

VORONOI STATE MANAGEMENT FOR PEER-TO-PEER MASSIVELY MULTIPLAYER ONLINE GAMES

Juan J. Jaramillo, Fabio Milan, R. Srikant,

**Department of Computer Science and Information Engineering
National Central University, Taiwan, R.O.C.**

Sistemas Distribuídos e Tolerância a Falhas

Estrutura da Apresentação

- Introdução
- VSM – Conceito
- Diagramas de Voronoi
- Abordagem e Pressupostos
- VSM
- Conclusões

Introdução

- *Massively Multiplayer Online Games (MMOG).*
 - Topologias *Client-Server (Server-Clusters)* e *Peer-to-Peer (Point-to-Point, P2P)*;
 - *Multiuser Virtual Environments (VE)*;
 - *Game States (GS)*.
- **Requisitos fundamentais.**
 - Manutenção da consistência dos *Game States* dos Objectos de Jogo(Localização, Posses, Estados);
 - Gestão adequada de custos computacionais e larguras de banda.

Introdução - Topologias

- ❑ Definem como está estruturada a rede.
- ❑ Topologia *Client-Server*:
 - *Game States* estão alojados de forma centralizada no(s) servidor(es);
 - Mensagens são enviadas de nodos cliente para o(s) servidor(es) e as respostas são sempre enviadas para todos os nodos cliente.
- ❑ Topologia *Peer-to-Peer*:
 - *Game States* estão distribuídos pelos clientes;
 - Existe distribuição do trabalho por todos os nodos.

Introdução - Topologias

- ▣ Problemas do *Client-Server*:
 - Escalabilidade;
 - Distribuição de uniforme da carga de trabalho.

- ▣ Problemas do P2P:
 - Heterogeneidade: Os clientes não têm todos os mesmos recursos;
 - “*Churn*”: Os clientes estão constantemente a entrar e sair.

Introdução – Modelos Consistência

- ▣ Definem como são feitas as actualizações dos *Game States* nos nodos.
- ▣ *Event-based*:
 - Todos os nodos armazenam os *Game States* de todo o jogo;
 - Todos os nodos executam a mesma lógica de jogo.
- ▣ *Update-based*:
 - Todos os *Game States* estão armazenadas apenas no nodo servidor;
 - Apenas o nodo servidor executa a lógica de jogo.

Introdução – Modelos Consistência

- *Event-based:*

- É sempre necessário que os eventos de todos os nodos sejam recebidos e processados para que o tempo possa avançar.

- *Update-based:*

- O tempo de jogo pode avançar em intervalos regulares;
- Mais fácil garantir segurança.

Introdução – Distribuição de GS

- ▣ Definem como dimensionar o VE com base na distribuição dos GS.
- ▣ *Replication-based:*
 - Servidores ligados numa topologia P2P;
 - Todos os servidores possuem um completa replicação dos GS;
 - Um evento pode ser processado por qualquer servidor;
 - Gestão flexível da distribuição de carga dos servidores;
 - Não é facilmente escalável.

Introdução – Distribuição de GS

- *Object-based:*
 - Os objectos de jogo são uniformemente distribuídos por todos os servidores;
 - Volume de comunicações entre servidores difícil de prever.

Introdução – Distribuição de GS

■ *Zone-based:*

- Os objectos de jogo são distribuídos geograficamente;
- Processamento dos eventos é maioritariamente local;
- O volume de comunicações entre servidores é geralmente constante;
- Podem haver restrições de acesso entre zonas;
- Mais facilmente escaláveis.

VSM - Conceito

- Proposta de novo sistema de gestão dos *Game States* em MMOGs chamado *Voronoi State Management (VSM)*:
 - Topologia: P2P;
 - Modelo de Consistência: *Update-based*;
 - Distribuição de *Game States*: *Zone-based*.

VSM - Conceito

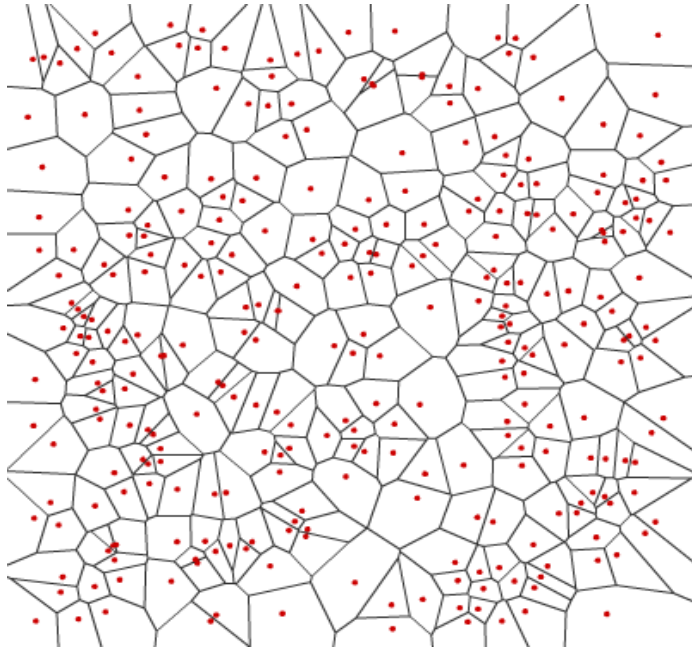
- Esta solução tem conta a heterogeneidade dos clientes e a constante “agitação” na rede.
- Particiona o VE num dado número de células através de diagramas de *Voronoi*, sendo cada uma destas células gerida por um cliente.
- Clientes com mais recursos, são designados “Agregadores” e passam a gerir várias células respeitantes a clientes sobrecarregados.

VSM - Conceito



- De forma a equilibrar a carga sobre nodos agregadores, o seu raio de influencia é ajustado dinamicamente e novas células são criadas quando necessário.
- A tolerância a falhas é garantida replicando os *Game States* em células de *backup*.

Diagramas de *Voronoi*



- Método de decomposição de um determinado espaço métrico, a partir de um conjunto discreto de pontos.
- A cada ponto é atribuída uma determinada célula (área).

Abordagem e Pressupostos

- O VE é um plano 2D com altura e largura fixas.
- “*Atributos*” são tuplos da forma (*tipo, nome, valor*), onde os tipos serão *int, char, float, string*.
- “*Objectos*” são tuplos da forma (*nome, atributos, x, y*).
- Os “*Objectos*” são criados, actualizados e destruídos através de eventos.
- Cada jogador controla um “*Objecto*” do tipo *avatar*, que tem uma área de interesse (AOI) de raio fixo.

VSM



- O VE é particionado em células de *Voronoi*, com a posição dos utilizadores a servirem de referência para a criação de cada célula;
- Cada nodo de utilizador gere as actualizações dos objectos na sua célula.
- Para gerir de forma eficaz todas as actualizações e a comunicação entre nodos são definidas entidades de controlo.

VSM

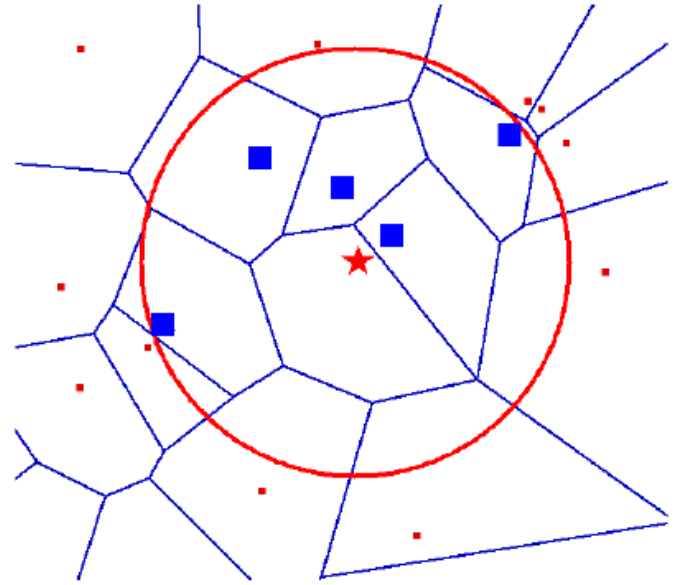
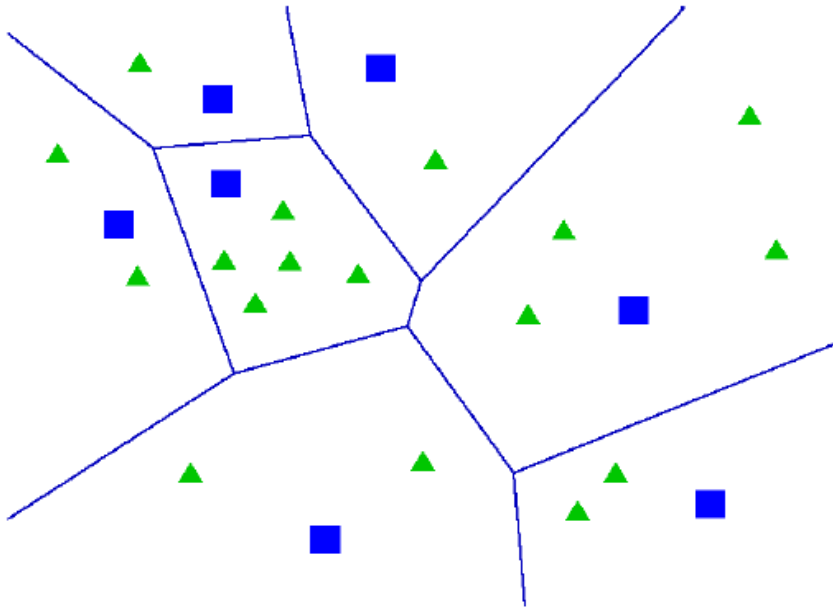
■ Entidades de controlo:

- *Peer*: apenas gera eventos e mostra resultados da resposta ao evento gerado;
- *Arbitrator*: é responsável por uma célula, decide como devem mudar os GS de acordo com os eventos e a lógica de jogo;
- *Aggregator*: é um *arbitrator* especial, com localização fixa e um determinado raio de controlo.

VSM

- Todos os nodos utilizador têm inicialmente os papéis de *arbitrator* e *peer*;
- Apenas nodos utilizador capazes ou servidores (se necessário) podem ser *arbitrators*;
- À medida que os nodos utilizador começam a ficar sobrecarregados, um só nodo utilizador (ou servidor) é seleccionado para ser o *arbitrator* responsável por um conjunto de nodos (*peers*) – *Aggregator*;
- Nodos independentes que tenham na sua AOI nodos vizinhos pertencentes ao raio de controlo de um *aggregator*, comunicam com eles através do *aggregator*.

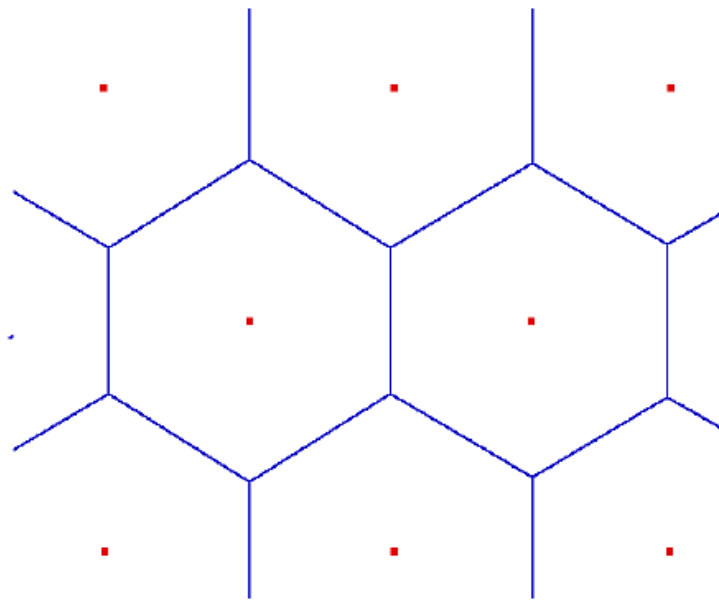
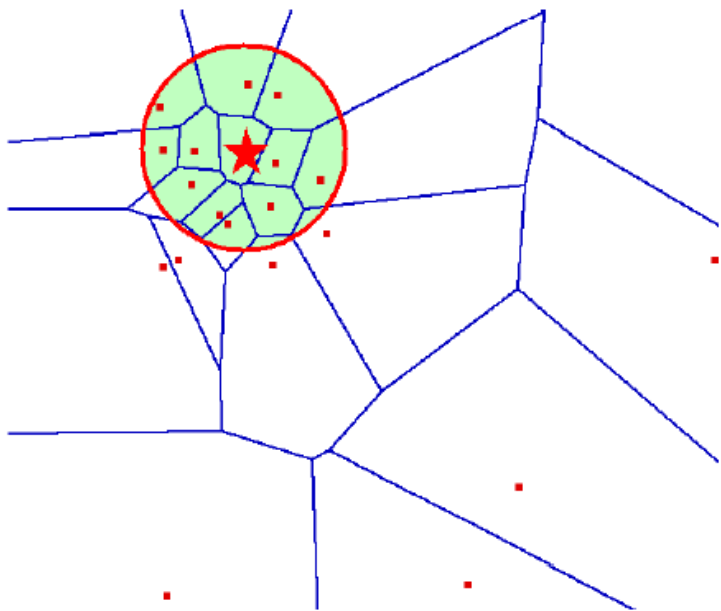
VSM



VSM

- Inicialmente são criados pelo servidor alguns nodos virtuais como *arbitrators* em localizações específicas;
- O controlo é depois transferido de forma gradual para nodos utilizador á medida que estes forem entrando;
- A transferência da propriedade sobre objectos do jogo é obrigatoriamente feita entre *arbitrators*;
- Existe um servidor de entrada.

VSM



VSM



- De forma a garantir as características pretendidas em termos de consistência, distribuição de carga e tolerância a falhas foram definidos os seguintes mecanismos:
 - Controlo de consistência;
 - Distribuição de carga;
 - Tolerância a falhas;

VSM - Controlo de consistência

- Modelo *update-based*;
- Eventos de uma determinada célula são enviados ao *arbitrator* responsável por essa célula. São depois processados directamente ou reencaminhados para outros *arbitrators* se afectarem objectos exteriores á célula;
- Cada *arbitrator* tem “bolsas de tempo” internas em que guarda eventos recebidos. O processamento é depois feito em intervalos regulares;
- Quando o estado de objectos muda a actualização de estado é enviada aos *arbitrators* que tenham nodos que consigam ver esses objectos.

VSM - Controlo de consistência

- Os eventos e actualizações podem ser *fiáveis* ou *não fiáveis* com base na sua importância;
- O envio de eventos críticos de forma fiável pode representar um acréscimo de atraso;
- A transferência da propriedade de objectos de jogo tem de ser feita de forma explícita entre *arbitrators*;
- Quando um *arbitrator* recebe eventos para objectos que não são seus os eventos são reencaminhados para o devido dono.

VSM - Controlo de consistência

- Existem dois tipos principais de acções em MMOGs:
 - Actualização básica:
 - Evento afecta um ou mais objectos directamente;
 - É processado pelo *arbitrator* dono dos objectos afectados. Actualizações são enviadas para *arbitrators interessados*.
 - Actualização transaccional:
 - Evento afecta estados de dois ou mais objectos simultaneamente;
 - O *arbitrator* do primeiro objecto requer o bloqueio dos atributos do segundo objecto ao seu *arbitrator*;
 - Os estados dos dois objectos são actualizados;
 - O bloqueio dos atributos do segundo objecto é levantado;

VSM - Distribuição de carga

- A carga é inicialmente distribuída por muitos nodos utilizador com baixa capacidade;
- À medida que vai acontecendo sobrecarga dos nodos é feita a agregação a nodos de grande capacidade ou a nodos servidor;
- Aquando da sobrecarga o *arbitrator* pede ao servidor de entrada um *aggregator* para se juntar;
- Quando um *aggregator* entra em sobrecarga diminui o seu raio de controlo.

VSM - Distribuição de carga

- Quando o *aggregator* diminui o seu raio de controlo, se existirem nodos sobrecarregados fora do raio de controlo, estes nodos pedem ao servidor de entrada novos *aggregators*;
- Quando um *aggregator* fica com um nível de carga baixa, desintegra-se e devolve o controlo das células a nodos utilizador.

VSM - Tolerância a falhas

- Esta arquitectura baseia-se na utilização dos recursos dos utilizadores;
- Cada *arbitrator* escolhe aleatoriamente um nodo utilizador como seu *backup arbitrator*;
- Quando um *arbitrator* falha o seu *backup* transfere a propriedade sobre os objectos de jogo do *arbitrator* que falhou para *arbitrators* adjacentes.

VSM - Tolerância a falhas

- Cada *aggregator* tem também um *backup aggregator*;
- Quando um *aggregator* falha o seu *backup aggregator* assume o papel do *aggregator* original;
- O *backup aggregator* pede depois ao servidor de entrada um novo *backup aggregator* para si.

VSM - Resumo

- Arquitectura responde a sequências Evento – Efeito em três saltos ponta-a-ponta;
- Escalável á medida que são adicionados utilizadores através da utilização dos seus recursos;
- A distribuição equilibrada da carga de trabalho é conseguida através do mecanismo de agregação em zonas sobrecarregadas;
- A tolerância a falhas é assegurada através do uso de nodos de *backup*;
- Esta arquitectura permite a migração dos actuais sistemas *Client-Server* uma vez que prevê a utilização de recursos de servidores e utilizadores.

Conclusões

- Foi apresentado o *Voronoi State Management* (VSM) uma arquitectura P2P para gestão de estados, que visa a utilização de recursos do lado do utilizador e servidor de forma transparente;
- Está a ser feita a avaliação em termos de eficácia e consumo de largura de banda dos mecanismos de distribuição de cargas e tolerância a falhas através de simulação;
- Este modelo proporciona possibilidades de desenvolvimento adicionais neste campo através da alteração de comportamento das entidades de controlo introduzias.

*VORONOI STATE MANAGEMENT
FOR PEER-TO-PEER MASSIVELY
MULTIPLAYER ONLINE GAMES*

Obrigado!

Sistemas Distribuídos e Tolerância a Falhas