Introducing DECICE

Heterogeneous Compute Continuum

Jonathan Decker
jonathan.decker@uni-goettingen.de

October 19, 2022

hpc@gwdg.de
GWDG – Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

About Us

Representing

About Us

Representing

- High-Performance Storage (HPS) Research Group
  - Georg-August-Universität Göttingen
  - Lead by Prof. Dr. Julian Kunkel
  - `https://hps.vi4io.org`

About Us

Representing

- High-Performance Storage (HPS) Research Group
  - Georg-August-Universität Göttingen
  - Lead by Prof. Dr. Julian Kunkel
  - `https://hps.vi4io.org`

- GWDG
  - Prof. Kunkel is Deputy head GWDG
  - Leads AG Computing with 25+ HPC experts
  - NHR Center (Tier 2) - former HLRN
  - Data center and IT-competence center

The DECICE Project

Project Goals

Addressing the Challenges

Conclusion

The DECICE Project

DECICE Project

Device-**E**dge-**C**loud **I**ntelligent **C**ollaboration fram**E**work

- EU Horizon project Start Q4 2022 (6 M € financing)
- Development of a **Cognitive Cloud**

DECICE Project

Device-**E**dge-**C**loud **I**ntelligent **C**ollaboration fram**E**work

- EU Horizon project Start Q4 2022 (6 M € financing)
- Development of a **Cognitive Cloud**
- AI-enabled computing continuum from Cloud to Edge
- Cooperation between
  - Universities
  - Data centers
  - Research institutes
  - Industry companies

13 DECICE Partner

HR@GÖTTINGEN

DECICE
Device-Edge-Cloud
Intelligent Collaboration framEwork

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
IN PUBLICA COMMODA SEIT 1737

GWDG
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

E4
COMPUTER ENGINEERING

KTH
VETENSKAP OCH KONST

University of Stuttgart
Germany

HUAWEI

FORSCHUNG Burgenland
RESEARCH & INNOVATION

SYNYO

MARMARA ÜNİVERSİTESİ
1883

top-ix
WE DO CONNECTIONS

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
A.D. 1088

nag

TRI

Project Goals

Problem Statement

- Cloud computing has become major factor
  - Improved productivity
  - Scalability via public clouds

Problem Statement

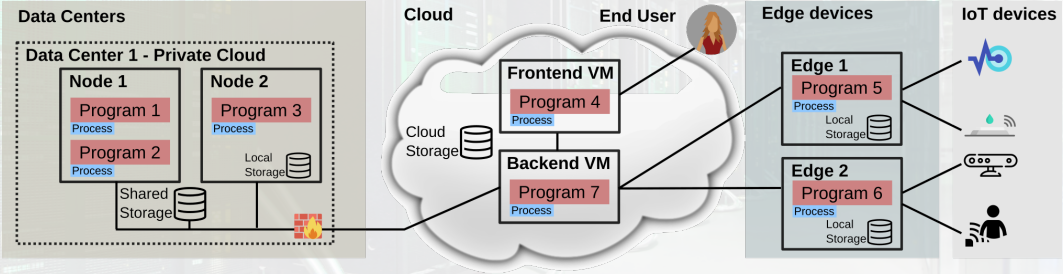
- Cloud computing has become major factor
  - Improved productivity
  - Scalability via public clouds
- Similarly Edge computing is gaining relevance
  - Location awareness
  - Ultra-low latency

Problem Statement

- Cloud computing has become major factor
  - Improved productivity
  - Scalability via public clouds
- Similarly Edge computing is gaining relevance
  - Location awareness
  - Ultra-low latency
- Produce data that requires HPC for processing
  - On-premise
  - Specialized hardware

$\Rightarrow$ System and hardware landscapes are complex

Topology


Data Centers
Data Center 1 - Private Cloud
Node 1
Program 1
Process
Program 2
Process
Node 2
Program 3
Process
Local Storage
Shared Storage
Cloud
Cloud Storage
Frontend VM
Program 4
Process
Backend VM
Program 7
Process
End User
Edge devices
Edge 1
Program 5
Process
Local Storage
Edge 2
Program 6
Process
Local Storage
IoT devices

Goals

1. Development of a cloud management framework
    - Based on Open source
    - Allow application deployment across HPC, Cloud and Edge

Goals

1. Development of a cloud management framework
   - Based on Open source
   - Allow application deployment across HPC, Cloud and Edge
2. Provide systems for administration/DevOps
   - Manage services, networks, compute resources, infrastructures

Goals

1. Development of a cloud management framework
   - Based on Open source
   - Allow application deployment across HPC, Cloud and Edge
2. Provide systems for administration/DevOps
   - Manage services, networks, compute resources, infrastructures
3. Maintain a digital twin of the compute continuum
   - Used for AI-forecasting and simulation

Goals

1. Development of a cloud management framework
   - Based on Open source
   - Allow application deployment across HPC, Cloud and Edge
2. Provide systems for administration/DevOps
   - Manage services, networks, compute resources, infrastructures
3. Maintain a digital twin of the compute continuum
   - Used for AI-forecasting and simulation
4. Automate decision making for application placement and optimization
   - Improve application performance, reliability, energy efficiency and cost

Challenges

- One system to interface with HPC, Cloud, Edge
  - HPC job-scheduler
  - Cloud containers and VMs
  - Distributed Edge-Devices

Challenges

- One system to interface with HPC, Cloud, Edge
  - HPC job-scheduler
  - Cloud containers and VMs
  - Distributed Edge-Devices
- Various HPC accelerators (e.g. GPU, TPU, IPU, FPGA)
- Various storage technologies
- Unstable networks, especially for Edge

Challenges

- One system to interface with HPC, Cloud, Edge
  - HPC job-scheduler
  - Cloud containers and VMs
  - Distributed Edge-Devices
- Various HPC accelerators (e.g. GPU, TPU, IPU, FPGA)
- Various storage technologies
- Unstable networks, especially for Edge
- Evolving system topology
  - Devices may be added or removed anytime
- Optimal scheduling is NP-complete problem
- Clusters across multiple countries
  - Respect data security and privacy laws

Addressing the Challenges

Orchestration

- Build on top of Kubernetes (K8s)
  - Cloud container orchestration platform
  - Already in use or evaluation by majority of companies[1]
  - Well established and Open source

[1] CNCF 2021 survey:
https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR_FINAL-edits-15.2.21.pdf

Orchestration

- Build on top of Kubernetes (K8s)
  - Cloud container orchestration platform
  - Already in use or evaluation by majority of companies[1]
  - Well established and Open source
- K8s is not perfect
  - Intended for homogeneous, (mostly) stable clusters
  - No concept of network bandwidth
  - No support for Edge (unstable network)

[1] CNCF 2021 survey:
https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR_FINAL-edits-15.2.21.pdf

Orchestration

- Build on top of Kubernetes (K8s)
  - Cloud container orchestration platform
  - Already in use or evaluation by majority of companies[1]
  - Well established and Open source
- K8s is not perfect
  - Intended for homogeneous, (mostly) stable clusters
  - No concept of network bandwidth
  - No support for Edge (unstable network)
- Use Volcano for HPC scheduling
- Use KubeEdge for Edge devices

[1] CNCF 2021 survey:
https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR_FINAL-edits-15.2.21.pdf

KubeEdge

- Extends K8s
- Open source

KubeEdge

- Extends K8s
- Open source
- KubeEdge cloud manager in main cluster
    - Fixed IP endpoint

KubeEdge

- Extends K8s
- Open source
- KubeEdge cloud manager in main cluster
  - Fixed IP endpoint
- KubeEdge agent on Edge hosts
  - Includes minimal K8s
  - Edge nodes require no public IP
  - Handles slow/unstable networks
  - Hosts can operate autonomous
- Number of IoT devices connect to one Edge host

Monitoring

- Build around Prometheus
  - Open source
  - Gather, store, visualize monitoring data
  - Administration, failure diagnosis, alerts
  - Provide data for scheduling system

Monitoring

- Build around Prometheus
  - Open source
  - Gather, store, visualize monitoring data
  - Administration, failure diagnosis, alerts
  - Provide data for scheduling system
- Collect relevant data
  - Node resource usage
  - Workload performance
  - Network usage and capability
  - Any-to-any node $\Rightarrow$ $(\frac{n(n-1)}{2})$ connections
- Monitoring should not add significant load

Scheduling

- Optimal scheduling is active research problem
- 4 categories of schedulers[2]
  1. Mathematical modeling
  2. Heuristics
  3. Meta-heuristics
  4. Machine learning (ML)

[2] Ahmad, Imtiaz et al. "Container Scheduling Techniques: A Survey and Assessment" journal of King Saud University 10.1016/j.jksuci.2021.03.002 (2021)

Scheduling

- Optimal scheduling is active research problem
- 4 categories of schedulers[2]
  1. Mathematical modeling
  2. Heuristics
  3. Meta-heuristics
  4. Machine learning (ML)
- Optimal scheduling is NP-complete problem
  - High-computational cost for large cluster
  - Compromise for near-optimal

$\Rightarrow$ Heuristics and ML only options (for large clusters)

- Implement as K8s custom scheduler

[2] Ahmad, Imtiaz et al. "Container Scheduling Techniques: A Survey and Assessment" journal of King Saud University 10.1016/j.jksuci.2021.03.002 (2021)

Rescheduling

- K8s supports auto-scaling
- Changing system topology may require rescheduling
  - Rescheduling is not supported

Rescheduling

- K8s supports auto-scaling
- Changing system topology may require rescheduling
  - Rescheduling is not supported
- Implement Rescheduling
  1. Regularly check long running workloads
  2. Evaluate placement options
  3. Consider cost of rescheduling
  4. Terminate sub-optimal workload and resubmit

Security and data privacy

- Internal access control
  - Role-Based Access Control (RBAC)

Security and data privacy

- Internal access control
  - Role-Based Access Control (RBAC)
- Edge node physical tampering
  - Prevent data theft through encryption
  - Prevent submission of malicious workloads via validation

Security and data privacy

- Internal access control
  - Role-Based Access Control (RBAC)
- Edge node physical tampering
  - Prevent data theft through encryption
  - Prevent submission of malicious workloads via validation
- Data placement and processing regulations/laws
  - Node labeling based on country
  - Scheduler filters eligible nodes

Energy Efficiency

- Compute centers have huge energy consumption
- Idle nodes still consume significant power

Energy Efficiency

- Compute centers have huge energy consumption
- Idle nodes still consume significant power
- Utilize *bin packaging* scheduling
  - Schedule first on nodes with high resource reservation
  - Reserve up to 70% to 90% of node resources
  - Maximize number of idle nodes
  - Automatically shutdown unneeded nodes

Conclusion

DECICE Planned Contributions

- Unified compute platform for HPC, Cloud and Edge
  - Released as Open source (MIT license)

DECICE Planned Contributions

- Unified compute platform for HPC, Cloud and Edge
  - Released as Open source (MIT license)
- Contributions to Open source software used by DECICE

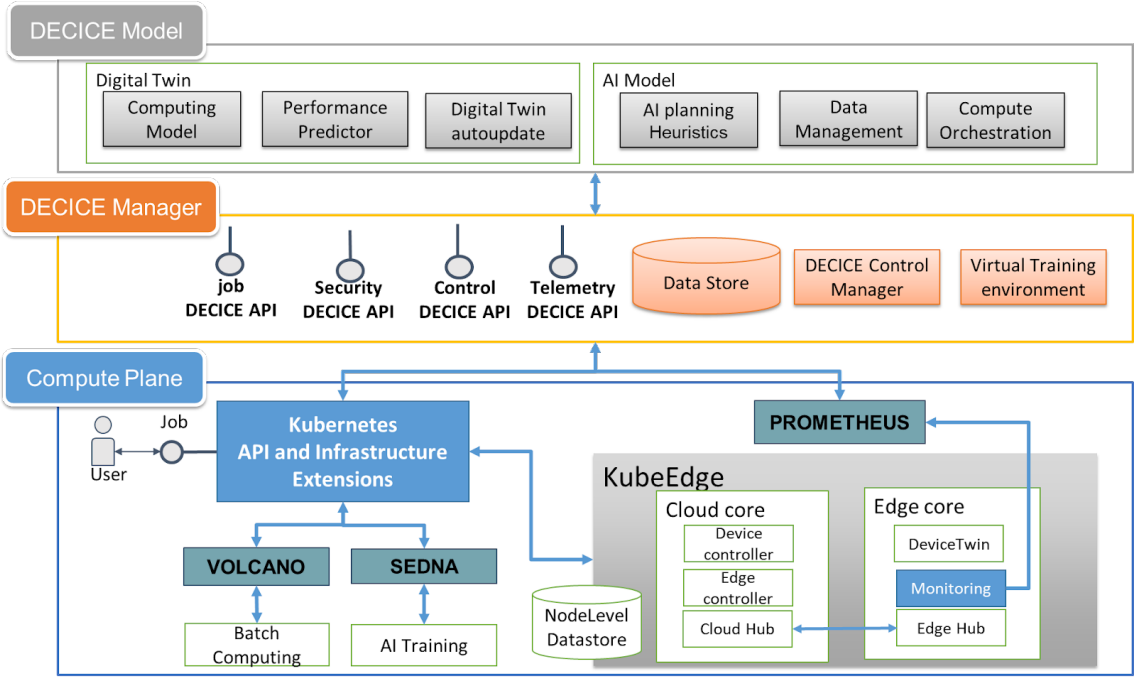
DECICE Planned Contributions

- Unified compute platform for HPC, Cloud and Edge
  - Released as Open source (MIT license)
- Contributions to Open source software used by DECICE
- Results as research papers with open access

The Decice project received funding from the European Union's Horizon 2022 research and innovation programme under grant agreement No **101092582**.

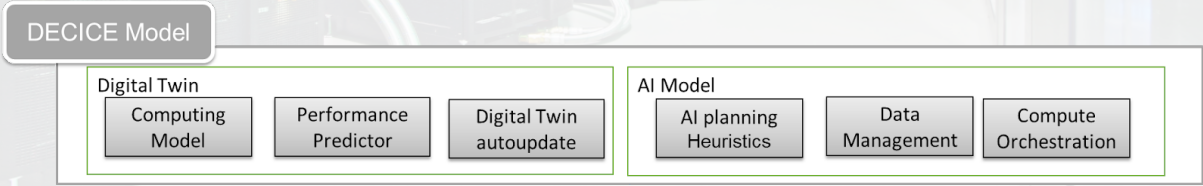
Disclaimer: This material reflects only the author's view and the EU-Commission is not responsible for any use that may be made of the information it contains.

Work plan


DECICE Model
Digital Twin
Computing Model
Performance Predictor
Digital Twin autoupdate
AI Model
AI planning Heuristics
Data Management
Compute Orchestration
DECICE Manager
job DECICE API
Security DECICE API
Control DECICE API
Telemetry DECICE API
Data Store
DECICE Control Manager
Virtual Training environment
Compute Plane
User
Job
Kubernetes API and Infrastructure Extensions
PROMETHEUS
KubeEdge
Cloud core
Device controller
Edge controller
Cloud Hub
Edge core
DeviceTwin
Monitoring
Edge Hub
VOLCANO
SEDNA
Batch Computing
AI Training
NodeLevel Datastore

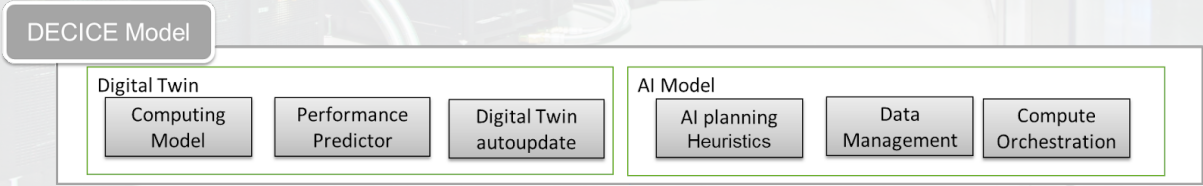
DECICE Model

- Digital twin represents entire system
  - Constructed from monitoring data
- AI model uses digital twin to make scheduling decisions



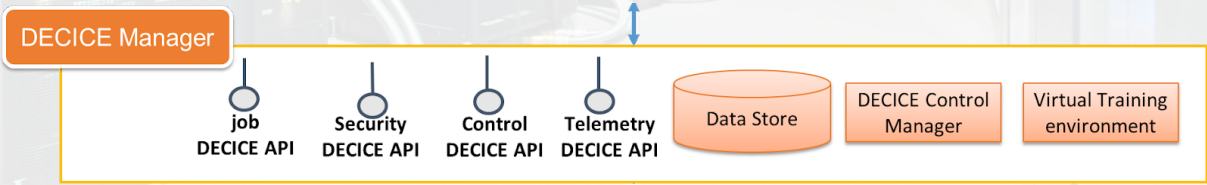
DECICE Model

- Digital twin represents entire system
  - Constructed from monitoring data
- AI model uses digital twin to make scheduling decisions
- Allows for disaster simulations
- Various AI models will be developed
- Heuristic model as baseline



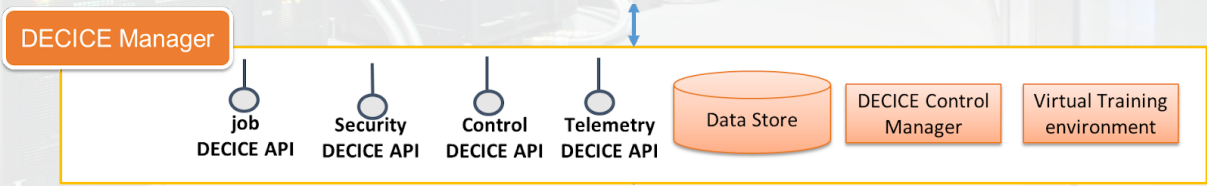
DECICE Manager

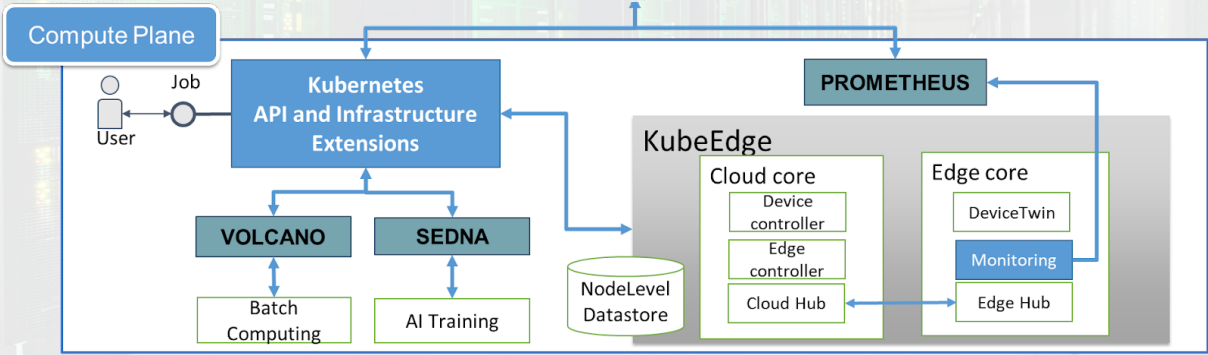
- Integrates DECICE Model with K8s
  - Keep DECICE Model adaptable to other platforms



DECICE Manager

- Integrates DECICE Model with K8s
  - Keep DECICE Model adaptable to other platforms
- Provides APIs for
  - Managing jobs/workloads
  - Adjusting security parameters
  - Adjusting system settings
  - Retrieving telemetry data
- APIs abstract underlying implementation-specifics



Compute Plane

- Integrate KubeEdge, Volcano with k8s
  - User only sees single API
  - Job submission as container
- Sedna: Toolkit for AI training