Future Generation Computer Systems 27 (2011) 823–835

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

CPRS: A cloud-based program recommendation system for digital TV platforms

Chin-Feng Lai [a], Jui-Hung Chang [a], Chia-Cheng Hu [b], Yueh-Min Huang [a,*], Han-Chieh Chao [c]

[a] Department of Engineering Science, National Chung Kung University Tainan, Taiwan
[b] Department of Information Management, Naval Academy, Kaohsiung, Taiwan
[c] College of Electrical Engineering and Computer Science, National Ilan University, Ilan, Taiwan

ARTICLE INFO

Article history:
Received 14 April 2010
Received in revised form
20 September 2010
Accepted 12 October 2010
Available online 28 October 2010

Keywords:
Map-reduce
Electronic program guide (EPG)
k-nearest neighbor (kNN)
k-means

ABSTRACT

Traditional electronic program guides (EPGs) cannot be used to find popular TV programs. A personalized digital video broadcasting-terrestrial (DVB-T) digital TV program recommendation system is ideal for providing TV program suggestions based on statistics results obtained from analyzing large-scale data. The frequency and duration of the programs that users have watched are collected and weighted by data mining techniques. A large dataset produces results that best represent a viewer's preferences of TV programs in a specific area. To process such a massive amount of viewer preference data, the bottleneck of scalability and computing power must be removed. In this paper, an architecture for a TV program recommendation system based on cloud computing and a map-reduce framework, the map-reduce version of k-means and the k-nearest neighbor (kNN) algorithm, is introduced and applied. The proposed architecture provides a scalable and powerful backend to support the demand of large-scale data processing for a program recommendation system.

1. Introduction

An electronic program guide (EPG), which consists of a digital set-top box (STB) [1], is a digital guide for scheduling broadcast television or radio programs with functions that allow a user to navigate, select, and discover content by criteria such as time, title, channel, and genre [2]. Due to the increase in the number of TV programs, several recommendation systems have been introduced [3,4]. These systems attempt to recommend programs that are likely of interest to the user. Some systems, such as TV-Scout and PTV, are web-based, whereas others, such as P-EPG and Multi-Agent, are STB-based [5,6].

Several methods for retrieving and filtering information from a TV program dataset have been proposed to allow a user to select a program based on topics of interest. However, a user is likely to spend a lot of time selecting programs; therefore, many recommendation systems for EPGs have been proposed for recommending TV programs according to users' habits [7,8]. However, the popularity of a TV program is also a key factor in whether a user selects to watch it, and the above recommendation systems cannot measure the popularity of a TV program.

Recent studies have been conducted on how to recommend the most popular TV programs, where a server is used for recording the users' behavior. Users are grouped with other users who have similar viewing patterns into a cluster during TV viewing. However, as the number of users increases, the server becomes the bottleneck of the system since grouping users into clusters is very time-consuming.

To lower the server's load, this work proposes a cloud-based program recommendation system (CPRS) for digital TV platforms. In CPRS, the time-consuming problem of grouping a large number of users into clusters is alleviated by using a map-reduce programming framework and a cloud computing technique.

The map-reduce programming framework was first introduced in [9]; it was originally designed for processing large amounts of data in a parallel way. In the map-reduce framework, a data processing task is automatically divided into a number of jobs in the map stage. Then, jobs are distributed to a cluster of machines for parallel processing. After jobs have completed, they are collected and combined in the reduce stage. Applying the map-reduce framework to a large cluster of machines increases both computation power and scalability; this is referred to as cloud computing.

In this paper, k-means and k nearest neighbor (kNN) are used to group users into clusters and to add a new user into a grouped cluster, respectively. Each program is assigned a weight, which is the sum of time periods that users have watched it. The weight of the program is used to indicate its popularity. Then, the k-means method is used to group users into clusters based on the weights. The kNN method is used to add a new user into a grouped cluster. Popular programs in the predicted cluster and in similar clusters are recommended to the user.

* Corresponding author.
E-mail addresses: cinfon@ieee.org (C.-F. Lai), changrh@mail.ncku.edu.tw (J.-H. Chang), cchu@cna.edu.tw (C.-C. Hu), huang@mail.ncku.edu.tw (Y.-M. Huang), hcc@mail.niu.edu.tw (H.-C. Chao).

0167-739X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2010.10.002

The viewership of TV programs is massive. Digital TV is also becoming increasingly popular; the number of available TV programs has correspondingly increased. A large dataset produces results that best represent a viewer's preferences of TV programs. However, large datasets require a lot of computation power. Cloud computing techniques can be used to analyze these datasets due to their computation power and scalable structure.

The rest of this paper is organized as follows. Section 2 describes related technologies and background. In Section 3, the architecture of CPRS is described. Section 4 describes the system implementation and experiment results. Finally, the contributions of this paper are summarized in Section 5.

2. Related works

Due to the increasing number of TV programs, choosing suitable programs has become a challenge. The number of studies on program recommendation has thus increased. Many methods have been applied to recommendation systems [10] for TV programs, such as fuzzy logic. In this paper, a system that uses *k*-means and kNN algorithms based on a cloud server is proposed. Details of the cloud computing, *k*-means, and kNN are described in this section.

2.1. Cloud computing

In recent years, cloud computing has become popular in both industry and academia. The concept of cloud computing is similar to that of distributed computing, but the key difference is the programming model: map-reduce. Map-reduce, which was first proposed by Dean et al. [11], makes clustering and parallel computing on a large set of machines easy and affordable. Yang et al. proposed a programming model called map-reduce-merge to support the need of using map-reduce to process multiple related heterogeneous datasets.

Many researchers have proposed novel ideas for using cloud computing to solve problems. McKinley et al. proposed a distributed infrastructure named a service cloud, which uses the concept of cloud computing. The service architecture is designed to facilitate the rapid prototyping and deployment of adaptive communication services. Cohen applied map-reduce to graph processing methods [12] and introduced map-reduce versions of several well-known graph processing methods. Grossman explained the current status of cloud computing [13].

Researchers have used cloud computing techniques to build applications or solve existing problems. Schaffer et al. described how NCSU's (North Carolina State University) virtual computing lab (VCL) applied cloud computing to achieve a cost-effective resource distribution of computation power [14,15]. The authors also mentioned that the scale and the utilization level are important for providing cost-efficient VCL services. Mika discussed the use of a semantic web content structure in the cloud [16] by using Yahoo! Pig to process RDF data in the cloud; the performance and working process were discussed in detail. Al-Zoube presented a cloud-computing-based solution for e-learning environments [17].

2.1.1. Overview of MapReduce

MapReduce [18,19] is a software programming framework introduced by Google. The main goal of using MapReduce is to develop a program to process a massive quantity of data over clusters of computers, in a distributed computing environment. The main purpose of MapReduce, to provide scalable computing power to solve mass information processing is also the core technology of cloud computing. The MapReduce framework provides the workflow for clusters of computers to handle input, output and processing. Since the design of MapReduce is revolutionary, to discover its potential in applying to different industrial fields has a become popular research topic in the academic field.

In MapReduce, there is a function called Map, and a function called Reduce. The Map function accepts key/value pair inputs, and outputs in another intermediate key/value format. The Reduce function will generate the final result in a key/value format, which merges the intermediate value according to the intermediate key in the output of the Map function.

Fig. 1 is the workflow of MapReduce. First, the master process of MapReduce framework splits the input data in M pieces. The worker processes are assigned a map reducing task by the master process. Each Map function in worker process analyzes the input key/value pair, processes it, then notifies the process address to the master process. The master process then passes the address to the reduce workers. The reduce workers use the Reduce function to merge or sort the processed results. When all results are processed and merged to a single result, the master process will raise a signal to notify the user that MapReduce is competed.[1]

2.2. K-means

K-means is an unsupervised learning algorithm. It can be used to classify or group objects or data into a number of k groups based on their attribute. The value k in k-means is a positive integer. The k-means algorithm has been widely applied to solve many data mining and information retrieval problems, such as market segmentation [20], recommendation systems [21], and dimension reduction [22]. Recently, k-means has also been applied to evidence accumulation (EAC) [23], the design of kernel methods [24], and the development of predictive QSAR models [25].

Given a positive integer k as a number of means, a set of means $C = \{c_1, c_2, \ldots, c_m\}$ where $m = k$. A set of objects or data $X = \{x_1, x_2, \ldots, x_n\}$ have n points to be classified into k groups. K-means clusters n points into k groups as follows. First, the k means in C are initialized randomly or by some heuristic. In the second step, each point in $x_i \in X$ is assigned to the nearest mean in $c_j \in C$, where $1 \le i \le n$ and $1 \le j \le m$:

$$\sum_{i=1}^{n} \min_{1\le j\le k} (x_i - c_j)^2.$$

After all points in X are grouped to a mean point, the centroids of each group are calculated as new means in the third step. The second and third steps are repeated until convergence is achieved.

In Fig. 2a, data are presented as black dots. When $k = 2$, k-means randomly generates 2 centers (which are shown as white dots). In Fig. 2b, black dots are assigned to the nearest center. If the object value does not converge, k-means regenerates each center based on black dots in the same group, as shown in Fig. 2c.

Assuming $k = 2$, the white dots are centers, and black dots are data. In Fig. 2a, 2 centers are randomly generated. In Fig. 2b, black dots are assigned to the nearest center. In Fig. 2c, the dots with a dotted line are new centers which are regenerated based on the black dots in the same group in Fig. 2b.

2.3. K nearest neighbors (kNN)

kNN [26] is a supervised learning method which classifies objects based on training data. It is generally used to solve

1 http://code.google.com/intl/zh-TW/edu/parallel/mapreduce-tutorial.html#MapReduce.