Design Questions

Design the recommendations list for amazon webpage

Recommendation Engine can be done by many ways:

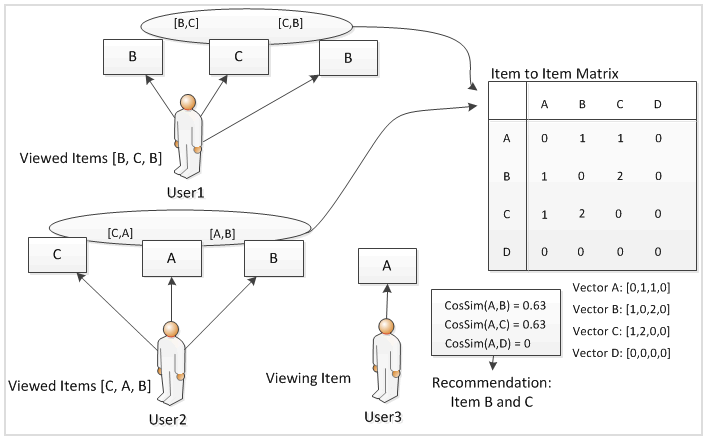
1. Amazon uses Item-Item collaborative filter

Amazon gets my data implicitly by

* The items that I visited
* The items that I purchased
* The items that I rated
* The items in my shopping cart
* The items in my wish list

It takes all the above item and maps it the related items. It does that by looking for items that customers tend to purchase together.

the recommendation system builds a product-to-product matrix by iterating through all item pairs and computing a similarity metric for every single pair.



There are any number of ways to compute a similarity metric. One common way is to use the cosine measure method, where each vector corresponds to an item rather than a customer, and the vectory's M dimensions correspond to customers have bought that item. This is a very time-intensive operation, as you might imagine - the running time is O(N^2 \* M) in the worst case, though it's usually O(N\*M) in practice. This is because most customers don't have many purchases.

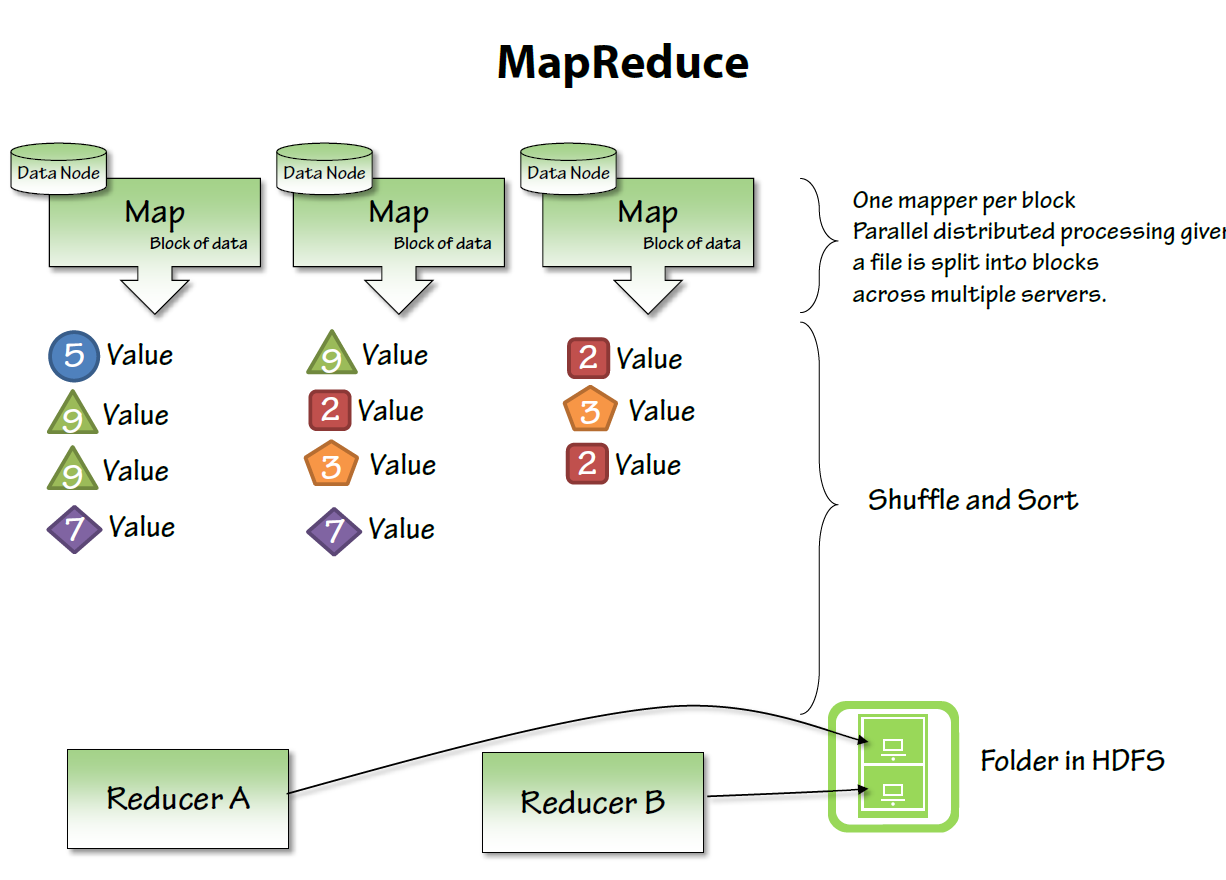
<http://kunuk.wordpress.com/2012/03/04/how-does-the-amazon-recommendation-system-work-analyze-the-algorithm-and-make-a-prototype-that-visualizes-the-algorithm/>

<http://blog.echen.me/2011/02/15/an-overview-of-item-to-item-collaborative-filtering-with-amazons-recommendation-system/>

<http://stackoverflow.com/questions/2323768/how-does-the-amazon-recommendation-feature-work>

<http://scenic.princeton.edu/network20q/wiki/index.php?title=Amazon_and_Youtube%27s_Recommendation_Systems>

Distributed System





Fault-Tolerant: It can recover from component failures without performing incorrect actions.

 Highly Available: It can restore operations, permitting it to resume providing services even when some components have failed.

 Recoverable: Failed components can restart themselves and rejoin the system, after the cause of failure has been repaired.

 Consistent: The system can coordinate actions by multiple components often in the presence of concurrency and failure. This underlies the ability of a distributed system to act like a non-distributed system.

 Scalable: It can operate correctly even as some aspect of the system is scaled to a larger size. For example, we might increase the size of the network on which the system is running. This increases the frequency of network outages and could degrade a "non-scalable" system. Similarly, we might increase the number of users or servers, or overall load on the system. In a scalable system, this should not have a significant effect.

 Predictable Performance: The ability to provide desired responsiveness in a timely manner.

 Secure: The system authenticates access to data and services [1]

