

슈퍼스칼라 프로세서에서 데이터 값 예측기의 성능효과

박희룡(朴喜龍), 전병찬(全炳燦), 이상정(李相靜)
김천대학 컴퓨터사무정보계열, 순천향대학교 컴퓨터공학부
전화 : (0547) 420-4162 / 팩스 : (0547) 436-8485

Efficient of The Data Value Predictor in Superscalar Processors

Hee Ryong Park, Byung Chan Jun, Sang Jeong Lee
Dept. of Computer & Office Infprmtion, Kimcheon College
Dept. of Computer Science and Engineering, Soonchunhyang University
E-mail : hrpark@kimcheon.ac.kr / {sjlee, bcjun}@archi-cse.sch.ac.kr

Abstract

To achieve high performance by exploiting instruction level parallelism(ILP) aggressively in superscalar processors, value prediction is used. Value prediction is a technique that breaks data dependences by predicting the outcome of an instruction and executes speculatively it's data dependent instruction based on the predicted outcome. In this paper, the performance of a hybrid value prediction scheme with dynamic classification mechanism is measured and analyzed by using execution-driven simulator for SPECint95 benchmark set.

I. 서론

최근의 고성능 프로세서는 성능향상을 위해 여러 개의 명령을 동시에 이슈하고 처리하는 명령어 수준 병렬성(Instruction Level Parallelism, ILP)을 이용하고 있다. [3] 이러한 ILP 처리의 주요장애는 명령간의 종속관계인데 이는 명령의 병렬처리를 방해한다. 즉, 현재의 한 명령이 이전 명령과 종속관계가 있으면 이전명령의 결과가 나올 때까지 현재의 명령은 이슈(issue)되어 병렬로 실행할 수 없게된다. 이러한 명령어간의 종속관계는

이름 종속(name dependency), 제어 종속(control dependency), 데이터 종속(data dependency)관계의 세 가지로 구분할 수 있다. 데이터 종속 관계는 현재의 명령이 이전 명령의 수행결과를 참조할 때 발생하고 이전 명령의 결과가 발생할 때까지 현재의 명령은 실행할 수 없게된다. 데이터 종속 관계는 명령간에 빈번히 발생하기 때문에 ILP 처리의 주요 장애가 되어 최근의 고성능 프로세서의 성능저하의 주된 요인이 되고 있다. 따라서 최근에는 실행되는 명령의 결과 값을 미리 예측하고 이후 데이터 종속관계가 있는 명령들에게 조기에 공급하여 실행시킴으로써 성능향상을 모색하는 데이터 값 예측(data value prediction)방식에 관하여 활발히 연구가 진행되고 있다.[3,4] 그림 1은 데이터 종속관계를 보여주는 그림으로 (a)는 j의 R3는 i의 R3 결과가 생성되어야만 사용이 가능하며, j가 실행된 후 R5의 결과가 생성되면 명령 k, l이 실행될 수가 있다. 이러한 데이터 종속관계가 있는 명령들을 (b)와 같이 i, j의 결과 값 R3, R5를 미리 예측하여 i, j와 데이터 종속관계를 갖는 명령 k, l들에 대한 종속관계를 제거함으로써 이들 명령들을 투기적으로 실행(speculative execution)할 수 있음을 보여주는 그림이다.

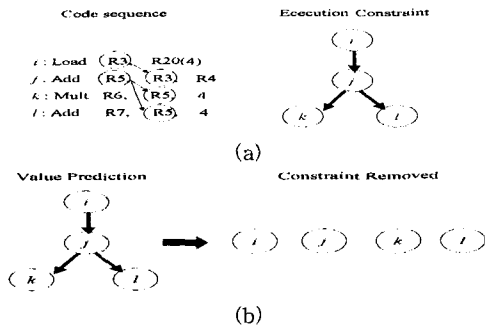


그림 1. 명령에서의 데이터 종속관계

본 연구에서는 동적 분류 능력을 갖는 혼합형 데이터 값 예측기를 실험한다. 실험되는 예측기는 최근 값 예측기[3], 스트라이드 예측기 [9] 및 2-단계 예측기 [4]를 결합한 혼합형으로 구성되며, 예측되는 명령은 하드웨어에 의한 동적 분류에 의해 각 예측기로 할당된다. 실험의 타당성 검증을 위해 실행 구동 방식(execution-driven) 시뮬레이터인 SimpleScalar/Alpha 3.0 tool set [8]에 데이터 값 예측기를 추가하고, 이를 이용하여 SPECint95 벤치마크를 시뮬레이션하여 비교한다.

II. 데이터 값 예측 유형과 방법

프로세서에 의해 반복 수행되어지는 명령들은 다음과 같은 상수(constant)나 스트라이드(stride) 또는 비 스트라이드(non-stride)와 같은 데이터 값의 유형에 대한 시퀀스 패턴(data value sequence pattern)을 갖는다.

데이터 값의 유형	일반적인 데이터 값
상수	5 5 5 5 5 ... 상수의 증감 폭이 모두 0임
스트라이드	1 2 3 4 5 ... 상수의 증감 폭이 모두 +1임
비 스트라이드	28 12 -13 99 456 ... 상수의 증감 폭이 일정하지 않음

즉, 상수(Constant) 시퀀스는 0값이 증감되는 특수한 형태의 스트라이드 시퀀스로 볼 수 있고, 스트라이드 시퀀스는 명령이 수행될 때마다 일정한 값만큼의 간격 차이로 증감되는 패턴으로 프로그램에서 배열(array)이나 루프 인덱스(loop index)등을 참조할 때 빈번히 발생하는 패턴이다. 또한 비 스트라이드(Non-Stride)시퀀스는 위의 상수 시퀀스 또는 스트라이드 시퀀스에 속하지 않은 모든 시퀀스를 의미한다. 이러한 데이터 값 시퀀스에서 특히 상수 시퀀스는 같은 값이 반복되어 사용되는 패턴으로 Lipasti 등은 이러한 패턴이 자주 발생함을 주목하여 최근 값 예측기를 개발하였다.[3] 또한 스트라이드와 비 스트라이드의 혼합된 형태의 반복 시퀀스(repeated sequences)로는 다음과 같은 패턴도 발생할 수 있다.

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

이와 같은 반복 시퀀스는 중첩된 루프에서 내부 루프(inner loop)가 스트라이드 또는 비 스트라이드 시퀀스를 생성하고 외부 루프(outer loop)가 이 시퀀스를 반복하는 경우가 발생한다. 이상과 같은 데이터 값 시퀀스 분류에 의해 데이터 값 예측기를 계산형 예측기(Computational Predictor)와 내용형 예측기(Context-based predictor)로 구분된다. 본 연구에서 실험되는 동적 분류 능력을 갖는 혼합형 데이터 값 예측기는 최근 값 예측(Last value prediction)과 스트라이드 예측(Stride value prediction), 데이터 값들의 문맥을 근간으로 한 데이터 값 예측(context-based prediction) 및 혼합된 예측(Hybrid value prediction)을 포함하고 있는데, 최근 값 예측은 한 명령에서 가장 최근에 생성된 데이터 값의 결과 값을 예측하는 것이고, 스트라이드 예측은 가장 최근에 생성된 두 개의 데이터 값에 대한 차(difference)를 계산하여 계산된 결과 값을 스트라이드로 하여 다음 값을 예측한다. 또한 데이터 값의 문맥을 근간으로 한 예측기는 이전의 여러 데이터 값들을 관찰하여 데이터 값들 사이의 반복되어지는 패턴들을 이용하여 다음 값을 예측방법으로 fcm(Finite Content Mechanism) [7]과 2-단계 데이터 값 예측기 [4]가 있다.

III. 동적 분류 데이터 값 예측기

그림 2는 본 연구에서 사용된 동적 분류 능력을 갖는 혼합형 데이터 값 예측기의 구성도를 나타낸 그림으로써, LVT(Last Value Table), SVT(Stride Value Table), 2VT(2-level Value Table)의 3개의 테이블로 구성하고 각 예측기의 분류동작은 SVT에 의해 분류 예측된다. 이들 각 예측 테이블은 모두 명령의 주소(PC)에 의해 인덱스 되는데, 명령의 처음 데이터 값의 예측은 예측분류 테이블인 SVT의 상태필드(state field)를 초기화하고 초기 데이터 값과 스트라이드를 계산하여 저장, pID필드(prediction identification field)의 초기 값으로 "stride"를 설정한다. 다음 동일한 명령의 주소에 의해 인덱스 되어지면 처음과는 달리 분류예측 테이블인 SVT의 pID필드를 참조하여 각 예측 테이블을 지정하고, 데이터 값 유형에 따른 분류를 시도한다. 즉, pID필드의 값을 "Stride," "Last," "2level," "Unknown"와 같은 분류정보를 갖고 데이터 값에 따른 예측을 수행한다.

LVT	최근 값 예측기	confidence counter 2	
SVT	스트라이드 데이터 값 예측기	confidence counter 2	pID 2
2VT	2-단계 데이터 값 예측기	PHT(counters)	

그림 2. 동적 분류 능력을 갖는 혼합형 예측기 엔트리

본 연구에서 사용된 동적 분류 능력을 갖는 혼합형 예측기에서 동적 분류의 목적은 데이터 값 예측을 위해 어느 한 예측기를 선택하고 갱신(update)을 위해 올바른 값으로 갱신할 예측기를 선택하는 것이다. SVT를 이용한 예측기의 분류는 그림 3과 같이 분류학습을 하는데, 먼저 새로운 명령이 들어오면 상태필드의 값이 "Init" 상태가 되고 결과 값으로 value1을 저장한다. 다음에 이 명령을 다시 수행되면 "Init" 상태에서 "Transient" 상태로 천이 되고 현재 결과 값 value2와 이전 값 value1의 차인 stride1을 계산하여 스트라이드 필드에 저장한다. 다음에 수행 시에는 "Steady" 상태로 천이 되고, 현재 값 value3과 이전 값 value2로부터 stride2를 계산한다. 먼저 stride1과 stride2를 서로 비교하여 두 값이 서로 동일하고 스트라이드 값이 0인가를 조사하여 만족하면 pID를 "Last"로 설정하고 최근 값 예측기(LVT)로 예측한다. 또한 비교한 스트라이드가 만일 0이 아니면 두 개의 스트라이드가 같은가를 조사하여 같으면 pID를 "Stride"로 설정하고 계속하여 스트라이드 예측기(SVT)로 예측한다. 그리고 스트라이드가 서로 같지 않으면 pID를 "2level"로 설정하고 이후 2-단계 예측기(2VT)로 분류하여 예측을 시도한다. 그러나 최근 값 예측기와 2-단계 예측기에서 확신 카운터 값이 0이면 각 테이블에서 퇴거(Evict)동작을 수행하게 되고 SVT의 pID를 "Unknown"으로 설정한다. 퇴거 동작에서는 마치 새로운 명령처럼 SVT를 이용하여 상태필드가 "Init" 상태에서부터 명령을 재분류(reclassification)하여 처음부터 다시 분류동작을 계속하게 된다.

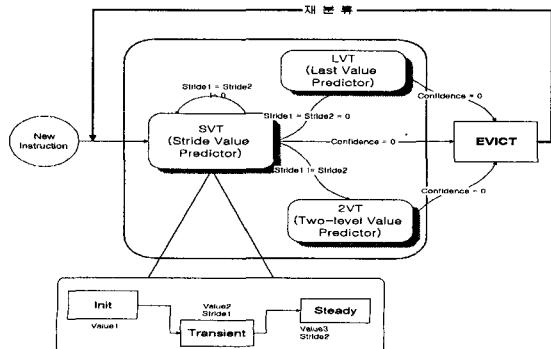


그림 3. 동적 데이터 값 예측기의 분류 동작

IV. 실험 방법

본 연구는 동적 분류 능력을 갖는 혼합형 데이터 값 예측기와 기존의 혼합형 데이터 값 예측기를 비교 실험하기 위하여 4~16 이슈(issues)의 명령의 크기(issue-width)에 대한 3가지 시스템의 기본 구조를 사용하였다. 기존의 혼합형 데이터 값 예측기와 동적 분류 능력을 갖는 혼합형 데이터 값 예측기의 파라미터로써 혼합형 데이터 값 예측기는 8K 엔트리의 VHT(Value History Table)과 4개의 히스토리를 갖고, 8K 엔트리로 구성된 PHT로 구성되고, 동적 분류 능력을 갖는 혼합형 데이터 값 예측기는 4K의 LVT(Last Value Table)와 4K의 SVT(Stride Value Table), 그리고 2K 엔트리로 구성된 2VT(2-Level Value Table)로 4개의 히스토리를 갖는다. 시뮬레이션을 수행하기 위해 사용되어진 벤치마크 프로그램은 표 1과 같이 시뮬레이션에 사용된 입력자료를 생성하기 위하여 GNU's gcc2.7.2을 UNIX 플랫폼상에 컴파일된 SPECint95 벤치마크 프로그램을 사용하였고, 수행 명령 결과들의 평균은 레지스터 기록(register-writing) 명령들이다.

벤치마크 프로그램	입력자료	명령의 크기	IPC(Instruction per Cycle)		
			4-이슈	8-이슈	16-이슈
compress	1000 e 2231	35,726,662	2.1091	2.6898	2.6711
gcc	jump.i	39,980,547	1.5303	1.8231	1.9188
jpeg	tinyrose.ppm	72,608,027	2.4839	3.1860	3.3690
li	queen6.lsp	41,732,243	1.7259	2.2214	2.3096
m88ksim	new.100	95,476,020	2.3579	3.0578	3.2831
perl	scrabbl.in	40,450,590	1.9842	2.6992	2.7978
vortex	persons.250	65,130,658	2.0742	3.1136	3.6511
평균		391,104,747	2.0379	2.6844	2.8572

표 1. SPECint95 벤치마크 프로그램

V. 성능분석

본 연구에서 동적 분류 혼합형 데이터 값 예측기의 IPC(instructions per cycle)를 실험하여 성능향상에 미치는 영향을 분석하기 위하여 이슈의 대역폭을 4~16이슈(issue)로 변화시켜 연구하였다. IPC는 이슈의 대역폭이 높아질수록 IPC의 성능이 평균 9%~17%이 향상됨을 실험결과(그림 4)로 보여준다.

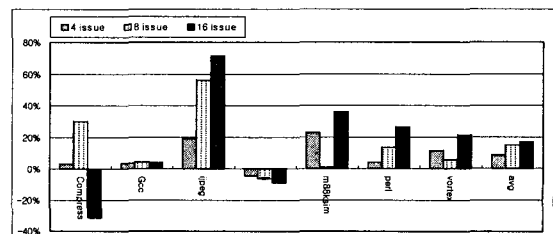


그림 4. 동적 분류 예측기의 IPC

또한 명령어에 대한 예측 실패율을 실험한 결과 그림 5와 같이 기존의 2-단계 데이터 값 예측기에서 평균 29%, 혼합형 데이터 값 예측기에서 50%, 동적 분류 능력을 갖는 혼합형 데이터 값 예측기에서 21%의 실험결과를 보여, 동적 분류 능력을 갖는 혼합형 데이터 값 예측기가 기존의 데이터 값 예측방법 보다 8%~29% 예측 실패율이 감소되었음을 보여주는데, 이러한 낮은 예측 실패율은 데이터 값을 예측하는데 있어서 예측 정확도에 많은 영향을 미치고 있음을 보여준다.

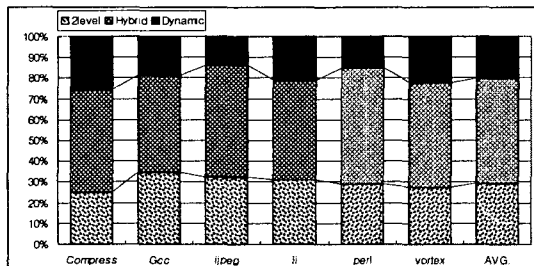


그림 5. 데이터 값 예측 실패율

따라서 기존의 데이터 값 예측방법 보다 다양한 유형의 데이터 값들을 동적으로 분류해 넘으로써 데이터 값 예측의 성능을 향상시키고 있음을 보였다. 그림 6은 2-단계 데이터 값 예측기 및 기존의 혼합형 데이터 값 예측기와 동적 분류 능력을 갖는 혼합형 데이터 값 예측기에 대한 예측 정확도를 실험한 결과이다. 각 데이터 값 예측기의 평균 예측 정확도는 2-단계 데이터 값 예측기에서 81%, 기존의 혼합형 데이터 값 예측기에서 67%, 동적 분류 능력을 갖는 혼합형 데이터 값 예측기에서 83%로 측정되어 기존의 2-단계 데이터 값 예측기와 혼합형 데이터 값 예측기 보다 동적 분류 능력을 갖는 혼합형 데이터 값 예측기에서 평균 16%의 높은 예측 정확도의 결과를 보여 성능이 우수함을 보여주었다.

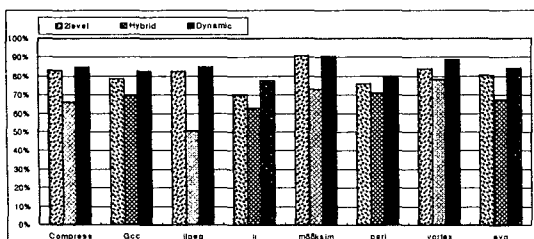


그림 6. 데이터 값 예측기의 예측 정확도

따라서 이슈의 대역폭이 높을수록 여러 개의 데이터 값들을 예측하여 예측율을 높이고 예측 시도된 명령어 중 예측 실패율을 감소시켜 예측 정확도를 높일 수 있다.

VI. 결론 및 추후 연구

본 연구결과 동적 분류 능력을 갖는 혼합형 데이터 값 예측기는 SPECint95 벤치마크 프로그램의 기본 설정된 구조에서 데이터 값에 대한 IPC와 예측 정확도의 성능실험 결과, 기존의 방법보다 성능향상이 0.7%~15%로 높았고, 예측 실패율은 8%~29% 낮았으며, 평균 16% 높은 예측 정확도를 나타내었다. 따라서 동적인 분류 예측 방법에 따른 연구가 더 필요하고, 각 예측기별 데이터 값 유형에 따른 연구로써 예측 시도율 및 예측성능과 하드웨어의 비용효과에 관한 연구도 있어야 할 것이다.

참고 문헌

- [1] F.Gabbay and A.Mendelson, "Can Program Profiling Support Value Prediction?," Proc. of 30th Annual ACM/IEEE International Symposium on Microarchitecture, December 1997.
- [2] Sang-Jeong Lee, Yuan Wang, and Pen-Chung Yew, "Decoupled Value Prediction on Trace Processors," IEEE 6th international Symposium on high performance computer architecture, 2000.
- [3] 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.
- [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] 박희룡, 전병찬, 이상정 "ILP프로세서에서 데이터 값 예측기의 성능평가," 98가을학술발표논문집(III) 제25권 2호, pp21-23, 한국정보과학회 1998.9.
- [6] 박희룡, 전병찬, 이상정 "ILP프로세서의 혼합형 데이터 값 예측기 모델," 99봄학술발표논문집(A) 제26권 1호, pp18-20, 한국정보과학회 1999.4.
- [7] Yiannakis Sazeides and James E. Smith, "The Predictability of Data Values," Proc. of 30th Annual ACM/IEEE International Symposium on Microarchitecture, December 1997.
- [8] D.Burger and T.Austin, "The Simplescalar Tool Set, Version 2.0," Technical Report CS-TR-97-1342, university of Wisconsin, Madison, June 1997.
- [9] B.Calder, G.Reinman and D.Tullsen, "Selective Value Prediction," Proceedings of the 26th International Symposium on Computer Architecture(ISCA-26), May 1999.
- [10] F.Gabbay, and A.Mendelson, "Speculative Execution Based on Value Prediction," EE Department TR 1080, Technion-Israel Institute of Technology, Nov. 1996.