

Alocarea eficientă a task-urilor într-un sistem distribuit

Cristian-Ilie Vasluianu

Rezumat

Aplicația dezvoltată are drept scop eficientizarea execuției task-urilor într-un sistem distribuit, prin utilizarea resurselor într-un mod cât mai eficient și o distribuție cât mai optimă a task-urilor utilizând diferiți algoritmi și strategii de mapare. Prin sistem distribuit se înțelege un sistem ale cărui componente se află pe mai multe calculatoare interconectate într-o rețea, care comunică și se coordonează prin mesaje.

Apariția sistemelor de calcul tot mai performante cu resurse hardware impresionante impune nevoia de dezvoltare a aplicațiilor care să utilizeze aceste cerințe hardware la maximum, prin rularea rapidă și fluentă a task-urilor. Această rulare rapidă și cu timpi de calcul foarte reduși a unui număr foarte mare de task-uri se poate pune în evidență cu ajutorul sistemelor distribuite, prin împărțirea task-urilor de prelucrat în subtask-uri care vor fi preluate și prelucrate simultan de componente ale sistemului numite workeri. Aplicația proprie este dezvoltată utilizând limbajul Java și folosește această arhitectură de sistem distribuit pentru a rezolva o serie de task-uri venite de la clienți sub forma unor obiecte serializate, comunicația realizându-se prin intermediul socket-urilor.

Datorită structurii modularizate sistemul poate fi modificat foarte ușor pentru a rezolva task-uri variate precum rezolvarea unor algoritmi sau prelucrarea unor imagini, în acest caz task-urile reprezentând generarea unor fractali. Fractalul poate fi considerat o figură geometrică cu o formă neregulată, creat cu ajutorul unor ecuații matematice simple. Clientul poate trimite spre generare un fractal în funcție de o serie de parametri, ulterior având posibilitatea de a prelucra acel fractal, prelucrare care în funcție de complexitate dar și de calculele necesare se poate face într-un timp mai îndelungat. Rezultatul returnat sub forma unei imagini se poate salva într-o bază de date, clientul având la dispoziție un istoric cu task-urile sale. Principalul motiv pentru care s-a ales generarea unui fractal drept mod de a pune în evidență funcționarea sistemului și distribuția task-urilor este timpul de prelucrare foarte variat, alături de caracteristicile dar și aspectul unic al acestora.

Structura aplicației cuprinde trei module principale. Modulul Client - reprezintă interfața la care au acces utilizatorii, implementată cu ajutorul tehnologiei JavaFX. Modulul Dispatcher - care realizează maparea task-urilor, având și el o interfață grafică, unde se pot monitoriza și gestiona în timp real clienții, dar și workerii. Modulul Worker - are implementată logica de calcul a fractalilor. Distribuția task-urilor către workeri reprezintă un criteriu definitoriu când vine vorba de performanța aplicației, mai ales când numărul de task-uri este foarte mare și avem nevoie de o distribuție uniformă și eficientă. Astfel s-a implementat o adaptare proprie a mai multor algoritmi de mapare a task-urilor, precum Round Robin sau Opportunistic Load Balancing, dar și o strategie de „Work stealing”, care asigură redistribuirea unor task-uri între workeri în funcție de o serie de criterii, precum gradul de ocupare sau durata de execuție a unui task.