Wide Area Placement of Data Replicas for Fast and Highly Available Data Access

Fan Ping

Xiaohu Li, Christopher McConnell
Rohini Vabbalareddy, Jeong-Hyon Hwang

State University of New York - Albany

Outline

- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

Outline

- **Background**
- **Network Coordinate System**
- **Data Replication**
  - **Data Replication for Performance**
  - **Data Replication for Performance and Availability**
- **Conclusion**

Data Intensive Distributed Systems

- Google, Amazon, Facebook, Microsoft…

Data Intensive Distributed Systems

- Google, Amazon, Facebook, Microsoft…

- Dynamo, Cassandra, PNUTS…

Data Replica Placement

- **Given a replication degree (e.g., 3), where should we put those data replicas in order to effectively improve the overall data access speed and availability?**

- **Challenges**
  - **Scalability**
  - **Certain SLA**

Outline

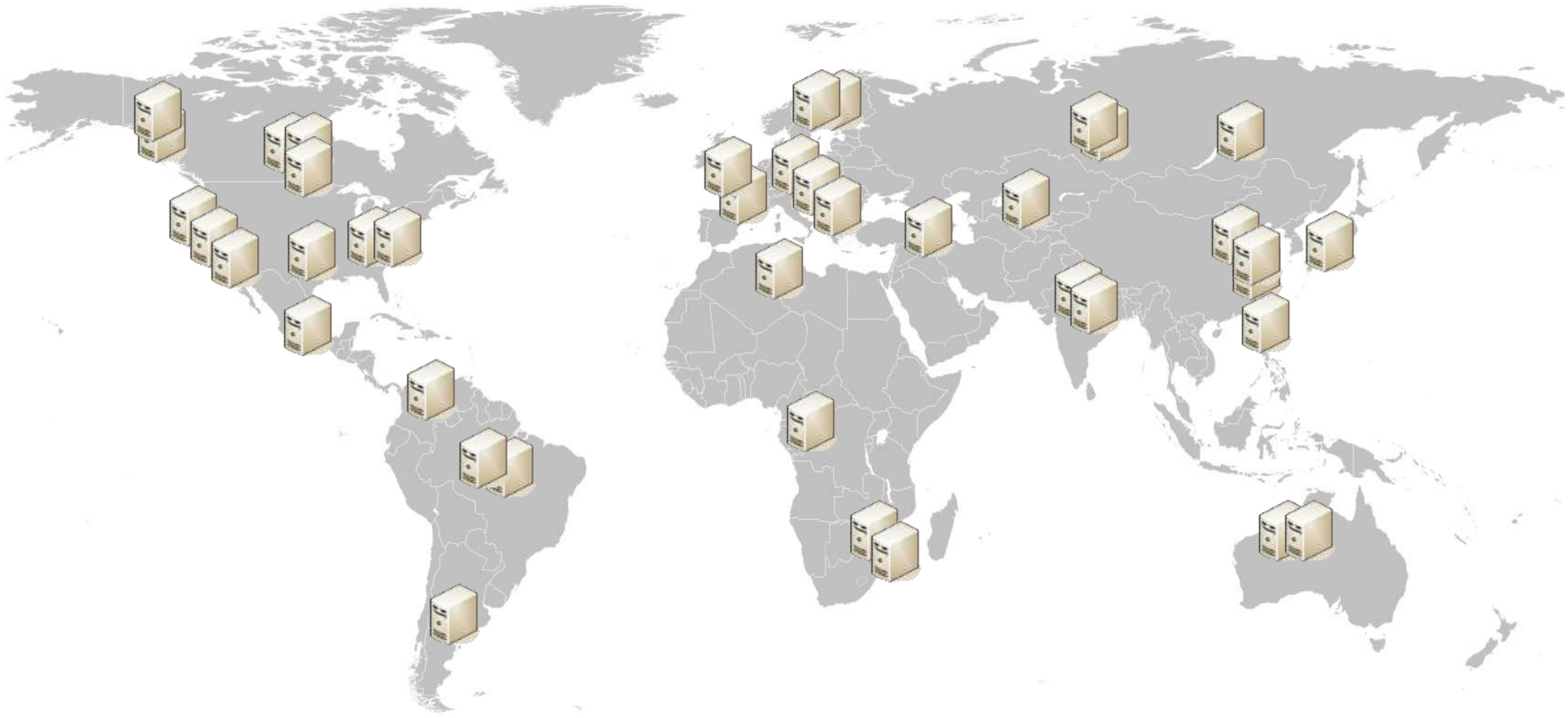
- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

Network Coordinate Systems

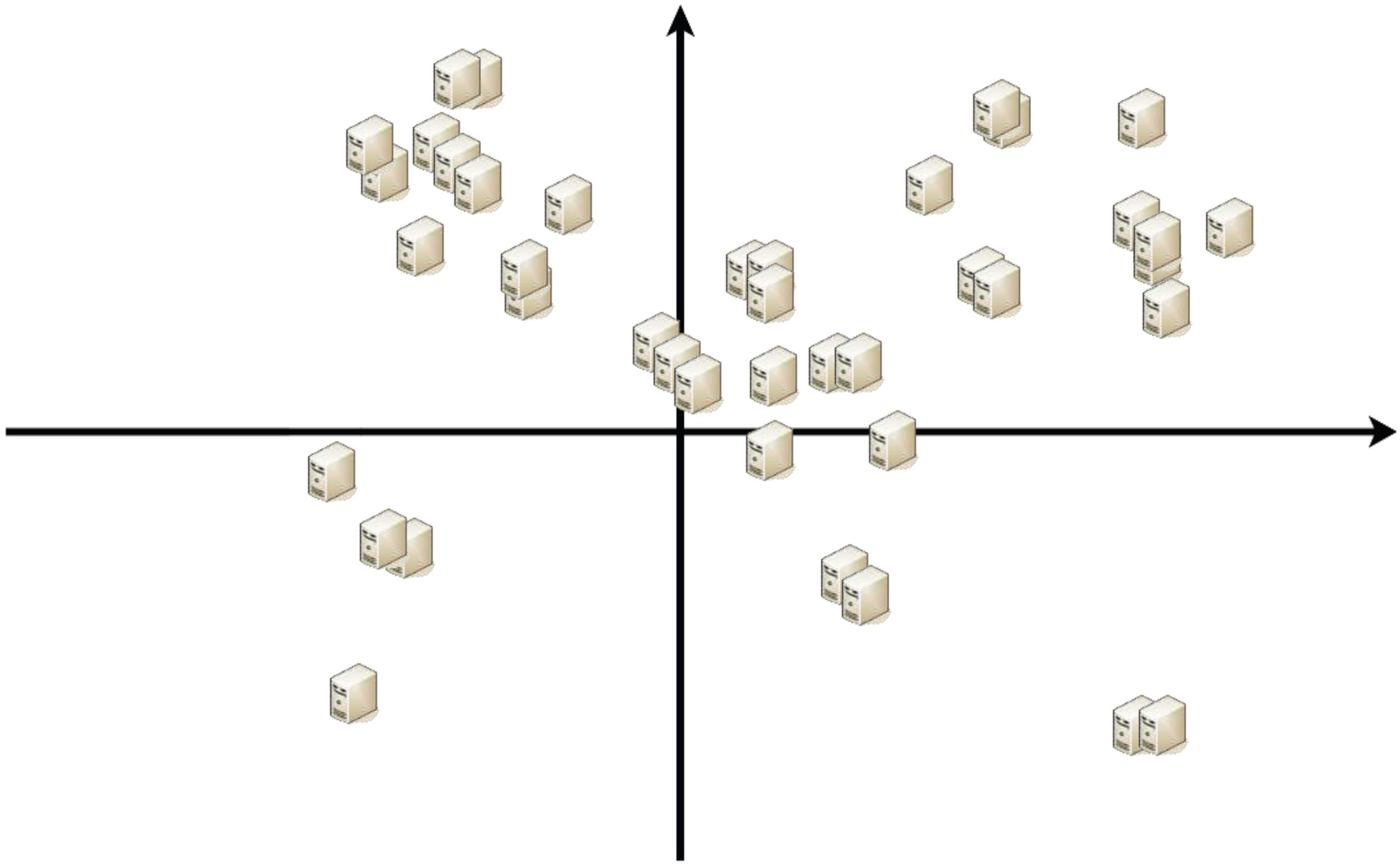
- **Based on the network latencies between each other, nodes are embedded into a virtual space so that their distances in this virtual space are close to the network latencies.**

- **E.g., Vivaldi, RNP**

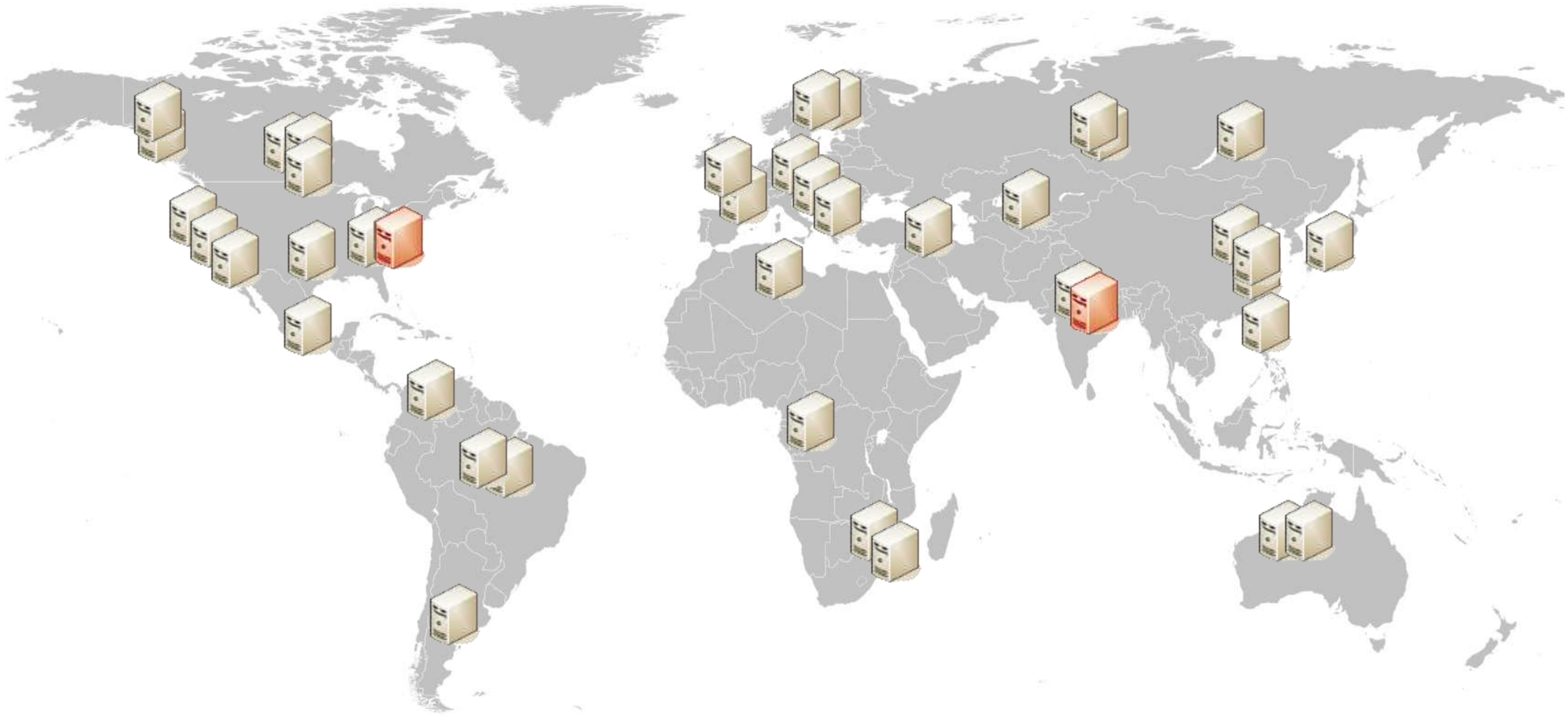
Network Coordinate Systems



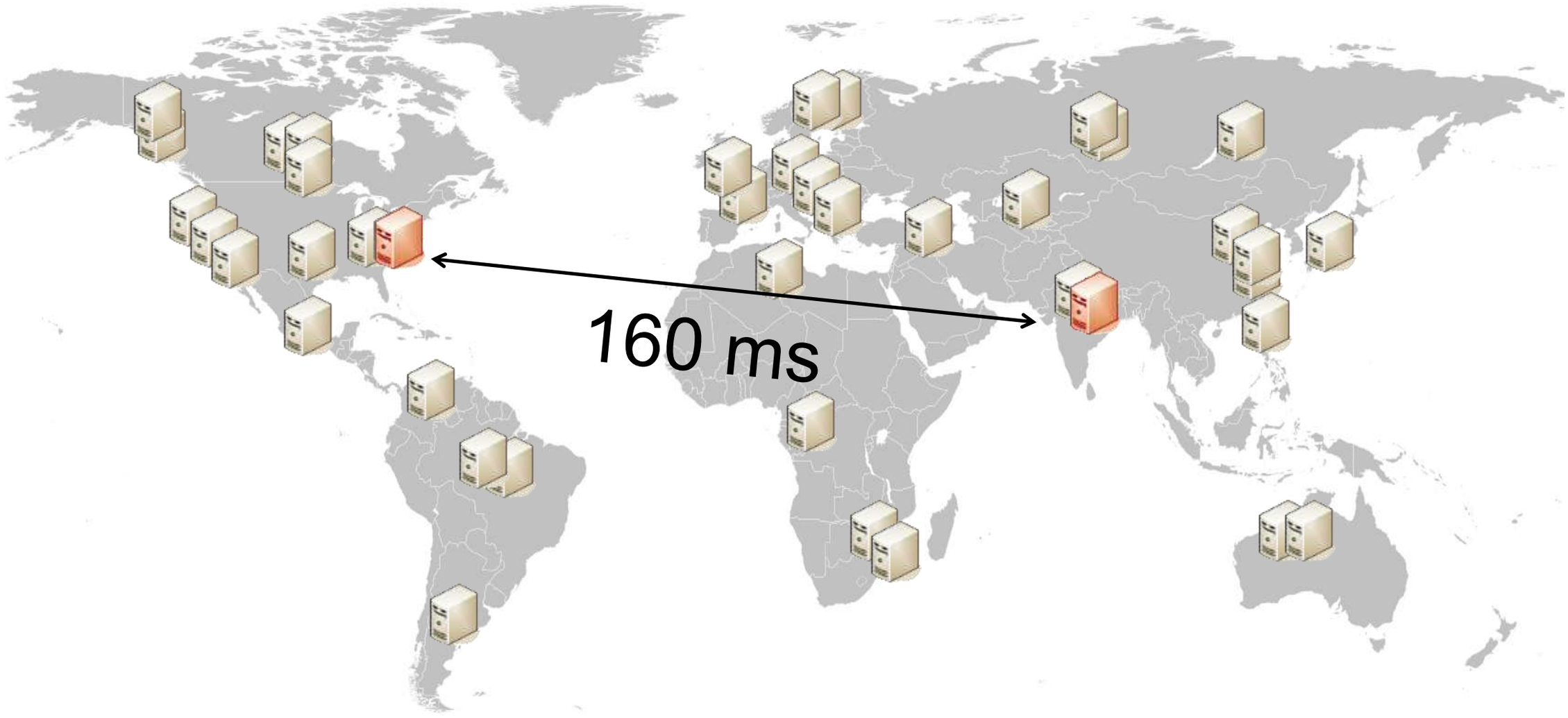
Network Coordinate Systems



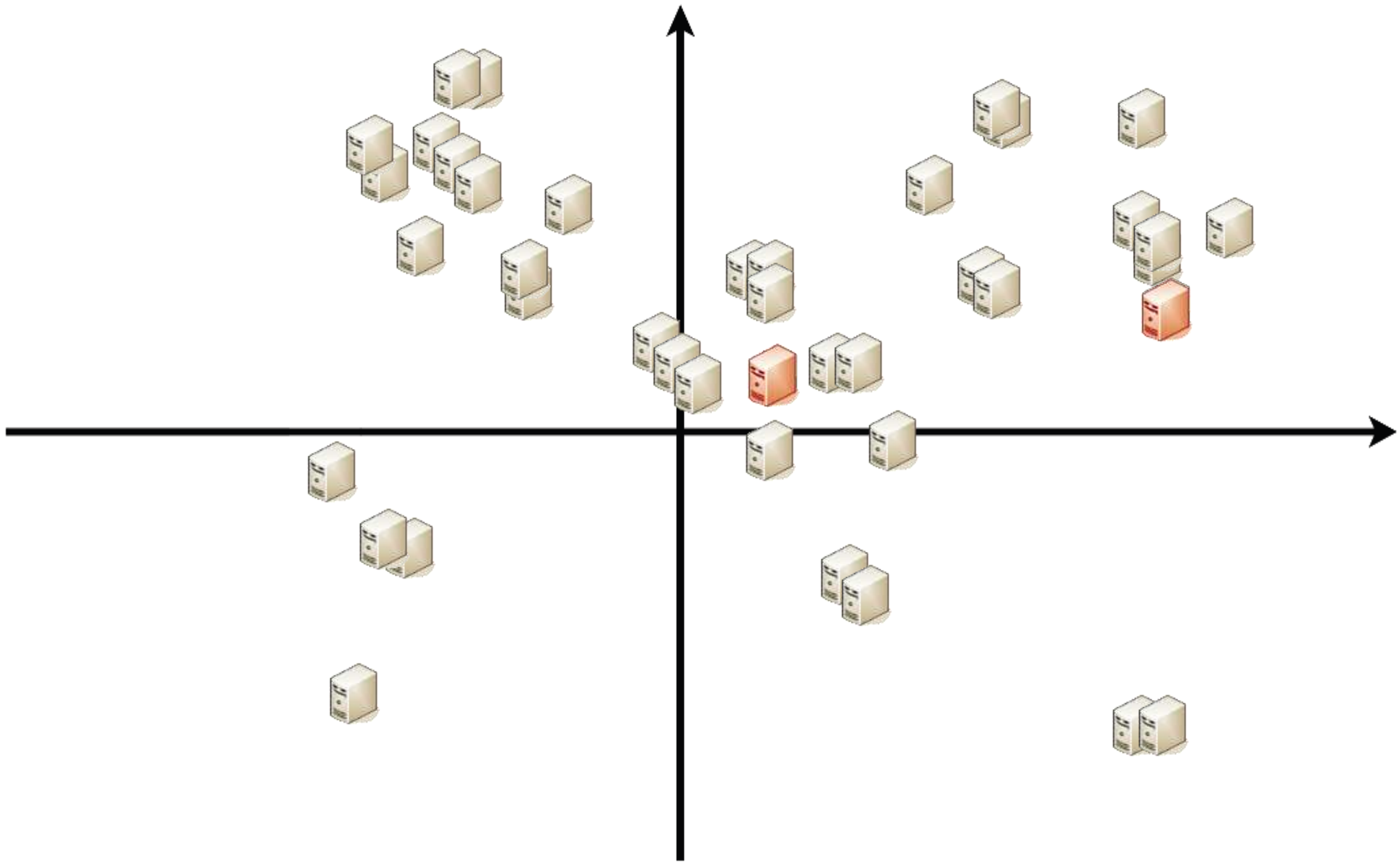
Network Coordinate Systems



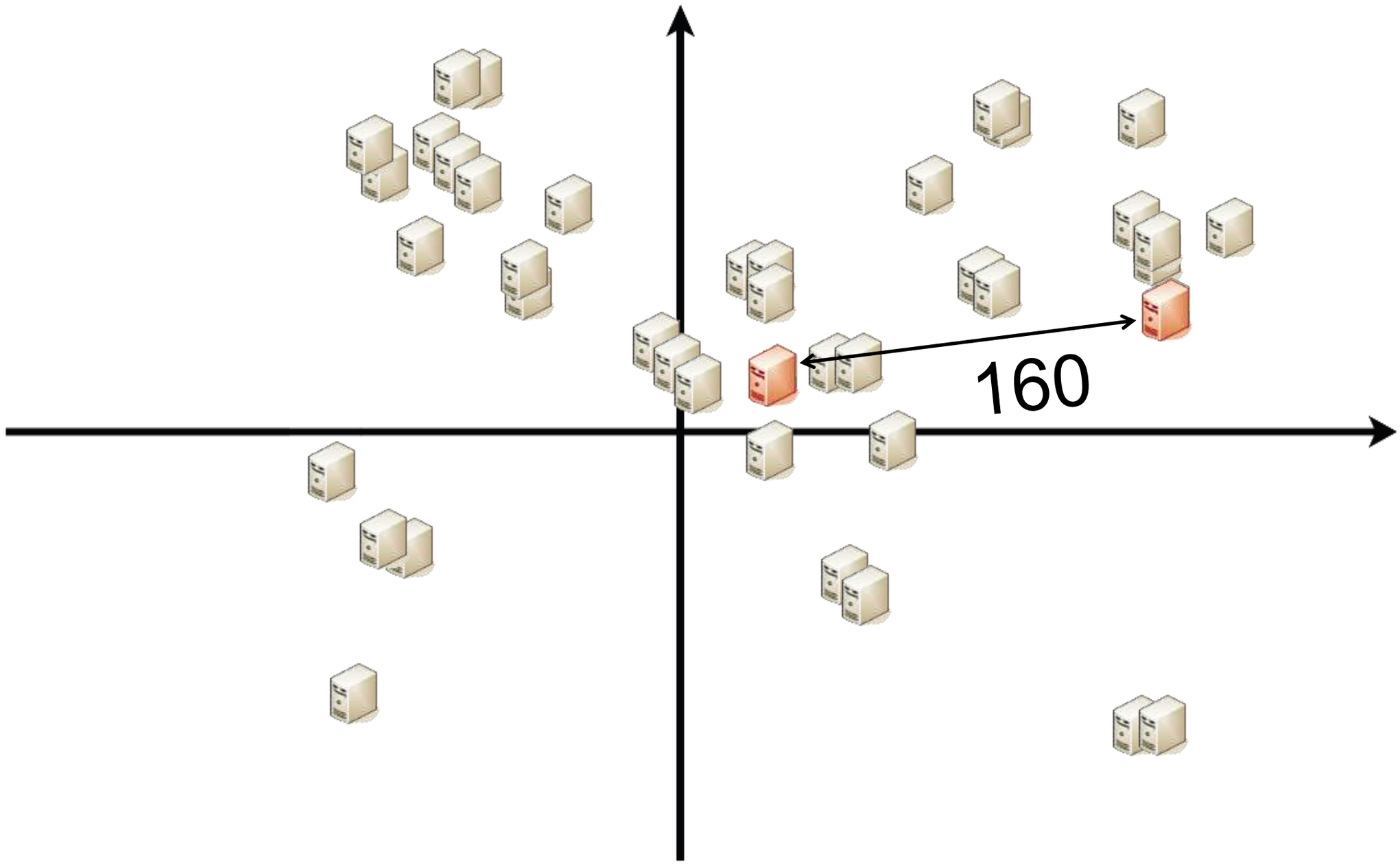
Network Coordinate Systems



Network Coordinate Systems



Network Coordinate Systems



Network Coordinate Systems

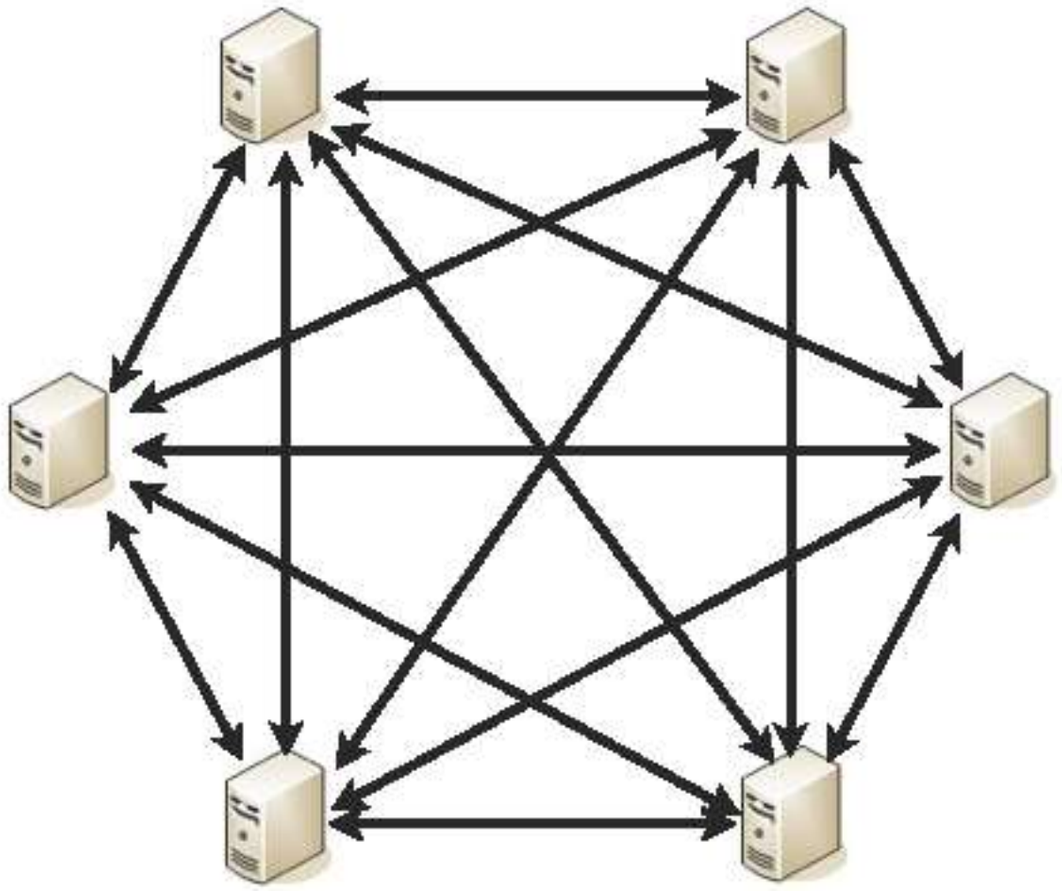
- Network Latencies vs Network Coordinates





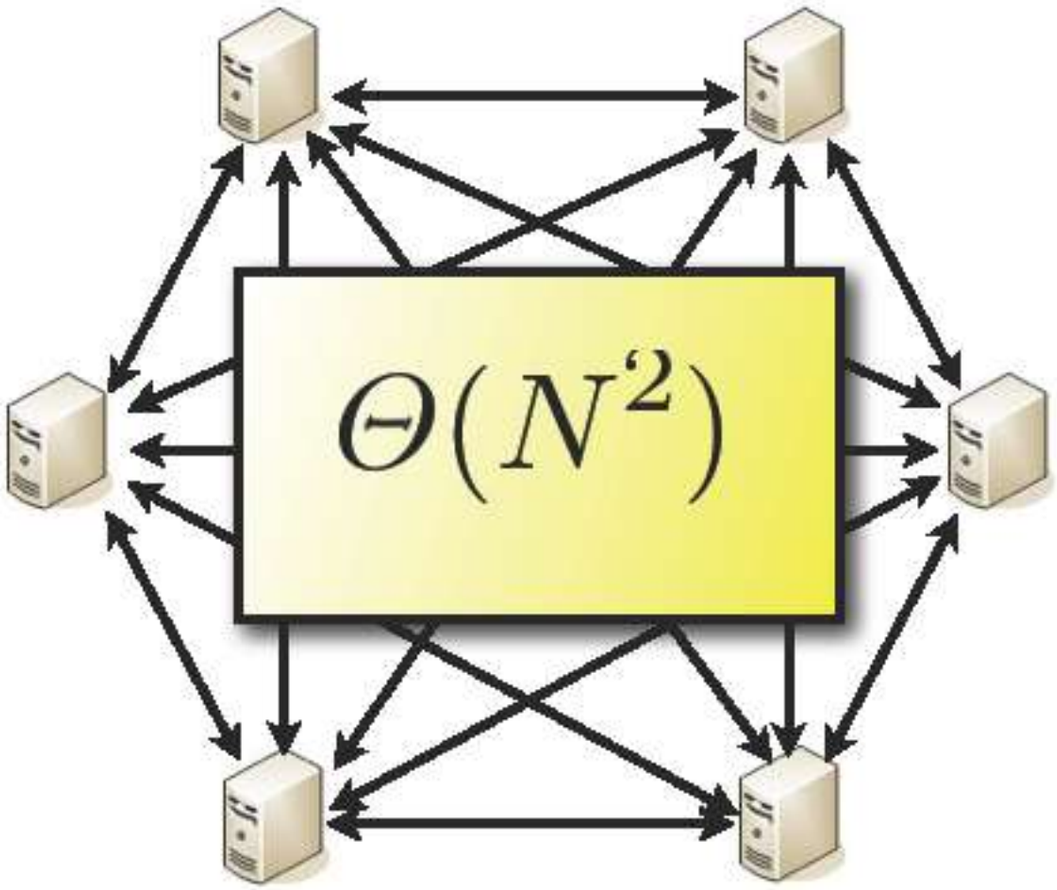

Network Coordinate Systems

- Network Latencies vs Network Coordinates



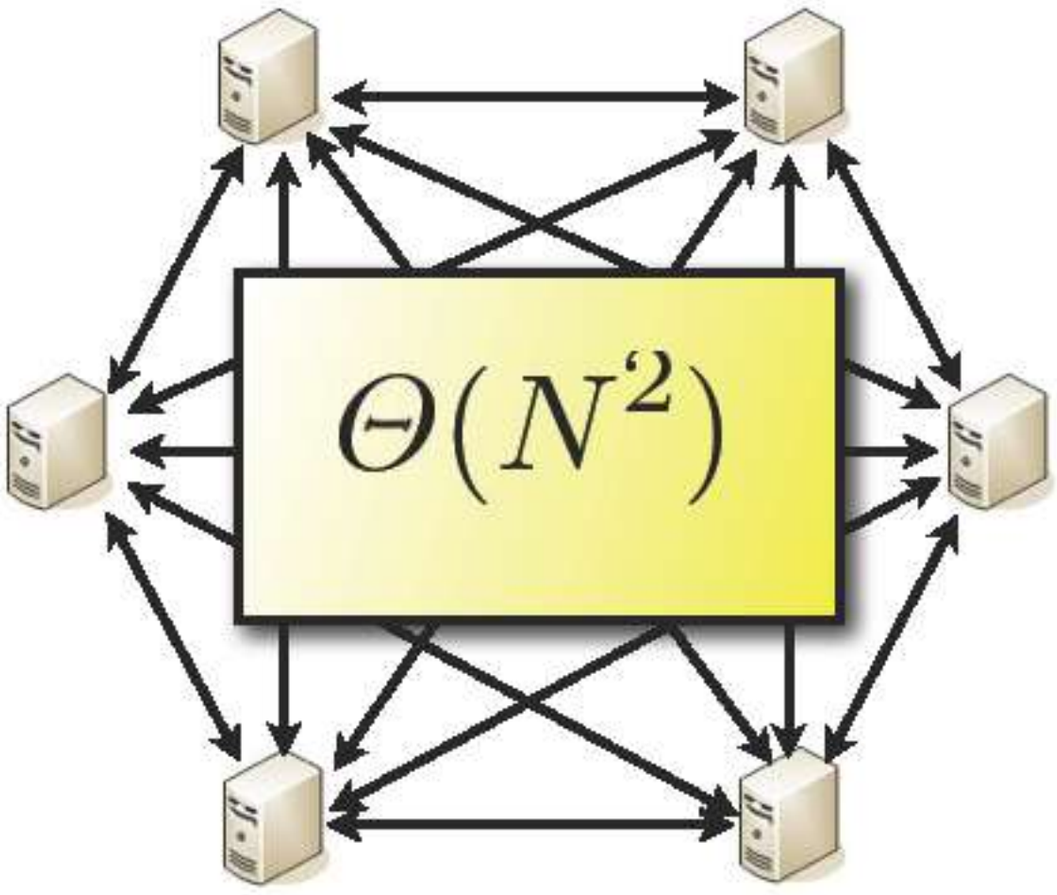
Network Coordinate Systems

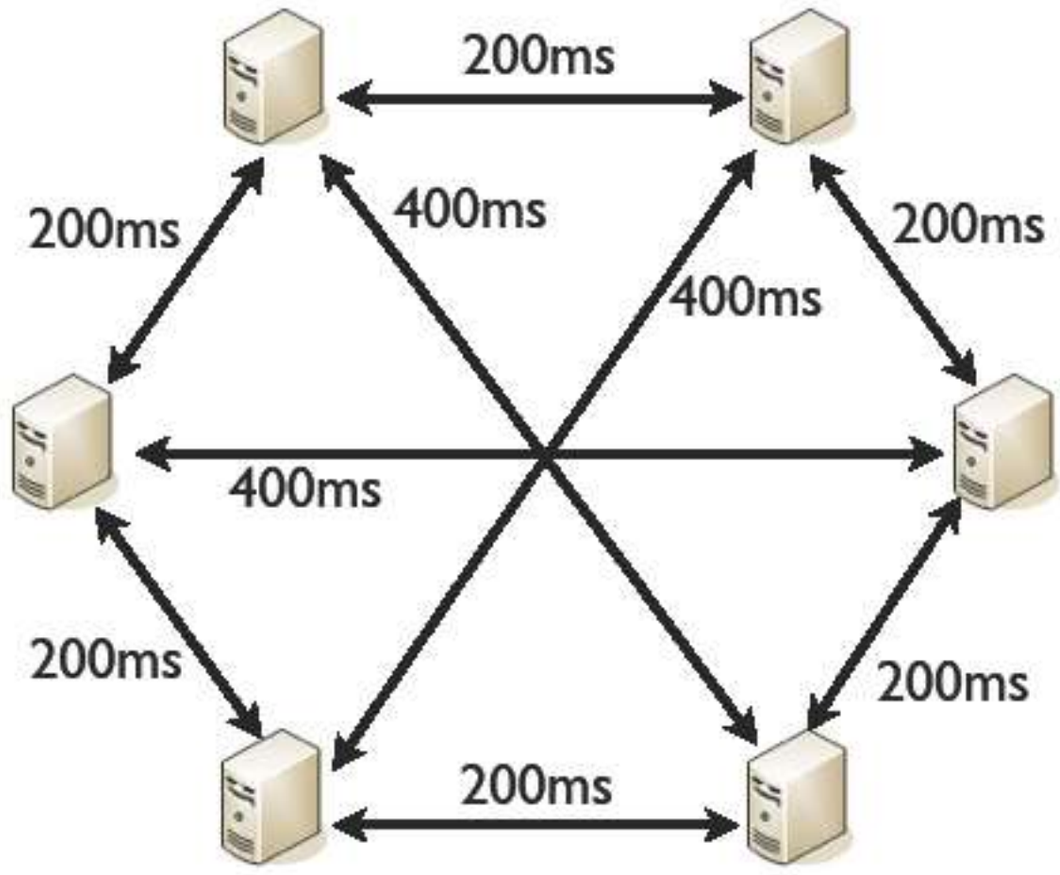
- Network Latencies vs Network Coordinates



Network Coordinate Systems

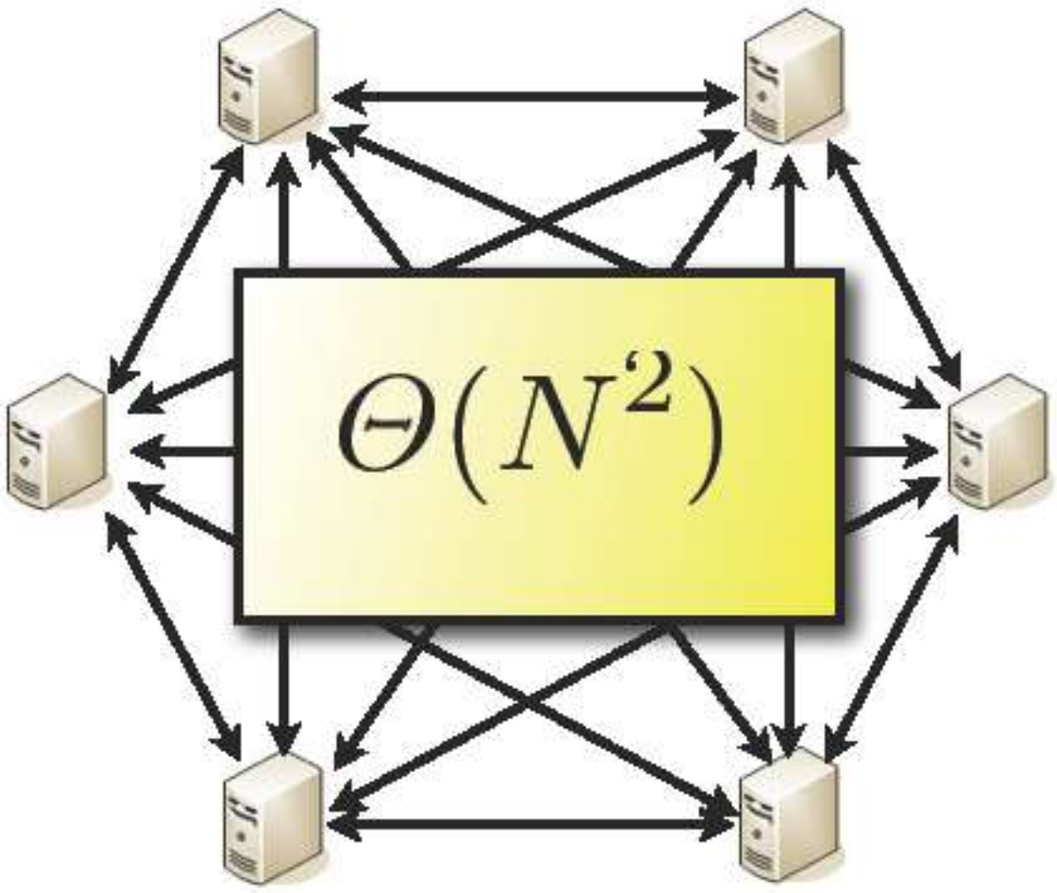
- Network Latencies vs Network Coordinates

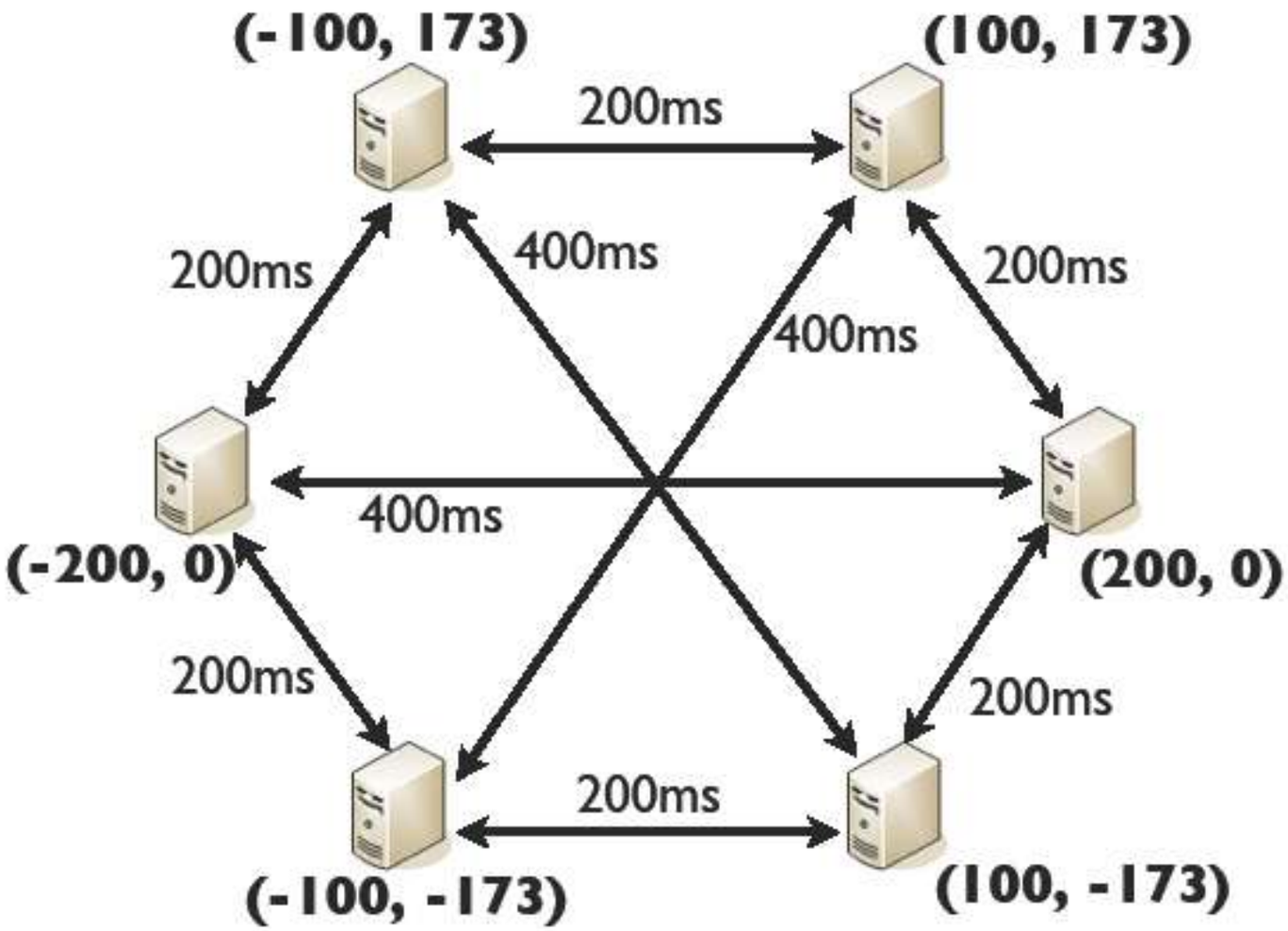




Network Coordinate Systems

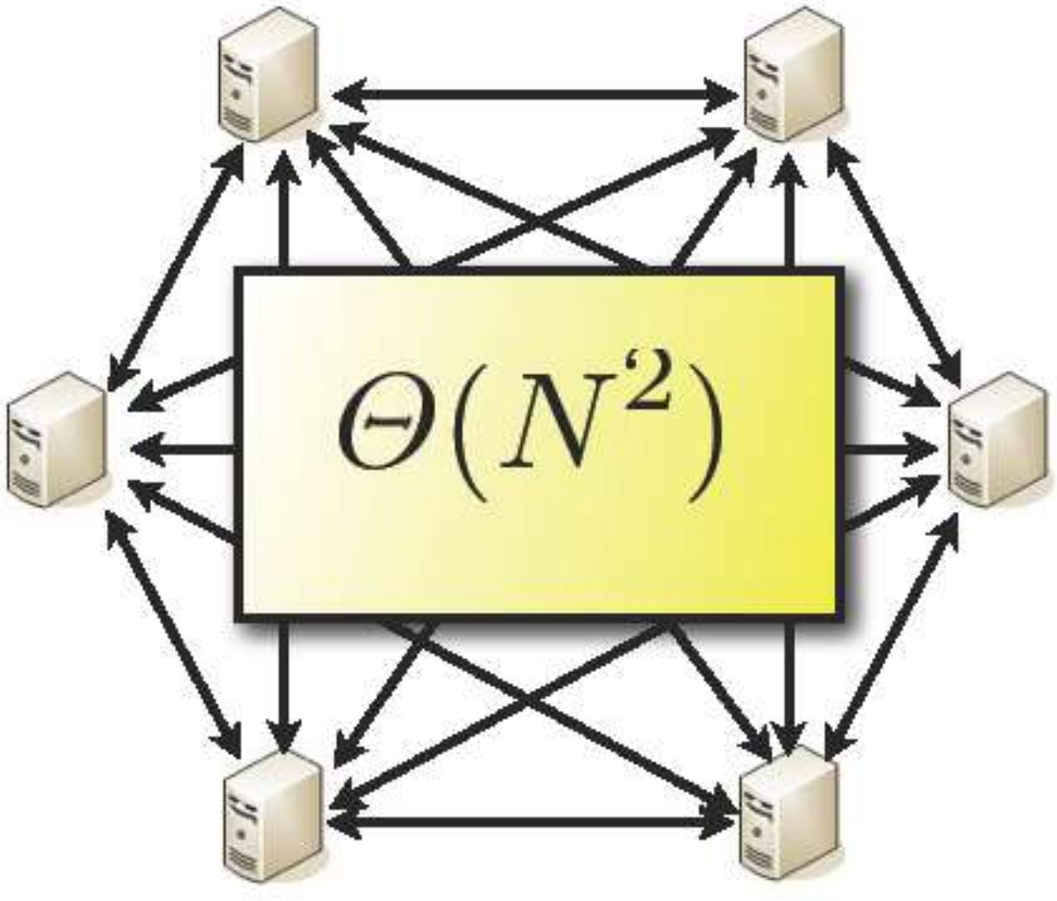
- Network Latencies vs Network Coordinates

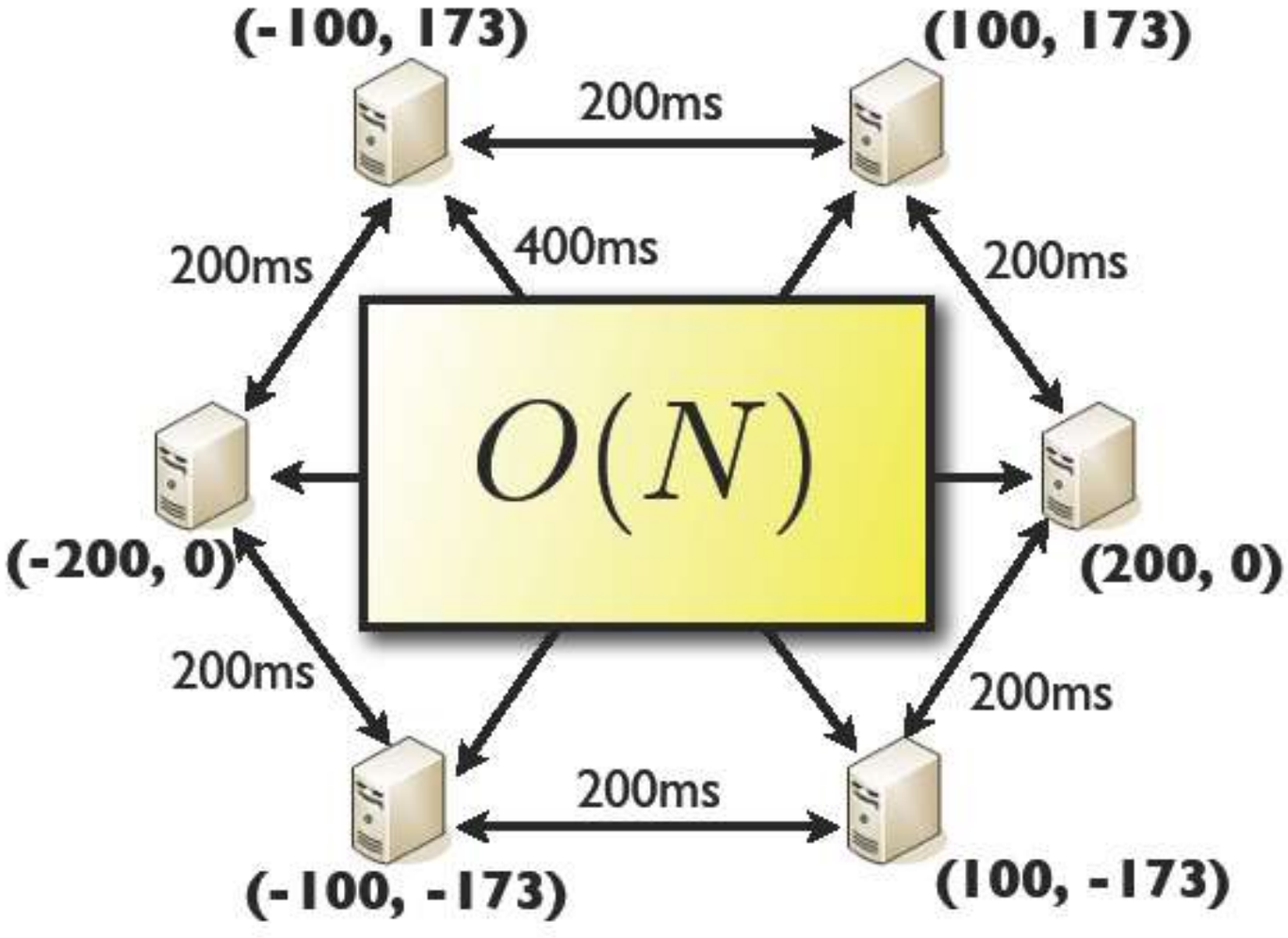




Network Coordinate Systems

- Network Latencies vs Network Coordinates





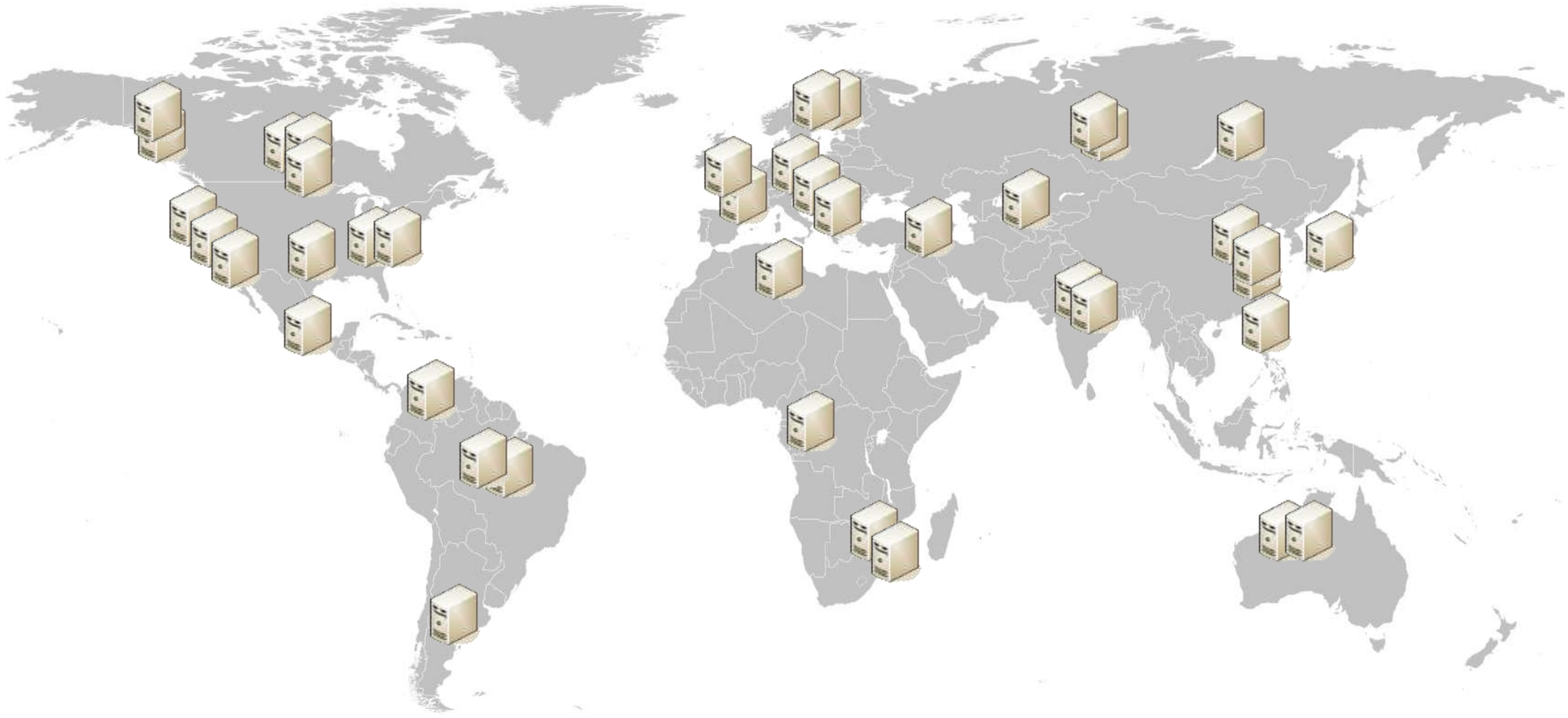
Outline

- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

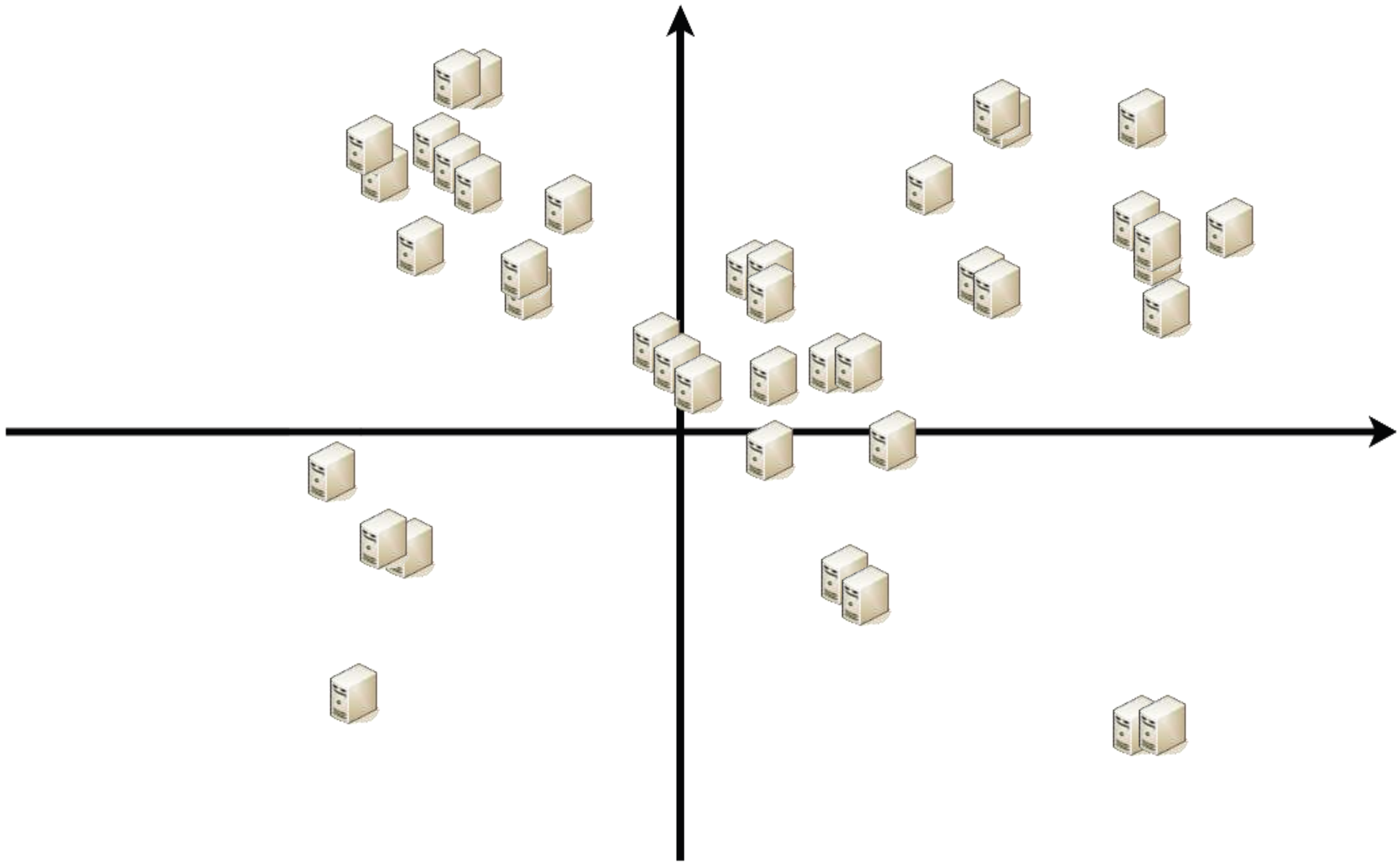
Outline

- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

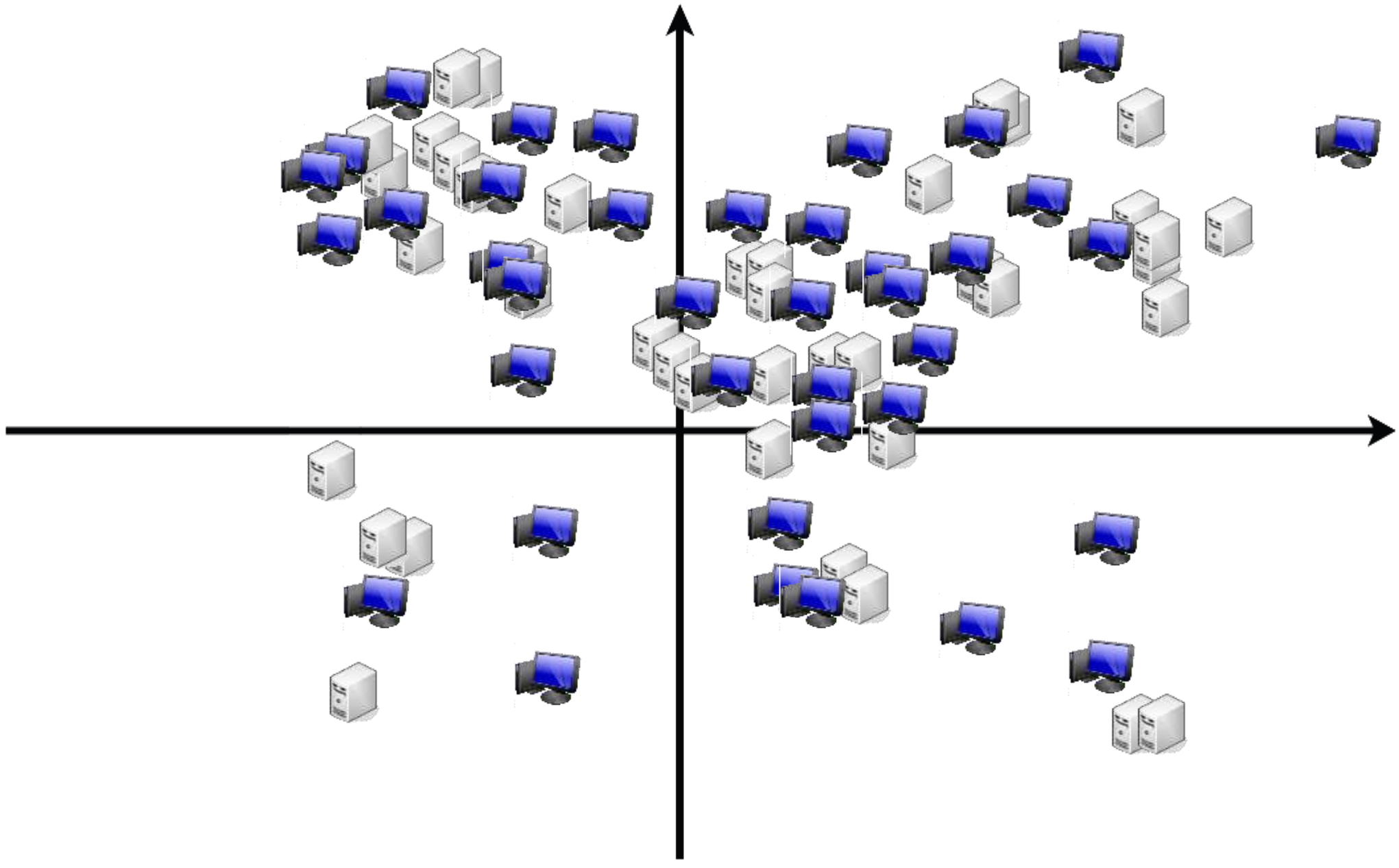
Servers on the map



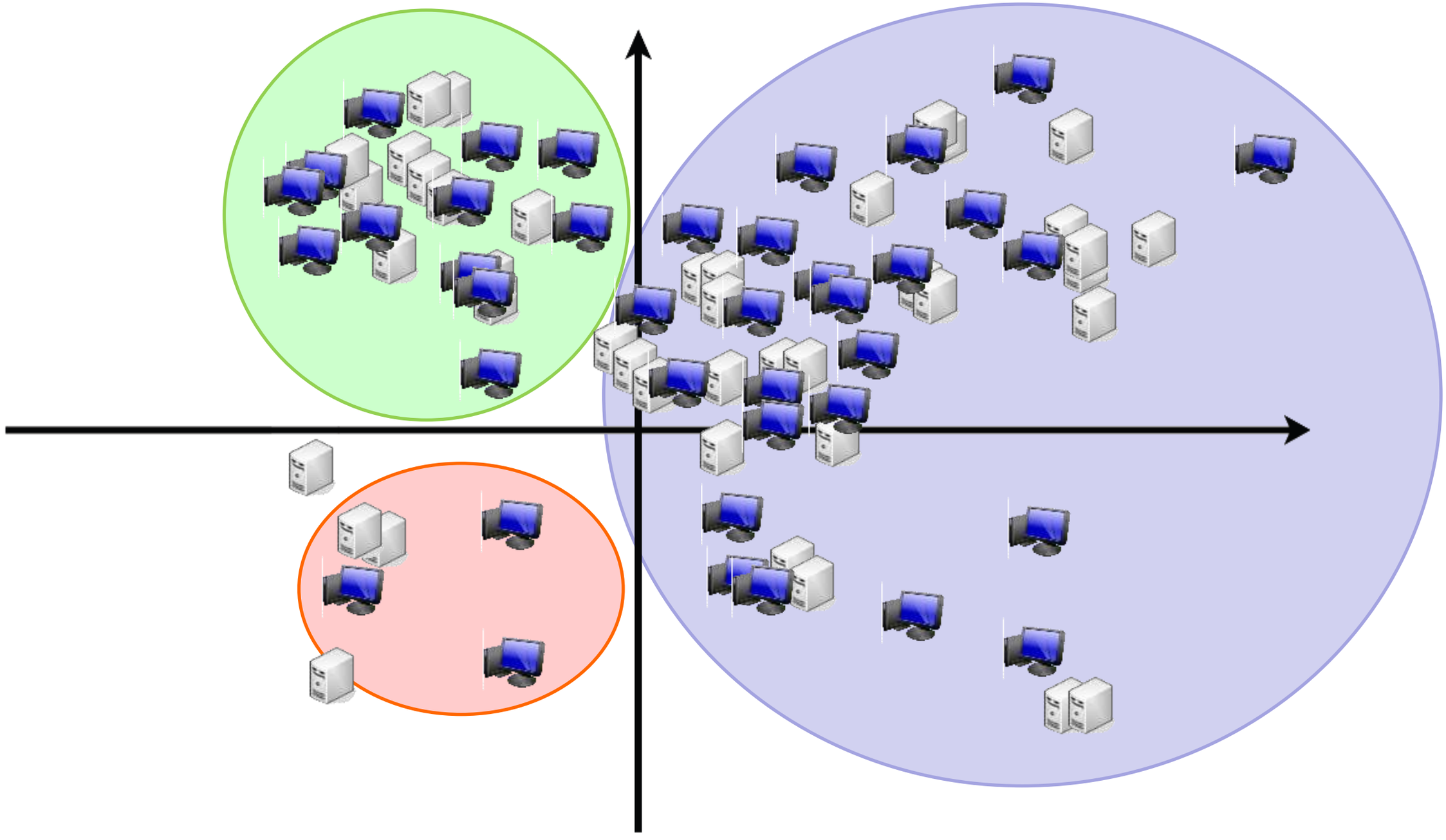
Servers in the coordinate system



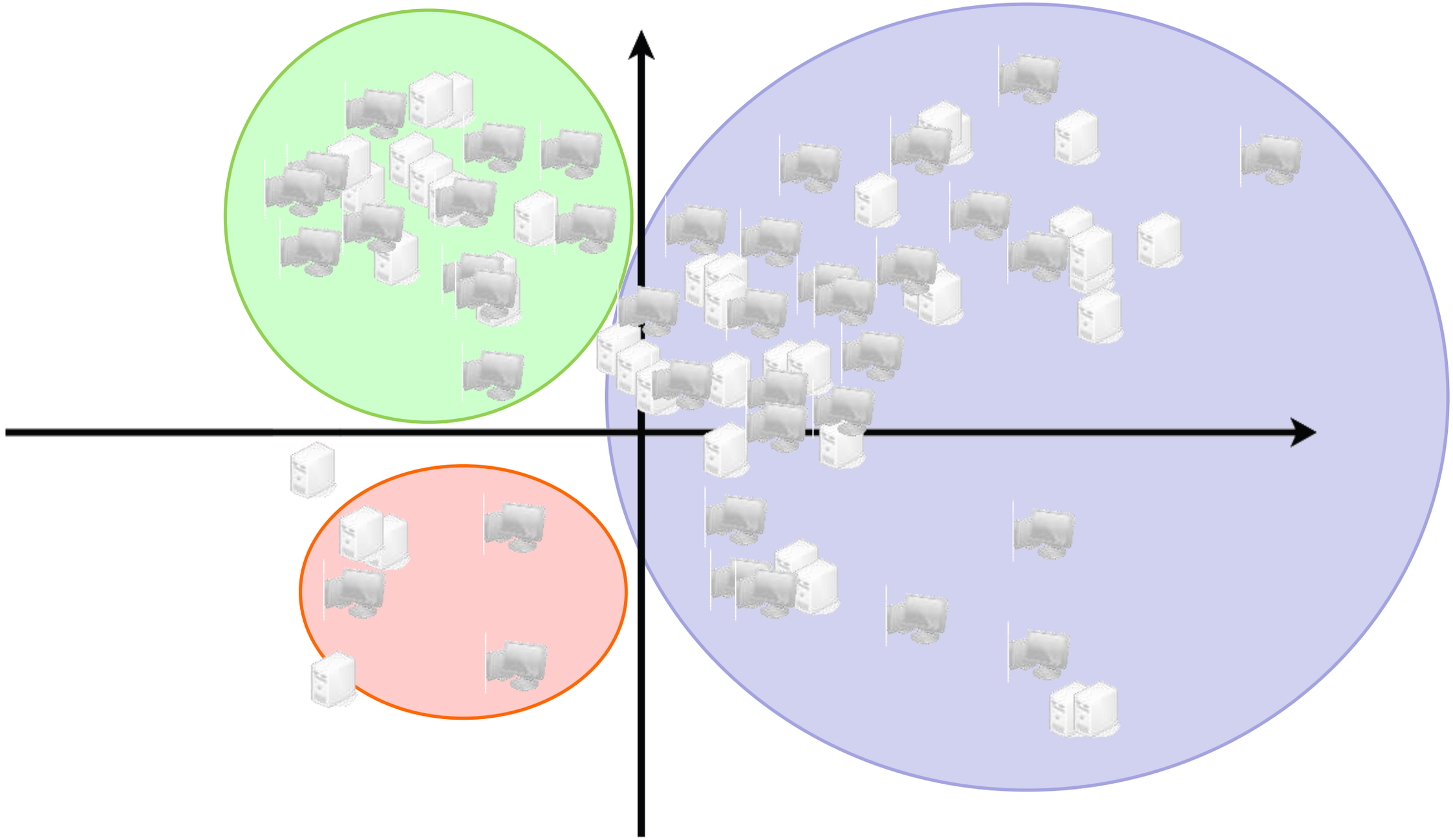
Clients in the coordinate system



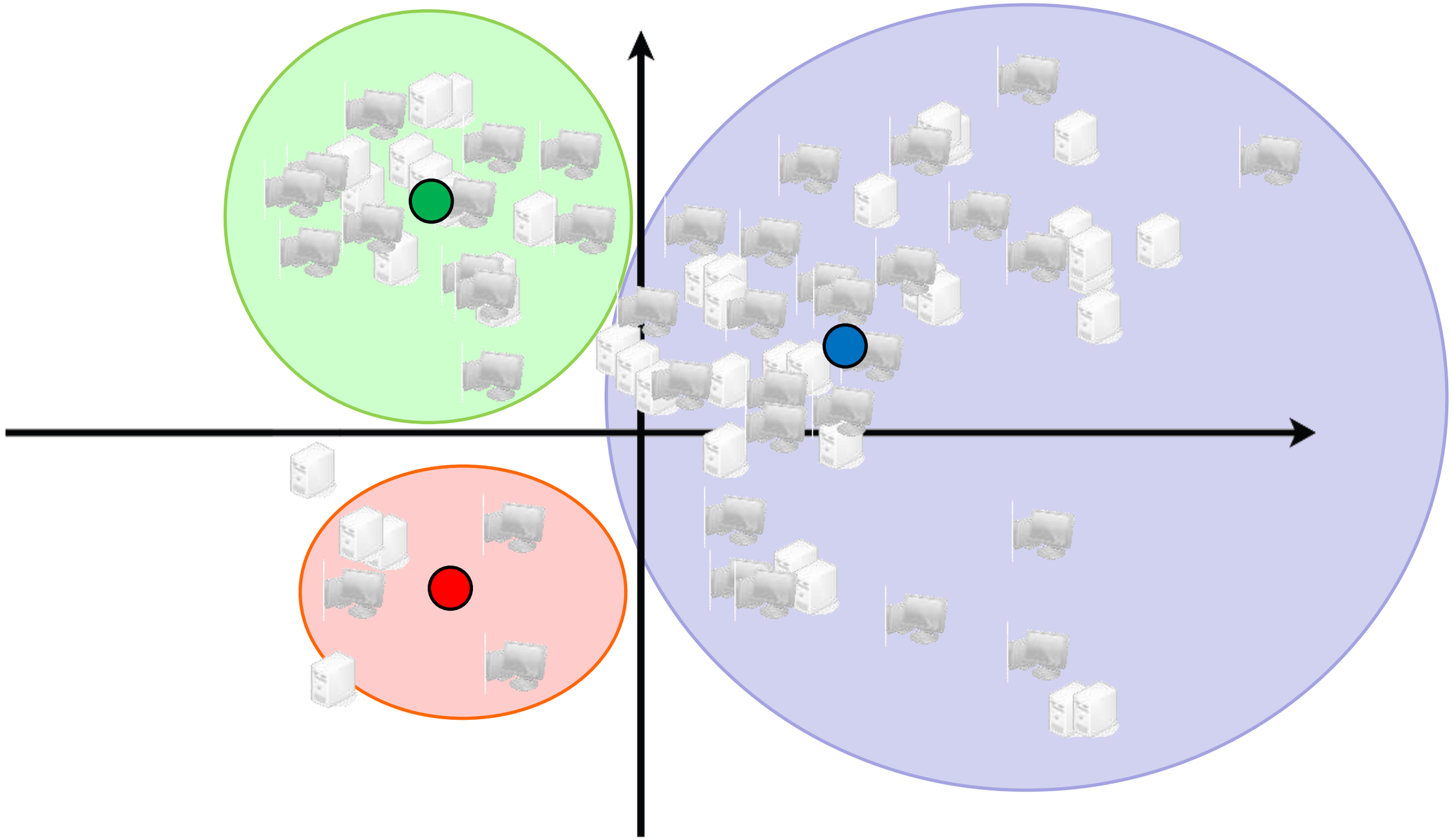
Cluster the clients in the coordinate system



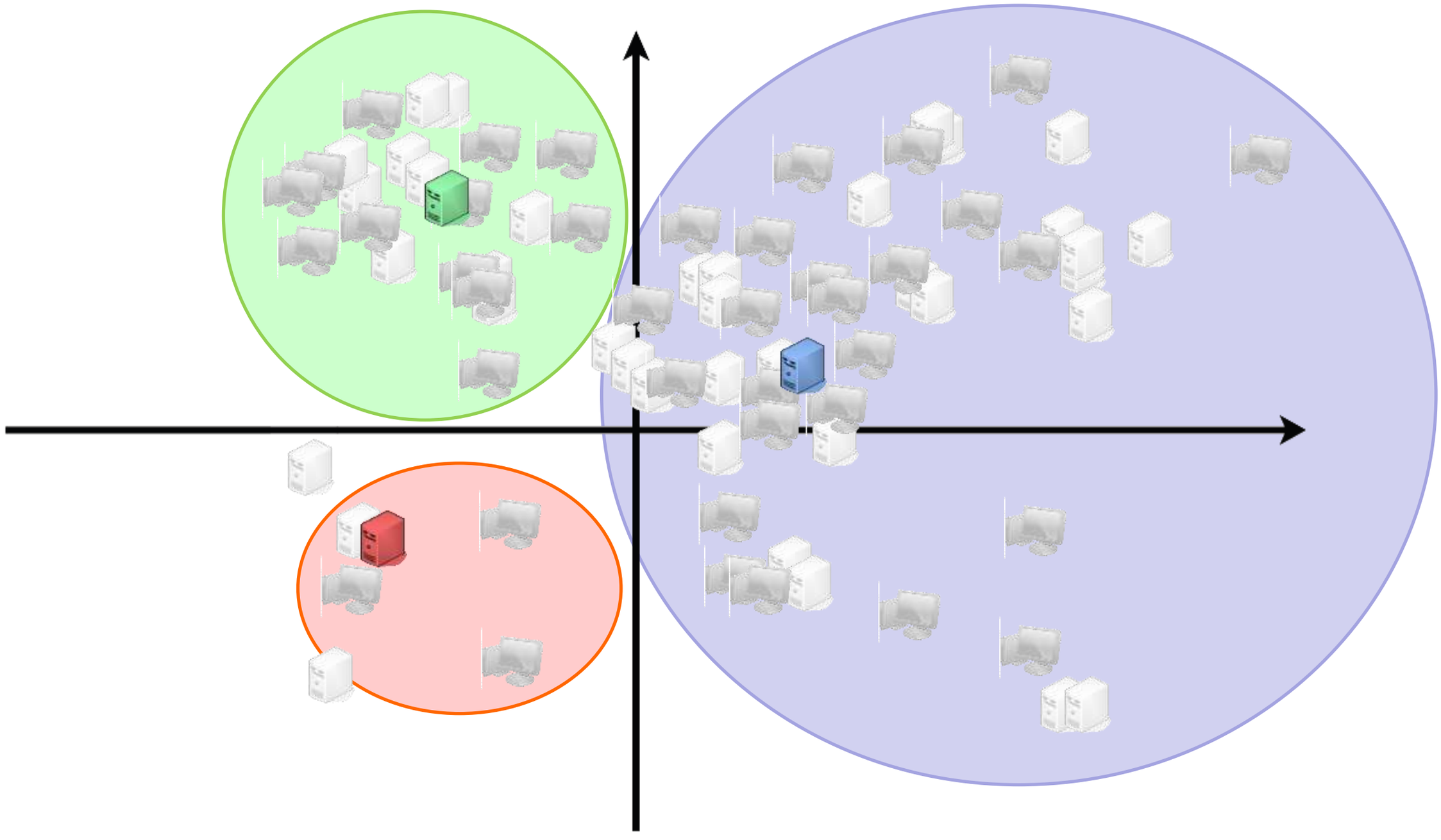
Cluster the clients in the coordinate system



Centroids of the clusters



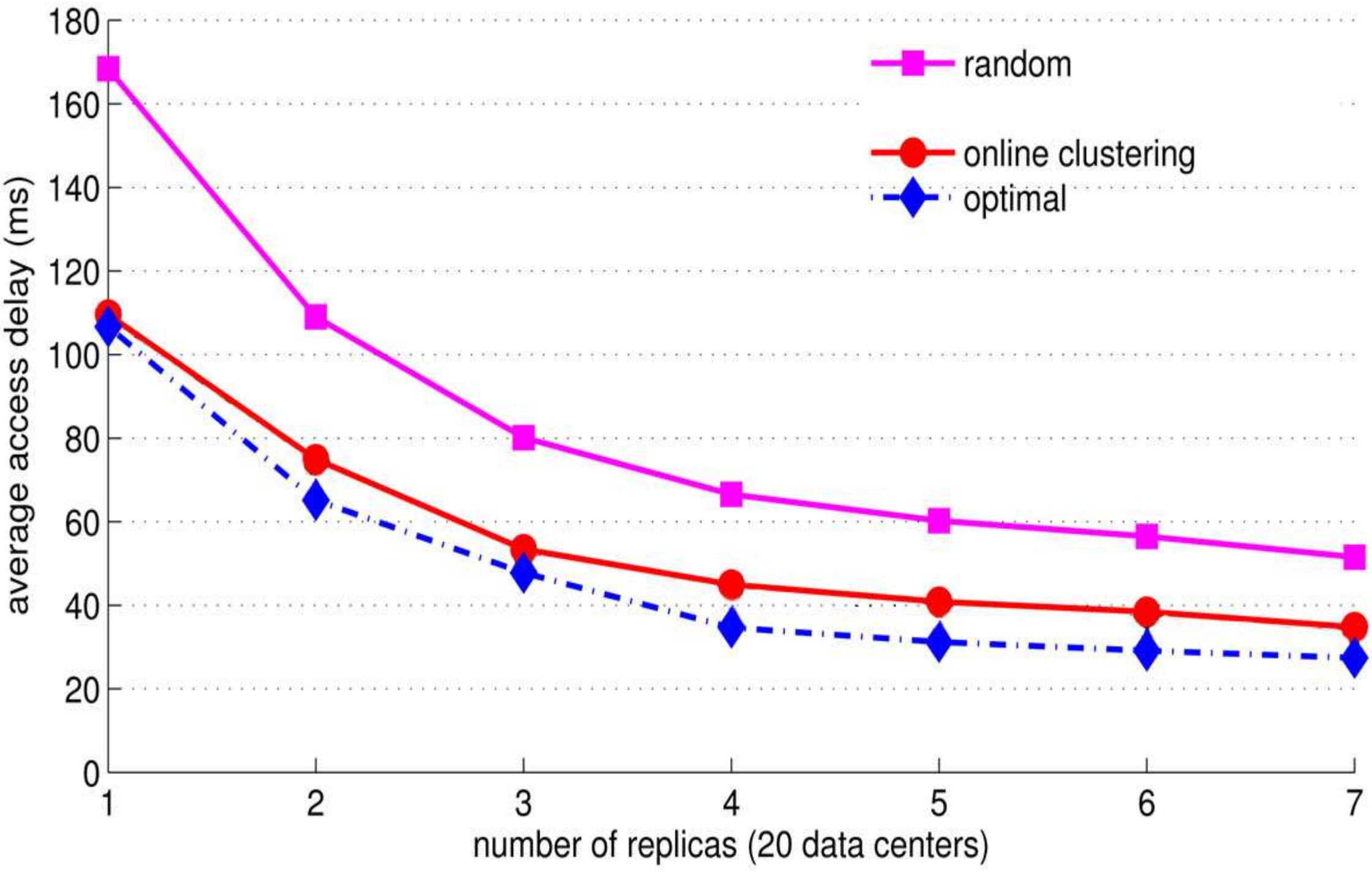
Servers near centroids of the clusters



Simulation Settings

- Java simulator
- ~200 Planetlab-node trace as input
- A certain number of nodes are selected as servers
- The other nodes are used as clients

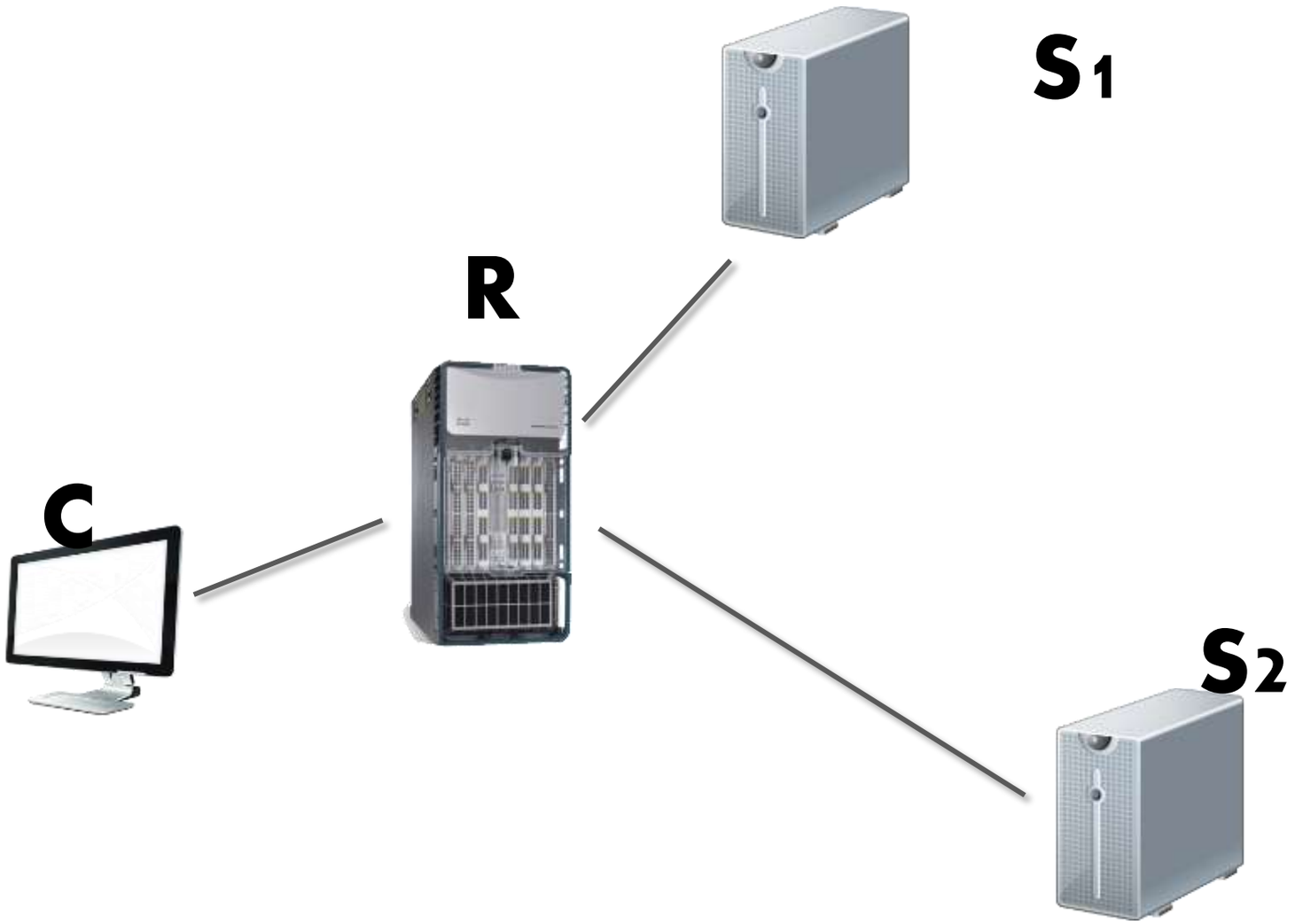
Performance VS. Number of Replicas



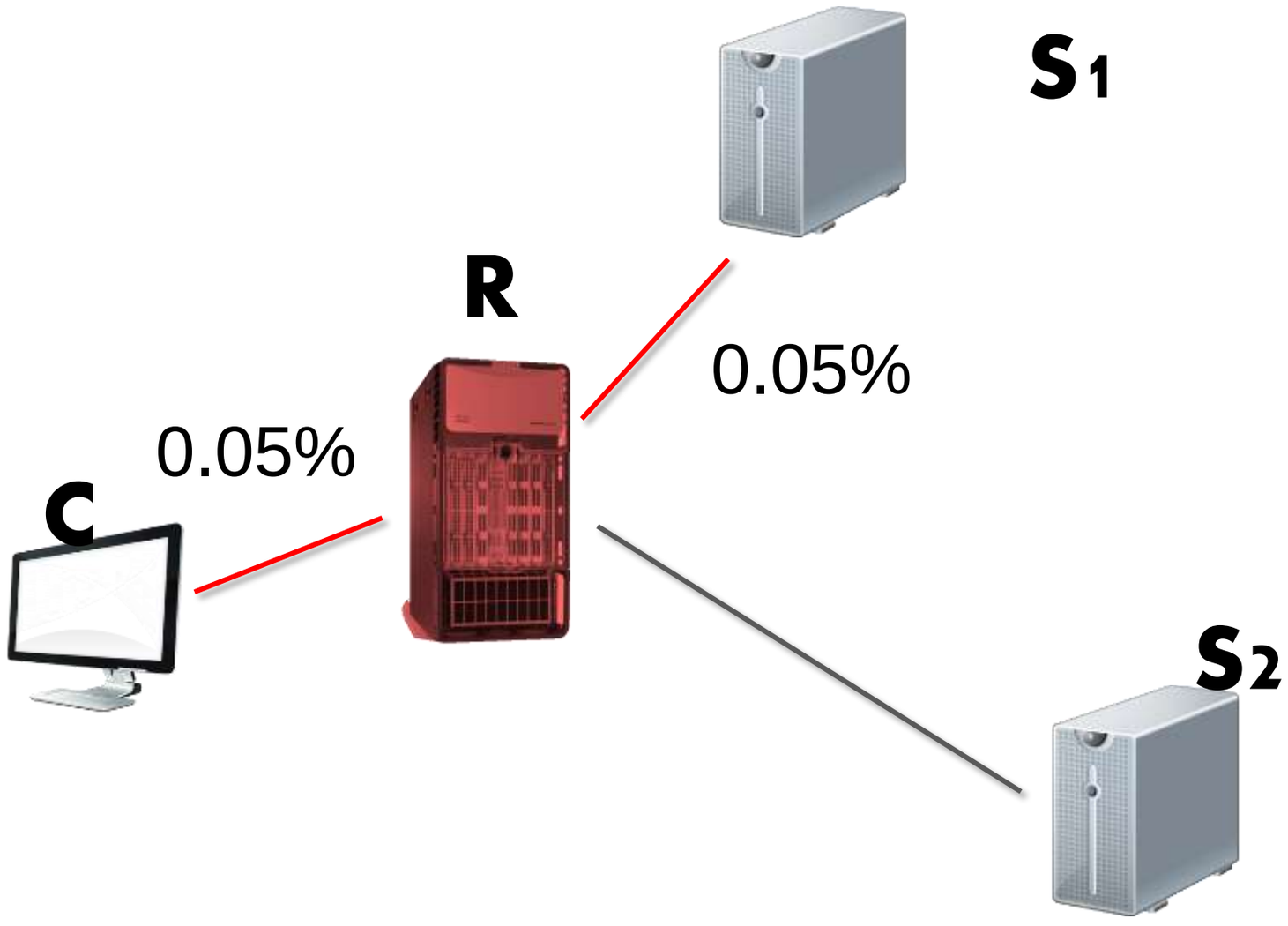
Outline

- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

Conditional Failure vs. Angle

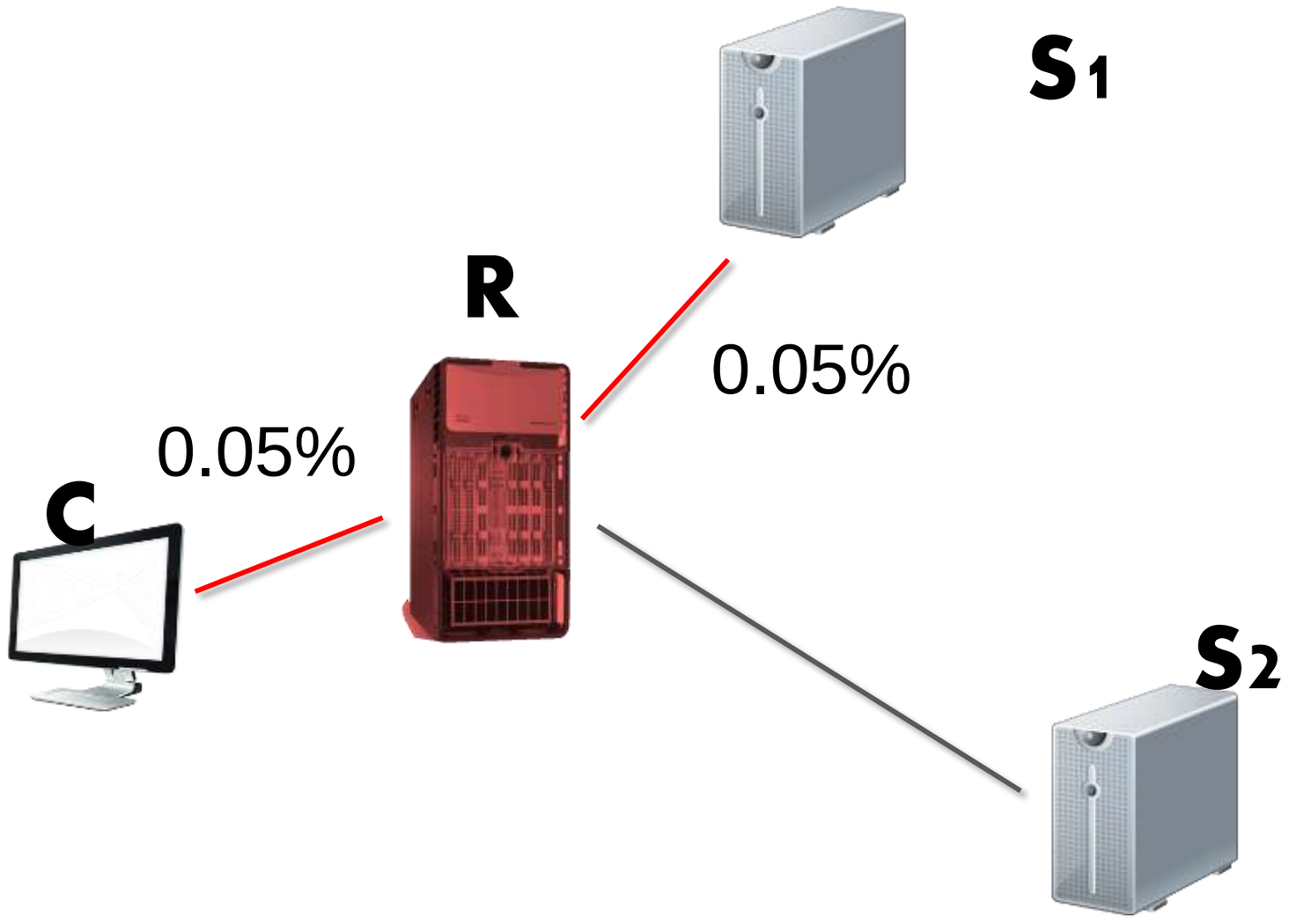


Conditional Failure vs. Angle

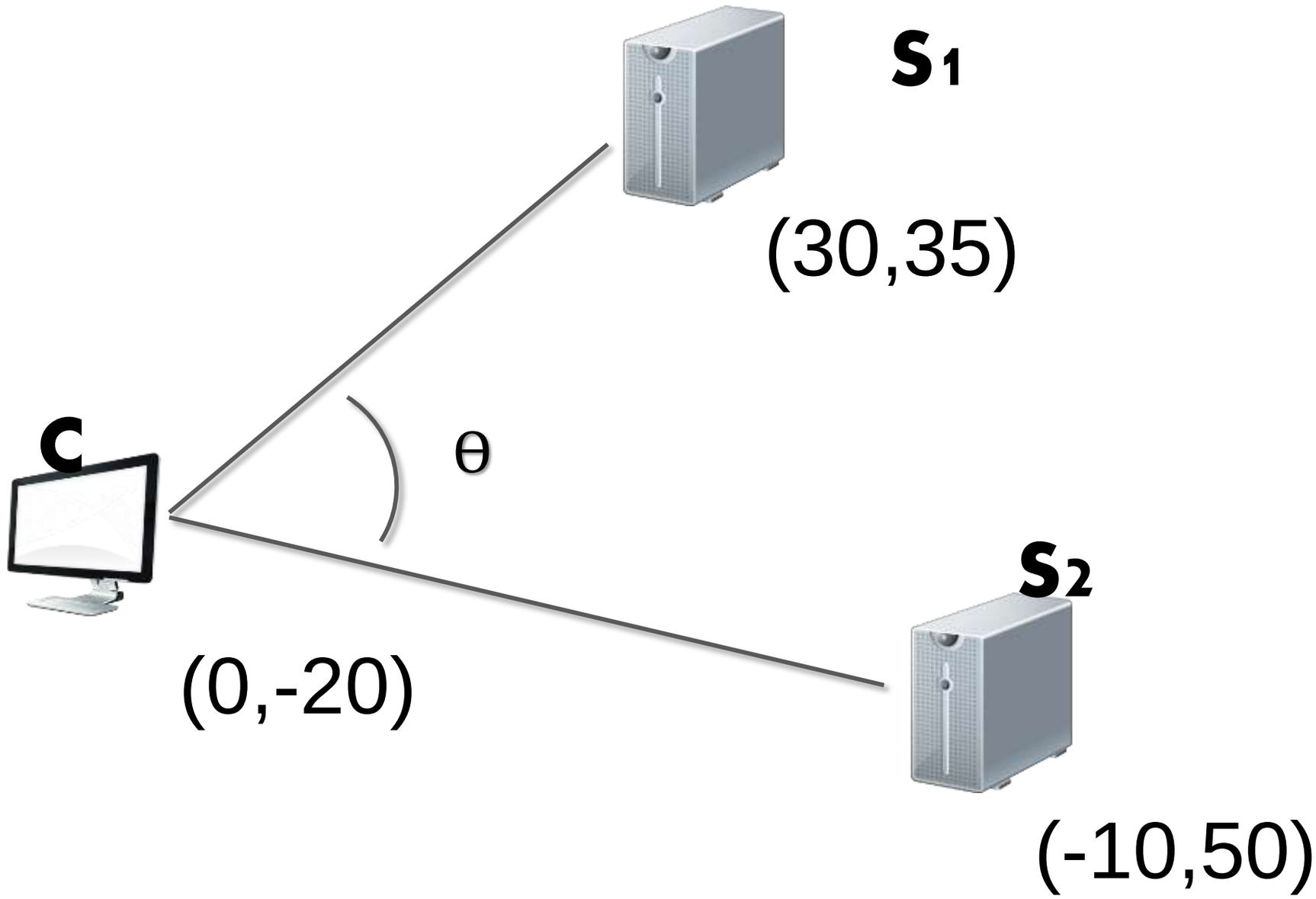


Conditional Failure vs. Angle

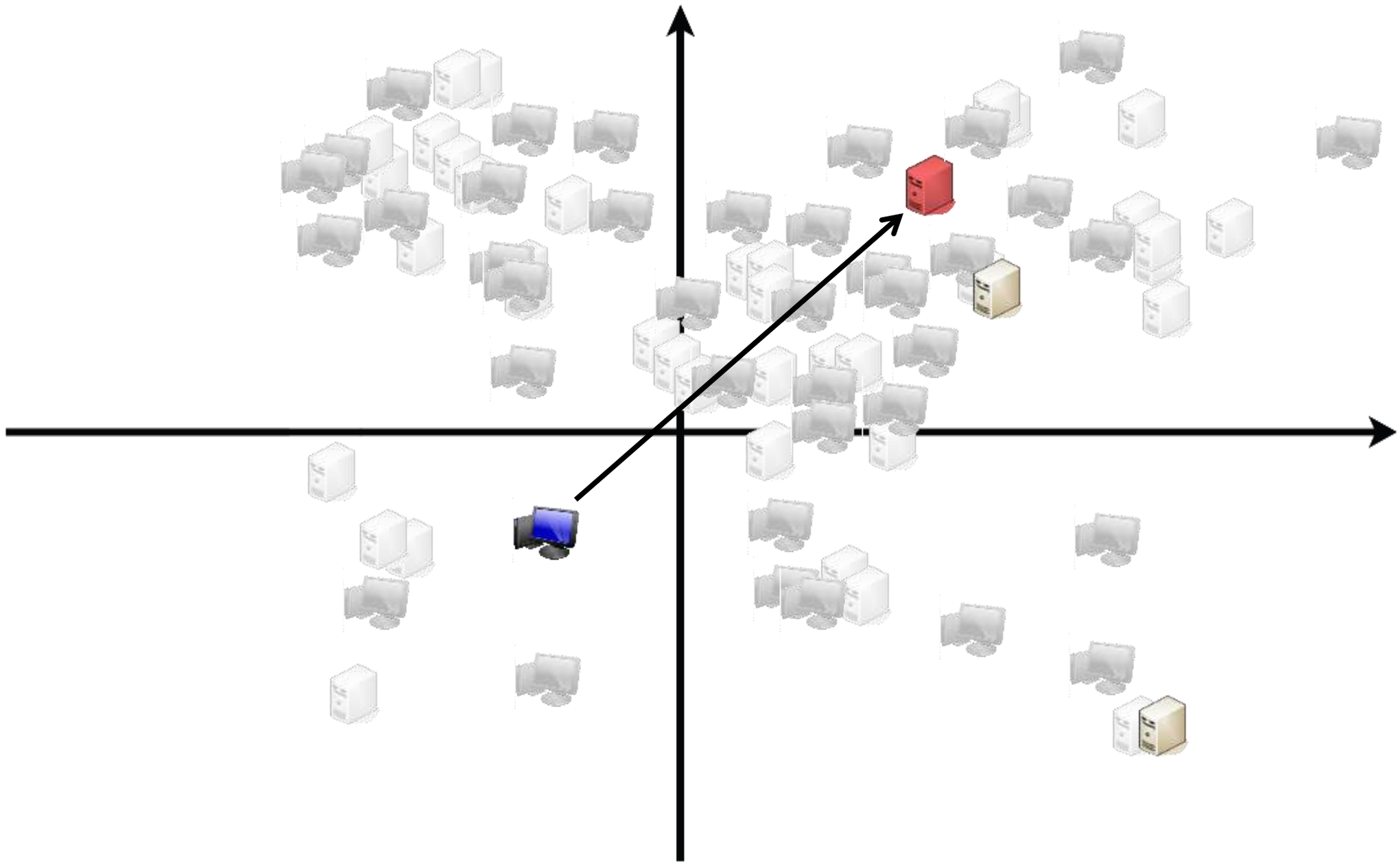
The conditional probability of the failure of (C, R, S_2) given the failure of (C,R,S_1) is more than **50%**!!



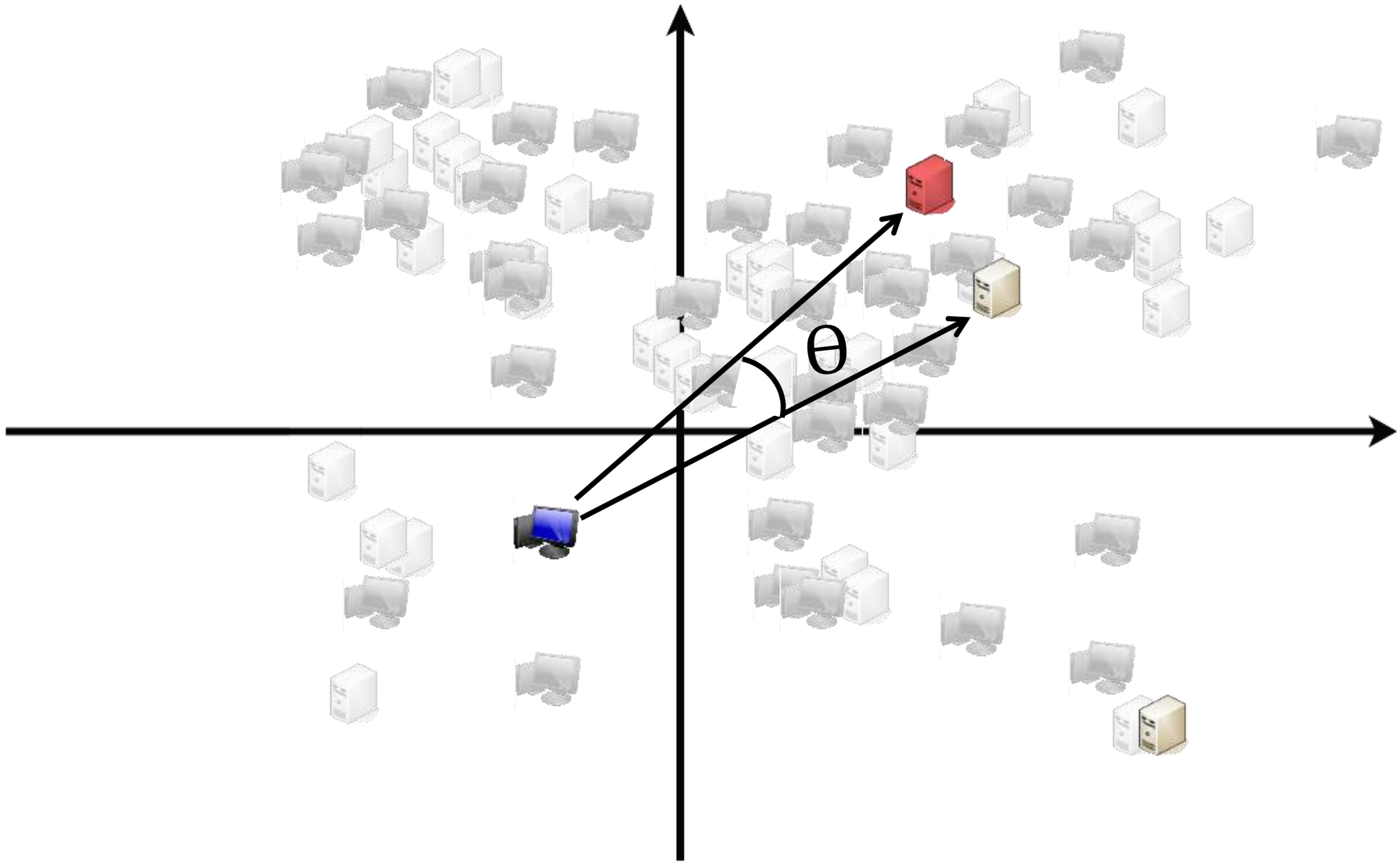
Conditional Failure vs. Angle



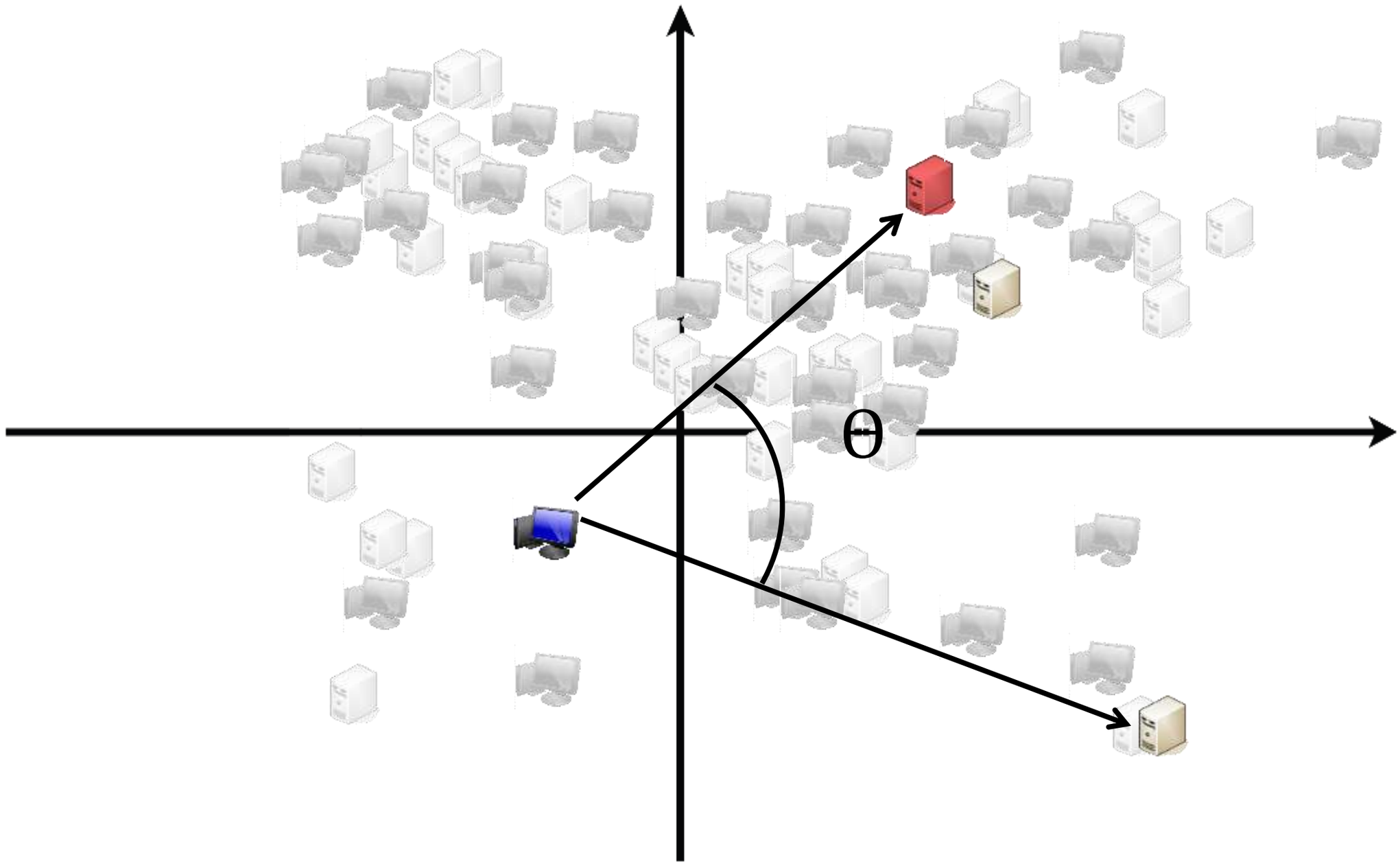
Conditional Failure vs. Angle



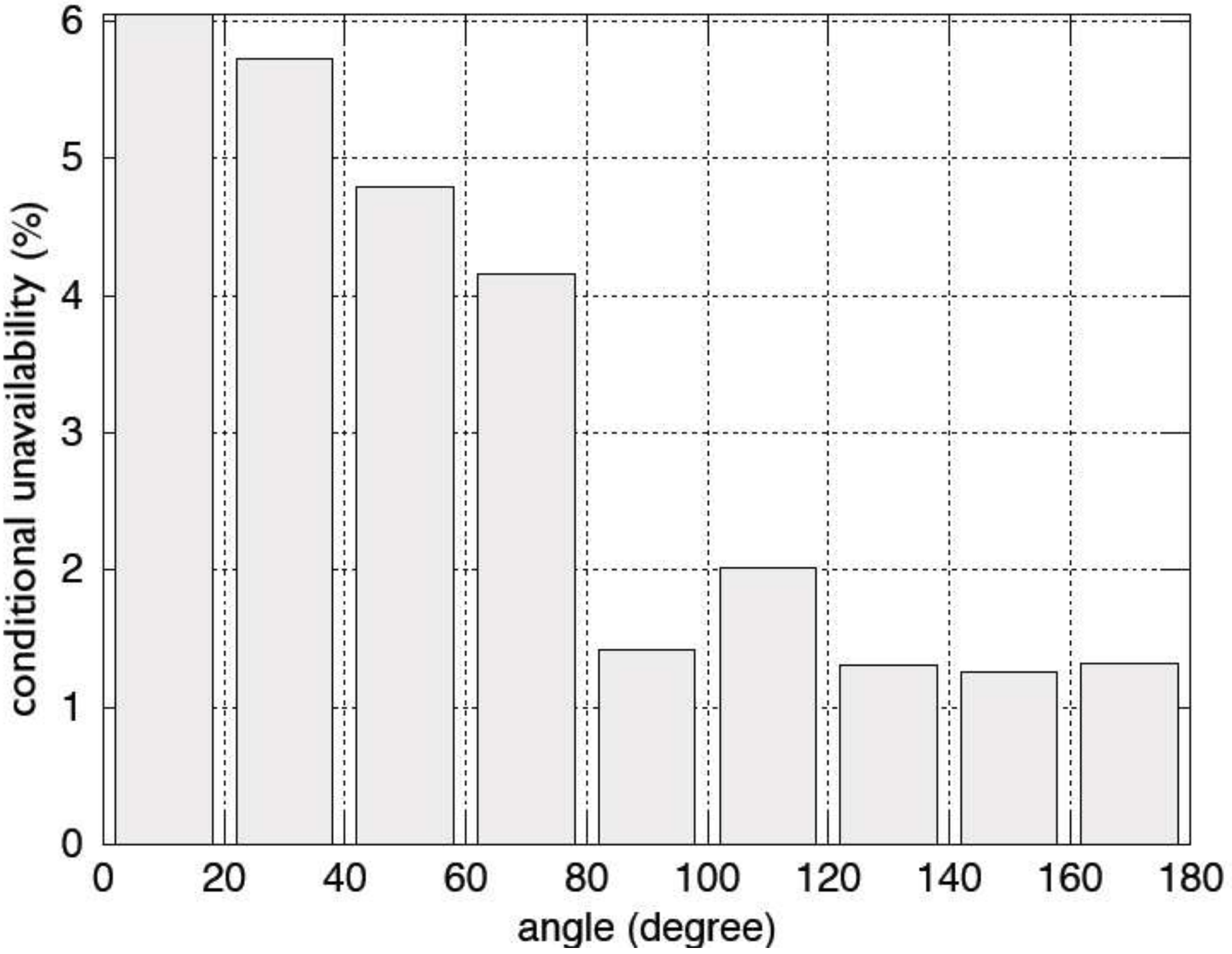
Conditional Failure vs. Angle



Conditional Failure vs. Angle



Conditional Failure vs. Angle



Estimations for Latency and Availability

- Per-client latency

$$L(c, S) = dist(c,s)$$

- Per-client availability

$$A(c,S) = 1- (F(c,S_1)*F(c,S_2|S_1)*\ldots*F(c,S_r|S_{r-1}))$$

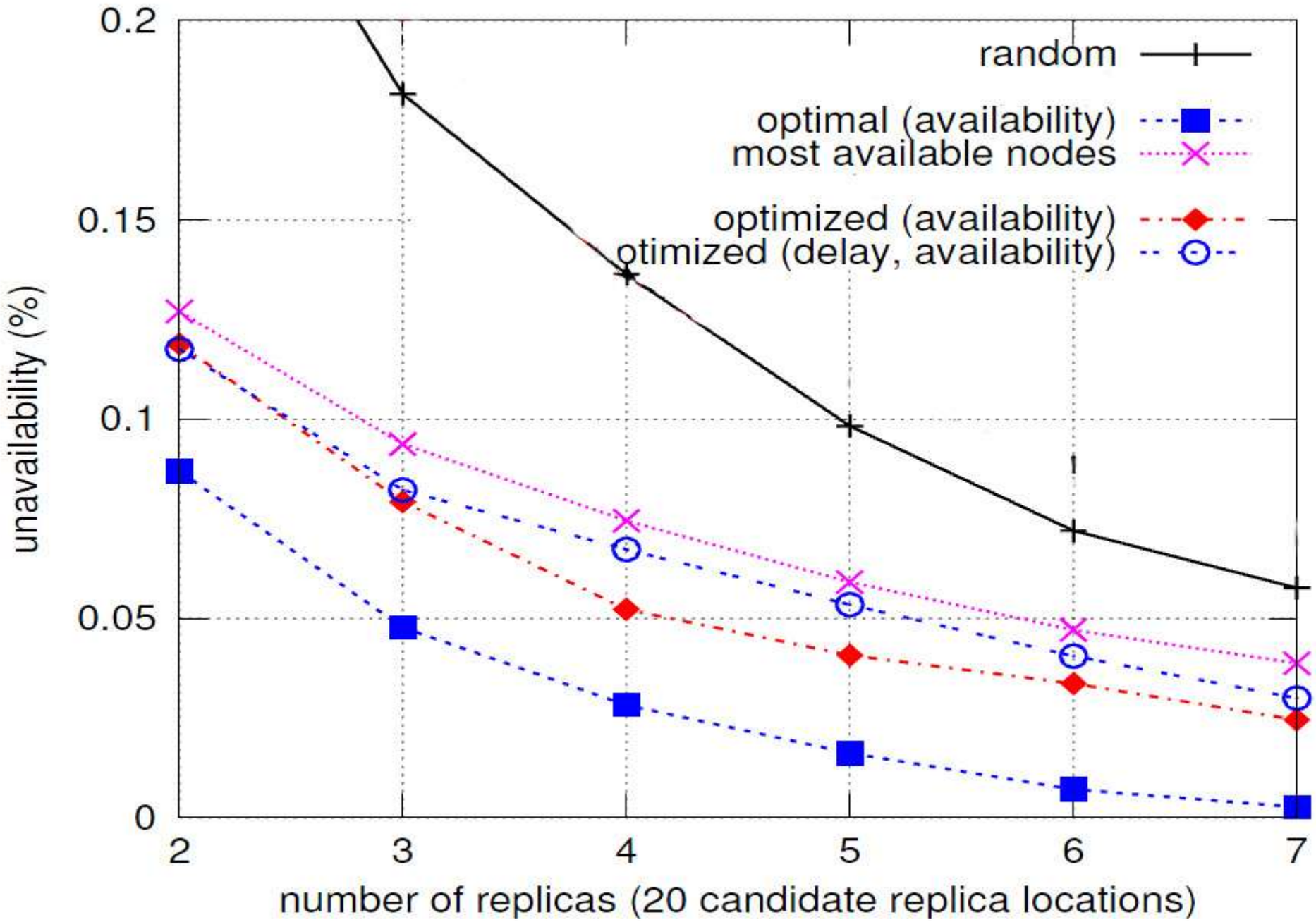
- Utility function to combine latency and availability

$$U = \frac{\overline{A}}{\overline{L}}$$

Simulation Settings

- Java simulator
- Traceroute and ping data collected from ~ 100 PlanetLab nods for a month
- Randomly select some nodes as servers
- The rest are clients

Unavailability vs. Number of Replicas



Outline

- Background
- Network Coordinate System
- Data Replication
  - Data Replication for Performance
  - Data Replication for Performance and Availability
- Conclusion

Conclusion and Future Work

- Improves the average user access latency by 35%
- Improves the overall availability
- Designs the utility function to take into account both latency and availability

- Needs more realistic dataset
- Better utility function
- Non-exponential algorithm (Greedy…)
- Take inter-datacenter cost into account

Questions?

THANK YOU!