Title	時間情報に基づいたキャッシュの置換方式に関する研究
Author(s)	塩安，麻人
Citation	
Issue Date	2006-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1957
Rights	
Description	Supervisor:田中　清史，情報科学研究科，修士

A Study of replacement policy based on Time Information in Cache

Asahito Shioyasu (410058)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 9, 2006

Keywords: time information, replacement policy, cache.

1 Introduction

Recently, there is a speed-gap between the processor and the main memory inth problem. The cache memory is used to deal with this problem, and the performance decreases when the access to the memory by cache misses happens frequently. It is necessary to maintain a high hit rate in the cache of a upper hierarchy to prevent an access to the memory.

Seeing from the viewpoint of access time is not effective though increasing in cache capacity and increasing the associativetiy to attempt the improvement of te hit rate. Therefore, the associative cache of the limited associativity is used. When cache miss is hapened by associative cache, it is necessary to select cache block in same cache entry. When the cache block that will be accessed in the near future is selected, the possibility that the cache miss happened is high. It is important to select the cache clock that will not be accessed in the near future. Present, LRU method is generally used as replacement policy in cache.

In this paper, it proposes replacement policy based on time information in data cache and instruction cache. This replacement polisy predict cache block that is accessed in the near future and select replacement target. As a result, it is purpose that the number of cache miss is reduced and improve cache hit rate.

2 Cache memory

2.1 Time information

In the time information of the cache block, there are live time and dead time.

- live time
  Time between when the block fill in cache, and ends with the last hit before the block is replaced.
- dead time
  Time between the last hit and when the block is actually replaced.

The Cache block in live time is accessed in near future, and in dead time is not accessed. It is important that the block in dead time give priority and replaced.

2.2 Replacement policy

It is necessary to decide which block to replace when cache miss happend by associative cache. The most general replacement policy is LRU method. LRU method replaces the block where the time that is not being accessed is the longest. This is replaced with time information, predict in dead time the block where time that is not being accessed is the longest.

3 Replacement policy based on time infomarion

The selection of the replacement policy is important in the cache of the upper level. This research propose that replacement policy based on time information in cache block predict in dead time or live time block and select replace target. This research use previous live time in cache block for judgement in dead time or live time. This mean that predict about same live time when the clock load from first L1 cache, and the block is loaded to L1 cache. As a result, block predict in live time is not replaced by priority as the replacement target during the time of previous live time. The threshhold in preparation for the cade where predict live time becomes small too much when a rival miss happen is installed. It proporses the policy to use previous live time when prediction live time is smaller than the threshhold.

4 Related work

The outline of three reference theses that use time information is shown.

5 Evaluation

The evaluation of this proposal technique was evaluated by simulation. Comparing four benchmark programs of SPECint95 were made an input file and LRU method and the proposal technique were evaluated.

6 Conclusion

The proposal method was evaluated by the simulation by using the benchmark program of SPECint95. It is necessary to set an appropriate threshold to predict live time accurately by this proposal techniqe. Live time that is more accurate than LRU method can be predict by substituting it after the best threshold is set, the cache hit rate has been improved.