



An Optimal Cloud Service Selection Based on the QoS Values: A Survey

Sobhika¹, Rajagopal²

M.E, Department of CSE, Vivekandha Institute of Engineering and Technology for Women, Trichengode, Tamilnadu, India

Asst Prof, Department of CSE, Vivekandha Institute of Engineering and Technology for Women, Trichengode, India²

ABSTRACT—Cloud computing is becoming popular. Building high-quality cloud applications is a critical research problem. QoS rankings provide valuable information for making optimal cloud service selection from a set of functionally equivalent service candidates. To obtain QoS values, real-world invocations on the service candidates are usually required. To avoid the time consuming and expensive real-world service invocations, this paper surveys different QoS ranking prediction methods like neighborhood based approach, ADF, Local optimization for ranking of cloud services by taking advantage of the past service usage experiences of other consumers. Finally this paper summarizes various approaches for ranking of services in the cloud that is used for optimal service selection.

KEY WORDS—Quality-of-service, cloud service, ranking prediction, personalization

I. INTRODUCTION

Cloud computing is Internet-based computing, whereby shared configurable resources (e.g., infrastructure, platform, and software) are provided to computers and other devices as services. Strongly promoted by the leading industrial companies (e.g., Amazon, Google, Microsoft, IBM, etc.), cloud computing is quickly becoming popular in recent years. Applications deployed in the cloud environment are typically large scale and complex.

Nonfunctional performance of cloud services is usually described by quality-of-service (QoS). QoS is an important research topic in cloud computing. When making optimal cloud service selection from a set of functionally equivalent services, QoS values of cloud services provide valuable information to assist decision making. In traditional component-based systems, software components are invoked locally, while in cloud applications, cloud services are invoked remotely by Internet connections. Client-side performance of cloud services is thus greatly influenced by the unpredictable Internet connections.

Therefore, different cloud applications may receive different levels of quality for the same cloud service. Quality-of-service can be measured at the server side or at the client side. While server-side QoS properties provide good indications of the cloud service capacities, client-side QoS Properties provide more realistic measurements of the user usage experience. The commonly used client-side QoS properties include response time, throughput, failure probability, etc.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

II. LITERATURE REVIEW

A. TRADITIONAL COLLABORATIVE FILTERING

Traditional collaborative filtering algorithm represents a customer as an N-dimensional vector of items, where N is a number of distinct catalog items. The components of the vector are positive for purchased or positively rated items and negative for negatively rated items and the algorithm generates recommendations based on a few customers who are most similar to the user. To generate recommendations using collaborative filtering is expensive and it is $O(MN)$ in the worst case, where M is the number of customers and N is the number of product catalog items.

B. ITEM-ITEM COLLABORATIVE FILTERING

Item to Item collaborative filtering matches each of the user's purchased and rated items to similar items then combines those similar items into a recommendation list. Then recommendation algorithms start by finding a set of customers who purchased and rated items overlap the user's purchased and rated items then the algorithm aggregates items from these similar customers eliminates items the user has already purchased or rated and recommends the remaining items to the user. The key to item to item collaborative filtering scalability and performance is that it creates the expensive similar items table offline. The algorithm's online component looking up similar items for the user's purchases and ratings scales independently of the catalog size or the total number of customers, it is dependent only on how many titles the user has purchased or rated. Thus the algorithm is fast even for extremely large data sets because the algorithm recommends highly correlated similar items recommendation quality are excellent.

C. MODEL BASED RECOMMENDATION TECHNIQUES

Model based recommendation techniques analyze the user item matrix to discover relations between the different items and use these relations to compute a list of recommendation. Algorithm used is model based recommendation algorithm that first determines the similarities between the various items and then uses them to identify the set of items to be recommended. The key step of algorithm includes the method to compute the similarity between the items and the method to combine these similarities in order to compute the similarity between a basket of items and a candidate recommender item. Algorithm used is model based recommendation algorithm that first determines the similarities between the various items and then uses them to identify the set of items.

Two Model Based Algorithms used, the first algorithm follows a probabilistic approach in which the users are clustered and the conditional probability distribution of different items in the cluster is estimated. The probability distribution that the active user belongs to a particular cluster, given the basket of items is then estimated from the clustering solution and the probability distribution of items in the cluster. The second algorithm is based on Bayesian network models where each item in the database is modeled as a node having status corresponding to the rating of that item.

D. ADF

Neighborhood based collaborative filtering approach called ADF is used to predict the unknown QOS values. ADF denotes adjusted cosine, data smoothing and fusion of similarity values. A cosine equation which takes difference in QOS scale between different users into account to calculate the service similarity instead of using the PCC approach. Unknown QOS of similar neighbors is replaced by zero which lowers the accuracy of prediction. In ADF a data smoothing



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

process is used and helps in finding unknown QOS values and is replaced by the average QOS of their cluster. Fusion approach is used to reduce data sparsity problem.

The CF approach is widely adopted in recommender system and is often classified as memory based or model based. In the former all the training data is stored in memory whereas in the prediction phase similar objects are stored on their similarities with the active object. The cosine and PCC are two widely used methods to compute similarities between objects and based on data from similar user to items a prediction result can be generated. The most analyzed examples include user based models and item based models. The advantage of memory based methods is that they are simple and intuitive on a conceptual level and avoid the complications of a potentially expensive model building stage.

E. EIGENRANK APPROACH TO COLLABORATIVE FILTERING

Eigenrank approach in recommender system must be able to suggest items that are likely to be preferred by the user and in the most systems degree of preference is represented by a rating score. Given a database of users past ratings on a set of items traditional collaborative filtering algorithms are based on predicting the potential ratings that a user would assign to unrated items so that they can be ranked by the predicted ratings to produce a list of recommended items. Recommender systems fall into two categories namely content based filtering and collaborative filtering. Content based filtering approach analyzes the content information associated with the items and users such as product descriptions, user profiles in order to represent users and items using a set of features. To recommend new items to user content based filters match their representation to those items known to be of interest to the user.

Collaborative filtering approach does not require any content information about the items it works by collecting ratings on the items by a large number of users and make recommendations to a user based on the preference patterns of other users. The CF is based on the assumption that a user would usually be interested in those items preferred by other users with similar interests and it requires no domain knowledge and can be easily adopted in different recommender systems. It is usually adopted in two classes of application scenarios, in the first class a user is presented with one individual item at a time along with a predicted rating indicating the user's potential interest in the item and in the second class of applications produce an ordered list of Top N recommended items where the highest ranked items are predicted to be most preferred by the user. The computation of the Top N list for making recommendations is essentially a ranking problem and for this collaborative filtering algorithms adopt a rating oriented approach which first predicts the potential ratings a target user would assign to the items and then rate the items according to the predicted ratings.

F. NEIGHBORHOOD-BASED APPROACHES

The most common form of neighborhood-based approach is the user-based model which estimates the unknown ratings of a target users that tend to rate similarly to the target user. A crucial component of the user-based model is the user-user similarity that is to select the set of neighbors. One difficulty in measuring the user-user similarity is that the raw ratings may contain biases caused by the different rating behaviors of different users. Second difficulty in user-based models arises from the fact that the known user-item ratings data is typically highly sparse which makes it very hard to find highly similar neighbors for making accurate predictions. An alternative form of neighborhood-based approach is the item-based model and it is used to select a set of neighboring items that have been rated by the target user and the ratings to the unrated items.

G. MODEL-BASED APPROACHES



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The model based approach to collaborative filtering use the observed user-item ratings to train a compact model that explains the given data so that ratings could be predicted via the model instead of directly manipulating the original rating database as the neighborhood-based approach does.

Methods included in model based approaches are,

- i) Clustering Method
- ii) Aspect model
- iii) Bayesian networks.

H. RANKING IN RECOMMENDATION SYSTEMS

The notion of user-and query-similarity, it appears that the techniques of collaborative and content filtering used in recommendation systems. However, there are some important differences. For instance, each cell in the user-item matrix of recommendation systems represents a single scalar value that indicates the rating/preference of a particular user towards a specific item. Similarly, in the context of recommendations for social tagging, each cell in the corresponding user-URL/item-tag matrix indicates the presence or absence of a tag provided by a user for a given URL/item.

To the best of knowledge, a model for establishing similarity between database queries (expressed in SQL) has not received attention. In addition, a user profile is unlikely to reveal the kind of queries a user might be interested in. Further, since we assume that the same user may have different preferences for different queries, capturing this information via profiles will not be a suitable alternative. In collaborative filtering, users are compared based on the ratings given to individual items (i.e., if two users have given a positive/negative rating for the same items, then the two users are similar). In the context of database ranking a rigorous definition of user similarity based on the similarity between their respective ranking functions, and hence ranked orders is made. Furthermore, it also extends user-personalization using context information based on user and query similarity instead of static profiles and data analysis.

I. CUMULATED GAIN BASED EVALUATION OF IR TECHNIQUES

Modern large retrieval environments tend to overwhelm their users by using large output. Since all documents are not of equal relevance to their users, highly relevant documents should be identified and ranked first for presentation. In order to develop IR techniques, it is necessary to develop evaluation approaches and methods that credit IR methods for their ability to retrieve highly relevant documents. This can be done by extending traditional evaluation methods alternatively novel measures based on the graded relevance judgment may be developed.

J. AN OPEN ARCHITECTURE FOR CF OF NETNEWS

Collaborative filters help people make choices based on the opinions of other people. Group lens is a system for collaborative filtering of Netnews to help people find articles they will like in the huge stream of available articles. News reader client displays predicted scores and make it easy for users to rate articles after they read them. Rating servers called Better Bit Bureaus gather and disseminate the ratings. The rating servers predict scores based on the heuristic that the people who agreed in the past will probably agree again. Users can protect their privacy by entering ratings under pseudonyms without reducing the effectiveness of the score prediction.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Group lens is a distributed system for gathering disseminating and using ratings from some users to predict other user's interest in articles. It includes news reading clients for both Macintosh and UNIX computers as well as better bit Bureaus servers that gather ratings and make predictions. The heart of group lens is an open architecture that includes news clients for entry of ratings and display of predictions and rating servers for distribution of ratings and delivery of predictions. Group lens uses the ratings in two ways; first it correlates the ratings in the order to determine which user's ratings are most similar to each other. Second it predicts how well user will like new articles based on the ratings from similar users.

K.LOCAL OPTIMIZATION APPROACH

The selection of the web service will execute a given task of a composite service specification is done at the last possible moment and without taking into account other tasks involved in the composite service. When the system actually needs to be executed the system collects information about the QOS of each of the web services that can execute this task. After collecting the QOS information a quality vector is computed for each of the candidate web services and based on this quality vectors the system selects one of the candidate web services by applying Multiple Criterion Decision Making (MCDM) technique. This selection process is based on the weight assigned by the user to each criterion and a set of user-defined constraints expressed using a simple expression language. A Simple Additive Weighting (SAW) technique is used to select an optimal web service.

There are two phases in SAW namely,

- i) Scaling Phase - In this phase higher the value lowers the quality. This includes criteria such as execution time and execution price. Other criteria are positive that is higher the value higher the quality.
- ii) Weighting Phase - In this phase the system will choose the web service which satisfies all the user constraints for that task and which has the maximal score. If there are several services with maximal score, one of them is selected randomly. If no service satisfies the user constraint for a given task an execution exception will be raised and the system will propose the user to relax these constraints.

L. RANKING PREDICTION FRAMEWORK

A personalized ranking prediction framework, named Cloud Rank, to predict the QoS ranking of a set of cloud services without requiring additional real-world service invocations from the intended users. The approach takes advantage of the past usage experiences of other users for making personalized ranking prediction for the current user. The approach identifies the critical problem of personalized QoS ranking for cloud services and proposes a QoS ranking prediction framework to address the problem. To the best of knowledge Cloud Rank is the first personalized prediction framework for cloud services. Extensive real-world experiments are conducted to study the ranking prediction accuracy of the ranking prediction algorithms compared with other competing ranking algorithms. The experimental results show the effectiveness of the approach. QoS dataset is publicly released for future research which makes the experiments reproducible.

Cloud Rank framework has several modules, first based on the user provided QoS values similarities between the active user and training users can be calculated. Second, based on the similarity values, a set of similar users can be identified. After that two algorithms namely Cloud Rank1 and Cloud Rank 2 to make personalized service ranking by taking advantages of the past service usage experiences of similar users.. Finally the ranking prediction results are provided to the active user. The training data in the Cloud Rank framework can be obtained from the QoS values provided by other users and the QoS values collected by monitoring cloud services.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

III. CONCLUSION

In this paper, we have surveyed different approaches and different architectures for service selection which will lead to ranking of services. The major aim of these approaches is to find the rank of individual services so that it is provided to the users as an optimal one. The architectures and algorithms used helps in providing best services to users that is mainly based on quality of service values.

REFERENCES

1. K. Arvelinand , J. Kekalainen , "Cumulated Gain-Based Evaluation of IR Techniques," ACM Trans. Information Systems, vol. 20, no.4,pp. 422-446, 2002.
2. P. Bonatti and P. Festa , "On Optimal Service Selection," Proc.14th Int'l Conf. World Wide Web (WWW '05), pp. 530-538, 2005.
3. R. Burke and P. Festa, "Hybrid Recommender Systems: Survey and Experi-ments, "User Modeling and User Adapted Interaction, vol.12, no. 4,pp. 331-370, 2002
4. M. Deshpande and G. Karypis , "Item-Based Top-n Recommenda-tion," ACM Trans. Information System, vol. 22, no. 1, pp. 143-177,2004.'04), pp. 337344,2004.
5. G. Linden, "Amazon.Com Recommendations: Item to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, an./Feb. 2003.
6. N. Liu and Q. Yang , "Eigenrank: A Ranking-Oriented Approach to Collaborative Filtering," Proc. 31st Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '08),pp. 83-90, 2008.
7. P. Resnick, N. Iacovou, M. Suchak and M. Riedl, "Grouplens: An Open Architecture for Collaborative Filtering of Netnews," Proc. ACM Conf. Computer Supported Cooperative Work, pp. 175-186, 1994.
8. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl , "Item-Based Collaborative Filtering Recommendation Algorithms," Proc.10thInt'l Conf. World Wide Web (WWW '01), pp. 285-295, 2001.
9. J. Wu , L. Chen, Y. Feng, Z. Zheng, M. Zhou, "Predicting QoS for Service Selection by Neighborhood-Based Collaborative Filtering," IEEE Trans. System, Man, and Cybernetics, Part A, to appear.
10. L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam , H. Chang , "QoS-Aware Middleware for Web Services Composi-tion," IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May2004.