Parallel Programming
Final Project-PSO

Group 29

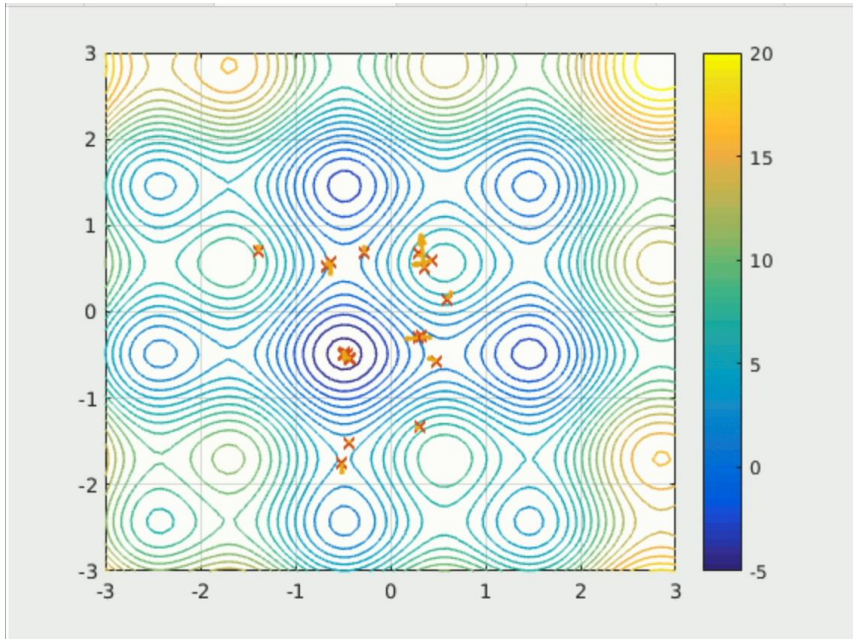
Outline

- What is particle swarm optimization (PSO) ？
- How to implement it?
- How to optimize?
- Experiment & Analysis

What is particle swarm optimization (PSO) ？

Particle swarm optimization (PSO)

- An Optimization Method Based on Population Dynamics
- The concept originally came from a simulation of social behavior
- Individuals will be influenced by their past experience and cognition and overall social behavior, and form an evolutionary advantage



Source:
https://en.wikipedia.org/wiki/Particle_swarm_optimization

Particle swarm optimization (PSO)

- Characteristic :
  - It can effectively search for problems with a huge solution space and find candidate solutions without knowing too much information about the problem.
  - Able to find the best solution or the value closest to the best solution
- Use cases :
  - Solve the problem of finding the best solution
  - Used in neural network algorithms
  - Solve efficiently CSPs (Constraint Satisfaction Problems)

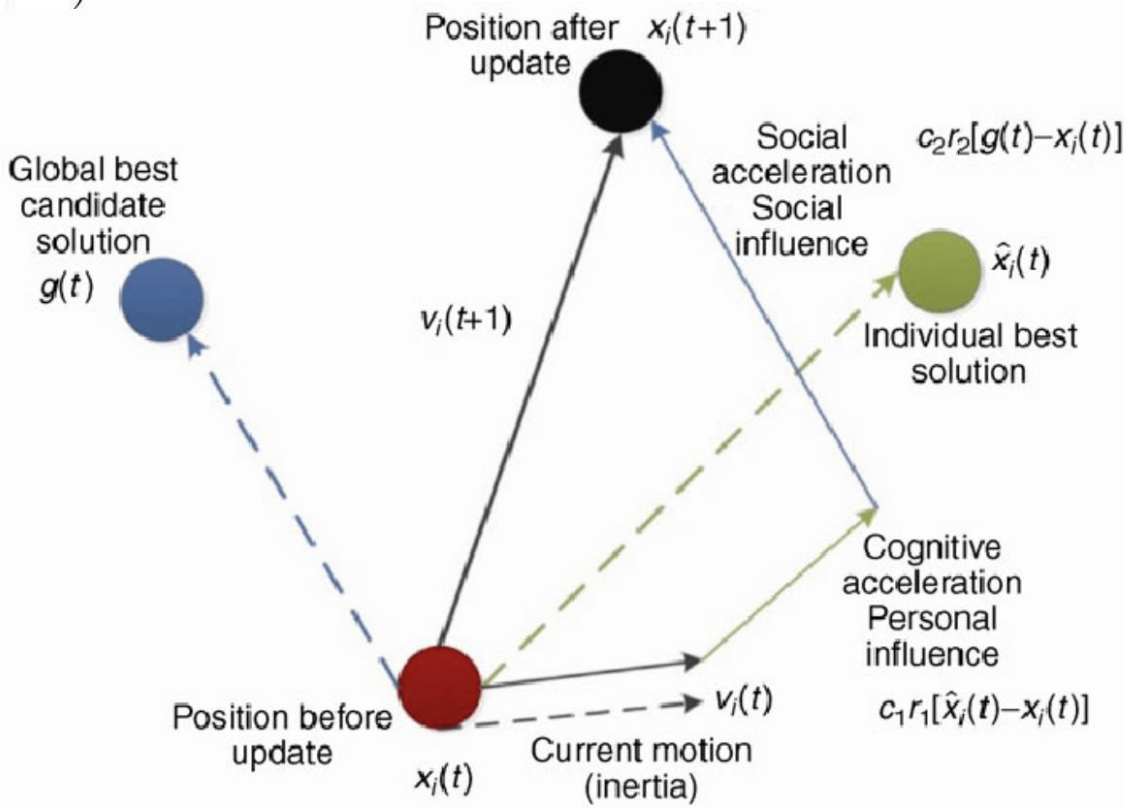
Particle swarm optimization (PSO)-Principle

- Update the velocity of each particle

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c_1(\mathbf{p}_i - \mathbf{x}_i(t))\mathbf{R}_1 + c_2(\mathbf{g} - \mathbf{x}_i(t))\mathbf{R}_2$$

- Update the position of each particle

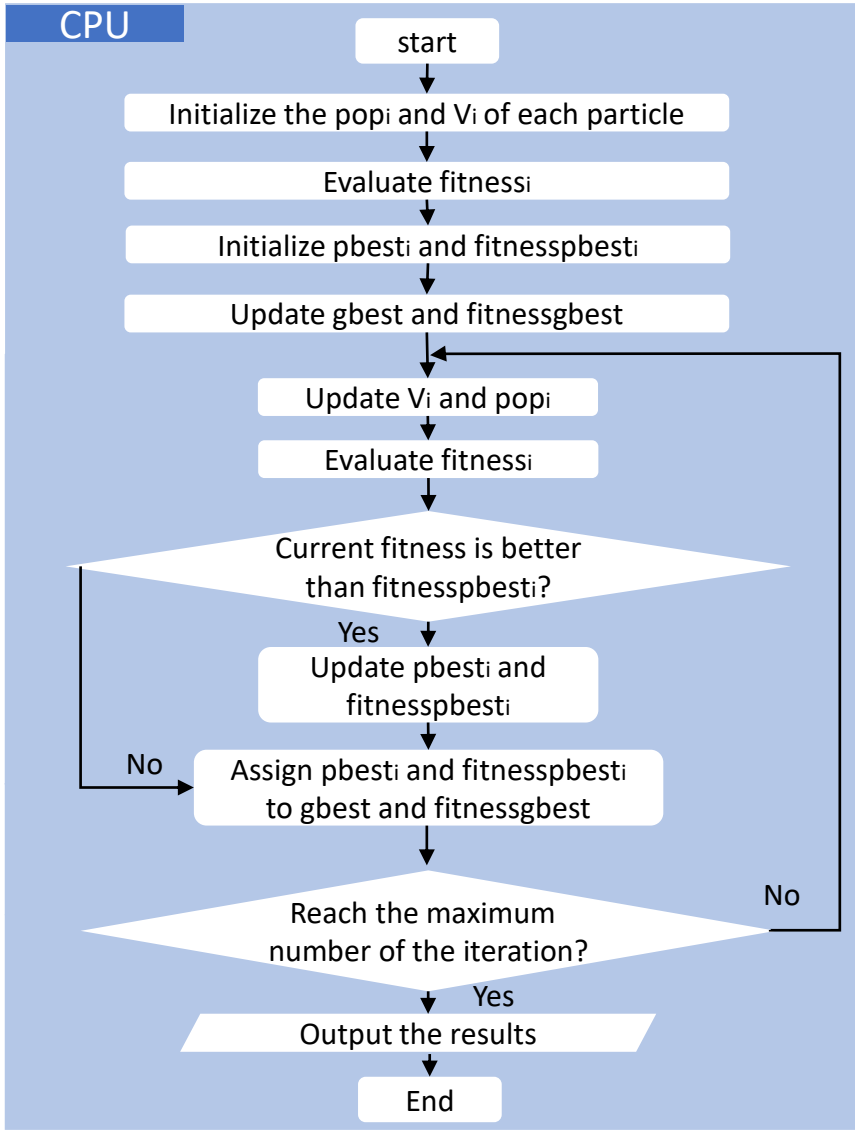
$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$



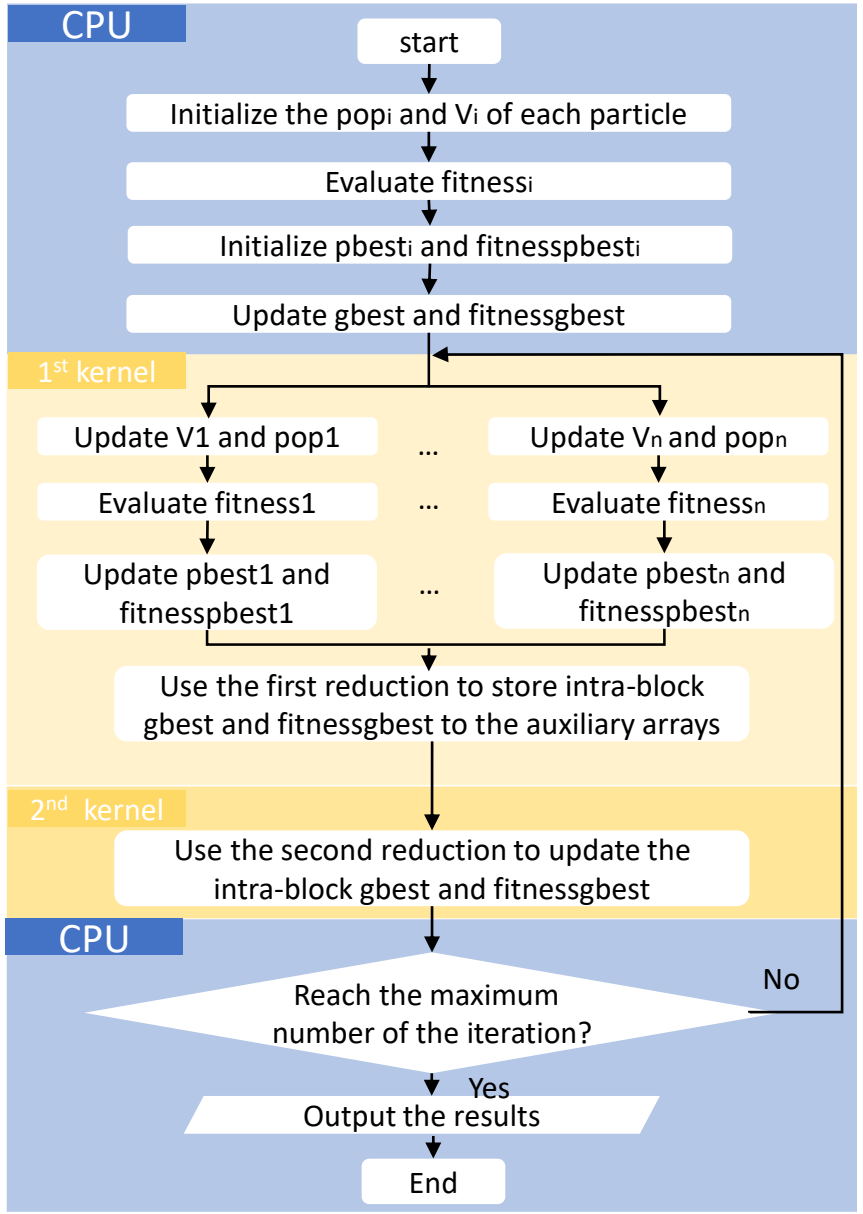
Source:https://ithelp.ithome.com.tw/articles/10231724

How to implement it?

The workflow of the sequential PSO algorithm



The workflow of the parallel PSO algorithm



How to optimize?

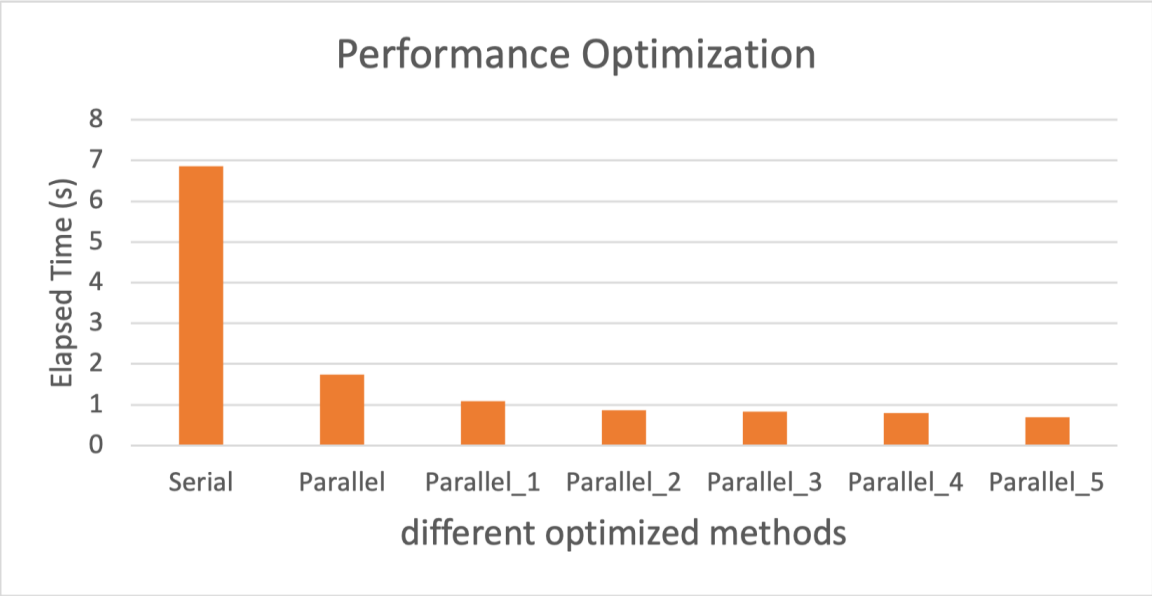
Optimization

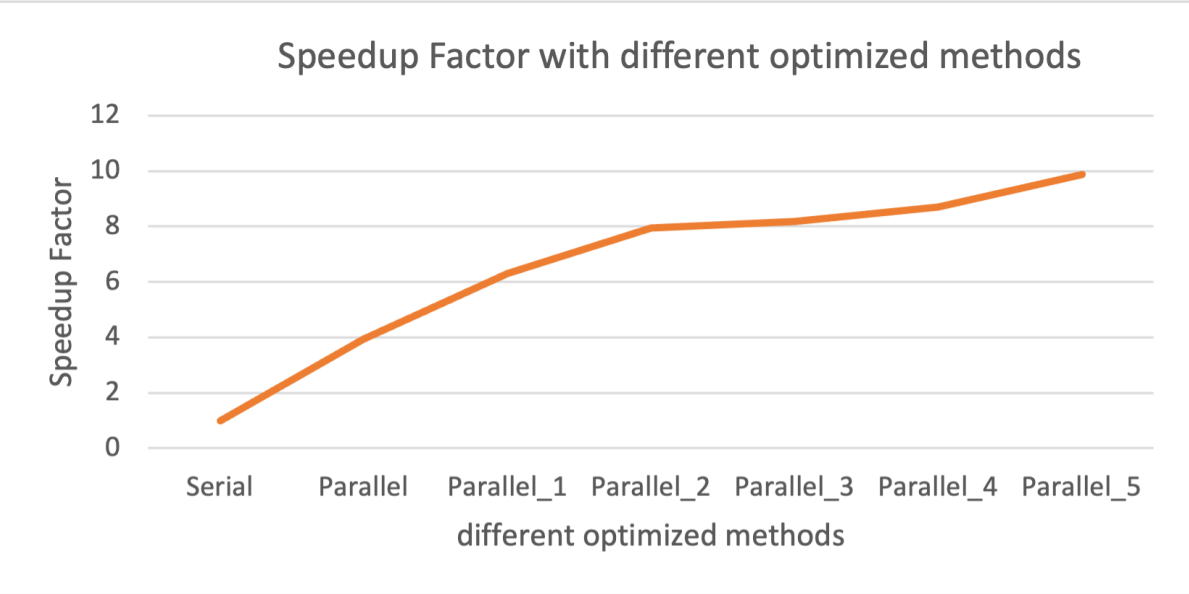
1. Shared memory、Coalesced Memory Access: Put frequently accessed data into share memory
2. Interleaved addressing: handle warp divergence
3. Sequential addressing: handle bank conflict
4. Unroll
5. Mixed-Precision(Use float to replace double)

Experiment & Analysis

Experiment & Analysis

- Parallel : Baseline
- Parallel_1 : Shared memory、Coalesced Memory Access
- Parallel_2 : Shared memory、Coalesced Memory Access 、handle warp divergent
- Parallel_3 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict
- Parallel_4 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict、Unroll
- Parallel_5 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict 、Unroll、double -> float

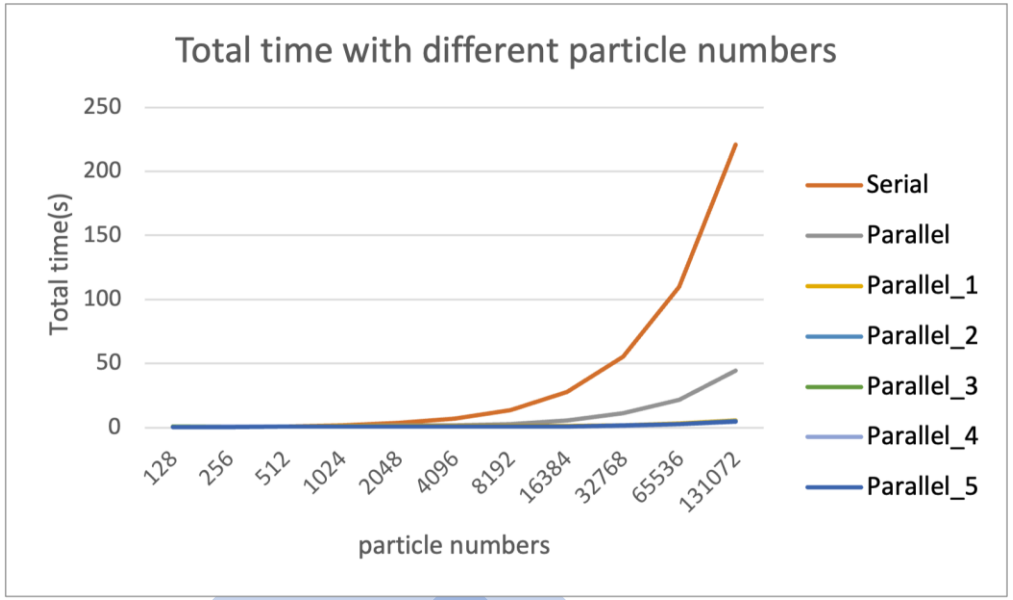




Experiment & Analysis

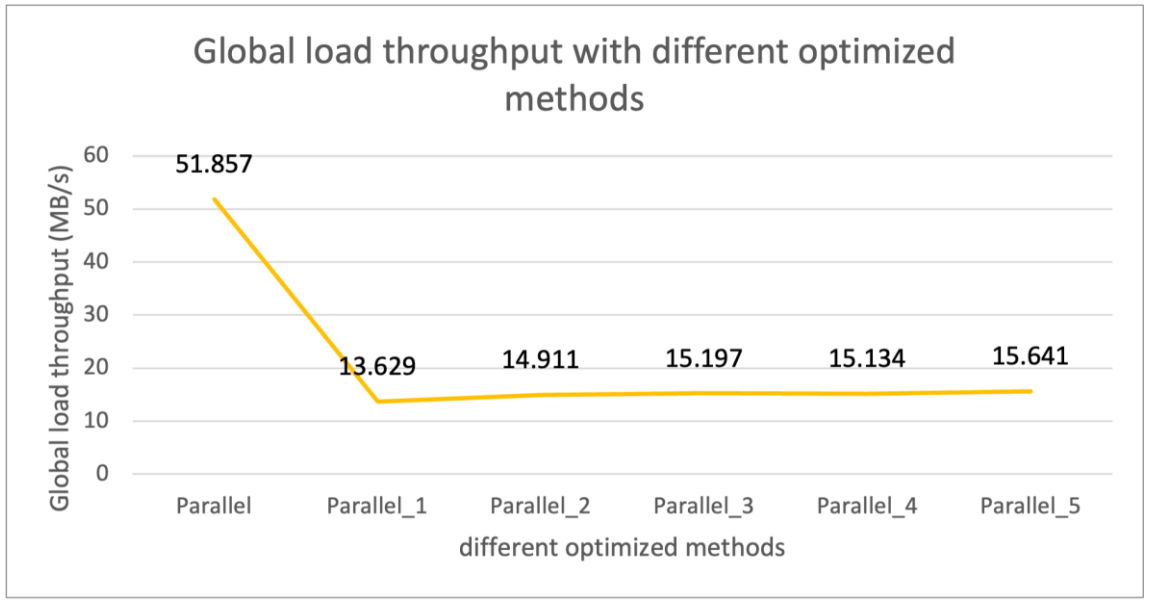
- Parallel : Baseline
- Parallel_1 : Shared memory、Coalesced Memory Access
- Parallel_2 : Shared memory、Coalesced Memory Access 、handle warp divergent
- Parallel_3 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict
- Parallel_4 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict、Unroll
- Parallel_5 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict 、Unroll、double -> float

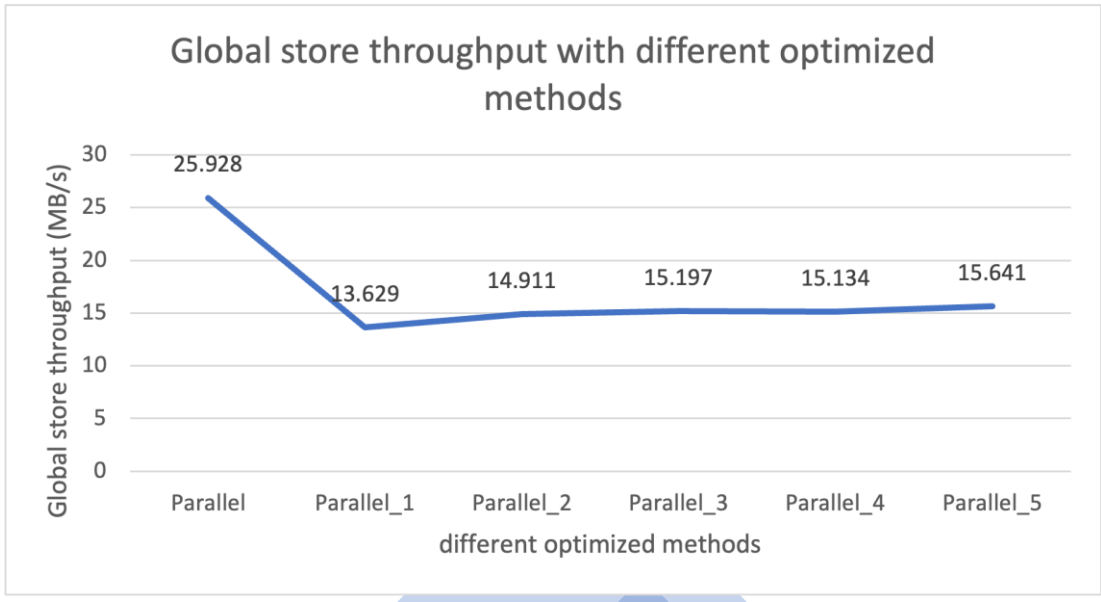
Particles	Iteration	Total Time_Serial(s)	Total Time_Parallel(s)	Total Time_Parallel_1(s	Total Time_Parallel_	Total Time_Parallel_3	Total Time_Parall	Total Time_Parallel_5
128	100000	0.22058	0.680846	0.616901	0.522089	0.538232	0.494345	0.479629
256	100000	0.450724	0.764967	0.628925	0.59071	0.583021	0.568412	0.489103
512	100000	0.8511	0.964059	0.711591	0.616674	0.598647	0.591764	0.535207
1024	100000	1.697453	1.372173	1.049532	0.827267	0.812177	0.754653	0.590641
2048	100000	3.404291	1.454872	1.024608	0.889581	0.778589	0.773349	0.602957
4096	100000	6.864119	1.747627	1.087807	0.863102	0.838115	0.788841	0.69506
8192	100000	13.726206	2.777033	1.091495	0.915003	0.821749	0.818323	0.675779
16384	100000	27.846163	5.50088	1.15595	0.932236	0.907452	0.896619	0.734849
32768	100000	55.258001	11.168123	1.637492	1.304801	1.186819	1.158356	1.548642
65536	100000	110.009652	21.587691	3.306193	2.798614	2.600244	2.571494	2.409548
131072	100000	221.051729	44.585769	5.379395	4.910112	4.530291	4.527019	4.729527



Experiment & Analysis

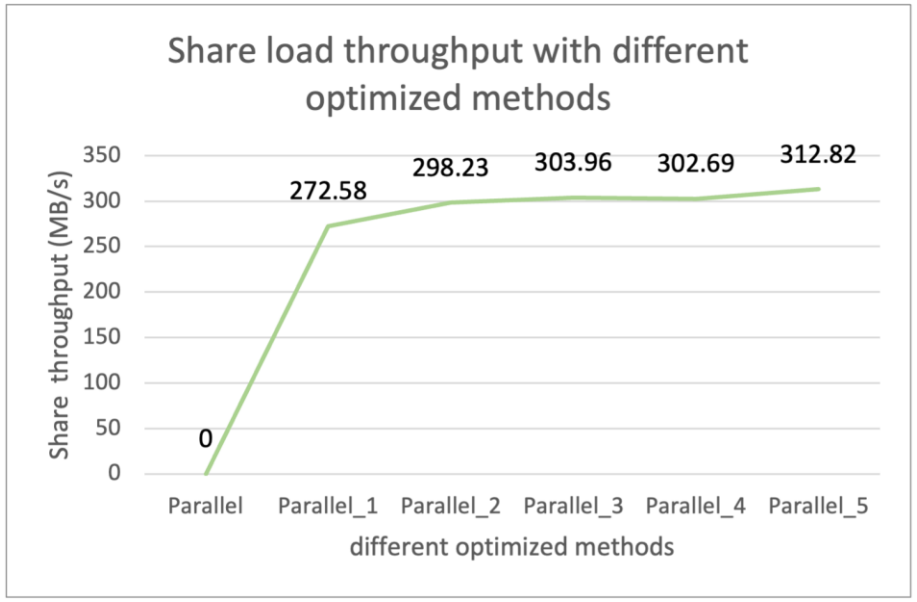
- Parallel : Baseline
- Parallel_1 : Shared memory、Coalesced Memory Access
- Parallel_2 : Shared memory、Coalesced Memory Access 、handle warp divergent
- Parallel_3 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict
- Parallel_4 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict、Unroll
- Parallel_5 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict 、Unroll、double -> float

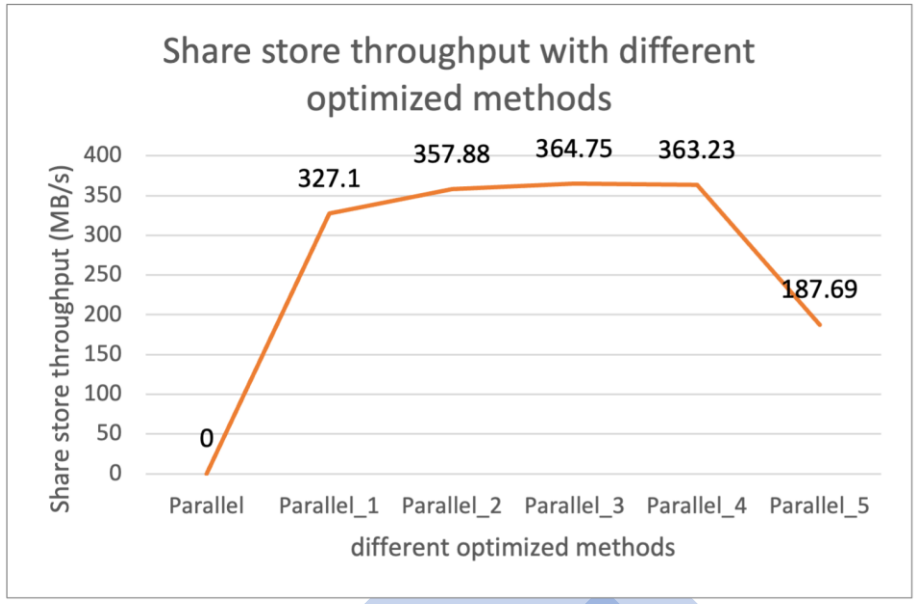




Experiment & Analysis

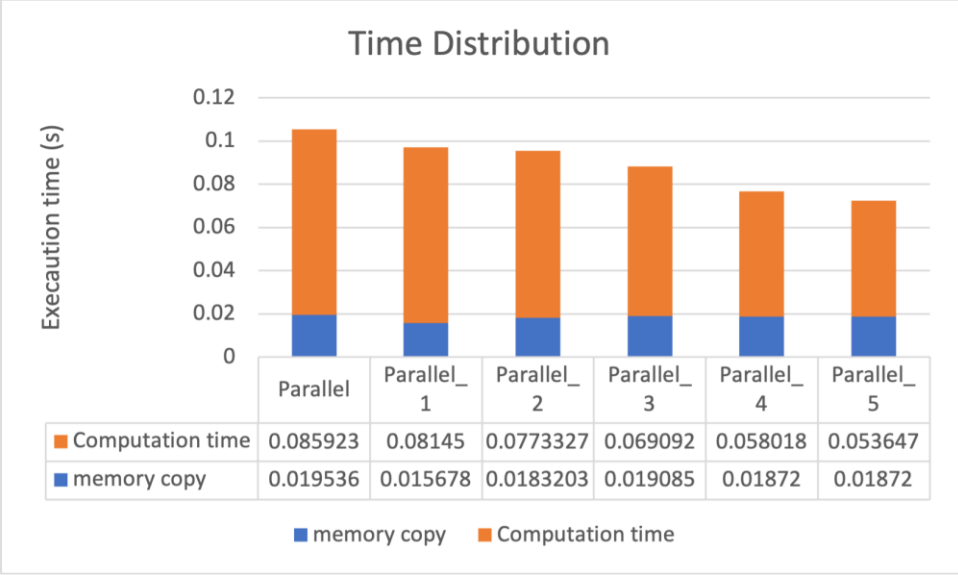
- Parallel : Baseline
- Parallel_1 : Shared memory、Coalesced Memory Access
- Parallel_2 : Shared memory、Coalesced Memory Access 、handle warp divergent
- Parallel_3 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict
- Parallel_4 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict、Unroll
- Parallel_5 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict 、Unroll、double -> float



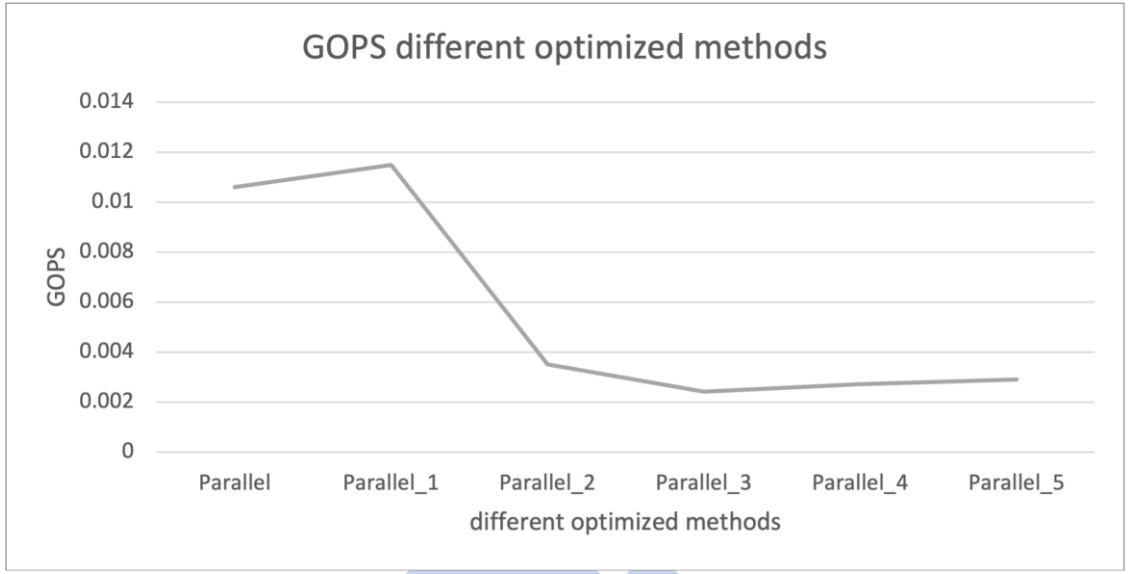


Experiment & Analysis

- Parallel : Baseline
- Parallel_1 : Shared memory、Coalesced Memory Access
- Parallel_2 : Shared memory、Coalesced Memory Access 、handle warp divergent
- Parallel_3 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict
- Parallel_4 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict、Unroll
- Parallel_5 : Shared memory、Coalesced Memory Access 、handle warp divergent、handle bank conflict 、Unroll、double -> float



	Parallel	Parallel_1	Parallel_2	Parallel_3	Parallel_4	Parallel_5
Computation time	0.085923	0.08145	0.0773327	0.069092	0.058018	0.053647
memory copy	0.019536	0.015678	0.0183203	0.019085	0.01872	0.01872



Thank you for listening

Group 29