Continuous Data Protection in iSCSI Storages

Qing (Ken) Yang, Weijun Xiao, Jin Ren

Dept. of ECE
University of Rhode Island
Kingston, RI

Background

- Online data storage doubles every 9 months
- Applications expand to mission critical, financial, and more →1 Hr down time → $millions lost
- IDC:
  - No.1 Top Storage Challenge is…

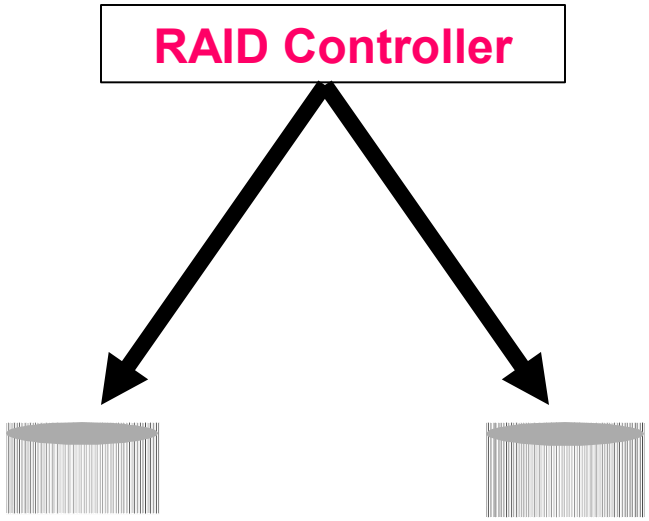
"Improving Data Availability and Recovery"

  - No. 1 Driver of Storage Capacity is...
- "Data Protection and Disaster Recovery"

Qing (Ken) Yang
qyang@ele.uri.edu
UNIVERSITY of Rhode Island

RAID Architecture

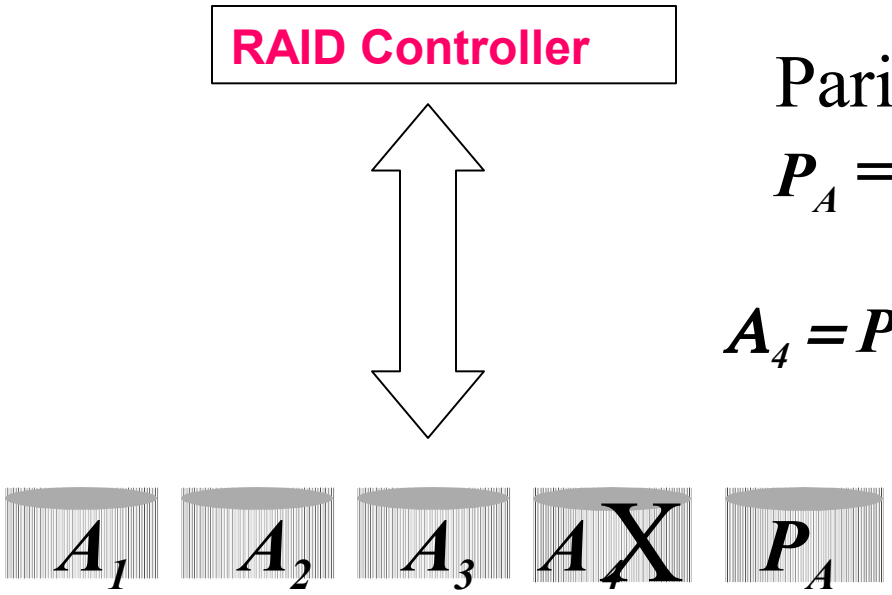
Redundant Array of Independent disks



RAID-1 Mirroring: 2xN Redundancy

RAID 4/5

Redundant Array of Independent disks



Parity:

$$P_A = A_1 \oplus A_2 \oplus A_3 \oplus A_4$$

$$A_4 = P_A \oplus A_1 \oplus A_2 \oplus A_3$$

If data A_4 lost, it can be recovered by using parity P_A, as shown above

What Can a RAID Do For Us?

- Tolerate 1 disk failure, or 2?
  - If a disk fails at time t_0,recover data to t_0
- ? Can we recover data
  - Human errors: 35% of data losses
  - Virus attacks and software corruptions: 25%
  - Disaster and outages: 5%

Qing (Ken) Yang
qyang@ele.uri.edu

Answer: No!

- Current RAID architecture does not deal with these types of data damages

Qing (Ken) Yang
qyang@ele.uri.edu

TRAP (ISCA06 paper)
Timely **R**ecovery to **A**ny **P**oint-in-time

- We Advocate for TRAP architecture
- A new architecture dimension, Time: emphasizing on recovery
  - RPO: Recovery Point Objective
    - How fresh the data can we recover upon failures
  - RTO: Recovery Time Objective
    - How long does it take to recover the data

Qing (Ken) Yang
qyang@ele.uri.edu
UNIVERSITY of Rhode Island

TRAP-1

- Data are backed up at discrete time points
- Examples: snapshots, daily differential backup
- This is commonly done today
- Problems:
  - Some data between backups are lost: large RPO
  - It usually takes a large amount of time to recover: large RTO
  - **T**ime-consuming **R**ecovery to **A**ssigned **P**oint-in-time: TRAP-1

Qing (Ken) Yang
qyang@ele.uri.edu

TRAP-2

- File Versioning:
  - Different versions of a file are kept as changes occur
  - Examples: Cedar, Elephant, CVFS, CVS, TOP-20, Ext2COW, etc.
- File system dependant, Not block level storage
- Enterprises need reliable block level storages

Qing (Ken) Yang
qyang@ele.uri.edu

TRAP-3

- Keep a journal of all changed data blocks
  - Instead of overwriting a block, replace it to another disk
  - Timely recovery to any point-in-time: CDP
- Problem: Huge amount of storage required, even more than RAID-1
- Cost and performance issues caused by prohibitive amount of storage required.

Qing (Ken) Yang
qyang@ele.uri.edu

TRAP-4

- Main Idea: Store Parities Along Time: Similar to RAID-4 storing parity of a stripe
- Architecture shift from TRAP-3 to TRAP-4, similar to architecture shift from RAID-1 to RAID-4
  - Using parities instead of data itself to save storage cost but achieve same level of fault-tolerance (recoverability)
- Advantages:
  - Block level CDP
  - Space optimal, orders of magnitudes savings over TRAP-3
  - True Timely Recovery to Any Point-in-time

Qing (Ken) Yang
qyang@ele.uri.edu
UNIVERSITY of Rhode Island

TRAP-4 Design

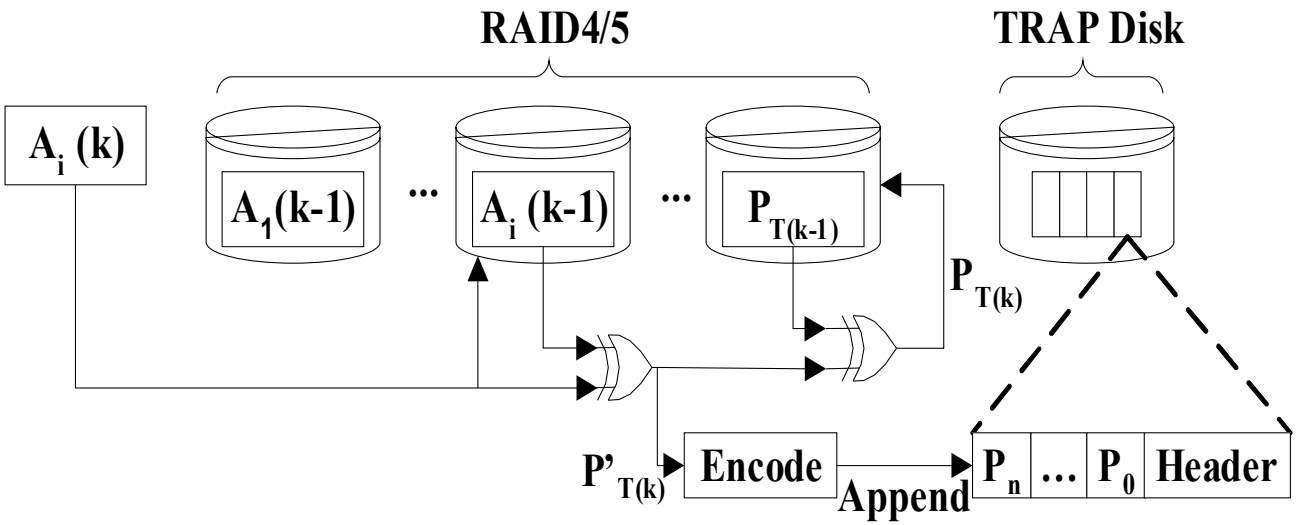


Figure 1. Block Diagram of TRAP-4 Design

Qing (Ken) Yang
qyang@ele.uri.edu

Beta Implementation at iSCSI Layer

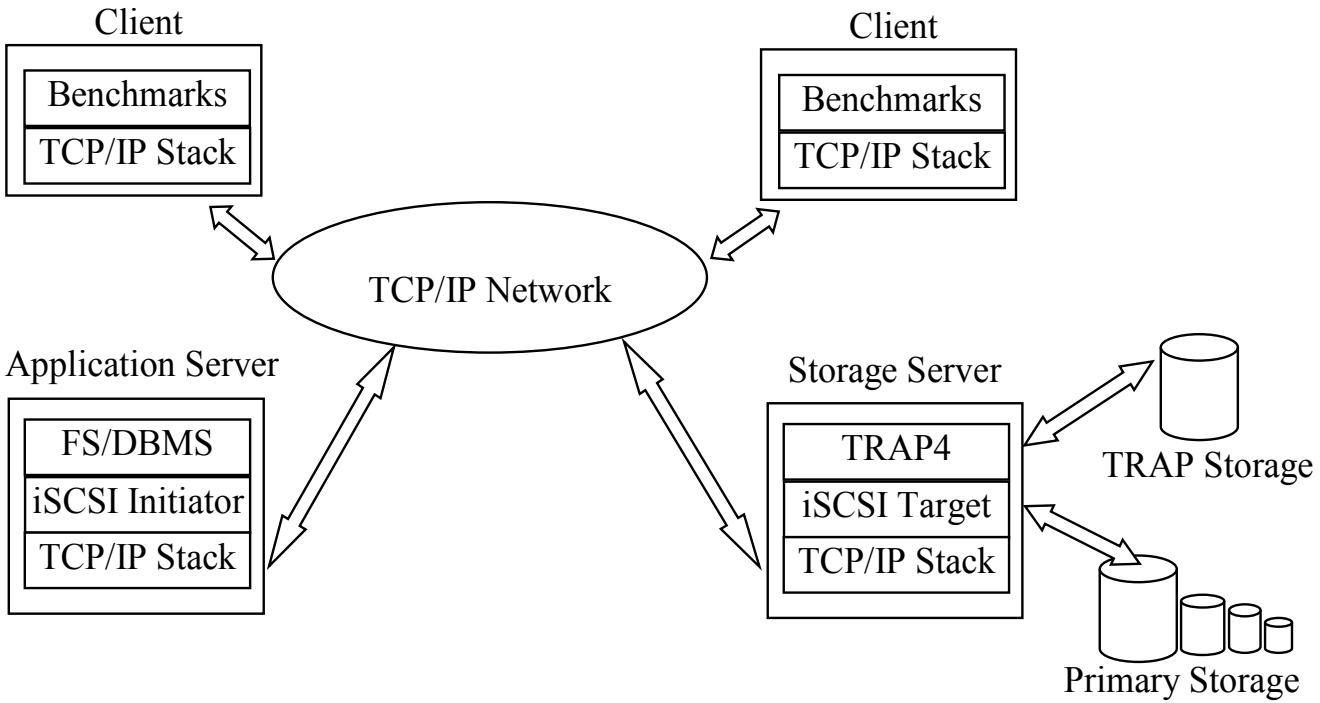


Figure 2. System Architecture of TRAP4 Implementation

Qing (Ken) Yang
qyang@ele.uri.edu

Work-In-Progress

- TRAP is a block level CDP, how to ensure data consistency viewed from a file system and applications?
- Install agents on servers:
  - Pros: agent knows the access behavior of the server making it easy to insert consistent checkpoints
  - Cons: server intrusive, inconvenience, consume server resources, different agents for different platforms, high TCO (total cost of ownership)
- Leveraging existing OS utilities such as Virtual Shadow-Copy Services of Microsoft

Qing (Ken) Yang
qyang@ele.uri.edu

Work-In-Progress

- Inserting check points at storage target independent of servers:
- Analyzing block access activities to look for consistent points
- Two possible ways that we are currently experimenting are
  - Detecting idle time
    - If a sufficient idle time is detected especially after a period of intensive storage activities, we may assume that the host has just finished a transaction or just flushed the cache and in a quiescent state.
    - In this way, we may catch most of consistent points but not guaranteed 100%.
    - It is challenging to determine the proper idle time period to trigger an action of checkpoint. It may become more difficult if the storage is shared and I/O traffic is high.

Qing (Ken) Yang
qyang@ele.uri.edu
UNIVERSITY of Rhode Island

Work-In-Progress

- Analyzing LBAs of block accesses at the storage target

Look for special LBA range of storage writes. Specifically, in NTFS system, a log write is usually performed at the beginning/end of each file change. By inserting checkpoints after each log write, we may catch most of consistency points for recovery purpose. Our preliminary experiments have shown that this approach does show better RTO than traditional CDP recovery approach. Additional experiments are being carried out to validate our approach.

Qing (Ken) Yang
qyang@ele.uri.edu

Life Demo of our TRAP Implementation

1. We have implemented a complete iSCSI target for Windows with the TRAP functions
2. Download the program from www.ele.uri.edu/hpcl
3. Uncompress the file and run TRAP.exe. An iSCSI target starts
4. Start iSCSI initiator and add a new target with the IP address of the machine running TRAP.exe
5. You now have a network drive of 50MB. Any data on this drive is continuously protected.
6. The demo shows that a few files on the new drive were accidentally deleted and changed.
7. TRAP is able to recover the files on the drive as shown.

Qing (Ken) Yang
qyang@ele.uri.edu

Qing (Ken) Yang
qyang@ele.uri.edu