System Design: Amazon's "Customers who bought this item also bought" recommendation system

Functional Requirements: Given a product details, find most frequent products bought by same customers who also bought this product.

Non-Functional: High available, consistency not must, Realtime,

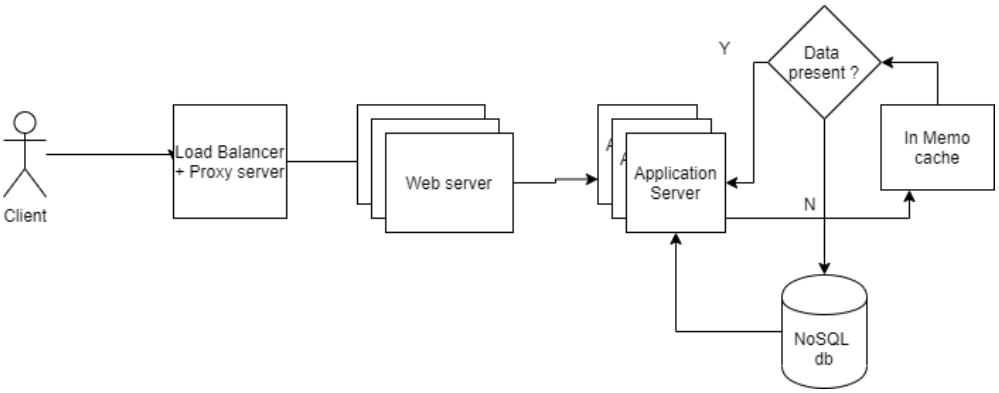
Use case: The feature should be exposed via a service. The application should invoke service when customer lands on a product page. The service should provide details in real-time which will be loaded by the application through async process.

API: REST can be preferred due to lightweight. SOAP is generally preferred for ACID transactional services not applicable here.

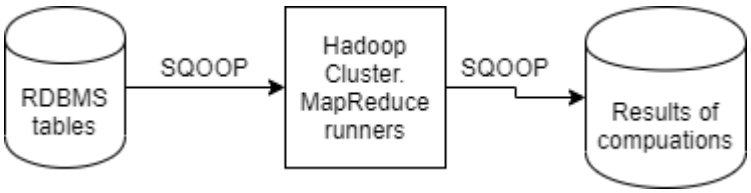
get: /products/recommendations/itemitem?product_id=1234,count=10 -> response: JSON: List of product ids similar to given product.

The system should have 2 parts, Online and Offline. Online service should fetch results from precomputed and stored details and return the results ASAP.

Online service: The service is a read heavy service. The preferred data storage could be a key-value store like NoSQL. Key=product_id,value={pid1:cid1, pid2:cid2…}. A cache can be used to reduce load on DB server. Preferred Cache aside approach. i.e hit cache, if miss then hit DB. Cache is updated periodically with some async process.



Offline process: Data is pulled from assumed RDBMS tables into HDFS, processed and results stored in NoSQL storage used by online service. For pulling data, technology like SQOOP can be used.



The related products are calculated by a pluggable algorithm. The processing task is performed by MapReduce jobs. The results are stored into the NoSQL db used earlier.