Developing A Lightweight Client Module for Large-scale Simulator

Akihiro Suzuki(1310033)

School of Infomation Science,
Japan Advanced Institute of Science and Technology

February 12, 2015

Keywords: home network system, simulation, modeling, HEMS.

With the advent of technologies such as M2M and IoT, systems that are cooperating with services that utilise a vast array of inter-networking devices are fast becoming a reality. Among such systems, large scale systems that cover the expanse of a whole city and enable network communications within it exist. For such systems, before their actual deployment and operation it is imperative to test their behaviour using simulation of similar scale and accuracy. However, to simulate the target large scale system, a vast amount of computational resources is necessary. It is due to this fact that alternative approaches that use fewer computational resources are pursued.

In this research, a simulator that can simulate large scale smart community systems of the scale of tens of thousands of houses was developed. In a smart community, a vast amount of clients exist, and if a highly accurate simulation was to be performed, simulated models for each such client that capture its minute details have to be developed. However, this is a major problem because the required computational resources for such an approach would be too great. To tackle this problem, in previous research a small sized smart community of a few houses that have the same characteristics was used as a module, and by modelling this module higher scalability in the simulation was achieved. However, this approach utilised information that was obtained only by the network communications between clients and servers and thus it was difficult to make any projections regarding the communications that occur in the application layer communication protocols.

To reduce the amount of required computational resources for city-sized large scale simulations, the target of this research is to design and implement a lightweight client module. This lightweight client module will be used in the simulation of large scale systems that consist of tens of thousands of clients and few servers, and there is network

communication among the various network equipment. Furthermore to achieve a highly accurate simulation of such large scale systems, an accurate simulation of the application layer protocol communication is necessary.

The development of the lightweight client module used in large scale accurate simulations is described below. In order to reduce the the computational resources required, the lightweight client module only models the external communications of the target application. By removing the internal processing of the client of the target application, a reduction in the necessary computational resources was measured. In order to create the lightweight module for the simulated application, the state machine for the application is modelled and characteristics of the input and output data are applied to the state machine. By creating the client module from this model, it is now possible to simulate the communications at the application layer protocol for the simulated application. With this method, the accuracy of simulations that make use of such lightweight modules improves. When creating the state machine for the target application by only observing the external communications, the resulting client module provides only limited accuracy and furthermore it takes a long time to actually develop, something that is deemed unproductive. Thus, in order to create the state machine for the application, information by the application author or the source code of the program should be used. The characteristics of of the state machine that are used in the client module is information that captures the nature of the relationship between the input and output data of the model, network related information such as throughput, packet size distribution, and others. As the testbed for simulation the StarBED facilities were used.

The client module that was designed as described above was evaluated in terms of computational resource reduction. For evaluation purposes, the lightweight client module and the actual target application were profiled. Among the performance indexes used were the CPU utilisation, heap memory usage, network throughput and response time. A client module that successfully communicates with the server that also reduces computational resources was developed.

As future work, further reduction in the computational resources used is possible. The client module is written in Java which is a garbage collected language. We argue that it is possible to achieve lower memory usage by manually managing the memory allocations. Furthermore, the client module is designed in such a way as to easily be modified to accommodate different target applications. It is left as future work to repurpose the client module for other scenarios as necessary.