Televido
My Personalized TV

Ľuboš DEMOVIČ, Eduard FRITSCHER, Jakub KŘÍŽ
Ondrej KUZMÍK, Ondrej PROKSA, Diana VANDLÍKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
`televido@googlegroups.com`

Extended Abstract

In today's world there is a large amout of media content produced every day: movies, TV shows, TV programs etc. With so many options, chosing the right content can be difficult and overwhelming for the user. Take television. There are tens of channels from which the user can chose at any given moment. If he choses the wrong channel, he might miss something he would enjoy on a different one. The problem becomes even larger with more and more popular television or video on-demand. The user has literally milions of possibilities from which to choose. The task becomes very difficult without a recommendation system.

Recommendation systems are systems which can analyse the taste, the mood or the context in which the user is at the moment. Based on the analysis, they create an accurate recommendation that suits the particular user. There are various techniques used to create recommendations. The two main categories of recommendation systems are content based and collaborative. In both categories the system has to go through the entire entity base to find the correct content to be recommended. There is no guarantee that the estimation is correct and the recommended content is accurate enough for the user. Many recommendation systems try to recommend content by pairing the extracted knowledge base with the user's context and taste. Using relational databases these tasks can be difficult and not very efficient. As a result, the recommendation systems may suffer from performance issues and, to be able to use them in real time, the recommendations need to be cached or the entity base simplified.

We aim to develop a system that uses a new approach for recommending multimedia content. Our goal is to develop an application which uses the most current, cutting edge tchnology, is fast, accurate and has a great user experience. Instead of the approaches mentioned above we decided to design a service which uses a hybrid model of relational and graph databases for recommending multimedia content, in particular movies, TV shows and TV programs.

The design of the architecture we use is shown in Figure 1. As mentioned, we use two types of databases – relational and graph database. Relational databases are widely used for storing basic

[*] Master study programme in field: Software Engineering
Supervisor: Dušan Zeleník, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2013, Bratislava, April 23, 2013, pp. 1–8.

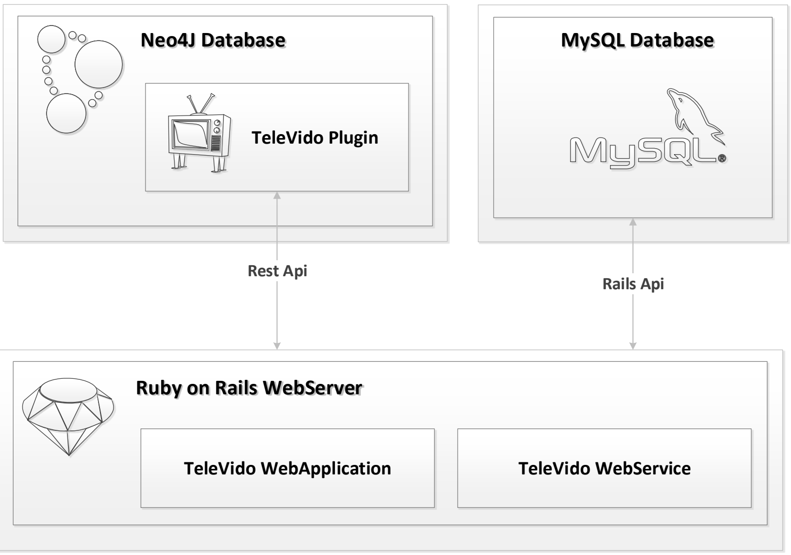

Figure 1. Televido architecture.

data. We use it to store basic information about entities, such as title, description or creation date. This information is easily and quickly fetched for the user interface and can be used to perform faceted searches.

Relational databases are inefficient for storing graph structures, which is the reason why we make use of a graph database. The graph database contains the domain model used for recommendations. The model is based on relations between entities – the main entities are movies, shows and programs. These are connected to each other via relationships, which are based on other entities like actors, producers, directors and genres.

The recommendation algorithms are created as plug-ins to the graph database. They are graph algorithms, which work from initial nodes. The initial nodes are currently selected explicitly, but we plan on selecting them implicitly based on the user model in the future. The algorithms try to find nodes which are closest in the graph to all initial nodes, which are then returned as recommendations. There are multiple ways of looking at the problem of finding the closest nodes, especially in a very complex graph, which is why we designed and implemented four separate algorithms. These algoritms were designed to take in multiple parameters, such as subset of nodes which can be returned as recommendations or subset of relationships which can be used to crawl the graph. Thanks to these parameters, the recommendation algorithms are quite agile. For example, the same algorithm with different parameters can be used to recommend movies which are currently being shown at the cinemas or only the TV programs which will be on tommorow night.

Our recommendation algorithms are currently being experimentaly evaluated. Based on the results of the experiments we will be able to determine the accuracy of each algorithm, select the best algorithm and further tweak it to perfection.

The main advantage of using a graph database is the performance. With enhanced graph algorithms we do not need caching of the recommended content because the service is fast enough to work in real time.