Pegasus: Load-Aware Selective Replication with an In-Network Coherence Directory

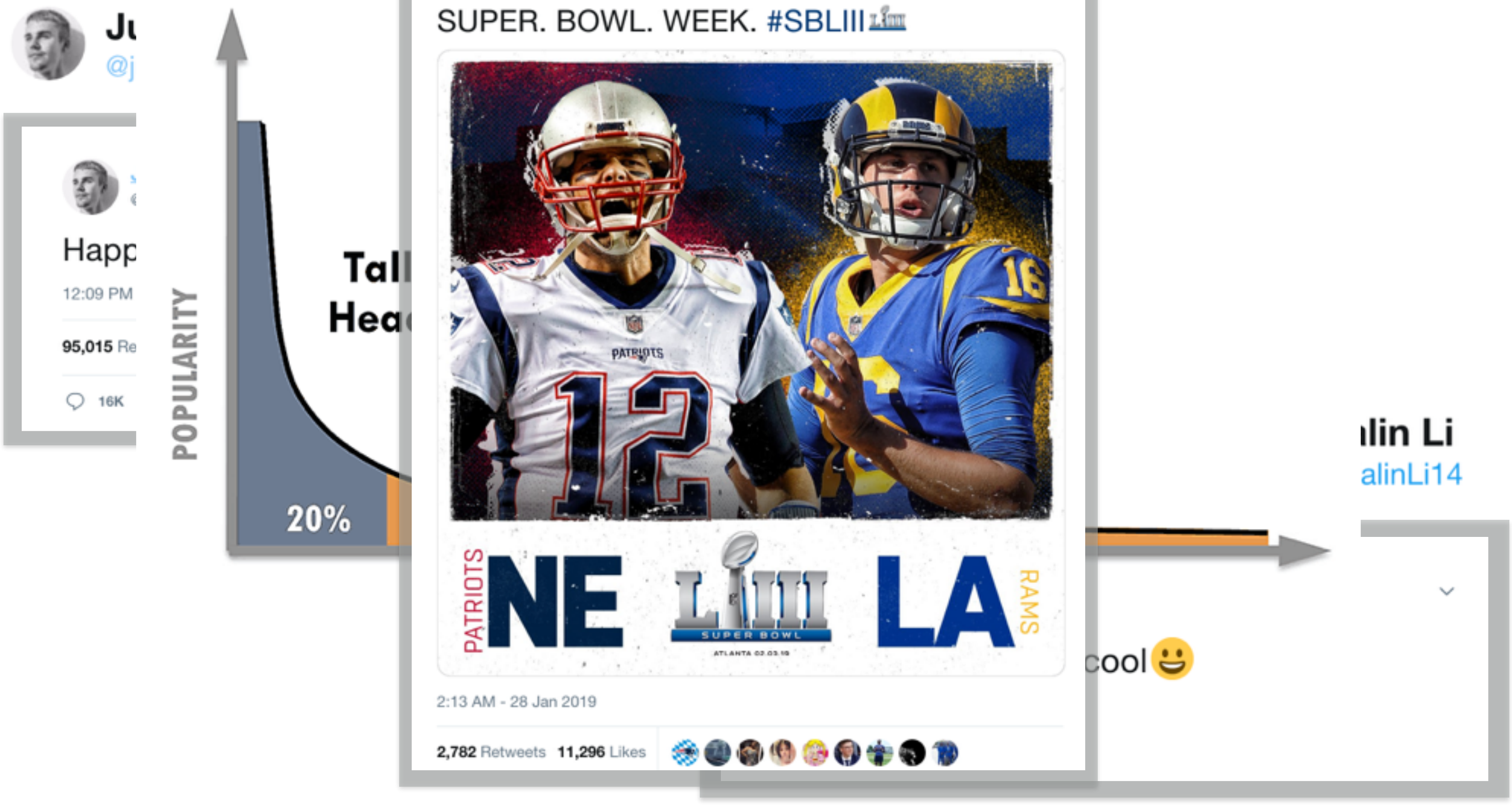
Jialin Li, Jacob Nelson, Ellis Michael,
Xin Jin, and Dan R. K. Ports

UNIVERSITY of WASHINGTON

Microsoft Research

JOHNS HOPKINS UNIVERSITY

Many workloads are **skewed** and **dynamic**



Skewed workloads lead to **load imbalance**

One approach: use caching to handle skewed workloads

Caching Layer

Limitations of Caching

- Caching layer needs to be **magnitude-faster** [Fan et al. '11]
    - Building such a layer for fast in-memory storage systems is challenging!
- Only effective for **read-heavy** workloads

Pegasus' Approach

provable load balancing for both **read-heavy** and **write-heavy** workloads

Selective Replication

In-Network Coherence Directory

Selective Replication

Q: How many objects should we replicate?

Storage Overhead **Low**

Load Balancing **Weak**

Storage Overhead **High**

Load Balancing **Strong**

Replicate Fewer Objects

Replicate More Objects

We only need to replicate the most popular **O(nlogn)** objects

n is the number of **storage servers**

Forward request to the **least-loaded server**

(generalization of previous result [Fan et al. ’11])

Challenges of Selective Replication

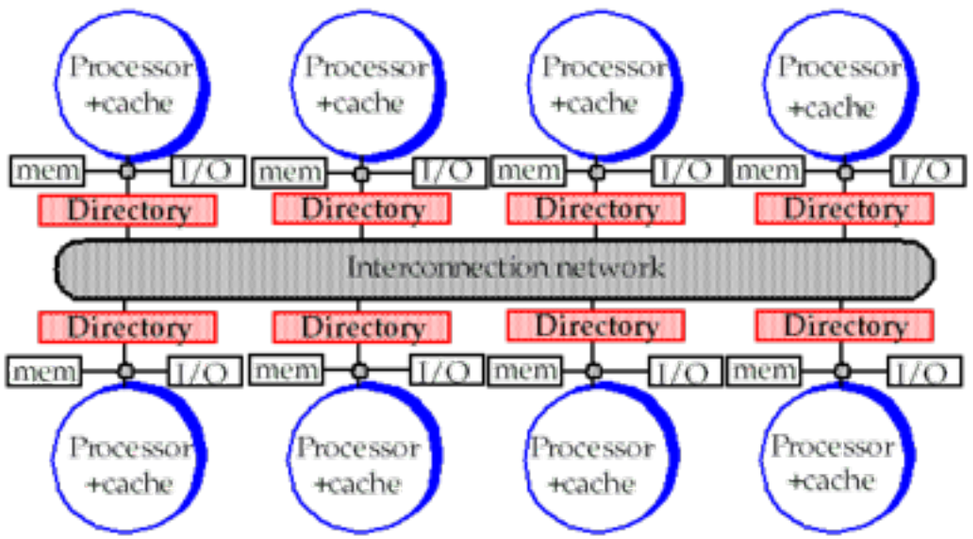
- How to track the most popular O(nlogn) objects?
  - Object popularity changes constantly
- How to manage the replica set?
- How to route requests to the least loaded server?
- How to ensure consistency?

Our Solution:

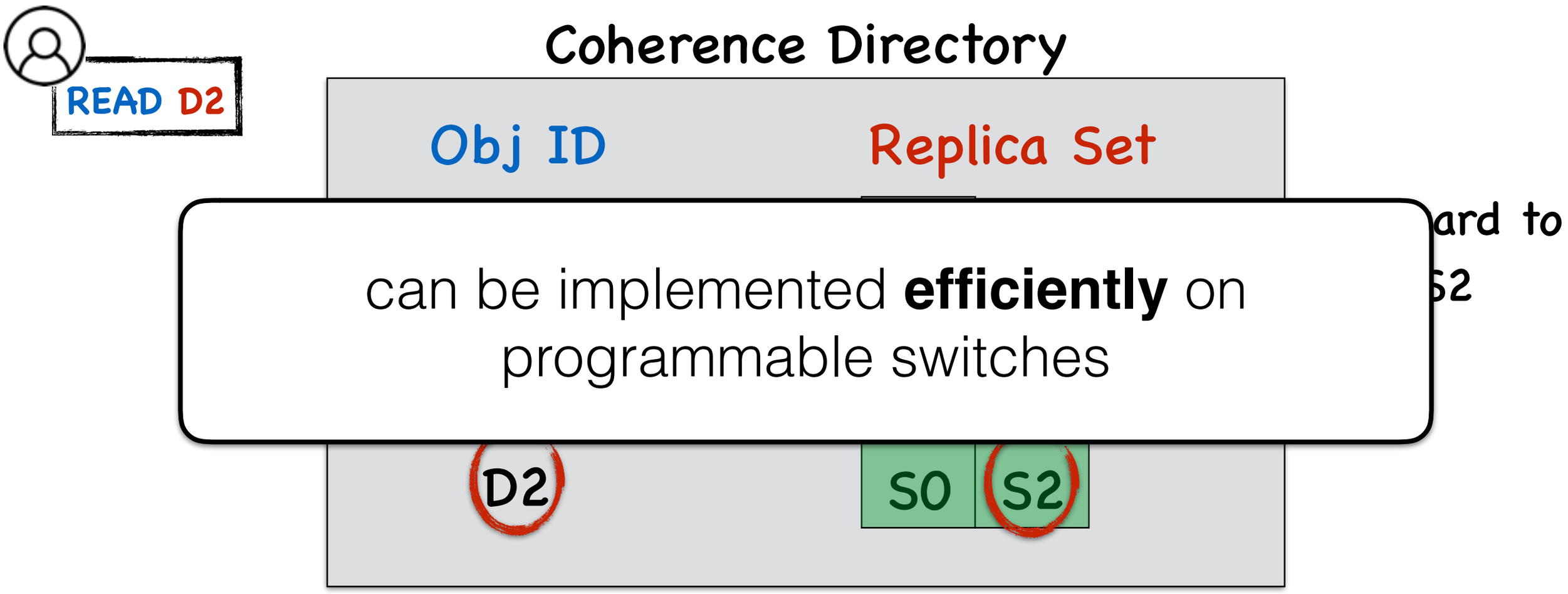
In-Network Coherence Directory

What is a **coherence directory?**

- Widely used in multi-processor architectures and distributed shared memory systems
- Tracks state of each cache block and location of shared copies
- Coordinates coherence protocol



Coherence directory applies nicely to selective replication



can be implemented **efficiently** on programmable switches

Implementing coherence directory **in the network**

Lookup Table

Match	pkt.obj = A1	pkt.obj = B4	pkt.obj = D2
Action	index = 0	index = 1	index = 2

Replica Set

0	1	2	
S1	S2	S0	Reg Array 0
	S1	S2	Reg Array 1
	S0		Reg Array 2

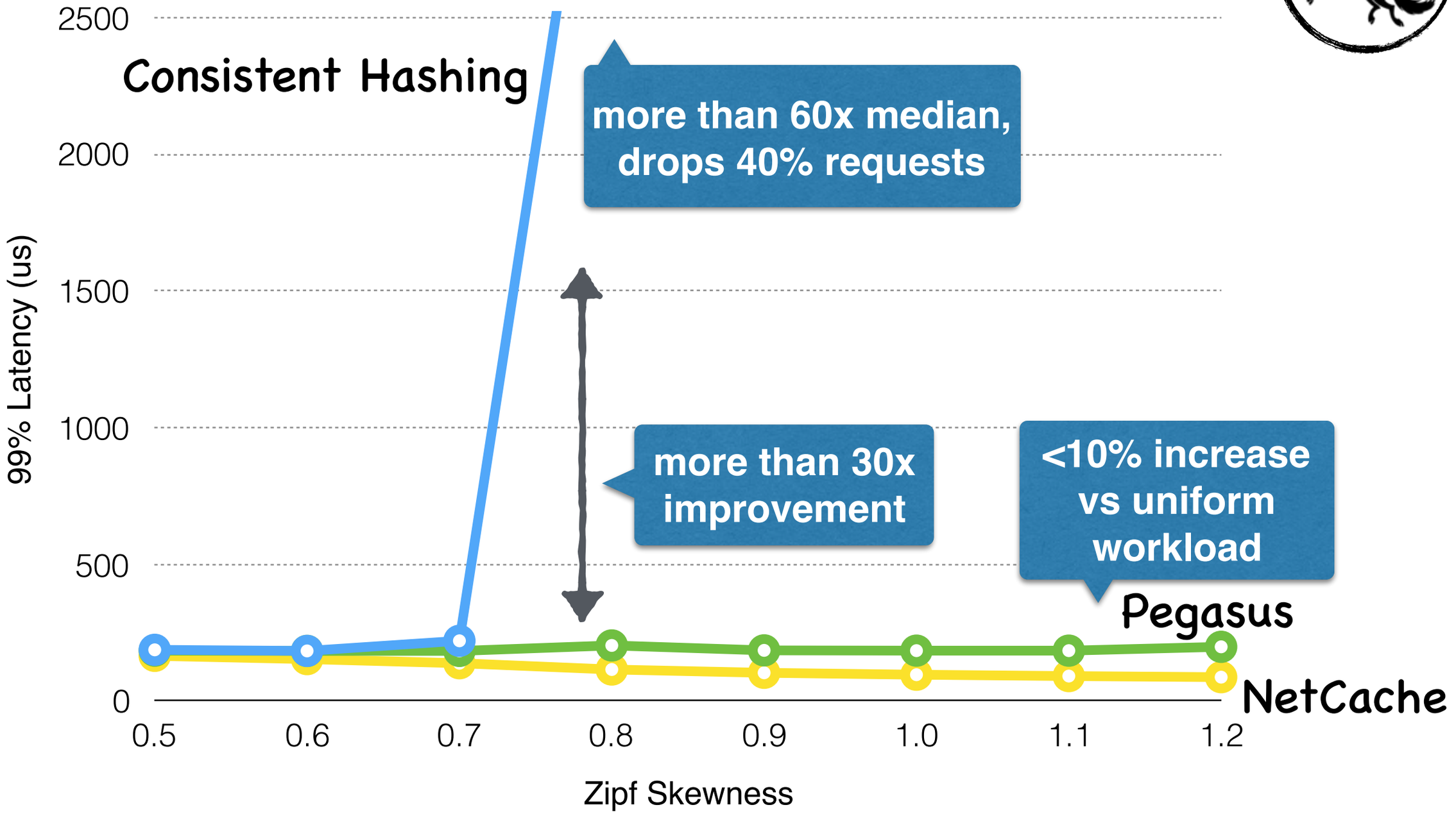
Forwarding Table

Match	server ID = S0	server ID = S1	server ID = S2
Action	pkt.dst = 10.0.0.1	pkt.dst = 10.0.0.5	pkt.dst = 10.0.0.7

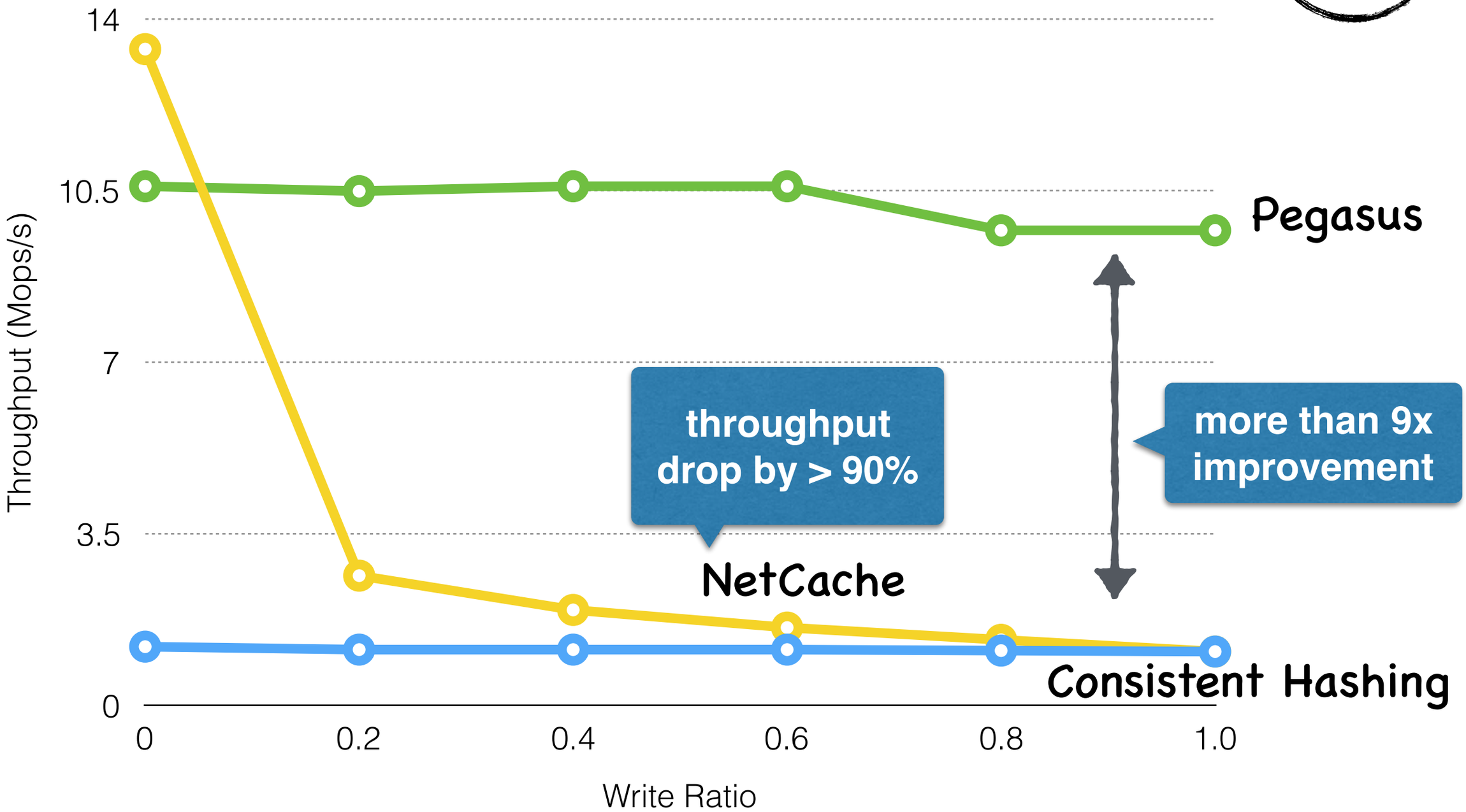
Challenges of In-Network Coherence Directory

- Track which server has the **minimum load**
  - Switch-based load prediction
- Track most popular O(nlogn) objects
- Balance load for **writes**
  - Dynamic replica set
- Ensure **strong consistency**
  - Version-based coherence protocol

Load balancing under highly skewed workloads



Load balancing under different read/write ratios



Summary

- Use programmable switch to improve load balancing of storage systems
- Selectively replicate the most popular objects
- Build in-network coherence directories to manage replicated objects
- 9x throughput improvement compared to consistent hashing