Exploiting player behavior in distributed architectures for online games

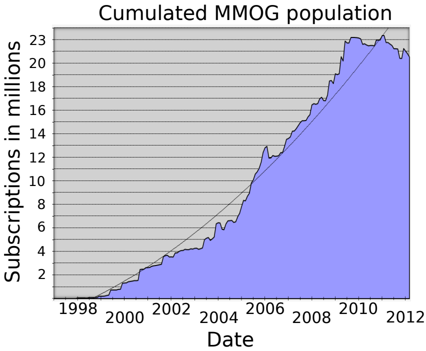
Ph.D. defense of Sergey Legtchenko

INRIA/LIP6/UPMC/CNRS

Supervision: Sébastien Monnet

Pierre Sens

Massively Multiplayer Online Games (MMOGs)



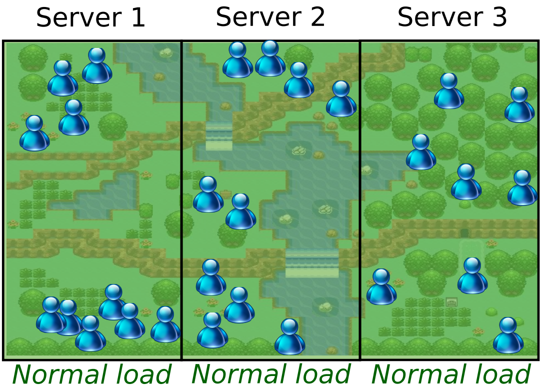
Market: $2.7 billions in 2010

MMOGs rely on expensive large-scale infrastructures

Datacenter-based:

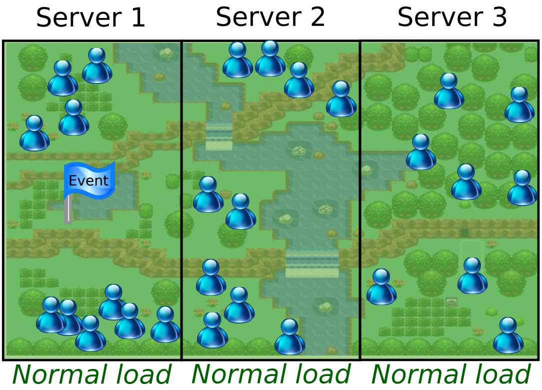
- 1000's of server-blades
- 100's of terabytes of DRAM
- **Up to 80%** of the financial revenue *[Kesselman'05]*

Problem: architectures are static, workload is dynamic



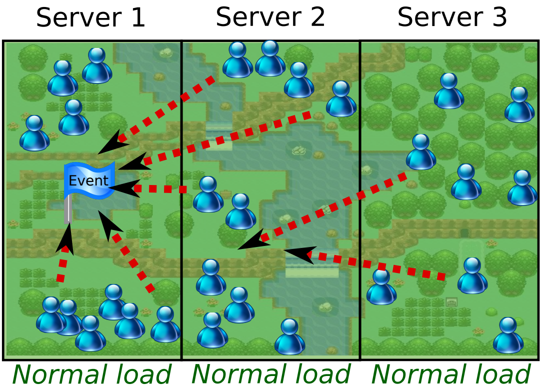
Static game partitioning unadapted to player density evolutions

Problem: architectures are static, workload is dynamic



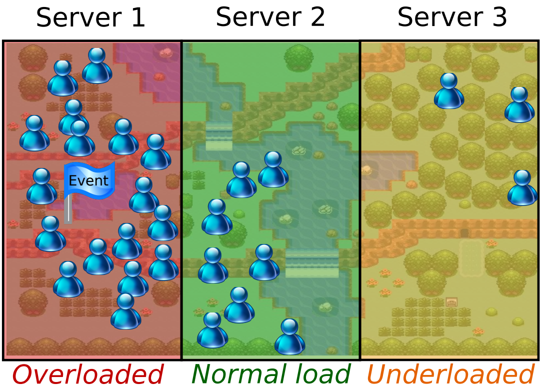
Static game partitioning unadapted to player density evolutions

Problem: architectures are static, workload is dynamic



Static game partitioning unadapted to player density evolutions

Problem: architectures are static, workload is dynamic



Static game partitioning unadapted to player density evolutions

Consequence: high cost, low efficiency

Current MMOGs:

- Lots of empty servers *[Cheslack-Postava et al, USENIX'12]*
- Lots of overloaded servers *[Varvello et al, NetGames'09]*
- Independent game instances limited to few 100's of players

- Low quality of service
- No geo-scale seamless virtual universe
- No epic battles

Academic research on MMOGs

Extensive efforts on adaptative mechanisms:

- Load balancing
- Interest management

Why no impact?

lack of robustness/performance

State of the art:

Well suited for:

Peer-to-peer (p2p)

[Colyseus, NSDI'06]
[Donnybrook, SIGCOMM'08]
[Solipsis, PDPTA'03]
[Hydra, NetGames'07]
[Walkad, IPTPS'09]

large scale MMOGs

Server based

[Sirikata, USENIX'12]
[Najaran et al., NetGames'10]
[ALVIC-NG, NetGames'08]

fast paced MMOGs

Hybrid

[Jardine et al., NetGames'08]

Contributions of the Thesis

Guideline:

improving MMOGs by making them aware of player behavior

Improving robustness:

- **BlueBanana**: increasing resilience of p2p MMOGs to player movement *[DSN10]*

Improving performance:

- **DONUT**: improving routing in large-scale p2p MMOGs with heterogeneous peer distributions *[SRDS11]*
- **QuakeVolt**: Efficient data management in server-based MMOGs with strong latency requirements *[ongoing work]*

Outline of the talk

Focusing on performance improvement:

- Part 1: Improving routing in p2p MMOGs with heterogeneous peer distributions with DONUT *(approx 20 minutes)*
- Part 2: efficient data management for large-scale virtual battlegrounds with QuakeVolt *(approx 15 minutes)*

Part 1: Improving routing in peer-to-peer MMOGs

Context: large-scale p2p MMOGs

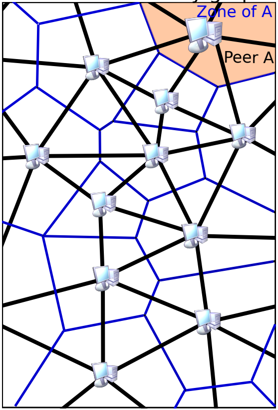
Nearest-neighbor overlays

Useful properties:

- Data locality
- Greedy routing
- Cheap
- Good fault resilience

Nearest-neighbors p2p overlays: [Mercury, VON, VoroNet, RayNet]

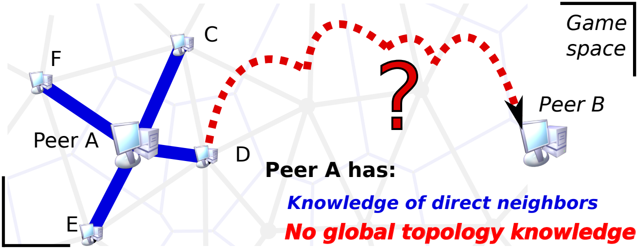
2D game partition with associated overlay graph



Problem: lack of routing efficiency

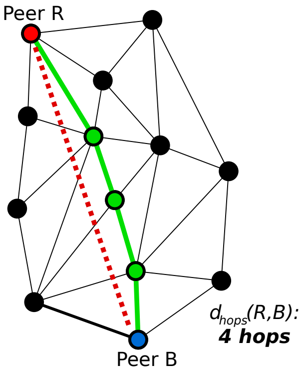
Efficient greedy routing:

- $O(log^d(N))$ with Small-World shortcuts $[Kleinberg, STOC'2000]$
- **Requires estimation of hop distances between peers**



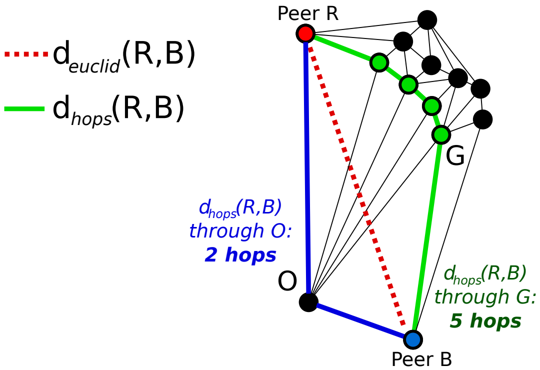
Estimating hop distance

Uniform distribution: **easy**



d_{hops} ***proportional to*** d_{euclid}

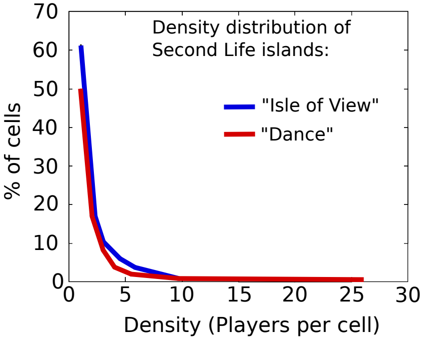
Heterogeneous distribution: **hard**



d_{hops} ***depends on density***

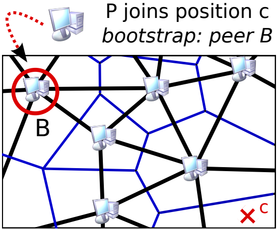
Efficiency required despite heterogeneity

Real distributions: **non-uniform**



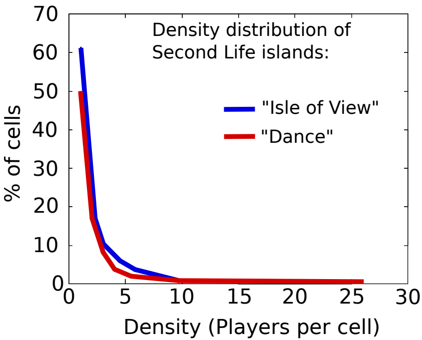
Routing in MMOGs:

- Joins
- Player teleportation



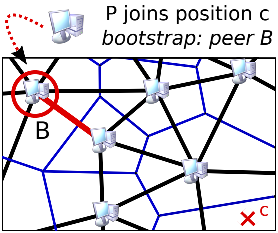
Efficiency required despite heterogeneity

Real distributions: **non-uniform**



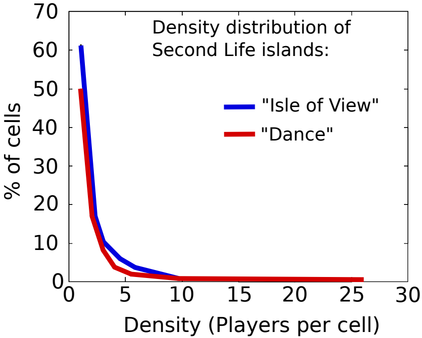
Routing in MMOGs:

- Joins
- Player teleportation



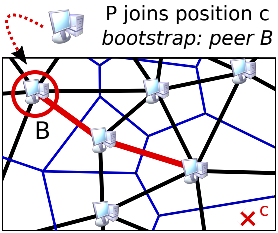
Efficiency required despite heterogeneity

Real distributions: **non-uniform**



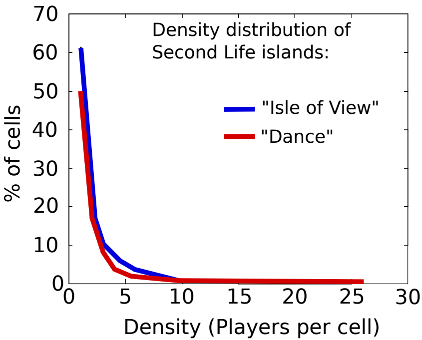
Routing in MMOGs:

- Joins
- Player teleportation



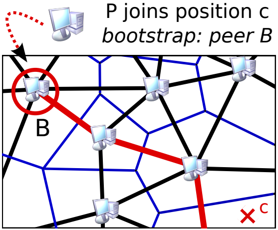
Efficiency required despite heterogeneity

Real distributions: **non-uniform**



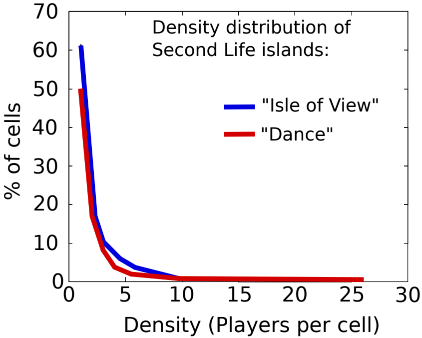
Routing in MMOGs:

- Joins
- Player teleportation



Efficiency required despite heterogeneity

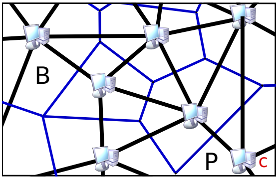
Real distributions: **non-uniform**



Routing in MMOGs:

- Joins
- Player teleportation

P joins position c
bootstrap: peer B



Contribution: DONUT, density-aware shortcut rewiring mechanism

Idea: Make peers "density-aware"

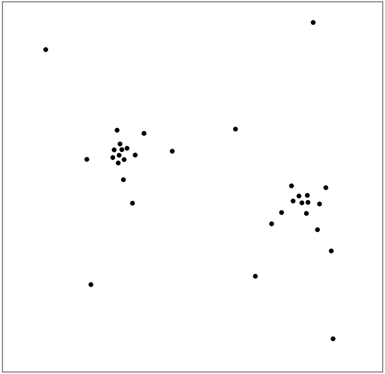
On each peer:

- Step 1: Dynamically create map of game space density
- Step 2: Use map to build density-aware small-world shortcuts

Step 1: Making peers “density aware”

Naive solution: Locally store coordinates of *all* peers

Heterogeneous host distribution
(Second Life trace snapshot)



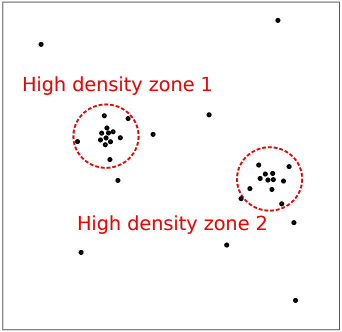
Problem: not scalable!

- Thousands of peers
- High churn rates
- Peer mobility

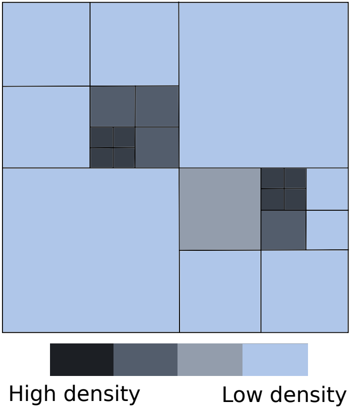
Need to approximate the distribution.

Approximate the density distribution

Distribution



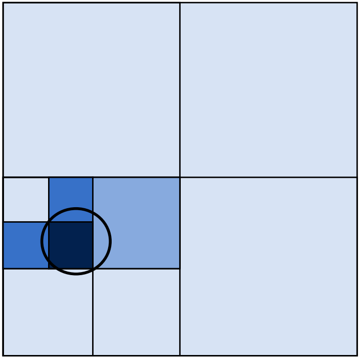
Approximation



Density map creation: two tasks

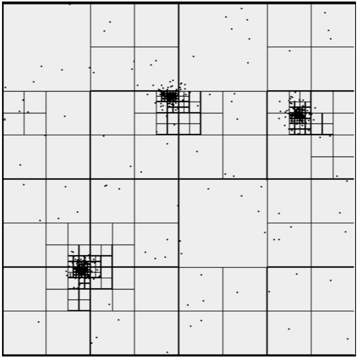
Task 1: Compute local density

Locally, using direct neighbors



Task 2: Exchange density info

Piggybacking and gossiping

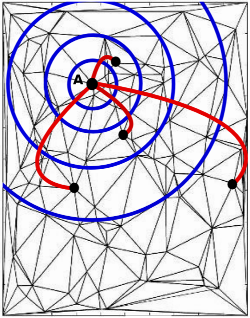


Step 2: Log-partitioning for Small-World

Shortcut link to one random peer in each partition

- Obtained distribution enables small-world property...
  [*Girdzijauskas*, *ICDEW*′05]
- ...if **hop distances** are accurately approximated

Log partitions of overlay graph by A and Small-World shortcuts of A

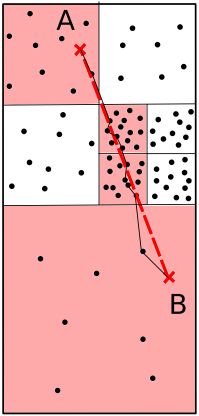


P_1, P_2,..., $P_{log(N)}$ partitions
$\forall B \in P_i$, $2^{-i} \leq d(A,B) \leq 2^{-i+1}$
with d(A,B) ***in hops***

Hop distance estimation using map

Estimate hop distance between A & B:

1. Find the regions that intersect [AB]
2. Estimate hops in each region
3. Sum hops for all regions



Uniform density inside region

Small-world shortcuts with Monte-Carlo sampling

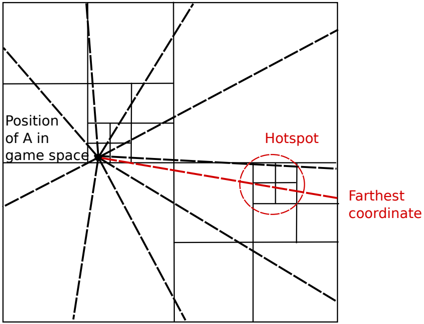
On **local** map:

1. Find max hop distance to peer
2. Approximate log partition

Cost:

- Depends on sampling precision
- **No remote operations**

Step 1: A Finds maximal hop distance



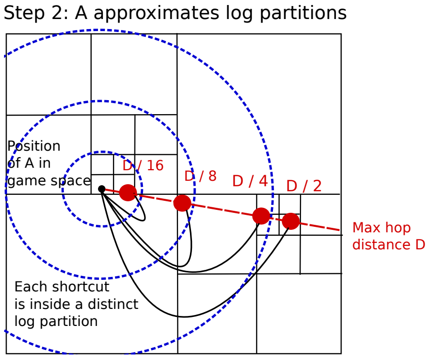
Small-world shortcuts with Monte-Carlo sampling

On **local** map:

1. Find max hop distance to peer
2. Approximate log partition

Cost:

- Depends on sampling precision
- **No remote operations**



Simulation based on real data

Latency traces:

- collected between 2500 hosts spread over the internet.

Churn traces:

- Overnet, Skype, Microsoft corporate desktops

Game space density traces:

- Traces derived from Second Life avatar distribution.

Data Credit:
[Latency: Madhyastha et al., Churn: http://fta.inria.fr/, Second Life: La et al.]

Comparison: DONUT vs

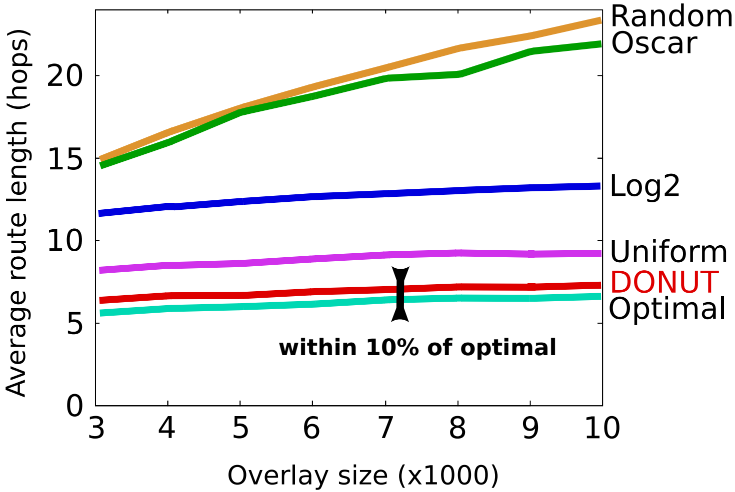
Non-small-world shortcuts:

- Random Uniform shortcut distribution

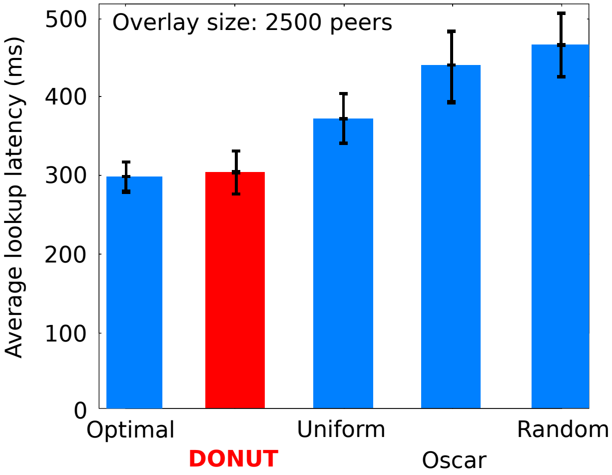
Small-world shortcuts:

- Uniform: no density information
- Oscar: density sampling with random walks
- Optimal: each peer holds the overlay graph

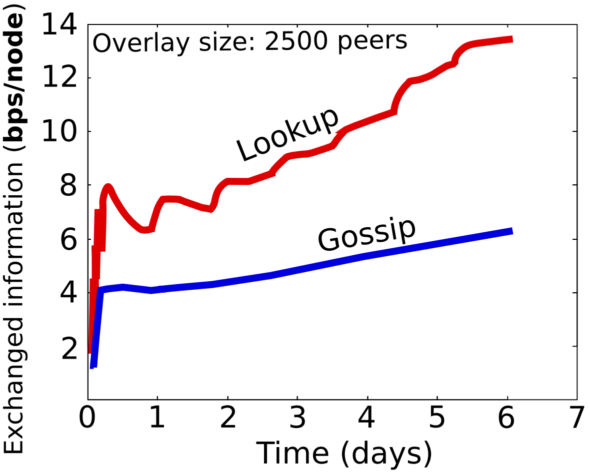
DONUT is close to optimal



DONUT latency is within 2% of optimal



Collecting global state is cheap



Contributions in p2p MMOGs:

Technique to aggregate distributed game space information.

- Cheap: a few bytes per second per peer.

Technique to build density aware shortcuts in the overlay

- Improves state-of-the-art by 20%
- Accurate: within 10% of optimal

Part 2: MMOGs with tight latency requirements

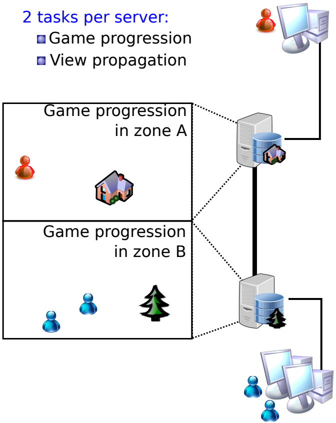
Work in progress!

Context: server-based MMOFPS

Large scale First Person Shooters

Partition-based design:

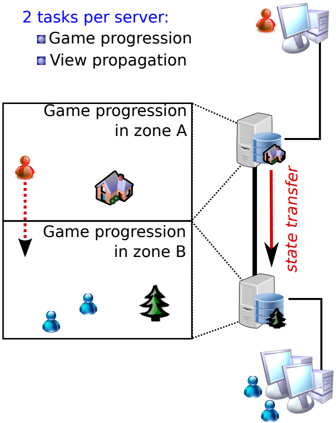
2 tasks per server:
- Game progression
- View propagation



Context: server-based MMOFPS

Large scale First Person Shooters

Partition-based design:

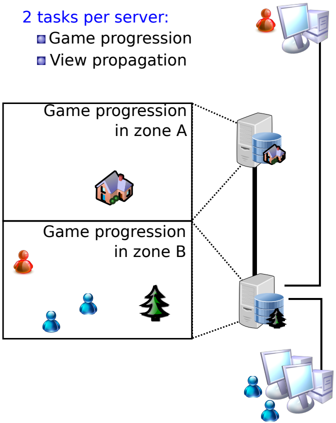


Context: server-based MMOFPS

Large scale First Person Shooters

Partition-based design:

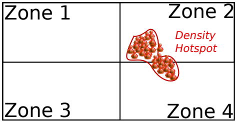
2 tasks per server:
- Game progression
- View propagation



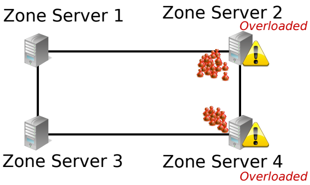
Problem: scalability ruins performance

Scalability mechanism:

Game space



Logical topology



Ensures scalability...

- Fair load balancing
- Elastic horizontal scalability

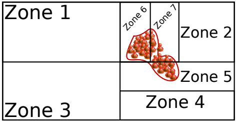
...but fragments game space:

- Increase of inter-zone transfers
- Increase of network traffic

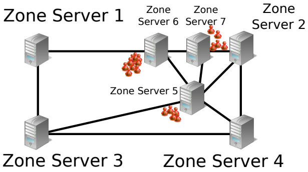
Problem: scalability ruins performance

Scalability mechanism:

Game space



Logical topology



Ensures scalability...

- Fair load balancing
- Elastic horizontal scalability

...but fragments game space:

- Increase of inter-zone transfers
- Increase of network traffic

What to do?

Fragmentation is harmful, so how to:

- Ensure scalability
- Limit fragmentation

Contributions:

- Scalability analysis of Quake III, a popular server-based game
- Scalable architecture that limits game space fragmentation

Quake III First Person Shooter

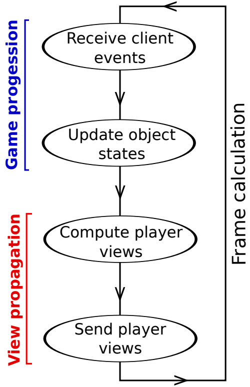
Not quite an MMOFPS:

- Single server
- Limited to less than 100 players

Benefits of using Quake III:

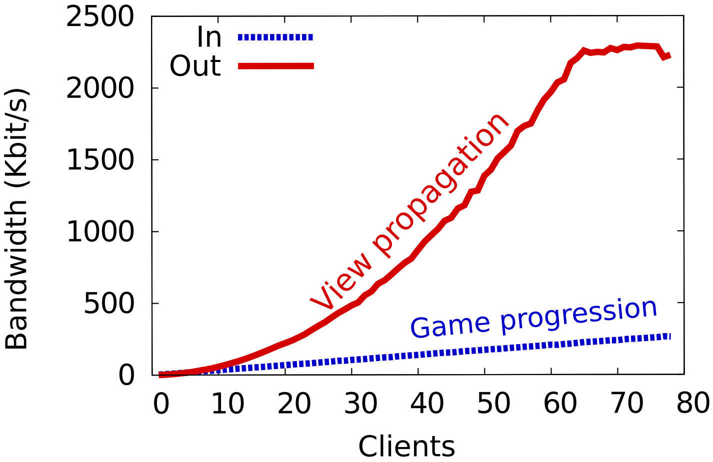
- Still popular and open source
- High responsiveness requirements
- Design similar to many MMOG servers

Q3 Server execution tasks

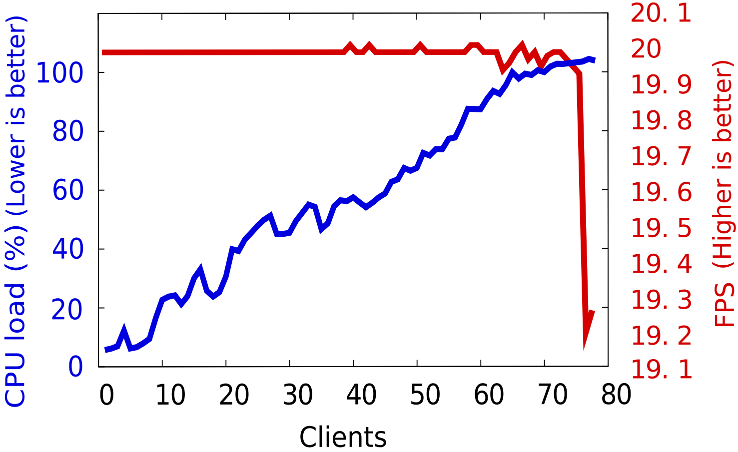


20 frames per second (fps) when not overloaded

Quake III bandwidth consumption



Quake III CPU consumption/framerate



QuakeVolt MMOG architecture

Result of scalability analysis:

Game progression scales much better than view propagation

Idea: Decouple Game progression from View propagation

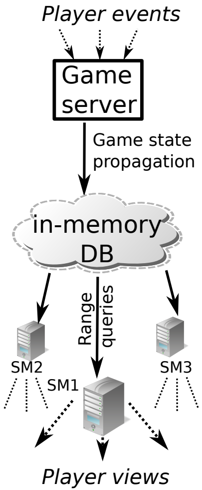
3-tier architecture:

- Quake III server ensures Game progression
- Low latency (in-memory) database stores game state
- View propagation is delegated to a set of *Snapshot Mirrors*

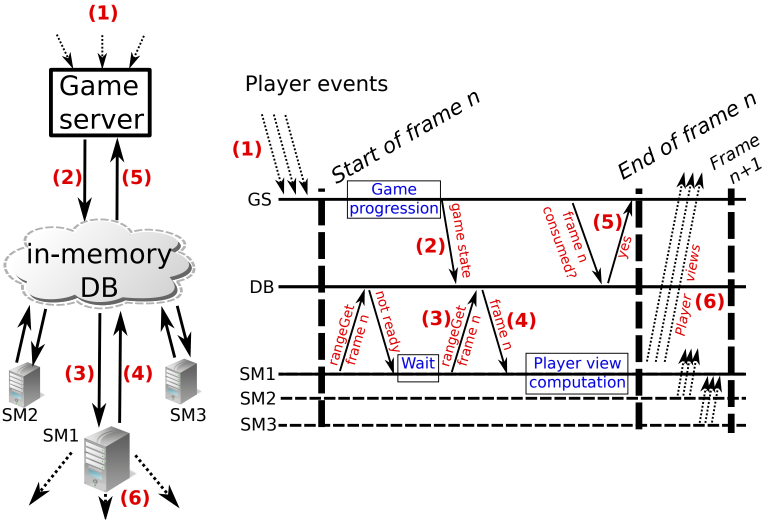
Benefits of the design

Eases dynamic adaptation to workload:

- Limited data fragmentation
- Easy dynamic reconfiguration
- Elastic scale out
- No client-side modification



Distributed frame computation



Promising work

- VoltDB used as in-memory database
- Runs on our cluster and on Grid'5000 with 10's of clients
- Native protocol unmodified (legacy Q3 clients can connect)
- Already playable (and fun!)
- Easy to implement (modification of $<$ 0.03% of Q3 code)

Conclusion

Integration of player behavior is good for MMOG architectures

Contribution summary

1. Better robustness of p2p MMOGs through prediction of player trajectories *(not detailed in the talk)*
2. Better routing performance in p2p MMOGs thanks to player distribution monitoring
3. Less data fragmentation in server-based MMOGs thanks to accurate player view management

Behavior-aware mechanisms are lightweight and generic

Future work and perspectives

Short term:

- DONUT on top of existing overlays for range querying
- Extensive evaluation of QuakeVolt
- Adaptative mechanisms for QuakeVolt elasticity and scale out

Long term: exploiting player behavior for future MMOGs

- Improvement of player behavior modeling
- Design of enhanced monitoring techniques
- Better integration of player behavior at systems level

Thank you for your attention!

Publications related to subject:

- DONUT: Building Shortcuts in Large-Scale Decentralized Systems with Heterogeneous Peer Distributions. *S.Legtchenko, S.Monnet and P.Sens [SRDS11]*
- BlueBanana: resilience to avatar mobility in distributed MMOGs. *S.Legtchenko, S.Monnet and G.Thomas [DSN10]*

Other publications:

- RelaxDHT: a churn-resilient replication strategy for peer-to-peer distributed hash-tables. *S.Legtchenko, S.Monnet, P.Sens and G.Muller [TAAS11]*
- Churn-Resilient Replication Strategy for Peer-to-Peer Distributed Hash-Tables. *S.Legtchenko, S.Monnet, P.Sens and G.Muller [SSS09]*