

ILP 프로세서에서 데이터 값 예측기의 성능 평가

박희룡*, 전병찬**, 이상정***
 순천향대학교 컴퓨터 공학부

The Performance evaluation of Data Value Predictor in ILP Processor

Hee-Ryong Park*, Byong-Chan Jun**, Sang-Jeong Lee***

요 약

본 논문에서는 ILP(Instruction Level Parallelism)의 성능향상을 위하여 데이터 값을 미리 예측하여 병렬로 이슈(issue)하고 수행하는 기존의 데이터 값 예측기(data value predictor)를 비교 분석하여 각 예측기의 예측율을 측정하고, 2-단계 데이터 값 예측기(Two-Level Data Value Predictor)와 혼합형 데이터 값 예측기(Hybrid Data Value Predictor)에서 발생하는 aliasing을 측정하기 위해 수정된 데이터 값 예측기를 사용하여 측정한 결과 aliasing은 50% 감소하였지만 예측율에는 영향을 미치지 못함과 데이터 값 예측기의 예측율을 측정한 결과 혼합형 데이터 값 예측기의 예측율이 2-단계 데이터 값 예측기와 스트라이드 데이터 값 예측기(Stride Data Value Predictor)에서 평균 5.7%, 최근 값 예측기(Last Data Value Predictor)보다는 평균 38%의 예측 정확도가 높음을 입증하였다.

1. 서론

최근 하드웨어 기술의 급속한 발달로 많은 고성능 프로세서(High Performance Processor)들이 개발되고 있으며, 고성능의 실현을 위해 ILP를 이용하여 성능의 극대화를 꾀하고 있다. ILP처리의 주요 장애요소는 명령들 간의 데이터 종속성(data dependency) 관계로 명령들의 병렬처리를 방해한다. 따라서 데이터 종속관계를 극복하는 방법으로 실행되는 명령들의 결과 값을 미리 예측하고 이후 데이터 종속관계가 있는 명령들에게 조기에 데이터 값을 공급하여 실행시킴으로써 성능향상을 꾀하는 데이터 값 예측방식의 연구가 진행되고 있다[1,2,4,5]. 데이터 값 예측기는 하나의 명령을 실행하기 전에 실행결과를 하드웨어에 의해 미리 예측하고 투기적(speculative)으로 실행하여 이 명령과 데이터 종속관계가 있는 명령들이 중지(stall)되지 않고 계속 이슈하여 실행함으로써 프로세서의 성능을 향상시킨다.

2. 데이터 값의 시퀀스

[그림 1]은 데이터 값의 시퀀스(data value sequence

)패턴으로 상수(Constant) 시퀀스는 0값이 증,감되는 특수한 형태의 스트라이드 시퀀스이고, 스트라이드 시퀀스는 명령이 수행될 때마다 일정한 값만큼의 폭 차이로 증감되는 패턴으로 배열이나 루프, 인덱스등을 참조할 때 빈번히 발생하는 패턴들이다. 비 스트라이드(Non-Stride)시퀀스는 상수 시퀀스나 스트라이드 시퀀스에 속하지 않은 모든 시퀀스이다.

	일반적인 데이터 값
상수(C)	5 5 5 5 5 ... 상수의 증감폭이 모두 0임
스트라이드(S)	1 2 3 4 5 ... 상수의 증감폭이 모두 +1임
비 스트라이드(NS)	28 12 -13 99 456 ... 상수의 증감폭이 일정하지 않음

[그림 1] 일반적인 데이터 값 시퀀스

스트라이드와 비 스트라이드의 혼합된 반복 시퀀스(repeated sequences)로는 [그림 2]와 같은 패턴도 발생할 수 있는데, 반복 시퀀스는 중첩된 루프(loop)에서 내부 루프(inner loop)가 스트라이드 또는 비 스트라이드 시퀀스를 생성하고 외부 루프(outer loop)가 이러한 시퀀스를 반복하는 경우에 발생한다.

	루프내의 반복 데이터 값					
반복 스트라이드(RS)	1	2	3	1	2	3
	내부 루프의 반복 스트라이드값(1, 2, 3)					
반복 비 스트라이드(RNS)	1	34	-3	76	1	34
	내부 루프의 반복 비 스트라이드 값(1 34 -3 76 ...)					
	내부 루프의 반복 비 스트라이드 값(1 34 -3 76)					

[그림 2] 루프내의 반복되는 데이터 값 시퀀스

3. 데이터 값 예측기 모델

데이터 값 예측기에서 계산형 예측방법은 예측하고자 하는 명령의 PC에 의해 VHT(Value History Table)내의 엔트리가 인덱스되어 이전 생성된 데이터 값의 함수로 예측하는 최근 값 예측기와 스트라이드 데이터 값 예측기와 특정 내용 값을 다음에 나타나는 값을 학습하여 동일한 값들이 반복될 때 이전 학습된 값으로 데이터 값을 예측하는 내용형 예측 방식으로 2-단계 데이터 값 예측기와 fcm 예측기가 있다.

4. 데이터 값 예측기의 실험방법

2-단계로 구성된 데이터 값 예측기에서 PHT참조시 exclusive-OR(XOR)사용하지 않았을 경우와 XOR를 사용하였을 경우를 비교 실험하여 전체 예측율을 측정하고, PHT참조시 XOR되는 비트 수와 각 테이블의 크기 및 히스토리의 크기를 변화시켜 aliasing 발생률에 대하여 측정하여 예측율과의 관계성을 실험하였다.

4.1 벤치마크 프로그램

시뮬레이션을 수행하기 위해 사용되어진 SPEC벤치마크를 사용하였다. 시뮬레이션에 사용된 컴파일러는 LINUX GNU Gcc 최적화 컴파일러를 사용하여 실험하였다.

4.2 시뮬레이터 소개

본 논문에서 사용된 시뮬레이터인 SimpleScalar Tool set의 시뮬레이션 과정은 시뮬레이터 입력의 벤치마크는 C 프로그램이나, "f2c 변환기"에 의해 C로 변환되는 FORTRAN 프로그램을 SimpleScalar 아키텍처 C 컴파일러인 "Gcc C 컴파일러"를 통해 어셈블리어를 생성하고, SimpleScalar 어셈블리 프로그램은 "GAS 어셈블러"와 "GLD 로더"를 거쳐 시뮬레이션 실행파일을 생성하여 SimpleScalar 시뮬레이터에 입력으로 사용되어 시뮬레이션을 수행하고 시뮬레이션 결과를 출력한다.

5. 시뮬레이션에 사용된 기본 요소

데이터 값 예측기에서 각 테이블에 참조되어지는 PC(31 비트)와 각 명령들은 8 바이트로 구성되어 상위 3 비트를 제외한 28 비트로 태그와 인덱스로 구성되어 캐시를 직접(direct mapping)참조한다.

5.1 최근 데이터 값 예측기

최근 값 데이터 예측기의 CT구성은 tag 필드와 카운터로 구성되고, 1K 엔트리와 2 비트 카운터로 구성되며, 참조되는 주소 비트는 기본 값은 10 비트로 구성되고, tag 필드는 기본 18 비트로 구성된다. VPT의 구성은 tag 필드와 히스토리 크기만큼의 데이터 값들로 VPT크기는 기본 4K엔트리로 31비트의 value 필드로 구성된다. 또한 히스토리는 LRU정책으로 데이터 값을 교체한다.

5.2 스트라이드 데이터 값 예측기

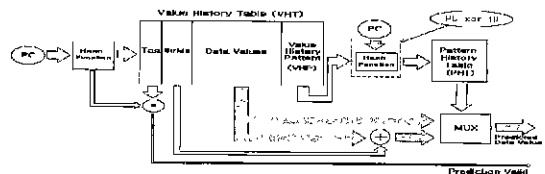
VHT의 엔트리는 태그를 저장하는 tag 필드와 현재 동작 상태를 나타내는 state 필드, 이전의 데이터 값을 저장하는 value 필드와 데이터 값의 반복 시퀀스인 스트라이드 값을 저장하는 stride 필드로 구성된다. VHT의 크기는 기본 4K의 엔트리로 구성하여 실험한다.

5.3 2-단계 데이터 값 예측기

VHT의 크기는 기본 4K의 엔트리와 데이터 값의 교체를 위한 LRU 정보 필드는 히스토리의 크기 * $\log_2(\text{히스토리의 크기})$ 의 데이터 값 위치를 인코딩한 비트 값을 순서대로 저장한다. value 필드는 히스토리의 크기 * 32비트로 구성되고, VHP의 비트수는 $\log_2(\text{히스토리의 크기}) * p$ 의 값으로 구성 되는데 p는 한 명령이 참조되는 마지막 p개의 결과이다. 또한 PHT의 엔트리는 $2^{(\text{VHT의 크기})}$ 로 계산되고, 참조되는 주소 비트는 $\log_2(\text{PHT의 엔트리수})$ 로 구성된다. PHT는 4 비트 증감 카운터로 구성되고, PHT 카운터의 한계값(threshold)은 3, 6, 9로 3씩 증가되며 최대값은 12가 된다. 히스토리의 크기는 2, 4, 6, 8, 10, 12, 14, 16의 값으로 측정 하였다. PHT의 엔트리를 참조하기 위해 PC의 주소 값과 0, 6, 8, 12, 16 비트의 XOR값으로 변화시켜 실험하였다.

5.4 혼합형 데이터 값 예측기

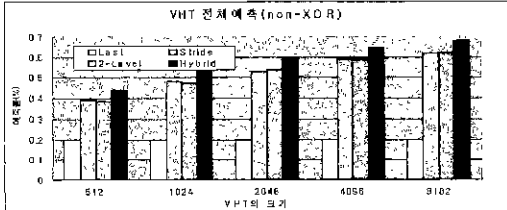
혼합형 데이터 값 예측기의 요소는 2-단계 데이터 값 예측기에서 사용되는 구성과 기본 값들로 동일하게 설정하였다. VHT의 크기는 기본 4K의 엔트리로 구성되고, VHT엔트리에 스트라이드 데이터 값 예측기에서 사용된 stride 필드와 state 필드를 삽입하여 측정하였다.



[그림 4] 수정된 혼합형 데이터 값 예측기

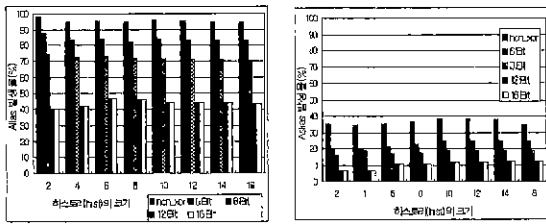
6. 데이터 값 예측기의 실험 결과

본 논문에서 실험한 데이터 값 예측기의 예측율을 측정하여 [그림 5]와 같은 실험결과를 얻었다. [그림 5]에서 혼합형 데이터 값 예측기가 다른 데이터 값 예측기에 비하여 예측히트율이 높음을 보였고, 특히

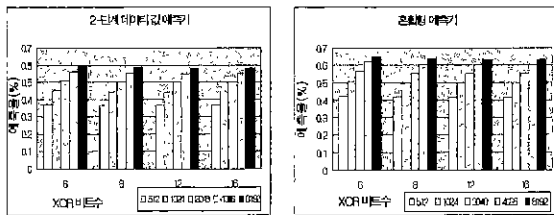


[그림 5] 예측기의 예측율

최근 값 예측기에 대하여는 평균 38% 정도의 예측율이 높음을 알 수 있었다. 2-단계 데이터 값 예측기와 혼합형 데이터 값 예측기에서 PHT참조시 발생하는 aliasing의 실험결과 [그림 6]과 같이 XOR를 하여 PHT를 참조할 경우 aliasing의 감소효과와 2-단계 데이터 값 예측기(a) 보다 혼합형 데이터 값 예측기(b)에서 더 많은 aliasing 감소효과를 나타내었음을 알 수 있었다. 그러나 PHT참조시 XOR할 때와 그렇지 않았을 경우에 평균 4%의 예측율이 감소됨을 볼 수



(a) 2-단계 데이터 값 예측기 (b) 혼합형 데이터 값 예측기
[그림 6] 데이터 값 예측기에 따른 alias발생율



(a) 2-단계 데이터 값 예측기 (b) 혼합형 데이터 값 예측기
[그림 7] XOR했을 경우 데이터 값 예측기의 예측율

있는데, [그림 7]은 PHT참조시 XOR하여 발생하는 alias가 예측율에 미치는 영향이 적음을 알 수가 있다.

6. 결론 및 향후 연구과제

기존 연구된 데이터 값 예측기에 대하여 예측 정확도를 측정된 결과 혼합형 데이터 값 예측기의 예측율

이 2-단계 데이터 값 예측기와 스트라이드 데이터 값 예측기에 비교하여 평균 5.7%, 최근 값 예측기에 대하여는 38%의 예측율이 높음을 입증하였다. 2-단계 데이터 값 예측기와 혼합형 데이터 값 예측기에서 PHT의 매핑시 발생하는 aliasing을 측정된 결과 XOR를 하지 않았을 경우 2-단계 데이터 값 예측기에서는 40%~98%이상 aliasing이 발생했지만, 혼합형 데이터 값 예측기에서는 40%미만의 aliasing감소효과를 보였다. 따라서 적은 aliasing이 발생되짐에 따라 aliasing의 히트율을 증가시킬 수 있었다. 향후 연구과제로는 value locality의 이용과 데이터 값 예측과 관련하여 많은 연구가 필요하고, 데이터 값의 예측은 투기적이기 때문에 예측의 정확성을 확인하여 예측 실패 시 데이터 값들을 효과적으로 복원하는 연구가 있어야 할 것이다.

참고 문헌

- [1] Yiannakis Sazeides and James E. Smith, "The Predictability of Data Values", Proc. of 30th Annual ACM/IEEE International Symposium on Microarchitecture, December 1997.
- [2] M.H.Lipasti and J.P.Shen, "Exceeding the Limit via Value Prediction", Proc. of 29th Annual ACM/IEEE International Symposium on Microarchitecture, December 1996
- [3] C.C.Lee, I.K.Chen, T.N.Mudge, "The Bi-Mode Branch Predictor", Proc. of 30th Annual ACM/IEEE International Symposium on Microarchitecture, December 1997.
- [4] Kai Wang, Manoj Franklin, "Highly Accurate data value Predictions using hybrid predictor", Proc. of 30th Annual ACM/IEEE International Symposium on Microarchitecture, December 1997.
- [5] M.H.Lipasti, C.B.Wilkerson and J.P.Shen. "Value Locality and Load Value Prediction". Proc. of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems(ASPLOS-VII), Oct 1996.
- [6] T.Y.Yeh and Y.N.Patt, "Two-level adaptive branch prediction" in Proceeding of the 24th Annual ACM/IEEE International Symposium on Microarchitecture, pp.51-61, 1991.