Outline

Previously we outlined functions to select a single column from a parquet file.

This notebook outlines how to reuse those functions to speed up the conversion of a large parquet file.

Distributed each is used to achieve this.

Files

qparquet.q

This file contains needed imports and functions

```
//parquet library prints many warnings - ignore for this example
p)import warnings
p)warnings.filterwarnings("ignore")

//Import pandas, numpy, and pyarrow
p)import pandas as pd
p)import numpy as np
p)import pyarrow as pa
p)import pyarrow.parquet as pq

p)def getColumnNames(file):    return (pq.read_schema(file)).names

getColumnNames:.p.get`getColumnNames

p)def getColumns(file, cols):    table=pq.read_table('file', columns=cols); return
(table.to_pandas()).to_dict('list')

getColumns:.p.get`getColumns

getColumn:{[file;column] first value getColumns[file;enlist column]`}
```

convert.q

This script coordinates distributing the work of converting the parquet file across multiple processes

```
//Load needed functions
\l qparquet.q

//Open handles to worker processes
.z.pd:`u#asc hopen each"J"$(.Q.opt .z.X)`slaves

file:"example.parquet";

columns:-1_getColumnNames[file]`
```

```
destination:`:splayed

//Distribute tasks to workers
//Each worker reads a column at a time
{[f;d;c] .Q.dd[d;`$c] set getColumn[f;c]}[file;destination] peach columns

//Add a .d file to the destination to inform q of the order of columns
.Q.dd[destination;`.d] set `$columns

//Load the converted table
\l splayed

//Query the q table
show select from splayed
```

Running the example

Start your worker processes

```
q qparquet.q - p 5001 &
q qparquet.q - p 5002 &
q qparquet.q - p 5003 &
```

Run the master process to distibute the work

```
q convert.q -s -3 -slaves 5001 5002 5003
```

The output shows that the qparquet data is now successfully a q splayed table

```
`:splayed/one`:splayed/two`:splayed/three
`:splayed/.d
`splayed
one two   three
---------------
-1  "foo" 1
0   "bar" 0
2.5 "baz" 1
```