PADERBORN UNIVERSITY

Supervised Distributed Computing

Project Group 2026/2027

Henning Hillebrandt, Jan Lüking, Christian Scheideler

Motivation: Volunteer Computing

Outsourcing massive computations to a global network of volunteers

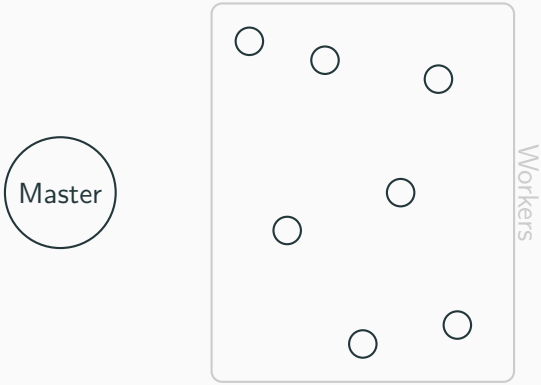
SETI@home

Analyzes radio telescope data to search for potential extraterrestrial signals

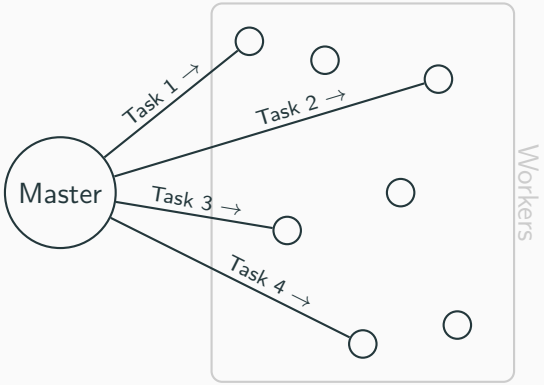
Folding@home

Simulates protein folding to understand diseases and develop new treatments

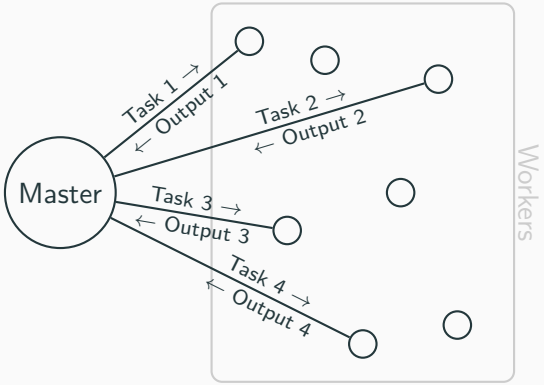
Master-Worker Approach



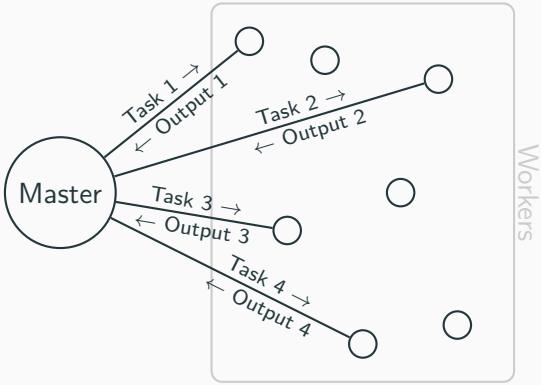
Master-Worker Approach



Master-Worker Approach

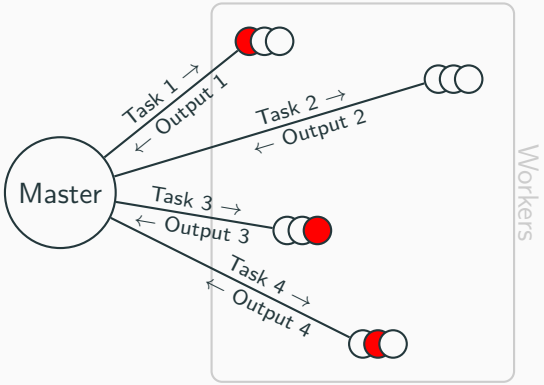


Master-Worker Approach



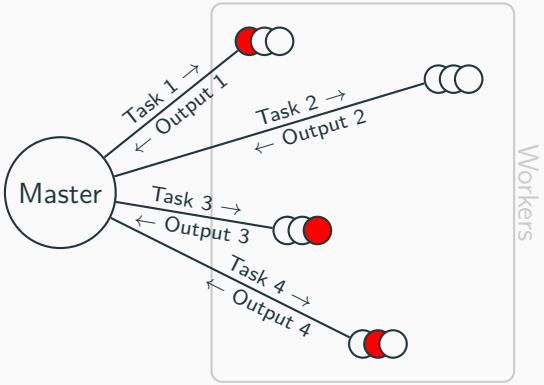
Problem: faulty / adversarial workers

Master-Worker Approach



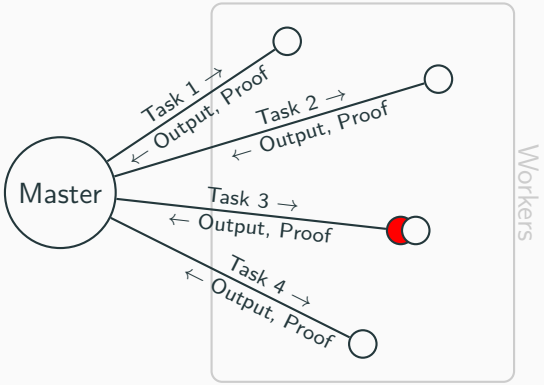
Simple solution: Assign multiple workers to each task

Master-Worker Approach



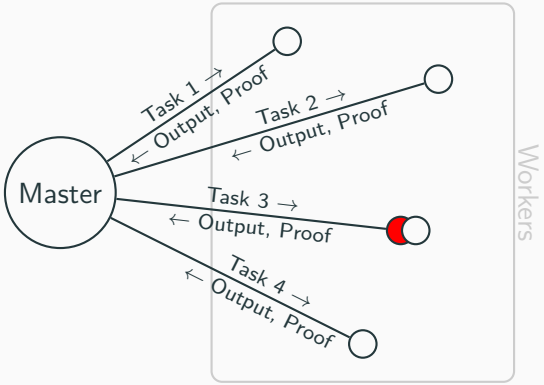
Problem: large computational overhead, only works if majority of workers is honest

Master-Worker Approach



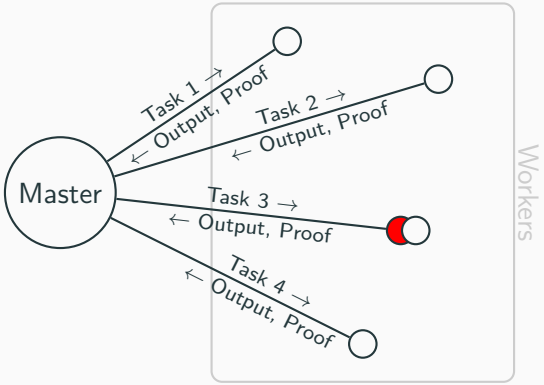
Solution: require workers to deliver output with proof that can be checked efficiently

Master-Worker Approach



Problem: Master becomes a bottleneck

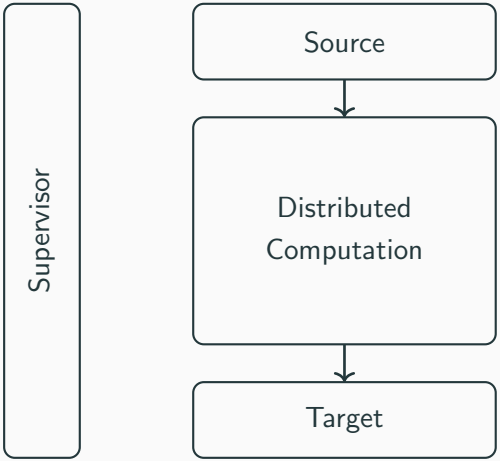
Master-Worker Approach



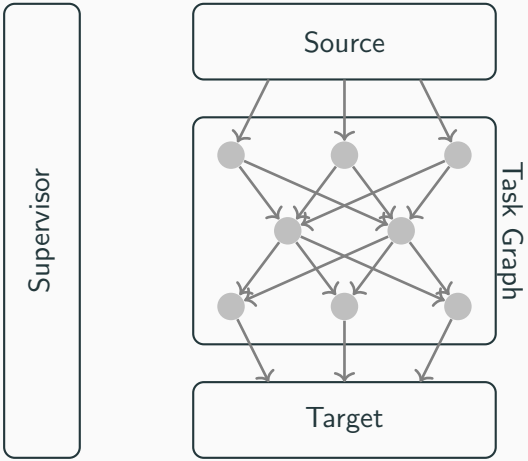
Problem: Master becomes a bottleneck

Can the verifications be offloaded to the workers?

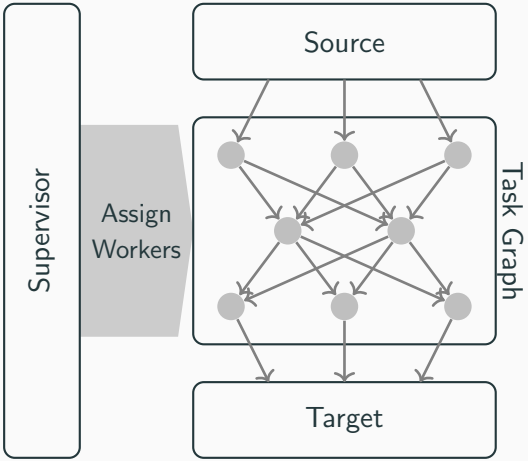
Supervised Distributed Computing



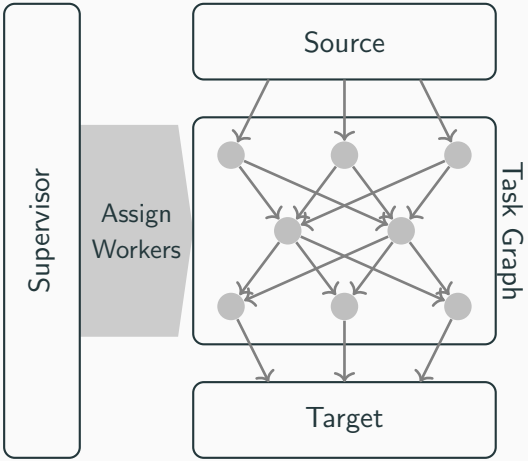
Supervised Distributed Computing



Supervised Distributed Computing

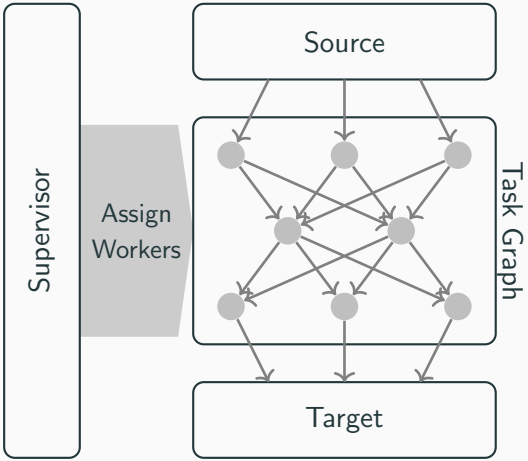


Supervised Distributed Computing



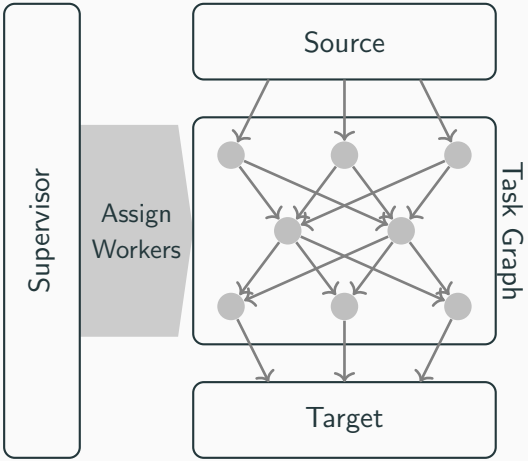
Large number of workers, β-fraction malicious, Supervisor picks u.i.r.

Supervised Distributed Computing



Supervisor never learns data! Workers verify each other's outputs

Supervised Distributed Computing



There is a strategy s.t. for any constant $\beta < 1$, the target learns the solution w.h.p.!

Goal: Developing a Supervised Distributed Computing platform

Server (Supervisor)

- Ability to upload task graphs
- Assign workers to tasks
- Send code to workers and tell them who to communicate with

Client (Workers)

- Ability to (dis-)connect with server
- Establish connection to other workers
- Verify outputs of other workers
- Execute code and send the output to other workers
- Design a UI

Structure of the Project Group

First Phase: Seminar

- Read papers about Supervised Distributed Computing
- Become familiar with client-server infrastructure

Second Phase: Implementation/Design/Analysis

- Implement and test the Supervised Distributed Computing platform
- Design your own task graphs to solve problems

Prerequirements

- Interest in developing a client-server system
- Basic programming skills
- Ability to understand scientific papers
- High motivation :)