Ripple: A publish/subscribe service for multidata item updates propagation in the cloud

Yuqing Zhu a,*, Jianmin Wang b,c,d,**, Chaokun Wang b

a Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
b School of Software, Tsinghua University, Beijing 100084, China
c Key Laboratory for Information System Security, Ministry of Education, China
d Tsinghua National Laboratory for Information Science and Technology(TNLList), China

Abstract

Cloud computing is becoming increasingly popular for the scalability and availability it promises. As existent cloud systems support only update interface on single data item, users and applications may require a convenient functionality that applies updates to a large number of data items. We propose an approach towards this end and exploit publish/subscribe service in update request dissemination. To effectively utilize computing resources of all nodes in a cloud, we devise a publish/subscribe service called Ripple which achieves the following goals: (1) fast connecting nodes according to their subscriptions on topics to exploit nodes in update request dissemination; (2) low cost in connection maintenance; (3) short delay for request dissemination. Extensive experiments based upon data collected from real Web applications demonstrate that Ripple satisfying the service requirement has good scalability and performs well even under conditions of data redistribution and node membership dynamicity. The service maintenance cost does not cause over stress on the underlying system.

1. Introduction

Cloud storage service promises scalability and availability. It gains increasing popularity both in industry and academy. A wide spectrum of applications, e.g. Yahoo! Facebook, Google Docs, Amazon WebStore and games,1 now take advantage of cloud storage. Each application has its own sets of data to operate on, and data sets of different applications may intersect. Users of one application possess personal sets of data, and users can also share parts of their data sets. A user’s data, e.g. emails, instant messages, browsing and shopping histories and social networks, may be accessed by multiple applications, each of which generally operates on data from multiple users. Among data items utilized by an application, some may belong exclusively to the application and not belong to any user. Users may also have exclusive ownership over parts of their data. This complicated relations between application-owned data and user-owned data can be depicted in Fig. 1.

A single data item, e.g. a key-valued record, may belong to multiple applications and user owners. This ownership, as well as access permission, varies from one data item to another. We may store such information local to each data item for easy access. An application or a user, both of which we denote as a client in the rest of the paper, can easily make updates to one data item or a number of data items, but the client may need to make updates to all or a large set of its data items, specifying the updates in rules. Consider the following two update examples:

1. A happy-farm like social application gives bonus to and doubles game points of users who have added more than 100 new friends in the past two months.
2. A user would like to update data privacy rules for application accesses, stating that (1) her profile information is denied for any access; (2) contents posted before this year and shared with more than 10 friends is allowed for access; otherwise, access is denied.

In this paper, we are interested in solving multidata item updates problem in a cloud system. Clients specify update rules, and the storage cloud system applies updates to a large number of its data items according to the rules. Efficient implementations of this function must avoid centralized processing and exploit all nodes in the cloud. We may use distributed computing paradigm like MapReduce (Dean and Ghemawat, 2004) or Pig (Olston et al., 2008) to apply update rules. However, such paradigm requires writing programs for every different update requests. The cloud storage system may need to provide such update facility to...