

PRÀCTICA 2 ARQUITECTURA DE COMPUTADORS: PREDICTORS DE SALT



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Alumnes: Joel Teodoro Gómez
Carlos Martínez García-Villarrubia

Contenido

FASE 1.....	3
1.1 Predictors estàtics	4
IPC	4
Taxa d'encert	5
1.1 Predictors dinàmics	6
Bimodal.....	6
Gshare	8
Gag	10
Pag.....	12
1.3 Comparació final	14
FASE 2.....	15
2.1 Modificacions bpred.h.....	15
2.2 Modificacions sim-outorder.c.....	16
2.2 Modificacions bpred.c	16
2.2.1 Funció bpred_create.....	16
2.2.2 Funció bpred_lookup.....	17
2.2.3 Funció bpred_update	18
2.3 Comparació de resultats respecte la fase 1	21
CASCADE.....	21
COMB	23
CONCLUSIÓ FINAL.....	25

FASE 1

Predictors estàtics

- Taken
- Not Taken
- Perfect

Predictors dinàmics

- One-level
 - Bimodal
- Two-level
 - Gshare
 - Gag
 - Pag

Per a avaluar els resultats de cada predictor, observarem el percentatge d'encert i l'IPC obtingut un cop executats els següents benchmarks:

- ammp
- eon
- quake
- gap
- mesa

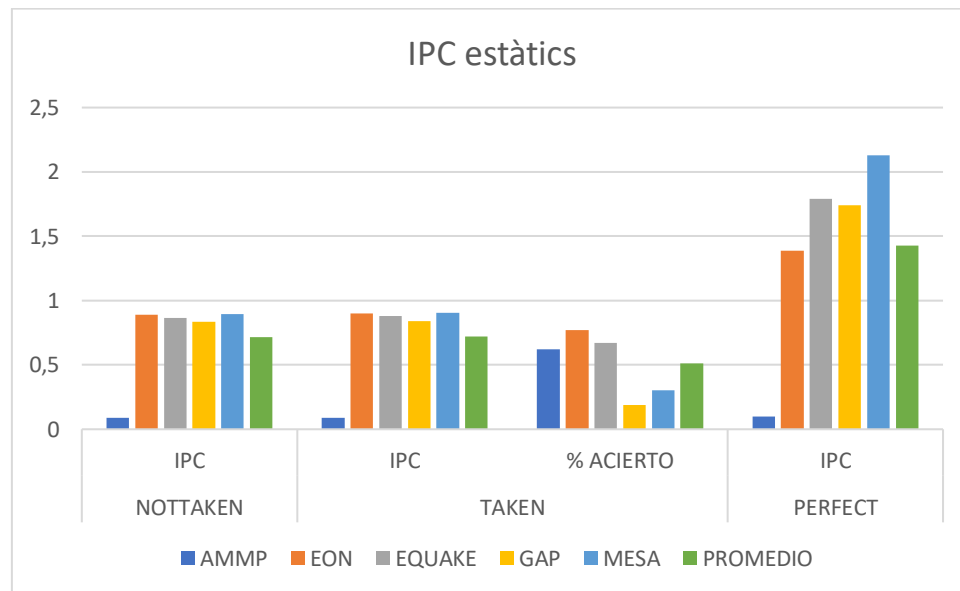
1.1 Predictors estàtics

IPC

Com veurem a continuació, els predictors estàtics són els que en han donat un pitjor resultat, ja que van ser dels primers en aparèixer i utilitzen una lògica molt més simple a l'hora de decidir quin salt agafar.

Bàsicament, els predictors estàtics decideixen, en temps de compilació, si s'agafarà el salt o no.

	Not taken	Taken	Perfect
	IPC	IPC	IPC
AMMP	0,0897	0,0899	0,0973
EON	0,8896	0,8982	1,386
EQUAKE	0,8647	0,8806	1,7914
GAP	0,8364	0,8414	1,7388
MESA	0,8938	0,9057	2,1271
PROMEDIO	0,71484	0,72316	1,42812



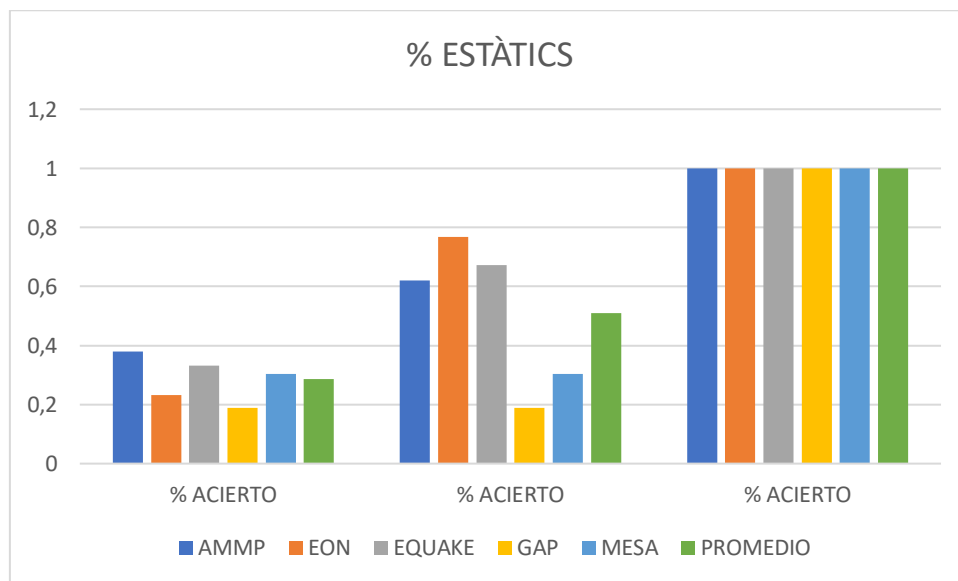
Com podem veure a la següent gràfica, la diferencia de rendiment entre testos és molt notable, sobretot si parem atenció a AMMP, això es pot deure a que ammp compta amb molts més salts condicionals que la resta i per això l'IPC que obtenim és molt més inferior.

A més, podem veure que el predictor *perfect* aporta un millor rendiment en tots els testos, tret de l'AMMP.

Si comparem taken i not taken, podem observar que els resultats són pràcticament idèntics, i poc eficients, ja que la predicció al cap i a la fi serà, com el seu nom indica, agafat o no agafat.

Taxa d'encert

	Not taken	Taken	Perfect
	% ACIERTO	% ACIERTO	% ACIERTO
AMMP	0,3804	0,6196	1
EON	0,2317	0,7683	1
EQUAKE	0,3318	0,6723	1
GAP	0,1896	0,1896	1
MESA	0,3026	0,3026	1
PROMEDIO	0,28722	0,51048	1



En quant al % d'encert que hem obtingut als diferents predictors, observem que el perfecte ens atorga un % superior, arribant al 100% d'encerts, respecte al taken i not taken. Això es deu a que la lògica és lleugerament més completa que els altres dos predictors esmentats, tot i que quan ho comparem amb els predictors dinàmics, veurem la significativa diferència.

En conclusió, el perfecte representaria el predictor "ideal", el taken té una taxa relativament baixa en comparació als predictors d'avui dia, i el not taken seria molt poc òptim i res convenient. Si ens adonem, la taxa d'encert que ens ofereix és literalment de la meitat que la del taken, així representant un 28.7% d'encerts respecte el 51% del taken.

1.1 Predictors dinàmics

Els predictors dinàmics, a diferència dels estàtics, ofereixen un % molt superior d'encerts.

Això es deu, entre altres coses, a les heurístiques més complexes que s'apliquen a l'hora de decidir si agafarem el salt o no. A més, fem ús d'estructures com el BTB, o BHT que ens permeten guardar les direccions de destí i els salts fets prèviament, respectivament.

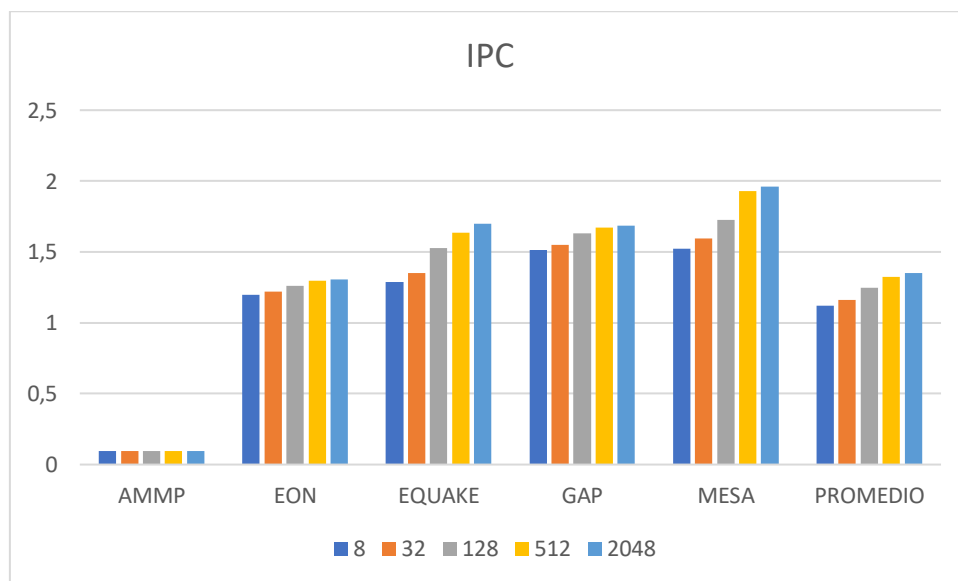
El fet de comptar amb aquestes heurístiques més complexes, ens aporten una gran millora, podent arribar fins a un 99,75% d'encert en els millors casos.

A continuació veurem alguns dels resultats que hem obtingut amb els diferents predictors dinàmics proposats a la pràctica, i una comparació dels resultats per a veure quin d'ells ens ofereix un millor rendiment i taxa d'encert.

Bimodal

IPC

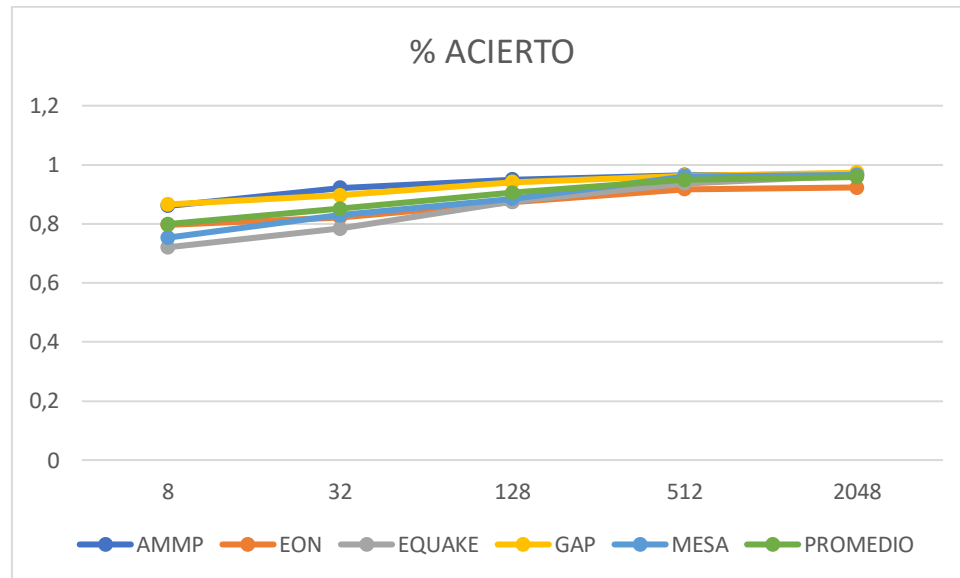
X	8	32	128	512	2048
AMMP	0,096	0,0965	0,0967	0,0969	0,0969
EON	1,1967	1,2221	1,2603	1,2962	1,306
EQUAKE	1,2868	1,3494	1,5258	1,6378	1,698
GAP	1,5116	1,5512	1,6306	1,6698	1,6872
MESA	1,522	1,5951	1,7276	1,9291	1,9588
PROMEDIO	1,12262	1,16286	1,2482	1,32596	1,34938



En quant a l'IPC obtingut al executar el test bimodal, obtenim un rendiment semblant als predictors estàtics. Veiem que el rendiment de l'AMMP continua sent molt baix, mentre que el test GAP manté un rendiment alt, lleugerament per sota del MESA, que arriba al seu top quan la mida del predictor és 2048.

% ENCERT

X	8	32	128	512	2048
AMMP	0,8605	0,9213	0,9495	0,9649	0,9669
EON	0,7964	0,822	0,8742	0,9174	0,9232
EQUAKE	0,7206	0,7844	0,8754	0,9366	0,9639
GAP	0,8658	0,8963	0,9416	0,9633	0,9732
MESA	0,7533	0,8296	0,8848	0,9604	0,9657
PROMEDIO	0,79932	0,85072	0,9051	0,94852	0,95858



Un cop executats els benchmarks fent servir el predictor bimodal, veiem una diferència considerable respecte als % d'encert dels predictors estàtics.

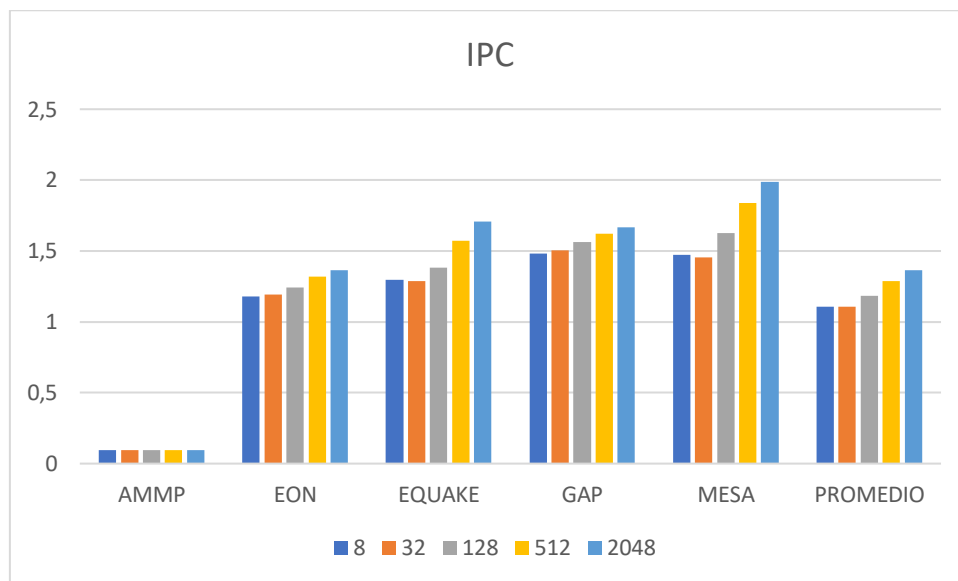
Primer de tot, podem veure que en cap cas la taxa d'encert baixa per sota del 70%, i en casos com GAP, veiem que la taxa d'encert és del 97,32% en la seva configuració més alta (2048).

En segona posició podem veure situat AMMP, el qual puja d'un 50% obtingut als estàtics, a no baixar del 86% en el cas del bimodal. En última posició podríem veure EON, tot i que no presenta un mal % d'encert, si el comparem directament amb la resta, presenta un % lleugerament inferior.

Gshare

IPC

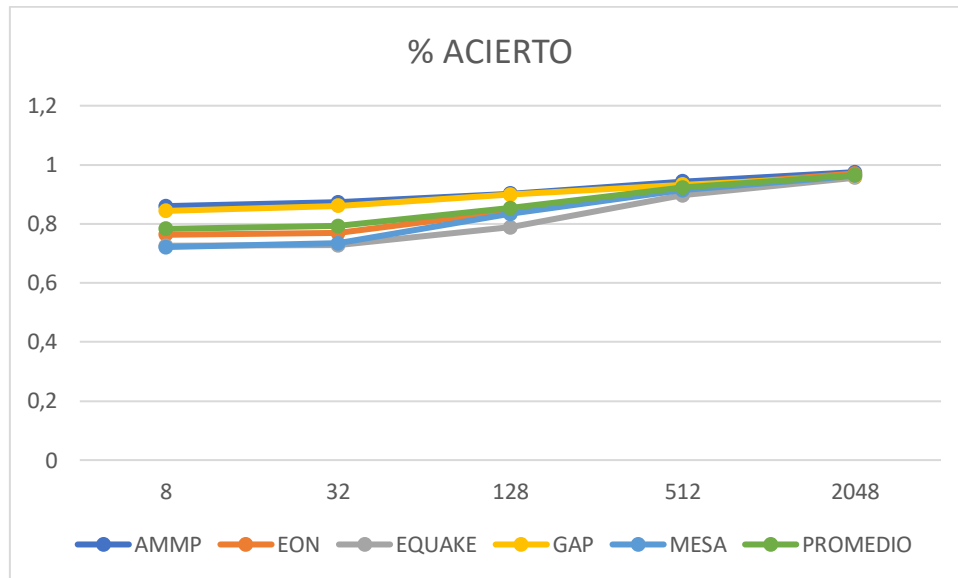
X	8	32	128	512	2048
AMMP	0,0957	0,0959	0,0962	0,0966	0,097
EON	1,1793	1,1926	1,2428	1,3173	1,3642
EQUAKE	1,2951	1,2883	1,3801	1,5722	1,7074
GAP	1,4795	1,5042	1,563	1,6196	1,668
MESA	1,4743	1,4537	1,6251	1,8394	1,9894
PROMEDIO	1,10478	1,10694	1,18144	1,28902	1,3652



Un cop executats els testos mitjançant el processador Gshare, tot i fer servir una lògica lleugerament respecte al predictor bimodal, obtenim un rendiment pràcticament igual. Un rendiment molt dolent en el cas d'AMMP, el top d'IPC l'assolim al test MESA, i en segona posició ens trobem el test EON, on el punt màxim el trobem quan la configuració del predictor val 2048.

% ENCERT

X	8	32	128	512	2048
AMMP	0,86	0,8728	0,9026	0,943	0,9751
EON	0,763	0,769	0,8426	0,9255	0,9706
EQUAKE	0,7258	0,7275	0,788	0,8967	0,956
GAP	0,8439	0,8609	0,8993	0,9326	0,9604
MESA	0,7214	0,7346	0,8339	0,9137	0,9623
PROMEDIO	0,78282	0,79296	0,85328	0,9223	0,96488



En quant al % d'encert, ens trobem amb un % lleugerament inferior, trobant únicament unes dècimes de diferència respecte el test bimodal.

Si ens fixem, el % d'encert màxim el trobem al test AMMP, trobant en segon puesto el GAP.

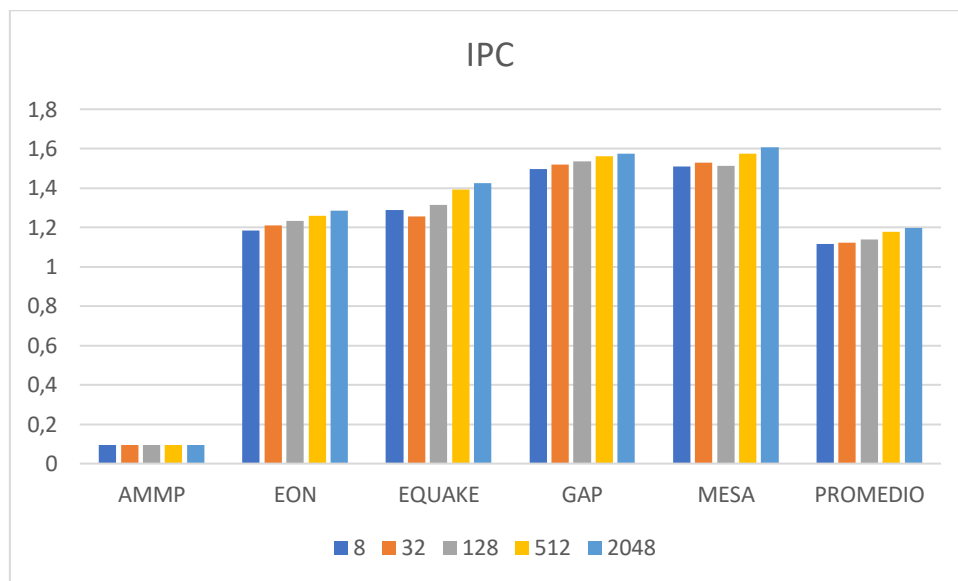
Com podem observar a la gràfica, els % d'encerts comencen a pujar quan la configuració del predictor passa de valdre 32 a 128.

També podem destacar que el test que ara trobem en últim puesto és equake, amb una mitja d'encert del 81,8%.

Gag

IPC

X	8	32	128	512	2048
AMMP	0,095	0,0951	0,0951	0,0954	0,0932
EON	1,1847	1,2091	1,2334	1,2596	1,2846
EQUAKE	1,2892	1,2544	1,3149	1,392	1,4258
GAP	1,4962	1,5193	1,5342	1,5612	1,5738
MESA	1,51	1,5274	1,5112	1,5735	1,6079
PROMEDIO	1,11502	1,12106	1,13776	1,17634	1,19706

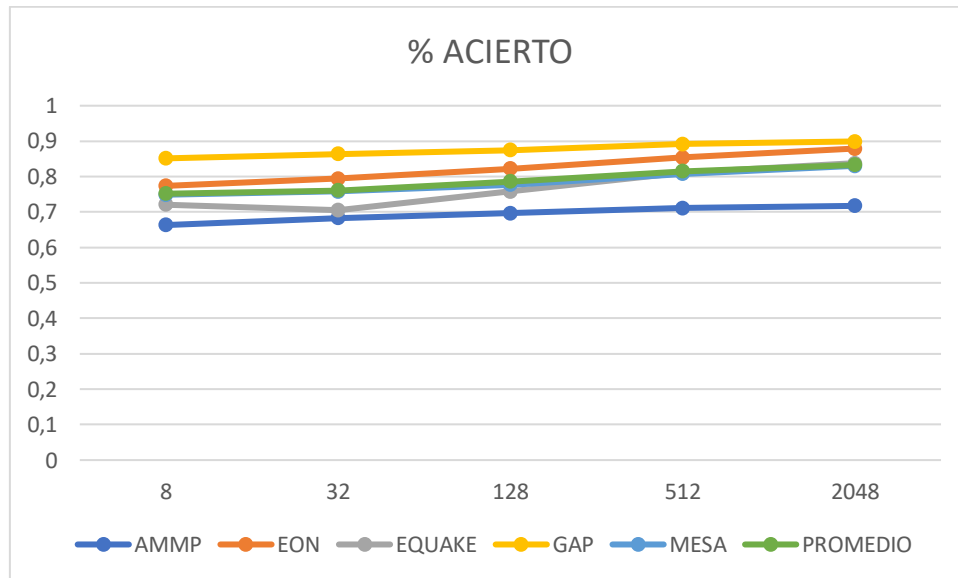


En quant al rendiment obtingut al utilitzar el predictor, obtenim que, com als altres tests, l'amp ens retorna el pitjor de tots els rendiments. El gap, no obstant, ocupa la segona posició, donant un IPC de 1.57 en la seva millor configuració, respecte de l'1,61 del MESA.

En tercera posició ens trobem el test EQUAKE, que no ens aporta un mal rendiment del tot, ja que en la seva millor configuració (2048) ens ofereix un IPC d'1,43. En general, ja podem veure que el rendiment acostuma a ser lleugerament superior al obtingut amb alguns dels predictors estàtics, com el taken o not taken.

% ENCERT

X	8	32	128	512	2048
AMMP	0,6632	0,683	0,6968	0,7107	0,7173
EON	0,7738	0,7941	0,8223	0,8536	0,8791
EQUAKE	0,7209	0,7052	0,7578	0,8116	0,8379
GAP	0,8514	0,8639	0,8749	0,8915	0,8992
MESA	0,7489	0,758	0,7768	0,808	0,8303
PROMEDIO	0,75164	0,76084	0,78572	0,81508	0,83276



Quant al percentatge d'encert, el que obtenim és també un increment respecte a algun dels predictors estàtics, no baixant del 66% d'encert que trobem a l'ammp amb la seva configuració més baixa (8).

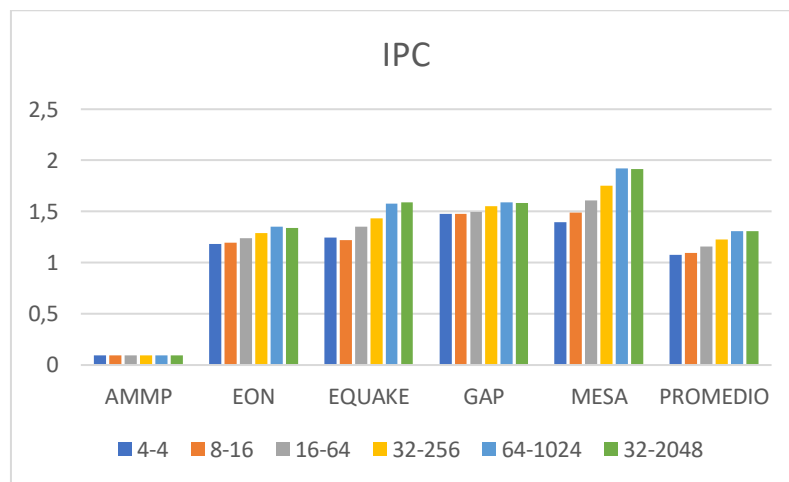
En general, no ens aporta un mal % d'encerts, tot i que continua sent lleugerament inferior al que ens aporta el predictor pag. Veiem que a mesura que augmenta la configuració del predictor, també augmenta com a conseqüència el % d'encert.

El test on trobem millor % d'encert és GAP, donant un 89% d'encert quan la configuració del predictor és 2048.

Pag

IPC

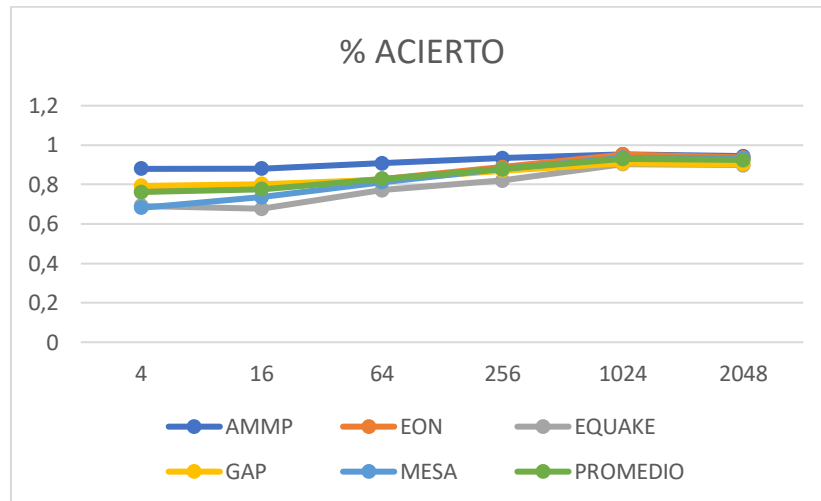
X	4	16	64	256	1024	2048
AMMP	0,0952	0,096	0,0962	0,0965	0,0967	0,0966
EON	1,18	1,1974	1,2396	1,2909	1,3542	1,3383
EQUAKE	1,2458	1,2224	1,3537	1,4318	1,5753	1,5885
GAP	1,4743	1,48	1,4976	1,5495	1,5921	1,5855
MESA	1,3973	1,4903	1,6081	1,7538	1,9217	1,9179
PROMEDIO	1,07852	1,09722	1,15904	1,2245	1,308	1,30536



Per últim, trobem que el predictor PAG, ens ofereix un bon rendiment quan parem atenció al MESA, donant un IPC de l'1,92. En segona posició trobem el test EON, on amb la seva millor configuració (1024) trobem un IPC del 1,59. Un altre cop, trobem en última posició trobem el test ammp, amb un IPC no superior al 0,097. A més trobem resultats similars si comparem l'IPC del eon amb el de l'equake.

% ENCERT

X	4	16	64	256	1024	2048
AMMP	0,881	0,8804	0,9081	0,9335	0,953	0,9442
EON	0,7639	0,7864	0,8276	0,8886	0,952	0,9375
EQUAKE	0,6923	0,6775	0,7708	0,8208	0,9032	0,899
GAP	0,793	0,803	0,8234	0,8702	0,9081	0,9026
MESA	0,6817	0,7362	0,8125	0,8789	0,935	0,9337
PROMEDIO	0,76238	0,7767	0,82848	0,8784	0,93026	0,9234



Quant al % d'encert del pag, ens trobem en que, en aquest cas, el test ammp es troba en primera posició, donant un % d'encert del 94% en la seva segona millor configuració (1024).

La última posició la ocupa el test EON en aquest cas, proporcionant-nos una mitja % d'encert del 79,39%.

1.3 Comparació final

En general, a la següent gràfica podem veure que el predictor que millors resultats ens ha donat ha sigut el bimodal si no tenim en compte el predictor perfect. En segona posició trobaríem el gshare, amb una mitja d'encert del 86,32% respecte el 89,24% del bimodal. Si fem una classificació obtenim els següents resultats:

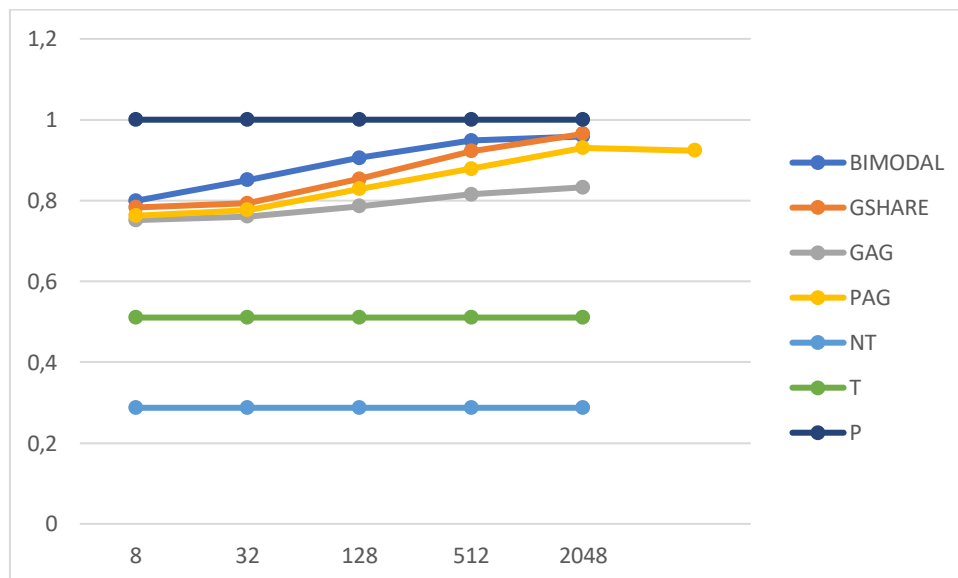
1. Perfect
2. Bimodal
3. Gshare
4. Pag
5. Gag
6. Taken
7. Not taken

En conclusió, com és obvi, els predictors dinàmics donen un molt millor resultat que els estàtics. La seva lògica més complexa en temps d'execució per considerar si saltar o no és el fet diferencial a l'hora d'obtenir aquests millors resultats.

Podríem equiparar els predictors dinàmics a una mena de Machine learning, ja que aprenen dels salts presos prèviament i dels errors, per a avaluar les decisions futures.

Per acabar, podem dir que els predictors dinàmics són més utilitzats a gran escala quan ens interessa decidir en temps d'execució si saltarem o no.

No obstant, els estàtics van ser una gran revolució en el seu temps, ja que van suposar un abans i un després en el món de la computació, tot i que avui dia s'utilitzen com a un "complement" del estàtics.



FASE 2

Al fer aquesta segona part de la pràctica, ens hem adonat de la complexitat del simulador SimpleScalar.

Hem aconseguit dur a terme una implementació d'aquest predictor en cascada, que aconsegueix un percentatge d'encert del 97% en el millor dels seus casos.

2.1 Modificacions bpred.h

```
/* branch predictor types */
enum bpred_class {
    BPredComb,                /* combined predictor (McFarling) */
    BPred2Level,              /* 2-level correlating pred w/2-bit counters */
    BPred2bit,                /* 2-bit saturating cntr pred (dir mapped) */
    BpredCascade,             /* CASCADE PREDICTOR */
    BPredTaken,               /* static predict taken */
    BPredNotTaken,            /* static predict not taken */
    BPred_NUM
};
```

Hem afegit el predictor cascada a la llista de enum bpred_class.

Com que ja tenim creades les estructures del bimodal i el gshare no hem modificat res més en aquest fitxer.

2.2 Modificacions sim-outorder.c

Com que no hem afegit cap paràmetre de configuració nou, no ha sigut necessari registrar en `sim_reg_options` el nou predictor. Per tant, no hem definit ni el `predictor_config` ni el `predictor_nelt`.

```
opt_reg_string(odf, "-bpred",
"branch predictor type {nottaken|taken|perfect|bimod|2lev|comb|cascade}", //modificat
&pred_type, /* default */"bimod",
/* print */TRUE, /* format */NULL);
```

Afegim el nom de cascade per identificar més endavant el tipus de predictor. Com que no hem afegit res nou no fa falta.

```
else if (!mystrcmp(pred_type, "cascade")){
    pred = bpred_create(BpredCascade,
    /* bimod table size */bimod_config[0],
    /* l1 size */twolev_config[0],
    /* l2 size */twolev_config[1],
    /* meta table size */0, //treiem el gselect
    /* history reg size */twolev_config[2],
    /* history xor address */twolev_config[3],
    /* btb sets */btb_config[0],
    /* btb assoc */btb_config[1],
    /* ret-addr stack size */ras_size);
}
```

Afegim la lectura dels paràmetres del predictor. Bàsicament és el mateix que el comb amb la mida de la meta-table a 0, ja que no utilitzarem el metapredictor en la implementació del predictor cascade.

2.2 Modificacions bpred.c

2.2.1 Funció bpred_create

```
case BpredCascade:
    /* bimodal component */
    pred->dirpred.bimod =
        bpred_dir_create(BPred2bit, bimod_size, 0, 0, 0);

    /* 2-level component */
    pred->dirpred.twolev =
        bpred_dir_create(BPred2Level, l1size, l2size, shift_width, xor);

    for (int cnt = 0; cnt < l2size; cnt++)
    {
        pred->dirpred.twolev->config.two.l2table[cnt] = 4; //inicialitzem a 4
    }

    for (int cnt = 0; cnt < bimod_size; cnt++)
    {
        pred->dirpred.bimod->config.bimod.table[cnt] = 3; //inicialitzem a 'l1'= 3
    }

    break;
```

Afegim a la funció `bpred_create` un nou cas i creem les estructures. Com que a la pràctica ens demanen que les estructures estiguin inicialitzades a 3 excepte la PHTg de la gshare ho fem sota després de crear les estructures. En cas que modifiquéssim la twolevel i la bimodal dins de `bpred_dir_create` el que provocariem és que aplicariem els canvis a tots els predictors com vam fer al principi.


```
/* allocate ret-addr stack */
switch (class)
{
case BPredComb:
case BpredCascade:
case BPred2Level:
case BPred2bit:
{
    int i;
```

També hem afegit un nou cas a la funció `bpred_config` per mostrar les configuracions del `cascade`.

2.2.2 Funció bpred lookup

La funció `dir lookup` selecciona el predictor que volem utilitzar per a fer al predicció.

Abans de entrar al lookup a la funció `bpred_reg_stats` hem afegit un nou cas pel cascade per registrar els estats del predictor de salt.

```

/* register branch predictor stats */
void bpred_reg_stats(struct bpred_t *pred,
                    struct stat_sdb_t *sdb)
{
    char buf[512], buf1[512], *name;

    /* get a name for this predictor */
    switch (pred->class)
    {
    case BPredComb:
        name = "bpred_comb";
        break;
    case BpredCascade:
        name = "bpred_cascade";
        break;
    }
}

```

Afegim un nou cas amb una or “|” al if ja que comb i cascade comparteixen les mateixes estructures.

```

    if (pred->class == BPredComb || pred->class == BpredCascade)
    {
        sprintf(buf, "%s.used_bimod", name);
        stat_reg_counter(sdb, buf,
            "total number of bimodal predictions used",
            &pred->used_bimod, 0, NULL);
        sprintf(buf, "%s.used_2lev", name);
        stat_reg_counter(sdb, buf,
            "total number of 2-level predictions used",
            &pred->used_2lev, 0, NULL);
    }
}

```

Dins de `bpred_lookup` afegim un nou cas pel cascade. El que fem és similar al comb però ens abstenim de la taula meta. Primer creem dos `char*` un que apuntarà al comptador saturat de la bimodal i l'altre apuntarà al comptador saturat de la PHTg de la gshare.

Amb `dir_lookup` obtenim els punters dels comptadors saturats i les igulem a les variables que prèviament hem creat.

Ens demanen que primer mirem a la gshare, en cas d'obtenir miss anem a la bimodal. En cas contrari el que fem és comparar si el valor de la gshare està buit. Si no està buit voldrà dir que prèviament ha fallat en la predicció la bimodal i utilitzarem la gshare per aquell salt, ja que pot ser massa complex aquell salt en específic per la bimodal per a realitzar-lo.

```
case BpredCascade:
    if ((MD_OP_FLAGS(op) & (F_CTRL | F_UNCOND)) != (F_CTRL | F_UNCOND))
    {
        char *bimod, *twolev;
        bimod = bpred_dir_lookup(pred->dirpred.bimod, baddr);
        twolev = bpred_dir_lookup(pred->dirpred.twolev, baddr);
        dir_update_ptr->dir.bimod = *bimod;
        dir_update_ptr->dir.twolev = *twolev;

        if (*twolev == 4)
        {
            dir_update_ptr->pdir1 = bimod;
            dir_update_ptr->pdir2 = twolev;
        }
        else
        {
            dir_update_ptr->pdir1 = twolev;
            dir_update_ptr->pdir2 = bimod;
        }
    }
    break;
```

2.2.3 Funció `bpred_update`

Aquesta és la funció més complexa de programar. Ja que és la que major lògica requereix. En aquesta funció el que fem és actualitzar les dades del comptador saturat.

```
if (dir_update_ptr->dir.meta || bpred_dir_lookup(pred->dirpred.twolev, baddr) == dir_update_ptr->pdir1)
    pred->used_2lev++;
else
    pred->used_bimod++;
}
```

Afegim un nou cas a l'if on comparem si les adreces són iguals per comprovar si la twolevel s'utilitza.

```
#endif /* RAS_BUG_COMPATIBLE */

/* update L1 table if appropriate */
/* L1 table is updated unconditionally for combining predictor too */
if ((MD_OP_FLAGS(op) & (F_CTRL | F_UNCOND)) != (F_CTRL | F_UNCOND) &&
    (pred->class == BPred2Level || pred->class == BPredComb || pred->class == BpredCascade))
{
    int l1index, shift_reg;
```

Primer afegim un nou cas al if, ja que el cascade té una gshare amb un BHR i s'ha d'actualitzar.

```
if (dir_update_ptr->pdirl)
{
    if (taken)
    {
        if (*dir_update_ptr->pdirl < 3)
            ++*dir_update_ptr->pdirl;
    }
    else
    { /* not taken */
        if (*dir_update_ptr->pdirl > 0)
            --*dir_update_ptr->pdirl;
    }
}

if (dir_update_ptr->pdirl2 != NULL && *dir_update_ptr->pdirl2 != 4)
{
    if (taken)
    {
        if (*dir_update_ptr->pdirl2 < 3)
            ++*dir_update_ptr->pdirl2;
    }
    else
    { /* not taken */
        if (*dir_update_ptr->pdirl2 > 0)
            --*dir_update_ptr->pdirl2;
    }
}
```

```

else if (!!taken != !!pred_taken && dir_update_ptr->pdire1 != NULL)
{
    if (taken)
    {
        if (*dir_update_ptr->pdire2 < 3)
            ++*dir_update_ptr->pdire2;
    }
    else
    { /* not taken */
        if (*dir_update_ptr->pdire2 > 0)
            --*dir_update_ptr->pdire2;
    }
}

/* meta predictor */
if (dir_update_ptr->pmeta)
{
    if (dir_update_ptr->dir.bimod != dir_update_ptr->dir.twolev)
    {
        /* we only update meta predictor if directions were different */
        if (dir_update_ptr->dir.twolev == (unsigned int)taken)
        {
            /* 2-level predictor was correct */
            if (*dir_update_ptr->pmeta < 3)
                ++*dir_update_ptr->pmeta;
        }
        else
        {
            /* bimodal predictor was correct */
            if (*dir_update_ptr->pmeta > 0)
                --*dir_update_ptr->pmeta;
        }
    }
}

```

El dir1 s'actualitza sempre tant la bimodal com la gshare falli o encerti, el problema està en la dir2.

El primer cas és que el valor del punter de la dir1 sigui diferent de 4. També hem de comprovar que no sigui NULL, ja que si no donarà un segmentation fault. Si és diferent de 4 el valor, voldrà dir que no estem al cascade, per tant, actualitzarem sempre perquè hem d'aconseguir que també funcioni pels altres predictors. Però també pot ser que estiguem en el cascade i dir2 sigui la bimodal a causa que hem fet hit a la gshare amb la funció lookup. Com que la bimodal sempre s'actualitza sigui dir1 o dir2 l'haurem d'actualitzar.

Hem afegit un elseif, ja que hi ha un altre cas que la gshare del cascade sigui la dir2, però la predicció de la bimodal a la dir1 falli. En aquest cas sí que hem d'actualitzar la gshare i també hem de comprovar que sigui NULL per evitar tenir segmentation fault.

2.3 Comparació de resultats respecte la fase 1

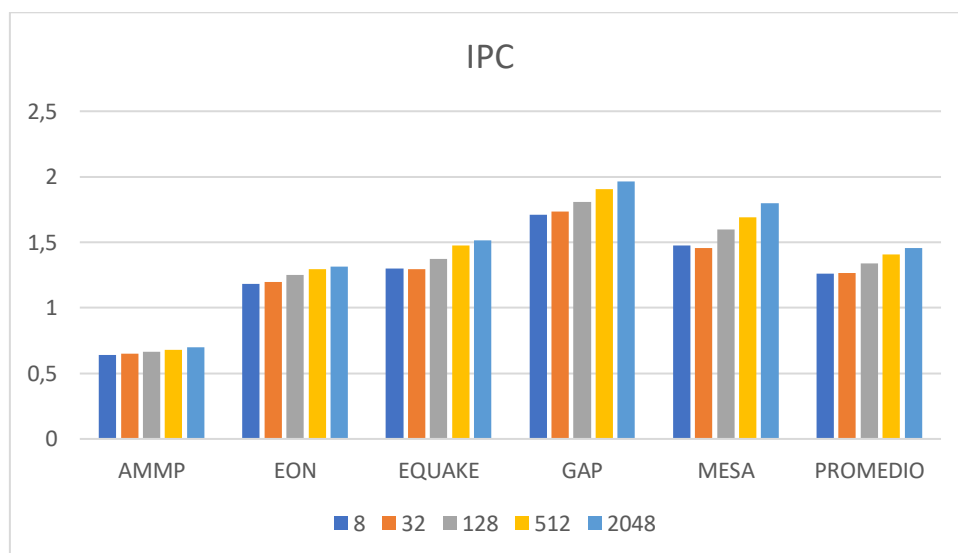
CASCADE

IPC

Durant la realització de les proves d'aquest nou predictor de salt "cascade" ens hem pogut adonar del potencial de combinar dos predictors que ja per si mateixos ens donen molt bon resultat.

En primer lloc analitzarem les dades obtingudes del predictor "COMB", les quals han sigut altament positives, deixant a aquest predictor en segona posició:

X	IPC				
	8	32	128	512	2048
AMMP	0,6423	0,6507	0,6658	0,6807	0,6975
EON	1,1838	1,1972	1,2488	1,2943	1,3133
EQUAKE	1,3021	1,2953	1,3756	1,4736	1,5172
GAP	1,708	1,7326	1,81	1,9043	1,9665
MESA	1,4756	1,455	1,5973	1,6914	1,7989
PROMEDIO	1,26236	1,26616	1,3395	1,40886	1,45868



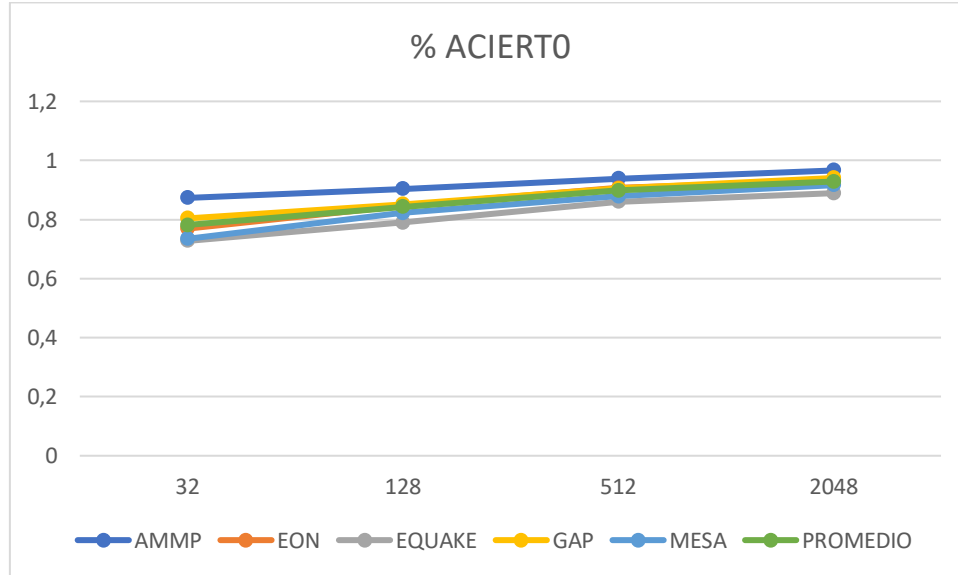
Un cop duts a terme els tests corresponents, veiem una millora en quant al IPC del test AMMP, on als previs predictors no arribava a superar el 0,10 d'IPC.

En primera posició trobem GAP, com era comú a bastants predictors de la primera fase, arribant al seu màxim quan la configuració del predictor és de 2048.

Si fem la mitja (1,347), veiem que obtenim un IPC lleugerament superior al que hem pogut trobar en els predictors de la primera fase, on l'IPC més gran que podíem trobar era en el cas del PERFECT, amb un IPC d'1,42.

% D'ENCERT

X	8	32	128	512	2048
AMMP	0,8592	0,8731	0,9037	0,9386	0,966
EON	0,763	0,7691	0,847	0,9073	0,9308
EQUAKE	0,7258	0,7275	0,7902	0,8598	0,8892
GAP	0,7899	0,8037	0,8514	0,9044	0,9406
MESA	0,7214	0,7346	0,8225	0,879	0,9162
PROMEDIO	0,77186	0,7816	0,84296	0,89782	0,92856



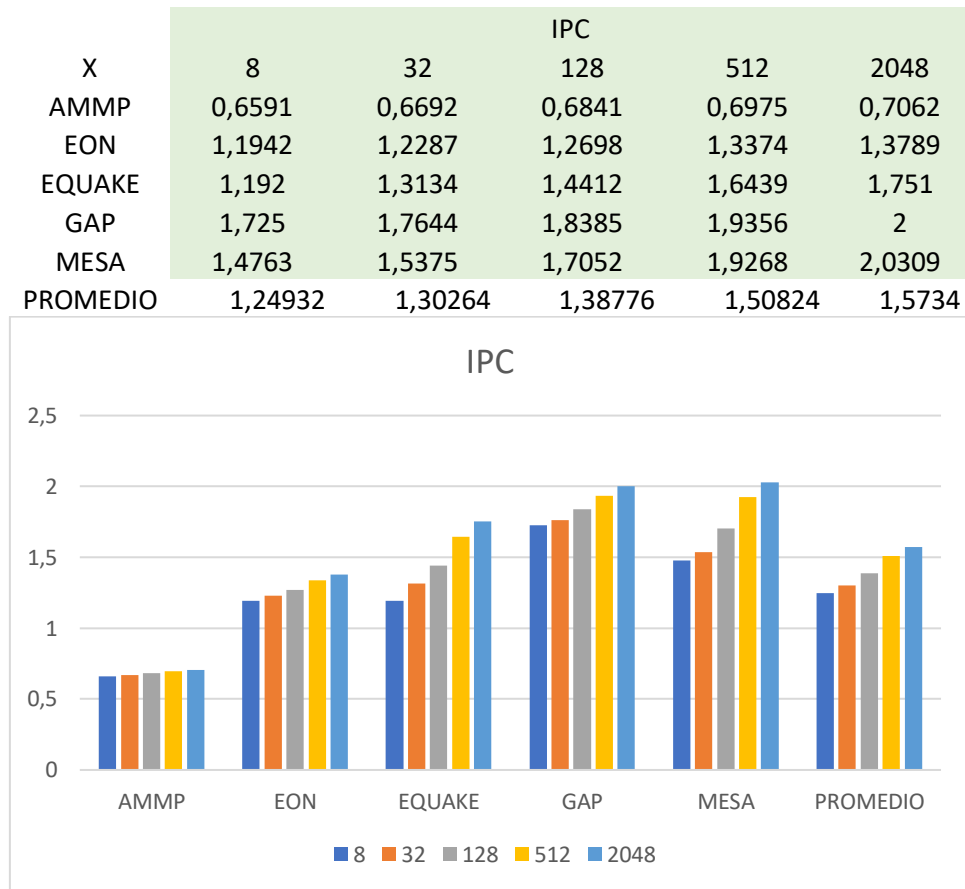
Si observem el % d'encert, podem veure que és un valor proper a la mitja d'encert de la fase anterior, situant al predictor "CASCADE" en la 5^a posició, directament per sota del "BIMODAL".

En general, considerem que no hem obtingut un mal % d'encert, arribant al 96,6% d'encert en la configuració més gran de l'AMMP.

Per a aconseguir aquest alt percentatge, vam fer varies proves com està explicat prèviament, i vam intentar cobrir el màxim de casos per a poder fer el switch de predictor la millor manera possible.

COMB

IPC

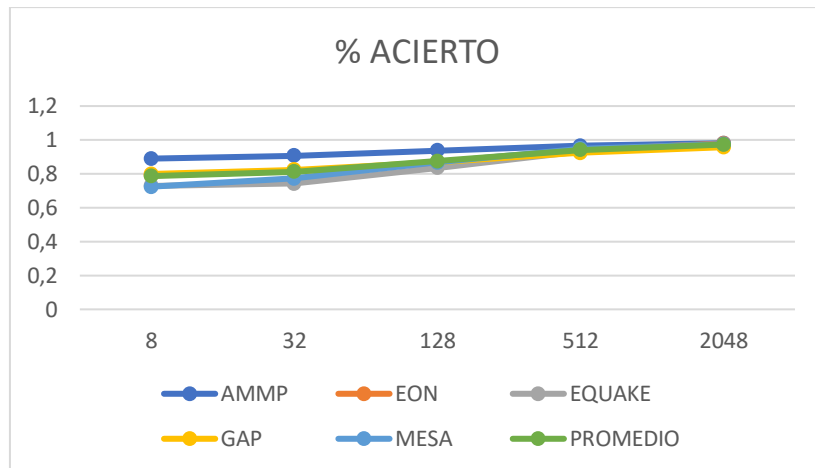


Un cop analitzades les dades obtingudes al executar els tests amb el predictor “COMB”, podem veure un IPC molt similar al del “CASCADE”, lleugerament superior, no obstant.

En general, és un IPC molt correcte, sobretot en el cas del GAP, arribant a l'1,73 en la seva millors configuració. Això situa a aquest predictor en segona posició, directament per sota del “PERFECT”. En quant al IPC, podem dir que s'aconsegueix un bon equilibri entre IPC - %d'encert, ja que no veiem una descompensació exagerada.

% D'ENCERT

X	% ACIERTO				
	8	32	128	512	2048
AMMP	0,8899	0,9071	0,9365	0,9652	0,9809
EON	0,7908	0,8154	0,8611	0,9397	0,9796
EQUAKE	0,7284	0,7422	0,8354	0,9305	0,972
GAP	0,7993	0,8242	0,8699	0,9231	0,9571
MESA	0,7242	0,7724	0,8679	0,9488	0,9755
PROMEDIO	0,78652	0,81226	0,87416	0,94146	0,97302



Al analitzar les dades d'aquest predictor, podem veure un % d'encert molt decent, sobretot en el cas de la màxima configuració d'AMMP, donant-nos un 98,1% d'encert.

Aquesta alta taxa d'encert, situa a aquest predictor en la segona posició, just per sota del "PERFECT".

Podem dir que és un dels millors que hem pogut analitzar en aquesta pràctica i així les seves dades ho demostren. En general, com he esmentat abans, s'ha aconseguit un equilibri molt correcte en quant al IPC - % d'encert i veiem que sempre es manté més o menys un % d'encert elevat i no trobem moltes pujades i baixades.

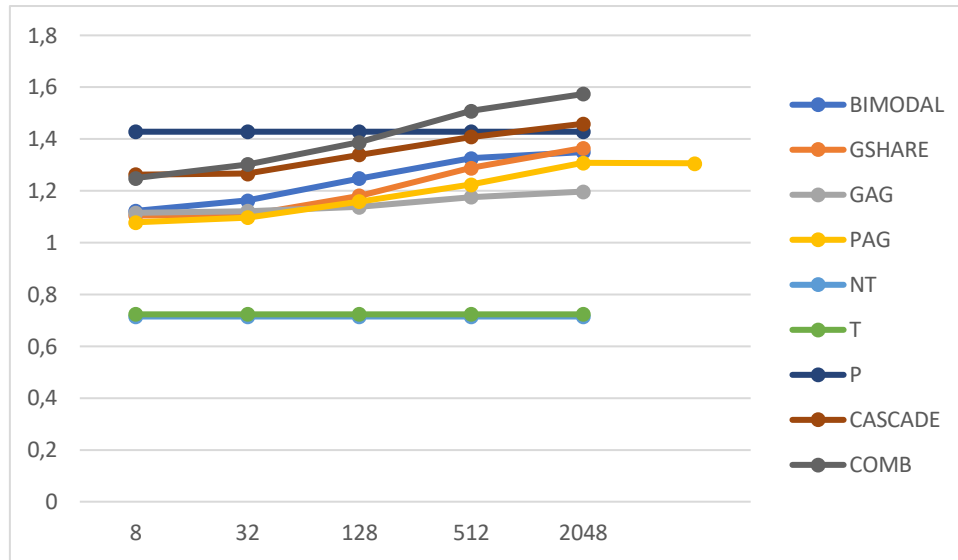
Hem escollit $arrayZ=(4\ 16\ 64\ 256\ 1024)$ per a la mida que són justament la meitat de la mida de l_2 de la twolevel. Hem observat que ha mesurat que incrementàvem els valors el nombre d'encerts creixia exponencialment. Ja que en haver-hi molts salts a les proves benchmark contra més gran siguin les caches més salts podrem guardar a les caches.

CONCLUSIÓ FINAL

Per a acabar aquesta pràctica, volem fer un breu “recap” dels resultats que hem obtingut.

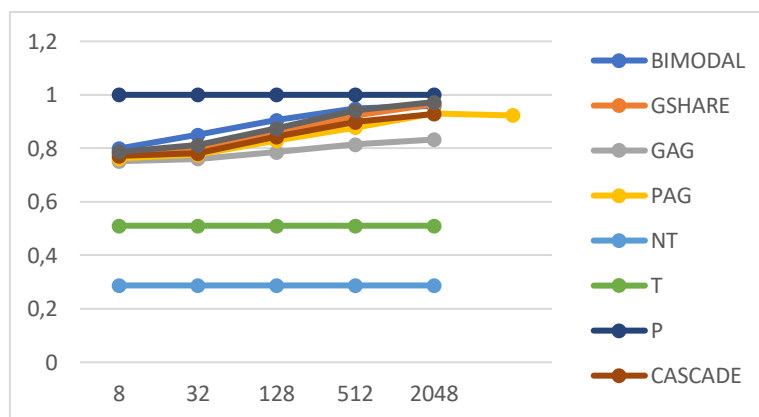
Si haguéssim de fer un rànkig de IPC, podríem obtenir la següent classificació:

Com veiem, el predictor que millor IPC ens ha aportat ha sigut el PERFECT, com era d'esperar. No obstant, la majoria de predictors donen un IPC correcte, tret del cas del Not taken.



En quant al % d'encert, obtenim la següent classificació, on el “CASCADE” veiem que ocupa la 5ª posició tot i haver aconseguit un molt bon percentatge d'encert, tractant-se del 96,6% en el millor dels casos. Per tant, els seus valors d'encert oscil·len per sobre de 80 de mitja fins arribar a obtenir 97%.

1. PERFECT
2. COMB
3. GSHARE
4. BIMODAL
5. CASCADE
6. PAG
7. TAKEN
8. NOT TAKEN



El comb i el cascade són predictors híbrids amb pràcticament les mateixes estructures i implementació. El comb utilitza un hardware addicional (meta table) per escollir quin predictor triem i el cascade, en canvi, inicialitza diferent les taules i primer mira si el salt està guardat en la gshare. En cas contrari agafem la bimodal com a direcció principal i només actualitzem la gshare si falla la predicció de la bimodal. A conseqüència d'aquestes similituds hem obtingut uns resultats semblants en totes les proves.