



AntCloud

Version 1.6

Bărar-Pintea Daniel-Ioan
Chiuță Mihai-Marcel
January 11, 2024

Contents

1	Version table	2
2	Introduction	4
3	Requirements	4
3.1	Functional Requirements	4
3.2	Non Functional Requirements	4
4	Architecture	5

1 Version table

Date	Version	Notes
15/10/2023	1.0	<ul style="list-style-type: none">• Created GitHub Repository• Added first version of documentation• Created rudimentary version of a client and workstation
29/10/2023	1.1	<ul style="list-style-type: none">• Modified documentation• Modified the client and workstation code• Added the server• Added a Makefile that automates compilation
3/12/2023	1.2	<ul style="list-style-type: none">• Implemented multithread architecture on the server• Client works directly with the workstation
10/12/2023	1.3	<ul style="list-style-type: none">• Created an algorithm on the server for distributing the tasks for the threads• Created a protocol for communication between stations

17/12/2023	1.4	<ul style="list-style-type: none"> • Optimized the control flow of connection on the server • Once connected, the station are able to hold the connection
5/1/2024	1.5	<ul style="list-style-type: none"> • Changed the architecture of the server from multithread to epoll • The client is able to send a source file with arguments • The workstation can work with arguments of an executable
11/1/2024	1.6	<ul style="list-style-type: none"> • Client and Workstation are no longer connected, the connection is made through the server • All the stations connects to the main server • Resolved bugs

2 Introduction

AntCloud is a distributed processing application that enables users to execute their code on a remote server with high-performance computing resources. AntCloud is designed for users who have limited processing power on their local PC and want to leverage the benefits of cloud computing and distributed processing without incurring high costs. By using AntCloud, users can easily upload their code (along with arguments if needed), the main server will choose a suitable workstation, the workstations will compile and execute it sending back the output generated. AntCloud is a convenient and affordable solution for users who need to run complex and intensive code on a regular basis.

3 Requirements

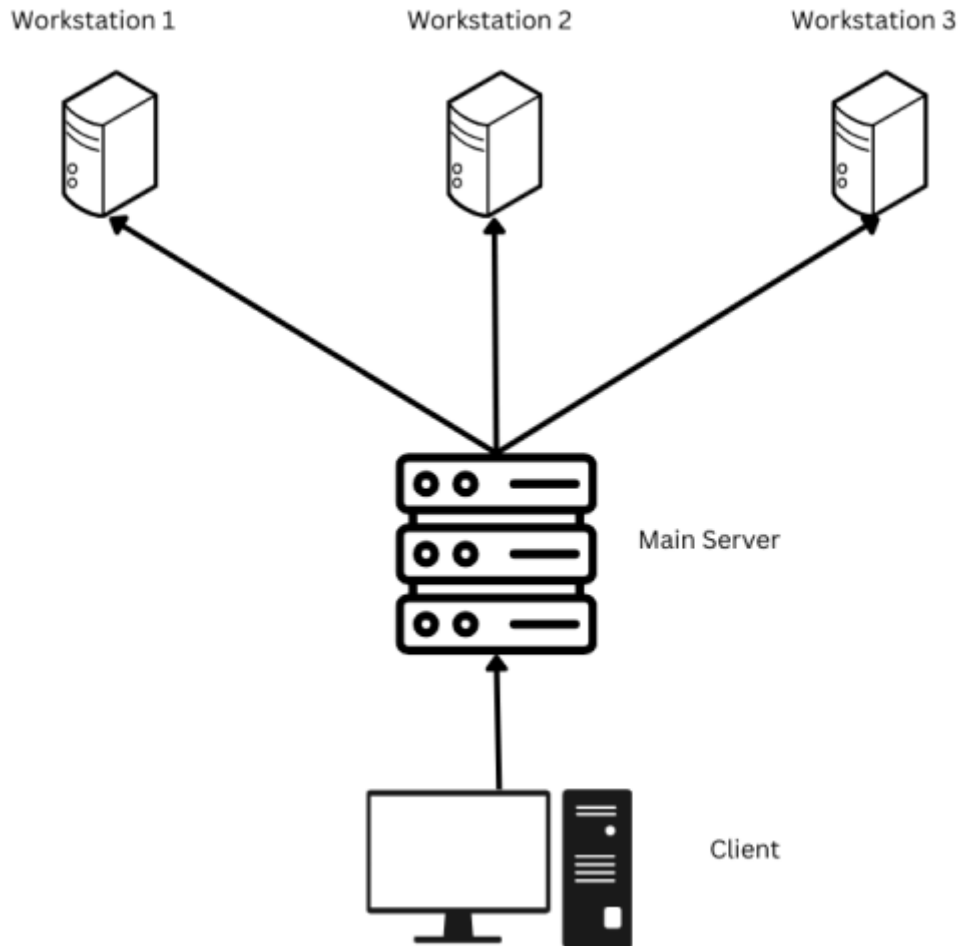
3.1 Functional Requirements

- Client sends source file to the server
- Client can send source file with arguments
- Workstation compiles the source file and executes it with the arguments (if there are any) and sends it back
- Client has a bash interface
- Client doesn't have a limit of uses

3.2 Non Functional Requirements

- Server supports multiple clients and workers
- The data sent between stations is in base64 encoding
- The server uses epoll to have a maximum efficiency
- Communication protocol between stations

4 Architecture



This application follows a master-slave architecture that consists of a central main node and multiple workstations. In this architecture, the master is responsible for coordinating and managing the overall system, while the slave nodes perform specific tasks or computations assigned by the master. For the user everything is transparent, he just needs to launch the client; in fact, a connection is made between the client and the server, from there the main server handles the data transfer. The user has at his disposal a bash interface through which he manages to perform his tasks.