

Travaux Pratiques Archi SLE-3A

Prédiction de branchements

1 Identification

Travail réalisé par Frédéric Pétrot

2 Prédicteur 1-bit : conception et résultats

2.1 code

Le prédicteur 1 bit est constitué d'un unique tableau de booleens. Son code est donné ci-dessous.

```
// Prédicteur naïf 1-bit qui recopie la dernière décision prise
// Ajout d'une information à la class branch_update à titre d'exemple
class my_update : public branch_update {
public:
    unsigned int index;
};

class my_predictor : public branch_predictor {
public:
    my_update u;
    branch_info bi;
    // 2^TABLE_BITS entrées de 2 bits
    // TABLE_BITS est passé sur la ligne de commande du compilateur
    unsigned char tab[1<<TABLE_BITS];

    // Constructeur
    my_predictor (void) {
        memset (tab, 0, sizeof (tab));
    }

    // Calcul de la prédiction
    branch_update *predict (branch_info & b) {
        bi = b;
        if (b.br_flags & BR_CONDITIONAL) {
            // Saut conditionnel
            // Récupération des bits de l'adresse pour indexer la table
            u.index = (b.address & ((1<<TABLE_BITS)-1));
            // Choix de la direction (la mise à jour se fait dans update
            u.direction_prediction (tab[u.index]);
        } else {
            // Saut inconditionnel
            u.direction_prediction (true);
        }
        // Adresse prédite, si on sait le faire
        u.target_prediction (0);
        return &u;
    }

    // Mise à jour de la table de prédiction
    void update (branch_update *u, bool taken, unsigned int target) {
        // Saut conditionnel
        // On peut forcer à true ou false pour avoir les extrêmes
        if (bi.br_flags & BR_CONDITIONAL) {
            tab[((my_update*)u)->index] = taken;
        }
    }
}
```

```
};  
// vim:se ts=3:
```

2.2 Résultats

Les résultats issus de la simulation sont les suivants.

2.3 Analyse

On voit une asymptote due à la disparition des collisions lorsque la taille du prédicteur augmente. Le coût du prédicteur est linéaire avec la taille du tableau, et il n'est pas raisonnable de dépasser 2^{16} éléments, d'autant que le gain à partir de 2^{12} devient très faible. Par ailleurs, il y a toujours moins de 7% de mauvaises prédictions, ce qui est remarquable pour une approche aussi simpliste.

3 Prédicteur 2-bits : conception et résultats

3.1 code

3.2 Résultats

3.3 Analyse

4 Prédicteur 2-bits avec historique : conception et résultats

4.1 code

4.2 Résultats

4.3 Analyse

5 Prédicteur 2-bits gshare : conception et résultats

5.1 code

5.2 Résultats

5.3 Analyse

6 Prédicteur corrélé : conception et résultats

6.1 code

6.2 Résultats

6.3 Analyse

7 Prédicteur local : conception et résultats

7.1 code

7.2 Résultats

7.3 Analyse