The performance analysis of parallel algorithms executed over grid networks

Asist. Felician ALECU
Catedra de Informatică Economică, ASE Bucureşti

A grid is a collection of machines, sometimes called as nodes, resources, members, donors, clients, hosts, engines and so on. The goal is to create the illusion of a powerful computer out of a large collection of connected systems sharing resources. Some resources may be used by all users of the grid while others may have specific restrictions. The most common resource is computing cycles provided by the processors. Grid computing represents unlimited opportunities in terms of business and technical aspects. The main reason of parallelization a sequential program is to run the program faster. The first criterion to be considered when evaluating the performance of a parallel program is the speedup used to express how many times a parallel program works faster than the corresponding sequential one used to solve the same problem.

Keywords: *grid computing, grid network, parallel processing, performance analysis, parallel speedup, parallel efficiency.*

Procesarea grid creează utilizatorilor impresia existenței unui singur calculator virtual puternic în condițiile în care sistemul este de fapt format prin interconectarea mai multor sisteme de calcul eterogene care partajează o serie de resurse. Un astfel de comportament poartă numele de uniprocesor virtual datorită faptului că utilizatorii nu trebuie să se preocupe de aspecte legate de localizarea resurselor, protocoalele de comunicație folosite, modalitățile de planificare utilizate, etc.

Standardizarea mecanismelor de comunicație dintre sisteme eterogene a condus la explozia internetului. Interconectarea unor sisteme individuale şi standardizarea modalităților de folosire a resurselor partajate vor conduce cu siguranță la explozia procesării de tip grid.

Acest tip de procesare reprezintă o modalitate de îmbunătățire a modului de utilizare a resurselor de calcul prezente în rețea. În marea majoritate a organizațiilor există resurse de calcul slab utilizate (procesor, memorie, discuri, etc.). Calculatoarele de tip desktop, de exemplu, au o rata de utilizare mai mică de 5%. Eficiența utilizării unor astfel de resurse poate creşte simțitor prin folosirea procesării de tip grid.

Rețelele de tip grid, denumite şi clustere de stații de lucru, oferă un suport deosebit pentru procesarea paralelă prin utilizarea simultană a mai multor elemente de procesare. În felul acesta se obține o putere de calcul sporită care este folosită de aplicații pentru minimizarea timpului în care sunt obținute rezultatele dorite [Jos03]. Procesarea de tip grid, datorită faptului că foloseşte resurse deja existente dar neutilizate, reprezintă a alternativă mult mai puțin costisitoare a calculatoarelor paralele. Această putere de calcul obținută prin interconectarea mai multor sisteme de calcul a deschis noi orizonturi nu numai în domeniul ştiințific dar şi în domenii cum ar fi biomedicina, modelarea financiară, explorarea petrolieră, animația video şi multe altele [Lad04].

Pentru ca un program să poată fi rulat în cadrul unei rețele de tip grid este necesar ca acesta să poată fi divizat în sarcini de calcul independente care să fie executate în paralel. Cu cât taskurile ce compun programul comunică mai puțin între ele cu atât aplicația devine mai scalabilă. Un program perfect scalabil ar fi de 10 ori mai rapid în cazul în care ar fi rulat pe 10 procesoare față de cazul în care s-ar executa într-un sistem uniprocesor. Scalabilitatea este măsura eficienței cu care sunt utilizate procesoarele din cadrul rețelei de tip grid.

Desigur că există o serie de bariere ce stau în

calea scalabilității totale cum ar fi numărul de taskuri în care poate fi divizat programul sau existența unor taskuri care nu sunt independente unele față de altele fiind necesar ca ele să comunice între ele şi să se sincronizeze [Wyr04]. De asemenea, mecanismele de comunicare şi sincronizare implementate la nivelul rețelei grid pot sta în calea atingerii scalabilității totale.

Rețelele de tip grid sunt formate din calculatoare individuale foarte diferite din punct de vedere al vitezei de calcul, capacității de memorare şi al lățimii de bandă de care beneficiază în ceea ce priveşte rețeaua de interconectare. Această eterogenitate a calculatoarelor individuale care compun rețeaua de tip grid face imposibilă folosirea metricilor tradiționale pentru exprimarea indicatorilor de performanță.

O rețea de tip grid poate fi privită ca un graf G = (V, M), unde V este mulțimea vârfurilor (stațiile individuale, în număr de p) iar M mulțimea muchiilor (conexiunilor) realizate cu ajutorul rețelei de interconectare. O muchie între două noduri există numai dacă cele două noduri sunt direct conectate. Datorită faptului că în general fiecare stație este conectată cu oricare alta, graful astfel obținut este în majoritatea cazurilor unul complet conectat.

Timpul de execuție al unui algoritm secvențial se exprimă ca funcție de dimensiunea datelor de intrare (numărul acestora). Timpul secvențial de execuție va fi notat în cele ce urmează cu T_s.

Un algoritm paralel este evaluat tot pe baza timpului de execuție însă acesta depinde nu numai de dimensiunea intrărilor ci şi de arhitectura calculatorului paralel şi de numărul de procesoare existente în sistem. Timpul de execuție paralelă (T_p) reprezintă timpul scurs din momentul în care programul paralel este lansat în execuție până când ultimul procesor îşi încheie execuția.

Atunci când se realizează analiza teoretică a algoritmilor, timpii de execuție sunt exprimați cu ajutorul numărului de paşi de calcul (operații elementare) ce se execută, în cazul cel mai defavorabil, pentru rezolvarea unei probleme de dimensiune dată.

Cel mai important criteriu luat în considerare atunci când se doreşte evaluarea performanțelor unui program paralel este accelerarea paralelă care exprimă de câte ori programul paralel este mai rapid față de varianta secvențială [Dod02]. Accelerarea paralelă se notează cu S şi reprezintă câştigul de viteză ce se obține datorită execuției algoritmului pe un sistem paralel în comparație cu timpul de execuție în varianta secvențială. Acest indicator se exprimă ca raport între timpul de execuție în cazul cel mai defavorabil al celui mai rapid algoritm secvențial care rezolvă o anumită problemă şi timpul de execuție necesar unui algoritm paralel pentru a soluționa aceeaşi problemă pe un sistem cu p procesoare identice cu cel al calculatorului secvențial:

$$S = \frac{T_s}{T_p}$$

Valoarea maximă a accelerării paralele este egală cu numărul de procesoare din sistem. O astfel de valoare poate fi atinsă într-un sistem ideal în care nu există costuri de comunicare iar procesoarele sunt încărcate echilibrat.

Teoretic, accelerarea paralelă nu poate avea o valoare mai mare ca p, numărul de procesoare. O accelerare paralelă egală cu numărul de procesoare din sistem poartă numele de accelerare liniară.

Cu toate acestea, în practică se pot obține şi valori mai mari ca numărul de procesoare în anumite situații speciale [Jor02]. Un astfel de fenomen poartă numele de accelerare supraliniară şi apare în situații în care algoritmul secvențial este pus în dificultate de factori hardware. Astfel, dacă datele programului nu pot fi memorate în întregime în memoria principală, algoritmul va pierde timpi importanți pentru accesarea memoriei secundare. Varianta paralelă nu se va confrunta cu această problemă deoarece datele vor fi distribuite procesoarelor şi vor fi memorate în memoriile interne ale acestora fără a mai fi necesar să se acceseze memoria secundară.

În conformitate cu legea lui Amdahl, chiar si într-un sistem paralel ideal este foarte dificil de obținut o accelerare paralelă egală cu numărul de procesoare datorită faptului că în cadrul oricărui program există o fracție α ca-

re nu poate fi paralelizată şi care trebuie executată secvenţial. Restul de (1 - α) paşi de calcul se pot executa în paralel pe procesoarele disponibile în sistem. În aceste condiţii, accelerarea maximă care se poate obţine atunci când o fracţie α a programului nu poate fi paralelizată este 1/α indiferent de numărul de procesoare din sistem.

Legea lui Amdahl exprimă în mod clar necesitatea minimizării fracţiei α ce nu poate fi paralelizată prin stabilirea unei limite superioare a acceleărării paralele.

Pentru ducerea la bun sfârşit a unei sarcini de calcul este necesar ca procesoarele sistemului să comunice între ele. Transferul de date între două procesoare ale sistemului este o operaţiune mare consumatoare de timp care poate genera inactivitate la nivelul elementelor de procesare. Din acest motiv, algoritmii paraleli ar trebui să fie proiectaţi astfel încât majoritatea acceselor la memorie să nu necesite transportul datelor solicitate prin reţeaua de interconectare.

Existenţa în cadrul unui program a unui punct de sincronizare între mai multe procesoare va genera timpi de aşteptare deoarece execuţia programului va continua numai după ce toate procesoarele au atins punctul respectiv.

Principală cauză pentru care la nivelul elementelor de procesare se înregistrează timpi de inactivitate o constituie încărcarea neechilibrată a procesoarelor care survine atunci când procesoarele din sistem au asociate sarcini de calcul de dimensiuni inegale. Într-un astfel de caz, unele procesoare îşi vor încheia execuţia mai devreme şi vor rămâne în aşteptare în timp ce altele vor fi încă în faza de execuţie.

O altă sursă care generează timpi de inactivitate şi implicit o încărcare neechilibrată a procesoarelor o reprezintă variaţia gradului de paralelism pe parcursul execuţiei programului, unde prin gradul de paralelism al unui program paralel înţelegem numărul de procesoare care sunt utilizate pentru execuţia acestuia.

Chiar şi prezenţa în cadrul unui program a unei părţi ce nu poate fi paralelizată atrage după sine o astfel de încărcare neechilibrată a procesoarelor deoarece fracţia respectivă va trebui să fie executată secvenţial, de către un singur procesor. O astfel de situaţie este exprimată de către legea lui Amdahl.

Din aceste motive, pentru obţinerea unui program paralel care să fie executat cât mai rapid este necesar să se reducă cât mai mult posibil fracţia de program ce nu se poate paraleliza şi în plus să se asigure o distribuţie cât mai uniformă a sarcinii de calcul pe elementele de procesare în condiţiile în care se are în vedere minimizarea timpilor de comunicaţie [Tan95].

Considerăm un program paralel ce este executat într-un timp T_p pe o reţea grid formată din p calculatoare individuale numerotate de la 1 la p. Timpul secvenţial de execuţie al programului pe sistemul individual cu indicele i va fi notat cu T_s^i. Accelerarea execuţiei programului pe cluster-ul de staţii poate fi exprimată prin raportul dintre cel mai bun timp secvenţial şi timpul paralel de execuţie:

$$S_{grid} = \frac{\min_{i=1}^{p} T_s^i}{T_p}$$

Datorită faptului că staţiile individuale nu sunt identice, acestea vor avea puteri de calcul diferite [Dod02]. Ponderea puterii de calcul a unei staţii de calcul individuale în comparaţie cu cea a celui mai rapid calculator din cadrul reţelei grid se poate calcula utilizând formula următoare:

$$P_i = \frac{\min_{j=1}^{p} T_s^j}{T_s^i},\ i = 1..\text{p}$$

Valorile pe care le pot lua aceste ponderi satisfac inegalitatea $P_i \le 1$.

Gradul de eterogenitate a calculatoarelor ce intră în componenţa reţelei grid se poate cuantifica cu ajutorul diferenţei de putere de calcul ce există între acestea:

$$GE = \frac{\sum_{i=1}^{p} (1 - P_i)}{p}$$

Variaţia gradului de paralelism pe parcursul execuţiei unui program generează încărcarea neechilibrată a procesoarelor sistemului, unde prin gradul de paralelism al unui program paralel înţelegem numărul de procesoare care

sunt utilizate pentru execuția acestuia. Gradul mediu de paralelism al unui program se defineşte ca fiind numărul mediu de stații de lucru active pe tot parcursul execuției acestuia:

$$GP_m = \frac{\sum_{i=1}^{p} T_p^i}{T_p}$$

unde T_p^i reprezintă timpul cât stația i a fost activă.

Formula următoare exprimă accelerarea paralelă în funcție de eterogenitatea rețelei grid şi de gradul mediu de paralelism al programului ce se execută pe cluster-ul de stații:

$$S_{grid} = GP_m \cdot (1 - GE).$$

Bibliografie

[Lad04], S. Ladd, *Guide to Parallel Programming*, Springer-Verlag, 2004

[Wyr04], R. Wyrzykowski, *Parallel Processing And Applied Mathematics*, Springer, 2004

[Jos03], J. Joseph, C. Fellenstein, *Grid Computing*, Prentice Hall, 2003

[Dod02], Gh. Dodescu, B. Oancea, M. Raceanu, *Procesare paralelă*, Editura Economica, Bucureşti, 2002

[Jor02], H. F. Jordan, H. E. Jordan, *Fundamentals of Parallel Computing*, Prentice Hall, 2002

[Tan95], A. S. Tanenbaum, *Distributed Operating Systems*, Prentice Hall, 1995