

# ArmoniK : An Open-Source Solution for Computation Orchestration and Distribution



Jérôme Gurhem -

Wilfried Kirschenmann - Aneo, Boulogne-Billancourt, France

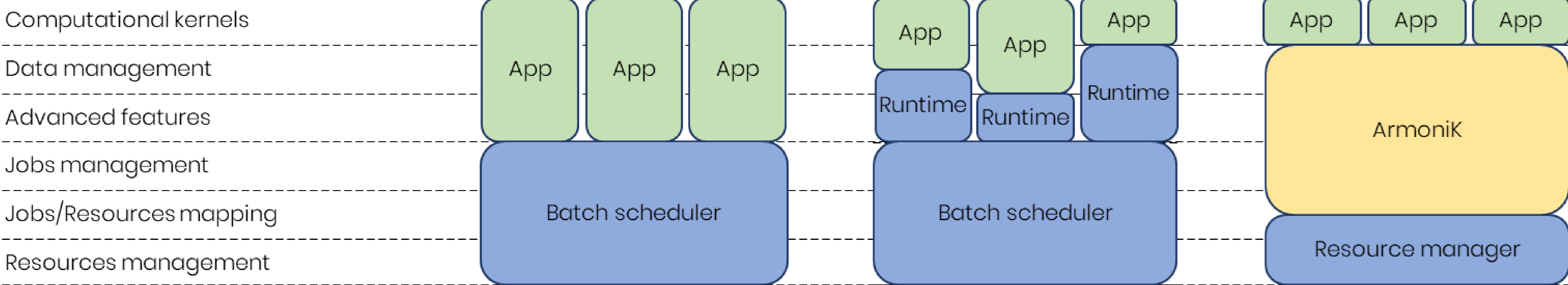
## Context

In a world of ever-growing needs for High-Performance Computing (HPC) and massive data processing, **ArmoniK** provides an Open-Source, scalable platform for executing distributed workloads efficiently on heterogeneous infrastructures.

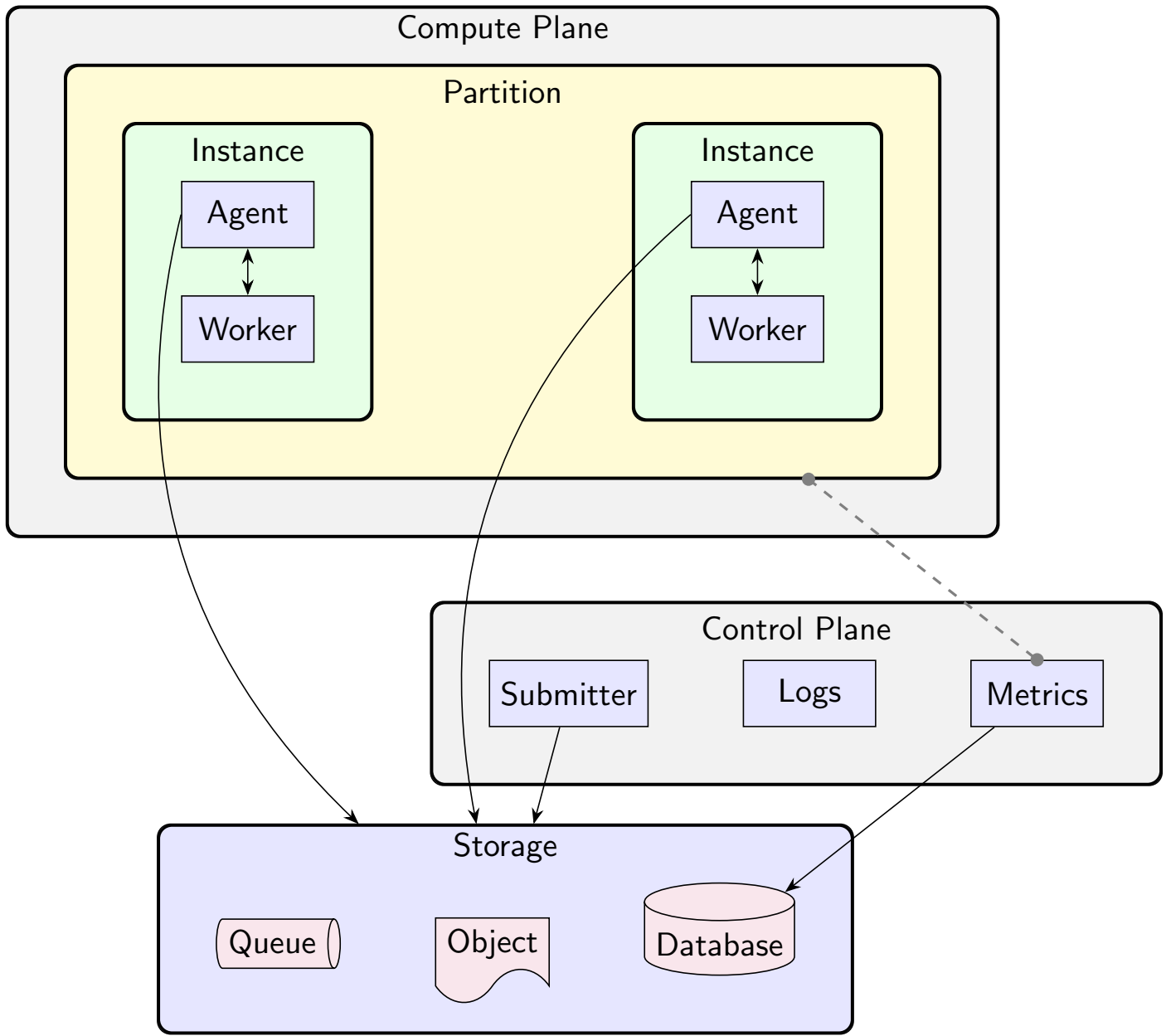
## Objectives

- ▶ Simplify the development and deployment of distributed computing applications
- ▶ Maximize resource utilization across private/public clouds and HPC clusters
- ▶ Provide a high-level abstraction for developers

## ArmoniK Positionning in HPC

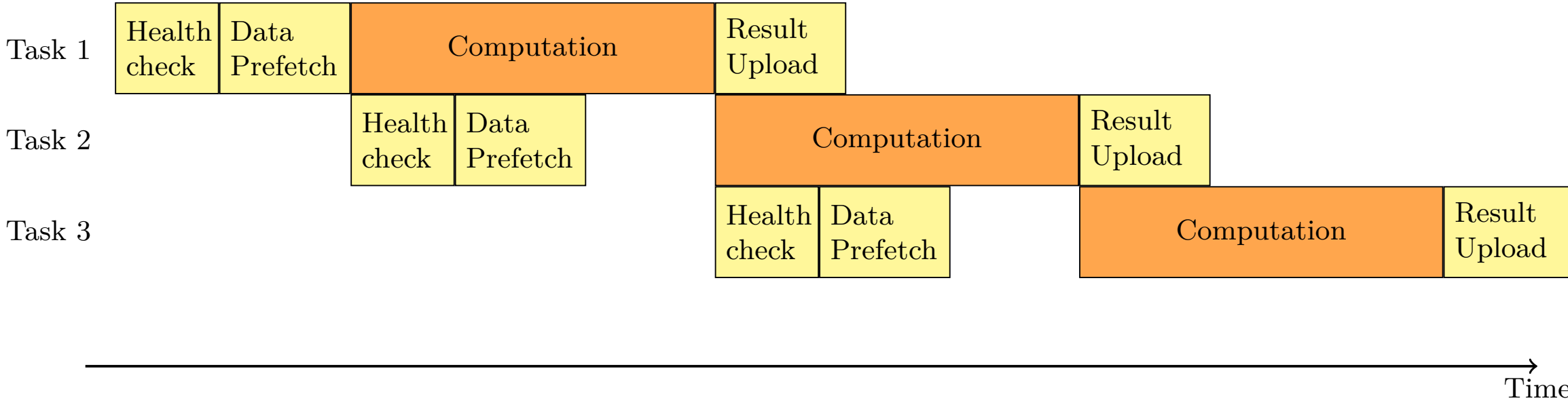


## Simplified Architecture



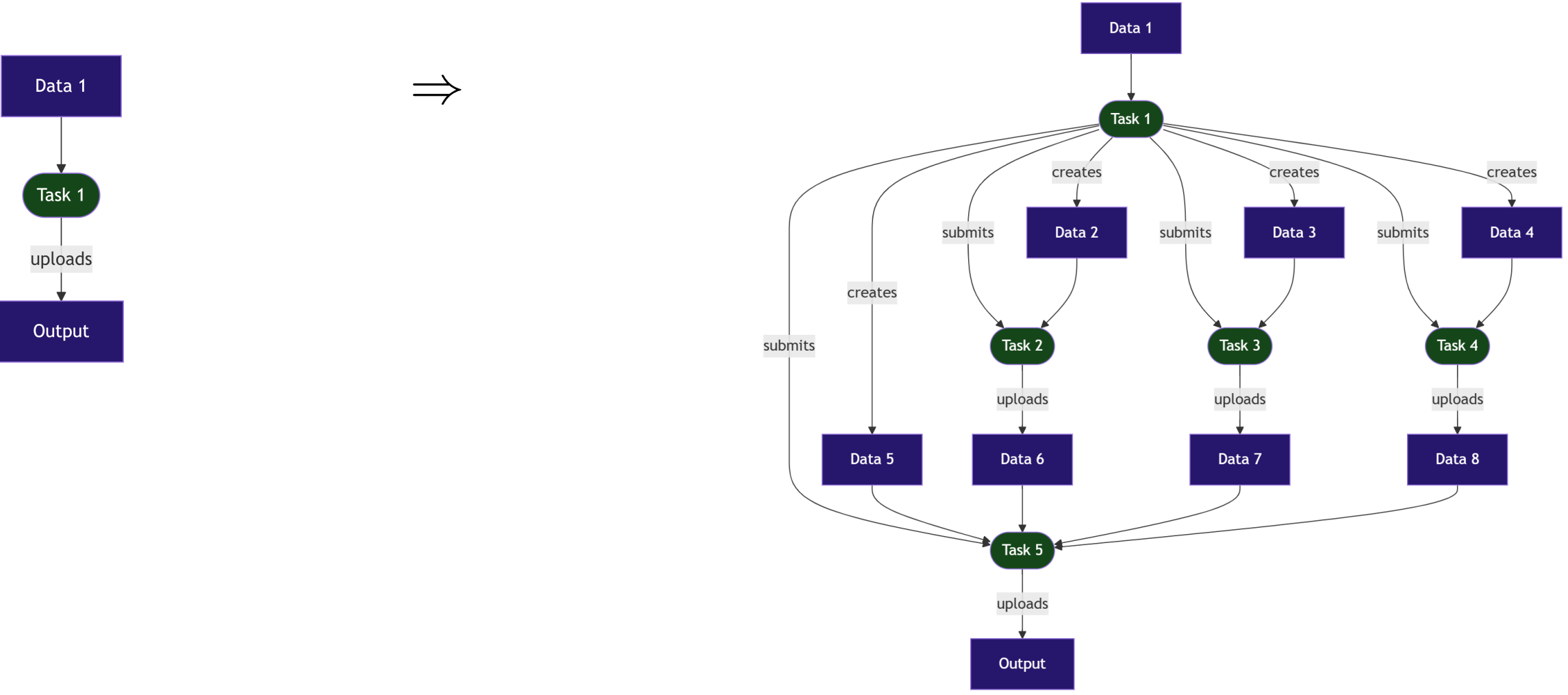
## Computations/Comm Overlapping

- ▶ ArmoniK is responsible for tasks input and output data management
- ▶ Allows for automatic communication + scheduling/task execution overlapping
- ▶ Automatic Uncoordinated Checkpointing
- ▶ Data communications through global storage



## Dynamic Graph

- ▶ Dependency graph is not fully known when scheduling starts
- ▶ Submissions can happen anytime
- ▶ Tasks can submit new tasks
- ▶ Tasks can delegate the production of their output to their new tasks

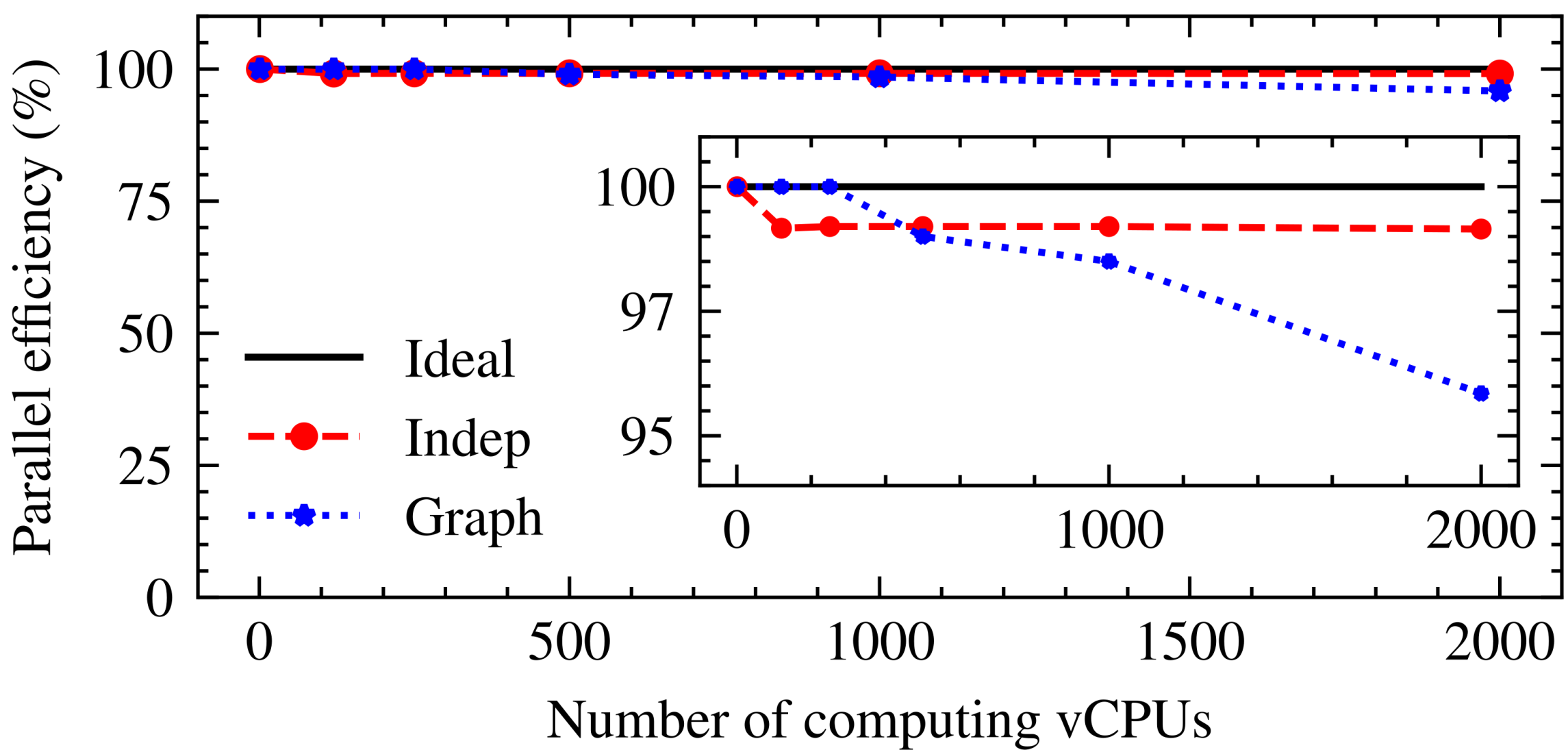


## Main features

- ▶ **Observability:** GUIs, CLIs, monitoring APIs, metrics, logs, and traces to understand of the state of the system
- ▶ **Portability:** Easy to transfer an application from one environment to another
  - ▶ Officially supported languages: C#, C++, Python, Rust, Java, and JavaScript
  - ▶ Tasks on different architectures (x86, ARM, GPU, Linux, Windows), applications, environments
- ▶ **Fault Tolerance:** Functions without interruption even when one or more nodes fail
  - ▶ Allow support for preemptible computing resources
  - ▶ Error management at the task level
- ▶ **Malleability:** Support dynamic reconfiguration of the number of allocated resources during execution without interruption
- ▶ **Resource Sharing:** Allows sharing resources between applications to execute as many as possible at the same
- ▶ **Modularity:** Modules can be swapped without modifying ArmoniK's code to suit user needs and constraints

## Performance & Scalability

- ▶ Efficient task retry on failure
- ▶ Load-aware scheduling
- ▶ Linear scalability on real workloads
- ▶ Optimal resource usage on hybrid clusters
- ▶ Indep : independent tasks workload
- ▶ Graph : nested fork-join workload



## Conclusion

- ▶ ArmoniK simplifies the development of distributed computing applications.
- ▶ It ensures efficient execution on clouds and HPC clusters through smart orchestration.
- ▶ Developers benefit from a high-level abstraction and multi-language SDKs.
- ▶ Its modular, scalable architecture adapts to changing workloads.
- ▶ Integrated observability guarantees reliability and performance.
- ▶ ArmoniK enables the next generation of high-performance, data-intensive computing.