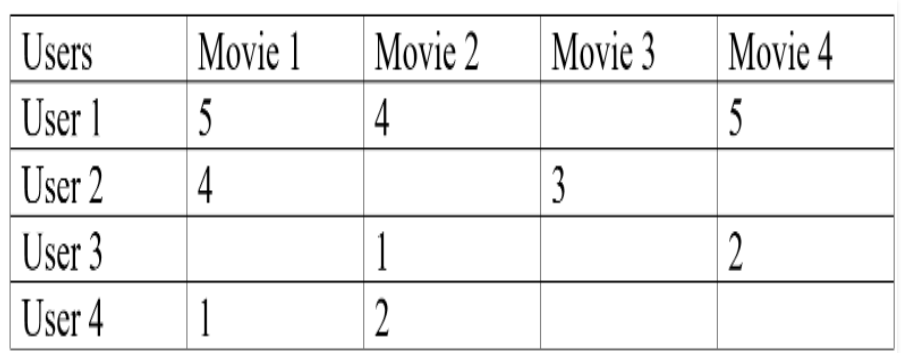
**Collaborative filtering**

* In Collaborative Filtering, we tend to find similar users and recommend what similar users like.
* In this type of recommendation system, we don’t use the features of the item to recommend it, rather we classify the users into the clusters of similar types, and recommend each user according to the preference of its cluster.

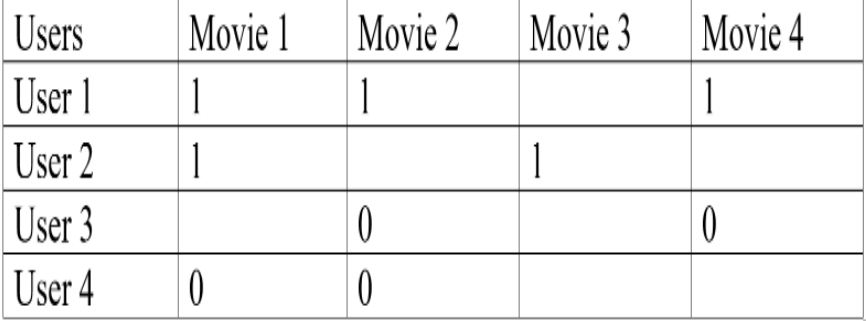
**Measuring Similarity:** A simple example of the movie recommendation system will help us in explaining:



* In this type of scenario, we can see that User 1 and User 2 give nearly similar ratings to the movie, so we can conclude that Movie 3 is also going to be averagely liked by the User 1 but Movie 4 will be a good recommendation to User 2, like this we can also see that there are users who have different choices like User 1 and User 3 are opposite to each other. One can see that User 3 and User 4 have a common interest in the movie, on that basis we can say that Movie 4 is also going to be disliked by the User 4.
* This is Collaborative Filtering, we recommend users the items which are liked by the users of similar interest domain.

**Cosine Distance:**

* We can also use the cosine distance between the users to find out the users with similar interests, larger cosine implies that there is a smaller angle between two users, hence they have similar interests.
* We can apply the cosine distance between two users in the utility matrix, and we can also give the zero value to all the unfilled columns to make calculation easy, if we get smaller cosine then there will be a larger distance between the users and if the cosine is larger than we have a small angle between the users, and we can recommend them similar things.
* **Rounding the Data:** In collaborative filtering we round off the data to compare it more easily like we can assign below 3 ratings as 0 and above of it as 1, this will help us to compare data more easily, for example:

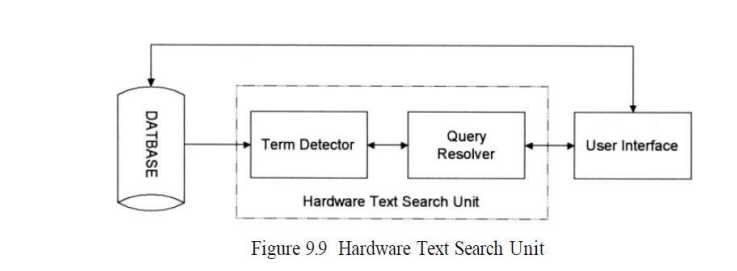


* We again took the previous example and we apply the rounding off process, as you can see how much readable the data has become after performing this process, we can see that User 1 and User 2 are more similar and User 3 and User 4 are more alike.

**Normalizing Rating:** In the process of normalizing we take the average rating of a user and subtract all the given ratings from it, so we’ll get either positive or negative values as a rating, which can simply classify further into similar groups. By normalizing the data we can make the clusters of the users which gives a similar rating to similar items and then we can use these clusters to recommend items to the users.

**Hardware Text Search Systems**

* Software text search is applicable to many circumstances but has encountered restrictions on the ability to handle many search terms simultaneously against the same text and limits due to I/O speeds.
* One approach that off loaded the resource intensive searching from the main processors was to have a specialized hardware machine to perform the searches and pass the results to the main computer which supported the user interface and retrieval of hits.
* The only limit on speed is the time it takes to flow the text off of secondary storage (i.e., disk drives) to the searchers. Since the searcher is hardware based, scalability is achieved by increasing the number of hardware search devices.
* Another major advantage of using a hardware text search unit is in the elimination of the index that represents the document database. Typically the indexes are 70% the size of the actual items.
* Other advantages are that new items can be searched as soon as received by the system rather than waiting for the index to be created and the search speed is deterministic.



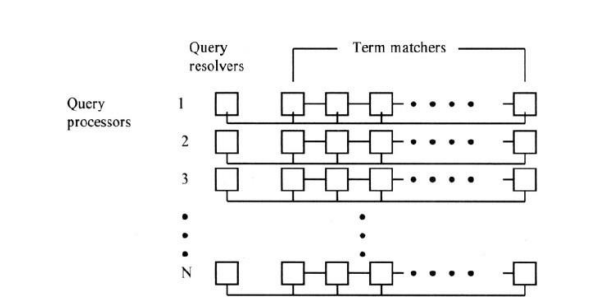
* One of the earliest hardware text string search units was the Rapid Search Machine developed by General Electric.
* The machine consisted of a special purpose search unit where a single query was passed against a magnetic tape containing the documents.

The two methods of hardware systems are:

GESCAN and FDF (Fast Data Finder).

**GESCAN Text Array Processor**

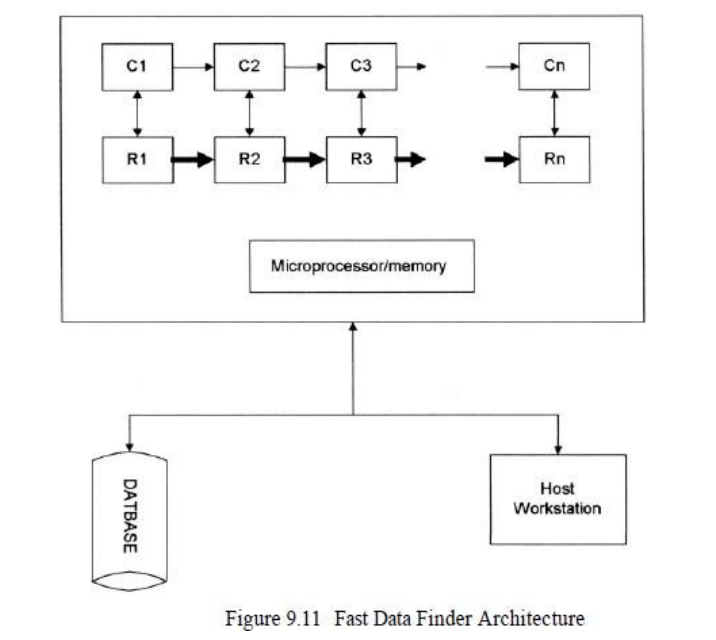
* In some of the systems, the Boolean logic between terms is resolved in the term detector hardware (e.g., in the GESCAN machine).
* The GESCAN system uses a text array processor(TAP) that simultaneously matches many terms and conditions against a given text stream.



* Then TAP receives the query information from the users computer and directly access the textual data from secondary storage.
* The TAP consists of a large cache memory and an array of four to 128 query processors.
* The text is loaded into the cache and searched by the query processors.
* Each query processor is independent and can be loaded at any time.
* A complete query is handled by each query processor.
* Queries support exact term matches, fixed length don't cares, variable length don't cares, terms may be restricted to specified zones, Boolean logic, and proximity.
* A query processor works two operations in parallel; matching query terms to input text and Boolean logic resolution.
* Term matching is performed by a series of character cells each containing one character of the query.
* A string of character cells is implemented on the same LSI chip and the chips can be connected in series for longer strings.
* When a word or phrase of the query is matched, a signal is sent to the resolution sub-process on the LSI chip.
* The resolution chip is responsible for resolving the Boolean logic between terms and proximity requirements.
* If the item satisfies the query, the information is transmitted to the users computer.

**Fast Data Finder:**

* The Fast Data Finder(FDF) is the most recent specialized hardware text search unit still in use in many organizations.
* It was developed to search text and has been used to search English and foreign languages.
* The early Fast Data Finders consisted of an array of programmable text processing cells connected in series forming a pipeline hardware search processor.
* The cells are implemented using a VSLI chip.
* Each cell is a comparator for a single character, limiting the total number of characters in a query to the number of cells.
* The cells are interconnected with an 8-bit data path and approximately 20-bit control path.
* The text to be searched passes through each cell in a pipeline fashion until the complete database has been searched.



* The input from the Document database is controlled and buffered by the microprocessor/memory and feed through the comparators.
* The search characters are stored in the registers. The connection between the registers reflects the control lines that are also passing state information.