

Fachpraktikum “Erweiterbare Datenbanksysteme” im WS 2017/18

Aufgabe der Phase 2

Eine Algebra für verteilte Relationen

Ralf Hartmut Güting, Thomas Behr, Holger Helmut Hennings, Fabio Valdés

Lehrgebiet Datenbanksysteme für neue Anwendungen

Fakultät für Mathematik und Informatik, Fernuniversität in Hagen

21.11.2017

Einführung

Verteilte Datenbanken und Anfrageauswertung möglich mit Secondo mit Hilfe der Distributed Algebra (genauer: Distributed2Algebra).

Secondo-System (Master) kontrolliert viele Secondo-Server (Worker) auf verschiedenen Rechnern. Master

- verteilt Daten
- veranlaßt parallele Berechnungen durch Worker (Abbildungen verteilter Daten)
- sammelt verteilte Daten (Ergebnisse) wieder ein

Anfragen auf ausführbarer Ebene formuliert auf Basis *verteilter Arrays*. Nicht einfach zu benutzen.

Aufgabe

Einfacher zu benutzende Algebra für verteilte Relationen

- neuer Datentyp für verteilte Relation enthält Info über Art der Verteilung. Automatisches Umverteilen, falls nötig
- Automatische Benutzung von Indexen

Distributed Algebra

- Zwei Ebenen
- Untere Ebene: Primitive
 - Verbinden mit Server (Worker)
 - beliebige Secondo-Befehle an einzelne Server schicken (Server über Nr. ansprechbar)
 - Secondo-Befehle zur parallelen Ausführung verschicken
 - Dateitransfer zwischen Servern bzw. Master/Server
 - Tupelströme partitionieren in Dateien
 - ...
- Obere Ebene: Abstraktion *verteilter Array* mit Operationen
 - Typen darray, dfarray, dfmatrix
 - darray: Felder enthalten Werte beliebiger Secondo-Typen (Relationen, Indexe, atomare Werte, ...). In Datenbanken der Worker gespeichert.
 - dfarray, dfmatrix: Felder enthalten Relationen. In Dateien der Worker gespeichert.

Distributed Algebra

Verteilter Array (*darray*, *dfarray*)

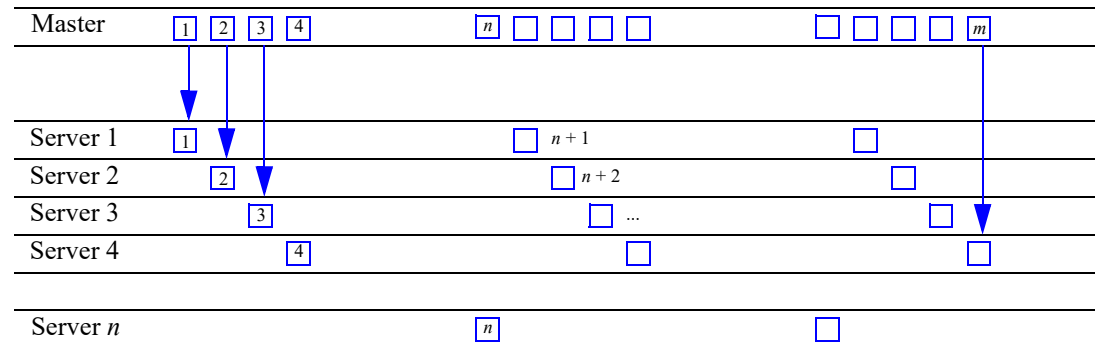


Figure 1: Creating a distributed array by partitioning data on the master.

Verteilte Matrix (*dfmatrix*)



Figure 2: A distributed file matrix.

Distributed Algebra

Matrix kann spaltenweise in verteilten Array eingesammelt werden.

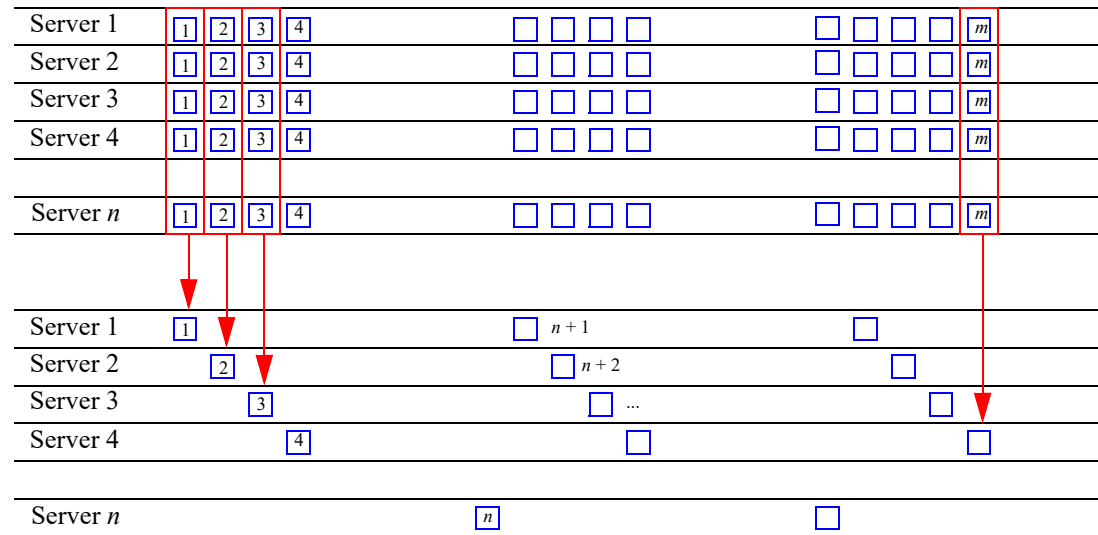


Figure 3: A distributed file matrix is collected into a distributed file array.

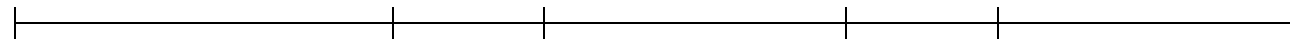
Distributed Algebra

- Obere Ebene: Operationen
 - **ddistribute, dfdistribute**: Tupelstrom verteilen; Ergebnis verteilter Array
 - **dloop, dmap**: jedes Feld bearbeiten (abbilden); Ergebnis neuer verteilter Array
 - **dloop2, dmap2**: für zwei Argumentarrays Felderpaare mit gleichem Index in Query bearbeiten (wichtig für Joins); Ergebnis neuer verteilter Array
 - **dproduct**: kombiniert für zwei Argumentarrays von Relationen jedes Feld des ersten Arrays mit *der Vereinigung aller Felder* des zweiten Arguments. Benötigt für allgemeine Joins.
 - **partition**: verteilten Array auf jedem Worker neu partitionieren (dfarray -> dfmatrix)
 - **collect2**: Matrix spaltenweise einsammeln (dfmatrix -> dfarray)
 - **areduce, areduce2**: eine oder zwei Matrizen dynamisch spaltenweise verarbeiten (dfmatrix -> dfarray, dfmatrix x dfmatrix -> dfarray)
 - **getValue, tie**: atomare Ergebnisse aus verteiltem Array auf den Master übertragen, dort aggregieren
 - **dsummarize**: Array von Relationen als Tupelstrom auf den Master bringen

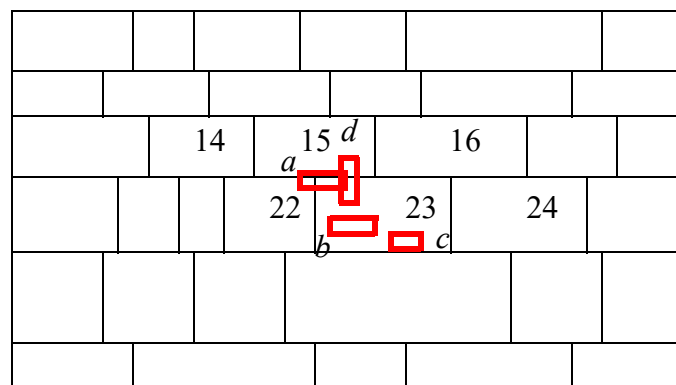
Verteilte Relation

Verteilungsarten

- Relationen können partitioniert oder repliziert verteilt sein
- Arten der Partitionierung
 - *random*: sequentiell round robin
 - *hash(X)*: nach Hashwert von Standardattribut X
 - *range(X)*: Wertebereich von X aufgeteilt (eindimensional)



- *spatial2d(X)*:



- *spatial3d(X)*

Verteilte Relation

Speicherungsformen

Verteilter Array (*darray*) von

- *rel*: persistenten Relationen
- *mrel*: Hauptspeicher-Relationen

Verteilter Array von Dateien (*dfarray*)

- *file*

Typ

Enthält Tupeltyp, Verteilungstyp, Speicherungsform

Wert

Enthält Name, Workerliste, Abbildung Felder → Worker, Partitionierungs-Datenstruktur

Operationen auf verteilten Relationen

Verteilen	drdistribute
Verarbeiten	drmap drselect drequijoin drspatialjoin drgenjoin drsort drgroupby
Einsammeln	drsummarize
Indexe	drcreateindex drdeleteindex
Konvertieren	drgetdarray drcreatedrel rename

Operationen auf verteilten Relationen

Selektion

```
query RoadsR drselect [; Name, "Universitätsstraße"; .]
```

```
query CitiesR drselect [; Population, 300000, 500000; . count]
```

```
let BuildingsHombruch =  
  BuildingsR drselect ["BuildingsHombruch", mrel; GeoData, hombruch; .  
  filter[.Geodata inside hombruch]]
```

Defaults

```
"", rel;
```

Mögliche Indexe

B-Baum oder R-Baum (*rel*), AVL-Baum oder MMR-Baum (*mrel*)

```
RoadsR_Name_btree  
CitiesR_Population_btree  
BuildingsR_GeoData_rtree
```

Werden verwendet, falls existent.

Operationen auf verteilten Relationen

Equijoin

```
select * from [Natural as n1, Natural as n2]
where [n1:Name = n2:Name, n1:Osm_id < n2:Osm_id]
```

```
query Natural {n1} Natural {n2}
  drequijoin[; Name_n1, Name_n2; . filter[.Osm_id_n1 < .Osm_id_n2] ]
```

Spatial Join

```
select count(*) from [Roads as r, Waterways as w]
where r:GeoData intersects w:GeoData
```

```
query RoadsR {r} WaterwaysR {w}
  drspatialjoin[; GeoData_r, GeoData_w; . filter[.GeoData_r intersects
  .GeoData_w] count]
```

Operationen auf verteilten Relationen

Spatial Join

```
select [r:Osm_id, r:Name, w:Osm_id, w:Name, intersection(r:GeoData, w:GeoData)
       as BridgePosition]
from [Roads as r, Waterways as w]
where r:GeoData intersects w:GeoData
```

```
query RoadsR {r} Waterways {w}
  drspatialjoin[; GeoData_r, GeoData_w; .
  filter[.GeoData_r intersects .GeoData_w]
  extend[BridgePosition: intersection(GeoData_r, GeoData_w)]
  project[Osm_id_r, Name_r, Osm_id_w, Name_w, BridgePosition]
]
```

Automatisches Umverteilen, falls nötig

Indexe benutzt, falls existent und vorteilhaft (B-Baum bzw. R-Baum).

Spatial Join vermeidet Duplikate.

Operationen auf verteilten Relationen

Allgemeiner Join

```
select * from [Roads as r, Waterways as w]
where r:Name contains w:Name
```

```
query RoadsR {r} Waterways {w}
  drngenjoin[; .Name_r contains ..Name_w; .]
```

Operationen auf verteilten Relationen

Konvertieren

drgetarray

drcreatedrel

rename

Duplikatvermeidung im Spatial Join

