Benchmarking Analysis for a Multi-Threaded Version of Quicksort

Authors: Oussama Oulkaid, ##, ##

November, 2021

1. Introduction

The aim of this activity is to analyse the time spent by Quicksort, by pinpointing the parameters that might affect its performance (array size, number of cores on the machine, nature of other applications running at the same time, etc.).

To get started, compile the program by running:

```
make -C src/
```

2. Choices we made

The `run_benchmarking.sh` script have been modified so that the array size samples are chosen to be incremented by the same amount. And to simplify the experimentation process, the Perl script that build the csv file is now being run within the `run_benchmarking.sh`. Also, the txt files are not anymore preserved.

To launch an experiment you need to set the `START_SIZE`, `MAX_SIZE` and `STEP` constants as you want. Then, run:

```
./scripts/run_benchmarking.sh
```

```
# The environment
library(tidyverse)
library(ggplot2)
library(reshape2)
```

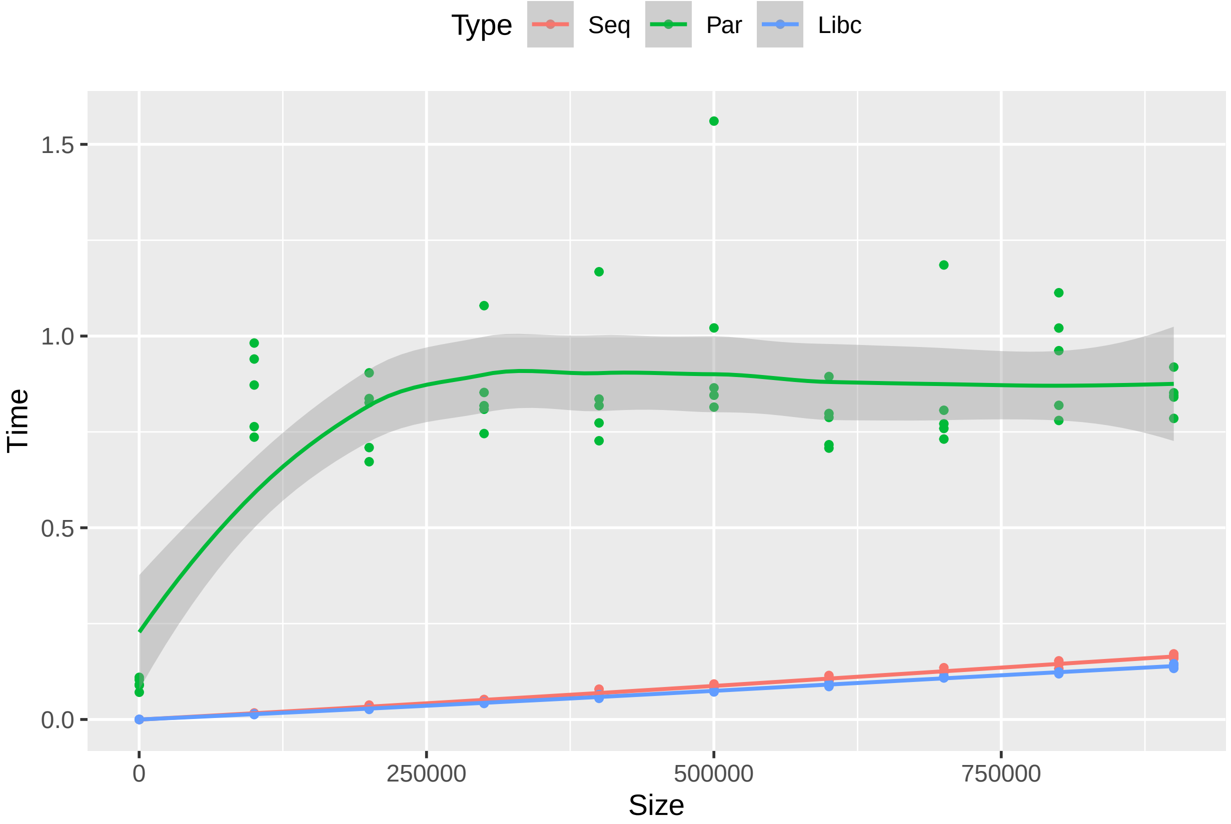
3. Build the Dataframe from file

```
#parameters: START_SIZE=100, MAX_SIZE=1000000, STEP=100000
df1 <- read.csv("data/in_2021-11-19/measurements_22:56.csv",header=T)
df1 <- melt(df1, id.vars="Size")
names(df1)[2] <- "Type"
names(df1)[3] <- "Time"

#parameters: START_SIZE=0, MAX_SIZE=1000000, STEP=10000
df2 <- read.csv("data/in_2021-11-20/measurements_00:12.csv",header=T)
df2 <- melt(df2, id.vars="Size")
names(df2)[2] <- "Type"
names(df2)[3] <- "Time"

#parameters: START_SIZE=0, MAX_SIZE=2000000, STEP=100000
df3 <- read.csv("data/in_2021-11-20/measurements_00:28.csv",header=T)
df3 <- melt(df3, id.vars="Size")
names(df3)[2] <- "Type"
names(df3)[3] <- "Time"

#parameters: START_SIZE=0, MAX_SIZE=2000000, STEP=100000
df3 <- read.csv("data/in_2021-11-20/measurements_00:28.csv",header=T)
df3 <- melt(df3, id.vars="Size")
names(df3)[2] <- "Type"
names(df3)[3] <- "Time"
```

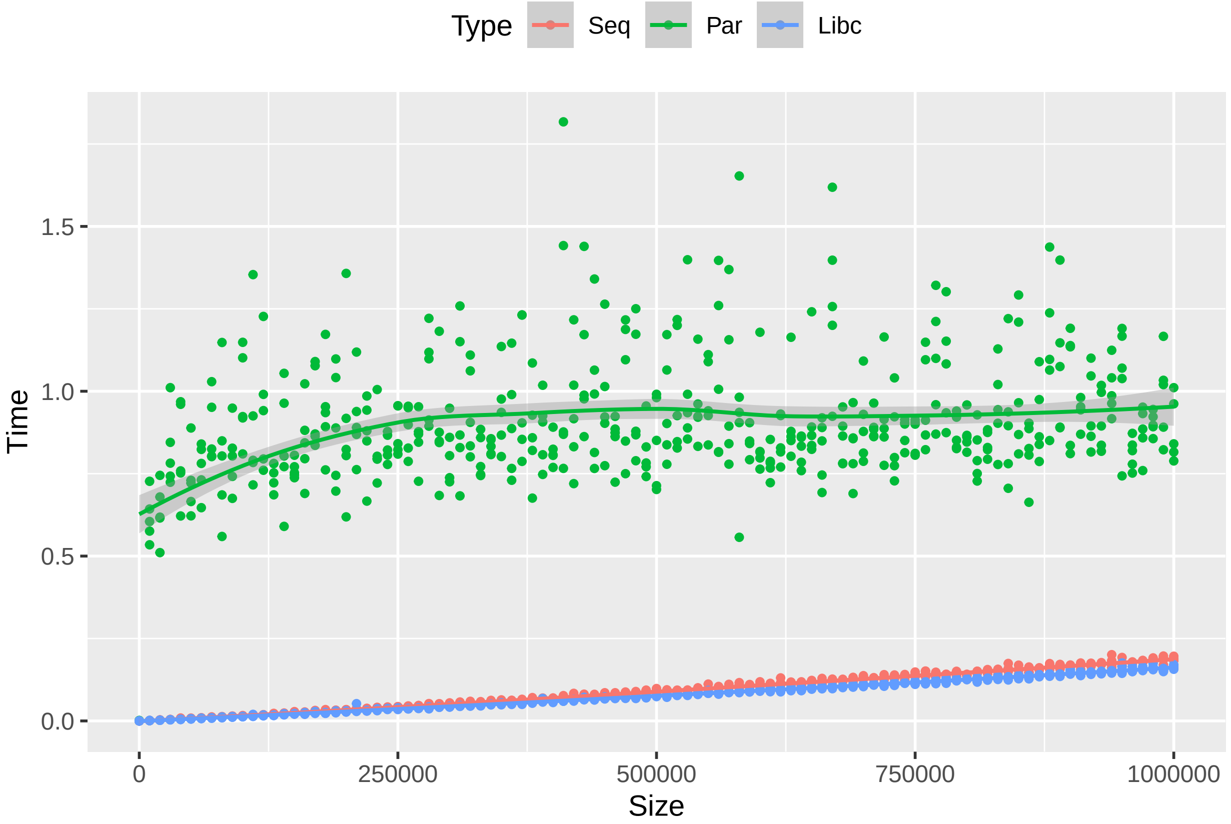
```
#parameters: START_SIZE=0, MAX_SIZE=10000000, STEP=100000
df4 <- read.csv("data/in_2021-11-20/measurements_00:33.csv",header=T)
df4 <- melt(df4, id.vars="Size")
names(df4)[2] <- "Type"
names(df4)[3] <- "Time"

#parameters: START_SIZE=0, MAX_SIZE=100000000, STEP=10000000
df5 <- read.csv("data/in_2021-11-20/measurements_01:07.csv",header=T)
df5 <- melt(df5, id.vars="Size")
names(df5)[2] <- "Type"
names(df5)[3] <- "Time"

#parameters: START_SIZE=0, MAX_SIZE=15000000, STEP=1000000
df6 <- read.csv("data/in_2021-11-20/measurements_01:44.csv",header=T)
df6 <- melt(df6, id.vars="Size")
names(df6)[2] <- "Type"
names(df6)[3] <- "Time"
```

4. Plotting

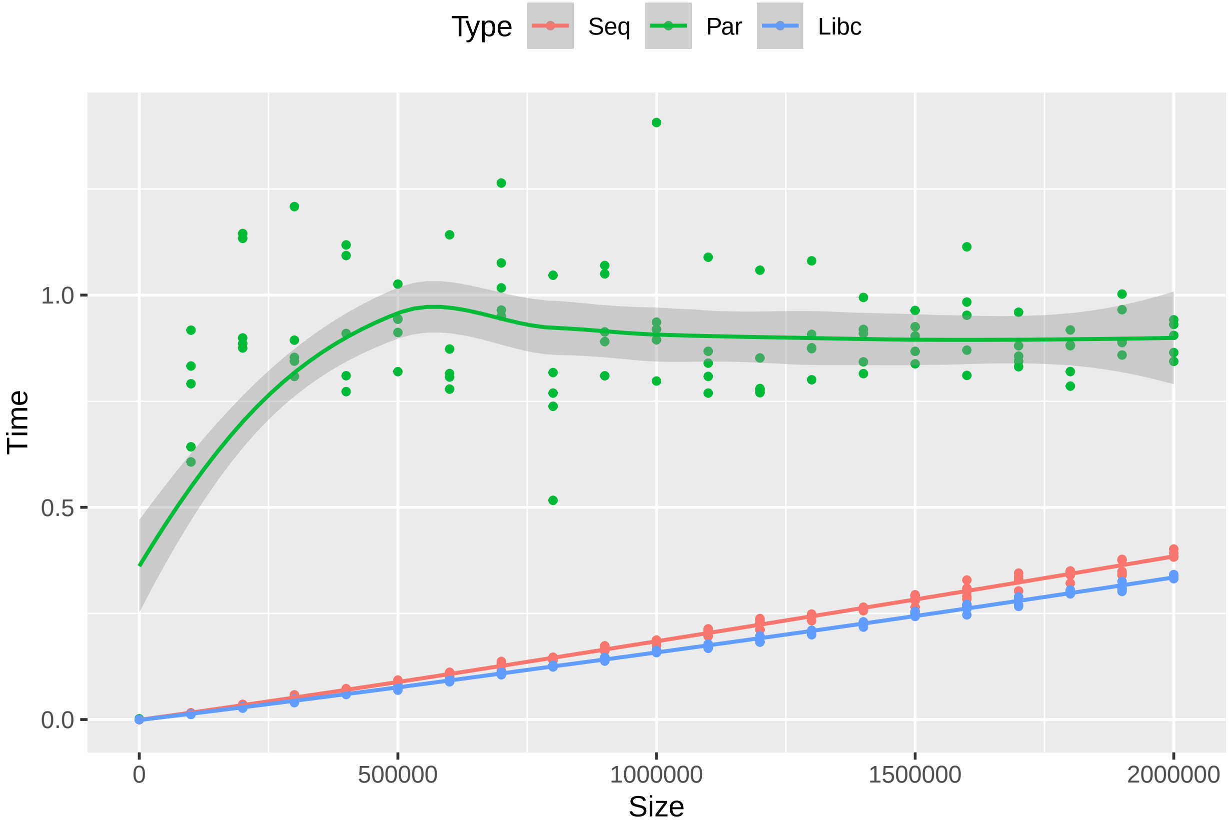
```
ggplot(df1, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



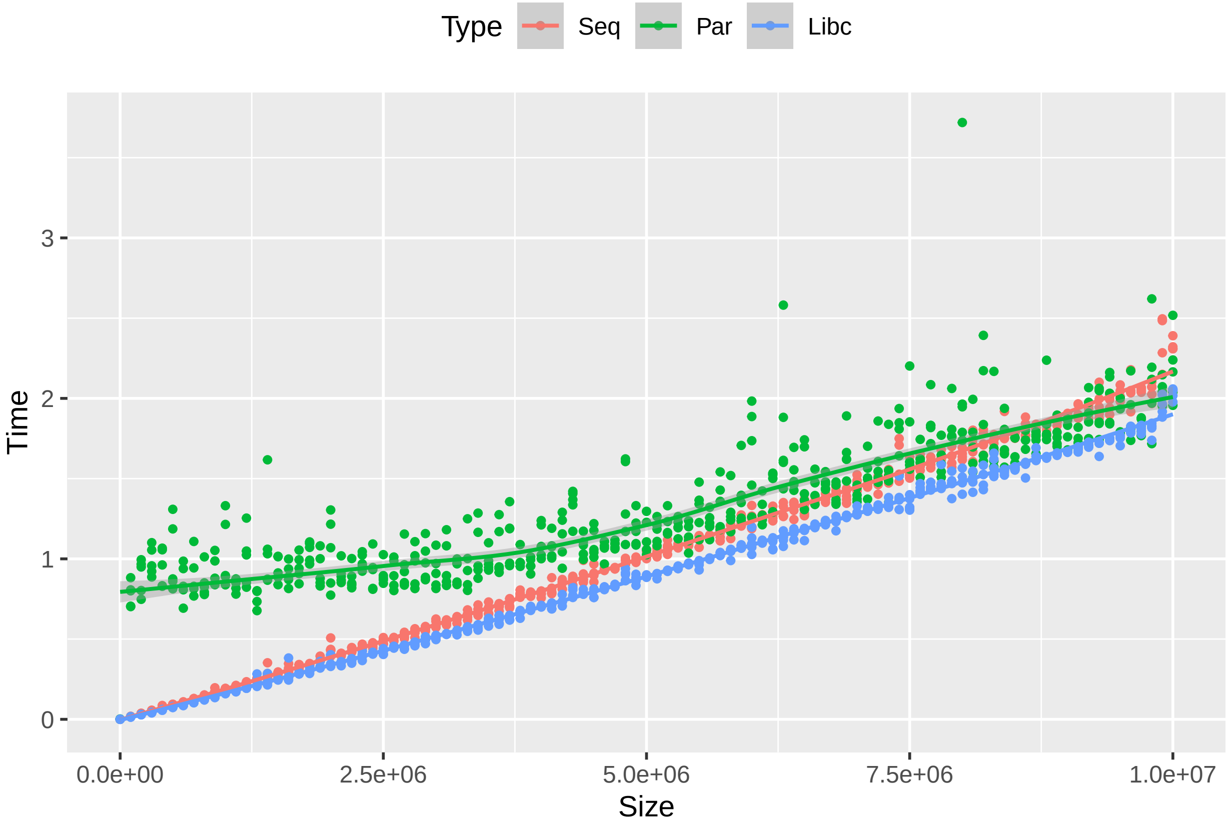
```
ggplot(df2, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



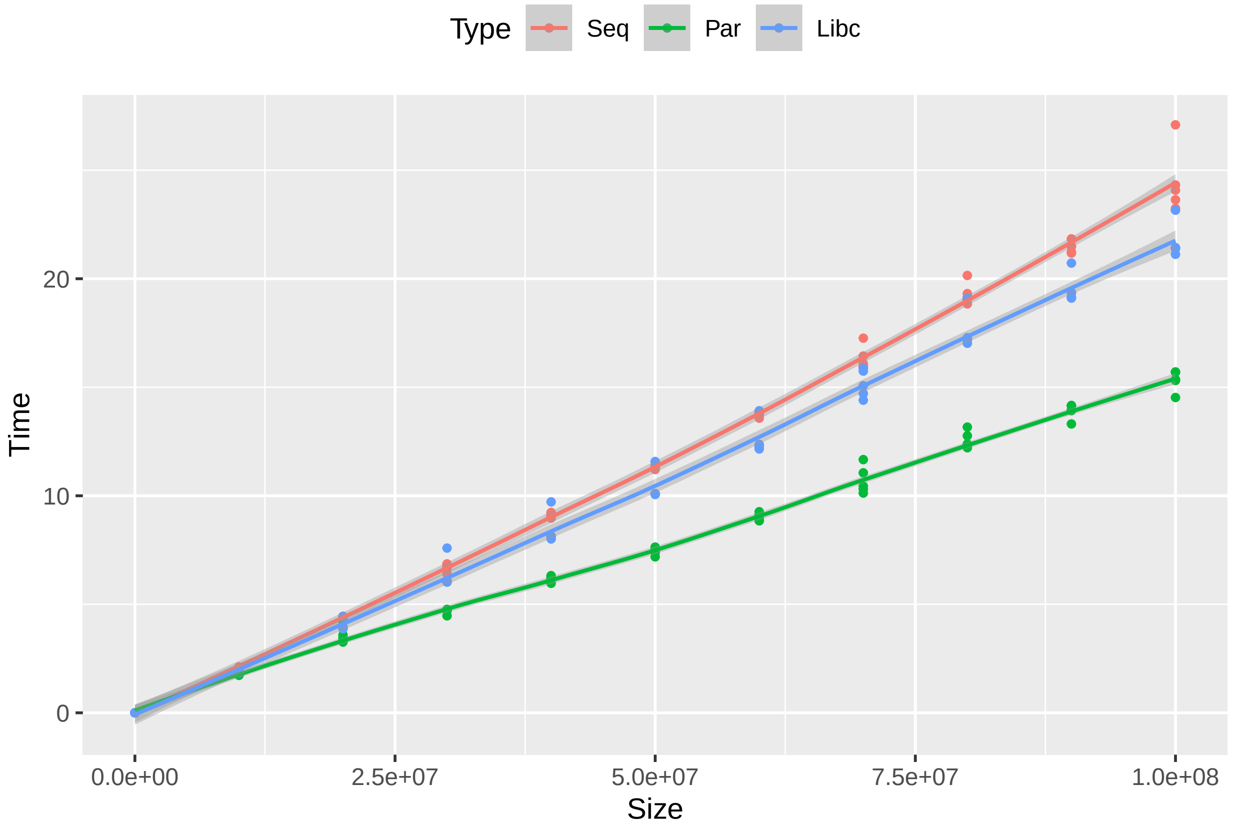
```
ggplot(df3, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



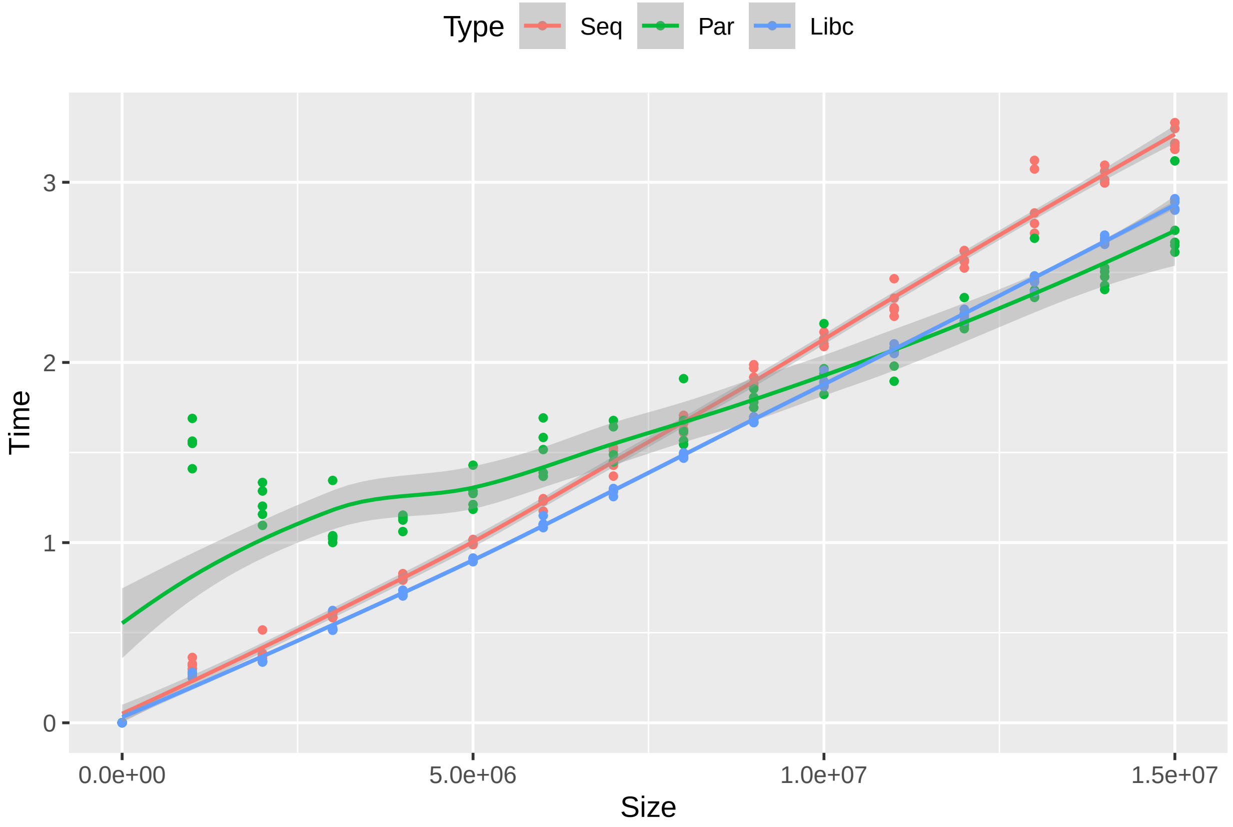
```
ggplot(df4, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



```
ggplot(df5, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



```
ggplot(df6, aes(Size, Time, colour=Type)) +
  geom_point(size = 1) +
  stat_smooth(size = 0.7) +
  theme( plot.title = element_text(hjust = 0.5), legend.position = "top" )
```



Comment: TODO