Evaluation of webRTC performance with various CPU scheduling, and Cache/DRAM partitioning strategies

Santosh Gondi, Siddhatha Biswas
{santoshg,sid.biswas}@ku.edu
The University of Kansas, USA

I. INTRODUCTION

Wide spread use of the Internet for day-to-day communications, and proliferation of smart phones and tablets is increasing the demand for real-time, video based communications. webRTC is a free, open project, consisting of standards and protocols for enabling plug-in-less, inter-operable, real-time communication between browsers. Video, unlike data, is delay sensitive. Hence, it poses additional challenges for real-time communications in terms of resource reservation and adaptation to varying conditions. Resource allocation, and adaptation to dynamic conditions, both, inside the network, and in the devices hosting the applications is necessary to maintain an acceptable level of services in the face of congestion and interference from other flows and applications. Also, multi-core platforms have become ubiquitous in all the hand held devices. Various techniques are used for performance improvement and isolation of applications running on these systems. In this paper, we propose to evaluate the impact of these mechanisms, such as, CPU scheduling, Cache and DRAM isolation methods, on real-time video chat using webRTC.

II. RELATED WORK

webRTC [3] project has two parts, webRTC, which provides HTML5/JavaScript API for real-time applications, and RTCWEB, which proposes standards for connection establishment, media exchange, and congestion control among other things. There are many works evaluating the performance of congestion control algorithm [1], [2]. We are proposing to evaluate webRTC performance while competing with other CPU and I/O bound tasks in the single and multi-core platforms. According to our knowledge there is no previous work in this space.

III. PROPOSED WORK

We will establish a evaluation set up similar to fig 1. A video chat will be launched between hosts (running Linux OS) using chromium browsers. We will use SPEC 2006 benchmark suite to simulate the interfering tasks to webRTC video chat application on one end. We intend to collect the delay, delay variation, and throughput metrics for the video chat session. We intend to compare the performance of webRTC with CFS (completely fair scheduler, current default Linux scheduler), and RT scheduler in Linux kernel for single-core platforms. We will make use of Linux cgroups [4] for resource isolation, and measure the performance variation of it on the video chat application. We also intend to evaluate the performance variation because of cache partitioning using Page Coloring [6], and DRAM Bank allocation [5] mechanism.

IV. DELIVERABLE

We show the graphs for performance metrics mentioned in above section for all the scenarios we are planning to evaluate the webRTC video chat performance. We believe that, by using performance and isolation techniques of scheduling and partitioning, we will achieve higher throughput and lower delay variation for real-time chat session, when competing with other CPU and I/O intensive applications.

Fig. 1: Experiment setup

REFERENCES

Proceedings of the 4th ACM European conference on Computer systems