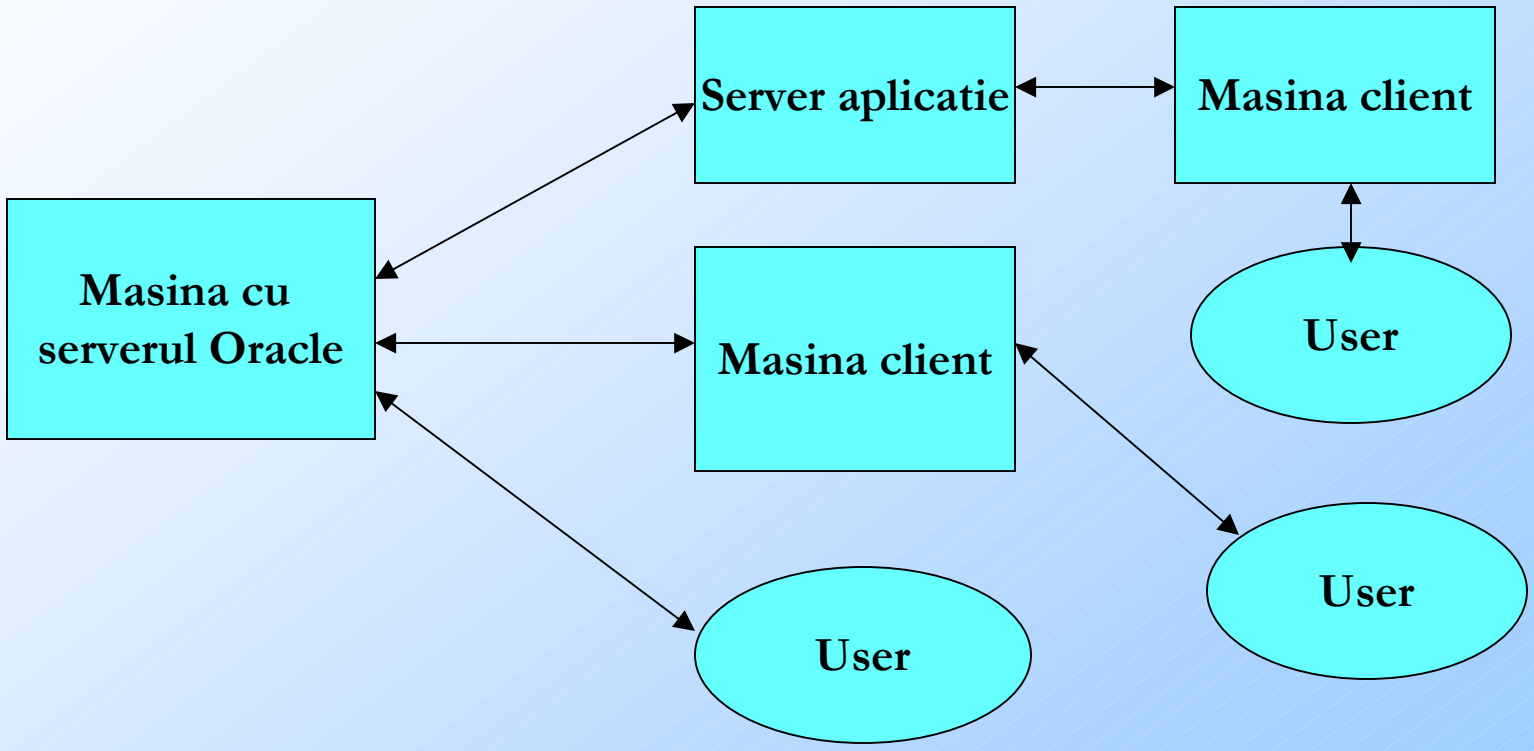
Capitolul 1

Arhitectura Oracle

Serverul ORACLE

- Este un sistem de gestiune a bazelor de date relationale
- Userul poate lucra:
  - Cu un client pe aceeasi masina cu serverul (de exemplu un client SQL*Plus ruland pe aceeasi masina cu serverul Oracle)
  - Clientul ruleaza pe o alta masina decat serverul (two-tiered)
  - Aplicatia userului acceseaza o alta aplicatie iar la randul ei aceasta e in comunicatie cu serverul (three-tiered)
- Iata o figura:

Serverul ORACLE



USER vs UTILIZATOR

- Vom numi in cele ce urmeaza user un cont de utilizator Oracle (exemplu: user ‘student’ cu parola ‘stud1234’
- Utilizator este o persoana care prin intermediul unui cont de user interactioneaza cu Oracle
- Utilizator poate fi numit intr-un sens mai larg si un proces, o aplicatie care foloseste un cont user pentru a interactiona cu Oracle

PROCESE

- Cand userul acceseaza o aplicatie care lucreaza cu Oracle sunt create 2 procese:
  - Proces user pe masina pe care lucreaza utilizatorul. Acesta interactioneaza cu utilizatorul si asigura comunicatia cu cel de-al doilea proces
  - Proces server, pe masina unde este instalat serverul Oracle. Acesta comunica cu serverul Oracle in numele procesului user.

SESIUNE

- O sesiune de lucru reprezinta o conexiune a unui user cu serverul Oracle.
- Un user poate avea mai multe sesini deschise simultan:
  - Pe aceeasi baza de date
  - Cu acelasi cont de user
  - Pe aceeasi masina sau pe masini diferite

PROCESUL USER

- Cum s-a vazut, este creat atunci cand un utilizator incepe o sesiune lucru folosind fie un client care interactioneaza direct cu Oracle (SQL*Plus) fie o unealta Oracle (Server Manager, Developer, etc)
- Ruleaza pe masina utilizatorului
- Asigura interfata - grafica uneori - pentru acesta (GUI, UPI – User Program Interface)
- Procesul se termina cand utilizatorul iese din programul folosit
- Trimite comenzile/cererile utilizatorului catre procesul server si afiseaza rezultatele (date, stare, eventualele mesajele de eroare)

PROCESUL SERVER

- Ruleaza pe masina unde este instalat serverul Oracle
- Deserveste un singur proces user sau mai multe procese user, in functie de configuratie: Dedicated Server sau Multithreaded server)
- Foloseste o zona de memorie (PGA – Program Global Area) care va fi descrisa mai jos.
- Este un intermediar intre procesul user si serverul Oracle, folosind OPI – Oracle Program Interface – pentru a interactiona cu acesta
- Procesul se incheie cand utilizatorul termina sesiunea.

INSTANTA ORACLE

- Un server Oracle consta dintr-o instanta Oracle si o baza de date Oracle.
- Instanta Oracle este compusa dintr-o structura de date in memorie numita SGA – System Global Area – si procese rulate in backgroung care sunt utilizate de Oracle pentru gestiunea bazei de date.
- O instanta Oracle deschide o singura baza de date.
- Este identificata prin ORACLE_SID (variabila la nivel de SO). SID = System ID

SGA – System Global Area

- Este alocata in memoria virtuala a sistemului pe care ruleaza serverul Oracle.
- Contine date si informatii de control
- Este de tip shared – mai multi useri pot folosi datele de aici pentru a se evita accesul repetat la disc
- Contine mai multe componente dintre care cele mai importante sunt:
  - Buffer Cache
  - Shared Pool
  - Redo Log Buffer

SGA – System Global Area -cont

- Mai exista de asemenea:
  - Java Pool – utilizata pentru cod si date specifice Java de catre Java Virtual Machine (JVM)
  - Large Pool – optional, pentru date de de mari dimensiuni.
  - Streams Pool – Folosita de produsul Oracle Streams

Procese BACKGROUND

- Database Writer (DBWR) – Este responsabil cu scrierea blocurilor de date modificate/inserate din bufferele de memorie in fisierele de pe disc. In Oracle 10g pot fi maximum 20 de procese de acest fel
- Log Writer (LGWR) – Scrie datele din Redo Log Buffer pe disc. Scrierea se face secvential intr-un fisier de Redo Log.
- Checkpoint (CKPT) – Acest proces scrie periodic pe disc toate blocurile de date din buffere care au fost modificate. O astfel de actiune este denumita un checkpoint. Procesul anunta actiunea lui DBWR, actualizeaza fisierele de date si control ale bazei de date si inregistreaza momentul in care s-a facut checkpointul.

Procese BACKGROUND - cont

- System Monitor (SMON) - Are functia de verificare consistenta date si sa initieze recuperarea dupa incident atunci cand o instanta Oracle care a avut un incident reporneste.
- Process Monitor (PMON) - Dealoca resursele unui proces care are un incident. Folosit pentru procesele user care au incidente.
- Archiver (ARCV) –Copiaza fisierele Redo Log in arhiva de pe disc atunci cand acestea sunt pline sau cand acestea se schimba. Actiunea are loc in anumite conditii.

Baza de date

- A doua componenta a serverului (pe langa instanta) este Baza de date
- Este identificata prin DB_NAME (variabila SO)
- Oracle recomanda ca instanta si BD sa aiba acelasi nume (pot fi si diferite) pentru usurinta administrarii
- Este compusa din mai multe tipuri de fisiere

Tipuri de fisiere

- Data files – fisiere de date. Fisierele de date contin:
  - dictionarul de date al BD (mai stiti ce este acesta?),
  - obiectele userului (mai stiti care sunt acestea?)
  - Vechile valori ale datelor modificare de tranzactiile curente (before image)
- O baza de date are cel putin un astfel de fisier.

Tipuri de fisiere - cont

- Redo Log Files – Fisiere Redo Log. Contin modificarile facute in baza de date care sunt necesare la reconstructia ei in caz de incident.
- Fiecare baza de date contine cel putin 2 astfel de fisiere care pot fi tinute pe dispozitive de stocare diferite, prevenind pierderi de date in cazul in care un disc este distrus.

Tipuri de fisiere - cont

- Control files – fisiere cu date de control. Contin informatii necesare pastrarii integritatii bazei de date
- Fiecare baza de date are cel putin un astfel de fisier.

Alte fisiere

- Pe langa fisierele bazei de date Oracle mai utilizeaza si alte fisiere, ca de exemplu:
  - Fisier de parametri: contine parametri care caracterizeaza instanta Oracle
  - Fisier de parole: utilizat pentru autentificarea userilor
  - Fisiere Redo Log arhivate: copii offline ale fisierelor Redo.

Etape de procesare cerere

- Exista mai multe etape parcurse de o cerere de la emiterea ei de catre utilizator pana la executia completa. Exemplu pentru o cerere SELECT:
  - Pasul 1: Parse: cererea primita de la user este analizata sintactic si semantic (compilata). Serverul intoarce o informatie de stare (ok sau eroare). Se foloseste ca zona de memorie Shared Pool pentru aceasta operatie

Etape de procesare cerere - cont

- Pasul 2: Execute. Cererea este executata si datele returnate sunt pregatite pentru a fi returnate userului.
- Pasul 3: Fetch. Liniile returnate de cerere sunt trimise userului (procesului user) pentru a fi procesate acolo (afisare sau procesare). In functie de dimensiunea datelor returnate se pot executa una sau mai multe operatii de tip FETCH

SHARED POOL

- Este parte a SGA. Este utilizata si in pasul 1 (parse) de executie a cererii.
- Dimensiunea sa e data de parametrul SHARED_POOL_SIZE (din fisierul de parametrii).
- Pentru Pasul 1 sunt utilizate urmatoarele componente ale Shared Pool:
  - Library cache
  - Data dictionary cache

Library cache

- Stocheaza informatii despre cele mai recente cereri SQL utilizate:
  - Textul cererii
  - Arborele cererii rezultat in urma etapei Parse (analiza semantica). Acesta este versiunea compilata a textului cererii
  - Planul de executie al cererii rezultat in urma optimizarii.
- Daca o cerere este re-executata pana sa apara alte cereri care sa afecteze planul de executie etapa Parse nu mai este necesara la re-executie (se foloseste rezultatul existent).

Data Dictionary Cache

- Contine cele mai recent folosite informatii din dictionarul de date:
  - Descrieri tabele si coloane
  - Date cont user
  - Drepturi (privilegii)
  - Etc.
- In etapa Parse acest cache e folosit de procesul server pentru informatiile necesare compilarii cererii (numele folosite in cererea analizata). Daca nu exista sunt incarcate din fisierele de pe disc.

Database Buffer Cache

- Parte a SGA. Tine cele mai recent utilizate blocuri de date.
- Cand o cerere este executata procesul server verifica aici existenta blocurilor necesare. Daca nu exista le citeste si le plaseaza aici.
- Dimensiunea sa este data de parametrul DB_BLOCK_BUFFERS
- Dimensiunea unui bloc e data de parametrul DB_BLOCK_SIZE.

Database Buffer Cache – cont.

- Oracle utilizeaza algoritmul LRU (ce este acesta?) pentru eliberare pozitii in buffer,
- Exceptie: operatiile de tip full table scan (cand o intreaga tabela este citita in memorie). In acest caz ultimele blocuri citite pot fi primele dealocate.
- Algoritmii utilizati sunt complecsi, cele de mai sus sunt o prezentare schematica a strategiilor de gestiune a bufferului.

(vezi de exemplu http://www.adp-gmbh.ch/ora/concepts/cache.html)

Program Global Area (PGA)

- Zona de memorie folosita in mod exclusiv de un proces (server sau background) – nu e comuna ca in cazul SGA. Ea contine:
  - Sort area – zona sortari, folosita la operatiile de ordonare (sortare).
  - Informatii sesiune, de exemplu drepturile userului acelei sesiuni.
  - Stare cursor continand starea cursorilor folositi in sesiunea respectiva
  - Stiva, continand diverse variabile de sesiune.

Exemplu: procesare UPDATE

- Sa luam exemplul unei cereri de actualizare:

```
UPDATE EMP
SET SAL = SAL * 1.1
WHERE EMPNO=1123
```

- Se executa intai Pasul 1 - Parse
- La etapa EXECUTE (Pasul 2) se efectueaza operatiile:

Exemplu: procesare UPDATE – cont

1. Procesul server citeste blocurile de date si de rollback – valori anterioare modificarilor necomise - din fisierele de date (daca nu sunt deja in Buffer Cache)
2. Copii ale blocurilor sunt puse in Buffer Cache (daca nu existau).
3. Procesul server blocheaza datele respective.
4. Procesul server inregistreaza in Redo Log Buffer schimbarile de facut in rollback si in date

Exemplu: procesare UPDATE – cont

5. Procesul server plaseaza versiunile originale ale blocurilor modificate in Rollback si modifica blocurile de date. Ambele operatii se fac in Buffer Cache. Ambele blocuri (rollback si date) sunt marcate ca ‘dirty blocks’ (blocuri modificate), adica blocuri care nu sunt identice cu cele de pe disc.

Segment de Rollback

- Inainte de a se face schimbari, serverul salveaza vechile valori de bloc intr-un segment de rollback. Aceasta salvare permite anularea tranzactiei (operatia ROLLBACK, opusa lui COMMIT), asigura ca alte tranzactii pot citi valorile anterioare inceputului de tranzactie (read consistency – mai stiti ce era asta?) si ne permit de asemenea recuperarea dupa incident.

Segment de Rollback - cont

- Segmentele de rollback sunt stocate in fisierele de date ale bazei de date si sunt aduse in buffer cache la cerere (cand este nevoie de ele).

Redo Log Buffer

- Procesul server inregistreaza schimbarile facute de o instanta in Redo Log Buffer.
- Are o dimensiune data de parametrul LOG_BUFFER (in bytes).
- Contine inregistrari Redo: blocul care a fost schimbat, pozitia schimbarii, noua valoare.

Redo Log Buffer - cont

- Contine toate schimbarile, atat in date cat si in blocuri de index, rollback, etc.
- Scrierea acestor inregistrari este secventiala
- Contine inregistrari privind schimbarile facute de toate tranzactiile.
- Este folosit prin parcurgere circulara. Cand un bloc e refolosit, el este scris anterior in fisierele de pe disc.

Database Writer

- Este un tip de proces de background
- Scrie ‘dirty blocks’ din buffer in fisierele de date, astfel incat sa existe suficiente blocuri libere care sa fie folosite de sistem
- A fost necesar pentru a degreva procesul server de aceasta operatie – imbunatatirea performantelor

Database Writer - cont

- Scrierea se face cand:
  - Numarul de ‘dirty blocks’ in buffer depaseste o anumita valoare
  - Un proces care cauta locuri libere in buffer nu le gaseste
  - Timeout
  - Evenimente care forteaza un checkpoint: exemplu: inchiderea bazei de date.

Log Writer

- Este un tip de proces de background
- Scrie intrari din Redo Log Buffer in fisierele de pe disc.
- Operatia se efectueaza cand:
  - Redo Log Buffer e aproape plin
  - Timeout
  - Inainte ca DBWR sa scrie blocurile modificate pe disc
  - Cand o tranzactie e comisa

COMMIT

- Oracle utilizeaza un mecanism rapid de commit care garanteaza recuperarea schimbarilor comise in caz de incident.
- Acest mecanism utilizeaza un System Change Number (SCN) fiecarei tranzactii care comite. Aceste numere sunt crescatoare si unice la nivelul bazei de date

COMMIT - Pasi

- La aparitia unui COMMIT:
  1. Procesul server plaseaza o inregistrare de commit, impreuna cu SCN-ul acesteia in Redo Log Buffer
  2. LGWR scrie o portiune contigua de inregistrari din buffer pana la cea care contine commit in fisierele redo de pe disc. In felul acesta este garantata recuperarea dupa incident

COMMIT - Pasi

3. Userul este informat ca s-a efectuat COMMIT-ul.
4. Procesul server inregistreaza ca tranzactia e completa si ca resursele blocate de ea pot fi eliberate.

Deci:

- Scrierea efectiva a datelor pe disc este efectuata independent de DBWR.
- Dimensiunea tranzactiei nu conteaza

Bibliografie generala

1. Colin McGregor - Oracle Database 2 Day DBA, 10g
2. Ulrike Schwinn, Vijayanandan Venkatachalam – Database administration

Lecturi obligatorii

1. Colin McGregor - Oracle Database 2 Day DBA, 10g

Link: http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14196.pdf

- Capitolul 5 (Managing the Oracle Instance) – integral
- Capitolul 6 (Managing Database Storage Structure) pag. 6-1 pana la 6-5 (fara Tablespaces)

Sfârşitul primului capitol