|  |  |  |  |
| --- | --- | --- | --- |
| **Varianta** | **Test** | **Nr threads** | **Timp executie** |
| **0** | **1** | **1** | **7154045** |
| **2** | **1** | **5169785** |
| **3** | **1** | **8962950** |
| **4** | **1** | **8179295** |
| **1** | **1** | **4** | **11696563** |
| **8** | **19657257** |
| **16** | **23856759** |
| **2** | **4** | **10771520** |
| **8** | **18677066** |
| **16** | **18562881** |
| **3** | **4** | **23755013** |
| **8** | **35394841** |
| **16** | **46911661** |
| **4** | **4** | **15344970** |
| **2** | **1** | **4** | **3535320** |
| **8** | **8337139** |
| **16** | **8849296** |
| **2** | **4** | **4722791** |
| **8** | **9090098** |
| **16** | **10671008** |
| **3** | **4** | **17158146** |
| **8** | **29225091** |
| **16** | **34588285** |
| **4** | **4** | **4569803** |
| **3** | **1** | **4** | **9296029** |
| **8** | **9238609** |
| **16** | **11529119** |
| **2** | **4** | **8278979** |
| **8** | **10703526** |
| **16** | **11367895** |
| **3** | **4** | **20084278** |
| **8** | **23185641** |
| **16** | **27149436** |
| **4** | **4** | **8609930** |

Pentru a calcula rezultatul intr-un proces, nu putem avea rezultatul final fara a avea si carry-ul. Deci, putem fie sa asteptam dupa el (care teoretic este costisitor ca timp), fie sa facem adunarea fara carry, si cand il primim sa reluam adunarea si s-o continuam cat timp mai avem carry la pasul curent (sau sa ne oprim daca carry-ul primit este 0). A doua abordare pentru prima varianta ar trebui sa fie mai rapida in medie, dar deoarece facem citirea block cu block si cream procesele in acelasi timp cu citirea, nu se observa vreo diferenta semnificativa de timp intre cele doua abordari, deoarece carry-ul este primit de la procesul anterior de obicei aproape imediat de la crearea procesului.

In a treia varianta cu transmiterea asincrona, timpii sunt mai mici fata de prima varianta (folosind ambele abordari cu asteptarea carry-ului/reluarea adunarii cand primim carry). Functiile de transmitere devin asincrone, si trebuie sa ne asiguram ca am primit/trimis datele la un moment dat folosind functiile wait.

Cu toate acestea, a doua varianta folosind gather/scatter este cea mai rapida pentru datele de test date. Acest lucru se datoreaza in mare parte IO-ului, deoarece o citire mai lunga a datelor de intrare este mult mai rapida decat mai multe citiri mai mici, cum se intampla la celelalte doua variante.

Deoarece datele de intrare sunt destul de mici, folosirea mai multor procese ce implica transmiterea de informatii intre ele are un overhead semnificativ din timpul de rulare, si rularea cu mai multe procese este mai inceata.