Anatomy of a Real E-Commerce System

Anant Jhingran
IBM T.J. Watson Research Center
Hawthorne, NY 10532
anant@us.ibm.com

ABSTRACT

Today's E-Commerce systems are a complex assembly of databases, web servers, home grown glue code, and networking services for security and scalability. The trend is towards larger pieces of these coming together in bundled offerings from leading software vendors, and the networking/hardware being offered through service delivery companies. In this paper we examine the bundle by looking in detail at IBM's WebSphere, Commerce Edition, and its deployment at a major customer site.

Keywords

E-Commerce, Middleware, Web Applications, Databases.

1. INTRODUCTION

IBM's Websphere, Commerce Edition (henceforth called WCS) is an E-Commerce offering built mostly in Java that offers the ability for businesses to set up B2B (business-to-business) and B2C (business-to-consumer) sell-side focussed sites. It is being extended so that it can be deployed in the emerging E-marketplaces, by including dynamic pricing, catalog aggregation, workflow and approvals. In order to achieve this functionality, this software must offer the following capabilities in a bundle:

1.1 Commerce Functions

- User Management: In more complex B2B sites, this translates to roles, organization and access control (e.g. a "purchaser" from "acme corporation" cannot see an item "x") and associated workflow around that. In B2C scenario, this translates to authentication, and profile management.
- Content Management, as it relates to transactions (typically catalog management) including content aggregation for e-marketplaces and distribution hubs. The second part of catalog management is the browsing metaphors – shoppers, "Mr. Know-it-all", "comparison shoppers", "matchmakers etc."
- Merchandising: Ability to do up-sell and cross-sells, both through automatic data mining and rules generation , but more typically, through administrator managed rules. Typically, real estate on the pages is allocated to advertisements, promotions and recommendations to accomplish that, and this real estate is allocated to products based on some model of customer behavior.
- Negotiations: The fixed price model of yesterday is being replaced by negotiated deals typified by auctions, reverse auctions (also known as Request for Proposal/Request for Quotes or RFP/RFQ), and general exchanges. In addition, contract based pricing is becoming increasingly popular.
- Order Fulfillment, including taxation, shipping instructions, delivery instructions etc.
- Payment Processing, including B2B payment options such as purchase orders (P.O) and lines of credits.
- Service and Support: For follow through post sales.

Over time, many of these services could be outsourced (leaving the commerce site to be the prime driver of traffic, and not the prime catcher of transactions). Examples of these that are beginning to emerge is the outsourcing of credit, transportation, logistics etc.

1.2 Underlying Infrastructure

Underneath these functions are the following:

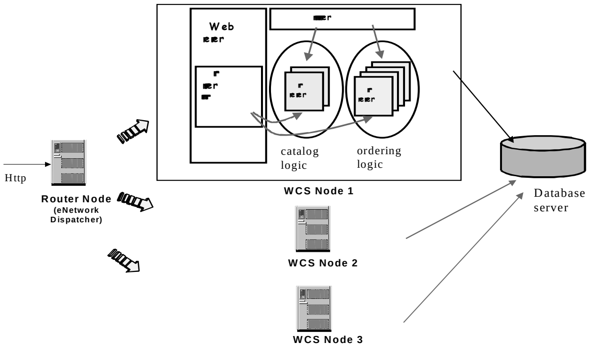
- Strong messaging layer that does protocol translation and service contracts (adherence to so-called Trading Partner Agreements, or TPA's)
- Websphere, which is an EJB application development environment with associated web servers. Typically, it also provides support for application/web page level caching (for repeated queries where database calls are too expensive) and other connection management tools.
- DB2 or Oracle, providing transactional support for the Commerce application. This support involves persistence store of all information (including catalogs, GIFs, videos, product specifications) and for all transactional pieces. In addition, decision Support through report generation, OLAP analysis (and rarely, as of now) and data mining is also provided.
- Asynchronous and synchronous connectivity to different back-end systems, including Enterprise Resource Planning (ERP) systems such as SAP/Baan/Peoplesoft, and other systems such as accounting, job floor scheduling, promotion management etc.
- Support for business rules – when "a value shopper shops, give him a 10% discount."

1.3 Typical Configuration

For 24x7 operations, elaborate efforts are made to handle load and failures. Load is typically handled at the application level (by creating more application threads) and by routing (using network dispatcher) to the free(er) nodes. In particular, care must be taken so that the "browsers/window shoppers" must not overrun the legitimate shoppers (who drive commerce transactions), and that implies quality of service within the application by keeping

MOD 2000, Dallas, TX USA
© ACM 2000 1-58113-218-2/00/05 . . .$5.00

separate queues and priorities for different parts of the commerce chain.



The size of the database is typically not very large, so the application model tends to be "parallel" application going against SMP based "database". Availability is typically handled through standard database techniques (replication, redundancy etc.) and fast restart at the application thread level.

In the figure above, we show a typical WCS installation, including the network dispatcher sending requests to different WCS server nodes (to balance load). Within a server node, different commerce functions (such as catalog browsing, order processing etc.) have a different degree of throughput requirements, managed by a controller process.

However, in addition to the application availability, sites need to maintain network availability. More robust web sites do that through elaborate set of routers and server nodes in front of the commerce site.

2. DATABASE ISSUES

Database is the repository of all information, and transactions. The transaction scope is typically the unit of recoverability. In the case of WCS, this translates to a "command" such as "add to shopping cart," "submit bid," "approve purchase order," etc. All higher level transactional semantics are maintained in the application (e.g., in the distributed transactional aspect, such as COYOTE [3]).

2.1 Information

Let us look at information at every level. A few months from now when XML support in relational databases becomes ubiquitous, this task might get simpler, but as of now, most commerce servers map information into relational tables one information at a time.

2.1.1 Catalogs

Catalogs are difficult to obviously map into relational tables. Not only are they hierarchical, but attributes vary considerably from node to node. Two obvious models exist: store in one relation, as name, value pairs (used in WCS); or store each category (with some uniformity of attributes within that category – after all men's shirts are likely to have similar attributes like collar size and sleeve length) in a separate relation. Text descriptions of products are entered in DB2 text extender, with appropriate efficient search techniques applied. Search against these catalog entries is typically handled as a series of point queries against the database tables often resulting in over 10 – 15 SQL queries for a simple act of browsing. As a result, "caching" in the application is used often.

2.1.2 User Attributes

User attributes again tend to be hierarchical and extensible, since what is relevant to Macys.com's (a retailer) will be very different from SciQuest.com (a parts marketplace). LDAP is often used for membership information (backed by a relational database), but it unfortunately falls short of the customer/user model that the commerce server needs.

2.1.3 Promotion/Merchandising Information

See [1] for some details on how e-coupons get mapped into relational attributes. For other cross-sell and up-sell, implicit or explicit 2-ary relations are used. The concept of the value shopper is typically stored as a (derived) attribute of DB2, however the rules and actions do not use DB2 trigger mechanisms. We do not use DB2 trigger mechanisms, since most of the rules are so-called select rules (i.e. to be applied on selects), and it is deemed expensive to do a dummy insert to trigger a DB2 rule. Another reason for this is "portability" – the trigger and rule language of the database is just not standardized enough to be used.

2.1.4 Decision Support

Use of OLAP analysis built around transactional data is very common. A separate copy (denormalized) of the transactional tables is used for such analysis. What is becoming more common is that people do not want to view their ".com" storefronts as separate entities – they want to make more integrated decisions across their "click and mortar" enterprise.

2.2 Transactions

2.2.1 Negotiations

See [2] for details of the objects used in e-auctions within WCS. There are two requirements for real-time bid capture – one is simple, efficient insertion into a bid table (easy) and second is efficient computation of the max (or other details), on a per user basis, so that one can generate – "you have been outbid" message. The query for this can easily get very complicated, and we use elaborate in memory data structures to keep its performance bounded. There are considerable challenges in matchmaking and exchanges, including approximate search applied against traditional databases. Typically, the queries to execute a maximal bipartite matching are complex enough that WCS fetches the rows in memory and then executes these algorithms against them.

2.2.2 Order Fulfillment

This part is simple, transaction wise. Insert into a table; commit. Two-phase commit is not used.

3. REFERENCES

[1] "Sales Promotions on the Internet," Manoj Kumar, Anand Rangachari, Anant Jhingran and Rakesh Mohan, 3rd USENIX Workshop on Electronic Commerce, 1998

[2] "Internet Auctions," Manoj Kumar and Stuart I. Feldman, 3rd USENIX Workshop on Electronic Commerce, 1998

[3] "The Coyote Project: Framework for Multi-party E-Commerce," Asit Dan et al, ECDL 1998