PERFORMANCE ANALYSIS OF DISTRIBUTED GPU-ACCELERATED TASK-BASED WORKFLOWS

Marcos Nogueira Lobo de Carvalho[1,2,3];
Anna Queralt[1]; Oscar Romero[1]; Alkis Simitsis[2];

Home Institution: Universitat Politècnica de Catalunya (UPC)[1]
Host Institution: Athena Research Center (ARC)[2]|University of Athens (UoA)[3]
Contact: marcos.nogueira@upc.edu

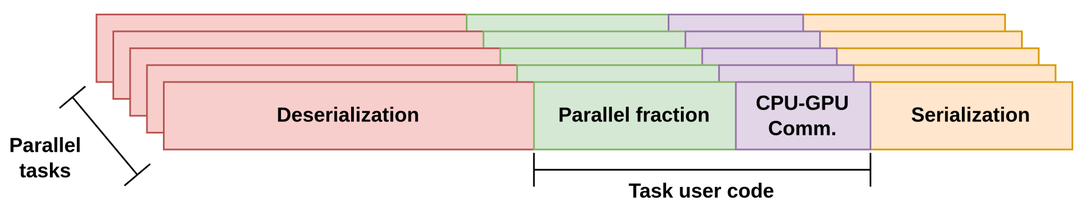
Motivation

Research Question: What are the main factors to consider to run distributed GPU-accelerated task-based workflows efficiently?

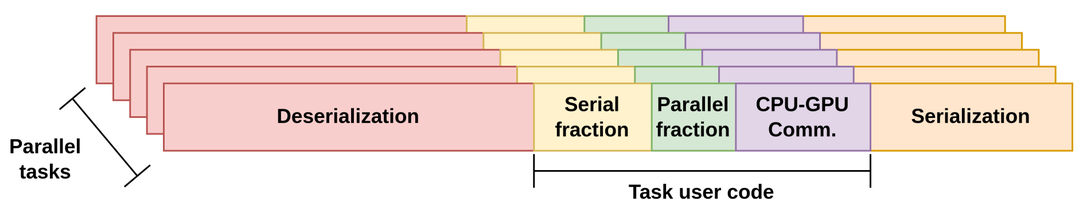
Contributions:

- A systematic performance analysis of thread-level parallelism:
  - Our results reveal that the gains provided by GPUs are highly affected by the parallel fraction processing within tasks
- A systematic performance analysis of task-level parallelism:
  - Our results reveal that depending on the amount of tasks, scheduling policy and storage architecture used, data (de-)serialization can be significantly high, dominating the total execution time
- Feature extraction considering algorithm, data set, resource and distributed execution framework:
  - We demonstrate that these features are highly correlated with the performance in terms of execution time and, therefore, can be considered as key features to characterize the execution of such workflows

Multi-level Parallelism



(a) Massively parallel algorithm (e.g. Matmul)



(b) Slightly parallel algorithm (e.g. K-means)

Figure 1: Monitored metrics in each class of algorithms

Execution Setup

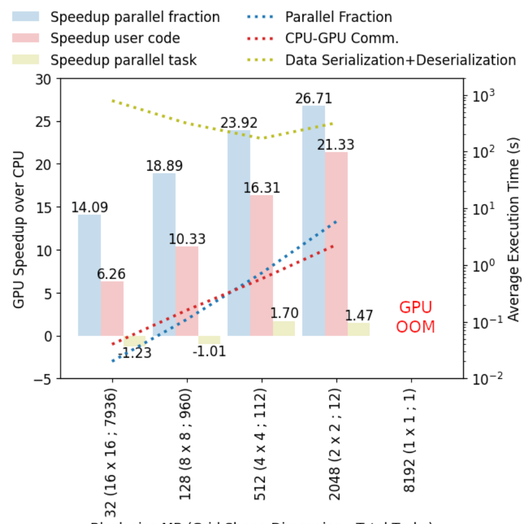
Total worker nodes	8	CPU-GPU bus	PCIe 3.0
Total CPU cores	128	Total GPU devices	32
RAM memory per node	128 GB	GPU memory per device	12 GB
CPU model	Intel Xeon E5-2630	GPU model	NVIDIA K80

Table 1: Cluster settings

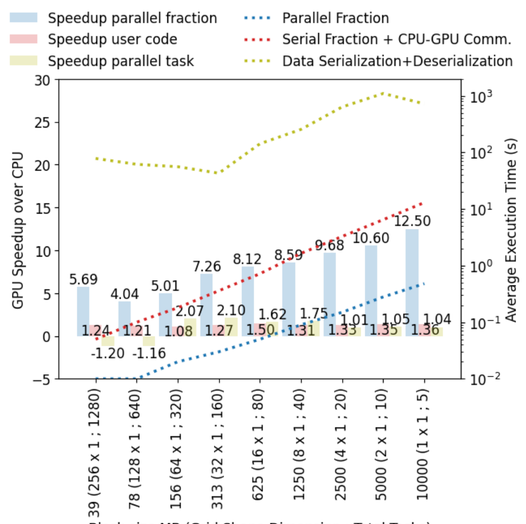
Algorithm	Task	Computational Complexity	Data Set
Matmul	*matmul_func*	$O(N^3)$	8 GB (32000 x 32000)
	add_func	$O(N)$	
K-means	*partial_sum*	$O(MNK^2)$	10 GB (12500000 x 100)

Table 2: Algorithm tasks and data sets (M and N are the number of rows and columns in a block, respectively, and K is the number of clusters)

End-to-end Analysis

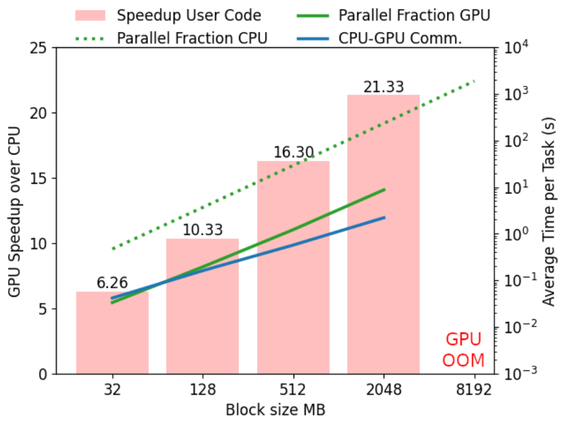


(a) Matmul

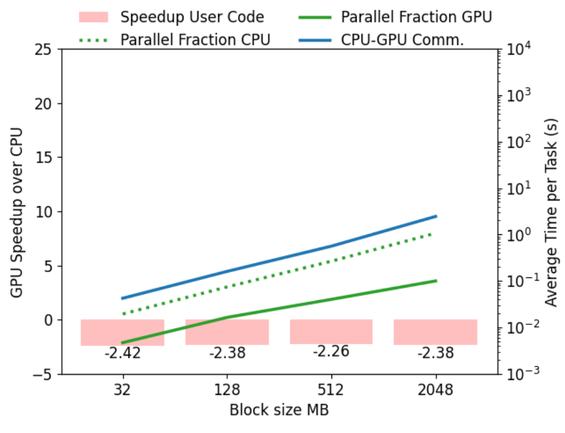


(b) K-means

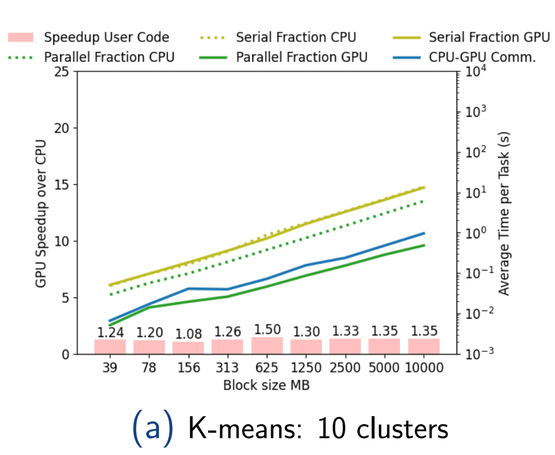
Performance Analysis of Task User Code Processing



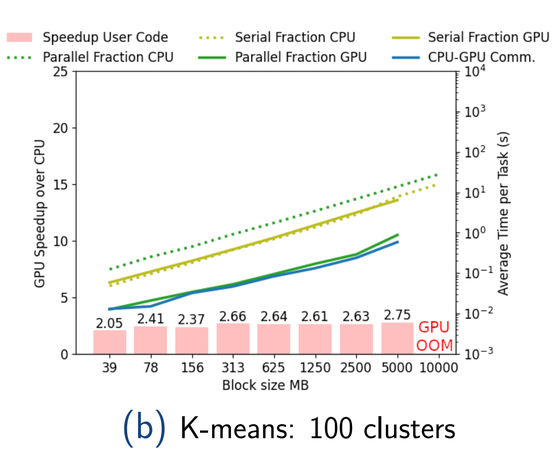
(a) Matmul: task *matmul_func*



(b) Matmul: task *add_func*



(a) K-means: 10 clusters



(b) K-means: 100 clusters