TECHNISCHE UNIVERSITÄT WIEN

E-Commerce Assignment

188.427 2016WS

12.10.2016

Filip Rydzi
eclecture-tutor@ec.tuwien.ac.at

Content

- Edge computing
- Infrastructure
  - Resources
  - SLAs
  - Workload
  - Failures
  - Migrations
- Your tasks

Edge computing

- internet of things
- bring cloud near to the user
- offload compute intensive operations
- low-latency, high-bandwidth

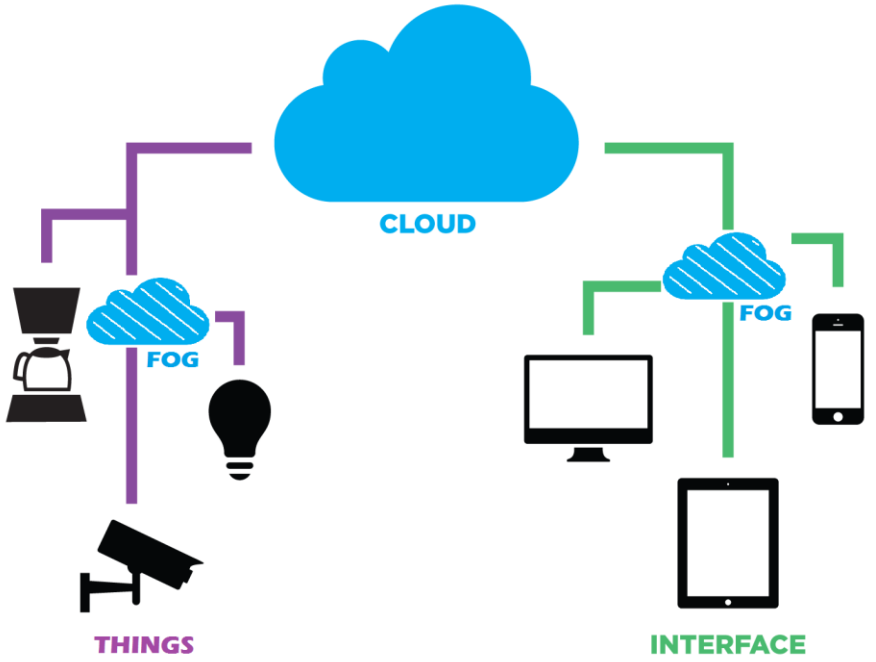


Figure 1: edge computing [1]

Your task in short

- develop an edge controller to efficiently distribute the workload among the edges by considering
  - expected workloads
  - failures, which are to occur
  - load balancing and migration of VMs

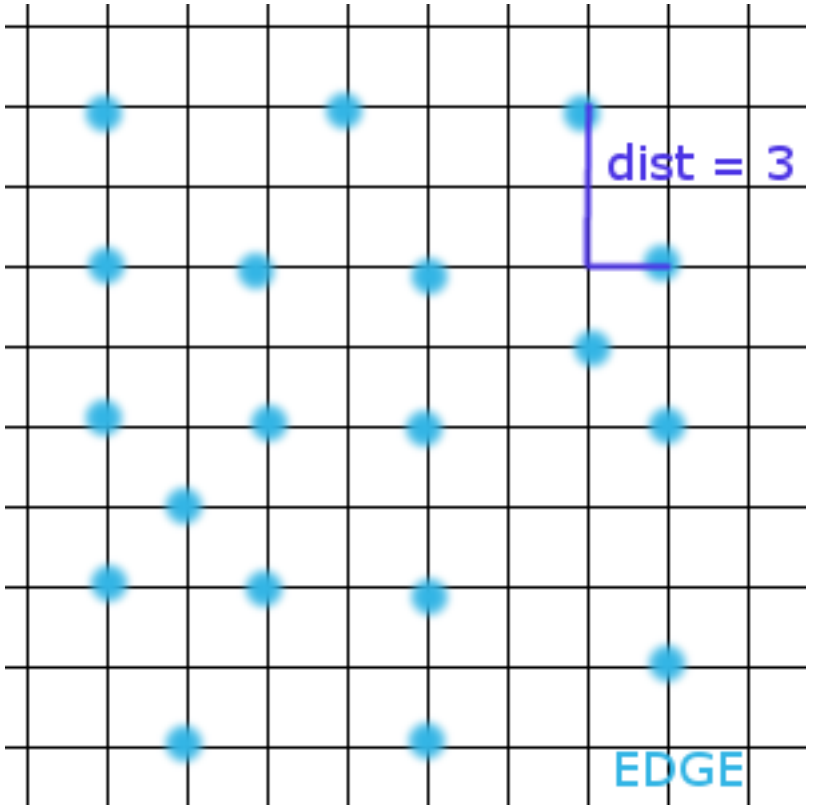
Infrastructure

Resources

- Virtual machines
  - Size, memory, CPU, page dirtying rate, running time …
  - Origin of the request
- Physical machines
  - Size, memory, CPU…
  - Energy utilization
- *Big datacenter (optional)*

Resources

- Edge
  - Max 10 PMs
  - Energy utilization
  - Location



SLAs

- Agreed VM characteristics: e.g. CPU: 3GHz, memory: 4GB, network bandwidth: 54 Mbit/s
- Network bandwidth between the edges (e.g. 600 Mbit/s)
- Availability

$$availabilty = \frac{uptime}{uptime + downtime}$$

- Latency (e.g. 70ms)
- Mean time to recover from failure (MTTR)

Workload

- requests for VMs were generated randomly by uniformly distributing the creation time and duration
- specifications of the requested VMs were modelled by normally distributing each resource type (CPU, memory,…)

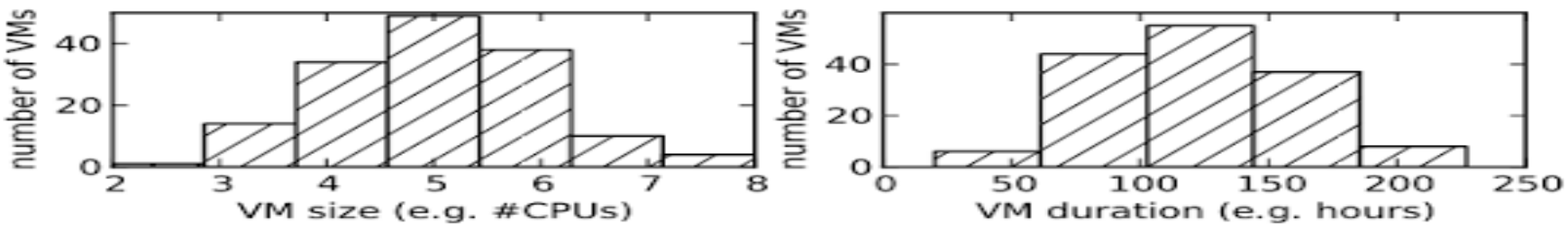


Figure 3: workload distribution

Failures

- Hardware/PM failure
  - the VMs should be handled according to your developed strategy
- Edge failure
- Anomaly detection system

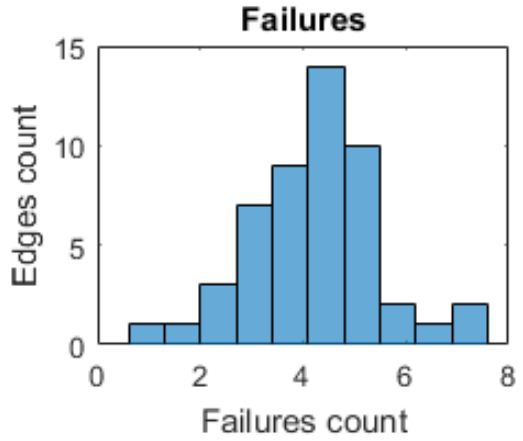


Figure 3: failure distribution

Migrations

- simplified version of **pre-copying algorithm** as baseline
- Basic idea
  1) Create VM memory image and push it to destination

$$V_0 = V_{mem}$$

$$T_0 = \frac{V_{mem}}{R}$$

Migrations

- simplified version of **pre-copying algorithm** as baseline
- Basic idea
  2) Push pages dirtied during last “push” step to destination until page dirtied falls below a threshold

$$V_i = D \,.\, T_{i-1}$$

$$T_i = \frac{V_i}{R}$$

R should be chosen by yourself as a function of distance between source and destination

  3) Stop VM for a short time and push pages dirtied in the previous round

Migrations

- Simplified energy model

$$V_{mig} = \sum_{i=0}^{n} V_i$$

$$U_{mig} \approx V_{mig}$$

- Goal: minimize the U_{mig}

Your tasks

Presentation 1: 19.10.2016

- Find your peers (groups of 3 people)
- Create a git repo (add eclecture-tutor@ec.tuwien.ac.at as guest/viewer)
- Select a baseline for your algorithm to handle the failures (see paper for baseline candidates)
  - Give some ideas for improvements of the baseline

Presentation 2: 16.11.2016

- Develop algorithm to handle the failures
  - Maximize the number of failures your infrastructure is able to handle
  - Minimize the latency
  - Minimize energy consumption
- Compare your algorithm to the baseline
- Present methodology how to test your algorithm
- Select a sampling time of your algorithm (1 minute, 5 minutes, 1 hour…)

Presentation 2: 16.11.2016

- Develop a controller, which is able to handle the failures utilizing
  - Your algorithm
  - The baseline
- Present the infrastructure you developed
- Discuss your implementation choices like programming language…
- Select SLAs

Presentation 3: 14.12.2016

- Perform first simulation for a short period of time (a couple of hours)
  - Generate & distribute the workload
  - Generate & handle failures
  - Run simulation one utilizing your algorithm and once the baseline
- Give some suggestions for improvements based on the results

Presentation 4: 18.01.2017

- Perform second simulation for a long period of time (at least one week)
  - Generate & distribute the workload
  - Generate & handle failures
  - Use at least 3000VMs, 1000PMs running in 100 edges
  - Run simulation one utilizing your algorithm and once the baseline
- Evaluation
  - Show what you have improved since the first simulation
  - Estimate the maximum amount of failures your controller is able to handle (under the selected SLAs) and compare it against the baseline

Your Output

- A short presentation
  - Describe what you have done
  - Max 10 minutes
- Implementation
  - Create a git repo (add eclecture-tutor@ec.tuwien.ac.at as viewer/guest)
- README
  - Instructions describing how to execute your solution

Summary

- Groups of 3 students

- Presentation 1: 19.10.2016
- Presentation 2: 16.11.2016
- Presentation 3: 14.12.2016
- Presentation 4: 18.01.2017

- 4-5 groups will be selected to present for each deadline

Questions ?

- See assignment paper in TISS

- Use TISS forum to ask questions during the semester
- Ask before/after the lecture
- eclecture-tutor@ec.tuwien.ac.at

Thank you for your attention!

References

[1] http://iot-labs.com.my/wp-content/uploads/2016/03/cloud-fog-and-things.png