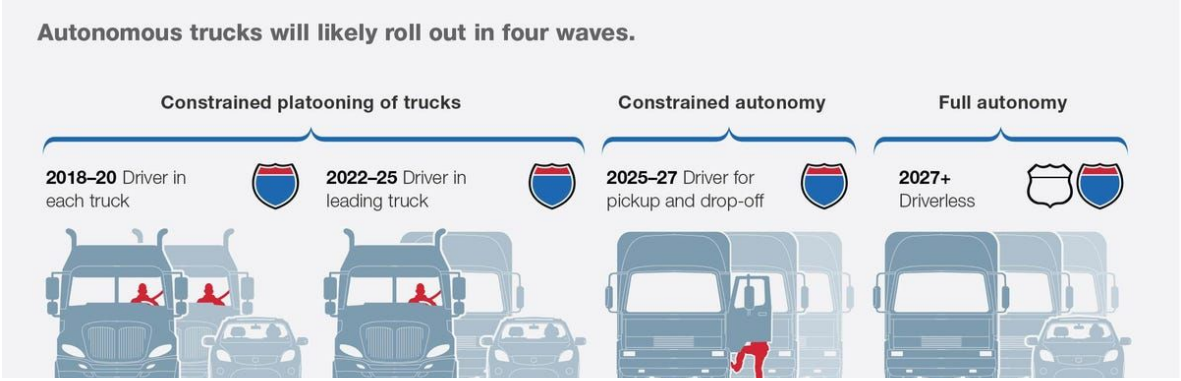
Distributed & Parallel Systems
In Truck Platooning

Presented by Team Unknown,
Omar Mammadov - 7213766
Namik Mert Tuncbilek - 7213712
Nijat Dashdamirov - 7213892
Vignesh Arumugam - 7213710

What really is Truck Platooning?



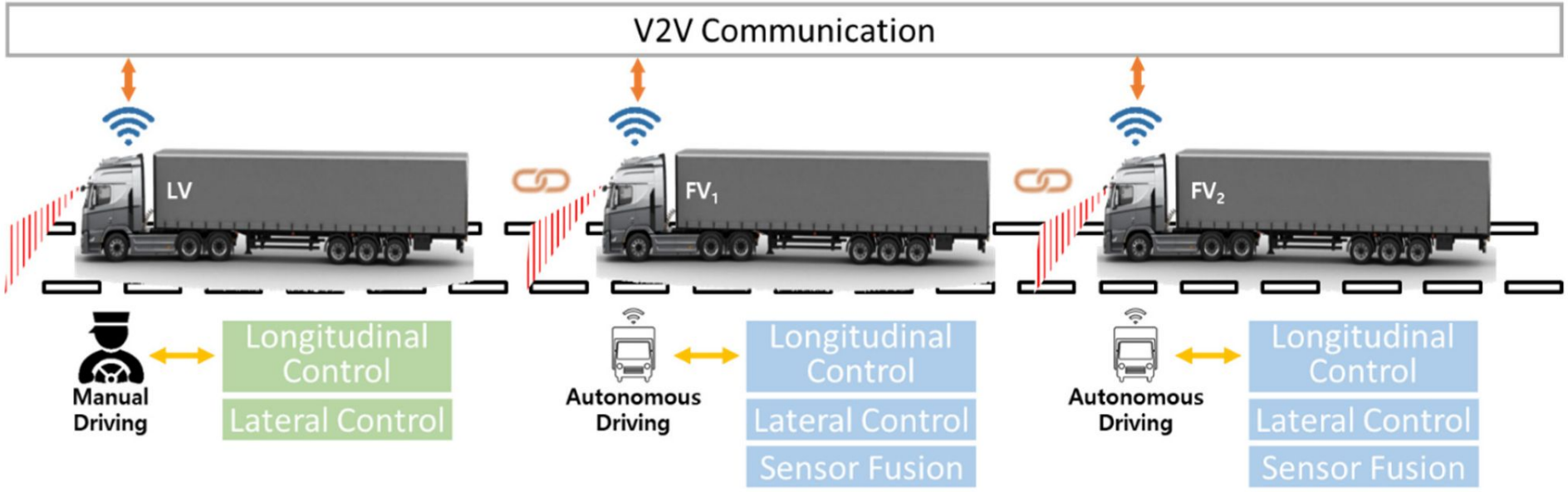
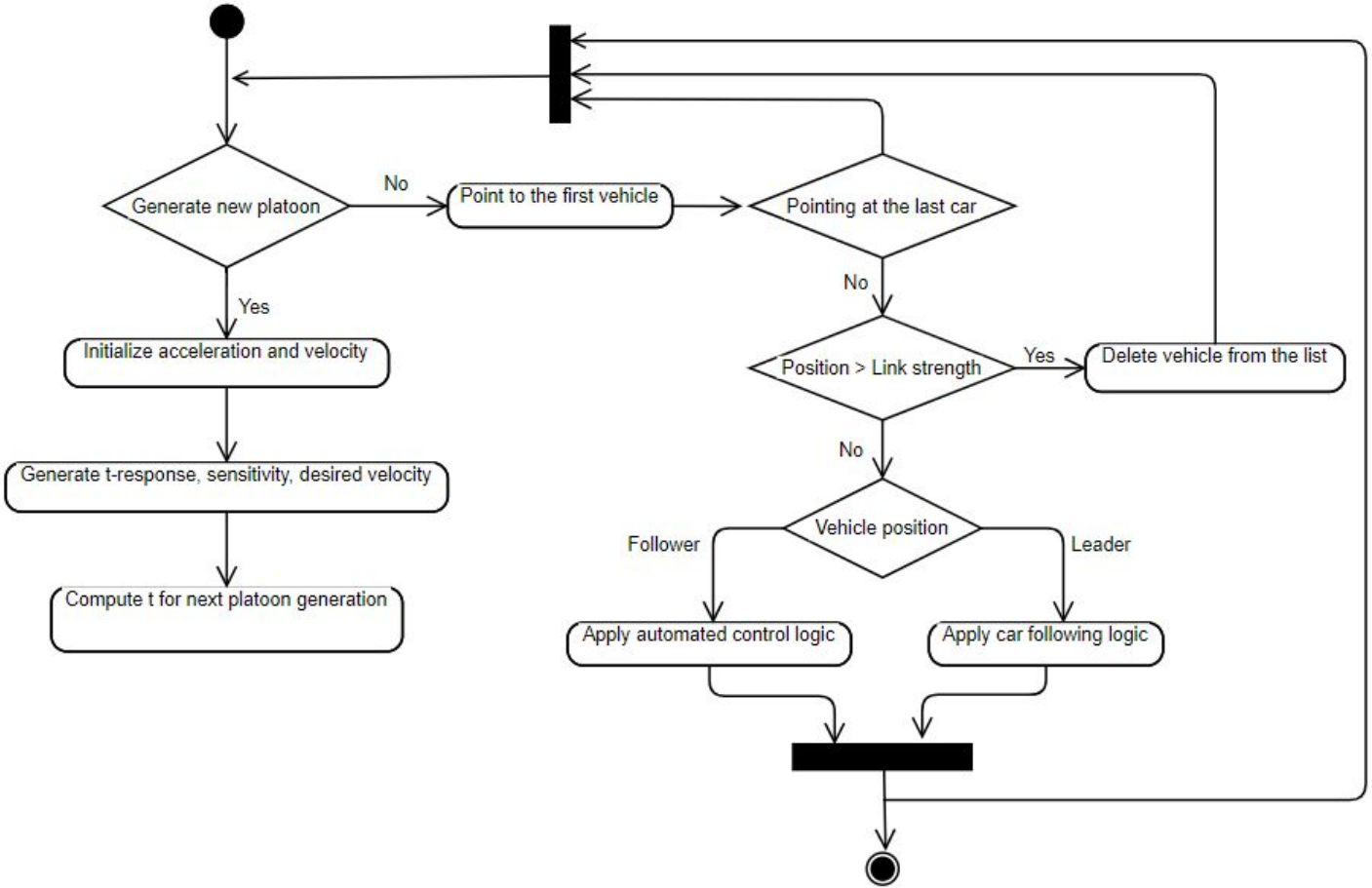


Figure 1. Overall architecture of truck platooning system.

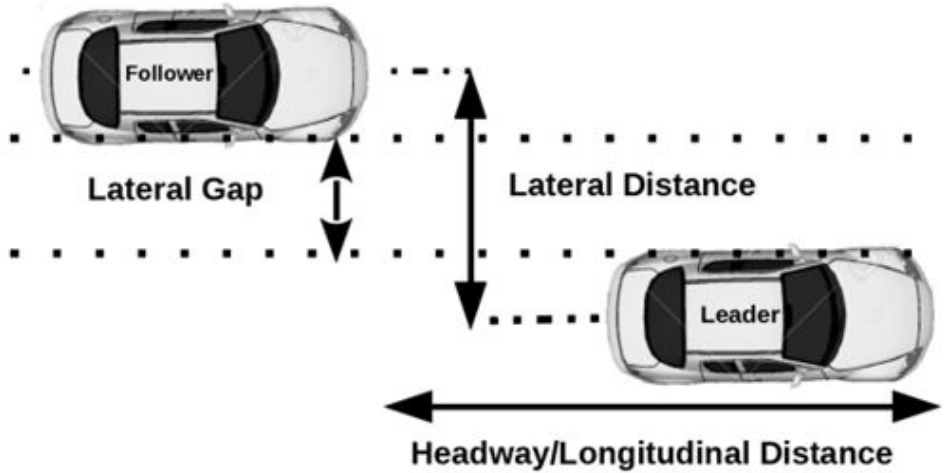
Communication / Interaction State Machine



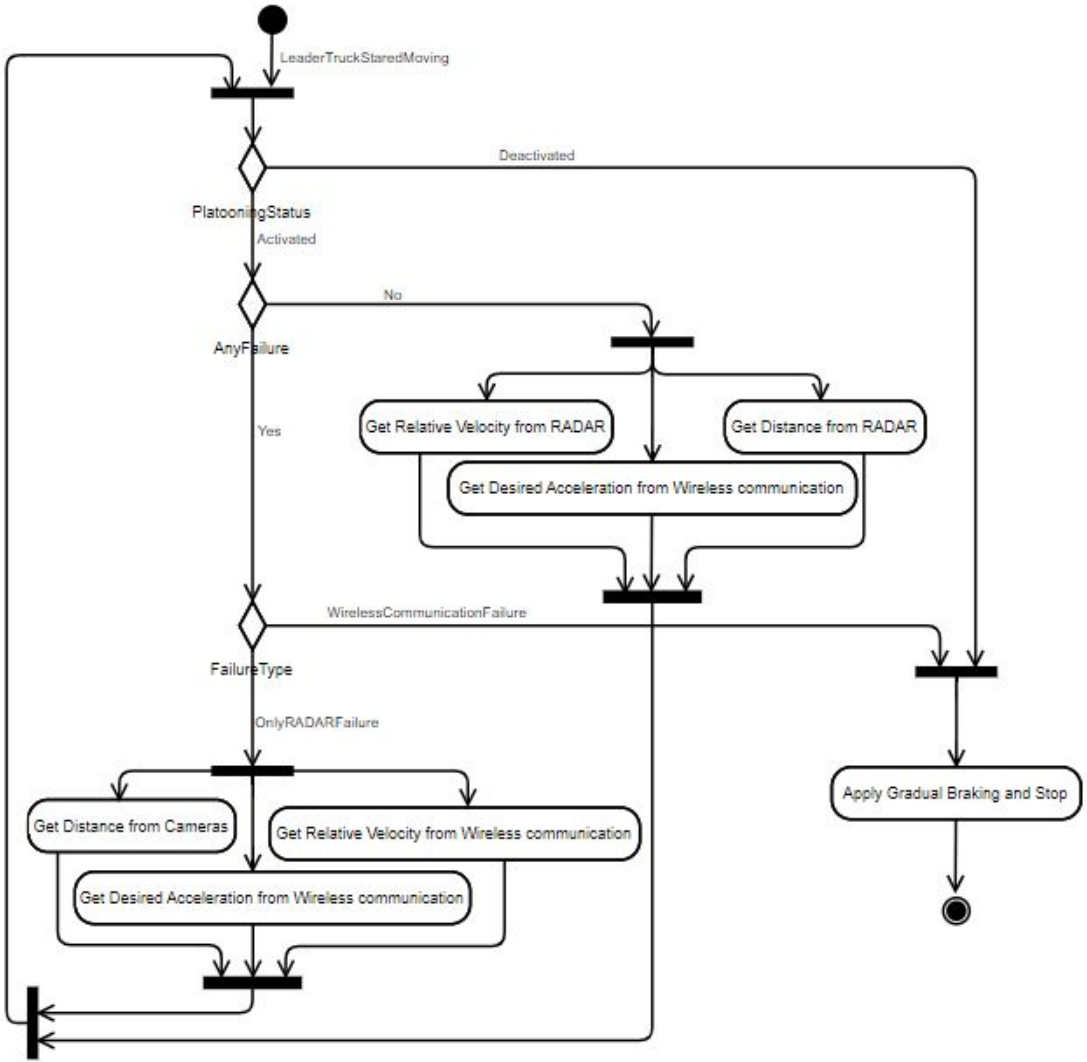
Control Behaviour

Distance to the precedence truck should be guaranteed.

The system should remain stable in cases of any failure.



Control Behaviour Activity Diagram



Pure Functional Behavior Implementation

The expected outputs of the program are x and y coordinates of each truck updating dynamically, after including the desired number of trucks, the distance between the trucks and the starting coordinates of the leader truck. The platoon may continue the drive with the same increment, or it may be stopped entering preset default (0 for our case).

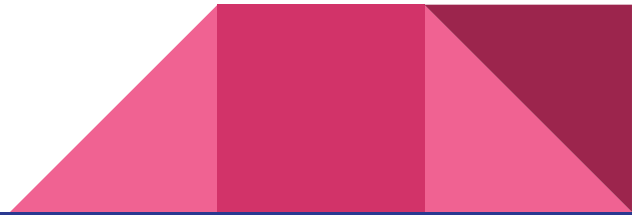
```
Please enter number of trucks: 3
Please enter number of distance: 5
Please enter start of x coordinate: 10
Please enter start of y coordinate: 10
x1:10-y1:10
x2:5-y2:5
x3:0-y3:0

x1:15-y1:15
x2:10-y2:10
x3:5-y3:5

x1:20-y1:20
x2:15-y2:15
x3:10-y3:10

x1:25-y1:25
x2:20-y2:20
x3:15-y3:15

x1:30-y1:30
x2:25-y2:25
x3:20-y3:20

run or stop
```

Architectural Model

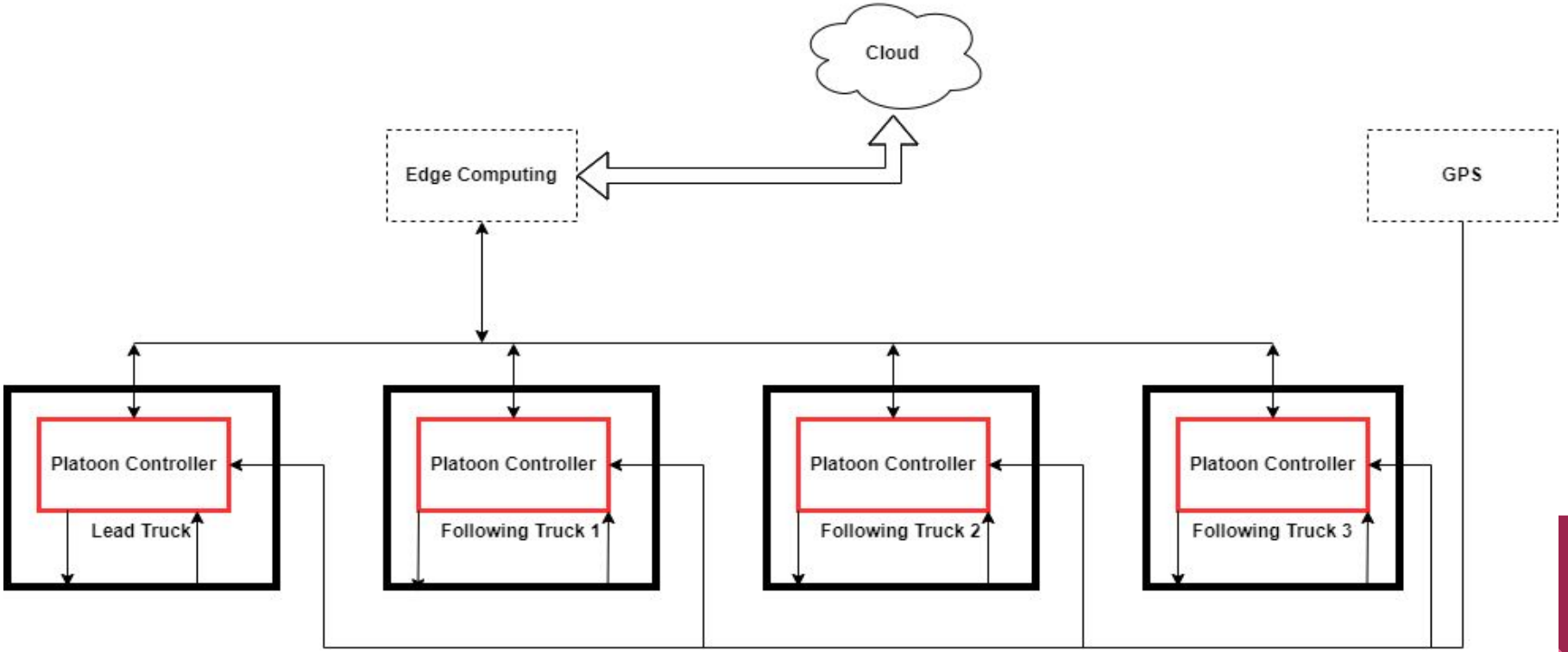
Important Elements of Distributed Architecture

- Entities of the Distributed network [Nodes - Truck]
- Communication paradigm [Remote Invocation - Message Passing]
- Roles & Responsibility [Client-Server]
- Pattern [Two-tier Layered (Fat-Client)]

And the Nodes - Truck is going to have parallel architecture to perform its tasks quickly.



Distributed Communication



Distributed Communication

The communication is about data sharing

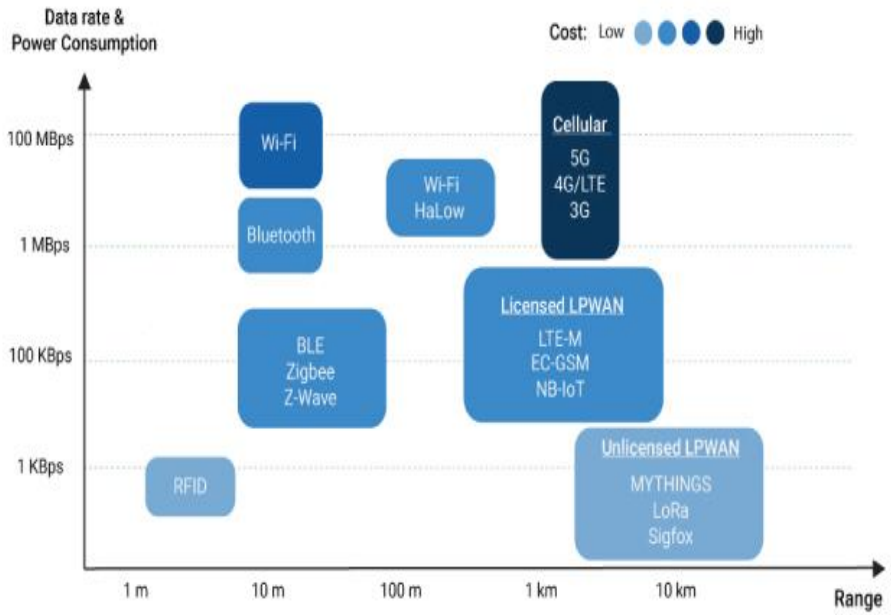
The data consist of GPS coordinates, Speed, Steering angle, Platoon Exit & Entry request, Truck number, Inter-distance, Platoon ID and other common information.

All these data are arranged in a Message frame [PL Frame] and it is been communicated between the nodes[Trucks] of the distributed system [Platooning] via Edge computing using wireless technology.

Cloud service is considered from AWS which provide exclusive service for Edge computing called Greengrass.

And for the Wireless technology we chose the one which suits better in range, efficient in data rate and reliable in connectivity is Cellular 5G/LTE/3G.

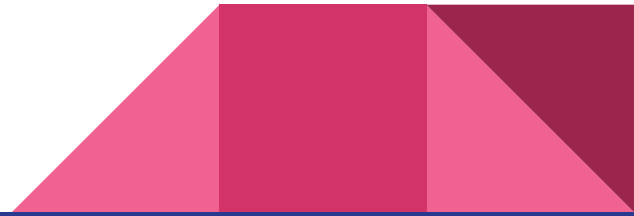
Distributed Communication



Distributed Interaction

Interaction is about, how one node is going to influence the other node in the distributed system.

Here in this platooning distributed system, the truck [Node] is going to interact with the another truck with the Data which is going to be communicated via Edge computing.



Distributed Interaction

As discussed earlier, the PL frame looks like the image as right.

The fields in the platoon data container influence the node's [Truck] action.

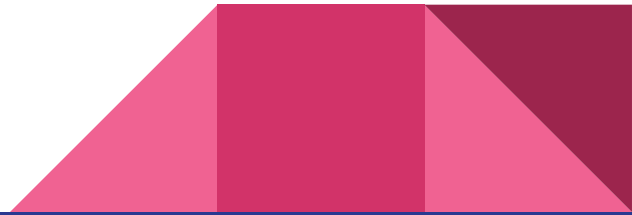
Container	Fields
Basic	ID (Sender)
	ID (Receiver)
	ID (Platoon)
	Position (Sender)
	Position (Receiver)
Platoon Data	Speed (Sender)
	Angle (Sender)
	Coordinates (Sender)
	Inter-Distance (Between Sender & Receiver measured from Sender rear Lidar sensor)
Insertion	Entry Request (Sender)
	Entry Permit (Receiver)
Disinsertion	Exit Request (Sender)
	Exit Permit (Receiver)

Parallel Programming

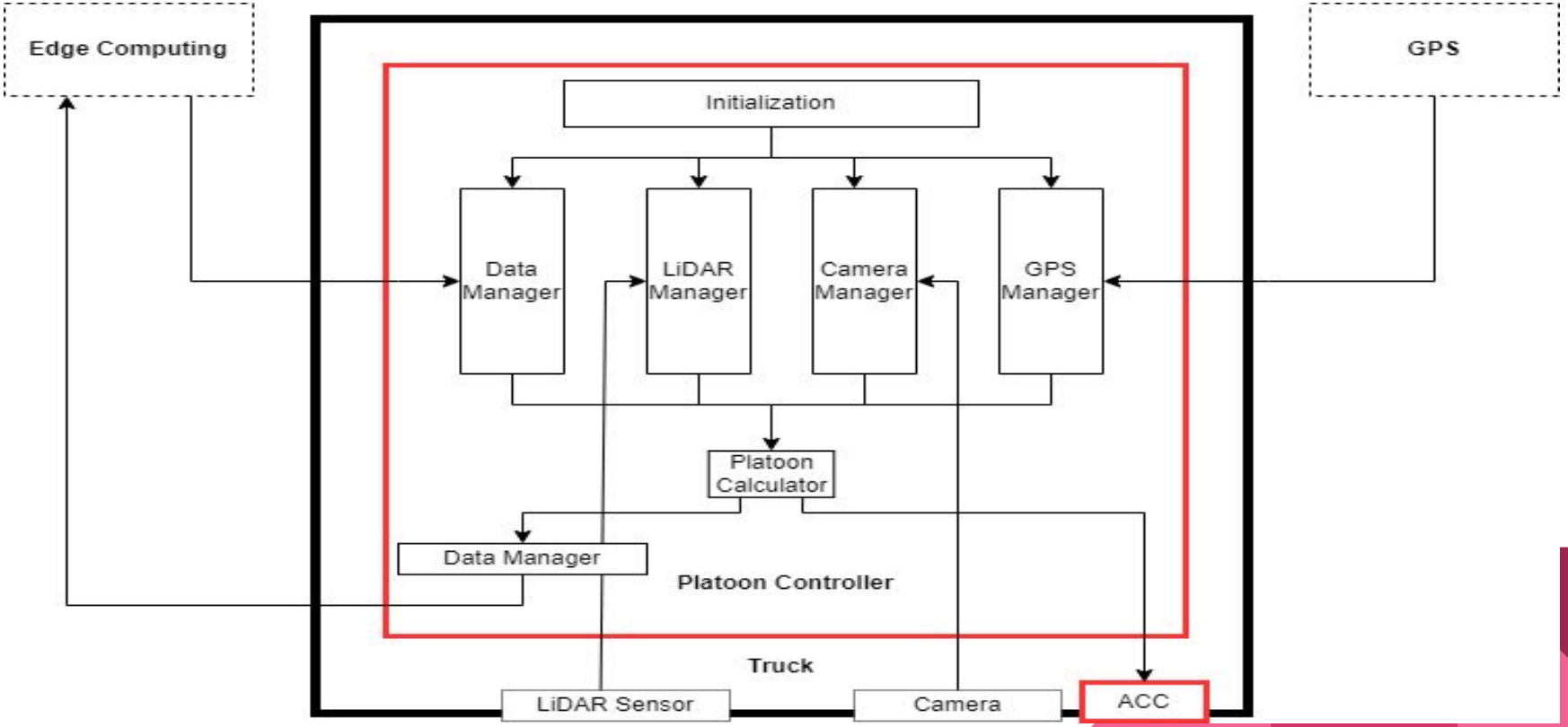
Task level parallelism has been designed for this project.

- Camera Manager - Get data from camera and check for any intruders
- LiDAR Manager - Get LiDAR sensor data and converts to required unit
- GPS Manager - Get coordinates from GPS system
- Data Manager - Collects Message data from Edge computing

All these four tasks are designed to run parallely. To achieve this corresponding parallel programming and hardware are chosen.



Parallel Programming



Hardware

As parallelism has been done as task level, it will be greater to have a multi-core ECUs.

The decision to go with Multicore is happened with performance improvement observed during the task scheduling.

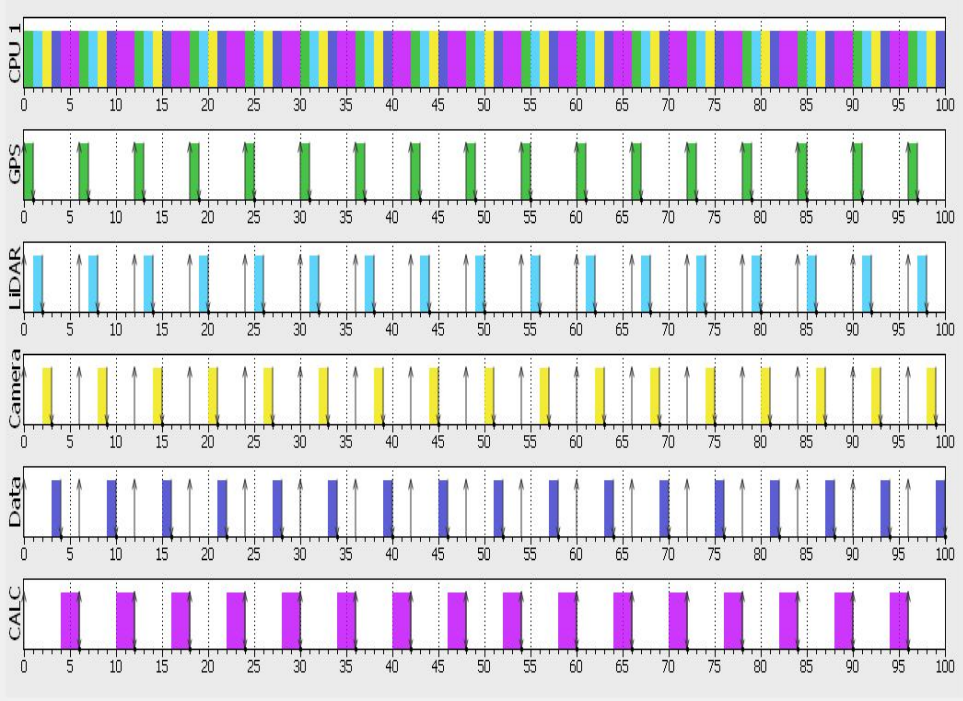
EDF method has been used for scheduling with different types of Hardware Single core vs Multi-core. And the difference seen is 100% percent increase in efficiency.

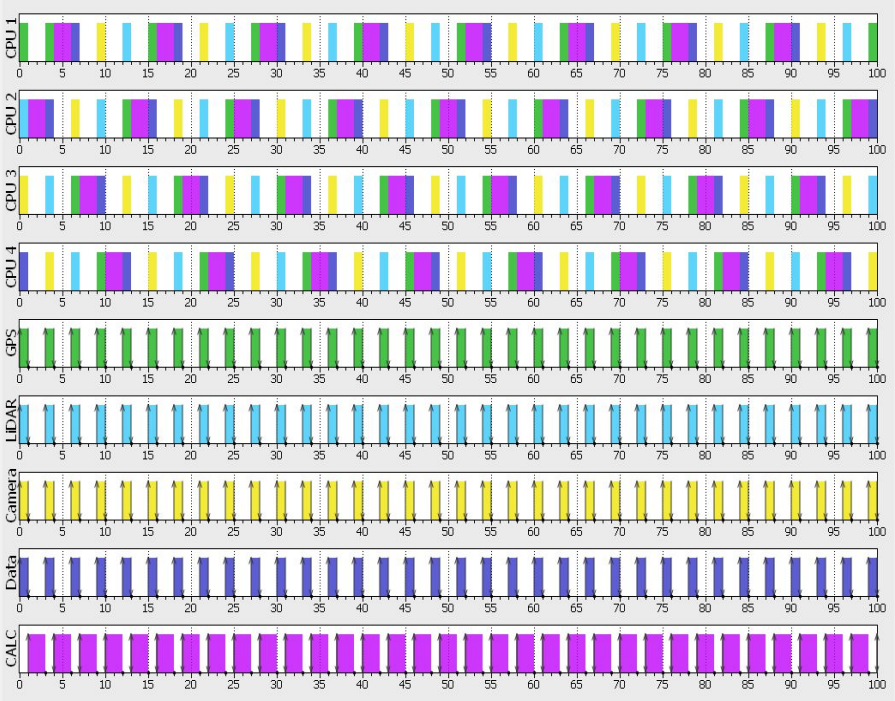
Single core 6ms for completing one cycle, whereas 3 ms in Multi-core

Single-core CPU load = 100%; Multi-core CPU load = 50%

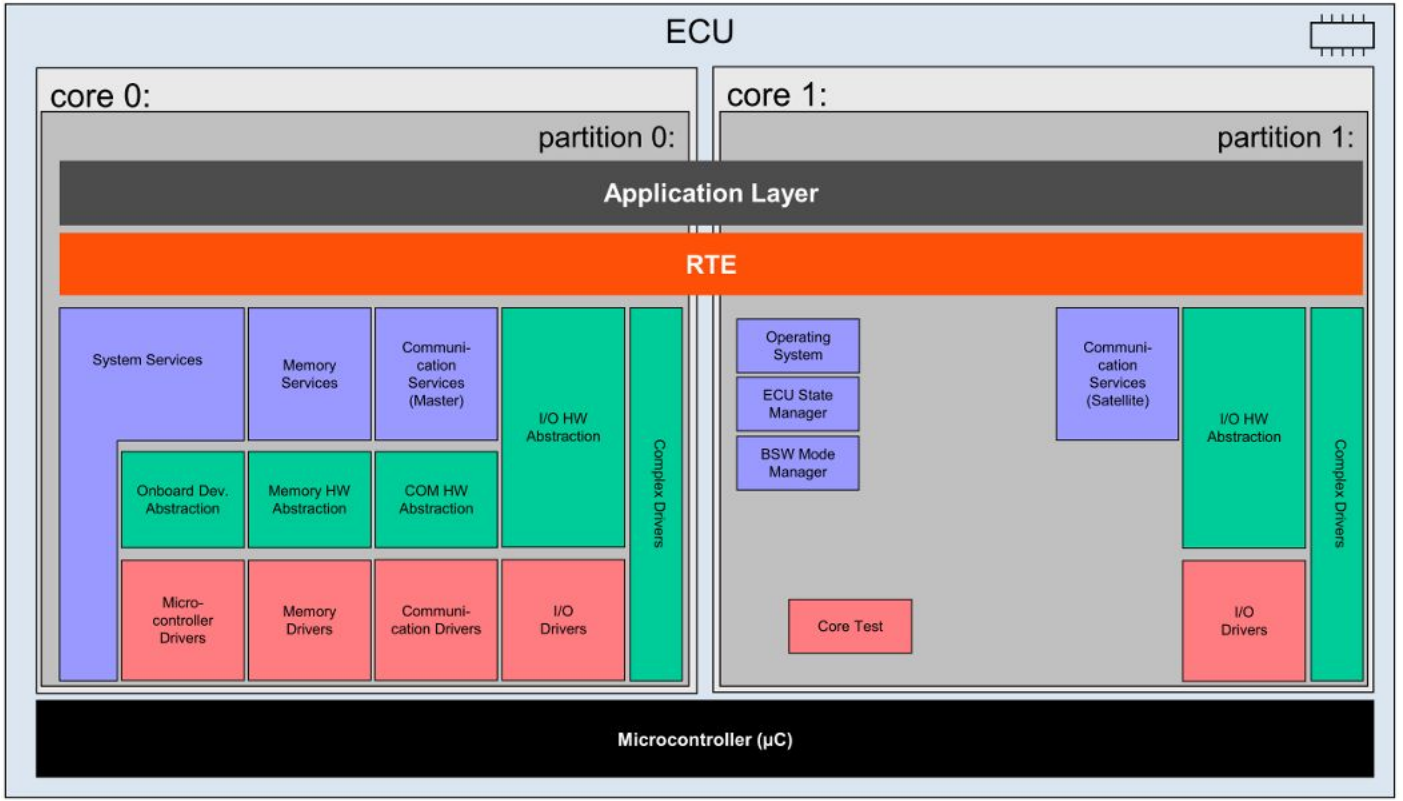
Even it can be improved by instruction level parallelism of the last task which is assigned as 2ms computational time.

Hardware - Scheduling





Hardware - AUTOSAR's Multi-Core Archi



Code Implementation

- C++ and Visual Studio is utilized
- Vector data structure is used
- Platooning is created
- While creating plate of previous truck is added
- Distance is check repeatedly

```
Please insert number of trucks in the platoon:3
Please insert front x coordinate of truck 1:10
Please insert back x coordinates of truck 1:15
Please insert front y coordinate of truck 1 :10
Please insert front y coordinate of truck 1 :15
Please enter desired distance between the trucks:5
Please insert plate number of truck 1 :a
Please insert plate number of truck 2 :b
Please insert plate number of truck 3 :c
Distance between 2 and 1 is :5
Distance between 3 and 2 is :5
```

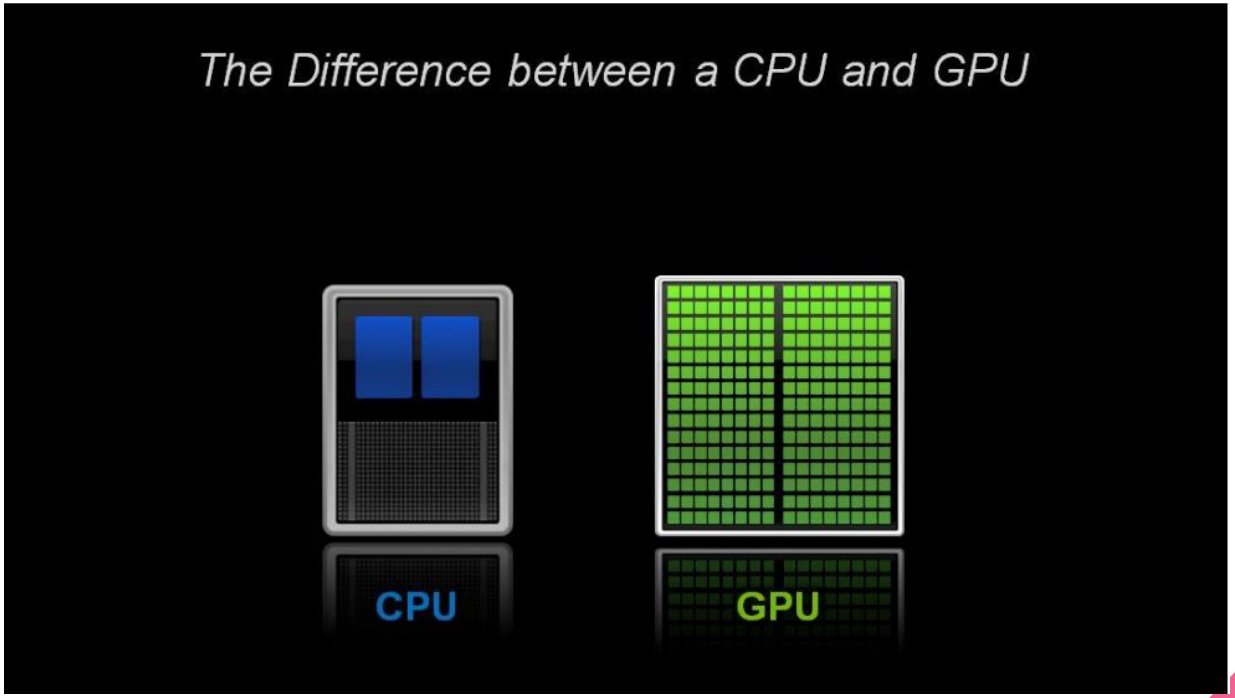
```
Front latitude :12
Back latitude :12
Front longitude :15
Back longitude :15
a   ==>

Front latitude :15
Back latitude :15
Front longitude :15
Back longitude :15
b   ==> a

Front latitude :18
Back latitude :18
Front longitude :15
Back longitude :15
c   ==> b
```

Code Implementation - Parallel Prog

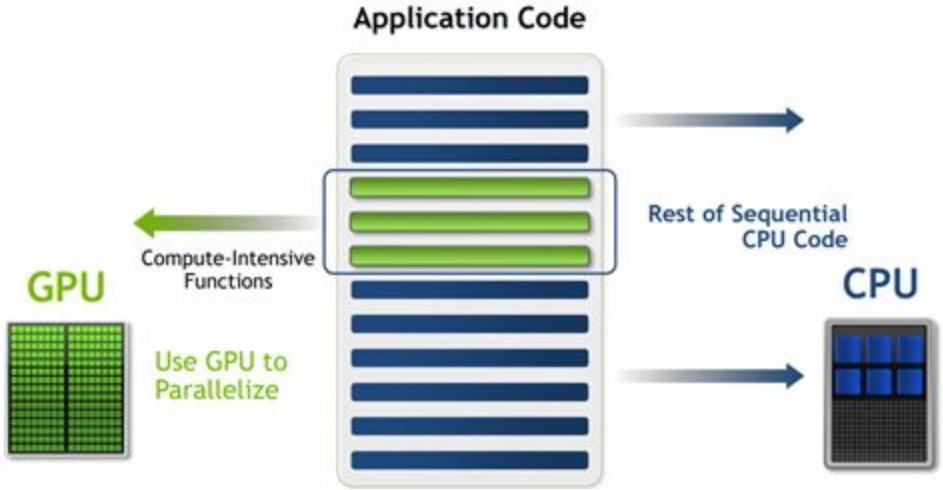
- Parallel programming is utilized for checking intruder
- Thread is method that is used
- By using previous data plate number we can make make sure whether it is working or not

```
Please insert number of trucks in the platoon:10
Please insert front x coordinate of truck 1:10
Please insert back x coordinates of truck 1:15
Please insert front y coordinate of truck 1 :10
Please insert front y coordinate of truck 1 :15
Please enter desired distance between the trucks:5
Please insert plate number of truck 1 :a
Please insert plate number of truck 2 :b
Please insert plate number of truck 3 :c
Please insert plate number of truck 4 :d
Please insert plate number of truck 5 :e
Please insert plate number of truck 6 :f
Please insert plate number of truck 7 :g
Please insert plate number of truck 8 :h
Please insert plate number of truck 9 :i
Please insert plate number of truck 10 :j
Checking for the intruder for all trucks :
Turck 2 and 1 are running suffeciently Turck 8Turck 9 and 8 are running suffeciently

 and 7 are running suffeciently
Turck 5 and 4Turck 7 and 6 are running suffeciently
Turck 3 and 2 are running suffeciently Turck 10 and Turck 6 and
9 are running suffeciently
5 are running suffeciently Turck 4 and
 are running suffeciently 3 are running suffeciently
```

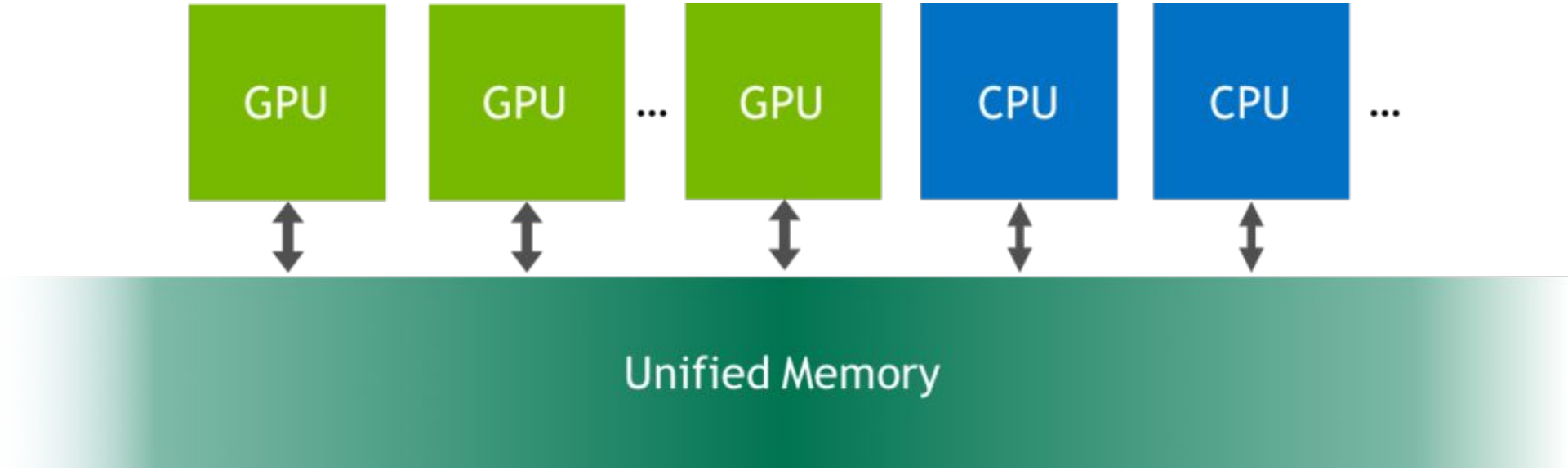
CPU & GPU



GPGPU - CPU & GPU



GPU Memory Management



CPU
18% 3,99 GHz

Bellek
8,2/15,8 GB (52%)

0 Diski (C:)
SSD
%1

Ethernet
VirtualBox Host-Only Network
G: 0 A: 0 Kb/sn

Wi-Fi
Wi-Fi 2
G: 0 A: 0 Kb/sn

GPU 0
Intel(R) UHD Graphics
6%

GPU 1
NVIDIA GeForce GTX 1650 Ti
100% (45 °C)

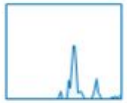
CPU
4% 3,64 GHz

Bellek
8,1/15,8 GB (51%)

0 Diski (C:)
SSD
%0

Ethernet
VirtualBox Host-Only Network
G: 0 A: 0 Kb/sn

Wi-Fi
Wi-Fi 2
G: 0 A: 0 Kb/sn



GPU 0
Intel(R) UHD Graphics
3%

GPU 1
NVIDIA GeForce GTX 1650 Ti
0% (45 °C)

```
Current speed of t1: 80
Speed is setting to avarage speed for t1...
Current speed of t2: 120
Speed is setting to avarage speed for t2...
Current speed of t3: 150
Speed is setting to avarage speed for t3...
Current speed 81.000000
Current speed 82.000000
Current speed 83.000000
Current speed 84.000000
Current speed 85.000000
Current speed 86.000000
Current speed 87.000000
Current speed 88.000000
Current speed 89.000000
Current speed 90.000000
Current speed 91.000000
Current speed 92.000000
Current speed 93.000000
Current speed 94.000000
Current speed 95.000000
Current speed 96.000000
Current speed 97.000000
Current speed 98.000000
Current speed 99.000000
Current speed 100.000000
Avarage speed achieved...
Current speed 119.000000
Current speed 118.000000
Current speed 117.000000
Current speed 116.000000
Current speed 115.000000
Current speed 114.000000
Current speed 113.000000
Current speed 112.000000
Current speed 111.000000
Current speed 110.000000
Current speed 109.000000
Current speed 108.000000
Current speed 107.000000
Current speed 106.000000
Current speed 105.000000
Current speed 104.000000
Current speed 103.000000
Current speed 102.000000
Current speed 101.000000
Current speed 100.000000
Avarage speed achieved...
Current speed 149.000000
```