

# Anleitung zur Nutzerstudie für die Masterarbeit "Ein IDE Plugin für Automatische MPI Code Modernisierung"

## 1. Voraussetzungen

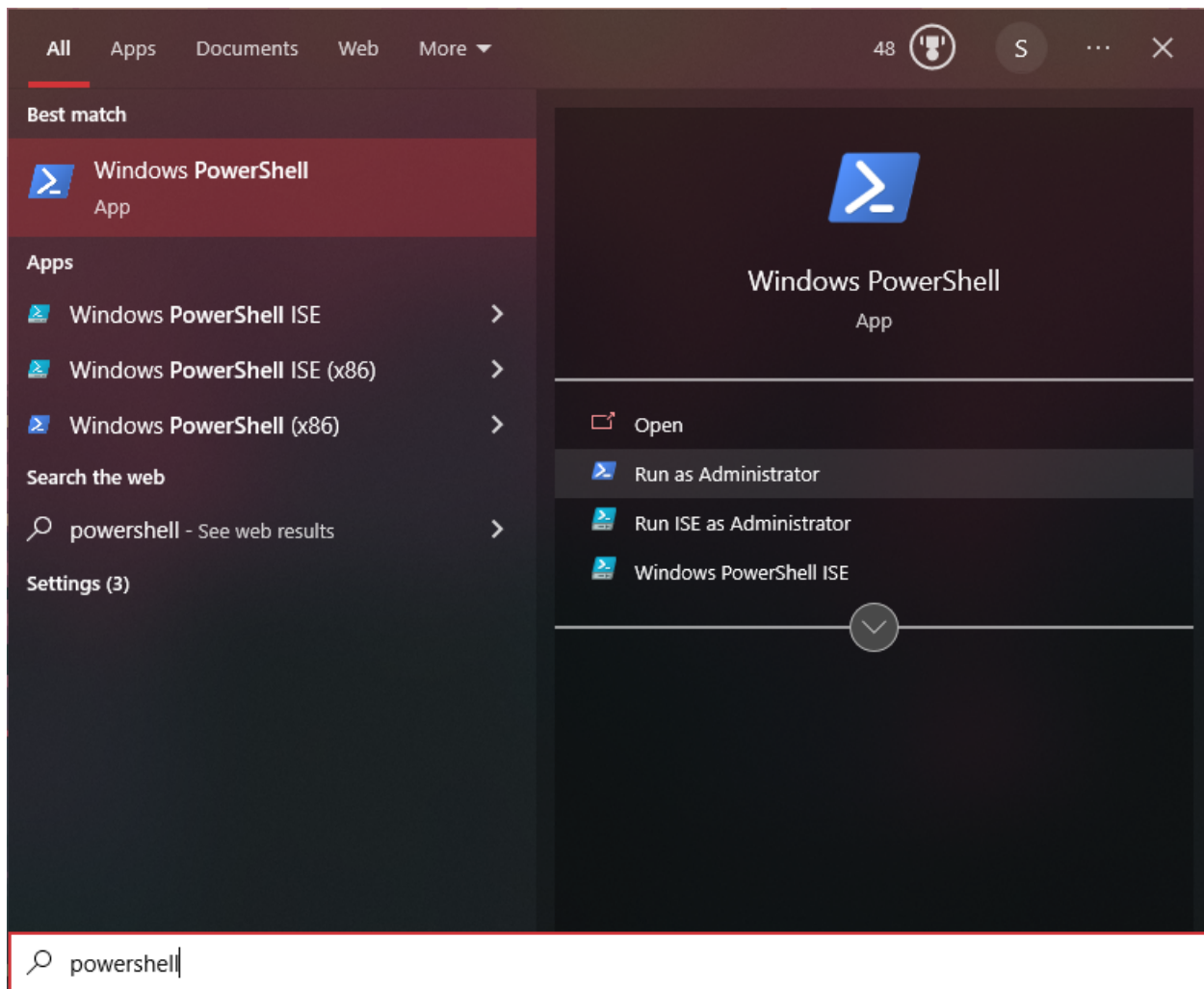
Sie benötigen die folgenden Programme zur Teilnahme an der Umfrage:

- Visual Studio Code
- Ein C++ Compiler, zum Beispiel gcc
- Eine MPI Library, zum Beispiel OpenMPI

### Installieren:

#### Windows

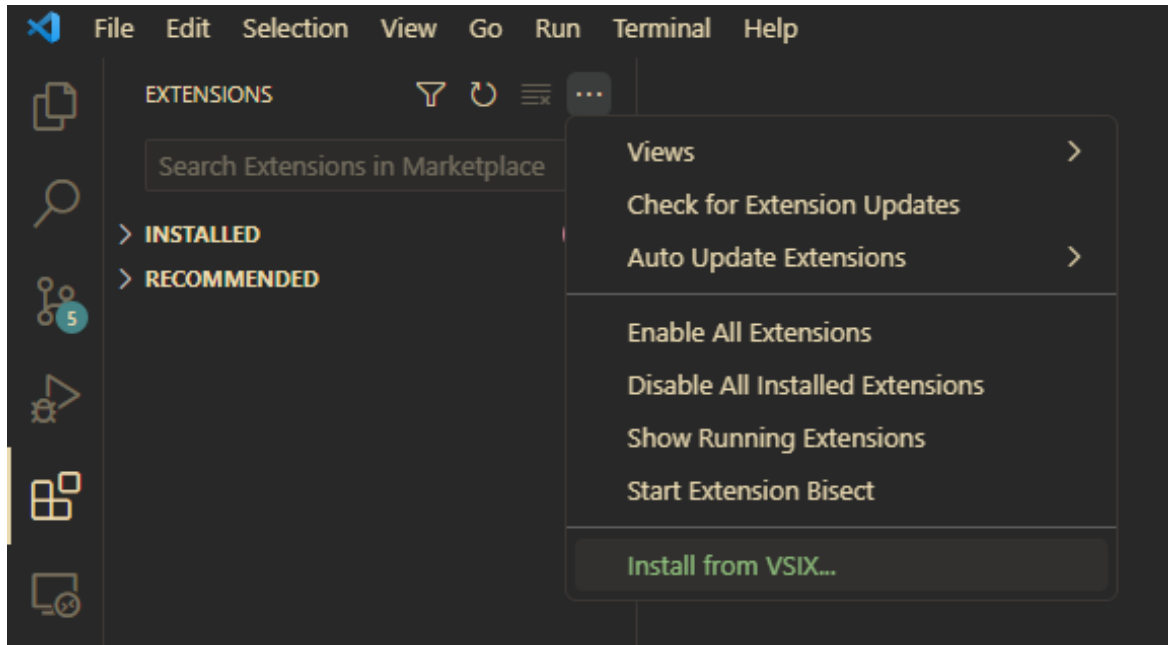
- 1 Laden Sie Visual Studio Code von hier herunter und installieren Sie es.
- 2 Installieren Sie einen C++ Compiler und eine MPI Library. Wenn Sie bereits einen C++ Compiler mit MPI haben, machen Sie mit schritt [3] weiter. Ansonsten empfehlen wir, Windows Subsystem for Linux zu nutzen. Führen Sie dafür folgende Schritte aus:
  - 1 Öffnen Sie die Windows PowerShell als Administrator.



- 2 Führen Sie den Befehl `wsl --install` aus.
  - 3 Warten Sie etwas, bis Sie nach einem Nutzernamen gefragt werden.
  - 4 Geben Sie ein Nutzernamen und ein Passwort an. Diese dienen als Account für das Linux Subsystem.
  - 5 Rebooten Sie ihren Computer
  - 6 Ein Fenster mit dem Titel Ubuntu sollte sich nach dem Login automatisch öffnen und den Installationsprozess fortsetzen.
  - 7 Nachdem Ubuntu fertig installiert ist, kann mit den folgenden Befehlen `gcc` und `openMPI` installiert werden.
 

```
sudo apt-get update
sudo apt-get install gcc make openmpi-bin
```
  - 8 Bestätigen Sie mit 'y'
  - 9 Sie haben nun ein Ubuntu terminal und können dort wie mit einem Linux System arbeiten. Das System kann auch durch die PowerShell mit dem Befehl `wsl` aufgerufen werden. Dies können sie im Visual Studio Code Terminal machen, um dort dann die `make` Befehle in der Durchführung auszuführen. Mit `exit` wechseln Sie wieder in die gewohnte PowerShell zurück.
- 3 Installieren Sie das Plugin in Visual Studio Code:
- 1 Öffnen Sie Visual Studio Code und navigieren Sie in der Aktivitätenleiste auf der

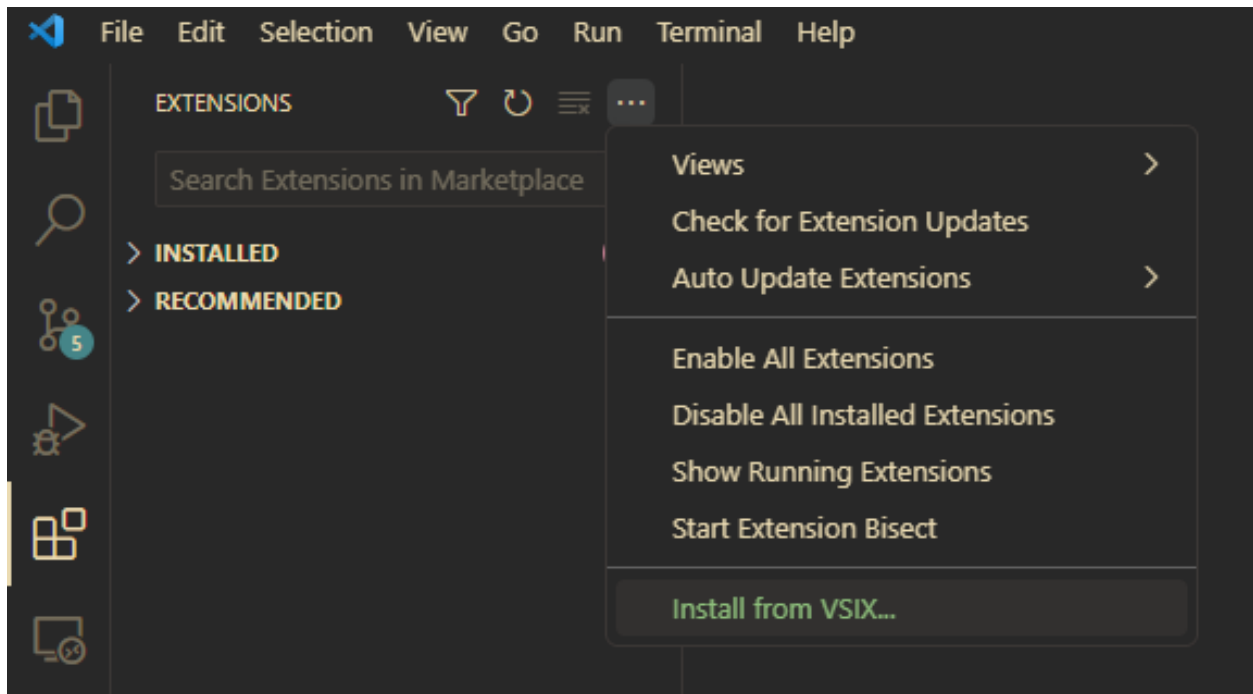
- linken Seite zu den Erweiterungen (eng. Extensions).
- Klicken Sie in dem Extension Tab auf ...
- Wählen Sie *Aus VSIX installieren / Install from VSIX*



- Wählen Sie in dem Fileselektor die bereitgestellte *mpiconv-0.0.X.vsix* Datei aus.

## Linux

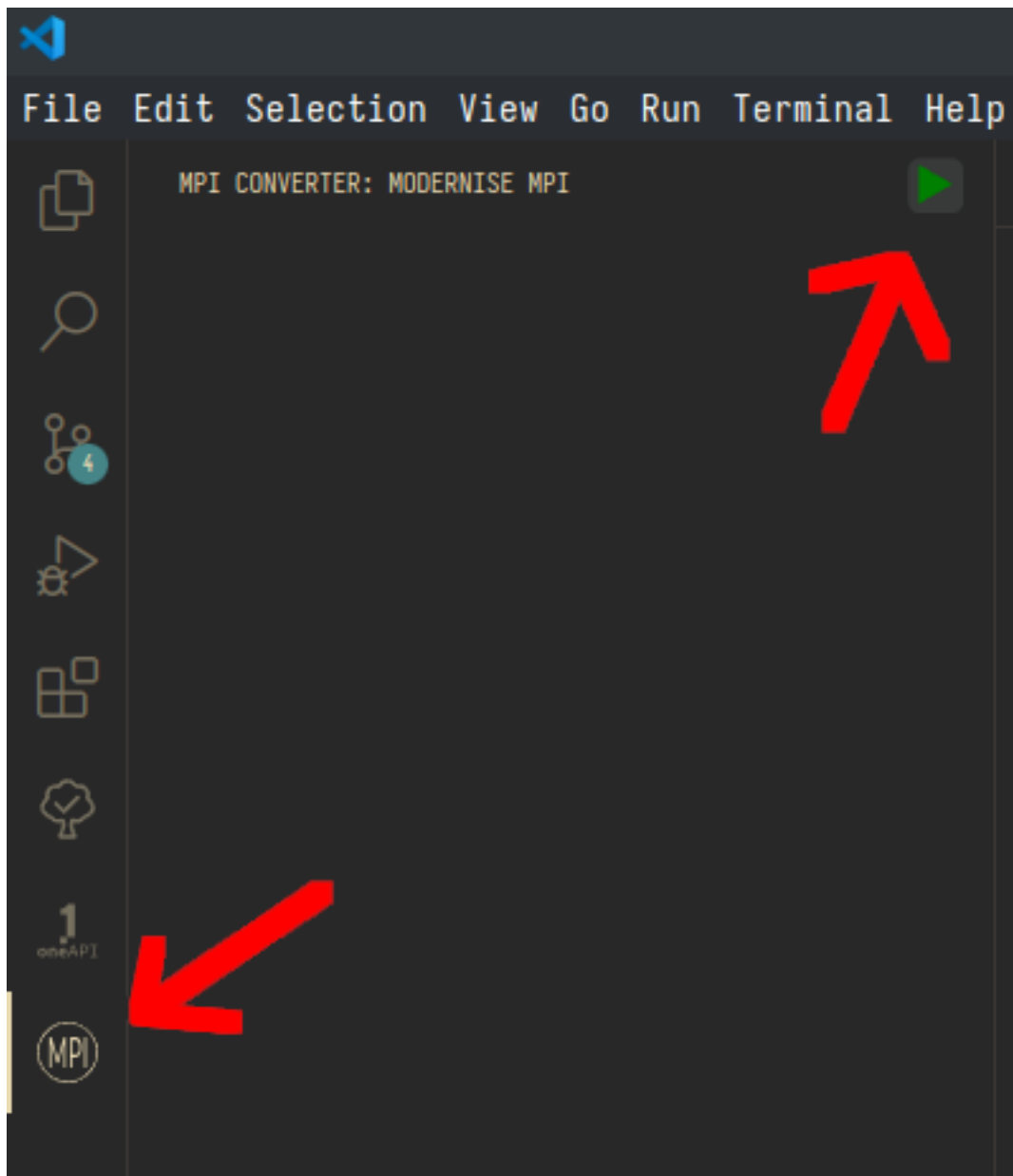
- Laden Sie Visual Studio Code von hier herunter und installieren Sie es. Alternativ kann Visual Studio Code auch von manchen package managern installiert werden.
- Wenn Sie noch keinen C++ Compiler, MPI Library oder Make installiert haben, nutzen Sie Ihren package manager, um 'gcc', 'make' und 'openmpi' zu installieren.
  - Ubuntu/Debian: `sudo apt-get install gcc make openmpi-bin`
  - Arch: `sudo pacman -S gcc make openmpi`
- Installieren Sie das Plugin in Visual Studio Code:
  - Öffnen Sie Visual Studio Code und navigieren Sie in der Aktivitätenleiste auf der linken Seite zu den *Erweiterungen* (eng. *Extensions*).
  - Klicken Sie in dem Extension Tab auf ...
  - Wählen Sie *Aus VSIX installieren / Install from VSIX*



- 4 Wählen Sie in dem Fileselektor die bereitgestellte *mpiconv-0.0.X.vsix* Datei aus.

## 2. Durchführung

- 1 Öffnen Sie den bereitgestellten Ordner *exercise1* in Visual Studio Code.
- 2 Öffnen Sie die Datei *exercise1.cpp*.
- 3 Machen Sie sich etwas mit dem Code vertraut. Das Programm wendet die Funktion *function* auf jedes Element eines Arrays an und berechnet dann die Summe aller Elemente. Dafür wird einer der Prozesse als Master und die anderen als Worker benutzt. Der Master generiert zunächst ein Array und ruft dann die Funktion *master\_work* auf. Dort verteilt er das Array gleichmäßig auf die Worker mit *MPI\_Send*. Dann empfängt er die Subergebnisse mit *MPI\_Recv*. Zum Schluss berechnet er noch den Rest des Arrays und summiert alle Subergebnisse auf. Die Worker rufen direkt die Funktion *worker\_work* auf. Dort empfangen Sie ihren Teil des Arrays und berechnen ihr Subergebnis. Sie geben ihre Subergebnisse an den Master zurück und sind fertig. Das verteilte Ergebnis wird dann noch mit der Funktion *verfity* geprüft. Ob das Ergebnis richtig ist, oder nicht wird auf der Kommandozeile ausgegeben.
- 4 Testen Sie das Programm, indem Sie folgende Befehle im Terminal eingeben:  
make  
make run
- 5 Wenden Sie nun das Plugin an, indem Sie entweder links auf das MPI Symbol klicken und dann den grünen Pfeil anklicken, oder mit der Tastenkombination "*ctr + shift + p*" das Ausführfenster öffnen und "Convert MPI Statements" ausführen.



- 6 Das Plugin wird Ihnen nun eine Reihe an Fragen mit dem Dialogfenster stellen. Versuchen Sie jede der 4 MPI send/recv Instruktionen mit dem Plugin zu ersetzen.
- 7 Testen Sie das Resultat mit den Befehlen:  
make  
make run
- 8 Sollte das Programm nicht direkt Kompilieren, oder das falsche Ergebnis ausrechnen, versuchen Sie die Fehler bitte manuell zu beheben.
- 9 Sie können optional die Schritte 1 bis 8 noch mit dem Ordner *exercise2* machen oder eigene MPI Programme testen. Feedback zu Letzterem wäre für die Weiterentwicklung des Plugins sehr hilfreich.
- 10 Wenn Sie alle Instruktionen ersetzt haben und das Programm das richtige Ergebnis ausgibt, oder wenn Sie es nicht schaffen, das Programm nach dem Anwenden des Plugins zum Laufen zu bringen, öffnen Sie bitte den Fragebogen und füllen ihn aus.

Vielen Dank, dass Sie sich die Zeit genommen haben. Ihre Angaben und ihr Feedback tragen maßgeblich zu meiner Masterarbeit bei.

Bei Fragen oder Problemen wenden Sie sich gerne an mich: [svn.donnerhak@stud.tu-darmstadt.de](mailto:svn.donnerhak@stud.tu-darmstadt.de)