



Triton Join Build Guide

안승민, 임세진

Github repo : <https://github.com/TU-Berlin-DIMA/fast-interconnects>

Guide : <https://github.com/TU-Berlin-DIMA/fast-interconnects/blob/master/guides/setup.md>

1. Nvidia CUDA toolkit

설치여부 및 버전 확인

- /usr/local에서 cuda-<ver> 폴더 확인
- 미설치 시 anaconda.org 에서 cudatoolkit-dev를 검색하거나
conda search cudatoolkit-dev로 적정 버전 선택 후 설치

```
(triton) hvdrk@aisys-small100:/usr/local$ ls
bin  cuda  cuda-11.7  cuda-11.8  cuda-12  cuda-12.0  cuda-12.3  etc  gar
(triton) hvdrk@aisys-small100:/usr/local$ conda search cudatoolkit-dev
Loading channels: done
# Name                                Version      Build      Channel
cudatoolkit-dev                       9.2          py36_2     conda-forge
cudatoolkit-dev                       9.2          py37_0     conda-forge
cudatoolkit-dev                       9.2          py37_1     conda-forge
cudatoolkit-dev                       10.0         1          conda-forge
cudatoolkit-dev                       10.0         2          conda-forge
cudatoolkit-dev                       10.0         py36_0     conda-forge
```

주의사항

- 깃허브에서 CUDA 10.2 or greater라고 하지만 12이상일때 빌드 오류 발생. 11.x 권장
- Sudo 권한이 없으므로 apt 명령 사용 불가
- Conda install로 가상환경에 설치
- 가상환경을 만들 때 파이썬 버전을 지정하지 않으면 필수 패키지 일부가 설치되지 않음
ex) conda create -n triton (x) conda create -n triton python=3.10 (o)

```
(triton) hvdrk@aisys-small100:~$ conda install -c conda-forge cudatoolkit-dev=11.7.0
Channels:
 - conda-forge
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/hvdrk/miniconda3/envs/triton

added / updated specs:
 - cudatoolkit-dev=11.7.0
```

2. GNU C++ compiler

버전 확인

- 깃허브에서 g++ 8.3 or greater라고 하지만 CUDA 버전에 따라 오류 발생
- 사용하는 CUDA 라이브러리 경로의 ./targets/x86_64-linux/include/crt/host_config.h에서
사용가능한 gnu 버전 확인

ex)

```
usr > local > cuda-11.2 > targets > x86_64-linux > include > crt > C host_config.h
124
125 #if defined(__ibmxl_vrm__) && !(__ibmxl_vrm__ >= 0x0d010000 && __ibmxl_vrm__ < 0x0d02
126 |      |      |      |      |      |      |      |      |      |      |      |
127 |      |      |      |      |      |      |      |      |      |      |      |
128 #error -- unsupported xlc version! only xlc 13.1 and 16.1 are supported. The nvcc fla
129
130 #endif /* __ibmxl_vrm__ && !(__ibmxl_vrm__ >= 0x0d010000 && __ibmxl_vrm__ < 0x0d02000
131 |      |      |      |      |      |      |      |      |      |      |      |
132 |      |      |      |      |      |      |      |      |      |      |      |
133 #endif /* __powerpc__ */
134
135 #if defined(__GNUC__)
136
137 #if __GNUC__ > 10
138
139 #error -- unsupported GNU version! gcc versions later than 10 are not supported! The
140
141 #endif /* __GNUC__ > 10 */
142
```

- Host library path

```
miniconda3 > envs > triton > pkgs > cuda-toolkit > include > crt > C host_config.h
118 #if defined(__ibmxl_vrm__) && !(__ibmxl_vrm__ >= 0x0d010000 && __ibmxl_vrm__ < 0x0d0
119 |      |      |      |      |      |      |      |      |      |      |      |
120 |      |      |      |      |      |      |      |      |      |      |      |
121 #error -- unsupported xlc version! only xlc 13.1 and 16.1 are supported. The nvcc fl
122
123 #endif /* __ibmxl_vrm__ && !(__ibmxl_vrm__ >= 0x0d010000 && __ibmxl_vrm__ < 0x0d0200
124 |      |      |      |      |      |      |      |      |      |      |      |
125 |      |      |      |      |      |      |      |      |      |      |      |
126 #endif /* __powerpc__ */
127
128 #if defined(__GNUC__)
129
130 #if __GNUC__ > 11
131
132 #error -- unsupported GNU version! gcc versions later than 11 are not supported! The
133
134 #endif /* __GNUC__ > 11 */
135
```

- Conda library path

2. GNU C++ compiler

설치여부 확인 및 설치

- /usr/bin에서 g++-<ver> 확인
- 미설치 시 anaconda.org 에서 gxx를 검색하거나
conda search gxx 로 적정 버전 선택 후 설치
- Conda의 경우 사용하는 env의 ./bin에 설치됨
ex) ~/miniconda3/envs/triton/bin/g++

주의사항

- Conda에 설치한 경우 gnu 컴파일러에서 zlib(압축)와 numa(메모리설정) 라이브러리의 헤더가 삭제됨
- conda install -c conda-forge zlib와 conda install -c conda-forge numactl로 재설치

```
▼ / [SSH: AISYS-SMALL01]
  ▼ usr
    ▼ bin
      ≡ fwupdate
      ≡ fwupdmgr
      ≡ fwupdtool
      ≡ g++
      ≡ g++-10
      ≡ g++-11
      ≡ g++-12
      ≡ gapplication
```

```
(triton) hvdrk@aisys-small00:~$ conda install gxx=10.4.0
Channels:
- conda-forge
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/hvdrk/miniconda3/envs/triton

added / updated specs:
- gxx=10.4.0

The following packages will be downloaded:
```

3. Rust

설치여부 확인 및 설치

- 터미널에 rustc 입력
- 미설치 시 `curl https://sh.rustup.rs -sSf | sh` 로 설치

```
⊗ (triton) hvdrk@aisys-small100:~$ rustc
Command 'rustc' not found, but can be installed with:
snap install rustup # version 1.27.1, or
apt install rustc # version 1.66.1+dfsg0ubuntu1-0ubuntu0.22.04.1
See 'snap info rustup' for additional versions.
○ (triton) hvdrk@aisys-small100:~$ curl https://sh.rustup.rs -sSf | sh
info: downloading installer

Welcome to Rust!

This will download and install the official compiler for the Rust
programming language, and its package manager, Cargo.

Rustup metadata and toolchains will be installed into the Rustup
home directory, located at:
```


4. Environment variable

Environment variable

- 실행 환경을 설정/제어하기 위해 사용
- 경로, 값, 동작 방식 등 정의

주요 환경변수

- PATH: 실행파일을 검색할 디렉토리 목록
- C_INCLUDE_PATH: C 컴파일러가 헤더 파일을 찾는 경로
- LD_LIBRARY_PATH: 동적 라이브러리를 찾는 경로
- **디렉토리의 경우 앞에 있는 값부터 탐색**

ex) cuda-11.2/bin:cuda/bin의 경우 cuda/bin이 아닌
cuda 11.2를 사용

```
(base) hvdrk@aisys-small01:~$ echo $PATH
/usr/local/cuda/bin:/home/hvdrk/.vscode-server/cli/servers/Stable-fabdb6a30b49f79a7aba0f2ad9df9b399473380f/server/bin/remote-cli:/usr/local/cuda/bin:/usr/local/cuda/bin:/home/hvdrk/.cargo/bin:/home/hvdrk/miniconda3/bin:/home/hvdrk/miniconda3/condabin:/usr/local/cuda/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
(base) hvdrk@aisys-small01:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Thu_Jun__6_02:18:23_PDT_2024
Cuda compilation tools, release 12.5, V12.5.82
Build cuda_12.5.r12.5/compiler.34385749_0
```

```
(base) hvdrk@aisys-small01:~$ echo $PATH
/usr/local/cuda-11.2/bin:/usr/local/cuda-11.2/bin:/usr/local/cuda/bin:/home/hvdrk/.vscode-server/cli/servers/Stable-fabdb6a30b49f79a7aba0f2ad9df9b399473380f/server/bin/remote-cli:/usr/local/cuda/bin:/usr/local/cuda/bin:/home/hvdrk/.cargo/bin:/home/hvdrk/miniconda3/bin:/home/hvdrk/miniconda3/condabin:/usr/local/cuda/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
(base) hvdrk@aisys-small01:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Nov_30_19:08:53_PST_2020
Cuda compilation tools, release 11.2, V11.2.67
Build cuda_11.2.r11.2/compiler.29373293_0
```

4. Environment variable

환경변수 설정

- export 명령을 통해 해당 터미널 한정으로 환경변수 재설정 가능
- 영구적으로 설정하려면 ~/.bashrc에 export 명령 추가 후 새 터미널 실행 또는 source ~/.bashrc 입력
- 깃허브 가이드에서 /usr/local/cuda와 \$CXX를 사용하려는 라이브러리에 맞게 수정

주의사항

- 깃허브에서 export PATH="\$PATH:/usr/local/cuda/bin"로 되어있는데, 기본 PATH에 이미 /usr/local/cuda/bin이 포함되어있고, 다른 버전을 사용해야 하는 경우 \$PATH 앞에 사용할 경로를 배치해야 함
- 다른 환경변수도 동일

```
$ .bashrc
1  export PATH="/home/hvdrk/miniconda3/envs/triton/pkgs/cuda-toolkit/bin:$PATH"
2  export C_INCLUDE_PATH="/home/hvdrk/miniconda3/envs/triton/pkgs/cuda-toolkit/include:$C_INCLUDE_PATH"
3  export LIBRARY_PATH="/home/hvdrk/miniconda3/envs/triton/pkgs/cuda-toolkit/lib64:/home/hvdrk/miniconda3/
4  export LD_LIBRARY_PATH="/home/hvdrk/miniconda3/envs/triton/pkgs/cuda-toolkit/lib64:/home/hvdrk/minicond
5  export CXX="/home/hvdrk/miniconda3/envs/triton/bin/g++"
6  # export CXX=/usr/bin/g++-10
7
8  # ~/.bashrc: executed by bash(1) for non-login shells.
9  # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
10 # for examples
```

5. Linux setting

Mem lock

- 메모리에서 디스크로 swap될 경우 심각한 성능 하락 발생
- 올바른 결과를 얻기 위해 이를 방지해야 함
- 조교 이메일을 통해 root 계정으로 사용하는 서버에서 아래 명령 실행 요청

```
sudo cat << EOF >> /etc/security/limits.conf
<username>    soft    memlock    unlimited
<username>    hard    memlock    unlimited
EOF
```

```
etc > security > limits.conf
51  #@faculty    hard    nproc      50
52  #ftp         hard    nproc      0
53  #ftp         -       chroot     /ftp
54  #@student    -       maxlogins   4
55
56  # 2025-01-07 added by mapae: for jolp students
57  hvdrk        soft    memlock    unlimited
58  hvdrk        hard    memlock    unlimited
59  sejinrim     soft    memlock    unlimited
60  sejinrim     hard    memlock    unlimited
61
62  # End of file
63
```

/etc/security/limits.conf

6. Build

Build

- 작업할 디렉토리에서 git clone 명령을 통해 레포지토리 복제
- 이동 후 빌드

```
• (triton) hvdrk@aisys-small00:~$ git clone https://github.com/TU-Berlin-DIMA/fast-interconnects.git
Cloning into 'fast-interconnects'...
remote: Enumerating objects: 7124, done.
remote: Counting objects: 100% (2352/2352), done.
remote: Compressing objects: 100% (1091/1091), done.
remote: Total 7124 (delta 974), reused 2352 (delta 974), pack-reused 4772 (from 1)
Receiving objects: 100% (7124/7124), 94.36 MiB | 13.93 MiB/s, done.
Resolving deltas: 100% (4255/4255), done.
• (triton) hvdrk@aisys-small00:~$ cd fast-interconnects/
• (triton) hvdrk@aisys-small00:~/fast-interconnects$ cargo build --release
   Compiling autocfg v1.0.1
   Compiling libc v0.2.80
   Compiling proc-macro2 v1.0.24
```