Experimental Evaluation of a Failure Detection Service Based on a Gossip Strategy

Leandro P. de Sousa and Elias P. Duarte Jr.
Federal University of Parana (UFPR) - Dept. Informatics
P.O. Box 19018 Curitiba 81531-980 PR Brazil
{leandrops,elias}@inf.ufpr.br

Abstract. Failure detectors were first proposed as an abstraction that makes it possible to solve consensus in asynchronous systems. A failure detector is a distributed oracle that provides information about the state of processes of a distributed system. This work presents a failure detection service based on a gossip strategy. The service was implemented on the JXTA platform. A simulator was also implemented so the detector could be evaluated for a larger number of processes. Experimental results show that increasing the frequency in which gossip messages are sent gives better results than increasing the fanout. Results are included for fault and recovery detection time and mistake rate of the detector.

Keywords: Failure Detectors, P2P, Probabilistic Dissemination.

1 Introduction

Several distributed applications involve some kind of agreement between their components [11]. Processes must reach a consensus whenever they need to decide on the same value given an initial entry consisting of a set of possible values. As both processes and communication channels can fail in real distributed systems, a basic condition for distributed processes to reach an agreement is that each process must know the state (working or failed) of the other processes in the system. In some types of distributed systems, this can be hard or even impossible to implement. That is the case of asynchronous systems: in this type of system, processes and its communication channels can behave arbitrarily slowly, making it impossible to distinguish slow and failed processes. Lynch and others proved in [1] that consensus is impossible in an asynchronous system in which even a single process can fail by crashing. This result is known as the FLP impossibility.

As way of avoiding the FLP impossibility and thus solving the consensus problem in asynchronous systems, Chandra et al. proposed abstractions called unreliable failure detectors [1]. A failure detector is a distributed oracle that provides information about the state of processes of a distributed system. Failure detectors can make mistakes, i.e. fault-free but slow processes can be erroneously
considered to be suspect. Chandra and Toueg proposed two properties to classify failure detectors: completeness and accuracy. Completeness requires that if a process has crashed then it is suspected by the failure detector, while accuracy restricts the mistakes that the detector can make. Even though it is impossible to implement perfect failure detectors in completely asynchronous systems, consensus algorithms using unreliable failure detectors can complete successfully if the detector output can be trusted for a long enough period [10]. Also, solutions built around failure detectors are simpler and more generic, as failure detectors encapsulate the timing properties of the system [1].

In order to allow real applications to use and take advantage of failure detectors, they might require some properties. Completeness [1] is said to be weak if eventually every process that crashes is permanently suspected by some correct process. Accuracy can be eventual: there is a time after which mistakes do not occur. Applications have timing restrictions, and detectors that are too slow may not suffice. For this very reason, [2] proposes some metrics for the quality of service, or simply QoS, of failure detectors. The metrics proposed by the authors are mostly used to describe the speed and accuracy of detection.

Relate work includes [3], a protocol that includes a failure detector and a dissemination protocol for membership information. This failure detector was initially proposed in [6]. The detector uses a randomized ping strategy, where each process periodically tests another process, which was selected randomly. Information about group membership and process failures are piggybacked in the ping messages sent by the detectors. Recently [13] an implementation of a failure detection service was reported for a P2P storage system that tries to improve the detection QoS by using monitoring together with a prediction model.

This work presents the specification and implementation of a distributed failure detection service based on epidemic dissemination. The detection service proposed in this paper is based on the gossip strategy proposed in [12]. In the algorithm, processes periodically send gossip messages to a group of other processes, which are chosen randomly. Failure detection is based on a heartbeat mechanism. Each gossip message contains the heartbeat value for the sending process and the last heartbeat values it has received for every other process. The detection protocol is probabilistic, and uses a gossip strategy [5]. To use the detector, a process must implement the service and then participate in a detection group. At any moment, the process can query its local detector and receive a list of processes suspected to have failed.

The detection service was implemented as a prototype in the P2P JXTA platform [7]. A simulator was also implemented, using the SMPL library [9]. Experimental results are reported and show that increasing the frequency in which gossip messages are sent gives better results than increasing the fanout. Results are given for fault and recovery detection time and mistake rate of the detector.

The rest of this paper is organized as follows. Section 2 presents the proposed detection service and the gossip algorithm on which the detector is based.