

## Timpi de executie pentru v1 (S primeste cate 20 de linii)

p_r	p_w	dt	dx	Timp	
				Server	Avg(Client)
2	2	1	1	49073.58	48821.616
			2	97698.92	97416.314
		2	1	49177.66	48872.666
			2	97525.44	97211.266
		4	1	49316.69	49073.428
			2	97549.94	97252.824
4	2	1	1	49598.46	49362.78
			2	97735.18	97447.424
		2	1	49571.47	49531.70
			2	97162.82	96999.226
		4	1	49553.95	49234.07
			2	97005.44	96872.4556
	4	1	1	50180.63	49814.356
			2	97649.26	97444.436
		2	1	50330.41	50091.604
			2	97675.29	97451.216
		4	1	49958.87	49750.826
			2	97444.84	97052.074
	8	1	1	48848.68	48724.994
			2	96958.32	96837.602
		2	1	48858.64	48753.374
			2	97080.57	96932.032
		4	1	48859.71	48712.408
			2	97151.32	97007.676

## Concluzii

- Evident, timpul pe server > timpul pe client (include timpul accept si clear resources)
- Variatia lui  $dx$  care simlueaza delay-ul user input-ului afecteaza linear timpul de executie ( $dx = 2$  inseamna dublarea timpului de executie)
- Unele combinatii de parametric pe server sunt relativ mai eficiente (ex.  $p_r = 4, p_w = 8$ ), aducand mici imbunaratiri sistemului. Cresterea numarului de workers pare sa aiba cel mai puternic efect.
- Cele mai slabe performante par sa se obtina cand  $p_r = p_w$
- Parametrul  $dt$  nu influenteaza performanta overall, din cauza parametrului  $dx$  care este ordini de magnitudine mai mari

Timpi de executie pentru v2 (S primește câte 128 bytes, reface fisierele)

p_r	p_w	dt	timp
2	2	1	3614.21
		2	3334.5
		4	2593.08
4	2	1	3329.77
		2	3814.78
		4	2418.26
	4	1	3289.82
		2	2565.38
		4	3262.82
	8	1	2480.1
		2	3277.23
		4	1948.55

### Concluzii

- În cazul  $dt = 4$  se observă o îmbunătățire cu ~40% a timpului de execuție (cache-uirea clasamentului pe țări la anumite intervale de timp ajută la economisirea resurselor procesor)
- Absența parametrului  $dx$  (pentru că datele se trimit acum automat în chunk-uri și este treaba protocolului TCP, nu a userului) aduce îmbunătățiri drastice spre deosebire de varianta 1.
- **De aceea este important să facem comunicarea cu serverul doar când este necesar, nu pe tot parcursul rularii clientului. Este mai ok ca datele să fie fragmentate fizic (pe bytes) și recombine pe server, decât logic ("calupuri" de câte 20 de înregistrări).**