



# Distributed futures for efficient data transfer between parallel processes

Pierre Leca<sup>1,3</sup>, Ludovic Henrio<sup>2</sup>, Françoise Baude<sup>3</sup>, and Wijnand Suijlen<sup>1</sup>

pierre.leca@huawei.com ludovic.henrio@cnsr.fr francoise.baude@univ-cotedazur.fr wijnand.suijlen@huawei.com

<sup>1</sup> Huawei Technologies – Paris, France

<sup>2</sup> Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP – Lyon, France

<sup>3</sup> Université Côte d'Azur, CNRS, I3S – Sophia Antipolis, France



## Context

- ▶ A *future* is a placeholder for a value being computed by a task
- ▶ *Futures* are useful for coordinating parallel tasks
- ▶ Data parallelism can speed up a task by splitting the input data
- ▶ Results from data parallel tasks are often distributed

## Distributed futures

- ▶ A *distributed future* represents a *distributed vector*
- ▶ Synchronizing a distributed future gets the metadata representing the vector distribution after it is produced
- ▶ Processes consuming distributed data request them from the producing processes

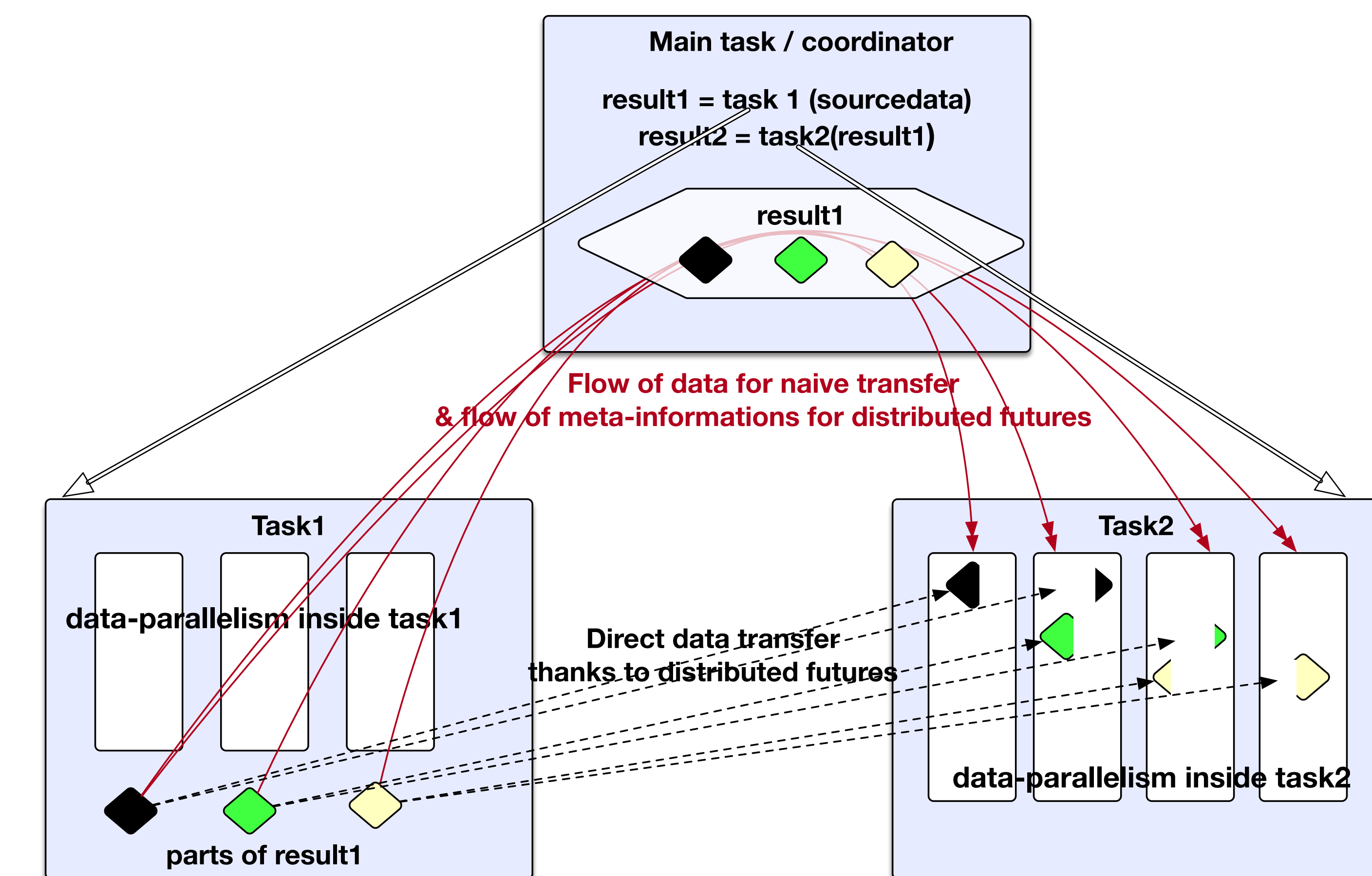


Figure 1: Distributed vector communication

## Problems

- ▶ Need to gather distributed output (future) of a parallel task
- ▶ Result often scattered again into another parallel task

## Solution

Manage the communication of distributed data through *distributed futures*

- ▶ Future data distributed among parallel processes
- ▶ Distributed data communicated in parallel between tasks

## Implementation

- ▶ Distributed future implementation with C++ ActiveBSP framework [1]
  - ▷ Active objects for task parallelism
  - ▷ BSP data parallelism inside tasks
- ▶ *Worker threads* execute user code
- ▶ *Management threads* communicate with other objects

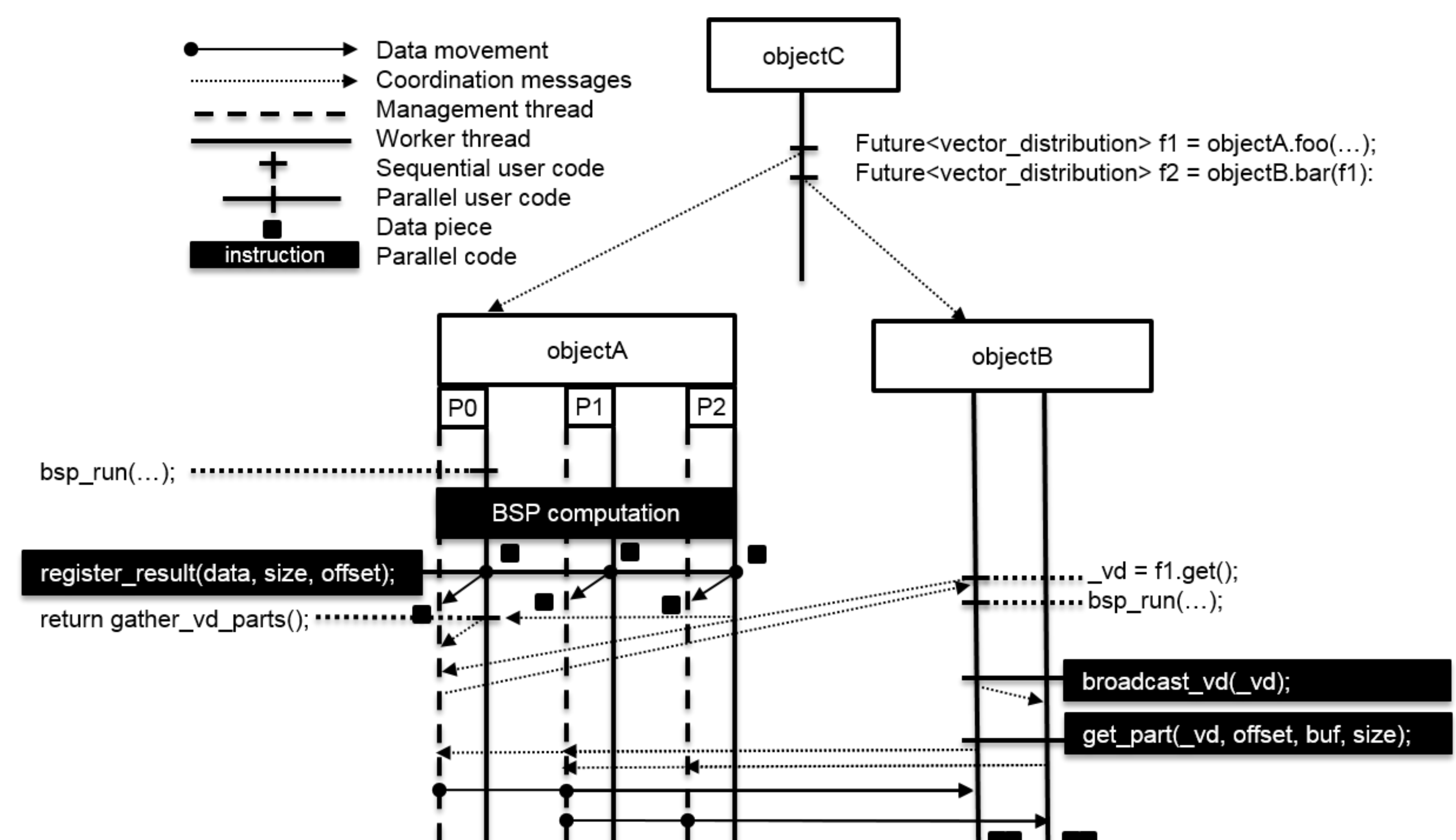


Figure 2: ActiveBSP implementation

## Experiments

**Scenario**

- ▶ Pipeline of 3 parallel objects
- ▶ Parallel image compressor as middle object
- ▶ Coordinator process wires the pipeline through future parameters

**Execution**

- ▶ 20 processes on first and last object
- ▶ 1000 images inserted into pipeline

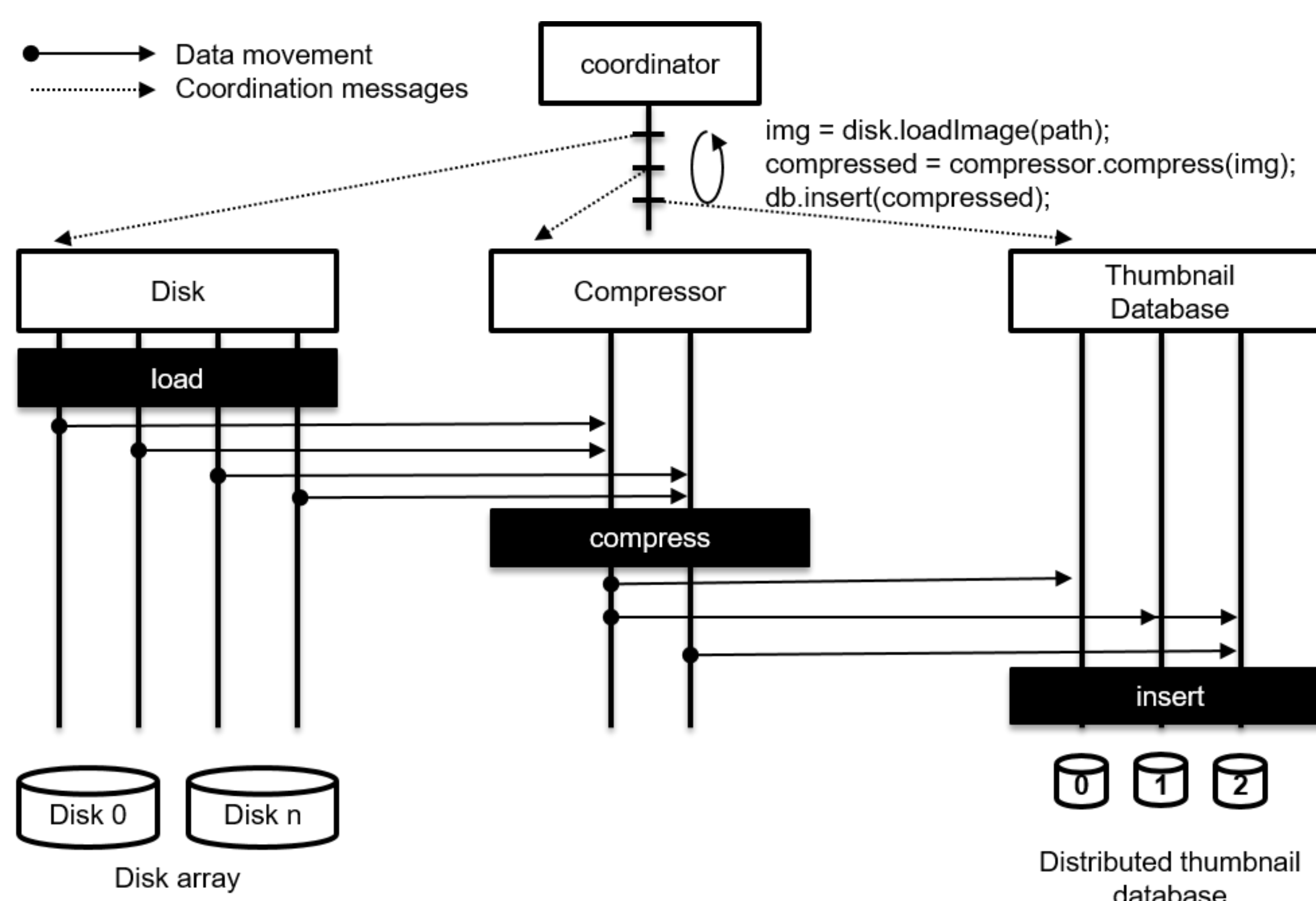


Figure 3: Experiment scenario

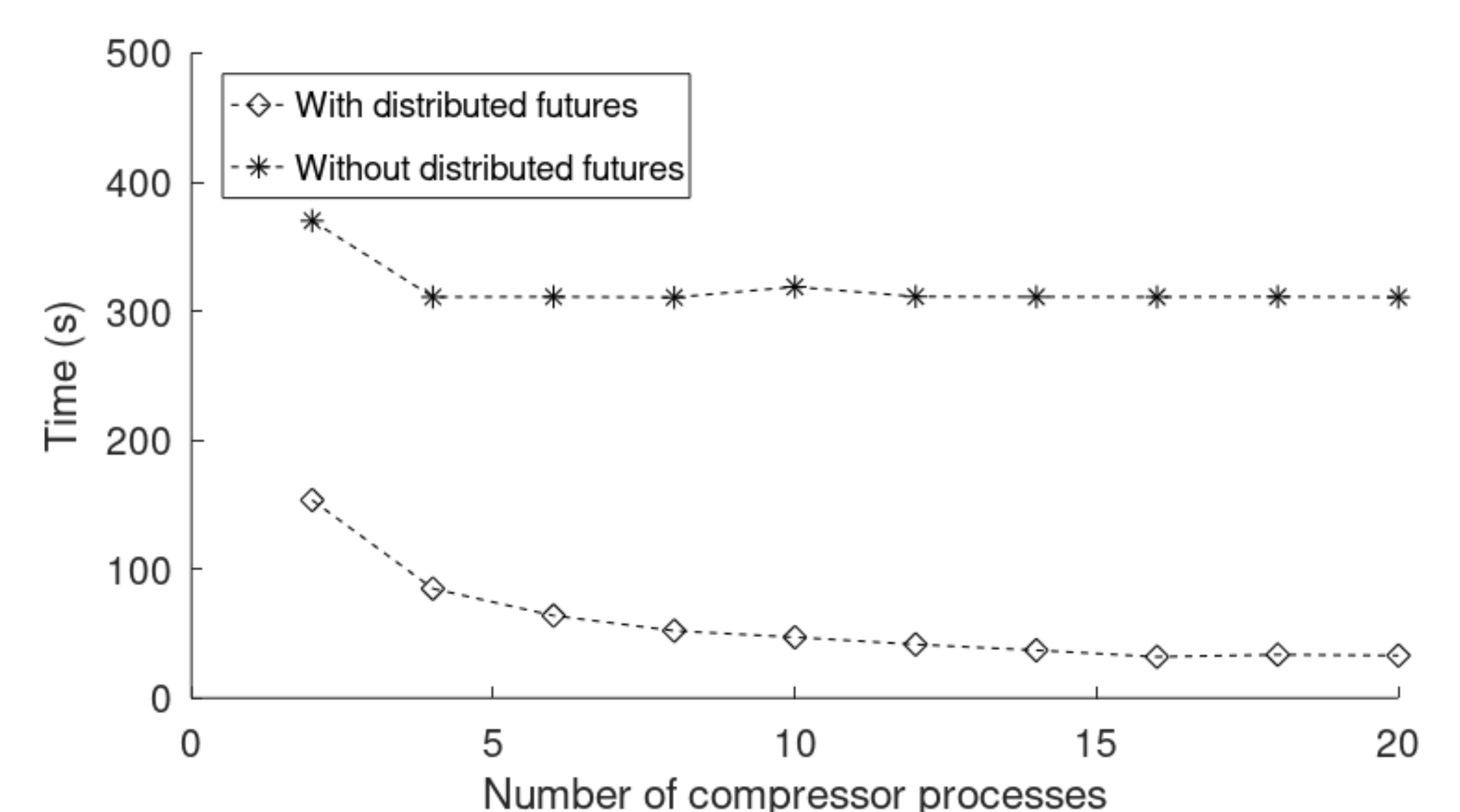


Figure 4: Varying the number of compressor processes

## Conclusion

- ▶ Increase in performance compared to standard futures

[1] G. Hains, L. Henrio, P. Leca, and W. Suijlen, "Active objects for coordinating bsp computations (short paper)," in *Coordination Models and Languages*, G. Di Marzo Serugendo and M. Loreti, Eds. Springer International Publishing, 2018, pp. 220–230.