Sustainable computing

Pedro Ibáñez Requena
Ecosystem Field Engineering Telco 5g Team

Index

- What is sustainable computing?
- How it works?
- Requirements
- Setup
- Demo
- Next steps
- References

What is Sustainable computing?

Sustainable computing

Sustainable computing concerns the **consumption of computing resources in a way that means it has a net zero impact on the environment**, a broad concept that includes energy, ecosystems, pollution and natural resources. David Mytton https://davidmytton.blog/sustainable-computing/

Computational sustainability is an **emerging field that attempts to balance societal, economic, and environmental resources for the future well-being of humanity** using methods from mathematics, computer science, and information science fields. Sustainability in this context refers to the world's ability to sustain biological, social, and environmental systems in the long term. https://en.wikipedia.org/wiki/Computational_sustainability

Green computing, green IT, or ICT sustainability, is the **study and practice of environmentally sustainable computing or IT**. The goals of green computing are similar to green chemistry: **reduce the use of hazardous materials, maximize energy efficiency during the product's lifetime, the recyclability or biodegradability of defunct products and factory waste.** https://en.wikipedia.org/wiki/Green_computing

Sustainable computing

Green Information Technology:

- Power measurement and management
- Overhead and waste reduction
- Resource optimization

Sustainable computing

https://github.com/sustainable-computing-io/

Report - KEPLER (**K**ubernetes-based **E**fficiency **P**ower **L**evel **E**xporter): container level power reporting.

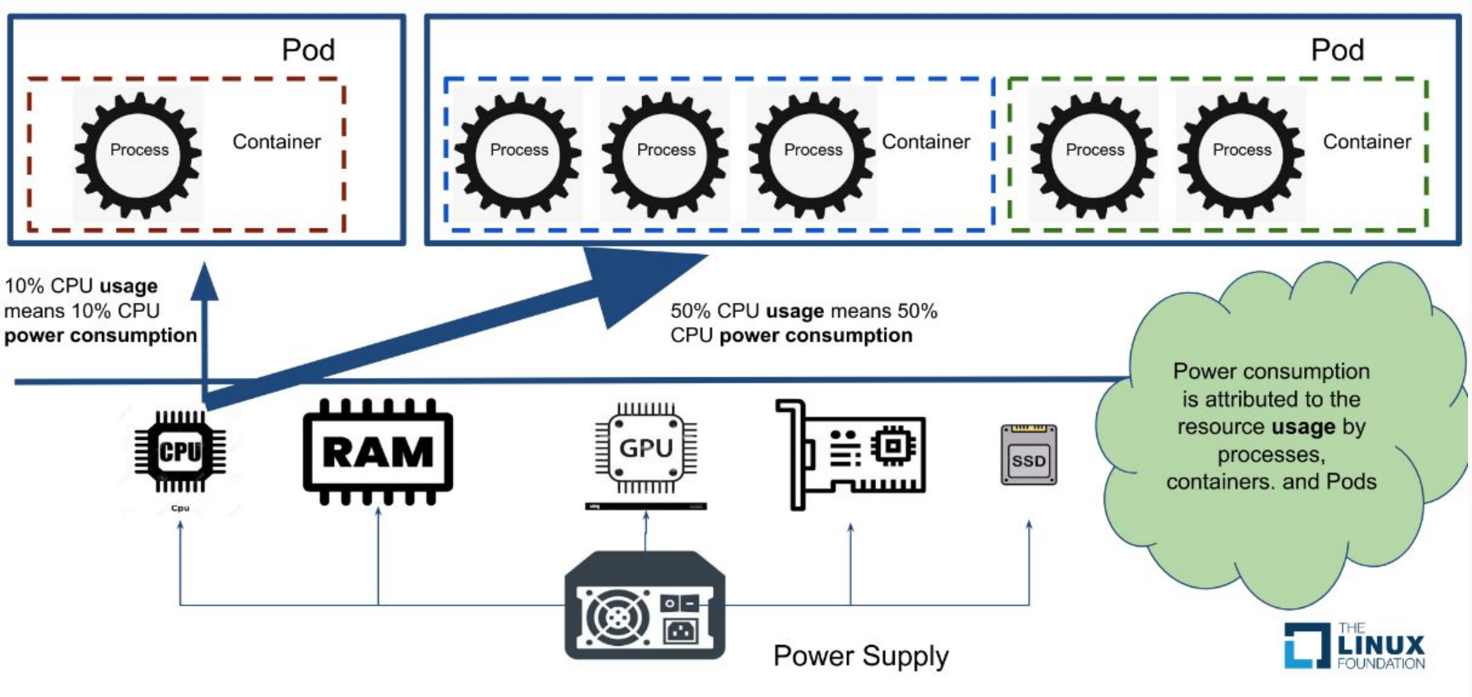
Instrument - PEAKS (**P**ower **E**fficiency **A**ware **K**ubernetes **S**cheduler): Energy efficiency aware workload scheduling.

Optimize - CLEVER (**C**ontainer **L**evel **E**nergy-efficiency **V**PA **R**ecommender): Online workload tuning for energy efficiency.

Applied for CNCF Sandbox project.

How it works?

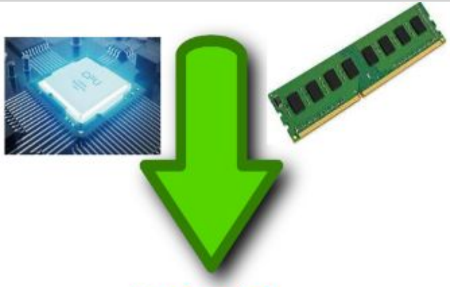
How it works?


Pod
Process
Container
Pod
Process
Process
Process
Container
Process
Process
Container
10% CPU **usage** means 10% CPU **power consumption**
50% CPU **usage** means 50% CPU **power consumption**
Power consumption is attributed to the resource **usage** by processes, containers. and Pods
CPU
Cpu
RAM
GPU
SSD
Power Supply
THE LINUX FOUNDATION

How it works?

Reporting

- Per Pod level energy consumption reporting, including **CPU/GPU, RAM**
- Support **bare metal** as well as **VM**
- Support **Prometheus**



Reduction

- Reduced computational resource used by the probe
- Using **eBPF**

Regression

- Support **ML** models to estimate energy consumption
- Science based approach

THE LINUX FOUNDATION

How it works?

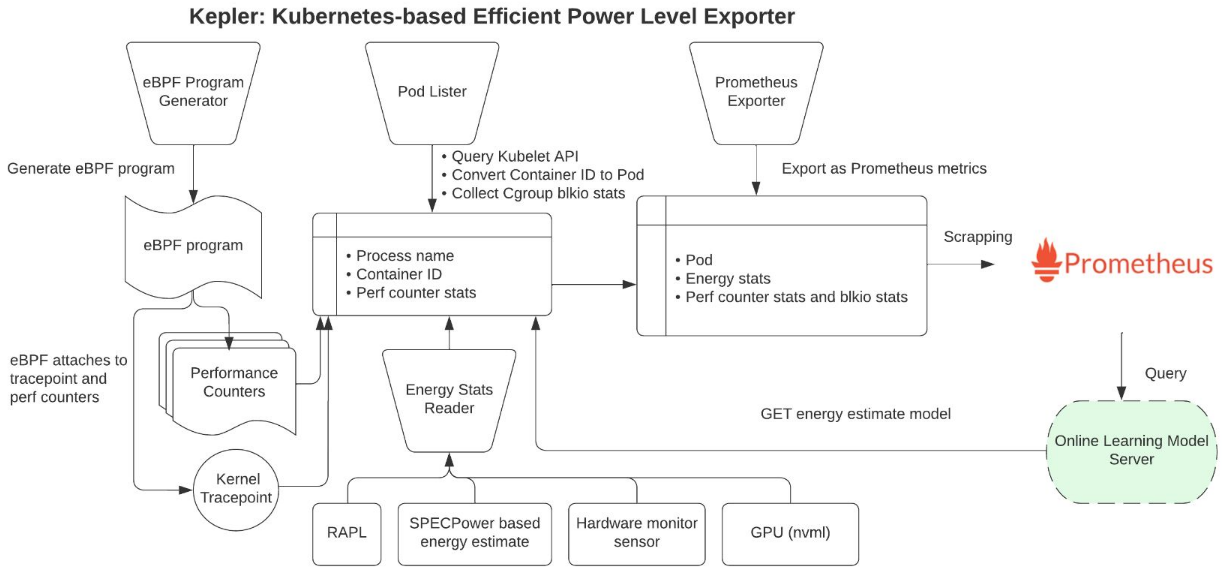
Report - KEPLER (**K**ubernetes-based **E**fficiency **P**ower **L**evel **E**xporter): Uses eBPF to probe energy related system stats and exports as Prometheus metrics.

- Container Native, Cloud Native
- Lightweight, Expansible
- Accurate and Fair

Instrument - PEAKS (**P**ower **E**fficiency **A**ware **K**ubernetes **S**cheduler): Uses metrics exported by KEPLER to schedule Pods to achieve optimal performance per Watt.

Optimize - CLEVER (**C**ontainer **L**evel **E**nergy-efficiency **V**PA **R**ecommender): Uses ML models to predict and tune Pod resources to optimize performance per Watt by tuning CPU/GPU frequency.

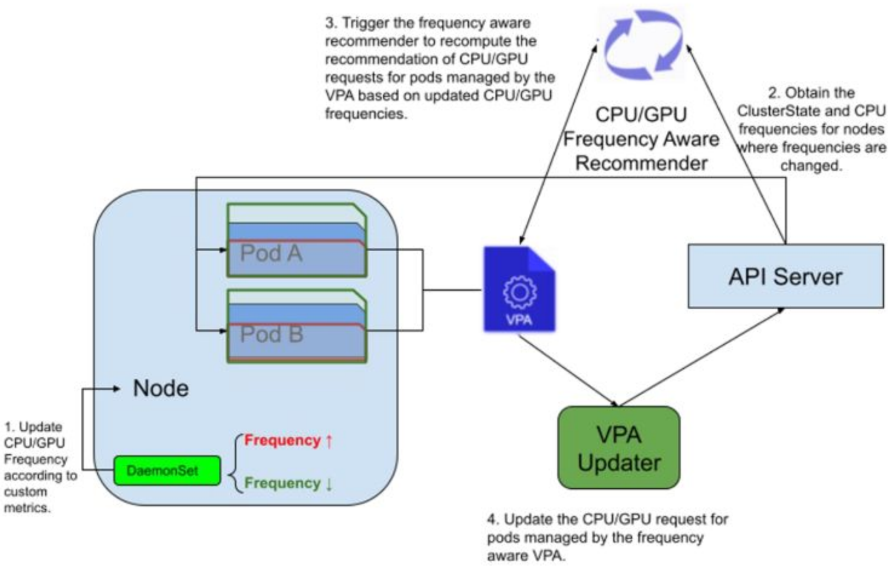
How it works?



Kepler: Kubernetes-based Efficient Power Level Exporter
eBPF Program Generator
Pod Lister
Prometheus Exporter
Generate eBPF program
• Query Kubelet API
• Convert Container ID to Pod
• Collect Cgroup blkio stats
Export as Prometheus metrics
eBPF program
• Process name
• Container ID
• Perf counter stats
• Pod
• Energy stats
• Perf counter stats and blkio stats
Scrapping
Prometheus
eBPF attaches to tracepoint and perf counters
Performance Counters
Energy Stats Reader
Query
GET energy estimate model
Online Learning Model Server
Kernel Tracepoint
RAPL
SPECPower based energy estimate
Hardware monitor sensor
GPU (nvml)

How it works?

CLEVER: Container Level Energy-Efficient VPA Recommender



Carbon footprint can be reduced via improving the objective of performance per watt for containers, namely finding efficient way to guarantee a certain QoS/performance for a workload.

- Lower Frequency: Reducing Energy Consumption.
- Increase CPU allocation: guarantee QoS

Requirements

Requirements

Requirements

1.1. Kernel 4.18+
1.2. Access to a Kubernetes cluster
1.3. kubectl v1.21.0+
1.4. cgroup v2 (systemd.unified_cgroup_hierarchy=1)
    1.4.1. Recommended Linux Kernel version is 5.8 or later
    1.4.2. Minimum version: 4.15
    1.4.3. Container runtime supports cgroup v2. For example:
        1.4.3.1. containerd v1.4 and later
        1.4.3.2. cri-o v1.20 and later
    1.4.4. The kubelet and the container runtime are configured to use the systemd cgroup driver

Setup

Setup

- Kepler

git clone https://github.com/sustainable-computing-io/kepler-helm-chart && cd kepler-helm-chart

helm install kepler . --values values.yaml --create-namespace --namespace <namespace>

- Prometheus

git clone https://github.com/prometheus-operator/kube-prometheus

cd kube-prometheus

kubectl apply --server-side -f manifests/setup

kubectl wait --for condition=Established --all CustomResourceDefinition --namespace=monitoring

kubectl apply -f manifests/

Setup

- Import the kepler grafana dashboard

https://github.com/sustainable-computing-io/kepler/blob/main/grafana-dashboards/Kepler-Exporter.json

Setup

- VPA

```
# git clone https://github.com/kubernetes/autoscaler.git
# cd autoscaler/vertical-pod-autoscaler/
# ./hack/vpa-up.sh
```

Setup

- Clever

git clone https://github.com/sustainable-computing-io/clever.git

kubectl apply -f manifests/clever.yaml

- Import the Clever grafana dashboard

https://raw.githubusercontent.com/sustainable-computing-io/clever/main/dashboards/clever-dashboard.json

Demo

Demo

- https://www.youtube.com/watch?v=qzeqk262n80

Next steps

Next steps

- Energy consumption and carbon emissions telemetry and dashboard
- Kepler and Model Server provide tuning and scheduling heuristics.
  - Dynamic power reduction
    - Energy Aware Pod Scheduling
    - DVFS based Vertical Pod Scaling
    - Energy Efficient Node Tuning
  - Leaky power reduction
    - Thermal Temperature Aware Scheduling and Scaling

References

References

Official web: https://sustainable-computing.io/

Source code: https://github.com/sustainable-computing-io/

Installing Kepler: https://sustainable-computing.io/installation/kepler/

Installing prometheus: https://github.com/prometheus-operator/kube-prometheus

Installing VPA:
https://github.com/kubernetes/autoscaler/blob/master/vertical-pod-autoscaler/README.md

Installing Clever: https://github.com/sustainable-computing-io/clever

cgroups v2: https://kubernetes.io/docs/concepts/architecture/cgroups/

References

Red Hat NEXT! 2022: Cloud Native Sustainability the Open Source Way: https://www.youtube.com/watch?v=bt54EidpE5w

"Sustainability the Container Native Way", Huamin Chen (Red Hat) & Chen Wang (IBM), Open Source Summit NA 2022 https://github.com/sustainable-computing-io/kepler/blob/main/doc/OSS-NA22.pdf

KubeConNA-2022 Sustainability Research the Cloud Native Way https://github.com/sustainable-computing-io/kepler-doc/blob/main/demos/KubeConNA-2022_Sustainability-Research-the-Cloud-Native-Way.pdf

Thanks!