

# UNIVERSITY OF SALERNO

DEPARTMENT OF INFORMATION ENGINEERING AND ELECTRICAL AND  
APPLIED MATHEMATICS

Master's Degree in Computer Engineering



Project report  
Parallel version of the Dijkstra's algorithm

Course: High Performance Computing  
Lecturer: Francesco Moscato      [fmoscato@unisa.it](mailto:fmoscato@unisa.it)  
Alberti Andrea    0622702370      [a.alberti2@studenti.unisa.it](mailto:a.alberti2@studenti.unisa.it)



ACADEMIC YEAR 2023/2024

## TABLE OF CONTENT

Problem description .....	5
1.1    Algorithm description .....	5
1.2    Algorithm implementation example .....	6
Proposed Solution .....	9
2.1    OMP + MPI Version.....	9
2.2    OMP + CUDA Version.....	9
Experimental Setup .....	10
3.1    Hardware .....	10
3.2    Software .....	11
Speedup, Performance & Efficiency.....	12
Test “OpenMP + MPI”.....	13
5.1    Optimization 0.....	13
5.1.1    Type 0.....	13
5.1.2    Type 1.....	16
5.1.3    Type 2.....	19
5.1.4    Type 3.....	22
5.2    Optimization 1.....	25
5.2.1    Type 0.....	25
5.2.2    Type 1.....	28
5.2.3    Type 2.....	31
5.2.4    Type 3.....	34
5.3    Optimization 2.....	37
5.3.1    Type 0.....	37
5.3.2    Type 1.....	40
5.3.3    Type 2.....	43
5.3.4    Type 3.....	46
5.4    Optimization 3.....	49
5.4.1    Type 0.....	49
5.4.2    Type 1.....	52
5.4.3    Type 2.....	55
5.4.4    Type 3.....	58
Test “OpenMP + CUDA”.....	61
6.1    Optimization 0.....	61
6.1.1    Type 0.....	61
6.1.2    Type 1.....	64

6.1.3	Type 2.....	67
6.1.4	Type 3.....	70
6.2	Optimization 1.....	73
6.2.1	Type 0.....	73
6.2.2	Type 1.....	76
6.2.3	Type 2.....	79
6.2.4	Type 3.....	82
6.3	Optimization 2.....	85
6.3.1	Type 0.....	85
6.3.2	Type 1.....	88
6.3.3	Type 2.....	91
6.3.4	Type 3.....	94
6.4	Optimization 3.....	97
6.4.1	Type 0.....	97
6.4.2	Type 1.....	100
6.4.3	Type 2.....	103
6.4.4	Type 3.....	106
Final Considerations .....		109
7.1	“OpenMP + MPI” .....	109
7.2	“OpenMP + CUDA” .....	109
API Documentation.....		110
Graph Functions Documentation.....		110
Utility Functions Documentation .....		112
Dijkstra_serial Functions Documentation .....		113
Dijkstra_MPIOpenMP Functions Documentation.....		114
Dijkstra_cuda Functions Documentation .....		115
How to run .....		116

## LIST OF FIGURES

Figure 1: Pseudo Code .....	5
Figure 2: Dijkstra's algorithm example starting graph G .....	6
Figure 3: Dijkstra's algorithm example Step 1 .....	7
Figure 4: Dijkstra's algorithm example Step 2 .....	7
Figure 5: Dijkstra's algorithm example Step 3 .....	8
Figure 6: Dijkstra's algorithm example Step 4 .....	8
Figure 7: CPU characteristics .....	10
Figure 8: RAM characteristics .....	10
Figure 9: GPU characteristics .....	10
Figure 10: Software.....	11

## Problem description

The problem involves implementation of the Dijkstra's algorithm in two parallel versions (OMP+MPI, OMP+CUDA) and evaluation of corresponding performance in terms of speedup and efficiency of the algorithms compared to the serial version.

Three important technologies were used for the implementation:

- OpenMp: a cross-platform API for creating parallel applications on shared-memory systems;
- MPI (Message Passing Interface): a programming model and standard library for communication between parallel processes in high-performance computing and distributed applications;
- CUDA (Compute Unified Device Architecture): a hardware architecture for parallel processing developed by NVIDIA. Through the CUDA development environment, software programmers can write applications capable of parallel computation on NVIDIA GPU graphics cards.

### 1.1 Algorithm description

The Dijkstra's algorithm is used to find the shortest paths from a source node to all other nodes in a weighted graph with non-negative edge weights. It calculates the distance to reach each node from the source and identifies the predecessor node to reach the considered node. Dijkstra's algorithm is a greedy algorithm, i.e. the algorithm makes the optimal choice at each step for finding the overall optimal solution.

A graph is a set of nodes (or vertices) connected by edges. The presented version of the algorithm works on undirected graphs which are represented by the adjacency matrix. Rows and columns of the matrix represent the nodes of the graph, and elements indicate the weight of the edge between the corresponding nodes (if the nodes are not connected the value is set to infinity).

```
1  function Dijkstra(Graph, source):
2
3      for each vertex v in Graph.Vertices:
4          dist[v]  $\leftarrow$  INFINITY
5          prev[v]  $\leftarrow$  UNDEFINED
6          add v to Q
7          dist[source]  $\leftarrow$  0
8
9      while Q is not empty:
10         u  $\leftarrow$  vertex in Q with min dist[u]
11         remove u from Q
12
13         for each neighbor v of u still in Q:
14             alt  $\leftarrow$  dist[u] + Graph.Edges(u, v)
15             if alt < dist[v]:
16                 dist[v]  $\leftarrow$  alt
17                 prev[v]  $\leftarrow$  u
18
19     return dist[], prev[]
```

Figure 1: Pseudo Code

## 1.2 Algorithm implementation example

Dijkstra's algorithm determines the shortest path in graph G, having as starting point the source vertex "s". Three sets, namely "Visited" (V), "Path" (P) and "Distance" (D), play the following roles:

- Visited keeps track of the visited nodes;
- Path keeps track of the predecessor to get to the actual node;
- Distance keeps track of the distances to get to the actual node.

The algorithm is structured like this:

- *Initialization*: begin with only the source vertex "s" in V, P contains the source node for all nodes (-1 for itself), D contains the initial distances from the source (infinity if they are not directly connected)
- *Select Shortest Vertex*: choose the vertex "u" with the shortest distance from D, add it to V.
- *Update Distances*: adjust the distances and the paths by considering the shortest path found through vertex "u." For instance, if the distance from "s" to vertex "v" is greater than the sum of distances from "s" to "u" and "u" to "v," update the distance of (s, v).
- *Repeat*: iterate through steps 2 and 3 until all vertices have been visited.

Following is reported an example starting from source vertex "A" (see Figure 2).

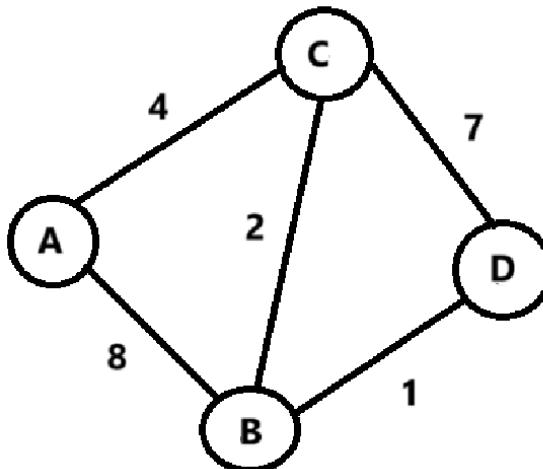


Figure 2: Dijkstra's algorithm example starting graph G

The graph G is represented by the following adjacency matrix (the nodes are sorted alphabetically):

0	8	4	$\infty$
8	0	2	1
4	2	0	7
$\infty$	1	7	0

Main steps of the algorithm are the following:

1. Choose vertex A:

V: {A}

D: {A(-1), B(8), C(4), D( $\infty$ )}

B(8) means the distance from vertex B to A is 8

P: {A(A), B(A), C(A), D(A))}}

B(A) means that the predecessor of B is A

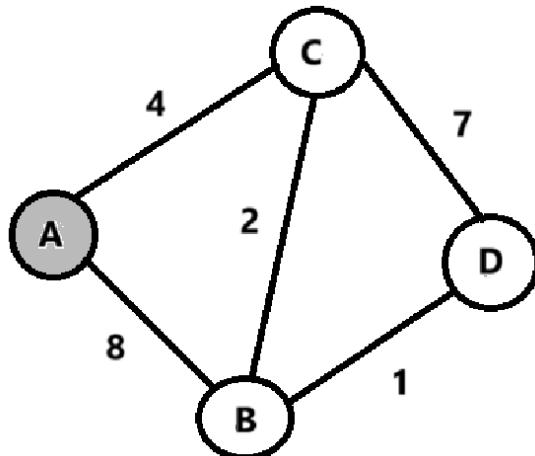


Figure 3: Dijkstra's algorithm example Step 1

2. Choose vertex C (minimum in D of the unvisited vertices)

Calculate the new distances and paths:

V: {A, C}

D: {A(-1), B(6), C(4), D(11)}

P: {A(A), B(C), C(A), D(C))}}

D(B) = 6, because the distance A-B is greater than the distance A-C + C-B

D(D) = 11, because the distance A-D is greater than the distance A-C + C-D

P(B) = C, because the path is through C

P(D) = C, because the path is through C

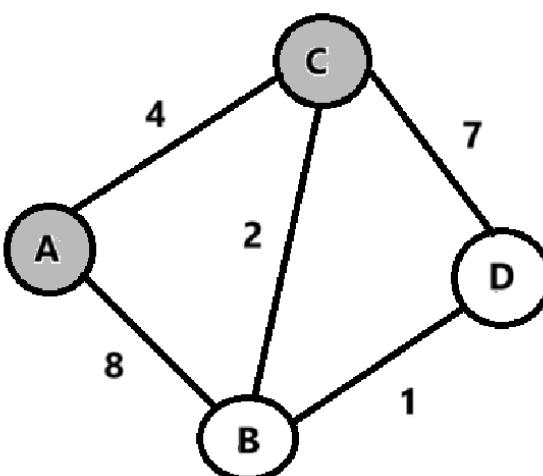


Figure 4: Dijkstra's algorithm example Step 2

3. Choose vertex B (minimum in D of the unvisited vertices)

Calculate the new distances and paths:

V: {A,C,B}

D: {A(-1),B(6),C(4),D(7)}

P: {A(A),B(C),C(A),D(B))})}

$D(D) = 7$ , because the distance A-D is greater than the distance A-B + B-D

$P(D) = B$ , because the path is through B

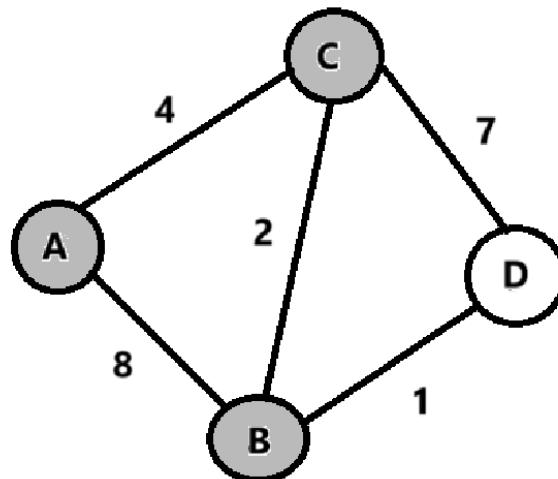


Figure 5: Dijkstra's algorithm example Step 3

4. Choose vertex D (minimum in D of the unvisited vertices)

V: {A,C,B,D}

D: {A(-1),B(6),C(4),D(7)}

P: {A(A),B(C),C(A),D(B))})}

All vertex are visited, the algorithm ends.

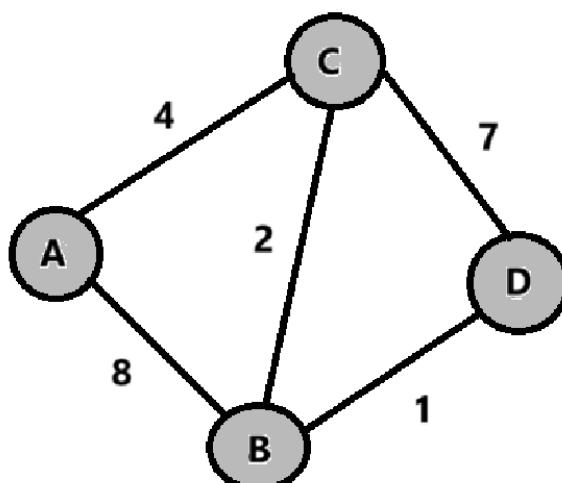


Figure 6: Dijkstra's algorithm example Step 4

## *Proposed Solution*

The graph in both cases was represented by the adjacency matrix, allocated as a 1-dimensional vector to increase efficiency. If two nodes are not directly connected, the distance is set to  $\infty$ .

### 2.1 OMP + MPI Version

Dijkstra's algorithm is parallelized thanks to the use of MPI and OpenMP.

The source process (process 0) creates the graph and divides it among the processes using *MPI\_SCATTER()*. The graph, represented by the adjacency matrix, is divided by rows, assigning to each process a number of nodes equal to the size of the graph divided by the number of processes, so the size of the graph must be divisible by the number of MPI processes.

At this point each process executes Dijkstra's algorithm on its local data as follows:

1. **initialization:** each process correctly initializes its local distance vectors, path and visited;
2. **search for the not visited vertex with minimum local distance:** each process searches for its local minimum vertex through the use of a for loop, appropriately parallelized with multiple OpenMP threads;
3. **search for the not visited vertex with minimum global distance from the source:** the processes communicate with each other using the *MPI\_ALL\_REDUCE()* function, finding the node U with the minimum global distance;
4. **update of the distances:** each process appropriately updates the distances from the node U of the not visited vertices of the graph under its responsibility, through a for loop, appropriately parallelized with multiple OpenMP threads;
5. **each process repeats step 2, 3, 4 until all the processes have visited all the vertices of their competence.**

At the end of the algorithm, the source process merge the local distance and path results using the *MPI\_GATHER()* function.

### 2.2 OMP + CUDA Version

Dijkstra's algorithm is parallelized thanks to the use of CUDA and OpenMP.

The optimal block size (768 threads) is calculated by using the function *cudaOccupancyMaxPotentialBlockSize()*.

The optimal grid size is calculated dynamically by the program, based on the size of the input data. Initially the graph is created and the distance vectors, path and visited, are appropriately initialized. Subsequently, data is transferred to the CUDA device using the *cudaMalloc()* and *cudaMemcpy()* functions.

At this point, Dijkstra's algorithm is executed by synchronizing hosts and devices as follows:

- 1 **search for the unvisited node with the minimum distance from the source:** implemented by using a function that executes a for loop, appropriately parallelized with multiple OpenMP threads;
- 2 **distance update:** the CUDA kernel is executed to update the distances of the unvisited nodes by the previously found node. Each thread updates the distances of the vertex corresponding to its threadID;
- 3 **transfer of the updated distance vector from device to host;**
- 4 **repeat steps 1,2,3 until every node has been visited.**

# Experimental Setup

## 3.1 Hardware

Main features of the used hardware are reported in Figure 7, Figure 8 and Figure 9 for what concern CPU, RAM and GPU respectively.

<pre> Architecture:          x86_64 CPU op-mode(s):       32-bit, 64-bit Address sizes:        48 bits physical, 48 bits virtual Byte Order:           Little Endian CPU(s):               16 On-line CPU(s) list: 0-15 Vendor ID:            AuthenticAMD Model name:           AMD Ryzen 7 6800HS with Radeon Graphics CPU family:           25 Model:                68 Thread(s) per core:   2 Core(s) per socket:   8 Socket(s):            1 Stepping:              1 CPU(s) scaling MHz:  28% CPU max MHz:          4785,0000 CPU min MHz:          400,0000 BogoMIPS:              6387,75 Virtualization features: Virtualization:       AMD-V Caches (sum of all): L1d:                 256 KiB (8 instances) L1i:                 256 KiB (8 instances) L2:                  4 MiB (8 instances) L3:                  16 MiB (1 instance) NUMA: NUMA node(s):         1 NUMA node0 CPU(s):    0-15 </pre>	<pre> Array Handle: 0x000F Error Information Handle: 0x0011 Total Width: 64 bits Data Width: 64 bits Size: 16 GB Form Factor: SODIMM Set: None Locator: DIMM 0 Bank Locator: P0 CHANNEL A Type: DDR5 Type Detail: Synchronous Unbuffered (Unregistered) Speed: 4800 MT/s Manufacturer: A-DATA Technology Serial Number: 00063949 Asset Tag: Not Specified Part Number: AO1V48UC8W1-BGIS Rank: 1 Configured Memory Speed: 4800 MT/s Minimum Voltage: 1.1 V Maximum Voltage: 1.1 V Configured Voltage: 1.1 V Memory Technology: DRAM Memory Operating Mode Capability: Volatile memory Firmware Version: Unknown Module Manufacturer ID: Bank 5, Hex 0xCB Module Product ID: Unknown Memory Subsystem Controller Manufacturer ID: Unknown Memory Subsystem Controller Product ID: Unknown Non-Volatile Size: None Volatile Size: 16 GB Cache Size: None Logical Size: None </pre>
--	---

Figure 7: CPU characteristics

Figure 8: RAM characteristics

<pre> +-----+   NVIDIA-SMI 535.146.02      Driver Version: 535.146.02 CUDA Version: 12.2    -----+   GPU  Name Persistence-M  Bus-Id Disp.A  Volatile Uncorr. ECC     Fan  Temp  Perf  Pwr:Usage/Cap   Memory-Usage   GPU-Util  Compute M.  0  NVIDIA GeForce RTX 3050 ... Off  00000000:01:00.0 Off   N/A     N/A  45C   P8    3W / 30W   201MiB / 4096MiB   1%     Default   +-----+ +-----+   Processes:   GPU  GI  CI   PID Type  Process name          GPU Memory     ID   ID          ID              name             Usage     -----+   0   N/A N/A  2880 G   /usr/lib/xorg/Xorg          108MiB     0   N/A N/A  3148 G   /usr/bin/gnome-shell        40MiB      0   N/A N/A  3654 G   /usr/libexec/xdg-desktop-portal-gnome 10MiB      0   N/A N/A 14542 G   ...,WinRetrieveSuggestionsOnlyOnDemand 33MiB    +-----+ </pre>
---

Figure 9: GPU characteristics

### 3.2 Software

Main features of the installed software are reported in Figure 10.



The image shows a terminal window with a light gray background and a dark gray border. Inside, there is a list of software packages and their versions, each on a new line:

- Ubuntu 23.10
- GCC 13.2.0
- Visual Studio 1.74.2
- GNU Make 4.3
- Python 3.11.6
- NVCC 12.0
- MPICC 13.2.0

*Figure 10: Software*

## Speedup, Performance & Efficiency

The tests were carried out with different compiler optimization (-O0, -O1, -O2, -O3), therefore, both parallel and sequential versions will be compiled with the corresponding optimization.

For each test, there are four different types of graph:

0. *Fully connected graph*, each node is connected to all the other nodes
1. *Dense graph*, a node has a 10% chance of not being connected to another node
2. *Medium dense graph*, a node has a 30% chance of not being connected to another node
3. *Sparse graph*, a node has a 50% chance of not being connected to another node

For each type of graph, 3 size configurations were tested. The nodes will therefore depend on size of the graph, while the edges will depend on the size and type of the graph.

Graph size configurations are:

- Small => 1536 nodes
- Medium => 7552 nodes
- Big => 13568 nodes

Graphs were randomly created, starting from a chosen seed, contained in the Makefile.

Test performance figures are speedup ( $S_n$ ) and efficiency ( $E_n$ ), evaluated such as:

$$S_n = \frac{T_1}{T_p(n)} \quad E_n = \frac{S_n}{n}$$

where  $T_1$  is the execution time of the serial algorithm and  $T_p(n)$  is the execution time of  $n$ -processor parallel algorithm.

## Test “OpenMP + MPI”

Tests were performed by iterating 15 trial for each configuration.

Configuration {8 OpenMP threads, 4 MPI processes} was removed due to its high execution time (speedup set to 0). In the following are reported test results.

### 5.1 Optimization 0

Optimization 0 introduces an improvement in speedup and performance.

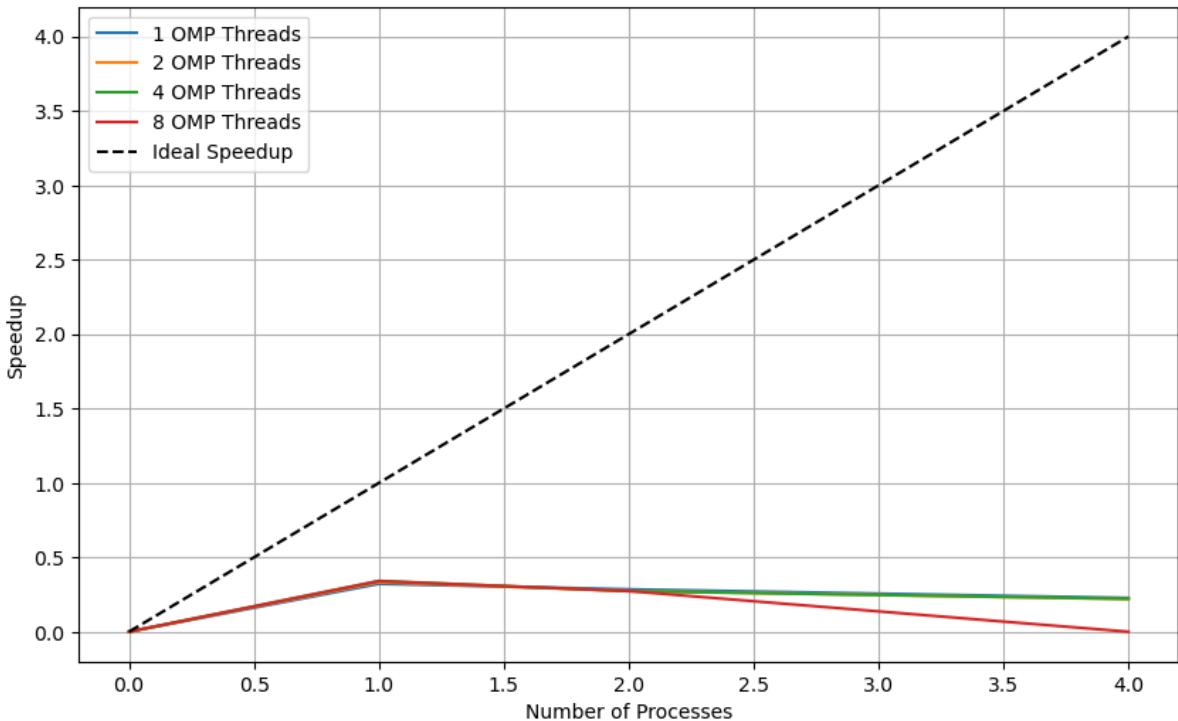
The algorithm remains efficient for large and mainly for medium, type 3, graphs.

The improvement in speedup is proportional to the size of the graph

#### 5.1.1 Type 0

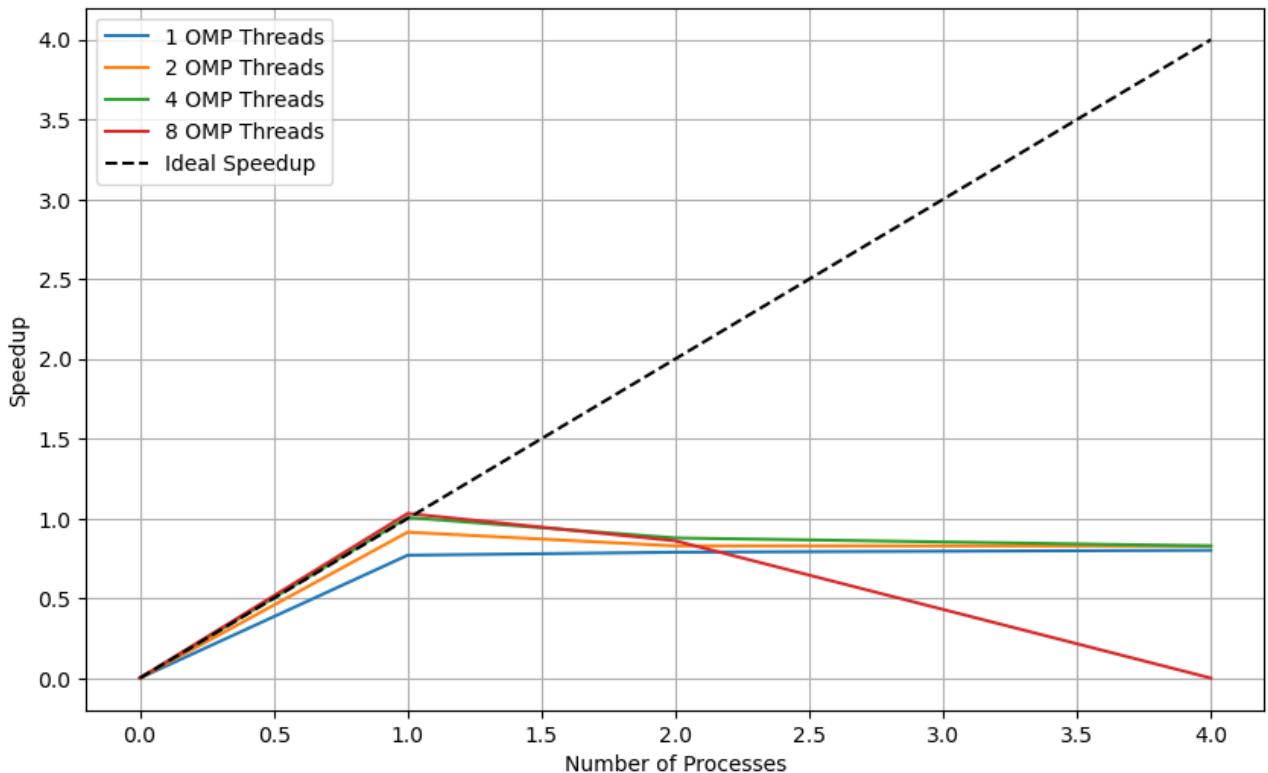
**1536 nodes & 2357760 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0270541	0.0203475	0.0000000	0.0066809	1.000000	1.000000
OpenMP+MPI	1	1	0.0841001	0.0193927	0.0000000	0.0092657	0.3216890	0.3216890
OpenMP+MPI	1	2	0.0948441	0.0195345	0.0036887	0.0067355	0.2852477	0.1426238
OpenMP+MPI	1	4	0.1190747	0.0204197	0.0042030	0.0053709	0.2272024	0.0568006
OpenMP+MPI	2	1	0.0813671	0.0195420	0.0000000	0.0068568	0.3324941	0.1662470
OpenMP+MPI	2	2	0.0985249	0.0200130	0.0036767	0.0079496	0.2745911	0.0686478
OpenMP+MPI	2	4	0.1231158	0.0201932	0.0042690	0.0082125	0.2197449	0.0274681
OpenMP+MPI	4	1	0.0791917	0.0195904	0.0000000	0.0057821	0.3416274	0.0854069
OpenMP+MPI	4	2	0.0991746	0.0205729	0.0036936	0.0070719	0.2727923	0.0340990
OpenMP+MPI	4	4	0.1222721	0.0208871	0.0042189	0.0101959	0.2212611	0.0138288
OpenMP+MPI	8	1	0.0795793	0.0193597	0.0000000	0.0064358	0.3399638	0.0424955
OpenMP+MPI	8	2	0.0990011	0.0206575	0.0037888	0.0086265	0.2732705	0.0170794



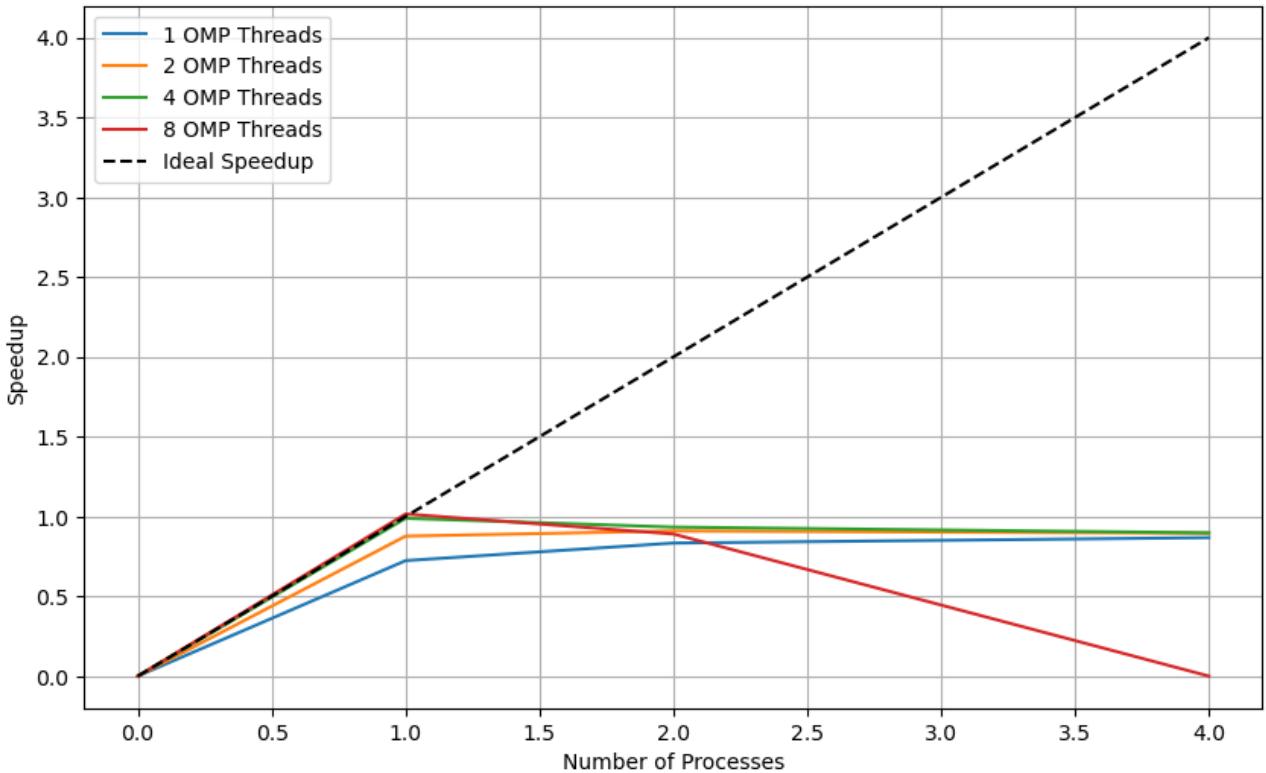
## 7552 nodes & 57025152 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6548891	0.4969229	0.0000000	0.1579379	1.0000000	1.0000000
OpenMP+MPI	1	1	0.8507321	0.5025238	0.0000000	0.2948941	0.7697947	0.7697947
OpenMP+MPI	1	2	0.8303115	0.5085971	0.0953365	0.1603664	0.7887269	0.3943635
OpenMP+MPI	1	4	0.8176739	0.5137159	0.1056791	0.0997740	0.8009172	0.2002293
OpenMP+MPI	2	1	0.7162577	0.5027029	0.0000000	0.1595358	0.9143205	0.4571602
OpenMP+MPI	2	2	0.7907951	0.5257395	0.0950565	0.1036828	0.8281401	0.2070350
OpenMP+MPI	2	4	0.7912217	0.5159429	0.1055231	0.0832156	0.8276935	0.1034617
OpenMP+MPI	4	1	0.6508945	0.4993481	0.0000000	0.0975758	1.0061371	0.2515343
OpenMP+MPI	4	2	0.7457323	0.5054975	0.0949983	0.0786034	0.8781826	0.1097728
OpenMP+MPI	4	4	0.7907723	0.5141616	0.1058398	0.0777124	0.8281639	0.0517602
OpenMP+MPI	8	1	0.6349216	0.4996676	0.0000000	0.0812865	1.0314487	0.1289311
OpenMP+MPI	8	2	0.7614405	0.5151544	0.0949045	0.0855263	0.8600659	0.0537541



## 13568 nodes & 184077056 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.6802869	2.1415253	0.0000000	0.5387284	1.0000000	1.0000000
OpenMP+MPI	1	1	3.7047255	2.1542035	0.0000000	1.4963909	0.7234779	0.7234779
OpenMP+MPI	1	2	3.2152015	2.1560489	0.3033133	0.6879717	0.8336295	0.4168148
OpenMP+MPI	1	4	3.0896087	2.1704198	0.3376694	0.4824395	0.8675166	0.2168792
OpenMP+MPI	2	1	3.0561888	2.1530039	0.0000000	0.8483187	0.8770030	0.4385015
OpenMP+MPI	2	2	2.9475173	2.1422331	0.3040777	0.4369043	0.9093371	0.2273343
OpenMP+MPI	2	4	2.9944417	2.1555007	0.3373080	0.4095719	0.8950873	0.1118859
OpenMP+MPI	4	1	2.7112802	2.1247153	0.0000000	0.5313888	0.9885687	0.2471422
OpenMP+MPI	4	2	2.8706310	2.1387759	0.3027350	0.3639065	0.9336926	0.1167116
OpenMP+MPI	4	4	2.9859198	2.1492626	0.3365357	0.4033911	0.8976419	0.0561026
OpenMP+MPI	8	1	2.6386862	2.1247294	0.0000000	0.4596869	1.0157657	0.1269707
OpenMP+MPI	8	2	3.0103323	2.1581047	0.3022055	0.4832614	0.8903625	0.0556477

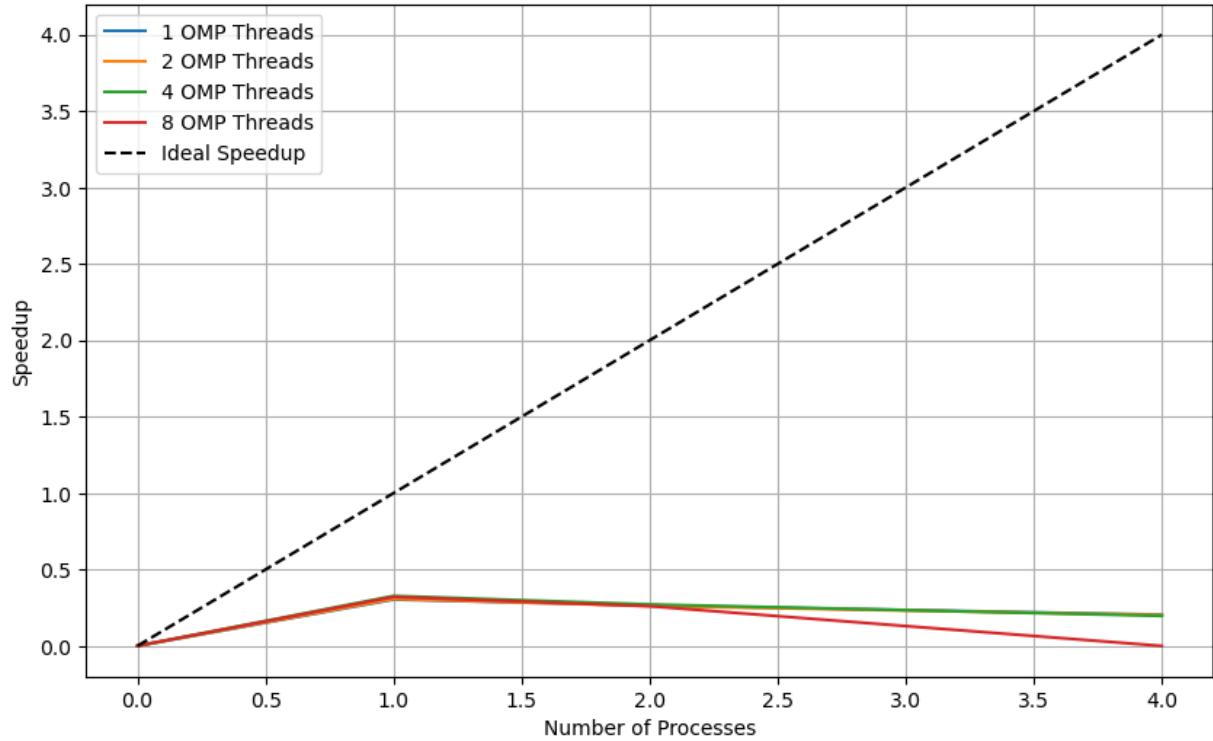


For optimization 0, the algorithm does not behave optimally for the type 0 (fully connected) graph. However, we see an improvement in speedup proportional to the size of the graph

### 5.1.2 Type 1

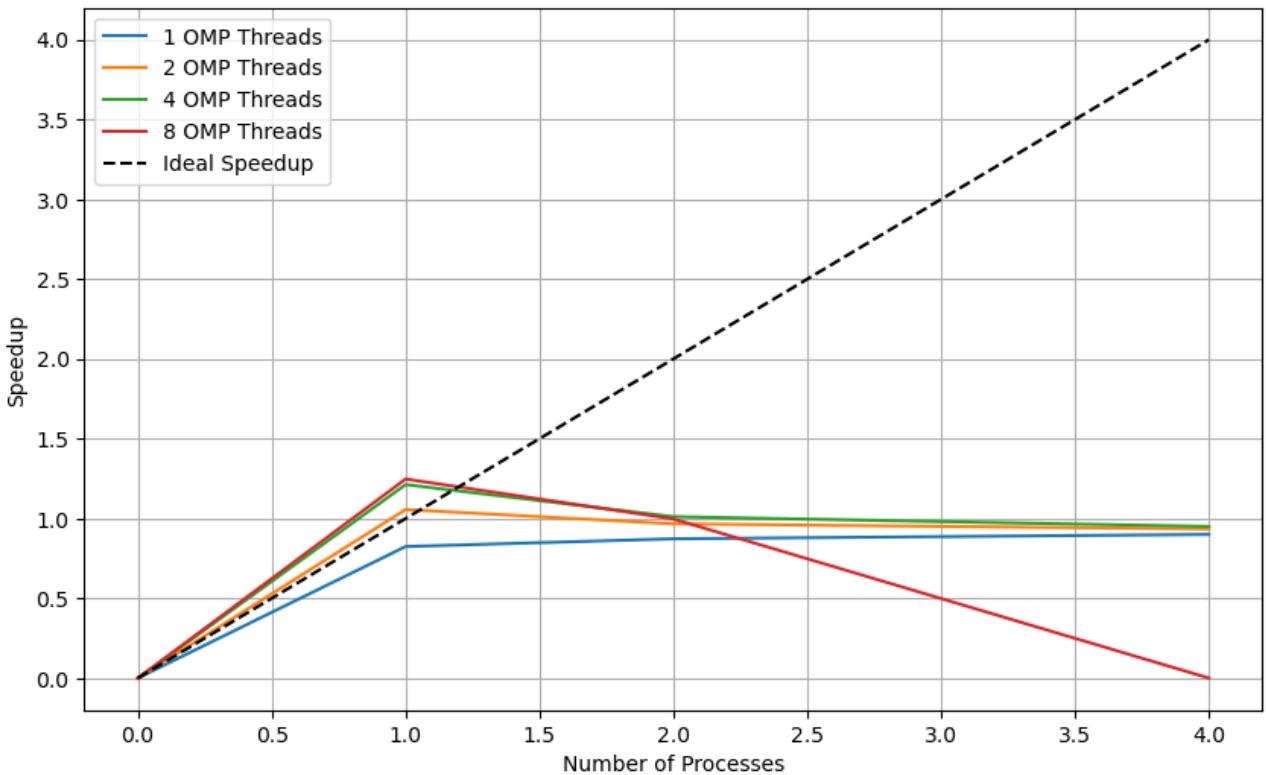
**1536 nodes & 1061043 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0240650	0.0146171	0.0000000	0.0094213	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0793121	0.0137689	0.0000000	0.0113133	0.3034214	0.3034214
OpenMP+MPI	1	2	0.0907560	0.0142409	0.0037261	0.0076716	0.2651615	0.1325808
OpenMP+MPI	1	4	0.1177399	0.0144501	0.0042413	0.0056496	0.2043911	0.0510978
OpenMP+MPI	2	1	0.0782933	0.0137464	0.0000000	0.0077027	0.3073697	0.1536849
OpenMP+MPI	2	2	0.0920257	0.0144269	0.0037397	0.0077565	0.2615029	0.0653757
OpenMP+MPI	2	4	0.1188514	0.0140737	0.0043093	0.0084627	0.2024797	0.0253100
OpenMP+MPI	4	1	0.0735227	0.0136970	0.0000000	0.0060320	0.3273137	0.0818284
OpenMP+MPI	4	2	0.0885794	0.0137779	0.0037093	0.0064880	0.2716772	0.0339596
OpenMP+MPI	4	4	0.1225371	0.0142485	0.0042255	0.0095939	0.1963896	0.0122743
OpenMP+MPI	8	1	0.0749299	0.0136593	0.0000000	0.0065634	0.3211667	0.0401458
OpenMP+MPI	8	2	0.0926258	0.0137749	0.0037162	0.0094895	0.2598088	0.0162381



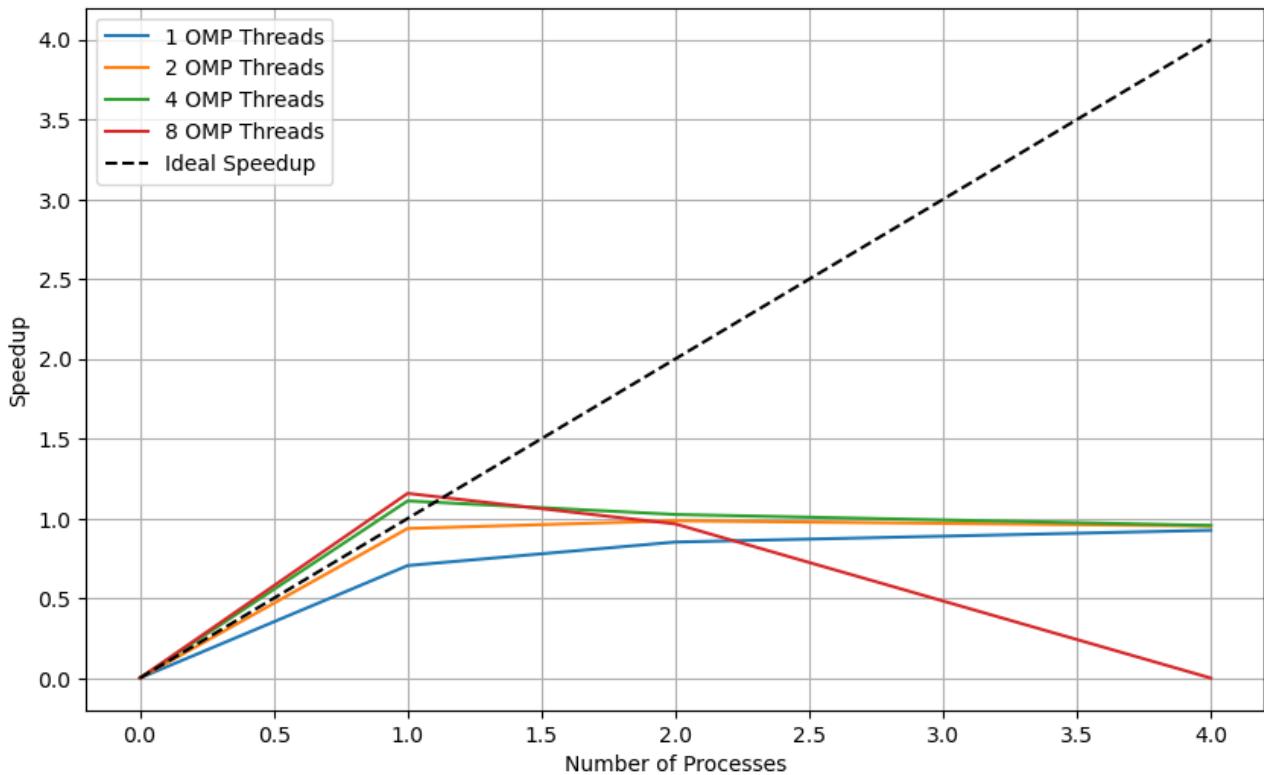
## 7552 nodes & 25660141 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5992059	0.3400475	0.0000000	0.2591287	1.0000000	1.0000000
OpenMP+MPI	1	1	0.7269637	0.3441461	0.0000000	0.3293019	0.8242583	0.8242583
OpenMP+MPI	1	2	0.6865475	0.3468319	0.0953091	0.1792627	0.8727813	0.4363907
OpenMP+MPI	1	4	0.6656549	0.3518957	0.1056450	0.1121299	0.9001749	0.2250437
OpenMP+MPI	2	1	0.5674118	0.3414113	0.0000000	0.1719181	1.0560335	0.5280167
OpenMP+MPI	2	2	0.6199441	0.3460751	0.0951940	0.1122251	0.9665483	0.2416371
OpenMP+MPI	2	4	0.6413372	0.3522551	0.1054095	0.0912317	0.9343070	0.1167884
OpenMP+MPI	4	1	0.4942180	0.3408578	0.0000000	0.0997816	1.2124323	0.3031081
OpenMP+MPI	4	2	0.5924751	0.3469672	0.0948267	0.0844321	1.0113604	0.1264200
OpenMP+MPI	4	4	0.6308072	0.3553954	0.1057527	0.0820883	0.9499033	0.0593690
OpenMP+MPI	8	1	0.4803667	0.3427179	0.0000000	0.0827735	1.2473925	0.1559241
OpenMP+MPI	8	2	0.6010401	0.3498853	0.0949376	0.0881920	0.9969482	0.0623093



## 13568 nodes & 82829469 edges

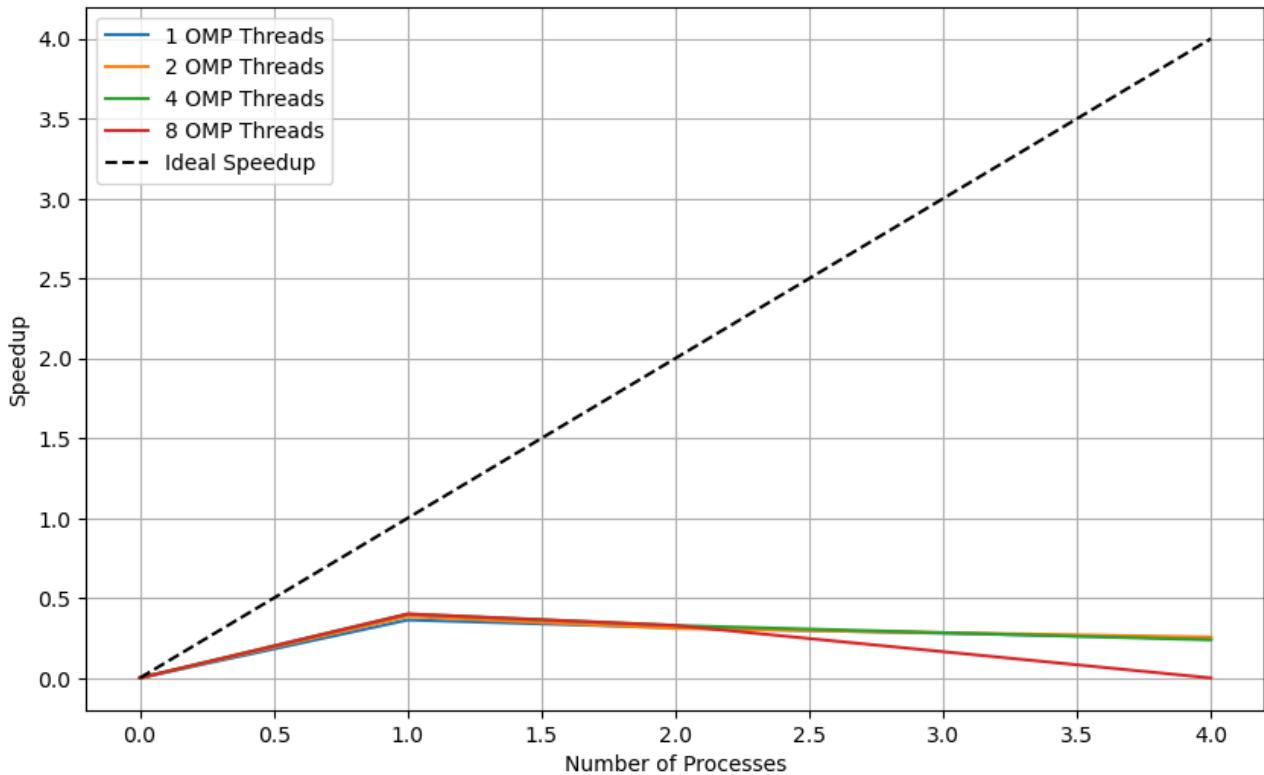
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.9658537	1.1683563	0.0000000	0.7974657	1.0000000	1.0000000
OpenMP+MPI	1	1	2.7884087	1.1728869	0.0000000	1.5609587	0.7050091	0.7050091
OpenMP+MPI	1	2	2.3059703	1.1858371	0.3060427	0.7452509	0.8525061	0.4262530
OpenMP+MPI	1	4	2.1229194	1.2049473	0.3373836	0.4861430	0.9260143	0.2315036
OpenMP+MPI	2	1	2.0968817	1.1792661	0.0000000	0.8624164	0.9375129	0.4687565
OpenMP+MPI	2	2	1.9945116	1.1806629	0.3037301	0.4452869	0.9856316	0.2464079
OpenMP+MPI	2	4	2.0625591	1.2161868	0.3376595	0.4153535	0.9531139	0.1191392
OpenMP+MPI	4	1	1.7707266	1.1673658	0.0000000	0.5488146	1.1101960	0.2775490
OpenMP+MPI	4	2	1.9174891	1.1839152	0.3035590	0.3635081	1.0252228	0.1281529
OpenMP+MPI	4	4	2.0552426	1.1994880	0.3372609	0.4228558	0.9565069	0.0597817
OpenMP+MPI	8	1	1.6985117	1.1654459	0.0000000	0.4781137	1.1573978	0.1446747
OpenMP+MPI	8	2	2.0354109	1.1854878	0.3037449	0.4804652	0.9658264	0.0603642



### 5.1.3 Type 2

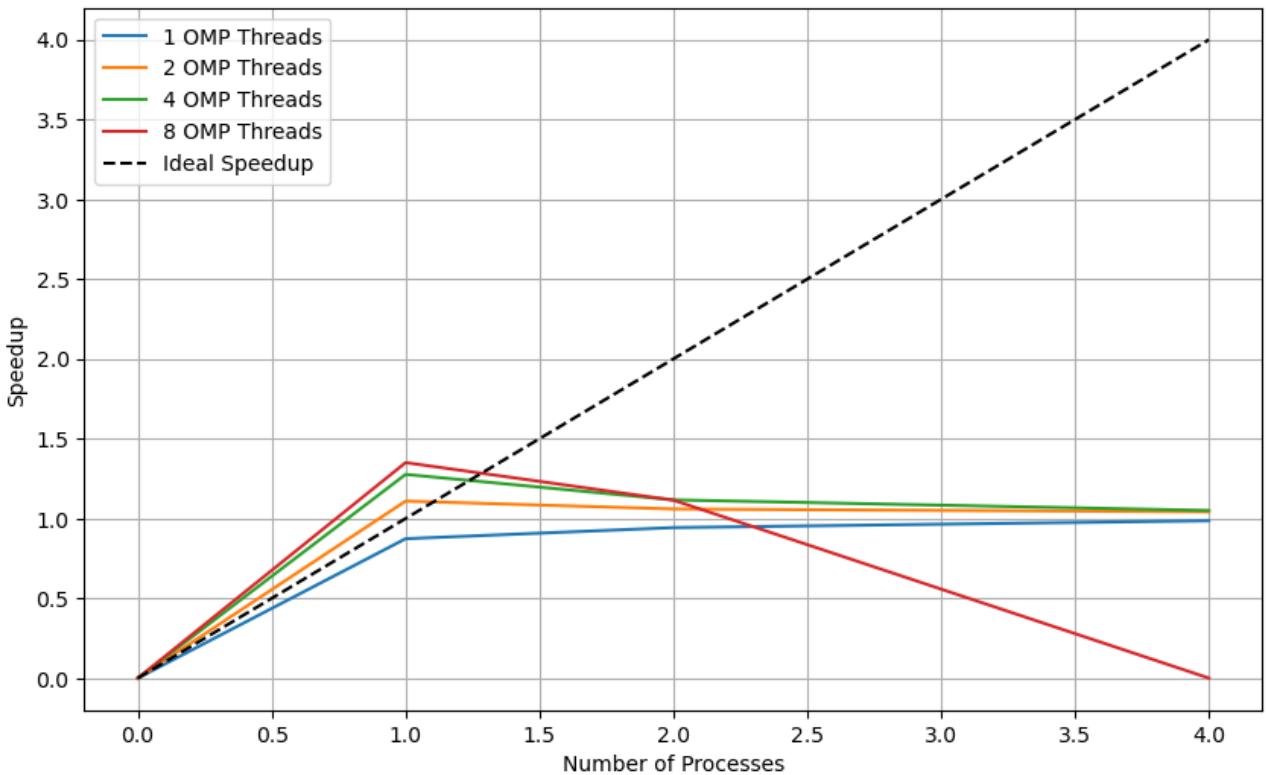
**1536 nodes & 790416 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0309549	0.0178296	0.0000000	0.0130994	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0853255	0.0170713	0.0000000	0.0146121	0.3627866	0.3627866
OpenMP+MPI	1	2	0.0983141	0.0174545	0.0036268	0.0092580	0.3148574	0.1574287
OpenMP+MPI	1	4	0.1259769	0.0173691	0.0041028	0.0060897	0.2457191	0.0614298
OpenMP+MPI	2	1	0.0803403	0.0171934	0.0000000	0.0093779	0.3852979	0.1926489
OpenMP+MPI	2	2	0.0998000	0.0174921	0.0037449	0.0088537	0.3101697	0.0775424
OpenMP+MPI	2	4	0.1208994	0.0176009	0.0041701	0.0087522	0.2560388	0.0320048
OpenMP+MPI	4	1	0.0773965	0.0171213	0.0000000	0.0068307	0.3999525	0.0999881
OpenMP+MPI	4	2	0.0942457	0.0172639	0.0036815	0.0065572	0.3284494	0.0410562
OpenMP+MPI	4	4	0.1300249	0.0174277	0.0041696	0.0109553	0.2380693	0.0148793
OpenMP+MPI	8	1	0.0773015	0.0170613	0.0000000	0.0066773	0.4004440	0.0500555
OpenMP+MPI	8	2	0.0941343	0.0171493	0.0036791	0.0085937	0.3288379	0.0205524



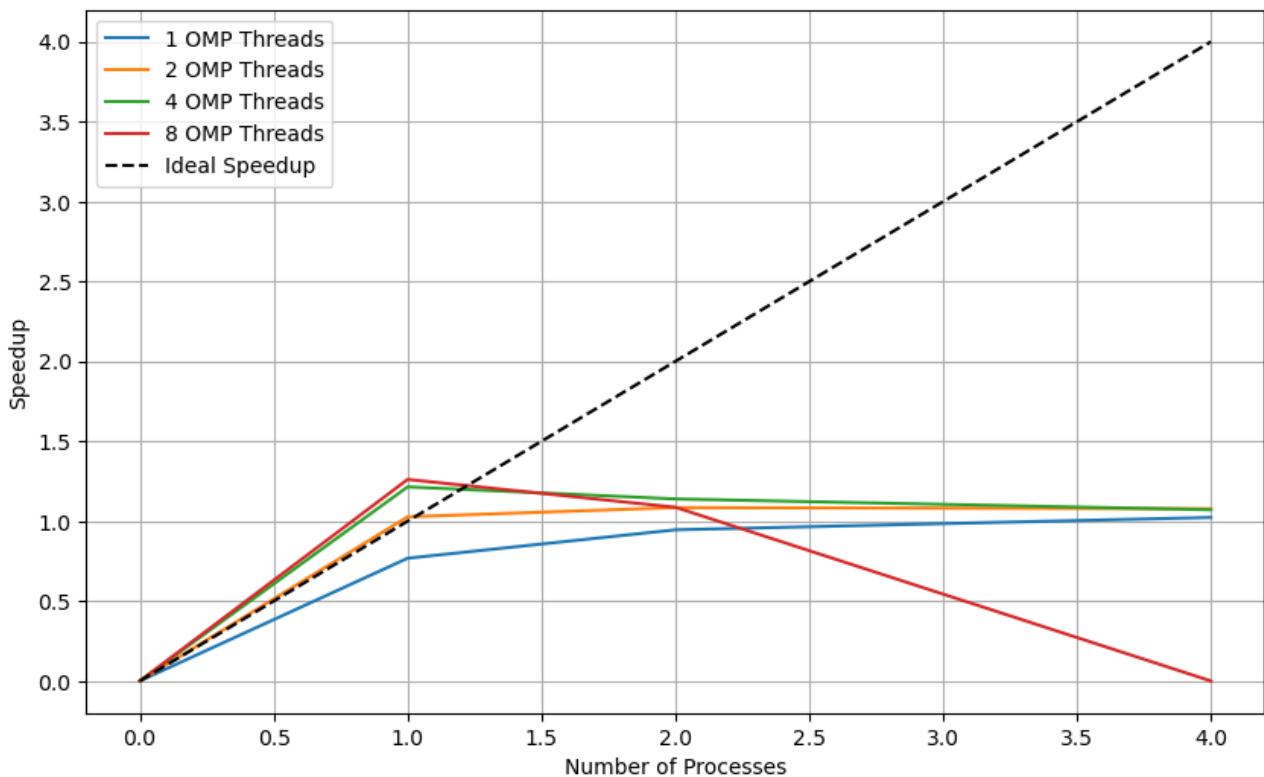
## 7552 nodes & 19101950 edges

	OMP	MPI	total time	graph creation time	comunication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.7584459	0.4191075	0.0000000	0.3393106	1.0000000	1.0000000
OpenMP+MPI	1	1	0.8684495	0.4187909	0.0000000	0.3963865	0.8733333	0.8733333
OpenMP+MPI	1	2	0.8044054	0.4264597	0.0953183	0.2134796	0.9428652	0.4714326
OpenMP+MPI	1	4	0.7693310	0.4320033	0.1056601	0.1309431	0.9858512	0.2464628
OpenMP+MPI	2	1	0.6838785	0.4219109	0.0000000	0.2079399	1.1090360	0.5545180
OpenMP+MPI	2	2	0.7157949	0.4253454	0.0955572	0.1298287	1.0595854	0.2648964
OpenMP+MPI	2	4	0.7280533	0.4324734	0.1055941	0.0989907	1.0417450	0.1302181
OpenMP+MPI	4	1	0.5942024	0.4189669	0.0000000	0.1211483	1.2764100	0.3191025
OpenMP+MPI	4	2	0.6791906	0.4273860	0.0951969	0.0912430	1.1166908	0.1395863
OpenMP+MPI	4	4	0.7221993	0.4376011	0.1058231	0.0868529	1.0501891	0.0656368
OpenMP+MPI	8	1	0.5616820	0.4208185	0.0000000	0.0863795	1.3503119	0.1687890
OpenMP+MPI	8	2	0.6802476	0.4261659	0.0953055	0.0909648	1.1149556	0.0696847



## 13568 nodes & 61665181 edges

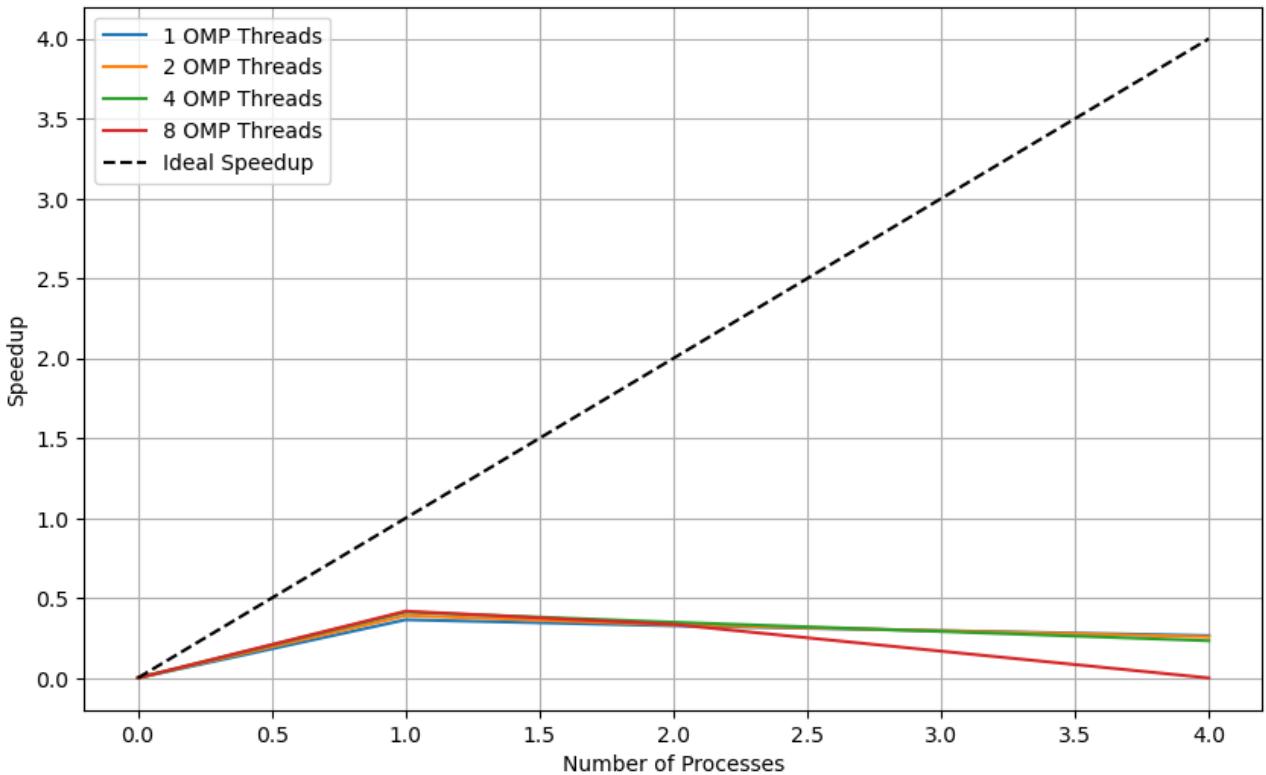
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.4746373	1.3941055	0.0000000	1.0805001	1.0000000	1.0000000
OpenMP+MPI	1	1	3.2229312	1.3998538	0.0000000	1.7685077	0.7678220	0.7678220
OpenMP+MPI	1	2	2.6163337	1.4238229	0.3040148	0.8189333	0.9458416	0.4729208
OpenMP+MPI	1	4	2.4184803	1.4574821	0.3389127	0.5262397	1.0232200	0.2558050
OpenMP+MPI	2	1	2.4089505	1.4085193	0.0000000	0.9453609	1.0272678	0.5136339
OpenMP+MPI	2	2	2.2812505	1.4274256	0.3034838	0.4862725	1.0847723	0.2711931
OpenMP+MPI	2	4	2.2966897	1.4543847	0.3389525	0.4137972	1.0774801	0.1346850
OpenMP+MPI	4	1	2.0376107	1.3969263	0.0000000	0.5859528	1.2144800	0.3036200
OpenMP+MPI	4	2	2.1727049	1.4254140	0.3050873	0.3774627	1.1389662	0.1423708
OpenMP+MPI	4	4	2.3097322	1.4510015	0.3384313	0.4259320	1.0713958	0.0669622
OpenMP+MPI	8	1	1.9623988	1.3999926	0.0000000	0.5074037	1.2610267	0.1576283
OpenMP+MPI	8	2	2.2768104	1.4265813	0.3039819	0.4787468	1.0868878	0.0679305



### 5.1.4 Type 3

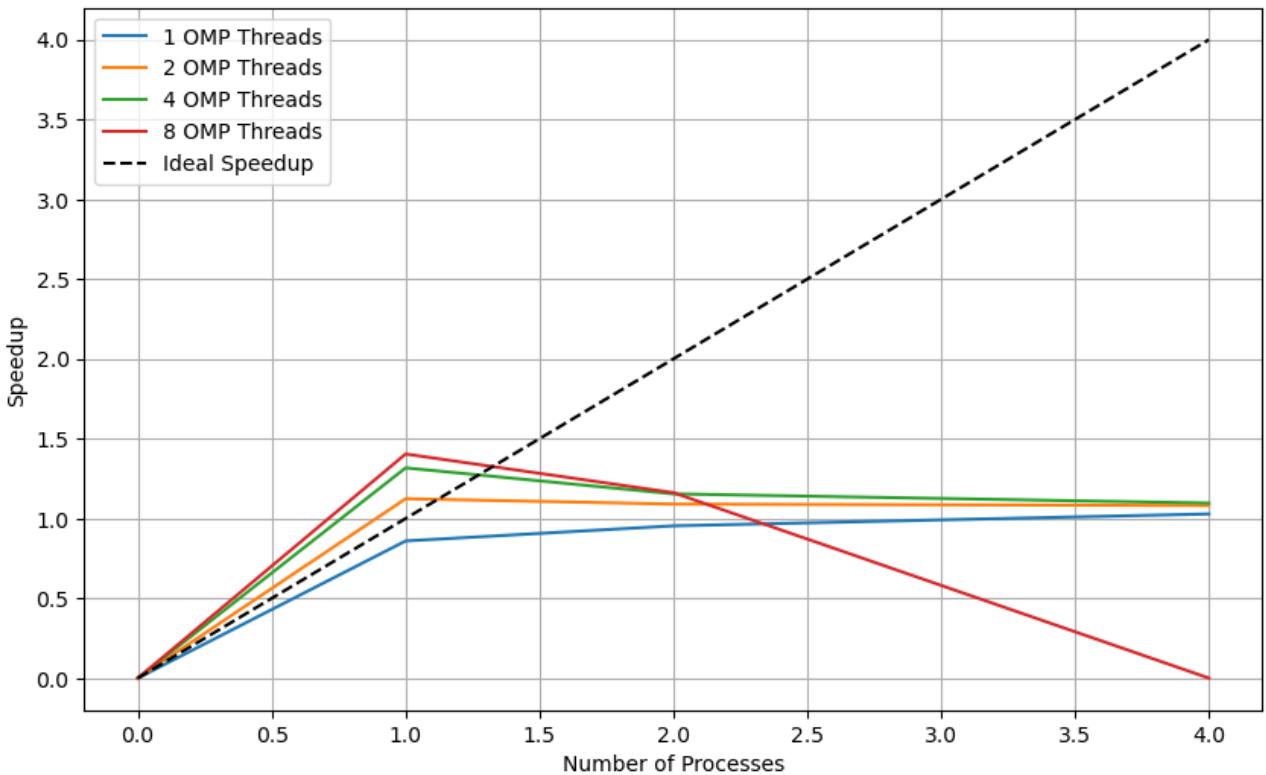
**1536 nodes & 588142 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0321457	0.0173455	0.0000000	0.0147734	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0882415	0.0167231	0.0000000	0.0166271	0.3642918	0.3642918
OpenMP+MPI	1	2	0.0983525	0.0171163	0.0036580	0.0101056	0.3268413	0.1634206
OpenMP+MPI	1	4	0.1213705	0.0177507	0.0041391	0.0068377	0.2648558	0.0662139
OpenMP+MPI	2	1	0.0823833	0.0167077	0.0000000	0.0104101	0.3901966	0.1950983
OpenMP+MPI	2	2	0.0964832	0.0172102	0.0037350	0.0081522	0.3331737	0.0832934
OpenMP+MPI	2	4	0.1247360	0.0174185	0.0041529	0.0099802	0.2577096	0.0322137
OpenMP+MPI	4	1	0.0780518	0.0168205	0.0000000	0.0071755	0.4118504	0.1029626
OpenMP+MPI	4	2	0.0921711	0.0168916	0.0037451	0.0071465	0.3487607	0.0435951
OpenMP+MPI	4	4	0.1379129	0.0172322	0.0041739	0.0131221	0.2330867	0.0145679
OpenMP+MPI	8	1	0.0769360	0.0167107	0.0000000	0.0068448	0.4178235	0.0522279
OpenMP+MPI	8	2	0.0958849	0.0169046	0.0037403	0.0088348	0.3352528	0.0209533



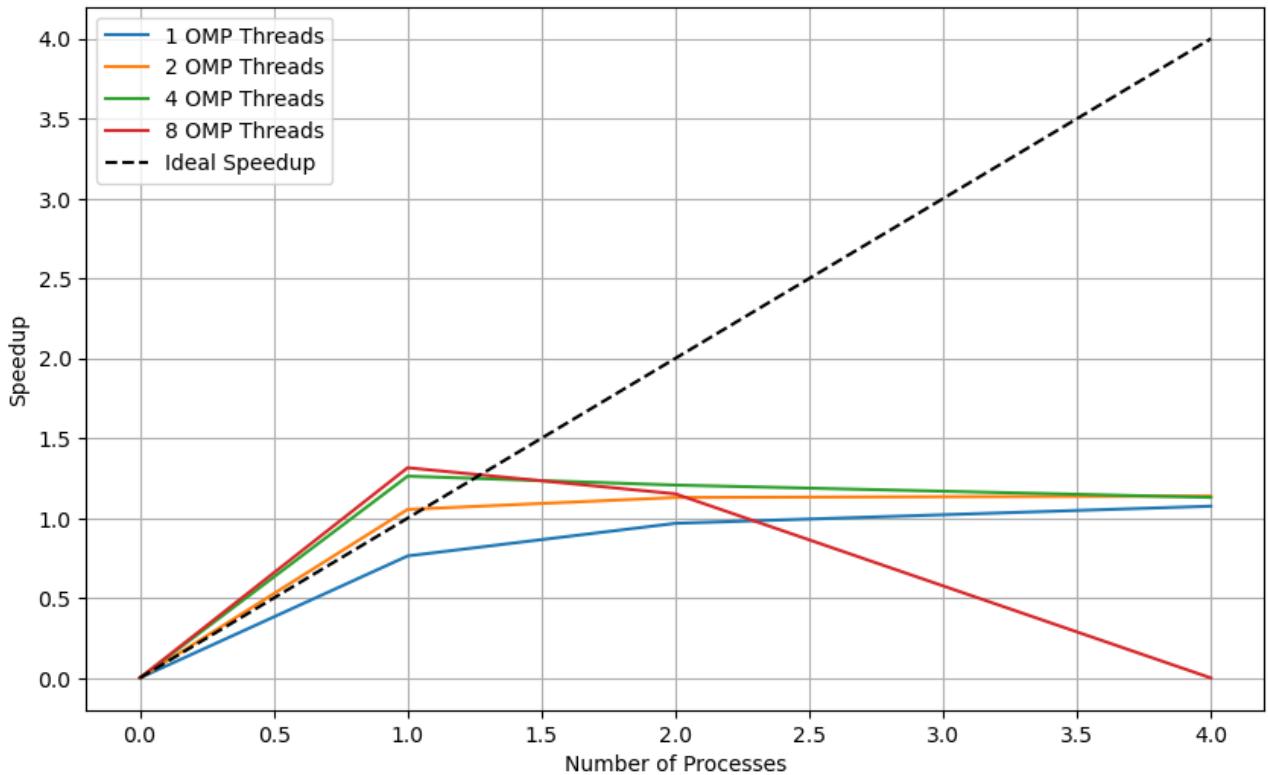
## 7552 nodes & 14260151 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.7811967	0.4102399	0.0000000	0.3709271	1.0000000	1.0000000
OpenMP+MPI	1	1	0.9091207	0.4099593	0.0000000	0.4450473	0.8592881	0.8592881
OpenMP+MPI	1	2	0.8189321	0.4206384	0.0949332	0.2354323	0.9539211	0.4769606
OpenMP+MPI	1	4	0.7595827	0.4232249	0.1062719	0.1373413	1.0284551	0.2571138
OpenMP+MPI	2	1	0.6951073	0.4094158	0.0000000	0.2315506	1.1238504	0.5619252
OpenMP+MPI	2	2	0.7163508	0.4157763	0.0956253	0.1378860	1.0905225	0.2726306
OpenMP+MPI	2	4	0.7227239	0.4288995	0.1057963	0.0978074	1.0809060	0.1351133
OpenMP+MPI	4	1	0.5930361	0.4085220	0.0000000	0.1297865	1.3172834	0.3293209
OpenMP+MPI	4	2	0.6766982	0.4166373	0.0954337	0.0984371	1.1544240	0.1443030
OpenMP+MPI	4	4	0.7123672	0.4239079	0.1060713	0.0874747	1.0966208	0.0685388
OpenMP+MPI	8	1	0.5563342	0.4097844	0.0000000	0.0923547	1.4041859	0.1755232
OpenMP+MPI	8	2	0.6724503	0.4174138	0.0957846	0.0929682	1.1617165	0.0726073



## 13568 nodes & 46014096 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.5986958	1.3947622	0.0000000	1.2039047	1.0000000	1.0000000
OpenMP+MPI	1	1	3.4036157	1.4118608	0.0000000	1.9375965	0.7635103	0.7635103
OpenMP+MPI	1	2	2.6857998	1.4198601	0.3031659	0.8934058	0.9675687	0.4837843
OpenMP+MPI	1	4	2.4175603	1.4386200	0.3371469	0.5473838	1.0749249	0.2687312
OpenMP+MPI	2	1	2.4640563	1.3878730	0.0000000	1.0216221	1.0546414	0.5273207
OpenMP+MPI	2	2	2.3014435	1.4136958	0.3029332	0.5182309	1.1291591	0.2822898
OpenMP+MPI	2	4	2.2824307	1.4351779	0.3373499	0.4197076	1.1385651	0.1423206
OpenMP+MPI	4	1	2.0579924	1.3852776	0.0000000	0.6180151	1.2627334	0.3156834
OpenMP+MPI	4	2	2.1529773	1.4105317	0.3030449	0.3733866	1.2070243	0.1508780
OpenMP+MPI	4	4	2.2997773	1.4448543	0.3405648	0.4248316	1.1299771	0.0706236
OpenMP+MPI	8	1	1.9754164	1.4069393	0.0000000	0.5135639	1.3155180	0.1644397
OpenMP+MPI	8	2	2.2554788	1.4083175	0.3032299	0.4778838	1.1521704	0.0720106



## 5.2 Optimization 1

Optimization 1 does not introduce an improvement in speedup and performance compared to O0.

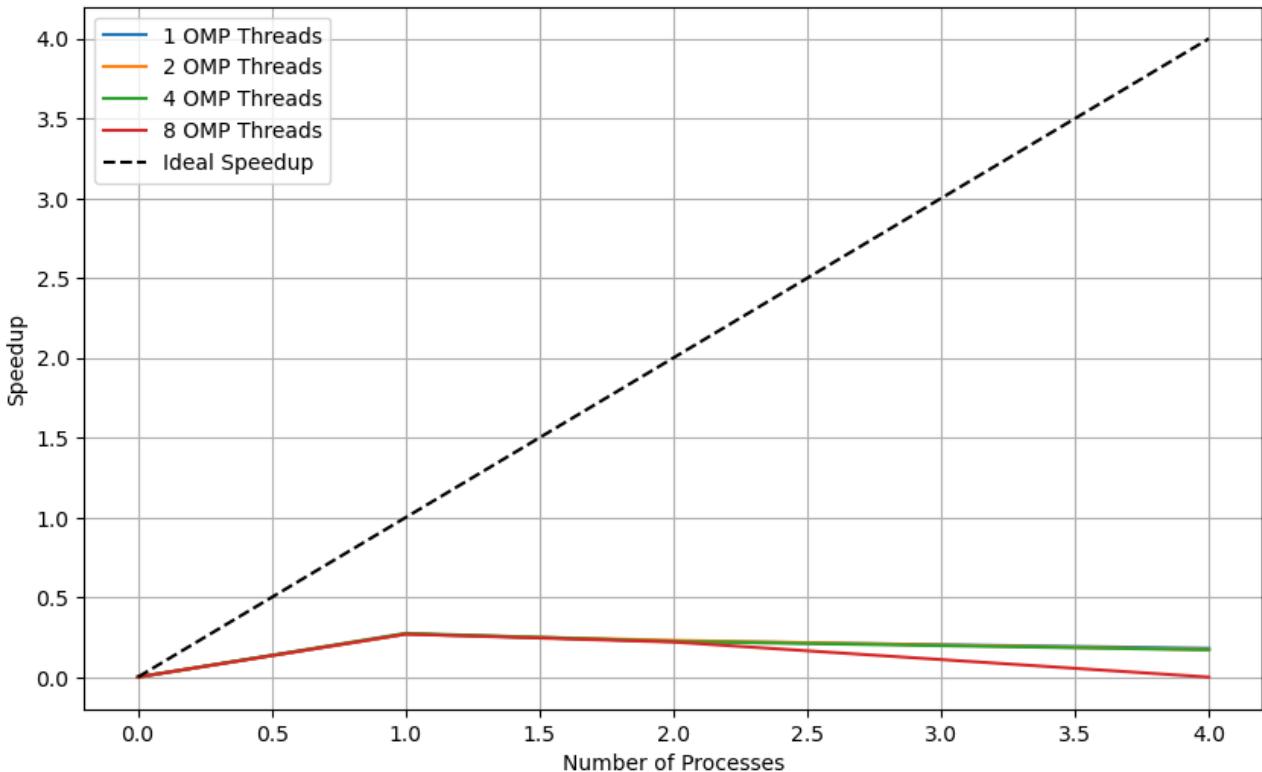
The algorithm remains efficient for large and mainly for medium, type 3, graphs.

Optimization 0 remains the one with the best performance

### 5.2.1 Type 0

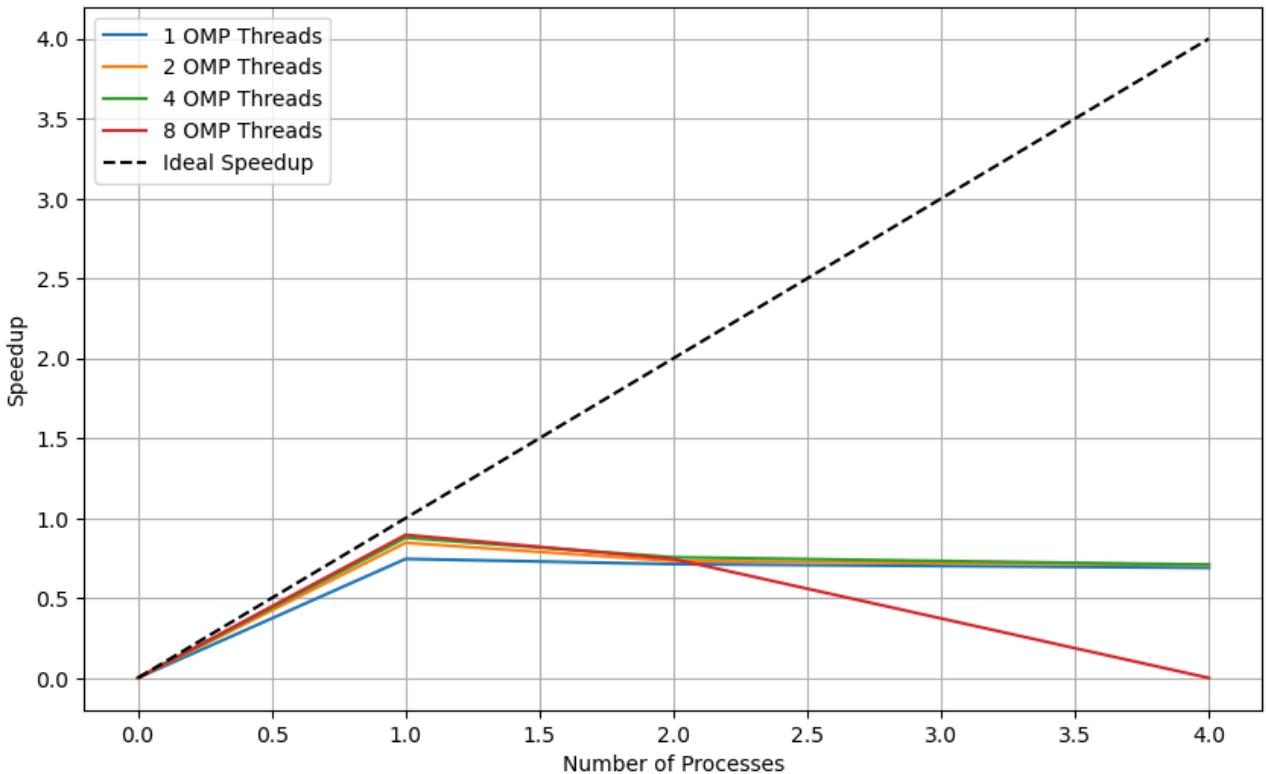
**1536 nodes & 2357760 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0212619	0.0185059	0.0000000	0.0027288	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0787853	0.0173029	0.0000000	0.0064082	0.2698709	0.2698709
OpenMP+MPI	1	2	0.0941003	0.0171694	0.0037445	0.0050026	0.2259491	0.1129745
OpenMP+MPI	1	4	0.1179080	0.0174109	0.0042829	0.0046580	0.1803259	0.0450815
OpenMP+MPI	2	1	0.0789691	0.0173802	0.0000000	0.0052928	0.2692428	0.1346214
OpenMP+MPI	2	2	0.0924380	0.0179101	0.0038357	0.0056426	0.2300122	0.0575030
OpenMP+MPI	2	4	0.1232693	0.0197612	0.0044604	0.0086715	0.1724831	0.0215604
OpenMP+MPI	4	1	0.0774712	0.0173133	0.0000000	0.0048825	0.2744487	0.0686122
OpenMP+MPI	4	2	0.0952523	0.0176989	0.0038207	0.0055877	0.2232162	0.0279020
OpenMP+MPI	4	4	0.1246047	0.0181302	0.0043885	0.0092781	0.1706346	0.0106647
OpenMP+MPI	8	1	0.0788689	0.0172613	0.0000000	0.0060081	0.2695850	0.0336981
OpenMP+MPI	8	2	0.0968061	0.0175693	0.0037852	0.0090761	0.2196336	0.0137271



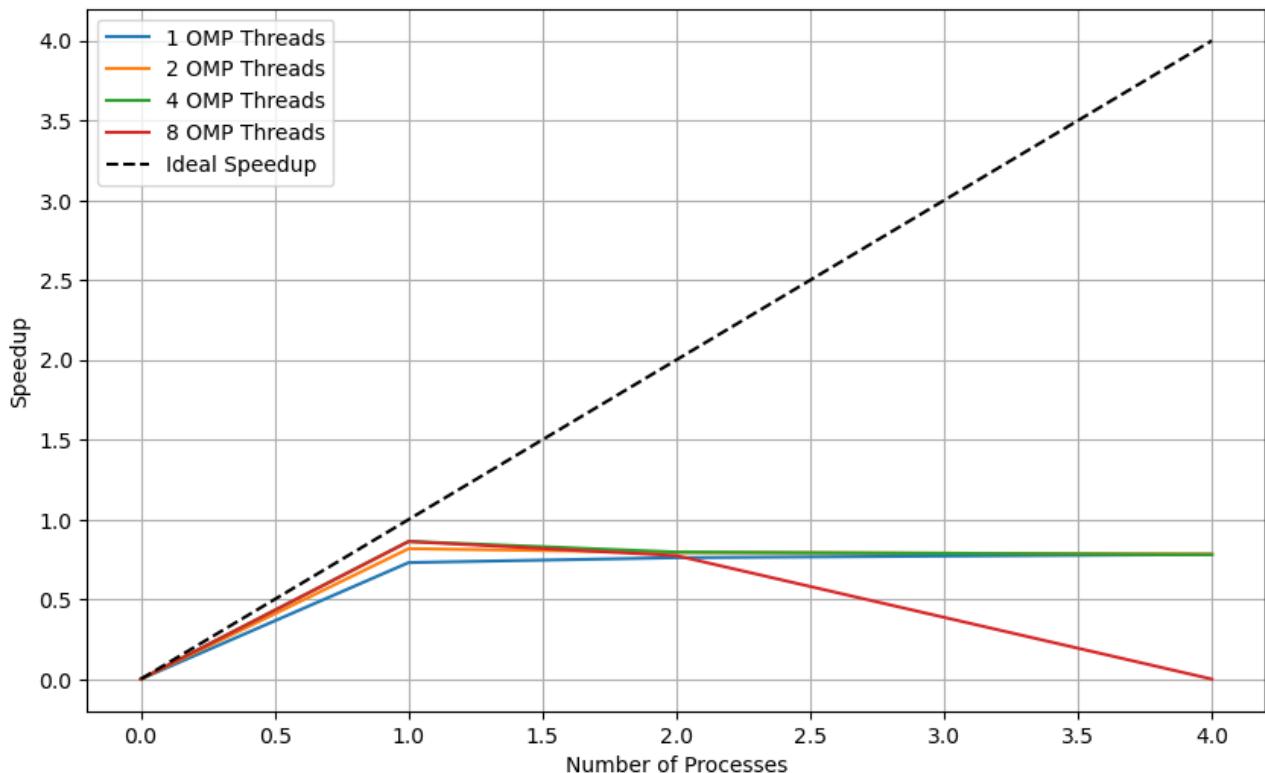
## 7552 nodes & 57025152 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5212525	0.4519587	0.0000000	0.0692682	1.0000000	1.0000000
OpenMP+MPI	1	1	0.6995416	0.4498085	0.0000000	0.1955861	0.7451344	0.7451344
OpenMP+MPI	1	2	0.7317999	0.4577298	0.0956071	0.1112601	0.7122884	0.3561442
OpenMP+MPI	1	4	0.7564347	0.4636719	0.1058869	0.0896763	0.6890912	0.1722728
OpenMP+MPI	2	1	0.6162607	0.4508173	0.0000000	0.1107643	0.8458313	0.4229156
OpenMP+MPI	2	2	0.7118403	0.4572571	0.0957257	0.0921904	0.7322605	0.1830651
OpenMP+MPI	2	4	0.7365204	0.4660580	0.1066498	0.0692531	0.7077231	0.0884654
OpenMP+MPI	4	1	0.5941185	0.4486477	0.0000000	0.0902741	0.8773544	0.2193386
OpenMP+MPI	4	2	0.6899495	0.4592999	0.0955249	0.0701563	0.7554938	0.0944367
OpenMP+MPI	4	4	0.7354444	0.4656551	0.1055903	0.0713955	0.7087586	0.0442974
OpenMP+MPI	8	1	0.5820908	0.4493738	0.0000000	0.0775861	0.8954832	0.1119354
OpenMP+MPI	8	2	0.6998659	0.4588737	0.0951917	0.0776415	0.7447892	0.0465493



## 13568 nodes & 184077056 edges

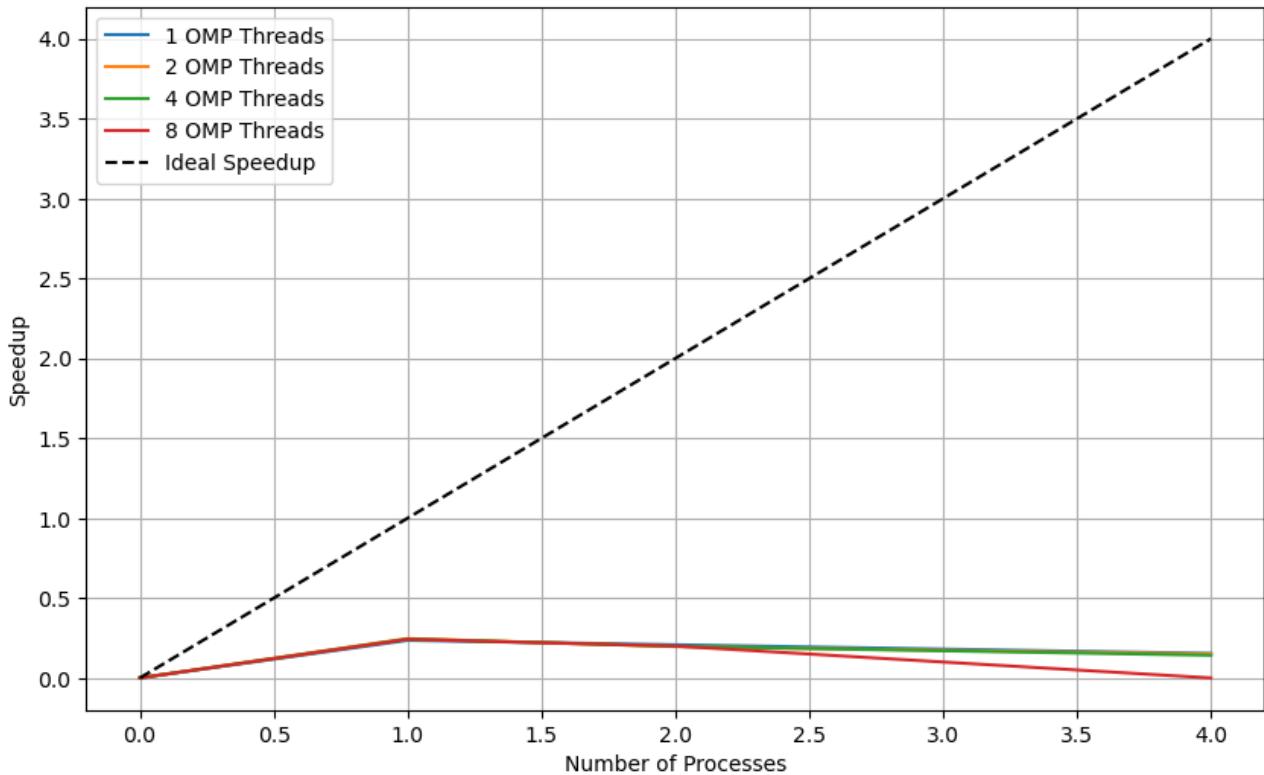
	OMP	MPI	total time	graph creation time	comunication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.2390395	2.0510851	0.0000000	0.1879222	1.0000000	1.0000000
OpenMP+MPI	1	1	3.0670788	2.0566485	0.0000000	0.9545069	0.7300235	0.7300235
OpenMP+MPI	1	2	2.9433489	2.0774227	0.3033856	0.4906827	0.7607116	0.3803558
OpenMP+MPI	1	4	2.8677811	2.0594689	0.3371663	0.3701202	0.7807568	0.1951892
OpenMP+MPI	2	1	2.7393995	2.0565109	0.0000000	0.6265723	0.8173469	0.4086734
OpenMP+MPI	2	2	2.8225355	2.0613126	0.3028667	0.3913522	0.7932724	0.1983181
OpenMP+MPI	2	4	2.8578963	2.0826493	0.3388954	0.3426534	0.7834572	0.0979322
OpenMP+MPI	4	1	2.5950473	2.0444269	0.0000000	0.4947726	0.8628126	0.2157032
OpenMP+MPI	4	2	2.8126693	2.0698695	0.3030713	0.3726751	0.7960550	0.0995069
OpenMP+MPI	4	4	2.8704990	2.0706331	0.3379839	0.3710840	0.7800175	0.0487511
OpenMP+MPI	8	1	2.5969097	2.0516260	0.0000000	0.4889192	0.8621938	0.1077742
OpenMP+MPI	8	2	2.8944021	2.0607815	0.3035300	0.4583809	0.7735758	0.0483485



### 5.2.2 Type 1

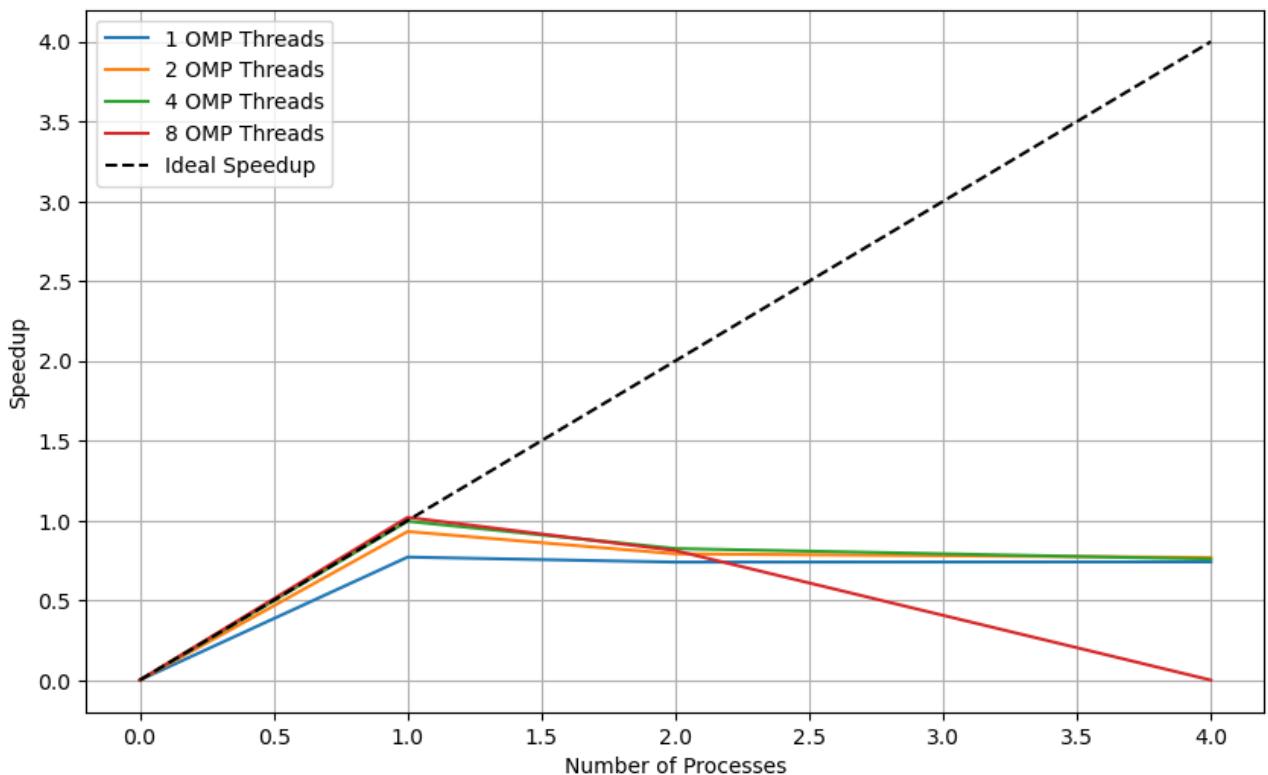
**1536 nodes & 1061043 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0179312	0.0133075	0.0000000	0.0046002	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0761645	0.0123397	0.0000000	0.0069731	0.2354274	0.2354274
OpenMP+MPI	1	2	0.0863001	0.0127038	0.0037790	0.0054233	0.2077772	0.1038886
OpenMP+MPI	1	4	0.1166969	0.0124774	0.0041915	0.0046970	0.1536561	0.0384140
OpenMP+MPI	2	1	0.0727172	0.0123179	0.0000000	0.0055563	0.2465882	0.1232941
OpenMP+MPI	2	2	0.0913044	0.0122986	0.0037011	0.0062637	0.1963892	0.0490973
OpenMP+MPI	2	4	0.1197355	0.0130108	0.0042383	0.0076244	0.1497567	0.0187196
OpenMP+MPI	4	1	0.0735184	0.0123799	0.0000000	0.0053255	0.2439008	0.0609752
OpenMP+MPI	4	2	0.0905605	0.0129137	0.0038481	0.0057799	0.1980025	0.0247503
OpenMP+MPI	4	4	0.1263864	0.0132068	0.0042851	0.0098405	0.1418760	0.0088673
OpenMP+MPI	8	1	0.0741541	0.0124008	0.0000000	0.0063027	0.2418101	0.0302263
OpenMP+MPI	8	2	0.0898564	0.0122933	0.0037217	0.0077783	0.1995540	0.0124721



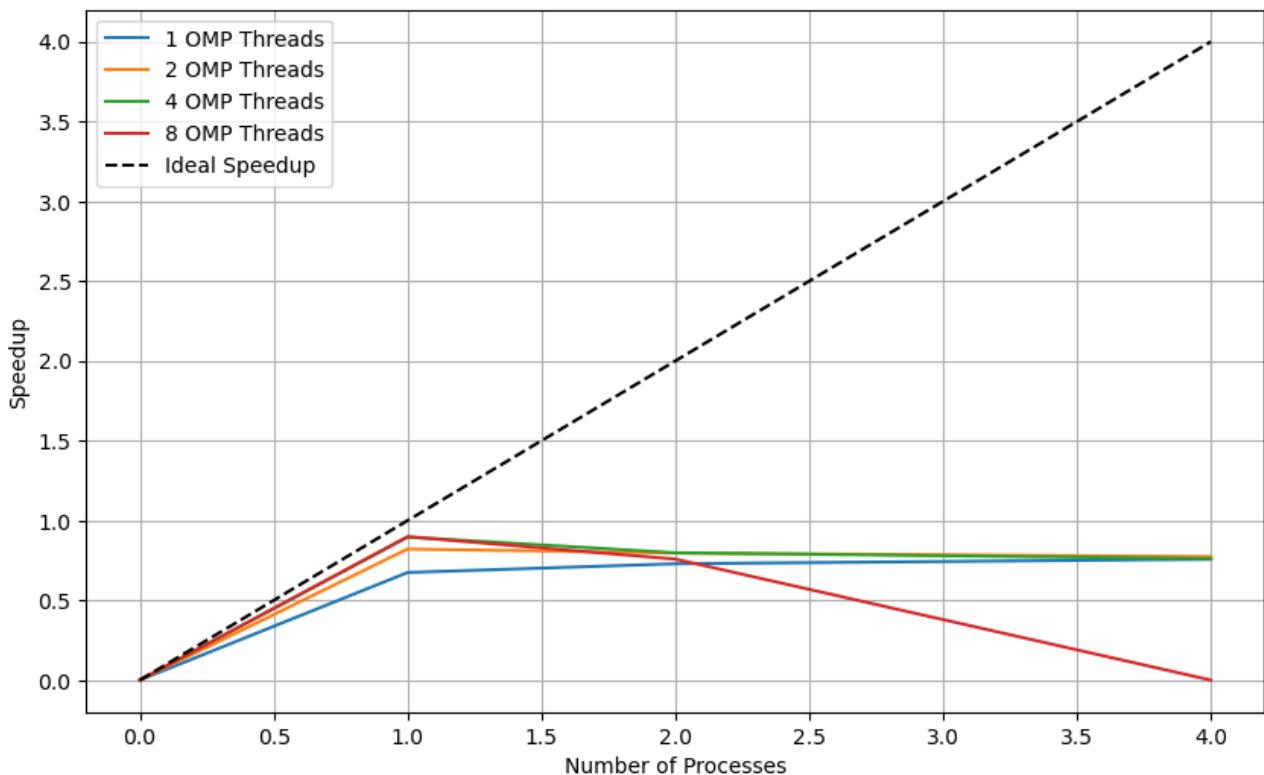
## 7552 nodes & 25660141 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4508875	0.3151238	0.0000000	0.1357376	1.0000000	1.0000000
OpenMP+MPI	1	1	0.5850642	0.3077719	0.0000000	0.2219006	0.7706633	0.7706633
OpenMP+MPI	1	2	0.6098873	0.3165786	0.0955989	0.1288074	0.7392964	0.3696482
OpenMP+MPI	1	4	0.6087929	0.3161430	0.1064387	0.0921765	0.7406255	0.1851564
OpenMP+MPI	2	1	0.4843511	0.3070781	0.0000000	0.1215887	0.9309106	0.4654553
OpenMP+MPI	2	2	0.5700411	0.3135444	0.0951365	0.0935809	0.7909737	0.1977434
OpenMP+MPI	2	4	0.5871866	0.3167959	0.1059655	0.0738960	0.7678778	0.0959847
OpenMP+MPI	4	1	0.4530210	0.3071571	0.0000000	0.0897934	0.9952906	0.2488226
OpenMP+MPI	4	2	0.5471129	0.3122572	0.0948825	0.0721421	0.8241215	0.1030152
OpenMP+MPI	4	4	0.5944072	0.3168557	0.1058447	0.0712571	0.7585499	0.0474094
OpenMP+MPI	8	1	0.4420968	0.3084031	0.0000000	0.0780941	1.0198842	0.1274855
OpenMP+MPI	8	2	0.5556205	0.3139382	0.0950073	0.0781423	0.8115026	0.0507189



## 13568 nodes & 82829469 edges

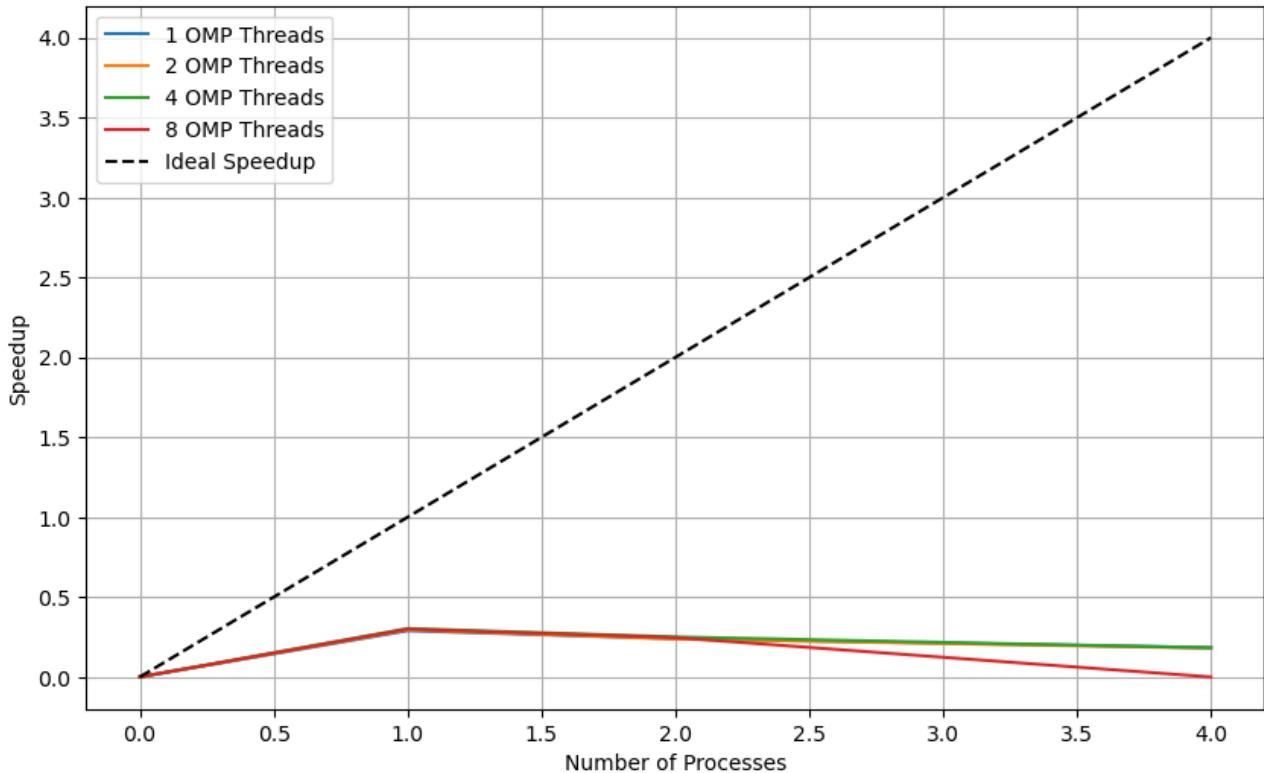
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4525081	1.0739543	0.0000000	0.3785215	1.0000000	1.0000000
OpenMP+MPI	1	1	2.1543789	1.0694628	0.0000000	1.0287157	0.6742120	0.6742120
OpenMP+MPI	1	2	1.9926805	1.0826283	0.3038808	0.5379255	0.7289217	0.3644609
OpenMP+MPI	1	4	1.9183413	1.1019969	0.3376669	0.3818800	0.7571687	0.1892922
OpenMP+MPI	2	1	1.7681295	1.0633567	0.0000000	0.6493969	0.8214941	0.4107471
OpenMP+MPI	2	2	1.8242126	1.0786592	0.3034007	0.3749271	0.7962384	0.1990596
OpenMP+MPI	2	4	1.8767451	1.0962827	0.3377655	0.3469739	0.7739507	0.0967438
OpenMP+MPI	4	1	1.6201171	1.0687786	0.0000000	0.4946905	0.8965451	0.2241363
OpenMP+MPI	4	2	1.8210017	1.0834930	0.3034064	0.3664947	0.7976424	0.0997053
OpenMP+MPI	4	4	1.9081437	1.0989161	0.3371804	0.3778215	0.7612153	0.0475760
OpenMP+MPI	8	1	1.6168154	1.0713705	0.0000000	0.4884410	0.8983759	0.1122970
OpenMP+MPI	8	2	1.9162633	1.0838089	0.30440237	0.4597378	0.7579898	0.0473744



### 5.2.3 Type 2

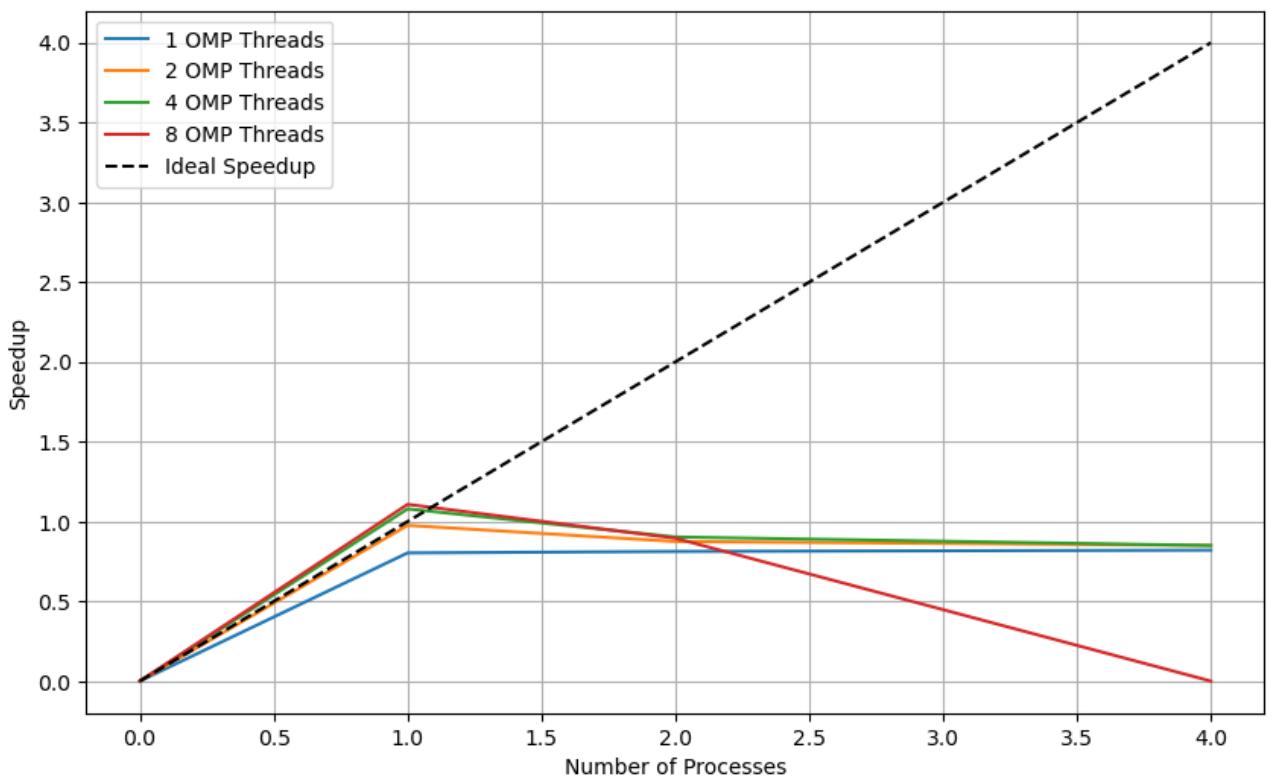
**1536 nodes & 790416 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0230232	0.0159009	0.0000000	0.0070965	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0795922	0.0153440	0.0000000	0.0091111	0.2892645	0.2892645
OpenMP+MPI	1	2	0.0962645	0.0153146	0.0037155	0.0063405	0.2391661	0.1195831
OpenMP+MPI	1	4	0.1251363	0.0155827	0.0041545	0.0048207	0.1839850	0.0459963
OpenMP+MPI	2	1	0.0780451	0.0152803	0.0000000	0.0065997	0.2949985	0.1474993
OpenMP+MPI	2	2	0.0968362	0.0164218	0.0040037	0.0079661	0.2377541	0.0594385
OpenMP+MPI	2	4	0.1275313	0.0158834	0.0042053	0.0084342	0.1805298	0.0225662
OpenMP+MPI	4	1	0.0759175	0.0152657	0.0000000	0.0054692	0.3032659	0.0758165
OpenMP+MPI	4	2	0.0918224	0.0156427	0.0037491	0.0056419	0.2507362	0.0313420
OpenMP+MPI	4	4	0.1263106	0.0158221	0.0042816	0.0097131	0.1822745	0.0113922
OpenMP+MPI	8	1	0.0769836	0.0152989	0.0000000	0.0062893	0.2990663	0.0373833
OpenMP+MPI	8	2	0.0934624	0.0154892	0.0037187	0.0078555	0.2463365	0.0153960



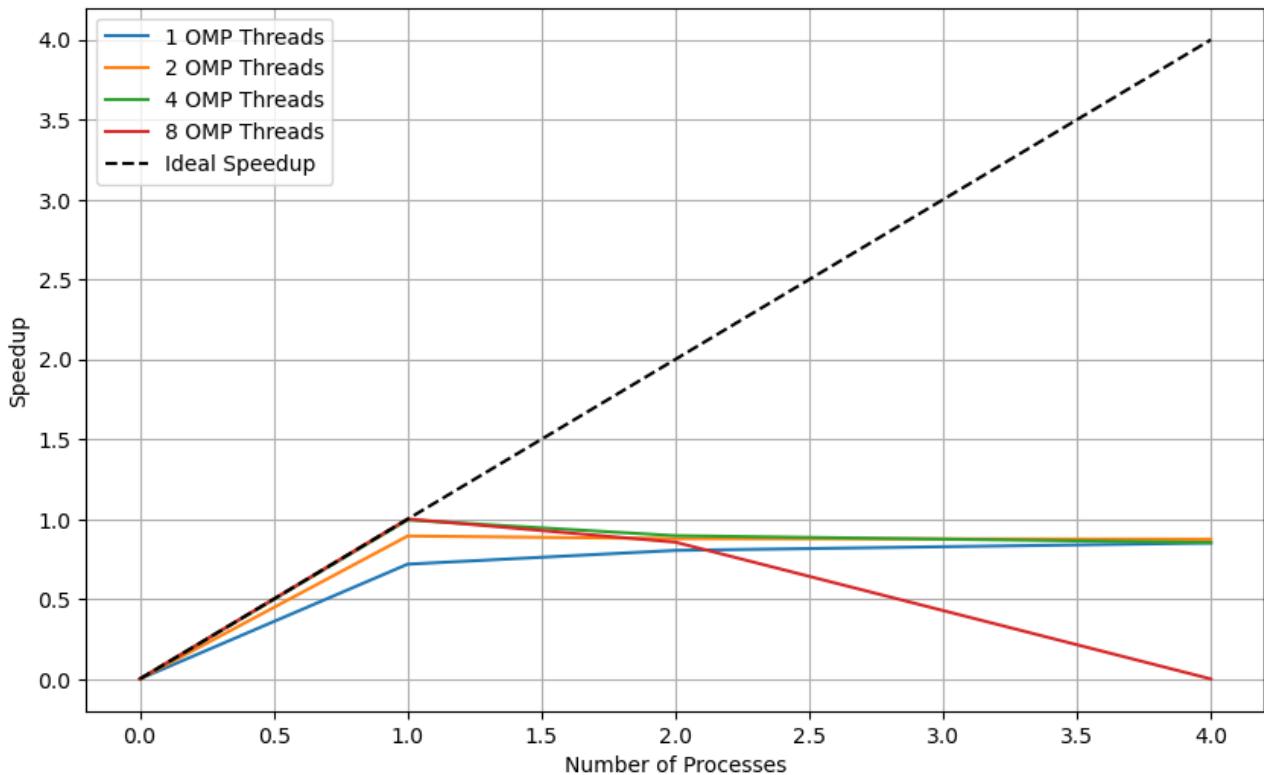
## 7552 nodes & 19101950 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5651073	0.3732839	0.0000000	0.1917963	1.0000000	1.0000000
OpenMP+MPI	1	1	0.7035799	0.3776164	0.0000000	0.2706495	0.8031885	0.8031885
OpenMP+MPI	1	2	0.6957941	0.3823506	0.0954277	0.1528015	0.8121760	0.4060880
OpenMP+MPI	1	4	0.6887249	0.3886799	0.1063257	0.0986058	0.8205124	0.2051281
OpenMP+MPI	2	1	0.5791857	0.3772245	0.0000000	0.1451259	0.9756928	0.4878464
OpenMP+MPI	2	2	0.6460813	0.3820789	0.0958562	0.0991407	0.8746689	0.2186672
OpenMP+MPI	2	4	0.6647075	0.3896696	0.1063057	0.0757826	0.8501593	0.1062699
OpenMP+MPI	4	1	0.5237233	0.3755897	0.0000000	0.0924743	1.0790188	0.2697547
OpenMP+MPI	4	2	0.6257980	0.3838679	0.0951824	0.0762233	0.9030187	0.1128773
OpenMP+MPI	4	4	0.6651071	0.3894373	0.1060943	0.0725513	0.8496486	0.0531030
OpenMP+MPI	8	1	0.5102021	0.3754956	0.0000000	0.0785911	1.1076146	0.1384518
OpenMP+MPI	8	2	0.6313500	0.3817092	0.0956499	0.0808687	0.8950776	0.0559424



## 13568 nodes & 61665181 edges

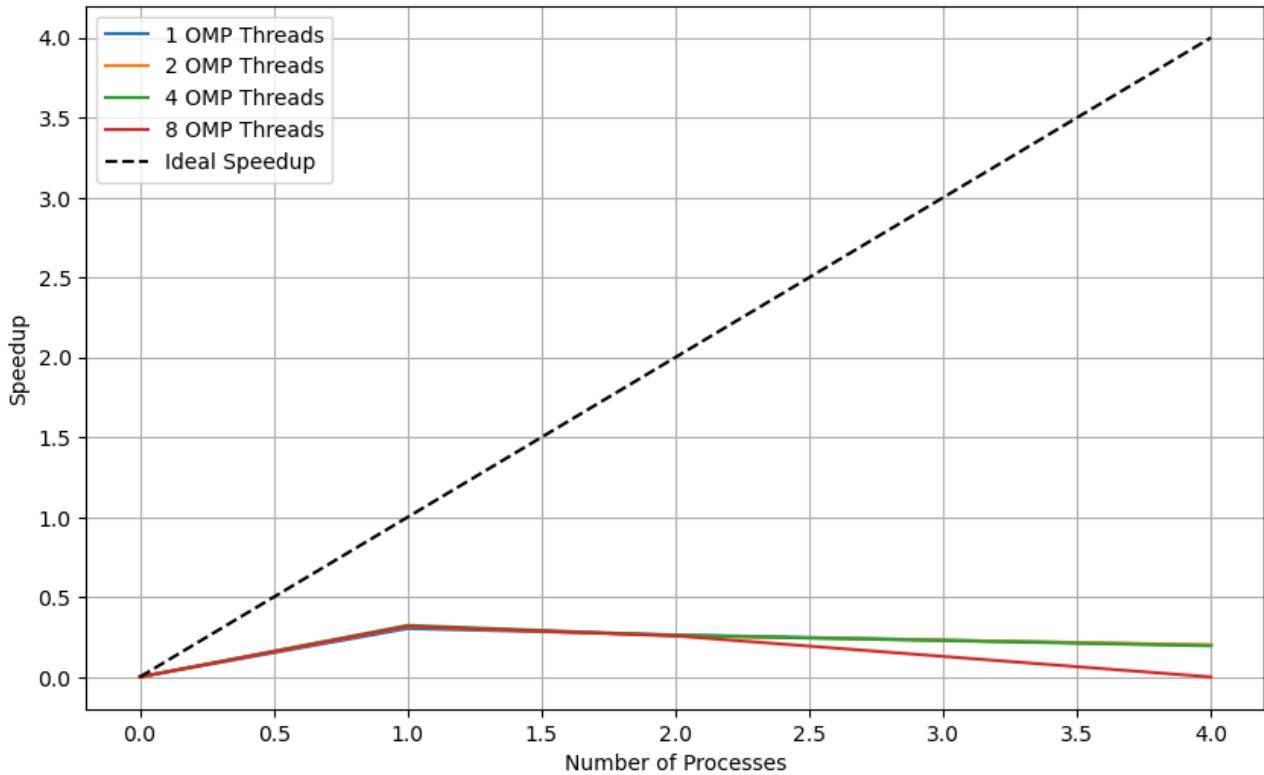
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8328761	1.2704297	0.0000000	0.5624140	1.0000000	1.0000000
OpenMP+MPI	1	1	2.5516403	1.2858759	0.0000000	1.2092265	0.7183129	0.7183129
OpenMP+MPI	1	2	2.2783923	1.2980997	0.3026697	0.6079701	0.8044603	0.4022301
OpenMP+MPI	1	4	2.1551494	1.3208861	0.3371927	0.4039113	0.8503452	0.2125863
OpenMP+MPI	2	1	2.0490919	1.2746711	0.0000000	0.7177379	0.8944822	0.4472411
OpenMP+MPI	2	2	2.0873909	1.2984073	0.3026553	0.4151503	0.8780704	0.2195176
OpenMP+MPI	2	4	2.0958987	1.3208101	0.3369677	0.3414520	0.8745061	0.1093133
OpenMP+MPI	4	1	1.8425421	1.2734252	0.0000000	0.5125958	0.9947540	0.2486885
OpenMP+MPI	4	2	2.0443648	1.3075556	0.3038862	0.3641869	0.8965504	0.1120688
OpenMP+MPI	4	4	2.1473108	1.3331343	0.3384291	0.3747373	0.8535682	0.0533480
OpenMP+MPI	8	1	1.8303320	1.2788335	0.0000000	0.4941919	1.0013900	0.1251737
OpenMP+MPI	8	2	2.1412021	1.3015367	0.3044952	0.4631741	0.8560033	0.0535002



### 5.2.4 Type 3

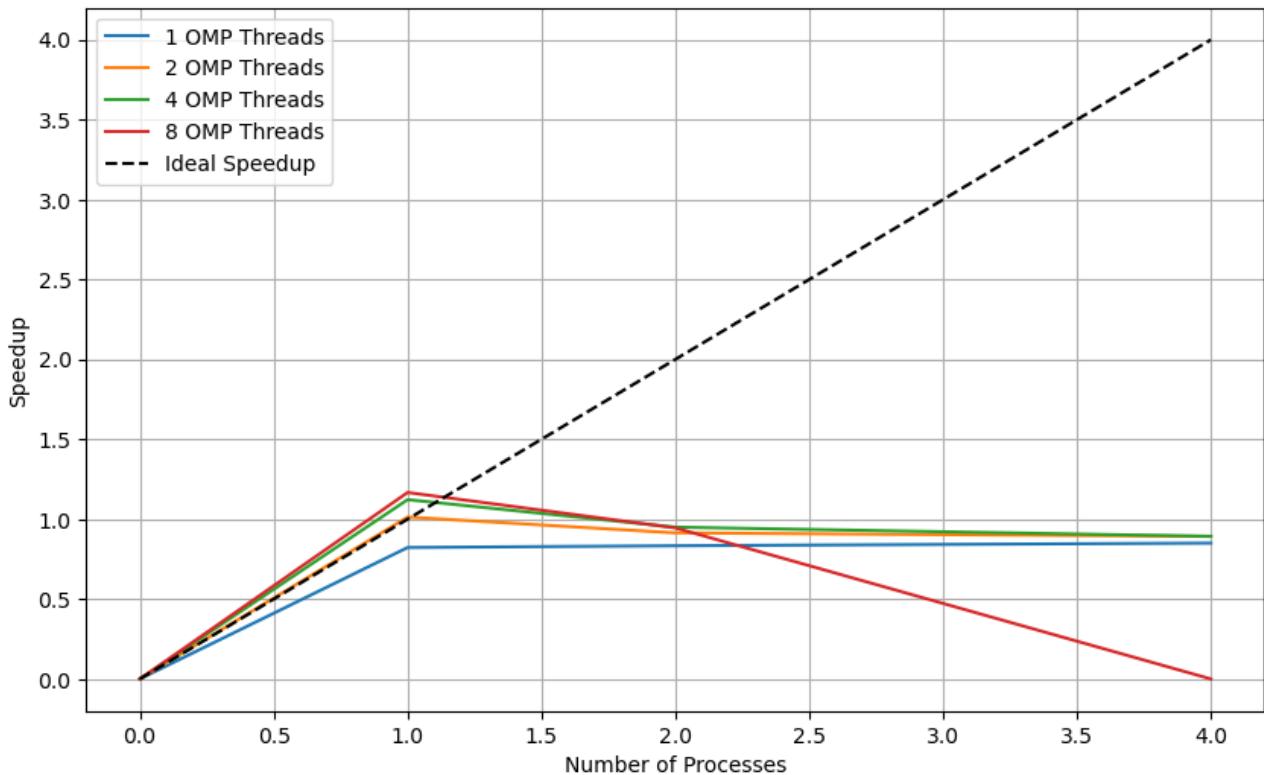
**1536 nodes & 588142 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0245517	0.0159185	0.0000000	0.0086071	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0810162	0.0153502	0.0000000	0.0104847	0.3030472	0.3030472
OpenMP+MPI	1	2	0.0931131	0.0155349	0.0037489	0.0070079	0.2636764	0.1318382
OpenMP+MPI	1	4	0.1240320	0.0157977	0.0041877	0.0049717	0.1979468	0.0494867
OpenMP+MPI	2	1	0.0777190	0.0152027	0.0000000	0.0072757	0.3159039	0.1579519
OpenMP+MPI	2	2	0.0950939	0.0163844	0.0038139	0.0068129	0.2581840	0.0645460
OpenMP+MPI	2	4	0.1218357	0.0157560	0.0042878	0.0080817	0.2015152	0.0251894
OpenMP+MPI	4	1	0.0759973	0.0152225	0.0000000	0.0056490	0.3230605	0.0807651
OpenMP+MPI	4	2	0.0944237	0.0156495	0.0037944	0.0064438	0.2600165	0.0325021
OpenMP+MPI	4	4	0.1261725	0.0157815	0.0042203	0.0092197	0.1945887	0.0121618
OpenMP+MPI	8	1	0.0775119	0.0151819	0.0000000	0.0063538	0.3167478	0.0395935
OpenMP+MPI	8	2	0.0955298	0.0159463	0.0039061	0.0079015	0.2570060	0.0160629



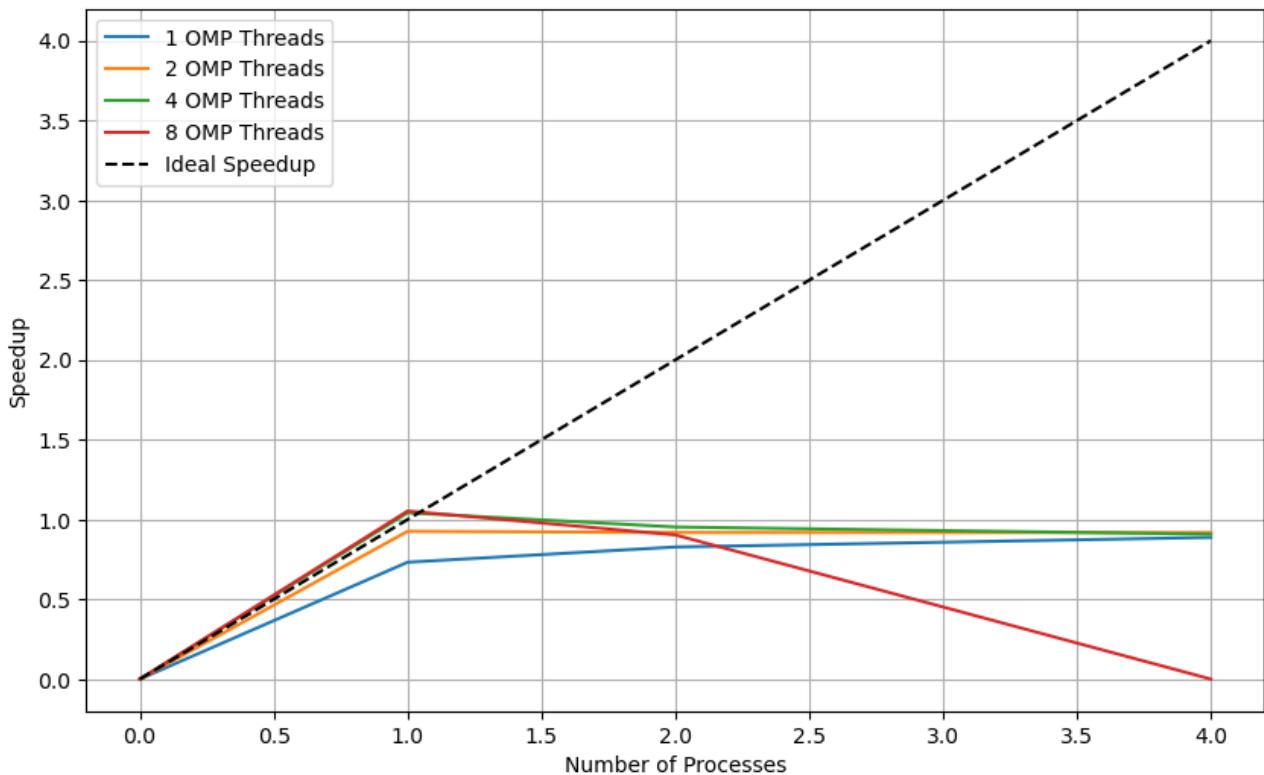
## 7552 nodes & 14260151 edges

	OMP	MPI	total time	graph creation time	comunication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5931903	0.3752528	0.0000000	0.2179093	1.0000000	1.0000000
OpenMP+MPI	1	1	0.7211479	0.3754671	0.0000000	0.2903039	0.8225641	0.8225641
OpenMP+MPI	1	2	0.7115705	0.3828685	0.0949626	0.1624685	0.8336353	0.4168177
OpenMP+MPI	1	4	0.6977405	0.3870778	0.1062141	0.1039936	0.8501590	0.2125397
OpenMP+MPI	2	1	0.5855179	0.3736196	0.0000000	0.1565488	1.0131036	0.5065518
OpenMP+MPI	2	2	0.6489041	0.3818251	0.0947140	0.1041062	0.9141418	0.2285355
OpenMP+MPI	2	4	0.6649607	0.3877181	0.1059960	0.0768095	0.8920683	0.1115085
OpenMP+MPI	4	1	0.5286671	0.3749736	0.0000000	0.0963018	1.1220490	0.2805122
OpenMP+MPI	4	2	0.6244469	0.3819911	0.0949771	0.0789120	0.9499451	0.1187431
OpenMP+MPI	4	4	0.6645146	0.3875443	0.1064224	0.0728525	0.8926671	0.0557917
OpenMP+MPI	8	1	0.5083115	0.3730663	0.0000000	0.0794425	1.1669818	0.1458727
OpenMP+MPI	8	2	0.6276181	0.3808649	0.0949405	0.0810753	0.9451454	0.0590716



## 13568 nodes & 46014096 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.9347739	1.2718554	0.0000000	0.6628874	1.0000000	1.0000000
OpenMP+MPI	1	1	2.6420407	1.2788902	0.0000000	1.3072565	0.7323029	0.7323029
OpenMP+MPI	1	2	2.3377105	1.3007094	0.3040795	0.6626913	0.8276362	0.4138181
OpenMP+MPI	1	4	2.1795641	1.3202437	0.3376422	0.4256646	0.8876885	0.2219221
OpenMP+MPI	2	1	2.0892587	1.2755145	0.0000000	0.7574653	0.9260576	0.4630288
OpenMP+MPI	2	2	2.1060414	1.2998939	0.3047650	0.4315476	0.9186780	0.2296695
OpenMP+MPI	2	4	2.1072875	1.3256306	0.3388201	0.3443077	0.9181348	0.1147668
OpenMP+MPI	4	1	1.8565443	1.2748995	0.0000000	0.5243093	1.0421372	0.2605343
OpenMP+MPI	4	2	2.0302331	1.3000109	0.3039523	0.3563593	0.9529812	0.1191226
OpenMP+MPI	4	4	2.1307748	1.3198059	0.3377091	0.3763873	0.9080143	0.0567509
OpenMP+MPI	8	1	1.8373677	1.2817935	0.0000000	0.4991886	1.0530140	0.1316268
OpenMP+MPI	8	2	2.1419777	1.3039283	0.3036777	0.4640841	0.9032652	0.0564541



### 5.3 Optimization 2

Optimization 2 does not introduce an improvement in speedup and performance compared to O0, but has better results than O1.

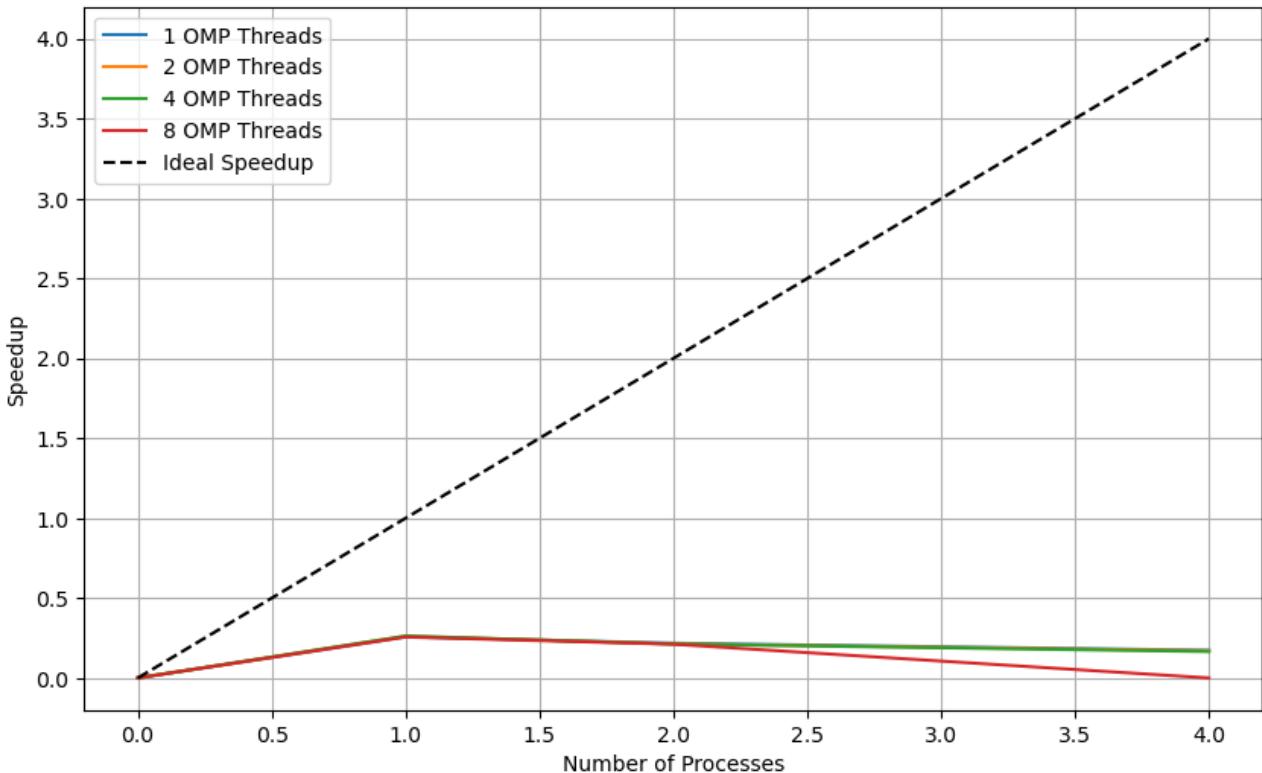
The algorithm remains efficient for large and mainly for medium, type 3, graphs.

Optimization 0 remains the one with the best performance

#### 5.3.1 Type 0

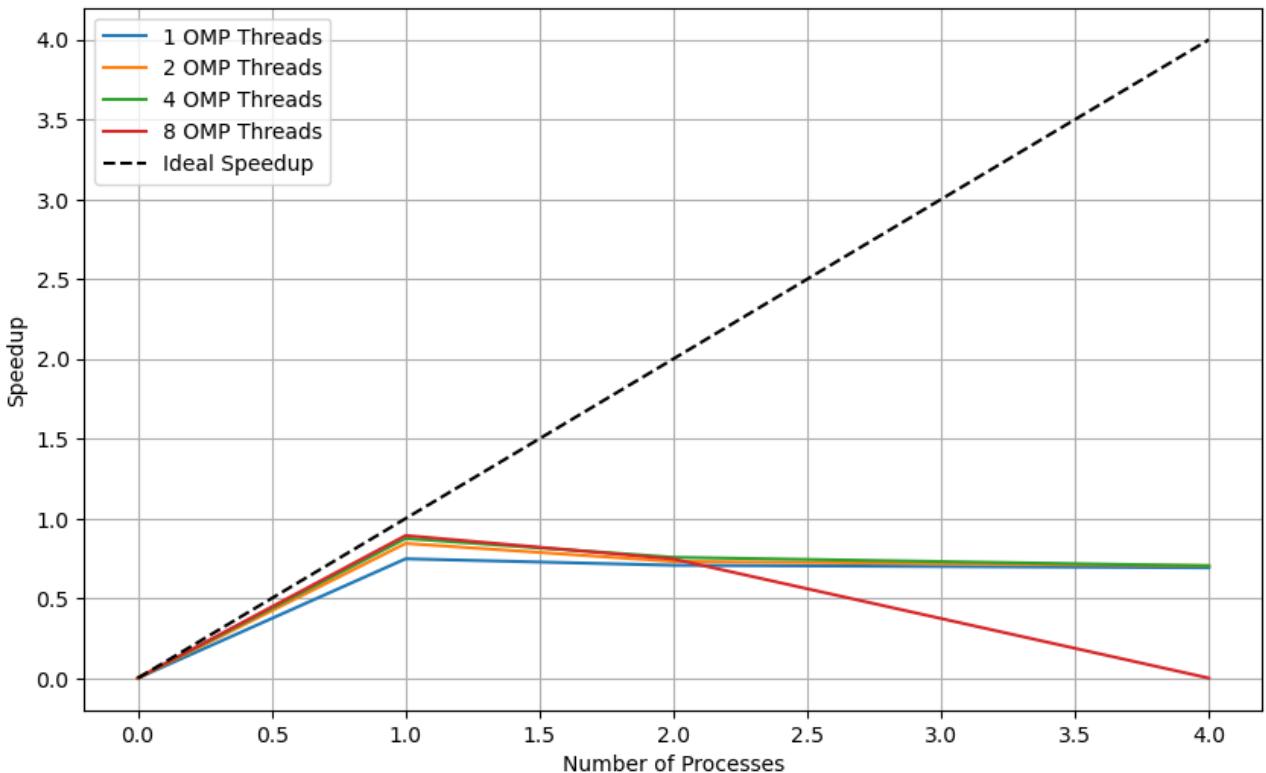
**1536 nodes & 2357760 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0208825	0.0178341	0.0000000	0.0030226	1.000000	1.000000
OpenMP+MPI	1	1	0.0810043	0.0173456	0.0000000	0.0062289	0.2577944	0.2577944
OpenMP+MPI	1	2	0.0960066	0.0180234	0.0038708	0.0050819	0.2175107	0.1087554
OpenMP+MPI	1	4	0.1205517	0.0186819	0.0043024	0.0047803	0.1732241	0.0433060
OpenMP+MPI	2	1	0.0799280	0.0172711	0.0000000	0.0052453	0.2612660	0.1306330
OpenMP+MPI	2	2	0.0977962	0.0173071	0.0037618	0.0055615	0.2135305	0.0533826
OpenMP+MPI	2	4	0.1227018	0.0176477	0.0043326	0.0074519	0.1701888	0.0212736
OpenMP+MPI	4	1	0.0794535	0.0170988	0.0000000	0.0049635	0.2628262	0.0657065
OpenMP+MPI	4	2	0.0983850	0.0175731	0.0038118	0.0058413	0.2122525	0.0265316
OpenMP+MPI	4	4	0.1263997	0.0182267	0.0043515	0.0088641	0.1652098	0.0103256
OpenMP+MPI	8	1	0.0815377	0.0171645	0.0000000	0.0061145	0.2561080	0.0320135
OpenMP+MPI	8	2	0.0988473	0.0181789	0.0038372	0.0080261	0.2112598	0.0132037



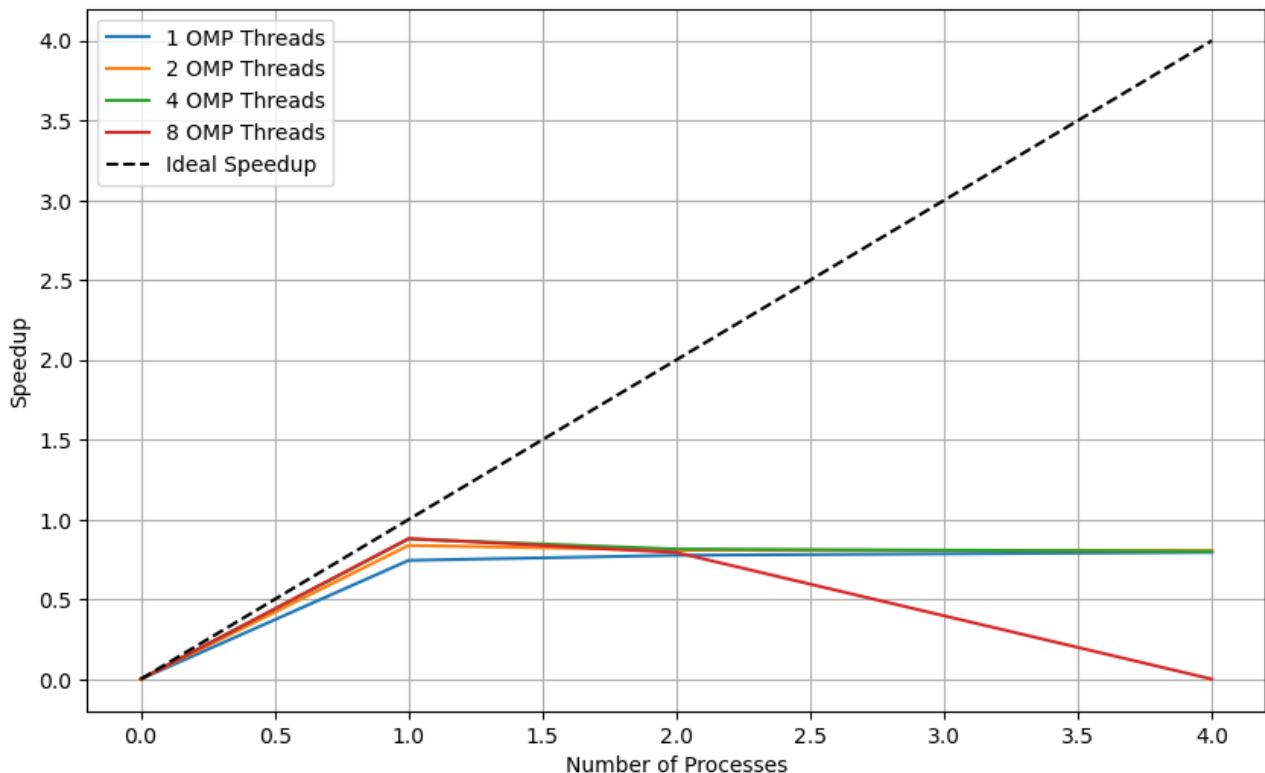
## 7552 nodes & 57025152 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5202267	0.4483063	0.0000000	0.0718910	1.0000000	1.0000000
OpenMP+MPI	1	1	0.6959564	0.4484207	0.0000000	0.1908655	0.7474990	0.7474990
OpenMP+MPI	1	2	0.7358058	0.4580125	0.0954519	0.1091407	0.7070164	0.3535082
OpenMP+MPI	1	4	0.7522166	0.4617559	0.1066897	0.0901315	0.6915917	0.1728979
OpenMP+MPI	2	1	0.6164907	0.4500568	0.0000000	0.1097009	0.8438518	0.4219259
OpenMP+MPI	2	2	0.7118506	0.4551355	0.0952069	0.0920725	0.7308089	0.1827022
OpenMP+MPI	2	4	0.7392951	0.4652904	0.1064207	0.0725473	0.7036794	0.0879599
OpenMP+MPI	4	1	0.5946177	0.4479661	0.0000000	0.0900382	0.8748928	0.2187232
OpenMP+MPI	4	2	0.6874473	0.4561870	0.0961940	0.0672621	0.7567514	0.0945939
OpenMP+MPI	4	4	0.7395935	0.4647901	0.1065291	0.0701272	0.7033955	0.0439622
OpenMP+MPI	8	1	0.5822546	0.4474070	0.0000000	0.0776081	0.8934695	0.1116837
OpenMP+MPI	8	2	0.6970459	0.4566567	0.0956561	0.0771698	0.7463307	0.0466457



## 13568 nodes & 184077056 edges

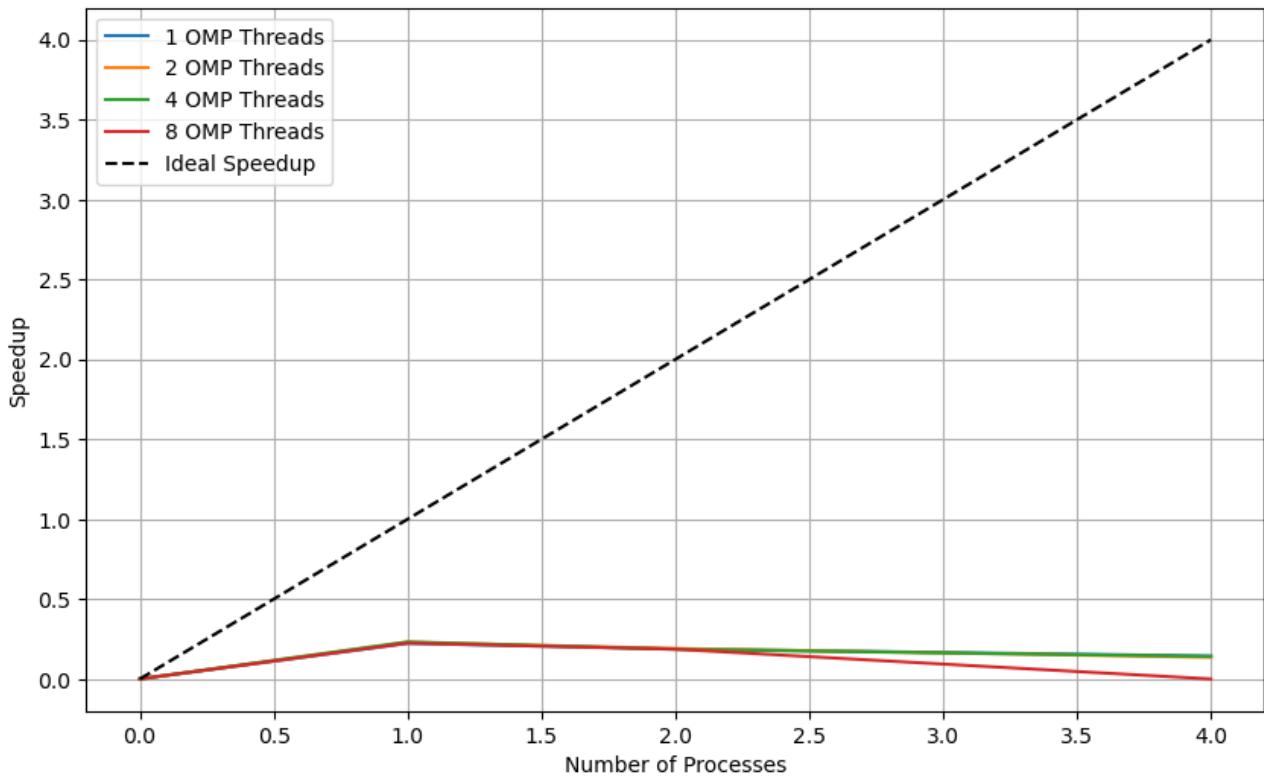
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.2993167	2.0608135	0.0000000	0.2384716	1.0000000	1.0000000
OpenMP+MPI	1	1	3.0907465	2.0706283	0.0000000	0.9632975	0.7439357	0.7439357
OpenMP+MPI	1	2	2.9647774	2.0693484	0.3022593	0.5234882	0.7755445	0.3877722
OpenMP+MPI	1	4	2.8945087	2.0831157	0.3383782	0.3738922	0.7943720	0.1985930
OpenMP+MPI	2	1	2.7449537	2.0578243	0.0000000	0.6291933	0.8376523	0.4188261
OpenMP+MPI	2	2	2.8480223	2.0776062	0.3034560	0.3957658	0.8073380	0.2018345
OpenMP+MPI	2	4	2.8554041	2.0868367	0.3378505	0.3405885	0.8052509	0.1006564
OpenMP+MPI	4	1	2.6198971	2.0599949	0.0000000	0.5024973	0.8776363	0.2194091
OpenMP+MPI	4	2	2.8213007	2.0788819	0.3043339	0.3702579	0.8149846	0.1018731
OpenMP+MPI	4	4	2.8725078	2.0800553	0.3367194	0.3570971	0.8004562	0.0500285
OpenMP+MPI	8	1	2.6102747	2.0671988	0.0000000	0.4856813	0.8808716	0.1101089
OpenMP+MPI	8	2	2.8982483	2.0703196	0.3010310	0.4570251	0.7933471	0.0495842



### 5.3.2 Type 1

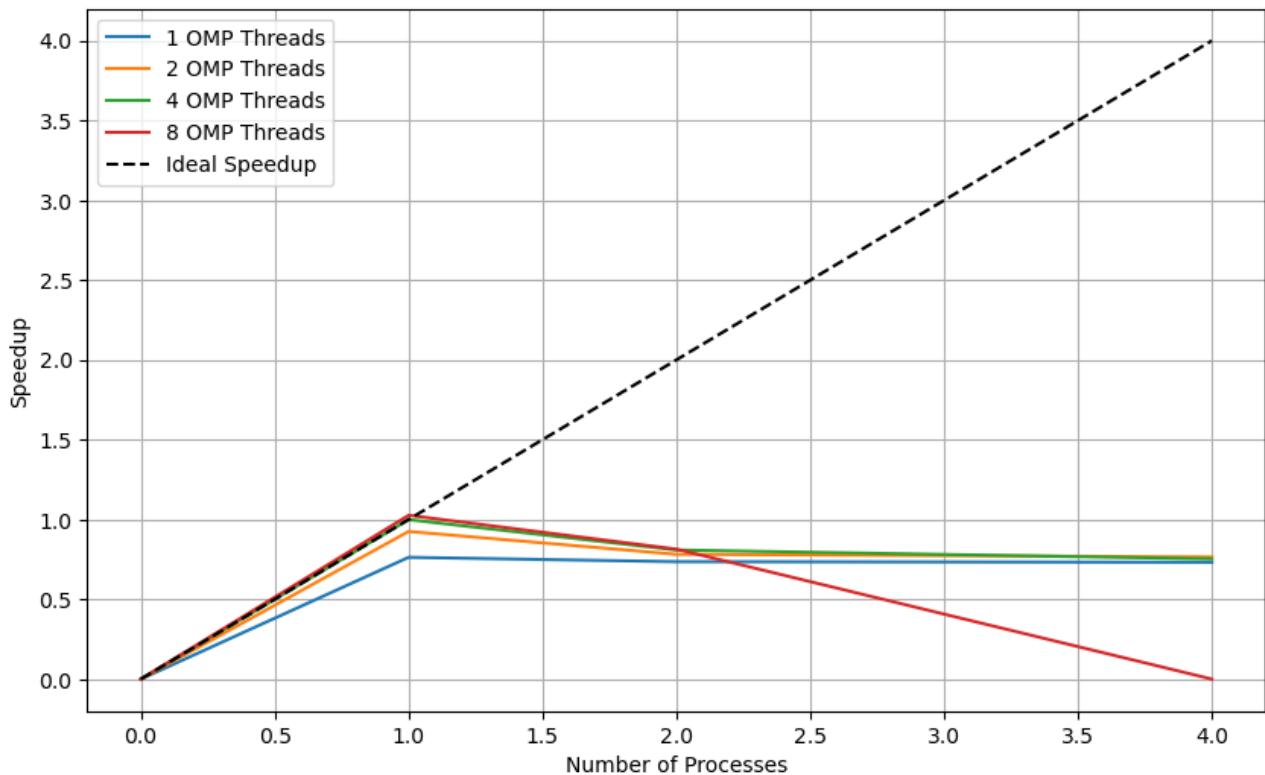
**1536 nodes & 1061043 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0171206	0.0122602	0.0000000	0.0048345	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0768606	0.0116422	0.0000000	0.0069373	0.2227487	0.2227487
OpenMP+MPI	1	2	0.0910106	0.0121663	0.0038464	0.0055566	0.1881165	0.0940583
OpenMP+MPI	1	4	0.1188474	0.0120503	0.0041600	0.0046411	0.1440553	0.0360138
OpenMP+MPI	2	1	0.0747098	0.0115299	0.0000000	0.0055247	0.2291614	0.1145807
OpenMP+MPI	2	2	0.0898029	0.0115339	0.0036934	0.0061920	0.1906463	0.0476616
OpenMP+MPI	2	4	0.1262887	0.0123114	0.0043132	0.0077517	0.1355671	0.0169459
OpenMP+MPI	4	1	0.0737081	0.0116009	0.0000000	0.0053665	0.2322758	0.0580689
OpenMP+MPI	4	2	0.0922907	0.0122476	0.0038443	0.0059382	0.1855072	0.0231884
OpenMP+MPI	4	4	0.1215052	0.0119693	0.0042646	0.0089745	0.1409043	0.0088065
OpenMP+MPI	8	1	0.0763101	0.0116322	0.0000000	0.0062330	0.2243557	0.0280445
OpenMP+MPI	8	2	0.0919297	0.0116049	0.0036847	0.0079483	0.1862357	0.0116397



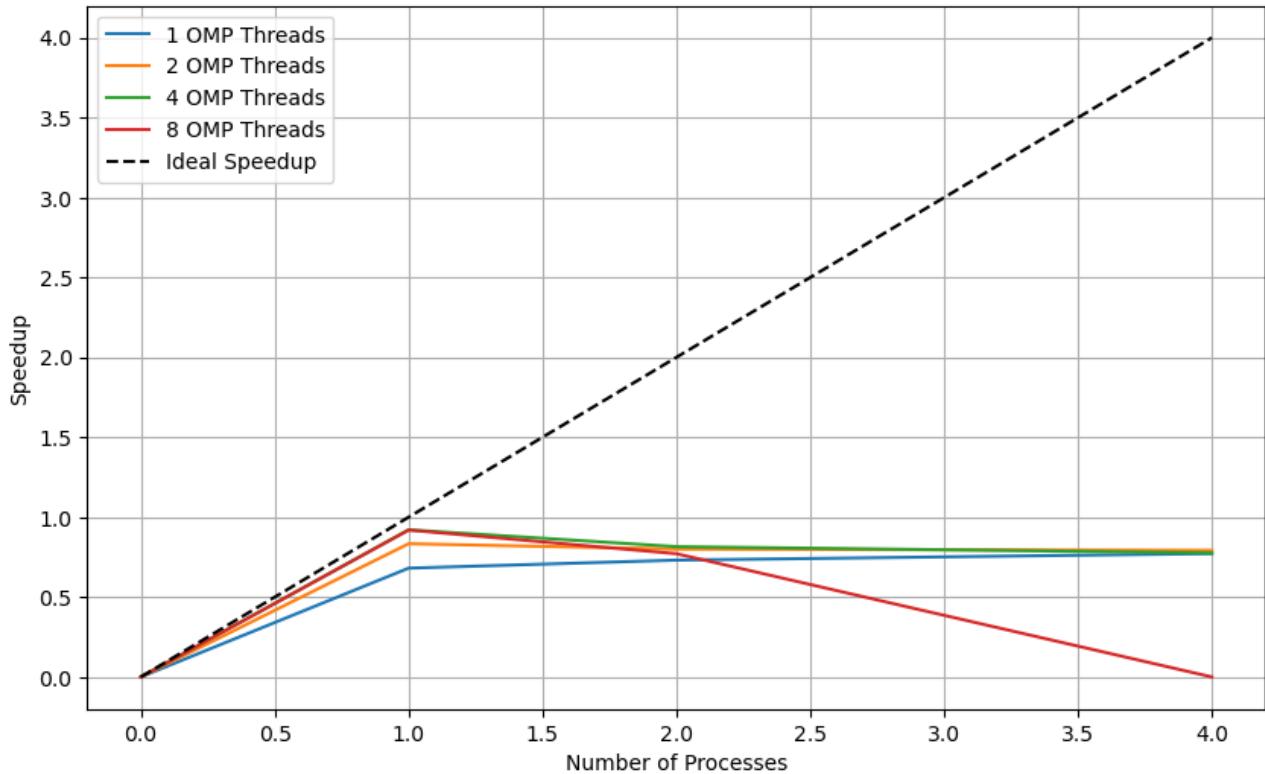
## 7552 nodes & 25660141 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4373078	0.2939786	0.0000000	0.1433019	1.0000000	1.0000000
OpenMP+MPI	1	1	0.5731190	0.2917594	0.0000000	0.2243693	0.7630314	0.7630314
OpenMP+MPI	1	2	0.5944439	0.2977715	0.0955122	0.1287097	0.7356587	0.3678293
OpenMP+MPI	1	4	0.5973802	0.3017738	0.1067873	0.0928163	0.7320427	0.1830107
OpenMP+MPI	2	1	0.4727211	0.2928995	0.0000000	0.1222374	0.9250862	0.4625431
OpenMP+MPI	2	2	0.5593818	0.2976864	0.0951121	0.0937473	0.7817698	0.1954425
OpenMP+MPI	2	4	0.5714718	0.3008793	0.1061317	0.0734390	0.7652308	0.0956538
OpenMP+MPI	4	1	0.4376422	0.2912780	0.0000000	0.0895589	0.9992359	0.2498090
OpenMP+MPI	4	2	0.5406594	0.2969802	0.0957577	0.0733438	0.8088416	0.1011052
OpenMP+MPI	4	4	0.5806997	0.3026085	0.1056677	0.0720199	0.7530705	0.0470669
OpenMP+MPI	8	1	0.4261119	0.2918669	0.0000000	0.0774949	1.0262746	0.1282843
OpenMP+MPI	8	2	0.5371176	0.2966417	0.0949699	0.0781433	0.8141751	0.0508859



## 13568 nodes & 82829469 edges

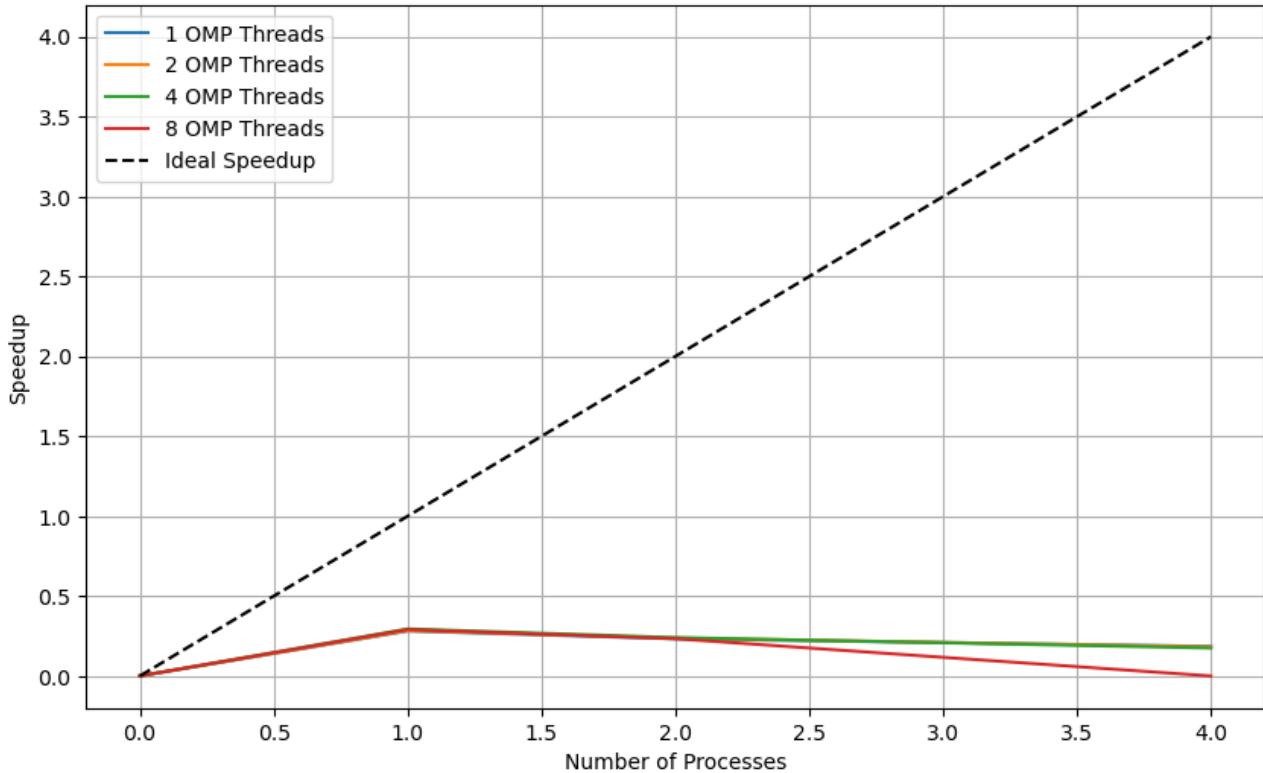
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4434622	1.0212595	0.0000000	0.4221686	1.0000000	1.0000000
OpenMP+MPI	1	1	2.1195991	1.0236413	0.0000000	1.0392361	0.6810072	0.6810072
OpenMP+MPI	1	2	1.9747516	1.0375117	0.3045255	0.5585909	0.7309589	0.3654794
OpenMP+MPI	1	4	1.8699335	1.0498223	0.3384023	0.3793563	0.7719324	0.1929831
OpenMP+MPI	2	1	1.7311668	1.0232871	0.0000000	0.6502344	0.8338089	0.4169044
OpenMP+MPI	2	2	1.8039927	1.0359825	0.3033873	0.3942401	0.8001486	0.2000371
OpenMP+MPI	2	4	1.8218853	1.0531806	0.3375225	0.3406627	0.7922904	0.0990363
OpenMP+MPI	4	1	1.5678732	1.0201705	0.0000000	0.4899658	0.9206498	0.2301625
OpenMP+MPI	4	2	1.7699474	1.0366415	0.3047027	0.3590457	0.8155396	0.1019425
OpenMP+MPI	4	4	1.8596093	1.0531883	0.3374543	0.3751156	0.7762180	0.0485136
OpenMP+MPI	8	1	1.5698983	1.0213061	0.0000000	0.4909315	0.9194622	0.1149328
OpenMP+MPI	8	2	1.8713522	1.0388689	0.3038972	0.4589641	0.7713472	0.0482092



### 5.3.3 Type 2

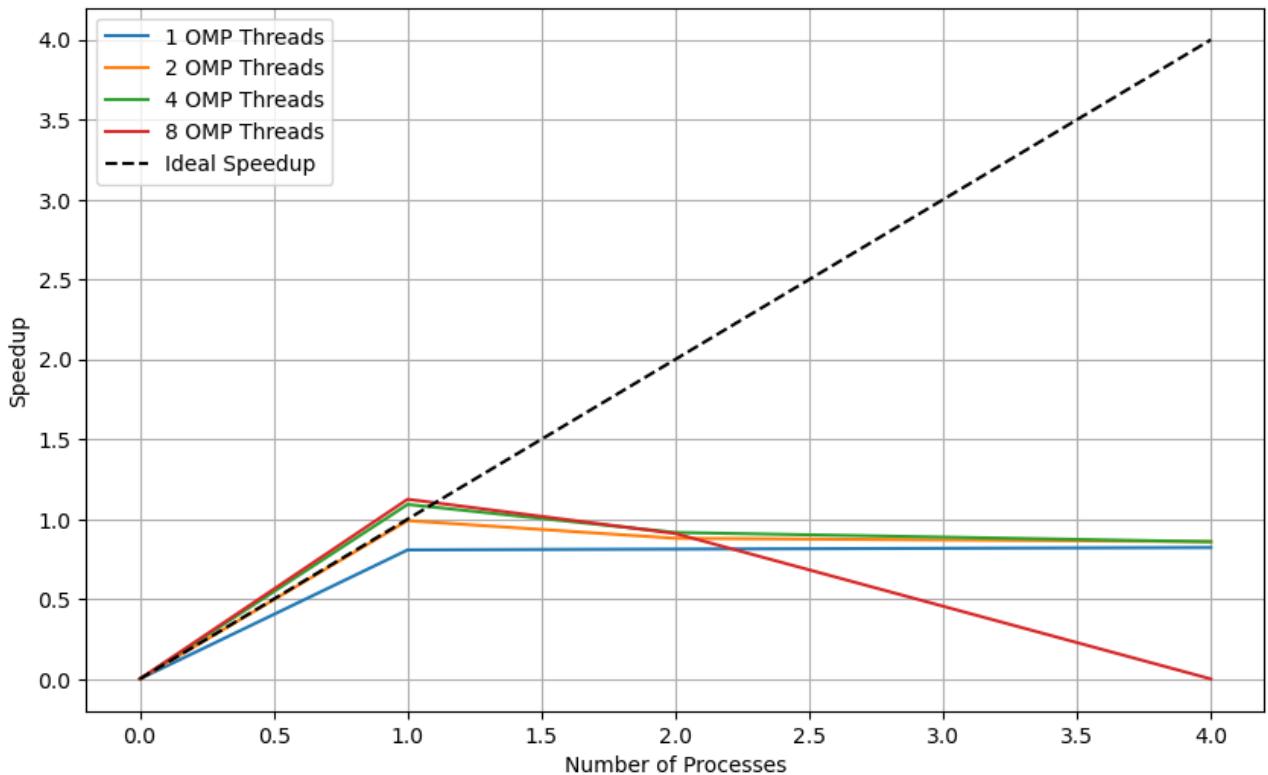
**1536 nodes & 790416 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0227595	0.0153609	0.0000000	0.0073741	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0808794	0.0145580	0.0000000	0.0091391	0.2814000	0.2814000
OpenMP+MPI	1	2	0.0978718	0.0157673	0.0039537	0.0070487	0.2325437	0.1162718
OpenMP+MPI	1	4	0.1228209	0.0147690	0.0042022	0.0048722	0.1853062	0.0463265
OpenMP+MPI	2	1	0.0802269	0.0144465	0.0000000	0.0066355	0.2836888	0.1418444
OpenMP+MPI	2	2	0.0958935	0.0153989	0.0039465	0.0070180	0.2373410	0.0593352
OpenMP+MPI	2	4	0.1248161	0.0151772	0.0043102	0.0073641	0.1823440	0.0227930
OpenMP+MPI	4	1	0.0772609	0.0144435	0.0000000	0.0054451	0.2945795	0.0736449
OpenMP+MPI	4	2	0.0946653	0.0146727	0.0037856	0.0057440	0.2404205	0.0300526
OpenMP+MPI	4	4	0.1303069	0.0161620	0.0043893	0.0098849	0.1746605	0.0109163
OpenMP+MPI	8	1	0.0783290	0.0144515	0.0000000	0.0062834	0.2905625	0.0363203
OpenMP+MPI	8	2	0.0976183	0.0150892	0.0038518	0.0080980	0.2331475	0.0145717



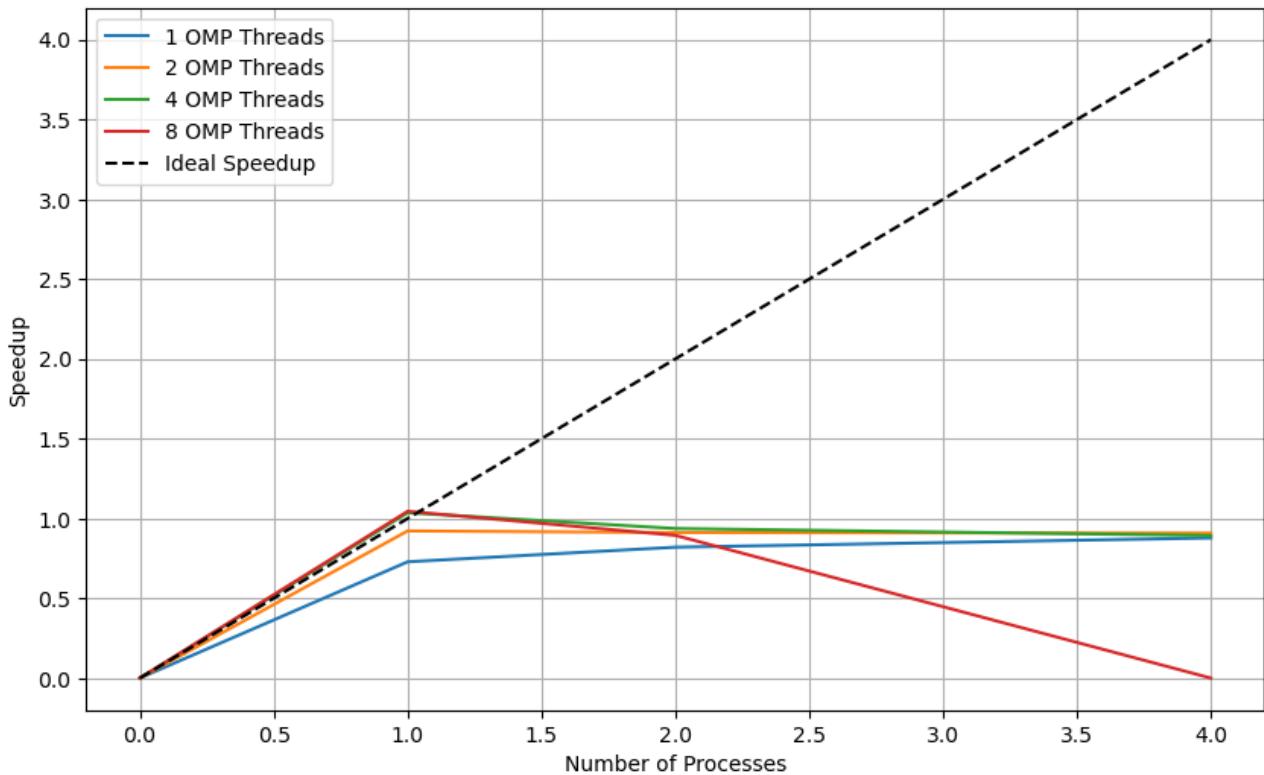
## 7552 nodes & 19101950 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5573064	0.3587729	0.0000000	0.1985069	1.0000000	1.0000000
OpenMP+MPI	1	1	0.6900585	0.3598918	0.0000000	0.2730997	0.8076219	0.8076219
OpenMP+MPI	1	2	0.6858775	0.3662890	0.0958283	0.1534683	0.8125451	0.4062725
OpenMP+MPI	1	4	0.6776282	0.3724041	0.1063543	0.1007001	0.8224368	0.2056092
OpenMP+MPI	2	1	0.5626014	0.3596333	0.0000000	0.1456773	0.9905884	0.4952942
OpenMP+MPI	2	2	0.6331016	0.3642214	0.0951905	0.1022372	0.8802796	0.2200699
OpenMP+MPI	2	4	0.6484460	0.3708213	0.1057763	0.0743261	0.8594492	0.1074312
OpenMP+MPI	4	1	0.5103334	0.3587117	0.0000000	0.0948929	1.0920438	0.2730109
OpenMP+MPI	4	2	0.6080161	0.3669509	0.0953964	0.0773244	0.9165981	0.1145748
OpenMP+MPI	4	4	0.6490325	0.3715988	0.1063143	0.0732415	0.8586725	0.0536670
OpenMP+MPI	8	1	0.4955393	0.3592337	0.0000000	0.0786921	1.1246461	0.1405808
OpenMP+MPI	8	2	0.6124206	0.3674936	0.0953611	0.0805341	0.9100060	0.0568754



## 13568 nodes & 61665181 edges

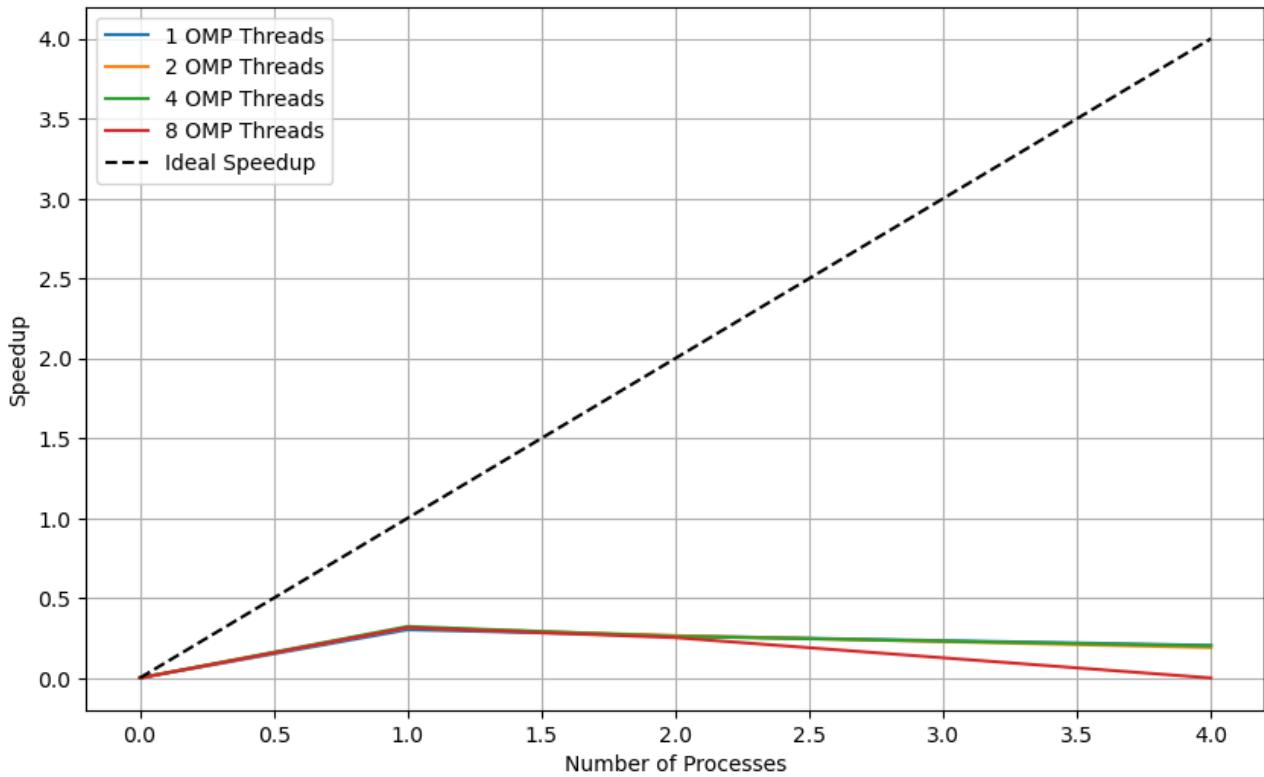
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8449185	1.2192096	0.0000000	0.6256766	1.0000000	1.0000000
OpenMP+MPI	1	1	2.5319195	1.2454163	0.0000000	1.2292161	0.7286640	0.7286640
OpenMP+MPI	1	2	2.2494600	1.2357599	0.3039081	0.6373187	0.8201606	0.4100803
OpenMP+MPI	1	4	2.1012132	1.2573762	0.3382351	0.4049666	0.8780254	0.2195063
OpenMP+MPI	2	1	2.0007069	1.2190841	0.0000000	0.7239607	0.9221333	0.4610667
OpenMP+MPI	2	2	2.0226543	1.2320708	0.3033022	0.4173129	0.9121274	0.2280319
OpenMP+MPI	2	4	2.0306148	1.2541736	0.3376370	0.3427341	0.9085517	0.1135690
OpenMP+MPI	4	1	1.7807974	1.2124215	0.0000000	0.5098857	1.0360070	0.2590017
OpenMP+MPI	4	2	1.9682625	1.2319441	0.3042559	0.3606719	0.9373336	0.1171667
OpenMP+MPI	4	4	2.0658689	1.2574355	0.3391187	0.3742121	0.8930473	0.0558155
OpenMP+MPI	8	1	1.7653876	1.2117213	0.0000000	0.4958457	1.0450501	0.1306313
OpenMP+MPI	8	2	2.0628662	1.2362493	0.3045524	0.4536319	0.8943472	0.0558967



### 5.3.4 Type 3

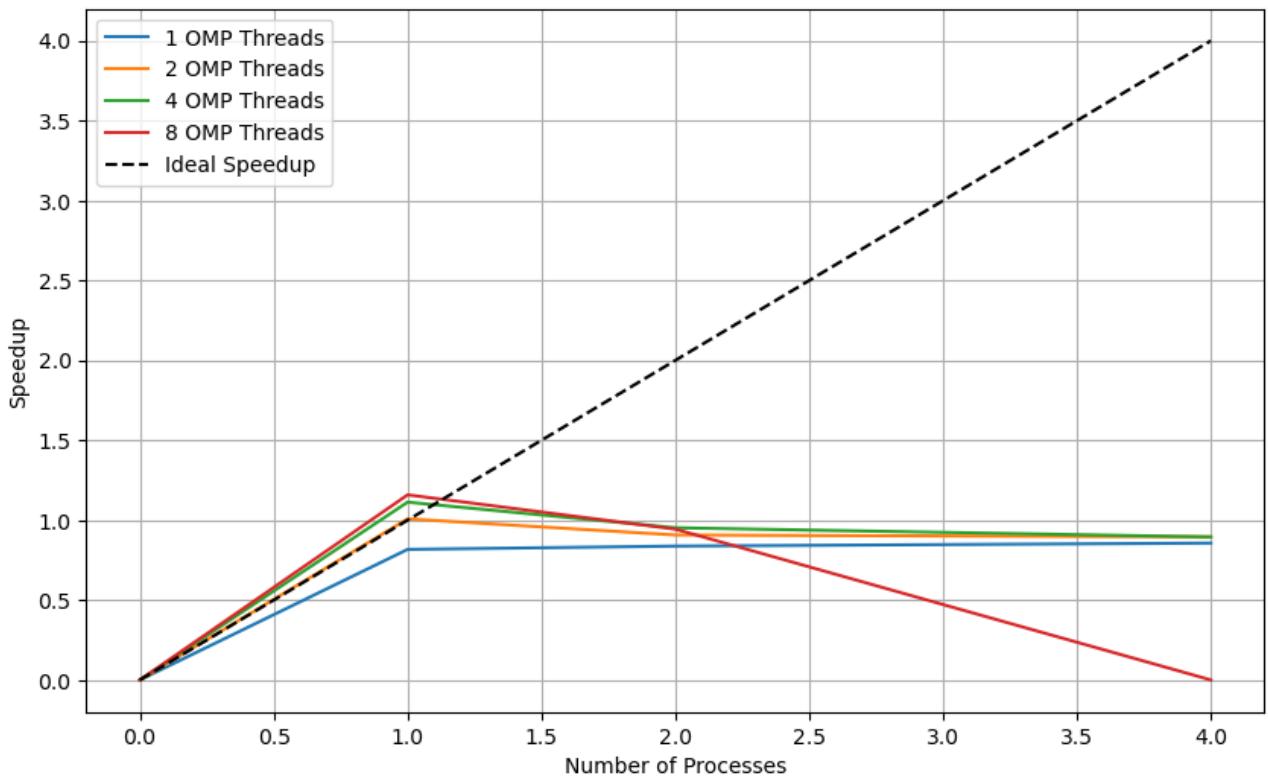
**1536 nodes & 588142 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0251513	0.0162583	0.0000000	0.0088655	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0833054	0.0156966	0.0000000	0.0106277	0.3019172	0.3019172
OpenMP+MPI	1	2	0.0955154	0.0158181	0.0037124	0.0070128	0.2633223	0.1316611
OpenMP+MPI	1	4	0.1233291	0.0161515	0.0041467	0.0050630	0.2039367	0.0509842
OpenMP+MPI	2	1	0.0796527	0.0156345	0.0000000	0.0073957	0.3157623	0.1578812
OpenMP+MPI	2	2	0.0949393	0.0160257	0.0037444	0.0060632	0.2649202	0.0662301
OpenMP+MPI	2	4	0.1323761	0.0163132	0.0042282	0.0081830	0.1899991	0.0237499
OpenMP+MPI	4	1	0.0780180	0.0155966	0.0000000	0.0056550	0.3223786	0.0805946
OpenMP+MPI	4	2	0.0967604	0.0162725	0.0038409	0.0061258	0.2599342	0.0324918
OpenMP+MPI	4	4	0.1254837	0.0161921	0.0044060	0.0100369	0.2004350	0.0125272
OpenMP+MPI	8	1	0.0799073	0.0155712	0.0000000	0.0063882	0.3147563	0.0393445
OpenMP+MPI	8	2	0.0997410	0.0164328	0.0037631	0.0093224	0.2521664	0.0157604



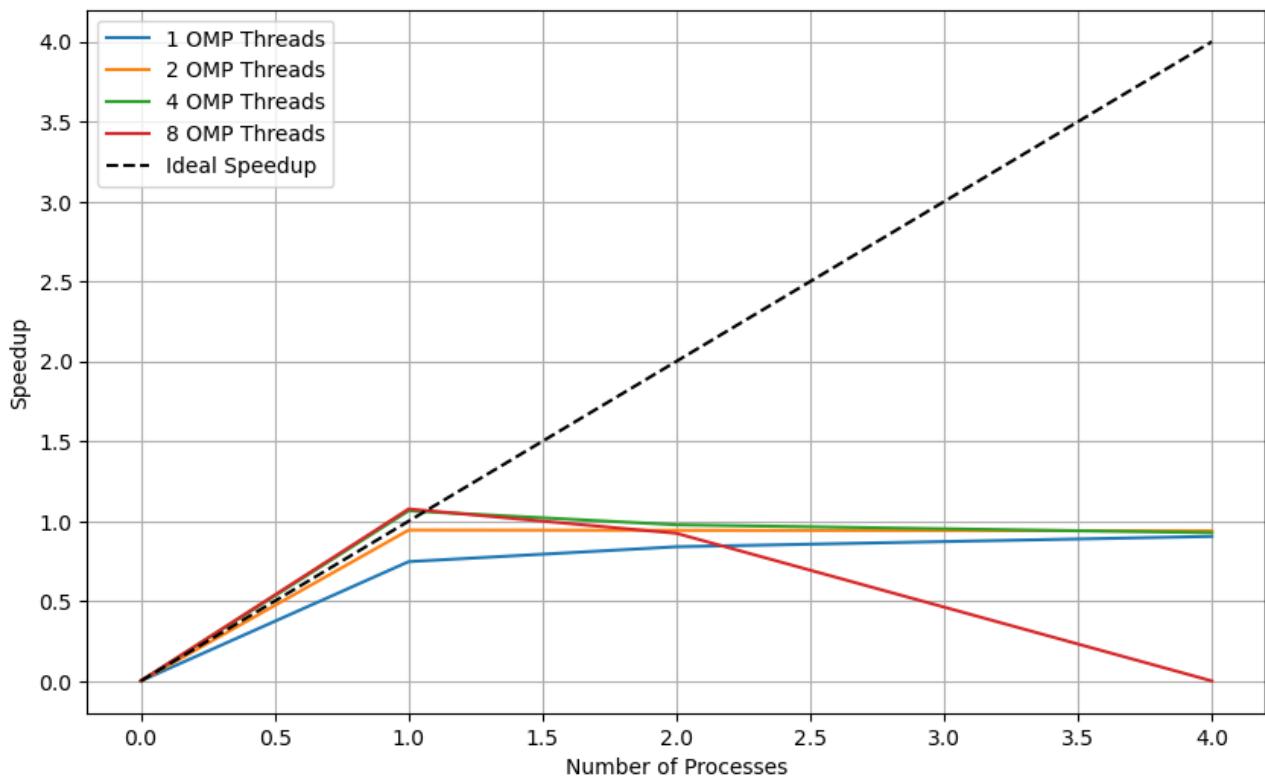
## 7552 nodes & 14260151 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6043514	0.3795036	0.0000000	0.2248215	1.0000000	1.0000000
OpenMP+MPI	1	1	0.7396980	0.3844415	0.0000000	0.2991729	0.8170245	0.8170245
OpenMP+MPI	1	2	0.7214307	0.3914987	0.0949118	0.1653305	0.8377124	0.4188562
OpenMP+MPI	1	4	0.7056527	0.3982627	0.1063473	0.1035777	0.8564431	0.2141108
OpenMP+MPI	2	1	0.5994117	0.3836627	0.0000000	0.1584209	1.0082409	0.5041204
OpenMP+MPI	2	2	0.6661765	0.3923644	0.0954338	0.1076938	0.9071941	0.2267985
OpenMP+MPI	2	4	0.6753893	0.4004323	0.1063891	0.0767724	0.8948193	0.1118524
OpenMP+MPI	4	1	0.5431583	0.3856106	0.0000000	0.0986735	1.1126617	0.2781654
OpenMP+MPI	4	2	0.6346146	0.3923061	0.0953829	0.0773552	0.9523125	0.1190391
OpenMP+MPI	4	4	0.6752548	0.3984181	0.1063295	0.0739299	0.8949976	0.0559373
OpenMP+MPI	8	1	0.5216871	0.3850786	0.0000000	0.0794266	1.1584556	0.1448070
OpenMP+MPI	8	2	0.6403920	0.3922255	0.0959965	0.0812615	0.9437210	0.0589826



## 13568 nodes & 46014096 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.0129378	1.2848842	0.0000000	0.7280221	1.0000000	1.0000000
OpenMP+MPI	1	1	2.6960687	1.3090785	0.0000000	1.3291950	0.7466196	0.7466196
OpenMP+MPI	1	2	2.3971311	1.3292792	0.3031302	0.6918862	0.8397279	0.4198639
OpenMP+MPI	1	4	2.2253545	1.3527909	0.3375355	0.4329795	0.9045470	0.2261368
OpenMP+MPI	2	1	2.1317815	1.3043843	0.0000000	0.7685223	0.9442514	0.4721257
OpenMP+MPI	2	2	2.1351468	1.3274420	0.3022480	0.4358778	0.9427632	0.2356908
OpenMP+MPI	2	4	2.1417321	1.3536803	0.3380240	0.3520817	0.9398644	0.1174831
OpenMP+MPI	4	1	1.8906845	1.3047493	0.0000000	0.5284983	1.0646609	0.2661652
OpenMP+MPI	4	2	2.0595772	1.3268270	0.3037191	0.3573045	0.9773549	0.1221694
OpenMP+MPI	4	4	2.1708140	1.3564566	0.3379235	0.3725134	0.9272733	0.0579546
OpenMP+MPI	8	1	1.8704318	1.3064471	0.0000000	0.5059020	1.0761888	0.1345236
OpenMP+MPI	8	2	2.1783133	1.3316904	0.3049593	0.4695963	0.9240809	0.0577551



## 5.4 Optimization 3

Optimization 3 does not introduce an improvement in speedup and performance compared to O0.

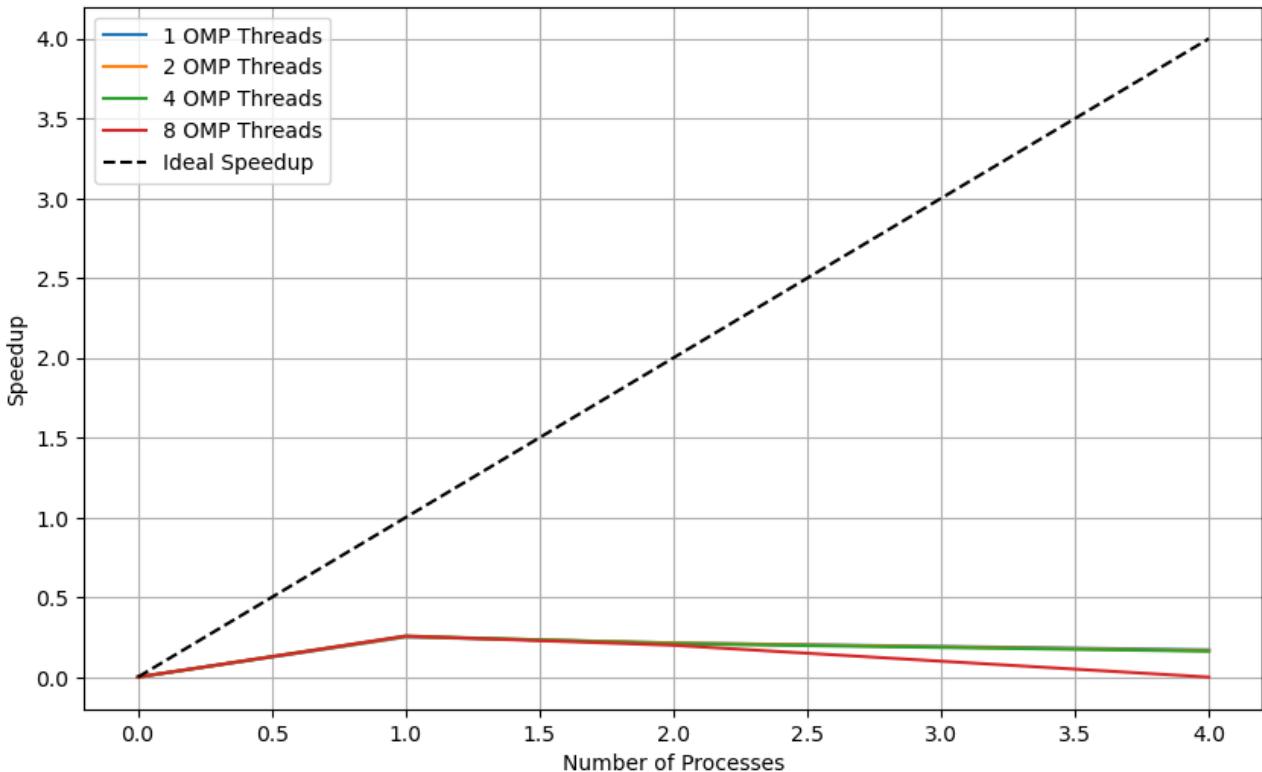
The algorithm remains efficient for large and mainly for medium, type 3, graphs.

Optimization 0 remains the one with the best performance

### 5.4.1 Type 0

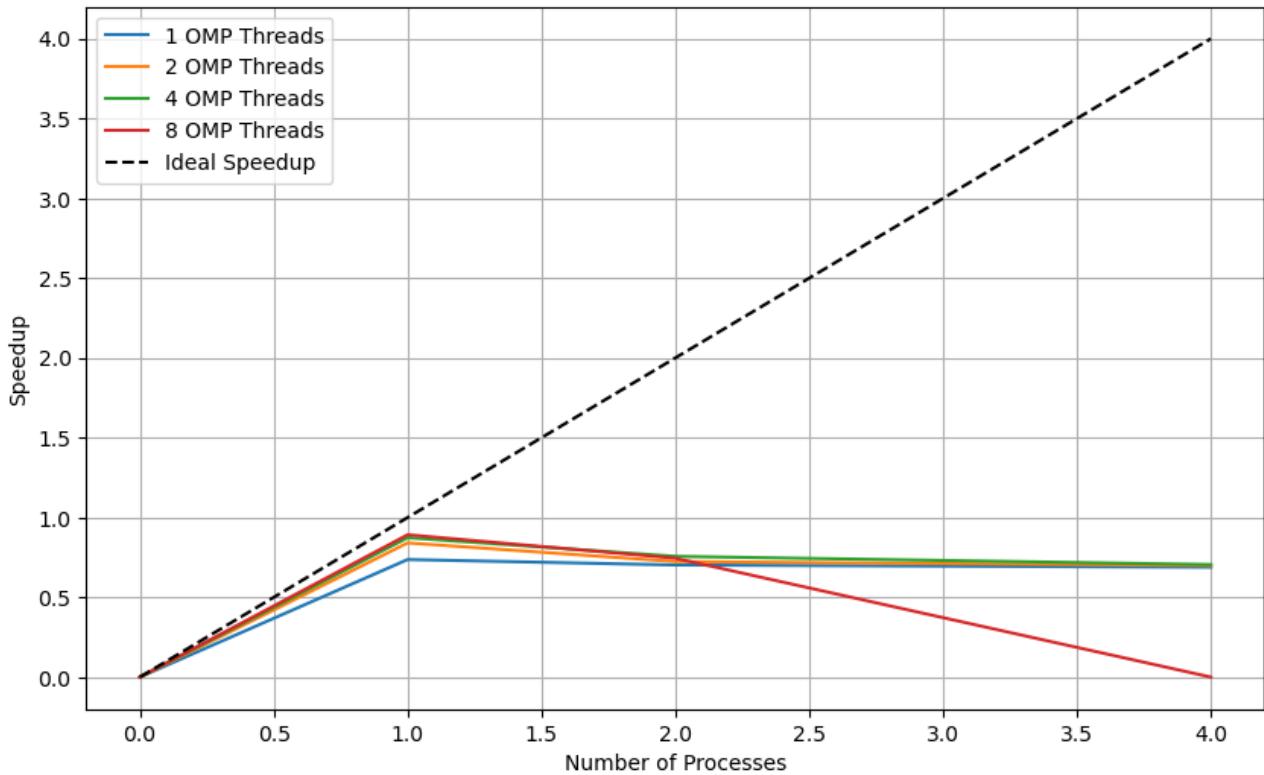
**1536 nodes & 2357760 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0209451	0.0178254	0.0000000	0.0030939	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0833953	0.0171962	0.0000000	0.0067571	0.2511547	0.2511547
OpenMP+MPI	1	2	0.0978265	0.0181926	0.0038132	0.0054394	0.2141048	0.1070524
OpenMP+MPI	1	4	0.1226348	0.0174936	0.0043939	0.0048376	0.1707927	0.0426982
OpenMP+MPI	2	1	0.0826077	0.0170147	0.0000000	0.0053527	0.2535495	0.1267748
OpenMP+MPI	2	2	0.0979830	0.0176549	0.0037892	0.0063826	0.2137629	0.0534407
OpenMP+MPI	2	4	0.1264964	0.0191924	0.0044598	0.0077381	0.1655789	0.0206974
OpenMP+MPI	4	1	0.0813989	0.0170203	0.0000000	0.0051349	0.2573148	0.0643287
OpenMP+MPI	4	2	0.1002285	0.0185239	0.0039181	0.0062312	0.2089738	0.0261217
OpenMP+MPI	4	4	0.1291599	0.0186795	0.0044503	0.0090061	0.1621643	0.0101353
OpenMP+MPI	8	1	0.0814345	0.0169647	0.0000000	0.0059597	0.2572021	0.0321503
OpenMP+MPI	8	2	0.1050705	0.0192035	0.0040202	0.0082900	0.1993436	0.0124590



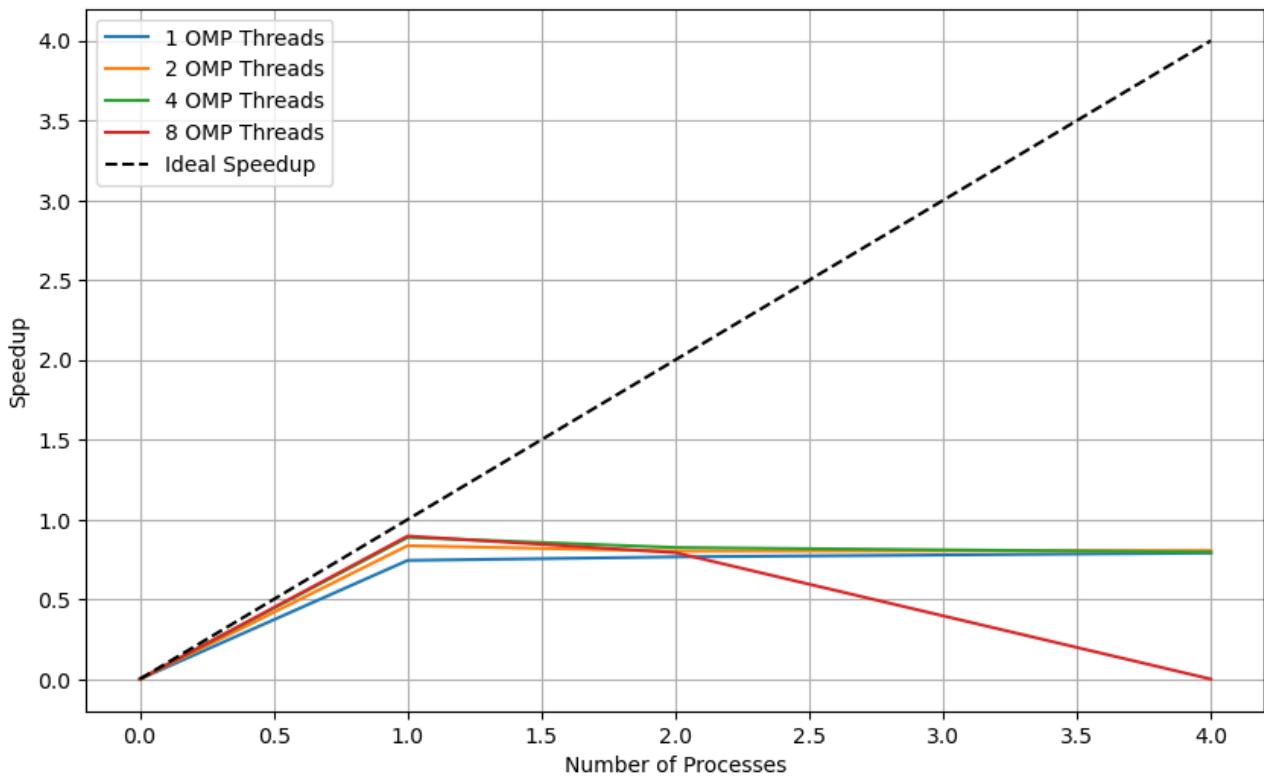
## 7552 nodes & 57025152 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5237787	0.4485539	0.0000000	0.0752013	1.000000	1.000000
OpenMP+MPI	1	1	0.7110386	0.4501924	0.0000000	0.2029952	0.7366389	0.7366389
OpenMP+MPI	1	2	0.7456155	0.4636036	0.0956239	0.1127307	0.7024783	0.3512391
OpenMP+MPI	1	4	0.7608168	0.4633307	0.1072029	0.0921835	0.6884426	0.1721106
OpenMP+MPI	2	1	0.6228999	0.4494939	0.0000000	0.1147387	0.8408713	0.4204356
OpenMP+MPI	2	2	0.7235426	0.4576820	0.0958736	0.0960187	0.7239085	0.1809771
OpenMP+MPI	2	4	0.7495526	0.4741555	0.1076223	0.0729903	0.6987884	0.0873486
OpenMP+MPI	4	1	0.5996480	0.4494001	0.0000000	0.0920597	0.8734769	0.2183692
OpenMP+MPI	4	2	0.6922505	0.4568551	0.0956288	0.0672953	0.7566317	0.0945790
OpenMP+MPI	4	4	0.7429291	0.4654999	0.1077038	0.0700961	0.7050183	0.0440636
OpenMP+MPI	8	1	0.5872217	0.4495075	0.0000000	0.0786125	0.8919607	0.1114951
OpenMP+MPI	8	2	0.7034083	0.4590689	0.0954705	0.0770515	0.7446296	0.0465394



## 13568 nodes & 184077056 edges

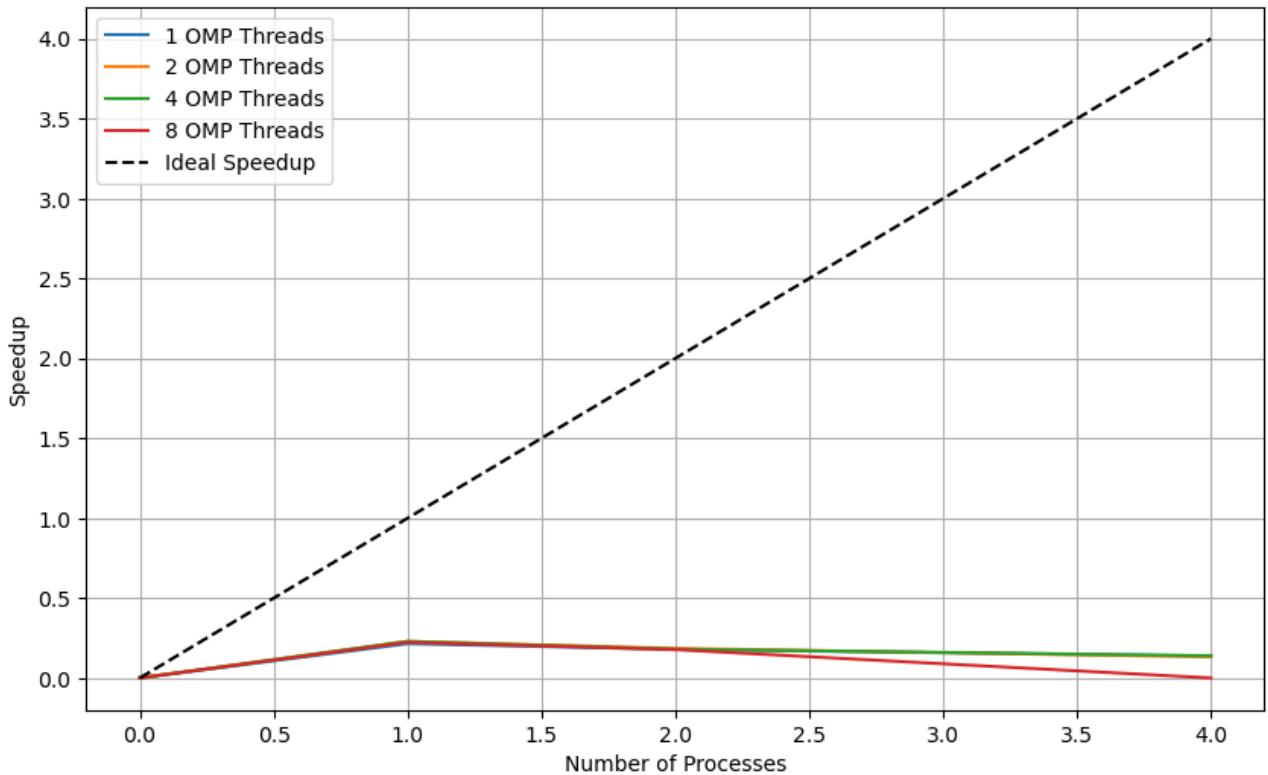
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.3147068	2.0656339	0.0000000	0.2490404	1.0000000	1.0000000
OpenMP+MPI	1	1	3.1128031	2.0609523	0.0000000	0.9929838	0.7436085	0.7436085
OpenMP+MPI	1	2	3.0231803	2.0847724	0.3030396	0.5646367	0.7656529	0.3828265
OpenMP+MPI	1	4	2.9301680	2.0939849	0.3381505	0.3871050	0.7899570	0.1974893
OpenMP+MPI	2	1	2.7691434	2.0617479	0.0000000	0.6487730	0.8358927	0.4179464
OpenMP+MPI	2	2	2.8870515	2.0912379	0.3031101	0.4219143	0.8017546	0.2004386
OpenMP+MPI	2	4	2.8741511	2.0810473	0.3372924	0.3597399	0.8053532	0.1006692
OpenMP+MPI	4	1	2.6077788	2.0637553	0.0000000	0.4848870	0.8876162	0.2219041
OpenMP+MPI	4	2	2.8078749	2.0789125	0.3031339	0.3555677	0.8243625	0.1030453
OpenMP+MPI	4	4	2.9104878	2.0893890	0.3385248	0.3811729	0.7952986	0.0497062
OpenMP+MPI	8	1	2.5826245	2.0564355	0.0000000	0.4671354	0.8962614	0.1120327
OpenMP+MPI	8	2	2.9203143	2.0755409	0.3030881	0.4704035	0.7926225	0.0495389



### 5.4.2 Type 1

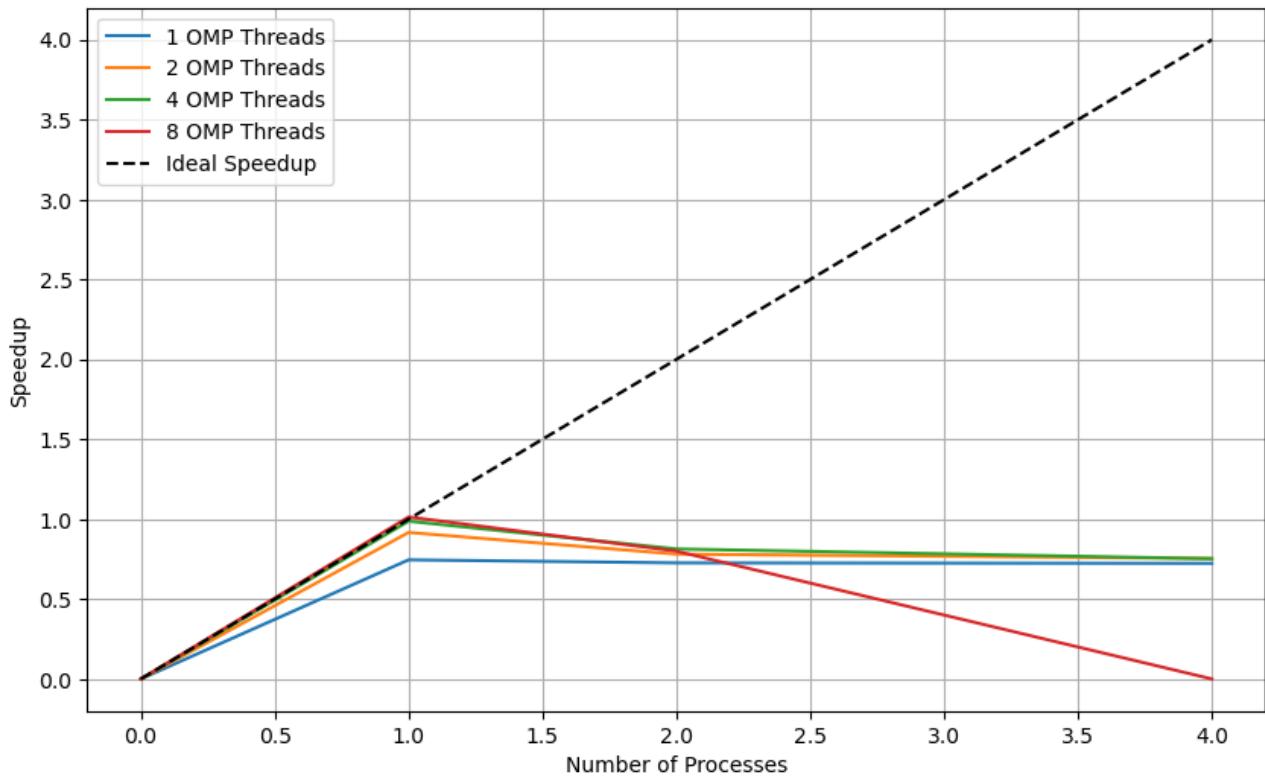
**1536 nodes & 1061043 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0172293	0.0124619	0.0000000	0.0047413	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0803231	0.0117431	0.0000000	0.0073365	0.2144994	0.2144994
OpenMP+MPI	1	2	0.0959377	0.0120169	0.0037918	0.0056502	0.1795880	0.0897940
OpenMP+MPI	1	4	0.1259529	0.0125811	0.0042021	0.0047882	0.1367913	0.0341978
OpenMP+MPI	2	1	0.0753997	0.0117319	0.0000000	0.0056883	0.2285059	0.1142529
OpenMP+MPI	2	2	0.0930333	0.0120395	0.0037639	0.0060749	0.1851947	0.0462987
OpenMP+MPI	2	4	0.1310677	0.0123852	0.0043351	0.0076555	0.1314532	0.0164317
OpenMP+MPI	4	1	0.0753161	0.0117024	0.0000000	0.0052575	0.2287593	0.0571898
OpenMP+MPI	4	2	0.0955181	0.0123033	0.0038843	0.0052079	0.1803770	0.0225471
OpenMP+MPI	4	4	0.1240032	0.0127507	0.0043082	0.0094615	0.1389421	0.0086839
OpenMP+MPI	8	1	0.0772055	0.0117266	0.0000000	0.0063463	0.2231612	0.0278952
OpenMP+MPI	8	2	0.0971385	0.0123623	0.0039072	0.0085503	0.1773681	0.0110855



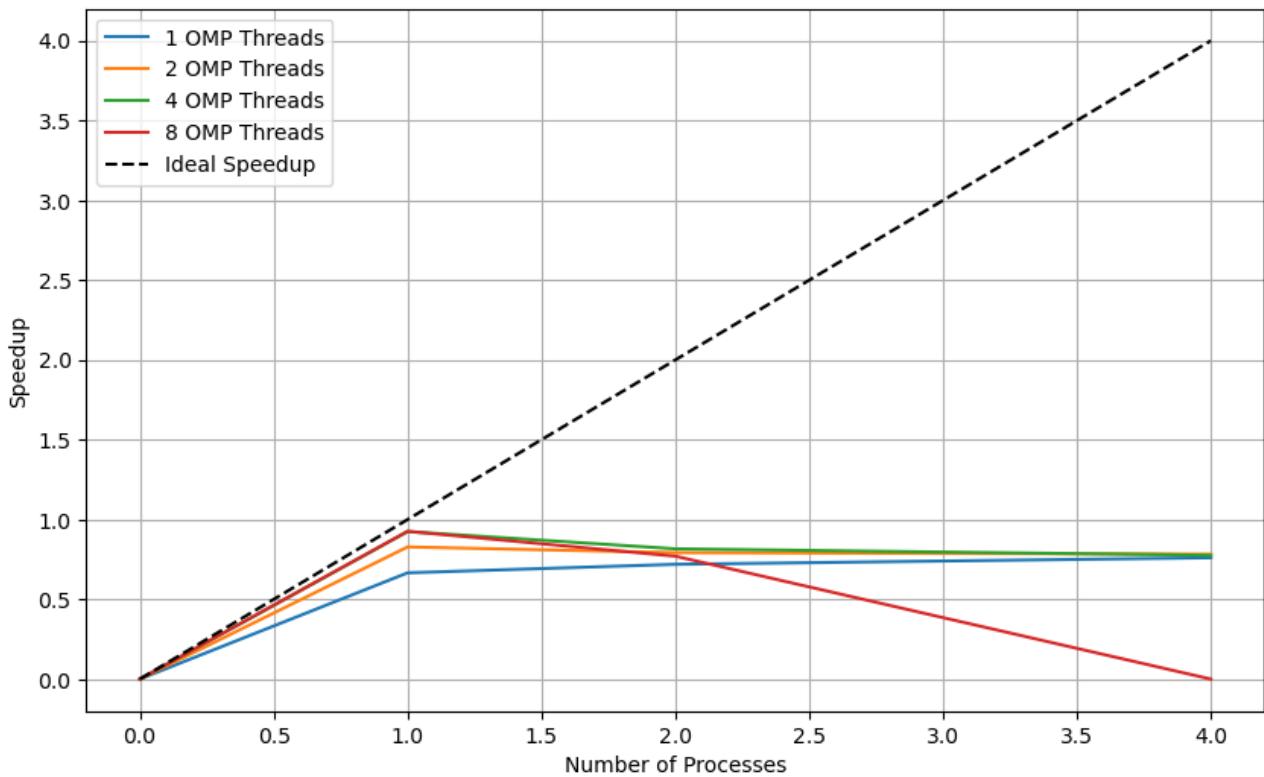
## 7552 nodes & 25660141 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4371048	0.2917672	0.0000000	0.1453100	1.0000000	1.0000000
OpenMP+MPI	1	1	0.5863579	0.2931741	0.0000000	0.2350273	0.7454573	0.7454573
OpenMP+MPI	1	2	0.6017890	0.2993735	0.0954607	0.1333533	0.7263423	0.3631711
OpenMP+MPI	1	4	0.6039490	0.3023054	0.1069616	0.0943667	0.7237446	0.1809361
OpenMP+MPI	2	1	0.4767940	0.2919966	0.0000000	0.1262518	0.9167582	0.4583791
OpenMP+MPI	2	2	0.5595194	0.2981555	0.0954067	0.0960189	0.7812147	0.1953037
OpenMP+MPI	2	4	0.5798480	0.3020134	0.1066436	0.0746309	0.7538265	0.0942283
OpenMP+MPI	4	1	0.4427793	0.2921713	0.0000000	0.0918115	0.9871843	0.2467961
OpenMP+MPI	4	2	0.5372755	0.2985388	0.0956367	0.0726513	0.8135581	0.1016948
OpenMP+MPI	4	4	0.5808178	0.3036551	0.1066416	0.0719445	0.7525678	0.0470355
OpenMP+MPI	8	1	0.4311691	0.2932897	0.0000000	0.0788876	1.0137664	0.1267208
OpenMP+MPI	8	2	0.5465373	0.3024086	0.0957691	0.0786105	0.7997712	0.0499857



## 13568 nodes & 82829469 edges

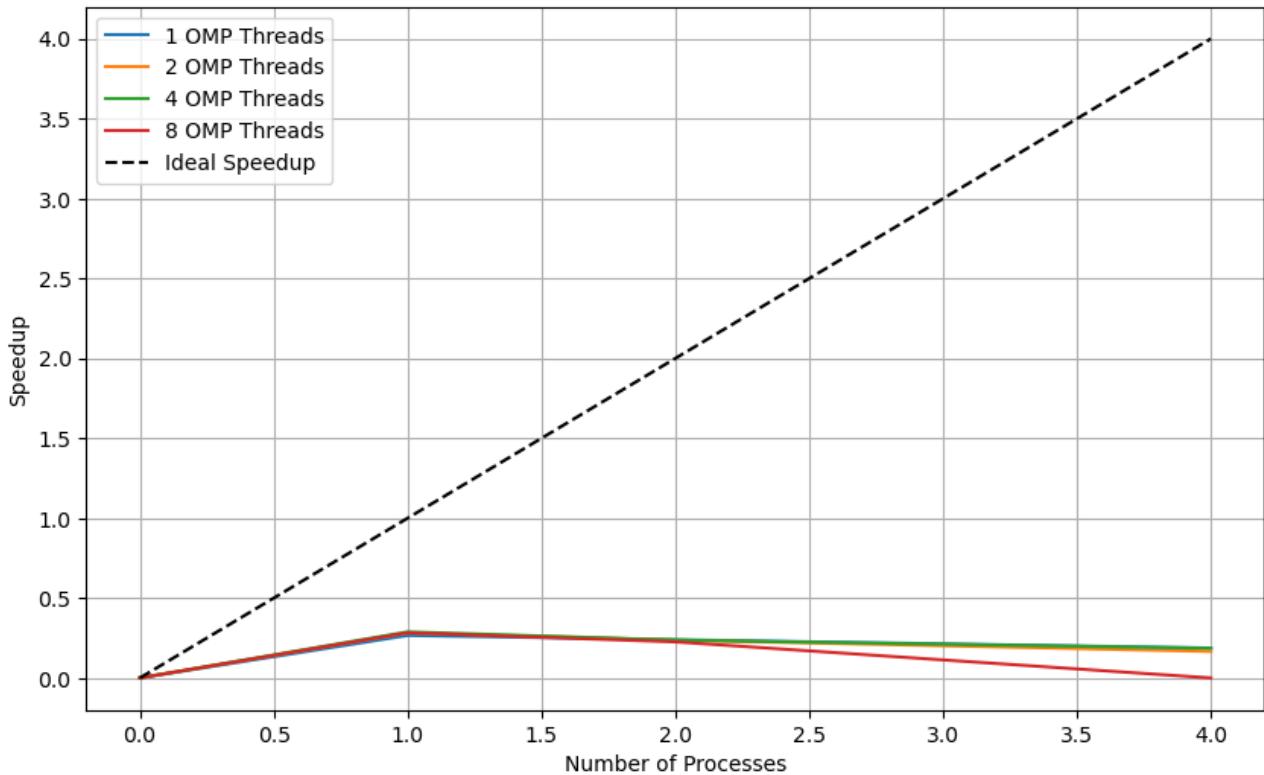
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4508061	1.0183621	0.0000000	0.4324105	1.0000000	1.0000000
OpenMP+MPI	1	1	2.1789093	1.0300757	0.0000000	1.0904264	0.6658405	0.6658405
OpenMP+MPI	1	2	2.0171735	1.0446292	0.3036096	0.5954100	0.7192272	0.3596136
OpenMP+MPI	1	4	1.9083009	1.0672551	0.3382837	0.3980146	0.7602607	0.1900652
OpenMP+MPI	2	1	1.7520705	1.0231449	0.0000000	0.6696054	0.8280523	0.4140262
OpenMP+MPI	2	2	1.8311455	1.0405094	0.3029069	0.4168018	0.7922943	0.1980736
OpenMP+MPI	2	4	1.8536805	1.0621916	0.3378329	0.3548951	0.7826624	0.0978328
OpenMP+MPI	4	1	1.5695087	1.0218009	0.0000000	0.4883433	0.9243695	0.2310924
OpenMP+MPI	4	2	1.7772415	1.0458479	0.3015379	0.3555401	0.8163246	0.1020406
OpenMP+MPI	4	4	1.8697211	1.0550506	0.3366640	0.3840517	0.7759478	0.0484967
OpenMP+MPI	8	1	1.5667798	1.0243497	0.0000000	0.4835201	0.9259796	0.1157474
OpenMP+MPI	8	2	1.8840524	1.0416343	0.3027153	0.4691628	0.7700455	0.0481278



### 5.4.3 Type 2

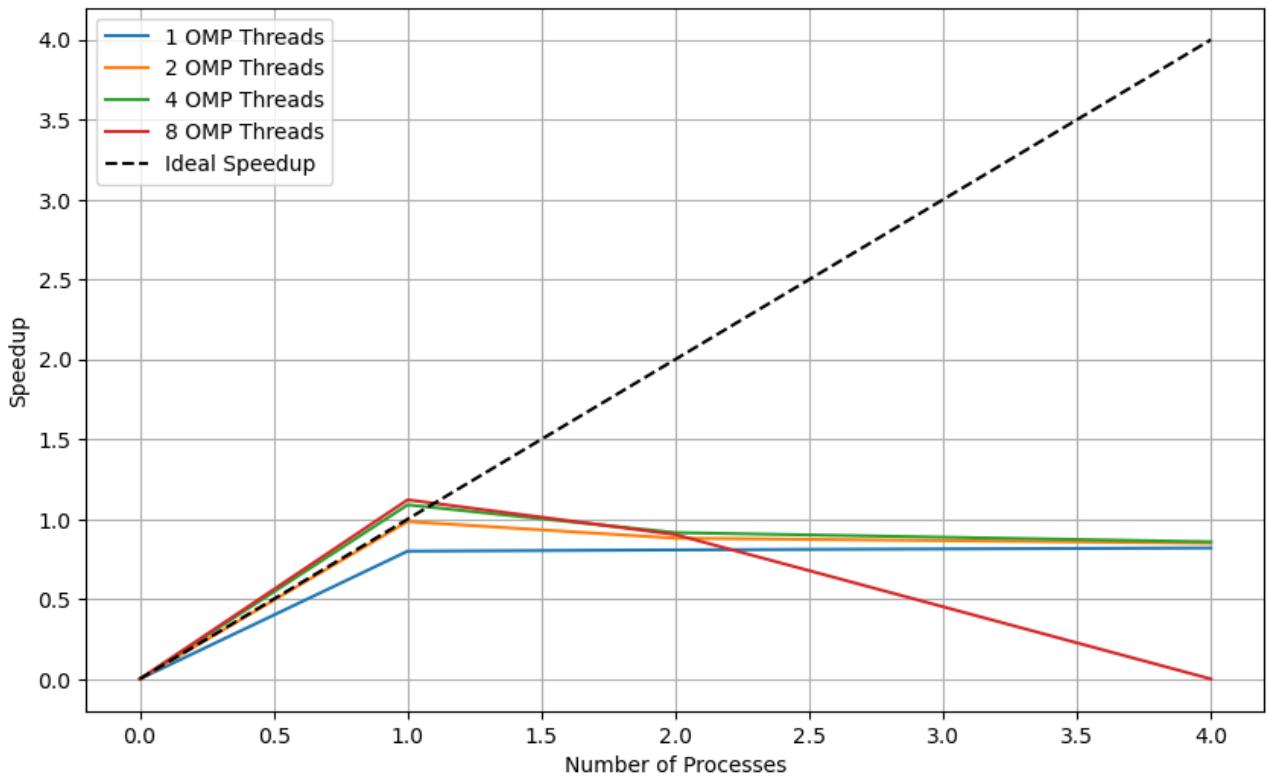
**1536 nodes & 790416 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0228476	0.0154823	0.0000000	0.0073375	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0859267	0.0146750	0.0000000	0.0095025	0.2658963	0.2658963
OpenMP+MPI	1	2	0.0954225	0.0146935	0.0037056	0.0064885	0.2394361	0.1197181
OpenMP+MPI	1	4	0.1220285	0.0148892	0.0041699	0.0049125	0.1872316	0.0468079
OpenMP+MPI	2	1	0.0807391	0.0146049	0.0000000	0.0068621	0.2829805	0.1414902
OpenMP+MPI	2	2	0.0964281	0.0152538	0.0037831	0.0073415	0.2369392	0.0592348
OpenMP+MPI	2	4	0.1376624	0.0162029	0.0042664	0.0083184	0.1659683	0.0207460
OpenMP+MPI	4	1	0.0791721	0.0146663	0.0000000	0.0052999	0.2885816	0.0721454
OpenMP+MPI	4	2	0.0964471	0.0152859	0.0038709	0.0059961	0.2368926	0.0296116
OpenMP+MPI	4	4	0.1231071	0.0151359	0.0042125	0.0083109	0.1855913	0.0115995
OpenMP+MPI	8	1	0.0807462	0.0146035	0.0000000	0.0063779	0.2829557	0.0353695
OpenMP+MPI	8	2	0.1011751	0.0149643	0.0037707	0.0079747	0.2258223	0.0141139



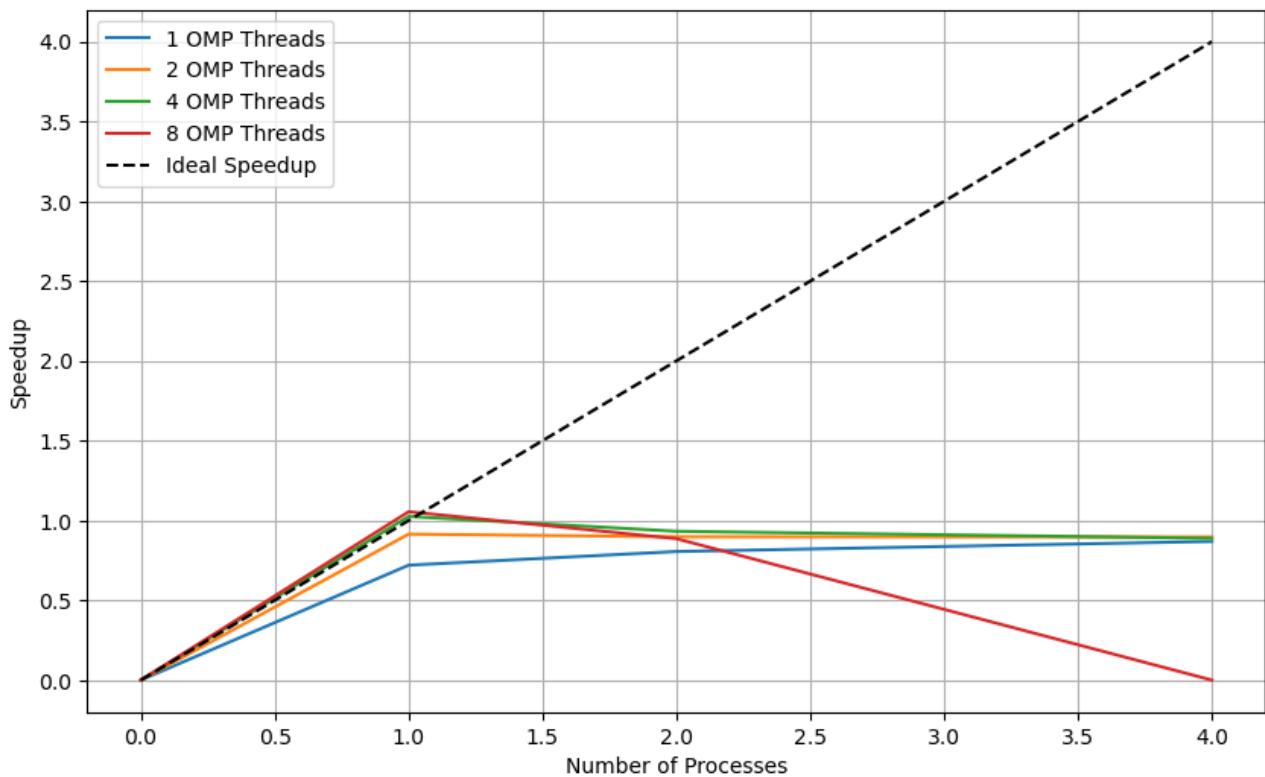
## 7552 nodes & 19101950 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5591937	0.3584046	0.0000000	0.2007601	1.0000000	1.0000000
OpenMP+MPI	1	1	0.6993591	0.3605919	0.0000000	0.2801094	0.7995802	0.7995802
OpenMP+MPI	1	2	0.6924614	0.3691060	0.0957680	0.1564557	0.8075449	0.4037724
OpenMP+MPI	1	4	0.6822391	0.3733787	0.1070958	0.1003762	0.8196447	0.2049112
OpenMP+MPI	2	1	0.5678592	0.3604121	0.0000000	0.1489463	0.9847400	0.4923700
OpenMP+MPI	2	2	0.6346480	0.3655418	0.0952829	0.1019877	0.8811084	0.2202771
OpenMP+MPI	2	4	0.6580047	0.3739859	0.1065339	0.0773577	0.8498324	0.1062290
OpenMP+MPI	4	1	0.5132633	0.3591935	0.0000000	0.0952340	1.0894870	0.2723718
OpenMP+MPI	4	2	0.6106747	0.3666328	0.0953981	0.0775523	0.9156981	0.1144623
OpenMP+MPI	4	4	0.6515014	0.3734704	0.1072255	0.0747227	0.8583154	0.0536447
OpenMP+MPI	8	1	0.4988333	0.3597037	0.0000000	0.0803521	1.1210032	0.1401254
OpenMP+MPI	8	2	0.6188985	0.3688651	0.0956766	0.0811749	0.9035305	0.0564707



## 13568 nodes & 61665181 edges

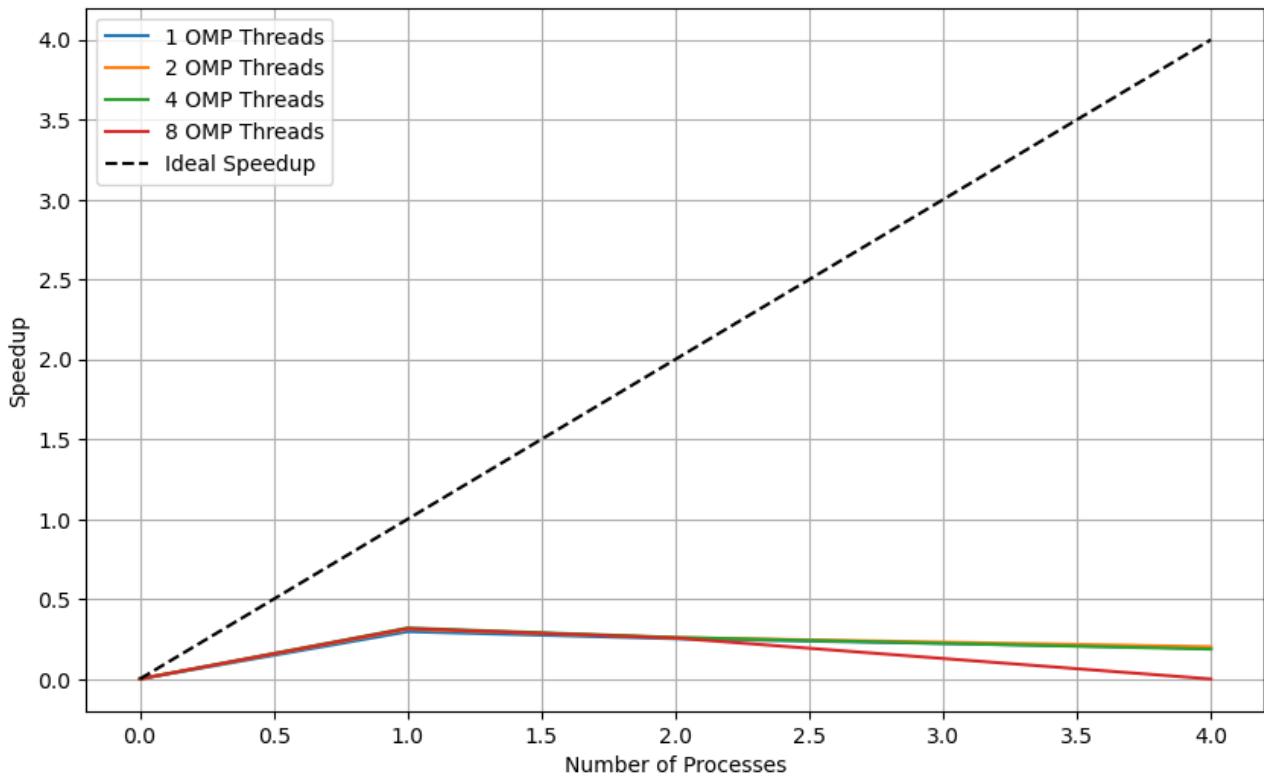
	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8570059	1.2169153	0.0000000	0.6400563	1.0000000	1.0000000
OpenMP+MPI	1	1	2.5789794	1.2283964	0.0000000	1.2915794	0.7200546	0.7200546
OpenMP+MPI	1	2	2.3050932	1.2505722	0.3037793	0.6768699	0.8056099	0.4028050
OpenMP+MPI	1	4	2.1371495	1.2733001	0.3383805	0.4258246	0.8689172	0.2172293
OpenMP+MPI	2	1	2.0303337	1.2283632	0.0000000	0.7425799	0.9146309	0.4573155
OpenMP+MPI	2	2	2.0682171	1.2487121	0.3033598	0.4432907	0.8978777	0.2244694
OpenMP+MPI	2	4	2.0708151	1.2754334	0.3381523	0.3582133	0.8967512	0.1120939
OpenMP+MPI	4	1	1.8111025	1.2397037	0.0000000	0.5117829	1.0253456	0.2563364
OpenMP+MPI	4	2	1.9913965	1.2479819	0.3037032	0.3684695	0.9325144	0.1165643
OpenMP+MPI	4	4	2.0900119	1.2714333	0.3370440	0.3824089	0.8885145	0.0555322
OpenMP+MPI	8	1	1.7600749	1.2224766	0.0000000	0.4785679	1.0550721	0.1318840
OpenMP+MPI	8	2	2.0953733	1.2454624	0.3025543	0.4764287	0.8862411	0.0553901



#### 5.4.4 Type 3

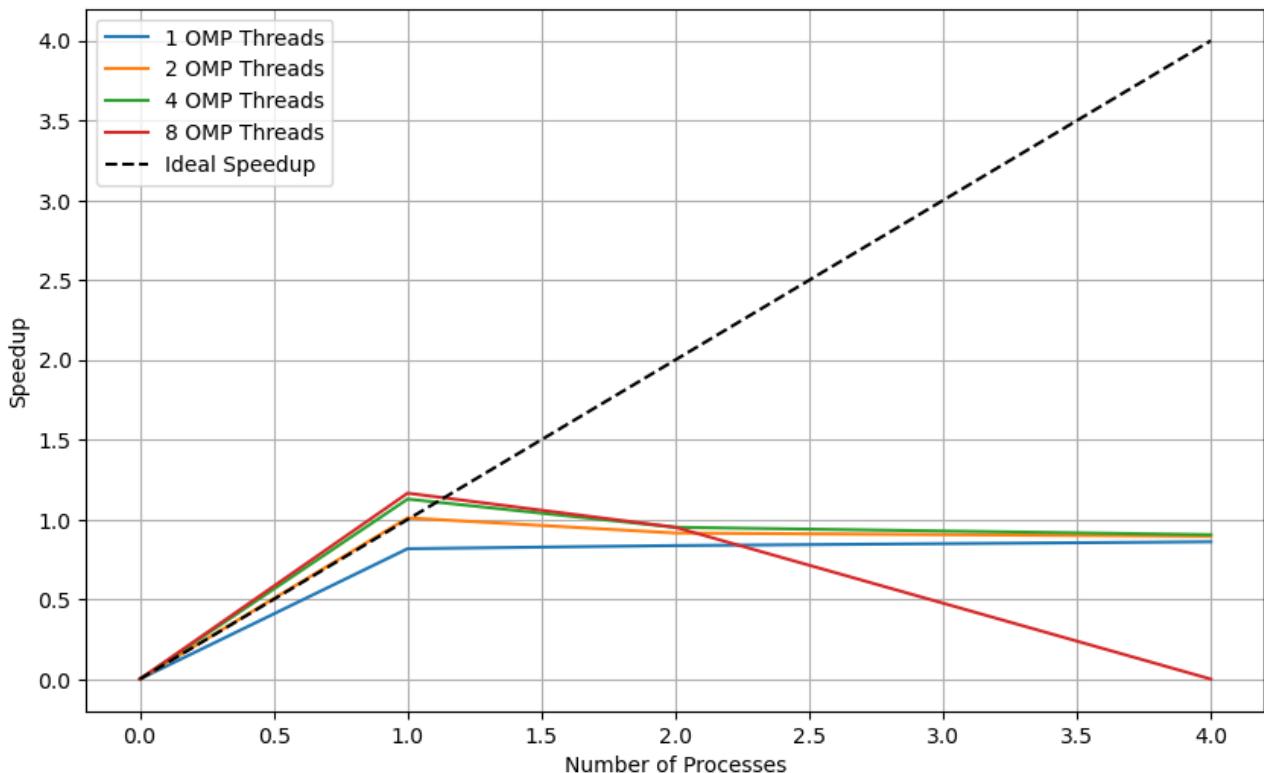
**1536 nodes & 588142 edges**

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0254169	0.0165269	0.0000000	0.0088626	1.0000000	1.0000000
OpenMP+MPI	1	1	0.0858719	0.0157605	0.0000000	0.0109709	0.2959867	0.2959867
OpenMP+MPI	1	2	0.1008042	0.0162527	0.0038810	0.0075199	0.2521416	0.1260708
OpenMP+MPI	1	4	0.1324831	0.0162266	0.0043147	0.0053773	0.1918504	0.0479626
OpenMP+MPI	2	1	0.0814345	0.0156113	0.0000000	0.0074422	0.3121149	0.1560575
OpenMP+MPI	2	2	0.0982218	0.0163065	0.0038497	0.0065929	0.2587708	0.0646927
OpenMP+MPI	2	4	0.1257411	0.0165181	0.0042694	0.0073478	0.2021370	0.0252671
OpenMP+MPI	4	1	0.0796581	0.0155847	0.0000000	0.0057689	0.3190752	0.0797688
OpenMP+MPI	4	2	0.0978647	0.0165829	0.0039151	0.0065180	0.2597151	0.0324644
OpenMP+MPI	4	4	0.1361111	0.0163667	0.0043939	0.0097279	0.1867367	0.0116710
OpenMP+MPI	8	1	0.0804837	0.0156511	0.0000000	0.0065018	0.3158024	0.0394753
OpenMP+MPI	8	2	0.0992376	0.0162687	0.0038450	0.0090541	0.2561220	0.0160076



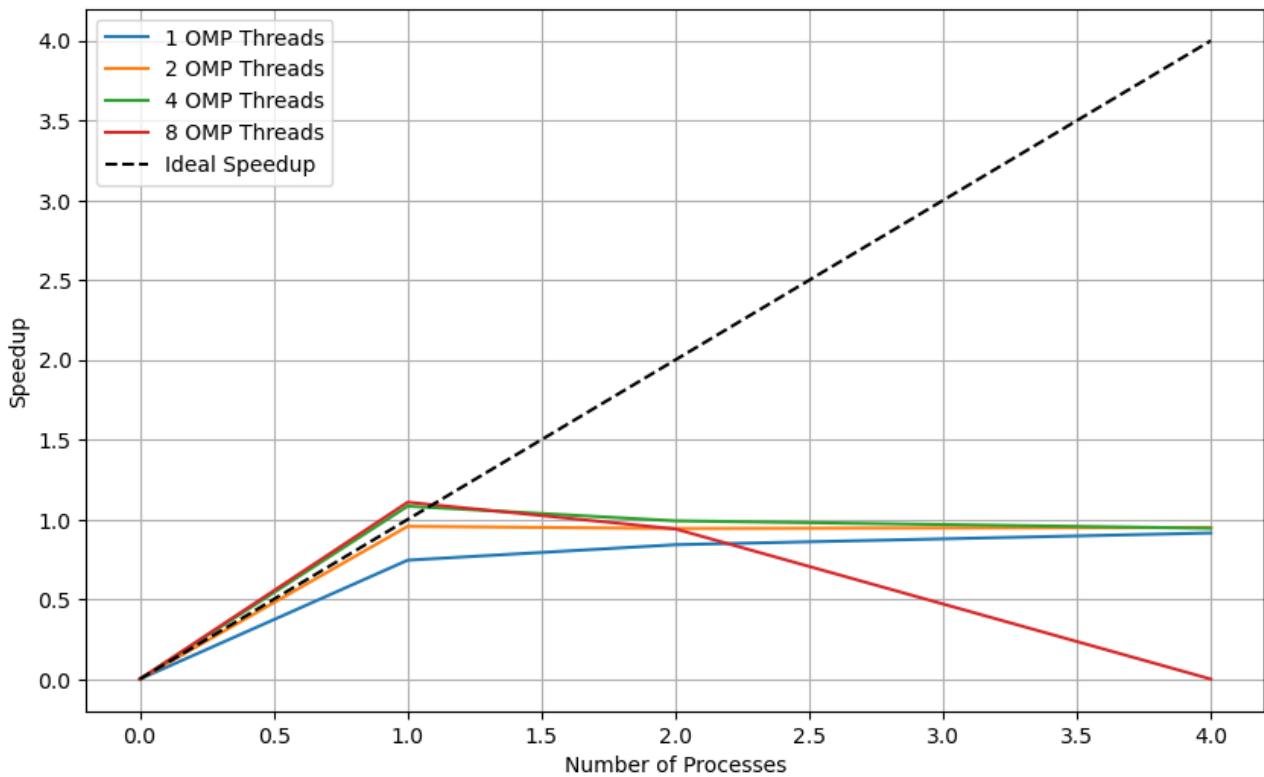
## 7552 nodes & 14260151 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6107635	0.3838983	0.0000000	0.2268377	1.0000000	1.0000000
OpenMP+MPI	1	1	0.7475200	0.3830120	0.0000000	0.3063148	0.8170531	0.8170531
OpenMP+MPI	1	2	0.7300691	0.3922818	0.0956545	0.1691069	0.8365832	0.4182916
OpenMP+MPI	1	4	0.7098637	0.3985559	0.1069237	0.1060855	0.8603955	0.2150989
OpenMP+MPI	2	1	0.6047256	0.3834597	0.0000000	0.1621615	1.0099846	0.5049923
OpenMP+MPI	2	2	0.6673685	0.3914210	0.0956033	0.1078958	0.9151818	0.2287954
OpenMP+MPI	2	4	0.6826013	0.3987279	0.1071845	0.0804256	0.8947589	0.1118449
OpenMP+MPI	4	1	0.5415343	0.3824989	0.0000000	0.0996311	1.1278391	0.2819598
OpenMP+MPI	4	2	0.6420221	0.3931797	0.0954281	0.0807347	0.9513123	0.1189140
OpenMP+MPI	4	4	0.6752830	0.4013138	0.1069187	0.0744706	0.9044557	0.0565285
OpenMP+MPI	8	1	0.5242348	0.3830999	0.0000000	0.0819307	1.1650572	0.1456322
OpenMP+MPI	8	2	0.6422214	0.3913970	0.0959172	0.0816224	0.9510171	0.0594386



## 13568 nodes & 46014096 edges

	OMP	MPI	total time	graph creation time	communication time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.0507477	1.3049343	0.0000000	0.7457805	1.0000000	1.0000000
OpenMP+MPI	1	1	2.7526332	1.3046186	0.0000000	1.3886734	0.7450131	0.7450131
OpenMP+MPI	1	2	2.4356716	1.3309676	0.3047695	0.7251997	0.8419639	0.4209820
OpenMP+MPI	1	4	2.2402074	1.3527619	0.3379133	0.4475279	0.9154276	0.2288569
OpenMP+MPI	2	1	2.1426988	1.3001253	0.0000000	0.7827033	0.9570863	0.4785432
OpenMP+MPI	2	2	2.1729029	1.3270553	0.3038253	0.4683909	0.9437825	0.2359456
OpenMP+MPI	2	4	2.1588714	1.3534511	0.3388693	0.3695097	0.9499165	0.1187396
OpenMP+MPI	4	1	1.8931871	1.2997619	0.0000000	0.5344530	1.0832250	0.2708063
OpenMP+MPI	4	2	2.0680037	1.3296829	0.3044093	0.3627861	0.9916557	0.1239570
OpenMP+MPI	4	4	2.1696772	1.3554725	0.3374403	0.3800904	0.9451856	0.0590741
OpenMP+MPI	8	1	1.8505749	1.3025110	0.0000000	0.4885751	1.1081679	0.1385210
OpenMP+MPI	8	2	2.1795735	1.3276766	0.3036874	0.4742213	0.9408940	0.0588059



## Test “OpenMP + CUDA”

Tests were performed by iterating 15 trial for each configuration.

In the following are reported test results.

The optimal block size (768 threads) is calculated by using the function `cudaOccupancyMaxPotentialBlockSize()`.

The optimal grid size is calculated dynamically by the program, based on the size of the input data

### 6.1 Optimization 0

Optimization 0 introduces an improvement in speedup and performance.

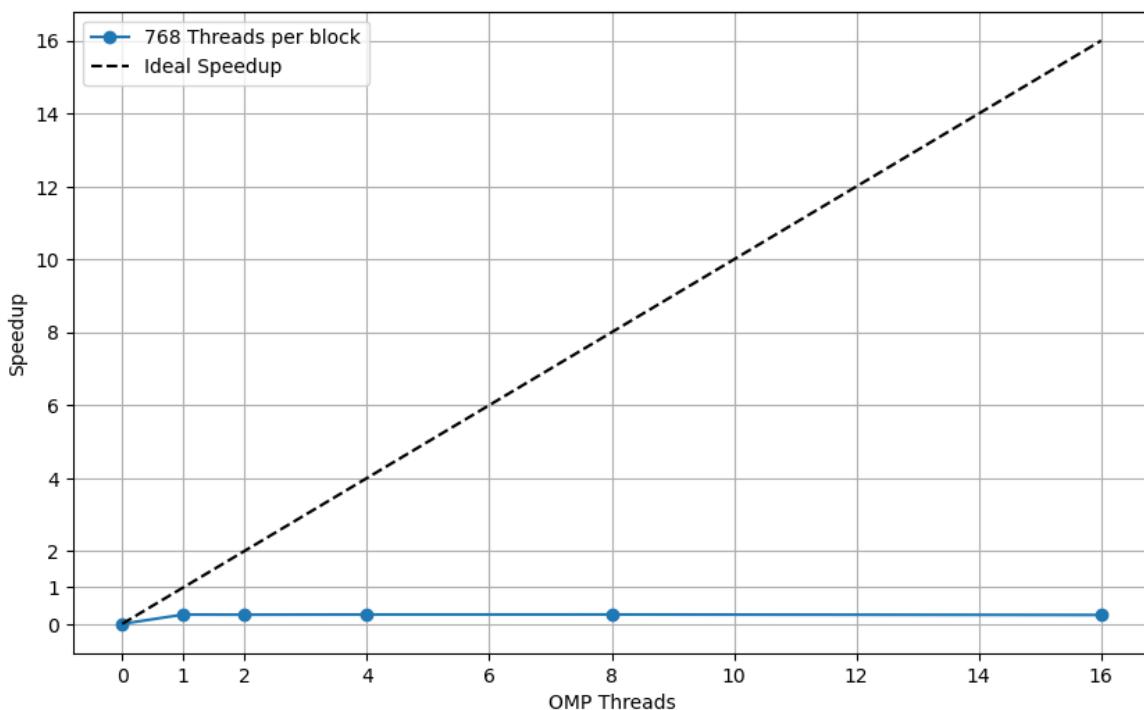
The algorithm remains efficient for medium and mainly for large, type 3, graphs.

The improvement in speedup is proportional to the size of the graph

#### 6.1.1 Type 0

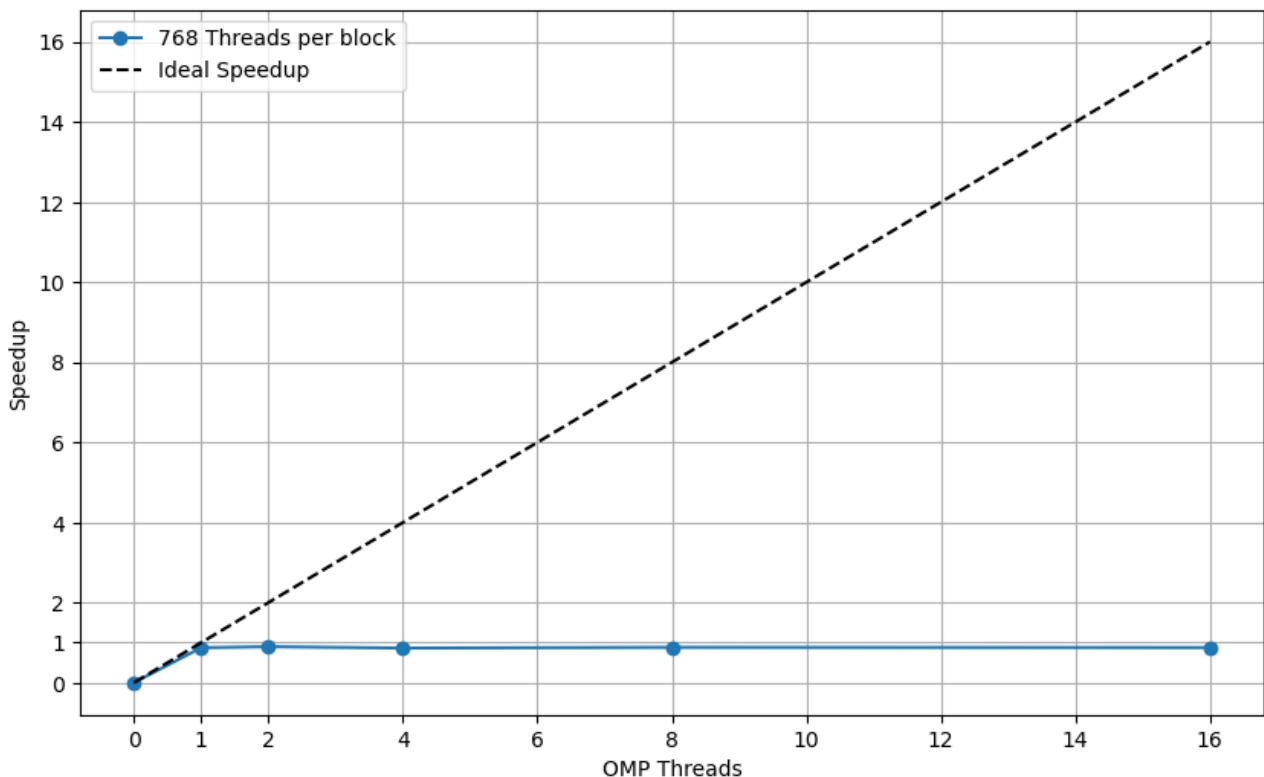
**1536 nodes & 2357760 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0270541	0.0203475	0.0000000	0.0066809	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1060341	0.0200728	0.0640603	0.0218882	0.2551449	0.2551449
OpenMp+CUDA	2	768	0.1073118	0.0196797	0.0628863	0.0247331	0.2521071	0.1260535
OpenMp+CUDA	4	768	0.1057022	0.0197121	0.0630027	0.0229747	0.2559461	0.0639865
OpenMp+CUDA	8	768	0.1053014	0.0195825	0.0628107	0.0228953	0.2569203	0.0321150
OpenMp+CUDA	16	768	0.1106298	0.0196803	0.0630355	0.0279015	0.2445459	0.0152841



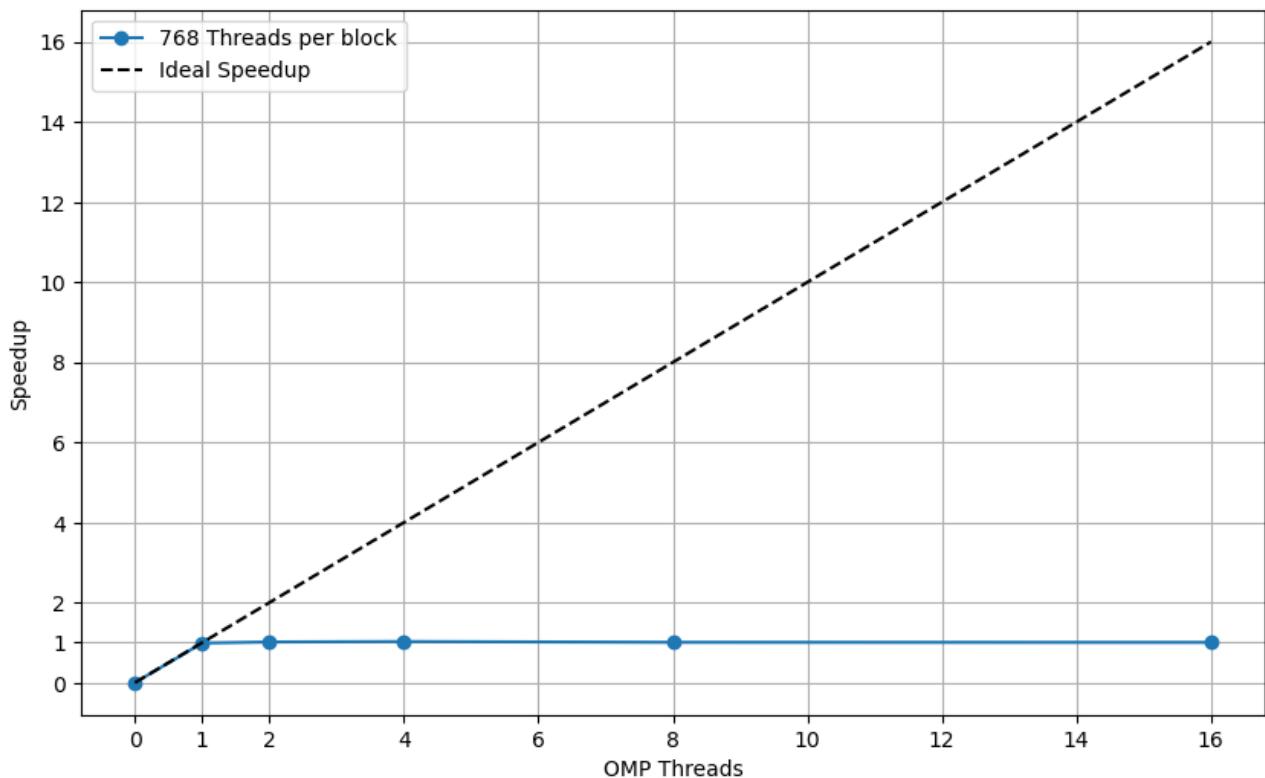
## 7552 nodes & 57025152 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6548891	0.4969229	0.0000000	0.1579379	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.7504301	0.4947141	0.0793579	0.1763066	0.8726850	0.8726850
OpenMp+CUDA	2	768	0.7281726	0.4938769	0.0794721	0.1547725	0.8993597	0.4496798
OpenMp+CUDA	4	768	0.7575801	0.4960140	0.0796809	0.1818339	0.8644486	0.2161121
OpenMp+CUDA	8	768	0.7420293	0.5025768	0.0796057	0.1597959	0.8825650	0.1103206
OpenMp+CUDA	16	768	0.7472343	0.4971706	0.0802531	0.1697587	0.8764173	0.0547761



## 13568 nodes & 184077056 edges

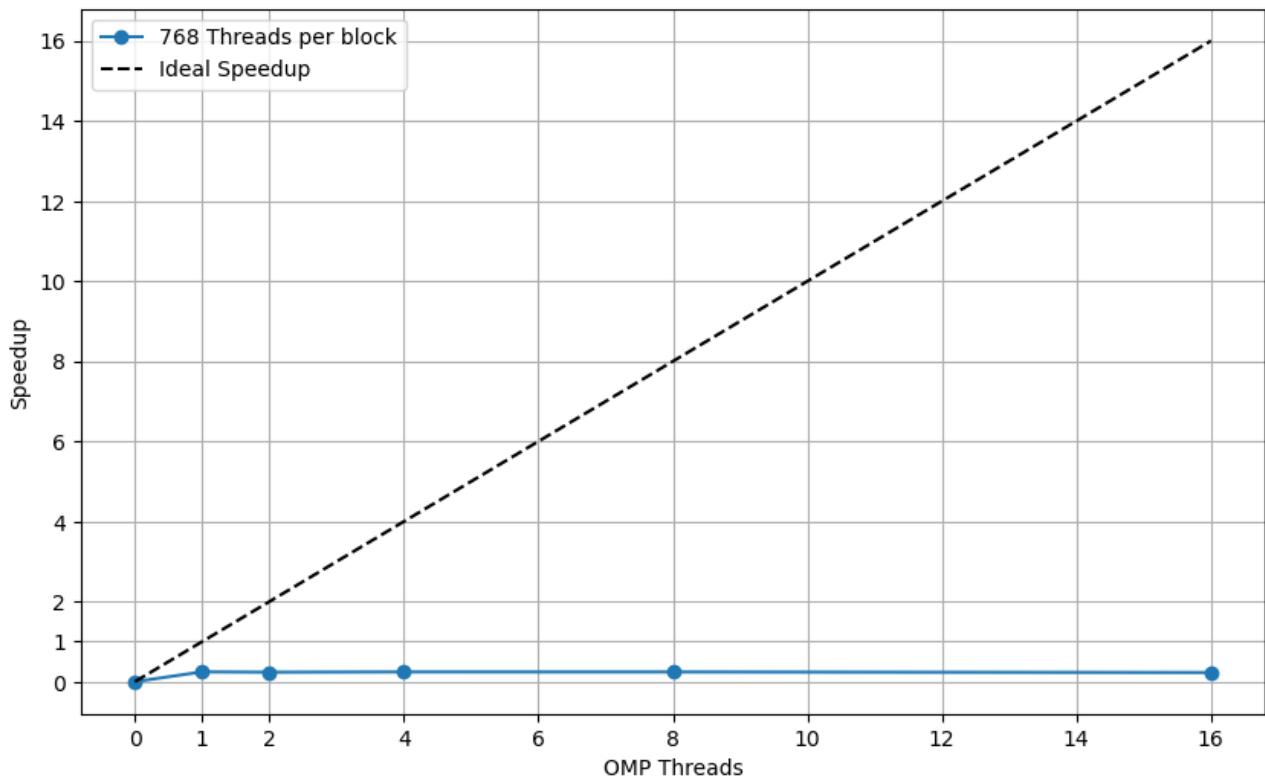
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.6802869	2.1415253	0.000000	0.5387284	1.000000	1.000000
OpenMp+CUDA	1	768	2.7121350	2.1543637	0.1202365	0.4374393	0.9882572	0.9882572
OpenMp+CUDA	2	768	2.6364543	2.1685646	0.1186511	0.3491425	1.0166256	0.5083128
OpenMp+CUDA	4	768	2.6101365	2.1542278	0.1189924	0.3368198	1.0268761	0.2567190
OpenMp+CUDA	8	768	2.6576223	2.1541966	0.1197141	0.3836156	1.0085281	0.1260660
OpenMp+CUDA	16	768	2.6643139	2.1611714	0.1199027	0.3831439	1.0059952	0.0628747



### 6.1.2 Type 1

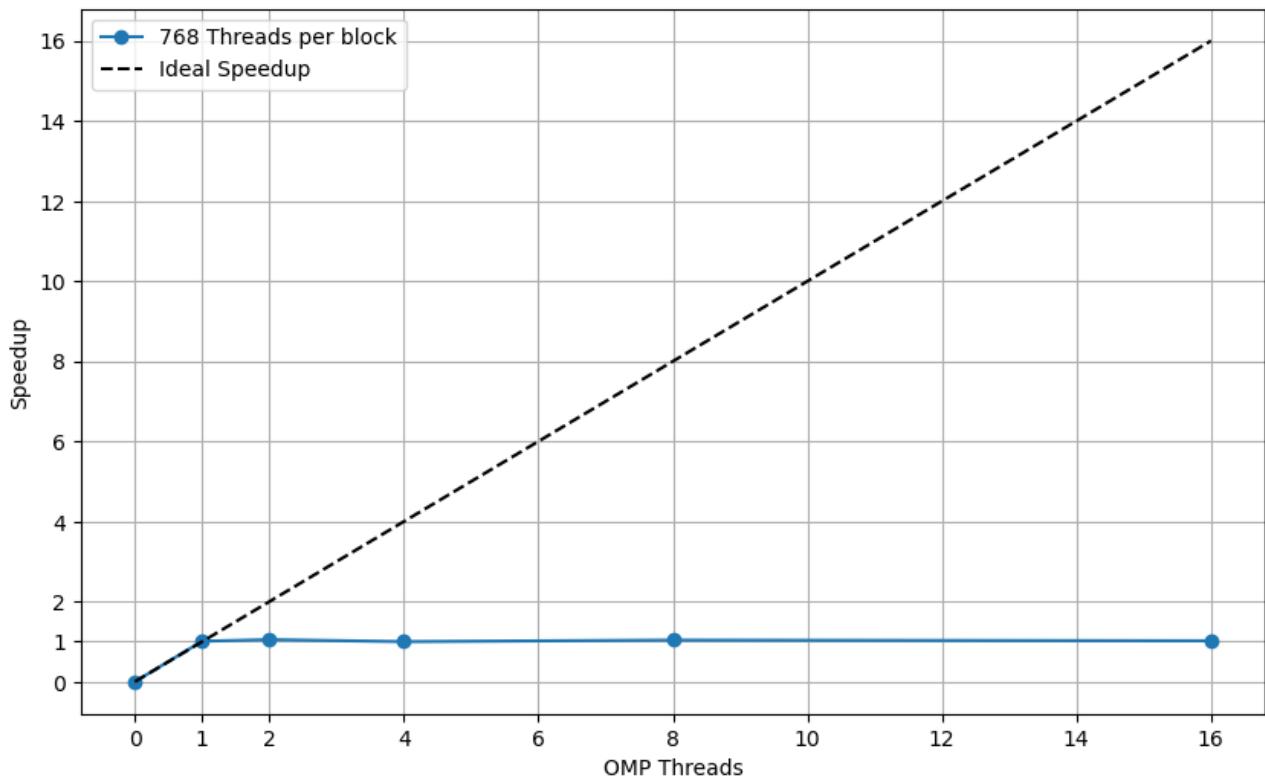
**1536 nodes & 1061043 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0240650	0.0146171	0.0000000	0.0094213	1.000000	1.000000
OpenMp+CUDA	1	768	0.0981935	0.0143491	0.0620963	0.0217364	0.2450772	0.2450772
OpenMp+CUDA	2	768	0.1015352	0.0138999	0.0627324	0.0248911	0.2370114	0.1185057
OpenMp+CUDA	4	768	0.0987267	0.0140464	0.0617765	0.0228917	0.2437538	0.0609384
OpenMp+CUDA	8	768	0.0989785	0.0138673	0.0620774	0.0230218	0.2431335	0.0303917
OpenMp+CUDA	16	768	0.1053126	0.0140793	0.0629345	0.0282869	0.2285102	0.0142819



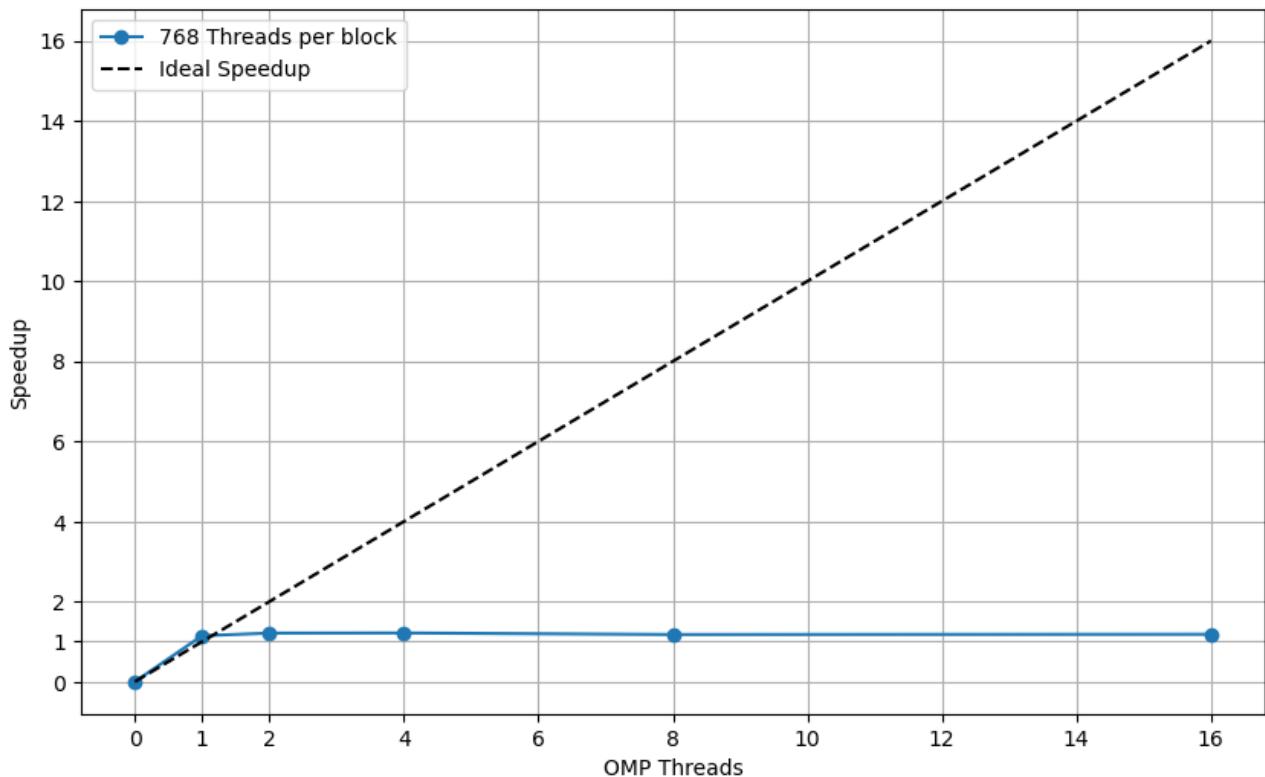
## 7552 nodes & 25660141 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5992059	0.3400475	0.0000000	0.2591287	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5932827	0.3394342	0.0782865	0.1755136	1.0099837	1.0099837
OpenMp+CUDA	2	768	0.5706074	0.3375843	0.0792406	0.1537353	1.0501193	0.5250597
OpenMp+CUDA	4	768	0.5995687	0.3376096	0.0783469	0.1835639	0.9993948	0.2498487
OpenMp+CUDA	8	768	0.5772583	0.3388723	0.0777305	0.1606078	1.0380203	0.1297525
OpenMp+CUDA	16	768	0.5870045	0.3393674	0.0781615	0.1694280	1.0207858	0.0637991



## 13568 nodes & 82829469 edges

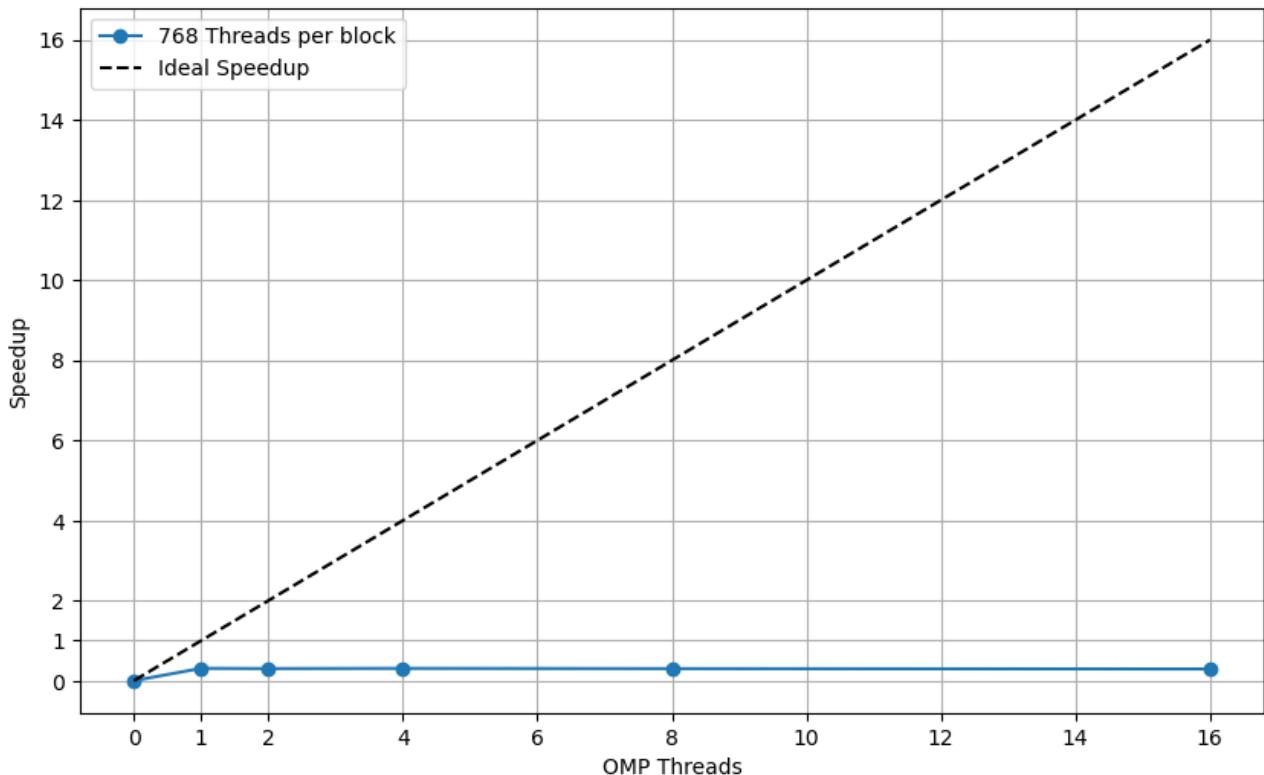
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.9658537	1.1683563	0.000000	0.7974657	1.000000	1.000000
OpenMp+CUDA	1	768	1.7122246	1.1559467	0.1184386	0.4377469	1.1481284	1.1481284
OpenMp+CUDA	2	768	1.6218102	1.1561951	0.1181037	0.3474193	1.2121355	0.6060677
OpenMp+CUDA	4	768	1.6166539	1.1622175	0.1178857	0.3364585	1.2160016	0.3040004
OpenMp+CUDA	8	768	1.6689222	1.1669787	0.1194997	0.3823511	1.1779181	0.1472398
OpenMp+CUDA	16	768	1.6608567	1.1613677	0.1175847	0.3818111	1.1836384	0.0739774



### 6.1.3 Type 2

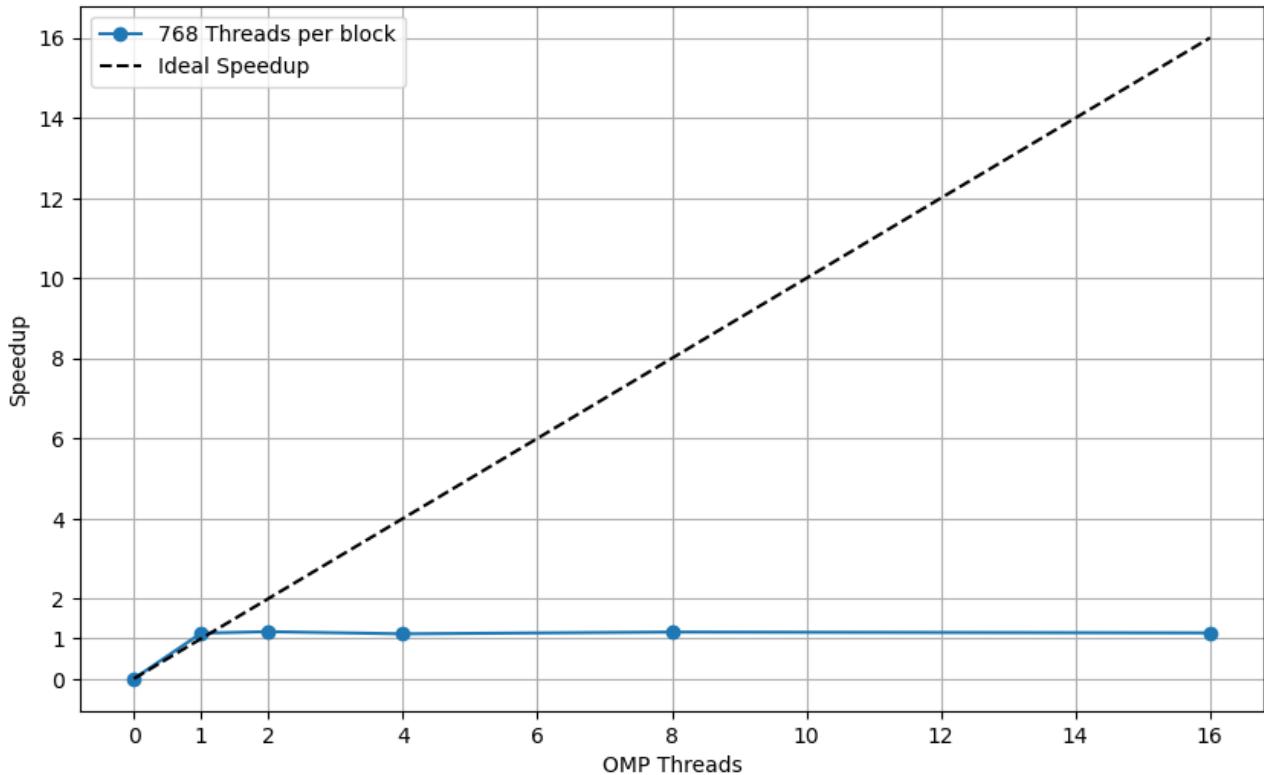
**1536 nodes & 790416 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0309549	0.0178296	0.0000000	0.0130994	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1004781	0.0173870	0.0616783	0.0214001	0.3080763	0.3080763
OpenMp+CUDA	2	768	0.1024774	0.0170453	0.0609266	0.0244937	0.3020660	0.1510330
OpenMp+CUDA	4	768	0.1007790	0.0170447	0.0612070	0.0225152	0.3071566	0.0767891
OpenMp+CUDA	8	768	0.1028896	0.0172238	0.0627852	0.0228687	0.3008558	0.0376070
OpenMp+CUDA	16	768	0.1060539	0.0171415	0.0612103	0.0276899	0.2918793	0.0182425



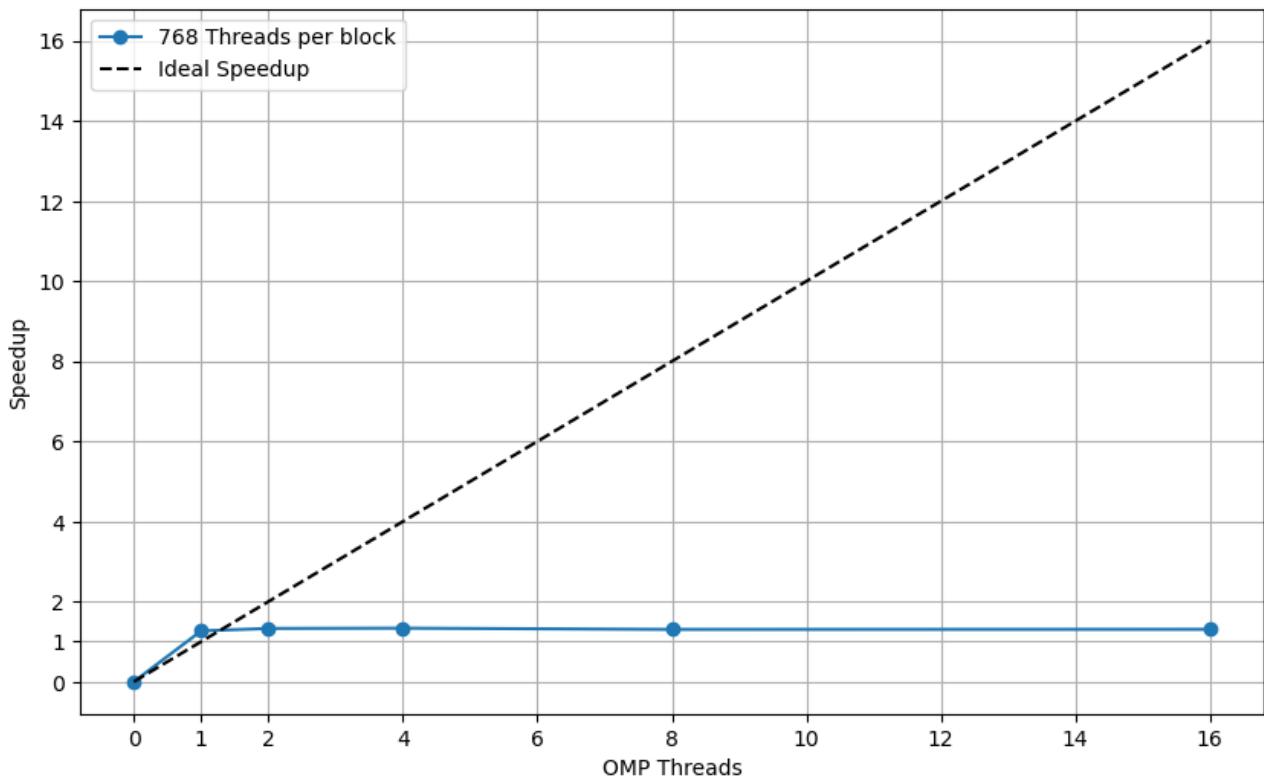
## 7552 nodes & 19101950 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.7584459	0.4191075	0.0000000	0.3393106	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6675794	0.4127067	0.0790129	0.1758118	1.1361133	1.1361133
OpenMp+CUDA	2	768	0.6457819	0.4129896	0.0789591	0.1537858	1.1744613	0.5872306
OpenMp+CUDA	4	768	0.6777464	0.4138011	0.0793182	0.1845794	1.1190703	0.2797676
OpenMp+CUDA	8	768	0.6511601	0.4126044	0.0787555	0.1597522	1.1647609	0.1455951
OpenMp+CUDA	16	768	0.6636531	0.4133945	0.0800697	0.1701413	1.1428348	0.0714272



## 13568 nodes & 61665181 edges

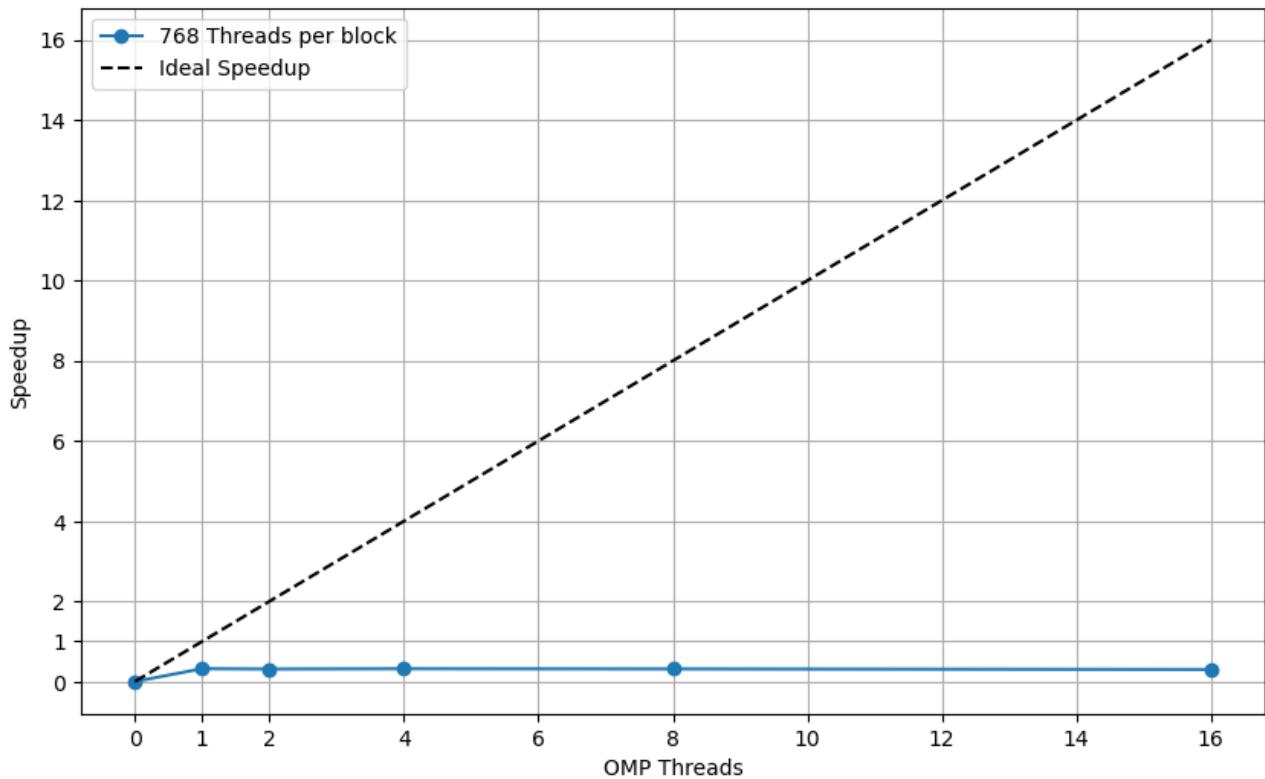
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.4746373	1.3941055	0.0000000	1.0805001	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.9523476	1.3941343	0.1201073	0.4380130	1.2675188	1.2675188
OpenMp+CUDA	2	768	1.8615969	1.3922720	0.1192021	0.3500303	1.3293089	0.6646544
OpenMp+CUDA	4	768	1.8517636	1.3924296	0.1186716	0.3405694	1.3363678	0.3340920
OpenMp+CUDA	8	768	1.8965609	1.3935471	0.1191335	0.3837877	1.3048025	0.1631003
OpenMp+CUDA	16	768	1.8932793	1.3912041	0.1188965	0.3830866	1.3070641	0.0816915



#### 6.1.4 Type 3

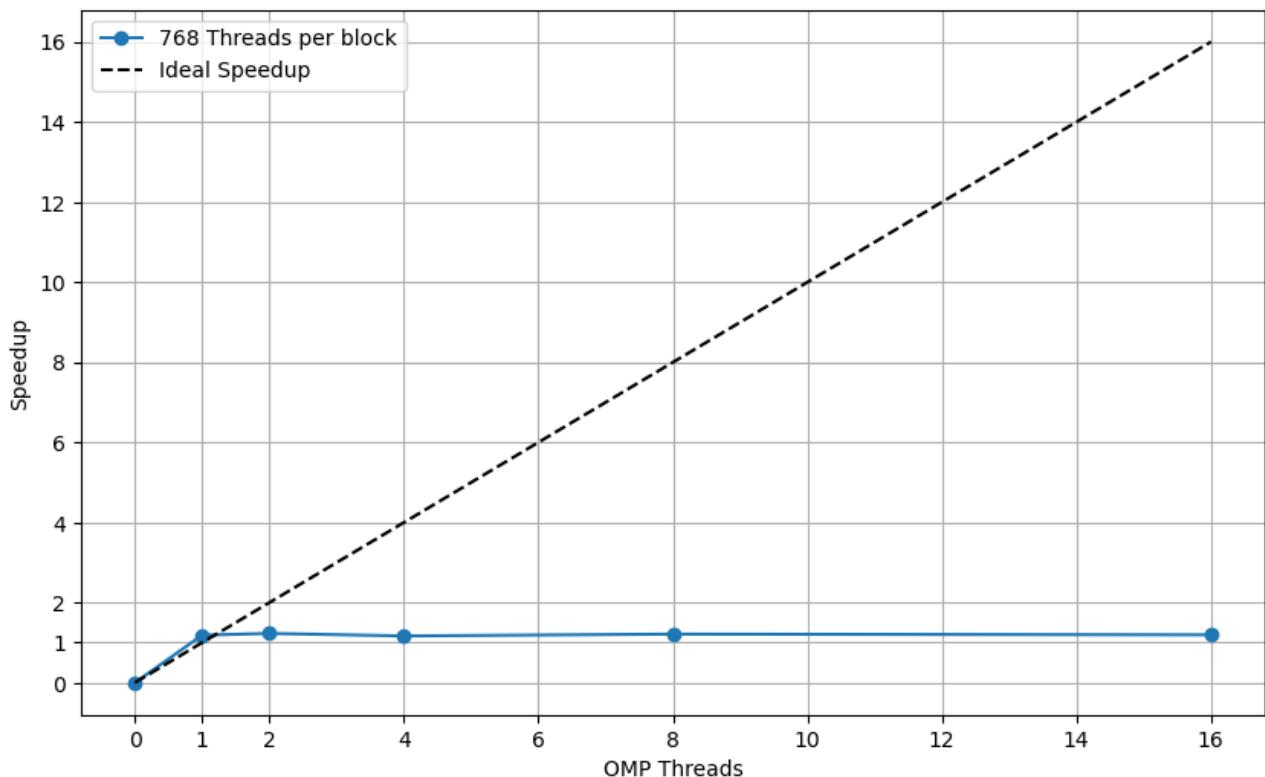
**1536 nodes & 588142 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0321457	0.0173455	0.0000000	0.0147734	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1004171	0.0170714	0.0621174	0.0212158	0.3201213	0.3201213
OpenMp+CUDA	2	768	0.1029341	0.0168640	0.0618161	0.0242424	0.3122936	0.1561468
OpenMp+CUDA	4	768	0.1007777	0.0169647	0.0611345	0.0226667	0.3189759	0.0797440
OpenMp+CUDA	8	768	0.1021899	0.0170186	0.0623574	0.0228024	0.3145680	0.0393210
OpenMp+CUDA	16	768	0.1066297	0.0176401	0.0613495	0.0276281	0.3014700	0.0188419



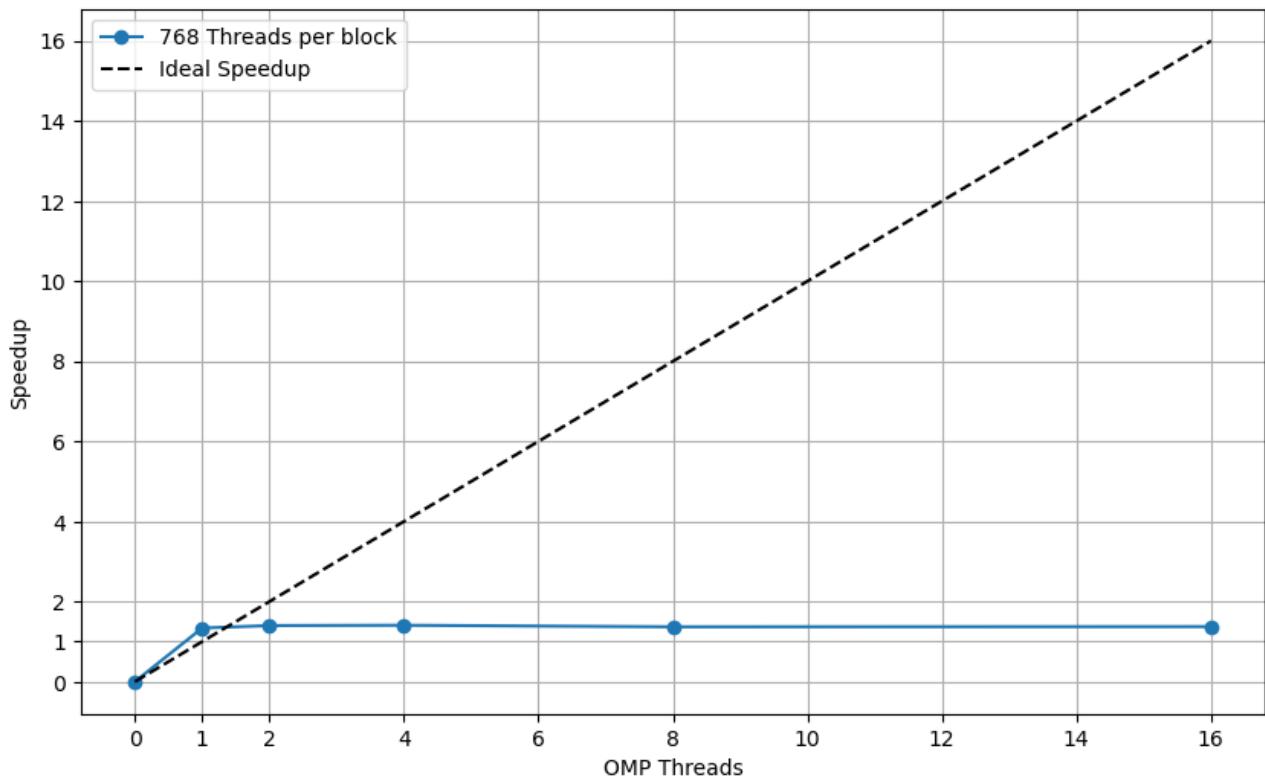
## 7552 nodes & 14260151 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.7811967	0.4102399	0.0000000	0.3709271	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6587103	0.4031187	0.0794844	0.1760597	1.1859487	1.1859487
OpenMp+CUDA	2	768	0.6351896	0.4035008	0.0785451	0.1530957	1.2298638	0.6149319
OpenMp+CUDA	4	768	0.6692264	0.4049513	0.0798269	0.1844000	1.1673130	0.2918282
OpenMp+CUDA	8	768	0.6442334	0.4046091	0.0794357	0.1601407	1.2125988	0.1515749
OpenMp+CUDA	16	768	0.6539701	0.4052505	0.0785249	0.1701469	1.1945450	0.0746591



## 13568 nodes & 46014096 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.5986958	1.3947622	0.0000000	1.2039047	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.9413642	1.3811015	0.1207717	0.4393986	1.3385926	1.3385926
OpenMp+CUDA	2	768	1.8548808	1.3875335	0.1197925	0.3474623	1.4010042	0.7005021
OpenMp+CUDA	4	768	1.8446048	1.3872180	0.1191130	0.3381814	1.4088090	0.3522022
OpenMp+CUDA	8	768	1.8980848	1.3926489	0.1201091	0.3852335	1.3691147	0.1711393
OpenMp+CUDA	16	768	1.8912617	1.3876835	0.1187941	0.3846911	1.3740541	0.0858784



## 6.2 Optimization 1

Optimization 1 does not introduce an improvement in speedup and performance compared to O0.

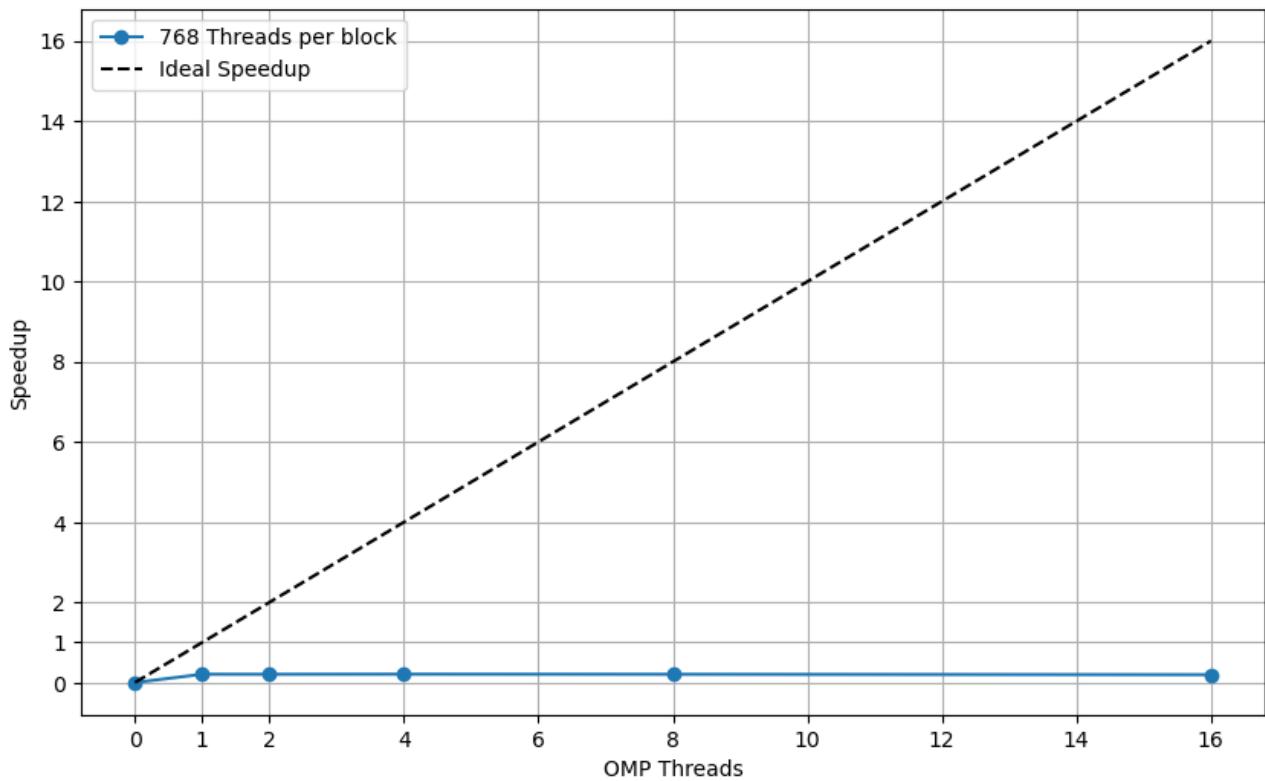
The algorithm remains efficient for medium and mainly for large, type 3, graphs.

Optimization 0 remains the one with the best performance

### 6.2.1 Type 0

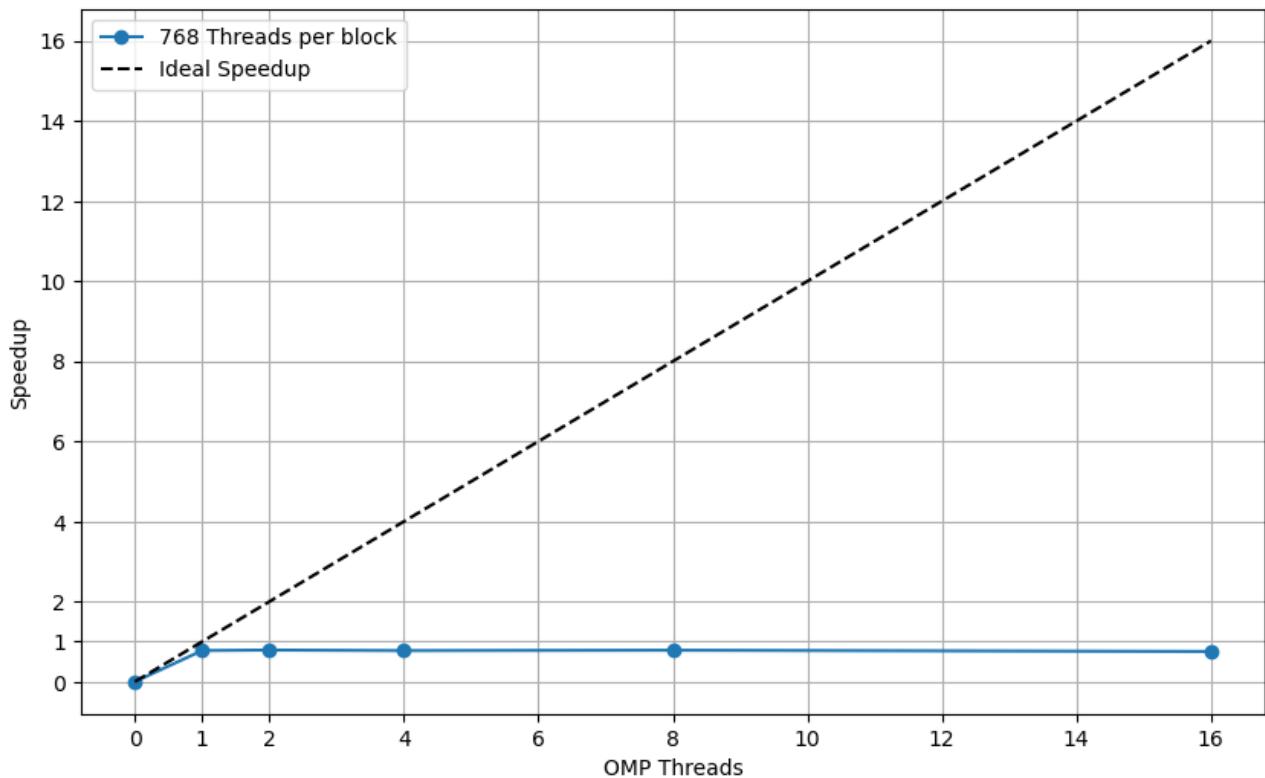
**1536 nodes & 2357760 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0212619	0.0185059	0.0000000	0.0027288	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1030644	0.0180643	0.0648803	0.0201093	0.2062969	0.2062969
OpenMp+CUDA	2	768	0.1034505	0.0174747	0.0643419	0.0216234	0.2055269	0.1027634
OpenMp+CUDA	4	768	0.1021019	0.0174219	0.0635751	0.0210946	0.2082417	0.0520604
OpenMp+CUDA	8	768	0.1035296	0.0175228	0.0638449	0.0221515	0.2053699	0.0256712
OpenMp+CUDA	16	768	0.1090171	0.0175291	0.0637643	0.0277137	0.1950325	0.0121895



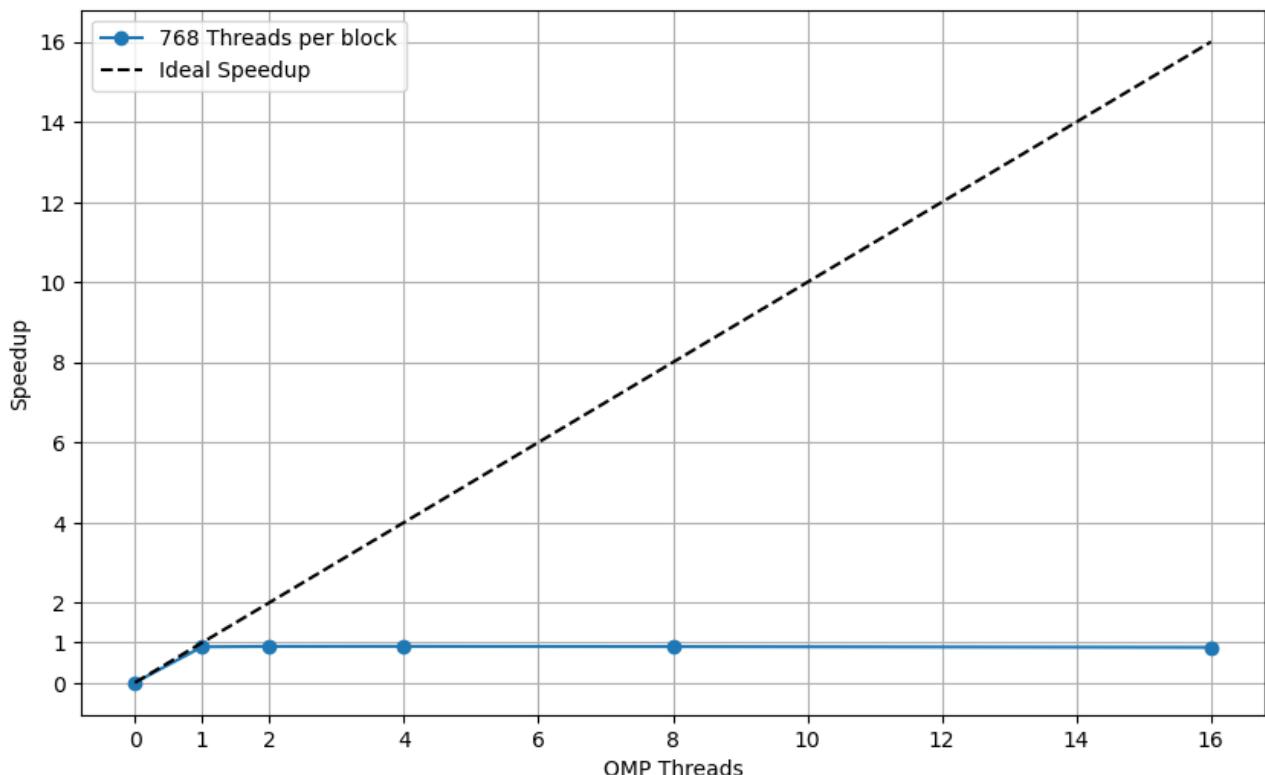
## 7552 nodes & 57025152 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5212525	0.4519587	0.0000000	0.0692682	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6707706	0.4477107	0.0802364	0.1427851	0.7770951	0.7770951
OpenMp+CUDA	2	768	0.6601683	0.4453651	0.08005951	0.1341695	0.7895753	0.3947876
OpenMp+CUDA	4	768	0.6720490	0.4469877	0.0802583	0.1447647	0.7756169	0.1939042
OpenMp+CUDA	8	768	0.6637359	0.4452546	0.0807387	0.1377039	0.7853313	0.0981664
OpenMp+CUDA	16	768	0.6911494	0.4449490	0.0803018	0.1658607	0.7541821	0.0471364



## 13568 nodes & 184077056 edges

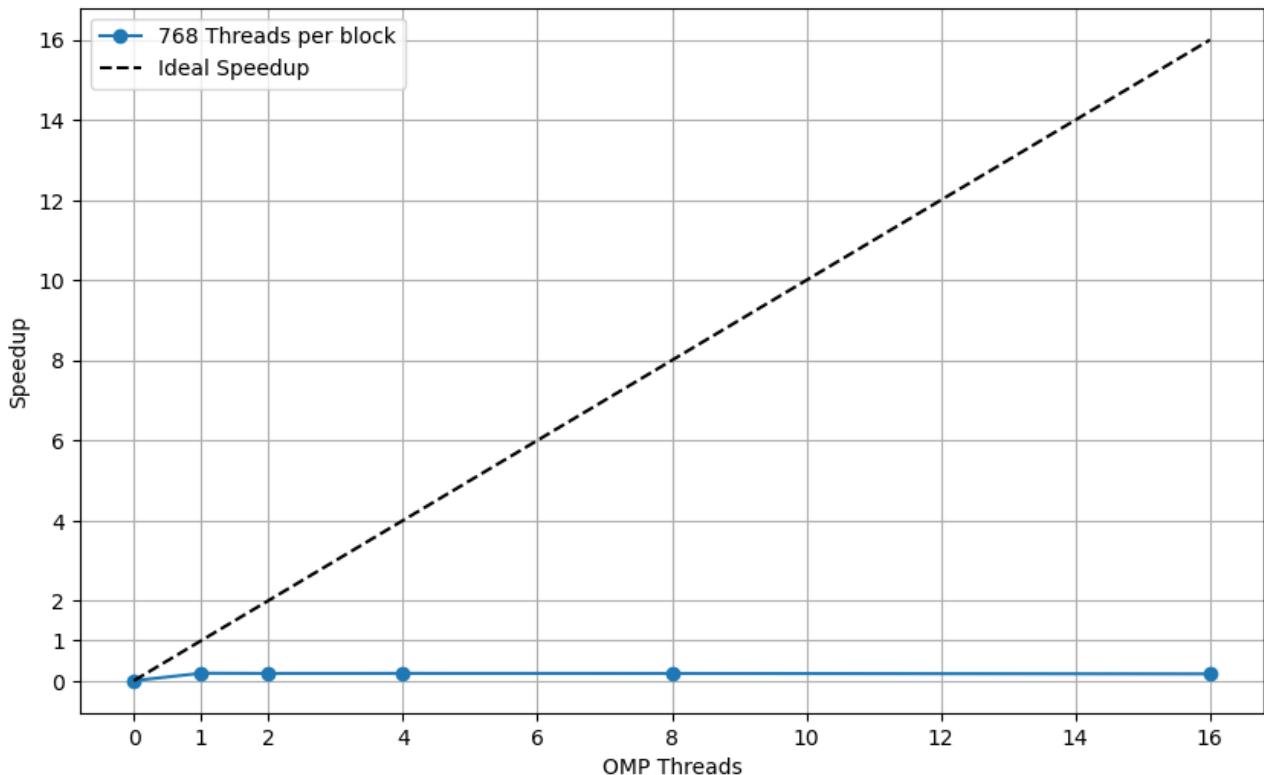
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.2390395	2.0510851	0.0000000	0.1879222	1.000000	1.000000
OpenMp+CUDA	1	768	2.4991040	2.0482643	0.1211539	0.3296144	0.8959369	0.8959369
OpenMp+CUDA	2	768	2.4736469	2.0578773	0.1211201	0.2945784	0.9051573	0.4525786
OpenMp+CUDA	4	768	2.4701732	2.0503476	0.1204468	0.2993061	0.9064302	0.2266075
OpenMp+CUDA	8	768	2.4846668	2.0505939	0.1195491	0.3144510	0.9011428	0.1126428
OpenMp+CUDA	16	768	2.5371553	2.0542205	0.1212205	0.3616436	0.8825000	0.0551562



### 6.2.2 Type 1

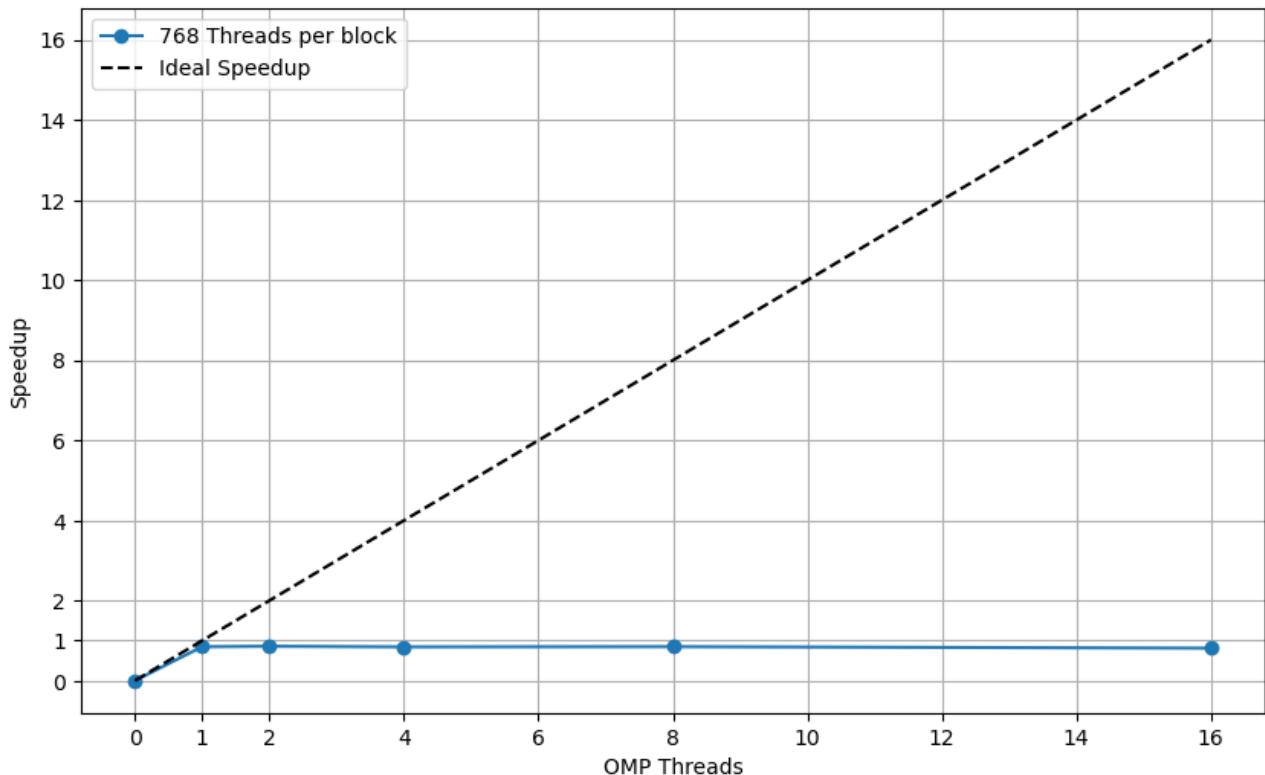
**1536 nodes & 1061043 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0179312	0.0133075	0.0000000	0.0046002	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0965583	0.0131487	0.0630081	0.0203925	0.1857033	0.1857033
OpenMp+CUDA	2	768	0.0989335	0.0130965	0.0639333	0.0218946	0.1812449	0.0906225
OpenMp+CUDA	4	768	0.0982846	0.0130172	0.0638700	0.0213886	0.1824416	0.0456104
OpenMp+CUDA	8	768	0.0992461	0.0130703	0.0633931	0.0227736	0.1806742	0.0225843
OpenMp+CUDA	16	768	0.1056235	0.0130849	0.0642484	0.0282813	0.1697653	0.0106103



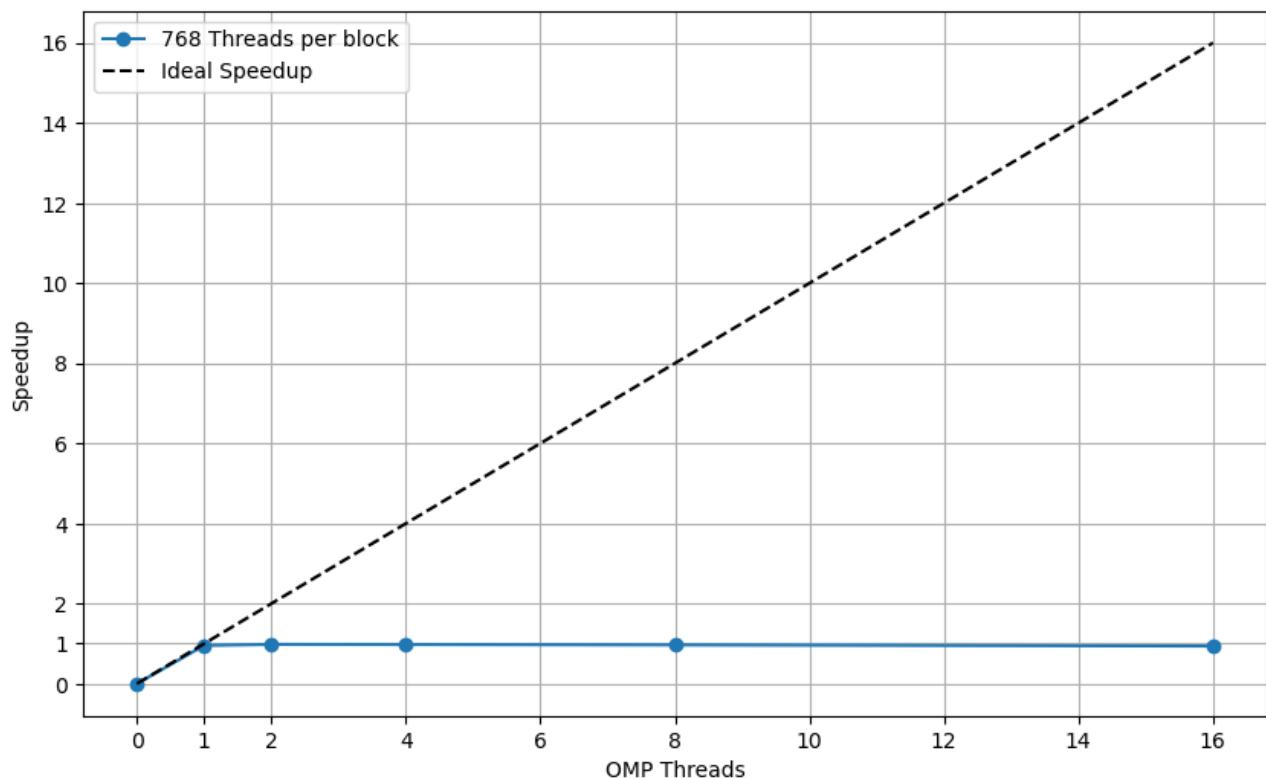
## 7552 nodes & 25660141 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4508875	0.3151238	0.0000000	0.1357376	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5319073	0.3088332	0.0801064	0.1429327	0.8476806	0.8476806
OpenMp+CUDA	2	768	0.5243976	0.3101830	0.0807406	0.1334396	0.8598200	0.4299100
OpenMp+CUDA	4	768	0.5342757	0.3108083	0.0798014	0.1436311	0.8439229	0.2109807
OpenMp+CUDA	8	768	0.5306971	0.3125701	0.0807707	0.1373218	0.8496137	0.1062017
OpenMp+CUDA	16	768	0.5552914	0.3112717	0.0797079	0.1642772	0.8119836	0.0507490



## 13568 nodes & 82829469 edges

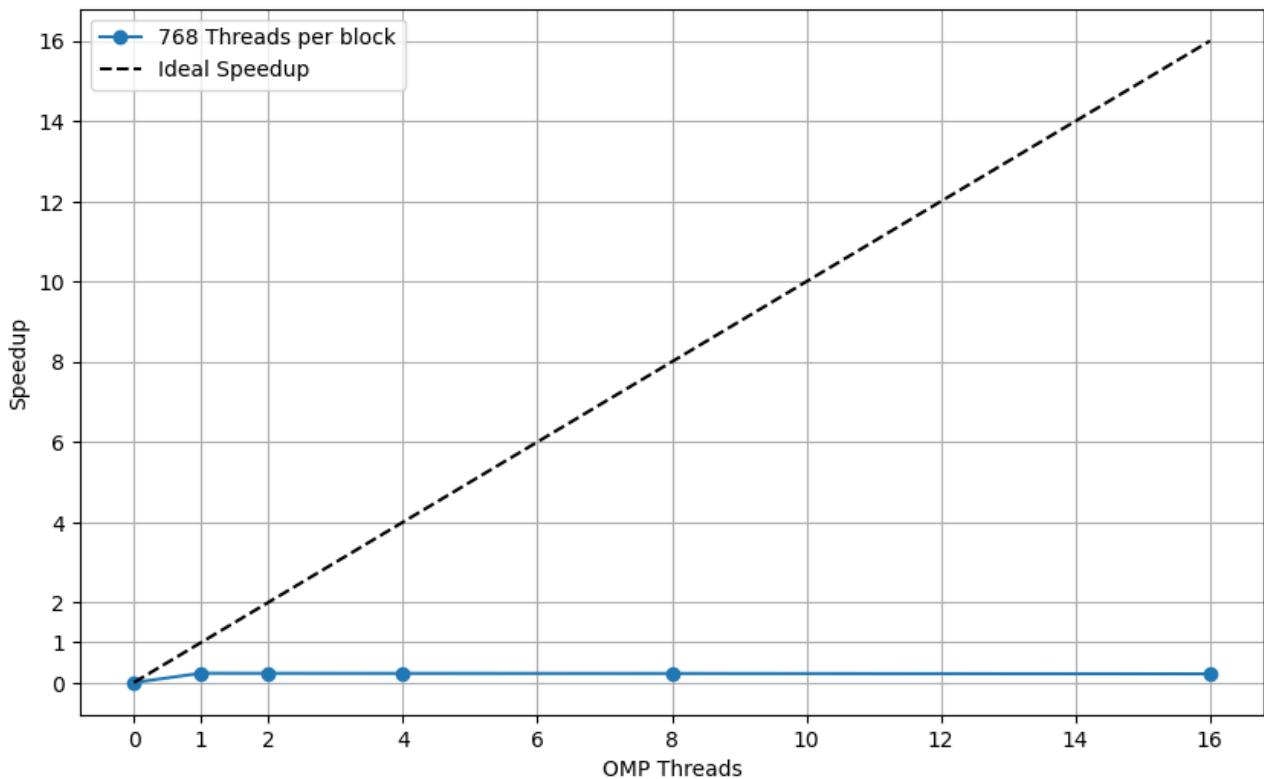
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4525081	1.0739543	0.0000000	0.3785215	1.000000	1.000000
OpenMp+CUDA	1	768	1.5217864	1.0728157	0.1194709	0.3294305	0.9544757	0.9544757
OpenMp+CUDA	2	768	1.4808226	1.0715034	0.1194458	0.2898041	0.9808792	0.4904396
OpenMp+CUDA	4	768	1.4848125	1.0747616	0.1190898	0.2908919	0.9782434	0.2445609
OpenMp+CUDA	8	768	1.5003785	1.0674828	0.1190681	0.3137581	0.9680944	0.1210118
OpenMp+CUDA	16	768	1.5462677	1.0677485	0.1197833	0.3586662	0.9393639	0.0587102



### 6.2.3 Type 2

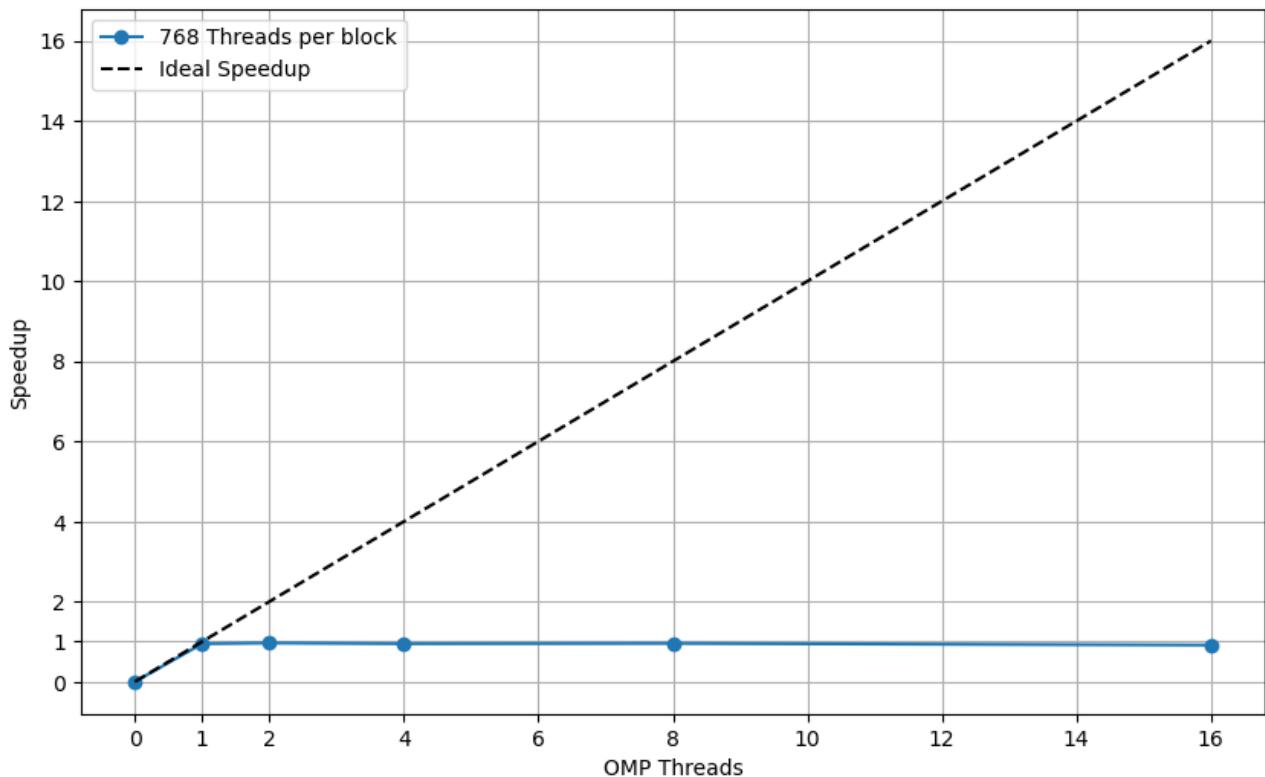
**1536 nodes & 790416 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0230232	0.0159009	0.0000000	0.0070965	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0992443	0.0156813	0.0632316	0.0203223	0.2319852	0.2319852
OpenMp+CUDA	2	768	0.1000597	0.0153849	0.0631143	0.0215512	0.2300947	0.1150474
OpenMp+CUDA	4	768	0.1006750	0.0155643	0.0637957	0.0213061	0.2286884	0.0571721
OpenMp+CUDA	8	768	0.1016289	0.0155509	0.0637594	0.0223094	0.2265418	0.0283177
OpenMp+CUDA	16	768	0.1060075	0.0155797	0.0628186	0.0276001	0.2171846	0.0135740



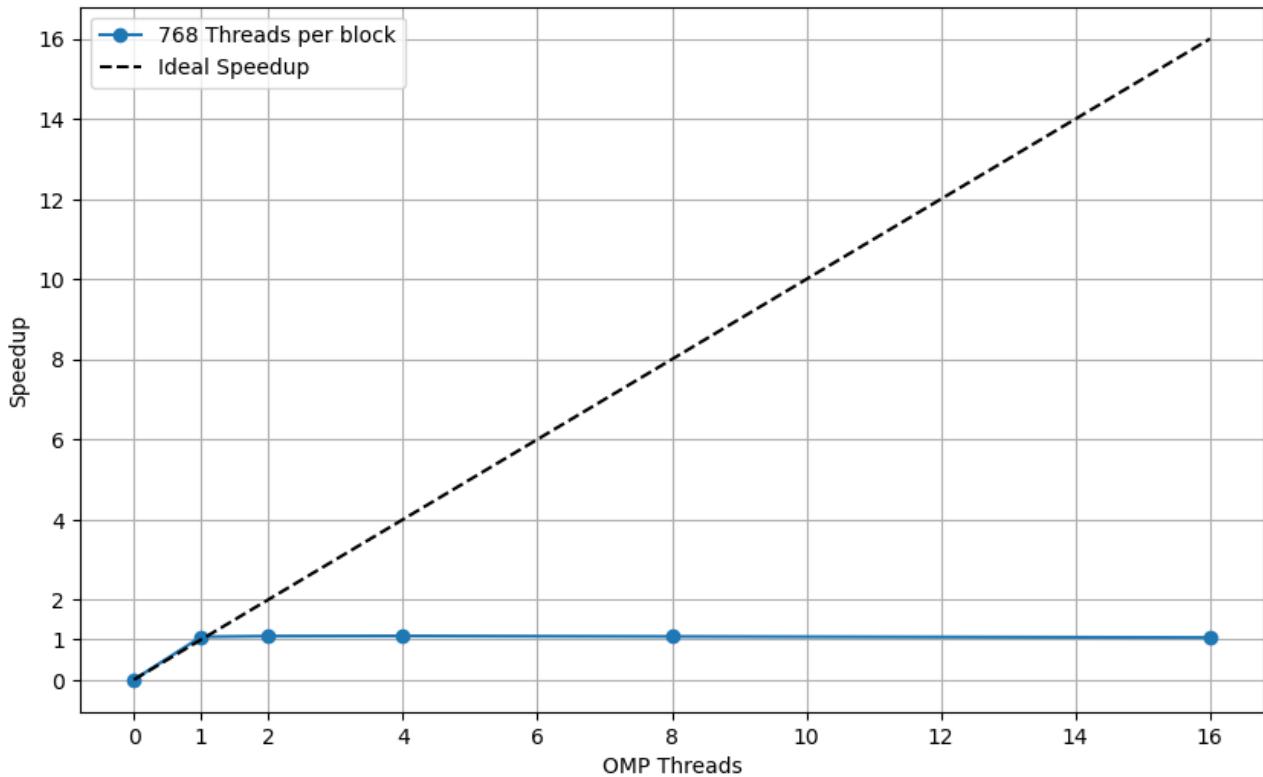
## 7552 nodes & 19101950 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5651073	0.3732839	0.0000000	0.1917963	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5959541	0.3723273	0.0808789	0.1427137	0.9482395	0.9482395
OpenMp+CUDA	2	768	0.5851469	0.3703904	0.0803063	0.1344162	0.9657529	0.4828764
OpenMp+CUDA	4	768	0.5956496	0.3703217	0.0808421	0.1444517	0.9487243	0.2371811
OpenMp+CUDA	8	768	0.5909761	0.3713429	0.0810713	0.1385275	0.9562269	0.1195284
OpenMp+CUDA	16	768	0.6184706	0.3725131	0.0807366	0.1651865	0.9137173	0.0571073



## 13568 nodes & 61665181 edges

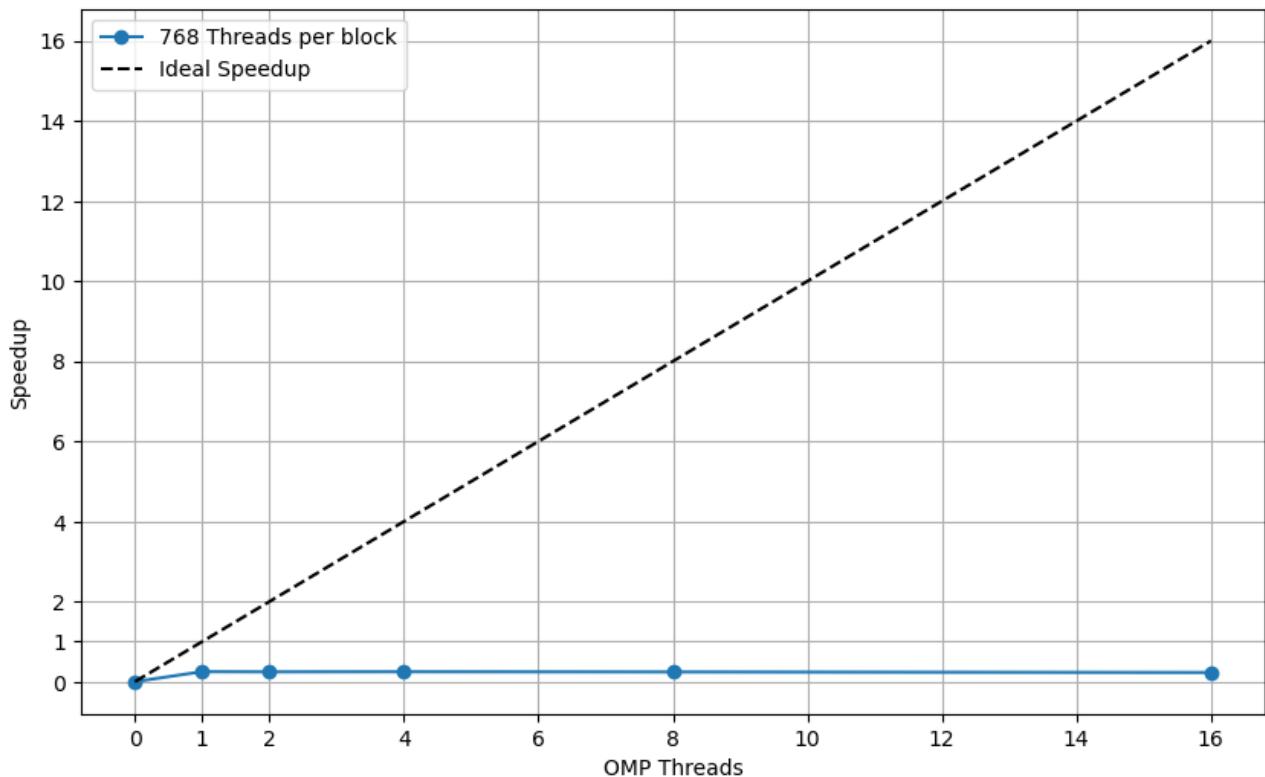
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8328761	1.2704297	0.0000000	0.5624140	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.7131308	1.2638185	0.1200916	0.3291511	1.0698985	1.0698985
OpenMp+CUDA	2	768	1.6851627	1.2759287	0.1196403	0.2895247	1.0876553	0.5438277
OpenMp+CUDA	4	768	1.6784559	1.2677341	0.1191043	0.2915471	1.0920014	0.2730003
OpenMp+CUDA	8	768	1.7000702	1.2696029	0.1186747	0.3117232	1.0781179	0.1347647
OpenMp+CUDA	16	768	1.7429868	1.2657393	0.1201754	0.3570027	1.0515720	0.0657233



#### 6.2.4 Type 3

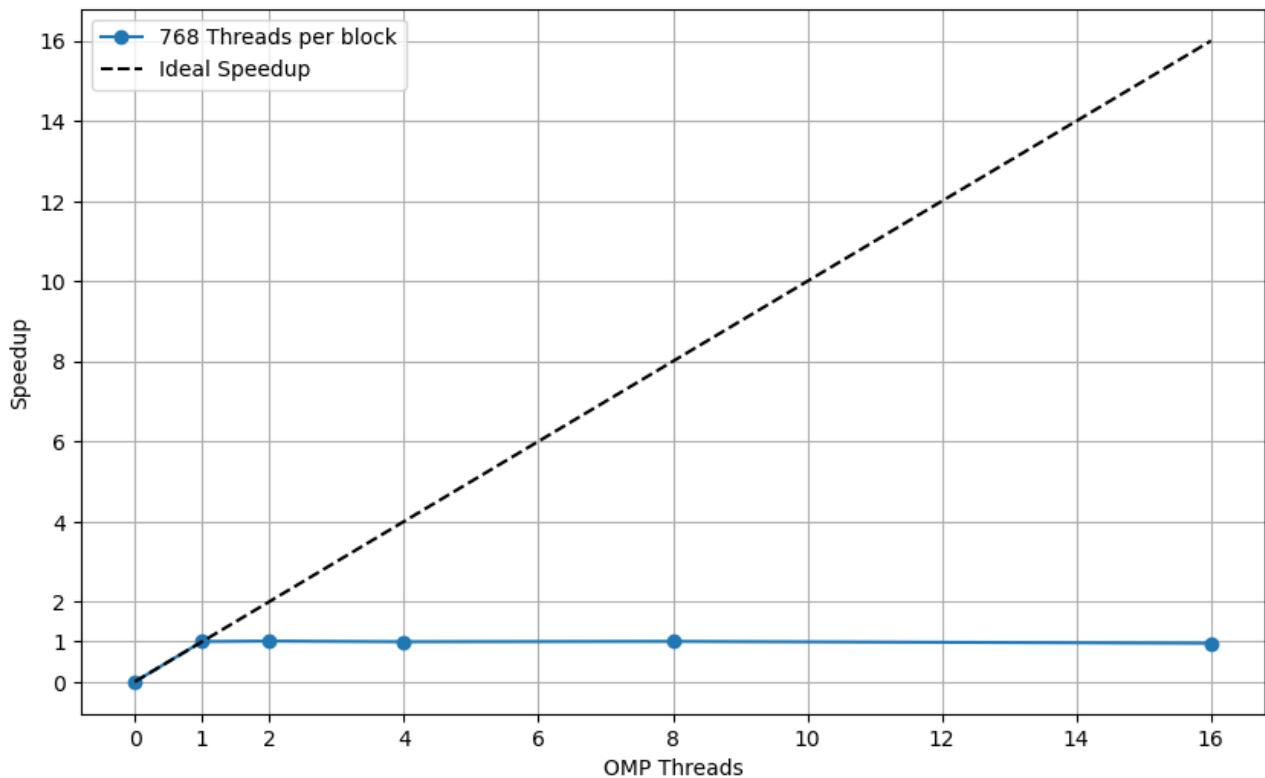
**1536 nodes & 588142 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0245517	0.0159185	0.0000000	0.0086071	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0982187	0.0157160	0.0625833	0.0199102	0.2499700	0.2499700
OpenMp+CUDA	2	768	0.1000256	0.0154599	0.0631245	0.0214318	0.2454545	0.1227272
OpenMp+CUDA	4	768	0.0989812	0.0155106	0.0622847	0.0211765	0.2480444	0.0620111
OpenMp+CUDA	8	768	0.1009600	0.0154836	0.0630851	0.0223824	0.2431828	0.0303978
OpenMp+CUDA	16	768	0.1072736	0.0155178	0.0630613	0.0286853	0.2288702	0.0143044



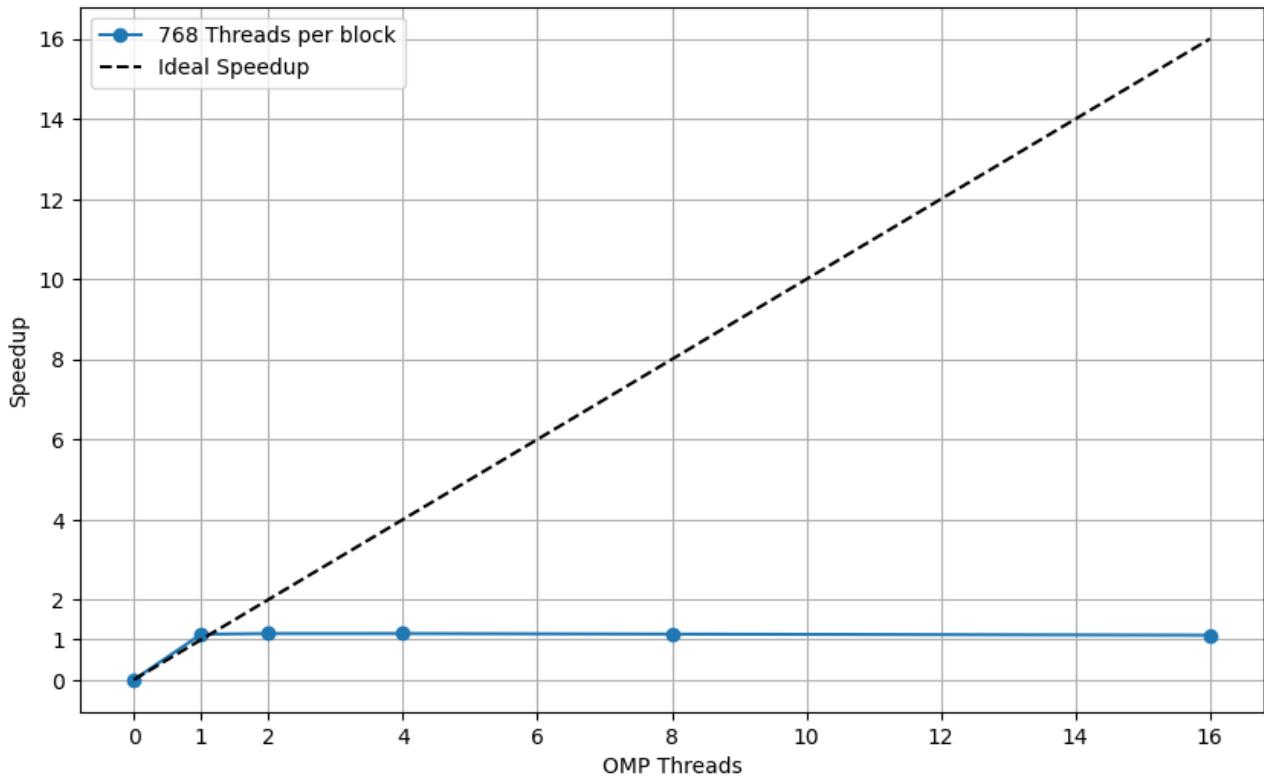
## 7552 nodes & 14260151 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5931903	0.3752528	0.0000000	0.2179093	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5921678	0.3691010	0.0800795	0.1429529	1.0017268	1.0017268
OpenMp+CUDA	2	768	0.5854253	0.3696523	0.0812192	0.1345193	1.0132639	0.5066319
OpenMp+CUDA	4	768	0.5947033	0.3701849	0.0804987	0.1439847	0.9974560	0.2493640
OpenMp+CUDA	8	768	0.5894594	0.3700189	0.0809149	0.1384907	1.0063294	0.1257912
OpenMp+CUDA	16	768	0.6183739	0.3714111	0.0811575	0.1657705	0.9592745	0.0599547



## 13568 nodes & 46014096 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.9347739	1.2718554	0.0000000	0.6628874	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.7063631	1.2568643	0.1197445	0.3296853	1.1338583	1.1338583
OpenMp+CUDA	2	768	1.6794451	1.2643269	0.1195355	0.2955131	1.1520316	0.5760158
OpenMp+CUDA	4	768	1.6785691	1.2637989	0.1188908	0.2958088	1.1526329	0.2881582
OpenMp+CUDA	8	768	1.7002508	1.2680110	0.1202586	0.3119120	1.1379344	0.1422418
OpenMp+CUDA	16	768	1.7407390	1.2625257	0.1197178	0.3584259	1.1114670	0.0694667



## 6.3 Optimization 2

Optimization 2 does not introduce an improvement in speedup and performance compared to O0, but has better results than O1.

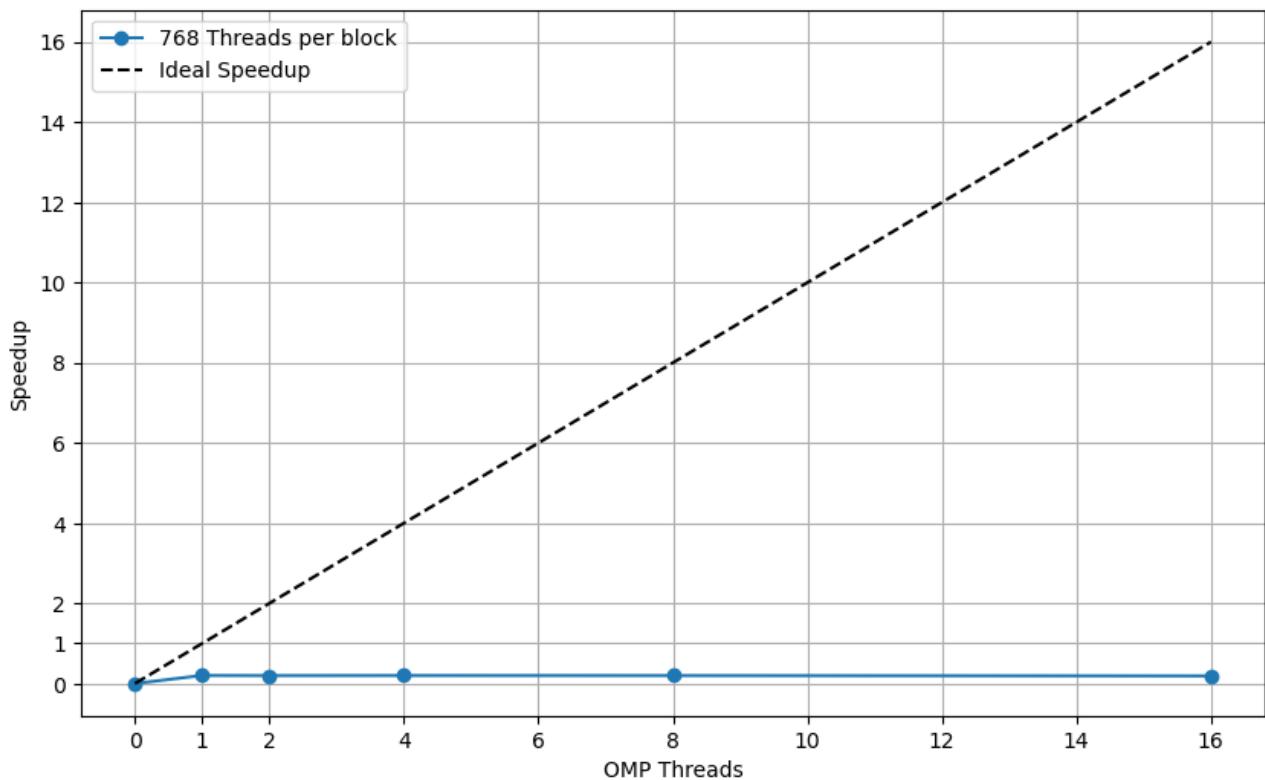
The algorithm remains efficient for medium and mainly for large, type 3, graphs.

Optimization 0 remains the one with the best performance.

### 6.3.1 Type 0

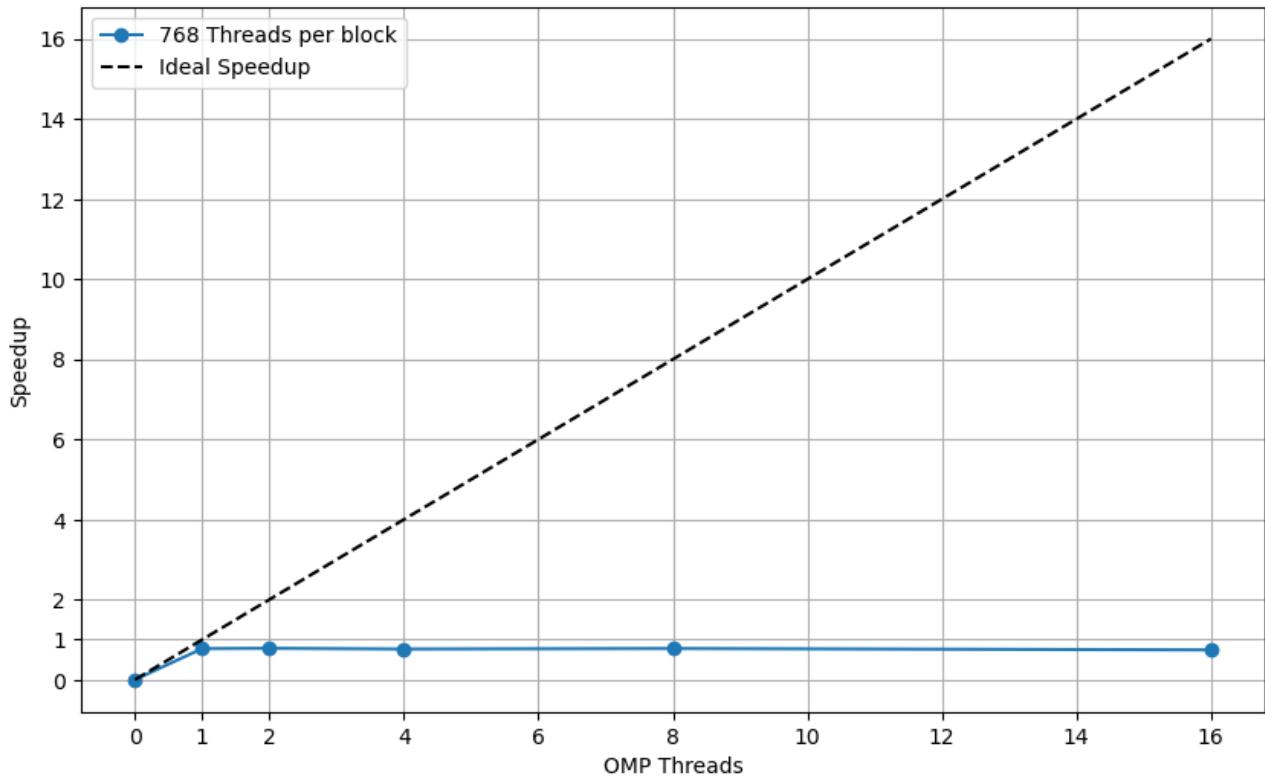
**1536 nodes & 2357760 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0208825	0.0178341	0.0000000	0.0030226	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1035237	0.0177886	0.0655218	0.0202036	0.2017168	0.2017168
OpenMp+CUDA	2	768	0.1050664	0.0186363	0.0647108	0.0217093	0.1987549	0.0993775
OpenMp+CUDA	4	768	0.1041070	0.0173665	0.0654192	0.0213110	0.2005866	0.0501466
OpenMp+CUDA	8	768	0.1048281	0.0174998	0.0649239	0.0223950	0.1992067	0.0249008
OpenMp+CUDA	16	768	0.1102075	0.0174113	0.0651963	0.0275875	0.1894832	0.0118427



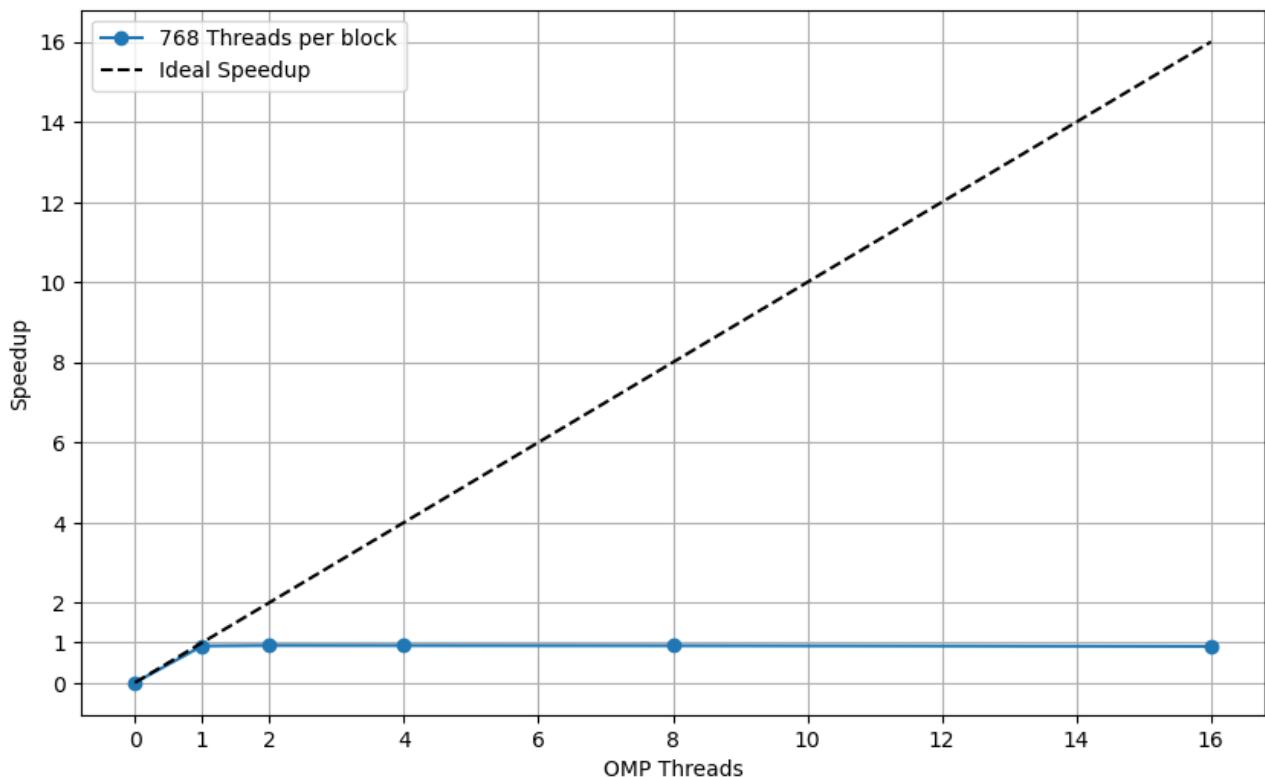
## 7552 nodes & 57025152 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5202267	0.4483063	0.0000000	0.0718910	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6709610	0.4466733	0.0812976	0.1429536	0.7753457	0.7753457
OpenMp+CUDA	2	768	0.6640319	0.4475241	0.0806393	0.1358322	0.7834363	0.3917182
OpenMp+CUDA	4	768	0.6794955	0.4544323	0.0806926	0.1443337	0.7656073	0.1914018
OpenMp+CUDA	8	768	0.6679727	0.4482963	0.0812642	0.1383763	0.7788143	0.0973518
OpenMp+CUDA	16	768	0.6994033	0.4535770	0.0804544	0.1653357	0.7438151	0.0464884



## 13568 nodes & 184077056 edges

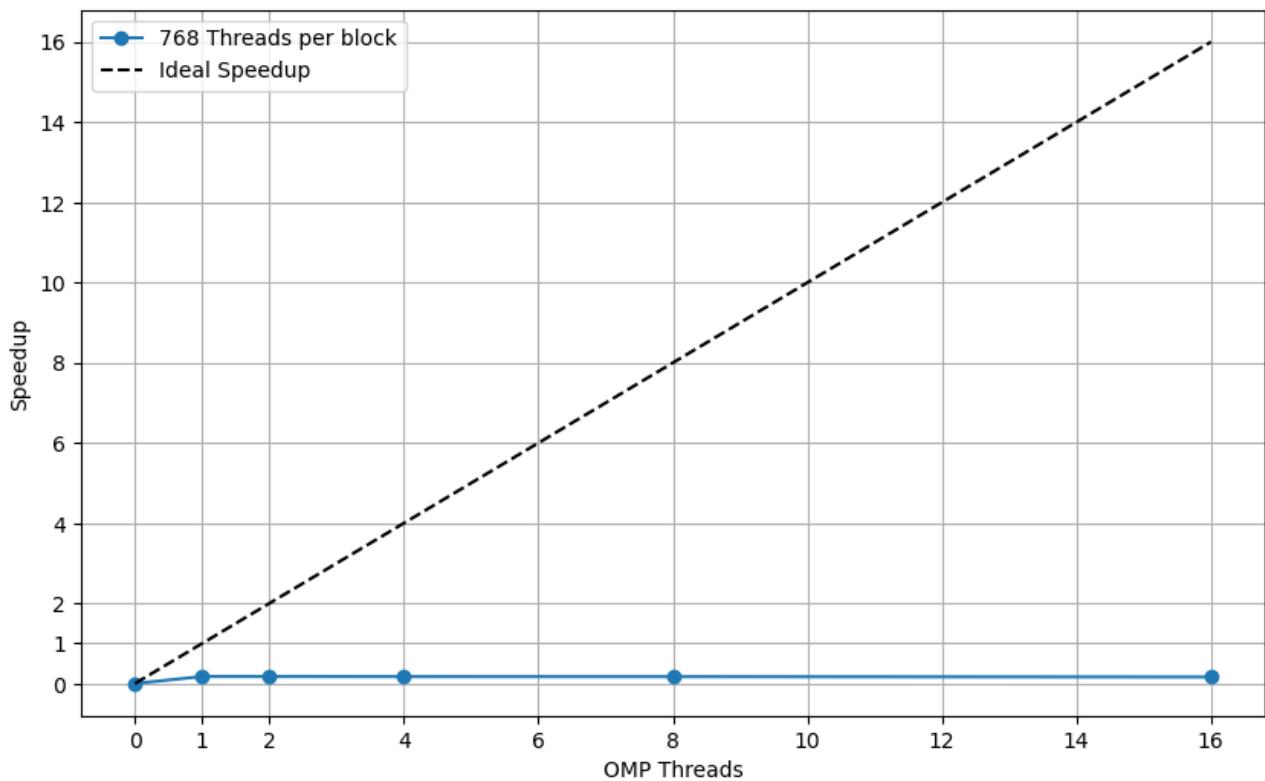
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.2993167	2.0608135	0.0000000	0.2384716	1.0000000	1.0000000
OpenMp+CUDA	1	768	2.5066091	2.0544363	0.1216024	0.3305010	0.9173017	0.9173017
OpenMp+CUDA	2	768	2.4733699	2.0575395	0.1205486	0.2952125	0.9296292	0.4648146
OpenMp+CUDA	4	768	2.4784081	2.0577373	0.1216532	0.2989481	0.9277393	0.2319348
OpenMp+CUDA	8	768	2.4882715	2.0526431	0.1214281	0.3141308	0.9240618	0.1155077
OpenMp+CUDA	16	768	2.5349760	2.0512135	0.1205495	0.3631429	0.9070369	0.0566898



### 6.3.2 Type 1

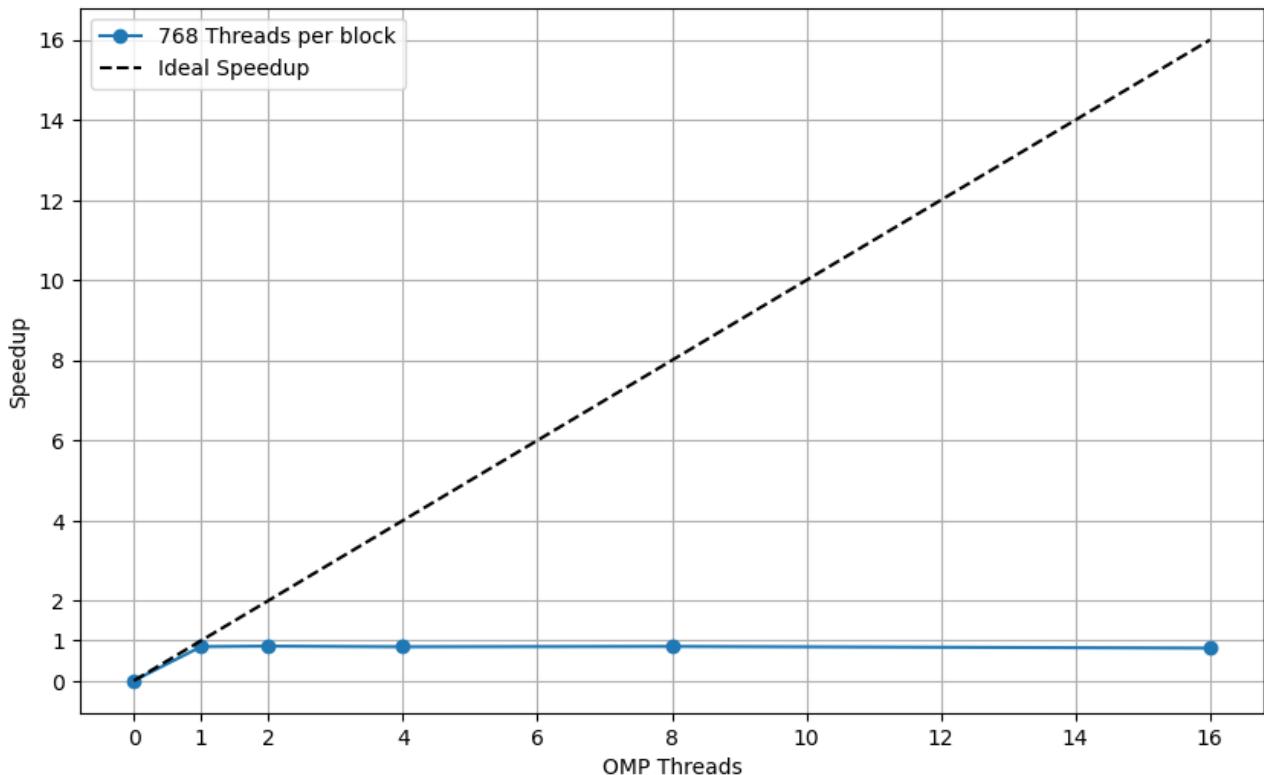
**1536 nodes & 1061043 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0171206	0.0122602	0.0000000	0.0048345	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0975553	0.0123448	0.0648167	0.0203841	0.1754963	0.1754963
OpenMp+CUDA	2	768	0.0978702	0.0122834	0.0639035	0.0216725	0.1749317	0.0874658
OpenMp+CUDA	4	768	0.0989689	0.0125984	0.0648593	0.0215026	0.1729896	0.0432474
OpenMp+CUDA	8	768	0.0997235	0.0121715	0.0649331	0.0226097	0.1716806	0.0214601
OpenMp+CUDA	16	768	0.1048046	0.0121559	0.0642606	0.0283787	0.1633573	0.0102098



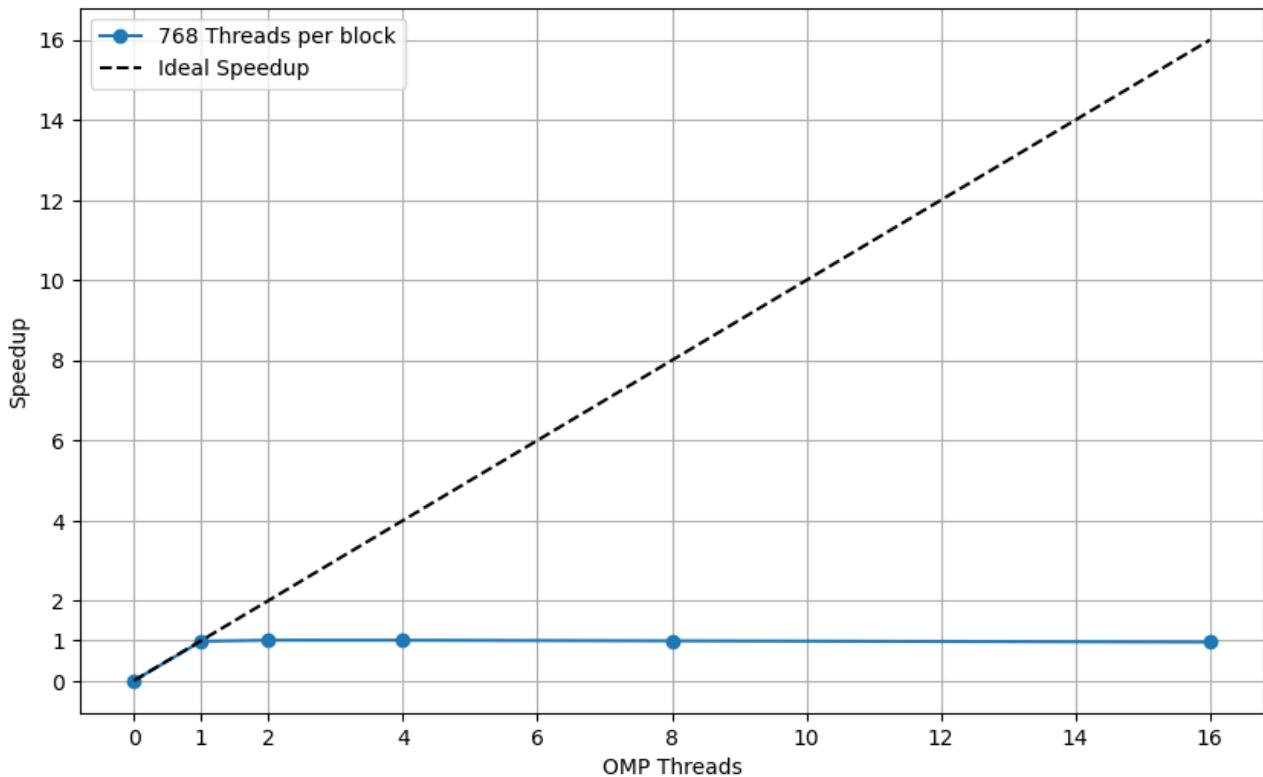
## 7552 nodes & 25660141 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4373078	0.2939786	0.0000000	0.1433019	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5144525	0.2904867	0.0804141	0.1435185	0.8500450	0.8500450
OpenMp+CUDA	2	768	0.5072853	0.2918349	0.0819717	0.1334450	0.8620550	0.4310275
OpenMp+CUDA	4	768	0.5164164	0.2916343	0.0799853	0.1447631	0.8468124	0.2117031
OpenMp+CUDA	8	768	0.5114032	0.2911103	0.0813105	0.1389494	0.8551135	0.1068892
OpenMp+CUDA	16	768	0.5377293	0.2919551	0.0805492	0.1651919	0.8132489	0.0508281



## 13568 nodes & 82829469 edges

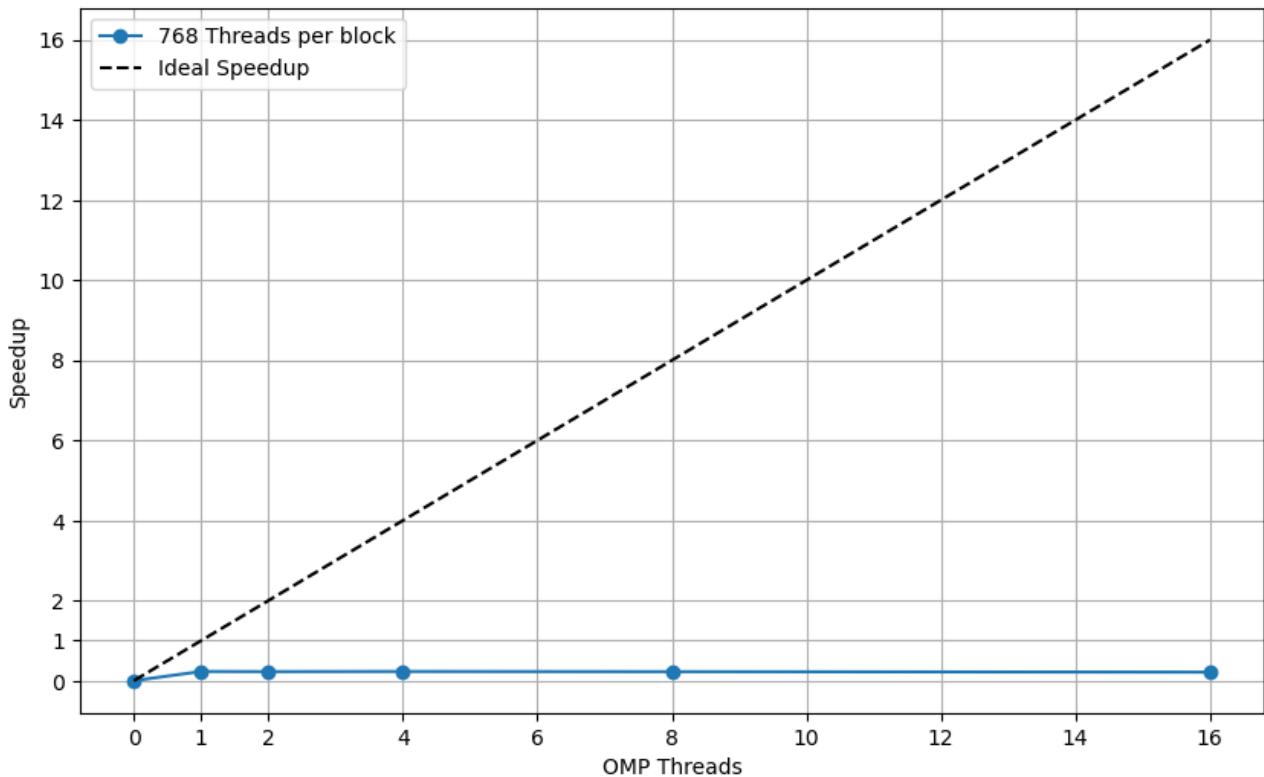
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4434622	1.0212595	0.0000000	0.4221686	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.4683305	1.0142913	0.1192778	0.3346935	0.9830635	0.9830635
OpenMp+CUDA	2	768	1.4278115	1.0153532	0.1202579	0.2921336	1.0109613	0.5054807
OpenMp+CUDA	4	768	1.4284469	1.0147814	0.1195293	0.2940692	1.0105116	0.2526279
OpenMp+CUDA	8	768	1.4510221	1.0175065	0.1205805	0.3128679	0.9947899	0.1243487
OpenMp+CUDA	16	768	1.4977417	1.0189442	0.1193379	0.3593928	0.9637591	0.0602349



### 6.3.3 Type 2

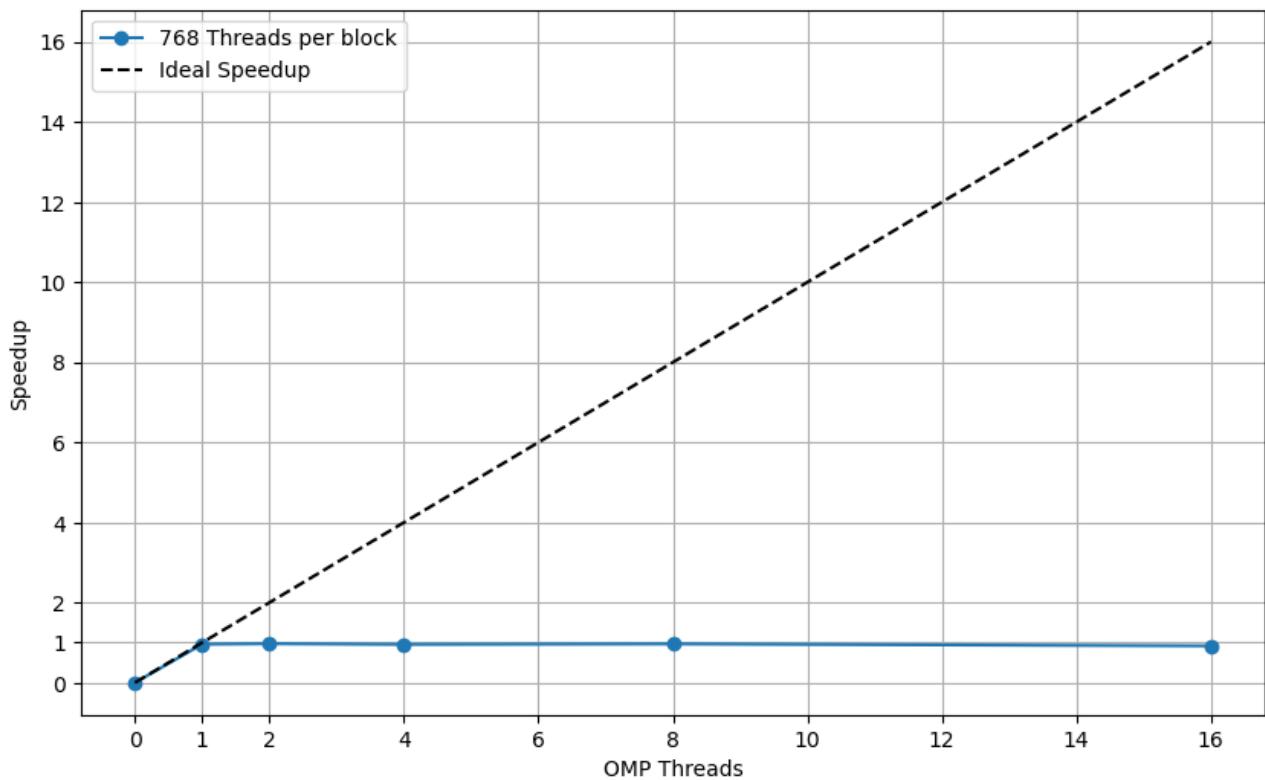
**1536 nodes & 790416 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0227595	0.0153609	0.0000000	0.0073741	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0990463	0.0151895	0.0636977	0.0201497	0.2297861	0.2297861
OpenMp+CUDA	2	768	0.1006345	0.0148671	0.0643275	0.0214311	0.2261596	0.1130798
OpenMp+CUDA	4	768	0.0990733	0.0149541	0.0631225	0.0209873	0.2297234	0.0574309
OpenMp+CUDA	8	768	0.1011845	0.0150451	0.0637307	0.0224001	0.2249304	0.0281163
OpenMp+CUDA	16	768	0.1069678	0.0150863	0.0644277	0.0274446	0.2127693	0.0132981



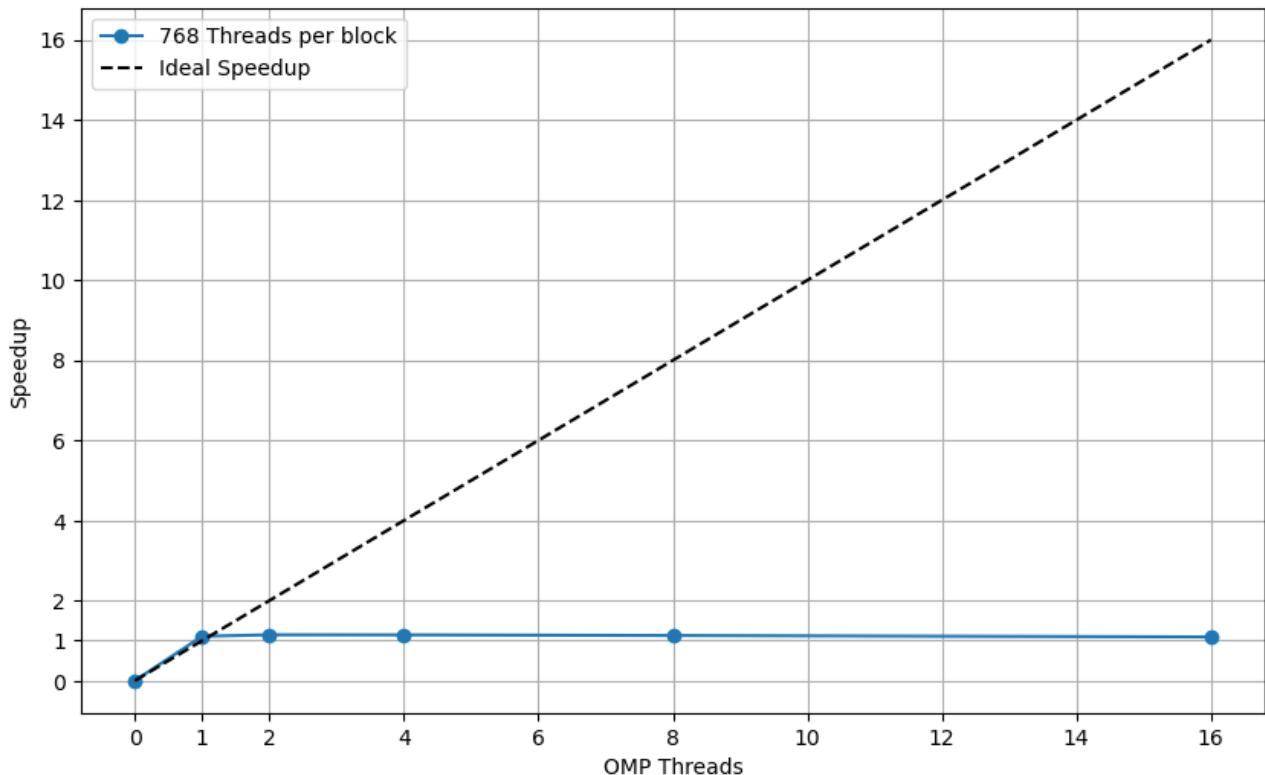
## 7552 nodes & 19101950 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5573064	0.3587729	0.0000000	0.1985069	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5802948	0.3567141	0.0811791	0.1423693	0.9603850	0.9603850
OpenMp+CUDA	2	768	0.5726398	0.3580939	0.0808323	0.1336802	0.9732233	0.4866117
OpenMp+CUDA	4	768	0.5830193	0.3577316	0.0813471	0.1439060	0.9558969	0.2389742
OpenMp+CUDA	8	768	0.5760627	0.3573705	0.0805109	0.1381478	0.9674406	0.1209301
OpenMp+CUDA	16	768	0.6065164	0.3607357	0.0809293	0.1648184	0.9188645	0.0574290



## 13568 nodes & 61665181 edges

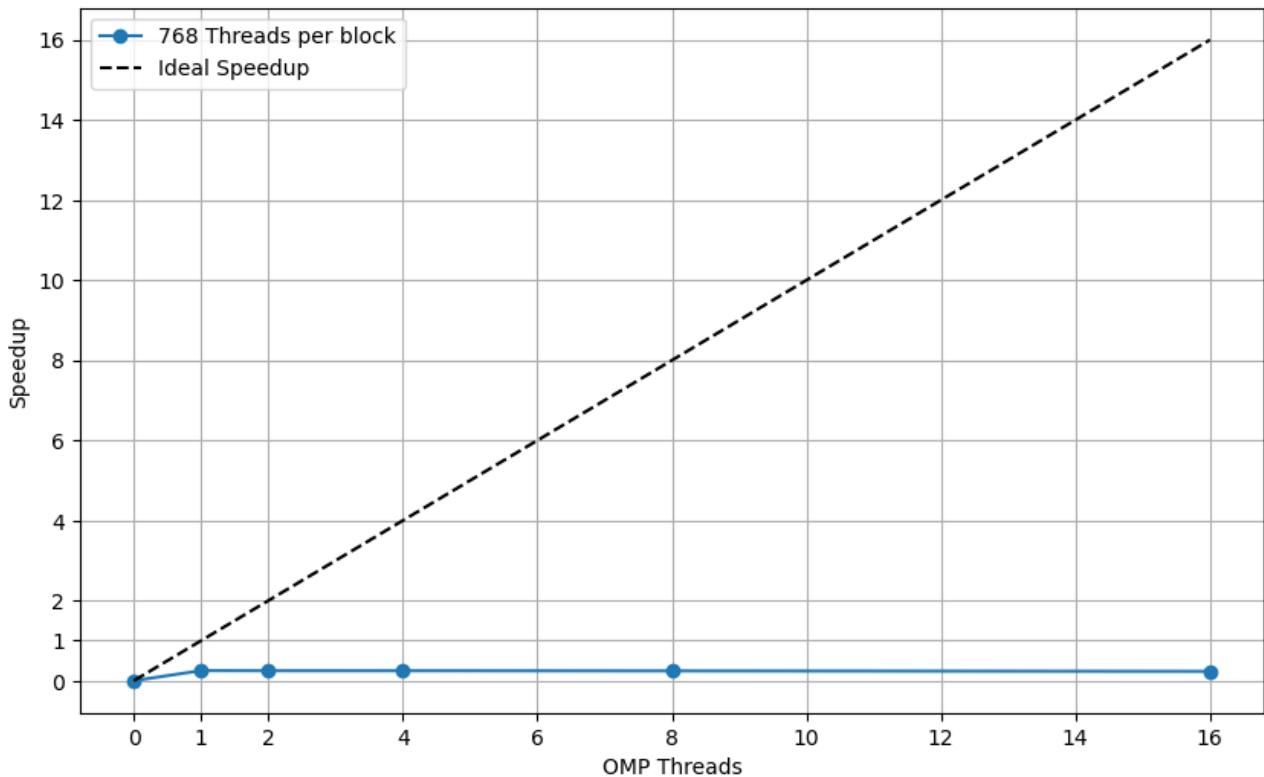
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8449185	1.2192096	0.0000000	0.6256766	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.6639267	1.2089533	0.1201793	0.3347282	1.1087739	1.1087739
OpenMp+CUDA	2	768	1.6111033	1.1993688	0.1204961	0.2911718	1.1451274	0.5725637
OpenMp+CUDA	4	768	1.6131745	1.2008835	0.1194923	0.2927316	1.1436572	0.2859143
OpenMp+CUDA	8	768	1.6350101	1.2035107	0.1202448	0.3111878	1.1283836	0.1410479
OpenMp+CUDA	16	768	1.6882392	1.2065057	0.1203535	0.3613131	1.0928064	0.0683004



### 6.3.4 Type 3

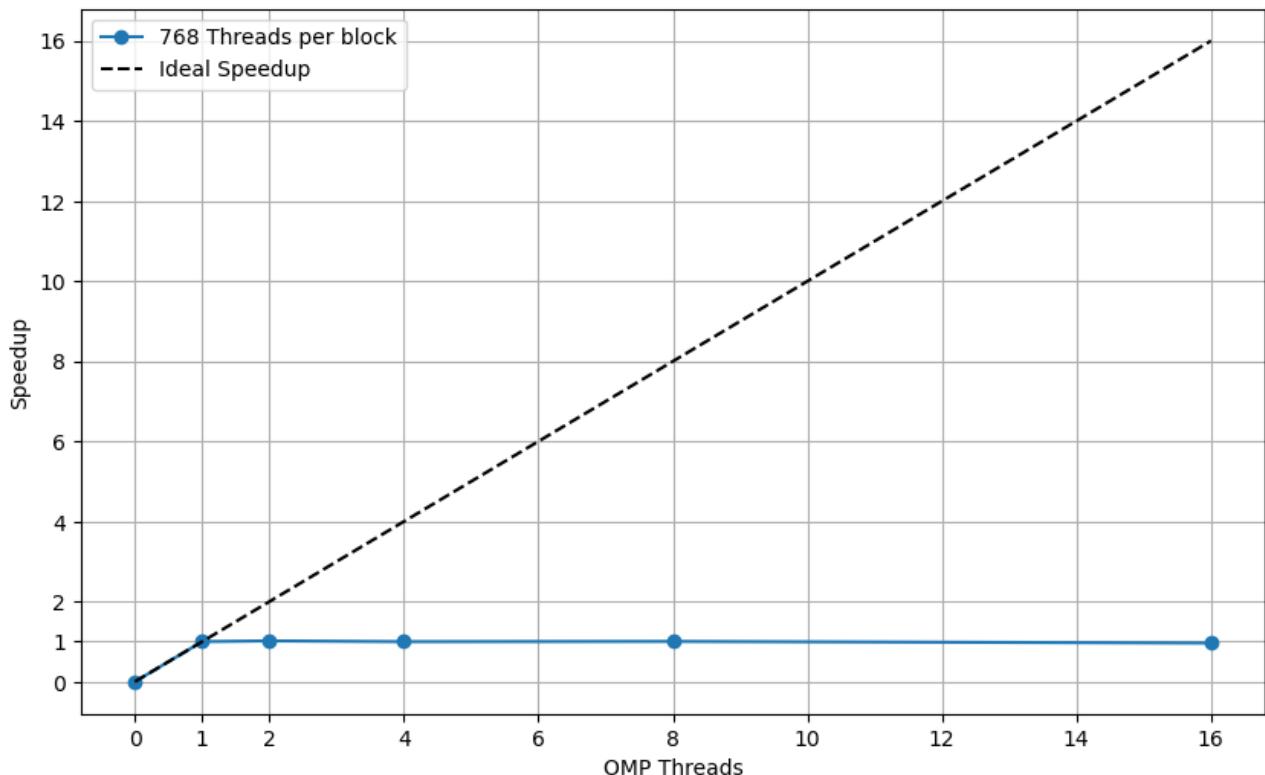
**1536 nodes & 588142 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0251513	0.0162583	0.0000000	0.0088655	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1000107	0.0161777	0.0638913	0.0199331	0.2514863	0.2514863
OpenMp+CUDA	2	768	0.1012829	0.0159433	0.0637487	0.0215823	0.2483276	0.1241638
OpenMp+CUDA	4	768	0.1013688	0.0160677	0.0643898	0.0209029	0.2481171	0.0620293
OpenMp+CUDA	8	768	0.1029315	0.0160955	0.0645603	0.0222670	0.2443501	0.0305438
OpenMp+CUDA	16	768	0.1077803	0.0161534	0.0643211	0.0272964	0.2333574	0.0145848



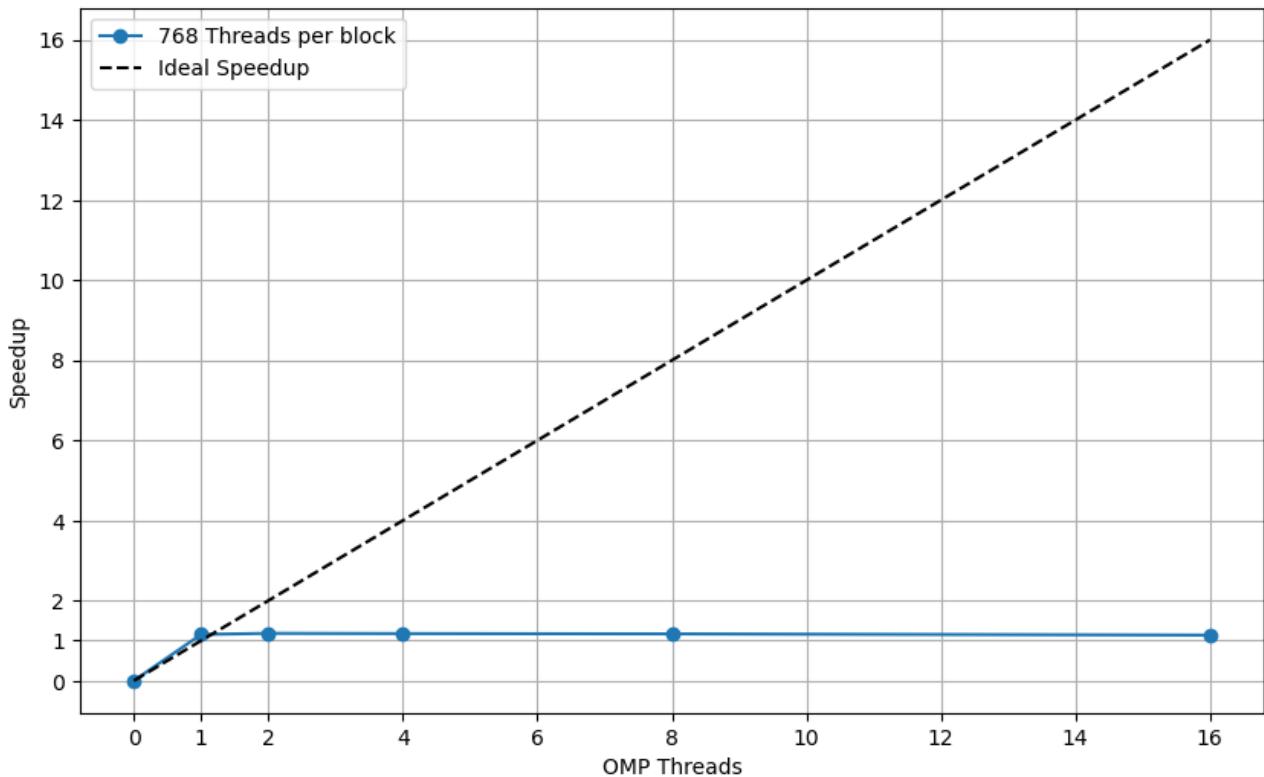
## 7552 nodes & 14260151 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6043514	0.3795036	0.0000000	0.2248215	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6040770	0.3808139	0.0810609	0.1421691	1.0004542	1.0004542
OpenMp+CUDA	2	768	0.5929361	0.3791136	0.0801351	0.1336537	1.0192522	0.5096261
OpenMp+CUDA	4	768	0.6046619	0.3797040	0.0808881	0.1440369	0.9994864	0.2498716
OpenMp+CUDA	8	768	0.6009273	0.3818345	0.0812674	0.1377926	1.0056981	0.1257123
OpenMp+CUDA	16	768	0.6269200	0.3819797	0.0803817	0.1645254	0.9640008	0.0602501



## 13568 nodes & 46014096 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.0129378	1.2848842	0.0000000	0.7280221	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.7437696	1.2906131	0.1196490	0.3334401	1.1543600	1.1543600
OpenMp+CUDA	2	768	1.7029847	1.2890923	0.1213454	0.2924797	1.1820058	0.5910029
OpenMp+CUDA	4	768	1.7116852	1.2957364	0.1214486	0.2944333	1.1759977	0.2939994
OpenMp+CUDA	8	768	1.7222089	1.2905665	0.1187697	0.3128057	1.1688116	0.1461015
OpenMp+CUDA	16	768	1.7704029	1.2915319	0.1194941	0.3593095	1.1369942	0.0710621



## 6.4 Optimization 3

Optimization 3 does not introduce an improvement in speedup and performance compared to O0.

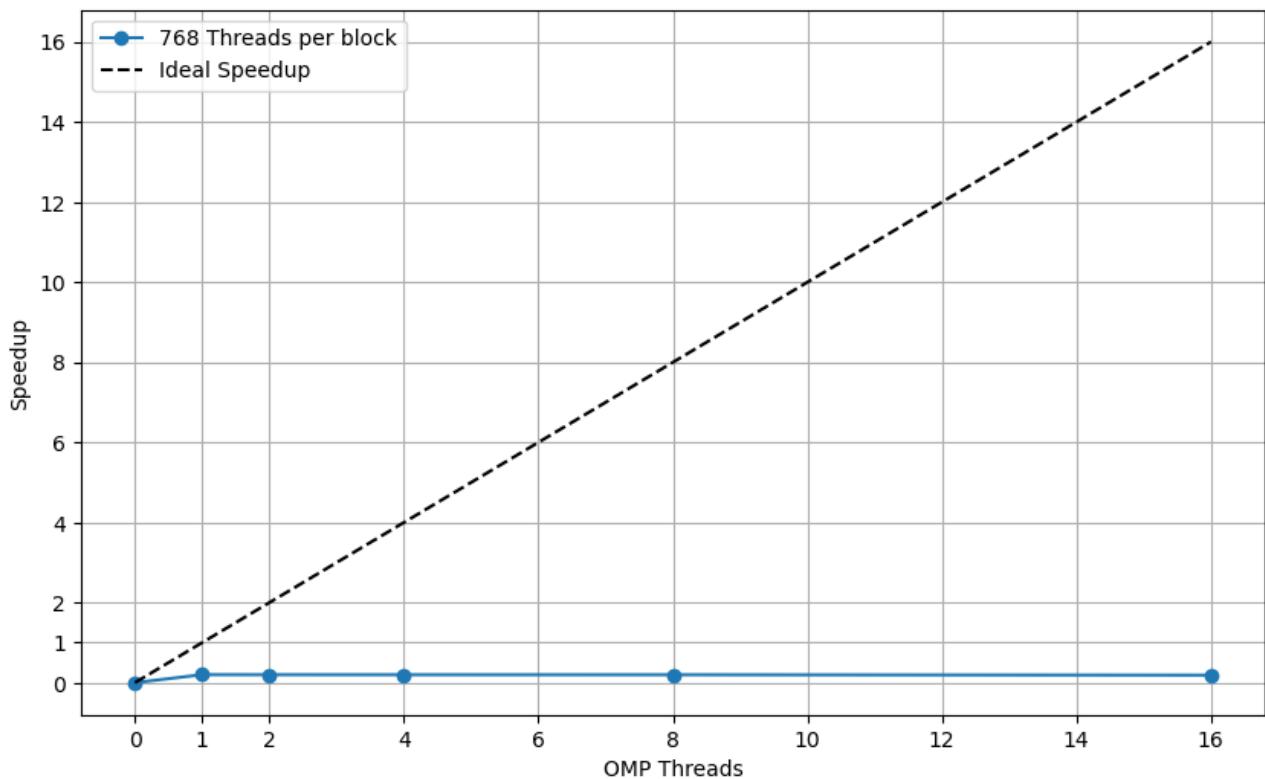
The algorithm remains efficient for medium and mainly for large, type 3, graphs.

Optimization 0 remains the one with the best performance

### 6.4.1 Type 0

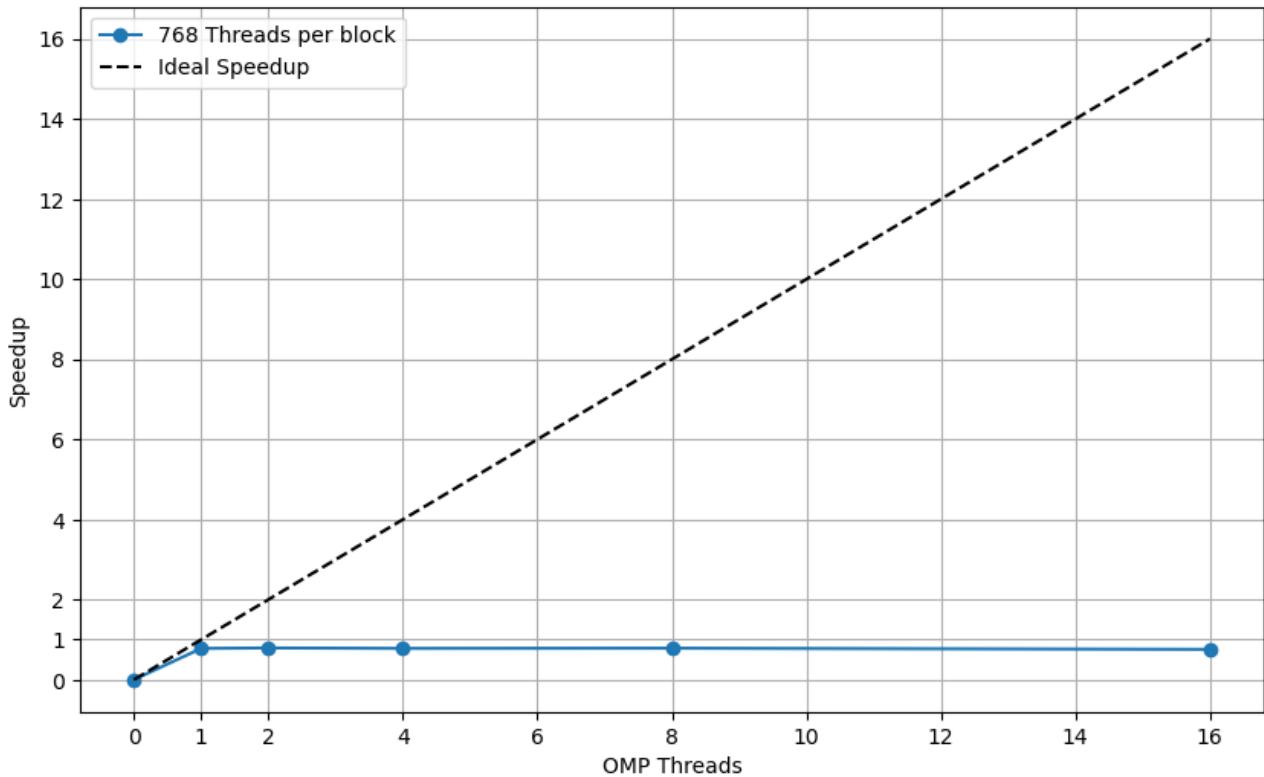
**1536 nodes & 2357760 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0209451	0.0178254	0.0000000	0.0030939	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1048261	0.0175357	0.0668853	0.0203954	0.1998084	0.1998084
OpenMp+CUDA	2	768	0.1053052	0.0172359	0.0662803	0.0217793	0.1988993	0.0994497
OpenMp+CUDA	4	768	0.1055786	0.0173945	0.0670202	0.0211543	0.1983843	0.0495961
OpenMp+CUDA	8	768	0.1058249	0.0172741	0.0660519	0.0224891	0.1979226	0.0247403
OpenMp+CUDA	16	768	0.1113637	0.0173256	0.0660402	0.0279883	0.1880786	0.0117549



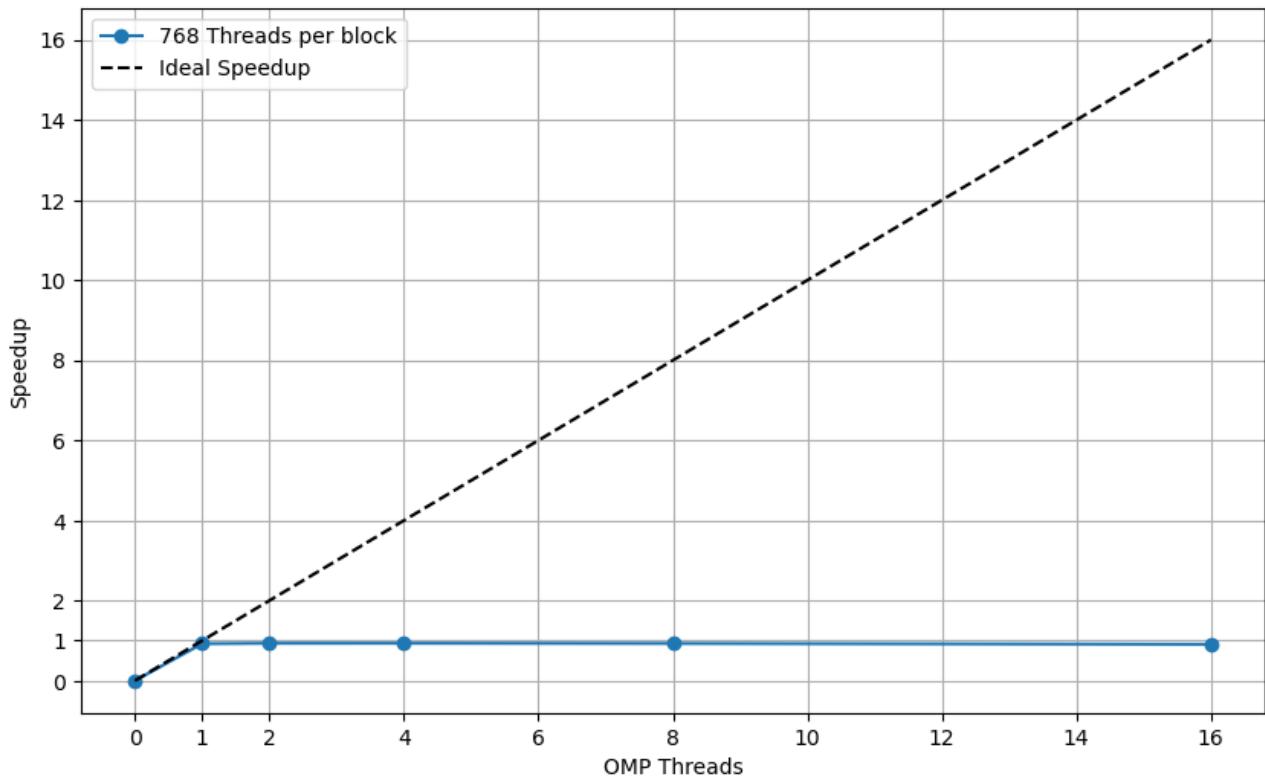
## 7552 nodes & 57025152 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5237787	0.4485539	0.0000000	0.0752013	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6716219	0.4468490	0.0820283	0.1427083	0.7798713	0.7798713
OpenMp+CUDA	2	768	0.6611663	0.4457055	0.0816097	0.1338159	0.7922042	0.3961021
OpenMp+CUDA	4	768	0.6706882	0.4447443	0.0821695	0.1437385	0.7809570	0.1952393
OpenMp+CUDA	8	768	0.6664409	0.4466204	0.0825003	0.1372843	0.7859342	0.0982418
OpenMp+CUDA	16	768	0.6916817	0.4458313	0.0816557	0.1641593	0.7572539	0.0473284



## 13568 nodes & 184077056 edges

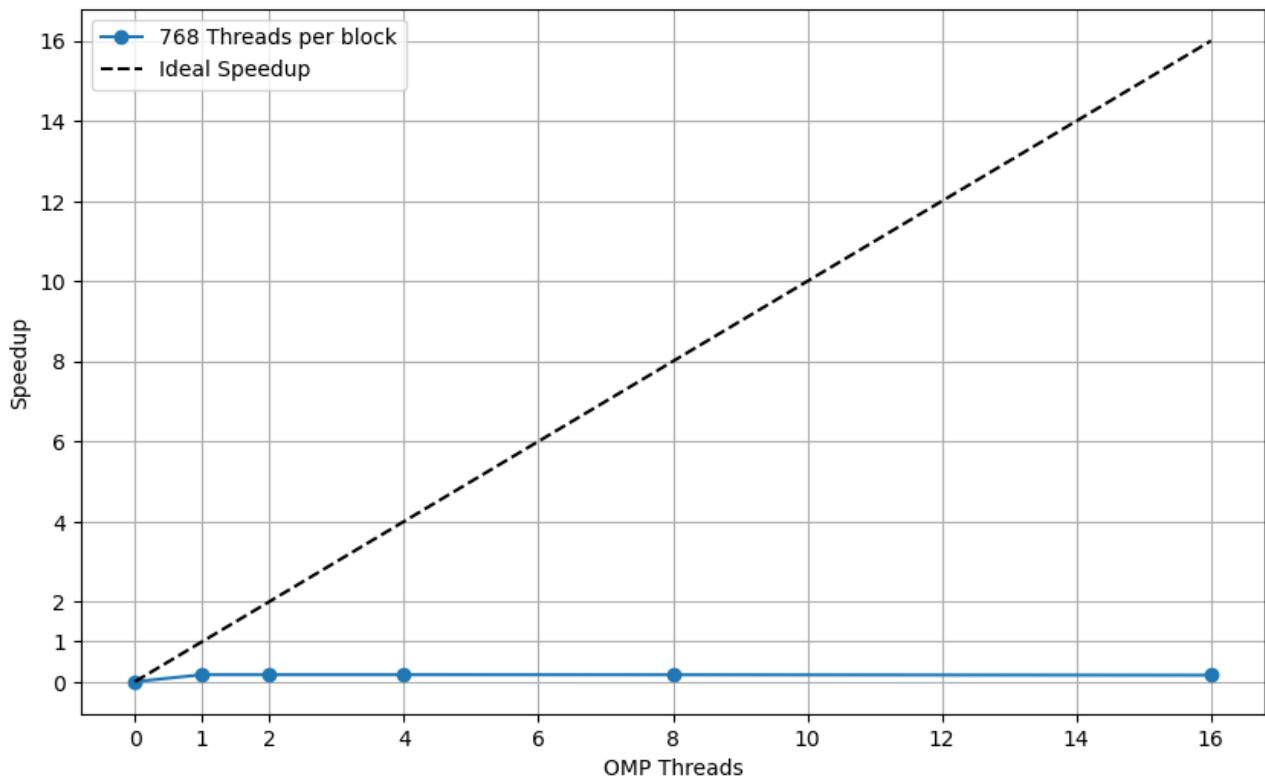
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.3147068	2.0656339	0.000000	0.2490404	1.000000	1.000000
OpenMp+CUDA	1	768	2.5068909	2.0550598	0.1220758	0.3296881	0.9233377	0.9233377
OpenMp+CUDA	2	768	2.4770669	2.0578029	0.1229257	0.2962722	0.9344547	0.4672273
OpenMp+CUDA	4	768	2.4750447	2.0556728	0.1227251	0.2965794	0.9352182	0.2338045
OpenMp+CUDA	8	768	2.4910981	2.0538112	0.1219959	0.3152233	0.9291913	0.1161489
OpenMp+CUDA	16	768	2.5485995	2.0632782	0.1218429	0.3634099	0.9082270	0.0567642



### 6.4.2 Type 1

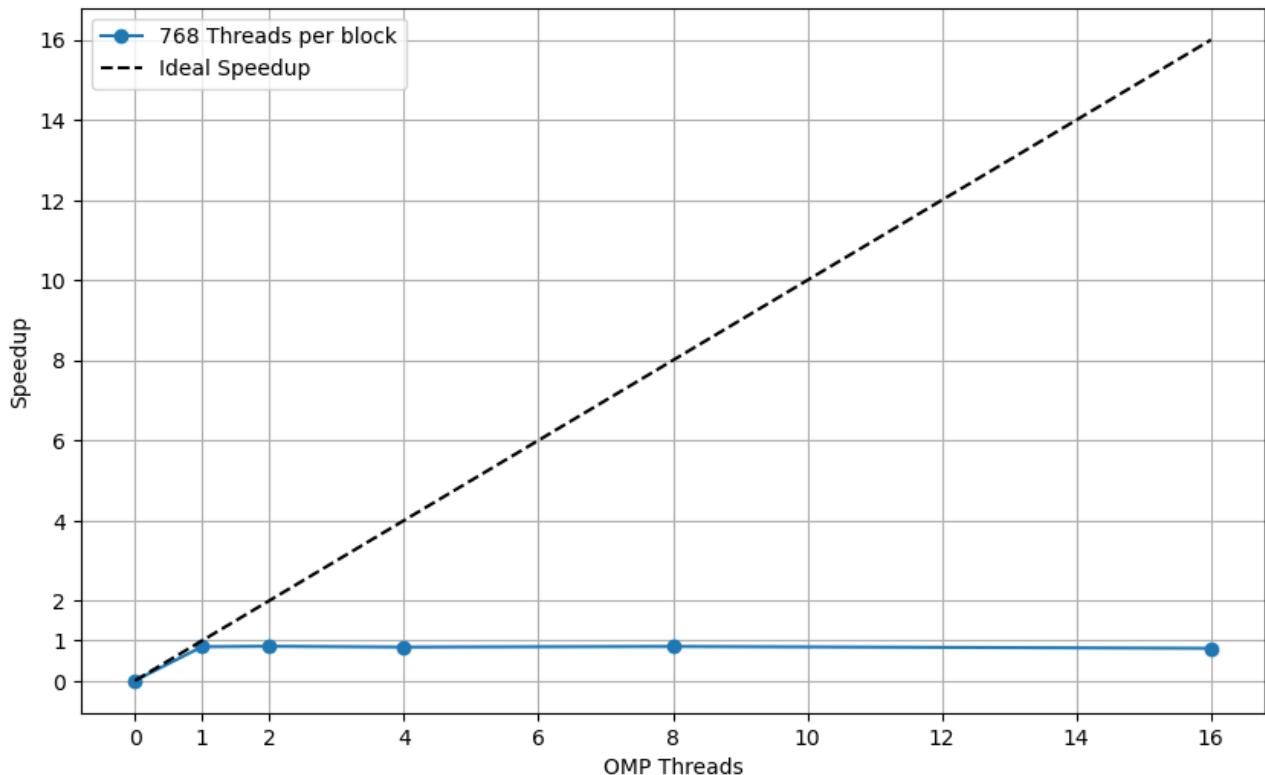
**1536 nodes & 1061043 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0172293	0.0124619	0.0000000	0.0047413	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.0989708	0.0122807	0.0662453	0.0204365	0.1740843	0.1740843
OpenMp+CUDA	2	768	0.0990189	0.0123449	0.0648653	0.0217993	0.1739997	0.0869999
OpenMp+CUDA	4	768	0.0988530	0.0122389	0.0652801	0.0213251	0.1742918	0.0435729
OpenMp+CUDA	8	768	0.0999301	0.0122403	0.0651846	0.0224962	0.1724131	0.0215516
OpenMp+CUDA	16	768	0.1060231	0.0122205	0.0659524	0.0278411	0.1625049	0.0101566



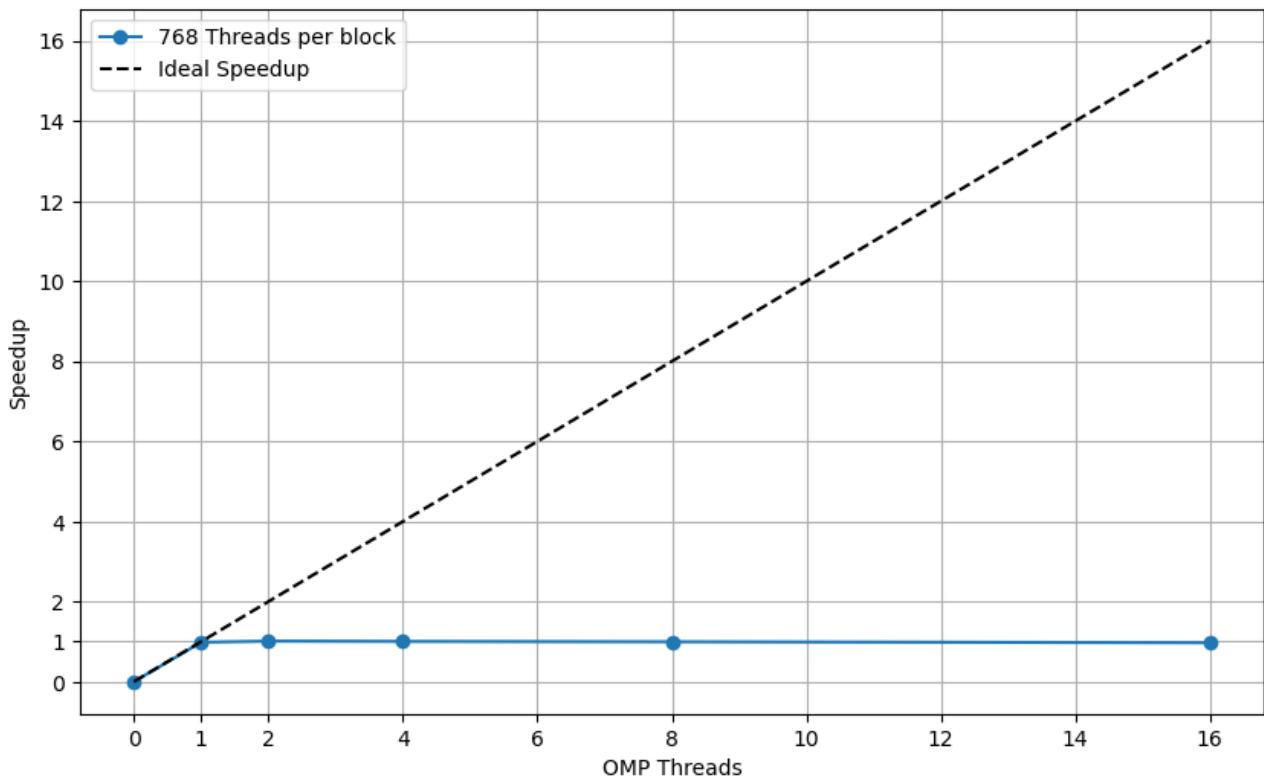
## 7552 nodes & 25660141 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.4371048	0.2917672	0.0000000	0.1453100	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5156147	0.2908033	0.0813153	0.1434642	0.8477354	0.8477354
OpenMp+CUDA	2	768	0.5085802	0.2923685	0.0822017	0.1339779	0.8594609	0.4297305
OpenMp+CUDA	4	768	0.5209723	0.2939885	0.0819029	0.1450486	0.8390173	0.2097543
OpenMp+CUDA	8	768	0.5115264	0.2911813	0.0818333	0.1384795	0.8545107	0.1068138
OpenMp+CUDA	16	768	0.5407760	0.2921766	0.0826074	0.1659589	0.8082918	0.0505182



## 13568 nodes & 82829469 edges

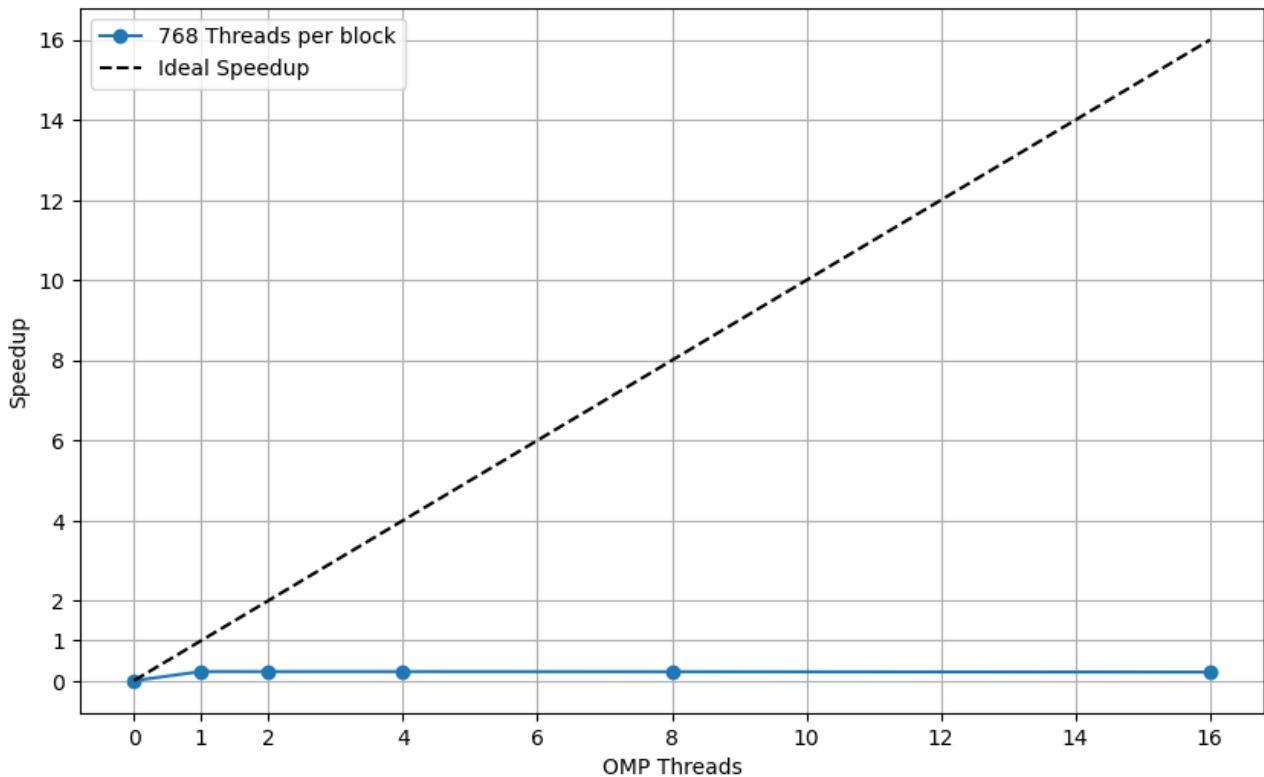
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.4508061	1.0183621	0.0000000	0.4324105	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.4757487	1.0232181	0.1213804	0.3310849	0.9830983	0.9830983
OpenMp+CUDA	2	768	1.4321161	1.0186548	0.1207754	0.2926213	1.0130506	0.5065253
OpenMp+CUDA	4	768	1.4422593	1.0271542	0.1207019	0.2943375	1.0059259	0.2514815
OpenMp+CUDA	8	768	1.4563663	1.0247162	0.1209199	0.3106651	0.9961821	0.1245228
OpenMp+CUDA	16	768	1.4922478	1.0168128	0.1203519	0.3550173	0.9722287	0.0607643



### 6.4.3 Type 2

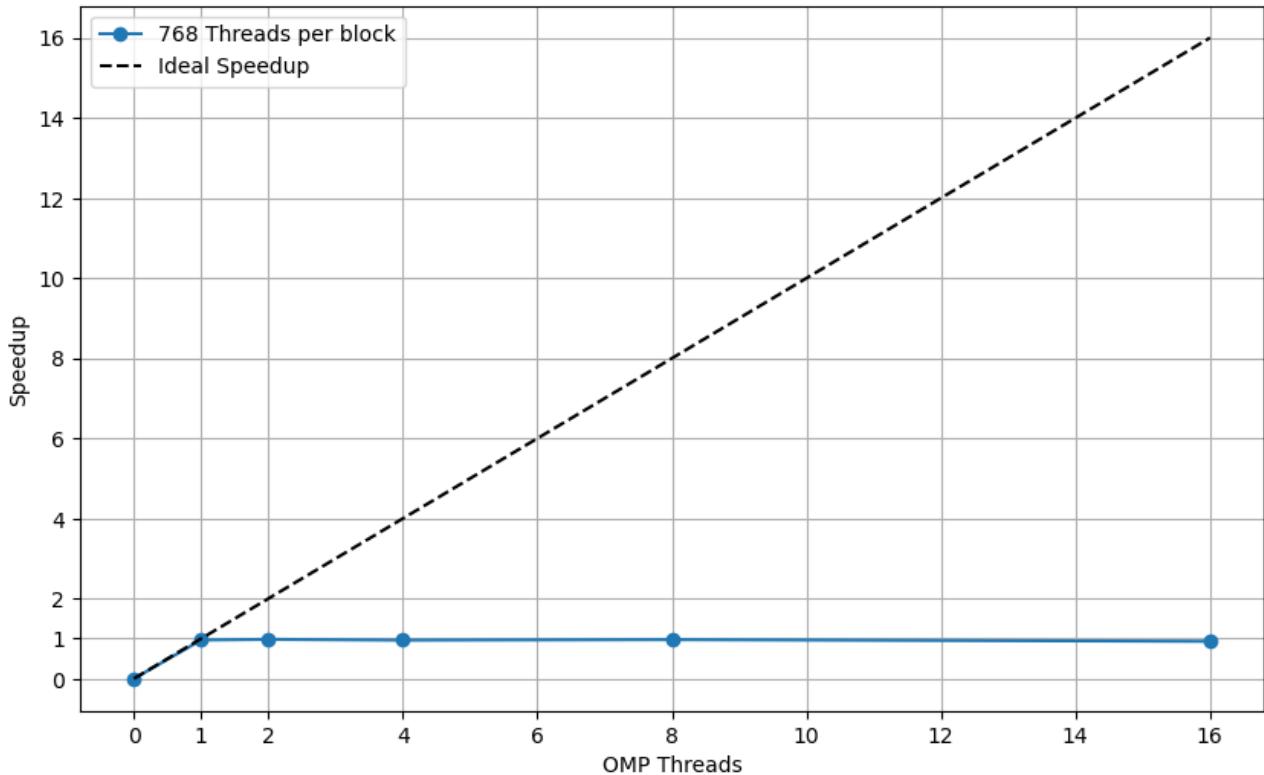
**1536 nodes & 790416 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0228476	0.0154823	0.0000000	0.0073375	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1001927	0.0150033	0.0650508	0.0201285	0.2280365	0.2280365
OpenMp+CUDA	2	768	0.1004905	0.0146442	0.0641570	0.0216807	0.2273607	0.1136804
OpenMp+CUDA	4	768	0.1004314	0.0147827	0.0645710	0.0210676	0.2274946	0.0568736
OpenMp+CUDA	8	768	0.1018597	0.0148111	0.0648044	0.0222341	0.2243047	0.0280381
OpenMp+CUDA	16	768	0.1065799	0.0148513	0.0637972	0.0279228	0.2143707	0.0133982



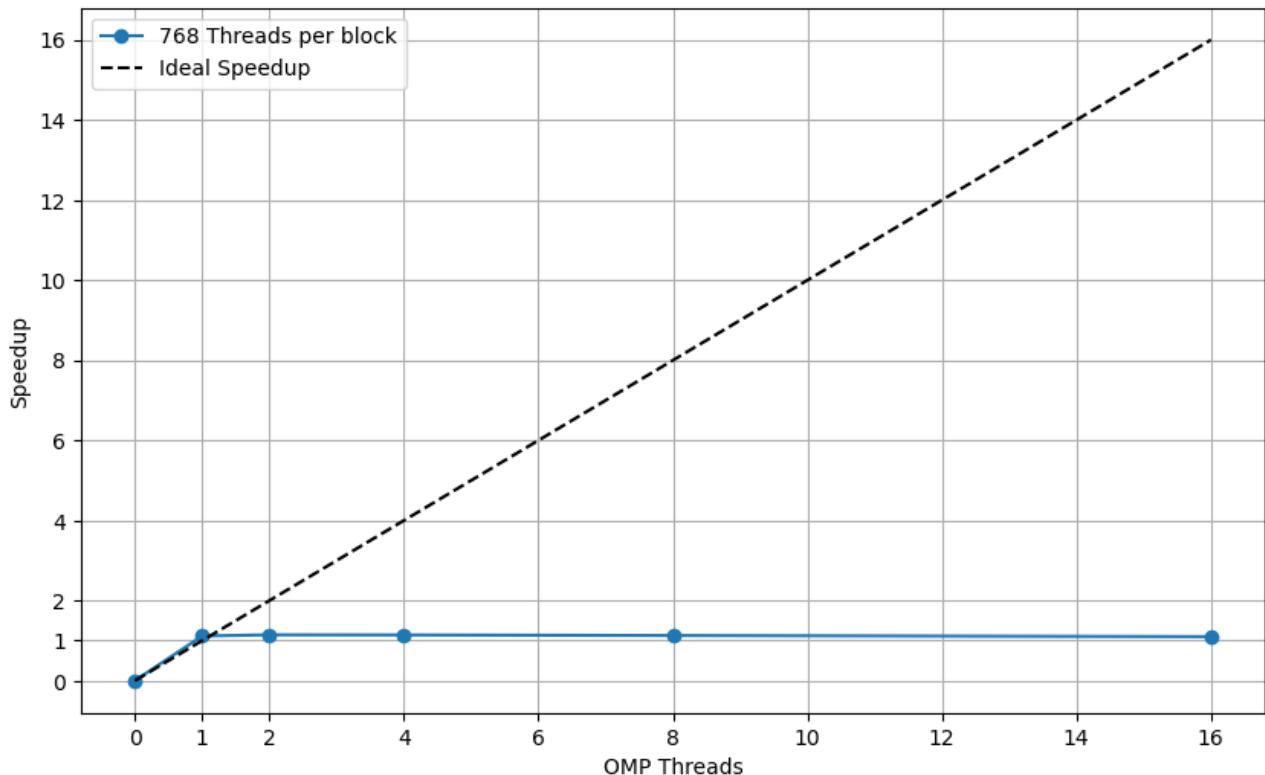
## 7552 nodes & 19101950 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.5591937	0.3584046	0.0000000	0.2007601	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.5796615	0.3544599	0.0824424	0.1427261	0.9646900	0.9646900
OpenMp+CUDA	2	768	0.5694961	0.3540001	0.0816574	0.1338065	0.9819096	0.4909548
OpenMp+CUDA	4	768	0.5816514	0.3535521	0.0823962	0.1456711	0.9613897	0.2403474
OpenMp+CUDA	8	768	0.5735578	0.3515620	0.0821371	0.1398262	0.9749561	0.1218695
OpenMp+CUDA	16	768	0.5999834	0.3533631	0.0818609	0.1647264	0.9320152	0.0582510



## 13568 nodes & 61665181 edges

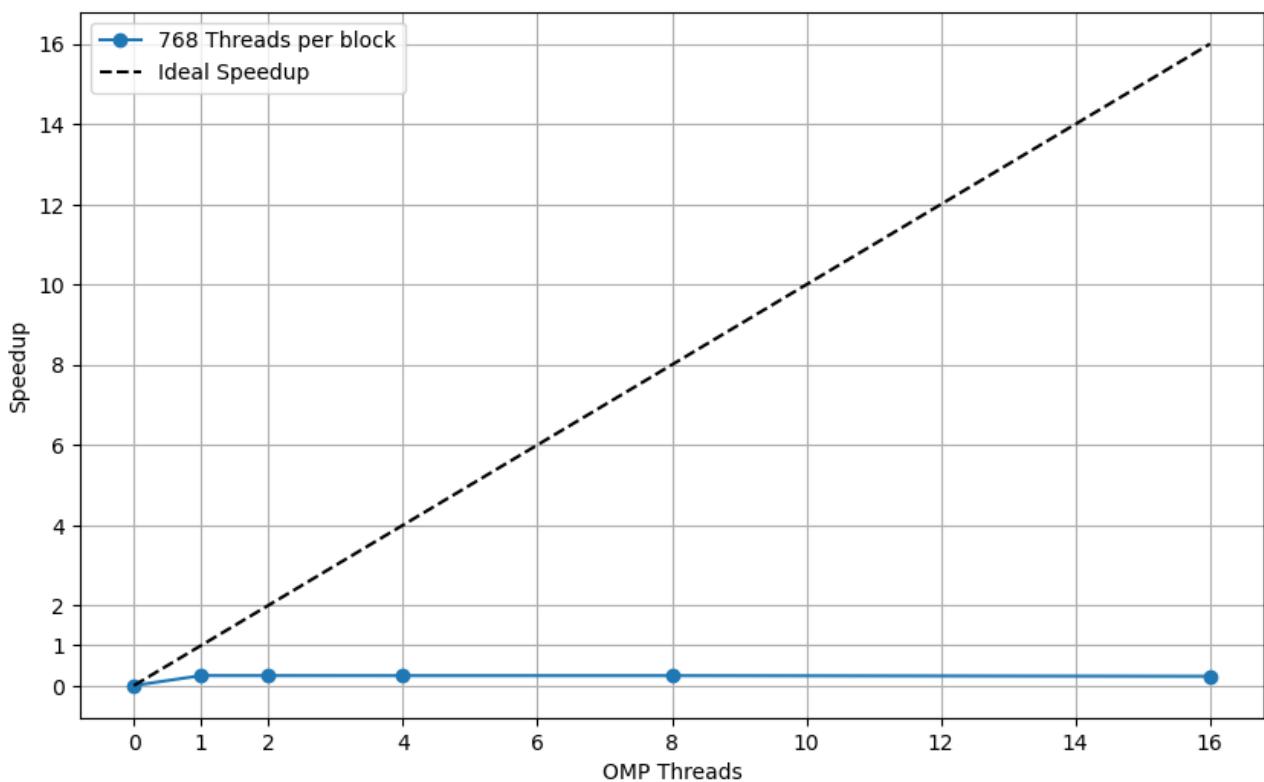
	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	1.8570059	1.2169153	0.0000000	0.6400563	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.6648618	1.2123030	0.1216324	0.3308607	1.1154115	1.1154115
OpenMp+CUDA	2	768	1.6226150	1.2073403	0.1220124	0.2931979	1.1444526	0.5722263
OpenMp+CUDA	4	768	1.6272685	1.2121425	0.1209272	0.2941276	1.1411798	0.2852950
OpenMp+CUDA	8	768	1.6462224	1.2150861	0.1217337	0.3093375	1.1280407	0.1410051
OpenMp+CUDA	16	768	1.6883224	1.2078148	0.1202805	0.3601608	1.0999119	0.0687445



#### 6.4.4 Type 3

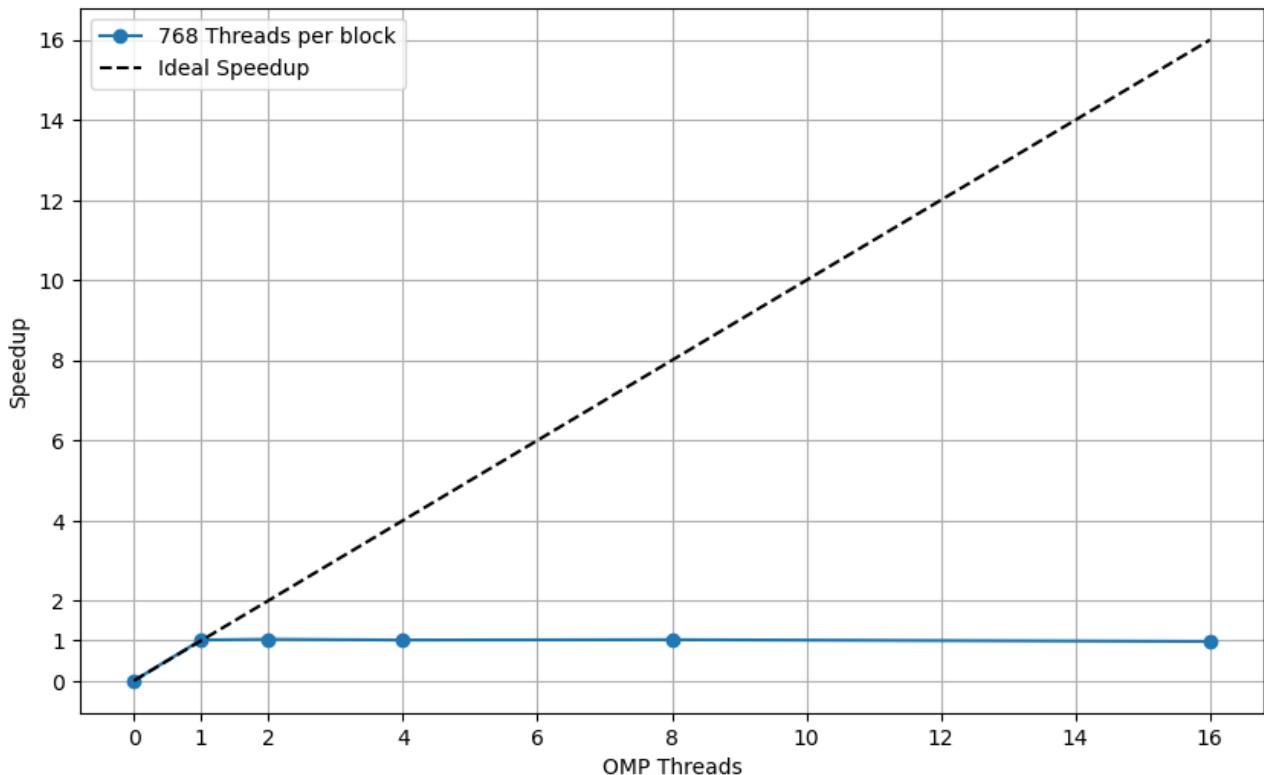
**1536 nodes & 588142 edges**

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.0254169	0.0165269	0.0000000	0.0088626	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.1026247	0.0159541	0.0665188	0.0201417	0.2476689	0.2476689
OpenMp+CUDA	2	768	0.1029043	0.0158452	0.0651667	0.0218829	0.2469958	0.1234979
OpenMp+CUDA	4	768	0.1032285	0.0158286	0.0660758	0.0213141	0.2462202	0.0615550
OpenMp+CUDA	8	768	0.1032129	0.0158583	0.0647064	0.0226391	0.2462574	0.0307822
OpenMp+CUDA	16	768	0.1094384	0.0159918	0.0658255	0.0276120	0.2322488	0.0145155



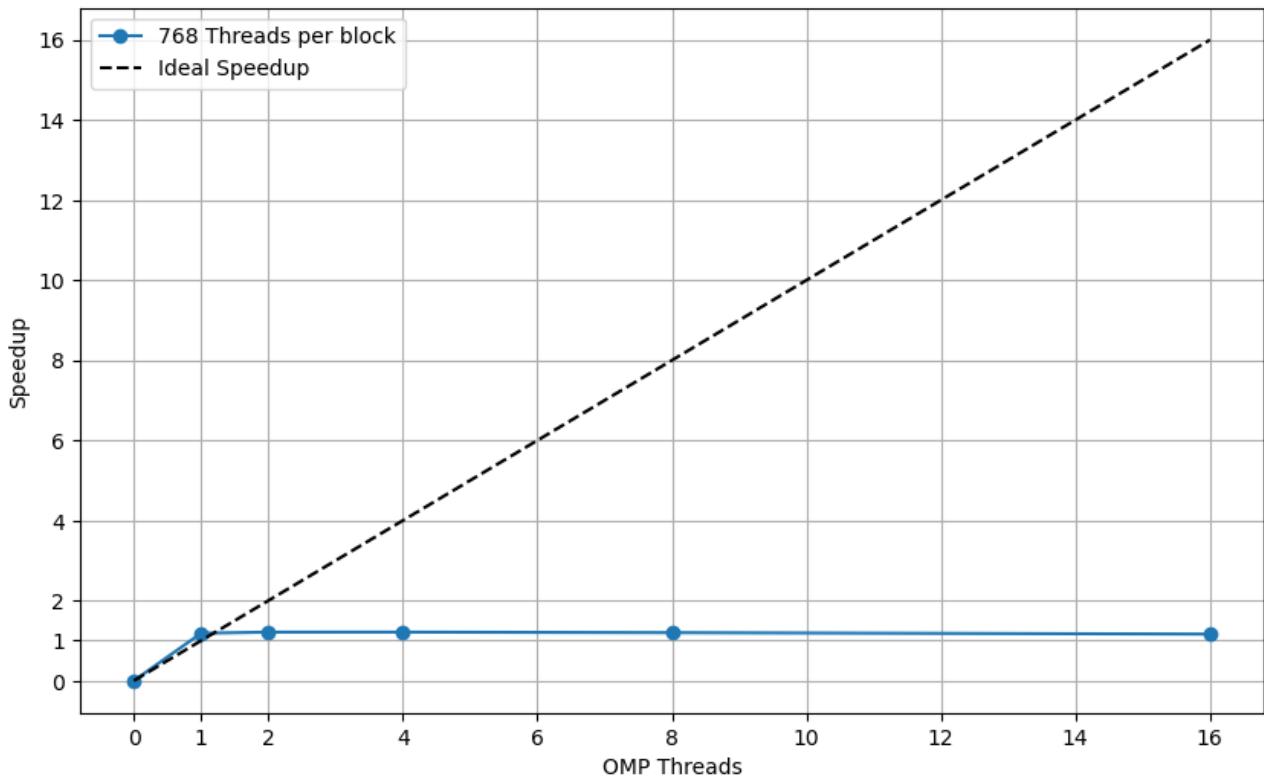
## 7552 nodes & 14260151 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	0.6107635	0.3838983	0.0000000	0.2268377	1.0000000	1.0000000
OpenMp+CUDA	1	768	0.6006327	0.3761459	0.0817578	0.1426959	1.0168669	1.0168669
OpenMp+CUDA	2	768	0.5910973	0.3745383	0.0826955	0.1338290	1.0332707	0.5166353
OpenMp+CUDA	4	768	0.6013736	0.3752033	0.0819806	0.1441570	1.0156141	0.2539035
OpenMp+CUDA	8	768	0.5966886	0.3755356	0.0820548	0.1390651	1.0235884	0.1279486
OpenMp+CUDA	16	768	0.6235841	0.3754244	0.0829057	0.1652225	0.9794406	0.0612150



## 13568 nodes & 46014096 edges

	OMP	CUDA	total time	graph creation time	GPU allocation time	Dijkstra execution time	Speedup	Efficiency
Sequential	0	0	2.0507477	1.3049343	0.0000000	0.7457805	1.0000000	1.0000000
OpenMp+CUDA	1	768	1.7305768	1.2838787	0.1212714	0.3253623	1.1850082	1.1850082
OpenMp+CUDA	2	768	1.6927841	1.2827099	0.1202575	0.2897501	1.2114644	0.6057322
OpenMp+CUDA	4	768	1.6917415	1.2780559	0.1204470	0.2931731	1.2122110	0.3030527
OpenMp+CUDA	8	768	1.7098761	1.27770920	0.1215935	0.3111261	1.1993546	0.1499193
OpenMp+CUDA	16	768	1.7626621	1.2846777	0.1213355	0.3565831	1.1634378	0.0727149



## *Final Considerations*

### 7.1 “OpenMP + MPI”

In this case, parallel algorithm introduces improvements for medium and large graphs, and it has best performance for less dense graphs with medium size.

It uses MPI for dividing the graph between processes and OpenMP for loops between threads in parallel for each process.

For small graphs, performance are very low, due to MPI environment initialization time.

Best compilation optimization, as evaluated after testing, is O0.

### 7.2 “OpenMP + CUDA”

In this case, parallel algorithm introduces improvements for medium and large graphs, and it has best performance for less dense graphs with large size.

For small graphs performances are very low, due to memory exchanges and synchronizations between host and device.

Best compilation optimization, as evaluated after testing, is O0.

# API Documentation

## Graph Functions Documentation

### 1. **int createDenseGraph (int N, int\* adj\_matrix, int seed )**

Creates an adjacency matrix for a dense graph.

#### Parameters:

N Number of vertices.

adj\_matrix The adjacency matrix to be filled.

seed Random seed for reproducibility.

#### Returns:

Number of edges in the graph.

### 2. **int createDisconnectedGraph (int N, int\* adj\_matrix )**

Creates an adjacency matrix for a disconnected graph.

#### Parameters:

N Number of vertices.

adj\_matrix The adjacency matrix to be filled.

#### Returns:

Number of edges in the graph.

### 3. **int createFullConnectedGraph (int N, int\* adj\_matrix, int seed )**

Creates an adjacency matrix for a fully connected graph.

#### Parameters:

N Number of vertices.

adj\_matrix The adjacency matrix to be filled.

seed Random seed for reproducibility.

#### Returns:

Number of edges in the graph.

### 4. **int\* createGraph (int N )**

Allocates and returns a vector of length  $N \times N$ .

#### Parameters:

N Number of vertices.

#### Returns:

Pointer to the allocated vector.

**5. int createGraphByType (int type, int number\_of\_vertices, int\* adj\_matrix, int seed )**

Creates a graph based on the specified type.

**Parameters:**

type Type of the graph (0: Fully Connected, 1: Dense, 2: Intermediate Density, 3: Sparse).

number\_of\_vertices Number of vertices in the graph.

adj\_matrix The adjacency matrix to be filled.

seed Random seed for reproducibility.

**Returns:**

Number of edges in the graph.

**6. int createIntermediateDensityGraph (int N, int\* adj\_matrix, int seed )**

Creates an adjacency matrix for a graph with intermediate density.

**Parameters:**

N Number of vertices.

adj\_matrix The adjacency matrix to be filled.

seed Random seed for reproducibility.

**Returns:**

Number of edges in the graph.

**7. int createSparseGraph (int N, int\* adj\_matrix, int seed )**

Creates an adjacency matrix for a sparse graph.

**Parameters:**

N Number of vertices.

adj\_matrix The adjacency matrix to be filled.

seed Random seed for reproducibility.

**Returns:**

Number of edges in the graph.

**8. void freeGraph (int\* adj\_matrix)**

Frees the memory allocated for the adjacency matrix.

**Parameters:**

adj\_matrix The adjacency matrix to be freed.

## Utility Functions Documentation

### 1. **void print\_matrix (int N, int\* adj\_matrix )**

Prints a matrix to the console.

#### Parameters:

N Number of rows/columns in the matrix.

adj\_matrix The matrix to be printed.

### 2. **void print\_matrix\_on\_file (FILE\* fp, int N, int\* adj\_matrix )**

Prints a matrix to a file.

#### Parameters:

fp File pointer.

N Number of rows/columns in the matrix.

adj\_matrix The matrix to be printed.

### 3. **void print\_vector (int N, int\* vec )**

Prints a vector to the console.

#### Parameters:

N Length of the vector.

vec The vector to be printed.

### 4. **void print\_vector\_on\_file (FILE\* fp, int N, int\* vec )**

Prints a vector to a file.

#### Parameters:

fp File pointer.

N Length of the vector.

vec The vector to be printed.

### 5. **void initializeArray (int\* array, int size, int value)**

Initializes an array with a specified value.

#### Parameters:

array The array to be initialized.

size The size of the array.

value The value to set for each element.

## Dijkstra\_serial Functions Documentation

### 1. **void Dijkstra (int source, int num\_vertices, int\* graph, int\* distances, int\* visited, int\* path)**

Performs Dijkstra's algorithm to find the shortest paths from a source vertex to all other vertices in a graph.

#### Parameters:

source Source vertex for the shortest paths.

num\_vertices Total number of vertices in the graph.

graph the adjacency matrix representing the graph.

distances Array to store the calculated shortest distances from the source.

visited Array to mark visited vertices during the algorithm.

path Array to store the predecessor in shortest paths.

### 2. **int main (int argc, char\* argv[] )**

Main function to execute Dijkstra's algorithm sequentially.

#### Parameters:

argc Number of command-line arguments.

argv Array of command-line argument strings.

#### Returns:

0 on successful execution, 1 on failure.

# Dijkstra\_MPIOpenMP Functions Documentation

## 1. int checkNprocess (int n\_vertices, int nprocess )

Checks if the given number of MPI processes is valid for the specified number of vertices.

### Parameters:

n\_vertices Number of vertices in the graph.  
nprocess Number of MPI processes.

### Returns:

TRUE if the number of MPI processes is valid, FALSE otherwise.

## 2. void Dijkstra (int n\_vertices, int source, int\* local\_adj\_matrix, int\* distance, int\* path, MPI\_Comm communicator )

Performs Dijkstra's algorithm to find the shortest paths from a source vertex to all other vertices in a graph.

### Parameters:

n\_vertices Number of vertices in the graph.  
source Source vertex for the shortest paths.  
local\_adj\_matrix Weighted adjacency matrix representing the local portion of the graph.  
distance Array to store the calculated shortest distances.  
path Array to store the shortest paths.  
communicator MPI communicator.

## 3. int main (int argc, char\* argv[] )

Main function to execute the MPI+OpenMP version of Dijkstra's algorithm.

### Parameters:

argc Number of command-line arguments.  
argv Array of command-line argument strings.

### Returns:

0 on successful execution, 1 on failure.

## Dijkstra\_cuda Functions Documentation

### 1. int searchMinIndex (int\* distances, int\* marker, int array\_size)

Finds the index of the minimum distance in the distances array that has not been visited.

#### Parameters:

distances      Array of distances.  
marker      Array indicating whether a node has been visited.  
Array\_size      Size of the arrays.

#### Returns:

Index of the minimum distance.

### 2. \_\_global\_\_ void cuda\_update\_distance (int\* graph, int\* node\_dist, int\* path, int\* visited, int source, int num\_vertices)

CUDA kernel to update distances based on the Dijkstra algorithm.

#### Parameters:

graph      Adjacency matrix of the graph.  
node\_dist      Array of distances from the source vertex.  
Path      Array representing the predecessor shortest path from the source node.  
visited      Array indicating whether a node has been visited.  
source      Source node to update distances for Dijkstra's algorithm.  
num\_vertices      Number of vertices in the graph.

### 3. int main (int argc, char\* argv[] )

Main function implementing the OMP/CUDA Dijkstra algorithm.

#### Parameters:

argc      Number of command-line arguments.  
argv      Array of command-line argument strings.

#### Returns:

0 on successful execution, 1 on failure.

## How to run

Code for generating directories, tables and plots requires python interpreter and matplotlib library, the last to be installed with the command ***pip3 install matplotlib***.

For running project programs, the following steps should be executed:

1. Navigate to the folder containing the makefile
2. To clear previously obtained achievements and previous builds, enter the command: ***make clean***
3. To generate the necessary directories and compile and linking the various source codes, enter the command: ***make all***
4. To run the algorithm for making tests, producing results, measurements, graphs and tables, enter the command: ***make test***
5. To clear previously obtained achievements and previous builds, enter the command: ***make clean***

Results are found in the folders "OPENMP+MPI" & "OPENMP+CUDA", organized by version.

Results of the algorithms can be viewed in the "ResultDijkstra" folder, organized by optimization and type of graph and size.

Execution times of the algorithms and their average values can be viewed respectively in the "InfoTimeDijkstra" and "FinalAnalysisDijkstra" folders, organized by optimization and type of graph and size.

Results in graphical and tabular format can be viewed respectively in the "Plots" and "Tables" folders, organized by optimization and type of graph and size.

**NB:** Due to high execution time, the number of trials has been reduced by setting the corresponding variable "it"=5 in the makefile.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.