

Article

Cloud Computing for Climate Modelling: Evaluation, Challenges and Benefits

Diego Montes ^{1,2,*}, Juan A. Añel ², David C. H. Wallom ³, Peter Uhe ⁴, Pablo V. Caderno ⁵
and Tomás F. Pena ⁶

¹ ESEI, Universidade de Vigo, Campus As Lagoas, 32004 Ourense, Spain

² EPhysLab & CIM-UVigo, Campus As Lagoas, 32004 Ourense, Spain; j.anhel@uvigo.es

³ Oxford e-Research Centre, University of Oxford, Oxford OX1 3QG, UK; david.wallom@oerc.ox.ac.uk

⁴ School of Geographical Sciences, University of Bristol, Bristol BS8 1SS, UK; peter.uhe@bristol.ac.uk

⁵ Wargaming, Sydney, NSW 2007, Australia; kaderno@gmail.com

⁶ CITIUS, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain; tf.pena@usc.es

* Correspondence: kabute@uvigo.es

Received: 1 June 2020; Accepted: 17 June 2020; Published: 22 June 2020



Abstract: Cloud computing is a mature technology that has already shown benefits for a wide range of academic research domains that, in turn, utilize a wide range of application design models. In this paper, we discuss the use of cloud computing as a tool to improve the range of resources available for climate science, presenting the evaluation of two different climate models. Each was customized in a different way to run in public cloud computing environments (hereafter cloud computing) provided by three different public vendors: Amazon, Google and Microsoft. The adaptations and procedures necessary to run the models in these environments are described. The computational performance and cost of each model within this new type of environment are discussed, and an assessment is given in qualitative terms. Finally, we discuss how cloud computing can be used for geoscientific modelling, including issues related to the allocation of resources by funding bodies. We also discuss problems related to computing security, reliability and scientific reproducibility.

Keywords: climate model; cloud computing; supercomputer

1. Introduction

The continuous and rapid increase in computing power has been a major factor in the progress of numerous scientific disciplines over the last few decades. Increased computing power in the field of climate modelling is leading to more accurate assessments of the impact of climate change [1]. This implies huge challenges from the point of view of both hardware and software, one of the most important being the ever-increasing volumes of data generated by both observation and simulation. This change, and therefore, the commensurate cycle of requiring ever-greater computational resources, is one that is happening across nearly all research domains [2], but is extremely prevalent in the grand challenge areas of climate and geoscience.

Over the last 20 years, many scientists have been running simulations in High Performance Computing (HPC) environments and transferring the output data to local systems for analysis. This was a perfectly reasonable proposition when data volumes were small. However, now that outputs from operational forecasting models are updated hourly and the amounts have reached more than 300 TB per day, this workflow is simply no longer possible. It is, therefore, necessary to bring computing and data together by no longer moving data to computing but computing to data. Hence, the data processing and analytical capabilities associated with the cloud and other distributed computing paradigms are an integral part of future climate modelling.

Cloud computing has emerged in recent years as both a new business model and a sensible technological choice, as it allows users to adapt resources to demand and/or budget relatively easily, reducing the need to manage a computing infrastructure on premises. In the private sector, the migration to cloud computing from traditional IT infrastructure is increasing and is expected to continue over the next few years [3,4]. Beyond the private sector, cloud computing is also increasingly popular in research laboratories around the world [5,6]. For example, in April 2015, Microsoft launched the Azure4Research Climate Data Award Program in support of the White House Climate Data Initiative, and the European Commission established a plan to develop an European Open Science Cloud by the end of 2016 [7] that continues to be developed. Institutions studying weather and climate have begun to explore the use of cloud platforms. For example, the United Kingdom Met Office is developing a distributed data analysis platform that obviates the need for scientists to move data around (<https://aws.amazon.com/solutions/case-studies/the-met-office/>).

Such platforms are designed to run a Hadoop [8] cluster in a hybrid (i.e., local and remote) cloud that shares storage space with the HPC infrastructure, using Python notebooks as the primary interface for the user. The data cluster of the Science and Technology Facilities Council (STFC, <https://www.stfc.ac.uk/>) Centre for Environmental Data Archival works in part as a piece of the cloud computing infrastructure [9]. NOAA has partially externalized its data storage and computing (for example, using Amazon AWS Lambda) through partnerships with major vendors in the framework of the Big Data Project [10]. This approach is also used by the Met Office [11]. Other applications of cloud computing in atmospheric and ocean sciences have been compiled [12] and range from data storage and analysis to visualization. However, these examples correspond more to what is known as Infrastructure as a Service (IaaS). In contrast, the use of cloud services to substitute pure computing power (HPC as a Service (HPCaaS)) has not been explored to the same degree. Already more than ten years ago, a single assessment of a basic cloud computing system was performed [13]. More recently, other experiments have been carried out on weather forecasting [14,15] and climate modelling [16–20].

It is interesting to note the need for and advantages of cloud computing technologies in the increasing framework of climate services. When delivering climate data and information to stakeholders, one of the main issues is the compatibility of formats [21] and IT infrastructure between the provider (for example, a national weather service) and the client. The use of cloud computing as a shared platform between the two parties could help to solve such problems.

However, decisions to move from traditional in-house HPC to cloud computing need careful preparation and studies [6]. Some factors to consider include:

- the suitability of hardware to the particular computing task (e.g., massively parallel tasks, IO intensive tasks);
- the overhead from using virtualization and the ability to optimize code on cloud resources;
- the cost of computational time;
- the requirement of storing data long term and data transfers out of the cloud (and related costs);
- the ability to process and analyse data within the cloud.
- security.
- user interface and ease of use.

For climate research, issues such as reliability and trust in the results are essential for ensuring that the results provided by the cloud services correspond to the computation that was originally requested. Errors from potential hardware failures are common to all kinds of computational systems. The impacts of these on the results, and how to work around the problems caused, have been exposed. To address such issues, a kind of backup infrastructure is proposed in which cloud computing can provide an optimal solution because, by their nature, they are located in hardware facilities much larger than those requested by a single user [22].

Here, to shed some light on the possibilities offered by cloud computing, we explore and discuss these issues, including the computational performance of the various options for running climate

models, their monetary cost, security and possible influences on funding models and scientific reproducibility. We focus on HPCaaS because the goal here is to highlight the ability of cloud services to substitute or complement local computing facilities. In our analysis, we use solutions offered by the three leading market providers according to a previous report [23]: Amazon Web Services (AWS, <https://aws.amazon.com>), Google Cloud Platform (GCP, <https://cloud.google.com/compute> and Microsoft Azure (<https://azure.microsoft.com>).

2. Methods and Results

2.1. Evaluation of Climate Model Performance

In this section, we split our analysis into two different parts: single and multiprocessor climate simulations (spcs and mpcs, respectively). There were several reasons for doing it in this way. First of all, the nature of the experiments that we performed in each cloud platform was different: in spcs, we tested the performance of climate simulations deployed as binary files running on single cores of a processor, avoiding compilation tasks in the cloud. However, for the mpcs case, we directly compiled a model on the cloud platform. Because of the focus of the cloud solution offered, Google Compute Engine (GCE) fit much better to address the mpcs problem than AWS or Azure. Furthermore, something to note is that AWS and Azure are marketed focusing on clients with similar profiles and different from the ones of GCE, therefore making it reasonable to balance AWS with Azure, but not to balance them with GCE. This is addressed in some way later, in the section about user experience, and is clear from statistics about cloud adoption [24,25].

2.1.1. Single Processor Climate Simulations

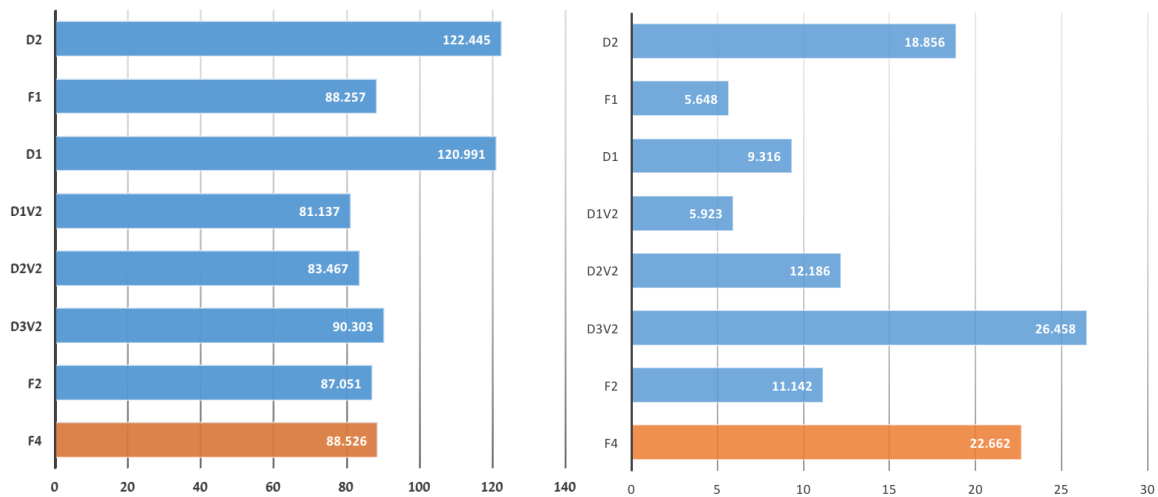
In order to evaluate the options for using cloud services to run models, we performed several experiments focusing on computing performance and cost. The first of these was developed adopting the well-known ClimatePrediction.net (CPDN) infrastructure [26] running Weather@Home2 [27] computational tasks, which uses BOINC [28] as a tool to distribute the computing work. A previous assessment, along with the technical details of this framework can be found in our earlier work [17], in which experiments using AWS and Azure were presented. Details of the configuration are included in Appendix A.

For this purpose, we ran a set of thirteen month climate simulations for CPDN in Azure and AWS using a range of different Virtual Machines (VMs) (with different hardware and allocated resources) in an optimised configuration. The simulations were run using the Met Office Hadley Centre regional atmospheric circulation model HadRM3P [29], at a resolution of 50 km that covered the South America CORDEX (Coordinated Regional Downscaling Experiment) region (e.g., [30]), nested within the global atmosphere-only model HadAM3P. These simulations were run on a single processor and lasted between three and five days depending on the VM type.

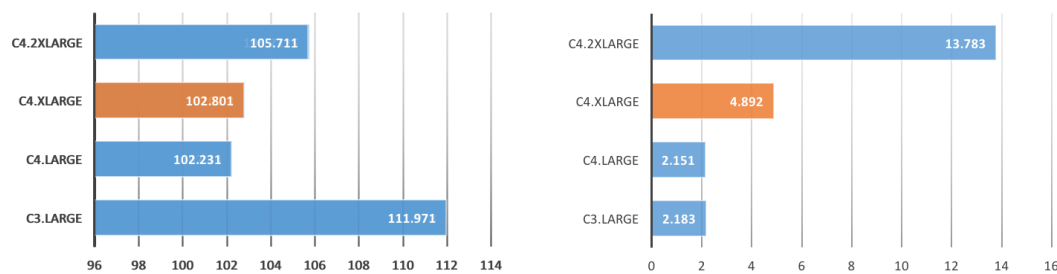
The results are shown in Figure 1. In order to obtain a (theoretically) similar performance between the two cloud providers, we focused on two similar VMs: Azure Standard F4 and AWS c4.xlarge. Details of the hardware are available in Tables 1 and 2. The results showed that the Azure F4 VM was 13.9% faster; however, the AWS c4.xlarge was 4.6 times less expensive. However, we should note that the AWS simulations were run using reduced-cost VMs. These VMs are known as spot instances (<https://aws.amazon.com/ec2/spot>) and are not available from any other vendor. They let us configure the maximum price of the VMs (which changes according to the demand that a given AWS region experiences) and to run or stop the model according to such a limitation. Furthermore, it must be noted that the cost of running over on-demand instead of on spot instances can be up to ten times more expensive [31]. Another conclusion is that the cost of using on-demand AWS VMs is slightly higher than for Azure.

It must be noted that we were not strictly comparing like with like in this instance. Beyond the small technical differences between the servers available in each platform, subtle distinctions in data transfer could also have an impact on the results obtained. For example, an AWS vCPU

(virtual/abstracted computing capacity) is a single thread rather than a dedicated CPU core. To select the best solution, as a means of optimizing performance, a user must evaluate the bottlenecks in a given application considering a wide range of hardware and configuration options. For example, if it is necessary to complete a large ensemble of climate simulations with a model that struggles to give a good performance because of the communication between cores (in a given CPU or full instances), the user could decide on the best provider of the cloud service taking into account such a limitation. On the other hand, the possibility of using less CPUs to run each member of the ensemble and to run several members simultaneously could be assessed.



(a) Azure: Time per Workunit (hours) (b) Azure: Price per Workunit (USD)



(c) AWS: Time per Workunit (hours) (d) AWS: Price per Workunit (USD)

Figure 1. CPDN simulation in Azure and AWS. Orange bars highlight the more similar VMs between vendors.

Table 1. Microsoft Azure Linux virtual machines' technical specifications.

Instance Type	CPU	Memory	Disk
F4	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (4 cores)	8 GB	64 GB SSD
F2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (2 cores)	4 GB	32 GB SSD
D3v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (4 cores)	14 GB	200 GB SSD
D2v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (2 cores)	7 GB	100 GB SSD
D1v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (1 core)	3.5 GB	50 GB SSD
D1	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz (1 core)	3.5 GB	50 GB SSD
F1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz (1 core)	2 GB	16 GB SSD
D2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz (2 cores)	7 GB	100 GB SSD

Table 2. Amazon Web Services instances' technical specifications.

Instance Type	CPU	Memory	Disk
C3.LARGE	Intel(R) Xeon(R) CPU E5-2680v2 @ 2.80GHz (2 cores)	3.75 GB	64 GB (Standard EBS)
C4.LARGE	Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz (2 cores)	3.75 GB	64 GB (Standard EBS)
C4.XLARGE	Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz (4 cores)	7.5 GB	64 GB (Standard EBS)
C4.2XLARGE	Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz (8 cores)	15 GB	64 GB (Standard EBS)

2.1.2. Multiprocessor Climate Simulations

A different technical approach lies in the possibility of running a climate model directly over the cloud by deploying several VMs working as a cluster. An analogy with a Supercomputer (SC) was undertaken using the FinisTerra II from the Centro de Supercomputación de Galicia, in Spain, and the cloud services provided by the Google Compute Engine (GCE), using Debian GNU/Linux 7 Wheezy as the operating system in the VMs. Again, the technical details can be found in Appendix B. The model selected to perform the test was WACCM [32]. Several versions of WACCM have been run in the past in FinisTerra I (the FinisTerra I SuperComputer has been in service from 2007 to 2015 and ranked 101 on the Top500 in 2007; <https://www.top500.org/system/175541/>) and the resulting simulations used for international reports and research papers [33–38]. A summary of the details of past performance is also available [39]. The results for the simulations performed here are shown in Figure 2.

It can be seen that the GCE gave a better performance than the SC for a smaller number of cores/MPI tasks. For example, for 32 cores, the performance was approximately 200% better, but this was considerably reduced when the simulation was more demanding and when more MPI tasks were used. The model throughput showed clearly how the SC performance was better after approximately 100 cores. Apart from technical differences between the processors of the SC and the GCE, it was plausible that the main cause of the inferior performance of a public cloud computing solution for bigger tasks was the interconnection network. The GCE features a speed of 1.9 Gbits/s when connecting computing nodes, while the SC has an Infiniband delivering 19 Gbits/s. Similar conclusions were reached for testing in AWS and ARCHER (U.K. National Supercomputing Service) when running the HadGEM3 model [40].

An analysis of the costs associated with these simulations showed how the GCE was both systematically and substantially less expensive than the SC, based on standard rates charged by the supercomputing centre to external users and GCE pricing (see Figure 2).

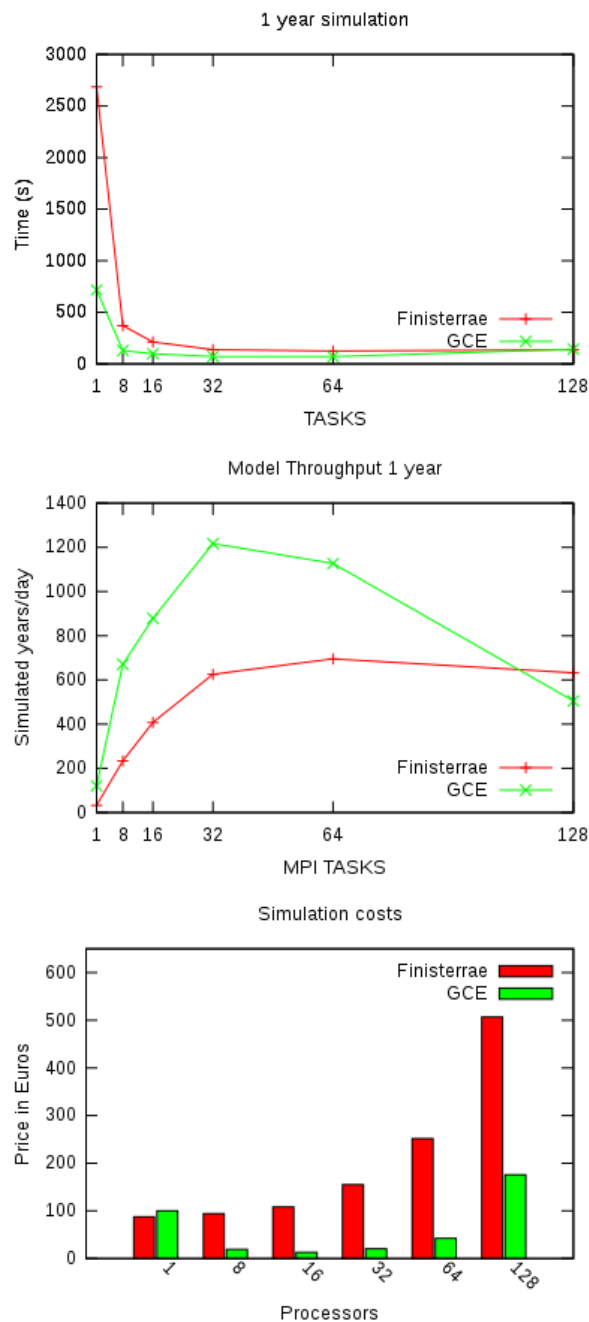


Figure 2. Performance and price for WACCMruns in Google Compute Engine (GCE) versus FinisTerraEII.

2.1.3. User Experience of Cloud Vendors

Different vendors of cloud computing services offer different products, meaning that one might fit a user's needs better than another. This could be related to user experience and quality of service, which can be assessed under the umbrella of what is known as "cloud resource orchestration" [41]. For example, after running our simulations, we could make the following specific observations:

- The prices previously described were based on standard rates; however, different discounts and specific payment plans can be discussed and negotiated directly with providers. Running simulations has costs associated with storage and transfer data. In some cases, these associated costs can be completely insignificant [17], but also can be slightly more expensive than for an

SC (e.g., comparing the ARCservices (<https://help.it.ox.ac.uk/arc/services>) provided by the University of Oxford to AWS) [31].

- For simulations using large ensembles with BOINC, for example, the main limiting factor is the CPU, not the memory [17]. However, when running a model directly over a cloud service (as, in this case, for the GCE), constraints very similar to a supercomputer are found (parallelization, network communication and memory). However, a given vendor could provide solutions for the issue of memory and CPU without any problems. These details can be negotiated directly with providers.
- AWS API calls (and related tools) are well documented and easy to integrate (different SDKs are available). Azure's API (and tools) have good documentation, but still have some way to go to achieve the same level as AWS.
- Writing code for Azure seems to be more oriented towards .NET developers than towards the general public, which made it difficult for us to create extensive automation for our simulations such as the agnostic/generic management of hundreds of VMs.
- In the same vein as AWS, the GCE provides an infrastructure that simplifies both the deployment of simulations and the use of VMs.
- AWS, Azure and GCP provide similar basic security mechanisms and systems: access control, audit trail, data encryption and private networks [42,43]. This was relevant for our tests as we wanted to assure the reproducibility and data validation (as well as the results' distribution), so it was required that the data integrity was guaranteed. All the evaluated cloud providers have data encryption available for both local and distributed (AWS S3, Google Cloud Storage (GCS) and Azure Storage). The security features (for the three providers) are easy to setup (and sometimes just out-of-the-box, like on the distributed storage). It is worth mentioning that the tested providers manage and process very sensitive data (such as governments' and medical information), so they have to comply with the highest security standards like SOC (Service Organization Control) or ISO/IEC 27001 and pass periodic audits [44,45].

On a more high-level approach, selecting a platform/infrastructure (SC vs. cloud) is not trivial, it requires the evaluation of different aspects that will probably have specific weights depending on the model and the experiment or simulation [6]. In Table 3, we provide some general observations, with the main advantages and disadvantages of an SC and the different vendors assessed.

Table 3. Summary of the platforms pros and cons.

Platform	Pros	Cons
Supercomputer	<ul style="list-style-type: none"> • Well known and very predictable environment. • Better institutional support and budget. 	<ul style="list-style-type: none"> • Limited elasticity and scalability. • Usually, shared environment. • Expected high queue wait times.
AWS	<ul style="list-style-type: none"> • Public cloud providers' leader. • Best support. Biggest number of solutions and integrations. 	<ul style="list-style-type: none"> • Cost optimization can be complex to understand. • Services are tailored to AWS; easy to get into a vendor lock-in situation.
Azure	<ul style="list-style-type: none"> • Best option for Windows-based software. • Very competitive pricing and waivers. 	<ul style="list-style-type: none"> • GNU/Linux-based simulations are not the ideal case for Azure. • Generally speaking, less mature than AWS.
GCP	<ul style="list-style-type: none"> • Appealing and comprehensive pricing model based on usage. • In many cases, services are easier to manage than with other providers. 	<ul style="list-style-type: none"> • Some of the services are still in the very early stages. • Very vanilla; this can also be seen as an advantage in some cases.

3. Discussion

While it might appear that for (public) cloud computing, there are no limitations on the computing power that a user can access, this perception is misleading. With cloud computing, there is a shift away from competing for computational resources with other users of the same SC to being limited by the computational resources that a user can afford. This can, in turn, make funding bodies and researchers more aware of the real and total cost of funding the research. Indeed, cloud computing was recommended some years ago as an option for inclusion when budgeting for research projects with

HPC needs [46]. This idea is consistent with the definition of cloud computing as a business model and points to the growing importance of moving to a public cloud computing infrastructure as a form of privatization or externalization of part of the research process.

It must be remembered that the use of an SC option implies large overheads for manual operations and thus a need for in-house staff dedicated to solving technical issues, rather than providing support for activities that maximize the scientific output of a project, such as more complete or additional analysis of the data, or better organization. An SC option also implies the need for regular hardware updates and upgrades.

A real example in the field of atmospheric sciences (and rather an exception, as it was built from scratch instead of using an external provider) is the model CloudMUSC supported by the Norwegian Meteorological Institute and run on cPouta (<https://www.csc.fi/en/web/atcsc/-/pilvilaskenta-avuksi-saamallien-kehitykseen>), a cloud service based on OpenStack (<https://research.csc.fi/pouta-user-guide>). However, researchers in the atmospheric sciences have been using models since the early studies in the 1950s [47], and transferring these to a cloud-based system could require a considerable upfront investment. Indeed, it is expected that over the next few years, the migration of applications to cloud computing services is a must, and most of the investment will be necessary for migration of applications or development from scratch to adapt them to the cloud [4].

One possible scenario is where data are stored in a cloud service. This can limit the cost of maintaining infrastructure for providers of large datasets (e.g., satellite data, reanalysis). In this way, users can contribute to the maintenance of the repository through payment for data transfer to their local machines or provision of VMs in the cloud to perform research using the data. Data transfers might be faster where mirror copies are established in geographical regions. In such a scenario, the budget allocation could shift partly from data providers to project funding, because budgeting for projects could provide a more realistic idea of the cost of using the data. A real example using a commercial vendor (although without any associated fee for the user) is the availability of the recent ERA5 Reanalysis [48] produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) from AWS (<https://registry.opendata.aws/ecmwf-era5/>). This is done through the known as “AWS Public Dataset Program (<https://aws.amazon.com/opendata/public-datasets/>)”. Furthermore, the CESM LENS climate simulations [49] are made available by the National Center for Atmospheric Research (NCAR) from AWS (http://ncar-aws-www.s3-website-us-west-2.amazonaws.com/CESM_LENS_on_AWS.htm).

It has been claimed that the adoption of cloud computing for research purposes could and should increase [50,51]. In 2012, the U.S. National Science Foundation (NSF) funded several projects concerned with applying this technology in environmental sciences, which had already been evaluated as early as for the period 2009–2011.

In general terms, a boost in investment in cloud computing might be expected over the next few years to consolidate infrastructure as part of attempts to improve data-sharing services in the scientific community. As an example, EU Horizon2020 planned to devote billion euros to cloud computing [7], and projects like the European Open Science Cloud-Synergy (<https://www.eosc-synergy.eu>) or the NSF BIGDATA program [52] have been very recently launched. Success stories are the EarthCube program, active since 2013 [53], or JASMIN (<http://www.jasmin.ac.uk>).

4. Conclusions

Whilst we have not assessed all the market providers for cloud computing services, we can nevertheless state that to be successful in the field of meteorological and climate research, cloud computing should deal effectively with some of the major concerns for any new technology: cost, improvement of daily work and the generation of new opportunities. The costs of cloud services continue to be high, mainly associated with permanent data storage and transfer, but also with computing. A more affordable option could be a private cloud solution [54]. However, cloud computing provides flexibility and is a sensible option when considering responsiveness. With

cloud computing services, it is possible to perform tasks very quickly, making research results relevant and timely. For example, the combination of cloud computing with BOINC [17] has the potential to “democratize” access to computing resources by researchers or institutions that do not have the capacity to host and maintain an in-house HPC facility.

However, it should be borne in mind that beyond any monetary arrangements made by institutions or organizations (as key accounts), the low cost of cloud computing services could be affected by the existence of market challengers. Market challengers have a loss related to the price they offer, intending to gain market share. Therefore, any migration of infrastructure to a cloud service should be undertaken with caution, taking into account that prices could increase in the future to reflect actual costs. In order to considerate all the variables, several methodologies to assess the “return on investment” of migrations to cloud computing services have been proposed and are available (e.g., [6,55,56]). Furthermore, each provider uses a different billing scheme [57].

All this is also the reason why the analysis performed here was not comprehensive from the point of view of performing every single simulation across all the cloud computing platforms used. That is, this was a feasibility and options study. A complete comparison would not make sense because the best solution for each case depends on the model used, its code, the infrastructure offered by a vendor at a given time and the price available. Therefore, an apples to apples comparison would not be possible, and consequently, it would not be more informative than the experiments exposed here.

Moreover, users need to evaluate issues related to security when deciding whether a cloud computing environment is the best solution for them, or when considering which approach to cloud computing best fits their needs. Methodologies to evaluate risks associated with the use of cloud computing have already been proposed (e.g., [58]). The perception persists that the security of in-house computing is better than it is for cloud services [59]. However, sometimes, such perceptions are wrong. Commercial providers usually have certifications such as ISO/IEC 27001 (<https://www.iso.org/standard/54534.html>) that are rarely obtained by in-house HPC facilities; for example, the European Union Agency for Network and Information Security is currently developing Cloud Certification Schemes related to security. Furthermore, where necessary, it is generally possible to deploy mixed environments with both private and public cloud services as an intermediate solution [60]. It is usually the case that the providers of cloud services care about physical security and the issues related to infrastructure. However, issues related to data transfer, applications, etc., are the responsibility of the customer [61].

Another issue is the reproducibility of research. Scientists are working hard to increase the level of reproducibility of published research. Because some computing applications are now inherent to this process, how we make use of them is key to assuring reproducibility. Related to the previous section, the externalization of computational resources could lead to some scepticism about the reproducibility of the results. However, this should not be a problem if providers of cloud computing are audited and receive certification regarding how the computational resources provided comply with reproducibility practices, such as the use of free software [62–64]. Indeed, applied in the right way, cloud computing could be seen as an opportunity to improve trust in research results.

Finally, although the ideas and results expressed here might appear to encourage the adoption of cloud computing, it has been pointed out that at least in the industry, the benefits of such adoption are usually below expectations [65]. Therefore, we suggest that approaches to cloud computing for HPC and its use in geoscientific modelling must be carefully evaluated.

Author Contributions: Conceptualization, all; methodology, D.M., J.A.A., P.V.C. and T.F.P.; software, D.M., P.U., P.V.C. and D.C.H.W.; validation, all; formal analysis, all; investigation, all; resources, all; data curation, D.M., J.A.A. and P.V.C.; writing, original draft preparation, D.M., J.A.A. and P.V.C.; writing, review and editing, all; visualization, all; supervision, J.A.A., D.C.H.W. and T.F.P.; project administration, J.A.A. and D.C.H.W.; funding acquisition, J.A.A., D.C.H.W. and T.F.P. All authors read and agreed to the published version of the manuscript.

Funding: J.A.A. was supported by a “Ramón y Cajal” grant from the government of Spain (RYC-2013-14560) and the EPhysLab by the European Regional Development Fund and Xunta de Galicia grants (ED431C 2017/64-GRC). This research was partially supported by a Microsoft Azure for Research grant.

Acknowledgments: We thank Alberto Arribas from the Met Office Informatics Lab, Paulo Rodríguez from Dropbox, Inc. and Francis Vitt from the National Center for Atmospheric Research (USA) for some useful discussions. Furthermore, we thank the Supercomputing Centre of Galicia.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Appendix A. Infrastructure Details for CPDN Experiments

CPDN simulations use BOINC; therefore, given the nature of this platform and the experiments, the most interesting aspects to evaluate, with the most significant impact, are computing (instances) and (local) storage.

Appendix A.1. Amazon Web Services

Amazon Elastic Compute Cloud (EC2) was the cloud computing service used and evaluated to run the AWS experiments. Additionally, we used Amazon Simple Storage Service (S3) for saving reports and data from the simulation runs. This is the same technology used by the ECMWF and AWS to store the ERA5 Reanalysis (<https://registry.opendata.aws/ecmwf-era5/>).

In the steps described, an instance is launched using the web interface (providing a script for automation); tasks are automatically processed; and it will shut down automatically once the work is completed (when no more tasks are available on the CPDN server).

All these steps are carried out on the AWS Console (<https://console.aws.amazon.com>), under the EC2 service management (by selecting Launch Instance)

- Step 1: Launch Instance.
- Step 2: Select Linux Distribution (Ubuntu Server 16.04 LTS).
- Step 3: Select instance type.
- Step 4: Configure Instance Details and select Request Spot Instances, selecting the maximum price to pay.
- Step 5: In the instance details, in advance, we added the script that installs, initializes, and runs the BOINC client automatically in the instance boot time [31]. The content of the script is:

```
#!/bin/bash

### Main variables ###
#S3
S3_BUCKET=""
S3_REGION="us-east-1"

# EC2/instances: Get instance information from metadata
TYPE='curl http://169.254.169.254/latest/meta-data/instance-type '
EC2ID='curl http://169.254.169.254/latest/meta-data/instance-id '
BATCH="TEST"

BOINC_CMD="/usr/bin/boinccmd"
BOINC_PROJECT="http://vorvados.oerc.ox.ac.uk/cpdnboinc_alpha"
BOINC_KEY=""

# Wait seconds (for new tasks)
WAIT_SECONDS="60"

# Function: Setup and Connect BOINC to CPDN project
function setup_boinc {
```

```

cd /var/lib/boinc-client

# Boot script for AWS Ubuntu VM to run CPDN runs through BOINC

# Install required packages for Ubuntu (and 32 bit compatibility)
sudo apt-get update
sudo apt-get -y install awscli lib32stdc++6 lib32z1 boinc

# Print date to see how long this has taken
date
${BOINC_CMD} --project_attach ${PROJECT} ${KEY}

# List Workunits running on this instance
${BOINC_CMD} --get_tasks | grep '^ name' > tasks.txt

# Then, prevent BOINC from getting new work
${BOINC_CMD} --project ${PROJECT} detach_when_done

echo "Polling whether BOINC is still connected"
}

# Function: Check (and run) for new tasks
function check_tasks {
while --get_project_status | grep '1)'; do
# Check spot instance termination
if curl -s \
http://169.254.169.254/latest/meta-data/spot/termination-time \
| grep -q .*T.*Z; then
# Update project in case we have successful tasks
# to report
/usr/bin/boinccmd --project ${PROJECT} update
# Report instance uptime
uptime > timing.txt
aws s3 cp timing.txt \
s3://${S3_BUCKET}/${BATCH}/${TYPE}/terminated_${EC2ID}.txt \
--region=${S3_REGION}
aws s3 cp tasks.txt \
s3://${S3_BUCKET}/${BATCH}/${TYPE}/tasks_${EC2ID}.txt \
--region=${S3_REGION}
sleep 10
/usr/bin/boinccmd --project ${PROJECT} detach
fi
sleep ${WAIT_SECONDS} # Wait polling secs
done
}

# Function: Generate reports and upload to S3
function report {
df -h | grep xvda1 > diskusage.txt
uptime > timing.txt

```

```

aws s3 cp timing.txt \
s3://${S3_BUCKET}/${BATCH}/${TYPE}/complete_${EC2ID}.txt \
--region=${S3_REGION}
aws s3 cp tasks.txt \
s3://${S3_BUCKET}/${BATCH}/${TYPE}/tasks_${EC2ID}.txt \
--region=${S3_REGION}
aws s3 cp diskusage.txt \
s3://${S3_BUCKET}/${BATCH}/${TYPE}/diskusage_${EC2ID}.txt \
--region=${S3_REGION}
}

# Function: Clean up and shut down
function clean_up {
sudo shutdown -h now
}

### MAIN ###
# Workflow: Setup BOINC in instance and wait (and run) for tasks
setup_boinc
check_tasks

# When completed: report (to S3) and cleanup
report
cleanup

```

- Step 6: Add the necessary storage, 64 GB.
- Step 7: Give a name to the instance (for better identification).
- Step 8: Select a security group (in this case, by default, having port 22 open is enough).
- Step 9: Review parameters and Launch.

Please note that shared storage setup is not described here, and S3 buckets (and directories) need to exist before running this script.

Appendix A.2. Microsoft Azure

Azure is the name given to the collection of Microsoft's cloud services, which includes Virtual Machines (VMs, for computing) and shared storage; the former is where the CPDN simulations are run, and the latter will save reports and data from the experiments.

Command-line tooling and Linux integration are not as mature as in AWS (for instance, when our experiments were performed, the shared filesystem was done over SMB; VM cloning was not a simple and atomic operation; or metadata access within VMs was limited), but it is undergoing continuous improvement.

All these steps are done on the Azure Portal (<https://portal.azure.com>), under Virtual Machines (Add):

- Step 1: Select Ubuntu Server (Ubuntu Server 16.04 LTS).
- Step 2: Select Create.
- Step 3: Give a name, user name and password (used for SSH access).
- Step 4: Select VM type/size.
- Step 5: On the VM Settings, Select Extensions, Add Extension and Custom Script for Linux, and upload the script with the content:

```

#!/bin/bash

### Main variables ###
# Storage
AZURE_ACCOUNT=""
FS_KEY_PASSWORD=""
SHARE_NAME=""
MOUNT_POINT=""
MOUNT_PARAMS="-o vers=3.0,username=${AZURE_ACCOUNT}, \
password=${FS_KEY_PASSWORD},dir_mode=0777,file_mode=0777,serverino"

# VM: Get instance information from metadata
VM_ID='curl -H Metadata:true
http://169.254.169.254/metadata/latest/InstanceInfo/ID'
BATCH="TEST"

BOINC_CMD="/usr/bin/boinccmd"
BOINC_PROJECT="http://vorvadoss.oerc.ox.ac.uk/cpdnboinc_alpha"
BOINC_KEY=""

# Wait seconds (for new tasks)
WAIT_SECONDS="60"

# Function: Setup and Connect BOINC to CPDN project
function setup_boinc {
cd /var/lib/boinc-client

# Boot script for Ubuntu VM to run CPDN runs through BOINC
# Install required packages for Ubuntu (and 32 bit compatibility)
# and shared storage
sudo apt-get update
sudo apt-get -y install cifs-utils lib32stdc++6 lib32z1 boinc

# Mount shared FS
sudo mount -t cifs //${AZURE_ACCOUNT}.file.core.windows.net
/${SHARE_NAME} \ ./${MOUNT_POINT} ${MOUNT_PARAMS}

# Print date to see how long this has taken
date
${BOINC_CMD} --project_attach ${PROJECT} ${KEY}

# List Workunits running on this instance
${BOINC_CMD} --get_tasks | grep '^ \ name' > tasks.txt

# Then, prevent BOINC from getting new work
${BOINC_CMD} --project ${PROJECT} detach_when_done

echo "Polling whether BOINC is still connected"
}

```

```

# Function: Check (and run) for new tasks
function check_tasks {
while --get_project_status | grep '1)'; do
# Check spot instance termination
if curl -s \
http://169.254.169.254/latest/meta-data/spot/termination-time \
| grep -q .*T.*Z; then
# Update project in case we have successful
# tasks to report
/usr/bin/boincmd --project ${PROJECT} update
# Report instance uptime
uptime > timing.txt
cp timing.txt \
${MOUNT_POINT}/${BATCH}/terminated_${VM_ID}.txt
cp tasks.txt \
${MOUNT_POINT}/${BATCH}/tasks_${VM_ID}.txt

sleep 10
/usr/bin/boincmd --project ${PROJECT} detach
fi
sleep ${WAIT_SECONDS} # Wait polling secs
done
}

# Function: Generate reports and upload to Shared FS
function report {
df -h | grep xvda1 > diskusage.txt
uptime > timing.txt
cp timing.txt ${MOUNT_POINT}/${BATCH}/complete_${VM_ID}.txt
cp tasks.txt ${MOUNT_POINT}/${BATCH}/tasks_${VM_ID}.txt
cp diskusage.txt ${MOUNT_POINT}/${BATCH}/diskusage_${VM_ID}.txt
}

# Function: Clean up and shut down
function clean_up {
sudo shutdown -h now
}

### MAIN ###
# Workflow: Setup BOINC on instance and wait (and run) for tasks
setup_boinc
check_tasks

# When completed: report (to Shared FS) and cleanup
report
cleanup

```

- Step 6: Add storage, 64 GB.
- Step 7: Start VM.

Please note that Azure shared storage setup is not described here, and endpoints (and directories) need to exist before running this script.

Appendix B. Infrastructure Details for WACCM Experiments

For this part, simulations within the Finisterrae II super computer and on Google's cloud platform are evaluated.

Appendix B.1. Finisterrae II super computer

The Finisterrae II super computer installed at CESGA (Centro de Supercomputacion de Galicia (The Supercomputing Center of Galicia)) is a system integrated by shared memory nodes with an SMPNUMA architecture. It is composed by:

- 143 computing nodes.
- 142 HP Integrity rx7640 nodes with 16 Itanium Montvale cores with 128 GB of RAM each.
- An Infiniband 4 × DDR 20 Gbps interconnection network.

The model was already available on FinisTerra II because it was being used by some research groups.

Appendix B.2. Google Compute Engine

On the other hand, within Google's cloud platform, because considerable computing resources were needed, the biggest instance available at the time was chosen, the n1-highcpu-16 instance type. This instance type has 16 virtual CPUs, 14.40 GB of RAM memory, and an estimated computing power of 44 GCEUs (a virtual CPU is implemented as a single hardware hyperthread on a 2.6 GHz Intel Xeon E5 (Sandy Bridge), 2.5 GHz Intel Xeon E5 v2 (Ivy Bridge), or 2.3 GHz Intel Xeon E5 v3 (Haswell)).

The chosen data centre location was the USA because it was less expensive, and for our purposes, it made no difference as long as all instances were within the same region. We needed eight instances in order to run simulations with 128 CPUs.

Appendix B.3. Cluster Creation

Cluster setup was achieved using a master node that contained the model and where a customized simulation, also known as a case, was created. To handle all the output data, an additional hard disk had to be attached. This master node ran an NFS server and could connect through SSH without a password to all the instances of the cluster. From this node, all simulations were executed. In order to automate cluster provisioning, configuration and simulations execution, a tool was developed (source code at the end of this Appendix).

Appendix B.4. Simulations

The configuration of the simulations was as follows:

- All components active: atmosphere, ocean, land, sea-ice and land-ice.
- Resolution of the grid of 1.9×2.5_1.9×2.5 (the approximately two-degree finite volume grid).
- MPI tasks of 1, 8, 16, 32, 64 and 128.
- Simulation length of one and ten years.

The model itself had a benchmarking tool that provided different metrics associated with each simulation run such as: model cost, throughput and run time.

```
#!/bin/bash
```

```
NUMNODES=8
```

```
INSTANCETYPE=n1-highcpu-16
```

```

REGION=us-central1-a
#1. Verifies that Google's utilities are installed. If not, the program
exits. command -v gcutil >/dev/null 2>&1 || { chho >&2 ''gcutil needs \
to be installed but it couldn't be found. Aborting.''; exit 1;}

#2. Sets the project name.
projectID='gcloud config list | grep project | awk '{ print $3}^

#3. Sets the number of nodes.
numNodes=${NUMNODES}

#4. Sets machine type and image.
machTYPE=${INSTANCETYPE}
imageID=https://www.googleapis.com/compute/v1/projects/debian-\
cloud/global/images/debian-7-wheezy-v20140807

#5. Adds nodes to the cluster and wait until they are running.
nodes=$(eval echo machine{0..$((($numNodes-1))})
gcutil addinstance --image=$imageID --machine_type=$machTYPE\
--zone=${REGION} --wait_until_running $nodes

#6. Uploads the file install.sh to the slave nodes.
for i in $(seq 1 $((($numNodes-1))); do
gcutil push machine$i install.sh .
done

#7. Executes previous script in each node and checks if
# the configuration ended successfully in every machine.

for i in $(seq 1 $((($numNodes-1))); do
gcutil ssh machine$i "/bin/bash ./install.sh machine$i >&\
install.log.machine$i" &
done

for i in $(seq 1 $((($numNodes-1))); do
gcutil ssh machine$i "grep DONE install.log.machine$i"
done

#8. Finally, configures ssh keys to allow the connection from
#the master node without password.

clave_pub='gcutil ssh machine0 ''sudo cat ~/.ss/id_rsa.pub''''
for i in $(seq 1$((($numNodes-1))); do
echo ''$clave_pub'' | gcutil ssh machine$i ''cat >> \
~/.ssh/authorized_keys''
done

cat << EOF > config
Host *
StrictHostKeyChecking no

```

```
UserKnownHostsFile=/dev/null
EOF
cat config | gcutil ssh machine0 "cat >> ~/.ssh/config"
rm config
```

References

- Palmer, T. Build high-resolution global climate models. *Nature* **2014**, *515*, 338–339. [CrossRef]
- Bell, G.; Hey, T.; Szalay, A. Beyond the data deluge. *Science* **2009**, *323*, 1297–1298. [CrossRef]
- EIU. Ascending Cloud: The Adoption of Cloud Computing in Five Industries; Technical Report. 2016. Available online: https://www.vmware.com/radius/wp-content/uploads/2015/08/EIU_VMware-Executive-Summary-FINAL-LINKS-2-26-16.pdf (accessed on 21 June 2020).
- Meinardi, M.; Smith, D.; Plummer, D.; Cearley, D.; Natis, Y.; Khnaser, E.; Nag, S.; MacLellan, S.; Petri, G. *Predicts 2020: Better Management of Cloud Costs, Skills and Provider Dependence Will Enable Further Cloud Proliferation*; Technical Report; ITRS Group Ltd.: London, UK, 2019.
- Zhao, Y.; Li, Y.; Raicu, I.; Lu, S.; Tian, W.; Liu, H. Enabling scalable scientific workflow management in the Cloud. *Future Gener. Comp. Syst.* **2015**, *46*, 3–16. [CrossRef]
- Añel, J.A.; Montes, D.P.; Rodeiro Iglesias, J. *Cloud and Serverless Computing for Scientists*; Springer: Berlin, Germany, 2020; p. 110. [CrossRef]
- Ayris, P.; Berthou, J.Y.; Bruce, R.; Lindstaedt, S.; Monreale, A.; Mons, B.; Murayama, Y.; Södergård, C.; Tochtermann, K.; Wilkinson, R. *Realising the European Open Science Cloud*; Technical Report; Publications Office of the European Union: Luxemburg, 2016. [CrossRef]
- White, T. *Hadoop: The Definitive Guide*, 4th ed.; O'Reilly Media: Sebastopol, CA, USA, 2015; p. 728.
- Lawrence, B.; Bennett, V.; Churchill, J.; Jukes, M.; Kershaw, P.; Pascoe, S.; Pepler, S.; Pritchard, M.; Stephens, A. Storing and manipulating environmental big data with JASMIN. *Proc. IEEE Big Data* **2013**, 68–75. [CrossRef]
- NOAA. Big Data Project. Available online: <https://www.noaa.gov/organization/information-technology/big-data-program> (accessed on 21 June 2020).
- AWS. The Met Office Case Study. Available online: <https://aws.amazon.com/solutions/case-studies/the-met-office/> (accessed on 21 June 2020).
- Vance, T.; Merati, N.; Yang, C.; Yuan, M. (Eds.) *Cloud Computing in Ocean and Atmospheric Sciences*, 1st ed.; Academic Press: Cambridge, MA, USA, 2016; p. 415.
- Evangelinos, C.; Hill, C.N. Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. In Proceedings of the First Workshop on Cloud Computing and its Applications (CCA'08), Chicago, IL, USA, 22–23 October 2008.
- Molthan, A.L.; Case, J.L.; Venner, J.; Schroeder, R.; Checchi, M.R.; Zavodsky, B.T.; Limaye, A.; O'Brien, R.G. Clouds in the cloud: weather forecasts and applications within cloud computing environments. *Bull. Am. Meteorol. Soc.* **2015**, *96*, 1369–1379. [CrossRef]
- McKenna, B. Dubai Operational Forecasting System in Amazon Cloud. *Cloud Comput. Ocean Atmos. Sci.* **2016**, 325–345. [CrossRef]
- Blanco, C.; Cofino, A.S.; Fernández, V.; Fernández, J. Evaluation of Cloud, Grid and HPC resources for big volume and variety of RCM simulations. *Geophys. Res. Abstracts* **2016**, *18*, 17019.
- Montes, D.; Añel, J.A.; Pena, T.F.; Uhe, P.; Wallom, D.C.H. Enabling BOINC in Infrastructure as a Service Cloud Systems. *Geosci. Mod. Dev.* **2017**, *10*, 811–826. [CrossRef]
- Chen, X.; Huang, X.; Jiao, C.; Flanner, M.G.; Raeker, T.; Palen, B. Running climate model on a commercial cloud computing environment. *Comput. Geosci.* **2017**, *98*, 21–25. [CrossRef]
- Zhuang, J.; Jacob, D.J.; Gaya, J.F.; Yantosca, R.M.; Lundgren, E.W.; Sulprizio, M.P.; Eastham, S.D. Enabling Immediate Access to Earth Science Models through Cloud Computing: Application to the GEOS-Chem Model. *Bull. Am. Meteorol. Soc.* **2019**, *100*, 1943–1960. [CrossRef]
- Zhuang, J.; Jacob, D.J.; Lin, H.; Lundgren, E.W.; Yantosca, R.M.; Gaya, J.F.; Sulprizio, M.P.; Eastham, S.D. Enabling High-Performance Cloud Computing for Earth Science Modeling on Over a Thousand Cores: Application to the GEOS-Chem Atmospheric Chemistry Model. *J. Adv. Model. Earth Syst.* **2020**, *12*, e2020MS002064. [CrossRef]

21. Goodess, C.M.; Troccoli, A.; Acton, C.; Añel, J.A.; Bett, P.E.; Brayshaw, D.J.; De Felice, M.; Dorling, S.E.; Dubus, L.; Penny, L.; et al. Advancing climate services for the European renewable energy sector through capacity building and user engagement. *Clim. Serv.* **2019**, *16*, 100139. [[CrossRef](#)]
22. Düben, P.D.; Dawson, A. An approach to secure weather and climate models against hardware faults. *J. Adv. Model. Earth Syst.* **2017**, *9*, 501–513. [[CrossRef](#)]
23. Wright, D.; Smith, D.; Bala, R.; Gill, B. *Magic Quadrant for Cloud Infrastructure as a Service, Worldwide*; Technical Report; Gartner, Inc.: Stamford, CT, USA, 2019.
24. RightScale. *RightScale 2018 State of the Cloud Report*; Technical Report; RightScale, Inc.: Santa Barbara, CA, USA, 2018.
25. Hille, M.; Klemm, D.; Lemmermann, L. *Crisp Vendor Universe/2017: Cloud Computing Vendor & Service Provider Comparison*; Technical Report; Crisp Research GmbH: Kassel, Germany, 2018.
26. Allen, M.R. Do-it-yourself climate prediction. *Nature* **1999**, *401*, 642. [[CrossRef](#)]
27. Guillod, B.P.; Jones, R.G.; Bowery, A.; Haustein, K.; Massey, N.R.; Mitchell, D.M.; Otto, F.E.L.; Sparrow, S.N.; Uhe, P.; Wallom, D.C.H.; et al. weather@home 2: validation of an improved global–regional climate modelling system. *Geosci. Mod. Dev.* **2017**, *10*, 1849. [[CrossRef](#)]
28. Anderson, D.P. BOINC: A System for Public-Resource Computing and Storage. In Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing, Pittsburgh, PA, USA, 8 November 2004; pp. 4–10.
29. Massey, N.; Jones, R.; Otto, F.E.L.; Aina, T.; Wilson, S.; Murphy, J.M.; Hassel, D.; Yamazaki, Y.H.; Allen, M.R. weather@home—Development and validation of a very large ensemble modelling system for probabilistic event attribution. *Quart. J. R. Meteor. Soc.* **2014**, *141*, 1528–1545. [[CrossRef](#)]
30. Lange, S. On the Evaluation of Regional Climate Model Simulations over South America. Ph.D. Thesis, Humboldt-Universität zu Berlin, Berlin, Germany, 2015.
31. Uhe, P.; Otto, F.E.L.; Rashid, M.M.; Wallom, D.C.H. Utilising Amazon Web Services to provide an on demand urgent computing facility for climateprediction.net. In Proceedings of the 2016 IEEE 12th International Conference on e-Science (e-Science), Baltimore, MD, USA, 23–27 October 2016; pp. 407–413. [[CrossRef](#)]
32. Marsh, D.R.; Mills, M.; Kinnison, D.; Lamarque, J.F.; Calvo, N.; Polvani, L.M. Climate change from 1850 to 2005 simulated in CESM1(WACCM). *J. Clim.* **2013**, *26*. [[CrossRef](#)]
33. CCMVal. *SPARC CCMVal Report on the Evaluation of Chemistry–Climate Models*; SPARC Report No. 5, WCRP-132, WMO/TD-No. 1526; SPARC Office: Toronto, ON, Canada, 2010.
34. Gettelman, A.; Hegglin, M.I.; Son, S.W.; Kim, J.; Fujiwara, M.; Birner, T.; Kremser, S.; Rex, M.; Añel, J.A.; Akiyoshi, H.; et al. Multi-model Assessment of the Upper Troposphere and Lower Stratosphere: Tropics and Trends. *J. Geophys. Res.* **2010**, *115*, D00M08. [[CrossRef](#)]
35. Hegglin, M.I.; Gettelman, A.; Hoor, P.; Krichevsky, R.; Manney, G.L.; Pan, L.L.; Son, S.W.; Stiller, G.; Tilmes, S.; Walker, K.A.; et al. Multi-model Assessment of the Upper Troposphere and Lower Stratosphere: Extra-tropics. *J. Geophys. Res.* **2010**, *115*, D00M09. [[CrossRef](#)]
36. Toohey, M.; Hegglin, M.I.; Tegtmeier, S.; Anderson, J.; Añel, J.A.; Bourassa, A.; Brohede, S.; Degenstein, D.; Froidevaux, L.; Fuller, R.; et al. Characterizing sampling biases in the trace gas climatologies of the SPARC Data Initiative. *J. Geophys. Res. Atmos.* **2013**, *118*, 11847–11862. [[CrossRef](#)]
37. Chiodo, G.; García-Herrera, R.; Calvo, N.; Vaquero, J.M.; Añel, J.A.; Barriopedro, D.; Matthes, K. The impact of a future solar minimum on climate change projections in the Northern Hemisphere. *Environ. Res. Lett.* **2016**, *11*, 034015. [[CrossRef](#)]
38. *The SPARC Data Initiative: Assessment of Stratospheric Trace Gas and Aerosol Climatologies from Satellite Limb Sounders*; Technical Report; ETH-Zürich: Zürich, Switzerland, 2017. [[CrossRef](#)]
39. Añel, J.A.; Gimeno, L.; de la Torre, L.; García, R.R. Climate modelling and Supercomputing: WACCM at CESGA. *Dixitos Comput. Sci.* **2008**, *2*, 31–32.
40. Wilson, S. (MetOffice Hadley Centre, Exeter, UK). Personal communication, 2018.
41. Ranjan, R.; Benatallah, B.; Dustdar, S.; Papazoglou, M.P. Cloud Resource Orchestration Programming: Overview, Issues and Directions. *IEEE Internet Comput.* **2015**, *19*, 46–56. [[CrossRef](#)]
42. Rath, A.; Spasic, B.; Boucart, N.; Thiran, P. Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure. *Computers* **2019**, *8*, 34. [[CrossRef](#)]
43. Mitchell, N.J.; Zunnurhain, K. *Google Cloud Platform Security*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 319–322. [[CrossRef](#)]

44. Kaufman, C.; Venkatapathy, R. *Windows Azure TM Security Overview*; Technical Report; Microsoft: Redmond, WA, USA, 2010.
45. Saeed, I.; Baras, S.; Hajjiab, H. Security and Privacy of AWS S3 and Azure Blob Storage Services. In Proceedings of the IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019. [CrossRef]
46. Craig Mudge, J. *Cloud Computing: Opportunities and Challenges for Australia*; Technical Report; ASTE: Melbourne, Australia, 2010.
47. Charney, J.G.; Fjörtoft, R.; von Neumann, J. Numerical Integration of the Barotropic Vorticity Equation. *Tellus* **1950**, *2*, 237–254. [CrossRef]
48. Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; et al. The ERA5 Global Reanalysis. *Q. J. R. Meteorol. Soc.* **2020**. [CrossRef]
49. Kay, J.E.; Deser, C.; Phillips, A.; Mai, A.; Hannay, C.; Strand, G.; Arblaster, J.M.; Bates, S.C.; Danabasoglu, G.; Edwards, J.; et al. The Community Earth System Model (CESM) Large Ensemble Project: A Community Resource for Studying Climate Change in the Presence of Internal Climate Variability. *Bull. Am. Meteorol. Soc.* **2015**, *96*, 1333–1349. [CrossRef]
50. Drake, N. Cloud computing beckons scientists. *Nature* **2014**, *509*, 543–544. [CrossRef]
51. Stein, L.; Knoppers, B.M.; Campbell, P.; Getz, G.; Korbel, J.O. Create a cloud commons. *Nature* **2015**, *523*, 149–151. [CrossRef]
52. Critical Techniques, Technologies and Methodologies for Advancing Foundations and Applications of Big Data Sciences and Engineering (BIGDATA NSF-18-539); Technical Report. 2018. Available online: <https://www.nsf.gov/pubs/2018/nsf18539/nsf18539.htm> (accessed on 21 June 2020).
53. EarthCube: Developing a Community-Driven Data and Knowledge Environment for the Geosciences (BIGDATA NSF 20-520); Technical Report. 2020. Available online: <https://www.nsf.gov/pubs/2020/nsf20520/nsf20520.htm> (accessed on 21 June 2020).
54. Raoult, B.; Correa, R. Cloud Computing for the Distribution of Numerical Weather Prediction Outputs. *Cloud Comput. Ocean Atmos. Sci.* **2016**, 121–135. [CrossRef]
55. Misra, S.; Mondal, A. Identification of a company’s suitability for the adoption of cloud computing and modelling its corresponding Return on Investment. *Mat. Comput. Model.* **2011**, *53*, 504–521. [CrossRef]
56. Bidosola, I.; Río-Belver, R.; Cilleruelo, E.; Garechana, G. Design and Implementation of a Cloud Computing Adoption Decision Tool: Generating a Cloud Road. *PLoS ONE* **2015**, *10*, e0134563. [CrossRef]
57. CLOUDYN. *Determining Your Optimal Mix of Clouds*; Technical Report; Cloudyn: Rosh Ha’ayin, Israel, 2015.
58. Oriol Fitó, J.; Macías, M.; Guitart, J. Toward Business-driven Risk Management for Cloud Computing. In Proceedings of the International Conference on Network and Service Management, Niagara Falls, ON, Canada, 25–29 October 2010; pp. 238–241.
59. Wallom, D.C.H. *Report from the Cloud Security Workshop: Building Trust in Cloud Services Certification and Beyond*; Technical Report; European Commission: Brussels, Belgium, 2016.
60. Kim, A.; McDermott, J.; Kang, M. Security and Architectural Issues for National Security Cloud Computing. In Proceedings of the IEEE 30th International Conference on Distributed Computing Systems Workshops, Genova, Italy, 21–25 June 2010; pp. 21–25. [CrossRef]
61. Bennett, K.W.; Robertson, J. Security in the Cloud: Understanding your responsibility. In Proceedings of the SPIE 11011, Cyber Sensing 2019, Baltimore, MD, USA, 17 May 2019; pp. 1–18. [CrossRef]
62. Añel, J.A. The importance of reviewing the code. *Commun. ACM* **2011**, *54*, 40–41. [CrossRef]
63. Hutton, C.; Wagener, T.; Freer, J.; Han, D.; Duffy, C.; Arheimer, B. Most computational hydrology is not reproducible, so is it really science? *Water Resour. Res.* **2016**, *52*, 7548–7555. [CrossRef]
64. Añel, J.A. Comment on ‘Most computational hydrology is not reproducible, so is it really science?’ by Hutton et al. *Water Resour. Res.* **2017**, *53*, 2572–2574. [CrossRef]
65. Perspectives on Cloud Outcomes: Expectation vs. Reality. 2020. p. 15. Available online: https://www.accenture.com/_acnmedia/pdf-103/accenture-cloud-well-underway.pdf (accessed on 18 June 2020).

