

CSC 263

Kamil Ahmed Aslam

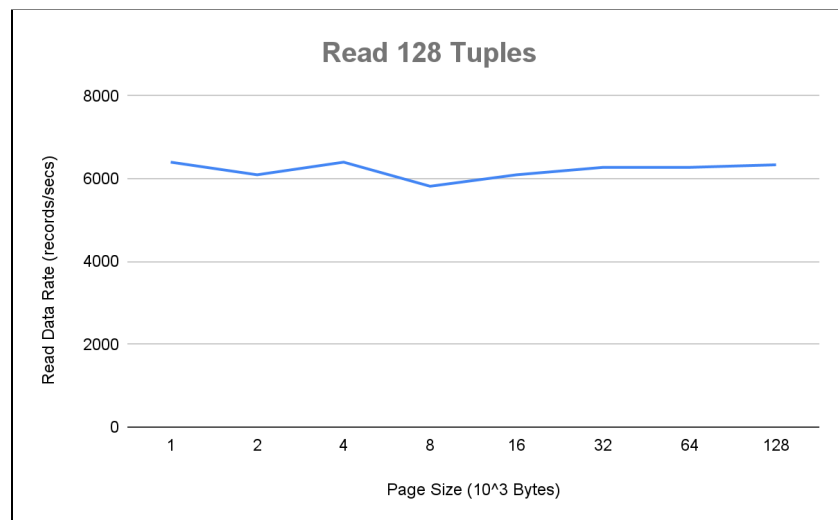
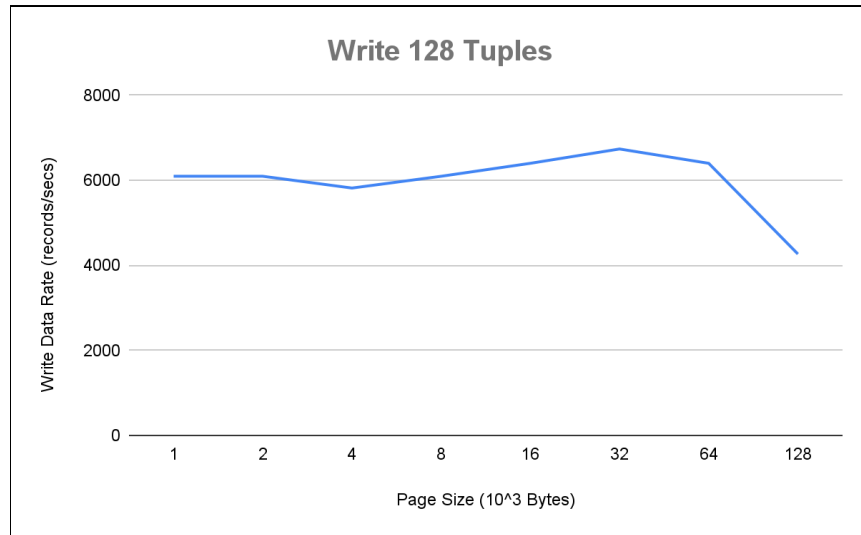
We perform our tests for 128 tuples (see csvGenerator.py).

Experiment 2.3:

We are given a standard for `Record record` that has 100 attributes and 10 bytes per attribute. Thus we need 1000 bytes to serialize each record. We initialized a record with 100 attributes where each attribute is of size 10.

Our method `fixed_len_sizeof(Record *record)` returns the same answer when passed with `record` as the parameter.

Experiment 3.2:



We conclude that reading from page based format is quicker than reading from csv files as observed by the plots. Each reading for the time was taken 10 times and then averaged to ensure accuracy and validity of the results.

In blocked disk I/O, we store our empty slots so we don't have to traverse through pages unnecessarily. However, in our model, we have to go through all our pages if there is an insertion to be made which increases the insertion time complexity.

We see that page based format or reading from file as opposed to csv is because we have to parse through more data in the latter in order to read all of our records. Since csv is a comma separated file, there is an abundance of delimiters because of which the program doesn't scale

well as the number of tuples increase. Additionally, page files can store data in binary while csv files have to store it in strings. Deletion of records is also more expensive for csv files because the file has to be reorganized after each deletion.

One of the shortcomings of our page organization is that we don't keep track of empty slots. We know how many of them there are but we have to parse again through our pages to find the slots. We could implement a bitmap and update it every time an empty slot is taken.