Abstract

Distributed database systems implement a transaction commit protocol to ensure transaction atomicity. A commit protocol guarantees the uniform commitment of distributed transaction execution, that is, it ensures that all the participating sites agree on the final transaction outcome (commit or abort). Most importantly, this guarantee is valid even in the presence of site or network failures. Over the last two decades, a variety of commit protocols have been proposed by database researchers. These include the classical two phase commit (2PC) protocol, its variations such as Presumed Commit and Presumed Abort, nested 2PC, broadcast 2PC and three phase commit. To achieve their functionality, these commit protocols typically require exchange of multiple messages, in multiple phases, between the participating sites where the distributed transaction executed. In addition, several log records are generated, some of which have to be "forced", that is, flushed to disk immediately. Due to these costs, commit processing can result in a significant increase in transaction execution times, and therefore the choice of commit protocol becomes an important decision in the design of a distributed database system. Surprisingly, however, no systematic studies are available on the relative performance of these protocols with respect to their quantitative impact on transaction processing performance, rendering it difficult for designers to make an informed choice. In this thesis, we address this lacuna for two kinds of distributed database systems: (1) Distributed On-Line Transaction Processing Systems (OLTP), and (2) Distributed Real-Time Database Systems (RTDB). A special feature of our study is that we consider both blocking commit protocols, wherein site or network failures can lead to indefinite transaction blocking, and non-blocking protocols, wherein blocking is eliminated for most failure conditions but at
the cost of incurring extra overheads during normal processing

Distributed OLTP systems are vital to a variety of business applications that include banking, transportation, electronic commerce, etc. For these systems, *transaction throughput* is usually the major performance criterion. We study here the transaction throughput performance of a representative suite of previously proposed commit protocols by using a detailed simulation model of a distributed OLTP system. We also propose and evaluate a new commit protocol, called OPT, that allows transactions to "optimistically" borrow uncommitted data in a controlled manner. The OPT protocol is easy to implement and incorporate in current systems, and can coexist with most of the other optimizations proposed earlier. For example, OPT can be combined with current industry standard commit protocols such as Presumed Commit or Presumed Abort.

The experimental results show that distributed *commit* processing can have considerably more influence than distributed *data* processing on the throughput performance in distributed OLTP systems. Moreover, the choice of commit protocol clearly affects the magnitude of this influence. Among the protocols evaluated, the new optimistic commit protocol, OPT, provides the best transaction processing performance for a variety of workloads and system configurations. In fact, OPT's peak throughput performance is often close to that of an equivalent centralized system, and more interestingly, a non-blocking version of OPT exhibits better peak throughput performance than all of the standard blocking protocols evaluated in our study.

While OLTP systems cater to a large segment of business applications, Real-Time Database (RTDB) systems cater to applications wherein transactions have specific *timing constraints*, usually in the form of completion *deadlines*, and the goal of the system is to meet these deadlines, that is, to process transactions before their deadlines expire. Such real-time applications include aerospace and military systems, computer integrated manufacturing, stock markets, etc. Many of these applications are inherently distributed in nature. We investigate here the performance implications of supporting transaction atomicity in distributed real-time database systems. In particular, we consider applications with "firm-deadlines"—for such applications, completing a task after its deadline
has expired is of no utility and may even be harmful. Therefore, transactions missing their deadlines are immediately discarded from the system without being executed to completion. The performance goal in a firm-deadline RTDB system is to minimize "deadline miss percent", that is, the percentage of transactions missing their deadlines.

Using a detailed simulation model of a distributed firm-deadline real-time database system, we profile the real-time performance of a representative suite of previously proposed commit protocols that are customized for the real-time domain. We also present and evaluate a new commit protocol, called RT-OPT, that is similar to OPT (the optimistic commit protocol proposed in the context of distributed OLTP systems) in its basic design but incorporates additional optimizations that cater to the special features of the real-time domain.

The results obtained from these simulation experiments are similar in flavor to those obtained for distributed OLTP systems. First, distributed commit processing can have considerably more influence than distributed data processing on the deadline miss percentage. Second, the choice of commit protocol clearly affects the magnitude of this influence. Third, the new real-time optimistic commit protocol, RT-OPT, provides the lowest deadline miss percentage among the real-time commit protocols evaluated in our study for a variety of workloads and system configurations. Finally, a non-blocking version of RT-OPT exhibits better real-time performance than all of the standard blocking real-time commit protocols evaluated in our study.

In summary, the new optimistic commit protocols, OPT and RT-OPT, which permit controlled access to uncommitted data, provide significantly better transaction processing performance as compared to standard commit protocols proposed in the literature, for distributed OLTP systems and distributed RTDB systems, respectively. Moreover, they can provide the highly desirable non-blocking functionality at a relatively modest cost. Finally, they can be easily incorporated into existing systems.