Smoke and Mirrors: Shadowing Files at a Geographically Remote Location Without Loss of Performance

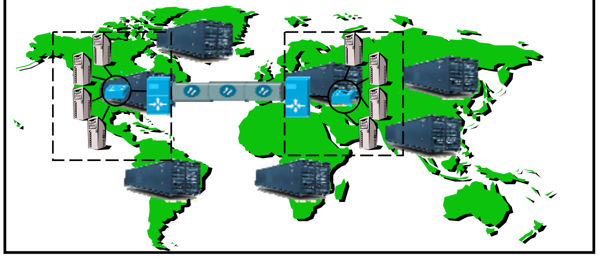
Hakim Weatherspoon, Lakshmi Ganesh, Tudor Marian, Mahesh Balakrishnan, and Ken Birman

Large-Scale Distributed Systems and Middleware (LADIS)

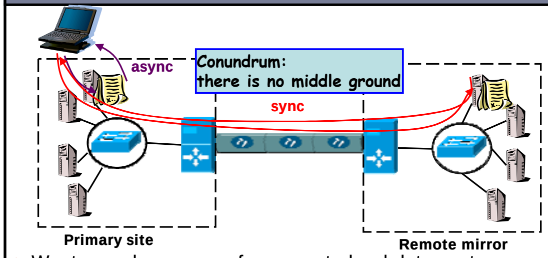
September 15th, 2008

Critical Infrastructure Protection and Compliance

- U.S. Department of Treasury Study
  - Financial Sector vulnerable to significant data loss in disaster
  - Need new technical options
- Risks are real, technology available, Why is problem not solved?

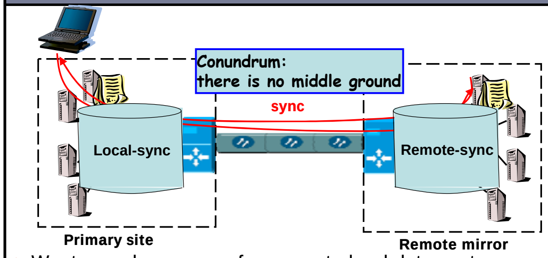


Mirroring and speed of light dilemma…



- Want asynchronous performance to local data center
- *And* want synchronous guarantee

Mirroring and speed of light dilemma…



- Want asynchronous performance to local data center
- *And* want synchronous guarantee

Challenge

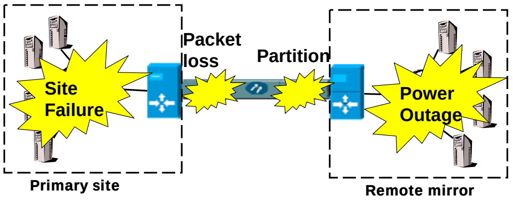
- How can we increase reliability of local-sync protocols?
  - Given many enterprises use local-sync mirroring anyways
- Different levels of local-sync reliability
  - Send update to mirror immediately
  - Delay sending update to mirror – deduplication reduces BW

Talk Outline

- Introduction
- **Enterprise Continuity**
  - How data loss occurs
  - How we prevent it
  - Smoke and mirrors file system
- Evaluation
- Conclusion

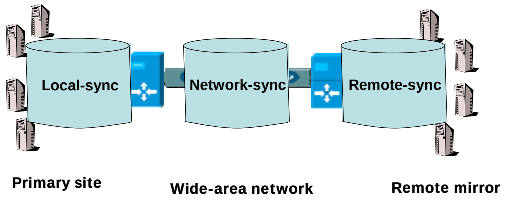
How does loss occur?

- Rather, where do failures occur?

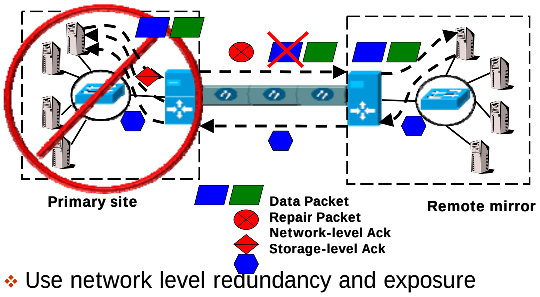


- Rolling disasters

Enterprise Continuity: Network-sync



Enterprise Continuity Middle Ground



- Use network level redundancy and exposure
  - reduces probability data lost due to network failure

Enterprise Continuity Middle Ground

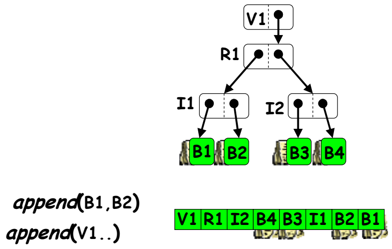
- Network-sync increases data reliability
  - reduces data loss failure modes, can prevent data loss if
  - At the same time primary site fail network drops packet
  - And ensure data not lost in send buffers and local queues
- Data loss can still occur
  - Split second(s) before/after primary site fails…
  - Network partitions
  - Disk controller fails at mirror
  - Power outage at mirror
- Existing mirroring solutions can use network-sync

Smoke and Mirrors File System

- A file system constructed over network-sync
  - Transparently mirrors files over wide-area
  - Embraces concept:

    file is in transit (in the WAN link) but with enough recovery data to ensure that loss rates are as low as for the remote disk case!
  - Group mirroring consistency

Mirroring consistency and Log-Structured File System



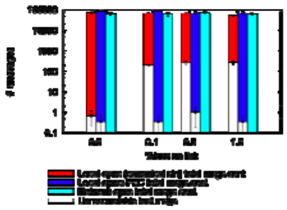
Talk Outline

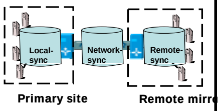
- Introduction
- Enterprise Continuity
- **Evaluation**
- Conclusion

Evaluation

- Demonstrate SMFS performance over Maelstrom
  - In the event of disaster, how much data is lost?
  - What is system and app throughput as link loss increases?
  - How much are the primary and mirror sites allowed to diverge?
- Emulab setup
  - 1 Gbps, 25ms to 100ms link connects two data centers
  - Eight primary and eight mirror storage nodes
  - 64 testers submit 512kB appends to separate logs
    - Each tester submits only one append at a time

Data loss as a result of disaster

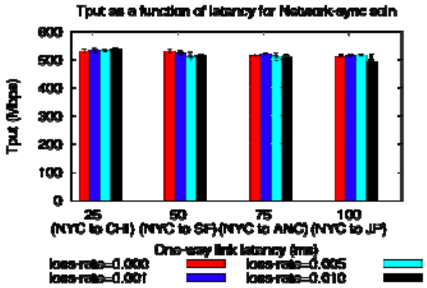




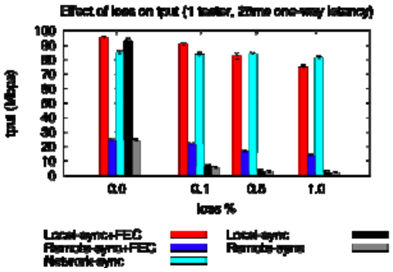
- 50 ms one-way latency
-FEC(r,c) = (8,3)

- Local-sync unable to recover data dropped by network
- Local-sync+FEC lost data not in transit
- Network-sync did *not* lose any data
  - Represents a new tradeoff in design space

High throughput at high latencies

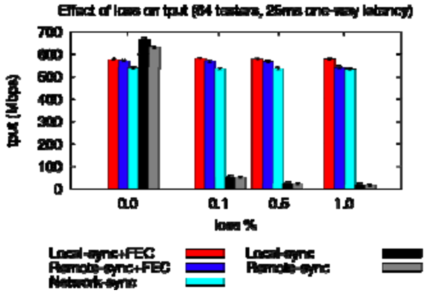


Application Throughput

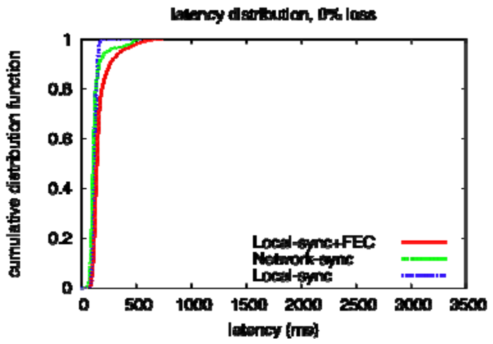


- App throughput measures application perceived performance
- Network and Local-sync+FEC tput significantly greater than Remote-sync(+FEC)

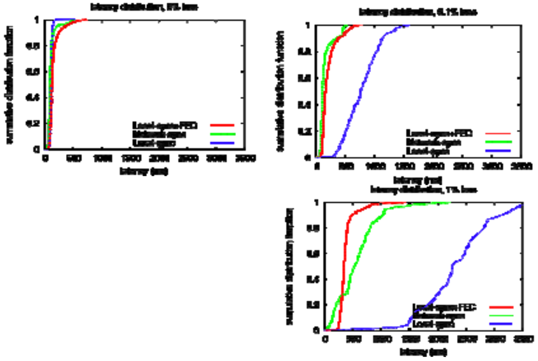
…There is a tradeoff



Latency Distributions



Latency Distributions



Conclusion

- Technology response to critical infrastructure needs
- When does the filesystem return to the application?
  - Fast — return after sending to mirror
  - Safe — return after ACK from mirror
- SMFS — return to user after sending enough FEC
- Network-sync:

LossyNetwork➔LosslessNetwork➔Disk!

- Result: Fast, Safe Mirroring independent of link length!

- Questions?