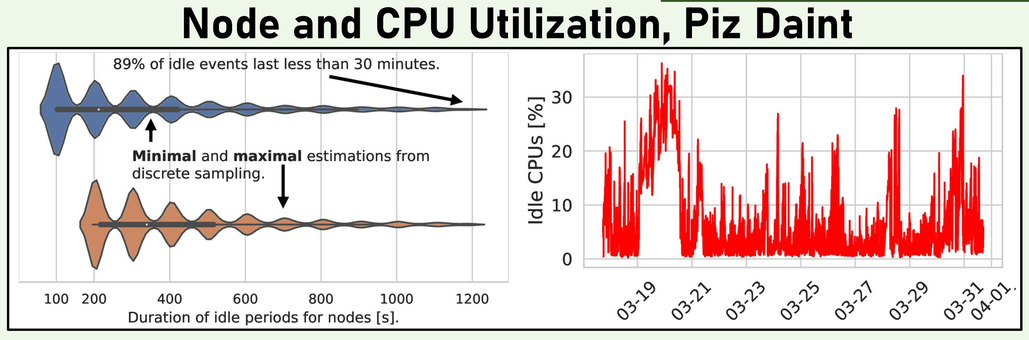
Marcin Copik
Alexandru Calotoiu (advisor)
Torsten Hoefler (advisor)

Software Resource Disaggregation for HPC with Serverless Computing

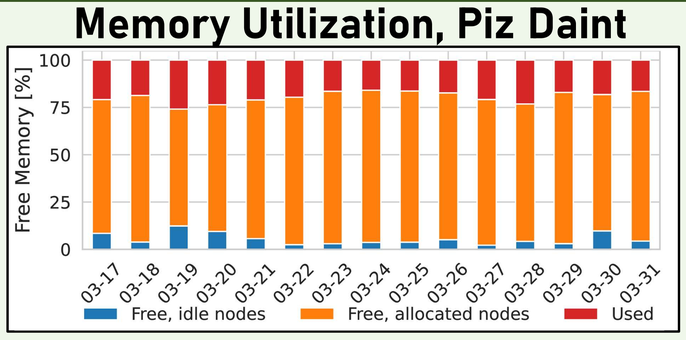
Challenge: Resource Underutilization in HPC Datacenters

Node and CPU Utilization, Piz Daint



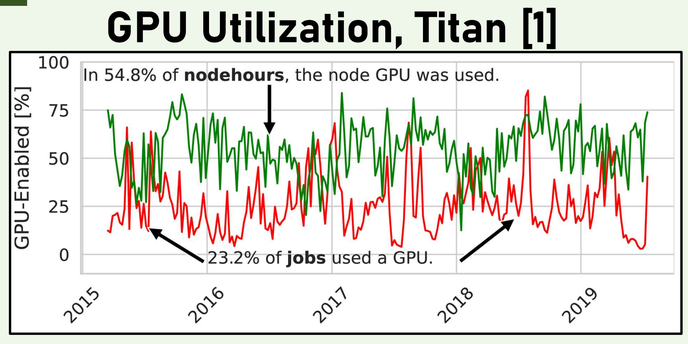
Nodes don't stay idle for an extended time – 70-80% are idle for less than 10 minutes. Long-running allocations cannot address these utilization gaps.

Memory Utilization, Piz Daint



Static allocations on homogeneous resources cannot improve memory utilization because these do not represent the heterogeneity of HPC workloads.

GPU Utilization, Titan [1]

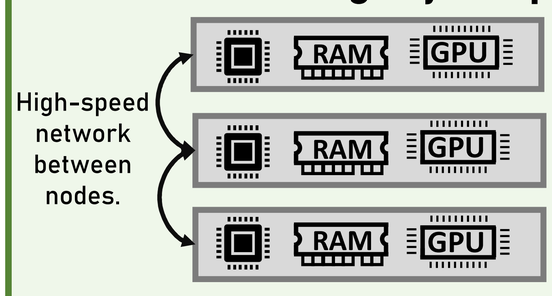


While HPC systems are getting more heterogeneous over time, GPU utilization by jobs is low, reinforcing the need to co-locate jobs.

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

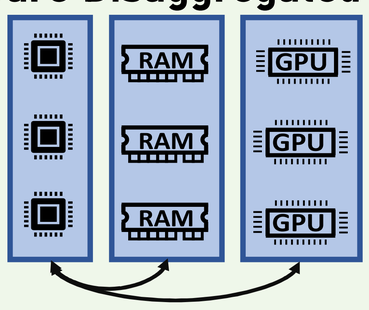
Solution: Software Resource Disaggregation with Serverless Functions

HPC Node – Tightly Coupled Hardware



- ✓ 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.

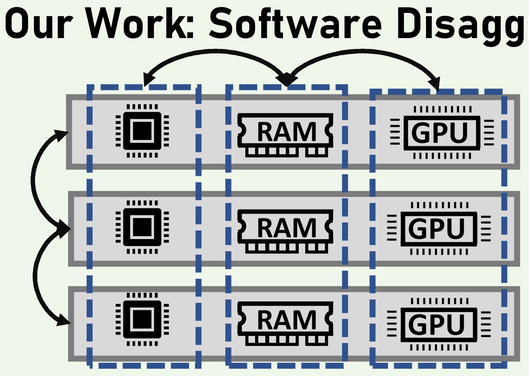
Hardware Disaggregated Data Center



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

Our Work: Software Disaggregation

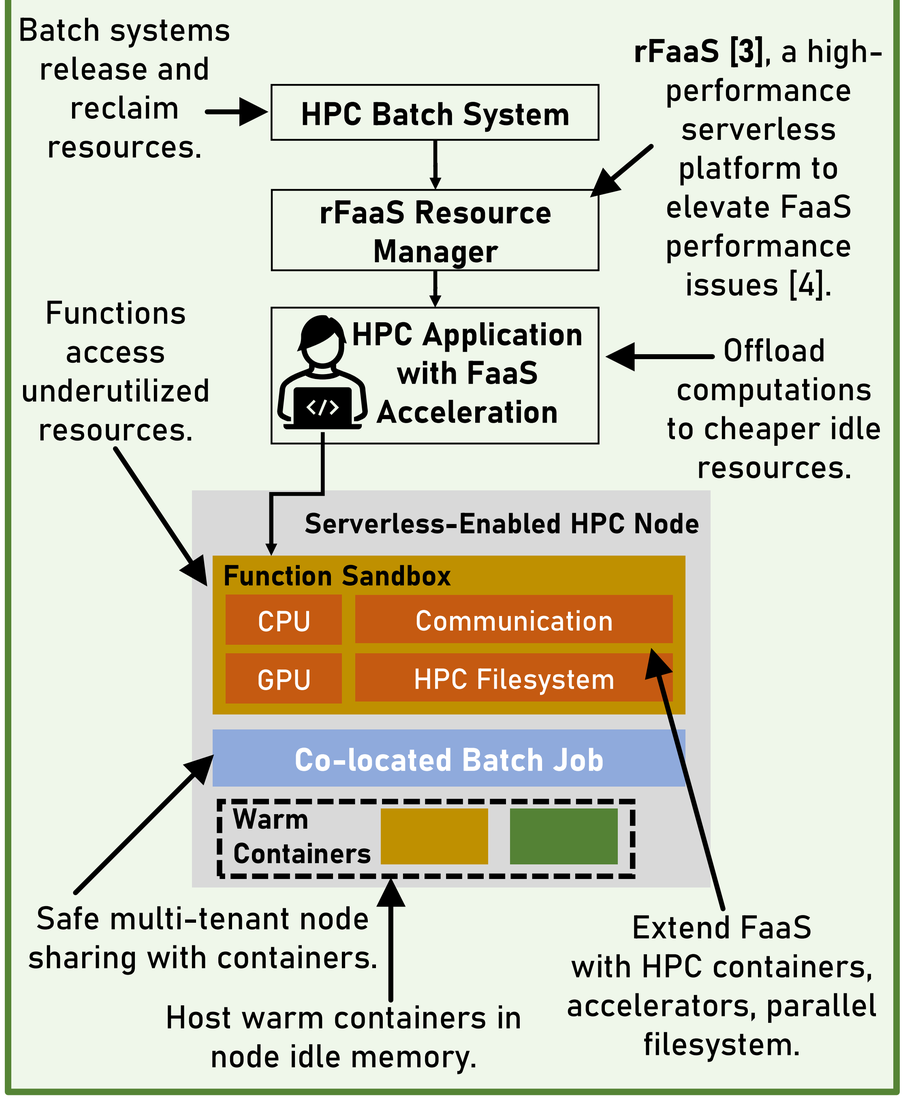
Deploy on existing HPC systems.



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.

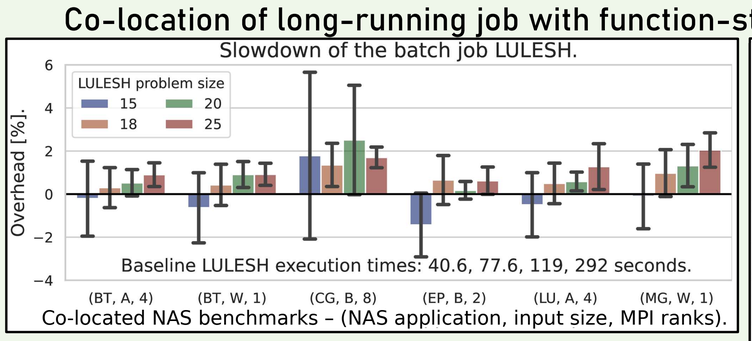
Bringing Serverless Disaggregation to HPC Systems with rFaaS

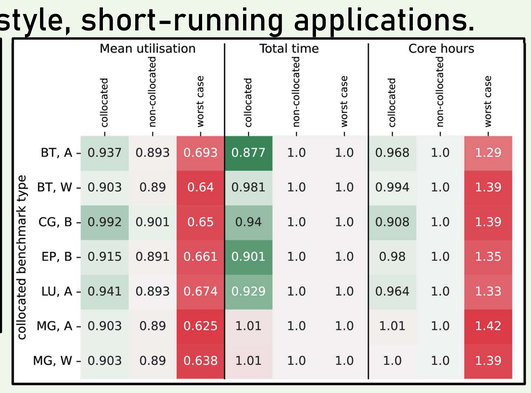


Evaluation

CPU Disaggregation: co-locating CPU workloads

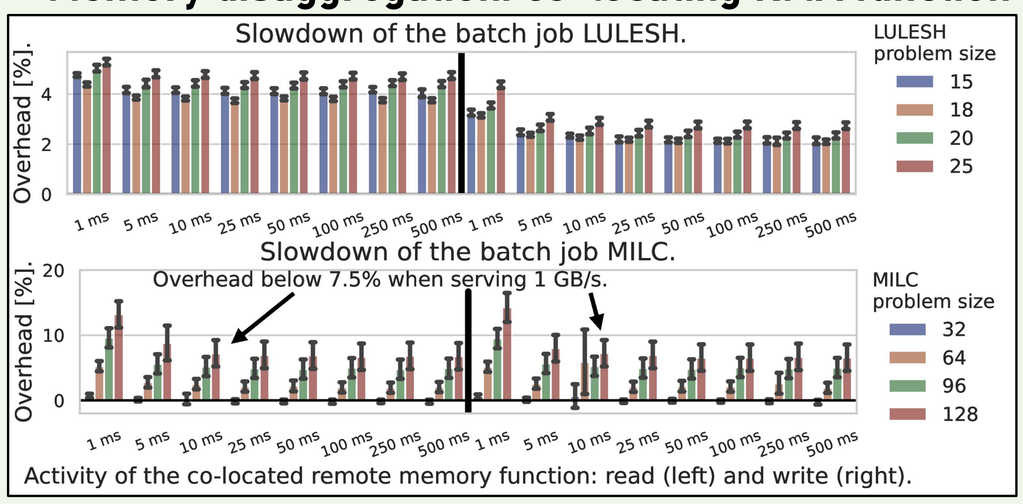
Co-location of long-running job with function-style, short-running applications.





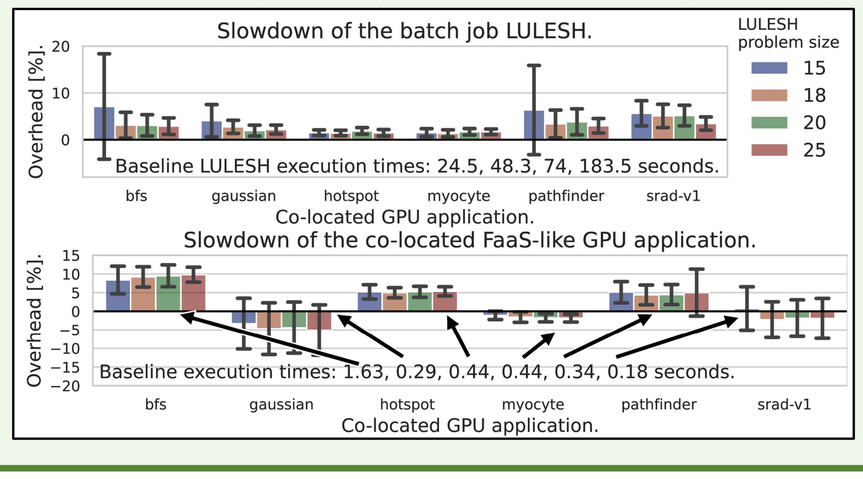
collocated benchmark type	Mean utilisation: collocated	Mean utilisation: non-collocated	Mean utilisation: worst case	Total time: collocated	Total time: non-collocated	Total time: worst case	Core hours: collocated	Core hours: non-collocated	Core hours: worst case
BT, A	0.937	0.893	0.693	0.877	1.0	1.0	0.968	1.0	1.29
BT, W	0.903	0.89	0.64	0.981	1.0	1.0	0.994	1.0	1.39
CG, B	0.992	0.901	0.65	0.94	1.0	1.0	0.908	1.0	1.39
EP, B	0.915	0.891	0.661	0.901	1.0	1.0	0.98	1.0	1.35
LU, A	0.941	0.893	0.674	0.929	1.0	1.0	0.964	1.0	1.33
MG, A	0.903	0.89	0.625	1.01	1.0	1.0	1.01	1.0	1.42
MG, W	0.903	0.89	0.638	1.01	1.0	1.0	1.0	1.0	1.39

Memory disaggregation: co-locating RMA function



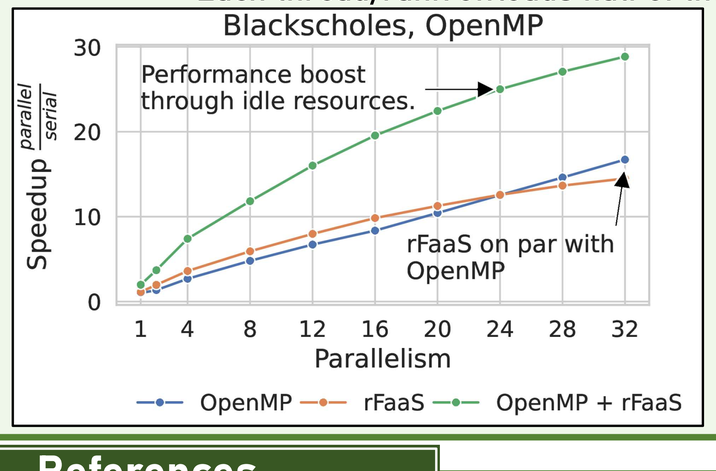
GPU disaggregation: co-locating GPU and CPU workloads

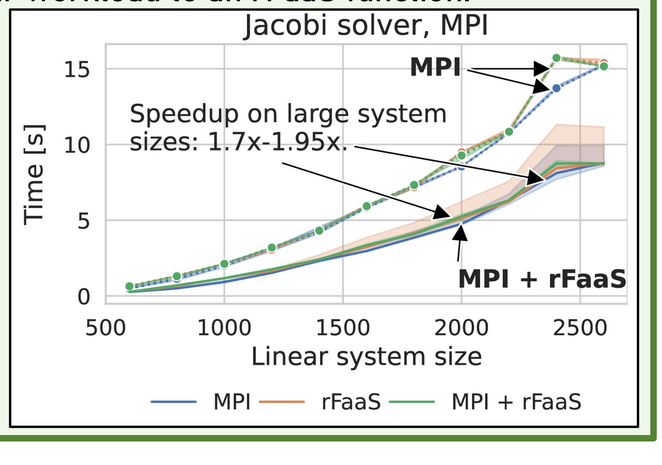
Co-location of long-running job with function-style GPU applications.



Integration of FaaS into HPC applications

Each thread/rank offloads half of their workload to an rFaaS function.





References

[1] Wang F. et al., "Learning from Five-year Resource-Utilization Data of Titan System", IEEE CLUSTER 2019
[2] Breslow A. et al., "The case for colocation of high performance computing workloads", Concurrency and Computation: Practice and Experience, 2013
[3] Copik M. et al., "rFaaS: RDMA-Enabled FaaS Platform for Serverless High-Performance Computing", arXiv 2021
[4] Copik M. et al., "SeBS: A Serverless Benchmark Suite for Function-as-a-Service Computing", ACM/IFIP Middleware 2021

ETH zürich

Poster voiceover

Preprint, code and artifact

Check our work on high-performance serverless!

SC22 Dallas, TX hpc accelerates.