Seamless Object Migration for Massively Multiplayer Online Games

CS237 Distributed Systems Middleware Course Project
Te-Yu Chen (96922083)

Motivation

Massively multiplayer online game (MMOG) is one of the most popular genres in the gaming industry. Games like World of Warcraft allows thousands of players to enjoy the gaming experience together in the same game world. However, no existing server can handle such a workload alone. Many load balance mechanisms have been proposed and used. The earliest approach is to divide the game into several parallel realms. Each realm is managed by a game server. A player is first assigned to a game server which is in charge of all computation relative to this player throughout the game. The player will not migrate to other servers and is not allowed to interact with players on other servers. To overcome this limitation, recent load balancing approaches do not have this type of fix assignment of players to game servers. Instead, the game world is divided into several areas, each is managed by a game server. A player can move from one area to another as he or she like. When it happens, the game objects related to this specific player need to be migrated, and the hand-over process usually takes a long time. Beside of the waiting time, one still cannot interact directly with players not located in the same area. Therefore, we believe there is still room for improvement.

Goals and Objectives

This project aims to provide a middleware that helps game developer build a truly seamless game world running on multiple game servers, in which the migrations of game objects take place without the players’ notice, and all players can interact with each other regardless of which server they are currently on. The ultimate goal of this project is to find an open source MMOG and use the middleware to make it run on multiple servers.

Detailed Plans

We plan to divide the game world into several areas, each area is managed by a virtual server, just like the approach we have mentioned before. However, we want to add two main new features: **seamless migration** and **interaction across different servers**. The technologies we need may includes:

- **Serialization** Serialization is essential for object migration. We will need to migrate all game objects related to a player, i.e. the game character, if the player moves across the boundary of the servers.

- **Object replication** The replication of objects has two main purposes. The first purpose is to enable fast migration. By creating a replica when an object is near the boundary, we can minimize the cost when the object is actually moving across the boundary. Second, by creating an additional shadow object in the near by servers, we can enable objects on different servers to interact with each other, as long as they are both near the boundary.

Group Member

Te-Yu Chen ID: 59682772
Schedule

- **Week 3** Project proposal.
- **Week 4-6** Developing the middleware.
- **Week 7-8** Modification of an open source MMOG, performance analysis.
- **Week 9-10** Project report.