

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

Abdul Ahad Ayaz



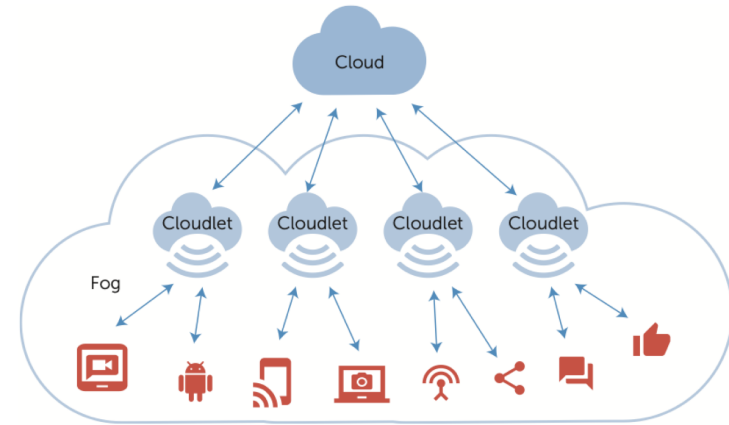
Computer Networks Group
Universität Paderborn

Outline

- What is IoT and Fog Computing?
- 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.

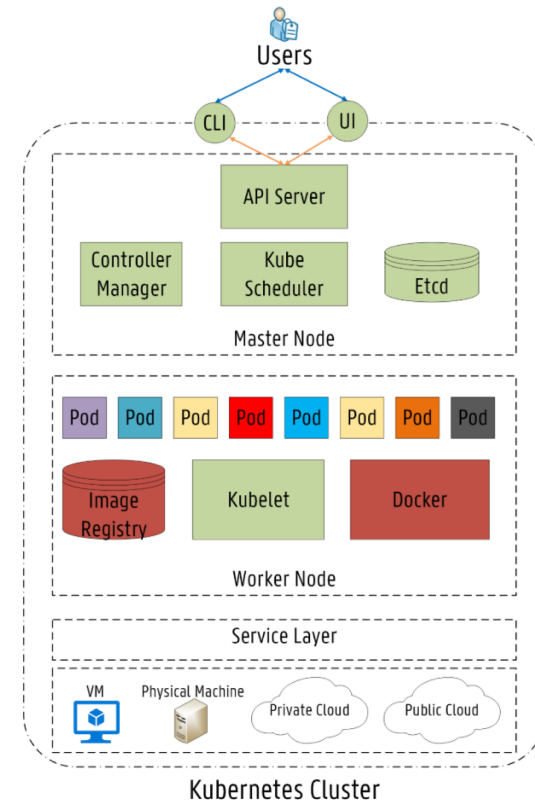
Fog Computing

- Concept of Fog Computing
- IoT based Applications
- IoT Application Resource:
 - VM
 - Containers
- Fog Computing vs Cloud Computing
 - Cloud: OpenStack, AWS
 - Containers: Kubernetes, Docker Swarn



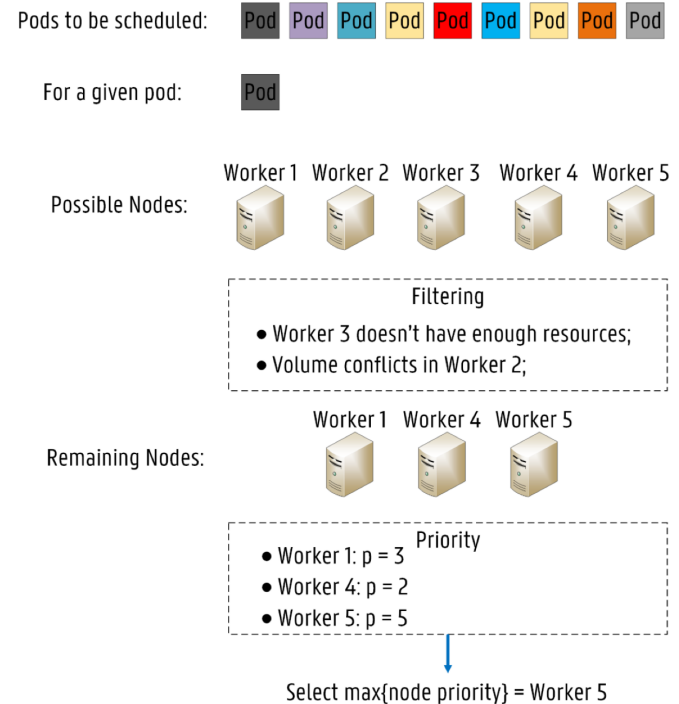
Kubernetes

- Architecture
 - Master Node
 - Worker Node
- Main Components
 - *Controller Manager*
 - *Kube Scheduler*
 - *Kubelets*
 - *Image Registry*
- Orchestration
 - Starting/stopping of applications
 - Scalability of applications
 - Load management
 - Health monitoring



Kubernetes Resource Provisioning

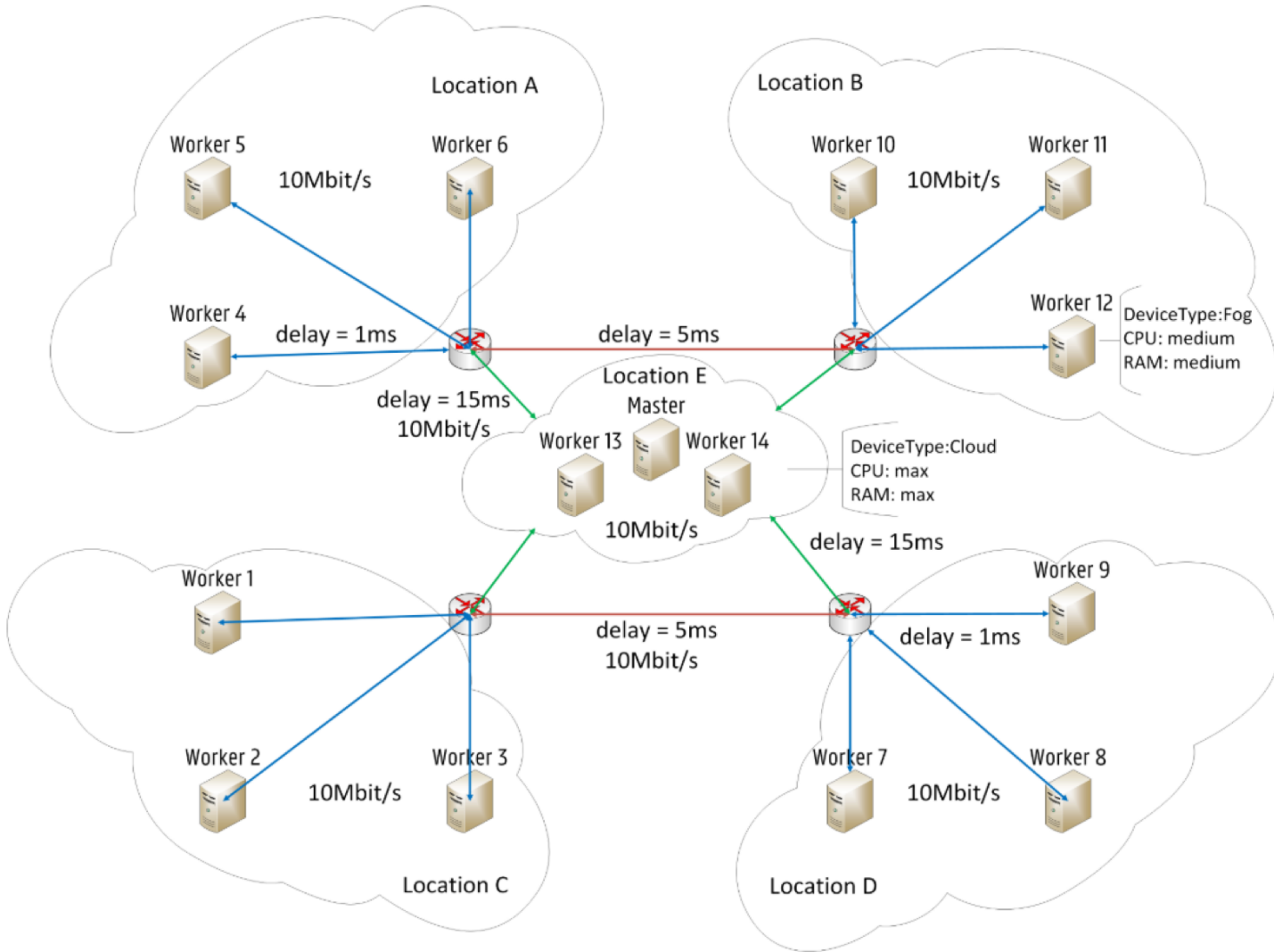
- Default Scheduler: “*Kube Scheduler*”
- Used for *Pod deployment across Worker Nodes*
- *Node selection criteria*:
 - Node Filtering
 - *PodFitsHostPorts*
 - *PodFitsResources*
 - Etc.
 - Node Priority/Scoring
 - *LeastRequestPriority*
 - *ImageLocalityPriority*
 - Etc.
- Drawbacks
 - No network resources consideration



Network-Based Scheduler

- Extending *Kube Scheduler*
 - Adding new filters or priorities
 - Building new scheduler from scratch
 - Calling external scheduling process by *Kube Scheduler*
- Uses “Affinity/Anti-Affinity rule”
- Uses Node labeling for resources
 - Resource: CPU, Memory -> {*Min, High, Medium*}
 - Device Type: {*Fog, Cloud*}
 - Network delay: RTT tags

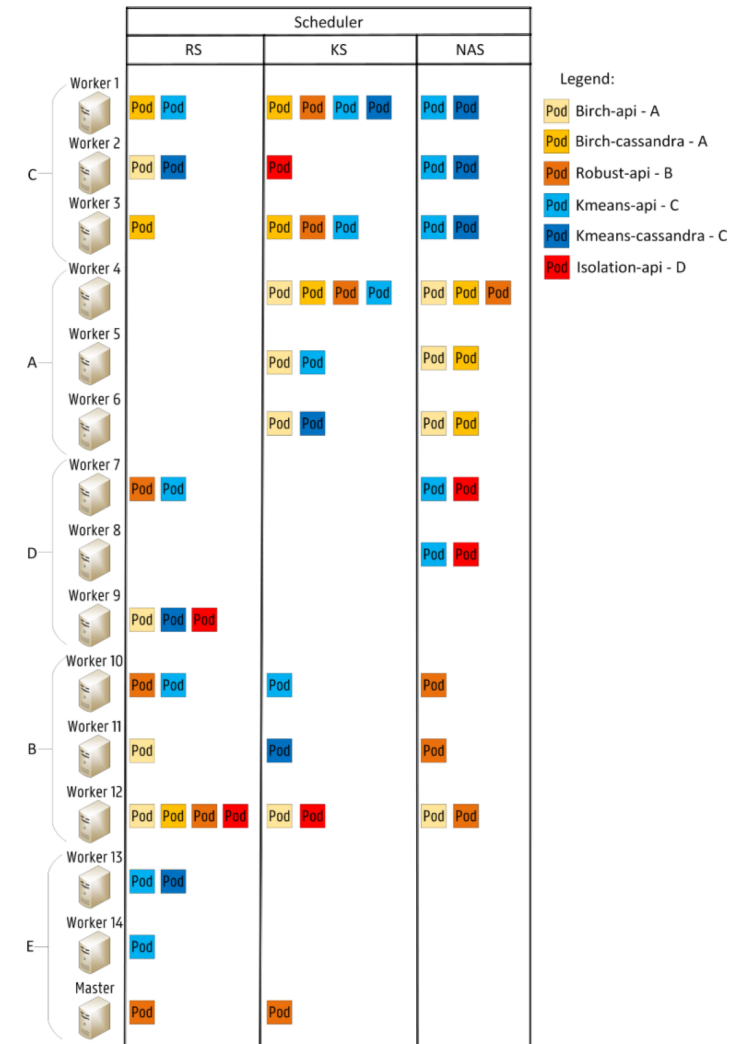
Fog Computing Infrastructure



Evaluation of Network-based Scheduler

- Smart City Scenario
- Comparison of Three Schedulers
 - *Kube Scheduler*
 - Network-based Scheduler
 - Random Scheduler

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
 - Docker Swarm
- Based on Scheduling Techniques
 - Technique One [paper ref]
 - Techniques Two [paper ref]

Conclusion

- Conclusion
 - Network-based Scheduler adds extra execution time
 - Optimized technique considering default scheduler
 - Works along side the default scheduler

References

- Added after advisor reviews