Ad-Hoc User-Defined Functions for MonetDB with R

Hannes Mühleisen

useR! 2015, 2015-07-01
Which Systems?

- Statistical Toolkits
- Data Management Systems

Flexibility vs. Efficiency
Bridge the Gap

+ Native operators, lazy evaluation

+ Cheap data transfer
Previously

MonetDB.R connector
DBI, dplyr backend
Now

Embedded R in MonetDB
Part of MonetDB distribution since 2014
## Postgres, Oracle, DB2, etc.:

### Conceptional

<table>
<thead>
<tr>
<th>class</th>
<th>speed</th>
<th>flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>NX</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Constitution</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Galaxy</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Defiant</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Intrepid</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Physical (on Disk)

<table>
<thead>
<tr>
<th>NX</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Defiant</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Constitution</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Galaxy</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Intrepid</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Column Store:

<table>
<thead>
<tr>
<th>class</th>
<th>speed</th>
<th>flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>NX</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Constitution</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Galaxy</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Defiant</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Intrepid</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NX</th>
<th>Constitution</th>
<th>Galaxy</th>
<th>Defiant</th>
<th>Intrepid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Performance...

TPC-H SF-100 Hot runs

Average time (s)

Query

log!
Relationally Integrated

Statistical analysis as operators in relational queries
CREATE FUNCTION rapi01(i INTEGER)
RETURNS TABLE (i INTEGER, d DOUBLE)
LANGUAGE R { data.frame(i=seq(1,i),d=42.0) };

SELECT i,d FROM rapi01(42) AS r WHERE i>40;
CREATE FUNCTION rapi02 (i INTEGER,
    j INTEGER, z INTEGER) RETURNS INTEGER
LANGUAGE R 
{
    i*sum(j)*z
};

SELECT rapi02(i,j,2) AS r02 FROM rval;
CREATE FUNCTION rapi03(i INTEGER, z INTEGER)
RETURNS BOOLEAN LANGUAGE R 
{ i>z }
;

SELECT * FROM rval WHERE rapi03(i,2);
CREATE AGGREGATE kmeans(data FLOAT, ncluster INTEGER) RETURNS INTEGER
LANGUAGE R { kmeans(data,ncluster)$cluster };

SELECT cluster FROM (SELECT MIN(x) AS minx, MAX(x) AS maxx, kmeans(x,5) AS cluster FROM xdata GROUP BY cluster) as cdata ORDER BY cluster;
Performance...
Code Shipping

\[ \text{rf.fit} \leftarrow \text{randomForest}(income \sim ., \\
\text{data}=\text{training}, \text{mtry}=2, \text{ntree}=10) \]

\[ \text{predictions} \leftarrow \text{mdbapply}(\text{con}, \text{"t1"}, \\
\text{function}(d) \{ \\
\text{p} \leftarrow \text{predict}(\text{rf.fit}, \text{type}="\text{prob}", \\
\text{newdata}=d)[,2] \\
\text{p}[\text{p} > .9] \\
\}) \]
Demo

```r
> system.time(mdbapply(con, "flights", summary))
  user  system elapsed
0.013  0.000   0.654
> system.time(summary(dbReadTable(con, "flights")))
  user  system elapsed
1.756  0.165   3.419
```
Thank You
Questions?

http://www.monetdb.org

@hfmuehleisen