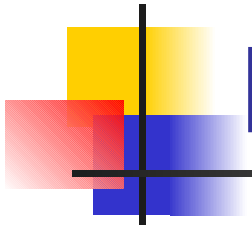
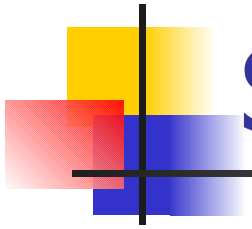
Implementation and Performance Evaluation of Two Snapshot Methods on iSCSI Target Storages

Weijun Xiao, Yinan Liu, and Qing (Ken) Yang
Dept. of Electrical and Computer Engineering
University of Rhode Island, Kingston RI 02881
Jin Ren and Changsheng Xie
Huazhong University of Science and Technology
Wuhan, Hubei, P. R. China



Motivations

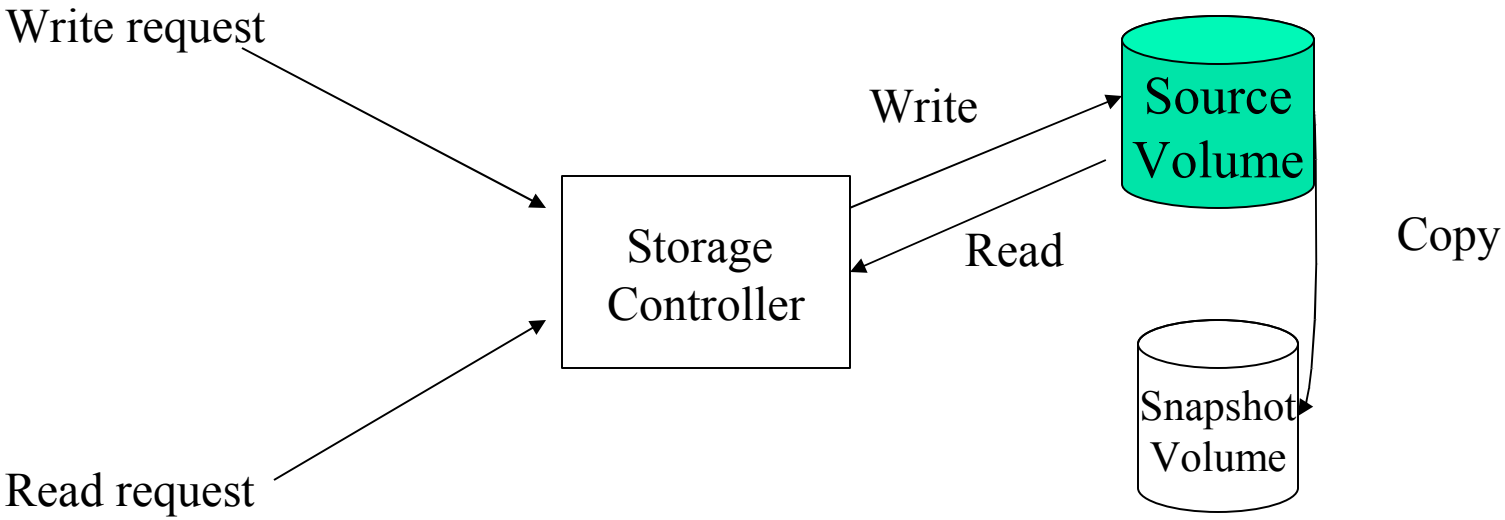
- Data protection and disaster recovery have become critically important
- Snapshots have been commonly used in data storages for backup and data protection
- What is the performance impact of snapshots on applications?



Snapshot

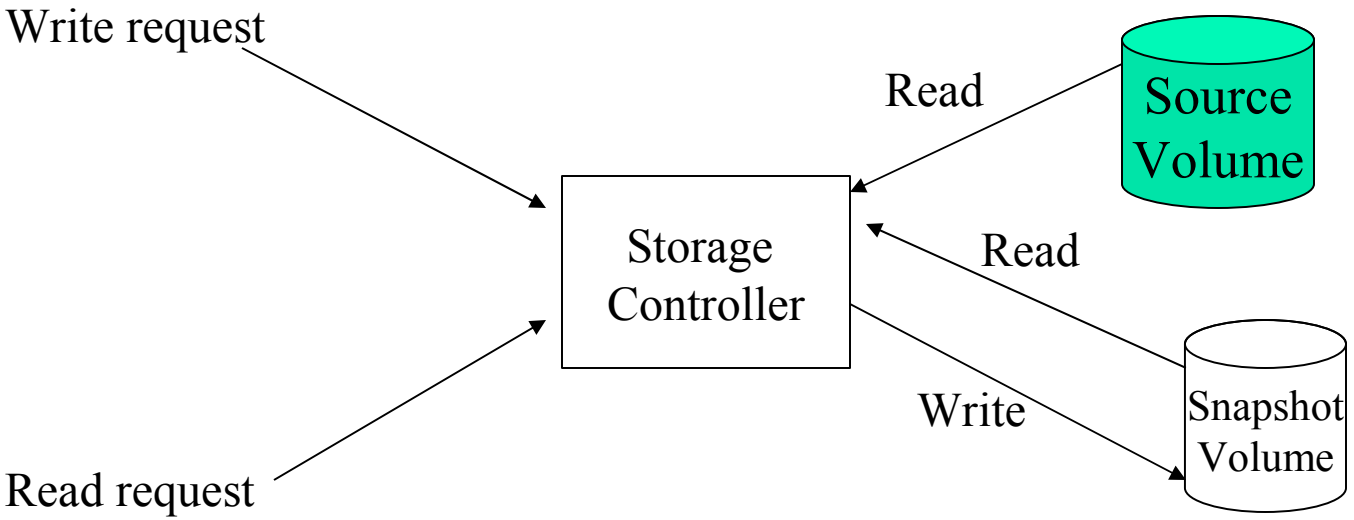
- A point-in-time image of a data storage volume
- Full-copy and differential copy
- Copy-on-write and Redirect-on-write

Copy-on-write



2 write and 1 read for first write request

Redirect-on-write



Redirect write,merge read

System design and implementation

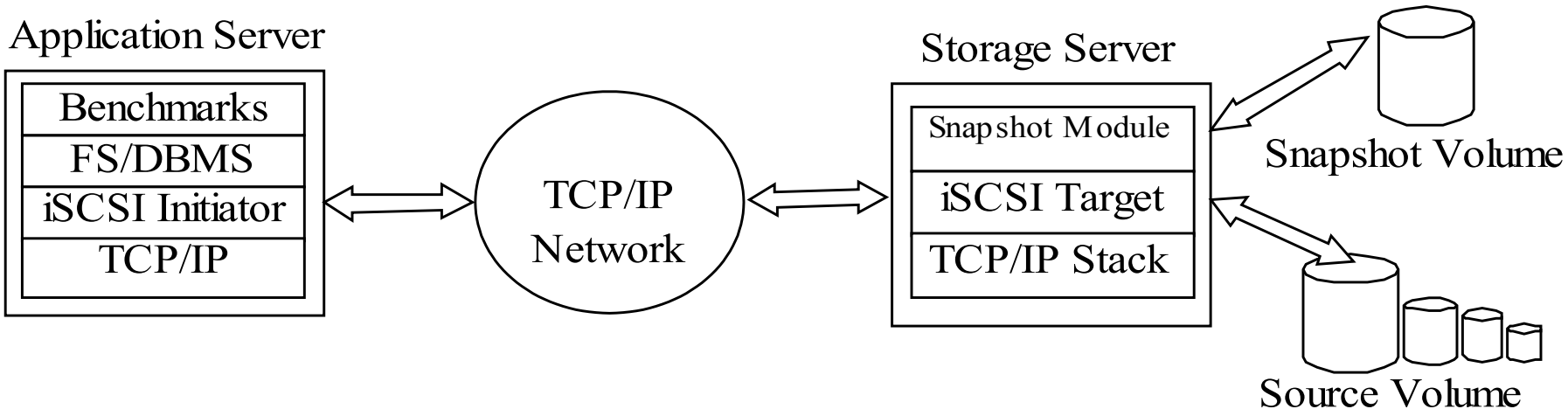
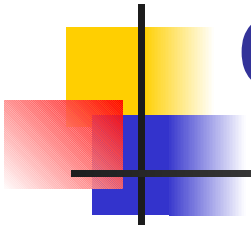


Figure 1. Software Stack of the iSCSI Implementation

iSCSI target: User Interface, basic I/O,disk and volume manager and iSCSI protocol

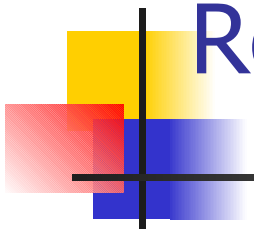
Snapshot module: embedded in the iSCSI target, fixed block size, hash table



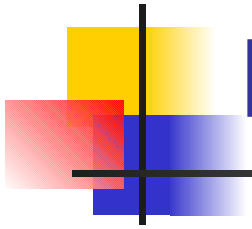
Copy-on-write implementation

- Write request (LBA, I/O size)
  - Alignment of a write request
  - Hash table lookup
  - Copy data blocks to snapshot volume if it is the first write
  - Perform the write I/O to the source volume
- Read request
  - Forward read I/Os directly to the source volume

Redirect-on-write implementation

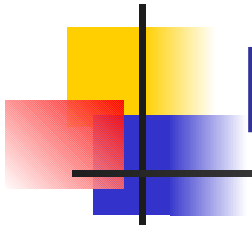


- Write request (LBA, I/O size)
  - Alignment of a write request
  - Hash table lookup
  - write operations are performed on the snapshot volume, no operations on the source volume
- Read request
  - Alignment of a read request
  - Hash table lookup
  - Merge the data from the source and snapshot volumes
- R-o-W makes CDP easy-→TRAP architecture: our ISCA06 paper (Timely Recovery to Any Point-in-time)



Fragmentation and alignment

- I/O request can start from any address
- It is possible to cross snap_block boundaries
- For example, snap_block is 4K, the starting address of a I/O request is 3, the data size is 8K, how to divide this I/O request? 0-3,4-7,8-11
- First fragmented request and last fragmented request may contain partial data
  - Internal fragmentation



Experiment and benchmarks

- Two PCs connected by a LAN switch
- OS: Windows XP professional and Fedora 2
- Databases: Postgres and MySQL
- File systems: NTFS and Ext3
- Benchmarks: TPC-C, TPC-W, IoMeter, PostMark

Numerical results (1)

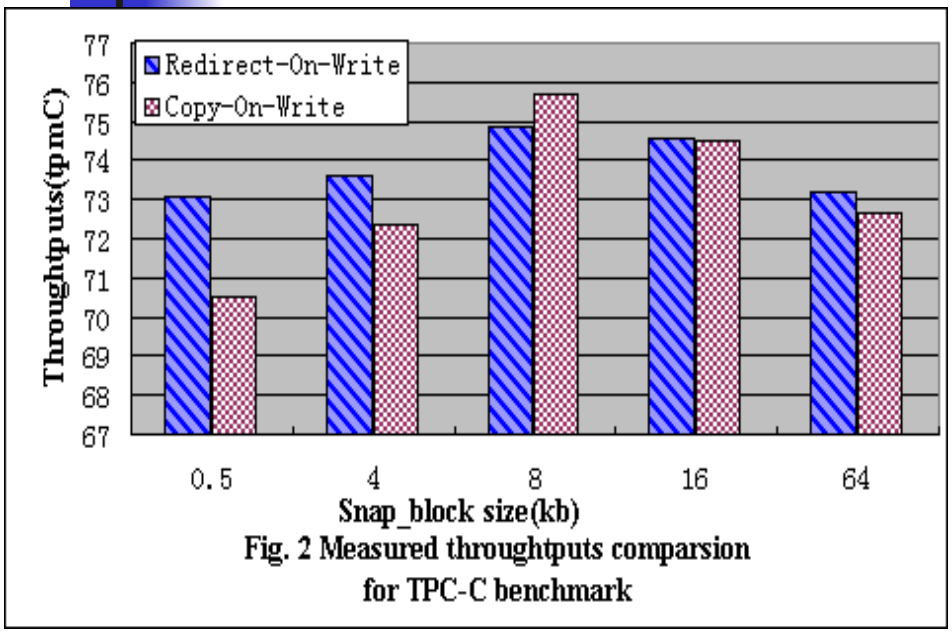


Fig. 2 Measured throughtputs comparsion for TPC-C benchmark

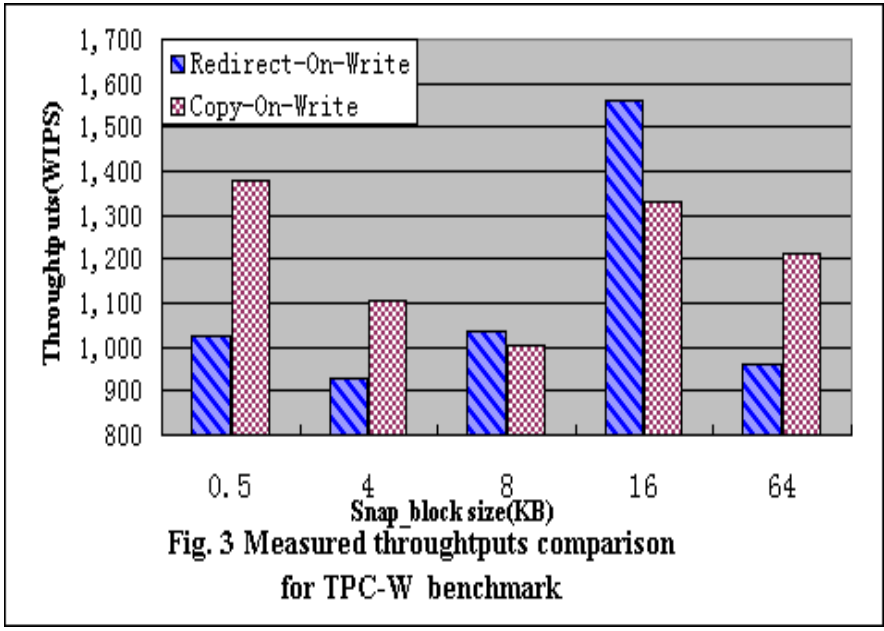


Fig. 3 Measured throughtputs comparison for TPC-W benchmark

Two snapshot techniques perform quite differently

Copy-on-write works well for read-intensive applications

Redirect-on-write works well for write-intensive applications

Numerical results (2)

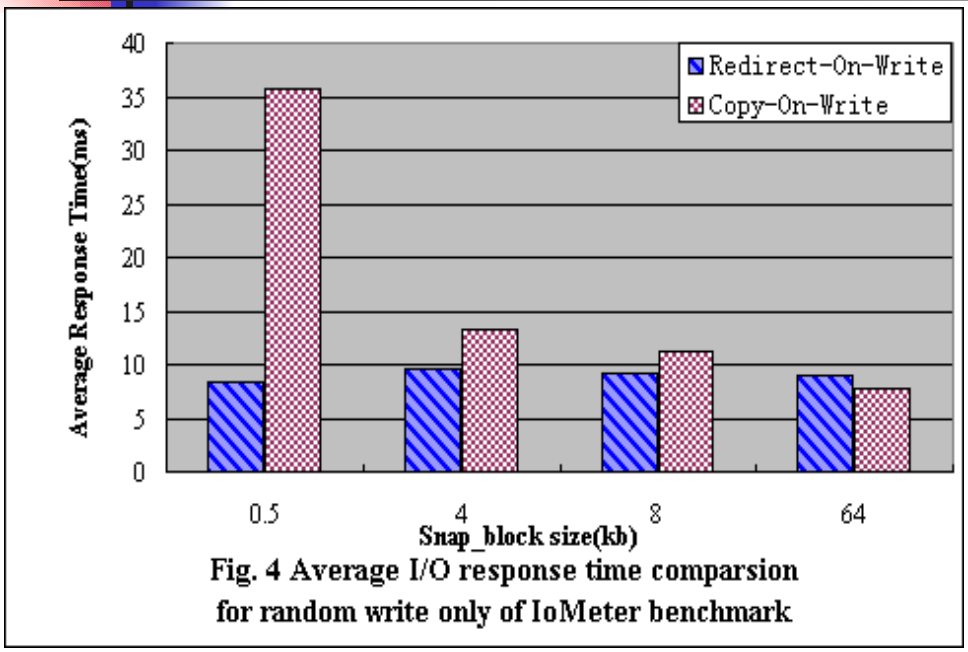


Fig. 4 Average I/O response time comparsion for random write only of IoMeter benchmark

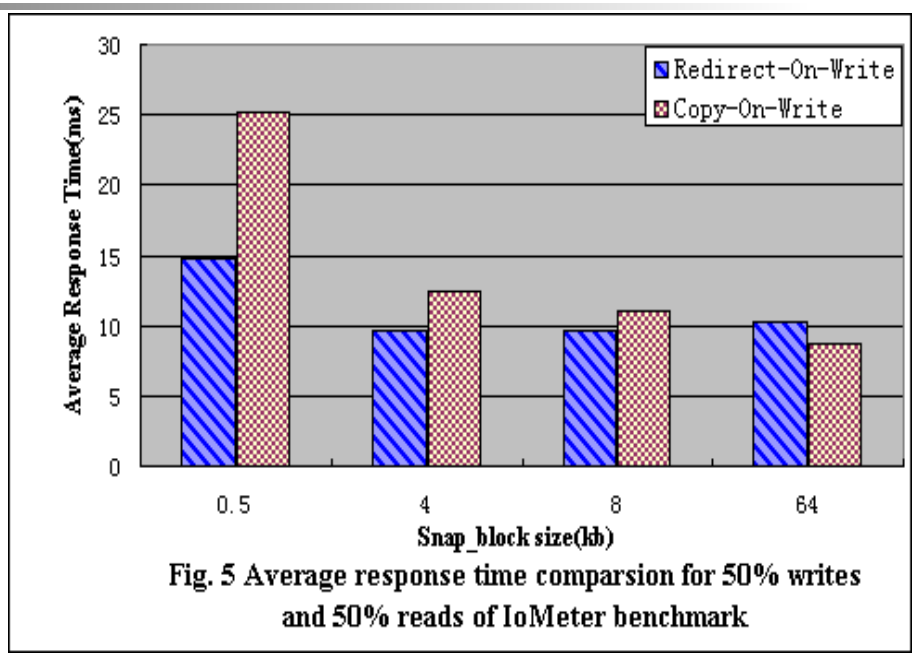


Fig. 5 Average response time comparsion for 50% writes and 50% reads of IoMeter benchmark

Similar performance characteristics with TPC-C

Smaller differences in figure 5 because of read operations

As the snap_block increases, the performance difference reduces because of the penalties of internal fragmentation and LBA alignment

Numerical results (3)

Snap_block size	WriteTime(ms)	ReadTime(ms)
64K	8.328	1.906
16K	8.359	2.344
8K	8.484	2.593
4K	12.516	3.594
0.5K	39.562	10.219

- larger block sizes take shorter time to write than smaller block sizes. However, the time differences for the block sizes of 8KB, 16KB, and 64KB are not significant.

Numerical results (4)

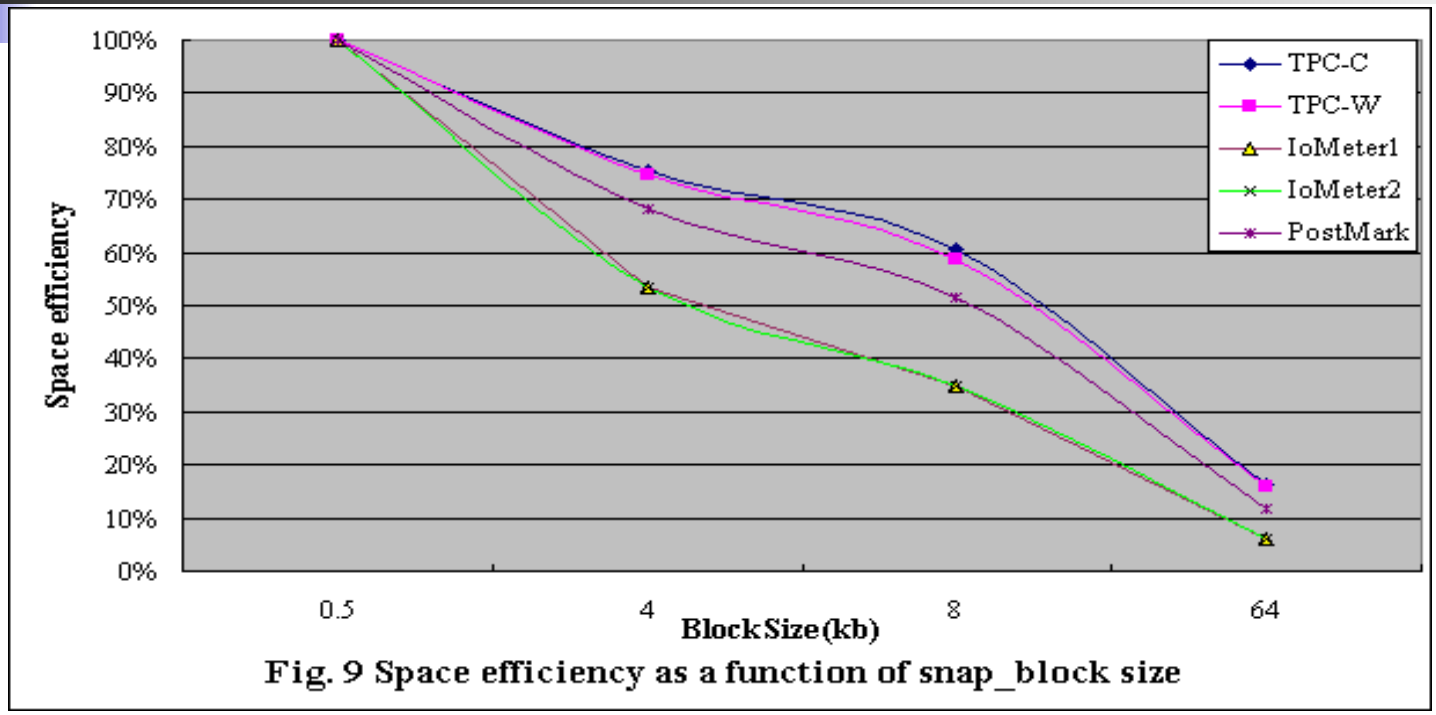
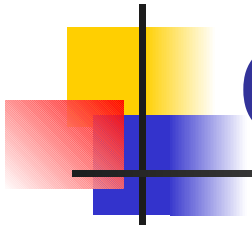


Fig. 9 Space efficiency as a function of snap_block size

- Space efficiency reduces with the increase of the snap_block size
- Two contradicting objectives: increasing block size for better performance and decreasing block size for space efficiency



Conclusions

- Implementation and performance evaluation of two differential snapshot methods
- A working iSCSI target for Windows with snapshots and CDP functionality
- Many important performance characteristics were uncovered through extensive experiments