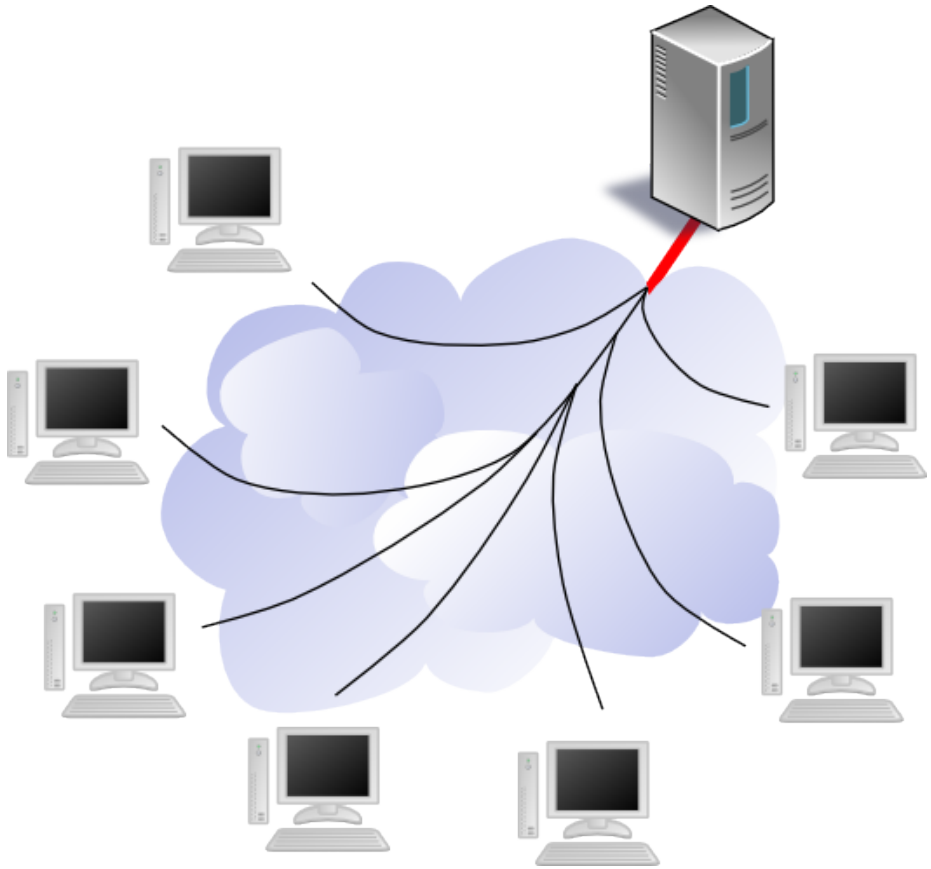
A Hybrid Architecture for Massively Multiplayer Online Games

Jared Jardine and Daniel Zappala
Internet Research Lab
Computer Science Department
Brigham Young University

The Seventh Annual Workshop on Network and Systems Support for Games (NetGames)
22 October 2008

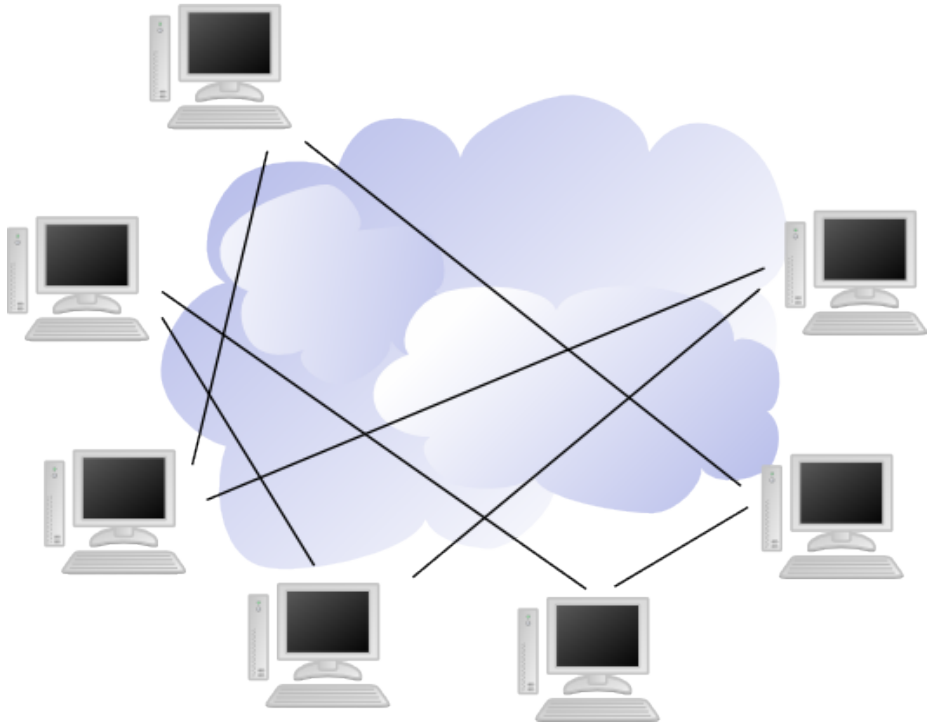
The Bandwidth Bottleneck

- bandwidth is a bottleneck for client-server game architectures
- peer-to-peer architectures
  - distribute bandwidth
  - lower latency
  - provide greater fault tolerance



The Bandwidth Bottleneck

- bandwidth is a bottleneck for client-server game architectures
- peer-to-peer architectures
  - distribute bandwidth
  - lower latency
  - provide greater fault tolerance



Advantages of a Client-Server Architecture

- server controls game state
  - simple to provide state consistency – a single source for state updates
  - simple to provide event ordering – order moves as they reach the server
  - (relatively) simple to prevent cheating – the server is a central authority for who did what when
- peer-to-peer architectures make all of this much harder
- any peer can be malicious

A Hybrid Architecture

- state-changing moves: any move that changes state, such as items, boundaries, attacks
- positional moves: any move that does not affect game state, such as walking
- hybrid of client-server and peer-to-peer
  - use server to distribute state-changing moves
  - use a hierarchy of peers to distribute positional moves
- goals
  - use centralized control to maintain consistency and prevent cheating
  - use peers to offload bandwidth from the central server
- challenges
  - ensuring peers are capable of distributing positional moves
  - detecting and removing malicious or poorly-performing peers

Contributions

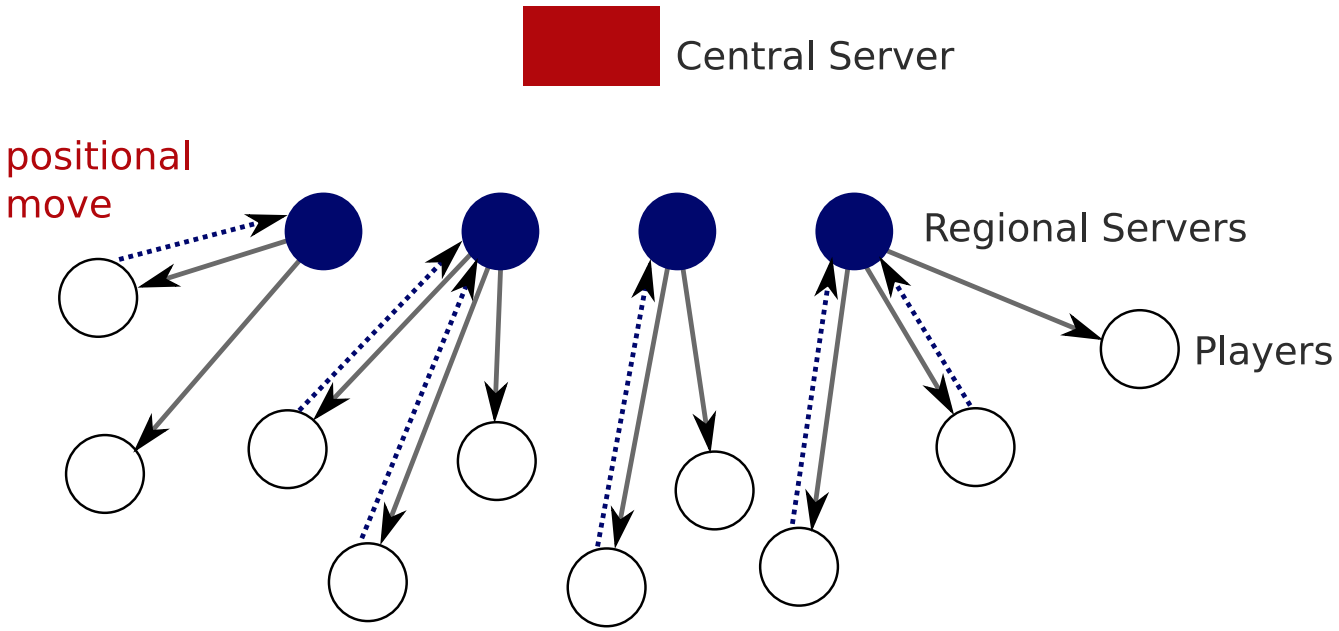
- experiments with 50 automated players on PlanetLab
- preliminary results show hybrid architecture scales better than client-server
- performance advantages depend on having enough capable players to participate in the distribution hierarchy

Related Work

- DHT Architectures
  - publish-subscribe for state changes
  - map game state to a node in the DHT
- hierarchical architectures
  - divide a game into regions
  - assign players as regional servers – manage state for the region
- both approaches susceptible to misbehaving or malicious users
  - allow players to store and manage game state
  - allow players to distribute updates
- hierarchy is more easily controlled

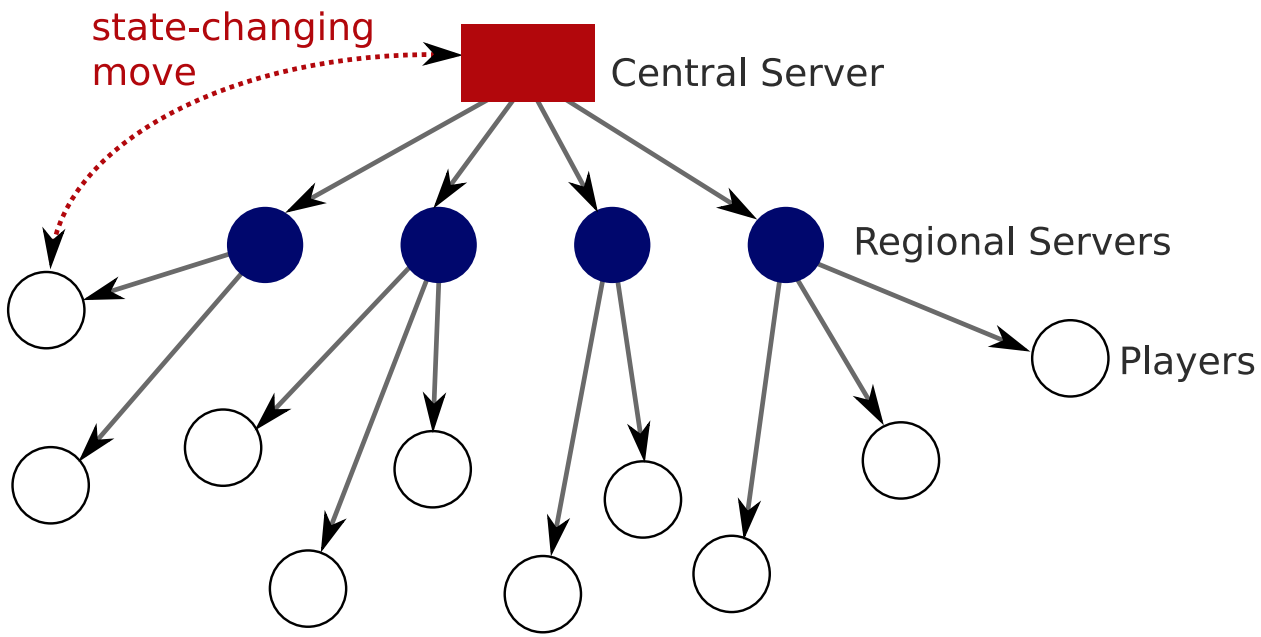
Hybrid Architecture

- divide the game into regions
- assign peers to act as regional servers
- players make positional moves through regional servers (e.g. walking across a room)



Hybrid Architecture

- divide the game into regions
- assign peers to act as regional servers
- players make state-changing moves through central server (e.g. picking up a coin off the floor)



Bandwidth Savings

- positional moves
  - central server uses no bandwidth
  - savings proportional to the percentage of positional versus state-changing moves
  - in role-playing games, a significant percentage of moves are positional
- state-changing moves
  - central server sends updates to regional servers instead of all players
  - savings proportional to the average density of a region

Measuring Capable Players

- regional servers need
  - enough bandwidth capacity
  - low latency to central server
- measurements
  - pathrate: estimate the capacity of the path between central server and regional server
  - ping: measure the lateny between central server and regional server
- requirements
  - minimum bandwidth, latency
  - no NAT
  - group players into bins, with bandwidth preferred over latency
- monitoring
  - regional servers send periodic positional updates
  - replace regional server if it misses 3 consecutive updates

Security Concerns I

- ability to cheat significantly limited compared to peer-to-peer architectures
  - state changes handled by central
  - security concerns limited to regional servers
  - reduced subscription fees to well-behaved servers to provide incentive to avoid cheating

1. denial of service: drop or delay state updates
   - players monitor regional server performance
   - report complaints to central server
   - central server may replace regional server

Security Concerns II

2. inside information: join own region, see how other players move before making its own move; join multiple regions
   - changing regions is considered a state-changing move
   - prevent regional server from moving into its own region
   - prevent players from joining more than one region
3. collusion: give advantages to some players in its region
   - central server conduct audits of state-changing moves
   - verify the player had enough time to move into area where state-changing move is made
   - remove offending servers from game

Treasure Hunt Game

- two opposing teams of players
- a player from each team assigned a particular treasure to find and return to home base
  - given region where treasure is located
  - must travel to region and search for treasure there
  - given updates of what they can see as they move
  - move 1/2 speed when carrying treasure
- players that meet fight for treasure
  - winner determined by previous victories, team score, random factor
  - winner keeps treasure, loser returns to base

Experimental Setup

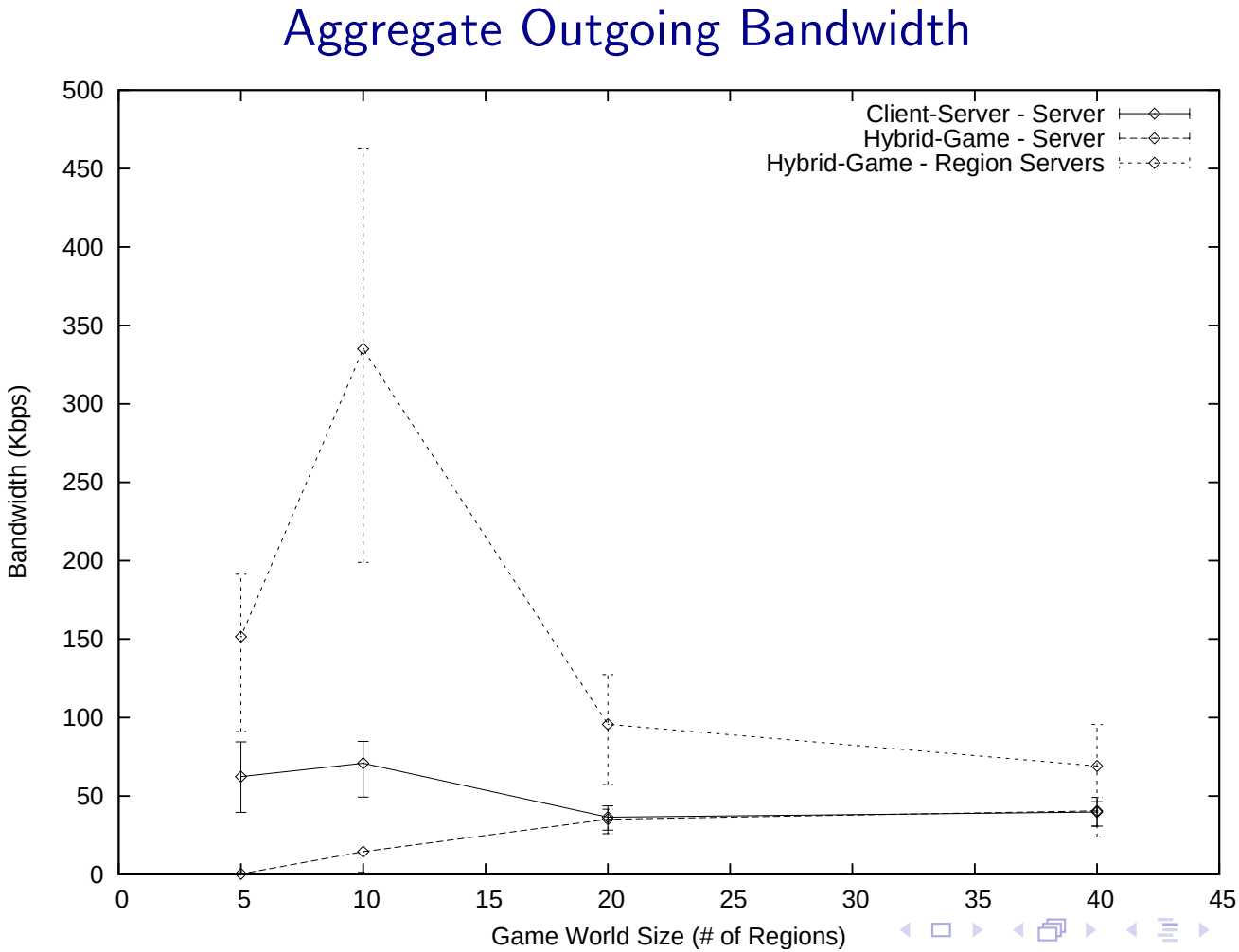
- PlanetLab experiments
  - 50 automated players, 10 minute game
  - rate limit players to 10 Mbps, 1 Mbps, 56 Kbps
  - 5 repetitions of each experiment
- workload
  - 5 moves per second maximum
  - player updates: 20 bytes
  - state updates: 320 bytes (sizes based on recent study)
- compare client-server and hybrid architectures

Varying Region Density

- 50 players
- reduce size of game world: 40 to 5 regions
- 12/50 players capable of begin regional servers

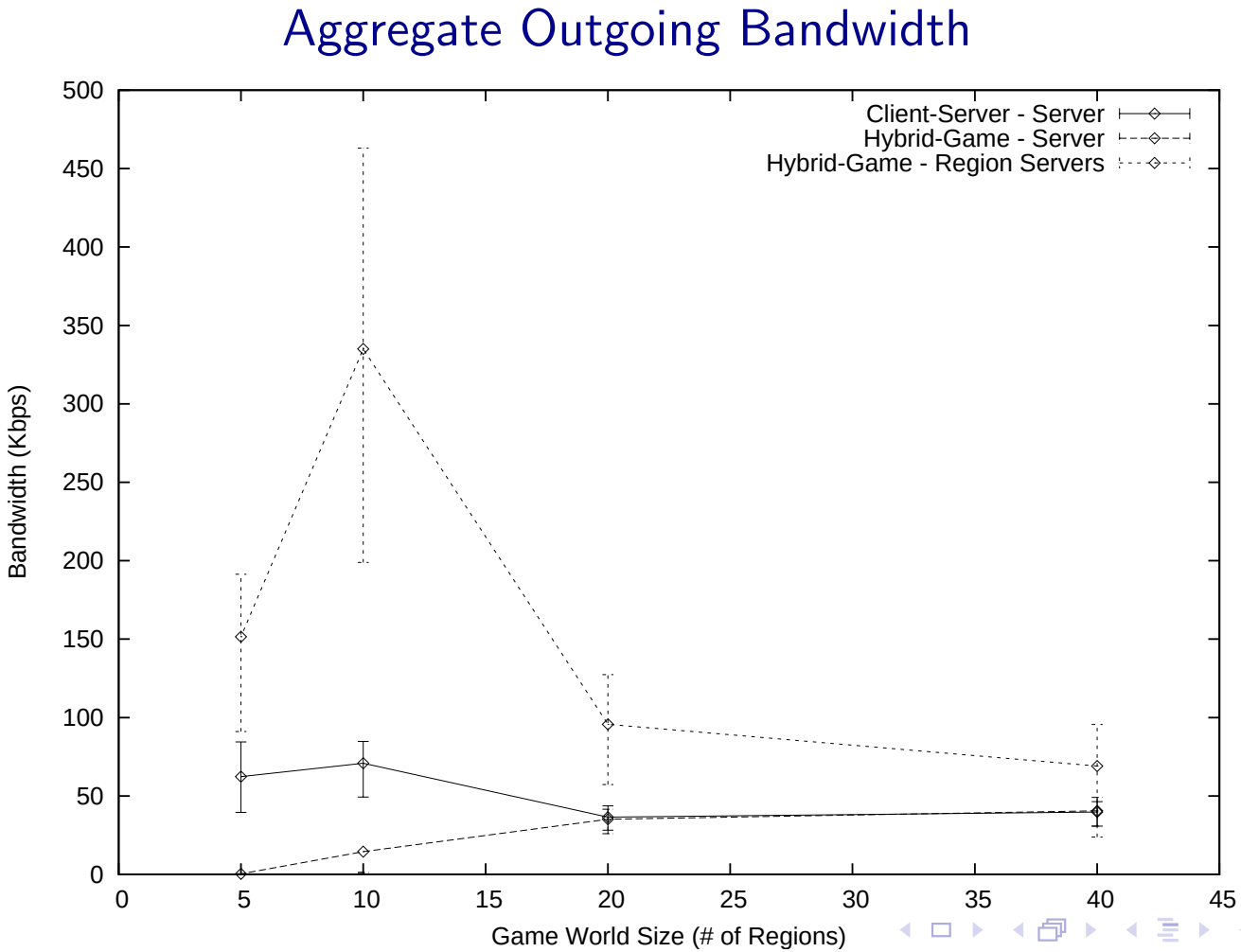
Varying Region Density

- reduces central server bandwidth once there are enough regional servers



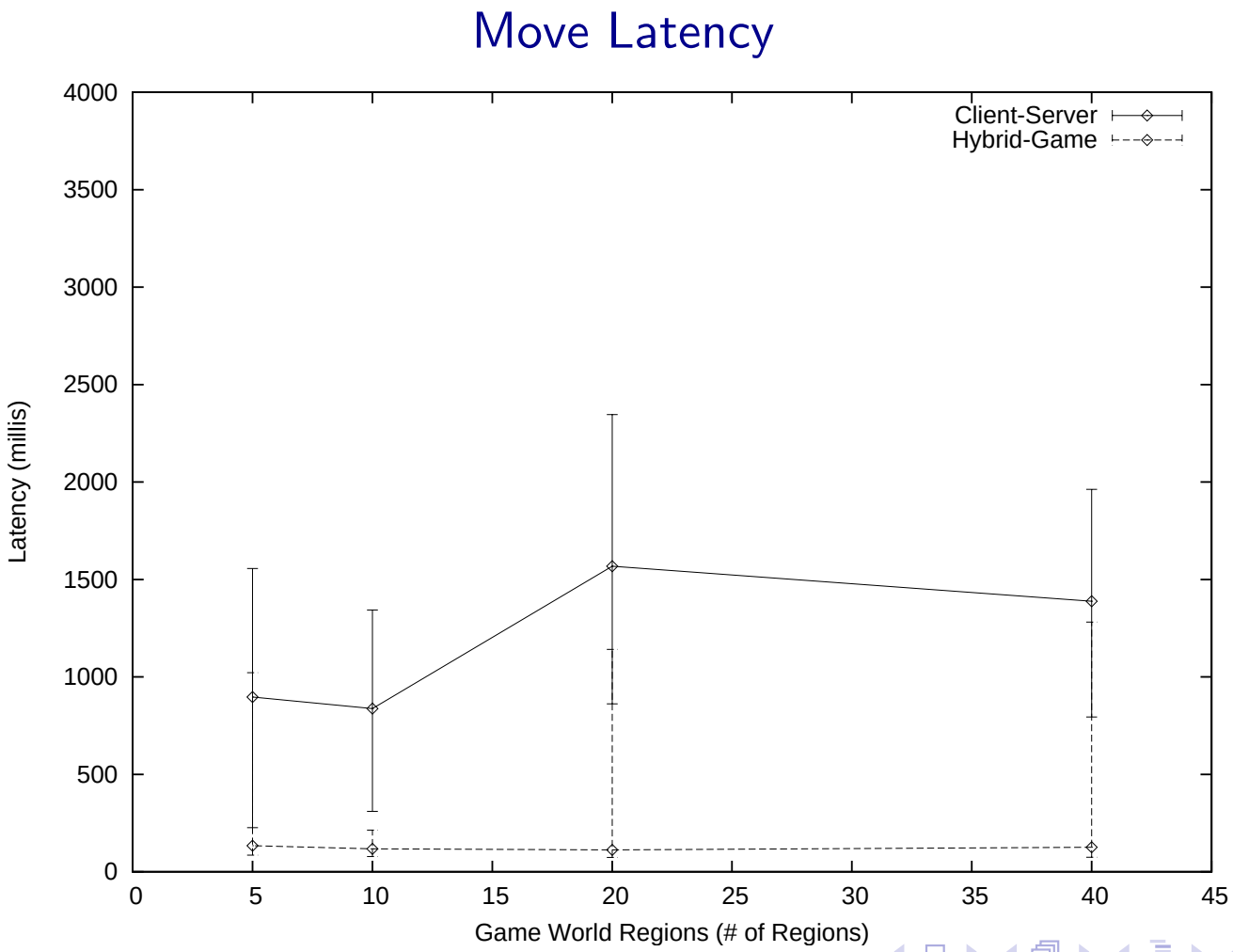
Varying Region Density

- high regional server bandwidth indicates handling more moves per second (2.5 versus 1)



Varying Region Density

- reduces latency relative to client-server

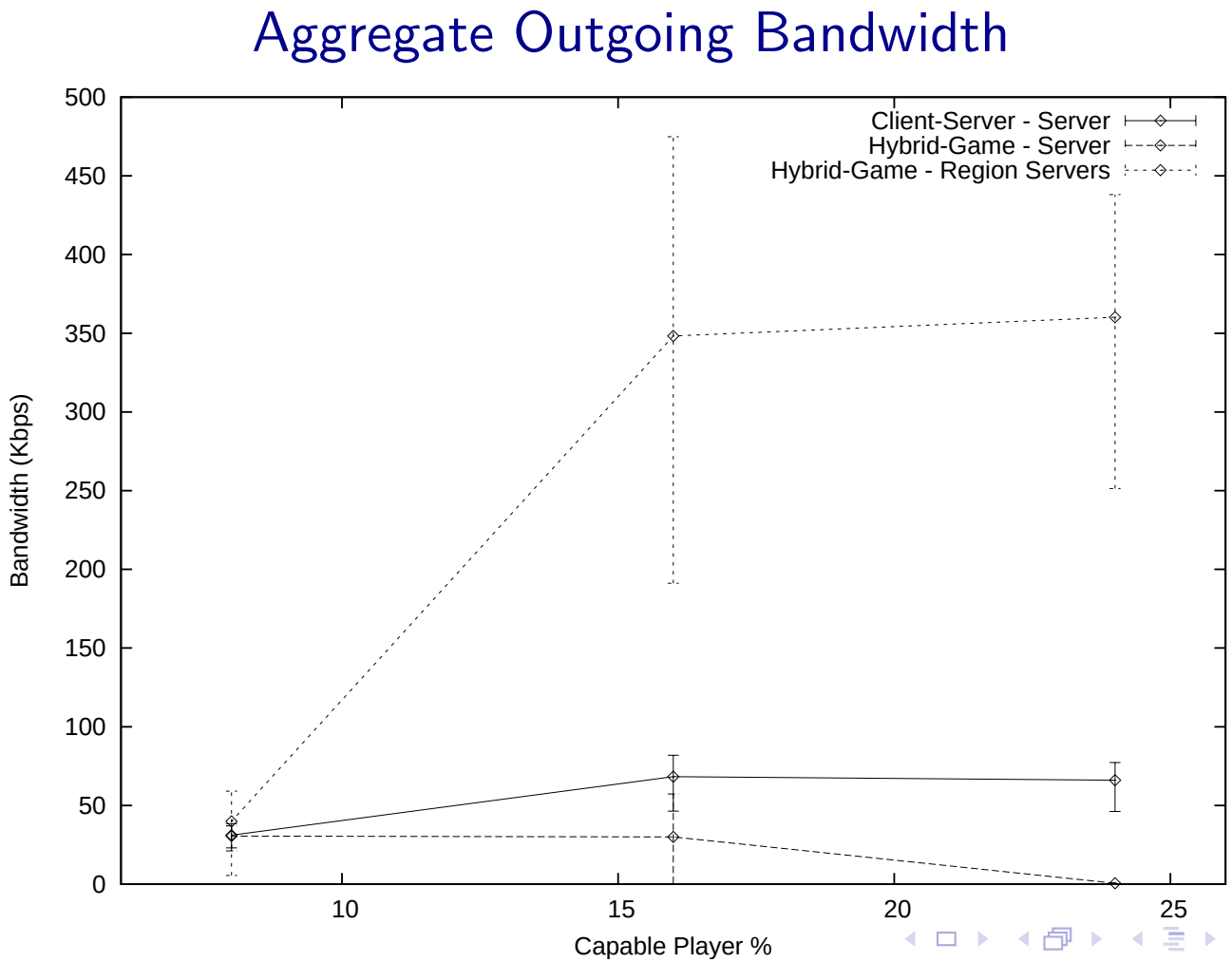


Varying Player Capability

- 50 players, 8 regions
- vary capable players: 4 to 12

Varying Player Capability

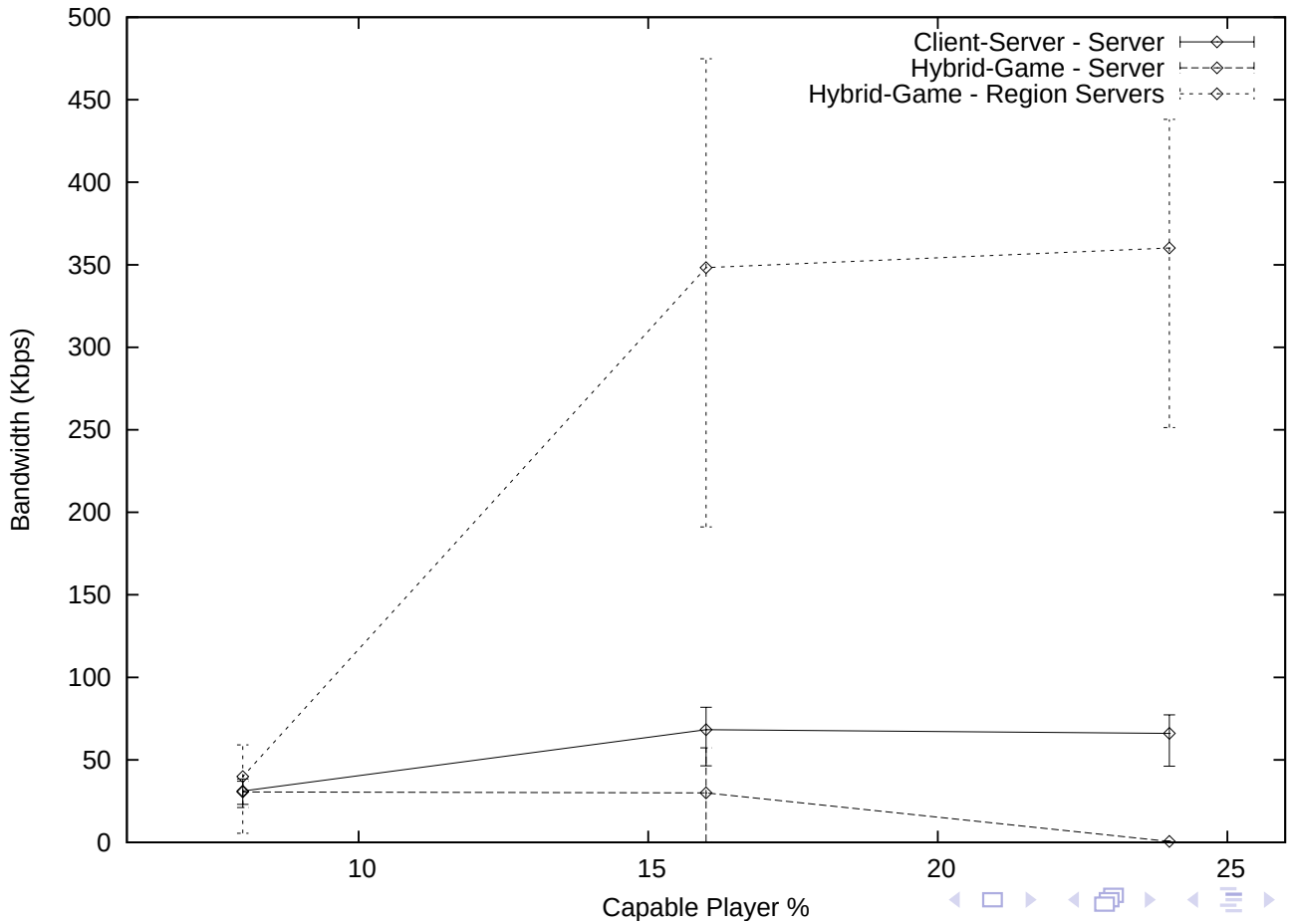
- more capable players means reduced bandwidth on central server



Varying Player Capability

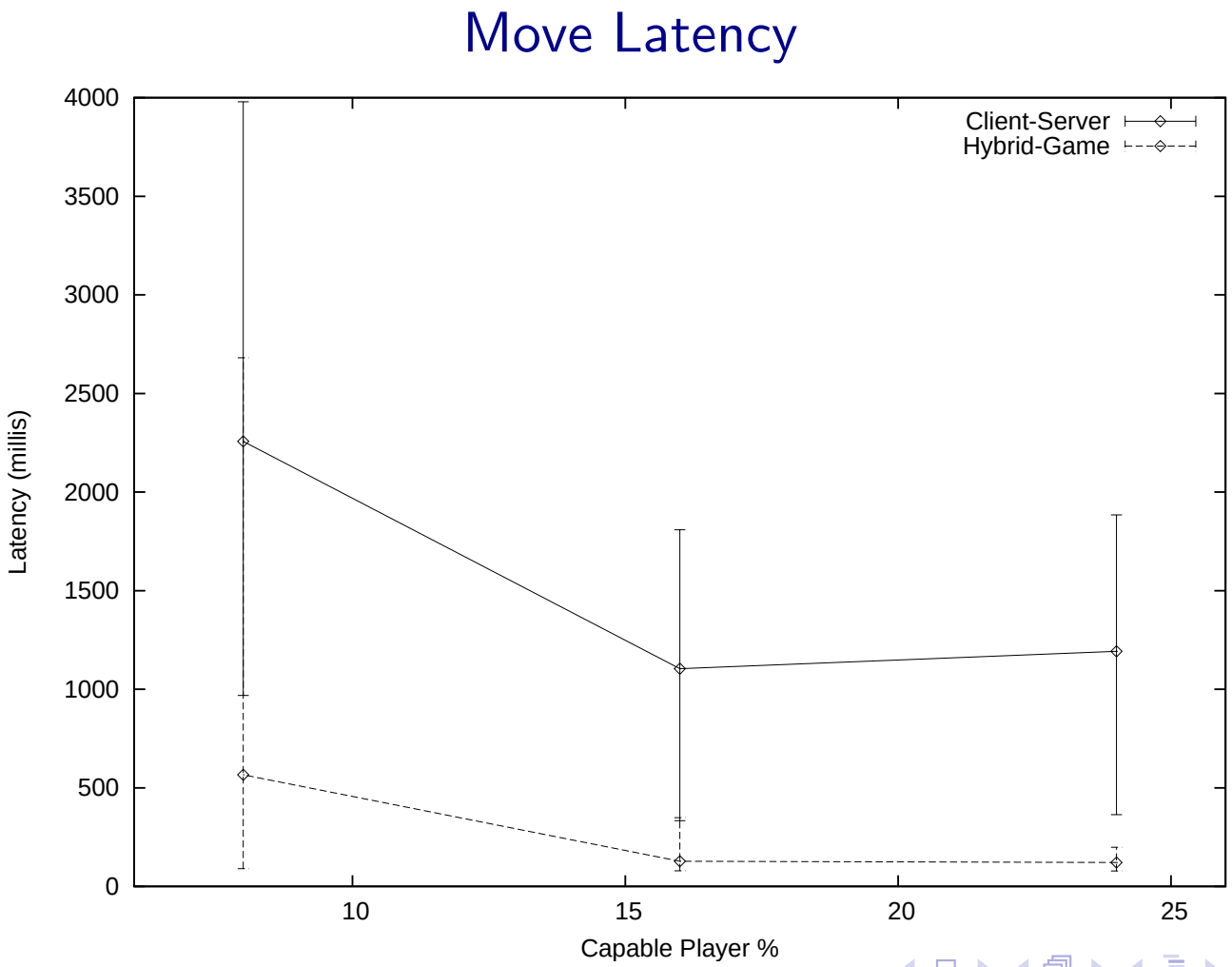
- high regional server bandwidth indicates handling more moves per second (2 versus < 1)

Aggregate Outgoing Bandwidth



Varying Player Capability

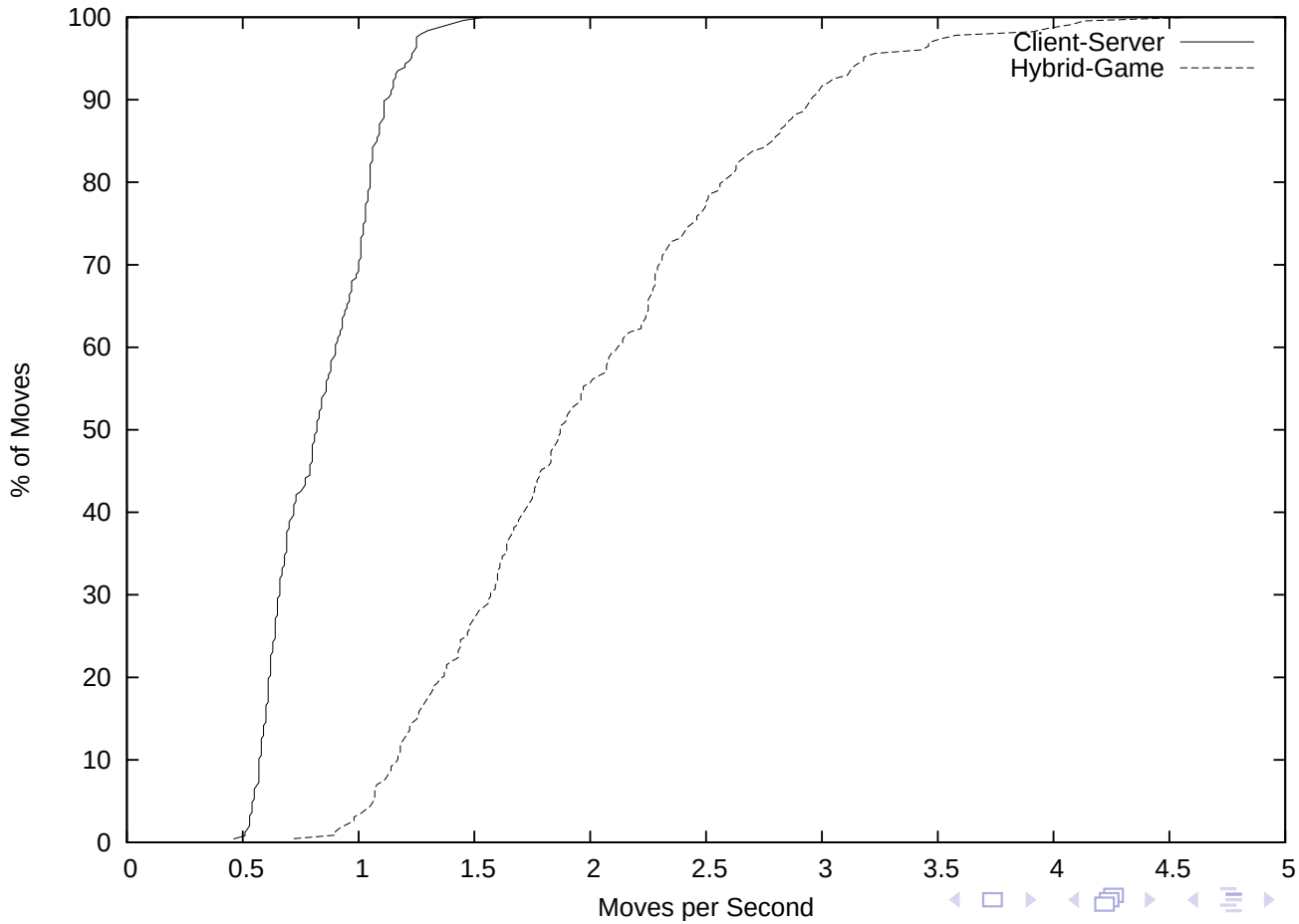
- more capable players means reduced latency



Move Latency

- 12/50 capable players, 8 regions
- hybrid: up to 4.5 moves per second

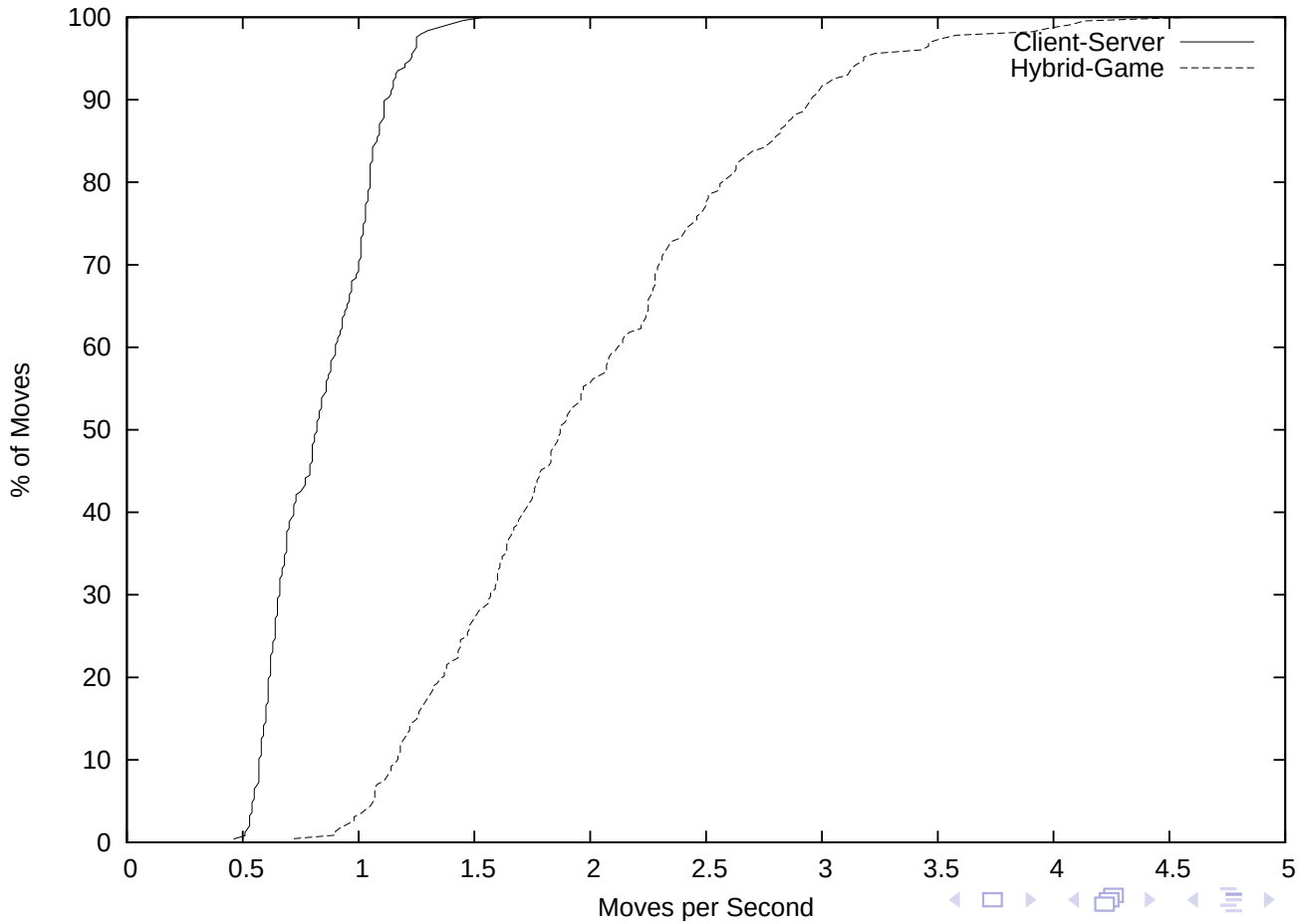
CDF of Move Latency



Move Latency

- 10th percentile for hybrid = 80th percentile for client-server
- some players may be stuck with high-latency paths

CDF of Move Latency



Future Work

- relatively poor performance in terms of moves per second
  - both architectures use same code base
  - improve implementation to make more realistic performance study
- try to decrease latency while coping with PlanetLab variance in capabilities and performance
- further examine
  - scalability
  - failure scenarios
  - cheating prevention
- integrate into an existing game