Hewlett Packard **Labs**

Enabling HPC Scientific Workflows for Serverless

Anderson Andrei Da Silva
Postdoctoral Research Engineer, Hewlett Packard Labs

Anderson Andrei Da Silva∗ , Rolando Pablo Hong Enriquez∗ , Gourav Rattihalli∗ , Vijay Thurimella∗, Rafael Ferreira da Silva‡ , Dejan Milojicic∗

∗ Hewlett Packard Labs, Milpitas, CA, USA
‡ Oak Ridge National Laboratory, Oak Ridge, TN, USA

6th workshop on Containers and new orchestration paradigms for isolated environments in HPC (CANOPIE-HPC) at Supercomputing 2024

Introduction

Serverless Computing

- A computing paradigm that deploys **fast execution functions** on **demand**, **reserving resources** just for the **time to run** the applications.
  - It is very beneficial for **auto-scaling** and **low data-communication** workloads;
  - It encompasses the FaaS (Function as a Service) approach:
    - Functions are deployed inside environments such as containers;
    - Functions respond to events.

HPC/ AI Workflows

- **Scientific jobs**, split in different **coordinated tasks**;
  - We have witnessed a **growing population** of different HPC/ AI workflows;
  - It is difficult but **crucial** to **identify patterns** for **management purposes**;

Motivation

Why is it important **?**

- Infrastructure Cloud computing entities have dealt with the difficulty of managing their resources for handling these many types of workflows;
- Serverless can address the problem above, but do not offer mechanisms to manage workflow invocations.

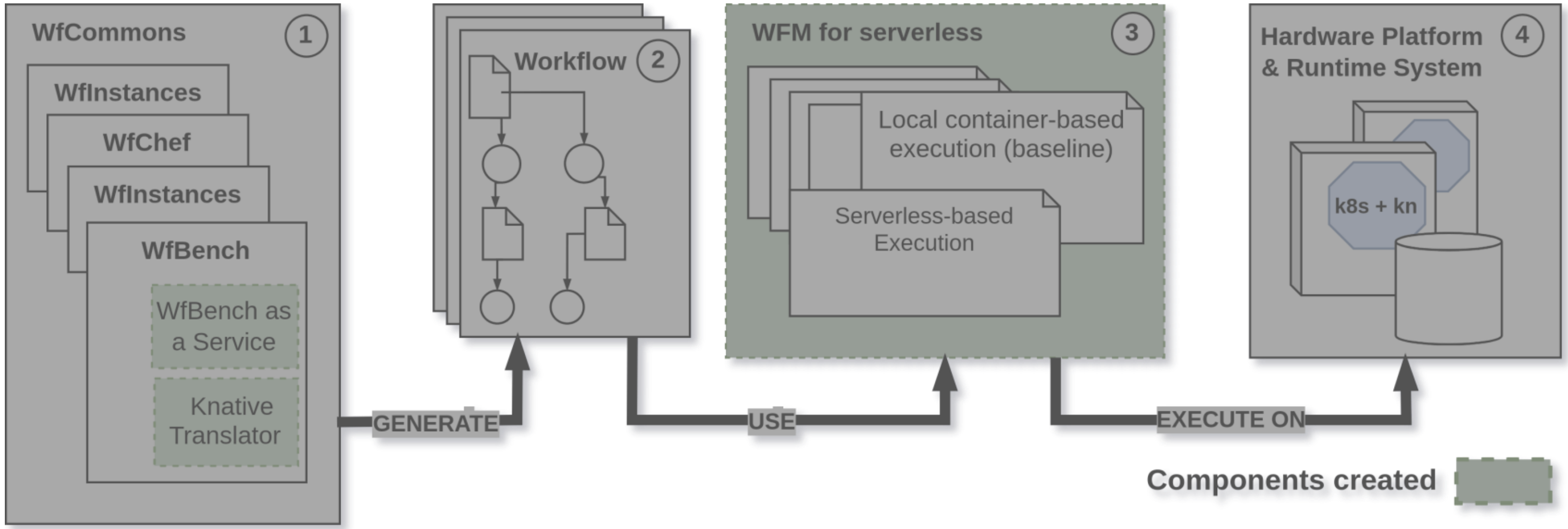
How are we tackling these challenges?

- We propose a framework for executing and evaluating HPC scientific workflows on serverless

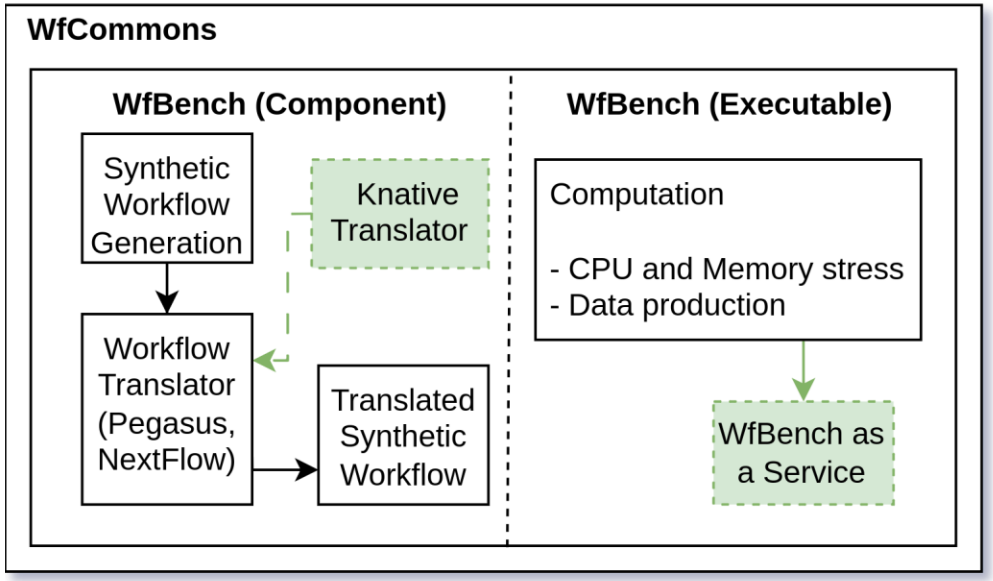
What are our contributions?

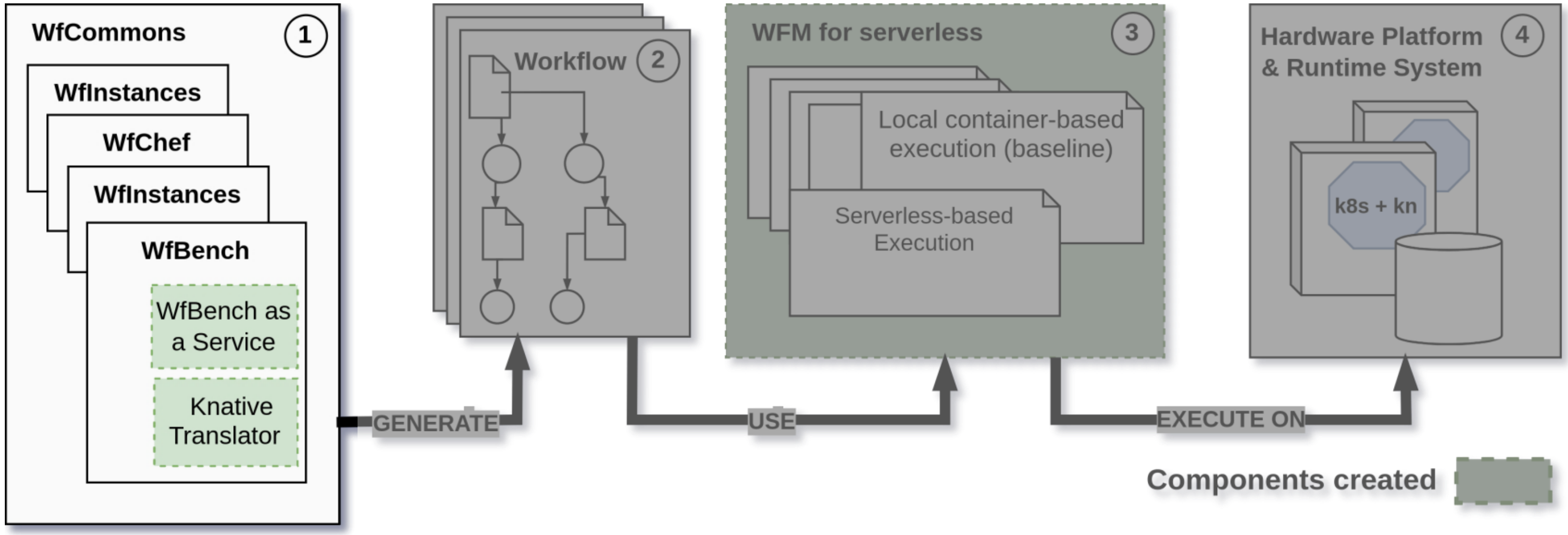
- A **framework** that enables **HPC scientific workflows** on **serverless computing**, assembling:
  - a **workflow manager for serverless** (evaluated using **Knative**);
  - the **WfCommons framework** (extended)**.**
- An **extensive evaluation**, comparing **serverless and bare-metal containers** in terms of granularity, execution time, power, CPU, and memory usage.

System Architecture

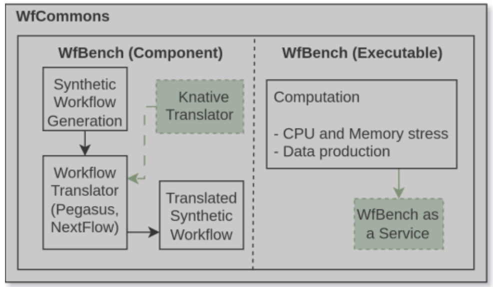


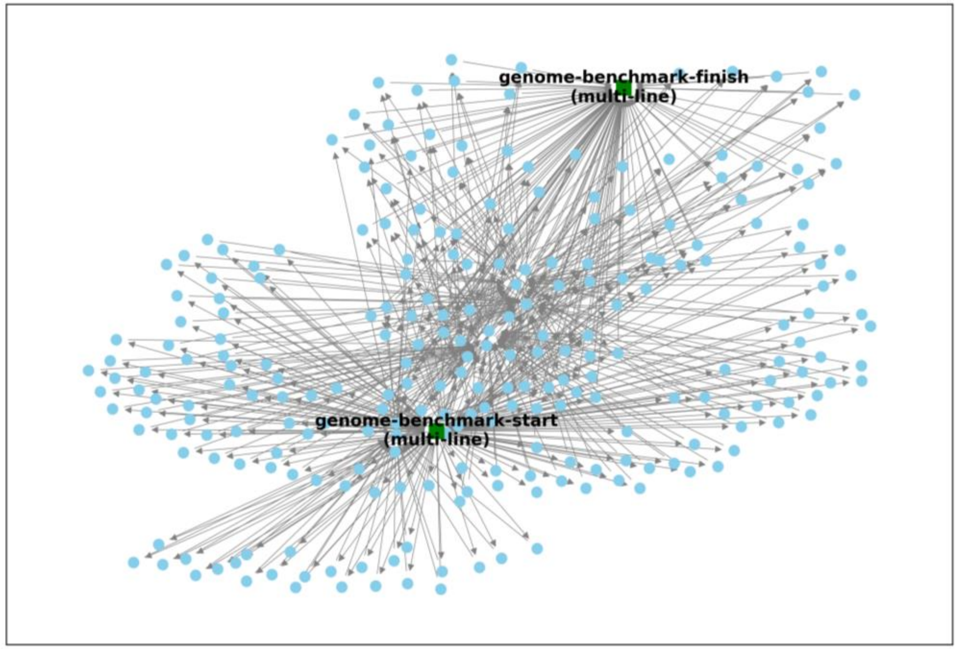
System Architecture

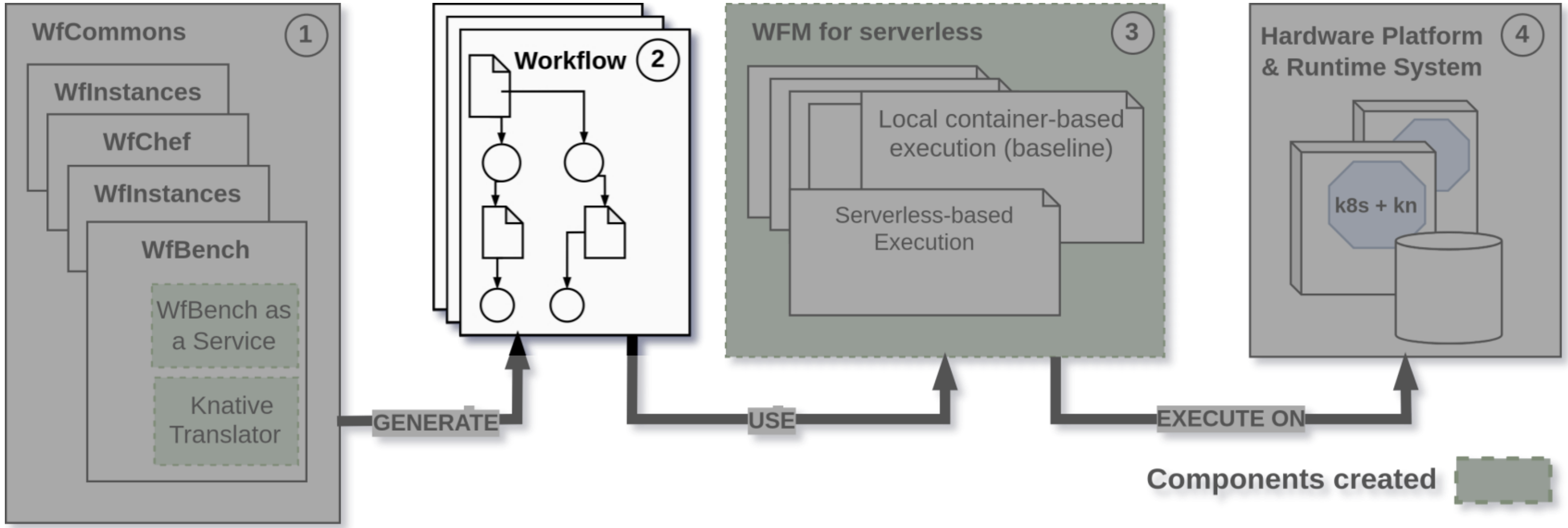




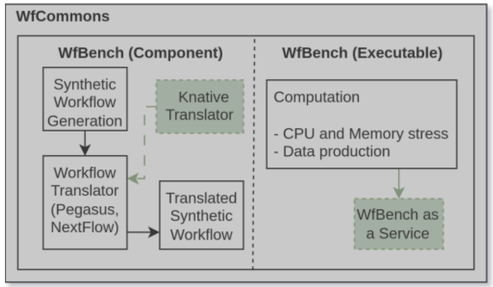
System Architecture




genome-benchmark-finish (multi-line)
genome-benchmark-start (multi-line)



System Architecture



WfCommons

WfBench (Component)

Synthetic Workflow Generation

Knative Translator

Workflow Translator (Pegasus, NextFlow)

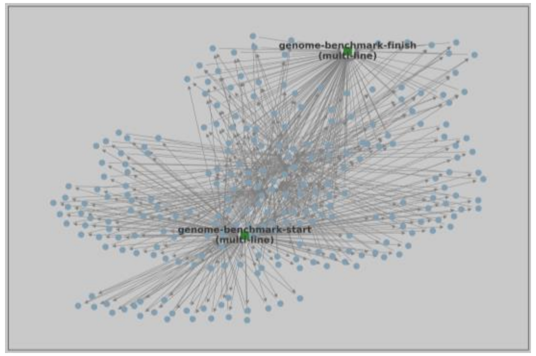
Translated Synthetic Workflow

WfBench (Executable)

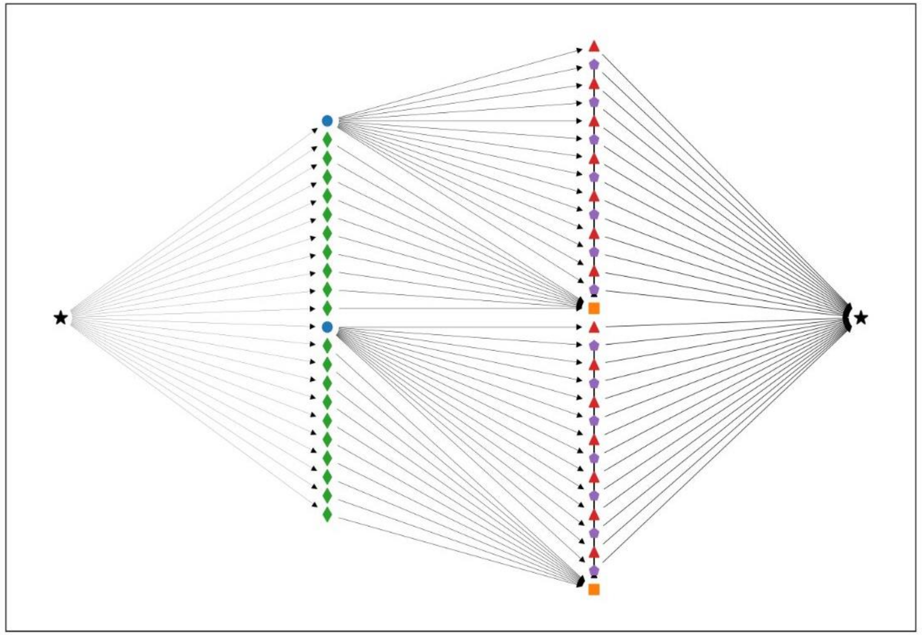
Computation

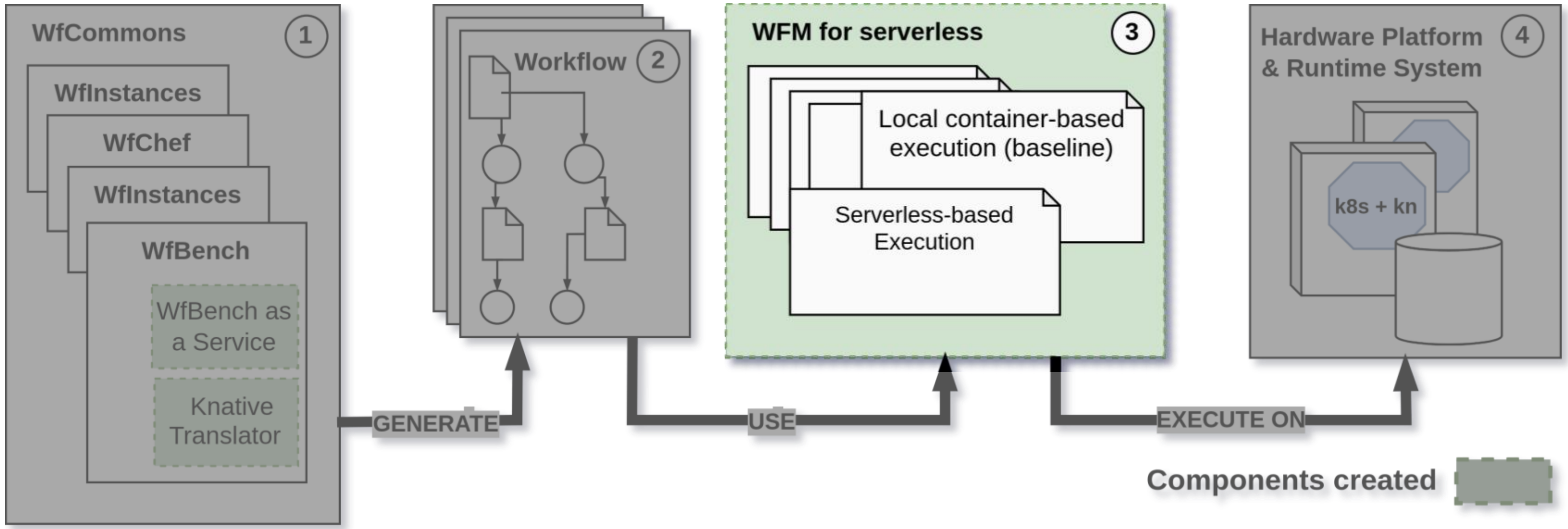
- CPU and Memory stress
- Data production

WfBench as a Service



genome-benchmark-finish (multi-line)
genome-benchmark-start (multi-line)





WfCommons 1

WfInstances

WfChef

WfInstances

WfBench

WfBench as a Service

Knative Translator

Workflow 2

WFM for serverless 3

Local container-based execution (baseline)

Serverless-based Execution

Hardware Platform & Runtime System 4

k8s + kn

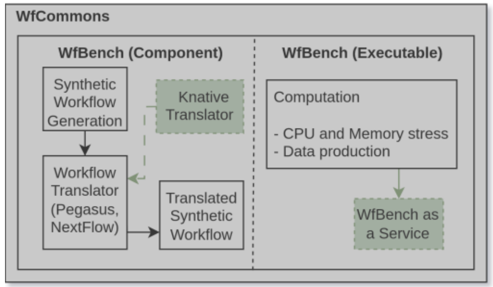
GENERATE

USE

EXECUTE ON

Components created

System Architecture



WfCommons

WfBench (Component)

Synthetic Workflow Generation

Knative Translator

Workflow Translator (Pegasus, NextFlow)

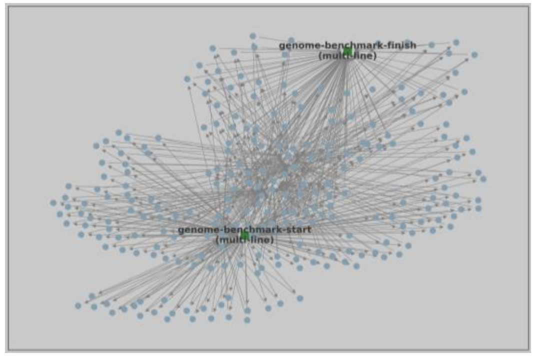
Translated Synthetic Workflow

WfBench (Executable)

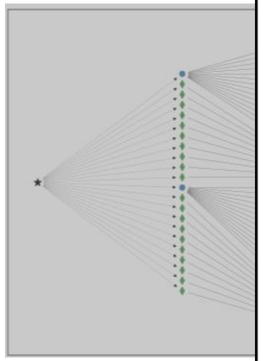
Computation

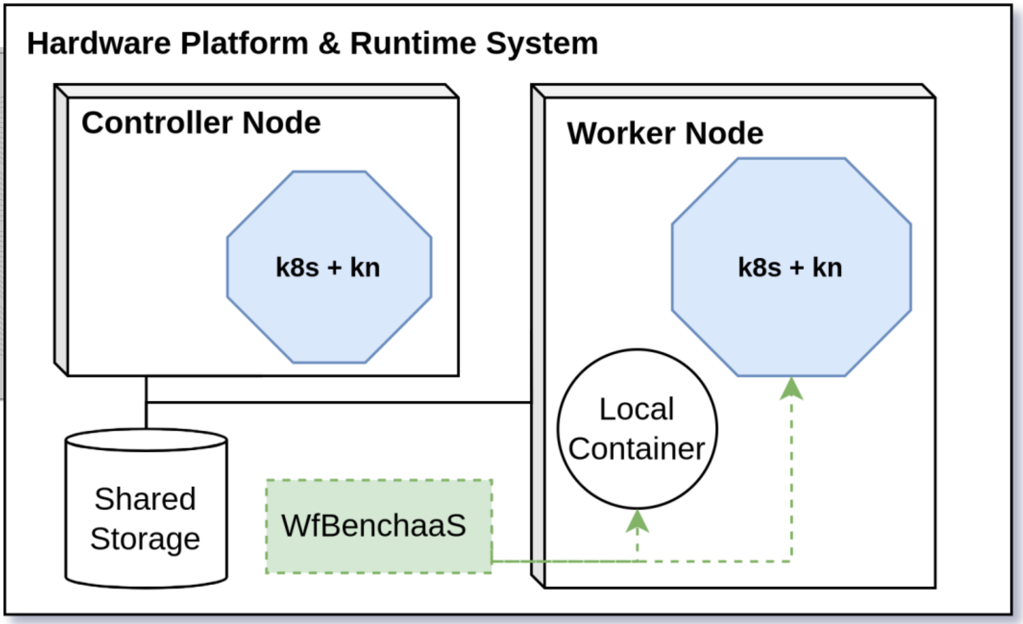
- CPU and Memory stress
- Data production

WfBench as a Service



genome-benchmark-finish (multi-line)
genome-benchmark-start (multi-line)





Hardware Platform & Runtime System

Controller Node

k8s + kn

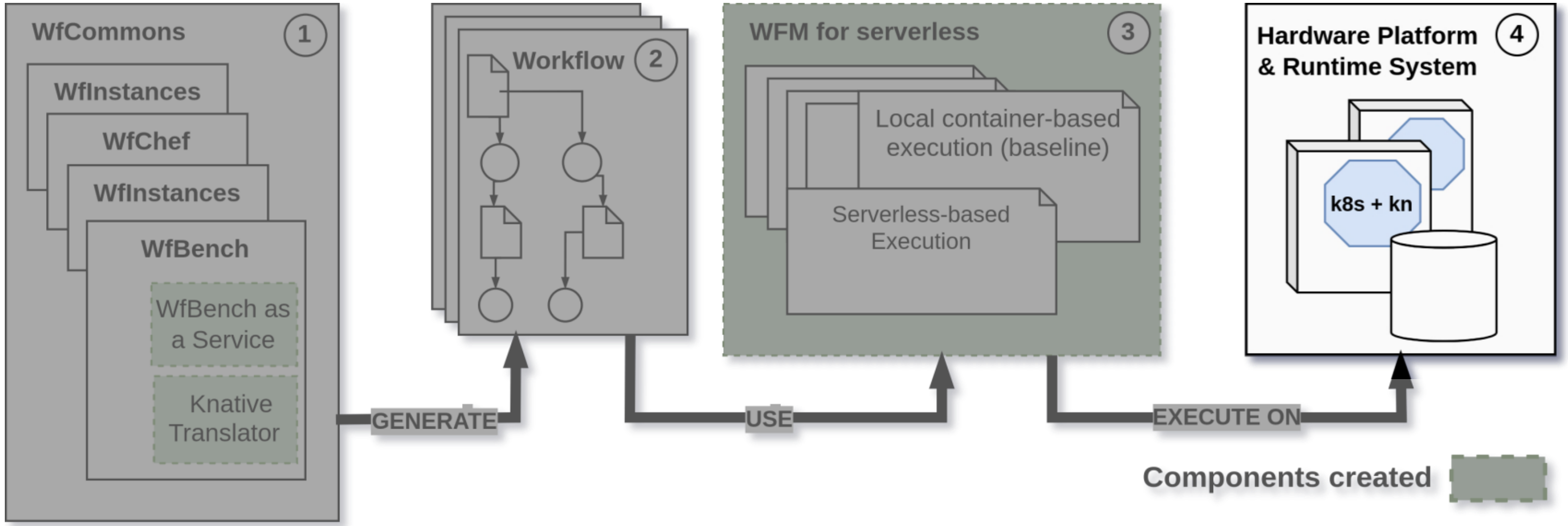
Worker Node

k8s + kn

Local Container

Shared Storage

WfBenchaaS



WfCommons 1

WfInstances

WfChef

WfInstances

WfBench

WfBench as a Service

Knative Translator

Workflow 2

WFM for serverless 3

Local container-based execution (baseline)

Serverless-based Execution

Hardware Platform & Runtime System 4

k8s + kn

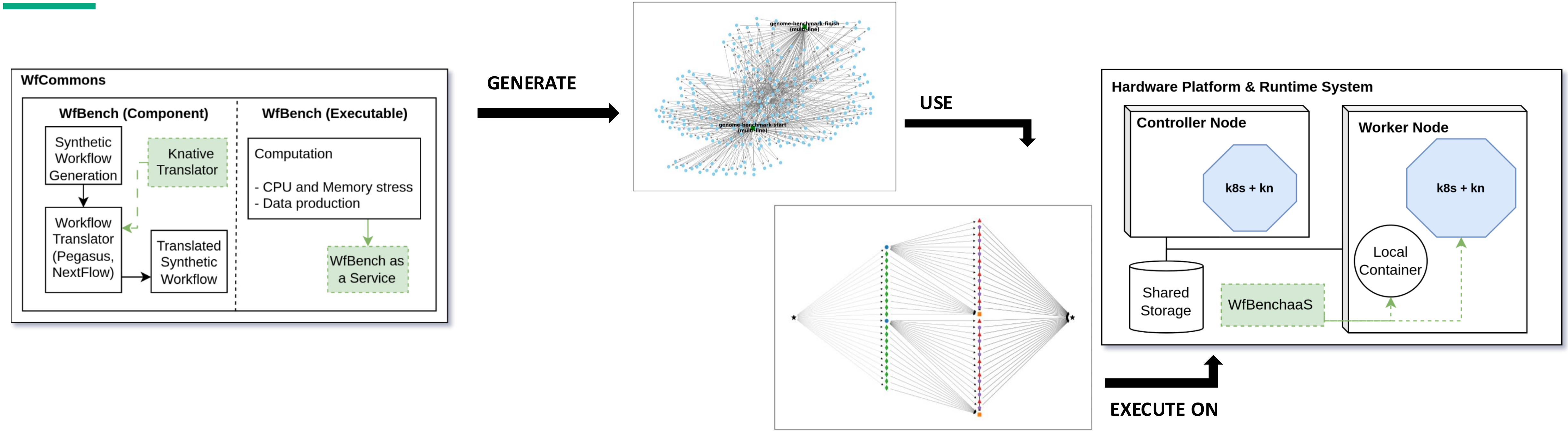
GENERATE

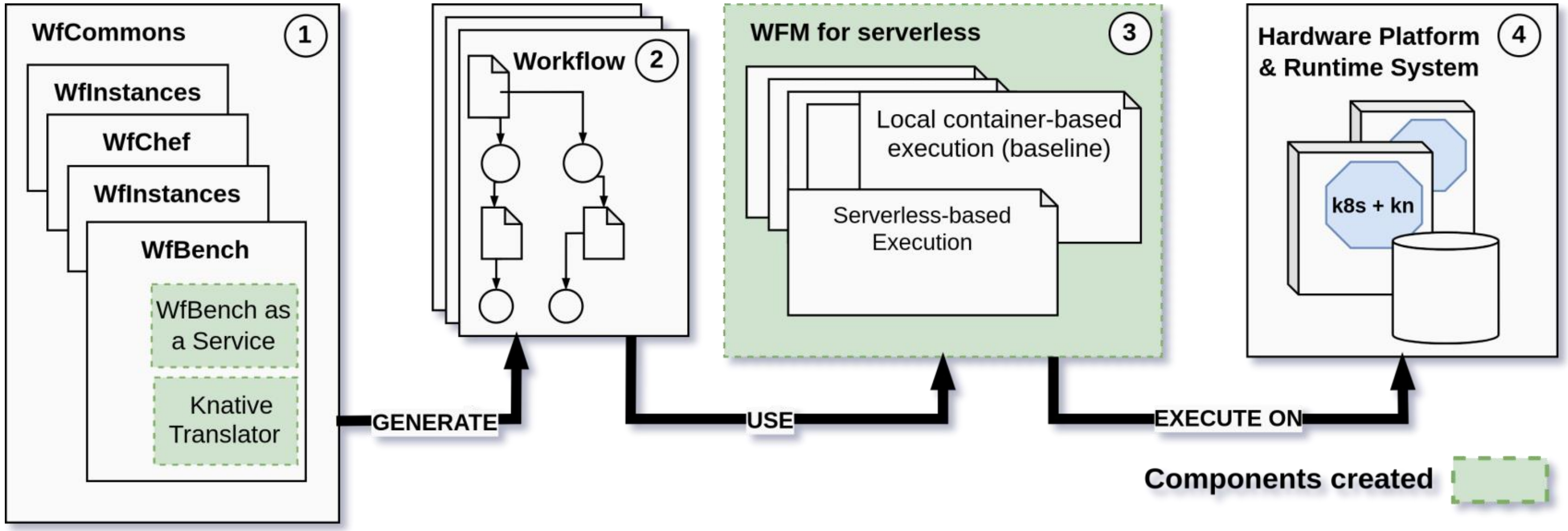
USE

EXECUTE ON

Components created

System Architecture

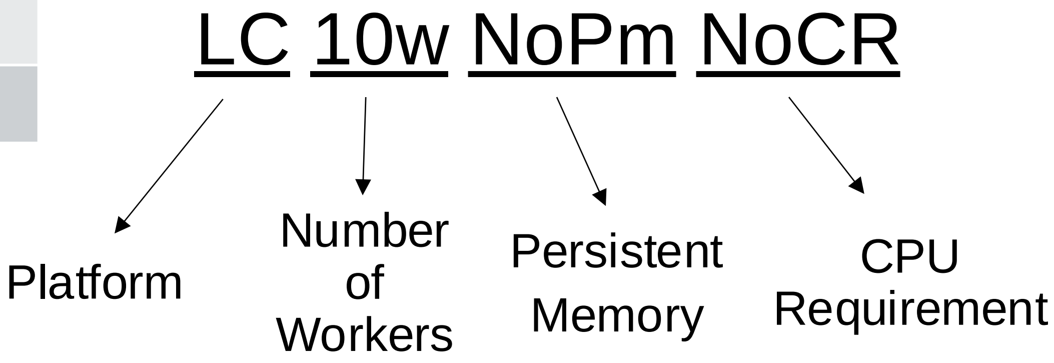




DoE and Methodology

Parameter	Value
Platform	Bare-metal local containers, Knative
Workflow	Blast, Bwa, Cycles, Epigenomics, Genomes, Seismology, Srasearch
Workflow size	250, 500, 1000 tasks
CPU stress	100%
Number of workers	1, 10 workers
Function's granularity	Coarse-grained, fine-grained
Persistent Memory	With, without

Knative	Local Containers
Kn1wPM, Kn1wNoPM, Kn10wNoPM, Kn1000wPM,	LC1wPM, LC1wNoPM, LC10wNoPM , LC10wNoPmNoCR, LC1000wPM
Total of experiments	140 experiments



Anderson Andrei Da Silva, R. P. Hong Enriquez, G. Rattihalli , V. Thurimella, R. Ferreira da Silva, and D. Milojicic ,
Enabling HPC Scientific Workflows for Serverless, CANOPIE-HPC at SC24

Experimental Results: Fine-grained Serverless and Local Containers

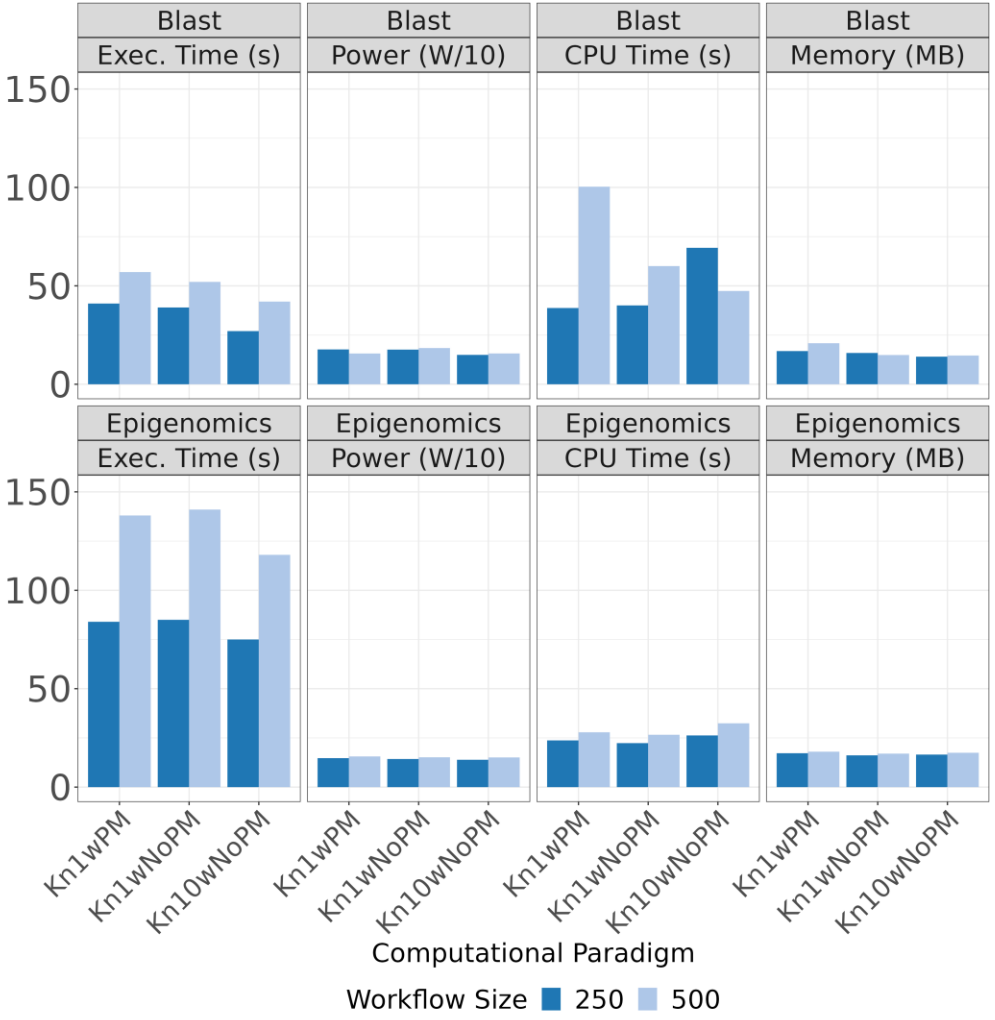


Fig. 4: Comparison between different setups for the Serverless Computational Paradigm.

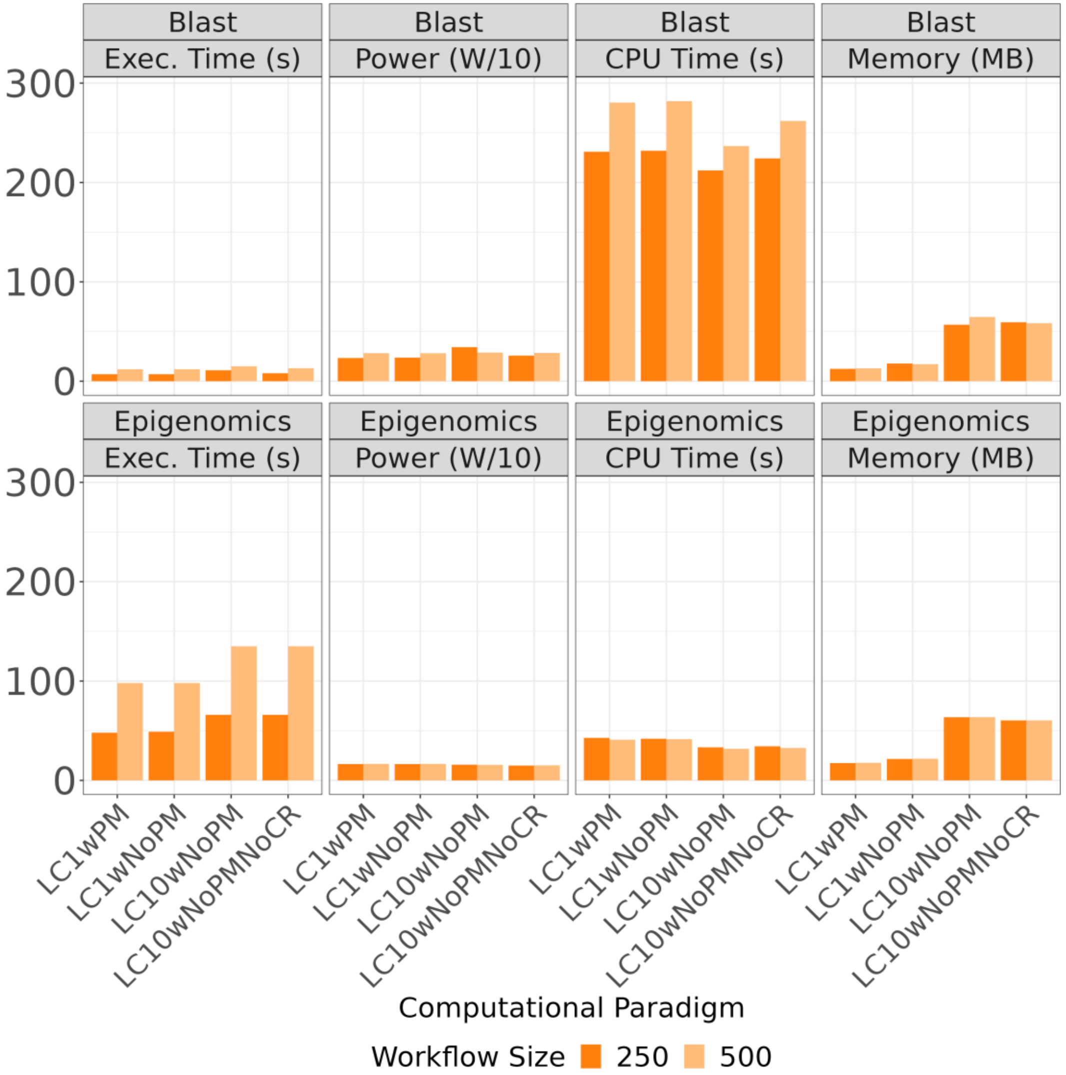


Fig. 5: Comparison between different setups for the Local Containers.

Experimental Results: Fine-grained Serverless and Local Containers

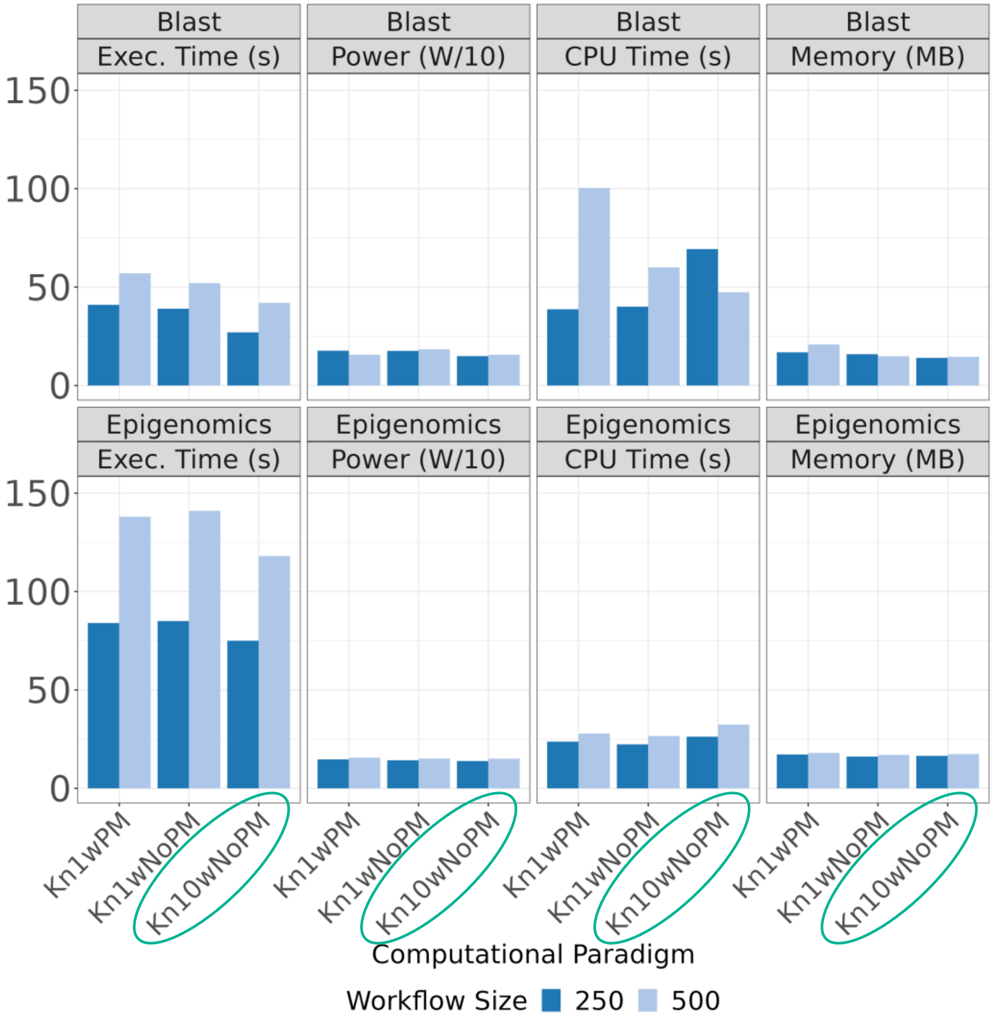


Fig. 4: Comparison between different setups for the Serverless Computational Paradigm.

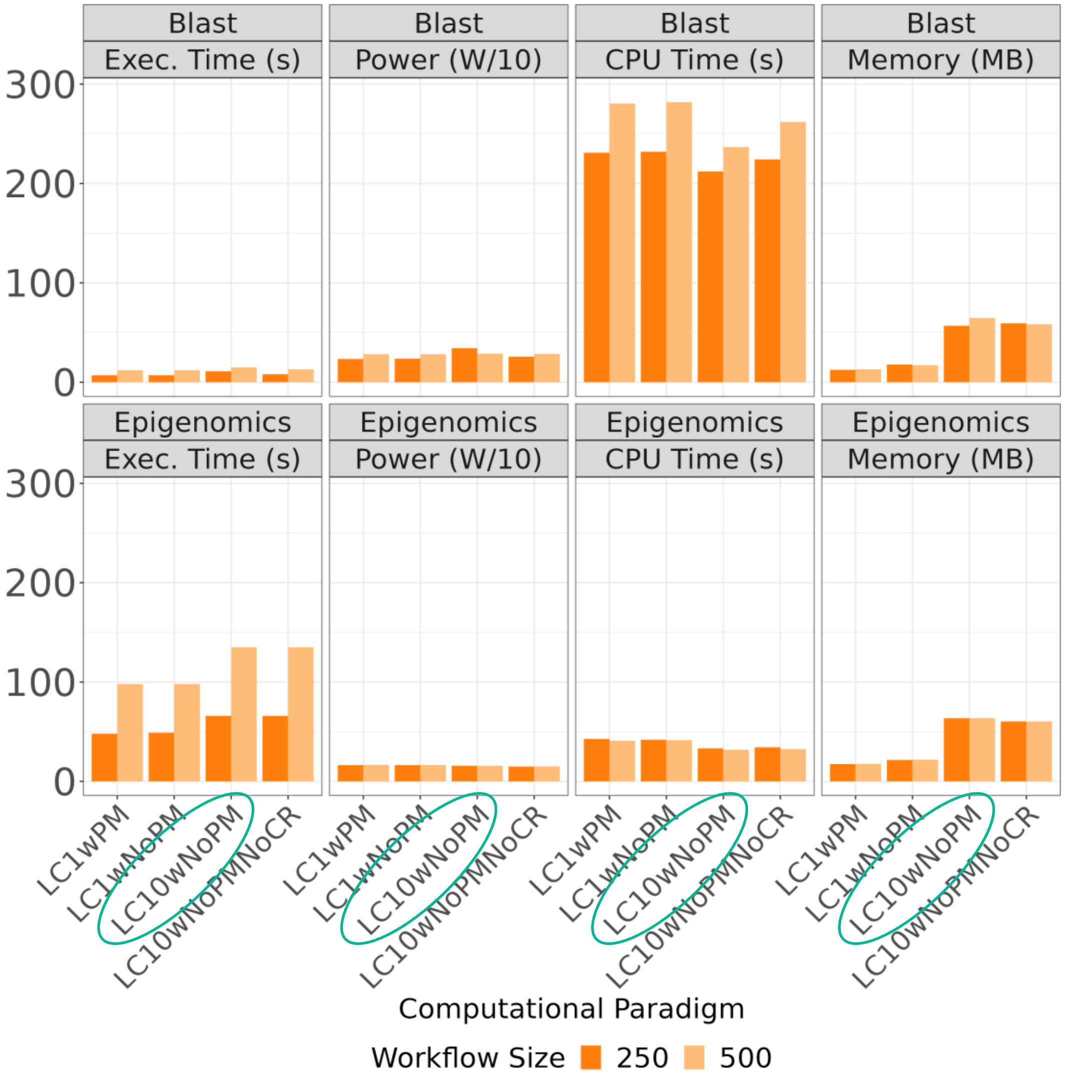


Fig. 5: Comparison between different setups for the Local Containers.

Experimental Results: Fine-grained Serverless and Local Containers

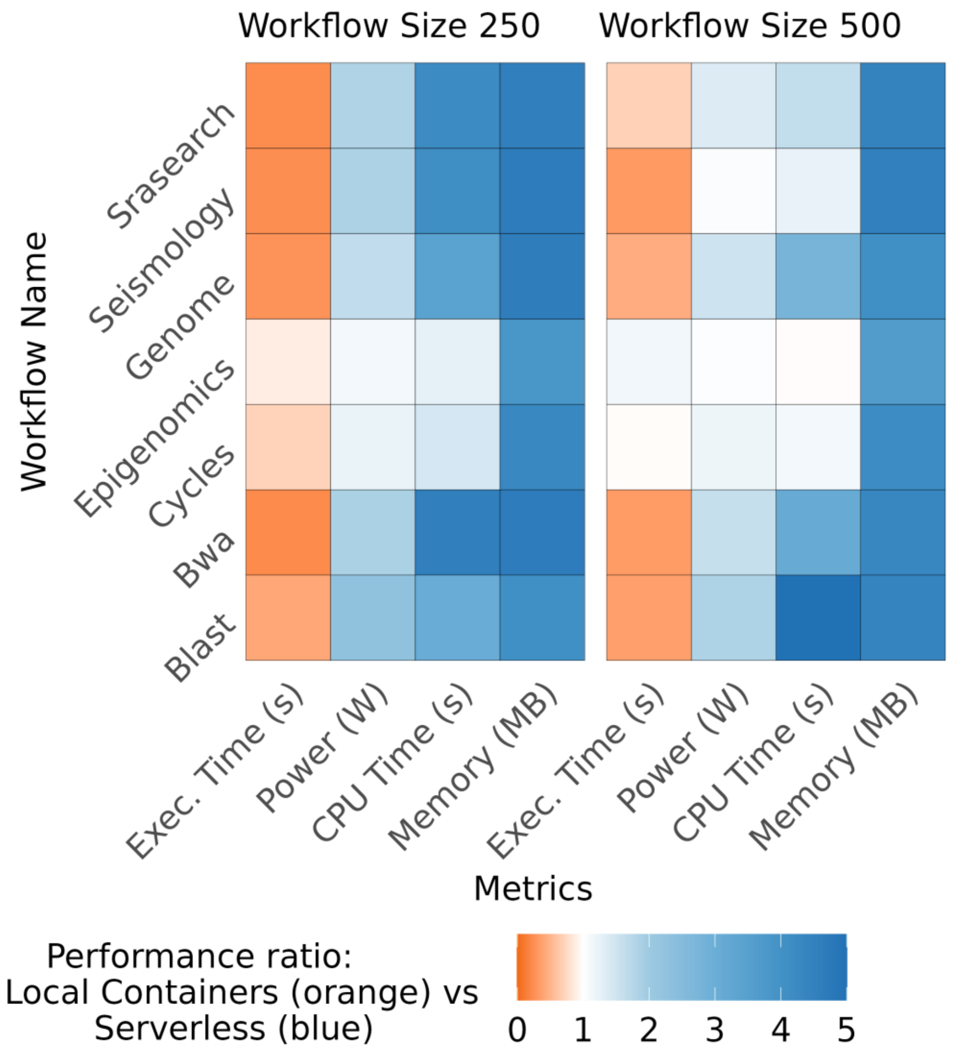
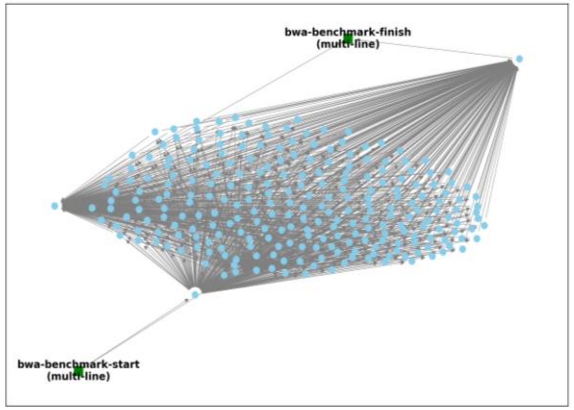


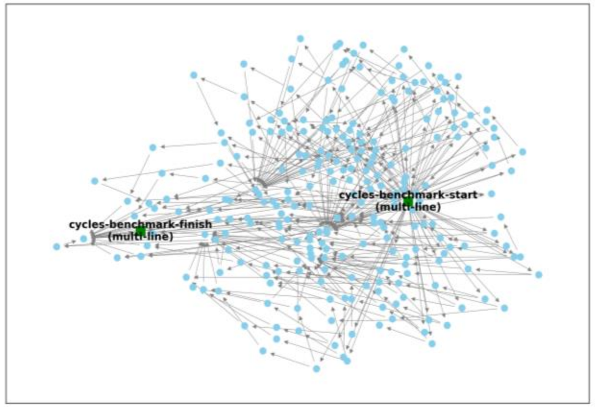
Fig. 7: Comparison between all workloads in best setups for Serverless and Local Container Computational Paradigms.

Experimental Results: Towards Workflow Characterization

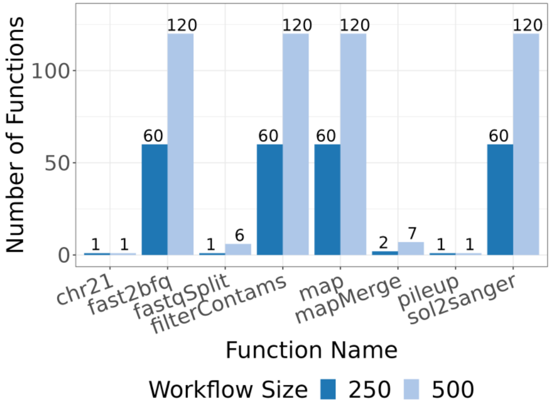
Group A (Dense DAGs)

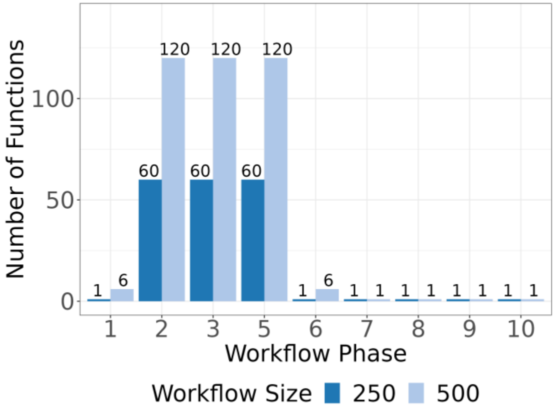


Group B (Sparse DAGs)



Dag Decompositions





Workflows	Common Performance
Blast, BWA, Genome, Seismology, SraSearch (Dense)	• Gain up to 5x in CPU and Memory Usage • Gain in Power Usage by 1 - 2x • Lose to bare-metal local containers for execution time by -1x
Cycles, Epigenomics (Sparse)	• Gain up to 5x in Memory Usage • Match or slightly gain in Power and CPU. • Match or slightly lose to bare-metal local containers for execution time.

Anderson Andrei Da Silva, R. P. Hong Enriquez, G. Rattihalli , V. Thurimella, R. Ferreira da Silva, and D. Milojicic ,
Enabling HPC Scientific Workflows for Serverless, CANOPIE-HPC at SC24

Conclusions (1) – The Framework

- We propose a framework for **executing** and **evaluating HPC scientific workflows** on **serverless** platforms:
  - a **serverless workflow manager** and
  - an **extension of the WfCommons framework** to translate traditional workflows into **serverless-compatible versions**.
- We empower researchers with new tools to study the performance of various HPC scientific workflows across multiple metrics, and scenarios

Conclusions (2) - Function's Granularity and Auto-scaling

- We evaluate **coarse-grained** scenarios for serverless.
  - The management of CPU and memory is simpler:
    - We can evaluate bigger workflows;
  - The results from serverless are closer to the bare-metal local containers;
  - However, the resource usage is not optimal for serverless as it is when using fine-grained resources.

- **Fine-grained** resources management and **auto-scaling** result in **better resource utilization**, however, we highlight that **it is more challenging**:
  - New processes can become either empty or underutilized due to mismatches between the action of creating new processes and finishing older processes;
    - More resources are used, and limits of memory and CPU may be reached.
  - Investigating different combination of parameters, we can do a better use of it.
    - We achieved good trade-offs of resource usage and execution time.

Conclusions (3) - Serverless for HPC/AI Scientific Workflows

- HPC scientific workflows can **significantly benefit from serverless** in terms of resource efficiency (CPU, memory, and power) while maintaining performance levels close to traditional execution times.
    - Still, not all evaluated workflows showed these benefits uniformly.

- We should recall that workflows can be composed of different steps and types of functions, not all of them are necessarily ideal for serverless execution.

- Therefore, it is likely that in some cases, a mapping of different execution paradigms per workflows might be a better choice.
    - The optimal strategy for complex workflows might be **combining executions** on **serverless and bare-metal local containers** for different tasks or groups of tasks.

Future Work

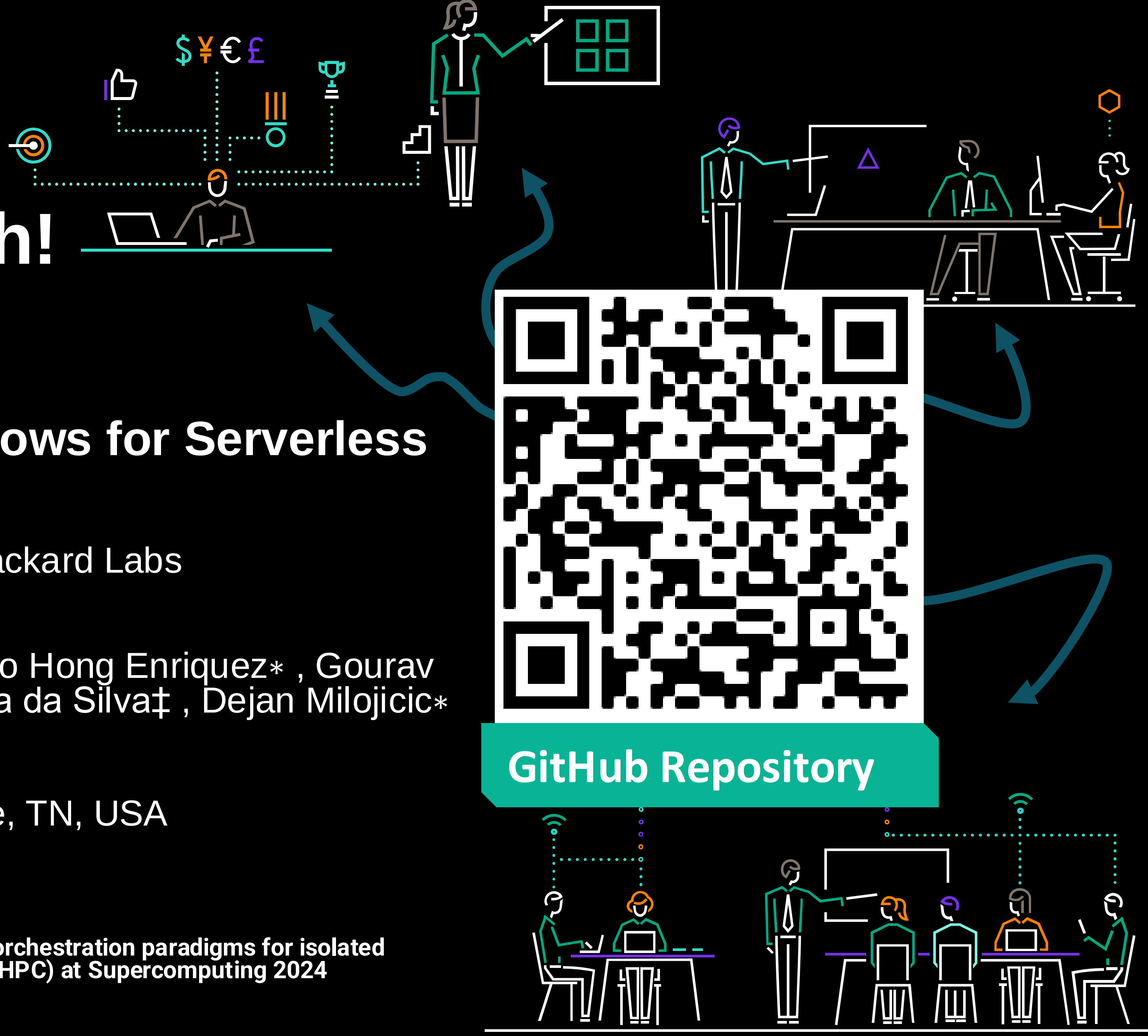
- We intend to leverage this study and **include more aspects of serverless**, by:
  - investigating the impacts of using **external distributed data storage;**
  - studying the impacts of serverless on **multi-workflows** invocation and **multi-cluster** scenarios.

- In addition to Knative, we will **explore other serverless platforms**, such as Globus Compute, AWS Lambda, Google Cloud Functions, and Azure.

- Finally, we expect that all these directions can lead us to the **characterization of HPC scientific workflows on serverless.**

Hewlett Packard
Labs

Thank you very much!
Any questions?



GitHub Repository

Enabling HPC Scientific Workflows for Serverless

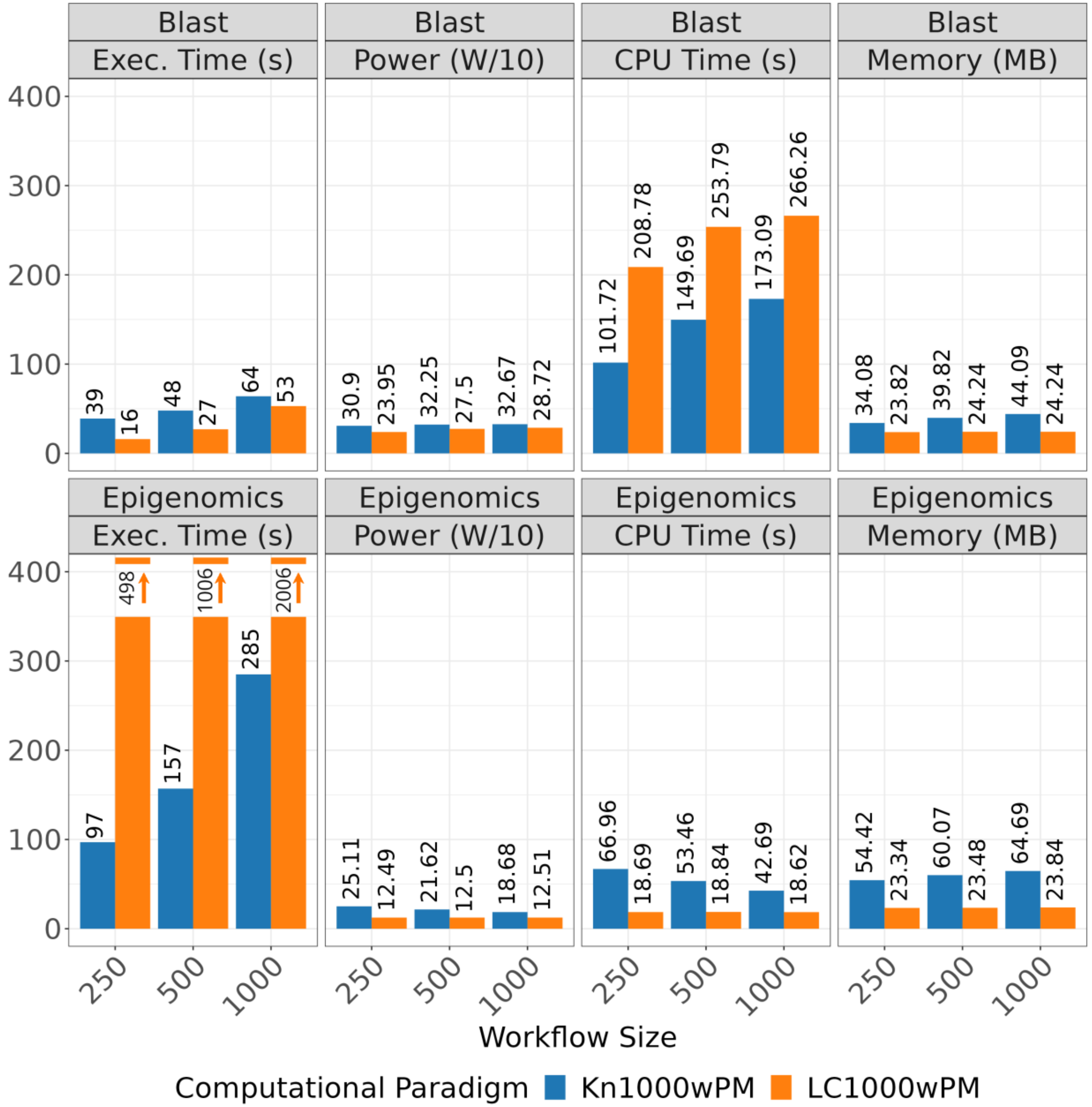
Anderson Andrei Da Silva
Postdoctoral Research Engineer, Hewlett Packard Labs

Anderson Andrei Da Silva∗ , Rolando Pablo Hong Enriquez∗ , Gourav Rattihalli∗ , Vijay Thurimella∗, Rafael Ferreira da Silva‡ , Dejan Milojicic∗

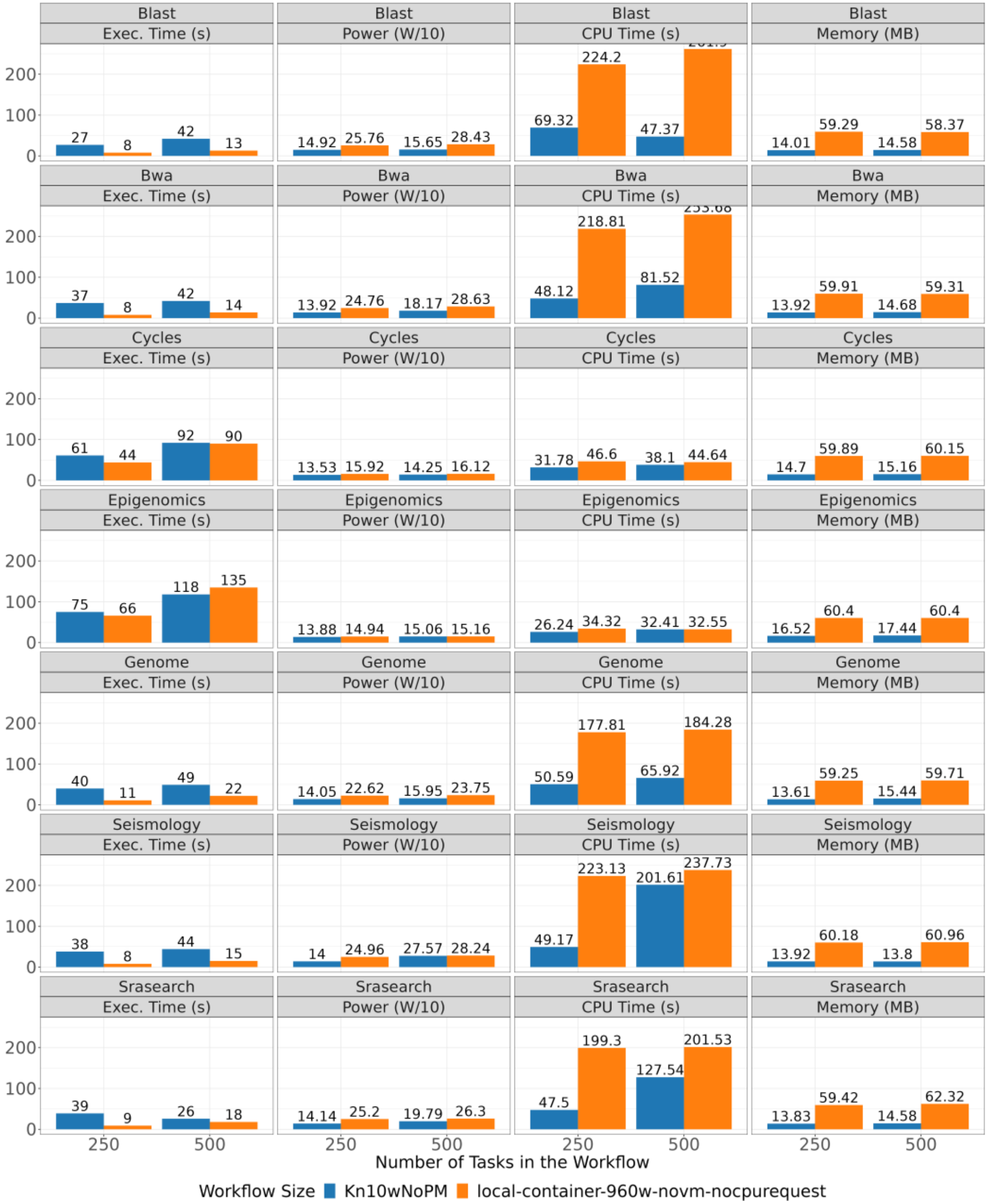
∗ Hewlett Packard Labs, Milpitas, CA, USA
‡ Oak Ridge National Laboratory, Oak Ridge, TN, USA

6th workshop on Containers and new orchestration paradigms for isolated environments in HPC (CANOPIE-HPC) at Supercomputing 2024

Experimental Results: Coarse-grained executions

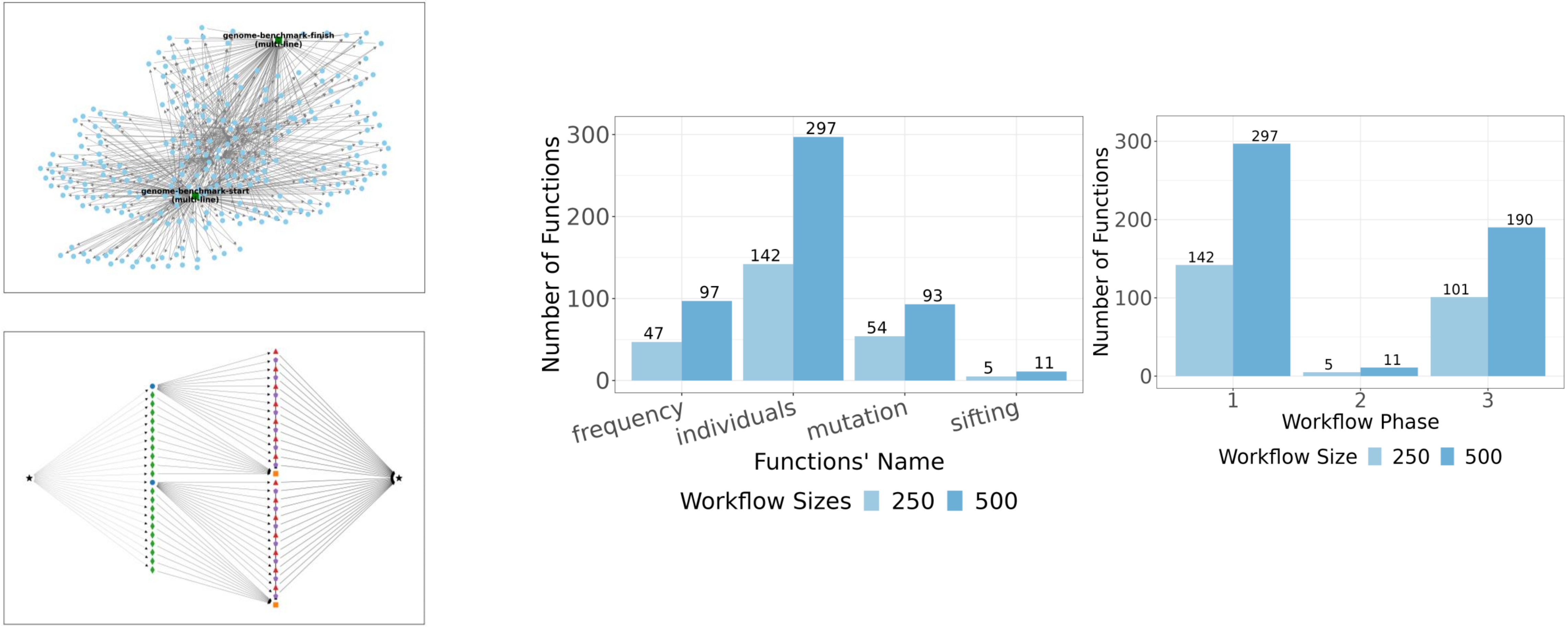


Experimental Results: Fine-grained executions



Experimental Results: Towards Workflow Characterization

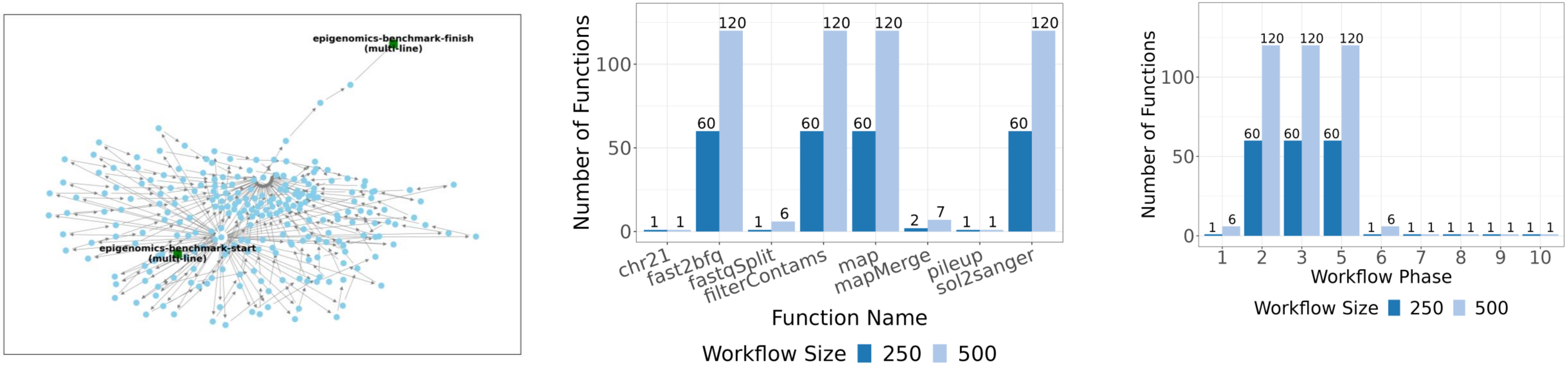
Fig. 3: Different workflows, its phase density in number of in functions, and its composition in function’s name and quantity.



Anderson Andrei Da Silva, R. P. Hong Enriquez, G. Rattihalli , V. Thurimella, R. Ferreira da Silva, and D. Milojicic ,
Enabling HPC Scientific Workflows for Serverless, CANOPIE-HPC at SC24

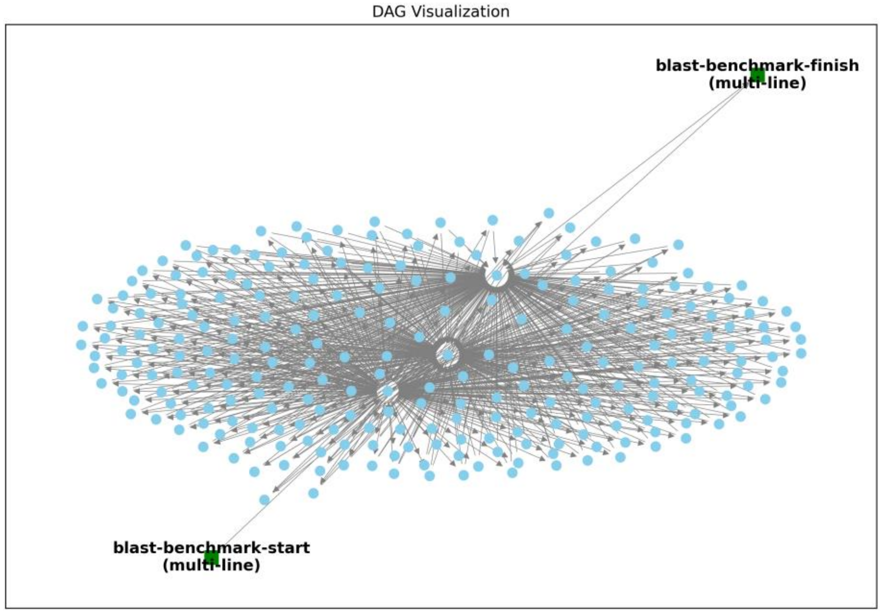
Experimental Results: Towards Workflow Characterization

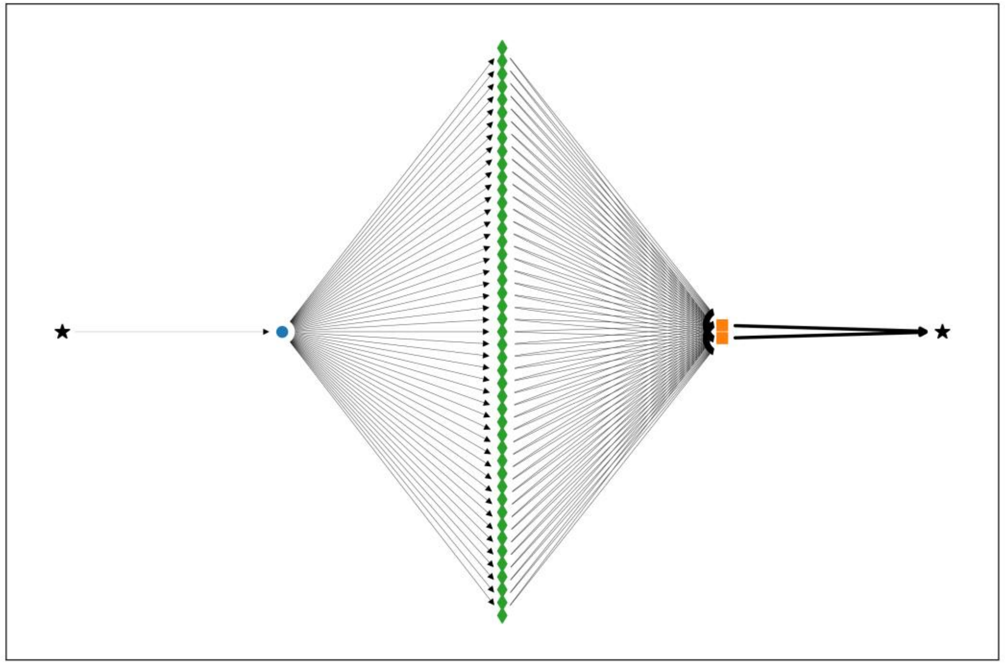
Fig. 3: Different workflows, its phase density in number of in functions, and its composition in function's name and quantity.



Anderson Andrei Da Silva, R. P. Hong Enriquez, G. Rattihalli , V. Thurimella, R. Ferreira da Silva, and D. Milojicic ,
Enabling HPC Scientific Workflows for Serverless, CANOPIE-HPC at SC24

Workflow characterization for serverless (4)





Anderson Andrei Da Silva, R. P. Hong Enriquez, G. Rattihalli , V. Thurimella, R. Ferreira da Silva, and D. Milojicic ,
Enabling HPC Scientific Workflows for Serverless, CANOPIE-HPC at SC24