

Análise da Arquitetura de Servidores de Jogos Multijogador Massivo Distribuído

Mateus Barbosa da Silva
Francisco Ivan

Tecnologia em Redes de Computadores
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Canindé – CE – Brasil

mateusbds222@email.com.br, ivan.oliveira@ifce.edu.br

Abstract. *O abstract apresenta o texto do resumo em inglês. Quando o autor não tem domínio da Língua Inglesa, recomenda-se que este peça para alguém corrigir o abstract do trabalho.*

Resumo. *O resumo consiste na apresentação clara e concisa dos pontos relevantes do trabalho e pode influenciar ou estimular a leitura do texto na íntegra. É redigido pelo autor em um único parágrafo, com 500 caracteres (sem espaço) no máximo. O texto, composto por uma sequência de frases completas e não por uma enumeração de tópicos, deve ressaltar o objetivo, o resultado e as conclusões do trabalho, bem como a metodologia utilizada para sua elaboração. Deve-se utilizar frases afirmativas, evitar a repetição de frases inteiras do original e respeitar a ordem em que as idéias são apresentadas no trabalho.*

1 Introdução

Jogos multijogadores massivos (MMOGs) tem tomado um grande espaço nos últimos tempos, pois podem ser jogados com vários outros jogadores através da internet. Podemos tomar como exemplo famosos jogo da atualidade, como League of Legends, World of Warcraft, Overwatch, entre outros.

Geralmente, cada jogador controla seu “personagem” no mundo do jogo, que executa ações determinadas por ele. Cada ação feita pelo jogador é processada pelo servidor, depois a ação é computada para todos os outros jogadores.

O problema nesse tipo de abordagem (cliente-servidor) é que os clientes podem facilmente saturar o servidor do jogo. Geralmente, o que se faz é uma infraestrutura com uma conexão extremamente alta. É daí que surge o maior custo dos servidores de jogos.

2 Fundamentação Teórica

A arquitetura mais usada em servidores de jogos é a cliente-servidor, figura 1, que consiste em um servidor que atende todos seus clientes, seja através da internet ou diretamente.

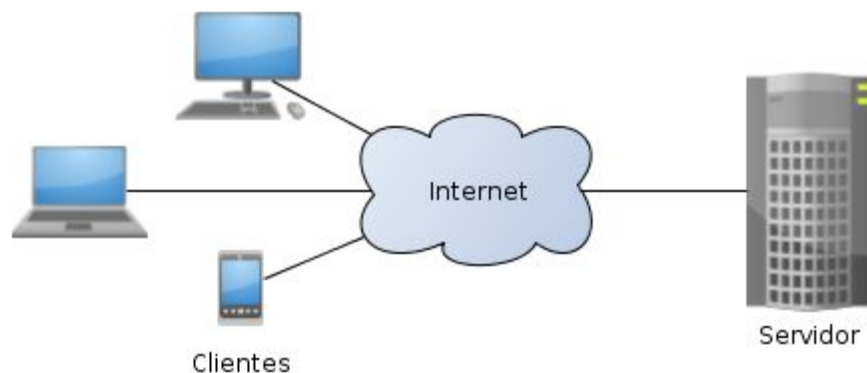


Figura 1 - Arquitetura cliente-servidor

Essa arquitetura é muito usada não só em jogos, mas em quase todo tipo de infraestrutura web por conta da sua simplicidade. As principais vantagens e desvantagens dessa arquitetura consiste em:

- Internet de um cliente não afeta outro cliente, pois a responsabilidade fica completamente nos computadores do servidor;
- Maior facilidade de manutenção, troca e reparo do servidor;

Todos os dados ficam nos servidores, onde a segurança geralmente é maior que nos clientes (cheating também é mais difícil por conta disso);

- Menor latência.

Algumas desvantagens podem ser vistas a seguir:

- Maior quantidade de clientes fazem mais requerimentos, o que pode fazer com que o servidor fique lento por não conseguir responder a todas as solicitações;
- Dados ficam todos em um servidor centralizado, caso o servidor venha a falhar, os clientes não conseguirão acessar os dados;
- Clientes não podem fornecer serviços para outros clientes;
- Se o servidor tiver problemas, todo mundo terá.

Outra arquitetura também muito conhecida é a ponto-a-ponto, ou arquitetura p2p, figura 2. Essa arquitetura consiste na descentralização da rede, onde cada nó é responsável tanto por responder a requisições como realizá-las, ou seja, cada ponto opera como cliente e servidor.

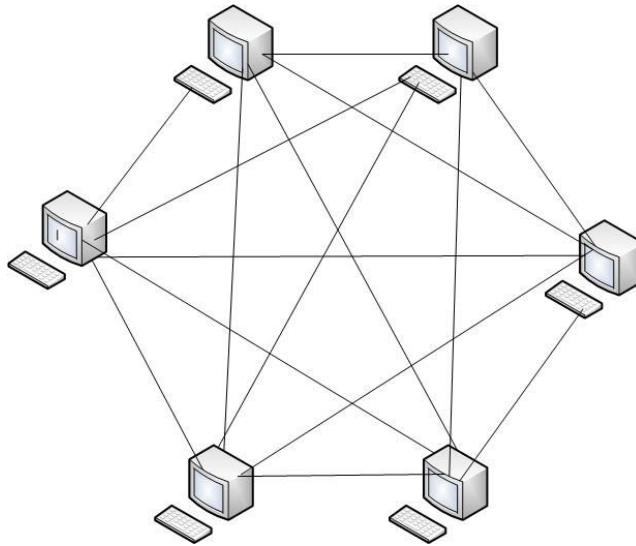


Figura 2 - Arquitetura ponto-a-ponto

Vantagens da arquitetura p2p:

- Não existe a necessidade de um servidor central;
- Nunca irá acontecer do servidor ter problemas e ninguém poder jogar;
- Bom para distribuição de dados;
- Quanto mais nós, melhor o desempenho da rede.

Desvantagens da arquitetura p2p:

- Implementação difícil;
- Muito difícil impedir o cheating;
- Difícil de conseguir segurança;
- Conexão de um cliente pode influenciar na jogabilidade de outros clientes;
- Geralmente possui uma latência alta.

Uma das soluções que é apresentada em alguns artigos usados como referência, é o uso da ferramenta FreeMMG. O FreeMMG é um modelo híbrido cliente-servidor e ponto-a-ponto ou de simulação híbrida para jogos MMO resistente a cheats. De acordo com a pesquisa feita sobre o FreeMMG, foi usado um protótipo de um jogo implementado num servidor FreeMMG que pode rodar centenas de clientes usando uma largura de banda pequena comparada com a alternativa cliente-servidor.

Em seu “top-level” o FreeMMG é baseado na arquitetura cliente-servidor só que com algumas diferenças, pois o servidor é responsável apenas por:

- Autenticação dos jogadores;
- Acompanhamento de sessão do jogador;
- Gerenciamento das áreas de interesse do jogador;
- Armazenamento das cópias de backup do mundo virtual.

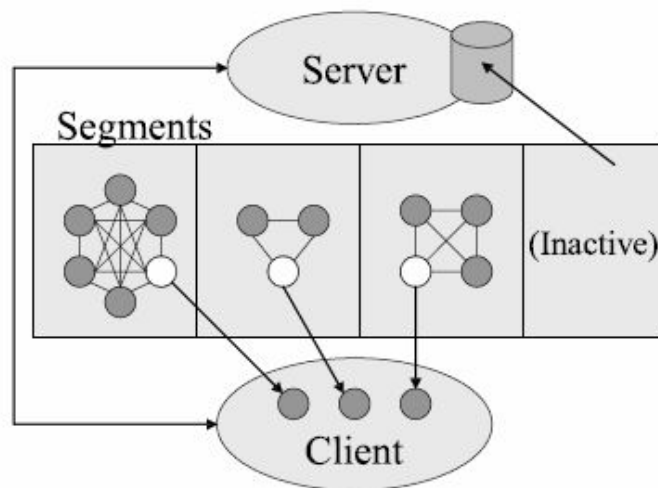


Figura 3 - Arquitetura FreeMMG

De acordo com a figura 3, os clientes interagem e repassam informações não tão cruciais com os segmentos p2p, enquanto o servidor é responsável por pegar todos os dados dos segmentos inativos para realocar com outros segmentos que entrarem na rede p2p.

Um cliente pode falhar por N motivos, tais como falha de sincronia como resultado de um bug, ou podem até mesmo perder a conexão com a internet. A figura 4 ilustra como o servidor age diante de uma falha com os clientes p2p.

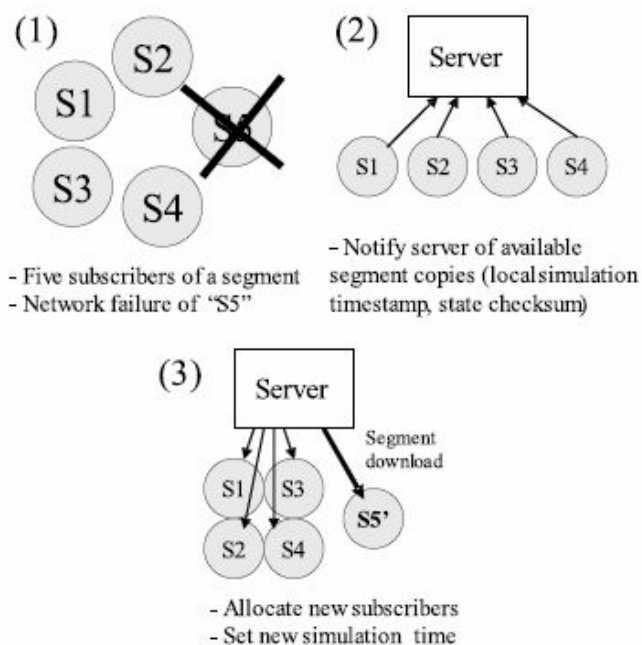


Figura 4 - Falha em um cliente p2p

Quando um cliente falhar, o servidor será notificado, pegará todos os dados daquele cliente no seu cache, distribuirá todos esses dados e alocará um novo cliente para ficar no lugar daquele que falhou.

Outro grande aspecto dessa arquitetura, é sua possibilidade de anti-cheating. Uma das principais ferramentas anti-cheating do FreeMMG é a Replicated Simulation, que consiste no servidor mandar cópias do mundo virtual para todos seus clientes e caso o usuário edite algo nesse mundo, ele estará editando apenas para si mesmo. Existem até outras formas de cheating, mas seria visível para os outros jogadores pois o cheater estaria fora de sincronia.

Existem outras ferramentas que emulam uma VPN e usam o modelo P2P na sua rede:

- Freelan;
- Wippien;
- LAN2LAN;
- Hamachi.

3 Considerações Finais

A velocidade de banda nas nossas casas vem crescendo muito, e isso pode tornar mais fácil o uso da arquitetura P2P ao invés do cliente-servidor, ou até mesmo o modelo híbrido do FreeMMG.

4 Referências

TANENBAUM, Andrew; VAN STEEN, Maarten. **Sistemas Distribuídos: princípios e paradigmas**. PEARSON, 2011. 402p.

POSEY, Brian. **Understanding the differences between client/server and peer-to-peer networks**. Disponível em: <<http://www.techrepublic.com/article/understanding-the-differences-between-client-server-and-peer-to-peer-networks/>>. Acesso em: 20mar. 2017.

Gamedev Stackexchange. **Limitations of p2p multiplayer games vs client-server**. Disponível em: <<https://gamedev.stackexchange.com/questions/67738/limitations-of-p2p-multiplayer-games-vs-client-server>>. Acesso em: 20mar. 2017.

FreeMMG: A Scalable and Cheat-Resistant Distribution Model for Internet Games

Fábio Reis Cecin, Rodrigo Real, Rafael de Oliveira Jannone, Cláudio Fernando Resin Geyer. FreeMMG: A Scalable and Cheat-Resistant Distribution Model for Internet Games. Universidade Federal do Rio Grande do Sul Porto Alegre, RS, Brazil, 2004. 8p.