WebGC: Browser-Based Gossiping [Live Demo/Poster]
Raziel Carvajal-Gómez, Davide Frey, Matthieu Simonin, Anne-Marie Kermarrec

To cite this version:
Raziel Carvajal-Gómez, Davide Frey, Matthieu Simonin, Anne-Marie Kermarrec. WebGC: Browser-Based Gossiping [Live Demo/Poster]. Middleware 2014, Dec 2014, Bordeaux, France. 10.1145/2678508.2678515 . hal-01080032

HAL Id: hal-01080032
https://inria.hal.science/hal-01080032
Submitted on 4 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright
WebGC: Browser-Based Gossiping

[RLive Demo/Poster]

Raziel Carvajal-Gómez, Davide Frey, Matthieu Simonin, and Anne-Marie Kermarrec
name.surname@inria.fr
INRIA Rennes, France

ABSTRACT

The advent of browser-to-browser communication technologies like WebRTC has renewed interest in the peer-to-peer communication model. However, the available WebRTC code base still lacks important components at the basis of several peer-to-peer solutions. We tackle this problem by proposing WebGC, a library for gossip-based communication between web browsers. Due to their inherent scalability, gossip-based, or epidemic protocols constitute a key component of a large number of decentralized applications. WebGC thus represents an important step towards their wider spread.

Categories and Subject Descriptors
I.5.3 [Clustering]: Algorithms; D.4.7 [Organization and Design]: Distributed systems

General Terms
Design

Keywords
Real Time Communication, Peer-to-Peer, Gossip based algorithms

1. INTRODUCTION

Epidemic (gossip) protocols have received enormous attention by the research community due to their inherent scalability, resilience to failures, and wide applicability. Recent work has exploited epidemic protocols in a variety of decentralized applications ranging from video streaming to recommendation systems [4]. Such applications follow a peer-to-peer interaction model and require users to install a specific piece of software [7] to handle peer-to-peer communication. In a world where more and more applications run within web browsers, this represents a clear disadvantage for the success of epidemic protocols.

Our demo and poster address this concern by presenting WebGC (Web Gossip Communication), the first—to the best of our knowledge—browser-based library for epidemic protocols. WebGC builds on the recent introduction of WebRTC [1], and on two related projects: PeerServer and PeerJS [2]. WebRTC extends browsers with real-time communication (RTC) capabilities. PeerServer bootstraps the connections between peers, while PeerJS provides high-level communication primitives that serialize messages through WebRTC streams.

Our demo will feature a protocol stack building an interest-based topology [4]. Participants will connect to a demo server with their own laptops and will enter keywords representing their interests on a web page. Their browsers will then initiate gossip communication and build a semantic overlay—grouping users with similar interests—while visualizing its evolution in real time.

2. WEBGC LIBRARY

The left of Figure 1 positions WebGC in the context of PeerServer, PeerJS, and WebRTC. WebGC uses a customized version of PeerServer to bootstrap gossip protocols. It then translates the peer identifiers, obtained from PeerServer or through the gossip protocols themselves using ICE (WebRTC’s NAT traversal framework). Finally, it uses primitives provided by PeerJS to transfer data to and from other peers.

WebGC Internals.

The right of Figure 1 depicts WebGC’s architecture. Its core consists of a COORDINATOR object inheriting from PeerJS. The COORDINATOR instantiates the gossip protocols and acts as a communication broker dispatching incoming messages to the various protocols. The library currently includes the implementation of two peer sampling protocols, CYCLON [10] and the generic protocol suite from [5], as well as a clustering protocol [11, 4]. All protocols implement a GOSSIPPROTOCOL “interface”—since Javascript does not natively support interfaces, we adopt the interface pattern described in [6]. The COORDINATOR makes it possible to stack these protocols on top of each other [4] to implement applications.

The GOSSIPPROTOCOL interface follows the scheme proposed in the literature [9, 8, 5] and defines the high-level operations that
WebGC provides signaling, while PeerJS implements serialization and provides high-level communication primitives for peer-to-peer applications. WebGC exploits these tools to provide an easy-to-use programming framework for gossip-based applications.

Although WebGC constitutes, to the best of our knowledge, the first gossip-oriented library over WebRTC, a lot of existing work has concentrated on the modular implementation of gossip protocols [8, 5, 9, 3]. The authors of [8] propose a decomposition of gossip applications into standard building blocks. [5] present a family of random-peer sampling protocols by highlighting the common structure exhibited by gossip-based overlay-maintenance solutions. Finally [9] and [3] propose two libraries for the development and deployment of gossip-based protocols. However, both libraries only provide a Java implementation and do not support operation within a web browser.

5. CONCLUSIONS

Web browsers have become powerful enough to support applications as complex as desktop applications. The emergence of platforms such as HTML5/Javascript and WebRTC have even suggested the idea of a web-based operating system. In this context, this demo and poster fill an important gap by showcasing a library for the deployment of gossip-based applications within web browsers. The demo demonstrates the effectiveness of our library in a real context, while the poster provides details on the structure of the library, on how to build applications, and on how to implement new protocols. We believe that our framework may give a second wind to gossip-based peer-to-peer applications. Yet it also needs to evolve. WebGC currently relies on WebRTC’s heavyweight connection-setup. We are therefore investigating how to replace this standard signaling mechanism with cheaper fully decentralized solutions.

6. REFERENCES