Rebuilding the tables

Yehuda Michelson

July 3, 2022

1 Introduction

In this document, I intend to describe the code I wrote, It's purpose is to rebuild a shuffled array of balls, which is part of our implementation of the oblivious RAM.

1.1 Overview

The code is composed of three stages:

1. Balls into bins.
2. Select & move the secret load.
3. Tight compaction on the secret Load

1.2 inputs and configurations

- Input: $DATA\_LOCATION$ - The path to the original balls.
- N - The amount of balls.
- $BALL\_SIZE$ - The size of a ball in bytes.
- μ - The average amount of balls per bin.
- $NUMBER\_OF\_BINS$ (also referred to as B) - the amount of bins N/μ.
- BIN_SIZE - the max size of a bin 2μ
- ε - The fraction of balls to be moved to the overflow pile $\frac{1}{\log \lambda}$.

2 Balls into bins

In this stage the code goes over the balls from the original array.

- 2μ balls at a time, to reduce round trips. this is done because our code is assumed to have approximately 2μ size of local memory
- The code determines to which bin each ball is thrown using a AES pseudo random function.
- The first ball in each bin contains the capacity of the bin (meaning how much of the bin is currently full) and is updated with every bulk insertion.

3 Select & move the secret load

In this stage the code goes over the bins from the previous stage.

- It goes over $1/\varepsilon$ bins at a time, to reduce round trips. This is done because our code is assumed to have approximately 2μ size of local memory, and from each bin we read $2\mu\varepsilon$ as explained in the next item.
- from each bin - we read the capacity-ball and then we read the $2\mu\varepsilon$ top most balls of the bin. This is done because as explained in a previous document - the secret load would be there with high likelihood.
- For each bin, the code selects a secret threshold and copies only the balls above the threshold.
- from the $1/\varepsilon$ bins, the code takes the copied balls, and adds up enough dummies at the end so that there are 2μ balls. Then the code writes the already tightly compacted bin to the overflow pile.

4 Tight compaction on the secret Load

Currently we have in the overflow pile εB bins which are already tightly compacted. We assume a mixed stripe along the middle and do a tight compaction on it (the likelihood of this mixed stripe still needs to be proven)

5 Results

- $N = 2^{20}$
- $BALL\_SIZE = 16$ bytes
- $\mu = 30 \log^3 \lambda = 21870$
- $NUMBER\_OF\_BINS = 47$
- $\varepsilon = \frac{1}{9}$

stage	RT-write	RT-read	Ball-write	Ball-read
balls into bins	24	48	1050888	1050888
move secret load	6	12	262440	262494
tight compaction	3	3	131220	131220
overall	33	63	1444548	1444602