Branch predictor 보고서

12194114 구현규

(1)만든 Predictor 설명 :

과정-1

1. 2bit 프레딕터를 만들어 실행 결과:

total : 299994 branch, correct : 244814 , ratio : 81.606299%

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

2. 2비트 프레딕터에서 가장 큰 저장 공간 2bit\*32000=64000(8kb) 실행결과:

total : 299994 branch, correct : 264129 , ratio : 88.044763%

결론: 단순히 history저장 공간을 늘리는 것만으로도 성능향상에 큰 도움이 된다.

또한 단순히 2bit로 만드는 것은 1bit 와 큰 차이가 없다. 즉 다른 방식의 predictor가 필요함.

최종 결과물

먼저 pc % num\_predictor\_entry 즉 num\_predictor\_entry의 크기 만한 history array를 만들었습니다.

이곳에는 pc % num\_predictor\_entry의 값 해당 브렌치의 과거 결과가 들어갑니다(최근 2개의 결과). 예를들어 pc % num\_predictor\_entry결과가 001일 때 아래 표에서 브렌치001에 접근해서 해당 브렌치의 최근 두개 결과를 참고한다. 과거 결과는 00,01,10,11총 4가지이다.

|  |  |
| --- | --- |
| 000 |  |
| 001 | 11(과거 결과가 모두 1,1) |
| 002 |  |
| ~ |  |
| 255 |  |

그리고 해당 결과에 맞게 해당하는 4개의 pred\_arr00, pred\_arr01, pred\_arr10, pred\_arr11에 접근한다. 위의 예시에서는 11로 결과가 나왔기 때문에 pred\_arr11에 접근하다. 아래는 pred\_arr11의 예시이다. 나머지 3개의 저장 공간도 똑같은 구조이다. 여기서도 마찬가지고 위의 브렌치 값 pc % num\_predictor\_entry의 결과 001에 접근하여 3bit prediction값을 가져와 prediction 값을 결정한다.

|  |  |
| --- | --- |
| 000 |  |
| 001 |  |
| 002 |  |
| ~ |  |
| 255 |  |

111,110,101,100은 prediction은 1 나머지는 0이다.

즉 해당 브렌치의 최근 두개 결과를 확인하고 그 결과를 바탕으로 해당 브렌치의 알맞은 pred\_arr에 접근하여 최근 결과값이 반영된 해당 브렌치의 3bit prediction값을 가져온다. 각각의 브렌치의 최근 결과를 중요시 하는Local branch predictor을 참고하였다.

값의 update는 pred\_arr는 맞고 틀림에 따라 3bit predicton의 값을 변경하여 해당 공간에 다시저장한다. 또한 history 저장공간은 실제 결과 값을 최신화하여 그 브렌치 어드레스에 업데이트한다.(ex)11이었는데 다음 실제 결과 값이 0일경우 10으로 최신화하여 저장한다.)

(2)history 저장공간 크기:

브렌치의 최근 결과 2개를 저장하는 history저장 공간이 있다. 이 배열의 총크기는num\_predictor\_entry의 값인 4914바이트이다. 그리고 최근 결과값에 따른 저장공간 pred\_arr00, pred\_arr01, pred\_arr10, pred\_arr11가 있고 이는 3bit predicton으로 작동하기 때문에 하나당 4914\*3= 14742바이트 그리고 총 4개 14742\*4=58968바이트이다. 그럼으로 총 58968+4914=63882바이트이다.

(3)과제를 수행하며:

저장공간이 넓어지는 것만으로 정확도 향상에 큰 영향을 준다는 것을 배울 수 있었고, 이번에는 각각의 브렌치가 자신의 최근 결과 값을 바탕으로 연관성이 있다는 Local branch predictor을 참고 했지만 global branch predictor등 다양한 predictor가 존재하고 이것을 하나만 사용하는 것보다 여러 개를 알맞은 조건을 적용하여 만들면 상당히 높은 정확도를 가지고 있는 predictor을 만들 수 있겠다는 생각할 수 있었다. 이렇게 규칙성을 참고하는 것만으로도 정확도 높은 예측을 할 수 있다는 것이 재밌는 점이었다.