Steel: Simplified Development and Deployment of Edge-Cloud Applications

Shadi A. Noghabi $^{1,3}$, John Kolb $^2$, Peter Bodik $^3$, Eduardo Cuervo $^3$

1. University of Illinois at Urbana-Champaign
2. University California, Berkeley
3. Microsoft Research
Cloud Services Have Grown

$383 Billion market by 2020 (20% annual growth) \(^1\)

-- increase in the **number** and **variety** of services

Many **complex multi-service** applications have emerged

1. https://www.gartner.com/newsroom/id/3815165
Growing Interest in the “Edge”
Growing Interest in the “Edge”

Why use the Edge?

• **Latency**: Edge is close by
• **Cost**: Reduced cost
  – network to cloud
  – services in cloud
• **Availability**: Continuous service despite disconnects
Industry at Infancy of Edge-Cloud

Cloud growth

Complex multi-service applications

Gap

Steel

Emerging Edge-Cloud environments
What is the Gap?

Many optional (in cloud-only) **optimizations** are **crucial** in the edge-cloud

– More heterogeneity, more failures, limited resources.
– Placement, communication, load-balancing, etc.

→ **Essential** to readily **move** services & dynamically **adapt**
However...

- **Configuring** services is **cumbersome** and **error-prone**.
- Wide diversity & **compatibility constraints** in services.
- **Optimizations** are **manual** and **non-reusable**.

**Steel**: A **simplified & unified** edge cloud framework with **modular** and **automated optimizations**

Integrated with *production* Azure services.
Abstraction & Fabric

Abstraction:
1. **Logical DAG** of the application
   - Main services & their connections
2. **Location** of each service

**Fabric:** physically materializing & connecting services
Optimization Modules

Placement
Where (edge/cloud) to place?
From the large search space

Communication
configure edge-cloud links
Adapt to short-term spikes

Load Balancing
Where to move?
Adapt to long-term changes

...
Optimizer Modules

**Placement**
*Where (edge/cloud) to place?*
From the large **search space**

**Communication**
Configure edge-cloud links
Adapt to short-term spikes

**Load Balancing**
*Where to move?*
Adapt to long-term changes

Added to Steel with ~500 lines of C# code
Evaluation
Experimental Setup

Compare many diverse real world applications:

1. Data persisting
2. Predictive maintenance
3. BLE sensor analyzer
4. Factory monitoring
5. Statistic generator
6. Anomaly detector
Steel reduces config 1.7x to 4.8x vs. current system
Modification Effort

Small, almost constant overhead of ~ 2 lines of config
Conclusion

**Steel**: A unified edge cloud framework with modular and automated optimizations

- Complex multi-service applications
  - Cloud growth
  - Gap
    - Simple Deployment
    - Easy Move & Adapt
    - Optimizations
  - Emerging Edge-cloud environments
What is Next? What is Hard?

1. Cloud **services** offering **Edge solutions**
2. Will a general optimization work for most? How to easily **customize**?
3. Incorporate **developer requirements** like SLAs
4. What about **security & privacy**?
Back up slides
Cloud is Not an One-Size-Fits-All

Latency

< 80ms

< 20ms

< 10 ms

The Cloud is simply **too far**!

> 70 ms round trip time
Cloud is Not an One-Size-Fits-All

Latency

Resources

Not enough resources to use the Cloud

Latency:
- < 80ms
- < 10 ms

Resources:
- Bandwidth
  - 10s – 100s GB/s
- Battery
- Energy
- Heating

Not enough resources to use the Cloud
Cloud is Not an One-Size-Fits-All

Latency

< 80ms
< 10 ms

Resources

Bandwidth
10s – 100s GB/s

Battery
Energy
Heating

Privacy & Security

…