Zapytania z ograniczeniem czasowym w Oracle

Tomasz Krasuski

22 stycznia 2009

Tytuł oryginalny

Supporting Time-Constrained Queries in Oracle

- Ying Hu, Seema Sundara, Jagannathan Srinivasan
- Oracle New England Development Center
- VLDB 2007
- Materiały źródłowe: referat, slajdy

Postawienie problemu

- Coraz większe bazy danych
  - Gigabajty, terabajty, petabajty
- Skomplikowane zapytania SQL
  - GROUP BY
  - ORDER BY
  - JOIN
- Efekt:
  - Zapytania trwające sekundy, minuty...
  - Trudny do przewidzenia czas trwania zapytań

Czas to pieniądz

- Czasem ważniejsza od odpowiedzi dokładnej jest odpowiedź **szybka**
- Oczekiwanie "nie wiadomo ile" na wynik zapytania jest niedopuszczalne

Przykłady

- Systemy wspomagania decyzji
  - Chcemy uzyskać częściowe wyniki, żeby na ich podstawie zauważyć trendy (np wzrost sprzedaży w różnych regionach)
- Aplikacje mobilne
  - Użytkownik jadący samochodem pyta o najbliższe kawiarnie w mieście przez które przejedzie za 2 minuty

Istniejące rozwiązania

- Zwracanie pierwszych n wyników
  - SELECT name FROM employees WHERE ROWNUM <= 100;
- **Top-k queries**: zwracanie k wyników o najwyższej randze
  - w Oracle klauzula ROWNUM + ORDER BY
- Przybliżone wyniki: na podstawie losowej próbki rekordów
  - SELECT sold_item FROM transactions WHERE year = 2005 SAMPLE BLOCK(10);
- Zastosowanie narzędzi optymalizacyjnych
  - w Oracle np. polecenie EXPLAIN PLAN

Wady istniejących rozwiązań

- Za zastosowanie tych mechanizmów odpowiada użytkownik
  - Jak "przetłumaczyć" zadane ramy czasowe na odpowiednie zapytanie?
- Nadal nie ma gwarancji, że zapytanie zawsze zakończy się w określonym czasie

Pomysł specjalistów z Oracle

- Dodatkowa klauzula w zapytaniu SELECT określająca ograniczenie czasowe:
  - Typ ograniczenia: "miękki", "twardy"
  - Limit czasu: w sekundach
  - Rodzaj wyniku: częściowy, przybliżony
- O zmieszczenie się w ramach czasowych niech już zadba sama baza danych

Działanie zapytań z ograniczeniem czasowym

- Rodzaj zwracanych odpowiedzi:
  - **wynik częściowy**: pierwsze n wyników lub wyniki o najwyższej randze (gwarancja dokładności)
  - **wynik przybliżony**: na podstawie próbki danych
- Rodzaj ograniczenia czasowego
  - **miękkie**: oczekiwany czas zakończenia zapytania
  - **twarde**: zapytanie musi się zakończyć w danym czasie
- Jeśli zapytanie skończy się szybciej, wynik może być dokładny

Składnia

SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
ORDER BY ...
[[**SOFT** | HARD] TIME CONSTRAINT (T)
[WITH **APPROXIMATE** | PARTIAL RESULT
];

Schemat działania

- Zapytanie częściowe przekształcane za pomocą klauzuli ROWNUM
- Zapytanie przybliżone przekształcane za pomocą klauzuli SAMPLE
- Przykład:
  - Przed: SELECT AVG(salary) FROM employees SOFT TIME CONSTRAINT (50) WITH APPROXIMATE RESULT;
  - Po: SELECT AVG(salary) FROM employees SAMPLE BLOCK (10);

Zapytanie z miękkim ograniczeniem

Definicja:
Dla zapytania Q z miękkim ograniczeniem czasowym t sek.,
czas $T_{estimated}(Q')$ oszacowany przez optymalizator BD ma mieć wartość z zakresu $[t - d, t]$
gdzie Q' - zapytanie w postaci przekształconej, d - mały przedział

Zapytanie z twardym ograniczeniem

Definicja:
Dla zapytania Q z twardym ograniczeniem czasowym t sek.,
czas $T_{elapsed}(Q')$ trwania zapytania musi być $<= t$
gdzie Q' - zapytanie w postaci przekształconej

Właściwa trudność

- Dla zapytań częściowych - **oszacowanie** ilości zwracanych wyników
- Dla zapytań przybliżonych - **oszacowanie** rozmiaru próbki, oraz **zdecydowanie** na których tabelach ma być dokonane próbkowanie (w przypadku zapytań na wielu tabelach naraz)
- Zapewnienie, że oszacowany czas mieści się w zadanym ograniczeniu

Szacowanie rozmiaru próbki

- Zał. mamy funkcję $f_Q(s)$ - czas potrzebny na wykonanie zapytania Q w zależności od rozmiaru próbki danych s
- (Analogicznie $f_Q(r)$ - czas wykonania zapytania Q w zależności od ilości zwracanych wyników r)
- Wtedy dla danego ograniczenia czasowego t, $f_Q(s) = t$
- Szukane s jest miejscem zerowym $f_Q(s) - t = 0$

Szacowanie rozmiaru próbki

1. Sprawdź szacowany czas trwania T_Q dla oryginalnego zapytania Q
2. Jeśli $T_Q < t$ - nie trzeba nic robić
3. Jeśli $T_Q > t$ - przekształć Q za pomocą ROWNUM lub SAMPLE na zapytanie Q'
4. Oszacuj czas trwania $T_{Q'}$ dla zapytania Q', startując od minimalnego rozmiaru próbki
5. Jeśli $T_{Q'} > t$ - "ERROR: NEED MORE TIME"
6. Iteruj algorytmem znajdowania MZ, aż $T_{Q'} \in [t - d, t]$

Zatem skąd wziąć funkcję $f_Q()$?

- W Oracle są już narzędzia do optymalizacji zapytań
- Wcześniej używano ich do szacowania czasu działania konkretnych zapytań
- Można ich użyć do szacowania czasu działania zapytań z ograniczeniem czasowym

Polecenie EXPLAIN PLAN

- Pozwala użytkownikowi poznać plan wykonania zadanego zapytania SQL
- Opisuje kolejność złączania tabel, sposób dostępu do tabel (pełne skanowanie / użycie indeksu)
- Wybiera najlepszy plan wykonania
- Szacuje czas wykonania

Polecenie EXPLAIN PLAN

- SELECT x from lineitem, part WHERE l_partkey = p_partkey AND l_shipdate >= date '1995-09-01' AND l_shipdate < date '1995-10-01' ORDER BY x;

```
| Id  | Operation           | Name     | Rows | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |          |    1|    51 |       | 88300   (9)| 00:04:39 |
|   1 |  SORT AGGREGATE     |          |    1|    51 |       |            |          |
|*  2 |   HASH JOIN         |          |  744K|    36M|    24M| 88300   (9)| 00:04:39 |
|*  3 |    TABLE ACCESS FULL | LINEITEM |  739K|    16M|       | 83047   (9)| 00:04:22 |
|   4 |    TABLE ACCESS FULL |     PART | 2010K|    53M|       |  2840   (8)| 00:00:09 |

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("L_PARTKEY"="P_PARTKEY")
   3 - filter("L_SHIPDATE">=TO_DATE('1995-09-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss') AND
              "L_SHIPDATE"<TO_DATE('1995-10-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss'))
```

- Szacowany czas wykonania możemy użyć w funkcji $f_Q()$

Schemat działania - ograniczenie twarde

1. Przekształć zapytanie tak jak dla miękkiego ograniczenia czasowego
2. Wygeneruj plan wykonania zapytania (EXPLAIN PLAN) i użyj oszacowań czasowych do:
   - utworzenia *licznika* zatrzymującego zapytanie po pewnym czasie
   - utworzenia *liczników* dla każdej operacji blokującej w zapytaniu

Operacje blokujące

- Operacje blokujące zaczynają zwracać wyniki dopiero po skonsumowaniu jednego ze źródeł danych wejściowych
- SELECT x from lineitem, part WHERE l_partkey = p_partkey AND l_shipdate >= date '1995-09-01' AND l_shipdate < date '1995-10-01' ORDER BY x;
- Sortowanie ORDER BY jest blokujące
  - Źródło danych wejściowych - złączenie tabel *lineitem*, *part*

Utworzenie liczników na podstawie EXPLAIN PLAN

Id	Operation	Name	...	Estimated Time
0	SELECT STATEMENT			300
1	SORT AGGREGATE			300
2	HASH JOIN			300
3	TABLE ACCESS FULL	LINEITEM		270
4	TABLE ACCESS FULL	PART		10

- **id = 0** timer dla zapytania 300 s
- **id = 1** timer dla operacji blokującej 300 s
- **id = 3** timer dla operacji blokującej 270 s

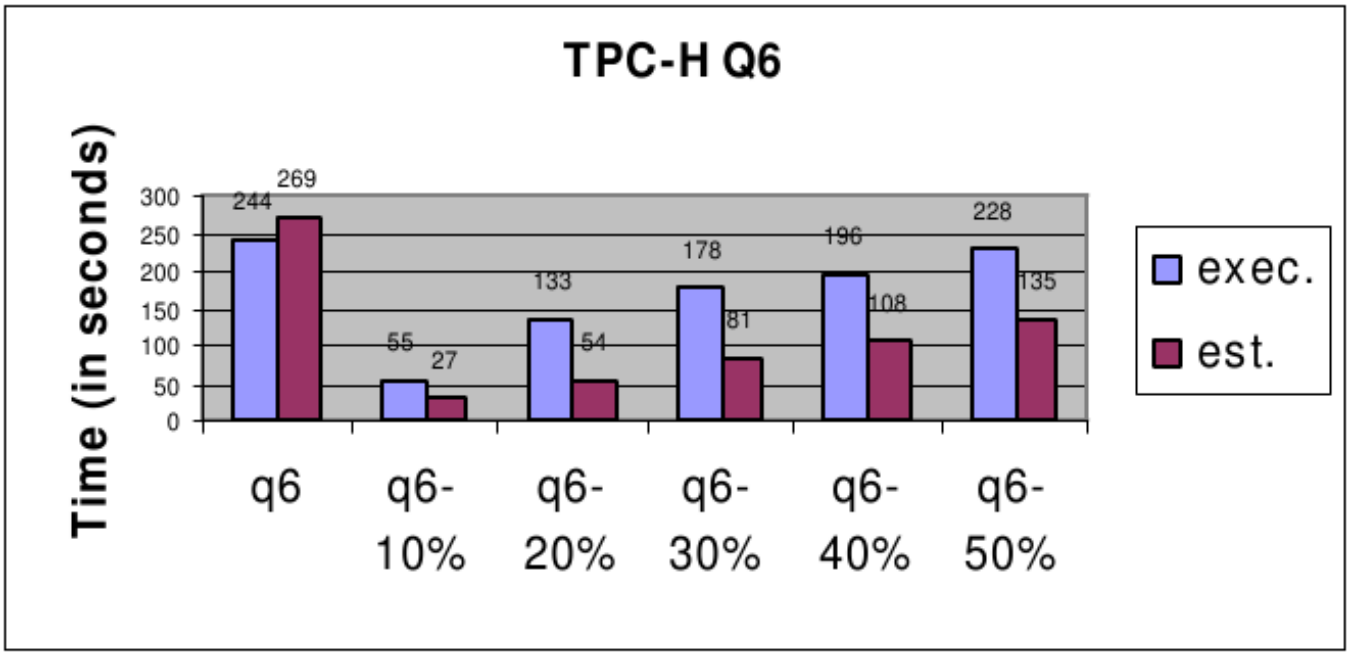
Środowisko do eksperymentów

- Platforma:
  - Intel P4, 3GHz, Hyper Threading, 2 GB RAM
  - RedHat Enterprise Linux 3, Oracle 10g Release 2 Enterprise Edition
- Parametry bazy danych:
  - Rozmiar bloku = 8192, rozmiar cache = 160MB
- Zestaw danych:
  - Baza TPC-H - standard do benchmarków
  - rozmiar bazy 10GB, 6 tabel
  - 2 największe tabele mają 60 mln i 15 mln wierszy

Zapytanie na pojedynczej tabeli z agregacją

- Zapytanie na tabeli o 60 milionach wierszy:
  SELECT SUM(l_extendedprice * l_discount) AS revenue
  FROM lineitem
  WHERE l_shipdate >= date '1994-01-01'
  AND l_shipdate < date '1994-01-01' + interval '1' year
  AND l_discount between 0.06 - 0.01 and 0.06 + 0.01
  AND l_quantity < 24
  SOFT TIME CONSTRAINT (x) WITH APPROXIMATE RESULT;

Zapytanie na pojedynczej tabeli z agregacją

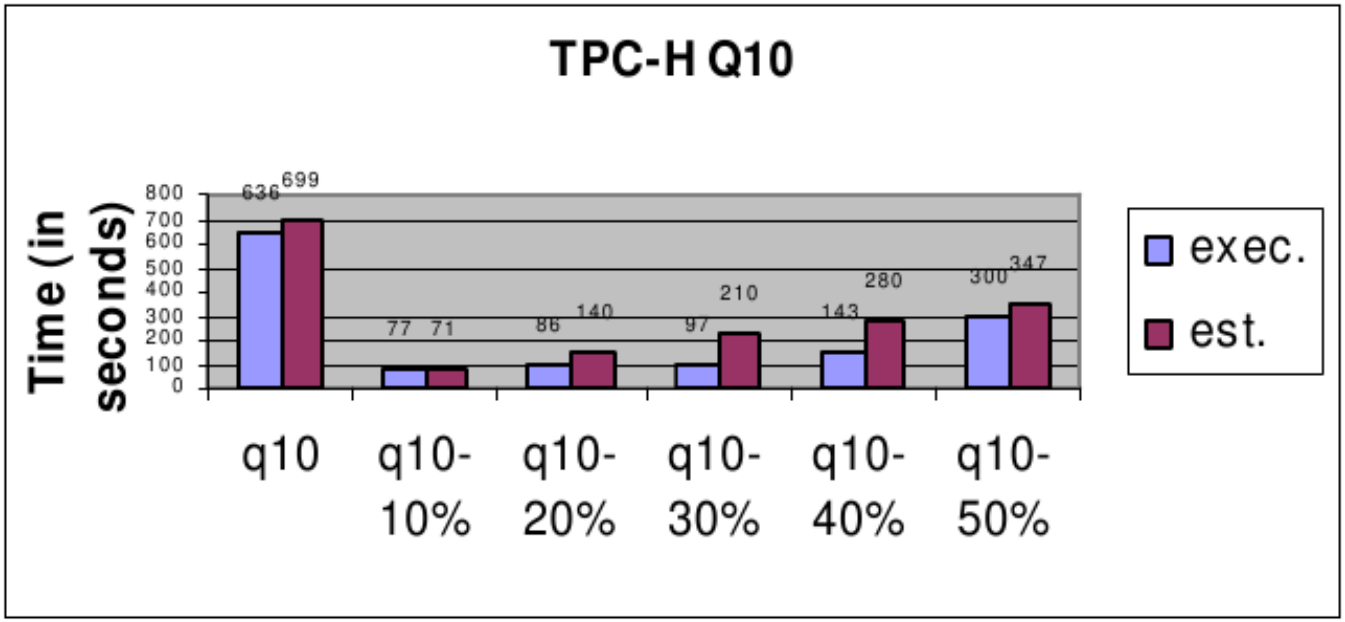


- Miękkie ograniczenie czasowe = 10%, 20%, ... oryginalnego czasu zapytania
- Optymalizator źle oszacował czas wykonania zapytań
- Mimo wszystko czas trwania się skrócił

Złączenie 4 tabel z GROUP BY i ORDER BY

```
SELECT c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) AS revenue,
c_acctbal, n_name, c_address, c_phone, c_comment
FROM customer, orders, lineitem, nation
WHERE c_custkey = o_custkey AND
l_orderkey = o_orderkey AND
o_orderdate >= date '1993-10-1' AND
o_orderdate < date '1993-10-1' + interval '3' month AND
l_returnflag = 'R' AND
c_nationkey = n_nationkey
GROUP BY c_custkey, c_name, c_acctbal, c_phone, n_name, c_address,
c_comment
ORDER BY revenue DESC
SOFT TIME CONSTRAINT (x) WITH APPROXIMATE RESULT;
```

Złączenie 4 tabel z GROUP BY i ORDER BY



- Miękkie ograniczenie czasowe = 10%, 20%, ... oryginalnego czasu zapytania
- Próbkowanie SAMPLE włączone tylko dla największej tabeli (60 mln wierszy)
- Ograniczenie czasowe działa skutecznie