Network-Accelerated Cluster Scheduler

Radostin Stoyanov
radostin.stoyanov@eng.ox.ac.uk

Wesley Armour
wes.armour@oerc.ox.ac.uk

Noa Zilberman
noa.zilberman@eng.ox.ac.uk

Oxford e-Research Centre

UNIVERSITY OF OXFORD

Introduction

Efficient use of computing clusters is crucial in large-scale data centers: even small gains in utilization can save millions of dollars. However, as the number of microsecond-scale tasks increases, using a CPU to schedule tasks becomes inefficient. Today, applications typically depend on low-latency tasks that require service times of a microsecond or less [1]. These tasks are inherently latency-sensitive and non-optimal scheduling placement can significantly impact performance. Therefore, efficient scheduling in large-scale clusters is becoming increasingly more important. At the same time, efficient resource utilization can significantly reduce expenditure and operational costs. A standard method used today to address this problem is to deploy a centralized scheduler that can monitor all cluster nodes and be able to make high-quality placement decisions [2]. However, in practice, scheduling a large number of tasks as short as 1μs can overwhelm centralized schedulers.

How can we reduce the scheduling overhead of short tasks and increase scheduling throughput of Kubernetes clusters?

P4-K8s-Scheduler is a network-accelerated scheduler for Kubernetes that runs as a P4 program on a programmable switch (shown in Figure. 1), or on Infrastructure/Data Processing Unit (IPU/DPU) attached to the Kubernetes Control Plane.

Design

P4-K8s-Scheduler is designed to address two main problems for achieving high-performance:

- Reduce the scheduling overhead for short tasks with service times of a microsecond or less.
- Increase scheduling throughput of large-scale Kubernetes clusters.

Figure 2 illustrates a comparison of the Pod scheduling workflow with standard Kubernetes scheduler (kube-scheduler) and P4-K8s-Scheduler.

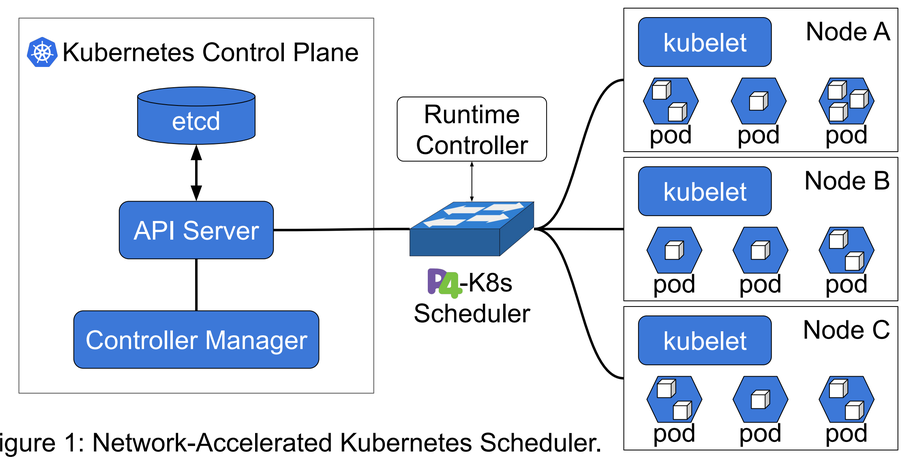

Figure 1: Network-Accelerated Kubernetes Scheduler.

Network Protocol. The Kubernetes API Server has been extended with support for a communication protocol that allows to i) register nodes and ii) send scheduling requests to the P4-K8s-Scheduler. These packets use UDP headers to enable processing with standard networking hardware.

Node Registration. Cluster nodes are registered with using registration packets with resource information. The scheduler then updates the stateful registers and sends the packet to the runtime controller to update the match-action tables used for filtering and scoring.

Pod Scheduling. Similar to node registration, the API Server sends scheduling request packets where each packet contains requirements for number of CPUs, amount of memory, storage, required/preferred affinity and scoring strategy.

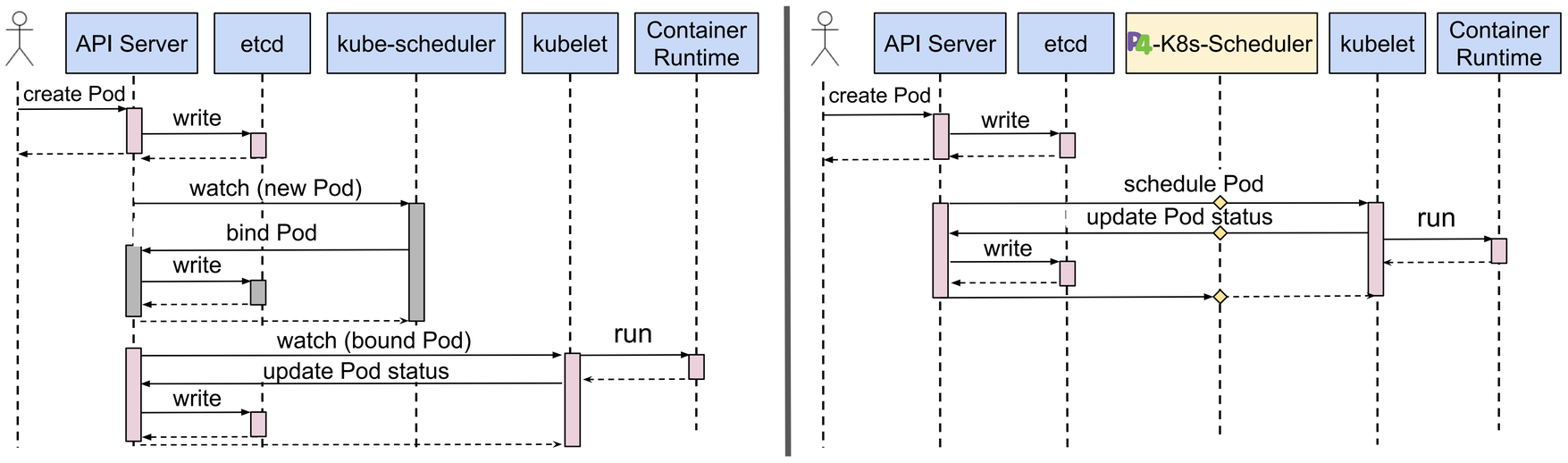

Figure 2: Pod scheduling workflow comparison between kube-scheduler and P4-K8s-Scheduler.

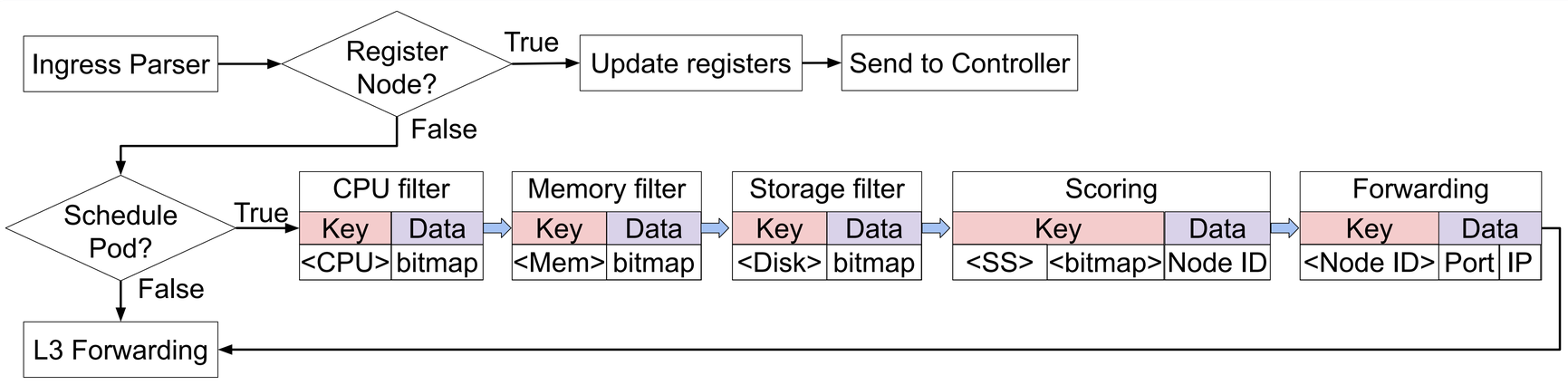

Figure 3: Node filtering and scoring mechanism. Node filters use match-action tables with bitmaps corresponding to node IDs, where a set bit indicates node satisfying the Pod requirements. A logical conjunction of filter bitmaps is used to identify and assign the node with highest score to the Pod request.

Results

P4-K8s-Scheduler achieves median task placement latency of ~50μs with 1,000-machine scale, and ~170ms total delay for 1,000 scheduling requests. The scheduling overhead has been reduced by an order of magnitude compared to state-of-the-art Kubernetes schedulers [2], and by up to 50% compared to other network-accelerated schedulers [3].

Analysis

The preliminary results demonstrate that P4-K8s-Scheduler can reduce task scheduling overheads by an order of magnitude compared to state-of-the-art Kubernetes schedulers.

Conclusion

P4-K8s-Scheduler is a network-accelerated cluster scheduler for Kubernetes. It runs on a high-performance programmable network device and improves the efficiency and scalability of Kubernetes by performing scheduling decisions at line-rate on packets exchanged between nodes in the cluster.

References

[1] Stephen Ibanez, Alex Mallery, Serhat Arslan, Theo Jepsen, Muhammad Shahbaz, Changhoon Kim, and Nick McKeown. 2021. The nanoPU: A Nanosecond Network Stack for Datacenters. In 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21).

[2] Ionel Gog, Malte Schwarzkopf, Adam Gleave, Robert N. M. Watson, and Steven Hand. 2016. Firmament: Fast, Centralized Cluster Scheduling at Scale. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16).

[3] Kettaneh, Ibrahim, et al. "Falcon: Low Latency, Network-Accelerated Scheduling." Proceedings of the 3rd P4 Workshop in Europe. 2020.

UK Research and Innovation

SIGCOMM '22 Poster and Demo Sessions, August 22 – 26, 2022, Amsterdam, Netherlands