MapReduce With Columnar Storage

Peitsa Lähteenmäki

Topics

- MapReduce
- Hadoop
- Columnar methods

MapReduce

- Programming paradigm/framework
- Two functions: map and reduce
- Map: transform a value to another
- Reduce: combine a set of values to a value
- Completely definable by user
- Automatically distribute these to multiple nodes

Hadoop

- A somewhat easy way to implement MapReduce programs
- Designated nodes for map, reduce and storage
  - File transmissions probable
- Done with Java
  - Slow de-serialization

IO in Hadoop

- Both situate (by default) in files
  - But can be configured by the user to something else
- Usually in TXT format
  - Clearly not the best choice

Columnar methods

- Ways to improve Hadoop's shortcomings
- Single column file
  - Store each column of input to a separate binary file
  - Split to multiple parts to avoid large transfers
  - Also possible to group frequently used columns together

Columnar methods

- Materialization
  - Early vs. late
  - First method: read all, de-serialize only when accessed
  - Second method: use map task to find out what to read and move processing to reduce

Example

- Assume a MapReduce task equivalent to the query:

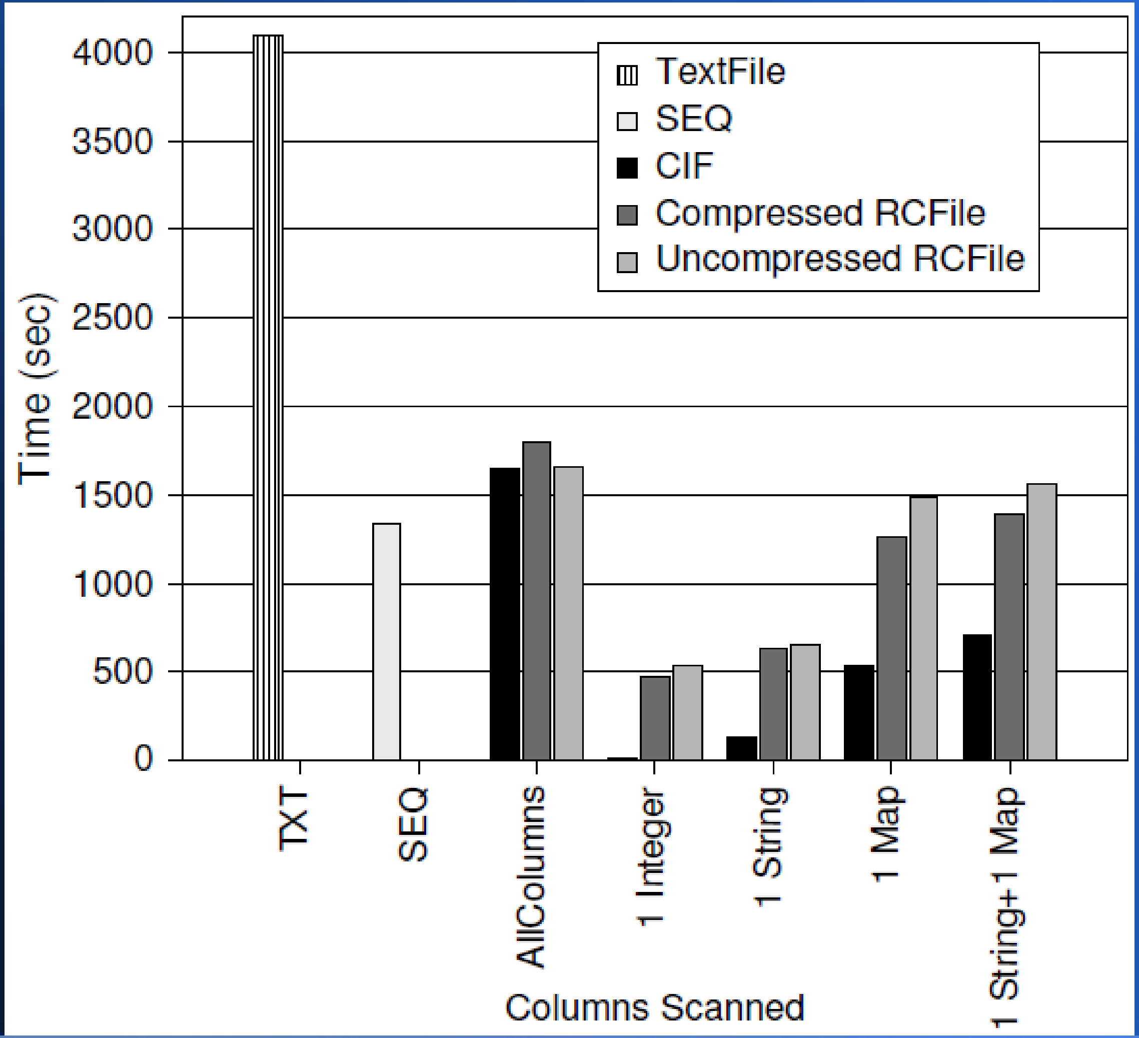
```
SELECT A, B, C, D
FROM table
WHERE A < 1000
```

- Use map to process where clause, reduce to produce the result

Columnar methods

- Compression
  - Reduce file transmissions
  - Increases CPU use, decreases IO costs
  - Note: compression method must allow random access

Results


Time (sec)
4000
3500
3000
2500
2000
1500
1000
500
0
TextFile
SEQ
CIF
Compressed RCFile
Uncompressed RCFile
TXT
SEQ
AllColumns
1 Integer
1 String
1 Map
1 String+1 Map
Columns Scanned