PARALLELIZING MAXIMUM SUM SUBSEQUENCE

Srinivas Rishindra Pothireddi

University at Buffalo The State University of New York

Table of Contents

Problem Definition

- Given a Sequence of numbers find a continuous subsequence of those numbers whose sum is maximum.
- This problem is only interesting only when there are negative numbers in the sequence.

Algorithm

- We first compute the parallel prefix sums of all the numbers in the sequence.
- S = {p0, p1, . . . , pn−1} of $X=\{x_0, x_1, \ldots, x_{n-1}\}$, where $p_i = x_0 \otimes \ldots \otimes x_i$.
- Next, compute the parallel postfix maximum of S.
- Let m_i denote the value of the postfix-max at position i, and let a_i be the associated index.
- Next, for each i, compute $b_i = m_i - p_i + x_i$ and the solution corresponds to the maximum of the b_i's, where u is the index of the position where the maximum of the b_i's is found and $v = a_u$.
- The maximum sum of any subsequence will be the maximum value of b and the subsequence starts from position u to position v.

Example

- Consider the input sequence X = {-3, 5, 2, -1, -4, 8, 10, -2}
- The parallel prefix sum of X is S = {-3, 2, 4, 3, -1, 7, 17, 15}

$m_0 = 17$	$a_0 = 6$	$b_0 = 17 - (-3) + (-3) = 17$
$m_1 = 17$	$a_1 = 6$	$b_1 = 17 - 2 + 5 = 20$
$m_2 = 17$	$a_2 = 6$	$b_2 = 17 - 4 + 2 = 15$
$m_3 = 17$	$a_3 = 6$	$b_3 = 17 - 3 + (-1) = 13$
$m_4 = 17$	$a_4 = 6$	$b_4 = 17 - (-1) + (-4) = 14$
$m_5 = 17$	$a_5 = 6$	$b_5 = 17 - 7 + 8 = 18$
$m_6 = 17$	$a_6 = 6$	$b_6 = 17 - 17 + 10 = 10$
$m_7 = 15$	$a_7 = 7$	$b_7 = 15 - 15 + (-2) = -2$

We have a maximum subsequence sum of $b_1 = 20$. This corresponds to $u = 1$ and $v = a_1 = 6$, or the subsequence {5, 2, −1, −4, 8, 10}.

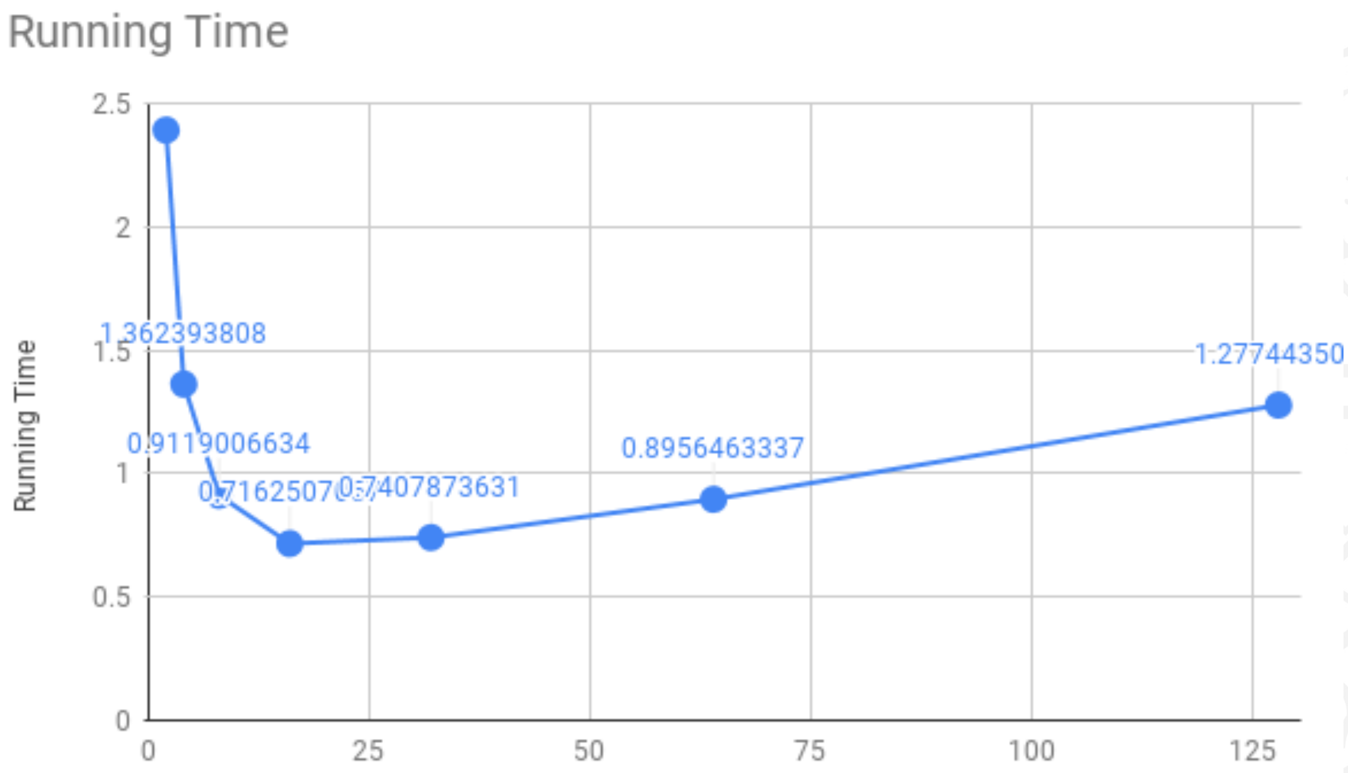
Amdahl's Law

- The maximum speedup achievable by an *n*-processor machine is given by $S_n \leq 1/[f + (1 - f)/n]$, where f is the fraction of operations in the computation that must be performed sequentially.
- So, for example, if five percent of the operations in a given computation must be performed sequentially, then the speedup can never be greater than 20, *regardless of how many processors are used.*
- Therefore, just a small number of sequential operations can significantly limit the speedup of an algorithm on a parallel machine.

1 Million

No of Processing Elements	Running Time
2	2.391577244
4	1.362393808
8	0.9119006634
16	0.7162507057
32	0.7407873631
64	0.8956463337
128	1.277443504

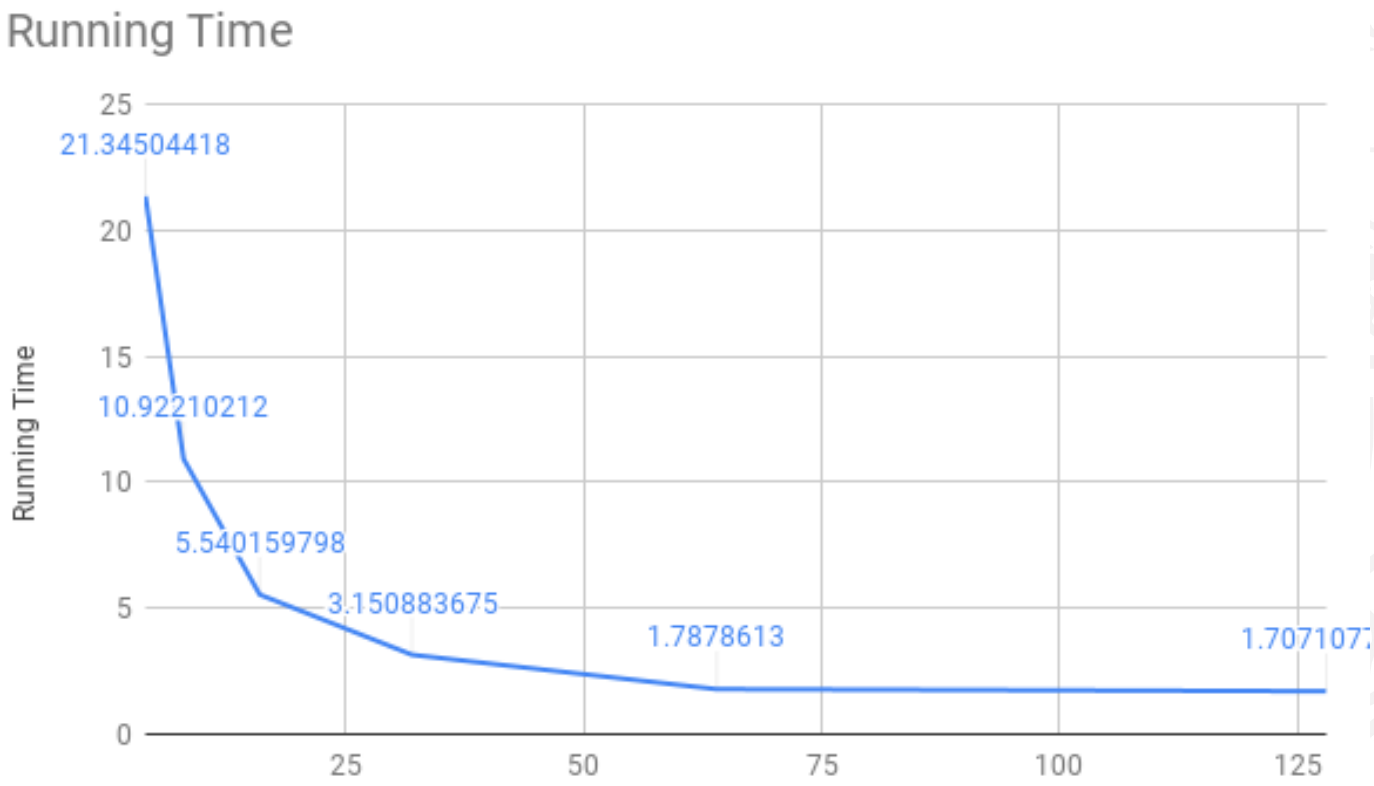
1 Million


Running Time
Running Time
2.5
2
1.5
1
0.5
0
0
25
50
75
100
125
1.362393808
0.9119006634
0.7407873631
0.8956463337
1.27744350

10 Million

No of Processing Elements	Running Time
2	21.34504418
4	10.92210212
8	5.540159798
16	3.150883675
32	1.7878613
64	1.707107735
128	1.308482885

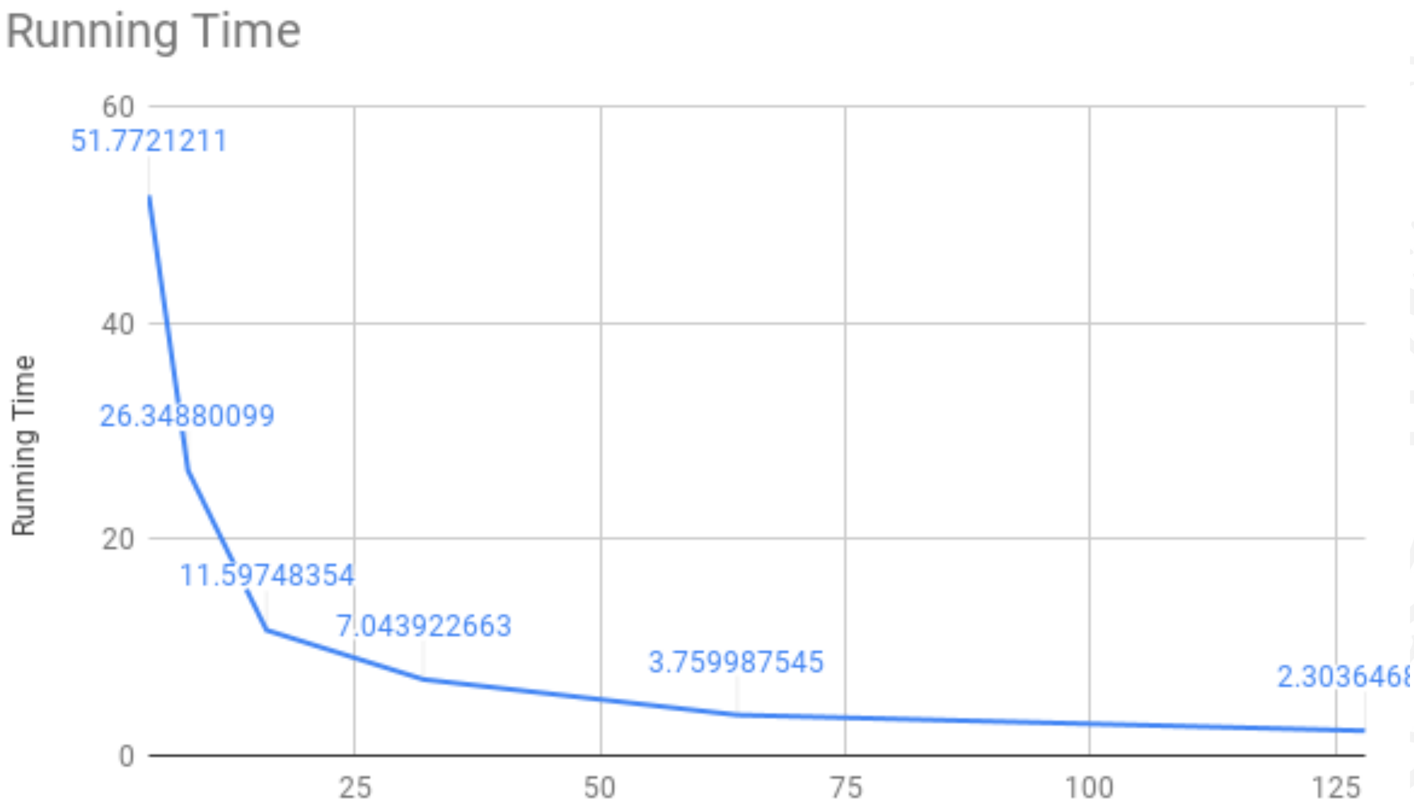
10 Million


Running Time
Running Time
25
20
15
10
5
0
25
50
75
100
125
21.34504418
10.92210212
5.540159798
3.150883675
1.7878613
1.707107

25 Million

No of Processing Elements	Running Time
2	51.7721211
4	26.34880099
8	11.59748354
16	7.043922663
32	3.759987545
64	2.303646898
128	1.823378897

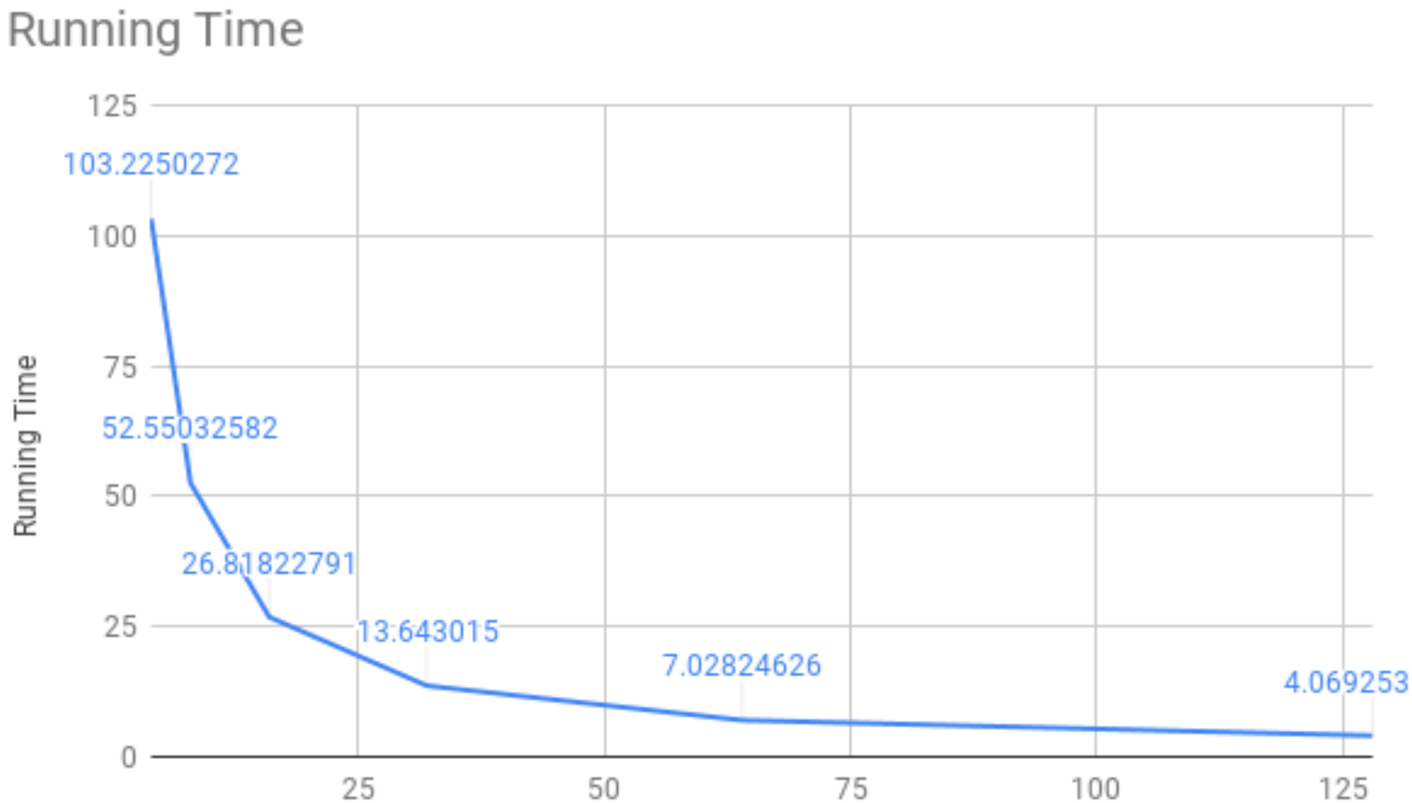
25 Million


Running Time
Running Time
60
40
20
0
25
50
75
100
125
51.7721211
26.34880099
11.59748354
7.043922663
3.759987545
2.303646

50 Million

No of Processing Elements	Running Time
2	103.2250272
4	52.55032582
8	26.81822791
16	13.643015
32	7.02824626
64	4.069253588
128	2.656966639

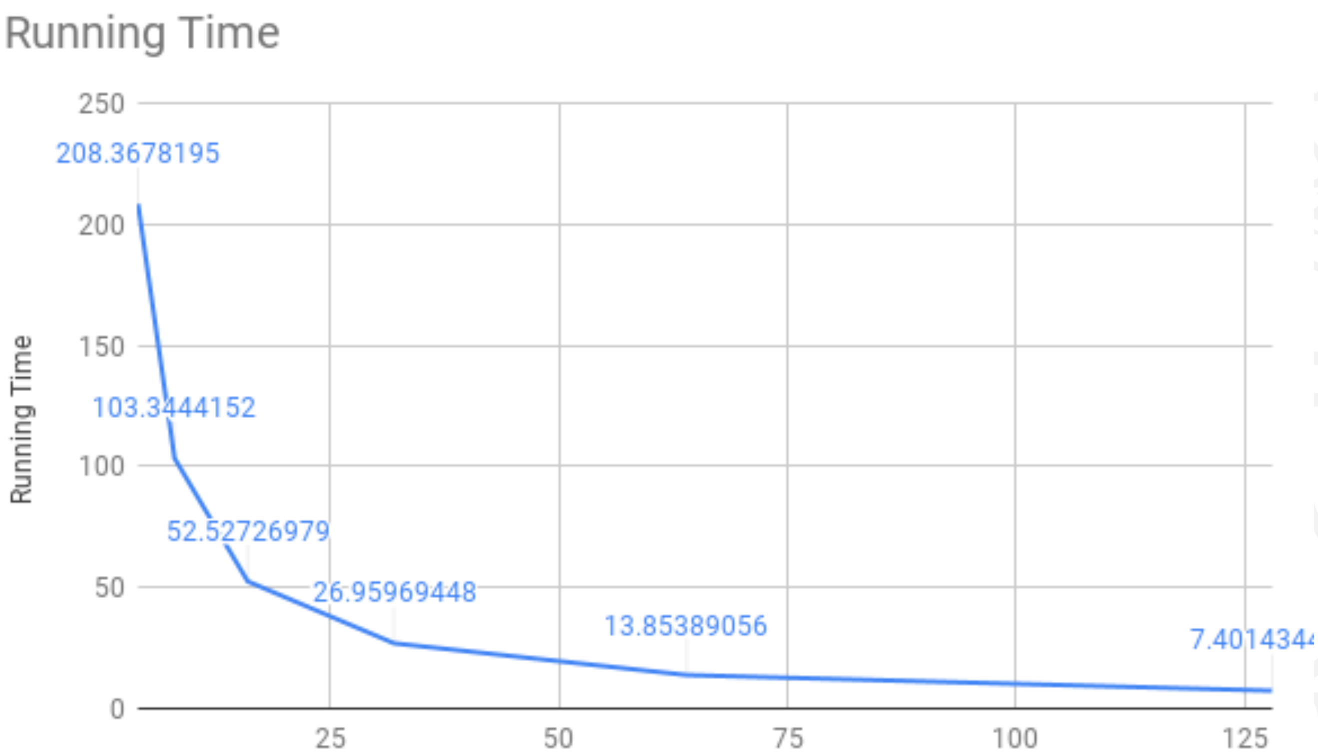
50 Million


Running Time
Running Time
125
100
75
50
25
0
25
50
75
100
125
103.2250272
52.55032582
26.81822791
13.643015
7.02824626
4.069253

100 Million

No of Processing Elements	Running Time
2	208.3678195
4	103.3444152
8	52.52726979
16	26.95969448
32	13.85389056
64	7.401434422
128	4.283970213

100 Million


Running Time
Running Time
250
200
150
100
50
0
25
50
75
100
125
208.3678195
103.3444152
52.52726979
26.95969448
13.85389056
7.4014344

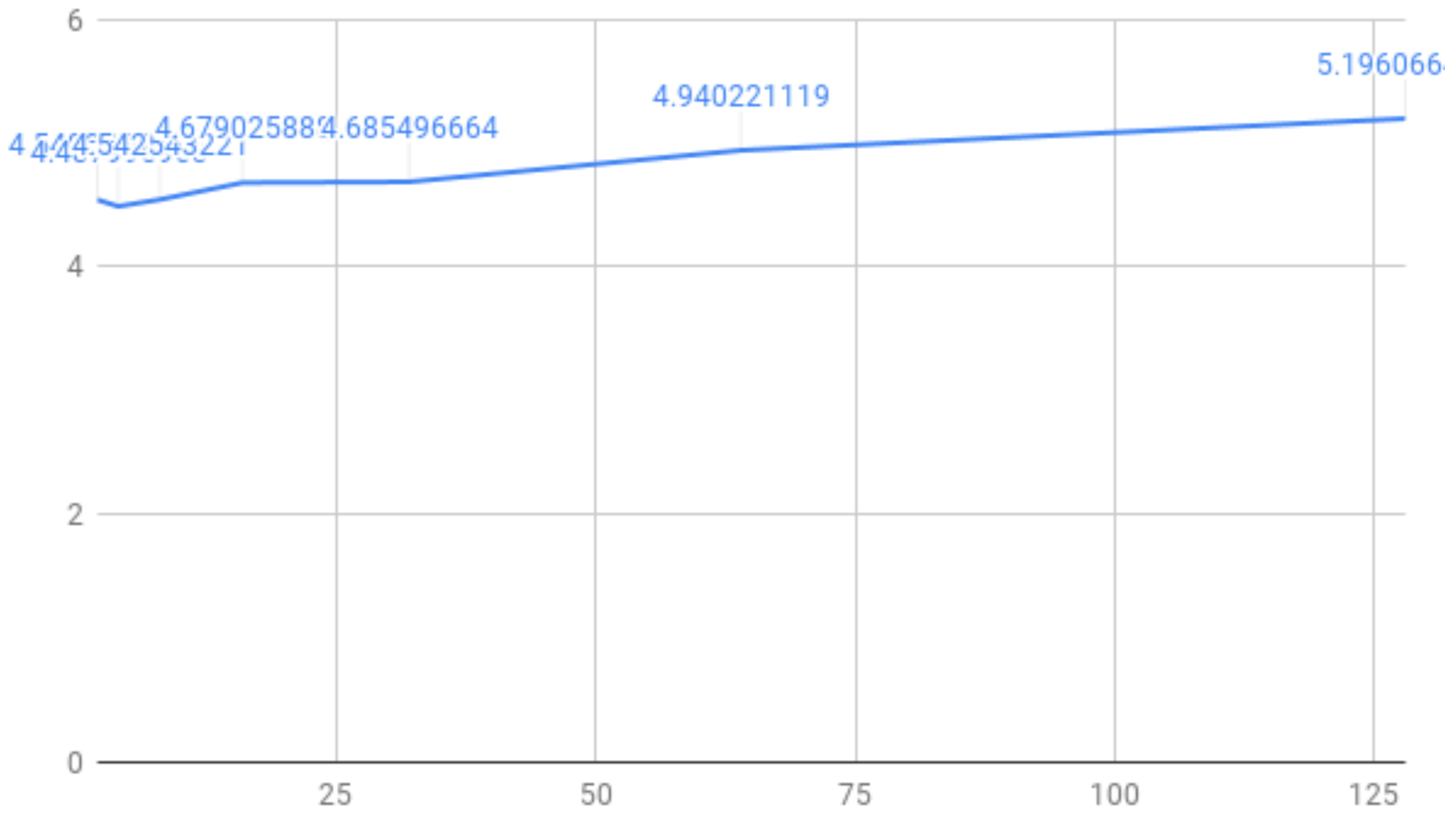
Reevaluating Amdahl's Law(1988)

- Amdahl's Law overlooks the fact that for many algorithms, the percentage of required sequential operations decreases as the size of the *problem* increases.
- Further, it is often the case that as one scales up a parallel machine, scientists often want to solve larger and larger problems, and not just the same problems more efficiently.
- That is, it is common enough to find that for a given machine, scientists will want to solve the largest problem that fits on that machine, and complain that the machine isn't just a bit bigger so that they could solve the larger problem they really want to consider.

1 Million

No of Processing Elements	Averages
2	4.540847921
4	4.487995958
8	4.542543221
16	4.679025888
32	4.685496664
64	4.940221119
128	5.196066475

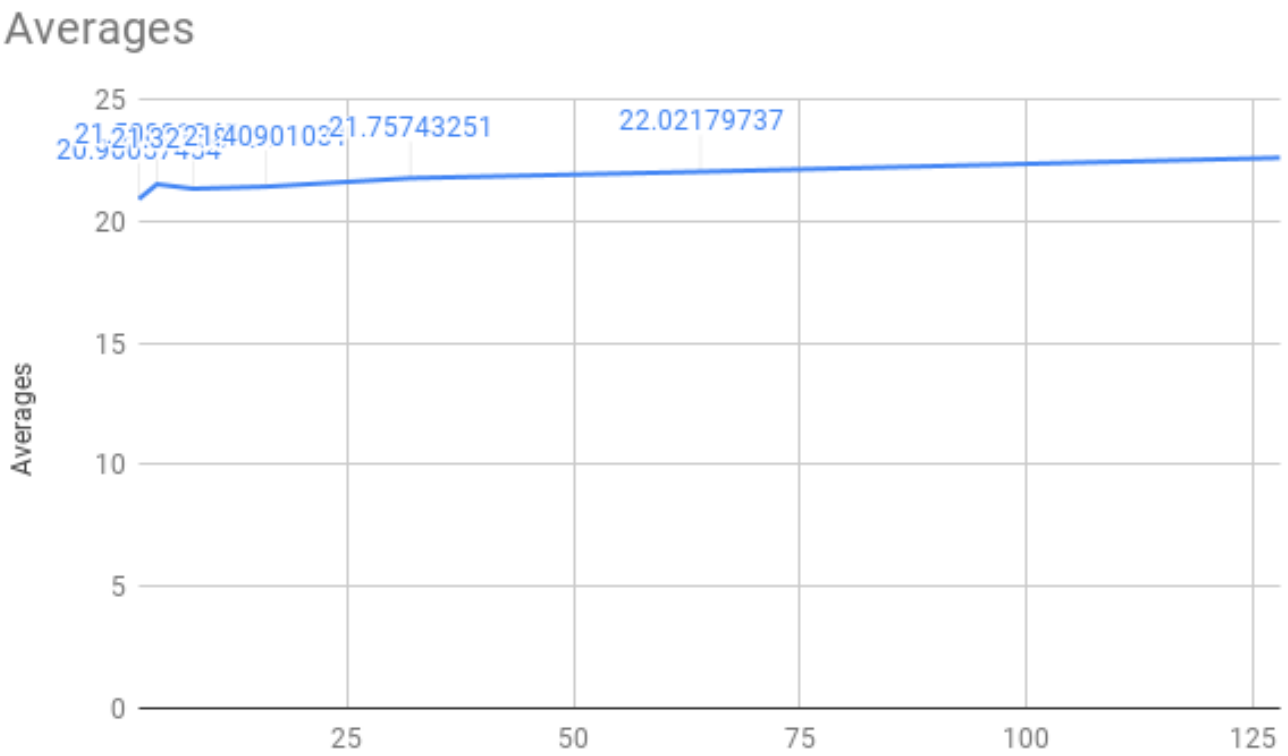
1 Million


6
4
2
0
25
50
75
100
125
4.679025881
4.685496664
4.940221119
5.196066

5 Million

No of Processing Elements	Averages
2	20.90057454
4	21.50881248
8	21.32558784
16	21.40901031
32	21.75743251
64	22.02179737
128	22.59391766

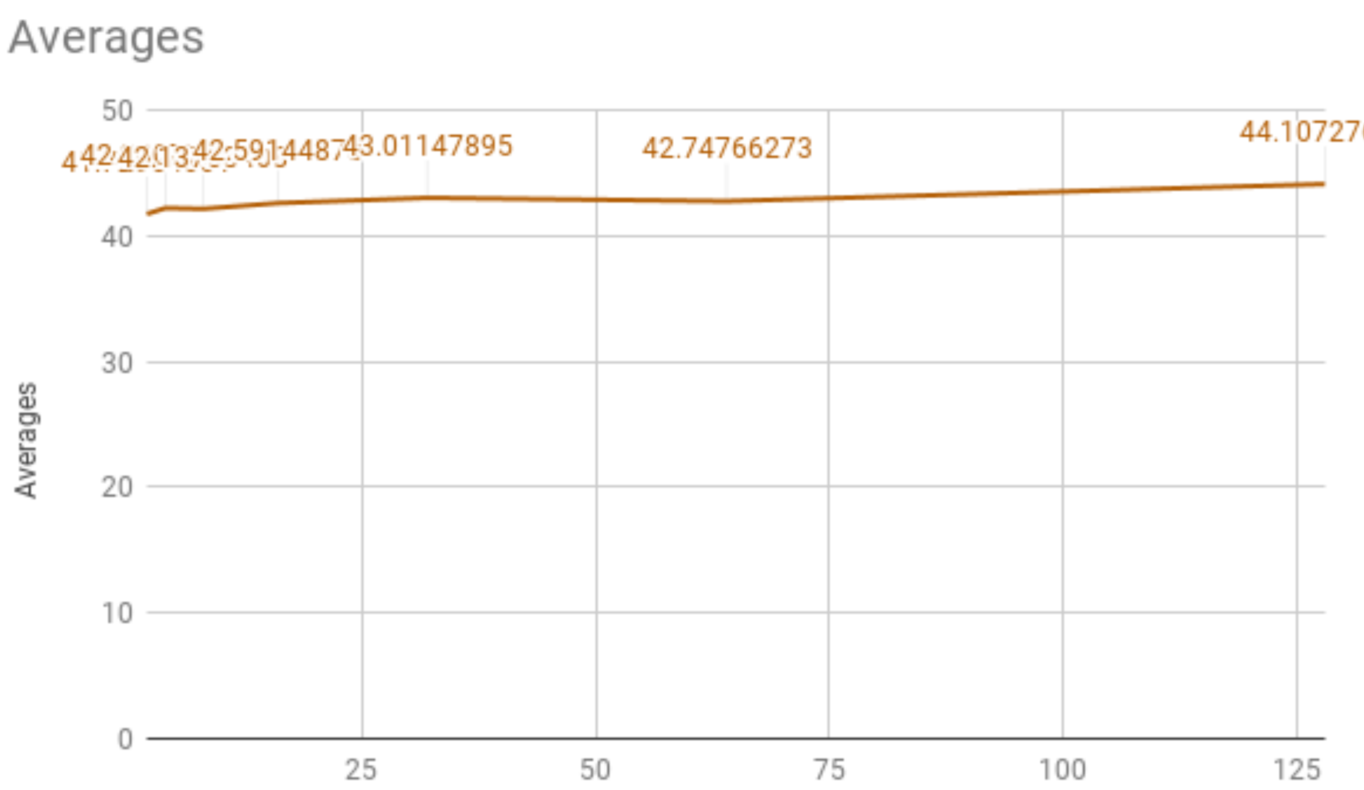
5 Million


Averages
25
20
15
10
5
0
25
50
75
100
125
Averages
21.75743251
22.02179737

10 Million

No of Processing Elements	Averages
2	41.72954001
4	42.20077882
8	42.13703408
16	42.59144878
32	43.01147895
64	42.74766273
128	44.10727668

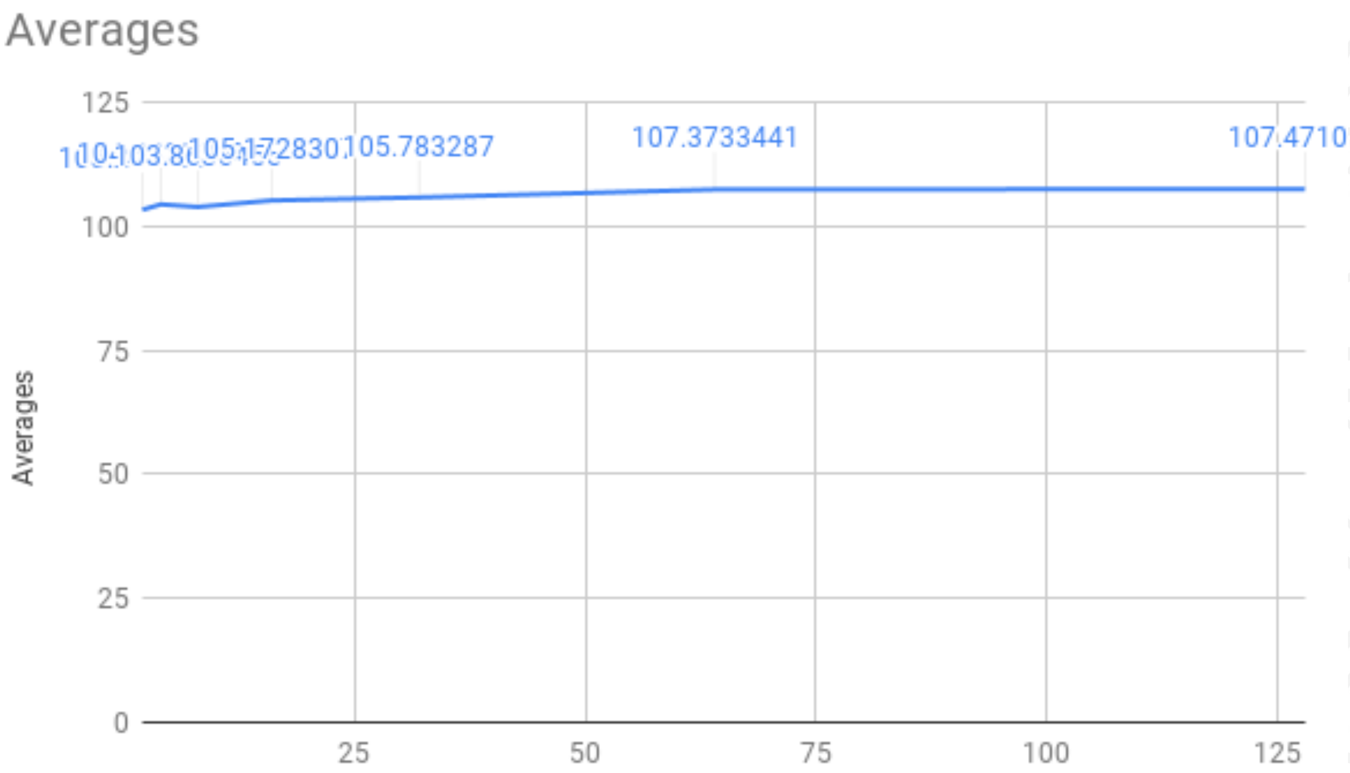
10 Million


Averages
43.01147895
42.74766273
44.10727
50
40
30
20
10
0
Averages
25
50
75
100
125

25 Million

No of Processing Elements	Averages
2	103.2897964
4	104.3829672
8	103.8636455
16	105.1728307
32	105.783287
64	107.3733441
128	107.471099

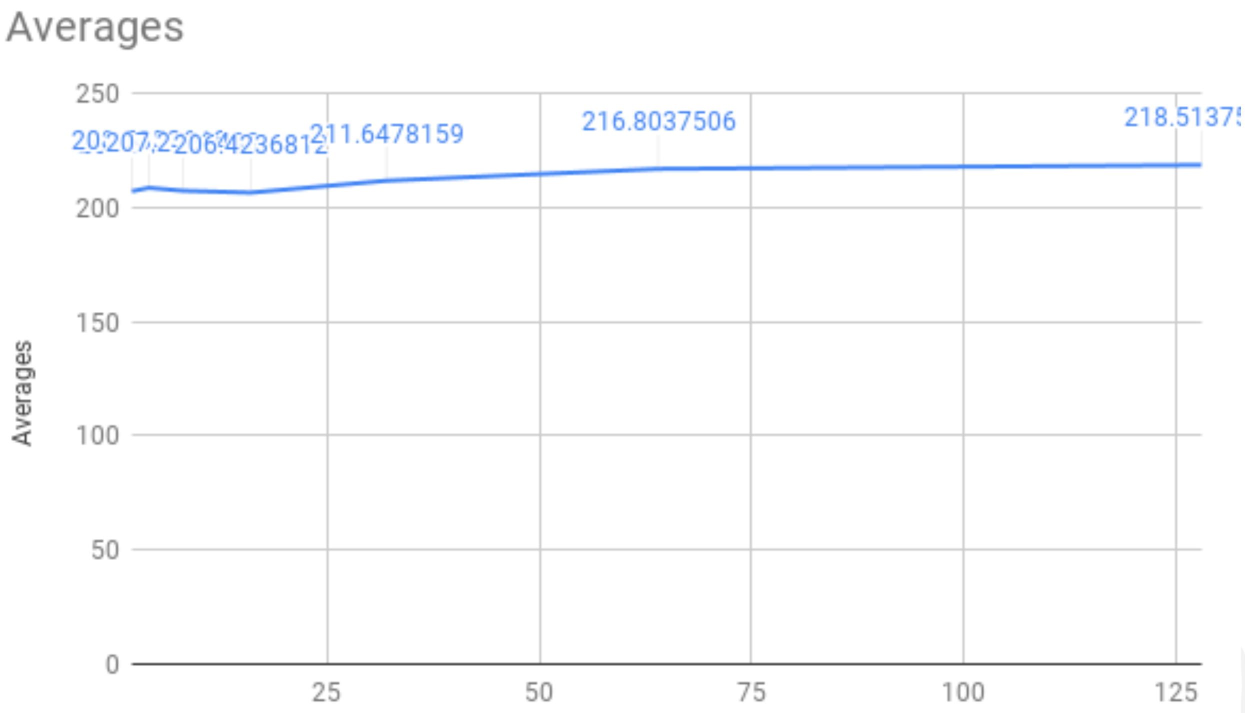
25 Million


Averages
105.783287
107.3733441
107.4710
Averages
125
100
75
50
25
0
25
50
75
100
125

50 Million

No of Processing Elements	Averages
2	207.0414
4	208.5352646
8	207.2463433
16	206.4236812
32	211.6478159
64	216.8037506
128	218.5137507

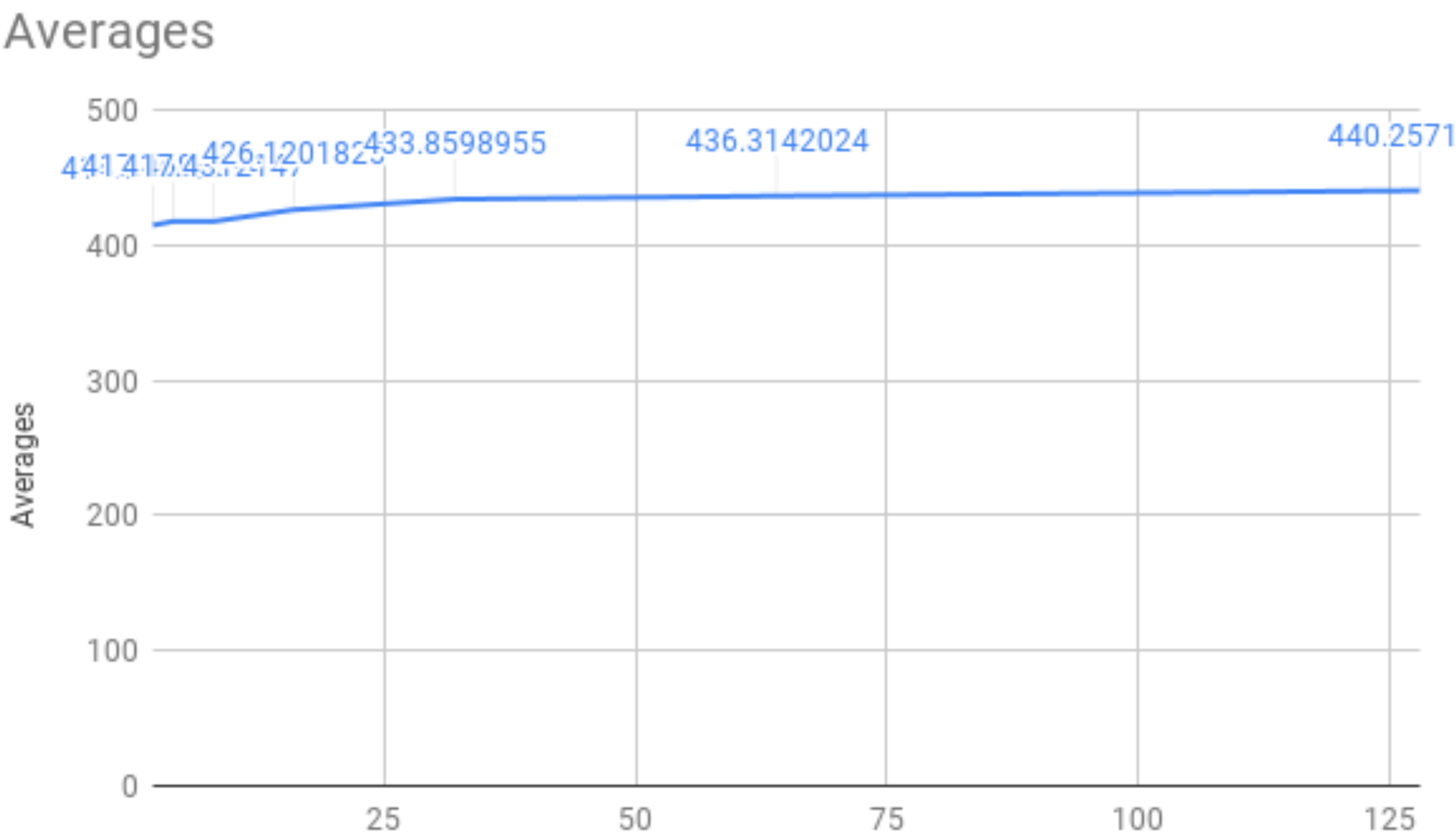
50 Million


Averages
Averages
250
200
150
100
50
0
25
50
75
100
125
211.6478159
216.8037506
218.51375
206.4236812

100 Million

No of Processing Elements	Averages
2	414.6446681
4	417.4684594
8	417.4312147
16	426.1201825
32	433.8598955
64	436.3142024
128	440.2571783

100 Million



Averages
500
400
300
200
100
0
Averages
25
50
75
100
125
433.8598955
436.3142024

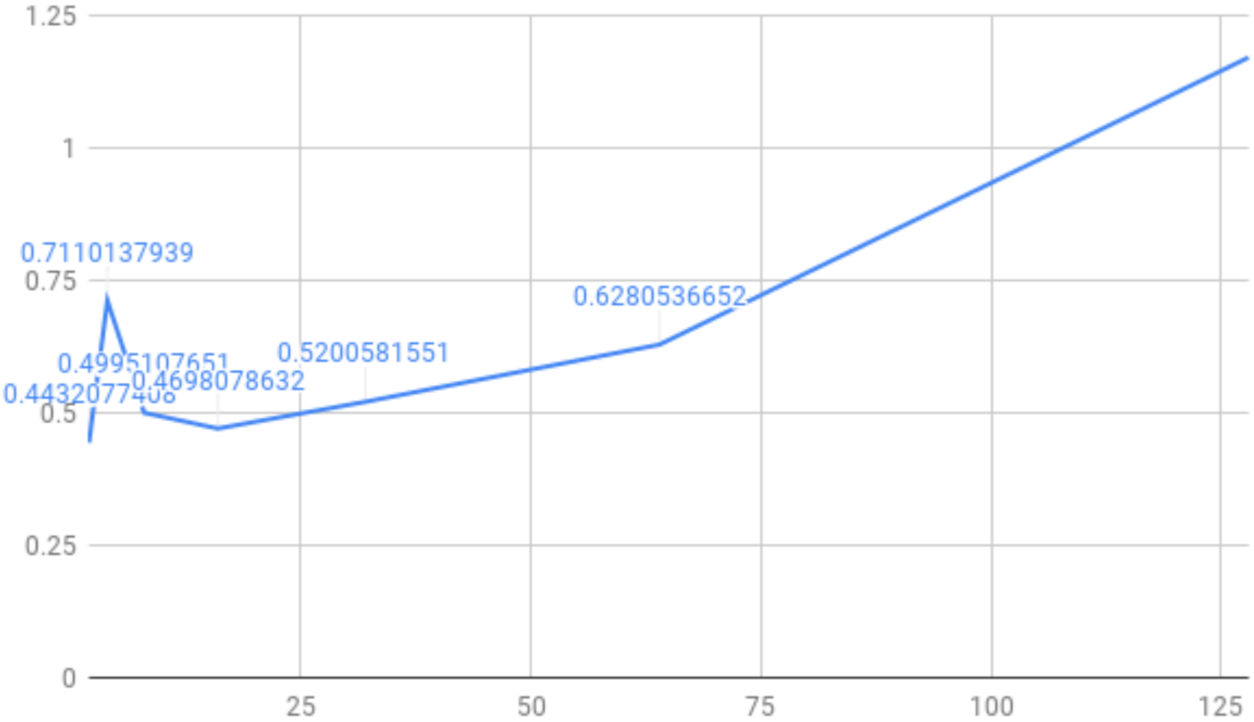
References

- Algorithms Sequential and Parallel, A Unified Approach
  ~Russ Miller, Laurence Boxer
- http://www.johngustafson.net/pubs/pub13/amdahl.html
- Mpi4py official documentation.

Appendix A : Gustafson's Small Data

2	0.4432077408
4	0.7110137939
8	0.4995107651
16	0.4698078632
32	0.5200581551
64	0.6280536652
128	1.169249296

Appendix A : Gustafson's Small Data



Appendix B : Amdahl's Small Data

2	0.278011322
4	0.4786112309
8	0.4027721882
16	0.5404441357
32	0.763256073
64	2.063477755
128	1.271056652

Appendix B : Amdahl's Small Data