Projekt – Ein Query-Optimizer für Cloudsysteme

Projektleiter: Foo Bar
Team: Autor 1, Autor 2

Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

14. Oktober 2009

Agenda

Projektziele

Effiziente Implementierung relationaler Joins im Hadoop-Framework

- 2 Verfahren
  - Realisierung ohne Veränderung des Hadoop-Frameworks
  - Erweiterung des Hadoop-Frameworks um einen Merge-Operator
- Tests der Verfahren gegeneinander

Agenda

Repartition Join

- Equi-Join über 2 Relationen
- Mapper: wird jedes Tupel einmal weitergegeben oder ausgefiltert
- Partitioner: Modulo-Division des Schlüssel-HashWerts

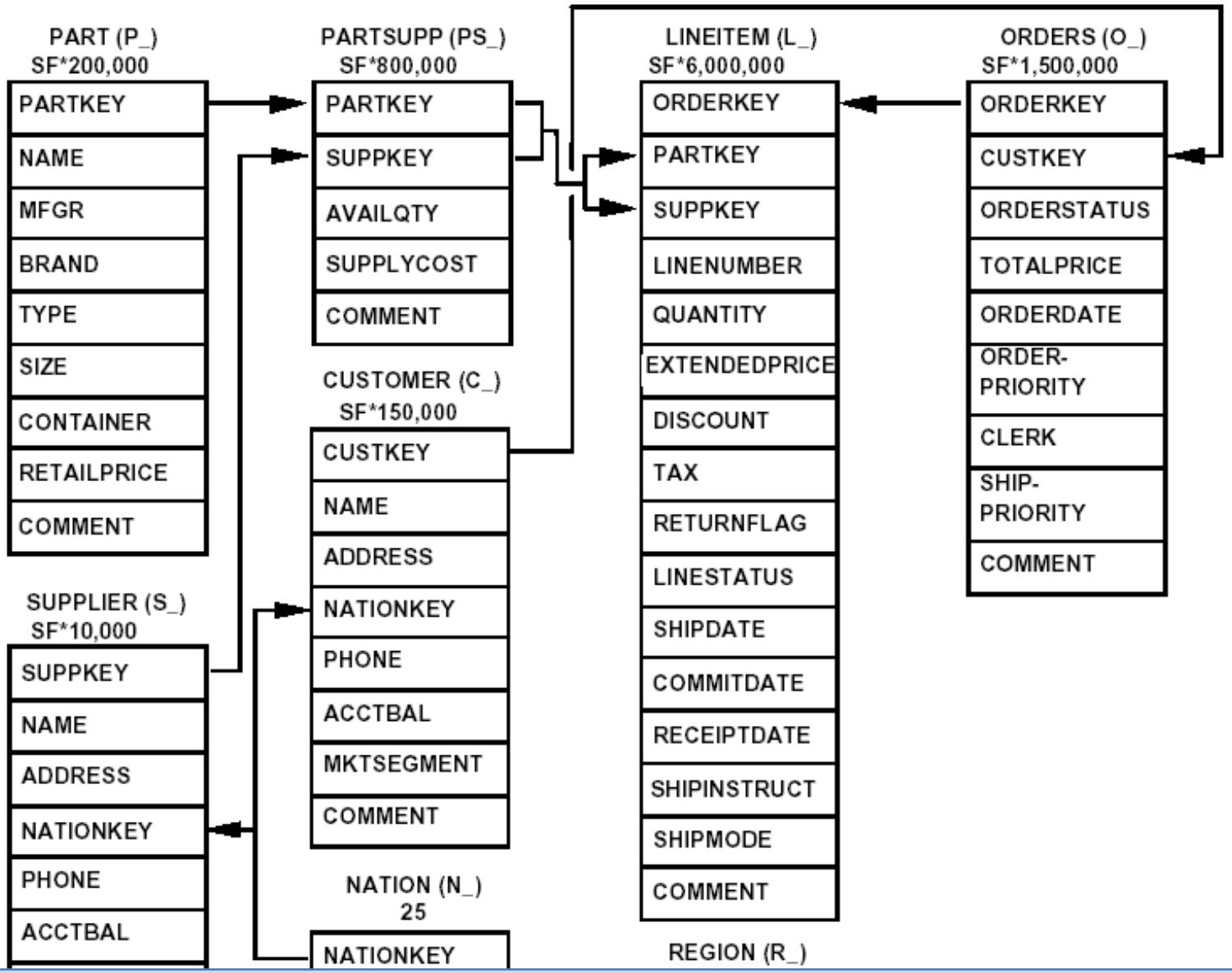
```
public void map(...) throws IOException
{
    RelationTuple tuple = new RelationTuple(value);
    ...
    if (!filter.eval(tuple)) output.collect(outputKey, tuple);
}
```

```
public int getPartition(RepartitionKey k, RelationTuple v, int ptns)
{
    return Math.abs(k.getKey().hashCode() % ptns);
}
```

Repartition Join

- Sort:
  - primär nach Key
  - sekundär nach Relation (*L* Relation kommt zuerst)
- Reduce:
  - alle *L*-Tupel zum aktuellen Schlüssel sammeln
  - mit allen passenden *R*-Tupel kombinieren (kommen gleich danach)

TPCH Schema


PART (P_)
SF*200,000
PARTKEY
NAME
MFGR
BRAND
TYPE
SIZE
CONTAINER
RETAILPRICE
COMMENT
PARTSUPP (PS_)
SF*800,000
PARTKEY
SUPPKEY
AVAILQTY
SUPPLYCOST
COMMENT
LINEITEM (L_)
SF*6,000,000
ORDERKEY
PARTKEY
SUPPKEY
LINENUMBER
QUANTITY
EXTENDEDPRICE
DISCOUNT
TAX
RETURNFLAG
LINESTATUS
SHIPDATE
COMMITDATE
RECEIPTDATE
SHIPINSTRUCT
SHIPMODE
COMMENT
ORDERS (O_)
SF*1,500,000
ORDERKEY
CUSTKEY
ORDERSTATUS
TOTALPRICE
ORDERDATE
ORDER-PRIORITY
CLERK
SHIP-PRIORITY
COMMENT
CUSTOMER (C_)
SF*150,000
CUSTKEY
NAME
ADDRESS
NATIONKEY
PHONE
ACCTBAL
MKTSEGMENT
COMMENT
SUPPLIER (S_)
SF*10,000
SUPPKEY
NAME
ADDRESS
NATIONKEY
PHONE
ACCTBAL
NATION (N_)
25
NATIONKEY
REGION (R_)

Vielen Dank für Ihre Aufmerksamkeit

Fragen?