ИНФОРМАТИКА

УДК 519.682; 681.324.06

Д. В. Антонов, В. С. Рублёв

Анализ технологий вычисления ODQL-запросов СУБД DIM

В данной работе проводится сравнительный анализ нескольких способов реализации запросов на языке ODQL [6] для СУБД DIM [3].

Ключевые слова: объектная СУБД, объектный язык запросов, метауровень, технологии запросов для языка ODQL

D. V. Antonov, V. S. Rubliov

Analyze of DIM DBMS ODQL-queries calculation technologies

In this work the comparative analyze of several ways of queries realization in the ODQL [6] language for DIM [3] DBMS is carried out.

Keywords: object DBMS, an object query language, metalevel, query technologies for the ODQL language

1. Постановка задачи

Разработка новой объектной технологии СУБД DIM [3] направлена на эффективность работы с базами данных (БД) [1−2]. В частности, объектный язык запросов ODQL [6] является удобным средством манипуляции данными, которое позволяет проще описывать сложные запросы и обладает запросной полнотой [4]. Однако требуется, чтобы и по времени выполнения запросов этот язык был эффективен [8]. В [5] разработан алгоритм выполнения объектных запросов ODQL, который эффективен по временной трудоемкости.

Этот алгоритм основан на анализе сложных связей данных в этой СУБД, и для обеспечения эффективности он использует новый аппарат индекс-выборок объектов классов и индекс-выборок отношений классов. Но является ли он лучшим по времени выполнения запроса по сравнению с подходом использования реляционных таблиц метауровня, положенным в основу организации данных [7]? Последний подход основан на трансляции объектного запроса ODQL в реляционный запрос SQL к метауровню БД и выполнению этого SQL-запроса.

Для ответа на этот вопрос необходимо проведение исследований по выбору запросной технологии. С целью объективности необходимо также сравнение времени запроса обоих технологий с лучшим времени, которое может быть получено для SQL-запроса реляционной БД. Реляционная технология должна быть лучшей в этом смысле, но реляционный запрос, соответствующий по результату полностью объектному ODQL-запросу может быть во много раз сложнее. Поэтому, если время SQL-запроса не более чем в 1,5–2 раза лучше по времени ODQL-запроса, то это вполне приемлемо для выбора объектной технологии.

2. Выбор тестового примера

К выбору тестового примера необходимо предъявить требование достаточной трудоемкости, чтобы время указанных запросных технологий различалось. Поэтому подходящим для этого является запрос с трудоемкостью $\theta(n^3)$: время выполнения такого запроса будет расти достаточно быстро, что позволит различить эффективность технологий.

Поэтому возьмем следующее задание из [6] в качестве тестового примера схемы БД для этих запросов:

Корабли из множества *Ships*, число которых n_s, перевозят грузы из множества *Cargoes*, число наименований которых n_c, в порты из множества *Ports*, число которых n_p. При этом корабль *s* везет груз *c* в порт *p* в количестве *k*(*s*,*p*). Необходимо написать следующий запрос:

Список наименований грузов, которые везут наиболее нагруженные корабли и при этом в наименьшее число портов среди таких кораблей.

Сначала определим схему реляционной БД и запрос к ней. В этом случае выделим таблицы:

Ships с полями *IdShip, ShipName*, главным индексом IdShip,

Cargoes с полями *IdCargo, CargoName*, главным индексом IdCargo и

Ports с полями *IdPort, PortName*, главным индексом IdPort, а также таблицу связи

Ships_Cargoes с полями *IdShip, IdCargo, IdPort, Quantity*, главным индексом (IdShip, IdPort, IdCargo) и тремя внешними индексами IdShip, IdCargoes, IdPort.

SQL-запрос, оптимальный по трудоемкости, может выглядеть следующим образом:

```
Select Distinct Cargoes.Name from
Cargoes,(select ShipsCargoes.IdShip,ShipsCargoes.IdCargo from ShipsCargoes,
  (select IdShip from
    (     select min(ShM1.Cntp) as minCntp from (Select Count (Ship_Port.IdPort) as CntP,
Ship_Port.IdShip
                From (Select sc.IdShip,sc.IdPort  From ShipsCargoes sc,
                      ( Select Sum (Quantity) as SummC, IdShip
                         From ShipsCargoes
                         Group by IdShip
                       ) Ship_Quantity,
                      ( Select Max (SummC) as maxSum
                         From ( Select Sum (Quantity) as SummC, IdShip
                         From ShipsCargoes
                         Group by IdShip
                         ) Sh1
                       )
                     Where Ship_Quantity.SummC = maxSum and sc.IdShip = Ship_Quantity.IdShip
                      Group by sc.IdShip, sc.IdPort
                    ) Ship_Port
                Group by Ship_Port.IdShip) ShM1) ShM,
                (Select Count (Ship_Port.IdPort) as CntP, Ship_Port.IdShip
                From (Select sc.IdShip,sc.IdPort  From ShipsCargoes sc,
                      ( Select Sum (Quantity) as SummC, IdShip
                         From ShipsCargoes
                         Group by IdShip
                       ) Ship_Quantity,
                      ( Select Max (SummC) as maxSum
                         From ( Select Sum (Quantity) as SummC, IdShip
                         From ShipsCargoes
                         Group by IdShip
                         ) Sh1
                       )
                     Where Ship_Quantity.SummC = maxSum and sc.IdShip = Ship_Quantity.IdShip
                      Group by sc.IdShip, sc.IdPort
                    ) Ship_Port
                Group by Ship_Port.IdShip) ShM2
                where ShM.minCntp=ShM2.cntp) ShN
              where ShipsCargoes.IdShip=ShN.IdShip) ShY
            where Cargoes.IdCargo=ShY.IdCargo
```

Определим теперь БД DIM и запрос к ней. Она описывается тремя классами (подобно таблицам реляционной БД):

Ships с параметрами *IdShip, ShipName*,
Cargoes с параметрами *IdCargo, CargoName* и
Ports с параметрами *IdPort, PortName*, а также классом связи
Ships_Cargoes с параметрами *IdShip_Cargo, IdPort, Quantity*, который имеет родителем класс **Ports**.

Между классами **Ships** и **Cargoes** устанавливается отношение включения с классом связи **Ships_Cargoes**. Запрос на ODQL имеет достаточно понятную (с точки зрения задачи) структуру:

```
select c.CargoName from Cargoes c, Ships_Cargoes sc,
 ( (select objmaxsum(sc.Quantity) on s from Ships s
                links s contains (sc) c) s1
  intersection
   (select objmincount(p.obj) on s1 from s1, Ports p
                links s1 contains (sc) c, p parent sc)) s2
 links s2 contains (sc) c
```

Реляционный запрос к метауровню СУБД DIM (назовем его SQL_DIM) выглядит следующим образом.

```
select v.value from
 validcargo v,
 (select distinct cargoes1.id_cargo from
  (select scq.id_ship,scq.id_cargo from
   ship_cargo_quantity scq) cargoes1,
  (select p3.id_ship from
   (select count(id_port) ports,id_ship from
    (select distinct p.id_port,p1.id_ship from
     ports_quantity p,
     (select s2.id_ship,s2.id_quantity from
      (select scq.id_ship,scq.id_quantity from
       ship_cargo_quantity scq) s2,
      (select s1.id_ship from
       (select scq.id_ship,sum(v.value) summ from
       validquantity v,ship_cargo_quantity scq
       where scq.id_quantity=v.idobject group by id_ship) s1,
       (select max(summ) as max from
        (select scq.id_ship,sum(v.value) summ from
        validquantity v,ship_cargo_quantity scq
        where scq.id_quantity=v.idobject group by id_ship) s) m
      where s1.summ=m.max) s3
     where s2.id_ship=s3.id_ship) p1
    where p.id_quantity=p1.id_quantity)
   group by id_ship) p3,
   (select min(ports) ports from
    (select count(id_port) ports,id_ship from
     (select distinct p.id_port,p1.id_ship from
      ports_quantity p,
      (select s2.id_ship,s2.id_quantity from
       (select scq.id_ship,scq.id_quantity from
        ship_cargo_quantity scq) s2,
       (select s1.id_ship from
        (select scq.id_ship,sum(v.value) summ from
        validquantity v,ship_cargo_quantity scq
        where scq.id_quantity=v.idobject group by id_ship) s1,
```

```
(select max(summ) as max from
  (select scq.id_ship,sum(v.value) summ from
  validquantity v,ship_cargo_quantity scq
  where scq.id_quantity=v.idobject group by id_ship) s) m
 where s1.summ=m.max) s3
 where s2.id_ship=s3.id_ship) p1
 where p.id_quantity=p1.id_quantity)
 group by id_ship)) p4
 where p3.ports=p4.ports) ships1
 where cargoes1.id_ship=ships1.id_ship) cargoes2
where cargoes2.id_cargo=v.idobject
```

Заметим, что для преобразования ODQL-запроса в SQL_DIM-запрос можно написать процедуру преобразования. Но это будет иметь смысл, если время выполнения такого запроса будет существенно лучше времени выполнения ODQL-запроса. Выяснение этого и имеет смысл настоящей работы.

3. Выбор параметров тестов для проведения вычислительных экспериментов

При выборе параметров проведения вычислительных экспериментов нужно выделить три уровня сложности заполнения таблиц: минимальное заполнение (каждый корабль везет один груз в один порт), среднее заполнение (каждый корабль везет 20 % наименований грузов в 20 % портов), максимальное заполнение (каждый корабль везет грузы всех наименований во все порты). Далее при проведении тестов необходимо постепенно увеличивать количество записей в таблицах, содержащих информацию о кораблях, грузах и портах.

4. Результаты вычислительных экспериментов и их анализ

Для начала введем обозначения для проводимых запросов:

1. Запрос для реляционной БД обозначим «SQL».

2. ODQL запрос к БД DIM, который выполняется с помощью технологии индекс-выборок назовем «ODQL».

3. Тот же самый ODQL-запрос, который выполняется с помощью SQL-запроса к метауровню назовем «SQL_DIM».

Таблица 1. Сравнительная таблица времени выполнения запросов при разной сложности заполнения таблиц БД

Параметры данных			Данные с минимальным временем			Данные со средним временем			Данные с максимальным временем		
Количество грузов	Количество кораблей	Количество портов	SQL	SQL_DIM	ODQL	SQL	SQL_DIM	ODQL	SQL	SQL_DIM	ODQL
50	50	50	0.0098	0.016	0.031	0.013	0.024	0.1188	0.110	0.3678	10.3618
100	50	50	0.0192	0.031	0.059	0.028	0.055	0.4068	0.354	1.3016	66.9158
100	100	50	0.0292	0.048	0.089	0.048	0.101	0.9832	0.848	3.1672	184.2448
100	100	100	0.0394	0.064	0.120	0.081	0.178	2.2398	1.801	6.7844	734.745
200	200	100	0.05	0.081	0.158	0.187	0.474	11.9652	-	-	-

Так как время «ODQL» растет катастрофически быстро, то приведем еще 1 таблицу для сравнения «SQL» и «SQL_DIM».

Таблица 2. Сравнительная таблицы времени выполнения запросов SQL и SQL_DIM

Параметры данных			Данные со средним временем	
Количество грузов	Количество кораблей	Количество портов	SQL	SQL_DIM
50	50	30	0.018	0.0346
75	50	30	0.034	0.065
75	75	30	0.0534	0.1002
75	75	45	0.0706	0.145
112	112	45	0.094	0.2186
112	168	67	0.1368	0.3592
168	168	100	0.2492	0.7364
252	252	150	0.6116	2.1286
378	252	150	1.1398	4.2288
378	378	150	1.944	7.5096

По результатам тестирования видно, что ODQL запрос существенно проигрывает SQL, но запрос к метауровню медленнее в среднем в 3 раза. Это позволяет сделать выбор в пользу реляционного запроса к метауровню, и хотя здесь имеется проигрыш по времени, но при этом мы получаем ряд преимуществ, которые нам предоставляет ODQL DIM.

Библиографический список

1. Гарсиа-Молина, Г. Системы баз данных. Полный курс [Текст]: пер. с англ. / Г. Гарсиа-Молина, Дж. Д. Ульман, Дж. Уидом. – М. : Издательский дом «Вильямс», 2003. – 1088 с.

2. Дейт, К. Дж. Основы будущих систем баз данных: третий манифест [Текст]: пер. с англ. / К. Дж Дейт, Хью Дарвен. – 2-е изд. – М. : Янус-К, 2004. – 656 с.

3. Писаренко, Д.С. Объектная СУБД Динамическая информационная модель DIM и ее основные концепции [Текст] / Д. С. Писаренко, В. С. Рублёв // Моделирование и анализ информационных систем. Т. 16. − Ярославль : изд-во ЯрГУ им. П. Г. Демидова, 2009. − С. 62−91.

4. Рублёв, В. С. Запросная полнота языка ODQL динамической информационной модели DIM [Текст] / В. С. Рублёв // Ярославский педагогический вестник. Серия «Физико-математические и естественные науки». Вып.1. – Ярославль : ЯГПУ, 2011. − С. 69−75.

5. Рублёв, В. С. Организация выполнения объектных запросов в динамической информационной модели DIM [Текст] / В. С. Рублёв // Моделирование и анализ информационных систем. Т. 18. – Ярославль : ЯрГУ им. П. Г. Демидова. − 2011. – № 2. − С. 39–51.

6. Рублёв, В. С. Язык объектных запросов динамической информационной модели DIM [Текст] / В. С. Рублёв // Моделирование и анализ информационных систем. Т.17. – Ярославль : ЯрГУ им. П. Г. Демидова. – 2010. − № 3. − С. 144−161.

7. Рублёв, В. С. Организация хранения данных и выполнения запросов в динамической информационной модели DIM [Текст] / В. С. Рублёв, А. Ш. Кайбышев // Ярославский педагогический вестник. Том III (Естественные науки). – Ярославль : ЯГПУ им. К. Д. Ушинского. – 2012. – № 1. с. 7−20.

8. M.J arke, J. Koch, Query Optimization in Database Systems // Computing Surveys, Vol.16, No. 2, June 1984.

Bibliograficheskij spisok

1. Garsia-Molina, G. Sistemy baz dannyh. Polnyj kurs [Tekst]: per. s angl. / G. Garsia-Molina, Dzh. D. Ul'man, Dzh. Uidom. – M. : Izdatel'skij dom «Vil'jams», 2003. – 1088 s.

2. Dejt, K. Dzh. Osnovy budushhih sistem baz dannyh: tretij manifest [Tekst]: per. s angl. / K. Dzh Dejt, H'ju Darven. – 2-e izd. – M. : Janus-K, 2004. – 656 s.

3. Pisarenko, D.S. Ob”ektnaja SUBD Dinamicheskaja informacionnaja model' DIM i ee osnovnye koncepcii [Tekst] / D. S. Pisarenko, V. S. Rubljov // Modelirovanie i analiz informacionnyh sistem. T. 16. − Jaroslavl' : izd-vo JarGU im. P. G. Demidova, 2009. − S. 62−91.

4. Rubljov, V. S. Zaprosnaja polnota jazyka ODQL dinamicheskoj informacionnoj modeli DIM [Tekst] / V. S. Rubljov // Jaroslavskij pedagogicheskij vestnik. Serija «Fiziko-matematicheskie i estestvennye nauki». Vyp.1. – Jaroslavl' : JaGPU, 2011. − S. 69−75.

5. Rubljov, V. S. Organizacija vypolnenija ob#ektnyh zaprosov v dinamicheskoj informacionnoj modeli DIM [Tekst] / V. S. Rubljov // Modelirovanie i analiz informacionnyh sistem. T. 18. – Jaroslavl' : JarGU im. P. G. Demidova. − 2011. – № 2. − S. 39–51.

6. Rubljov, V. S. Jazyk ob’ektnyh zaprosov dinamicheskoj informacionnoj modeli DIM [Tekst] / V. S. Rubljov // Modelirovanie i analiz informacionnyh sistem. T.17. – Jaroslavl' : JarGU im. P. G. Demidova. – 2010. − № 3. − S. 144−161.

7. Rubljov, V. S. Organizacija hranenija dannyh i vypolnenija zaprosov v dinamicheskoj informacionnoj modeli DIM [Tekst] / V. S. Rubljov, A. Sh. Kajbyshev // Jaroslavskij pedagogicheskij vestnik. Tom III (Estestvennye nauki). – Jaroslavl' : JaGPU im. K. D. Ushinskogo. – 2012. – № 1. s. 7−20.

8. M. Jarke, J. Koch, Query Optimization in Database Systems // Computing Surveys, Vol.16, No. 2, June 1984.