Location-aware data management for mobile applications

Luís M. Silva, João Leitão and Nuno Preguiça
NOVA LINCS & NOVA School of Science and Technology
lmt.silva@campus.fct.unl.pt,jc.leitao@fct.unl.pt,nuno.preguica@fct.unl.pt

ABSTRACT

Over the last decade, mobile applications increased the volume of location-dependent information shared among users, from collaborative applications and social networks to location-based games (e.g. augmented reality). Clients tend to hold an interest in data that is conceptually near them (i.e. associated with their location). Addressing the inherent challenges of this emergent paradigm can improve systems by the amount, detail, and freshness of data exposed to clients. Our solution consists of a system that dynamically adapts the consistency level of data that is exposed according to the distance of that data from the device requesting it. Being closer to the data location automatically means having stronger consistency guarantees. As a device moves further away from the data, the consistency guarantees become laxer while enabling the system only to expose aggregations or summaries of such data.

KEYWORDS

Location-dependent Data, Edge Computing, Dynamic Consistency

1 INTRODUCTION

Mobile devices have become the preferred platform for deploying new applications as modern applications shift towards a paradigm of significant interaction among users. Supporting effective data sharing among mobile devices has become an increasingly important problem. [3]

Many applications base their concept on social and collaborative features that directly relate to the user's location. Examples include vehicular applications, where users share traffic, incidents, and parking information in a real-time crowdsourced environment. Also relevant are mobile multiplayer games where users interact with other nearby users and virtual objects that map to real-world locations.[1]

These applications need to share and manipulate location-dependent data, requiring novel data management solutions for systems that span from cloud data centres to edge servers and end-user devices.

2 THE PROBLEM

We see an opportunity to improve applications that share and manipulate location-dependent data. Most current applications rely on centralised infrastructures for data storage and coordination. With a fast-expanding number of clients, such solutions can easily become a bottleneck limiting the interactivity between users, particularly those in close vicinity. As in such applications, every client interaction translates to a request to a cloud server that might need to be propagated to users nearby.

In the type of applications described above, the relevance of data (to users) is intrinsically entwined with space, meaning the proximity between the user location and the location associated with the data. Data linked to locations closer to the user will be most accessed and manipulated most often. As a client moves further away from an object, data becomes less relevant.

The last suggests that a system does not need to provide the same consistency level for all data. More distant data can experience less strict levels of consistency than closer data. Intuitively, a user's interest in a data object is somewhat proportional to its distance. [2] By blending that with looser consistency constraints, the need to maintain the most up-to-date version of that object also loses importance.

Commercial data stores are designed to address many use cases, thus being general purpose. They show a need to make inevitable trade-offs. However, some do offer tunable consistency guarantees. A concession is necessary; developers must choose the consistency constraints guarantees from the start, even if that feature is available on multiple levels - transaction, bucket or the database.

The argument above indicates that such a problem must be solved in an ad-hoc fashion for most applications, demanding the implementation of custom mechanisms at the application level and without support from datastores. A more suitable solution is to rely on a datastore with mechanisms to deal with the dynamic nature of users' interest in data, given their evolving locations.

3 A USE CASE

This article focuses on a simple example to motivate and explain this need for a datastore that provides a new dynamic and tunable consistency model that is location dependent. We focus on an application that allows users to find available parking locations.

Where to Park? App A user is travelling and wants to park at the end of that trip. The journey's destination is known, desiring to park as near as possible to its destiny. He wants to know the potential free parking spaces around that location during the trip. As he approaches the destination, more precise information is required. However, the exact location or number of available parking spaces is not required nor needed at a significant distance. At that point, the user may only need to know if he will be able to park in that neighbourhood. As the trip progresses and the user becomes ever so closer, the precision and accuracy of the parking information starts to become increasingly relevant. When the user reaches or crosses the neighbourhood's outskirts, he may find it helpful to learn of streets with available parking spots at that moment. Since provided with that information, he can decide whether to go directly to the destination or slightly change the route to encounter a street with empty parking places. When advancing even further, entering a specific street, it would be perfect if the user could know the exact location where to park. Thus, all the information about the free parking places should be presented to the user, but only for that street.

4 TOWARDS A SOLUTION

The main focus is exploiting all the system components' locations for performance gains through several steps. As defining a hierarchy among system nodes, and mapping the dynamic location changes to the interest in data, thus creating a data model that provides clients with multiple levels of data exposure and consistency.

As the distance between a user and data increases, the detail is intended to diminish gradually and gracefully. To the point where the most remote data loses so much detail that it is only represented as a result of a function (e.g., aggregation). Moreover, the distance should impact the delivery of updates to clients, implicitly exploiting a trade-off between data freshness and resource consumption (both for infrastructure and the user's mobile device). Contrarily, consider that a user moves closer to an object's location - the level of detail exposed to the user should automatically increase. In our example, when requesting data about streets closer to the user, the available parking spaces have their location shown in full detail. In streets that are further away, only the (approximated) number of spaces is relevant. Beyond that, just the existence of any parking spaces in the neighbourhood can be exposed.

The proposed strategy entails a dynamic variation of the consistency level of data exposed to different users. Offering the opportunity to reduce the coordination between data replicas and the client applications by softening some freshness constraints. The payoff is reducing the load imposed by replicating data objects among interested parties. Such gains are crucial for mobile settings with computation, energy and bandwidth restrictions.

Our contributions to solving the problem The novelty in our solution lies in devising a data model where location is a fundamental property entwined with the operation of the datastore, akin to time in time-series databases. That is associated with both the clients and data to provide multiple hierarchical degrees of consistency. On a system level, developers should define areas of influence for data. That is a well delimited geographical or logical area that will determine data fidelity when a client is within the bounds of that region. To regulate what to expose, we employ views as the result of applying a function to a data collection. The outcome is a subset of that collection or a collection of derived objects containing fewer details than the original or forming a summary value. That calls for the design of algorithms capable of propagating updates throughout the network to clients and servers.

In summary, this work pursues the following goals: First, devise efficient algorithms to disseminate location-dependent data combining peer-to-peer and edge-server mediated interactions exploiting 5G and V2X communications. [4] Second, it proposes novel replication schemes where consistency and fidelity are location-dependent and gracefully degrade as users move away from data locations. Third, we will study how to mitigate the inherent security and privacy issues involved in sharing location-dependent data in a distributed context.

REFERÊNCIAS

[1] Carsten Magerkurth, Adrian David Cheok, Regan L Mandryk, and Trond Nilsen. 2005. Pervasive games: bringing computer entertainment back to the real world. *Computers in Entertainment (CIE)* 3, 3 (2005), 4–4.

[2] Nuno Santos, Luís Veiga, and Paulo Ferreira. 2007. Vector-field consistency for ad-hoc gaming. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 80–100.

[3] Mahadev Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (2017), 30–39.

[4] Luís M Silva, Frederico Aleixo, Albert van der Linde, João Leitão, and Nuno Preguiça. 2022. Geo-located data for better dynamic replication. *arXiv preprint arXiv:2205.01045* (2022).