Reducing the Carbon Footprint of Ensemble Weather Forecasting with GPUs

Jeff Adie
NVIDIA AI Technology Centre
Singapore, Singapore
jadie@nvidia.com

Terry Yin
NVIDIA AI Technology Centre
Singapore, Singapore
jianxiongy@nvidia.com

Stan Posey
NVIDIA Corporation
Santa Clara, USA
sposey@nvidia.com

Simon See
NVIDIA AI Technology Centre
Singapore, Singapore
ssee@nvidia.com

***Abstract*— Climate and Weather Modelling is a highly complex and computationally intensive task which consumes substantial amounts of energy. A desire to improve forecast skill demands further advances in these forecasts, such as increased model fidelity and more comprehensive physical representations of the underlying processes. Another driver towards better forecasts is the goal of uncertainty quantification, with ensembles of forecasts a popular technique. But ensembles place much higher demands on the computational workload as N ensemble members require N times the compute cycles. This leads to an even higher energy demand.**

This study examines one approach to reducing the energy demands of ensembles by taking advantage of a hardware feature in modern NVIDIA GPUs known as Multi Instance GPU (MIG). This feature allows us to run multiple ensemble members on hardware-isolated GPU slices to maximize efficient use of the GPU resources and subsequently reduce the Carbon Footprint for an Ensemble forecast. We examine both small and large test cases across a range of setups to determine the optimal runtime configuration. Our study shows a 2.5-2.8x reduction in CO2 emissions across all cases which translates into a savings of between 141-171 tonnes of carbon emissions annually per GPU.

Keywords*— *Numerical Weather Prediction, Energy Efficiency, GPU MIG, Ensemble Forecasting

I. Introduction

Climate change is arguably the most important event occurring on the planet today and understanding its impact is increasingly vital to our society. We need more accurate and more timely forecasts for both short-term extreme weather event forecasting as well as longer term scenario modelling.

Modern climate research places extreme demands on computational resources. These demands are driven by many factors that relate to improving the accuracy of the climate models and the subsequent scenario results. One such example is the modelling of clouds. The greatest amount of uncertainty in climate models usually comes from the prediction of incoming (downward) solar radiation which affects the energy budget of the planet. The amount of this energy is directly related to the cloud reflectance, and therefore it is driven by the accuracy of the cloud model. Accurate models of clouds need to be run at a very high resolution that is generally less than 3km – these are so-called cloud resolving models (CRM). But running global climate models for many years at such a high resolution is beyond todays' capabilities. A team at MeteoSwiss and ETH Zurich [4] showed that a 36-year simulation of a 1km climate model would take 840 days to complete a single run, consuming some 22GWh of power in the process. It is clear that we are on an ever-increasing energy utilization path that is unsustainable in the future. In 2018, computers in data centers consumed 1% of all power worldwide with that figure estimated to grow anywhere from 3-13% by 2030 [1].

Improving the accuracy of individual forecast runs by itself is insufficient. Weather is an inherently chaotic phenomena with highly non-linear behavior and substantial sensitivity to initial conditions. This leads to uncertainty in the prediction of the model. One method to improve our confidence in the forecasts is to use a statistical technique known as ensemble forecasting in which multiple forecasts are executed with minor variations in the model parameters and/or perturbations of the initial state of the atmosphere. The various runs, known as ensemble members, can be analysed to produce a mean and a variance which leads to both an improved forecast as well as uncertainty quantification [3]. These improvements in the predictions come at a substantial cost in additional computational resources, power, and carbon emissions. The goal of this study is to look at how new features in GPUs can be exploited to ameliorate these costs.

Prior work in studying GPU energy efficiency, such as [2], have mostly focused on energy efficiency at the kernel level and how that applies to overall GPU utilization vs power. Our study considers application performance efficiencies by taking advantage of the ability to perform fractional GPU matching to application requirements through the MIG feature.

This paper is organized as follows: In section 2, we introduce GPU accelerated computing and MIG. In section 3, we discuss the experiment setup and methodology. In section 4, we present the results of our experiments. Finally, we summarize the conclusions from our results and discuss future steps.

II. GPUs and MIG

In the past decade, we've witnessed the fast development of high-performance computing (HPC) technologies, e.g., heterogenous computing, and applications, e.g., Large Scale Simulation, Artificial Intelligence model training. NVIDIA Graphics Computing Unit (GPU) is one of the most widely adopted accelerators for heterogenous HPC computing because of its performance leadership and the dominance position of CUDA based software eco-system that has deeply rooted in HPC and AI development lifecycle. In HPC domain, although continuously scaling out is always the main theme,

which requires GPU to be as powerful as possible, fine grained GPU resource allocation and scheduling have become more and more critical because of the rapidly emerging nano to medium scale HPC workloads and Service Layer Objectives (SLO) requirements from computing service QoS perspective. In response to such new technical challenges, as part of NVIDIA A100 GPU's seven key features, NVIDIA announced Multi-Instance GPU (MIG) technology which basically allows users to physically provision one high end GPU into multiple independent smaller GPU instances. Compared with the GPU sharing technologies that were available, MIG has multiple desirable advantages including, but not limited to, flexible provision, Isolated GPU failure, Isolated workload Interference, Low power overhead. Such advantages will not be possible without MIG's capability of partitioning physical GPU into GPU instance(s) with dedicated stream multi-processor clusters, GPU memory, L2 cache as well as the IO bandwidth, which minimized many technical issues that caused by time sharing GPU resource. MIG is currently also supported by NVIDIA A30 and H100 GPUs

III. Methodology and Setup

The goal of this study is to examine the best method to run an ensemble of forecasts on a given hardware solution that minimizes the energy footprint of that run. To that end, we devised and undertook a series of experiments to determine the energy profiles of representative forecast models covering both large and small model sizes.

A. Capturing energy usage

In capturing the energy consumption or our experiments, we followed the methodology of [5] which defines three quality levels of measurement. For this experiment we used Level 1, which is based on version 0.9 of the Green500 run rules[1]. For level 1, power needs to be sampled at one-second intervals over at least 20% of the run – in fact we capture the full run. Power measured at level 1 is also limited to the computing elements only and does not include other components such as I/O devices or network components. We believe this is an appropriate level to base this study on as we are interested in the difference in computing element consumption, rather than the total power consumption.

To measure the CPU power, we used the lm-sensors package on Ubuntu which can return, using the sensors command, the instantaneous CPU Power at any interval.

To measure the GPU power we used the nvidia-smi command which has a built-in loop option for regular measurements.

To determine the carbon footprint, we use published figures for both the USA and Singapore that determine the rate of CO_2 emitted per kWh used. The experiments were run in Singapore – we include the USA energy values as Singapore is somewhat skewed due to the country's high dependence on fossil fuels. These values are summarized in Table I.

B. Workload Cases

As the MIG capability allows us to slice the GPU into 1-7 slices, we wished to test both a small case that could fit into the smallest slice as well as a case that required larger slices.

TABLE I. POWER GRID CARBON FOOTPRINT VALUES

Country	*Year*	*kg CO_2/kWh*	*Source*
USA	2021	0.385	www.eia.gov
Singapore	2020	0.408	www.ema.gov.sg

To meet this requirement, we used two cases names Ksmall, and IP4, depicted in Fig. 1.

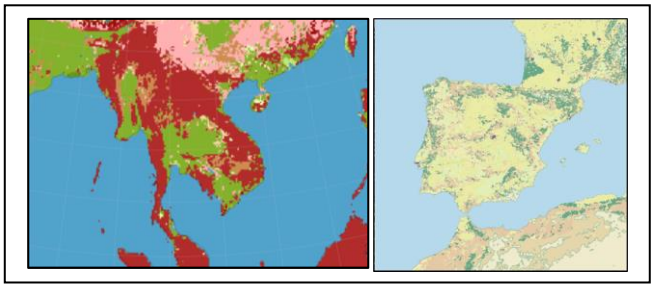

Fig. 1. Ksmall (left) and IP4 (right) case model domains

Ksmall is a 24 hour, 18km resolution forecast over South East Asia centered on the Khorat plateau in central Thailand. The case is derived from a nested 18km-6km-2km forecast of the same region called ModelK. The model uses a tropical physics suite and consists of about 1.2 million grid cells.

IP4 is a 24 hour, 4km resolution forecast over the Iberian Peninsula centered on Spain. The model uses a temperate physics suite and consists of about 6 million cells.

The two models and their parameters are summarized in Table II.

C. Experiment Setup

1) Experimental Hardware

Our experiments were run on the NVIDIA ARM HPC Devkit system. The NVIDIA Arm HPC Developer Kit is an integrated hardware and software platform for creating, evaluating, and benchmarking HPC, AI, and scientific computing applications on a heterogeneous GPU- and CPU-accelerated computing system. The kit includes a single Ampere Altra Q80-30 Arm CPU, 512GB DDR4 RAM, 6TB SAS/SATA HDD, an NVIDIA A100 Tensor Core GPU server, and the.

2) Base Software Layers

The software stack was comprised of several layers with the base OS layer of Ubuntu 20.04.3 LTS (Focal Fossa) with the NVIDIA GPU driver version 510.47 and CUDA version 11.5. On top of this was the NVIDIA HPC SDK suite of tools versions 22.1 providing the Compilers and accelerated HPC libraries. Finally, the HDF5 version 1.12.0, NetCDF-C version 4.7.4 and NetCDF-Fortran version 4.5.3 packages provided the I/O libraries.

3) Application Software

For the application software (WRF), CPU runs of the workload were performed using WRF Version 3.8.1. from NCAR [6] whereas GPU runs of the workload were performed

[1] http://green500.org/docs/pubs/RunRules_Ver0.9.pdf

TABLE II. MODEL COMPARISONS

Model	*Ksmall*	*IP4*
Resolution	18km	4km
Meteorology	Tropical	Temperate
Horizontal Grid Size	230x170	400x400
Vertical Levels	31	38
Total number of cells	1.2 million	6 million

using AceCAST Version 1.2 from TQI [7]. AceCAST V1.2 is a GPU accelerated version of NCAR WRF V3.8.1.

D. Experimental Runs

Baseline measurements were first gathered for the CPU and GPU computing elements without any workload. These measurements were performed three times and averaged to derive device P_{IDLE} values. These values are considered 'fixed' costs for the subsequent calculations with the same justification as was used for the non-computing elements. We measured CPU P_{IDLE} at 9.84 Watts and GPU P_{IDLE} at 63.61 Watts

We then ran a standard CPU only workload for both forecast cases. These are our gold runs that all subsequent runs are compared to. We also ran standard (without MIG enabled) GPU workloads. These runs were used to compare the cost of enabling MIG for the same workload.

Once gold runs were complete, we enabled MIG mode on the GPU. We configured the MIG in various different geometries and ran comparison studies. We restricted our slice geometries to symmetric combinations only. Table III shows the geometries that were tested on the GPU.

The 4g.40gb slice geometry was not used as it has no benefit over the 7g.80gb slice (because we are only testing symmetric slices). In theory we could have used a 4g.40GB plus other smaller slices but that would give the ensemble members uneven resources leading to uneven completion times. For the smaller case we ran up to the number of instances shown in the table (i.e. 7 instances on the 1g.10gb geometry). The large case required more resources by design – it specifically required a minimum 20GB GPU memory footprint to run. As such, we could only run up to three ensembles members for the large case.

For each run, we gathered and calculated various metrics. The definition and derivation for each of the metrics are enumerated in Table IV.

TABLE III. MIG SLICE GEOMETRIES

Geometry	*Instances*	*Ksmall*	*IP4*
MIG Disabled	1	YES	YES
1g.10gb	7	YES	NO
2g.20gb	3	YES	YES
3g.40gb	2	YES	YES
4g.40gb	1	NO	NO
7g.80gb	1	YES	YES

TABLE IV. METRICS COLLECTED AND THEIR DERIVATION

Metric	*Description*	*Units*	*Derivation*
T_{WALL}	Total Wallclock Runtime	s	Measured
T_{IOR}	Time in I/O Reads	s	App reported
T_{IOW}	Time in I/O Writes	s	App reported
T_{IOT}	Total I/O Time	s	$T_{IOR} + T_{IOW}$
T_{COMP}	Total Compute Time	s	$T_{WALL} - T_{IOT}$
N_{SAMP}	Number of Samples		Counted
P_{INST}	Instantaneous Power	W	Measured
P_{INJ}	Instantaneous Job Power	W	$P_{INST} - P_{IDLE}$
P_{TOT}	Total Job Power	W	$\sum_{i=0}^{N_{SAMP}} P_{INJ}$
P_{AVG}	Average Power	W	P_{TOT}/N_{SAMP}
P_{EFF}	Power Efficiency	J/s	$P_{TOT}*1000/T_{WALL}$
CONS	Power Consumption	kWh	$(P_{EFF}*T_{WALL})/3.6e6$

IV. EXPERIMENT RESULTS

A. Baselines

The baseline measurements on CPU and GPU (both in standard mode and MIG mode) are summarized below for key metrics in Table V and Table VI respectively.

TABLE V. BASELINE CPU RESULTS

Case	*Ksmall*		*IP4*	
Members	1	7	1	3
T_{WALL}	185	1293	667	2000
P_{TOT}	12194	85358	49270	147809
P_{AVG}	66.3	66.3	73.9	73.9
P_{CONS}	3.387	23.711	13.686	41.058

TABLE VI. BASELINE GPU RESULTS (SINGLE INSTANCE)

Case	*Ksmall*		*IP4*	
MIG	Disabled	Enabled	Disabled	Enabled
T_{WALL}	116	110	175	180
P_{TOT}	6337	4376	16221	17386
P_{AVG}	55.1	40.1	92.7	96.6
P_{CONS}	1.760	1.215	4.506	4.829

B. Full MIG GPU Runs

We next summarise the results for MIG mode for both a single ensemble member and the maximum number of ensemble members (7 for the Ksmall model and 3 for the IP4 model). These results are included in Table VII.

TABLE VII. GPU MIG RESULTS

Case	*Ksmall*		*IP4*	
Members	1	7	1	3
T_{WALL}	185	1203	180	977
P_{TOT}	4376	34098	17386	54333
P_{AVG}	40.1	28.4	96.6	55.6
P_{CONS}	1.215	9.471	4.829	15.092

C. Cost of MIG (MIG slice size vs Compute)

The A100 GPU has 112 streaming multiprocessors (SM) contained within 8 GPCs (each GPC has 14 SMs). When running in MIG mode, only 7 of the GPCs are available, which sets the upper bound on the number of MIG slices per GPU as each slice needs at least 1 GPC. The total number of SMs available for each slice configuration is shown in Table VIII.

TABLE VIII. AVAILABLE SM PER MIG SLICE

Slices	0	1	2	3	7
No. SM	112	98	42	28	14

Clearly having a smaller number of cores available has an impact on the performance of the model. We examine that impact of using these smaller slices on our two cases. We plot the inverse of the average integration time for various slices in Fig. 2 for Ksmall and Fig. 3 for IP4 respectively. In the left column the full GPU, and all 112 SMs are available for the code to use. In the worst case the Ksmall model with a 7 slice geometry configuration has only 14 SMs to execute on and compute performance is 3x slower. For the large IP4 model the impact is greater; at 3 slices (28 SMs) it is 2.36x slower. This cost is of course more than made up for by the fact that the remainder of the GPU is available for other ensemble members to run on.

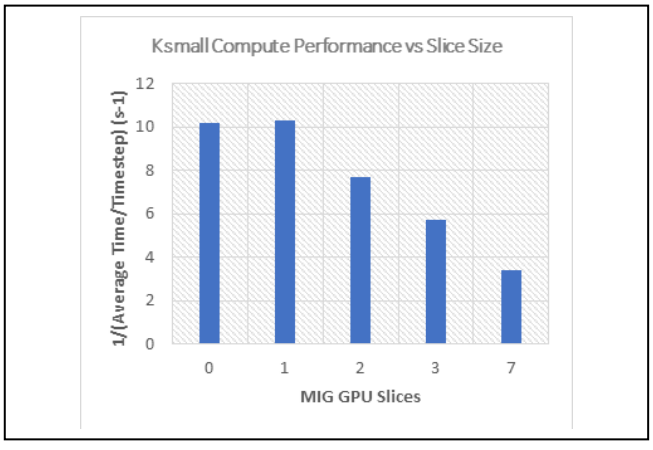

Fig. 2. Ksmall performance with different slices

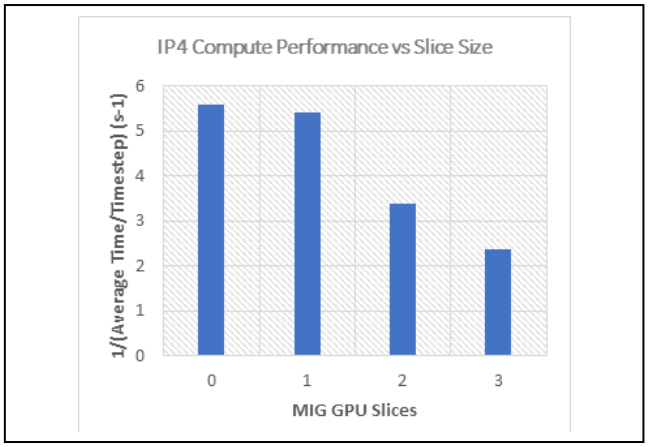

Fig. 3. IP4 performance with different slices

D. Power consumption with MIG

When the GPU is sliced, total GPU power increases as more and more slices are utilized. This is shown below for Ksmall in Fig. 4 and IP4 in Fig. 5 respectively

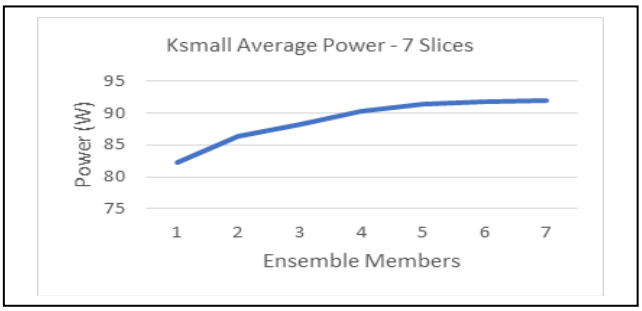

Fig. 4. Ksmall average power per member

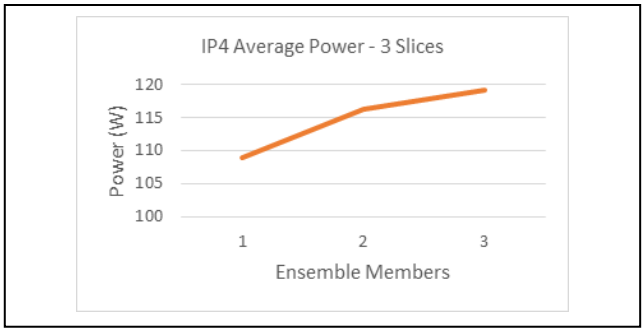

Fig. 5. IP4 average power per member

Plots of power for each member are shown in Fig. 6 (Ksmall) and Fig. 7 (IP4) below

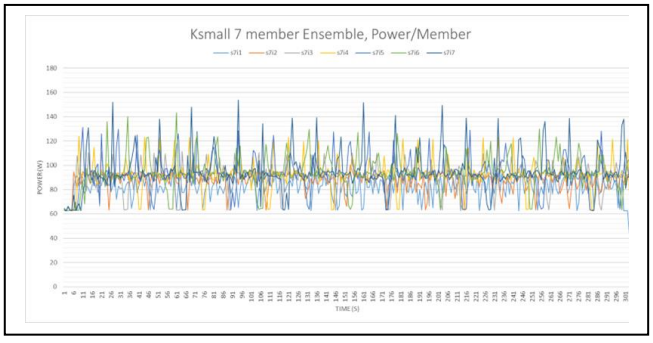

Fig. 6. Ksmall power per member

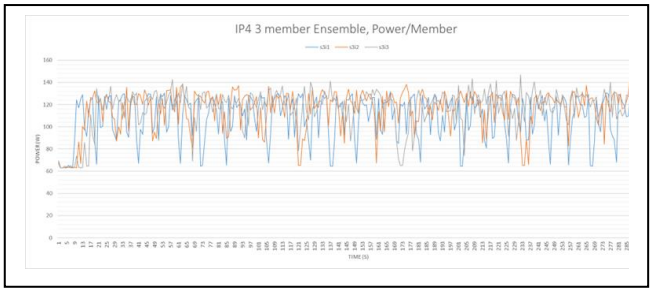

Fig. 7. IP4 power per member

E. *Total power consumption comparisons*

The total power consumption for both CPU only and GPU MIG-enabled runs is summarized in table IX. The table also includes the total power saved through the GPU for each of the runs in kilowatt-hours.

TABLE IX. TOTAL POWER CONSUMPTION

Case	Ksmall		IP4	
Members	1	7	1	3
CPU P_{CONS}	3.387	23.711	13.686	41.058
GPU P_{CONS}	1.215	9.471	4.829	15.092
Power Saving (kWh)	2.172	14.24	8.857	25.966

F. *Carbon footprint and efficiency*

Finally, we plot the total CO2 emissions for both models using power generation values for Singapore in Fig. 8 and power generation values for the USA in Fig. 9

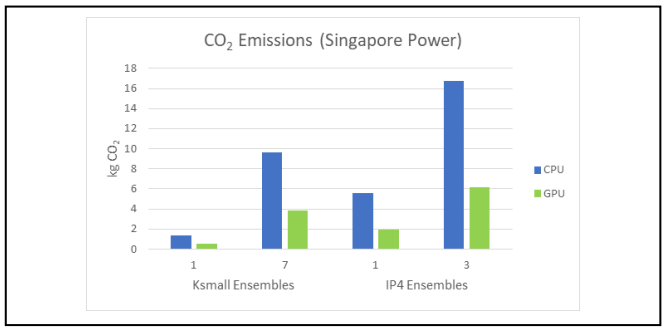

Fig. 8. Total CO_2 Emissions for Singapore (in kg CO_2)

For single members we see a reduction of 2.78 times and 2.83 times CO2 emissions for the small and large cases, respectively. For fully GPU loaded ensembles the reduction is 2.50 times for the small case and 2.72 times for the large case.

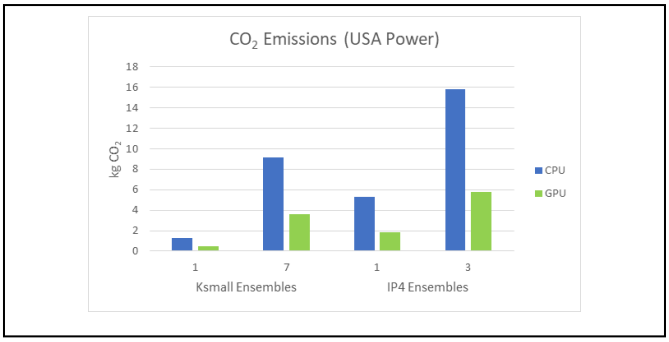

Fig. 9. Total CO_2 Emissions for USA (in kg CO_2)

V. SUMMARY

Our study confirmed that GPU accelerated forecasting can provide tangible value in reducing energy consumption, leading to reduced operational cost and lowered greenhouse gas emissions. We measured a minimum of 250% reduction and up to a 283% reduction in CO2 emissions for the various experiments. Extrapolating the reduction in CO2 emissions over longer timescales demonstrates the potential for greater savings. For 24 hour runs using Singapore power, we save between 388 – 468 kg per run per GPU. Annualised, this would equate to 141 – 171 tonnes of saved emissions (134-161 tonnes for US power). That is the equivalent of removing up to 36.8 gasoline-powered vehicles from the roads for a year[2]

The main conclusion from our research is that the use of the MIG facility in GPU accelerated computing allows us to utilise the GPU resources more completely and therefore more efficiently. This effect is particularly demonstrated by workloads with a larger number of smaller jobs. This allows us to deliver improved carbon emission reductions at the application level, rather than simply at a kernel level.

One negative effect that we noted during the experiments was that the I/O time per ensemble member grew much larger than expected as the number of members computed simultaneously was increased. This led to longer run times and this effect somewhat reduces the overall benefit of using MIG. We suspect that the I/O may be a bottleneck in getting the data to the storage over the PCIe interface as all members attempt to write at the same time. We theorise that staggering the ensemble members by a small amount, such that each MIG slice is doing I/O at a different time, may be one method to mitigate this effect and testing this theory could be a good future extension of this work.

REFERENCES

[1] A. Anders, and T Edler. 2015. "On global electricity usage of communication technology trends." Challenges 6. 117-157.

[2] J. M. Cebrian, G. D. Guerrero and J. M. Garcia, "Energy Efficiency Analysis of GPUs", 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, Shanghai, China, 2012, pp. 1014-1022, doi: 10.1109/IPDPSW.2012.124

[3] D. Chen, M Rojas, B H Samset, K Cobb, A Diongue Niang, P Edwards, S Emori, et al. 2021. "Climate Change 2021: The Physical Science

[2] Source: epa.gov/energy/greenhouse-gas-equivalencies-calculator, accessed November 2022

Basis." the Sixth Assessment Report of the Intergovernmental Panel on Climate Change (Cambridge University Press).

[4] Oliver Fuhrer, Tarun Chadha, Torsten Hoefler, Grzegorz Kwasniewski, Xavier Lapillonne, David Leutwyler, Daniel Lüthi, et al. 2017. "Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0." Geoscientific Model Development 11 (4): 1665-1681. Accessed 3 27, 2021. https://geosci-model-dev.net/11/1665/2018/gmd-11-1665-2018.pdf.

[5] Thomas R. W. Scogland, Craig P Steffen, Torsten Wilde, Florent Parent, Susan Coghlan, Natalie Bates, Wu-chun Feng, and Erich Strohmaier. 2021. "A Power-Measurement Methodology for Large-Scale, High-Preformance Computing." ICPE.

[6] W. C. Skamarock, B. J. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. Duda, X. Y. Huang, W. Wang, and J. G. Powers. 2008. "A Description of the Advanced Research WRF Version 3." NCAR Tech. Note TN-475+STR: 113. doi:10.5065/D68S4MVH.

[7] TempoQuest Inc. 2022. tempoquest.com/acecast. 12. Accessed 12 12, 2022. https://tempoquest.com/acecast.