



LIBRESPEED

An Open Source Speed Test Tool

Software Architecture Report of Librespeed.org

Course: SE 211 01 Software Design and Architecture

Instructor: ENSAR GUL

Prepared By

MUSTAFA ÇİFTÇİ

190706028

Table of Contents

Speed Tests	2
What is a Speed Test?	2
Why Speed Tests are important?	2
How Speed Tests Works in Summary?	3
Mostly Known Speed Test Tools or Websites	3
LibreSpeed.....	4
What is LibreSpeed?	4
Comparison against known speed test applications.....	4
Architecture.....	5
Software Architecture Style.....	6
Front-End or Client.....	7
Web Workers.....	7
Speedtest.js and speedtest_worker.js	7
Backend or Server	8
Database.....	10
Communication Diagram.....	10
Deployment.....	11
Summary	12

Speed Tests

What is a Speed Test?

An internet speed test measures the connection speed and quality of your connected device to the internet. It does so by running multiple consecutive tests that analyze different aspects of your internet connection, namely ping (latency), download speed, and upload speed. Each of these values represents the connection's specific qualities.

Ping: A ping (Packet Internet or Inter-Network Groper) is a basic Internet program that allows a user to test and verify if a particular destination IP address exists and can accept requests in computer network administration.

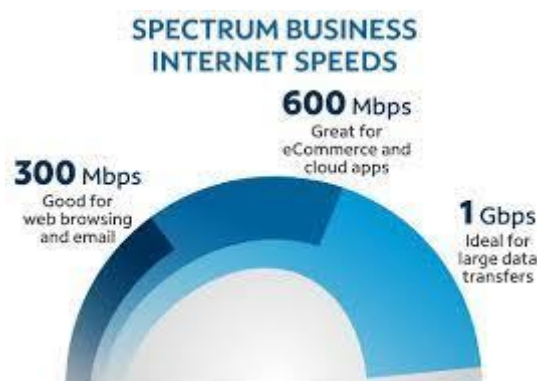
Jitter(latency): Jitter is when there is a time delay in the sending of these data packets over your network connection. This is often caused by network congestion, and sometimes route changes.

Download speed: How fast you can pull data from the server to your location, measured in megabits per second (Mbps)

Upload speed: How fast you send data to others, measured in megabits per second (Mbps)

Why Speed Tests are important?

Internet speed and network speed aren't only important to everyday users, but it is also vital for companies. Many companies rely on their online presence and online sales for a successful business operation. There are more than 9.1 million online retailers around the world and 2.5 million of which are in the United States. Even small business owners rely on their websites to help with sales. During the pandemic year, it is estimated that the percentage of online business and sales increased by 27.9%. Internet connectivity is a crucial part of the way businesses operate.



How Speed Tests Works in Summary?

To start a speed test on web browser user should select a backend server or it will be selected automatically. Selecting servers is substantial because user can select hundreds of servers around the globe. While selecting servers users should consider the distance between their location and server location because the further away the server's location is, the greater the latency.

When user start testing the client establishes multiple connections with the server over port: 8080 (generally). The client requests the server to send an initial chunk of data. The client calculates the real-time speed of the transfers then adjusts the chunk size and buffer size based on this calculation to maximize usage of the network connection. During the first half of the test, the client will establish extra connections to the server if it determines additional threads are required to measure the download speed more accurately. The test ends once the configured amount of time has been reached.



A Speed Test Example from Ookla Speed Test

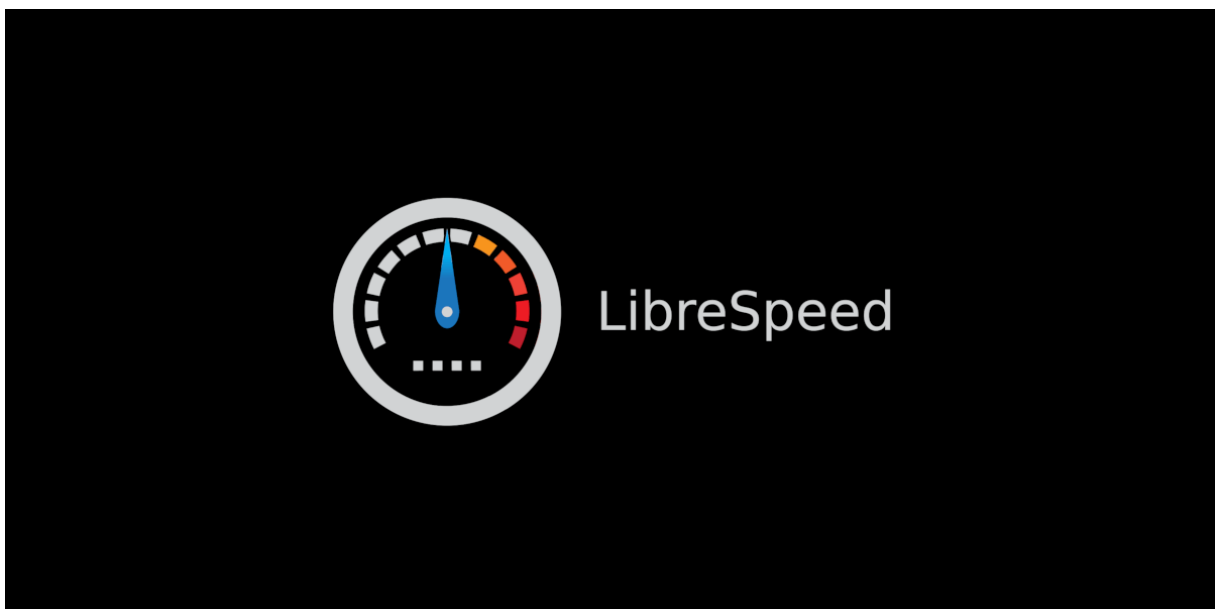
Mostly Known Speed Test Tools or Websites

- Ookla Speedtest
- Meteor
- SpeedTest Master
- V-SPEED
- FAST
- FCC Speed Test
- LibreSpeed

LibreSpeed

What is LibreSpeed?

LibreSpeed is a simple and powerful open-source tool that uses JavaScript, XMLHttpRequest and Web Workers with HTML5 because of that LibreSpeed doesn't use Flash, Java or any Web Socket so it's ideal for situations where you want to self-host a speed test, measure upload and download speeds on multiple sensors concurrently and mimic the user experience using standard web-based methods. In addition, there is also a public LibreSpeed service the sensors can test.



Comparison against known speed test applications

LibreSpeed	Microsoft Network Speed Test	Ookla Speedtest	Xfinity Speed Test
Learn More Update Features	Learn More Update Features	Learn More Update Features	Learn More Update Features
Platforms Supported	Platforms Supported	Platforms Supported	Platforms Supported
Windows	Windows	Windows	Windows
Mac	Mac	Mac	Mac
Linux	Linux	Linux	Linux
SaaS / Web	SaaS / Web	SaaS / Web	SaaS / Web
On-Premise	On-Premise	On-Premise	On-Premise
iPhone	iPhone	iPhone	iPhone
iPad	iPad	iPad	iPad
Android	Android	Android	Android
Chromebook	Chromebook	Chromebook	Chromebook

Architecture

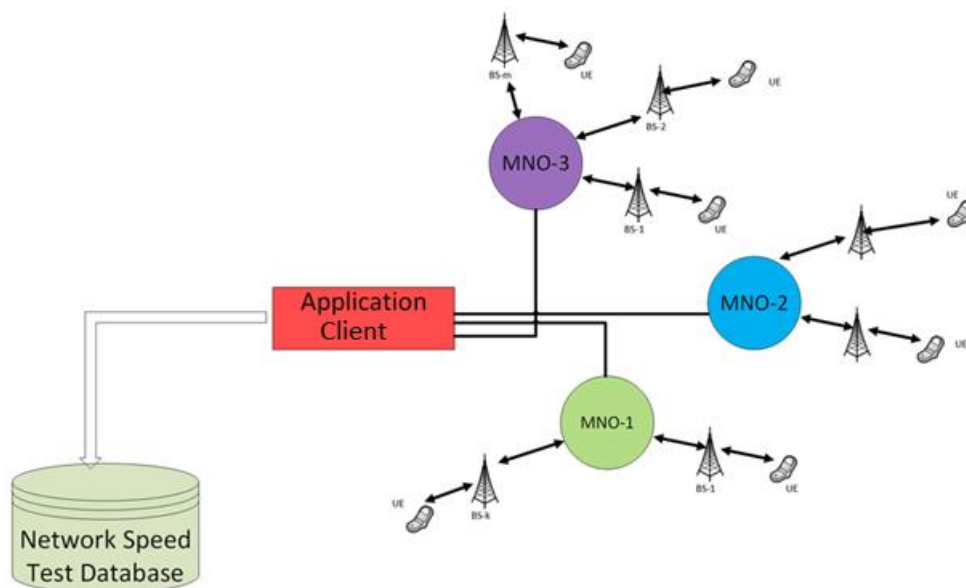
LibreSpeed is an open-source tool that allows users to measure the speed and performance of their internet connection. It is written in php, JavaScript and uses Node.js as its runtime environment.

The software architecture of LibreSpeed can be divided into three main components: the front-end, the back-end, and the database.

The front-end of LibreSpeed is responsible for the user interface and the interactions with the user. It is built using HTML, CSS, and JavaScript and makes use of libraries such as React and Bootstrap. The front-end communicates with the back-end through an API (Application Programming Interface) using HTTP requests. Web workers have plays a big role here

The back-end of LibreSpeed is responsible for handling the requests from the front-end, communicating with the database, and performing the necessary calculations and operations. It is built using php.

The database stores the measurement data collected by LibreSpeed. It is implemented using MySQL, PostgreSQL or SQLite database management system.



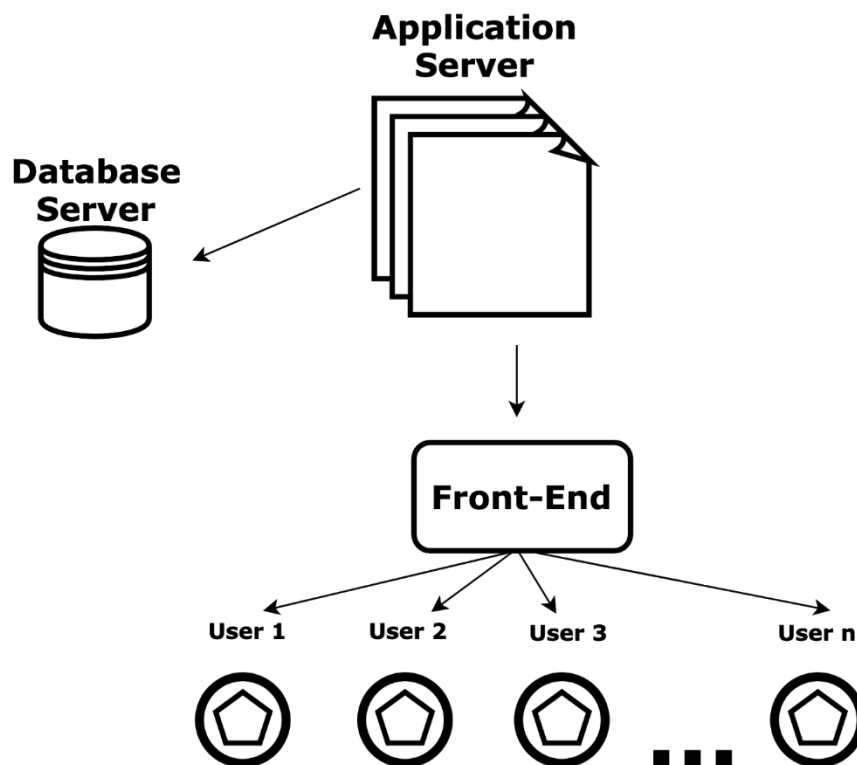
The network architecture of LibreSpeed likely to the above figure. Client access to the database server and multiple servers across its network to measure ping, download speed etc. at intended servers there is more than 20 servers actively working on the LibreSpeed's official server.

Software Architecture Style

LibreSpeed has a client-server architecture, which means that it consists of two main components: the client, which is the web browser running on the user's device, and the server, which is the machine running the LibreSpeed software.

The client initiates the speed test by sending a request to the server, and the server responds by sending a large amount of data back to the client. The client measures the time it takes to receive this data and uses this time to calculate the speed of the network connection.

The client-server architecture allows the speed test to be conducted remotely, making it possible to test the speed of a network connection from any location. It also enables the speed test server to handle requests from multiple clients simultaneously, allowing multiple users to conduct speed tests at the same time.



At the figure above we see the multiple users can access application server with Front-End and with the application server speed test data and measurement can be stored on the database server.

Front-End or Client

There is 9 customizable html pages that contains CSS, JavaScript and HTML code which operate web workers in it.

Web Workers

Web Workers is a feature of modern web browsers that allow developers to run JavaScript code in the background, separate from the main thread of the application. This can be useful for offloading CPU-intensive or time-consuming tasks to a separate thread, allowing the main thread to remain responsive to user input.

Web Workers to perform tasks such as speed tests or data processing in the background, without blocking the main thread of the application. To use Web Workers in an application, developers need to create a separate JavaScript file that contains the code to be run in the worker thread. The main application can then create a new worker instance and communicate with it using messages. The worker can perform its tasks and return results to the main application through the message channel.

Speedtest.js and speedtest_worker.js

This is the main interface between your webpage and the speed test. It hides the speed test web worker to the page and provides many convenient functions to control the test.

To initialize the test, create a new Speed test object: `var s=new Speedtest();`

Now you can think of this as a finite state machine. There are 5 stages on the speed test those are:

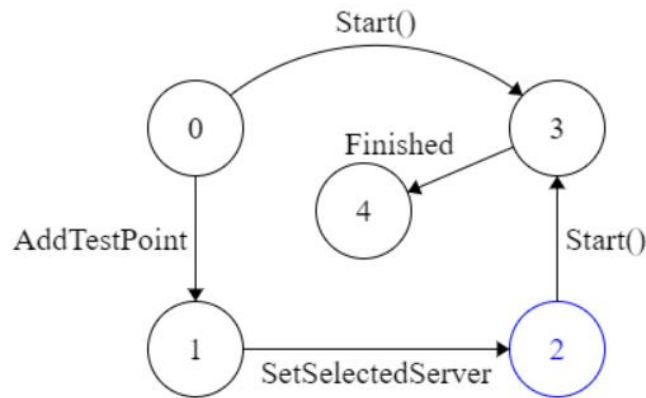
0 - Here developer can change the speed test settings (such as test duration) with the `setParameter("parameter",value)` method. From here you can either start the test using `start()` (goes to state 3) or you can add multiple test points using `addTestPoint(server)` or `addTestPoints(serverList)` (goes to state 1). Additionally, this is the perfect moment to set up callbacks for the `onupdate(data)` and `onend(aborted)` events.

1 - Here you can add test points. You only need to do this if you want to use multiple test points.

2 - Test point selected, ready to start the test. Use `start()` to begin, this will move to state 3

3 - Test running. Here, your `onupdate` event callback will be called periodically, with data coming from the worker about speed and progress.

4 - test finished. You can run it again by calling `start()` if you want.



Backend or Server

The back-end of LibreSpeed is the part of the application that handles requests from the front-end, communicates with the database, and performs the necessary calculations and operations. It is typically implemented using Node.js and communicates with the front-end through an API (Application Programming Interface).

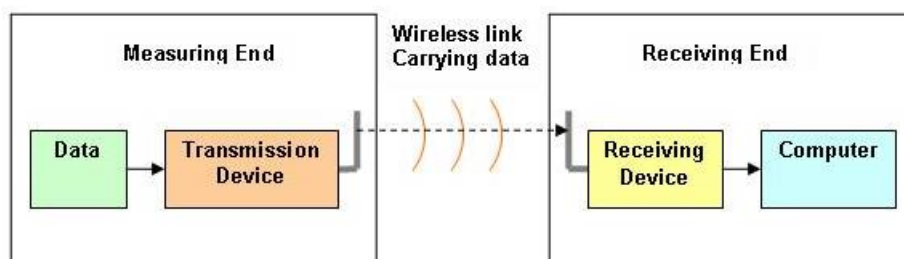
The back-end is responsible for handling requests from the front-end and communicating with the database to store and retrieve data as needed. It may also perform calculations and other operations to support the functionality of the tool.

Connection with front-end or client made on HTTP requests using NGINX to serve static content.

NGINX is a web server software that is often used to handle HTTP requests and serve static content, such as HTML, CSS, and JavaScript files. It is known for its high performance, scalability, and ability to handle large numbers of concurrent connections.

Telemetry system used to collect data about the performance and usage of the tool, such as the number of users, the types of internet connections being tested, and any errors or issues that may arise. Telemetry used on the: Data collection, Data transmission, Data storage and analysis, Data reporting and visualization.

Telemetry is the process of collecting and transmitting data from a remote location, typically for the purpose of monitoring or tracking some aspect of the system or environment.



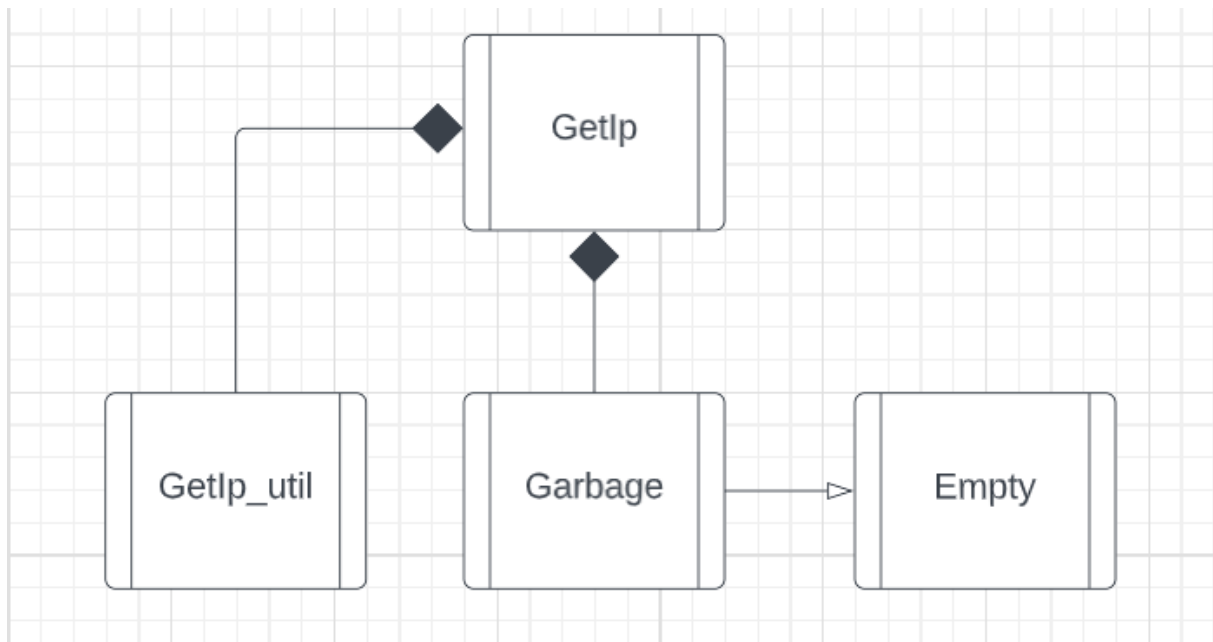
Simple Telemetry Example

In the getIp file with the getIp_util server sent an HTTP request to the client and retrieve the data which is HTTP_CLIENT_IP and decides the Ip type (IPv4 or 6 private or public etc.) so that server is enable to access the Ip then to get the distance between server and ISP (internet service provider) info the Ip information will be sent to ipinfo.io.

After all the data is retrieved then speed test is good to go.

When the speed test is started with garbage.php chunks is sent from server each chunk is 1.04 Mb and on each chunk ping, jitter(latency) and download speed are tested. Chunks made of garbage HTTP headers and number of chunks depends on user-agents or versions.

After the testing stage is done all the retrieved data sent to database one by one.



As can be see on the figure above getIP has composition relationship with garbage and getIp_util classes because can't and no need to exist if getIp is not initiated. Garbage class inherited from empty because empty class have empty HTTP headers so that garbage can fill them with appropriate data.

Database

The database architecture of LibreSpeed refers to the way in which the data is stored and organized in the database, as well as the processes and technologies used to manage and access the data.

Database architecture of LibreSpeed follows a standard model for storing and organizing data such as a relational model.

In a relational model, data is stored in tables with rows and columns, and relationships between data are established using foreign keys. This model is well-suited for storing structured data and allows for efficient querying and data manipulation using SQL.

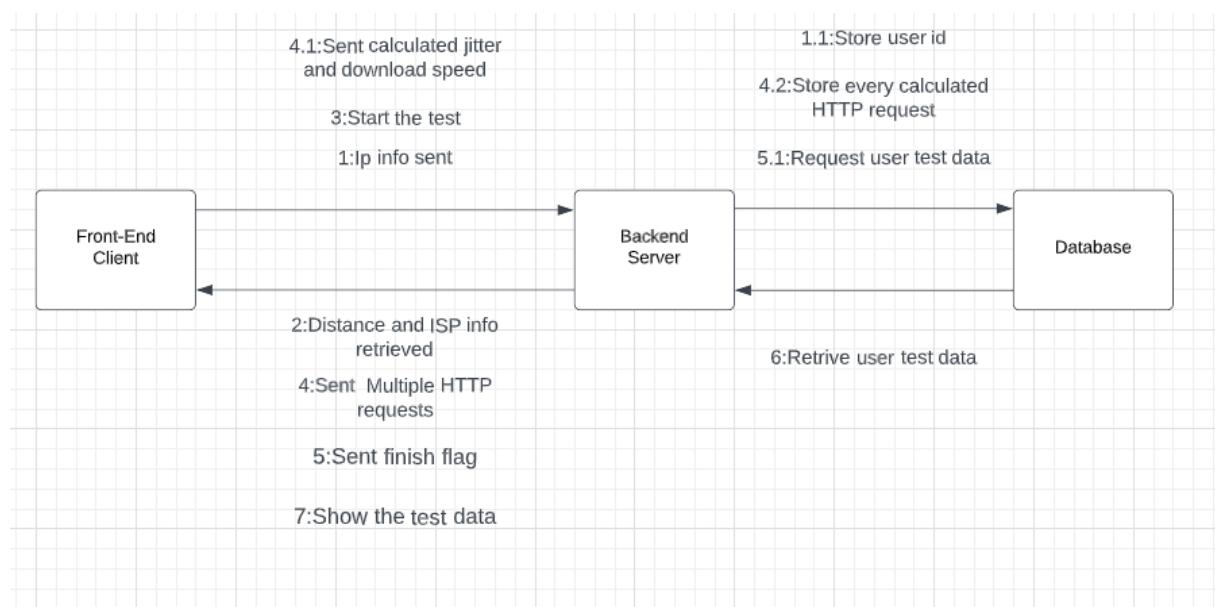
There are 3 configurable database management system which are MySQL, PostgreSQL and SQLite.

Every test is distinguished with id number and this id numbers is encrypted with salt obfuscation at idObfuscation.php and store in the database according to that.

In the database part of the architecture telemetry system defined to securely transfer the data between frontend back-end and the database. It's defined in the files telemetry.php and telemetry_db.php also in this pages main database queries involved which are insert and select queries but before the system deployed database must be initialized.

The telemetry_db retrieves the telemetry settings, detects the database type (MySQL, PostgreSQL, or SQLite) from the settings, and sends the information to the database in chunks. After the test is finished, it retrieves all the information from the database that matches the unique id. Then with the index.php post the data on the picture.

Communication Diagram



1. When user enter the website Ip info is sent to the backend servers to see available servers and store the obfuscated user id in database.
2. Server sent back the distance between the server locations and client. Server sent client to ISP (Internet service provider) info.
3. Client sent request that server can start the speed test.
4. Server sent multiple HTTP request to client. Client measure the data and sent back to server so that server can store the calculations on database.
5. Server sent finish flag to the client and request user test data from database.
6. Database sent user test data to the server
7. Backend sent the final information that contains ISP, ping, jitter, download and upload speed to the client.

Deployment

The LibreSpeed web application can be deployed in several different ways, depending on the requirements of the application. Some possible deployment scenarios include:

Single server deployment: In this scenario, the web server, application layer, data access layer, and data storage layer are all hosted on a single server. This is a simple and cost-effective deployment option, but it may not scale well if the application receives a large number of requests.

Load-balanced (multiple servers) deployment: In this scenario, multiple servers are used to host the web server and application layer, and a load balancer is used to distribute requests evenly across the servers. This deployment option can handle a larger number of requests than a single server deployment, but it is more complex and costly to set up and maintain.

Cloud deployment: In this scenario, the speed test web application is hosted on a cloud platform, such as Docker. This deployment option allows for easy scalability and high availability, but it may be more expensive than a single server or load.

It is possible that LibreSpeed could be deployed using Docker, but it is not a requirement for the software architecture of LibreSpeed.

Docker is a tool that allows developers to package applications and their dependencies into containers, which can be easily deployed and run on any host system. Using Docker can make it easier to deploy and run applications, as it allows developers to package their applications in a portable and self-contained way.

Summary

Most of the speed test tools does not have customizable software and network infrastructure. Which leads speed tests to become inaccurate because if a company want to make a speed test among its servers, they have to make connection trough the speed tests servers and since more than 1000 users are doing the test at the same time test may become inaccurate.

LibreSpeed has approached this problem from different perspective and made it open source and configurable source files. Because of this approach companies can make their own speed tests without being interrupt or need connection from 3rd party server.

Some of the reasons why LibreSpeed may be considered better than other tools include:

Accuracy: LibreSpeed is designed to provide accurate and reliable measurements of internet connection speed and performance. It uses industry-standard protocols and techniques to measure the speed and latency of internet connections, and the results are designed to be representative of the user's actual internet experience.

Simplicity: LibreSpeed has a simple and intuitive interface, making it easy for users to initiate speed tests and view the results. The interface is designed to be user-friendly and straightforward, allowing users to measure their internet connection speed quickly and easily.

Customization: LibreSpeed allows users to customize the speed test settings to better match their specific needs and requirements. Users can choose the type of connection (wired or wireless), the test server location, and other parameters to tailor the test to their specific circumstances.

Community support: LibreSpeed is an open-source project, and it has a strong community of developers and users who contribute to the project and help improve the tool. This means that the tool is continuously being improved and updated, and users have access to a wealth of knowledge and resources to help them get the most out of the tool.

Overall, LibreSpeed is a powerful and effective tool for measuring internet connection speed and performance, and it is well-suited for a wide range of users and circumstances. Its accuracy, simplicity, customization options, and community support make it a valuable resource for anyone looking to measure their internet connection speed.