A Distributed Market Framework for Large-scale Resource Sharing

Marian MIHAILESCU
Yong Meng TEO

Department of Computer Science
National University of Singapore

Euro-Par 2010, August 31st - September 3rd, 2010, Ischia-Naples, Italy

Overview

- Background and Motivation
- Proposed Distributed Market Architecture
  - Distributed Auctions Mechanism
  - Deadlock-free Auction Protocol
- Results
  - Vertical Scalability
  - Horizontal Scalability
- Conclusions and Remarks

Introduction

- Distributed systems converge towards large-scale resource sharing
  - Peer-to-peer, grid, cloud computing
- Users sharing resources are rational
- Users require multiple resource types
- Dynamic demand and supply

Solution: market-based approaches for the allocation of shared resources

Market-based models for the allocation of shared resources

- Combinatorial Auctions
  - Computationally hard
  - Approximations are not efficient and strategy-proof
- Double Auctions
  - Centralized information
  - Single resource type allocation
- Bargaining
  - High communication costs
- Fixed pricing
  - Price does not adapt to changes in demand and supply

Market-based models for the allocation of shared resources

- Efficiency is user-centric: Economic (Pareto) efficiency
  - Computationally hard (e.g. combinatorial auctions)
- Strategy-proof: users are incentivized for being truthful
  - Centralized mechanism (e.g. double auctions)
- Increasing the number of users or the number of resource types leads to large allocation times

Motivation of this work: scalability of the resource pricing mechanism

Dynamic pricing mechanism [ICPP09]

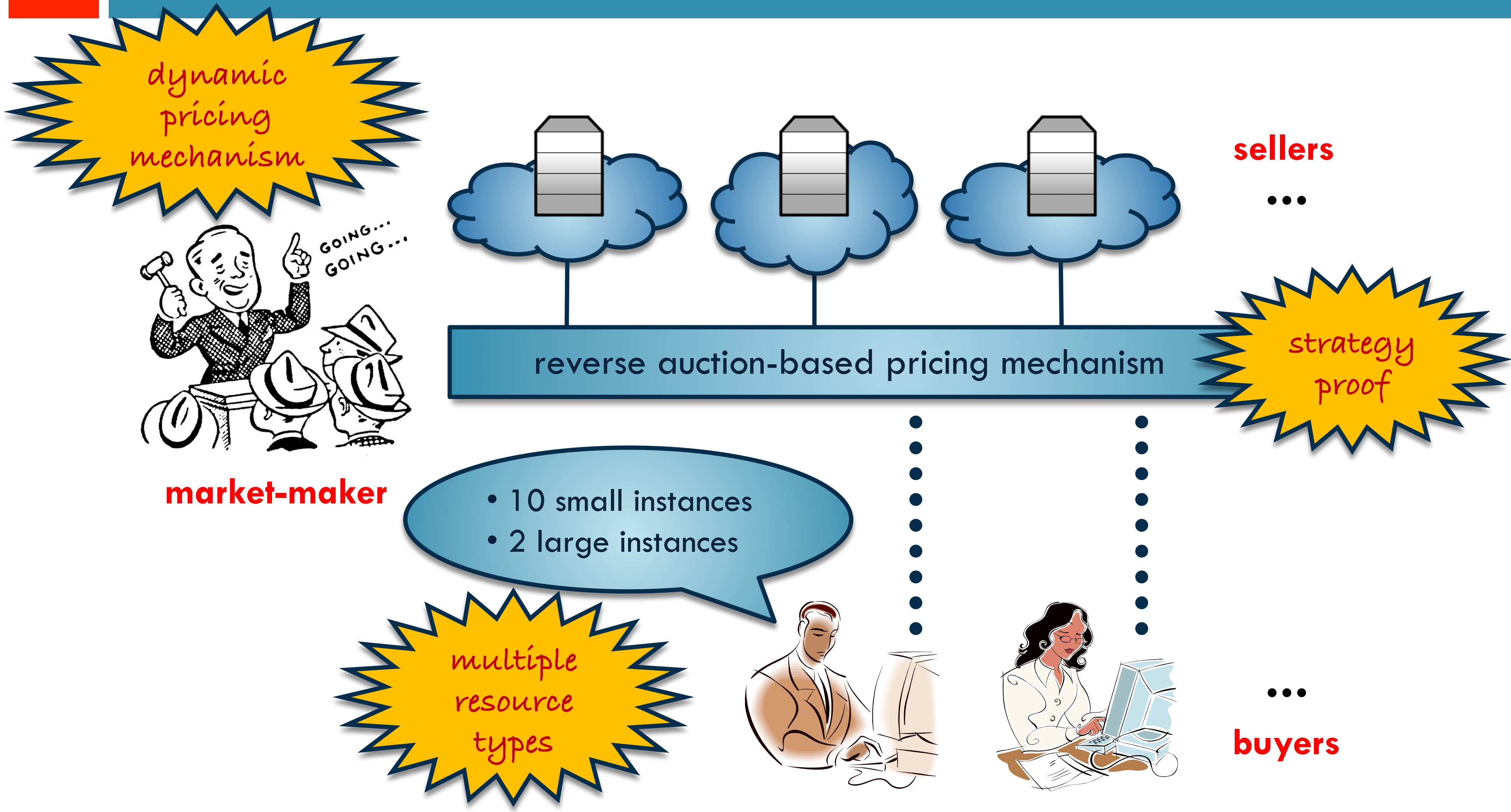
- Payment functions

$$p_s = \begin{cases} 0, & \text{if seller } s \text{ does not contribute with} \\ & \text{resources to satisfy the request} \\ c_{M|s=\infty} - c_{M|s=0} & \\ & \text{if seller } s \text{ contributes with} \\ & \text{resources to satisfy the request} \end{cases}$$

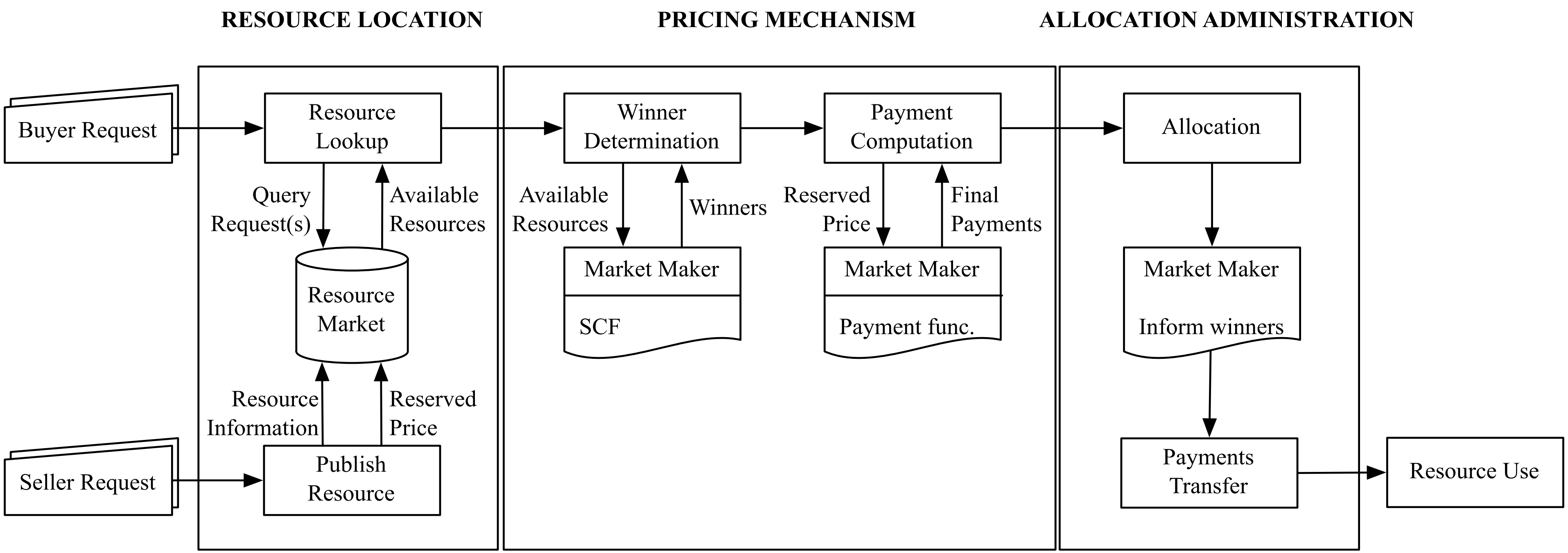
$$p_b = -\sum_{s \in S} p_s$$

- Properties
  - **Economic**
    - Strategy-proof
    - Budget balance
    - Multiple resource type allocations
  - **Computational**
    - Low algorithm complexity

Dynamic pricing mechanism



Market architecture overview



Proposed distributed framework

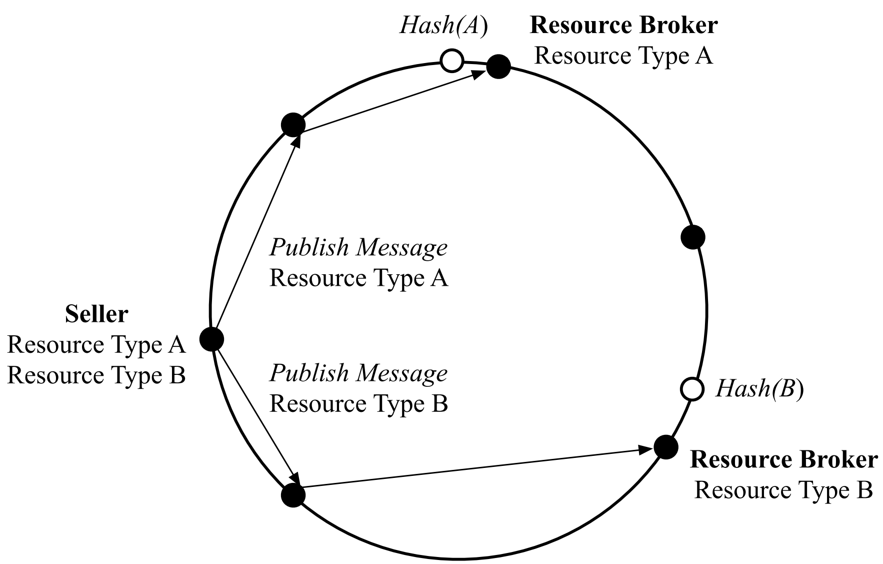
- Maintain strategy-proof
  - Payment computation requires complete information to structure correct incentives
- Add scalability
  - Distribute resource information in an overlay network **according to resource type**
  - Divide a request into DHT lookup operations **for each resource type**
  - Peer roles: *seller, buyer, resource broker, request broker*

Peer roles

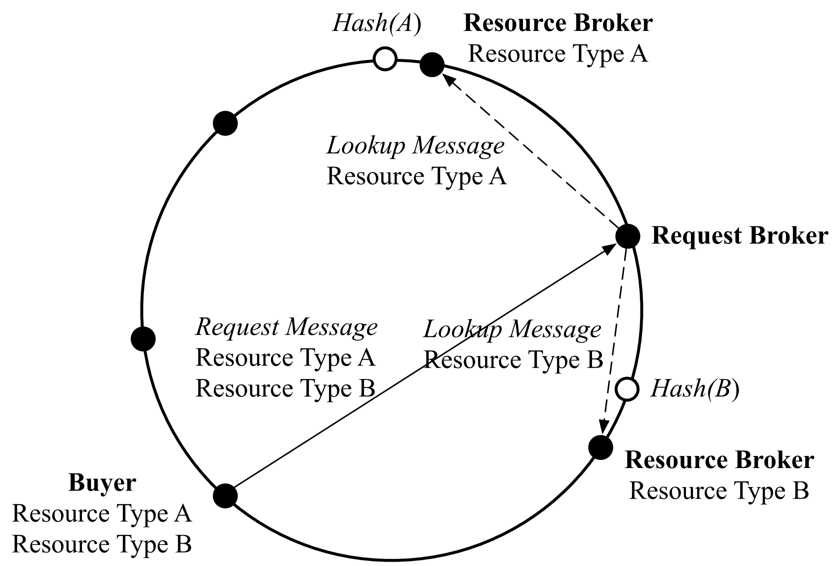
- *Seller*
  - Publishes resources using the DHT store interface
  - Uses the hash of the resource type as the key
- *Resource Broker*
  - Peer with the identifier closest to a resource type key
  - Maintains list of available resources for the respective resource type
  - Computes payments for the respective resource type
- *Buyer*
  - Sends a request message using the hash of all resource types as the key
- *Request Broker*
  - Peer with the identifier closest to a request key
  - Sends lookup requests for each resource type in the request
  - Receives payments from resource brokers and decides if allocation takes place

Peer roles

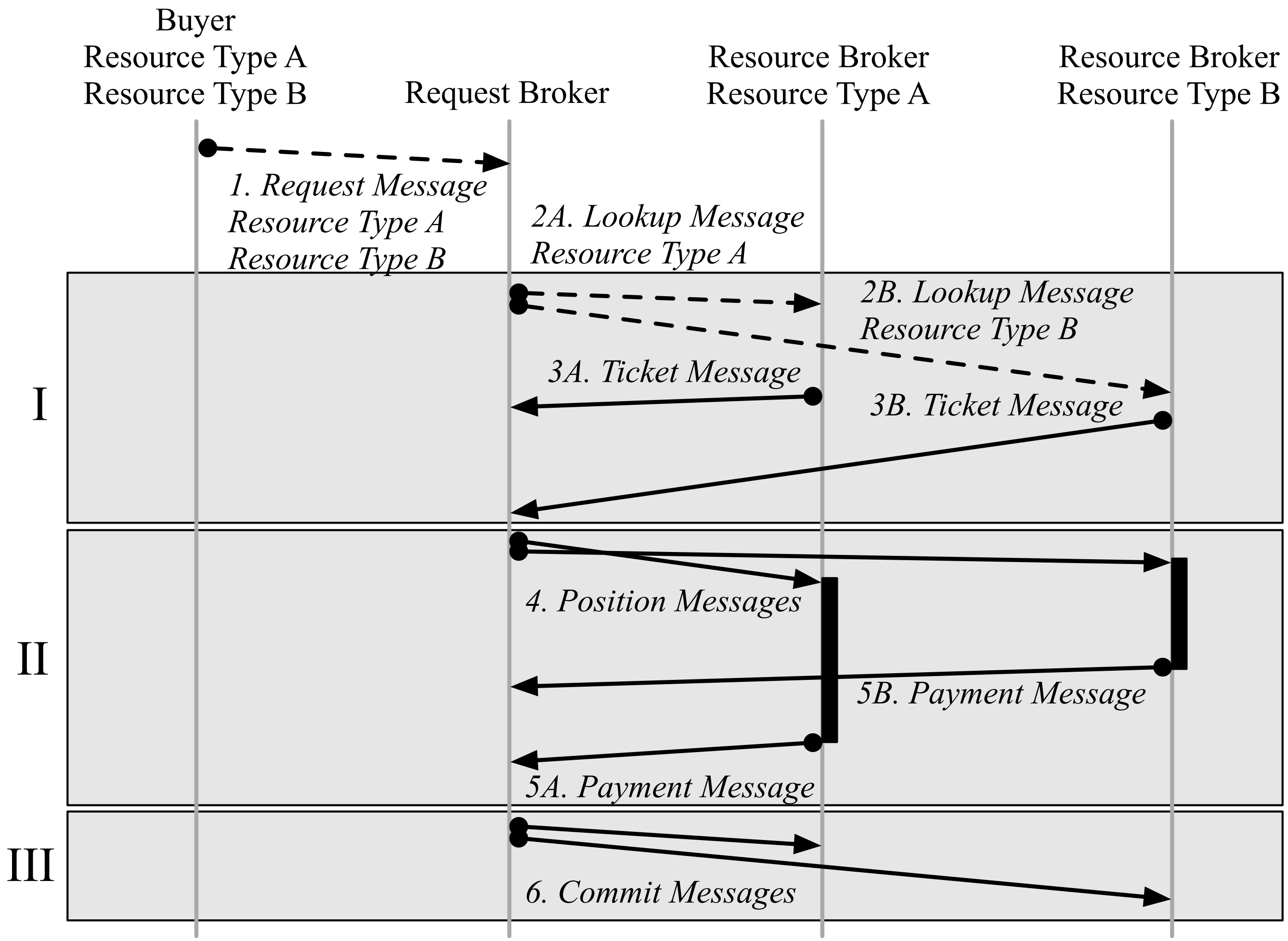
Seller and Resource Brokers



Buyer and Request Broker



Deadlock-free auction protocol



Analysis

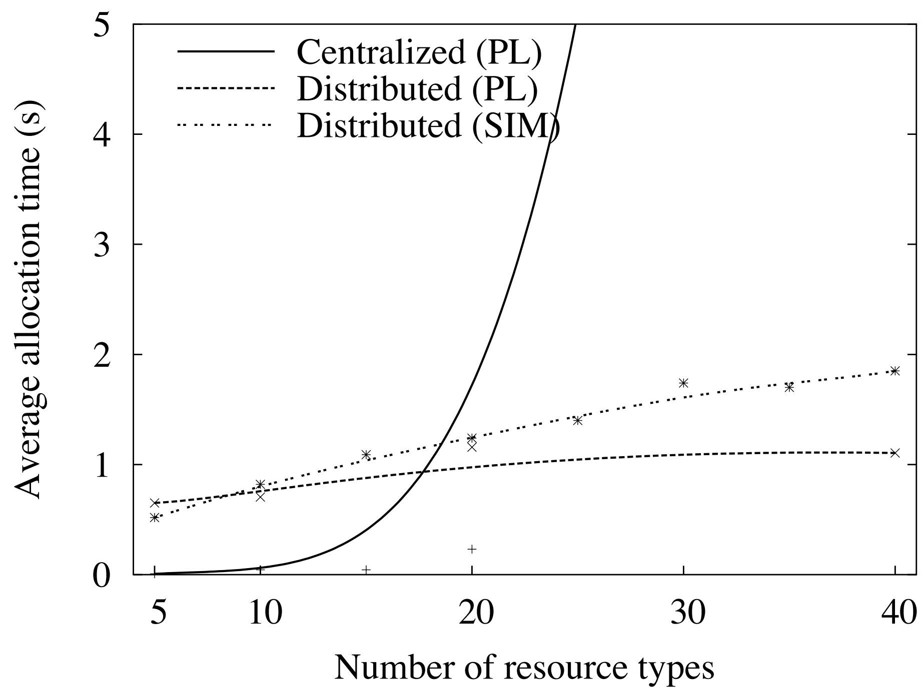
- Prototype implementation using *FreePastry* overlay network (centralized and distributed)
  - FreePastry simulator (large number of peers)
  - PlanetLab deployment (validation)
- *Vertical scalability:* increase the number of resource types
- *Horizontal scalability:* increase the number of peers

Theoretical analysis

- Average allocation time
  - Size of the overlay (log)
  - Network delay
  - Request arrival rate *for each resource type*
- Vertical scalability
- Horizontal scalability

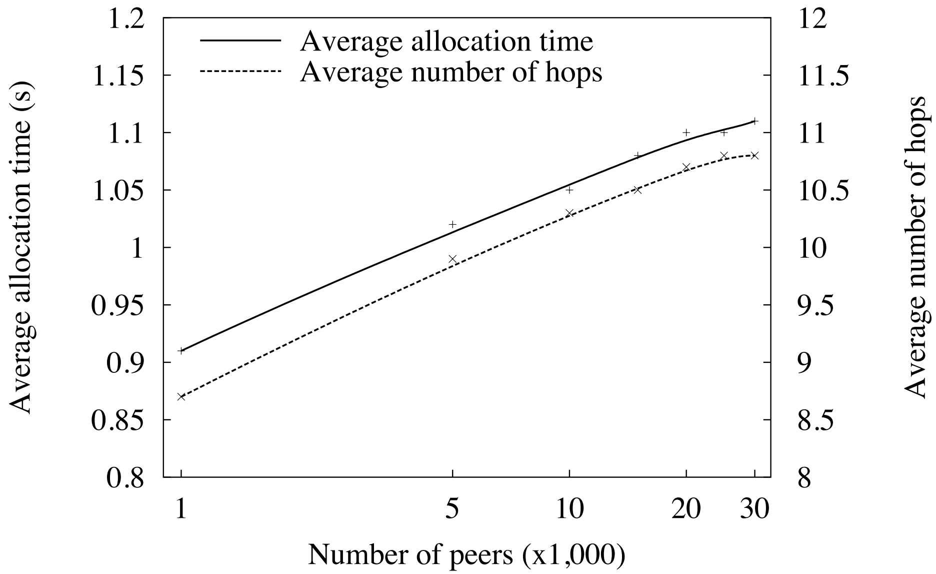
Vertical scalability

- Vary number of resource types
- PlanetLab deployment (50 peers)
- Average network delay for simulator: 100ms
- Distributed auction protocol has greater overhead
- Distributed auction protocol overhead is almost constant (average network delay)



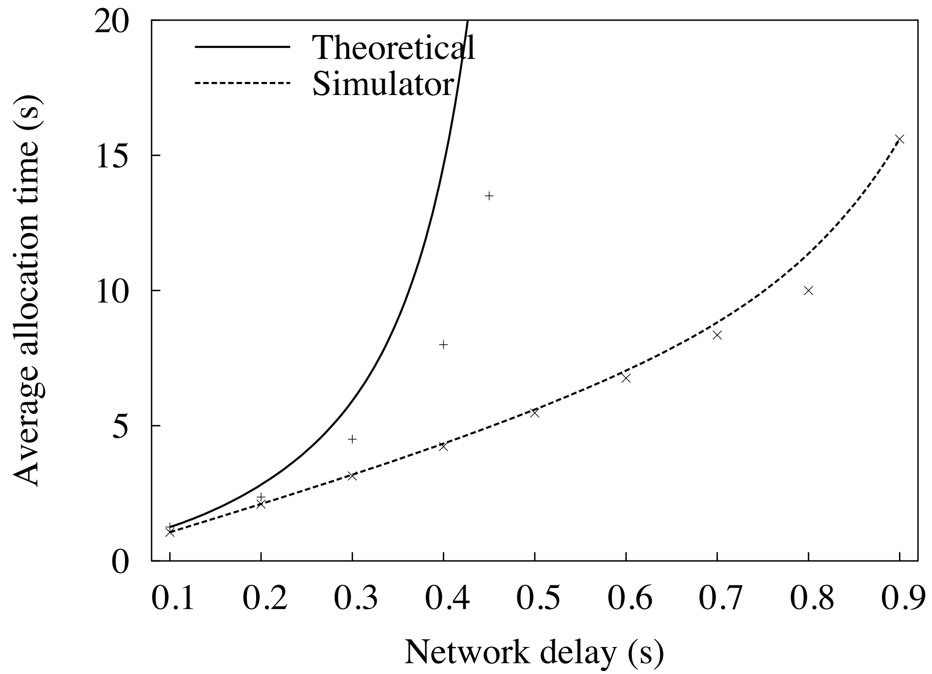
Horizontal scalability

- Vary number of peers
- FreePastry simulator
- Average allocation time increases logarithmic with the number of peers
- Comparison with the average number of hops



Impact of network delay

- Vary network delay: 100ms – 900ms
- PlanetLab average delay: ~40ms
- Large network delay (>300ms) results in increased allocation times



Summary

- Vertical Scalability
  - Parallelize payments for different resource types
  - Preserve economic properties: payment computation requires complete information about resources in the system
  - Distribute resource information according to resource type
- Horizontal Scalability
  - Use a distributed hash table (DHT) to maintain resource information and build a scalable resource location service
  - Distribute resource information between different hosts according to resource type

Conclusions and remarks

- Main contribution: *a distributed auction framework that scales with the number of users and the number of resource types*
- Vertical scalability still remains a problem when having a small number of resource types
- Further increase scalability: relax the strategy-proof condition, which eliminates the synchronization mechanism
  - Network delay and request arrival rate will not influence the allocation time

Thank you!

Q & A

Contact: marianmi@comp.nus.edu.sg