Characterizing Execution Times on Realistic Programs

Database & Big Data Systems Laboratory,
School of Computer Science and Engineering,
Kyungpook National University,
Young-Kyoon Suh

March 1, 2018

1 Description

This document characterizes execution times measured on several real-world programs with different input sizes. To achieve this characterization, we discuss various histograms of execution times, measured in program time (PT), of the programs throughout this document. In this work we wish to achieve several goals as follows. The first goal is to unravel any structure behind the histograms and present insights into how such structure is formed. Another goal is to build a statistical distribution (or model) fitting in the histograms. From that distribution, we may reach predicting a concrete execution time considering system noise via the model on an arbitrary algorithm with a given input on a real execution environment. As a note, the execution times were measured along with the EMPv5 [1] protocol.

The following section shows histograms for runs on different real-world programs with varying input sizes.

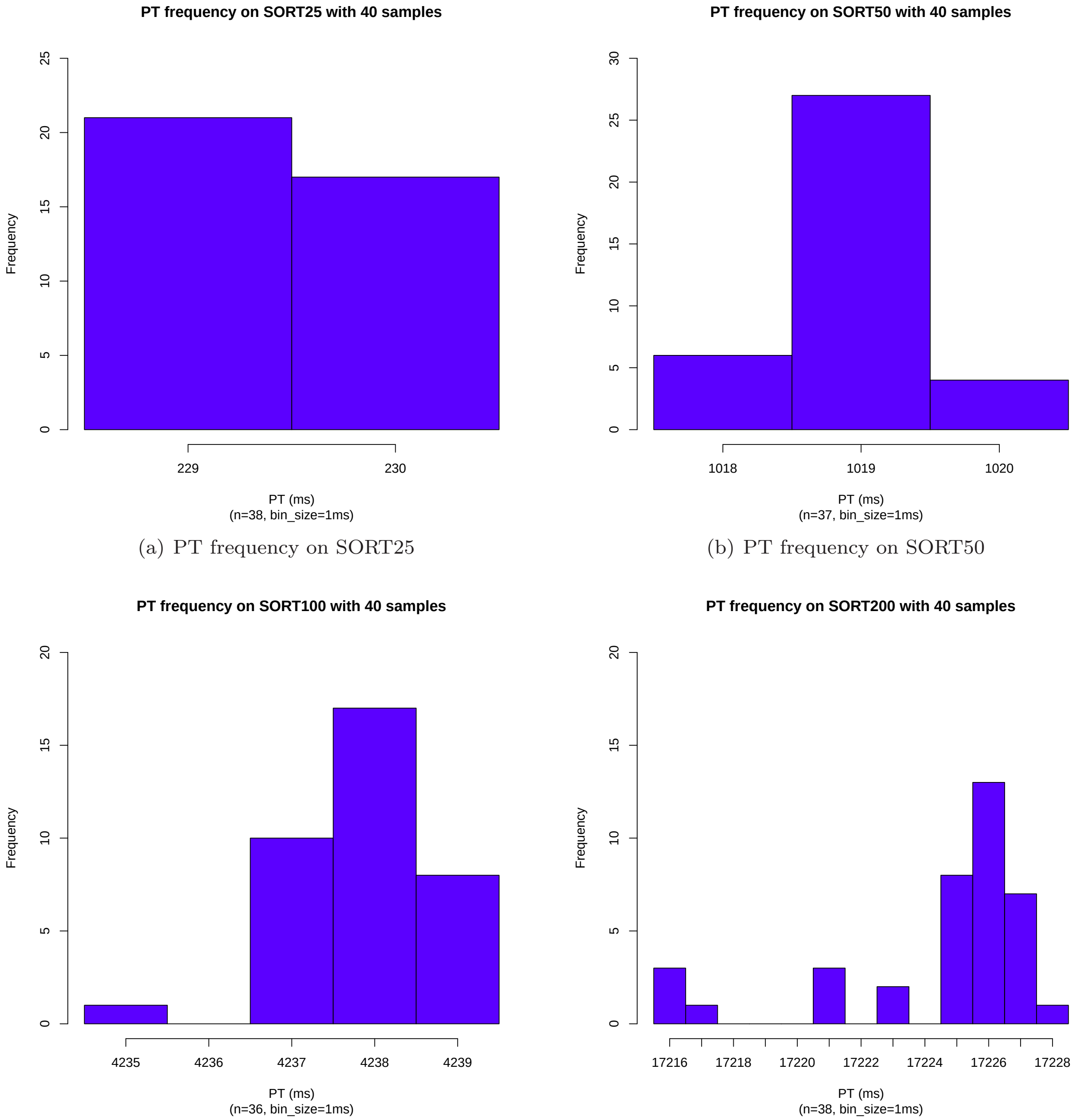
2 Histograms of the Execution Times on Real-World Programs

In this section we present histogram data for two main programs: *insertion sort* and *matrix multiplication*. For the runs of these programs, we varied their input sizes by 2× and measured execution times of the programs over each input size.

2.1 Insertion sort

This section shows a series of histograms of an insertion sort program. The program repeatedly runs 40 times for a given input size. The input size for the program varies from 100,000 to 3,200,000 integer elements, which are randomly generated. Note that each sort program over a specific input size is termed SORTx: for instance, SORT100 indicates the insertion sort program over 100K elements.

Figures 1 and 2 exhibit histograms of the execution times measured on the same insertion sort program as the input size grows from 100K to 800K elements by the steps of 2x. Note that we used one standard deviation in Figure 2(a) as a couple of outliers, which were not eliminated by the original protocol, resulted in disturbing the rendering of a clean distribution.



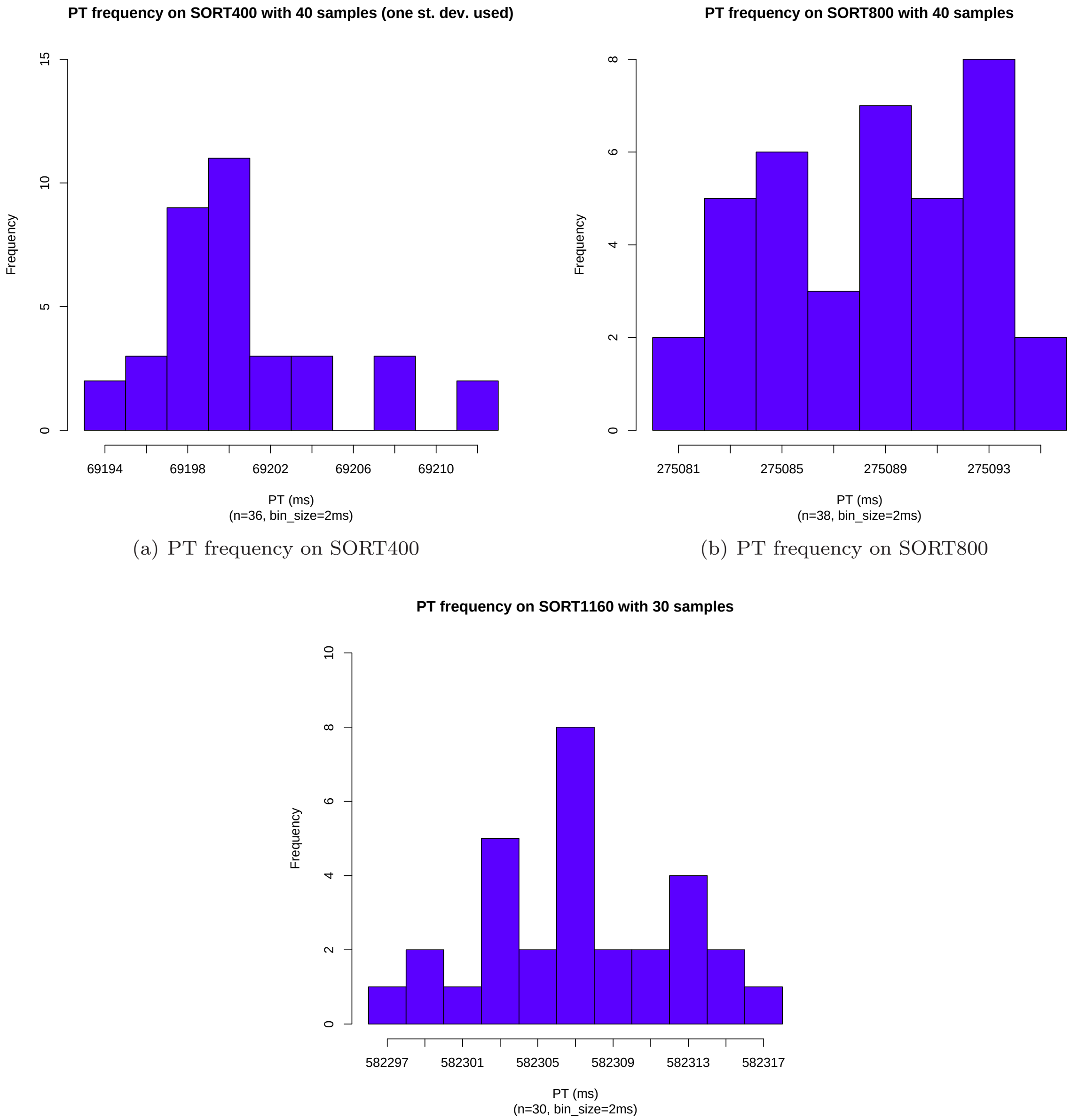
(a) PT frequency on SORT25

(b) PT frequency on SORT50

(c) PT frequency on SORT100

(d) PT frequency on SORT200

Figure 1: PT Histograms of SORT25 ... SORT200

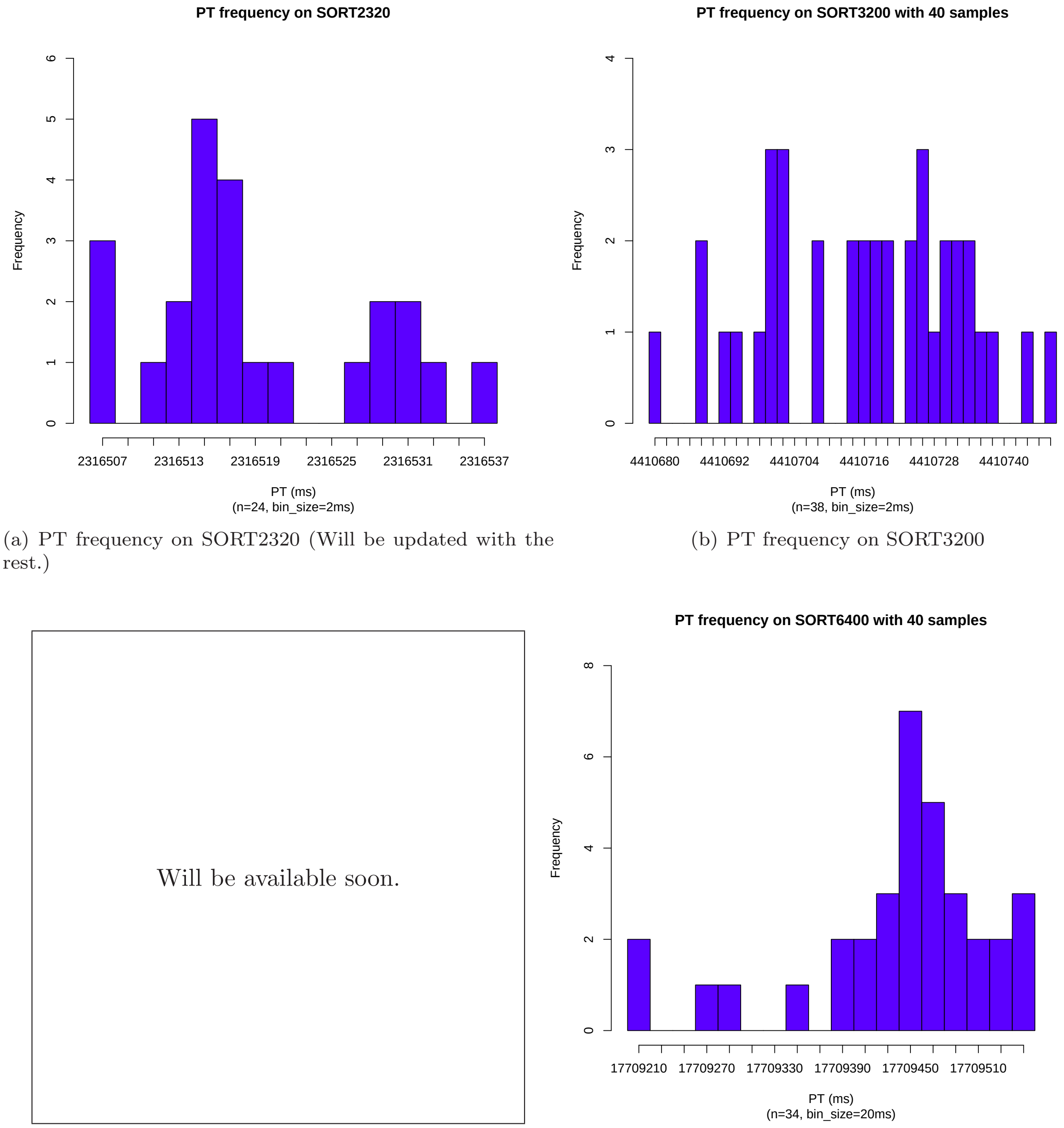


(a) PT frequency on SORT400

(b) PT frequency on SORT800

(c) PT frequency on SORT1160

Figure 2: PT Histograms of SORT400 ... SORT1160



(a) PT frequency on SORT2320 (Will be updated with the rest.)

(b) PT frequency on SORT3200

(c) PT frequency on SORT4640

(d) PT frequency on SORT6400

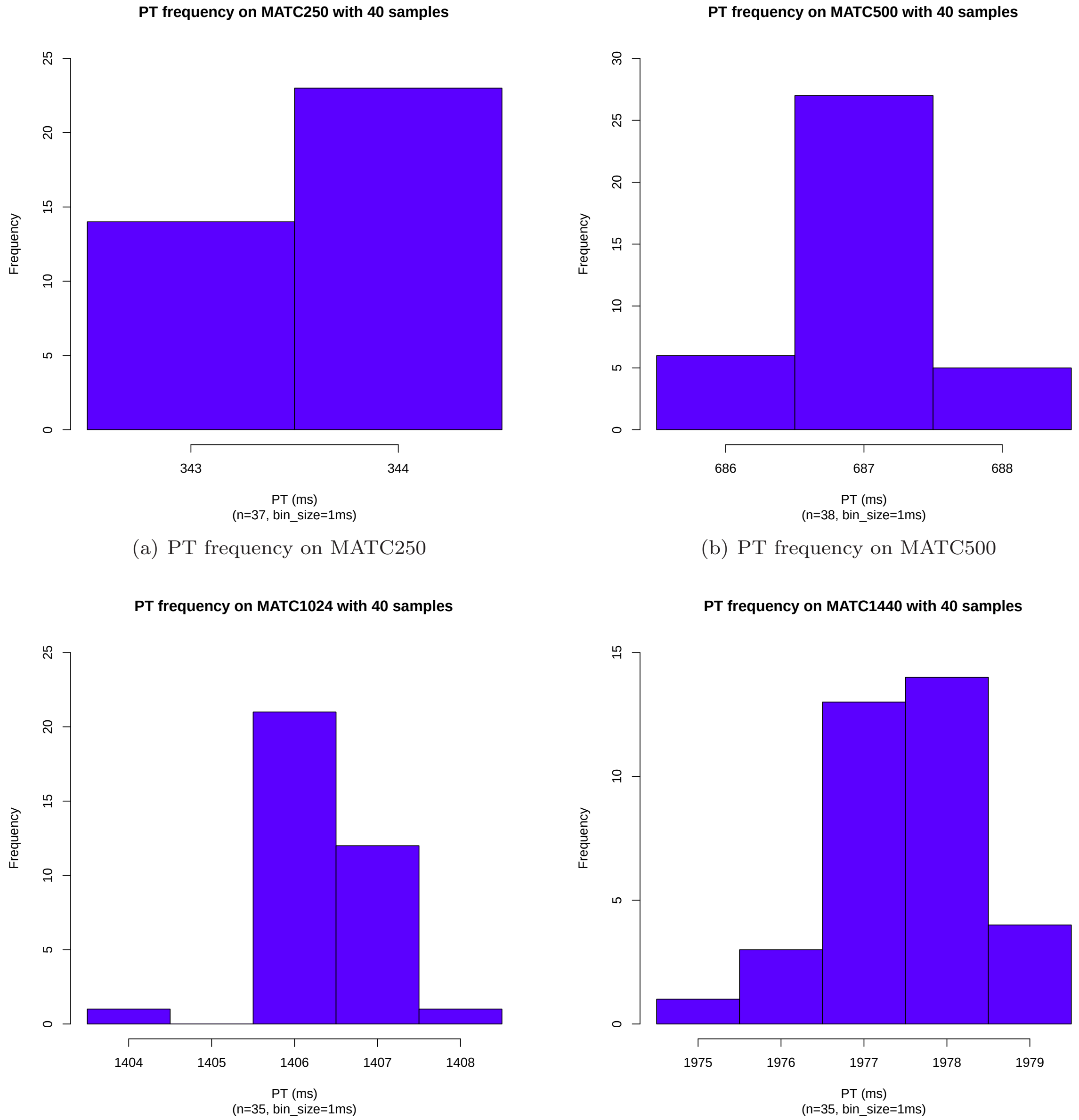
Figure 3: PT Histograms of SORT2320 ... SORT6400

2.2 Matrix Multiplication

This section shows a series of histograms of the execution times of an matrix multiplication program. We used the same sample size for each input size of this program: i.e., 40 iterations. For simplicity, we used two square matrices for performing their multiplication in the program. We also varied the input sizes of each of the two matrices: from 1K×1K to 8K×8K integer elements that are also randomly generated. Note that each matrix multiplication program for a specific size is called MAT*xyyyy*, where *x* indicates which major, specifically *column* vs. *row*, is used, and *yyyy*, how large a given matrix is. For instance, MATC1000 represents a matrix multiplication program in column major over two square matrices having 1,000 integer (random) elements in a row (and a column).

2.2.1 Column Major

Figure 4 shows a series of histograms of the execution times measured on the same matrix multiplication program in column major as the input sizes grows.



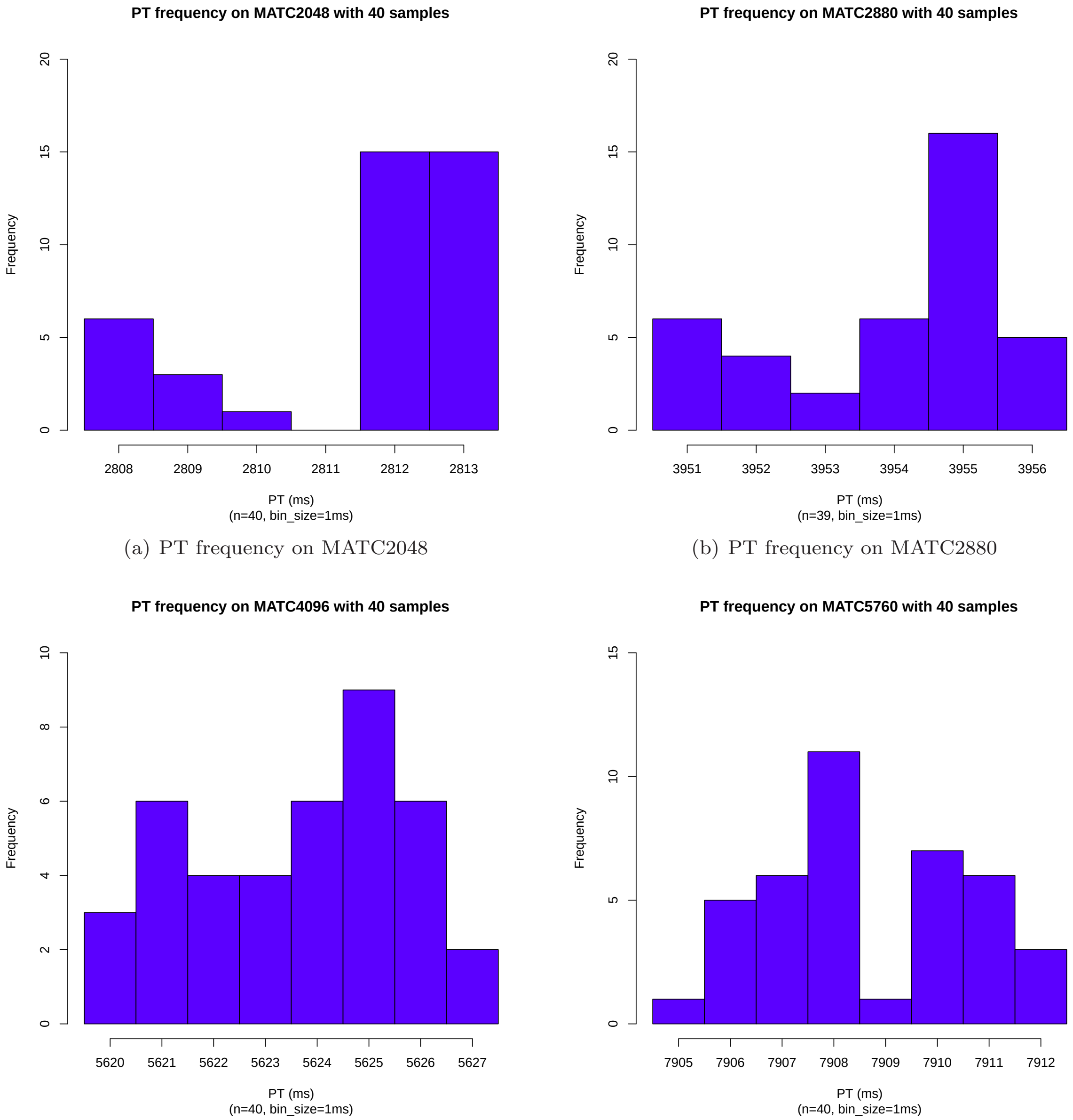
(a) PT frequency on MATC250

(b) PT frequency on MATC500

(c) PT frequency on MATC1024

(d) PT frequency on MATC1440

Figure 4: PT Histograms of MATC250 ... MATC1440



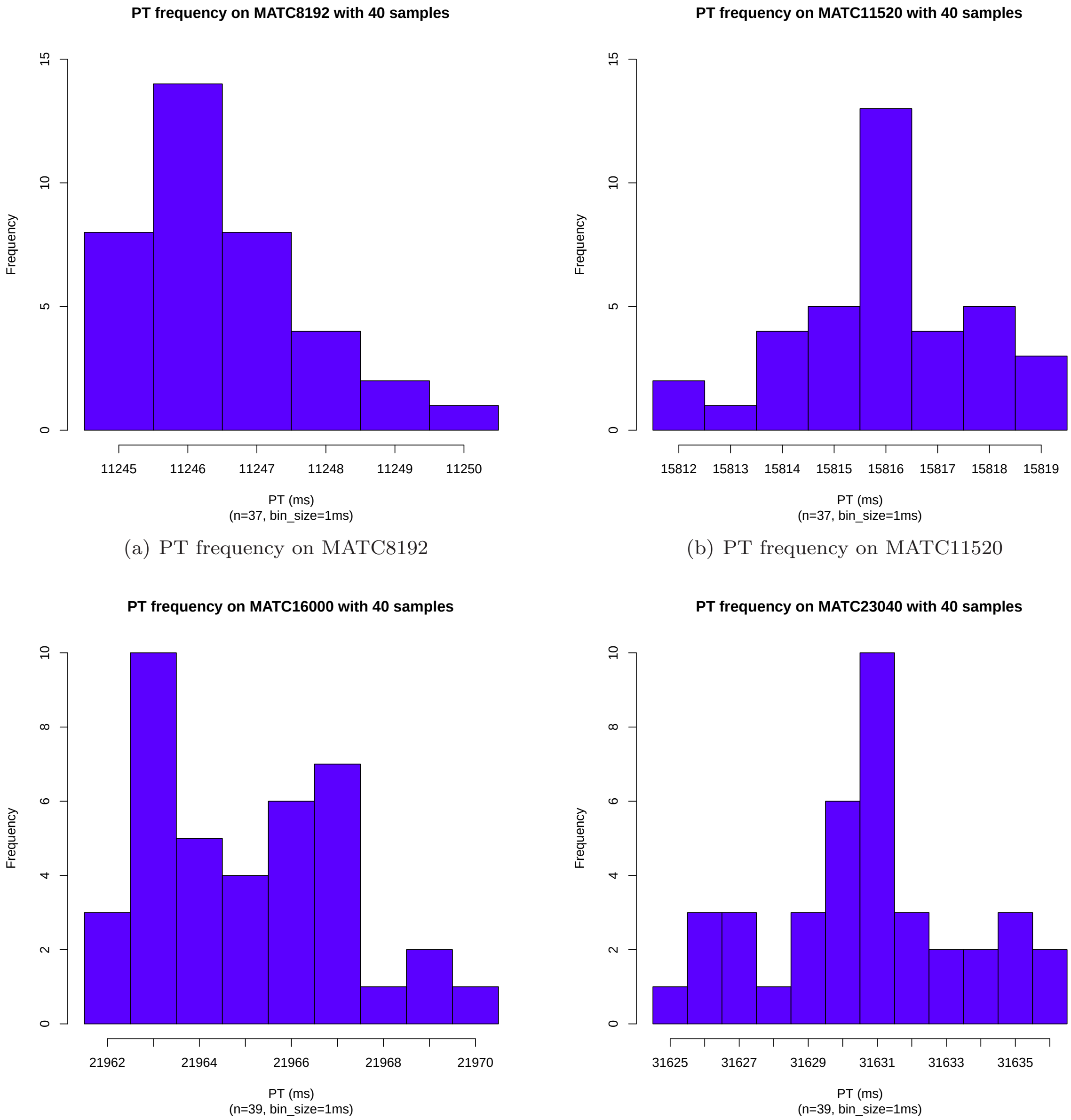
(a) PT frequency on MATC2048

(b) PT frequency on MATC2880

(c) PT frequency on MATC4096

(d) PT frequency on MATC5760

Figure 5: PT Histograms of MATC2048 ... MATC5760



(a) PT frequency on MATC8192

(b) PT frequency on MATC11520

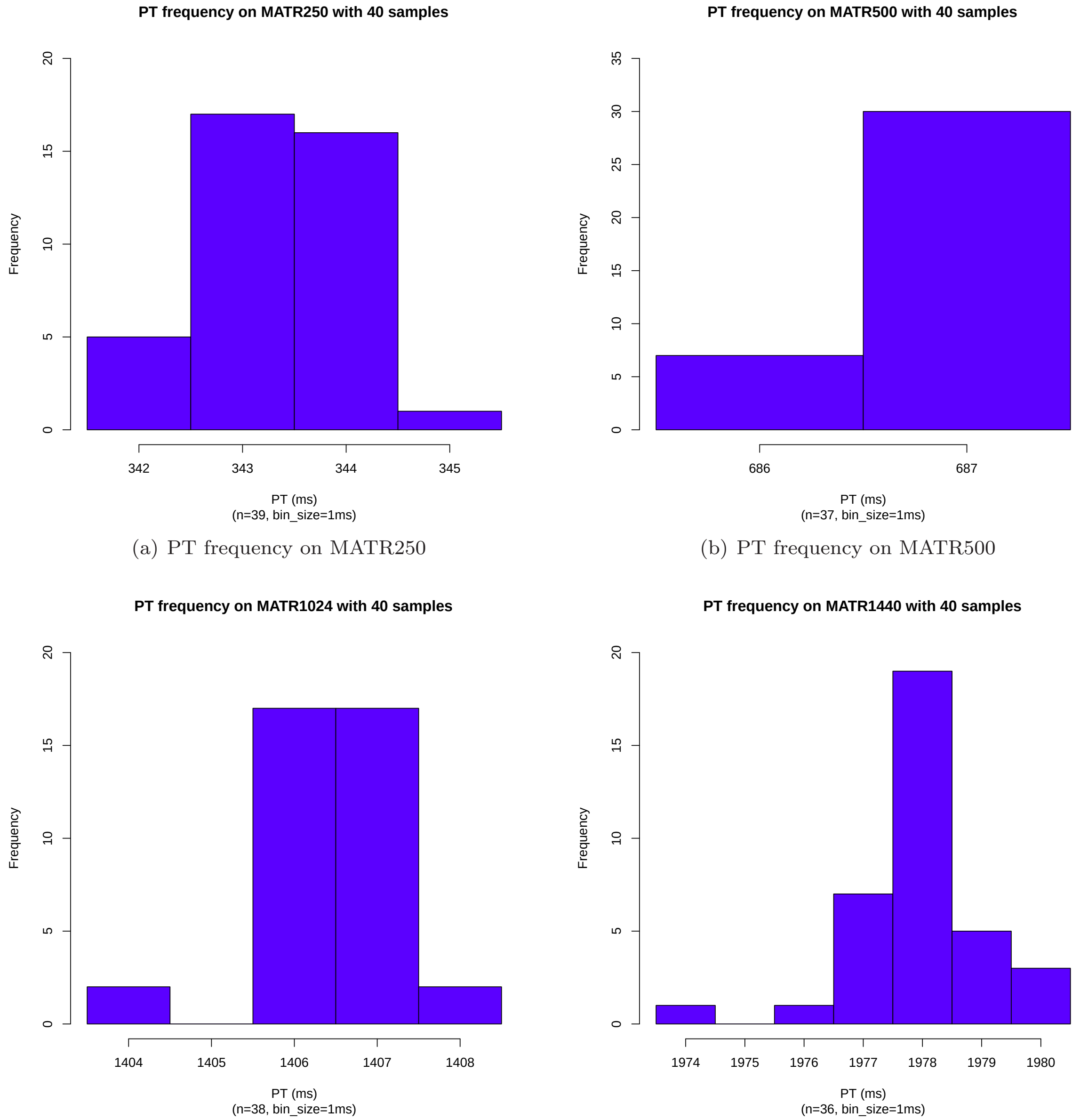
(c) PT frequency on MATC16000

(d) PT frequency on MATC23040

Figure 6: PT Histograms of MATC8192 ... MATC23040

2.2.2 Row Major

Figure 4 shows a series of histograms of the execution times measured on the same matrix multiplication program in row major as the input sizes grows.



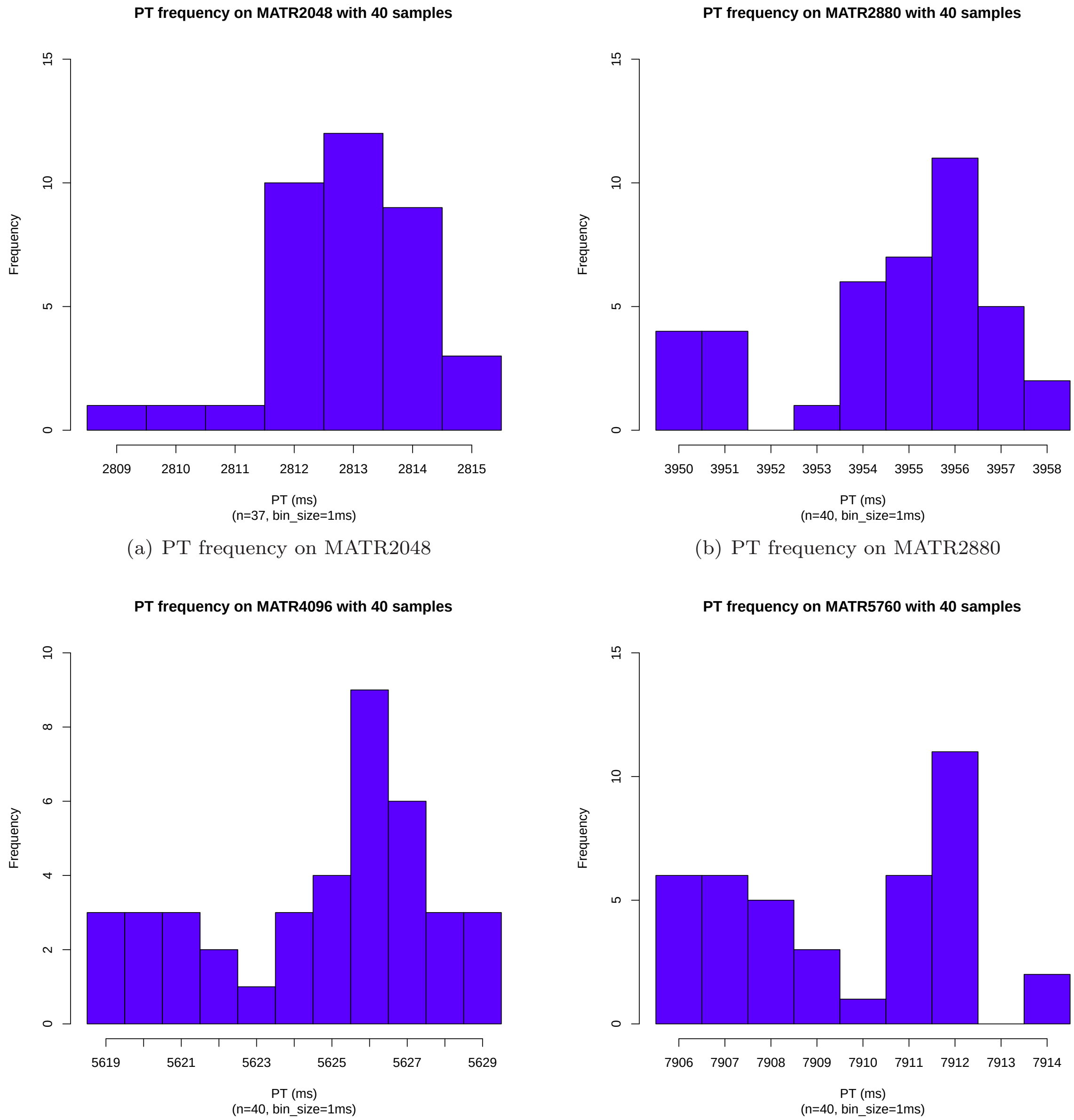
(a) PT frequency on MATR250

(b) PT frequency on MATR500

(c) PT frequency on MATR1024

(d) PT frequency on MATR1440

Figure 7: PT Histograms of MATR250 ... MATR1440



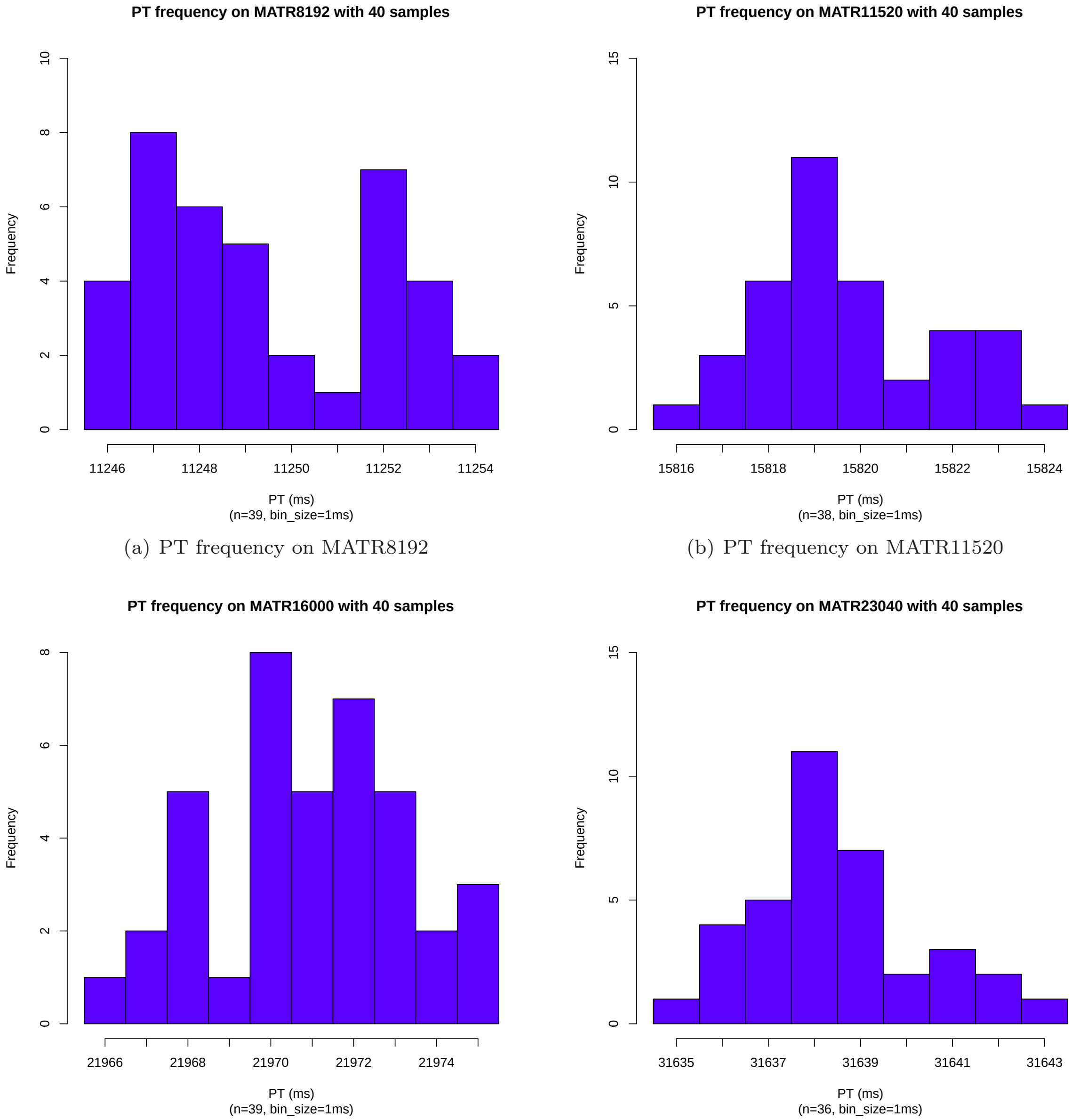
(a) PT frequency on MATR2048

(b) PT frequency on MATR2880

(c) PT frequency on MATR4096

(d) PT frequency on MATR5760

Figure 8: PT Histograms of MATR2048 ... MATR5760



(a) PT frequency on MATR8192

(b) PT frequency on MATR11520

(c) PT frequency on MATR16000

(d) PT frequency on MATR23040

Figure 9: PT Histograms of MATR8192 ... MATR23040

References

[1] Young-Kyoon Suh, Richard T. Snodgrass, John Kececioglu, Peter J. Downey, Rob S. Maier, and Cheng Yi, “EMP: Execution Time Measurement Protocol for Compute-Bound Programs”, in *Software: Practice and Experience*, 47(4):559–597, 2017.

[2] Sabah Currim,Richard T. Snodgrass, Young-Kyoon Suh, and Rui Zhang, “DBMS Metrology: Measuring Query Time”, in *ACM Transactions on Database Systems*, 42(1):3:1–42(+8), 2017.