

SudokuMP User Manual, RPC Version

Johan Kütt, A32401

January 2, 2018

1 Installation

The application is available from https://github.com/johankytt/SudokuMP/tree/rpc_p2p. No special installation procedures are required by the application itself.

The following libraries are required to run the application:

- The graphical user interface relies on the PySide library which is freely available both from the web and from package managers. Version 1.2.2 of the library was used for development.
- RPC over message queue is implemented using SnakeMQ. The version of the library included with the application should be used as the one available on the web contains a critical bug that renders the library non-functional under certain conditions.

The logging output shown on the command line is controlled by adjusting `level=logging.XXX` in `common/smp_common.py`. Setting it below INFO for normal use is not recommended as the console will be flooded with large amounts of messages.

2 Running the Server

The server is a purely command line application. It is started by running the script "start_server.py". The only configurable options are it's listening address and port, which are set as shown in its command line help (also shown below). If the host argument is not given, the public address of the default network interface is used. The server process can be stopped by issuing Ctrl-C on the command line.

Note: The SnakeMQ based indirect messaging system is built into the server and loaded automatically.

```
usage: start_server.py [-h] [-H HOST] [-P PORT]
```

SudokuMP Server

optional arguments:

-h, --help show this help message and exit

-H HOST, --host HOST IP or Name of the host server (default: 192.168.1.65)

-P PORT, --port PORT Listening port of the host server (default: 22122)

3 Running the Client Application

The client application is run similarly to the server from the command line by executing `python start_client.py`. The client takes no command line arguments.

On opening the application the lobby window will be shown. On the right hand side are the user controls and on the left the server game list which will be populated once a connection is made.

Before a connection is established most controls are disabled to avoid inadvertent clicking.

3.1 Connecting to the server

To be able to connect to a server, the user must fill out the player name, server address and server port fields. Once that is done the "Connect" button will be enabled and the user can proceed with establishing the connection.

Note: In this version of the application, automatic server discovery is automatically started on application startup. When a server is found, the server address and port fields are automatically updated with the correct values.

3.2 Joining a game

Once the connection is established the server game list will be automatically populated with all games currently available on the server. Note, from here on to obtain latest information the list has to be manually refreshed by clicking the corresponding button.

The desired game should be selected from the list and it will be joined by clicking the "Join Game" button.

3.3 Creating a new game

To create a new game, the desired number of players must be entered on the bottom right of the window and the "Create Game" button clicked.

3.4 Playing the game

On game join or creation the game window will be shown. On the left hand side is the Sudoku board and on the right game info, including running game duration, player scores and notifications. Clicking the "Leave Game" button on the bottom right will remove the player from the game and take him back to the lobby screen.

Initially the game board will be empty while waiting for other players. Once all required players have joined, the game board will be populated with the initial state and the simultaneous solving of the puzzle can begin.

Number entry mode is entered as usual in tabular applications by double clicking, tabbing or hitting enter.

4 Network Architecture

Figures 1 and 2 show the application network architecture and layout.

Automatic server discovery is implemented using UDP broadcast messages and responses. When started, the client periodically sends out a structured broadcast message. When the server, that is listening for broadcast messages, receives and verifies the message it responds directly to the client with its game communication public address.

The game network protocol is implemented using RPC over the SnakeMQ indirect messaging API. The client initially connects to a public connection handling port of the server and is then redirected to a private port belonging to a specific client handler. Both connections use RPC over SnakeMQ.

Note: This network module utilizing SnakeMQ and RPC is a direct port of the previous socket-based implementation. Therefore it may not be the most efficient use of RPC. At the same time the easy portability demonstrates the benefits of the application's modular design.

Note: As the main goal of multiplayer Sudoku is to be concurrent, time uncoupling is in no way a reasonable requirement for the networking solution.

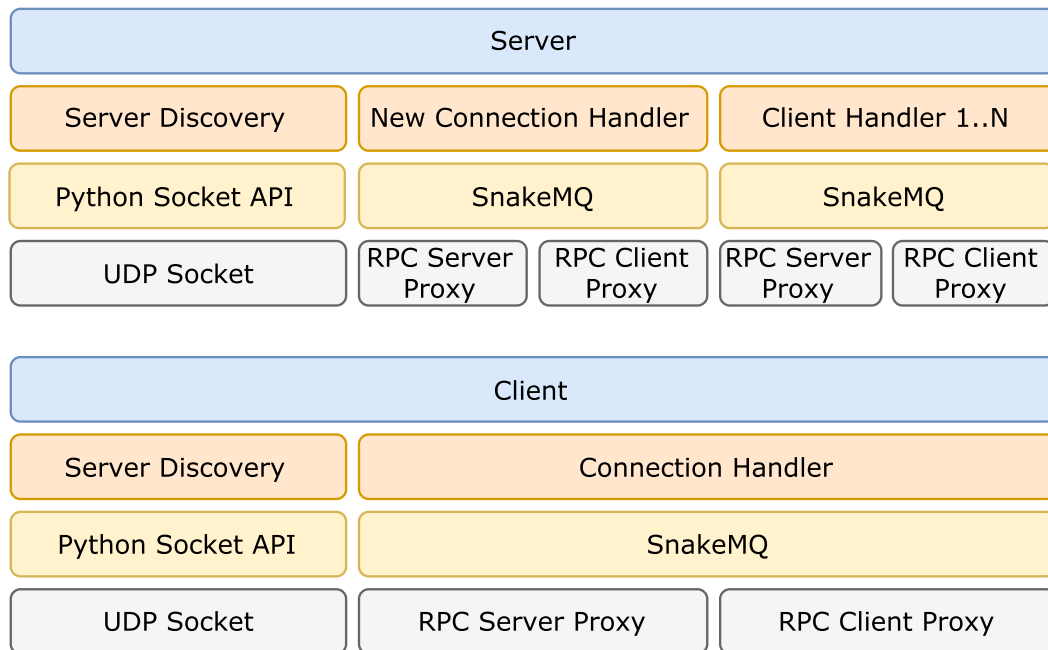


Figure 1: SudokuMP Network Architecture

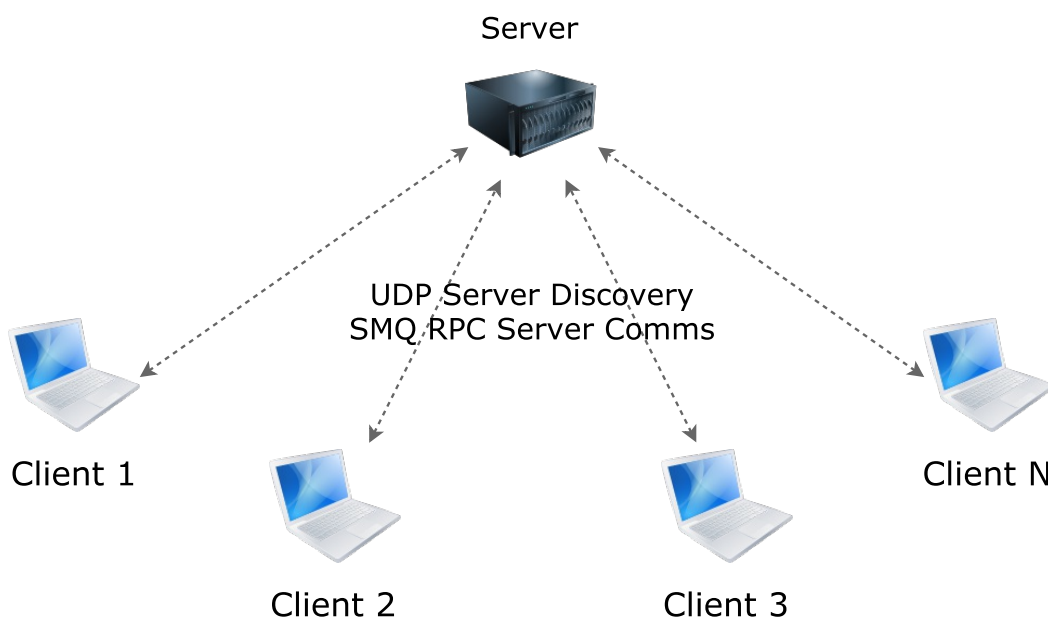


Figure 2: SudokuMP Network Layout