

Cross-App Communication Over TCP/IP With Python

Table of Contents

Table of contents	1
Analysis	2
Summary	2

Analysis

Historically, networked application developers were forced to write a lot of boilerplate code to discover other services/nodes, connect to them, and transfer information between themselves. This resulted in inconsistencies across systems and if not implemented correctly or checked with full custom test suites, unnecessary headache. At best, such changes could lead to fragile code, and at worst unreliable behaviour. The introduction of libraries such as NServiceBus alleviated the previous code-compliance issues quite a lot, but there is still implicit reliance on the database connection (when using MSMQ or RabbitMQ), which in one way or another encumbers the developers further. Current approaches also enforce a specific P2P protocol to be used with the framework and therefore limit the applications that can be built to a single type of networks.

This project introduces a highly generic P2P framework which does not rely on any specific P2P protocol, on top of TCP/IP. This means the framework could potentially work over IP networks, ad-hoc networks, unidirectional message queue brokers and databases without modification to the framework. A concurrency-safe messaging pattern has also been implemented to allow asynchronous message passing between nodes.

This project results in a framework which is generic enough that application developers can focus on the application code instead of APIs, bandwidth and message-passing protocols or discovery techniques which is similar to how one would write the normal API wrappers today. It improves scalability of networked applications by avoiding having backbone servers load all the logic and results in a more flexible framework for all kinds of cross-application protocols.

Plugins for popular distributed system frameworks could also be designed which would further bridge the gap between high-level modules and the underlying logic of the networked applications. A transparent layer for cross-application protocols in such framework would bring node discovery and message transfer down to the framework level, which contributes a lot to collective stability of distributed systems across heterogeneous platforms.

Summary

With this project I aim to bring a customisable cross-app communication framework for Python. The main points in the early development stage of the project are to develop the framework and ensure the framework is extensible, flexible, reliable and stable, while the intermediate goal is to build a reliable and extensible plugin system which other developers can use to build their own plugins. The main goal is to contribute a good open source, extensible framework.

