

# ISSPのkuguiとohtakaでのインストール

---

## kuguiでのインストール

1. ログイン。
2. [mise](#)あるいは[anaconda](#)を用いてpythonとツールをインストール
3. [~/ .bashrc](#)の書き換え

[~/ .bashrc](#)の最後に

```
ulimit -s unlimited
export PATH=$HOME/bin:$PATH
module purge
module load nvhpc-nompi/24.7 openmpi_nvhpc compiler-rt tbb mkl
if which nvidia-cuda-mps-control > /dev/null 2>&1 ; then
    export CUDA_MPS_PIPE_DIRECTORY=$(pwd)/nvidia-mps-$(hostname)
    export CUDA_MPS_LOG_DIRECTORY=$(pwd)/nvidia-log-$(hostname)
    echo "start nvidia-cuda-mps-control at" $(hostname)
    nvidia-cuda-mps-control -d
fi
```

を挿入。

[!TIP]

Script for loading the required [modules](#) and starting MPS.

## 4.ecaljのインストールとテスト

```
git clone https://github.com/tkotani/ecalj.git
cd ecalj
```

そのあとインストール+テストのコマンドを打ちます。コンソールで

```
qsub jobinstall_kugui.sh
```

です。これを実行すると、コンパイルのあとテスト計算が進んでいく。[jobinstall\\_kugui.sh](#)ではCPU版とGPU版の両方がインストールされます。

InstallAllは[\\$HOME/bin](#)ディレクトリを作りそこへecaljのバイナリやスクリプトのコピーを行います。

困る場合は現在のコピーのバックアップをとったうえでecaljに上書きさせる、

というのでもいいかもしれません。あるいはInstallAllの中でbinを検索し書き直してください。

そこにパスを通しておく必要があります。テストはOKを出しながら進んでいきます  
以下のようなエラーっぽいメッセージをだすのですが無視してください。正常に動いています。

```
[cpu121:54969] 7 more processes have sent help message help-mpi-
common-cuda.txt / dlopen failed
[cpu121:54969] Set MCA parameter "orte_base_help_aggregate" to 0 to
see all help / error messages
-----
-----
The library attempted to open the following supporting CUDA
libraries,
but each of them failed.  CUDA-aware support is disabled.
libcuda.so.1: cannot open shared object file: No such file or
directory
libcuda.dylib: cannot open shared object file: No such file or
directory
/usr/lib64/libcuda.so.1: cannot open shared object file: No such
file or directory
/usr/lib64/libcuda.dylib: cannot open shared object file: No such
file or directory
If you are not interested in CUDA-aware support, then run with
--mca opal_warn_on_missing_libcuda 0 to suppress this message.  If
you are interested
in CUDA-aware support, then try setting LD_LIBRARY_PATH to the
location
of libcuda.so.1 to get passed this issue.
```

テストの最後は

```
PASSED! ni_crpa/Screening_W-v_crpa.h
PASSED! srvo3_crpa/Screening_W-v.h
PASSED! srvo3_crpa/Screening_W-v_crpa.h
OK! ALL PASSED ===
    See work/summary.txt
Elapsed time for make          : 51 seconds
Elapsed time for testecalj.py: 409 seconds
```

という感じで終了します。これでインストールとテスト完了です。合計10分ぐらいです。  
これができたら次に進んでください。コンパイル、テストと進行していくのですが、  
テストに突入すれば、

```
tail -f ecalj/SRC/TestInstall/summary.txt
```

を見ていれば、順にテストが進んでいくのが確認できます。最後にPASSED! srvo3\_crpa/Screening\_W-  
v\_crpa.h

と表示され、FAILとかエラーがなければOKです。

#### 4.GPU計算テスト

ecalj/Samples/Samples\_ISSP/inas2gasb2\_kugui  
に移動して、

```
qsub job_kugui.sh
```

を実行。cat lgwscしてみて

```
===== QSGW iteration end   iter 2 ===  
OK! ===== All calculation finished for   gwsc =====
```

で終了していればOKです。

#### 6. 使い方メモ

##### 使い方メモ

### ohtakaでのインストール

ほぼ同様です。~/.bashrcの最後に

```
ulimit -s unlimited  
export PATH=$HOME/bin:$PATH  
module purge  
module load openmpi/4.1.5-oneapi-2023.0.0-classic
```

PROF

と書いておきます。インストールは

```
sbatch jobinstall_ohtaka.sh
```

です。の段階でその最後がOK! ALL PASSED ! が見れないならインストールできてないです。

それを確認後、計算テストは  
ecalj/Samples\_ISSP/inas2gasb2  
にて

```
sbatch job_ohtaka.sh
```

を行ってください。GPUテストはinas2gasb2\_ohtakaにあります。

## ISSPシステムでのカスタマイズについて

[ecaljソースコードの変更点](#)

あるいはecalj packageのecalj/job\_install\_kugui.sh  
を見てください。