

Dátové sklady Techniky a optimalizácia

Miroslav Dávid



IT řešení, které přináší ovoce

1.10.2010

Pokročilé databázové technológie, FIIT STU

Obsah

Dátový sklad - OLAP

Load do DWH

Constrainty

Materializované pohľady

Optimizer a Hints

Partitioning

Paralelizmus

Príklad priebežnej optimalizácie



Dátový sklad - OLAP

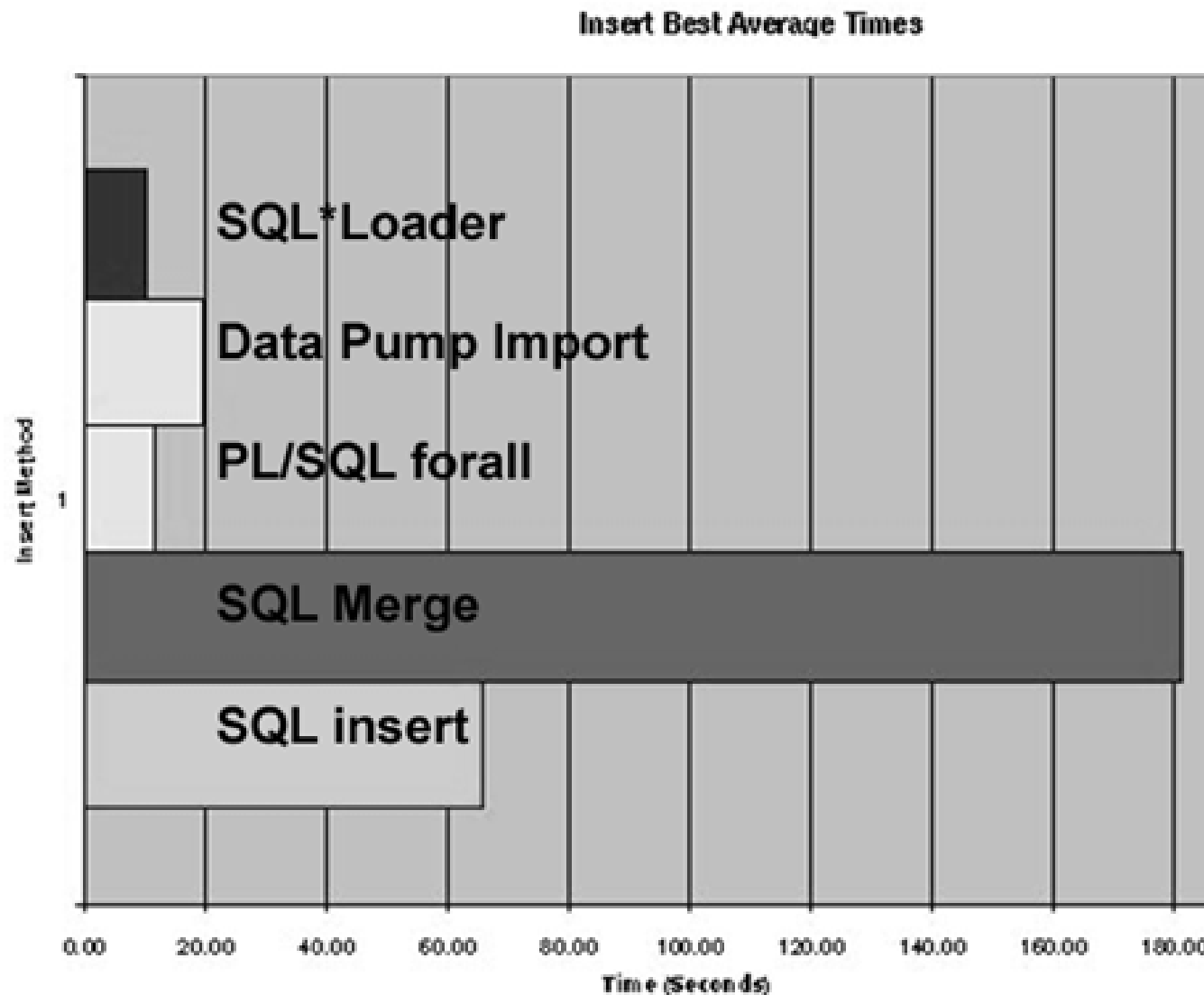
- Zhrnutie - Dátový sklad (OLAP)
 - Veľký objem dát (TB)
 - Read-only snapshot
 - Zjednodušený dátový model
 - Dôraz na rýchlosť vyhľadávania (→SELECT), menej na rýchlosť vkladania (→INSERT), malý dôraz na modifikáciu dát(→ UPDATE a DELETE)
 - Veľké množstvo historických údajov (nemenných)
 - Reprodukovateľnosť loadu dát
 - Dôležité pre business, avšak nie priamo kritické pre beh firmy

Load do DWH

- Load do DWH – proces náročný na čas
 - Spracuj minimálne množstvo dát, ktoré je nutné pre aktualizáciu dát
 - CDC – Change data capture
 - Inkrementálny load
 - ...
 - Dáta vlož do DWH najrýchlešie, ako je to možné
 - Bulk load - vytváranie priamo db blokov, obchádzanie DB logu, využitie len pre insert



Load do DWH - Oracle



Rýchlosť: DML vs. DDL

- Update

- Update – pomalá operácia
- Delete + Insert
- Truncate vs. Delete
 - Truncate = DDL (no rollback)
 - Truncate oveľa rýchlejší ako Delete
 - Delete môže mať WHERE podmienku
- Merge (kombinácia INSERT a UPDATE – UPSERT)

- DDL

- Drop + Create as select



Constrainty

- Constrainty: Primary K, FK, Unique, Check, ...
- Rýchlosť ?
 - → V dátovom sklade často vypnuté indexy
 - → Za referenčnú integritu je zodpovedné ETL



- Validate/Novalidate
 - Conform existing data
- Enabled/Disabled
 - Conform new data
- Rebuild
- Drop/Recreate

Materializované views

- SQL dotaz: komplexný objekt pozostávajúci z atomárnych častí, vytváraný v runtime
- Materializovaný view:
 - view fyzicky uložený v DB
 - obsahuje ako definíciu, tak aj dáta
 - redukuje opakujúce sa I/O operácie
 - Dôvody: pre-join, pre-sort, pre-aggregation, ...
- Refresh mview
 - Manuálne (complete, fast)
 - automaticky
- Automatické SQL Query rewrite
 - Query optimizer identifikuje vhodnosť využitia mview v SQL query

Optimizer a Execution Plan

- Pred vykonaním SQL príkazu Optimizer vypočíta (čo najefektívnejší) spôsob, akým bude pristupovať k dátam – **Query Execution Plan** (dôležitý pri performance analýze)
- Definuje:
 - Použitie indexov (prípadne ktorých)
 - Prístup k dátam (full scan, index scan, ...)
 - Poradie tabuliek pri joine
 - Typ joinu (nested loop, hash, merge,...)
- Optimizer používa na nájdenie efektívneho Query execution planu **štatistiky**.



Hints

- SQL direktívy špecifické pre konkrétny DBMS
- Override defaultného Query Execution planu
 - Hints for Access Paths
 - Join hints
 - Hints for parallel execution
 - Ďalšie ...

```
SELECT /*+ ALL_ROWS */  
FROM servers;
```

I want it that way



Logovanie

- REDO log – logovanie všetkých operácií na DB
- Vypnutie logovania – umožňuje performance nárast
 - Oracle: nologging (vo využití s /*+ APPEND */ hintom)
 - MSSQL: minimally logged operations



```
INSERT /*+ APPEND */  
INTO t  
SELECT * FROM servers;
```

Partitioning

- Rozdelenie logickej databázy alebo jej častí (tabuliek) na viac fyzicky oddelených častí
- **Výkon**, zjednodušenie správy, dostupnosť
- Typické použitie:
 - Rozdelenie tabuliek s veľkým počtom riadkov (rádovo desiatky miliónov a viac)
 - Napr. rozdelenie faktových/agregačných tabuliek podľa dátumu



Partitioning 2

→ Horizontálny
→ Vertikálny

→ Statický
→ Dynamický

- Partition pruning
 - Automatická optimalizácia SQL dotazu podľa partícií
- Metódy:
 - Range (interval)
 - Hash (podľa hash funkcie)
 - List (vopred zadefinovaný zoznam hodnôt pre part.)
 - Composite (subpartitioning)



Partitioning 3

- Partitioning a indexy ?
- Operácie: create, drop, coalesce, split, merge, truncate, ...

```
create table sales (  
    year number(4),  
    product varchar2(10),  
    amt number(10,2)  
)  
partition by range (year)  
partition p1 values less than (2008) tablespace u1,  
partition p2 values less than (2009) tablespace u2,  
partition p3 values less than (2010) tablespace u3,  
partition pm values less than (MAXVALUE) tablespace u4;
```

```
alter table sales add partition p4 values less than (2011);
```

- Druhy paralelizmu:
 - Parallel query (SELECT) – full scan
 - Parallel DML, Parallel DDL – partitionované tabuľky
 - Parallel Data loading

```
SELECT /*+ PARALLEL(emp,4) */  
COUNT(*)  
FROM emp;
```

```
CREATE INDEX emp_ix ON emp (emp_id)  
TABLESPACE ind  
STORAGE (INITIAL 1M NEXT 1M  
PCTINCREASE 0 MAXEXTENTS 20)  
PARALLEL (DEGREE 4);
```



DB tuning

1. DB dizajn (3NF, denormalizácia, partitioning, ...)
2. Application tuning (optimálne SQL, hints)
3. Memory tuning (nastavenie buffers)
4. Disk I/O tuning (chýbajúce indexy, sorty, full scany,...)
5. Eliminácia kolízií na DB (waits, DB locks, ...)
6. OS tuning (špecifické pre konkrétne OS, napr. počet otvorených súborov, ...)

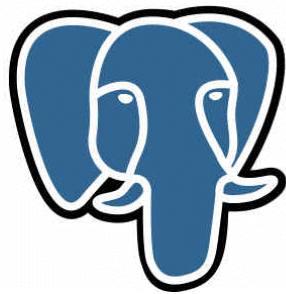
Nástroje

ORACLE®



Microsoft®
SQL Server® 2008

PostgreSQL



SYBASE®
*i*Anywhere®



Priebežná optimalizácia: Zrýchlenie ETL procesov



Požiadavky



- Problém: Neskoré dodávanie reportov na dennej báze
 - Rozvoj DWH
 - Riešenia vyvíjané so slabým dôrazom na čas spracovania
 - Žiadne globálne optimalizácie (na úrovni workflowov)
- Jasné požiadavky –
 - Zrýchliť denné spracovanie**
 - Zlepšiť dostupnosť DWH**



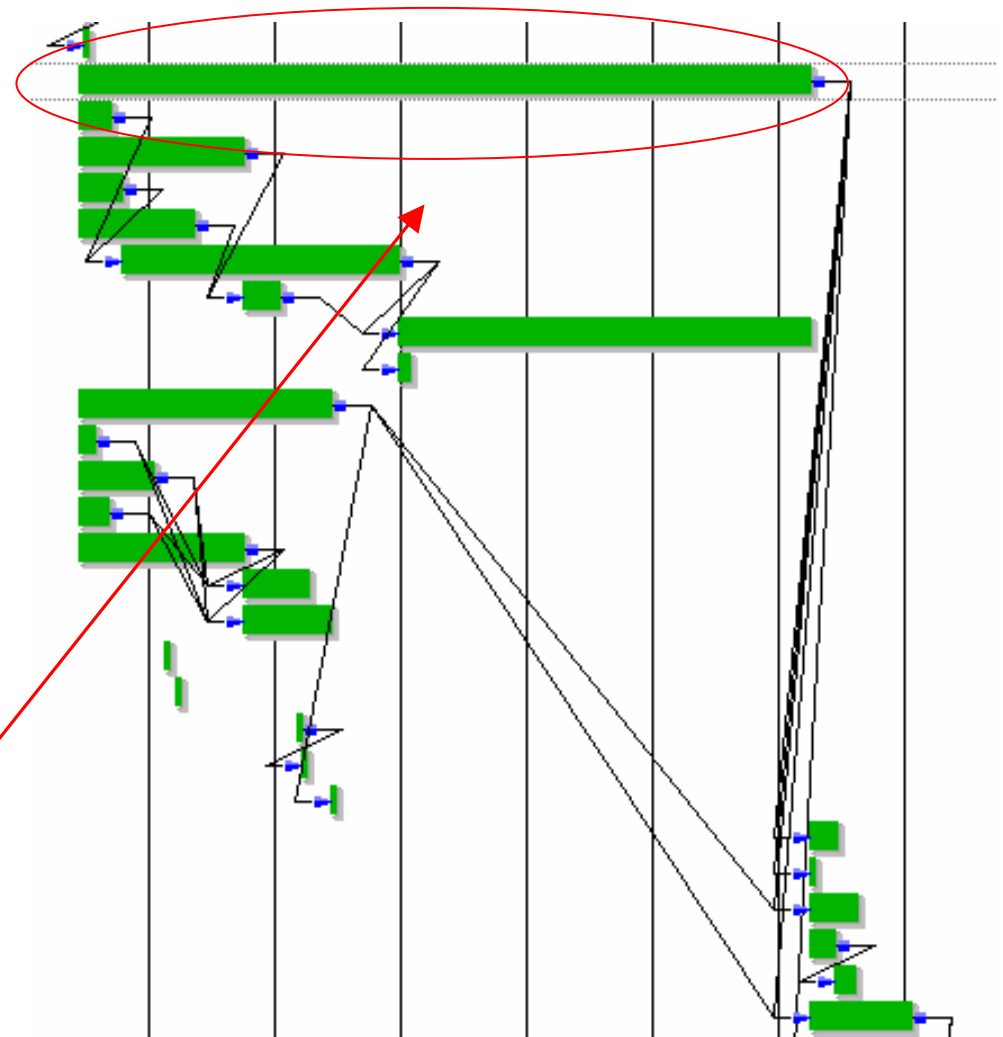
Riešené problémy

- Identifikácia menených ETL procesov
- Analýza ETL a hľadanie možností optimalizácie
- Dôraz najmä na technickej časti riešenia



Identifikácia menených ETL

- Vstup: Záznamy o ETL spracovániach na produkčnom prostredí
- Na úrovni session:
 - **Metrika** - Čas spracovania / množstvo prenesených dát
- Na úrovni workflowu:
 - Procesy, na ktoré čakajú iné procesy



Analýza ETL a hľadanie možností optimalizácie

1. Zmena logiky ETL procesov

- Obmedzenie množstva prenesených dát
- Spojenie viacerých ETL do jedného procesu

2. Možnosti prostredia

- Rozdelenie záťaže medzi Informatica server a DB server

3. Zmena parametrov

- Úprava konfigurácie jednotlivých transformácií

Výstup:

- Návrh riešenia pre identifikované ETL
- Dokumentácia, ktorá pomáha pri vývoji nových riešení priamo zapracovať vytvorené **optimalizačné postupy**

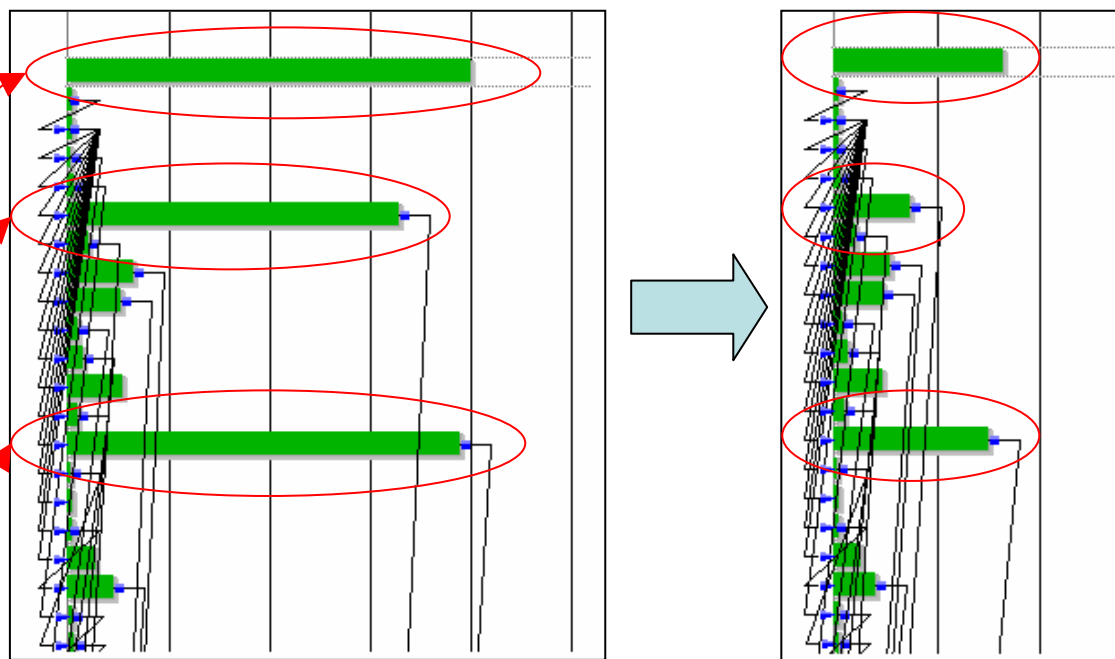
Unit testy

- Vývojové prostredie (DEV) vyťažené, nie je možné testovať čas behu
- Riešenie
 - DEV – korektnosť
 - TEST3 – čas behu, je potrebné preniesť dáta z DEVu

Integračné testy

- Štandardne na TEST1, TEST2 prostrediach

Výsledky



- Zrýchlenie spracovania nasadených riešení:
20 – 60 % pôvodného času
- Optimalizácie stále prebiehajú

	Pôvodný čas	Nový čas	Zrýchlenie	Počet prenesených riadkov
ETL proces	03:53:20	01:30:22	39 %	~ 33 400 000
ETL proces	03:17:07	00:43:55	22 %	~ 14 800 000
Workflow	04:00:25	01:38:08	41 %	

Diskusia

