Improve Distributed Client Lifecycle Control in ShadowStream

Junhua Yan, Huazhong University of Science and Technology, Wuhan, China
Chen Tian, Huazhong University of Science and Technology, Wuhan, China
Jingdong Sun, Huazhong University of Science and Technology, Wuhan, China
Hanzi Mao, Huazhong University of Science and Technology, Wuhan, China

ABSTRACT

ShadowStream is a novel Internet live streaming system that integrates performance evaluation as an intrinsic capability. An essential component in ShadowStream is distributed lifecycle control mechanism, which assigns each client a virtual arrival/lifetime to create a particular scenario to evaluate the performance of streaming system. The original design focuses on utilizing stable streaming viewers in physical world to guarantee the accuracy of ShadowStream, which, on the other hand, significantly limits the scale of the experiment. The authors’ research develops a novel distributed client lifecycle control to get rid of restrictions caused by the limited number of stable viewers in live-testing streaming networks. The core idea of their research is to match the desired experimental scenario with real viewers’ behavior in physical world. The result demonstrates that with the authors’ methodology, the scale of experiments can be doubled.

Keywords: Lifetime, Live Streaming, PCE, ShadowStream, User Behavior

1. INTRODUCTION

ShadowStream (Chen&Richard, 2012, pp. 347-358) is a novel Internet live streaming system that integrates performance evaluation as intrinsic capability. It introduces a novel production-CDN-experiment (PCE) streaming machine layout to protect real viewers’ quality-of-experience (QoE) in experiment, at the same time gets accurate results.

An essential component in ShadowStream system is lifecycle control. In general, it lets a production viewer participate in the experiment by assigning arrival time and lifetime to emulate a virtual client’s arrive and departure events. To distinguish it with clients’ real behavior time in physical world, we call it virtual arrival/departure time.

To achieve this, we have introduced distributed mechanism in the process of control. When appointed a specific behavior scenario, orchestrator will send relevant parameters to testing clients to let them locally compute their arrival/lifetime for testing. In case of a
client quits the experiment ahead of its virtual departure time unexpected, which we call *early-quitted* client, orchestrator will choose another viewer as a replacement and duplicate its status to the client.

However, *early-quitted* clients do have a negative influence on accuracy, since the process of replacement cannot be totally seamless. Thus we merely sort out stable viewers in physical world to perform distributed control for a specific behavior scenario to minimize the impact of replacement, which, on the other hand, has imposed restrictions on the scale of experiment in ShadowStream. Since in general cases stable clients are usually too small a group to be effectively utilized in production channel, which has been proved in Figure 1 (Wang&Liu, 2008). Figure 1(a) indicates that the majority of viewers just stay in channel for a quite short period in physical world, and if consider a client as stable when its lifetime exceed 40% of the observed session, it takes up only 5% to 18% of the whole viewers in different traces. Furthermore, Figure 1(b) explains the percentage of stable clients in a per-snapshot view in channel, and it is clear that there exist only 54% to 90% stable clients in a snapshot under the circumstances above.

In this paper, a novel distributed client lifecycle control is developed to get rid of restrictions caused by limited number of stable viewers in live-testing streaming networks. And dedicates to increasing real viewers’ utilization level in physical world and decreasing replacement times in the process of experiment.

The major challenge in the course of experiment control in live testing platform is about the real viewers. As a live testing system, ShadowStream is designed to orchestrate desired experimental scenarios from production viewers, without disturbing their quality of experience. Other than clients in a traditional testing platform, production viewers in live testing cannot be controlled. Furthermore, we are not allowed to interfere in their own behaviors (e.g., arrive, depart), or change their behaviors casually. That is to say, before sending a command to a client, guarantees should be made to ensure that the command which determines client’s act in the experiment will not be in conflict with its real behavior at some point in the future.

The core idea is to match the desired experimental scenario with viewers’ real behavior in physical world to scale up the experiment.

---

**Figure 1.** The general situation about lifetime distribution and ratio of stable nodes per snapshot in physical world. (a) CCDF of life time distribution in different traces. (b) Ratio of stable nodes to All Nodes per snapshot.
Related Content

Web Services Compositions Modelling and Choreographies Analysis

Selective Service Provenance in the VRESCo Runtime
Pattern-Based Design of an Asynchronous Invocation Framework for Web Services
www.igi-global.com/article/pattern-based-design-asynchronous-invocation/3044?camid=4v1a

Unleashing Artificial Intelligence Onto Big Data: A Review
www.igi-global.com/chapter/unleashing-artificial-intelligence-onto-big-data/217932?camid=4v1a