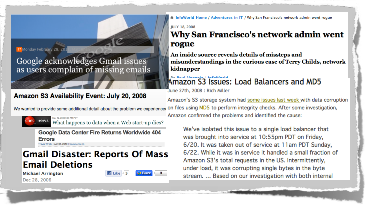
Reasoning with MAD distributed systems

Lorenzo Alvisi
The University of Texas at Austin

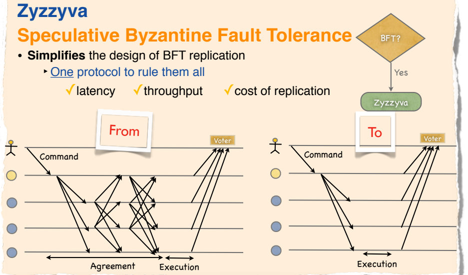
Tolerating arbitrary faults

Bad things do happen to good systems …



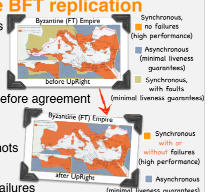
Zyzzyva
Speculative Byzantine Fault Tolerance

- **Simplifies** the design of BFT replication
  - One protocol to rule them all
  - √ latency √ throughput √ cost of replication



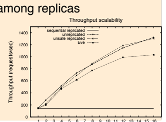
UpRight
Practical and configurable BFT replication

- Applying BFT replication to **real** services
  - BFT HDFS
  - BFT ZooKeeper
- Refining the architecture and APIs
  - Introduce Request Quorum stage before agreement
  - Clean application API for:
    - processing requests
    - taking application state snapshots
- **Configurable** replication
  - u: services are Up (live) despite u failures
  - r: services are Right (safe) despite r commission failures
  - Replication costs expressed as a function of u and r



EVE
Replicating multithreaded servers

- Replication state-of-the-art
  - **Agree** on order of requests, then **execute** them
  - Requires **deterministic** execution
    - Practically this means **single-threaded** execution
- Eve
  - First **execute** requests nondeterministically, without agreeing on order
  - Then **verify** if state and responses match among replicas
  - Efficient rollback on divergence
- Benefits
  - Allows multithreaded execution
  - Up to 12x speedup on a 16-core machine
  - 25% slower than an unreplicated server



MAD Systems

What is a MAD system?

Any system that spans Multiple Administrative Domains (e.g. peer-to-peer services, cloud/outsourced storage, Internet routing, and wireless mesh routing.

What is so special about MAD systems?

Traditional threshold FT does not apply!

- Nodes can be selfish: cooperation requires incentives
- Sybil attacks can overwhelm any threshold mechanism

If this were not enough, each domain is a black box to its peers: what basis is there for trust?

The BAR Model

Three classes of MAD nodes:

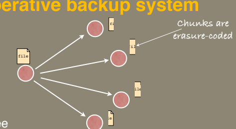
- Byzantine: deviate arbitrarily, for any reason
- Acquiescent: follow the assigned protocol obediently
- Rational: deviate iff doing so increases their utility

BAR Tolerant Systems

- No more than $n/3$ Byzantine nodes
- No bound on number of rational nodes

BAR-B
A BAR-tolerant cooperative backup system

- Peers assign each other *chunks* to store on their behalf
- Assignment need not be symmetric
- Deterministic retrieval guarantee despite Byzantine and Rational peers

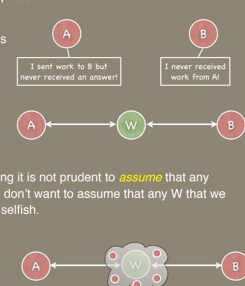


When assigning work, challenge is handling ``he said / she said''

Problem could be solved by interposing an acquiescent witness W between A and B

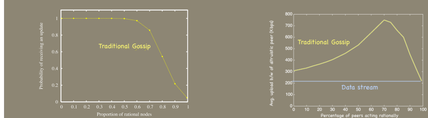
But, just in FT distributed computing it is not prudent to *assume* that any particular node will be correct, we don't want to assume that any W that we may use will not either fail or turn selfish.

Solution: use BAR-tolerant State Machine Replication to build the abstraction of an acquiescent W out of node each of which may be Byzantine or selfish

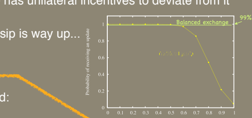


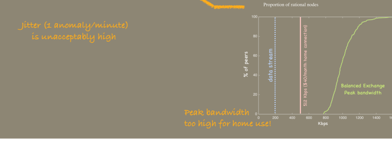
BAR-Gossip
BAR-tolerant Nash for P2P live streaming

- Gossip is attractive infrastructure for P2P live streaming
- But gossip protocols perform poorly if many peers behave selfishly



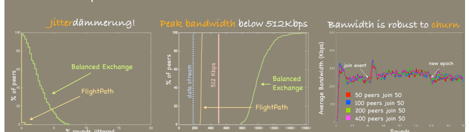
- BAR Gossip relies on Balance Exchange, a provably incentive compatible protocol: no selfish node has unilateral incentives to deviate from it
- Reliability with BAR Gossip is way up...
- But not all news are good:





Flightpath
Approximate equilibria for practical live streaming

- The price for proving BAR Gossip a Nash equilibrium is lack of flexibility:
  - peers cannot join streaming mid-way
  - communication patterns are inflexible
  - extra overhead
- Flightpath balances obedience with choice through approximate equilibria
  - not Nash, but ε-Nash: selfish node deviate only if doing so increases their utility by more than a factor of ε
- Flightpath supports dynamic membership; provides stable performance despite flash crowds; minimizes jitter; and lowers peak bandwidth below home-use threshold



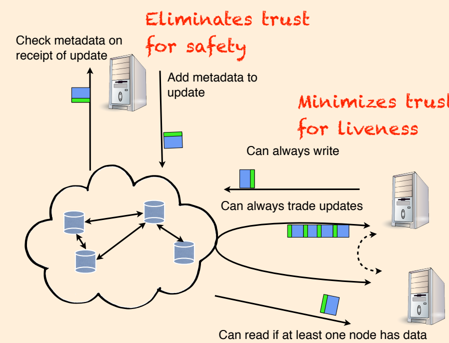
Just in! Local social defenses against Sybil attacks

- Current sybil defenses can distinguish honest from forged identities in social graphs that are fast mixing (equivalently, have constant conductance)
- Alas, many social graphs are not fast mixing!
- We are developing a new approach that provides better protection without relying on global graph properties, such as constant conductance, but rather leverages the social graph's community structure.

Dependable Storage

Depot
Cloud storage with minimal trust

- Removes trust from providers
- Not the same thing as making providers more trustworthy!



Teapot

Minimal trust for today's cloud

- Same guarantees of Depot, but using unmodified Amazon S3 servers

In progress!

Lorenzo Alvisi is a Professor in the Department of Computer Science at UT Austin, where he is a co-director of the Laboratory for Advanced Systems Software (LASR). He holds a Ph.D. and M.S. in Computer Science from Cornell University, and a Laurea summa cum laude in Physics from the University of Bologna, Italy. He is a Fellow of the ACM and the recipient of an Alfred P. Sloan Fellowship and of the NSF CAREER Award.

Collaborators on the efforts noted here include my UT Austin colleagues Mike Dahlin and Mike Walfish; Allen Clement (MPI-SWS); Rama Kotla (MSR Silicon Valley); Alessandro Panconesi (Sapienza University Rome), Silvio Lattanzi (Google); and Edmund Wong, Manos Kapritsos, and Yang Wang, all Ph.D. students at UT.