Platforms and Graphication Tools for a mobile application: Simulation of Planetary Objects' Trajectories

Analuz Silva Silverio[1,*], Mario A. De Leo Winkler[2], Antonio Alarcón Paredes[1]

[1]Universidad Autónoma de Guerrero, Unidad Académica de Ingeniería, Chilpancingo, Mexico
sisianlight@gmail.com,aalarcon@uagro.mx
[2]University of California, Physics and Astronomy Department, Riverside, USA
mario.deleo-winkler@ucr.edu

Abstract. During the development of mobile applications some of the first things that should be taken under consideration are the platform, code and development tools to be used. To do this, it is important to keep in mind compatibility of devices and operating systems, the focus audience, the project's budget, and other factors that end up being crucial for the successful completion of a project of this type. Because this particular case we want to develop applications for scientific outreach in astronomy, we investigate and present trends in developing mobile applications as well as propose a number of tools for our particular project, themed on planetary objects in a bi-dimensional simulation of positions, as well as the charted and tabulated data.

Keywords: Physics computing, Mobile computing, Native app, Application software, Open source software, Software tools, Programming environments, Computer science education, Astronomy.

1 Introduction

In mobile applications' (apps) development, one of the first things to take into account is the type of application that is being developed. For this reason, it becomes necessary to know positive and negative aspects in the trends of apps development.

Nowadays, the guidelines of app development have a wider scope than the initial ones where only two paradigms existed: native apps and web apps. Native applications are developed on the native language of an operating system (OS). A web application consists of a website specifically optimized with an interface and a set of functions to be used in mobile devices. Recently, with the arose of HTML5, the concept of hybrid applications has emerged and comes with added features and functionalities related to direct hardware control and the possibility to be implemented in different OS. Those new features have made hybrid apps become a new trend of development.

Collaborative Reputation Mechanism for Cloud Storage Service

Gerardo García-Rodríguez and Francisco de Asís López-Fuentes

Departament of Information Technology
Universidad Autónoma Metropolitana-Cuajimalpa (UAM-C)
Av. Vasco de Quiroga 4871, Cuajimalpa, Santa Fe
05348 Mexico City, México
{flopez}@correo.cua.uam.mx

Abstract. Cloud computing technology has emerged during the last years as a promising solution to transform the IT industry. Cloud computing is able to offer storage service, computing power and flexibility to end-users. Nevertheless, cloud computing technology still faces several challenges such as privacy, robustness, security and throughput. Distributed infrastructures as P2P networks have emerged as promising solutions for the management and storage all data. This paper proposes a collaborative mechanism based on reputation for cloud storage services. The proposed mechanism is implemented on a P2P infrastructure, which is used as an alternative platform for deploying storage services. Our solution integrates a qualified storage mechanism based on reliability indices. All peers collaborate to build the individual reputation of each peer in the storage system.

Keywords: P2P networks, Cloud computing, Distributed systems.

1 Introduction

Recently, cloud computing has become more popular, and many users and companies use these services to obtain several benefits such as more store capacity or computing power. Cloud computing is defined by M. Armbrust et al [2], as the applications delivery and services over the Internet, as well as the hardware and system software in the datacenters that provide these services. Based on this definition we can deduce that cloud computing is a model that allows access to files, applications or services in a ubiquitous and pervasive way through network in order to share a set of configurable computing resources. These resources can be servers, storage, applications and services, which can be rapidly provisioned and released with a minimal effort in service management or interacting with the provider. Therefore, cloud computing provides the illusion of unlimited and on-demand scalability. Essential characteristics of a cloud are [3]: on-demand self-service, network access, resource pooling, elasticity and measured service. In cloud computing two concepts are very basic: abstraction and virtualization [6].

Abstraction means that the implementation details of the system users and developers are abstracted. Therefore, applications run on physical systems that are not specified, the files are stored in places where users do not know their actual location, the system can be managed via outsourcing, and clients can access to the system in a ubiquitous manner. On the other hand, resources of the systems are pooled and shared in a virtualized way. Regarding virtualization, Sosinsky, B. states in [6] that systems and storage can be provisioned as needed from a centralized, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

Cloud computing introduces several benefits such as massive computing power, storage capacity and great flexibility; however this new computing paradigm still faces several challenges. Most popular cloud systems are centralized and its structure is based on the client-server paradigm. A centralized structure introduces several limitations such as storage dependence, privacy, scalability, privacy locally or connectivity. Peer-to-peer (P2P) networks have emerged as a promising distributed information management platform. A P2P network is a distributed network formed by a group of nodes, which build a network abstraction on top of the physical network, known as an overlay network. A peer can take the role of both, a server and of a client at the same time. An important advantage of these networks is that all available resources such as processing, memory and bandwidth are provided by the peers. Thus, when a new peer arrives to the P2P system the demand is increased, but the overall capacity too. This is not possible in a distribution infrastructure based on the client server model with a fixed number of servers. P2P paradigm allows that a system distributes its load and duties between all participating peers. Several examples of cloud computing platform based on P2P networks are reported in the literature [4-5], [7-9]. In this paper, we describe operation of our collaborative reputation mechanism for a storage service based on P2P cloud system [12]. Reputation systems has been proposed with the purpose to ensure that peers obtain reliable information about the quality of the resources they are receiving [11]. The main characteristic of our proposed mechanism is its reputation strategy which is based on based on reliability indices. These indices are totally transparent to the user as it is in the centralized cloud computing. Our project aims to integrate shared resources as volunteer computing to be used as a source of computing power and storage in social communities or research groups.

The remainder of this paper is organized as follows. In Section 2, we introduce our proposed model for a P2P cloud storage service. We present our collaborative mechanism in Section 3, and we explain its operation. This paper concludes in Section 4.

2 Architecture

A distributed storage system is an infrastructure that allows to store files in nodes, which are connected through a computer network. These systems are characterized by their wide range of applications such as backup files, sharing of files in network and edition of documents from different locations. This section introduces our proposed model. We propose a distributed storage service based on cloud computing, which takes advantage of the benefits introduced by the P2P networks. These networks have the advantage that all available resources are provided by the peers [1]. Peers can greatly benefit from the capacity of other requesting peers via collaboration. Thereby, file storage is realized in a decentralized manner and its location is completely transparent to the users. Our proposal differs from traditional distributed file systems which are equally used to share resources, but they have a lower level of transparency, and users know about the file system. A P2P Cloud system consists of peers which contain the client application. Peers come together to share their available hard disk capacity in order to create a total storage space together. The contents are split and each of these fragments is distributed within the peers. If content is not split, then it only is sent by the tracker to any of the nodes that is part of the cloud. Thus, a user can realize a replication of files within the cloud.

Our proposed architecture is shown in Figure 1. Reputation levels play an important role in our model, because they allow the system to define the way how files are stored within the peers. Our proposal considers five levels of reliability (or reputation) which are poor, fair, good, very good and excellent. These levels are recorded in each peer using values from 1 to 5, where 1 means that a peer has a poor reputation, while 5 means that a peer has an excellent reputation. Fair, good and very good scores are represented by the values 2, 3 and 4, respectively. When a tracker is created, it must contain the table of confidence (trust) levels which consists of an average of the levels of trust, dynamism and availability of each peer in the system.

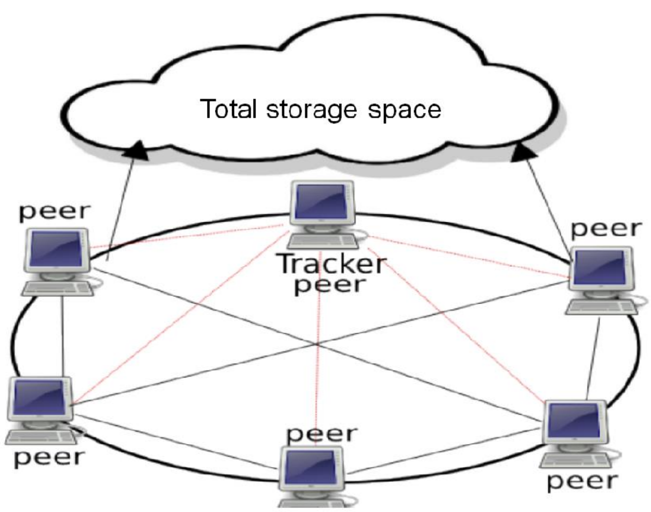

Fig. 1. Proposed architecture

The tracker decides where to store the files according to the level of reputation of each peer. For example, if the peer P1 has a level 5, then it can store its content on peers with same reputation level. In contrast, if P1 has a low reputation level its content is sent to peers with the same level of reputation. We use this strategy for the benefit of the cloud, and thus any type of contingency in the cloud as file access failure by peers with greater reliability. Figure 1 illustrates the tracker operation in the cloud. The tracker or server routes data and establishes communication between peers. These communications are based on the confidence level of each peer and contain the information required for each.

3 Mechanism

Our mechanism is composed by the following entities:

Peer. is an application hosted on each node in the P2P network, and it is responsible that each computer to be involved in the P2P cloud. Since a peer is an entity that receive and send files at the same time, we need to implement an application to perform these two tasks simultaneously. These tasks are realized through the peer application. Peer application also is responsible for monitoring each peer and reports its shared resources to tracker. Parameters to be monitored in each peer are its store capacity (trust), the number of disconnections during a day (dynamicity), its availability (number of storage requests that are rejected by a peer), and if a peer is cheating or no respect to a stored file. These three metrics are averaged in order to obtain the reliability level in each peer. After all, reliability is the metric used by a tracker to route content between peers. Peer application is formed by two parts: a server and a client. Server part always is listening in order to attend to other peers. In this case, to store files in the host computer. On the other hand, client part realizes different functions such as uploading files, display files and exit. Additionally, client program is who communicates with tracker in order to report all information about this computer in the cloud system. Figure 2 shows how an application hosted on a peer reports its data to the tracker while monitoring its host peer.

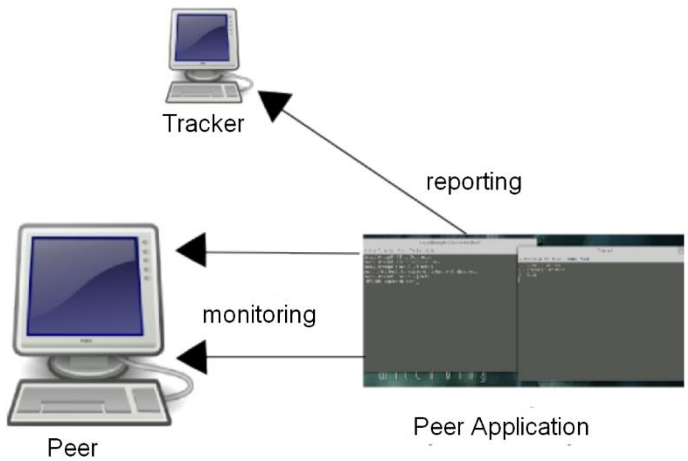

Fig. 2. Monitoring and reporting of the peer application

Tracker is responsible for system initialization. This application establishes the communications among peers. Figure 3 shows this scenario. Tracker applications also manipulates the database in which the reports generated by each peer and its contents are registered. Management of the database realized by the tracker is illustrated in Figure 4. It is important to note that the tracker is not a storage server, so the tracker never manage files. Tracker only redirects the contents received from a peer to another depending on levels of reliability of each peers. However, tracker offers localization transparency to all users in the systems. In this way, a peer sends its content to tracker, whom decides in which peer will be placed this content. This location is transparent for the requesting peer. When a requesting peer whishes recovery its content, it is requested via the tracker. Then, content is be addressed from the host peer to the requested peer. To allocate contents, the tracker queries the levels of dynamism, reliability and availability of each peer in the database and calculates an average integer value. The addressing is realized as follows. For example, if a peer P1 wishes to store a file in the cloud, it must submit a request to tracker, which selects the host peer. Peer P1 can receive an IP address to upload its file to the host peer by itself, or the file can be addressed by the tracker directly. Different host peers can have the same availability (or reputation) level. In this case, tracker selects peer with the largest storage space.

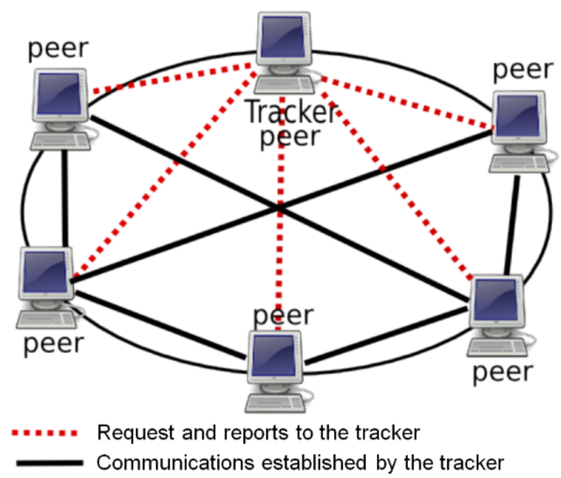

Fig. 3. Communication between nodes based on indications received from the tracker peer

Database, is designed to register and control all information received by the tracker from the peers. The database in this cloud system is based on a relational model. In its design we consider three entities: reputation metrics, peers and files. Recorded and monitored data in each peer are: physic address, IP address, date and time of the last connection, available space in disk, number of rejections, availability, dynamicity, true and reliability. Our design is based on the relational model and we used Wokbench [10] as a unified visual tool to produce a detailed data model of the database and tables. Visual database design is shown in Figure 5. Our scheme was developed in Spanish language. In this case, “confiabilidad” corresponds to reliability, while “dinamicidad”, “disponibilidad” and “confianza” correspond to dynamicity, availability, and trust, respectively. Our database also captures information related to the behavior of each peer in the system. That is, if a node is cheating or not with respect to a stored file. Data recorded for each file are its name, size and location. Files are localized using the peer’s physical address. Relational model is useful to maintain a right relationship between file, proprietary and storage place. Figure 6 shows how the peer P5 uploads a file to the cloud. We can see how the tracker is responsible for routing the file to a peer with the same level of reliability as P5. In this case tracker selects peers P3 to store the file of peer P5, because both peers have same level of reliability.

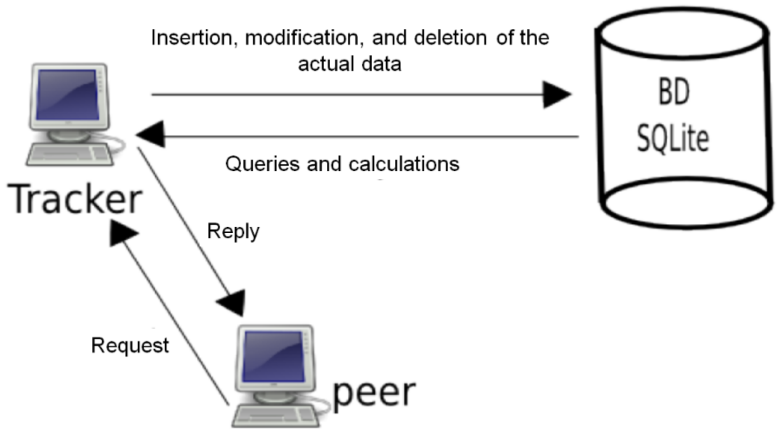

Fig. 4. Management of the database realized by the tracker peer

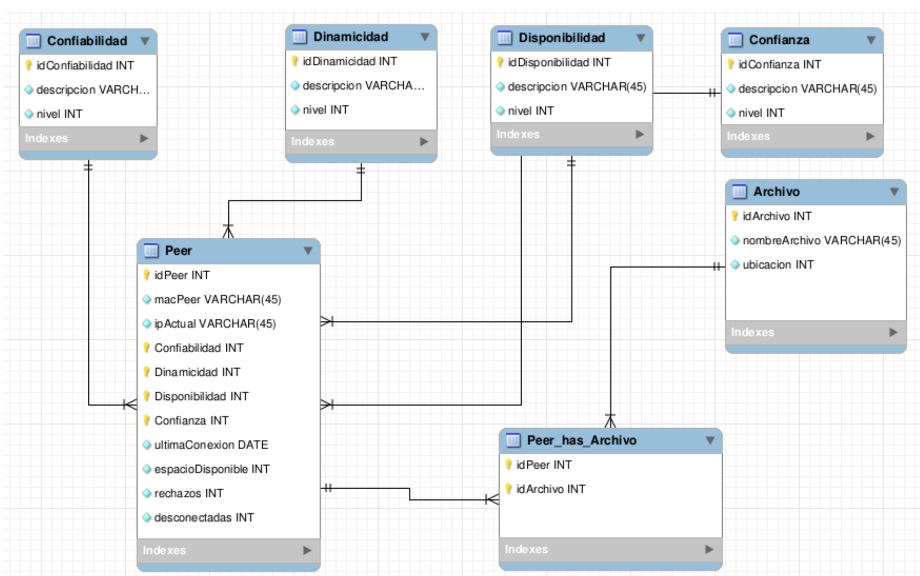

Fig. 5. Relational model of the database where all the information of peers and its contents are stored

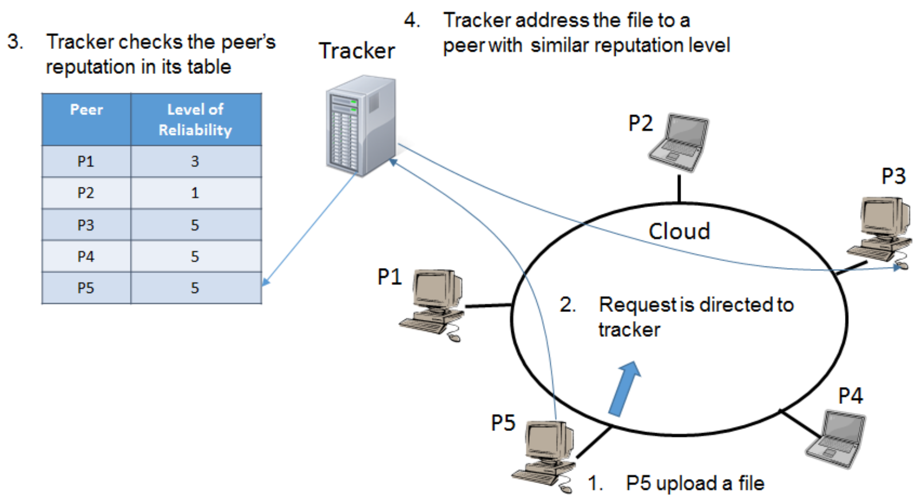

Fig. 6. System operation during a content distribution [12]

We implemented a small and experimental prototype of our proposal mechanism in our lab. This prototype has been developed on Linux Fedora version 19 using language C/C++. Our peers are communicated via TCP. We use this protocol because TCP is reliable, ordered and provides error-checked delivery between applications running on Internet.

4 Conclusions

Cloud computing have emerged as an ideal solution for storage service during the last years. This paper proposes a collaborative mechanism for cloud storage service based on P2P networks. Our solution exploits several characteristics introduced by the P2P networks such as collaboration, flexibility and scalability. Our proposed mechanism considers different approaches such as reputation and collaborative storage. Compared with other storage models such as cloud storage based on client-server, distributed storage or P2P distributed storage, our proposed model offers different benefits such as confidentiality, file sharing, data replication, data management, quality of service, decentralization, and transparency. A limitation of our proposed model is file fragmentation.

Although our experimental prototype was implemented using a reduced number of nodes, it can be scaled due to the properties of the P2P paradigm. In the future we plan to integrate a fragmentation policy in order to manipulate large multimedia contents.

Specifically, we are interested in adding a heterogeneous fragmentation policy in peers with heterogeneous reputation in our prototype.

References

1. Milojicic, D., Halogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard B., Rolling, S. and Xu, Z.: Peer-to-Peer Computing. In: HP Labs Technical Report HPL-2002-57, (2002).
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I. and Zanaria, I.: Above the Clouds: A Berkeley View of Cloud. In: Technical Report No. UCB/EECS-2009-28, University of California at Berkeley, Berkeley, CA, USA (2009).
3. Mell, P. and Grance, T.: The NIST Definition of Cloud Computing. In: Special publication 800-145 (draft), Gaithersburg, MD, USA (2011).
4. Lakshman, A. and Malik, P.: Cassandra - A Decentralized Structured Storage System. In: Proc. of the Workshop on Large-Scale Distributed Systems and Middleware (LADIS'09), Big Sky, MT, USA (2009).
5. Babaoglu, O., Marzolla, M. and Tamburini, M.: Design and Implementation of a P2P Cloud System. In: Proc. of the 27th Annual ACM Symposium on Applied Computing (SAC'12), pp. 412–417, Trento, Italy (2012).
6. Sosinsky, B.: Cloud Computing Bible. In: Wiley Publishing Inc., Indianapolis, IN, USA (2011).
7. Cunsolo, V., Distefano, S., Puliafito, A. and Scarpa, M.: Cloud@home: Bridging the gap between volunteer and cloud computing. In: Proc. of the 5th Int. Conference on Emerging Intelligent Computing Technology and Applications (ICIC'09), Ulsan, South Korea (2009).
8. Trajkovska, I., Salvachua Rodriguez, J and Mozo-Velasco, A.: A Novel P2P and Cloud Computing Hybrid Architecture for Multimedia Streaming with QoS Cost Functions. In: Proc. of the ACM International Conference on Multimedia (MM'10), pp. 1227-1230, Firenze, Italy (2010).
9. Xu, K., Song, M., Zhang, X., and Song, J. A.: Cloud Computing Platform based on P2P. In: Proc. of the IEEE International Symposium on IT in Medicine Education, pp. 427-432, Shandong, China, (2009).
10. MySQLWorkbench. Web site: http://www.mysql.com/products/workbench/.
11. Kamvar, S. D., Schlosser, M. T. and H. García-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: Proc. 12th Int. World Wide Web Conference, Budapest, Hungary (2003).
12. García-Rodríguez G., and López-Fuentes, F.: A Storage Service based on P2P Cloud System. In: Research in Computing Science 76, pp.89-96 (2014).
13. García-Rodríguez G.: Cómputo en la Nube basado en Redes Peer-to-Peer enfocados al Almacenamiento. Proyecto Terminal, Licenciatura en Tecnologías y Sistemas de la Información, UAM-Cuajimalpa, (2014).