TOWARD A FASTER FAULT TOLERANT CONSENSUS TO MAINTAIN DATA CONSISTENCY IN COLLABORATIVE ENVIRONMENTS

FOUAD HANNA
UMR CNRS FEMTO-ST Institute, Franche-Comte University,
25000 Besanon, France
fouad.hanna@femto-st.fr

LIONEL DROZ-BARTHOLET
Covalia Interactive
25000 Besanon, France
lionel.droz@covalia.com

JEAN-CHRISTOPHE LAPAYRE
UMR CNRS FEMTO-ST Institute, Franche-Comte University,
25000 Besanon, France
jc.lapayre@femto-st.fr

Received (Day Month Year)
Revised (Day Month Year)

The consensus problem has become a key issue in the field of collaborative telemedicine systems because of the need to guarantee the consistency of shared data. In this paper we focus on the performance of consensus algorithms. First, we studied, in the literature, the most well-known algorithms in the domain. Experiments on these algorithms allowed us to propose a new algorithm that enhances the performance of consensus in different situations. During 2014, we presented our very first initial thoughts to enhance the performance of the consensus algorithms, but the proposed solution gave very moderate results. The goal of this paper is to present a new enhanced consensus algorithm, named FLC (*Fouad, Lionel and j.-Christophe*). This new algorithm was built on the architecture of the Mostefaoui-Raynal (MR) consensus algorithm and integrates new features and some known techniques in order to enhance the performance of consensus in situations where process crashes are present in the system. The results from our experiments running on the simulation platform Neko show that the FLC algorithm gives the best performance when using a multicast network model on different scenarios: in the first scenario, where there are no process crashes nor wrong suspicion, and even in the second one, where multiple simultaneous process crashes take place in the system.

Keywords: Fault tolerance; consensus; asynchronous distributed systems; unreliable failure detectors.

1. Introduction

Over the past ten years, telemedicine systems have become more popular. These systems offer new tools to physicians and improve both efficiency and quality of care. Some of these systems, known as collaborative systems, allow several healthcare professionals to communicate together to diagnose a patient.

In telemedicine applications where several practitioners collaborate to establish a diagnostic, all participating practitioners must have the same view of the shared workspace. Therefore, these applications need to guarantee the consistency of shared data objects. Another important aspect is that telemedicine applications have to be fault tolerant (the crash of one or several participants should not forbid others from continuing their work).

In collaborative platforms, each participant performs operations on the shared workspace. Then, these operations must be exchanged among all participants. Data inconsistency occurs when the order of execution of exchanged operations is not the same for one or more participants.

A typical example of data inconsistency problems is illustrated in Figure 1, where two practitioners work together on the same patient and they start executing operations. When these operations are exchanged in order to collaborate their work we might end up with different order of execution of these operations at each practitioner; and this results in data inconsistency among the participants.

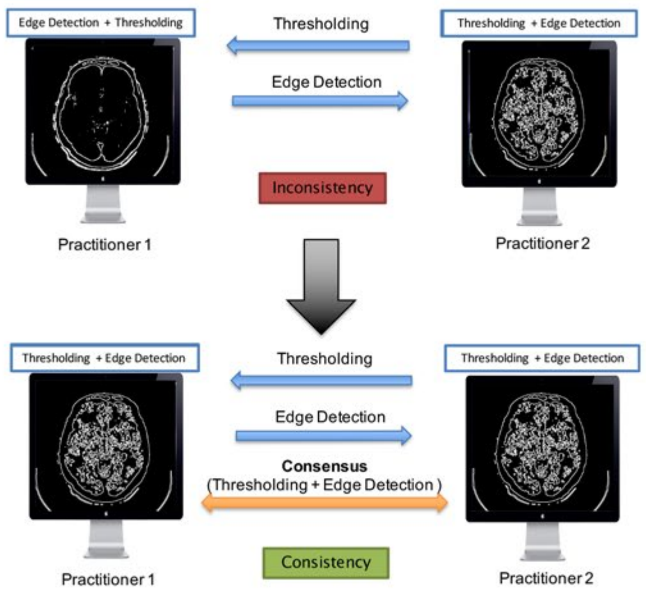
Edge Detection + Thresholding
Thresholding
Thresholding + Edge Detection
Edge Detection
Inconsistency
Practitioner 1
Practitioner 2
Thresholding + Edge Detection
Thresholding
Thresholding + Edge Detection
Edge Detection
Consensus
(Thresholding + Edge Detection)
Consistency
Practitioner 1
Practitioner 2

Fig. 1: Using consensus to guarantee data consistency in collaborative work

To solve this problem, all participants must agree on the same order of execution of operations. To do so, they must use a consensus algorithm that allows them to have a common decision regarding the order of executions of these operations. In order to manage data consistency in collaborative systems, our consistency management protocol Ramos [1] was designed to implement a shared memory in a dynamic environment. It is based on the use of a consensus algorithm which is a fundamental building block for managing data consistency. Moreover, Ramos [1] was especially developed to manage data consistency in medical collaborative environments where usually the maximum number of participants is less than or equals 8. Our goal is to enhance the performance of this protocol (Ramos) and to develop a new version better suited to high process mobility environments. The choice of consensus algorithm affects the global performance of the protocol using it.

In this paper we present the results of our work on the newly proposed consensus algorithm called FLC. Our algorithm uses the leader oracle Ω to circumvent the impossibility result of the FLP theory [11]. It adopts a decentralized communication pattern and tolerates a maximum number of process failures (process crashes) $f < \lceil \frac{n}{2} \rceil$.

The principal idea of the new algorithm is to use a leader election phase that guarantees the existence of only one leader process per execution round. In addition, we use two techniques: the first is to piggyback the leader vote messages $LeaderAck(r_i)$ of a round $r+1$ in the EST messages of the previous round r (this will reduce the number of communication steps when processes proceed from one round to another); and the second is to make use of the system stability and especially from the fact that the elected leader process does not crash from one consensus run to another. Therefore, at the beginning of a consensus run c, the leader of the previous terminated consensus run $c-1$ will be designated, if not crashed, as the leader of the current run and will start directly by sending its estimate of the decision to all other processes. Our consensus algorithm achieves the lower bounds (two message delays) defined for asynchronous consensus [19] [5].

The performance of our algorithm was analyzed and compared to the performance of five of the most known algorithms in the domain: Chandra-Toueg [2] (CT), Mostefaoui-Raynal [21] (MR), MRLeader [22], L. Lamport (Paxos [17] [18]) and multi-Paxos (MPaxos [17] [8]).

In the first section of this paper, we present the context of our work; and then, we introduce the new consensus algorithm (the FLC algorithm) by presenting a detailed description of its execution. Next, the results of the experiments that were conducted using the Neko platform are presented. FLC is compared to the five most known algorithms in the literature. The last section is dedicated to conclude and to propose possible future work.