Un Análisis Cuantitativo de Spec95

Agustín Fernández
Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya

"Todo lo que usted quiso saber sobre Spec95 y nunca se atrevió a simular."

UPC

Guión

Objetivos y antecedentes
Análisis de Instrucciones
Análisis de localidad
Análisis de comportamiento dinámico
Conclusiones, Trabajos pendientes e Ideas Futuras

UPC

Objetivos y antecedentes

SPEC es una compañía creada por varios fabricantes de workstations

Objetivo: definir un benchmark para evaluar computadores de nivel medio:

- standarizado
- realista
- consensuado (el benchmark y la forma de utilizarlo)
- Spec89, Spec92, Spec95, ???

Actualmente disponen de otros tests

- Plataformas Java (SPECjvm98).
- Sistemas de ficheros (SPEC SFS 2.0).
- Servidores WWW (SPECweb96).
- Supercomputadores (SPEChpc96).

Objetivos y antecedentes

Spec95

- 10 programas aritmética en coma flotante
- 8 programas de aritmética entera

- Para cada test: tres conjuntos de datos de entrada
  `*/data/ref, */data/train, */data/test`

- En cada conjunto puede haber 1 ejecución o varias:
  - 101.tomcatv (1 ejecución)
  - 126.gcc (56 ejecuciones)

Objetivos y antecedentes

Uso de benchmarks en el DAC (junio95-junio98)

- 48 artículos utilizan Spec95, Spec92 o Perfect Club
- 46 artículos utilizan otro tipo de benchmarks (o ninguno)
- Microarquitectura: 85 % de los artículos utilizan Spec95, Spec92 o Perfect Club

Uso de benchmarks en el MICRO-31 (diciembre 1998)

- 17 artículos utilizan Spec95
- 2 artículos utilizan Spec92
- 2 artículos utilizan Perfect Club
- 7 artículos utilizan otros benchmarks (Multimedia)

Objetivos y antecedentes

No es un objetivo: presentar ninguna propuesta nueva ni idea revolucionaria.

Si es un objetivo: elaborar un documento que sirva de consulta a los investigadores que están utilizando Spec95 para evaluar sus propuestas.

- Utilidad
  - estudio TLB en 145.fpppp (32 páginas)
  - Predicción saltos en 102.swim (99,61 % taken)
  - cache de instrucciones en 101.tomcatv (4.256 bytes código)
  - comportamiento subrutinas en 101.tomcatv (7 calls)
  - . . .

Objetivos y antecedentes

Características de los tests realizados

- Se ha utilizado el Spec95 completo, es decir con los conjuntos de datos que se utilizan para obtener el Specratio.
- Las simulaciones se han realizado utilizando ATOM
- Los programas se han compilado con las opciones por defecto
- Se han utilizado las librerías "shared".
- Todas las simulaciones se han ejecutado en abrell (Alpha 21164) y romeu (Alpha 21064).

Análisis de instrucciones

Objetivo: comportamiento dinámico del Spec95 a nivel de instrucciones (ISA Alpha 21164).

Para cada test se dispone de la siguiente información dinámica:

- instrucciones
- loads
- stores
- bloques básicos
- subrutinas
- saltos condicionales (+ not taken)

Esta misma información está disponible a nivel de subrutina

Análisis de instrucciones

Información que se puede obtener:

- porcentaje de referencias
- porcentaje de loads
- porcentaje de stores
- porcentaje de saltos condicionales realizados
- instrucciones por bloque básico
- instrucciones por subrutina

Análisis de instrucciones

Ejecución dinámica de instrucciónes en el SPEC95 FP

	Instructions	Loads	Stores	Blocks	Calls
101.tomcatv	26.707.284.939	7.985.136.397	2.253.657.624	406.423.209	7
102.swim	35.023.227.316	8.510.112.420	3.079.869.716	714.924.277	2.707
103.su2cor	28.387.286.599	5.796.180.333	2.882.632.014	1.052.304.445	12.223.238
104.hydro2d	42.585.845.189	11.368.661.365	3.361.224.872	2.856.047.793	12.882.427
107.mgrid	72.025.693.903	26.955.620.933	1.407.621.682	1.151.288.189	50.183
110.applu	46.051.881.414	11.102.509.094	4.565.949.126	2.342.776.038	37.845
125.turb3d	79.680.616.852	18.819.655.643	12.972.175.305	3.669.630.496	92.290.900
141.apsi	27.999.179.162	6.465.084.134	3.154.880.324	1.102.724.825	27.571.955
145.fpppp	117.968.461.010	40.443.111.523	14.545.804.110	1.665.109.066	17.842.575
146.wave5	29.889.331.462	7.004.602.522	4.338.918.540	1.115.233.685	21.446.171
Total FP	506.318.807.846	144.450.674.364	52.562.733.313	16.076.462.023	184.348.008
Avg	50.631.880.785	14.445.067.436	5.256.273.331	1.607.646.202	18.434.801

Análisis de instrucciones

Ejecución dinámica de instrucciónes en el SPEC95 INT

	Instructions	Loads	Stores	Blocks	Calls
099.go	45.032.374.295	18.583.281.711	3.996.714.279	6.521.380.084	388.708.580
124.m88ksim	78.696.259.339	19.393.203.051	8.171.100.817	16.631.751.893	1.060.850.149
126.gcc	9.815.097.687	2.326.348.993	938.761.179	1.748.014.821	87.871.617
129.compress	64.599.969.967	16.534.945.843	5.417.835.204	8.984.313.860	1.148.724.117
130.li	65.762.809.488	17.965.017.562	9.896.659.893	15.942.048.634	1.710.851.217
132.ijpeg	111.584.623.217	20.125.838.690	6.043.217.112	6.293.693.228	136.442.185
134.perl	40.241.942.899	10.926.020.005	6.169.288.541	8.123.439.630	563.894.137
147.vortex	90.180.454.922	22.780.681.055	11.845.204.647	15.209.997.974	1.407.893.727
Total INT	505.913.531.814	128.635.336.910	52.478.781.672	79.454.640.124	6.505.235.729
Avg	50.220.372.933	13.433.786.758	5.555.416.051	8.685.040.569	755.753.090

UPC

Análisis de instrucciones

Datos relevantes en el SPEC95 FP

	%loads	%stores	%refer	Inst / BB	Inst / call	% taken
101.tomcatv	77,99	22,01	38,34	65,71	3.815.326.419,86	97,47
102.swim	73,43	26,57	33,09	48,99	12.938.022,65	99,61
103.su2cor	66,79	33,21	30,57	26,98	2.322,40	86,18
104.hydro2d	77,18	22,82	34,59	14,91	3.305,73	77,42
107.mgrid	95,04	4,96	39,38	62,56	1.435.260,82	91,23
110.applu	70,86	29,14	34,02	19,66	1.216.855,10	61,93
125.turb3d	59,20	40,80	39,90	21,71	863,36	47,61
141.apsi	67,20	32,80	34,36	25,39	1.015,49	71,30
145.fpppp	73,55	26,45	46,61	70,85	6.611,63	44,05
146.wave5	61,75	38,25	37,95	26,80	1.393,69	73,63
Total FP	73,32	26,68	38,91	31,49	2.746,54	68,58

UPC

Análisis de instrucciones

Datos relevantes en el SPEC95 INT

	%loads	%stores	%refer	Inst / BB	Inst / call	% take
099.go	82,30	17,70	50,14	6,91	115,85	59,60
124.m88ksim	70,36	29,64	35,03	4,73	74,18	56,80
126.gcc	71,25	28,75	33,27	5,61	111,70	56,72
129.compress	75,32	24,68	33,98	7,19	56,24	51,62
130.li	64,48	35,52	42,37	4,13	38,44	42,13
132.ijpeg	76,91	23,09	23,45	17,73	817,82	80,50
134.perl	63,91	36,09	42,48	4,95	71,36	46,79
147.vortex	65,79	34,21	38,40	5,93	64,05	31,09
Total INT	71,02	28,98	35,80	6,37	77,77	50,53

UPC

Análisis de instrucciones

Resumen ejecución dinámica SPEC95

	Instructions	Loads	Stores	Blocks	Calls
FP	506.318.807.846	144.450.674.364	52.562.733.313	16.076.462.023	184348008
INT	505.913.531.814	128.635.336.910	52.478.781.672	79.454.640.124	6505235729
Spec95	1.012.232.339.660	273.086.011.274	105.041.514.985	95.531.102.147	6689583737
Avg FP	50.631.880.785	14.445.067.436	5.256.273.331	1.607.646.202	18434800,8
Avg INT	50.220.372.933	13.433.786.758	5.555.416.051	8.685.040.569	755753090,4
Avg Spec95	50.448.988.406	13.995.609.357	5.389.225.651	4.753.154.810	346131818,4

	%loads	%stores	%refer	Inst / BB	Inst / call	% taken
FP	73,32	26,68	38,91	31,49	2.746,54	68,58
INT	70,74	29,26	37,81	5,78	66,45	48,20
Spec95	72,20	27,80	38,42	10,61	145,75	53,29

UPC

Análisis de instrucciones

Tipos de instrucciones ejecutadas

		FP	INT	Spec95	FP	INT	Spec95
0	Floating Point Operate	198.425.141.165	677.569.885	199.102.711.050	39,19	0,13	19,67
1	Floating Point Load/Store	181.033.282.500	1.411.348.024	182.444.630.524	35,75	0,28	18,02
2	Floating Point Branch	1.709.220.758	261.086.041	1.970.306.799	0,34	0,05	0,19
3	Integer Load/Store	16.018.227.295	183.553.916.056	199.572.143.351	3,16	36,28	19,72
4	Byte Manipulation	9.763	26.976.878.073	26.976.887.836	0,00	5,33	2,67
5	Control	12.816.401.658	67.747.902.347	80.564.304.005	2,53	13,39	7,96
6	Integer Arithmetic	89.152.175.909	143.133.931.526	232.286.107.435	17,61	28,29	22,95
7	Logical and Shift	7.164.348.798	82.150.899.344	89.315.248.142	1,41	16,24	8,82
8	Miscellaneous	0	394	394	0,00	0,00	0,00
9	VAX Compatibility	0	0	0	0,00	0,00	0,00
10	Processor Specific PALcode	0	122	122	0,00	0,00	0,00
	Total	506.318.807.846	505.913.531.812	1.012.232.339.658	100	100	100

Análisis de localidad: datos

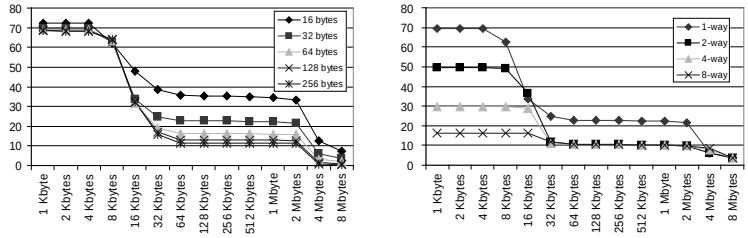
Objetivo: estudiar como se comportan las referencias a datos.

- Caches de datos (L1), variando parámetros básicos:
  - Tamaño: 1Kbyte hasta 8 Mbytes
  - Asociatividad: 1-way hasta 8-way
  - Tamaños de línea: 16 bytes hasta 256 bytes (1-way)
  - Sólo loads.
  - Loads y Stores (write allocate y copy back).
- Mapas de memoria
  - footprint
  - histogramas de localidad

Análisis de localidad: datos

Cache de datos (tasa de fallos)

- 101.tomcatv (only loads)



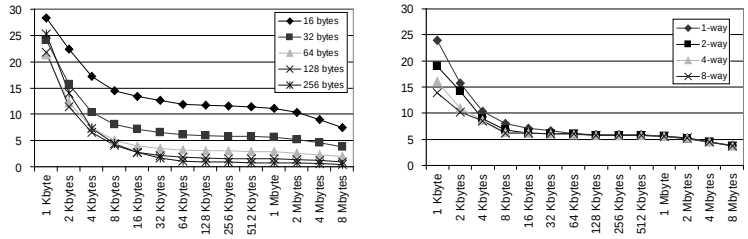
  - Características similares en muchos programas FP
    - Sensible al tamaño de línea (localidad espacial)
    - Sensible a la asociatividad en caches pequeñas
    - Punto de inflexión

UPC

Análisis de localidad: datos

Cache de datos (tasa de fallos)

- 110.mgrid (only loads)

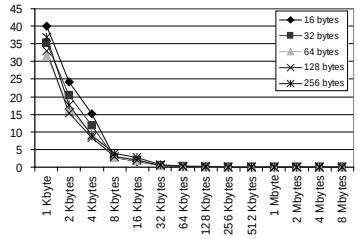


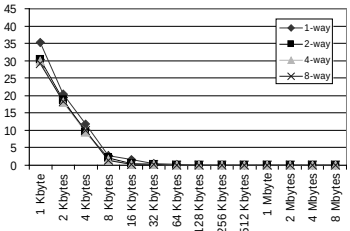
UPC

Análisis de localidad: datos

Cache de datos (tasa de fallos)

- 145.fpppp (only loads)



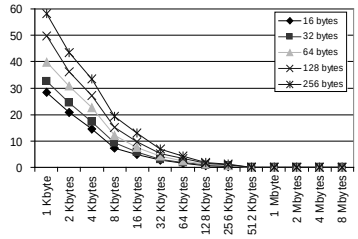


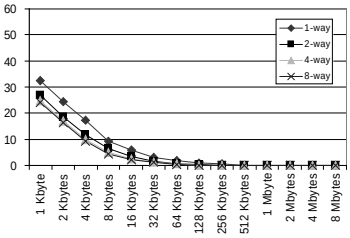
UPC

Análisis de localidad: datos

Cache de datos (tasa de fallos)

- 099.go (only loads)





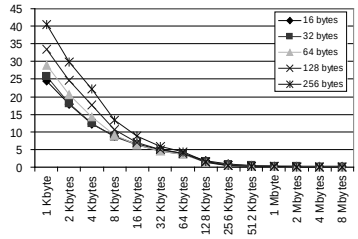
- Características similares en muchos programas INT
  - Sensible al tamaño de línea (poca localidad espacial)
  - Sensible a la asociatividad en caches pequeñas

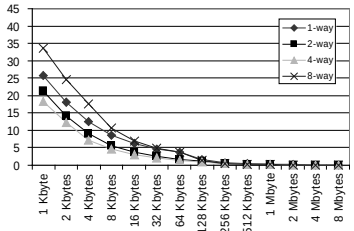
UPC

Análisis de localidad: datos

Cache de datos (tasa de fallos)

- 126.gcc (only loads)



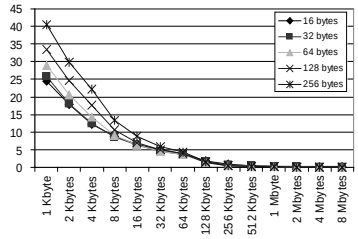


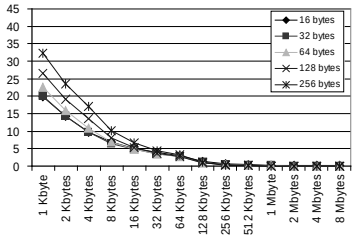
UPC

Análisis de localidad: datos

Cache de datos (only loads, respecto a loads & stores)

- 126.gcc





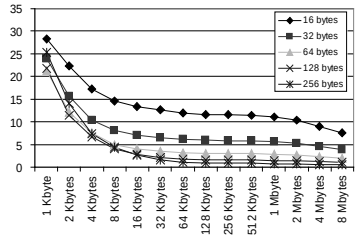
- La localidad puede comportarse de forma similar, pero las tasas de fallos pueden ser muy diferentes.

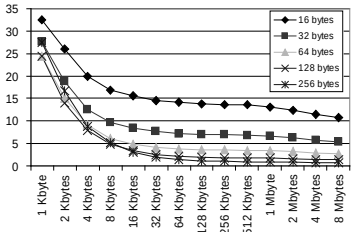
UPC

Análisis de localidad: datos

Cache de datos (only loads, respecto a loads & stores)

- 110.mgrid



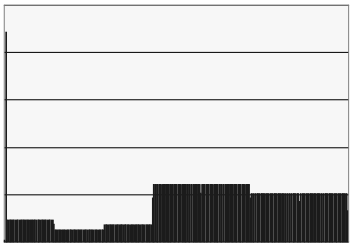


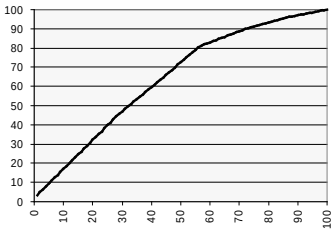
UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 101.tomcatv





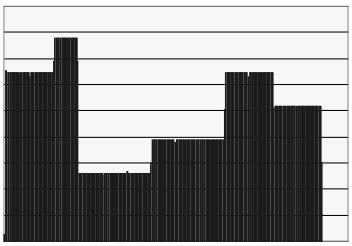
- Comportamiento similar en la mayoría de los programas FP
  - footprint con grandes bloques
  - Baja reutilización

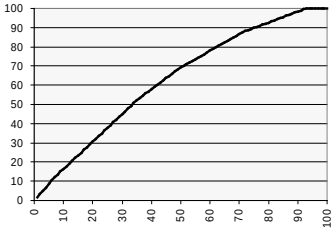
UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 102.swim



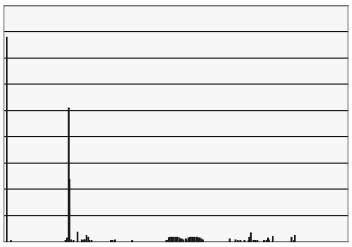


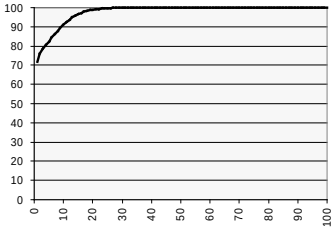
UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 099.go





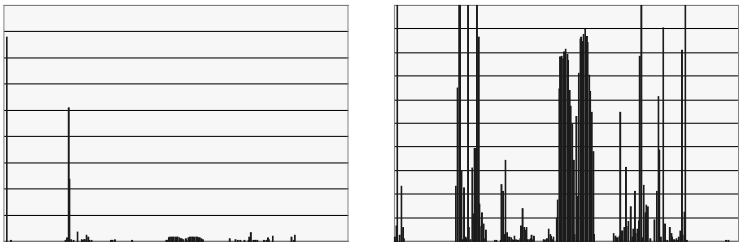
- Comportamiento similar en la mayoría de los programas INT
  - footprint con grandes picos
  - Alta reutilización reutilización

UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 099.go

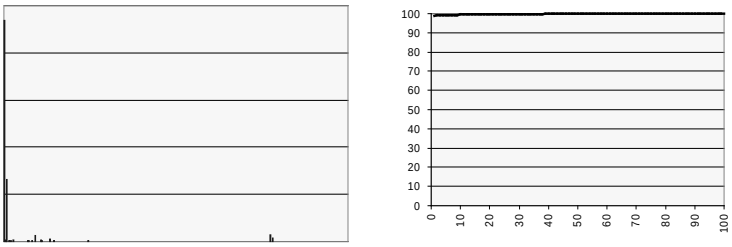


UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 147.vortex

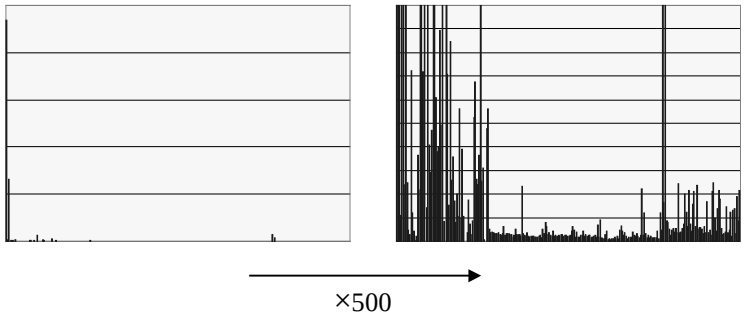


UPC

Análisis de localidad: datos

Mapas de memoria: footprints e Histogramas

- 147.vortex



UPC

Análisis de localidad: instrucciones

Objetivo: estudiar como se comportan las referencias a datos.

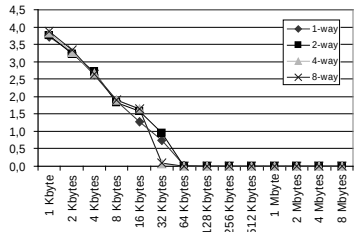
- Caches de instrucciones (L1), variando parámetros básicos:
  - Tamaño: 1Kbyte hasta 8 Mbytes
  - Asociatividad: 1-way hasta 8-way
  - Tamaños de línea: 32 bytes
- Mapas de memoria
  - histogramas de localidad

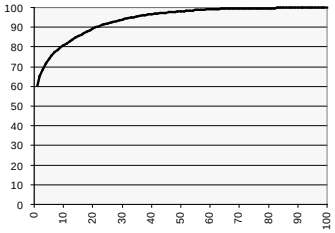
UPC

Análisis de localidad: instrucciones

Cache de datos (tasa de fallos)

- FP





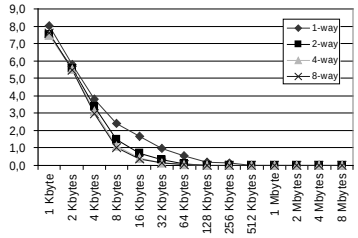
  - Características similares en muchos programas FP
    - Poca influencia de la asocitividad
    - Tasas de fallos pequeñas (excepción del 145.fpppp, hasta 12%)
    - Poca reutilización

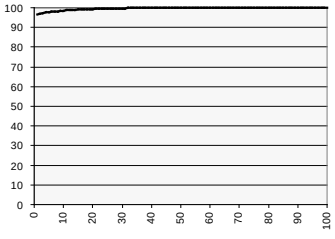
UPC

Análisis de localidad: instrucciones

Cache de datos (tasa de fallos)

- INT





  - Características similares en muchos programas FP
    - Poca influencia de la asocitividad (excepto 1-way)
    - Tasas de fallos algo mayores
    - Alta reutilización

UPC

Análisis de la localidad

Working sets y TLB

	Instructions	WS codigo	References	WS datos	% fallos TLB
101.tomcatv	26.707.284.939	4.256	10.238.794.026	14.709.792	0,039065
102.swim	35.023.227.316	7.200	11.589.982.142	14.738.912	0,040816
103.su2cor	28.387.286.599	38.912	8.678.812.351	24.355.840	0,040151
104.hydro2d	42.585.845.189	37.696	14.729.886.236	8.814.656	0,068152
107.mgrid	72.025.693.903	11.936	28.363.242.614	7.644.512	0,016529
110.applu	46.051.881.414	47.968	15.668.458.219	28.262.176	0,246118
125.turb3d	79.680.616.852	28.032	31.791.830.947	25.966.944	1,378334
141.apsi	27.999.179.162	68.256	9.619.964.457	2.012.416	0,329980
145.fpppp	117.968.461.010	43.904	54.988.915.632	173.920	0,000000
146.wave5	29.889.331.462	52.032	11.343.521.061	41.968.704	1,902845

UPC

Análisis de la localidad

Working sets y TLB

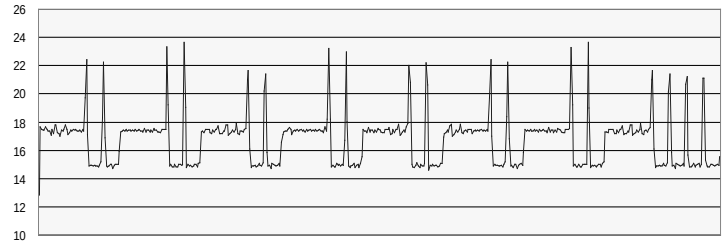
	Instructions	WS codigo	References	WS datos	% fallos TLB
099.go	45.032.374.295	285.920	22.579.995.990	459.200	0,000000
124.m88ksim	78.696.259.339	78.208	27.564.303.868	38.721.184	0,002172
126.gcc	9.815.097.687	355.872	3.265.110.172	1.680.096	0,971733
129.compress	64.599.969.967	6.208	21.952.781.047	36.796.736	1,054559
130.li	65.762.809.488	35.936	27.861.677.455	576.192	0,000165
132.ijpeg	111.584.623.217	63.008	26.169.055.802	7.632.192	0,001874
134.perl	40.241.942.899	89.824	17.095.308.546	37.177.952	0,009050
147.vortex	90.180.454.922	342.496	34.625.885.702	33.146.304	0,296361

UPC

Análisis del comportamiento dinámico

¿Es necesario, simular el programa completo para obtener un determinado dato? (p.e. la tasa de fallos de la cache)

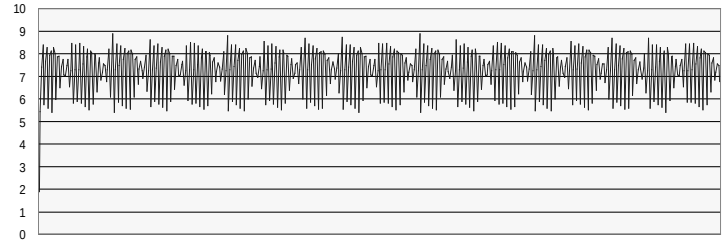
- cache 32 Kbytes, directa, 32 bytes/línea
- 103.su2cor



UPC

Análisis del comportamiento dinámico

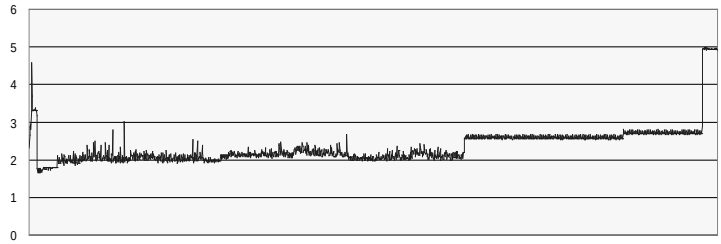
- 110.applu



UPC

Análisis del comportamiento dinámico

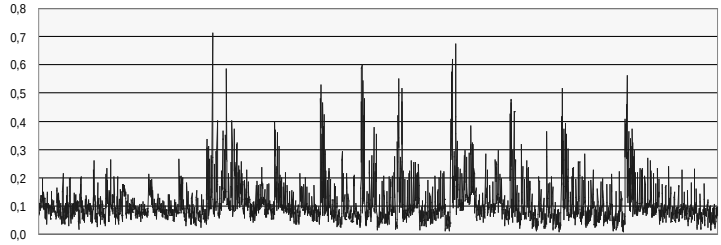
- 130.li



UPC

Análisis del comportamiento dinámico

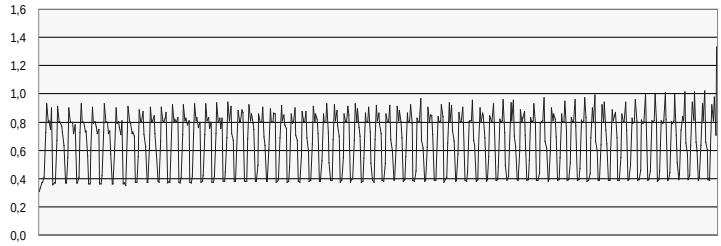
- 145.fpppp



UPC

Análisis del comportamiento dinámico

- 132.ijpeg



UPC

Conclusiones, Trabajos pendientes e Ideas futuras

Hemos realizado un análisis cuantitativo de Spec95. Los resultados obtenidos pueden servir para explicar el comportamiento de un programa al aplicarle alguna transformación (hardware o software).

Trabajos Pendientes

- Caches unificadas de datos e instrucciones
- Estudio más específico del comportamiento de los stores en la cache
- nivel L2 de cache

Ideas futuras

- ¿Es necesario utilizar el conjunto de datos "ref", o podríamos utilizar el conjunto de datos de "test" ("train") y extrapolar los resultados?
- Estudiar el comportamiento dinámico de los programas, para realizar estimaciones de rendimiento sin tener que simular todo el test.

UPC