

Prelucrarea interogărilor în baze de date distribuite

Radu Țigănescul-Amariții, Daniel-Gabriel Chiruță, Mirela Danubianu

Abstract—Distributed databases in Oracle have some performance issues. These performance issues can occur when a query requests data from distributed databases that are slower than standalone databases. In this article, I'll present the results of a series of queries executed on distributed databases.

Index Terms—distributed databases, distributed queries.

I. INTRODUCERE

Sistemele de baze de date au produs mutații semnificative în maniera de prelucrare a datelor, făcând trecerea de la metoda în care fiecare aplicație își definea și întreținea propriile sale date, la una în care datele sunt definite și sunt gestionate centralizat. Evoluția tehnologiei informației, a atras după sine și o evoluție spectaculoasă în ceea ce privește sistemele de baze de date, astfel încât, tehnologia bazelor de date distribuite a permis schimbarea modului de lucru centralizat într-unul descentralizat.

În esență o bază de date reprezintă un ansamblu de date integrat, anume structurat și dotat cu o descriere a acestei structuri. Descrierea structurii poartă numele de dicționar de date sau metadata și are menirea de a crea independența dintre datele aflate în baza de date și aplicațiile care le gestionează[2].

Baza de date poate fi interpretată ca o colecție de fișiere interconectate care conțin nucleul de date necesar unui sistem informatic. Diferitele obiecte din cadrul realității sunt denumite clase sau entități. Baza de date se constituie ca un ansamblu intercorelat de colecții de date, prin care se realizează reprezentarea unei realități[2].

Un sistem de baze de date distribuit permite aplicațiilor să acceseze date din baze de date locale și baze de date aflate la distanță. Sunt două tipuri de baze de date distribuite: omogene și eterogene[2].

II. BAZE DE DATE DISTRIBUITE OMOGENE

Un sistem de baze de date distribuite omogen este o rețea de două sau mai multe baze de date ce rulează sub același sistem de gestiune a bazelor de date care se află pe una sau mai multe mașini fizice. Figura 1 ilustrează un sistem distribuit care conectează trei baze de date Oracle: *hq*, *mfg* și *sales*. O

aplicație poate accesa sau modifica date simultan din câteva baze de date într-un singur mediu distribuit.

Pentru o aplicație client locația și platforma bazelor de date sunt transparente. Se pot crea și sinonime pentru obiecte aflate la distanță într-un sistem distribuit, astfel încât utilizatorii le pot accesa cu aceeași sintaxă ca obiectele locale.

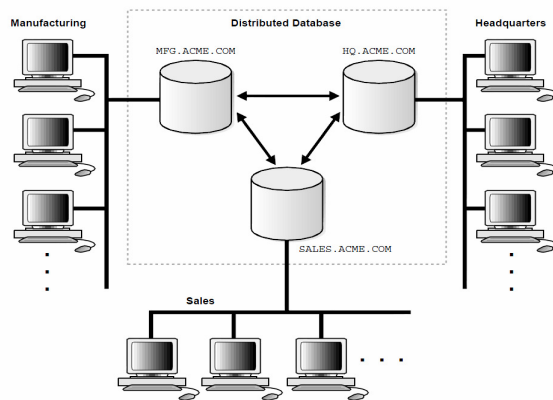


Fig. 1. Bază de date distribuită omogenă (figură preluată din DBA Guide[1]).

III. BAZE DE DATE DISTRIBUITE ETEROGENE

Într-un sistem de baze de date distribuit eterogen (figura 2) există cel puțin două sisteme diferite de gestiune a bazelor de date care rulează în diferitele noduri ale rețelei. În cazul pe care îl considerăm, cel puțin un nod este non-Oracle. Pentru aplicație sistemul de baze de date distribuit eterogen apare ca o bază de date Oracle singulară. Baza de date Oracle locală ascunde distribuția și eterogenitatea datelor.

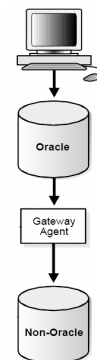


Fig. 2. Bază de date distribuită eterogenă (figură preluată din DBA Guide[1]).

IV. LEGĂTURILE DINTRE BAZE DE DATE DISTRIBUITE

Legătura (link-ul) dintre site-urile unei baze de date distribuite este o conexiune între două servere de baze de date fizice care permite unui client să le acceseze ca o singură bază de date logică.

Link-ul este un pointer care definește o cale de comunicație unidirecțională de la un server de bază de date Oracle către un alt server de bază de date. Link-ul este unidirecțional în sensul că un client conectat la o bază de date locală A, poate folosi un link stocat în baza de date A să acceseze informații într-o bază de date B aflată la distanță, dar utilizatorii conectați la baza de date B nu pot folosi același link pentru a accesa date din A. În figura 3 este prezentată structura unui link care folosește utilizatorul *scott* care accesează tabelul *emp* aflat într-o bază de date la distanță numită *hq.acme.com*

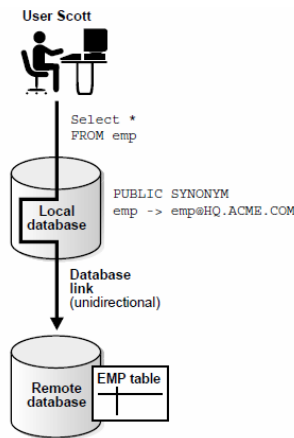


Fig. 3. Structura unui link (figură preluată din DBA Guide[1]).

Link-urile dintre bazele de date distribuite pot fi private, publice sau globale. Dacă sunt private numai utilizatorul care le-a creat are acces, iar dacă sunt publice toți utilizatorii bazei de date au acces, iar în cazul celor globale administratorul poate administra link-urile pentru toate bazele de date. Avantajul acestor link-uri este acela că permite utilizatorilor să aibă acces la obiectele altui utilizator dintr-o bază de date aflată la distanță și beneficiază de privilegiile setate de proprietarul obiectelor.

Crearea link-urilor între baze de date se poate crea cu comanda *CREATE DATABASE LINK* sau în mod vizual prin deschiderea consolei Oracle, se navighează prin arbore până la găsirea subarborelui *Distributed*, se selectează *Database Links*, se efectuează click dreapta pe acesta și se selectează opțiunea *create* cum este în figura 4 și se urmăresc pașii wizard-ului.

După crearea link-ului acesta trebuie activat. Activarea se face cu ajutorul *Oracle Net Configuration Assistant*(figura 5), se selectează opțiunea *Local Net Service Name configuration* și se urmăresc pașii wizard-ului.

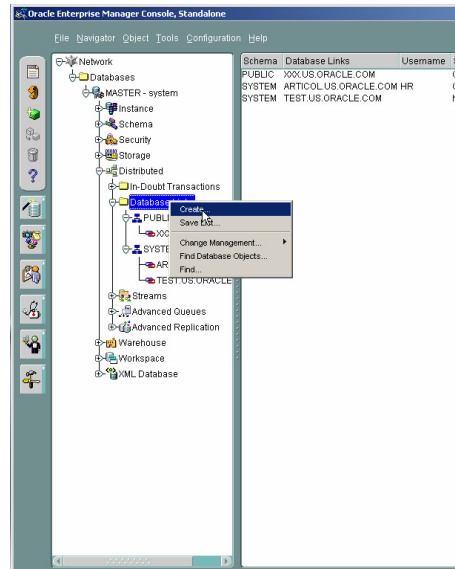


Fig. 4. Crearea unui link în mod vizual.

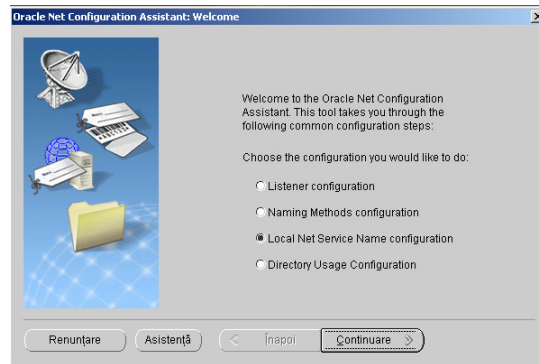


Fig. 5. Oracle Net Configuration Assistant.

V. PRELUCRAREA INTEROGĂRILOR ÎN BAZE DE DATE DISTRIBUITE

În acest capitol vom testa câteva interogări pentru a face o comparație a performanțelor în termenii timpilor de execuție. Pentru aceasta vom utiliza SQL Scratchpad (figura 6).

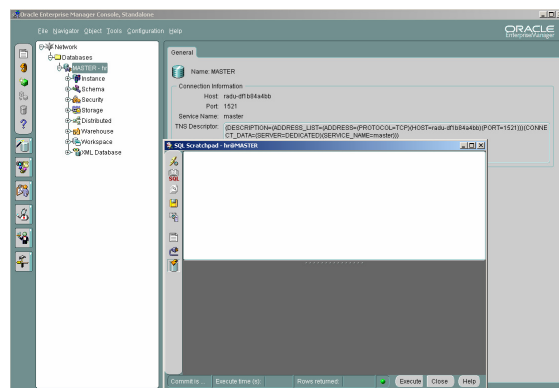


Fig. 6. SQL Scratchpad

Caracteristicile hardware ale sistemului pe care rulează

serverul de baze de date Oracle local sunt:

- Procesor Intel la 2.8 GHz;
- Memorie RAM 324 MB DDR II;

Sistemul de operare este Windows Xp SP 2.

Caracteristicile hardware ale sistemului pe care rulează serverul de baze de date Oracle aflat la distanță sunt:

- Procesor Intel la 2.4 GHz;
- Memorie RAM 2GB;

Sistemul de operare este Windows Server 2003 R2.

Am ales să facem o comparație a performanțelor de prelucrare a interogărilor atunci când sunt adresate date din site-uri diferite și date din același site.

Testul a fost efectuat pe un tabel aparținând unei baze de date locale cu 107 înregistrări, identică cu tabelul plasat în site-ul aflat la distanță.

Pentru început am făcut o uniune între două tabele, unul local și unul aflat la distanță:

```
select last_name, first_name, department_id
  from employees
 union
 select last_name, first_name, department_id
  from employees@xxx;
```

unde xxx reprezintă numele link-ului către baza de date de la distanță.

Timpul de execuție a fost de 4.486 secunde (figura 7).

LAST_NAME	FIRST_NAME	DEPARTMENT_ID
Abel	Ellen	80
Ande	Sundar	80
Atkinson	Mozhe	50
Austin	David	60
Baer	Hermann	70
Baida	Shelli	30
Banda	Amit	80
Bates	Elizabeth	80
Bell	Sarah	50

Fig. 7. Rezultatul uniunii a două tabele.

O variantă optimizată a interogării de mai sus este următoarea:

- Se creează un tabel local temporar în care se vor stoca datele preluate de la baza de date aflată la distanță:

```
create global temporary table tranzactie(
  last_name varchar2(25),
  first_name varchar2(20),
  department_id number(4,0))
on commit preserve rows;
```

- Se preiau datele de la baza de date aflată la distanță și se inserează în tabelul temporar *tranzactie*:

```
insert into tranzactie(last_name, first_name,
  department_id)
  select last_name, first_name, department_id
    from employees@xxx;
```

- Se realizează operația de uniune dintre tabelul temporar *tranzactie* și tabelul local *employees*:

```
select last_name, first_name, department_id
  from employees
 union
 select last_name, first_name, department_id
  from tranzactie;
```

Timpul total de execuție a celor trei operații a fost de 1.232 secunde.

Operația de uniune executată pe serverul local:

```
select e1.last_name, e1.first_name, e1.department_id
  from employees e1
 union
 select e2.last_name, e2.first_name, e2.department_id
  from employees e2;
```

a avut timpul de execuție de 1.382 secunde.

Operația de uniune executată pe serverul aflat la distanță:

```
select e1.last_name, e1.first_name, e1.department_id
  from employees@xxx e1
 union
 select e2.last_name, e2.first_name, e2.department_id
  from employees@xxx e2;
```

a avut timpul de execuție de 15.933 secunde.

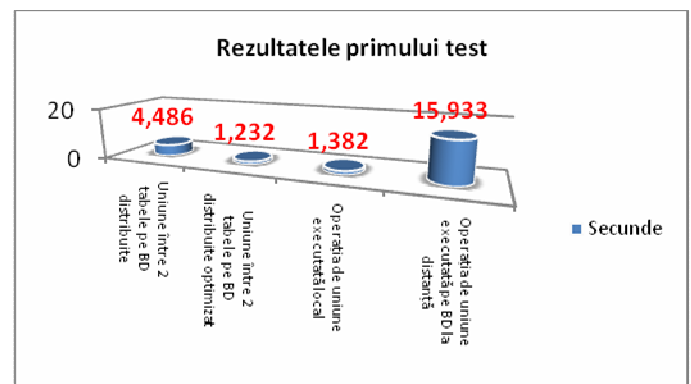


Fig. 8. Rezultatele primului test.

Pentru al doilea test am ales să facem o joncțiune între două tabele din două baze de date distribuite. Tabelul aflat la distanță conține 2247 de înregistrări iar tabelul local conține 107 înregistrări. Interogarea:

```
select x.last_name, x.first_name, x.employee_id
  from hr.copy_empl@xxx x, employees e
 where x.employee_id=e.employee_id;
```

s-a executat în 0.19 secunde.

O altă variantă a acestei interogări poate fi creată cu ajutorul

comenzii */*+driving_site(id)*/*, această comandă are ca efect transmiterea tabelului din baza de date locală către baza de date la distanță. Această metodă este eficace atunci când tabelul din baza de date de la distanță este de dimensiuni mult mai mari decât tabelul de pe baza de date locală astfel evitându-se un trafic prea mare pe rețea. Interogarea este:

```
select /*+driving_site(x)*/ x.last_name, x.first_name,
x.employee_id
from hr.copy_empl@xxx x, employees e
where x.employee_id=e.employee_id;
```

iar durata de execuție a fost de 38.845 secunde.

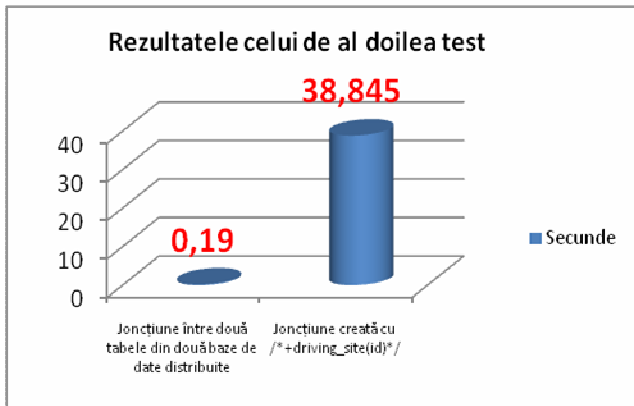


Fig. 9. Rezultatele celui de al doilea test.

În al treilea test vom efectua o inserare într-un tabel aflat în baza de date locală dintr-o bază de date aflată la distanță utilizând */*+driving_site(id)*/*.

```
insert into copy_empl(employee_id, first_name, last_name,
email, phone_number, hire_date, job_id,
commision_pct, manager_id, department_id)
select /*+driving_site (x)*/ x.employee_id, x.first_name,
x.last_name, x.email, x.phone_number, x.hire_date,
x.job_id, x.commision_pct, x.manager_id, x.department_id
from hr.employees@xxx x, employees e
where x.employee_id=e.employee_id;
```

Timpul de execuție a fost de 8.533 secunde.

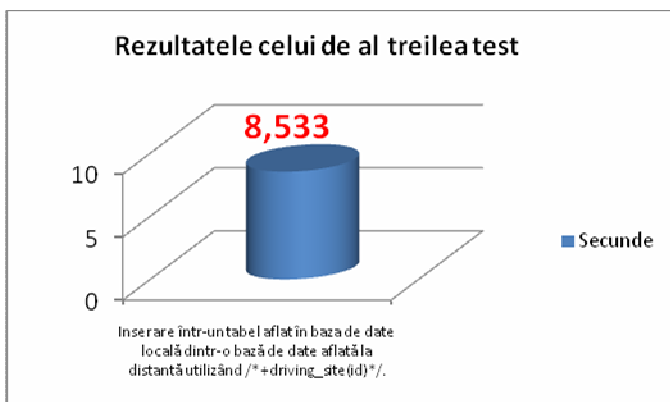


Fig. 10. Rezultatele celui de al treilea test.

VI. CONCLUZII

Bazele de date distribuite au avantaje în creșterea performanțelor sistemului, de reducere a costurilor de prelucrare și transmitere a datelor și reflectă structura oragnizatorică a întreprinderilor.

Dezavantajul bazelor de date distribuite este că prezintă o complexitate ridicată de proiectare, din acest motiv multe organizații nu le implementează recurgând la baze de date centralizate, și adaugă un trafic suplimentar pe rețea pentru vehicularea datelor. La aceste dezavantaje se mai adaugă problemele de administrare:

- ținerea evidenței persoanelor care au acces la bazele de date;
- în cazul unor dificultăți sau a unor aplicații care nu funcționează cum trebuie cum știm care bază de date cauzează probleme ?
- când se creează un nou user pe ce instanță de bază de date să fie creat ?

REFERINȚE

- [1] DBA Guide 9i.
- [2] Clasic și modern în teoria și practica bazelor de date relationale, Mirela DANUBIANU, Ed. InfoData, Cluj-Napoca, 2009
- [3] Baze de date. Proiectare. Gestiune. Implementare. Thomas Connolly, Carolyn Begg, Anne Strachan ; Ed. Teora 2001
- [4] Baze de date relationale – M. Fotache
- [5] Baze de date și gestiunea tranzacțiilor – R. Dollinger, Ed. Albastra