UNIVERSITY OF MINNESOTA

DIGITAL TECHNOLGY CENTER

An Efficient Data Sharing Scheme for iSCSI-Based File Systems

Dingshan He and David H.C. Du

Department of Computer Science and Engineering
DTC Intelligent Storage Consortium
University of Minnesota
{he,du}@cs.umn.edu

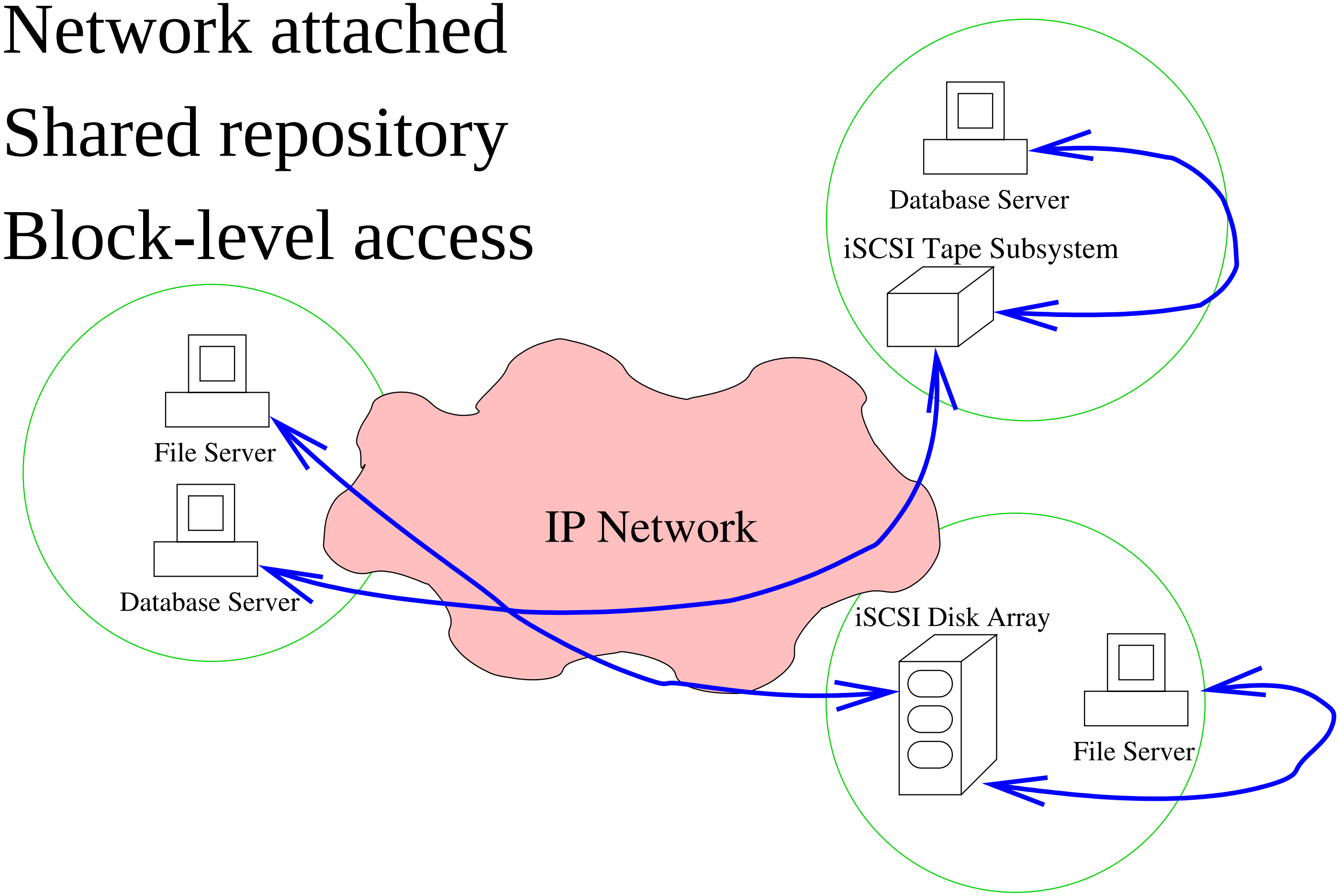
NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on
Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004

NASA

IEEE
COMPUTER
SOCIETY

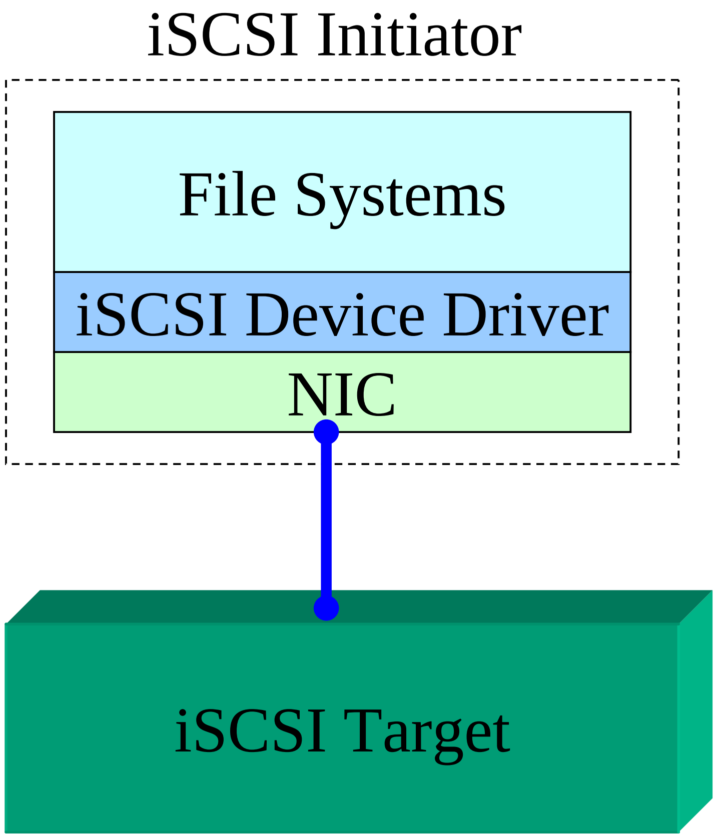
iSCSI-based Storage Systems

- Network attached
- Shared repository
- Block-level access



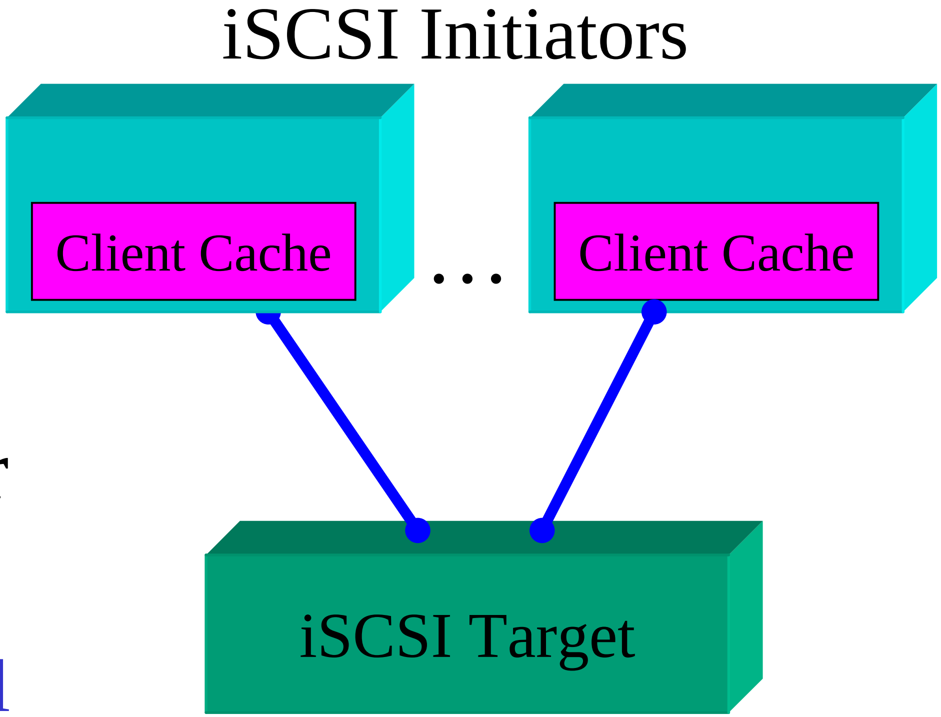
iSCSI-Based File Systems

- FS is unaware of sharing storage
- iSCSI target read/write physical blocks dumbly
- Network connection is over WAN. Therefore, client caching is a must



Data Sharing Conflicts

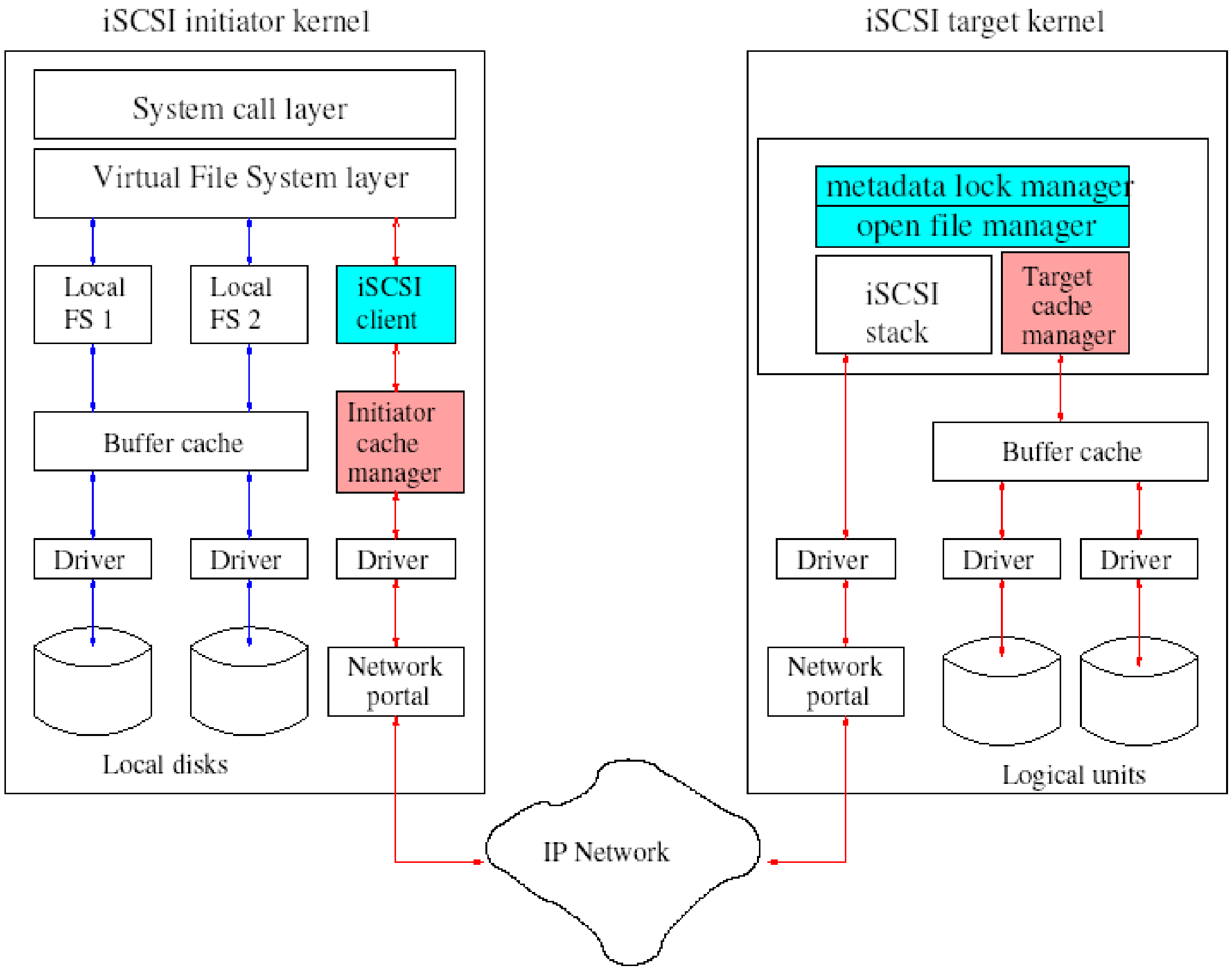
- Client cache may conflict with data on target
- Client cache may conflict with other client caches
- Concurrency Control is a must



Our Contributions

- Locking mechanism for concurrency control
  - Separate metadata and data locking mechanisms
  - Metadata: Semi-preemptible Sharing Locking
  - File Data: Hierarchical Locking
- Callback based mechanism for client cache consistency
- Transaction file sharing semantics to support transaction applications

Architecture Overview



Locks on Metadata Object

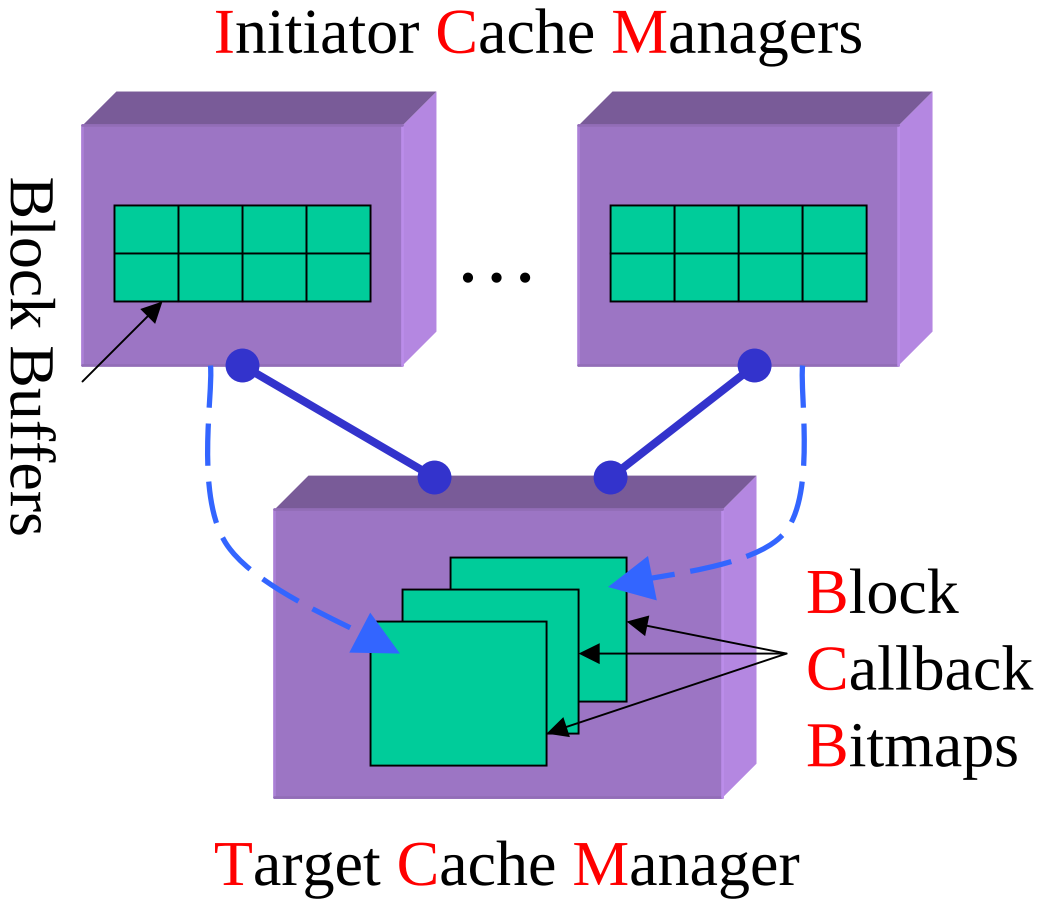
- Roselli et al. found the percentage of metadata reads >> metadata writes
- Shared lock can be cached at initiator
- Exclusive lock request invalidates cached shared locks
- Exclusive lock granted after all invalidation responses received

Locks on File Data

- Locking granularity is a design tradeoff
  - Fine granularity: high concurrency, but high overhead
  - Coarse granularity: Low overhead, but low concurrency
- Hierarchical locking balances between concurrency and overhead

Client Cache Consistency

- TCM maintain one BCB for each ICM
- Block read sets bitmap
- Block write causes callback



Transaction File Sharing Semantics

- Several operations are grouped as a transaction
- Locks are held throughout a transaction
- Deadlock could happen
- Rollback is supported

Thank you!

File System Objects

- Metadata objects
  - – Directory file – i-node + directory data blocks
  - – Normal file – i-node + indirect blocks
  - – Super block
  - – I-node bitmap blocks
  - – Data-block bitmap blocks
- Normal data
  - – Data blocks

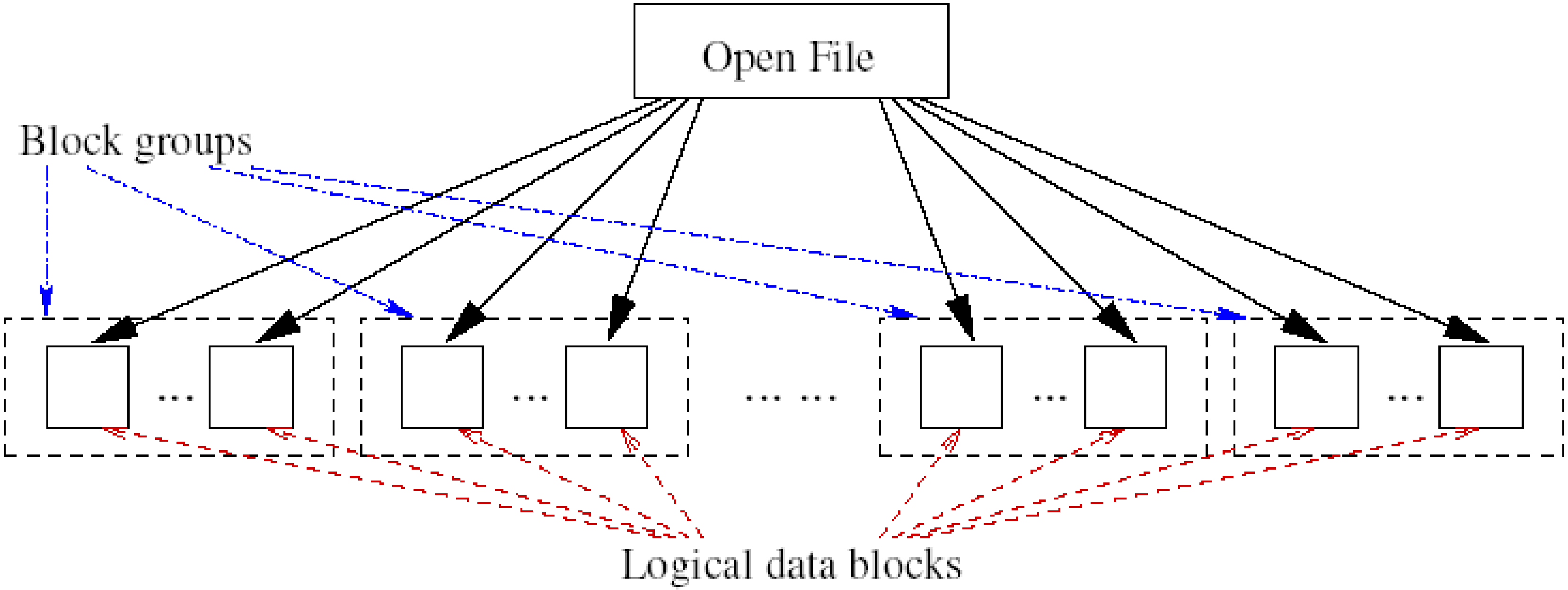
Semi-preemptible Shared Lock

- M_S: gives share access to the requested object
- M_X: gives exclusive access to the requested object
- Semi-preemptible Shared Locking
  - Caching of M_S lock
  - Request M_X lock each time, and release after

	M_S	M_X
M_S		*
M_X	*	*

* conflict

Two-Tier File Data Organization



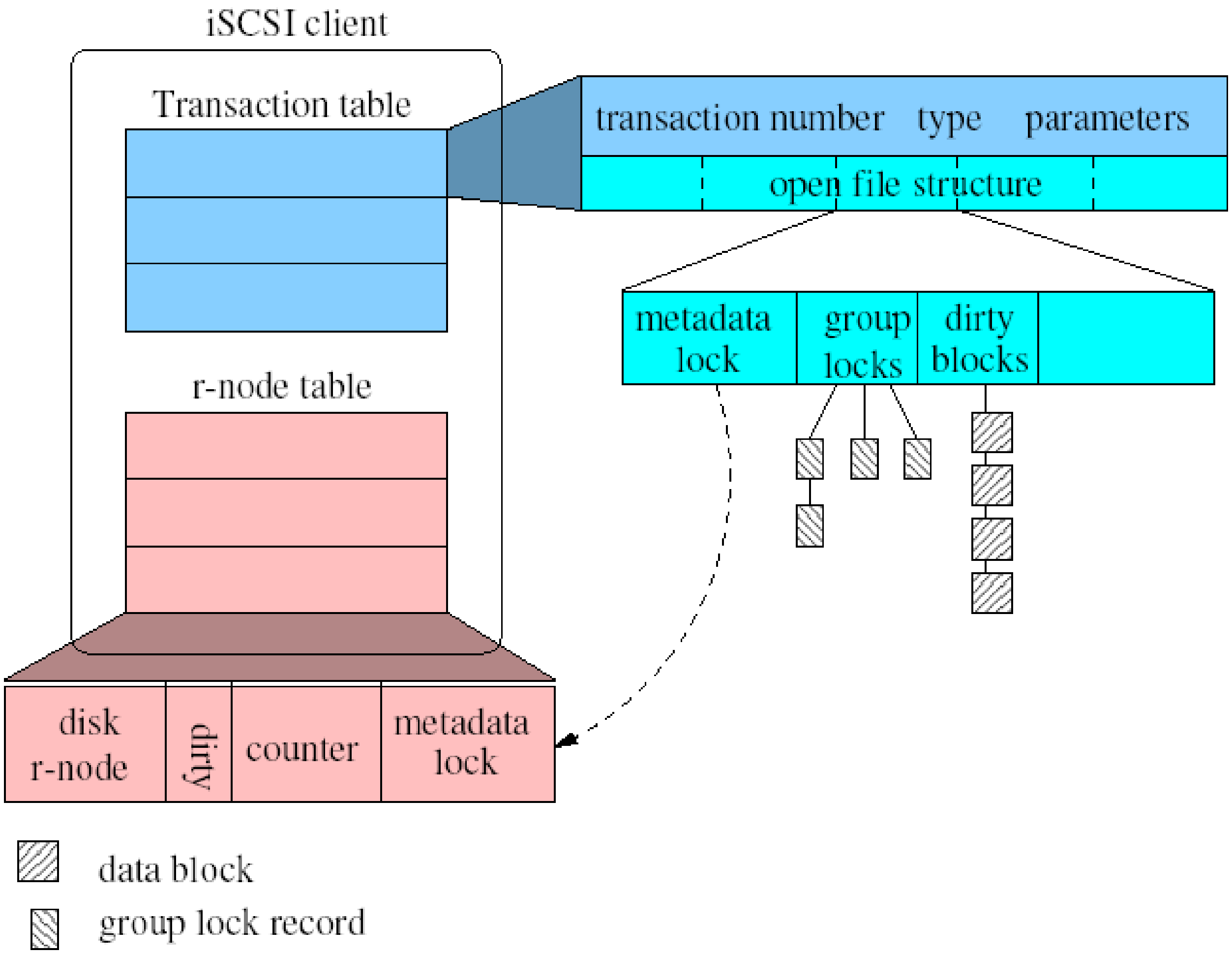
Hierarchical Locks on File Data

- Intention Locks (D_IS,D_IX) are only used on file level
- D_S and D_X can be used on both levels
- Open operation requests a lock (D_IS, D_IX, D_S, D_X) for the whole file
- Read/write operations on a specific logical block requests D_S/D_X locks on the block group

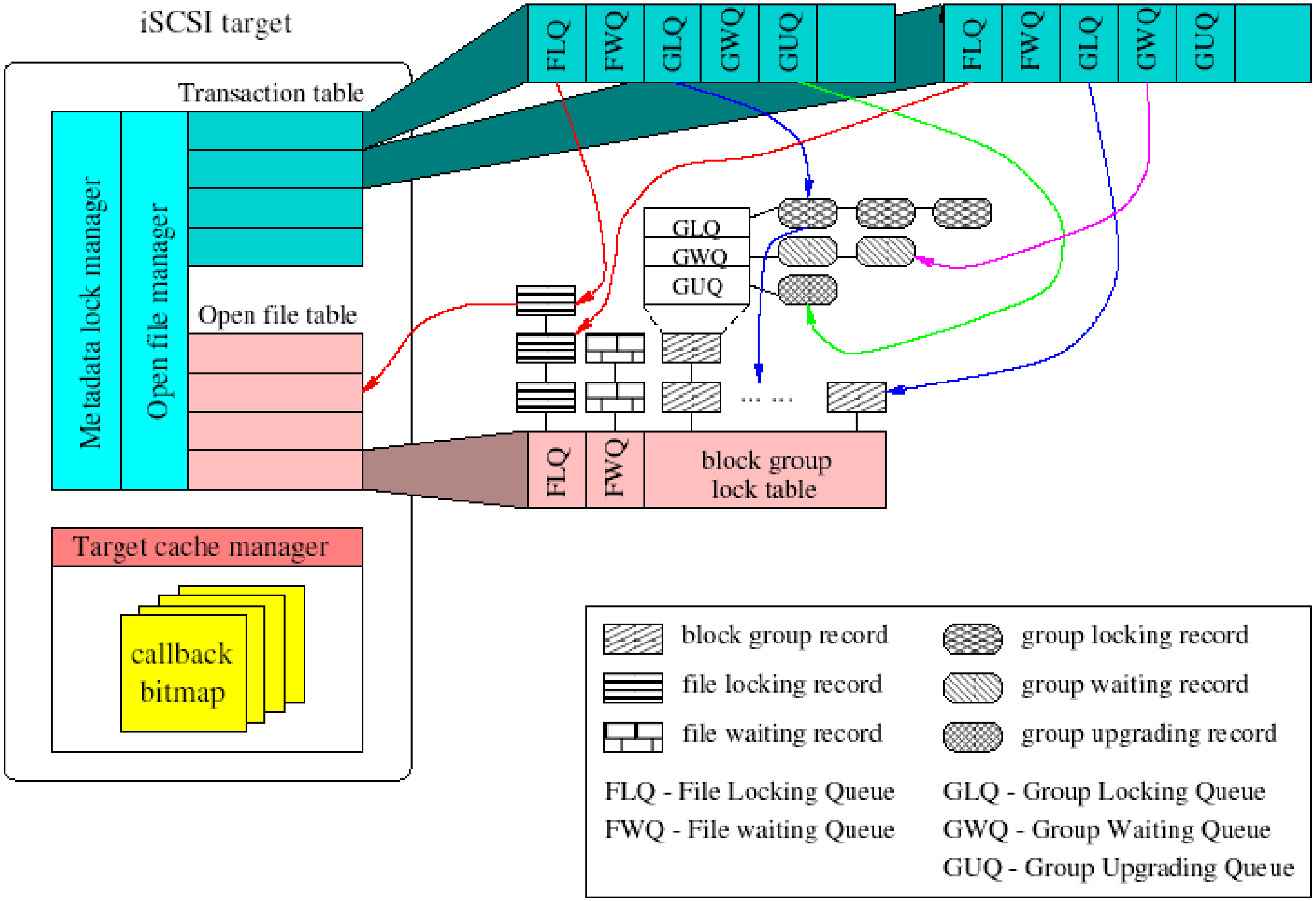
	D_S	D_X	D_IS	D_IX
D_S		*		*
D_X	*	*	*	*
D_IS		*		
D_IX	*	*		

* conflict

Inside iSCSI Client

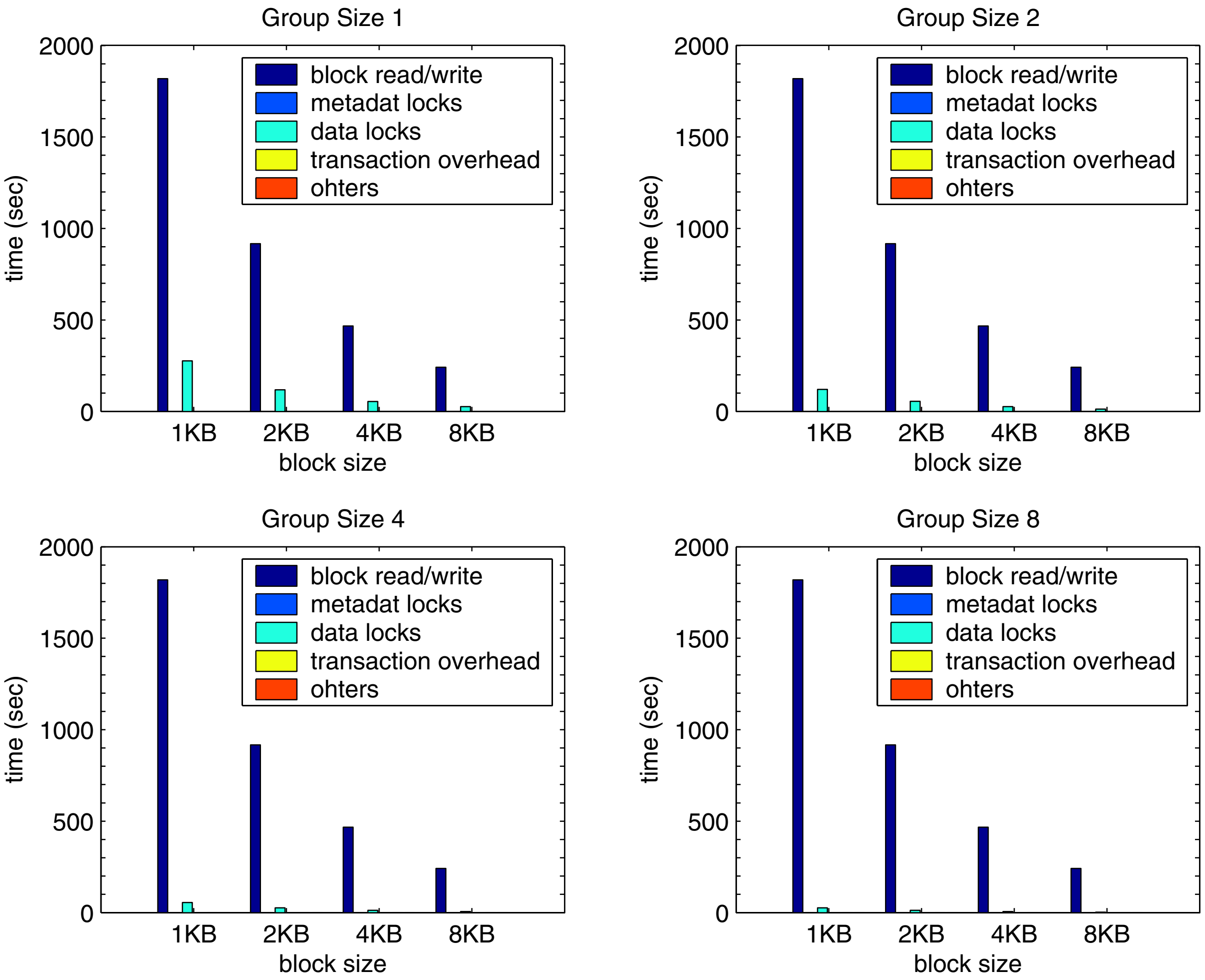


Inside iSCSI Target



Scheme Overhead

Bandwidth = 100Mbps, Latency = 1ms


Group Size 1
Group Size 2
Group Size 4
Group Size 8
time (sec)
1KB
2KB
4KB
8KB
block size
block read/write
metadat locks
data locks
transaction overhead
ohters