A Case for Fluid Replication

by Brian Noble, Ben Fleis, Minkyong Kim
Presenter: Ionut Trestian

Motivation

- Scaling distributed systems leads to unpredictable performance
- More reasons for unpredictability:
  - Rapid changes in users interests
  - Network resource demand changes
  - Mobility of users

Current solutions

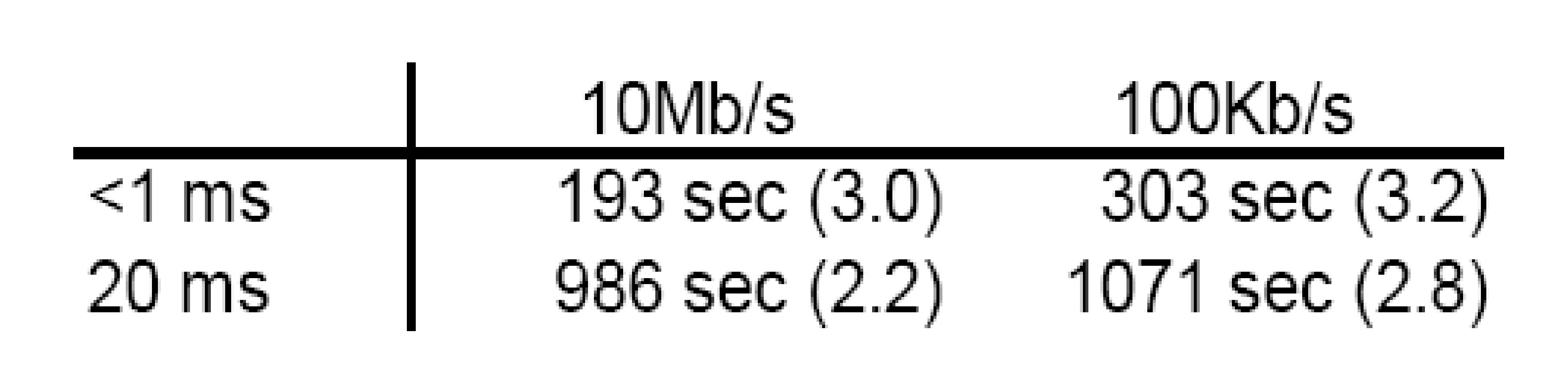
- Cluster based replication
  - Provider adds more machines
  - Doesn't actually address all issues
- Peer replicated systems
  - Caching
  - Clients are resource limited and aren't trustworthy

Solution – Fluid replication

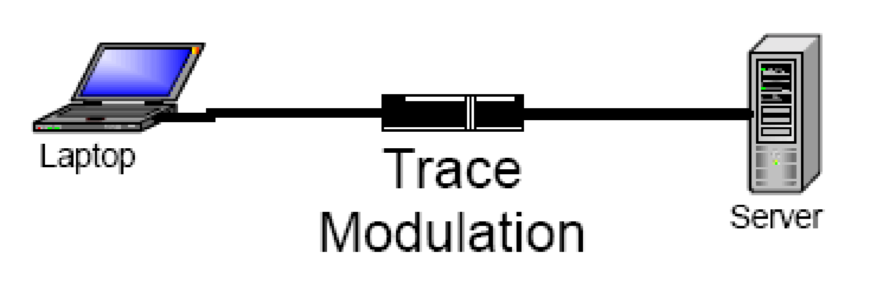
- Replication still used
- Replicas created anywhere, instantiated by clients
- Main component: WayStation

Impact of networking costs

- NFS Server stored a small source tree compiled by the client



	10Mb/s	100Kb/s
<1 ms	193 sec (3.0)	303 sec (3.2)
20 ms	986 sec (2.2)	1071 sec (2.8)



Monitoring Network Performance

- Establish a baseline for clients when performance is Ok.
- Delays, packet loss, etc
- Estimator should be agile and stable
- Agile = reacts quickly
- Stable = does not react to transient changes in performance

$$E_n = \alpha O + (1-\alpha)E_{n-1}$$

Waystations

- When clients see that performance is poor
  - Find a Waystation to hold replica
  - Close enough to be useful
  - If clients are mobile this is hard
- Uses current routing infrastructure and multicast to locate closest Waystation.
- Lazily populate replica on Waystation

Consistency maintenance

- 2 aspects:
  - Strength of Guarantee – what clients can assume
  - Frequency of Guarantee – how often the guarantees are enforced
- Client far from service so replica posted close to client. But replica is also far from the service.
- The bottleneck is the distance between replica and server.

Strengths of guarantee

- Last writer.
  - No guarantees, each replica updates independently, updates logged, if updated in 2 places keep only one
- Optimism
  - Guarantees detection of conflicts, log and exchange service checks for serializable operations
- Pessimism
  - Guarantees prevention of conflicts. Exclusive access obtained by replica before each write

Frequency of guarantee

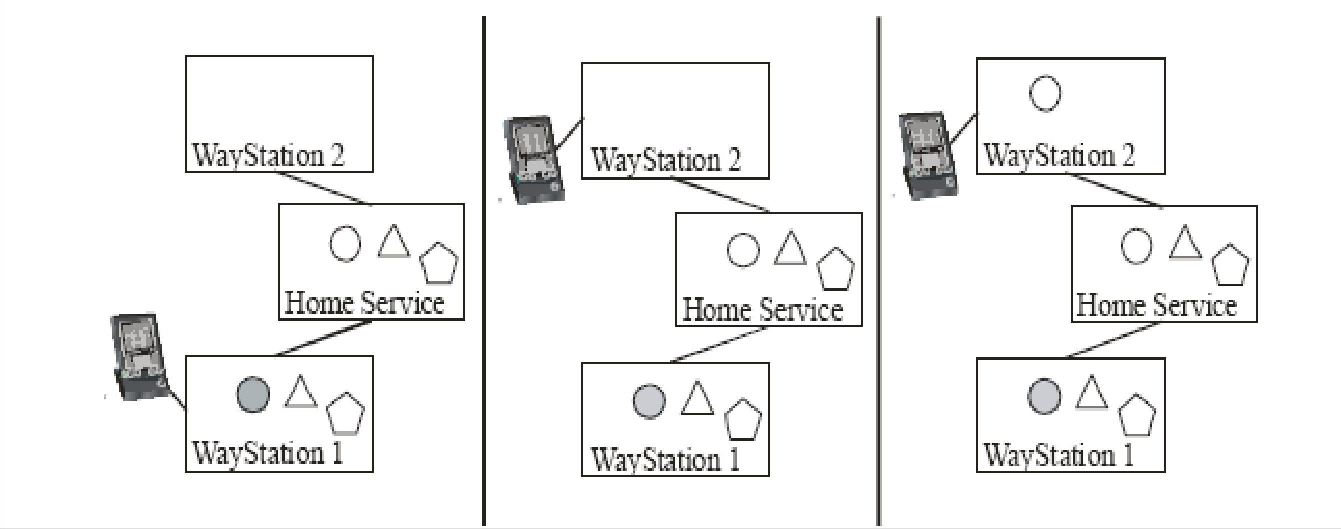
- Pessimistic – interacts with each write
- Optimistic, last-writer – only periodically

Selection of Consistency schema

- Publish-subscribe, mirror: last-writer is fine
- Workloads with very high write locality – optimism
- Workloads with fine-grained write sharing – pessimism.
- It seems that different replicas in the system can have different policies
- Doesn't really tell how conflicts are handled

Destroying and migrating Replicas

- Waystation replica might be destroyed when client is not interested anymore or when clients move away.
- Not interested clients can be handled easily
- Moving clients need client-consistent updates



Current status

- Nice “buzzwords” - no real implementation
- Only played around with estimation filters
- Current testbed consists of a WayStation, client and server to add fluid replication to NFS

Thank you !
Questions ?