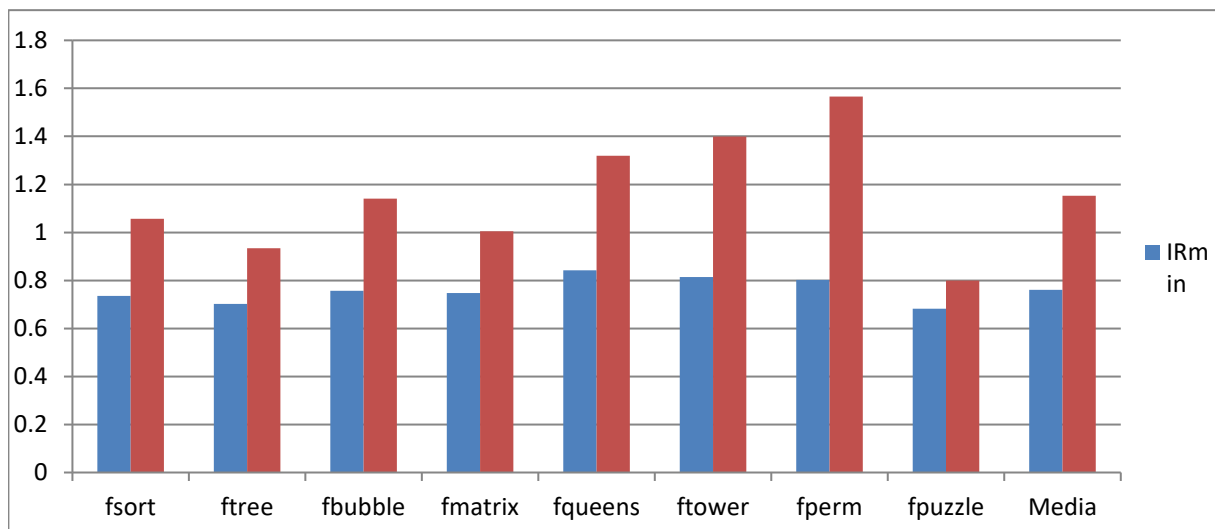


	<i>f</i> sort	<i>f</i> tree	<i>f</i> bubble	<i>f</i> matrix	<i>f</i> queens	<i>f</i> tower	<i>f</i> perm	<i>f</i> puzzle	<i>Media</i>
IRmin	0.736	0.701682	0.757493	0.747852	0.842562	0.81353	0.80178	0.682565	0.760414
IRmax	1.057	0.933834	1.140583	1.004132	1.318666	1.39933	1.565578	0.798885	1.15227

<i>Minim</i>	<i>f</i> sort	<i>f</i> tree	<i>f</i> bubble	<i>f</i> matrix	<i>f</i> queens	<i>f</i> tower	<i>f</i> perm	<i>f</i> puzzle
r. Instr exe	72101	136040	206035	231814	206420	251149	355643	6825651
. Cicli in ex	97983	193877	271996	309973	244991	308715	443567	10000001

<i>Maxim</i>	<i>f</i> sort	<i>f</i> tree	<i>f</i> bubble	<i>f</i> matrix	<i>f</i> queens	<i>f</i> tower	<i>f</i> perm	<i>f</i> puzzle
r. Instr exe	72101	136040	206035	231814	206420	251149	355643	6881833
. Cicli in ex	68203	145679	180640	230860	156537	179478	227164	8614302



Concluzii:

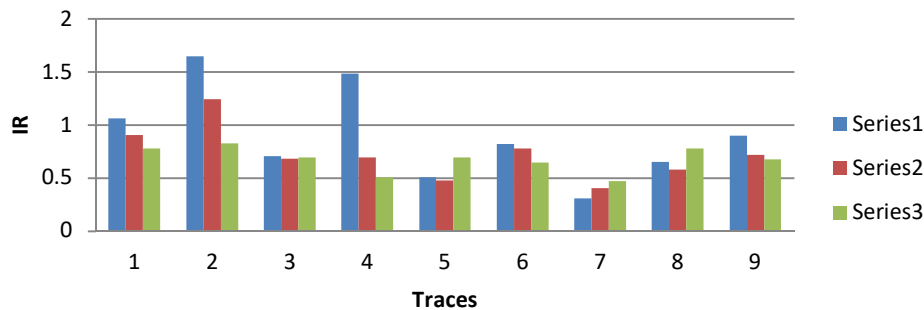
1. Pentru simulările maxime sunt mai puțini cicli de execuție
2. IRmax folosește mai puțini cicli de execuție
3. Performanța este semnificativ mai bună pe modelul maximal

PROBLEM Rezultate urmate de grafice privind influenta ratei de fetch (FR) asupra ratei de procesare IR(FR) si asupra ratei de miss în cache-ul de instructiuni RmissIC(FR).

ISSUE RATE

	SORT	BUBBLE	MATRIX	PERM	PUZZLE	QUEEN	TOWER	TREE	MEDIE
FR=4	1.058	1.648	0.705	1.483	0.505	0.818	0.308	0.651	0.897
FR=8	0.904	1.242	0.678	0.692	0.474	0.777	0.401	0.578	0.71825
FR=16	0.776	0.825	0.69	0.507	0.691	0.644	0.469	0.776	0.67225

ISSUE RATE

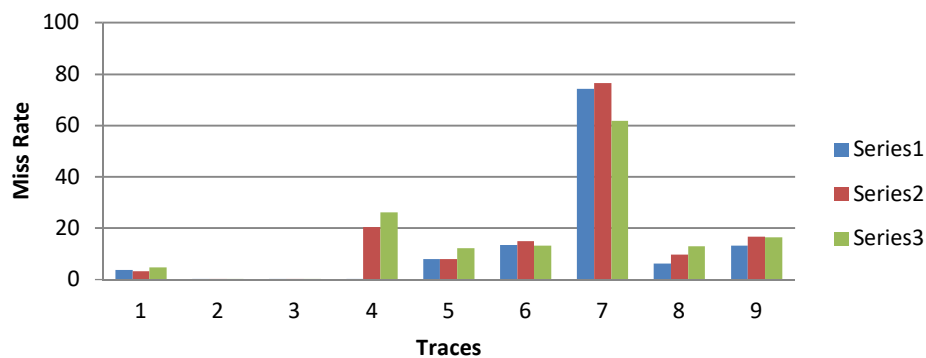


MISS RATE

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEEN	TOWER	TREE	MEDIE
FR=4	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.1025
FR=8	3.09	0.04	0.05	20.38	7.94	14.82	76.46	9.52	16.5375
FR=16	4.7	0.05	0.06	26.13	12	13.2	61.7	12.94	16.3475

Rata de miss in cacheul de instructiuni este cea mai mare in cazul trace-ului tower cand FR=8

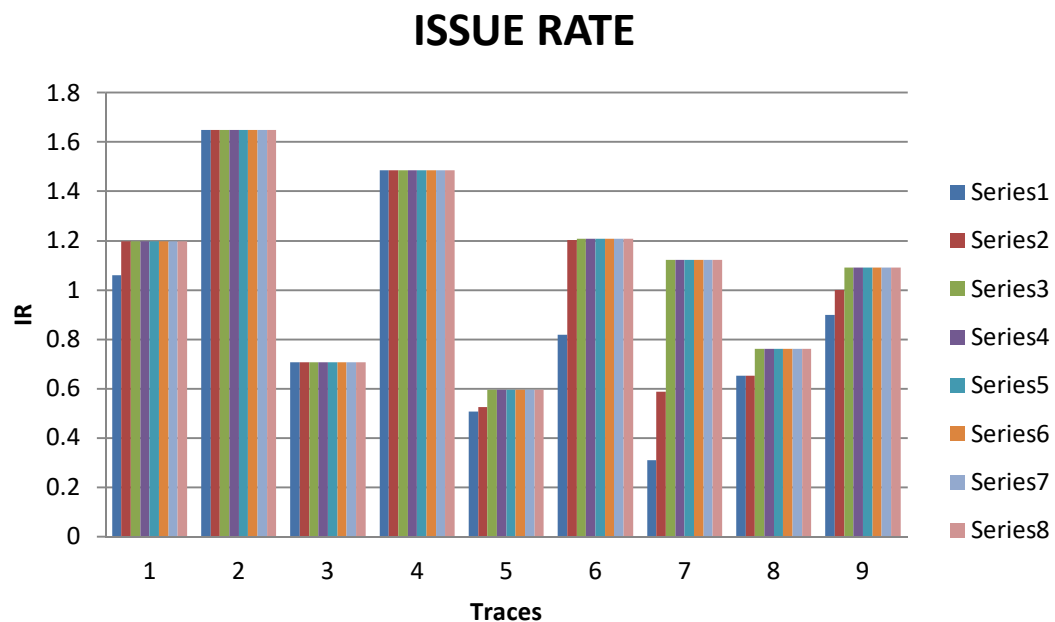
MISS RATE



PROBLEMA : Studiați influența capacității cache-ului de instrucțiuni asupra ratei de procesare $IR(SIZE_IC)$ și asupra ratei de miss la cache-ul de instrucțiuni $RmissIC(SIZE_IC)$.

	ISSUE RATE								
	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENS	TOWER	TREE	MEDIE
SIZE_IC=64	1.058	1.648	0.705	1.483	0.505	0.818	0.308	0.651	0.897
SIZE_IC=128	1.196	1.648	0.706	1.484	0.524	1.201	0.586	0.651	0.9995
SIZE_IC=256	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938
SIZE_IC=512	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938
SIZE_IC=1024	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938
SIZE_IC=2048	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938
SIZE_IC=4096	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938
SIZE_IC=8192	1.196	1.648	0.706	1.484	0.595	1.207	1.12	0.759	1.08938

Daca cache-ul de instructiuni este mai mic in cazul IC=64 si IC=128 se va obtine un issue rate mai mic. De la IC=256 in sus se vor obtine aceleasi performante.

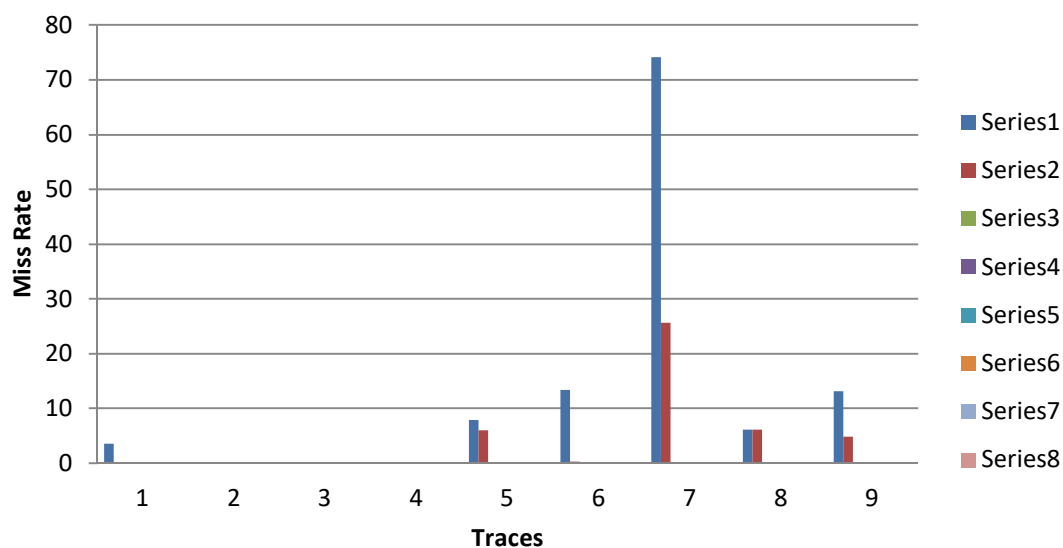


MISS RATE

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEEN	TOWER	TREE	MEDIE
SIZE_IC=64	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.1025
SIZE_IC=128	0.18	0.05	0.05	0.02	6	0.2	25.65	6.06	4.77625
SIZE_IC=256	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375
SIZE_IC=512	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375
SIZE_IC=1024	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375
SIZE_IC=2048	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375
SIZE_IC=4096	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375
SIZE_IC=8192	0.17	0.05	0.05	0.02	0.05	0.07	0.08	0.1	0.07375

Rata de miss creste in cazul in care dimensiunea cacheului de instructiuni este de 64 si 128.

MISS RATE



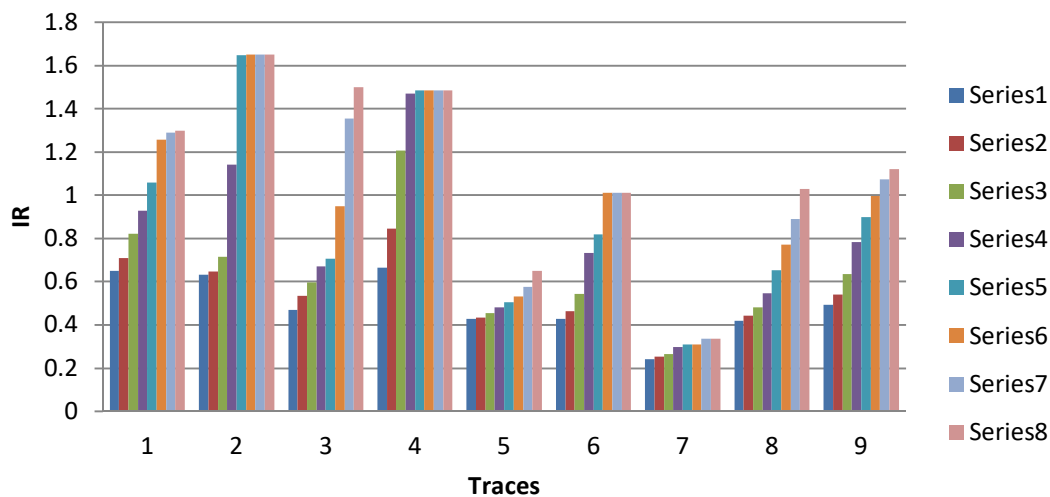
PROBLEM/ Studiati influenta capacitatii cache-ului de date asupra ratei de procesare
IR(SIZE_DC) si asupra ratei de miss la cache-ul de date **RmissDC(SIZE_DC)**.

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENST	TOWER	TREE	MEDIE
SIZE_DC=64	0.648	0.632	0.468	0.664	0.428	0.428	0.241	0.419	0.491
SIZE_DC=128	0.707	0.647	0.532	0.845	0.433	0.462	0.252	0.44	0.5398
SIZE_DC=256	0.82	0.715	0.595	1.206	0.454	0.543	0.264	0.48	0.6346
SIZE_DC=512	0.928	1.139	0.67	1.468	0.481	0.733	0.297	0.546	0.7828
SIZE_DC=1024	1.058	1.648	0.705	1.483	0.505	0.818	0.308	0.651	0.897
SIZE_DC=2048	1.257	1.649	0.948	1.483	0.529	1.009	0.308	0.77	0.9941
SIZE_DC=4096	1.288	1.649	1.354	1.483	0.576	1.009	0.334	0.888	1.0726
SIZE_DC=8192	1.297	1.649	1.5	1.483	0.649	1.009	0.334	1.027	1.1185

Pe masura ce creste dimensiunea cache-ului de creste si IR

In cazul cache-ului de instructiuni daca cresteam dimensiunea cache-ului mai mult de 256 Ir ramanea la fel

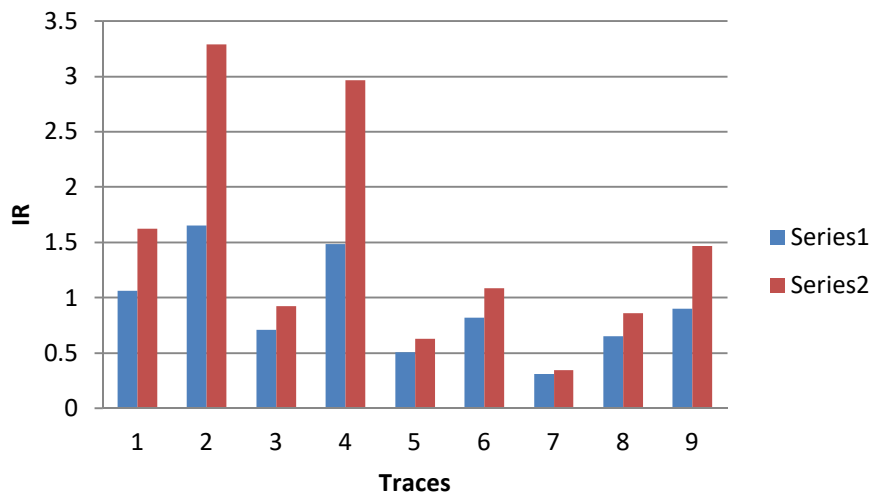
Issue Rate



	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENST	TOWER	TREE	MEDIE
SIZE_DC=64	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=128	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=256	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=512	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=1024	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=2048	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=4096	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103
SIZE_DC=8192	3.5	0.05	0.05	0.03	7.79	13.3	74.04	6.06	13.103

PROBLE Determinati influenta numarului maxim de instructiuni ce pot fi trimise simultan în executie asupra ratei de procesare IR(IRmax).

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEEN	TOWER	TREE	MEDIE
ir=2	1.058	1.648	0.705	1.483	0.505	0.818	0.308	0.651	0.897
ir=4	1.621	3.285	0.922	2.962	0.627	1.081	0.341	0.856	1.46188



Se vor genera graficele IR(BLOC_SIZE) si RmissDC(BLOC_SIZE) în cele doua ipostaze:
scriere în cache prin write back si scriere în cache prin write through.

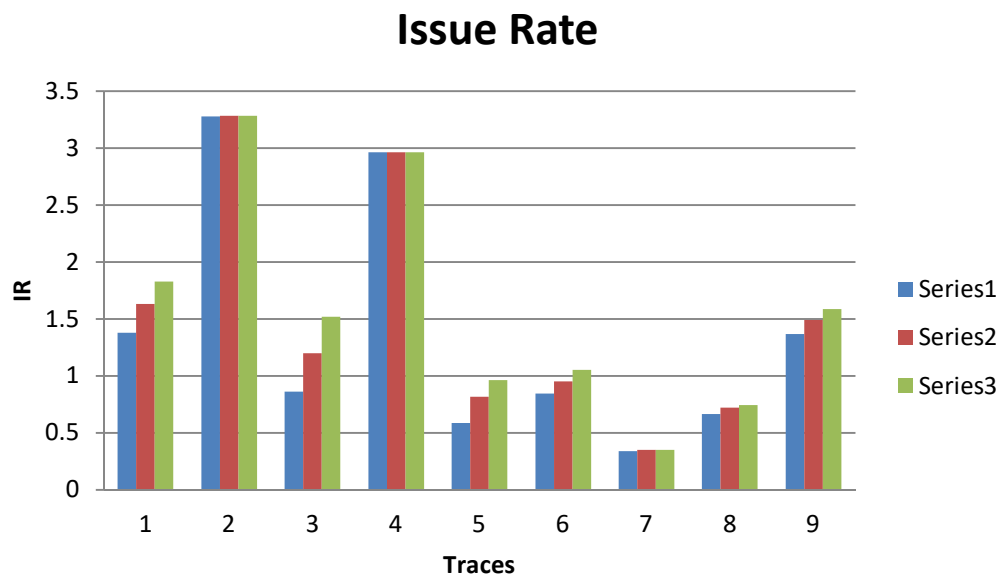
WRITE BACK

IR

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENS	TOWER	TREE	MEDIE
BLOC_SIZE=4	1.379	3.28	0.861	2.963	0.586	0.844	0.336	0.665	1.36425
BLOC_SIZE=8	1.631	3.282	1.199	2.962	0.813	0.952	0.35	0.72	1.48863
BLOC_SIZE=16	1.828	3.284	1.517	2.962	0.962	1.051	0.346	0.743	1.58663

Daca crestem bloc size-ul obtinem Ir mai mare

cele mai bune performante se obtin in cazul trace-urilor fsort, matrix, puzzle si queens
in rest chiar daca crestem dimensiunea blocului nu obtinem Ir mai mare

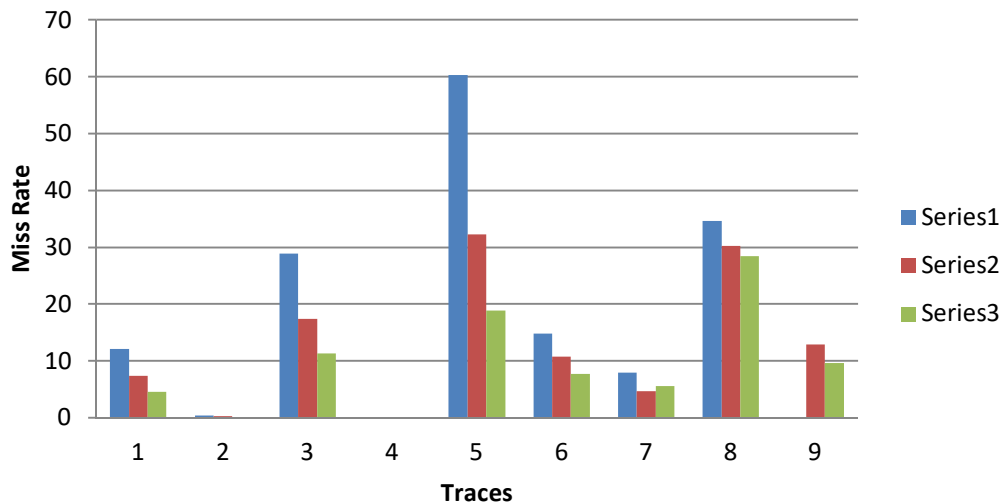


DC MISS

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENS	TOWER	TREE	MEDIE
BLOC_SIZE=4	12.1	0.36	28.84	0.04	60.21	14.75	7.93	34.54	
BLOC_SIZE=8	7.29	0.19	17.39	0.02	32.16	10.68	4.63	30.14	12.8125
BLOC_SIZE=16	4.46	0.1	11.31	0.01	18.81	7.71	5.55	28.36	9.53875

In cazul cu write back daca creste dimensiunea blocului obtinem rata de miss mai mica
 Cu cat bloc size-ul este mai mic rata de miss este mai mare.

Miss Rate

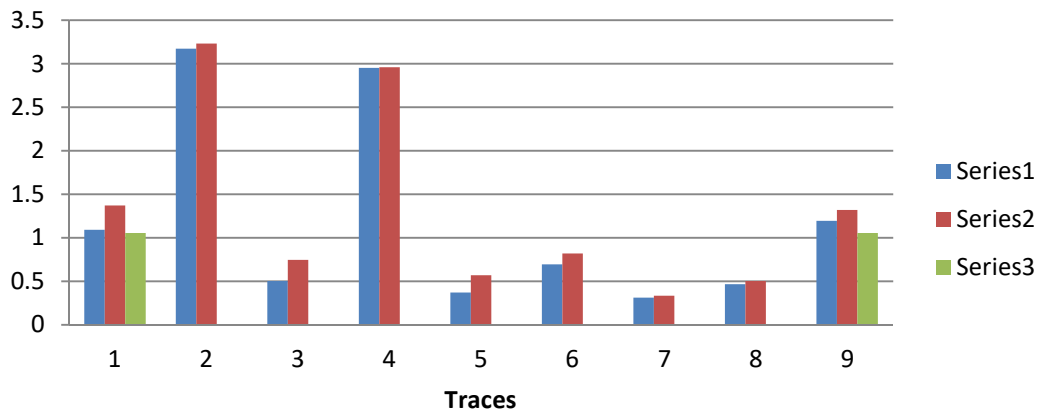


WRITE THROUGH

IR

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENS	TOWER	TREE	MEDIE
BLOC_SIZE=4	1.086	3.174	0.502	2.95	0.369	0.692	0.307	0.466	1.19325
BLOC_SIZE=8	1.37	3.227	0.746	2.955	0.568	0.814	0.331	0.502	1.31413
BLOC_SIZE=16	1.056								1.056

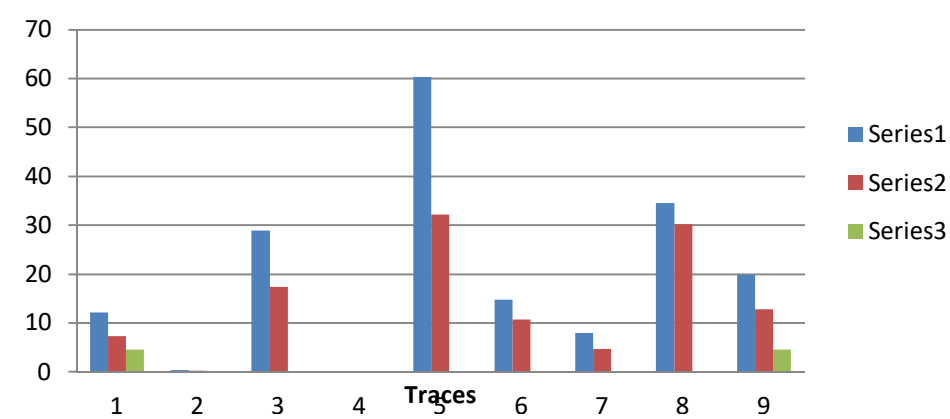
La write through cele mai bune performante ale IR sunt in cazul dimensiunii blocului de 8
 Se obtine performanta mai buna la Bloc_size=8 decat in cazul bloc_size=16.



DC MISS

	FSORT	FBUBL	MATRIX	PERM	PUZZLE	QUEENS	TOWER	TREE	MEDIE
BLOC_SIZE=4	12.1	0.36	28.84	0.04	60.21	14.75	7.93	34.54	19.8463
BLOC_SIZE=8	7.29	0.19	17.39	0.02	32.16	10.68	4.63	30.14	12.8125
BLOC_SIZE=16	4.46								4.46

In cazul write through, rata de miss creste cand marimea blocului=4
Daca crestem marimea blocului rata de miss scade
La marimea blocului=16 obtinem cea mai mica rata de miss



1. Rezultate urmate de grafice privind influența capacității cache-ului asupra ratei de procesare IR(DM_size) și asupra ratei de miss în cache-ul de instrucțiuni RmissIC(DM_size) în cele trei solutii:

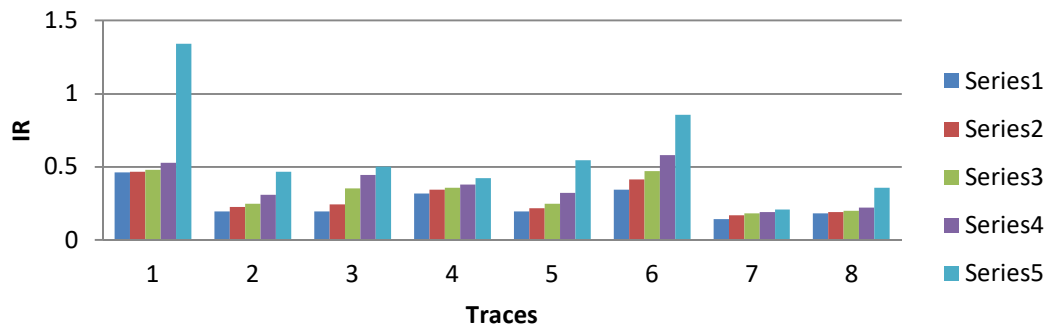
- a) fără victim cache.
- b) cu victim cache simplu.
- c) cu selective victim cache.

a)

ISSUE RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	0.4612	0.1955	0.1915	0.3179	0.1921	0.3428	0.1422	0.1786
FR_size =64	0.4658	0.2235	0.2396	0.3408	0.2159	0.4112	0.1693	0.1871
FR_size =128	0.4774	0.2448	0.3511	0.3543	0.2455	0.4708	0.1806	0.1975
FR_size =256	0.5281	0.3092	0.4429	0.3787	0.3219	0.5786	0.1903	0.2215
FR_size =1024	1.3385	0.4654	0.499	0.4208	0.5446	0.8542	0.208	0.3566

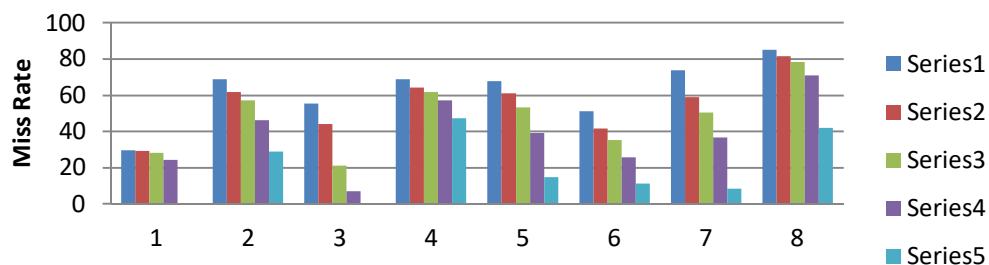
Issue Rate



MISS RATE

	bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	29.36	68.8	55.39	68.7	67.69	51	73.67	84.81
FR_size =64	28.97	61.77	43.85	64.2	60.86	41.46	58.89	81.51
FR_size =128	28.03	57.15	21.01	61.52	53.16	34.91	50.12	78.21
FR_size =256	24.29	45.87	6.72	57.08	38.97	25.59	36.39	70.91
FR_size =1024	0.19	28.72	0.02	47.13	14.72	11.17	8.2	41.97

Miss Rate



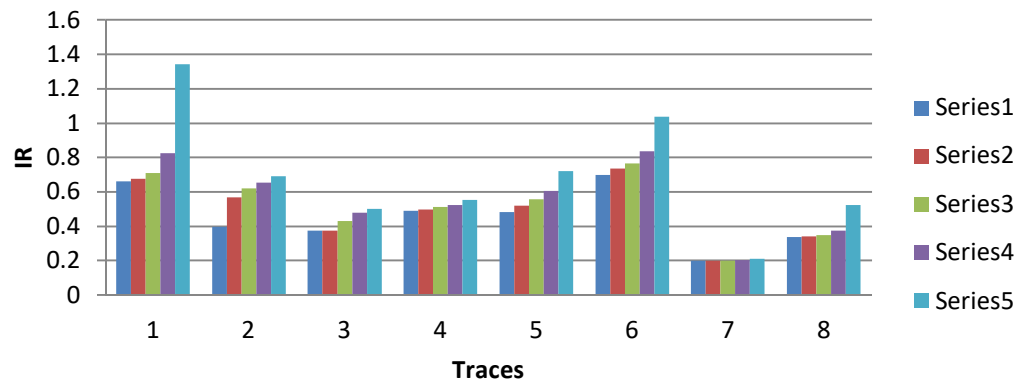
Traces

b)

ISSUE RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	0.6613	0.3936	0.3731	0.4877	0.4792	0.6978	0.1969	0.3348
FR_size =64	0.6738	0.5671	0.3748	0.4945	0.5181	0.7338	0.1978	0.338
FR_size =128	0.707	0.6177	0.4307	0.5097	0.5543	0.7636	0.1963	0.3471
FR_size =256	0.8246	0.6529	0.4768	0.5209	0.6046	0.8357	0.2006	0.374
FR_size =1024	1.3395	0.6907	0.4991	0.5519	0.7208	1.0357	0.2106	0.5233

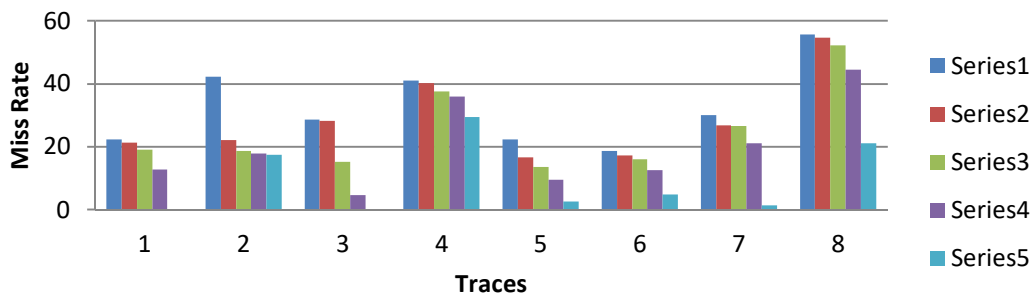
Issue Rate



MISS RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	22.16	42.12	28.49	40.91	22.15	18.6	29.92	55.56
FR_size =64	21.25	22.07	28.14	40.1	16.55	17.24	26.74	54.64
FR_size =128	19	18.53	15.05	37.55	13.54	15.87	26.47	52.03
FR_size =256	12.68	17.67	4.57	35.93	9.37	12.4	20.97	44.39
FR_size =1024	0	17.42	0	29.3	2.4	4.71	1.3	21.09

Miss Rate

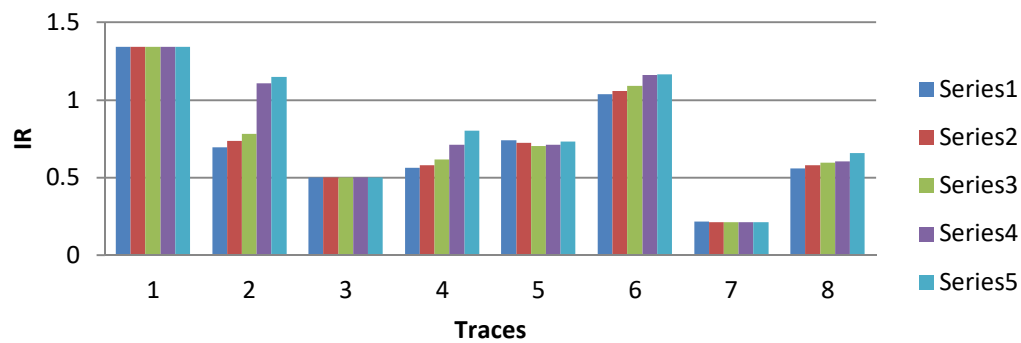


c)

ISSUE RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	1.3394	0.6949	0.4991	0.5606	0.7368	1.0372	0.217	0.5586
FR_size =64	1.3394	0.7329	0.4991	0.5766	0.7205	1.0577	0.2107	0.5767
FR_size =128	1.3394	0.7801	0.4991	0.6131	0.7007	1.0907	0.2107	0.5929
FR_size =256	1.3394	1.1072	0.4991	0.71	0.7105	1.1594	0.2107	0.6027
FR_size =1024	1.3394	1.1458	0.4991	0.8009	0.7324	1.1632	0.2108	0.6546

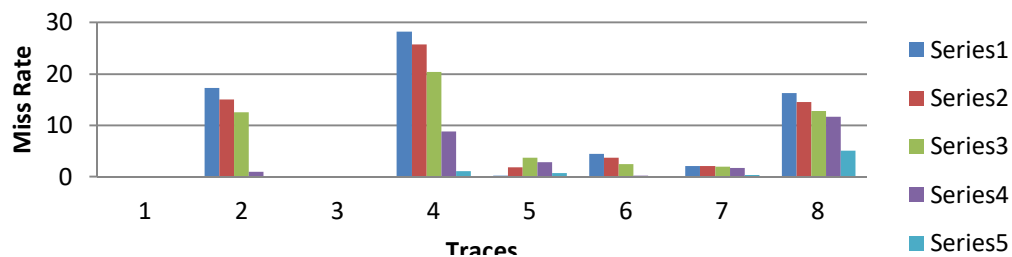
Issue Rate



MISS RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	0	17.2	0	28.17	0.18	4.4	2.05	16.23
FR_size =64	0	14.96	0	25.68	1.76	3.61	2.04	14.45
FR_size =128	0	12.49	0	20.32	3.62	2.45	1.93	12.69
FR_size =256	0	0.95	0	8.78	2.79	0.18	1.69	11.67
FR_size =1024	0	0	0	1.02	0.66	0	0.31	5.02

Miss Rate

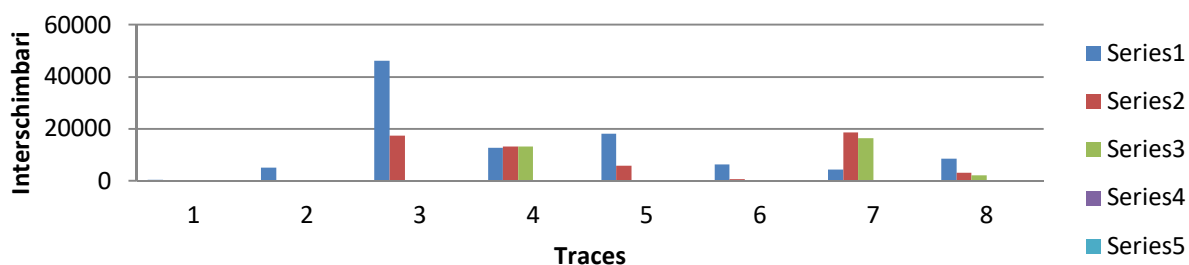


2. Determinați în ce măsură selective victim cache-ul reduce numărul de interschimbări dintre cache-ul principal și cel victimă Interchgs(DM_size) în situațiile:

- cu victim cache simplu.
- cu selective victim cache.

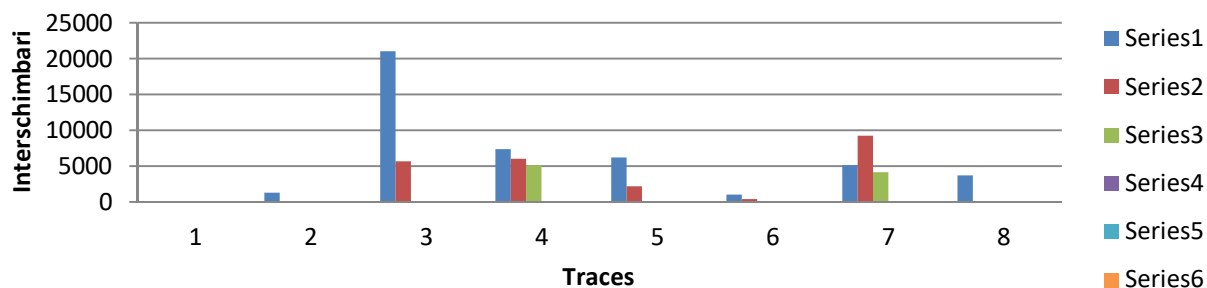
a)

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
DM_size=32	399	5049	46135	12685	17971	6241	4224	8477
DM_size=64	2	5	17321	13197	5746	459	18483	2992
DM_size=128	0	2	0	12992	56	2	16374	1996
DM_size=256	0	0	0	4	0	0	0	0
DM_size=512	0	0	0	5	0	0	0	0
DM_size=1024	0	0	0	0	0	0	0	0



b)

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
DM_size=32	7	1256	21006	7307	6218	987	5156	3680
DM_size=64	9	11	5682	6006	2137	340	9222	18
DM_size=128	0	2	0	5126	25	5	4119	13
DM_size=256	0	0	0	30	0	0	0	0
DM_size=512	0	0	0	6	0	0	0	0
DM_size=1024	0	0	0	0	0	0	0	0

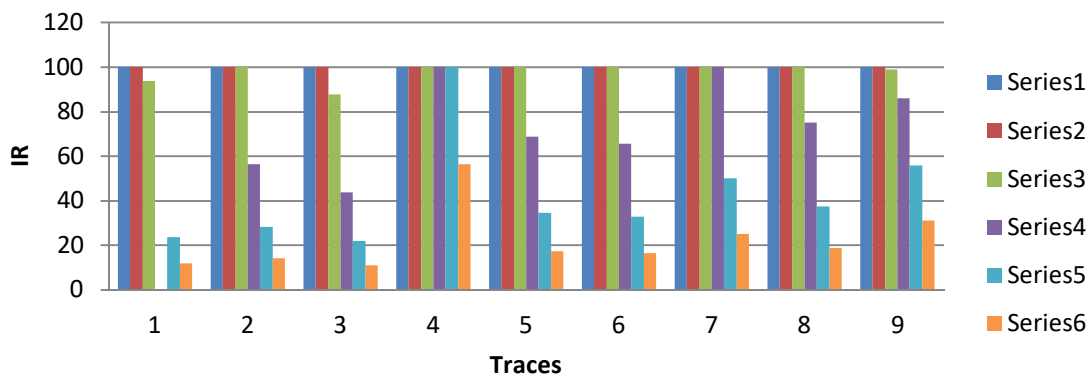


3. Studiați influența capacității cache-ului de instrucțiuni asupra ratei de utilizare a resp. cache Usage(DM_size) în situațiile:

- fără victim cache.
- cu victim cache simplu.
- cu selective victim cache.

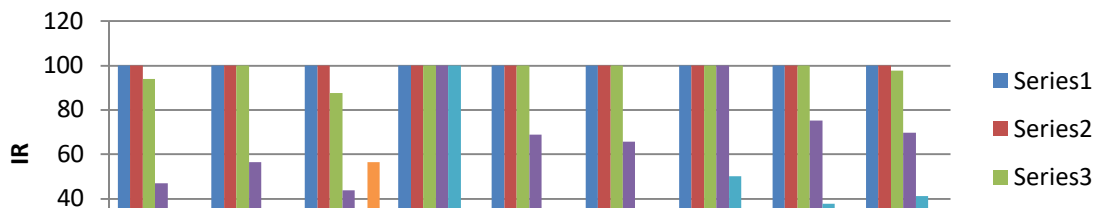
a)

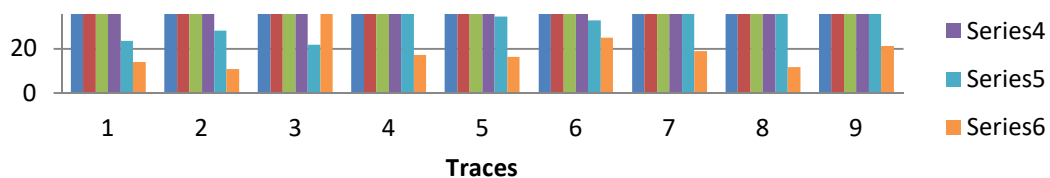
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
32	100	100	100	100	100	100	100	100	100
64	100	100	100	100	100	100	100	100	100
128	93.75	100	87.5	100	100	100	100	100	98.8281
256	46.88	56.25	43.75	100	68.75	65.63	100	75	86.042
512	23.44	28.3	21.88	100	34.38	32.81	50	37.5	55.846
1024	11.72	14.06	10.94	56.25	17.19	16.41	25	18.75	31.1644



b)

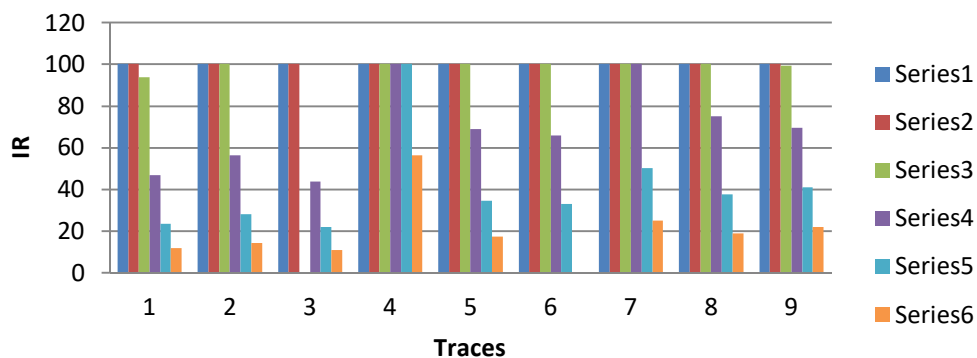
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
32	100	100	100	100	100	100	100	100	100
64	100	100	100	100	100	100	100	100	100
128	93.75	100	87.5	100	100	100	100	100	97.6563
256	46.88	56.25	43.75	100	68.75	65.63	100	75	69.5325
512	23.44	28.13	21.88	100	34.38	32.81	50	37.5	41.0175
1024	14.01	10.94	56.25	17.19	16.41	25	18.75	11.72	21.2838





c)

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
32	100	100	100	100	100	100	100	100	100
64	100	100	100	100	100	100	100	100	100
128	93.75	100	87,5	100	100	100	100	100	99.1071
256	46.88	56.25	43.75	100	68.75	65.63	100	75	69.5325
512	23.44	28.13	21.88	100	34.38	32.81	50	37.5	41.0175
1024	11.72	14.06	10.94	56.25	17.19	16,41	25	18.75	21.9871



Gradul de utilizare al cache-ului e de asteptat sa scada cu cresterea capacitatii, astfel mai multe blocuri necesare ar incapa in cache - insa imbunatatirea cu VC sau SVC nu aduce nicio imbunatarire legata de gradul de utilizare

4.Determinați influența dimensiunii victim cache-ului de instrucțiuni asupra ratei de procesare IR(Victim_size) și asupra ratei de miss în cache-ul de instrucțiuni RmissIC(Victim_size) în cele doua solutii:

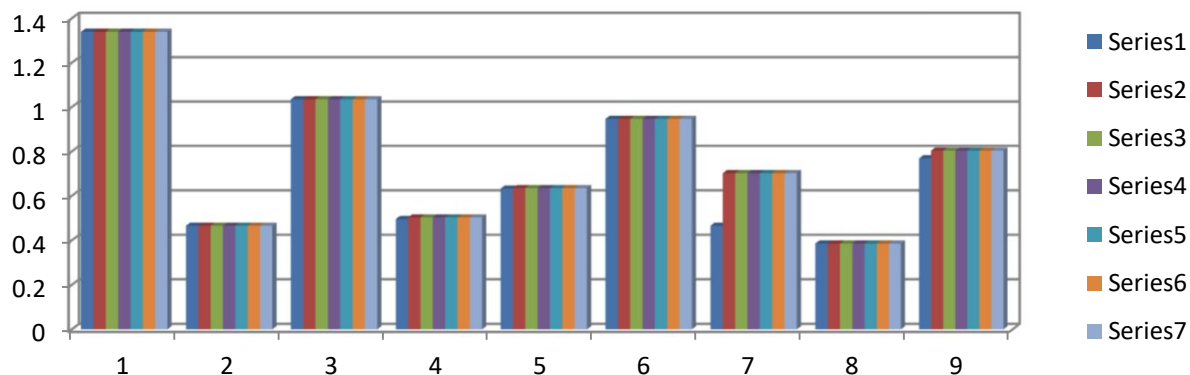
- cu victim cache simplu.
- cu selective victim cache.

a)

Victim_si IR (IPC)

ze

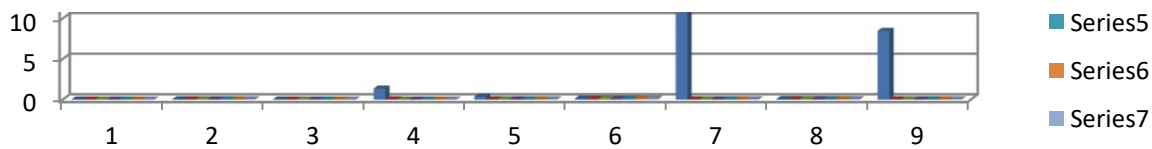
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
8	1.3396	0.4655	1.0352	0.4955	0.6324	0.9467	0.4656	0.3857	0.7686
32	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037
64	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037
128	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037
256	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037
512	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037
1024	1.3396	0.4655	1.0352	0.503	0.6346	0.9467	0.7016	0.3857	0.8037



Victim_sizeRmissIC (%)

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
8	0.03	0.04	0.02	1.36	0.37	0.14	32.13	0.06	8.47
32	0.03	0.04	0.02	0.04	0.05	0.12	0.05	0.06	0.05
64	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
128	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
256	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
512	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
1024	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05

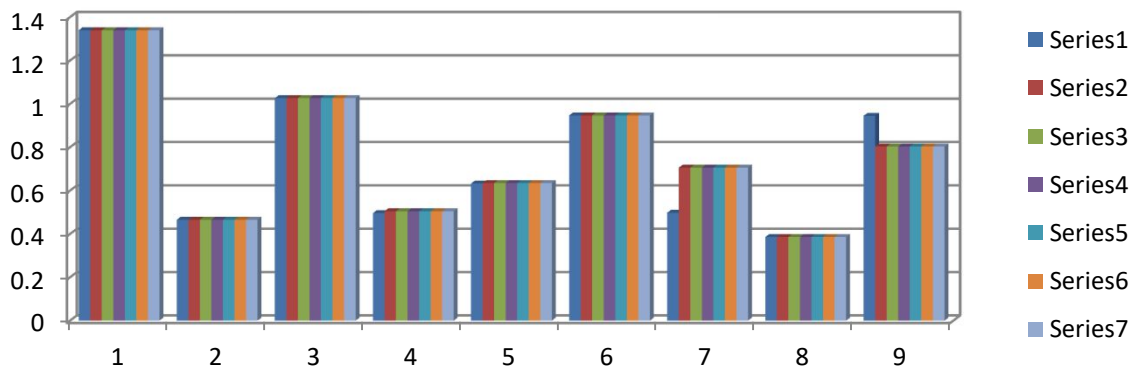




b)

Victim_siz
ze IR (IPC)

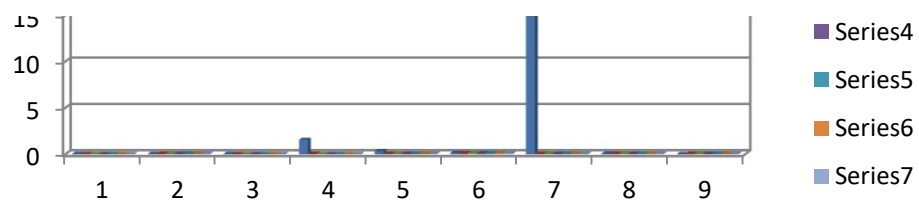
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
8	1.3396	0.4655	1.0267	0.4962	0.6327	0.9467	0.4979	0.3858	0.9446
32	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033
64	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033
128	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033
256	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033
512	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033
1024	1.3396	0.4655	1.0267	0.5044	0.6346	0.9467	0.7058	0.3858	0.8033



Victim_sizRmissIC (%)

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree	Media
8	0.03	0.03	0.02	1.54	0.33	0.14	29.46	0.07	8.474.51
32	0.03	0.04	0.02	0.04	0.05	0.12	0.05	0.06	0.05
64	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
128	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
256	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
512	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05
1024	0.03	0.04	0.02	0.03	0.05	0.12	0.05	0.06	0.05





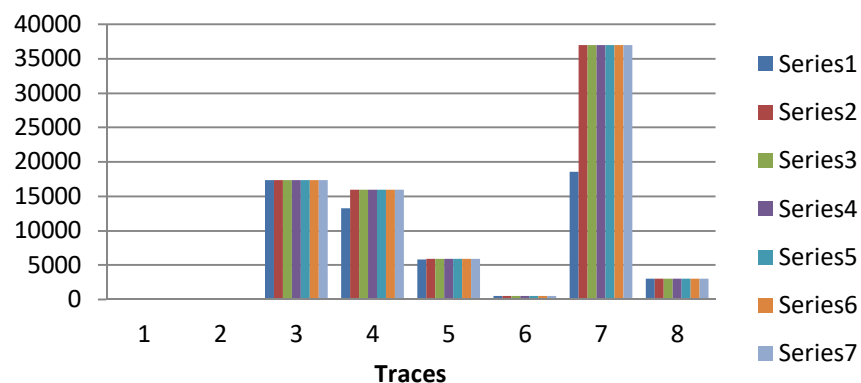
5.Studiați variația numărului de interschimbări produse între cache-ul principal și victim cache în cazul creșterii capacității celui din urmă, în ipostazele:

- cu victim cache simplu.
- cu selective victim cache.

a)

Victim_size Interchanges

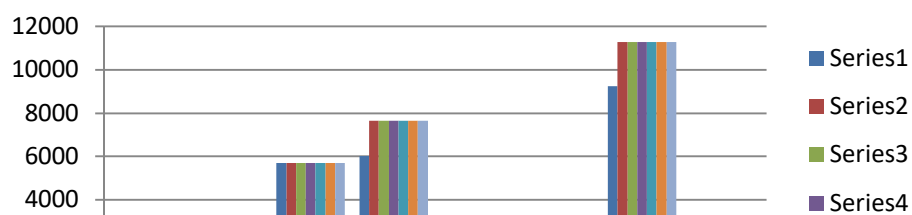
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
8	2	5	17321	13197	5746	459	18483	2992
32	2	5	17321	15923	5874	461	36907	2993
64	2	5	17321	15925	5874	461	36907	2993
128	2	5	17321	15925	5874	461	36907	2993
256	2	5	17321	15925	5874	461	36907	2993
512	2	5	17321	15925	5874	461	36907	2993
1024	2	5	17321	15925	5874	461	36907	2993

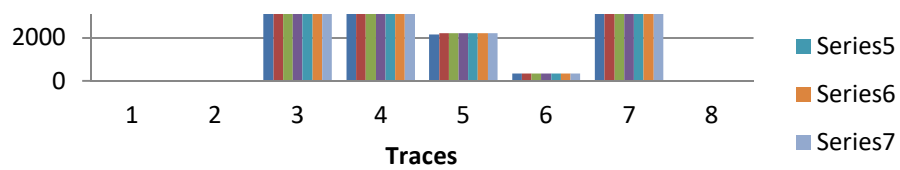


b)

Victim_size Interchanges

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
8	9	11	5682	6006	2137	340	9222	18
32	9	11	5682	7625	2209	340	11278	19
64	9	11	5682	7625	2209	340	11278	19
128	9	11	5682	7625	2209	340	11278	19
256	9	11	5682	7625	2209	340	11278	19
512	9	11	5682	7625	2209	340	11278	19
1024	9	11	5682	7625	2209	340	11278	19

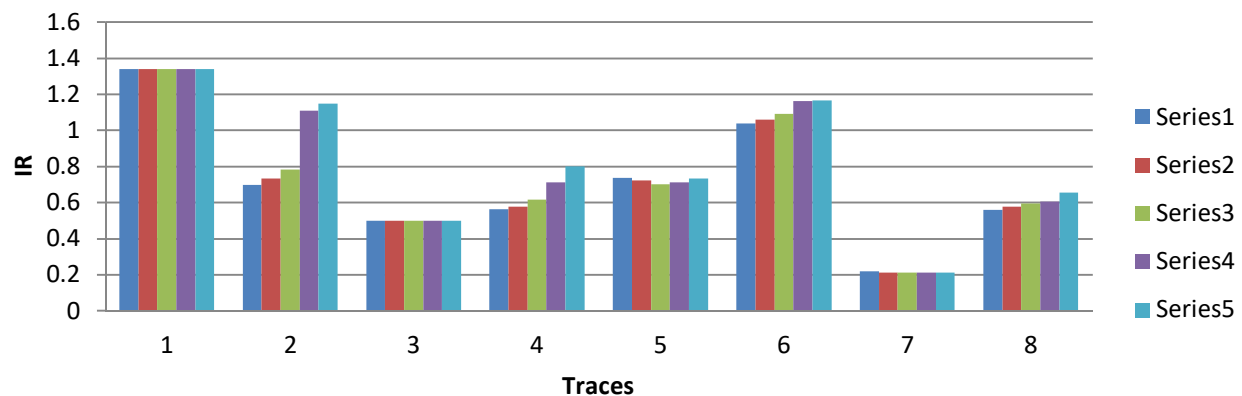




6. Pentru configurația optimă găsită determinați ce algoritm de înlocuire este mai bun în Victim Cache: LRU, Random sau FIFO.

ISSUE RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	1.3394	0.6949	0.4991	0.5606	0.7368	1.0372	0.217	0.5586
FR_size =64	1.3394	0.7329	0.4991	0.5766	0.7205	1.0577	0.2107	0.5767
FR_size =128	1.3394	0.7801	0.4991	0.6131	0.7007	1.0907	0.2107	0.5929
FR_size =256	1.3394	1.1072	0.4991	0.71	0.7105	1.1594	0.2107	0.6027
FR_size =1024	1.3394	1.1458	0.4991	0.8009	0.7324	1.1632	0.2108	0.6546

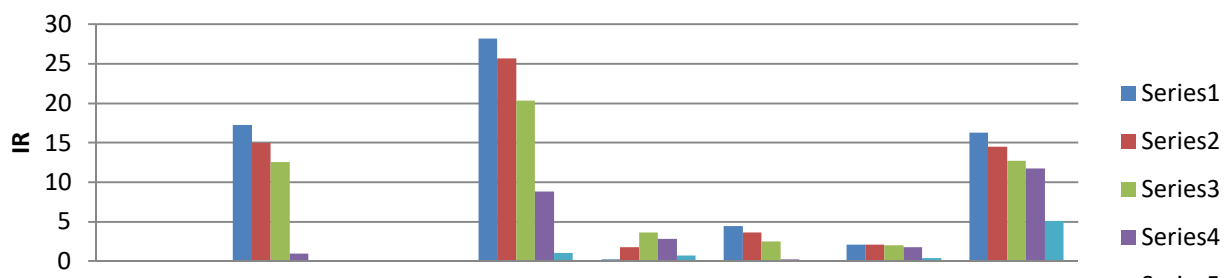


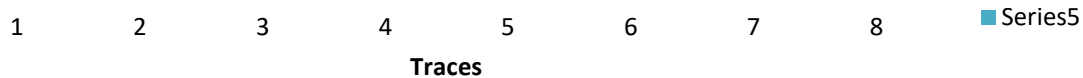
In cazul algoritmului de inlocuire LRU issue rate(IR) va creste pe masura ce FR creste.

MISS RATE

LRU

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	0	17.2	0	28.17	0.18	4.4	2.05	16.23
FR_size =64	0	14.96	0	25.68	1.76	3.61	2.04	14.45
FR_size =128	0	12.49	0	20.32	3.62	2.45	1.93	12.69
FR_size =256	0	0.95	0	8.78	2.79	0.18	1.69	11.67
FR_size =1024	0	0	0	1.02	0.66	0	0.31	5.02

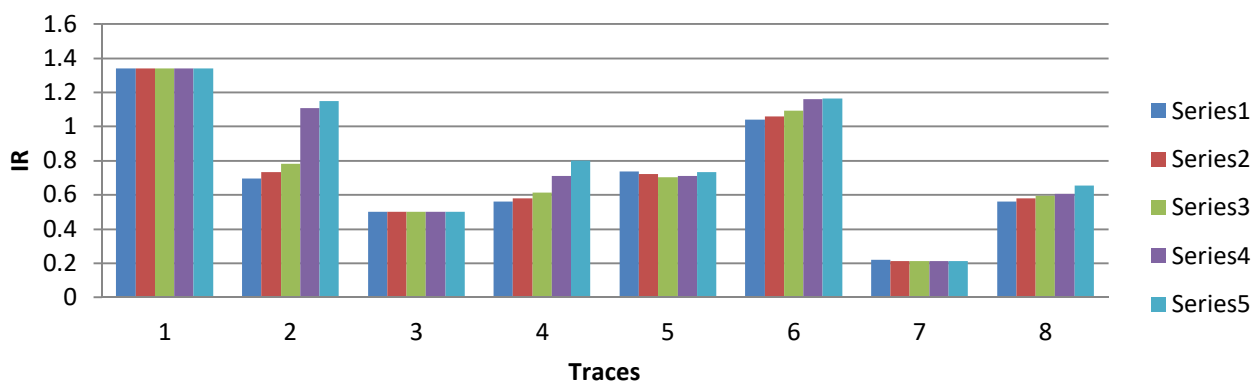




In cazul algoritmului LRU, rata de miss va creste daca Fr este mic.

ISSUE RATE

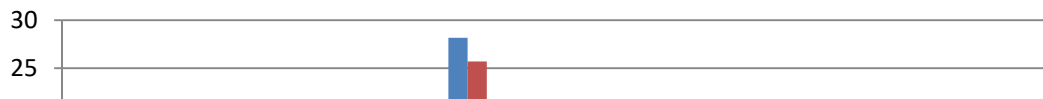
	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	1.3396	0.6949	0.4991	0.5606	0.7368	1.0372	0.217	0.5586
FR_size =64	1.3394	0.7329	0.4991	0.5766	0.7205	1.0577	0.2107	0.5767
FR_size =128	1.3394	0.7801	0.4991	0.6131	0.7007	1.0907	0.2107	0.5929
FR_size =256	1.3394	1.1072	0.4991	0.71	0.7105	1.1594	0.2107	0.6027
FR_size =1024	1.3394	1.1458	0.4991	0.8009	0.7324	1.1632	0.2108	0.6546

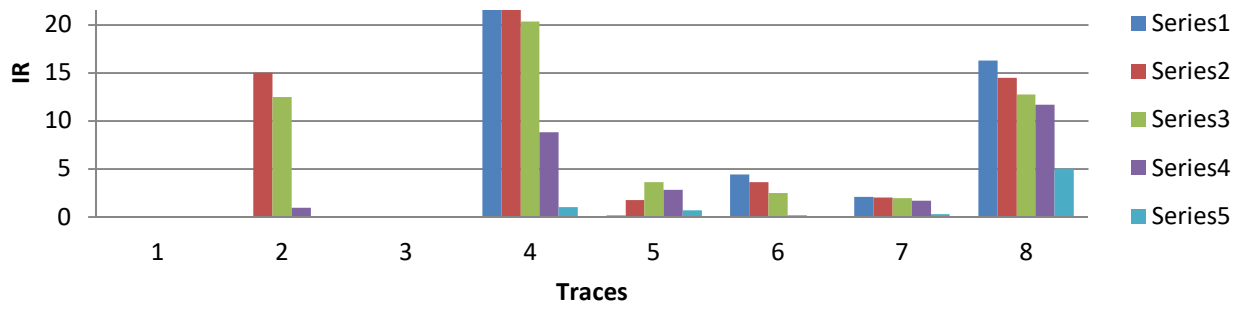


In cazul Random, issue rate(IR) va creste pe masura ce FR creste.

MISS RATE

	Bubble	Matrix	Perm	Puzzle	Queens	Sort	Tower	Tree
FR_size =32	0	0.11	0	28.17	0.18	4.4	2.05	16.23
FR_size =64	0	14.96	0	25.68	1.76	3.61	2.04	14.45
FR_size =128	0	12.49	0	20.32	3.62	2.45	1.93	12.69
FR_size =256	0	0.95	0	8.78	2.79	0.18	1.69	11.67
FR_size =1024	0	0	0	1.02	0.66	0	0.31	5.02

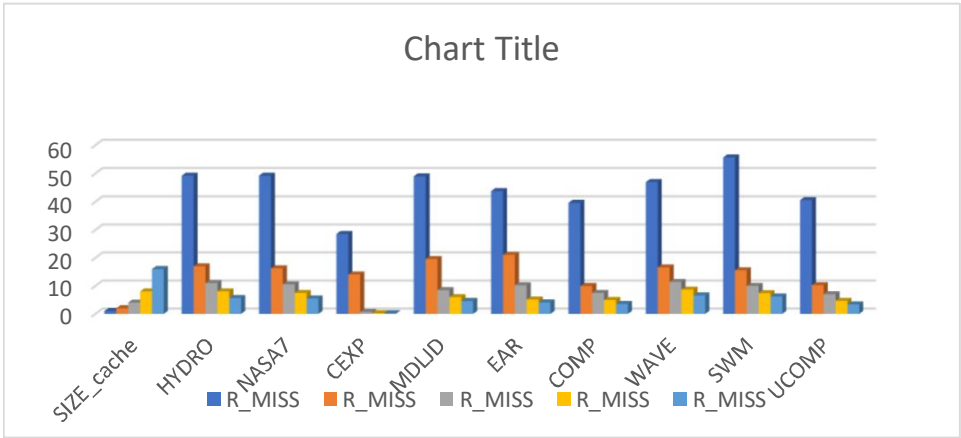




In cazul algoritmului Random, rata de miss va creste daca Fr este mic.

Studiați influența capacității cache-ului de instrucțiuni (în KB)
asupra ratei de miss în cache Rmiss(SIZE_Cache).

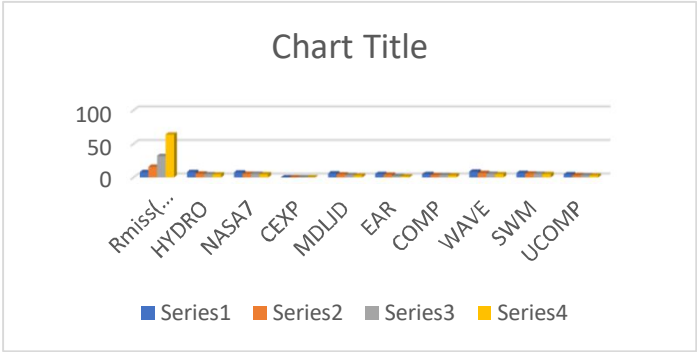
SIZE_cache	R_MISS 1	R_MISS 2	R_MISS 4	R_MISS 8	R_MISS 16
HYDRO	49.177	16.972	11.001	8.0395	5.6888
NASA7	49.218	16.28	10.566	7.4933	5.4987
CEXP	28.535	14.05	0.7	0.235	0.215
MDLJD	48.99	19.565	8.465	5.94	4.595
EAR	43.726	21.044	10.211	5.1432	4.107
COMP	39.58	9.945	7.449	4.9525	3.5658
WAVE	46.922	16.603	11.409	8.6373	6.6239
SWM	55.655	15.535	10	7.375	6.22
UCOMP	40.542	10.245	6.989	4.6469	3.4029



OBS.
Din ce crestem SIZE_cacheul scade rata de miss.

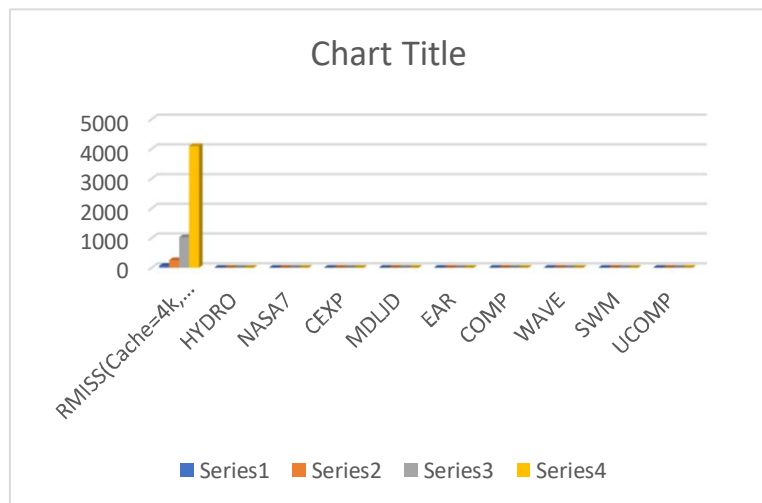
Pornind de la configurația inițială generați graficul Rmiss(BLOC_SIZE).

Rmiss(BLOC_SIZE)	HYDRO	NASA7	CEXP	MDLJD	EAR	COMP	WAVE	SWM	UCOMP
8	8.09	7.49	0.24	5.96	5.1	4.95	8.63	6.72	4.64
16	5.68	5.49	0.21	4.59	4.1	3.56	6.63	5.71	3.4
32	4.88	5.39	0.21	3.61	2.3	3.44	5.48	5.57	3.25
64	4.56	4.69	0.21	2.86	1.5	3.05	4.78	4.96	2.89



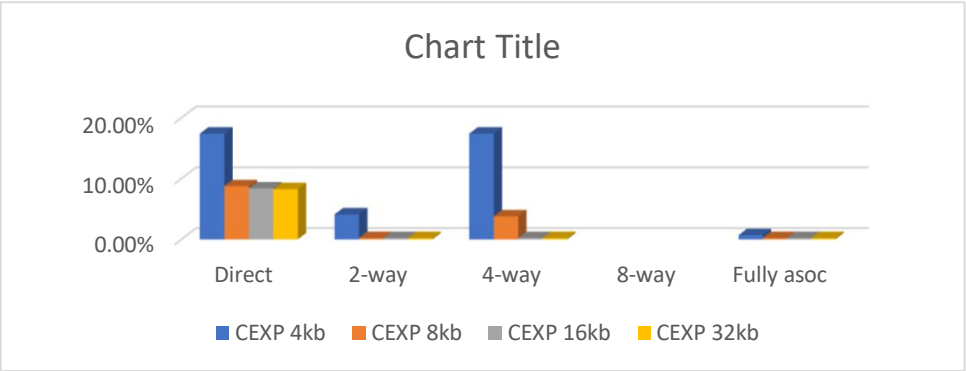
Determinați rata de miss variind dimensiunea blocului de date pentru diferite dimensiuni de cache.

RMISS(C	HYDRO	NASA7	CEXP	MDLJD	EAR	COMP	WAVE	SWM	UCOMP
64	25.15%	25.05%	1.31%	25.94%	26.92%	29.73%	29.73%	33.04%	22.24%
256	14.81%	13.36%	0.55%	11.88%	10.79%	12.00%	17.50%	11.48%	14.07%
1024	11.00%	10.56%	0.70%	8.46%	10.21%	7.44%	11.40%	10.00%	6.98%
4096	47.95%	47.87%	19.72%	42.81%	38.32%	38.98%	45.37%	54.14%	39.95%



Determinați rata de miss variind gradul de asociativitate pentru diferite dimensiuni de cache.

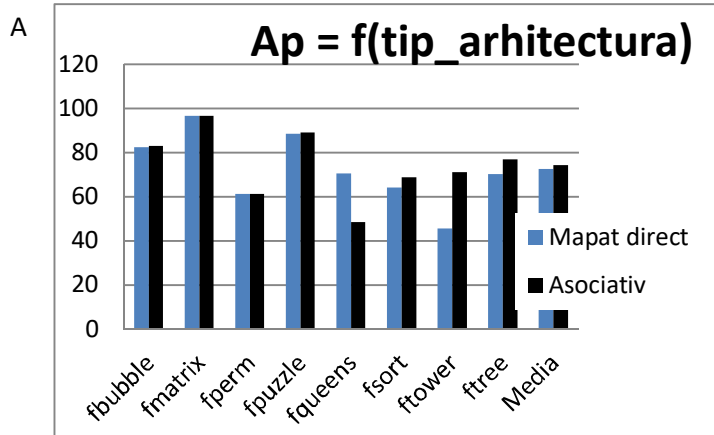
CEXP	miss rate	Direct	2-way	4-way	8-way	Fully asoc
	4kb	17.39%	4.11%	17.39%		0.70%
	8kb	8.76%	0.24%	3.79%		0.24%
	16kb	8.38%	0.22%	0.22%		0.22%
	32kb	8.26%	0.21%	0.21%		0.21%



Sa se reprezinte sub forma grafica functiile utilizând
implicit automatul de predictie pe doi biti:

a) $A_p=f(\text{tip_arhitectura})$

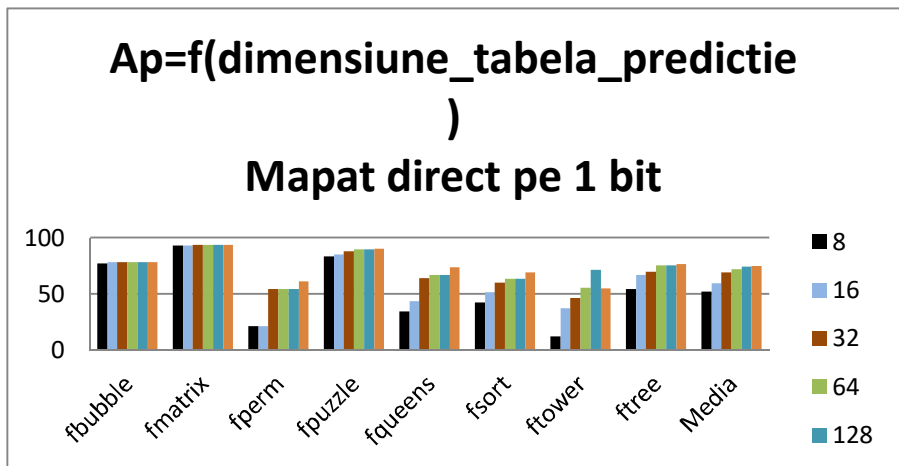
$f(\text{tip_arhitectura})$	<i>fbubble</i>	<i>fmatrix</i>	<i>fperm</i>	<i>fpuzzle</i>	<i>fqueens</i>	<i>fsort</i>	<i>ftower</i>	<i>ftree</i>	<i>Media</i>
Mapat direct	82.43	96.58	61.15	88.51	70.54	64.2	45.4	70.15	72.37
Asociativ	82.9	96.58	61.16	88.99	48.29	68.7	71	76.9	74.31



B b) $A_p=f(\text{dimensiune_tabela_predictie})$

Mapat direct pe 1 bit

<i>siune_tabela</i>	<i>fbubble</i>	<i>fmatrix</i>	<i>fperm</i>	<i>fpuzzle</i>	<i>fqueens</i>	<i>fsort</i>	<i>ftower</i>	<i>ftree</i>	<i>Media</i>
8	76.83	92.65	20.95	83.13	34.49	42.1	12.3	54.46	52.10
16	77.8	92.87	20.95	84.73	43.41	51.1	37.4	66.76	59.37
32	77.8	93.31	54.44	87.97	63.66	59.7	46.1	69.78	69.10
64	77.8	93.31	54.44	89.25	66.49	63.5	55	75.04	71.85
128	77.8	93.31	54.44	89.58	66.49	63.5	71	75.04	73.89
256	77.8	93.31	61.15	89.86	73.28	68.7	55	76.14	74.41

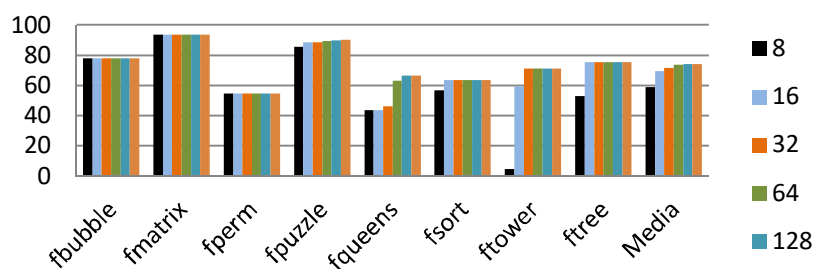


Complet asociativ pe 1 bit

siune_tabela	bubble	matrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	Media
8	77.8	93.31	54.44	85.26	43.38	56.3	4.28	52.89	58.46
16	77.8	93.31	54.44	88.4	43.39	63.5	59	75.04	69.36
32	77.8	93.31	54.44	88.12	46.05	63.5	71	75.04	71.15
64	77.8	93.31	54.44	89.31	62.98	63.5	71	75.04	73.42
128	77.8	93.31	54.44	89.73	66.38	63.5	71	75.04	73.89
256	77.8	93.31	54.44	89.99	66.46	63.5	71	75.04	73.94

**Ap=f(dimensiune_tabela_predict
ie)**

Complet Asociativ pe 1 bit



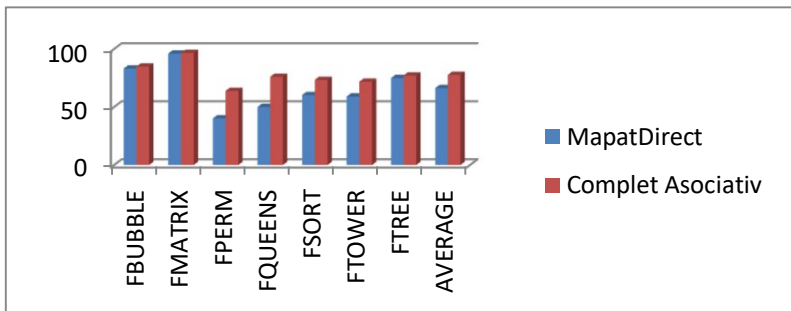
Să se reprezinte sub formă grafică funcțiile utilizând implicit automatul de predicție pe doi biți:

1. $Ap = f(\text{tip_arhitectură})$

Table = 256, HR = 4, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34271	20518	22023	19245	7600	22396	24644	
Raport	83.15	96.1436	40.174	50.0364	60.31	59.0456	74.94	66.2568

Table = 256, HR = 4, LRU = 2, Arhitectura = CompletAsociativ, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	2601	37930	32887	
Ap	34966	20613	35050	29257	9251	27269	25385	
Raport	84.836	96.589	63.938	76.067	73.410	71.893	77.190	77.704

	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
MapatDirect	83.149	96.143	40.174	50.036	60.31	59.045	74.94	66.2567
Complet Asociativ	84.836	96.588	63.938	76.067	73.41	71.893	77.19	77.7031



2. Analizați influența gradului de localizare al saltului asupra acurateții de predicție: $Ap = f(i)$ unde $i = \text{dim. PClow}$.

Table = 256, HR = 1, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34965	20613	35048	28508	9157	27265	25401	
Raport	84.83	96.59	63.93	74.12	72.67	71.88	77.24	77.32

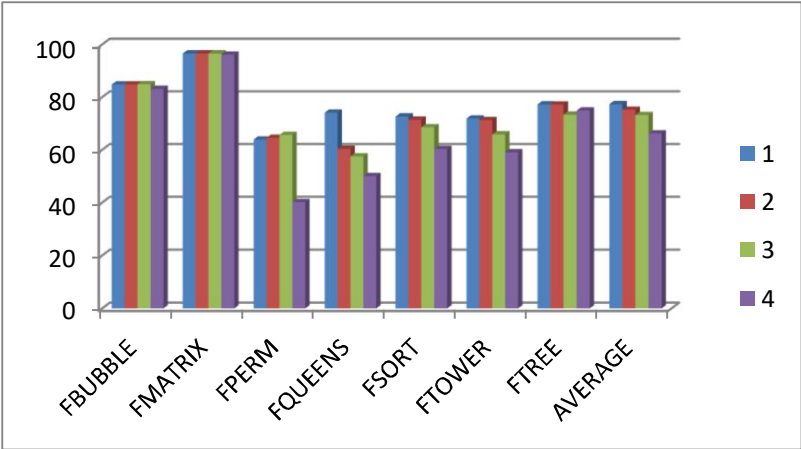
Table = 256, HR = 2, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34978	20613	35388	23224	9001	27037	25393	
Raport	84.865	96.588	64.554	60.381	71.430	71.281	77.212	75.187

Table = 256, HR = 3, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34992	20613	35994	22118	8642	24977	24120	
Raport	84.899	96.588	65.659	57.506	68.581	65.850	73.342	73.203

Table = 256, HR = 4, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE

Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34271	20518	22023	19245	7600	22396	24644	
Raport	83.149	96.143	40.174	50.036	60.312	59.045	74.935	66.256

	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
1	84.833	96.588	63.934	74.119	72.668	71.882	77.237	77.323
2	84.865	96.588	64.554	60.381	71.430	71.281	77.212	75.187
3	84.899	96.588	65.659	57.506	68.581	65.850	73.342	73.203
4	83.149	96.143	40.174	50.036	60.312	59.045	74.935	66.256



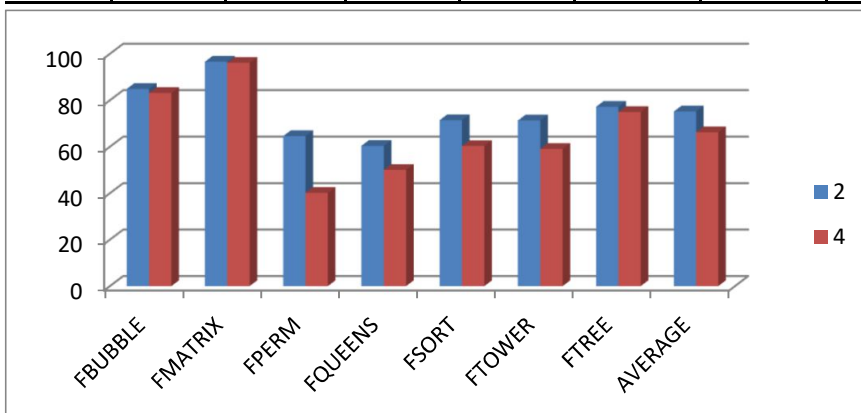
3. Stabiliți influența contextului în care se situează saltul în program:

$Ap = f(HR_{global})$.

Table = 256, HR = 2, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216.00	21341.00	54819.00	38462.00	12601.00	37930.00	32887.00	
Ap	34978.00	20613.00	35388.00	23224.00	9001.00	27037.00	25393.00	
Raport	84.87	96.59	64.55	60.38	71.43	71.28	77.21	75.19

Table = 256, HR = 4, LRU = 0, Arhitectura = MapatDirect, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216.00	21341.00	54819.00	38462.00	12601.00	37930.00	32887.00	
Ap	34271.00	20518.00	22023.00	19245.00	7600.00	22396.00	24644.00	
Raport	83.15	96.14	40.17	50.04	60.31	59.05	74.94	66.26

	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
2	84.87	96.59	64.55	60.38	71.43	71.28	77.21	75.19
4	83.15	96.14	40.17	50.04	60.31	59.05	74.94	66.26

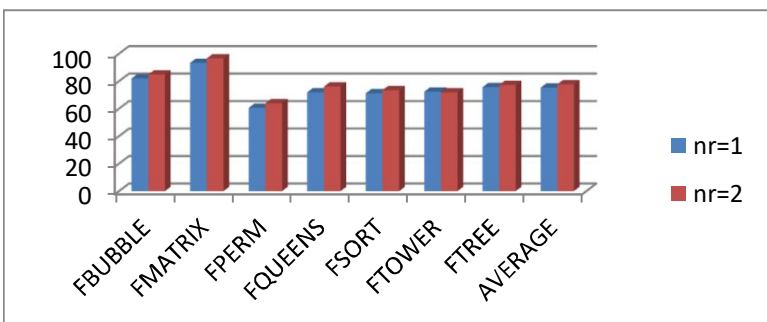


4. Reprezentați $Ap = f(nr_biți_automat_predicție)$ considerând parametrii optimi (PClow, HRglobal) rezultați în urma simulării efectuate la 1), 2) și 3).

Table = 256, LRU = 2, Arhitectura = CompletAsociativ, Automat = 1 bit								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	33798	19915	33209	27659	8966	27437	24864	
Raport	82.0021	93.3180	60.5794	71.9125	71.1531	72.3359	75.6043	75.2722

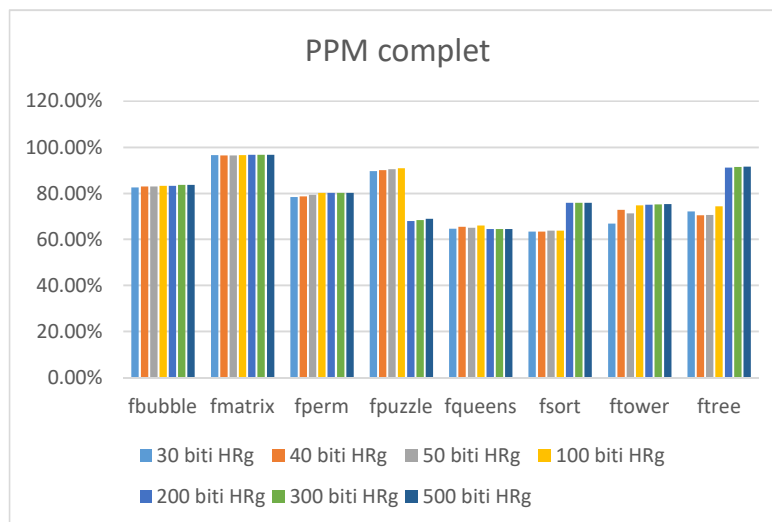
Table = 256, LRU = 2, Arhitectura = CompletAsociativ, Automat = 2 biti								
	FBUBBLE	FMATRIX	FPERM	FQUEENS	FSORT	FTOWER	FTREE	AVERAGE
Instr. Proc.	41216	21341	54819	38462	12601	37930	32887	
Ap	34966	20613	35050	29257	9251	27269	25385	
Raport	84.836	96.5887	63.9377	76.0673	73.4148	71.893	#####	77.7037

	nr=1	nr=2
FBUBBLE	82.0021	84.836
FMATRIX	93.318	96.5887
FPERM	60.5794	63.9377
FQUEENS	71.9125	76.0673
FSORT	71.1531	73.4148
FTOWER	72.3359	71.893
FTREE	75.6043	77.1886
AVERAGE	75.2722	77.7037

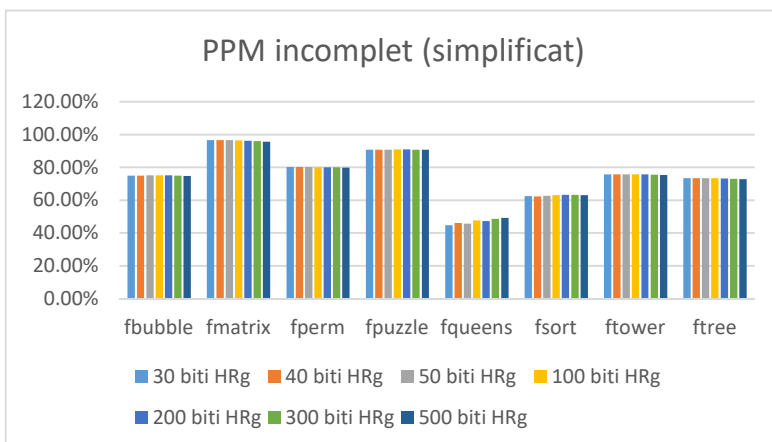


Sa se determine acuratetea de predictie pentru fiecare din cele doua tipuri de predictoare markov ppm complet sau simplificat in functie de nr de biti de istorie globala.

PPM comp	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	average
30 biti HRg	82.56%	96.56%	78.38%	89.67%	64.66%	63.36%	66.88%	72.21%	76.79%
40 biti HRg	83.04%	96.45%	78.68%	90.06%	65.45%	63.43%	72.89%	70.49%	77.56%
50 biti HRg	82.93%	96.45%	79.38%	90.46%	65.01%	63.75%	71.34%	70.68%	77.50%
100 biti HR	83.21%	96.66%	80.17%	90.89%	66.10%	63.80%	74.74%	74.43%	78.75%
200 biti HR	83.21%	96.68%	80.17%	68.04%	64.55%	75.85%	75.01%	91.15%	79.33%
300 biti HR	83.66%	96.69%	80.17%	68.44%	64.52%	75.85%	75.17%	91.46%	79.50%
500 biti HR	83.69%	96.69%	80.17%	68.96%	64.55%	75.85%	75.32%	91.61%	79.61%



PPM simpl	fbubble	fmatrix	fperm	fpuzzle	fqueens	fsort	ftower	ftree	average
30 biti HRg	74.94%	96.63%	80.16%	90.70%	44.84%	62.58%	75.82%	73.34%	74.88%
40 biti HRg	74.99%	96.60%	80.16%	90.74%	46.19%	62.33%	75.81%	73.42%	75.03%
50 biti HRg	75.19%	96.58%	80.15%	90.77%	45.76%	62.62%	75.81%	73.40%	75.04%
100 biti HR	75.24%	96.46%	80.12%	90.88%	47.67%	63.12%	75.76%	73.33%	75.32%
200 biti HR	75.15%	96.23%	80.05%	90.86%	47.23%	63.29%	75.69%	73.18%	75.21%
300 biti HR	75.03%	96.00%	79.98%	90.84%	48.60%	63.33%	75.58%	73.03%	75.30%
500 biti HR	74.79%	95.54%	79.84%	90.79%	49.32%	62.99%	75.40%	72.73%	75.18%



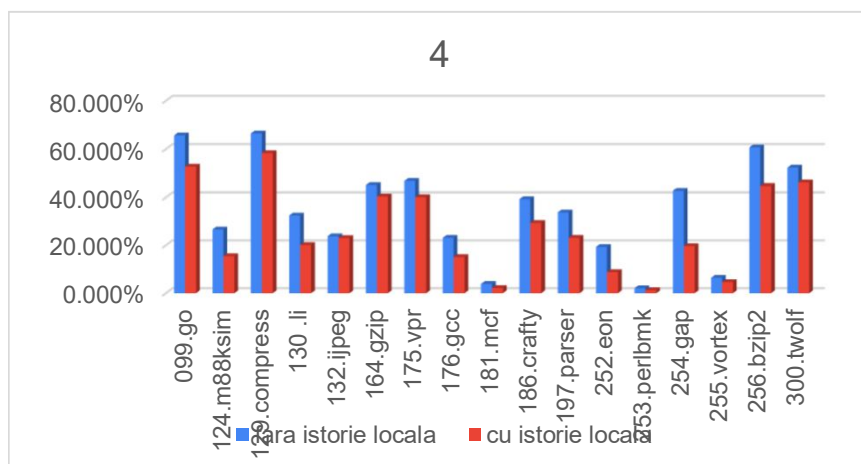
Contextele dificil de prezis[%]:

a)fara istorie locala,istoria globala sa varieze intre 4 si 28;

b)cu istorie locala,istoria globala se variaza intre 4 si 28;

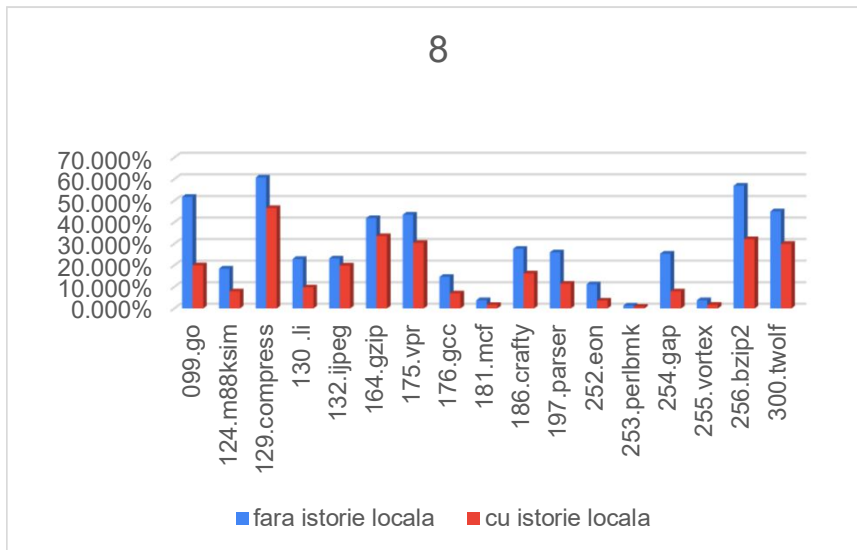
Realizati o comparatie intre cele doua.

SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	65.688%	52.775%	4
124.m88ksim	26.621%	15.497%	
129.compress	66.501%	58.366%	
130 .li	32.455%	20.157%	
132.jpeg	23.79%	23.015%	
164.gzip	45.14%	40.39%	
175.vpr	46.827%	40.048%	
176.gcc	23.103%	15.142%	
181.mcf	3.871%	2.199%	
186.crafty	39.249%	29.338%	
197.parser	33.728%	23.144%	
252.eon	19.329%	8.827%	
253.perlbmk	2.049%	1.282%	
254.gap	42.739%	19.625%	
255.vortex	6.44%	4.588%	
256.bzip2	60.683%	44.742%	
300.twolf	52.343%	46.243%	



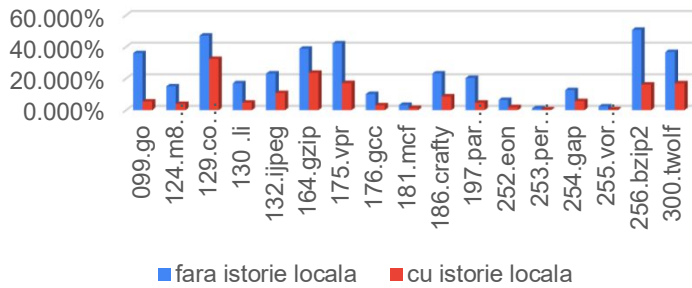
SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	51.784%	20.008%	8
124.m88ksim	18.475%	7.947%	
129.compress	60.792%	46.588%	
130 .li	22.945%	9.785%	
132.jpeg	23.221%	19.936%	
164.gzip	41.881%	33.598%	
175.vpr	43.496%	30.423%	
176.gcc	14.669%	7.036%	
181.mcf	3.855%	1.658%	
186.crafty	27.674%	16.253%	
197.parser	26.036%	11.515%	

252.eon	11.236%	3.684%
253.perlbnk	1.360%	0.834%
254.gap	25.411%	7.958%
255.vortex	3.839%	1.693%
256.bzip2	56.954%	32.138%
300.twolf	45.061%	29.985%



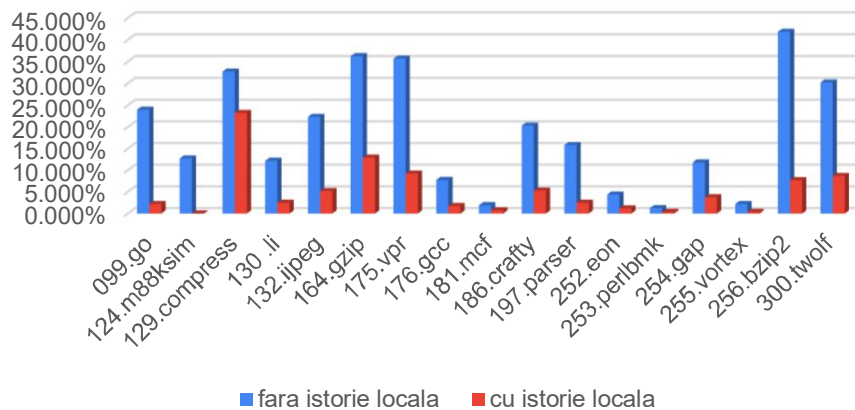
SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	36.073%	5.546%	12
124.m88ksim	15.098%	3.973%	
129.compress	47.188%	32.484%	
130.li	17.108%	4.804%	
132.jpeg	23.212%	10.866%	
164.gzip	38.907%	23.649%	
175.vpr	42.289%	17.183%	
176.gcc	10.245%	3.110%	
181.mcf	3.341%	1.394%	
186.crafty	23.245%	8.718%	
197.parser	20.395%	4.771%	
252.eon	6.569%	1.931%	
253.perlbnk	1.219%	0.531%	
254.gap	12.770%	5.746%	
255.vortex	2.395%	0.590%	
256.bzip2	50.813%	16.247%	
300.twolf	36.823%	17.021%	

12

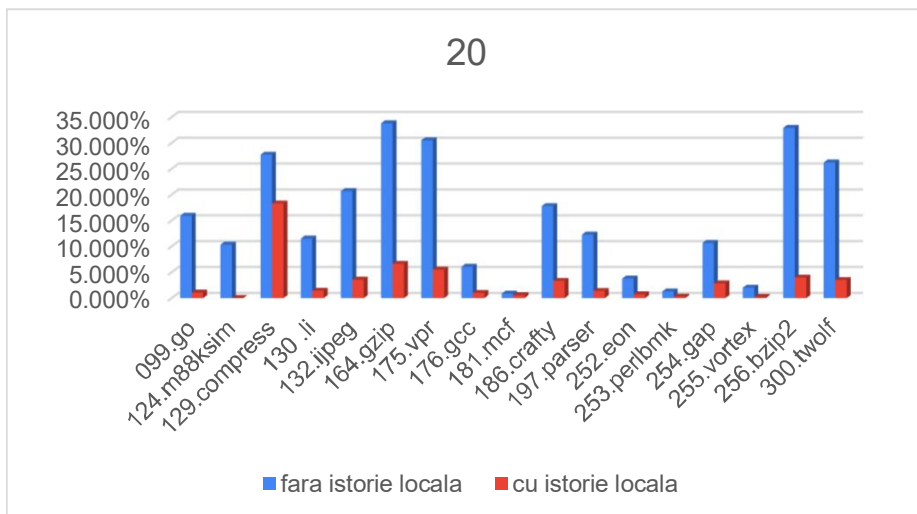


SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	23.965%	2.180%	16
124.m88ksim	12.714%	0.000%	
129.compress	32.693%	23.233%	
130 .li	12.171%	2.469%	
132.ijpeg	22.332%	5.180%	
164.gzip	36.244%	12.893%	
175.vpr	35.662%	9.227%	
176.gcc	7.737%	1.711%	
181.mcf	1.906%	0.718%	
186.crafty	20.303%	5.286%	
197.parser	15.802%	2.472%	
252.eon	4.368%	1.182%	
253.perlbmk	1.235%	0.341%	
254.gap	11.785%	3.731%	
255.vortex	2.192%	0.362%	
256.bzip2	41.803%	7.672%	
300.twolf	30.205%	8.674%	

16

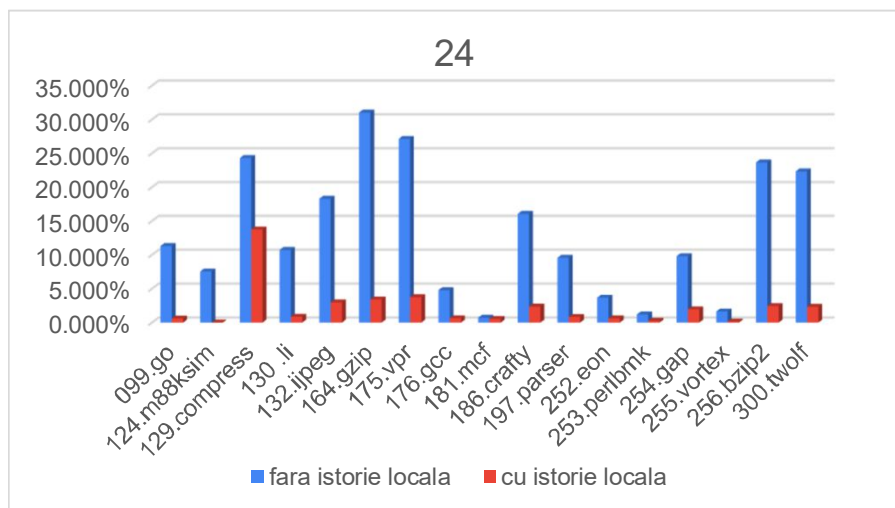


SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	15.934%	1.065%	20
124.m88ksim	10.330%	0.000%	
129.compress	27.733%	18.295%	
130 .li	11.504%	1.429%	
132.ijpeg	20.682%	3.558%	
164.gzip	33.799%	6.635%	
175.vpr	30.505%	5.508%	
176.gcc	6.064%	1.012%	
181.mcf	0.853%	0.550%	
186.crafty	17.793%	3.337%	
197.parser	12.277%	1.380%	
252.eon	3.777%	0.720%	
253.perlbmk	1.286%	0.292%	
254.gap	10.668%	2.849%	
255.vortex	2.020%	0.232%	
256.bzip2	32.913%	3.970%	
300.twolf	26.224%	3.499%	

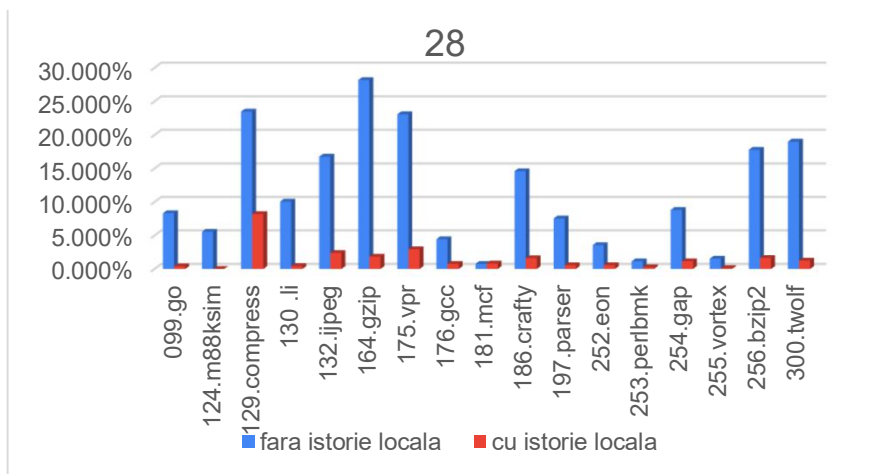


SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	11.307%	0.605%	24
124.m88ksim	7.549%	0.000%	
129.compress	24.312%	13.749%	
130 .li	10.737%	0.861%	
132.ijpeg	18.296%	2.975%	
164.gzip	31.025%	3.410%	
175.vpr	27.134%	3.744%	
176.gcc	4.791%	0.640%	
181.mcf	0.709%	0.533%	

186.crafty	16.060%	2.369%
197.parser	9.577%	0.826%
252.eon	3.678%	0.619%
253.perlbmk	1.201%	0.260%
254.gap	9.807%	1.959%
255.vortex	1.638%	0.155%
256.bzip2	23.657%	2.438%
300.twolf	22.327%	2.332%



SPEC2000	fara istorie locala	cu istorie locala	istorie globala
099.go	8.315%	0.383%	28
124.m8ksim	5.562%	0.000%	
129.compress	23.486%	8.204%	
130.li	10.062%	0.424%	
132.jpeg	16.756%	2.383%	
164.gzip	28.172%	1.831%	
175.vpr	23.071%	2.940%	
176.gcc	4.441%	0.757%	
181.mcf	0.743%	0.829%	
186.crafty	14.599%	1.607%	
197.parser	7.523%	0.550%	
252.eon	3.557%	0.558%	
253.perlbmk	1.149%	0.267%	
254.gap	8.821%	1.146%	
255.vortex	1.540%	0.123%	
256.bzip2	17.794%	1.626%	
300.twolf	19.003%	1.242%	



Observatii:

- Cu cat creste istoria globala, cu atat scad procentele in ambele cazuri
- Procentele cu istorie locala sunt mai mici decat cele fara istorie locala