An Operational Transformation Algorithm and Performance Evaluation

DU LI & RUI LI
Department of Computer Science, Texas A&M University, College Station, TX, 77843-3112, USA (E-mail: lidu@cs.tamu.edu)

Abstract. Operational transformation (OT) is an optimistic concurrency control method that has been well established in realtime group editors and has drawn significant research attention in the past decade. It is generally believed that the use of OT automatically achieves high local responsiveness in group editors. However, no performance study has been reported previously on OT algorithms to the best of our knowledge. This paper extends a recent OT algorithm and studies its performance. By theoretical analyses and performance experiments, this paper proves that the worst-case execution time of OT only appears in rare cases, and shows that local responsiveness of OT-based group editors in fact depends on a number of factors such as the size of the operation log. The paper also reveals that these two results have general implications on OT algorithms and hence the design of OT-based group editors must pay attention to performance issues.

Key words: concurrency control, consistency, group editing, operational transformation, performance evaluation

1. Introduction

Realtime group editors are a classic model of interactive groupware applications that feature coordinated manipulation of shared data (Ellis et al., 1991; Prasun et al., 1994). They are also useful tools that allow a distributed group of people to cooperatively edit a shared document at the same time. For local responsiveness, they usually replicate the shared data to hide communication latencies. Then consistency maintenance and concurrency control become critical issues.

A special optimistic concurrency control method called operational transformation (OT) (Ellis and Gibbs, 1989; Sun and Ellis, 1998) has been well accepted in this context. Conceptually, OT always allows local operations to execute in a nonblocking manner and transforms remote operations before execution such that inconsistencies are repaired. In this way, local response time is not sensitive to networking latencies. As a result, it is generally believed in the literature that the use of OT algorithms automatically achieves high local responsiveness (Ellis and Gibbs, 1989; Ressel et al., 1996; Suleiman et al., 1998; Sun et al., 1998; Sun and Ellis, 1998; Li and Li, 2004).
A number of OT-based algorithms have been proposed since 1989, such as dOPT (Ellis and Gibbs, 1989), adOPTed (Ressel et al., 1996), SOCT (Suleiman et al., 1998; Vidot et al., 2000), and GOT/GOTO (Sun et al., 1998; Sun and Ellis, 1998; Shen and Sun, 2002b; Sun, 2002). State difference transformation (SDT) (Li and Li, 2004, 2005a) is a recent OT algorithm developed in our group that is proved to achieve convergence in pure peer-to-peer group editors for the first time, to the best of our knowledge. However, our early work on SDT was only intended to solve the correctness problem while leaving out performance issues. In this paper, we present a follow-up (called SDT Optimized or SDTO) that extends SDT in the following two directions: First, SDT follows the classic simplified operation set (insert and delete) and can only be used for cooperative editing of pure texts. SDTO includes a third primitive operation (update) and hence supports formatted documents. Secondly, the time complexity of SDT is $O(n^3)$ and space complexity $O(n^2)$, where $n$ is the size of the operation log. SDTO improves the expected time complexity to $O(n^2)$ and space complexity by a large factor.

Moreover, this paper studies the performance of SDTO and argues that our findings have general implications on OT algorithms. First, we prove that the worst-case execution time of OT only happens in rare situations in which OT needs to break ties between two concurrent insert operations to address the convergence problem. Second, our experiments show that the performance (or local response time) of OT is actually dependent on a number of factors such as the ratio of operation types and the size of the operation log. The performance of concurrency control algorithms such as OT is important because interactive groupware applications such as group editors are demanded to achieve a local response time of 100 ms or less (Shneiderman, 1984; Bhola et al., 1998). The proof is also important because it justifies the optimization (for the first time to our knowledge) and improves our understanding of the causes of the convergence problem. Results in this paper inform the design of OT-based group editors in general.

The rest of this paper is organized as follows. Section 2 gives background information of group editors and the consistency model. Then Section 3 presents the SDTO algorithm and related correctness proofs. An example is described in Section 4. Section 5 analyzes the complexity of SDTO and experimental results. Section 6 compares related works. Section 7 summarizes contributions of this paper and points out future directions of research.

2. Background and models

2.1. A MODEL OF GROUP EDITORS

Group editors generally replicate the shared data (document) at each site. For simplicity and as a convention (Ellis and Gibbs, 1989; Sun and Ellis,