Scalable Programming Models and Strategies for Efficient High-Performance Serverless in Hybrid and Heterogeneous Systems

Valerio Besozzi
Department of Computer Science
University of Pisa
Pisa, Italy

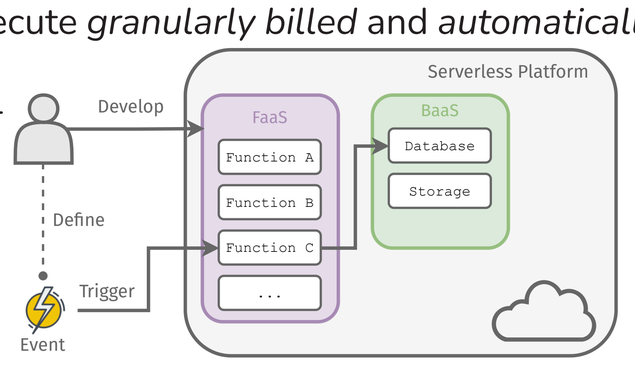
SCAN ME

Problem Statement

- **Problem:** Current HPC infrastructures are constrained by strict resource allocation systems and rigid programming models.
- **Idea:** The serverless execution model enables fine-grained application decomposition, leading to:
  - Improved resource utilization.
  - Easier deployment of highly distributed parallel applications.
  - *Statelessness* allows for virtually unlimited horizontal scaling.
- **Challenges:** In order to effectively exploit the serverless paradigm to support HPC applications, several challenges must be addressed:
  - Lack of high-level parallel programming models suited for serverless.
  - Complexities in managing highly parallel workloads and support function composition in serverless environments.
  - Limited support for accelerators (e.g., GPUs, FPGAs).

Serverless Computing

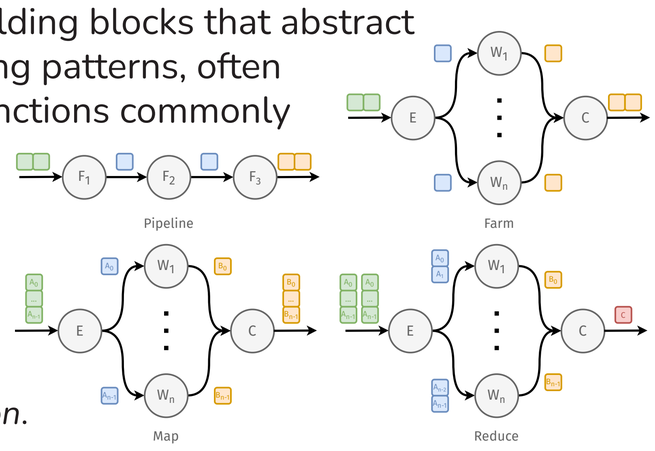
- **Serverless Computing:** It is a form of cloud computing that allows users to deploy and execute *granularly billed* and *automatically scaled* applications, without having to address the underlying operational logic.



- The typical workflow of a serverless application consists of the following steps:
  1. A pre-defined event triggers a serverless function that was bound to it earlier.
  2. The serverless platform prepares the *execution environment* for the triggered function to run.
  3. After the execution is completed, the serverless platform releases the resources previously acquired.
- The execution environment typically relies on *containers* or other forms of *lightweight virtualization*.

Skeletal Programming

- **Algorithmic Skeletons:** Introduced by Murray Cole in the late 1980s, algorithmic skeletons are building blocks that abstract common parallel programming patterns, often derived from higher-order functions commonly found in functional programming.



- **Key features:**
  - *Separation of Semantics and Implementation.*
  - *Correctness by Construction.*
  - *Abstraction of low-level parallelism mechanisms.*
- **Transformation Rules:** Different skeleton compositions can represent the same computation, but exploit parallelism differently. Transformation rules define semantic equivalences that help developers identify optimal compositions in terms of performance.

Research Methodology

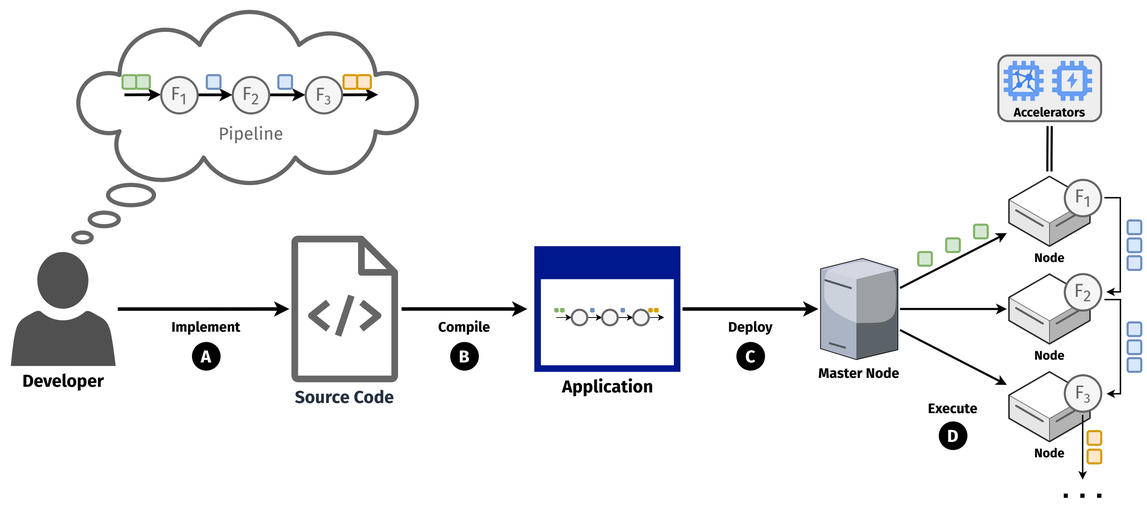
Main Objective:
To develop a novel *methodology* for the development and execution of parallel and distributed applications that leverage the serverless execution model, targeting hybrid HPC/Cloud infrastructures. In order to achieve that, we are currently addressing the following objectives:

Goal A: Skeleton-based Programming Model
- Develop a high-level parallel programming model based on algorithmic skeletons, tailored for serverless.
- Skeletons provide abstraction for parallelism, simplifying scaling and processes coordination.

Goal B: High-Performance Serverless Framework
- Design a lightweight, high-performance serverless framework optimized for parallel workloads.
- Prioritize low resource footprint and efficient communication mechanisms.

Goal C: Integration of Heterogeneous Accelerators
- Develop a high-level interface for seamless offloading of compute-intensive tasks to accelerators (e.g., GPUs, FPGAs).
- Enable efficient use of accelerators in serverless platforms.

Proposed Workflow:



A The user design a parallel application in terms of skeletons.

B The application code is compiled into a suitable format.

C The application is deployed on the high-performance serverless framework.

D The framework executes the application and manages function instantiations.

Preliminary Results and Next Steps

- **First Results:**
  - Conducted a survey on literature related to virtualization approaches and accelerator support in serverless architectures.
  - Started implementing a framework prototype using unikernels and leveraging *Cloud-Hypervisor* and *Firecracker* for lightweight virtualization.

- **Next Steps:**
  - Formalize the design of the proposed skeleton-based parallel programming model.
  - Start experimenting with the proposed programming model and prototype high performance serverless framework.
  - Add support to performance modeling through algorithmic skeleton *transformation rules*.

PUBLICATIONS

Besozzi, Valerio. *"**PPL: Structured Parallel Programming Meets Rust**"*. In *2024 32nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 78-87, 2024. doi:10.1109/PDP62718.2024.00019.

Besozzi, Valerio, and Patrizio Dazzi. *"**Boosting Serverless Computing: A Survey on Architecture Designs and Accelerator Support for Serverless Platforms**"*. In *2024 30th International European Conference on Parallel and Distributed Computing (Euro-Par 2024)*. Accepted for publication.

MORE INFORMATION

Valerio Besozzi
University of Pisa
Department of Computer Science
Parallel Programming Models Group

valerio.besozzi@phd.unipi.it