Gateway to Data Science Final Project

The Research of Different CPUs released from 2000 and the future trends.

Randy Li

Zhixuan Huang

Randy Li
Zhixuan Huang

Introduction

This dataset has different CPUs and GPUs released until 2021, and their development during times. The goal of the visualization project is to help build a clear vision of the development trend. In this project, we are more focused on the dataset of CPUs and looking forward to predicting its trend in the future: Is the Process Size getting smaller or larger? Is the Thermal Design Power increasing? Furthermore, we will compare and analyze different factors that will affect CPU performance.

Data Preparation

This is a source from the website Kaggle, an online data science community platform under Google. Michael Bryant creates the dataset. Since the dataset is a large one with 2185 CPUs and 2688 GPUs, we are going to focus more on the CPU data in this project. To do that, we

1. delete the unused columns, the one containing all NAN.
2. convert the date from categorical to numerical
3. split the data

Descriptive Statistics

According to the dataset, both Intel and AMD tend to make smaller process-size CPUs as time progresses. That being said, the later the release date is, the smaller the CPUs are. However, the rate of change in the size is also decreasing. The Thermal Design Power is also increasing as the release date increases.

Visualization

According to the “process size vs. release date” graph, we can see a curve pointing downwards, indicating that as the release date increases, the process size decreases. Furthermore, the curve is concave up and decreasing, which indicates that the rate of decrease is also decreasing. This is probably due to the limits of current technology. For Thermal Design Power, two graphs are significantly important: the “TDP vs. release date” graph and the “TDP vs. process size” graph. The previous graph shows an ambiguous relationship. However, we still observe that CPUs with higher TDP appeared after 2010, and the maximum TDP still increases as the release date increases. The other graph is clearer and indicates less TDP with a larger process size. It is a negative relation.

Randy Li
Zhixuan Huang

Statistical Inference

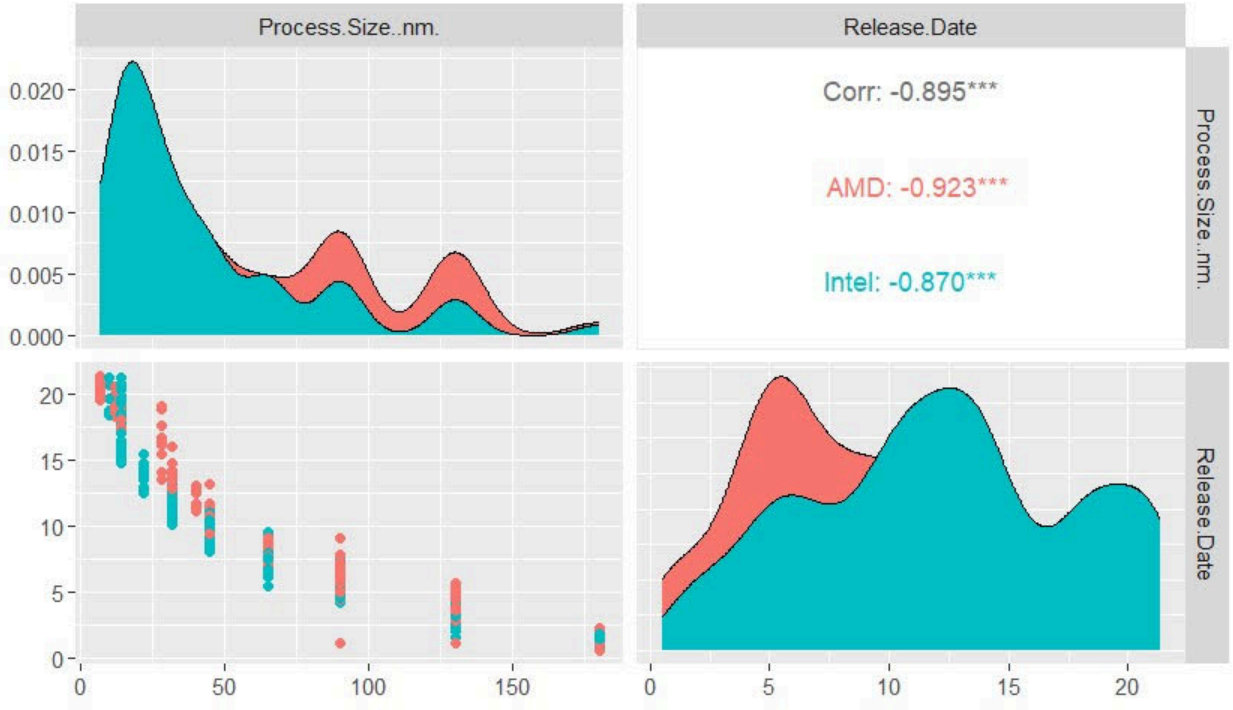
Hypothesis test of the relation of frequency and process size.
Hypothesis: Smaller process size causes less frequency.
Applying the hypothesis test

Confidence interval of process size: 50.20401 53.73486
Confidence interval of frequency: 2450.761 2514.025
p-value: < 2.2e-16

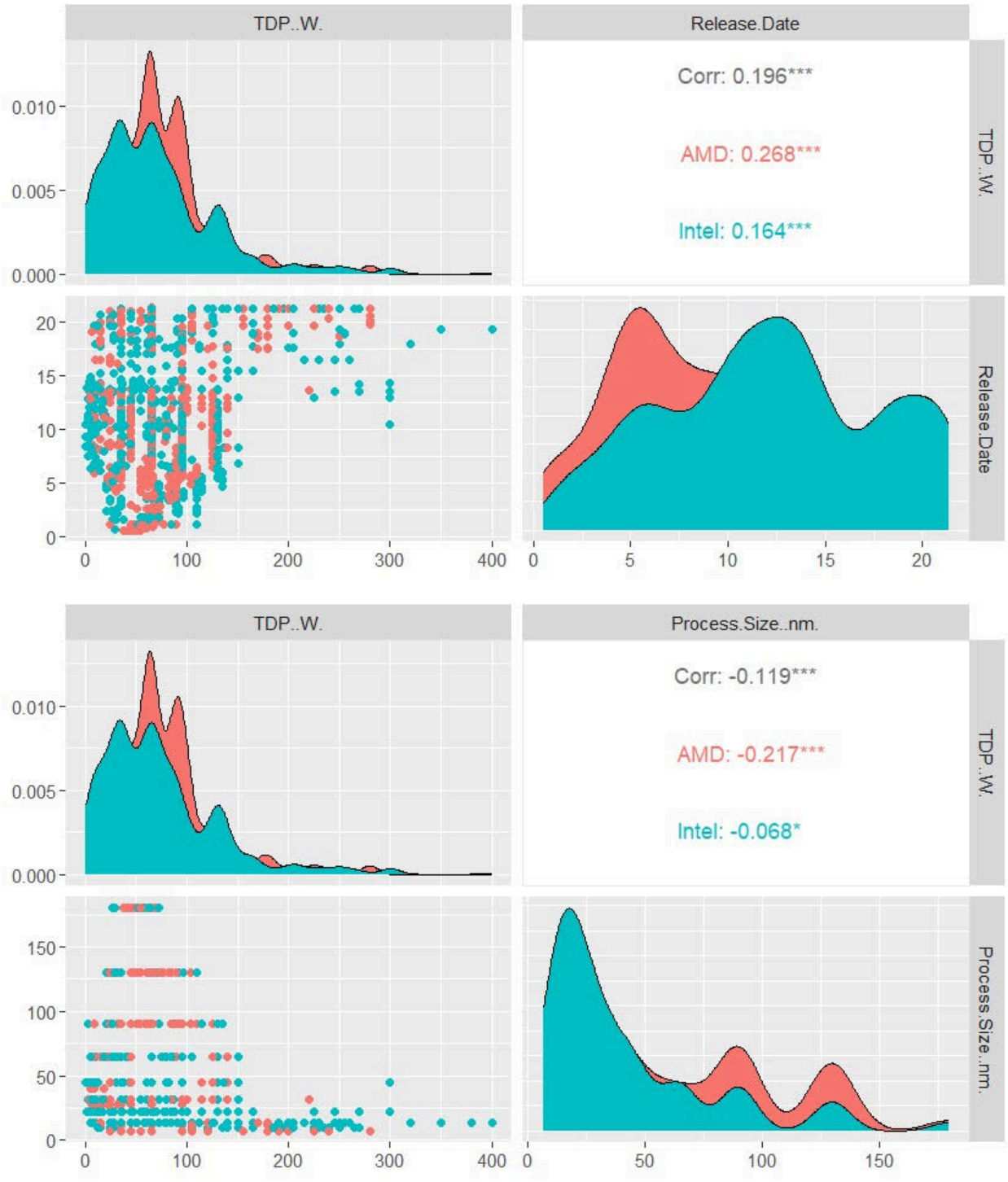
Conclusion

- The size of CPUs is getting smaller.
- Thermal Design Power decreases as the size decreases.

Table/Plots



Randy Li
Zhixuan Huang



R Appendix

```
#load the data
chip_dataset <- read.csv("chip_dataset.csv")
```

```
chip_dataset <- chip_dataset[,-1]
print(chip_dataset)


# Convert the release date
for(i in 1:nrow(chip_dataset)) {
  # Check if 'Na' is in the 'Release Date' column
  if(grepl("NaT", chip_dataset$`Release.Date`[i])) {
    # Replace with NA
    chip_dataset$`Release.Date`[i] <- NA
  } else {
    # Extract year, month, and day, then calculate new value
    date_split <- unlist(strsplit(chip_dataset$`Release.Date`[i], "/"))
    month <- as.numeric(date_split[1]) /12
    day <- as.numeric(date_split[2]) /365
    year <- as.numeric(date_split[3]) -2000
    chip_dataset$`Release.Date`[i] <- year + month + day
    }
  }
#change the release date to numerical
chip_dataset$`Release.Date` <- as.numeric(chip_dataset$`Release.Date`)
print(chip_dataset)

grouped <- split(chip_dataset, chip_dataset$Type)
#Group the data to CPU
CPU <- grouped$CPU
#remove all NA columns
CPU <- CPU[, !(names(CPU) %in% c('FP16.GFLOPS', 'FP32.GFLOPS', 'FP64.GFLOPS'))]


#find all the numerical columns
numerical_columns<- sapply(CPU, is.numeric)
# Verify that "Vendor" is a column name in CPU
print(numerical_columns)

#Extract the vendor columns
vendor_column <- CPU$Vendor
numerical_CPU <- CPU[, numerical_columns]

print(numerical_CPU)

# install the package ggally if not available
if (!require("GGally")) install.packages("GGally")
library(GGally)
```

Randy Li
Zhixuan Huang

```
# Create a pair plot
#plot between TDP and process_size
plot_data <- numerical_CPU[, c('TDP..W.', 'Process.Size..nm.')]
ggpairs(plot_data, mapping = aes(color = vendor_column))
#plot between TDP and release_date
plot_data <- numerical_CPU[, c('TDP..W.', 'Release.Date')]
ggpairs(plot_data, mapping = aes(color = vendor_column))

#plot between process_size and release_date
plot_data <- numerical_CPU[, c('Process.Size..nm.', 'Release.Date')]
ggpairs(plot_data, mapping = aes(color = vendor_column))

#----calculate confidence interval-------
alpha <- 0.05  # 95% confidence interval
#extract process_size from CPU
process_size=numerical_CPU$Process.Size..nm.

mean_process_size <- mean(process_size, na.rm = TRUE)
# Compute standard error
process_std_error <- sd(process_size, na.rm = TRUE) / sqrt(length(na.omit(process_size)))
process_size_t_value <- qt(1 - alpha/2, df = length(na.omit(process_size)) - 1)
conf_interval_process_size <- c(mean_process_size - process_size_t_value *
process_std_error, mean_process_size + process_size_t_value * process_std_error)

print(conf_interval_process_size)

#calculate the frequency confidence interval
#extract frequency from CPU
frequency_size=numerical_CPU$Freq..MHz.
mean_frequency_size <- mean(frequency_size, na.rm = TRUE)
std_error_frequency_size <- sd(frequency_size, na.rm = TRUE) /
sqrt(length(na.omit(frequency_size)))
conf_interval_frequency_size <- mean_frequency_size + c(-1, 1) * qt(0.975,
length(na.omit(frequency_size)) - 1) * std_error_frequency_size
print(conf_interval_frequency_size)
#fit the data
fit <- lm(Freq..MHz. ~ Process.Size..nm., data = chip_dataset)
summary(fit)
```