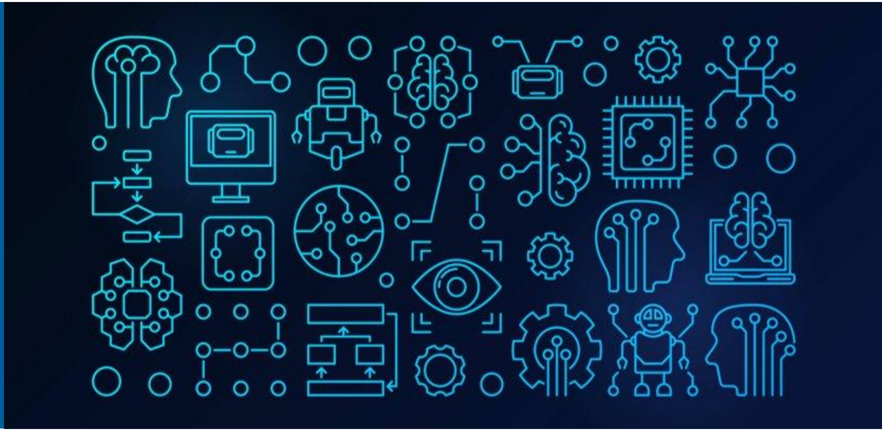
Scalable Edge Cyberinfrastructure for Science-driven AI Workflows



Yongho Kim
yongho.kim@anl.gov
Assistant Computer Scientist
Mathematics and Computer Science division
Argonne National Laboratory

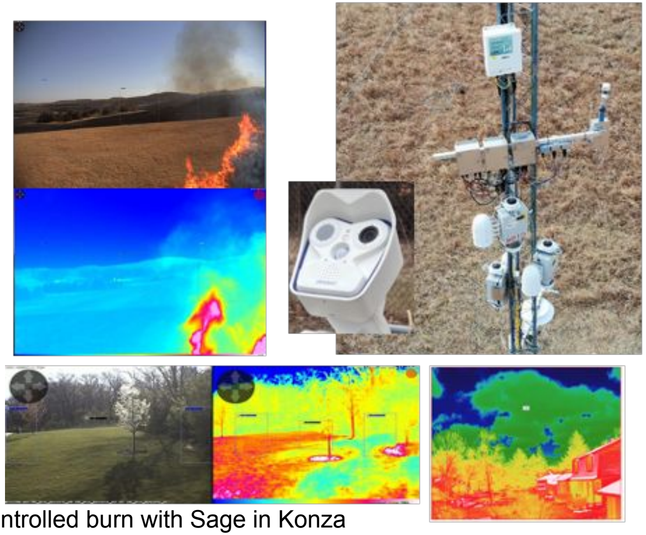
Invited Talk
May 19th, 2023

Wildfire Detection and Prediction

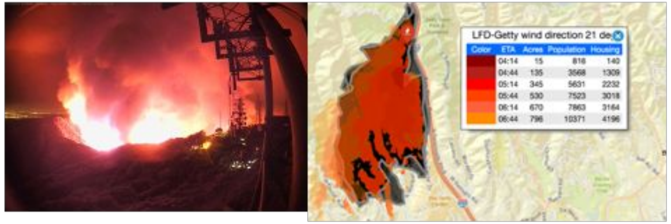
AI@Edge for wildfire detection linked to HPC simulations

WIFIRE

Ilkay Altintas, UCSD, Co-PI for SAGE



source: controlled burn with Sage in Konza (https://sagecontinuum.org/news/sage-neon-deploy-konza)



LFD-Getty wind direction 21 de

Color	ETA	Acres	Population	Housing
🟥	04:14	15	818	140
🟥	04:44	135	3568	1309
🟥	05:14	345	5631	2232
🟧	05:44	530	7523	3018
🟧	06:14	670	7863	3164
🟧	06:44	796	10371	4196

Sage project will move Pan-Tilt-Zoom cameras toward suspected outbreaks, and use infrared cameras to build self-supervised AI training

source: WIFIRE: wildfire detection and monitoring (https://wifire.ucsd.edu/)

Sage (beta)

Edge (Field cameras)

1. Smoke detection at the edge of the network using Camera Imagery

WIFIRE

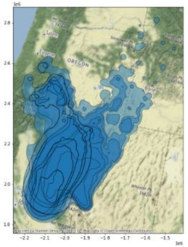
Cloud (SDSC)

2. Wildfire simulations to determine the severity and direction of fires

THE UNIVERSITY OF UTAH

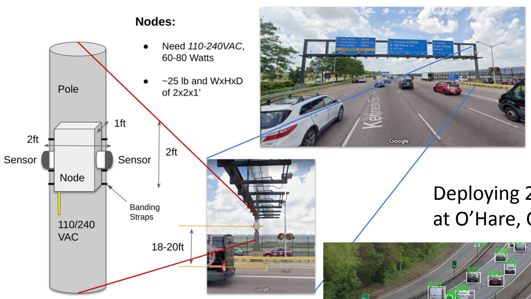
HPC (Utah)

3. Pollution Concentration maps to support decision-making



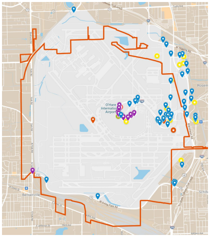
source: a scientific workflow on understanding air pollution caused by wildfires, Daniel Balouek-Thomert, Scientific Computing and Imaging (SCI) Institute, University of Utah

DOE VTO: Understanding traffic type, flow, and density

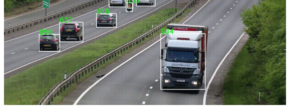


Nodes:

- Need *110-240VAC*, 60-80 Watts
- ~25 lb and WxHxD of 2x2x1'



Deploying 25+ at O'Hare, Chicago

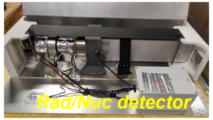


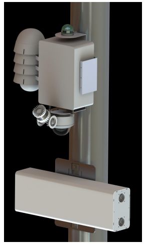
sources: Street view in Google Maps (https://www.google.com/maps)

Dual cameras to capture approaching and leaving traffic for in-situ analysis.

NNSA: Advanced radiation sensors for safer cities

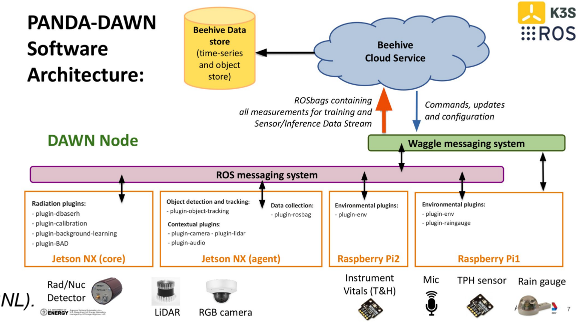
Physical sensors include - Temp, Pressure & Humidity, Rain Gauge, Microphone, Camera, LiDAR, and Rad/Nuc Detector





Deploying 20+ Sensors in Chicago

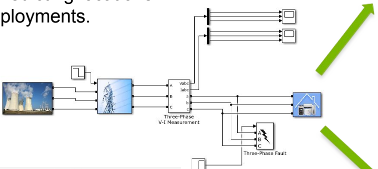
A deployment illustration by Victor Negut (LBNL).



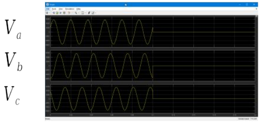
PANDA-DAWN Software Architecture:

DOE CRADA with Exelon: Advanced Sensors for Grid Stress and Load Forecasting

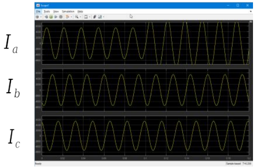
Simulations predicting locations for sensor deployments.



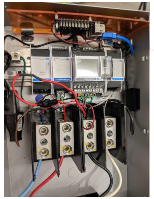
Three-Line-to-Ground Fault

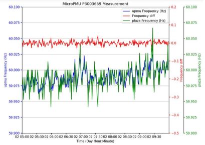


Single-Line-to-Ground Fault



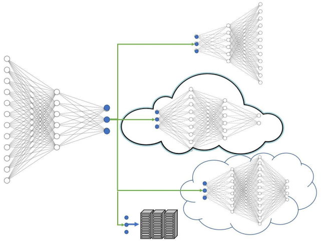
uPMU

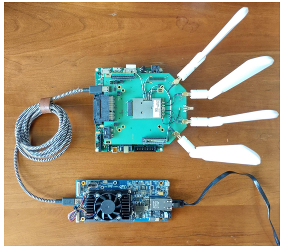




Analysis to quantify stress and forecast load (weather data, traffic data) at the edge.

DOE ASCR - Migratory Computation for the Wireless 5G Digital Continuum

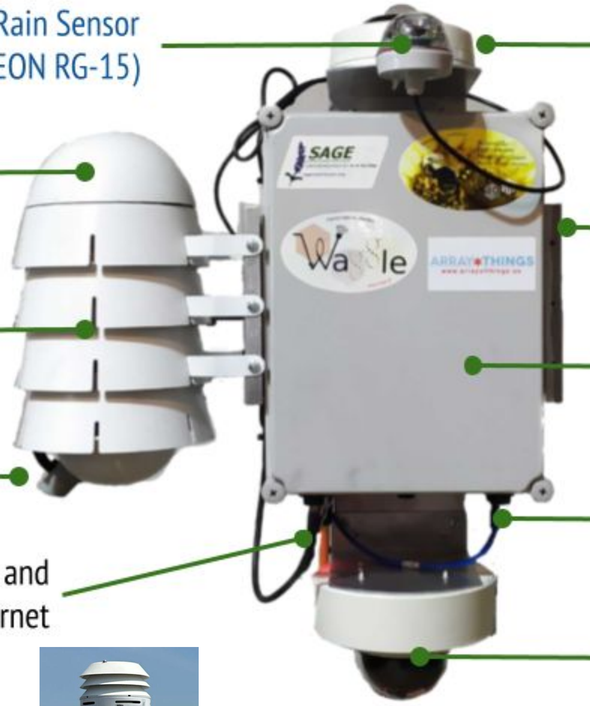




mmWave enabled Waggle nodes will connect with Nokia's NDAC solution with a 5G non-standalone configuration to offload computation on the MEC.

Extend anytime networks to be anywhere/anytime. Explore approaches where ensemble is executed on multiple parts of the continuum, and may use a variety of inputs.

The nodes are about 3 ft tall, 2 ft wide, and 1 ft deep, and wight about 30 lb.



Optical Rain Sensor (HYDREON RG-15)

POE Sky facing camera (HANWHA XNF-8010RV or XNV-8081Z)

POE Sensor Stevenson Shield

Additional Sensors Mounting Location

Relative humidity, barometric pressure, ambient temperature sensor (BOSCH BME680)

NVIDIA Xavier NX GPU, power, wireless communication (4G/WiFi) and node management.

Microphone (ETS ML1-WS)

POE and USB ports for Sensors (additional expansion ports)

110/230V AC Power and Ethernet

POE Ground Facing Camera (HANWHA TEC XNV-8081Z)

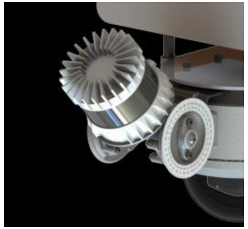
Attachable sensors and actuators

micro synchrophaser

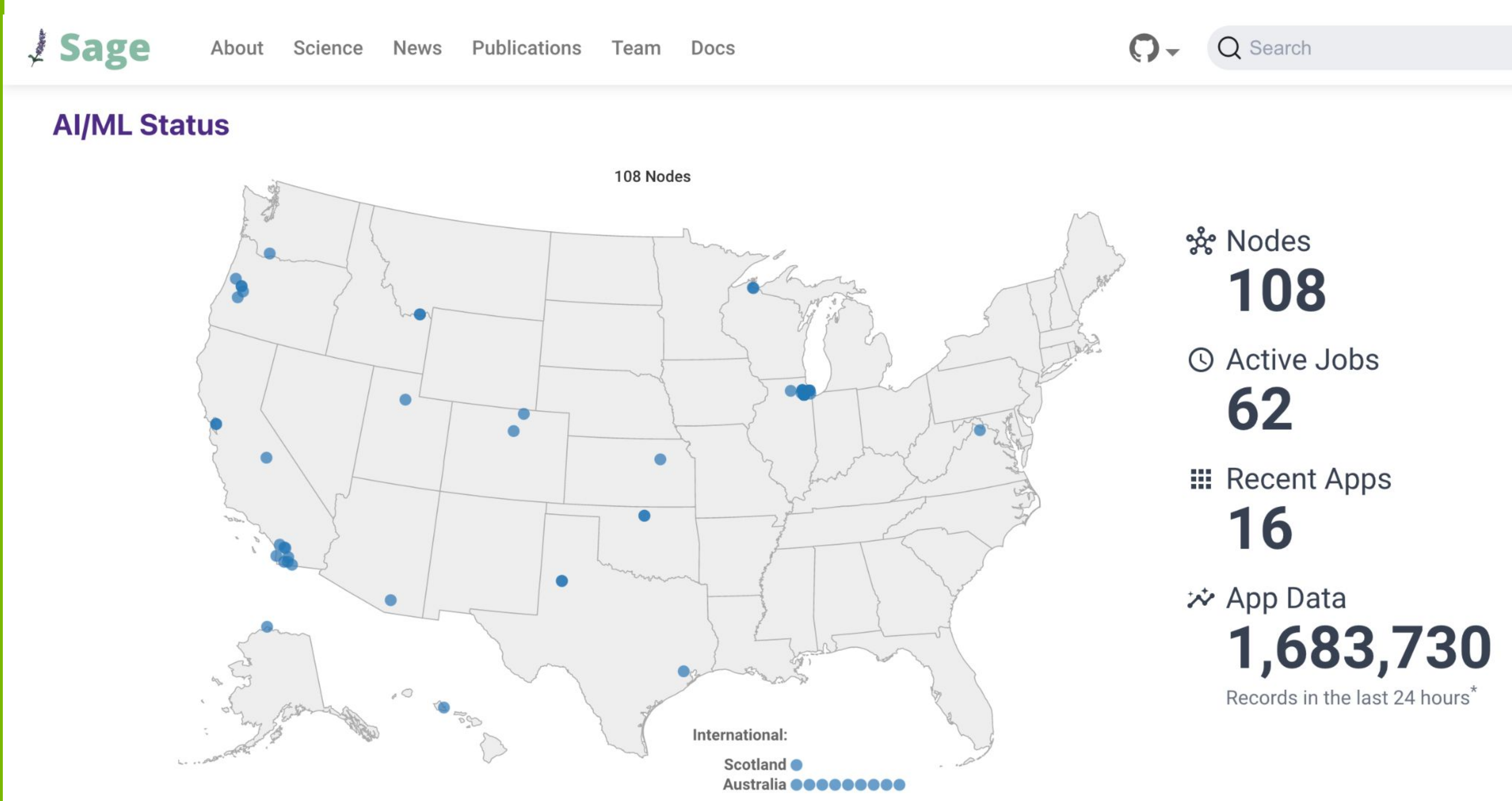
thermal imaging camera with a pan-tilt mount

< image credit: Scott Collis, CROCUS project, Argonne National Laboratory

air-quality sensor



LiDAR



AI/ML Status

108 Nodes

International:

Scotland ●

Australia ●●●●●●●●●

Nodes
108

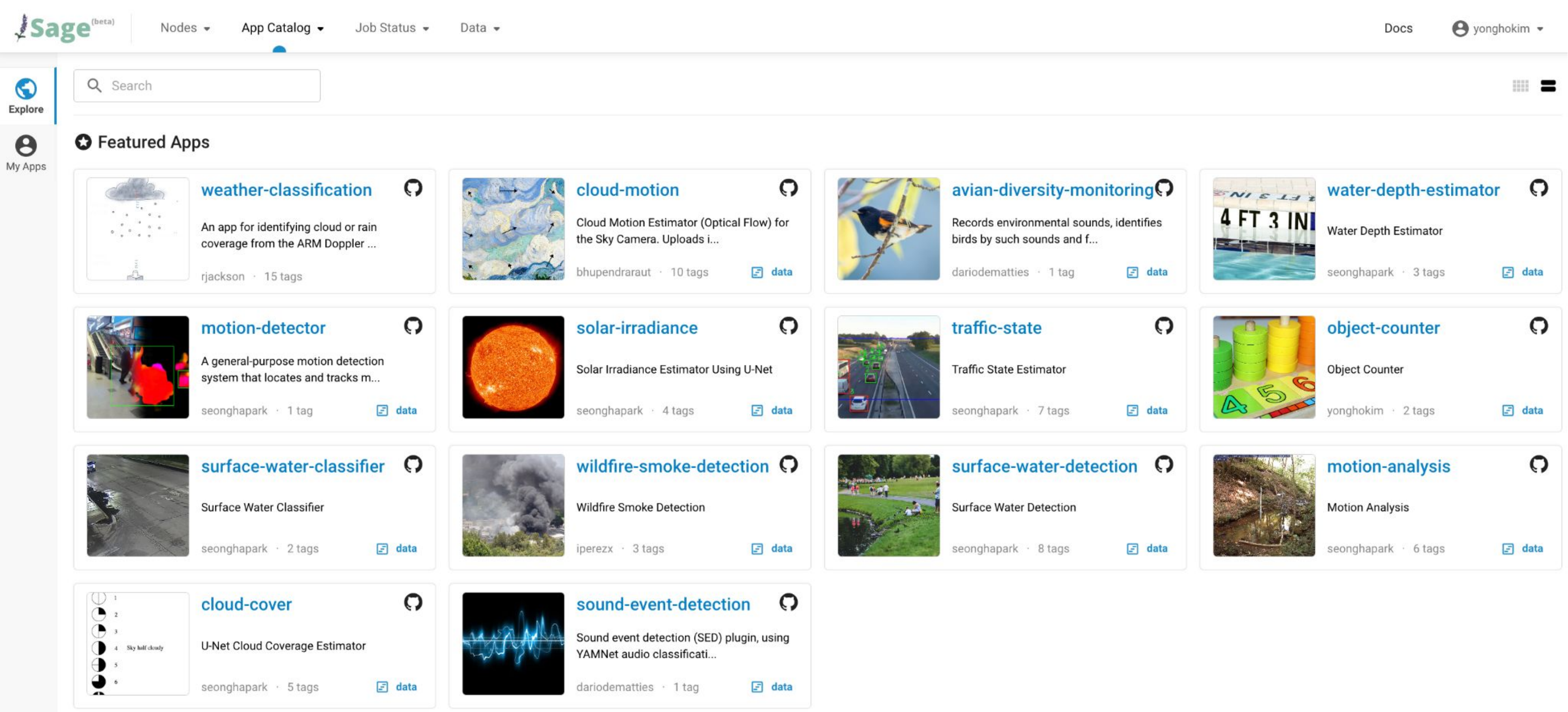
Active Jobs
62

Recent Apps
16

App Data
1,683,730

Records in the last 24 hours*

Source: https://sagecontinuum.org/



Featured Apps

weather-classification
An app for identifying cloud or rain coverage from the ARM Doppler ...
rjackson · 15 tags

cloud-motion
Cloud Motion Estimator (Optical Flow) for the Sky Camera. Uploads i...
bhupendraraut · 10 tags

avian-diversity-monitoring
Records environmental sounds, identifies birds by such sounds and f...
dariodematties · 1 tag

water-depth-estimator
Water Depth Estimator
seonghapark · 3 tags

motion-detector
A general-purpose motion detection system that locates and tracks m...
seonghapark · 1 tag

solar-irradiance
Solar Irradiance Estimator Using U-Net
seonghapark · 4 tags

traffic-state
Traffic State Estimator
seonghapark · 7 tags

object-counter
Object Counter
yonghokim · 2 tags

surface-water-classifier
Surface Water Classifier
seonghapark · 2 tags

wildfire-smoke-detection
Wildfire Smoke Detection
iperezx · 3 tags

surface-water-detection
Surface Water Detection
seonghapark · 8 tags

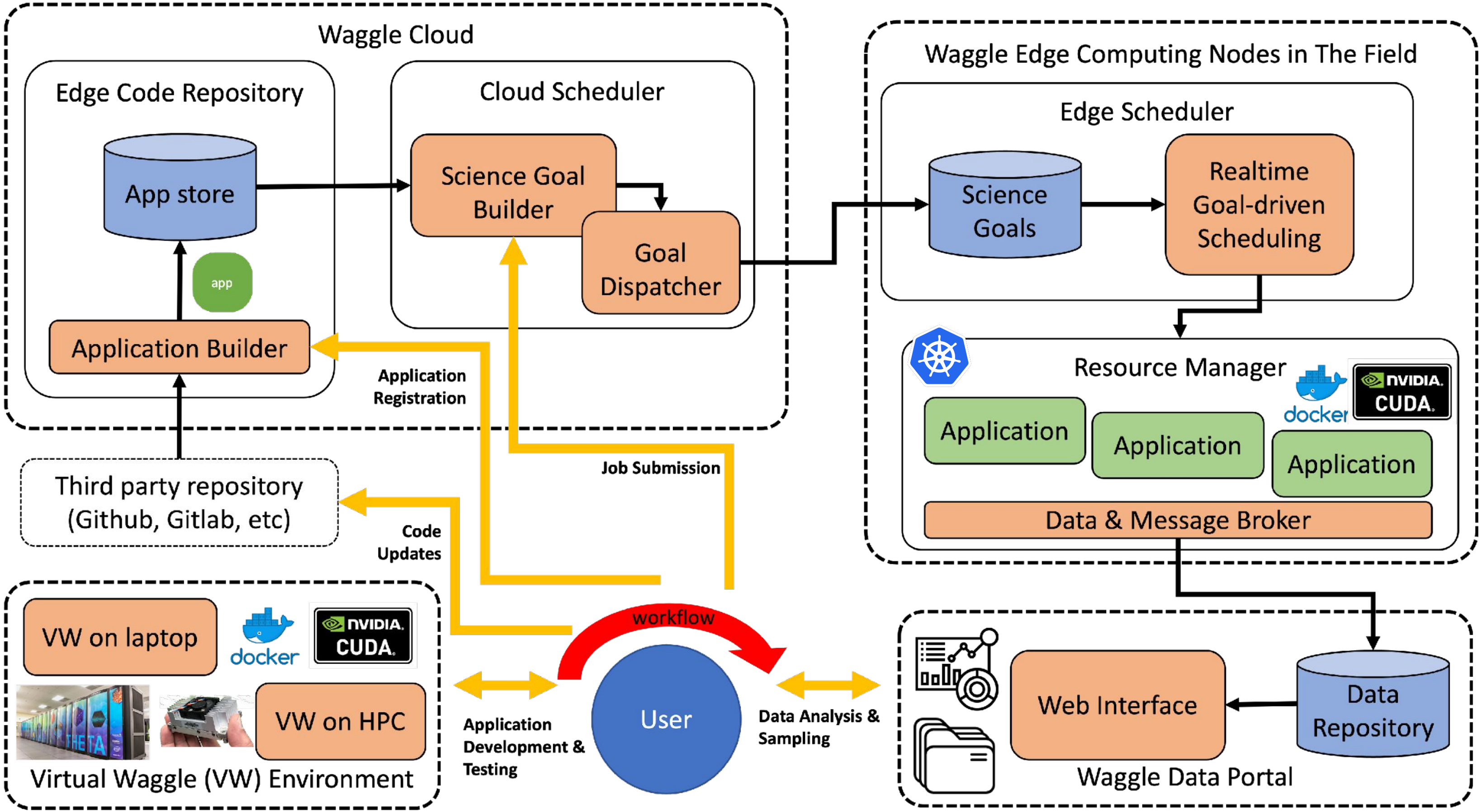
motion-analysis
Motion Analysis
seonghapark · 6 tags

cloud-cover
U-Net Cloud Coverage Estimator
seonghapark · 5 tags

sound-event-detection
Sound event detection (SED) plugin, using YAMNet audio classificati...
dariodematties · 1 tag

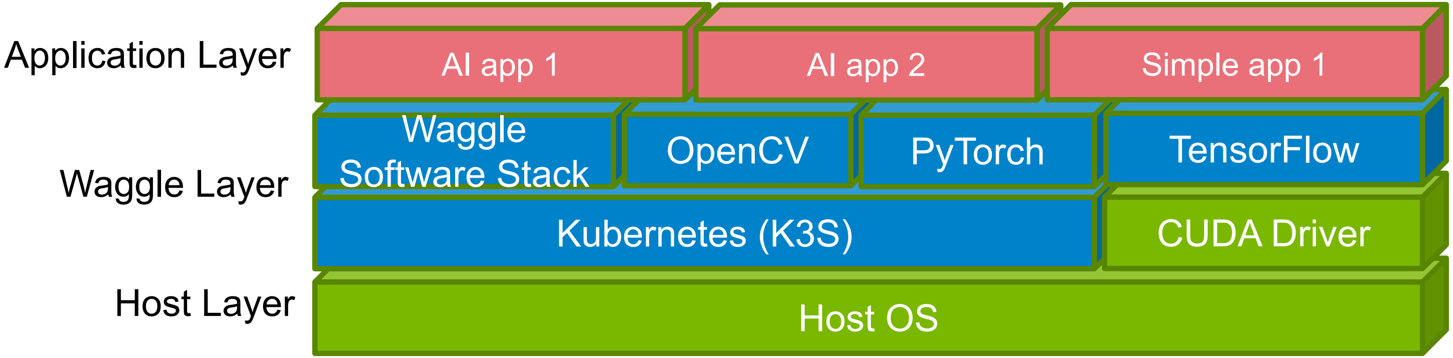
Source: https://portal.sagecontinuum.org/apps/explore

The Waggle Programming Model

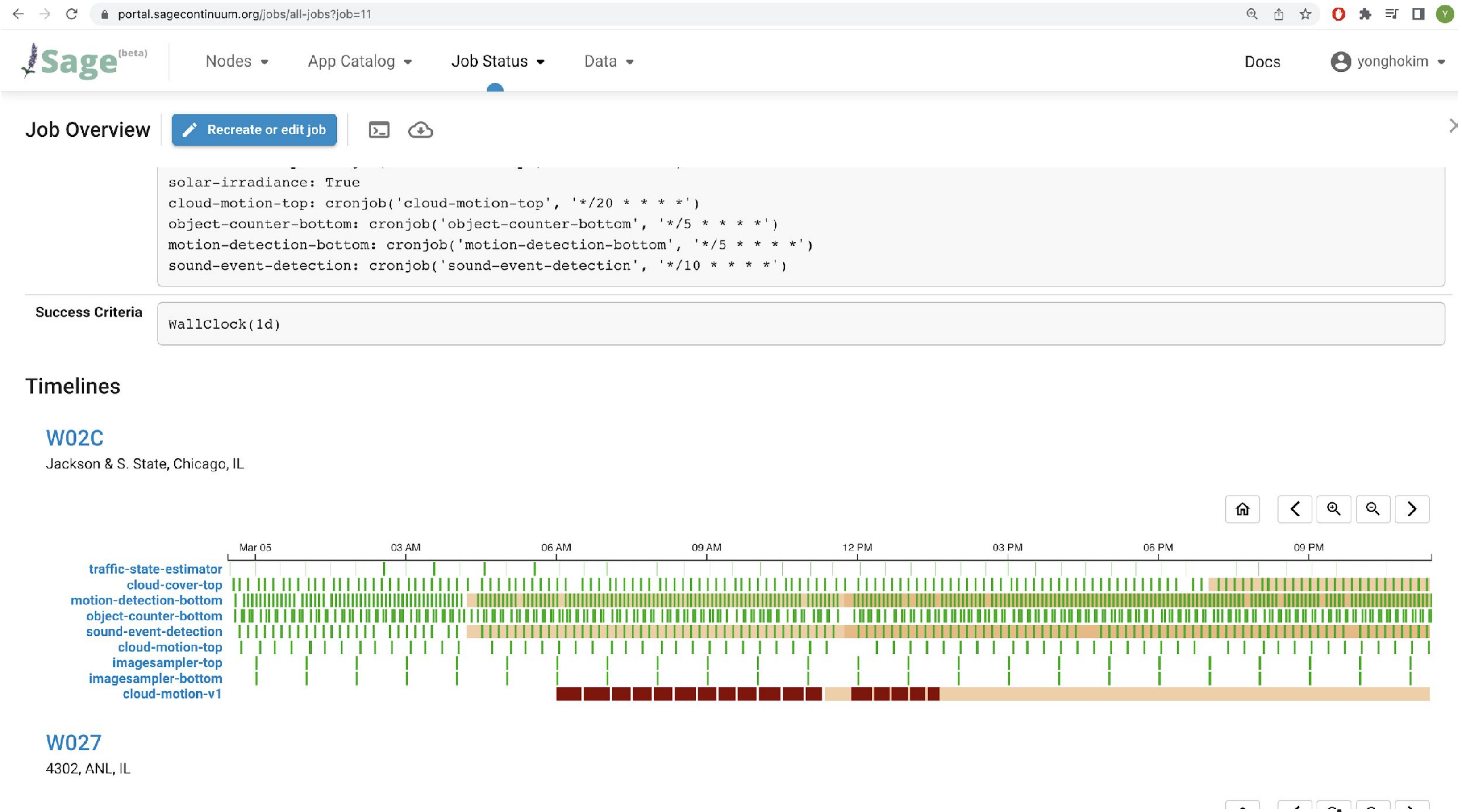


Waggle Software Stack and AI@Edge Applications

- Packaging applications including popular machine learning tools
- The Waggle layer ensures that packaged applications access hardware resources including CUDA on Waggle nodes
- Enabling a multi-tenant environment by isolating computing environment and allocating requested resources to individual application



Job Dashboard



```
solar-irradiance: True
cloud-motion-top: cronjob('cloud-motion-top', '*/20 * * * *')
object-counter-bottom: cronjob('object-counter-bottom', '*/5 * * * *')
motion-detection-bottom: cronjob('motion-detection-bottom', '*/5 * * * *')
sound-event-detection: cronjob('sound-event-detection', '*/10 * * * *')
```

Success Criteria

```
WallClock(1d)
```

Timelines

W02C

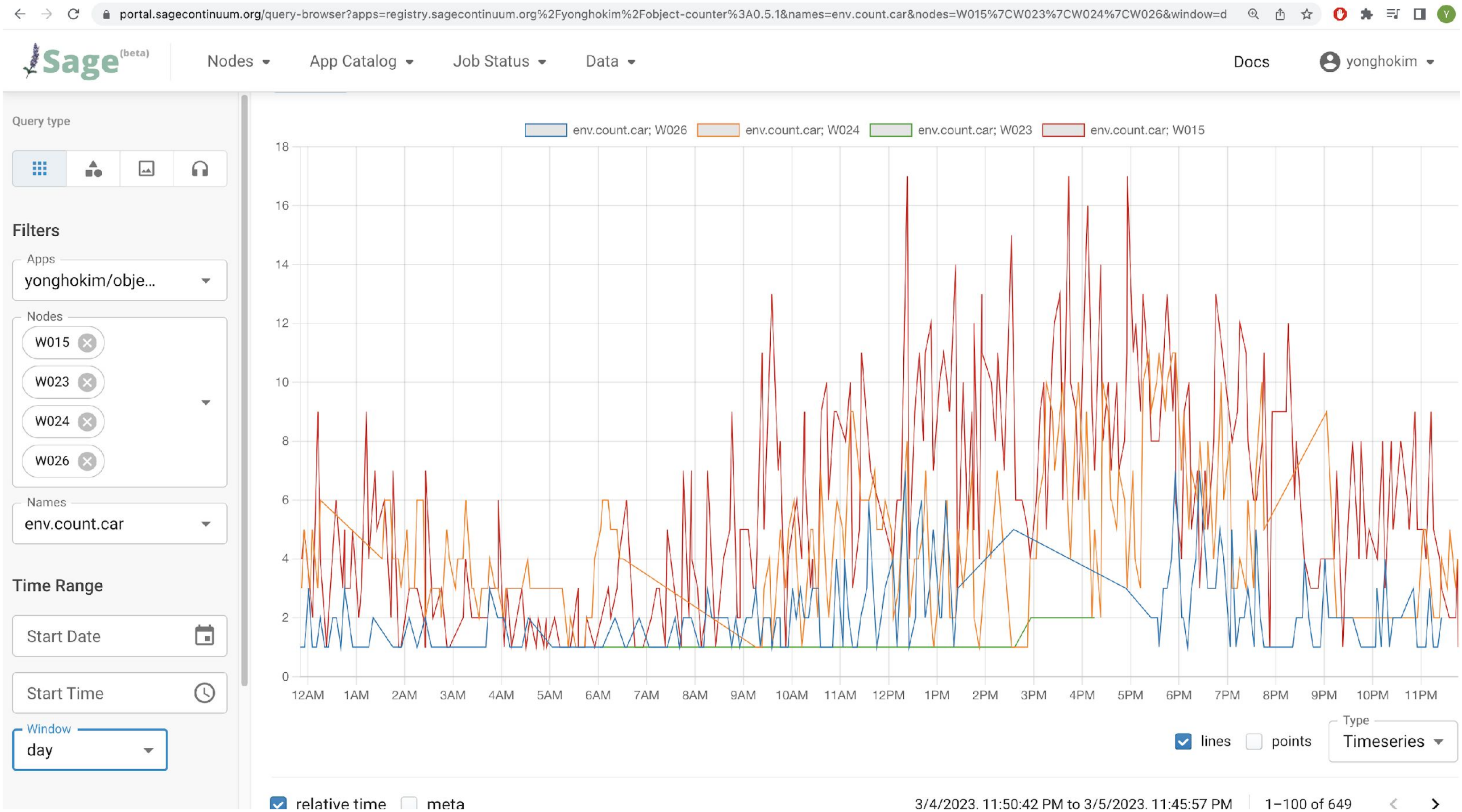
Jackson & S. State, Chicago, IL

W027

4302, ANL, IL

Source: https://portal.sagecontinuum.org/jobs/all-jobs

Viewing Data from Cloud (Waggle Beehive)



portal.sagecontinuum.org/query-browser?apps=registry.sagecontinuum.org%2Fyonghokim%2Fobject-counter%3A0.5.1&names=env.count.car&nodes=W015%7CW023%7CW024%7CW026&window=d

Query type

Filters

Apps: yonghokim/obje...

Nodes: W015 ⊗ W023 ⊗ W024 ⊗ W026 ⊗

Names: env.count.car

Time Range

Start Date

Start Time

Window: day

☑ lines □ points

Type: Timeseries

☑ relative time □ meta

3/4/2023, 11:50:42 PM to 3/5/2023, 11:45:57 PM 1–100 of 649

U.S. DEPARTMENT OF ENERGY Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

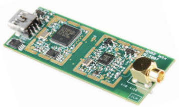
Credit for the Web UI design and implementation: Neal Conrad, Argonne

AI@Edge: Digital Continuum

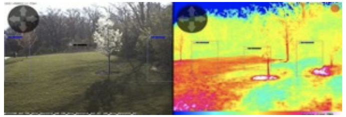
Sensors

LIDAR

Software Defined Radios



< image credit: Scot Collis, CROCUS project, Argonne National Laboratory



Thermal Imaging

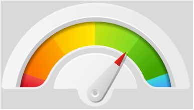
Facilities

NEON tower

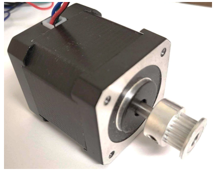
NEON mobile deployment platform

Actuators

Dynamic adaptation

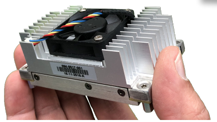


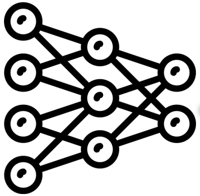
source: Freepik.com



Motors

Edge Computing





Scientific Data Analysis & Control

Artificial Intelligence
Deep Learning
Inference
Lightweight Training
Autonomous Action

Advanced Networking

New inference (model)
Adaptive controls / steering

Computation

Predictive Sim
Digital Twins
Data Analysis
Machine Learning

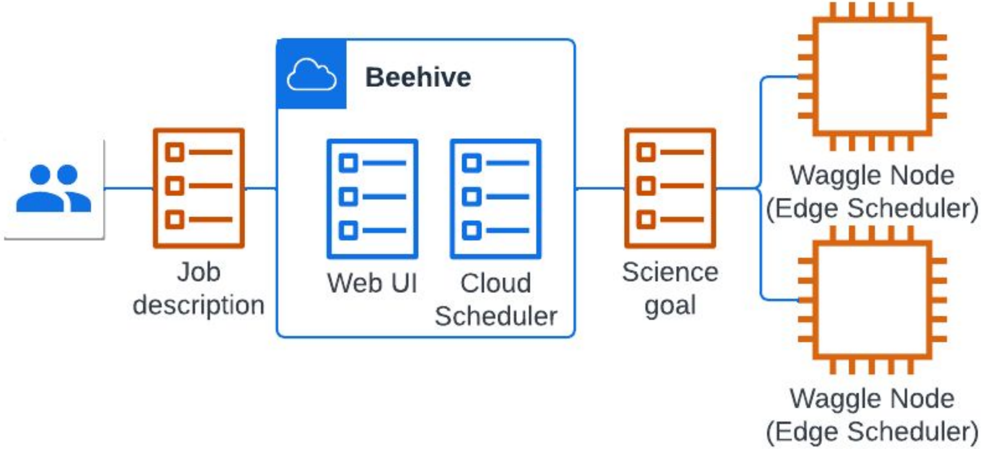
^ Image source: Aurora, Argonne Leadership Computing Facility, Argonne National Laboratory (https://www.alcf.anl.gov/aurora)

Challenges for Scalable Edge CI Supporting Science-driven Workflows

- An abstraction to express science-focused workflow execution of applications
  - execution timing and dependencies
- Vertical scaling
  - Self-monitoring and failover mechanisms as edge nodes are resource limited and suffered from dynamic environmental conditions
  - Dynamic parameter tuning supported by applications
- Horizontal scaling
  - Science-level scaling over multiple edge nodes

Waggle Edge Scheduler and Science Goals

- Users submit a job and creates corresponding science goal. The science goal is validated within Waggle context and propagated to target Waggle nodes for execution
- Science goal describes conditions on when and how to run user plugins
- To prevent resource conflicts between user jobs, we schedule user plugins and control their execution on Waggle nodes



More details: Yongho Kim, et. al, 2022, “Goal-driven scheduling model in edge computing for smart city applications”, Journal of Parallel and Distributed Computing

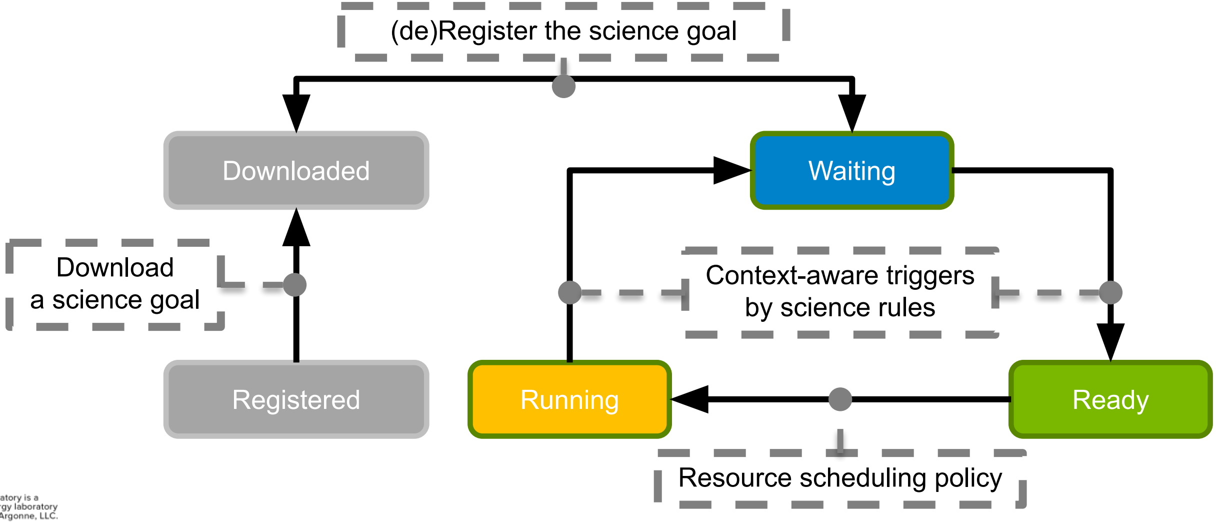
Life Cycle of A Plugin

- Plugins specified in science goals are downloaded from the edge code repository registry
- The edge scheduler,

  > **promotes** them from Waiting List based on conditions (Waiting > Ready)

  > **schedules** them when the resource is available (Ready > Running)

  > **cleans** them up after execution (Running > Waiting)


(de)Register the science goal
Downloaded
Waiting
Download a science goal
Context-aware triggers by science rules
Registered
Running
Ready
Resource scheduling policy

Science Rules for Application Scheduling

- A set of IF-THEN statements instructing the scheduler to server science goal

  - schedule my plugin if it is O'clock

  - schedule my plugin when there are more than 5 cars in the last minute

  - schedule my plugin when the sun rises

  - schedule my plugin if the dependent plugin has run

  - schedule my plugin when averaged noise level exceeds 50 dB

Science rules can also do other actions (*currently in alpha testing*)

  - publish "moderate rain" when total accumulation of rain in the last hour is greater than 3 mm

  - set my state to "30, 60" for my plugin to continue sweeping the pan-tilt camera

Science rules documentation at https://github.com/waggle-sensor/edge-scheduler/blob/main/docs/sciencerules/README.md

Example Job 1

```
---
name: vto-job                  << name of the job
plugins:
- name: image-sampler-left              << list of plugins to run
  pluginSpec:
  ...                          << specification of the plugin
- name: image-sampler-right
  pluginSpec:
  ...
- name: object-counter-left
  pluginSpec:
   ...
- name: object-counter-right
  pluginSpec:
   ...
nodeTags:
- WSN
- VTO     << list of nodes (all Waggle nodes under a project)
scienceRules:            << conditions on which the plugins run
- "image-sampler-left: cronjob('image-sampler-left', '0 * * * *')"
- "image-sampler-right: cronjob('image-sampler-right', '0 * * * *')"   << run every hour
- "object-counter-left: cronjob('object-counter-left', '15/45 * * * *')"
- "object-counter-right: cronjob('object-counter-right', '0/30 * * * *')"
successCriteria:
- WallClock(1d)              << criteria on when the job is considered as complete
```

Example Job 2

```
---
name: vto-video-sampler
plugins:
- name: video-sampler-left
  pluginSpec:
  image: registry.sagecontinuum.org/theone/video-sampler:0.2.4
  args:
  - -stream
  - rtsp://10.31.81.16:554/0/profile6/media.smp
  - -duration
  - 60
nodes:
  W023:
scienceRules:
- "video-sampler-left: cronjob('video-sampler-left', '*/5 12,13,14,20,21,22 * * *')"
successcriteria:
- WallClock(1d)
```

<< plugin Docker image from ECR

<< list of nodes; Waggle node W023 is specified

^ run every 5 minutes in rush hours

Example Job 3

```
---
name: water-detection
plugins:
- name: water-detector
  pluginSpec:
  image: registry.sagecontinuum.org/seonghapark/surface-water-detection:0.0.6
  selector:
    resource.gpu: true
nodeTags:
- WSN
- raingauge
- camera_bottom
scienceRules:
- "water-detector: rate('env.raingauge.total_acc') > 3 and cronjob('water-detector', '*/10 * * * *')"
successcriteria:
- WallClock(1d)
```

<< the plugin requires GPU resource to run

<< list of nodes; all Waggle nodes that have a bottom-facing camera and rain gauge

^ run every 10 minutes if it is raining

- More science rule functions can be found:

  https://github.com/waggle-sensor/sciencerule-checker

Scheduling Policies

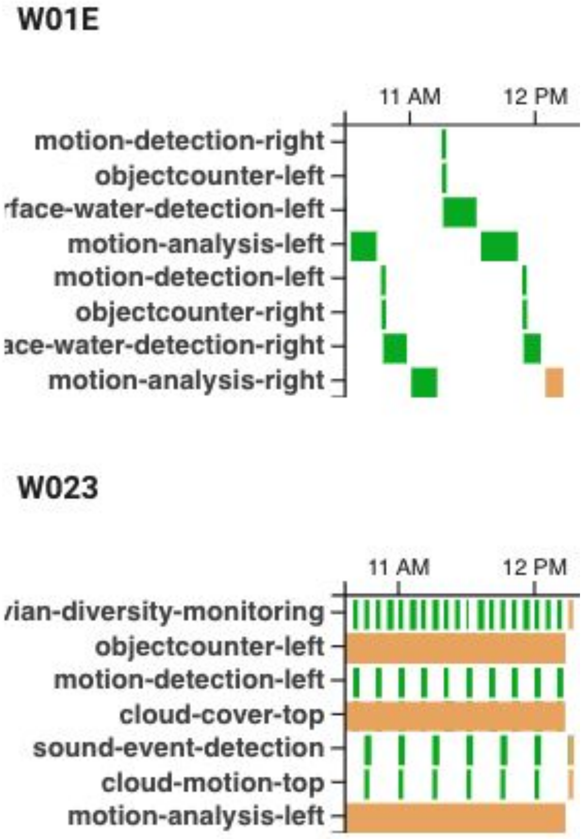
- Scheduling policies select the best plugins to run

> round-robin: selects the most starving plugin since its last execution

> run-all: selects all plugins triggered by science rules

> gpu-aware: selects GPU-demand plugins when there is no GPU-demand plugins running

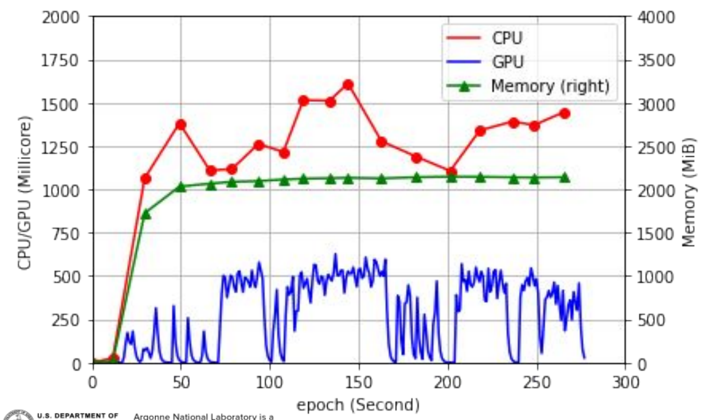
> and more!

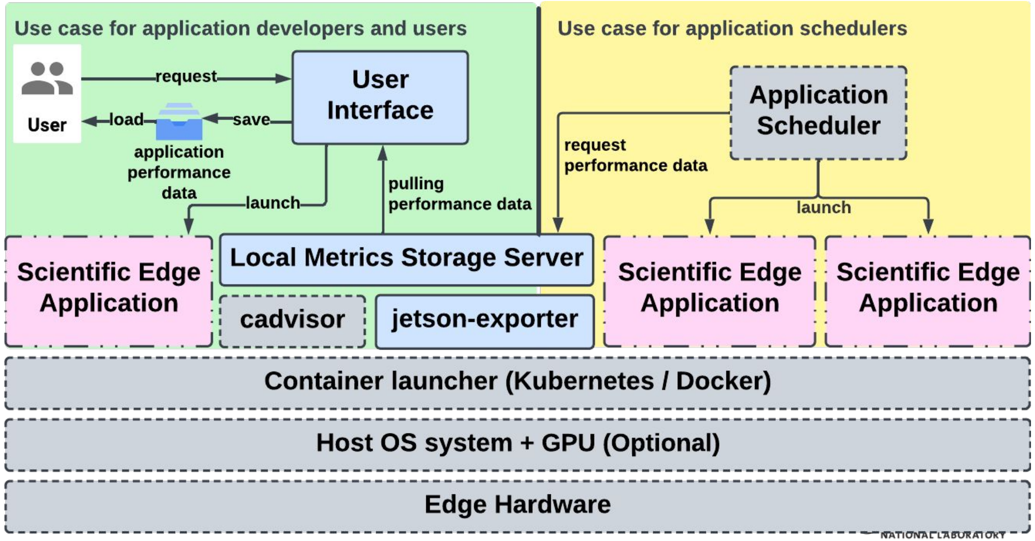


Job status on Sage nodes. The node (W01E) uses round-robin scheduling and the other (W023) is based on science rules-driven scheduling

Automated Vertical Scaling inside A Node

- Edge scheduler needs performance metrics of applications for resource-aware scheduling. Currently all applications are limited by 1000 millicore (1 CPU) and 1 GB memory
- Prototyping performance metrics storage server that constantly monitors performance of application containers
- Application developers can also use this service to ensure the application works on edge hardware




Use case for application developers and users

User

request

load

save

User Interface

application performance data

launch

pulling performance data

Use case for application schedulers

Application Scheduler

request performance data

launch

Scientific Edge Application

Local Metrics Storage Server

cadvisor

jetson-exporter

Scientific Edge Application

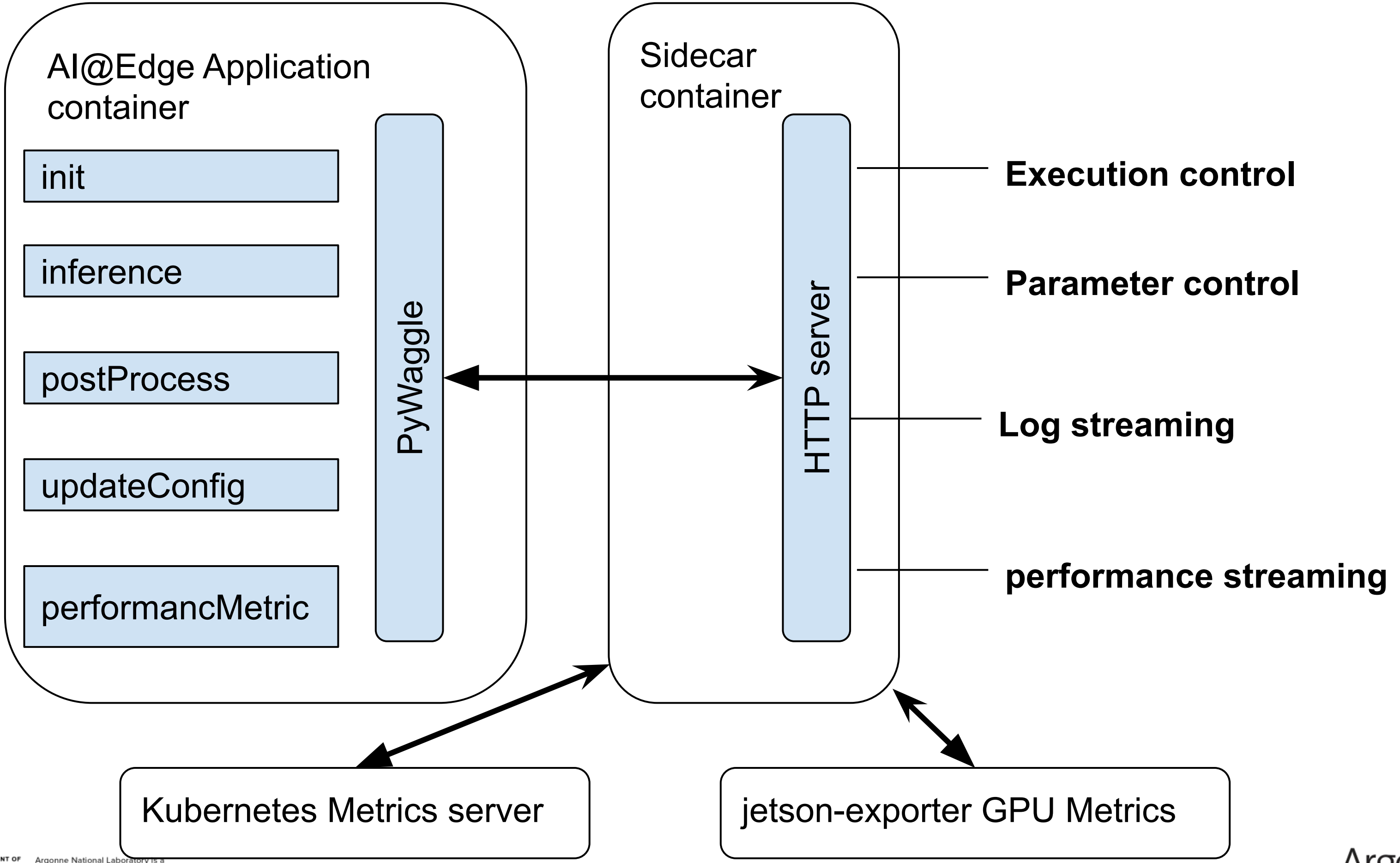
Scientific Edge Application

Container launcher (Kubernetes / Docker)

Host OS system + GPU (Optional)

Edge Hardware

Application Abstraction: Sidecar Approach to Control Applications



Horizontal Scaling of Jobs across Edge Nodes

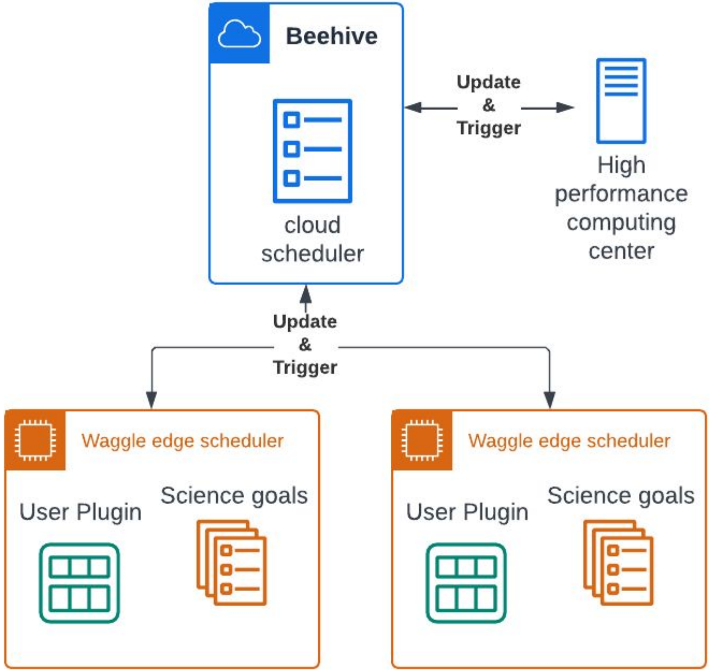
- Scientific workflows may specify quantitative requirements, e.g., data volume, number of application executions per node

```
---
name: water-detection
plugins:
- name: water-detector
  pluginSpec:
  image: registry.sagecontinuum.org/seonghapark/surface-water-detection:0.0.6
  selector:
    resource.gpu: true
nodeTags:
- WSN
- raingauge
- camera_bottom
scienceRules:
- "water-detector: rate('env.raingauge.total_acc') > 3 and cronjob('water-detector', '*/10 * * * *')"
successcriteria:
- DataQuantity(1000, 24h)
- WallClock(30d)
```

<< specifying desired quantity of measurement to decide scaling

Horizontal Scaling via Cloud and Edge interactions

- Edge schedulers interact with the cloud scheduler / HPC simulations to understand and drive itself towards the global goal
- Edge updates their local measurements
- Cloud triggers HPC simulations and updates science rules in the job
- HPC simulation may send predicted events to edge schedulers to change their scheduling behavior


Beehive
cloud scheduler
Update & Trigger
High performance computing center
Update & Trigger
Waggle edge scheduler
User Plugin
Science goals
Waggle edge scheduler
User Plugin
Science goals