Title	スレッドレベル投機実行を支援するキャッシュ機構に関する研究
Author(s)	越前，智史
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1793
Rights	
Description	Supervisor:田中 清史，情報科学研究科，修士

A Study on Cache Mecanisms Supported Thread-level Speculation

Satoshi Koshimae (210036)

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

February 13, 2004

Keywords: Cache, Thread-level Speculation, Syncronization.

1 Introduction

It is important to exploit instruction level parallelism (ILP) for superscalar processor. But it is performance limitations of instruction level paralellism by data dependences among instructions. So there are thread-level speculation on chip multiprocessor (CMP),that is architecture of extracting thread-level execucion from sequential program. Thread-level speculation is a techniqe that enable parallel execution of squential program, that cannot completely resolve the data dependences problem. In thread-level speculation, once detecting data dependence violation, it ensures the true execution of sequential program by *squash* the executing thread.

This paper shows efficient memory system which has data dependence detection in distributed chahes supported thread-level speculation.

2 Thread-level Speculation

A cache system of the architecture which execute thread-level speculation has a crucial role to detection of data dependence violation for speculative memory access in addition to the original. By type of detection of data dependence violation, threre are constitution of the cache added Read bit

(R) which means that the processor read data speculatively and Speculative Number field (Spn) which represent speculative version of data in extisting cache. Spn which is speculative version of data means the larger number as more speculative and in case of Spn 0 represent non-speculative (called Head) in the executing thread. It can detect Data dependence violation by the store propagation ,that is, when one processor whites data in the own cache, its data is propageted for the other caches on distributed snooping-bus cache. In case of one cache propages the data which written the processor and an another chache has data whose Spn is lower than the Spn of propageted data and which set R bit, the cache act data dependence violation.

3 Design of Advanced Cache Architecture

3.1 Syncronization of Data Dependence

When speculative thread occur the data dependence violation, the thread as well as the all threads which are its child threads squashed as these must be occured dependence violation. So our cache appends store propagation history among theads in the existing cache. This appended cache inhibits the additional squash after the thread is squashed again. The cache line of this cache has additional control field (Curr, Prev, Sp). The addition is used syncronization for each thread in reference to data dependence. Here is the execution of operation after the squash. The parent thread does not propagete store in case of less store count (Curr) that is up to now than that executed (Prev) before the thread squashed. On the other hand the child thread is stalled in case of acessing the cache line that is set Sq (When squashed). When the parent thread propagate store, the cache line of setting Sq bit in the cache of the child thread is clear and the child thread release stall.

Using this technique, the each thread can syncronize data dependence and it can reduce data dependence violation times.

3.2 Efficient Technique by Redundancy Cache

In exsisting speculation, a next thread starts execution after a Head thread finished in same processor. However, the cache of the processor has to be Invalidation/Write back corresponding cache line before the next thread start. To reduce the overhead of invalidation/Write Back transaction, the exsisting cache reconfigure as redundancy distribute caches. The redundancy cache which is not assigned the thread execution behave the waiting cache. Then the redundancy cache snoop the current data whose the others propagete in shared bus and can be executed the next thread by the processor immediately. The cache which was used the processor earlier done invalidation/Write Back transaction in the background.

4 Result

We proposed the technique of reducing squash times for data dependence violation and reducing cache miss ratio. In the technique of reducing squash times, it can avoid predictable data dependence violation and inhibiting traffic volume of shared bus. Additionally, To use redundancy cache, the next thread can start the execution immediately and can reduce cache miss ratio.