

DBscan clustering algorithm - CSI 2520

Kanjanokphat Kitisuwanakul
300170040

Device Specifications:

Processor : Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz
Installed RAM : 6.00 GB (5.87 GB usable)
System type : 64-bit operating system, x64-based processor
Number of cores : 2
Logical processors : 4

Outputs: (eps=0.0003 et minpts=50)

Version non-concurrente N=4 : Execution time: 37.3728125s of 206102 points
N=2 et Fils = 4 : Execution time: 52.508719s of 198194 points
N=4 et Fils = 4 : Execution time: 20.3246042s of 206102 points
N=4 et Fils = 10 : Execution time: 21.9311678s of 206102 points
N=4 et Fils = 16 : Execution time: 26.4389978s of 206102 points
N=10 et Fils = 4 : Execution time: 5.8611958s of 222488 points
N=10 et Fils = 10 : Execution time: 5.7799414s of 222488 points
N=10 et Fils = 50 : Execution time: 6.0603858s of 222488 points
N=20 et Fils = 10 : Execution time: 3.838445s of 255300 points
N=20 et Fils = 50 : Execution time: 4.2423186s of 255300 points
N=20 et Fils = 200 : Execution time: 4.1169782s of 255300 points
N=20 et Fils = 500 : Execution time: 3.879108s of 255300 points

Observations et conclusions:

On observe que la version concurrente est plus rapide que la version non-concurrente pour le même nombre de N, et que la performance dépend beaucoup de N et du nombre de fils.

Si on a peu (2-4) de partitions, l'algorithme DBscan devrait s'exécuter sur un grand nombre de points pour chaque fil d'exécution. Donc, cela s'exécute lentement. (environ 20s - 1 min)

Avec N plus grand (10-20), on peut bénéficier plus de la version concurrente car chaque fil concurrent exécute l'algorithme sur un petit nombre de points. Comme il y a plusieurs partitions, les fils qui s'exécutent en même temps produisent la sortie plus rapidement. (environ 5s)

Quand on a plus de fils d'exécution, le programme s'exécute plus rapidement mais on remarque aussi que pour N=20, le temps d'exécution pour Fils = 50 est un légèrement plus élevé que Fils = 10. Il faut donc ne pas utiliser trop de fils d'exécution car cela peut aussi bien ralentir l'exécution (N=20 et T = 10 donne 3.83s mais T = 50 donne 4.24s pour le même nombre de N).