# Memory Bus Bit Transition Statistics for 500-perlbench-r-1–6942989

December 31, 2018

## 1 Scalar Statistics

**Elapsed time**: 39201.1
**Cache size**: 16777216 B
**Associativity**: 8 ways
**Line size**: 64 B
**DRAM bus width**: 8 B
**Instructions cache simulation**: on
**LLC Load Miss Count**: 6047759
**LLC Load Hit Count**: 1589304694976
**LLC Load Miss Ratio**: 0.000380527%
**LLC Store Miss Count**: 3823226
**LLC Store Hit Count**: 219517893794
**LLC Store Evict Count**: 5714373
**LLC Store Miss Ratio**: 0.00174162%
**LLC Total Miss Count**: 9.87098e+06
**LLC Total Hit Count**: 1.80882e+12
**LLC Total Miss Ratio**: 0.00054571%
**Total number of bit transitions**: 1283127649
**Bit entropy**: 0.18377
**Cache line utilization ratio**: 0.633052

## 2 Graphs

The graphs give the data either **transfer-wise** or **bus-wise**. Bus-wise refers to data sent within the same transfer (i.e. as part of the same bus transfer,

as defined by bus-width). Transfer-wise refers to data from different bus transfers, which cause the bit transitions to occur.

## 2.1 Distribution of all byte values transmitted over the bus

Figure 1 and Figure 2 give the distribution of all transmitted bytes, either through a memory read or a memory write. The first one (pareto chart) is in terms of total number of occurrences, while at the same time showing the accumulation in the total percentage of transmitted byte values. The second one displays the absolute percentage of the most popular values transferred over the bus.

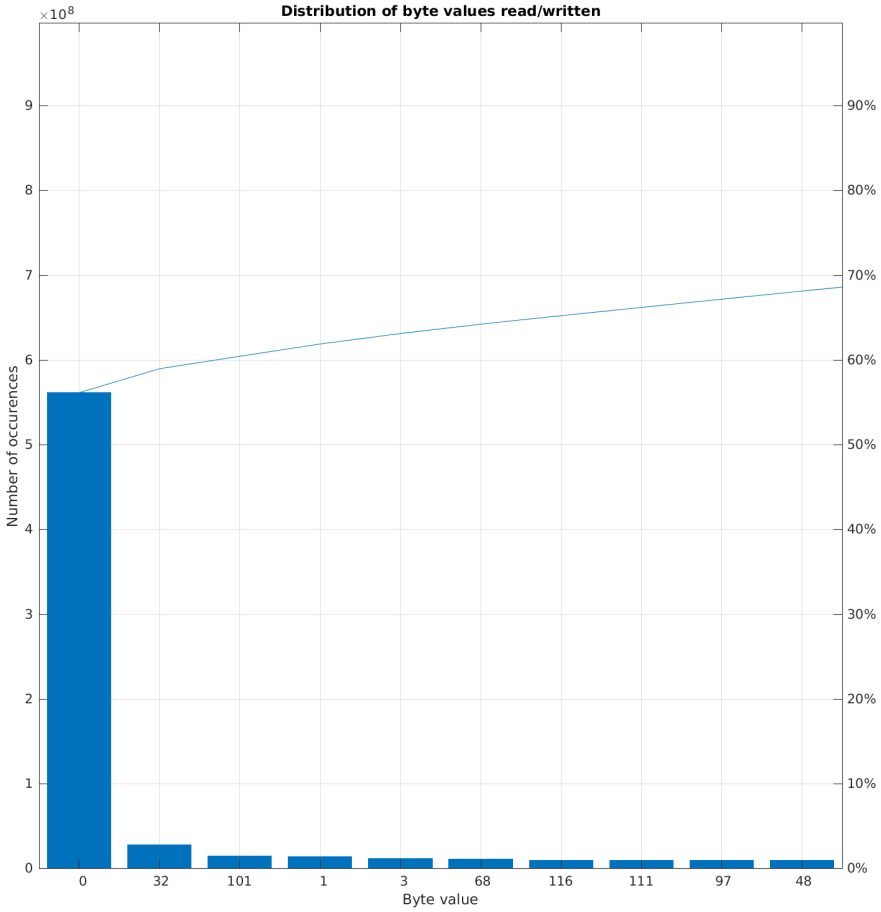


Figure 1: Distribution of all bytes transmitted over the bus - pareto chart

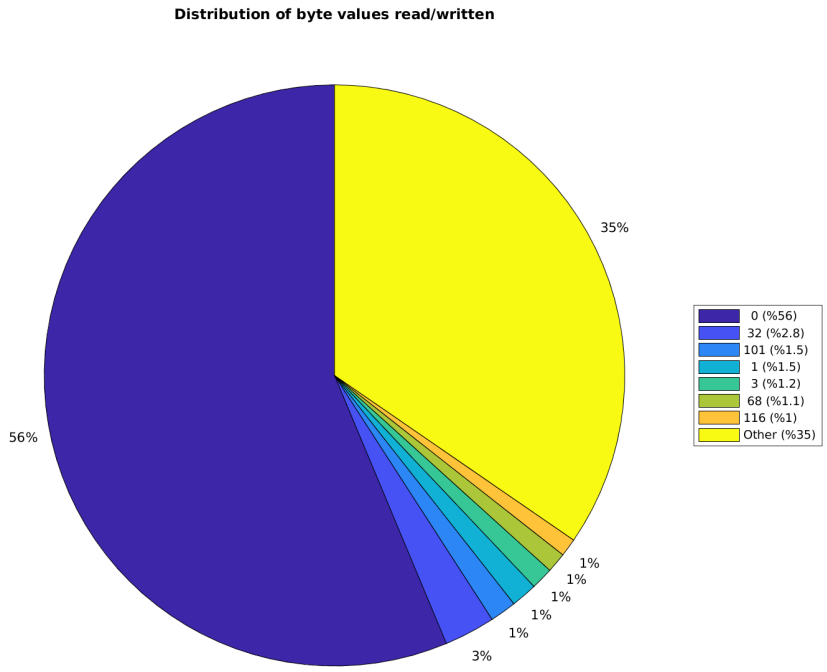


Figure 2: Distribution of all bytes transmitted over the bus - pie chart

## 2.2 Distribution of transitions (transfer-wise)

### 2.2.1 All transfers (transfer-wise)

Figure 3 illustrates the distribution of all transitions. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

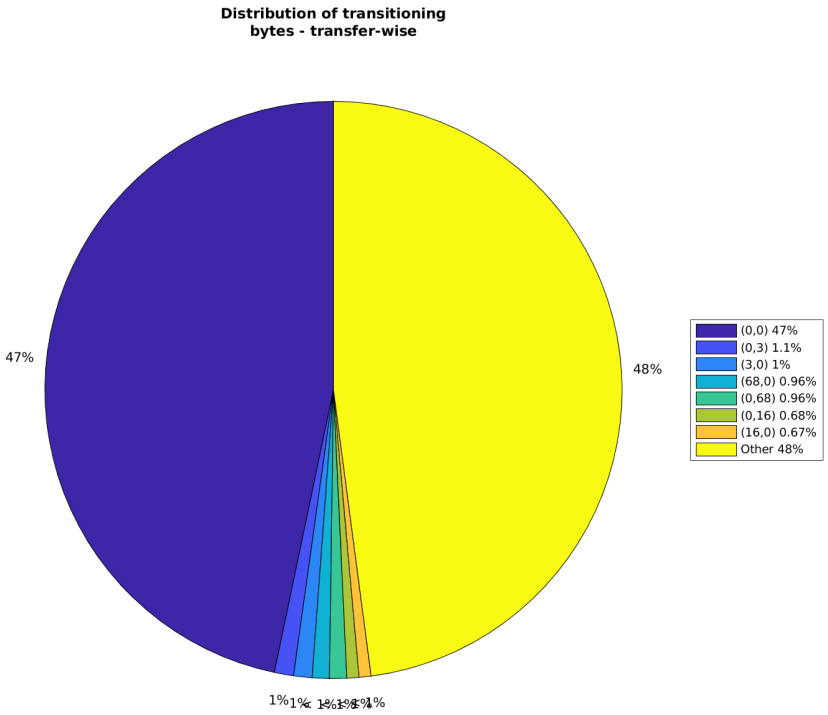


Figure 3: Distribution of transitioning bytes - transfer-wise

### 2.2.2 Only differing byte transfers (hamming distance $> 0$)(transfer-wise)

Figure 4 illustrates the distribution of transitions, filtered only to show differing byte transitions. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

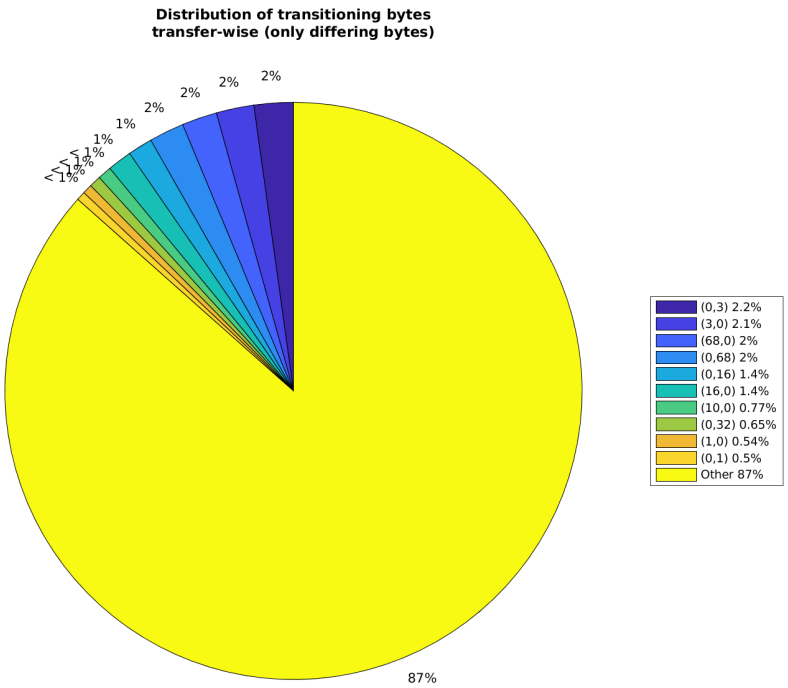


Figure 4: Distribution of transitioning bytes - transfer-wise (only differing bytes)

### 2.2.3 Only same byte transfers (hamming distance = 0)(transfer-wise)

Figure 5 illustrates the distribution of transitions, filtered only to show non-changing byte transfers. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

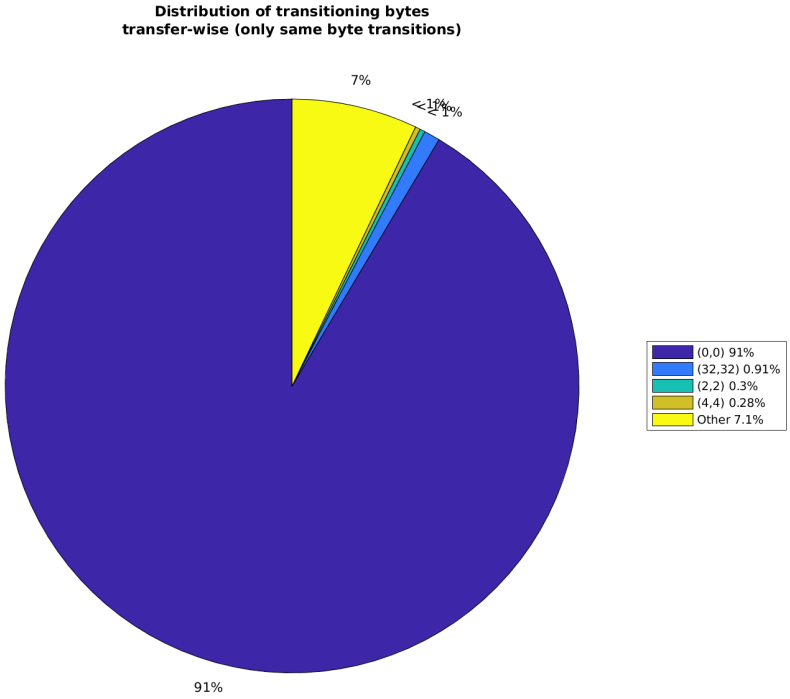


Figure 5: Distribution of transitioning bytes - transfer-wise (only same byte transitions)

### 2.2.4 Number of consecutive "n" zeros encountered (transfer-wise)

Figure 6 illustrates the distribution of consecutive "n" zeros encountered, transfer-wise. The x axis is the "n" value, and the y axis is the frequency.

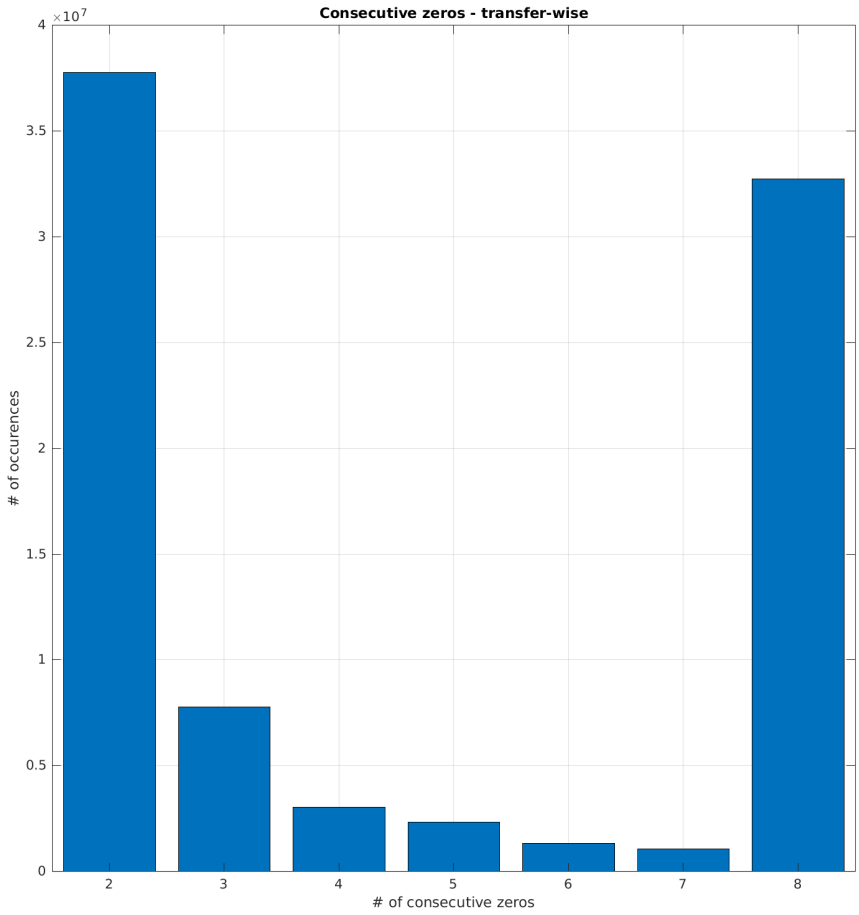


Figure 6: Number of consecutive `0` values encountered - transfer-wise

## 2.3 Distribution of transitions (bus-wise)

### 2.3.1 All transfers (bus-wise)

Figure 7 illustrates the distribution of all transitions happened during a whole benchmarking, bus-wise. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

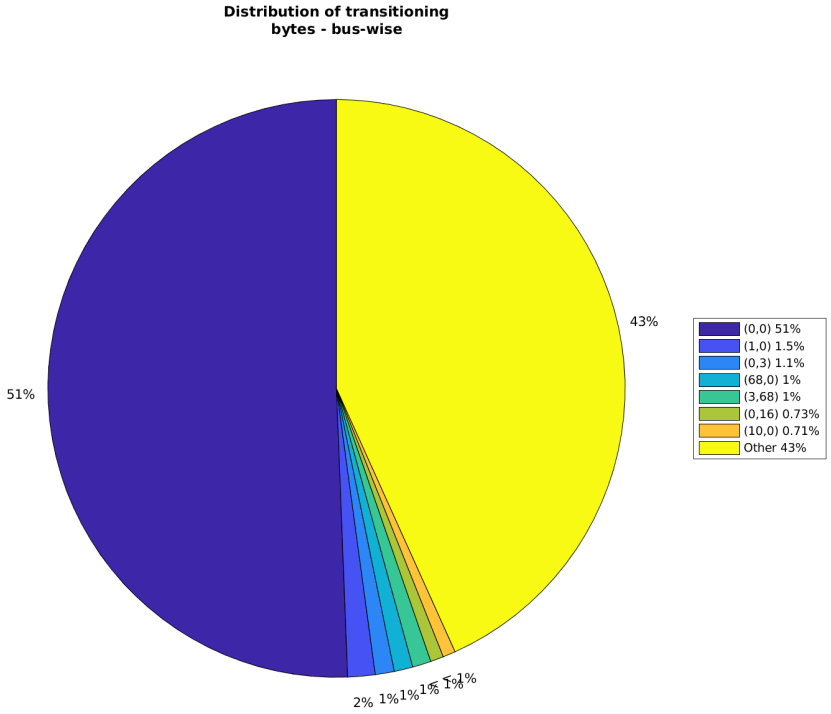


Figure 7: Distribution of transitioning bytes - bus-wise

### 2.3.2 Only differing byte transfers (hamming distance $> 0$)(bus-wise)

Figure 8 illustrates the distribution of transitions happened during the whole benchmarking. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

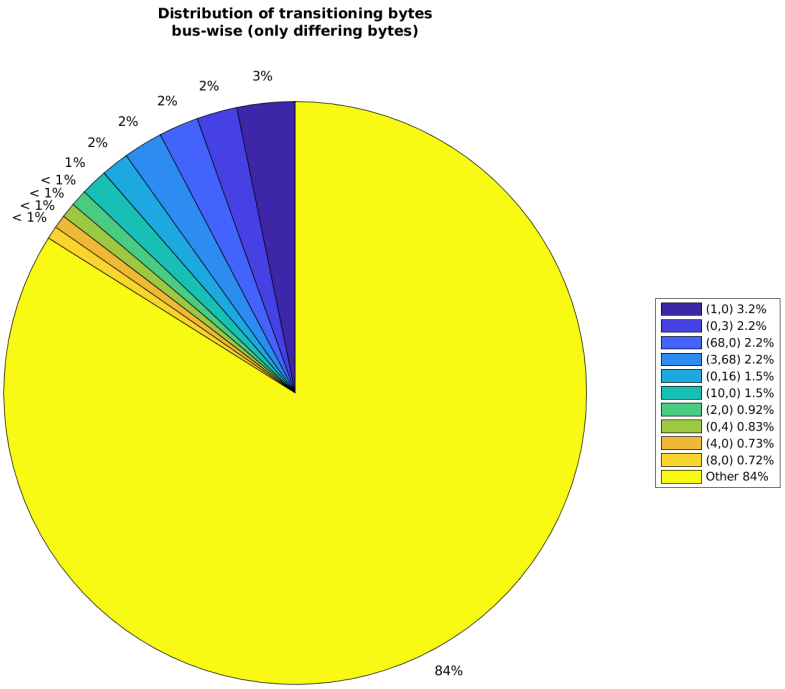


Figure 8: Distribution of transitioning bytes - bus-wise (only differing bytes)

### 2.3.3 Only same byte transfers (hamming distance = 0)(bus-wise)

Figure 9 illustrates the distribution of transitions, filtered only to show non-changing byte transfers. Tuples $(n, n)$ have been used as $(old\_value, new\_value)$ to show the transition.

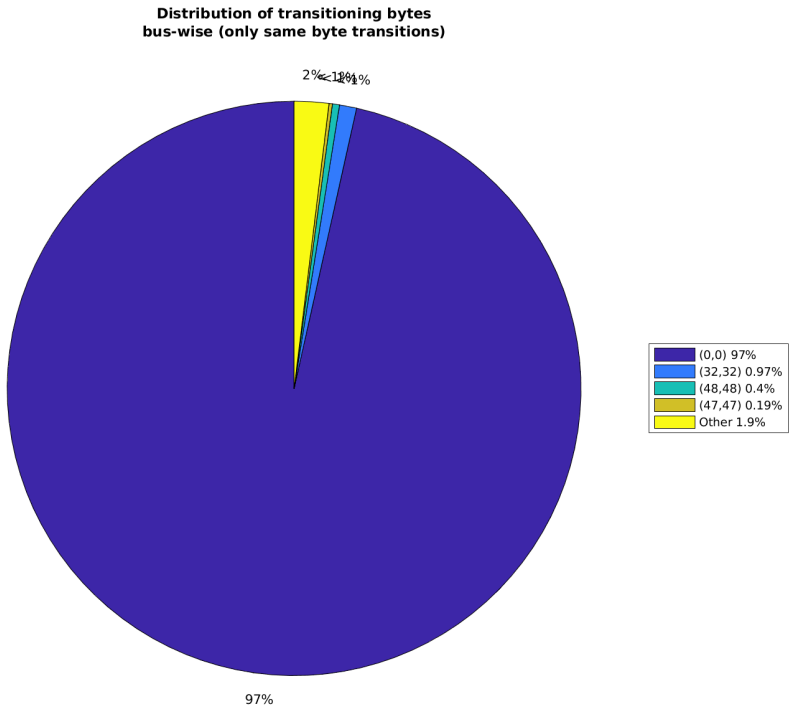


Figure 9: Distribution of transitioning bytes - bus-wise (only same byte transitions)

### 2.3.4 Number of consecutive “n” zeros encountered (bus-wise)

Figure 6 illustrates the distribution of consecutive “n” zeros encountered, transfer-wise. The x axis is the “n” value, and the y axis is the frequency.

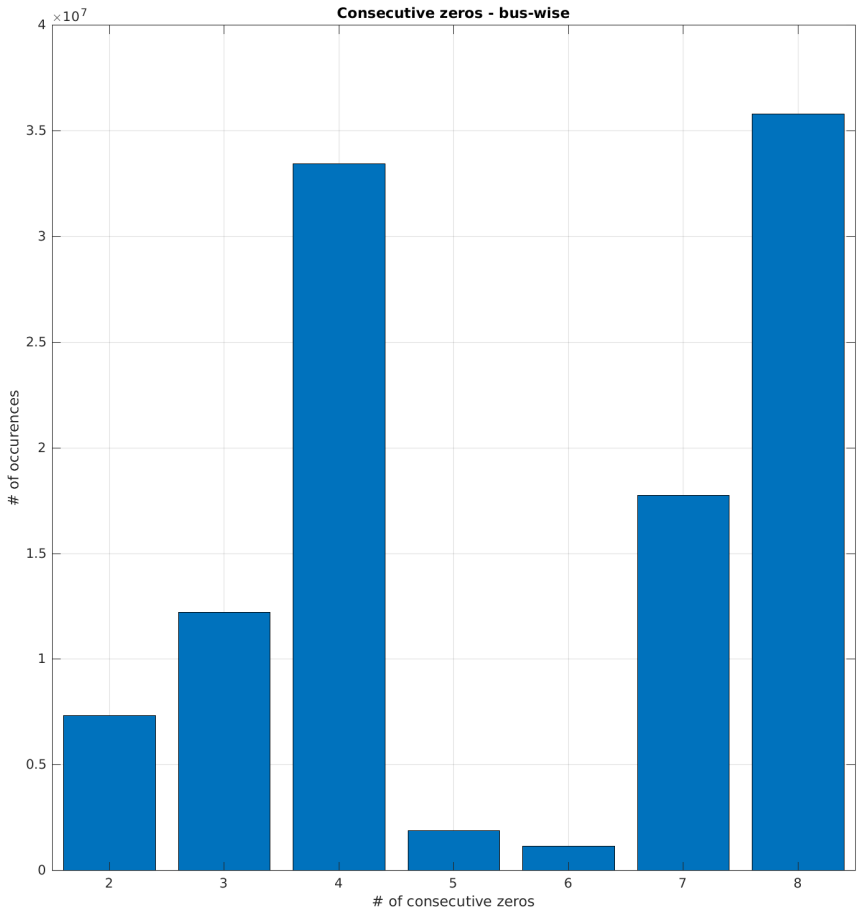


Figure 10: Number of consecutive `0` values encountered - bus-wise

## 2.4 Byte reuse ratios at eviction

Figure 11 and Figure 12 give the reuse ratios of the bytes for each byte value. Note that the byte values are the last values the cacheline contained **at eviction**, not the original byte values brought in to the cache. Also note that these numbers do not give an idea on *how much* the cacheline was reused. The values in the cacheline are marked as reused as soon as the value is read/written to after the initial access.
For the overall cacheline reuse statistic, see Section 1.

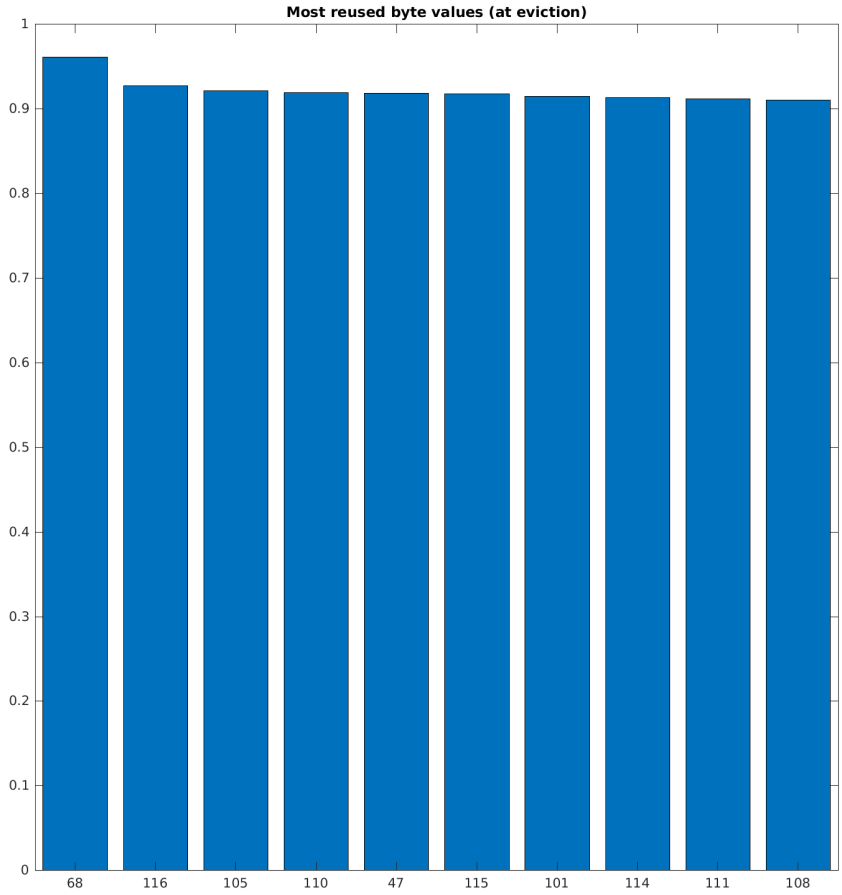


Figure 11: Most reused byte values at eviction

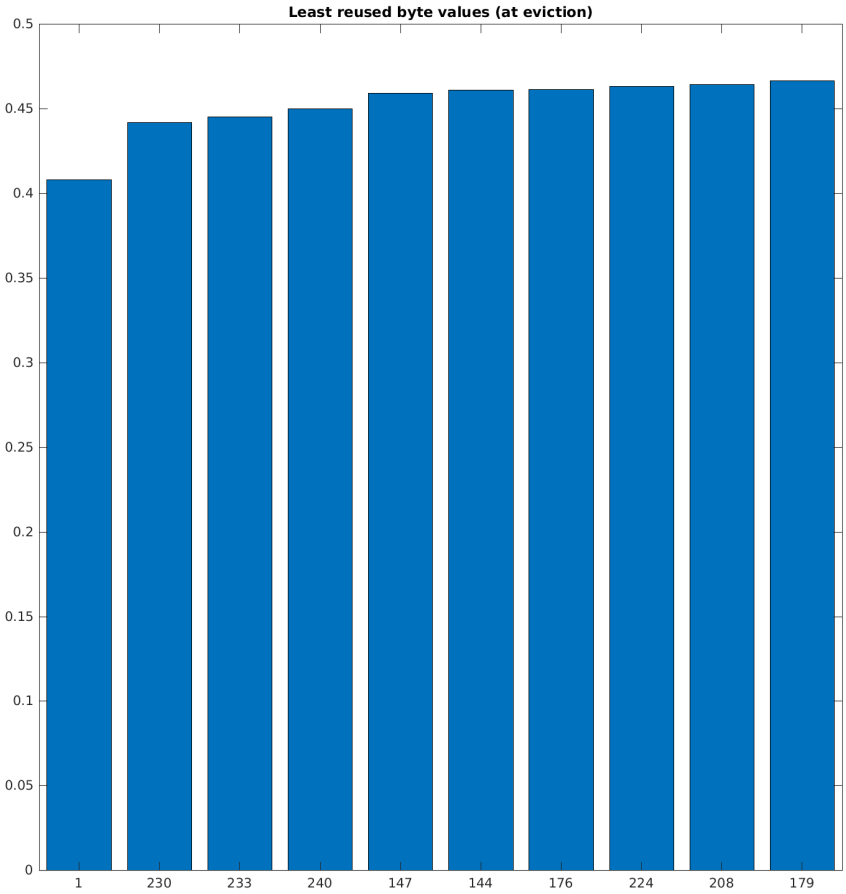


Figure 12: Least reused byte values at eviction