Technical Disclosure Commons

June 2020

Cache Coherence Protocol

Long Nguyen

Mohsen Vakilian

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

CACHE COHERENCE PROTOCOL

ABSTRACT

Disclosed herein is a cache coherence protocol for a distributed cache and a distributed strongly-consistent database in which an improved mechanism is provided for determining the validity of cached profile values and determining whether to update cached profile values. The mechanism can store profile values in a cluster. The mechanism can read a profile value from the cluster and store the profile value in a cache in connection with a read timestamp and a staleness value. The mechanism can detect an event for which the profile value is to be used. The mechanism can then determine, based on the read timestamp and the staleness value, whether the profile value stored in the cache is valid. The mechanism can use the profile value stored in the cache in response to determining that the profile value stored in the cache is valid. Alternatively, in response to determining that the profile value stored in the cache is not valid, the mechanism can update the profile value stored in the cache and use the updated profile value.

BACKGROUND

Services, such as social networking services, media content sharing services, search engine services, etc. collect volumes of data from many users. Data collected from a user may be stored as part of a user profile, which can be used for different purposes. For example, user profile information that indicates media content a user has previously viewed or indicated enjoyment of may be used to recommend other media content items to the user. In some cases, a user profile may be updated over time. For example, a user profile may be updated as a user performs different actions within a particular service, such as starting presentation of a particular media content item, sharing a media content item with a friend, etc.

It may be useful to store user profile information in a cache to allow the user profile information to be quickly accessed when needed. For example, at a time point at which a system determines that a media content recommendation is to be made, the system may access user profile information associated with a user corresponding to the recommendation. However, it can be difficult to determine whether cached user profile information is valid.

Thus, there is a need for an improved mechanism for determining the validity of cached profile values and determining whether to update cached profile values.

<u>DESCRIPTION</u>

The systems and techniques described in this disclosure relate to determining the validity of cached profile values and determining whether to update cached profile values. The system can be implemented on a server, such as a server associated with a social networking service, a media content sharing service, a search engine service, and/or any other suitable service. More particularly, the service can receive requests from users and can store information associated with requests in connection with user profiles. FIG. 1 illustrates an example process for determining the validity of cached profile values and determining whether to update cached profile values based on read timestamps and staleness values.
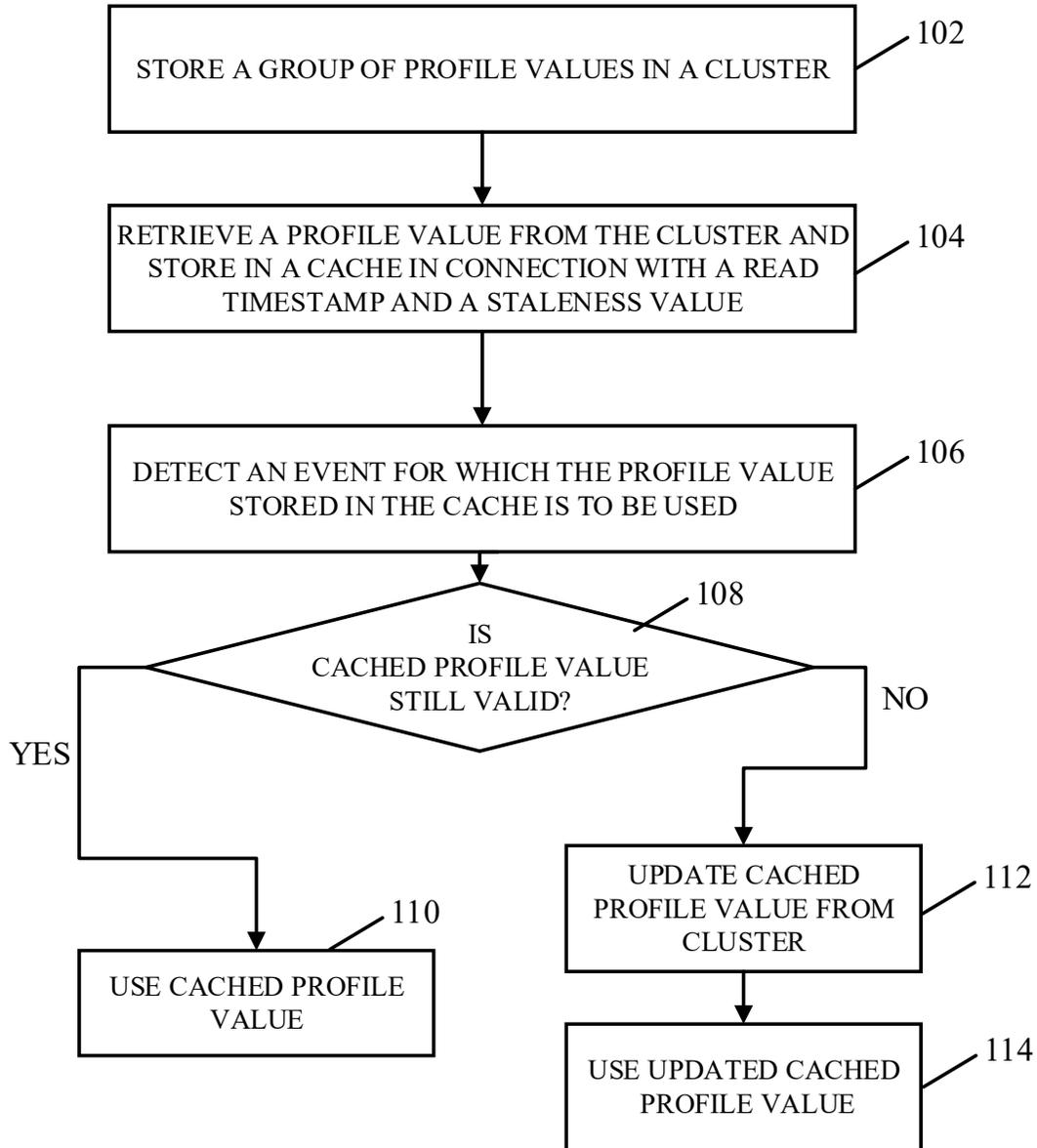
```
┌─────────────────────────────────────────────┐
│   STORE A GROUP OF PROFILE VALUES IN A CLUSTER │  — 102
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│ RETRIEVE A PROFILE VALUE FROM THE CLUSTER AND │  — 104
│   STORE IN A CACHE IN CONNECTION WITH A READ  │
│       TIMESTAMP AND A STALENESS VALUE         │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│  DETECT AN EVENT FOR WHICH THE PROFILE VALUE  │  — 106
│     STORED IN THE CACHE IS TO BE USED         │
└─────────────────────────────────────────────┘
                      │
                      ▼
                  ◇ 108
                 IS
         CACHED PROFILE VALUE
            STILL VALID?
   YES ◄──────            ──────► NO
```

IS CACHED PROFILE VALUE STILL VALID? — 108

YES

NO

UPDATE CACHED PROFILE VALUE FROM CLUSTER — 112

USE CACHED PROFILE VALUE — 110

USE UPDATED CACHED PROFILE VALUE — 114

FIG. 1

At 102, the system can store a group of profile values in a cluster. In some instances, the

cluster can be any suitable type of distributed cluster or distributed database. In some instances,

different profile values in the group of profile values can correspond to different users, for

example, different users of a service associated with the system. In some instances, a profile

value can indicate any suitable information about a user. For example, in some instances, a

profile value can indicate information about a demographic of the user. As another example, in some instances, a profile value can indicate actions performed by the user. As a more particular example, in some instances a profile value can indicate media content consumed by the user (e.g., viewed, downloaded, etc.), media content or other content endorsed or shared by the user, days or times the user has used the service, and/or any other suitable type of profile information.

In some instances, the system can store the group of profile values in the cluster in any suitable manner. For example, in some instances, the system can write each profile value to a database associated with the cluster. Note that, in some instances, each profile value can be associated with an identifier of a user corresponding to the profile value. Additionally, note that, in some instances, the system can store profile values in the cluster at any suitable time points, and at any suitable frequency. For example, in some instances, the system can update a profile value associated with a particular user at any suitable time points (e.g., once per hour, once per day, and/or any other suitable time points) and/or based on any suitable events (e.g., in response to a user performing any suitable action in connection with the service, and/or any other suitable events).

At 104, the system can retrieve a profile value from the cluster and can store the retrieved profile value in a cache in connection with a read timestamp and a staleness value. In some instances, the cache can be any suitable memory associated with the system. In some instances, the system can retrieve a profile value and store the retrieved profile value in the cache at any suitable time points and at any suitable frequency.

In some instances, the read timestamp can indicate any suitable information. For example, in some instances, the read timestamp can indicate a timestamp from a clock associated with the cluster corresponding to a time point at which the profile value was retrieved from the

cluster. In some instances, the staleness value can be a time duration over which the profile value is valid. In some instances, the staleness value can be any suitable value (e.g., one hour, one day, one week, and/or any other suitable duration). Note that, in some instances, the read timestamp and the staleness value can be provided by the cluster.

At 106, the system can detect an event for which the profile value stored in the cache is to be used to perform an action. In some instances, the event can be any suitable event. For example, in some instances, the event can correspond to detecting that content personalized to a user corresponding to the profile value is to be presented. As a more particular example, in some instances, the event can correspond to detecting that media content recommendations for the user corresponding to the profile value are to be presented. As another more particular example, in some instances, the event can correspond to detecting that a curated social networking feed is to be presented. In some instances, the system can detect the event in any suitable manner. For example, in some instances, the system can detect that a page that includes a section that includes personalized content (e.g., media content recommendations, a curated social networking feed, and/or any other suitable personalized content) is to be presented. Note that, in some instances, the event can be associated with an event timestamp, which can indicate a time at which the event was detected.

At 108, the system can determine whether the cached profile value is still valid. In some instances, the system can determine whether the cached profile value is still valid based on the read timestamp and the staleness value received from the cluster in connection with the profile value, as described above in connection with 104. For example, in some instances, the system can subtract the staleness value from the event timestamp and can compare the difference to the read timestamp. As a more particular example, in an instance in which the event timestamp is

1:05:45, and in which the staleness value is 5 minutes, the difference between the event

timestamp and the staleness value can be 1:00:45. Continuing further with this particular

example, in an instance in which the read timestamp is prior to the difference between the event

timestamp and the staleness value, the system can determine that the cached profile value is no

longer valid ("NO" at 108). As a specific example, in an instance in which the read timestamp is

54:00, the system can determine that the read timestamp is prior to the difference between the

event timestamp and the staleness value (i.e., 1:00:45, as calculated above), and can therefore

determine that the cached profile value is no longer valid. Conversely, in an instance in which

the read timestamp is after the difference between the event timestamp and the staleness value,

the system can determine that the cached profile value is still valid ("YES" at 108). As a specific

example, in an instance in which the read timestamp is 1:02:00, the system can determine that

the read timestamp is after the difference between the event timestamp and the staleness value

(i.e., 1:00:45, as calculated above), and can therefore determine that the cached profile value is

still valid. It should be noted that, in some instances, the system can determine that the cache

profile value is still valid and that a more recent profile value should be dropped so that an older

profile value can be used.

If, at 108, the system determines that the cached profile value is still valid ("YES" at

108), the system can use the cached profile value to perform any suitable action(s) in connection

with the detected event at 110. For example, in an instance in which the event corresponds to

detecting that a page that includes personalized content is to be presented, the system can cause

the page to be presented with personalized content that is identified based on the cached profile

value. As a more particular example, in some instances, the system can identify media content

items to recommend based on the cached profile value and can cause a page that includes the

identified media content items to be presented. As another more particular example, in some instances, the system can identify social media posts to include in a curated feed based on the cached profile value and can cause a page that includes the curated feed to be presented.

If, at 108, the system determines that the cached profile value is not still valid ("NO" at 108), the system can update the cached profile value from the cluster at 112. In some instances, the system can update the cached profile value in any suitable manner. For example, in some instances, the system can query a database corresponding to the cluster for an updated profile value stored on the cluster and can store the updated profile value in the cache.

The system can then use the updated cached profile value to perform any suitable action(s) in connection with the detected event at 114. In some instances, the system can use the updated cached value to perform any suitable action(s), such as described above in connection with 110.

Accordingly, a cache coherence protocol for a distributed cache and a distributed strongly-consistent database in which an improved mechanism is provided for determining the validity of cached profile values and determining whether to update cached profile values is provided.