

Rezultate:

$\Delta x$ (sec)	Send files	$\Delta t$ (ms)	pr	pw	Results
1	yes	1	4	4	20.19
			2	2	29.37
			4	2	20.24
			4	8	20.14
		2	4	4	20.19
			2	2	29.60
			4	2	20.14
			4	8	20.12
		4	4	4	20.15
			2	2	28.16
			4	2	20.02

			4	8	20.19
	no	1	4	4	84.99
			2	2	84.10
			4	2	84.20
			4	8	84.11
		2	4	4	85.32
			2	2	86.54
			4	2	83.45
			4	8	82.93
		4	4	4	85.76
			2	2	84.56
			4	2	84.95

			4	8	84.54
--	--	--	---	---	-------

$\Delta x$ (sec)	Send files	$\Delta t$ (ms)	pr	pw	Results
2	yes	1	4	4	30.29
			2	2	39.47
			4	2	33.74
			4	8	37.23
		2	4	4	30.39
			2	2	39.34
			4	2	30.24
			4	8	31.62
		4	4	4	32.65

			2	2	38.97
			4	2	31.56
			4	8	30.34
	no	1	4	4	96.59
			2	2	94.27
			4	2	94.36
			4	8	93.47
		2	4	4	94.43
			2	2	94.56
			4	2	93.48
			4	8	95.67
		4	4	4	98.23

			2	2	97.32
			4	2	98.45
			4	8	97.36

Analiza rezultate:

Optiunea de a trimite bucati din fisier este mult mai eficienta decat cea in care citim si apoi trimitem calupuri de cate 20 de perechi. In urma rularii consideram ca "munca" executata de reader este la fel de costisitoare ca "munca" executata de writer, asa credem ca se explica faptul ca indiferent ca avem multi reader sau writeri timpul este aproximativ egal.

Explicarea solutiei:

Folosim sincronizarea prin socketuri, impartind solutia in 2 directii: cea prin care citim rezultatele asociate clientului si apoi trimitem calupuri de cate 20 de perechi la intervalul de timp setat (tot trimis ca argument) si cea prin care trimitem calupuri din fisiere. Clientul se ocupa de partea de citire si trimitere sincronizata a datelor si apoi a cererii unui clasament partial ca in final sa primeasca clasamentul final.

Serverul se ocupa de partea de primire a datelor initiale si de prelucrare a lor. Prelucrarea lor se face exclusiv de o clasa numita RankingManager (incapsuleaza practic ce am facut in laboratorul trecut cu coada) unde se face adaugarea fiecarui scor in coada, crearea unui clasament cacheuit si in final adaugarea lui in lista finala care va fi sortata pentru a putea fi trimis fiecarui client la sfarsit. Serverul va primi ca argument la pornire si numarul de clienti care se vor conecta, prin urmare isi va da seama de terminarea tuturor clientul cu ajutorul unor variabile care contorizeaza numarul de producatori care au terminat, respectiv numarul de clienti care s-au terminat. In momentul in care executia tuturor clientilor s-a terminat atunci toti ceilalti vor primi o notificare si se va trimite rankingul sortat. Serverul poate sa handleuiasca maxim 3 clienti deodata si thread poolul este asociat readerilor.

( $\Delta x=1$ , 2 sec

Testare:

A)

p\_r = 4

$$p_w=4$$

$$\Delta t = 1\text{ms};$$

$$\Delta t = 2\text{ms};$$

$$\Delta t = 4\text{ms}$$

B)

$$p_r = 2$$

$$p_w=2$$

$$\Delta t = 1\text{ms};$$

$$\Delta t = 2\text{ms};$$

$$\Delta t = 4\text{ms}$$

C)

$$p_r = 4$$

$$p_w=2$$

$$\Delta t = 1\text{ms};$$

$$\Delta t = 2\text{ms};$$

$$\Delta t = 4\text{ms}$$

D)

$$p_r = 4$$

$$p_w=8$$

$$\Delta t = 1\text{ms};$$

$$\Delta t = 2\text{ms};$$

$$\Delta t = 4\text{ms})$$