

Comparação de um jogo RPG em ambiente Distribuído com relação a um ambiente Cliente-Servidor

Leonildo José de Melo de Azevedo¹, Evanise Araujo Caldas¹, Andres Jessé Porfírio², Hermano Pereira²

¹Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste (UNICENTRO) Caixa Postal 85.040-080 – Guarapuava – PR – Brasil

²Departamento de Tecnologia em Sistemas para Internet – Universidade Tecnológica Federal do Paraná (UTFPR) Caixa Postal 85.053-510 – Guarapuava – PR – Brasil

{leonildo.azevedo, evanise_caldas}@hotmail.com, andresjesse@yahoo.com.br, pereira@hermano.com.br

Abstract. *Currently the area of digital games has grown massively, using client-server network environments. However, this environment has a high cost and limited scalability. This makes the distributed architectures to become an alternative to the use of multiplayer games. In this study, a client-server and a distributed environment were implemented and compared in terms of bandwidth in a RPG (Role Playing Game). In the results, the distributed environment had a better performance than the client-server for the amount of connected clients, which were up to five.*

Resumo. *Atualmente a área de jogos digitais tem crescido massivamente, utilizando ambientes de rede cliente-servidor. Contudo, esse ambiente tem custo alto e escalabilidade limitada, e isso faz com que arquiteturas distribuídas se tornem uma alternativa para o uso de jogos multijogadores. Neste estudo, um ambiente cliente-servidor e outro distribuído foram implementados e comparados em relação ao consumo de largura de banda em um jogo de RPG (Role Playing Games). Nos resultados obtidos, o ambiente distribuído obteve melhor desempenho que o cliente-servidor para a quantidade de clientes conectados, que foram até cinco.*

1. Introdução

Diariamente, milhões de pessoas utilizam a *Internet* para se conectarem a diferentes gêneros de jogos de computadores, caracterizando uma classe denominada *Massively Multiplayer Online Game* (MMOG), que é uma derivação dos jogos *multiplayer* ou multijogadores [Novak 2010].

A grande maioria desses jogos utiliza um ambiente de rede cliente-servidor, contudo, esse ambiente possui certas limitações com relação à escalabilidade e consistência [Shen 2011]. Escalabilidade corresponde a quantidade de jogadores possíveis de conectar, e consistência refere-se a perda de acesso às informações [Shen 2011]. Tais limitações fazem com que ambientes distribuídos se tornem uma alternativa para o campo de jogos eletrônicos.

Os ambientes de rede cliente-servidor, têm preferência na área de jogos eletrônicos devido a sua simplicidade e organização na implementação [Hallberg 2003]. Outro motivo pelo qual os ambientes de rede cliente-servidor são mais utilizados é devido ao fato de que os primeiros jogos *multiplayer* optaram por esse tipo de ambiente [Novak 2010]. Além disso, o controle é centralizado, ou seja, o controle fica restrito

somente ao administrador do servidor, facilitando assim a comercialização e obtenção de lucro com o jogo [Novak 2010; Cecin 2005].

Por outro lado, ambientes de rede distribuídos para jogos eletrônicos são mais complexos em sua implementação. Por não terem um controle centralizado, necessitam de um tratamento mais elaborado na distribuição das informações e na forma com que trafegam na rede [Cecin 2005; Kozovits 2003; Shen 2011].

Para Shen (2011), um motivo que tem influência sobre a quantidade de dados trafegados pela rede e sua escalabilidade, é modelo de interação do jogo. Tais modelos se dividem em diversos gêneros, entre eles: RPG (*Role Playing Games*) jogos de interpretação de papéis, onde o jogador controla um ou vários personagens virtuais; FPS (*First Person Shooter*) jogos de tiro em primeira pessoa, no qual visualiza apenas do ponto de vista do protagonista, com o jogador e personagem sendo o mesmo observador; RTS (*Real Time Strategy*) jogos de estratégia em tempo real, em geral, baseiam-se na construção de um “mundo”, com o objetivo destruir ou conquistar o “mundo” dos outros jogadores. Dentre esses, os jogos de gênero RTS são os que têm um maior consumo de largura de banda [Cecin 2005].

Neste trabalho os ambientes de rede distribuídos são tratados no contexto *peer-to-peer*, onde cada computador é responsável por fornecer recursos e utilizar os de outros computadores. Com isso, propõem-se a implementação dos dois conceitos básicos de cada ambiente de rede para um jogo RPG.

2. Jogos Multiplayer

Segundo Novak (2010), os jogos *multiplayers* surgiram décadas antes da comercialização via Internet. Tudo começou com o PLATO (*Programmed Logic for Automatic Teaching Operations*) introduzido em 1961 na Universidade de Illinois, e era utilizado inicialmente em pesquisas na área de educação em computadores e mais tarde acabou se tornando uma rede de jogos *multiplayer*. A partir daí, o crescimento se expandiu, dando origem aos mais diversos tipos de jogos. O primeiro jogo *multiplayer* foi o *Space War*. Na década de 70, o PLATO ofereceu jogos com *Avatar* (inspirados em *Dungeons & Dragons*), dando início então aos jogos em RPG *multiplayers*, e junto a eles a Internet.

2.1. Dimensão de jogadores em Jogos Multiplayer

Em jogos *multiplayers*, há duas dimensões de jogadores, divididas em MOGs e MMOGs [Novak 2010];[Cecin 2005]. MMOGs são classes que derivam dos MOGs, apresentando duas características particulares. A primeira é que podem envolver uma quantidade grande de jogadores (chegando a casa dos milhares) interagindo simultaneamente em tempo real. A segunda é que MMOGs apresentam uma simulação de estado persistente em permanente mutação, ao contrário dos MOGs que tem simulações de curta duração e, suportam um número relativamente pequeno de participantes, que em sua maioria não passam de cem jogadores [Novak 2010; Cecin 2005; Kozovits 2003; Shen 2011; Shen et al. 2012].

2.2. Modo de Comunicação em Jogos Multiplayer

O tipo de comunicação mais comum é a cliente-servidor, na qual cada *host* (computador conectado em rede) conectado ao servidor é um cliente, que troca mensagens apenas com o servidor, o qual é responsável por armazenar ou distribuir as informações do jogo para os demais clientes [Hallberg 2003].

A troca dessas mensagens geralmente é feita utilizando os protocolos TCP (*Transmission Control Protocol*) ou UDP (*User Datagram Protocol*). O protocolo TCP é um protocolo com conexão e garantia de entrega de pacotes. Já o protocolo UDP é um protocolo construído sobre *datagramas*, que por ser um protocolo sem conexão e sem garantia de entrega de mensagens, evita sobrecargas de processamento associadas ao protocolo TCP [Coulouris *et al.* 2007]. Entretanto, independente da conexão utilizar UDP ou TCP, a troca de mensagens feita de forma centralizada pode gerar uma escalabilidade baixa e alta latência na conexão em jogos MMOG [Cecin 2005; Shen 2011; Shen *et al.* 2012].

Existe também o modo de comunicação distribuído. Segundo Tanenbaum (2007), “um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários”. No contexto de jogos, uma comunicação distribuída (com exceção dos servidores distribuídos) é usualmente entendida como o modelo *Peer-to-Peer* (P2P) [Novak 2010]. Kozovits (2003), em seu trabalho, faz uma análise considerável em jogos com ambientes distribuídos, concluindo que um mal gerenciamento da rede pode resultar em um esforço de $O(N^2)$ para a distribuição dessas informações, precipitando um congestionamento na rede, e causando um problema de escalabilidade. Isso pode ocorrer em sistemas distribuídos *unicast*, onde todos os computadores estão conectados entre si e disparam mensagem para todos um a um.

Devido ao problema de escalabilidade dos sistemas *unicast* houve a necessidade do uso de *broadcast* de mensagens. Em vez de ocorrer envio de mensagens para cada participante, apenas uma única mensagem *broadcast* será enviada a cada atualização, diminuindo o esforço para $O(N)$. Outro problema considerável é que não é possível utilizar *broadcast* na Internet, para solucionar isso ambientes virtuais são criados, usando então a comunicação *multicast* (*broadcast* na camada de aplicação) [Kozovits 2003; Cecin 2005].

Outras técnicas poder ser aplicadas para contornar esse problema, uma muito interessante e aplicável é separar o jogo por área de interesse (*Area of Interest – AOI*), onde o jogador manda atualizações somente dos objetos alterados em determinada região e não do cenário inteiro [Shen 2011]. Até o presente momento, não foram encontrados jogos na literatura que possuíssem comunicação totalmente distribuída. Há os que tem servidores distribuídos como o *EverQuest*.

3. Materiais e Métodos

Para a realização deste trabalho foram utilizadas algumas ferramentas de apoio. As subseções a seguir descrevem as ferramentas utilizadas, e a metodologia aplicada para a realização deste trabalho.

3.1. Ferramentas

Para a implementação de um ambiente de rede cliente-servidor e outro distribuído em um jogo RPG, foi utilizado o IrrRPG Builder uma ferramenta para criação de jogos RPG, a ENet uma biblioteca para auxiliar a comunicação em rede e o TCPCDump uma ferramenta para analisar o estado da rede (trafego de pacotes, destino de pacotes, entre outros).

3.1.1. IrrRPG Builder

O IrrRPG Builder é uma ferramenta de criação de jogos *open source*, com manipulação de objetos em 3D e é compilada em C++. Ela abrange uma grande massa de usuários e

pode ser utilizada para diversos fins [Porfirio and Hild 2012]. O gerenciamento das ações de iteração do jogador, com os diversos elementos presentes no cenário do jogo, é feita por *scripts* escritos em linguagem LUA, uma linguagem muito utilizada em jogos eletrônicos. A ferramenta apresenta uma interface simples, facilitando assim um desenvolvimento fácil e rápido de jogos RPG.

3.1.2. Biblioteca ENet

A ENet é uma biblioteca gratuita para C++, com objetivo de dar suporte a jogos. A biblioteca fornece uma camada de comunicação de redes simples, e trabalha com protocolo UDP, entretanto, pode funcionar com conexão e garantia de entrega de pacotes em ordem de envio. Por utilizar UDP a ENet fornece uma flexibilidade maior no envio dos pacotes, onde o usuário pode definir se o envio será com conexão, garantia de entrega e controle de fluxo, ou ainda se o envio será sem conexão, sem garantia de entrega e sem controle de fluxo.

3.1.3. Análise de tráfego com o TCPDump

TCPDump é uma ferramenta utilizada em linha de comando, contida na maioria dos sistemas UNIX para gerenciar o estado da rede. Ela mostra uma descrição do conteúdo dos pacotes trafegados na rede, permitindo-se utilizar de sinalizadores. Tais sinalizadores podem determinar em qual IP (*Internet Protocol*) será feita a análise do tráfego, assim como em qual porta e qual a rede (cabeadada ou sem fio).

3.2. Metodologia

O trabalho procedeu da seguinte maneira: 1. Implementação do ambiente cliente-servidor com o uso da biblioteca ENet em modo texto; 2. Criação do protótipo do jogo e integração do ambiente cliente-servidor; 3. Adaptação da versão cliente-servidor do protótipo do jogo para comunicação distribuída, e; 4. Elaboração e realização dos testes com o uso do TCPDump.

3.2.1. Implementação do ambiente cliente-servidor em modo texto

Primeiramente foi implementada uma comunicação cliente-servidor em modo texto e, para tal, foi utilizada a biblioteca ENet. A comunicação em modo texto tinha as funcionalidades de *chat* (bate papo), onde todos os clientes conectados ao servidor podem trocar mensagens entre si (com protocolo UDP) por intermédio do servidor.

Nesse ambiente, cada cliente conecta-se ao servidor, havendo troca de informações apenas entre cada cliente e o servidor, e quando o servidor recebe alguma mensagem ele distribui essa mensagem por *multicast*. A Figura 1 apresenta uma representação gráfica da comunicação cliente-servidor.

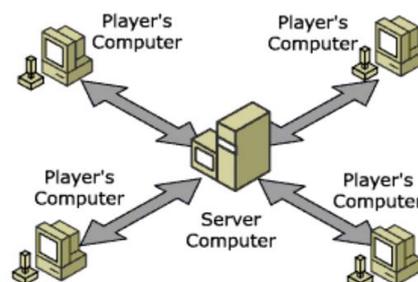


Figura 1. Ambiente cliente-servidor típico [Kozovits 2003].

O envio das informações foram feitos através do protocolo UDP, entretanto, foi tratada a garantia de entrega, envio com conexão e controle de fluxo. Tais tratamentos são feitos através de parâmetros da biblioteca Enet, mais especificamente, utilizando o parâmetro ENET_PACKET_FLAG_RELIABLE que define esses tratamentos. Com exceção dos pacotes, que foram definidos na criação antes de ser enviado, as conexões foram realizadas com os parâmetros *default* da biblioteca ENet.

3.2.2. Criação do protótipo do jogo e integração do ambiente cliente-servidor

Após feita a comunicação em modo texto, foi criado um protótipo de jogo virtual RPG com a ferramenta de criação de jogos IrrRPG Builder. O protótipo do jogo apresenta um cenário simples, resumindo-se a um RPG de caça ao tesouro.

Com o protótipo do jogo criado, foi realizado a integração da comunicação em rede com o jogo. Na comunicação em rede foi aplicada a técnica de Área de Interesse, onde o jogador manda atualizações somente da região em que está e não do cenário inteiro, neste caso a sua posição [Shen 2011]. Para se conectar ao servidor, cada cliente possui uma lista de possíveis endereços de servidores e, conecta-se primeiro endereço em que o servidor está disponível. Os pacotes enviados contem a posição do jogador ou mensagens correspondentes do *chat*, os pacotes que são trafegados no ambiente cliente-servidor têm o mesmo tamanho dos pacotes trafegados no ambiente distribuído.

O servidor continua em modo texto, entretanto, foi modificado para fazer a verificação de qual jogador encontrou antes o tesouro e se esse jogador requisitou a missão de caça ao tesouro, ou seja, se o jogador encontrar o tesouro sem ter requisitado a missão ele não ganha o jogo, contudo, se outro jogador requisitou a missão e encontrou o tesouro, no momento em que ele requisitar a missão será exibida uma mensagem “*YOU LOST!*” notificando-o de que perdeu.

3.2.3. Adaptação da versão cliente-servidor do protótipo do jogo para comunicação distribuída

No ambiente distribuído, a parte do servidor migrou para a parte de cliente. Nesse ambiente os jogadores não se conectavam mais a um único ponto (servidor), portanto, foi necessário um tratamento melhor na conexão e na distribuição das informações. Esse tratamento é descrito pelo Algoritmo 1.

Algoritmo 1: Conexão em ambiente distribuído

início

O jogador tem inicialmente uma lista de IPs
Jogador procura outros jogadores *onlines* na lista de IPs
Jogador conecta-se ao primeiro jogador disponível

repita

se algum Jogador se conectar ou recebeu notificação de novo jogador **então**

Conecta-se ao novo Jogador caso não tenha conectado
Faz *broadcast* avisando que tem um novo jogador

fim

até terminar a execução do jogo

fim

Após todos os jogadores conectados, a estrutura da comunicação entre eles se torna uma malha em redes de computadores, tendo conexão de todos para todos, como mostra a Figura 2.

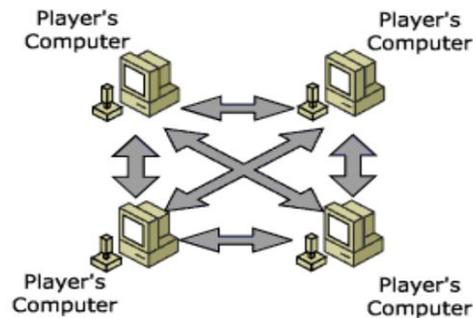


Figura 2. Ambiente *peer-to-peer* típico [Kozovits 2003].

3.2.4. Elaboração e realização dos testes com o uso do TCPDump

Na fase de testes, para analisar o estado da rede foi utilizado o comando “*tcpdump -ni <interface> -s <tamanho> udp port <porta> and host <ip>*”, os parâmetros do comando foram retirados da documentação do TCPDump, onde: *tcpdump*, executa a ferramenta; *-ni*, *n* exibe somente o IP, ou seja, abstrai o nome dos *host*, e *i* é utilizado pra definir onde será analisado o tráfego (rede cabeada ou rede não cabeada), neste contexto foi a cabeada; *-s*, utilizada para definir o tamanho da captura dos pacotes, sem esse parâmetro o TCPDump pode omitir o tamanho de alguns pacotes, para que não houvesse omissão, o tamanho foi definido como 1500; *udp*, para analisar apenas pacotes com protocolo UDP; *port*, filtra a análise pela porta definida; *and*, e lógico e *host* para filtrar a análise pelo IP definido.

4. Resultados e Discussões

Para todos os testes foram executadas interações entre os jogadores do protótipo de RPG de caça ao tesouro, testando a utilização do *chat*, comunicação com os objetos do jogo e jogabilidade, aplicando o mesmo cenário para todos os testes.

Foram realizados testes com cinco computadores clientes e um computador para o servidor, e os mesmos cinco computadores utilizados para os clientes foram utilizados para os testes do ambiente distribuído. Os computadores apresentavam especificações semelhantes, com processador equivalente ou superior a um Intel(R) Dual-core, memória equivalente ou superior a 4 GB e todas com sistema operacional Ubuntu 13.04 (ou superior) rodando nativo nas máquinas. Com estes computadores foram realizados testes com dois, três, quatro e cinco jogadores. Cada teste foi executado dez vezes e com duração de dois minutos, sendo obtida a média simples.

O Gráfico 1 exibe o consumo de banda médio total do ambiente cliente-servidor e do ambiente distribuído, variando o número total de jogadores *online*. Essa média total, se dá pela soma da média do consumo de banda de cada computador para cada teste, somado com a média do consumo de banda por parte do servidor para cada teste (no ambiente cliente-servidor), de acordo com o número de clientes conectados. Observa-se também que o crescimento do consumo de banda conforme a quantidade de jogadores é quase diretamente proporcional. O crescimento não linear se torna mais notável conforme a quantidade de jogadores, devido ao fato de que, uma quantidade maior de jogadores tendem a ter um consumo maior de banda.

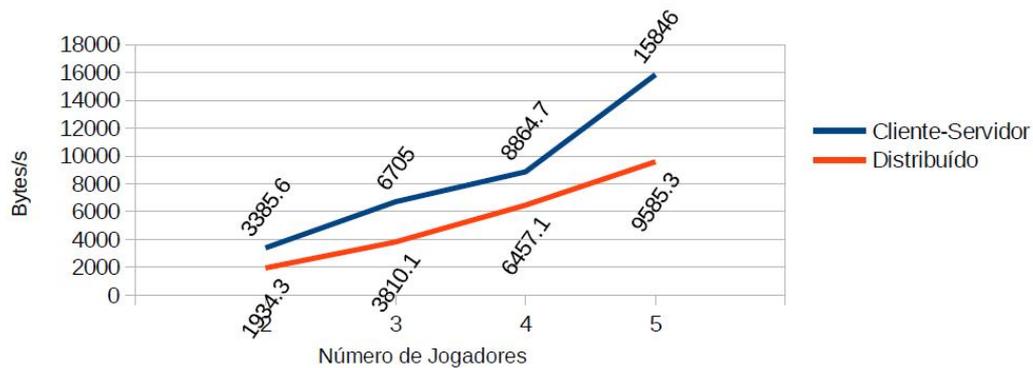


Gráfico 1. Média total de consumo de banda nos ambientes cliente-servidor e distribuído, variando a quantidade total de jogadores.

Analisando os resultados obtidos, observa-se que o consumo de largura de banda pelo ambiente distribuído foi consideravelmente menor que a consumida pelo ambiente cliente-servidor, que obteve um consumo de banda de 15846 Bytes/s com cinco jogadores, enquanto que o ambiente distribuído obteve com a mesma quantidade de jogadores, um consumo de banda de 9585,3 Bytes/s, e tendo um crescimento de até 1,76 vezes menor que o ambiente cliente-servidor.

5. Conclusão e Trabalhos Futuros

Os jogos com ambiente cliente-servidor podem apresentar problemas de escalabilidade, devido ao grande tráfego de informações gerado pelo servidor, isso faz com que ambientes distribuídos se tornem uma alternativa para a comunicação em jogos *multiplayer*.

Nesse contexto, a fim de avaliar o quanto essa alternativa é promissora, foi realizado uma comparação de largura de banda entre os dois ambientes, na qual concluiu-se que com poucos jogadores o ambiente cliente-servidor apresenta um consumo maior de banda que o distribuído e, que o servidor realmente é o gargalo da comunicação, o que o determina como fator limitante de escalabilidade. Portanto, a aplicação de ambientes distribuído em jogos eletrônicos, apresentou-se como uma alternativa promissora, considerando uma quantidade pequena de jogadores.

Referências

- Cecin, F. R.. (2005) “FreeMMG: uma arquitetura cliente-servidor e par-a-par de suporte a jogos maciçamente distribuídos”. Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 101 p..
- Coulouris, G., Dollimore, J., Kindberg, T.. (2007) “Sistemas Distribuídos: conceito e projeto”; tradução João Tortello. 4th ed. Porto Alegre: Bookman. 792 p..
- Cormen, T. H., Leiserson, C. E., Rivest, Ronald L., Stein, C.. (2002) “Algoritmos: Teoria e Prática”. Rio de Janeiro: Elsevier. Tradução da 2th edição americana.
- ENet. (2014) “ENet: Reliable UDP networking library”. Disponível em: <<http://enet.bespin.org/usergroup0.html>>. Acessado em: 20 de fevereiro de 2014.
- Hallberg, B. A.. (2003) “Networking: Redes de Computadores/Teoria e Prática”. Rio de Janeiro: Alta Biiiks. 292 p.

- Kozovits, L. E.. (2003) “Arquiteturas para Jogos *Massive Multiplayer*”. PUC-RioInf.MCC36/03, Rio de Janeiro.
- Novak, J.. (2010) “Desenvolvimento de Games”. São Paulo: Cengage Learning,. 443 p..
- Porfirio, A. J., Hild, T. A.. (2012) “IrrRPG Builder: uma ferramenta livre para desenvolvimento de jogos eletrônicos de RPG”. In: X FITEM - Fórum de informática e tecnologia de Maringá, 2012.
- Shen, B., Guo, J., Chen, P.. (2012) “A survey of P2P virtual world infrastructure”. *Ninth IEEE International Conference on e-Business Engineering*.
- Shen, S.. (2011) “Survey of P2P Game”. Mekelweg 4, Delft, the Netherlands.
- Tanenbaum, A. S., Steen M. V.. (2007) “Sistemas Distribuídos: princípios e paradigmas”. 2th ed. São Paulo: Pearson Prentice Hall.