

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE****Algoritmi distribuiți**

Autor: Denisa-Teodora URCAN

1. Algoritmi pentru alegere leader

În contextul algoritmilor distribuiți, diverse procesări necesită un coordonator (lider) cu rol distinctiv, urmărind modelul uman. În contextul lipsei acestuia, se determină un nou astfel de proces distinctiv înainte finalizării procesării. Orice algoritm distribuit care are posibilitatea determinării unui astfel de proces trebuie să fie sigur și longeviv. Printre algoritmii din această categorie putem menționa Bully și Ring, care se vor compara în cele ce urmează.

Algoritmul Bully folosește 3 tipuri de mesaje și începe cu o alegere a unui coordonator atunci când realizează că recentul lider a esuat (fie prin intermediul unui timeout, fie printr-un mesaj de tip Election). La începerea acestui proces de alegere mesajul Election este trimis tuturor proceselor cu număr de identificare mai mare. Dacă nu se recepționează nici un mesaj de tip Answer într-un interval de timp predefinit, procesul care a început alegerea va decide că trebuie să devină coordonator și va trimite un mesaj de tipul Coordinator către toate celelalte procese. În caz contrar, dacă un astfel de mesaj de tip Answer este recepționat, procesul care a inițiat această alegere așteaptă, un interval de timp predefinit, un mesaj de tip Coordinator. Un proces care primește un mesaj de tipul Election poate anunța instantaneu că el este coordonatorul dacă știe sigur că deține numărul de identificare cel mai mare, fapt pentru care este numit și forța brută. Pseudocodul este prezentat în cele ce urmează:

1. Fiecare proces are un identificator unic (numărul procesului),
2. Fiecare proces cunoaște identificatele tuturor celorlalte procese,
3. Procesele pot fi active sau nu,
4. Un proces este lider deoarece îndeplinește un rol special în sistemul distribuit,
5. Dacă liderul cade, celelalte procese trebuie să aleagă unul dintre ele pentru a fi noul lider,
6. Transferarea mesajelor este sincronă,
7. Când un proces observă că liderul este căzut, procesul trimite un mesaj "election" la toate procesele cu identificate mai mari decât cel propriu,

8. Când un proces primește un mesaj "election" de la un proces cu număr mai mic, procesul trimite un mesaj "OK" înapoi la procesul cu număr mai mic, apoi trimite un mesaj "election" tuturor proceselor cu numere mai mari,
9. Când un proces primește un mesaj "OK", procesul nu mai face nimic și așteaptă un mesaj "leader",
10. Dacă un proces nu primește un răspuns la un mesaj "election", acest proces devine noul lider,
11. Noul lider trimite un mesaj de "lider" tuturor proceselor, informându-i cine este noul lider (coordonatorul),
12. Procesul cu numărul cel mai mare devine întotdeauna liderul (bully process),
13. Când începe un proces, inițiază o alegere și devine lider (sau nu) dacă este cel mai mare număr.

Algoritmul Ring este o alternativa a algoritmului Bully, procesele avand o organizare diferita, cea de tip inel si folosind doar doua tipuri de mesaje: election si leader. Analog algoritmului anterior, un proces incepe alegerea in momentul in care observa ca liderul a esuat, moment in care mesajul de tip election este trimis pe inel. Acest mesaj este trimis mai apoi prin inel, fiecare proces adaugandu-si propriul identificator la mesaj. Cand mesajul ajunge inapoi la cel care a inaintat procesul si implicit alegerea, va deveni lider procesul cu identificator maxim si se trimite un mesaj de tip lider, informand cine este castigatorul.

Pseudocodul este prezentat in cele ce urmeaza:

1. Procesele sunt aranjate sub forma unui inel, in ordinea numarului.
2. Cand un proces observa faptul ca liderul a esuat, trimite un mesaj de tip election procesului vecin, lista ce contine initial numarul procesului ce a declansat alegerea.
3. Procesul asteapta sa primeasca o confirmare de la destinatar.
4. Daca nu a primit o confirmare, va trimite mesajul urmatorului proces din inel.
5. Cand un proces primeste un mesaj de tip election, numarul lui este adaugat in lista din mesaj si acesta este trimis mai departe.
6. Procesul ce a declansat alegerea primeste un mesaj de tip election ce contine numarul tuturor proceselor si alege noul lider, trimitand un mesaj de tipul leader in jurul inelului.
7. Cand mesajul ajunge inapoi la procesul ce a declansat alegerea, aceasta se incheie.

Prin comparatie, se observa ca in cazul algoritmului Bully, "worst case" apare atunci cand procesul cu identificatorul cel mai mic observa esecul liderului, caz in care se vor trimite

$O(n^2)$ mesaje, cauzand si un trafic mare de mesaje in retea. In cel mai bun caz, procesul care detecteaza esuarea este cel cu identificatorul cel mai mare, rezultand $O(n-1)$ mesaje. Algoritmul Ring trimite $n-1$ mesaje de tip election si $n-1$ mesaje de tip lider si este constant fata de identificatorul procesului care detecteaza esecului, rezultand mereu $2(n-1)$ mesaje pentru o alegere.

2. Algoritmi de excludere mutuala care folosesc ceasuri logice

Primul astfel de algoritm discutat este cel propus de Ricart si Agrawala, care presupune ca fiecare proces participat p_i retine un ceas de tip Lamport si toate procese trebuie sa comunice in perechi. In orice moment al procesarii un proces se poate afla in una din urmatoarele stari: Released, Wanted si Held, stari in cadrul carora un proces oscileaza. Acest algoritm e o imbunatatire a algoritmului publicat initial de Lamport, avand 3 componente care coordoneaza modul in care se comporta un proces cand vrea sa intre intr-o zona critica, sa iasa din zona critica, sau cand primeste un mesaj de solicitare de la un alt proces. Acest algoritm necesita o metoda de asignare a timpilor catre evenimente care sunt asignate in ordinea cresterii marcajelor de timp. Fiecare mesaj dintre procese e insotit de un marcaj de timp care indica timpul procesului care a trimis mesajul. Două procese nu pot avea același marcaj de timp și fiecare proces trebuie să fie de acord cu ordonarea cererilor. Algoritmul eșuează dacă oricare din procesele participante este incapabil să răspundă la mesaje. Se poate observa faptul că sunt necesare $2(n-1)$ mesaje pentru fiecare intrare în zona critică. O posibila optimizare este implementarea unui proces care sa emita o confirmare a fiecarui mesaj de solicitare.

Algoritmul lui Maekawa este un algoritm pentru excluderea reciprocă pe un sistem distribuit. Acesta presupune ca fiecare proces p_i va avea o multime de cereri, notata cu R_i , multimi de cereri care trebuie sa detina o serie de proprietati, printre care: intersectia a doua multimi de cereri sa nu fie vida, fiecare proces sa apartina multimii sale de cereri, fiecare multime de cereri sa aiba acelasi numar de elemente notat K si fiecare proces sa apartina la exact K multimi. Se folosesc trei tipuri de mesaje: request, ok si end si se poate descrie astfel: Un dispozitiv de calcul care ruleaza acest algoritm si este solicitant trimite un mesaj catre toate dispozitivele care mai ruleaza acest algoritm, din setul sau. Dispozitivele care primesc un mesaj, efectueaza urmatoarele: daca nu are nici un mesaj care nu a fost lansat, trimite un mesaj catre dispozitivul de la care a primit; daca are un mesaj restant cu proces care detine prioritate mai mare, atunci dispozitivul trimite un mesaj si dispozitivul care a trimis mesajul pune in coada solicitarea acestuia; daca procesul are prioritate mai mica, atunci dispozitivul curent trimite un mesaj procesului caruia i s-a acordat acces la sectiunea critica.

In comparatie cu metoda anterioara, sunt necesare doar $3\sqrt{N}$ mesaje pentru intrarea intr-o zona critica. Posibilele probleme care pot aparea sunt cele de aparitie a unui blocaj la nivelul proceselor. Un alt aspect diferit pentru cei doi algoritmi este coada care va memora mesajele sosite. Pentru algoritmul Ricart-Agrawala, coada este non-fifo (non first in-first out), fapt care se implementeaza printr-o coada simpla care plaseaza mesajele in ordine aleatoare. Pe de alta parte, algoritmul Maekawa, foloseste o coada fifo uzuala.

De asemenea, algoritmul Ricart si Agrawala foloseste $2(n-1)$ mesaje, dintre care $n-1$ mesaje de tip request si $n-1$ mesaje de tip reply, in timp ce Maekawa scade numarul de mesaje, desi foloseste 3 tipuri.