Scalability Evaluation of the Replication Support of JOnAS, an Industrial J2EE Application Server

A. Paz, F. Pérez-Sorrosal, M. Patiño-Martínez, R. Jiménez-Peris

Lsd Distributed Systems Laboratory

Universidad Politécnica de Madrid (UPM)
Focus

- Reporting on the implementation of replication protocol in an industrial application server, JOnAS, for J2EE-based multi-tier architectures
- and its evaluation using the industrial benchmark for J2EE, SPEC AppServ.
Replication Model

![Diagram of replication model with clients, application servers, and communication channel]

- Replication Protocol
- App. Data
- DB

EDCC 2010 - Valencia, Apr. 2010
Replication Protocol: Features

- Multi-master/Update everywhere
- Based on eager replication
- Preserves state/data consistency
  - Exactly-once semantics
- Transparent replication via specialized replication proxies.
- Suitable for J2EE environments, which include:
  - Transactional Service (JTA/JTS)
  - Component Model: Enterprise Java Beans (EJBs)
    - Stateless (SLSB) and Stateful (SFSB) Session Beans, Entity Beans (EB) & Message-Driven (MDB)
Replication Protocol

- Each client session/request is encompassed in a TX (container managed transactions)
- An invocation performs changes on client-related state (SFSFs) and persistent state (EBs)
- At commit time…
  - The client session state (SFSBs) is multicast to the replicas acting as backups (FIFO Order)
  - EBs are written to the shared DB
Replication Protocol (II)

- To preserve exactly-once semantics...
  - As part of an update request, a marker (e.g. TX Id) is written in the shared DB with the returned response
  - When SFSBs are replicated, the marker and the response are attached to the message

- In case of failover...
  - The client connects to a new replica
  - If client has not received a response from the old primary...
    - Resubmits the request to the new primary
    - The new primary checks if the marker is present in the DB
    - If present, it returns the stored result. Otherwise, it re-executes the request
Evaluation: SPECjAppServer 2004

- Benchmark to compare J2EE AS implementations
- Car Dealer Application Domain
  - Workload Generator injects load (IR) in the App.
  - Clients = IR x 10
  - 3 TX Types:
    - Browse Vehicle Catalog (50%) Read-only
    - Purchase Vehicles (25%) Read-write
    - Manage Dealer Inventory (25%) Read-write
  - Max. Response Time for requests (2 secs.)
  - Throughput measured as TX/Sec.
  - 3 Phases: Ramp-up (4 Min) + Measure (10 Min) + Cool-down (4 Min)
Evaluation: Environment

**HW Environment**
- Up to 10 nodes
- Connected through 100 MBps Ethernet
- Node: 2xAMD Athlon 2GHz CPU, 1GB RAM, 2x320 GB HD

**SW Environment**
- Fedora Core 3 OS
- Apache 2.2.0 + MOD_JK (*Load Balancing*)
- Tomcat 5.5.17 Servlet Container
- JOnAS AS 4.8.1 (*Baseline*) and 4.8.5 (*Rep. Service*)
- Two DBMSs: PostgreSQL 8.2 and Oracle 10
- JGroups GCS
Evaluation: Configuration
Results: PostgreSQL Throughput

![Throughput Graph](image)

- JOnAS 4.8.1
- 1 Rep.
- 2 Rep.
- 3 Rep.
- 4 Rep.
- 5 Rep.
- 6 Rep.

Throughput (Tx/seg)

Ir

- No replication
- 1 replica
- 2 replicas
- 3-6 replicas
Results: Postgres Resp. Time
(Browse)

- 1 replica
- 2-5 replicas
- 6 replicas

No replication
Results: Oracle Throughput

![Graph showing throughput with different replication levels](image-url)

- **1 replica**
- **2 replicas**
- **3 replicas**
- **4 replicas**
- **5-9 replicas**

No replication option is also shown for comparison.
Results: Oracle Resp. Time (Browse)
Results: Postgres Resp. Time (Purchase)

- JOnAS 4.8.1
- 1 Rep.
- 2 Rep.
- 3 Rep.
- 4 Rep.
- 5 Rep.
- 6 Rep.

- No replication
- 1 replica
- 2-5 replicas
- 6 replicas
Results: Oracle Resp. Time (Purchase)

The graph shows the response time (sec) for different numbers of replicas for Oracle Purchase transactions. The x-axis represents the number of transactions (Ir), and the y-axis shows the response time in seconds. The graph includes data for different numbers of replicas: 1, 2, 3, 4-9, and none (no replication). Each line represents a different number of replicas, with colors and markers distinguishing each configuration.
Results: Postgres CPU Usage (No Replication IR=3 and 4)

Time (secs x 10)

CPU usage (%)

AS

DB

JOnAS Ir 3
PostgreSQL Ir 3
JOnAS Ir 4
PostgreSQL Ir 4
Results: Postgres CPU Usage (1 Replica IR=3 and 4)
Results: Postgres CPU Usage (6 Replicas IR=3 and 4)
Results: Oracle CPU Usage
(1 Replica With and WO Rep. IR=8)
Results: Oracle CPU Usage (2 Replicas IR=13)
Conclusions

- We have presented an experience in implementing replication for an industrial J2EE application server, JOnAS,
- evaluating the protocol with an industrial benchmark
- Provides exactly-once semantics
- The evaluation has shown...
  - Scalability can be achieved in addition to HA
  - The database is the limiting factor for scalability