

Cloud Computing (24/25)

Q/A Session 4

Ilja Behnke (i.behnke@tu-berlin.de)

Philipp Wiesner (wiesner@tu-berlin.de)

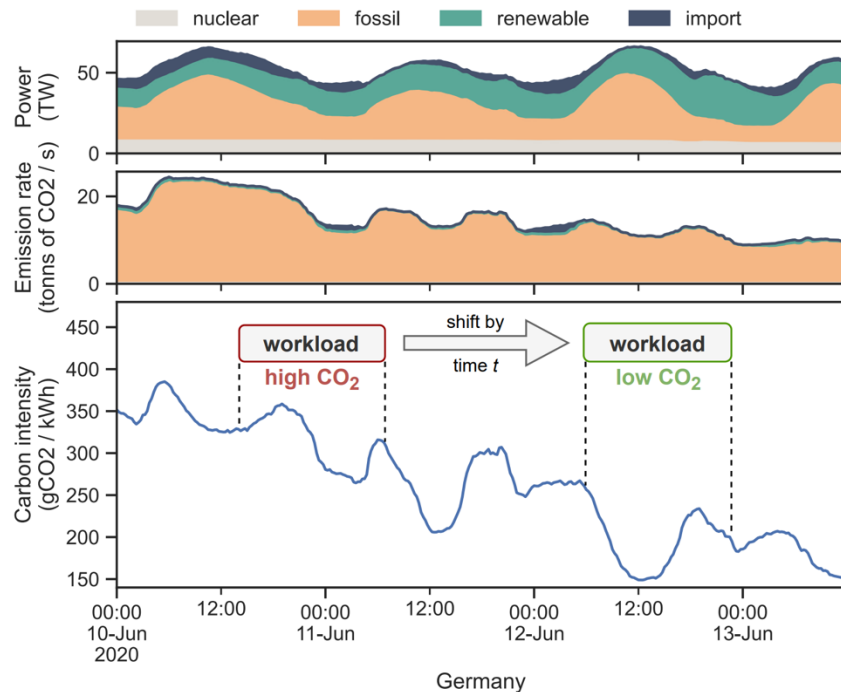
Assignment 4

- Due: 03.02.2025
- Adapt the placement of Kubernetes pods using a custom scheduling strategy
- Investigate a carbon-aware computing use case
- Tasks:
 1. Prepare an exemplary workload
 2. Implement a carbon-aware scheduler service
 3. Run scheduler in cluster and conduct experiments

Carbon-Aware Computing

Load shifting & workload migration to reduce the **operational** carbon emissions of data centers

Carbon intensity describes the amount of associated carbon emissions from consuming energy



Task 1: Prepare a Workload

- Should generate system load
- Execution time must be configurable
- Implement something yourself or use existing solutions (e.g. sysbench)
- We will later attempt to schedule these workloads

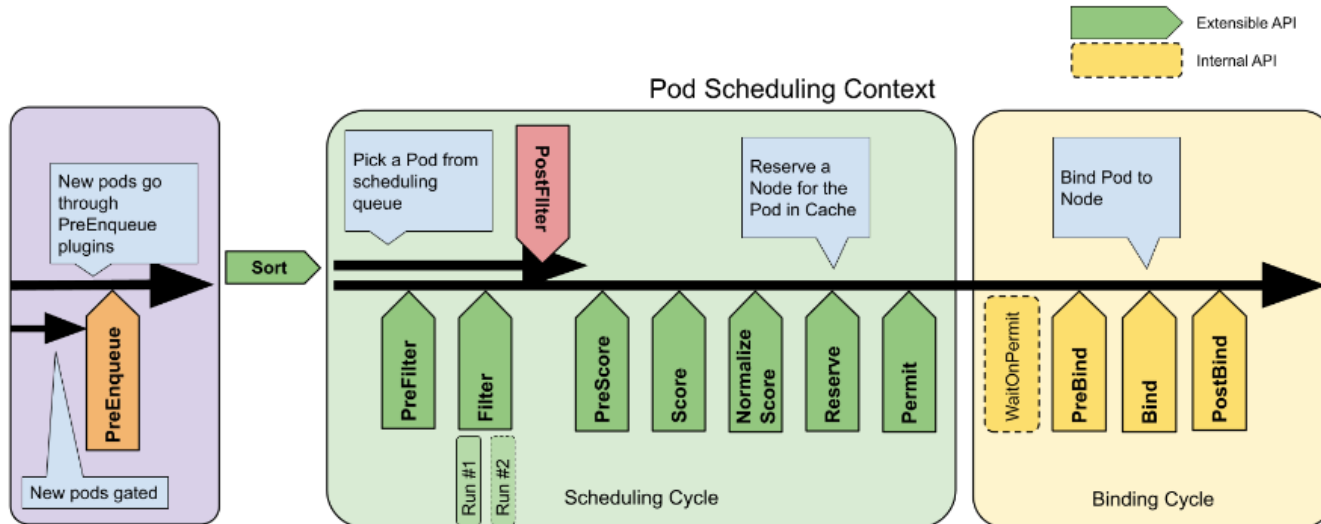
Output: workload.yaml with Kubernetes pod specification

```
pods/simple-pod.yaml   
  
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx  
spec:  
  containers:  
  - name: nginx  
    image: nginx:1.14.2  
    ports:  
    - containerPort: 80
```

Task 2: Carbon-Aware Pod Placement

Kubernetes: scheduling framework is a pluggable architecture

- defines a few extension points. Scheduler plugins register to be invoked at one or more extension points
- Out of scope for assignment -> we go for an easier solution



Task 2: Carbon-Aware Pod Placement

```
{  
  "DE": 344.77,  
  "ERCOT": 363.53,  
  "NL": 285.45  
}
```

Implement a custom scheduler that

- periodically deploys workloads
 - each workload execution has a different execution time
- fetches carbon intensity forecasts and uses them for custom *node affinities*
- uses the *kopf* framework to subscribe to Kubernetes events and logs relevant information to a file

Output: scheduler.py

```
import kopf  
  
@kopf.on.create('kopfexamples')  
def my_handler(spec, **_):  
    pass
```

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: with-affinity-anti-affinity  
spec:  
  affinity:  
    nodeAffinity:  
      requiredDuringSchedulingIgnoredDuringExecution:  
        nodeSelectorTerms:  
          - matchExpressions:  
            - key: kubernetes.io/os  
              operator: In  
              values:  
                - linux  
      preferredDuringSchedulingIgnoredDuringExecution:  
        - weight: 1  
          preference:  
            matchExpressions:  
              - key: label-1  
                operator: In  
                values:  
                  - key-1  
        - weight: 50  
          preference:  
            matchExpressions:  
              - key: label-2  
                operator: In  
                values:  
                  - key-2
```

Task 3: Experiments & Discussion

Package scheduler service and deploy it to Kubernetes cluster (check [this guide](#))

Experiments: Sequentially schedule 300 workloads to the cluster, for both:

- The carbon-aware pod placement strategy
- The default pod placement strategy (omit the node affinities)

Collect the results (log files) and answer the questions