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

**Motivation**

Collaborative and Interactive Supercomputing

Human in the Loop Scientific Simulation

Scientific Applications

Grid/High-Performance Computing

Biomedical Engineering Medical Applications

Efficient rendering using multiple mobile devices

**Simulation Caching**

The Simulation Caching utilizes a remote computing server (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 intervention, and 2)keeps the accuracy of the cached simulation by weakly cooperating with the original simulation running on the remote server.

**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 following figures shows the concepts of 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 code.

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 users. 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 code on remote server, also maintains the security between domains though it is not implemented in the prototype implementation.

**Results**

Before consistency, local server A or B don't know what happened by others. After consistency, local server A knows B's latest simulation or input, same to B.

**Conclusion**

In this paper, we showed our work-in-progress research 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 leaving weak consistency among the remote users.

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-national 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.

**Further Extension**: Web-based monitoring and steering through WebGL and WebSockets make it possible!!