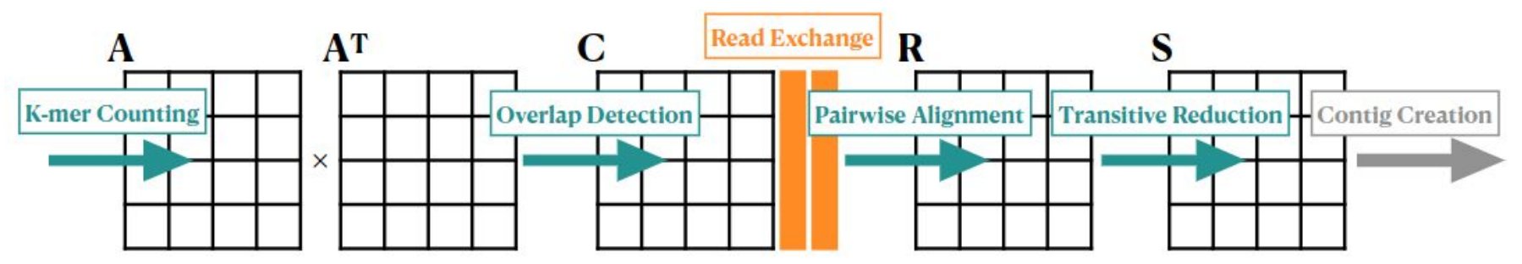
GPU Scheduler for *De Novo* Genome Assembly with Multiple MPI Processes

Minhao Li, Siyu Wang, Gary Wei

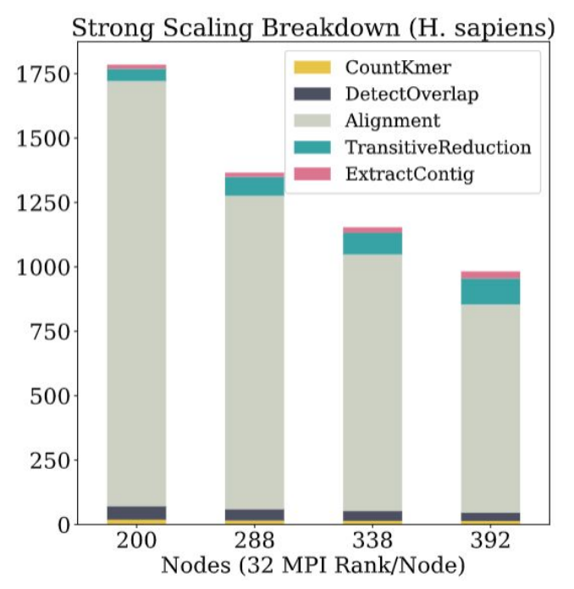
{ml2499, sw988, gw338}@cornell.edu

Introduction

We are trying to improve the efficiency of constructing unknown genome sequences from a set of short, fragmented sequences. The state of the art approach, ELBA, uses MPI and OpenMP to accelerate the calculation on CPU.



The bottleneck and most time consuming part is pairwise alignment, which can potentially take up to 90% of the total runtime.



(Giulia Guidi, Oguz Selvitopi, Marquita Ellis, Leonid Oliker, Katherine Yelick, Aydin Buluç. Parallel String Graph Construction and Transitive Reduction for De Novo Genome Assembly. IPDPS 2021)

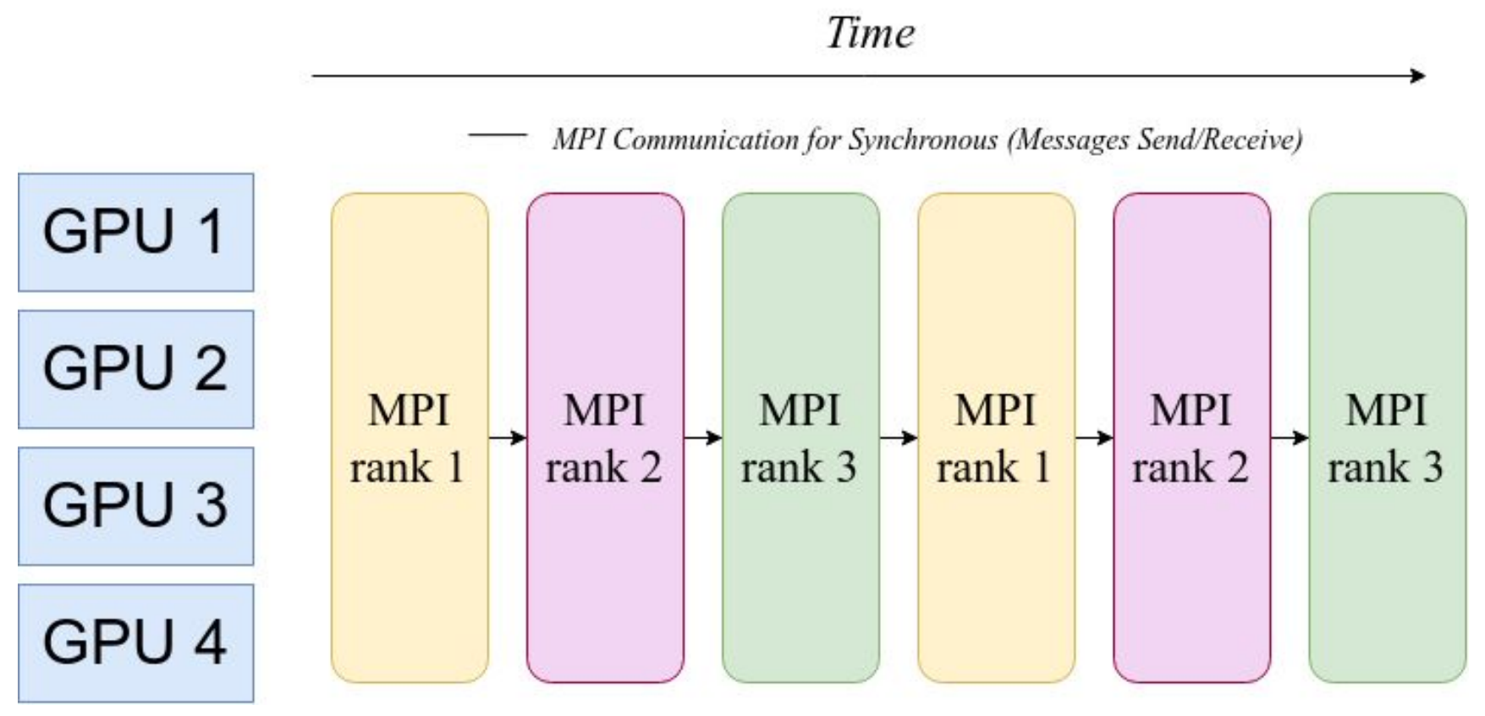
Recent research has successfully used GPU to accelerate pairwise alignment. However, ELBA can only accommodate 1 MPI process in order to exploit multiple GPUs for alignment. We aim at implementing a GPU scheduler such that the whole algorithm can benefit from both MPI and CUDA acceleration.

Aim

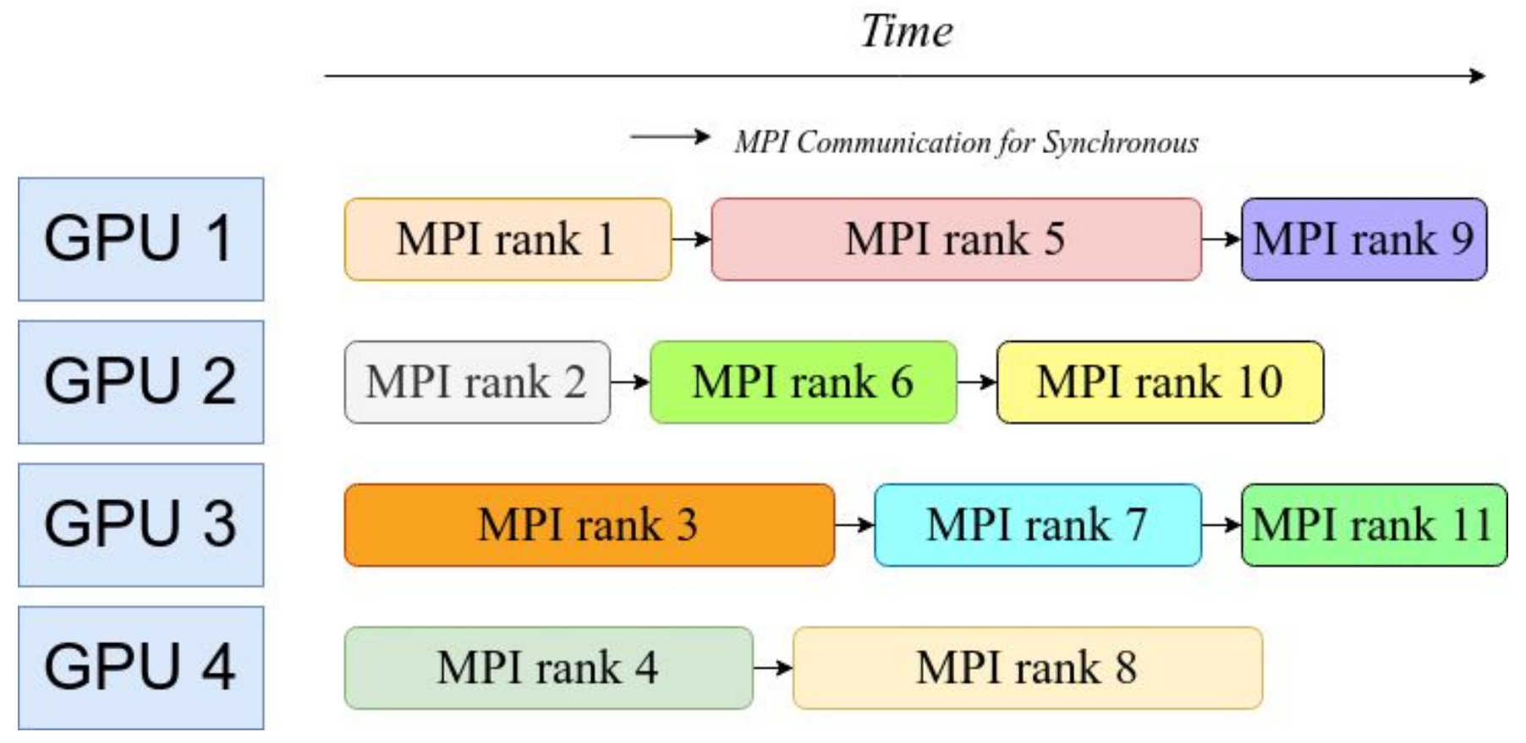
- Support multiple MPI processes in GPU-based pairwise alignment
- Design algorithms to efficiently communicate between MPI processes
- Achieve good strong scaling for multiple MPI processes
- Compare and reason experiment results of different scheduling policies

Method

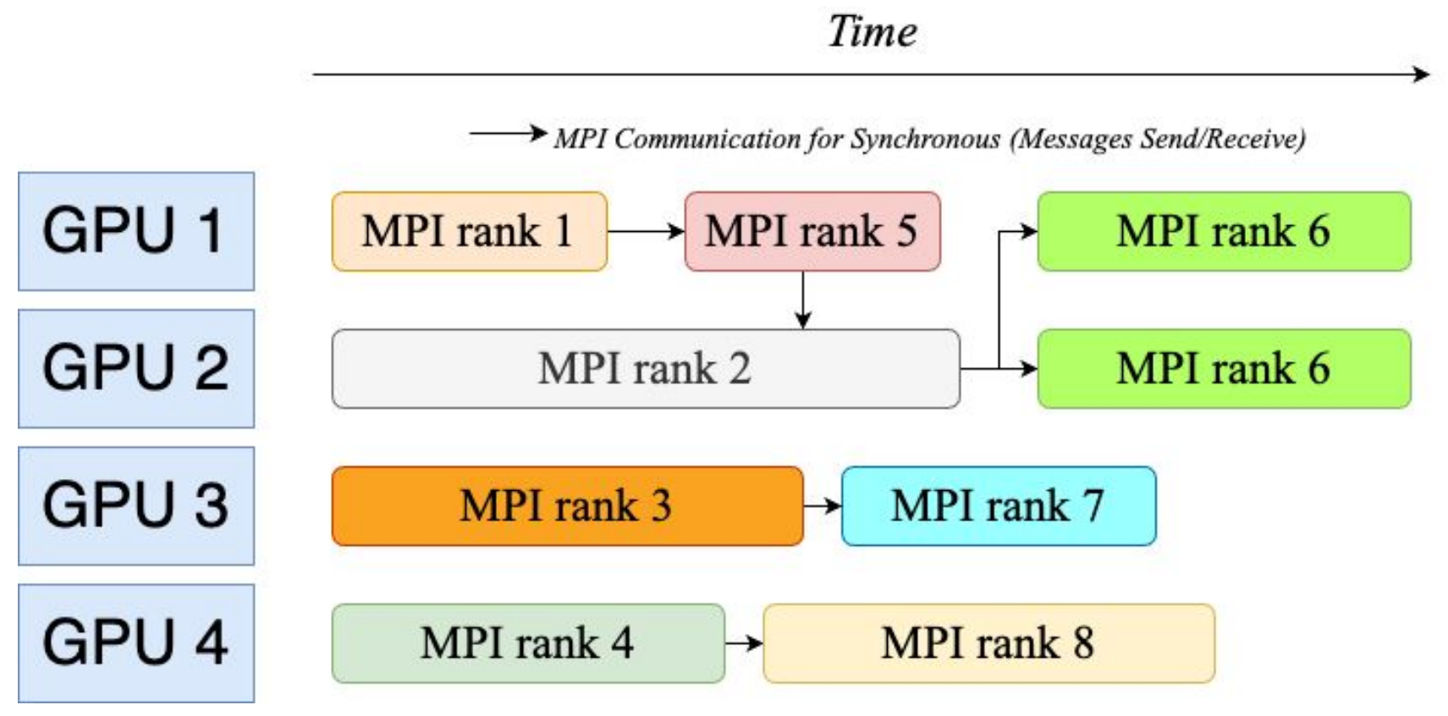
1. **Each MPI Proc Uses All GPUs**
   - **Description**: Each process uses all the GPU resources in round-robin.
   - **Pros**: Easy to implement. Low communication cost.
   - **Cons**: Only one MPI process is using GPU. Does not fully exploit the benefits of multiple processors.



2. **One Proc Uses One GPU**
   - **Description**: Processes are divided into #GPU pipelines, and processors in the same pipeline use their assigned GPU in round-robin.
   - **Pros**: Better parallelism, as #GPU processors are moving memory to separate GPU at the same time.
   - **Cons**: Uneven work between pipelines may cause load imbalance on different GPUs.

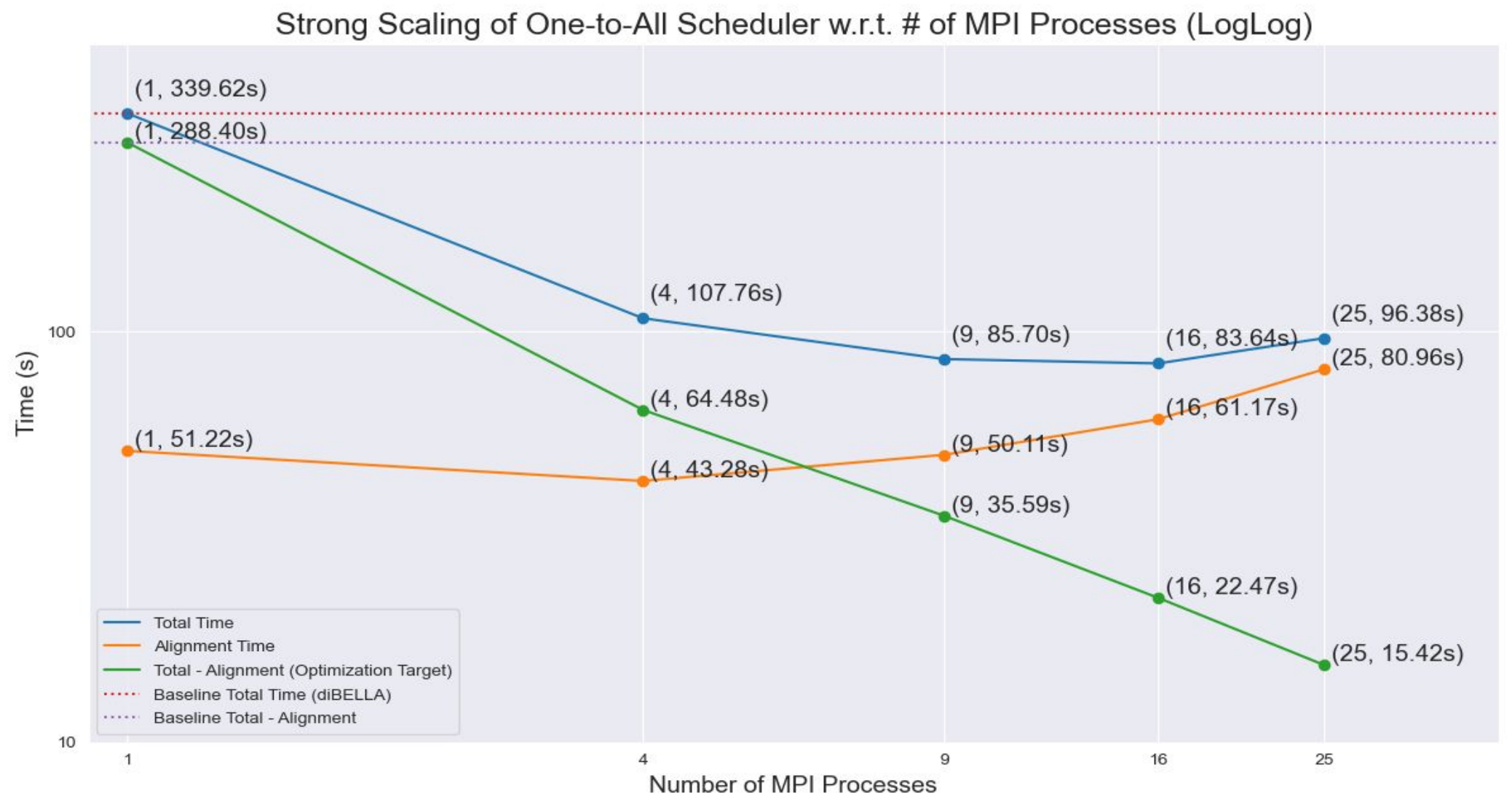


3. **Procs Use Free GPU(In Progress)**
   - **Description**: Processes are divided into #GPU pipelines. Each pipeline releases GPU resource to other pipelines on completion
   - **Pros**: Better load balancing for usage of GPUs.
   - **Cons**: Checking free GPUs before each iteration at MPI level increases the communication cost.



Results

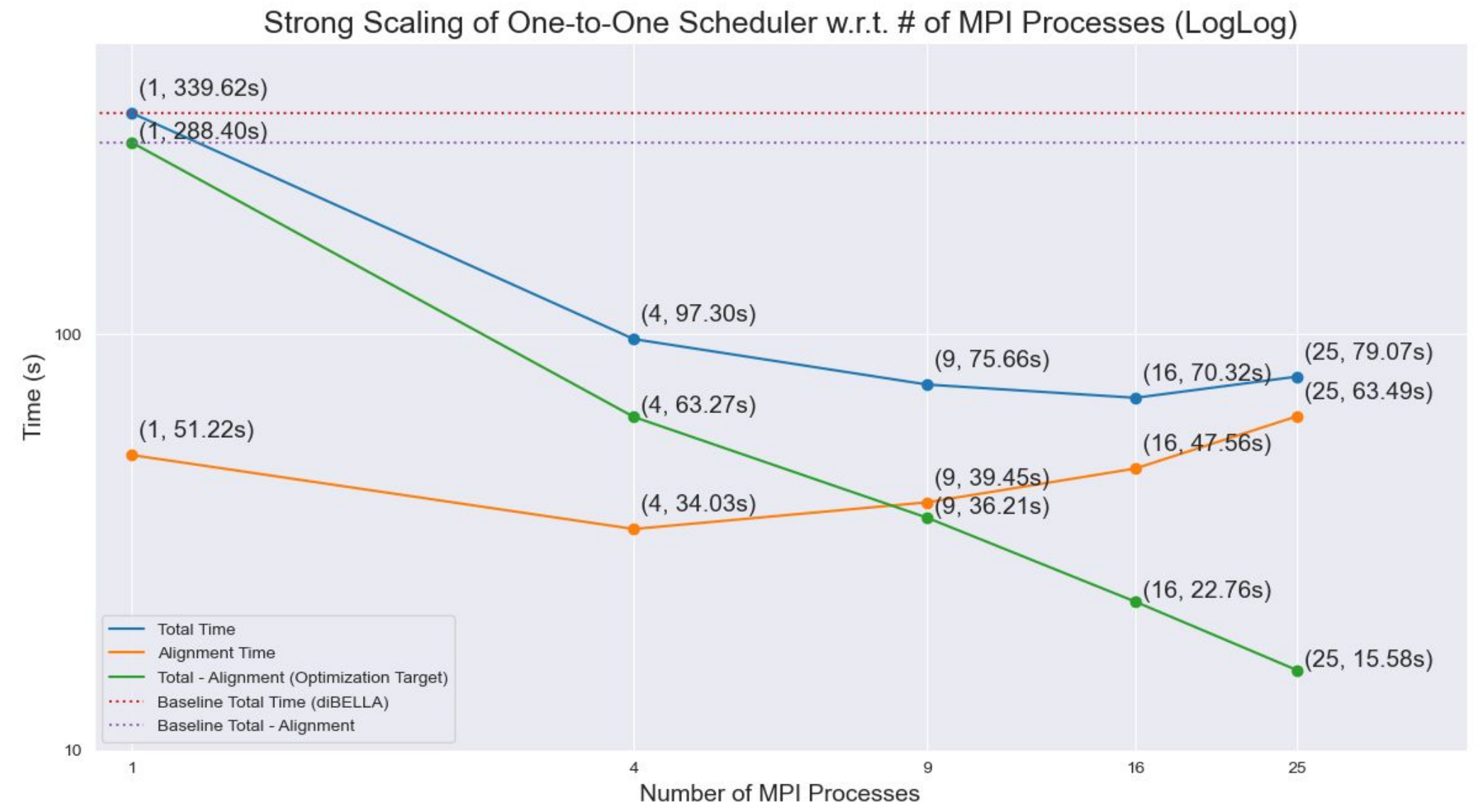
1. Strong-Scaling Experiment of method 1



Alignment time increases as there are more MPI processes, due to the increase of communication overhead. The overall time generally scales down, but increases when the MPI communication of the scheduler is too large.

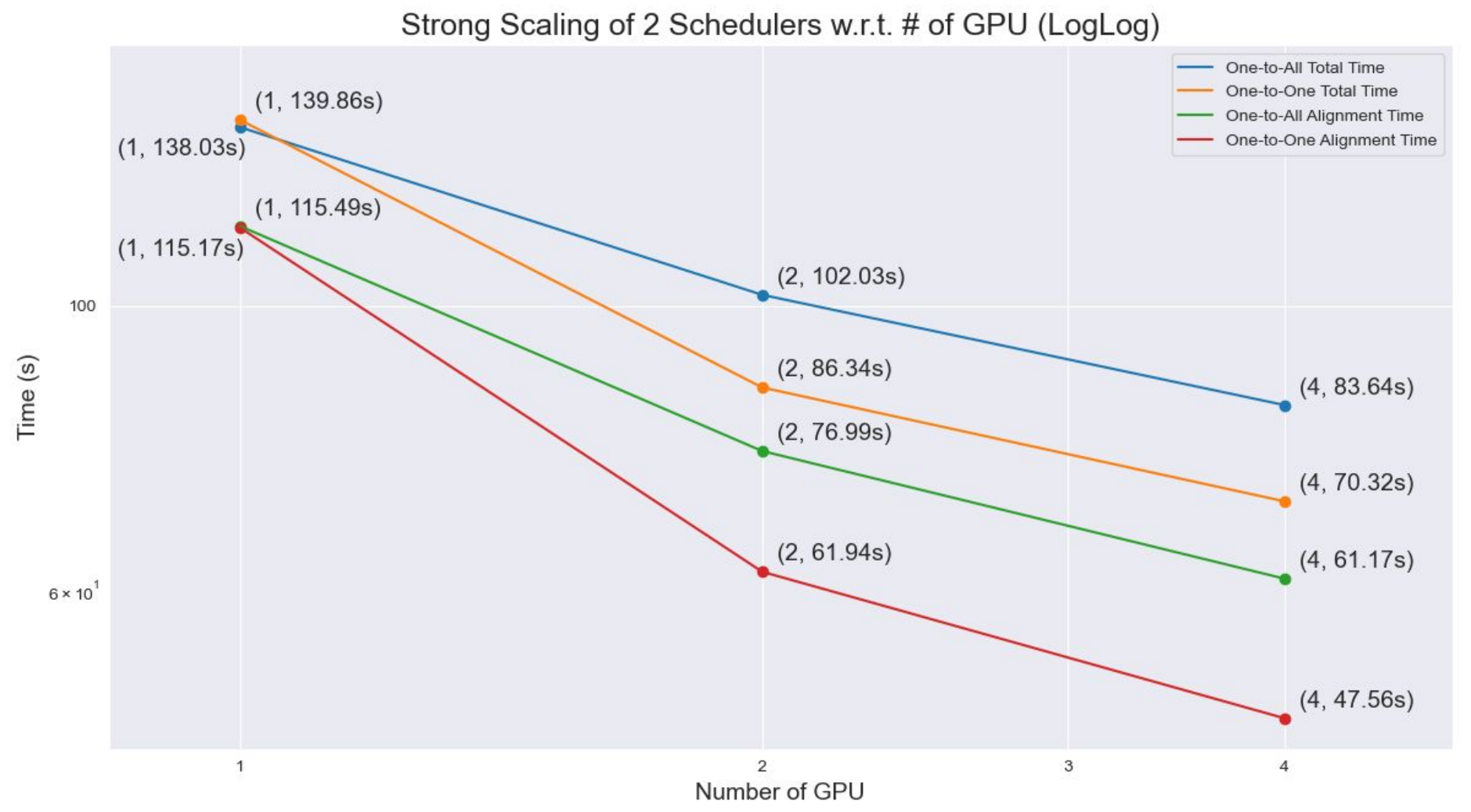
All experiments are conducted using 4 Nvidia A100 GPU.

2. Strong-Scaling Experiment of method 2



The communication overhead leads to an increase in alignment phase. The whole algorithm benefits from multiple MPI processes and has reduced overall time.

3. GPU Scaling Experiment



All experiments are conducted using 16 MPI processes.

Alignment time of method 2 is smaller than method 1 because MPI processes are divided into different pipelines. Both overall time and alignment time decreases as # GPU increases because each alignment iteration becomes faster.

Conclusion

We successfully implement two GPU schedulers which can significantly boost *De Novo* Genome Assembly given multiple MPI processes and GPUs. The strong scaling of overall runtime is high for # MPI processes less than 16. The communication overhead of GPU scheduler increases heavily as # MPI processes increases, which shows that our scheduler can be further improved by reducing the messages.

In the future we will implement a scheduler which releases GPU on completion and work on reducing MPI communication.

Acknowledgements

We are grateful to Professor Giulia Guidi for her tutorial on the background of ELBA and guidance on implementing this project.