Goldfish: Serverless Actors with Short-Term Memory State for the Edge-Cloud Continuum

Authors: Cynthia Marcelino, Jack Shahhoud, Stefan Nastic

1

Presented by: Sanya Sinha & Jackson Davis

Introduction to Goldfish & Serverless Computing

2

Outline

3

2

Overview

[Main idea is to address the research challenges (Sec. 2.2)

there are 3]

4

2

Background

[BACKGROUND HERE]

Related Work

[RW SECTION HERE]

Key Contributions

LCM, SIM & Goldfish

7

Critique: Strengths

[RW SECTION HERE]

8

Critique: Weaknesses

[(profs words) one weakness is the error in figure 2. they don't connect the RUNNING state to the ERROR state, but in the paper, they suggest this state transition is possible]

9

2

Challenges in Traditional Serverless Architecture

1

State management limitations

2

Latency and cold start issues

10

3

Resource allocation issues

Solution for Stateful Serverless Actors
Innovative Stateful Architecture

GoldFish takes advantage of the actor model to enable stateful interactions within serverless environment.

Allows actors to:

- Manage their own state
- Communicate directly

Key Features & Innovations of GoldFish

1
Lightweight stateful processing

2
WebAssembly Execution Environment

3
Direct actor communication

GoldFish Architecture & Functionality

13

1

GoldFish Serverless Lifecycle Model Explained
Lifecycle Management Efficiency

GoldFish Serverless Lifecycle model optimizes resource utilization and operational efficiency by clearly defining the transitions between states

14

Invocation Model & Middleware Functionality

1

Asynchronous message handling

2

Actor state management

3

Middleware efficiency

Actor Dispatcher & Buffer Management

Efficient Actor Life Cycle Control

Ensures optimal management of serverless actors by overseeing their life cycle transitions

Robust Message Handling Mechanism

Plays a critical role in maintaining message integrity during high demand periods by queuing incoming messages for busy actors

Advantages of GoldFish Architecture in Edge-Cloud Environments

17

1
Optimized resource utilization

2
Dynamic scaling capabilities

3
Enhanced real-time processing

1

Experimental Setup and Methodology
Controlled Environment Testing

- **Defined parameters for experiments**
- **Utilized consistent hardware configurations**
- **Implemented standardized workloads**
- **Monitored performance metrics**
- **Conducted trial runs**
- **Analysed data for accuracy**

18

1

OpenFaaS & Spin
Performance Benchmarking Insights

Comparative analysis reveals that GoldFish consistently outperforms OpenFaaS and Spin in both latency & throughput

Stats:

Latency reduction: up to 30%

Throughput enhancements: 50% during peak loads

Latency & Throughput Improvements
GoldFish Performance Metrics

Experimental results show that GoldFish achieves latency reductions of up to 30% and throughput improvements of 50% compared to traditional serverless frameworks

20

Thank You