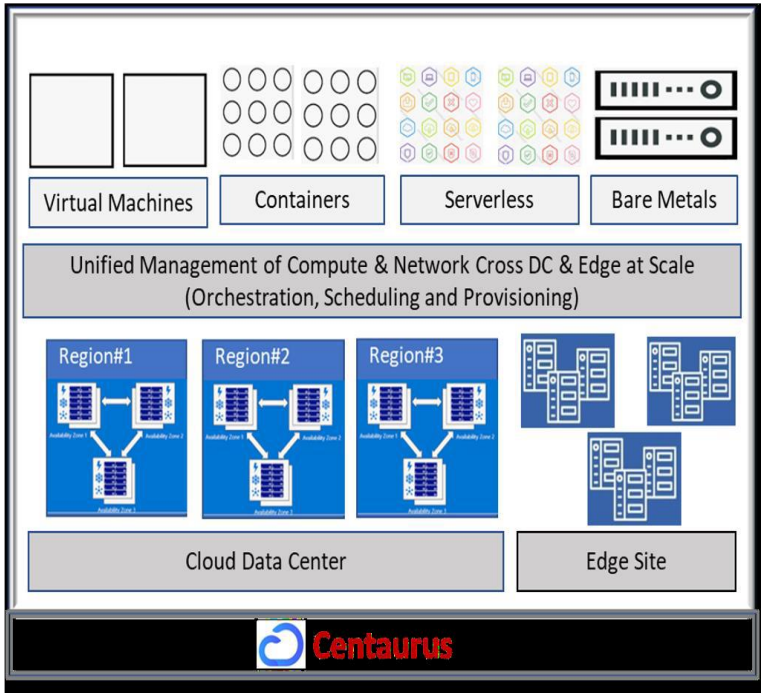
Centaurus Distributed Cloud Infrastructure

A Deep-Dive

Futurewei Cloud Lab.

Centaurus Distributed Cloud Infrastructure Overview

- Project **Centaurus** is an open source (LF) platform targeted towards building unified and highly scalable public or private **distributed** cloud infrastructure.
- Aims to meet the challenges for new types of workloads such as AI and 5G applications landscape.
- Offers enterprises the hyper-scaler like capabilities that dramatically changes the economics of enterprise IT.
- Key underlying technology pillars of Centaurus project:
  - **Arktos** – a large scale cloud compute
  - **Mizar** – high scale and high performant cloud networking
  - **Fornax** – Autonomous and flexible edge computing
  - **Alnair** – Intelligent platform for AI workloads

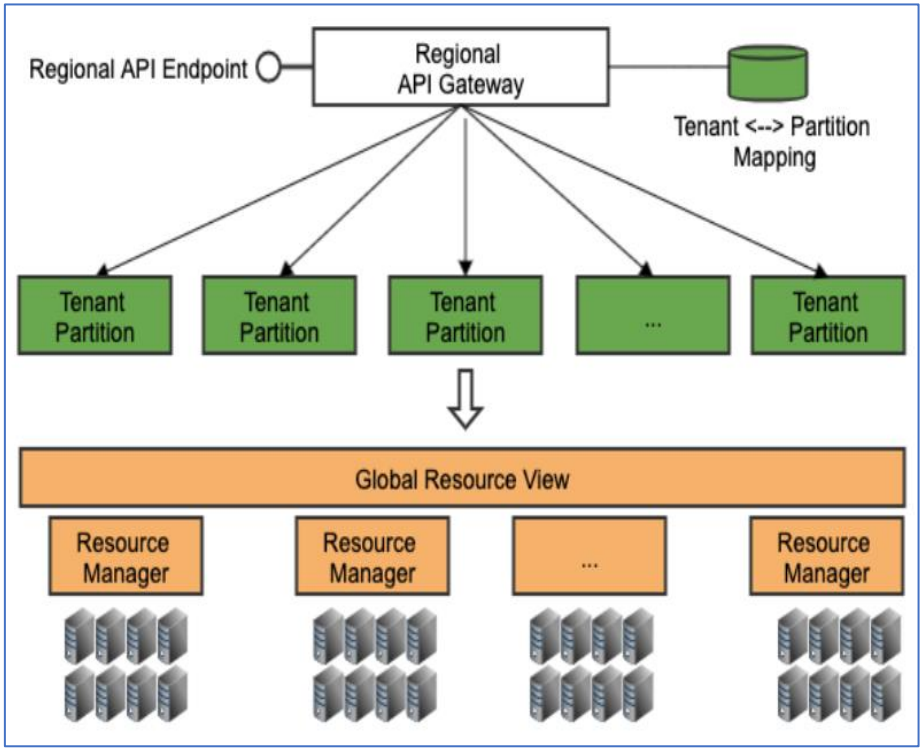


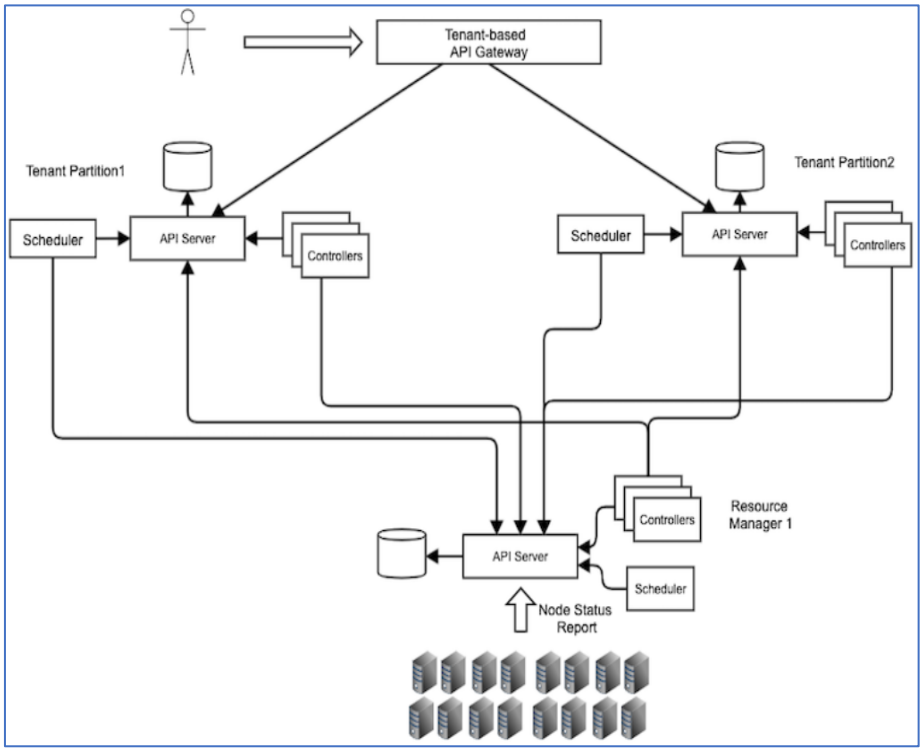
Arktos Compute

Arktos Compute Layer Overview

- Arktos is an open source project designed for large-scale cloud infrastructure.
- Arktos was evolved from the Kubernetes codebase and features a lot of similar API objects — like pods and replica sets.
- Arktos introduces core design changes in order to enable the following key features:
  - Unified cloud infrastructure resource support
  - High throughput and low latency
  - Multi-tenancy support

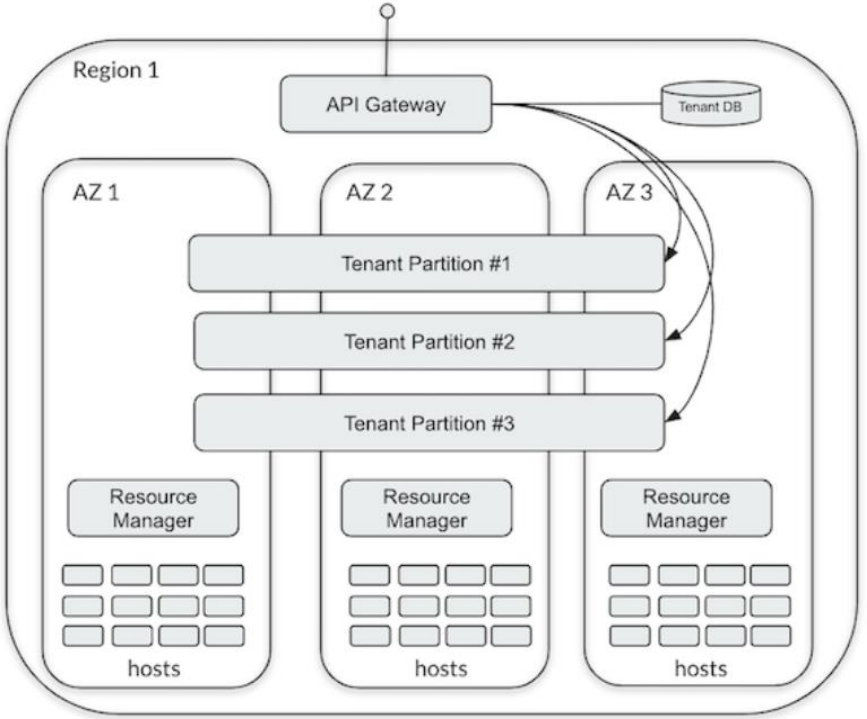
Arktos Architectural Overview





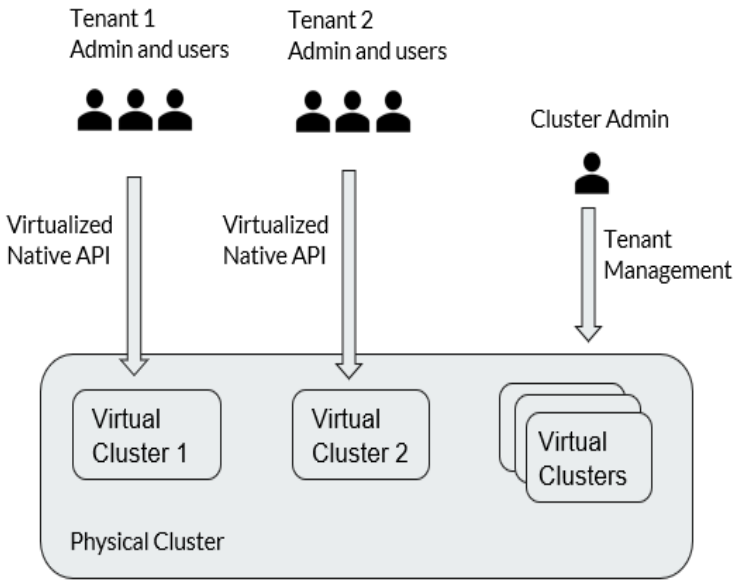
Arktos Hyper-Scaler Cloud Scalability

- Public cloud level scalability — it aims to support 300,000 hosts per region and 100,000 hosts per cluster.
- All the control plane components can scale-out and are highly available — tenant workloads are partitioned



Deployment View

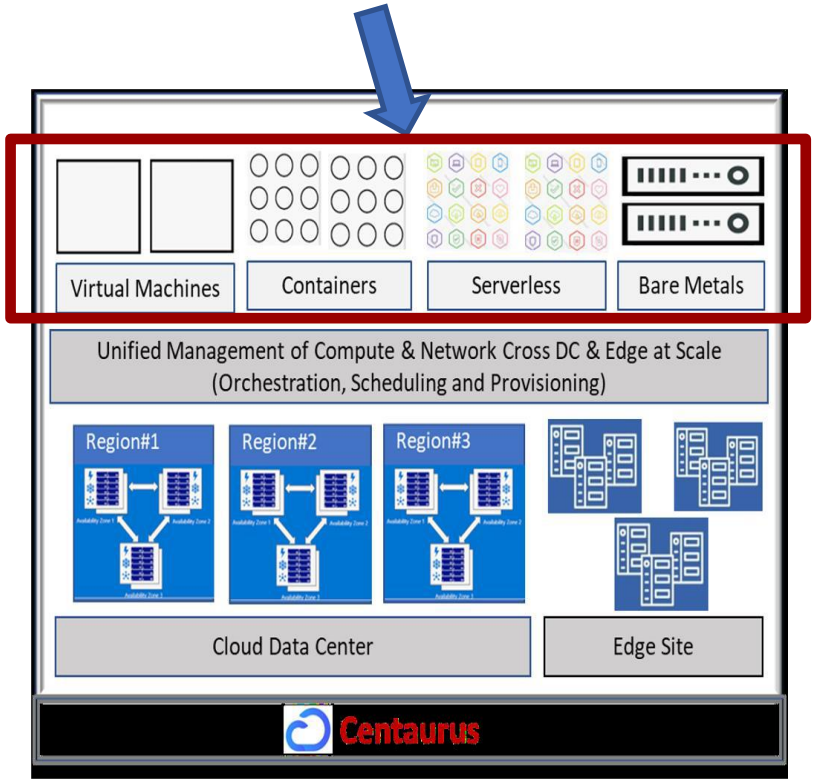
Arktos Multi-Tenancy



- Vision: enable multi organizations to safely and easily share same physical cluster infra.
- Design Principles
  - Isolation: each tenant has its own resource view without awaring of each other.
  - Autonomy: tenant manages its own resources and policies without turning to cluster admins.
  - Compatibility: tenants can still use existing APIs and tools.
  - Manageability: cluster admins can perform cross-tenant management tasks.

Arktos Unified Runtime Orchestration

- Contemporary fragmented orchestration stacks for containers and VMs introduces resource pool in-efficiencies, duplicated components, increased maintenance and operational cost.
- Arktos introduces native support of VM, in addition to the mature container support inherited from Kubernetes — a unified resource pool.



Mizar Networking

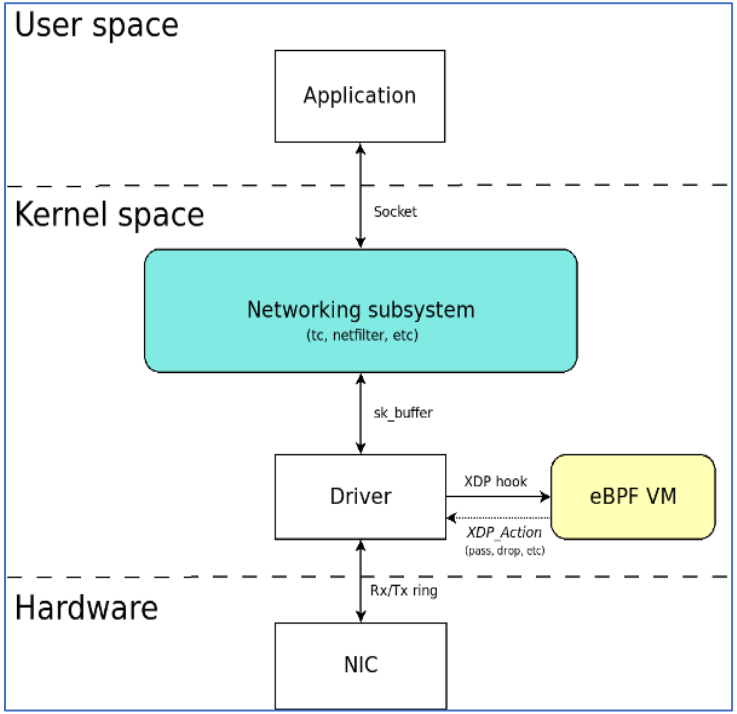
Mizar: Problems with programmers thinking in flow-rules

- Current flow-based programming solutions are not scalable and have a multitude of issues and quirks.
- Time to provision ports increases significantly as the number of ports increases.
- High CPU utilization during flow-parsing.
- Packets traverse multiple network stacks on the same host.
- Provisioning time of a new workload depends on the number of workloads already existing in the system.

Mizar Networking Layer – XDP

Enter eXpress Data Path (XDP) – A Linux Kernel Superpower

- Safely and Dynamically modify the NIC device driver behavior without packet processing interruption
- Process Packets before delivering it to the stack
  - PASS, TX, REDIRECT, DROP
- API interfaces that programmers understand!
- Does not require dedicated CPUs and Off-loadable to SmartNICs
- Small programs 4K ebpf instructions!



The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel

Toke Høiland-Jørgensen
Karlstad University
toke@toke.dk

Jesper Dangaard Brouer
Red Hat
brouer@redhat.com

Daniel Borkmann
Cilium.io
daniel@cilium.io

John Fastabend
Cilium.io
john@cilium.io

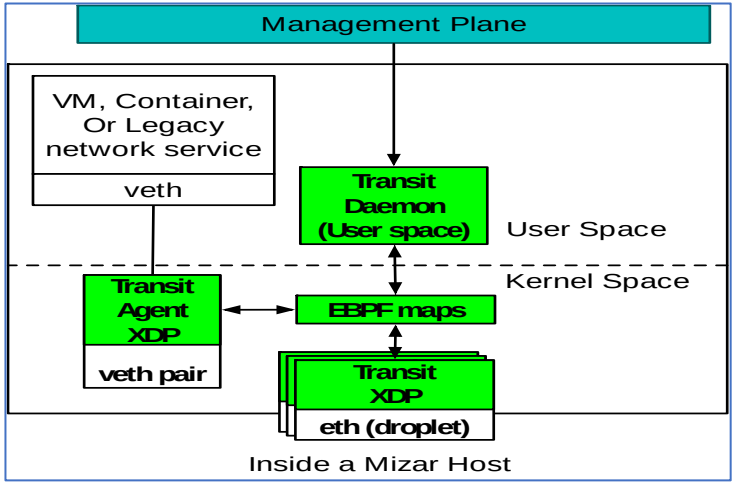
Tom Herbert
Quantonium Inc.
tom@herbertland.com

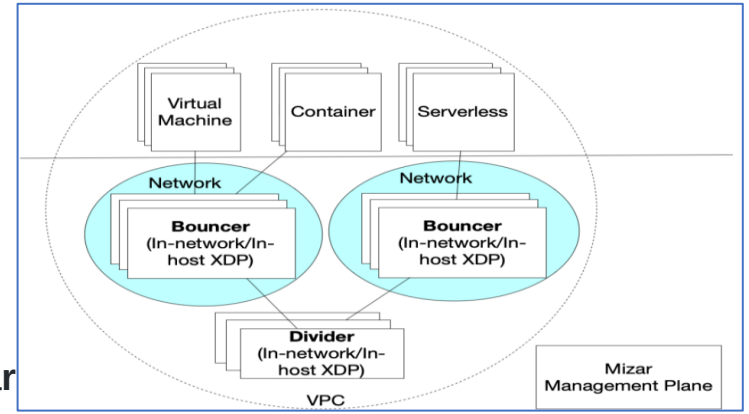
David Ahern
Cumulus Networks
dsahern@gmail.com

David Miller
Red Hat
davem@redhat.com

Mizar Networking Architecture

- One XDP Program attached to NIC
  - Processes all ingress packets
- One XDP program attached to the veth-pair of a container
  - Process egress packets from that container
- Expose RPC interface to the management plane
  - Load/Unload the XDP programs
  - Push any form of configuration to ebpf maps





Logical Architecture of Mizar

Mizar Networking Layer – a summary

- The flow-programming model is great for programmable switches but not scalable for multi-tenant cloud networks
- Tremendous Provisioning throughput & Run-time CPU/Memory performance gains
- Create an extensible plugin framework for cloud networking
- Unify the network data plane for VMs, Containers, Serverless and other workload types
- Label-based Network Policy enforcement
- Programming the SmartNICs with small, safe, and dynamically loadable programs enable the management-plane to even higher scale overlay networks

Fornax Edge Computing

Fornax Edge Computing – an Overview

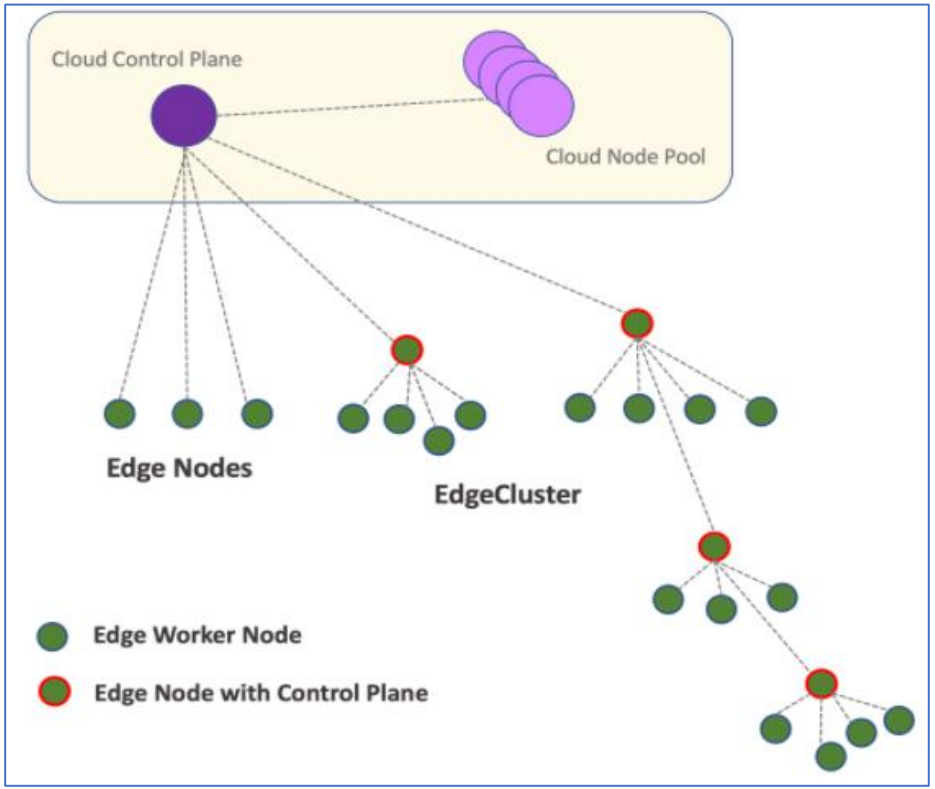
- Fornax is an open source edge-computing framework for managing compute resources on the edge environment.
- Fornax is designed to solve some of the key edge computing challenges such as limited computing resources, heterogeneous resource types, layered topology, unreliable network, and long latency.
- With Fornax, end-user's edge application workloads could be easily deployed in a distributed hierarchical edge environment with topologies that best matches the physical and logical structure.
- Fornax also offers high performance virtualized networking for workload communication within and between edge clusters.

Fornax Edge Computing – Key Features

- **Computing nodes and clusters on the edge**: Both computing nodes and full-fledged clusters can run on the edge.
- **Hierarchical topology**: Edge clusters can be structured in multi-layer tree-like topologies, providing best mapping to end-user scenarios.
- **Flexible flavors**: Supports multiple flavors of clusters on the edge, e.g. Arktos, K8s and K3s.
- **Edge networking**: Multi-tenant edge cluster networking (Supporting concepts like VPC, Subnet) and high performance inter-cluster communication.

Fornax Edge Computing – Design Overview

- Fornax models edge as an m-ary tree where an Arktos control plane sits at the root of the tree in the cloud, and leaf tree nodes represent computing nodes on the edge.
- The sub-trees in the m-ary tree are standalone clusters, and the roots of the sub-trees are control planes for edge clusters.
- As usual with Arktos clusters, there are also compute nodes in the cloud managed by the root level Arktos control plane.



Alnair AI

Alnair Vision

- Building an intelligent platform to improve AI workloads efficiency.
- AI workloads will be the critical/dominant workloads for cloud and edge computing.
- Current cloud/edge systems leverage existing hardware/software architecture to support new AI workloads, which limits the capability of AI training/inferencing and also increases the model serving cost.
- More efficient and more intelligent hardware/software frameworks and architectures are needed to support AI workloads.
- Focus on the resources management aspects, to analyze and schedule AI workloads on existing/new systems, with intelligent methods.
- We also explore new architecture to orchestrate heterogenous resources, and new service model to facilitate AI workloads.

Alnair – Key Features

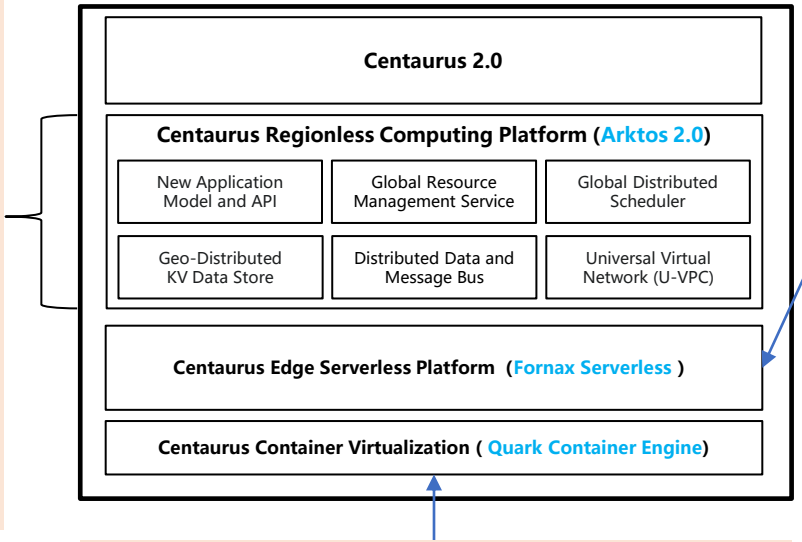
- Elastic platform with self-learning capability
  - Elastic training, dynamic GPU allocation
  - GPU utilization profiling, precise resource management
  - GPU fine-grained sharing, optimized resource utilization
  - Autonomous scheduler, continuous scheduling decision learning, policy improvement
- Optimized ML framework
  - Parallelism (data/model/pipeline) Optimization
  - Hyperparameters auto tuning

Transitioning to Centaurus 2.0 (Distributed Cloud Infrastructure)

2022/23 Cloud Compute Project – Centaurus 2.0

Centaurus Regionless Computing
Goals & challenges

- New model to use cloud – from resource-based to application-based cloud (cloud native 2.0)
- Manage **2 million+** compute nodes from big, small & edge data centers as global resources
- Distributed scheduling algorithm and scheduler architecture to scale to **10K RPS** throughput.
- Design geo-distributed data consistent KV store to manage **100 millions** application instances
- Large scale virtual network (VPC) to provision & manage **1 million+** application/vm instances



Centaurus Edge & Serverless Computing
Goals & challenges

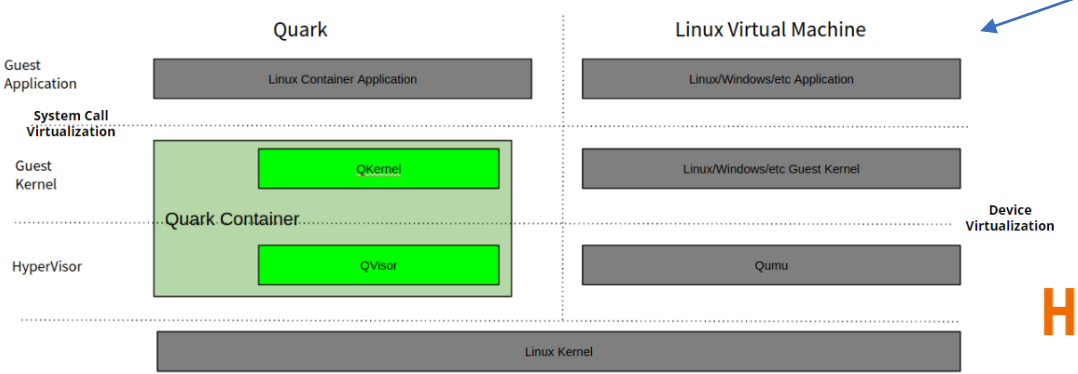
- Extreme low latency (<**100 ms**) for starting application instance at edge & auto scale to handle burst requests
- The platform itself must be very Lightweight – use minimum resources (less than **10GM/3CPU** to manage **5K** application instances
- Multi-tenant edge computing clusters with strong computing/networking isolation.
- High performance scheduling algorithm (<**10ms**) to allocate application instance onto a node

Centaurus Container Runtime Engine
Goals & challenges

- High performance and secure container runtime – **3X** Kata & gVisor in RPS & Throughput
- Light weighted container – memory overhead **1/3** of gVisor and **1/15** of Kata container
- **RDMA** based network communication & **NVMe/NVMeOF** based direct device access – **30%** Performance Gain

Centaurus 2.0 – Quark Secure Container (V1.0)

Architecture (System Call Virtualization vs Device Virtualization)



Centaurus Container Runtime Engine

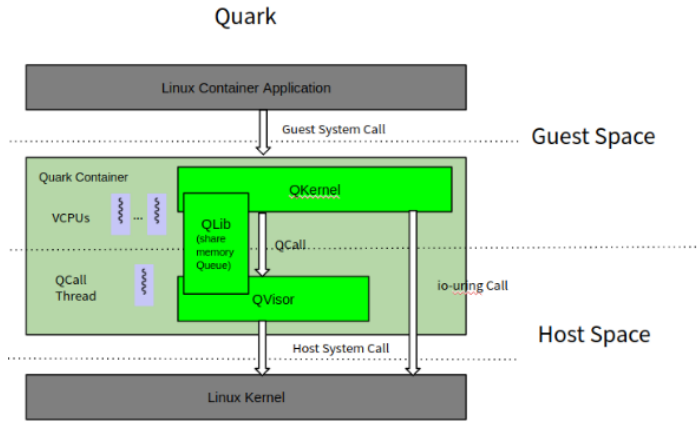
Goals & challenges

- High performance and secure container runtime – **3X** Kata & gVisor in RPS & Throughput
- Light weighted container – memory overhead **1/3** of gVisor and **1/15** of Kata container
- **RDMA** based network communication & **NVMe/NVMeOF** based direct device access – **30%** Performance Gain

High level design

High Level Design Points

- System Call virtualization – Reimplement 80% system calls
- QCall: Share memory-based communication between QKernel and QVisor
- IO-Uring: IO data plane between QKernel and host Kernel



Cloud AI – Alnair Platform

ALNAIR

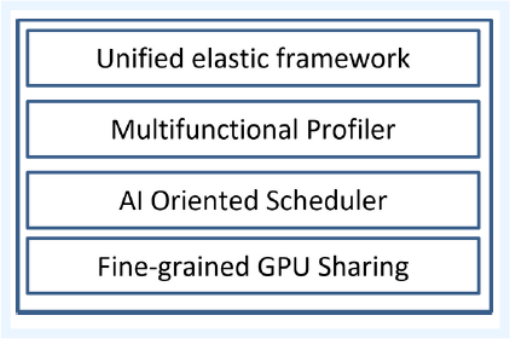
Intelligent platform for AI workloads

Elastic
Self-learning
Training and serving efficiency
Monitoring and logging

Project Background

- Serving AI workloads is one of the most important missions for next generation cloud platform
- Cloud platform needs to be tailored based on the special characteristics of AI workloads, e.g., parallel computation and heavy data ingestion in training, low latency in inference
- AI platform touches various domain, e.g., hardware accelerators, data storage/pipeline, resource management, ML framework, etc.
- This year focus on (platform building blocks, small and medium size training jobs)
  - **GPU sharing and profiling**
  - **Intelligent scheduling**
  - **Data orchestration / cache for AI jobs**

Alnair Architecture



Project Milestones

1. vGPU resource management
   - Device Plugin
   - Manager
   - Scheduler
2. Resource utilization per job profiling & storing

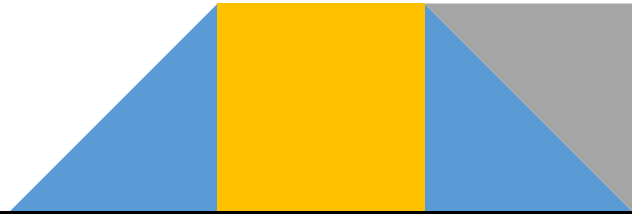
1. Local AI storage with distributed caching
2. CUDA-level exporter to profile workload metrics with no code changes
3. Secure container runtime with GPU support
4. 3D reconstruction use case

1. Data orchestrator, efficient data caching, scheduling policy
2. Intelligent scheduler with history learning, and minimized rescheduling cost
3. Profiler report data related metrics download/preprocess…
4. RDMA enabled container runtime to demo network acceleration

1. Data orchestrator, data loader optimization
2. Integration Test, documents
3. Cloud AI platform trends and insights report

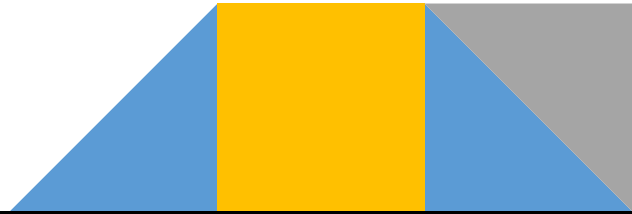
Who We Are

- A small group of people elected from member groups and projects
- The governing body that oversees Centaurus project execution from technical perspective
- Operating under the TSC Charter from Linux Foundation
- Currently 7 TSC members
- We also have:
  - Advisory board
  - Sub-committee of marketing and outreaching



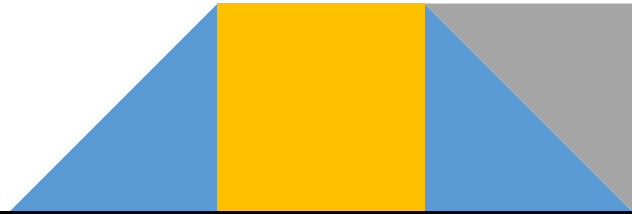
What We Do

- Coordinating the technical direction of the projects from the four Special Interest Groups (SIGs)
- Approving sub-projects and removing sub-projects
- Cross-project technical issues and requirements
- Establishing community norms, workflows and technical policies
- Coordinating marketing, events, or external communications



How We Execute

- Principles: Open, public and easily accessible
- TSC meets regularly on the last Tuesday of the month. TSC meetings are open and public, everyone can dial in. (but only TSC members can vote)
- Topics are proposed before a meeting in a public document
- Public email groups and slack channels are used for offline discussions
- All TSC decisions, meeting notes and presented material are publicly accessible to everyone



Resources

- TSC Repo: https://github.com/CentaurusInfra/tsc
- TSC Email Groups: centaurus-tsc@googlegroups.com
- TSC Meeting Notes: https://docs.google.com/document/d/1nfGJ_9nudQWjbEx2f21_kkf15quuw7fOQaqF_EtQc-I/edit?usp=sharing

Welcome to Join us!