

TEHNOLOGII CLIENT-SERVER

Dr. mat. Florin Păunescu

Global Services and Systems - Glossy S.A.

Din punct de vedere al raportului dintre producătorii (furnizorii) de produse și servicii din domeniul tehnologiei informaticii (TI) și utilizatorii acestora, se constată o basculare spre aceștia din urmă. Această basculare se manifestă printr-o schimbare de atitudine a utilizatorilor față de oferta de produse informatice: utilizatorii solicită de la producători/furnizori soluții/rezolvări pentru problemele lor, nu calculatoare și programe care să fie utilizate în soluționarea unor cerințe/probleme specifice.

Aceste soluții trebuie să fie eficiente sub aspectul raportului performanță/cost, cu o interfață om/calculator convivială și transparentă față de complexitatea arhitecturală și/sau structurală.

În plus, utilizatorii solicită ca soluțiile oferite să le garanteze investițiile făcute și să le asigure independența în raport cu producătorul/furnizorul ales la un moment dat.

Această schimbare de atitudine a utilizatorilor este crucială pentru că ea cuprinde implicit, unele aspecte care țin de arhitectura, flexibilitatea, caracteristicile de performanță și extensibilitatea (deschiderea) soluției, care anterior erau mai puțin specificate sau chiar neglijate. Termenul de extensibilitate, care exprimă potențialul unei soluții de a se adapta unui context dinamic și evolutiv fără a afecta opțiunile arhitecturale și indicii de performanță, completat cu cerința ca aceasta să poată fi realizată pe baza produselor achiziționate de la furnizori diferiți, conferă caracterul "deschis" al soluției.

Aplicațiile sau categoriile de aplicații au, prin prisma obiectivelor urmărite, cerințe specifice privind caracteristicile arhitecturală și de performanță ale sistemelor de calcul suport.

De asemenea, este necesar ca aceste aplicații să comunice cu alte aplicații și să evolueze în timp, o dată cu realitatea informatizată și/sau modificarea contextului tehnologic.

Deschiderea aplicațiilor, în sensul adaptării funcționale la modificările sau cerințele externe cerințelor și/sau al comunicării cu alte aplicații, nu poate fi asigurată cu sisteme suport particulare specializate. Pentru a fi posibilă concepția unor aplicații flexibile și deschise este necesar ca sistemul suport să prezinte un nivel de generalitate și de

standardizare, corespunzător gradului de flexibilitate și de deschidere dorit.

În primul rând, arhitectura unui sistem suport pentru aplicații ce se doresc flexibile și deschise, ele însele trebuind să fie modulare și deschise, în sensul conformanței cu un anumit model care poate fi considerat o referință a compatibilității. Această compatibilitate trebuie înțeleasă în cazul unui sistem modular structurat pe niveluri de abstractizare, ca o standardizare a interfețelor de acces la serviciile oferite de un anumit nivel și a protocoalelor de comunicație între obiectele distribuite, aparținând aceluiași nivel. Cerința de standardizare nu impune nici un fel de restricții asupra implementărilor pe elemente de prelucrare care, la rândul lor, pot fi de tipuri diferite.

Standardizarea interfețelor de acces și a protocoalelor de comunicații asigură deschiderea sistemelor cu prelucrare distribuită și, în mod potențial, condițiile de dezvoltare a aplicațiilor deschise. Standardizarea implică însă un înalt grad de complexitate și prin aceasta o penalizare a caracteristicilor de performanță și a costurilor de întreținere. Într-adevăr, în cazul arhitecturilor pe niveluri de abstractizare, fiecare nivel introduce un cost suplimentar (informații adăugate mesajelor expediate, întârzierii datorate prelucrărilor ocazionate de expedierea/primirea și tratarea mesajelor). În ipoteza că se consideră justificată adoptarea unei variante arhitecturale generale, dublată de standardizarea corespunzătoare, efortul trebuie îndreptat spre implementarea acestei arhitecturi. Rolul implementării este important, deoarece, pentru funcționalități identice, delimitate în faza de proiectare, pot fi atinse performanțe substanțiale diferite pentru diverse variante de implementare. Aplicarea unei metodologii coerente, bazată pe principiile ingineriei software, conduce la implementări coerente și uniforme.

În afară de îmbunătățirea strategiilor și tehnicilor de implementare a unei arhitecturi generale și deschise, se pot aplica și unele metode simplificatoare pentru diminuarea impactului generalității și standardizării.

În cazul arhitecturilor stratificate se pot, de exemplu, reduce sau transparentiza unul sau mai multe niveluri. Acest lucru trebuie aplicat în limitele păstrării potențialului de capacitate sau deschidere, cerut de evoluția previzibilă a sistemului sau aplicației. De asemenea, în contextul oferit de o arhitectură generală se pot concepe și implementa soluții speciale/specifice, care să exploateze unele aspecte particulare cunoscute ale aplicațiilor.

De remarcat consacarea modelului de calcul de tip producător-consumator ("client-server"), pe baza căruia se pot implementa soluții de integrare a sistemelor de calcul eterogene, pentru a constitui, din punctul de vedere al utilizatorului, sisteme deschise. O aplicație deschisă trebuie să fie conformă cu paradigmele sistemului de calcul: interoperabilitatea

componentelor fizice și logice, standardizarea la toate nivelurile de abstractizare (în particular, la nivelurile superioare ale mediului aplicațiilor: interfețe grafice, comunicații, gestiunea datelor) și portabilitate (asigurată de mediul aplicațiilor, nu de cel de exploatare).

monitoare de tranzacții și gestionare de resurse (SGBD, fișiere ISAM, etc.) să comunice și să coopereze în cadrul aceleiași tranzacții; permiterea înlocuirii unui monitor de tranzacții sau a unui SGBD printr-un produs care respectă aceleași interfețe, fără reconsiderarea concepției și implementării aplicațiilor.

a. Aplicații RDBMS Client / Server



b. Aplicații OLTP Client / Server

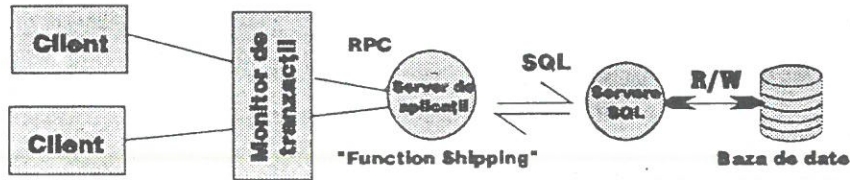


Figura 1.

Există două tipuri principale de modele de tehnologii client-server: distribuirea datelor ("data fetching", SQL - figura 1.a, aplicabil în contextul oferit de sistemele de gestiune a bazelor de date - SGBD); distribuirea prelucrărilor ("function shipping", RPC - figura 1.b, specific modului oferit de monitoarele de tranzacții - OLTP).

Aceste caracteristici fac obiectul standardului DTP ("Distributed Transaction Processing" definit de X/Open). Cea mai mare parte a monitoarelor tranzacționale sub UNIX implementează aceste concepte și interfețe asociate: TUXEDO (UNIX System Laboratories - Novell), Top End (NCR), CICS/6000 (IBM), Encina (Transarc), UTM (SNI).

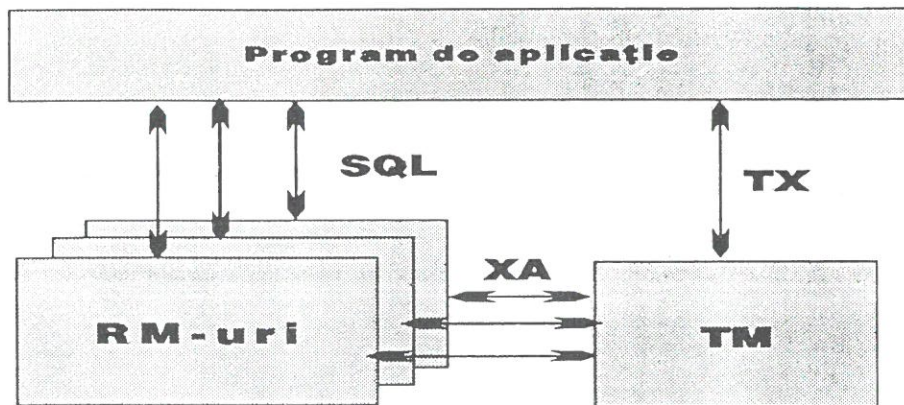


Figura 2.

Față de modelul de aplicații client-server în context SGBD, modelul de tip OLTP asigură eterogenitatea, transparența distribuției și scalabilității, performanțe ridicate.

Obiectivele principale ale modelului client-server global sunt: asigurarea portabilității codului sursă al aplicațiilor; oferirea posibilității diferitelor

În ceea ce privește sistemele de gestionare a bazelor de date, există sisteme ca Oracle, Informix etc. care integrează deja biblioteca de funcții X/Open care permite tranzacții globale standardizate.

Modelul funcțional al tranzacției globale DTP este descris în figura 2; semnificația prelucrărilor folosite este următoarea: AP (Application Program);

program de aplicație; RM (Resource Manager): baza de date, sistem de fișier; TM (Transaction Manager): gestiunea tranzacțiilor; XATMI (X/Open Application Transaction Manager Interface): API pentru apelul de servicii de către aplicație; XA (Transaction): API pentru gestiunea tranzacțiilor TM; TX (Transaction Demarcation): API pentru delimitarea tranzacțiilor de către aplicație.

Din punct de vedere al elaboratorului de aplicații, acest model presupune utilizarea mai multor tipuri de interfețe pentru gestiunea interacțiunilor dintre principalele funcțiuni tranzacționale: interacțiunea dintre aplicație și monitorul de tranzacții se realizează prin intermediul unei interfețe unice, denumită TX, publicată de X/Open în 1992; interacțiunea dintre aplicații și date se realizează prin intermediul interfețelor SQL sau ISAM; interfețe de acces a unei tranzacții la o resursă partajată din rețea (XATMI, TXRPC sau de tip "Peer-to-Peer CPI-C"); interfața XA, care permite monitorului de tranzacții să coordoneze tranzacții executate pe mai multe SGBDR-uri în cadrul unei tranzacții globale unice.

TUXEDO (Novell/UNIX Systems Laboratories) este cel mai răspândit sistem tranzacțional

distribuit care respectă principiile de bază ale unei tranzacții globale: proprietățile de ACID-itate ("Atomicity", "Coherence", "Isolation", "Durability") ale unei tranzacții: atomicitate, coerență, integritate și remanență (durabilitate).

Algoritmii implementați sunt capabili să garanteze fie execuția corectă a tranzacției, fie abandonarea acesteia cu anularea acțiunilor deja întreprinse și deci păstrarea coerenței bazei de date.

Experimentele privind utilizarea sistemului tranzacțional TUXEDO și Oracle (RM), sub UNIX SVR.4.1./UnixWare (pentru platforme Ix86), realizate la ComputerLand, demonstrează viabilitatea unor soluții performante și mai economice pentru probleme care presupun tranzacții în timp real distribuite.

Exemple de aplicații implementate pe baza tehnologiei descrise mai sus sunt:

- Sistem de verificare a pașapoartelor la trecerea frontierei în Aeroportul Otopeni;
- Sistem de informare conferințe și expoziții exploatat în timpul Forum-ului de la Crans-Montana, Expoziția CERF'94 și Conferința ROSE'94;
- Sistem de gestiune a resurelor umane cu aplicabilitate în Administrația publică, industrie, etc.

Prin prisma obiectivelor societății informaționale globale, tehnologiile client-server tranzacționale oferă un potențial tehnic capabil să asigure transpunerea în viață a acestora.