Characterizing Execution Times on Realistic Programs

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

March 8, 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.

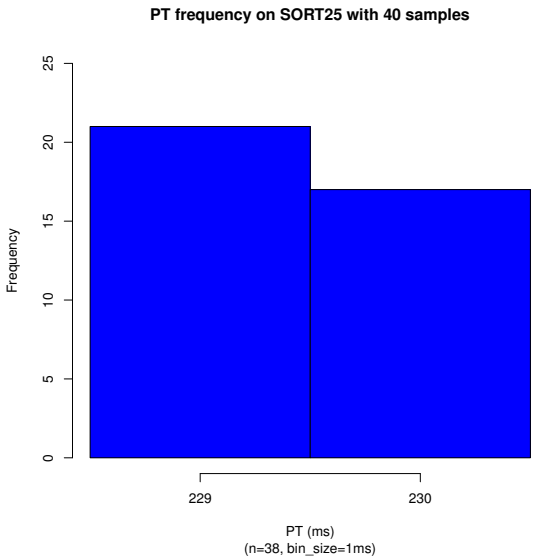
2 Histograms of the Execution Times on Real-World Programs

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.

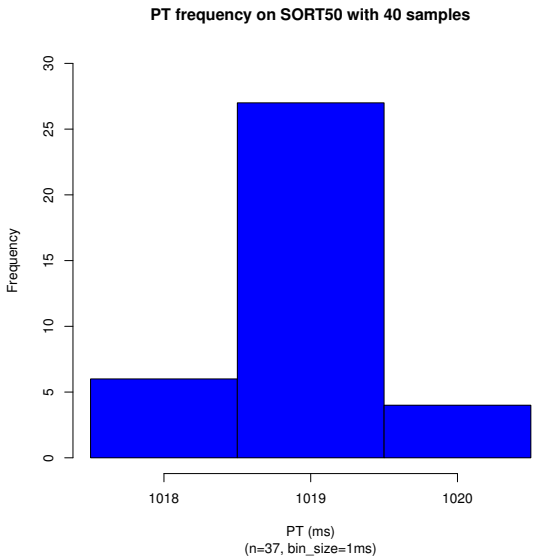
2.1 Insertion sort

This section shows a series of histograms of an insertion sorting program that sorts the elements of a given array in non-decreasing order. The program repeatedly runs 300 times for a given input size. The input size for the program varies from 100,000 to 1,160,000 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.

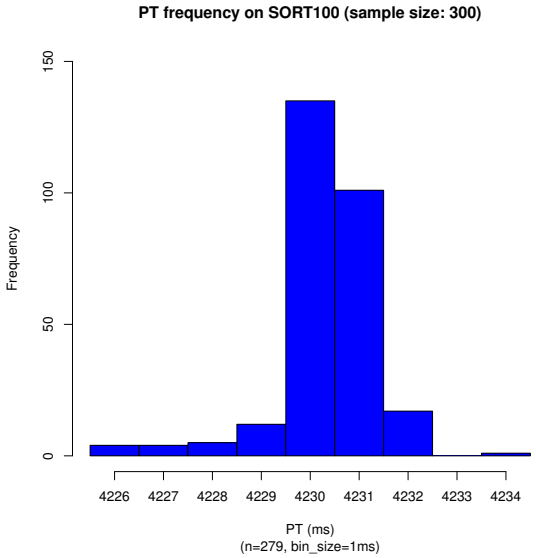
Figures 1 and 2 exhibit histograms of the execution times measured on the same insertion sort program as the input size grows from 100K to 1,160K elements.Note that we used one standard deviation in Figure 2(a) as a couple of outliers, which were not eliminated by the original protocol, resulted in disturbing the rendering of a clean distribution.



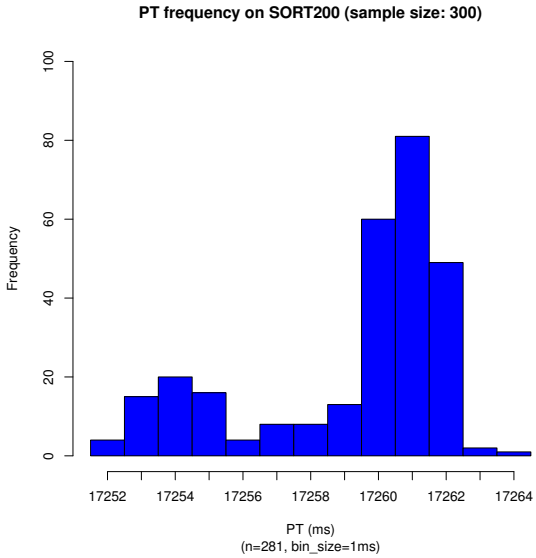
(a) PT frequency on SORT25



(b) PT frequency on SORT50

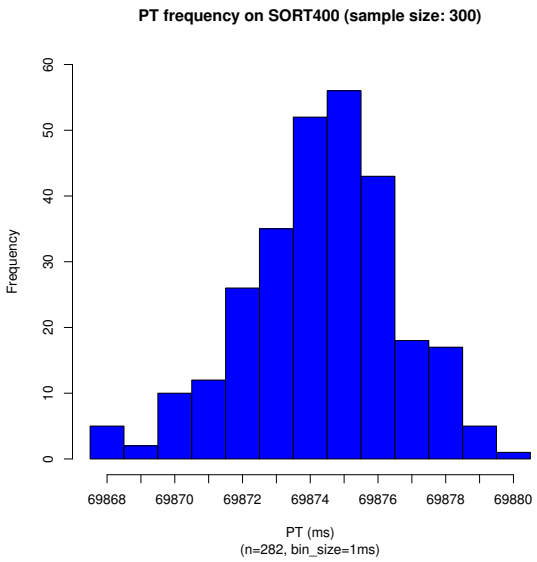


(c) PT frequency on SORT100

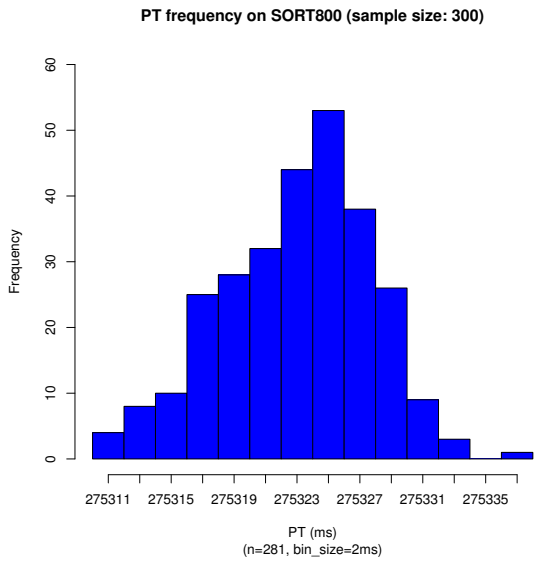


(d) PT frequency on SORT200

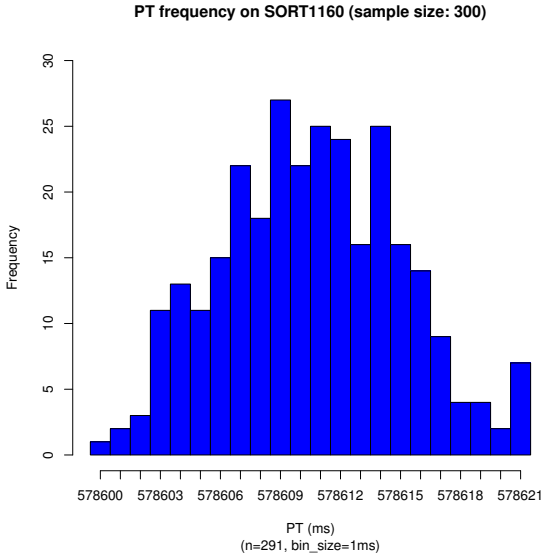
Figure 1: PT Histograms of SORT25 ... SORT200



(a) PT frequency on SORT400



(b) PT frequency on SORT800



(c) PT frequency on SORT1160

Figure 2: PT Histograms of SORT400 ... SORT1160

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.