0-1 KNAPSACK PROBLEM USING MPI

CSE 633 Parallel Algorithms

Instructor: Professor Russ Miller

Author: Pushkar Pandey

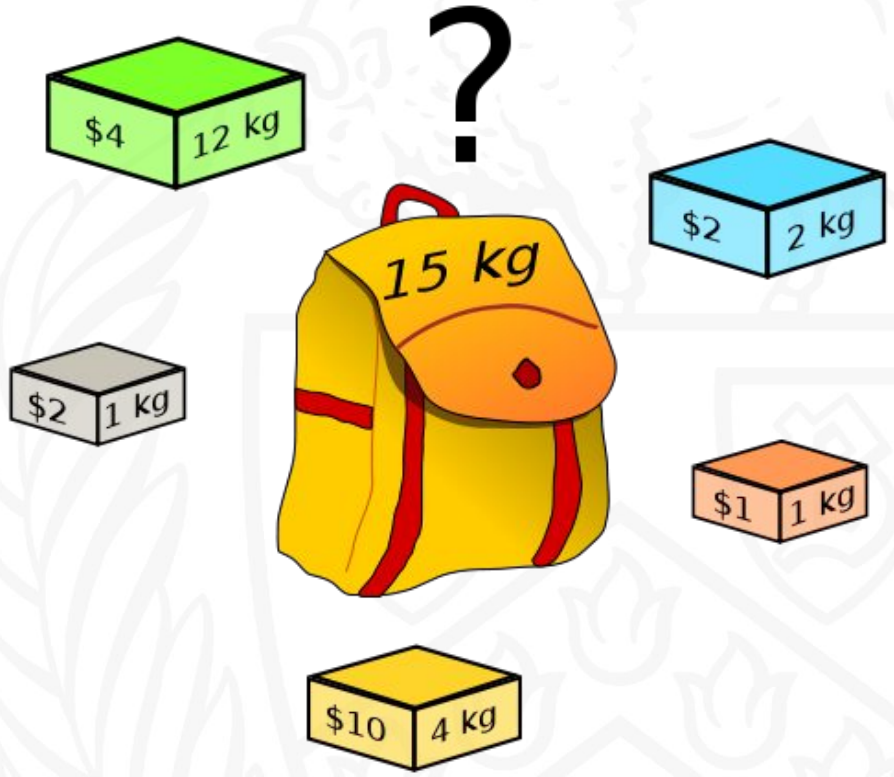
University at Buffalo The State University of New York

CONTENT:

Introduction to 0-1 Knapsack Problem

- Problem of combinatorial optimization
- A set of items with a weight and a value given a knapsack with a maximum weight it can carry

Find which items to take to get the best value but not exceed the knapsack capacity



Example of Knapsack Problem

0-1 Knapsack Problem

value[] = {60, 100, 120};
weight[] = {10, 20, 30};
W = 50;

Solution: 220

Weight = 10; Value = 60;
Weight = 20; Value = 100;
Weight = 30; Value = 120;
Weight = (20+10); Value = (100+60);
Weight = (30+10); Value = (120+60);
Weight = (30+20); Value = (120+100);
Weight = (30+20+10) > 50

Sequential Implementation

```
1  // Input:
2  // Values (stored in array v)
3  // Weights (stored in array w)
4  // Number of distinct items (n)
5  // Knapsack capacity (W)
6  // NOTE: The array "v" and array "w" are assumed to store all relevant values starting at index 1.
7
8  array m[0..n, 0..W];
9  for j from 0 to W do:
10     m[0, j] := 0
11 for i from 1 to n do:
12     m[i, 0] := 0
13
14 for i from 1 to n do:
15     for j from 0 to W do:
16         if w[i] > j then:
17             m[i, j] := m[i-1, j]
18         else:
19             m[i, j] := max(m[i-1, j], m[i-1, j-w[i]] + v[i])
```

Sequential Implementation Example

i	v	w
1	5	4
2	4	3
3	3	2
4	2	1

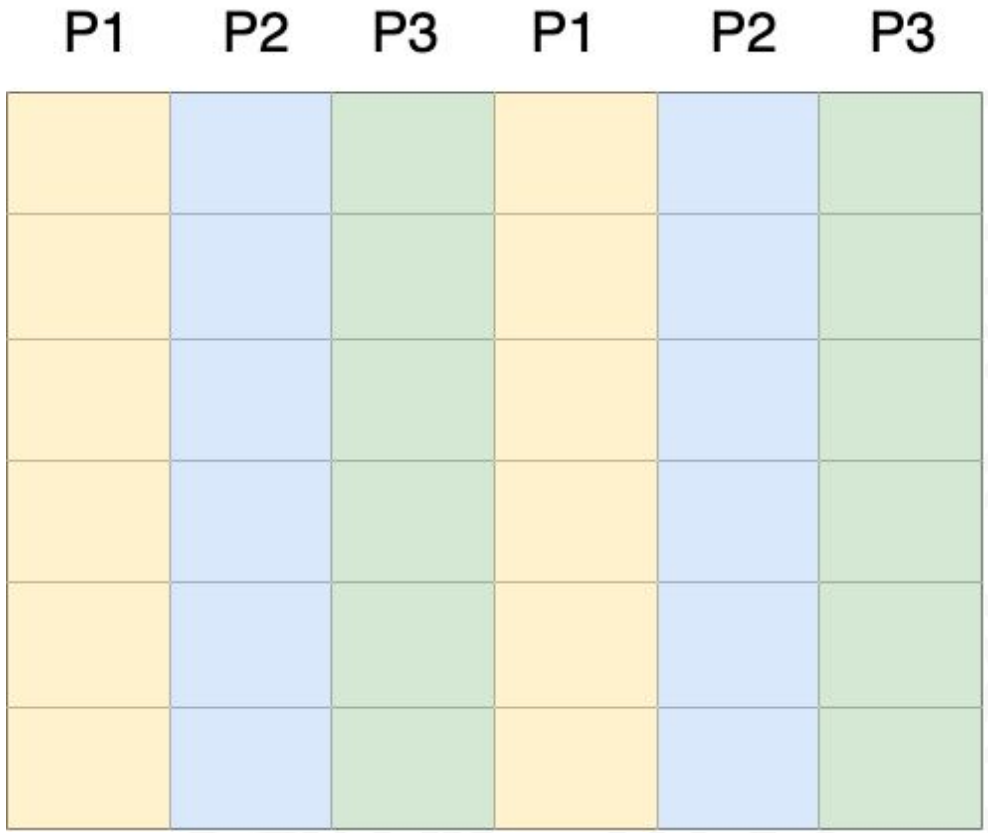
Capacity=6

		w						
		0	1	2	3	4	5	6
i	0							
	1							
	2							
	3							
	4							

Parallel Implementation

We do column parallelization

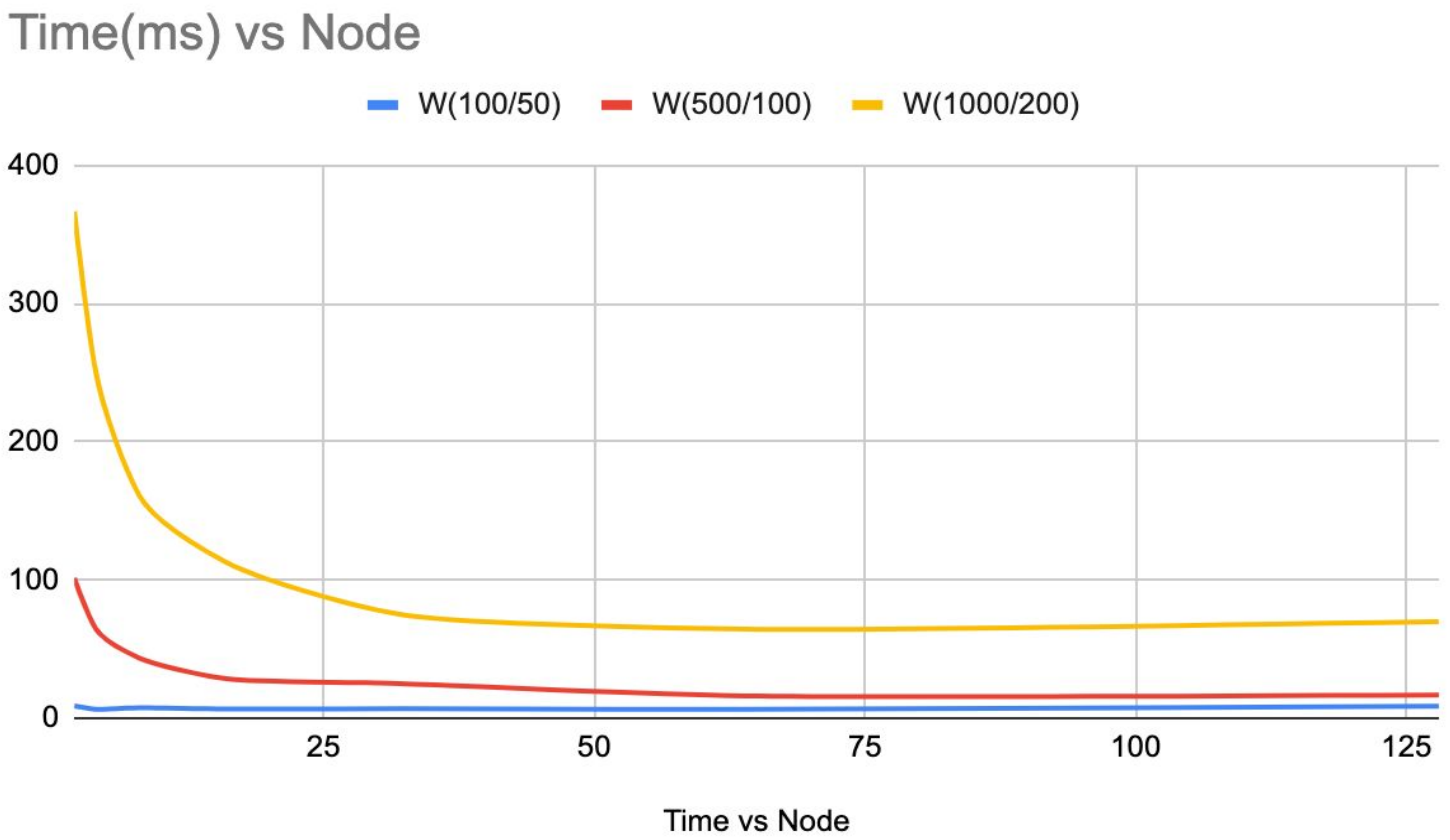
- Compute the maximum value achievable using the item of the row
- Compute the value without the new item. This value is the value just above in the matrix or 0 if it is the first item.
- Save in the cell the maximum value achievable using or not the new item
- Send to all the processors that could need it in future iteration the new value.



Output Analysis

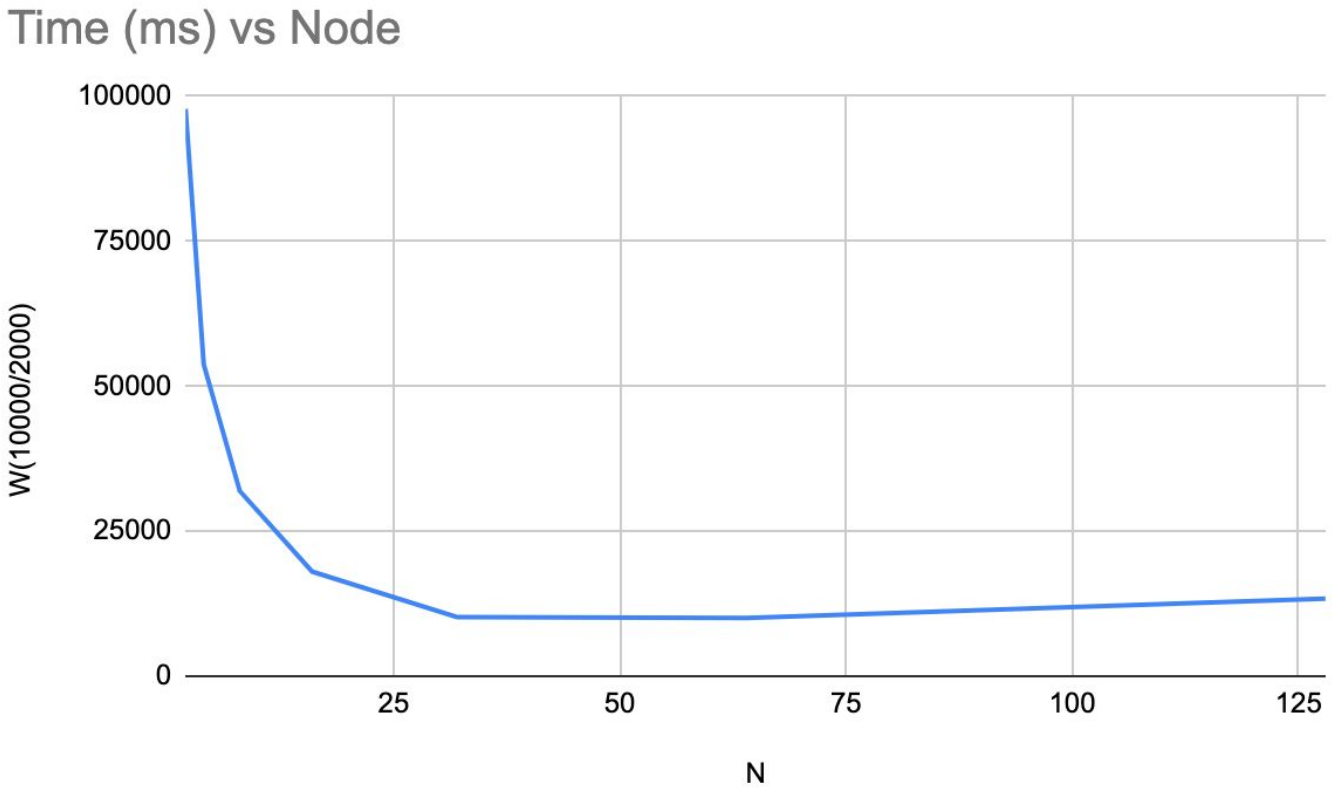
N	W(100/50)	W(500/100)	W(1000/200)	W(10000/2000)
2	8.914	101.475	366.621	97813.2
4	6.491	64.479	249.763	53803.5
8	7.472	43.489	161.116	31933.9
16	6.62	28.546	112.93	18028.7
32	6.772	25.019	75.17	10178
64	6.364	16.072	64.352	10046.5
128	7.543	16.772	69.912	13348.6

Graph and Analysis



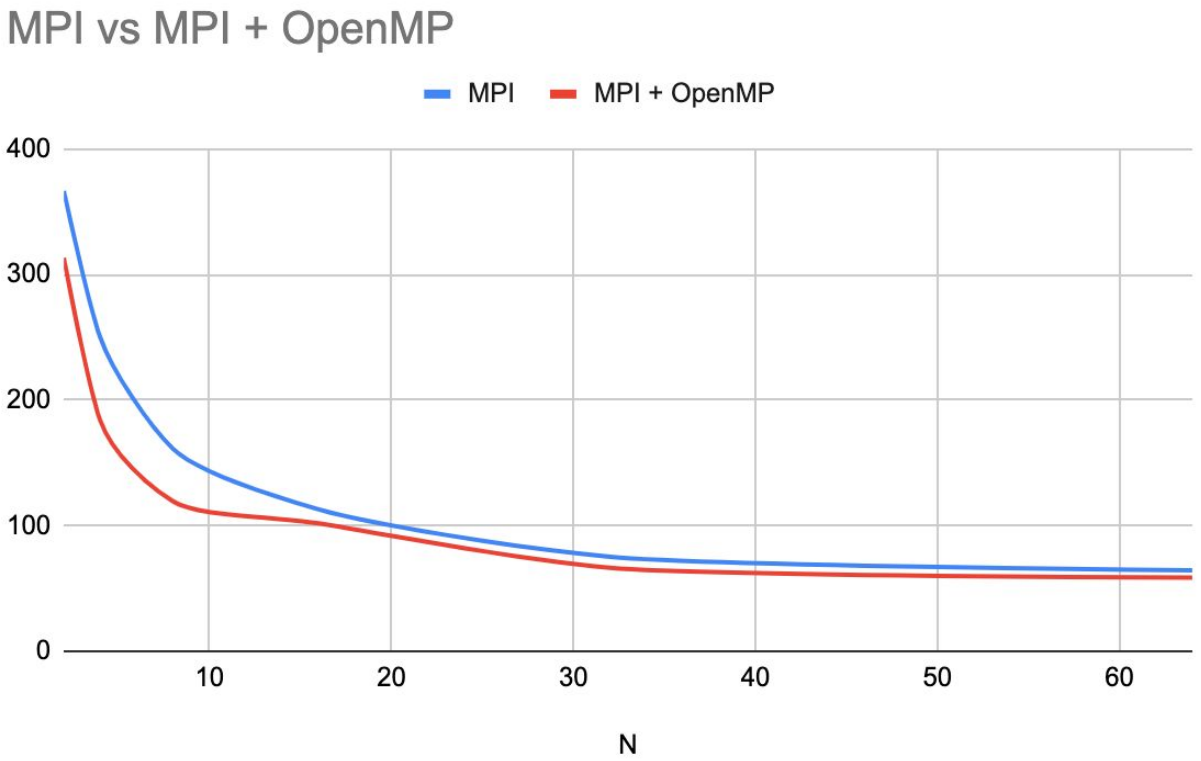
Amdahl's Graph for W(10000/2000)

N	W(10000/2000)
2	97813.2
4	53803.5
8	31933.9
16	18028.7
32	10178
64	10046.5
128	13348.6



MPI vs MPI + OpenMP (hybrid model)

N	MPI	MPI + OpenMP
2	366.621	313.2
4	249.763	183.5
8	161.116	119.3
16	112.93	101.8
32	75.17	66.4
64	64.352	58.6



References:

- https://en.wikipedia.org/wiki/Knapsack_problem
- https://ieeexplore.ieee.org/abstract/document/6234092
- https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/

Thanks You