

Universitat Rovira i Virgili
Escola Tècnica Superior d'Enginyeria

ARQUITECTURA DE COMPUTADORES

PRÁCTICA 2: Simulación de un procesador superescalar: Predictores de salto

AUTORES:

YANI AICI LOUNIS

IVAN MORILLAS GÓMEZ

DOCENTE:

CARLES ALIAGAS CASTELL

29/11/2023

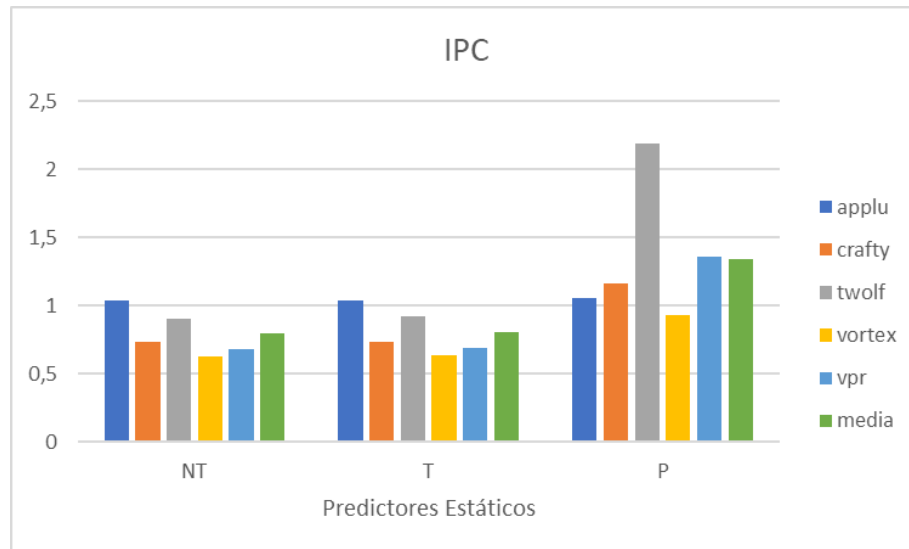
2023-2024

ÍNDICE

Fase 1	3
Predictores estáticos	3
Predictores dinámicos	5
Resumen gráficas	13
Fase 2	19
Predictor Gskew	19
Comparación predictores	20

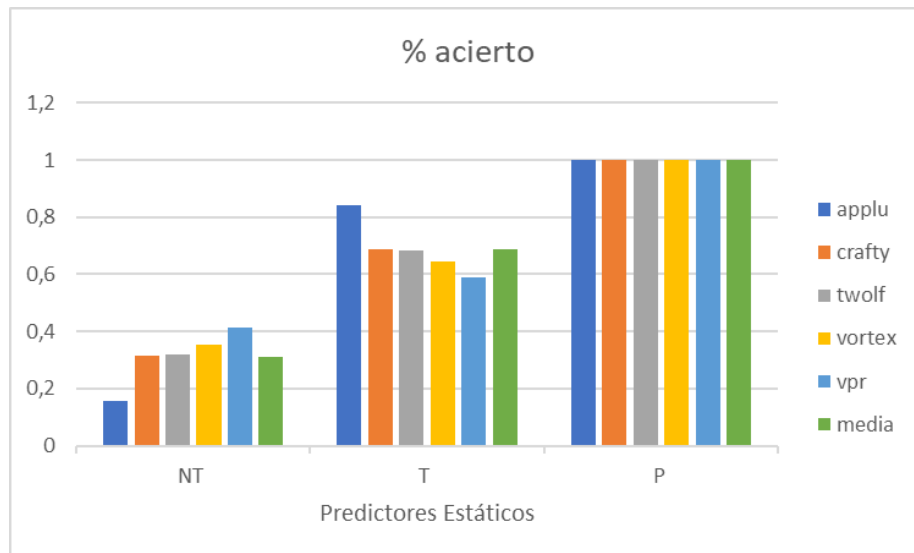
FASE 1

Predictores estáticos:



IPC	applu	crafty	twolf	vortex	vpr	media
NT	1,0368	0,7311	0,9072	0,6261	0,6794	0,7961
T	1,0370	0,7388	0,9226	0,6331	0,6872	0,8037
P	1,0550	1,1616	2,1878	0,9326	1,3599	1,3393

En este estudio del **IPC**, donde tenemos los diferentes predictores de saltos estáticos, podemos observar que, efectivamente, el predictor **Perfect** obtiene un mayor resultado del número de instrucciones por ciclo. Esto se debe por el hecho de que, como deja pasar todas las instrucciones sin que haya un fallo de predicción, el **front end** no tiene que cargar instrucciones que no estaban cargadas por la mala predicción. En cambio, con los otros tipos de predictores, cuando se falla en una predicción, el **front end** tiene que volver a cargar instrucciones al **Instruction Window**, de tal manera que salen menos instrucciones por ciclo.



% acierto	applu	crafty	twolf	vortex	vpr	media
NT	0,1580	0,3137	0,3176	0,3533	0,4122	0,3109
T	0,8420	0,6878	0,6842	0,6467	0,5878	0,6897
P	1	1	1	1	1	1

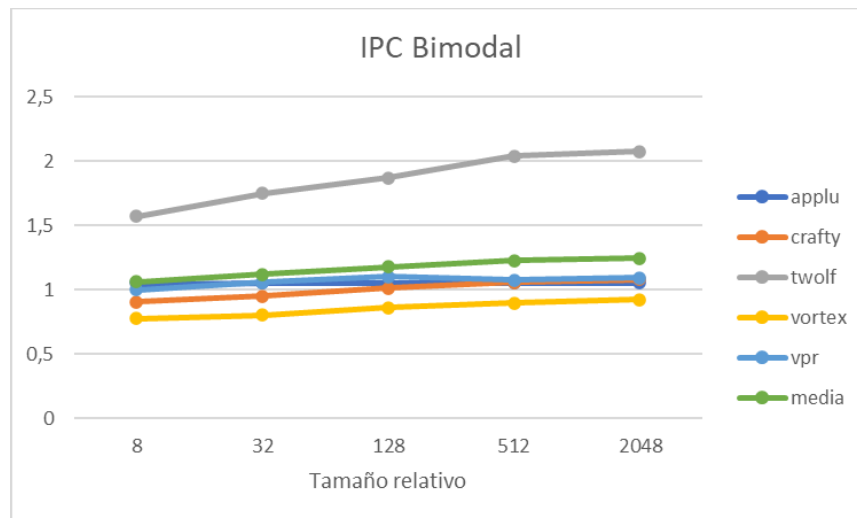
Con el porcentaje de aciertos se pueden comparar los valores de los predictores de salto respecto al **Perfect**, pues este predictor es teórico y simula un acierto con el 100% de probabilidad.

En el caso de los **benchmarks** estudiados, en todos, el porcentaje de acierto del predictor **Taken** es más grande que el **Not Taken**. Esto puede deberse a que las pruebas contienen bucles que se repiten un número de veces considerable.

Predictores dinámicos:

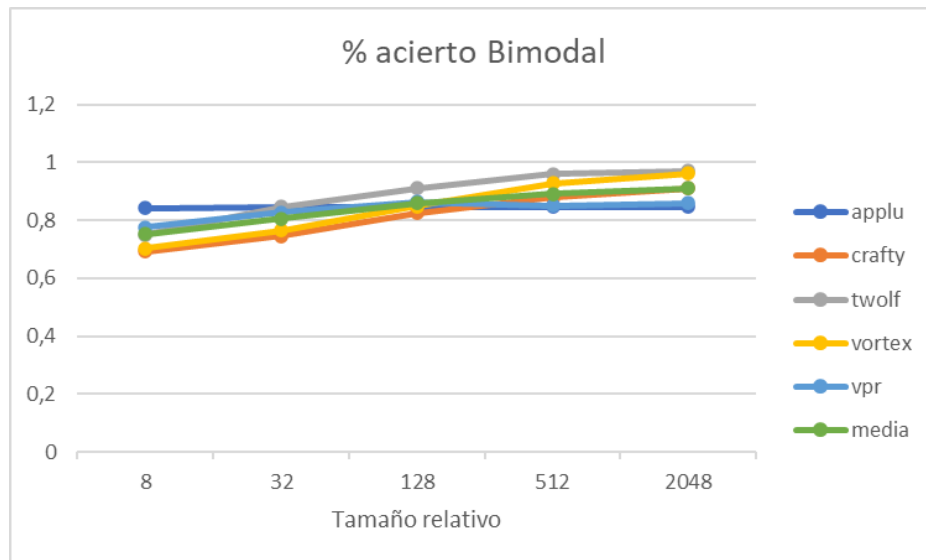
Por tal de mejorar el porcentaje de aciertos de los predictores estáticos, se utilizan los dinámicos, que se caracterizan por hacer cambios de predicción en tiempo de ejecución basándose en los saltos hechos anteriormente.

Bimodal



IPC	applu	crafty	twolf	vortex	vpr	media
8	1,0529	0,9052	1,5717	0,7739	0,9986	1,0604
32	1,0530	0,9487	1,7469	0,8005	1,0563	1,1210
128	1,0530	1,0124	1,8704	0,8605	1,1008	1,1794
512	1,0530	1,0562	2,0403	0,8979	1,0751	1,2245
2048	1,0530	1,0800	2,0747	0,9243	1,0912	1,2446

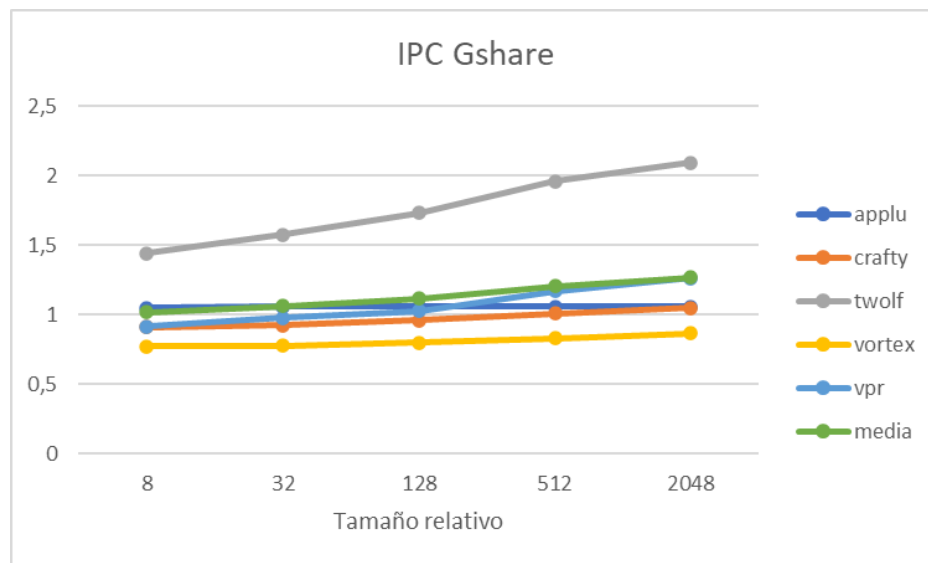
Los resultados para el **IPC** no muestran un cambio muy pronunciado exceptuando el **Twolf**, que aumenta su **IPC** a medida que aumenta el valor de la **PHT**, en cambio, los demás **benchmarks** adquieren un mínimo aumento al aumentar el valor de la **PHT**.



% acierto	applu	crafty	twolf	vortex	vpr	media
8	0,8429	0,6930	0,7508	0,7032	0,7760	0,7531
32	0,8446	0,7450	0,8466	0,7659	0,8268	0,8057
128	0,8446	0,8262	0,9119	0,8496	0,8635	0,8591
512	0,8463	0,8784	0,9611	0,9274	0,8502	0,8926
2048	0,8463	0,9099	0,9704	0,9627	0,8600	0,9098

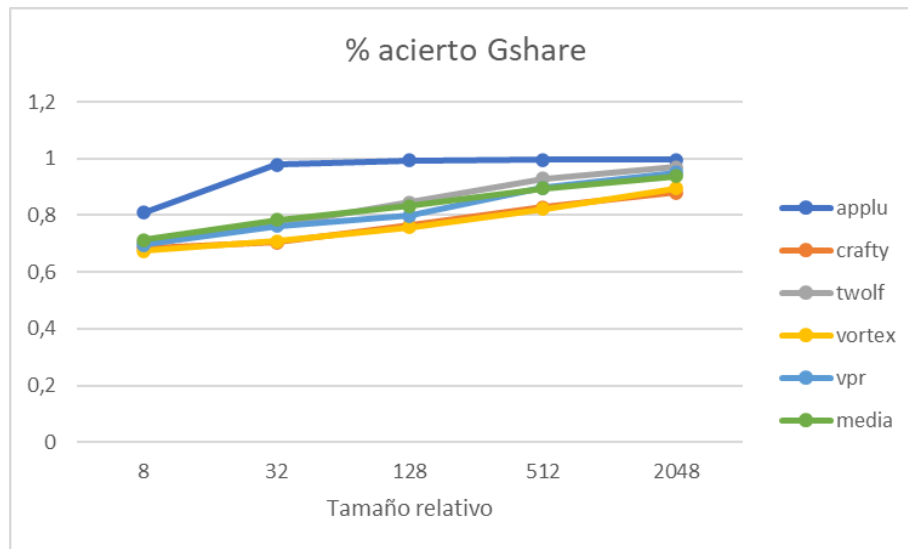
El estudio que hemos hecho del predictor de 1-lvl **Bimodal**, ha obtenido una buena media de predicción, llegando hasta un 97% de aciertos con el **Twolf**. Vemos que, a medida que aumenta el número de entradas de la tabla **PHT**, el porcentaje de aciertos del **Applu** se mantiene mínimamente igual, independientemente de la dimensión de la tabla. El comportamiento de este **benchmark** puede ser debido a que no implementa un gran nombre de saltos en diferentes PCs. De otra banda, tenemos a los otros **benchmarks** que según aumenta el tamaño de la tabla, aumenta el porcentaje de aciertos.

Gshare



IPC	applu	crafty	twolf	vortex	vpr	media
8	1,0490	0,9091	1,4431	0,7720	0,9135	1,0173
32	1,0560	0,9220	1,5735	0,7785	0,9770	1,0614
128	1,0564	0,9606	1,7305	0,8014	1,0250	1,1147
512	1,0564	1,0081	1,9599	0,8301	1,1696	1,2048
2048	1,0564	1,0492	2,0950	0,8664	1,2626	1,2659

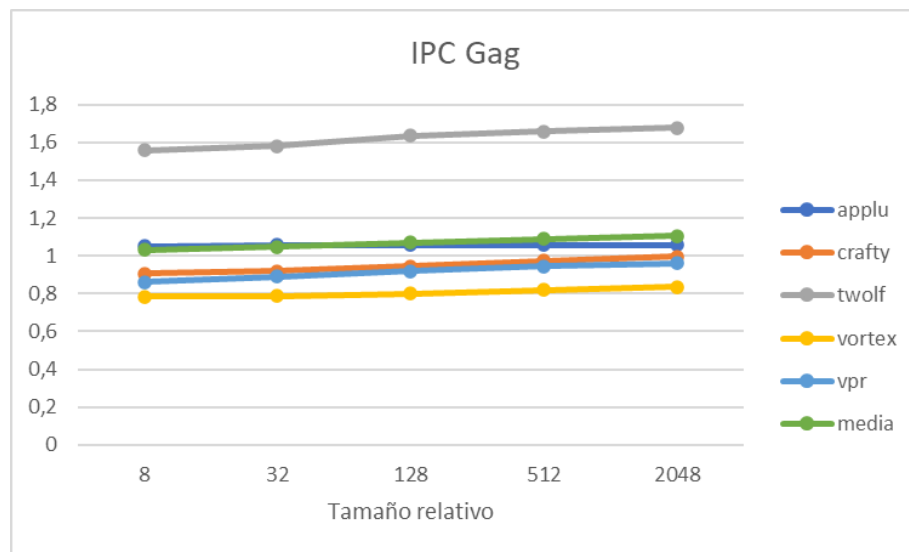
Como podemos observar en la gráfica anterior, el comportamiento del **benchmark Applu** se mantiene constante y no varía según se aumenta el tamaño de la tabla de predictores. En cambio, los otros **benchmarks** se observa un ligero aumento según se aumenta dicho tamaño. El **Twolf** es el único **benchmark** que obtiene un gran aumento al aumentar el tamaño.



% acierto	applu	crafty	twolf	vortex	vpr	media
8	0,8090	0,6854	0,7019	0,6736	0,6948	0,7129
32	0,9781	0,7051	0,7672	0,7076	0,7626	0,7841
128	0,9940	0,7641	0,8448	0,7567	0,7987	0,8316
512	0,9957	0,8281	0,9296	0,8216	0,8960	0,8942
2048	0,9957	0,8801	0,9701	0,8959	0,9502	0,9384

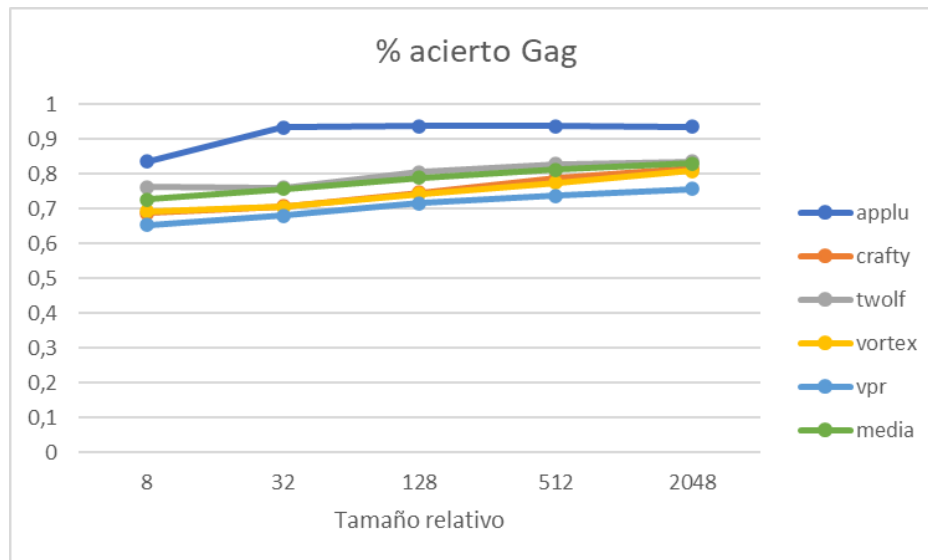
En el caso del **Gshare**, se obtiene una gráfica donde se observa como el aumento del tamaño de la tabla **PHT** provoca que en todos los **benchmarks** suba el porcentaje de aciertos. Esto es debido a que, como hemos visto en el **Bimodal**, hace que haya menos **aliasing** y, por tanto, haya menos fallos de predicción. Además, el **Gshare**, tiene un mayor crecimiento según se aumenta el tamaño de la tabla, ya que, para acceder a ella, se hace una operación **XOR** del **BTB** y el **BHR** haciendo que el acceso a esta tabla disminuya este **aliasing**.

Gag



IPC	applu	crafty	twolf	vortex	vpr	media
8	1,0525	0,9048	1,5564	0,7848	0,8623	1,0321
32	1,0557	0,9197	1,5788	0,7865	0,8902	1,0461
128	1,0557	0,9438	1,6338	0,7993	0,9190	1,0703
512	1,0557	0,9746	1,6567	0,8194	0,9445	1,0901
2048	1,0555	0,9966	1,6763	0,8364	0,9609	1,1051

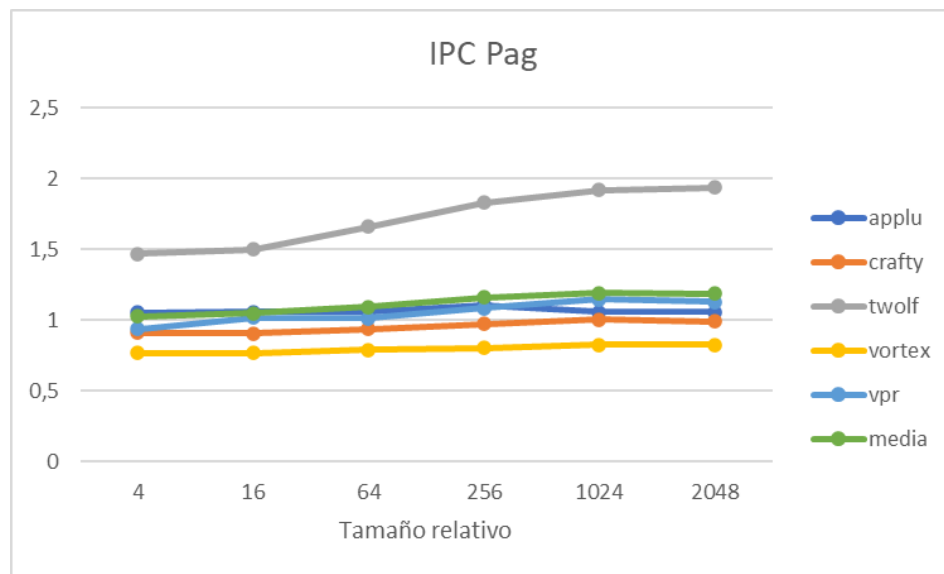
Como podemos observar, el rendimiento de las instrucciones por ciclo, en general, a todos los **benchmarks** no se les ve afectado por el crecimiento de la tabla **PHT**. Obviamente al hacer crecer esta tabla, se produce una disminución del **aliasing**, haciendo que los **benchmarks**, con muchos saltos, tengan un poco mejor rendimiento al aumentar la tabla. En cambio, como hemos mencionado antes, no se observa una mejora del IPC.



% acierto	applu	crafty	twolf	vortex	vpr	media
8	0,8352	0,6866	0,7617	0,6924	0,6524	0,7256
32	0,9326	0,7073	0,7598	0,7038	0,6797	0,7566
128	0,9365	0,7456	0,8041	0,7418	0,7152	0,7886
512	0,9364	0,7876	0,8270	0,7738	0,7364	0,8122
2048	0,9352	0,8161	0,8356	0,8087	0,7556	0,8302

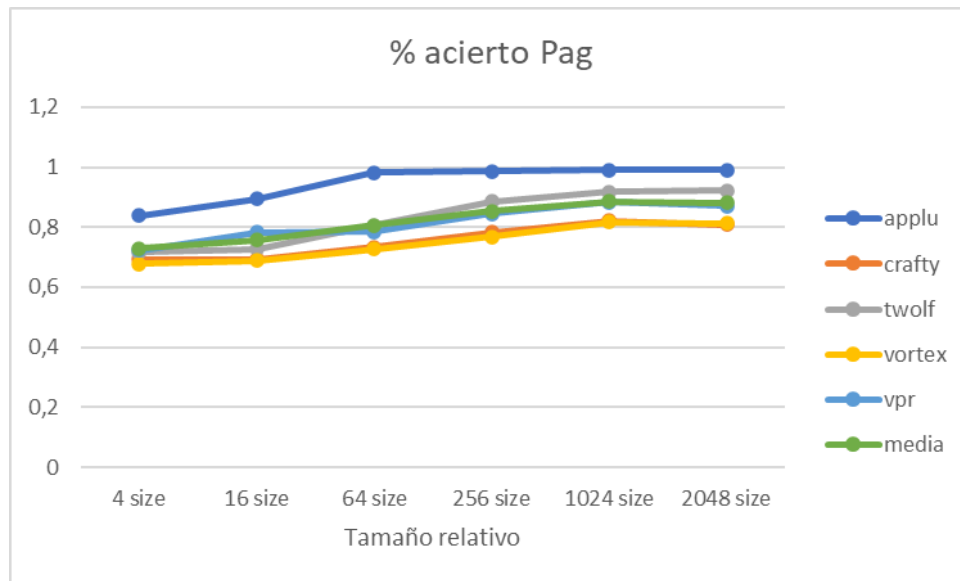
En el predictor **Gag**, podemos observar que el aumento del tamaño del buffer realmente afecta poco en el porcentaje de aciertos exceptuando algún caso puntual. La principal diferencia entre el **Gag** y el **Gshare** es la forma en que se accede a la **PHT**, donde el **Gshare** hace una operación **XOR** entre el **BTB** y el **BHR** y el **Gag** no. Por lo tanto, podemos concluir que esta operación **XOR** que se hace, realmente da un porcentaje de acierto más elevado que si no se hace.

Pag



IPC	applu	crafty	twolf	vortex	vpr	media
4	1,0527	0,9108	1,4700	0,7676	0,9387	1,0279
16	1,0556	0,9074	1,4999	0,7684	1,0181	1,0498
64	1,0560	0,9383	1,6603	0,7901	1,0128	1,0915
256	1,1056	0,9739	1,8273	0,8033	1,0852	1,1590
1024	1,0562	1,0034	1,9172	0,8273	1,1462	1,1900
2048	1,0562	0,9924	1,9359	0,8249	1,1268	1,1872

Observando la gráfica anterior, podemos ver que el comportamiento del **benchmark Applu** se mantiene constante y no varía según se aumenta el tamaño de la tabla de predictores. En cambio, en los otros **benchmarks**, se observa un ligero aumento de rendimiento según se aumenta dicho tamaño. El único **benchmark** se presenta un cambio significativo del **IPC** es el **Twolf**, el cual puede presentar una cantidad de saltos bastante altos y por lo tanto al aumentar el tamaño de la tabla, hacemos que baje el **aliasing**.

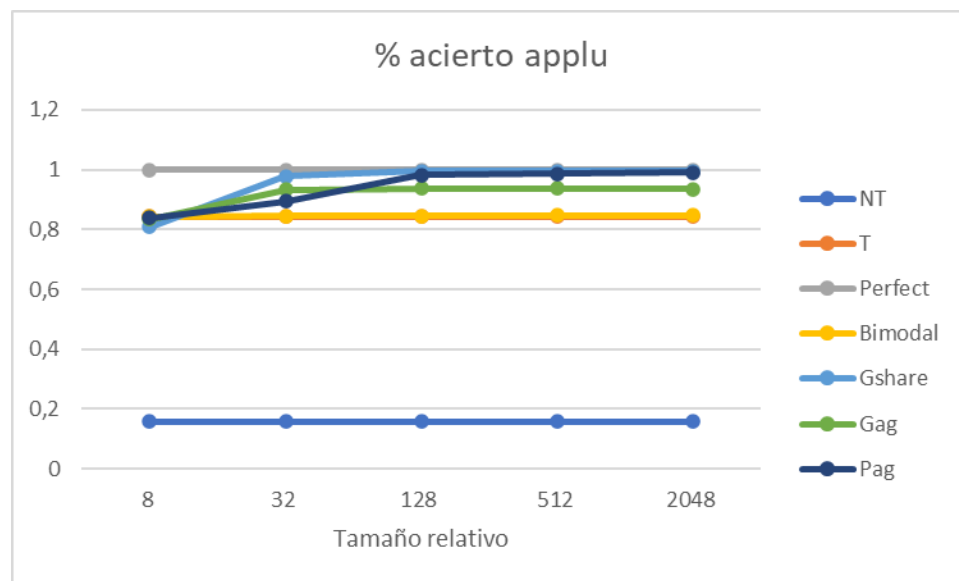


% acierto	applu	crafty	twolf	vortex	vpr	media
4	0,8377	0,6937	0,7161	0,6774	0,7222	0,7294
16	0,8945	0,6928	0,7275	0,6883	0,7832	0,7572
64	0,9811	0,7336	0,8070	0,7271	0,7854	0,8068
256	0,9862	0,7824	0,8861	0,7683	0,8449	0,8535
1024	0,9897	0,8223	0,9167	0,8174	0,8840	0,8860
2048	0,9897	0,8075	0,9219	0,8123	0,8709	0,8804

Como podemos observar con la gráfica anterior, el predictor **Pag** tiene uno de los porcentajes de acierto en general más altos sin llegar a superar al **Gshare**. Esto es debido a la variabilidad que tienen en cuanto a tablas, haciendo posible cambiar tanto el tamaño de la tabla **PHT** como la del **BHR**. Por lo tanto, vemos que sin hacer una operación **XOR** para determinad la posición de la tabla, podemos obtener un buen rendimiento si aumentamos el tamaño de estas.

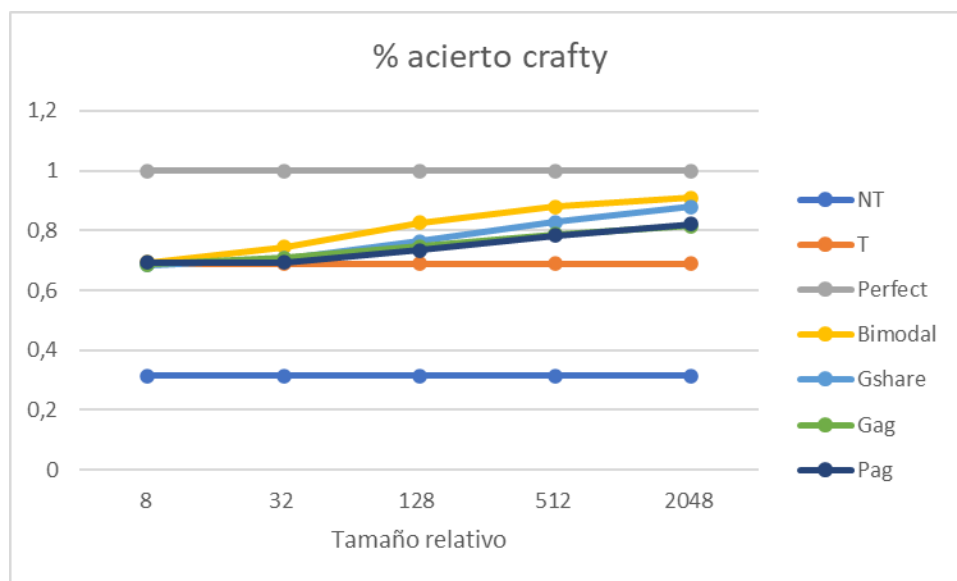
Resumen gráficas:

En las siguientes gráficas mostraremos el porcentaje de aciertos de cada predictor en los diferentes benchmarks con los que hemos trabajado.



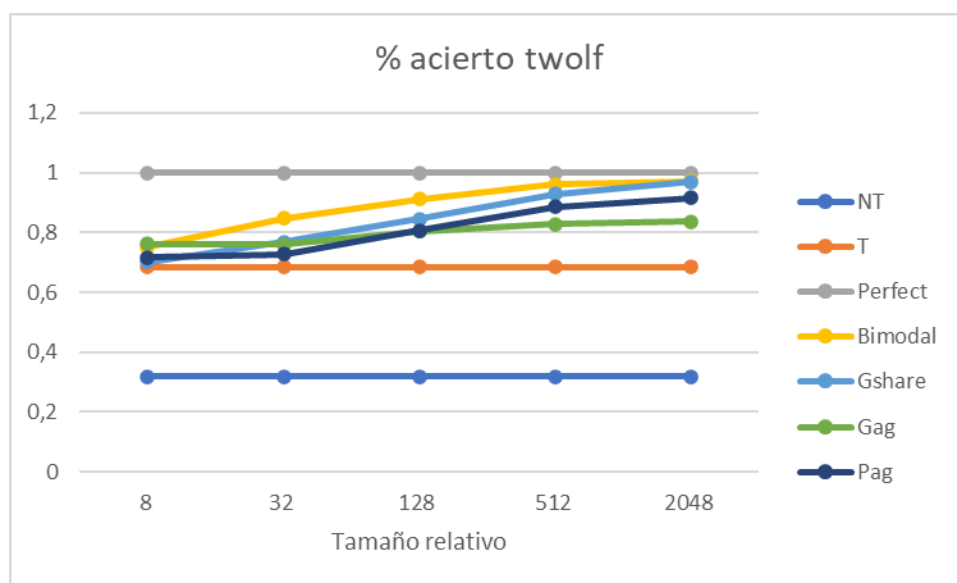
%acierto	8	32	128	512	2048
NT	0,1580	0,1580	0,1580	0,1580	0,1580
T	0,8420	0,8420	0,8420	0,8420	0,8420
Perfect	1	1	1	1	1
Bimodal	0,8429	0,8446	0,8446	0,8463	0,8463
Gshare	0,8090	0,9781	0,9940	0,9957	0,9957
Gag	0,8352	0,9326	0,9365	0,9364	0,9352
Pag	0,8377	0,8945	0,9811	0,9862	0,9897

En el **benchmark Applu** podemos ver que en general, a medida que se aumenta el tamaño de la tabla **PHT**, obtiene un aumento en todos los predictores desde el tamaño de 8 hasta el de 128, exceptuando los predictores estáticos y el predictor 1-lvl **Bimodal**.



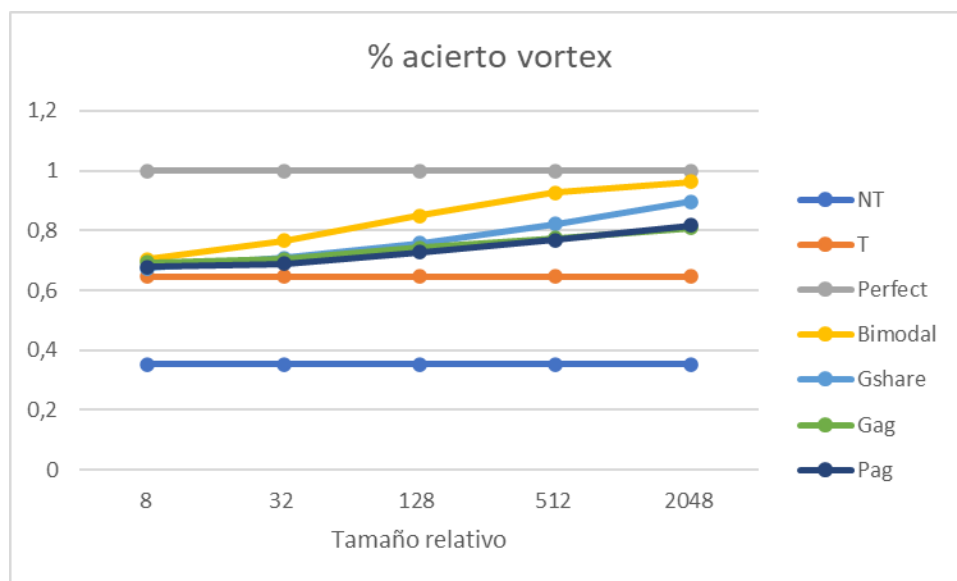
%acierto	8	32	128	512	2048
NT	0,3137	0,3137	0,3137	0,3137	0,3137
T	0,6878	0,6878	0,6878	0,6878	0,6878
Perfect	1	1	1	1	1
Bimodal	0,6930	0,7450	0,8262	0,8784	0,9099
Gshare	0,6854	0,7051	0,7641	0,8281	0,8801
Gag	0,6866	0,7073	0,7456	0,7876	0,8161
Pag	0,6937	0,6928	0,7336	0,7824	0,8223

En el **benchmark Crafty** podemos observar un aumento del porcentaje de acierto en todos los predictores, exceptuando los predictores estáticos, en el que reciben un gran aumento a partir del tamaño de 32.



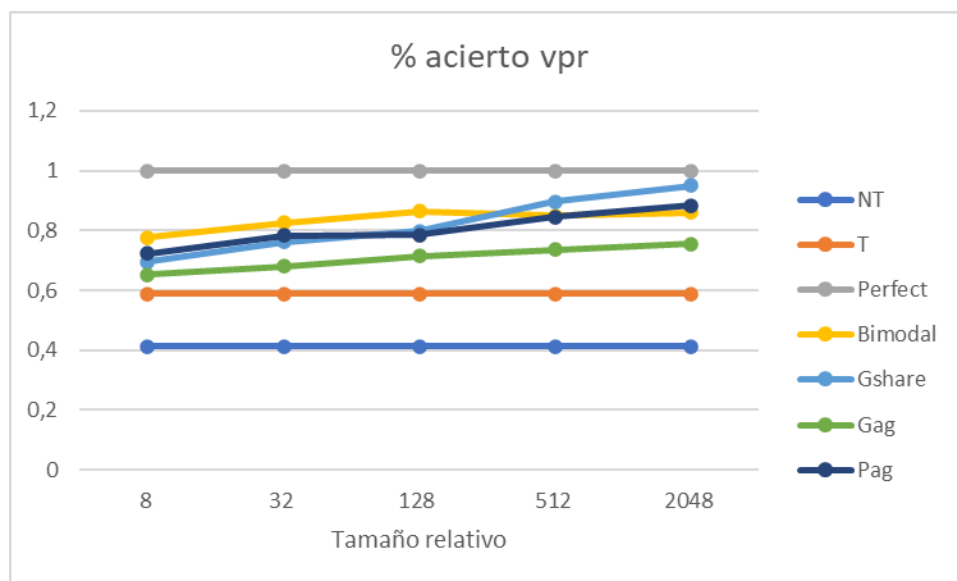
%acierto	8	32	128	512	2048
NT	0,3176	0,3176	0,3176	0,3176	0,3176
T	0,6842	0,6842	0,6842	0,6842	0,6842
Perfect	1	1	1	1	1
Bimodal	0,7508	0,8466	0,9119	0,9611	0,9704
Gshare	0,7019	0,7672	0,8448	0,9296	0,9701
Gag	0,7617	0,7598	0,8041	0,8270	0,8356
Pag	0,7161	0,7275	0,8070	0,8861	0,9167

En el **benchmark Twolf** podemos ver que el porcentaje promedio de los diferentes predictores es más alto que con los otros **benchmarks**.



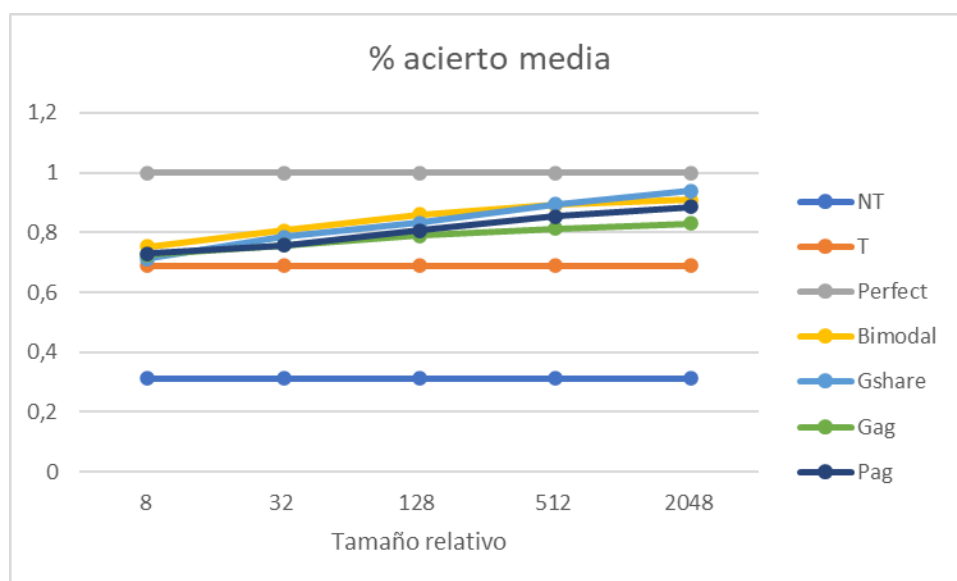
%acierto	8	32	128	512	2048
NT	0,3533	0,3533	0,3533	0,3533	0,3533
T	0,6467	0,6467	0,6467	0,6467	0,6467
Perfect	1	1	1	1	1
Bimodal	0,7032	0,7659	0,8496	0,9274	0,9627
Gshare	0,6736	0,7076	0,7567	0,8216	0,8959
Gag	0,6924	0,7038	0,7418	0,7738	0,8087
Pag	0,6774	0,6883	0,7271	0,7683	0,8174

En el **benchmark Vortex** podemos ver que el porcentaje de acierto aumenta más a medida que el tamaño de la tabla **PHT** es más grande, exceptuando los predictores estáticos.



%acierto	8	32	128	512	2048
NT	0,4122	0,4122	0,4122	0,4122	0,4122
T	0,5878	0,5878	0,5878	0,5878	0,5878
Perfect	1	1	1	1	1
Bimodal	0,7760	0,8268	0,8635	0,8502	0,8600
Gshare	0,6948	0,7626	0,7987	0,8960	0,9502
Gag	0,6524	0,6797	0,7152	0,7364	0,7556
Pag	0,7222	0,7832	0,7854	0,8449	0,8840

En el **benchmark Vpr** podemos observar que el porcentaje de acierto depende del tipo de predictor que se implementa y, por tanto, podemos decir que este **benchmark** contiene una gran cantidad de salto que no son bucles o, al menos, los bucles no son de grandes iteraciones.

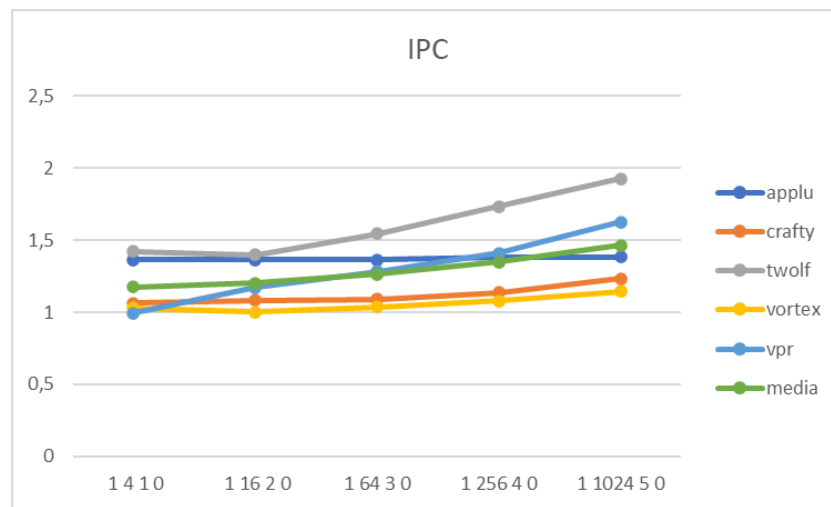


%acierto	8	32	128	512	2048
NT	0,3109	0,3109	0,3109	0,3109	0,3109
T	0,6897	0,6897	0,6897	0,6897	0,6897
Perfect	1	1	1	1	1
Bimodal	0,7531	0,8057	0,8591	0,8926	0,9098
Gshare	0,7129	0,7841	0,8316	0,8942	0,9384
Gag	0,7256	0,7566	0,7886	0,8122	0,8302
Pag	0,7294	0,7572	0,8068	0,8535	0,8860

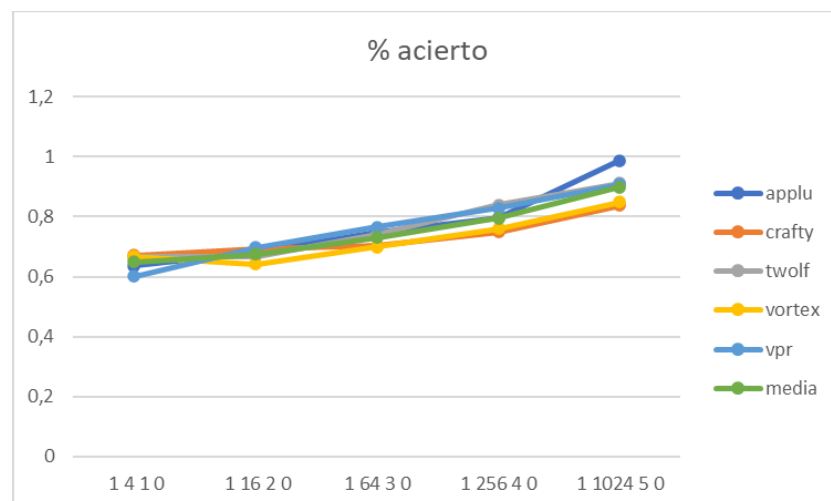
En general, como podemos observar en esta gráfica, en los predictores estáticos, el predictor **Perfect** teóricamente siempre tiene un 100% de probabilidad de aciertos, el **Not Taken** tiene menos porcentaje que el **Taken**. En los predictores dinámicos, todos aumentan su porcentaje de aciertos a medida que se aumenta el tamaño de la tabla **PHT**, pero acercándose lo máximo al 100% sin llegar.

FASE 2

Predictor Gskew:



IPC	applu	crafty	twolf	vortex	vpr	media
1 4 1 0	1,3630	1,0653	1,4225	1,0290	0,9933	1,1746
1 16 2 0	1,3634	1,0830	1,3991	1,0022	1,1702	1,2035
1 64 3 0	1,3641	1,0888	1,5453	1,0374	1,2832	1,2637
1 256 4 0	1,3818	1,1361	1,7343	1,0775	1,4097	1,3478
1 1024 5 0	1,3822	1,2322	1,9267	1,1461	1,6270	1,4628



% acierto	applu	crafty	twolf	vortex	vpr	media
1 4 1 0	0,6365	0,6715	0,6661	0,6674	0,601	0,6485
1 16 2 0	0,6798	0,6931	0,6683	0,6420	0,6967	0,6759
1 64 3 0	0,7471	0,7024	0,7413	0,6988	0,7656	0,7310
1 256 4 0	0,7965	0,7494	0,8394	0,7609	0,8290	0,7950
1 1024 5 0	0,9875	0,8377	0,9126	0,8489	0,9081	0,8989

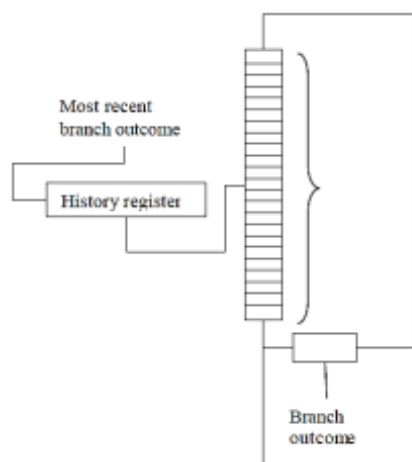
¿A que otro predictor Gag, Pag, Gap, Pap se parece? ¿Con qué parámetros? Simuladlo y comparadlo con el Gskew implementado.

La predicción realizada por **Gskew** es una predicción de dos niveles (**2-lvl**) ya que estamos usando dos niveles: **Branch History Register (BHR)** que almacena los últimos 'g' saltos que se han ejecutado en el procesador (0 **NotTaken**, 1 **Taken**) y el **Pattern History Table (PHT)** que almacena dos bits para implementar un contador saturado del comportamiento de los saltos (11 -> **Strong Taken**; 10 -> **Weak Taken**; 01 -> **Weak NotTaken**; 00 -> **Strong NotTaken**).

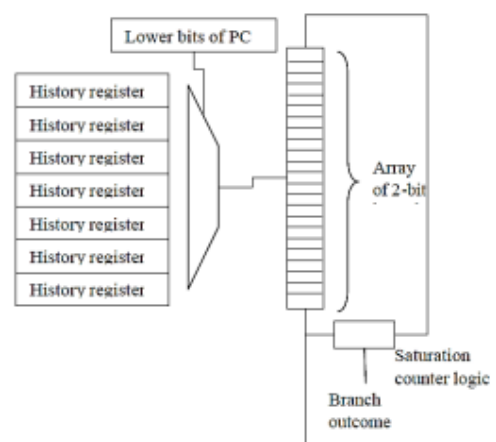
Análisis de los otros predictores:

Figure 6. 2-level adaptive predictor structure

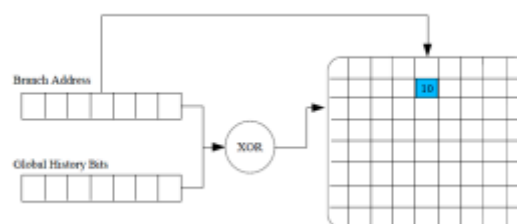
predictor	l1_size	hist_size	l2_size	xor
GAg	1	W	2^W	0
GAp	1	W	$>2^W$	0
PAg	N	W	2^W	0
PAP	N	W	2^{N+W}	0
gshare	1	W	2^W	1



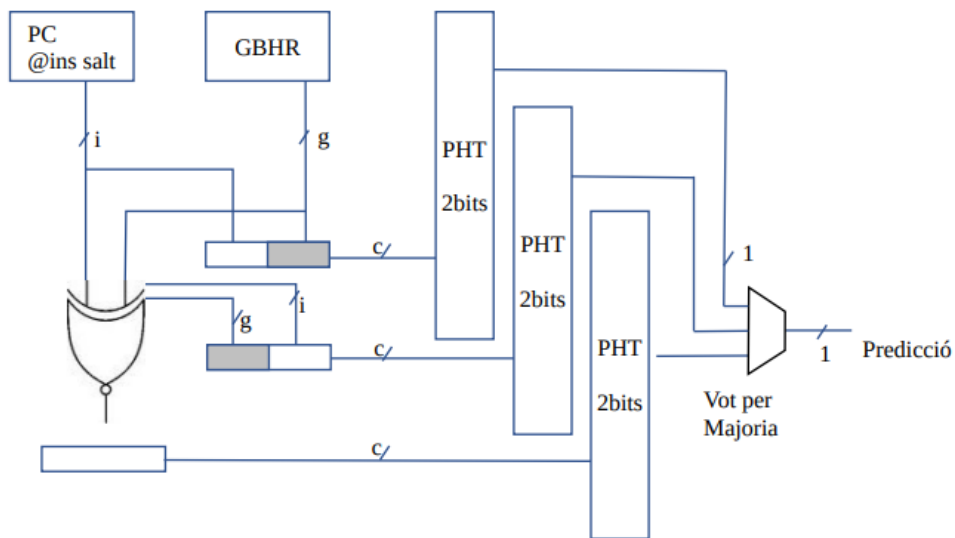
Predictor Gap



Predictor Pap

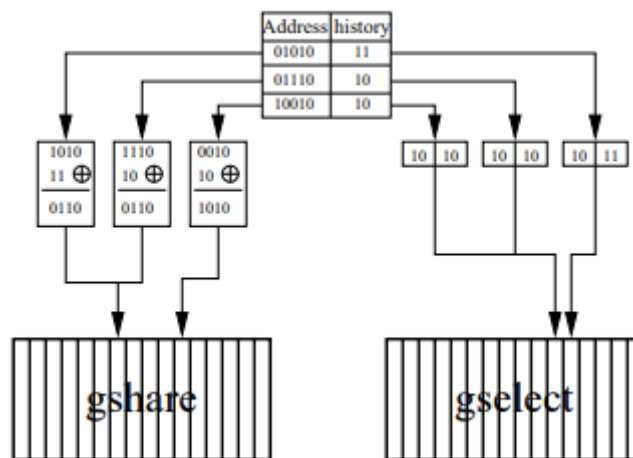


Predictor Gshare



Predictor Gskew

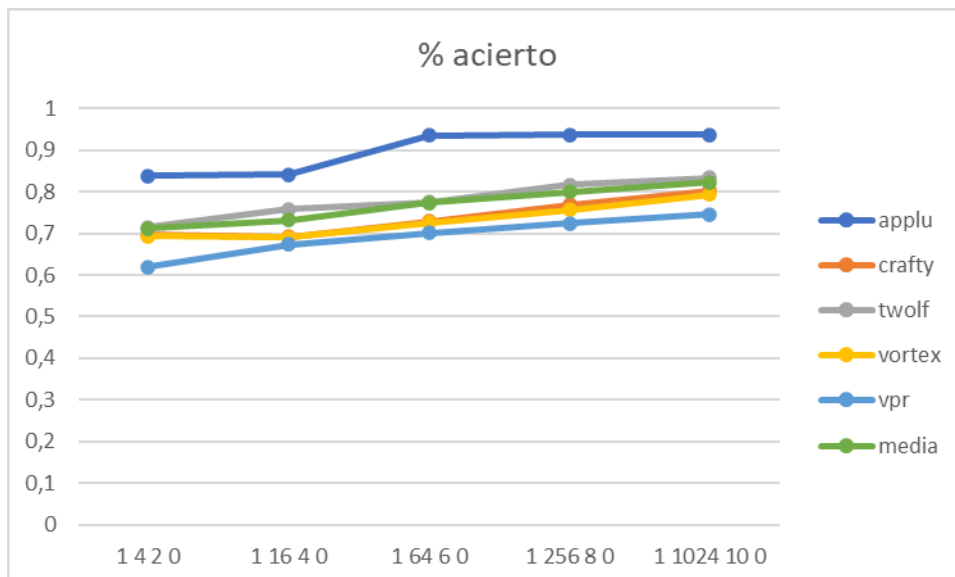
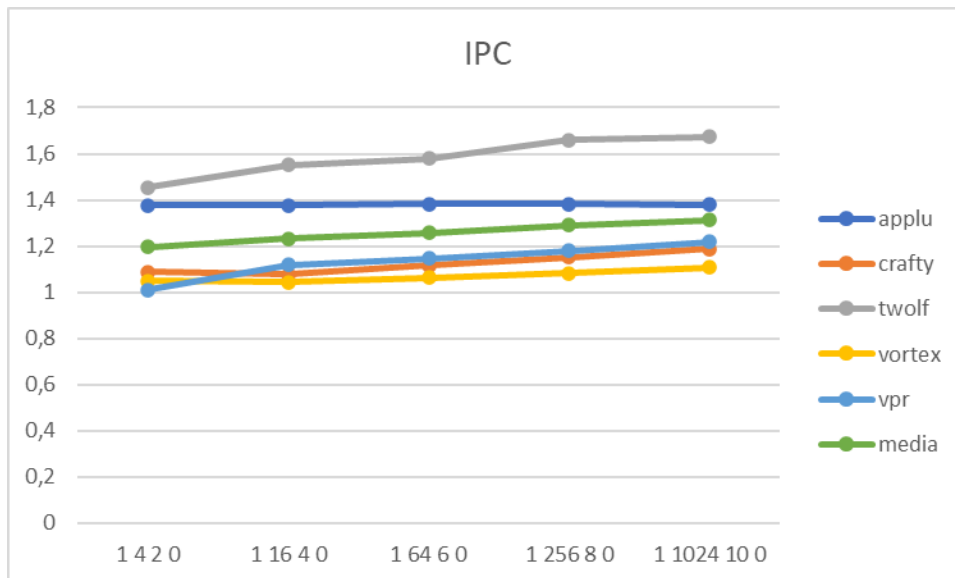
Entonces, si tenemos que comparar el predictor **Gskew** con los predictores mencionados anteriormente, en cuanto a estructuras, podemos decir que **Gskew** se parece a una mezcla de los predictores **Gag** + **Gshare**.



Para comparar **Gskew** con el resto de predictores, simularemos unos benchmarks y todos tendrán parámetros equivalentes. Como vemos en la tabla **2-level adaptive predictor sctructure**, si queremos comparar **Pag** y **Pap** en igualdad de condiciones con **Gskew**, tendremos que poner sus L1 en un tamaño de 1, lo que hace que **Pag** sea igual que **Gag**.

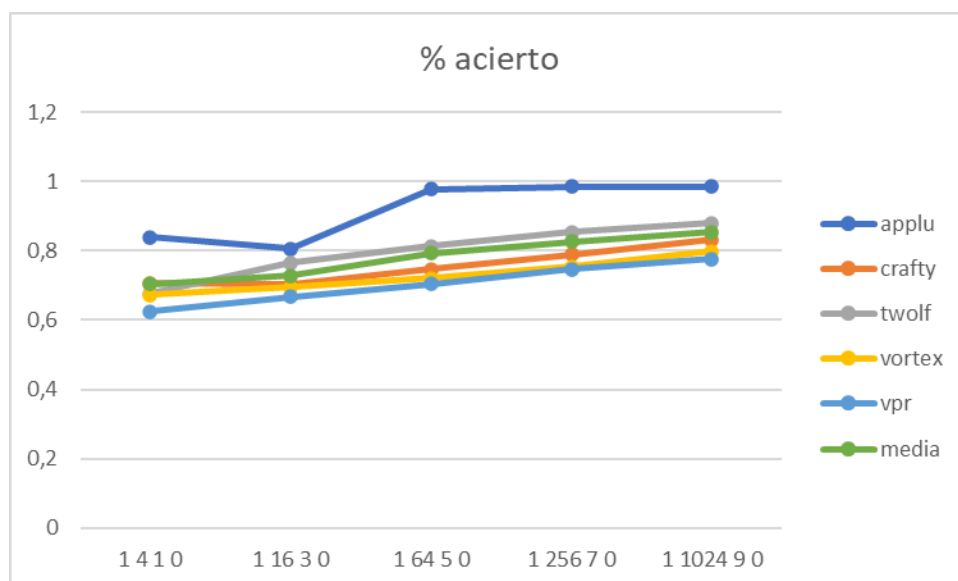
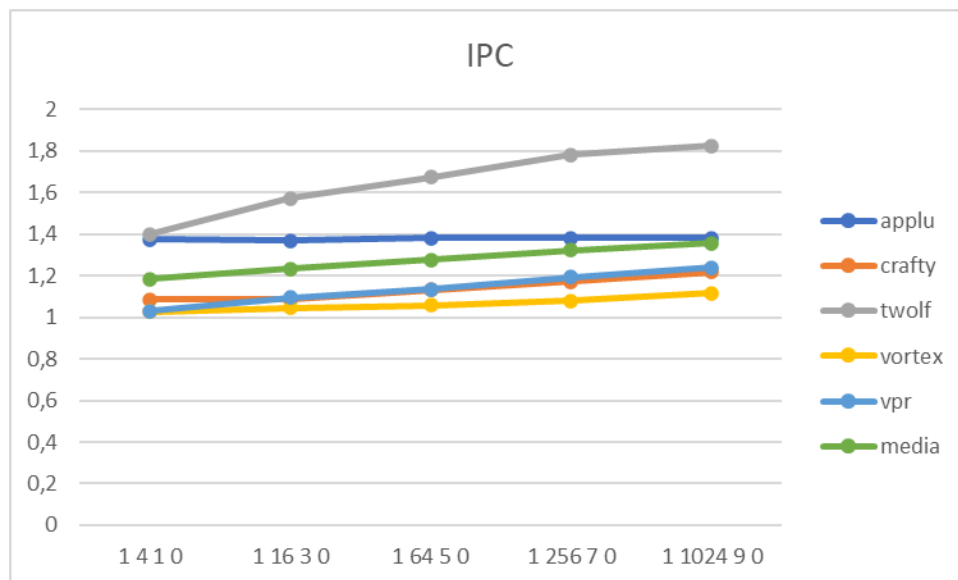
Gag/Pag:

GAg	1	W	2^W	0
PAg	N	W	2^W	0



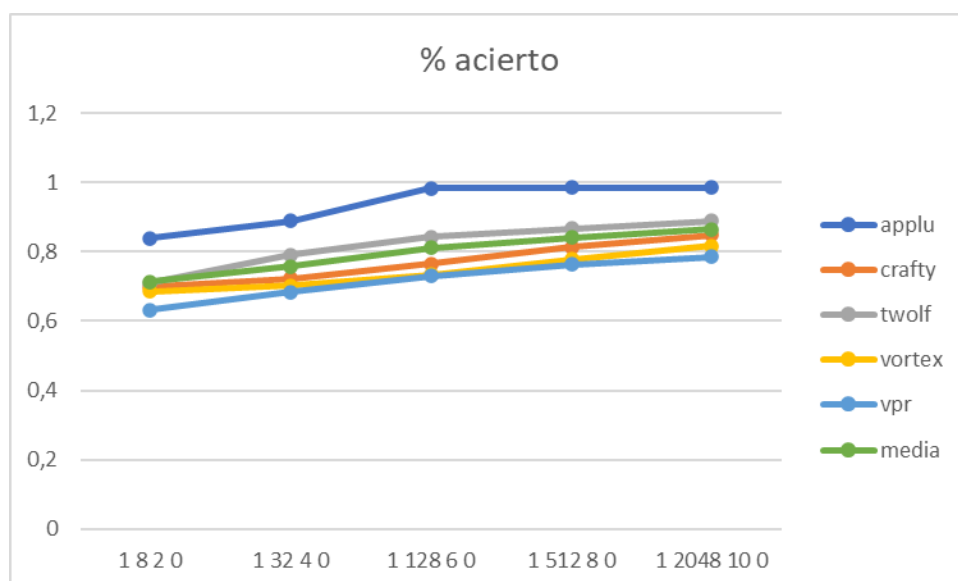
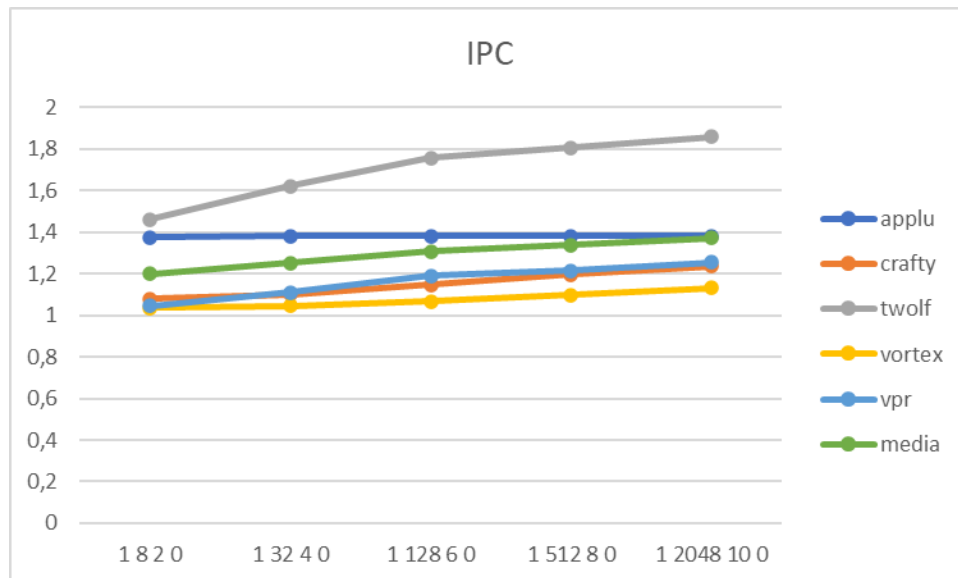
Gap:

Gap	1	W	$>2^W$	0
-----	---	---	--------	---



Pap:

Pap	N	W	2^{N+W}	0
-----	---	---	-----------	---



Como podemos observar en cuanto a las estructuras **Gskew** y funcionamiento del predictor **Gskew** se parece a una mezcla de **Gag** con una parte de la lógica del **Gshare**. Cuando comparamos las simulaciones de los diferentes predictores vemos que en cuanto a resultados el que más se parece es **Pap**. El caso es que al simular en igualdad de condiciones con **Pap** tenemos que poner L1-size a 1, por lo que los 4 predictores mencionados van a tener un funcionamiento muy parecido ya que habrá un **GBHR** y varios **PHTs**.

En cuanto a estructura se parece más a la mezcla de **Gag** y **Gshare**, pero en cuanto a performance, se parece más a **Pap**.

