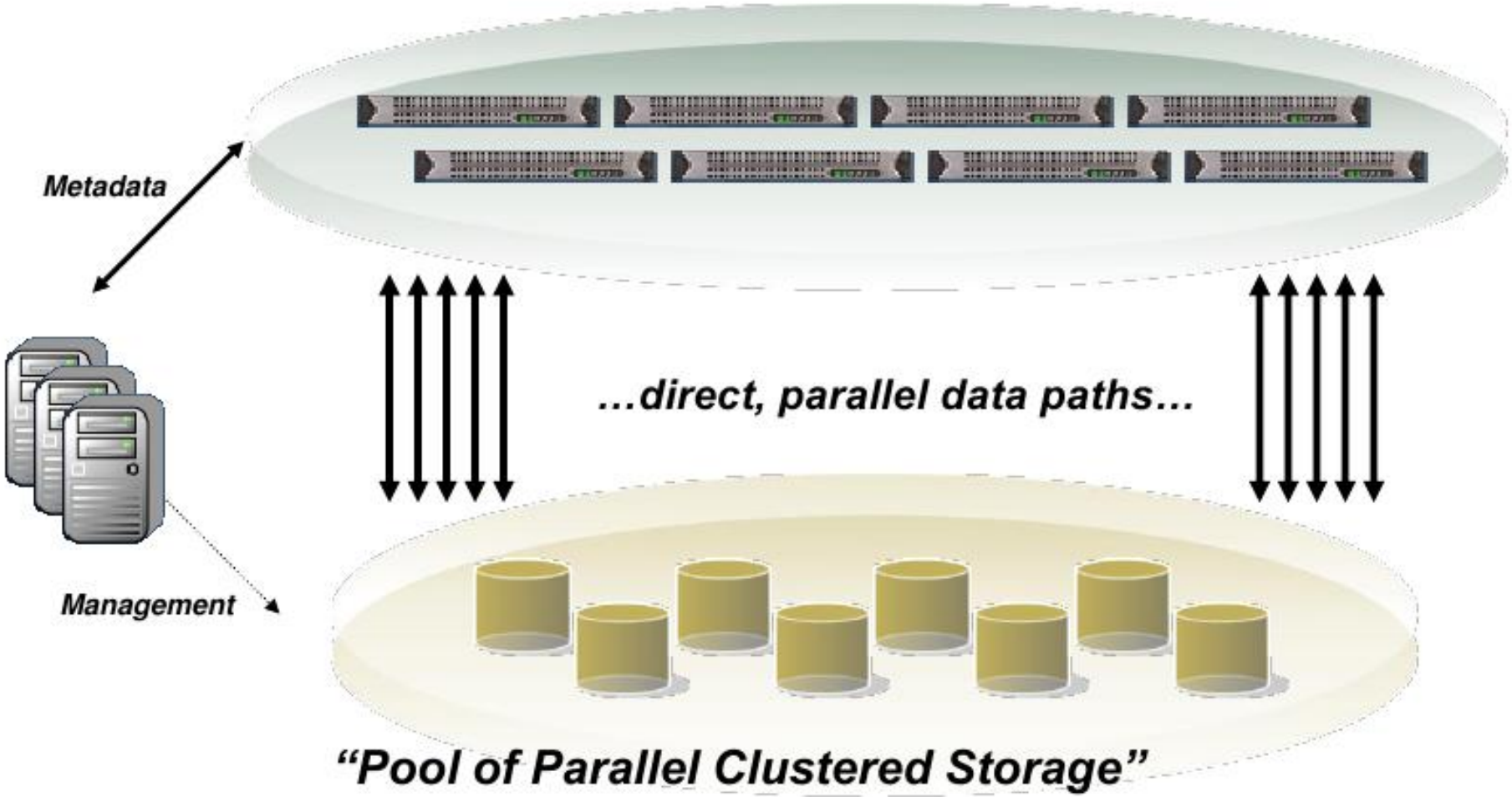
Overcoming write penalties of conflicting client operations in distributed storage systems

26.10.2012

Outline

pNFS

pNFS architecture (theory)



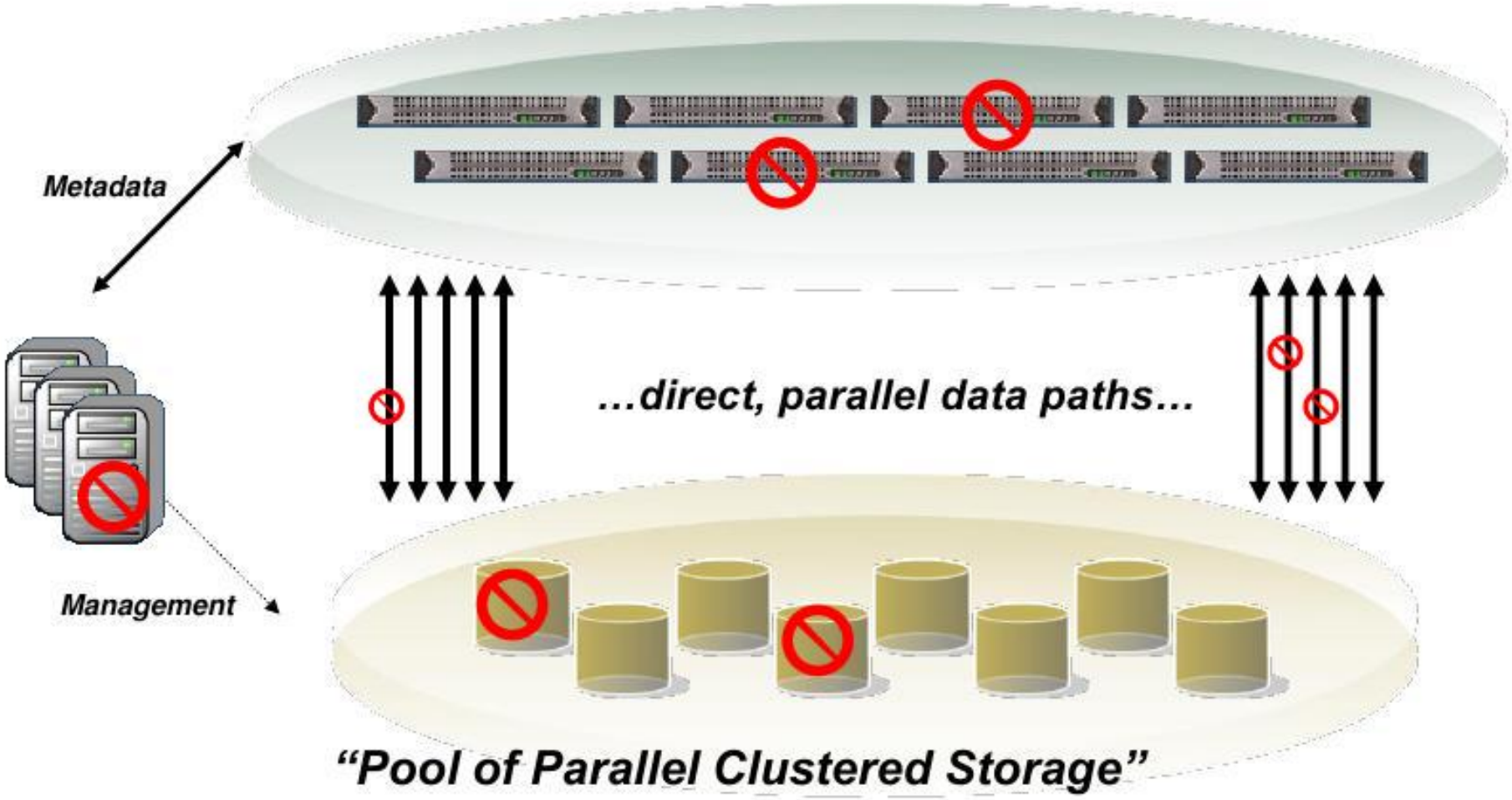
pNFS

Advantages over single server NFS

- performance: due to parallel access to data server
- extensibility: file system with unlimited capacity (theory)
- efficiency: fast metadata operations
- maintainability: real global name space

pNFS

pNFS architecture (reality)



pNFS

Parallel file systems issues

Reliability

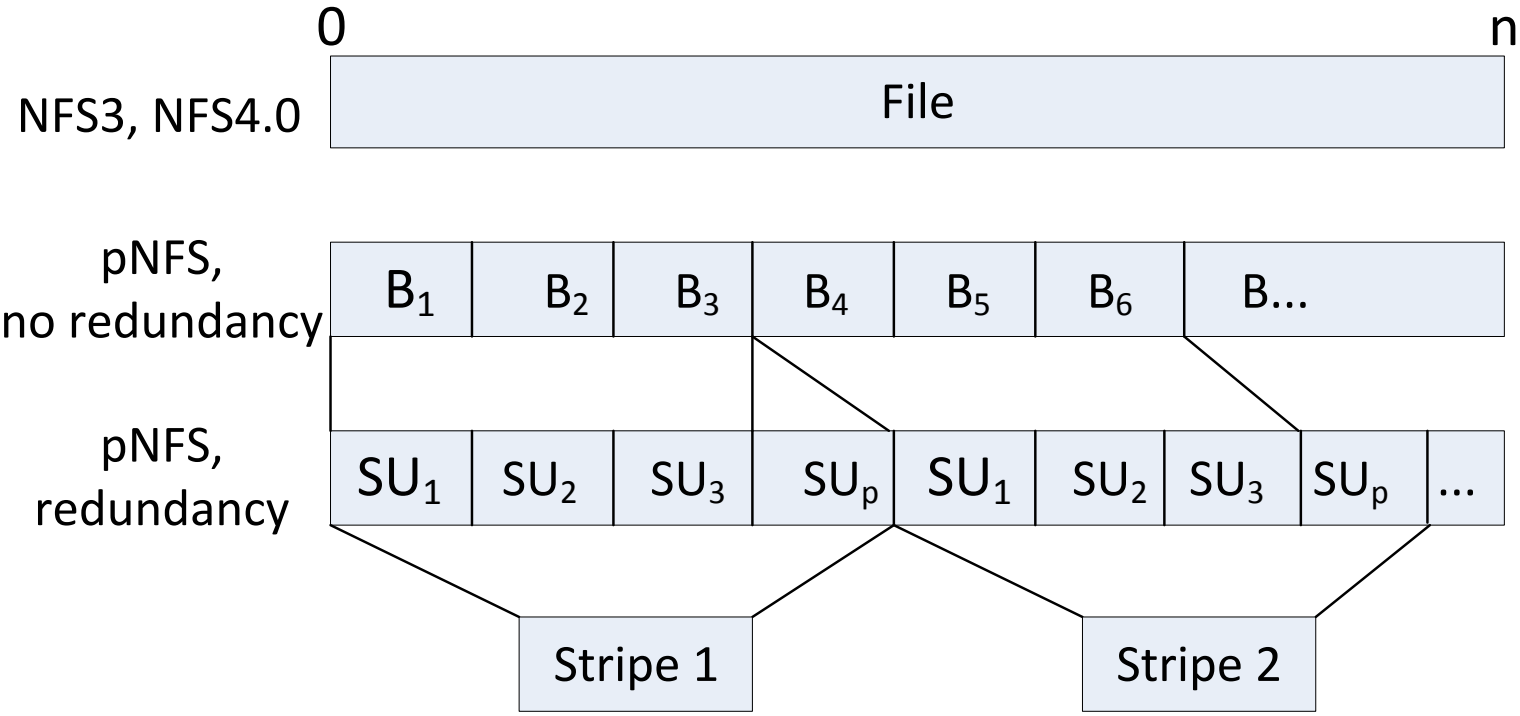
- more devices cause more failures
- redundancy needed

Correctness

- atomicity of operations
- consistency of redundant data
- access management of client requests

pNFS

Redundancy - single failure protection



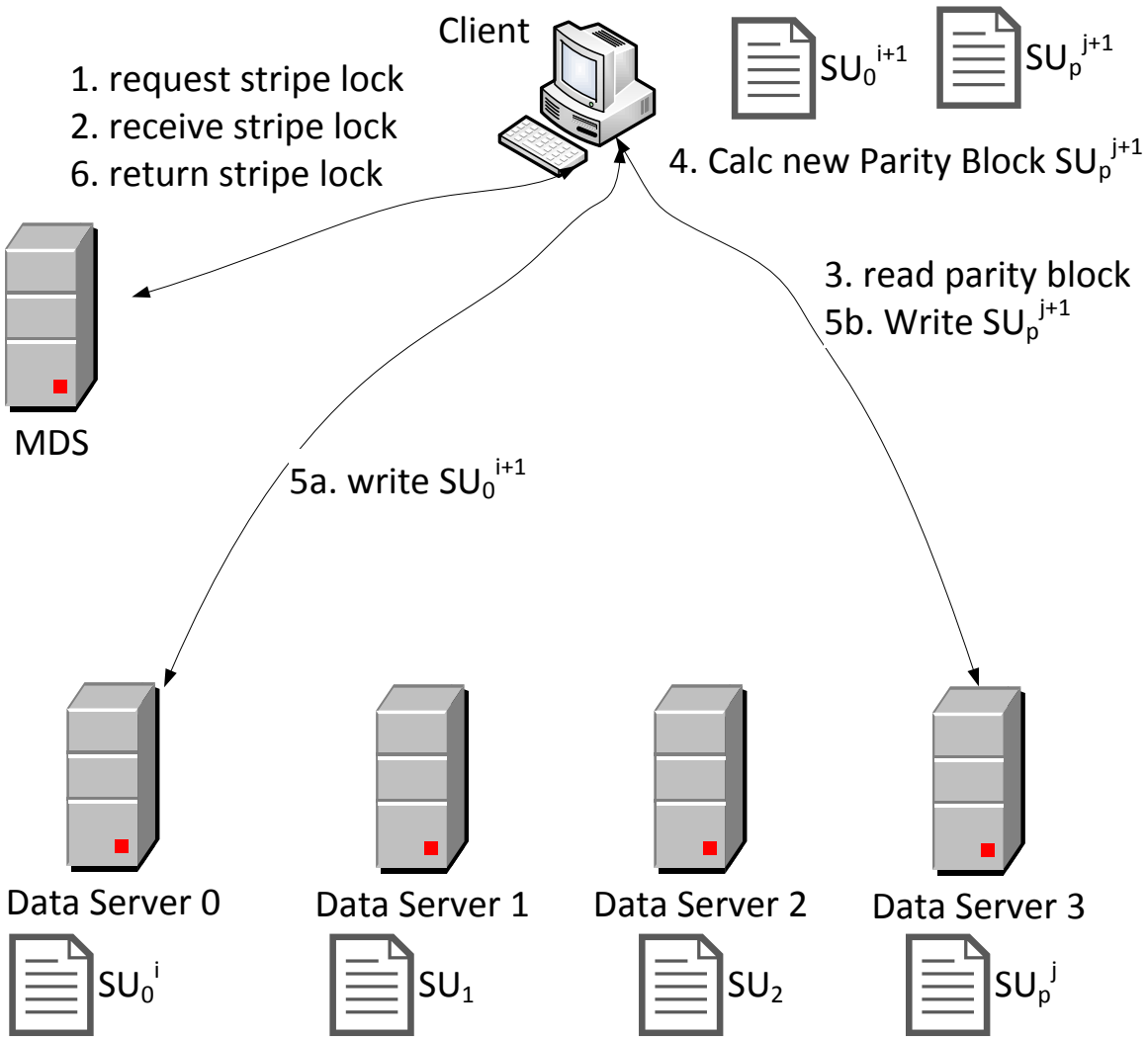
$SU_p = SU_1 \text{ XOR } SU_2 \text{ XOR } SU_3$

$SU_1 = SU_2 \text{ XOR } SU_3 \text{ XOR } SU_p$

⇒ Byte range locks not sufficient to protect stripe consistency

pNFS

Client write operation using stripe locks



Outline

Coordinated Architecture

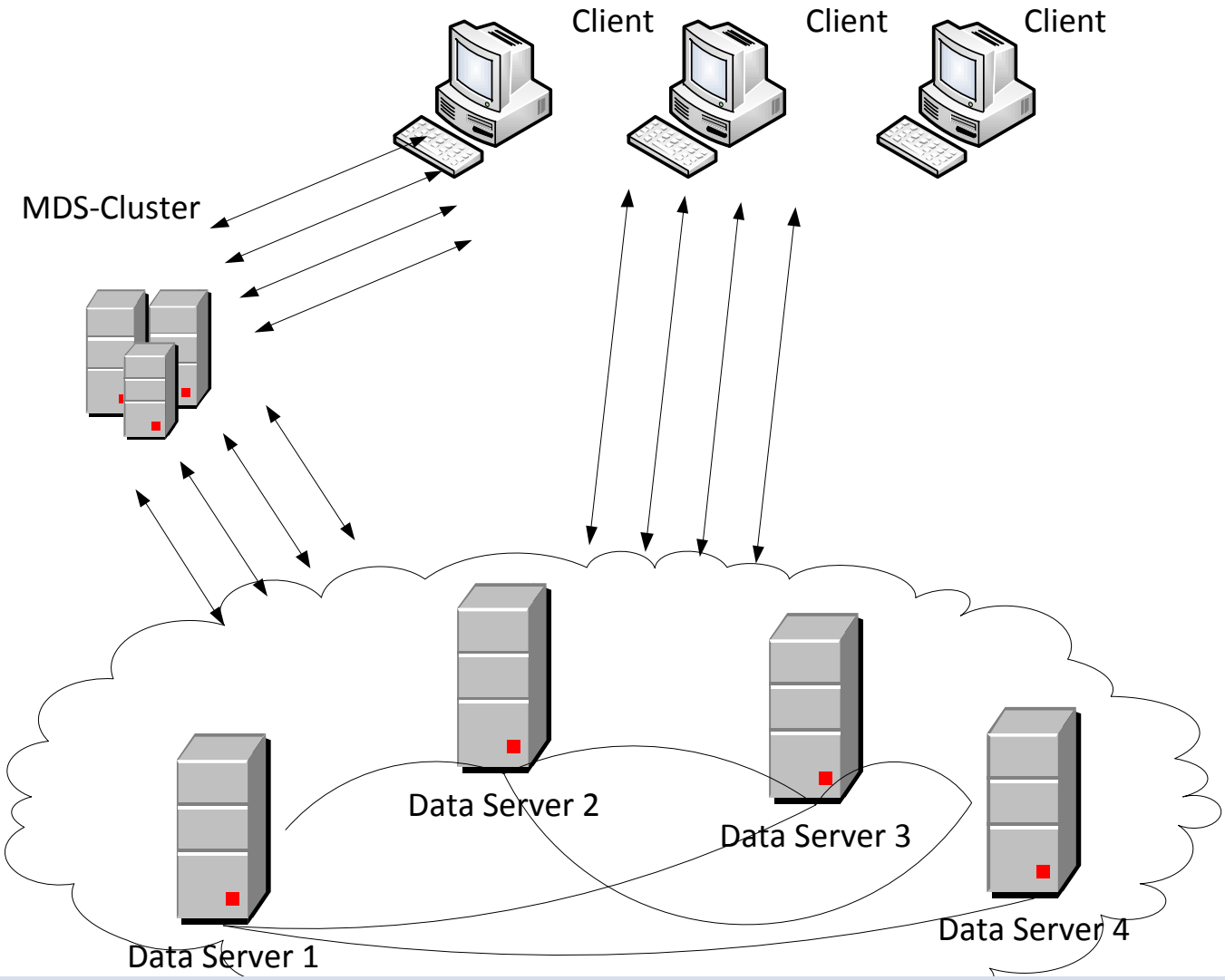
Idea

P2P data server cluster coordinates client operations to...

- resolve conflicting client operations efficiently
- guarantee stripe consistency
- identify and recover from failure
- perform loadbalancing (future work)

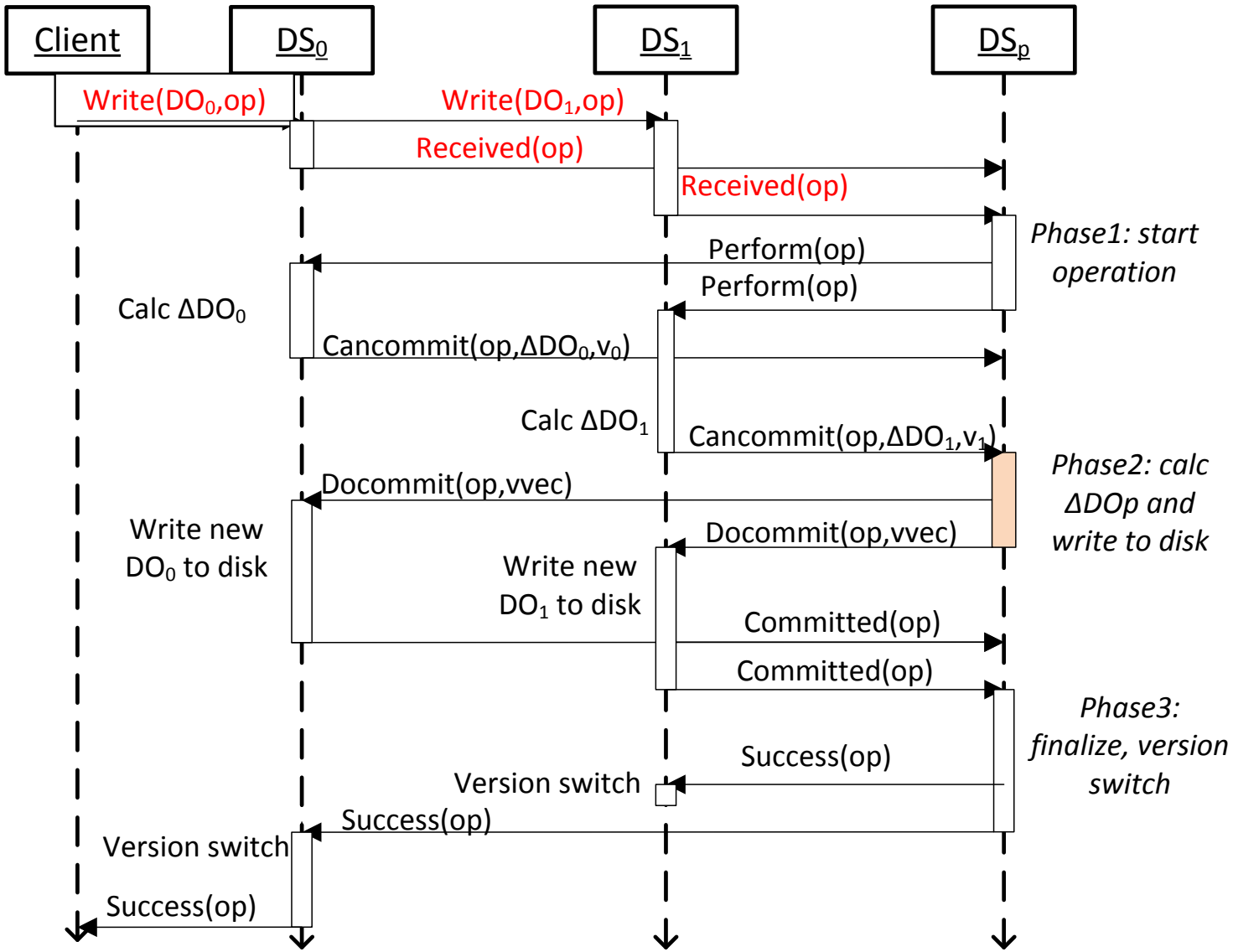
Coordinated architecture

Basic architecture



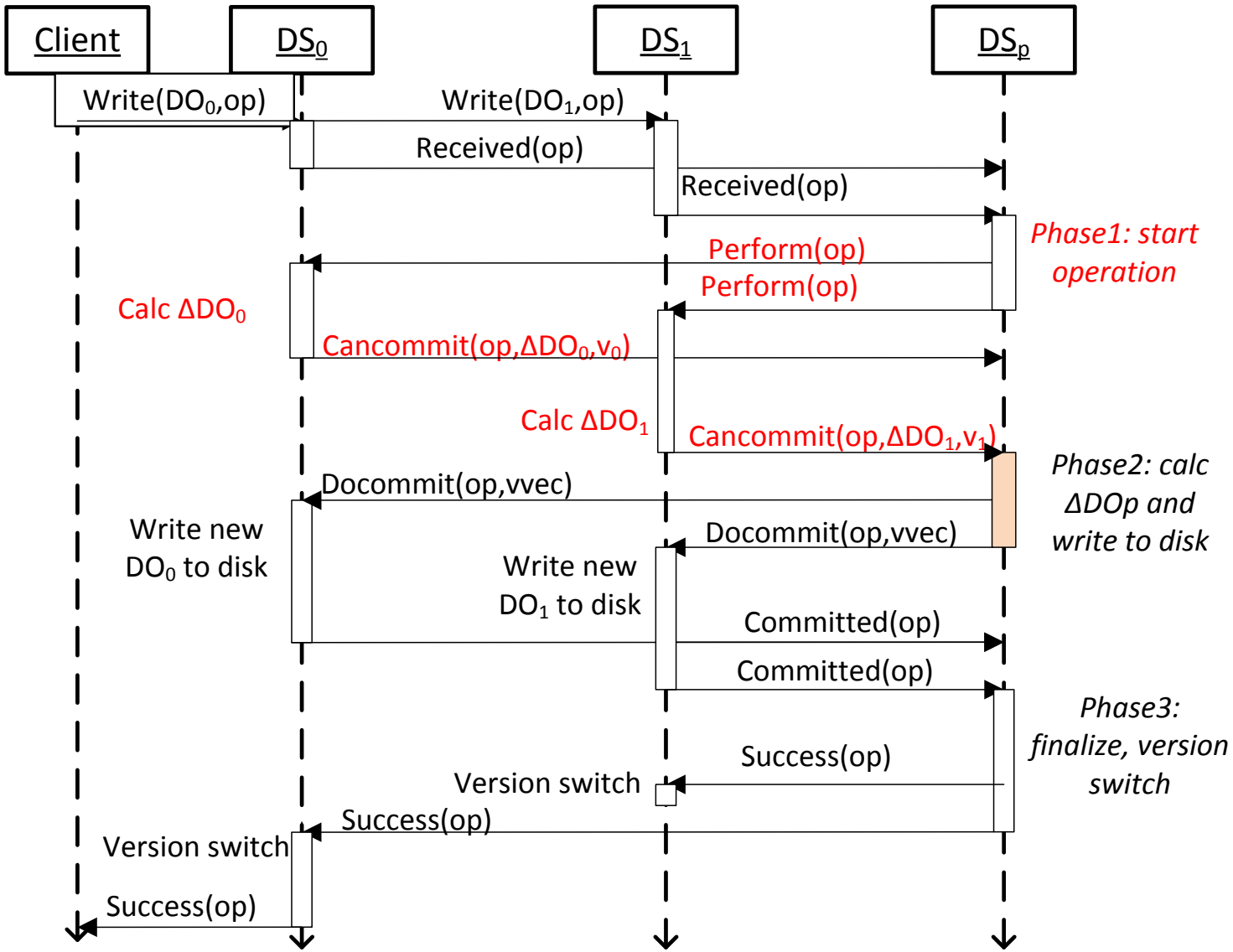
Coordinated architecture

Workflow of a client write operation



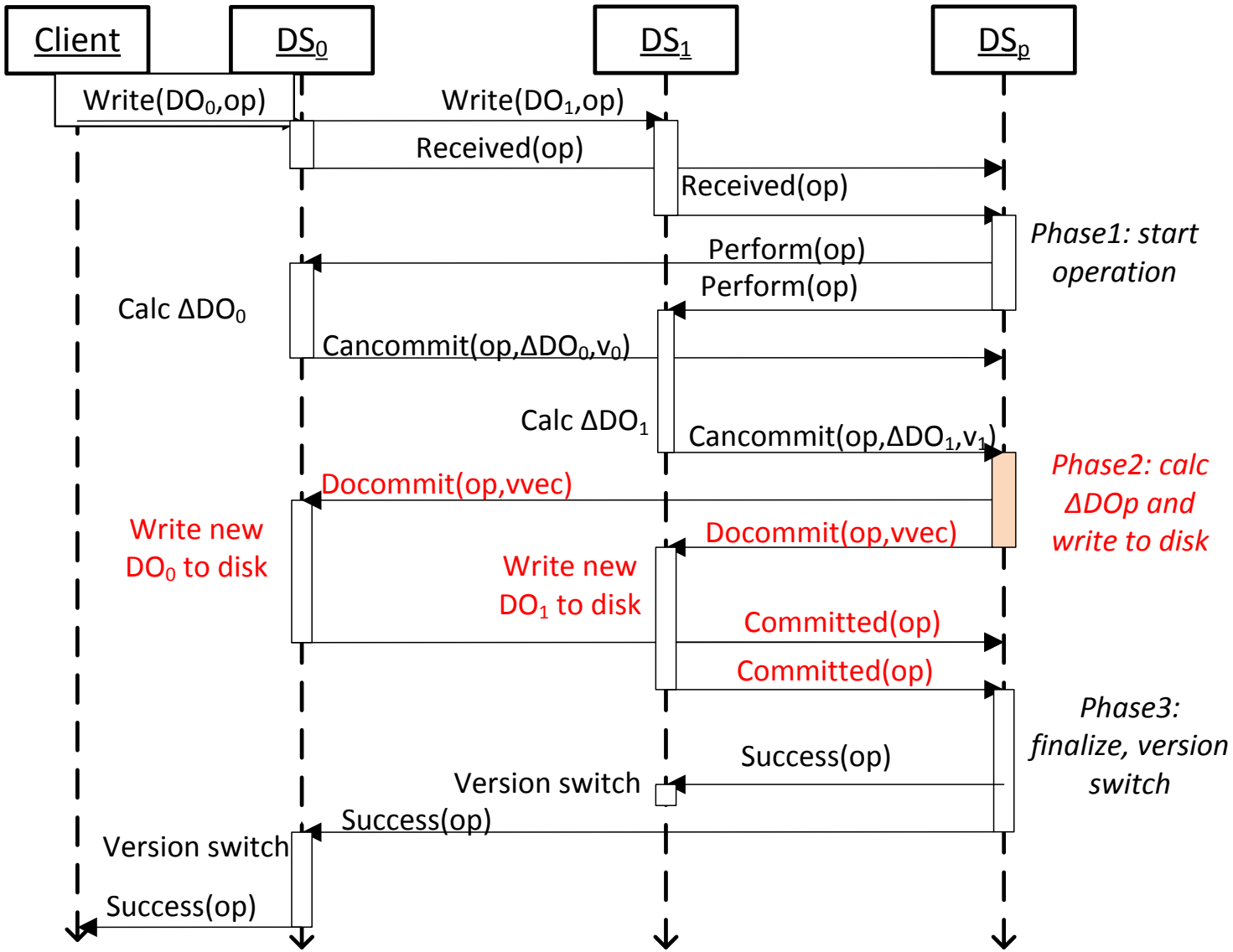
Coordinated architecture

Workflow of a client write operation



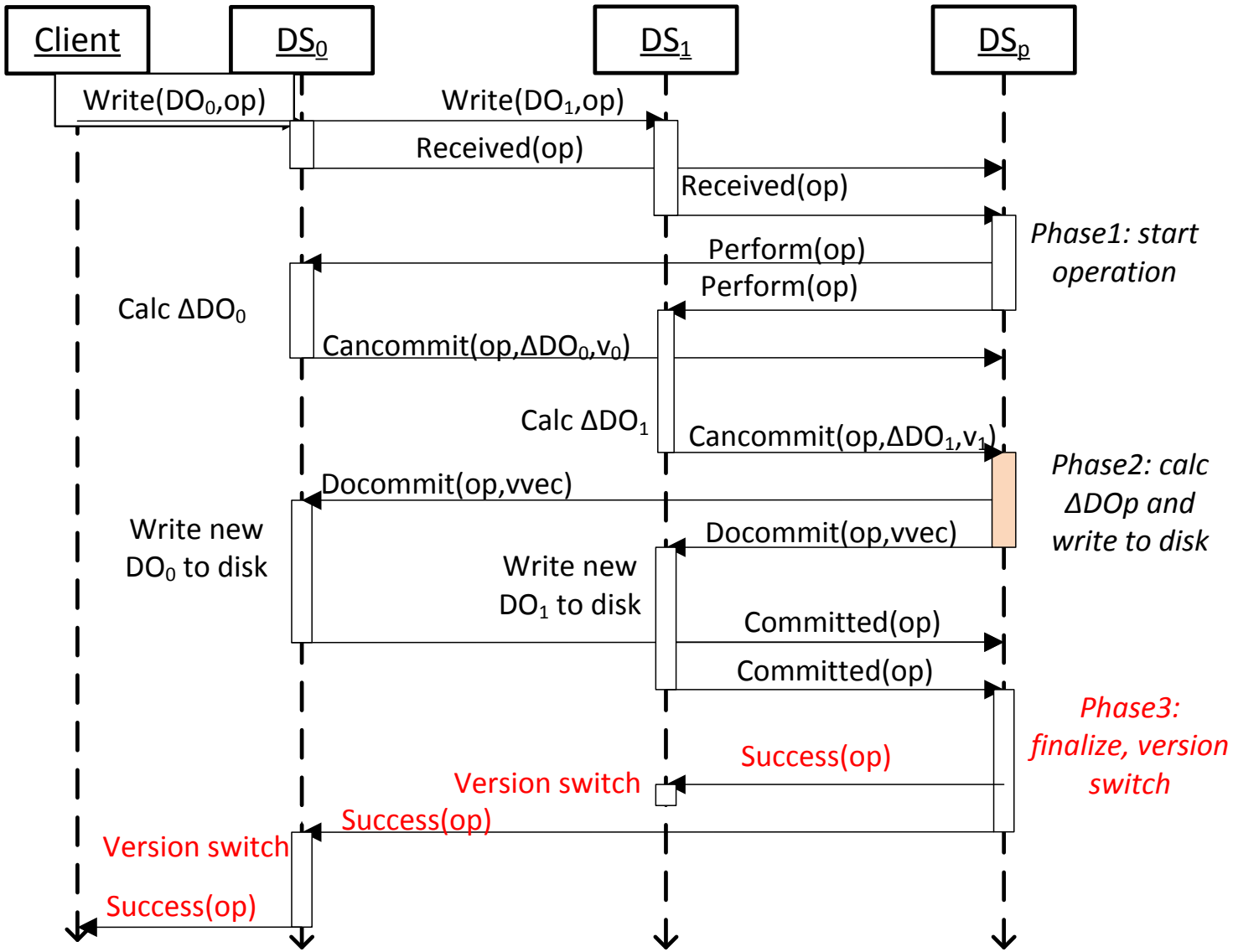
Coordinated architecture

Workflow of a client write operation



Coordinated architecture

Workflow of a client write operation



Outline

Evaluation

Grid5000 (France)

Cluster Suno

- 2x Intel Xeon Quadcore at 2.26 GHz
- 32 GB
- 2x 300 SAS-II (RAID-0)
- Gigabit Ethernet

Cluster Griffon

- 2x Intel Xeon Quadcore at 2.5 GHz
- 16 GB
- 320 SAS-II (used RAM instead)
- Infiniband-G20

64-bit Debian 6.0 (2.6.32-41) and ext3 host file system

Evaluation

Stripelock mode vs. coordinated mode

Evaluate updates on a single stripe:

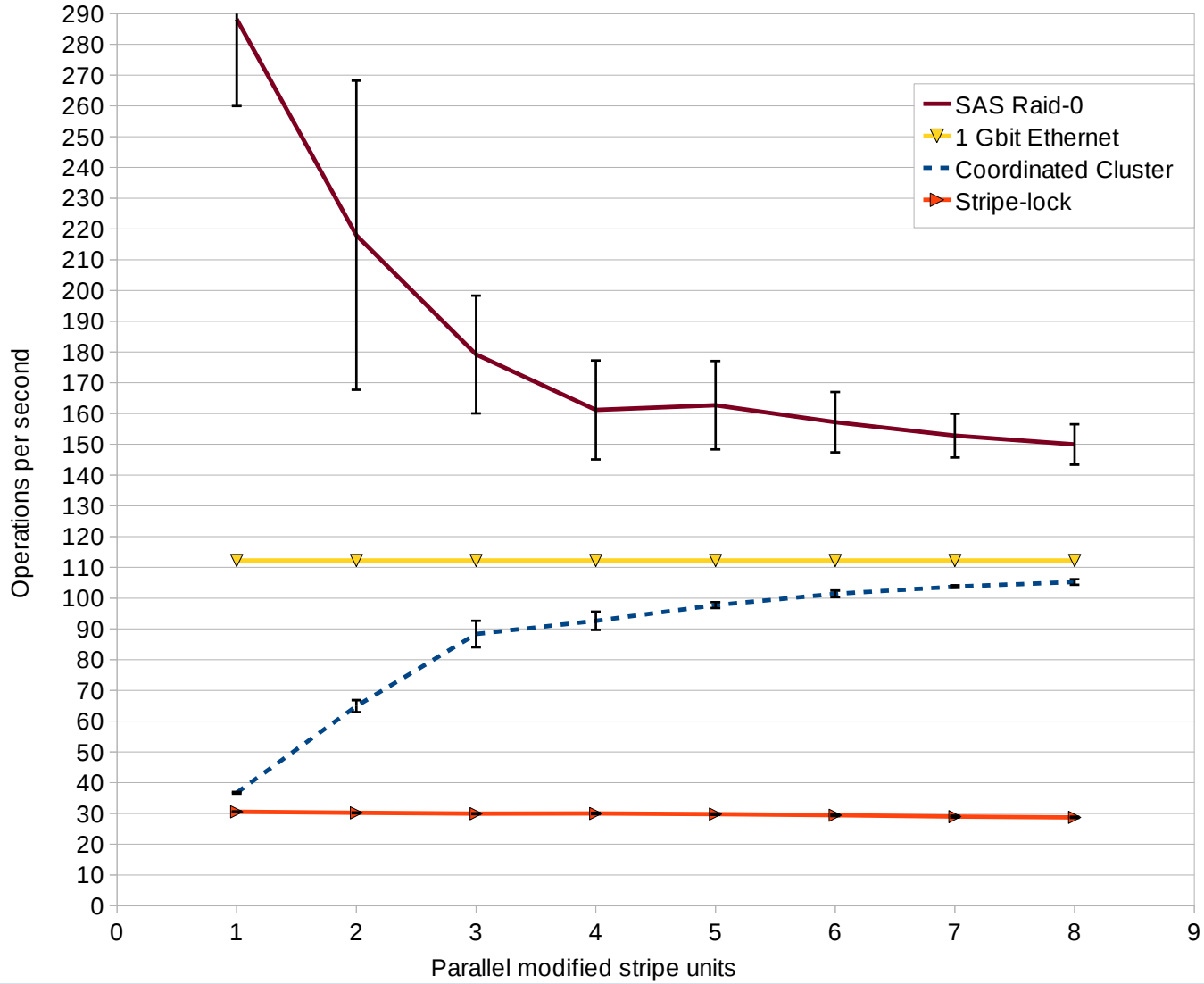
- 8 stripe units + parity block form a stripe
- 1-8 clients repeatedly modify their stripe unit
- 1 MB stripe unit size

Results:

- Throughput at the parity block in operations per second
- Latency of each operation at the client

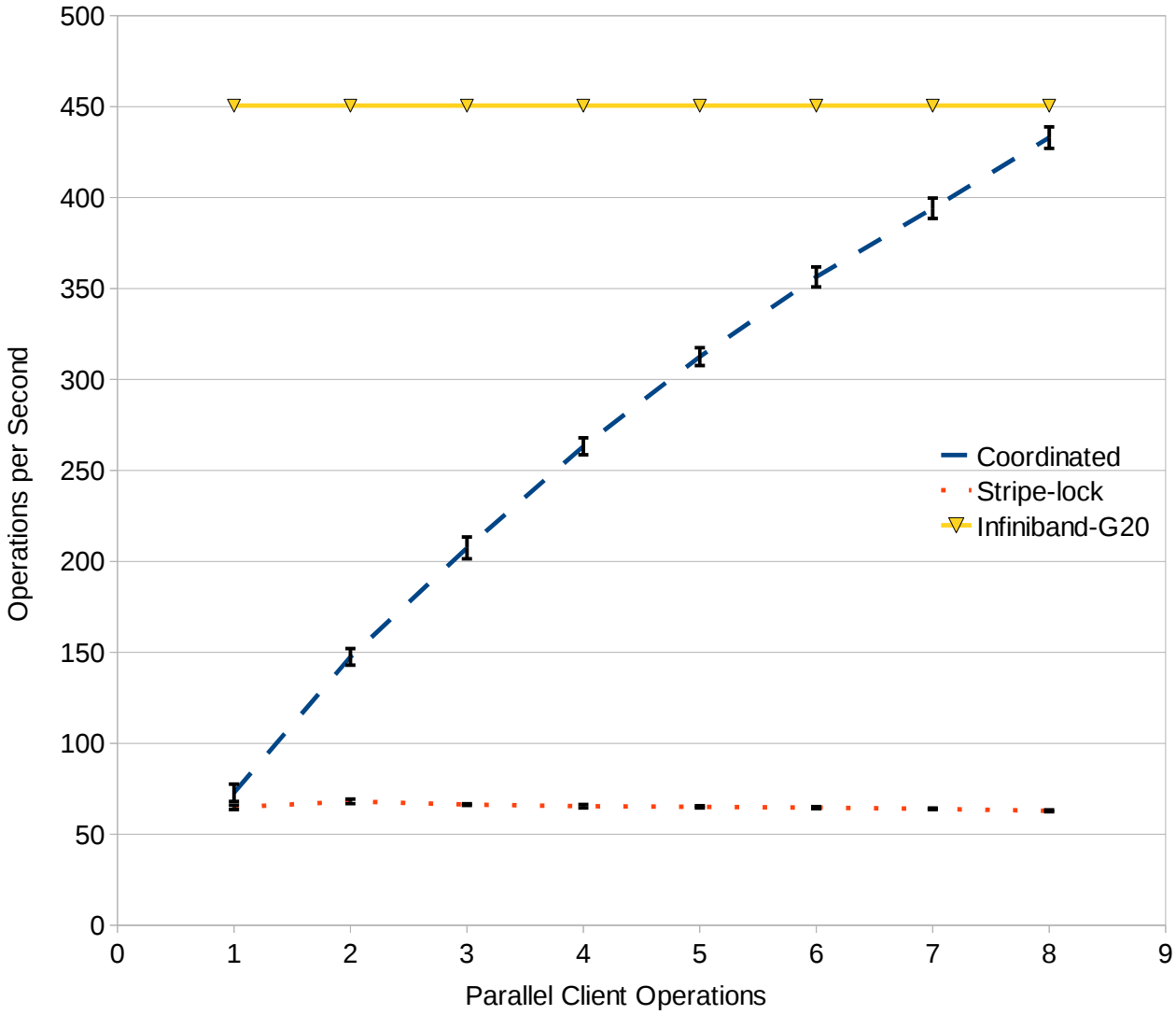
Evaluation: Suno cluster

Throughput in Ops/sec, 1 MB unit size



Evaluation: Griffon cluster

Throughput in Ops/sec, 1 MB unit size



Evaluation: Suno cluster

Latency in ms, 1 MB unit size

Latency in ms	Operation
10,633 ± 0,021	1 MB message transfer
18.172 ± 1,445	two parallel 1 MB messages
4,284 ± 1,726	disk write
0,657 ± 0,000	parity calculation
0,416 ± 0,024	lock acquire/release cycle
0,080 ± 0,000	small message transfer
34,242 ± 3,216	theoretical stripe lock mode latency
32,897 ± 0,197	measured stripe lock mode latency
27,264 ± 1,768	theoretical coordinated mode latency
27,197 ± 0,762	measured coordinated mode latency

Outline

Conclusion

Advantages - Disadvantages

Advantages:

- Critical section: very small, bound to parity server (no messaging!)
- No stripe locks needed
- Guaranteed stripe consistency
- MDS not bothered with I/O operations (stripe lock handling)
- MDS not bothered with recovery operations
- Client not critical to file consistency
- No single-point-of-failure

Disadvantage:

- Generally: High load on data server cluster
- Particularly: Cache hit crucial for write operations

Conclusion

Cache issue

Cache-miss causes disk I/O:

- Disk I/O adds several ms latency
- Interrupts disk I/O of other operations

Hide Cache-miss:

- SSD as second level cache
- Prefetch existing data object while receiving client write operation

Q & A

Thank you for your attention

Are there any questions?