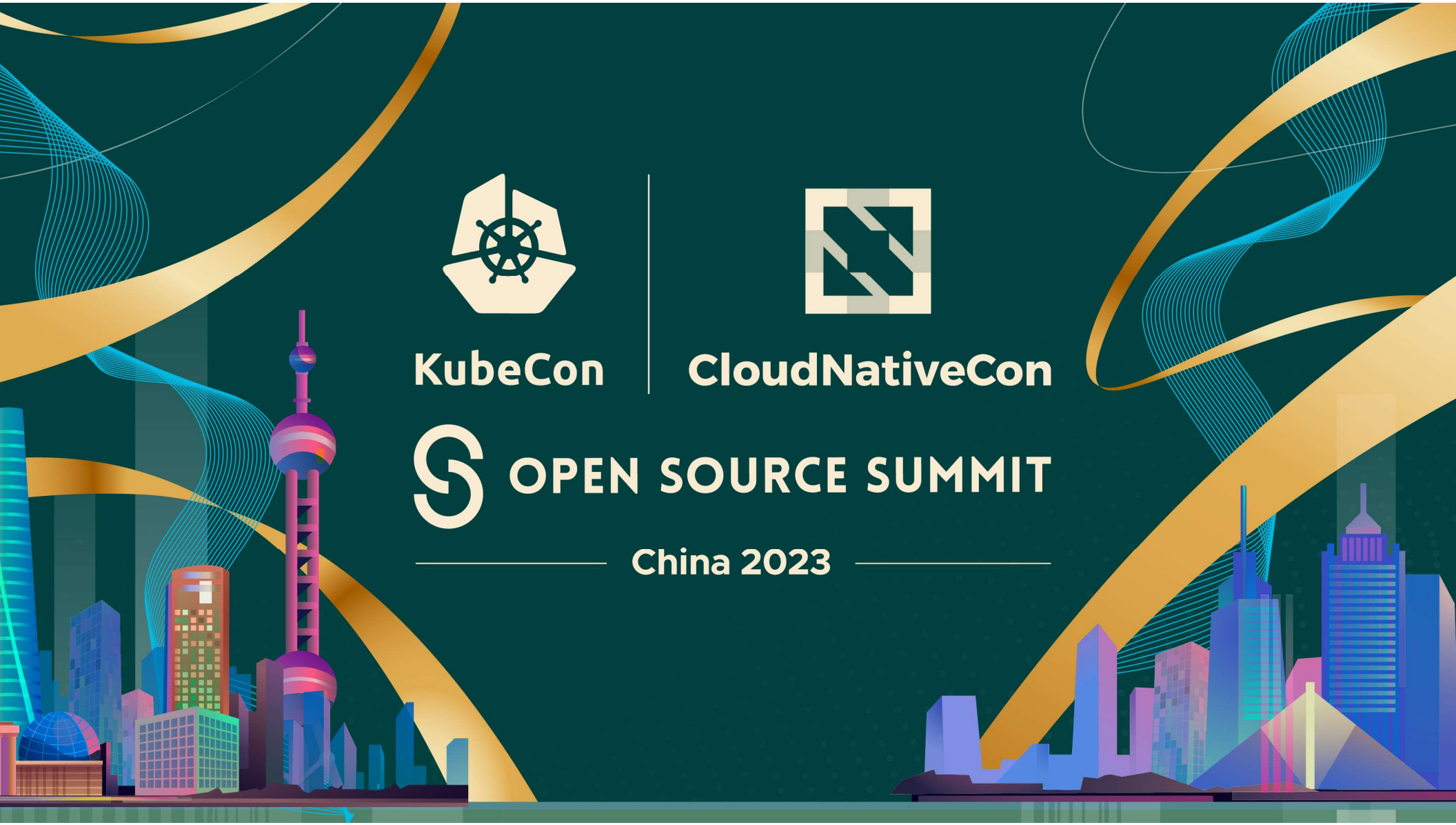
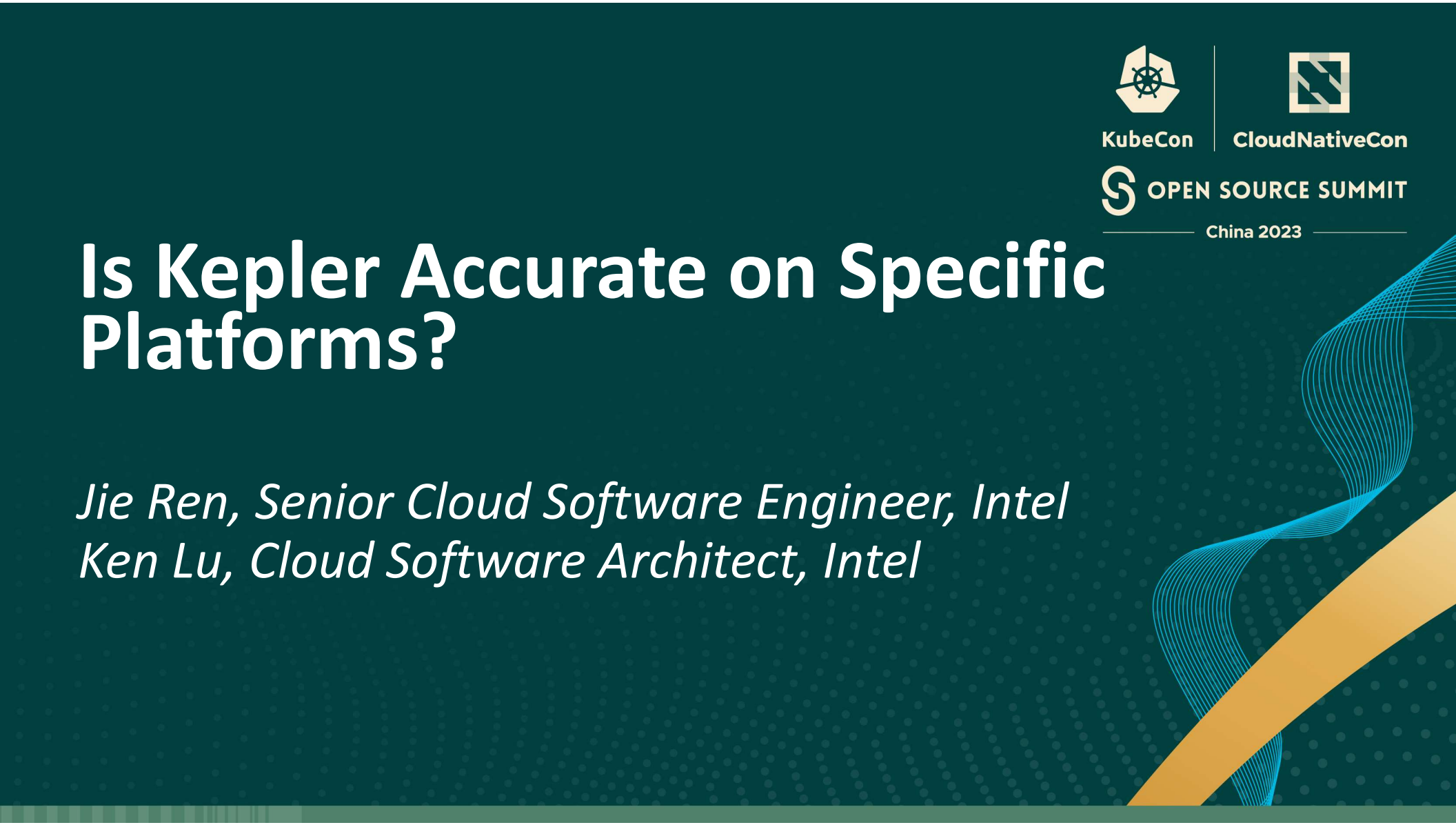

KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023



Is Kepler Accurate on Specific Platforms?

Jie Ren, Senior Cloud Software Engineer, Intel
Ken Lu, Cloud Software Architect, Intel

Agenda

- ❑ Kepler Fundamental
- ❑ Kepler Power Model
- ❑ Platform Validation for Kepler
- ❑ Furthermore

Kepler Fundamental

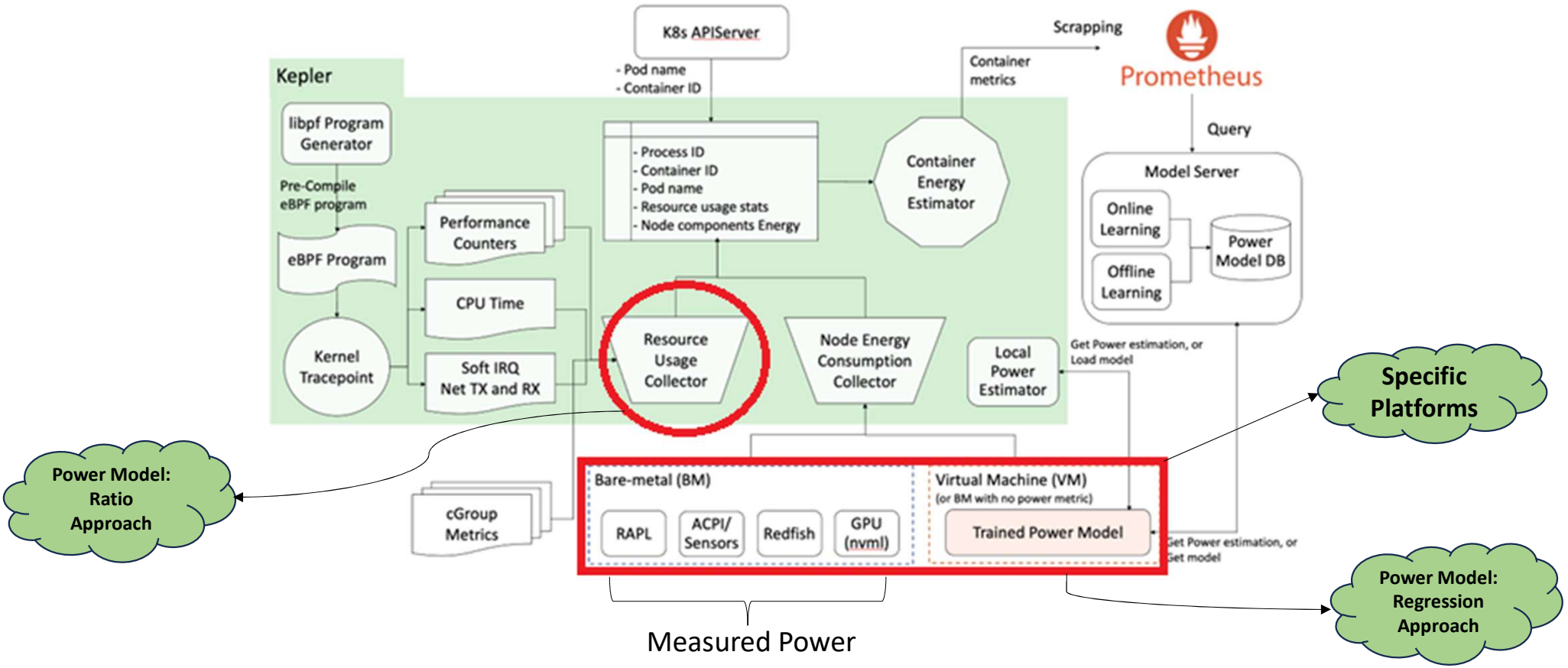
KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

❑ Kepler (Kubernetes Efficient Power Level Exporter)

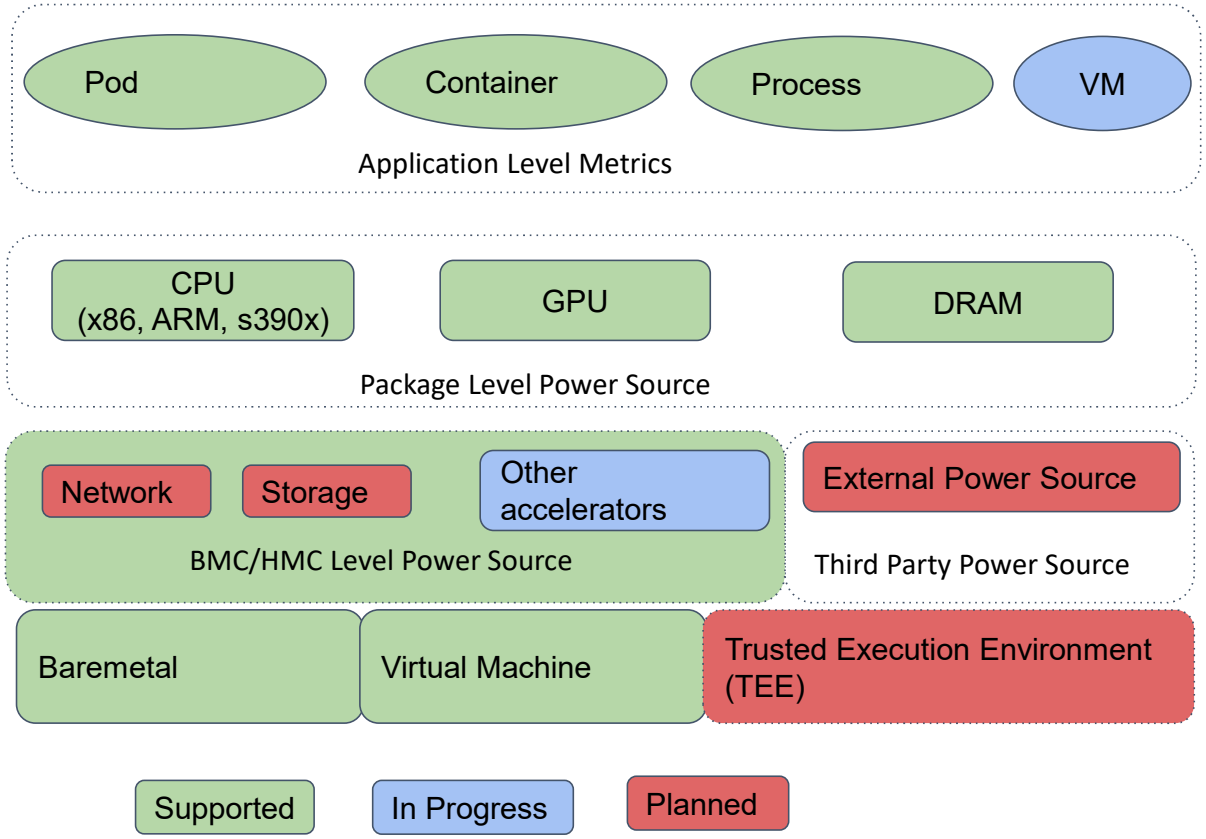
- CNCF Sandbox onboarding in 2023.6
- eBPF for energy-related data probe
- Power related metrics collector and exporter
- Power Modeling for cloud native applications
- Power = Energy / time

More details could be found in Kiosk session of Kelper in KubeCon China 2023

Kepler Architecture



Kepler Support Matrix



Support Pod/Container/Process level metrics

Support CPU (x86, arm64, s390x)/GPU/DRAM

Support BMC/HMC power sources

Kepler supports power measurement on bare metal and VMs. TEE support will be investigated.

Power Model in Kepler

❑ Kepler Energy Source

- CPU component energy (Inband)
  - x86
    - RAPL(Running Average Power Limit) defines CPU Power Domains such as Package/Core/Uncore/DRAM
    - RAPL can be read through Linux SysFS(/sys/class/powercap/) or MSR(Model Specific Register)
  - ARM
    - Ampere Xgene hwmon (/sys/class/hwmon/)
- Platform energy (Out of band)
  - ACPI (Advanced Configuration Power Interface)
  - HMC (Hardware Management Console): supported on s390x
  - BMC (Baseboard Management Controller): managed by Redfish/IPMI, Kepler implements Redfish currently
- GPU energy
  - NVML(Nvidia Management Library)

❑ Absolute Power = Dynamic Power + Idle Power

- Absolute Power: The APIs that expose the real-time power metrics export the absolute power
- Dynamic Power: directly related to the resource utilization
- Idle Power: the constant power that does not vary regardless if the system is at rest or with load

https://sustainable-computing.io/design/kepler-energy-sources

Power Model in Kepler

❑ Ratio approach

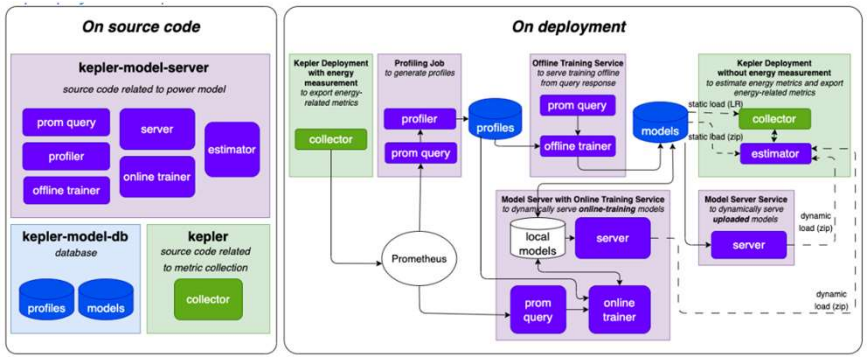
- Based on measured power
- Calculate the ratio of the container's resource usage
- Do the power attribution among containers based on ratio

$$Proc_{res}^{dyn} = \frac{process\_res\_utilization}{total\_res\_utilization} * res^{dyn}$$

Platform	Resource	default metric for resource usage calculation	metric origin
BM	CPU.PKG	cpu_instructions	performance counter
BM	CPU.Core	cpu_instructions	performance counter
BM	CPU.Uncore	N/A (evenly divided)	N/A
BM	CPU.DRAM	cache_miss	performance counter
BM	Network	bpf_net_tx_irq, bpf_net_rx_irq	eBPF
VM	CPU(all components)	bpf_cpu_time_us	eBPF

❑ Regression approach

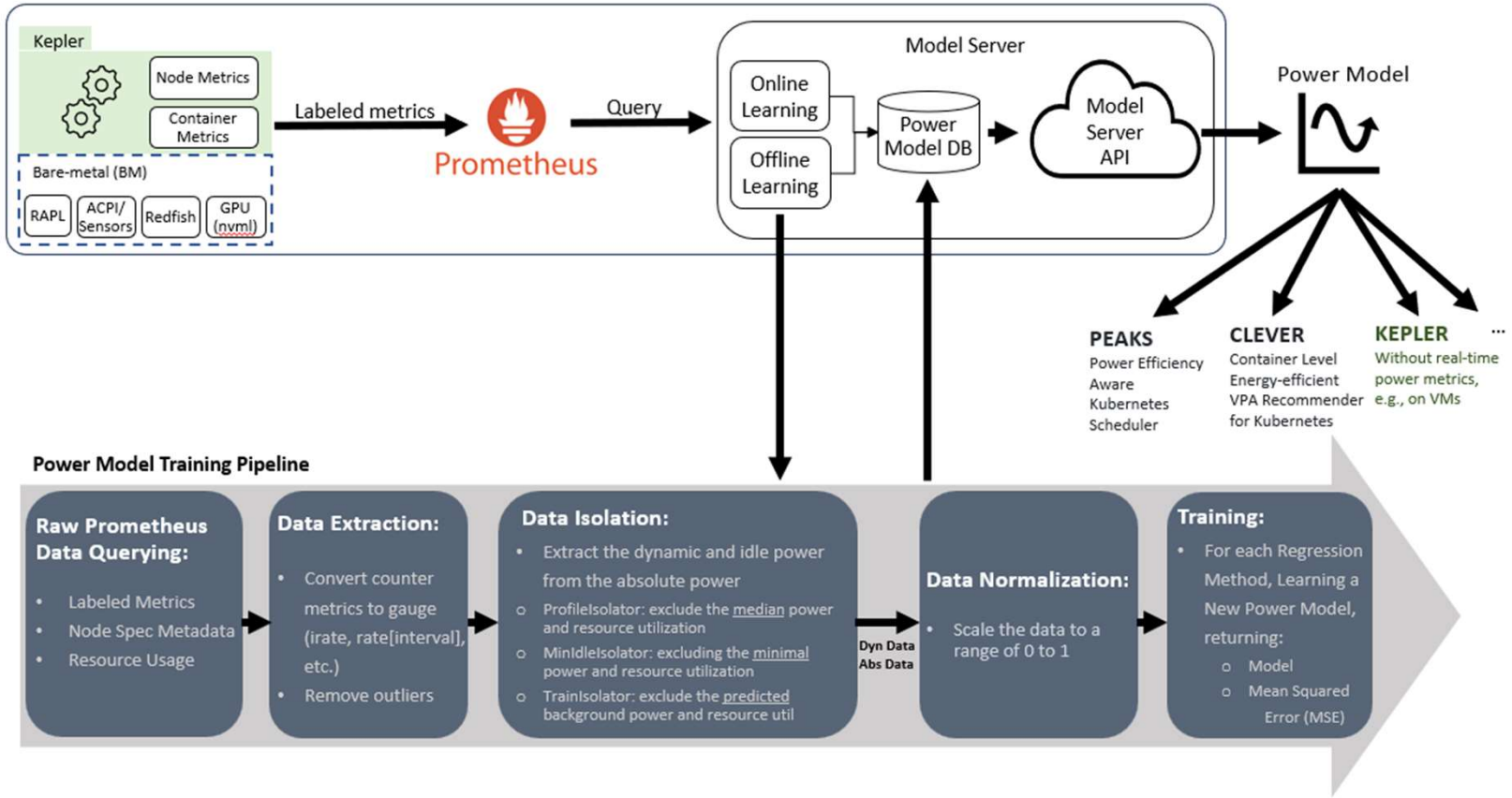
- Based on performance counters
- Use machine learning to train and validate models
- Use trained model to estimate(predict) power



https://sustainable-computing.io/design/power_model/

https://github.com/sustainable-computing-io/kepler-model-server

Power Model training in Kepler



references:

1. Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications (IEEE CLOUD 2023 paper, IBM Research Publication)
2. Exploring Kepler's Potentials: Unveiling Cloud Application Power Consumption (to-be-published on CNCF blog) by Marcelo Amaral, Sunyanan Choochotkaew, Huamin Chen and others

Platform Validation for Kepler

❑ Address concerns

- Is Kepler fully tested?
- Is Kepler well supported by specific platform?
- Is Kepler accurate on power attribution on specific platform?

❑ Framework design

- Automation workflow
- Follow test framework as Kepler
- Validator design
- Data validity check and data accuracy check
- Validation result evaluation

❑ More scenarios and validation perspectives

- Actual workload carbon footprint
- Power visualization on different perspectives
- Extend platform scenario from Bare Metal to VM
- Extend validation scenario from platform validation to model validation

https://github.com/sustainable-computing-io/kepler/blob/main/enhancements/platform-validation.md

Platform Validation Framework

❑Mechanism and methodology

- Automation workflow

✓ Manual triggered Github Action
✓ Runs on self-hosted runner
✓ Containerized test image, platform agnostic

- Follow test framework as Kepler: Ginkgo

✓ Domain Specific Language
✓ Generate customized formatted test report, such as JSON.

- Validator: independent tool for power calculation and comparison

✓ Independent RAPL-based energy collection and power consumption calculation tool for Intel X86 BareMetal platforms
✓ Work mechanism: sampling and calculation
✓ Comparison assumption: Collect power before and after target application deployment, use delta as comparison base.
✓ For other platforms, developers may use other specific measurement methods and tools to implement similar logic.

- Data validity check and data accuracy check.

✓ Validity: CPU Architecture check(cupid tool for Intel X86), component power source support status check, etc
✓ Accuracy: node level check, container level check.

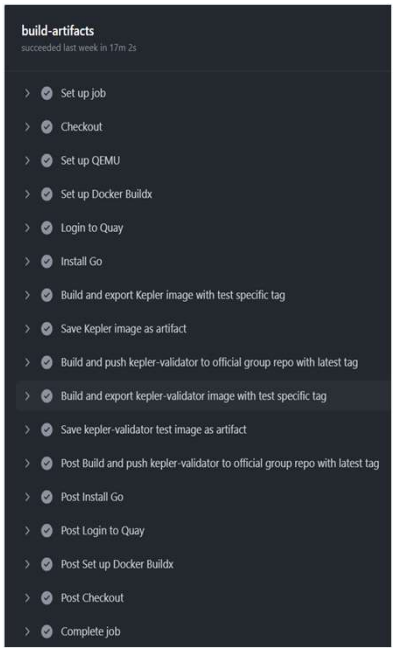
https://github.com/sustainable-computing-io/kepler-doc/blob/main/docs/platform-validation/index.md#mechanism-and-methodology

Platform Validation Framework

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

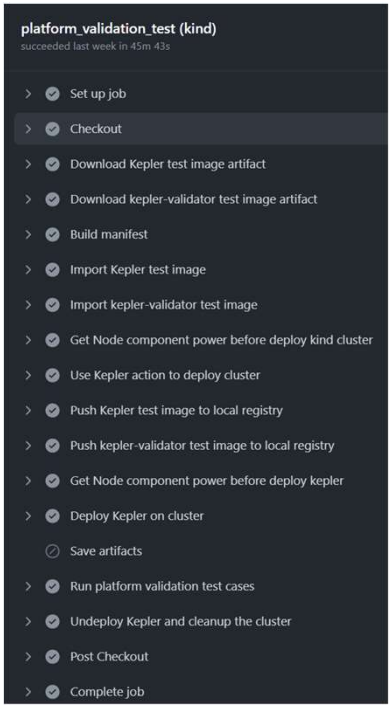
❑ Automation workflow

- Phase 1: Artifacts build.



https://github.com/jiere/kepler/actions/runs/6142358045/job/16663974439

- Phase 2: Validation test.



https://github.com/jiere/kepler/actions/runs/6142358045/job/16664233239

https://github.com/sustainable-computing-io/kepler/blob/main/.github/workflows/platform-validation.yml

Platform Validation Framework

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

❑ Automation test cases

▪ Case design

✓ Check point: Kepler exporter side, Prometheus side
✓ Check perspective: data validity, data accuracy

▪ Case limitation

✓ Comparison base is node level or CPU package level.
✓ Container level accuracy check assumption
✓ Test environment interference: Multi-tenants, container/VM coexistence, etc.

▪ Test result evaluation

✓ Data validity: ☑
✓ Data accuracy:
☑ Node level accuracy is good
🔔 Container level accuracy varies (manual check currently).

	Before Kepler Deployment		After Kepler Deployment										Kepler				All namespaces				system_processes			
Test round	Validator PKG	Validator DRAM	Validator PKG	Validator DRAM	Prometheus PKG	Prometheus DRAM	Node PKG power deviation	Node DRAM power deviation	Validator kepler PKG power	Validator kepler DRAM power	Kepler PKG power deviation	Kepler DRAM power deviation	PKG Dyn	PKG Idle	DRAM Dyn	DRAM Idle	PKG Dyn	PKG Idle	DRAM Dyn	DRAM Idle	PKG Dyn	PKG Idle	DRAM Dyn	DRAM Idle
1	373.002	18.259	373.087	18.263	373.35	18.322	0.07%	0.32%	0.085	0.004	-51.76%	25.00%	0.04	3.526	0.005	0.345	0.822	69.514	0.005	0.345	3.875	299.15	0.227	15.761
2	334.285	18.521	372.225	18.271	372.533	18.264	0.08%	-0.04%	37.94	-0.25	-99.87%	-103.20%	0.05	4.094	0.008	0.668	0.454	32.524	0.062	4.317	4.701	334.87	0.192	13.707
3	334.212	18.511	372.501	18.263	372.546	18.243	0.01%	-0.11%	38.289	-0.248	-99.11%	-120.97%	0.34	4.191	0.052	0.669	2.693	32.697	0.315	3.921	25.73	311.44	1.038	12.698

```
861 [ReportAfterSuite] PASSED [0.009 seconds]
862 [ReportAfterSuite] Autogenerated ReportAfterSuite for --json-report
863 autogenerated by Ginkgo
864 ------------------------------
865
866 Ran 13 of 17 Specs in 0.213 seconds
867 SUCCESS! -- 13 Passed | 0 Failed | 0 Pending | 4 Skipped
868 PASS
```

Container level accuracy – Actual Workload

- Power consumption change before and after Redis deployment

Validator Sampling and Calculation

```
jie@jie-nuc:~$ docker run -i --rm -v $(pwd):/output localhost:5001/platform-validation:x86-ra
gl /usr/bin/validator
I0922 01:32:48.993433       1 gpu.go:46] Failed to init nvml, err: could not init nvml: error
 opening libnvidia-ml.so.1: libnvidia-ml.so.1: cannot open shared object file: No such file o
r directory
Dump flag parameters value...
gen-env:false
gen-power:true
sampleCount:20
sampleDuration:15
I0922 01:32:48.995019       1 redfish.go:169] failed to get redfish credential file path
I0922 01:32:48.995565       1 acpi.go:67] Could not find any ACPI power meter path. Is it a V
M?
Sample 1:
pre: map[0:Pkg: 92372784 (Core: 108295814, Uncore: 10168978) DRAM: 0]
cur: map[0:Pkg: 92425837 (Core: 108315652, Uncore: 10169130) DRAM: 0]
Sample 2:
pre: map[0:Pkg: 92425837 (Core: 108315652, Uncore: 10169130) DRAM: 0]
cur: map[0:Pkg: 92476066 (Core: 108333635, Uncore: 10169292) DRAM: 0]
Sample 3:
pre: map[0:Pkg: 92476066 (Core: 108333635, Uncore: 10169292) DRAM: 0]
cur: map[0:Pkg: 92526472 (Core: 108351620, Uncore: 10169443) DRAM: 0]
Sample 4:
pre: map[0:Pkg: 92526472 (Core: 108351620, Uncore: 10169443) DRAM: 0]
cur: map[0:Pkg: 92577952 (Core: 108371129, Uncore: 10169607) DRAM: 0]
Sample 5:
```

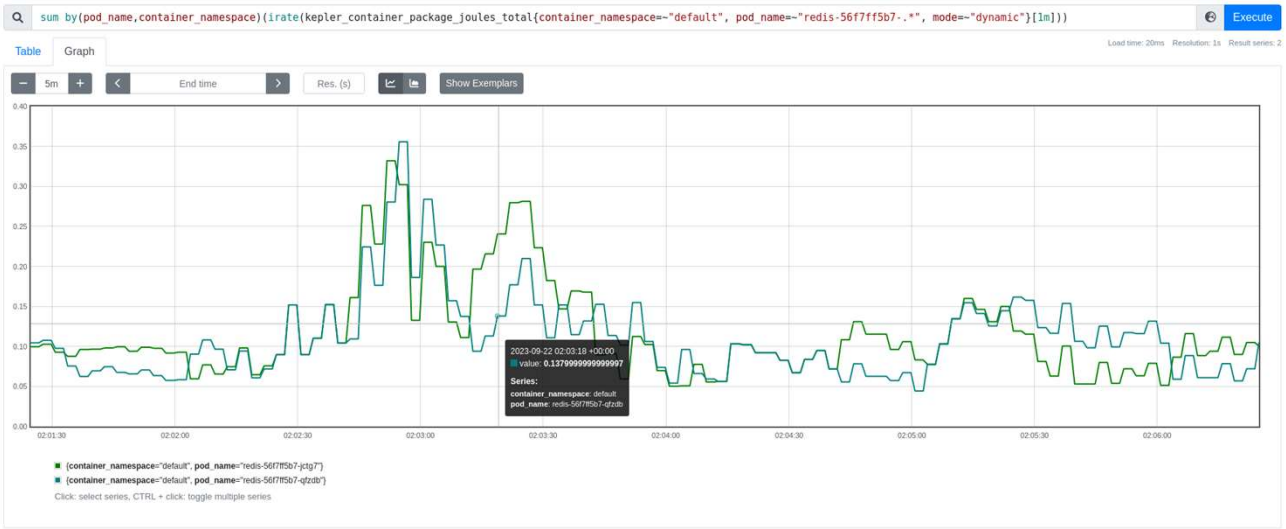
```
jie@jie-nuc:~$ cat power.csv
Pkg,Core,Uncore,Dram
3.315,1.168,0.010,0.000
3.384,1.234,0.011,0.000
```

- Validator calculated PKG power increase: 0.069W
- Prometheus Queried Dynamic Power: 0.071W

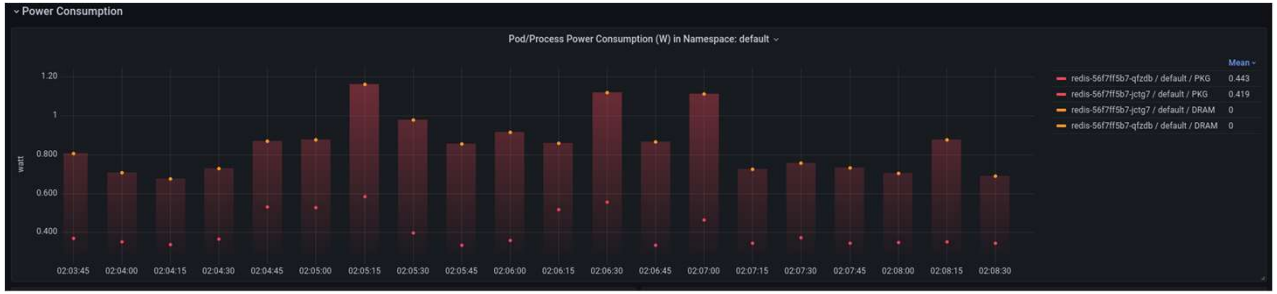
Container level accuracy – Actual Workload

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

- Scale up Redis (Replica: 1->2)




```
jie@jie-nuc:~$ cat power.csv
Pkg,Core,Uncore,Dram
3.315,1.168,0.010,0.000
3.384,1.234,0.011,0.000
3.527,1.386,0.011,0.000
```



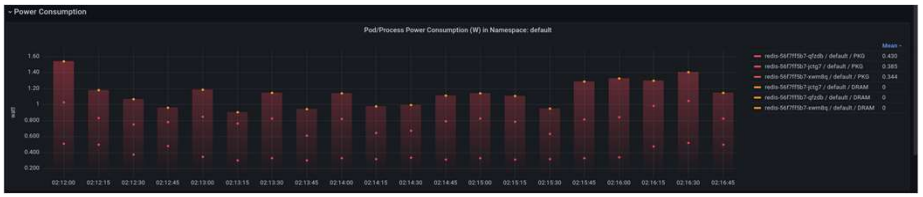
- Validator calculated PKG power increase:
  0.143W
- Prometheus Queried new scaled POD's Dynamic Power:
  0.138W

Container level accuracy – Actual Workload

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

- Scale up Redis (Replica: 2->3)


```
jie@jie-nuc:~$ kubectl get pod
NAME                     READY   STATUS    RESTARTS   AGE
redis-56f7ff5b7-jctg7    1/1     Running   0          4h2m
redis-56f7ff5b7-qfzdb    1/1     Running   0          116m
redis-56f7ff5b7-xwm8q    1/1     Running   0          100m
jie@jie-nuc:~$ cat power.csv
Pkg,Core,Uncore,Dram
3.315,1.168,0.010,0.000
3.384,1.234,0.011,0.000
3.527,1.386,0.011,0.000
3.599,1.498,0.010,0.000
```

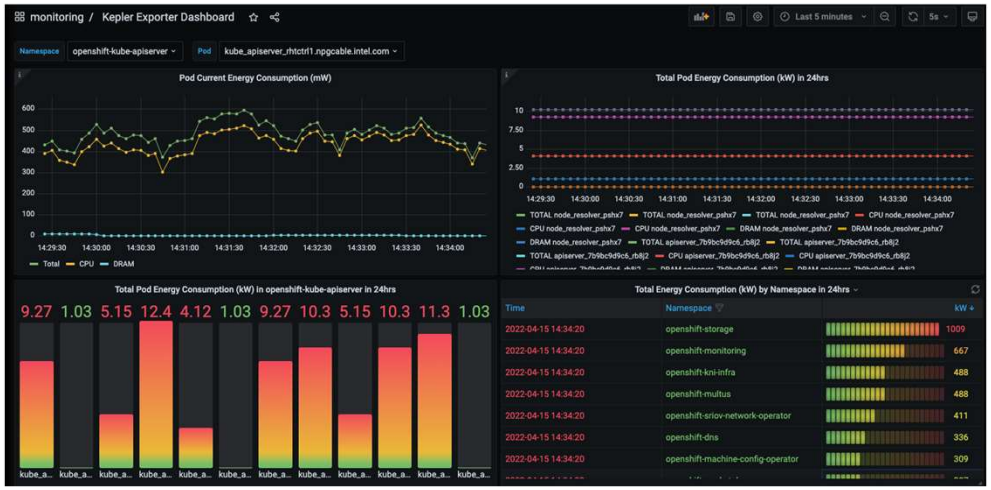


Replica: 0->1	Replica: 1->2	Replica: 2->3	Kepler			Comparison		
Validator PKG Delta(W)	Validator PKG Delta(W)	Validator PKG Delta(W)	POD1 Dyn PKG	POD2 Dyn PKG	POD3 Dyn PKG	POD1 Deviation	POD2 Deviation	POD3 Deviation
0.069	0.143	0.072	0.071	0.138	0.068	2.90%	-3.50%	-5.56%

Platform Validation Framework

❑Manual test cases

- Involve more actual workloads
  - ✓ Cloud native AI pipeline
  - ✓ ……
- Leverage cloud native observability and data visualization solutions
  - ✓ Prometheus
  - ✓ Grafana
  - ✓ Open Telemetry
  - ✓ …
- Continuous case automation



https://github.com/sustainable-computing-io/kepler/blob/main/doc/dashboard.png

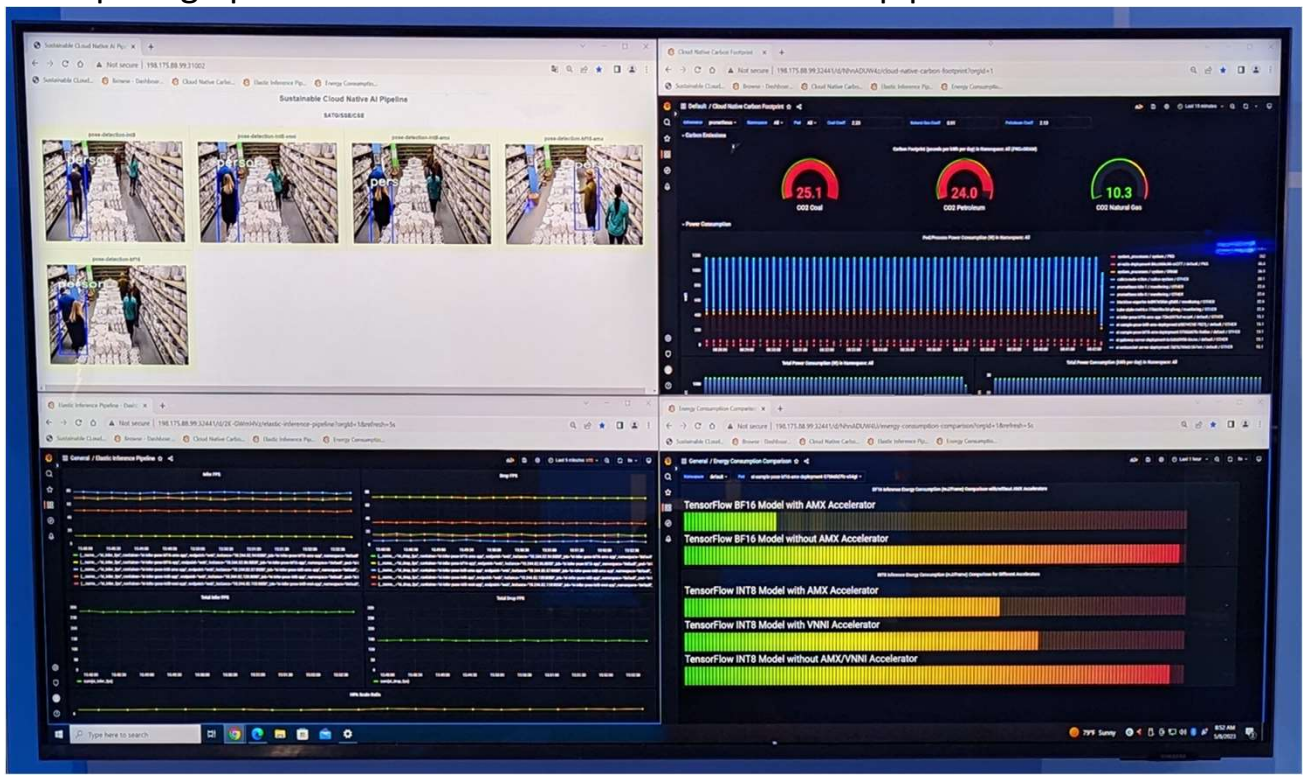
Practice on actual workload

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

❑ Cloud Native AI pipeline

- : https://github.com/intel/cloud-native-ai-pipeline
- Sustainable computing optimization on cloud native AI inference pipeline

Single Page Application: Multi-input video's Inference result comparison.



Kepler Dashboard: Container Power Consumption Comparison

Inference FPS Comparison

Inference Energy Efficiency (Joule/Frame) Comparison

Furthermore…

❑ Is Kepler accurate?

- On node level, Kepler is accurate
- On container level, there is no effective method to check, but Kepler's methodology and its power model is based on scientific method and could leverage the AI ML/DL technologies to continuously improve.

❑ Future works

- Actual workload carbon footprint
- Extend platform scenario from Bare Metal to VM, especially for CSPs
- Extend validation scenario from platform validation to model validation, improve accuracy
- …

Q & A


KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
Thank you