Ίδρυμα Προώθησης Έρευνας
ΚΥΠΡΙΑΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΔΙΑΡΘΡΩΤΙΚΑ ΤΑΜΕΙΑ της Ευρωπαϊκής Ένωσης στην Κύπρο
οι ιδέες μας, πράξη και ανάπτυξη
ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ

Η ΔΕΣΜΗ 2009-10 ΣΥΓΧΡΗΜΑΤΟΔΟΤΕΙΤΑΙ ΑΠΟ ΤΗΝ ΚΥΠΡΙΑΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΚΑΙ ΤΟ ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΕ

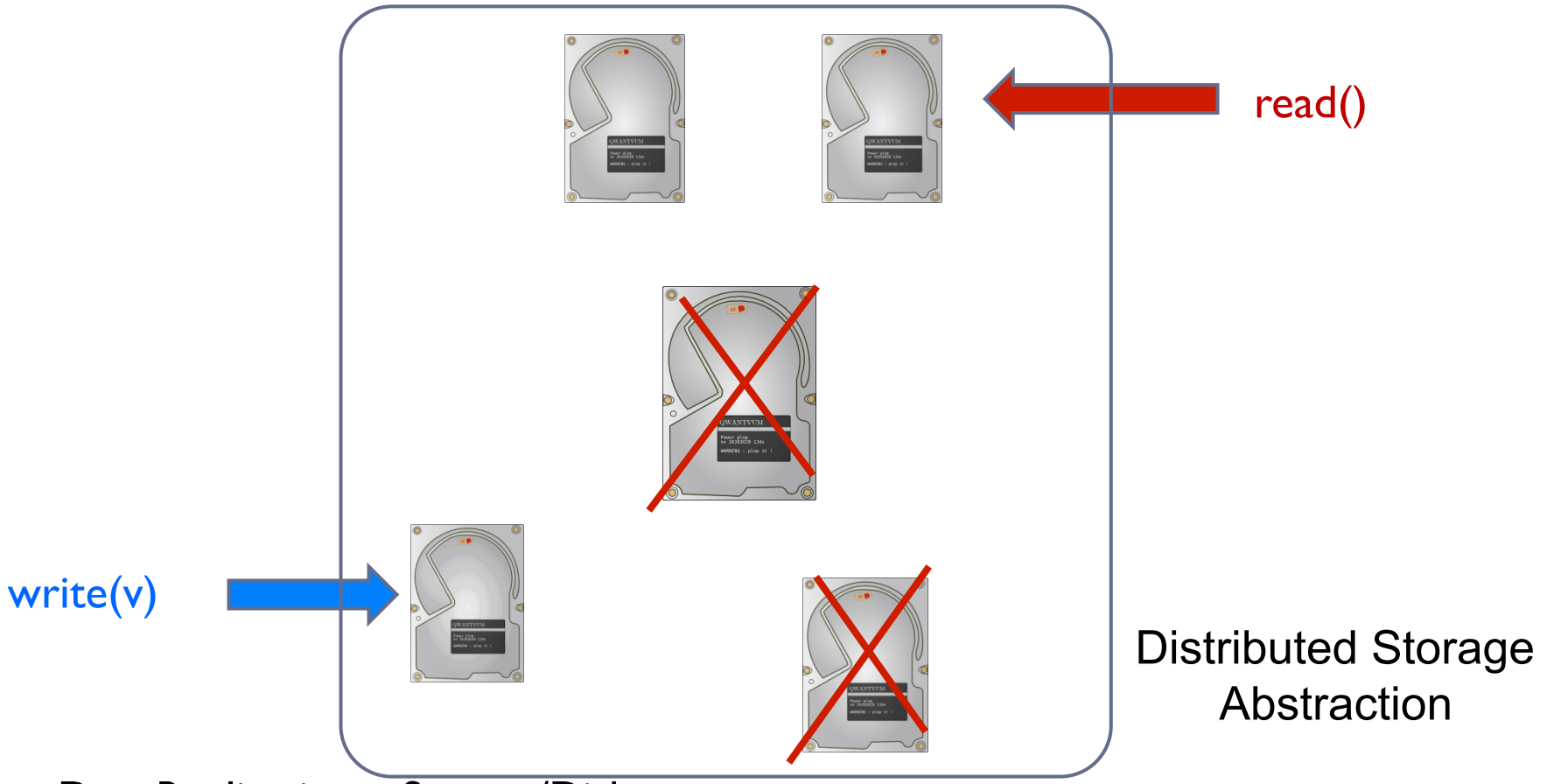
Seeking Fast Operations in MWMR Atomic Register Implementations

Nicolas Nicolaou

University of Cyprus &
University of Connecticut

Funded by the Cyprus Research Promotion Foundation and co-funded by the Republic of Cyprus and the European Regional Development Fund

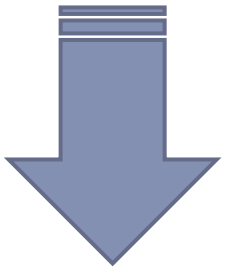
What is a Distributed Storage System?



- Data Replication – Servers/Disks
  - Survivability and Availability
- Read/Write operations
- Consistency Semantics
  - Atomicity

Complexity Measure-Operation Latency

- Consistent Register Implementations
  - **Message-Passing, Asynchronous** model
  - Access **multiple replicas** per operation
  - Perform **multiple accesses** per operation



Operation Latency is measured in
Communication Rounds (round-trips)

What was known…

Traditional SWMR

- Timestamp-Value pair [ABD96]
- Single round writes
- Two round reads
  - Phase 1: Obtain latest value
  - Phase 2: Propagate latest value
    - Folklore belief: "Reads must Write"

Traditional MWMR

- Two round writes
  - Phase 1: Discover latest value
  - Phase 2: Order new value after the latest and propagate
    - Belief: "Writes must Read"
- Two round reads

The Era of Fast Implementations…

SWMR Fast

- Single round (***fast***) writes and reads [Dutta et al. 04]
  - Bounded readers: $R<(S/f)-2$ where S servers & f failures
  - Impossible in MWMR model

SWMR Semifast

- Fast writes [Georgiou et al. 06]
- Only a single complete 2-round (***slow***) read per write
  - Unbounded readers
  - Impossible in the MWMR model

SWMR Weak-Semifast

- General Quorum System [Georgiou et al. 08]
- Fast writes and Multiple slow reads per write
  - Allows concurrent fast reads with writes
  - Unknown if applicable in MWMR model

Exploring the possibilities in the MWMR

Can we obtain fast write and read operations in the MWMR model?

Achievements…

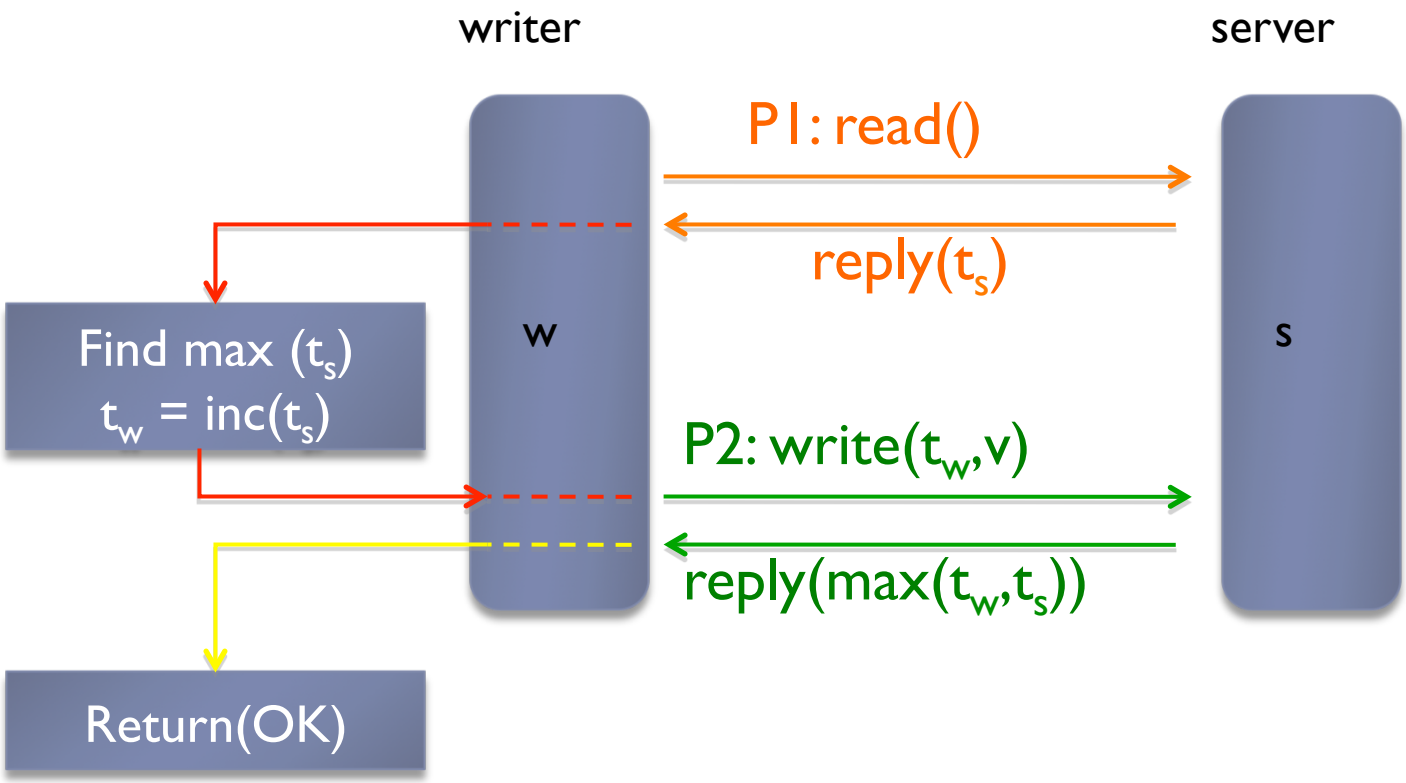
- Introduced new technique for value ordering
  - Server Side Ordering (SSO)

- Devised an Atomic Register implementation in the MWMR model
  - Enables Fast Writes and Reads -- first such algorithm
  - Allows Unbounded Participation

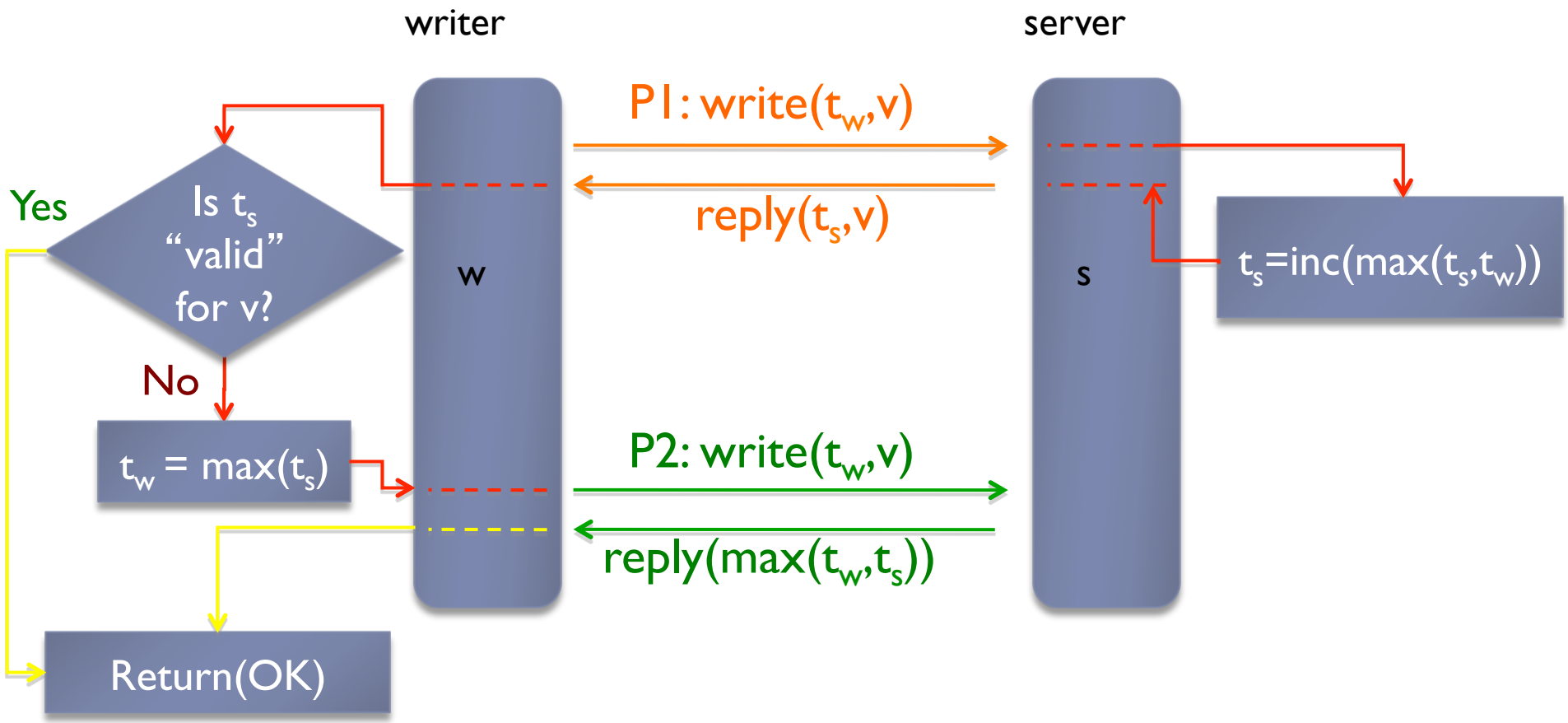
SFW Algorithm – An intuitive idea

- Value versioning
  - Associate a tag with each value
- Traditional Approaches
  - Increment tag (version) at the writer
- SSO Approach
  - Increment tag (version) at the server
    - Avoid discovery of the latest version from the writer

Traditional Writer-Server Interaction



SSO Writer-Server Interaction



Definition: n-wise Quorum Systems

Definition: A quorum system Q is an n-wise quorum system, if:

$$\mathbf{Q} = \{Q : Q \subseteq S\} \; where \; \forall A \subseteq \mathbf{Q} : |A| = n \text{ and } \bigcap_{Q \in A} Q \neq \emptyset$$

Algorithm: SFW (in a glance)

Write Protocol: one or two rounds

- P1: Collect candidate tags from a quorum
  - Exists tag t propagated in a bigger than (n/2-1)-wise intersection (PREDICATE PW)
    - YES – assign t to the written value and return => **FAST**
    - NO - propagate the unique largest tag to a quorum => **SLOW**

Read Protocol: one or two rounds

- P1: collect list of writes and their tags from a quorum
  - Exists max write tag t in a bigger than (n/2-2)-wise intersection (PREDICATE PR)
    - YES – return the value written by that write => **FAST**
    - NO - propagate the largest confirmed tag to a quorum => **SLOW**

Server Protocol

- Increment tag when receive write request and send to read/write the latest writes

Lower Bounds

Theorem: No execution of safe register implementation that use an N-wise quorum system, contains more than $N-1$ consecutive, quorum shifting, fast writes.

Remark: SFW algorithm is near optimal since it allows up to $\left(\frac{N}{2}-1\right)$ consecutive, quorum shifting fast writes.

The Weak Side of SFW

- Predicates are Computationally Hard
  - NP-Complete
- Restriction on the Quorum System
  - Deploys n-wise Quorum Systems
  - Guarantees fastness iff n>3

NP-Completeness

K-SET-INTERSECTION: (captures both PR and PW)

Given a set of elements U, a subset of those elements $M \subseteq U$, a set of subsets $\mathbb{Q} = \{Q_1, \ldots, Q_n\}$ s.t. $Q_i \subseteq U$, and an integer $k \leq |\mathbb{Q}|$, a set $I \subseteq \mathbb{Q}$ is a k-intersecting set if: $|I| = k$, $\bigcap_{Q \in I} Q \subseteq M$, and $\bigcap_{Q \in I} Q \neq \emptyset$.

Theorem: K-SET-INTERSECTION is NP-complete (reduction from 3-SAT).

The Good News…

- Approximation Algorithm (APRX-SFW)
  - Greedy Algorithm
  - Log-approximation
    - log|S| the number of slow operations

- Algorithm CWFR
  - Based on Quorum Views
    - SWMR prediction tools
  - Fast operations in General Quorum Systems
  - Trades Speed of Write operations
    - Two Round Writes