Sisteme Distribuite – Teorie
11. Modele de calcul distribuit

Modele de baza

- Procesoarele dintr-un sistem de calcul distribuit pot fi organizate in numeroase moduri.
- Vom urmari principalele:
  1. Modelul statiilor de lucru si
  2. Modelul gruparii de procesoare,
  3. Hibrid oferind facilitatile celor doua de mai sus.
- Context: interesul curent in “calcul la cerere” (on-demand computing)

Modelul statiilor de lucru

- Sistemul consta din statii de lucru (PCuri) aflate intr-o cladire sau campus si conectate printr-o retea LAN de viteza mare.
- Anumite statii pot fi in birouri si astfel sunt dedicate implicit unui singur utilizator, pe cand altele pot fi in domenii publice si pot avea mai multi utilizatori in cursul unei zile
- In ambele cazuri, la orice moment de timp, o statie are un singur utilizator care este logat, si astfel are un “proprietar" (temporar), sau este neutilizat.
- Avantajele modelului statiilor de lucru sunt clare:
  - Modelul este usor de inteles
  - Utilizatorii au o cantitate specifica de putere de calcul si astfel se garanteaza timpul de raspuns.
  - Programe grafice complicate pot fi foarte rapide deoarece au acces direct la ecran
  - Fiecare utiluzator are un grad mare de libertate si poate aloca resursele statiei sale asa cum doreste
  - Discurile locale contribuie la aceasta independenta si fac posibil continuarea lucrului chiar daca apar erori.

Problemele modelului statiilor de lucru

- Modelul are doua probleme majore.
  1. In masura in care cipurile procesor devin din ce in ce mai ieftine, va fi in curand fezabil ca fiecare utilizator sa dispuna de exemplu de 1000 CPUuri.
  2. O parte mare a timpului utilizatorii nu folosesc statiile lor de lucru, care sunt astfel neutilizate, pe cand alti utilizatori care necesita capacitate suplimentara de calcul nu o obtin.
     - Dintr-o perspectiva a sistemelor, alocarea resurselor astfel incat anumiti utilizatori au resurse de care nu au nevoie pe cand alti utilizatori au nevoie de asemenea resurse, este ineficient.
- Prima problema poate fi adresata realizand dintr-o statie de lucru un multi-procesor individual
  - Aceasta situatie poate conduce la o utilizare ineficienta a resurselor, dar cu cat tehnologia devine din ce in ce mai ieftina, irosirea lor devine un pacat mai redus.
- Problema a doua, cea a statiilor de lucru neocupate, este subiect pentru numeroase cercetari, in mod primar pentru ca numeroase universitati au numar mare de statii de lucru individuale din care multe sunt inactive.
  - Masuratorile arata ca, chiar is in perioade maxime de utilizare in decursul unei zile, adesea peste 30 % din statiile de lucru sunt neocupate. In timpul noptii, cu atat mai mult.

Utilizarea statiilor neocupate

- Incercarea cea mai timpurie pentru a permite statiilor de lucru neocupate sa fie utilizate a fost programul care vine de la Berkeley Unix: *rsh*
  - Primul argument numeste masina si a doua comanda care sa fie rulata.
  - Ruleaza comanda specificata pe o masina specificata.
  - Desi este des utilizata, acest program are mai multe probleme:
  1. Utilizatorul trebuie sa spuna care masina sa fie utilizata, punand astfel intreaga povara de a tine evidenta masinilor neocupate asupra utilizatorului.
  2. Programul se executa in mediul masinii la distanta, care este in mod uzual diferite de mediul local.
  3. Daca cineva trebuie sa se logheze pe masina pe care procesul ruleaza, procesul va continua sa ruleze si utilizatorul nou logat trebuie sa accepte performanta scazuta sau sa gaseasca alta masina.

Cercetarea asupra statiilor de lucru inactive

Intrebarile cheie sunt:

1. Cum sunt gasite statiile de lucru inactive?
2. Cum poate rula transparent un proces la distanta?
3. Ce se intampla daca proprietarul masinii se intoarce?

Cum sunt gasite statiile de lucru inactive?

- Statii de lucru inactive?
  - La o prima privire, poate aparea ca o statie de lucru pe care nu este nimeni logat de la consola este o statie de lucru inactiva,
  - Dar in numeroase sisteme, desi nu este nimeni logat exista o multime de procese care ruleaza, precum procese demoni pentru sincronizarea timpului, mail, stiri etc.
- Pe de alta parte, un utilizator care se logheaza cand vine dimineata la servici poate sa atinga calculatorul pentru multe ore sau nu foloseste toate resursele.
- Sisteme diferite iau decizii diferite a ceea ce inseamna “inactiv”
- *Uzual* se considera ca daca nimeni n-a atins tastatura sau mouse-ul pentru mai multe minute si nu exista nici un proces in rulare care a fost initiat de utilizator, statia de lucru este considerata inactiva.
  - In consecinta, este posibil sa existe diferente substantiale in incarcarile dintre o statie de lucru si alta, datorate, de exemplu, volumului de mailuri care ajunge la primul si nu si la al doilea.

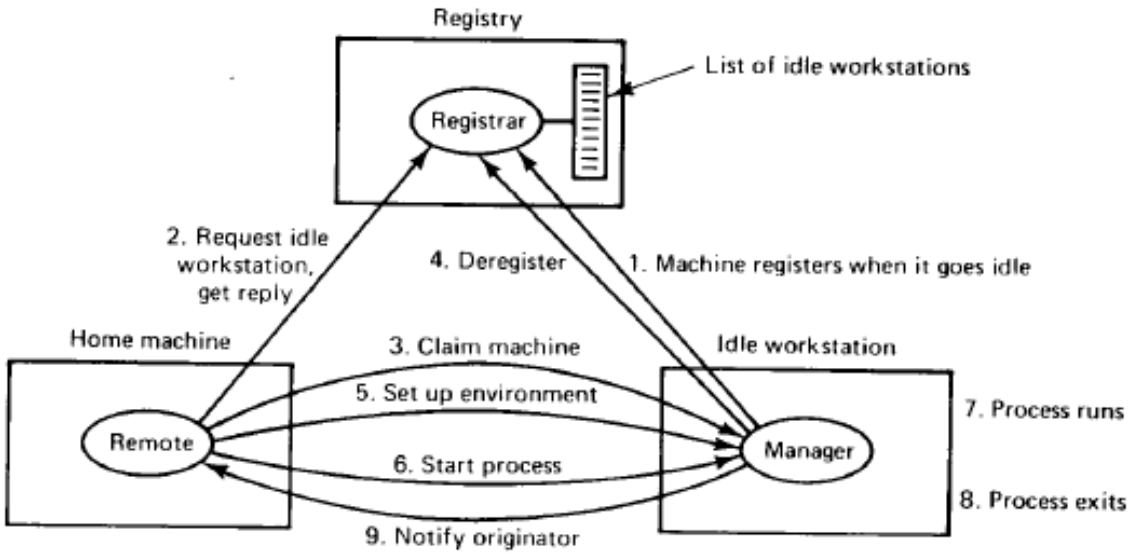
Localizarea statiilor de lucru inactive

- Algoritmii utilizati pentru localizarea statiilor libere pot fi divizate in doua categorii:

1. Conduse de server
2. Conduse de client

- Conduse de server
  - Cand o statie de lucru devine inactiva, si devine astfel un server potential pentru calcule, anunta disponibilitatea sa.
  - Poate sa faca acest lucru anuntand numele sau, adresa de retea, si proprietatile de exemplu la un registru (fisier sau baza de date).
  - Cand un utilizator doreste sa execute o comanda la o statie inactiva, introduce comanda si are loc o cautare in registru (pe baza de program) pentru a gasi o statie de lucru inactiva adecvata.
  - Din motive de incredere este posibil sa existe mai multe copii ale registrului

Localizare condusa de server



- O modalitate alternatica pentru statiile de lucru care devin inactive este sa anunte faptul ca sunt neutilizate este aceea de a difuza un mesaj in retea.
  - Toate celelalte statii de lucru inregistreaza acest fapt.
  - De fapt, fiecare masina mentine propria sa copie privata a registrului
  - Avantaj: surplus mai mic in gasirea unei statii de lucru inactive si o redundanta mai mare.
  - Dezavantajul este necesitatea lucrului pentru mentinerea registrului.
- Indiferent daca este un registru sau mai multe, exista un pericol potential datorat conditiilor de tratare concurenta
  - Daca doi utilizatori invoca o metoda la distanta simultan si amandoi descopera aceeasi masina care este inactiva, amandoi pot porni procese in acelasi timp pe aceasta.
  - Pentru a detecta si evita aceasta situatie, sistemul la distanta poate verifica daca statia la distanta este inca libera si marca daca nu este cazul
  - Apelantul poate trimite indicatii de mediu de executie si poate starta procesele cf. Fig.

Localizare condusa de client

- Cand se invoca un program la distanta, se difuzeaza o cerere indicand care program se doreste a fi rulat, cata memorie este necesara, etc.
- Aceste detalii nu sunt necesare daca toate statiile de lucru sunt identice, dar daca sistemul este eterogen si nu orice program poate rula pe orice statie, acestea sunt esentiale.
- Cand replicile vin inapoi, se considera unul dintre ele si se seteaza corespunzator
- O facilitate adecvata este aceea de a solicita statiilor de lucru inactive sa intarzie raspunsurile proportional cu incarcarea pe care o au
  - Astfel raspunsul de la cea mai putin incarcata statie va ajunge primul si va putea fi selectat.

Intrebarea cheie 2: Rularea la distanta

- Mutarea codului este simpla.
- Pentru a rula trebuie setat mediul la distanta pentru a se asemana cu cel dorit pentru rulare, a.i.
  - Sa fie executat ca si cum ar fi executat local.
  - Este necesara aceeasi imagine asupra sistemului de fisiere, acelasi director de lucru, aceleasi variabile de sistem (daca sunt necesare).
- Probleme pot sa apara cand primul apel de sistem, fie un READ, este executat.
  - Ce trebuie sa faca nucleul?
  - Raspunsul depinde in mare masura de arhitectura sistemului.
    - Daca toate fisierele sunt localizate pe servere de fisiere, nucleul poate emite o cerere catre serverul de fisiere adecvat in aceeasi modalitate in care o face pentru un proces local.
    - Daca sistemul are discuri locale, fiecare cu un sistem de fisiere complet, cererea trebuie inaintata la masina gazda pentru executie.

Apeluri de sistem la distanta

- Anumite apeluri de sistem trebuie inaintate la masina gazda
  - De exemplu, citirile de la tastatura si afisarea pe ecran nu pot fi realizate la masina la distanta.
- Alte apeluri de sistem trebuie realizate la distanta.
  - De exemplu, toate apelurile sistem care interogheaza starea masinii trebuie sa fie executate pe masina pe care procesul ruleaza.
    - Acestea include interogari precum numele masinii si adresa de retea, cata memorie este disponibila, etc.
- Apelurile de sistem ce implica timp sunt o problema pentru ca ceasurilor diferitelor masini pot sa nu fie sincronizate.
  - Forwardarea apelurilor legate de timp catre masina gazda introduce de asemenea intarzieri.
- Anumite apeluri precum crearea si scrierea de fisiere temporare pot fi realizate mai eficient pe masina la distanta.

=> A face programele sa ruleze pe masini la distanta ca si cum ar rula pe masina locala este o afacere complexa si delicata.

Intrebarea cheie 3: ce se intampla cand proprietarul masinii se intoarce?

1. Varianta cea mai simpla este aceea de a nu face nimic, dar aceasta abordare este impotriva ideei de statie de lucru "personala".
   - Daca alte persoane pot rula programe pe statia ta de lucru in acelasi timp in care incerci s-o utilizezi, se modifica timpul garantat de raspuns.
2. O alta posibilitate este de a omori toate procesele intruse.
   - Modalitatea cea mai simpla este aceea de a o face abrupt si fara avertizare.
     - Dezavantajul este acela ca rezultatele sunt pierdute si sistemul de fisiere poate fi intr-o stare haotica.
   - O modalitate mai buna este de a oferi procesului o avertizare prin expedierea unui semnal legat de o moarte iminenta si oprirea cu gratie (scrierea bufferelor pe disc, inchidere fisiere etc).
     - Daca nu s-a oprit in cateva secunde, este terminat.
     - Desigur, programul trebuie scris astfel incat sa astepte si sa trateze acest semnal, ceea ce nu fac majoritatea programelor.

Migrare

3. O abordare complet diferita este migrarea procesului catre o alta masina, fie la masina locala, fie la o alta statie de lucru inactiva.
   - Partea grea nu este mutarea codului si datelor utilizatorului, ci gasirea si strangerea tuturor datelor legate de procesul care paraseste sistemul.
     - De ex, poate avea fisiere deschise, contoare, mesaje de primit in coada si o serie de alte informatii imparstiate in nucleu.
     - Toate aceastea trebuie inlaturate cu atentie din masina si resintalate cu succes la noua masina destinatie.
     - Desi nu exista probleme teorerice dificile, dificultatile practice ingineresti sunt substantiale.
- Cand procesul pleaca, trebuie sa lase masina in aceeasi stare in care a gasit-o, pentru a evita disturbarea proprietarului.
  - Aceasta cerinta inseamnca ca nu numai procesul in sine trebuie sa plece, dar si tiate procesele fii precum si fii acestora.
  - Conexiunile de retea si alte structuri de date legate de sistem pot fi sterse si trebuie igonarate replicile la RPC sau akte messaje care sosesc pentru proces dupa ce acesta a plecat.
  - Fisierele temporare trebuie sterse, si daca este posibil, sa se restaureze cacheul.

Caz particular: calcul voluntar

- BOINC - standard
  - Standard open-source pentru volunteer computing
  - Utilizeaza timpul de inactivitate a calculatoarelor (Windows, Mac, sau Linux) pentry numeroase tipuri de cercetari stiintifice.
  - Aplicatii: studii asupra bolilor, incalzirea globala, descoperirea pulsarilor…
  - Aplicatii curente:
    - Astronomie/Fizica/Chimie:
      - Einstein@home, Milkyway@home, Quantum Monte Carlo@Home, Spinhenge@home,LHC@home, SETI@home, Cosmology@Home,uFluids@home
    - Biologie si Medicina:
      - Rosetta@home, GPUGrid.net, Superlink@Technion, POEM@HOME, Malariacontrol.net, Docking@Home
    - Matematica, calcul, si jocuri
      - Rectilinear Crossing Number, NFS@home, VTU@home, SHA-1 Collision Search, ABC@home, AQUA@home, PrimeGrid, Chess960@home, NQueens@home
- Altele:
  - XtreemWeb
  - Desktop Grid
  - GridRepublic + Intel programme Progress Thru Processors

Modelul gruparii de procesoare (Cluster)

- Desi utilizarea statiilor de lucru inactive adauga putere de calcul la sistem, nu adreseaza o tema fundamentala mai importanta:
  - Ce se intampla cand este fezabil sa se ofere de 10 sau 100 de ori mai mult CPUs decat utilizatori activi?

1. O solutie, indicata anterior, este de a oferi fiecaruia un multi-procesor personal - totusi acesta este un design ineficient.
2. O abordare alternativa este aceea de a construi o *grupare de procesoare (processor pool)*, un rack plin de CPUuri dintr-o sala a serverelor, care este alocata dinamic utilizatorilor la cerere.
   - In locul ofertei de statii de lucru invididuale pt. utilizatori, in acest model utilizatorii au terminale pentru facilitati de calcul de perfoamanta inalta.

- Conceptual, acest model este mai apropriat de modelul traditional de partajare a timpului decat de modelul statiilor de lucru, desi este construit pe baza tehnologiilor moderne.

Motivare

- Daca sistemul de fisiere poate fi centralizat intr-un numar mic de servere de fisiere pentru a castiga economii in ceea ce priveste scara, este posibil sa se faca aceasta lucru si cu serverele de calcul.
- Prin punerea a mai multor CPUuri intr-un singur rack mare in sala serverelor, se fac redceri in ceea ce priveste consumul de energie.
- Modelul permite de asemenea crestere incrementala.
  - Daca incarcarea de calcul creste cu 10 procente, se pot cumpara cu 10% mai multe procesoare care sa fie introduse in grupare.
- Puterea de calcul este transformata in “statii de lucru inactive” care sunt accesate dinamic
  - Utilizatorii pot fi asignati la numeroase CPUuri pentru perioade scurte dupa care returneaza controlul asupra acestora.
  - Nu exista un concept de proprietate: toate procesoarele apartin in mod egal la fiecare.

Natura incarcarii

- Daca utilizatorul realizeaza o simpla editare si ocazional expedeaza mesaje electronice => o statie de lucru individuala este suficienta
- Daca, pe de alta parte, utilizatorii
  - Sunt angajati intr-un proiect mare de dezvoltare, ruland frecvent sau pe directoare mari
  - Incearca sa faca calcule numerice pe structuri de date foarte mari sau
  - Fac simularri sau ruleaza programe mari de inteligenta artificiala sau programe de rutare VLSI

  => Sunt in permanenta vanatoare pentru un numar substantial de calculatoare inactive.

  => In toate aceste situatii, ideea grupuli de procesoare este fundamental simpla si atractiva.

Sisteme de cozi

- Un sistem de cozi trateaza o situatie in care utilizatorii genereaza cereri intr-un mod aleator pentru lucru catre un server.
  - Cand serverul este ocupat, utilizatorii sunt pusi in coada pentru procesare si servicii.
  - Exemple comune sin viata de zi cu zi sunt: cozile la super-market-uri, la inregistrarea la airoport, etc.
- Sistemele de cozi pot fi modelate analitic.

Exemple de sisteme de cozi pt. clustere

- Condor
  - Este un sistem de administrare a incarcarii specilaizat pentru sarcini de calcul intensiv
    - Ofera un mecanism al cozilor de sarcini, politici de planificare, scheme de prioritate, mnitorizarea resurselor, si administrarea resurselor.
  - Pentru sisteme batch
  - Utilizatorii submit sarcinile la Condor, care
    - Le plaseaza intr-o coada
    - Alege cand si unde sa ruleze sarcinile si cu ce prioritate
    - Moitorizeaza cu atentie progresul acestora
    - Informeaza utilizatorul asupra terminarii
- LSF
  - JobScheduler este parte a unei solutii de administrare a incarcarii
    - Ofera o imagine singualare pentru o retea eterogena de calculatoare
    - Mapare dinamica si inteligenta a resurselor si balansarea incarcarii
    - Monitorizare centralizata a resurselor, informare asupra incracarii si sarcinilo
    - Planificare bazata pe calendar/eveniment/sarcina
- PBS
  - Opereaza in medii multi-platforme Unix

Model hibrid

- Un compromis posibil intre cele doua modele analizate este acela de a avea un PC si acces la un cluster.
  - Desi aceasta solutie este mai costisitoare decat celelalte doua, combina avantajelor lor.
    - Lucrul interactiv poate fi realizat pe statii de lucru, oferind un timp de raspuns garantat.
    - Statiile de lucru inactive raman inactive.
    - Toate procesele ne-interactive ruleaza in cluster, deoarece in general presupun in general calcule complicate.
- Acest model ofera un timp de raspuns rapid si o utilizare eficienta a resurselor
- Referinte: Grid computing, Cloud computing