

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CARLOS EDUARDO BENEVIDES BEZERRA

**Suporte distribuído para jogos MMG em
cenários com recursos limitados
(Título Provisório)**

Plano de Estudos e Pesquisa

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, dezembro de 2007

SUMÁRIO

RESUMO	3
1 MOTIVAÇÃO	4
2 OBJETIVO	7
3 METODOLOGIA	8
4 CRONOGRAMA	9
REFERÊNCIAS	11
ASSINATURAS	13

RESUMO

Este trabalho visa analisar a aplicabilidade das técnicas utilizadas para dar suporte a jogos maciçamente multijogador em cenários onde os recursos computacionais são mais limitados. Como geralmente são utilizadas redes de servidores de alto custo, com links que dão suporte à comunicação entre dezenas de milhares de jogadores (CECIN et al., 2004), é necessário revisar o que foi feito e analisar sua aplicabilidade em arquiteturas que não dispõem de tais recursos. Assim sendo, objetiva-se avaliar e, quando possível, aperfeiçoar as técnicas pesquisadas, considerando um cenário em que a infra-estrutura de suporte se restringe a um conjunto de nodos de capacidade menor - ou super-nodos, no sentido em que têm links e capacidade de processamento superiores às dos usuários de Internet em geral, mas inferiores às dos grandes servidores. Pretende-se, por fim, construir um middleware que dê suporte à distribuição das cargas de processamento e comunicação associadas ao jogo entre o conjunto de nodos servidores e integrar - tantas quanto for possível - as técnicas pesquisadas e/ou propostas. Após isso, serão feitas simulações para validar o trabalho.

Palavras-chave: Jogos maciçamente multijogador, MMG, simulação interativa distribuída.

1 MOTIVAÇÃO

Jogos do tipo MMG (*massively multiplayer games*, jogos maciçamente multijogador) têm se popularizado bastante nos últimos tempos. Além de poderem ser jogados on-line, permitem a interação simultânea de um grande número de participantes. Nos casos de maior sucesso, como World of Warcraft (BLIZZARD, 2004) e Lineage II (NCSOFT, 2003), por exemplo, é dado suporte a uma base de dezenas a centenas de milhares de jogadores simultâneos (CHEN; MUNTZ, 2006). Estes jogadores podem, então, interagir entre si e com o ambiente virtual do jogo. Cada jogador controla uma entidade chamada de *avatar*, que nada mais é que sua representação no ambiente virtual.

Jogos dessa natureza têm alguns requisitos (SCHIELE et al., 2007), dentre os quais podem ser citados: **persistência** do estado do mundo virtual e dos avatares; **disponibilidade** do serviço; **consistência** do estado do mundo e dos avatares entre os diferentes participantes; **segurança**, no que se refere a tentativas de subversão das regras do jogo por trapaceiros e **escalabilidade**, tendo em vista o grande número esperado de jogadores.

Para dar suporte à grande quantidade de participantes, geralmente é usado o paradigma cliente-servidor, sendo que toda interação é feita através da máquina servidora. Neste paradigma, o computador servidor possui a cópia oficial do estado e é responsável pela computação sobre a mesma, enquanto o cliente atua na apresentação do estado para o usuário. O servidor é responsável, também, por atualizar os clientes, em tempo real, sobre as alterações que ocorrem no ambiente virtual do jogo. A centralização nesse computador servidor, devido ao grande número de participantes, faz com que seu custo de manutenção seja bastante elevado. Geralmente, é necessária uma capacidade de comunicação de dezenas de GBps e um razoável poder de processamento para executar toda a simulação. Torna-se inviável, por exemplo, a manutenção de jogos maciçamente multijogador por grupos independentes sem grande disponibilidade de recursos financeiros, como o Indigente (INDIGENTE, 2004), por exemplo.

A **arquitetura cliente-servidor** supre diversos aspectos necessários para a execução satisfatória de jogos do tipo MMG, o que inclui um alto nível de controle sobre o sistema como um todo, o que facilita autenticação, persistência e segurança. Porém isso custa caro, como já foi dito, além de ser um possível gargalo. Objetivando minimizar este problema, foram propostas algumas alternativas. Uma delas é a de usar computação agregada, onde um cluster, ao invés de um computador único, faz o papel de servidor. Tal abordagem tem um ganho expressivo de poder de processamento, mas não resolve todos os problemas dos jogos maciçamente multijogador. Deve-se prover também a largura de banda necessária para dar suporte ao tráfego intenso entre o servidor e os jogadores.

Outra abordagem possível é a **arquitetura par-a-par**, ou P2P, onde se divide a simulação entre os computadores envolvidos. Pode-se ter um sistema sem qualquer servidor, onde os pares (antes clientes), que são as máquinas dos jogadores, entram em algum tipo

de acordo para os diversos passos da simulação. No que se refere à escalabilidade, tal abordagem não é ótima, pois garantir esse "acordo" é custoso em termos de troca de mensagens (LAMPORT; SHOSTAK; PEASE, 1982). Ainda que seja eleito um dos pares para decidir o andamento da simulação, ainda haverá o problema de que todos os pares precisarão trocar mensagens com todos. Tendo-se n pares, há uma complexidade de $O(n^2)$ trocas de mensagem para cada passo da simulação. É evidente que tal abordagem não é tão escalável quanto se possa querer para um sistema onde se pretende executar um jogo maciçamente multijogador. Além disso, seria necessário prover armazenamento distribuído e recuperação dos estados do jogo.

Alguns trabalhos já foram realizados no sentido de tornar jogos em redes P2P mais escaláveis, como (SCHIELE et al., 2007). Para reduzir o tráfego entre os pares, cada um envia atualizações de estado apenas àqueles que tiverem interesse nas mesmas. Para atingir este objetivo, o ambiente virtual é dividido em regiões, cada uma com um coordenador. O coordenador de cada região é eleito entre os pares ali presentes e se encarrega, apenas, de decidir para quem cada atualização de estado interessa - ele *não* intermedia as trocas de mensagens entre os pares, que se comunicam diretamente entre si. No entanto, tal abordagem se baseia no fato de que aquele par seria confiável, o que não pode ser garantido, já que o software utilizado por aquele usuário pode ter sido alterado de forma a agir de maneira incorreta. Outra abordagem seria a de eleger múltiplos coordenadores por região, mas isso implicaria na implementação de algum mecanismo de votação, além de depender da disponibilidade de pares para gerenciarem. Além disso, não seria eliminada a necessidade de cada par enviar atualizações de estado a diversos outros pares. Outros trabalhos que têm proposta bastante semelhante são (EL RHALIBI; MERABTI, 2005) e (IIMURA; HAZEYAMA; KADOBAYASHI, 2004).

Existem também as propostas de **arquiteturas híbridas**, que utilizam servidores ao mesmo tempo em que fazem uso de infra-estrutura P2P. É proposta, por exemplo, uma arquitetura onde pares e servidor dividem a simulação do jogo (CHEN; MUNTZ, 2006). O ambiente virtual é dividido em regiões, cada uma gerenciada por um dos pares, que atua como um sub-servidor. É para ele que cada um dos outros pares naquela região envia cada atualização de estado, que de lá é encaminhada aos outros pares interessados na mesma. Isto pode gerar problemas de segurança e disponibilidade, já que não há garantias de que aquele par é confiável para servir aquela região, ou que ele irá permanecer no sistema enquanto for necessário. Para tratar a segurança, os autores sugerem o uso dos agregados de pares - que consistem no gerenciador de região, mais pares "reserva" - para fazerem verificação da simulação, além de prover robustez ao sistema, no caso do gerenciador sofrer colapso. Porém, tal abordagem novamente remete ao problema da eficiência de comunicação e escalabilidade, já que é necessário haver acordo entre os pares integrantes daquele agregado.

Outra arquitetura híbrida é a do FreeMMG (CECIN et al., 2004). Nela, os pares se organizam de maneira P2P em cada região do ambiente virtual e o servidor intermedia a comunicação entre diferentes regiões. Novamente, tem-se o problema da segurança: os pares dentro de uma região controlam a simulação que ocorre ali, podendo subvertê-la. É feita uma abordagem probabilística, utilizando um par selecionado aleatoriamente de outro ponto do ambiente para verificar a simulação naquela região. Espera-se que, caso os pares ali desejem entrar em conluio para subverter o jogo, pelo menos o nodo inserido ali detecte as ações inválidas e reporte ao servidor. No entanto, além de não garantir completamente segurança, persiste o problema de poder haver muitos pares se comunicando com muitos, o que pode comprometer a qualidade do jogo.

O que se pretende investigar é o uso de um sistema servidor composto de "super-nodos" distribuídos geograficamente. Cada super-nodo teria uma capacidade de processamento e link de comunicação mínimos exigidos para fazer parte do sistema. No entanto, como a simulação seria distribuída entre eles, não seria necessário que cada um destes super-nodos tivesse a mesma capacidade de processamento ou comunicação possuída pelos grandes servidores centrais usados comumente para jogos MMG. O mundo do jogo poderia então ser dividido em regiões, cada uma gerenciada por um destes super-nodos. Dessa forma, os jogadores que estivessem em uma dada região apenas enviariam atualizações de estado àquele nodo servidor, que as processaria, armazenaria e enviaria para a quem fossem relevantes.

Utilizando esta abordagem, seria preciso apenas garantir a confiabilidade dos nodos servidores, que, então, verificariam se cada ação dos jogadores é válida. Assim, torna-se mais simples a tarefa de garantir segurança no sistema, pois pode-se assumir que cada super-nodo é confiável. Outro aspecto - o da disponibilidade - seria melhor satisfeito, pois cada região do mundo virtual estaria sendo servida por um nodo que estaria dedicado a esta tarefa.

Seguindo esta última vertente, existem alguns problemas que, ou estão por resolver, ou estão sendo pesquisados atualmente, tais como:

- Encontrar uma maneira de particionar o espaço virtual, mapeando cada região a um nodo servidor, de forma que nodos com boa comunicação entre si estejam próximos na topologia lógica da rede;
- Balancear dinamicamente a carga da simulação entre os nodos servidores, de forma que seja mantida uma qualidade mínima de serviço em cada uma das diferentes regiões do ambiente virtual;
- Manter consistente a simulação nas diferentes regiões, entre jogadores da mesma região, e de jogadores em regiões adjacentes;
- Coordenar a entrada, a saída e a detecção de colapso de nodos do sistema;
- Distribuir o armazenamento persistente do estado dos jogadores, de forma a poder recuperá-lo quando necessário;
- Otimizar a comunicação entre clientes e servidores, de forma a economizar o máximo possível de largura de banda, sem que isso seja perceptível pelos jogadores;
- Como cada nodo tem recursos mais limitados que os de um grande servidor central, deve-se buscar algoritmos adaptativos que tornem a simulação (atualizações de estado e sincronização da simulação entre os clientes) coerente com os recursos disponíveis no momento;

Foram feitos trabalhos que buscam tratar alguns destes problemas, como balanceamento de carga (LEE; LEE, 2003), assim como foram feitas algumas propostas de arquiteturas de suporte a MMG (NG et al., 2002) e (ASSIOTIS; TZANOV, 2006). No entanto, tais propostas levam em consideração que os servidores são de grande porte e ligados através de links de comunicação poderosos, havendo muito pouco atraso entre eles. Cabe, então, fazer a análise da aplicabilidade das técnicas já pesquisadas em um contexto de recursos mais escassos e propor, tendo isso em vista, aperfeiçoamento das mesmas e uma arquitetura que as integre, além de efetuar implementação e testes de um protótipo que as utilize.

2 OBJETIVO

O objetivo é criar um middleware de suporte a jogos maciçamente multijogador. Abaixo deste middleware, poderá ser formado um sistema geograficamente distribuído de nodos servidores. Sobre ele, deverá ser possível implementar este tipo de jogo, podendo ser utilizado um paradigma cliente/servidor, possibilitando um menor esforço de projeto para distribuir as cargas de processamento e de comunicação impostas pelo acréscimo de jogadores. Além de distribuir as cargas computacionais impostas pelo jogo, este middleware deverá utilizar técnicas com o fim de reduzir o tráfego entre os componentes deste sistema (nodos servidores e máquinas clientes dos jogadores), economizando largura de banda. Tais técnicas são o foco deste trabalho, tendo em vista que se considera um cenário com recursos voláteis e limitados.

Existem outros aspectos relevantes, além da distribuição do servidor e da economia de largura de banda, que poderão ser acrescentados ao middleware, dependendo da disponibilidade de tempo para realizar tal tarefa.

3 METODOLOGIA

Para atingir o objetivo definido na seção anterior, deverão ser cumpridas algumas etapas, que são as seguintes:

1. Revisar as técnicas utilizadas para dar suporte distribuído a jogos MMG;

Será dada continuidade à investigação bibliográfica que já foi iniciada para elaboração deste plano de estudos e pesquisa, além de buscar separá-las em categorias segundo: contexto considerado e problemas que buscam solucionar.

2. Analisar a aplicabilidade destas técnicas em cenários em que cada nodo do sistema tem recursos mais limitados;

Técnicas utilizadas para dar suporte distribuído a jogos MMG costumam se basear em cenários onde há grande disponibilidade de recursos, com grandes servidores ligados através de rede local com capacidade de comunicação super-dimensionada. Será analisada a viabilidade de serem utilizadas em um contexto onde os nodos têm recursos mais limitados.

3. Propor integração das mesmas, com otimizações focadas no contexto definido (disponibilidade de super-nodos, ao invés de servidores poderosos);

Buscar-se-á fazer aperfeiçoamento de algumas destas técnicas, tendo como ênfase o contexto de super-nodos distribuídos geograficamente e, posteriormente, será proposta uma integração das mesmas.

4. Implementar um protótipo que integre as técnicas estudadas e/ou desenvolvidas nas etapas anteriores;

5. Efetuar simulações que demonstrem o resultado obtido do trabalho e avaliar seus resultados.

4 CRONOGRAMA

A Tabela 4.1 apresenta o cronograma de atividades a serem desenvolvidas, conforme citadas abaixo

1. Levantamento bibliográfico do estado-da-arte das técnicas de distribuição do suporte a MMG;
2. Levantamento bibliográfico de técnicas de distribuição em geral, que possam ser aplicadas a jogos MMG;
3. Análise e categorização destas técnicas;
4. Elaboração de um modelo arquitetural de suporte distribuído a jogos MMG;
5. Integração das técnicas no modelo proposto;
6. Implementação de um protótipo com o fim de testes;
7. Levantamento dos simuladores mais indicados para efetuar tais testes;
8. Execução dos testes em um dos simuladores pesquisados;
9. Análise dos resultados;
10. Apresentação do seminário de andamento na semana acadêmica;
11. Escrita da dissertação;
12. Escrita de artigos;
13. Entrega da dissertação;
14. Defesa da dissertação;

Tabela 4.1: Cronograma

[illegible]

REFERÊNCIAS

ASSIOTIS, M.; TZANOV, V. A distributed architecture for MMORPG. **Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games**, [S.l.], 2006.

BLIZZARD. **World of Warcraft**. <http://www.worldofwarcraft.com/>.

CECIN, F.; REAL, R.; OLIVEIRA JANNONE, R. de; GEYER, C.; MARTINS, M.; BARBOSA, J. FreeMMG: a scalable and cheat-resistant distribution model for internet games. **IEEE Int. Sym. on Distributed Simulation and Real-Time Applications**, [S.l.], p.83–90, 2004.

CHEN, A.; MUNTZ, R. Peer clustering: a hybrid approach to distributed virtual environments. **Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games**, [S.l.], 2006.

EL RHALIBI, A.; MERABTI, M. Agents-based modeling for a peer-to-peer MMOG architecture. **Computers in Entertainment (CIE)**, [S.l.], v.3, n.2, p.3–3, 2005.

IIMURA, T.; HAZEYAMA, H.; KADOBAYASHI, Y. 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**, [S.l.], p.116–120, 2004.

INDIGENTE. **Interactive Digital Entertainment**. Grupo de pesquisa na área de jogos da UFBA, formado em abril de 2004. <http://indigente.dcc.ufba.br/>.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The Byzantine Generals Problem. **ACM Transactions on Programming Languages and Systems (TOPLAS)**, [S.l.], v.4, n.3, p.382–401, 1982.

LEE, K.; LEE, D. A scalable dynamic load distribution scheme for multi-server distributed virtual environment systems with highly-skewed user distribution. **Proceedings of the ACM symposium on Virtual reality software and technology**, [S.l.], p.160–168, 2003.

NCSOFT. **Lineage II**. <http://www.lineage2.com/>.

NG, B.; SI, A.; LAU, R.; LI, F. A multi-server architecture for distributed virtual walk-through. **Proceedings of the ACM symposium on Virtual reality software and technology**, [S.l.], p.163–170, 2002.

SCHIELE, G.; SUSELBECK, R.; WACKER, A.; HAHNER, J.; BECKER, C.; WEIS, T. Requirements of Peer-to-Peer-based Massively Multiplayer Online Gaming. **Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid**, [S.l.], p.773–782, 2007.

ASSINATURAS

Carlos Eduardo Benevides Bezerra

Prof. Dr. Cláudio Fernando Resin Geyer