Departament d'Enginyeria Informàtica i Matemàtiques
[DEIM] Universitat Rovira i Virgili

Arquitectura de Computadors
AC
Curso 25/26
Primera Convocatòria
Pràctica 2: Comportament Predictors de Salt *Simplescalar*

Simulació processador Superescalar:

Predictors de Salt

La pràctica consisteix en l'anàlisi de l'estructura i el comportament de diferents predictors de salt en un processador Superescalar amb execució fora d'ordre (OoO). Per fer-ho, s'utilitzarà el simulador Simplescalar i s'avaluarà el comportament del processador executant els programes de prova SpecCPU2000.

La predicció de salts permet en un processador superescalar l'execució especulativa de les instruccions d'una de les dues alternatives d'un salt (instrucció de salt: branch). Així, el processador davant d'un hazard (risc) de control, pot aprofitar els cicles que perdria fins a saber la resolució del salt, en provar una de les dues alternatives i guanyar d'aquesta manera aquests cicles d'espera.

Direm que un predictor és bo quan encerta el camí correcte amb una alta probabilitat. Amb freqüència els predictors bons són complexos i utilitzen estructures de dades per emmagatzemar el comportament de les instruccions de salt. Un Branch Address/Target Buffer (BTB) emmagatzema les adreces efectives de les instruccions de salt a mesura que les va executant. Un Return Address Stack (RAS) emmagatzema en forma de pila les adreces de retorn de les instruccions Call. De la mateixa manera, un Branch History Register (BHR) emmagatzema la resolució (Taken / Not_Taken) dels darrers salts acumulats. Pot ser Global (GBHR) o associat a cada instrucció de salt de manera individual Per Address BHR (PaBHR). Per acabar, un Pattern History Table (PHT) emmagatzema la predicció que es farà en un possible salt. D'aquesta manera, manté un comptador saturat de 2 bits i utilitza el bit de més pes per a la predicció. Aquesta estructura pot ser global, individual a cada salt o fins i tot compartir-se entre alguns salts que tinguin comportaments semblants.

Simplescalar, mitjançant els simuladors sim-bpred i sim-outorder, permet configurar cadascuna d'aquestes quatre estructures per analitzar el comportament de diferents configuracions.

Per a aquesta pràctica, es farà servir un subconjunt de 5 benchmarks disponibles a través de la imatge i també de l'espai moodle de l'assignatura.

Comentaris

- El grup de pràctiques estarà format per 3 membres: **2 PERSONES** i ChatGPT (o equivalent :-)**. Si el feu servir**, el text generat per l'eina constarà a l'informe en lletra ***cursiva***.
- Es realitzarà una entrevista presencial amb els integrants del grup (humans) a la sessió de laboratori que tenen assignada. On es defensarà amb arguments la feina realitzada a qualsevol part de la pràctica.
- L'informe (obligatòriament en PDF), el vídeo resum de la pràctica i els fitxers auxiliars que s'hagin utilitzats per les diferents eines/simuladors es guardaran al moodle abans de fer l'entrevista (units en format ZIP).

Especificació

La tasca d'aquesta pràctica es divideix en dues fases: (1) Estudi dels predictors que es poden simular amb el Simplescalar i (2) Modificació del Simplescalar per simular un nou predictor de salts.

1a Fase:

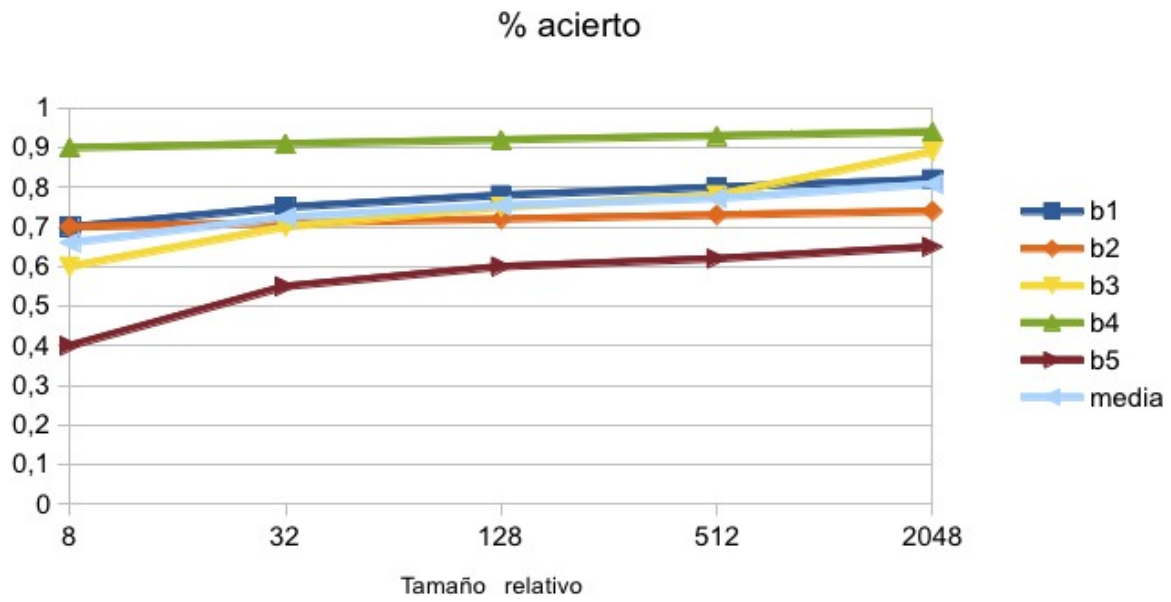
Els predictors que implementa Simplescalar son: nottaken, taken, perfect, bimodal, 2-level y comb.

Es faran servir per avaluar el comportament quant a IPC i percentatge d'encerts de les següents alternatives de predictors de salts:

- **nottaken**: opció: -bpred nottaken
- **taken**: opció: -bpred taken
- **perfect**: opció: -bpred perfect
- **bimodal**: opció: -bpred bimod
  - Mida del PHT <l2-size> : 8,32,128,512,2048
  - -bpred:bimod X
- **Gshare**: opción: -bpred 2lev
  - Mida del BHR <l1-size>: 1 i del PHT<l2-size>: 8,32,128,512,2048
  - -bpred:2lev 1 <X> <$\log_2$X> 1
- **Gag (Gselect):** opció: -bpred 2lev
  - Mida del BHR <l1-size>: 1 i del PHT<l2-size>: 8,32,128,512,2048
  - -bpred:2lev 1 <X> <$\log_2$X> 0
- **Pag**: opció: -bpred 2lev
  - Mida del BHR <l1-size> i del PHT<l2-size> respectivament: (4-4), (8-16), (16-64), (32-256), (64-1024), (32-2048) son (Y-X)
  - -bpred:2lev <Y> <X> <$\log_2$X> 0

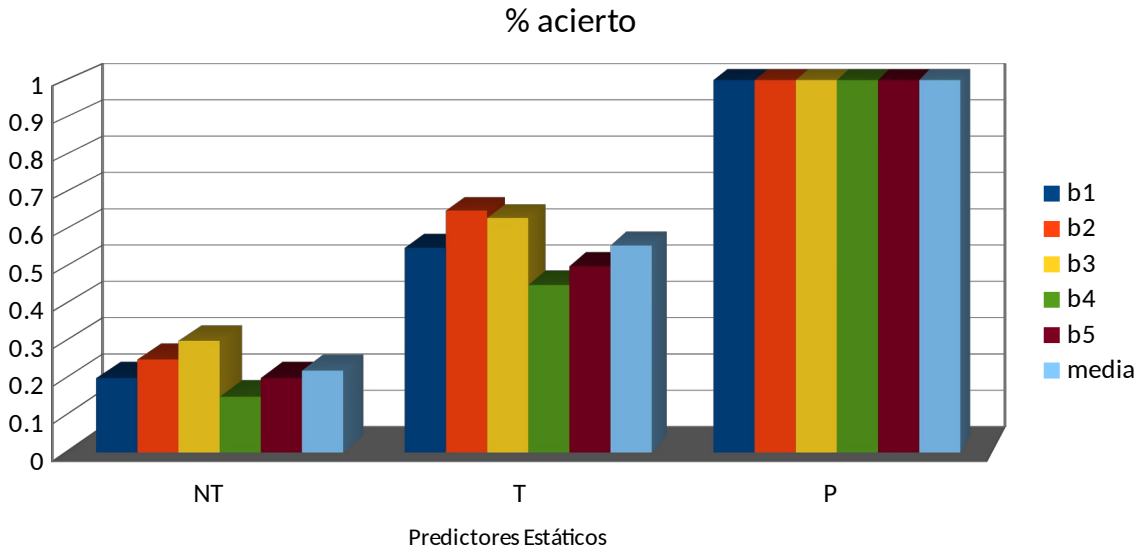
L'objectiu és observar i generar gràfiques que mostrin el comportament de l'IPC i el percentatge d'encert (.bpred_dir_rate) per a diferents valors de configuració dels predictors amb els cinc benchmarks que es disposen de la pràctica anterior.

Per als diferents estudis que s'han de realitzar, i si no s'indica el contrari, tingueu en compte els comentaris següents:



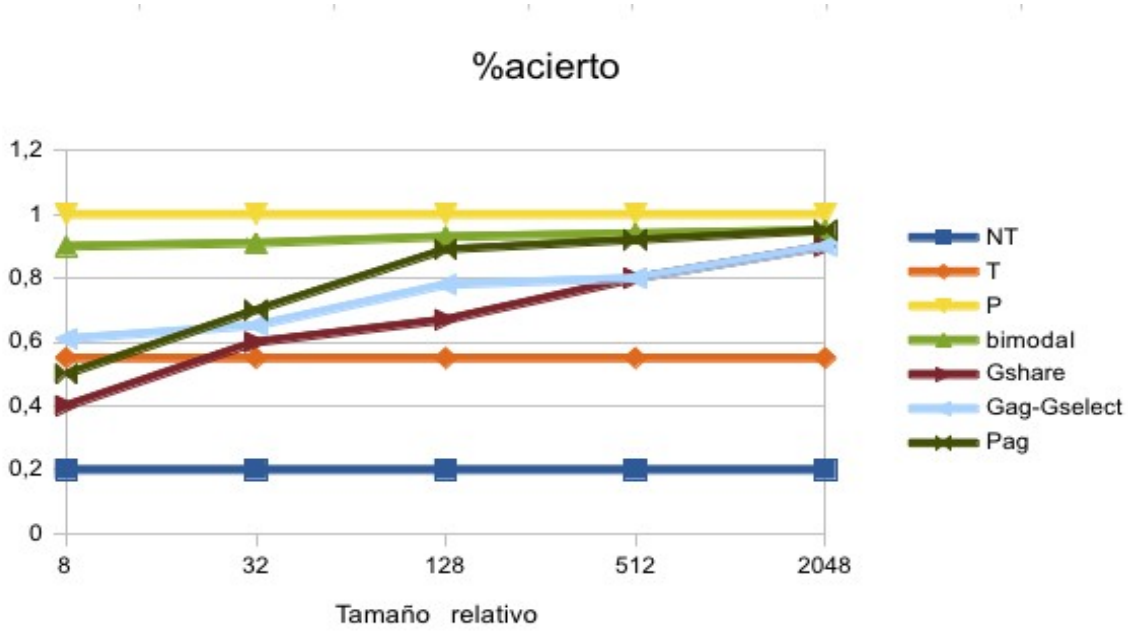
1) Es mostrarà una gràfica de cada paràmetre per a cadascun dels set predictors indicats anteriorment, on a l'eix de les Y hi haurà el paràmetre i a l'eix de les X les diferents configuracions en mida del predictor estudiat. A la mateixa gràfica apareixeran els cinc benchmarks i la mitjana dels mateixos.

2) Per als predictors estàtics es poden resumir tots en una única gràfica de la manera següent:



3) Per a cada estudi es presentaran gràfiques amb els resultats individuals de cada benchmark i amb la mitjana de tots.

4) També es generaran gràfiques resumeixen del comportament de cadascun dels predictors. A la mateixa gràfica apareixeran els set predictors tenint a l'eix de les X les diferents configuracions.



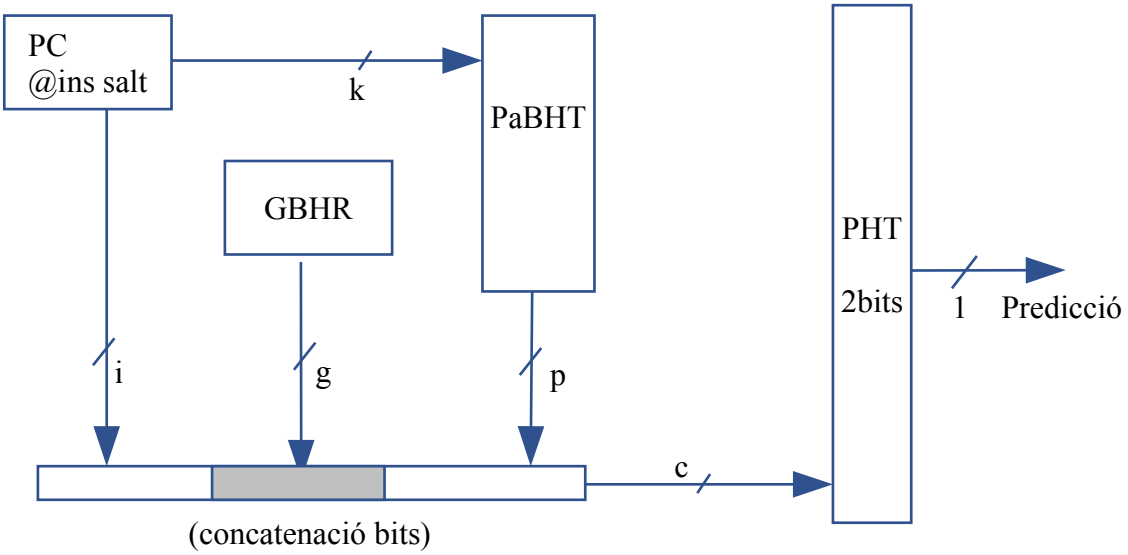
5) Per a cada benchmark simulat, se saltaran 100 milions d'instruccions i es recol·lectaran les estadístiques per als 100 milions següents. A més, es faran servir les dades d'entrada REF.

6) La mida de BTB i RAS no es modificaran del valor per defecte.

7) Per a la resta de paràmetres del simulador s'utilitzaran els valors per defecte, a excepció del bus d'accés a memòria (de 32 bytes) i de la seva latència (300 cicles inicials i 2 per accés consecutiu).

2 fase:

Implementació del predictor de salts *Alloyed*.



El predictor manté la informació de salts executats en quatre estructures.

1. Global Branch History Register (GBHR) desa la història dels darrers 'g' salts que s'han executat al processador. Guarda un 1 si ha estat Taken i un 0 si ha estat NotTaken. (Internament serà un registre de 32 bits on agafarem 'g' bits o 'c' segons ens interessi)
2. Per address Branch History Table (PaBHT) és una taula de dimensió 2k entrades i cadascuna amb 'p' bits. L'objectiu d'aquesta taula és guardar la història dels darrers 'p' salts de cada instrucció per separat. S'utilitza la direcció en memòria de les instruccions de salt com a identificador d'elles mateixes. Com que la taula té un nombre limitat d'entrades, s'utilitzen únicament els 'k' bits de menor pes per seleccionar una entrada per a una instrucció de salt. Com a inconvenient, passa que aquelles instruccions de salt que tinguin els mateixos 'k' darrers bits compartiran la mateixa entrada i emmagatzemaran una història barrejada (Aliasing).
3. Pattern History Table (PHT) és una taula on s'emmagatzemen dos bits per implementar un comptador saturat del comportament dels salts. De manera que cada cop que el salt és efectiu Taken, s'incrementa i si el salt no és efectiu NotTaken es decrementa. Així s'obtenen quatre estats on dos (els que tenen el bit de més pes a '1') prediuen que el salt serà efectiu i els altres dos (els que tenen el bit de menor pes a '0') prediuen que la instrucció de salt no saltarà (comptador de 2-bits saturats).

Cada cop que s'executa un salt aquestes estructures s'hi accedeixen dues vegades. La primera per obtenir una predicció i actuar en conseqüència al pipeline i la segona per actualitzar el comportament del salt executat i afinar subsegüents prediccions.

La predicció s'obté a partir de la funció «Majoria» del bit de més pes de l'entrada del PHT seleccionada de cada taula.

Els paràmetres que cal definir en aquest predictor són:

1. Nombre de bits del GBHR → defineix 'g'

2. Nombre d'entrades de PaBHT i bits de cada entrada → defineix 'k' i 'p'
3. Nombre d'entrades del PHT→ defineix 'c'
4. A partir d'aquests valors es dedueix 'i' com a 'c'-'g'-'p'. Que no pot ser inferior a 1.

Els valors inicials d'aquestes estructures són bits a '1'. D'aquesta manera, la predicció de qualsevol salt inicial serà Taken.

Per modificar el simplescalar i afegir un nou predictor es pot fixar en la implementació d'un ja existent i duplicar afegir el necessari. En aquest cas, el predictor **2lev** és prou semblant i serà el model a seguir.

Les parts de simplescalar a modificar són:

- A bpred.h:
  - Afegir el nou predictor a la llista d'enum bpred_class.
  - A la definició de tipus dels predictors afegir a l'estructura struct bpred_dir_t dins de la subestructura two les variables i apuntadors extres. Tingueu en compte que level-1 podria ser PaBHT i level-2 podria ser PHT replicat tres vegades.
- A sim-outorder.c:
  - Registrar a sim_reg_options els paràmetres de configuració del predictor: amb `opt_reg_int_list(opt,"-bpred:alloy",.....`

    Cal definir predictor_config i predictor_nelt.
  - Afegir a sim_check_options la lectura dels paràmetres del predictor. Trucar a la funció de creació del predictor bpred_create(,,,,,,,). Fixar-se al 2lev.
- A bpred.c:
  - Afegir a bpred_create i bpred_dir_create un nou cas (case) d'inicialització de predictors on a partir dels paràmetres definits del predictor, es reserva la memòria necessària i s'inicialitzen les variables i les taules als valors necessaris.
  - Afegir a bpred_config i bpred_dir_config un nou cas (case) per mostrar per la sortida del simulador la configuració del predictor.
  - Afegir a bpred_reg_stats un nou cas (case) per poder veure el resultat de la simulació.
  - Afegir a bpred_lookup i bpred_dir_lookup un nou cas (case) per obtenir la direcció de la següent instrucció, atenent a la predicció que realitza a partir de la instrucció de salt que s'ha fet el fetch. (sempre que es trobi al BTB). Aclariment: bpred_dir_lookup torna l'adreça de l'entrada de la taula PHT (L2 Table) i bpred_lookup la guarda al camp pdir1 de la pròpia instrucció perquè a la següent trucada a bpred_update ja tingui calculada la posició en PHT (és una optimització)
  - Afegir a bpred_update el cas per actualitzar el GBHR, el PaBHT (L1 Table), i el PHT (L2 Table). Cal tenir en compte aquesta darrera taula es pot actualitzar a partir de l'adreça guardada anteriorment (l'optimització)

En general la simulació a simplescalar comença a main.c. Per a les funcions que hem esmentat s'anomena sim_reg_options(), després sim_check_options() i després sim_main(). Totes tres estan implementades dins del codi principal del simulador sim-outorder i a partir d'aquest instant comença la simulació.

La funció sim_main implementa el pipeline del superescalar. Truca a diferents funcions per simular el comportament del processador i d'entre elles ens interessa la funció ruu_fetch().

ruu_fetch() realitza la lectura d'instruccions de memòria, simula la memòria cau, el TLB i el predictor de salts. En llegir la instrucció la descodifica parcialment i si s'adona que és una instrucció de salt truca a bpred_lookup() perquè li torni l'adreça de la següent instrucció que segons la seva predicció s'ha de continuar buscant en memòria.

La funció bpred_update() actualitza la informació del predictor a partir de l'execució real de la instrucció de salt. S'anomena generalment des del ruu_commit() però també es pot trucar abans des del ruu_writeback() o fins i tot des de ruu_dispatch().

Un cop implementat el predictor s'afegirà el comportament del mateix a les gràfiques anteriors.

Els paràmetres a tenir en compte del nou predictor:

- **Alloy**: opció -bpred alloy
  - Mida del PaBHT <l1-size> i del PHT<l2-size> respectivament: (8-8), (16-32), (32-128), (64-512), (128-2048), (64-4096) son (Y-X)
  - Ample del GBHR <g> i PaBHT <p> que concatenata amb 'i' formen 'c' i serán: (1-1), (2-2), (3-2), (3-3), (4-4) i (4-4)
  - Així la configuració de c=i+g+p quedaria: (1+1+1→3), (1+2+2→5), (2+3+2→7), (3+3+3→9), (3+4+4→11), (4+4+4→12).
  - -bpred:alloy <y> <X> <p> <g> 0

Responeu a l'informe: A que un altre predictor Gag, Pag, Gap, Pap s'assembla? Amb quins paràmetres? Simuleu i compareu-ho amb el alloy aquí implementat.