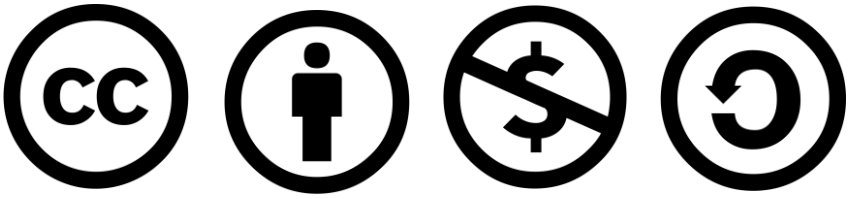


Lo scopo di questo studio è quello di esaminare le performance della parallelizzazione dell'algoritmo di ordinamento **"counting sort",** il quale è semplice da implementare senza il processo di parallelizzazione. Per raggiungere l'obiettivo, abbiamo utilizzato le direttive OpenMP.

```
$ cat /proc/cpuinfo
```

```
processors      : 16
vendor_id       : AuthenticAMD
cpu family      : 23
model           : 113
model name      : AMD Ryzen 7 3700X 8-Core Processor
stepping        : 0
microcode       : 0x8701021
cpu MHz         : 3599.998
cache size      : 512 KB
physical id     : 30
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 30
initial apicid  : 30
fpu             : yes
fpu_exception   : yes
cpuid level     : 16
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext
fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl tsc_reliable
nonstop_tsc cpuid extd_apicid pni pclmulqdq ssse3 fma cx16 sse4_1
sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand hypervisor
lahf_lm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw
topoext ssbd ibpb vmmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx
smap clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves clzero
wbnoinvd arat umip rdpid overflow_recov succor
bugs            : fxsave_leak sysret_ss_attrs spectre_v1 spectre_v2
```

```
spec_store_bypass
bogomips        : 7199.99
TLB size        : 3072 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 45 bits physical, 48 bits virtual
```

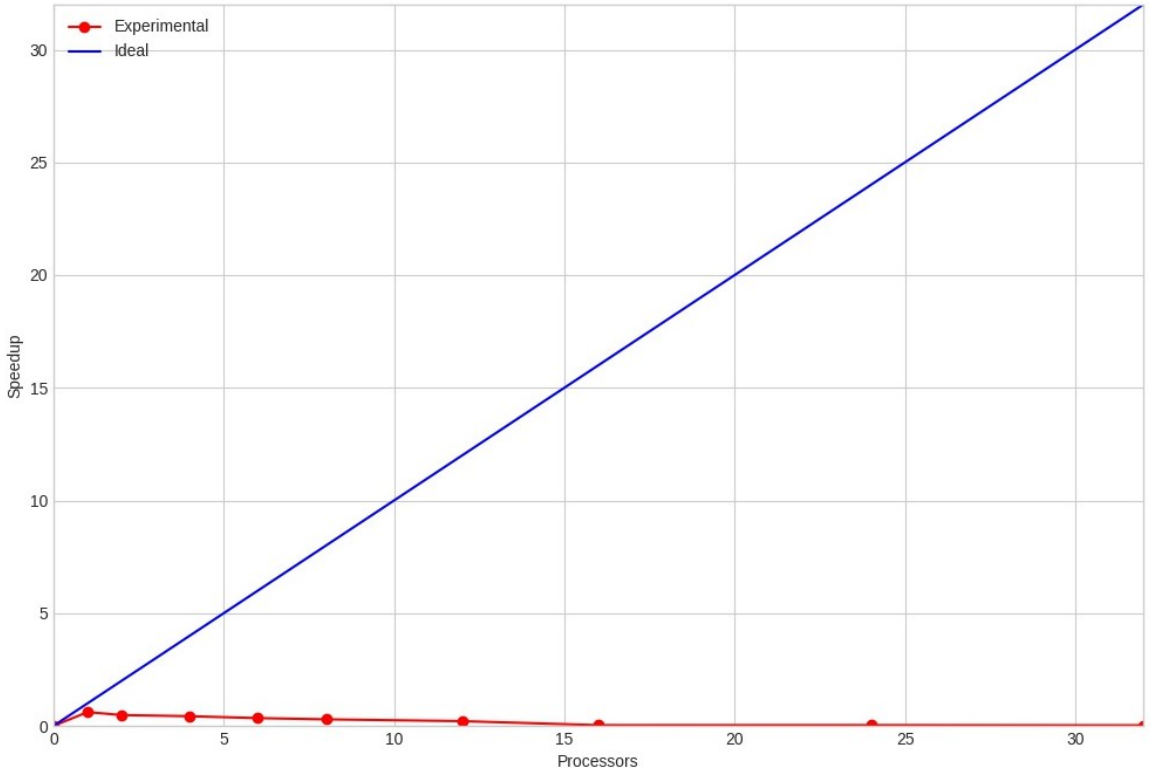
$ cat /proc/meminfo

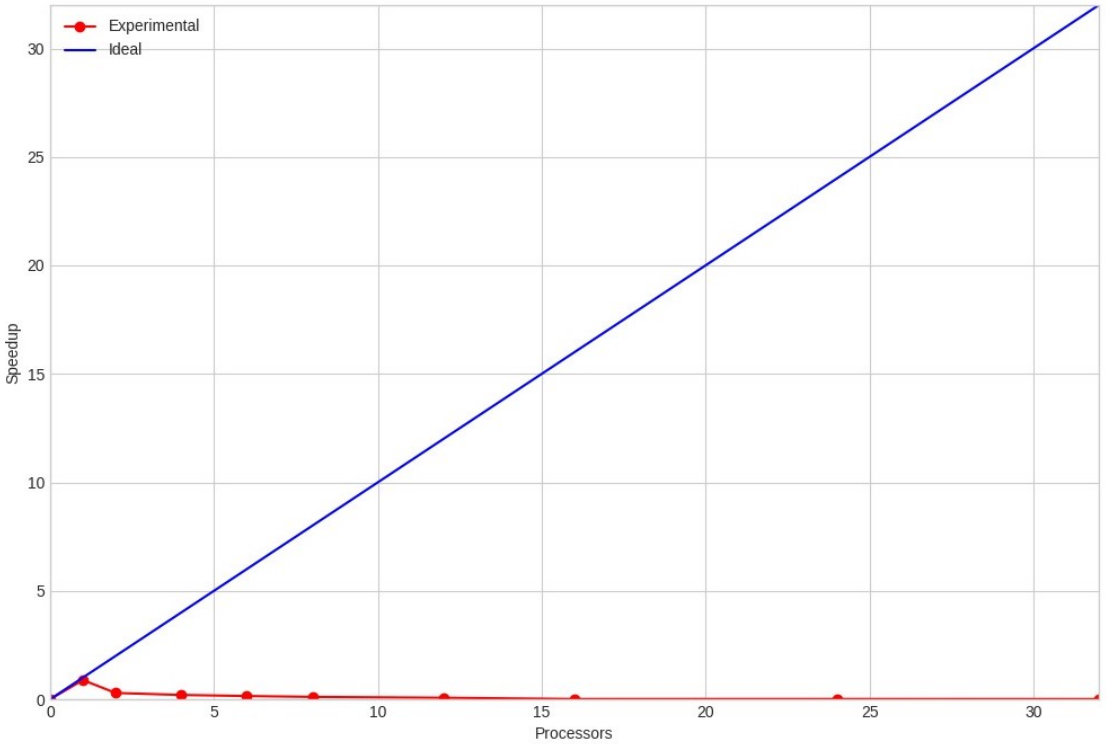
```
MemTotal:       12356456 kB
MemFree:         9120332 kB
MemAvailable:   10299128 kB
Buffers:           50220 kB
Cached:          1366740 kB
SwapCached:            0 kB
Active:           710156 kB
Inactive:        1780916 kB
Active(anon):       2292 kB
Inactive(anon):  1119192 kB
Active(file):     707864 kB
Inactive(file):   661724 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:        945368 kB
SwapFree:         945368 kB
Dirty:                 4 kB
Writeback:             0 kB
AnonPages:       1074220 kB
Mapped:           477688 kB
Shmem:             47364 kB
KReclaimable:      94364 kB
Slab:             203680 kB
SReclaimable:      94364 kB
SUnreclaim:       109316 kB
KernelStack:       14176 kB
PageTables:        21292 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     7123596 kB
Committed_AS:    4808032 kB
VmallocTotal:   34359738367 kB
```

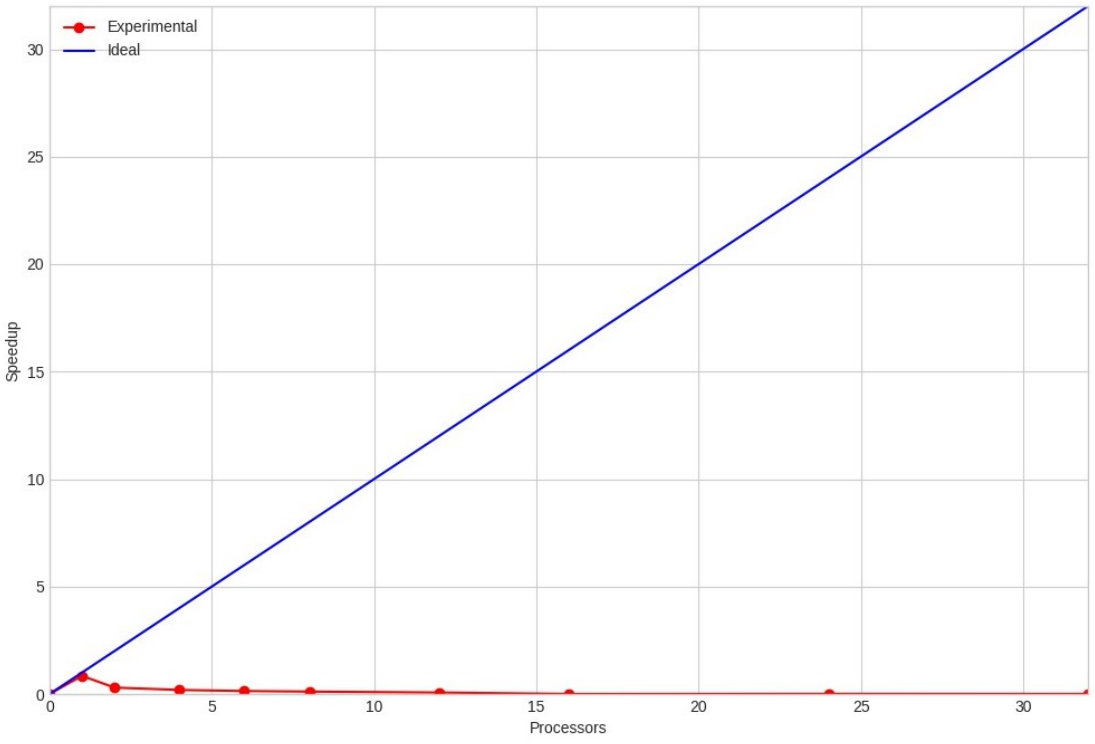
```
VmallocUsed:       62152 kB
VmallocChunk:          0 kB
Percpu:            91136 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:      296768 kB
DirectMap2M:     6125568 kB
DirectMap1G:     7340032 kB
```

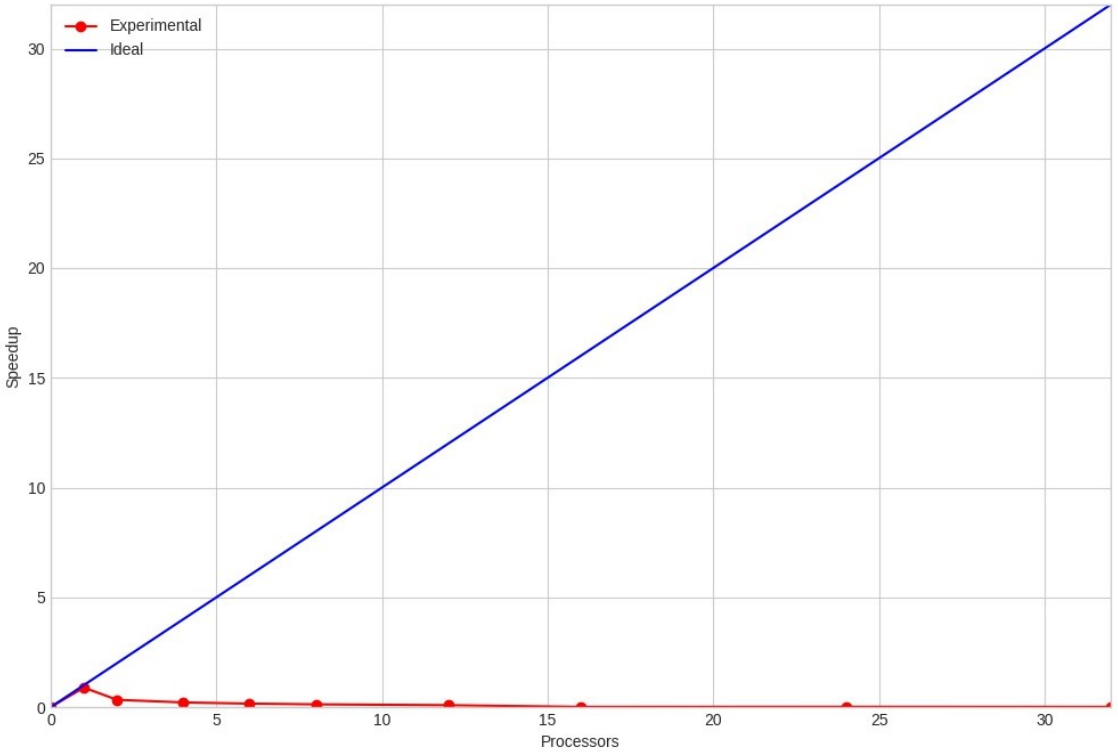
10’000, 50’000, 100’000, 500’000, 1'000’000

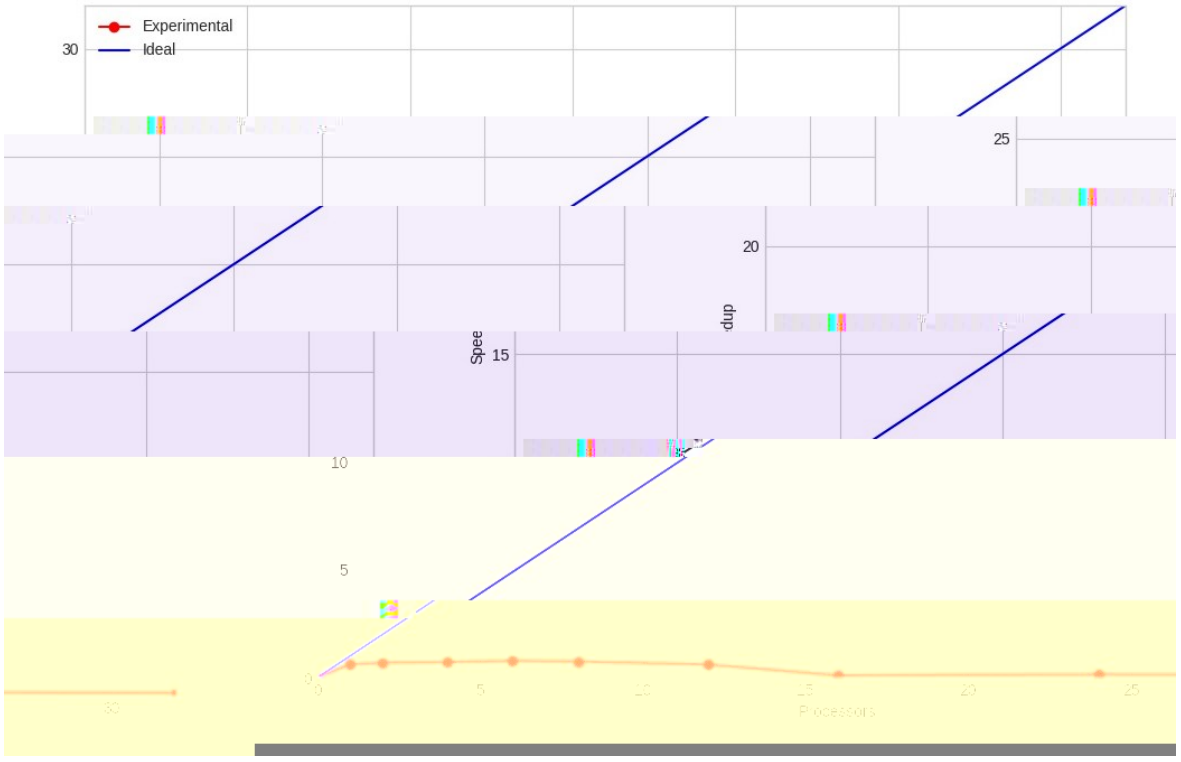
1, 2, 4, 6, 8, 12, 16, 24, 32

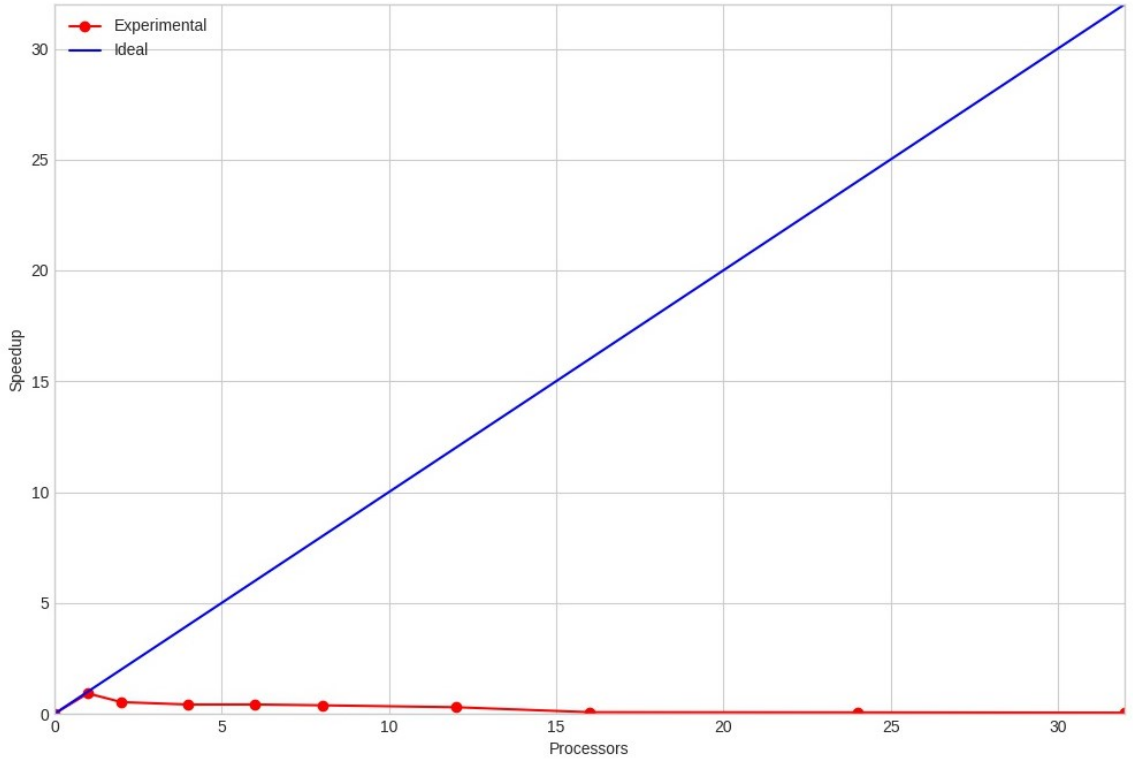

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

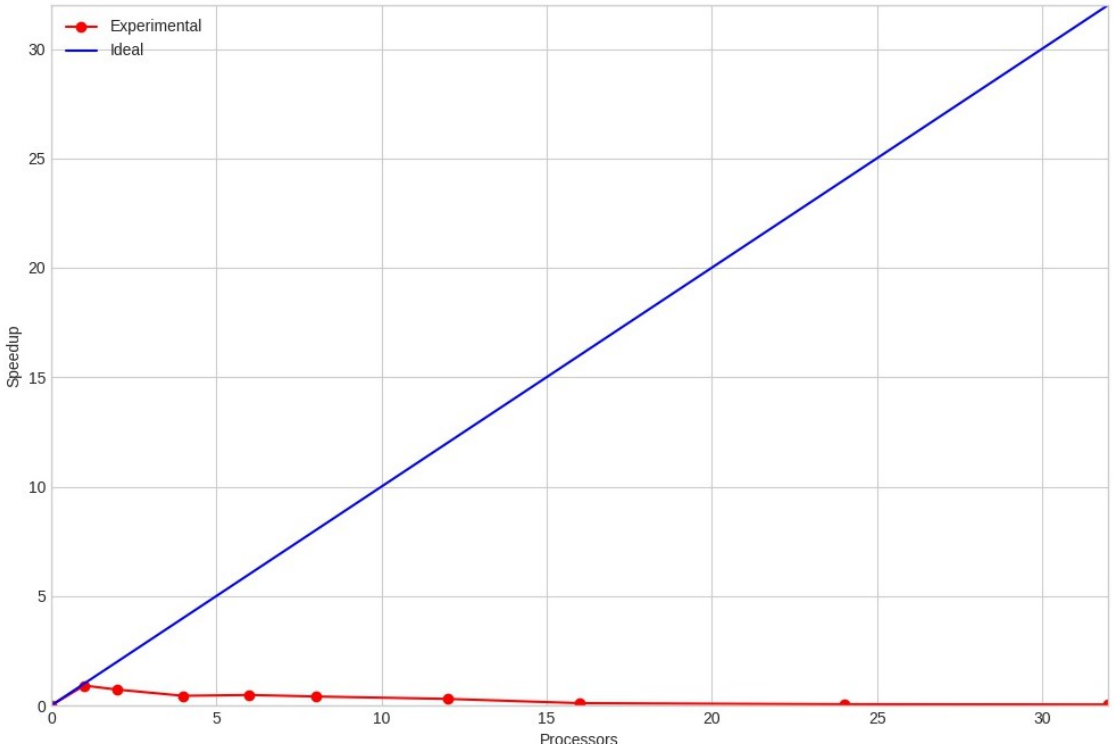

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

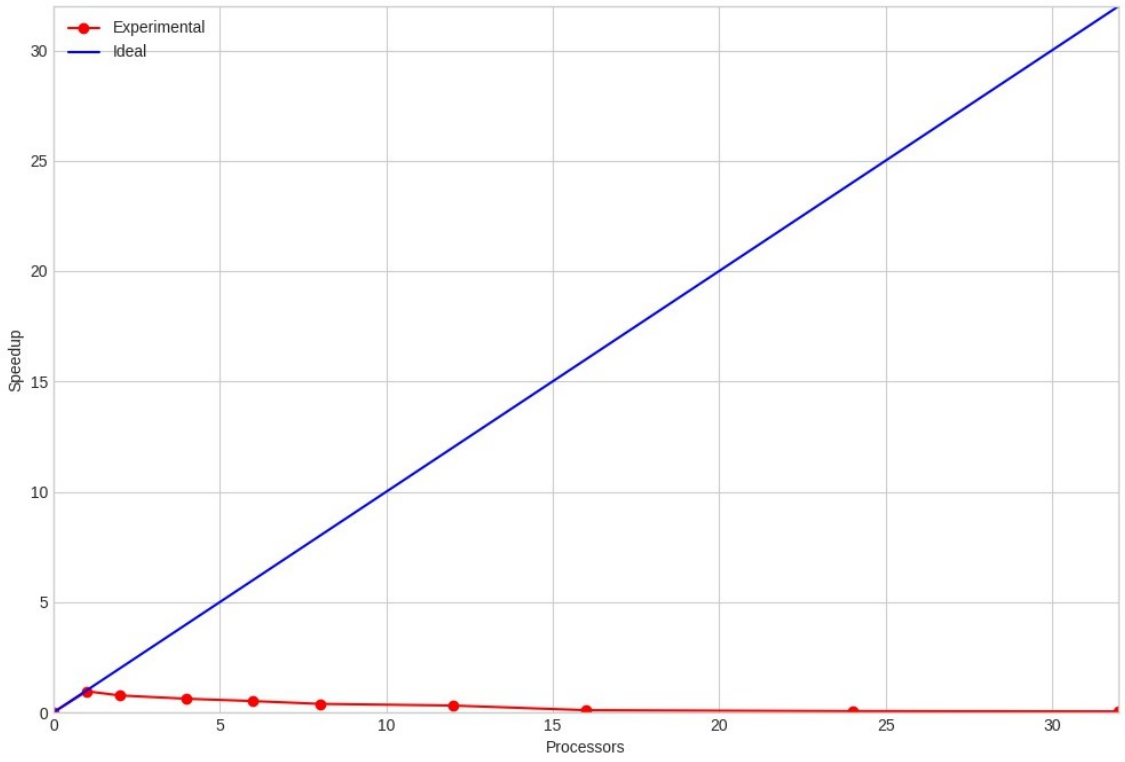

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

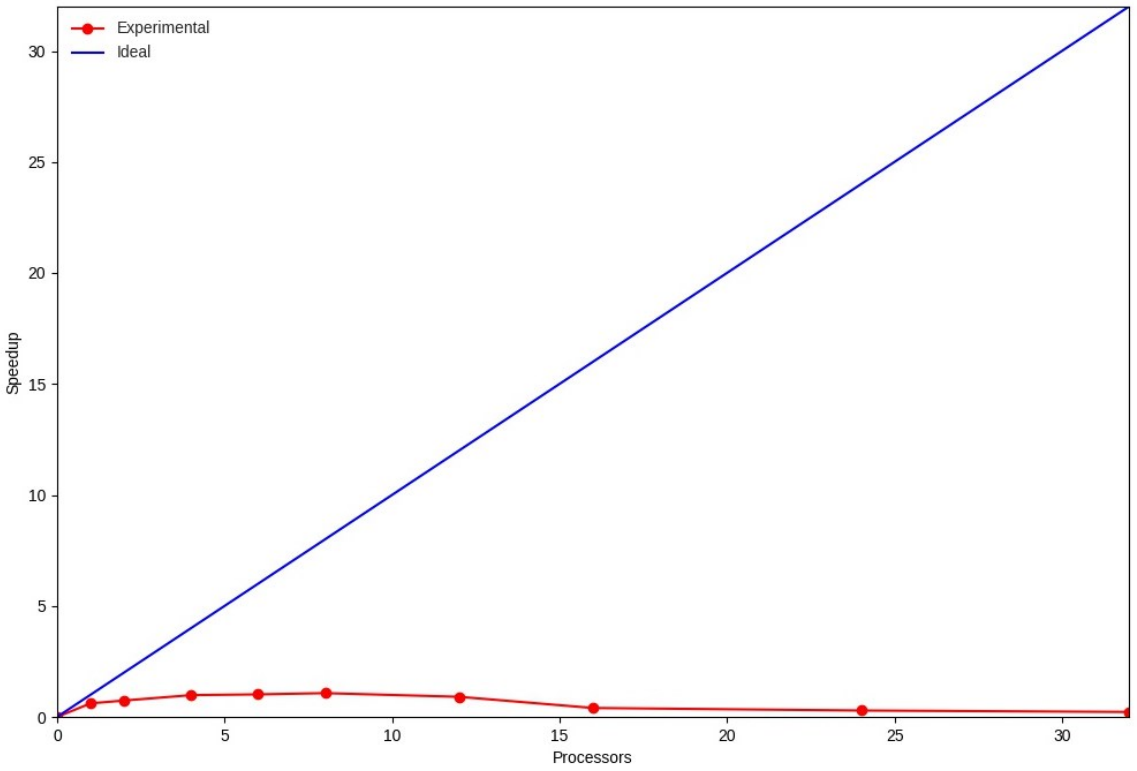

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

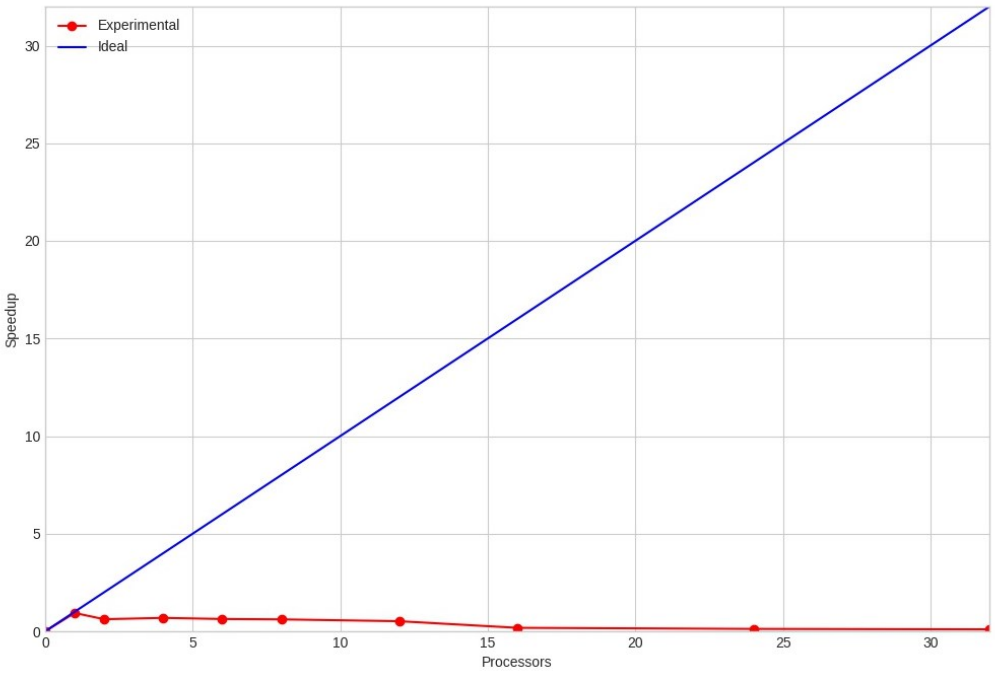

Experimental
Ideal
30
25
20
15
10
5
0
Spee
dup
0
5
10
15
20
25
30
Processors

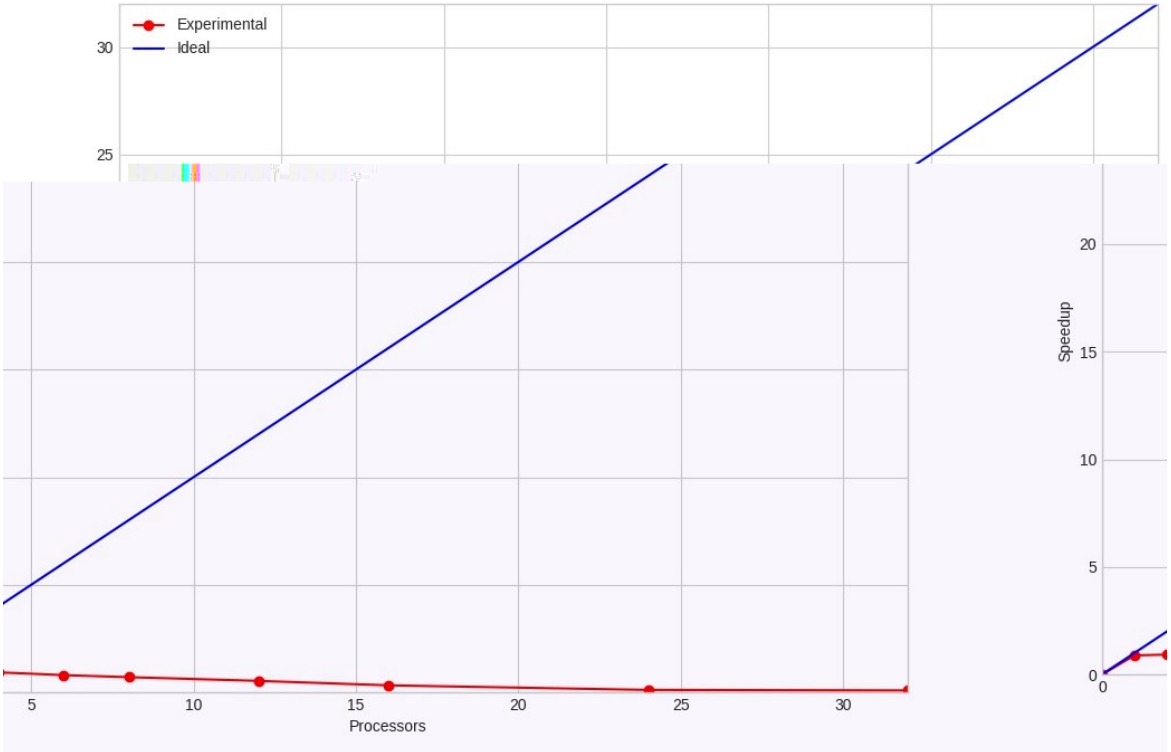

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

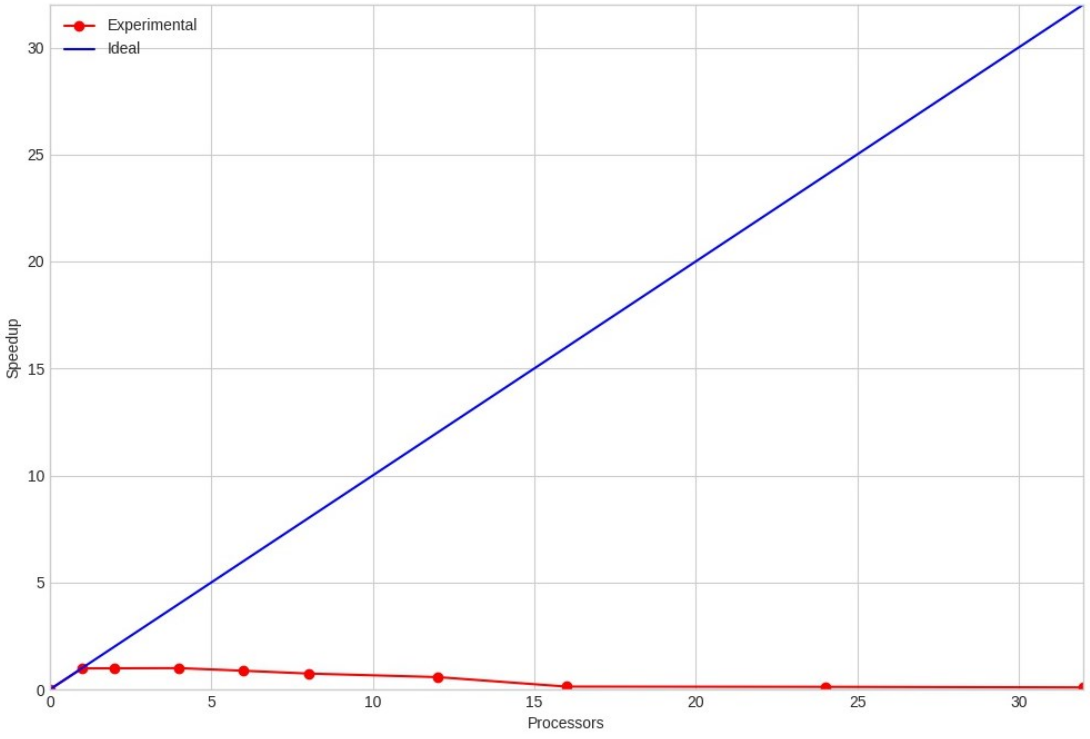

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

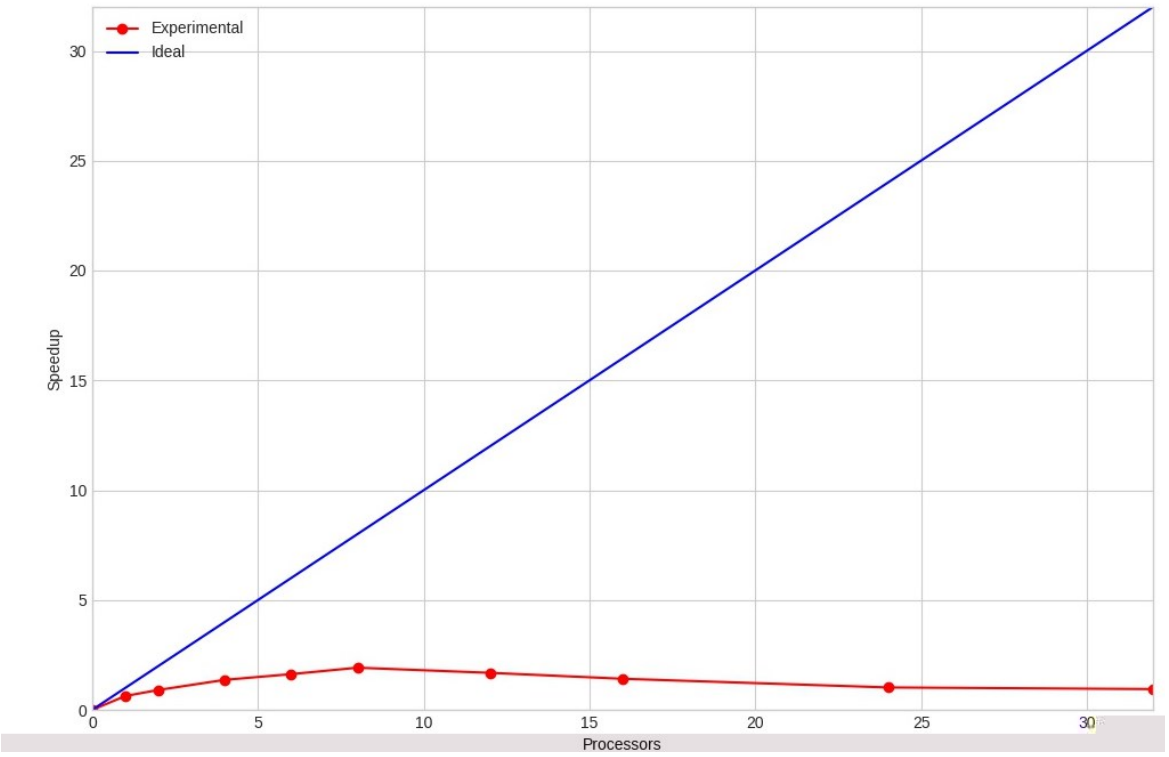

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

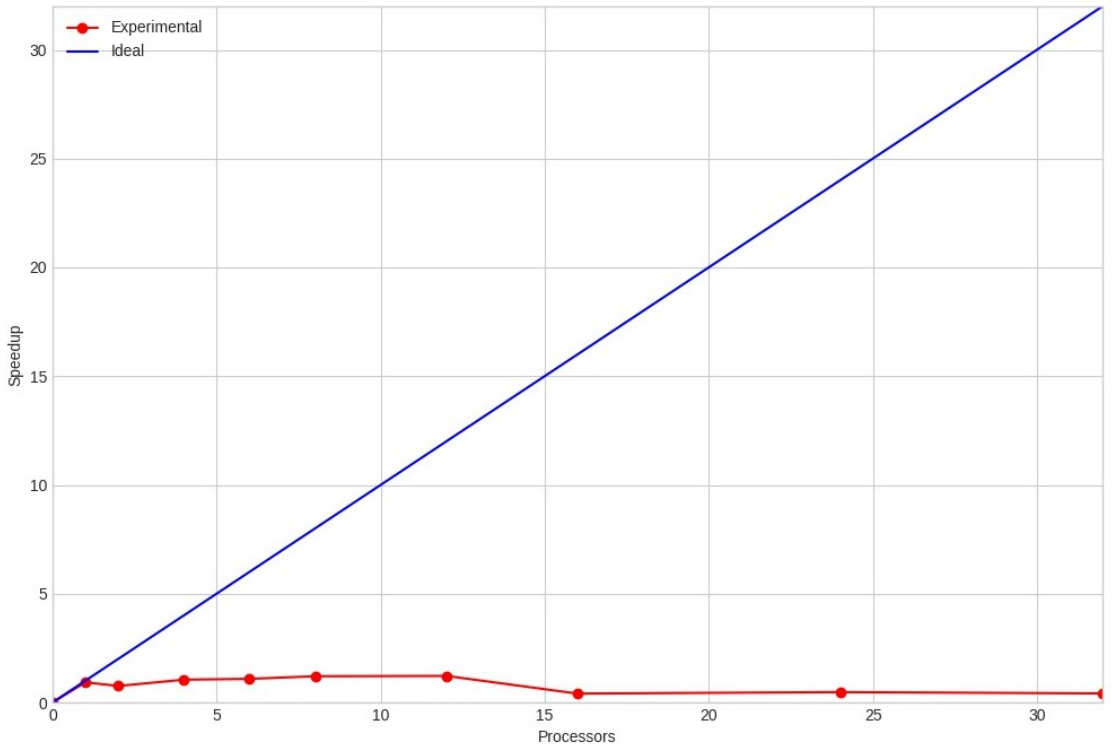

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

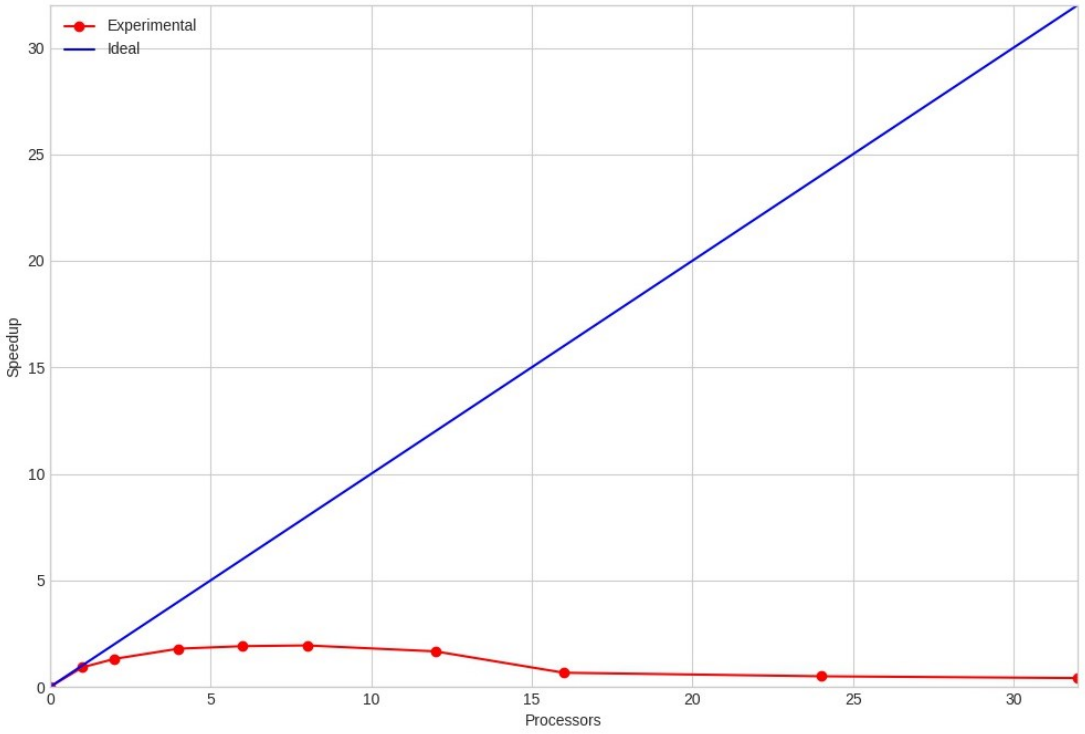

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

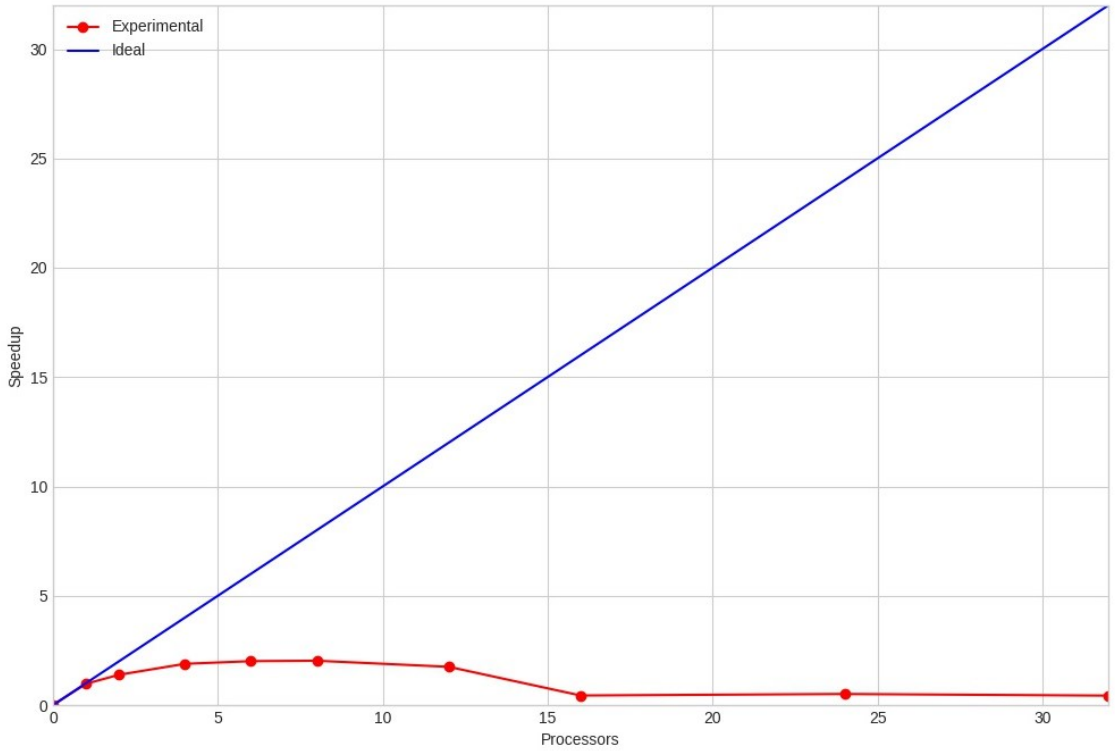

Experimental
Ideal
30
25
20
15
10
5
0
Speedup
5
10
15
20
25
30
Processors

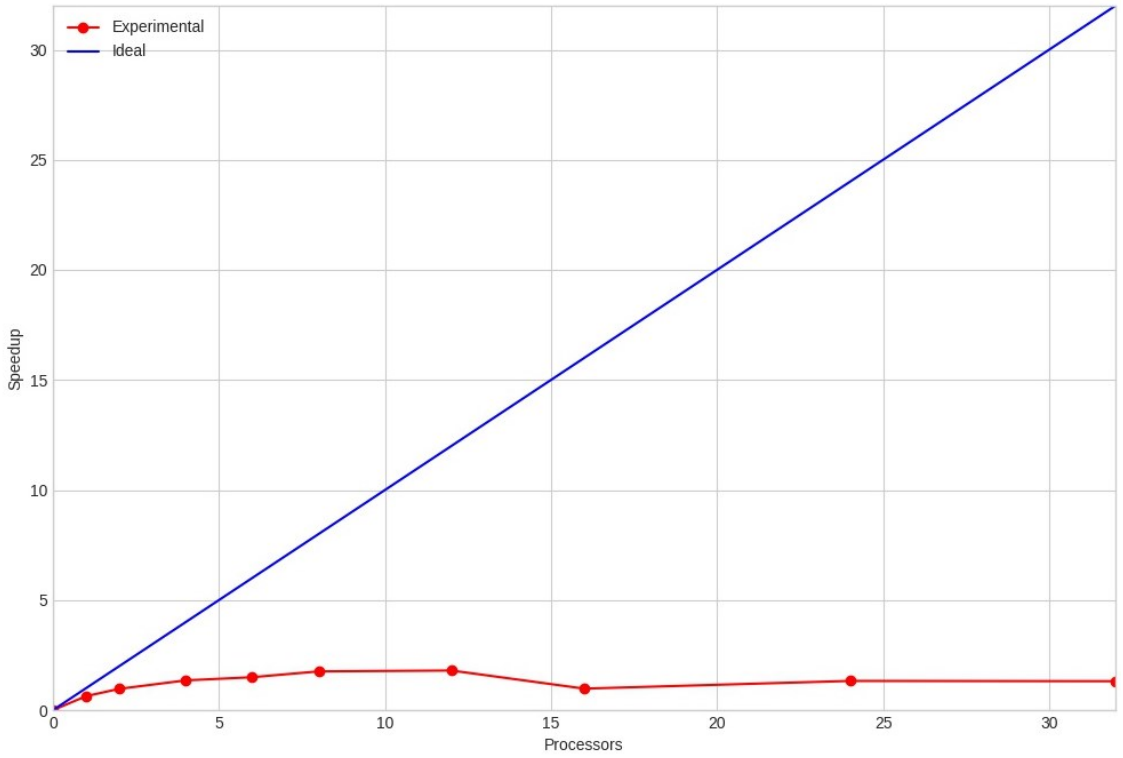

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

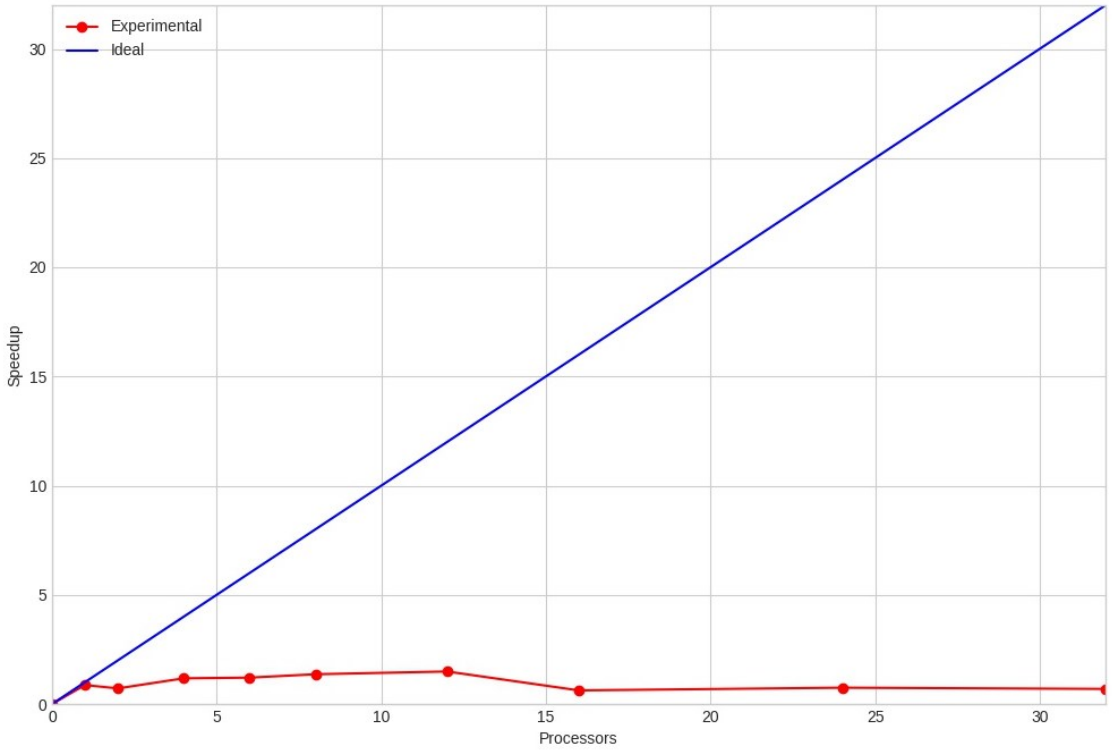

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

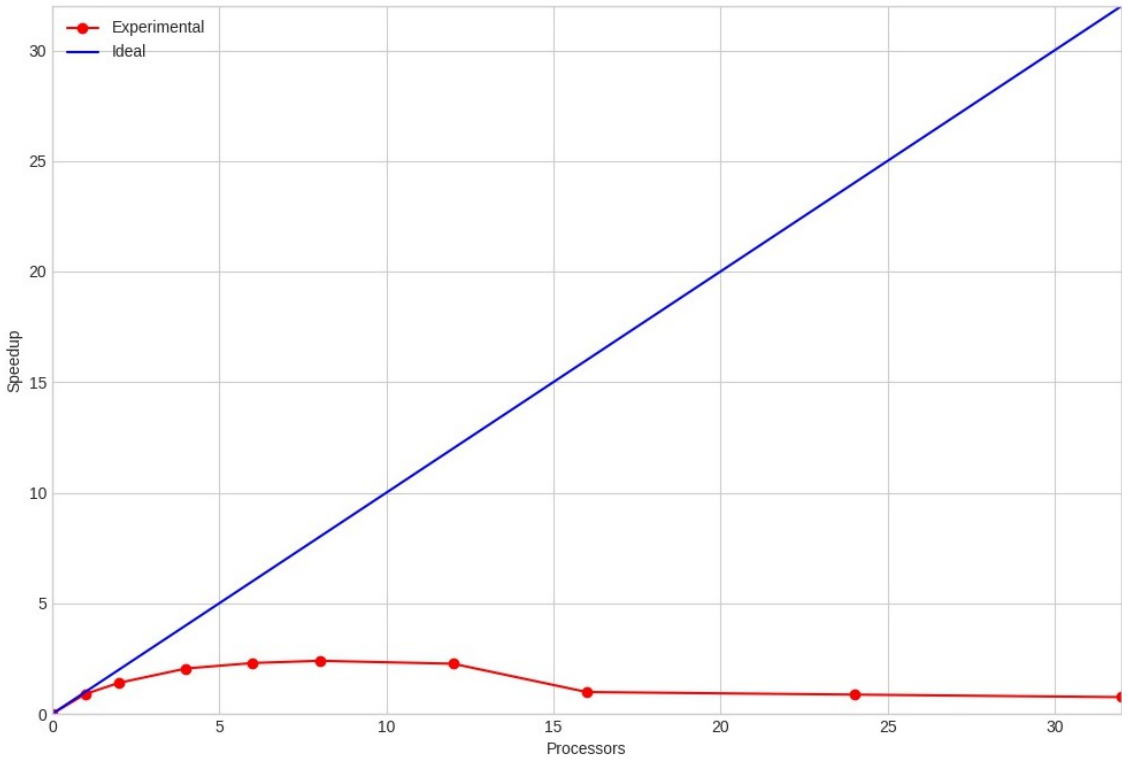

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

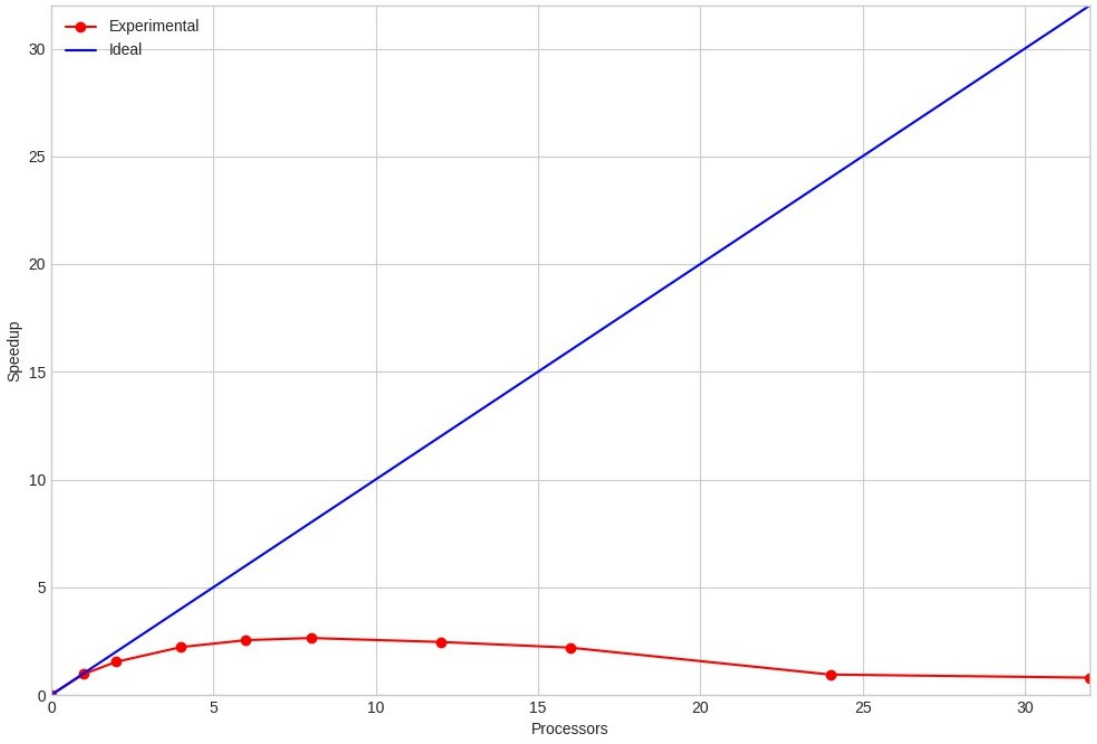

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

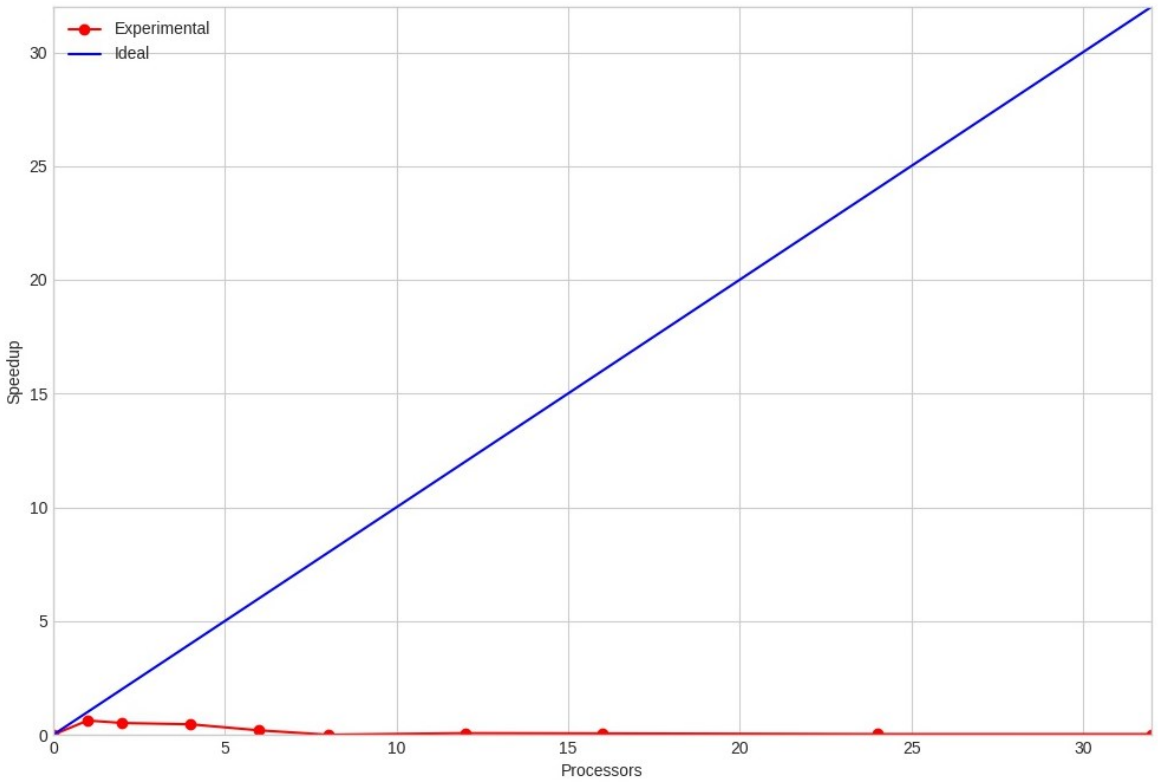

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

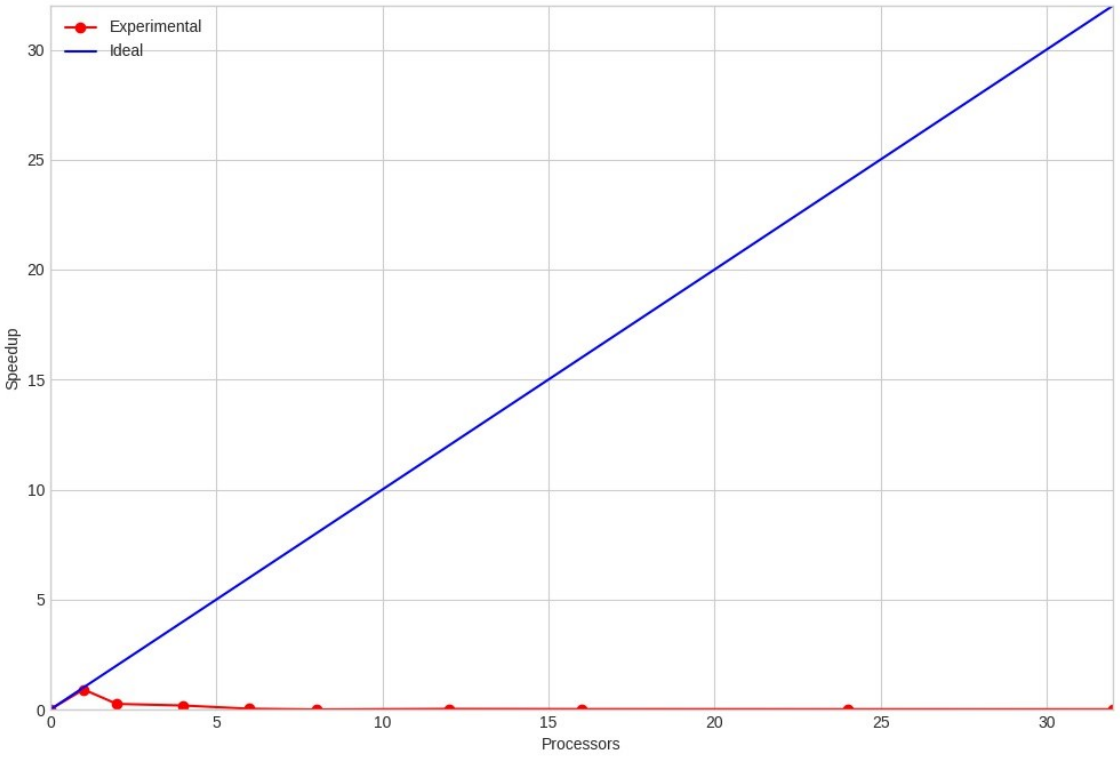

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

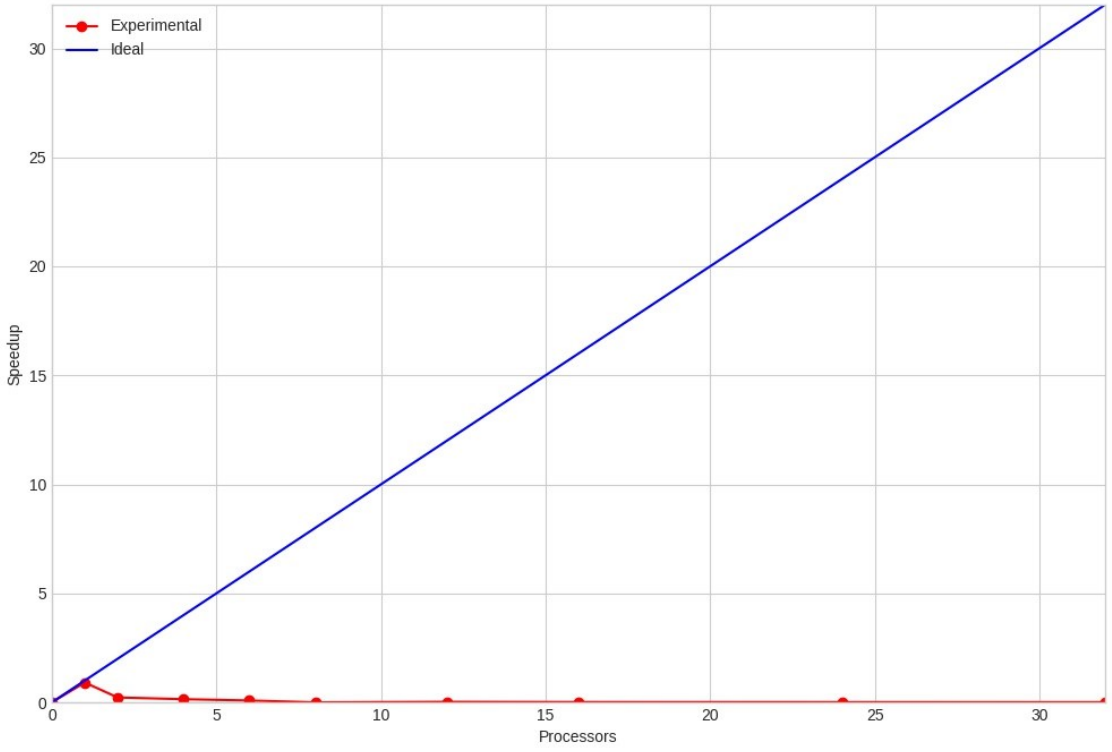

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

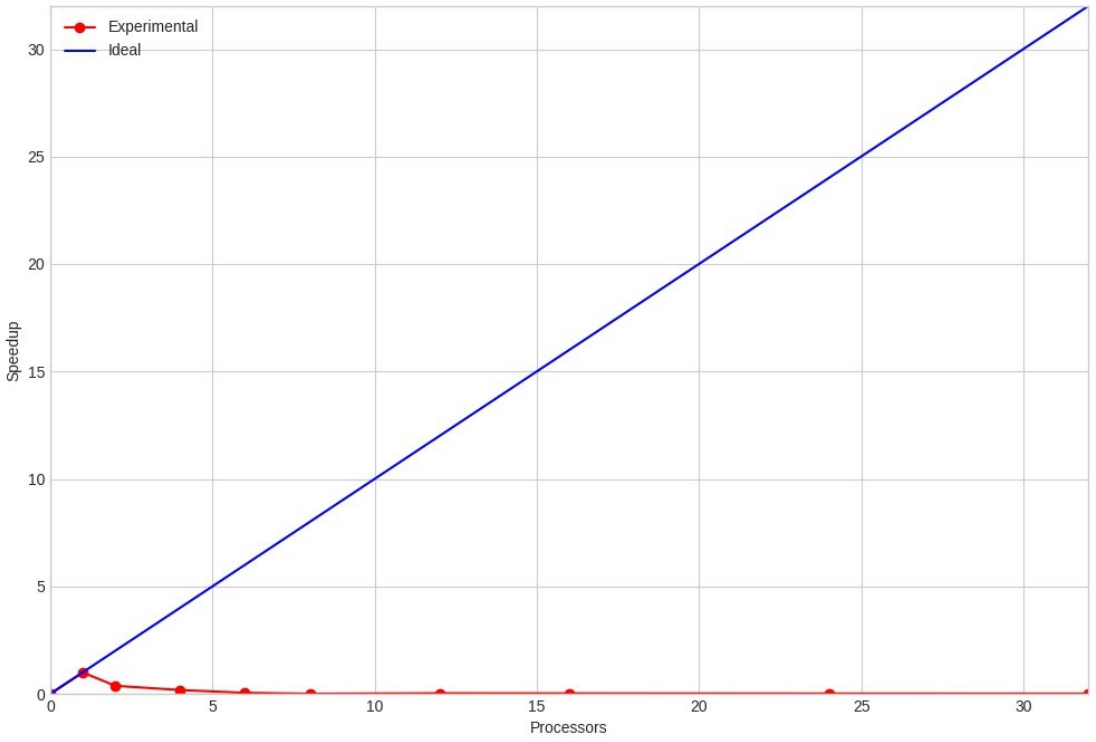
10’000, 50’000, 100’000, 500’000, 1'000’000

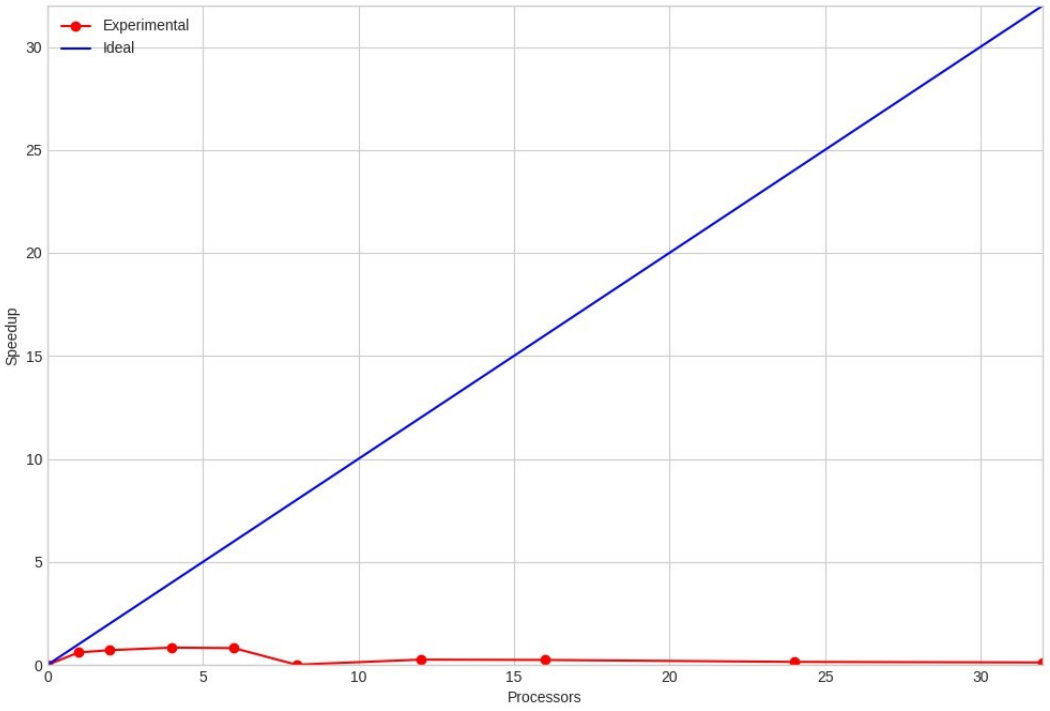
1, 2, 4, 6, 8, 12, 16, 24, 32

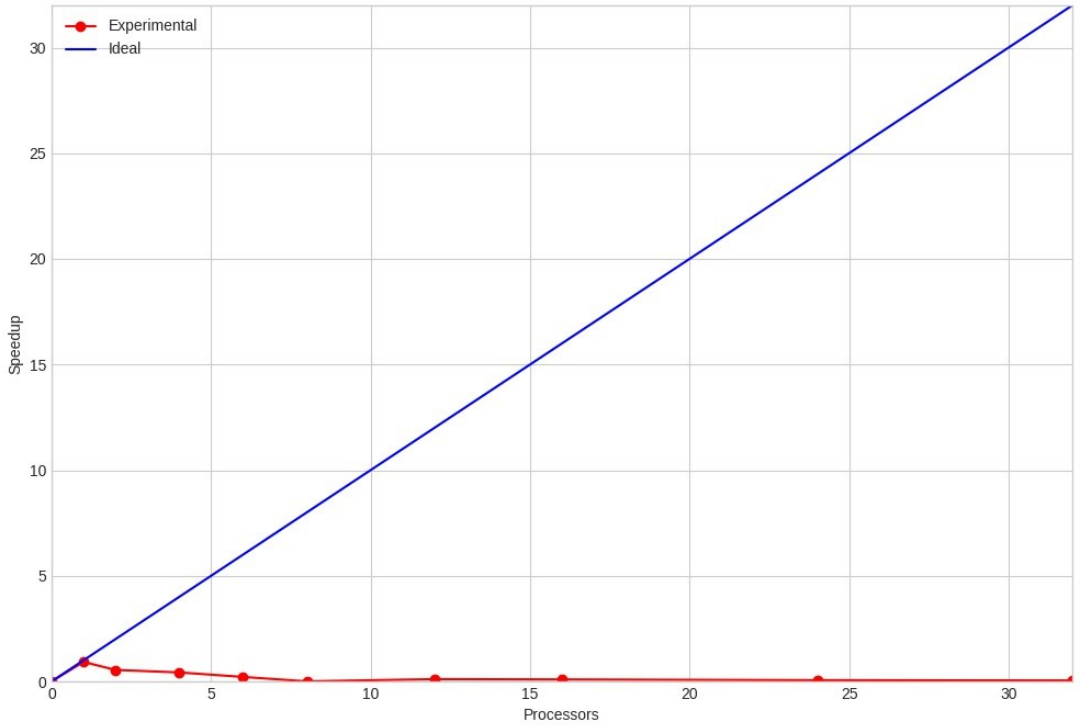

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

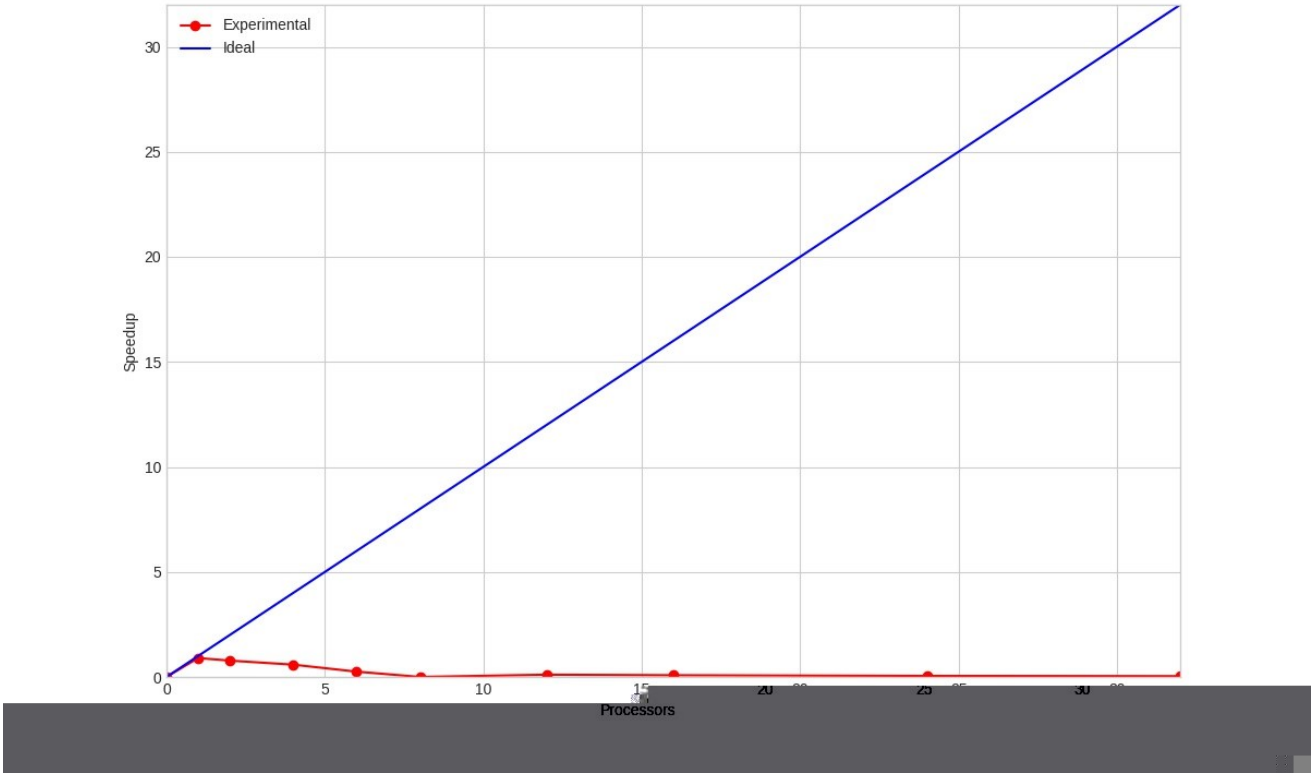

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

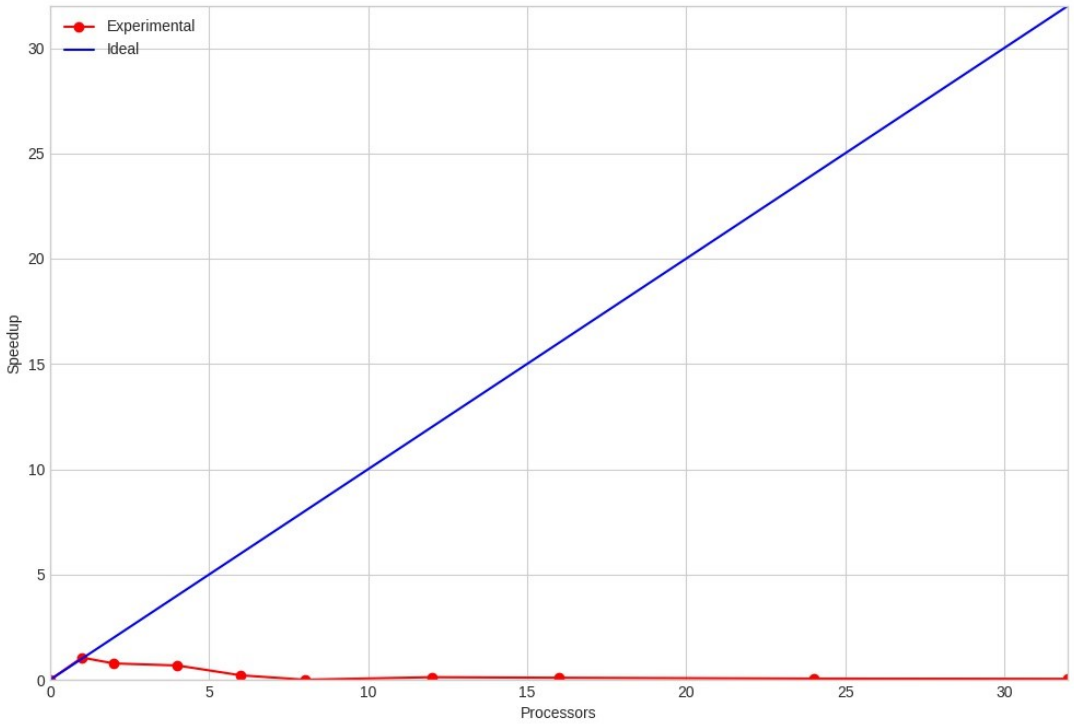

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

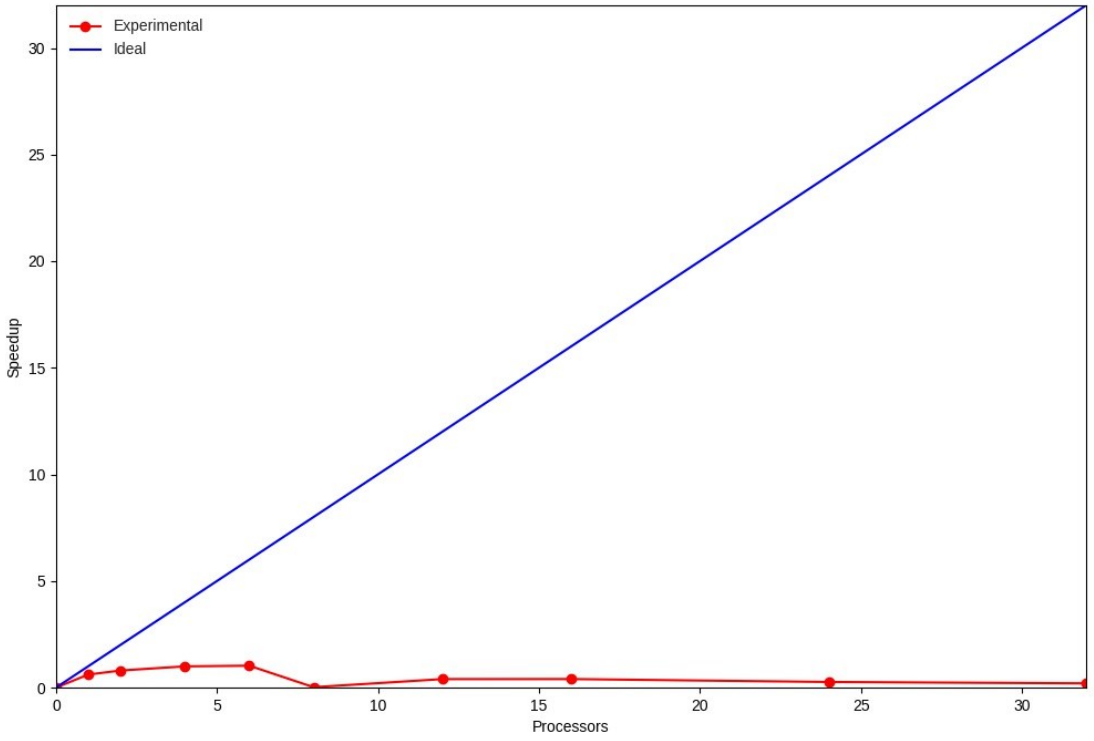

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

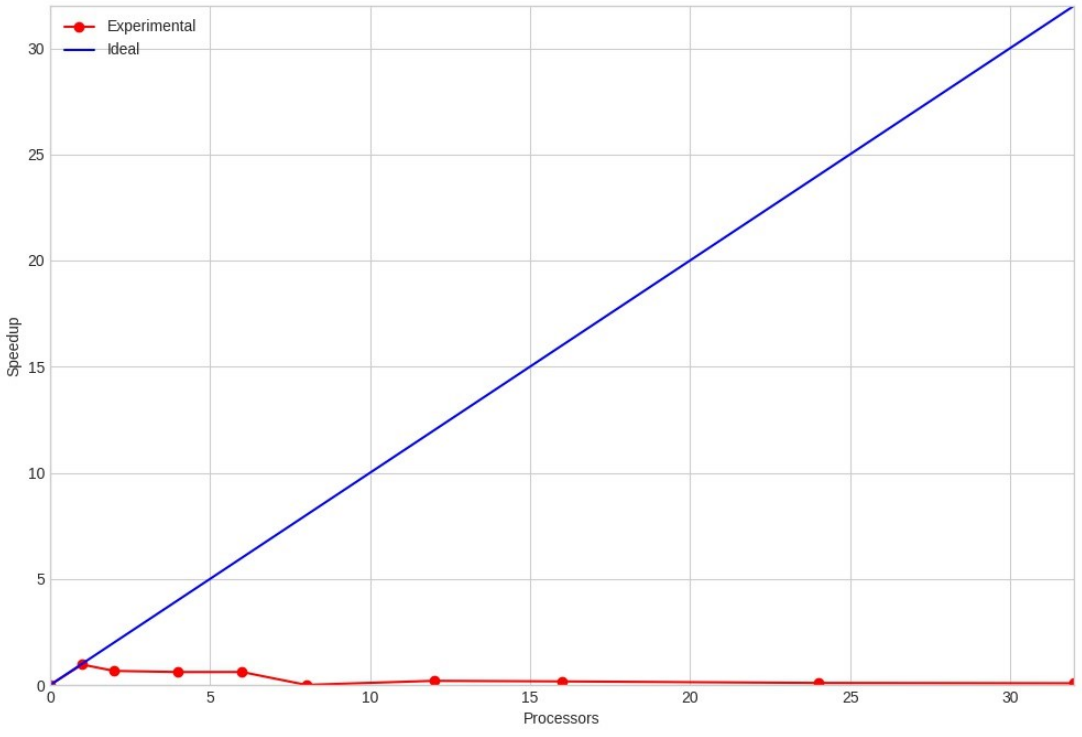

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

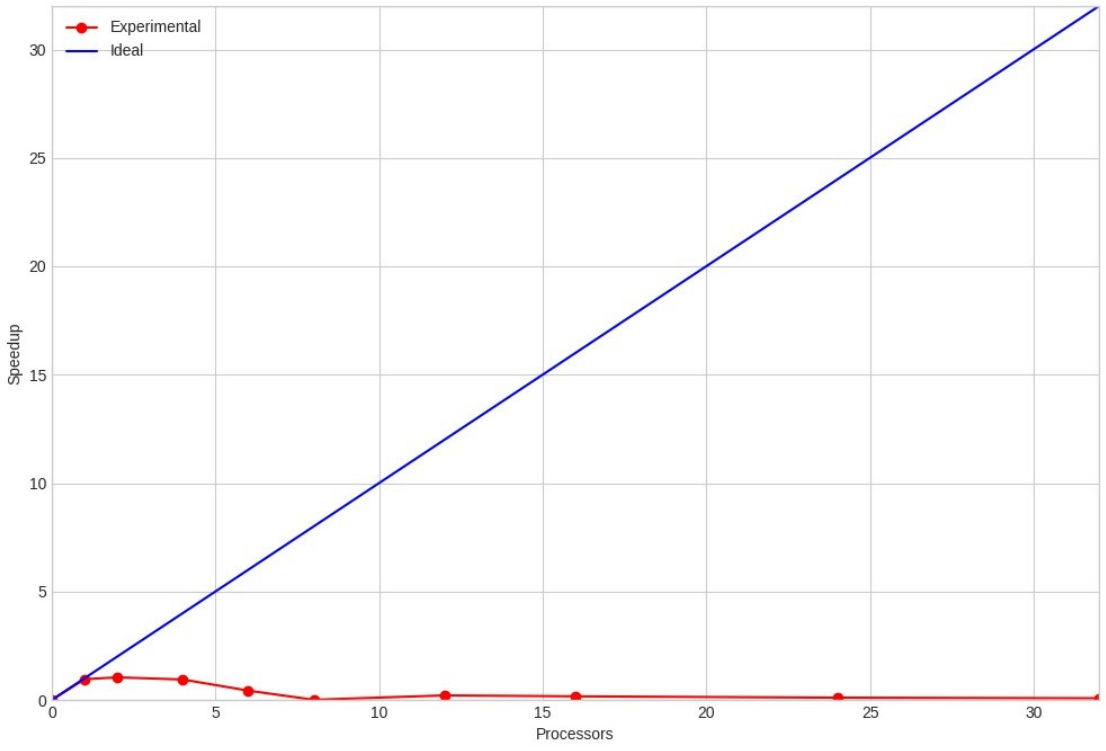

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

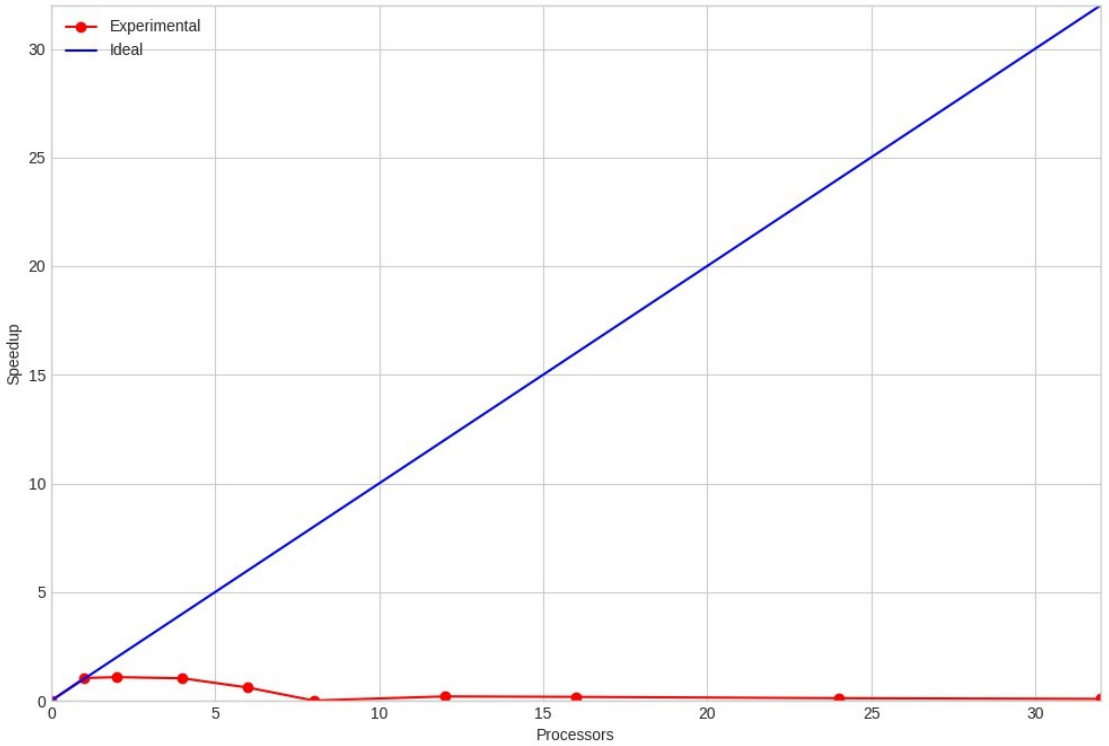

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

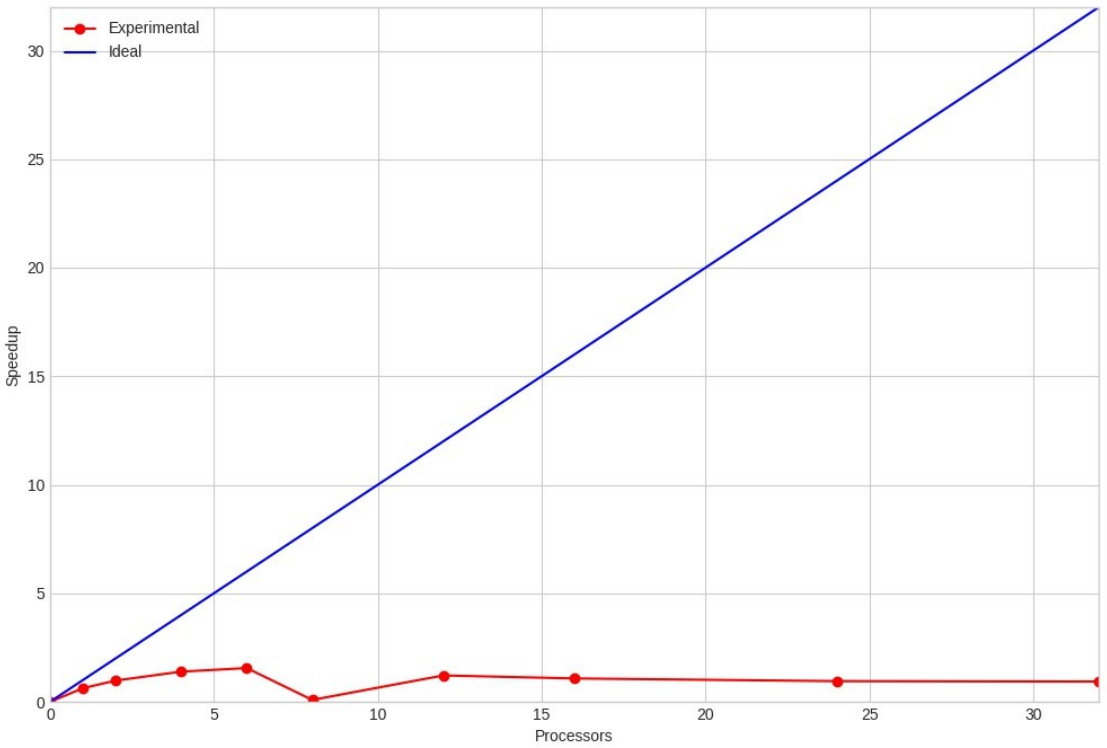

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

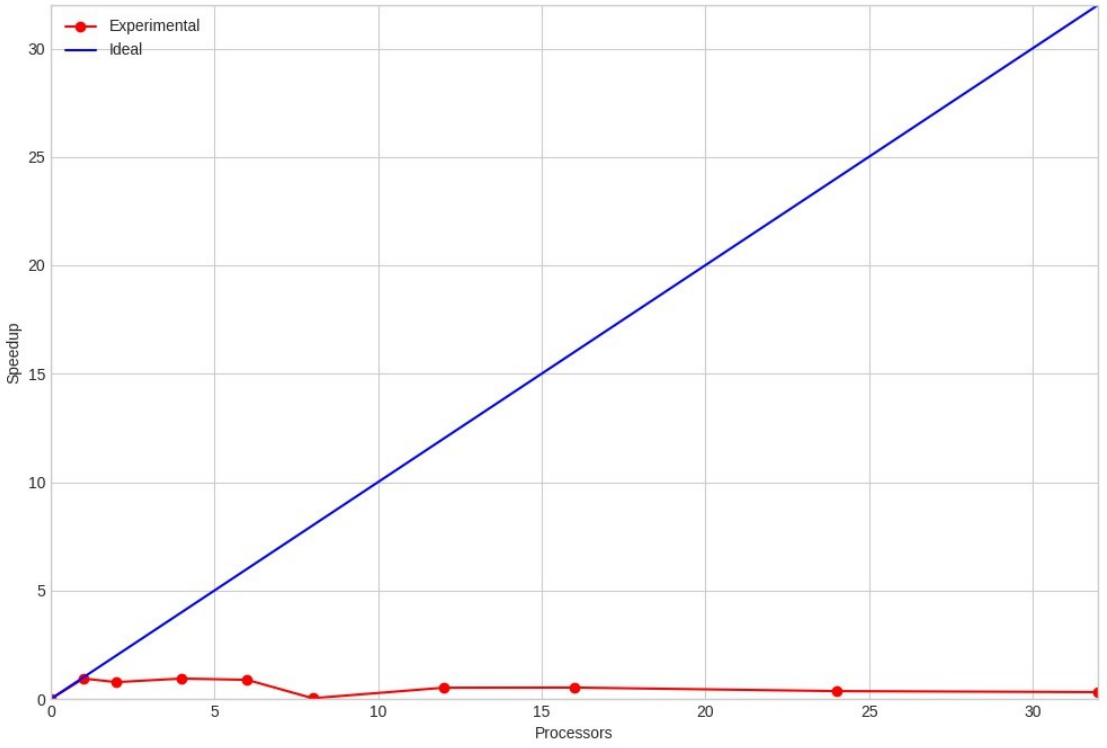

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

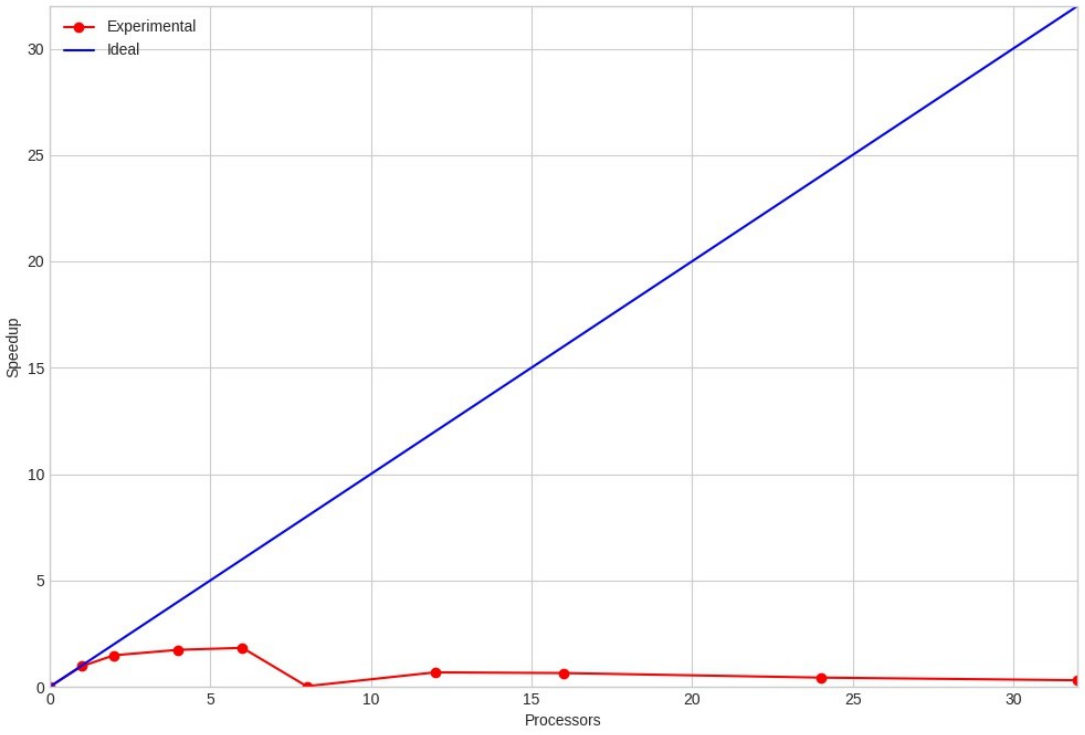

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

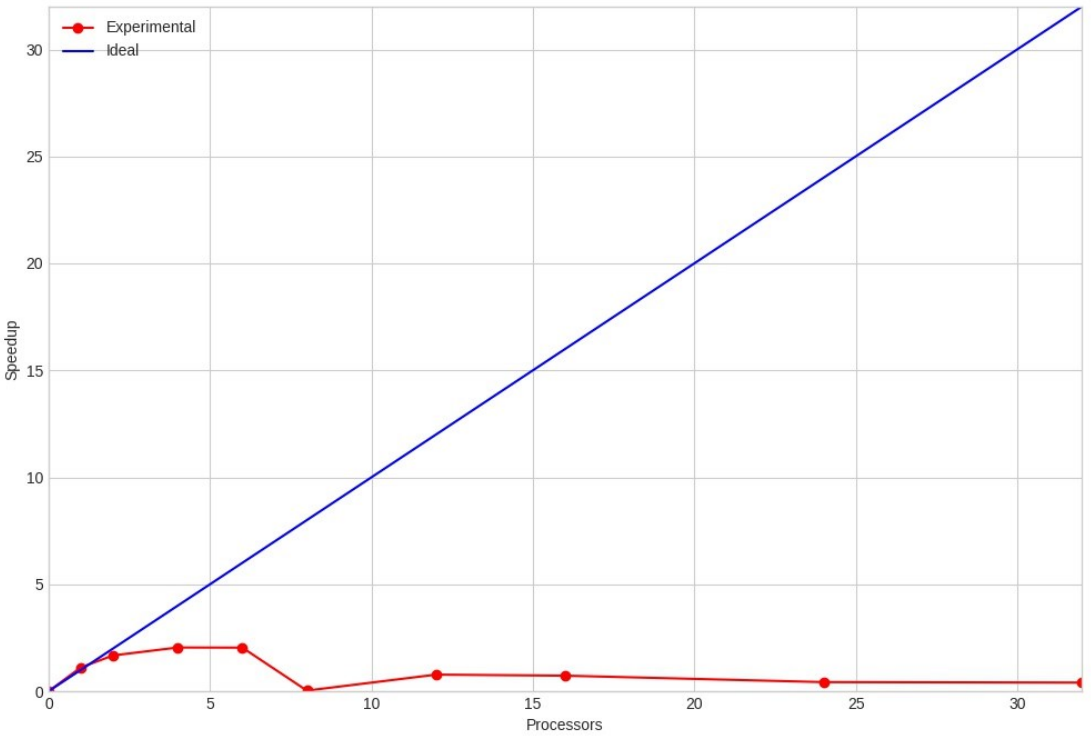

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

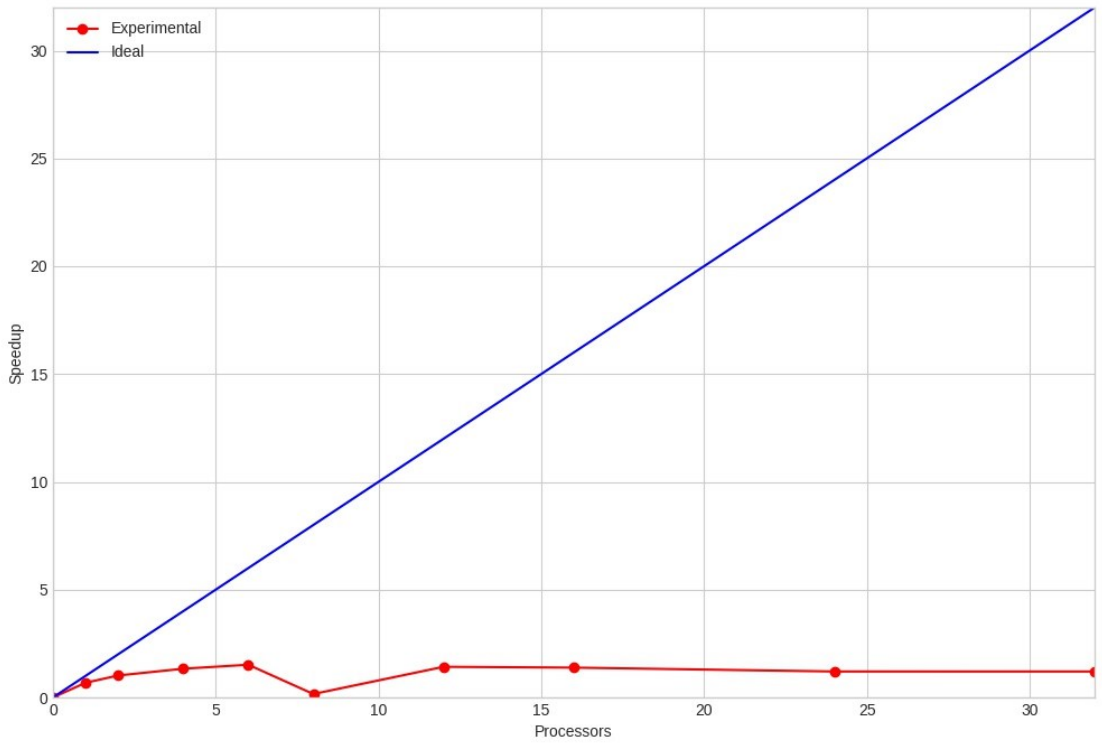

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

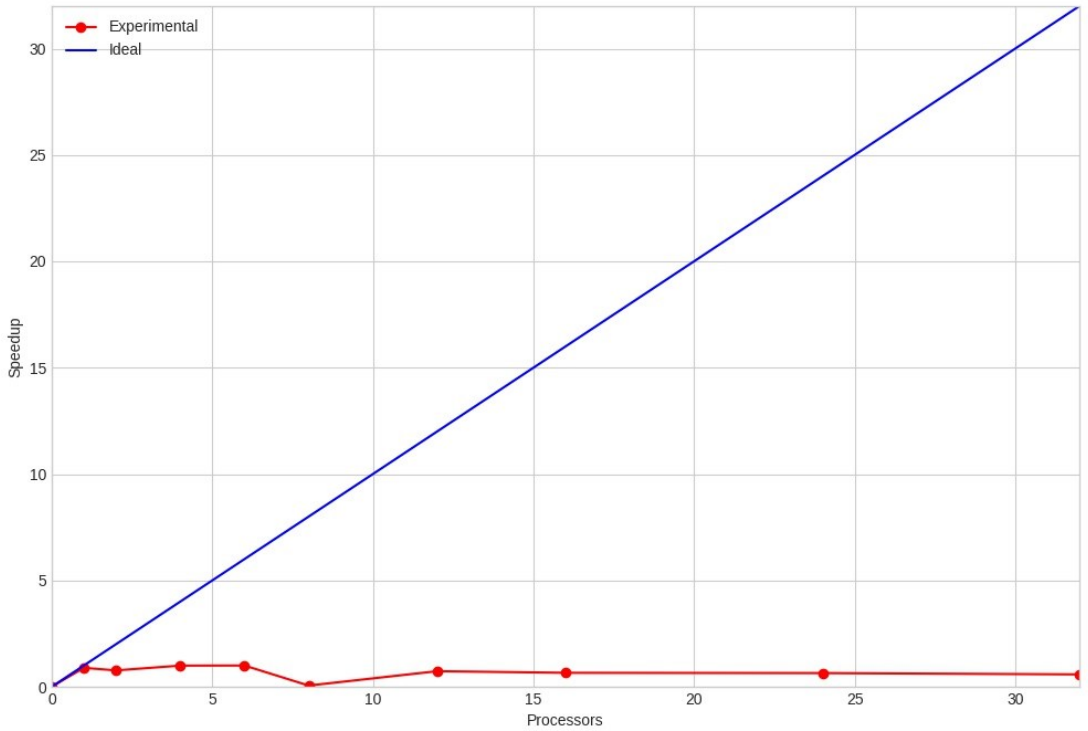

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

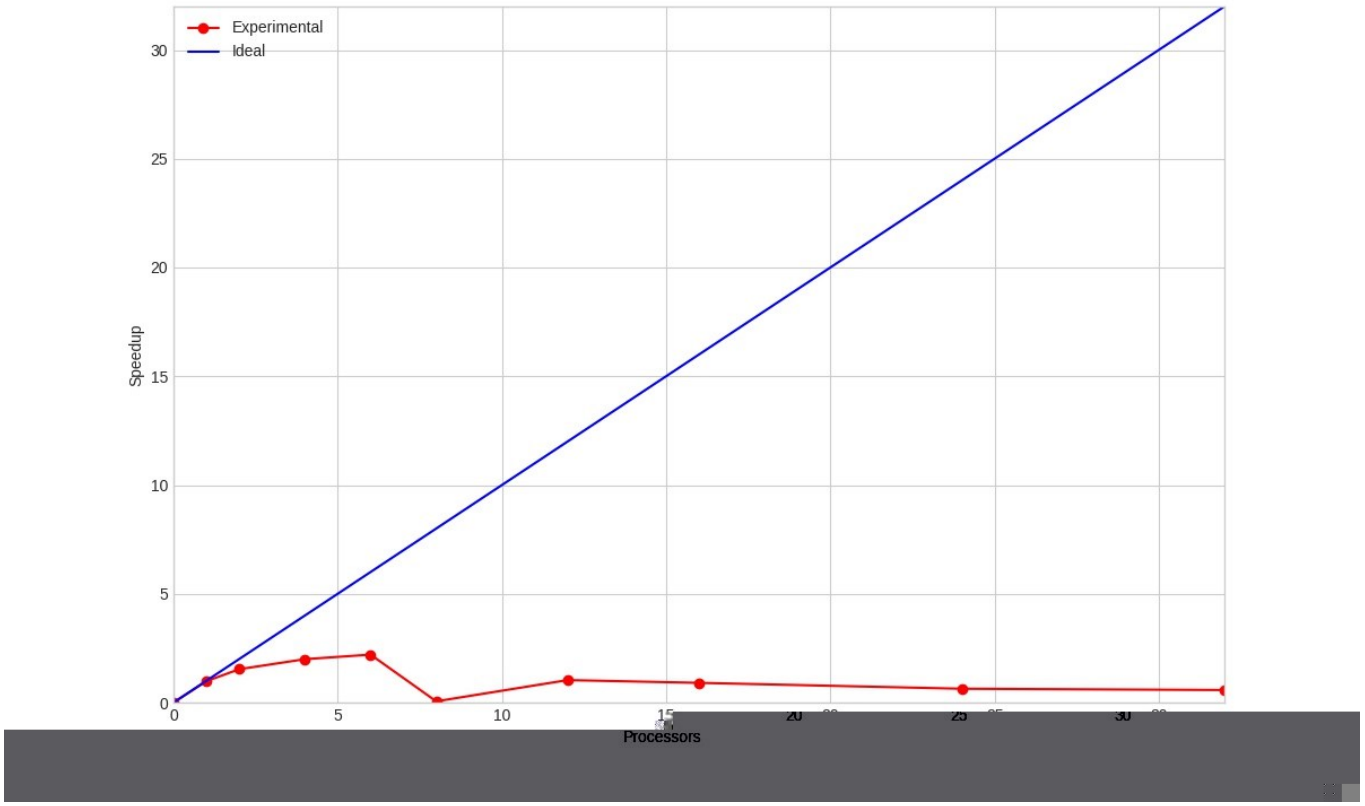

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

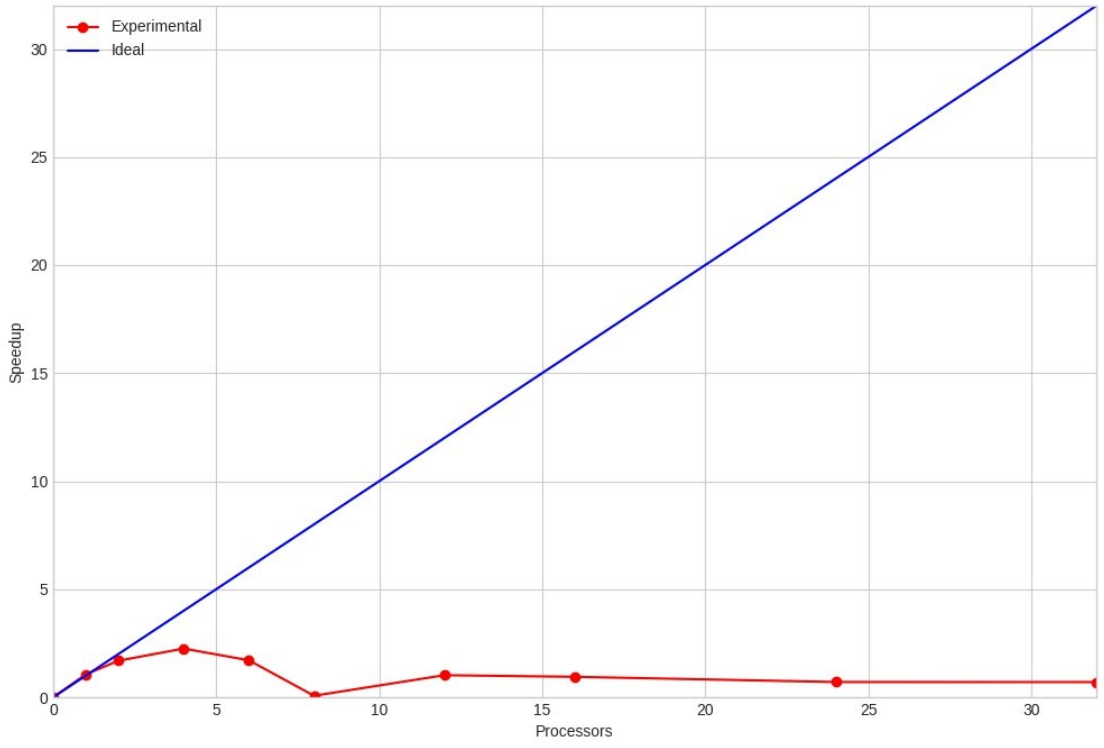

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

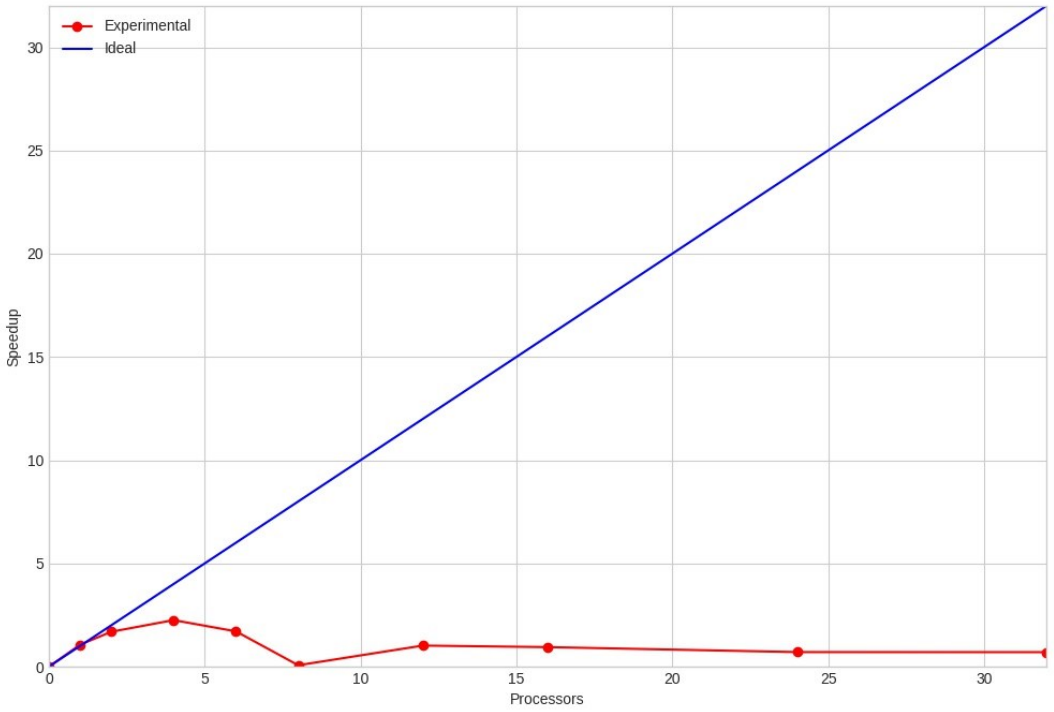

Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30


Experimental
Ideal
Speedup
Processors
0
5
10
15
20
25
30

prima fase ***pragma omp for*** ***reduction***

seconda fase

pragma omp single



define

Public Functions

This function takes in input an int pointer, the vector size and the number of threads. This function sorts the vector using parallel mode with a specific number of threads.

This function takes in input an int pointer and the vector size. This function sorts the vector using sequential mode with only one thread.

Public Functions Documentation

a

n

a

n

build ***cmake***

```
mkdir build
cd build
cmake ..
```

make

```
make generate_output
```

```
make extract_measures
```

Nota:

`measure/$TIMESTAMP`