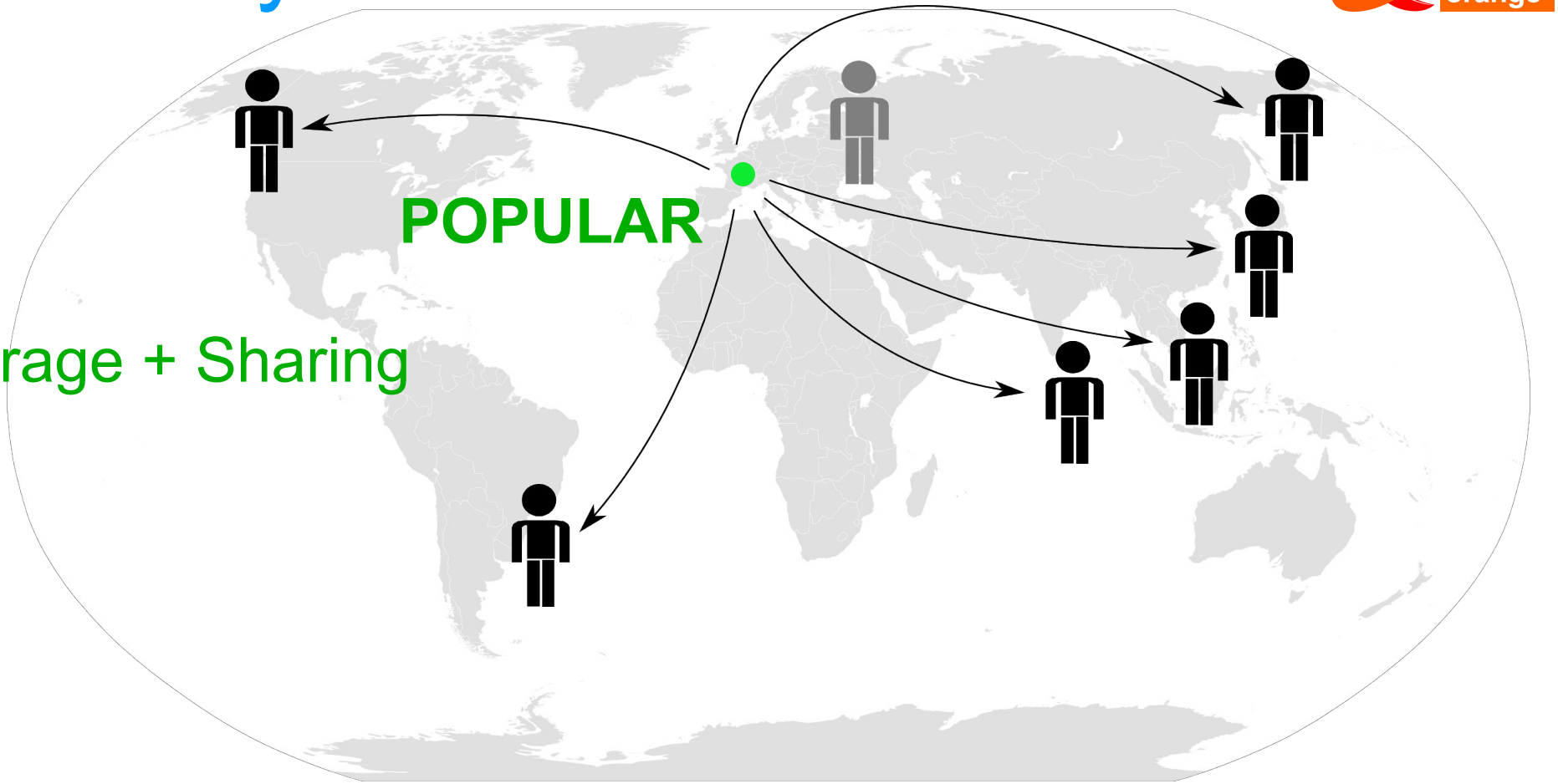
AREN: a popularity aware replication scheme for cloud storage

Guthemberg Silvestre[1,2], Sébastien Monnet[1],
Ruby Krishnaswamy[2] , and Pierre Sens[1]

[1]LIP6/UPMC/CNRS/INRIA
[2]Orange Labs

ICPADS 2012, Singapore
17 December 2012

Challenges for content delivery



Storage + Sharing

CDN ⟷ **REPLICATION** ⟷ **SLA**

State-of-the-art

- CDN architectures
  - Infrastructure-based (datacenter): Akamai, Google
  - Hybrid design: NaDa [FP7'11], Echos[Laoutaris'08]
- Content replication
  - Uniform and fixed: GFS [Ghemawat'03],CEPH[Weil'06]
  - Adaptive: non-collaborative cache(LRU), EAD [Shen'10], Skute [Bonvin'10], DAR [Zhou'12]
- SLA
  - Poor content availability: uptime (Amazon S3)
  - Very high content availability: deadline-aware approaches (D3[Wilson'11])

Open issues

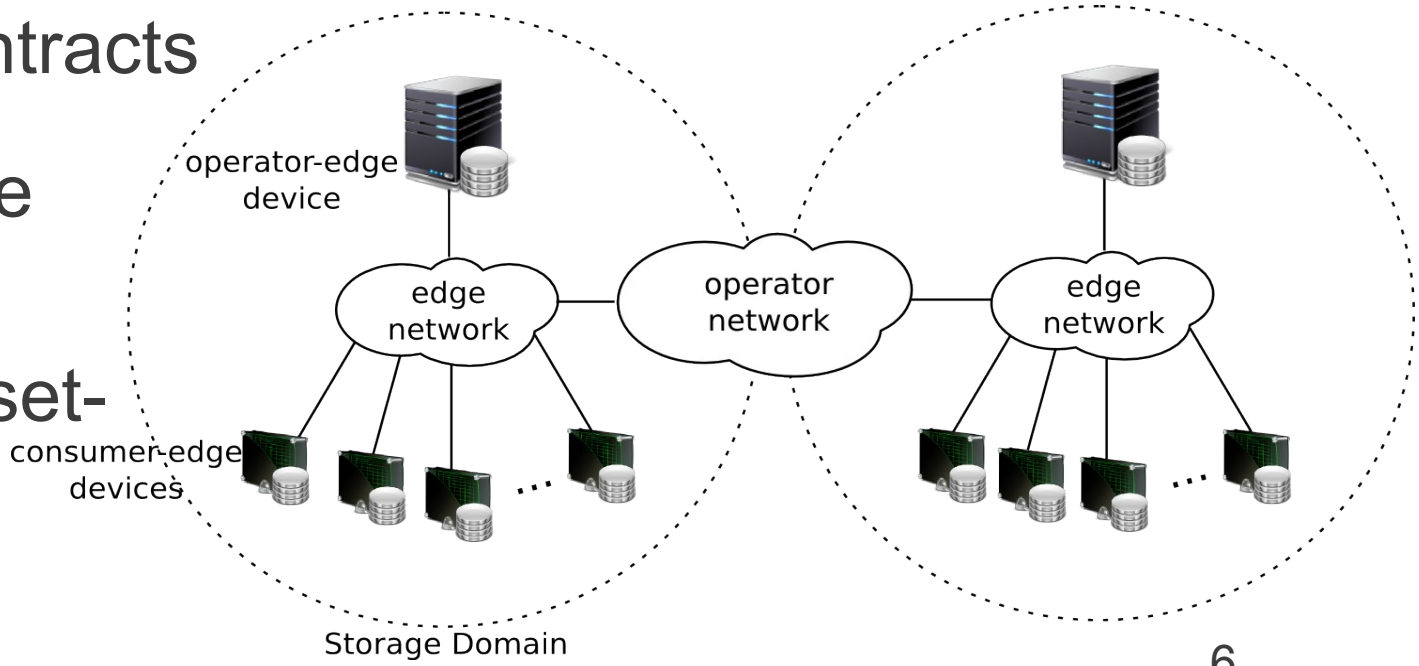
- How do we handle edge network devices for object-based storage systems?
- Where do we place clients' objects?
- How many replicas per object should the system create?
- How could we prevent SLA violations and optimize edge resources utilization?

Main contributions

- **Performance evaluations (PeerSim simulations)**
  - Replication schemes and popular content
  - Strict SLA metrics on edge networks
- **AREN: an Adaptive Replication scheme for Edge Networks**
  - Provides high-quality content delivery for popular content
  - Prevents SLA violations, improves cloud storage usage and aggregate bandwidth

Content distribution system for edge networks with Caju

- Hybrid architecture organized in storage domains
  - Two classes: operator-edge and consumer-edge
- Workload: object size, request rate, popularity distributions
- Strict SLA contracts
  - Transfer rate
  - A SLA per consumer (set-top box)

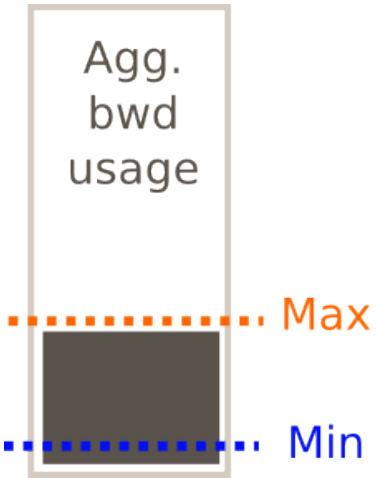


Replication schemes

- Uniform with fixed number of replicas
  - *n* replicas
  - Scheduling: Each request is served by at most *R* nodes with equal load
- Non-collaborative LRU caching
- DAR [Zhou et al.,INFOCOM'12]
  - Goal: balance the expected bandwidth load per node
- Unlimited

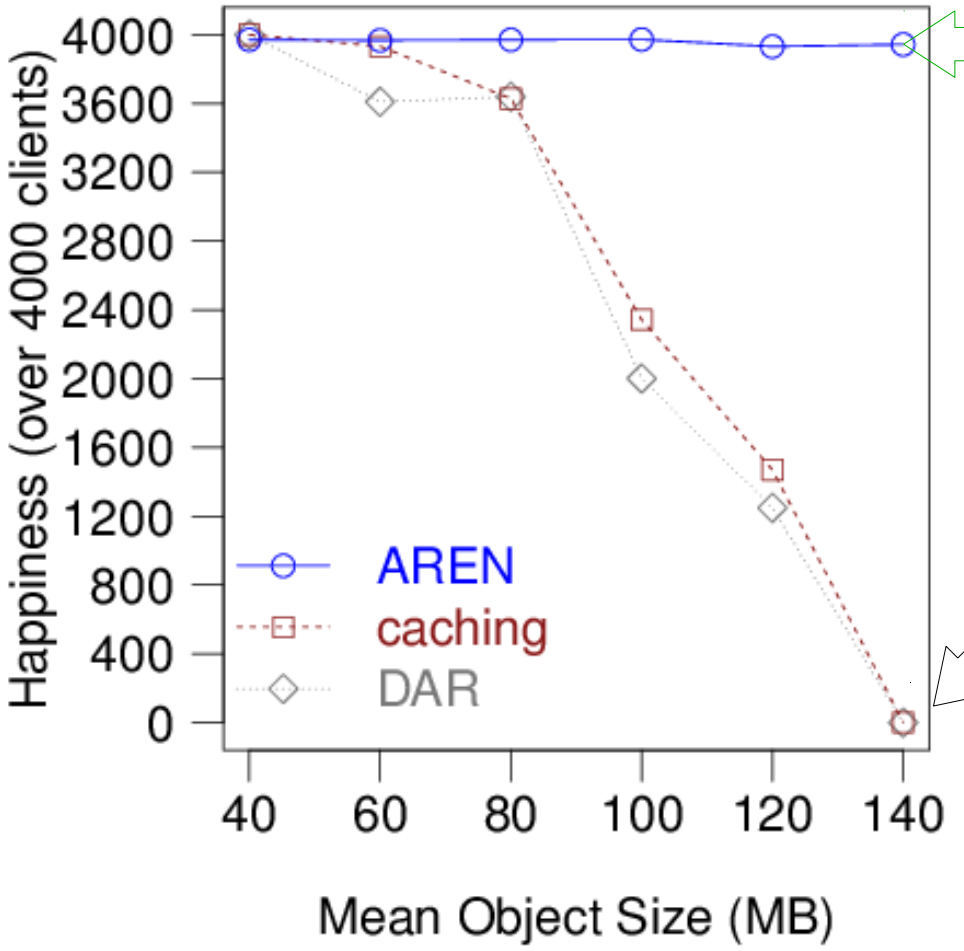
AREN: Adaptive Replication scheme for Edge Networks

- The performance goals are twofold:
  - Reduce SLA violations
  - Improve the usage of edge resources
- Combine bandwidth reservation, collaborative caching, and thresholds
- Simple approach:
  - Popular content classification
  - Replicas maintenance



Evaluation results

Performance evaluation with increasing load

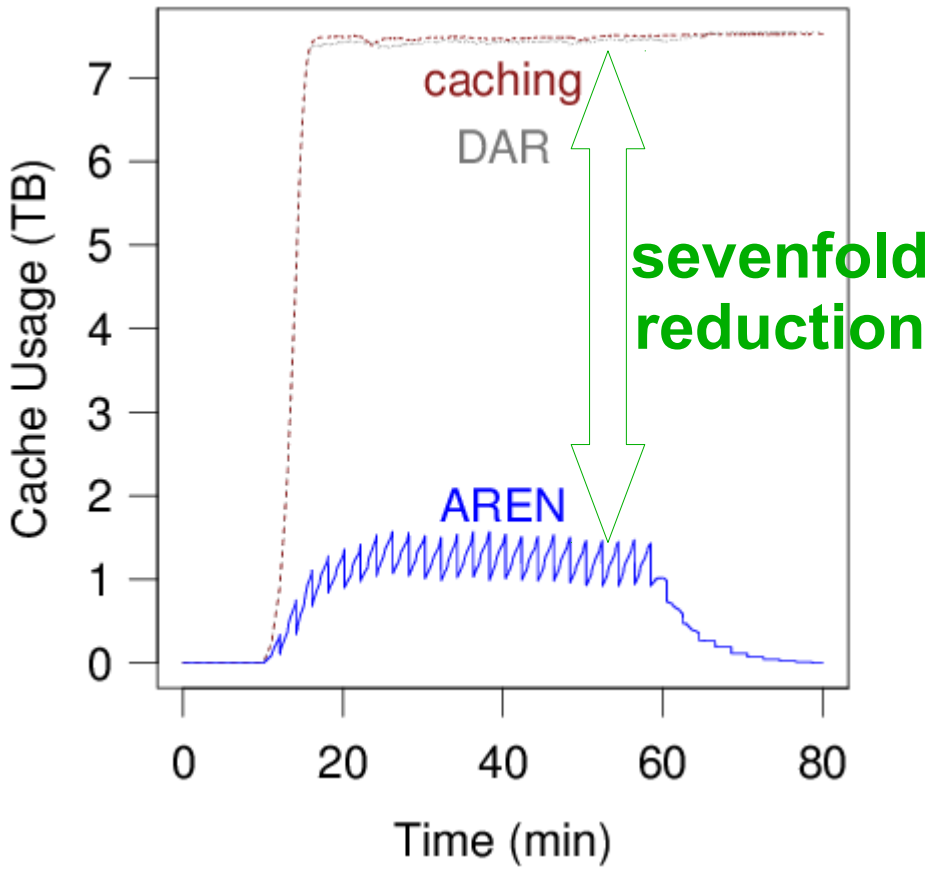


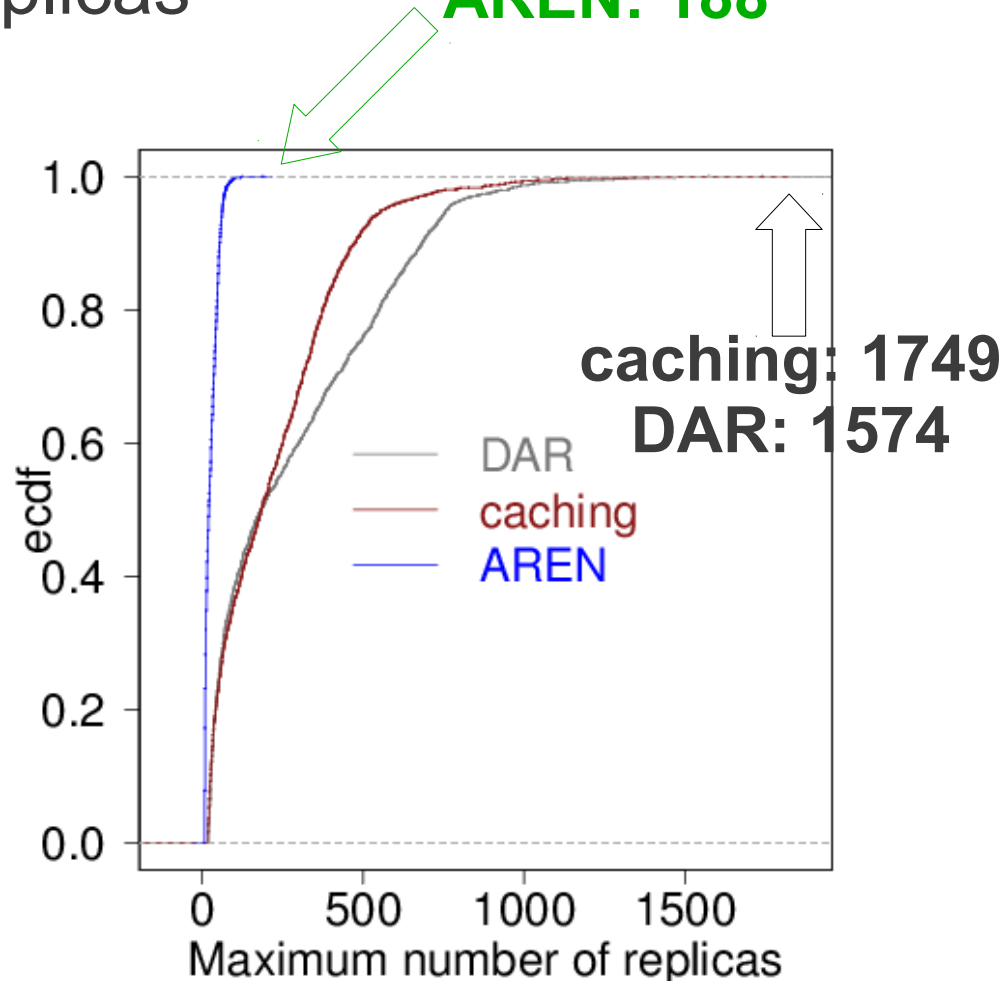
51 violations

caching: 27539 violations
DAR: 30071 violations

Evaluation results

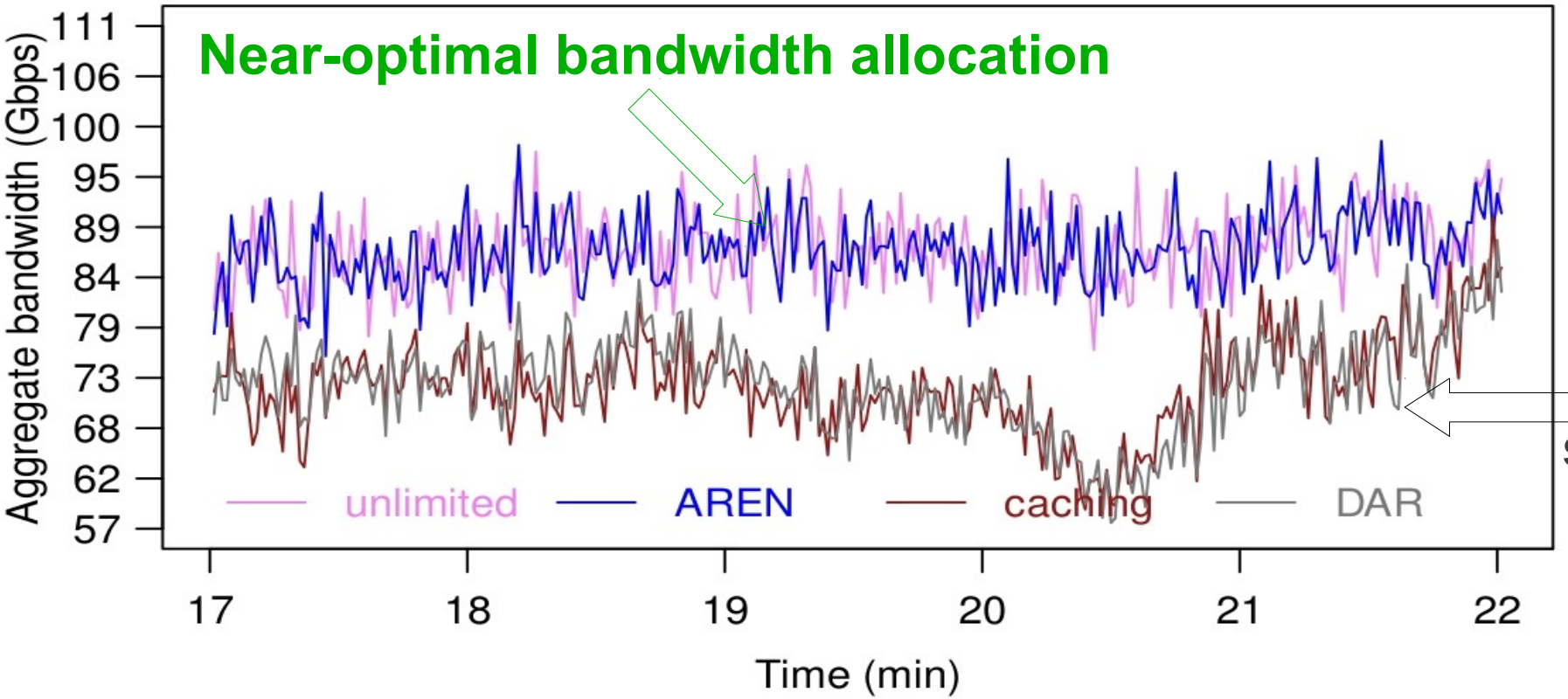
Cache size and number of replicas





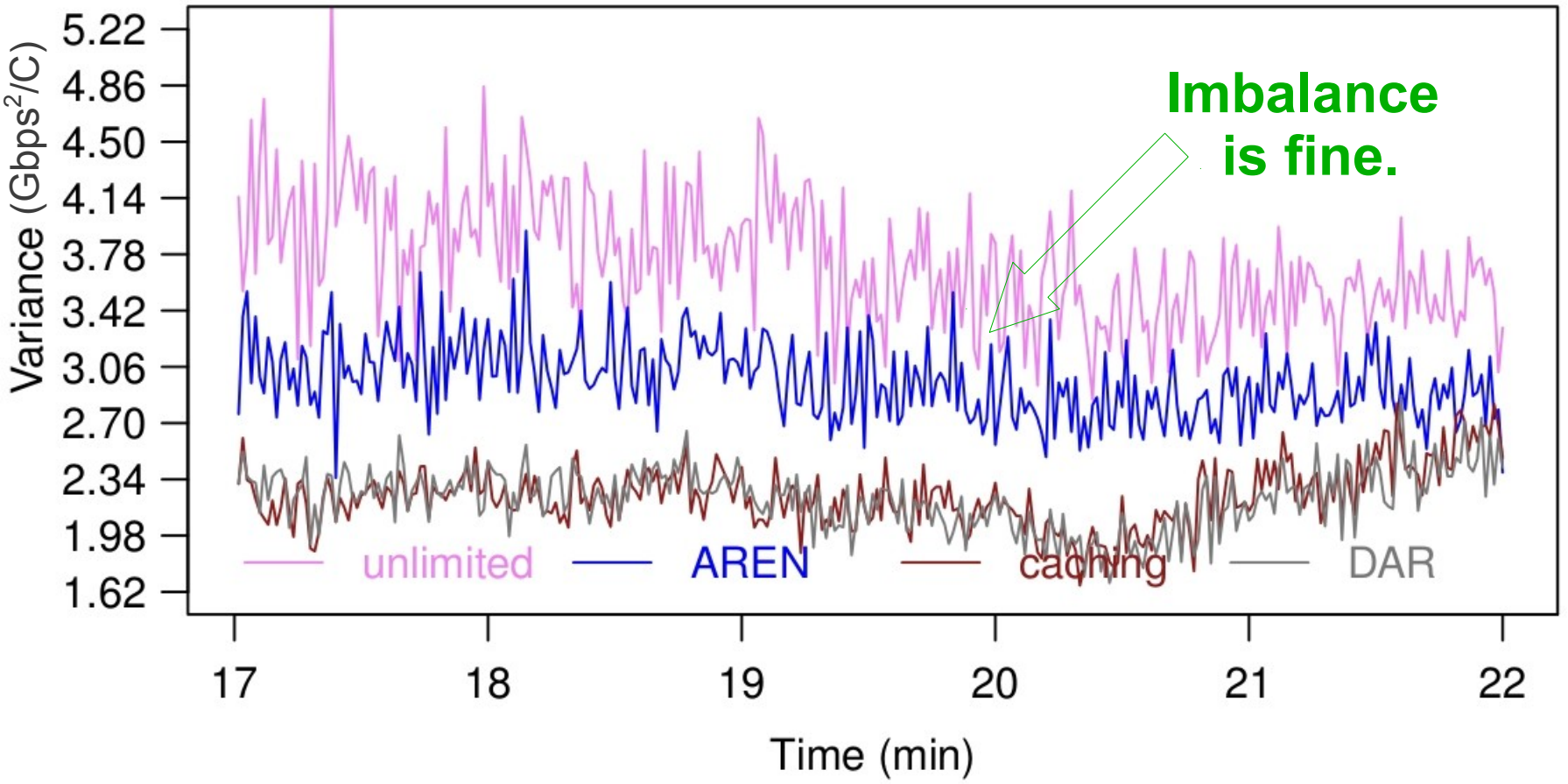
Evaluation results

Aggregate bandwidth during utilization peak



Evaluation results

An indicator of the spread of bandwidth allocation during utilization peak



Conclusions

- AREN, a novel adaptive replication scheme for cloud storage in edge networks
  - Enforces strict SLA contracts efficiently
  - Improves resource allocation
- AREN tracks bandwidth reservation on edge nodes for operating collaborative caching mechanism
- Perspectives: build and evaluate a proof-of-concept prototype on PlanetLab

For further information:

http://lip6.fr/Guthemberg.Silvestre

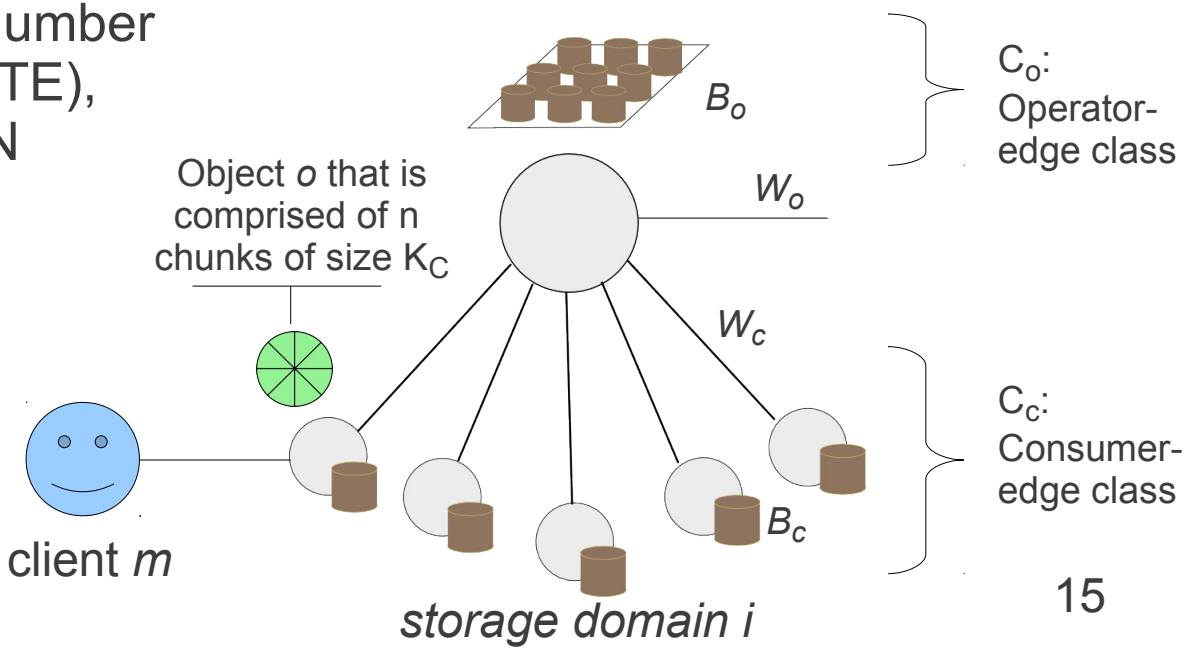
AREN: a popularity aware replication scheme for cloud storage

Backup slides.

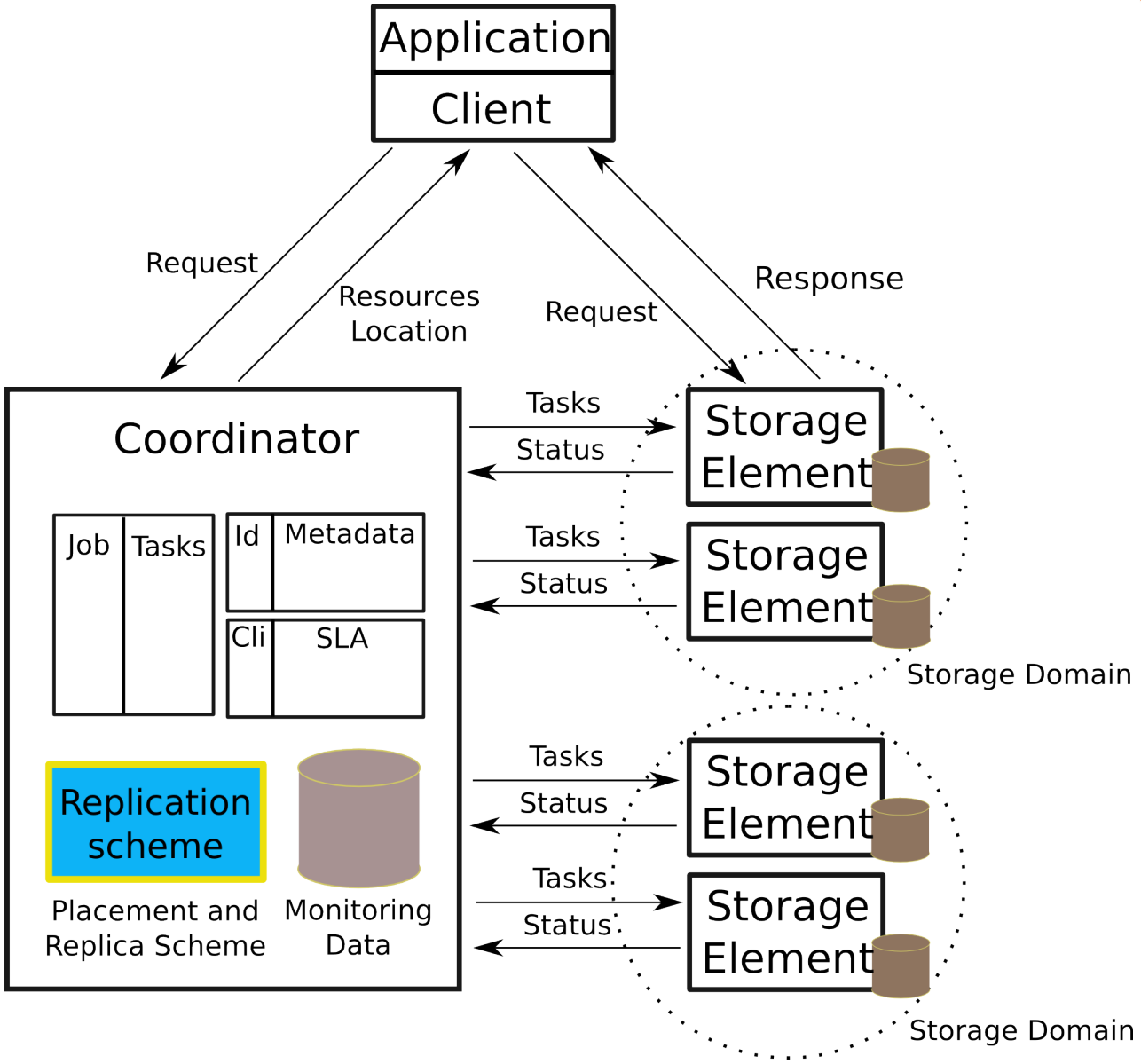
Caju: Model

- Storage domain, storage nodes and objects:
  - Storage domains are partitioned in two distinct classes, and that basically differ in storage and network capacity
  - Any client *m* is connected to the system through a consumer-edge device, and assigned to a home storage domain
  - Objects are divided in chunks

- Clients are able to do any number of request (GET,PUT,DELETE), system does REPLICATION
- Clients' SLA
  - Transfer rate:
  - Acceptable number of violations



Caju: Main functional blocks

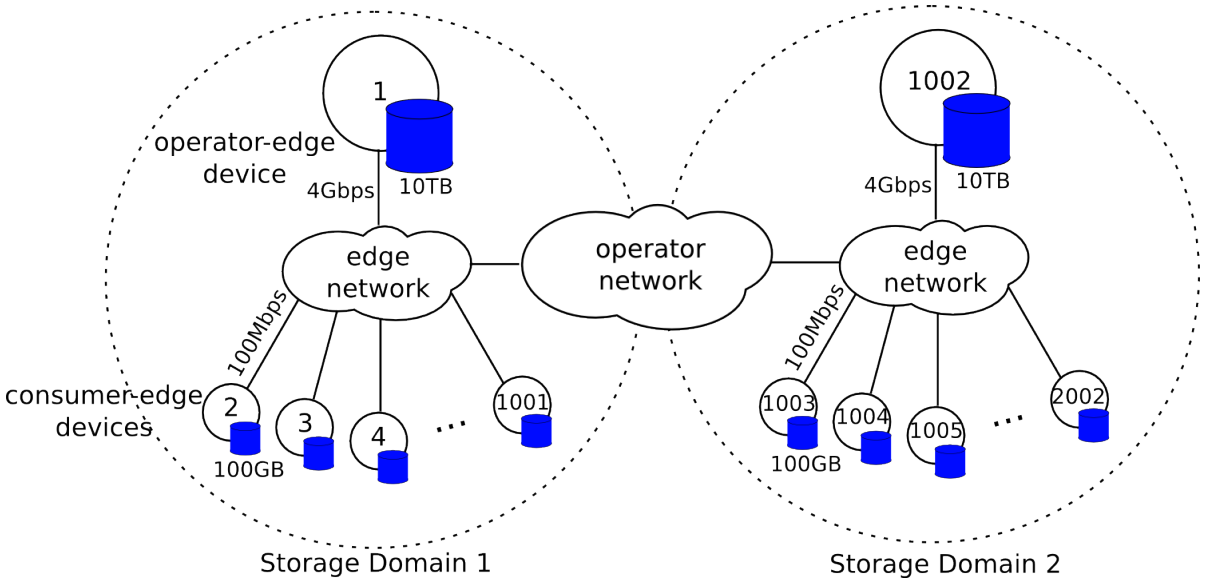


Simulations

- Protocol stack (on PeerSim)

Application	**Target service:** Multi-purpose object-based storage, and sharing **Workload:** Users, objects, interactions, and SLA
Storage	**Handle requests:** Jobs and tasks, storage resources **Quality Control**: Content popularity, SLA, and storage/network resource allocation
Transport	**Communication interface:** keep source-destination map, collect and export network flow information, handle network connection events
Network	**Network functionalities:** data transfer between nodes, network resources, fair-sharing bandwidth, reservation, deadline enforcement

Evaluation: Scenario



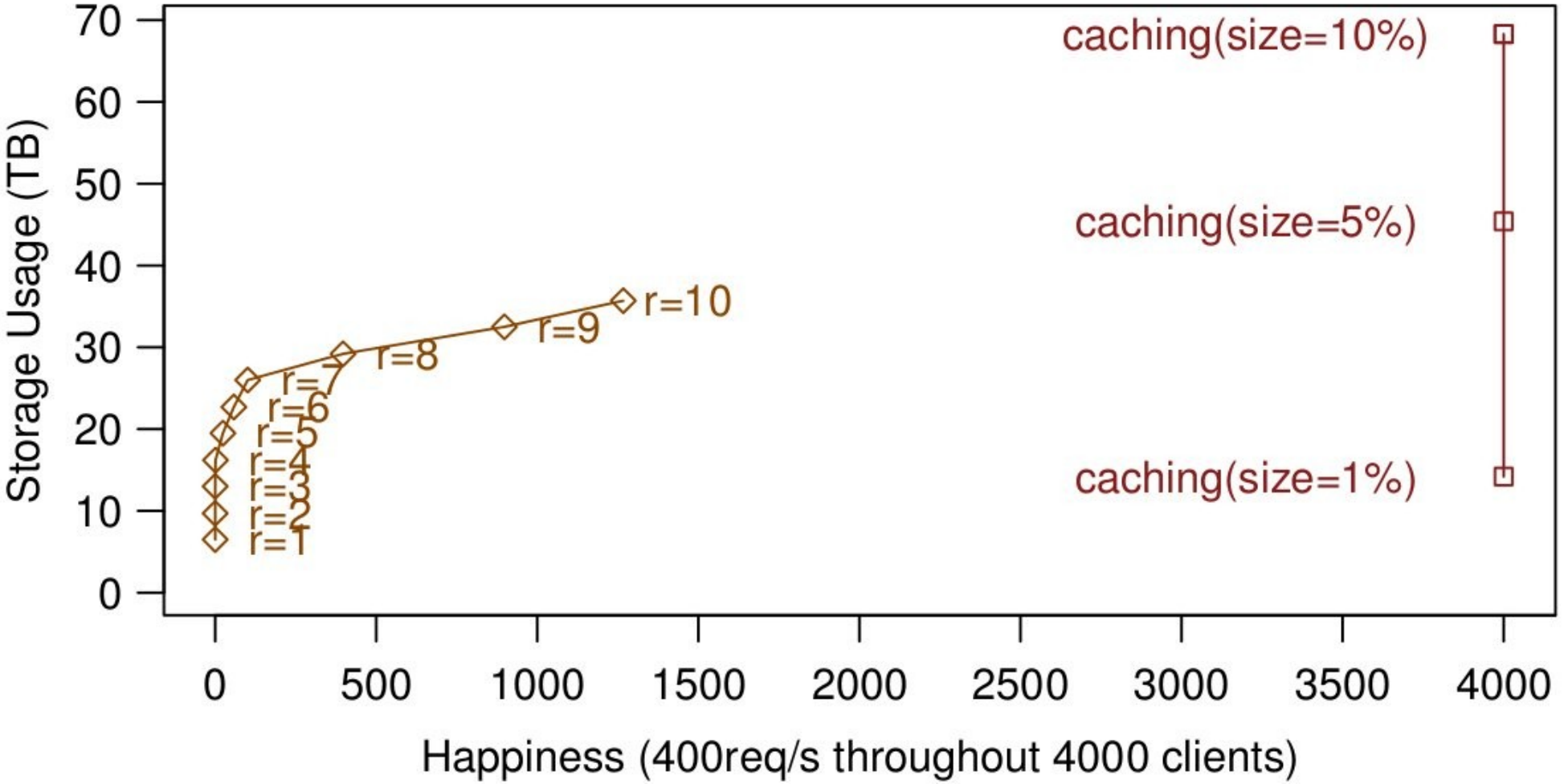
Three SLA contracts:

Rate (chunks/second)	%clients
41	40
21	40
14	20

Workload	
Requests per client	uniform
Experiment duration	1h 40min
Mean requests per second	400
Requests division	5% of PUTs, 95% of GETs
Object size (follows Pareto)	shape=3, smallest=26MB, biggest=1.6GB
Content popularity (Zipf-Mandelbrot)	shape=0.8, cutoff=number of objects
PUTs (Poisson)	λ=PUTs per second
Popularity growth (Weibull)	shape=2, scale $\propto$ duration

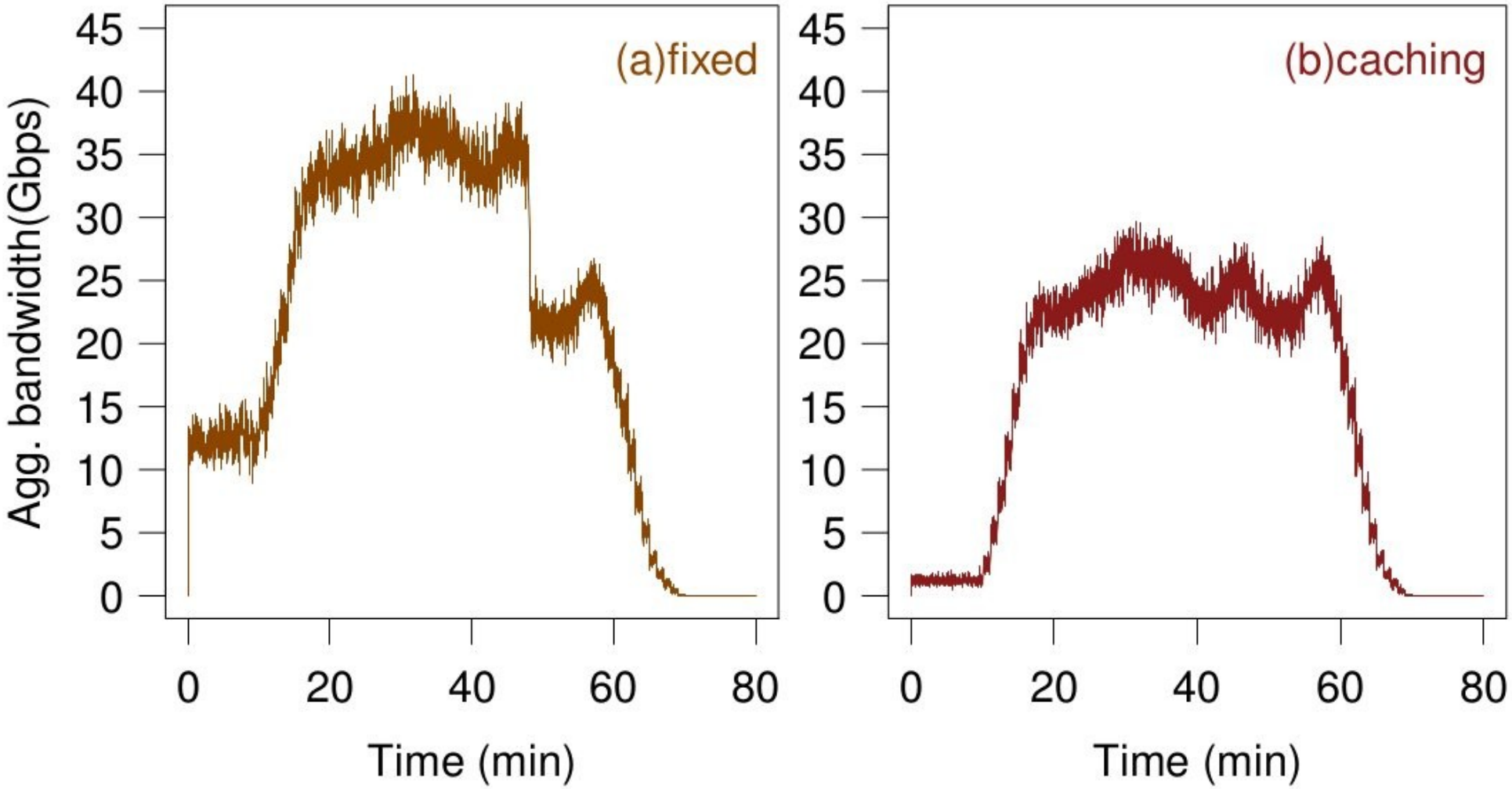
Evaluation results

Reduce required storage size for replicas



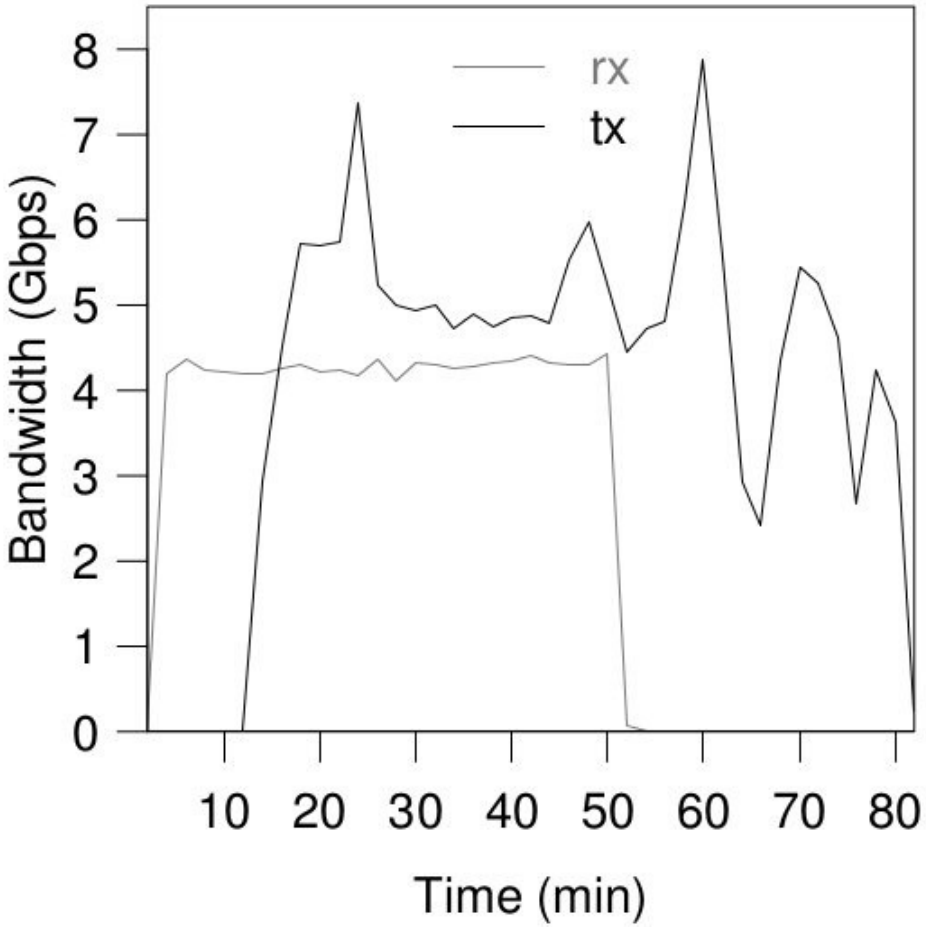
Evaluation results

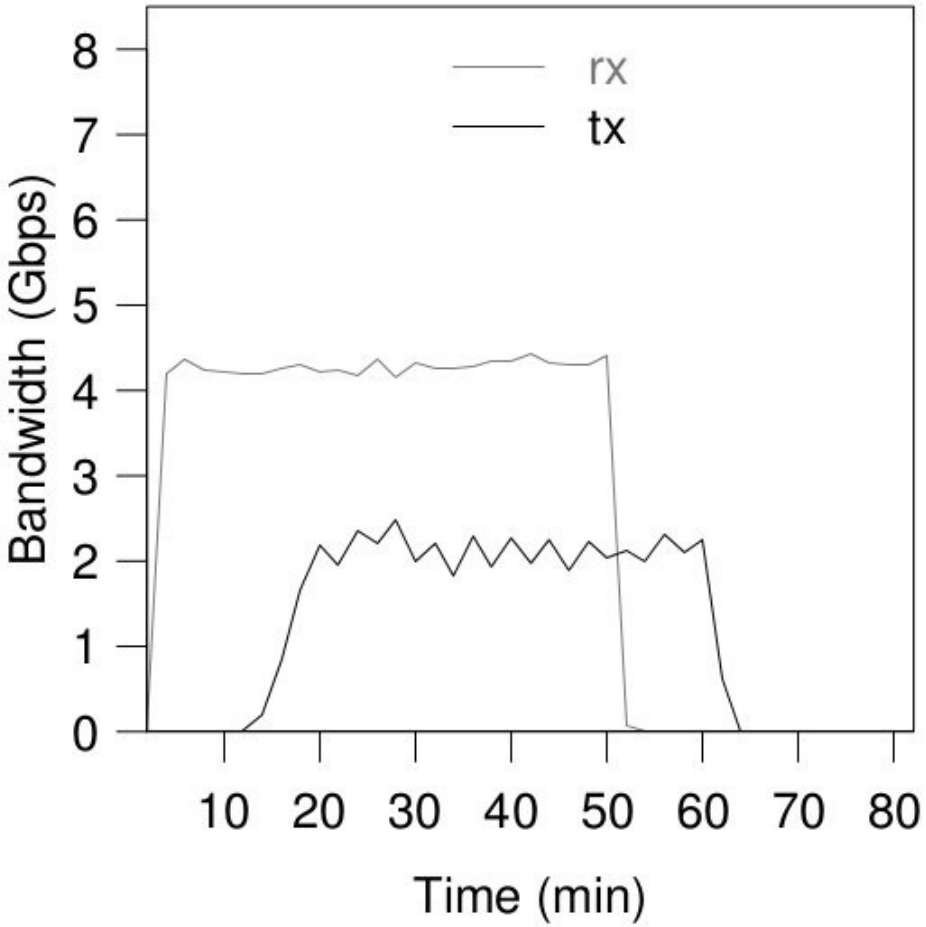
Reduce bandwidth usage



Evaluation results

Reduce operator-edge bandwidth usage with *divide-and-conquer* policy





Evaluation results

Reduce inter-domain bandwidth usage with nearest source selection policy