

Cosmmus? Suporte a jogos maciçamente multijogador em cenários com recursos limitados

Carlos Eduardo Benevides Bezerra¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

Abstract. *Traditionally, a central server is utilized to provide support to massively multiplayer games, where the number of participants is of the order of tens of thousands. In this work, it is proposed the utilization of geographically distributed lower-cost nodes, employing techniques to minimize the necessary bandwidth to these server nodes. One of these techniques is a refinement of interest management algorithm, which obtained significant results in simulations. Other techniques still in preliminary phase of specification are: server nodes overlay network topology construction using knowledge of the real network topology, load balancing and hotspots detection.*

Resumo. *Tradicionalmente, utiliza-se um servidor central para prover suporte a jogos maciçamente multijogador, onde o número de participantes é da ordem de dezenas de milhares. Neste trabalho, propõe-se utilizar nodos geograficamente distribuídos de menor custo, empregando-se técnicas para minimizar o consumo de largura de banda necessário para estes nodos servidores. Uma das técnicas é um refinamento do gerenciamento de interesse, que obteve resultados significativos nas simulações realizadas. Outras técnicas ainda em fase preliminar de especificação são: construção da topologia de rede overlay de nodos servidores utilizando conhecimento da topologia real da rede, balanceamento de carga e detecção de aglomerados de jogadores (hotspots).*

1. Introdução

Atualmente, jogos eletrônicos têm se tornado bastante populares, especialmente os jogos maciçamente multijogador, onde há um número de participantes simultâneos da ordem de dezenas de milhares [Cecin et al. 2004]. Como exemplos, podemos citar World of Warcraft [Blizzard 2004], Lineage II [NCsoft 2003] e Guild Wars [ArenaNet 2005].

Usualmente, o suporte de rede para este tipo de aplicação consiste em um servidor central com recursos - capacidade de processamento e largura de banda para comunicação com os jogadores - super-dimensionados, ao qual se conectam as máquinas clientes. Cada jogador interage através de um destes clientes, que envia suas ações para o servidor, que as processa, verificando que alterações no jogo elas causam, e difunde o resultado para todos os clientes envolvidos. Em virtude do número de participantes simultâneos que este tipo de jogo costuma ter, percebe-se que tais tarefas demandam por uma quantidade de recursos significativa, no que tange a poder de processamento e, principalmente, largura de banda disponível para que sejam enviadas e recebidas as atualizações de estado.

Nos últimos anos, têm-se pesquisado alternativas à abordagem com servidor centralizado. Uma delas é a distribuição, entre os próprios participantes, tanto da

simulação do jogo quanto da responsabilidade de atualizarem-se entre si quando realizam ações. A comunicação entre eles ocorre par-a-par, formando uma rede descentralizada [Schiele et al. 2007]. Esta abordagem seria o ideal, não fossem alguns prolemas que lhe são inerentes. Por exemplo, como os jogadores participam do processamento da simulação, é necessário que eles entrem em acordo no que diz respeito ao estado da partida, sob pena de haver inconsistências caso isto não seja feito.

Outra questão se refere ao número de envios que cada participante tem que executar. No modelo cliente-servidor, basta que cada um envie suas ações para o servidor, que se encarrega de simular e difundir o novo estado para os outros jogadores. No caso do modelo par-a-par, cada par envolvido torna-se responsável por enviar atualizações de estado para os outros participantes. O problema disto reside no fato de que não se pode garantir que todos os jogadores possuam conexões de rede com largura de banda suficiente para isso. Por fim, sem um servidor central, que poderia atuar como árbitro, o jogo torna-se dependente da simulação que os próprios jogadores executam, que pode ser desvirtuada de forma a chegar a um resultado inválido, que beneficie indevidamente determinado jogador ou mesmo que invalide a sessão de jogo.

Além do modelo par-a-par, existe também a alternativa de utilizar um servidor distribuído, em que diversos nodos conectados entre si dividem a tarefa de simular o jogo, como também de enviar as atualizações de estado aos jogadores [Assiotis and Tzanov 2006]. Tal abordagem possibilita o uso de computadores de menor custo para comporem o sistema distribuído servidor, barateando a infra-estrutura de suporte. Questões como consistência e vulnerabilidade a trapaça podem ser abstraídas, restringindo o conjunto de nodos servidores a computadores comprovadamente confiáveis, o que é plausível, levando em conta que o número de nodos servidores deverá ser algumas ordens de grandeza menor do que o número de jogadores. Além disso, não é necessário exigir que cada jogador envie atualizações de estado para os outros jogadores. Com menores exigências de largura de banda e processamento aos jogadores, o jogo torna-se acessível para um maior público.

O presente trabalho propõe uma abordagem de servidores distribuídos, utilizando técnicas para reduzir o consumo de largura de banda causado pelo tráfego do jogo entre os servidores e os clientes, diminuindo a quantidade de recursos necessários, através de um refinamento da técnica de gerenciamento de interesse dos jogadores. [rever-repensar]Uma das propostas consiste em um refinamento da técnica de gerenciamento de interesse [Boulanger et al. 2006], que obteve resultados significativos nas simulações realizadas. Além disso, propõe-se uma técnica que visa prover qualidade de serviço, adaptando a frequência de atualizações enviadas pelo servidor à disponibilidade de recursos. [sure?]Outra consiste em um heurística para detecção de aglomerados de jogadores - ou hotspots - de forma a otimizar o balanceamento de carga entre os servidores.[/sure?]

O artigo está dividido da seguinte maneira: na seção 2 são citados alguns trabalhos relacionados onde buscou-se distribuir o servidor do jogo; na seção 3, são apresentadas as definições de alguns conceitos utilizados ao longo do texto; na seção 4, é descrito o modelo de distribuição proposto; na seção 5 é apresentada a otimização proposta para reduzir o tráfego sem comprometer a qualidade do jogo; nas seções 6 e 7 é descrita a simulação realizada para validar a técnica proposta e os resultados obtidos, respectivamente e, na seção 8, são apresentadas as conclusões a que se chegou neste trabalho.

2. Trabalhos relacionados

Como já foi dito, alguns trabalhos já foram feitos nos últimos anos visando distribuir o suporte a jogos maciçamente multijogador. Uma das abordagens é o modelo par-a-par, que tem algumas dificuldades, no que se refere a consistência do estado do jogo nos diferentes pares participantes, vulnerabilidade a trapaça e uso eficiente de largura de banda. Alguns autores propõem abordagens cujo objetivo é minimizar estes problemas. Um destes trabalhos [Schiele et al. 2007] propõe a divisão do ambiente virtual simulado no jogo em regiões, e dentro de cada região é escolhido um par que será eleito coordenador daquela região. Sua função será a de gerenciar o interesse dos jogadores, verificando para quais pares cada atualização realmente precisa ser enviada. Dessa forma, reduz-se o uso de largura de banda de envio dos pares. No entanto, o uso de largura de banda de envio de cada participante ainda tende a ser significativamente superior àquele necessário quando utilizado o modelo cliente-servidor, pois neste é necessário que cada jogador envie suas ações para apenas um destino. No modelo par-a-par, cada jogador deve atualizar, normalmente, mais de um outro jogador. Além disso, é necessário que o par escolhido para gerenciar o interesse naquela região seja confiável.

Outro trabalho voltado para o modelo par-a-par [Iimura et al. 2004] tem uma abordagem semelhante à de [Schiele et al. 2007], mas sugere que, para cada região do ambiente virtual, seja criada uma "federação de servidores", formada por pares escolhidos entre os participantes. A simulação torna-se mais confiável, já que diferentes nodos irão gerenciar aquele lugar no mundo do jogo e precisarão estar em acordo para que a simulação prossiga. Porém, o risco dos nodos escolhidos para gerenciarem aquela região cometerem trapaça de conluio [Yan and Randell 2005] não é eliminado. Além disso, o próprio acordo entre os nodos servidores, que provê maior confiabilidade na simulação, implica em grande quantidade de tráfego entre os nodos participantes, além de potencialmente atrasar cada passo da simulação.

Em [Assiotis and Tzanov 2006], é proposta uma arquitetura distribuída para jogos maciçamente multijogador, também baseada na divisão do ambiente virtual do jogo em regiões, porém a cada uma destas estaria associado um nodo servidor. O jogador que estivesse situado em determinado lugar no mundo virtual deveria conectar-se ao servidor responsável por aquela região. Desta forma, cada servidor agruparia diferentes jogadores, baseado em sua localidade no ambiente do jogo. Para alcançar consistência entre os diferentes nodos servidores efetuando a simulação, é utilizado o conceito de travas. Quando um determinado nodo servidor precisa alterar o estado de uma entidade qualquer da partida, primeiro precisa obter acesso exclusivo àquela entidade. Para isso, ele negocia com os outros nodos servidores que possam também querer fazer alguma alteração, para somente então efetuar a mudança. Quando termina, o acesso é liberado, e os outros servidores são avisados através de mensagens.

A primeira grande restrição no trabalho de [Assiotis and Tzanov 2006], no entanto, é a premissa de que os nodos servidores estão conectados através de uma rede de alta velocidade e baixa latência, o que não pode ser assumido quando se trata de nodos de mais baixo custo geograficamente distribuídos. Outro problema é que a questão da escalabilidade é tratada através da pura e simples expansão do ambiente virtual, supondo que os jogadores se espalharão por ele. Por último, sugere-se resolver o problema de haver um grande número de jogadores no mesmo lugar através de sucessivos reparticionamen-

tos recursivos das regiões, de forma a dividir os jogadores entre diferentes servidores. No entanto, existe um limite para o reparticionamento do ambiente virtual, e é deixado de lado o que fazer quando é atingido este limite.

3. Definições

Neste artigo será descrito o modelo de suporte a jogos maciçamente jogador proposto e, ao longo do texto, serão utilizados alguns termos que precisam antes ser definidos, pois alguns deles são específicos da área de jogos de computador. Define-se avatar como sendo a representação no ambiente virtual do jogador, que, através dele, interage com o mundo do jogo e com outros jogadores. Regiões são divisões do ambiente virtual, cada uma podendo conter jogadores presentes. Assumindo regiões contíguas, define-se fronteira como sendo a divisa entre duas regiões adjacentes. Será utilizado o termo cliente para referir-se ao computador utilizado por cada jogador para conectar-se a um dos servidores do jogo, assim como o termo servidor fará referência a cada nodo integrante do sistema distribuído que estará servindo o jogo.

4. Modelo de distribuição

Usualmente, o ambiente virtual onde ocorre a simulação do mundo do jogo fica inteiramente a cargo de um servidor central - ou um cluster de servidores interligados por algum tipo de rede de alta velocidade e baixa latência - que processa as ações de cada jogador e calcula o resultado das mesmas, que é difundido para todos os jogadores envolvidos com aquela ação. Neste trabalho, o ambiente virtual é particionado em regiões, cada uma gerenciada por um servidor. As regiões serão contíguas, explorando a localidade dos avatares dos jogadores. Dessa forma, jogadores próximos no jogo tenderão a estarem conectados ao mesmo servidor, fazendo com que sua interação seja beneficiada. Em situações em que jogadores interagindo entre si estivessem conectados a diferentes servidores implicaria em maior tráfego, pois seria necessário algum tipo de negociação entre os servidores aos quais os diferentes jogadores estão conectados, para que os estados da simulação em ambos fossem idênticos.

Nas próximas seções serão descritos alguns mecanismos básicos do modelo de distribuição proposto. Na seção 4.3 é detalhado melhor a comunicação entre Na seção 4.1, explica-se como é iniciado o sistema servidor, e de que forma o ambiente é reparticionado à medida em que novos nodos servidores são introduzidos.

4.1. Bootstrap

4.2. Disseminação de informações

O objetivo deste modelo é ser o mais escalável possível, no que diz respeito à extensão do ambiente virtual e do número de nodos servidores. Sendo assim, deseja-se que um número maior de servidores não implique em um sobrecusto crescente, a ponto de inviabilizar a utilização deste modelo de suporte. Contudo, permanece a necessidade de disseminar tanto a lista de nodos servidores, quanto os estados dos diferentes avatares entre através do sistema. Caso houvesse obrigação de garantia de entrega de cada uma dessas informações a todos os nodos envolvidos, ter-se-ia uma complexidade alta, pois cada um dos n nodos servidores estaria enviando continuamente informações para $n - 1$ outros servidores. Assim, o número de mensagens trocadas teria uma complexidade $O(n^2)$,

levando a uma saturação da banda de comunicação dos nodos à medida em que fosse aumentada a escala, tanto em número de nodos servidores, quanto de jogadores.

Para evitar isso, propõe-se a utilização de difusão probabilística destas informações, de maneira que não é necessário garantir a entrega de todas elas. Segundo [Kermarrec et al. 2003], é possível tornar esse tipo de difusão quase tão confiável quanto a difusão determinística, porém reduzindo sensivelmente o tráfego gerado pelas mensagens e confirmações.

A difusão do estado dos avatares ocorre da seguinte forma: como cada cliente está conectado a um servidor, este servidor terá o estado mais recente do avatar daquele jogador, que então é difundido para os servidores vizinhos, que repassa para seus vizinhos e assim por diante. Dessa forma, em algum momento todos servidores - ou boa parte deles - terão o estado daquele avatar. Além disso, quanto mais próximo o servidor estiver da região onde está o avatar, maior a probabilidade dele ter recebido o estado, assim como mais atualizado ele estará.

4.3. Jogadores e servidores

Cada servidor terá uma lista dos endereços e portas dos outros servidores, assim como a região a que cada um está associado. Quando um jogador deseja participar do jogo, deve conectar-se a um servidor qualquer. Se o jogador entrar com um avatar novo, é escolhida alguma região para ele começar a jogar, e o servidor a que ele se conectou irá redirecioná-lo para o servidor correto. O critério de escolha pode ser o número de jogadores em cada região, como também pode ser feita simplesmente uma seleção aleatória da região. Se já tiver jogado antes, será redirecionado para o servidor responsável pela área na qual ele parou de jogar da última vez. Desta forma, aumenta-se a probabilidade do jogador recomençar a partida com as informações mais recentes a respeito de seu estado.

5. Área de interesse gradual

6. Simulação

7. Resultados

8. Conclusões

9. General Information

All full papers and posters (short papers) submitted to some SBC conference, including any supporting documents, should be written in English or in Portuguese. The format paper should be A4 with single column, 3.5 cm for upper margin, 2.5 cm for bottom margin and 3.0 cm for lateral margins, without headers or footers. The main font must be Times, 12 point nominal size, with 6 points of space before each paragraph. Page numbers must be suppressed.

Full papers must respect the page limits defined by the conference. Conferences that publish just abstracts ask for **one**-page texts.

10. First Page

The first page must display the paper title, the name and address of the authors, the abstract in English and “resumo” in Portuguese (“resumos” are required only for papers written

in Portuguese). The title must be centered over the whole page, in 16 point boldface font and with 12 points of space before itself. Author names must be centered in 12 point font, bold, all of them disposed in the same line, separated by commas and with 12 points of space after the title. Addresses must be centered in 12 point font, also with 12 points of space after the authors' names. E-mail addresses should be written using font Courier New, 10 point nominal size, with 6 points of space before and 6 points of space after.

The abstract and "resumo" (if is the case) must be in 12 point Times font, indented 0.8cm on both sides. The word **Abstract** and **Resumo**, should be written in boldface and must precede the text.

11. CD-ROMs and Printed Proceedings

In some conferences, the papers are published on CD-ROM while only the abstract is published in the printed Proceedings. In this case, authors are invited to prepare two final versions of the paper. One, complete, to be published on the CD and the other, containing only the first page, with abstract and "resumo" (for papers in Portuguese).

12. Sections and Paragraphs

Section titles must be in boldface, 13pt, flush left. There should be an extra 12 pt of space before each title. Section numbering is optional. The first paragraph of each section should not be indented, while the first lines of subsequent paragraphs should be indented by 1.27 cm.

12.1. Subsections

The subsection titles must be in boldface, 12pt, flush left.

13. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

14. Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.



Figure 1. A typical figure

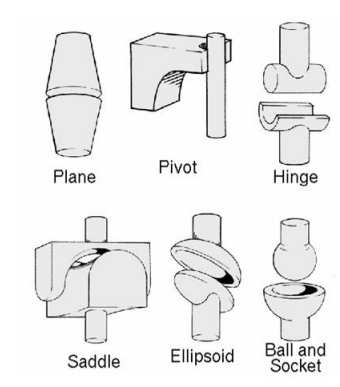


Figure 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 13.

15. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

References

ArenaNet (2005). Guild wars. <http://www.guildwars.com/>.

Assiotis, M. and Tzanov, V. (2006). A distributed architecture for MMORPG. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*.

Blizzard (2004). World of warcraft. <http://www.worldofwarcraft.com/>.

Table 1. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 \pm 5.22	7.01 \pm 6.84
Selection with in- depth movements	6.29 \pm 4.99	12.22 \pm 11.33
Manipulation with side movements	4.66 \pm 4.94	3.47 \pm 2.20
Manipulation with in- depth movements	5.71 \pm 4.55	5.37 \pm 3.28

- Boulanger, J., Kienzle, J., and Verbrugge, C. (2006). Comparing interest management algorithms for massively multiplayer games. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*.
- Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons Ltd.
- Cecin, F., Real, R., de Oliveira Jannone, R., Geyer, C., Martins, M., and Barbosa, J. (2004). FreeMMG: A Scalable and Cheat-Resistant Distribution Model for Internet Games. *IEEE Int. Sym. on Distributed Simulation and Real-Time Applications*, pages 83–90.
- Iimura, T., Hazeyama, H., and Kadobayashi, Y. (2004). Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games. *Proceedings of ACM SIGCOMM 2004 workshops on NetGames' 04: Network and system support for games*, pages 116–120.
- Kermarrec, A., Massoulié, L., and Ganesh, A. (2003). Probabilistic Reliable Dissemination in Large-Scale Systems. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, pages 248–258.
- Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.
- NCsoft (2003). Lineage ii. <http://www.lineage2.com/>.
- Schiele, G., Suselbeck, R., Wacker, A., Hahner, J., Becker, C., and Weis, T. (2007). Requirements of Peer-to-Peer-based Massively Multiplayer Online Gaming. *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, pages 773–782.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.
- Yan, J. and Randell, B. (2005). A systematic classification of cheating in online games. *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–9.