**Installation manual**

The computational performance evaluation framework operates by quantifying the hardware-level behavior of algorithms through measurements of CPU instructions, processor cycles, and total execution time using the *perf* utility, a native Linux tool that provides direct access to processor performance counters. Since *perf* requires low-level kernel access, it operates exclusively under Linux environments. For systems based on Microsoft Windows, this requirement is satisfied through the **Windows Subsystem for Linux (WSL)**, which provides a fully functional Linux kernel capable of running *perf* natively.

The installation process begins with the activation of WSL. This step is carried out from the Windows PowerShell terminal with administrative privileges by executing the command:

wsl –install

This command enables the necessary system features, installs the default Linux distribution (typically Ubuntu), and sets WSL2 as the active version. Once the installation is completed, a system restart is required. After the reboot, the Linux environment can be accessed either from the Start menu or by typing wsl in the PowerShell or Command Prompt terminal. During the first execution, it is necessary to define a username and password to finalize the initialization process of the Linux subsystem. After this initial configuration, the Linux terminal is ready for software installation.

The automated setup of the environment is performed entirely within WSL using a Python-based installation script named setup.py. Before executing the script, it is essential that the file be accessible from within the WSL environment. If the script was created or stored in the Windows filesystem, it must be copied into the WSL home directory. This can be accomplished by navigating to the corresponding path under /mnt/, where all Windows drives are automatically mounted. For example, if the script is located on the Windows desktop, the following command copies it to the WSL home directory (it’s necessary to change the <username> with the corresponding username):

cp /mnt/c/Users/<username>/Desktop/setup.py ~/

Once the script has been transferred, it can be executed directly from the Linux terminal. In most modern versions of Ubuntu distributed through WSL, Python 3 is preinstalled by default. If the interpreter is not present, it can be installed manually by executing:

sudo apt update && sudo apt install -y python3

With Python available, the automated setup is initiated using the command:

python3 setup.py

The setup.py script was developed to perform the entire configuration process autonomously and reproducibly. It begins by verifying that the execution environment corresponds to a WSL instance through inspection of the system descriptor /proc/version. Once verified, it updates the package repositories, installs all development tools, and retrieves the Microsoft WSL2 kernel source tree from the official GitHub repository. It then installs the necessary libraries for the compilation of the *perf* utility, such as those related to event tracing, compression, stack unwinding, and numerical computations. The compilation is carried out using internal kernel implementations of the libraries libtraceevent and libtracefs to ensure compatibility with the Microsoft-maintained kernel version. After successful compilation, the *perf* binary is automatically copied into /usr/local/bin/, making it globally accessible.

The script integrates an internal logging system that records every operation in real time. All progress information and potential errors are simultaneously displayed on screen and stored in a log file named perf\_setup.log, located in the home directory. This record provides a complete trace of the installation and facilitates verification of each step. Additionally, the script supports optional execution parameters to modify its behavior: the argument --skip-update omits the initial system update when it is already current, while --only-build-perf restricts the process to recompiling *perf* without reinstalling dependencies.

Upon completion, the script verifies the correct installation of *perf* by executing perf --version and performing a test measurement with sudo perf stat -- sleep 0.1. The resulting output contains a summary of processor performance events, confirming that the profiler operates correctly within the system. A final report is displayed summarizing the total execution time and indicating the location of the log file.

This procedure ensures a deterministic and reproducible configuration of the Linux environment, eliminating manual dependency management and potential inconsistencies between kernel and library versions. As a result, the system obtains a fully functional *perf* utility compiled directly from the Microsoft kernel sources, providing accurate and stable hardware-level measurements for computational performance evaluation.