# Common Software Installation

牟懋軒

國立清華大學叢集電腦競賽團隊
Student Cluster Competition Team of NTHU

# Outline

1. Intel Application (include CC, FC, MKL, Vtune )
2. ucx
3. openMPI
4. BLAS

# LVM(Logical Volume Manager）

df -h ：可以看到 volume

volume 是 partition 的邏輯單位，會被物理上的硬碟影響

透過 path 去 access

-> 會有 volume 和 path 是因為 user 不在意 file system實際上怎麼管理

-> LVM 一開始不會開到最大，因為可能會有新的 volume 或者操作 RAID

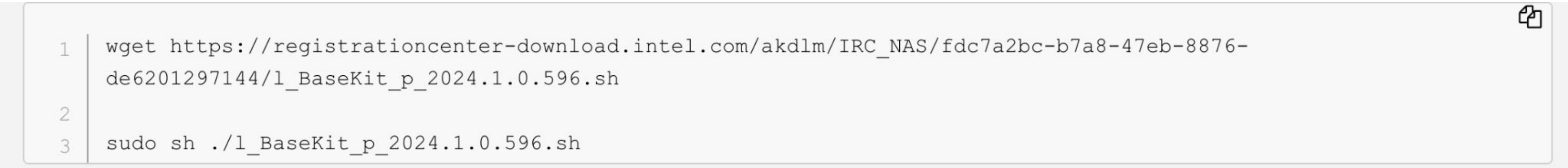
# command

```
sudo lvextend -L +5G /dev/mapper/ubuntu--vg-ubuntu--lv

sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

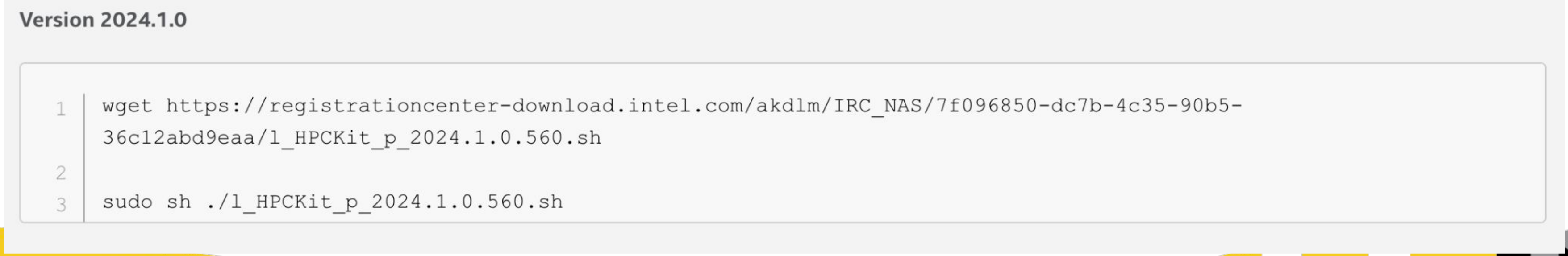
# Install Intel compiler

# Intel

1. Install Intel oneAPI Base ToolKit (include CC, vtune)
2. Install Intel oneAPI HPC ToolKit (include FC)
3. `source /opt/intel/oneapi/setvars.sh --force`



```
wget https://registrationcenter-download.intel.com/akdlm/IRC_NAS/fdc7a2bc-b7a8-47eb-8876-de6201297144/l_BaseKit_p_2024.1.0.596.sh

sudo sh ./l_BaseKit_p_2024.1.0.596.sh
```



Version 2024.1.0

```
wget https://registrationcenter-download.intel.com/akdlm/IRC_NAS/7f096850-dc7b-4c35-90b5-36c12abd9eaa/l_HPCKit_p_2024.1.0.560.sh

sudo sh ./l_HPCKit_p_2024.1.0.560.sh
```

# Additional Information

Guide : To add flags for customized installation

`-s or --silent :` Install in non-interactive mode, but the error messages lack clarity.

compiler name : `echo $PATH` and go to `/opt/intel/oneapi/compiler/2024.1/bin`

`C compiler : icx | C++ compiler : icpx     (without icc)`

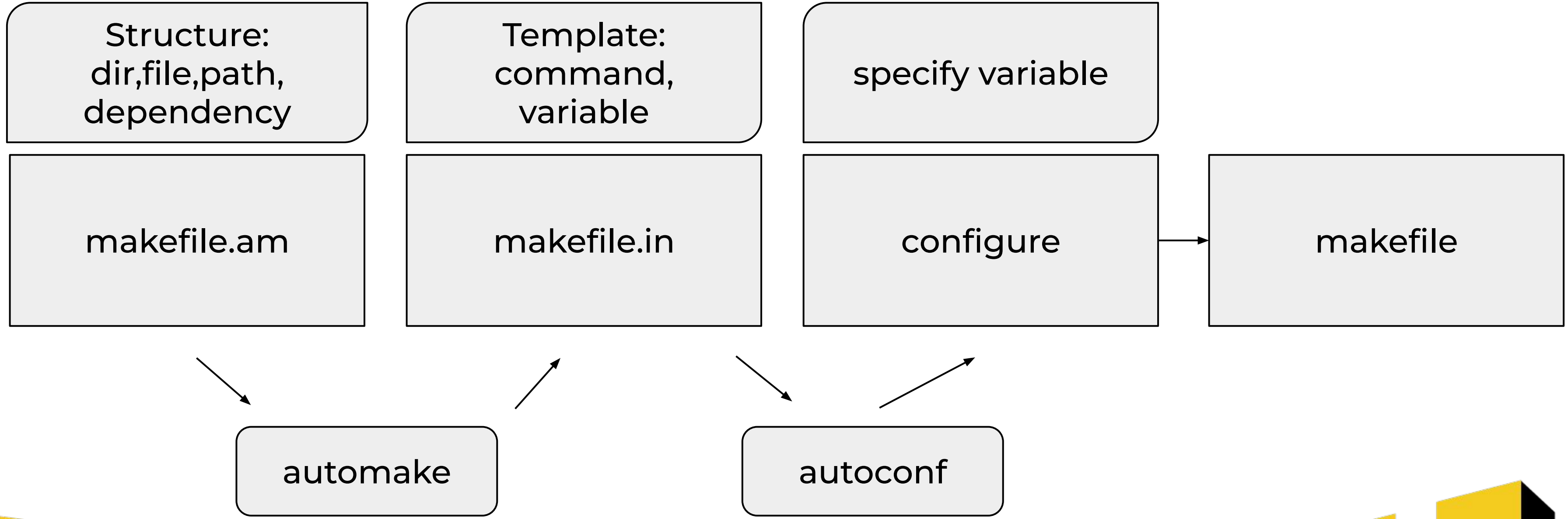
`Fortran compiler : ifort and ifx`

# Install UCX

```
git clone https://github.com/openucx/ucx.git

which autoreconf

sudo apt-get install autoconf

which automake

sudo apt-get install libtool

./autogen.sh

cd contrib/

./configure-release -h
```

# automake

## GNU Autotools



# autoreconf

Condition:

1. autotools upgrade
2. modify file

autoreconf help generate a new configure

有時候 autotools 會自動偵檔案是否有變動，會主動要求重新 automake，

像是 cp 一整個 dir 的時候，time stamp 會跑掉

# libtool

不同 OS 處理 shared lib 的方法不同, shared lib 的名稱也可能有一些變化

所以用一個 interface 來 access lib

可以想成比較 readible 的 #ifdef

# configure

```
CC="icx -O3" CXX="icpx -O3" FC="ifx -fPIC -O3" \

CXXFLAGS="-qopenmp -Wno-tautological-constant-compare $CXXFLAGS" CFLAGS="-qopenmp -Wno-tautological-constant-compare $CFLAGS" \

./contrib/configure-release \

--prefix=$HOME/ucx/built \

--disable-logging \

--disable-debug \

--disable-assertions \

--disable-dependency-tracking \

--disable-params-check \

--without-java \

--enable-shared \

--enable-static \

--with-avx
```

# compile

```
make clean

make -j

make install -j
```

# modify env var

```
PATH=$HOME/ucx/built/bin:$PATH

LIBRARY_PATH=$HOME/ucx/built/lib:$LIBRARY_PATH

LD_LIBRARY_PATH=$HOME/ucx/built/lib:$LD_LIBRARY_PATH

CPATH=$HOME/ucx/built/include:$CPATH
```

check：`ucx_info -d`

# Install openMPI

# step

1. ./configure -h
2. ./configure
3. make -j
4. make install -j

if need, `sudo apt-get install libpmix-dev`

# configure

```
CC="icx -O3" CXX="icpx -O3" FC="ifx -O3 -fPIC" \
LDFLAGS="-Wl,-rpath,/opt/intel/oneapi/compiler/2024.1/lib/libimf.so" \
./configure \
--prefix=$HOME/openmpi-5.0.1/built \
--enable-mpi1-compatibility \
--enable-orterun-prefix-by-default \
--with-ucx=$HOME/ucx/built \
```

# modify env var

```
export OMPI_CC=icx

export OMPI_CXX=icpx

export OMPI_FC=ifx
```

```
PATH=$HOME/openmpi-5.0.1/built/bin:$PATH

LIBRARY_PATH=$HOME/openmpi-5.0.1/built/lib:$LIBRARY_PATH

LD_LIBRARY_PATH=$HOME/openmpi-5.0.1/built/lib:$LD_LIBRARY_PATH

CPATH=$HOME/openmpi-5.0.1/built/include:$CPATH
```

check: mpicxx --version, mpicc --version, mpifort --version

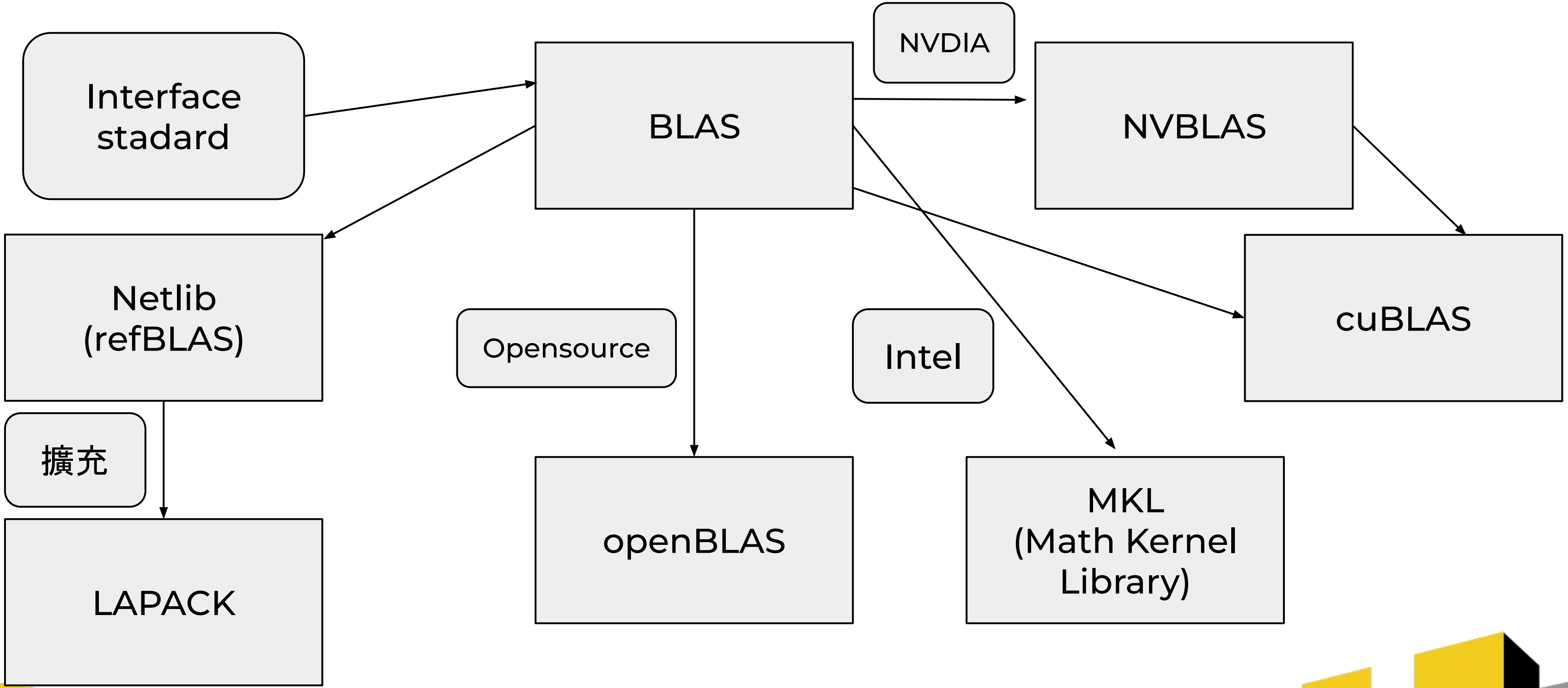
# if env var doesn’t work

$HOME/openmpi-5.0.1/built/share/openmpi/mpicc-wrapper-data.txt

```
compiler=/opt/intel/oneapi/compiler/2024.1/bin/icx -O3
```

# MKL and BLAS

# 數學 library 的各種關係


Interface stadard
BLAS
NVDIA
NVBLAS
Netlib (refBLAS)
Opensource
Intel
cuBLAS
擴充
openBLAS
MKL (Math Kernel Library)
LAPACK