Members

Kyu Hou
kyuh@andrew.cmu.edu
MSE

Min Ho Jeung
mjeung@andrew.cmu.edu
MSE

Wangbong Lee
wangbonl@andrew.cmu.edu
MSE

Heejoon Jung
wangbonl@andrew.cmu.edu
MSIT–SE

Wen Shu Tang
wtang@andrew.cmu.edu
ECE

http://www.ece.cmu.edu/~ece749/teams-06/team7/
Baseline Application

- Express Bus Ticket Center

Application
Online express bus ticketing application

Configuration
- Operating System: Linux servers
- Programming Language: Java
- Database: MySQL
- Middleware: CORBA

Baseline Application Feature
- Users can retrieve bus schedules and tickets.
- Users can buy tickets.
- Users can cancel the tickets.
Baseline Architecture (before)

Allocation View-Deployment Style

Legend

- Client
- Server
- Internet
- Database
- Request message
- Response message
- Request Query
- Response Query
Baseline Architecture (after)

Allocation View-Deployment Style

Pool of Customers

Bus Ticket System

Database

Legend

Client
Server
Database
Request Message
Response Message
Request Query
Response Query
Client requests should be preserved, when exception is occurred.

Replication
- There are 2 copies of server which perform same operations for fault-tolerance on the chess and risk machine.

Replication Type
- Active Replication
  - Advantage: Performance
  - Disadvantage: More memory and processing cost

Replication Manager
- No specific replication manager exists.
- As soon as client application begins, the application acquires the replication server name which is stored in Naming Server.

Elements of Fault-Tolerance Framework
- Global Manager: Heartbeat
- Recovery Manager
  - Re-instantiating a failed replication
  - The recovery result is written into a log file in Database.
- Fault injector: Shell
Scenario

1. Client requests the names of server to the naming server.
2. The naming server sends the names of servers.
3. Client requests to all servers.
   a. When the client receives an exception message, then the fault is detected.
   b. The client already communicates with another replication server.
4. All servers send the results to clients.
5. Client receives the results, and checks duplication.
Mechanisms for Fail-Over

- Exception Cases
  - **Server_Timeout**
    - Checked by using thread pool
  - **Database_Timeout**:
    - Checked by using connection pool
  - **Dead_Server**
    - Solved by using heartbeat (check servers per 2 seconds)
- Global Recovery Manager: Heartbeat
Performance measurement

- 48 Configurations
- Buy and cancel ticket
Performance measurement

Latency for increasing number of clients

Mean latency vs. Number of Clients
Performance measurement

![Area plot of mean and 99% latency](image)

Note: See endnotes for configurations.
Performance measurement

3D scatter plot of reply size, request rate impact on max latency

- X-axis: Request size [bytes]
- Y-axis: Request rate [req/s]
- Z-axis: Maximum latency [us]

The plot shows scatter points representing the relationship between request size, request rate, and maximum latency.
Performance measurement

3D scatter plots of reply size and request rate impact on 99% latency
Fault Injection measurements

- 1 Client
- 1000 requests
- Cancel ticket request
Performance measurement comparison
RT-FT Baseline Architecture

- Active Replication
RT-FT Performance Strategy

- Thread Pool
  - We need to avoid the overhead of thread creation for each request.
  - Create a number of threads at initialize time
    - Dynamic configuration
  - Without Thread Pool
    - AVG RTT: 40.5 msec
  - With Thread Pool
    - AVR RTT: 38.0 msec
  - Improve the performance about 4%
RT-FT Performance Measurement

- Thread Pool / No Thread Pool
Other Feature

- List other features
  - Fault Injector – Shell Script
  - Log4j – Logging information
  - Apach DB Connection Pool (DBCP)

- What lessons by other features?
  - Useful utilities
  - Improve performance by DBCP
  - Powerful shell scripts
Insights from Measurement

- **FT Measurement**
  - File I/O for logging time grows as the file size increases

- **RT–FT Measurement**
  - No RTT difference between fault-free and fault-injected test cases
    - Duplicated values reach the client almost at the same time.

- **RT–FT Performance Measurement**
  - Thread creation time is not trivial when the number of replica increase
    - Need more test cases
Open Issue

Issues

- Test environment
  - How to set up same test environment for each test case.
  - How to decide test environment is good enough to get the meaningful data.

Additional features

- Load balancing for active replication
  - Organizing active replication group
  - Passive replication for each group
Conclusion

- What did we learn?
  - Handling thread
  - Data gathering and analyzing
  - Useful open source program
    - Apache project :log4j, dbcp

- What did we accomplish?
  - succeed to build active replication system

- If we could start our project again,
  - focus on only FT features