



University College Dublin
An Coláiste Ollscoile,
Baile Átha Cliath

Performance Modeling and Prediction of Enterprise JavaBeans with Layered Queuing Network Templates

Jing Xu
Murray Woodside
Carleton University

Alexandre Oufimtsev
Liam Murphy
University College Dublin

Motivation and Approach

- **Motivation:**

- Performance analysis should be done in early stage of design (otherwise you going to pay for it...)
- Modeling J2EE is non-trivial

- **Approach:**

- Define LQN templates for different types of EJBs
- Construct a model of real-life application by instantiating the templates and composing them
- Profile and calibrate the model from app. traces
- Perform measurements on the real running application
- Compare model prediction with measurement results

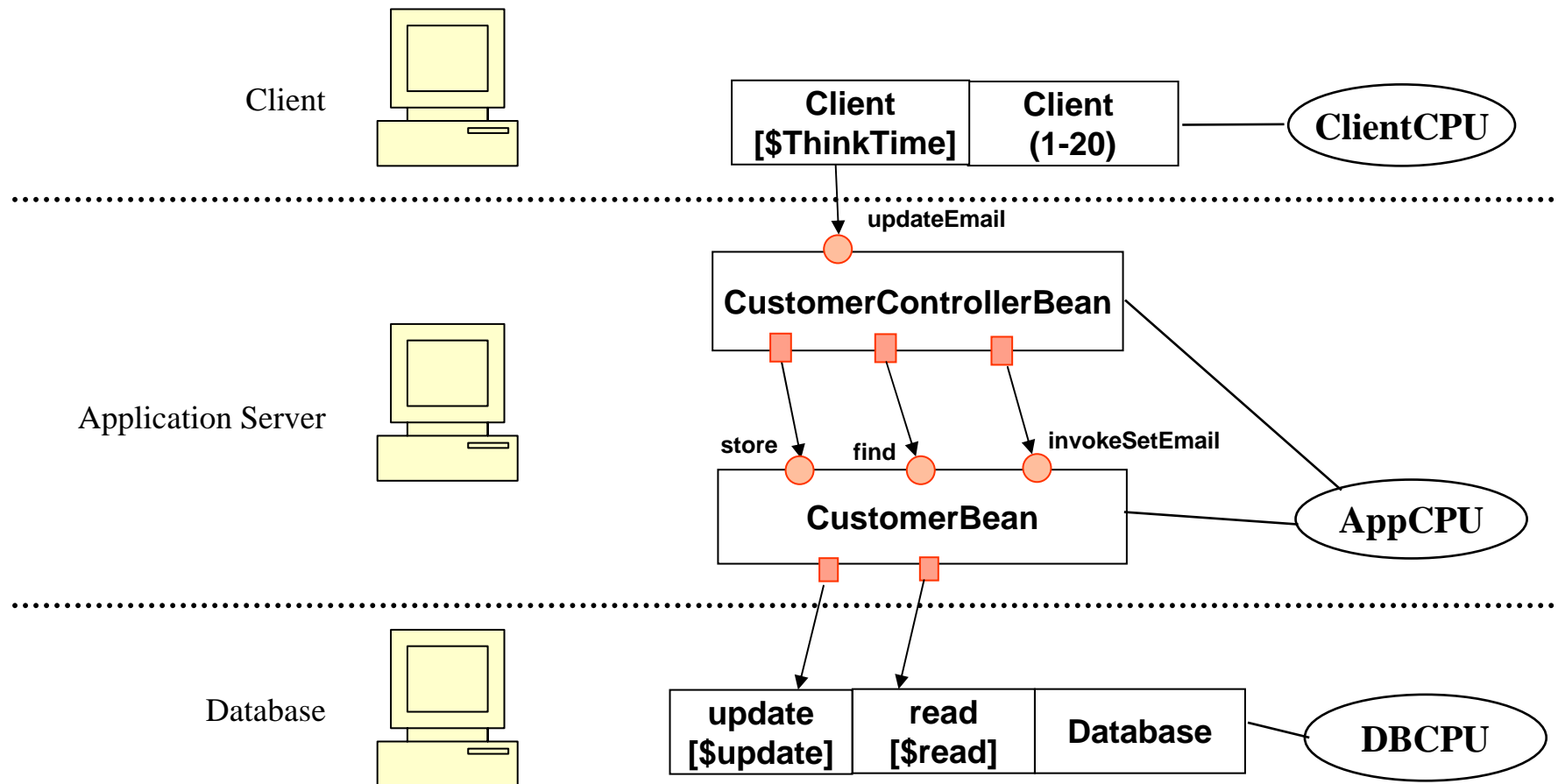
LQN: what and why?

- **LQN (Layered Queuing Network)**
 - **Is a performance modeling language**
 - **Models system resources and behaviour in an intuitive way**
 - **Allows nested software structure and composition with component concepts**
 - **Captures resource contentions effectively**
 - **Does not suffer from state explosion problem**
 - **Provides Analytical & Simulation solver**

Ref on LQN

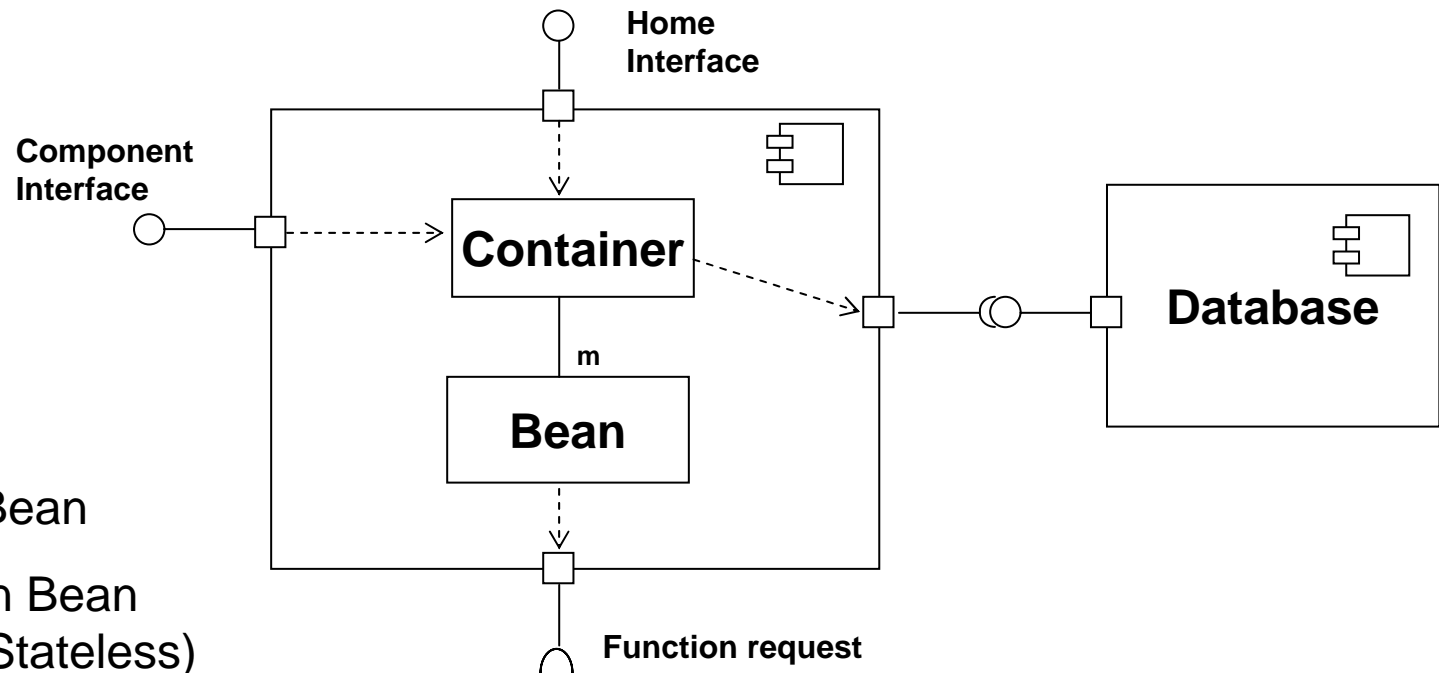
<http://www.sce.carleton.ca/rads/lqn/lqn-documentation/>

A LQN model of a 3-tier System



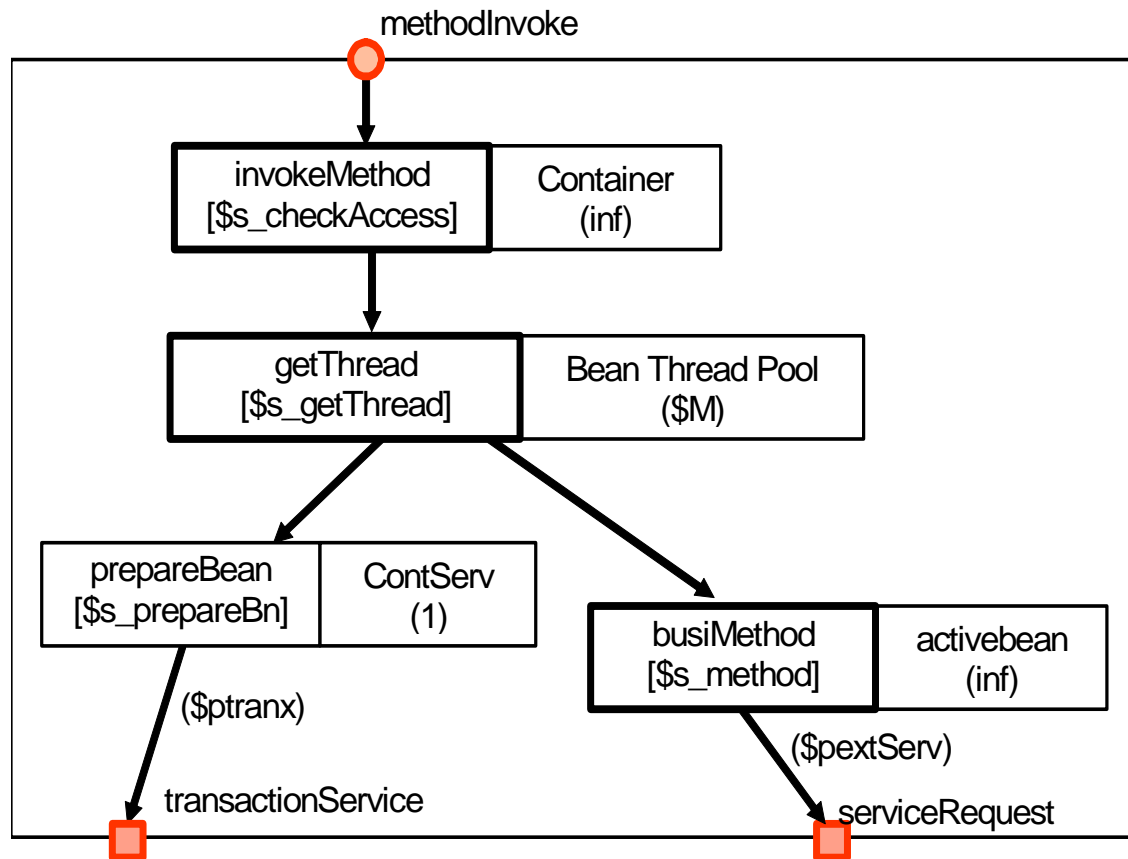
Introduction of EJB

- **Core technology of J2EE**
- **Server-side component architecture**

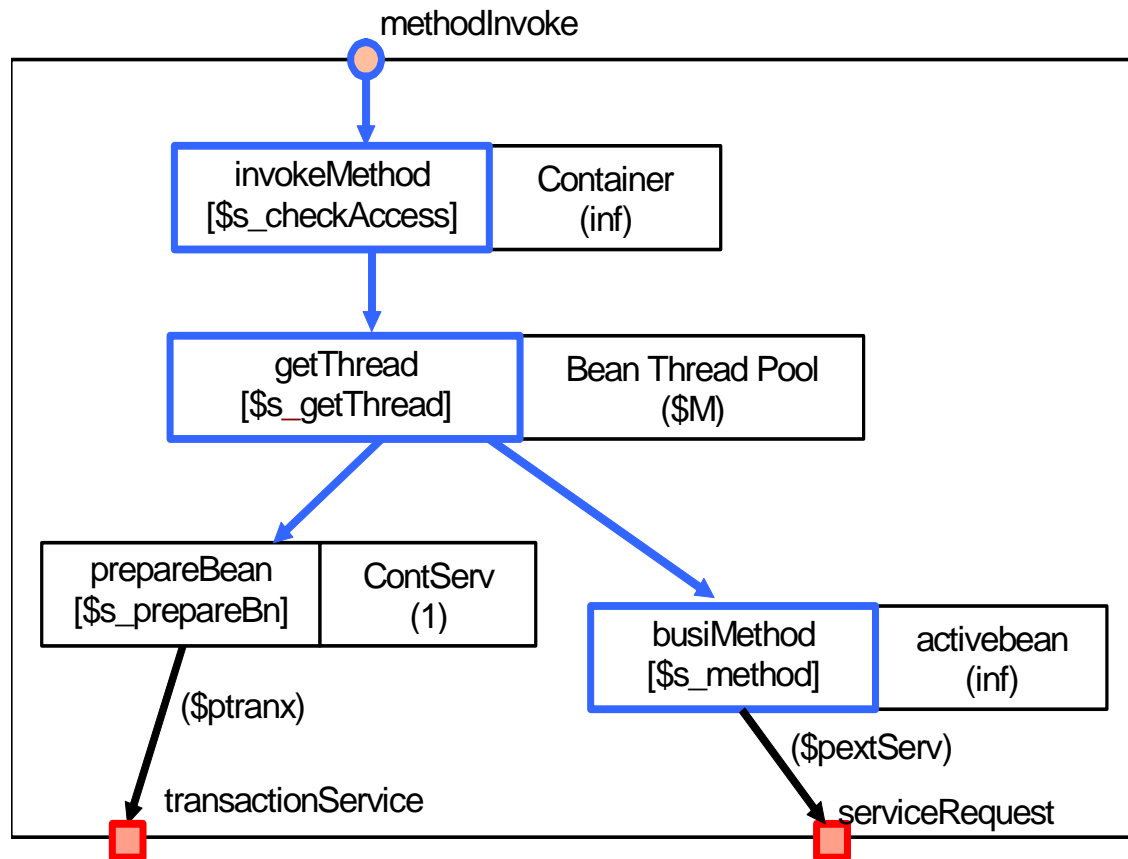


- ✓ Entity Bean
- ✓ Session Bean (Stateful/Stateless)
- ✓ Message Driven Bean

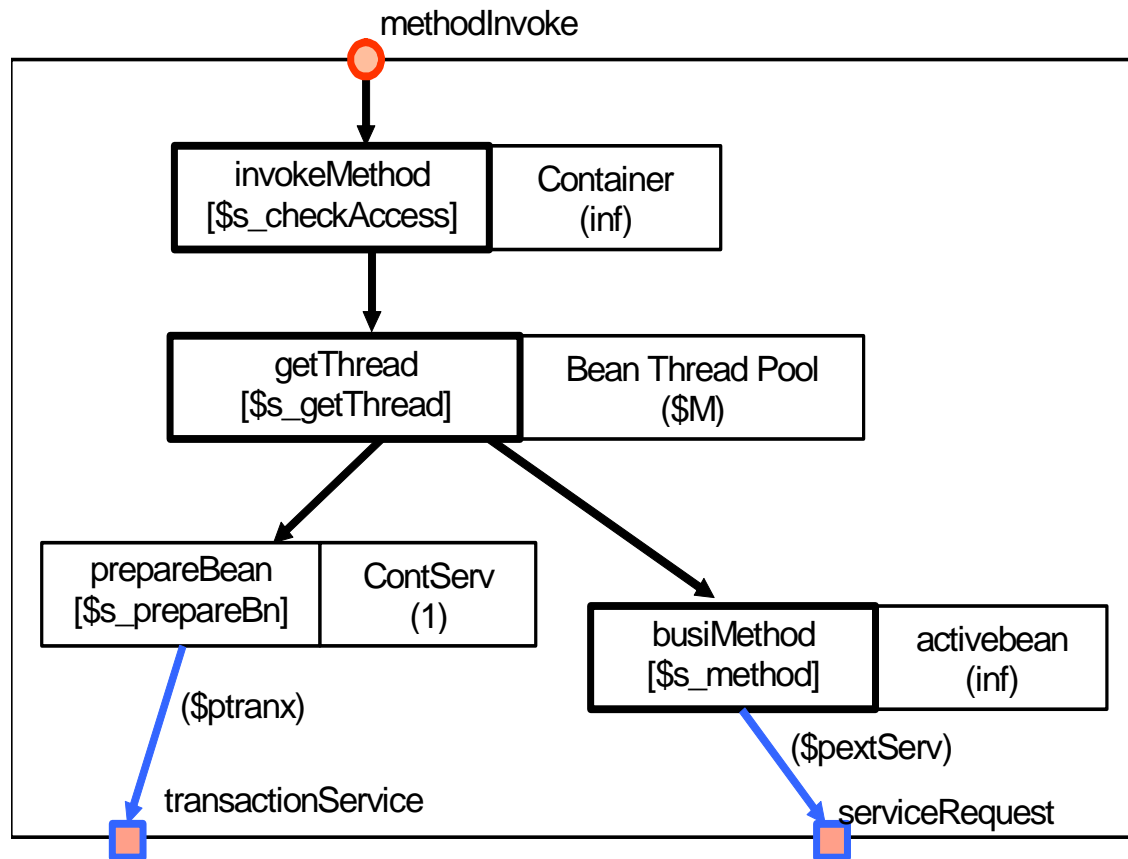
LQN Component Model Template for Session Bean (Stateless)



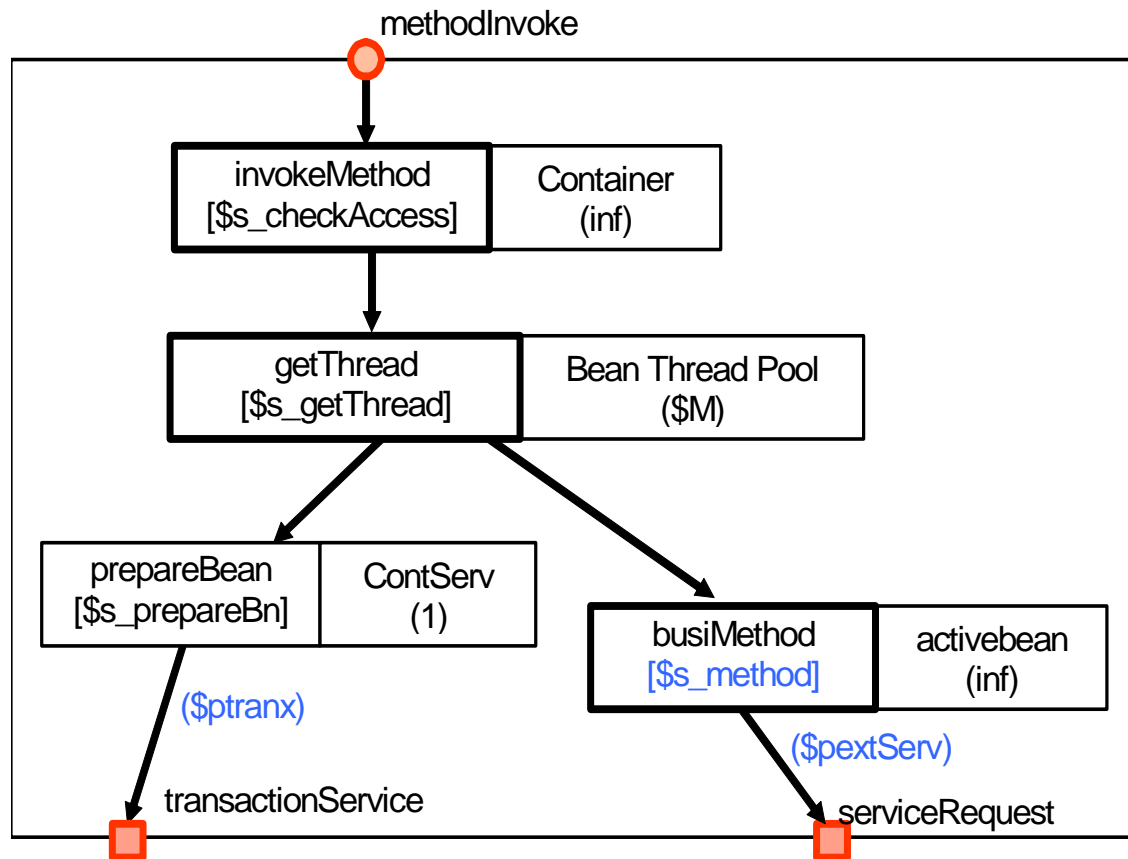
LQN Component Model Template for Session Bean (Stateless)



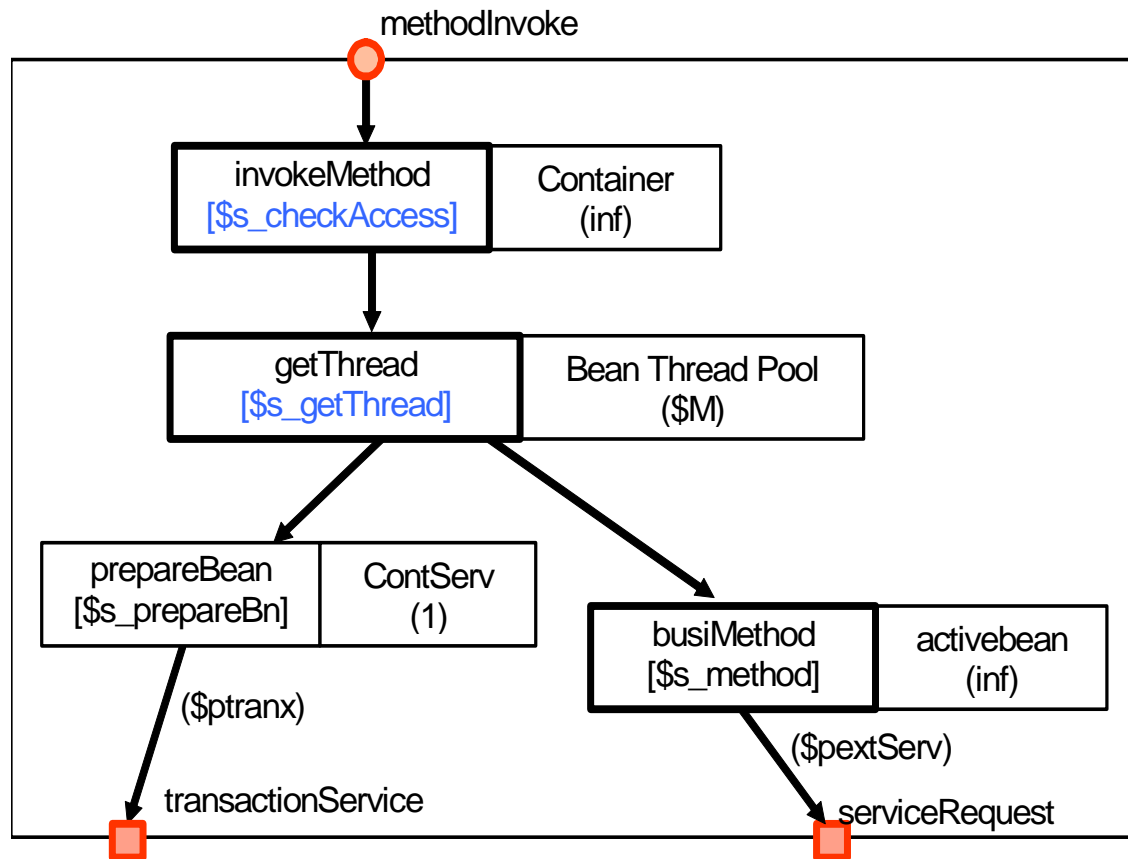
LQN Component Model Template for Session Bean (Stateless)



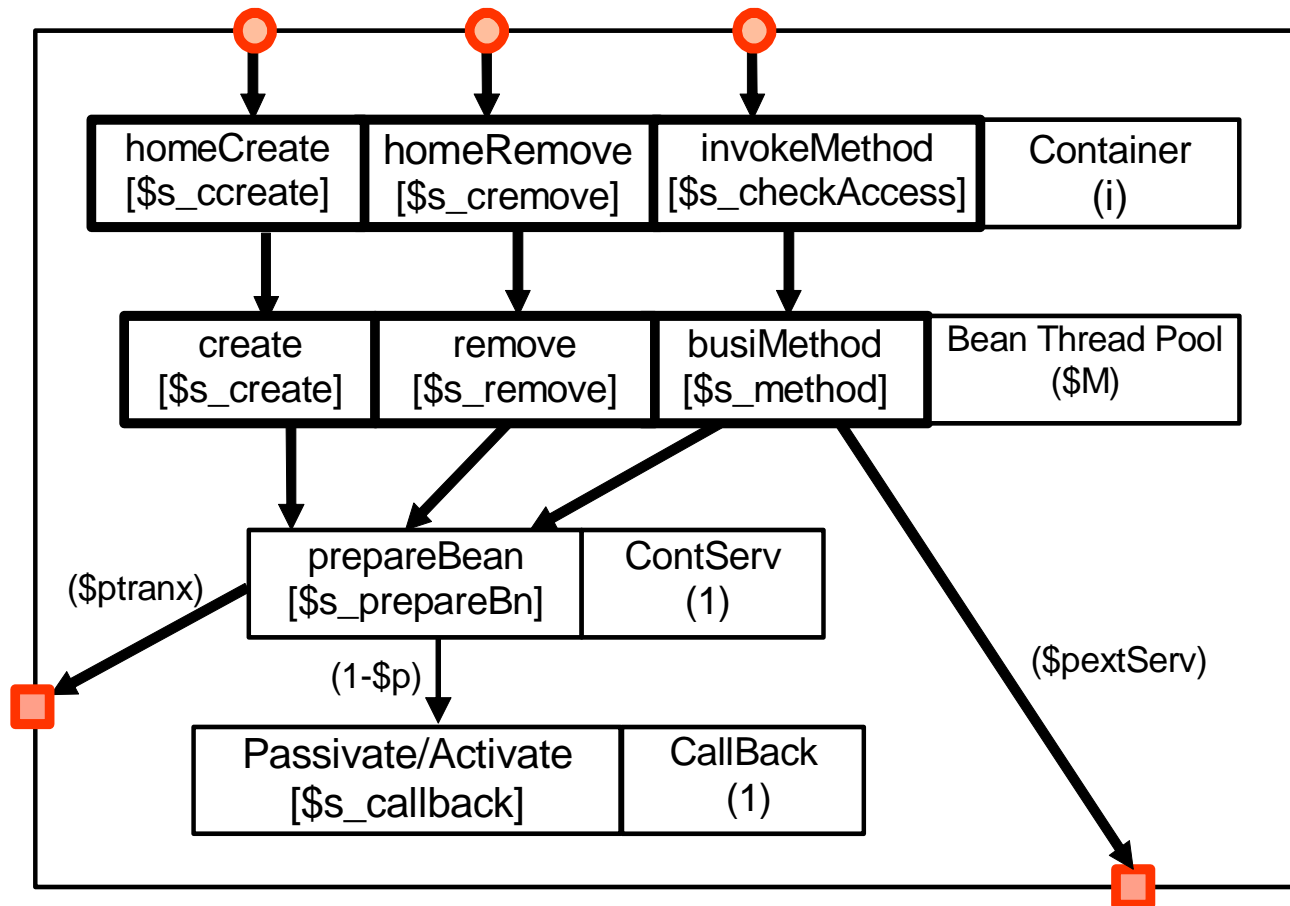
LQN Component Model Template for Session Bean (Stateless)



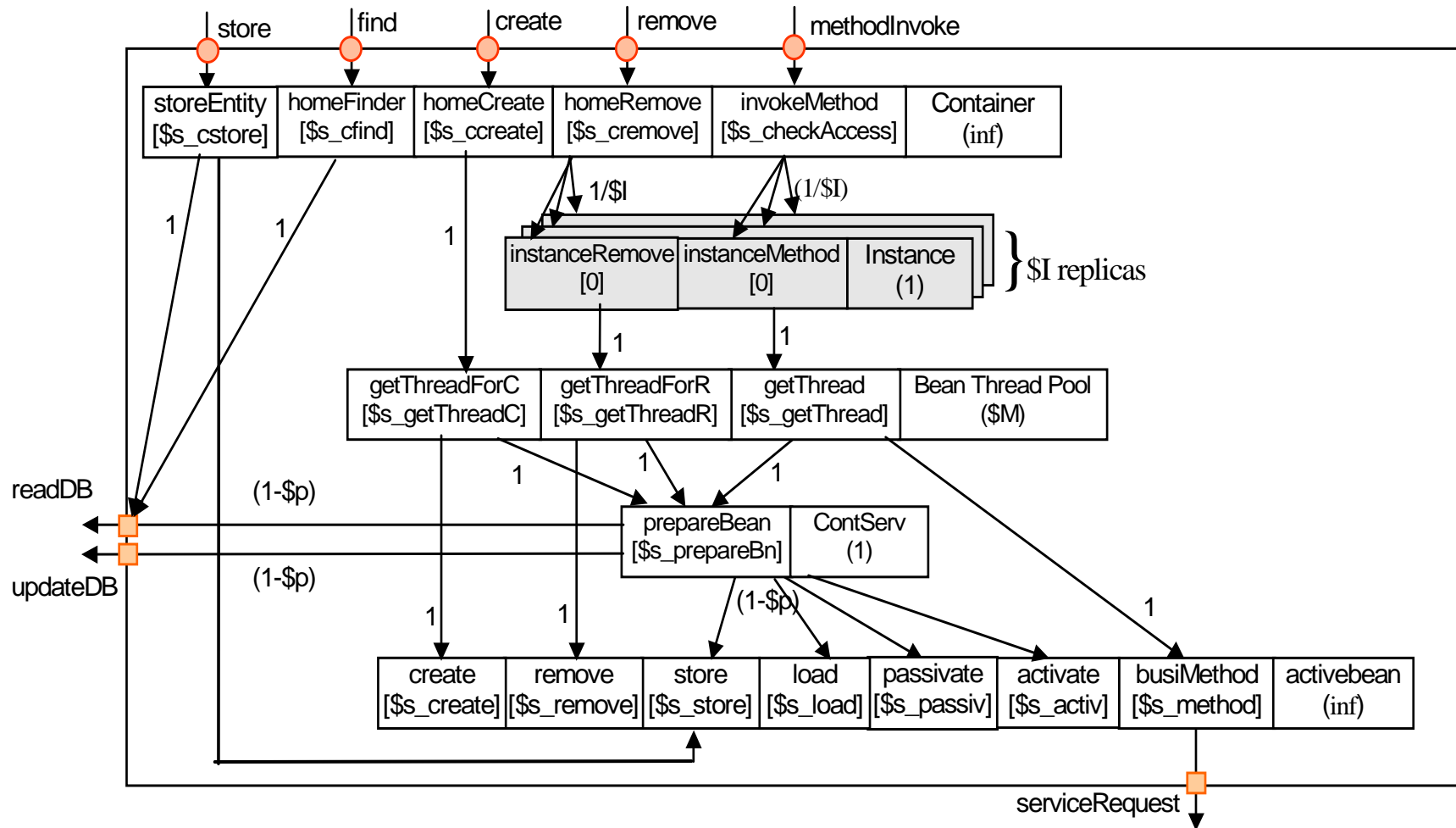
LQN Component Model Template for Session Bean (Stateless)



LQN Component Model Template for Session Bean (Stateful)



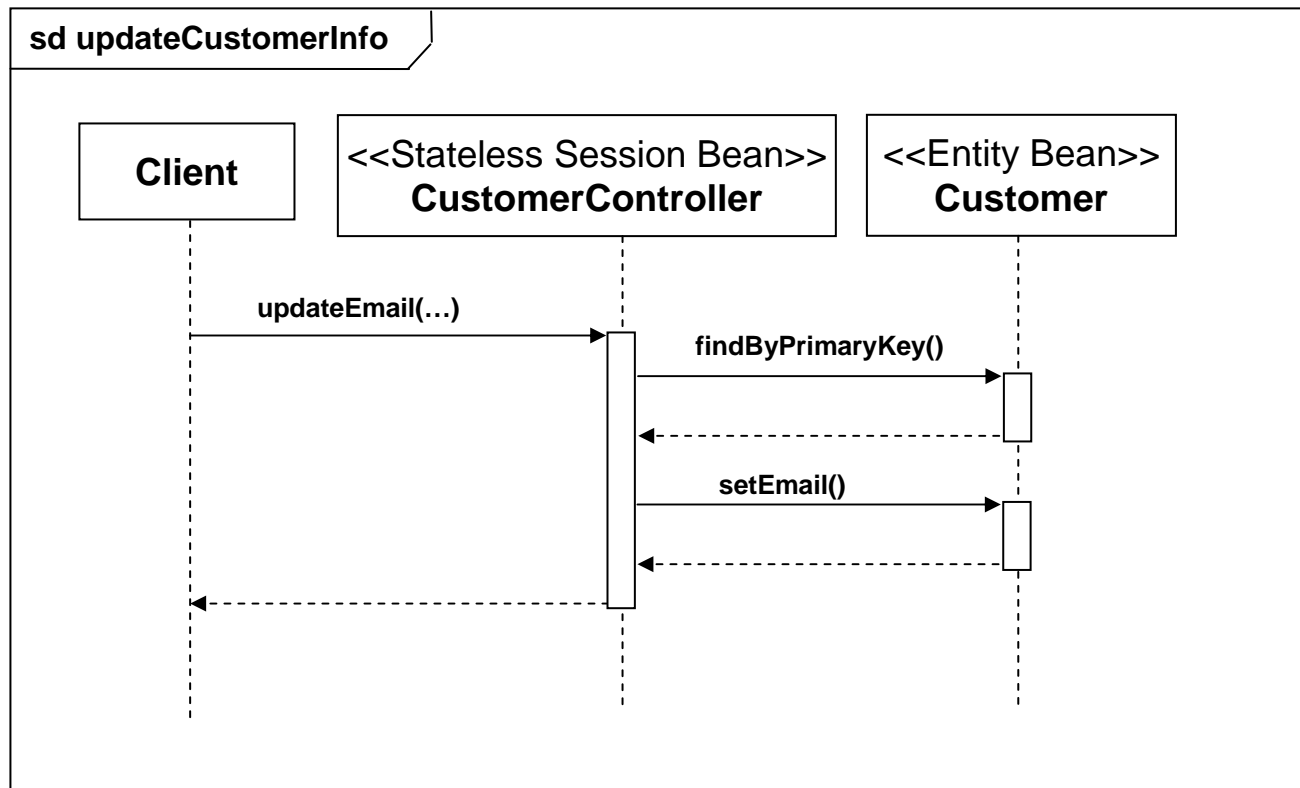
LQN Component Model Template for Entity Bean



Example Application – Duke's Bank

- **Standard J2EE sample application by Sun's Microsystems, shipped with every J2EE tutorial**
- **Modifications**
 - **CMP instead of BMP;**
 - **multiple users support;**
 - **stateful SB converted to stateless SB;**
 - **artificial congestion at the pool/cache**
- **Running on**
 - **MySQL, JBoss, Sun's JVM**

Testing Scenario: Update Customer Information



Testing Scenario

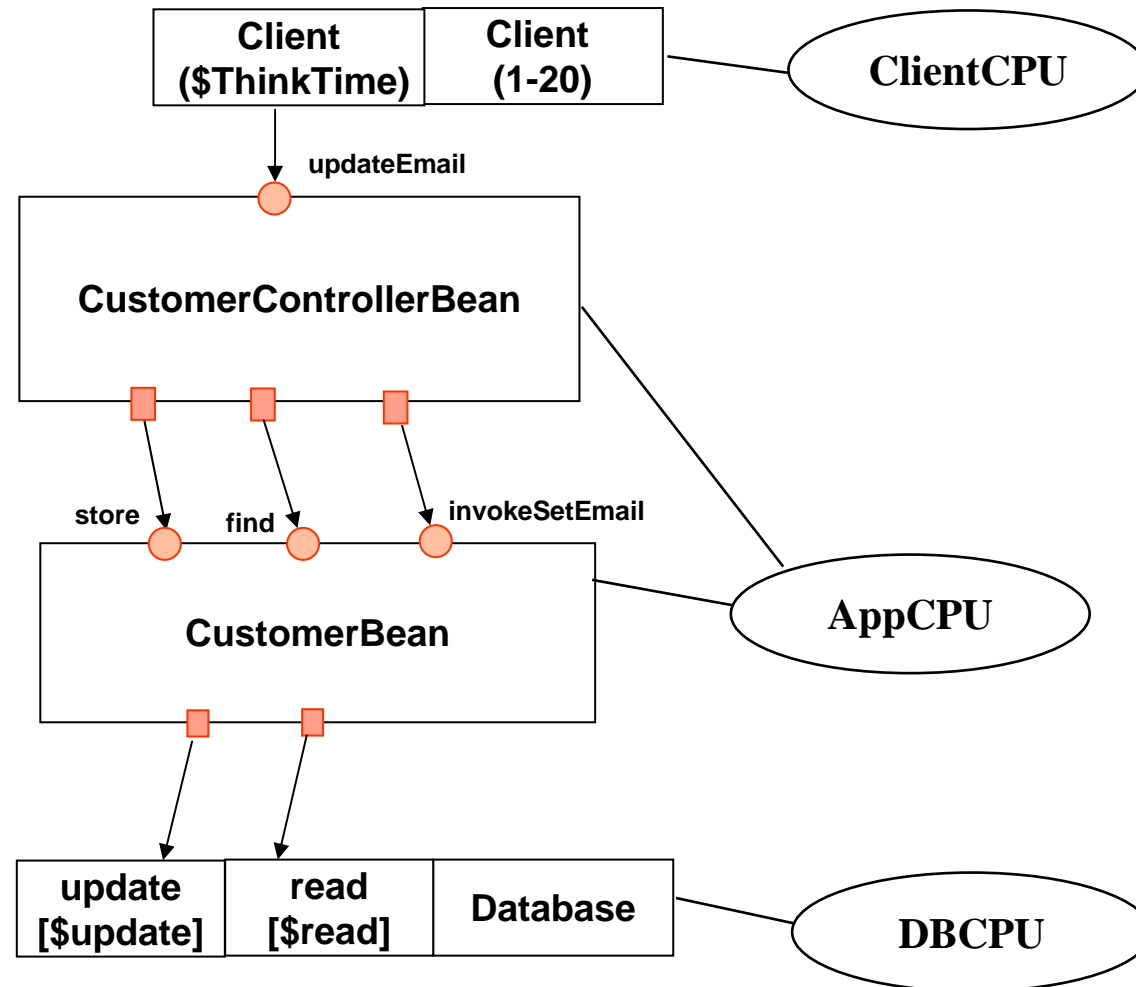
- **Data access pattern:**
 - **Sequential access vs. Random access**

- **Scenario:**

Gradually increasing workload from 1 to 20 users with step size 1. Each user:

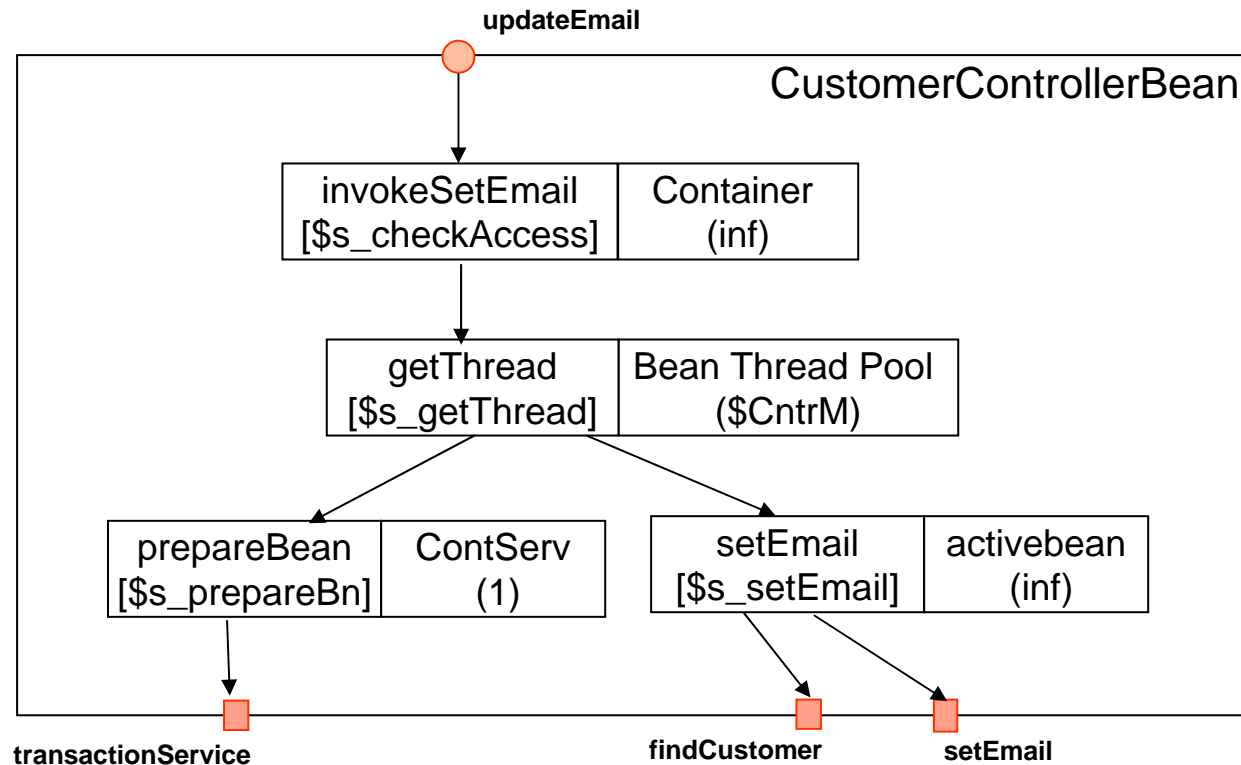
 - **Updates each customer record in ascending order (300 records in total);**
 - **Waits for other clients to finish;**
 - **Updates a random customer record 300 times;**
 - **Waits for other clients to finish.**

High Level LQN Model for the Scenario



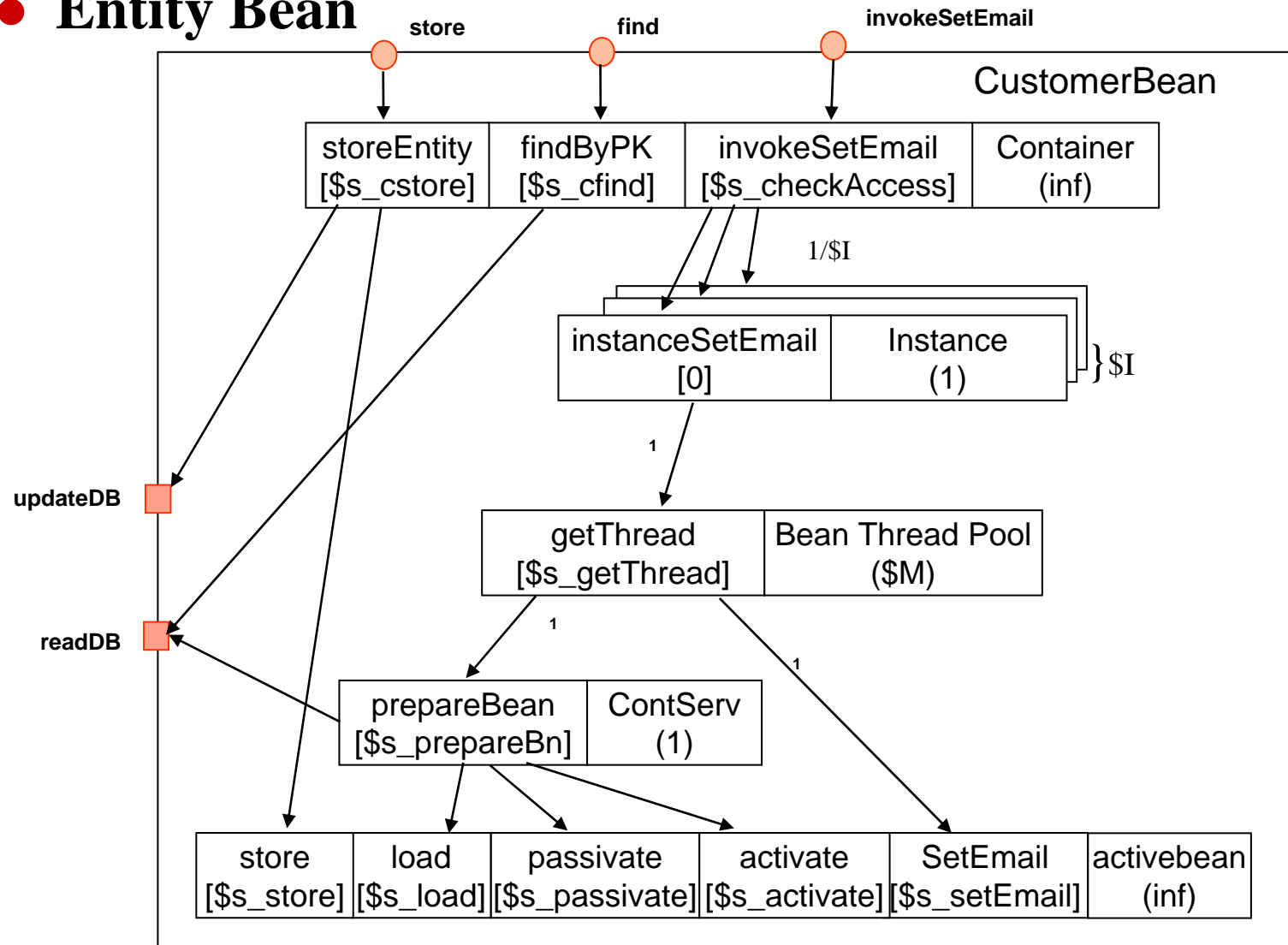
LQN Component Model for Customer Controller

- **Stateless Session Bean**



LQN Component Model for Customer

● Entity Bean

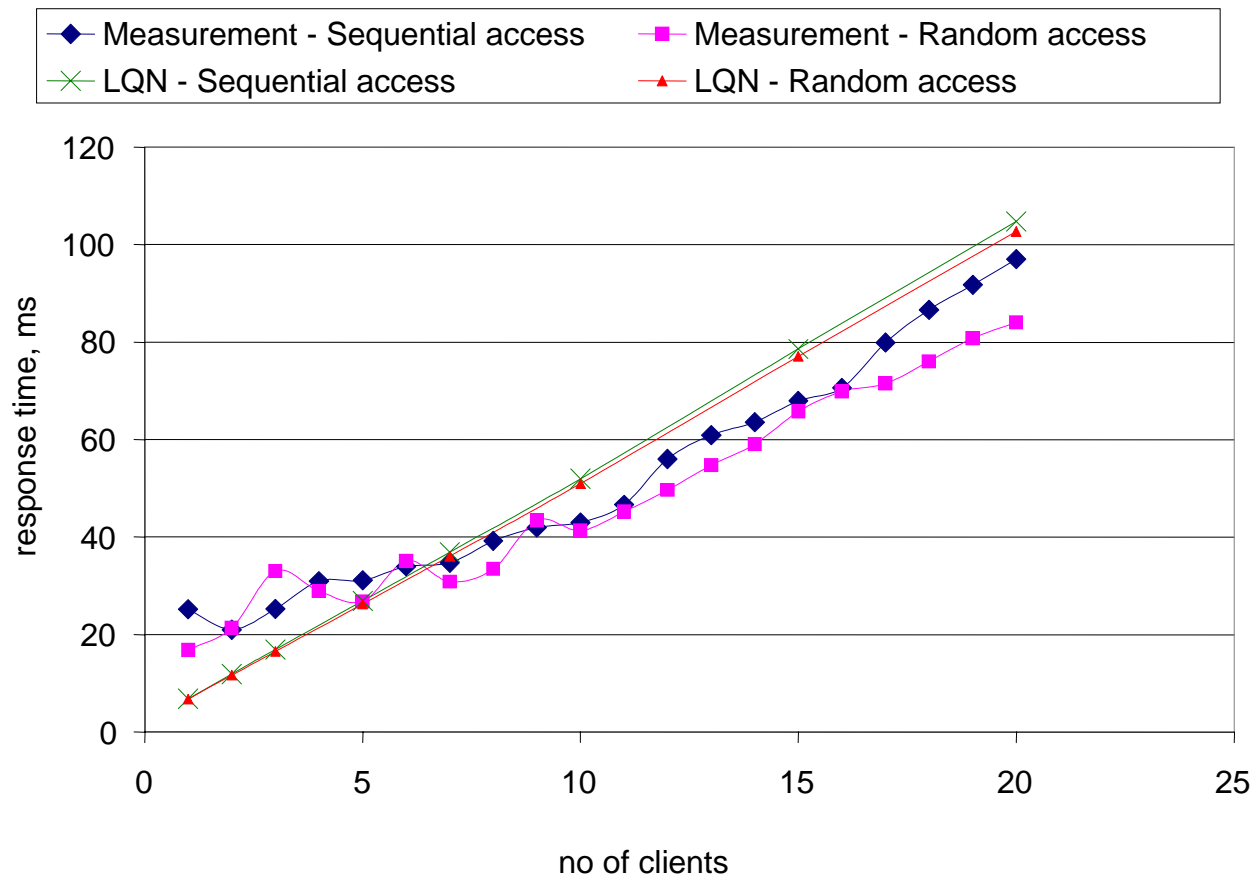


Profiling and Model Calibration

- **Tool: JProbe profiler**
- **Calibration done using single-user profiling data**
- **2 factors used for adjusting execution demand parameters**
 - **Profiling Ratio Factor (PFC): to remove overhead introduced by the profiling tool**
 - **Warm System Factor (WSF): to adjust the parameter values obtained in cold system status to reflect the warm system behavior**

System Measurement and Comparing with Model Prediction

- **Measured and predicted response times**



Conclusion and Future Research

- **A procedure for constructing, calibrating, solving and analyzing models of real-life J2EE applications;**
- **Investigate if the similar approach can be applied to other technologies such as .NET;**
- **Extend our work to incorporate crucial parts of underlying software layers into models (JVM, OS).**

Appendix A: System Measurement of Throughput

