Strukturierte Peer-to-Peer-Netze für Mehrspieler-Online-Spiele

Heiko Niedermayer, Marc Fouquet, Simon Rieche,
Klaus Wehrle, Georg Carle

University of Tübingen

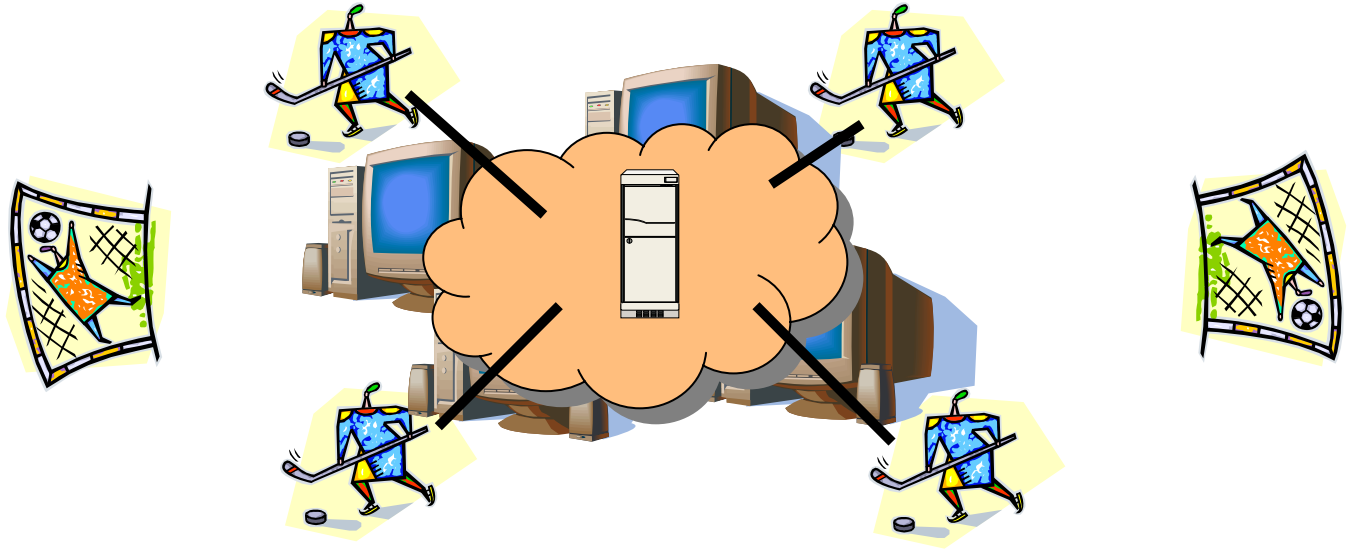
Überblick

Überblick

- Motivation
- MMORPGs
- Verwandte Arbeiten
- Probleme und Annahmen
- Peer-to-Peer-Infrastruktur
- Zusammenfassung

Motivation

Mehrspieler-Spiele



Popularität

Popularität

- Ego-Shooter
  - Populär, aber nicht mit großen Spielerzahlen (vielleicht max. 100)
  - Problematisch: „Low Ping“-Anforderung
- Echtzeit-Strategiespiele
  - Gameplay eher nur für begrenzte Spielerzahlen (üblich 2-8, max. 20)
- Adventures/RPGs
  - Eignen sich für massiven Mehrspielerbetrieb
  - Sehr populär
  - Everquest (2004)
    - Verkaufszahlen > 500.000
  - World of Warcraft (2004/2005)
    - 250.000 am 1. Tag
    - Verkaufszahlen insgesamt > 2.000.000

Situation: heute (Spielererfahrungen)

Ausfallzeiten

- Pro Woche mehrere Stunden, i.a. planmäßig zu Niedriglastzeit
- Ab und zu kein Login möglich
- Vereinzelt Erfahrung der Degradierung der Dienstqualität
  - Langsame Reaktion
  - System interagiert weniger mit dem Spieler
    - Handlungsfiguren, virtuelle Gegner, …
  - Erzwungener Logout

- Schlechte Erfahrungen => schlechte Kritik => Popularitätseinbußen => weniger Einnahmen
- Businessmodell meist Abo-artig

Situation: heute (Technisch)

Technische Realisierung

- Server Farm

Rechenzentren

- Nicht weltweit, eher eines pro Kontinent

Spielwelten

- Teilen sich in parallel ablaufende Spielwelten auf
- Teilungskriterium: Sprache (auch weitere Unterteilung)

MMORPGs

Anfänge

- Textbasierte Multi-User Dungeons (MUDs) seit 1970er

Begriff

- **M**assively **M**ultiplayer **O**nline **R**ole **P**laying **G**ames

Spielsystem

- Spieler steuert Spielfigur („Held“)
- Spielfigur / Spiel
  - Hat gewisse Eigenschaften, sammelt Erfahrung, usw.
  - i.a. Gruppe von Spielfiguren („Heldengruppe“)
  - Kollektives Lösen von Aufgaben, Besiegen von Gegnern (anderen Spielerfiguren, „Monstern“, Computergegner)
- Spielwelt (Karte)

MMORPGs – Skalierungsfragen

Begrenzungen

- Latenz
  - Nicht so wichtig bei MMORPGs, da Kämpfe nicht durch Reaktion, sondern durch Zufall und Strategie entschieden werden
  - Grenzen eher durch Spielerinteraktionsrate gegeben
- Technisch
  - Serverleistung
  - Anbindung (bei DSL-Peers als Server z.B. Problem der geringen Uplink-Bandbreite)
- Spieltechnisch
  - Größe der Spielwelt (heute auf bis zu 4000 Helden ausgelegt)
  - Anzahl der Aufgaben, Geschichten, usw.
  - Erst viele Anfänger (Ansturm), dann viele Fortgeschrittene

MMORPGs – Skalierungsfragen

Instanzen

- Ziele
  - Kontrolle des Schwierigkeitsgrads von Aufgaben,…
- Skalierbarkeitseinfluss
  - Unnötigen Designaufwand vermeiden
  - Wiederverwendung von Kreaturen, Storyelementen,…
  - Verwaltungsaufwand reduzieren
  - Stärkere Clusterbildung
- Methode
  - Abspalten einer Gruppe in eigene Instanz eines Teiles der Spielwelt
  - Solange die Gruppe nicht wieder in gemeinsamen Teil zurückkehrt, so interagiert sie mit keinen weiteren Spielern (unabh. Cluster).
- Anwendung
  - Bei *Guild Wars*
    - Nur kleine Bereiche (z.B. Städte) der Spielwelt gemeinsam, sonst gehen Spielergruppen getrennte Wege.

MMORPGs – Aufgaben der Server

Serveraufgaben

- Verwaltung der Spieler
- Verwaltung der Spielerposition
- Verwaltung der Spielwelt
- Kampf
- Chat
- Accounting

Verwandte Arbeiten

Zonen-basierte Ansätze

- Aufteilung der Spielwelt in Zonen
- Peers können Zonenverwalter werden
- Probleme
  - Dynamik von Peer-to-Peer-Netzen
  - Einfache Betrugsmöglichkeiten
  - Sichere Speicherung von Spiel- und Charakterdaten

Solipsis

- Reines Peer-to-Peer
- Struktur über Nachbarschaftsbeziehungen in der Spielwelt
- Verbindungen zu allen Peers im Sichtbereich
- Bisher nur Chat-Funktionalität umgesetzt

Probleme und Annahmen

Heutiger Serveransatz

- Wenig flexibel, Fehleranfällig
- Ausfallzeiten durch Software- und Hardware-Probleme
- Lastprobleme durch schwankende „Bevölkerung“ der Spielwelten

Annahmen

- Software für P2P muss spontanen Eintritt und Austritt unterstützen
- Instabile Peers führen nicht zu instabilem Gesamtsystem

Peer-to-Peer-Infrastruktur

Realisierung

- Infrastruktur-Peers
  - Keine Trust-Probleme
- Dynamische Aufteilung in Regionen
- Verwendung von CAN
  - Spielwelt ist 2d-Karte!
  - Lookup von Daten (ggf. in anderer DHT)
- Redundanz durch Weiterleitung der Daten an direkte CAN-Nachbarn

Peer-to-Peer-Infrastruktur

Vorteile eines Peer-to-Peer-Infrastruktur-Ansatzes

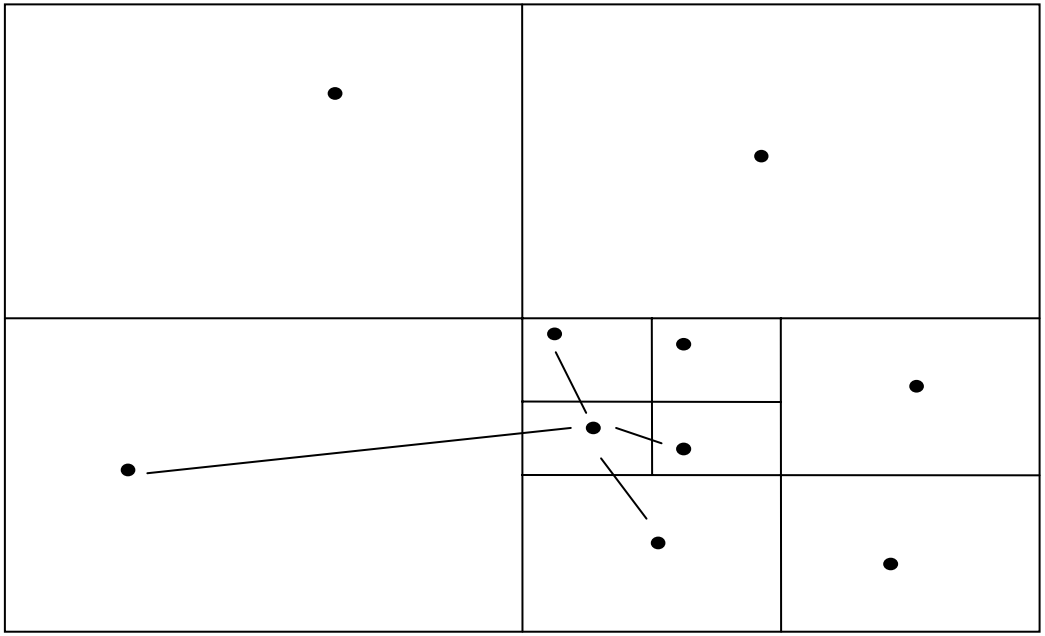
- Kaum Problem mit Churn

  (schnellen Ein- und Austreten von Peers)
- Geringere Fragmentierung
  - Bei Spielerrechners als Peers verfällt die Spielwelt in viele kleinste Ausschnitte
- Keine Probleme mit der Uplink-Bandbreite von DSL-Clients
- Sicheres Accounting
- Leichtes Verhindern von Mogelei
- **Infrastruktur kann verteilt sein!!!**
  - **Gegebenenfalls auch Latenz-Optimierung zwischen beteiligten Peers und Clients (denkbar)**

CAN

CAN als DHT

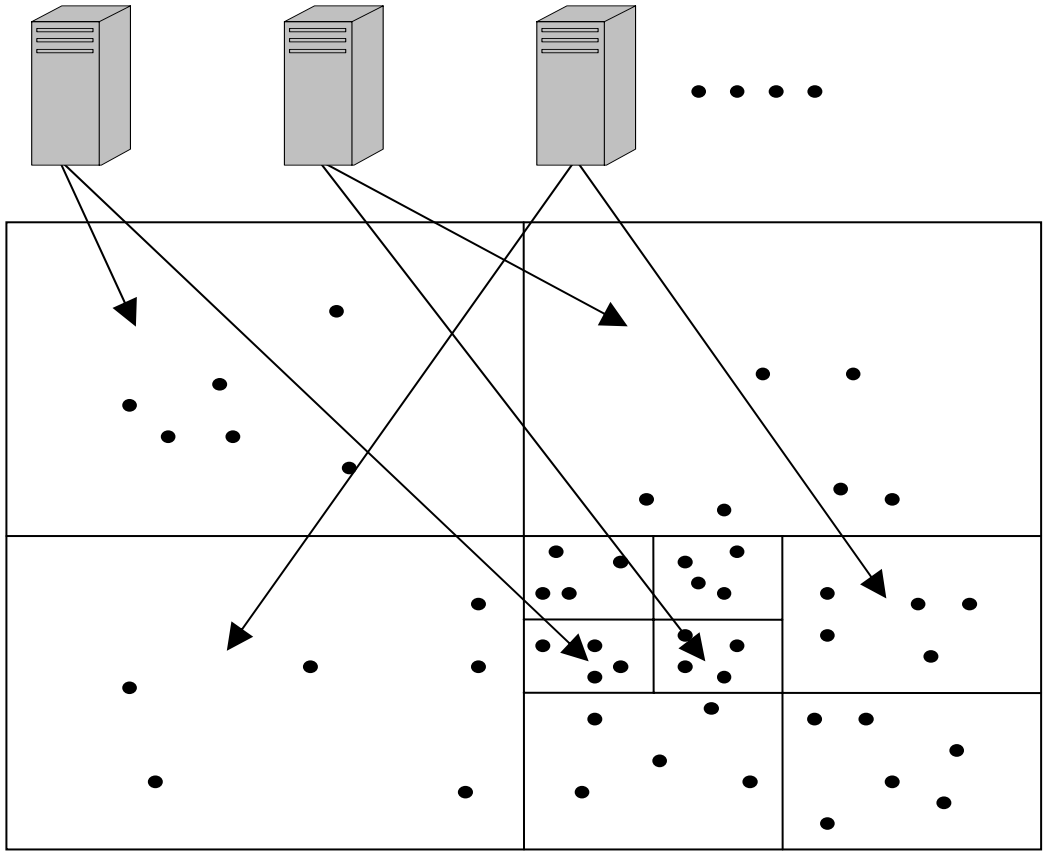
- d-dimensionaler ID-Raum
- Aufteilung in Zonen
- Jeder Knoten hat Kontakt zu 2d Nachbarn



Aufwand

- Zustand pro Knoten $O(d)$
- Suche $O(dn^{1/d})$
  - Für uns nicht relevant (können für Inhaltsspeicherung DHT mit log(n) nehmen)

CAN als unterstützende DHT



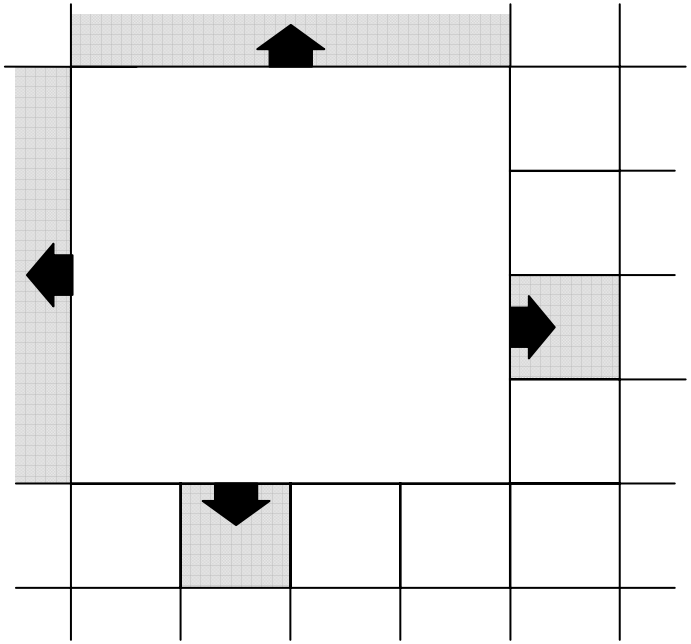
Infrastruktur-Host

Spieler in der Spielwelt

Redundanz

Redundanz

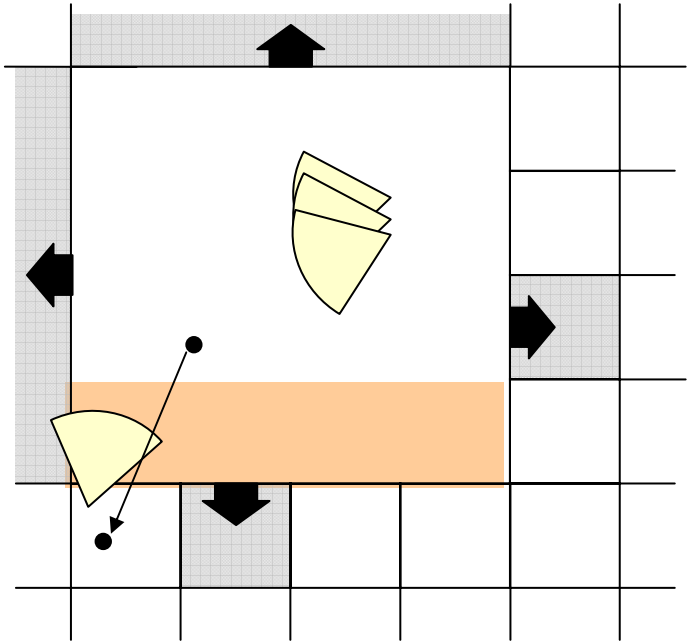
- Informationen werden an 2d Nachbarn weitergegeben
- Bei Ausfall springt einer dieser Knoten ein



Verteilen der relevanten Information

Problem, dass nicht jeder Peer, der eine Spielfigur verwaltet **alle Informationen im Sichtbereich** bekommt

- Anfordern der relevanten Informationen von den Peers, die die Bereiche kontrollieren
  - Teilbereich evtl. ausreichend

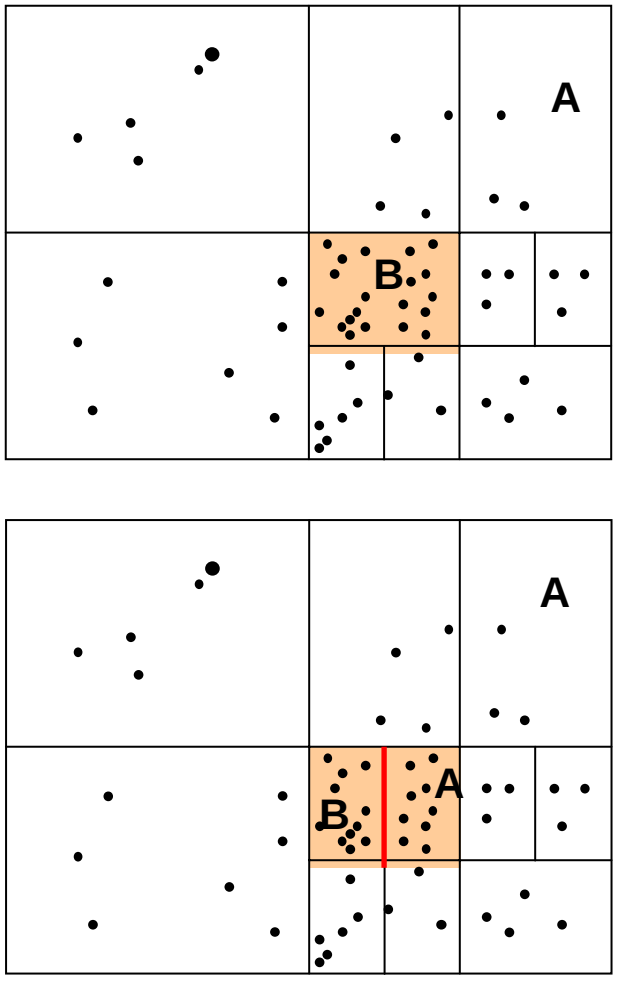


Behandeln von Hotspots – Lastbalancierung

Hot Spots

- Einzelne Spielbereiche unterschiedlich populär (Städte, usw.)
- Hot Spots dynamisch (Treffen größerer Gruppen irgendwo, Stellen, die nur Zeitweise interessant sind)

⇒ Dynamische Lastanpassung notwendig

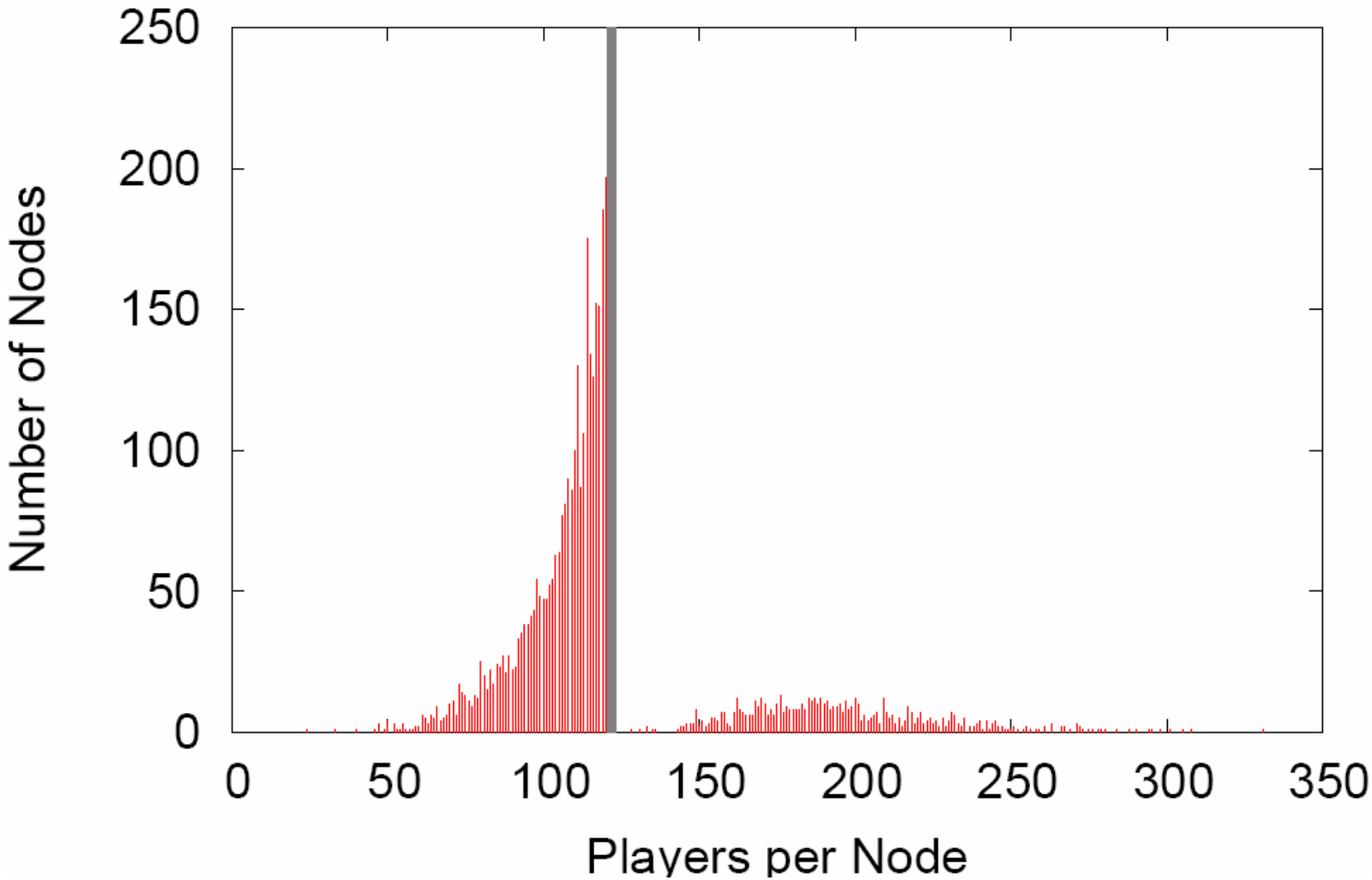


Lastbalancierung mit Virtuellen Servern

Lastbalancierung mit Virtuellen Server

- Infrastruktur-Peer nicht nur fest an einer Stelle
- Peers in CAN nur virtuell
- Peers mehrfach virtuell vorhanden
- Anpassung an Last
  - Überlastete Knoten ziehen virtuelle Server zurück
  - Austausch von Zonen zwischen belastetem und weniger belasteten Peer
- Problem: Interaktion mit der Redundanz
  - Vermeidung von Virtuellen Servern, wo ein Peer seine Daten sich selbst zur Redundanz weitergibt

Erfahrungen mit der Lastbalancierung mit VS



Verfügbarkeit und Teilen von Ressourcen

Verfügbarkeit

- Erhöht, da Ausfall einzelner Peers behandelt werden kann
  - Client, der an einem Infrastruktur-Peer hängt, muss Ausfall behandeln und den Peer wechseln können
- Datenverlust wird durch Redundanz vermieden

Teilen von Ressourcen

- P2P muss Ein- und Austritt unterstützen
- Ressource kann flexibel zu Spielwelt mit hoher hinzugefügt werden
  - Anpassung an Bedarfsschwankung manuell
  - Automatisches Pooling mit **Virtuellen Servern**
- Profit möglich durch Economy-of-Scale-Effekt

Zusammenfassung

Ziele

- Weniger Ausfallzeiten
- Weniger Kosten
- Höhere Skalierbarkeit

Methoden

- Dezentralere Infrastruktur
- Nur Vertrauenswürde Peers unter eigener Kontrolle
- Knoten müssen pluggable und unpluggable sein (Notwendig für jeden P2P-Ansatz!)
- Struktur der Wahl: CAN
- Lastbalancierung
- Teilen von Ressourcen

?

??

???

????

?????

??????

???????

????????

?????????

??????????

???????????

????????????

?????????????

??????????????