Software Resource Disaggregation for HPC with Serverless Computing

Challenge: Resource Underutilization in HPC Datacenters

Node and CPU Utilization, Piz Daint
Memory Utilization, Piz Daint

Nodes don’t stay idle for an extended time – 70–80% are idle for less than 10 minutes. Long-running allocations cannot address these utilization gaps.

Static allocations on homogeneous resources cannot represent the heterogeneity of HPC workloads.

While HPC systems are getting more heterogeneous over time, GPU utilization by jobs is low, reinforcing the need to co-locate jobs.

To improve utilization of supercomputers, we need to enable sharing resources with fine-grained and short-term allocations.

Solution: Software Resource Disaggregation with Serverless Functions

HPC Node – Tightly Coupled Hardware
Hardware Disaggregated Data Center

High-speed network between nodes.
Homogeneous nodes with aggregated resources.

Disaggregated resources with on-demand allocation.
Dedicated interconnect for remote resource access.

No latency accessing resources.
Allows for node sharing and job co-location.
when resource consumption is compatible [2].
Nodes are overprovisioned to support all jobs.
No support for short allocations.

Higher resource utilization.
Requires new, dedicated hardware.
Latency and bandwidth penalty.

Our Work: Software Disaggregation
Disaggregated computing with serverless functions on remote resources.

Fine-grained allocations with functions.
Short-term allocations.
Applies to existing HPC systems.
No penalty on standard allocations.

Bringing Serverless Disaggregation to HPC Systems with rFaaS

Batch systems release and reclaim resources.

rFaaS [3], a high-performance serverless platform to elevate FaaS performance issues [4].

Functions access underutilized resources.

Dedicated interconnect for remote resource access.

Safe multi-tenant node sharing with containers.
Host warm containers in node idle memory.
Extend FaaS with HPC containers, accelerators, parallel filesystem.

Evaluation

CPU Disaggregation: co-locating CPU workloads
Memory disaggregation: co-locating RMA function

Co-location of long-running job with function-style, short-running applications.
Slowdown of the batch job LULESH.

Overhead below 7.5% when solving 1GB.
Activity of the co-located remote memory function: read (left) and write (right).

Integration of FaaS into HPC applications
Each thread/rank offloads half of their workload to an rFaaS function.

CPU Disaggregation: co-locating GPU and CPU workloads
GPU disaggregation: co-locating GPU and CPU workloads

Co-location of long-running job with function-style GPU applications.
Slowdown of the batch job LULESH.

Slowness of both CPU and GPU co-located application.

References