Karlsruhe Institute of Technology

Technologische Entwicklungen zur Nutzung heterogener Ressourcen

Computingstrategie in der HL-LHC-Ära Workshop 2020

Max Fischer

Helmholtz Research Field Matter

What are Heterogeneous Resources?

- Homogeneous Resources == “The Grid”
  - CE, SE, GSI, LHCOne, LHCOPN, …
  - N × 8 Cores / 24±8GiB RAM / 80GiB Disk, …
- HEP Resources outside the WLCG
  - Tier 3 Clusters, Desktops, …
  - Analysis Facilities
- Third-Party Computing Clusters
  - High-Performance Computing Cluster
  - Commercial/Scientific Cloud Providers

Motivation for Heterogeneous Resources

- Similar resources to increase resource volume
  - Integrate dedicated or opportunistic compute resources
  - Increase volume permanently or temporarily
- Different resources to increase resource coverage
  - Access specialised or novel compute resources
  - Enable optimised and prototype software solutions
- Generally: augment or replace existing heterogenous resources

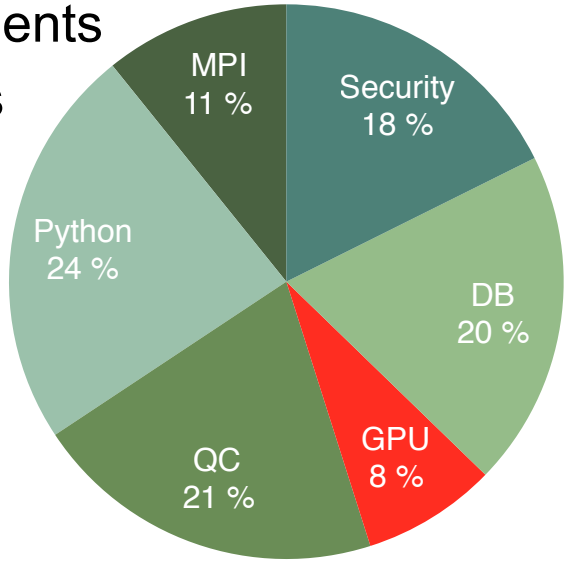
Technical Challenges

- Portability to multiple hardware architectures
  - Different flavour (x86/ARM/POWER/…)
  - Different kind (CPU/GPU/FPGA/…)
  - General challenges shared with other communities
- Portability to multiple infrastructure setup
  - Runtime environment (OS/Software/Services/…)
  - Physical environment (CPU/RAM/Disk/Network/…)
  - Specific challenges unique per community

Soft Challenges

- Training how to use heterogeneous resources
  - Programming (Vectorization, CUDA,…)
  - Collaborative Development (CI/CD, Testing, …)
  - Packaging/Distribution of specialised components
- Policies what to do with heterogeneous resources
  - Specialised versus portable tasks
  - Resource weighting (shares/weights/cost/…)
  - Existing policies do not apply anymore

GridKa School 2018
[registered participants]

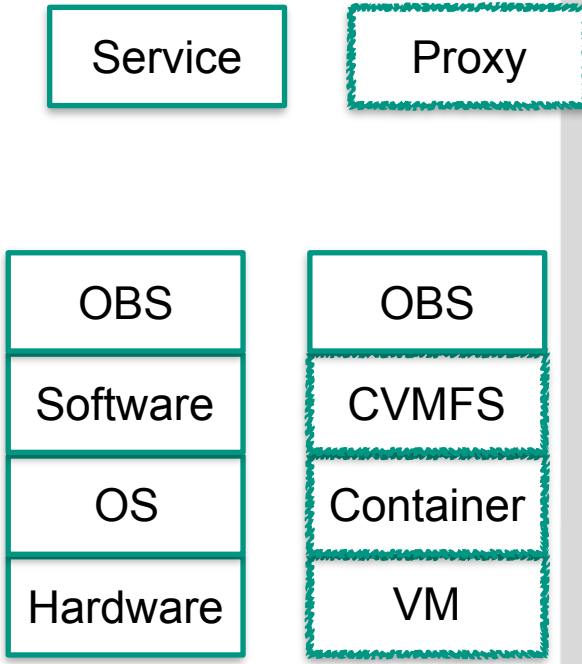


Integration of Resources

- De-Facto Standard: Overlay Batch System (OBS)
  - Wrap resource as dynamic worker node
  - Decouple acquisition and usage
  - Works with HPC, Cloud, …
- Various flavours: HTCondor, DIRAC, …
  - Range of WMS, Pilot, raw WNs
  - Trend: Generic provisioning + HTCondor
- Upcoming: Token Authentication for services
  - Expendable low-privilege passwords
  - Safe integration by resource owners

Runtime Environment

- Encapsulate local software stack
  - Layered VM, container, CVMFS as needed
  - Many available high-quality implementations
- Emulate external environment
  - Modular SQUID, SE Proxy/Cache, …
  - Technology exists, policy is work in progress
  - Hindered by VO frameworks (WPAD, …)
- Delegate Grid services
  - Grid Site as gatekeeper / service provider
  - Resources with as little “Grid” as possible



Provisioning: Backfilling / Donation

- Resource owner provides access
  - …via individual OBS worker node
  - Independent of user demand
  - Truly opportunistic / volatile
- Active donation (BOINC, HTCondor, …)
  - Provider enables/disables access
  - Often automatised via other means
- Passive backfilling (VAC, HPC schedulers, …)
  - Publish unused/fragmented resources
  - Improves utilisation for resource owner

Provisioning: Request / Scheduled

- Resource broker requests access
  - …as OBS owner/contributor
  - Estimation of user demand
  - Success is opportunistic
- Usage Prediction (CloudScheduler, ROCED, …)
  - Compute resource need from user requests
  - Predicts OBS resource broker decision
- Usage Feedback (Kubernetes, COBaID/TARDIS, …)
  - Predict resource need from observed usage
  - Observes OBS resource broker decision

Data Access: Network / Caching

- Generally inferior network capacities
  - Less/no outgoing network at HPC
  - Pay-per-use network at Clouds
  - Classical approach: Simulation only
- Actively cache data close to resource
  - Used in WLCG, prototypes for non-WLCG
  - Technology exists, policy is work in progress
- Passively schedule network by correlation
  - Limit worst-case network congestion
  - R&D for feedback based provisioning

Summary

- Heterogeneous resources are different kind of challenge
  - Need both HowTo skills and WhatTo policies
  - Mixture of training, research and development
  - Current goal is to emulate WLCG-like behaviour
- Dynamic integration via Overlay Batch System is standard
  - Backend for grid workflows, notebooks, …
  - Challenge to co-locate auxiliary services
- Major R&D: Provisioning and data access
  - Estimate resource demand by feedback vs. prediction
  - Cache data close to jobs or filter for data-access