B-Scale: Bottleneck-aware VNF Scaling and Flow Routing in Edge Clouds

Chen Chen, Lars Nagel, Lin Cui, Posco Tso

July 8, 2022, Coseners'22

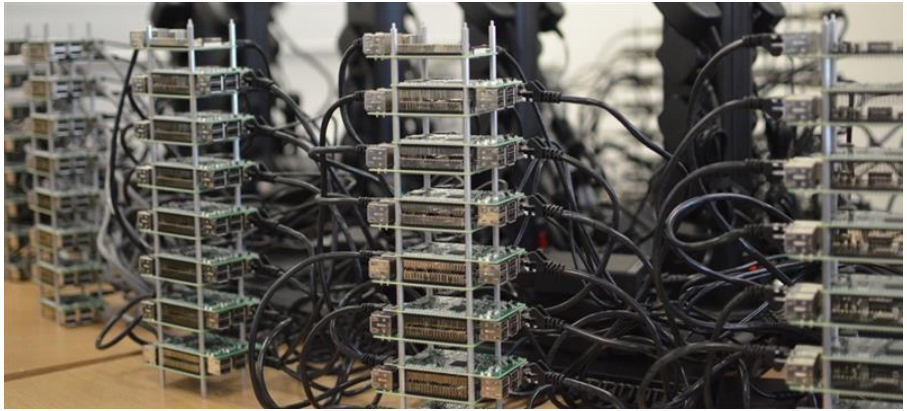
Motivation → Idea → Results → Conclusion

Motivation

Loughborough University

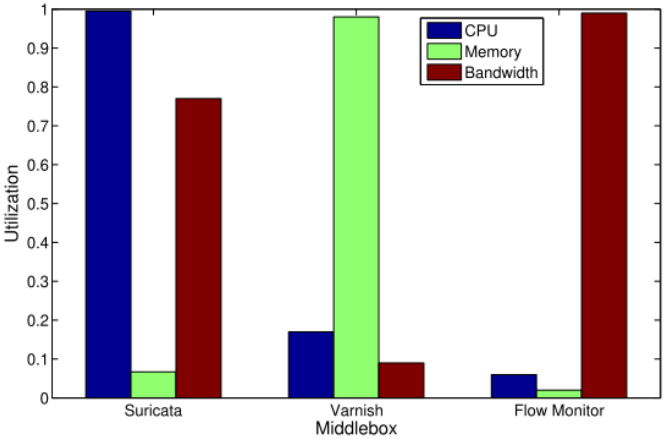
暨南大學 JINAN UNIVERSITY

- Resource is scarce at the edge.
- Constrained resources with low richness of CPU, RAM and bandwidth such as single-board computers.
- Improving resource utilization at the edge is important.



Motivation

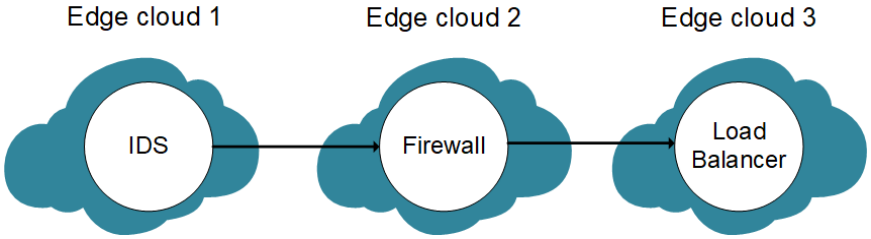
- VNFs are bottlenecked on physical resources.
- Adding irrelevant resources cannot improve performance.
- How to scale VNFs? Vertical or Horizontal? Or Hybrid?
- We are motivated to reduce the resource waste.



Source: ENSC: Multi-Resource Hybrid Scaling for Elastic Network Service Chain in Clouds

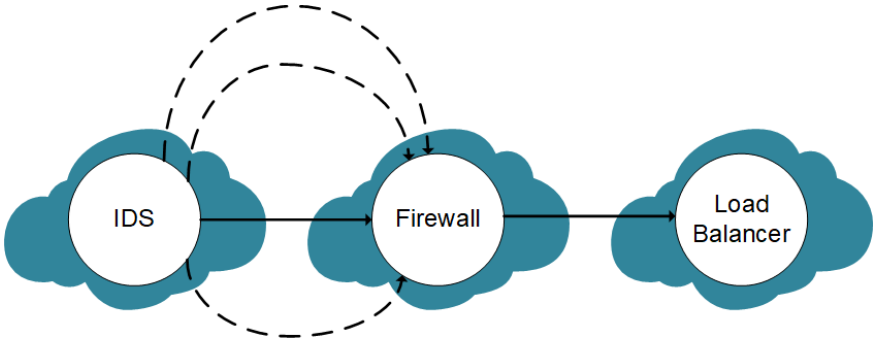
Motivation

- Only using shortest path leads to congestion on hotspot links.
- It also wastes bandwidth resources on spare links.



Shortest path

- Using all paths makes it time-consuming.



All paths

Contributions

Loughborough University

暨南大學 JINAN UNIVERSITY

- **New insights:** We investigate that VNFs can be bottlenecked on different physical resources and hence we need to prioritize vertical and horizontal scaling. We also aim to take advantage of spare links without significantly increasing the execution time.

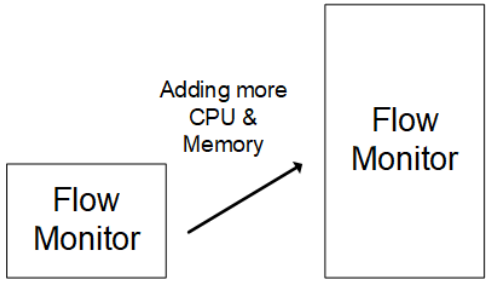
- **New method:** We devise B-Scale that senses the bottleneck and the VNF utilization.

- **Improved performance:** B-Scale improves the VNF utilization by 15% while optimizing end-to-end latency (9% more to optimal) and execution time.

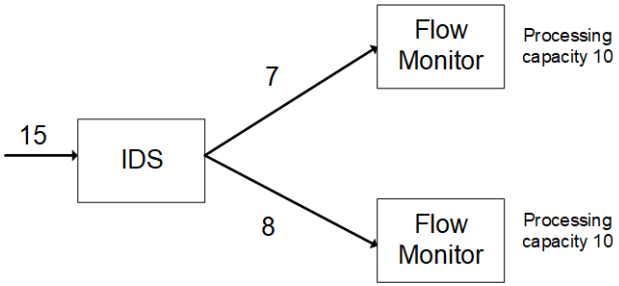
Motivation | Idea | Results | Conclusion

VNF Scaling

- Scale VNFs based on their category.
- I/O-bound: Flow monitor
  - Avoid vertical scaling because it cannot improve performance.
- Non-I/O bound: DPI, IDS
  - Prioritize vertical Scaling.
  - Reduce waste due to resource reservation.



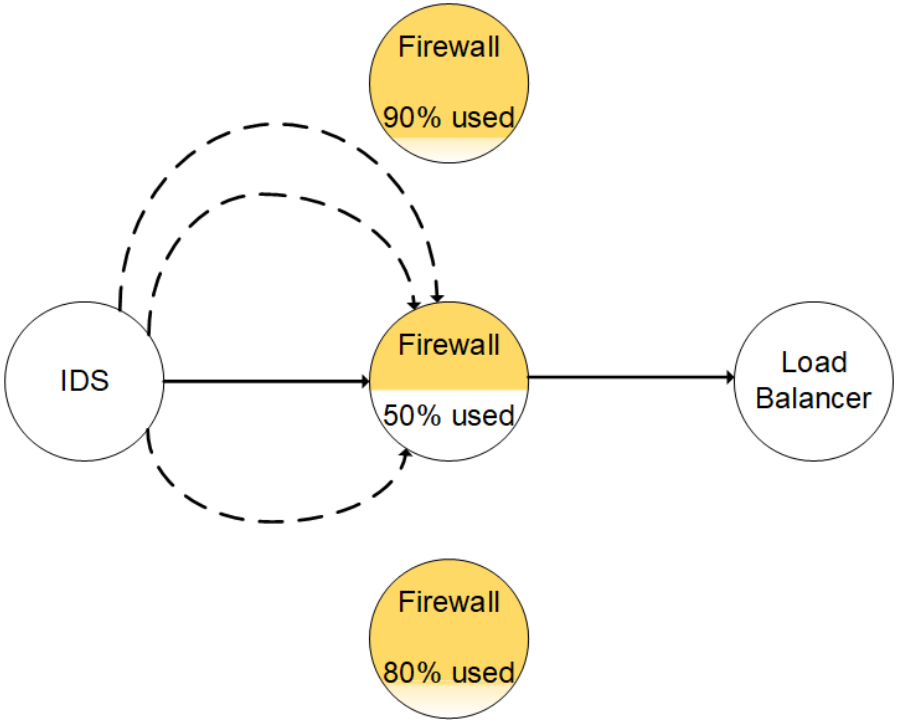
Vertical scaling



Horizontal scaling

Flow Routing

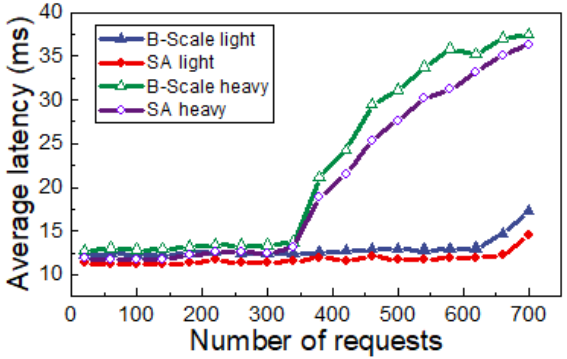
- Choose the instance with lowest utilization rate at next hop
- Use the next shortest available path
- Stop searching at k-th shortest path



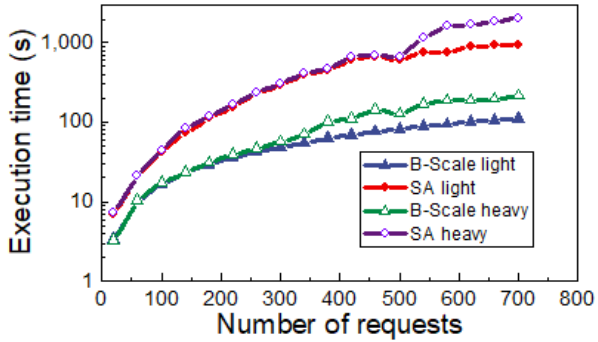
Flow routing use non-shortest path

Motivation → Idea → Results → Conclusion

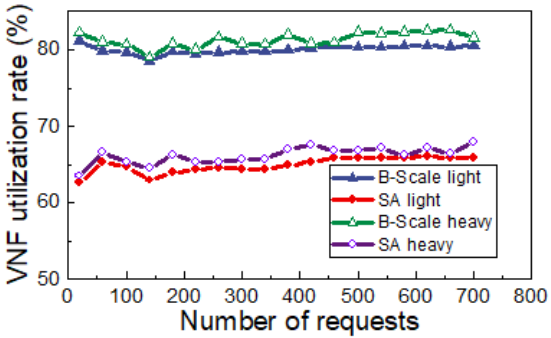
Offline Setting



Avg. latency

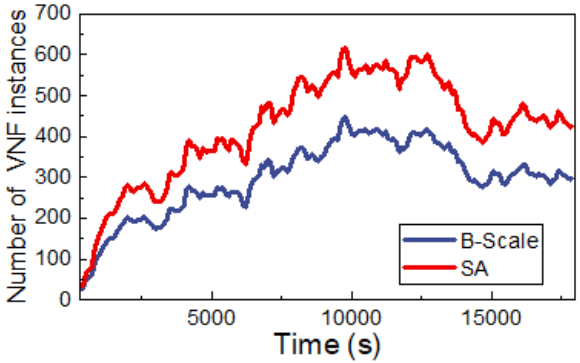


Execution time

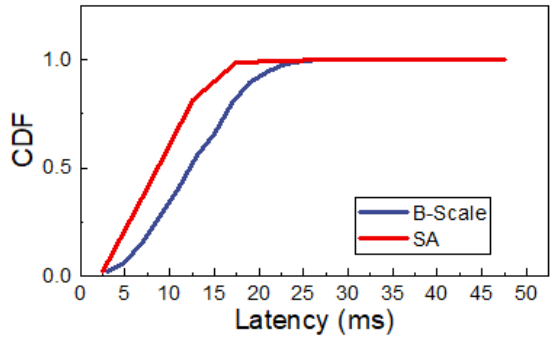


VNF utilization rate

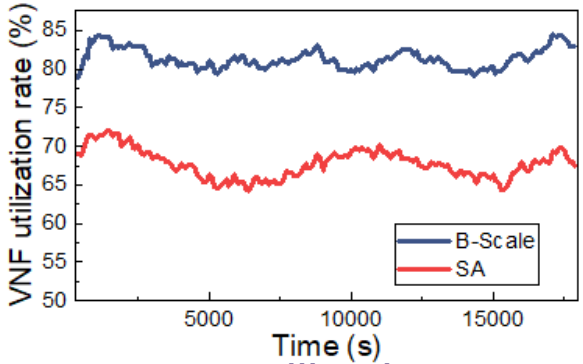
Online Setting



Number of VNF instances



CDF of latency



VNF utilization

Motivation → Idea → Results → Conclusion

Summary and conclusions

- We demonstrate a bottleneck-aware and resource-efficient scheme.
- Improved VNF utilization when compared with other state-of-the-art.
- Near-optimal latency.
- Improved execution time.

Questions?

Thank you!

Chen Chen (c.chen@lboro.ac.uk) (I’m looking at Research Associate jobs, please feel free to get in touch)