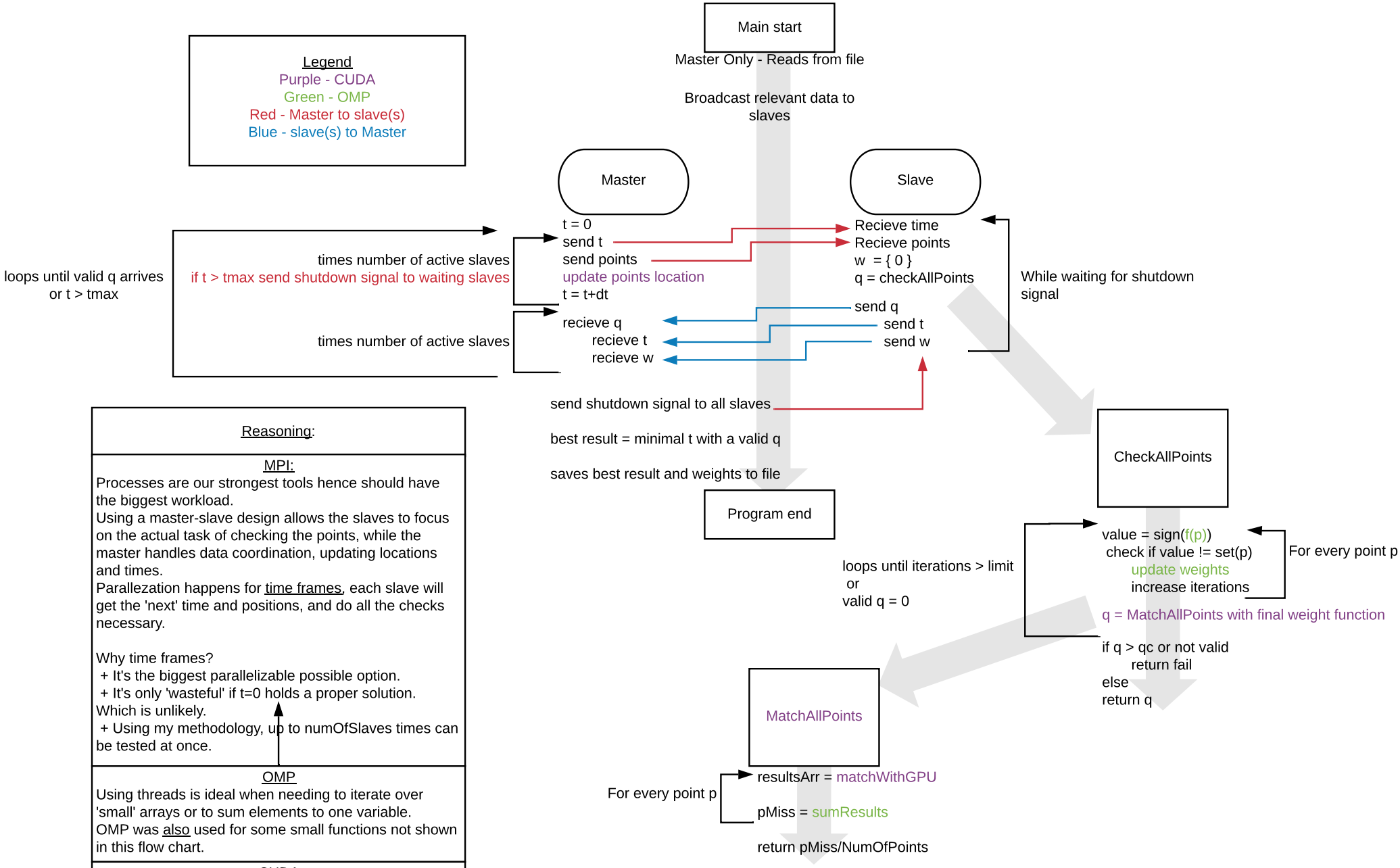


Reasoning:

MPI:

Processes are our strongest tools hence should have the biggest workload.
Using a master-slave design allows the slaves to focus on the actual task of checking the points, while the master handles data coordination, updating locations and times.
Parallezation happens for time frames, each slave will get the 'next' time and positions, and do all the checks necessary.

Why time frames?
+ It's the biggest parallelizable possible option.
+ It's only 'wasteful' if t=0 holds a proper solution. Which is unlikely.
+ Using my methodology, up to numOfSlaves times can be tested at once.

OMP

Using threads is ideal when needing to iterate over 'small' arrays or to sum elements to one variable.
OMP was also used for some small functions not shown in this flow chart.

CUDA

Because the transfer times from the CPU to the GPU or back takes a considerable amount of time, it's ideal to use it when needing to work on very large arrays.
I've used CUDA in two cases where there was a huge number of small tasks.
- Updating point locations (number of points times dimension size)
- Match points (number of points)

Project by Chen Zecharya (Dreamest)