Traffic Loop Data

A case study in RHadoop

Marco Puts, B. Meindl, L. Ma, P. Del Rey

#NTTS2015
library("rmr2")
library("stringr")
mapfunction <- function(k, v)
{
  text = gsub("\n|\[[:punct:]\]", " ", v)
  k = str_split(string = tolower(text), pattern = " ")[1]
  v = rep(1, length(k));
  return (keyval(k, v))
}

reducefunction <- function(k, v)
{
  key = unique(k)
  value = sum(v)
  return (keyval(key, value))
}

data = to.dfs(text);
res = mapreduce(
  input = data,
  map = mapfunction,
  reduce = reducefunction,
  combine = TRUE
)

wordcount = from.dfs(res)
Road sensors

Road sensor data
- Passing vehicle counts for each minute (24/7) at about 60,000 sensors in the Netherlands
- Types of sensors:
  - Induction loop
  - Camera
  - Bluetooth
- Length categories (e.g. small, medium, long vehicles)
- Large volume
Dutch road sensors
Dutch road sensors
What about all data

Size in Bytes
- 1 TB (compressed) per year
- 2.7 GB (compressed) per day

Size in Records
- 84 Billion records a year
- 230 Million records a day
Data Size

- Raw Data: 4 TB (2010 - 2014)
- Transformed Data: 60 GB
- Clean Data: 1 GB
- Traffic Index: 6 KB
Reduce the Volume of the Data

Select
- Only necessary variables
- On the main routes (without ramps)

Transform
- Put one day in one record
Experiment at Sandbox

Transport data to Ireland
- Mailing a NAS server serves more bandwidth than internet, hence faster (and cheaper)

Processing on RHadoop
- Slow
  • We used gnuzipped files
  • Chunks of data not optimal
Traditionally:
- 80:20 rule automated/manual editing

Big Data:
- Manual editing on micro level impossible
- Alternative: Digital Filtering
Cleaning the Data
Recursive Bayesian Estimation

X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4

Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Y_4

state
observation

Smoothing
Filtering

Hadoop not the first choice
- Data already ordered
- Running the process on locally available datasets is better

Spark!
Filtering

- `scp` chunks of data to different nodes
- Process chunks in parallel using `ssh`
Conclusion

- RHadoop = R + Hadoop
  - Pro: Nice combination
  - Con: R only fast with dedicated packages
- Big data processing brings Big data challenges