Carbon aware spatial shifting using Karmada

carbon-aware-karmada-operator

Ross Fairbanks @rossf7

About Me

- Developer based in Spain (@rossf7 most places)
- Interested in green software and especially carbon awareness
- Advisor on Infrastructure at the Green Web Foundation which is a member of the Green Software Foundation
- Attended a few CNCF TAG Environmental Sustainability meetings and trying to be active there

Why Green Software?

- IEA - Data centers and data transmission both account for 1-1.5% of global electricity usage
- About the same emissions impact as aviation
- Energy usage is growing rapidly due to new workloads like generative AI

https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

Green Software Principles

Energy Efficiency
Consume the least amount of electricity possible

Hardware Efficiency
Use the least amount of embodied carbon possible

Carbon Awareness
Do more when the electricity is clean and less when it's dirty

- Energy efficiency
  - Lower PUE of servers in data center
  - Use more energy efficient language like Rust
- Hardware efficiency
  - Increase server utilization
  - Increase server lifespan
 - Provide software updates to devices for longer

https://learn.greensoftware.foundation/introduction

Carbon Intensity

- Carbon intensity measures how much carbon (CO_2e) is emitted per kilowatt-hour (KWh) of electricity consumed
- gCO_2eq/kWh is the standard unit of carbon intensity
- Carbon intensity is lower when more renewable energy is available
- APIs such as WattTime and Electricity Maps provide carbon intensity of electricity grids for multiple countries

Carbon Awareness - temporal shifting

Carbon intensity forecast - CAISO_NORTH
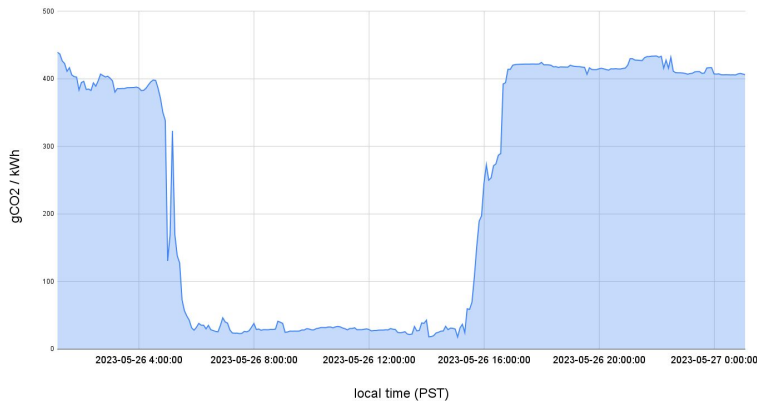
- Move non-urgent workloads to times when carbon intensity is lower
- Microsoft have developed an operator that adds carbon awareness to KEDA
  - https://github.com/Azure/carbon-aware-keda-operator
- Uses Carbon Aware SDK from GSF and WattTime carbon intensity forecast

Carbon Awareness - spatial shifting

- Move workloads to physical locations with lower carbon intensity

- I think Karmada is a great tool for this!
- I've developed a prototype operator to demonstrate the approach
  - https://github.com/rossf7/carbon-aware-karmada-operator
- Simple scheduling algorithm
  - Ranks clusters by carbon intensity
  - Selects desired number of clusters and sets cluster affinity
- Targets ClusterPropagationPolicy and PropagationPolicy CRDs
- Uses grid-intensity-go library and Electricity Maps API

CarbonAwareKarmadaPolicy CRD

```yaml
apiVersion: carbonaware.rossf7.github.io/v1alpha1
kind: CarbonAwareKarmadaPolicy
metadata:
  name: nginx-policy
spec:
  clusterLocations:
  - name: prd-de-01
    location: DE
  - name: prd-fr-01
    location: FR
  - name: prd-sv-01
    location: US-CAL-CISO
  desiredClusters: 2
  karmadaTarget: propagationpolicies.policy.karmada.io
  karmadaTargetRef:
    name: nginx-propagation
    namespace: default
```

Demo

Improvements

- Support more Karmada CRDs
- Helm chart for easier deployment
- Include carbon intensity forecast in scheduling algorithm
- Include distance between clusters in scheduling algorithm

Links

- Code: https://github.com/rossf7/carbon-aware-karmada-operator
- Blog: https://rossfairbanks.com/2023/07/12/carbon-aware-spatial-shifting-with-karmada/
- Carbon intensity library: https://github.com/thegreenwebfoundation/grid-intensity-go
- Green Software Principles: https://learn.greensoftware.foundation/introduction
- TAG Environmental Sustainability https://tag-env-sustainability.cncf.io/
- Landscape https://tag-env-sustainability.cncf.io/landscape/

Thank you!

- Questions?