



HAL
open science

Data distribution on a multi-GPU node for iterative Computed Tomography reconstruction

Mohammed Chghaf, Nicolas Gac

► **To cite this version:**

Mohammed Chghaf, Nicolas Gac. Data distribution on a multi-GPU node for iterative Computed Tomography reconstruction. 2020. hal-02586239v2

HAL Id: hal-02586239

<https://hal.science/hal-02586239v2>

Preprint submitted on 20 May 2020 (v2), last revised 16 Jun 2020 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data distribution on a multi-GPU node for iterative Computed Tomography reconstruction

Mohammed Chghaf

Université Paris-Saclay, CentraleSupélec, L2S
Gif-sur-Yvette, France
mohammed.chghaf@gmail.com

Nicolas Gac

Université Paris-Saclay, CentraleSupélec, L2S
Gif-sur-Yvette, France
nicolas.gac@universite-paris-saclay.fr

Abstract—Computed tomography (CT) is an imaging technique that uses iterative algorithms to reconstruct the interior of a volume. Graphics Processing Units (GPUs) are currently the preferred technology for reconstruction in CT given their computational performance. In order to have a good reconstruction, the number of images needed is very large. Therefore, the collected data requires several Gigabytes of memory storing and can exceed the internal memory of current GPUs. In this paper, we present a strategy of data distribution between multiple GPUs. We provide experimental results showing that our memory-saving method increases the time of effective computing. We achieve a parallelization efficiency factor of 1 using 8 GPUs to reconstruct a volume of size 4 Gigabytes.

Index Terms—Computed Tomography, Multi GPU, Data Distribution, Iterative reconstruction

I. INTRODUCTION

Using GPUs based servers in iterative CT algorithms has become almost a necessity due to their ability to reduce significantly the calculation time [1]. However, the internal memory of GPUs is limited and can not store the data and the reconstruction result. Indeed, some CT applications, such as, medical imaging use volumes of size 2048^3 which requires 32 Gigabytes of memory storing. Hence, the use of multiple GPUs is essential to reconstructing volumes with high quality.

Previous studies have tackled the problem of multi-GPU parallelization. In [2], the data are centralized in the CPU. In each iteration, chunks of data are transferred from CPU to GPU in order to perform the parallel calculation. The results are then collected by the CPU. The strategy proposed in [3] is similar. The data are stored in the CPU and are partitioned into same size stacks between the GPUs. The stacks are then copied to the GPUs before every operation. Finally, the partial result of every GPU is copied to the CPU. The authors of [4] proposed a distributed solution over multiple nodes in a computing cluster. The communication between nodes and between GPUs require the use of message-passing interface (MPI). Recent work refined this solution by using a geometric characterization to develop a partitioning algorithm [5].

In this paper, we propose an efficient data distribution strategy between multiple GPUs on the same node. This solution is based on high-speed peer-to-peer GPU communication. We provide experimental results of our method for an iterative algorithm.

II. ITERATIVE COMPUTED TOMOGRAPHY ALGORITHMS

The aim of Iterative CT algorithms is to create a three-dimensional (3D) representation of an object using only external measurements. The object is irradiated by an X-ray source while a flat detector is placed on the opposite side. The intensity of the X-rays decreases due to the object variable density before reaching the detector. The measurements of the decreased intensities by the cells of the detector form an image which is called a projection. In order to acquire several views, the object is rotated around Z-axis. A sinogram is created by stacking all of the measured attenuation from the cells of the detector acquired at different angles.

Mathematically, the projection operation can be expressed by an algebraic equation with matrix-vector product : $g = Hf$. Where f is the volume to be reconstructed, g is the vector of measurements and H is the projection matrix. The backprojection operation is represented by H^T .

In iterative reconstruction methods, the projection and the backprojection operators correspond to the main computational burden. Therefore, reducing the calculation time implies speeding up these two operations through GPU parallelization.

III. PROPOSED METHOD: DATA PARTITIONING

The main constraint of parallelization in iterative CT algorithms is data distribution between GPUs in the case of cone-beam acquisition. In the parallel-beam case, the problem can be easily solved by splitting the volume to equal slices perpendicular to the axis of rotation and equivalent to the number of GPUs used. The slices of the volume corresponding to the projections on the detector will be separate and will allow the calculations to be carried out independently. However, in the case of cone-beam CT, the divergent beams between the source and the detector can intersect more than one area of the volume. Therefore, to perform a projection where each GPU is responsible for a slice of the volume, some data from the two adjacent slices are needed to perform the necessary calculation of the projection in these overlapping regions.

The proposed strategy decomposes the volume and the projections into slices of the same number of the used GPUs. This makes it possible to copy from the CPU to the GPU only the bare minimum of data in order to speed up the calculations. The remaining data needed for computing is then transferred using only the high speed peer-to-peer GPU communication.

For the projection operator, the volume data are stored in the CPU, then we distribute them equally over all the GPUs used. Before starting the operation, we compute the limits of the sub-volume necessary to perform the calculation of each zone of the detector. The missing parts are then copied with direct peer-to-peer communication from the adjacent GPUs to the concerned GPU. Once the projection operation is completed, the measurements are then retrieved to the CPU.

Similarly, we can find the limits of each subset of projection measurements that will be used for backprojection for each GPU. The remaining data needed are then copied to the adjacent GPUs.

As shown in Fig. 1, generally, in order to calculate the projection P_n of a given sub-volume, we will need the sub-volume V_{n_middle} stored in GPU_n , the sub-volume V_{n_up} stored in GPU_{n+1} and the sub-volume V_{n_bottom} stored in GPU_{n-1} . In this case, V_{n_up} and V_{n_bottom} are copied using peer-to-peer communication from GPU_{n+1} and GPU_{n-1} respectively to GPU_n just before the projection operation.

The realization of the iterative algorithm requires intermediate operations such as the calculation of the differences between the estimated measurements and the real measurements, image filtering steps or reduction operations. The proposed solution allows all these operations to be carried out directly on the GPUs without the need for intermediate exchanges between CPU and GPU.

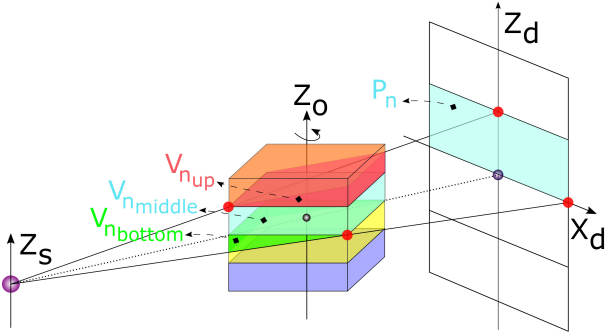


Fig. 1. Data partitioning for a forward projection in the case of cone-beam acquisition using 4 GPUs.

IV. RESULTS

The experiments conducted to evaluate the proposed strategy focus on a simplified version of the iterative CT algorithms and the intermediate operations are not taken into account. Therefore, at each iteration we calculate $f_{i+1} = \mathbf{H}\mathbf{H}^T f_i$.

The volumes and sinograms obtained with the projection and backprojection operators were validated by comparing them with those obtained with centralized data version of TomoBayes [2]. This comparison was elaborated for a 1024^3 (4 Gigabytes) sized Shepp-Logan ghost.

In order to assess the different evaluations obtained, we define η which is an indicator of the efficiency of the parallelization. (1) defines this quantity which always has a value between 0 and 1.

$$\eta = \frac{\text{Acceleration Factor}}{N_{GPU}} = \frac{t_{N_{GPU}}}{t_{1_{GPU}} \times N_{GPU}} \quad (1)$$

Where $t_{N_{GPU}}$ is the calculation time using N GPUs and $t_{1_{GPU}}$ is the time of calculation on a single GPU.

In practice, we tend to choose a large value for the number of iterations of the iterative algorithms in order to enhance the quality of the reconstructed volume. Thus, the loss of time in data transfert will be even greater in the classical solutions proposed in literature. Fig. 2 shows that our solution has a factor of parallelization efficiency almost equal to 1 while continues to decrease by increasing the number of GPUs in the centralized version.

Consequently, the memory transfer time tend to 0 in the distributed data version of the algorithm while it increases by augmenting the number of GPUs and/or iterations. In summary, the proposed optimization shows a clear improvement in performance of the iterative CT reconstruction algorithms.

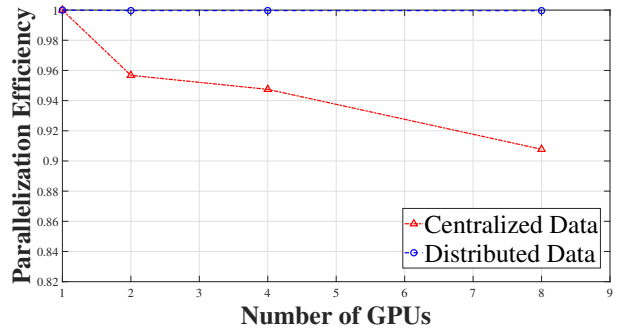


Fig. 2. Parallelization efficiency of the iterative algorithm for different GPU counts. 100 iterations for a volume size of 1024^3 .

V. CONCLUSION

In this work, we clarified the existing parallelization bottleneck. Furthermore, we presented a data distribution and synchronization strategy for iterative CT algorithms. To evaluate our method, we compared it to an existing multi-GPU iterative algorithm. As a result, we achieved a factor of parallelization efficiency of 1.

REFERENCES

- [1] ZHU, Yining, ZHAO, Yunsong, et ZHAO, Xing. A multi-thread scheduling method for 3D CT image reconstruction using multi-GPU. Journal of X-ray Science and Technology, 2012, vol. 20, no 2, p. 187-197.
- [2] BOULAY, Thomas, GAC, Nicolas, MOHAMMAD-DJAFARI, Ali, et al. TomoBayes v1. 0-logiciel de reconstruction en tomographie CT-Ref CNRS du pré-dépôt APP 11562-03 (num IDDN prochainement disponible). 2015.
- [3] BIGURI, Ander, LINDROOS, Reuben, BRYLL, Robert, et al. Arbitrarily large iterative tomographic reconstruction on multiple GPUs using the TIGRE toolbox. arXiv preprint arXiv:1905.03748, 2019.
- [4] PALENSTIJN, Willem Jan, BÉDORF, Jeroen, SIJBERS, Jan, et al. A distributed ASTRA toolbox. Advanced structural and chemical imaging, 2016, vol. 2, no 1, p. 1-13.
- [5] BUURLAGE, Jan-Willem, BISSELING, Rob H., PALENSTIJN, Willem Jan, et al. A projection-based data partitioning method for distributed tomographic reconstruction. In : Proceedings of the 2020 SIAM Conference on Parallel Processing for Scientific Computing. Society for Industrial and Applied Mathematics, 2020. p. 58-68.