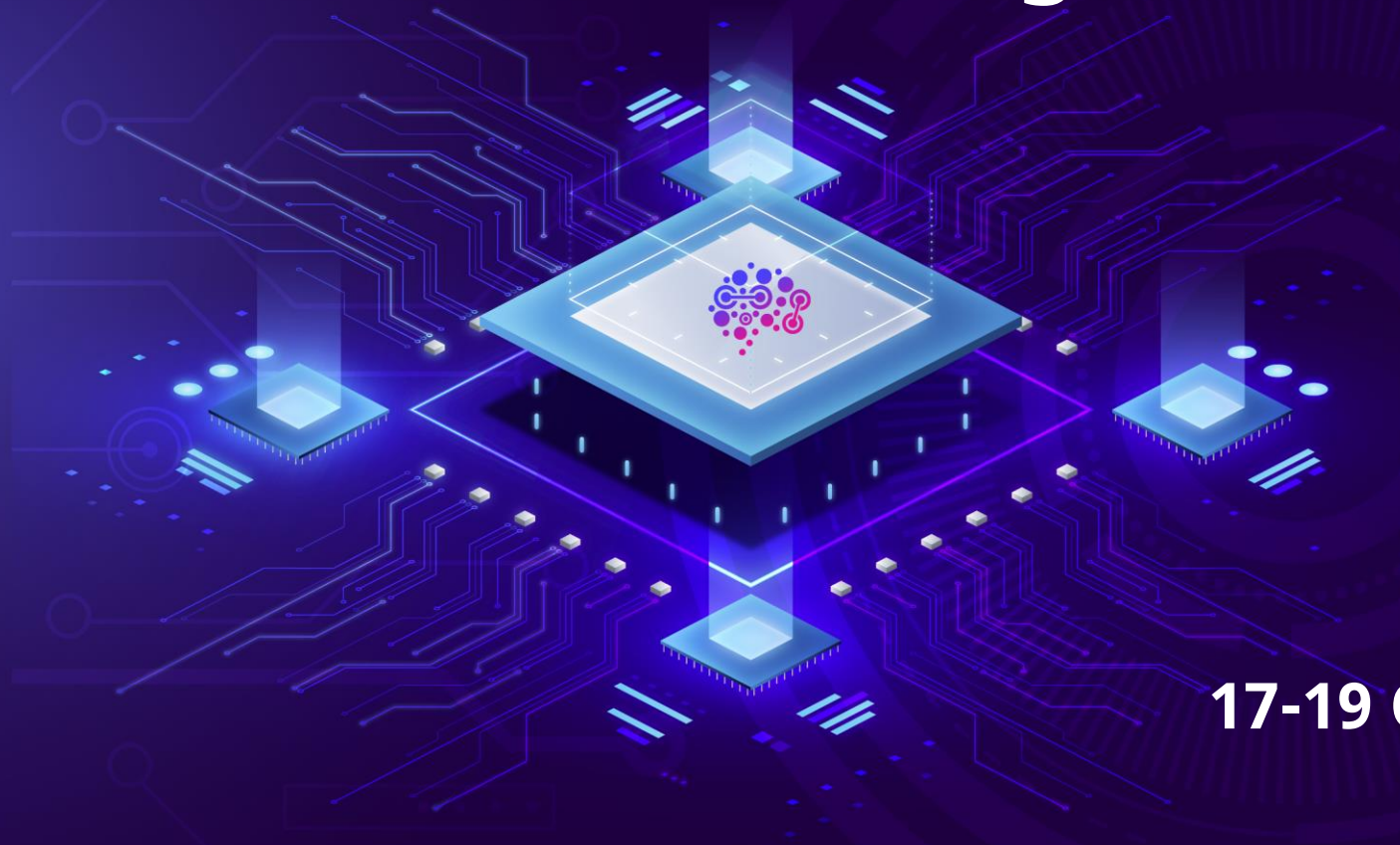
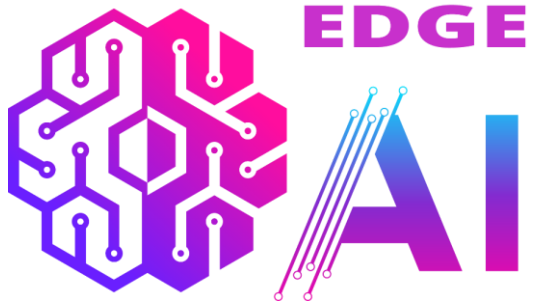
European Conference on EDGE AI Technologies and Applications - EEAI



Advancing emerging edge AI technologies and driving next generation intelligent applications.

17-19 October 2023, Athens, Greece

European Conference on EDGE AI Technologies and Applications - EEAI



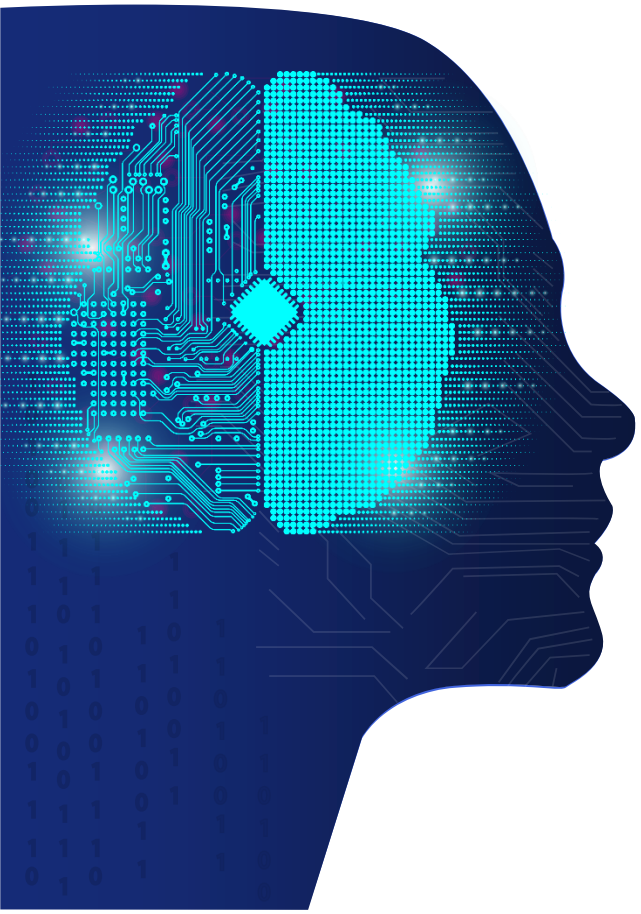
Why running Kubernetes on RISC-V is important for AI applications

Antony Chazapis, FORTH, Greece

17-19 October 2023 Athens, Greece

Copyright © 2023

Presentation Outline



- Introduction
- State-of-the-art
- Research Goal
- Methodology
- Design and Implementation
- Experimental Results
- Future Research
- Conclusion

Copyright © 2023

Introduction

Why RISC-V?[1]

- Flexibility → Customize, optimize, decide your own tradeoffs
- Control → No reliance on third parties
- Visibility → Security concerns
- (Cost)

Sovereignty and open hardware and software are important to the European Union. The European Commission is investing, via EuroHPC, about $300 million in RISC-V development.[2]

[1] https://www.qualcomm.com/news/onq/2023/09/what-is-risc-v-and-why-were-unlocking-its-potential
[2] https://www.intel.com/content/www/us/en/developer/articles/community/takeaways-from-the-first-risc-v-summit-europe.html

Copyright © 2023

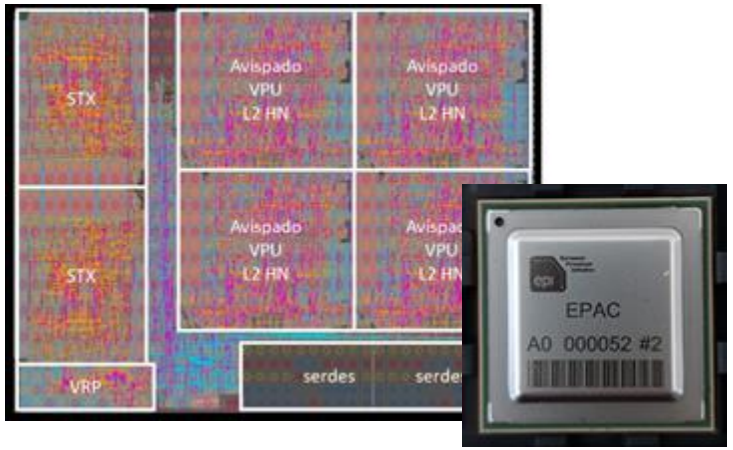
State-of-the-art

EPAC architecture includes RISC-V vector tiles (VTILE), specialized Deep Learning and Stencil accelerators (STX), and variable floating point precision cores (VRP) all carefully engineered in a heterogeneous tile architecture whose subunits comply to the RISC-V standardization efforts.[3]

(An incomplete list of) European projects

- European Processor Initiative (EPI) → EPAC (RISC-V accelerator)
- EUPILOT → HPC system with many VEC, MLS (EPAC derived)
- RISER → RISC-V Cloud server (H/W and S/W)
- eProcessor → New RISC-V CPU and S/W
- REBECCA → RISC-V Edge with AI acceleration

EPAC/VEC runs Linux!



[3] https://www.european-processor-initiative.eu/accelerator/

Copyright © 2023

Research Goal

Use Kubernetes to orchestrate AI app deployments on the Edge

- Implements a generic distributed system
  - Handles resource management, networking failures
  - Supports different architectures, including federations
- Defines the abstraction
  - Rich, extensible API
  - Portability of operations across platforms (with exceptions, as always)
    - Local (minikube, MikroK8s, k3s, kind, …)
    - Cloud (Amazon EKS, Azure AKS, Google GKE, DigitalOcean Kubernetes, …)
  - Inherent support for heterogeneity
- It is extensible
  - Core components are easy to customize or replace
  - Many extensions and third-party tools (workflows, …)

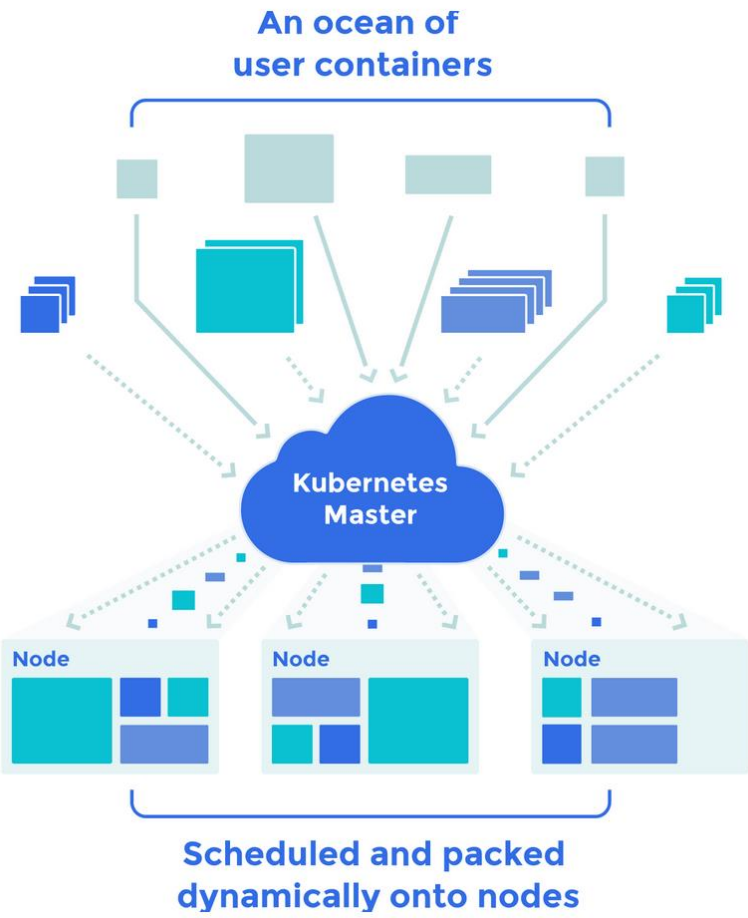


Image source: https://discuss.newrelic.com/t/relic-solution-what-you-need-to-know-about-new-relic-when-deploying-with-docker/52492

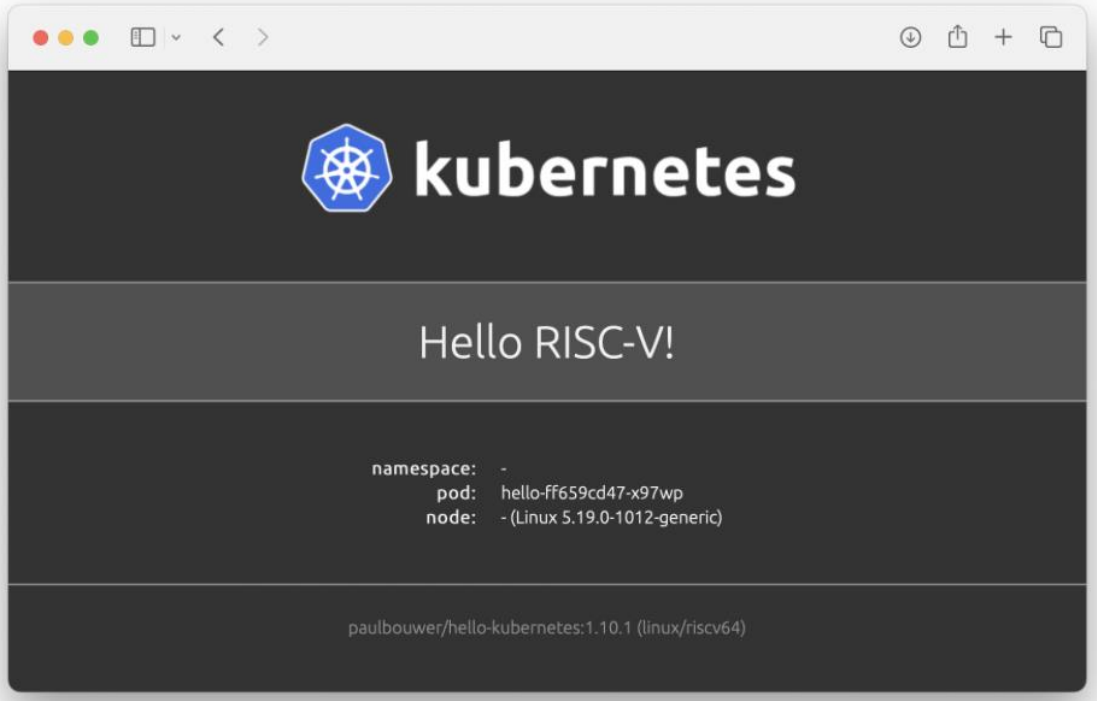
Copyright © 2023

Methodology

Port K3s to RISC-V → Light-weight Kubernetes distribution

- Single binary
- Smaller memory footprint

- Steps involved
  - Cross-compile code, supporting services
  - Take care of packaging and distribution → New containers
  - Send changes upstream
    - Currently 10 PRs upstream → 4 merged (3 CNCF, 1 Linux Foundation)



Copyright © 2023

Design and Implementation

Challenges

- Base containers (Debian, Ubuntu) still do not have a “stable” RISC-V variant → *However, there are “experimental” versions*
- No standardized build system → Custom scripts and methods for each project

However…

- “Experimental” RISC-V containers are available (i.e., riscv64/debian:unstable)
- Golang cross-compiles to RISC-V easily
- Docker supports building RISC-V containers with buildx

Copyright © 2023

Design and Implementation

1st phase

- K3s compilation
- k3s-root which includes supporting utilities for running K3s
- Backport RISC-V support in runc

2nd phase

- CoreDNS for internal service discovery
- Helm/klipper-helm used by K3s to install extras
- Local Path Provisioner for storage provisioning (extra)
- Traefik ingress service (extra)
- Klipper-lb for load balancing across services (extra)

3rd phase

- Workflow controller…
- Apps…

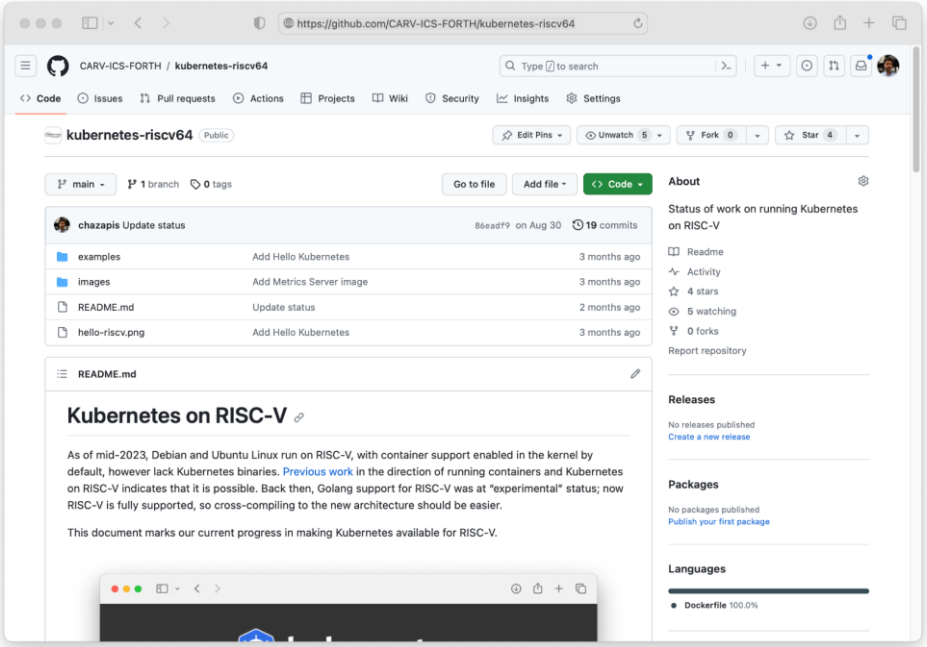
Copyright © 2023

Experimental Results

Current status

- K3s binary for RISC-V available → Tested on QEMU, real hardware
- Half PRs merged, released → Waiting for the rest…

Check https://github.com/CARV-ICS-FORTH/kubernetes-riscv64

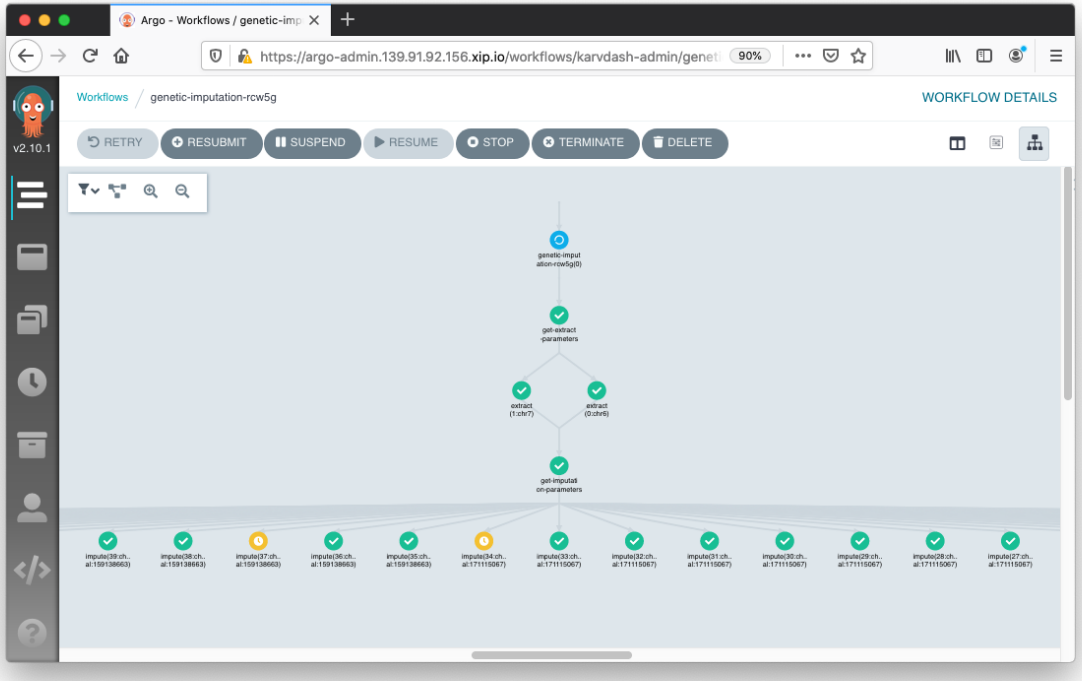


Copyright © 2023

Future Research

Argo Workflows

- Open source, container-native workflow engine
  - Easy, YAML-based language
  - Each step in the workflow is a container
  - Portable and cloud agnostic, workflows can run on any Kubernetes cluster (with heterogenous H/W)
- Workflow features
  - Integration with CI/CD pipelines
  - Cron-based scheduled execution
  - Mixed sequential/parallel execution
  - Retries on failures
  - Real-time state/log inspection
- Actively developed
  - CNCF hosted project
  - Many integrations with other projects



Copyright © 2023

Conclusion

- RISC-V is an emerging architecture → Suitable for the Edge
- We anticipate both stand-alone RISC-V devices and Linux-enabled RISC-V accelerators
- Kubernetes integrates heterogenous devices into a common, distributed H/W platform
- Argo provides the language and runtime for portable AI workflows

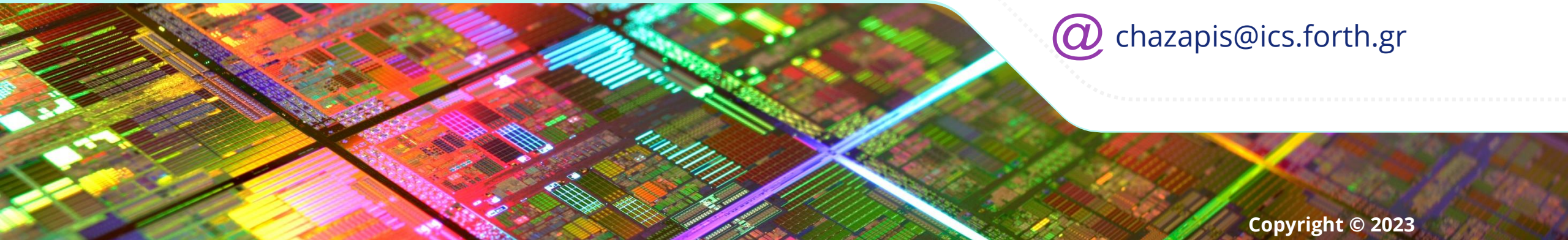
Work supported by:

Copyright © 2023

EDGE AI

Thank You

For your attention



@ chazapis@ics.forth.gr

Copyright © 2023