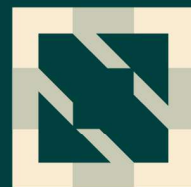


**KubeCon**



**CloudNativeCon**

**S OPEN SOURCE SUMMIT**

**China 2023**





KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

# 在特定平台上，开普勒准确吗？

任捷 高级云软件工程师 英特尔  
陆科进 云软件架构师 英特尔

# 议程

- ❑ 开普勒 (Kepler) 基础知识
- ❑ 开普勒功率模型
- ❑ 针对开普勒的平台验证
- ❑ 未来的工作

# 开普勒基础知识

## ❑ Kepler (Kubernetes Efficient Power Level Exporter)

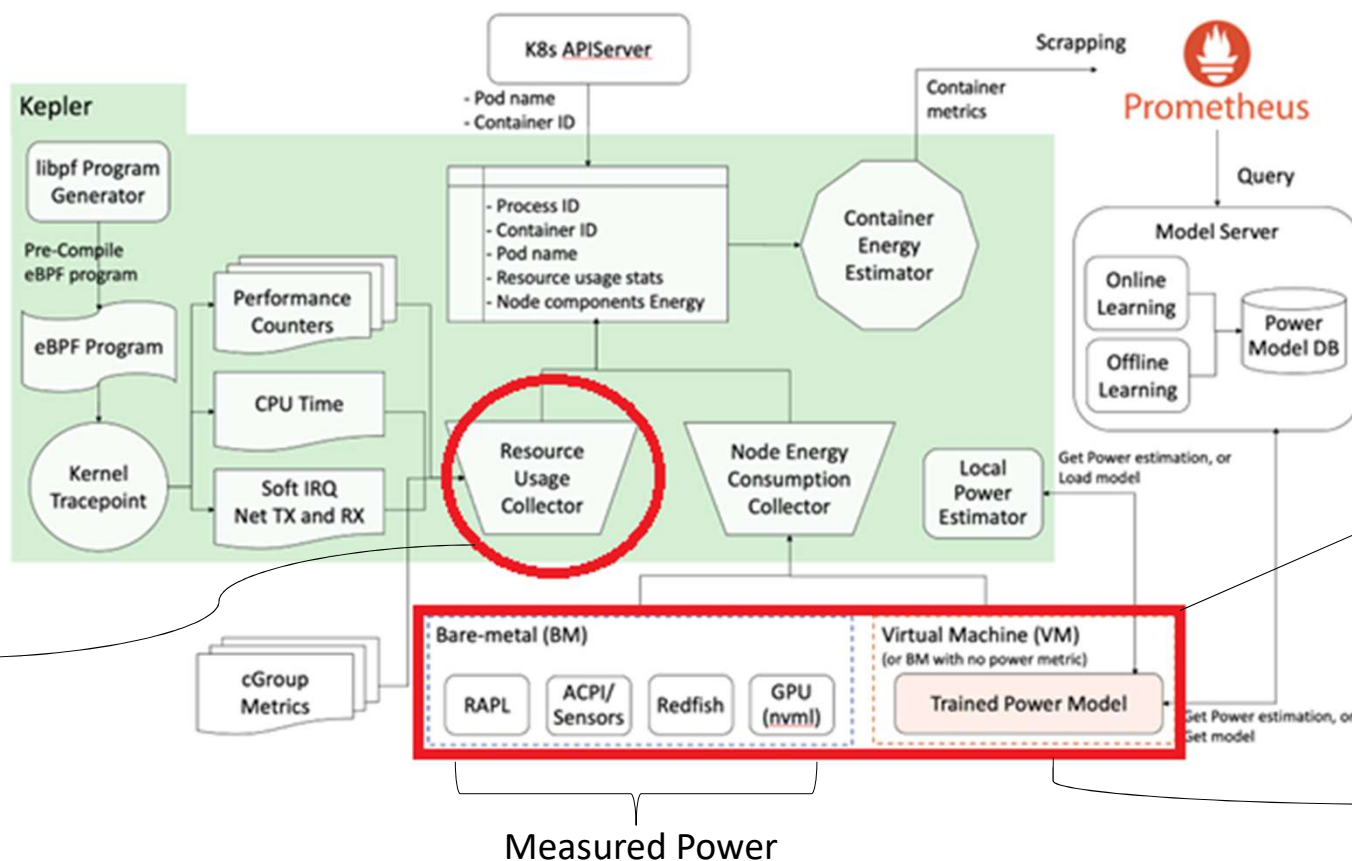


- 开普勒于2023年6月正式登陆CNCF，成为沙箱（Sandbox）项目
- 它利用eBPF技术对能耗相关的数据进行采集
- 它是能耗相关度量指标的收集器和上报工具
- 它特别针对云原生应用进行功率建模
- $\text{功率} = \text{能耗} / \text{时间}$

更多细节请参考本次KubeCon China 2023展会期间Kepler社区信息亭发布



# 开普勒基础架构

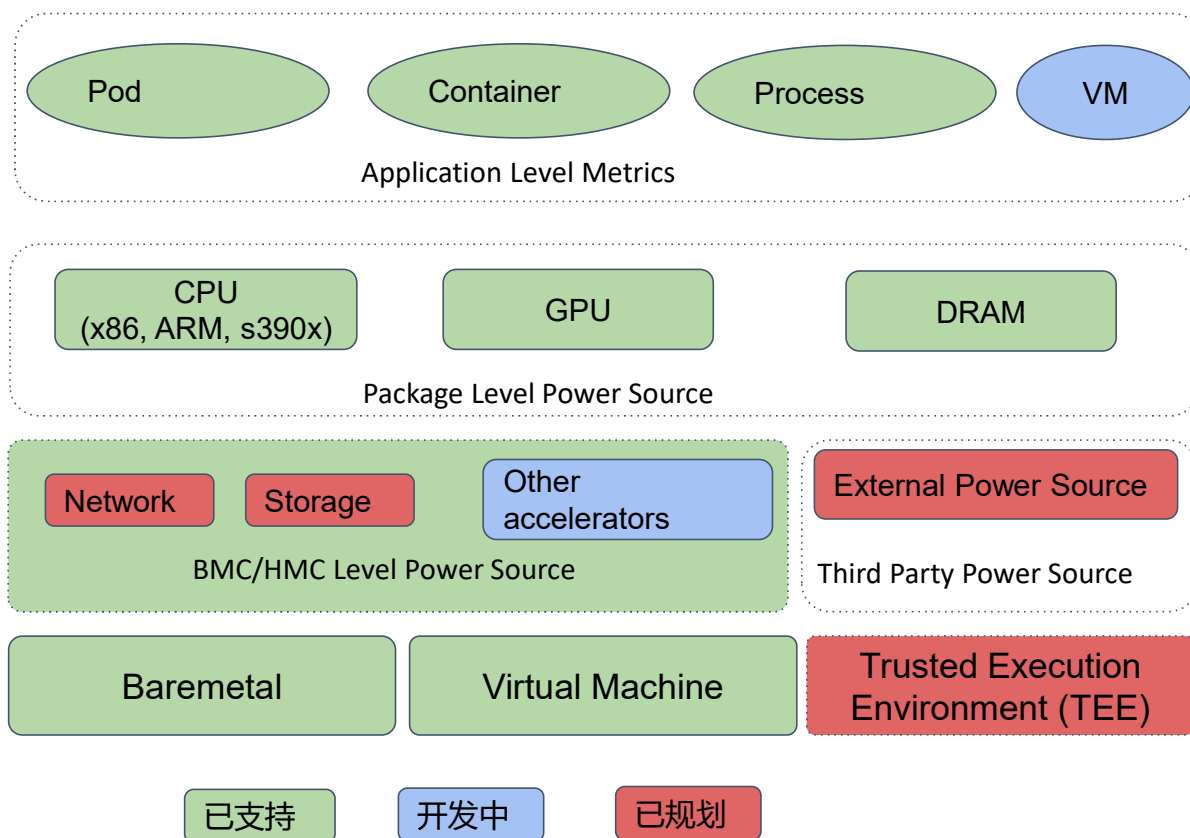


功耗模型：  
比率分配

特定平台

功耗模型：  
线性回归

# 开普勒支持矩阵



支持Pod/Container/Process级别的度量指标

支持x86, arm64, s390x架构的CPU, GPU以及内存功率统计

支持 BMC/HMC等外部功率数据源

支持基于裸金属和虚拟机的能耗测量, 计划调研可信执行环境 (TEE) 下的能耗测量.



# 开普勒功率模型

## □ 开普勒能耗数据来源

- 中央处理器（CPU）组件能耗 (带内)
  - x86
    - 运行时平均功率限制（RAPL, Running Average Power Limit）定义了诸如Package/Core/Uncore/DRAM的CPU功率域
    - RAPL 能耗数值可以从Linux 系统文件路径(/sys/class/powercap/) 或者 特定模型寄存器(Model Specific Register)中读取
  - ARM
    - Ampere Xgene系列CPU的硬件监控寄存器文件 (/sys/class/hwmon/)
- 平台能耗 (带外)
  - 高级配置与电源接口（ACPI）
  - 硬件管理控制台（HMC）(IBM s390x系列设备支持)
  - 基板管理控制器（BMC）(基于Redfish/IPMI管理)
- 图形处理器（GPU）能耗
  - 英伟达管理库(NVML, Nvidia Management Library)

## □ 绝对功率 = 动态功率 + 待机功率

- 绝对功率：暴露系统实时能耗度量指标的编程接口上报的数据即为绝对功率
- 动态功率：直接与系统资源利用率相关的功率
- 待机功率：系统恒定功率，与系统待机或运行负载与否无关

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

# 开普勒功率模型

## 比率方法

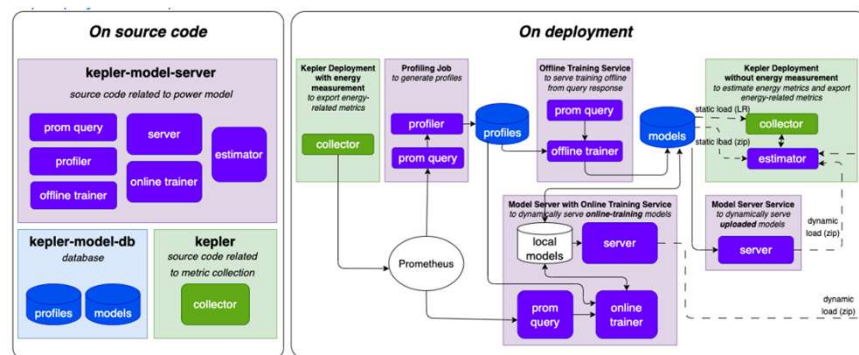
- 基于可测量的功率
- 计算容器资源利用率占整体工作节点资源利用率的比率
- 在各个容器中针对上述比例进行功率的分配

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

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

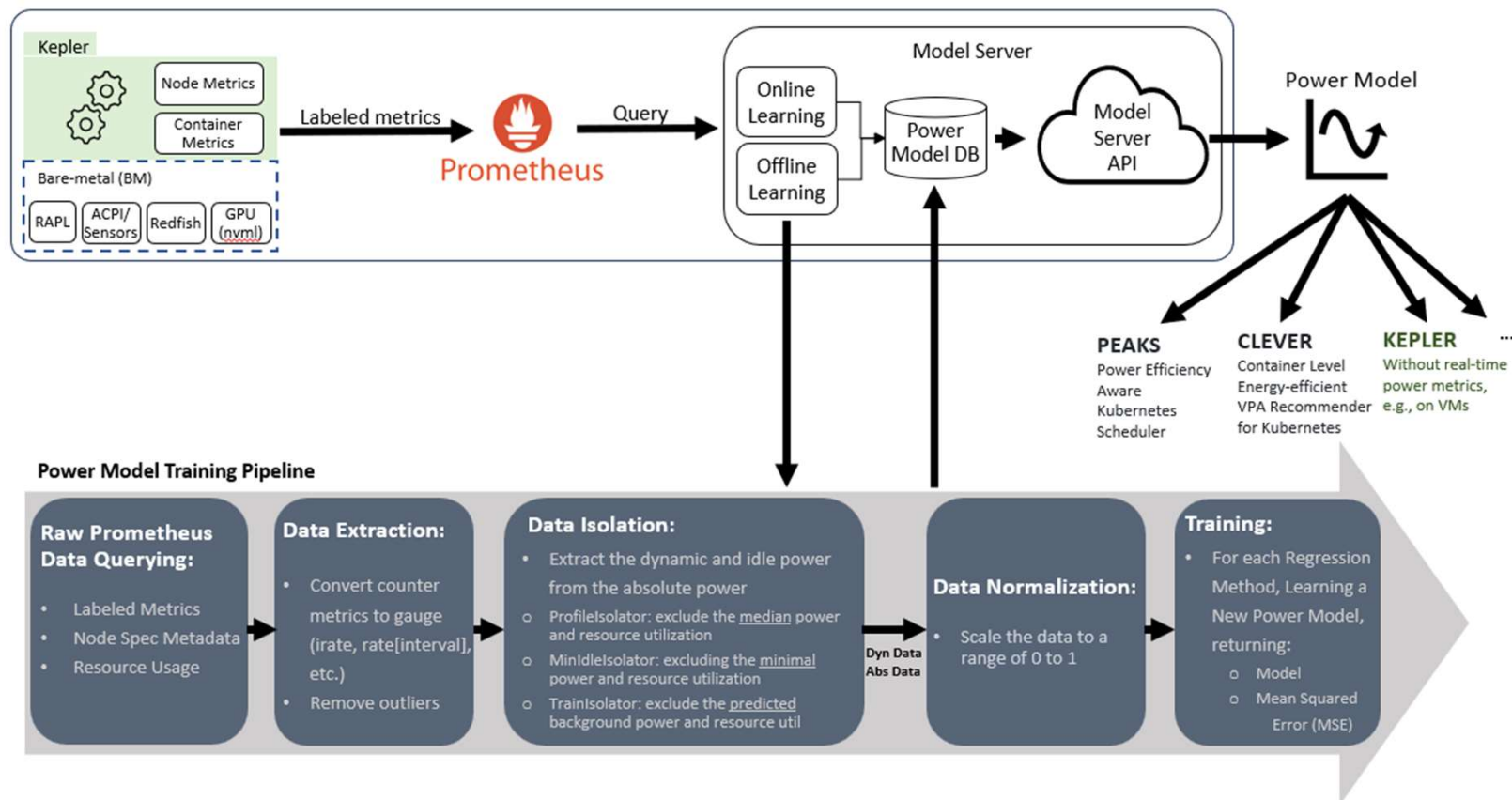
## 回归方法

- 基于系统硬件性能计数器中的数据
- 使用机器学习的回归方法训练模型和验证模型
- 使用训练好的模型进行功率预测





# 开普勒的功率模型训练



## 参考文献:

1. Kepler: A Framework to Calculate the Energy Consumption of Containerized Applications ([IEEE CLOUD 2023](#) 会议论文, IBM研究院 发布)
2. Exploring Kepler's Potentials: Unveiling Cloud Application Power Consumption (即将发表于CNCf技术博客), 作者: Marcelo Amaral, Sunyanan Choochootkaew, Huamin Chen 等

# 开普勒平台验证

## □ 解决了哪些关注

- 开普勒得到充分测试了么？
- 开普勒被特定平台很好地支持了么？
- 在特定平台上，开普勒的功率分布是准确的么？

## □ 框架设计

- 自动化 workflow
- 遵循开普勒的测试框架
- 独立验证工具包的设计
- 数据正确性和数据准确性检查
- 验证结果评估

## □ 更多的场景和验证视角

- 真实工作负载的碳足迹
- 不同视角下的功率可视化
- 将平台验证从裸金属扩展到虚拟机
- 将验证场景从平台验证扩展到模型验证

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

# 平台验证框架

## □ 工作机理与方法论

### ■ 自动化 workflow

- ✓ 手动触发 Github Action
- ✓ 在自管理的测试机上运行 workflow
- ✓ 容器化的测试镜像，平台无关

### ■ 遵循开普勒既有的测试框架: Ginkgo

- ✓ 领域特定的测试语言 (Domain Specific Language)
- ✓ 生成可定制的格式化的测试报告，比如JSON格式报告文件

### ■ 验证工具: 独立的用于功率计算和对比的工具

- ✓ 独立地在Intel x86裸金属平台上基于RAPL进行能耗收集和功耗计算的工具
- ✓ 工作机理: 采样与平均计算
- ✓ 对比的前提假设: 在目标应用部署前后收集功率数据，计算差值作为对比基准
- ✓ 对于非x86平台，开发者可以根据该平台特有的测量方法和工具实现类似的功能

### ■ 数据正确性和数据准确性检查

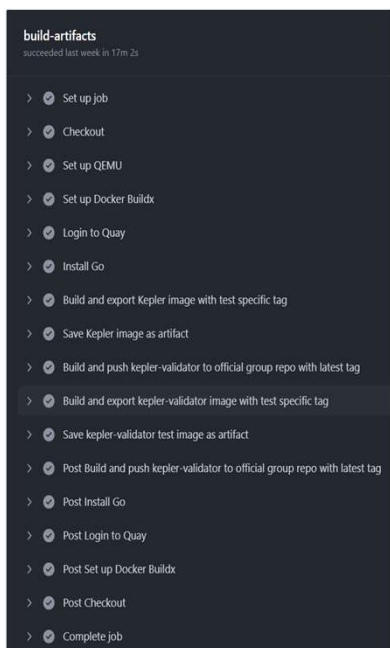
- ✓ 正确性检查测试用例: CPU 架构信息检查 (以Intel x86平台为例，利用cpuid工具), CPU组件功率数据源支持状况检查
- ✓ 准确性检查测试用例: 节点级别功率报告的准确性检查, 容器级别功率报告的准确性检查

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

# 平台验证框架

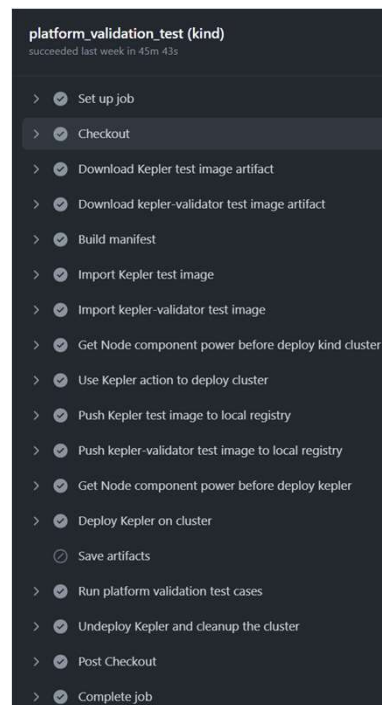
## 自动化 workflows

### ■ 第一阶段: 测试工件编译



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

### ■ 第二阶段: 验证测试



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

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

## ■ 测试用例设计

- ## ■ 测试用例的局限性

- ✓ 测试比较基准目前还局限于工作节点级别或CPU package级别功耗
- ✓ 容器级别功耗准确性判断的假设条件
- ✓ 测试环境扰动的影响: 多租户场景, 容器与虚拟机共存等

## ■ 测试结果评估

- ✓ 数据正确性: ☒
- ✓ 数据准确性:
  - ☒ 节点级别准确性符合预期
  - ☐ 容器级别准确性差异很大 (目前引入人工判断)

	Before Kepler Deployment		After Kepler Deployment																									
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	Kepler				All namespaces				system_processes							
													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
```



# 容器级别功耗准确性 - 实际工作负载

## 观察Redis部署前后的功耗变化（单实例）

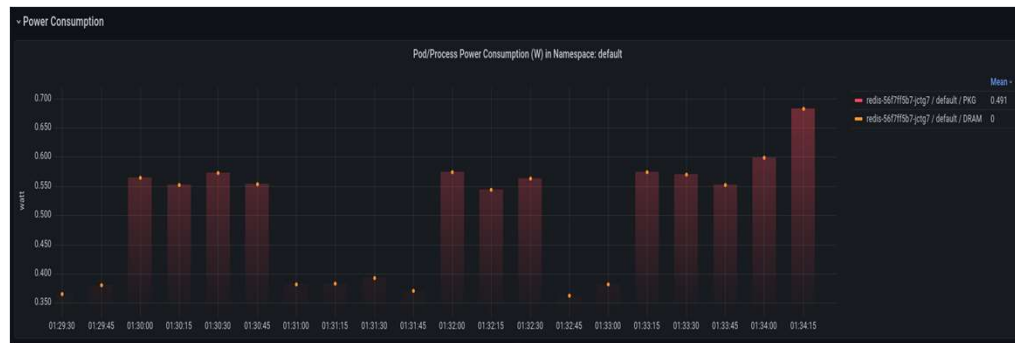


### Validator采样计算

```
jie@jie-nuc:~$ docker run -t --rm -v $(pwd):/output localhost:5801/platform-validation:x86-rapl /usr/bin/validator
I0922 01:32:48.993433 1 gpu.go:46] Failed to init nvmf, err: could not init nvmf: error
opening libnvidia-ml.so.1: libnvidia-ml.so.1: cannot open shared object file: No such file or
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: 92476666 (Core: 108333635, Uncore: 10169292) DRAM: 0]
Sample 3:
pre: map[0:Pkg: 92476666 (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
```

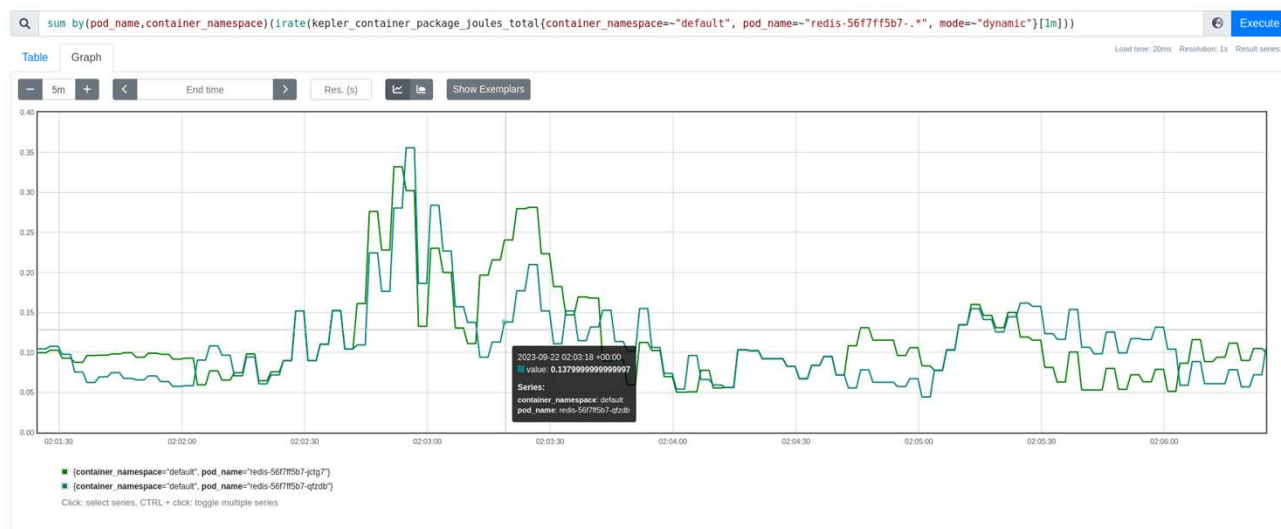
- ❖ Validator测量PKG 功耗增加:  
0.069W
- ❖ Prometheus查询新部署POD的动态功耗（平均值）:  
0.071W



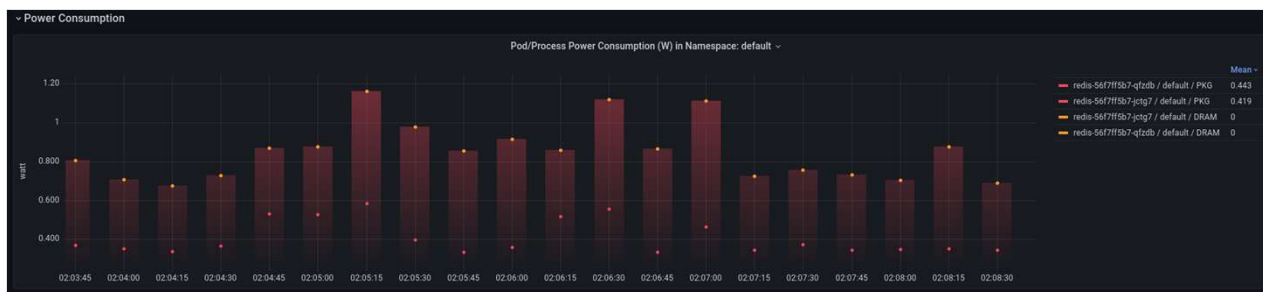


# 容器级别功耗准确性 - 实际工作负载

## 扩展Redis实例数 (replica=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
```



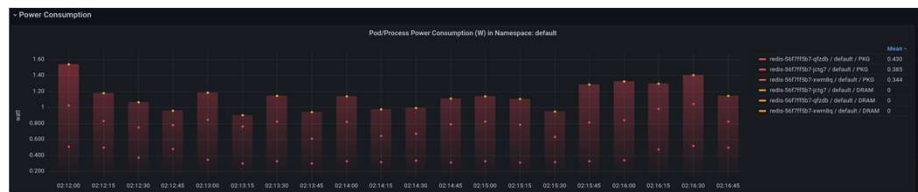
- ❖ Validator测量PKG 功耗增加: 0.143W
- ❖ Prometheus查询新扩展POD的动态功耗 (图示点): 0.138W

# 容器级别功耗准确性 - 实际工作负载

## 扩展Redis实例数 (replica=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	Validator	Validator	POD1	POD2	POD3	POD1	POD2	POD3
PKG Delta(W)	PKG Delta(W)	PKG Delta(W)	Dyn PKG	Dyn PKG	Dyn PKG	Deviation	Deviation	Deviation
0.069	0.143	0.072	0.071	0.138	0.068	2.90%	-3.50%	-5.56%

# 平台验证框架

## □ 人工测试用例

### ■ 引入更多的实际工作负载进行测试

✓ 云原生人工智能推理流水线 (Cloud native AI pipeline)

✓ .....

### ■ 充分利用云原生可观测性和数据可视化解决方案

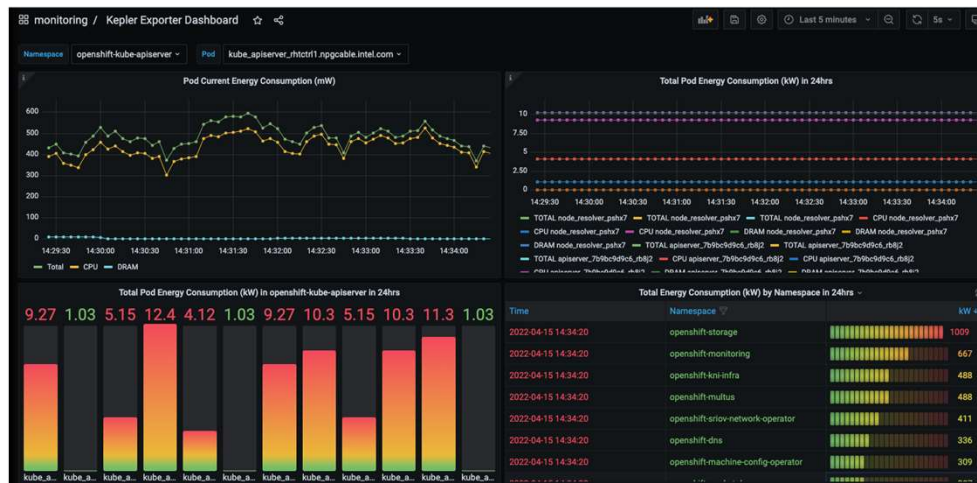
✓ Prometheus

✓ Grafana

✓ Open Telemetry

✓ ...

### ■ 人工测试用例的持续自动化




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

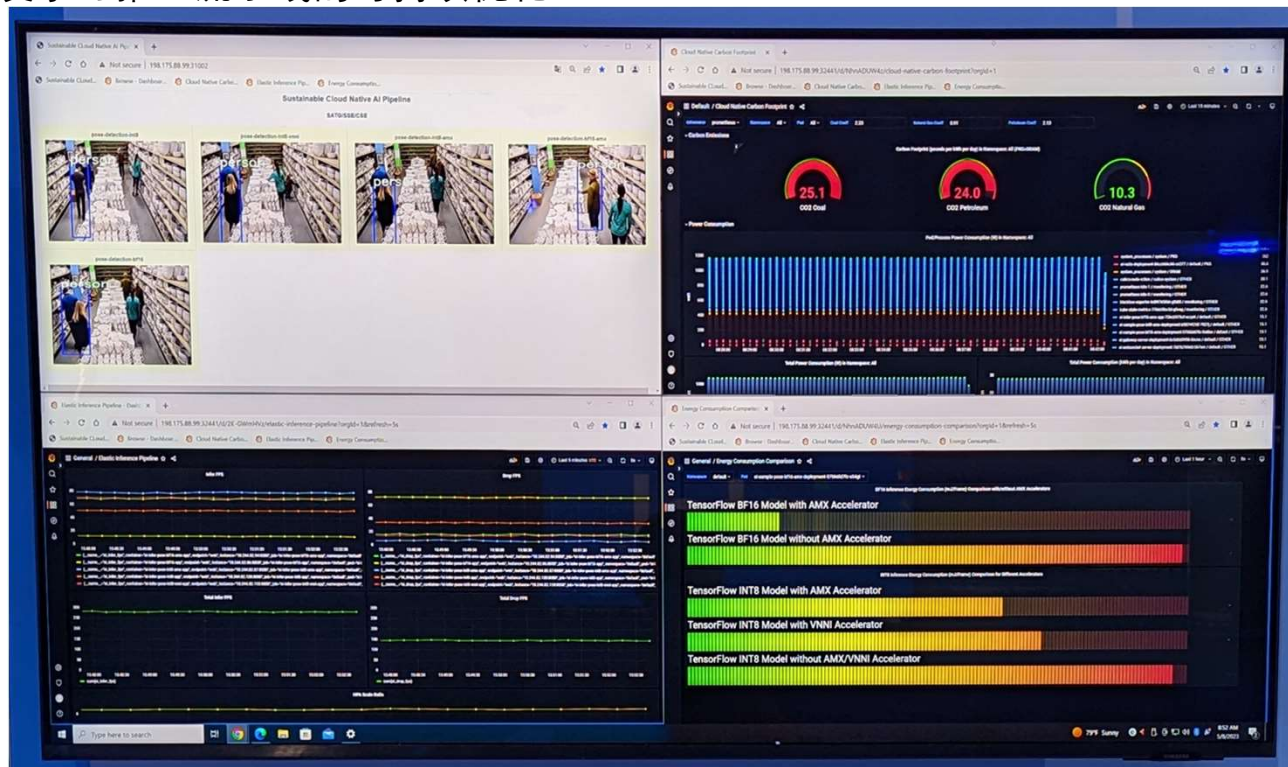


# 开普勒在实际工作负载运行中的实践

## ☐ 云原生人工智能推理流水线 (Cloud Native AI pipeline)

-  : <https://github.com/intel/cloud-native-ai-pipeline>
- 云原生深度学习推理流水线的可持续优化

单页网络应用程序:  
动态对比多路视频  
输入在云原生边缘  
服务上进行AI推理



开普勒仪表盘:  
容器功耗对比展示

推理指标对比:  
每秒推理帧数  
运用加速器后  
指标提升明显

推理能耗效率对比:  
每帧推理消耗焦耳数

# 总结与展望...

## □ 开普勒的准确性

- 在节点级别是基本准确的
- 在容器级别，目前尚缺乏有效的验证手段，但开普勒的工作机理，功率模型是基于科学的手段，并可以结合人工智能深度学习技术持续改进的

## □ 未来的工作

- 实际工作负载的碳足迹
- 将平台场景从裸金属扩展到虚拟机，特别是针对云服务提供商提供可持续计算解决方案
- 将验证场景从平台验证扩展到模型验证，改进模型推理的准确性
- ...

提问环节?



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

感谢参与