J2EE and .Net Reloaded
Yet Another Performance Case Study

The Middleware Company Case Study Team

Presented by Mark Grechanik

The Goal of This Study

- This is an experience report that compares J2EE and .Net performances
- What makes this study special?
  - Economic aspects: billions of dollars involved
  - Religious beliefs: *"Microsoft is an evil empire"*
  - The complexity of these platforms
- A fundamental question that this study answers is what technology is better

The Goal of This Study

- This is an experience report that compares J2EE and .Net performances

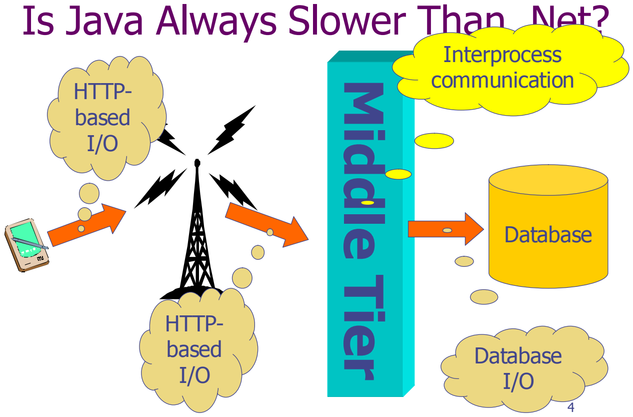
.NET or J2EE?

ved

pire"

- A fundamental question that this study answers is what technology is better

Is Java Always Slower Than .Net?



It Is a Difficult Question!

- How to compare two functionally rich platforms?
  - Benchmarks?
  - Experience reports?
  - Word of mouth?
- Can we compare .Net and J2EE independently of the application software?

Specification

- Specifies an enterprise application that represents typical customer requirements
  - Created by the Middleware company with the help of Expert group and many other people
  - Does not contain or specify any code
  - Allows this application to be implemented using any languages and run on different platforms
- Serves as a foundation for a number of case studies
- Purely functional, describes the behavior
- Other specs are technology specific

History

- Performance case study was done in October 2002
  - Showed that .Net performed better than J2EE
  - Generated enormous publicity
  - J2EE vendors were not invited
  - Microsoft reimbursed expenses
- Case study vs. benchmarks
  - Bechmarks invoke emotional responses
  - Too much formal overhead with benchmarks
  - Case studies are less formal, more collaborative, and open

Existing Benchmarks

- TPC-W specification
  - Precise
  - Provides the freedom in choosing platforms
  - All results use C++ with ISAPI DLLs
- SPECjAppServer
  - Compares only J2EE applications
  - Difficult to compare variations of used technologies

Petstore Application

- Has web-based front end that allows customers to buy pets
- Original Petstore sample application was released by Sun Microsystems
  - Not a specification
  - Was not meant to be used in performance comparison
- Raised to the specification level by the Middleware company

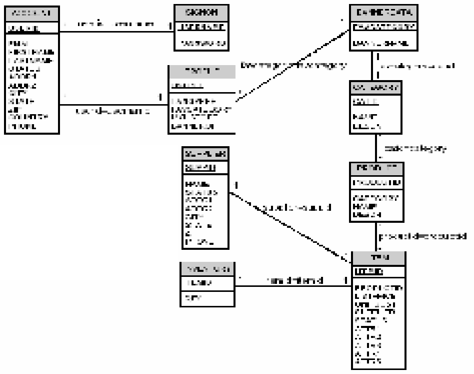
Why PetStore?

- It represents a well-understood data-driven Web application that exercises the commonly used features of application servers
- It represents functionality that customers commonly implement in their own Internet and intranet-based Web applications

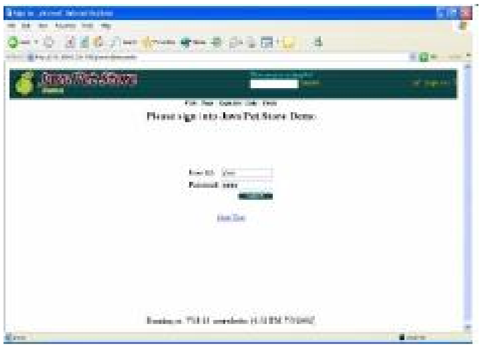
PetStore's Functionality

- Thin client HTML UI layer
- Server-side script pages (JSP or ASP.NET) to generate HTML on the server
- Data access
- Middle-tier components
- Ad-hoc database searching
- Database transactions
- Middle-tier data caching
- User/Web session management
- Web Services
- Forms-based authentication

Petstore Database Schema

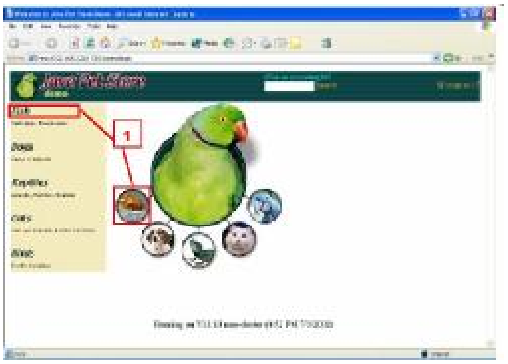


Sign-On Page

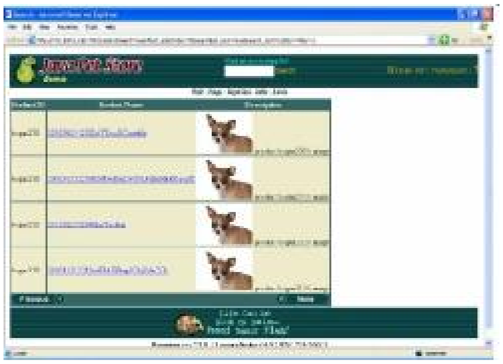


Please sign into Java Pet Store Demo

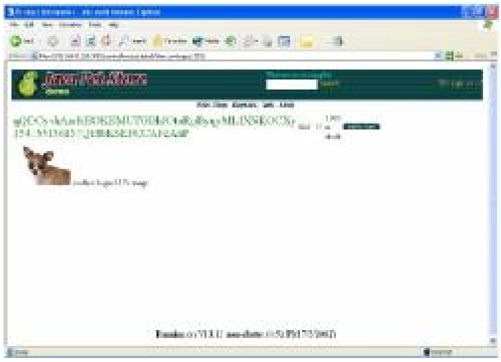
Main Page



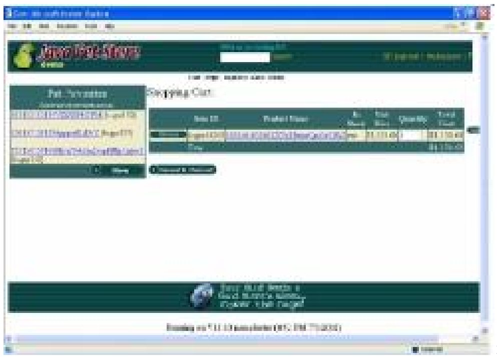
Product Search Page



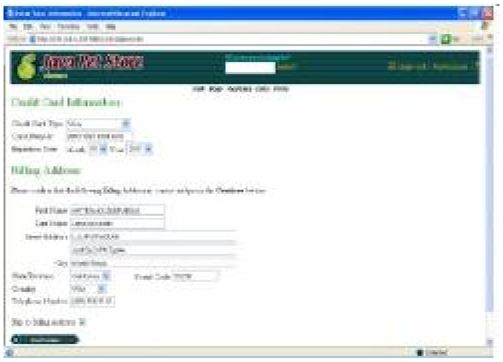
Item Detail Page



Shopping Cart



Billing Info Page



Basic Performance Comparisons

- **Web Application Performance Comparison**
  - Run with image download off, at varying user loads on an 8 CPU application server configuration
- **24-hour Transaction Performance Comparison**
  - Run with image download off, at product determined sustainable throughput user load for 24 hours of straight order placement on a 4 CPU system
- **Web Services Performance Comparison**
  - Direct activation and remote client/proxy activation via two separate Mercury Scripts
  - Run on 8 CPU web service host/application server configurations for direct activation test
  - Run on 8 CPU web service host with 2, 4 and 8 CPU web service client machine configurations

Test Databases

- Submissions must use the existing database schemas for both Oracle 9i and SQL Server 2000
- Submissions in each category and on each platform will be tested on both databases
- The implementations of the performance comparison application for each database may differ slightly, for example, in the query syntax and data access logic/driver usage
- For each submission and vendor’s product to work across two different mainstream databases is also put to the test
- The database schemas cannot be changed.

Rules

- No exclusive access to the database
- Certain data can be cached for a certain period of time
- Cache size must be finite
- Cache item removal algorithm must be described
- A clustered deployment should have the same logical behavior as a non-clustered one

Distributed Transactions

- Application code must be blind to location of both databases
  - The golden rule is that you cannot modify application code
- The Web Application performance comparison does not use distributed transactions
  - Both the Customer/Product and Orders tables are located on the same database

Session Management

- All users access the application on a forms-based authentication approach
- On sign on, a user session is created
  - All shopping cart and account information must be stored in middle-tier session state
  - This info is tracked by a unique identifier
- Multiple sessions allowed for a single user
- Sessions must automatically expire after a 10 minute inactive period

Entity Beans

- An entity bean models a business entity and performs actions within a business process
  - An entity bean to retrieve and perform computation on items within a purchase order
  - The data is stored by the container in some form of data storage system, such as a database
  - The data survive beyond the end of an application session, even a server crash or a network failure
- Persistence mechanisms
  - Bean-managed persistence (BMP)
  - Container-managed persistence (CMP)

Container Managed Persistence

- The container handles persistence tasks automatically
  - Keeps an entity bean class separate from its persistent representation
  - Enables developers to change a bean's data source without affecting its implementation
- Describe CMP declaratively in the deployment descriptor file
- When a bean is deployed, the container provider's tools parse the deployment descriptor and generate code to implement the underlying classes
- At runtime, the container manages the bean's data by interacting with the datasource (e.g. a relational database designated in the deployment descriptor)

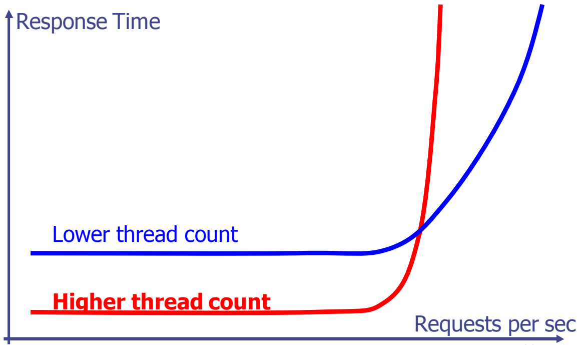
Types of Applications

- J2EE-EJB-CMP
- J2EE-SERVLET-JSP
  - No EJBs are used explicitly
- .Net-C#

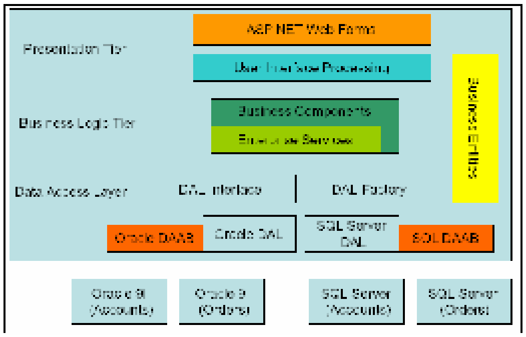
Tuning

- JVM
  - Memory
  - Garbage collector
  - CPU affinity
- Application server's runtime settings
  - Execution threads
  - Logging
  - JDBC
- Application's deployment characteristics
  - Stateful session EJBs
    - Cache sizes
    - Idle times
  - Entity Beans
    - Cache and pool sizes

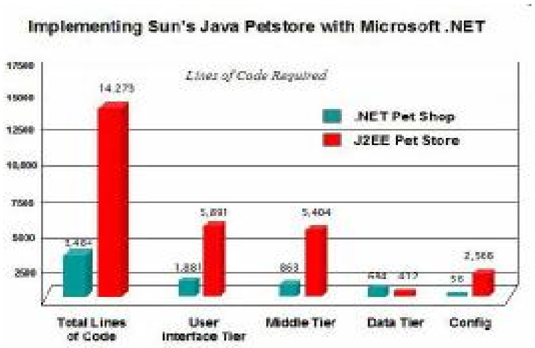
Execution Threads



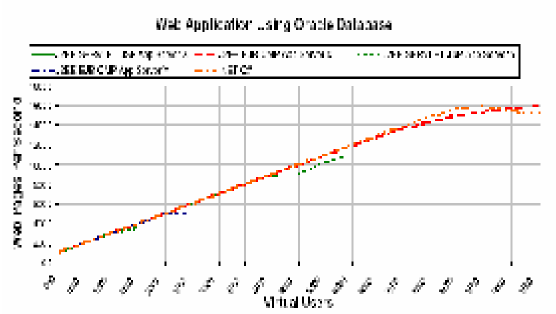
Architecture Diagram of the New .NET-C# Codebase



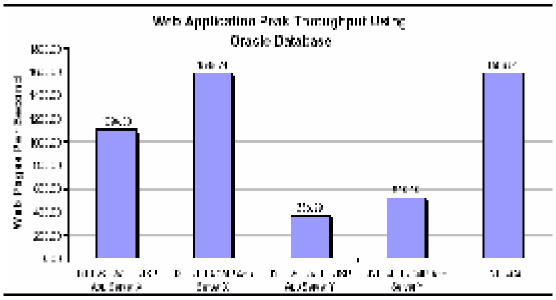
Lines of Code



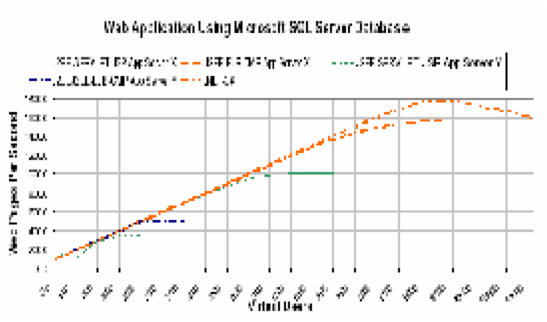
Throughput, web pages per second, increases as user load increases



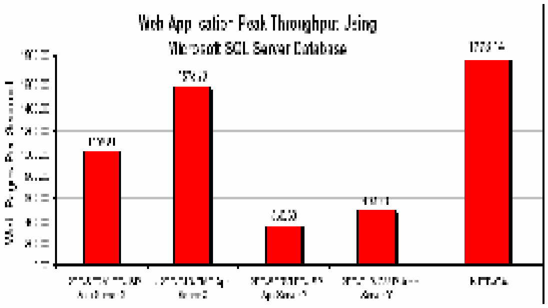
The Maximum Throughput Achieved During the Web Application Tests Using Oracle 9i Database



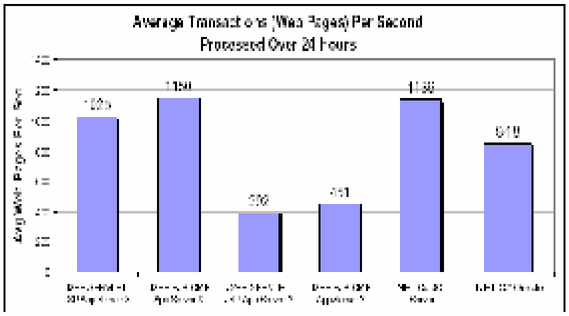
Throughput Increases As User Load Increases Running the Web Application Codebases Using Microsoft SQL Server 2000 Database



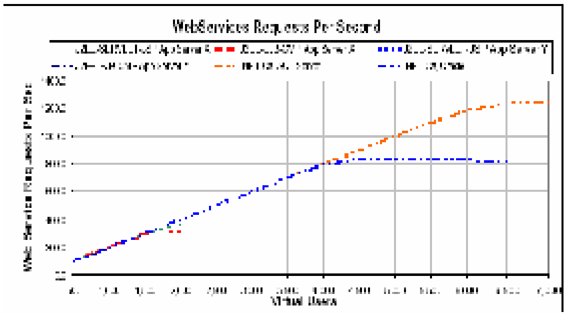
The Maximum Throughput Achieved Using Microsoft SQL Server 2000 Database



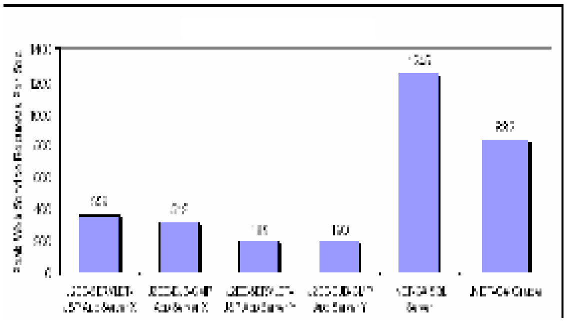
The Average Number of Web Pages Processed Per Second Over a 24-hour Period



The Throughput Increases as User Load Increases



Web Services Peak Throughput



Conclusions

- Web Application Test
  - .Net ≈ fastest J2EE platform
  - .Net is better by 11% when used with MS SQL Server
  - J2EE is better by 2% when used with Oracle 9i
- 24 Hour Reliability Test
  - .Net ≈ fastest J2EE platform (less than 2%)
- Web Services Test
  - .Net ≥ fastest J2EE platform by over 200%

Questions

- Does it fully conform to the Solomon empirical study definition?
- What is the most important lesson learned by doing this study?
- Can it serve as a precedent to form an independent consortium doing similar studies?
- How can each tested platform be improved based on the results of this study?