Scaling on the Edge: A Benchmarking Suite For Human-in-the-Loop Applications

ROYAL INSTITUTE OF TECHNOLOGY

Manuel Olguín, Junjue Wang,
Mahadev Satyanarayanan, James Gross

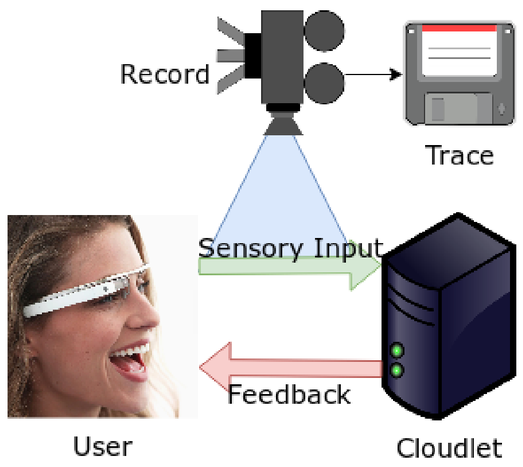
Carnegie Mellon University Computer Science Department

Abstract

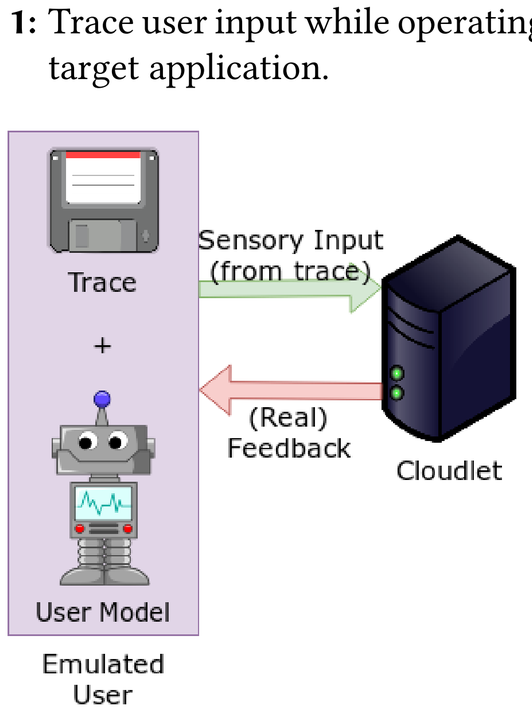
Benchmarking human-in-the-loop applications running on edge computing infrastructure is complex given their nature, which heavily depends on the actions taken by the human user. This limits reproducibility as well as feasibility of performance evaluations. We propose a methodology and present a benchmarking suite that can address these challenges. Our core idea rests on recording traces of these applications which are played out in a controlled fashion based on an underlying model of human behavior. The traces are exposed to the original backend compute process of the respective human-in-the-loop application, generating realistic feedback. This allows for an automated system which greatly simplifies benchmarking large scale scenarios.

Basic Idea

- Benchmarking human-in-the-loop applications is **hard** due to human users:
  - They are unpredictable.
  - They make scaling difficult (you need more of them!).
- What if we could cut out the user?



Step 1: Trace user input while operating the target application.



Step 2: Replace user with recorded trace plus a user model.

Design & Implementation

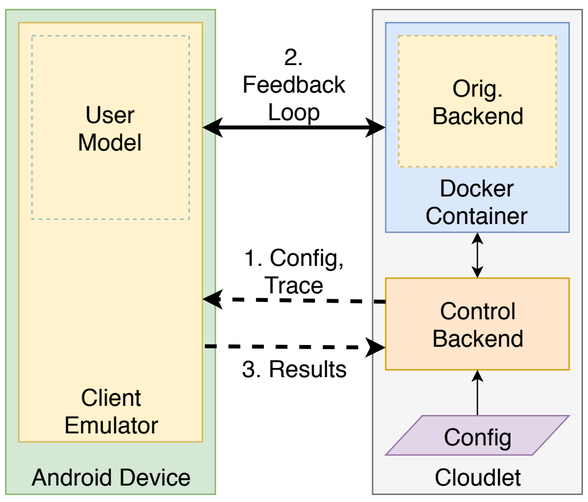


Figure 1: Suite Architecture

- The *control backend* controls the experiments and collects measurements from the application and the cloudlet itself.
- The *client emulators* play out a prerecorded sensory input trace over the network in a controlled fashion, while collecting client-side metrics.

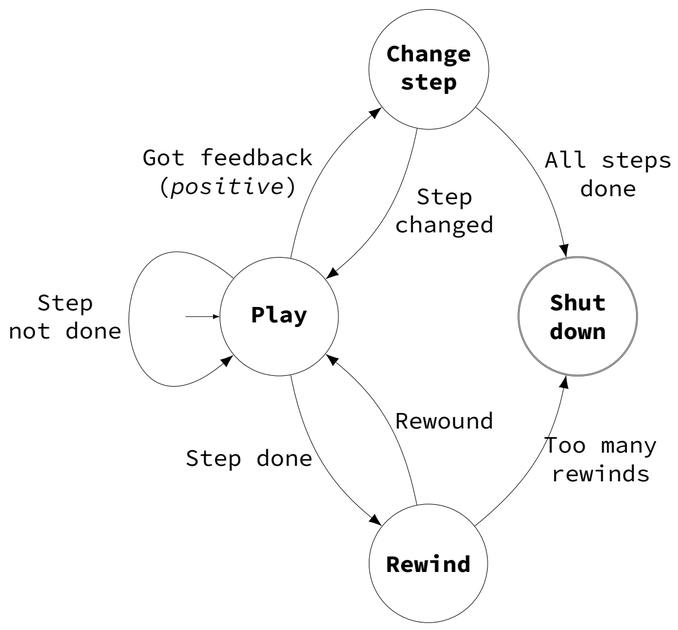
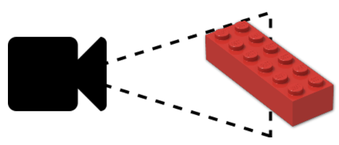


Figure 2: Simple user model used for the initial iteration of the suite.

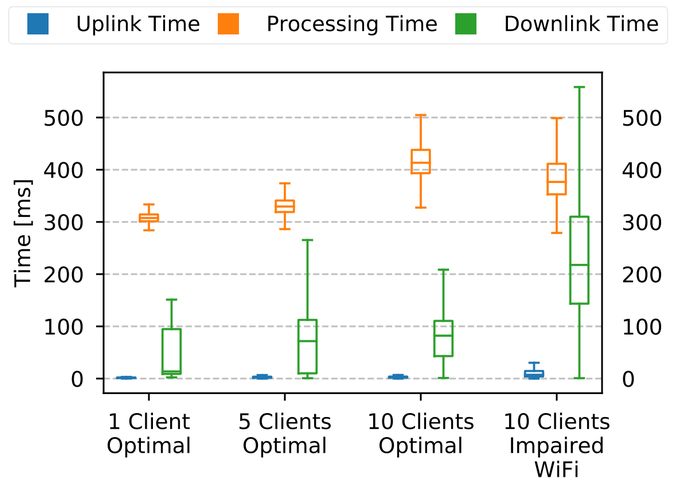
We implement a very simple user model in order to be able to react to feedback from the application and adapt our replay of the trace to the current system conditions.

Demo

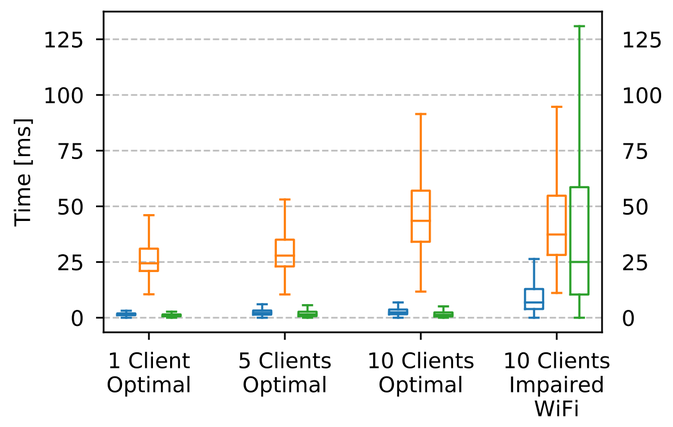
The client emulators in particular will be showcased in the DEMO. We will demonstrate the workings of the system on a simple LEGO task assistance application, and spectators will be able to follow the replay of the trace as well as the behavior of the user model through the client devices.



Some Example Results



a: Inputs that triggered feedback.



b: Inputs that did not trigger feedback.

Figure 3: Comparison of the latency distributions across system components for a series of scenarios.

These results could be useful for:

- System designers wishing to identify bottlenecks in their hardware stack. In this case, they could determine that the wireless link seems to be the weak point when the connection is not optimal.
- Application developers who want to find potential for optimization in their applications.

We will eventually make the benchmark suite available as Free and Open Source Software. The traces will also be made available under a Creative Commons License.

References

[1] K. Ha *et al.*, "Towards wearable cognitive assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*.

[2] Z. Chen *et al.*, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*.

[3] M. Satyanarayanan *et al.*, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009.

[4] T. Bittmann, "The edge will eat the cloud," *Gartner Research*, no. G00338633, 2017.

[5] K. Sasaki *et al.*, "Layered vehicle control system coordinated between multiple edge servers," in *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft)*.