Lidando com Trapaças em uma Arquitetura Multi-Servidor para Jogos Online Massivamente Multijogadores

Felipe L. Severino[1], **Cláudio R. Geyer**[1]

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

1. Introdução

Com o crescimento considerável na qualidade de conexão para o usuário doméstico é possível notar o aumento da popularidade de serviços online. Um destes serviços são os jogos online, em especial os jogos do tipo *Massively Multiplayer Online Games* (MMOG), ou jogos online maciçamente multijogadores, que têm agregado milhares de jogadores nos últimos anos.

Entre os jogos de maior destaque nos últimos anos estão World of Warcraft, Lineage II, Guild Wars, PlanetSide, entre outros. As características comuns a estes jogos é a existência de uma grande quantidade (dezenas a centenas de milhares) de jogadores online simultaneamente. A expectativa, com a evolução dos jogos online, é a abrangência de um maior número de usuários e o aumento em relação à complexidade (jogos mais elaborados), necessitando cada vez mais de recursos computacionais como poder de processamento e conexão de qualidade (alta vazão e baixa latência).

Outra característica comum a jogos online é a existência de trapaça, ou *cheating*, que é definida como sendo qualquer comportamento que um jogador assuma para ganhar vantagem sobre outros jogadores ou atingir um alvo se, de acordo com as regras do jogo, esta vantagem ou alvo não poderia ser atingido [YAN, RENDELL 2005]. A presença de jogadores trapaceiros em um jogo põe em risco a popularidade do mesmo. É sabido que, quanto maior a quantidade trapaças, menor a probabilidade de outros jogadores continuarem jogando de acordo com as regras, e torna-se comum a desistência de jogadores.

Este trabalho irá considerar a questão de trapaças em jogos online massivamente multijogadores a partir da arquitetura proposta por [BEZERRA 2009] que utiliza um conjunto de servidores para suprir o gargalo apresentado pela estrutura cliente-servidor. Nessa solução o mundo virtual é dividido em regiões, e cada região é atribuída a um servidor, sendo este responsável pelo gerenciamento dos jogadores presentes nessa região. Esta arquitetura, porém, introduz novos problemas em relação à segurança pois, além de jogadores maliciosos, poderão existir servidores maliciosos, que permitam ações ilegais por parte dos jogadores.

2. Técnicas anti-cheating

São diversas as técnicas utilizadas para prevenção ou detecção de trapaças, ou *cheating*, para arquiteturas *peer-to-peer* ou cliente-servidor.

AC/DC (*Algorithm for Cheating Detection by Cheating*) é um algoritmo desenvolvido por Ferretti e Roccetti [FERRETTI, ROCCETTI 2006] que busca a detecção de *timing cheat* (ou *lookahead time cheat*) através do atraso no envio de mensagens de atualização para um jogador suspeito.

DACA (*Dynamic Anti-Cheating Architecture*) [LIU, TANG 2009] realiza a divisão do mundo virtual em regiões centralizadas em *hotspots*, onde cada região utiliza uma rede P2P entre seus jogadores. Apesar de utilizar uma arquitetura P2P, DACA não é uma arquitetura totalmente descentralizada. Servidores centrais são utilizados para funções como autenticação e base de dados. Além disso, são utilizados servidores centrais para realizar o balanceamento de carga nas divisões de regiões e computação do jogo em regiões com suspeitas de trapaça.

Outros trabalhos podem ser citados como [LIU, LO 2008], [BAUGHMAN et al. 2007] e [KABUS et al. 2005] porém, assim como os trabalhos anteriores, estes trabalhos também ou estabelecem uma nova arquitetura, focando-se em questões de segurança, ou são técnicas para detecção/prevenção de uma única forma de trapaça ou ainda, possuem problemas de desempenho. Além disso, muitos destes trabalhos não permitem a utilização de outras técnicas.

Como objetivo principal deste trabalho, tem-se o desenvolvimento de uma técnica de detecção de trapaças através do uso de níveis de confiabilidade entre os servidores e verificação de estado dos jogadores, buscando independência em relação à arquitetura utilizada em cada região na qual o mundo virtual é dividido, além de permitir a utilização de outras técnicas como uma forma complementar de garantir níveis de segurança aceitáveis.

References

Baughman, N.; Liberatore, M.; Levine, B. Cheat-Proof Playout for Centralized and Peer-to-Peer Gaming. In: IEEE/ACM Transactions on Networking, 2007. Piscataway: IEEE Press, 2007. p.1–13.

Bezerra, C.; Geyer, C. Lidando com Recursos Escassos e Heterogêneos em um Sistema Distribuído Atuando como Servidor de MMOG. 2009. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Ferretti, S.; Roccetti, M. AC/DC: an Algorithm for Cheat Detection by Cheating. In: 2006 international workshop on Network and operating systems support for digital audio and video, 2006, Newport, Rhode Island. New York: ACM, 2006.

Kabus, P.; Terpstra, W.; Cilia, M.; Buchmann, A. Addressing Cheating in Distributed MMOGs. In: 4th ACM SIGCOMM workshop on Network and system support for games, 2005, Hawthorne, USA. New York: ACM, 2004. p.1–6.

Liu, H.; Lo, Y. DaCAP - A Distributed Anti-Cheating Peer to Peer Architecture for Massive Multiplayer On-line Role Playing Game. In: 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid, 2008. Washington, DC: IEEE Computer Society, 2008. p.584–589.

Liu, H.; Tang, B. DaCA: Dynamic Anti-Cheating Architectur for MMOGs. In: 2009 International Conference on Advanced Information Networking and Applications, 2009. Washington, DC: IEEE Computer Society, 2009. p.892–897.

Yan, J.; Rendell, B. A Systematic Classification of Cheating in Online Games. In: 4th ACM SIGCOMM workshop on Network and system support for games, 2005, Hawthorne, USA. New York: ACM, 2005. p.1–9.