Towards Network-Aware Resource Provisioning in Kubernetes for Fog Computing Applications

Abdul Ahad Ayaz

Computer Networks Group
Universität Paderborn

Outline

- What is Fog Computing and IoT?
- Why using Kubernetes for Fog Computing Applications?
- What are the main components of Kubernetes?
- How Kubernetes schedule resources and its drawbacks?
- New scheduling technique
- Comparison of new scheduling technique with other solutions
- Conclusion

What is Fog Computing and IoT?

Fog Computing

- Concept of Fog Comping
- Fog Computing vs Cloud Computing
  - Cloud: OpenStack, AWS
  - Containers: Kubernetes, Docker Swarm

- IoT based Applications
- IoT Application Resource:
  - VM
  - Containers

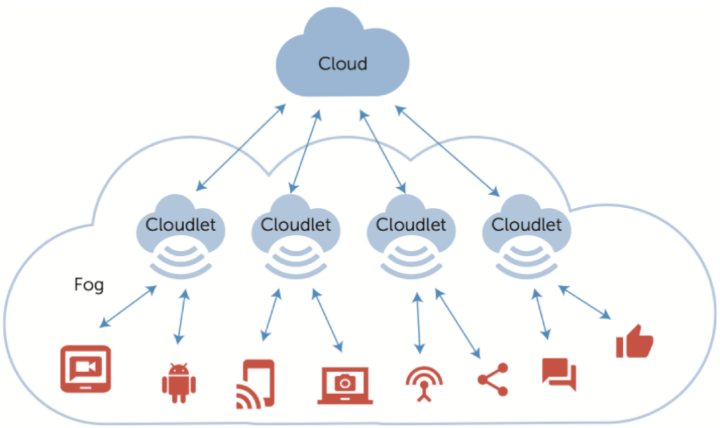

Fig 1: High-level view of Fog Computing[2]

Why using Kubernetes for Fog Computing Applications?

Kubernetes

- Architecture
  - Master Node
  - Worker Node
- Main Components
  - Controller Manager
  - Kube Scheduler
  - Pod
  - Image Registry
- Orchestration
  - Starting/stoping of applications
  - Scalabilty of applications
  - Load management
  - Health monitoring

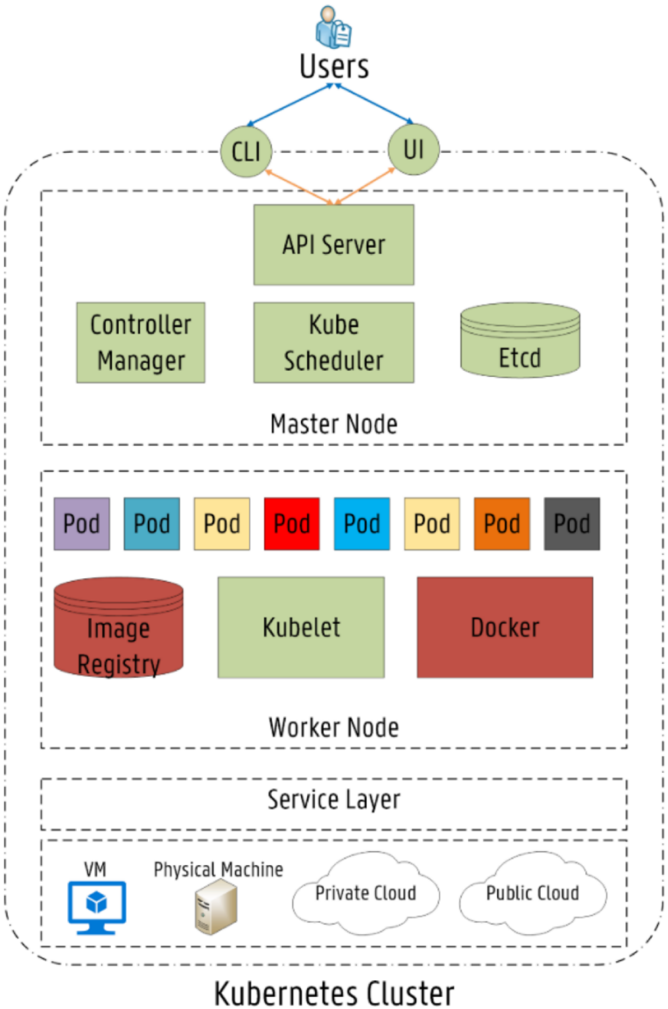

Fig 2: Kubernetes Architecture[1]

How Kubernetes schedule resources and its drawbacks?

Kubernetes Resource Provisioning

- Default Scheduler: “Kube Scheduler”
- Used for Pod deployment across Worker Nodes
- Node selection criteria:
  - Node Filtering
    - Pod Fits Host Ports
    - Pod Fits Resources
  - Node Priority/Scoring
    - Least Request Priority
    - Image Locality Priority
- Drawbacks
  - No network resources consideration

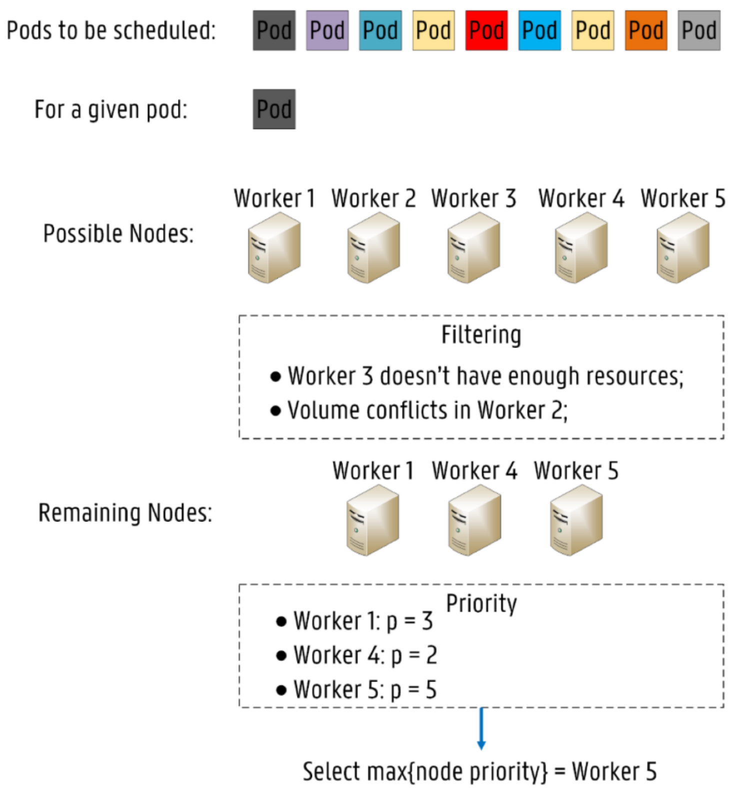

Fig 3: Kubernetes Scheduler workflow[1]

Need for New Scheduling Technique

How to Add New Scheduling Technique?

- Building new scheduler from scratch
- Extending Kube Scheduler
  - Adding new filters or priorities
  - Calling external scheduling process by Kube Scheduler

Network-Based Scheduler

- Implemented using Go Language
- Deployed as Pod with two container
  - Extender
  - Network-scheduler
- Node Selection Criteria
  - Minimum Round Trip Time (RTT)
  - Enough Bandwidth support

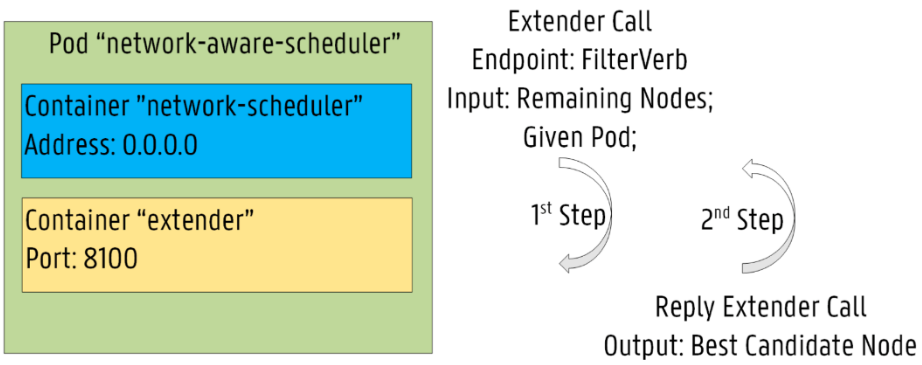

Fig 4: Pod structure of Network-based Scheduler

Fog Computing Infrastructure

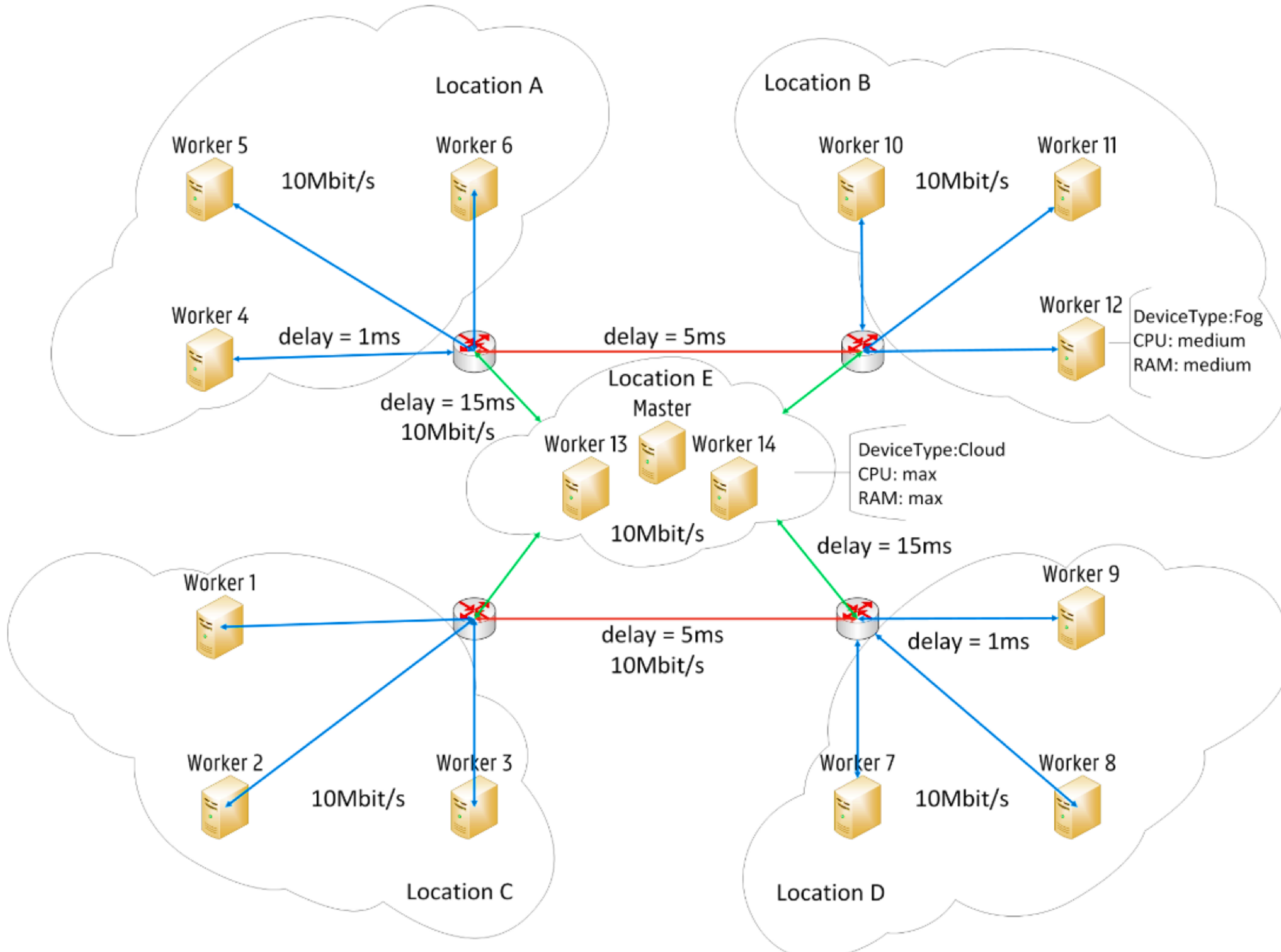

Fig 5: Kubernetes-based Fog Computing Infrastructure[1]

Comparison of new scheduling technique with other solutions

Evaluation of Network-based Scheduler

- Smart City Scenario
  - Collection of air quality data
- Comparison of Three Schedulers
  - Kube Scheduler
  - Network-based Scheduler
  - Random Scheduler

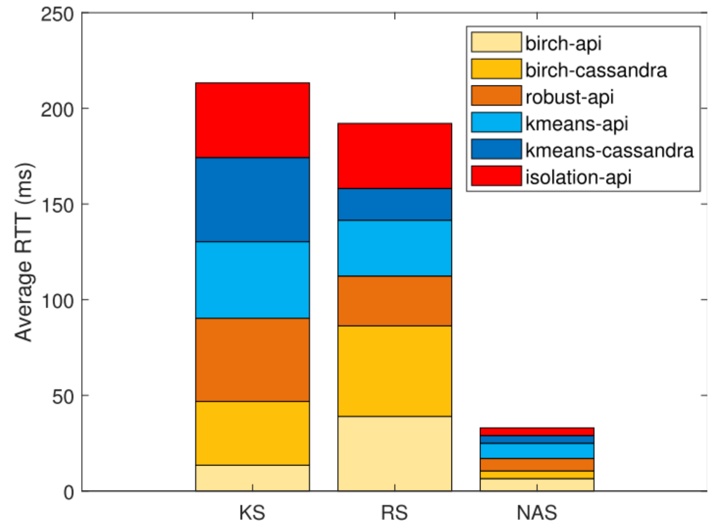

Fig 6: RTT of service deployment by different schedulers[1]

Results

Table 1: Execution time of different schedulers[1]

Scheduler	Extender decision	Scheduling decision	Binding operation	Pod Startup Time
KS	-	2.14 ms	162.7ms	2.02 s
RS	5.32 ms	7.71 ms	178.2ms	3.04 s
NAS	4.82 ms	6.44 ms	173.1ms	2.10 s

Comparsion

- Based on Orchestrator
  - Fogernetes [3]
- Based on Scheduling Techniques
  - Scheduler [5]
  - DYSCO [6]

Conclusion

- Fog Computing enables IoT applications
- Network-based Scheduler adds extra execution time
- Optimized technique considering default scheduler
- Works along side the default scheduler

References

[1] J. Santos, T. Wauters, B. Volckaert, and F. De Turck. “Towards network-Aware resource provisioning in kubernetes for fog comput- ing applications”. In: http : / / physics . nist.gov/Document/sp811.pdf. IEEE Conference on Network Softwarization Un- leashing the Power of Network Softwariza- tion, NetSoft 2019, 2019.

[2] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar. “Mobility- Aware Application Scheduling in Fog Com- puting”. In: (2017). https://ieeexplore.ieee.org/document/7912261

[3] C. Wöbker, A. Seitz, H. Mueller, and B. Bruegge. “Fogernetes: Deployment and man- agement of fog computing applications”. In: https : / / ieeexplore . ieee . org / document / 8406321. IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018, 2018.

[4] A. Reale, P. Kiss, M. Tóth, and Z. Horváth. Designing a decentralized container based Fog computing framework for task distribu- tion and management. Tech. rep. http:// www.naun.org/main/UPress/cc/2019/ a022012-044.pdf. 2019.

[5] D. Haja, M. Szalay, B. Sonkoly, G. Pongracz, and L. Toka. “Sharpening Kubernetes for the Edge”. In: https://dl.acm.org/doi/ 10.1145/3342280.3342335. SIGCOMM 2019 - Proceedings of the 2019 ACM SIG- COMM Conference Posters and Demos, Part of SIGCOMM 2019, 2019

[6] L. Mittermeier, F. Katenbrink, A. Seitz, H. Muller, and B. Bruegge. “Dynamic schedul- ing for seamless computing”. In: https : / / ieeexplore . ieee . org / stamp / stamp . jsp ? tp = &arnumber = 8567371. Proceedings-8th IEEE International Sympo- sium on Cloud and Services Computing, SC2 2018, 2018.