Compute Continuum
Managing Resources between Cloud, Edge, and Endpoint

A **Lecture** and **Demo** by **Matthijs Jansen**

m.s.jansen@vu.nl
http://atlarge.science/mjansen

@Large Research
Massivizing Computer Systems

VU VRIJE UNIVERSITEIT AMSTERDAM

Continuum
Open-source Code

1

Compute and Data Offloading

VU VRIJE UNIVERSITEIT AMSTERDAM

Requirement: Process live video

Problem: Little resources for native processing
Solution: Offload data to other devices

2

2

Offloading Scenarios

VU VRIJE UNIVERSITEIT AMSTERDAM

There are many offloading scenarios
Many questions to answer:
- Where to offload from/to?
- Available resources?
- Application requirements?

3

3

Compute Continuum in 3 Layers

VU VRIJE UNIVERSITEIT AMSTERDAM

Increasing Resource Constraints
Increasing Scale, Bandwidth and Communication Latency to Endpoints
Owned by Users
Owned by Service Providers

Endpoint

- Generates data through users and sensors
- Native processing requires no offloading
- No need to share resources
- Limited resources and energy
- Mobile

4

4

Compute Continuum in 3 Layers

VU VRIJE UNIVERSITEIT AMSTERDAM

Increasing Resource Constraints
Increasing Scale, Bandwidth and Communication Latency to Endpoints
Owned by Users
Owned by Service Providers

- More resources than endpoints
- Low latency to endpoint and cloud

Edge

- Limited reliability
- Offload target for endpoint users AND cloud services

5

5

Compute Continuum in 3 Layers

VU VRIJE UNIVERSITEIT AMSTERDAM

Increasing Resource Constraints
Increasing Scale, Bandwidth and Communication Latency to Endpoints
Owned by Users
Owned by Service Providers

- High latency to endpoints
- Owned by cloud provider (privacy?)
- Near-infinite resources
- Variety of services

Cloud

6

6

Compute Continuum in 3 Layers

Increasing Resource Constraints

Increasing Scale, Bandwidth and Communication Latency to Endpoints

Owned by Users | Owned by Service Providers

Endpoint | Edge | Cloud

7

7

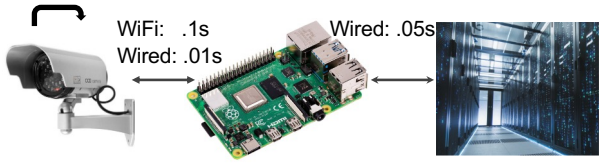
Offloading Choices: Big Impact

Time to process 1 image

Native: 1s | Edge: .5s | Cloud: .2s

WiFi: .1s
Wired: .01s

Wired: .05s



Endpoint | Edge | Cloud

8

8

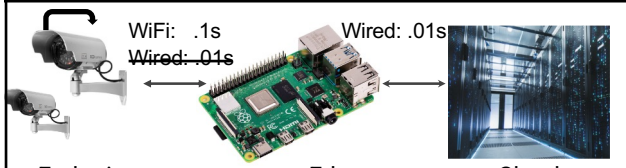
Offloading Choices: Big Impact

Time to process 2 images in parallel

Native: 1s | Edge: .8s | Cloud: .3s

WiFi: .1s
~~Wired: .01s~~

Wired: .01s



Endpoint | Edge | Cloud

9

9

Offloading Choices: Big Impact

How do I find a deployment in the Compute Continuum that works best for my requirements?

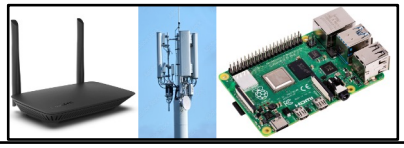
10

10

How Do I Use the Compute Continuum?

Software dev.
How to test software across devices?

Service provider
How to offer services across all devices?



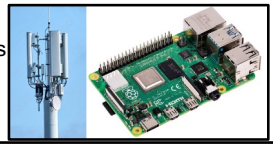
11

11

Challenges in the Compute Continuum

Q1: Where to deploy?
Need access to infrastructure
→X86 vs ARM
→Wired vs Wireless
→**Expensive!**

Q2: What software to deploy?
Layers have different requirements
→Tensorflow vs Tensorflow Lite
→Kubernetes vs KubeEdge
→**Takes a lot of time!**



12

12

Challenges in the Compute Continuum

Q1: Where to deploy?
Need access to infrastructure
→X86 vs ARM
→Wired vs Wireless
→**Expensive!**

Q2: What software to deploy?
Layers have different requirements
→Tensorflow vs Tensorflow Lite
→Kubernetes vs KubeEdge
→**Takes a lot of time!**

Conclusion:
Problems are difficult to solve
→**What do we do?**

13

13

Continuum

Automated infrastructure and software exploration for the compute continuum

https://github.com/atlarge-research/continuum

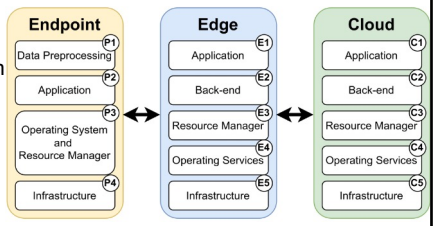
14

14

Reference Architecture

Compute Continuum Abstraction layers

Changing any layer impacts offloading



The SPEC-RG Reference Architecture for the Compute Continuum, Jansen et al, CCGRID'23

15

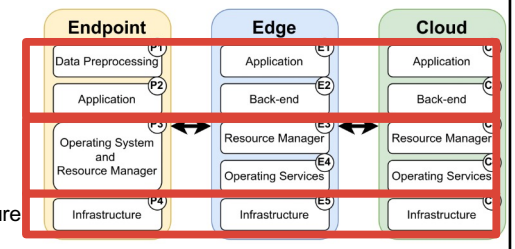
15

Reference Architecture Pillars

Workload

Software

Infrastructure



16

16

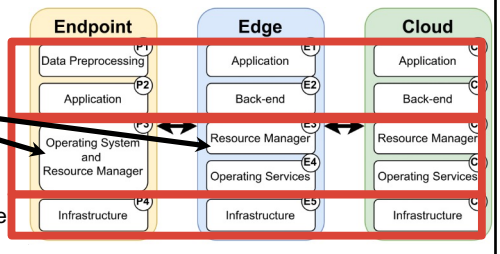
Reference Architecture Pillars

Workload

#Tenants

Software

Infrastructure



17

17

Reference Architecture Pillars

Goal

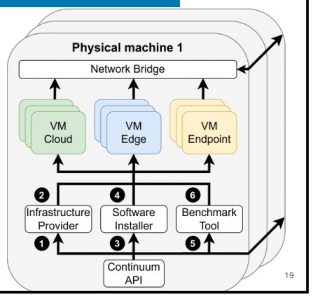
Build a deployment framework where users can change any layer

18

18

The Continuum Framework

Execution phases:
1. Provision **infrastructure**
2. Install **software**
3. Benchmark **applications**

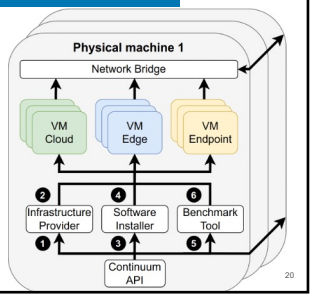


19

19

The Continuum Framework

Design principles
1. Accurate
   → Deploy on hardware
1. Flexible
   → Emulate hardware and networks you have
1. Automated
   → Script deployments
1. Extendable
   → Components as modules



20

20

Today's Use Case

Video camera processing
with image classification

- Generate X frames per second
- Analyze each frame with ML

ML can't run on endpoint
So, offload to cloud

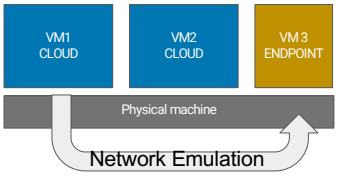
Using cloud-native technologies

21

21

Step 1: Infrastructure Provisioning

```
config-file.cfg
[infrastructure]
provider = Local / QEMU
cloud vms          = 2
cloud cores        = 4
cloud memory       = 16
endpoint vms       = 1
endpoint cores     = 1
endpoint memory    = 0.5
network = 4G
```

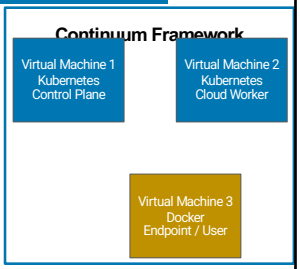


22

22

Step 2: Software Installation

```
config-file.cfg
resource manager = kubernetes
```

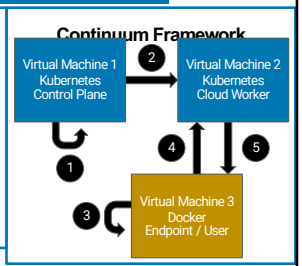


23

23

Step 2+3: Software and Benchmark

Software: Kubernetes + Docker
1. User registers app to K8s
2. Kubernetes deploys server component on cloud worker
3. User deploys generator component on endpoint
4. Generator sends data to cloud
5. Cloud processes data, sends result back to endpoint



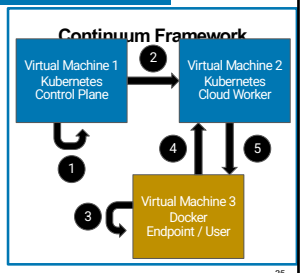
24

24

Step 3: Benchmark Execution

config-file.cfg

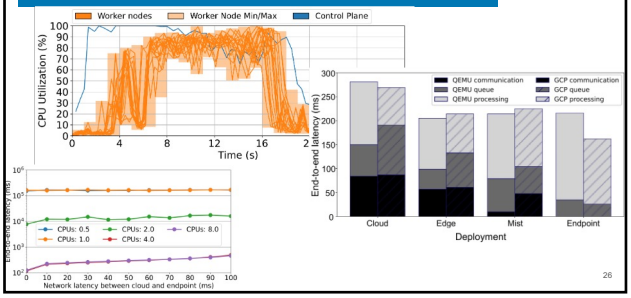
application = image class.
app cloud cpu = 1.0
app cloud memory = 3.0
app endpoint cpu = 1.0
app endpoint memory = 1.0



25

25

Step 4: Data Gathering & Visualisation



26

26

Today, You Will

1. Deploy image classification app with Continuum, on VMs and Kubernetes
2. Get familiar with Kubernetes
3. Visualize cloud operations with Prometheus and Grafana
4. Deploy and instrument apps by hand on Kubernetes

docker

kubernetes

27

27

Before We Start The Tutorial

- We provide access to hardware
- Send your PUBLIC SSH key to m.s.jansen@vu.nl (as text in the email and not as attachment)
- May need to group up depending on no. participants. If so, only 1 person per group sends SSH key

START HERE:

https://github.com/atlarge-research/continuum/tree/tutorial-2024

SSH

28

28