

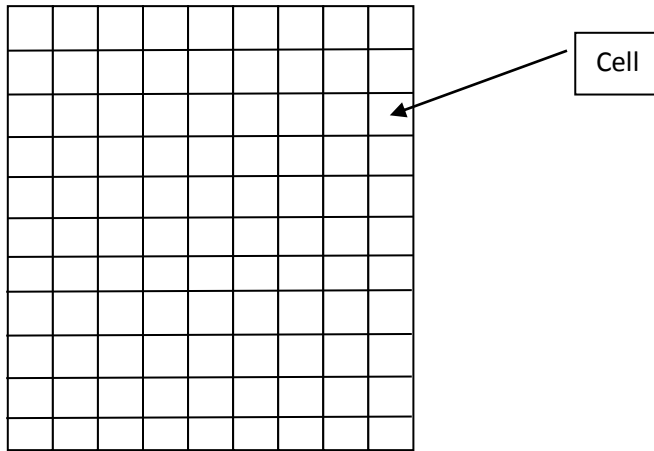
“Parallel and distributed computations” – course 10234

Final Project, summer 2012

Definitions and Concepts

Project Space – a two dimensional rectangular area of known size $A \times B$.

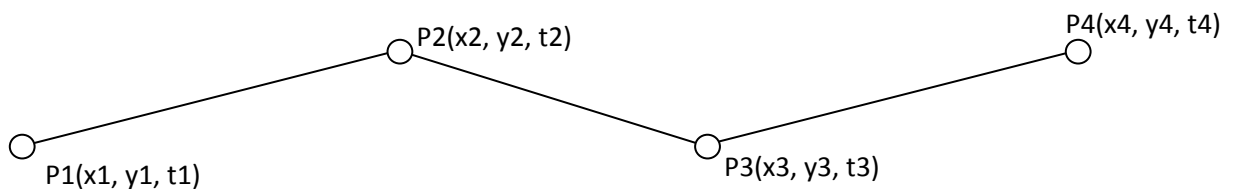
Cell in Project Space – a space unit with the Project Space is divided by m, n in each dimension.



Control Point – certain point in the Project Space with attached timestamp. It is used for the Flight Path definition. The timestamp defines the moment when this Control Point has to be passed.

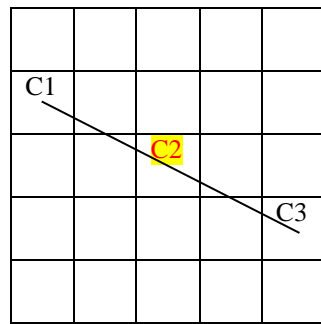
Flight Leg – a part of the Flight Path between two Control Points.

Flight Path – ordered set of Control Points. The velocity on each Flight Leg is constant.



Occupied Cell – the cell with at least one object at the given moment.

Critical Situation – there are objects that are unseen by at least one other object. Let's assume that object O1 occupies cell C1, O2 occupies cell C2 and O2 occupies cell C2, P1 is a center point of the cell C1 and P3 is a center point of C3. We will say that O2 hides C1 from C3 (and C3 from C1) at the given moment if the line P1P3 crosses the cell C2. If the line P1P3 passes through the corner of the cell C2 it is also defined as hiding. Two or more objects in the same cell do not create a Critical Situation with each other, but are taken into account for counting of Critical Level.



Critical Level (CL) of the object at the specific moment – the number of objects that do not see this object at the given moment. The CL of the cell C2 is equal to zero, CL of both C1 and C3 are equal to one.

Critical Degree – the sum of all Critical Levels of the object during its Flight (calculated for the given sets of moments – $T_{\text{start}}, T_{\text{star}}+dT, T_{\text{star}}+2dT, \dots, T_{\text{End.}}$).

The Project Goal

Write an effective parallel program to find an object with the maximum Critical Degree and display id of all other object that did not see it during the Flight.

Requirements and Constraints

1. Use a number of computers to produce an effective parallel program. You will be asked to demonstrate your Application in the lab with at least three computers.
2. All computers have to work as optimal as possible (**Load Balancing**).
3. The data file may be very big (according to the number of objects) and initially is accessible to one computer only.
4. The program has to be based on **MPI** and **OpenMP** to implement **Effective** and **Scalable** application. There must not more than one MPI process on each computer. MPI process is responsible mainly for communications between computers . Heavy computations are performed with OpenMP.
5. It is possible to prepare Offline some needed data.

Grades and Estimations

- 70 Points – for implementation of the program according to all requirements.
- 10 Points – for design documentation including proposals how to incorporate CUDA into the project.
- 5 Points – for code quality – modularity, comments,
- 10 Points – for effective implementation of Load Balancing.
- 5 Points – for the complexity estimation for time/memory consuming parts of the program.

Bonuses

- 10 Points – for correct implementation of GPGPU in your application.
- 5 Points – for your own proposed and implemented additional interesting and non-trivial feature (have to be approved by lecturer).

Delivery and Additional info – will be announced on the HighLearn site of the course.

All Flight Paths are defined in a single file with a following format:

- Flight Code.
- The number of Control Points in the Path.
- Two coordinates (in meters) of each Control Point and Timestamp (hh:mm:ss format).
- Data of the next Flight Path.

For example, two Flight Paths may be defined as follows:

102

4

5 6 10:00:00

57 82 12:30:20

100 65 12:50:00

130 70 13:00:00

101

3

100 20 10:00:00

200 300 11:10:00

250 350 12:00:00

4

Additional data in the header of the data file:

A , B – dimensions of the Project Space in **meters**.

m , n – the number of Cells in each dimensions.

dT – time interval for hiding checks in **seconds**.

בהצלחה