

# **Paralelné spracovanie Oracle**

Administrácia Oracle (NDBI013)

Rene Fischer

# Obsah

1. Všeobecne
2. Ako to funguje
3. Typy
4. Parametre

# Kedy to pomôže

- Operácie náročné na dáta
  - na veľkých databázach typicky spojené so systémami pre podporu rozhodovania (**D**ecision **S**upport **S**ystem) a dátovými skladmi
    - spočítať celkový predaj za rok - proces pracuje napr. s dátami z ¼ roka
- Dávkové spracovanie
  - v **O**n**L**ine **T**ransaction **P**rocessing
- Údržba schéme DB
  - napríklad vytvorenie indexu

# S ktorými operáciami

1. Dotazy
  - scanujúce veľké tabuľky, joiny, delený scan indexov
2. Vytváranie veľkých indexov
3. Vytváranie veľkých tabuliek
  - vrátane materializovaných pohľadov
4. Dávkové operácie insert, update, delete, mergeovanie
5. Na prístup k objektom v DB
  - napríklad prístup k veľkým objektom **LargeOBjects**

# Merge

- Používa sa pri výbere riadkov z viacerých zdrojov pre update alebo insert do tabuľky/pohľadu. Vyhneme sa tým viacerým operáciám insert,update,delete

```
MERGE INTO bonuses D
USING (SELECT employee_id, salary, department_id
       FROM employees WHERE department_id = 80) S
ON (D.employee_id = S.employee_id)
WHEN MATCHED THEN
    UPDATE SET D.bonus = D.bonus + S.salary*.01
    WHERE (S.salary > 8000)
WHEN NOT MATCHED THEN
    INSERT (D.employee_id, D.bonus)
    VALUES (S.employee_id, S.salary*.01)
    WHERE (S.salary <= 8000);
```

# Kedy a kde používať

Na systémoch, ktoré spĺňajú

1. Symetrické multiprocesory(SMP), clusters, masívne paralelné systémy
2. Dostatočná I/O priepustnosť
3. Nevyužité alebo prerušovane používané procesory
  - napr. kde je využitie procesora menej ako 30%
4. Dostatok pamäte
  - na podporu dodatočných procesov náročných na pamäť(triedenie, hashovanie...)

# Kedy a kde nepoužívať

1. V prípade nedostatku zdrojov(viz. pred. slajd), paralelné spracovanie zníži výkon
2. Prostredia, kde typické dotazy trvajú krátko
  - typicky OLAP
  - režia na paralelizmus prevyšuje zrýchlenie
3. Prostredia kde sú zdroje(CPU, Memory, IO) silne využívané

Orion – zdarma nástroj od Oracle na meranie výkonu systému

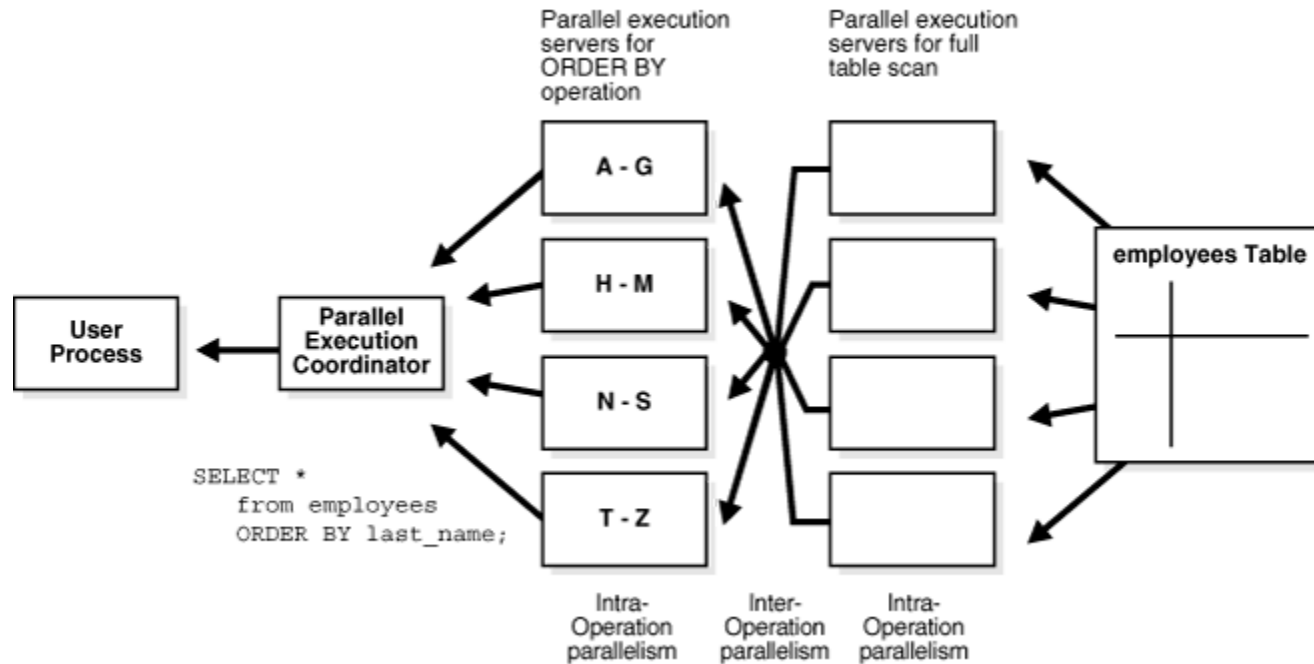
# Princíp

- Každý príkaz prechádza optimalizačným a paralelizačným(ak je p. prevádzanie zapnuté) procesom počas parsovania
  - po vytvorení plánu optimalizátorom, p. koordinátor zvolí p. execution metódu pre každú operáciu v pláne
    1. session užívateľa alebo shadow process prevezme rolu koordinátora
    2. koordinátor si zadováži potrebné množstvo p. serverov(paralel execution server)
    3. prevedenie príkazu – p. serveri pracujú samozrejme paralelne
      - full table scan, join, order by
    4. po skončení prací p. serverov prevedie koordinátor prácu, ktorá paralelne prevádzať nejde
      - vyhodnotenie SUM
    5. koordinátor vráti výsledok

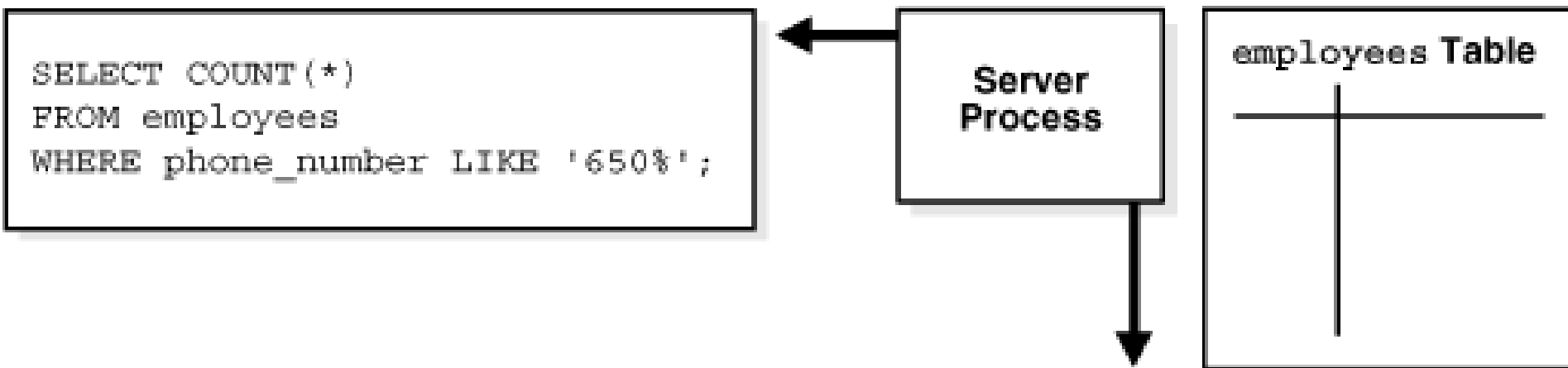


# Princíp

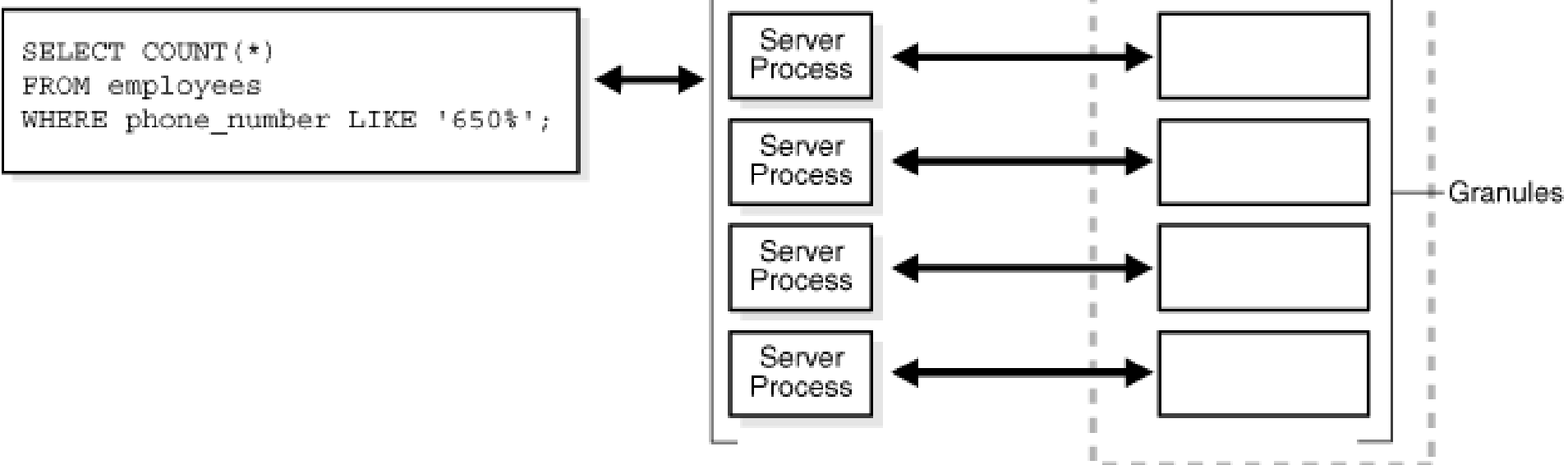
- `SELECT * FROM employees ORDER BY last_name;`
  - predpoklad, že nad `last_name` nie je vytvorený index



## Serial Process



## Parallel Execution Coordinator

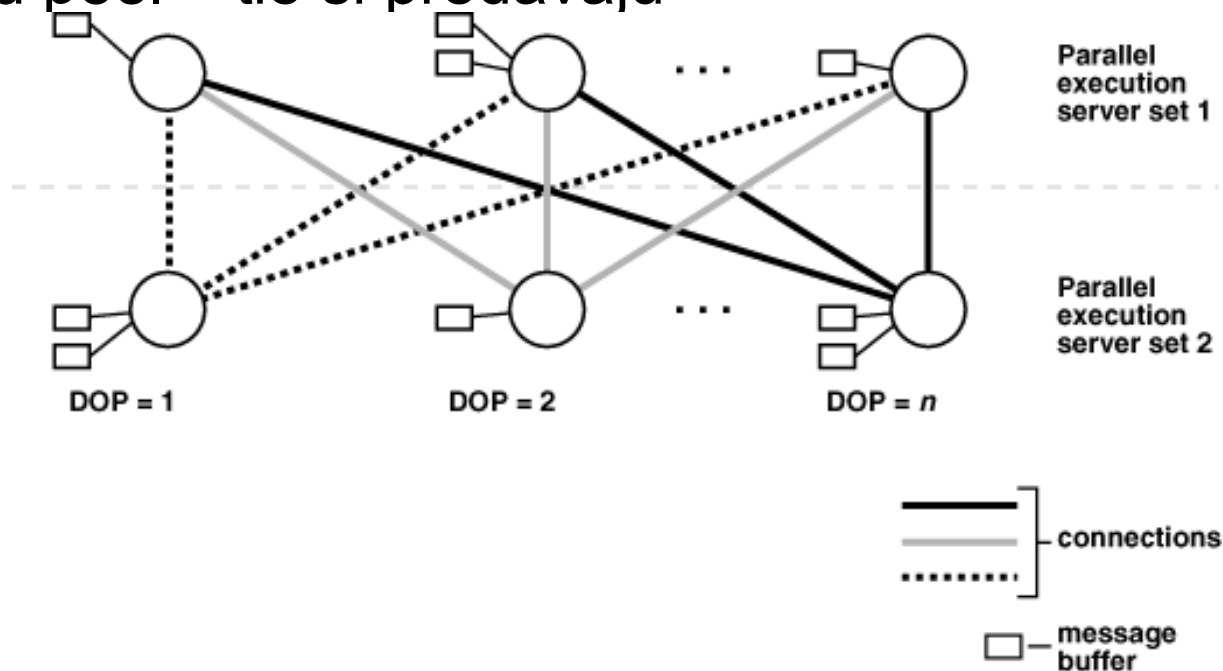


# Princíp

- Predávanie dát medzi servermi
  1. server spracuje nejaké riadky
  2. Pošle ich operátorovy
  3. Redistribuje ďalej
- Producent – produkuje dáta na výstup, na ktoré niekto čaká
- Konzumer – operácia potrebujúca na prácu výstup inej operácie
- Typy paralelizmu
  - intra operation – vrámci 1 operácie
  - inter operation – medzi operáciami v data-flow strome

# Komunikácia medzi servermi

- Medzi každým serverom v skupine producentov a skupine konzumentov existuje virtuálne spojenie
  - každý server má aj spojenie na koordinátora (nie je na obr.)
- Každé spojenie používa 1 až 4 buffre alokované zo shared pool – tie si predávajú



# Degree Of Parallelism

- počet serverov asociovaných s 1 operáciou
  - manuálne - ALTER TABLE customers PARALLEL 4;
    - v prípade operácie nad 2 tab. s iným DOP, zvolené vyššie
  - defaultne
    - jedno-inštančná DB - PARALLEL\_THREADS\_PER\_CPU x CPU\_COUNT
    - RAC - PARALLEL\_THREADS\_PER\_CPU x CPU\_COUNT x INSTANCE\_COUNT
  - automatic degree policy
    - či previesť hneď, alebo čakať na ďalšie zdroje(do fronty)
    - porovnanie odhadovaného času na prevedenie s PARALLEL\_MIN\_TIME\_THRESHOLD
    - pre istotu ešte z hora obmedzené PARALLEL\_DEGREE\_LIMIT

# Manuálne nastavovanie param.

- ALTER SESSION SET parallel\_degree\_policy = limited;
- ALTER TABLE emp PARALLEL (degree default);
- Hinty

```
SELECT /* hint */ ename, dname
```

```
FROM emp e, dept d
```

```
WHERE e.deptno=d.deptno;
```

- /\*+ parallel \*/ - vykonať paralelne
- /\*+ parallel(10) \*/ - DOG na 10
- /\*+ no\_parallel \*/ - vykonať sériovo
- /\*+ parallel(auto) \*/ - automatické
- /\*+ parallel(manual) \*/ - správanie sa ako Oracle 11.1

# In-memory & statement queuing

- in-memory
  - Oracle podľa parametrov (veľkosť, frekvencia prístupu) a heuristik rozhoduje, či objekt, s ktorým bude príkaz pracovať, bude načítaný celý do pamäte
- queuing – po naplánovaní a určení DOP je kontrola dostatku voľných zdrojov (neovplyvňuje sériové prevádzanie)
  - ak je dostatok zdrojov a vo fronte nie sú žiadne príkazy, príkaz sa prevedie (horná hranica `PARALLEL_SERVERS_TARGET`)
  - ak nedostatok, do fronty (based on specified conditions?) a čaká na splnenie podmienok

# Statement queuing

- Defaulte funguje fronta ako FIFO
- BEGIN\_SQL\_BLOCK ... END\_SQL\_BLOCK
  - skupina príkazov vložená do fronty s rovnakým časovým razítkom(prvého príkazu)
- Hinty
  - `/* NO_STATEMENT_QUEUING */` - ak je queuing zapnuté tak obíde
  - `/* STATEMENT_QUEUING */` - ak je queuing vypnuté, tak „zapne“ pre daný príkaz
- Resource Manager
  - nastavovanie plánu zdrojov určuje poradie vyberania z fronty
    - podľa zaťaženia
    - podľa „CUSTOMER GROUP“



# PARALLEL\_DEGREE\_POLICY

- ALTER SESSION | SYSTEM SET ...
- MANUAL
  - disablované
    - automatic degree of parallelism
    - statement queuing
    - in-memory parallel execution
- AUTO – opak MANUAL
- LIMITED – oproti MANULA
  - povoľuje automatic... pre príkazy pracujúce s tabuľkami alebo indexami sa nastavením príznakom PARALLEL

# Parallel Execution Server Pool

- pri štarte Oracle v poole vytvorí PARALLEL\_MIN\_SERVERS serverov
- spracovávané príkazy si berú z poolu servery, po ukončení vrátia
- v prípade potreby, vytvorí sa viac
  - maximálne PARALLEL\_MAX\_SERVERS
    - ak je vytvorené max a je potreba viac, Oracle prepína do seriového spracovania
  - po vrátení a nič nerobení daného intervalu je server ukončený
  - neklesá pod PARALLEL\_MIN\_SERVERS
  - Oracle vie pracovať aj s menej ako požadovaným množstvom serverov

# Typy paralelizmu

- Dotazy
- DDL príkazy
- DML príkazy
- Prevádzanie funkcií
- Ostatné
  - Recovery
  - Propagácia(replikácia)
  - Load
- Paralelný koordinátor v session môže koordinovať naraz iba 1 typ operácií

# Paralelné dotazy a pod-dotazy

- Dotazy a pod-dotazy môžu byť vykonávané paralelne aj ako časti DDL a DML príkazov
- Pravidlá pre určovanie DOP
  - dotaz – max z DOP pre všetky zúčastnené objekty
  - DML operácia – určuje menená tabuľka
    - prípadné pod-dotazy prevezmú DOD od DML
  - DLL – určuje tabuľka, index alebo partícia ktorá je vytváraná, rebuildovaná, delená alebo presúvaná
    - pod-dotazy majú DOD ako DLL

# Paralelné dotazy - pravidlá

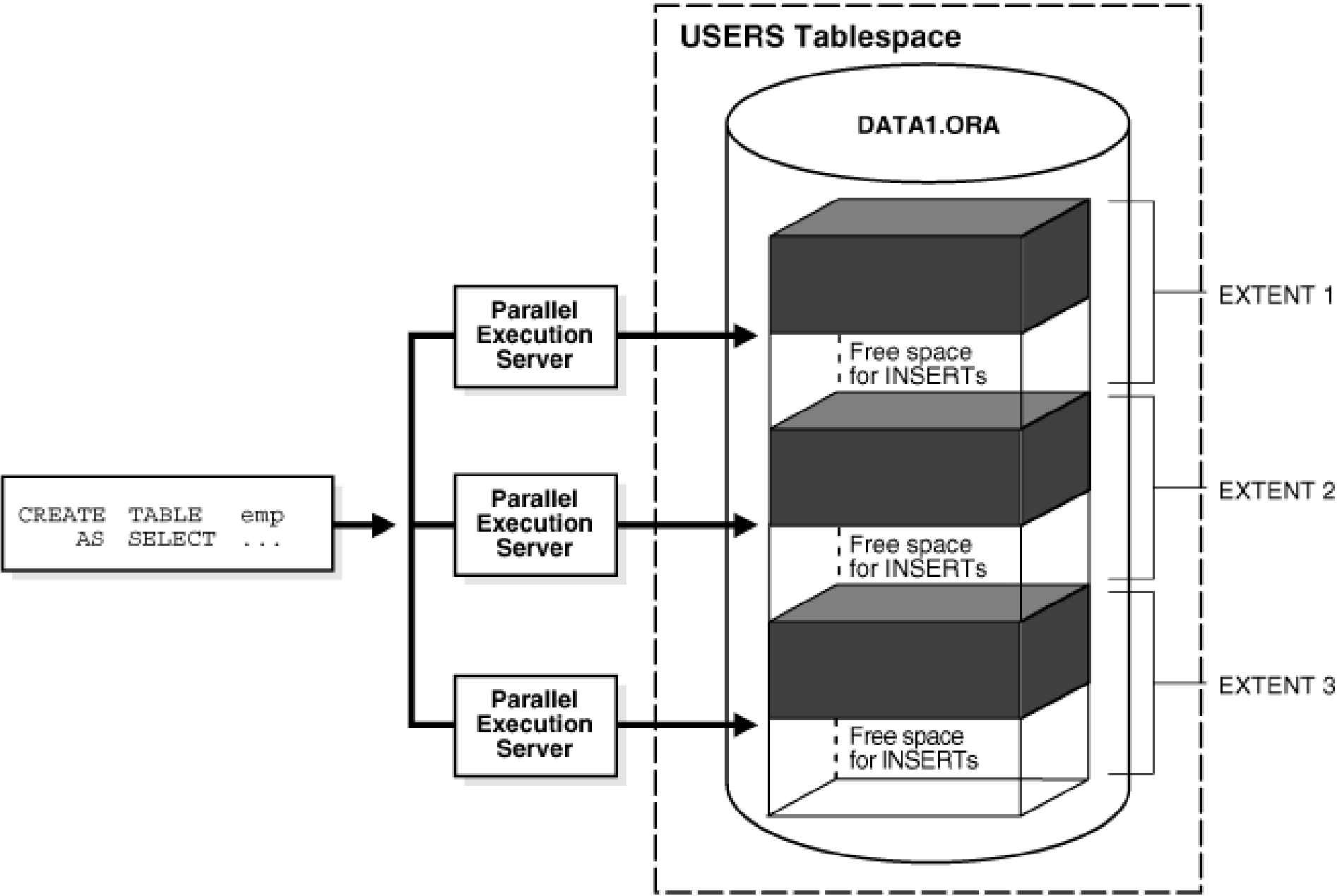
- Aby prebehol SELECT dotaz paralelne, musia byť splnené :
  1. jedno z
    1. HINT(PARALLEL or PARALLEL\_INDEX)
    2. referencovaná objekt je deklarovaný ako PARALEL
  2. aspoň jedna tabuľka potrebuje
    1. full table scan
    2. index range scan cez viac particíí
  3. s SELECT liste sa nenachádzajú skalárne poddotazy
- Hintové DOD má prednosť pre Objektovým

# Paralelné DLL

- neparticiované tabuľky
  - CREATE INDEX
  - CREATE TABLE ... AS SELECT
  - ALTER INDEX ... REBUILD
- particiované tabuľky
  - CREATE INDEX
  - CREATE TABLE ... AS SELECT
  - ALTER TABLE ... [MOVE|SPLIT|COALESCE] PARTITION
  - ALTER INDEX ... [REBUILD|SPLIT] PARTITION
- nemôže byť použité na tabuľkách s objektmi alebo LOB

# Paralelné DDL – priestor

- Pri vytváraní tabuľky každý server dostane dočasný segment veľkosti NEXT nastavenia
  - ak napr. NEXT = 4MB, POD = 16 => výsledná tabuľka môže mať až 64MB
  - ak ostane viac MINIMUM EXTENT prázdneho miesta v segmente, Oracle ich trimuje





# Paralelné DDL - pravidlá

- paralelizované, ak
  - použitá klauzula PARALLEL
  - ALTER SESSION FORCE PARALLEL DDL
  - Hint - prekrýva klauzulu

# Paralelné DML

- INSERT, UPDATE, DELETE, MERGE
- Nutné mať povolené v session
  - ALTER SESSION ENABLE PARALLEL DML;
    - a buď menená tabuľka definovaná ako paralel
    - alebo HINT
      - hneď sa úvodným kľúčovým slovom
      - UPDATE /\*+ PARALLEL(tbl\_2,4) \*/ tbl\_2 SET c1=c1+1;
  - rôzne spôsoby lockovania, transakcií, disk management
  - defaultne disablované
  - aj keď disablované, tak prípadne SELECT časti môžu stále paralelne

# Paralelné DML - obmedzenia

- Každý server vytvára vlastnú transakciu
  - 2 fázový commitovací protokol
- Výsledok paralelných zmien nie sú viditeľné počas transakcie
- Pokusy o prístup k tabuľke, ktorá bola menená paralelne v rámci rovnakej transakcie končia chybou.
- Nemôžu byť prevádzané na tabuľkách s triggermi

# Paralelné DML - obmedzenia

- Vylučujúce constrainty :
  - self-referential integrity
  - delete cascade
  - deferred integrity
- Obmedzenia na referenčnú integritu
  - ak by operácia na tabuľke A mohla vyvolať operácie na tabuľke B

# Paralelné prevádzanie funkcií

- volania sa môžu vyskytnúť v dotazoch ako súčasti
  - SELECT listu
  - SET klauzule
  - WHERE klauzule
- Prevádzané paralelne
  - statické(Java)/package variabie(PL-SQL) sú inicializované nanovo a sú úplne privátne na každé paralelné volanie
- Tabuľkové funkcie použité vo FROM klauzule inicializované raz pre každé paralelný proces
  - všetky premenné sú komplet súkromné pre proces

# Paralelné prevádzanie funkcií

- Vykonávané paralelne ak platí jedno z
  - PARALLEL\_ENABLE
  - ak je z package ktorý má
    - PRAGMA RESTRICT\_REFERENCES clause that indicates all of RNDS, WNDS, RNPS, and WNPS
  - ak je deklarovaná ako CREATE FUNCTION a systém môže zanalyzovať telo funkcie
- Ak DML má funkciu, ktorá nemôže byť volaná paralelne, celý DML príkaz sériovo. Neplatí pre SELECT časti v
  - INSERT ... SELECT
  - CREATE TABLE ... AS SELECT

# Session

- ALTER SESSION DISABLE PARALLEL DML|DDL|QUERY
  - ignorujú sa PARALEL klauzule pre tabuľky, indexi
  - Hinty sú silnejšie
- ALTER SESSION ENABLE PARALLEL DML|DDL|QUERY
- ALTER SESSION FORCE PARALLEL DML|DDL|QUERY [PARALLEL N]

# Parametre

- Oracle vypočítava defaultne hodnoty pre parametre ovplyvňujúce p. spracovanie
  - na základe CPU\_COUNT a PARALLEL\_THREADS\_PER\_CPU
- Možnosť zmeny parametrov pre
  - systém
  - samostatné dotazy
  - session
    - ALTER SESSION FORCE PARALLEL QUERY;
    - všetky nasledujúce dotazy sú prevádzane p., bez porušovania obmedzení
    - prepisuje všetky p. klauzule nasled. dotazov v session
    - hinty prepisujú



# Parametre 1

- **PARALLEL\_ADAPTIVE\_MULTI\_USER**
  - (true) spôsobuje potláčanie DOP aby zabránilo preťaženiu systému
- **PARALLEL\_AUTOMATIC\_TUNING**
  - (false) automatické ladenie parametrov ovplyvňujúcich paralelné spracovanie
- **PARALLEL\_DEGREE\_LIMIT**
  - kontroluje maximum DOP operácií ak je zapnuté automatické DOP
  - CPU\_COUNT x PARALLEL\_THREADS\_PER\_CPU x ACTIVE\_INSTANCES
- **PARALLEL\_DEGREE\_POLICY**
  - (manual, limited, auto)
- **PARALLEL\_EXECUTION\_MESSAGE\_SIZE**
  - (16KB) veľkosť bufferov používaných na zasielanie správ medzi servermi navzájom a koordinátorom dotazov
- **PARALLEL\_FORCE\_LOCAL**
  - (false) vymedzenie paralelného vykonávania iba na aktuálny ORACLE RealApplicationCluster

# Parametre 2

- **PARALLEL\_MAX\_SERVERS**
  - určuje max. množstvo p.prevádzaných procesov a p.zotavovacích(recovery) procesov zotavenia na inštanciu. Podľa rastu potreby, je procesu pridávaných procesov pod toto číslo.
  - $\text{CPU\_COUNT} \times \text{PARALLEL\_THREADS\_PER\_CPU} \times (2 \text{ if } \text{PGA\_AGGREGATE\_TARGET} > 0; \text{ otherwise } 1) \times 5$
- **PARALLEL\_MIN\_SERVERS**
  - (0) množstvo p. procesov štartovaných a rezervovaných pre p. operácie, pri štarte Oracle
- **PARALLEL\_MIN\_PERCENT**
  - min. percento požadovaných p. procesov pre p. spracovanie
- **PARALLEL\_MIN\_TIME\_THRESHOLD**
  - (10s.) čas spracovania operácie, nad ktorú je operácia označená pre p. spracovanie
- **PARALLEL\_SERVERS\_TARGET**
  - množstvo p. execution server procesov dostupných pre beh dotazu pred použitím p. statment queing
  - $4 \times \text{CPU\_COUNT} \times \text{PARALLEL\_THREADS\_PER\_CPU} \times \text{ACTIVE\_INSTANCE}$
- **PARALLEL\_THREADS\_PER\_CPU**
  - počet procesov(vlákién) spracovateľných na CPU počas p. spracovania

# Plán prevádzania

```
CREATE TABLE rf_customers  
( idCustomer INT NOT NULL,  
  firstName VARCHAR(255) NOT NULL,  
  lastName VARCHAR(255) NOT NULL  
);
```

```
BEGIN  
for i in 1 .. 100000 loop  
insert into rf_customers values  
  ( i,  
    'firstName_' || (100000 - i),  
    'lastName_' || i,  
  );  
end loop;  
end;
```

# Plán prevádzania

- set timing on;
- `SELECT [/*+ PARALLEL(4) */] COUNT(lastName) FROM rf_customers ORDER BY lastName DESC, firstName ASC;`
- 100 000 riadkov
  - sériovo - 00:00:00.01
  - paralelne - 00:00:00.09
- explain plan for  
`SELECT [/*+ PARALLEL(4) */] COUNT(lastName) FROM rf_customers ORDER BY lastName DESC, firstName ASC;`
- `select plan_table_output from table(dbms_xplan.display('PLAN_TABLE',null,'ALL'))`

# Sériovo

-----

Id	Operation
----	-----------

-----

0	SELECT STATEMENT
---	------------------

1	SORT AGGREGATE
---	----------------

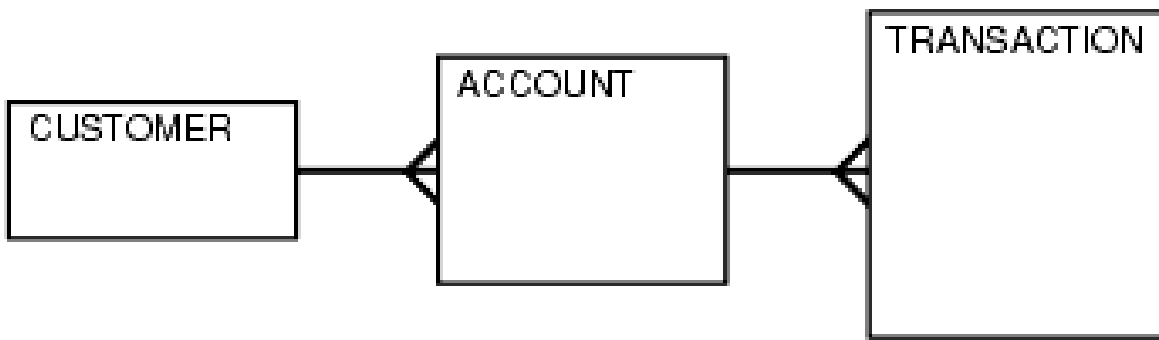
2	TABLE ACCESS FULL
---	-------------------

-----

# Paralelne

---

Id	Operation	... IN-OUT	PQ Distrib	
0	SELECT STATEMENT			
1	SORT AGGREGATE			
2	PX COORDINATOR			
3	PX SEND QC (RANDOM)	...  P->S	QC (RAND)	
4	SORT AGGREGATE	...  PCWP		
5	PX BLOCK ITERATOR	...  PCWC		



SELECT SUM(salary) FROM emp2 GROUP BY department\_id;

-----		...-----
Id	Operation	...  PQ Distrib
-----		...-----
0	SELECT STATEMENT	...
1	PX COORDINATOR	...
2	PX SEND QC (RANDOM)	...  QC (RAND)
3	HASH GROUP BY	...
4	PX RECEIVE	...
5	PX SEND HASH	...  HASH
6	HASH GROUP BY	...
7	PX BLOCK ITERATOR	...
8	TABLE ACCESS FULL	...

# Plán prevádzania

- Scanovanie je robené jednou skupinou serverov
- Druhá skupina prevádza GROUP BY
- PX BLOCK ITERATOR delí riadky medzi servery
- PX SEND, PX RECEIVE – spracované riadky prechádzajúce z 1 skupiny serverov do druhej
  - rozdeľované pomocou Hash tabuľky
- PX SEND QC – agregované hodnoty posielaný na QC v random poradí.
- PX COORDINATOR predstavuje QC or Query Coordinator koordinuje a plánuje paralelné spracovanie



# Zdroje

- Dokumentácia Oracle

- [http://download.oracle.com/docs/cd/E11882\\_01/server.112/e16541/parallel001.htm](http://download.oracle.com/docs/cd/E11882_01/server.112/e16541/parallel001.htm)

- 7 dokumentačných stránok v rade za sebou

- [http://download.oracle.com/docs/cd/E11882\\_01/server.112/e16638/ex\\_plan.htm#i26005](http://download.oracle.com/docs/cd/E11882_01/server.112/e16638/ex_plan.htm#i26005)

- plán prevádzania posledného príkladu