

Autonomic Energy Management in a Replicated Server System

Gadafi, A.; Broto, L.; Sayah, A.; Hagimont, D.;
Depalma, N.; Sixth International Conference on
Autonomic and Autonomous Systems (ICAS), 2010.

Summarized by:

Mohan Krishnamoorthy

Overview

- Introduction
- Problem
- Approach
- Evaluation and Results
- Discussion

Introduction

- Resource management in clusters
- Goal: conserve energy
- Achieved by adding/removing replicas
- Self-Optimizing
- Reactive approach
- Specific adaptation

Problem Details

- Cluster systems
- Different loads of requests
- Scalability
- Replicate servers for peak loads
- Load Balancing
- Resource overbooking
- Energy consumption of idle node

Example Architecture

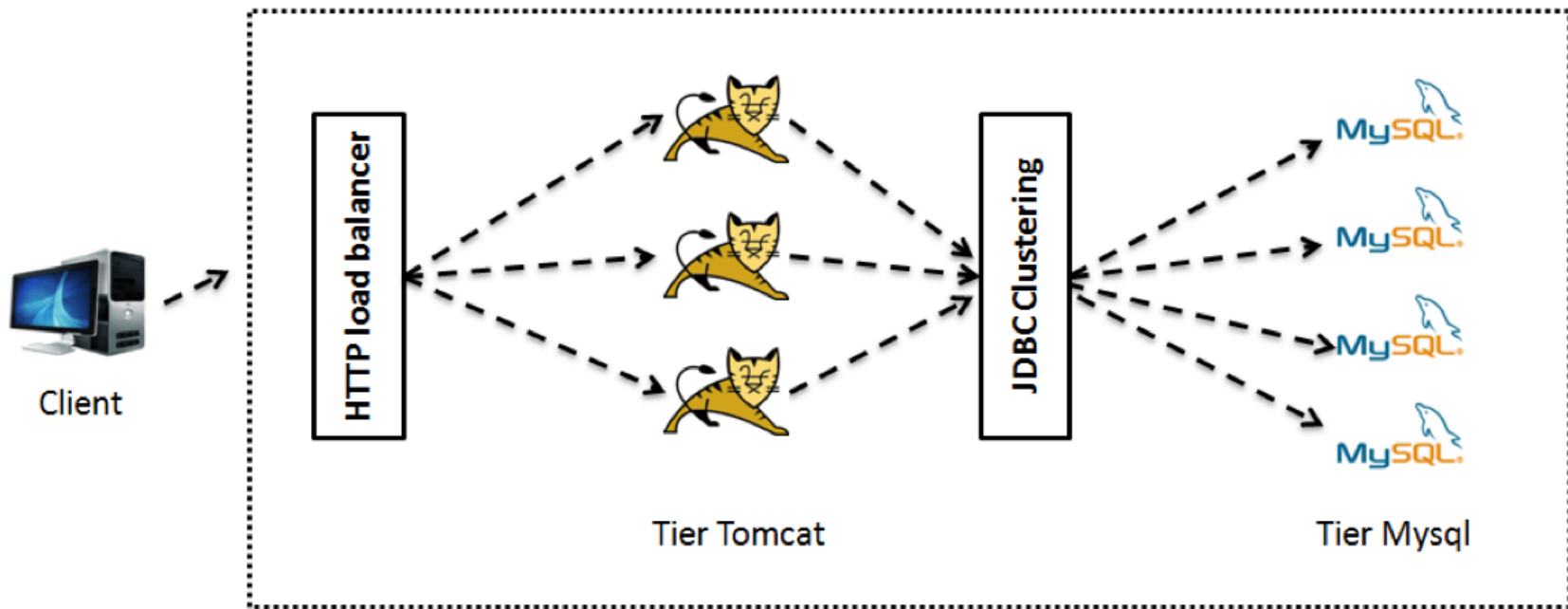


Figure 1 of Gadafi et al. Paper

Solution steps

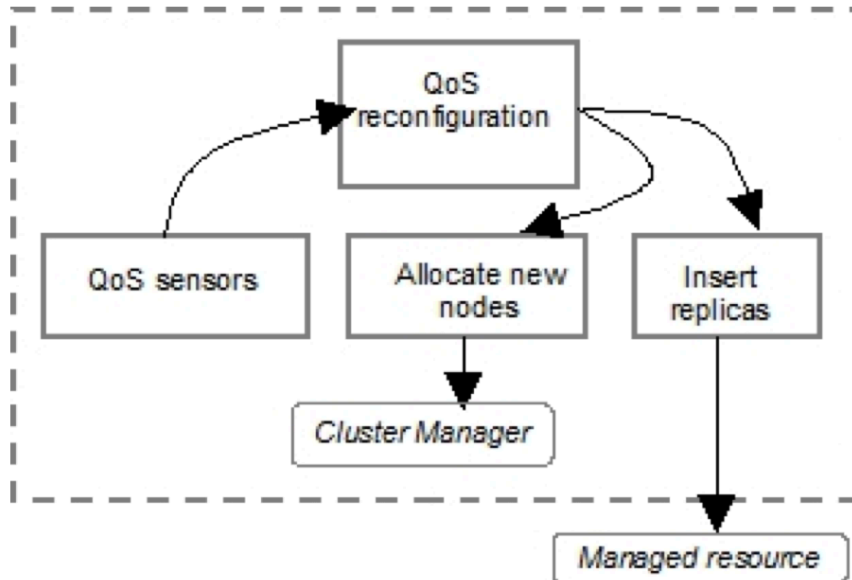


Figure 3 of Gadafi et al. Paper

- Load Increase
 - Turn on a new node
 - Start a replica
 - Integrate replica to load-balancer
- Load Decrease
 - Remove one replica from load-balancer
 - Stop replica
 - Turn-off node (RAM suspension)

Approach: Overhead

- System Oscillation
 - System load unpredictable
 - Perform add or removal of replicas only after the system is relatively stable for a certain time
- Suspension to RAM
 - Turning machines on/off is expensive (45 seconds)
 - Store application-level and user-level information in RAM
 - Switch on a node from RAM (4 seconds)

Approach: Overhead

- System Oscillation
 - System load unpredictable
 - Perform add or removal of replicas only after the system is relatively stable for a **certain time**
- Suspension to RAM
 - Turning machines on/off is expensive (45 seconds)
 - Store application-level and user-level information in RAM
 - Switch on a node from RAM (4 seconds)
 - **RAM and network devices use power**

Approach: DB Modification

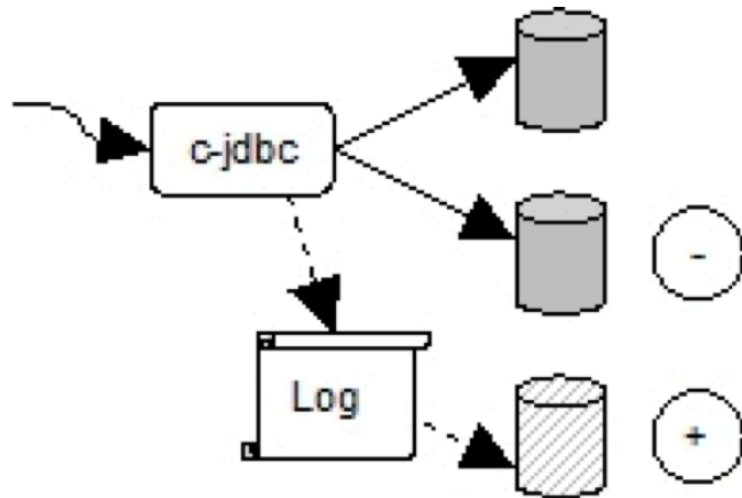


Figure 4 of Gadafi et al. Paper

- Read-only
 - All replicas have same content
 - No modifications
- Read-write
 - Log file replays SQL statements
 - SQL statements executed on new replica

Approach: DB Modification

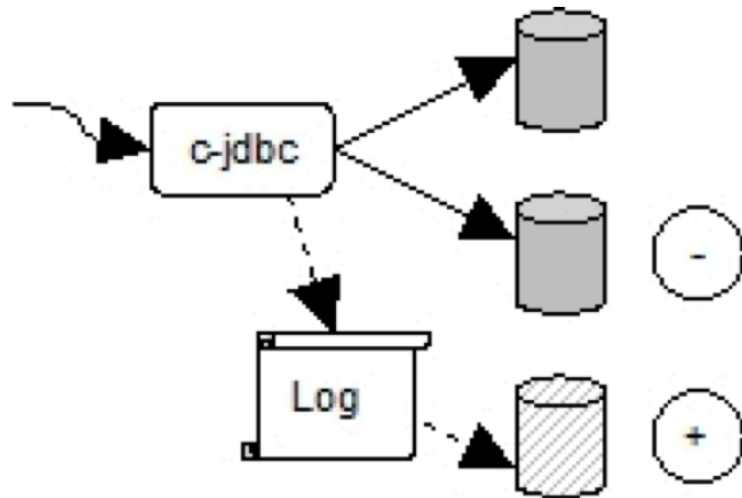


Figure 4 of Gadafi et al. Paper

- Read-only
 - All replicas have same content
 - No modifications
- Read-write
 - Log file replays SQL statements
 - SQL statements executed on new replica
 - **Relatively fast for high read/write ratio**

Evaluation

- RUBiS
 - J2EE application benchmark based on servlets
 - Implements an auction site modeled over eBay
- Static Configuration
 - Few servers: save energy, degrade QoS
 - More Servers: optimize QoS, waste energy
- Dynamic Configuration
 - Tune provides higher-level formalisms for the specification of deployment and management policies
 - Tune dynamically adapts the no. of DB servers

Response Time and Throughput for Static Configuration

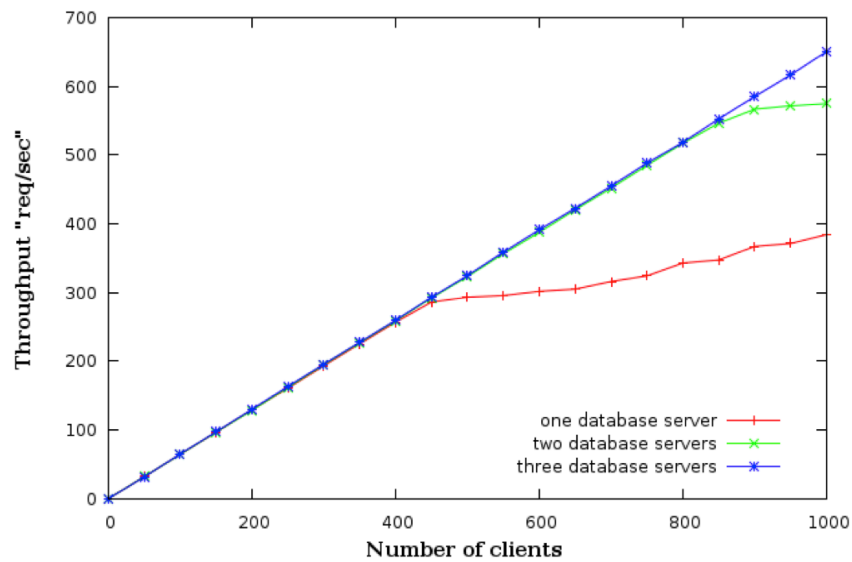


Figure 5 of Gadafi et al. Paper

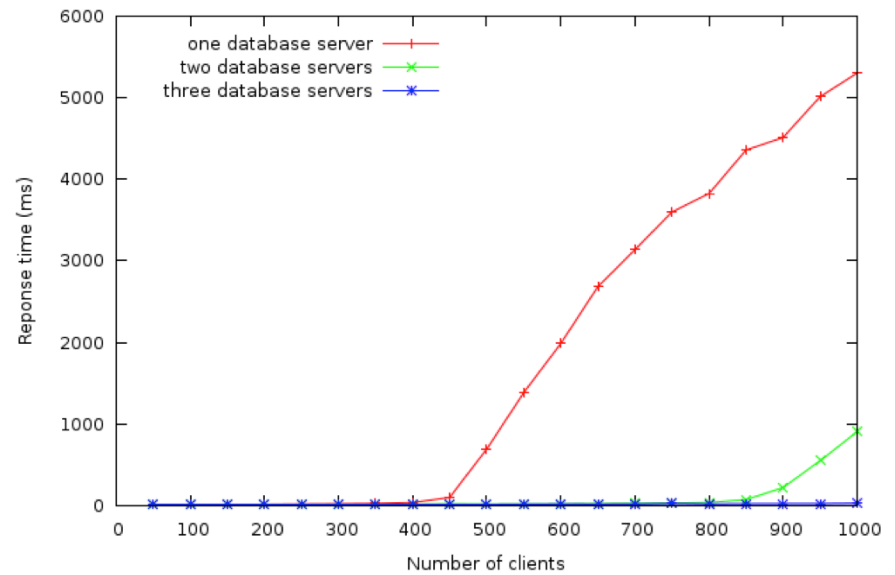


Figure 6 of Gadafi et al. Paper

Energy Consumption

Static Configuration

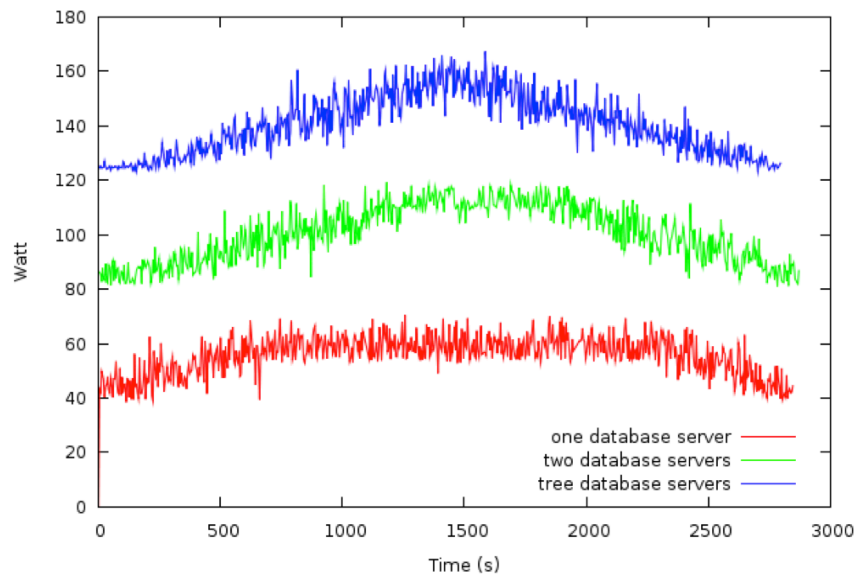


Figure 7 of Gadafi et al. Paper

Dynamic Configuration

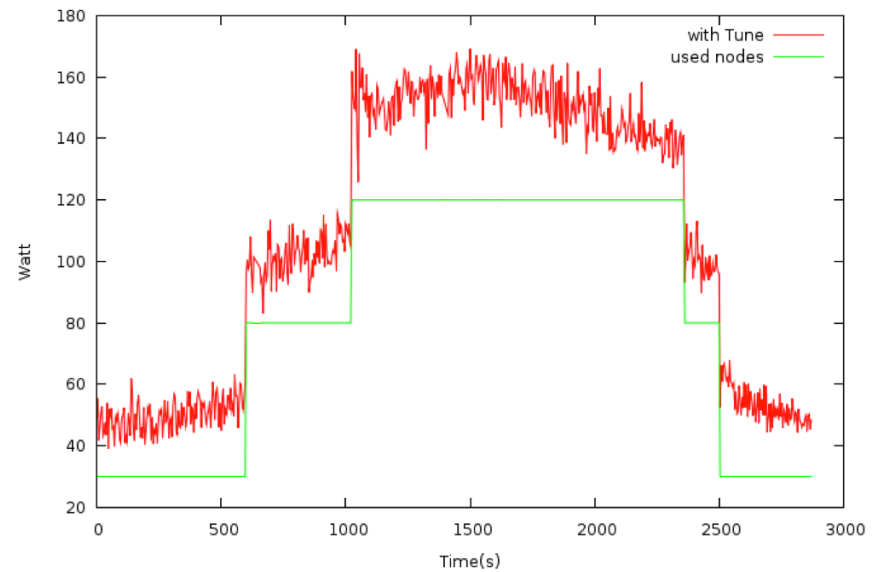


Figure 8 of Gadafi et al. Paper

Comparison of Static and Dynamic Configuration

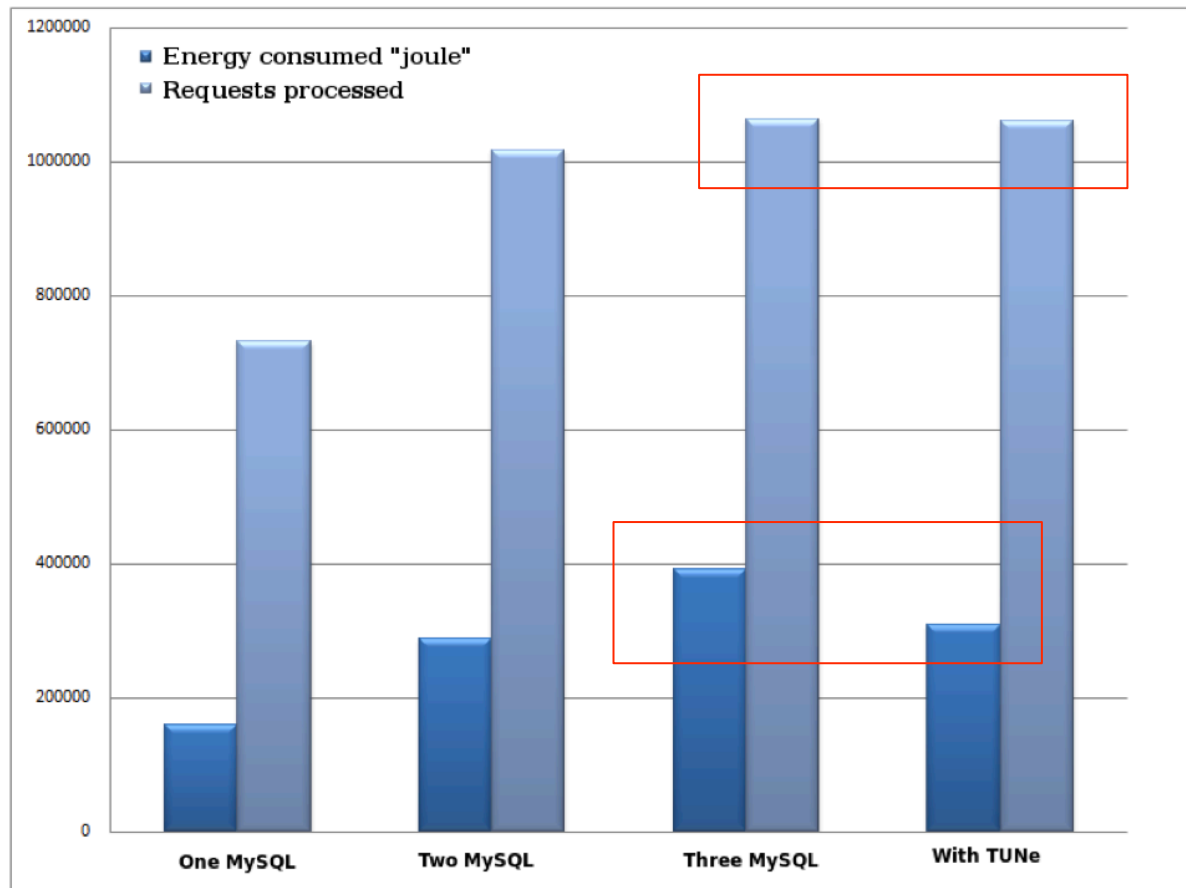


Figure 9 of Gadafi et al. Paper

Discussion

- Overhead of keeping replicas consistent
- Deciding the performance threshold to add/remove replicas
- System Oscillation: Missing in evaluation
- Reducing energy consumed vs. reducing throughput
- Log
 - Central point of failure
 - Logging sql statements is a security risk
 - Length of the log