No description, website, or topics provided.

README.md

50.043 Project

Description

In this project, you will build a web application for Kindle book reviews, one that is similar to Goodreads. You will start with some public datasets from Amazon, and will design and implement your application around them. The requirements below are intended to be broad and give you freedom to explore alternative design choices.

If I can see your effort, your understanding and putting the key technologies together, you'll get an A.

Dataset

You will be using two dataset.

- Amazon Kindle's reviews, available from Kaggle website.
  5-core dataset of product reviews from Amazon Kindle Store category from May 1996 - July 2014. Contains total of 982619 entries. Each reviewer has at least 5 reviews and each product has at least 5 reviews in this dataset.

  Columns
  - `asin` - ID of the product, like B000FA64PK
  - `helpful` - helpfulness rating of the review - example: 2/3.
  - `overall` - rating of the product.
  - `reviewText` - text of the review (heading).
  - `reviewTime` - time of the review (raw).
  - `reviewerID` - ID of the reviewer, like A3SPTOKDG7WBLN
  - `reviewerName` - name of the reviewer.
  - `summary` - summary of the review (description).
  - `unixReviewTime` - unix timestamp.

  This dataset has 982,619 entries (about 700MB).

- Amazon Kindle metadata, available from UCSD website

  Sample metadata:


```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "price": 3.17,
  "imUrl": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "related":
  {
    "also_bought": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S", "0000031895",
"B003AVKOP2", "B003AVEU6G", "B003IEDM9Q", "B002R0FA24", "B00D23MC6W",
"B00D2K0PA0", "B00538F5OK", "B00CEV86I6", "B002R0FABA", "B00D10CLVW",
"B003AVNY6I", "B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
    "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E", "B003AVKOP2",
"B00D9C1WBM", "B00CEV8366", "B00CEUX0D8", "B0079ME3KU", "B00CEUWY8K",
"B004FOEEHC", "0000031895", "B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2",
"B00EM7KAG6", "B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ", "B00538F5OK",
```

```
"B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U", "B00CEUWUZC", "B00IJVASUE",
"B00GOR07RE", "B00J2GTM0W", "B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G",
"B008VV8NSQ", "B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M", "B00EHAGZNA",
"B0046W9T8C", "B00E79VW6Q", "B00D10CLVW", "B00B0AVO54", "B00E95LC8Q",
"B00GOR92SO", "B007ZN5Y56", "B00AL2569W", "B00B608000", "B008F0SMUC",
"B00BFXLZ8M"],
    "bought_together": ["B002BZX8Z6"]
  },
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports", "Dance"]]
}
```


where

- `asin` - ID of the product, e.g. 0000031852
- `title` - name of the product
- `price` - price in US dollars (at time of crawl)
- `imUrl` - url of the product image
- `related` - related products (also bought, also viewed, bought together, buy after viewing)
- `salesRank` - sales rank information
- `brand` - brand name
- `categories` - list of categories the product belongs to

This dataset has 434,702 products (about 450MB)

WARNING There are some weird characters in these datasets that cause troubles to the loading tools in MySQL and MongoDB. I have manually removed them, and uploaded the *clean* datasets to Dropbox. You can download them using the provide script: `./scripts/get_data.sh`.

The frontend

This consists of a web page that let an user perform at least the following:

- See some reviews
- Add new review
- Add a new book

You are free to use any Web framework you want, and free to decide your own structure and layout. As long as I can enter input and get my output. Pretty website will earn you more points, to a certain limit.

And feel free to add more functionalities as the project progresses.

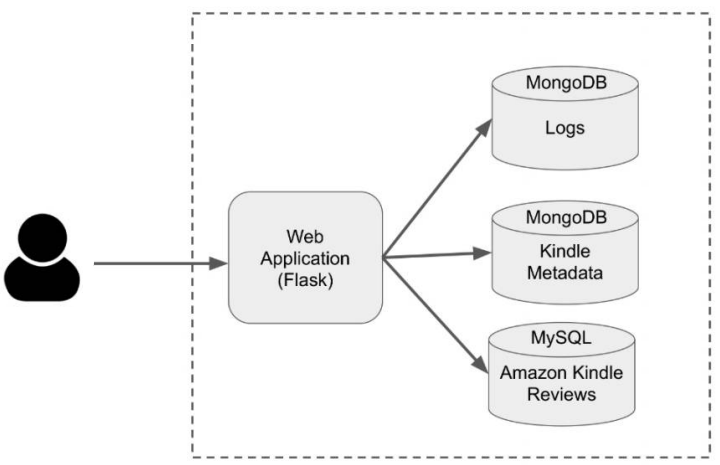
The backend

The backend will be the meat of your application, and it accounts for most of the grade. The are two types of backend: a production backend, and a analytics backend.

- *Production backend*: consists of the web server, and transaction databases that serve requests from the Internet.
- *Analytics backend*: consists of clusters of machines that crunch numbers on the regularly dumped data. In practice, the production system will regularly backup its data and move it to the analytics backend, both for archival storage and for analytics. Another benefit of this practice is that the production system can remove the old data, thus saving resources and making it lightweight.

Production backend requirements

You will build a web server and several databases like below.



- The web server receives requests and computes the responses by interacting with the databases.
- The reviews are stored in a relational databases (SQL is recommended).
- The metadata (book descriptions) is in a document store (MongoDB is recommended).
- The web server logs are recorded in a document store. Each log record must have at least the following information:
  - Timestamp

- Timestamp
- What type of request is being served
- What is the response

CHECKPOINT 1

For CP1, you will implement the production backend on your local machine. That is, the web server and databases run on the same machine.

Automate as much as you can

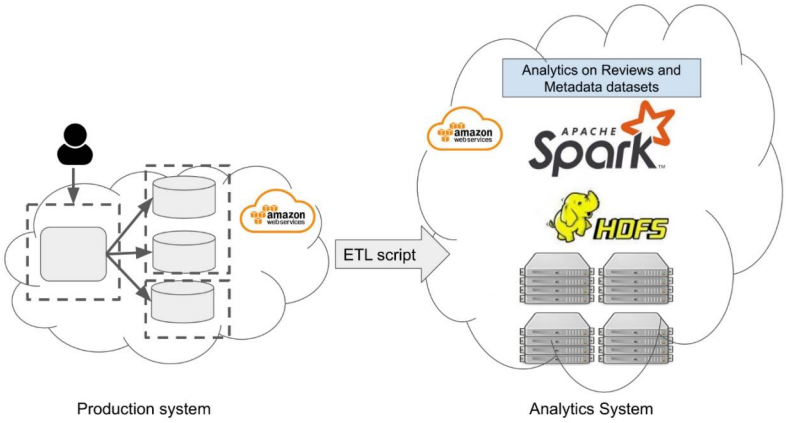
CHECKPOINT 2

For CP2, you will move your backend to Amazon EC2. The server, relational database, and document store, will each run in a separate machine. You will need 3 machines on EC2 for your application to work.

Again, automate as much as you can

Analytics backend requirements

You will build a analytics pipeline and system that looks like the figure below.



[Task 1] You will first write a script that saves data from the production system, and then loads the data to a distributed file system (HDFS) in the analytics systems.

[Task 2] Write the following applications in Spark.

- *Correlation*: compute the Pearson correlation between price and average review length. You are to implement in a map-reduce fashion, and are not allowed to use `mllib.stat.Satistics`.
- *TF-IDF*: compute the term frequency inverse document frequency metric on the review text. Treat one review as a document.

[Task 3] Demonstrate Task 2 running on 2,4,8-node clusters.

CHECKPOINT 3

You will build the analytics backend as specified above. You will integrate with the production backend to make your application complete.

You will not (be able to) do this without automation scripts.