A Study of the Performance on EJB Entity Bean with Value Object

Ms. Eun Hee Choi, danke1@selab.ssu.ac.kr
Mr. Doe Kyun Jung, iqzero@selab.ssu.ac.kr
Dr. Nam Yong Lee, lylee@computing.ssu.ac.kr

Software Engineering Lab.
Dept. of Computer Science
Soongsil Univ.

CONTENTS

INTRODUCTION

Research Purpose

- Examine how Value Object can improve the performance of Entity Bean
  - Examine the relationships where come from kinds of situation, querying Inquire SQL Query in a single table.
  - Kinds of Situation
    - Relationship between CMP Entity Bean and CMP Entity Bean with Value Object
    - Relationship between BMP Entity Bean and BMP Entity Bean with Value Object
    - Relationship between CMP Entity Bean with Value Object and BMP Entity Bean with Value Object
- Examine how Value Object can transfer more data in fewer remote calls
  - Examine the relationship between the data size of a value object and the number of remote calls of Entity Bean.

INTRODUCTION

Research Scope

- Value Object of J2EE patterns suggested by SunMicrosystems
- Performance Testing for Entity Bean with Value Object
  - Experiment Design : IMPS(Internet Medicine Prescription System)
    - Doctor CMP Entity Bean
    - Doctor BMP Entity Bean
    - Doctor CMP Entity Bean with DoctorValue value object
    - Doctor BMP Entity Bean with DoctorValue value object
  - Relationship Graph
    - X axis : Number of SQL Queries
    - Y axis : Time (sec)

INTRODUCTION

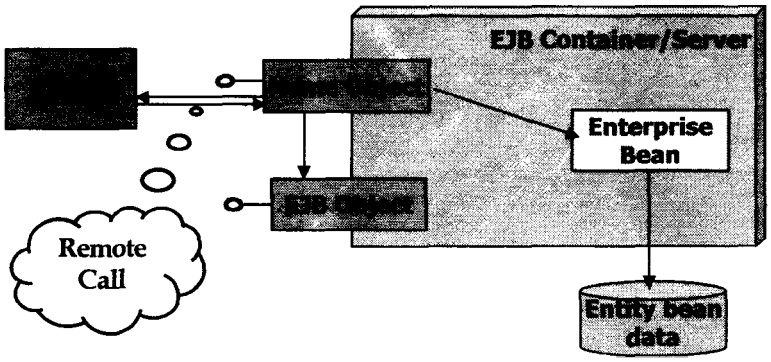
■ Research Method

- Experimental Design
  - When Entity Bean is deployed and Client request to inquire a specific information of Doctor Table, we experiment Total Time for Query Execution according to Time Measurement Operation in Client code.
  - We can apply the statistics for the experiment to the design of Entity Beans.

INTRODUCTION

■ Research Design

- In an EJB 1.1 spec, every method call made to the business service object, be it an entity bean or a session bean, is potentially remote call.

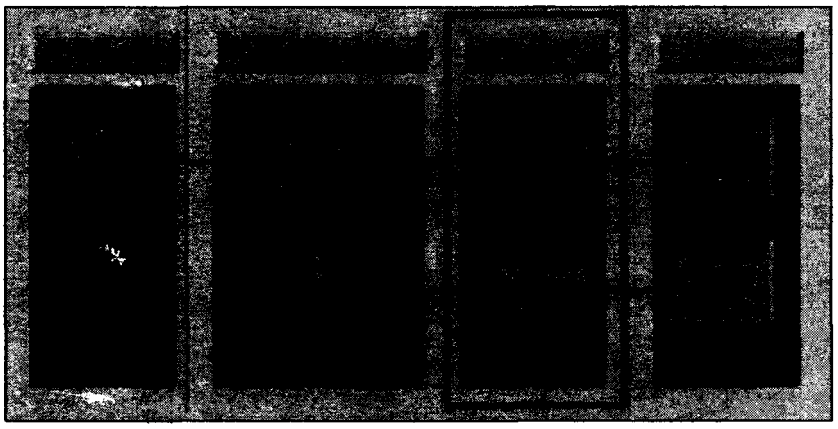


INTRODUCTION

- **Research Design**
  - Such remote invocations use the network layer regardless of the proximity of the client to the bean, creating a network overhead.
  - Using multiple calls to get methods that return single attribute values is inefficient for obtaining data values from an enterprise bean.
  - The client updates the data in the business tier much less frequently than it reads the data.

VALUE OBJECT

- **Structure in Value Objects**



VALUE OBJECT

Definition of Value Object

- One of J2EE patterns suggested by Sun microsystems
- A value object is an plain serializable java class that represent a snapshot of some server side data.
- The value object contain and encapsulate bulk data in one network transportable bundle.
- Typically, a value object is defined as public, thus eliminating the need for get and set methods.

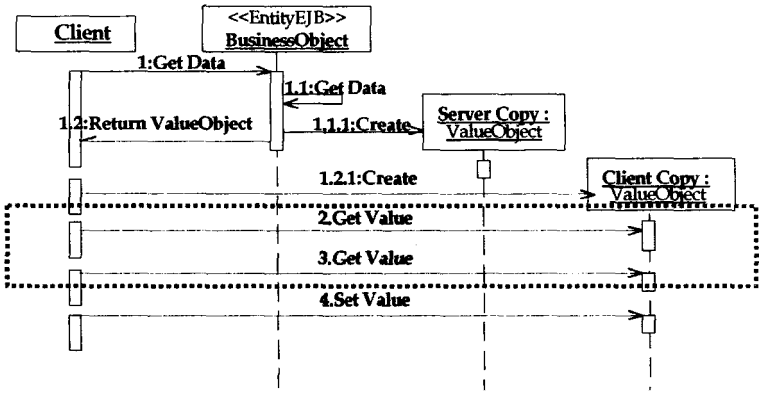
VALUE OBJECT

Feature of Value Object

- The client makes a single remote method invocation to the enterprise bean to request the value object instead of numerous remote method calls to get individual attribute values.
- Because the value object is passed by value to the client, all calls to the value object instance are local calls instead of remote method invocations.
- The value object not only carries the values from the Entity Bean to the client, but also can carry the changes required by the client back to the Entity Bean.

Performance Testing of the Entity Bean

Sequence Diagram of Value Object



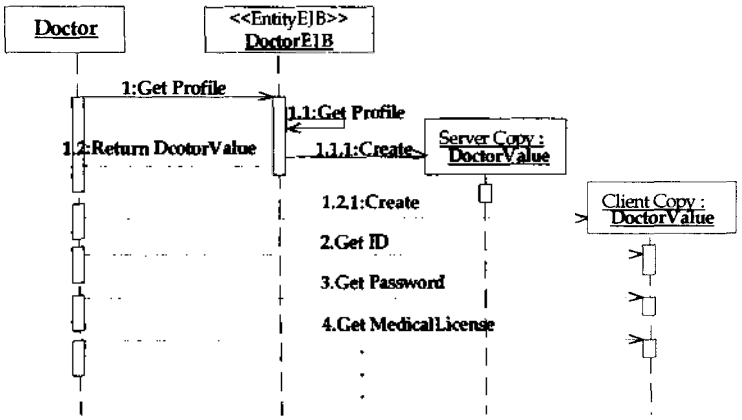
Performance Testing of the Entity Bean

Scenario

- In Internet Medical Prescription System, the Doctor(actor) must offer the required profile to become a IMPS member.
- The required profile items consist of ID, Name, Password, Address, E-mail, Medical License, etc.
- DoctorValue is the value object that accepts all attribute values of Doctor entity bean.

PERFORMANCE TESTING OF THE ENTITY BEAN

- Sequence Diagram of DoctorValue



TESTING OF THE ENTITY BEAN

- Testing Environment
  - DBMS
    - Oracle 8i Version 8.1.6
  - Application Server
    - BEA WebLogic Server 6.0
  - Network Environment
    - T3
  - Testing PC
    - Intel Pentium III
    - 1CPU, 384MB Ram, 10GB HDD
    - Windows 2000 Pro

Testing Code of Client

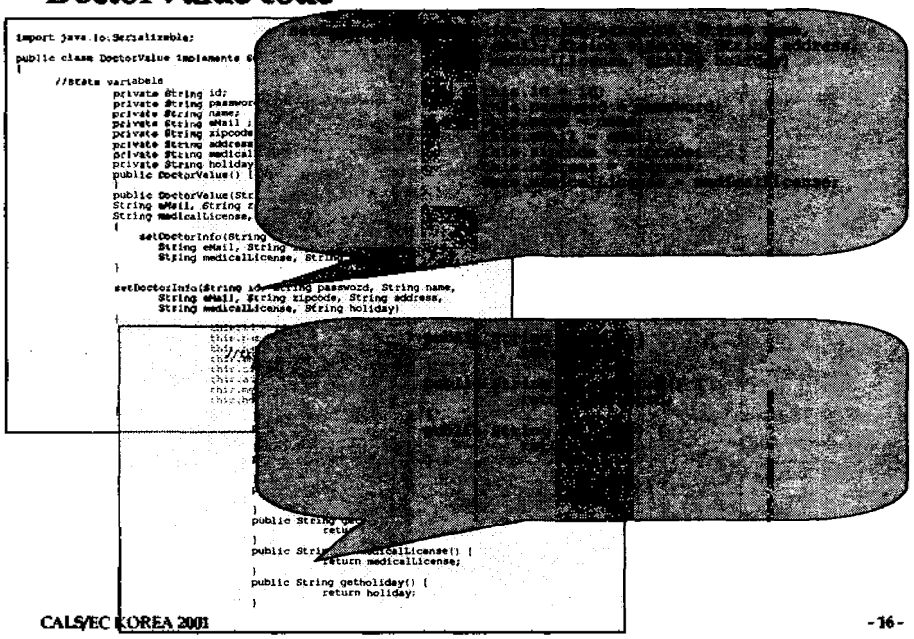
```
long startTime = System.currentTimeMillis();

// . . .
//
// Client's Business Logic
//
// . . .

long stopTime = System.currentTimeMillis();

System.out.println("Average Ping = " +
  Float.toString((float)(stopTime-startTime)+"sec");
```


DoctorValue code



Performance Testing of the Entity Bean

■ Testing Result – Outline Table

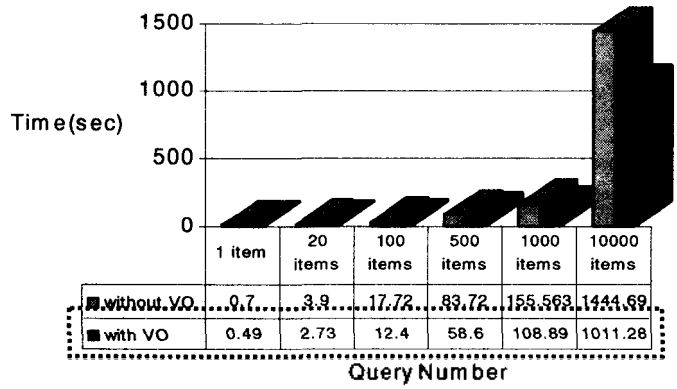
(unit: sec)

Type	CMP EntityBean		BMP EntityBean	
Table Type	without VO	with VO	without VO	with VO
1 item	0.7	0.49	0.56	0.42
20 items	3.9	2.73	3.12	2.32
100 items	17.72	12.4	14.18	10.54
500 items	83.72	58.6	66.98	49.81
1000 items	155.563	108.89	124.45	92.56
10000 items	1444.69	1011.28	1155.75	859.59

* VO : Value Object

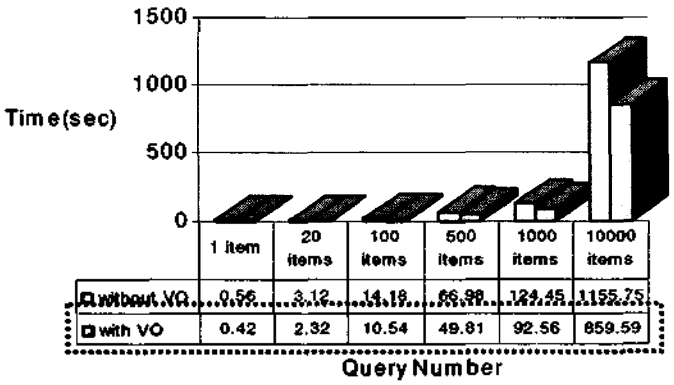
Performance Testing of the Entity Bean

■ Testing Result – CMP Entity Bean



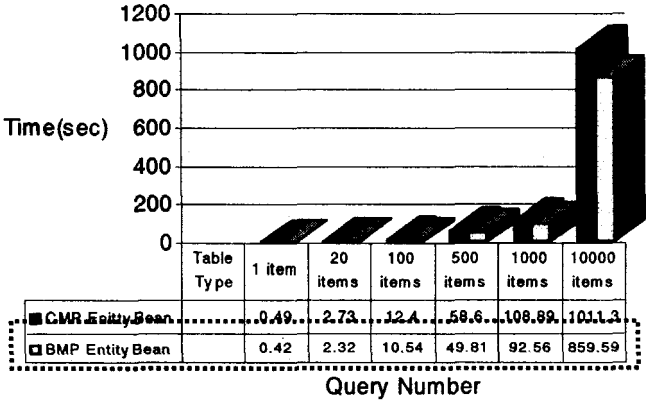
Performance Testing of the Entity Bean

- **Testing Result – BMP Entity Bean**



Performance Testing of the Entity Bean

- **Testing Result – CMP with VO vs BMP with VO**



FINDINGS & CONCLUSION

Findings

- Improved the performance of Entity Bean:
  - Because value object acts as a data carrier and reduces the number of remote network method calls, it improve the performance.
  - Implications of the testing results:
    - In both CMP Entity Bean and BMP Entity Bean, it takes less the total time for the client query execution with Value Object than without Value Object.
    - In BMP Entity Bean with Value Object, it takes less the total time for the client query execution than CMP Entity Bean with Value Object.
  - Consequently, we can expect to improve the performance of Entity Bean with Value Object.

FINDINGS & CONCLUSION

Findings (cont'd)

- Transfers More Data in Fewer Remote Calls:
  - One method call returns a greater amount of data to the client than each 'get method'.
  - Implications of the testing results:
    - Original Doctor Entity Bean contains 15 'get methods'. On the other hand, Doctor Entity Bean with Value Object contains 2 'get method,s' required to call DoctorValue Value Object.
    - Original Doctor Entity Bean contains 1 Field of Doctor DB Table, but Doctor Entity Bean with Value Object contains 15 Fields of Doctor DB Table.
  - Consequently, we expect to transfer More Data in Fewer Remote Calls with Value Object.

Findings(cont'd)

- Simplifies the design of Entity Bean and Remote Interface:
  - Entity bean provides a `getData()` method to get the value object containing attribute values.
  - Value Object technique can eliminate some of 'get methods' to implement the entity bean and remote interface

Further Researchable Areas

- About Performance Testing between multiple Entity Beans with multiple Value Objects
- About the Performance Difference between Entity Bean for Read Transaction and Entity Bean for Update Transaction
- About applying the Performance Testing Results of Entity Bean with Value Object in the Enterprise JavaBeans Specification, Version 2.0

REFERENCE

- Deepak Alur, John Crupi, Dan Malks, Core J2EE Patterns, Sun Microsystems, 2001
- Enterprise JavaBeans Specification, version 2.0, Sun Microsystems, 2001
- Ed Roman, Mastering Enterprise JavaBeans, WILEY, 2000
- http://theserverside.com/
- http://www.devx.com/upload/free/features/entdev/