

CS312-Modern Development Tools



Dr. Mohamed Refaat Mohamed

Lecture 1

CS312 Course Marking Scheme

- **Exam 1,2:** **20**
- **Project:** **100** using Django or Flutter OR (AI Project)

 70 Project + 10 Doc. + 20 Attendance
- **Final Exam:** **30**



What Is a Software Build?

What Is a Software Build?

Software builds are a part of the continuous integration/continuous delivery (CI/CD) pipeline and are normally done when a particular milestone is achieved, such as a release or audit. The process of creating builds can be done manually or using automated tools.

Many organizations incorporate **automated build tools** in their development.

However, this does not invalidate the use of **manual build processes**.

They are particularly useful when carrying out complex tasks that may not necessarily be done in an iterative framework.

Types of Software Builds

1.Full Build:

This requires compiling every object from scratch. It's necessary to ensure that all the code runs properly. Full builds can be slow to achieve. In some cases, they could take minutes to finish. Some ways to tackle this issue is through parallel execution, e.g., building and testing simultaneously.

Types of Software Builds

2.Incremental Build

This involves compiling only the objects that have changed and those that reference them. This process relies on the previous build and adds on it.

For that reason, it's faster than a full build.

Incremental builds can be done through content-aware hashing and also the use of timestamps.

However, it does come with some challenges. Sometimes, the build framework may not be able to correctly register all changes, for example, in the instance of a file deletion.

Build Tools

Build tools are software that automates the process of compiling, testing, and packaging software. These tools are used to implement development workflows in CI/CD pipelines.

Some of the most popular build tools are **Travis CI, Make, and Maven**. Notably, Travis CI supports over 30 programming languages. Make only supports C/C++, while Maven is a popular tool for Java applications.

Why Do We Need Build Tools?



- **Processes are less error-prone.** Due to the predictable and repetitive nature of the setup, we can catch bugs and code failures early
- **Faster development.** These tools automate a number of standard procedures in the DevOps process. For example: build triggers to automate checks for code formatting.
- **They facilitate packaging software into deployable formats, e.g., WAR** (Web Application ARchive) files, bytecode, and binaries. This comes with numerous benefits, such as ease of distribution and control over the installation process

Why Do We Need Build Tools?



- **Foster collaboration.** Define a standard directory structure & best practices. This eases collaboration & maintainability since all the team members know the standard place where files must go.
- **Automated tests.** This software provides unit tests, integration tests, and regression tests for your code
- **Job Logs.** For every build, there's an accompanying log. This enables developers to know at which build number an error first occurred and which ones were okay. This simplifies the software management process. In particular, debugging becomes easier



Examples of the Best Software Development Tools for 2025

www.unifiedinfotech.net

1. Visual Studio Code (VSC)

Rating: 4.8/5

Best For: General-purpose code editing, lightweight development, extensibility

Developed by: Microsoft

2. Git

Rating: 4.6/5

Best For: Version control, collaboration, distributed development

Developed by: The Linux Foundation

3. Jira

Rating: 4.4/5

Best For: Agile project management, bug tracking, sprint planning

Developed by: Atlassian

4.GitHub

Rating: 4.7/5

Best For: Open-source collaboration, version control, project management

Developed by: Microsoft

5.IntelliJ IDEA

Rating: 4.5/5

Best For: Java and Kotlin development

Developed by: JetBrains

6.BitBucket

Rating: 4.4/5

Best For: Version control, enterprise software development

Developed by: Atlassian

7.Bootstrap

Rating: 4.5/5

Best For: Front-end development, responsive web design

Developed by: Mark Otto and Jacob Thornton at Twitter

8.Codeium

Rating: 4.3/5

Best For: AI-powered code completion

Developed by: Exafunction

9.JetBrains Suite

Rating: 4.4/5

Best For: Comprehensive IDEs for multiple stages of software development

Developed by: JetBrains



The Function of TIME_WAIT

The Function of TIME_WAIT

The cost of many TIME_WAITs on the server side. First, we'll understand the function of the TIME_WAIT by examining its role in a TCP connection. TCP (Transmission Control Protocol) provides reliable, ordered, and error-checked transmission of byte streams between hosts on an IP network. As a connection-oriented protocol, TCP requires the client and server hosts to establish a connection before transferring any data. Therefore, any packet transmitted via TCP must be associated with a specific TCP connection.

The Function of TIME_WAIT

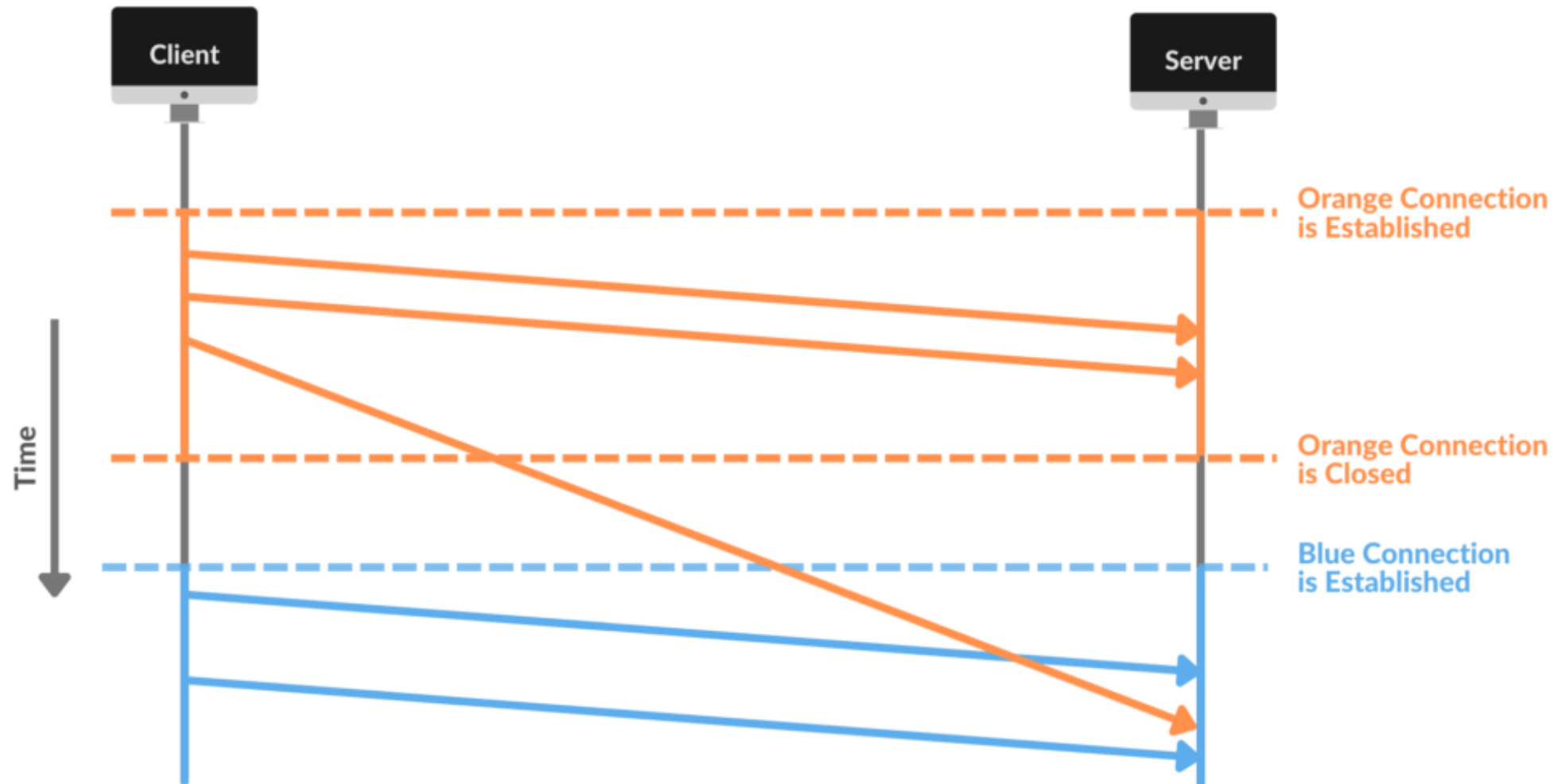


Each **TCP connection** is tied to a tuple of **four elements**: client address, client port, server address, and server port. Thus, multiple simultaneous TCP connections between two hosts using different ports can exist. Also, two hosts can open and close many connections over time and can even use the same address/port pair.

large-scale networks, such as the Internet, can delay the delivery of packets or generate duplications. **Therefore, TCP needs to apply some methods to isolate old connections from new ones.** TIME_WAIT is a key component in this connection isolation.

The Function of TIME_WAIT

In particular, TCP uses TIME_WAIT to prevent packets from an old (closed) connection from being delivered during a new one:



The Function of TIME_WAIT

In this example, a client and server establish a TCP connection (in orange), send three data packets, and close it. Shortly afterward, the same hosts open a new TCP connection (in blue) using the same address/port pair and transfer more data.

However, the network delayed the delivery of the third packet of the first connection. **Thus, a packet belonging to the orange connection arrived at the server during the blue connection.** This can cause various problems since that packet doesn't belong to the blue connection.

TCP deals with this issue by defining that every connection that has been closed must wait a certain amount of time before it can be established again.

The value set for TCP's **TIME_WAIT** defines the **total waiting time to allow new connections with the same addresses/ports pair.**

The Cost of Many Connections in TIME_WAIT State

As we've seen, TCP blocks new connections on the same addresses/ports pair as a recently closed connection until there are no more related packets on the network. For this to happen, one of the endpoints needs to keep a copy of the TCP control block (TCB) indicating that the connection has been terminated recently. **We say that such a connection is in the TIME-WAIT state. In most cases, the host that keeps it is the server.**

So, a possible effect of TCP's TIME-WAIT is to **prevent a client from establishing a connection with the server.** However, this occurs only in two specific cases.

The Cost of Many Connections in TIME_WAIT State

As we've seen, TCP blocks new connections on the same addresses/ports pair as a recently closed connection until there are no more related packets on the network. For this to happen, one of the endpoints needs to keep a copy of the TCP control block (TCB) indicating that the connection has been terminated recently. **We say that such a connection is in the TIME-WAIT state. In most cases, the host that keeps it is the server.**

So, a possible effect of TCP's TIME-WAIT is to **prevent a client from establishing a connection with the server.**

 **However, extreme cases of too many connections in a TIME_WAIT state on the server can cause problems for both the server and the client.**

When TCP's TIME-WAIT prevent connection with the server

When both client and server try to connect using the same ports and addresses, they used in a connection currently in the TIME-WAIT state. But the client typically uses a different port for each new connection. This means that each new connection a client tries to establish doesn't match the previous one. can only occur if the client has used up all their ephemeral ports.

For the server, many connections in a TIME-WAIT state can affect the server's performance when processing active connections. multiple clients can coordinate a distributed denial-of-service DDoS attack by deliberately generating connections in a TIME-WAIT state.

To avoid these situations, we can force the closing or allow the reuse of sockets in a TIME-WAIT state. However, we must be very careful when doing this.