ML-1-2PC: An Adaptive Multi-level Atomic Commit Protocol

Yousef J. Al-Houmaily[1] and Panos K. Chrysanthis[2]

[1] Dept. of Computer and Information Programs
Institute of Public Administration
Riyadh 11141, Saudi Arabia
houmaily@ipa.edu.sa
[2] Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260, USA
panos@cs.pitt.edu

Abstract. The one-two phase commit (1-2PC) protocol is a combination of a one-phase atomic commit protocol, namely, implicit yes-vote, and a two-phase atomic commit protocol, namely, presumed commit. The 1-2PC protocol integrates these two protocols in a dynamic fashion, depending on the behavior of transactions and system requirements, in spite of their incompatibilities. This paper extends the applicability of 1-2PC to the multi-level transaction execution model, which is adopted by database standards. Besides allowing incompatible atomic commit protocols to co-exist in the same environment, 1-2PC has the advantage of enhanced performance over the currently known atomic commit protocols making it more suitable for Internet database applications.

1 Introduction

The *two-phase commit* (2PC) protocol [9,12] is one of the most widely used and optimized *atomic commit protocols* (ACPs). It ensures atomicity and independent recovery but at a substantial cost during normal transaction execution which adversely affects the performance of the system. This is due to the costs associated with its *message complexity* (i.e., the number of messages used for coordinating the actions of the different sites) and *log complexity* (i.e., the amount of information that needs to be stored in the stable storage of the participating sites for failure recovery). For this reason, there has been a re-newed interest in developing more efficient ACPs and optimizations. This is especially important given the current advances in electronic services and electronic commerce environments that are characterized by high volume of transactions where commit processing overhead is more pronounced. Most notable results that aim at reducing the cost of commit processing are *one-phase commit* (1PC) protocols such as *implicit yes-vote* (IYV) [4,6] and *coordinator log* (CL) [19].

Although 1PC protocols are, in general, more efficient than 2PC protocols, 1PC protocols place assumptions on transactions or the database management

A. Benczúr, J. Demetrovics, G. Gottlob (Eds.): ADBIS 2004, LNCS 3255, pp. 275–290, 2004.

systems (DBMSs). Whereas some of these assumptions are realistic (i.e., reflect how DBMSs are usually implemented), others can be considered restrictive in some applications [1,6]. For example, 1PC protocols restrict the implementation of applications that wish to utilize *deferred consistency constraints validation*, an option that is specified in the SQL standards.

The *one-two phase commit* (1-2PC) protocol attempts to achieve the best of the two worlds. Namely, the performance of 1PC and the wide applicability of 2PC. It is essentially a combination of 1PC (in particular, *IYV*) and 2PC (in particular, *Presumed Commit – PrC* [16]). It starts as 1PC and dynamically switches to 2PC when necessary. Thus, 1-2PC achieves the performance advantages of 1PC protocols whenever possible and, at the same time, the wide applicability of 2PC protocols. In other words, 1-2PC supports deferred constraints without penalizing those transactions that do not require them. Furthermore, 1-2PC achieves this advantage on a participant (cohort) basis within the same transaction in spite of the incompatibilities between the 1PC and 2PC protocols.

This paper extends the applicability of 1-2PC to the multi-level transaction execution (MLTE) model, the one adopted by database standards and implemented in commercial systems. The MLTE model is specially important in the context of Internet transactions since they are hierarchical in nature, making 1-2PC more suitable for Internet database applications.

In Section 2, we review PrC, IYV and 1-2PC. *Multi-level 1-2PC* is introduced in Section 3. The performance of 1-2PC is analytically evaluated in Section 4.

2 Background

A distributed/Internet transaction accesses data by submitting operations to its *coordinator*. The coordinator of a transaction is assumed to be the *transaction manager* at the site where the transaction is initiated. Depending on the data distribution, the coordinator decomposes the transaction into a set of *subtransactions*, each of which executes at a single participating database site (*cohort*).

In the *multi-level transaction execution* (MLTE) model, it is possible for a cohort, to decompose its assigned subtransactions further. Thus, a transaction execution can be represented by a multi-level execution tree with its coordinator at the root, and with a number of intermediate and leaf cohorts. When the transaction finishes its execution and submits its final commit request, the coordinator initiates an atomic commit protocol.

2.1 Presumed Commit Two-Phase Commit Protocol

Presumed Commit (PrC) [16] is one of the best known variants of the two-phase commit protocol which consist of a *voting phase* and a *decision phase*. During the voting phase, the coordinator requests all cohorts to *prepare to commit* whereas, during the decision phase, the coordinator either commits the transaction if *all* cohorts are prepared-to-commit (voted "yes"), or aborts the transaction if any cohort has decided to abort (voted "no").