# Web-based Ubiquitous Simulation Steering based on Simulation Caching

Shinpei Masui Naoki Shibata Fumitaka Arakawa Shinji Fukuma* Shin-ichiro Mori†
Bunkyo 3-9-1, Fukui, 910-8507, Japan
University of Fukui

## Abstract

We showed our work-in-progress researches toward ubiquitous and interactive supercomputing. In the proposed system, the Simulation Caching technique is used to realize interactive remote steering of simulation by hiding the network latency, while the Web-based HCI provides the support for the ubiquitous and multi-user interaction with the ongoing simulation.

**Index Terms:** H.3.5 [Information Storage and Retrieval]: Online Information Services—Data Sharing; C.3 [SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS]: Real-time and embedded systems; I.3.2 [Computer Graphics]: Graphics Systems—Distributed/Network Graphics

## 1 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 operator's 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.

Once the remote steering of the simulation becomes possible, the next requirement toward interactive supercomputing would be the ubiquitous and simultaneous multi-user interaction to the remote simulation. To realize such a simulation environment, we have been developing a web-based HCI using latest dynamic web technologies, like WebGL[2] and WebSocket[3].

In this abstract, the next section describes Simulation Caching in much detail, followed by the discussion on the Web-based HCI in section 3.

## 2 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 operator's intervention, and 2) keeps the accuracy of the cached simulation by weakly cooperating with the original simulation running on the remote server.

*e-mail:fukuma@u-fukui.ac.jp
†e-mail:moris@u-fukui.ac.jp

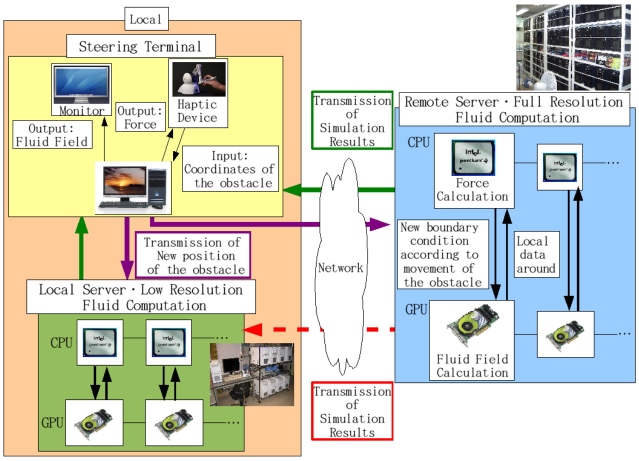


Figure 1: An example of Simulation Caching : – An Application to the Interactive Remote Fluid Stirring Simulation –

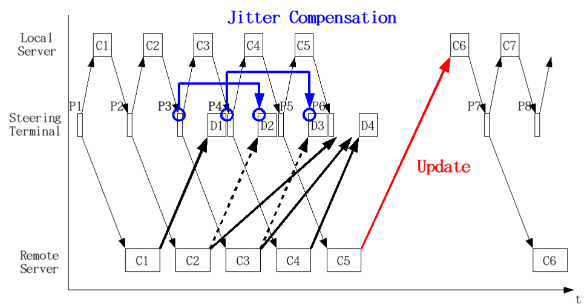


Figure 2: Cooperation of the Local and Remote Servers in Simulation Caching.

Figure 1 showcases an example of the Simulation Caching environment adopted to the interactive remote fluid simulation[4]. 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 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 the remote server.

## 3 Web-based HCI for Interactive Simulation Steering

Once the interactive simulation steering framework is provided, the next step toward interactive supercomputing should be the ubiquitous steering of the simulation, we believe. In figure 1, we assume the desktop environment as local steering terminal, however, we should consider various environments, like android tablet and smart

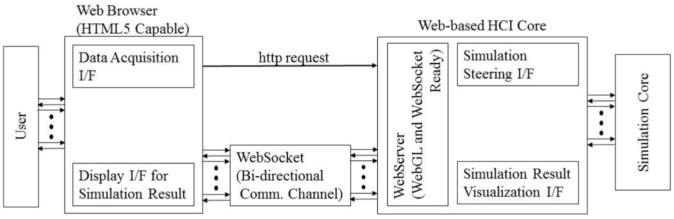


Figure 3: Overview of Web-based HCI for Simulation Steering

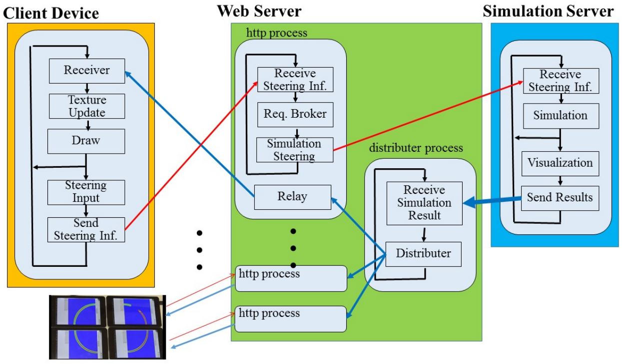


Figure 4: Detail of Web-based HCI with multi-user extension

phone, to realize ubiquitous simulation steering. As such a HCI for interactive simulation steering, the HCI should have the capability to visualize the stream-lined simulation results and to interact on the visualized result to steer the simulation. For this purpose, we utilized the WebGL technology[2] assuming recent media-rich mobile terminal as well as desktop and notebook environments. Through the event driven interface of WebGL, mouse and/or touch event to the representing simulation result on the terminal device can be captured as the next steering information to the simulation server. The captured steering information should be sent back to the simulation core to change the simulation parameters, such as the boundary condition, of the ongoing simulation on the simulation core. To realize such bi-directional communication between steering terminal and simulation core, we utilized the WebSocket[3] in our Web-based HCI. Furthermore, the multi-client capability of pywebsocket module for Apache server[5] makes our Web-based HCI easier to extend for a multi-client environment. Once a client sends an http request to the Web-based HCI site, a bi-directional virtual channel between the client and the simulation core is established and the streaming of the simulation results starts on this channel. Through this Web-based HCI site, multiple user can share the ongoing simulation results. Occasional user interaction on a client device is captured by WebGL interface and is sent back to the simulation core through the same channel. Figure 3 shows the overview of the system.

Figure 4 shows a little bit detailed implementation with multi-user extension. With this extension, multiple user can share the ongoing simulation results. Occasional user interaction on a client device is captured by WebGL interface and is sent back to the simulation core through the same channel. Then, the result of this interaction will be reflected to all the client device sharing the simulation.

Based on this web-based HCI, we made further extension to provide scalability of pixel counts and frame rate by aggregating multiple mobile devices to form one virtual HCI device as showed in Figure 5. In this configuration, each mobile device works as a part of a large virtual display, maintaining the similar frame rate by gathering the network bandwidth of these four devices.

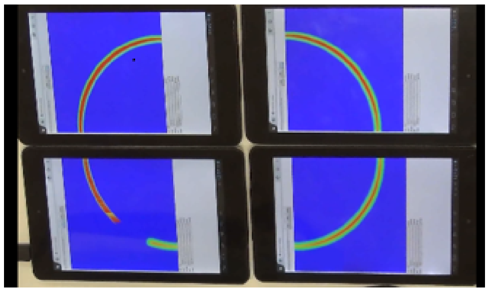

Figure 5: Tiled Touchpad

As the feature of the bi-directional interaction, using the touch interface, it could change visualization parameter immediately from client side like SAGE2[6] and further it could directly steer the simulation parameter to realize real-time interaction with the ongoing simulation collaboratively with multiple clients.

## 4 Conclusions

In this poster, 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 Web-based HCI provides the support for the ubiquitous and multi-user interaction to ongoing simulation. Since these two works are not integrated yet, we are currently going to integrate them to construct the interactive supercomputing framework. Coherency control in case of simultaneous steering in multi-user environment is also remained as the future work.

## Acknowledgement

This research is supported in part by JSPS KAKENHI Grant Number 25280043 and 25580042.

## References

[1] Kensuke Hashimoto, Shunsaku Tezuka, Shin-ichiro Mori : Simulation Caching and Its Application to Remote Interactive Fluid Simulation, IPSJ Trans. on Advanced Computing Systems, Vol.5, No.4. pp.76–86(2012)(in Japanese).

[2] WebGL Specification Ver.1.0 Feb. 2011 (http://www.khronos.org/registry/webgl/specs/latest/)

[3] The WebSocket API Editor's Draft 8 Feb. 2012 (http://dev.we.org/html5/websockets)

[4] Ryota Henmi, Yusuke Nishimura, Hiroaki Suzuki, Shinji Fukuma, Shin-ichiro Mori, Akinori Yamaguchi, Shinji Tomita : Prototype Implementation of a GPU-based Interactive Coupled Fluid-Structure Simulation, Proc. of Int'l Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2012), pp.721–725(2012).

[5] pywebsocket Project Home, (http://code.google.com/p/pywebsocket)

[6] T.Marrinan, J.Aurisano, A.Nishimoto, K.Bharadwaj, V.Mateevitsi, L. Renambot, L.Long, A.Johnson, J.Leigh : SAGE2:A New Aooriach for Data Intensive Collaboration Using Scalable Resolution Shared Displays, Proc. of Int'l Conf. on Collaborative Computing, Networking, Applications and Worksharing, pp.177-186,2014.