GROUP 18

ACCELERATION OF DTW ALGORITHM FOR REAL-TIME NANOPORE SELECTIVE SEQUENCING USING GPUS

TEAM MEMBERS

E/16/089
Denuke Dissanayake

E/16/313
Maneesha Randeniya

E/16/360
Nipun Dewanarayana

SUPERVISORS

Prof. Roshan
Ragel

Dr. Hasindu
Gamaarachchi

OBJECTIVE

- What if, each time, the **data needs to be uploaded to a remote server** for DNA analysis?

- What if we can analyse the data from the sampling place itself?

BACKGROUND

- **Nanopore sequencing** is a dna sequencing technology that enables direct, real-time analysis of **DNA or RNA** fragments.
- Generate an electrical current as nucleic acids are passed through a protein nanopore.
- Modern nanopore sequencers offer **selective sequencing** capability.

BACKGROUND

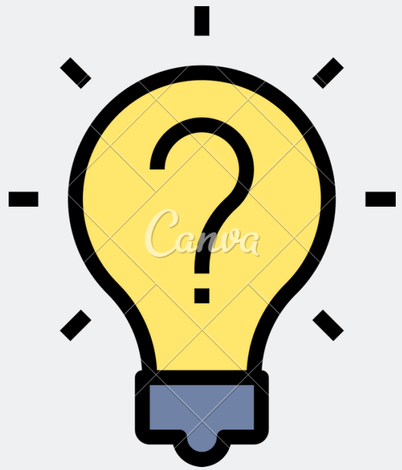
- **Dynamic Time Warping ALGORITHM**
  - measures the **optimal alignment** between signals
  - Time and Space complexity - **$O(n^2)$**
  - starts by building the **distance matrix.**
  - Next, **finds the alignment path** that runs through the cost matrix's low-cost areas

PROBLEM DEFINITION

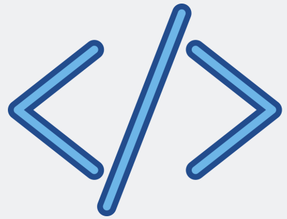
- DTW algorithm has a **High Computational Demand**.
- So portable MinION sequencers **must connect to a large server** to do the analyses.

SOLUTION

- **Implement** the DTW algorithm using GPUs
- **Optimise** the DTW algorithm to Reduce the runtime



EXISTING CODE



- All calculations are done in the **CPU**
- Calculations are done in **sequentially**
- CPU DTW run time
  - Intel i7 10th Gen 16GB RAM - **~27 sec**
  - Kepler Workstation (CE Department) - **~23 sec**

MILESTONES

- **Identify** the places that need to be **parallelized**.
- Implement a **parallel** mechanism using **CUDA**
- Apply **Optimization** techniques to the algorithm

CHALLENGES IN PHASE 1 & 2

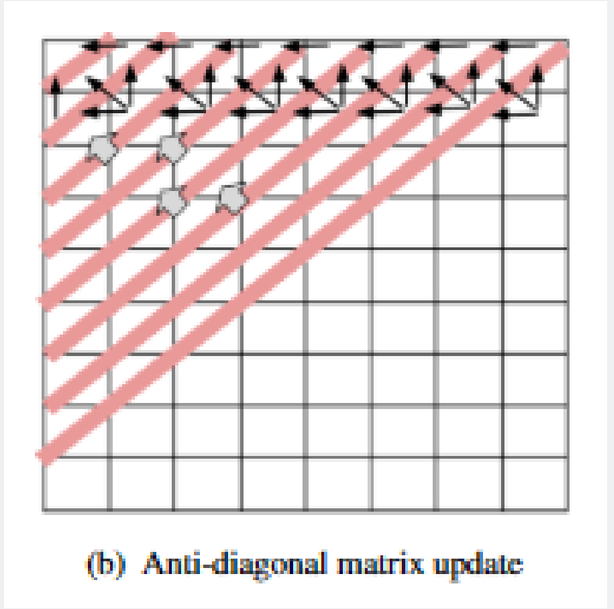
- **Complex** data structures
- **Defining sizes** for the CUDA pointers
- Illegal memory access **errors and segmentation fault** errors
- **Higher time consumption** to debug the code
- NVIDIA **driver issues**

CHALLENGES IN PHASE 3

- **Not Enough Memory** to allocate for CUDA pointers
- **Execution Time Varies** with the devices
- Block and thread counts & size of memory vary with devices
- Calculating the cost matrix **depends on previous values**
- When no of threads per block increase, runtime also increase
- Invoking kernels inside a kernel does not increase performance

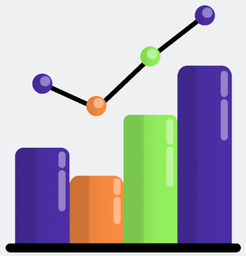
ONGOING PROCESS

- **Diagonal Method to calculate** the DTW algorithm
- Use Shared Memory
- Invoke kernel inside the kernel

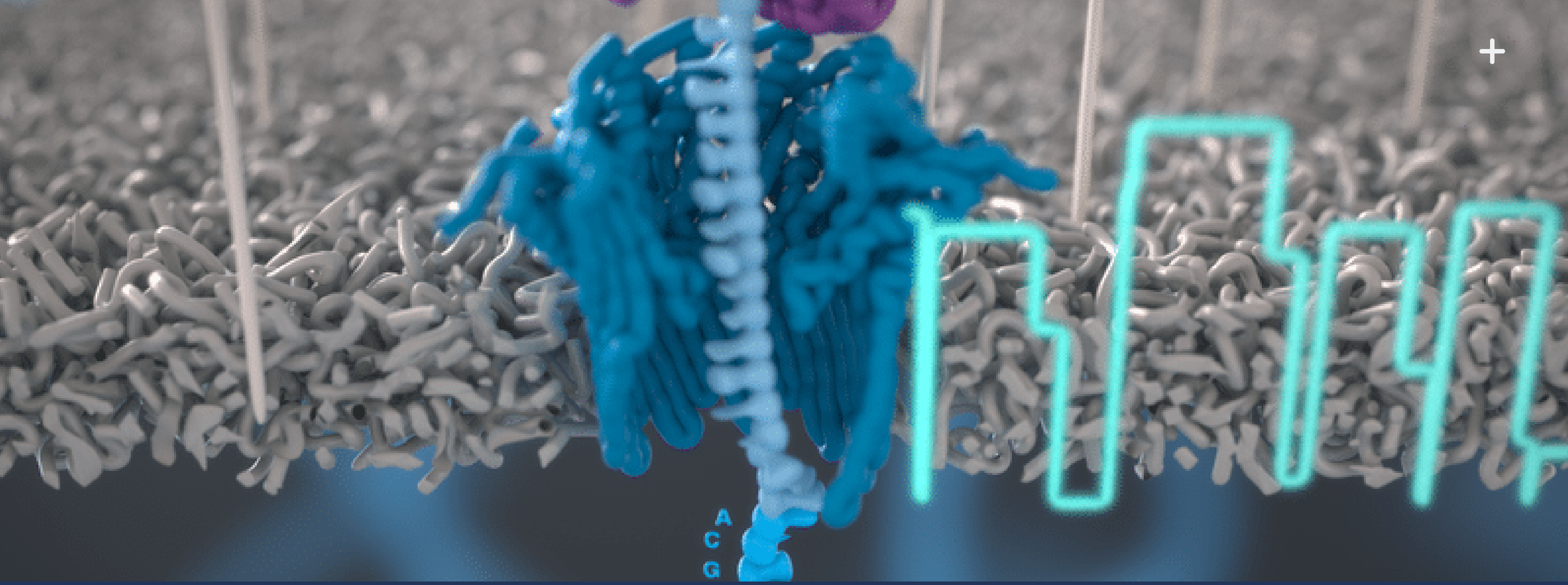


(b) Anti-diagonal matrix update

RESULTS



	NVIDIA GEFORCE MX 330	Kepler Server
CPU code	30 s	23 s
GPU code	258 s	991 s
Optimization 1	421 s	228 s
Optimization 2	190 s	158 s
Optimization 3	341 s	252 s
Optimization 3	Pending	Pending

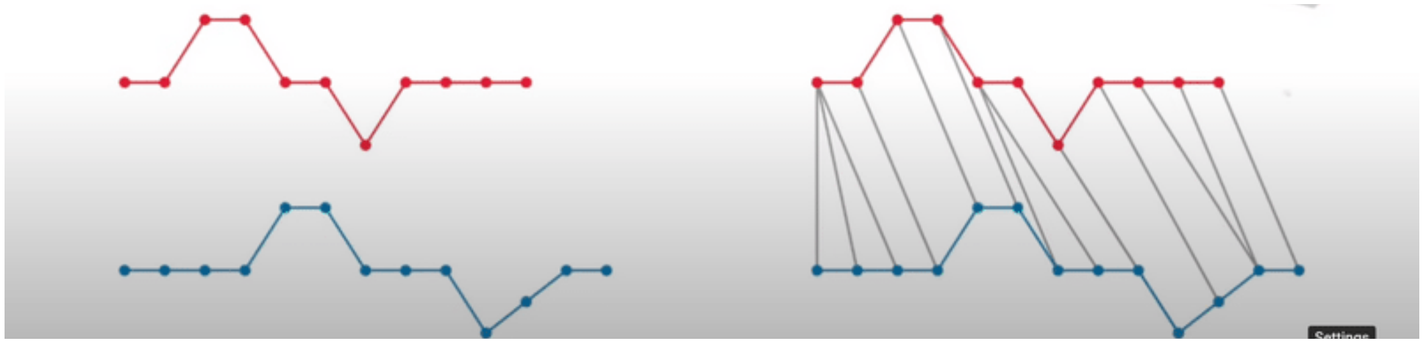


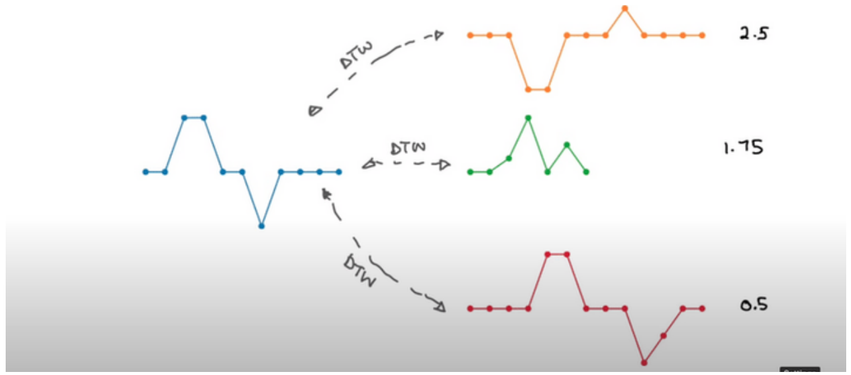
THANK YOU

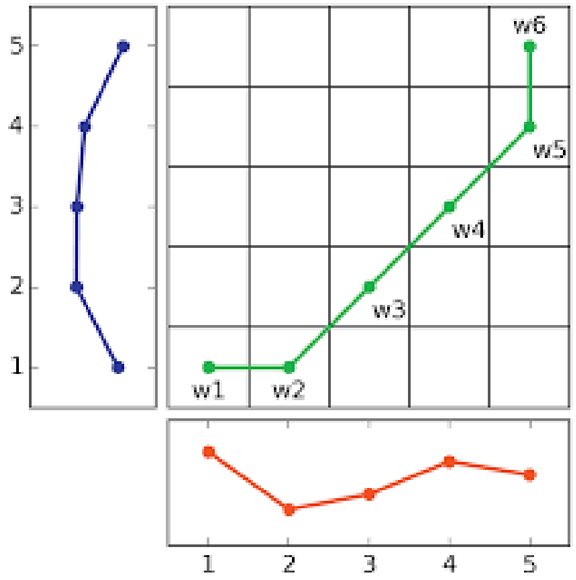
DTW ALGORITHM

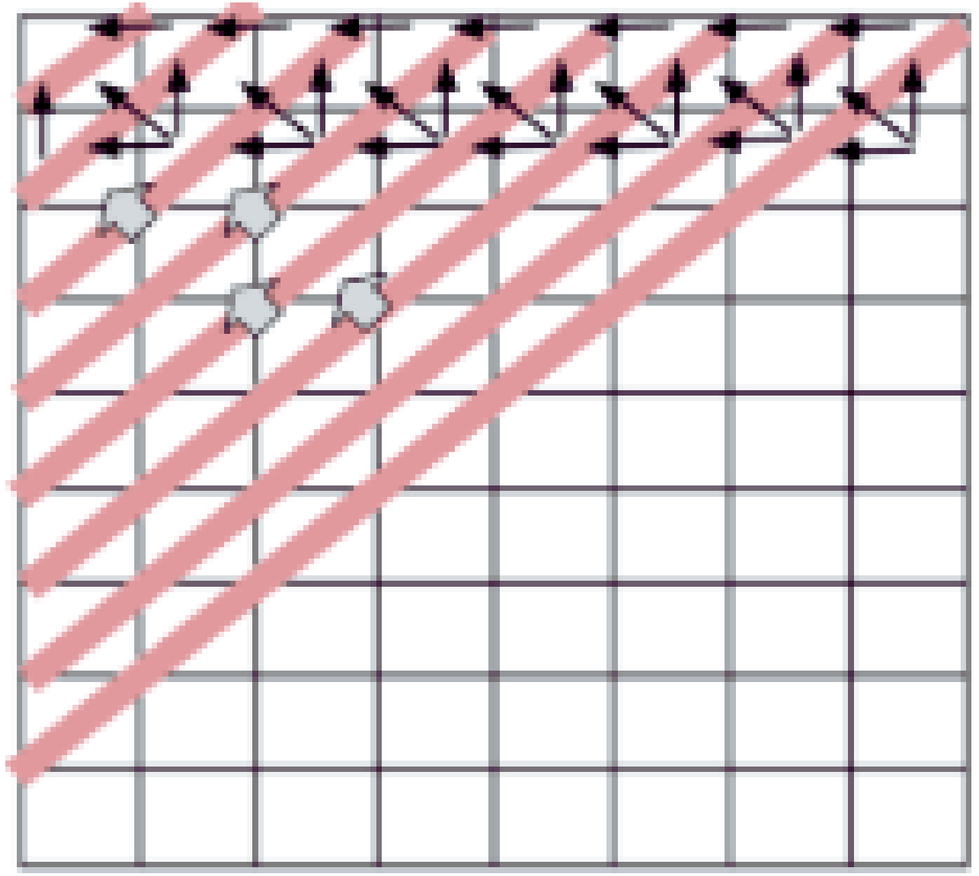
Align two Signals

- Which point on one signal corresponds to which point of the second signal
- How similar two signals are









(b) Anti-diagonal matrix update