Marcin Copik
Torsten Hoefler (advisor)

High-Performance Serverless for HPC and Clouds

Serverless Functions & High Performance: Challenges, Restrictions, Opportunities

- How does serverless performance look like? Can we measure it?
- Can functions communicate efficiently in FaaS?
- How to build serverless services?
- Can we make serverless invocations fast?
- How can functions improve the efficiency of HPC systems?
- How to make the programming model more efficient?

SeBS, the Serverless Benchmark Suite

Understanding FaaS performance with a representative and standardized benchmark suite.

Functions

- Website and utility functions.
- Multimedia processing.
- Machine learning inference.
- Scientific applications.
- Serverless workflows.
- Communication benchmarks.

Platforms: AWS Lambda, Azure Functions, Google Cloud Functions, APACHE OpenWhisk

Languages: python, node.js, C++

Insights Into Serverless Performance

- Performance overheads of FaaS are not uniformly distributed across application types.
- Transition from a VM to serverless can be accompanied by significant performance losses.
- Static billing and allocation policies for I/O and CPU lead to large resource waste.
- Container eviction policies are agnostic to function properties.
- We derive analytical models of container recycling.

Middleware'21 paper. **spcl/serverless-benchmarks**

FMI, Serverless Communication

Bringing direct and collective communication to serverless with MPI-compatible interface.

FaaS Without FMI

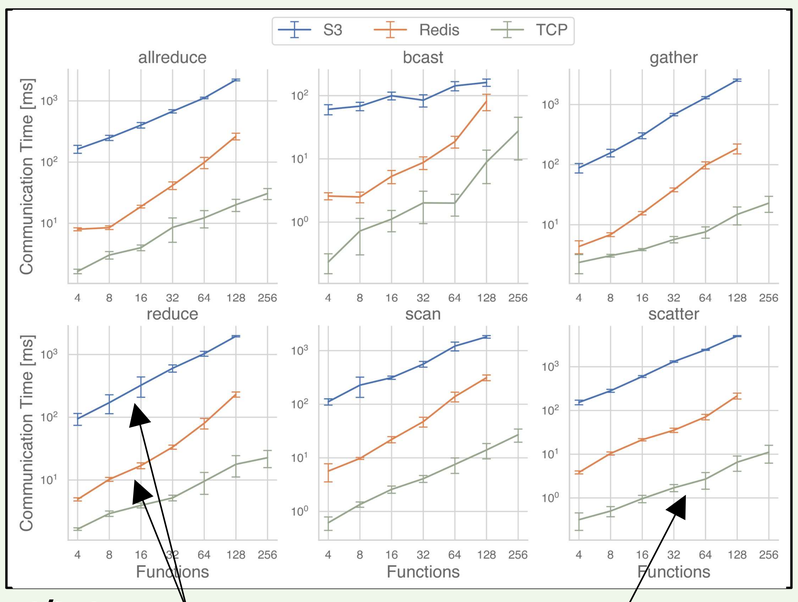
Data moved through storage and caches

FaaS With FMI

NAT Hole Punching

Data moved directly over TCP.

FMI Collectives on AWS Lambda



Storage too slow for HPC – FMI brings TCP to help.

ICS '23 paper. **spcl/fmi**

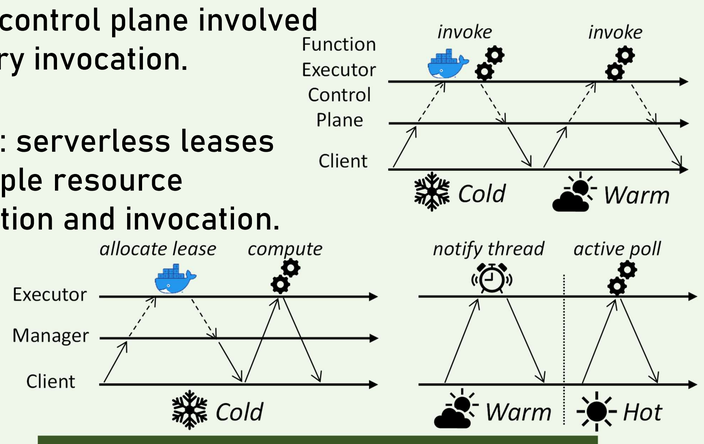
rFaaS: Serverless + RDMA

Using RDMA and leases for FaaSt invocations in HPC.

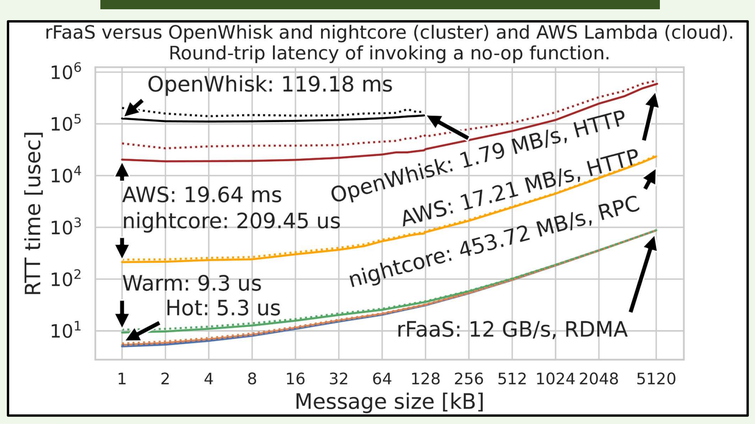
Optimized Invocation Path in rFaaS

FaaS: control plane involved in every invocation.

rFaaS: serverless leases decouple resource allocation and invocation.



rFaaS Invocations on HPC Cluster



IPDPS '23 paper. **spcl/rFaaS**

Software Resource Disaggregation with Serverless Functions

Co-locating HPC workloads and functions targets nodes with short availability and improves system utilization.

HPC Node – Tightly Coupled Hardware

High-speed network between nodes.

Homogeneous nodes with aggregated resources.

Deploy on existing HPC systems.

Software Disaggregation

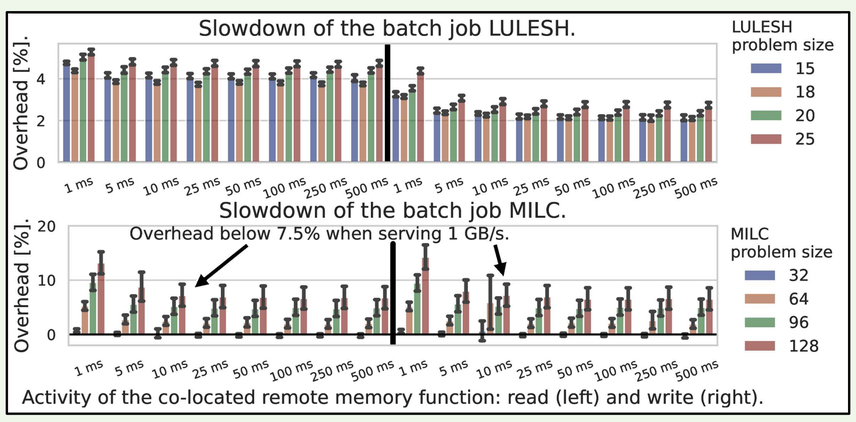
Hardware Disaggregated Data Center

Disaggregated resources with on-demand allocation.

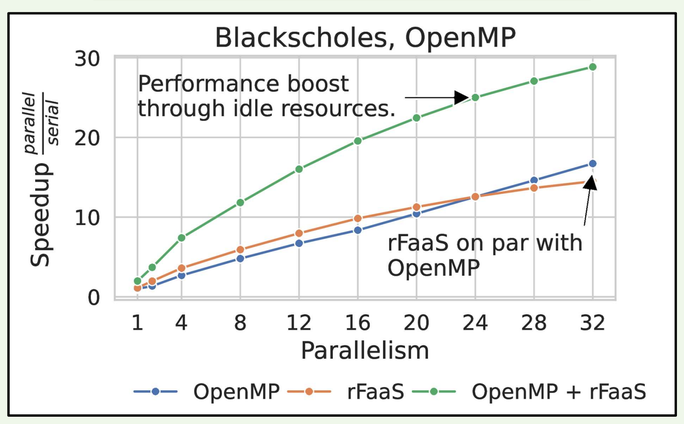
Dedicated interconnect for remote resource access.

Disaggregated computing with serverless functions on remote resources.

Colocating Memory Sharing Functions with Batch Workloads



Offloading HPC to Functions



Paper preprint.

Building Serverless Services with FaaSKeeper.

Path from server-centric deployment to FaaS on the example of a complex service: ZooKeeper.

"Serverful"

- Compute and storage coupled in a server.
- Persistent allocations.
- Difficult scaling.

Serverless

- Disaggregated compute and storage.
- Flexible resource allocation.
- Scale down to zero.

spcl/FaaSKeeper

Paper preprint.

Cost ratio of ZooKeeper and FaaSKeeper, 90% reads.

ZooKeeper configuration	100K	500K	1M	2M	5M
3 x t3.small	10.15	2.03	1.01	0.51	0.20
5 x t3.small	16.91	3.38	1.69	0.85	0.34
7 x t3.small	23.67	4.73	2.37	1.18	0.47
9 x t3.small	30.44	6.09	3.04	1.52	0.61
3 x t3.medium	20.29	4.06	2.03	1.01	0.41
5 x t3.medium	33.82	6.76	3.38	1.69	0.68
7 x t3.medium	47.35	9.47	4.73	2.37	0.95
9 x t3.medium	60.88	12.18	6.09	3.04	1.22
3 x t3.large	40.58	8.12	4.06	2.03	0.81
5 x t3.large	67.64	13.53	6.76	3.38	1.35
7 x t3.large	94.70	18.94	9.47	4.73	1.89
9 x t3.large	121.75	24.35	12.18	6.09	2.44

Requests per day.

Larger VM allocations increase data durability.

Serverless shines in low op/s scenarios.

PraaS: Process-as-a-Service

Serverless process: introducing new abstraction to improve data locality and integration.

OS Process
Nano- and micro-second latency of OS primitives.

Serverless Function
Millisecond latency of cloud proxies.

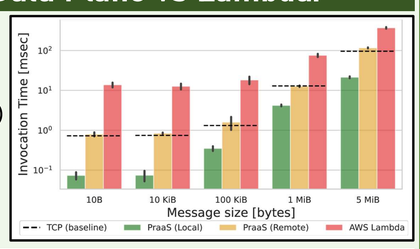
Serverless Process
Microsecond latency of PraaS backend.

spcl/PraaS

Paper preprint.

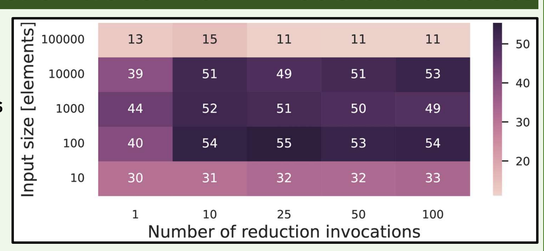
PraaS Data Plane vs Lambda.

Invoking function in the same (**local**) and another (**remote**) process is more efficient than **FaaS** Invocations.



Reduction Benchmark: PraaS State vs S3.

Accessing local state in process is faster than using remote cloud storage.



ETH zürich

Electronic poster version.

IPDPS 2023 • St. Petersburg, Florida USA