TPC-Hベンチマーク Query8 資料

1051090 福澤優

2011.11

まず TPC-H で定義されている Query8 の SQL 文を以下に示す.

```
select
    o_year,
    sum(case
        when nation = '[NATION]'
        then volume
        else 0
   end) / sum(volume) as mkt_share
from (
    select
        extract(year from o_orderdate) as o_year,
        l_extendedprice * (1-l_discount) as volume,
        n2.n_name as nation
    from
        part,
        supplier,
        lineitem,
        orders,
        customer,
        nation n1,
        nation n2,
        region
    where
        p_partkey = l_partkey
        and s_suppkey = l_suppkey
        and l_orderkey = o_orderkey
        and o_custkey = c_custkey
        and c_nationkey = n1.n_nationkey
        and n1.n_regionkey = r_regionkey
        and r_name = '[REGION]'
        and s_nationkey = n2.n_nationkey
        and o_orderdate between date '1995-01-01' and date '1996-12-31'
        and p_type = '[TYPE]'
```

```
    ) as all_nations
group by
    o_year
order by
    o_year;
```

今回のシステムにおいて，集約演算中に条件文が用いることが出来ないため，今回はパラメータを具体化して以下のクエリを考える．

```
select
    o_year,
    sum(volume)
from (
    select
        extract(year from o_orderdate) as o_year,
        l_extendedprice * (1-l_discount) as volume,
        n2.n_name as nation
    from
        part,
        supplier,
        lineitem,
        orders,
        customer,
        nation n1,
        nat\begin{center}
\begin{verbatim}ion n2,
        region
    where
        p_partkey = l_partkey
        and s_suppkey = l_suppkey
        and l_orderkey = o_orderkey
        and o_custkey = c_custkey
        and c_nationkey = n1.n_nationkey
        and n1.n_regionkey = r_regionkey
        and r_name = '[REGION]'
        and s_nationkey = n2.n_nationkey
        and o_orderdate between date '1995-01-01' and date '1996-12-31'
        and p_type = '[TYPE]'
    ) as all_nations
group by
    o_year
order by
    o_year;
```

次に SQL 文に対する PostgreSQl の EXPLAIN 文での実行計画を以下に示す．ここでは，イン

デックが無い状態での実行計画である.

```
                                QUERY PLAN
-------------------------------------------------------------------------------
GroupAggregate  (cost=265150.12..265152.28 rows=72 width=20)
 ->Sort  (cost=265150.12..265150.30 rows=72 width=20)
   Sort Key: (date_part('year'::text,
   (orders.o_orderdate)::timestamp without time zone))
   ->Hash Join  (cost=63938.18..265147.90 rows=72 width=20)
     Hash Cond: (supplier.s_nationkey = n2.n_nationkey)
     ->Hash Join  (cost=63924.35..265132.72 rows=72 width=24)
       Hash Cond: (lineitem.l_suppkey = supplier.s_suppkey)
       ->Hash Join  (cost=63474.35..264680.92 rows=72 width=24)
         Hash Cond: (lineitem.l_partkey = part.p_partkey)
         ->Hash Join  (cost=56803.69..257968.92 rows=10832 width=28)
           Hash Cond: (lineitem.l_orderkey = orders.o_orderkey)
           ->Seq Scan on lineitem  (cost=0.00..178551.77
           rows=6001377 width=28)
           ->Hash(cost=56769.85..56769.85 rows=2707 width=8)
             ->Hash Join  (cost=5727.84..56769.85 rows=2707 width=8)
                Hash Cond: (orders.o_custkey = customer.c_custkey)
                ->Seq Scan on orders  (cost=0.00..49289.00
                rows=460253 width=12)
                  Filter: ((o_orderdate >= '1995-01-01'::date)
                  AND (o_orderdate <= '1996-12-31'::date))
                ->Hash  (cost=5716.82..5716.82 rows=882 width=4)
                  ->Hash Join  (cost=24.50..5716.82 rows=882 width=4)
                    Hash Cond: (customer.c_nationkey = n1.n_nationkey)
                    ->  Seq Scan on customer  (cost=0.00..5121.00
                    rows=150000 width=8)
                    ->  Hash  (cost=24.48..24.48 rows=1 width=4)
                      ->  Hash Join  (cost=12.14..24.48 rows=1 width=4)
                          Hash Cond: (n1.n_regionkey = region.r_regionkey)
                          ->  Seq Scan on nation n1  (cost=0.00..11.70
                          rows=170 width=8)
                          ->  Hash  (cost=12.12..12.12 rows=1 width=4)
                              ->  Seq Scan on region  (cost=0.00..12.12
                              rows=1 width=4)
                                  Filter: (r_name = '[REGION]'::bpchar)
         -> Hash  (cost=6654.00..6654.00 rows=1333 width=4)
            ->  Seq Scan on part  (cost=0.00..6654.00 rows=1333 width=4)
              Filter: ((p_type)::text = '[TYPE]'::text)
    ->  Hash  (cost=325.00..325.00 rows=10000 width=8)
```

```
        ->  Seq Scan on supplier  (cost=0.00..325.00 rows=10000 width=8)
  ->  Hash  (cost=11.70..11.70 rows=170 width=4)
     ->  Seq Scan on nation n2  (cost=0.00..11.70 rows=170 width=4)
------------------------------------------------------------------------------
```

この実行計画の記述を木構造表現で表すと図 1 となる.

PosgreSQL のクエリ実行計画を参考に今回のシステムでのクエリ実行木を行指向データに対するものを図 2 に，列指向データに対するものを図 3 に示す.

最後に，これらの実行プランから得られた実行時間を表 1 に示す.

表 1: Query8 の実行時間

	Posgre(インデックスなし)			Posgre(インデックスあり)		
real	23.541	23.385	25.573	41.814	44.636	41.446
user	0.03	0.04	0.02	0.02	0.02	0.02
sys	0.01	0.00	0.02	0.02	0.02	0.02

	行指向			列指向		
real	35.504	35.521	35.736	48.220	48.729	48.718
user	31.15	31.08	31.48	44.62	45.14	45.07
sys	3.61	3.66	3.50	3.06	3.04	3.18

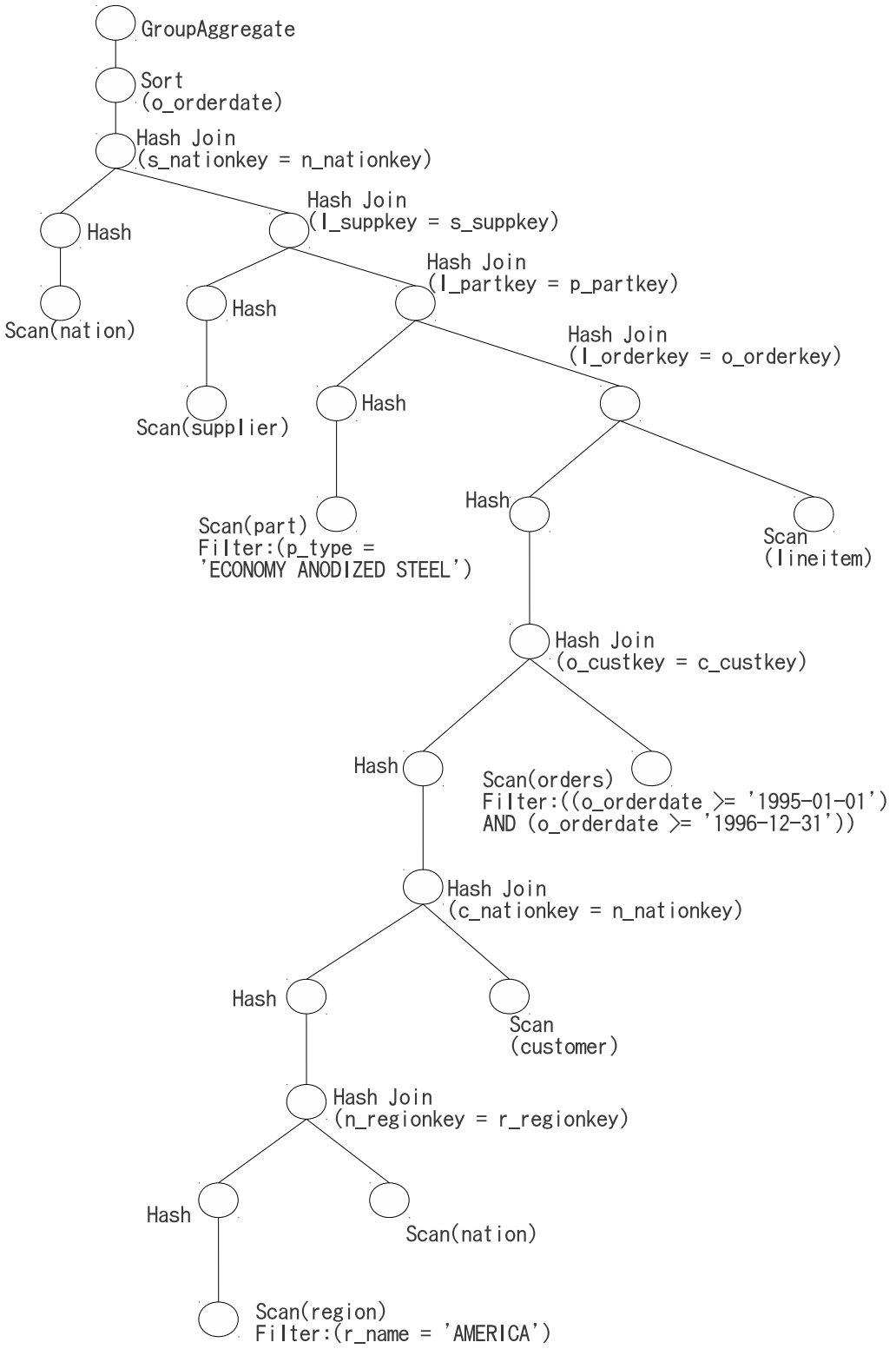

図 1: PosgreSQL でのクエリ実行木

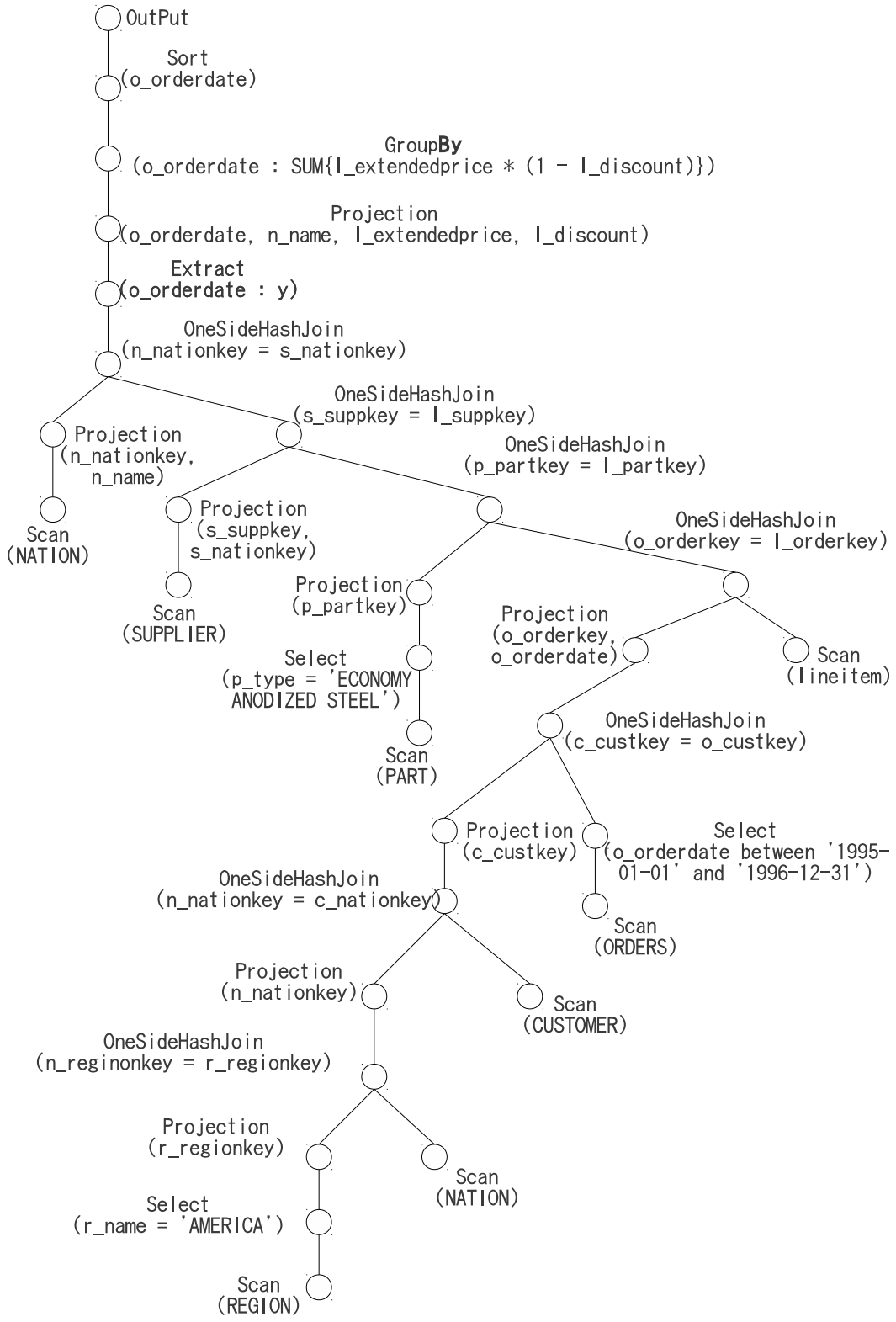

図 2: 行指向データに対するクエリ実行木

図 3: 列指向データに対するクエリ実行木