

389604

Aplicações dos computadores -
Simulações
Sistemas: Tempo real
Jogos: Estratégia

FreeMMG: Uma Proposta Baseada em Peer-to-Peer para Simulação Interativa Competitiva em Tempo Real e Distribuída em Grande Escala

ENPq 1.03.04.00-2

Fábio Reis Cecin¹
Cláudio Fernando Resin Geyer²

Resumo

Este artigo trata do problema do custo associado ao projeto e implementação de jogos com grande número de usuários, ou MMGs (Massively Multiplayer Games). Mostra-se que o paradigma de sistemas *Peer-to-Peer* pode ser empregado na implementação de MMGs, reduzindo custos. Por fim, propõe-se o *framework* FreeMMG, uma infra-estrutura *Peer-to-Peer* para MMGs.

1 Introdução

Recentemente, profissionais do ramo de entretenimento eletrônico, ligados tanto à indústria quanto à academia, vêm juntando esforços no sentido de estabelecer fóruns de cunho científico para a área (DIGRA, 2002; JOGD, 2002; GDC, 2002). Esta indústria atualmente produz, com sucesso comercial, sistemas de jogo on-line denominados MMGs (Massively Multiplayer Games) (ALEXANDER, 2003). Estes sistemas, como Ultima Online (UO, 2002) e EverQuest (EVERQUEST, 2002), suportam um grande número de participantes (tipicamente entre 1 mil e 1 milhão) e permitem uma interação competitiva e em tempo real entre os mesmos.

A pesquisa em Modelagem e Simulação Computacional abrange diversas sub-áreas, como a Simulação Distribuída e sua respectiva sub-área, a Simulação Distribuída Interativa em Tempo Real (DS-RT'02, 2002). Neste artigo, afirmamos que a implementação de MMGs é uma especialização do problema da Simulação Distribuída Interativa em Tempo Real, sendo

¹fcecin@inf.ufrgs.br Bolsista CAPES
²geyer@inf.ufrgs.br

diferenciada pelo suporte a um grande número de participantes (distribuição em larga escala) e pelo suporte à competição justa entre os mesmos.

Este artigo está organizado em oito seções. A segunda seção define o problema da Simulação Interativa Competitiva em Tempo Real e Distribuída em Grande Escala. A terceira seção discute o tratamento atualmente dado a este problema na implementação de MMGs. A quarta seção explora a adoção do paradigma *Peer-to-Peer* como forma de redução dos custos associados aos MMGs. A quinta seção defende a Simulação Replicada como uma solução *Peer-to-Peer* para jogos de estratégia em tempo real distribuídos em pequena escala. A sexta seção apresenta o *framework* FreeMMG, uma extensão da proposta de Simulação Replicada que suporta distribuição em grande escala. A sétima seção aponta trabalhos relacionados. A oitava seção apresenta conclusões.

2 Problema

Em geral, a pesquisa em Simulação Distribuída Interativa em Tempo Real não supre totalmente as necessidades dos desenvolvedores de MMGs, pois esta linha de pesquisa não trata explicitamente da questão fundamental da competição entre participantes anônimos e a conseqüente necessidade de prevenção, detecção e correção de trapaças. Trabalhos recentes propõe tratamento para o problema da trapaça em jogos distribuídos (BAUGHMAN; LEVINE, 2001), alguns atacando também o problema da distribuição em grande escala (CUNNIN, 2000).

Neste trabalho, portanto, atacamos o problema da Simulação Interativa Competitiva em Tempo Real e Distribuída em Grande Escala. O termo *Simulação Interativa e Competitiva* será abreviado para *jogo*. Por *jogo* entende-se uma aplicação onde há interação e competição entre os participantes e, portanto, há participantes que estão dispostos a trapaçar. A distribuição em grande escala será abreviada pelo termo *MMG: Massively Multiplayer Game* ou jogo com grande número de participantes.

3 Soluções Atuais

Atualmente utilizam-se variantes do paradigma cliente-servidor para tratar do problema. Esta solução resolve o problema da trapaça à medida em que a cópia oficial do estado da simulação está no lado servidor, que só é acessível aos clientes (participantes) através de um protocolo de rede. Os primeiros MMGs, como Meridian59 (MERIDIAN59, 2002), utilizavam uma única máquina para cada servidor. Atualmente, a maioria dos MMGs em funcionamento ou em desenvolvimento empregam técnicas variadas para distribuir o processo servidor em várias máquinas, mas ainda sob a mesma entidade administrativa, que é responsável por manter o serviço. O custo de manutenção desta infra-estrutura pode chegar a um milhão de dólares anuais ou mais (IGDA, 2002).

Recentemente, a iniciativa Butterfly Grid (BUTTERFLY.NET, 2002) propôs a Computação em Grade como forma de reduzir custos e aumentar a qualidade do serviço oferecido pelos MMGs. Na proposta Butterfly Grid os servidores de simulação são delegados para provedores de serviço, e o middleware do Grid realiza a gerência destes provedores de serviço de forma transparente à aplicação. Apesar das vantagens oferecidas, o Butterfly Grid depende da existência de uma poderosa infra-estrutura para prover a simulação. A empresa desenvolvedora e os consumidores do MMG, de forma direta ou indireta, continuam dependentes de um considerável investimento financeiro para manter o serviço.

4 Peer-to-Peer: Reduzindo Custos

Como visto na seção anterior, o custo da plataforma de execução de um MMG é alto, tanto para a empresa que mantém o serviço, quanto para o cliente, que paga mensalidades. Para uma maior redução de custos poderia ser utilizado *hardware* voluntário. Por exemplo, poderia ser utilizado o próprio *hardware* dos participantes, enquanto eles jogam, para o fornecimento do serviço.

Atualmente, o *hardware* dos jogadores é utilizado essencialmente como terminal de entrada e saída da simulação. Como os computadores pessoais estão cada vez mais baratos e poderosos, há cada vez mais poder computacional disponível para a realização de uma simulação cooperativa, o que eliminaria a dependência de uma infra-estrutura central executando em *hardware* caro.

Porém, a distribuição do processo simulador não pode ser simplesmente delegado a um cliente, pois um sistema deste tipo seria inviabilizado pela vulnerabilidade à trapaças (CUNNIN, 2000). Além disso, ao delegar uma parte da simulação a um cliente qualquer, se este falhar, uma parte da simulação seria perdida.

5 Simulação Replicada

Uma maneira de realizar simulação distribuída envolve a distribuição de dados, ou seja, cada participante da simulação possui apenas uma parte do estado. No contexto de jogos, uma solução para distribuir a simulação, enquanto previne-se trapaças, é replicar o estado completo da mesma em todos os participantes. Isto pode ser feito utilizando-se uma topologia *Peer-to-Peer* de malha completa e um protocolo simples do tipo *stop-and-wait* para a sincronização das entradas entre os participantes. Desta forma, garante-se que todas as réplicas da simulação passam por uma evolução idêntica do estado ao longo do tempo e de maneira aceitável para uma simulação em tempo real. Esse método foi empregado com sucesso na implementação de Age of Empires (BETTNER; TERRANO, 2001), um dos jogos de estratégia em tempo real de grande sucesso comercial. Neste trabalho denominaremos esta técnica de Simulação Replicada (SR, para abreviar).

A SR, porém, apresenta algumas limitações. A topologia *Peer-to-Peer* de malha completa não é escalável, sendo viável para simulações que envolvem uma ou duas dezenas de participantes, mas inviável para uma simulação com mil ou um milhão de jogadores. Outro problema é a necessidade de um protocolo *stop-and-wait*: além de a simulação executar na velocidade do participante mais lento, o protocolo estabelece que cada vez que houver uma dependência de dados não-satisfeita, a simulação deve parar e esperar o fornecimento do dado faltante. Isto resultaria em uma experiência insatisfatória para o participante, onde a saída do simulador ficaria constantemente parando e avançando, sem uma correspondência convincente com o tempo real. Porém, como discutido em (BETTNER; TERRANO, 2001), este problema pode ser resolvido com a introdução de uma latência mínima entre os eventos gerados pelo participante e o acontecimento dos mesmos, proporcional à latência de comunicação. Assim, a simulação replicada se torna viável, mas apenas para simulações interativas tolerantes a uma latência mínima entre o comando do usuário e seu processamento pelo simulador. Jogos de estratégia em tempo real, como *Age of Empires*, possuem esta tolerância (tipicamente em torno de 400ms a 800ms), porém isto não ocorre com jogos de ação em tempo real, como, por exemplo, corridas de carro, onde um tempo de resposta ideal nunca é maior do que 100ms.

6 Proposta FreeMMG

Utilizando a SR como base, propomos o *framework* FreeMMG (figura 1). Como todas as propostas de arquiteturas para MMGs, o FreeMMG depende de um processo servidor centralizado sob responsabilidade de alguma entidade confiável. Porém, o servidor delega grande parte do peso da simulação para os nodos participantes da simulação (os *clientes*). A idéia central da proposta é dividir o estado global da simulação em segmentos e utilizar a técnica de SR em cada um deles, de forma individual. Todos os eventos de simulação que envolvem apenas um segmento do estado são tratados localmente por clientes-assinantes do segmento através do protocolo SR. Eventos que envolvem dois segmentos distintos resultam em comunicação através do servidor com outro protocolo específico. O servidor também é responsável por gerenciar a autenticação, inserção e a remoção de clientes em cada um dos círculos *Peer-to-Peer* de simulação. Segmentos do estado que não possuem nenhum cliente-assinante ficam guardados em disco, no servidor, até que sejam reativados por um novo pedido de assinatura de um grupo de clientes.

A segmentação do estado não é transparente ao usuário: as fronteiras entre os segmentos ficam, de alguma forma, aparentes para o participante da simulação. Por sua vez, o programador da aplicação deve tentar diminuir a frequência com que os eventos envolverão mais de um segmento, pois isso gera comunicação com o servidor, que é o gargalo do sistema. A grande vantagem da proposta é o custo associado: o servidor não precisa de nenhum *hardware* excepcional. Estima-se que o *hardware* utilizado por um servidor web de pequeno porte

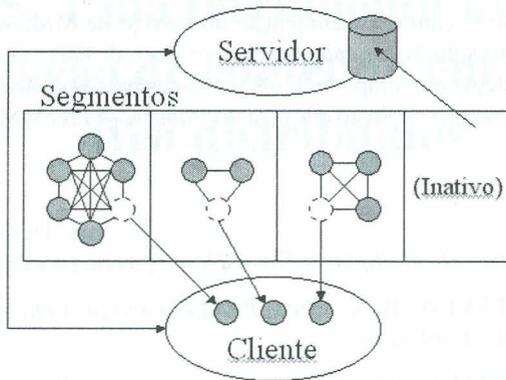


Figura 1: Framework FreeMMG

seja suficiente. Este trabalho, portanto, contribui para que os *Massively Multiplayer Games* se tornem mais acessíveis a pesquisadores, desenvolvedores e usuários.

7 Trabalhos Relacionados

Baughman et al (BAUGHMAN; LEVINE, 2001) formalizam, pela primeira vez, o problema da trapaça em jogos multi-jogador centralizados e distribuídos, e propõem um protocolo seguro denominado *Asynchronous Synchronization* para jogos em tempo real cujo processo servidor é distribuído (arquiteturas sem servidor) e que pode ser aplicado para jogos com grande número de usuários. Cunnin (CUNNIN, 2000) trata de diferenciar o problema dos MMGs dos problemas tradicionais de simulação distribuída, e propõe um novo algoritmo de balanceamento de carga para MMGs cliente-servidor com múltiplos servidores. Keller et al propõem o sistema SOLIPSIS (KELLER; SIMON, 2002) para simulações interativas e distribuídas em grande escala. SOLIPSIS utiliza uma abordagem totalmente *Peer-to-Peer*, sem necessidade de servidores ou de qualquer outro tipo de hierarquia entre os participantes. Porém, segundo o autor, o sistema não suporta competição, pois não há preocupação com trapaças. Também não há uma preocupação forte com o aspecto de tempo real.

8 Conclusão

O MMG é, atualmente, uma aplicação de entretenimento. Porém, há grande potencial para sua aplicação em outras áreas, por exemplo, como ferramenta para estudos sócio-econômicos (SIMPSON, 1999). Este trabalho discutiu alternativas para redução dos custos

associados ao desenvolvimento e à manutenção do serviço de MMGs. Foi proposto o *framework* FreeMMG, uma solução baseada em *Peer-to-Peer*, de baixo custo, para implementação de MMGs de estratégia em tempo real. Os próximos passos deste trabalho consistem na conclusão do projeto e na prototipação em Java do *framework* FreeMMG.

Referências

- ALEXANDER, T. *Massively Multiplayer Game Development*. [S.l.]: Charles Rivers, 2003.
- BAUGHMAN, N. E.; LEVINE, B. N. *Cheat-Proof Payout for Centralized and Distributed Online Games*. 2001. IEEE Infocom.
- BETTNER, P.; TERRANO, M. *1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond*. 2001. Game Developer's Conference. [Http://www.gamasutra.com/features/20010322/terrano_01.htm](http://www.gamasutra.com/features/20010322/terrano_01.htm).
- BUTTERFLY.NET. *Powering Next-generation Gaming with Computing On-Demand*. 2002. White Paper. [Http://www.butterfly.net/platform/technology/idc.pdf](http://www.butterfly.net/platform/technology/idc.pdf).
- CUNNIN, I. J. *Load Balancing Schemes for Distributed Real-Time Interactive Simulations*. Dissertação (Mestrado) — Universidade de Waterloo, Ontario, Canada, 2000.
- DIGRA. *Digital Games Research Association*. 2002. [Http://www.digra.org/](http://www.digra.org/).
- DS-RT'02. *6th IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications*. 2002. [Http://www.computer.org/cspress/CATALOG/pr01853.htm](http://www.computer.org/cspress/CATALOG/pr01853.htm).
- EVERQUEST. 2002. [Http://everquest.station.sony.com/](http://everquest.station.sony.com/).
- GDC. *Game Developer's Conference*. 2002. [Http://www.gdconf.com/](http://www.gdconf.com/).
- IGDA. *IGDA Online Games White Paper*. 2002. [Http://www.igda.org/online/IGDA_Online_Games_Whitepaper_2002.pdf](http://www.igda.org/online/IGDA_Online_Games_Whitepaper_2002.pdf).
- JOGD. *Journal of Game Development*. 2002. [Http://www.jogd.com/](http://www.jogd.com/).
- KELLER, J.; SIMON, G. Toward a peer-to-peer shared virtual reality. In: *IEEE Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02)*. [S.l.: s.n.], 2002.
- MERIDIAN59. 2002. [Http://www.meridian59.com/](http://www.meridian59.com/).
- SIMPSON, J. B. *The in-game economics of Ultima Online*. 1999. Computer Game Developer's Conference, San Jose, CA. [Http://www.mine-control.com/zack/uoecon/uoecon.html](http://www.mine-control.com/zack/uoecon/uoecon.html).
- UO. *Ultima Online*. 2002. [Http://www.owo.com/](http://www.owo.com/).