Virtual Student Cluster Competition 2020, team Clemson University: Reproducing Performance of Massively Parallel Memory-Centric X-ray CT Reconstruction with MemXCT

Team 15, Dabo's Tech Support[1]

Clemson University, SC, USA

Abstract

This paper seeks to reproduce and verify the results of the SC'19 paper, MemXCT: Memory-Centric X-ray CT Reconstruction with Massive Parallelization by M. Hidayetoğlu et al. While exact duplicate hardware was not feasible, similar hardware and configurations were used to recreate the environment of the paper. This paper recreates the Single CPU-GPU performance experiments and the Strong Scaling experiments and compares the results to the original paper. Overall, our results match the original paper to the best ability of the utilized hardware. Differences in memory bandwidth are due to the use of GPUs rather than the original paper's KNL architecture.

Keywords: Reproducible computation, Student Cluster Competition

1. Introduction

Each year for the Reproducibility Challenge at the Student Cluster Competition, teams are required to reproduce results from a paper accepted to the previous year's SC Technical program. For this year's reproducibility challenge, teams were tasked to reproduce results from M. Hidayetoğlu et al. and their SC'19 paper, MemXCT: Memory-Centric X-ray CT Reconstruction with Massive Parallelization. This paper presents MemXCT, a system that uses a memory-centric approach to iterative reconstruction of X-ray computed tomography while avoiding redundant computation [1]. In this report, we detail our reproduction of the single CPU-GPU performance and strong scaling results. Our testing methodology for the reproducibility challenge consisted of determining the correct tuning parameters and Azure Virtual Machine (VM) type most accurately represents the hardware used in these tests. Thus, we settled on using a configuration that would most closely resemble Blue Waters in terms of it's GPU performance.

2. Experimental Setup

In order to provide the most accurate results, a separate cluster was used for reproduction of CPU-GPU performance and strong scaling tests. For scaling tests we used a virtual cluster, while for the single CPU-GPU performance tests we used single VMs to simplify setup. SKU numbers for both configurations are shown in Table 1 and their respective hardware specifications are shown in Table 2.

Test	VM SKU	VM Count
Strong Scaling	NC12	4
CPU-GPU Performance	NC12	1
	NC12_v2	1
	NC12_v3	1

Table 1: Reproducibility VM SKU Specifications [2].

In the hardware configuration shown in Table 2, each node contains 12 CPU cores and 2 GPUs. We chose this configuration in order to provide the most similar GPU performance to Blue Waters, which is used as a test system in the MemXCT paper. Some variation in performance is attributed to performance

[1] cavendh@clemson.edu, gdube@clemson.edu, hollowe@clemson.edu, nheitze@clemson.edu, sansrir@clemson.edu, splacke@clemson.edu

-	NC	NC_v2	NC_v3
CPU	Intel Haswell	Intel Broadwell	
CPU Mem.	112 GiB	224 GiB	
GPU	K80	P100	V100
GPU Mem.	24 GiB	32 GiB	
Interconnect	Ethernet		

Table 2: VM Hardware Specifications [2].

differences between Blue Waters' Interlagos processors and K20X GPUs and to our previously described configuration.

We use Azure Cyclecloud's Cycle CentOS 7 image as our operating system. In order to compile MemXCT, we employ gcc 9.2.0, OpenMPI 4.0.3, and CUDA 11.1, and we submit jobs using PBSPro. In reproduction of results, we use two provided competition datasets – similar Shale Sample RDS1 [1] – as our testing input. In order to prevent the need to recompile software each time, persistent storage was added to our virtual clusters at `/persist`.

Runs of the experiments mentioned in the MemXCT paper are replicated as closely as possible. Due to limitations of the hardware used, we are only able to perform vectorization using AVX2, which uses vectorization registers up to 256 bits, as opposed to AVX-512. Further, memory bandwidth measurements are recorded for GPU nodes as opposed to CPU nodes due to the lack of KNL VMs on Azure.

3. Description of Experimental Run

We planned our experimental runs such that an average of the performance and timing results could be taken, protecting results from any variability in the system. We repeated our experimental runs five times for each experiment and each dataset. This experimental design is chosen in order to provide a high quality result without excessive use of cloud resources. Upon completion of our experiments we address performance differences between the experiments run in the paper and our own results.

Compiling MemXCT is done using the `gpuinit.sh` script located in the `/compile` directory of the digital artifact. This script installs the necessary software, mentioned in Section 2, then sets the necessary environment variables and compiles the application. The executable created during this step are submitted and run with the PBS scripts, `azure_scaling.pbs` and `azure_perf.pbs`, located in the `run/scripts` directory of the digital artifact.

4. Single CPU-GPU Performance Comparison

Single CPU-GPU performance is measured on three versions of Azure's NC12 VMs. Each VM contains a different GPU type, which are used to reproduce the GPU Performance Optimization and Memory B/W Utilization graphs from Figure 9 in the MemXCT paper [1]. These results are shown in Figure 1 and Figure 2.

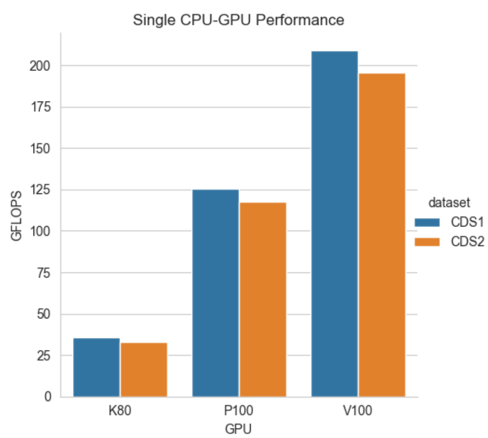

Figure 1: Reproduced GPU GFLOPS Performance

The GPU Performance results shown in Figure 1 shows very similar performance to that shown in the paper, largely due to the use of the same GPU types. Variations in the input datasets, and newer CPUs attribute to the degradation in performance because the higher computation speed results in performance bounded by communication. The reproduced results in Figure 2 are fairly different to those observed in the MemXCT paper due to the use of GPU nodes as

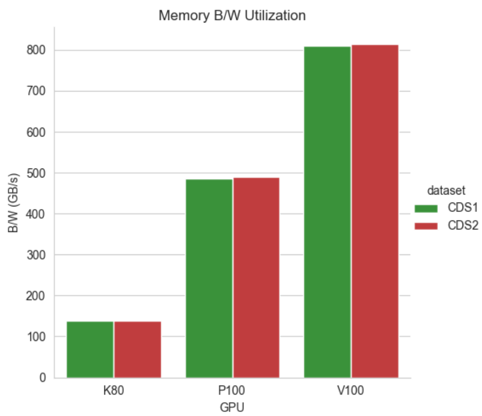

Figure 2: Reproduced GPU Memory B/W Utilization

opposed to KNL nodes. For K80 GPU nodes, memory bandwidth utilization performs poorest, while the memory bandwidth utilization on V100 and P100 nodes compares more closely to the paper's results. This is most likely due to differences in the memory bandwidth speeds of the V100, P100, and K80 GPUs.

5. Strong Scaling on GPUs and CPUs

Strong scaling experiments take place over a much smaller set of nodes when compared with those run in the the MemXCT paper. This is largely due to a lack of compute resources available to us on Azure, however the trends shown can be compared to those investigated in the paper. We perform strong scaling experiments using between one and four NC12 VMs in Azure. The average timing results of these runs are shown below.

Our strong scaling results are shown for an average of all our runs, and for both competition datasets, CDS1 and CDS2. These results show the Total, Kernel (A_p), Communication (C) and Reconstruction (R) times for each dataset. Although the number of nodes used is not identical to the number of nodes used in scaling tests for the MemXCT paper, we see similar trends in our scaling results. This proves that our experimental and hardware setup replicates the results from the paper well.

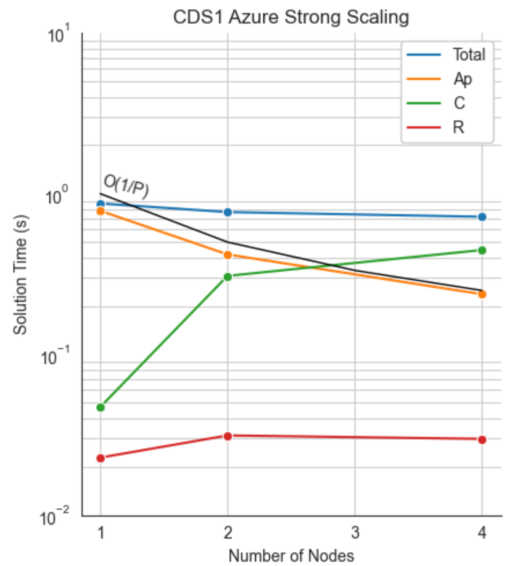

Figure 3: Strong Scaling, with CDS1

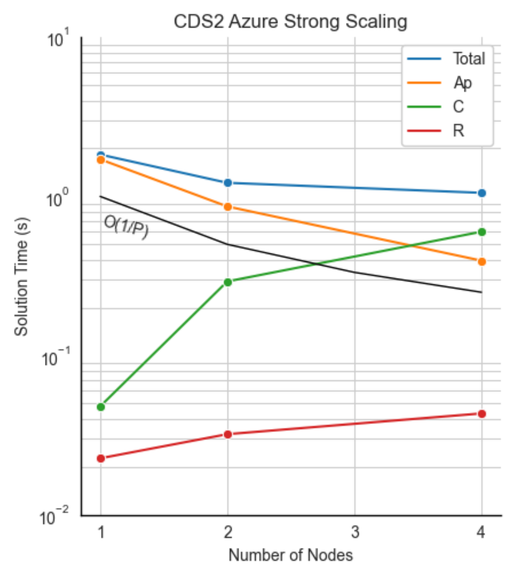

Figure 4: Strong Scaling, with CDS2

6. Visualization

Figures 5 and 6 show the reconstructed outputs for CDS1 and CDS2 generated with the Fiji tool. Correct reconstruction of these shale samples is done following the steps in the `README` located in the directory `/figures/scripts` of our digital artifact. Correct output with respect to these images shows the full image, reconstructed iteratively with MemXCT.

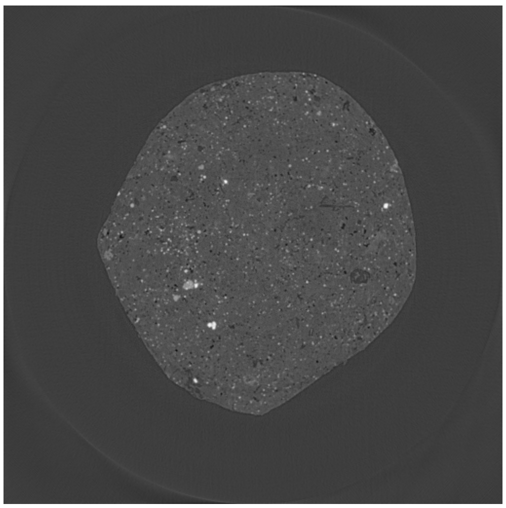

Figure 5: CDS1 Reconstruction Image

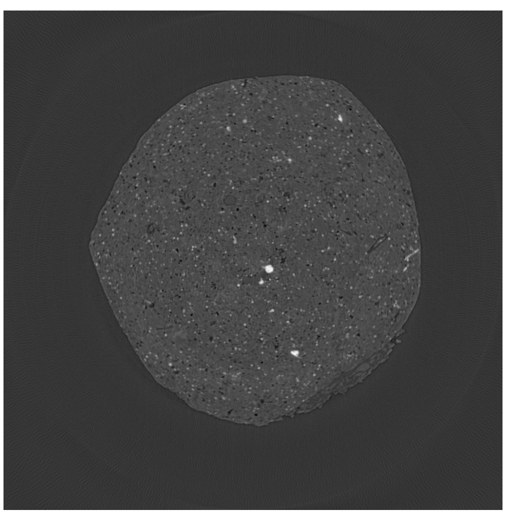

Figure 6: CDS2 Reconstruction Image

7. Conclusion

Although our results were not accurate to those produced in the paper, many similar trends exist between the two sets of results. Single CPU-GPU performance tests show similar results to the paper in terms of performance, but differ in memory bandwidth utilization due to a lack of similar hardware on Azure. Our strong scaling results present very similar trends to those seen in the MemXCT paper on a small number of nodes. Increasing this scale would provide similar results to those discussed in the paper. Reproducing these results confirm validity of the results presented by the MemXCT paper and show that this application is capable of running on a variety of different hardware with consistent performance.

Acknowledgements

We would like to acknowledge the guidance and support of our team advisor, Dr. Jon Calhoun. We would also like to acknowledge the support of Clemson Computing & Information Technology and Dell.

References

[1] M. Hidayetoğlu, T. Biçer, S. G. de Gonzalo, B. Ren, D. Gürsoy, R. Kettimuthu, I. T. Foster, W.-m. W. Hwu, Memxct: Memory-centric x-ray ct reconstruction with massive parallelization, SC '19, Association for Computing Machinery, New York, NY, USA, 2019. `doi:10.1145/3295500.3356220`. URL `https://doi.org/10.1145/3295500.3356220`

[2] M. Azure, Linux Virtual Machines Pricing, `https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/` (2020).