Generalized Agreement Between Concurrent Fail-Stop Processes

James E. Burns, Rolando I. Cruz, and Michael C. Loui

Abstract. For a system of concurrent processes that can fail by stopping, we study a generalization of the traditional binary agreement problem having more than two possible input values. We provide bounds on the number of possible inputs for which agreement is possible in a system of $n$ concurrent processes that communicate using read-modify-write operations on $m$ shared memory cells of sizes $r_1, r_2, \ldots, r_m$. Let $V$ be the set of input values. We present an agreement protocol for two processes

$$V \leq \left( \prod_{j=1}^{m-1} r_j \right) \left( r_m - 1 \right),$$

for $m = 1$ and $m = 2$, we prove that this upper bound on $|V|$ is the best possible.

A protocol for $n$ processes is fully resilient if it tolerates up to $n - 1$ failures; a fully resilient protocol is wait-free, because no process needs to wait for any other. In a write-once protocol, each memory cell changes value at most once during each execution of the protocol. We present a fully resilient write-once agreement protocol for

$$|V| \leq \sum_{j=1}^{m} (r_j - 1).$$

We show that no fully resilient write-once agreement protocol exists when

$$|V| > \sum_{j=1}^{m} (r_j - 1)$$

and $n \geq m$.

1 Introduction

Reaching agreement between processes is a fundamental problem of distributed computation. An important application of agreement is in commitment protocols for distributed databases. In such protocols, all processes must agree whether to write (commit) or discard (abort) the new values of the database records. Another application for agreement is in the design of mutual exclusion algorithms for processes that share a resource. In mutual exclusion algorithms, processes must agree on which process gains access to the resource. A fortiori, agreement is part of any distributed system that involves coordinated activity such as the synchronization of clocks [15] and the election of leaders [12].

Agreement protocols depend on the characteristics of the system. The processes may communicate by exchanging messages or by accessing cells in a shared memory. Process failures may be fail-stop, in which processes fail by stopping, or Byzantine, in which processes can be actively malicious. In practice, most components of a distributed system fail by stopping, but it is also worth knowing whether agreement can be achieved in the worst case, with Byzantine failures.

In the traditional agreement (or consensus) problem, each process has a binary input, and at the end of the execution of the protocol, all processes must agree on a decision that

* Supported by the National Science Foundation under Grant CCR-8922008.
is one of their inputs. This problem has been studied in both message passing systems and shared memory systems, with both stopping failures and Byzantine failures.

Fischer et al. [8] showed that it is impossible to achieve agreement in asynchronous message passing systems, even with only one stopping failure. Their result also holds for Byzantine failures. Loui and Abu-Amara [11] proved that it is impossible to reach agreement in an asynchronous shared memory system that provides only read and write operations with one stopping failure. They also showed the impossibility of agreement in an asynchronous shared memory system that supports atomic read-modify-write operations with two stopping failures, in the special case of binary cells. Generalizing the results of Loui and Abu-Amara, Merritt and Taubenfeld [13] introduced the concept of knowledge in shared memory systems. They analyzed the number of shared memory cells needed to achieve various levels of knowledge.

All of the papers cited above treat only deterministic protocols. Several researchers [1, 2, 3, 4] designed randomized protocols for shared memory systems with read and write operations. With randomized protocols, it is possible to reach agreement in the presence of stopping failures. The algorithm of Abrahamson [1] has an exponential expected number of operations. Subsequently Aspnes and Herlihy [4], Aspnes [3], and Abrahamson and Karkare [2] presented algorithms that successively reduced the expected number of operations to $O(n^4)$, $O(p^2 + n)$, and $O(n)$, respectively, where $n$ is the number of processes in the system and $p$ is the number of active processes. Fich et al. [7] proved that every randomized protocol for wait-free agreement requires $\Omega(\sqrt{n})$ memory cells.

Turpin and Coan [16] and Chaudhuri [6] studied extensions of the binary agreement problem to message-passing systems in which the input at each process may come from a set of more than two elements. Turpin and Coan showed how to modify every binary Byzantine agreement protocol to achieve multivalued agreement in just two additional rounds, where full values are sent only on the first round. Chaudhuri defined the $k$-set consensus problem, in which the processes must agree on at most $k$ different values, and each of these values must be an input of some process. Chaudhuri showed that protocols exist if there are at most $k - 1$ stopping failures. Recently, Borowski and Gafni [5], Herlihy and Shavit [10], and Saks and Zaharoglou [14] proved that it is impossible to achieve $k$-set consensus with $k$ failures.

In this paper we consider a system of $n$ concurrent processes that communicate using read-modify-write operations on $m$ shared memory cells. Processes fail by stopping. Each process has an input from a set $V$. Our work is a generalization of the traditional agreement problem from 2 to $|V|$ possible inputs.

In Section 2 we define the system model formally. There are $n$ processes and $m$ shared memory cells $c_1, \ldots, c_m$. Cell $c_j$ has $r_j$ possible values. Without loss of generality, let $r_m = \max_j \{r_j\}$. In Section 3 we present an agreement protocol for two processes with $|V| \leq (\prod_{j=1}^{m-1} r_j)(r_m - 1)$. In addition, for $m = 1$ and $m = 2$, we prove that this upper bound on $|V|$ is the best possible: there is no agreement protocol if $m = 1$ and $|V| > r_1 - 1$, or if $m = 2$ and $|V| > r_1(r_2 - 1)$. In Section 4 we study write-once protocols, in which each cell changes value at most once. Write-once protocols are easier to design and analyze than general protocols. We present a write-once agreement protocol for $|V| \leq \sum_{j=1}^{m}(r_j - 1)$, and we prove that there is no write-once agreement protocol when $|V| \geq \sum_{j=1}^{m} (r_j - 1)$ and $n \geq m$, when up to $n - 1$ processes may fail.