

# merge\_sort\_report

February 28, 2020

## 1 Merge Sort

## 2 Machine Information

```
[15]: import platform
print('uname      :', platform.uname())
print('system     :', platform.system())
print('node       :', platform.node())
print('release    :', platform.release())
print('version    :', platform.version())
print('machine    :', platform.machine())
print('processor:', platform.processor())

('uname      :', ('Darwin', 'wlan802-3210.resnet.cwu.edu', '19.3.0', 'Darwin
Kernel Version 19.3.0: Thu Jan  9 20:58:23 PST 2020;
root:xnu-6153.81.5~1/RELEASE_X86_64', 'x86_64', 'i386'))
('system     :', 'Darwin')
('node       :', 'wlan802-3210.resnet.cwu.edu')
('release    :', '19.3.0')
('version    :', 'Darwin Kernel Version 19.3.0: Thu Jan  9 20:58:23 PST 2020;
root:xnu-6153.81.5~1/RELEASE_X86_64')
('machine    :', 'x86_64')
('processor:', 'i386')
```

## 3 Benchmark

Two algorithms are tested using goLang. First one is the merge sort, a sequential version and a multithreaded version is tested. Second one is the p merge sort, a sequential one, a multithreaded one and two optimized version is tested. First optimized version uses depth to limit the number of goroutine. If function is executed when depth is larger than the maximum depth, sequential merge and sequential sort will be called. Second optimized version is based upon the first optimized version and creates a pool of channels beforehand to reduce the time of channel creation during execution.

```
[25]: import csv
import pandas as pd
result_data = pd.read_csv("../out/results.csv")
```

```

result_data
result_data.plot(figsize=(15,5),x='population', y=['merge sort( $\mu$ s)', 'mt merge_  

↳ sort( $\mu$ s)', 'p merge sort( $\mu$ s)', 'mt p merge sort( $\mu$ s)', 'mt p merge sort_  

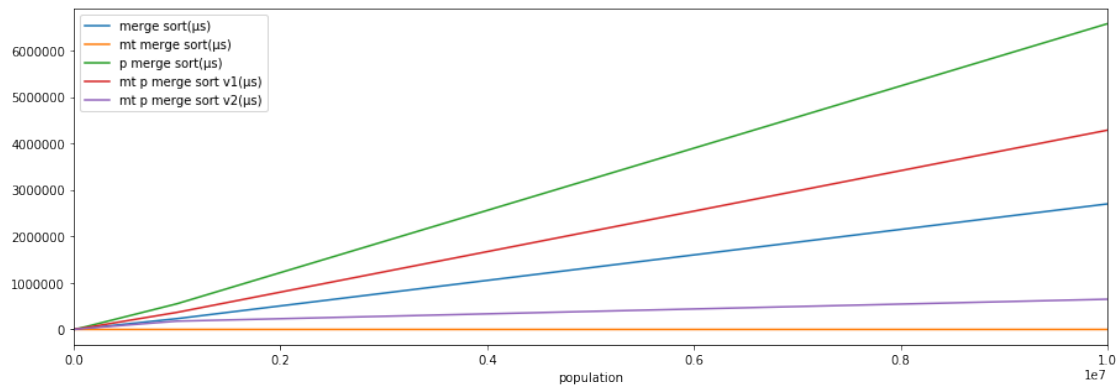
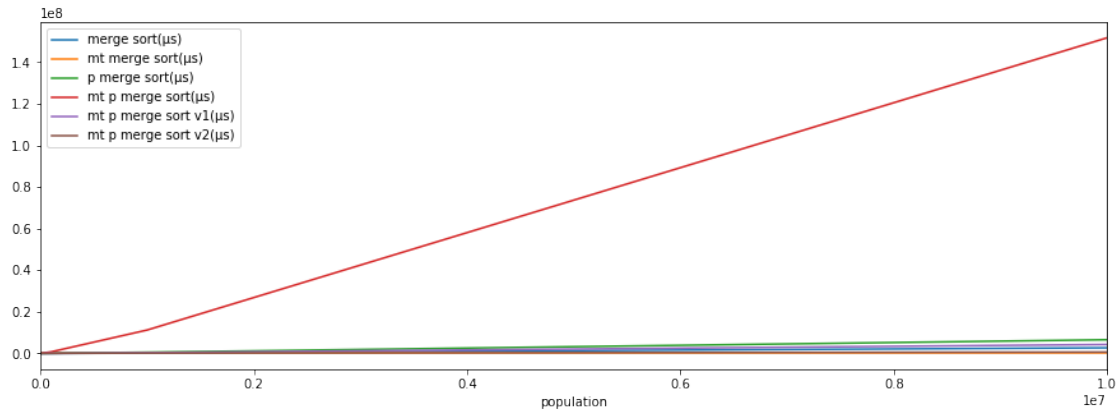
↳ v1( $\mu$ s)', 'mt p merge sort v2( $\mu$ s)'])
result_data.plot(figsize=(15,5),x='population', y=['merge sort( $\mu$ s)', 'mt merge_  

↳ sort( $\mu$ s)', 'p merge sort( $\mu$ s)', 'mt p merge sort v1( $\mu$ s)', 'mt p merge sort_  

↳ v2( $\mu$ s)'])

```

[25]: <matplotlib.axes.\_subplots.AxesSubplot at 0x10bca95d0>



[ ]: