E-Business Transaction in Web Integrated Network Environment

V.K. Murthy
University of New South Wales, Australia

E.V. Krishnamurthy
Australian National University, Australia

INTRODUCTION

The World Wide Web (or the "Web") is revolutionizing the concept of service -oriented computing. It permits the integration of the local area networks (LAN) and wide-area networks (WAN), thereby providing servers that are interconnected at a worldwide or an intergalactic level. Such an integration serves as a backbone for the Web-based business or e-business to access information, as well as perform e-business transactions across the globe with adequate security and reliability. In this article, we describe the issues involved in the design of an e-business transaction processing system and the solutions that have been proposed for these problems using the techniques of AI, conventional database transaction processing methodology and protocol engineering principles. These techniques will be useful for improved transaction throughput and scalability in e-commerce (Brancheau & Shi, 2001; Menasce & Almeida, 2000; Murthy, 2002).

BACKGROUND

Transaction management is a well-established research area with many successful results having been achieved so far. Transactions that have ACID (atomicity, consistency, isolation and durability) properties have traditionally been used to ensure consistent database management through atomicity (all or none) of actions as well as isolation of user actions (Elmagarmid, 1995; Krishnamurthy & Murthy, 1992). Recent advancements in transaction management have relaxed some of these traditional properties of transactions. The appropriate properties for transactions in a *Web-integrated network environment* (we use the acronym WINE, for brevity) have been studied recently by Chen and Dayal (2000), Ghezzi and Vigna (1997), and Murthy (2002). These papers also discus issues and problems that mobility brings into transaction management, and describe new methods using workflow and mobile agents and various types of new software tools currently available such as Java, Java database connectivity and CORBA (Common Object Request Broker Architecture-Dignum & Sierra, 2001). A distributed multi-database system with many autonomous and heterogeneous component databases will provide support for the management of global e-business (also known as "intergalactic") transactions and data resources in a WINE with mobile and stationary hosts.

The WINE consists of a collection of dissimilar (heterogeneous) computers (fixed host computers (FH) and mobile client computers (MC)) connected through the fixed-wired networks (such as the Ethernet), as well as wireless (mobile) networks connected together via a fixed high-speed network (Mbps to Gbps). The mobile clients (MC) are capable of connecting to the fixed network via a wireless link. Fixed host (FH) provides mobile application services and coordinates tasks to mobile hosts. MC supports query invoking and information filtering from FH to provide personal information service. Since the computers may have different computational powers and may use different representations for data we need to take care of not only the incompatibility among their representations, but also about their interoperability in using different pieces of software. Also, to achieve high performance and reliability (that provides maximal concurrency and recovery under failure), we need suitable computational models to help understand and analyze their behavior.

MAIN THRUST OF THE ARTICLE

The following subsections bring out the main thrust of the chapter, and deal with the following aspects: requirements for e-business transactions, operational models for e-business systems, intergalactic computing for e-business, new logical modes for e-business environment, communication bandwidth management, agent technology, and e-business protocols.

REQUIREMENTS FOR E-BUSINESS TRANSACTIONS

E-business transactions need to have the following properties: attribute-sensitivity, attribute tolerance, time-criticality, and time-tolerance and eventual consistency. Attribute and time tolerant, and time critical transactions arise in planning or "what-if" programming mode (subjunctive) where we execute hypothetical or pseudo-transactions to test the intention of actions for trial-error design. For example, such transactions arise in real-time transactions in e-shopping carts.

The e-shopping cart holds a record of the selection the buyer intends to buy. At any point the buyer can review the items, remove items, or change their quantity, type and brand. It is useful to have the shopping cart in place even if the buyer leaves the Web site to do something else and come back later. Such a persistent e-shopping cart is very useful for grocery shopping. This of course requires a deadline on the availability, as well as pricing, since some items can go up or down in pricing. Thus the buyer does not only control the items selected, but the shop automatically changes its current availability and pricing at the time when intention commit becomes an action commit. The time-tolerant and eventual consistency property is used in e-tailing. To handle time-criticality, for example, inconsistency on deadlines, we need to ensure that the fixed host can meet the required deadline by determining a priori whether an incoming transaction or a part of a transaction from a mobile host is schedulable within that deadline. A bounded amount of inconsistency may be introduced to finish a task within a deadline or accept inconsistency only when the transaction is about to miss the deadline. This can be specified by condition-event-action or rule based systems. Thus in e-business, the traditional transaction model needs to be replaced by a more realistic model called "workflow" between the customer and the trader (Geppert, 2000; Murthy, 1998, 2001).

OPERATIONAL MODELS FOR E-BUSINESS SYSTEMS

Many applications in e-business require several different services: file service, transaction processing, database access, and applications including graphics. Since no one system can be versatile enough to provide all these services, we need to bring in a client-server relationship among the consumer and the service provider so that the required server processes run in separate machines and the client requests any one of these or a combination of these services through a network of interconnected servers. Typically, these services are built using middleware packages that tie the client to the servers by coordinating client-server interactions. A beautiful property of the client-server relationship is that it is flexible and malleable.

In fact, the distributed applications can be split between client, as well as server by appropriately adding extra functions to either one of them, making them "fat clients" or "fat servers" or even peers so that they are equals. Hence the name peer-to peer or P2P computing is currently widely used.

As examples of fat client and fat server, we can recall the functions such as Code on Demand (COD) and Remote Evaluation (REV) used in earlier days. The COD represents a simple fat client; here, to obtain a single value from a table of data, the searching routine is transmitted from the client to the remote server. In REV we have the simplest "fat server"; here, to obtain a single value from a table of data, the searching routine is implemented in the server and this routine is activated by remote invocation or by a trigger. In fat clients the bulk of the applications run in clients, while in fat servers the bulk of the applications run in several servers. To make the task easier, the two-tier client server is replaced with three or more tiers, where the middle tier or tiers handle the application services.

For example, in the N-tier system (NTS), the middle tier is not implemented as a monolithic program, but as a collection of components that are used in a variety of client-initiated business transactions. Each such component realizes a small business function.

The three-tier and N-tier models provide for excellent security, better performance, ease of development, hardware flexibility, rich communication choices and heterogeneous functionality support (Orfali, Harkey & Edwards, 1996, 1999).

INTERGALACTIC COMPUTING FOR E-BUSINESS

To develop a suitable inter-galactic computing model for e-business the client-server needs to provide a very high bandwidth for WINE so that it provides an information highway for an electronic bazaar. This requires handling architectural heterogeneity between communicating machines. Also appropriate choice of programming paradigms and related software tools are to be made available to the mobile host so that clients can install special purpose interfaces with appropriate properties they require at the remote fixed-host.

Some key technologies necessary for these developments are:

a. ***New logical modes of computation:*** These modes support a variety of trial and error transactions, long