하이브리드 메모리 시스템의 지역 가중 선형 회귀 프리페치 방법

당천, 김정근, 김신덕
연세대학교 컴퓨터공학과
quintang@yonsei.ac.kr , junggeun@yonsei.ac.kr, sdkim@yonsei.ac.kr

Locally weighted linear regression prefetching method for hybrid memory system

Qian Tang, Jeong-Geun Kim, and Shin-Dug Kim
Dept. of Computer Science, Yonsei University

Abstract

Data access characteristics can directly affect the efficiency of the system execution. This research is to design an accurate predictor by using historical memory access information, where highly accessible data can be migrated from low-speed storage (SSD/HHD) to high-speed memory (Memory/CPU Cache) in advance, thereby reducing data access latency and further improving overall performance. For this goal, we design a locally weighted linear regression prefetch scheme to cope with irregular access patterns in large graph processing applications for a DARM-PCM hybrid memory structure. By analyzing the testing result, the appropriate structural parameters can be selected, which greatly improves the cache prefetching performance, resulting in overall performance improvement.

1 Introduction

In recent years, the performance of the CPU has developed rapidly, but due to the limitation of the Memory Wall [1], the access speed of the memory has not been improved as much as the CPU, which has greatly affected the improvement of the performance of the computer system.

Prefetching is a technique for improving performance by pre-loading the data to be accessed next. However, if the prefetch data is inaccurate, it can degrade performance.

Prefetching is fundamentally a regression problem. The output space, however, is both vast and extremely sparse, making it a poor fit for standard regression models.

This paper makes the following contributions:

(1) Analyze the characteristics of cache failure behavior and design a prefetching mechanism of locally weighted linear regression.

(2) Simulate and test the performance of the local linear regression prefetch mechanism, and select appropriate structural parameters based on

the simulation results to maximize the prefetch performance.

(3) propose a cost-effective DRAM-PCM hybrid main memory structure.

Our experimental evaluation shows a performance improvement of 8 percent over existing prefetch models such as GHB, SMS, AMPM, and Bestoffset, and 2 percent over the latest model.

2 Background

Prefetchers can largely be separated into two categories: stride prefetchers and correlation prefetchers.

Stride prefetchers are only suitable for stable and repeatable address differences [2].

Compared with stride prefetchers, Associated prefetchers are better at predicting the irregular pattern. Typical Associated prefetchers such as GHB prefetchers [3]. But Associated prefetchers can not aggressively and accurately prefetch on workloads with such characteristics.

To solve this problem, Ji-Tae Yun proposed a regression prefetching mechanism for aggressive prefetching [4].

Based on these observations, we propose a locally weighted linear regression method that can aggressively prefetch. Yun's algorithm is optimized for the overall data, but ignores the principle of locality. We use the local weighted linear regression algorithm to not only consider the importance of local data, but also make other data participate in the overall prediction.

3 Preprocessing Algorithm

The prefetch operation method is as follows. First, the engine finds the entry with the largest number of offsets in the region table. We define x as a sequence of sorted offsets, and y as its offset value. a set of (x, y) pairs can be used as training data. Second, according to the distance from other points to the last point, the weight of each point is calculated. Then, If the amount of data is large enough, perform a locally weighted linear regression algorithm, if the amount of data is too small, perform a simple linear regression algorithm. And the coefficients of this regression model are obtained using Least Squares Method with pivoting.

Last, if we denote the function of these coefficients as F and the number of preprocessed addresses as n, we solve F(n+1), which is the prefetch offset address. Then the next address to be prefetched by adding F(n+1) to the region address of the offset entries.

This prefetching operation is the same as that shown in Algorithm 1.

Algorithm 1: prefetching algorithm

```
//Step1:
For address in Region table
If offsetTable[i]dataSize > offsetSize
offsetEntries ← offsetTable[i]
//Step2:
x ← i+1;
y ← offset[i]Value;
weight[i]= Exp(-(y[i] -y[maxNum]^2)/2)
//Step3:
If offsetSize > 15
y ← y * weight
  else
y ← y
sortAscendingOrder(sortArray,offsetTable[i])
//fined regression coefficient
coefficient ←findRegressionCoefficient(sortArray)
//predict offset
 predictOffsetValue (coefficient, offsetSize + 1)
 PrefetchAdd ← regionTable[i] + predictedOffset
  Return PrefetchAdd
```

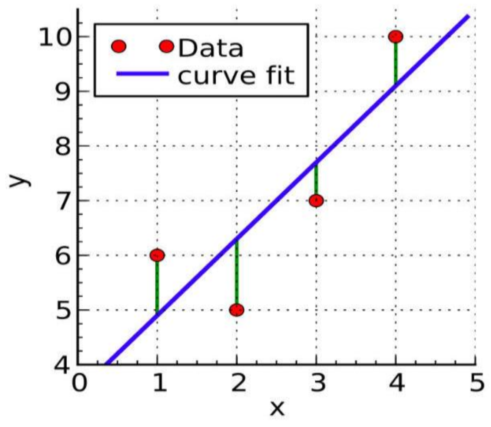

Fig.1 Least Squares Method of Linear Regression

4 Evaluation

To evaluate the proposed model, we used a trace-driven simulator. The trace files were generated by pin tool [5]. We chose GraphBig workloads and our workloads are depicted in Table 1. Tables 2 and 3 depict the parameters used to evaluate our proposed prefetching scheme and the hybrid main memory architecture based on DRAM and PCM.

TABLE 1

Evaluation Workload

GraphBig	**BFS,CCOMP,Dcentr,TC**
Mix 0	GemsFDTD, mcf, raytrace, canneal
Mix 1	facesim, streamcluster, GemsFDTD, mcf
Mix 2	fluidanimate,GemsFDTD,raytrace, canneal
Mix 3	facesim, mcf, fluidanimate, raytrace

TABLE 2

Simulation Configuration

L1 I/D cache 32 KB/8-way/64 B/LRU
L2unifiedcache 256 KB/8-way/64 B/LRU
L3 unified cache 8 MB/16-way/64 B/LRU
Prefetch buffer 128 MB DRAM/4 KB/LRU
Hybrid main memory 128 MB DRAM/2 GB PCM /4 KB/ LRU

TABLE 3

Simulation Parameters

Parameter	**DRAM**	**PCM**	**HDD**
Write latency	20~50 ns	~1 us	~5 ms
Read latency	20~50 ns	~50 ns	~5 ms
Idle power	100mW/GB	1mW/GB	10W/TB
Density/Cost	1x / 4x	4x / 1x	N/A

From the depicted configurations, we chose 256 MB DRAM with 2 GB PCM memory as a middle-ground between execution time reduction (performance) and memory total hit rate.[6] We applied this configuration for the following experimental prefetch model evaluation. Figs. 2 and 3 present the normalized execution time and total hit rate for our proposed model and the other models included in our evaluation.

On average, the proposed model shows 52 percent,13 percent,12 percent, 4.5 percent, and 5 percent improvement over the conventional DRAM model and GHB, SMS, AMPM, and Bestoffset prefetcher.[7] For the Dualbuffer model, the performance improvement is about 2 percent.

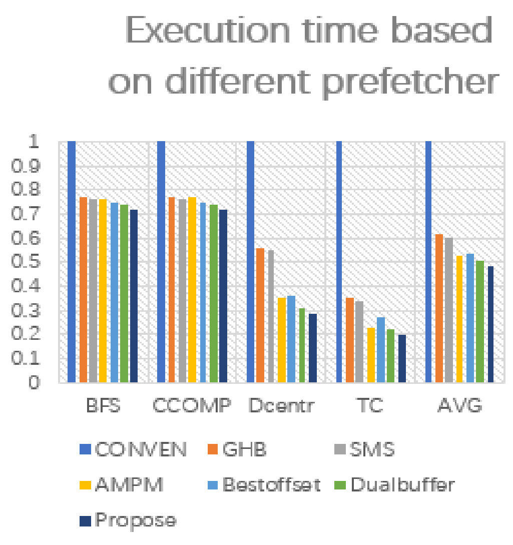

Fig.2 Execution time

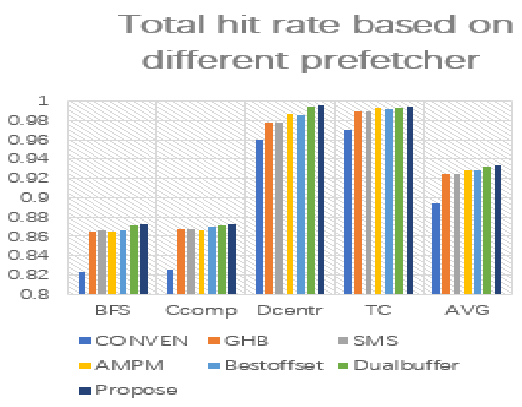

Fig.3 Total hit rate

Fig. 4 shows the performance of our proposed model for various offset sizes with the GraphBig workloads.

It can be seen from the figure.4 that if the offset is too small, the accuracy of the weighted linear regression will decrease. Therefore, when it is large enough, we use locally weighted linear regression algorithms; when it is not large enough, we use simple linear regression algorithms.

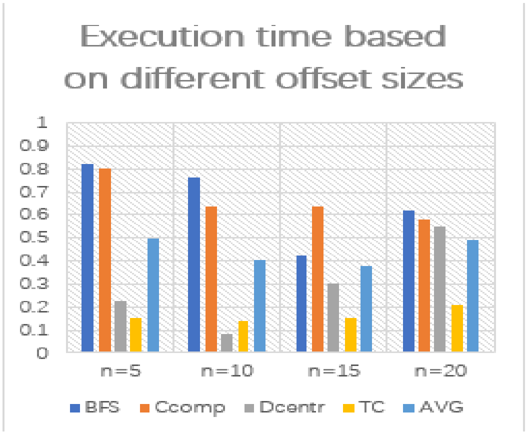

Fig.4 Execution time for different offset sizes

5 Conclusion

In this paper, we propose Locally weighted linear regression prefetching method for hybrid memory system.

Our proposed model shows a 52 percent improvement over the conventional model and a 2 percent improvement over the latest prefetching scheme.

Through the analysis of the simulation results, choosing appropriate structural parameters can better play the role of prefetching.

Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2015M3C4A7065522).

References

[1] Wulf, et al., "Hitting the memory wall: Implications of the obvious." ACM Computer Architecture News, 1995.

[2] J. W. C. Fu, et al., "Stride directed prefetching in scalar processors." in Proc. IEEE/ACM Int. Symp. Micro architecture, 1992.

[3] Nesbit, et al., "Data cache prefetching using a global history buffer. " InIEEE International Symposium on High Performance Computer Architecture, 2004.

[4] Ji-Tae Yun, et al., "Regression Prefetcher with Preprocessing for DRAM-PCM Hybrid Main Memory." in IEEE Computer Architecture Letters,2018.

[5] C.-K. Luk, et al., "Pin: Building customized program analysis tools with dynamic instrumentation, " in Proc. ACM SIGPLAN Conf. Program. Language Des. Implementation, 2005.

[6] K. H. Park, et al., "Mn-mate: Resource management of many cores with dram and non-volatile memories," Proc. 12th IEEE Int. Conf. HPCC,Sep.2010.

[7] P. Michaud, "Best-offset hardware prefetching, " in Proc. IEEE Int. Symp. High Perform. Comput. Archit., 2016.