universität innsbruck

HyPA: A Hybrid Proactive Autoscaler

Area: Proactive Scaling of Containerized Orchestration Environments

Dominik Gratz, René Hueber

Supervisors: Zahra Najafabadi Samani, PhD
Juan Aznar Poveda, PhD

The trend towards VoIP

- No more classic telephony
- No fixed wiring
- Migrate into the cloud
- Many supported protocols:
  - SIP
  - H.323
  - RTP
  - WebRTC
- Challenges:
  - Efficiency
  - Scalability
  - Maintenance

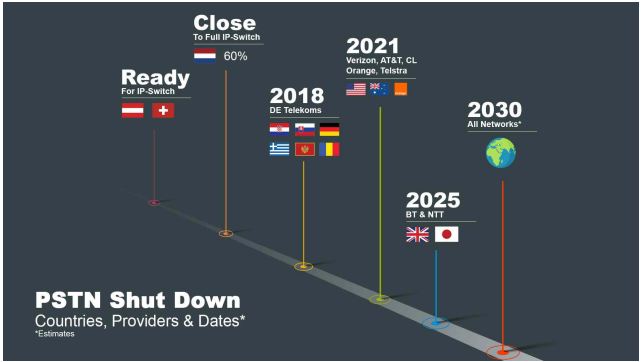


Figure: PSTN Shut Down (As of 2016) [1]

Cooperation with World-Direct

- Subsidiary company of A1
- Manages over 90.000 VoIP ports
- Transitions its telephone infrastructure to **Kubernetes**
  - Microservices
  - API-first approach
  - **Zero-Downtime** architecture

Pros

- Less complexity on the tenant side
- Central maintenance
- High flexibility and **scalability**

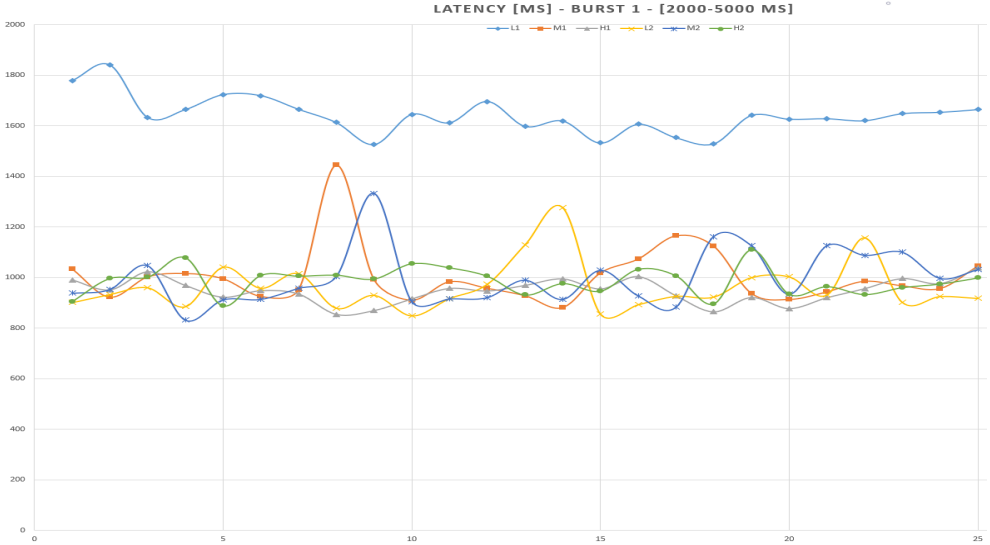
Cons

- Complex infrastructure
- Efficient architectures necessary
- **Timely scaling**

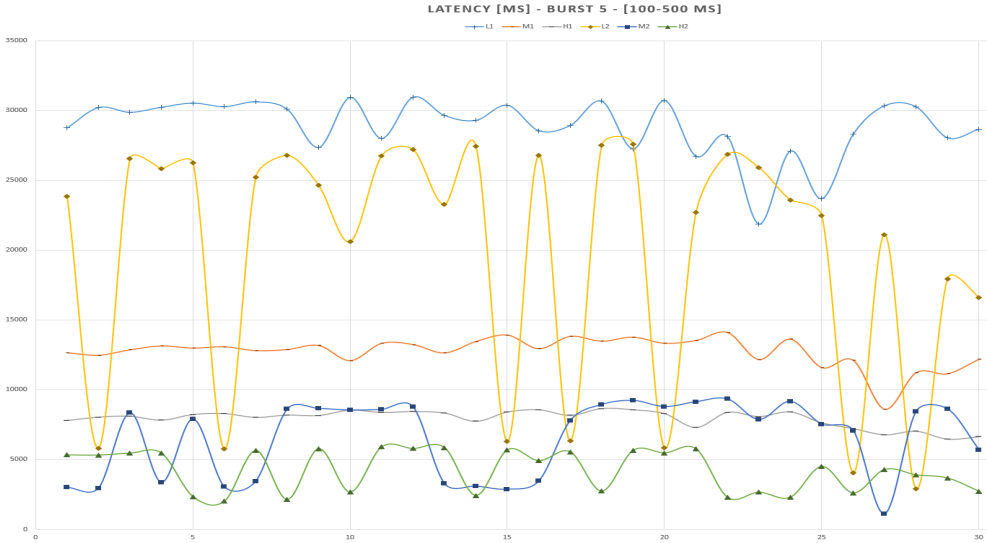
The specific problem: Scaling in real time

- High call traffic:
  - Increased load on the telephone system core
  - Latency spikes
  - Call cancellations
- Scaling methods:
  - Variable number of instances → **horizontal**
  - Variable CPU/memory assignment of a instance → **vertical**
  - Combining both approaches → **hybrid**
- **Reactive** scaling is not enough:
  - Static thresholds
  - Scaling after the system is **already compromised**
  - Unusable for real-time applications

Example load test - Low traffic



Example load test - High traffic



Proactive scaling

- Predict future resource needs → scale **preemptively**
- Ensuring **sufficient** resources during service lifetime
- Different models:
  - Statistical → easy but **slow** (ARMA, ARIMA, etc.)
  - ML based models → complex but **fast**

Related work

- Online Workload Burst Detection for Efficient Predictive Autoscaling of Applications [2]
- Machine learning-based auto-scaling for containerized applications [3]
- Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network [4]

Thesis goal

HyPA

- **Hy**brid scaling:
  - Horizontal $\rightarrow$ number of instances
  - Vertical $\rightarrow$ resource assignment
- **P**roactive approach:
  - Reduce call latency
- **A**utoscaler:
  - Automatically scale services at runtime

Challenges

- Handling sporadic **bursts**
- Resource **conservation**
- Mitigating scaling **oscillation**
- Ensuring no scaling **downtime**

Proposed method

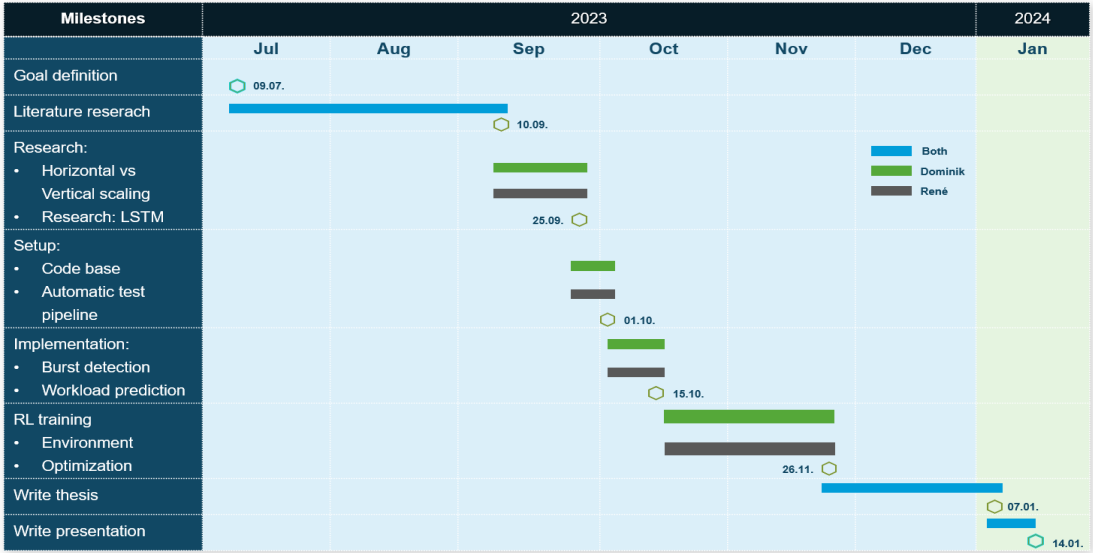
Reinforcement Learning Model

- Environment composed of:
  - Burst detection $\rightarrow$ statistical
  - Workload prediction $\rightarrow$ LSTM
  - Data/Metric connectors
- Environment designed for containerized orchestration environments
- Custom reward function

Deployment

- Train with synthetic data $\rightarrow$ automatic test pipeline
- Deploy in tenant namespace
- Models learns call **patterns** of the tenant

Milestones


Milestones
2023
2024
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Goal definition
Literature reserach
Research:
• Horizontal vs Vertical scaling
• Research: LSTM
Setup:
• Code base
• Automatic test pipeline
Implementation:
• Burst detection
• Workload prediction
RL training
• Environment
• Optimization
Write thesis
Write presentation
09.07.
10.09.
25.09.
01.10.
15.10.
26.11.
07.01.
14.01.
Both
Dominik
René

References I

Blueface VoIP.
Uk pstn shut down.
https://www.blueface.com/blog/ip-network-migration/ (Access: 23.09.2023, 15.38 MEZ) 2023.

Fatima Tahir, Muhammad Abdullah, Faisal Bukhari, Khaled Mohamad Almustafa, and Waheed Iqbal.
Online workload burst detection for efficient predictive autoscaling of applications.
IEEE Access, 8:73730–73745, 2020.

Imtiaz Ahmad Mahmoud Imdoukh and Mohammad Gh. Alfailakawi.
Machine learning-based auto-scaling for containerized applications.
Springer, 32:9745–9760, 2019.

References II

Ashraf A. Shahin.
Automatic cloud resource scaling algorithm based on long short-term memory recurrent neural network.
International Journal of Advanced Computer Science and Applications, 7(12), 2016.

Achim Zeileis Reto Stauffer.
UIBK Latex Beamer Theme.