Team 4: IJAMS

17-654: Analysis of Software Artifacts
18-846: Dependability Analysis of Middleware

IJAMS

(Integrated Job Applicants Management System)

Daehyun Jung // Inhong Kim // Jungjin Suh
Hyoungiel Park // Huiyeong Choi

Electrical & Computer
ENGINEERING

Team 4 Members

DaeHyun Jung

djung@andrew.cmu.edu

InHong Kim

ihk@andrew.cmu.edu

JungJun Suh

jjsuh@andrew.cmu.edu

Hyoungiel Park

hip@andrew.cmu.edu

Huiyeong Choi

hchoi1@andrew.cmu.edu

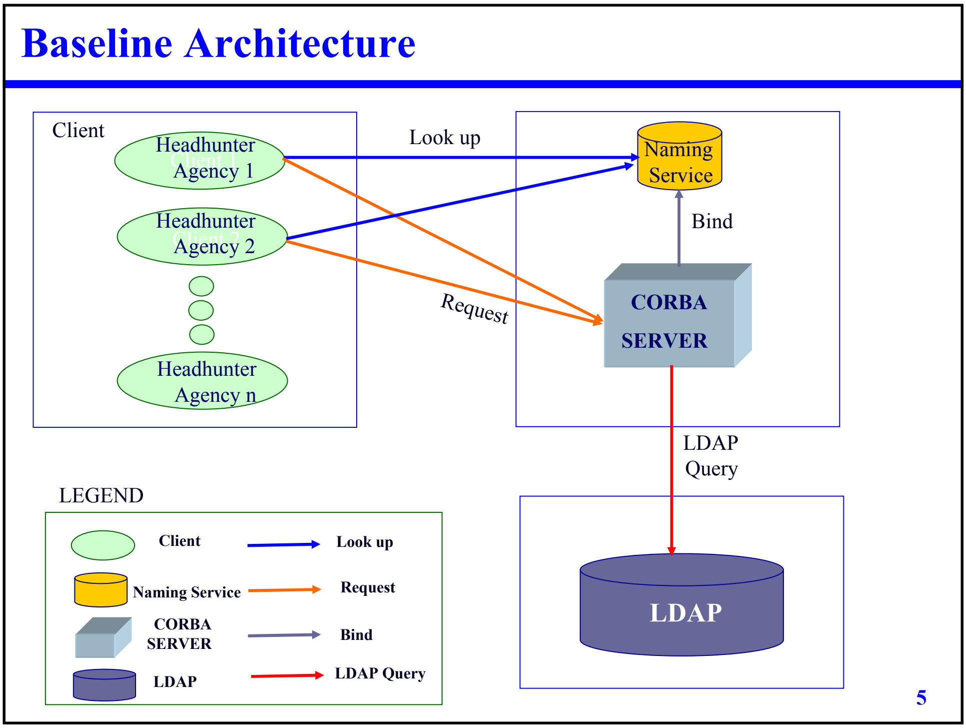
Team 4 Homepage:
http://www.ece.cmu.edu/~ece846/team4/index.html

Application Overview

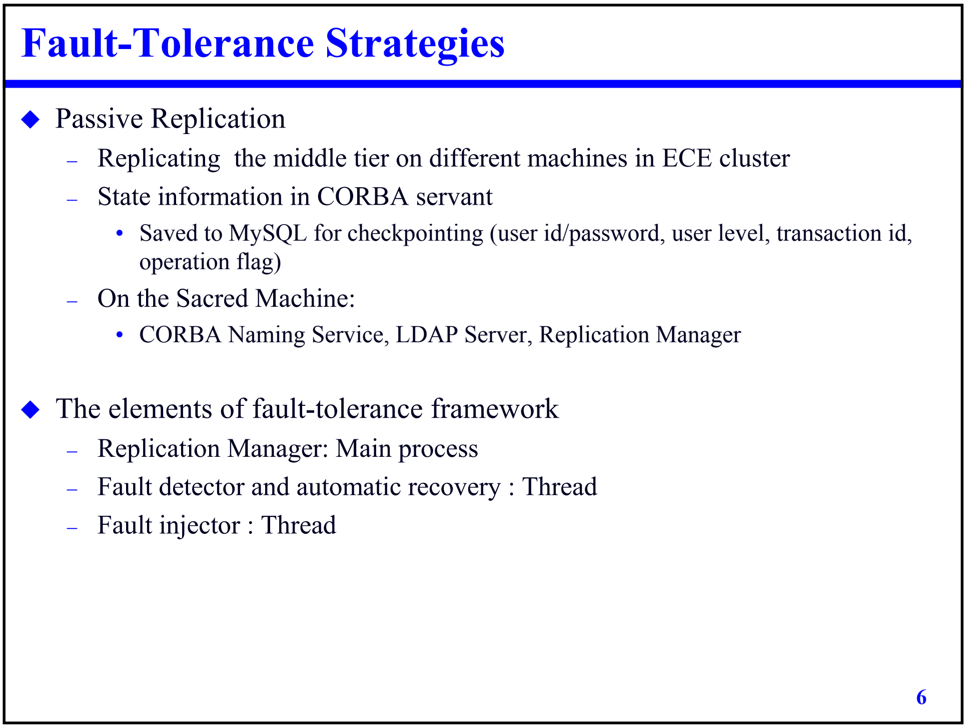
- What does the application do?
  - Integrated Job Applicants Management System (IJAMS)
  - IJAMS connects registered headhunting agencies to integrated HR database
- What makes it interesting?
  - LDAP (Lightweight Directory Access Protocol) Server as Database
  - Heavy data transmission per transaction
    (100 applicants entries searched for query)
  - Remote LDAP (in Korea) and Local LDAP (in MSE Cave)
- LDAP & RDBMS
  - Reading time is faster than RDBMS
  - Less resources are used (memory resource)
  - Connection to external DB is easier.
  - http://www.openldap.org/

Development Environment

- Middleware : CORBA 2.3.1 (embedded in JDK 1.4.2, idlj 3.1)
  Light-weight (No administrative privilege required for handling server, less time is taken for restart and less resource consumed at runtime compared to EJB server)
- Language : Java 1.4.2
- API : Netscape Directory SDK 4.0 for Java
- Platform :Linux
  ECE cluster
  (ssh is used for building replication manager)
- Main Database : SunOne LDAP 5.1
  Back-end data tier, high performance in data retrieving
- Checkpointing Database : MySQL 4.0 (Sun Solaris 2.9) (for FT-Baseline)



Baseline Architecture

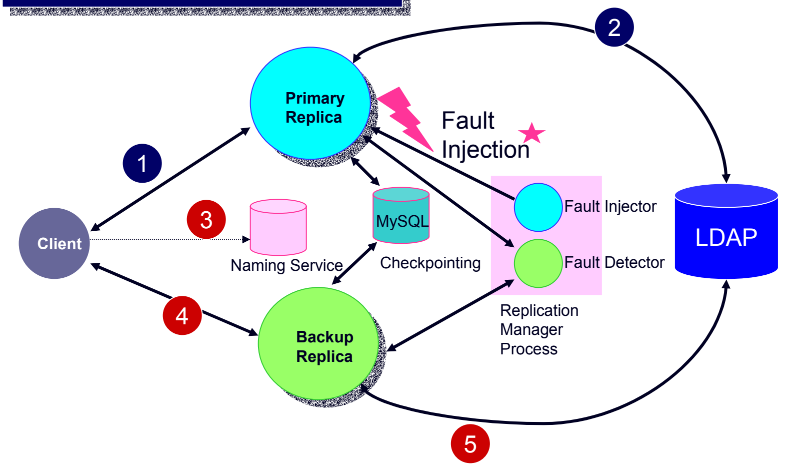


Fault-Tolerance Strategies

- Passive Replication
  - Replicating the middle tier on different machines in ECE cluster
  - State information in CORBA servant
    - Saved to MySQL for checkpointing (user id/password, user level, transaction id, operation flag)
  - On the Sacred Machine:
    - CORBA Naming Service, LDAP Server, Replication Manager
- The elements of fault-tolerance framework
  - Replication Manager: Main process
  - Fault detector and automatic recovery : Thread
  - Fault injector : Thread

FT-Baseline Architecture (1)

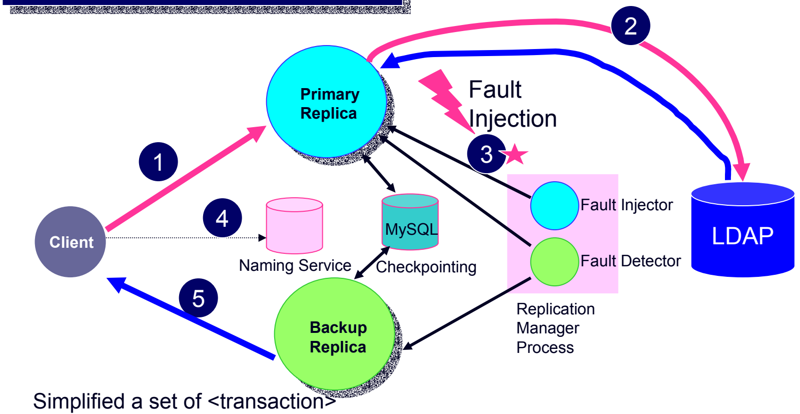
Data Retrieving Operation



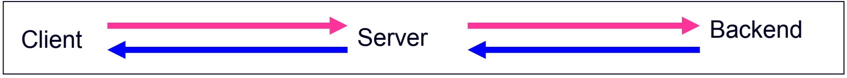
★ **Fault injection point is not determined for this scenario because whenever an error occurs during this operation, the operation is re-tried from scratch**

FT-Baseline Architecture (2)

Data Updating Operation



Simplified a set of <transaction>

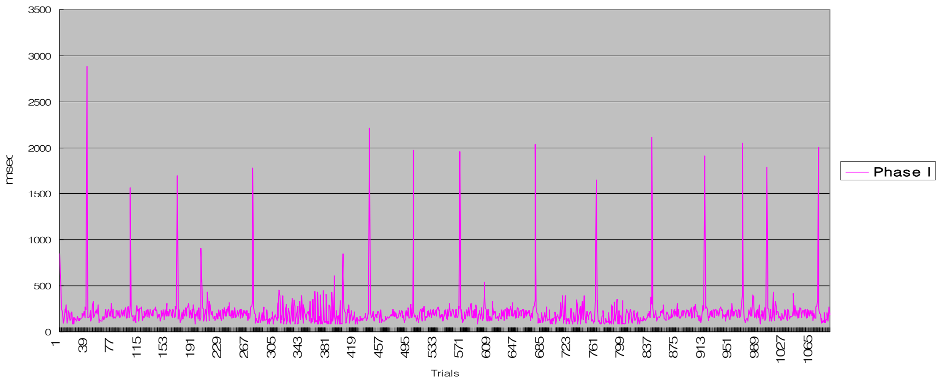


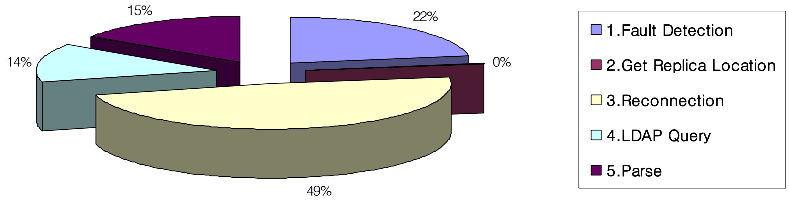
★ **Since data update operation affects data on LDAP server, the operation must not be duplicated. Recovery is possible by having new primary replica continue processing without starting from scratch**

Mechanisms for Fail-Over

- Fault Detection
  - Client obtains the names of the replica reference when it starts
  - We use one of CORBA Exceptions: COMM_FAILURE
  - The client gets a new CORBA replica from Naming Service
- Fail-over
  - Backup replica waits to take over
  - The client retries the operation with the new replica
  - The user of the client is reauthenticated on the new replica with the user data in the checkpointing DB
  - If fault occurs, then the backup replica takes over with saved checkpointing information

Fail-Over Measurements



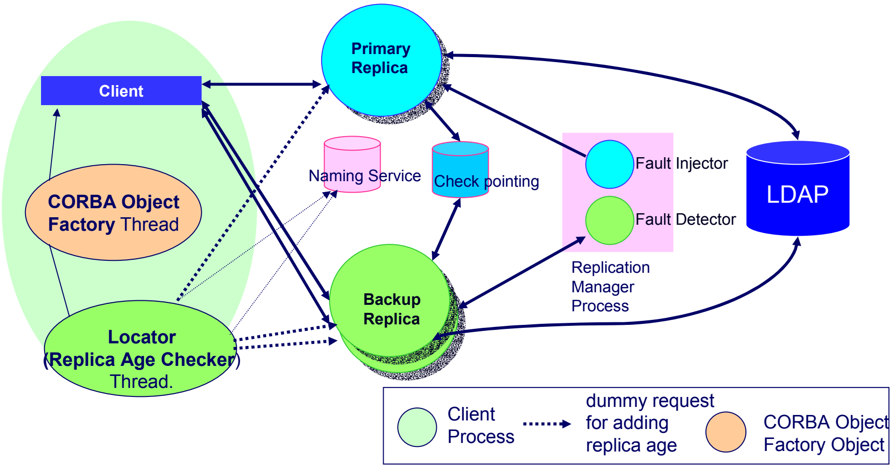


RT-FT-Performance Strategy

- The primary reason of fail-over “spike”
  - TCP/IP reconnection between client and server

- RT-FT strategy is to reduce the time of TCP/IP reconnection
  - Client pre-establishes connections with both replicas
  - 3 Replicas (Primary and 2Backup)
  - Locator : Choose replica which is most likely to have established its connection.
    - » Send dummy request to the replicas for adding number which represents the age of replica.
    - » Oldest replica is prepared for fail-over
  - CORBA Object Factory : Return CORBA object reference to client
    The object reference of the oldest replica

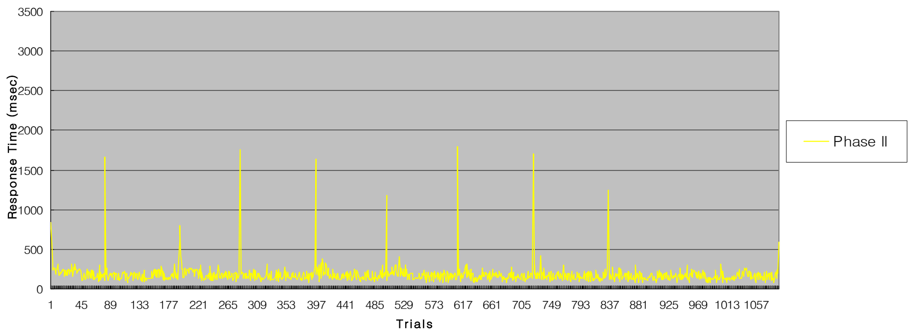
RT-FT Baseline Architecture


Primary Replica
Client
Naming Service
Check pointing
Fault Injector
Fault Detector
LDAP
CORBA Object Factory Thread
Replication Manager Process
Backup Replica
Locator (Replica Age Checker) Thread.
Client Process
dummy request for adding replica age
CORBA Object Factory Object

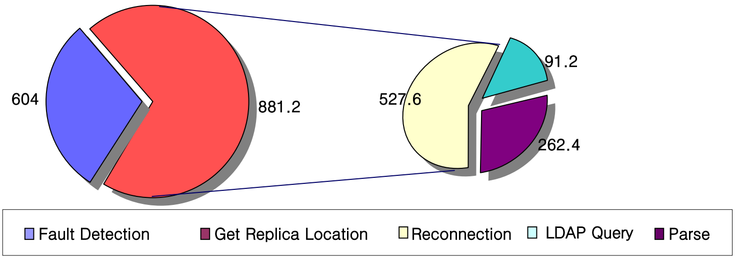
★ **Locator : Decide which replica is more likely to have established its connection.**
Increase replica age by sending dummy request to the replica periodically.

★ **CORBA Object Factory : Returns CORBA object reference to the client**

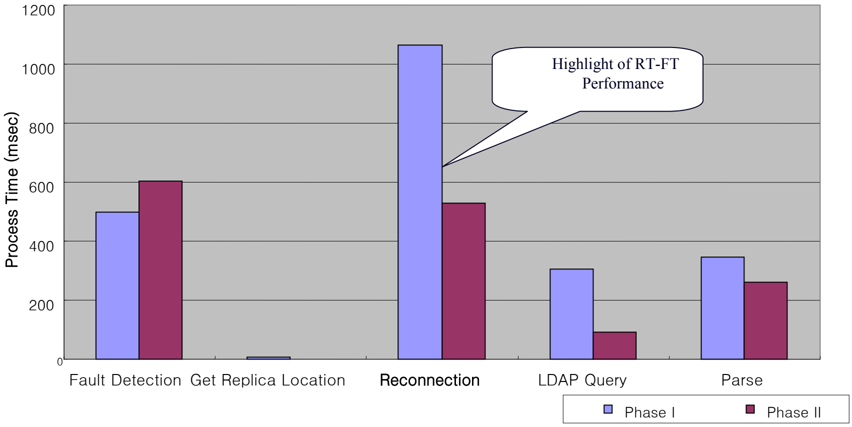
RT-FT-Performance Measurements



<Contribution Factors for Failover (msec)>



Bounded "Real-Time" Fail-Over Measurements

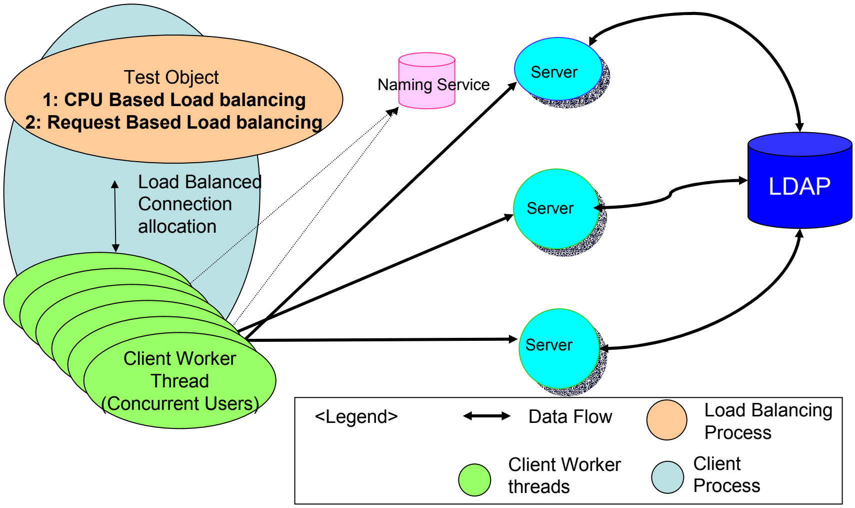


Highlight of RT-FT Performance

- Reconnection time is reduced by 49.5 %
- LDAP query time : Different location of LDAP server (Phase1 : In Korea, Phase2 :Local)
- Fault Detection Time : Client side load caused from background TCP/IP connection
- Parsing time : Affected by variability of system environment.

Performance Strategy

- Load balancing
  - Number of clients connected to a server
  - CPU load of a server



Test Object
1: CPU Based Load balancing
2: Request Based Load balancing

Load Balanced Connection allocation

Client Worker Thread (Concurrent Users)

Naming Service

Server

Server

Server

LDAP

<Legend>

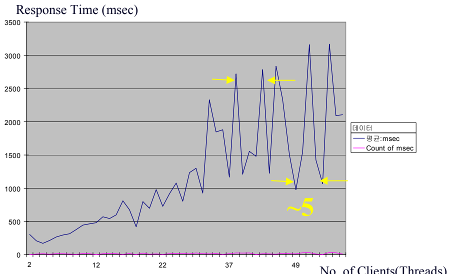
Data Flow

Load Balancing Process

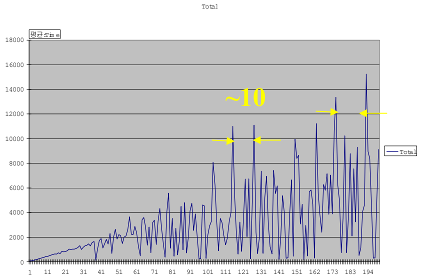
Client Worker threads

Client Process

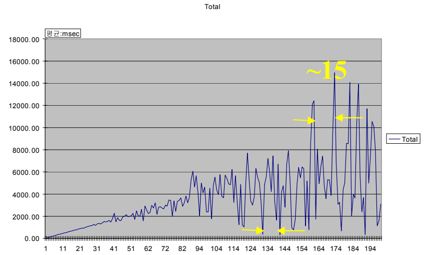
Performance Measurements



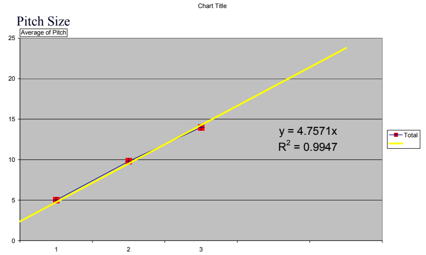
Single server



Two servers

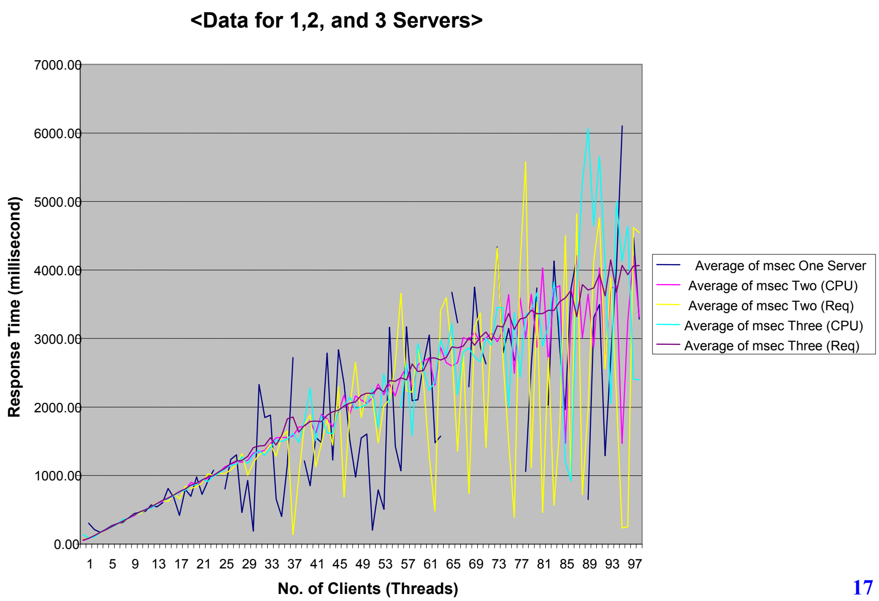


Three servers



Scalability Trend

Superimposed Performance Data



Insights from Measurements

- Fault Tolerance
  - TCP/IP connection is most significant factor for latency in fault recovery
- RT-FT
  - Background connection pre-establishment helps to reduce failover time
- Performance
  - Thread pool and bottleneck tier must be considered

Open Issues

- Issues
  - What is the exact source of the bottleneck ?
  - We suspect the backend database
- Additional features if we have more time
  - FT
    - Replication of sacred services such as Replication Manager or Naming Service
    - Active replication
  - RT-FT
    - Saving LDAP connection object as checkpointing for saving time for reauthentication.
    - Optimization of CORBA server
    - CORBA persistent reference to reduce Naming Service load
  - Performance
    - Reducing delay time for checking CPU load using “ssh” command

Conclusions

- Accomplishments
  - FT: Passive replication strategy and selection criteria
  - RT-FT: Background connection pre-establishment strategy
  - Performance: Load balancing strategy (CPU load and # of connections)
  - RT and Performance analysis

- Lessons learned
  - Identifying exact points of the bottleneck
  - Checkpointing strategy

- Considerations if we restarted afresh
  - Set replication point after a complete performance analysis at each tier
  - Analyze WAN vs LAN impact