Reduced precision on a Microphysics model

1st Marcelo A. Sudo
Institute of Science and Technology
University of São Paulo
São José dos Campos-SP, Brazil
marcelo.sudo@unifesp.br

2nd Álvaro L. Fazenda
Institute of Science and Technology
University of São Paulo
São José dos Campos-SP, Brazil
alvaro.fazenda@unifesp.br

3rd Roberto P. Souto
National Laboratory of Scientific Computing
Petropolis-RJ, Brazil
rpsouto@lncc.br

***Abstract*—In high-performance computing (HPC), mixed-precision techniques—where calculations use varied levels of precision—seek to boost performance with a little impact on output quality. This study explores integrating mixed-precision arithmetic on a module into the MPAS atmospheric model. The analysis focuses on balancing precision with computational efficiency and ensuring accuracy within acceptable limits. Applying lower precision the study shows improved energy efficiency and throughput in atmospheric modeling, especially on GPUs. Preliminary results for the WSM6 microphysics model in MPAS on both GPUs and CPUs revealed that energy efficiency (GFLOPS/W) increases as precision decreases, with notable speedups on GPUs—up to $109\times$ when using FP16 compared to FP64 on CPUs. Accuracy analysis, using Mean Squared Error (MSE), indicated small precision trade-offs, with values ranging from 10^{-11} to 10^{-9}.**

***Index Terms*—Reduced precision, GPU, Energy efficiency.**

I. Introduction

Approximate Computing (AC) represents a paradigm that challenges the traditional notion of requiring maximum precision in computational tasks. It operates on the principle that employing simpler, less accurate approximate functions instead of precise ones can yield benefits in power consumption and performance, particularly since not all applications demand exact results [1].

Among the various AC methods, the reduction of floating-point precision—commonly known as mixed precision—has gained significant traction recently. Studies have demonstrated that this approach can markedly improve the performance of scientific applications [2].

This study aims to implement mixed precision as an AC technique within the MPAS numerical atmospheric model, specifically focusing on the WSM6 microphysics. This builds upon the research conducted by Kim, Kang, and Joh [3], which executes the module on a GPU by using a source code with OpenACC directives, and evaluates the potential performance and energy efficiency gains about any resultant accuracy losses.

A. Related Work

In [3], the authors optimized the WSM6 microphysics routine within the MPAS model using OpenACC directives. They parallelized WSM6 on a GPU aiming to reduce data transfer between the CPU and GPU. This approach led to an average speed-up of $2.38\times$ compared to 48 MPI processes and a $5.71\times$ speed-up when excluding I/O communication.

In [4], the authors applied mixed precision to matrix multiplication and stencil algorithms using OpenACC directives. For the first one, they achieved a $16.60\times$ speed-up by using the ”Matmul” intrinsic function with FP16 on Tensor Cores, compared to a naive FP64, with accuracy loss from 10^{-26} to 10^{-1}. On the other, reducing precision from FP64 to FP16 led to a $1.60\times$ speed-up, with accuracy loss up to 10^{-9}.

Our study applied reduction precision converting variable types, i.e. comparing double, float, and half precision, and our purpose was to verify the speedup and energy gains with these variations versus the precision loss.

II. Methodology

In this study, we built upon the research conducted by Kim, Kang, and Joh [3], where they implemented OpenACC directives to optimize the MPAS code, initially designed for CPU execution using MPI, to run on a GPU device. Our investigation extended this work by examining reduced precision with a focus on Approximate Computing. We analyzed performance, accuracy, and energy costs across different precision levels: double, single, and half precision.

While the previous study addressed performance metrics for double and single precision, additional modifications to the source code were necessary for half-precision in WSM6. The porting process involved converting all GPU variables to FP16 format and validating the accuracy of these changes.

The results highlight the percentage increase in execution time, using FP64 on the CPU as a reference. Additionally, we computed the speedup and GFLOPS. Accuracy was assessed using Mean Squared Error (MSE) based on CPU execution with FP64, alongside the performance evaluation results.

A. Experimental Setup

The hardware used for the experiment is a server with two Intel Xeon Gold 6252 @ 2.10GHz, and 384GB as main memory, with four NVIDIA V100-16GB GPU cards as hardware accelerator hosted on a supercomputing at the National Scientific Computing Laboratory (LNCC), a Brazilian institution of the Ministry of Science, Technology and Innovation and Communications (MCTIC) specialized in scientific computing.

The source-code is available at: https://github.com/marcelosudo/MPAS_wsm6_GPU_for_CAG.

The study collected energy measurements using ”nvidia-smi” at 100ms intervals, capturing power consumption in

Watts. Energy efficiency was evaluated using GFLOPS/W, calculated by measuring floating-point operations with PAPI on the CPU. Energy data included the average and maximum values from ”nvidia-smi” outputs.

III. Results

Figure 1 illustrates the increasing energy efficiency in GFLOPS/W, being the maximum value (blue bar) and the mean value among all the energy measurements (red bar). The preliminary analysis considered meteorological forecasts for just 1 hour ahead. It was observed that energy efficiency increases, as desired, as floating point precision decreases.

Table I represents a 1-hour forecast considering FP64, FP32, and FP16 in CPU and GPU. It includes the execution time of WSM6 in seconds, along with standard deviation, speedup, and GFLOPS. The reference execution was always FP64 in the CPU, highlighted in bold. There was a speedup of 1.14 between FP64 and FP32 in the CPU, and the GPU exhibited an impressive speedup of 109.28 using FP16 precision compared to FP64 on the CPU. It is important to note that the base CPU executions for performance evaluations use only one thread/process, despite the possibility of executing MPAS with several processes with MPI, which certainly does not represent the best performance possible for the model.

The accuracy analysis, necessary since the reduced floating point precision implies a possible increase of numerical truncation errors, focused on a specific group of arrays (qv, qc, qi, qr, qs, qg), as used by [3], relevant to the study of precipitation with WSM6 model, representing meteorological fields known as hydrometeors, which means the mixing ratios of water vapor (qv), cloud water (qc), cloud ice (qi), rain (qr), snow (qs), and graupel (qg), respectively.

The analysis included the Mean Squared Error (MSE), keeping the FP64 execution over the CPU as a base. The Table II shows MSE in the order of 10^{-11}, considering the mean of all six previously depicted hydrometeors.

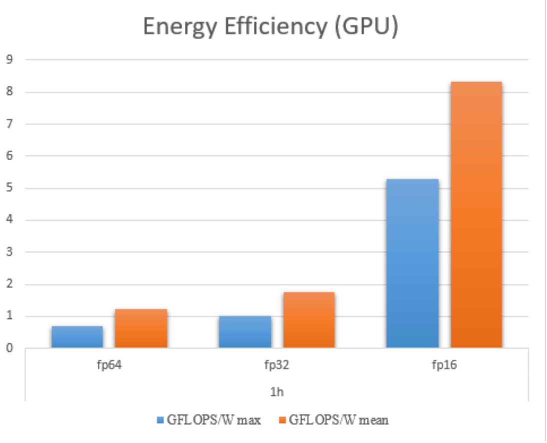

Fig. 1. Energy Efficiency in GPU, in GFLOPS/W for Max and Mean energy consumption, applied to FP64, FP32, and FP16, and forecast 1h.

1h	precision	time ± std (s)	speedup	GFLOPS
CPU	FP64	$\mathbf{44.25} \pm 0.73$	1.00	4.65
	FP32	38.76 ± 0.15	1.14	5.31
GPU	FP64	2.67 ± 0.56	16.60	77.19
	FP32	1.99 ± 0.02	22.28	103.61
	FP16	0.40 ± 0.01	109.28	508.21

TABLE I
Performance in CPU and GPU, for FP64, FP32 and FP16, 1h.

all vars	precision	MSE
CPU	FP64	**0.00**
	FP32	1.39E − 11
GPU	FP64	1.25E − 11
	FP32	2.06E − 11
	FP16	6.62E − 09

TABLE II
Accuracy in CPU and GPU, for FP64, FP32 and FP16, 1h.

IV. Conclusion

In this study, we analyzed the applications of reduced precision to verify the trade-off between performance gains, accuracy losses, and energy savings in a microphysics model (WSM6) embedded in a well-known meteorological numerical model MPAS. The results from changing FP64, FP32, and FP16 in floating point operations executed over GPUs are mutually compared, expanding the achievements obtained in [3]. The preliminary results showed an extraordinary speedup of 109.28 for a 1-hour prediction when comparing an FP16 GPU with an FP64 base CPU execution with one unique thread. Accuracy showed that there were no significant losses, in the order of 10^{-09} of MSE, for the average of all six variables considered $(qv, qc, qi, qr, qs,$ and $qg)$ when comparing FP64 in CPU with FP16 in GPU. Regarding energy efficiency measured in GPU, in GFlops/W, getting the maximum Watt in the time interval corresponding to the GPU processing for microphysics model WSM6, it was achieved 5.29 in FP16 against 0.71 for FP64 in GPU, which means 86.57% energy savings. For future works, we intend to extend the integration time and apply the same methodology to other numerical models, likewise considering performance with parallel processing over MPI on the CPU.

Acknowledgment

The authors acknowledge the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing HPC resources. This work was also partially financed by grant #2019/26702-8, São Paulo Research Foundation (FAPESP).

References

[1] E. Hussein, B. Waschneck and C. Mayr, 2024. “Automating application-driven customization of ASIPs: A survey,” Journal of Systems Architecture, DOI: https://doi.org/10.1016/j.sysarc.2024.103080.

[2] K. Parasyris, I. Laguna, et. al, 2020. “HPC-MixPBench: An HPC Benchmark Suite for Mixed-Precision Analysis,” IEEE Int. Symp. on Workload Characterization (IISWC). DOI:10.1109/IISWC50251.2020.00012.

[3] J. Kim, J. Kang and M. Joh, 2021. “GPU acceleration of MPAS microphysics WSM6 using OpenACC directives: Performance and verification,” DOI: https://doi.org/10.1016/j.cageo.2020.104627.

[4] M. Sudo, A. Fazenda and R. Souto. 2022. ”Mixed precision applied on common mathematical procedures over GPU.” 23rd Brazilian Symp. on HPC Systems. DOI: https://doi.org/10.5753/wscad.2022.226312.