




MASSIV

Middleware pro tvorbu online her

Obsah prezentace

- **Úvod**
 - **Prostředky poskytované Massivem**
 - **Využití jádra Massivu v Demu**
 - **Zhodnocení projektu**
 - **Prezentace Dema**
- 

Úvod

Část 1.



Tým projektu Massiv

- **Zahájení projektu: říjen 2001**
- **Vedoucí projektu: Ing. Petr Tůma, Dr.**
- **Řešitelé:**
 - Štěpán Vondrák - stoupik@users.sourceforge.net
 - Marek Vondrák - markoid@users.sourceforge.net
 - Petr Tovaryš - boovie@users.sourceforge.net
 - Ondřej Pečta - octa@users.sourceforge.net
 - Marek Švantner - marekus@users.sourceforge.net
 - Martin Havliša - hafik@users.sourceforge.net

Cíl projektu

- **Systém pro tvorbu online multiplayer her**
- **Online hry**
 - **Běh 24 hodin denně**
 - **Perzistence**
 - **Interaktivnost**
 - **Mnoho hráčů**



Základní charakteristiky Massivu

- **Distribuvanost**
 - Svět může být simulován více servery
- **Objektovost**
 - Svět se sestává z objektů, které mohou mezi servery volně migrovat
- **Statická data**
 - Správa a distribuce dat, která se příliš často nemění

Komu je Massiv určen

- **Middleware pro nekomerční sféru**
 - **Nezávislí vývojáři – Open Source**
 - **Přenositelnost**
 - **Win32, Linux**
 - **Nelze předpokládat nasazení serverů na jediné lokální síti**

Součásti projektu Massiv

- **Jádro**
 - Objektově orientovaný distribuovaný systém
 - Zdrojový kód knihovny
 - Nástroje pro překlad
- **Demo**
 - Jednoduchá ukázková online hra
 - Demontrace použití prostředků jádra
 - Nástroje pro konfiguraci a správu Dema

Prostředky poskytované Massivem

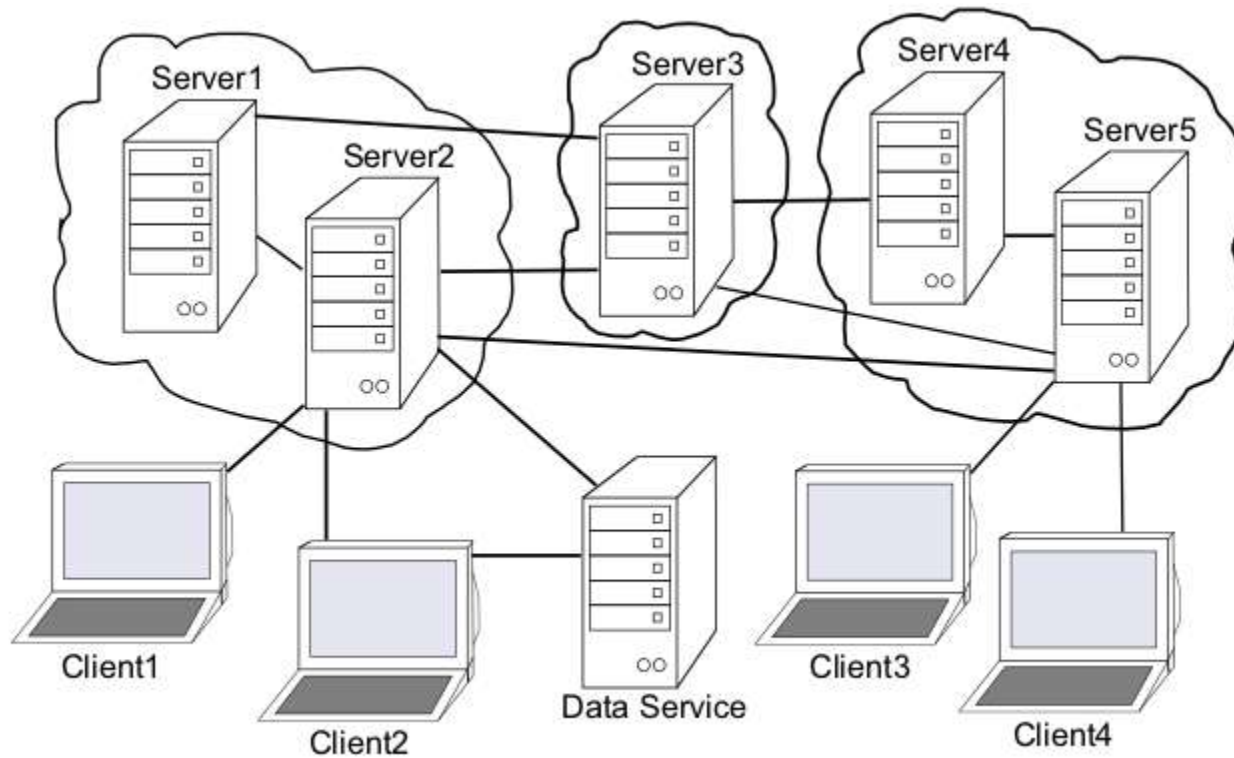
Část 2.



Distribuvanost

- **Servery rozptýleny libovolně po síti**
 - **Potenciálně velké latence**
 - **Nepředpokládá se statické rozdělení herního světa, které by minimalizovalo komunikaci mezi servery**
 - **Automatická a transparentní distribuce objektů**
 - **Bez nutnosti ruční administrace**
- **Tři druhy „uzlů“**
 - **Simulační servery, klienti, datové servery**

Nasazení Massivu



Bezpečnost

- Šifrování dat mezi každými dvěma uzly
 - Autentizace pomocí RSA klíčů
 - Symetrické šifrování při přenosu dat
- Omezená práva klientských uzlů
 - Klienti nemohou přímo měnit stav simulace, pouze zasílat požadavky speciálním objektům

Objektový model

- „Managed objects“
 - Objekty automaticky spravované systémem
 - Příslušné třídy se píší v C++
 - Musí se dodržovat zvláštní pravidla
 - Speciální datové typy
 - Popis tříd v IDL (Interface Definition Language)
 - Serializace, introspekce, RPC
 - Povolen přímý přístup na lokální objekty
- Lokální garbage collector

Migrace

- Každý objekt vlastněn právě jedním uzlem
- Migrace objektů
 - Základní forma komunikace
 - Adresáti migrace jsou objekty
 - Objekt = zpráva
 - Základní prostředek běhu simulace
 - Běh Massivu řízen událostmi
 - Doručení migrace = událost

Replikace

- **Použití:**
 - Přenos dat nutných pro prezentaci světa na klienty
 - Optimalizace komunikace mezi servery
- **Objekt může být replikován na libovolný počet uzlů**
 - Kopie (částí) objektů určené výhradně pro čtení
 - Automaticky udržované v aktuálním stavu

Migrační a replikační skupiny

- **System ke skupinám přistupuje jako k celku**
- **Objekty v migrační skupině jsou vzájemně lokální**
 - **Základní prostředek pro zajištění efektivity v distribuovaném prostředí s velkými latencemi**
- **Příslušnost do skupiny určována dynamicky**
 - **Příklady skupin:**
 - **Dynamické datové struktury (spojový seznam)**
 - **Postava hráče s celým inventářem**

Vzdálené volání metod

- Implementace pomocí migrací
- **Asynchronní RPC**
 - Doručení lze načasovat na konkrétní simulační čas
 - Možnost získání výsledků volání
- **Synchronní RPC**
 - Neblokuje doručování ostatních událostí

Další funkcionality jádra

- **Archivace konzistentního stavu světa**
 - Nenarušuje plynulost běhu simulace
- **Stahování dat na pozadí**
 - Data se mohou měnit za běhu simulace
 - Hierarchické uspořádání dat
 - Použití: konfigurační soubory, textury, modely
- **Vyvažování zátěže serverů**

Využití jádra Massivu v Demu

Část 3.



Demo

- **Ukázková 3D aplikace**
- **Využití všech prostředků poskytovaných jádrem**
- **Ukázky řešení základních situací a problémů online her pomocí prostředků Massivu spíše než základ pro reálné hry**

3D Mapa

- **Simulovaný svět tvořen mapou, po které se pohybují hráči**
- **Mapa rozdělena na sektory**
 - Každý sektor může být na jiném serveru
- **Sektory nejsou data objekty, ale managed objekty**
 - Modifikace terénu v reálném čase (kopce, údolí)
 - Změny prezentovány klientům pomocí replikace

Sektor (1)

- **Výšková mapa (kopce, údolí)**
 - Rozdělena na 8x8 čtvercových polí
 - Každé pole má vlastní materiál (tráva, skála, ...)
- **Obsahuje entity**
 - Pohyblivé: postavy hráčů, ovce
 - Nepohyblivé dekorace: stromy, tráva, budovy

Sektor (2)

- **Sektor a všechny entity, které obsahuje, tvoří jedinou migrační skupinu**
 - **Všechny operace v rámci sektoru vždy lokálně a rychle**
 - **Sektor může manipulovat přímo s entitami**
 - **Přesun sektoru na jiný server přesune i všechny jeho entity**

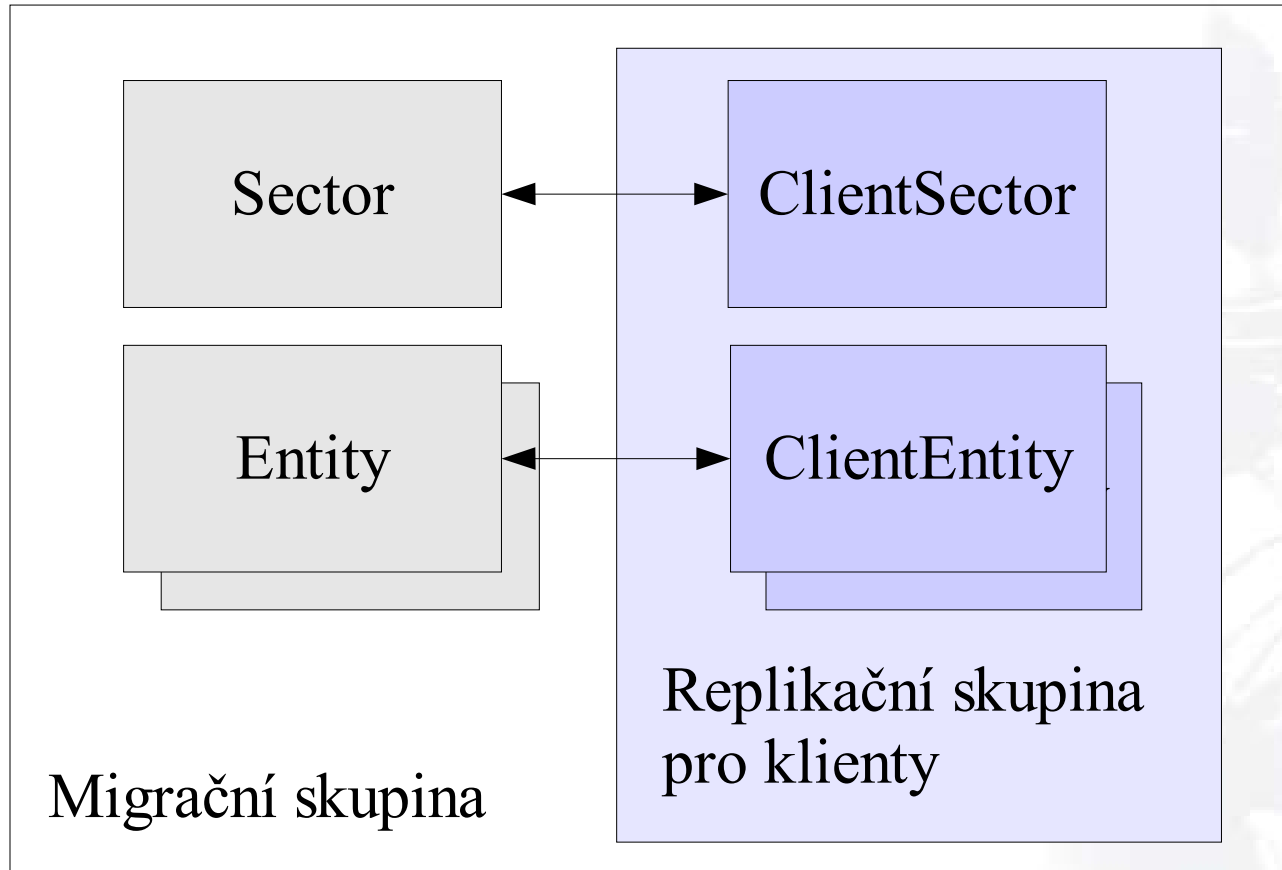
Pohyb entit

- **Uvnitř sektoru jako u nedistribučované aplikace**
- **Mezi sektory**
 - **Odpojení entity z aktuálního sektoru**
 - **Rozdělení migračních skupin**
 - **Migrace k novému sektoru**
 - **Napojení na nový sektor**
 - **Spojení migračních skupin**

Replikace

- Každý objekt rozdělen na dvě části
 - Veřejná část – replikující se na klienta
 - Obsahuje data nutná pro prezentaci
 - Privatní část
 - Obsahuje vše pro vlastní logiku hry
- Rozdělení umožňuje
 - Přenášet jen minimum potřebných dat
 - Zvyšuje bezpečnost
 - Klientské aplikace nevidí strukturu serverových objektů

Sektory a entity



Zhodnocení projektu

Část 4.



Výhody a nevýhody objektového modelu

- **Klady**
 - Použití C++ – vysoká efektivita
 - Obecnost objektového modelu
 - Použití nemusí být omezeno pouze na online hry
- **Zápory**
 - Z důvodu velkých latencí mezi servery nejsou k dispozici aktuální dupliky objektů – při pádu serveru je nutno provést restart ze zálohy

Co šlo dobře

- **Abstraktní model doručování zpráv**
 - **Není rozdíl mezi zprávou a objektem**
 - **Jednoduchá implementace RPC**
- **Plné využití C++ a STL**
- **Mnoho vlastností navíc, které nebyly v plánu**
 - **Synchronní RPC a vyjímky**
 - **Garbage collector**

Co šlo špatně

- **Nepřesný návrh**
 - **Mnoho změn až v průběhu vývoje**
- **Ladění**
 - **Více serverů a klientů znesnadňuje využití standardních debugovacích technik**
- **Mnoho vlastností navíc, které nebyly v plánu**

Prezentace Dema

Část 5.



Prezentace Dema

- **Data download**
- **Predikce pohybu**
- **Editor**
- **Konzole**
- **Chat**



Data download

- **Všechna statická data (textury, modely) implementována jako datové objekty**
 - **Umožňuje spustit klienta s minimálními daty**
 - **Přenos dat až když je potřeba (rychlost stahování volitelná)**
 - **Pokud model není k dispozici, je použit náhradní (zástupný) model**

Editor

- **Za běhu simulace umožňuje**
 - **Editovat výškové mapy (modelování krajiny)**
 - **Měnit druhy povrchu (materiály)**
 - **Přidávat, mazat a přesouvat entity**
 - **Měnit vlastnosti objektů (změna modelů dekorací)**
- **Editor přístupný pouze pro privilegované klienty (administrátoři)**

Predikce pohybu

- Data přenášena ze serveru na klienta jen několikrát za sekundu
- Pro plynulou prezentaci klient provádí predikci pohybu



Replikace entit

- **Veřejná část: ClientEntity**
 - Pozice na mapě
 - Typ (postavy hráčů, ovce, dekorace, ...)
 - Model tags
 - Určují model, který se použije na prezentaci entity
- **Privátní část: Entity**
 - Napojení na aktuální sektor
 - Data pro pohyb entity po mapě

Replikace sektoru

- **Veřejná část: ClientSector**
 - Výšková mapa a materiály
 - Explicitní seznam entit není potřeba
 - Entity se replikují automaticky, neboť jsou ve stejné replikační skupině jako ClientSector
- **Privátní část: Sector**
 - Seznam entit
- **Na klienty se replikuje pouze část mapy**
 - V definovaných intervalech požadavek na replikaci sektorů v blízkosti hráče
 - Při odpojení klienta zrušení všech požadavků