† University of Connecticut

‡ institute IMDEA networks

BRIEF ANNOUNCEMENT:
Oh-RAM! ONE AND A HALF ROUND READ/WRITE ATOMIC MEMORY

Theophanis Hadjistasi †

Nicolas Nicolaou ‡

Alexander A. Schwarzmann †

ACM Symposium on Principles of Distributed Computing

Chicago IL, July 2016

INTRODUCTION

Problem Statement: Emulating atomic read/write shared objects in an *asynchronous, messaging-passing* system where processors are *prone to crash*.

- To cope with processor failures, distributed object implementations use ***redundancy*** by replicating the object at multiple replica servers.
- Since read and write operations may access different object replicas, replication introduces the problem of ***consistency***.

Atomicity [7] (*or linearizability* [6]) is the most intuitive consistency semantic as it provides the illusion of a single-copy object.

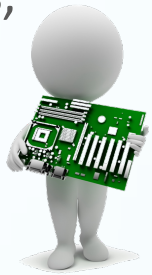
Related Work

- The seminal work of Attiya et al. [1] provided the first algorithm – ABD that implements single-writer/multiple-reader atomic objects.
  - Each operation is guaranteed to terminate as long as some majority of replica servers do not crash.
- Subsequently, Lynch et al. [8] showed how to implement MWMR atomic memory where read and write operations take 4 communication exchanges.
- Dutta et al. [2] introduced a SWMR implementation where both read and writes involve 2 communication exchanges.
  - This is possible only when the number of readers *r* is bounded with respect to the number of servers *s* and the server failures *f*, $r < (s/f) - 2$.
- The later work of Georgiou et al. [4] focused in relaxing the bound on the number of readers and the writers.
  - Proposes hybrid approaches where some operations terminate in 2 and some in 4 communication exchanges.
- Englert et al. [5] provide tight bounds on the number of exchanges that read and write operations require in the MWMR model.

System Model & Efficiency

System: Consists a collection of a failure prone, asynchronous processes with unique identifiers from a totally ordered set $\mathcal{I}$. Set $\mathcal{I}$ consists 3 disjoint sets,

- Set $\mathcal{W}$ of writer identifiers
- Set $\mathcal{R}$ of reader identifiers
- Set $\mathcal{S}$ of replica servers identifiers (maintaining copy of the object)



Communication: *achieved by exchanging messages via asynchronous point-to-point reliable channels.*

Failure Model: *Up to f servers may fail where, f < |S|/2*

Efficiency:

- *Message complexity* - the worst case number of messages exchanged
- *Operation latency,*
  - ***Communication time*** accounts the computation *steps* in each operation
  - ***Communication delay*** - communication "***exchanges***" -> A collection of sends and receives for a specific message type within the protocol

ALORITHM Oh-SAM (SWMR)

Write Protocol: *Identical to the work of Attiya et. al. – algorithm ABD.*

- Writer increments its local timestamp and broadcasts a write request message to all the servers.
- Writer terminates once it receives write acknowledgment messages from a majority of servers.

Server Protocol:

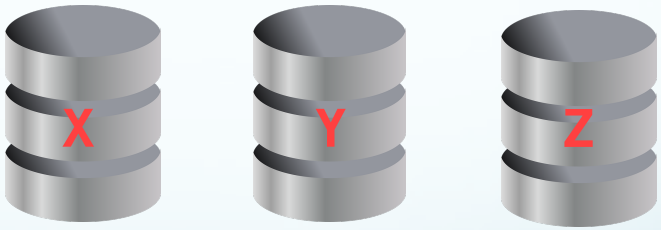
- When a server receives the write request message, it compares the *incoming* with its *local information* and updates accordingly. It then sends an acknowledgment message to the requesting writer.

Write Complexity: *2 Communication exchanges and 2|S| messages.*

READ PROTOCOL (SWMR)

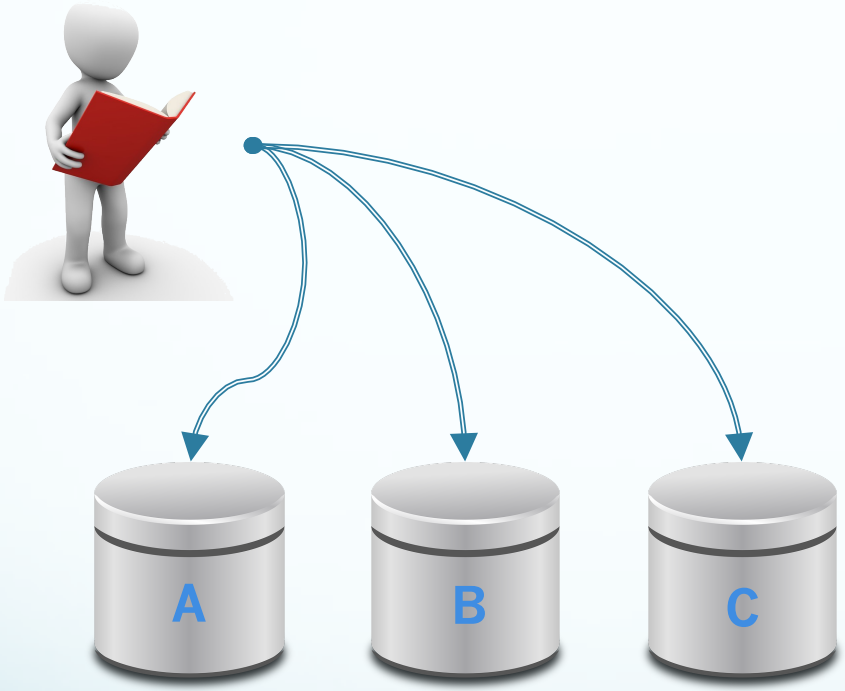
***3** Communication Exchanges Oh-SAM*

***4** Communication Exchanges ABD*



READ PROTOCOL (SWMR)

***3** Communication Exchanges Oh-SAM*



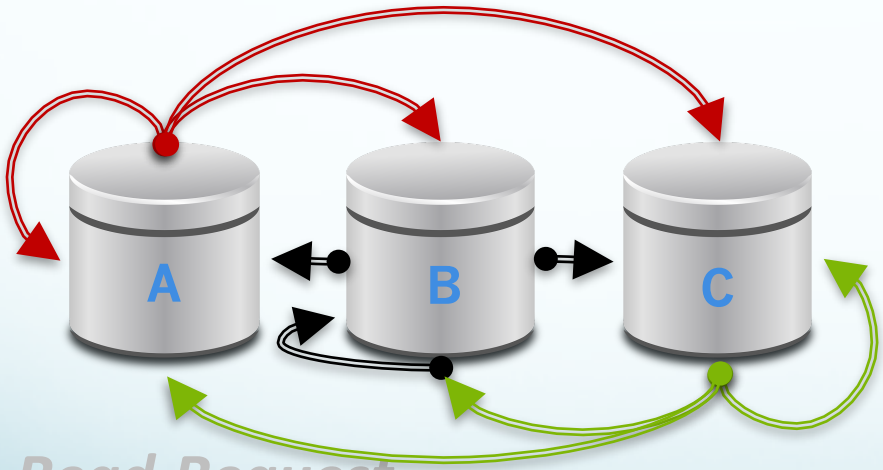
(i) Read Request

***4** Communication Exchanges ABD*

(i) Read Request

READ PROTOCOL (SWMR)

***3** Communication Exchanges Oh-SAM*



(i) Read Request

(ii) Servers Relay

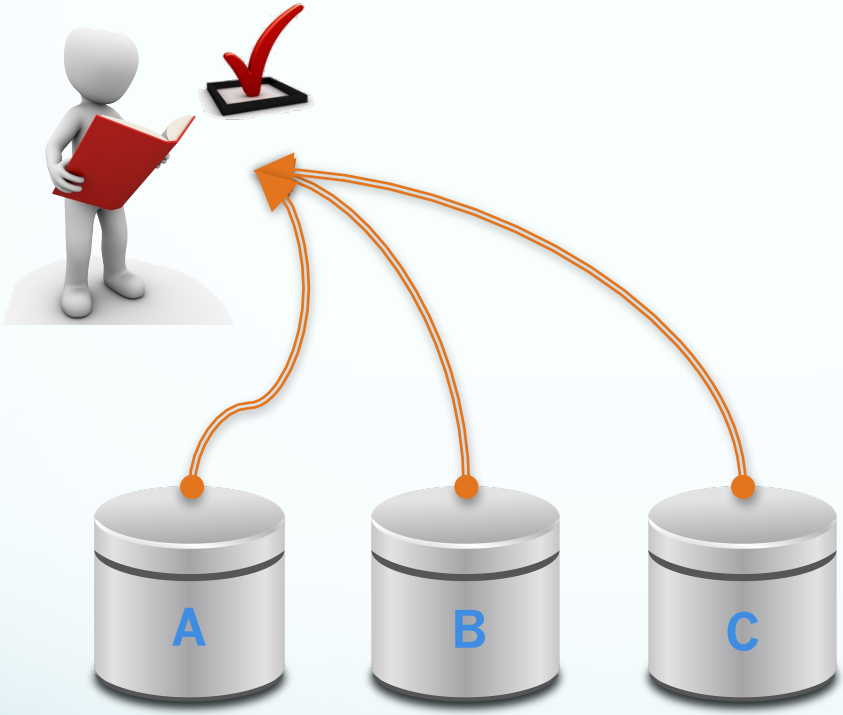
***4** Communication Exchanges ABD*

(i) Read Request

(ii) Servers Reply

READ PROTOCOL (SWMR)

3 Communication Exchanges Oh-SAM



(i) Read Request
(ii) Servers Relay
(iii) Servers ACK

4 Communication Exchanges ABD

(i) Read Request
(ii) Servers Reply
(iii) Reader Writes

READ PROTOCOL (SWMR)

***3** Communication Exchanges Oh-SAM*

(i) Read Request
(ii) Servers Relay
(iii) Servers ACK

***4** Communication Exchanges ABD*

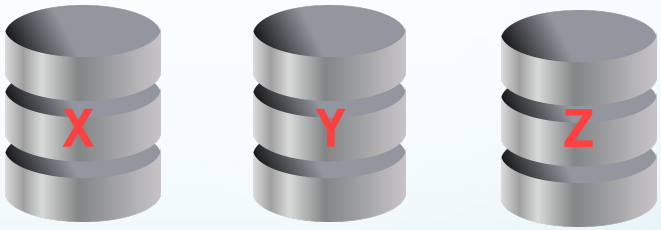
(i) Read Request
(ii) Servers Reply
(iii) Reader Writes
(iii) Servers ACK

READ PROTOCOL (SWMR)

***3* Communication Exchanges Oh-SAM** **4 Communication Exchanges ABD**

Message Complexity

- *ABD: 4|S|*
- *Oh-SAM: $|S^2|$ + 2|S|*



THEOREM: Algorithm Oh-SAM implements an atomic SWMR read/write register.

IMPOSSIBILITY RESULT

We examine if it is possible to implement MWMR atomic read/write objects in an *asynchronous*, *message-passing* system with *crash-prone* processors where both read/write operations take ***three*** communication exchanges.

We consider the following ***three-phase*** scheme,

- The invoker *p* sends a message to a set of servers
- Each server that receives the message from *p* sends a certain relay message to a set of servers
- Once a server receives enough relay messages it replies to *p*

IMPOSSIBILITY RESULT

Why this ***three-phase*** scheme it is reasonable,

1) The invoker *p* sends a message to a set of servers

   Servers ***cannot know*** about a write operation unless writer contacts them

2) Each server that receives the message from *p* sends a certain relay message to a set of servers

   Must be the ***transitional*** phase for the servers to move from phase (1) to (3)

   Must facilitate the ***dissemination of the information*** regarding any write operation to the rest of the servers.

3) Once a server receives enough relay messages it replies to *p*

   It must be the servers who inform the writer about the status/completion of the write operation. Otherwise, a writer ***will wait indefinitely***

THEOREM: It is not possible to obtain an atomic read/write register implementation, where all operations perform 3 communication exchanges, when $|\mathcal{W}| = |\mathcal{R}| = 2$, $|\mathcal{S}| \geq 3$ and $f = 1$

ALORITHM Oh-MAM (MWMR)

Motivated by the impossibility result, we sought a solution that involves 3 or 4 communication exchanges per operation.

Compared to the SWMR setting, we need to impose an ordering on the values that are concurrently written by multiple writers.

- We associate each value with a ***tag*** consisting of a pair of a timestamp ***ts*** and the ***id*** of the writer - ***tag = <ts, id>***
- We use ***lexicographic*** comparison to order tags *(cf. [8])*

ALORITHM Oh-MAM (MWMR)

Write Protocol: *Identical to algorithm ABD for MWMR.*

Read Protocol: Identical to the 3 communication exchanges protocol we discussed earlier for algorithm Oh-SAM (SWMR)

Complexities:

- ***Write Oh-MAM & ABD:*** 4 Communication exchanges and $4|S|$ messages.
- ***Read Oh-MAM:*** 3 Communication exchanges and $|S^2| + 2|S|$ messages.
- ***Read ABD:*** 4 Communication exchanges and $4|S|$ messages.

THEOREM: Algorithm Oh-MAM implements an atomic SWMR read/write register.

CONCLUSIONS

We focused on the problem of emulating *atomic read/write shared objects* in a message-passing setting using ***three*** communication exchanges – equivalent of *one-and-a-half* traditional rounds.

We presented,

- An algorithm for the SWMR setting Oh-SAM
- The Impossibility to implement an algorithm for the MWMR setting where the operations take 3 communication exchanges
- An algorithm for the MWMR setting Oh-MAM

We note that the algorithms ***do not*** impose any constrains on the number of readers (SWMR & MWMR) or the writers (MWMR)

Both algorithms are ***optimal*** in terms of communication exchanges when no bounds are imposed on participation.

Thank you!

REFERENCES

[1] Attiya, H., Bar-Noy, A., and Dolev, D. Sharing memory robustly in message passing systems. *Journal of the ACM 42(1) (1996),* 124 - 142.

[2] Dutta, P., Guerraoui, R., Levy, R. R., and Chakraborty, A. How fast can a distributed atomic read be? *In Proceedings of the 23rd ACM symposium on Principles of Distributed Computing (PODC) (2004),* pp. 236-245.

[3] Englert, B., Georgiou, C., Musial, P. M., Nicolaou, N., and Shvartsman, A. A. On the efficiency of atomic multi-reader, multi-writer distributed memory. *In Proceedings 13th International Conference On Principle Of DIstributed Systems (OPODIS 09) (2009),* pp. 240 – 254.

[4] Georgiou, C., Nicolaou, N. C., and Shvartsman, A. A. Fault-tolerant semifast implementations of atomic read/write registers. *Journal of Parallel and Distributed Computing 69, 1 (2009),* 62-79.

[5] Hadjistasi, T., Nicolaou, N., and Schwarzmann, A. A. Oh-ram! one and a half round read/write atomic memory. *www.arxiv.com, 2016*.

[6] Herlihy, M. P., and Wing, J. M. Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems (TOPLAS) 12, 3 (1990)*, 463-492.

[7] Lamport, L. How to make multiprocessor computer that correctly executes multiprocess program. *IEEE Trans. Comput. 28, 9 (1979)* 690-691.

[8] Lynch, N. A., and Shvartsman, A. A. Robust emulation of shared memory using dynamic quorum-acknowledged broadcasts. *In Proceedings of Symposium on Fault-Tolerant Computing (1997)*, pp. 272 - 281.