Serving AI on a Distributed Architecture

THE UNIVERSITY OF HONG KONG
DEPARTMENT OF COMPUTER SCIENCE

FYP19060 - Ali Waqas

Supervisor: Dr. Heming Cui - Second Examiner: Dr. Chim T.W.

Background

Figure A shows the steps a typical machine learning application has to go through for **one request** in production.

Problem

Machine learning in production is **slow**, **costly** and unable to handle **high traffic**.

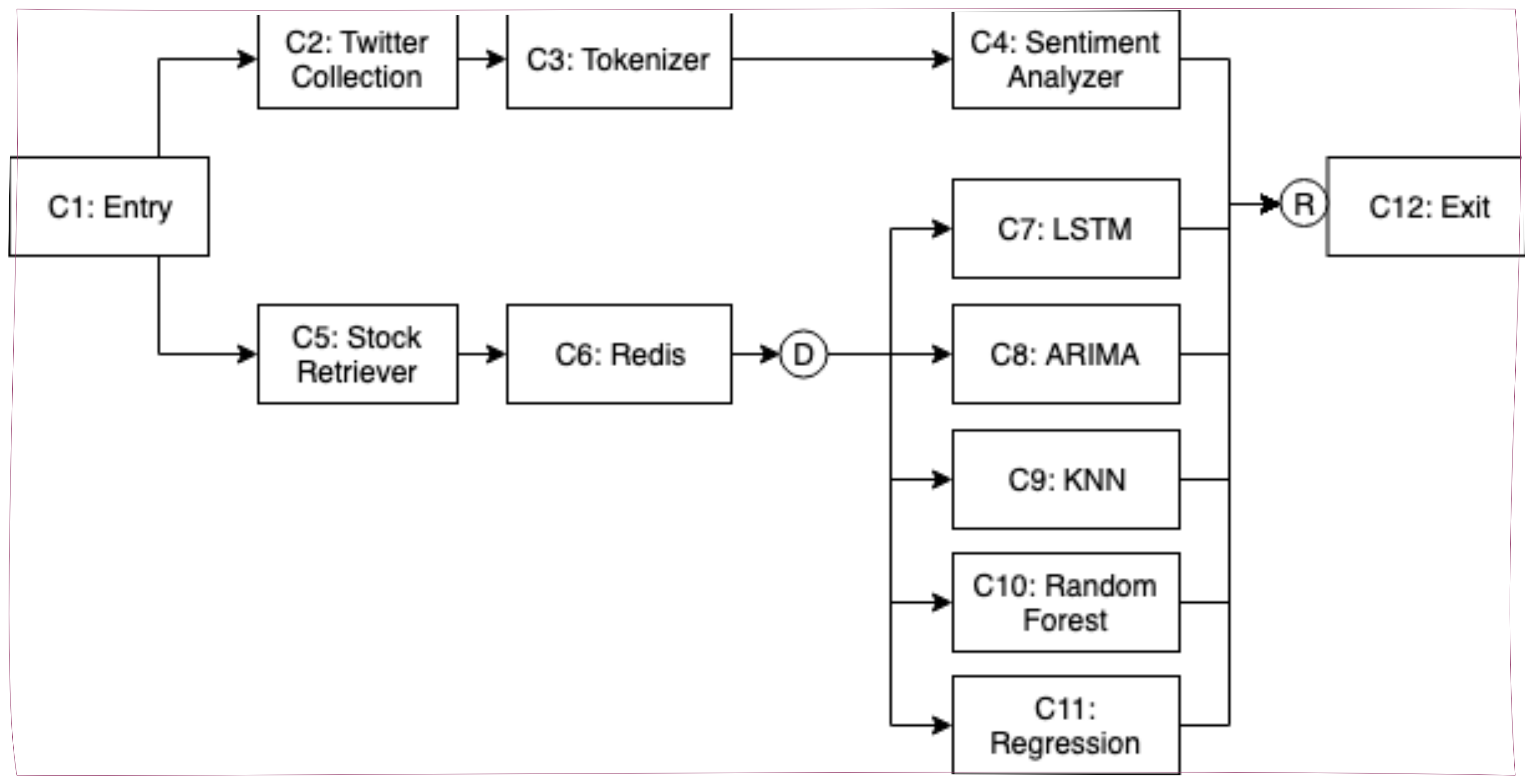


Figure A: Inference pipeline of a stock price prediction service

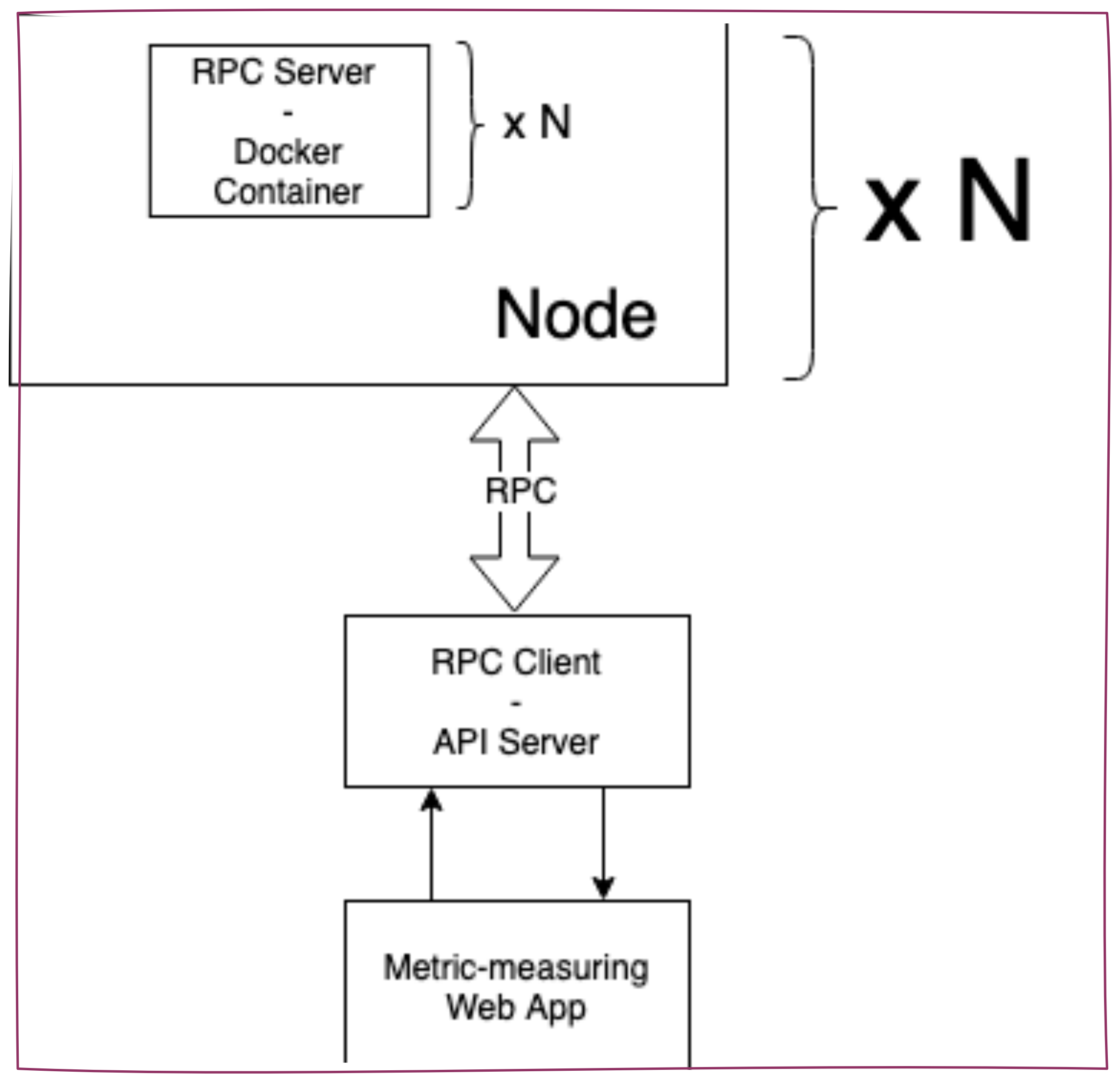


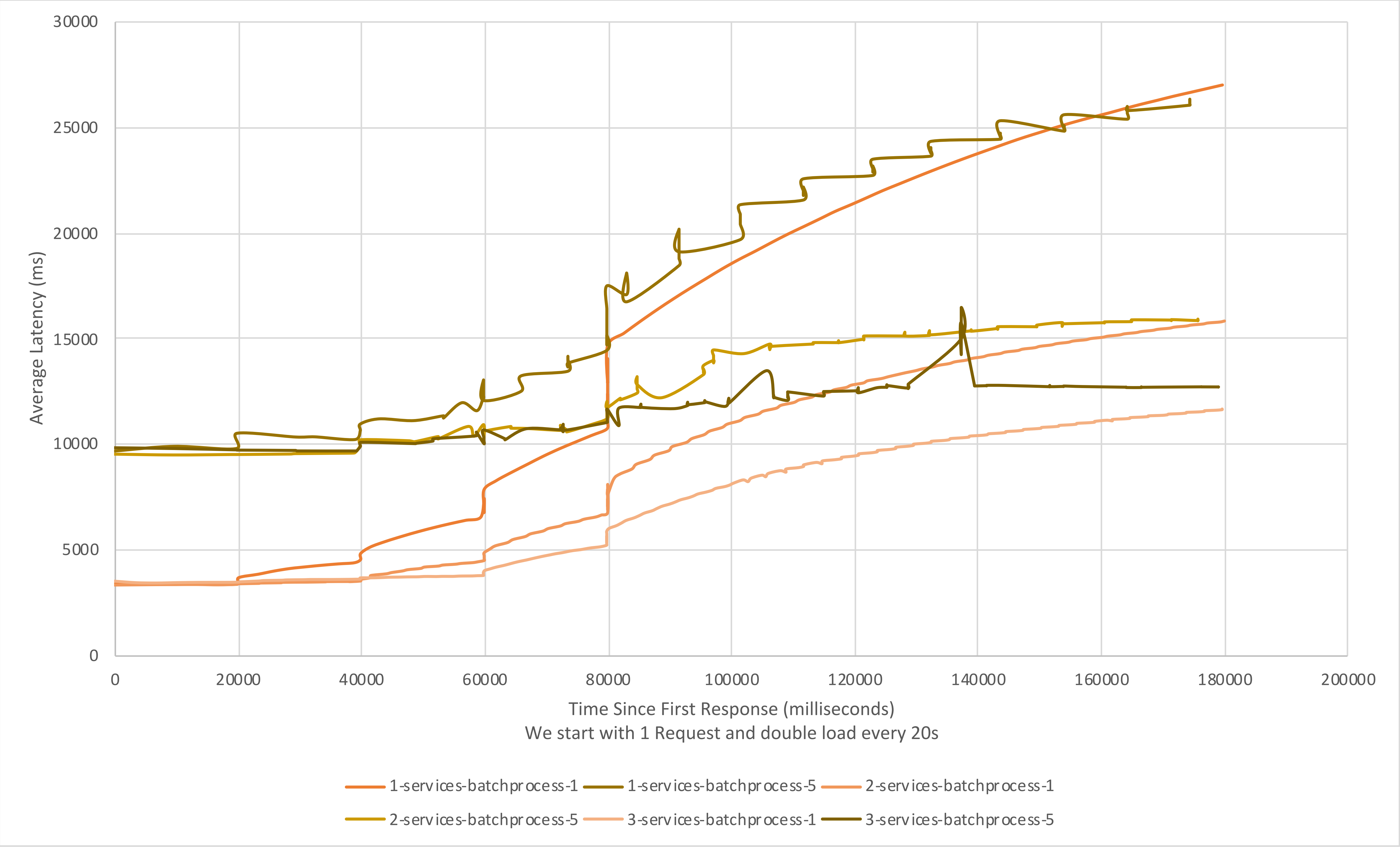
Figure B: Solution Architecture

Solution

1. **Containerise** each pipeline task using Docker
2. **Programmatically deploy containers** using Terraform
3. **Batch process** requests by task using Redis
4. **Remote Procedure Calls** (RPC) using gRPC
5. **Architecture-agnostic metrics** measurement using React/TypeScript
6. **Deploy on HKU Servers** using SSH Tunnelling

Results

- ✓ Reduced latency by more **than 2x**
- ✓ Decreased throughput/latency growth rate from **exponential to linear**
- ✓ **Provided tooling** for further research in distributed systems for microservice-based pipelines


Average Latency (ms)
Time Since First Response (milliseconds)
We start with 1 Request and double load every 20s
1-services-batchprocess-1
1-services-batchprocess-5
2-services-batchprocess-1
2-services-batchprocess-5
3-services-batchprocess-1
3-services-batchprocess-5