Introduction to script interface and understanding default parallelism

Click here to download the supporting SQL script

Training Overview – This is the R script version

- Introduction to external script interface
- Controlling Script Processes – threads and nodes
- Controlling Data Processing – partitions
- Passing parameters through to scripts

Kognitio

Introduction to external scripting interface:
using R

```
create external script Kog_R_AvgOverAllRows
environment rscript
receives(part_id int, value decimal(10,6))
input 'column_headers on,
column_header_format 0'
sends(average decimal(10,6))
script
S'EOF(
  read_in<-read.csv(file=file("stdin"),
      header=TRUE)
  if (nrow(read_in)>0){
    avg1 <-sum(read_in$VALUE)/nrow(read_in)
    cat(file="",paste(avg1, sep=",",
      collapse="\n"), "\n")
   }
)EOF';
```

Create script and declare environment

receives – declare the input data types for the script

column headers can be passed as the first row of input. Note
format 0 passes headers only.
Additional input info, such as data type, can be passed using other format options

sends– declare the output data types from the script

script– contained within EOF
imports data, computes average value and writes result out

```
select * from (external script
Kog_R_AvgOverAllRows from (select prodno,
price/100.00 from demo_ret.v_ret_sale where
storeno = 1)) ext;
```

Once script is created any user can invoke it in any SQL query
(! provided privileges are set up on script AND schema)

Kognitio

Introduction to external scripting interface:

some useful R syntax

```
create external script Kog_R_AvgOverAllRows environment
rscript
receives(part_id int, value decimal(10,6))
input 'column_headers on, column_header_format 0'
sends(average decimal(10,6))
script
S'EOF(
   read_in<-read.csv(file=file("stdin"),
      header=TRUE)
   if (nrow(read_in)>0){
      average <-sum(read_in$VALUE)/nrow(read_in)
      cat(file="",paste(average, sep=",",
          collapse="\n"), "\n")
   }
)EOF';

select * from
(external script Kog_R_AvgOverAllRows from
          (select prodno, price/100.00 from
           demo_ret.v_ret_sale
           where storeno = 1)) ext;
```

Use read.csv to import data via stdin. Kognitio uses stdin to stream data into external scripts.
Set header = TRUE to read first input row as the column headers. These were passed by having column_headers set on in the input command

Within **any** script empty input must always be handled as Kognitio does not always send data to every script invocation

Results are output from R via the cat command which can be used to format R objects into row output.
Note file="" sends output to stdout which is used by Kognitio to obtain results from an external script.
Set sep="," as Kognitio expects columns to be comma separated

What does output look like?

Kognitio

Introduction to external scripting interface:
default parallelism

```
19 limit 1 threads
20 requires 3000 mb ram
21 script S'script(
22 read_in<-read.csv(file=file("stdin"), header=TRUE)
23 average <- sum(read_in$VALUE)/nrow(read_in)
24 cat(file="",paste(average, sep=",", collapse="\n"), "\n"
25 )script';
26 select * from (external script r_average from (select prod
27
```

Results (100 rows, limit hit) | Metadata | Info (Aborted)

	AVERAGE
1	1.118298
2	1.124987
3	1.118490
4	1.112094
5	1.118782
6	1.121269
7	1.114953
8	1.122078
9	1.115308
10	1.124272
11	1.118566
12	1.105795

There is not one result row. This is due to the **DEFAULT PARALLELISM**

The number of result rows obtained depends on the number of script invocations that process data.

The default (and maximum) number of invocations is equal to the number of RAM stores on the system. This equals the number of results rows obtained by default

! except when input data is very small. In this case not all script invocations may be used.

Kognitio External Scripting is very flexible

There are many ways to control the processing.
See part 2 for more details

Kognitio