frequency of meteorological history files is changed from 60 to 30 min, and the output frequency of restart files is set to 3 h. [Section 3b](#) discusses the effects of CVM size and time-stepping method (adaptive or static) on computation time (excluding time required to output WRF history and restart files), history file I/O time (time required to output WRF history files), and restart file I/O time (time required to output WRF restart files; [Table 2](#); Arctic-2, Arctic-3).
- 3) File compression is tested to minimize egress costs for a WRF run (initialized at 0000 UTC 28 September 2017). [Section 3c](#) addresses compression time and factor,

and how compression can be used to reduce egress costs if data transfer is necessary ([Table 2](#); Arctic-2).

- 4) Real-time forecasts are run between 29 September and 6 October 2017 for both the 0000 and 1200 UTC GFS initializations, all made with preemptible CVMs in a real-time setting. The runs are scripted to start on a twice-daily basis, with five separate runs concurrently for each initialization corresponding to five different CVM sizes: 4, 8, 16, 32, and 64 vCPUs per CVM. Because of the time needed for the GFS files to be produced and downloaded from NCEP and then locally preprocessed, the 0000 and 1200 UTC WRF runs are started approximately five hours past GFS initialization (i.e., at 0500 and 1700 UTC, respectively). The adaptive time step is used to optimize simulation time without violating the Courant–Friedrichs–Lewy numerical-stability condition ([Hutchinson 2007](#)). Output frequencies are identical to [section 3b](#). [Section 3d](#) discusses the results from these tests ([Table 2](#); Arctic-3), which demonstrates the viability of preemptible CVMs.

2) WEST AFRICAN DOMAIN

The case study for the West African domain involves a precipitation event spanning 18–19 August 2017 during the WAM. A 0000 UTC 17 August 2017 initialization is chosen because of the presence of an inland precipitation band in the NCEP–NCAR reanalysis for daily precipitation rate between 18 and 19 August 2017 ([Fig. 3](#)), giving a forecast with 24–48 h of lead time, and allowing for model spinup to occur before the event. The methods which result in the greatest cloud-cost savings in [section 3](#) are applied to

FIG. 3. NCEP–NCAR reanalysis of daily surface precipitation rate over Africa, plotted by the Earth System Research Laboratory Physical Sciences Division (ESRL PSD, available at <http://www.esrl.noaa.gov/psd>). Contours show accumulated precipitation over 24 h (mm).

this case study (Table 2; WAM-1), to illustrate the applicability of these methods in a different geographic region, and to demonstrate how experimenting with different physics schemes can be done affordably on the cloud (section 4).

The forecasts are evaluated against the raw GFS forecast that was used for initialization, and against calibrated satellite gauge precipitation datasets from NASA's Global Precipitation Measurement (GPM) Mission, specifically the final research runs of the 0.1° Integrated Multisatellite Retrievals for GPM (IMERG; Huffman et al. 2017). Although sophisticated post-processing and gridded verification techniques can be applied to the forecasts, they are beyond the scope of this study.

3. Results and discussion for the Arctic domain

a. Effect of compilers and WRF version on simulation time

The results of the Arctic-1 test investigating simulation time differences between different compiler and ARW versions are summarized in Table 3. The

GNU compilers, version 5.3.1, and MPICH, version 3.1.3, implementation used are newer than the ones used in SWMSS (versions 4.4.7 and 3.1, respectively). WRF, version 3.7.1, compiled with the newer GNU compilers runs substantially faster (251-min run time) than with the older GNU compilers (320 min). Intel MPI is the fastest for version 3.7.1, requiring 188 min for the Arctic-1 run using 64 vCPUs. PGI OpenMPI is only 11 min slower than Intel MPI, requiring 199 min to complete the simulation.

The newer version of WRF, version 3.9.1.1, is slower than version 3.7.1. The GNU-compiled WRF, version 3.9.1.1, takes an extra 34 min compared to version 3.7.1. The PGI-compiled WRF, version 3.9.1.1, is slower by 52 min. WRF compiled with the PGI suite, however, runs faster than WRF compiled with GNU, and based on its close performance with Intel MPI for the version 3.7.1 simulation, PGI is a good choice for those not wanting to pay for a compiler.

In between versions 3.7.1 and 3.9.1.1, changes have been made to the WRF software, including bug fixes and improvements to several physics schemes. These changes may have affected the compute time in between the two versions, when they are compared using the Arctic-1 test on the GCP.

The total times taken for the compute and I/O portions of each simulation, as percentages of the total simulation time, are shown in Fig. 4. Though I/O takes approximately 30 min total for each simulation, compute time varies between WRF and compiler versions. The newer WRF version spends more time on the time steps taken by the 4-km domain, while version 3.7.1 spends similar amounts of time on the 12- and 4-km domains.

b. Effect of compute virtual machine size on simulation time

Compute and I/O timings for the static time step simulations for different CVM sizes, including restart file creation time, are shown in Fig. 5 (Arctic-2). Compute time generally increases with CVM size (number of vCPUs per CVM), while output time generally decreases with CVM size. Larger CVMs negatively affect latency while improving write speed. Latency is the time delay between when data is first offered for a calculation between processes and when the result is returned for future use (i.e., in the next model integration forward in time).

When the adaptive time step is used, the total run times for each CVM size are all shortened by over 1 h (Fig. 6). Though compute time increases with CVM size from 4 vCPUs to 32 vCPUs per CVM, there is some decrease in the compute time of the simulation

TABLE 3. WRF version and MPI implementation run times for the benchmark simulation, initialized at 1200 UTC 25 Sep 2015 (Table 2; Arctic-1). The simulations are conducted with the adaptive time step using 8 CVMs with 8 vCPUs each.

	GNU MPICH, version 3.1.3	PGI OpenMPI, version 1.10.2	Intel MPI, version 5.1.1
ARW, version 3.7.1	251 min (version 3.1: 320 min.)	199 min	188 min
ARW, version 3.9.1.1	285 min	251 min	—

with one 64-vCPU CVM compared to the simulation with two 32-vCPU CVMs. This slight decrease in the compute time, alongside its relatively low I/O time for restart and history files (which decreases with increasing CVM size), results in the 64-vCPU CVM simulation having the lowest total simulation time across all CVM sizes.

The low simulation time for the 1 CVM $\times$ 64 vCPU configuration may be attributed to network load variability on the GCP. Figure 7 shows the averaged compute, history, and restart output times for each CVM size between 29 September and 6 October 2017 (Arctic-3). The averaged results show that compute time increases monotonically with CVM size, while history and restart I/O decreases. The low compute time for the 1 CVM $\times$ 64 vCPU simulation from 28 September 2017 (Fig. 6; not included in the averaging) lies within

one standard deviation of the mean compute time. The standard deviation for compute time also increases with an increasing CVM size, and decreases for I/O with an increasing CVM size.

Additionally, the averaged total simulation time is lowest for the 4 CVM $\times$ 16 vCPU configuration, in contrast to SWMSS, which found that 8 CVM $\times$ 8 vCPU was optimal. This difference may be due to the differences in WRF version, compiler, and output frequency. Assuming an average simulation time of 5.75 h, one simulation on nonpreemptible CVMs would cost \$13.05, or \$4,763.25 yr⁻¹, for one run each day. With preemptible CVMs, the equivalent simulation would cost \$2.76, or \$1,007.40 yr⁻¹. This simulation cost could be further reduced if restart or history files are needed less frequently, because simulation time decreases with a decrease in output frequency.

FIG. 4. Bar graph of the total simulation times for each compiler and WRF version, separated by compute and history output times for the 1200 UTC 25 Sep 2015 initialization (Table 2; Arctic-1). The compute portions of the bars are further subdivided by total time spent in each nested domain. The scale on the ordinate is set to match with Figs. 5–7. Timing is from a single measurement. Smaller time is better.

FIG. 5. As in Fig. 4, except the bars are plotted for different counts of compute virtual machines (CVM) and virtual central processing units (vCPU) per CVM, and for a different forecast initialization. Simulation times are divided into compute, history, and restart components for the 0000 UTC 28 Sep 2017 initialization (Table 2; Arctic-2). The results shown are from the simulation using static time steps. Node configurations are identified by (number of CVMs) $\times$ (number of vCPUs per CVM), for a total of 64 vCPUs used for each simulation. Timing is from a single measurement. Smaller time is better.

c. Effect of compression on egress costs

Data egress fees contribute significantly to the total simulation cost, and should be kept to a minimum. Assuming that only hourly history output files are kept, the total size of the Arctic-2 simulation is 60 GB. Long-term storage costs on the GCP can range from \$0.42 to store 60 GB in cold storage for 1 month (\$13.02 for 31 forecasts per month), to \$1.56 month⁻¹ for high-performance multiregional data storage (\$48.36 for 31 forecasts per month; <https://cloud.google.com/storage/pricing>). Egress is substantially more expensive, costing \$7.20 to egress one 60-GB forecast, or \$214.60 month⁻¹, but can be reduced if file compression is used. If full history files are not needed, then postprocessing can be done on the GCP, and much smaller (e.g., point forecasts for a modest number of locations) output files are egressed.

Figure 8 shows the compression factors for the first 48 h of history files for the 28 September 2017 simulation, as a function of grid spacing (Arctic-2). The compression factor increases linearly from compression levels 1 to 9, and the inclusion of the shuffling option improves the compression factor by 10%. Including dimension conversion

does not substantially change the compression factor. The compression factor does change over the course of a simulation, due to changes in the spatial homogeneity of the meteorological fields. Regardless of the variability in compression factor, compressing before data egress or storage is successful in reducing file sizes, ranging from just under 47% compression factors in the 36-km files (roughly 28 MB saved per file) to 54% compression factors in the 4-km files (roughly 200 MB saved per file) when shuffling is used with level-9 compression.

Though compression factors increase linearly with the compression level, the increase in compression time is nonlinear (Fig. 9). The result is compression times of over 50 s for 4-km history files with shuffling. In most applications, this should not be a large factor, because file compression and egress can occur as the history files are being produced, and total compute time is negligibly affected by simultaneous egress. Users should run `nccopy` with level-9 compression and shuffling enabled if they require maximum file compression (option `-d 9-s`). The average cost and compression timings for a full Arctic simulation with hourly output of files are

FIG. 6. As in Fig. 5, except the results shown are from the simulation using adaptive time steps (Table 2; Arctic-2). Timing is from a single measurement. Smaller time is better.

summarized in Table 4. When level-9 compression with shuffling is used, it would cost \$1,400 less annually than egressing uncompressed files.

d. Viability of using preemptible compute virtual machines

The last part of the testing involves experimenting with preemptible CVMs. If any CVM involved in a simulation is preempted, the simulation stops and is automatically restarted from its last restart file (Arctic-3). Figure 10 shows the preemption counts for different dates and initialization times across each CVM configuration. Preemption counts are only shown for runs that have finished successfully (a simulation restart is tried up to nine times before failing). Runs that failed are marked with “×,” and completed runs are marked with “+.”

Of the 56 fully finished runs, there were 63 total preemptions for the 0000 UTC initializations (23 simulations preempted at least once), and 44 preemptions for the 1200 UTC initializations (17 simulations preempted at least once). This disparity in the preemption frequency indicates that the 0000 UTC–initialized simulations (started at 0500 UTC; i.e., nighttime in North America) have to compete with higher demands from other GCP users. This could be because more large computing runs are scheduled to start overnight in North America, whereas scripting and small-scale tests are done during

the day. Maintenance periods and host migrations also generally occur overnight. When demands for certain machine types are higher, preemptible instances with that machine type are more likely to become preempted. Given that smaller (e.g., 4 vCPU), cheaper CVMs are likely in higher demand than larger CVMs (e.g., 64 vCPU), simulations with these smaller machines are more likely to be preempted. Further, because the 16-CVM simulations employ more CVMs than others, it is more likely that one of those CVMs will be preempted, causing the whole simulation to stop.

Of the 24 failed runs, 13 occurred from the 0000 UTC initialization, and 11 occurred from the 1200 UTC initialization. Most of the failures are due to unbootable CVMs (Table 5), which can occur when CVMs are migrated to a different physical host by the GCP, but kernel updates are not applied properly due to the lack of live migration for preemptible instances (<https://cloud.google.com/compute/docs/instances/live-migration>). Typically, the only way to fix an unbootable CVM is to replace it. This was not done automatically in this case, leading to high rates of failure especially for the 4 vCPU per CVM configuration. Failures due to incomplete restart files are also relatively common, which can occur if a run is preempted during a write to a restart file. This scenario (about a third of failures) can be prevented by automatically checking the file sizes of

FIG. 7. As in Figs. 5 and 6, except the bars represent average simulation times for each CVM size, with averages calculated from successful runs between 29 Sep and 6 Oct 2017 for 0000 and 1200 UTC initializations (Table 2; Arctic-3). Error bars show one standard deviation about the mean. Smaller time is better.

the latest restart files to ensure that they are complete. User scripts should restart from a previous set of files if the sizes of the most recent set are smaller than expected.

Though restarting can add minutes to the total simulation time, the net effect of preemption is not the dominant factor when calculating simulation times (Table 6). An analysis of variance (ANOVA) was done to determine which of two factors contributed more to variability in simulation time: average 4-km adaptive time step (which varies with the Courant number) and preemption count. The bottom row of Table 6 includes the CVM size as an independent variable, which implicitly is also tied to preemption count. The sum of squares (SS) as a percentage of the total sum of squares (SST) is presented in the rightmost column, and represents the contribution of a factor to the total variance in total simulation time. The residual represents factors unaccounted for in the ANOVA, such as variability in the network speed of the data center.

For all CVM sizes, the residual is by far the largest. Changes in the average 4-km adaptive time step do not have a large effect on the variability of the simulation time, and similar ANOVAs conducted using the average time steps from the parent domains have similar results

(not shown). The preemption count has a larger effect on the total variance than the time step in general, though it is still overshadowed by the residual. Therefore, the high variability in total simulation times is due to unexplained sources, but could be associated with the ambient load on the data center.

e. Summary of section 3 results

- 1) For our particular cloud setup, WRF, version 3.7.1, compiled with Intel results in the fastest run time. If users do not want to pay for the cost of a compiler license, the PGI Community Edition suite of compilers is a good alternative to Intel. Because WRF, version 3.9.1.1, is slower than version 3.7.1 when equivalent compilers are used, users who are more sensitive to costs or run times may decide that running WRF, version 3.7.1, makes more sense for their application. However, WRF, version 3.9.1.1, is a more updated version of the model, and is still reasonably fast.
- 2) As CVM size increases while holding the total number of vCPUs constant, the average compute time of a simulation increases, but the average output time of files decreases (Fig. 7). The respective variabilities for compute and output times change with

FIG. 8. Compression factors for each history file for the first 48 h of the 28 Sep 2017 simulation (Table 2; Arctic-2). Compression levels are color coded, and additional options for the nccopy utility are specified in each subplot title. Higher values indicate greater compression, implying less expensive egress costs. The large compression factors for the initial files are due to microphysics- and convection-related variable fields being 0 at the beginning of the simulation.

CVM size in a similar fashion. The 4 CVM $\times$ 16 vCPU configuration provides the lowest average simulation time, when history files are output every 30 min and restart files are output every 3 h.

- 3) Level-9 compression with shuffling produces the greatest compression factor, resulting in the smallest compressed file size. The lengthy compression time with this configuration can be mitigated by compressing output files as they are being produced. By postprocessing gridded files directly on the cloud and only outputting point-forecast files, data egress costs can be largely eliminated.
- 4) Preemptions occur more often during evenings in North America than during daytime, and affect smaller CVMs more frequently than larger ones. If a simulation has a low output frequency, modelers should use smaller CVMs (16 CVM $\times$ 4 vCPU) to minimize compute time. If a simulation has a higher output frequency, or users require fewer preemptions, the larger CVMs (1 CVM $\times$ 64 vCPU) are the

better option. The mid-sized CVMs (4 CVM $\times$ 16 vCPU) provide a fair balance. Though restarts can increase total simulation times, the variability is most attributed to other sources, including ambient data center load.

To determine whether our results in section 3 can be applicable to a different geographical region, we test our best-practice methods for the WAM case study in the next section.

4. Illustration of cost-saving measures for the West African monsoon case study

This section details the West African precipitation case study initialized at 0000 UTC 17 August 2017 to show how an experiment involving different physics schemes can be conducted on the cloud (WAM-1). The WRF forecast is made using a 16 CVM $\times$ 4 vCPU configuration, chosen because this configuration was

FIG. 9. As in Fig. 8, except the compression times for each history file are plotted (Table 2; Arctic-2). Smaller times are better.

found to take the least compute time in section 3b. The output frequencies for the history and restart files are halved (1 and 6 h, respectively) compared to the Arctic simulations, for a more reasonable representation of how a daily forecast run might be conducted. Output files are compressed immediately after being produced, using level-9 compression with shuffling. Nonpreemptible instances are used to prevent restarts from affecting simulation times, though the timing should be the same as nonpreempted simulations made with preemptible CVMs. Cost estimates are thus given for the equivalent simulations using preemptible CVMs. Before these estimates are discussed, we first detail the qualitative differences between each of the precipitation forecasts.

a. Qualitative description of the case-study forecasts

The IMERG 6-h-averaged precipitation dataset has a precipitation band over Burkina Faso and Ghana for the period ending 0000 UTC 18 August 2017, with some weaker precipitation over Sierra Leone and over the Atlantic (Fig. 11). The GFS forecast (using output

provided by NCEP) captures the convective precipitation over Burkina Faso and the Atlantic well, but misses the convection over Sierra Leone. On the other hand, the WRF forecast (produced by our own runs) using the S2016 physics suite is able to capture the convection over Burkina Faso, the Atlantic, and Sierra Leone albeit with smaller magnitude than shown in the IMERG set

TABLE 4. Summary of monthly data storage and egress costs, with no compression (option “-d 0” in the nccopy utility) and with compression (level-1 compression with “-d 1” and level-9 compression with “-d 9”). The columns with “-s” indicate compression with memory shuffling enabled. Estimates assume one simulation per day. The bottom row is an estimate of the time needed to compress hourly output for one Arctic simulation (60 GB of uncompressed files) for each compression option (Table 2; Arctic-2).

	Compression flags				
	-d 0	-d 1	-d 1-s	-d 9	-d 9-s
Multiregional storage (\$)	48.36	26.68	21.96	26.24	21.24
Cold storage (\$)	13.02	7.18	5.91	7.13	5.72
Egress (\$)	214.60	122.88	101.37	120.00	98.03
Compression time (min)	—	37.96	22.73	74.96	151.49

FIG. 10. Preemption counts for successful runs between 29 Sep and 6 Oct 2017 (Table 2; Arctic-3). Successful runs are marked by “+” above the bars, and failed runs (≥ 10 preemptions) are marked by “x.” Hatched bars are to distinguish the 1200 UTC runs from the 0000 UTC runs. Fewer preemptions are better.

(only 12-km domain shown for WRF plots; 36- and 4-km domains have similar results). Both the GFS and WRF models also forecast precipitation in locations that did not appear in the IMERG set, with the GFS predicting strong convection along the border between Niger and Nigeria, and WRF forecasting precipitation over southwest Mali.

Beyond 24 h, both models have trouble predicting specific precipitation bands that qualitatively agree with the IMERG dataset. In particular, both models miss the precipitation event over Senegal and the Gambia for the period ending 1200 UTC 18 August 2017, as well as the band over the Ivory Coast. The GFS also misses the precipitation over Sierra Leone and Liberia for the period ending 0000 UTC 19 August 2017, while WRF underpredicts the magnitude. However, the WRF Model predicts heavy rainfall over Burkina Faso and the Ivory Coast on that day, which never materialized. Though the WRF Model using the S2016 set of physics does not produce a precipitation field that agrees well with the IMERG dataset beyond 24 h, a meteorologist could experiment with a different set of physics that may produce a better forecast. This study can be done quickly and affordably on public cloud infrastructure due to its flexibility and on-demand cost model, without having to purchase an HPC cluster or queue on shared resources.

Recently, NCAR started to provide preconfigured name lists that specify physics suites best suited for particular use cases. One such suite is the convection-permitting (C-P) physics suite (C-P; Table 1), used in the NCAR Model for Prediction Across Scales (MPAS) and available in WRF (Skamarock et al. 2012). Unlike the WRF simulation with the S2016 set of physics, the C-P suite misses the precipitation band over Burkina Faso. However, precipitation over Senegal and The Gambia is better captured. It is also able to simulate the small convective cells in northeastern Mali, which were missed in the GFS simulation. Placement of the heavier

TABLE 5. Reasons for simulation failure, separated by counts of occurrence for each CVM size, for the simulations between 29 Sep and 6 Oct 2017 (Table 2; Arctic-3). Runs are marked as failed when ≥ 10 preemptions occur for a single simulation; when a restart is attempted from an incomplete file that was being written when a preemption occurred; or when a CVM becomes unbootable, and thus inaccessible for MPI communications.

	vCPU/CVM				
	4	8	16	32	64
Preempted ≥ 10 times	2	2	0	0	0
Failed from incomplete restart file	1	4	1	1	0
Failed from unbootable CVM	10	2	1	0	0

TABLE 6. ANOVA table summarizing factors influencing total simulation time, for each CVM size as well as a summary across all sizes, computed for simulations between 29 Sep and 6 Oct 2017 (Table 2; Arctic-3). The contribution of a factor to the variance in simulation time is represented by the sum of squares (SS) as a percentage of the total sum of squares (SST). The 16 CVM $\times$ 4 vCPU row has been excluded, because not enough runs finished successfully to extract meaningful results.

	Factor	SS/SST $\times$ 100 (%)
8 CVM $\times$ 8 vCPU	Avg 4-km adaptive time step	18.674 961
	Preemption count	4.806 808
	Residual	76.518 231
4 CVM $\times$ 16 vCPU	Avg 4-km adaptive time step	1.515 933
	Preemption count	31.301 963
	Residual	67.182 105
2 CVM $\times$ 32 vCPU	Avg 4-km adaptive time step	0.061 120
	Preemption count	27.309 966
	Residual	72.628 913
1 CVM $\times$ 64 vCPU	Avg 4-km adaptive time step	0.083 309
	Preemption count	7.128 803
	Residual	92.787 887
Summary	Avg 4-km adaptive time step	3.804 175
	CVM Size	1.086 079
	Residual	95.109 746

precipitation cells over Burkina Faso remains a challenge. In general, though the S2016 suite of physics has a tendency to forecast large, spread-out precipitation bands, the C-P suite produces smaller, more focused high-intensity cells.

The different representations of convection between these two simulations can largely be attributed to differences in the cumulus, microphysics, and planetary boundary layer schemes (Klein et al. 2015; Heinzeller et al. 2016). Being able to experiment with different physics schemes or even ensembles using cloud computing could provide forecasters and researchers with a greater understanding of how these heavily coupled processes interact, which may result in better weather and climate predictions for precipitation in the region.

b. Cost estimation for the case-study forecasts

Though both simulations can provide valuable information for local forecasters in addition to the GFS output, they differ quite substantially in cost and total simulation time (Table 7). The simulation time (6.22 h on the GCP) and cost (\$2,139 yr⁻¹, including data egress) of the S2016 set of physics for the West African domain is comparable to the time and cost of the Arctic domain. It is somewhat higher, however, due to the prevalence of strong updrafts from tropical convection. The adaptive time step reduces time step length as

velocities increase to maintain numerical stability. When compared to the 4-km nest in the Arctic-2 simulation, the 4-km nest for the West African domain has on average a 27% shorter time step, resulting in more time steps needed to finish the simulation. The simulation with the S2016 suite (6.22 h) is also nearly 3 h shorter than the simulation with the C-P suite (9.19 h), because the latter requires more sophisticated physics calculations.

When compared against the best Intel-compiled estimate in SWMSS, the West African domain with the S2016 physics suite costs 60% less annually when data egress is taken into account (\$2,139 compared to \$5,500), if preemptible instances and file compression are used. With no data egress, the cost reduction increases to 70% (\$1,088 compared to \$3,532). Because the tests in this study were conducted using free PGI-compiled OpenMPI libraries, the yearly cost savings would be even higher when the annual cost of an Intel compiler is added to the simulation costs (\$1,088 compared to \$6,532 with the Intel compiler, a cost reduction close to 84%).

Unlike the S2016 suite, the C-P suite is too memory intensive to run on the n1-highcpu variety of CVMs (0.9 GB of memory per vCPU), and instead has to use the more expensive n1-standard variety (3.75 GB of memory per vCPU). The output size is also larger, resulting in a total cost that is 60% greater than the S2016 simulation. Though the C-P physics suite is more expensive and memory intensive, its use in an experimental setting is still feasible due to the scalability and flexibility of cloud infrastructure. If users require faster simulations and are able to afford higher costs, they can perform horizontal scaling tests to determine the optimal total vCPU count for their needs. The ability to create compute resources to meet computational and memory requirements on demand is an inherent advantage of cloud infrastructure over HPC clusters.

For researchers undertaking a time-insensitive year-long study of dynamics and precipitation patterns similar to the WAM case study, the S2016 configuration on the GCP may be an adequate, cost-effective alternative to investing in an HPC facility, or queuing on a shared cluster. However, a 6-h-long simulation may be too slow for time-sensitive real-time forecasting, despite the relative cost effectiveness of the setup. Modelers who require a faster simulation without using more resources could incorporate the following changes:

- Reduce the restart file output frequency. This carries the risk that if a preemption occurs, the simulation would restart at a much earlier point and therefore waste time.
- Attempt additional vertical scaling tests to determine an optimal vCPU count. Though the simulation may

FIG. 11. Global Precipitation Measurement (GPM), GFS, and WRF 6-h-averaged precipitation rates over Africa, with the ending times for each averaging period shown. Only the 12-km WRF forecasts are plotted; the 36- and 4-km forecasts show similar results. The GPM and GFS regions are scaled to match the WRF domain boundaries.

progress faster with more vCPUs, there would be a threshold at which the cost increase would overtake the simulation time decrease.

- Remove any unnecessary variables from writing to history output files that are not needed for archiving or the final forecast products, to reduce the required I/O time.
- Reduce the domain size to forecast only the necessary regions. Users will have to experiment to determine a domain size that will meet timing and cost constraints without resulting in adverse boundary effects over the target area.
- Use simpler physics schemes, with a potential decrease in forecast quality.

Our testing with the two physics suites is not meant to show which suite produces the better forecast. Instead, our results are meant to highlight the cost effectiveness of running WRF in the cloud over a region with deep

tropical convection. Though researchers could perform these experiments on local HPC clusters, those without access to these resources could perform research affordably on public cloud infrastructure, and run experiments quickly without having to queue on shared clusters. Researchers with access to HPC resources could also leverage the cloud as a supplemental compute platform. Users could follow our testing methodology to determine which physics schemes and machine configurations would best suit their forecasting or research needs, while staying within their timing and budget constraints.

5. Summary

Cloud-computing infrastructure provides a flexible and cost-effective alternative for academia and smaller weather services that cannot afford up-front purchases of expensive computer clusters. Although the use of

TABLE 7. Comparison of WAM case-study run times and simulation and egress costs (boldface) for the S2016 and C-P physics suites (Table 2; WAM-1). Egress estimates do not include restart files. The S2016 setup uses a $16 \text{ CVM} \times \text{n1-highcpu-4}$ configuration (0.9 GB of memory per vCPU), and the C-P setup uses a $16 \times \text{n1-standard-4}$ configuration (3.75 GB of memory per vCPU). Extrapolated annual cost assumes daily forecasts with identical simulation times.

	Physics suite	
	S2016	C-P
Total compute time (h)	3.98	7.06
Total history I/O time (h)	0.88	1.03
Total restart I/O time (h)	1.36	1.09
Total simulation time (h)	6.22	9.19
Simulation cost (\$)	2.98	5.88
Output size (-d 9 -s; GB)	24	29
Overall compression factor	0.56	0.60
Egress cost (\$)	2.88	3.48
Total cost per run (\$)	5.86	9.36
Extrapolated annual cost (\$)	2,138.90	3,416.40

cloud-based resources without cost-optimization techniques is expensive, our experiments with preemptible resources and file compression show that costs can be substantially reduced. By combining the use of the PGI Community Edition compiler suite, preemptible resources, and level-9 file compression with memory shuffling, modelers can save 60% in total simulation costs compared to the best estimate stated in SWMSS for an Arctic domain-sized simulation. By postprocessing files on the cloud and egressing point-forecast text files instead of gridded files, the cost saving is increased to 70%. When the price of a \$3,000 Intel license is factored in to the SWMSS annual cost estimate, the cost saving with negligible data egress is increased to 84%.

The timing and cost results from the West African case-study simulation (section 4) are comparable to the Arctic test runs using the same set of physics and same domain sizes (section 3), indicating that time and cost estimates are transferable to similar domains in other parts of the world. Though the convection-permitting physics suite is much more computationally expensive, it produces a precipitation forecast that differs substantially from that using the S2016 suite. Researchers could experiment with different physics schemes and machine configurations to further study short-range ensemble forecasting in the region, or implement interseasonal or yearlong experiments to study WAM dynamics, all without having to invest up-front in HPC cluster hardware.

Despite the relative affordability and flexibility associated with cloud-based approaches to NWP, certain issues still need to be addressed in regards to reliability. Among such issues for new users include developing a

robust CVM-provisioning system to manage cloud-based clusters if one is not available from the resource provider. A CVM-provisioning system provides an interface for users to create, start, and stop compute resources in a fashion similar to an HPC scheduling system. GCP does not have a provisioning system currently available for its users, but AWS does, which may be a factor for new users to consider. Though external proprietary provisioning services are available, users can write their own resource-provisioning scripts. However, users must ensure that CVMs should be started only when MPI processes are about to begin, and turned off once those processes stop, to prevent runaway resource-use costs. Additionally, if preemptible resources are used, the user-created restart scripts must be able to read from complete restart files, and be able to repair or replace any unbootable CVMs.

Future work on the use of cloud-based infrastructure for NWP should include exploring parallel I/O methods to reduce simulation times. In particular, on the recommendation of a reviewer, we found that splitting restart file output by processor using `namelist.input` option “`io_form_restart = 102`” can reduce output times by 1–1.5 h (up to 37.5% decrease in total I/O time) for the Arctic domain. We also plan to explore the use of other cloud providers, such as Microsoft Azure, that support CVMs with Infiniband interconnect. Infiniband interconnect may provide an advantage over the TCP/IP currently used by the GCP for interCVM communications, and result in faster simulation times (Shainer et al. 2011). Further testing with cloud-based forecasting could include running MPAS or the NOAA Finite-Volume Cubed-Sphere Dynamical Core (FV3) in the cloud, to test the feasibility of running next-generation global models on cloud resources.

Acknowledgments. The authors thank BC Hydro, Mitacs, and the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing the funds to perform this research. We also thank the three anonymous reviewers whose comments improved this manuscript.

REFERENCES

- Bauer, P., A. Thorpe, and G. Brunet, 2015: The quiet revolution of numerical weather prediction. *Nature*, **525**, 47–55, <https://doi.org/10.1038/nature14956>.
- Chang, W. Y., H. Abu-Amara, and J. F. Sanford, 2010: *Transforming Enterprise Cloud Services*. Springer, 428 pp., <https://doi.org/10.1007/978-90-481-9846-7>.
- Colle, B. A., and Y. Zeng, 2004: Bulk microphysical sensitivities within the MM5 for orographic precipitation. Part I: The Sierra 1986 event. *Mon. Wea. Rev.*, **132**, 2780–2801, <https://doi.org/10.1175/MWR2821.1>.

- Dudhia, J., 1989: Numerical study of convection observed during the Winter Monsoon Experiment using a mesoscale two-dimensional model. *J. Atmos. Sci.*, **46**, 3077–3107, [https://doi.org/10.1175/1520-0469\(1989\)046<3077:NSOCOD>2.0.CO;2](https://doi.org/10.1175/1520-0469(1989)046<3077:NSOCOD>2.0.CO;2).
- , 2014: A history of mesoscale model development. *Asia-Pac. J. Atmos. Sci.*, **50**, 121–131, <https://doi.org/10.1007/s13143-014-0031-8>.
- Ek, M. B., 2003: Implementation of Noah land surface model advances in the National Centers for Environmental Prediction operational mesoscale Eta model. *J. Geophys. Res.*, **108**, 8851, <https://doi.org/10.1029/2002JD003296>.
- Fink, A. H., and Coauthors, 2011: Operational meteorology in West Africa: Observational networks, weather analysis and forecasting. *Atmos. Sci. Lett.*, **12**, 135–141, <https://doi.org/10.1002/asl.324>.
- Graham, R., 2014: DFID-Met Office Climate Science Research Partnership (CSRP): CSRP phase 1—Final report. Met Office Tech. Rep., 105 pp., https://www.metoffice.gov.uk/binaries/content/assets/mohippo/pdf/5/csrp1_report.pdf.
- Grell, G. A., and S. R. Freitas, 2014: A scale and aerosol aware stochastic convective parameterization for weather and air quality modeling. *Atmos. Chem. Phys.*, **14**, 5233–5250, <https://doi.org/10.5194/acp-14-5233-2014>.
- Grimit, E. P., and C. F. Mass, 2002: Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest. *Wea. Forecasting*, **17**, 192–205, [https://doi.org/10.1175/1520-0434\(2002\)017<0192:IROAMS>2.0.CO;2](https://doi.org/10.1175/1520-0434(2002)017<0192:IROAMS>2.0.CO;2).
- Heinzeller, D., M. G. Duda, and H. Kunstmann, 2016: Towards convection-resolving, global atmospheric simulations with the Model for Prediction Across Scales (MPAS) v3.1: An extreme scaling experiment. *Geosci. Model Dev.*, **9**, 77–110, <https://doi.org/10.5194/gmd-9-77-2016>.
- Hong, S.-Y., J. Dudhia, and S.-H. Chen, 2004: A revised approach to ice microphysical processes for the bulk parameterization of clouds and precipitation. *Mon. Wea. Rev.*, **132**, 103–120, [https://doi.org/10.1175/1520-0493\(2004\)132<0103:ARATIM>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<0103:ARATIM>2.0.CO;2).
- , Y. Noh, and J. Dudhia, 2006: A new vertical diffusion package with an explicit treatment of entrainment processes. *Mon. Wea. Rev.*, **134**, 2318–2341, <https://doi.org/10.1175/MWR3199.1>.
- HPC Advisory Council, 2015: WRF 3.7.1 performance benchmarking and profiling. HPC Advisory Council Tech. Rep., 16 pp., http://www.hpcadvisorycouncil.com/pdf/WRF_Analysis_and_Profiling_Intel_E5-2697_CONUS12KM_CONUS25KM.pdf.
- Huffman, G. J., D. T. Bolvin, and E. J. Nelkin, 2017: Integrated Multi-Satellite Retrievals for GPM (IMERG) technical documentation. NASA Tech. Rep., 59 pp., https://pmm.nasa.gov/sites/default/files/document_files/IMERG_doc_171117b.pdf.
- Huntington, J. L., K. C. Hegewisch, B. Daudert, C. G. Morton, J. T. Abatzoglou, D. J. McEvoy, and T. Erickson, 2017: Climate engine: Cloud computing and visualization of climate and remote sensing data for advanced natural resource monitoring and process understanding. *Bull. Amer. Meteor. Soc.*, **98**, 2397–2410, <https://doi.org/10.1175/BAMS-D-15-00324.1>.
- Hutchinson, T. A., 2007: An adaptive time-step for increased model efficiency. NCAR Tech. Rep., 4 pp.
- Iacono, M. J., J. S. Delamere, E. J. Mlawer, M. W. Shephard, S. A. Clough, and W. D. Collins, 2008: Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models. *J. Geophys. Res.*, **113**, D13103, <https://doi.org/10.1029/2008JD009944>.
- James, R., and Coauthors, 2017: Evaluating climate models with an African lens. *Bull. Amer. Meteor. Soc.*, **99**, 313–336, <https://doi.org/10.1175/BAMS-D-16-0090.1>.
- Kain, J. S., 2004: The Kain-Fritsch convective parameterization: An update. *J. Appl. Meteor.*, **43**, 170–181, [https://doi.org/10.1175/1520-0450\(2004\)043<0170:TKCPAU>2.0.CO;2](https://doi.org/10.1175/1520-0450(2004)043<0170:TKCPAU>2.0.CO;2).
- Klein, C., D. Heinzeller, J. Bliefernicht, and H. Kunstmann, 2015: Variability of West African monsoon patterns generated by a WRF multi-physics ensemble. *Climate Dyn.*, **45**, 2733–2755, <https://doi.org/10.1007/s00382-015-2505-5>.
- Mass, C. F., and Y.-H. Kuo, 1998: Regional real-time numerical weather prediction: Current status and future potential. *Bull. Amer. Meteor. Soc.*, **79**, 253–263, [https://doi.org/10.1175/1520-0477\(1998\)079<0253:RRTNWP>2.0.CO;2](https://doi.org/10.1175/1520-0477(1998)079<0253:RRTNWP>2.0.CO;2).
- McCollor, D., and R. Stull, 2008: Hydrometeorological accuracy enhancement via postprocessing of numerical weather forecasts in complex terrain. *Wea. Forecasting*, **23**, 131–144, <https://doi.org/10.1175/2007WAF2006107.1>.
- 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, <https://doi.org/10.1016/B978-0-12-803192-6.00016-5>.
- MeteoWorld, 2017: WMO severe weather forecasting demonstration project expands to West Africa. WMO, <https://public.wmo.int/en/resources/meteworld/wmo-severe-weather-forecasting-demonstration-project-expands-west-africa>.
- Mlawer, E. J., S. J. Taubman, P. D. Brown, M. J. Iacono, and S. A. Clough, 1997: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *J. Geophys. Res.*, **10**, 16 663–16 682, <https://doi.org/10.1029/97JD00237>.
- Molthan, A. L., J. L. Case, J. Venner, R. Schroeder, M. R. Checchi, B. T. Zavodsky, A. Limaye, and R. G. O'Brien, 2015: Clouds in the cloud: Weather forecasts and applications within cloud computing environments. *Bull. Amer. Meteor. Soc.*, **96**, 1369–1379, <https://doi.org/10.1175/BAMS-D-14-00013.1>.
- Nakanishi, M., and H. Niino, 2009: Development of an improved turbulence closure model for the atmospheric boundary layer. *J. Meteor. Soc. Japan*, **87**, 895–912, <https://doi.org/10.2151/jmsj.87.895>.
- NASA, 2018: SERVIR Global: The regional visualization and monitoring system. SERVIR, <https://servirglobal.net>.
- Niang, A. D., 2011: Operational forecasting in Africa: Advances, challenges and users. *African Weather and Climate: Unique Challenges and Application of New Knowledge*, Boulder, CO, NCAR, https://ral.ucar.edu/csap/events/ISP/presentations/Diongue_weather_forecast_Africa.pdf.
- Pandya, R., and Coauthors, 2015: Using weather forecasts to help manage meningitis in the West African Sahel. *Bull. Amer. Meteor. Soc.*, **96**, 103–115, <https://doi.org/10.1175/BAMS-D-13-00121.1>.
- Park, R. J., 2014: Development and verification of a short-range ensemble numerical weather prediction system for southern Africa. M.S. thesis, Faculty of Natural and Agricultural Sciences, University of Pretoria, 107 pp.
- Powers, J. G., and Coauthors, 2017: The Weather Research and Forecasting Model: Overview, system efforts, and future directions. *Bull. Amer. Meteor. Soc.*, **98**, 1717–1737, <https://doi.org/10.1175/BAMS-D-15-00308.1>.
- Prein, A. F., and Coauthors, 2015: A review on regional convection-permitting climate modeling: Demonstrations, prospects, and challenges. *Rev. Geophys.*, **53**, 323–361, <https://doi.org/10.1002/2014RG000475>.
- Ramamurthy, M., 2016: Data-driven atmospheric sciences using cloud-based cyberinfrastructure: Plans, opportunities, and

- challenges for a real-time weather data facility. *Cloud Computing in Ocean and Atmospheric Sciences*, T. C. Vance et al., Eds., Academic Press, 43–58, <https://doi.org/10.1016/B978-0-12-803192-6.00004-9>.
- Shainer, G., P. Lui, T. Liu, T. Wilde, and J. Layton, 2011: The impact of inter-node latency versus intra-node latency on HPC applications. *23rd Int. Conf. on Parallel and Distributed Computing and Systems*, Dallas, TX, International Association of Science and Technology for Development, 757–005, <http://www.hpcadvisorycouncil.com/pdf/The-Impact-of-Inter-node-Latency-vs-Intra-node-Latency-on-HPC-Applications.pdf>.
- 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, <https://doi.org/10.1175/WAF-D-16-0075.1>.
- Skamarock, W. C., and Coauthors, 2008: A description of the Advanced Research WRF version 3. NCAR Tech. Note NCAR/TN-475+STR, 113 pp., <https://doi.org/10.5065/D6DZ069T>.
- , J. B. Klemp, M. G. Duda, L. D. Fowler, S.-H. Park, and T. D. Ringler, 2012: A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tessellations and C-grid staggering. *Mon. Wea. Rev.*, **140**, 3090–3105, <https://doi.org/10.1175/MWR-D-11-00215.1>.
- Smith, J., 2015: Saving farmers money in tropical West Africa. UCAR, <https://news.ucar.edu/17896/saving-farmers-money-tropical-west-africa>.
- Thomas, S. J., J. P. Hacker, M. Desgagné, and R. B. Stull, 2002: An ensemble analysis of forecast errors related to floating point performance. *Wea. Forecasting*, **17**, 898–906, [https://doi.org/10.1175/1520-0434\(2002\)017<0898:AEAOFE>2.0.CO;2](https://doi.org/10.1175/1520-0434(2002)017<0898:AEAOFE>2.0.CO;2).
- Thompson, G., P. R. Field, R. M. Rasmussen, and W. D. Hall, 2008: Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. Part II: Implementation of a new snow parameterization. *Mon. Wea. Rev.*, **136**, 5095–5115, <https://doi.org/10.1175/2008MWR2387.1>.
- Unidata, 2017: NetCDF version 4.5.0-rc1. UCAR/Unidata, <https://doi.org/10.5065/D6H70CW6>.
- Vance, T. C., 2016: A primer on cloud computing. *Cloud Computing in Ocean and Atmospheric Sciences*, T. C. Vance et al., Eds., Academic Press, 1–13, <https://doi.org/10.1016/B978-0-12-803192-6.00001-3>.
- Vellinga, M., M. Roberts, P. L. Vidale, M. S. Mizielski, M. E. Demory, R. Schiemann, J. Strachan, and C. Bain, 2016: Sahel decadal rainfall variability and the role of model horizontal resolution. *Geophys. Res. Lett.*, **43**, 326–333, <https://doi.org/10.1002/2015GL066690>.
- Voorsluys, W., J. Broberg, and R. Buyya, 2011: Introduction to cloud computing architecture. *Cloud Computing: Principles and Paradigms*, Wiley, 3–37, <https://doi.org/10.1002/9780470940105.ch1>.
- WMO, 2016: WMO Severe Weather Forecasting Demonstration Project (SWFDP). WMO, <https://www.wmo.int/pages/prog/www/swfdp/>.
- Yuan, M., 2016: Conclusion and the road ahead. *Cloud Computing in Ocean and Atmospheric Sciences*, T. C. Vance et al., Eds., Academic Press, 385–391, <https://doi.org/10.1016/B978-0-12-803192-6.00020-7>.
- Zhang, D., and R. A. Anthes, 1982: A high-resolution model of the planetary boundary layer—Sensitivity tests and comparisons with SESAME-79 data. *J. Appl. Meteor.*, **21**, 1594–1609, [https://doi.org/10.1175/1520-0450\(1982\)021<1594:AHMOT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1982)021<1594:AHMOT>2.0.CO;2).