KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

China 2024

Wasm for Portable AI Inference Across GPUs, CPUs, OS & Cloud-Native Environments

Hung-Ying Tai, WasmEdge
Miley Fu, WasmEdge

GitHub / Twitter:
@mileyfu / @mileyfu
@hydai / @hydai_tw

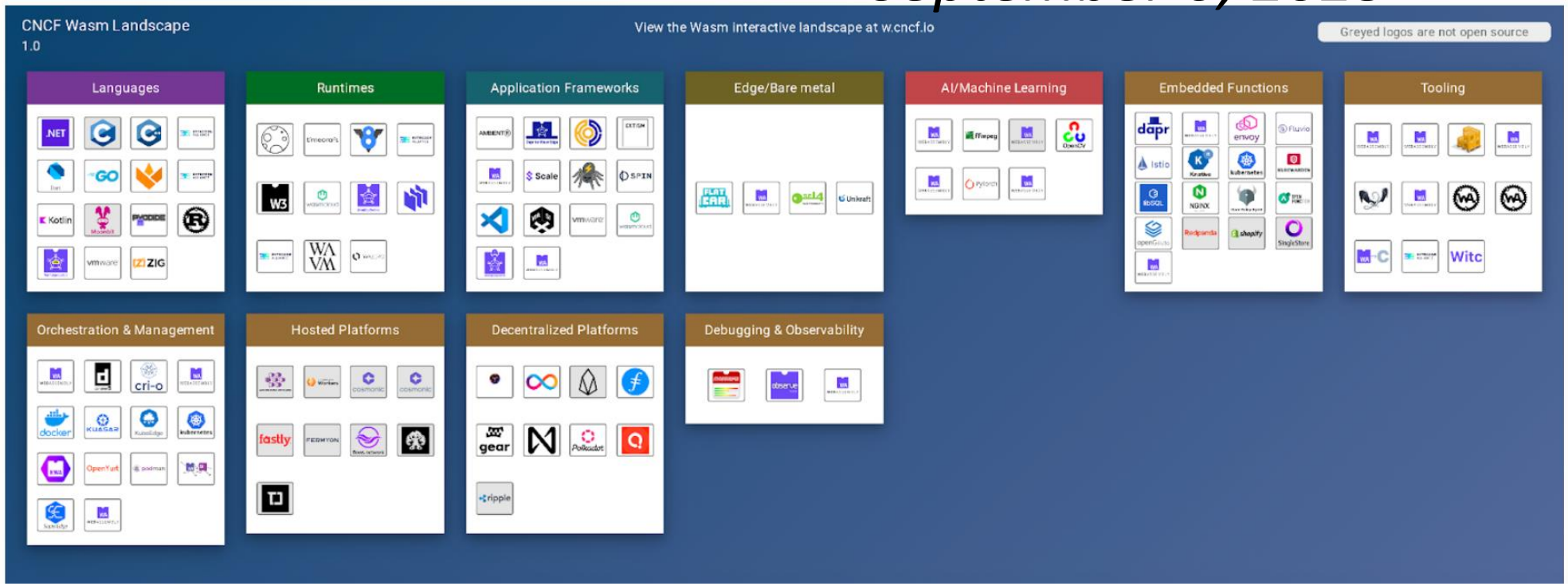
https://github.com/WasmEdge/WasmEdge

https://github.com/LlamaEdge/LlamaEdge

Wasm, on the server side?

The **initial Wasm landscape**, published in time for the WasmCon conference, includes 11 categories and 120 projects or products, representing $59.4B in total economic value. The Wasm landscape is divided into two large areas: Dev (application development) and Ops (application deployment).

Wasm landscape
September 6, 2023



Microservices

SaaS/UDF

Streaming data

Embedded

AI inference

Why Wasm for AI Inference?

- **Problem**: The need for consistent, efficient AI inference across diverse environments.
- **Key Points**:
  - Cross-platform consistency (run anywhere without rebuilding image targeting different underlying architecture or cuda version etc).
  - High performance and small footprint.
  - Security and sandboxing benefits.

Wasm in Cloud-Native Environments

- Integration with Kubernetes for scalable AI deployments.
- Use with Docker for containerized AI inference.
- Benefits for microservices and serverless architectures.

WasmEdge: A WebAssembly Runtime Optimized for AI

- Lightweight, high-performance WebAssembly runtime.
- Designed for AI, IoT, and edge computing.
- Compatibility with TensorFlow Lite, ONNX, and other AI frameworks.

LlamaEdge, AI tools built on top of WasmEdge runtime

a versatile AI deployment platform that enables running large language models (LLMs) across various environments, including edge devices, cloud infrastructure, and multiple platforms.

- across various hardware and OS.
  - CPU and GPU compatibility.
  - Execution on different operating systems (Windows, Linux, macOS).
  - Flexibility in deployment: from PCs to cloud environments.

Intel® Extension for Transformers

LLaMA C++

Tensor RT

arm

mlX

AMD

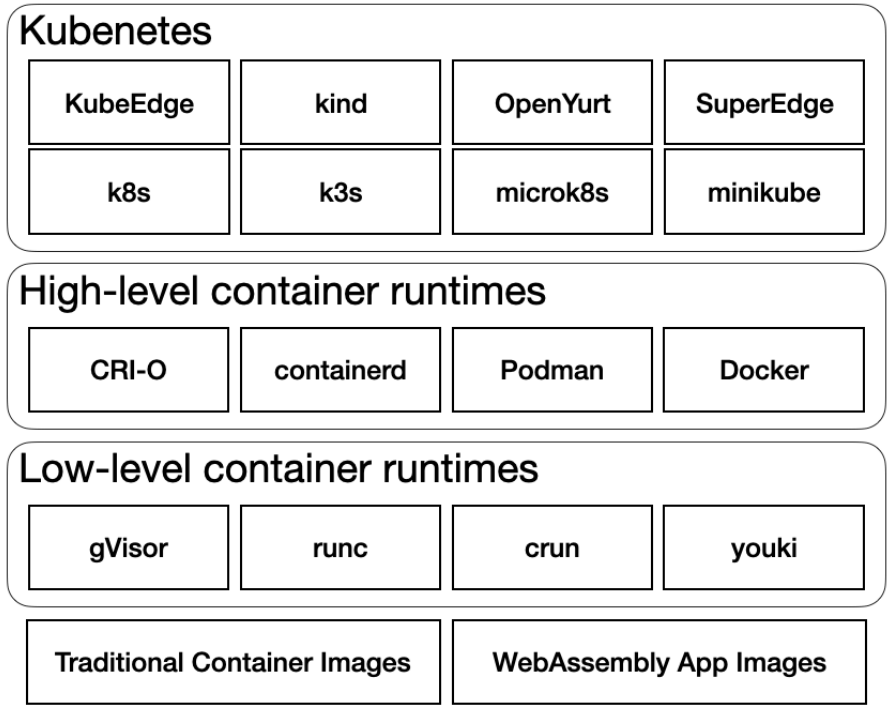
intel.

Key Features and Benefits

- **Lightweight:** The core runtime and API server are less than 30MB, making it suitable for resource-constrained devices.
- **Fast:** Takes full advantage of device hardware and software acceleration for optimal performance.
- **Cross-platform:** Enables LLM deployment across CPUs, GPUs, TPUs, NPUs etc.
- **Open-source:** Built on open standards, fostering community contributions and customization.
- **Privacy-focused:** Allows for local LLM execution, protecting sensitive data.

LlamaEdge in Action

- integrating Wasm with cloud-native AI workloads.
  - Combining Wasm with Kubernetes for scalable AI inference.
  - Leveraging Wasm containers in multi-cloud environments.
  - Using Wasm for serverless AI functions.



Demo Time!

- Setting up LlamaEdge API server.
- Packaging it as a Wasm container image.
- Running the image using Docker.
- Deploying in a Kubernetes cluster.

Demo 1: Running LlamaEdge API Server

- Build LlamaEdge API Server from source
  - Install Rust + wasm32-wasip1 target
    - `rustup target add wasm32-wasip1`
  - Compile it
    - `git clone git@github.com:LlamaEdge/LlamaEdge.git`
    - `cd LlamaEdge/api-server`
    - `cargo build --release --target wasm32-wasip1`
    - `# The built wasm is here`
    - `./target/wasm32-wasip1/release/llama-api-server.wasm`
- Or download the pre-built assets on the LlamaEdge repo
  - `curl -LO https://github.com/LlamaEdge/LlamaEdge/releases/latest/download/llama-api-server.wasm`

Demo 2: Run LLM using GPU in a container

- Workflows
- Podman/Docker
  - -> crun
  - -> libcrun-wasmedge
  - -> wasmedge-runtime
  - -> wasmedge-llm-plugin (based on llama.cpp)

Demo 2: Writing the Containerfile

- Pretty simple; developers and users are happiest.
- Just add the wasm file into container file

- `FROM scratch`
- `ADD llama-api-server.wasm /`
- `CMD ["/llama-api-server.wasm"]`

Demo 2: Pack and Publish images

- Still easy, still happy
- Pack
  - `[docker/podman] build . \`
  - `--platform wasip1/wasm \`
  - `-t hydai/kubecon2024:llama-api-server`
- Publish
  - `docker push hydai/kubecon2024:llama-api-server`

Demo 2: Run LLM using GPU in a container

- The complex one, no longer happy
- Set up the environment
  - Install CUDA Driver
  - Install NVIDIA Container Toolkit
  - Install CDI configuration
  - Install wasmedge libraries and LLM plugins
  - Build crun with wasmedge support from source
- Steps:
  - Podman: https://wasmedge.org/docs/develop/deploy/gpu/podman_wasm_gpu
  - Docker: https://wasmedge.org/docs/develop/deploy/gpu/docker_wasm_gpu

Demo 2: Run LLM using GPU in a container


```
# The first two lines are mapping WasmEdge plugins for enabling the Wasm+LLM support
# The cuda related lines are required if you want to use CUDA. Otherwise, it can run with CPU only.
# We also need to map the model folder to `/resource` because we don't want to package a whole model into a container image.
# There are some environment variables are needed for the safety issue. It will let `crun` know which plugins and models can be accessed.
Docker/podman run \
     -v ~/.wasmedge/plugin/libwasmedgePluginWasiNN.so:/.wasmedge/plugin/libwasmedgePluginWasiNN.so \
     -v /usr/local/cuda/targets/x86_64-linux/lib/libcudart.so.12:/lib/x86_64-linux-gnu/libcudart.so.12 \
     -v /usr/local/cuda/targets/x86_64-linux/lib/libcublas.so.12:/lib/x86_64-linux-gnu/libcublas.so.12 \
     -v /usr/local/cuda/targets/x86_64-linux/lib/libcublasLt.so.12:/lib/x86_64-linux-gnu/libcublasLt.so.12 \
     -v /lib/x86_64-linux-gnu/libcuda.so.1:/lib/x86_64-linux-gnu/libcuda.so.1 \
     -v /disk:/resource \
     --env WASMEDGE_PLUGIN_PATH=/.wasmedge/plugin \
     --env WASMEDGE_WASINN_PRELOAD=default:GGML:AUTO:/resource/Meta-Llama-3-8B-Instruct-Q5_K_M.gguf \
     -p 8080:8080 \
     --rm --device nvidia.com/gpu=all --runtime=crun \
      --annotation=module.wasm.image/variant=compat-smart --platform wasip1/wasm \
     hydai/kubecon2024:llama-api-server llama-api-server.wasm -p llama-3-chat
```

Demo 3: Run LLM using GPU in k8s

- Just like Demo 2, set up the environment again
- sudo ./kubernetes/cluster/kubectl.sh run -i --restart=Never testggml --image=ghcr.io/captainvincent/runwasi-demo:llama-api-server --annotations="module.wasm.image/variant=compat-smart" --overrides='
- {
- "apiVersion": "v1",
- "kind": "Pod",
- "metadata": {
- "name": "testggml"
- },
- "spec": {
- "hostNetwork": true,
- "containers": [..omitted..],
- "volumes": [..omitted..]
- }
- }'

Demo 3: Run LLM using GPU in k8s


```
"containers": [
  {
    "name": "simple",
    "image": "ghcr.io/captainvincent/runwasi-demo:llama-api-server",
    "command": ["/app.wasm", "-p", "llama-2-chat"],
    "stdin": true,
    "tty": true,
    "env": [
      {
        "name": "WASMEDGE_PLUGIN_PATH",
        "value": "/opt/containerd/lib"
      },
      {
        "name": "WASMEDGE_WASINN_PRELOAD",
        "value": "default:GGML:CPU:/resource/llama-2-7b-chat.Q5_K_M.gguf"
      }
    ],
    "volumeMounts": [
      {
        "name": "plugins",
        "mountPath": "/opt/containerd/lib"
      },
      {
        "name": "model",
        "mountPath": "/resource"
      }
    ]
  }
],
```

Demo 3: Run LLM using GPU in k8s


```
"volumes": [
      {
        "name": "plugins",
        "hostPath": {
          "path": "'"$HOME"'/.wasmedge/plugin/"
        }
      },
      {
        "name": "model",
        "hostPath": {
          "path": "'"$PWD"'"
        }
      }
    ]
```

Demo 3: Run LLM using GPU in k8s

- Output: https://github.com/second-state/wasmedge-containers-examples/actions/runs/10135790100/job/28023642175

Challenges and Future Directions

- **Overview:** Challenges
- **Key Points:**
  - Yong ecosystem
  - Low learning curve
  - More diverse infra to be supported

Calling contributors

help wanted

LFX Mentorship

Google Summer of Code

Filters is:issue is:open label:"LFX Mentorship" Lab

Clear current search query, filters, and sorts

26 Open 36 Closed Author Label Projec

- **LFX mentorship (Sept-Nov 2024): WASM Serializer with new proposals** enhancement LFX Mentorship
  #3585 opened 3 weeks ago by q82419
- **LFX mentorship (Sept-Nov 2024): Fix bugs found by fuzzer** enhancement LFX Mentorship
  #3584 opened 3 weeks ago by hydai
- **LFX Mentorship (Sept-Nov 2024): Create an LLM app with deep understanding of a GitHub repo** LFX Mentorship
  #3581 opened last month by juntao
- **LFX Mentorship (Sept-Nov 2024): Create a Wasm-based LLM app for financial analysts** LFX Mentorship
  #3580 opened last month by juntao

WasmEdgeRuntime

WasmEdge

https://github.com/WasmEdge/WasmEdge

LlamaEdge

https://llamaedge.com/

https://github.com/LlamaEdge/LlamaEdge

GaiaNet: The RAG API server and node

https://github.com/GaiaNet-AI

CLOUD NATIVE

COMPUTING FOUNDATION

YouTube

Bilibili

Stay in Touch!
Twitter/ Github
@mileyfu / @mileyfu
@hydai_tw / @hydai

LlamaEdge Github

WasmEdge Github