Numerical Project Report- Linear Fit of Cache Miss Rate

Rusu Wu 11694764

Introduction

Memory is one of the most important parts in computer architecture. Considering performance, computer pioneers hope that there are infinite size of fast memory, which is not a reality. In this case, developing memory hierarchy is widely adapted to computer architecture nowadays. Memory hierarchy refers to applying muti-level memory, such as registers of central processing unit (CPU), muti-level caches, and memory storage. Cache is the first level of memory regardless of CPU registers, and using caches to improve memory performance is effective[1]. Commonly, caches store the data that are predicted to be used by CPU in the next machine cycle. If CPU get the requested data from caches, we call a cache hit. On reverse, When CPU could not find the requested data form caches, it is a cache miss. Since directly searching data from normal memory is time-intensive. When cache misses happen, CPU has to find data from memory storage, which would increase the reaction time. Therefore, reducing cache miss rate is one of the hottest areas of research at present.

Problem

There are three simple categories that cause cache misses, and they are Compulsory, Capacity and Conflict. Compulsory misses are independent of cache size, while capacity misses decrease as capacity of cache increases, and conflict misses decrease as associativity increases.[1] Problem is that we need to find the curve that best fits the data to show the

decreasing trend trying to get the most effective way to reduce cache miss.

Commonly, computer architecture applies 4 categories of associativity including one-way, two-way, four-way and eight-way. Higher associativity also are used, but it would cost more address index resources. Moreover, cache size would increase from 4 KB to 512 KB in this project. One thing we should known that enlarging cache size is easier than increasing associativity.

Table 1 shows the relationship between total miss rate and cache size or associativity.

In this project, I would analyze the decreasing trend of cache misses when the capacity and associativity increase. By using the Best-Fit Method, I would find the most suitable linear curve for the decreasing trend of cache miss rate as factors change.

Table 1. The miss rate for each cache size at the corresponding associativities[1]

	4KB	8KB	16KB	32KB	64KB	128KB	254KB	512KB
1-way	0.098	0.068	0.049	0.042	0.037	0.021	0.013	0.008
2-way	0.076	0.049	0.041	0.038	0.031	0.019	0.012	0.007
4-way	0.071	0.044	0.041	0.037	0.03	0.019	0.012	0.006
8-way	0.071	0.044	0.041	0.037	0.029	0.019	0.012	0.006

Method

This project mainly bases on the least square method. The least square method is a mathematical optimization technique. It finds the best functional match curve by minimizing the sum of squares of errors.The unknown data such as prediction data can be obtained simply by using the least square method, and the sum of squares of the error between the obtained data and the actual data is minimized.

With a a number of unknown parameters(x1,y1), (x2,y2).....(xn, yn) ,we assume that the data adapts to the formula

$$\mathrm{w_i} = \mathrm{f}\ (x_i, w_i) = \mathrm{mx_i} + \mathrm{c}$$

we could find the sum of the squares of R^2 using the least square method.

$$R^2 = \sum_{i=1}^{n} (y_i - f(x_i, \mathrm{w}_i))^2$$

$$\frac{\partial(R^2)}{\partial m} = -2\sum_{i=1}^{n} [yi - (mxi + c)]^2 x_i = 0$$

$$\frac{\partial(R^2)}{\partial c} = -2\sum_{i=1}^{n} [yi - (mxi + c)]^2 = 0$$

By finding the minimum value of R^2, we could determine the most suitable m and c to achieve linear fit.

In matrix form, if the data that do not math Ax=b; x=$\begin{bmatrix} \mathrm{m} \\ \mathrm{c} \end{bmatrix}$. By using the least square method, we can seek the x that gets closest to being Ax=b. Let A be an (m * n) matrix, Then we have a formula as below[3],so the system has the unique solution of x.

$$A^T A\mathrm{x} = A^T \mathrm{b}$$

Hence, the system has the unique solution of x.

$$\mathrm{x} = (A^T A)^{-1} A^T \mathrm{b}$$

Computation process

Considering cache size only can be the multiple of 2, I use 1 to 8 to represent 4KB to 512 KB cache size for adapting to linear curve. B represents the cache size ,and y represents the total miss rate. Base on the method above, $x=\begin{bmatrix} m \\ c \end{bmatrix}$

a) One-way associativity

$$B_1=\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \end{bmatrix}; \quad y_1=\begin{bmatrix} 0.098 \\ 0.068 \\ 0.049 \\ 0.042 \\ 0.037 \\ 0.021 \\ 0.013 \\ 0.008 \end{bmatrix}; \quad B_1^{\mathrm{T}} B_1 \begin{bmatrix} m_1 \\ c_1 \end{bmatrix} = B_1^{T} y_1$$

Then we could get $m_1 = -0.0118; c_1 = 0.0953$; The linear fit curve is

$$y_1 = -0.0118x+0.0953$$

b) Two-way associativity

$$B_2=\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \end{bmatrix}; \quad y_2=\begin{bmatrix} 0.076 \\ 0.049 \\ 0.041 \\ 0.038 \\ 0.031 \\ 0.019 \\ 0.012 \\ 0.007 \end{bmatrix}; \quad B_2^{\mathrm{T}} B_2 \begin{bmatrix} m_2 \\ c_2 \end{bmatrix} = B_2^{T} y_2$$

Then we could get $m_2 = -0.0088; c_2 = 0.0738$; The linear fit curve is

$$y_2 = -0.0088x+0.0738$$

c) Four-way associativity

$$B_3 = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \end{bmatrix}; \quad y_3 = \begin{bmatrix} 0.071 \\ 0.044 \\ 0.041 \\ 0.037 \\ 0.030 \\ 0.019 \\ 0.012 \\ 0.006 \end{bmatrix}; \quad B_3^{\mathrm{T}} B_3 \begin{bmatrix} m_3 \\ c_3 \end{bmatrix} = B_3^{T} y_3$$

Then we could get $m_3 = -0.0082$; $c_2 = 0.0694$; The linear fit curve is

$$y_3 = -0.0082x+0.0694$$

d) Eight-way associativity

$$B_4 = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & 1 \\ 8 & 1 \end{bmatrix}; \quad y_4 = \begin{bmatrix} 0.071 \\ 0.044 \\ 0.041 \\ 0.037 \\ 0.029 \\ 0.019 \\ 0.012 \\ 0.006 \end{bmatrix}; \quad B_4^{\mathrm{T}} B_4 \begin{bmatrix} m_4 \\ c_4 \end{bmatrix} = B_4^{T} y_4$$

Then we could get $m_4 = -0.0082$; $c_4 = 0.0693$; The linear fit curve is

$$y_4 = -0.0082x+0.0693$$

Results

In this part, I conducted comparison between actual data curve and the linear fit curve that I found with the least square method.

a)One-way associativity

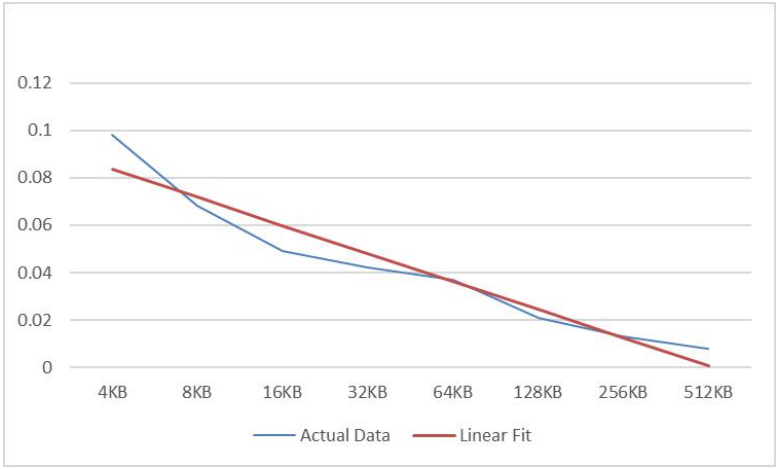

Figure 1. Miss rate changing trend of One-way associativity

b)Two-way associativity

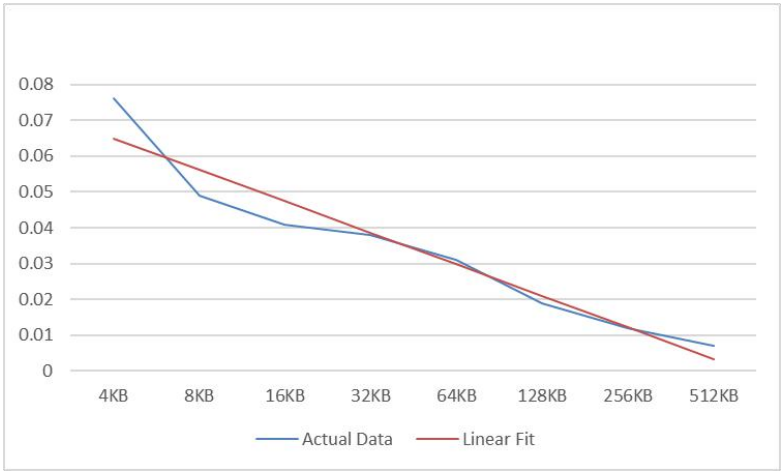

Figure 2. Miss rate changing trend of Two-way associativity

c) Four-way associativity

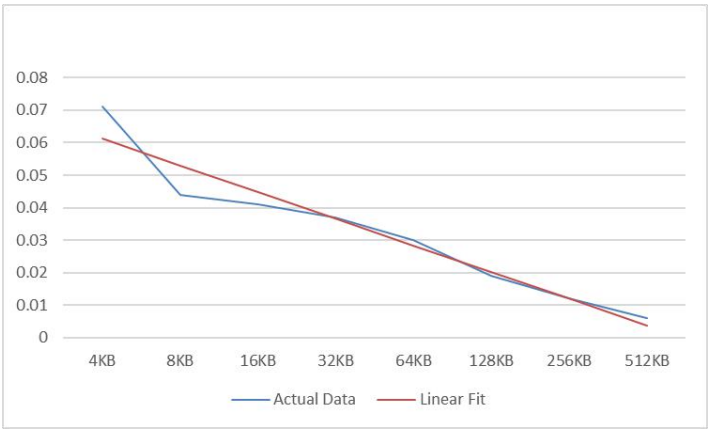

Figure 3. Miss rate changing trend of Four-way associativity

d)Eight-way associativity

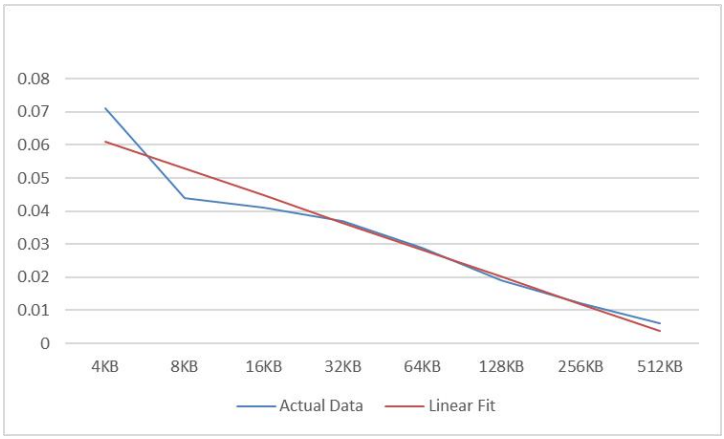

Figure 4. Miss rate changing trend of Eight-way associativity

d)Comparison of miss rate between different associativity under different cache size

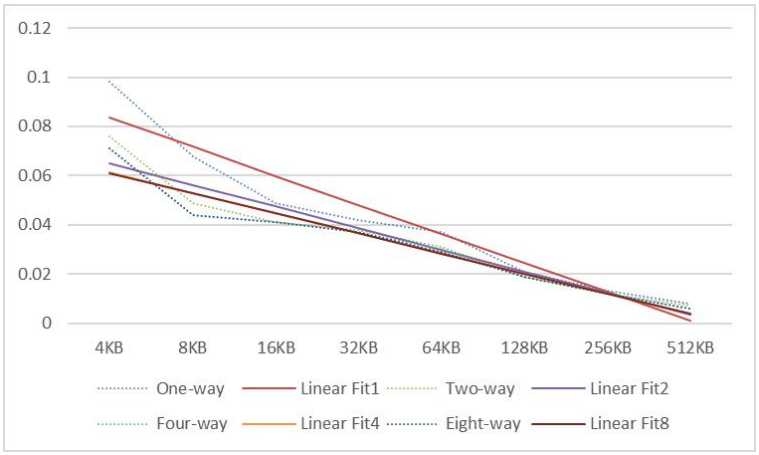

Figure 5. Miss rate changing trend of differnent associativities

Conclusion

As Figure 1, Figure 2, Figure 3, Figure 4 show, when we increase the cache size from 4KB to 512 KB, total miss rate would decrease linearly no matter for which associativity. Total miss rate would decrease as associativities and cache size increase. Also, There is a shock reduce when the cache size changes from 4 KB to 8 KB, while other level of changes have stable decreases. This means that linear can not fit well when the size change from 4KB to 8KB.

From Figure 5, another notable result is that increasing cache size is more efficient on reducing cache miss rate than improving the associativity of cache. To explain, When the cache size changing from 4KBto 32KB, the cache miss rate decrease near 0.03, while there is only near 0.025 decrease of miss rate when associativity is improving from one-way to eight-way in 4KB. Moreover, based on Figure 5, the curves of four-way assiciativity are almost the same as Eight- way associativity; in other words, there is little changes of miss rate when we increase associativity from 4-way to 8-way. When cache size is in a very big size such as 256 KB, the impact from associativity to miss rate can be neglected. Therefore, under a high cache associativity, increasing the cache size could be the optimal choice to reduce total miss rate.

References

[1] Hennessy, J. L., & Patterson, D. A. (2019). *Computer architecture: A quantitative approach* (6th ed.). Cambridge (Ma): Morgan Kaufmann.

[2] Least Squares Fitting. (n.d.). Retrieved October 05, 2020, from https://mathworld.wolfram.com/LeastSquaresFitting.html

[3]Margalit, D., & Rabinoff, J. (2018). *Interactive Linear Algebra.* Georgia Institute of Technology.