Kubernetes supports multiple architectures. Are you leveraging them?

Fabio Alessandro "Fale" Locati
EMEA Associate Principal Specialist Solutions Architect

Red Hat

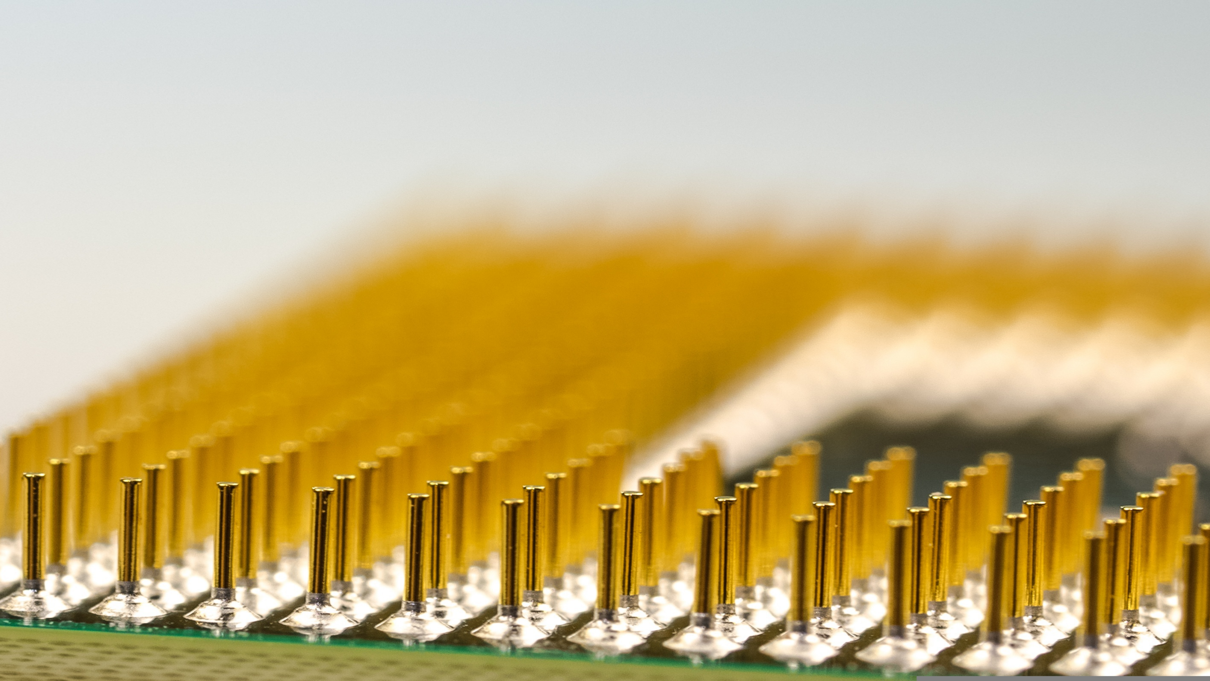
TOC

- Kubernetes and architectures
- Complex environments management
- Multi-arch images
- Conclusions

About me

- GNU/Linux user since 2001
- Kubernetes user since 2016
- Currently working for Red Hat

Kubernetes and architectures



Architectures

- Examples: x86, AMD64, ARM, AArch64, PowerPC, S390x
- We have so many architectures for historical reasons:
  - Product differentiation
  - Different philosophies (CISC vs RISC)
- Are they still relevant?
  - Yes, because they developed in **very** different ways
  - No, because our developers’ code (usually) is not architecture specific

Is Kubernetes multi-arch?

Short answer: **Yes!**

- Officially:
  - x86 and AMD64
  - ARM and AArch64
  - PowerPC
  - S390x
- Unofficially: MIPS, loong64, RISC-V, SPARC, etc.
- Mixed-archs

x86 and AMD64

- By large the most widely used architecture
- Foundable in all environments (edge, every public cloud, data-centers)
- Cores: 16-32 cores, max 100 cores @2-3GHz | CISC
- RAM: 128-512GB, max 12TB
- Features:
  - SMT2
  - Extensions (AES, Video encoding/decoding, ...)

ARM and AArch64

- Very common in smaller systems, are starting to become common in bigger environments
- Foundable in most environments (edge, every public cloud)
- Cores: 4-8, max 256 cores @2-3GHz | RISC
- RAM: 2-16GB, max 8TB
- Features:
  - Usually no SMT
  - External hardware accellerators
- Low(er) cost and performance hardware

PowerPC

- Very good architecture for vertical scaling and low power consumption
- Foundable in data-centers and most clouds (Azure, Google Cloud, IBM Cloud)
- Cores: 24-48 cores, max 240 cores @3-4GHz | RISC
- RAM: 1-8TB, max: 64TB
- Features:
  - SMT8
  - Full encryption of data in transit between components with no performance hit
- Higher performance per Watt

S390x

- Very good architecture for vertical scaling and low power consumption
- Foundable in data-centers and IBM Cloud
- Cores: 39-82 cores, max 200 cores @5.2GHz | CISC
- RAM: 10-20TB, max 40TB
- Features:
  - Huge CPU Caches (L2: 32MB, L3: 256MB, L4: 2GB)
  - Full encryption of data in transit between components with no performance hit
  - Internally redundant
  - Scale-out to multiple systems (up to 10km apart)
- Extensions and accellerators (AES, Quantum Encryptions, ...)
- Direct-networking with no additional cost
- Up to 80% more efficient
- Often containing the System of Record

My rules of thumb

- Do not try to use all architecture just because, focus on the ones that give you benefits
- Aarch64 for “edge” cases, or highly horizontal-scalable or stateless workloads
- PowerPC for high density and highly vertically-scalable workloads
- S390x for workloads that require very high resiliency or data-colocation
- AMD64 for all other workloads (at least for now)

Complex environments management

Multi-cluster

- Have many clusters with a central Cluster Management system
- Suggested option: Open Cluster Management
- Advantages:
  - Easily reuse known architectures and distributions
- Disadvantages:
  - More systems to manage

Mixed-architectures clusters

- Have few clusters with mixed-architectures nodes
- Advantages:
  - Less clusters to manage
- Disadvantages:
  - Probably still needing a Cluster Management system
  - Potentially unsupported by your vendor
- Ideas:
  - *etcds* on S390x?
  - databases on PowerPC?

Multi-arch images

Multi-arch images

- Store images
- Build images

Store images

- Majority of Container Registry support multi-arch images by default

Build images

- Leverage the build tool you are already using!

docker manifest

```
docker build -t fale/hello:manifest-amd64 --build-arg ARCH=amd64/ .
docker push fale/hello:manifest-amd64
docker build -t fale/hello:manifest-arm64v8 --build-arg ARCH=arm64v8/ .
docker push fale/hello:manifest-arm64v8
docker build -t fale/hello:manifest-ppc64le --build-arg ARCH=ppc64le/ .
docker push fale/hello:manifest-ppc64le
docker manifest create fale-hello:manifest-latest \
    --amend fale/hello:manifest-amd64 \
    --amend fale/hello:manifest-arm64v8 \
    --amend fale/hello:manifest-ppc64le
docker manifest push fale/hello:manifest-latest
```

buildah

```
buildah manifest create manifest-latest
buildah bud --tag "ghcr.io/fale/hello:latest" \
    --manifest manifest-latest --arch amd64 .
buildah bud --tag "ghcr.io/fale/hello:latest" \
    --manifest manifest-latest --arch arm64v8 .
buildah bud --tag "ghcr.io/fale/hello:latest" \
    --manifest manifest-latest --arch ppc64le .
buildah manifest push --all manifest-latest \
    "docker://ghcr.io/fale/hello:latest"
```

Build images

- Leverage the build tool you are already using!
- Use automation
- Use docker buildx

docker/buildx

```
docker buildx build \
    --platform linux/amd64,linux/arm64v8,linux/ppc64le \
    --tag ghcr.io/fale/hello \
    --push backend .
```

Conclusions

Wrapping up

- Evaluate the various architectures with no preassumptions
- Design your architecture to accommodate multiple architectures
- Build all your images for all reasonable architectures
- Have fun :)

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat