

# Prototype Implementation of Simulation Caching Framework for Multi-user Interaction

Jiachao Zhang, Yu Yamamoto, Shinji Fukuma, and Shin-ichiro Mori  
 Dept. of Information Science  
 University of Fukui  
 Bunkyo 3-9-1, Fukui, 910-8507, Japan  
 Email: {zhang, yyu, fukuma, moris}@sylph.fuis.u-fukui.ac.jp

**Abstract**—In order to realize remote steering and interactive collaboration over on-going simulation, we have developed a simulation framework based on simulation caching technique and extended this framework for the multi-user environment.

**Keywords:** Multiuser Interaction, Interactive Simulation, Ubiquitous HPC

## I. INTRODUCTION

In order to realize human-in-the-loop scientific computing in cloud like environment, we have to conquer the problem of network latency. For this purpose, we propose a simulation model which we referred to it as Simulation Caching[1]. The infrastructure for Simulation Caching is a sort of cooperative cloud where a high performance server (a remote server) somewhere on the cloud cooperates with a moderate scale server (a local server) to make an immediate and reasonable response to the operators interactive steering of the simulation. To hide the latency to the remote server, Simulation Caching lets the local server caches a part of the simulation from the remote server and performs the duplicated simulation concurrently with the remote server, while keeping the accuracy of the cached simulation by weakly cooperating with the original simulation running on the remote server.

For the ease of utilizing this technique, we have developed a simulation framework based on simulation caching technique and extended it for the multi-user environment. In this abstract, the next section describes Simulation Caching in much detail, followed by the discussion on our prototype implementation of Simulation Caching Framework for remote multi-user interaction.

## II. SIMULATION CACHING

The Simulation Caching 1) utilizes a moderate scale computing server (a local server), associate with local operation terminal, to perform a sort of simulation, low resolution simulation for example, to make an immediate and reasonable response to the operators intervention, and 2) keeps the accuracy of the cached simulation by weakly cooperating with the original simulation running on the remote server.

Figure 1 showcases an example of the Simulation Caching environment adopted to the interactive remote fluid simulation[2][3]. This simulator has a capability of steering the simulation by interactively changing the boundary conditions through the haptic device, while the device is also used as a

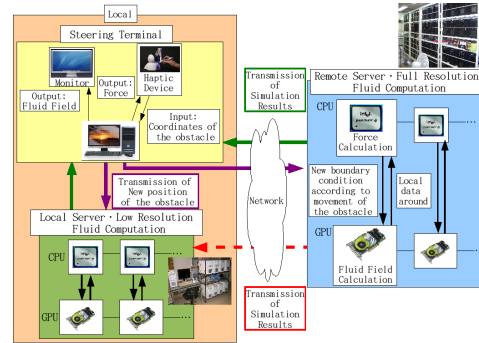


Fig. 1. An example of Simulation Caching : – An Application to the Interactive Remote Fluid Stirring Simulation –

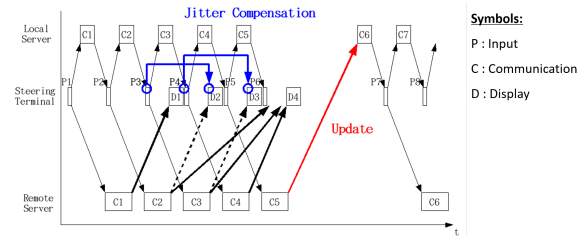


Fig. 2. Cooperation of the Local and Remote Servers in Simulation Caching.

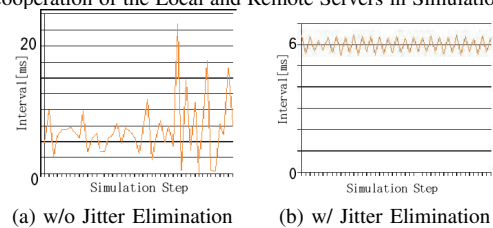


Fig. 3. Effect of the Jitter Elimination.

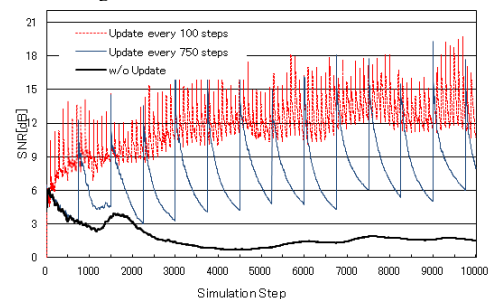


Fig. 4. Effect of Update Frequency.

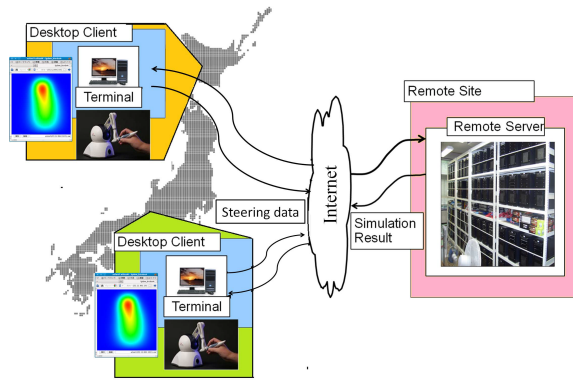


Fig. 5. Multi-User Simulation Caching.

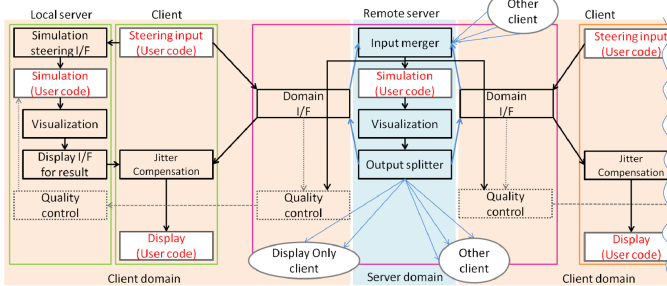


Fig. 6. Simulation Caching Framework.

force display device. Figure 2 represents how the local and remote servers interact with the steering terminal. In this figure, when unacceptable network delay due to the jitter happens, the results of the local simulation is displayed at the steering terminal instead of waiting for the response from remote server. Figure 3 shows the effect of this jitter elimination. The synchronization action, or the coherency control, between local and remote server is take place at time step 6 in this example to maintain the accuracy of the cached simulation. Figure 4 shows the quality of displayed image using the signal to noise ratio (SNR) assuming the result of the remote server as the reference signal and the difference between the results in local and remote servers as the noise. From this figure, we could confirm that if the frequency of the coherence action is higher, the SNR also get well. It is also confirmed that though the SNR gradually get worse for a short period, it is recovered at the next coherence action.

However, the frequent consistency action may reduce the simulation throughput, so this frequency should be adjusted by the user considering the tradeoff between simulation throughput, latency tolerance and simulation quality in local servers.

### III. SIMULATION CACHING FRAMEWORK FOR MULTI-USER INTERACTION

In order to provide Simulation Caching Environment to simulation users, we have developed a Simulation Caching API with the functionality of both real-time sharing and interactive steering of the simulation among distanced users. The Figure 5 and 6 show the concepts of distanced simulation

sharing and the overview of the prototype implementation of Simulation Caching Framework. With this framework, simulation programmers may concentrate on the programming of their own simulation provided that the steering input function and the visualization of simulation result should be separated from the simulation core.

The key components of this framework are listed below:

**Quality control** : assures 1) the quality of communication between distanced users and 2) the weak consistency of local simulation performed locally at user sites.

**Input Merger** : merges the steering inputs from users and feeds them to the simulation program. Merged inputs are also send to the local servers to weakly maintain the input consistency.

**Output Splitter** : appropriately splits the simulation result to uses. In order to reduce the network traffic data compression is also take place here.

**Domain Interface** : which provides interface between the simulation caching domains of each user and the simulation core on remote server, also maintains the security between domains though it is not implemented in the prototype implementation.

### IV. CONCLUSIONS

In this abstract, we showed our work-in-progress researches toward ubiquitous and interactive supercomputing. The Simulation Caching realizes interactive remote steering of simulation by hiding the network latency while the integrated simulation caching framework realizes remote multi-user collaboration over the on-going simulation keeping weak consistency among the remote user.

Currently, we have been improving our framework so that it could deal with some other collaboration scenarios, for example, the scenario where the local server performs limited-regional finer-grid simulation. We are also developing the Web-based monitoring and steering interface to allow the sharing of the simulation among multiple remote users.

### V. ACKNOWLEDGEMENTS

This research is supported in part by JSPS KAKENHI Grant Number 25280043.

### REFERENCES

- [1] K. Hashimoto, S. Tezuka, and S. Mori. Simulation caching and its application to remote interactive fluid simulation. *IPSPJ Trans. on Advanced Computing Systems*, 5(4):76–86, 4 2012.
- [2] R. Henmi, Y. Nishimura, H. Suzuki, S. Fukuma, S. Mori, A. Yamaguchi, and S. Tomita. Prototype implementation of a gpu-based interactive coupled fluid-structure simulation. In *Proc. of Int'l Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2012)*, pages 721–725, September 2012.
- [3] J. Zhang, S. Yuasa, S. Fukuma, and S. Mori. A real-time gpu-based coupled fluid-structure simulation with haptic interaction. In *Proc. of Int'l Conf. on Computer and Information Science (ICIS 2016)*, pages 1–6, June 2016.