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

CARLOS EDUARDO BENEVIDES BEZERRA

Um modelo de suporte distribuído para jogos MMOG em cenários com recursos computacionais limitados (Título Provisório)

Plano de Curso para Doutorado

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, dezembro de 2008

SUMÁRIO

RESUMO

Este plano de curso de doutorado propõe um modelo de suporte a jogos online maciçamente multijogador fazendo uso de sistemas de compartilhamento em larga escala de recursos computacionais, tais como redes par-a-par. Seguindo este modelo, sobre uma rede par-a-par seria formado um sistema que atuaria como servidor de um jogo MMOG. A cada nodo deste sistema seria designada uma porção da carga de comunicação e processamento do jogo – mais especificamente, o ambiente virtual do jogo seria particionado em regiões, sendo que a cada uma delas seria designado um conjunto de um ou mais nodos servidores. Tal sistema seria percebido como um servidor central pelos jogadores, que conectar-se-iam a ele através de um *Gateway*, sendo cada um redirecionado para o nodo servidor adequado. Para que esse sistema servidor funcione como esperado, será necessário tratar algumas questões, tais como manter consistente o andamento do jogo nos diferentes nodos servidores, de forma que cada jogador tenha a mesma visão; gerenciamento da carga atribuída a cada servidor, sendo que maior parte dela recai sobre a rede e, por se tratar de um jogo, é bastante variável; o que fazer quando um nodo servidor sai da rede ou entra em colapso e, por fim, como difundir e armazenar os estados dos jogadores de maneira distribuída, de forma que sempre se possa recuperar o estado mais recente.

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

1 INTRODUÇÃO

Jogos online maciçamente multijogador – ou MMOGs, *massively multiplayer online games* – são o tema central deste plano de doutorado. Geralmente, utiliza-se uma infra-estrutura cliente-servidor para dar suporte a este tipo de jogo. Neste plano será proposto um modelo de distribuição do lado servidor, de maneira a reduzir ou eliminar seu custo de manutenção. Nas próximas seções será apresentado o contexto referente a jogos multijogador e MMOGs.

1.1 Jogos Multijogador

Nas últimas décadas, desde o surgimento do computador pessoal, jogos de computador tem se popularizado de maneira crescente. Mais ainda, após o surgimento da Internet aliado ao fato de as conexões domésticas estarem cada vez mais baratas e mais rápidas, jogos online multijogador têm merecido atenção especial. Estes jogos se caracterizam por haver vários participantes interagindo uns com os outros, utilizando uma conexão através da Internet entre seus dispositivos computacionais, que são geralmente computadores pessoais ou consoles (dispositivos computacionais específicos para jogos). Este tipo de jogo pode ser dividido em vários gêneros, como por exemplo:

- FPS: *first-person shooting*, ou jogos de tiro com visão em primeira-pessoa, em que cada jogador controla um personagem que dispõe de diferentes armas, com as quais enfrenta monstros ou personagens de outros jogadores. São caracterizados por ação rápida, incluindo atirar, mover-se, esquivar-se etc.;
- RTS: *real-time strategy*, ou jogos de estratégia em tempo-real, onde cada jogador controla um exército, mas suas ações são executadas simultaneamente às ações dos outros jogadores, ao invés de serem separadas em turnos. A ação é mais lenta que em jogos FPS;
- RPG: *role-playing game*, ou jogo de interpretação de papéis, no qual cada jogador tem um personagem que evolui com o tempo, adquirindo mais poder e acumulando tesouros. Jogos deste tipo são caracterizados por não terem um início ou fim de partida definidos, constituindo um mundo virtual de estado persistente. Os jogadores então podem evoluir seus personagens, desconectarem-se e reconectarem-se mais tarde para continuarem jogando a partir do ponto onde pararam, pois seu estado estará armazenado.

Como exemplos de jogos multijogador online conhecidos, podem ser citados *Quake* (id Software, 1996), um jogo do gênero FPS lançado em meados da década de 90; *Starcraft: Brood War* (BLIZZARD, 1998), da mesma época, porém consistindo de um jogo

RTS, Counter-Strike (VALVE, 2000), FPS lançado em 2000 e Diablo II (BLIZZARD, 2000), RPG lançado no mesmo ano.

1.2 Jogos Online Maciçamente Multijogador (MMOGs)

Da mesma forma que jogos de computador evoluíram para jogos multijogador on-line após a popularização da Internet, evoluiu-se em seguida para jogos online maciçamente multijogador (ou MMOG, *massively multiplayer online games*), graças ao barateamento das conexões à Internet e o aumento da sua velocidade. Jogos deste tipo 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), Lineage II (NCSOFT, 2003) e EVE Online (CCP, 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.

Uma das principais características destes jogos, presente em quase todos os MMOGs comerciais, são as partidas muito longas. Em alguns MMORTS (MMOGs de estratégia em tempo real), como Travian (TRAVIAN, 2004), as partidas chegam a durar de seis meses a um ano. Já nos MMORPGs (MMOGs no estilo de RPG), como é o caso do World of Warcraft, as partidas não tem um momento determinado para que acabem. Os mundos destes jogos são tão vastos e populosos – no que diz respeito a número de jogadores –, que eles passam a evoluir e modificar seu estado independente de parte dos jogadores estarem jogando ou não, pois sempre tem alguém conectado dando continuidade à partida. Por esta razão, espera-se que cada jogador possa desconectar-se e continuar jogando mais tarde, sem ter que recomeçar do zero. Além disso, espera-se que as alterações feitas sobre o próprio mundo do jogo persistam, não importando quantas vezes cada jogador entrou e saiu.

Cada jogador controla uma entidade chamada de *avatar*, que nada mais é que sua representação no ambiente virtual - um personagem que executa as ações determinadas pelo jogador, que dessa forma passa a interferir na história e no encaminhamento do jogo. Para que a partida se mantenha consistente para os diferentes participantes, cada ação executada por cada avatar deve ser processada pelo servidor, que calculará o estado resultante dessa ação no jogo. Esse novo estado é então difundido para os outros jogadores. Outras alterações no mundo do jogo, como destruição ou criação de objetos no ambiente, e eventos, como ações de personagens controlados pelo servidor, devem também ser transmitidos aos jogadores envolvidos. É então através deste envios de comandos do jogador e recebimento do novo estado do servidor que o jogadores conseguem interagir entre si e com o mundo simulado no jogo.

No entanto, percebe-se que tal mecanismo pode facilmente saturar a banda e sobrecarregar o(s) processador(es) do servidor. Geralmente o que se faz é montar uma infraestrutura que dispõe de um grande servidor central, geralmente um cluster com seus nodos ligados por uma rede local de alta velocidade e baixo atraso, com conexão à Internet de algumas centenas ou milhares de MBps. O maior peso reside na manutenção desta conexão, que visa a enviar para cada jogador o estado mais recente do jogo sempre que este mudar, dentro de um limite estabelecido de atraso, que irá depender do gênero do jogo.

2 MOTIVAÇÃO E ESTADO DA ARTE

Já foi dito que MMOGs se caracterizam por uma grande quantidade de jogadores. Para dar suporte a essa 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 do jogo 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 com a Internet de alguns GBps e poder de processamento suficiente para executar toda a simulação. Torna-se inviável, por exemplo, a manutenção de jogos online maciçamente multijogador por empresas pequenas ou grupos independentes com orçamento limitado.

Em razão deste custo elevado, tem-se buscado soluções alternativas, distribuindo-se de alguma maneira as cargas de processamento e de comunicação decorrentes do jogo. Geralmente, é proposto o uso de redes par-a-par ou, em alguns casos GRIDs (IBM, 2003; WANG; WANG; LAU, 2004). Porém, um MMOG é uma simulação de um ambiente virtual dinâmico, com limite máximos de atraso entre o envio das ações do jogador e recebimento do estado resultante. Além disso, possui outros requisitos, que precisam ser observados com atenção de maneira que os jogadores tenham uma experiência adequada de jogo. Distribuir toda esta estrutura entre diferentes máquinas ligadas através de conexões através da Internet – como no caso das redes par-a-par – e ao mesmo tempo satisfazer os requisitos de um MMOG torna-se uma tarefa desafiadora, que tem motivado diversos trabalhos de pesquisa nos últimos anos.

Nas seções a seguir, serão apresentados alguns requisitos dos jogos MMOG, além do estado da arte no que se refere a distribuição da infra-estrutura de suporte a este tipo de jogo. Por fim, serão apresentadas questões de pesquisa que ainda estão em aberto, e que podem ser tratadas em um trabalho de doutorado.

2.1 Requisitos dos jogos MMOG

Durante o desenvolvimento de trabalhos anteriores (BEZERRA; CECIN; GEYER, 2008; VILANOVA et al., 2008) e deste plano de curso de doutorado, além de consulta à literatura da área (DARLAGIANNIS; HECKMANN; STEINMETZ, 2006; SCHIELE et al., 2007) foram identificados alguns requisitos dos jogos MMOG, que são os mais importantes do ponto de vista da infra-estrutura de suporte:

- **Persistência** - Jogos MMOG, em geral, consistem em um mundo virtual onde os jogadores interagem entre si e com o mundo virtual, alterando o estado do mesmo e de seus avatares (representação virtual do jogador no jogo). Espera-se que cada jogador possa deixar o jogo e voltar algum tempo depois sem que essas mudanças tenham sido desfeitas. Por este motivo, é necessário suporte a persistência dos estados de cada avatar e do mundo virtual como um todo;

- **Disponibilidade** - Em geral, jogos MMOG são pagos - tais como World of Warcraft (BLIZZARD, 2004) - e, por causa disso, seus usuários costumam ser exigentes quanto ao quesito de disponibilidade. Assim, deseja-se que, a qualquer momento, o sistema esteja disponível;

- **Consistência** - Jogos multijogador consistem de um ambiente virtual compartilhado entre os jogadores, que disputam a mesma partida. Sendo assim, é necessário que aquele ambiente seja representado da mesma maneira nas máquinas dos diferentes jogadores para que estes possam interagir entre si e com o mundo de maneira adequada;

- **Segurança** - Em um sistema onde há dezenas de milhares de participantes, não é razoável confiar em todos eles, o que torna necessário algum mecanismo que proveja segurança. Segurança aqui se refere não somente a autenticação de usuários cadastrados, como também verificar se os estados/ações que os diferentes jogadores enviam são válidos. Jogadores trapaceiros, por exemplo, podem alterar o software executado em suas máquinas de forma que sejam enviados estados/ações que lhes dêem alguma vantagem no jogo;

- **Escalabilidade** - Levando-se em conta o número de participantes em um jogo MMOG, é necessário que o sistema que lhe dá suporte seja capaz de manter a qualidade do serviço a despeito da carga crescente de processamento e comunicação exigida para isso.

2.2 Trabalhos relacionados

A **arquitetura cliente-servidor** supre diversos aspectos necessários para a execução satisfatória de jogos do tipo MMOG, 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 online 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**, 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 online 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 par-a-par mais escaláveis, como (SCHIELE et al., 2007; RIECHE et al., 2007; HAMPEL; BOPP; HINN, 2006; EL RHALIBI; MERABTI, 2005; IIMURA; HAZEYAMA; KADOBAYASHI, 2004; KNUTSSON et al., 2004). Por exemplo, para reduzir o tráfego entre os pares, cada par pode enviar atualizações de estado apenas àqueles que tiverem interesse nas mesmas. Para atingir este objetivo, (SCHIELE et al., 2007) sugere que o ambiente virtual seja dividido em regiões. O objetivo disto é que cada região funcione como se fosse um jogo independente, de menor escala. Os jogadores cujos avatares estivessem em determinada região poderiam formar uma pequena rede par-a-par, e decidirem entre eles o andamento do jogo naquela área do mundo do jogo. Quando um avatar se movesse de uma região para outra, deixaria o grupo da primeira e se uniria ao grupo da sua nova região. Para que o ambiente seja contíguo, ou seja, para que avatares possam mover-se livremente entre regiões adjacentes, alguns dos pares mantêm conexões com pares das regiões vizinhas, e é através destes pares que se consegue as informações necessárias para que haja essa transferência de grupo.

Ainda na proposta de (SCHIELE et al., 2007), cada região tem um coordenador. O coordenador 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 o coordenador é confiável, o que não pode ser garantido, já que o software utilizado pelo jogador pode ter sido alterado de forma a se comportar de maneira incorreta a fim de beneficiá-lo. 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.

Existem também as propostas de **arquiteturas híbridas** (VILANOVA et al., 2008; CHEN; MUNTZ, 2006), que utilizam servidores ao mesmo tempo em que fazem uso de infra-estrutura par-a-par. Nestas propostas, pares e servidor dividem a simulação do jogo. Em (VILANOVA et al., 2008), o mundo do jogo é divido em espaço social e espaços de ação. O primeiro é voltado para interações sociais, como conversar, trocar objetos do mundo do jogo, formar grupos etc., que são ações que não impõem grande peso sobre o servidor. No entanto, se os jogadores quiserem, por exemplo, lutar – que é o principal objetivo na maioria dos jogos MMOG –, eles têm de requisitar ao servidor a criação de um espaço de ação, dentro do qual a interação é mais rápida e com pouca tolerância a atrasos de comunicação. Este espaço é uma *instância* do jogo, isolada do resto do mundo, dentro da qual um **um número limitado de jogadores** formam uma pequena rede par-a-par, sendo eles próprios responsáveis por simular o jogo e atualizar seus companheiros daquele espaço de ação. Para resolver inconsistências da simulação devido a falta de ordenação das mensagens, um dos jogadores é eleito como *super-peer*, sendo responsável por ordenar eventos que sejam "fortemente acoplados", cuja ordem interfere de maneira significativa no andamento do jogo. Essa abordagem apresenta alguns problemas, como o fato de não poder haver interações de luta, por exemplo, entre um grande número de jogadores. Além disso, nada garante que o par escolhido para ser o super-peer é con-

fiável. Se pertencer a um jogador desonesto, ele pode ordenar os eventos de maneira a beneficiar aquele jogador. No mais, o problema de sobrecarregar a banda dos pares é contornado simplesmente estabelecendo um limite para o número de participantes em cada grupo de ação, quando é desejável que nos MMOGs haja um grande número de jogadores interagindo entre si, não apenas em situações sociais, mas também em situações mais dinâmicas.

Já na proposta de (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 participantes 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 prover tolerância a falhas, os autores sugerem o uso dos agregados de pares - que consistem no gerenciador de região, mais pares "reserva" - eles recebem do gerenciador da região as ações recebidas de jogadores, que são processadas por estes "reservas", que mantêm uma réplica do estado do jogo contido no gerenciador daquela região. Além de prover robustez ao sistema, no caso do gerenciador sofrer colapso, estes nodos reserva podem detectar falhas na simulação e executar procedimentos de recuperação. Porém, tal abordagem não resolve o problema do par eleito para ser o gerenciadores da região não ser confiável, pois não há qualquer tipo de acordo entre ele e seus reservas em relação ao andamento do jogo. Por fim, se um nodo gerenciador estiver sobrecarregado, ele simplesmente devolve ao servidor parte da carga que lhe foi atribuída, mantendo a dependência de uma infra-estrutura centralizada.

Outra arquitetura híbrida é a do FreeMMG (CECIN et al., 2004). Nela, os pares se organizam de maneira par-a-par 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.

Já foram propostos alguns modelos que sugerem o uso de clientes conectados a um servidor distribuído. Em (ASSIOTIS; TZANOV, 2006), por exemplo, é proposta a divisão do mundo virtual em regiões menores, cada uma atribuída a um diferente nodo do sistema servidor. Os autores tratam a questão da consistência utilizando um mecanismo de travas – a cada entidade presente no jogo é associada uma trava, que precisa ser obtida antes de algum dos servidores fazer quaisquer alterações. Além disso, tentou-se atacar o problema dos *hotspots* (aglomerados de jogadores em uma área relativamente pequena do ambiente virtual) através do particionamento recursivo da área sobrecarregada, até que esta sobrecarga seja eliminada ou não houver mais servidores disponíveis. No entanto, há um limite para este reparticionamento, pois áreas muito pequenas implicam em avatares migrando entre regiões com maior freqüência e, conseqüentemente, em mais tráfego entre servidores.

Por fim, os autores desse trabalho, assim como em vários outros com propostas semelhantes (NG et al., 2002; CHERTOV; FAHMY, 2006; LEE; LEE, 2003), consideram que os nodos servidores estão conectados através de uma rede de alta velocidade e pouco

atraso, sendo que suas soluções são apenas parcialmente aplicáveis em um cenário com recursos altamente dinâmicos e voláteis, como são as redes par-a-par montadas com recursos de voluntários. Tais redes têm como problemas inerentes: baixa disponibilidade e dependabilidade, largura de banda escassa e baixo poder de processamento, quando comparada com servidores dedicados mantidos por empresas fabricantes de MMOGs.

2.3 Questões de pesquisa

Em um MMOG completamente descentralizado e montado sobre uma arquitetura par-a-par, cada par é responsável por decidir o resultado das ações de seu jogador. Apesar disto tornar o sistema menos dependente de cada um deles – se um jogador se desconectar do jogo, apenas o estado referente a seu avatar será perdido – e de tornar o jogo independente de servidores centrais, tal abordagem se mostrou muito vulnerável a trapaça, justamente por cada par executar parte da simulação, além dessa abordagem ser propícia à saturação da banda de upload dos participantes, que tem que enviar atualizações de estado da simulação para seus companheiros de jogo.

O que se pretende investigar é o uso de um sistema servidor composto de nodos voluntários distribuídos geograficamente. Esse sistema seria heterogêneo, onde cada nodo poderia ser desde um computador doméstico até um cluster pertencente a alguma empresa que tivesse interesse em servir o jogo. Seria exigida uma capacidade de processamento e link de comunicação mínimos para poder fazer parte do sistema servidor, além de ser necessário garantir de alguma forma que cada um desses nodos seria confiável. Porém, levando-se em consideração que o número de nodos servidores seria muito menor que o de jogadores, seria muito mais viável garantir a sua confiabilidade em relação à possibilidade de haver trapaça do que em uma rede par-a-par completamenet descentralizada, onde essa tarefa é praticamente impossível. Além do que os nodos servidores não pertenceriam necessariamente a jogadores. Poderiam ser pequenos provedores locais de serviço de Internet, por exemplo, que lucrariam ao disponibilizarem recursos para um jogo. Isso seria interessante para essas empresas, pois os recursos – pelo menos banda de comunicação – do provedor seriam utilizados de qualquer forma pelos jogos, pois todo o tráfego de jogadores que também fossem seus assinantes passaria por lá, independente de a qual servidor cada jogador estivesse conectado.

De qualquer forma, como a simulação seria distribuída entre os nodos servidores, não seria necessário que cada um destes tivesse a mesma capacidade de processamento ou comunicação possuída pelos grandes servidores centrais usados comumente para jogos MMOG. Para fazer essa distribuição, o mundo do jogo poderia ser dividido em regiões, cada uma gerenciada por, pelo menos, um destes nodos. Mais de um nodo servidor deverá ser usado por região, de maneira que a simulação seria distribuída em dois níveis: primeiro em regiões, depois dentro de cada uma delas, entre os nodos que a estivessem gerenciando-a. Dessa forma, cada jogador enviaria suas ações a um destes nodos, que as processaria, em seguida se sincronizaria com os outros nodos da mesma região, e por fim, persistiria de maneira distribuída o novo estado do jogo e enviaria para os jogadores a ele conectados. Assim, também seria provida melhor disponibilidade, pois cada região do mundo virtual estaria sendo servida por um conjunto de servidores com estado do jogo sincronizado. Caso algum destes falhasse, o(s) outro(s) poderia(m) tomar para si os jogadores que estavam conectados ao servidor que não está mais disponível.

Para atingir o objetivo de distribuir um servidor de MMOG sobre uma rede dinâmica, com recursos voláteis, mais escassos, e com menor dependabilidade, como são as redes

par-a-par formadas por nodos voluntários, surgem algumas questões que ou tem sido objeto de pesquisa nos últimos anos, ou que ainda estão por resolver. Algumas delas são:

- Encontrar uma maneira de particionar o espaço virtual, mapeando cada região a um conjunto de nodos servidores, 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, além de tratar a questão dos *hotspots* (aglomerados de avatares em uma área relativamente pequena);
- Manter consistente a simulação nas diferentes regiões, entre servidores da mesma região, e de servidores em regiões adjacentes;
- Coordenar a entrada, a saída e a detecção de colapso de nodos do sistema, provendo tolerância a falhas;
- Distribuir e armazenar de maneira persistente o estado do mundo e dos jogadores, de forma a poder recuperá-lo sempre que 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.

Pretende-se, durante o curso de doutorado, pesquisar e desenvolver novas soluções para estes problemas, buscando as que forem mais adequadas em um contexto de jogos online maciçamente multijogador. Com tais questões resolvidas, poderão ser providos a MMOGs disponibilidade e tolerância a falhas – devido a replicação de nodos servidores em uma mesma região – e escalabilidade, mantendo o atraso de comunicação dentro de um limite aceitável para o jogo, tudo isso por um baixo custo.

3 OBJETIVO

3.1 Objetivo geral

O objetivo é criar um modelo arquitetural de suporte a jogos online maciçamente multijogador. Seguindo este modelo, deverá ser possível formar um sistema geograficamente distribuído de nodos servidores, utilizando a Internet como meio de ligação entre os mesmos. Essa distribuição deverá ser transparente aos jogadores, que verão o sistema como um único servidor de MMOGs, que conseguirá manter o mundo do jogo consistente, disponível e com boa qualidade de serviço.

3.2 Objetivos específicos

Para atingir este objetivo geral, algumas metas foram traçadas:

1. Desenvolver um mecanismo que lide com entrada, saída e colapso destes, provendo tolerância a falhas, ao mesmo tempo em que busca manter próximos na rede *overlay* nodos servidores que estejam próximos geograficamente;

2. Criar um algoritmo de balanceamento da carga do MMOG, que ao mesmo tempo em que mantém a carga em cada servidor proporcional à sua capacidade, leva em consideração a existência de *hotspots*, evitando que algum destes seja particionado entre regiões diferentes, dado que isto causaria forte dependência entre elas;

3. Definir um esquema de sincronização do estado da simulação entre os diferentes nodos servidores designados para a mesma região do mundo do jogo, assim como sincronizar periodicamente, ou quando necessário, os estados das diferentes regiões, de forma a manter consistência global;

4. Definir um mecanismo de persistência distribuída dos estados dos jogadores e do mundo virtual do jogo como um todo, de maneira que estejam sempre disponíveis para qualquer nodo servidor, ainda que alguns destes inesperadamente se tornem indisponíveis;

5. Pesquisar e desenvolver técnicas e algoritmos que visem a reduzir o tráfego entre clientes e servidores e entre servidores, de maneira a economizar recursos dos nodos servidores, diminuir atraso na comunicação entre eles e ampliar o público do jogo devido à redução do requisito de largura de banda para jogá-lo, além de diminuir os requisitos para que um computador possa fazer parte do sistema servidor, porém sem prejudicar a experiência dos jogadores.

4 MODELO PRELIMINAR: COSMMUS

Como parte do mestrado, definiu-se um modelo preliminar, que visa justamente ao particionamento do ambiente virtual em regiões, cada uma atribuída a um servidor, sobre o qual foi definido um esquema de balanceamento de carga. O Cosmmus – *Cooperative System for Massive Multiplayer Service* – será uma extensão desse modelo, a ser definida durante o doutorado. No entanto, já há algumas idéias de como poderá ser este modelo.

4.1 Distribuição da carga do jogo

Já foram feitos alguns trabalhos relacionados ao particionamento do ambiente virtual do jogo (DE VLEESCHAUWER et al., 2005; MORILLO; FERNÁNDEZ; ORDUÑA, 2006; LUI; CHAN, 2002). Para o Cosmmus, pensa-se em fazer a distribuição da carga do jogo entre os nodos servidores em dois níveis:

1. Assim como no mestrado, será utilizada a idéia de células de mesmo tamanho e posição fixa, que são áreas do ambiente. Essas células então são agrupadas em regiões, cujo tamanho e formato irá depender da maneira como os avatares dos jogadores estão mais ou menos concentrados em determinados lugares. Isto deverá ser feito de maneira transparente para o jogador, que verá um mundo grande e contíguo através do qual poderá mover-se livremente;

2. Cada região poderá ser designada a mais de um nodo servidor, com dois objetivos principais: primeiramente, prover tolerância a falhas por meio de replicação e, em segundo lugar, permitir que regiões que já estão muito sobrecarregadas sejam gerenciadas por mais de um nodo servidor.

No caso de uma região estar muito sobrecarregada, poderia-se pensar em fazer um novo particionamento, em duas ou mais novas regiões, mas isto poderia ser contraproducente em determinadas situações, como no caso em que a região engloba um hotspot e sua divisão resultaria em duas regiões excessivamente dependentes entre si. Para isso deverá ser criado também um algoritmo, métrica ou heurística que determine se uma região deve ou não ser subdividida em novas regiões.

Foi proposto durante o mestrado um esquema de balanceamento de carga dinâmico, baseado em particionamento de grafo e refinamento de suas partições (**?**). O mundo é dividido em células de mesmo tamanho e posição fixa, cada uma representada por um vértice em um grafo. A interação (medida em kilobytes por segundo) entre as células são representadas pelas arestas dos vértices. O algoritmo consiste no particionamento do grafo, sendo que cada partição representa uma região a ser atribuída a um servidor. Tal balanceamento leva em conta diferentes níveis de recursos entre os diferentes servidores

e leva em conta as necessidades dos MMOGs, que se concentram com maior peso no uso de largura de banda dos servidores.

Também seria necessário um esquema de balanceamento de carga, que, pretende-se, será semelhante ao proposto durante o mestrado. A questão é que, diferente do modelo do mestrado, não haverá mais apenas um nodo servidor por região, mas um grupo de nodos servidores. Neste caso, serão investigadas maneiras de balancear a carga do sistema, considerando essa distribuição em dois níveis. Uma idéia seria a de usar apenas réplicas, com o intuito de prover tolerância a falhas na região. A carga dentro de uma região seria redistribuída apenas quando a mesma estivesse sobrecarregada e nao fosse viável reparticioná-la.

4.2 Formação da rede lógica do sistema servidor

Quanto à formação da rede overlay sobre a qual será executado o sistema servidor, existem propostas, como em (**??**), que buscam montar a rede de forma que nodos com boa conexão de rede entre si estejam adjacentes na rede lógica. Foi proposto, por exemplo, agrupar os nodos de acordo com o atraso de rede entre eles. Pretende-se utilizar estas técnicas de maneira que certos nodos servidores servindo regiões adjacentes e, principalmente, nodos que estejam servindo a mesma região, pertençam ao mesmo grupo. Pode-se também adaptá-las, incluindo outros critérios além do atraso, como a velocidade da transmissão de dados entre eles, em bits por segundo.

O propósito de montar a rede overlay seguindo essa estratégia seria o de minimizar o atraso na interação entre jogadores. Essa atraso na interação seria o tempo decorrido desde o envio de uma ação de um jogador, processamento pelo servidor e recebimento do estado resultante por outro jogador. Se dois jogadores quaisquer estiverem interagindo um com o outro, porém conectados a nodos servidores diferentes, será necessário que cada uma de suas interações seja intermediada pelos dois servidores. Assim, se estes dois servidores tiverem baixo atraso entre eles, a interação dos jogadores será menos prejudica pelo acréscimo de um segundo servidor intermediário.

4.3 Sincronização de estado

Uma das questões centrais que o Cosmmus deverá tratar é a da manutenção da consistência da simulação entre os diferentes nodos servidores. Ainda que cada jogador veja apenas uma parte do mundo virtual, este mundo deve ser o mesmo para todos os servidores. Assim, se dois jogadores estão com seus avatares no mesmo lugar do ambiente virtual, eles deverão visualizar o mesmo ambiente. Outra questão importante é a ordenação dos eventos. Supondo que dois jogadores estivessem lutando no jogo, e o avatar de um deles atirasse com uma arma qualquer e o outro se movesse, a ordem desses eventos seria crucial para decidir se o outro jogador teve seu avatar morto, apenas ferido ou se evitou completamente o disparo.

No caso em que os dois jogadores estivessem conectados ao mesmo servidor, como é tradicionalmente feito por empresas de jogos, isto seria trivial, bastando verificar qual das ações chegou primeiro. Ainda que se utilizasse um mecanismo de compensação de atraso – o *timestamp* da ação de cada jogador seria igual ao tempo atual menos o seu atraso –, como é feito em Half-Life (**?**), por exemplo, não seria muito problemático ordenar estes eventos. No caso de um servidor distribuído, seria necessário que os nodos servidores entrassem em algum tipo de acordo ou sincronia no que diz respeito a essa ordenação,

cuja complexidade cresceria com o número de servidores envolvidos.

Levando em conta que há várias regiões formando um mundo contíguo, e que em cada uma existe um ou mais servidores, estas questões referentes a manter a consistência da simulação se tornam ainda mais difíceis quando um avatar se aproxima de uma fronteira da região onde está localizado. Espera-se que o espaço seja contíguo, ou seja, que a divisão do ambiente não seja percebido pelo jogador. Então, ao se aproximar de uma fronteira, ele deverá ser capaz de ver tudo que estiver além dela. Para isto, será necessário que o servidor ao qual ele está conectado comunique-se com aquele que serve a região, além da fronteira, que está sendo visualizada pelo jogador. Serão transmitidas informações como: estado do ambiente, avatares que estejam próximos, estados desses avatares e assim por diante.

Em um trabalho do GPPD (**?**), foi proposto um esquema de sincronização de estado que consistem em dois simuladores: um conservador e um otimista. Enquanto que o otimista executa as ações dos jogadores assim que são recebidas, o simulador conservador divide o tempo em turnos e somente executa as ações dos jogadores após haver uma sincronização. Esta sincronização consiste em que cada nodo executando a simulação receba as ações de todos os participantes envolvidos – ou estes notificaram que não fariam nada –, que foram em seguida ordenadas e processadas. Somente após este passo é que o simulador conservador avança para o próximo turno. No caso da simulação otimista discrepar além do tolerável da simulação conservadora, esta sobrescreve o seu estado sobre a primeira, que continua a partir daí. No entanto, este esquema foi proposto para uma arquitetura completamente descentralizada, sendo necessário analizá-lo e fazer as adaptações necessárias para que possa ser utilizado em um sistema servidor distribuído em dois níveis, como se pretende para o Cosmmus. Existem também outros trabalhos que visam a manter consistente a simulação, como (**?**), que serão também estudados e possivelmente alguns de seus princípios farão parte da solução para consistência a ser definida para o modelo final.

4.4 Persistência distribuída

No que tange ao armazenamento do estado do mundo virtual e dos jogadores, deverá haver uma maneira de fazer isto de maneira distribuída, já que não haverá um único servidor central. Poderia ser mais simples escolher um dos nodos servidores para guardar estas informações, no entanto isso poderia fazer com que o estado do jogo dependesse da disponibilidade e confiabilidade deste nodo escolhido e, ainda que estes dois requisitos fossem satisfeitos, este servidor poderia ser saturado pelo número de requisições, tornando mais lento a recuperação dos estados nele armazenados.

O Cosmmus deverá então utilizar algum esquema de armazenamento distribuído, onde porções da informação de estado são armazenados em diferentes lugares. Além disso, deverá ser utilizada replicação para que o estado esteja disponível ainda que alguns nodos do sistema se desconectem inesperadamente. Redes lógicas baseadas em DHT, como Pastry (**?**), Tapestry (**?**), Chord (**?**) e Koorde (**?**), parecem ser possíveis soluções para prover o tipo de armazenamento distribuído e redundante que se quer para o estado do jogo. Pretende-se, portanto, investigar o quanto essas redes são realmente adequadas para o armazenamento distribuído dos estados do jogo. Existem simuladores, tais como o OverSim (**?**), que permitem a simulação de redes overlay, tais como a Pastry e, através dele, será avaliado o comportamento do sistema utilizando essas redes lógicas.

O que pode ocorrer é que seja excessivo o custo de manter uma rede como a Pastry

sobre o sistema servidor, recuperando e armazenando os estados utilizando multicast no nível de aplicação e utilizando replicação, se somado ao próprio custo que os nodos servidores já enfrentarão para executar a simulação em si, recebendo ações dos jogadores e devolvendo-lhes a resposta. Para não sobrecarregar o sistema com o tráfego referente à persistência do estado do jogo, poderiam ser ajustados parâmetros, tais como freqüência de atualização e, no nível da aplicação, o nível de detalhamento do estado a ser persistido. Por exemplo, ao invés de armazenar a localização, direção e ação sendo executada por cada avatar em uma área do ambiente virtual, poderia ser armazenada apenas sua localização.

4.5 Otimização do uso de largura de banda

Pelo fato dos nodos servidores serem, idealmente, de baixo custo, é necessário economizar seus recursos tanto quanto for possível, mas sem prejudicar a qualidade do jogo. Por exemplo, podem ser feitas filtragens por interesse, onde a cada jogador só é enviado aquilo que ele pode visualizar, de forma a economizar tráfego tanto entre clientes e servidores, quanto entre os próprios servidores. Seguindo esse mesmo princípio, duas regiões podem ter uma versão antiga do estado uma da outra, se não estiver havendo interação entre jogadores presentes nela. Dessa forma, os servidores a elas designados não precisarão atualizar-se um ao outro com tanta freqüência.

Periodicamente, ou no momento em que for necessário (e.g. dois jogadores de diferentes regiões interagindo próximo a fronteira), poderão ser trocados os estados das regiões. A granularidade dessa informação pode ser ainda mais fina, e cada jogador poderia enviar para o outro apenas a informação de que ele precisa. Por exemplo, cada servidor enviaria ao outro apenas o estado do ambiente virtual próximo à fronteira entre as regiões, ao invés do estado da região inteira.

Diversos propostas já foram realizados com o intuito de fazer filtragem por interesse (**??**). Além de investigar qual a que melhor se aplica ao modelo proposto, além de verificar possíveis adaptações que sejam necessárias, outras maneiras de economizar os recursos do sistema servidor serão investigadas.

4.6 Questões filosóficas

Apesar de, geralmente, ser necessário uma infra-estrutura central poderosa e cara (**?**), um sistema que siga a proposta aqui apresentada permitiria que recursos de empresas menores e usuários domésticos (com o mínimo exigido de capacidade de processamento e largura de banda, o que irá depender do jogo específico) fossem utilizados para formar um sistema para servir um jogo online maciçamente multijogador com qualidade comparável à dos MMOGs mais populares.

O Cosmmus tem como idéia fundamental formar um sistema servidor sobre uma rede par-a-par, ao invés de distribuir indiscriminadamente a simulação entre quaisquer jogadores, como são diversas propostas pesquisadas. Existindo essa separação, onde somente os servidores poderiam arbitrar sobre o jogo, poderiam ser formadas comunidades cujo único intuito seria o de manter jogos MMOGs funcionando, organizando quem seria autorizado a participar do sistema como um nodo servidor e assim por diante. Poderia haver um esquema semelhante ao utilizado em diversos projetos de software livre, na Wikipédia (**?**) e outros, onde a pessoa que teve a idéia inicial vai gradualmente delegando autoridade a membros da comunidade, de acordo com o quanto cada um colaborou,

criando uma hierarquia, onde há diferentes níveis de permissão, mas todos participam em um esforço comum.

Além dessa questão filosófica, para um servidor de MMOGs, que age como árbitro central e por onde toda a simulação tem que passar, é mais fácil detectar e impedir trapaça do que em um MMOG com a simulação completamente descentralizada, onde o computador de cada jogador decide em parte o que irá acontecer como resultado das ações no jogo. No entanto, mesmo em uma rede colaborativa como a que se tem em mente nesta proposta, não há garantias de que alguns dos integrantes não irão tentar subverter as regras do jogos, obtendo vantagens de forma injusta. Porém, imagina-se um sistema de reputação. Ao ingressar no sistema servidor, cada participante terá um registro de como têm realizado a simulação do jogo, armazenado em algum local confiável.

A princípio, seria designada ao novo servidor uma porção do mundo que seja considerada de importância mínima, de tal modo que tentativas de subversão por parte dele causarão um dano muito pequeno ao jogo, se causarem. Tanto jogadores, quanto outros nodos servidores, poderão reportar casos como esse, de forma a punir ou banir o nodo que agiu de maneira maliciosa. Integrantes honestos do sistema servidor irão ganhando pontos de reputação com o passar do tempo, podendo administrar porções e regiões do mundo do jogo cada vez mais importantes. Acredita-se que seja desmotivadora o suficiente a idéia de destruir a reputação conseguida após um longo tempo de colaboração para o jogo por causa de uma tentativa de trapaça.

Dessa maneira, diversos MMOGs poderiam surgir e se popularizar com mais facilidade, abrindo campo para empresas pequenas e grupos independentes de desenvolvimento de jogos. A cada jogo estaria associada uma comunidade, que se auto-regularia. O nível de justiça ou trapaça em cada um deles iria depender de quem fossem os responsáveis. Comunidades bem organizadas poderiam fazer com que um jogo fosse tão justo quanto um administrado por provedores com contratos assinados, sob pena de multa no caso de tentativas de subversão por parte do servidor, por exemplo.

5 PLANO DE TRABALHO

Este capítulo do plano de curso de doutorado tem como propósito detalhar os passos que vão se seguir para atingir os objetivos definidos anteriormente, desde a metodologia pretendida até o cronograma e planejamento dos créditos.

5.1 Metodologia

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

1. **Exigências do doutorado**: requisitos impostos pelo programa de pós-graduação em computação da UFRGS ao estudante de doutorado;

   (a) Revalidação dos 24 créditos cursados durante o mestrado;

   (b) Cumprimento dos créditos restantes exigidos para o doutorado;

   (c) Realização de atividade didática referente ao estágio-docência;

   (d) Apresentação de seminário de andamento na Semana Acadêmica do PPGC;

   (e) Realização do exame de qualificação em abrangência;

   (f) Apresentação da proposta de tese;

2. **Revisão bibliográfica**: tem por objetivo expandir o domínio sobre o estado da arte relacionado a distribuição de jogos MMOG;

   (a) Estudo de tecnologias de formação de redes overlay e verificar sua compatibilidade com o tipo de gerenciamento de entrada, saída e colapso de nodos servidores pretendido;

   (b) Pesquisa de artigos que propõem técnicas visando a colocar os servidores distribuídos na rede overlay, levando em consideração a qualidade da comunicação entre eles (i.e. servidores próximos geograficamente tenderão a ser adjacentes na rede overlay);

   (c) Análise de algoritmos de balanceamento de carga, no que se refere à sua aplicabilidade em um servidor distribuído de MMOGs, de maneira a determinar o que se precisaria modificar para adaptar ao contexto de jogos;

   (d) Pesquisa de trabalhos que têm por objetivo a descoberta e gerenciamento de hotspots, de forma que sua existência não degrade a qualidade de serviço do jogo;

(e) Revisão de técnicas para sincronizar o estado do jogo entre diferentes servidores de maneira dinâmica, sem que isso gere tráfego excessivo no sistema servidor;

(f) Pesquisar mecanismos de persistência distribuída, consistente e tolerante a falhas que possam ser utilizados para armazenar os estados dos jogadores e do ambiente virtual;

(g) Estudo sobre técnicas existentes para reduzir o tráfego entre os clientes e os servidores e entre os servidores, de maneira a reduzir custos sem prejudicar o jogo;

3. **Definição do modelo**: tem como objetivo definir os componentes necessários à arquitetura para que seja capaz de fornecer o tipo de serviço pretendido;

   (a) Definição do tipo de topologia que a rede overlay deverá ter e de qual tecnologia será utilizada/adaptada para o gerenciamento desta rede;

   (b) Definição dos critérios utilizados para efetuar balanceamento de carga e como isso será feito;

   (c) Definição do esquema de sincronização entre os nodos servidores;

   (d) Definição do modelo de persistência utilizado para armazenar de maneira distribuída e tolerante a falhas o estado dos mundo virtual e dos jogadores;

   (e) Integração do modelo definido com técnicas de otimização do uso de largura de banda e de redução de atraso na comunicação;

   (f) Cooperação internacional com outro grupo de pesquisa (sanduíche ou cotutela).

   Foram feitos contatos durante o DS-RT, em Vancouver, com professores do SITE – *School of Information Technology and Engineering*, uOttawa (Ottawa, Canadá) –, havendo grande interesse por parte destes, e negociações já estão em andamento, faltando alguns detalhes para que isso seja possível. Outra instituição com possibilidade de cooperação é a Universidade de Lugano, na Suíça.

4. **Implementação do modelo**: visa a criar um protótipo que possa ser utilizado tanto em simulações quanto em execuções reais do modelo proposto;

   (a) Implementação da camada de gerenciamento da malha de conexões entre os nodos servidores e entre servidores e clientes;

   (b) Implementação do módulo de distribuição de carga e gerenciamento de hotspots;

   (c) Implementação da camada de sincronização da simulação e persistência de estados;

   (d) Criação e integração do módulo de gerenciamento do uso de recursos do nodo e adaptar a qualidade do jogo à capacidade de cada servidor;

5. **Simulação e testes**: visa validar o modelo proposto e verificar se os objetivos foram alcançados;

(a) Análise e comparação de simuladores de rede e de sistemas em larga escala, um dos quais será escolhido para realização das simulações do modelo proposto de suporte distribuído a MMOGs;

(b) Adaptação do protótipo para que execute sobre o simulador escolhido;

(c) Execução dos testes com respeito a: qualidade da comunicação entre nodos adjacentes, eficiência do balanceamento de carga e detecção dos hotspots, consistência da simulação entre os diferentes nodos servidores, disponibilidade do jogo e dos estados dos jogadores mesmo com a presença de falhas;

(d) Testar o protótipo, sobre um sistema em larga escala real, como o Grid5000 (**?**) e analisar os resultados;

6. **Escrita do trabalho**: tem por objetivo documentar o trabalho realizado, por meio da tese, além de divulgá-lo, através da escrita e publicação de artigos;

   (a) Escrita da tese;

   (b) Escrita de artigos para publicação nacional (SBGames) e internacional (DS-RT, NetGames) e periódicos internacionais (*Multimedia Tools and Applications*).

5.2 Planejamento dos créditos

De forma a cumprir os créditos exigidos pelo curso de doutorado, pretende-se:

1. Revalidar os 24 créditos obtidos durante o curso de mestrado acadêmico no Programa de Pós Graduação em Computação do Instituto de Informática da UFRGS (PPGC/UFRGS). As disciplinas cursadas estão sumarizadas na Tabela 5.1;

2. Obter 16 créditos adicionais, através do curso das disciplinas CMP182 - Redes de Computadores I B, CMP147 - Redes de Computadores II, CMP200 - Criptografia de Dados e CMP260 - Redes P2P: características, protocolos e aplicações;

3. Obter mais 3 créditos, referentes a um trabalho individual e a realização de atividade didática (Tabela 5.2).

5.3 Cronograma

A Tabela 5.3 apresenta o cronograma de atividades a serem desenvolvidas, conforme listadas na seção de metodologia.

Tabela 5.1: Créditos a revalidar

Código	Disciplina	Créditos	Conceito
CMP139	Tolerância a Falhas em Sistemas Distribuídos	4	A
CMP255	Prog. Paralela para Processamento de Alto Desempenho	4	A
CMP167	Programação com Objetos Distribuídos B	4	A
CMP256	Proj. de Prog. Paralelos para Proc. de Alto Desempenho	4	A
CMP157	Programação Paralela e Distribuída	4	A
CMP250	Aspectos Conceituais de Tolerância a Falhas	1	B
CMP401	Trabalho Individual I	2	A
CMP410	Atividade Didática I	1	A

Tabela 5.2: Créditos a obter

Código	Disciplina	Créditos
CMP182	Redes de Computadores I B	4
CMP147	Redes de Computadores II	4
CMP200	Criptografia de Dados	4
CMP260	Redes P2P: características, protocolos e aplicações	4
CMP40(X)	Trabalho Individual	2
CMP411	Atividade Didática II	1

Tabela 5.3: Cronograma

Etapa	2009/1	2009/2	2010/1	2010/2	2011/1	2011/2	2012/1	2012/2
1a	X							
1b	X	X						
1c		X						
1d			X		X		X	
1e			X					
1f				X				
2a	X							
2b	X							
2c	X							
2d	X							
2e		X						
2f		X						
2g		X	X					
3a		X						
3b		X						
3c			X					
3d			X					
3e			X					
3f		X	X					
4a				X				
4b				X				
4c					X			
4d					X			
5a						X		
5b						X		
5c						X		
5d						X	X	
6a							X	X
6b			X	X	X	X		

6 INFORMAÇÕES DO CANDIDATO

6.1 Dados de identificação

6.2 Experiência com pesquisa

Antes de realizar o mestrado acadêmico, o candidato já havia trabalhado com pesquisa e desenvolvimento, no grupo indigente (INDIGENTE, 2004), sendo um dos idealizadores e desenvolvedor do motor para jogos InGE – *indigente game engine* – (**?**), tendo sido o único responsável pelo módulo de rede. Posteriormente, foi um dos desenvolvedores do jogo *We Are The Campignons*, que obteve premiação em um festival de jogos de âmbito nacional.

O curso de doutorado pretendido terá por objetivo dar continuidade à pesquisa iniciada no mestrado acadêmico pelo PPGC/UFRGS, entre os anos de 2007 e 2009, sendo a defesa da dissertação prevista para fevereiro ou março. Durante o mestrado, foram propostas algumas técnicas que visam a otimizar o uso de largura de banda em um sistema distribuído atuando como servidor para um jogo online maciçamente jogador. Uma destas técnicas consiste no gerenciamento de interesse do jogador, levando em conta diferentes níveis de relevância para diferentes entidades do jogo. Outra técnica proposta é a de balanceamento de carga dinâmico, baseado em particionamento de grafo e refinamento de suas partições (**?**). Este trabalho está em fase de conclusão, faltando coletar os resultados e escrever artigo para publicar no journal *Multimedia Tools and Applications*, da Springer, que terá uma edição especial a respeito de MMOGs.

O candidato também participou do projeto P2PSE (VILANOVA et al., 2008), sendo um dos responsáveis pelo desenvolvimento da funcionalidade de multi-servidor, utilizando linguagem de programação C++. Também é o responsável pelas simulações de rede, realizadas com o auxílio do simulador ns-2 (**?**), presentes no artigo publicado em simpósio nacional a respeito do projeto.

Tais trabalhos desenvolvidos durante o mestrado, além do envolvimento anterior com pesquisa e desenvolvimento, renderam algumas publicações e uma premiação:

- Full paper, em simpósio internacional Qualis A:

  A^3: a novel interest management algorithm for distributed simulations of MMOGs, BEZERRA, Carlos Eduardo B.; CECIN, Fábio R.; GEYER, Cláudio F. R., em *Proc. of the 12-th IEEE International Symposium on Distributed Simulation and Real Time Applications*, 2008, Vancouver, BC, Canada.

- Full paper, em simpósio nacional:

P2PSE - A peer-to-peer support for multiplayer games, VILANOVA, Felipe J.; BEZERRA, Carlos Eduardo B.; CRIPPA, Marcos R., CECIN, Fábio R.; GEYER, Cláudio F. R., em *Proc. of SBGames 2008 - VII Brazilian Symposium on Computer Games and Digital Entertainment - Computing Track*, 2008, Belo Horizonte, MG, Brazil.

- Short paper, em simpósio nacional, a respeito do desenvolvimento do InGE:

  Desenvolvendo um Motor Multiplataforma para Jogos 3D, ROCHA, Rodrigo; BESSA, Aline; BEZERRA, Carlos E. B.; MEDEIROS, Ivan; OLIVEIRA, Caio; BANDEIRA, H., em *Anais do SBGames 2007 - VI Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital - Trilha Computação*, 2007, São Leopoldo, RS, Brazil.

- 3º lugar no Festival de Jogos Independentes do SBGames 2006:

  We Are The Champignons, *V Simpósio Brasileiro de Jogos para Computador e Entretenimento Digital*, 2006, Recife, PE, Brasil.

Para o curso de doutorado, o candidato pretende-se criar um modelo mais detalhado que cubra os requisitos definidos para o sistema servidor distribuído. Pelo menos devem ser satisfeitos os requisitos de distribuição da persistência do estado dos jogadores e do mundo virtual, da sincronização da simulação distribuída entre os nodos servidores, provendo também tolerância a falhas, além de desenvolver técnicas que otimizem tanto quanto possível o uso de largura de banda, evitando que a qualidade do jogo seja prejudicada.

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.

BEZERRA, C. E. B.; CECIN, F. R.; GEYER, C. F. R. A3: a novel interest management algorithm for distributed simulations of mmogs. **Proceedings of the 12th IEEE International Symposium on Distributed Simulation and Real Time Applications, 2008, Vancouver, BC, Canada**, [S.l.], p.35–42, 2008.

BLIZZARD. **Starcraft**: brood war. http://www.blizzard.com/broodwar/.

BLIZZARD. **Diablo 2**. http://www.idsoftware.com/games/quake/.

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

CCP. **EVE Online**. http://www.eve-online.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.

CHERTOV, R.; FAHMY, S. Optimistic Load Balancing in a Distributed Virtual Environment. **Proceedings of the 16th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)**, [S.l.], 2006.

DARLAGIANNIS, V.; HECKMANN, O.; STEINMETZ, R. Peer-to-peer applications beyond file sharing: overlay network requirements and solutions. **e & i Elektrotechnik und Informationstechnik**, [S.l.], v.123, n.6, p.242–250, 2006.

DE VLEESCHAUWER, B.; VAN DEN BOSSCHE, B.; VERDICKT, T.; DE TURCK, F.; DHOEDT, B.; DEMEESTER, P. Dynamic microcell assignment for massively multiplayer online gaming. **Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games**, [S.l.], p.1–7, 2005.

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.

HAMPEL, T.; BOPP, T.; HINN, R. A peer-to-peer architecture for massive multiplayer online games. **Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games**, [S.l.], 2006.

IBM. **Butterfly.net: powering next-generation gaming with on-demand computing**. 2003.

id Software. **Quake**. http://www.idsoftware.com/games/quake/.

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/.

KNUTSSON, B.; LU, H.; XU, W.; HOPKINS, B. Peer-to-peer support for massively multiplayer games. **IEEE Infocom**, [S.l.], 2004.

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.

LUI, J.; CHAN, M. An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems. **IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS**, [S.l.], p.193–211, 2002.

MORILLO, P.; FERNÁNDEZ, M.; ORDUÑA, J. M. Solving the Partitioning Problem in Distributed Virtual Environment Systems Using Evolutive Algorithms. **Handbook of bioinspired algorithms**, [S.l.], p.531–554, 2006.

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

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

RIECHE, S.; FOUQUET, M.; NIEDERMAYER, H.; PETRAK, L.; WEHRLE, K.; CARLE, G. Peer-to-Peer-based Infrastructure Support for Massively Multiplayer Online Games. **Consumer Communications and Networking Conference, 2007. CCNC 2007. 2007 4th IEEE**, [S.l.], p.763–767, 2007.

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.

TRAVIAN. **Travian**. http://www.travian.com/.

VALVE. **Counter-Strike**. http://www.counter-strike.net/.

VILANOVA, F. J.; BEZERRA, C. E. B.; CRIPPA, M. R.; CECIN, F. R.; GEYER, C. F. R. P2PSE - A Peer-to-Peer Support for Multiplayer Games. **VII Brazilian Symposium on Computer Games and Digital Entertainment - Computing Track, 2008, Belo Horizonte, MG, Brazil**, [S.l.], p.47–53, 2008.

WANG, T.; WANG, C.; LAU, F. **A Grid-enabled Multi-server Network Game Architecture**. 2004. 26–27p.

ASSINATURAS

_____

Carlos Eduardo Benevides Bezerra

_____

Prof. Dr. Cláudio Fernando Resin Geyer