Parallelizing Maximum Sum Subsequence

SAKET ADUSUMILLI
SPRING 2014
CSE 633
Dept. Computer science and engineering

Table of Contents

Maximum Sum Subsequence Problem

Determining a subsequence of a data set that sums to the maximum value with respect to any subsequence of the data set.

- Example:
- X = {−3, 5, 2, −1, −4, 8, 10, −2}
- The maximum sum subsequence= {5,2,-1,-4,8,10}

Sequential Algorithm

Global_Max ← x_0
u ← 0 {Start index of global max subsequence}
v ← 0 {End index of global max subsequence}
Current_Max ← x_0
q ← 0 {Initialize index of current subsequence}
For *i* = 1 to *n* − 1, do {Traverse list}
    If *Current_Max* ≥ 0 Then
        Current_Max ← *Current_Max* + x_i
Else
    Current_Max ← x_i
    q ← *i* {Reset index of current subsequence}
End Else
If *Current_Max* > *Global_Max* Then
    Global_Max ← *Current_Max*
    u ← *q*
    v ← *i*
End If
End For

Complexity: $\Theta(n)$

Parallelization

Approach:

Linear array Implementation

Using parallel prefix.

We first compute the parallel prefix sums $S = \{p_0, p_1, \ldots, p_{n-1}\}$ 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 $v = a_u$.

Example

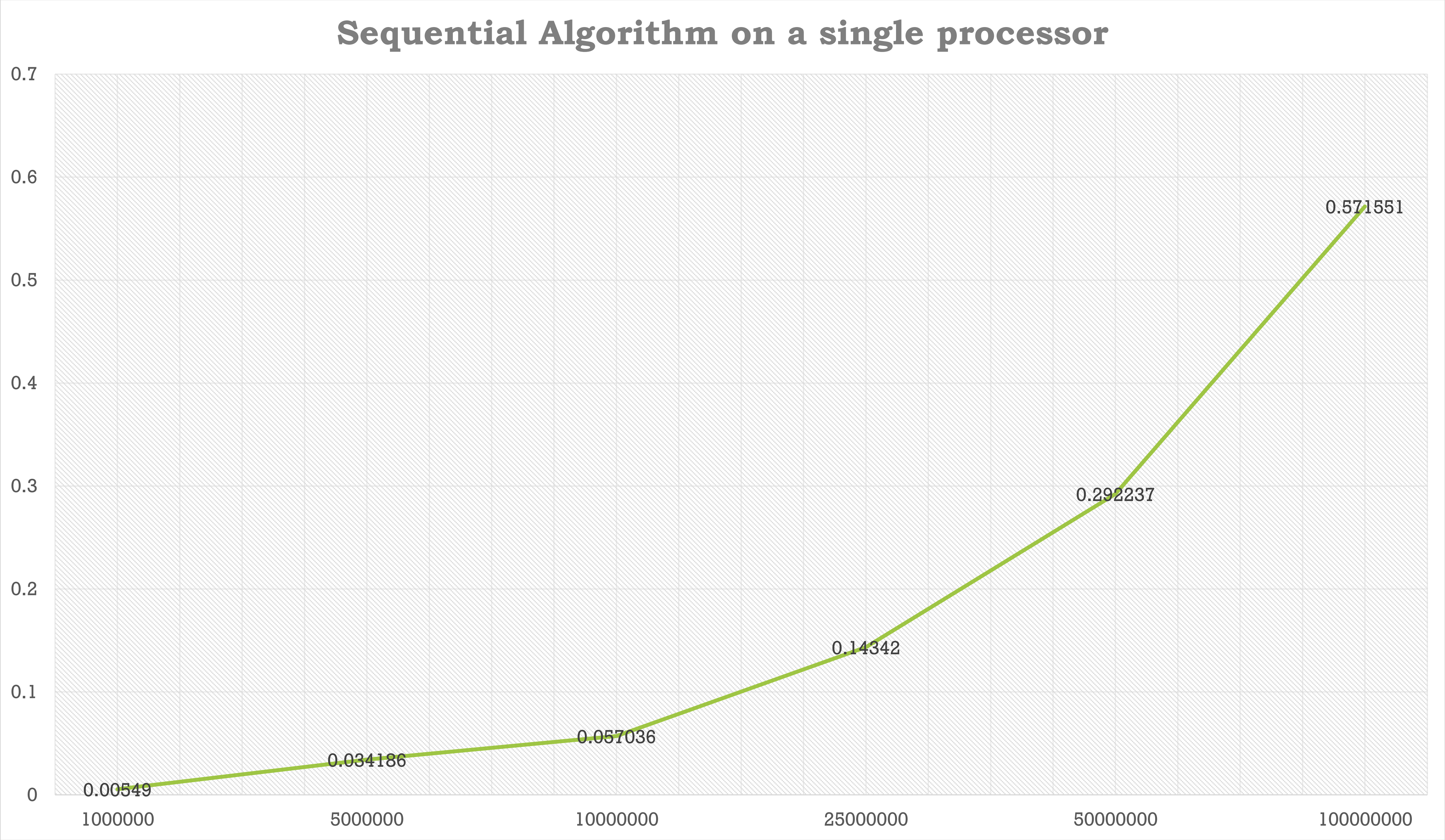
Consider the input sequence
$X = \{-3, 5, 2, -1, -4, 8, 10, -2\}$. The parallel prefix sum of X is $S = \{-3, 2, 4, 3, -1, 7, 17, 15\}$.

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

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

Running time(in seconds) of Sequential Algorithm on a single processor

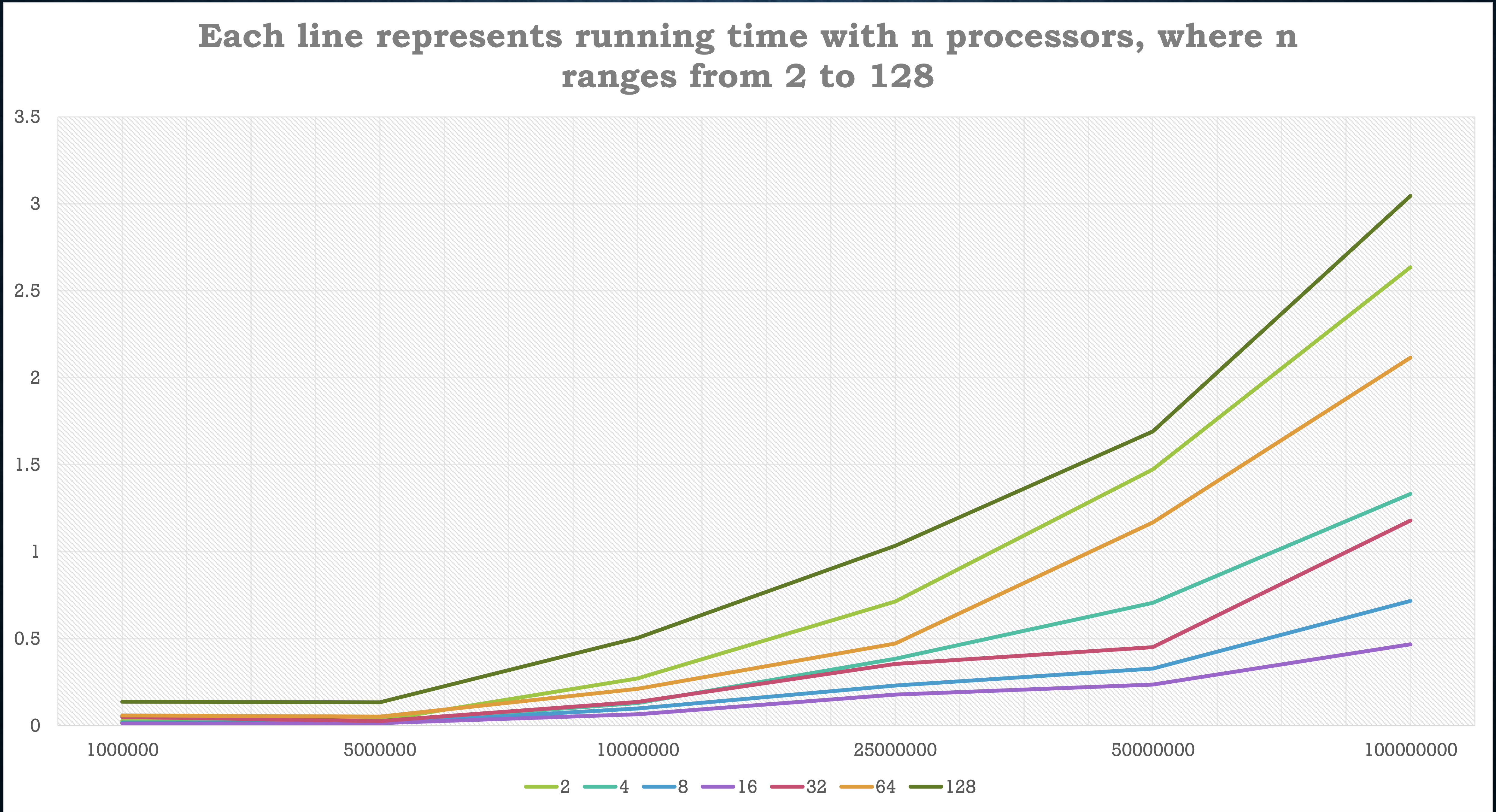
Data Items	Time Taken
1,000,000	0.00549
5,000,000	0.034186
10,000,000	0.057036
25,000,000	0.14342
50,000,000	0.292237
100,000,000	0.571551

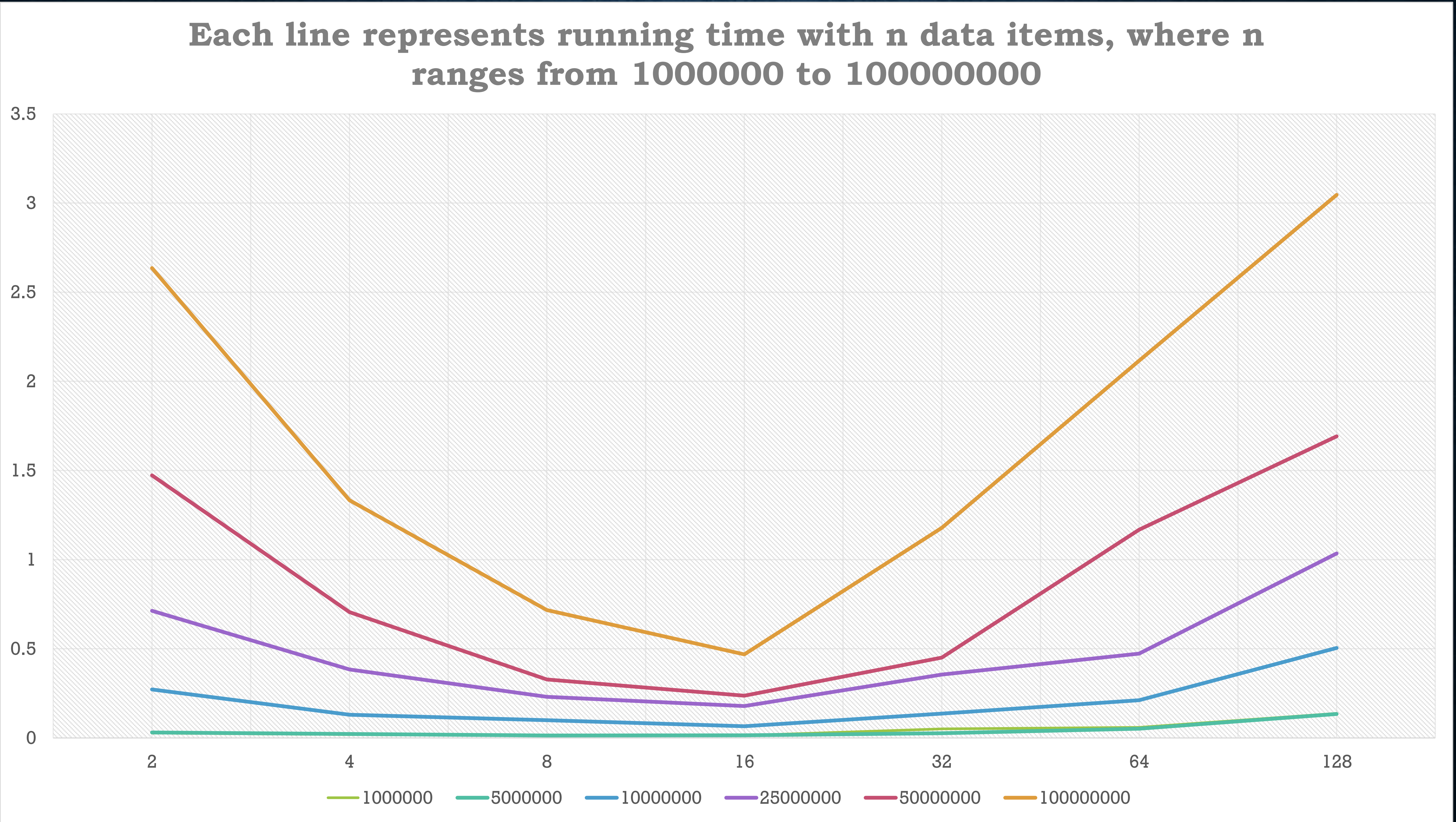


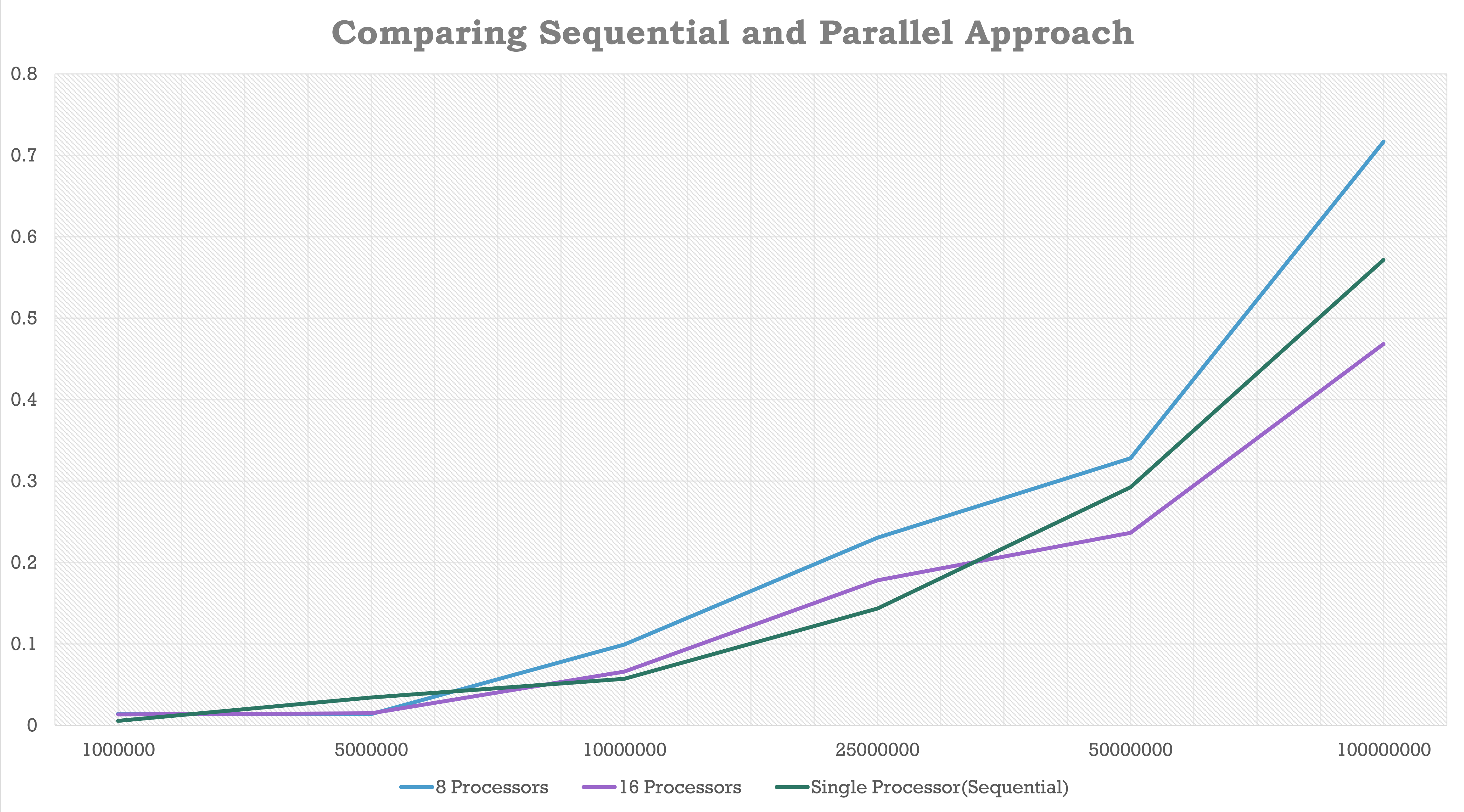
Sequential Algorithm on a single processor
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00549
0.034186
0.057036
0.14342
0.292237
0.571551
1000000
5000000
10000000
25000000
50000000
100000000

Running time(in seconds) of Parallel Prefix Approach

Number of Processors / Data Items	2	4	8	16	32	64	128
1,000,000	0.027357	0.018844	0.014121	0.013244	0.050397	0.059169	0.137612
5,000,000	0.030936	0.021605	0.013785	0.014788	0.026265	0.052187	0.134169
10,000,000	0.271881	0.129775	0.099201	0.065835	0.135805	0.211822	0.504434
25,000,000	0.71324	0.384105	0.23034	0.17815	0.354905	0.472544	1.034197
50,000,000	1.472292	0.705677	0.328034	0.236339	0.45073	1.16811	1.69151
100,000,000	2.634892	1.332371	0.716712	0.468214	1.17876	2.115506	3.14566


Each line represents running time with n processors, where n ranges from 2 to 128
3.5
3
2.5
2
1.5
1
0.5
0
1000000
5000000
10000000
25000000
50000000
100000000
2
4
8
16
32
64
128


Each line represents running time with n data items, where n ranges from 1000000 to 100000000
3.5
3
2.5
2
1.5
1
0.5
0
2
4
8
16
32
64
128
1000000
5000000
10000000
25000000
50000000
100000000


Comparing Sequential and Parallel Approach
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1000000
5000000
10000000
25000000
50000000
100000000
8 Processors
16 Processors
Single Processor(Sequential)

Conclusions

- Increasing the processors is not going to reduce the running time. In this problem there is no use in increasing the number of processors over 16.
- Communication time between the processors will take over the processing time within the processors.
- Efficient Parallel Algorithm is not possible with master worker approach.

References

Algorithms Sequential and Parallel, A Unified Approach ~Russ Miller, Laurence Boxer

Thank you!