A Comparison of Software Architectures for E-Business Applications

Emmanuel Cecchet, Anupam Chanda, Sameh Elnikety, Juli Marguerite and Willy Zwaenepoel
Rice University – Department of Computer Science

Presented By: Stan Rajan

Dynamic Web Content

- Front-end Web server
  - Executes the application logic
  - Returns the page as an HTTP response to the client
- Back-end database
  - Stores the content of the site

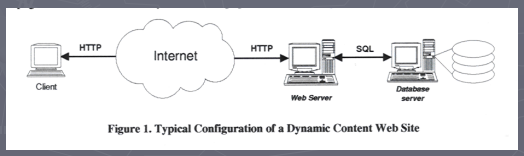

Figure 1. Typical Configuration of a Dynamic Content Web Site

Application Logic

- Provides access to the content
- Issues queries to the database
- Formats the results as an HTML page

Application Logic

- Various Forms
  - Scripting Languages
  - A Module in the Apache Web server
  - Microsoft Active Server Pages
  - Separate Java virtual machine

Application Logic

- PHP
  - Scripting language with embedded SQL queries
- Java Servlets
  - Allow embedded SQL Queries in Java code

*In both cases, the application programmer writes the SQL Queries

Application Logic

- Java Servlets with Enterprise Java Beans
  - Beans – represent items in the database
  - Java servlets call bean methods
  - Bean methods issue SQL Queries to the database

Application Logic

- Java Servlets with Enterprise Java Beans
  - Provides a level of indirection between the application and the database
  - Servlets do not have to be located on the Web Server
  - Can be located on the database machine or on an altogether separate machine

PHP (Hypertext Preprocessor)

- Scripting Language
  - An extension of the HTML language
  - Can be directly embedded into an HTML page
  - Executed within a Web server process
    - No inter-process communication overhead
  - HTTP invokes PHP interpreter that executes the script
    - Requests to the database performed using ad hoc interface

Java HTTP Servlets

- Java class that can be dynamically loaded by a servlet server
  - Runs in a Java Virtual Machine
  - Server invokes the servlet
    - Inter-process communication
    - Concurrent requests handled by separate threads
  - Servlets access database using standard JDBC interface

Enterprise Java Beans (EJB)

- Server abstracts the application business logic from underlying middleware
- Two Types of EJB
  - Entity Beans
    - Map data stored in the database
  - Session Beans
    - Perform temporary operations (stateless)
    - Represent temporary objects (stateful)

Enterprise Java Beans (EJB)

- Services provided by EJB server
  - Database access (JDBC)
  - Transactions (JTA)
  - Security (JMS)
  - Naming (JNDI)
  - Management Support (JMX)

EJB Server Example

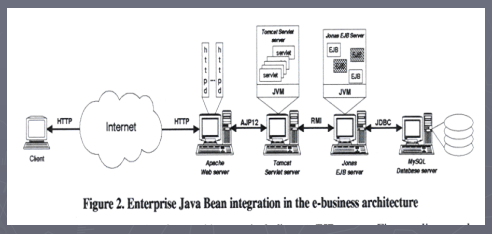

Figure 2. Enterprise Java Bean integration in the e-business architecture

1 - Client sends request to the HTTP server

EJB Server Example

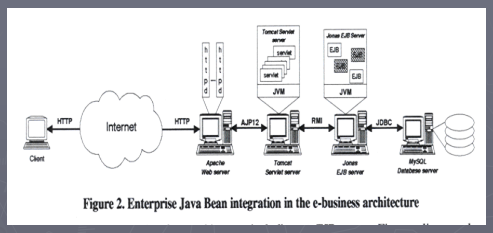

Figure 2. Enterprise Java Bean integration in the e-business architecture

2 - HTTP server invokes servlet using AJP12 protocol

EJB Server Example

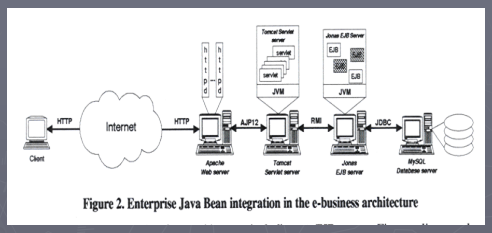

Figure 2. Enterprise Java Bean integration in the e-business architecture

3 - Servlet queries the EJB server using RMI to retrieve info from the database in order to generate the HTML reply

EJB Server Example

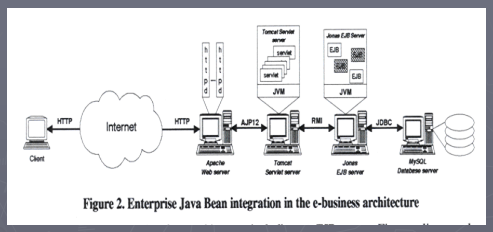

Figure 2. Enterprise Java Bean integration in the e-business architecture

4 – EJB server calls the database

PHP

- Pros
  - Easy to write
  - Reasonably efficient
  - Minimized communication overheads
    - Scripts execute in same address space as Web server
- Cons
  - Database interfaces are ad hoc
    - New code needed for each new database access

Java Servlets

- Pros
  - Easily portable between databases (JDBC interface)
  - Servlet server can be placed on a machine different from the Web server to balance the load
- Cons
  - Overhead of the JVM
  - Cost of IPC with servlet and Web server executing on separate machines

EJB

- Pros
  - Abstracts the application logic from any specific platform, protocol, or middleware infrastructure
  - Level of indirection between the application and the database
- Cons
  - Seen in the application benchmarks

Auction Site

- Significant load on the Web server
- Functionality
  - Selling, browsing, bidding
- User Sessions
  - Visitor (only allowed to browse)
  - Buyer, seller (require registration)

Auction Site

- Buyer Session
  - Bid on items
  - Summary of current bids
  - Rating and other users comments
- Seller Session
  - Require a fee before selling an item
  - Specify a minimum price

Auction Site

- Seven database tables
  - Users, items, bids, buy_now, comments, categories, regions
- 26 client web browser interactions
  - Bidding, buying, selling, leaving comments, etc.

Auction Site

- Workload Mix
  - 33,000 items for sale
  - 40 categories
  - 62 regions
  - 500,000 auction history (old-items)
  - 10 bids per item average
    - 330,000 bids table entries

Auction Site

- Workload Mix
  - 1,000,000 users table entries
  - Feedback for 95% of transactions
    - 31,500 new-comments table entries
    - 475,000 old-comments table entries
- Total Size = 1.4 GB

Online Bookstore

- Significant load on the database
- Eight database tables
  - Customers, address, orders, order_line, credit_info, items, authors, countries
- 14 Interactions
  - 6 read-only
  - 8 cause database updates

Online Bookstore

- Payment gateway emulator (PGE)
  - Represents an external system that authorizes payment of funds during purchasing
  - Web server contacts the PGE using an SSL session to send the credit card information
  - PGE replies with an authorization number

Online Bookstore

- Workload Mix
  - 95% read-only scripts
  - 80% shopping mix
  - 50% ordering mix
- Database sizes
  - 350 MB
  - 3.5 GB

3-Phase Experiment

- Warm-up phase
  - Initializes the system until it reaches a steady-state throughput level
- Steady-state phase
  - Measurements performed
- Cool-down phase
  - Slows down the incoming request flow
- Auction – 5 minutes, 30 minutes, 5 minutes
- Bookstore – 1 minute, 10 minutes, 10 seconds

Configurations

- PHP on same machine as the Web server
- Java servlets on same machine as Web server
- Java servlets on same machine as database
- Java servlets on dedicated machine
- Web server, servlet server, EJB server, and database server each on different machines

Auction Site Results

- Configurations using PHP and Java servlets Browsing

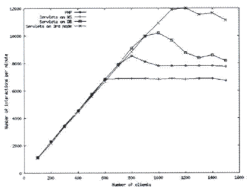

Figure 3. Auction site throughput in interactions per minute as a function of number of clients for the browsing mix using PHP and Java servlets.

Java servlets on DM

Java servlets on DB

PHP

Java servlets on WS

Auction Site Results

- Configurations using PHP and Java servlets Browsing
  - Java servlets on Web server
    - 6,840 interactions per minute for 700 clients
  - PHP
    - 8,520 interactions per minute for 800 clients
  - Java servlets on database machine
    - 10,200 interactions per minute for 1,000 clients
  - Java servlets on dedicated machine
    - 12,000 interactions per minute for 1,200 clients

Auction Site Results

- Configurations using PHP and Java servlets Bidding

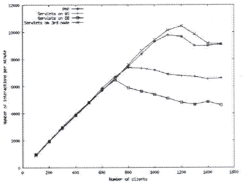

Figure 5. Auction site throughput in interactions per minute as a function of number of clients for the bidding mix using PHP and Java servlets.

Java servlets on DM

PHP

Java servlets on WS

Java servlets on DB

Auction Site Results

- Configurations using PHP and Java servlets Bidding
  - Java servlets on database machine
    - 6,480 interactions per minute for 700 clients
  - Java servlets on Web server
    - 7,380 interactions per minute for 700 clients
  - PHP
    - 9,780 interactions per minute for 1,100 clients
  - Java servlets on dedicated machine
    - 10,440 interactions per minute for 1,200 clients

Auction Site Results

- Configurations using PHP and Java servlets
  - Servlet running on the Web Server
    - Web server CPU is the bottleneck with 100% CPU util.
    - PHP is more efficient than Java servlets
      - Due to separate process for communication between Web server and servlet server which is not needed in PHP
  - Servlet running on the database machine
    - Database machine CPU is not a bottleneck
      - Util. substantially lower in browsing than bidding mix
      - 21% browsing, 45% bidding

Auction Site Results

- Configurations using PHP and Java servlets
  - Servlet running on a dedicated machine
    - Best performance achieved for both mixes
    - Benefit of an extra CPU outweighs the extra communication costs of the servlet server being on a separate machine

Auction Site Results

- Configurations using PHP, Java servlets and EJB

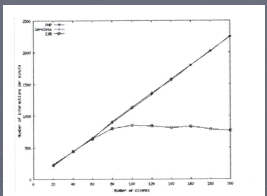

Figure 7. Auction site throughput in interactions per minute as a function of number of clients for the browsing mix using PHP, Java servlets and EJB.

Browsing

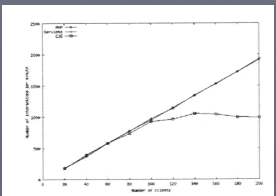

Figure 8. Auction site throughput in interactions per minute as a function of number of clients for the bidding mix using PHP, Java servlets and EJB.

Bidding

Auction Site Results

- Configurations using PHP, Java servlets and EJB
  - EJB throughput initially grows linearly with # of clients
  - Peaks at 850 ipm with 100 clients (browsing)
  - Peaks at 1,051 ipm with 140 clients (bidding)
  - Throughput of PHP and Java servlets continues to increase beyond peaks of EJB

Auction Site Results

- Configurations using PHP, Java servlets and EJB

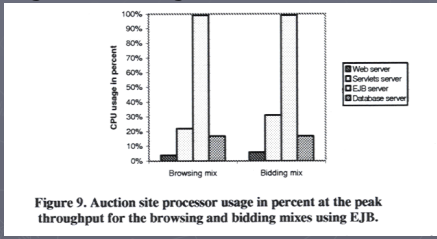

Figure 9. Auction site processor usage in percent at the peak throughput for the browsing and bidding mixes using EJB.

CPU on the EJB server is the bottleneck with average 99% util for both mixes

Bookstore Results

- Configurations using PHP and Java servlets

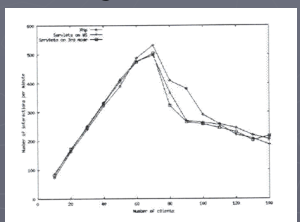

Figure 10. Online bookstore throughput in interactions per minute as a function of number of clients using PHP and Java servlets.

Bookstore Results

- Configurations using PHP and Java servlets
  - Throughput about the same for all 3 configurations
  - Java servlets on Web server
    - 497 interactions per minute for 70 clients
  - Java servlets on dedicated machine
    - 504 interactions per minute for 70 clients
  - PHP
    - 532 interactions per minute for 70 clients

Bookstore Results

- Configurations using PHP and Java servlets

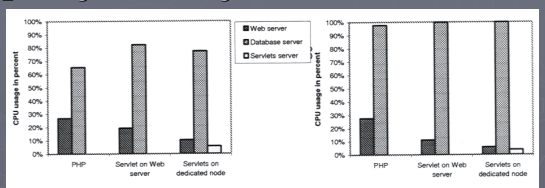

Figure 11. Online bookstore processor usage in percent at the peak throughput using PHP and Java servlets.

Figure 13. Online bookstore processor usage in percent with 90 clients using PHP and Java servlets.

Bottleneck is the database CPU

Bookstore Results

- Configurations using PHP, Java servlets and EJB

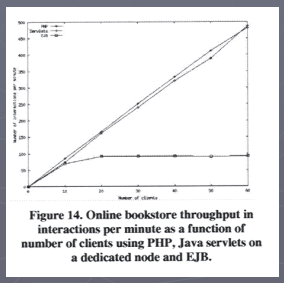

Figure 14. Online bookstore throughput in interactions per minute as a function of number of clients using PHP, Java servlets on a dedicated node and EJB.

EJB version peaks at 92 ipm with 20 clients

Bookstore Results

- Configurations using PHP, Java servlets and EJB

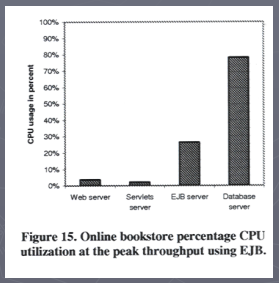

Figure 15. Online bookstore percentage CPU utilization at the peak throughput using EJB.

Both the database and the EJB servers CPU are alternatively the bottleneck resources limiting the throughput to under 100 ipm

Bookstore Results

- Database CPU is the main bottleneck
  - Minimal differences between PHP and Java servlets in performance
- EJB throughput problematic
  - Overall performance considerably lower than with PHP or Java servlets

Conclusions

- Web server and business logic on same machine
  - PHP somewhat faster than Java servlets
- Web server or business logic bottleneck
  - Re-locate Java servlets to database machine or separate machine to increase performance over PHP
- Database is the bottleneck
  - No difference between PHP and Java servlets
- In all cases
  - EJB considerably slower than PHP and Java servlets