Parity Replication in IP Network Storages

Weijun Xiao, Jin Ren, and
Qing Yang
Dept. of ECE
University of Rhode Island

Motivations

- Performance
  - CPU performance: over 6 orders of magnitude change
  - Memory Performance: several orders of magnitude
  - Network performance: LAN speed: over 4 orders of magnitude
- Cost: Servers:25%; data storage 75% of IT Cost
- Reliability and Availability
  - If CPU Burned: Replace it, re-compute.
  - Memory Lost: Replace with new card, reboot
  - Network Down Fix it, rebuild, comm possible w/ other means
- What about data storage?

Motivations (cont.): Real World Demand

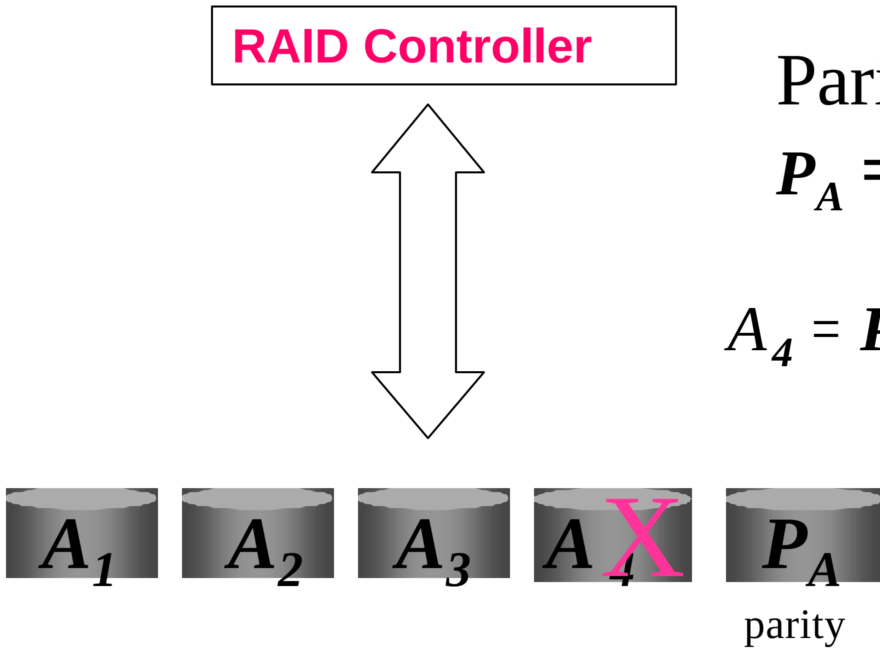
- In 18 months (Jim Gray)
  - New Storage = sum of all old storage (ever)
- Online data storage
  - doubles every 9 months
- Cost of one hour data not available
  - up to millions $
- IDC
  - **#1 Top Challenge…"Improving Data Availability and Recovery"**
  - **#1 Driver of Storage ..."Data Protection and Disaster Recovery"**
  - **#1 Priority of storage users… "Replication"**

The State-of-the-Art Technologies

- File system replication
  - LBFS, rsync, NSI, XOsoft
- Block level replication
  - Synchronous vs Asynchronous
  - Delta blocks and delta set
- WAN bandwidth limitations
  - TCP optimization and data sequencing
  - Data compression before replication

Our Approach

- 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 show above

PRINS: Parity Replication in IP-Network Storages

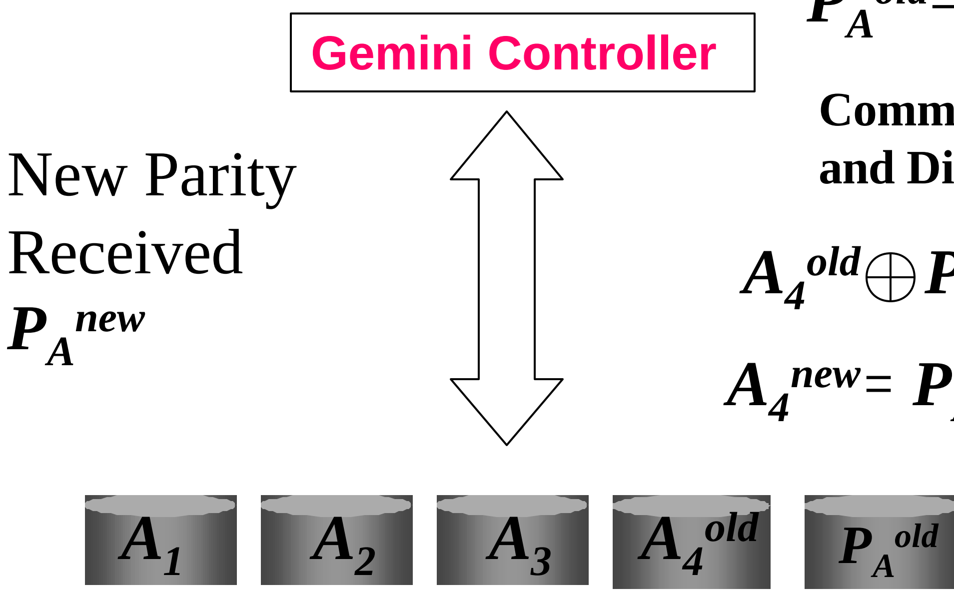
$$P_A^{new} = A_4^{new} \oplus A_1 \oplus A_2 \oplus A_3$$

$$P_A^{old} = A_4^{old} \oplus A_1 \oplus A_2 \oplus A_3$$

Commutativity, Associativity, and Distributivity lead to:

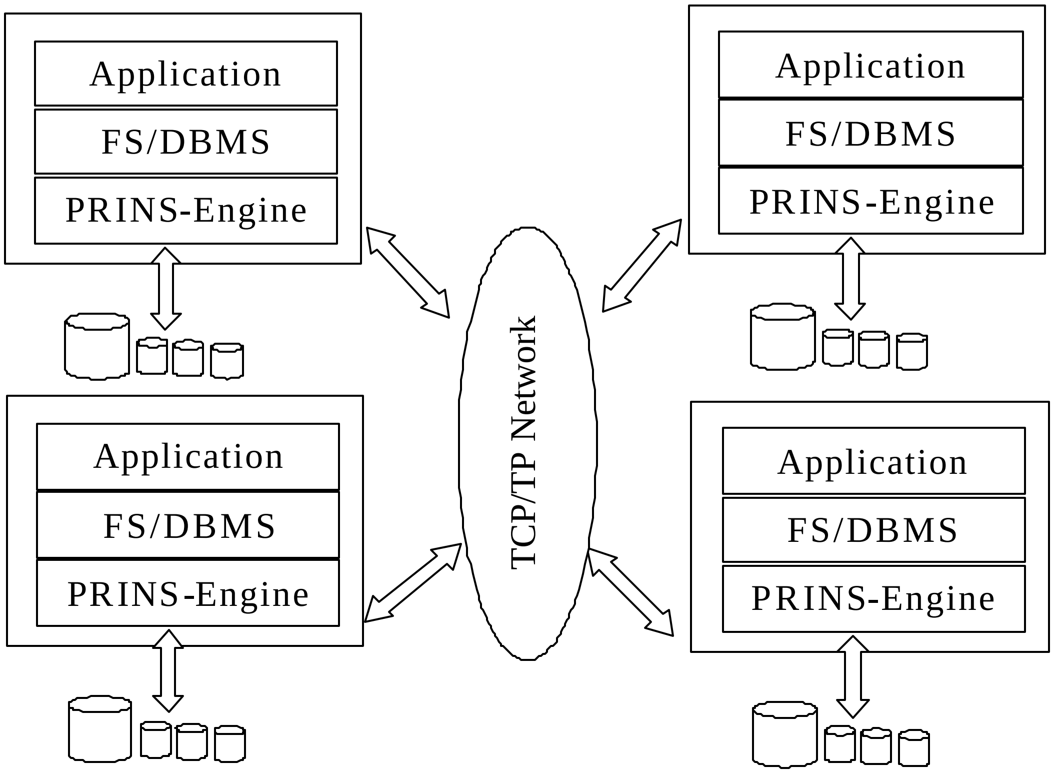
$$A_4^{old} \oplus P_A^{old} = A_1 \oplus A_2 \oplus A_3$$

$$A_4^{new} = P_A^{new} \oplus A_4^{old} \oplus P_A^{old}$$



New data A_4 can be computed using the new parity P_A, old parity, and old data already stored at the storage at mirror site

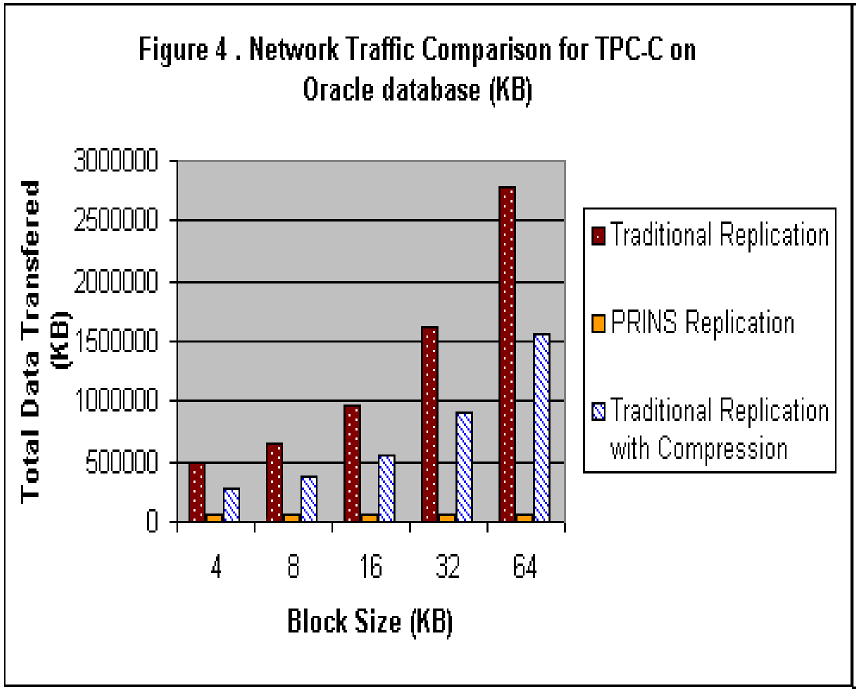
PRINS Design & Implmentation

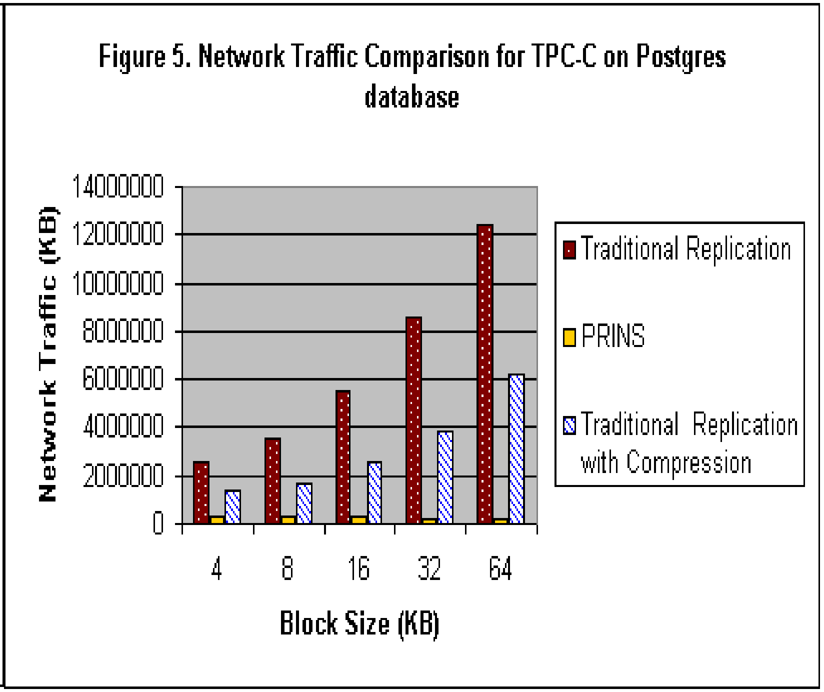


Evaluation Methodology

- Measurement on Real Implementation using iSCSI protocol
- Workloads:
  - TPC-C, TPC-W, on Oracle, Postgres, MySQL Databases
  - File system micro benchmarks on MS and Linux

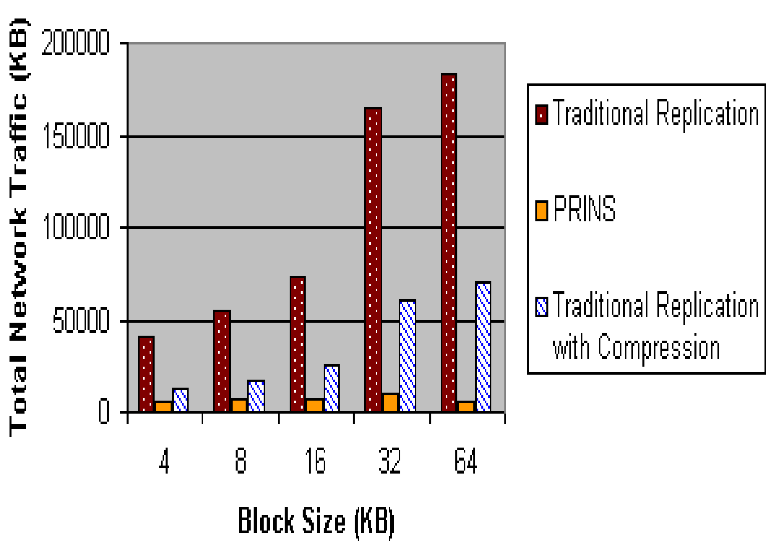
Network Traffic Comparison: TPC-C on Oracle and Postgres Databases





Network Traffic Comparison: TPC-W MySQL and File System Micro-benchmarks

Figure 6. Network Traffic Comparison for TPC-W on MySQL (KB)



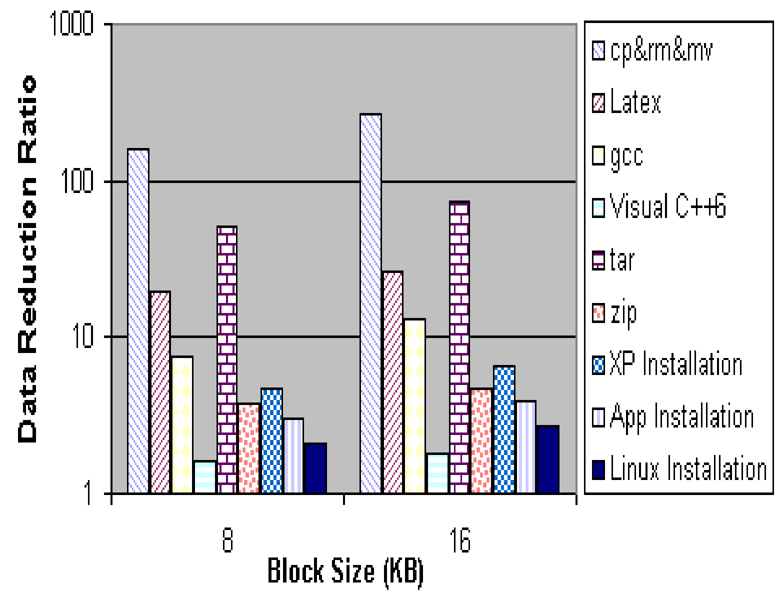


Figure 10. Data Reduction Ratio for Micro Benchmarks

Overhead Evaluation

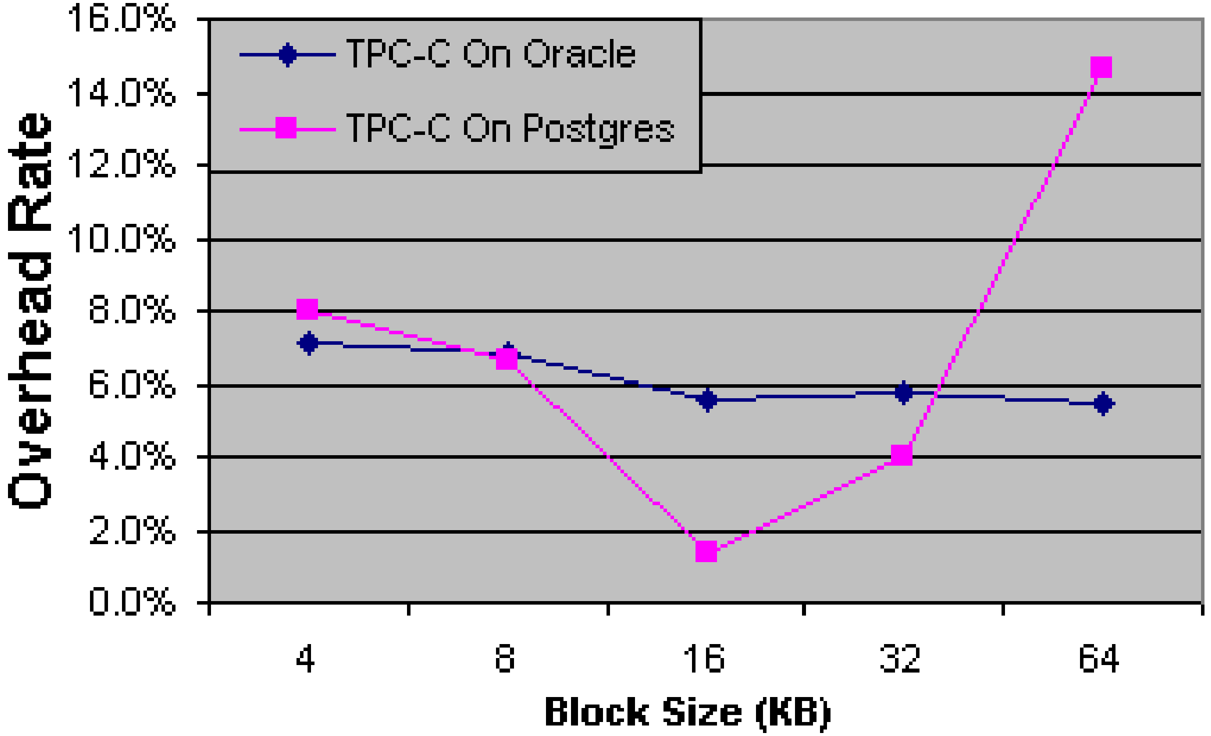


Figure 14. Overhead of PRINS

Conclusions

- A New Data Replication Methodology:
  PRINS
- Prototype Implementation
- Measurements using real world workloads
- 2 orders of magnitudes BW savings