
VIII. Costuri de comunicare,
mecanism de rutare, tehnici de mapare,
compromisuri cost-performanță

Costuri de transmitere a mesajelor

- Surplusul major în execuția programelor paralele: de la comunicarea informațiilor între elementele de procesare.
- Costul comunicării depinde de o varietate de caracteristici, inclusiv:
 - semantica modelului de programare,
 - topologia de rețea,
 - gestionarea și rutarea datelor,
 - protocoale software asociate.
- Timp necesar pentru a comunica un mesaj între două noduri dintr-o rețea
= timp pentru pregătirea unui mesaj pentru transmisie + timp preluat de mesaj pentru traversarea rețelei până la destinația sa.

Parametrii care determină latența comunicării

■ **Timpul de pornire (t_s):**

- Timpul de pornire este timpul necesar pentru a gestiona un mesaj la nodurile de trimitere și primire.
- Include
 1. timpul pentru pregătirea mesajului (adăugarea informațiilor despre antet, remorcă și corectarea erorilor),
 2. timpul pentru a executa algoritmul de rutare,
 3. timpul pentru a stabili o interfață între nodul local și router.
- Această întârziere este suportată o singură dată pentru un singur mesaj de transfer.

■ **Timpul per-hop (t_h):**

- După ce un mesaj lasă un nod, este nevoie de o perioadă finită de timp pentru a ajunge la următorul nod din calea sa.
- Timpul necesar antetului unui mesaj pentru a călători între două noduri conectate direct în rețea.
- Este cunoscut și sub denumirea de latență nodală.
- Este direct legată de latența din comutatorul de rutare pentru a determina care este bufferul de ieșire sau canalul care trebuie trimis mesajul.

■ **Timpul de transfer per-cuvânt (t_w):**

- Dacă lățimea de bandă a canalului este r cuvinte pe secundă, fiecare cuvânt necesită timp $t_w = 1 / r$ pentru a traversa legătura.
- Acest timp include surplusul rețelei, precum și cele legate de stocare în tampon.

Rutare stocare-si-inaintare

- Când un mesaj parcurge o cale cu mai multe legături, fiecare nod intermediar de pe traseu transmite mesajul către următorul nod după ce a primit și a stocat întregul mesaj.
- Să presupunem că un mesaj cu dimensiunea m este transmis printr-o astfel de rețea. Presupunem că traversează l legăturile.
- La fiecare legătură, mesajul suportă un cost t_h pentru antet și $t_w m$ pentru restul mesajului pentru a traversa linkul.
- Deoarece există l astfel de legături, timpul total este $(t_h + t_w m)l$.
- Prin urmare, pentru rutarea de stocare-înainte, costul total de comunicare pentru un mesaj de dimensiuni m cuvinte pentru a traversa l legăturile de comunicare este

$$t_{comm} = t_s + (mt_w + t_h)l.$$

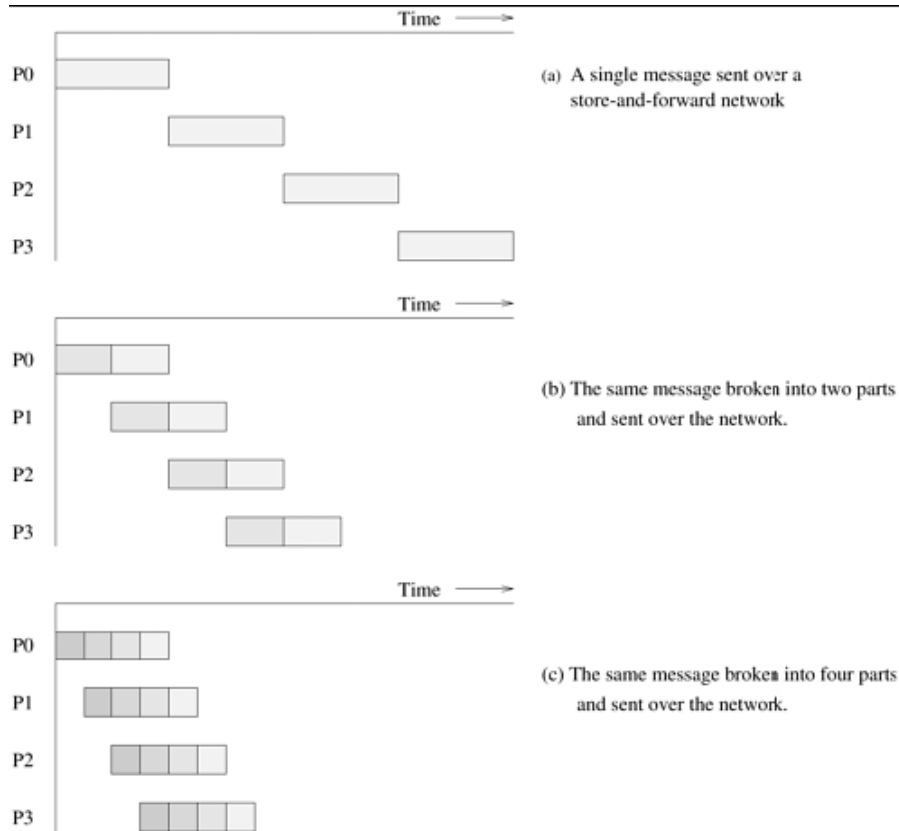
- În calculatoarele paralele curențe, timpul per-hop t_h este destul de mic.
- Pentru majoritatea algoritmilor paraleli, acesta este mai mic decât $t_w m$ chiar și pentru valori mici de m și astfel pot fi ignorate.
- Pentru platformele paralele care utilizează rutarea de stocare-înainte, timpul acordat de ecuația de mai sus poate fi simplificat

$$t_{comm} = t_s + mlt_w.$$

Rutare de pachete

- Stocare și înaintare: mesajul este trimis de la un nod la următorul numai după ce întregul mesaj a fost primit.
- Fie scenariul în care mesajul original este împărțit în două părți de dimensiuni egale înainte de a fi trimis.
 - Un nod intermediar așteaptă să ajungă doar jumătate din mesajul original înainte de a-l transmite.
- Un pas mai departe: se rupe mesajul în patru părți.
- Pe lângă utilizarea mai bună a resurselor de comunicare, acest principiu oferă și alte avantaje:
 - reducerea surplusului din pierderea pachetelor (erori),
 - posibilitatea ca pachetele să urmeze căi diferite,
 - o mai bună capacitate de corectare a erorilor.
- Această tehnică stă la baza rețelelor de comunicații de lungă durată, cum ar fi Internetul, unde ratele de eroare, numărul de hopuri și variația stării rețelei pot fi mai mari.
- Surplusul este dat fiecărui pachet ce trebuie să poarte informații de rutare, corectare a erorilor și secvențiere.

Stocare și înaintare vs. rutare pachet



- Trecerea unui mesaj de la nodul P0 la P3:
 - printr-o rețea de comunicare stocare-și înaintare;
 - extinderea conceptului.
- Regiunile umbrite reprezintă timpul în care mesajul este în tranzit.
- Se presupune că timpul de pornire asociat cu acest transfer de mesaje este zero.

Costul comunicării în rutarea pachetelor

- Fie transferul unui mesaj cu m cuvinte prin rețea.
- Se presupune:
 - Tabelele de rutare sunt statice în timpul transferului de mesaje - toate pachetele traversează aceeași cale;
 - Timpul necesar pentru programarea interfețelor de rețea și pentru calcularea informațiilor de rutare, este independent de lungimea mesajului și este agregat în timpul de pornire t_s
 - Mărimea unui pachet: $r+s$, r -mesaj original, s -informații suplimentare transportate în pachet;
 - Timpul pentru impachetarea mesajului este proporțional cu lungimea mesajului: mt_{w1} .
 - Rețeaua este capabilă să comunice un cuvânt la t_{w2} secunde,
 - Apare o întârziere de t_h la fiecare hop,
 - Primul pachet traversează l hopuri,
- Atunci pachetul ia timp $t_h l + t_{w2}(r + s)$ pentru a ajunge la destinație.
- Nodul de destinație primește un pachet suplimentar la fiecare $t_{w2}(r + s)$ secunde.
- Deoarece sunt $m/r - 1$ pachete aditionale, timpul total de comunicare este dat de:
$$t_{comm} = t_s + t_{w1}m + t_h l + t_{w2}(r + s) + \left(\frac{m}{r} - 1\right) t_{w2}(r + s)$$
$$= t_s + t_{w1}m + t_h l + t_{w2}m + t_{w2}\frac{s}{r}m$$
$$= t_s + t_h l + t_w m,$$
$$t_w = t_{w1} + t_{w2}\left(1 + \frac{s}{r}\right).$$
- Adecvat rețelelor cu stări extrem de dinamice și rate de eroare mai mari,
 - cum ar fi rețelele locale și largi.
 - Pachetele individuale pot lua diferite rute, iar retransmisiile pot fi localizate la pachete pierdute.

Rutare prin taiere (cut-through)

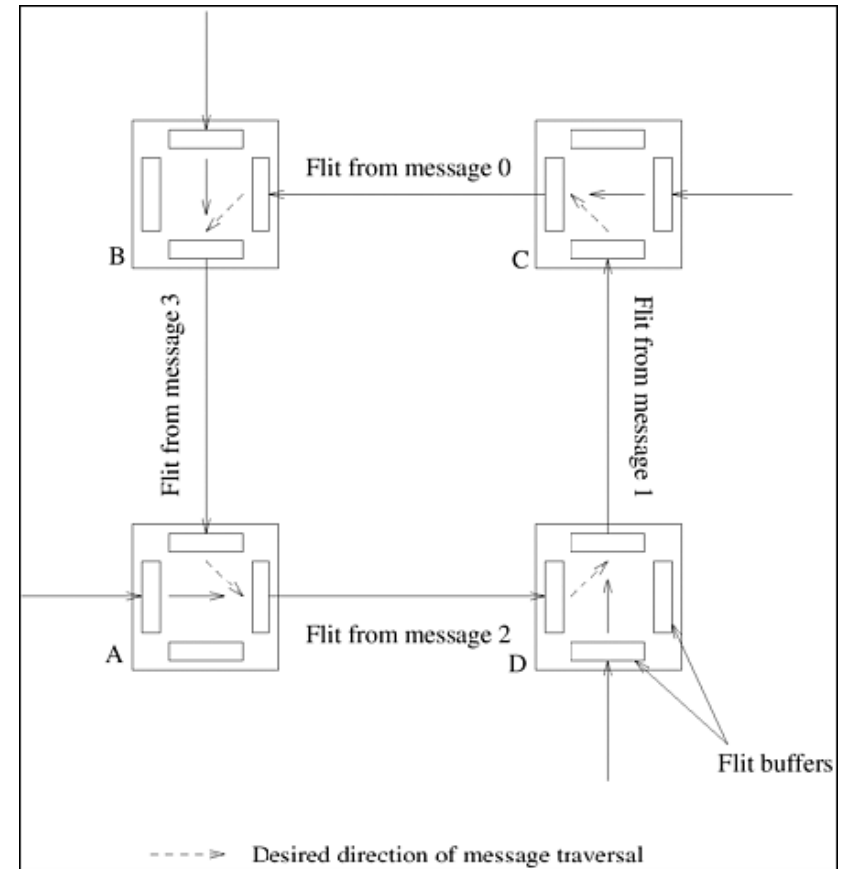
- Scop: reducerea suplimentară a cheltuielilor asociate cu comutarea pachetelor.
 - Forțând toate pachetele să ia aceeași cale, putem elimina cheltuielile generale de transmitere a informațiilor de rutare cu fiecare pachet.
 - Prin forțarea livrării în secvență, informațiile de secvențare pot fi eliminate.
 - Prin asocierea informațiilor despre erori la nivelul mesajului, mai degrabă decât la nivelul pachetelor, se poate reduce cheltuielile generale legate de detectarea și corectarea erorilor.
 - Deoarece ratele de eroare în rețelele de interconectare pentru mașini paralele sunt extrem de mici, mecanismele de detectare a erorilor slabe pot fi utilizate în locul schemelor de corectare a erorilor costisitoare.
- Aplicand aceste optimizari:
 - Un mesaj este împărțit în unități cu dimensiuni fixe numite cifre de control de debit sau flit.
 - Fliturile nu conțin capetele de pachete => mult mai mici decât pachetele.
 - Un trasor este trimis de la sursă la nodul de destinație pentru a stabili o conexiune.
 - Odată stabilită o conexiune, fișierele sunt trimise una după alta.
 - Toate flit-urile urmează aceeași cale într-o manieră columbofilă.
 - Nodul intermediar nu așteaptă să apară întregul mesaj înainte de a-l trimite.
 - Imediat ce un flit este primit la nodul intermediar, acesta este trecut pe următorul nod.
- Nu este necesar un buffer la fiecare nod intermediar pentru a stoca întregul mesaj.
=> rutarea cut-through folosește mai puțină memorie la nodurile intermediare și este mai rapidă.

Costul rutarii cut-through

- Presupunere:
 - mesajul parcurge l legături,
 - th este timpul per-hop => antetul mesajului necesită timp lt_h pentru a ajunge la destinație.
 - mesajul are m cuvinte => întregul mesaj ajunge în timp $t_w m$ după sosirea antetului mesajului.
- Timpul total de comunicare este
$$\underline{t_{comm} = t_s + lt_h + t_w m.}$$
- Îmbunătățire față de rutare stochează și înaintea
- Dacă comunicarea este între vecinii apropiați (adică $l = 1$), sau dacă dimensiunea mesajului este mică, atunci timpul de comunicare este similar pt. stocare și înaintare
- Majoritatea computerelor paralele actuale și multe LAN-uri acceptă rutarea cut-through.
 - Mărimea unui flit este determinată de o varietate de parametri de rețea.
 - Circuitul de comandă trebuie să funcționeze la viteza de flit.
 - Selectând o dimensiune de flit foarte mică, pentru o lățime de bandă a unui link dat, rata de flit necesară devine mare.
 - Pe măsură ce dimensiunile flit devin mari, dimensiunile tamponului intern cresc (și latența transferului de mesaje)
 - Mărimile de viteză în rețelele de interconectare recente sunt cuprinse între patru biți și 32 octeți.
- În multe paradigme de programare paralele care se bazează în principal pe mesaje scurte (cum ar fi liniile de cache), latența mesajelor este critică.
- Routerile utilizează rutare de tip multilane.
 - În rutarea cut-through multilane, un singur canal fizic este împărțit într-un număr de canale virtuale.

Impasuri in rutarea cut-through

- Dacă un mesaj trebuie să utilizeze o legătură care este în prezent utilizată, atunci mesajul este blocat.
 - Acest lucru poate duce la un impas.
- Fig. ilustrează un blocaj într-o rețea de rutare de tip cut-through.
 - Destinațiile mesajelor 0, 1, 2 și 3 sunt A, B, C și D, respectiv.
 - Un flit din mesajul 0 ocupă legătura CB (și tamponul asociat).
 - Deoarece legătura BA este ocupată de un flit din mesajul 3, flit-ul de la mesajul 0 este blocat.
 - În mod similar, flit-ul din mesajul 3 este blocat de când este utilizată legătura AD.
 - Niciun mesaj nu poate progresa în rețea și rețeaua este blocată.
- Poate fi evitată folosind tehnici de rutare adecvate și tamponul de mesaje.



Reducerea costului

1. Comunică în vrac:

- În loc trimiterii de mesaje mici cu un cost de pornire pentru fiecare, agregează mesaje mici într-un singur mesaj mare și amortizează latența de pornire pe un mesaj mai mare.
- Deoarece pe platformele tipice, precum clusterelor și mașinile care transmit mesaje, valoarea t_s este mult mai mare decât t_h sau t_w .

2. Minimizarea volumului de data.

- Pentru a minimiza surplusul pentru transferul fiecare cuvânt t_w , este de dorit să reducem cât mai mult volumul de date comunicate.

3. Minimizarea distanței de transfer a datelor.

- Reducerea la minimum numărul de salturi pe care trebuie să le parcurgă un mesaj.
- 2 relativ ușor, 3 este dificil
 - În MPI, programatorul are puțin control asupra mapării proceselor asupra procesoarelor fizice.
 - Multe arhitecturi se bazează pe rutarea randomizată (în doi pași),
 - Timpul per hop (t_h) este de obicei dominat fie de latența de pornire (t_s) pentru mesaje mici sau pe componente per-cuvant ($t_w m$) pentru mesaje mari.

Simplificarea modelului de cost

- Costul transferului unui mesaj între două noduri dintr-o rețea este dat de:

$$t_{comm} = t_s + t_w m$$

- Este nevoie de aceeași perioadă de timp pentru a comunica între orice pereche de noduri => corespunde unei rețele complet conectate.
- În loc să proiectăm algoritmi pentru fiecare arhitectură specifică (o plasă, un hiper cub sau un arbore), proiectăm algoritmi cu acest model de costuri în minte și îl portăm la orice computer paralel țintă.
 - Pierderea exactității (sau fidelității) predicției atunci când alg. este portat din modelul simplificat (pentru o rețea complet conectată) la un arh. de mașină efectivă.
 - Dacă presupunerea inițială că t_h este de obicei dominat de t_s sau t_w este validă, atunci pierderea de precizie ar trebui să fie minimă.
- Valabil numai pentru rețelele necongestionate.
 - Arhitecturile au praguri diferite pentru momentul în care sunt congestionate;
 - un tablou liniar are un prag mult mai mic pentru congestie decât un hiper cub.
- Valabil numai atâta timp cât modelul de comunicare nu congestionează rețeaua.
 - Diferite modele de comunicare congestionează o anumită rețea în diferite zone.

Efectele congestiei asupra costului comunicatiei

- Fie o grila 2D cu $\sqrt{p} \times \sqrt{p}$ noduri.
 - Deoarece nu există legături în rețea pentru mai multe comunicări, timpul pentru această operațiune este $t_s + t_w m$ (conform cu modelul simplificat).
- Fie un scenario în care fiecare nod comunică cu un nod selectat la întâmplare.
 - Această aleatorie implică faptul că au loc comunicări $p / 2$ (sau $p / 4$ comunicări bidirecționale) în orice echi-partitie a masinii.
 - Unele legături ar trebui acum să poarte cel puțin $\sqrt{p} / 4$ mesaje pe canale de comunicare bidirecționale.
 - Dacă fiecare mesaj are dimensiunea m , timpul pentru această operație este cel puțin $t_s + t_w m \times \sqrt{p} / 4$ (nu este conform cu modelul simplificat).
- ⇒ Sarcina modelării costurilor de comunicare depinde nu doar de arhitectură, ci și de sablonul de comunicare.
- ⇒ Pentru a rezolva acest lucru, se introduce noțiunea de *lățime de bandă eficientă*.
 - Pentru tiparele de comunicare care nu congestionează rețeaua, este identică cu lățimea de bandă a legăturii.
 - Pentru operațiunile de comunicare care congestionează rețeaua este lățimea de bandă a legăturii redusă în funcție de gradul de aglomerare pe cea mai congestionată legatura.

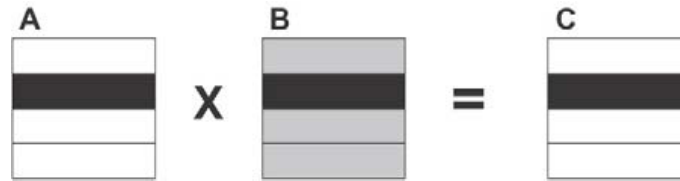
Model de performanță care să dovedească scalabilitatea MPP

- Obiectivele rețelei de comunicare a unui MPP:
 - ofera o comunicare rapida, bine echilibrata in numărul și cu performanța procesoarelor, omogen.
 - nu ar trebui să provoace nici o degradare a vitezei de comunicare chiar și atunci când alte procesoare efectuează simultan operațiuni intense de transfer de date.
 - ar trebui, de asemenea, să asigure aceeași viteză de transfer de date între oricare două procesoare MPP.
- Studiu de caz: multiplicarea matriceala paralela
 - $C = A \times B$ pe un p -procesor MPP, unde A, B sunt matrice dense patratice $n \times n$,
 - multiplicarea seriala necesita $O(n^3)$ operatii.
 - Timpul de executie paralela este

$$\left| t_{\text{comp}} = \frac{t_{\text{proc}} \times n^3}{p}, \right|$$

unde t_{proc} caracterizează viteza unui singur procesor.

Înmulțirea matrice-matrice cu matricile împărțite uniform într-o singură dimensiune



- Each element c_{ij} in C is computed as $c_{ij} = \sum_{k=0}^{n-1} a_{ik} \times b_{kj}$.
- The A , B , and C matrices are evenly (and identically) partitioned into p horizontal slices (for simplicity we assume that n is a multiple of p). There is one-to-one mapping between these slices and the processors. Each processor is responsible for computing its C slice (see Figure).
- In order to compute elements of its C slice, each processor requires all elements of the B matrix. Therefore, during the execution of the algorithm, each processor receives from each of $p - 1$ other processors n^2/p matrix elements (shown in gray in Figure).

Multiplicarea matriceala

- Timpul total de executie

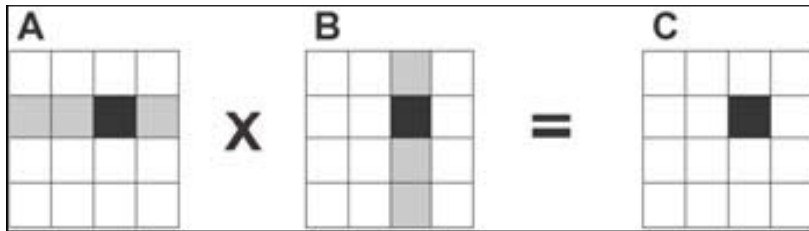
$$t_{\text{total}} \approx t_{\text{proc}} \times \frac{n^3}{p} + t_s \times p + t_e \times n^2.$$

- Scalabilitate: cum să se execute mai rapidă a algului pe un $(p + 1)$ - proc. configurație în comparație cu configurația procesorului p ($p = 1, 2, \dots$)?
 - dacă t_{total} este o funcție în scădere monotonă a p , adică dacă

$$n^3 > 2 \times \frac{t_s}{t_{\text{proc}}} + \frac{t_e}{t_{\text{proc}}} \times n^2$$

Inegalitatea de mai sus va fi adevărată dacă n este rezonabil mai mare decât p .

Descompunere bi-dimensionala



- Each element c_{ij} in C is computed as $c_{ij} = \sum_{k=0}^{n-1} a_{ik} \times b_{kj}$.
- The A , B , and C matrices are identically partitioned into p equal $n/\sqrt{p} \times n/\sqrt{p}$ squares so that each row and each column contain \sqrt{p} squares (for simplicity we assume that p is a square number and n is a multiple of \sqrt{p}). There is one-to-one mapping between these squares and the processors. Each processor is responsible for computing its C square (see Figure 4.3).
- To compute elements of its C square, each processor requires the corresponding row of squares of the A matrix and column of squares of the B matrix (shown in gray in Figure 4.3). Therefore, during the execution of the algorithm, each processor receives from each of its $\sqrt{p} - 1$ horizontal and $\sqrt{p} - 1$ vertical neighbors n^2/p matrix elements.

- Timpul total de executie in caz paralel,

$$t_{\text{total}} \approx t_{\text{proc}} \times \frac{n^3}{p} + 2 \times t_s \times \sqrt{p} + 2 \times t_e \times \frac{n^2}{\sqrt{p}},$$

- Algoritmul este eficient și scalabil pentru orice dimensiune rezonabilă a sarcinilor și numărul de procesoare.

Costuri de comunicare în mașini cu spațiu de adresă partajată

- Motive de dificultate:
 - Disponibilitatea memoriei este de obicei determinată de sistem.
 - Dimensiunile fine ale cache-urilor pot duce la zdrobirea cache-ului.
 - Surplusurile asociate operațiilor de invalidare și actualizare sunt greu de cuantificat.
 - Localitatea spațială este dificil de modelat.
 - Pregătirea poate juca un rol în reducerea cheltuielilor aeriene asociate cu accesul la date.
 - Partajarea falsă este adesea o importanță importantă în multe programe.
 - Disputa în accesari partajate este adesea o contribuție majoră a surplusului.
- Construirea acestora într-un model de cost unic are ca rezultat un model care este
 - prea greoaie pentru a proiecta programe
 - prea specifice mașinilor individuale pentru a fi în general aplicabile.
- Un model simplificat ține seama de accesul la date la distanță, dar nu modelează o varietate de alte cheltuieli generale.

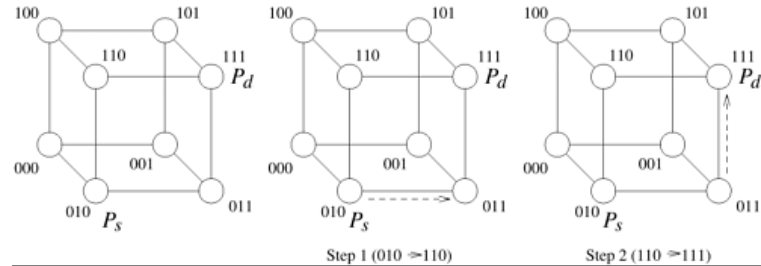
Mecanisme de rutare pentru rețele de interconectare

- Un mecanism de rutare:
 - Determină calea pe care un mesaj o parcurge prin rețea pentru a ajunge de la sursă la destinație.
 - Este nevoie de intrare de nodurile sursă și destinație ale mesajului.
 - De asemenea, poate utiliza informații despre starea rețelei.
 - Întoarce una sau mai multe căi prin intermediul rețelei de la sursă la destinație.
- Clasificare bazată pe selecția rutei:
 - Un mecanism de rutare minim:
 - selectează întotdeauna una dintre cele mai scurte căi între sursă și destinație.
 - fiecare link aduce un mesaj mai aproape de destinația sa,
 - poate duce la congestie în anumite părți ale rețelei.
 - O schemă de rutare non-minimă:
 - poate orienta mesajul pe o cale mai lungă pentru a evita congestionarea rețelei.
- Clasificare pe baza informațiilor privind starea rețelei:
 - O schemă de rutare deterministă:
 - determină o cale unică pentru un mesaj, pe baza sursei și a destinației acestuia.
 - nu folosește nicio informație cu privire la starea rețelei.
 - poate duce la utilizarea neuniformă a resurselor de comunicare dintr-o rețea.
 - O schemă de rutare adaptivă:
 - folosește informații cu privire la starea curentă a rețelei pentru a determina calea mesajului
 - detectează congestia din rețea și rutează mesajele din jurul acesteia.

Rutarea in ordinea dimensiunilor

- Tehnica de rutare minimă deterministă folosită frecvent.
- Alocă canale succesive pentru parcurgerea unui mesaj bazat pe o schemă de numerotare determinată de dimensiunea canalului.
- Pentru o grila 2D este numita rutare *XY*
- Pentru un hiper cub este numit rutare *E-cub*.
- Rutarea *XY*:
 - Fie un tor bi-dimensional.
 - Un mesaj este trimis mai întâi de-a lungul dimensiunii X până când ajunge în coloana nodului de destinație și apoi de-a lungul dimensiunii Y până când ajunge la destinație.
 - Fie P_{S_y, S_x} reprezintă poziția nodului sursă și P_{D_y, D_x} reprezintă cea a nodului de destinație.
 - Schema ar trebui să returneze o cale de lungime $|S_x - D_x| + |S_y - D_y|$.
 - Presupunem ca $D_x \geq S_x$ și $D_y \geq S_y$.
 - Mesajul este transmis prin noduri intermediare P_{S_y, S_x+1} , P_{S_y, S_x+2} , ..., P_{S_y, D_x} de-a lungul dimensiunii X
 - Apoi prin nodurile P_{S_y+1, D_x} , P_{S_y+2, D_x} , ..., P_{D_y, D_x} de-a lungul dimensiunii Y pentru a ajunge la destinație.

Rutare E-cub



- Fie un hipercub d -dimensional cu p noduri.
- Fie P_s si P_d etichetele sursei si destinatiei
- Știm că reprezentările binare ale acestor etichete au o lungime de d biți.
- Distanța minimă dintre aceste noduri este dată de numărul de 1-uri din P_s o P_d , unde o reprezintă operația de sau exclusiv pe biti.
- Nodurile P_s calculează P_s o P_d si trimite mesajul de-a lungul dimensiunii k , unde k este poziția celui mai puțin semnificativ bit non-zero din P_s o P_d .
- La fiecare pas intermediar, nodul P_i , care primește mesajul, calculează P_i o P_d și transmite mesajul de-a lungul dimensiunii corespunzătoare celui mai puțin semnificativ bit care non-zero.
- Acest proces continuă până când mesajul ajunge la destinație.
- Exemplu – Fig.
 - Fie $P_s = 010$ si $P_d = 111$ sursa, respective destinatia.
 - Nodul P_s calculează 010 o $111 = 101$.
 - In primul pas, P_s transmite mesajul de-a lungul dimensiunii corespunzătoare bitului cel mai puțin semnificativ la nod 011.
 - Nodul 011 trimite mesajul de-a lungul dimensiunii corespunzătoare bitului cel mai semnificativ (011 o $111 = 100$).
 - Mesajul ajunge la nodul 111, care este destinația mesajului.

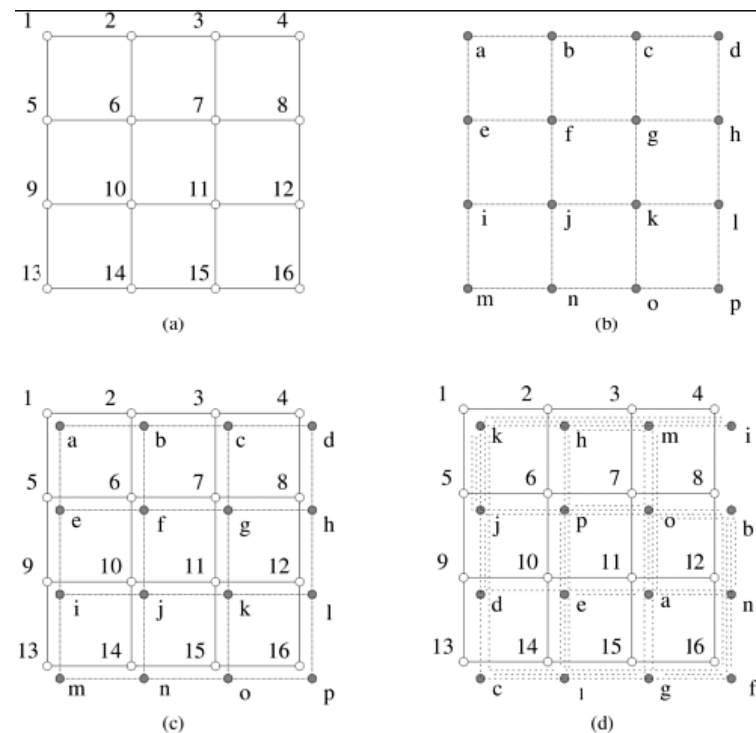
Impactul mapării proces-la-procesor

Problema:

- Adesea, un programator nu are control asupra modului în care procesele logice sunt mapate către nodurile fizice dintr-o rețea.
 - chiar și tiparele de comunicare care nu sunt congestionate în mod inerent pot congestiona rețeaua.

Exemplu – fig.

- (a) arhitectura subiacentă;
- (b) procesele și interacțiunile lor;
- (c) o mapare intuitivă a proceselor către noduri:
 - o singură legătură din arhitectura subiacentă transportă doar date corespunzătoare unui singur canal de comunicare între procese.
- (d) o mapare aleatorie a proceselor către noduri:
 - fiecare legătură asigură până la șase canale de date între procese.
 - timi de comunicare considerabil mai mari dacă ratele de date necesare pe canalele de comunicare între procese sunt mari.

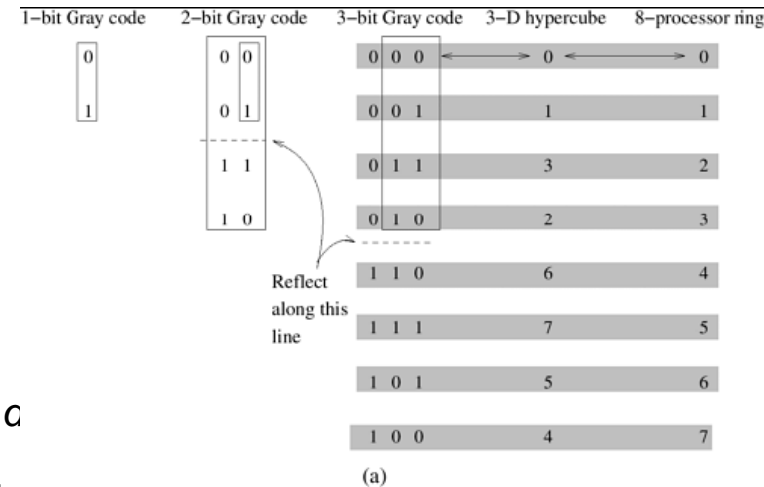


Tehnici de mapare pentru grafuri

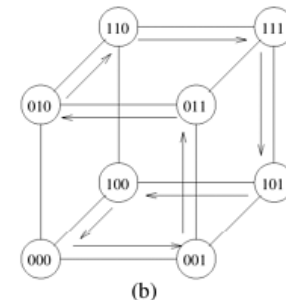
- Date 2 grafuri, $G(V, E)$, $G'(V', E')$, maparea grafului G în graful G'
 - fiecare vârf din setul V pe un vârf (sau un set de vârfuri) în setul V'
 - fiecare muchie din setul E pe o margine (sau un set de muchii) din E' .
- 3 parametrii sunt importanți:
 - Congestionarea mapării:
 - Numărul maxim de muchii din E mapate pe orice muchie din E'
 - este posibil ca mai mult de o muchie din E să fie mapată pe o singură muchie în E' .
 - Dilatarea mapării:
 - Numărul maxim de legături din E' pe care este mapata orice muchie din E .
 - O muchie din E poate fi mapată pe mai multe muchii contigente din E' .
 - Este semnificativ, deoarece traficul pe legătura de comunicare corespunzătoare trebuie să traverseze mai mult de o legătură, contribuind la congestiunea în rețea.
 - Expansiunea mapării:
 - Raportul dintre numărul nodurilor din setul V' și cel din setul V .
 - Seturile V și V' pot conține diferite numere de vârfuri, iar un nod din V corespunde mai multor noduri din V' .
 - Trebuie să fie identică cu raportul dintre procesoarele virtuale și fizice.

Încorporarea unui tablou liniar într-un hipercub

- Un tablou liniar / inel format din 2^d noduri ($0: 2^d - 1$) poate fi încorporat într-un hipercub d -dimensional prin maparea nodului i la nodul $G(i, d)$



- G : codul Gray reflectat binar (RGC).
 - Intrarea $G(i, d)$ este a ia in secventa de coduri Gray de c biti.
 - Codul Gray de $d + 1$ biti sunt derivate din coduri Gray de d biti reflectand tabloul si prefixand intrarile cu un 1, iar intrarile originale cu 0.
 - Intrarile adjunct ($G(i, d)$ si $G(i + 1, d)$) diferă între ele la o singură poziție de biți.
- Nodul i din tabloul liniar este mapat la nodul $G(i, d)$, si nodul $i + 1$ este mapat la $G(i + 1, d) \Rightarrow$ este o legătură directă în hipercub care corespunde fiecărei legături directe din tabloul liniar.
 - Maparea specificată de funcția G are o dilatare de 1 și o congestie de 1.
- Figura (b) : Înglobarea unui inel cu opt noduri într-un hipercub tridimensional.



Încorporarea unei grile într-un hipercub

- Extensie naturală a încorporării unui inel într-un hipercub.

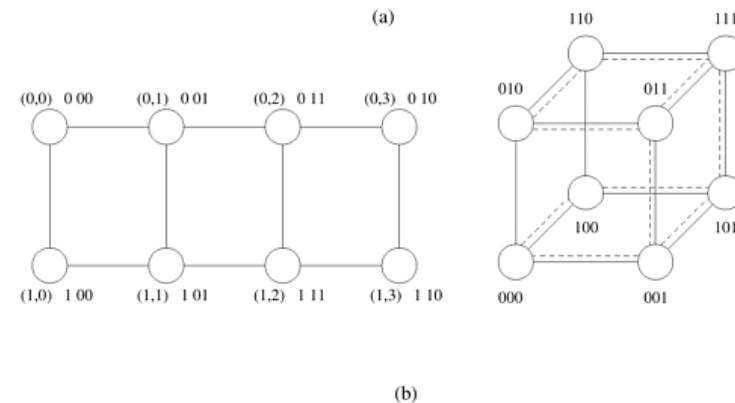
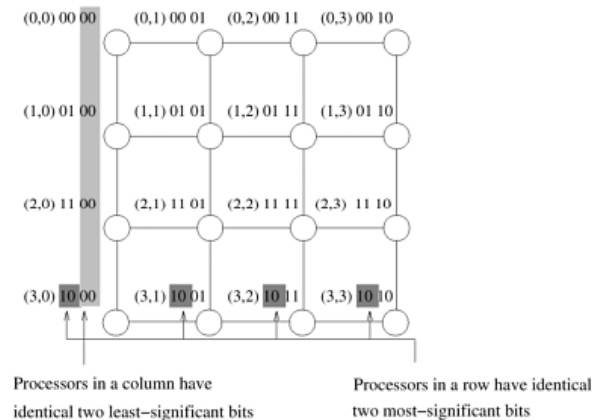
- Incorporarea unui tor bidimensional $2r \times 2s$ într-un hipercub cu $2r+s$ -noduri prin maparea nodului (i, j) a grilei în nodul $G(i, r - 1) || G(j, s - 1)$ a hipercubului ($||$ reprezintă concatenarea a doua coduri Gray).
- Vecinii din tor sunt mapati în noduri hipercub ale căror etichete diferă exact într-o poziție de bit \Rightarrow maparea are o dilatare de 1 și o congestie de 1.

- Exemplu: o grila 2×4 într-un hipercub cu 8 noduri.

- Nodul (i, j) a grilei este mapat în nodul $G(i, 1) || G(j, 2)$ a hipercubului
- Nodul $(0, 0)$ al grilei este mapat în nodul 000 a hipercubului, deoarece $G(0, 1)$ este 0 și $G(0, 2)$ este 00;
- Nodul $(0, 1)$ al grilei este mapat în nodul 001 a hipercubului

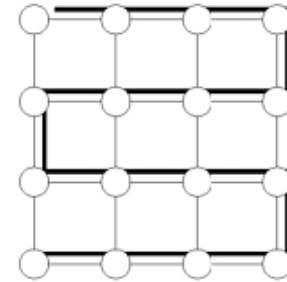
- Figura:

- (a) O grila 4×4 mesh într-un hipercub 4-dimensional
- (b) O 2×4 grila într-un hipercub 3-dimensional.

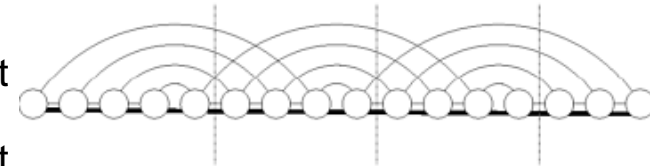


Mapari intre grila si tablou linear

- O mapare intuitiva a unui tablou linear intr-o grila este ilustrata in Fig. (a):
 - liniile solide corespund legăturilor din tabloul linear
 - linii normale la legăturile din grila.
 - congestie de 1, dilatare de 1
- Fie maparea inversa,
 - data o grila, maparea intr-un tablou linear -Fig. (b)
 - liniile solide corespund muchiilor din tabloul linear și liniile normale la muchiile din grila.
 - congestia este 5 – adică, nicio linie solidă nu poartă mai mult de cinci
 - În general: congestionarea acestei mapări (inversă) este $\sqrt{p} + 1$ pentru o mapare generală a p -nodurilor (una pentru fiecare dintre muchiile \sqrt{p} către rândul următor și o muchie suplimentară).
- Se poate mai bine?
 - Congestia oricarei mapari este marginita inferior de \sqrt{p}
 - Vezi textbook
 - In general:
 - limita inferioară la congestionarea unei mapări a rețelei S cu x legături în rețea cu Q legături y este x / y .
 - In cazul de mai sus este $2p/p$, adică 2.



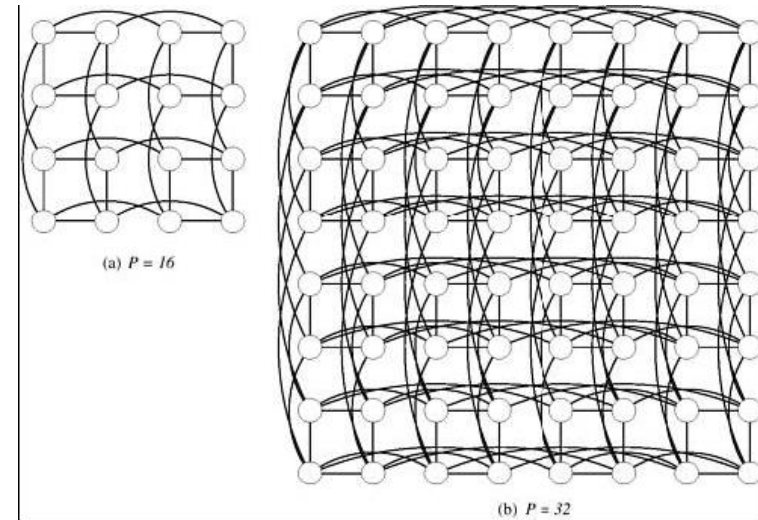
(a) Mapping a linear array into a 2D mesh (congestion 1).



(b) Inverting the mapping – mapping a 2D mesh into a linear array (congestion 5)

Încorporarea unui hipercub într-o grila 2D

- Hipercub cu p -noduri într-o grila cu același număr de noduri, unde p este putere a lui 2.
- Vizualizarea hipercubului ca \sqrt{p} subcuburi, fiecare cu \sqrt{p} noduri.
 - fie $d = \log p$ dimensiunea hipercubului.
 - Cei $d/2$ cei mai puțin semnificativi biți sunt utilizați pentru a defini subcuburile de \sqrt{p} noduri.
 - De exemplu, în cazul unui hipercub 4D, folosim cei doi biți inferiori pentru a defini subcuburile ca fiind (0000, 0001, 0011, 0010), (0100, 0101, 0111, 0110), (1100, 1101, 1111, 1110), și (1000, 1001, 1011, 1010).
- Maparea poate fi definită astfel:
 - Fiecare subcub cu \sqrt{p} noduri este mapat la o linie cu \sqrt{p} noduri ale grilei.
 - Facem acest lucru printr-o simplă inversare a tabloului linie.
 - Congestia: $\sqrt{p}/2$
- Fig: $p = 16$ și $p = 32$
- Mapare pe coloane:
 - Se obține tot o congestie de $\sqrt{p}/2$.



Costuri de performanță

- Observație: este posibilă maparea rețelelor mai dense în rețelele mai reduse cu depășirile de congestie asociate.
 - Exemplu: o grila a carui legaturi sunt mai rapide cu un factor de $\sqrt{p}/2$ decat ale hipercubului pot conduce la performante similar cu acesta => se poate numi grila-grasa.
 - O grila-grasa are aceeași lățime de bandă de bisecție ca un hipercub;
 - Cu toate acestea, are un diametru mai mare.
 - Folosind tehnici adecvate de rutare a mesajelor, efectul distanței nodului poate fi redus la minimum.
- Analiza performanței unei rețele grila & a unui hipercub cu costuri identice:
 - Costul unei rețele este proporțional cu numărul de fire,
 - cost similar.
 - Pentru $p > 16$ și dimensiuni de mesaj suficient de mari, o grilă depășește un hipercub cu același cost.