Causal Cycle Based Communication Pattern Matching
(Short Paper)

Himadri Sekhar Paul

Interra Systems India (P) Ltd.,
Sector-V, Salt Lake,
Kolkata - 700091
himadrip@cal.interrasystems.com

Abstract. A distributed system employing checkpoint and rollback-recovery as a fault tolerance mechanism, suffers from overhead attributed by the technique. Authors in \cite{4} proposes a technique to automatically identify a checkpoint and recovery protocol based on a pre-estimated database of overhead measures. The technique depends on computation of similarity between a pair of communication patterns. The computation involves first partitioning both the communication patterns into small pieces or *splices*. A pair of splices, one taken from each of the two communication patterns in question, are then compared to compute a similarity measure. Splicing a communication pattern is an important step in the method since it bears heavy significance for later steps in the computation. This paper introduces a new method for splicing. Experimental results show that the technique yields better similarity measure values in comparison to results reported in \cite{4}.

1 Introduction

Fault tolerance is an important issue in distributed systems, and checkpoint and rollback-recovery is a cost effective technique to address the issue. A variety of checkpoint and recovery protocols have been proposed for distributed systems \cite{2}. However, there are some overheads associated with all these protocols. For a distributed system hosting long running applications, performance of a checkpoint and recovery protocol is an important issue because it directly affects the overall throughput of the system. The overheads depend on various system and application characteristics, like network speed, stable storage latency, checkpoint size, fault distributed, etc. \cite{5}. Since the performance of a checkpoint and recovery protocol depends on the application characteristic, like communication pattern, the choice of a protocol may change with the application hosted in the system. In \cite{4} the authors proposed an automated method to dynamically identify a checkpoint and recovery protocol which is likely to affect better throughput in the system. The method utilizes an archive of performance values of different checkpoint and recovery protocols under different application and system conditions. The method proposes to determine the parameters from the system and the application and then choose the checkpoint and recovery
Comparing parameters, except communication pattern, are trivial. The first step in the computation of similarity between a pair of communication patterns, is to partition both the communication patterns into splices. The next step is to find a similarity measure between a pair of splices, one taken from each of the communication patterns. This step will generate a series of similarity values, which are then combined into a single measure. This measure is then declared as the similarity measure for the pair of communication patterns.

The results published in [4] indicate that in many cases where the two communication patterns are same, the similarity measure obtained by the technique implies some difference between them. In this paper we propose a splicing technique based on causal information present in the communication patterns. Experimental results based on the proposed splicing technique show more accurate similarity measure values when the patterns are same.

Section 2 presents some definitions required for presentation of our work and also discusses the related works. Section 3 we present our motivation behind this work and in section 4 we describe the algorithm for causal cycle based splicing. Experimental results are presented and discussed upon in section 5. Section 6 presents some concluding remarks.

2 Background

In this section, first we present some definitions which are related to our work. Then we describe the basic communication pattern matching technique proposed in [4] which serves as the basis of this paper.

Definitions. An event \( e_1 \) is said to have happened before another event \( e_2 \) or, \( e_2 \) causally depends on \( e_1 \) (denoted as \( e_1 \succ e_2 \)), iff (1) \( e_1 \) and \( e_2 \) are the events in the same process and \( e_1 \) occurred before \( e_2 \) or, (2) \( e_1 \) and \( e_2 \) are the events in different processes and \( e_1 \) is the send event of a message \( m \) and \( e_2 \) is the receive event of \( m \) or, (3) there is an event \( e' \), such that \( e_1 \succ e' \) and \( e' \succ e_2 \) [1].

A causal path is an ordered set of events in a distributed system, where each event happened before on all other preceding events in the set [2].

A causal cycle is a causal path, where the first event and the last event in the path belong to the same process.

This paper is based on the work reported in [4] which is briefly presented in the following section.

2.1 Related Work

A wide variety of checkpoint and recovery protocols for distributed system has been proposed in literature [2]. Authors in [4] proposed a technique to dynamically determine a checkpoint and recovery protocol best suited for the current