

2. Sisteme distribuite

Noțiuni și principii

2.1. Justificarea folosirii sistemelor descentralizate și distribuite

2.2 Caracteristicile sistemelor distribuite

2.3 Sisteme cyber-fizice (cyber-physical systems)

Sistemele distribuite sunt implementate pe rețele de calculatoare, iar cele descentralizate pot fi implementate pe un singur calculator.

Descentralizarea sau distribuirea task-ului nu reduce volumul de calcul (procesare), și nici volumul de memorie necesar.

Mai mult, implementarea task-urilor pe o rețea de calculatoare implică și schimbul de informații între diferitele componente => creșterea timpului de execuție (în comparație cu sistemele centralizate)

Întrebare:

Dacă descentralizarea și distribuirea task-urilor duce la creșterea timpilor de execuție, de ce am face asta ?

- resurse
- reactivitate
- etc.

2.1 Justificarea folosirii sistemelor descentralizate și distribuite

Descentralizare

Avantaje:

- Partiționarea calculelor pe diferite componente software (de obicei, specializate), cu scopul de a **reduce complexitatea** algoritmului sau a **volumului de calcul** și **memorie** necesară pe un singur calculator într-un moment dat.
- Achiziția datelor de către componente specializate
- Partiționarea calculelor pe mai multe fire de execuție are avantaje relevante pentru implementarea algoritmilor pe sisteme multiprocesor.

Descentralizarea permite **descompunerea unei probleme** mari într-un set de probleme mai mici care sunt mai ușor de gestionat și rezolvat separat sau/și într-un mod coordonat.

Dezavantajele descentralizării:

- Necesitatea unui (sau mai multor) **coordonatori** – lipsa lor poate duce la conflicte între comportamentele componentelor descentralizate.
- Necesitatea **sincronizării** componentelor pentru a realiza unele operații sau pentru a accesa unele resurse (hardware sau software).

Distribuire

Justificarea (avantajele) folosirii **sistemelor de control distribuit**:

- Sistemul funcționează într-un mediu ostil => erori pentru procesoarele de înaltă performanță. Motivul este deci creșterea **fiabilității**.
- Sistemul este compus dintr-un set de procese distribuite spațial într-o **zonă extinsă**.
- Sistemul funcționează cu **informație distribuită**, achiziționată de la senzori situați în diferite locuri în sistem, informație necesară diferitelor noduri.
- Există nevoi computaționale ridicate (mult mai convenabil pe calculatoare diferite). Motivul este deci lipsa unei **capacități computaționale** suficiente.
- Există un necesar de volum mare de memorie, care nu este disponibil într-o singură unitate de calcul. Motivul este deci lipsa unei capacități de **memorie suficiente**.
- Achiziția datelor și luarea tuturor deciziilor de către o componentă centralizată ar duce la **întârzieri** inacceptabile legate de **transmisia** datelor și a semnalelor de control la distanțe mari.

- Sistemul de control conține componente specializate pentru unele task-uri. Motivul distribuirii este deci **partiționarea funcțiilor**.
- Sistemul trebuie să reacționeze la evenimente care apar la momente de timp diferite, și trebuie să reacționeze imediat. Deci, motivul este existența unor **constrângeri temporale**.

Dezavantajele unui sistem de control distribuit:

- Necesitatea unui sistem de comunicație necesar pentru actualizarea informațiilor în fiecare componentă.
- Schimbul de informații între componente se face cu întârzieri semnificative și variabile relativ la viteza de procesare a datelor.
- Există nevoia sincronizării între diferitele componente software, pentru a putea realiza unele operații sau pentru a folosi unele resurse (hardware sau software)

Sistemul de comunicații reduce fiabilitatea sistemului.

2.2. Caracteristicile unui sistem de control distribuit

Un **sistem distribuit** conține o colecție de **noduri** legate printr-o structură de comunicație.

Un nod are:

- o capacitate de procesare (proprie)
- un ceas propriu
- capacitate de comunicație cu celelalte noduri

Resurse

Deseori, anumite resurse sunt folosite în comun (*shared*), deci utilizarea lor trebuie controlată ➔ ***problemă de alocare a resurselor***

Resursele aparțin:

- sistemelor de calcul (hardware sau software) sau
- sistemelor controlate

Un sistem de control distribuit are și următoarele caracteristici:

- Dificultatea achiziției datelor și a luării deciziilor într-o perioadă scurtă de timp
- Existența unui număr mare de date care sunt distribuite în întreg sistemul și care se modifică fără a fi sincronizate
- Dificultatea de a trimite la timp semnalele de control la distanțe mari.

Este evident că ***nodurile nu au memorie comună***, deci nu o pot folosi pentru comunicație. Dar semafoare distribuite ?

Caracteristicile nodurilor:

- Nodurile nu sunt întotdeauna identice
- Nodurile nu sunt perfect fiabile (sigure) → sistemul totuși rezilient
- Comunicarea dintre noduri implică întârzieri semnificative și nedeterminate.
- Schimbul de informații între noduri este (de obicei) mai lent decât viteza de procesare.
- Sistemul de comunicație nu este perfect fiabil.
- Ceasurile nodurilor nu sunt perfect sincronizate.

Sisteme de timp real (STR)

De obicei, SCD sunt STR

Schimbul de informații depinde de:

- timpii de transfer a informației între calculatoare
- durata de procesare
- timpii de distribuție

Sistemul de procesare este concurent cu SO. Ce implică asta ?

- priorități
- excludere mutuală
- verificare

Evenimente în sisteme distribuite

- antrenat de evenimente (event triggered, event driven)
- antrenat de timp (time triggered, time driven)

Sisteme antrenate de evenimente

Deseori, multiplicarea semnalării evenimentelor (event signal replication) duce la un număr inacceptabil (și nenecesar) de mare de semnale care corespund la aceeași cauză.

Concurența sistemelor antrenate de evenimente

➔ planificator pentru reacția (răspunsul) la eveniment

Planificatorul trebuie să fie on-line și sistemul trebuie să fie preemptiv.

- *Degradare armonioasă (gracious degradation)*

Sisteme antrenate de timp

- se bazează pe un set de presupuneri și șabloane de evenimente (event templates). Trebuie considerat cel mai defavorabil caz pentru încărcarea sistemului.
- acest fel de sisteme sunt potrivite pentru cazul în care se controlează un mediu static cu **dinamicitate redusă**, sau pentru procese repetitive.

Extinderea sistemelor antrenate de timp

- destul de dificilă.
- Planificarea lor este făcută luând în considerare un număr specificat de task-uri care se execută într-o perioadă de timp dată.
- Adăugarea unor noduri noi poate duce la necesitatea realocării duratei pentru diferite activități
- Sistemele antrenate de timp au stare distribuită => probleme, atunci când se adaugă noduri.

Preemptiunea task-urilor mai urgente și mai importante !

Cine face preemptiunea dacă nu există nod central ?

Preemptiunea trebuie să fie eficientă, deci să fie pentru toate activitățile desfășurate de sistemul distribuit.

Preemptiunea se implementează prin schimb de mesaje.

Întârzierile la schimbul de informații (comunicare)

Întârzierile nu pot fi prevăzute.

Uneori depind de lățimea de bandă și de modul de transmisie.

Algoritmii de distribuire și sincronizare pot duce la întârzieri.

Întârzierile:

- **întârzieri constante** – cunoscute apriori și nu se modifică într-o sesiune
- **întârzieri de hazard și independente** – nu pot fi prezise (nu sunt stohastice) și nu pot fi evaluate folosind funcții de distribuție care se bazează pe încărcarea sistemului de comunicație
- **întârzieri stohastice** – pot fi estimate pe baza încărcării sistemului de comunicație

Caracteristici (de justificat pentru trenuri)

- **partajarea (folosirea in comun a) resurselor (resource sharing)**. Resursele pot fi hardware sau software. Pot fi *alocate în mod static sau dinamic*
- **gradul de deschidere** – la extinderi hardware sau software
- **concurența**
 - mulți actori (users/actors) care apelează și interacționează cu sistemul – ei solicită aceleași resurse sau interacționează în diferite moduri cu sistemul
 - unele evenimente apar aproximativ în același timp și sistemul trebuie să reacționeze.
 - informația stocată într-o bază de date trebuie să fie accesată concurent pentru a crește eficiența sistemului.
 - accesul concurent la resursele comune trebuie să fie sincronizat.
- **scalabilitatea** – sistemele distribuite trebuie să funcționeze efectiv și eficient la diferite scale (mărimi). Cel mai mic sistem distribuit: două calculatoare și un server de fișiere. Sistemul și aplicația software nu trebuie să aibă nevoie de modificări atunci când sistemul crește sau scade în mărime.

- **toleranța la defecte** – atunci când apar defecte în hardware sau software, programele pot produce rezultate incorecte sau se pot opri înainte să termine procesarea. Trebuie avut în vedere:
 - **redundanța hardware (hardware redundancy)**
 - **recuperarea software (software recovery)**
 - **sistemele distribuite** trebuie să aibă un grad mare de **disponibilitate** relativ la defectele hardware. Disponibilitatea unui sistem este o măsură a fracției de timp în care este disponibil pentru folosință
- **transparența** – este o ascundere față de actor (utilizator) a separației între componentele sistemului distribuit, astfel încât sistemul este perceput ca un întreg, nu o colecție de componente independente. Referitor la “obiectul informație”:
 - **Transparența la acces** – obiectele locale sau la distanță pot fi accesate folosind operații identice
 - **Transparența de loc** – obiectul informație poate fi accesat fără a cunoaște locația sa
 - **Transparența de concurență** – mai multe procese funcționează concurrent folosind obiecte-informație comune fără a apărea interferență între ele.

- **Transparența multiplicării** – mai multe instanțe ale obiectelor-informație pot fi folosite pentru a crește fiabilitatea și performanța, fără ca utilizatorul (actorul) să fie în cunoștință de cauză
- **Transparența la defect** – se ascund defectele, utilizatorii (actorii) reușind să-și îndeplinească task-urile în pofida eșecului unor componente hardware sau software
- **Transparența migrației** – mișcarea informației în sistem fără ca utilizatorii să fie afectați în vreun fel
- **Transparența performanței** – sistemul poate fi reconfigurat pentru a crește performanța, dacă încărcarea sistemului se modifică
- **Transparența scalării** – sistemul poate crește în dimensiuni fără a modifica fundamental structura sau algoritmi care funcționează intern
- **Flexibilitatea** – Relația dintre componentele fizice și componentele software corespunzătoare.

Probleme legate de **actuatori**:

- sincronizarea actuatorilor distribuiți
- granularitatea temporală a utilizării actuatorilor
- confirmarea (acknowledgement)
- interfațarea actuatorilor

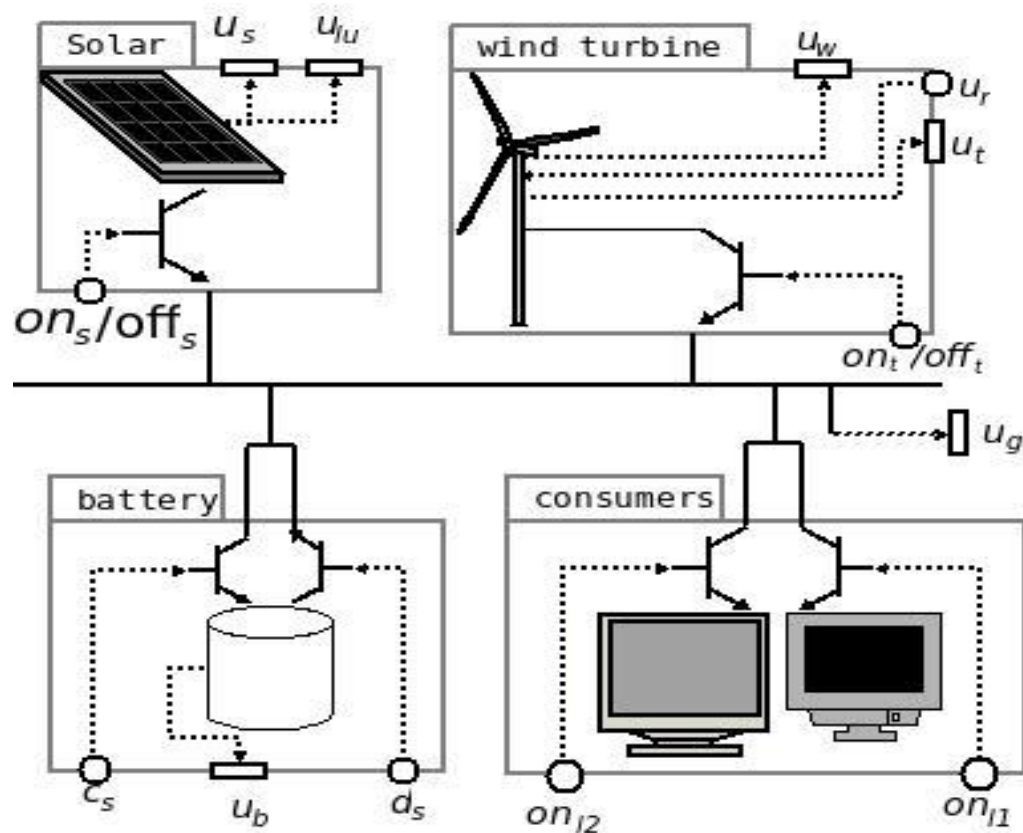
Probleme legate **senzori și traductoare**:

- întârzierea informației provocată de aceste componente
- granularitatea temporală a achiziționării informației
- interfațarea senzorilor și a traductoarelor

2.3 Systeme cyber-fizice (Cyber-Physical)

Definirea sistemelor Cyber-Fizice

Sistemele **cyber-fizice** leagă tipuri diferite de entități (software și hardware) împreună cu actorii umani (utilizatori).



Pot fi privite ca fiind compuse din componente hardware dotate cu senzori și actuatori și de asemenea componente software (de control) care trebuie să reacționeze și să se adapteze la schimbările mediului înconjurător.

De obicei, aplicațiile cu caracteristici cyber-fizice sunt incluse în device-uri fizice. Oamenii nu interacționează direct cu aceste device-uri, ci prin intermediul unei interfețe cyber-fizice (cyber-physical layer) care implementează unele funcții de control complexe.

Componentele de control trebuie să reacționeze la evenimente **discrete, asincrone, interne și externe**, și de asemenea și la schimbarea continuă a unor variabile. În general, specificațiile pentru acest tip de sistem de control hibrid se implementează folosind **task-uri concurente** care comunică între ele.

Caracteristicile CPS

- structura (distribuită, ierarhică, intrări și ieșiri ad-hoc, etc.)
- comportament (concurent, interactive, non-reversibil, etc.)
- comunicare (entități software/hardware, oameni)
- IoT

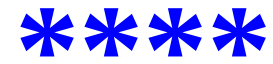
Directii de cercetare

- modelare
- control (ex: control hibrid, etc)
- siguranță
- fiabilitate
- sinteza algoritmilor
- specificare (formală și non-formală)
- proiectare
- implementare (distribuită, mobile, cloud, sisteme multi-agent, rutare, migrarea task-urilor, comunicații mobile distribuite, etc.)
- verificare (real-time, analiza concurenței, stări reachable/unreachable, alocarea resurselor, constrângeri, proprietăți non-funcționale, etc)
- testare

Abordări ale sistemelor cyber-fizice

Subiectele principale:

- **specificarea**
- **modelarea**
- **controlul → sisteme hibrid**
- **proiectarea → bazată pe modelare**
- **implementare**
- **comunicația**
- **siguranța**
- **securitatea**
- **verificarea**
- **testarea**



Sfârșit

