Cloud-based Recommendation Systems: Applications and Solutions

# Cloud-based Recommendation Systems: Applications and Solutions

Erion Çano
Polytechnic University of Turin
erion.cano@polito.it

**Abstract:** Recommender systems have become extremely common in recent years, and are applied in a variety of applications. They help businesses increase their sales and customer satisfaction. More and more computing applications including recommender systems, are being deployed as cloud computing services. This papers presents some of the most common recommendation applications and solutions which follow SaaS, PaaS or other cloud computing service models. They are provided both from academia and business domain and use recent data mining, machine learning and artificial intelligence techniques. The tendency of these kind of applications is towards SaaS service model which seems the most appropriate especially for businesses.

**Keywords:** Cloud Computing, Recommender Systems, Cloud-based Recommenders

## I. Introduction

Even though it is not easy to have a unique and clear definition of cloud computing technologies [1], they are commonly considered as a set of easily accessible and virtualized resources that can be dynamically and easily adapted allowing an optimum resource utilization. This pool of resources is made up of a lot of server computers which enable users to get computing power, storage capacities and networking bandwidth according to the needs in a flexible and customizable way. Thus it enables the service providers to offer diversity of access, integration of multiple services, flexibility and scalability in resource provisioning and pricing etc, greatly reducing costs of infrastructure and maintenance. Recommender systems are applications which try to predict user behavior using machine learning, data mining or artificial intelligence techniques. They are mostly used to recommend consumable items or services such as movies, music, books, devices, restaurants, touristic sites, news, blogs etc and increase the income of the vendors. To scale efficiently the recommendation system requires large-scale computational and storage resources. They are among the many recent application categories which are being conceived and offered using cloud computing infrastructure. The migration to the cloud allows the improvement of the scalability of the recommender systems, given the high level of computational power required by such systems.

These software systems are now helping the world’s leading brands increase revenue, bounce rate and retention, enhance user discovery etc by creating a personalized experience for their customers and shaping the future of e-commerce. The goal of this paper is to present the basic characteristics of different cloud-based recommendation applications and solutions proposed both by academia and industry. Most of the applications are offered by companies whom clients are other companies operating in the domains of e-commerce, finance or market research. Their products follow the SaaS model and are usually closed. There are also open source business-oriented solutions developed by companies or volunteers. They are usually hybrid or PaaS applications that enable web recommendations, prediction of user preferences etc. The contributions from academia tent to follow hybrid architectures and are more focused in recommending websites, news, blogs etc. Recommendations and analytical results are provided using various up to date data mining, machine learning and artificial intelligence techniques.

## II. Academic Solutions

With the rapid development of the Internet, the number of blogs grows sharply, it becomes more difficult for users to find the valuable and personalized blog feeds they want. Therefore it is meaningful to widen the ways of obtaining blogs and improve the quality of blog recommendations. In [2] the authors present a blog recommender system based on cloud computing infrastructure. The authors integrate the user-based collaborative filtering into traditional blog search system to realize the personalized customization of blog searching. Their recommender uses Hadoop distributed file system to store mass blog data and implement the distributed processing of blog crawl and index creation stages. The architecture mainly includes three layers: blog collection and create index, blog search and clustering view of the search results, user preference collection and personalized blog recommendation.

Technology Enhanced Learning (TEL) is a relatively new application domain of recommender systems. Recommender systems are among the instruments used to provide learning environments with personalization. In [3] the authors present a review of the approaches followed so far to implement and deploy recommender systems in TEL. A cloud-based architecture for a system that recommends learning resources is presented at [4]. The purpose of the proposed architecture is to deliver a set of learning resources meta-data following a Software as a Service approach. The architecture is composed of two layers: a service layer that executes the storage and recommendation tasks, and a client layer embedded in a learning environment. Their recommendation process follows the method known as user-based collaborative filtering. The environment where they have implemented the recommender service is Amazon Elastic Computing Cloud (EC2), which is part of Amazon Web Services (AWS).

Good website recommendations to users is a need of the time. In [5] the authors present an efficient cloud-based framework for analyzing surfing behavior of internet users and providing website recommendations. A website recommendation system is a web based interactive software agent which predicts the user's web browsing behavior. This is accomplished by analyzing the user's browsing habits. The system is based on web usage mining concepts, which analyses the user's web usage statistics and performs data mining tasks like clustering. The proposed framework infrastructure is based on cloud implementation which is freely available on the internet in the forms of Google App engine, Big table and Google File System for easy deployment.

In [6] the authors propose **OmniSuggest**, a cloud-based recommendation framework that utilizes ant colony algorithms, social filtering and hub and authority scores to address data sparseness, cold start and scalability. This solution leverages cloud infrastructure and service-based interfaces to process, mine, compare and manage large-scale datasets for real-time recommendations in a scalable architecture. OmniSuggest combines memory-based and model-based approaches of collaborative filtering on a cloud framework. The framework follows a SaaS approach through a modular service based architecture. The SaaS forms the top layer of the cloud stack offering real-time personalized recommendations to a user or groups of users while abstracting underlying implementation details.

**LensKit** is a Java-based recommender toolkit from GroupLens [7]. It provides a common API for recommender algorithms, an evaluation framework for offline evaluation of recommender performance, and highly modular implementations of common collaborative filtering algorithms. The aim of the project is to produce a common recommender framework, reusable in applications, and clear, readable implementations of common algorithms employing best practices with regards to implementation strategies, tuning and normalizations. The project currently has several rating-based algorithms and is adding basket/history-based algorithms and additional evaluation strategies. LensKit is intended to be particularly useful in recommender systems research.

## III. SaaS Recommender Applications

SaaS analytics and recommendation applications are becoming more and more common as they offer payment for value with a low overhead rather than having a large upfront investment. They generally have a clear integration path and they provide continual development and improvement.

**Peerius** is a closed source e-commerce recommender system adapted for product suggestions, messages, promotions, images and campaigns that are personally relevant [8]. It provides personalization for any aspect of the customers' website by using individual user behavior, social inputs, product attributes, behavioral clusters etc. There is little information about the cloud infrastructure and recommendation technology it uses.

**Strands** is also a closed source product and e-commerce recommender system [9] focused in finance and retail services. The recommendation engine leverages big data insights such as past customer behavior, historical purchase history and sales performance and like-minded shopper intent to derive the most meaningful results and promote the right product to the right customer at the right time. The is no information about the infrastructure it uses.

**SLI System Recommender** is another closed recommendation solution which offers SaaS-based learning search, navigation and mobile product recommendations [10].

**Google Cloud Prediction API** offers Google's machine learning algorithms to analyze and predict future outcomes [11]. The prediction API is available through libraries for many popular languages, such as Python, JavaScript and .NET. It also offers cloud integration and flexible pricing for the cloud capacities.

**ParallelDots** is a tool that offers semantic proximity, entity extraction, taxonomy analysis and sentiment analysis on published content [12]. Its artificial intelligence techniques are based mostly on neural networks. The main applications are market research, news recommendations and social media analysis.

**Amazon Machine Learning** is a machine learning platform of Amazon that models data and creates predictions [13]. It makes it easy to obtain predictions for application using simple APIs, without having to implement custom prediction generation code, or manage any infrastructure. Amazon Machine Learning is based on the same proven highly scalable ML technology used for years by Amazon’s internal data scientists community. It is highly scalable and can generate billions of predictions daily, and serve those predictions in real-time and at high throughput.

**Azure Machine Learning** is based on Microsoft Azure Cloud platform and offers a streamlined experience for all data scientist skill levels, from setting up with only a web browser to using drag-and-drop gestures and simple data-flow graphs to set up experiments [14]. Azure Machine Learning Studio features a library of time-saving sample experiments supporting R and Python packages. The model deployment is fast and easy to update.

**Gravity R&D** is a company built by some of the winners of the 2009 Netflix prize [15]. They offer a solution that provides targeted, customized recommendations to users of websites. Gravity R&D’s recommendation system automatically learns and analyses the browsing and/or shopping behavior of the user on a website or platform. The system collects data from other users and items and compares this data to create a list of recommendations that are personally relevant to the visitor’s taste. Gravity’s recommendation system uses collaborative filtering and matrix factorization techniques, constantly enhancing the quality of recommendations.

## IV. Non SaaS Recommendation Solutions

These recommendation solutions are mostly open source and use more heterogeneous cloud architectures.

**PredictionIO** is an open-source Machine Learning server for developers and data scientists to build and deploy predictive applications in a fraction of the time [16]. The core part of PredictionIO is an engine deployment platform built on top of Apache Spark. In addition, there is an Event Server. It is a scalable data collection and analytics layer built on top of Apache HBase. It enables developers to build predictive engine components with separation-of-concerns. The source code is hosted in github and is regularly updated.

**HapiGER** is an open source collaborative filtering engine which can use "in-memory" event store (default) but can be also configured to use PostgreSQL or rethinkdb [17]. It is based on Hapi.js framework and designed to be easy to use and very scalable.

**Apache Mahout** is a project of the Apache Software Foundation to build an environment for quickly creating free implementations of distributed and scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification [18]. The three major components of Mahout are an environment for building scalable algorithms, many new Scala and Spark algorithms, and Hadoop MapReduce algorithms. It comprises robust matrix decomposition algorithms as well as a Naive Bayes classifier and collaborative filtering.

**Seldon** is a full-stack open machine intelligence solution that features a recommendation engine [19]. The prediction engine is based in Java and built on technologies like Apache Spark. Seldon predicts the future actions of consumers of media and e-commerce services across web, mobile and tablet. Easy to integrate REST and JavaScript APIs and self-optimising algorithms ensure a smarter personalised user service.

**Oryx** open source project provides simple, real-time large-scale machine learning analytics infrastructure [20]. It can continuously build models from a stream of data at large scale using Apache Hadoop. The two-tier design is comprised of the Computation Layer and Serving Layer which respectively build models and server models. The Computation Layer is a long-running Java-based server process which just builds models using MapReduce, Hadoop's environment for computation. The Serving Layer is a long-running Java-based server process, which exposes a REST API. It can be accessed from a browser, or any language or tool that can make HTTP requests.

**Dato** is a company that provides a python package and servers for business machine learning including many predictive algorithms for recommendations [21]. Their technology can easily integrate with Apache Spark, Hadoop and Yarn. The distributed infrastructure which is very fast and scalable can combine with any client python code including scikit-learn model.

## V. Discussion

Because of the cost savings, flexibility and outsourcing abilities it offers, cloud computing is today an imminent computing paradigm in the business domain and computation research communities. Most of daily user applications like email, office, file sharing, multimedia etc have been already moved in the cloud. Cloud computing model is also being adopted by many providers to offer business services like website hosting, storage capacities, traffic analysis, data analytics, recommendations etc. Recommender systems are software agents that elicit the interests and preferences of individual consumers and make recommendations accordingly. They are especially beneficial in the domain of e-commerce by offering sales assistance, increasing profits and credibility etc. This paper presents various up to date recommendation applications running on different cloud-based architecture infrastructures. Several frameworks or system architectures are proposed by academic researchers. They are usually focused in recommending websites, news, blogs and learning resources. There are also open source applications or frameworks which usually follow the PaaS service model. Some of them are developed and maintained by research groups and volunteers and others by companies. Most of the presented applications and solutions come from the industry and follow the SaaS service model. They are usually closed source and focused in the domain of e-commerce, finance and market research. The SaaS model represents a common tendency in offering analytics and recommendation applications and solutions.

## References


[1] L. M. Vaquero, L. Rodero-merino, J. Caceres, M. Lindner: “A Break in the Clouds: Towards a Cloud Definition”, ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2009.

[2] Jie Jiang, Weiwei Pang, Yule Deng, Kate He, Zhuyan Gu: "A blog personality recommender system based on cloud computing infrastructure", 2012 International Joint Conference on Service Sciences.

[3] N. Manouselis, H. Drachsler, K. Verbert, and E. Duval, “Recommender systems for learning,” SpringerBriefs in Computer Science, 2012.

[4] D. Loney, A. Pardo, Hugo A. Parada, G. Carlos: "A cloud-based architecture for an affective recommender system of learning resources", 1st International Workshop on Cloud Education Environments.

[5] Sowmya Menon, Varghese Paul, Sudheep Elayidom, Ratan Kuar: "An efficient cloud based framework for Web recommendation systems", International Journal of Advances in Computer Science and Technology, Vol. 3, No. 2, pp 07-11.

[6] Osman Khalid, uhammad Usman Shahid Khan, Samee U. Khan, Albert Y. Zomaya: "OmniSuggest: A Ubiquitous Cloud-Based Context-Aware Recommendation System for Mobile Social Networks", IEEE Transactions on service computing, vol 7, No. 3.

[7] http://lenskit.org/

[8] http://www.peerius.com/

[9] http://strands.com/

[10] http://www.sli-systems.com/

[11] https://cloud.google.com/prediction

[12] http://www.paralleldots.com/

[13] http://aws.amazon.com/machine-learning/

[14] http://azure.microsoft.com/en-us/services/machine-learning/

[15] http://www.gravityrd.com/

[16] https://prediction.io/

[17] http://www.hapiger.com/

[18] http://mahout.apache.org/

[19] http://www.seldon.io/

[20] https://github.com/cloudera/oryx

[21] https://dato.com/