

Taichi-based GPU Accelerated Evolutionary Algorithm

GPU Accelerated Programming

Ali Asghar Chakera
Mustafa Sohail

LaunchPresentationKernel<<<1,12>>>



Table of Contents

- Introduction
- Motivation
- Relevant Work
- Novelty
- Timeline
- Work Division

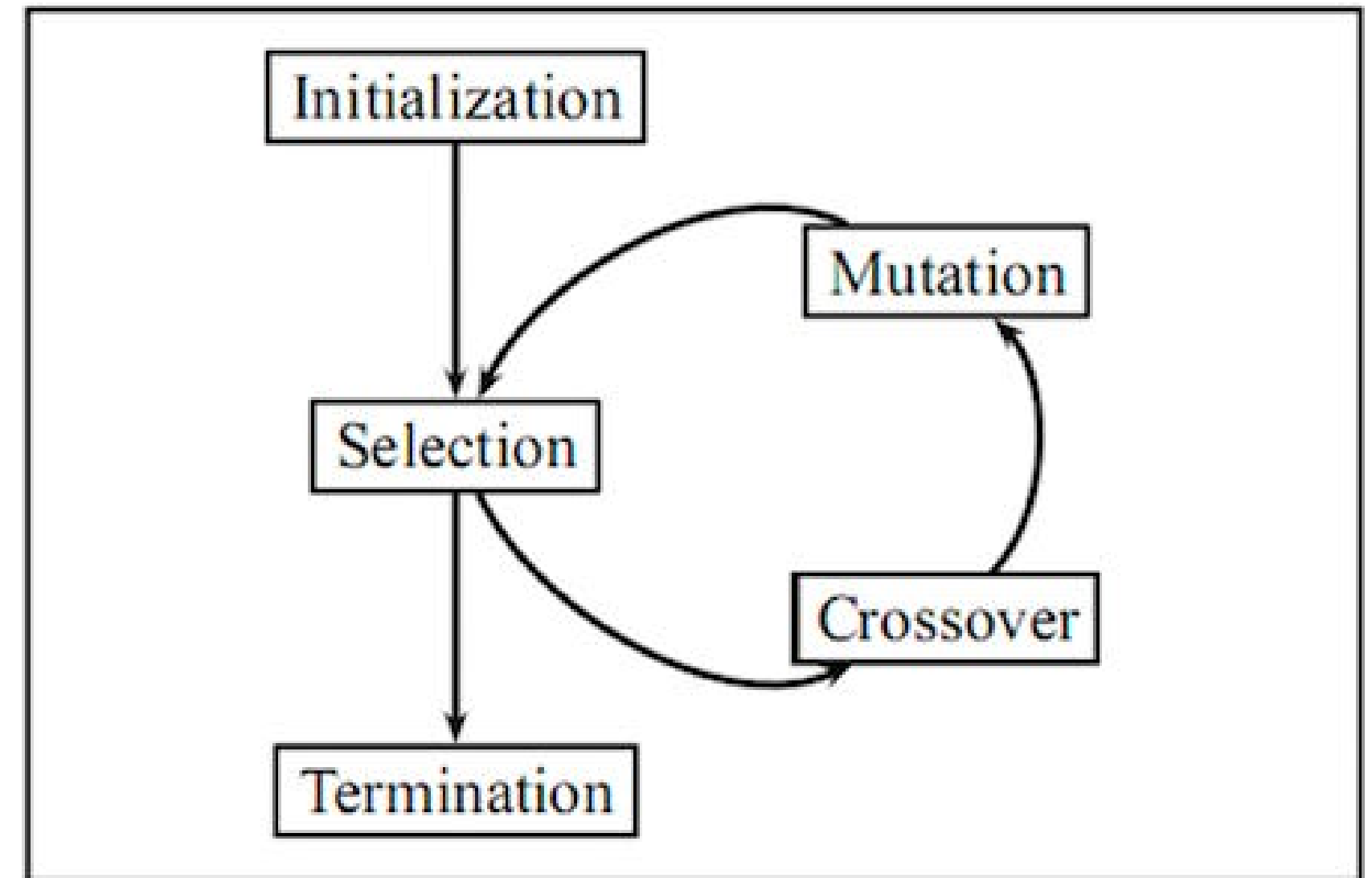
t_idx = 1



Introduction

What are Evolutionary Algorithms (EA) ?

- Based on biological evolution formula.
- Used for optimization problems.
- Solution space is initialized with multiple solutions.
- Evolution is done based on 'fitness' values of individual solutions.
- These fitness functions can be costly based on the problem statement.



t_idx = 2



Introduction

Okay now what is Taichi?

- Open-source, parallel programming language
- Embedded in Python.
- Integrated with major GPUs and GPU APIs like CUDA.
- You don't need to write CUDA!



t_idx = 3



Introduction

Okay now what is Taichi?

- Open-source, parallel programming language
- Embedded in Python.
- Integrated with major GPUs and GPU APIs like CUDA.
- You don't need to write CUDA!



t_idx = 4



Introduction

```
@ti.kernel
def fill():
    total = 0
    for i in range(10): # Parallelized
        for j in range(5): # Serialized in each parallel thread
            total += i * j

    if total > 10:
        for k in range(5): # Not parallelized because it is not at the outermost scope
```

t_idx = 5



Our Goal

- Utilize the massive potential in parallelizing evolutionary algorithms
- Write a framework for EAs in taichi language.
- Framework can be used by anyone planning to write evolutionary algorithms in python.

t_idx = 6



Motivation

Leveraging new technology

Building on a relatively new innovation for future users.

Diverse Ecosystems

Since Taichi is built in Python, this can be used in different environments e.g PyTorch

Intuitive Syntax

Python is already very user-friendly and it can eliminate the learning curve of C++.

Accelerating optimization problems

Evolutionary algorithms can be costly if initial population size or number of generations are high.

t_idx = 7



Relevant Work



Evolutionary Algorithms frameworks available in different languages:

- PyGAD
- Open Beagle (C++)
- EvoX (GPU optimisation)



Evolutionary Algorithms implementation in Cuda:

- M. Oiso, Y. Matsumura, T. Yasuda, K. Ohkura. Implementing Genetic Algorithms to Cuda using Data Parallelization
- P. Valkovič. GPU Parallelization of Evolutionary Algorithms
- Asafuddoula et al. A CUDA implementation of an improved decomposition based evolutionary algorithm for multi-objective optimization

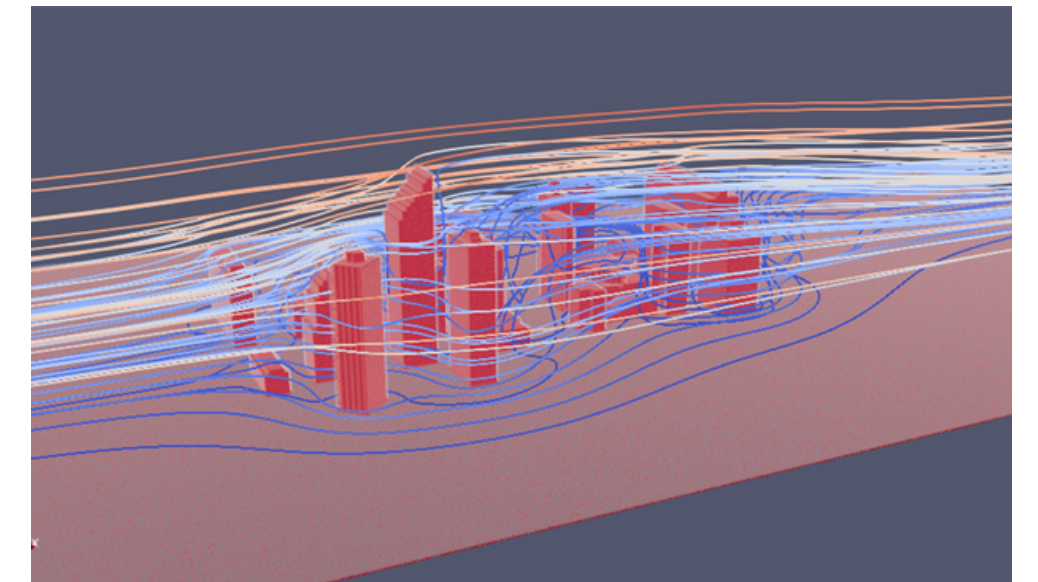
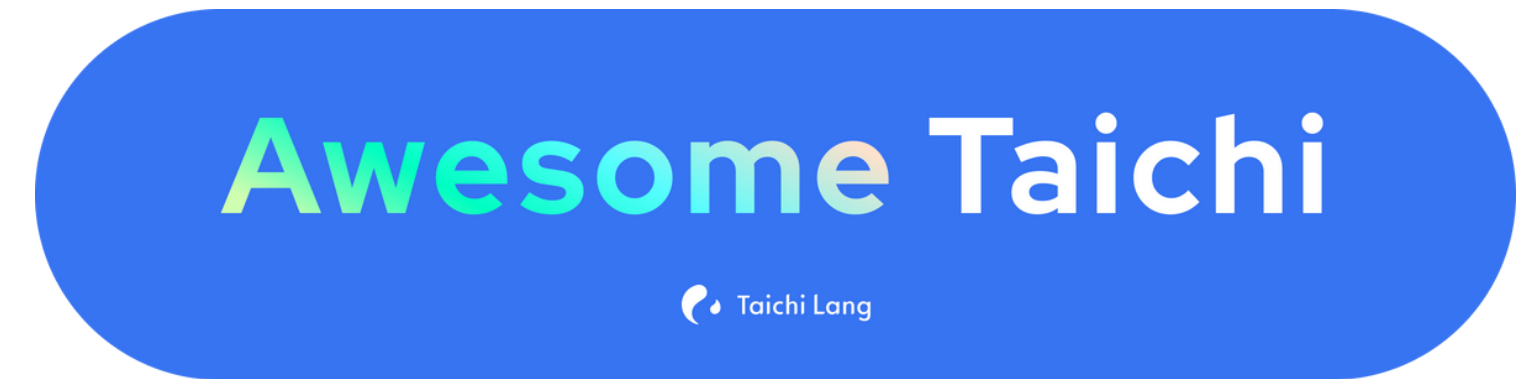
t_idx = 8



Relevant Work

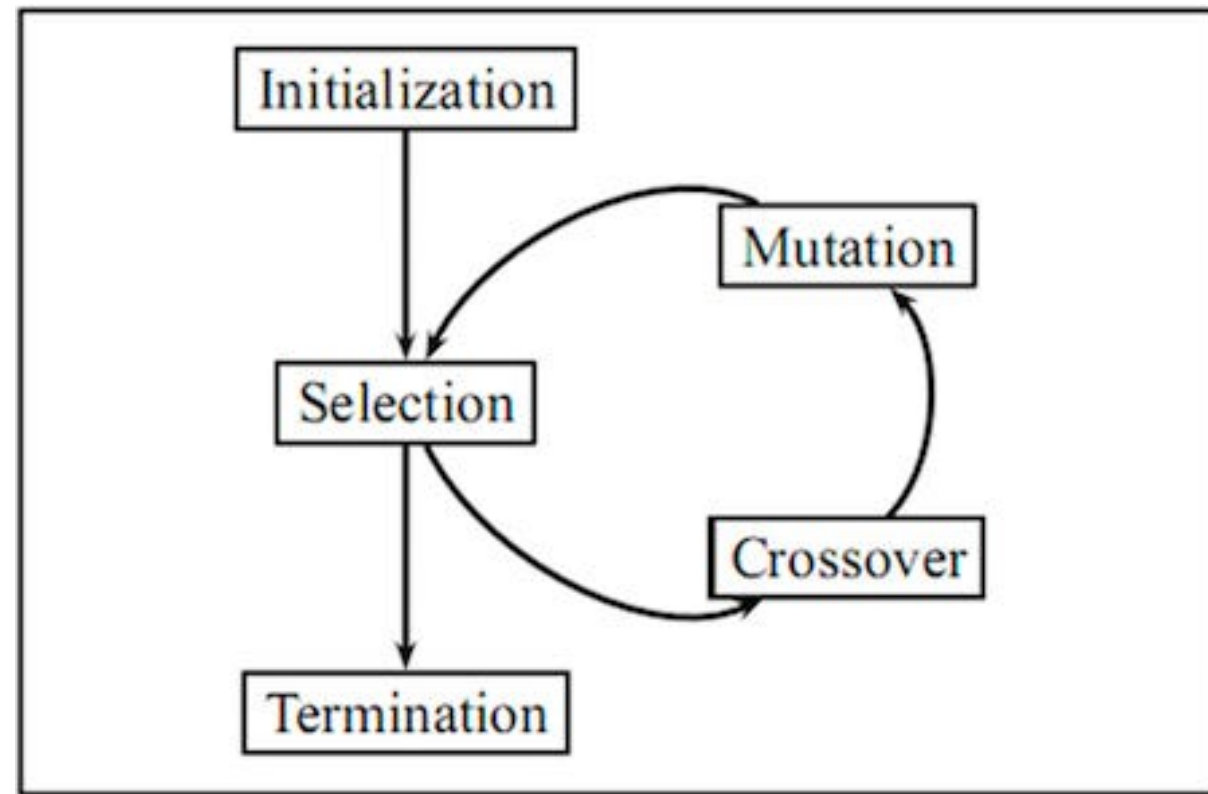
Work done on Taichi:

- Taichi has been prominently used in:
 - Rendering graphics
 - Numerical Simulations
 - General Purpose Speeding up of Python code
 - Image Processing
 - More [here](#)
- [Differential Evolution](#)
- [Ant Colony Simulation](#)



t_idx = 9





Novelty

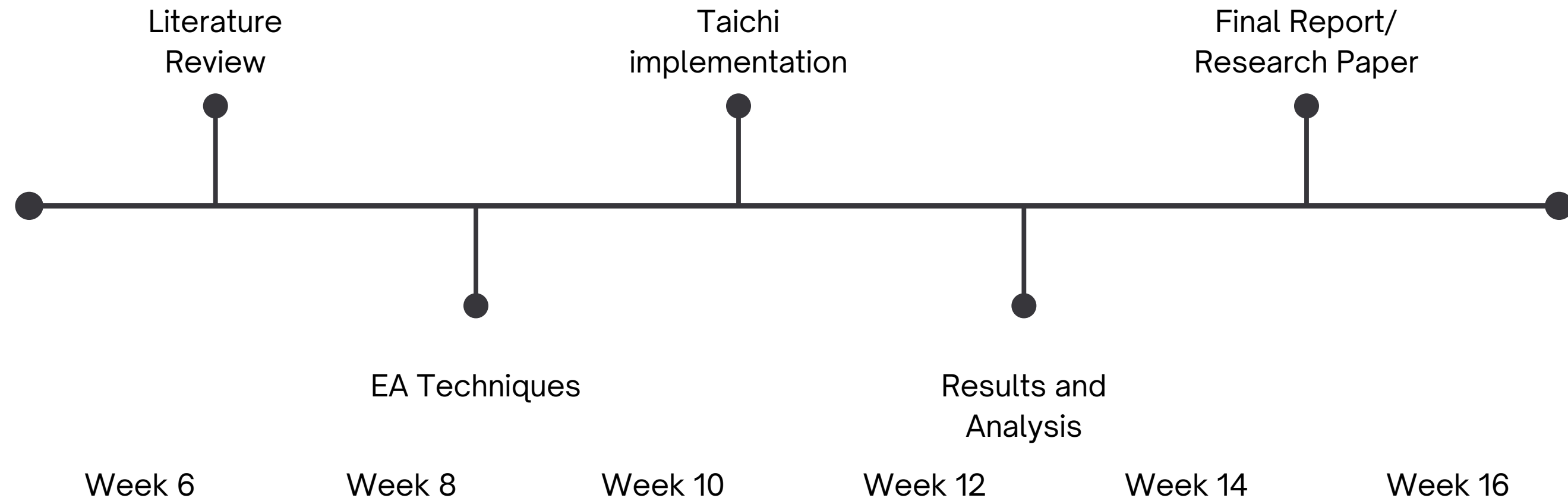


- Providing a framework written in Taichi to use while writing EAs.
- No published research in evolutionary algorithms domain in Taichi

t_idx = 10



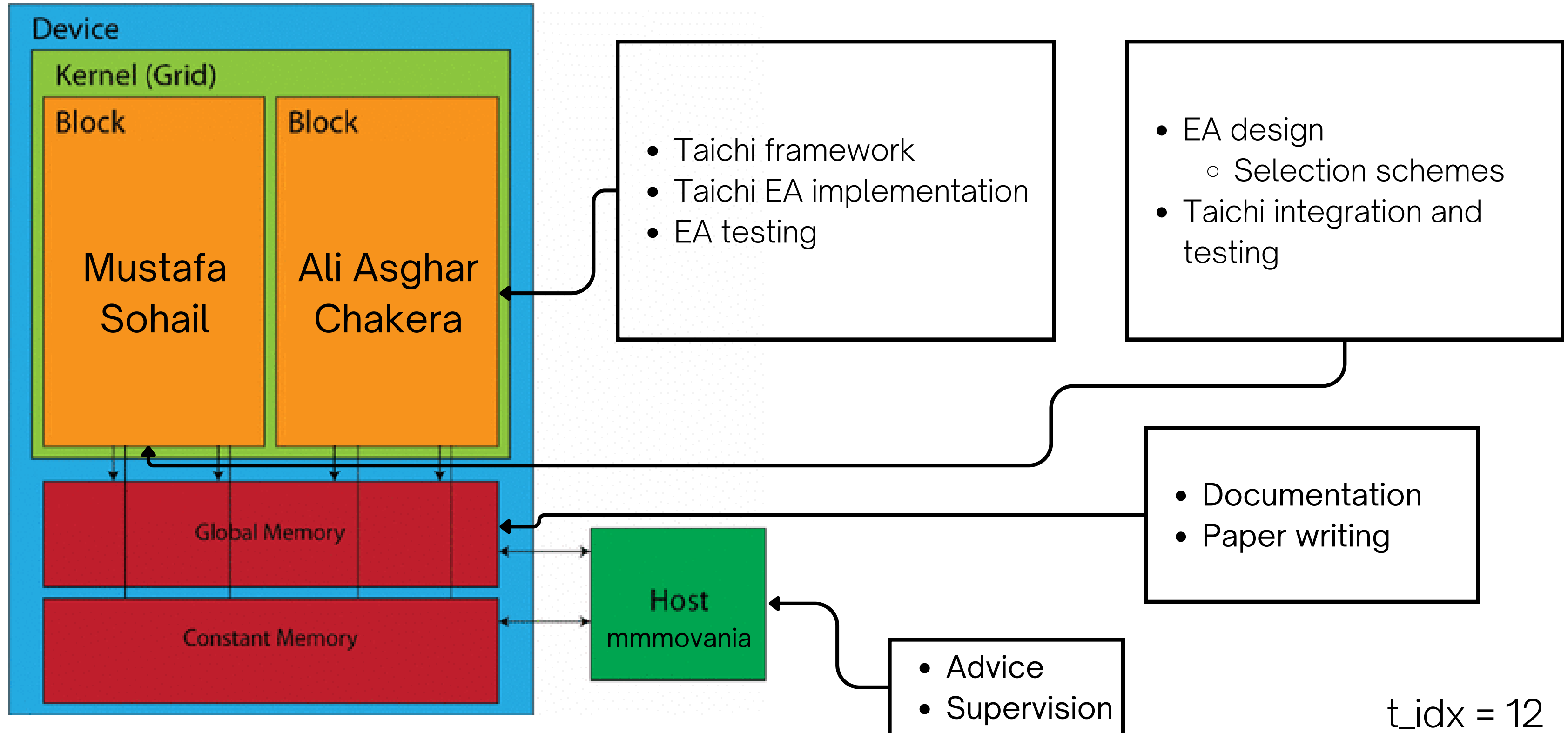
Timeline



t_idx = 11



Work Division



t_idx = 12



References

- Chen-Erqi. “Chen-Erqi/de-on-Taichi: Differential Evolution on Taichi and Cuda.” GitHub, github.com/chen-erqi/DE-on-taichi. Accessed 15 Feb. 2024.
- “Taichi Lang: High-Performance Parallel Programming in Python.” Taichi Lang: High-Performance Parallel Programming in Python, www.taichi-lang.org/. Accessed 15 Feb. 2024.
- Sumida, Brian H., et al. "Genetic algorithms and evolution." Journal of Theoretical Biology 147.1 (1990): 59-84.
- Hu, Yuanming, et al. "Taichi: A Language for High-Performance Computation on Spatially Sparse Data Structures." ACM Transactions on Graphics (TOG), vol. 38, no. 6, 2019, pp. 201.
- Valkovič, Patrik. GPU Parallelization of Evolutionary Algorithms, Charles University Digital Repository, 2021
- Asafuddoula, Md, et al. “A CUDA implementation of an improved decomposition based evolutionary algorithm for multi-objective optimization.” Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, 20 July 2016, <https://doi.org/10.1145/2908961.2908971>.
- M. Oiso, Y. Matsumura, T. Yasuda, K. Ohkura. “Implementing Genetic Algorithms to Cuda using Data Parallelization.” Hrčak Croatian Journal, 2011

CudaDeviceSynchronize()

