Randomized Consensus in Expected $O(n^2 \log n)$ Operations

Gabi Bracha Ophir Rachman
Technion, Haifa, Israel

Abstract

We consider asynchronous shared memory distributed systems, and investigate coordination problems in this model. We provide a wait-free randomized consensus protocol that requires an expected $O(n^2 \log n)$ atomic operations.

1 Introduction

In the theory of distributed computation, the behavior of multi-processor environments is being studied. The various processors can be allocated in different sites, and can communicate through various communication media, such as massage passing networks, shared variables, and others. The behavior of the distributed system is studied with respect to the type of the communication media, and other characteristics of the system. For instance, the system may be synchronous, i.e., the processors have an access to a global clock, or to the contrary, the system may be totally asynchronous, i.e., the individual processors operate in rates that are independent on one another. Furthermore, the processors may be totally reliable, or may crash fail during their execution, (i.e., a processor may stop executing its code and never recover), or even worse, the processors may operate maliciously against the system.

All these characteristics and others specify the model of the distributed system. In spite of the large variety of models, in the heart of any distributed system is the ability to achieve coordination between the individual processors. The ability to coordinate gives an arbitrary set of scattered processors the power of a distributed system. Consequently,

*This research was partially supported by the U.S.-Israel bi-national science foundation grant 88-00-282.

†Both authors can be reached at: Department of Computer Science, Technion, Haifa 32000, Israel. (At e-mail: fimfam@csb.cs.Tecnion.ac.il).
coordination problems are one of the most investigated subjects in the field of distributed computation.

In this paper we consider coordination problems in the shared memory model, where the processors are totally asynchronous, and they communicate through shared single-writer multi-reader registers. We develop coordination protocols in this model that are highly resilient in the presence of crash faults. The correctness of the protocols is guaranteed even when all the processors crash fail except for one. Such protocols are called wait-free protocols since a processor can not wait for any event that depends on the activity of the other processors. A full and formal definition of the model can be found in several papers, [H 88], [L 86], [ALS 90], and others.

One of the most widely studied coordination problem in all distributed systems models and in particular in the shared memory model is the consensus problem. In a wait-free consensus protocol, each processor starts with a binary input value, (initially not available to other processors), and upon termination, decides on an output value. The protocol must satisfy the following conditions:

- **Consistency**: All the processors that terminate decide on the same value.
- **Validity**: The processors decide on a value that is an input value of some processor.
- **Termination**: Each processor terminates the protocol after a finite number of its own steps, regardless of the other processors' activity.

It has been shown in [CIL 89], [DDS 87], [LA 87], and others, that no deterministic protocol can satisfy the above conditions. However, it has been found that probabilistic protocols, that allow the processors to flip coins, can satisfy consistency and validity, and a probabilistic version of the termination condition:

- **Probabilistic termination**: Each processor terminates the protocol after an expected finite number of its own steps, regardless of the other processors' activity.

A probabilistic version of the consensus problem, namely, the randomized consensus problem, is defined by the probabilistic termination condition together with the previous consistency and validity conditions.

One of the first randomized consensus protocols was presented in [CIL 89]. This protocol assumed that the processors have an access to a global shared coin. A protocol that overcomes this strong assumption was introduced in [A 88]. However, the expected number of register operations in this protocol is exponential in the number of processors. The first randomized consensus protocol with expected polynomial number of operations was offered in [AH 90], and the first polynomial protocol that was also memory bounded was offered in [ADS 89]. In the latter two protocols, the expected number of operations is \( O(n^4) \). So far, the most efficient protocols were introduced in [SSW 90], and in [BR 90]. The expected number of operations in these protocols is \( O(n^3) \), and they use