Brief Announcement: Space Bounds for Reliable Multi-Writer Data Store

Inherent Cost of Read/Write Primitives*

Gregory Chockler[1], Dan Dobre[2], Alexander Shraer[3], and Alexander Spiegelman[4]

[1] Royal Holloway, University of London, Gregory.Chockler@rhul.ac.uk
[2] European Patent Office, dan@dobre.net
[3] Google, Inc., shralex@google.com
[4] Technion, sashas@tx.technion.ac.il

1 Introduction

Reliable storage emulations from fault-prone components have established themselves as an algorithmic foundation of modern storage services and applications. Most existing reliable storage emulations are built from storage services supporting *custom-built* read-modify-write (RMW) primitives (e.g., [2]). Since such primitives are not typically available with pre-existing or off-the-shelf components (such as cloud storage services or network-attached disks), it is natural to ask if they are indeed essential for efficient storage emulations.

In this paper, we answer this question in the affirmative. We prove that the number of registers required to emulate a reliable multi-writer register for k clients from a collection of multi-writer multi-reader (MWMR) atomic base registers hosted on crash-prone servers requires at least kf registers where f is the maximum number of tolerated server failures. We further show that this bound cannot be circumvented even in the failure-free runs where emulated register operations do not execute concurrently, which implies that no such algorithm can be adaptive to point contention.

Given the base registers are stored on servers, we also address the number of servers required to support the emulation under assumption that the number of registers per server is bounded by a known constant m. We show that the number of servers required to support k clients exceeds the requisite kf/m servers stipulated by our space bound by at least $f + 1$ additional servers.

Our bounds apply to any reliable implementations of a multi-writer register, which are at least *safe* [8], and *solo-terminating*. They complement and tighten the lower bounds of [1], and shed light onto inherent costs of the existing constructions of reliable services out of unreliable MWMR registers [4].

On a positive side, we show that Compare-and-Swap (CAS) primitives, which readily available with many popular cloud data stores (such as Amazon DynamoDB) can be used to emulate a reliable multi-writer atomic register with constant storage and adaptive time complexity.

* This work was partially supported by Royal Society International Exchanges grant.

2 Overview of the Results

We assume an asynchronous distributed system where clients coordinate by accessing shared objects stored on a collection of servers. Both clients and servers can fail by crashing. We study *space complexity* of storage services that mask the client and server failures by emulating a single reliable MWMR register. Our reliability requirement is f*-tolerance*, that is, the register must remain correct as long as at most f servers and any number of clients crash.

Lower Bounds Below, we give statements of our space lower bounds for f-tolerant register emulations where the objects stored on servers are MWMR atomic wait-free registers. The proofs can be found in [5].

Theorem 1. *For any $k \geq 0$, $f \geq 0$, there is no f-tolerant algorithm emulating a multi-writer safe [8] solo-terminating register for k clients that uses less than kf base registers in all failure-free runs r such that no two emulated register operations execute concurrently in r.*

Theorem 2. *For any $f > 0$, there is no f-tolerant algorithm that emulates a multi-writer safe solo-terminating register such that the number of registers used by the emulation is adaptive to point contention [3].*

Theorem 3. *For any $m > 0$, $\ell > 0$, and $f \geq 0$, there is no f-tolerant algorithm emulating a multi-writer safe solo-terminating register register for $k \geq \ell m$ clients using less than $\ell f + f + 1$ servers if each server can store at most m registers.*

Upper Bound In [5], we show that a single CAS object per server is sufficient to implement an f-tolerant wait-free atomic MWMR register for any number of clients whose time complexity is adaptive to point contention. Our result is derived in a modular fashion by first obtaining the RMW primitive required by the multi-writer ABD emulation (MW-ABD) of [7] from a sinlge CAS, and then plugging the resulting primitive into MW-ABD.

References

1. Aguilera, M., Englert, B., Gafni, E.: On Using Network Attached Disks As Shared Memory. In: PODC 2003.
2. Attiya, H., Bar-Noy, A., Dolev, D.: Sharing Memory Robustly in Message-Passing Systems. J. ACM 42(1).
3. Attila, H., Fouren, A.: Algorithms Adapting to Point Contention. J. ACM 50(4).
4. Basescu, C., Cachin, C., Eyal, I., Haas, R., Sorniotti, A., Vukolic, M., Zachevsky, I.: Robust Data Sharing With Key-Value Stores. In: DSN 2012.
5. Chockler, G., Dobre, D., Shraer, A., Spiegelman, A.: Space Bounds for Reliable Multi-Writer Data Store: Inherent Cost of Read/Write Primitives (2015). Available at `https://pure.royalholloway.ac.uk/portal/files/25314522/main.pdf`
6. Gafni, E., Lamport, L.: Disk Paxos. Distributed Computing 16(1).
7. Gilbert, S., Lynch, N.A., Shvartsman, A.A.: Rambo: A Robust, Reconfigurable Atomic Memory Service for Dynamic Networks. Distributed Computing 23(4).
8. Shao, C., Pierce, E., Welch, J.L.: Multi-writer Consistency Conditions for Shared Memory Objects. In: DISC 2003.