Demo: Seamless Transitions Between Filter Schemes for Location-based Mobile Applications

Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe and Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
E-Mail: {firstname.lastname}@kom.tu-darmstadt.de

***Abstract*—A vast amount of mobile applications utilizes the current location of the user as key input parameter to determine relevant content. At the same time, these applications become more focused on direct interaction between users within close proximity. One popular example are mobile augmented reality multiplayer games, such as Google's Ingress or Pokémon Go. To support interaction between users in such interactive applications, publish/subscribe and its event filter schemes can be utilized on the server side. In this demonstration, we showcase the concept of executing transitions between such filter schemes to achieve high adaptivity in the face of user dynamics, as proposed in our accompanying conference paper. Attendees can explore the execution of such transitions and their implications on performance and cost of the system in an interactive demonstration setup. Our demonstration allows attendees to interact with the scenario via a web-based application using their own smartphones.**

I. Introduction

With the trend towards mobile applications and the increasing number of powerful hand-held devices, a new breed of interactive location-based applications is on the rise. A prominent example are interactive mobile augmented reality games, with Pokémon Go being a recent and vastly successful addition. From a communication perspective, these applications produce a number of events based on direct player interactions or caused by a player's movement, that are to be distributed to other devices within a given proximity, usually referred to as area of interest. Relying on the popular publish/subscribe paradigm for message filtering and distribution, an application specifies interest in events using subscriptions that include the client's current physical location. Filtering, which is usually done on cloud-based broker networks, matches incoming events to the current set of subscriptions for later distribution to interested users. In location-based mobile applications, the subscriptions would need to be updated whenever the respective client moves, as otherwise they become inaccurate. This causes significant communication overhead and limits the potential for optimizations within the broker network, which are usually targeted at rather long-lived subscriptions.

In recent years, different schemes have been proposed on how to filter events within a publish/subscribe system based on a user's physical location [1]–[3], mostly with the goal of limiting the number of re-subscriptions required to maintain accurate event delivery. However, these filter schemes are usually targeted towards a specific expected client behavior in terms of dynamics and workload characteristics. Therefore, we proposed a method for transitions between such filter schemes during runtime of a publish/subscribe system, allowing the system to adapt to client dynamics while at the same time enabling a fine-grained configuration of the observed performance and cost trade-offs [4]. We briefly introduce this work in the following section, before detailing the demonstration setup and its opportunities to study the impact of the proposed filter scheme transitions in an interactive setting.

II. Seamless Filter Scheme Transitions

As briefly motivated in the previous section, executing transitions between different filter schemes for location-based publish/subscribe is a viable way to adapt the overall system to client dynamics. However, the transition itself involves coordination with mobile clients as well as state transfer operations at the broker. Furthermore, missing state might need to be requested from clients, if it was not stored within the previously active filter scheme. This is best explained considering the example shown in Figure 1. Here, three schemes are considered: (i) a location-based scheme (LPS) relying on a client's location and a radius defining the area of interest, (ii) a grid-based scheme (rGrid), where a client subscribes to all rectangular cells that intersect with its area of interest, and (iii) a simple grid-base scheme (Grid), where clients only subscribe to a single grid cell that covers their current position. The state transformations and lost state during transitions is illustrated at the respective arrows: While the transitions at the top of the diagram can be realized solely with state transformations, the transitions at the bottom require state updates from clients, as not all required information is available at the broker.

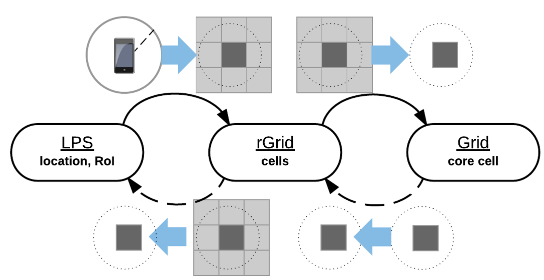

Fig. 1. Example transitions between three filter schemes. State transformation and missing state information is illustrated for the respective transitions. Figure taken from [4].

The transition from LPS to rGrid, for example, can be executed at the broker without additional state updates, as the current location of a client and the radius of its area of interest is known. The broker can, thus, calculate the set of grid cells the client needs to be subscribe to and transform the subscription accordingly. However, when switching from rGrid back to LPS, only the set of cells is known for each client. Thus, we can only estimate the rough position of a client and a pessimistic region of interest (given that the region of interest is circular) – for accurate event dissemination the client needs to update the state at the broker by sending its current location and radius of interest.

For an in-depth discussion of transitions between filter schemes and our proposed system design, the interested reader is referred to our accompanying conference paper [4].

III. Demonstration

To demonstrate the impact and effects of filter scheme transitions for mobile location-based applications, we rely on the simulation and prototyping platform SIMONSTRATOR [5]. The platform allows us to simulate a larger quantity of mobile clients and their activity in a mobile location-based application. Client movement is based on OpenStreetMap data, as clients follow streets to reach a set of attraction points. These attraction points correspond to points of interest within the application. Clients issue events containing their current game status as well as their location periodically, and the broker distributes these events to subscribed clients based on the currently utilized filter scheme.

For this demonstration, we extend our system model presented in [4] with a REST-based API that allows live client manipulation and returns detailed per-client status information. Relying on an interactive web application, conference attendees can take control of the simulated mobile clients via one of the provided smartphones or by using their own device. The web application reports the currently experienced performance and cost metrics for the respective client and allows manipulation of client movement and application properties. The extensions to the SIMONSTRATOR platform developed for this demonstration are available online[1] and might serve as a starting point for similar demonstration setups.

A second web application running on a tablet device allows attendees to trigger different filter scheme transitions and reconfigurations of the currently active filter scheme. Additionally, broker-related performance and cost metrics related to a given filter scheme are visualized on the device. Metrics observed by clients (e.g., amount of status updates, achieved accuracy, traffic characteristics) are directly visualized on the participants' smartphones. Thereby, the demonstration provides two views on the impact of filter scheme transitions: from the perspective of the broker (or, in more general terms, the application provider) and from the perspective of a mobile client. In addition to the web-based applications, a central visualization illustrates the overall scenario from a bird's-eye view. The overall setup of the demonstration is illustrated in Figure 2. It consists of a computer executing the simulation environment and providing a central visualization of the map and the mobile clients, and two or more mobile devices connected to the laptop via a Wi-Fi access point.

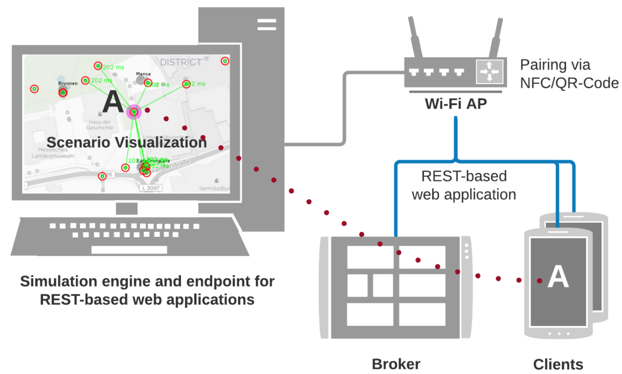

Fig. 2. Setup of the demonstration, consisting of a computer running the simulation software and visualizing the scenario, a Wi-Fi access point and two or more mobile devices. Two web-based control applications enable interaction with the broker and the clients in the simulation, respectively.

A. Interaction with the Broker

Using the broker-side application running on the provided tablet, attendees can manually trigger transitions and reconfigurations of filter schemes. The resulting performance and cost metrics as observed by the broker are visualized within the application. This allows attendees to assess the impact of a transition in terms of client state updates, required match-operations on the broker as well as traffic required to distribute state updates.

Furthermore, global scenario parameters such as the number of active clients can be altered to directly study the impact of different node densities and workload characteristics. In combination with the client application introduced in the following section, this enables conference attendees to explore the performance and cost characteristics of the individual filter schemes and their transitions, augmenting the results presented in the conference paper [4]. One example is the grid-based filter scheme, where attendees can reconfigure the size of the utilized grid cells to study the trade-off between precision and recall for individual users and different sizes of their region of interest. Furthermore, the effects of suboptimal grid cell placement become apparent, motivating the future use of map-based information on places, streets, and natural obstacles when determining the individual grid cells.

B. Interaction with Mobile Clients

Interested attendees can use one of the provided devices or connect their own smartphone to our access point[2] to gain access to the client-side application. As the client-side application is purely web-based, no additional application installation is required. The client-side application is assigned

[1] Git-repositories at http://simonstrator.com (not all sources are publicly visible, please contact the authors for full access).

[2] Credentials are provided via a QR code and an NFC-tag for easy setup.

randomly to one of the simulated mobile clients, and the respective client is highlighted in the central visualization component (c.f. Figure 2). On their mobile device, attendees can observe relevant performance and cost metrics for their client, such as the traffic and relevance (recall, precision) of the received events, represented as interactive charts. Additionally, the currently used filter scheme is indicated. Attendees can control the behavior of their assigned client through the client-side application. The application provides means to alter (i) the client's movement speed, (ii) the target point of interest, (iii) the region of interest, and (iv) the frequency of events.

To study the resulting effects for larger groups of clients, the application allows to optionally apply the changes not only to the currently assigned node, but to a fraction of all nodes. Consequently, interesting scenarios such crowds forming nearby a specific point of interest or a sudden increase in the workload can be explored. Combined with the broker-side interaction, the impact of transitions in such scenarios and the performance and cost characteristics of individual filter schemes can be explored.

IV. Conclusion

We believe that the proposed demonstration helps in understanding and exploring the effects of transitions between filter schemes in an interactive fashion, and thereby poses a valuable addition to our main conference paper [4]. By combining a simulation-based approach with direct – and potentially concurrent – interaction through a REST-based web application, conference attendees can trigger transitions and alter client and broker behavior to study interesting use cases (such as spontaneous gatherings and sudden increases in workload). As client and broker metrics are visualized directly on the respective mobile devices, the impact of transitions from both perspectives becomes clearly visible.

V. Technical Requirements

The demonstration consists of a laptop, a Wi-Fi access point, and two hand-held devices, all provided by the authors. It requires a small table (preferably a bar table for easier interaction with attendees) and two power outlets. If available, we would kindly ask for an additional computer monitor to increase the visibility of the central visualization (shown on the laptop screen otherwise). Dedicated access to the Internet is *not* required for this demonstration. Setup of the demonstration requires five to ten minutes.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of projects C2 and B1 within the Collaborative Research Center (CRC) 1053 – MAKI.

References

[1] A. Brimicombe and Y. Li, "Mobile space-time envelopes for location-based services," *Transactions in GIS*, vol. 10, no. 1, pp. 5–23, 2006.

[2] K. Jayaram, P. Eugster, and C. Jayalath, "Parametric content-based publish/subscribe," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 2, p. 4, 2013.

[3] I. Leontiadis, P. Costa, and C. Mascolo, "A hybrid approach for content-based publish/subscribe in vehicular networks," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 697–713, 2009.

[4] B. Richerzhagen, N. Richerzhagen, J. Zobel, S. Schönherr, B. Koldehofe, and R. Steinmetz, "Seamless transitions between filter schemes for location-based mobile applications," in *Proc. 41st IEEE Conference on Local Computer Networks (LCN) [Accepted for Publication]*, 2016, pp. 1–9.

[5] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications," in *Proc. EAI Conference on Simulation Tools and Techniques (SIMUTOOLS)*. ACM, 2015, pp. 99–108.