Atomic Memory in BAMP systems

Yosef Goren

On **'Atomic Read/Write Memory in Signature-Free Byzantine Asynchronous Message-Passing Systems'**.

A paper by:

Achour Mostefaoui, Matoula Petrolia, Michel Raynal, Claude Jard

Table of Contents

1. Introduction
   - System Model
   - Solution Requirments
   - Background
2. The Algorithm
   - (Very) Naive Algorithm
3. Analysis

BAMP: Byzantine Asynchronous Message Passing

There are up to $t < \frac{n}{3}$ **Byzantine Processes**.

Messages are delivered **Asynchronously**.

Processes communicate by **Message Passing** (Clique).

'*Signature Free*' - No cryptographic primitives required.

Atomic Concistency

Atomic Concistency is how we intuitively expect memory to be, it is also known as Linearizability.

A Simple Definition

'for any execution of the system, there is some way of totally ordering the reads and writes so that the values returned by the reads are the same as if the operations had been performed in that order, with no overlapping.'

- 'On Interprocess Communication' (1985), Leslie Lamport.

Single Writer Multiple Reader Registers (*SWMR*)

A single process can write; everyone can read.

Single Writer & Byzantine Processes

If all shared memory can be written by all processes - a single Byzantine process can destroy it.

Local Copies

Each process p_i has Reg_i, but can only write to $Reg_i[i]$.

p_1	p_2	p_3
$Reg_1[1]$	$Reg_2[1]$	$Reg_3[1]$
$Reg_1[2]$	$Reg_2[2]$	$Reg_3[2]$
$Reg_1[3]$	$Reg_2[3]$	$Reg_3[3]$

But who cares?

What we get

This implementation provides a reduction from Message Passing models to Atomic Memory models.

What it can be used for

Many distributed algorithms are based on atomic memory; this reduction provides instant implementations of these algorithms in message passing systems.

Examples

- Atomic, multi-writer multi-reader registers
- Concurrent time-stamp systems
- Atomic snapshot scan

Previous Atomic Register Algorithms

(Very) Naive Algorithm