Panel Discussion Reliability in Distributed Database Systems

by
Bharat Bhargava
Department of Computer Sciences
Purdue University
W Lafayette, Indiana 47906
(317) 494-6013

Why this Panel?

"System is down" is an unacceptable response for a user whose work depends on it Users access a database system via an abstraction called "Transaction" A transaction can be implemented as a database query or as program The goal of a highly reliable database systems is to provide efficient and uninterrupted transaction processing

The purpose of this panel is to discuss approaches that will allow us to build highly reliable database systems The approaches will be judged as exhibited by available performance studies and the experience of the builders of such systems In the last five years, many research papers have appeared in database journals and conferences In particular, the IEEE Computer Society's has sponsored an annual symposium "Reliability in Distributed Database and Software Systems" since 1981 which is solely dedicated to to address these issues At the same time several transaction processing systems have been introduced by companies dedicated to fault tolerance or non stop operations as their main mission

The demand for such systems is extensive Most service oriented organizations such as banks and airlines, defence applications, real-time processing systems have a need for high reliability systems

General Background

Distributed database systems increase the availability of database by replicating them on different sites But the synchronization and consistency control problems become more complex They are further complicated in case of failure of some component of the system Some of these problems are summarized as follows

a) **Incorrect transaction** Since transaction are execution of programs written by users and the programs can neither be fully proven correct or tested for all possible cases, incorrect transactions are a reality The idea of recovery block scheme where acceptance tests for the correct execution of the transaction and employment of alternate transactions appears to be a viable approach In addition, the primitives such as "undo", "redo", and compensation of transactions provide support for this problem

b) **Incomplete transaction execution** It may not be possible to complete the execution of a transaction due to many reasons For example, a hardware failure, deadlock problem, violation of database security, etc , may cause the system to abort a transaction in the middle of its execution Site failures or communication system failure are other possible reasons Once again the availability of undo operation can provide a viable solution In addition the termination protocols help in either committing or aborting the transaction across the distributed system

c) **Incorrect concurrent execution** Several transactions may run in parallel and not maintain consistency For example they may read incorrect data or write in the database in an incorrect order In other words, serializability of concurrent transactions may not hold This area has received much attention in the literature and many concurrency control algorithms have been proposed However the concurrency control problems become more complex when failures occur and the replicated copies are not available

d) **Site failure** A particular processor may cease operations due to software or hardware problems The contents of the memory may be lost or even contents of secondary storage may be effected A crash occurs when a processor stops all activities A Byzantine failure occurs when no assumptions can be made about the behavior of the faulty processor It may send inconsistent messages to other site, may act dead and revive without informing others, or it may not follow the protocols Much theoretical research has been done in the area of commitment and agreement to deal with such failures but their application to transaction processing is still to be explored

A site failure could be local to the transaction or it could be remote A local site failure will cause the halt in the transaction processing A remote site may effect the the completion of the transaction because a data value updated by the transaction resides on that site

e) **Communication System failure** If the communication links between two sites are broken, they may no longer be able to communicate In addition due to malfunction of the communication facility messages may be lost, delayed arbitrarily, or delivered in wrong order Essentially a reliable broadcast facility is needed In addition, some way of continuing the processing of transactions without blocking them is desired Optimistic protocols and the notion of degrees of commitment to avoid blocking appear to have the solution to this problem

f) **Miscellaneous Problems** We have to be able to detect and distinguish between the various types of failures before we can attempt to treat them Protocols for bypassing some of the failures and reconfiguring the system are also needed Multiple failures need to be treated

Focus of the panel

We plan to bring up the following issues during the discussion

a) Should the various subsystems for integrity, concurrency, commitment, termination, site recovery, network partitioning be layered at different levels of the system or since they use similar resources, they should cooperate at the same level?

b) Where should the software for the above system reside with respect to the operating systems? Do current operating systems provide good facilities for reliable transaction processing? What is the interdependence between distributed database systems and distributed operating systems?

c) How can the performance of the reliability subsystems be improved? What are the available performance results based on simulation/analytical/empirical studies?

d) What research issues have been resolved or are under control and which problems we should attack immediately?

Organization of the panel discussion

The panel will be formed by invited individuals who have had experience with either building reliable systems or researching them for some time Since we do not know all of such individulas, our panel members although experts will not know all the answers For this reason, we will appreciate extensive discussion from the audience

The plans are as follows

a) Prof Bharat Bhargava will give a thirteen minutes brief overview of the general research area and what has been already reported in various conferences

b) The different panel members will give their perspective on the issues that we have selected for discussion For example, Prof

Gerry Popek from UCLA will discuss the support of the distributed operating system for distributed database systems Other panelists have been contacted but could not confirm their participation at the time of writing this position statement The panelists perspectives will last for up to eight minutes Assuming four panelists it will take approximately 32 minutes

c) The rest of the time will be devoted for facts, questions, and opinions from the audience During this time the panelist will give elaborate further in response to specific questions If any member of the audience would like to show a prepared statement on the overhead projector, they may contact panel chairman before the start of panel discussion

d) Prof Bharat Bhargava will attempt to summarize the discussion in the last five minutes He will also provide a list of key references for those who may be unfamiliar with the area