

下記の手順でPrivate Fugakuを実行可能な環境を構築する

1. AWSのEC2インスタンス作成
2. ParallelClusterのインストール
3. ParallelClusterでクラスタを作成するための定義ファイルの作成
4. クラスタの作成
5. Singularityのインストール
6. アプリの実行

※AWSのEC2インスタンス作成はユーザーにより実施とし、上記2.以降の手順を説明する。

## AWS Document

- 「Install AWS ParallelCluster in a virtual environment (recommended)」
- <https://docs.aws.amazon.com/parallelcluster/latest/ug/install-v3-virtual-environment.html>

# ParallelCluster定義ファイルの作成

## AWS Document

- 「Cluster configuration file」
- <https://docs.aws.amazon.com/parallelcluster/latest/ug/cluster-configuration-file-v3.html>
- <https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues>

## AWS Document

- 「Configure and create a cluster with the AWS ParallelCluster command line interface」
- <https://docs.aws.amazon.com/parallelcluster/latest/ug/install-v3-configuring.html>

# Singularityのインストール

## SingularityCE Admin Guide

- 「Install from Source」
- <https://docs.sylabs.io/guides/4.3/admin-guide/installation.html#install-from-source>

# コンテナのダウンロードから実行までの流れ

Generate a Singularity SIF file or Spack environment with the selected applications.

Please select the applications you want to install.

<input type="checkbox"/> AutoDock-Vina	<input type="checkbox"/> CP2K	<input type="checkbox"/> CPMD	<input type="checkbox"/> Darshan-runtime	<input type="checkbox"/> FDS	<input type="checkbox"/> FFmpeg
<input type="checkbox"/> FrontISTR	<input type="checkbox"/> GENESIS	<input type="checkbox"/> GROMACS	<input type="checkbox"/> GSL	<input type="checkbox"/> Gnuplot	<input type="checkbox"/> GrADS
<input type="checkbox"/> Julia	<input type="checkbox"/> LAMMPS	<input type="checkbox"/> Metis	<input type="checkbox"/> OpenBabel	<input type="checkbox"/> OpenFoam-org	<input type="checkbox"/> OpenFoam
<input type="checkbox"/> OpenMX	<input type="checkbox"/> PETSc	<input type="checkbox"/> POV-Ray	<input type="checkbox"/> Paraview	<input type="checkbox"/> Parmetis	<input type="checkbox"/> Quantum-Espresso
<input type="checkbox"/> SALMON-tddft	<input type="checkbox"/> SCALE	<input type="checkbox"/> WRF	<input type="checkbox"/> py-Tensorflow	<input type="checkbox"/> py-Torch	<input type="checkbox"/> py-ase
<input type="checkbox"/> py-matplotlib	<input type="checkbox"/> py-mpi4py	<input type="checkbox"/> py-netcdf4	<input type="checkbox"/> py-numpy	<input type="checkbox"/> py-pandas	<input type="checkbox"/> py-sckit-learn
<input type="checkbox"/> py-scipy	<input type="checkbox"/> py-toml	<input type="checkbox"/> py-xarray	<input type="checkbox"/> tmux		

**① アプリを選択する**

**② 「Generate SIF file」をクリックする**

Processing Status  
The task is still waiting to start.  
Elapsed time : 00:00



Processing Status

The following applications will be included in the SIF file:

-----  
GROMACS  
-----

The SIF file size is 1.7Gbytes

To download the file to your server, please use the following command:

```
singularity pull https://virtual.fugaku.r-ccs.riken.jp/container/SIF/2lkCB2/virtual_fugaku.sif
```

**③ 「Copy」をクリックする**

To download the file to your current machine, please click the link below:  
[Download the file.](#)

```
$ singularity pull https://virtual.fugaku.r-ccs.riken.jp/container/SIF/2lkCB2/virtual_fugaku.sif
```

**④ 自身のインスタンス上でペーストする**



```
#!/bin/bash
#SBATCH -p satf01
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --ntasks-per-node=8
#SBATCH --cpus-per-task=8

SIFFILE=./virtual_fugaku.sif
export SINGULARITY_BIND=/opt/amazon,/usr/lib64/libefa.so.1,/usr/lib64/libibverbs.so.1

export OMP_MAX_THREADS=8

time -p mpiexec -n 8 singularity run ${SIFFILE} gmx_mpi mdrun -s benchMEM.tpr -nsteps 5000 -resetway
```

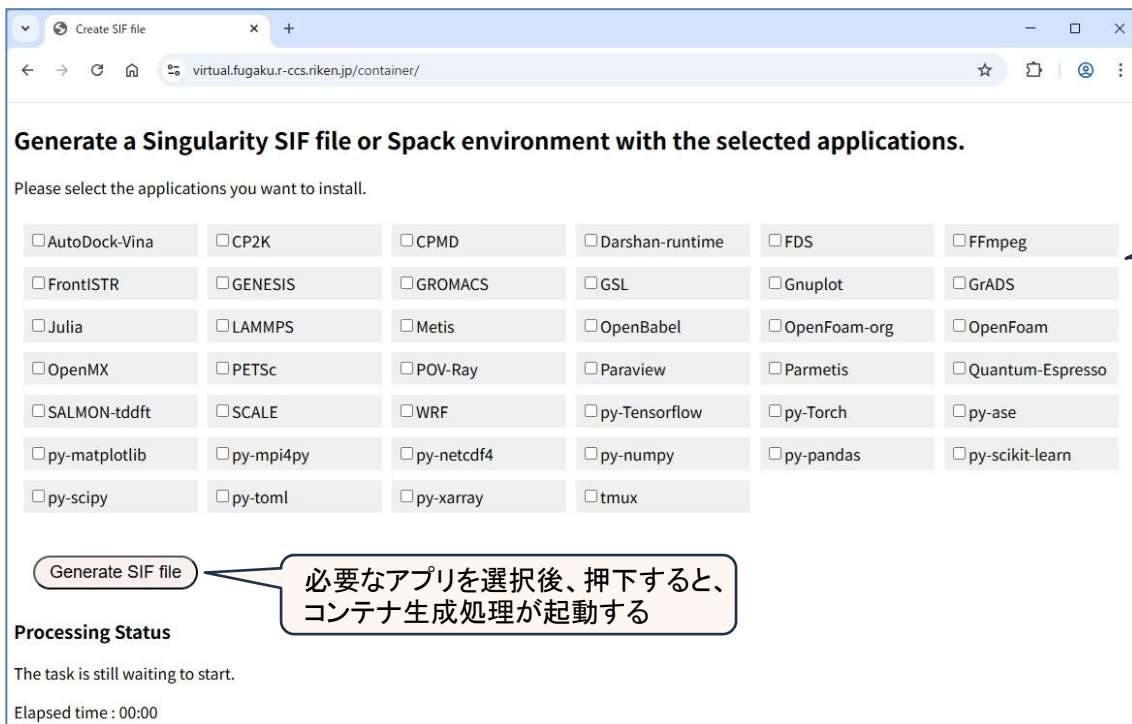
**⑤ Singularityで起動する**

# コンテナのダウンロード [画面遷移①]

- virtual\_fugaku コンテナサーバからSingularityコンテナイメージをダウンロードする

<https://virtual.fugaku.r-ccs.riken.jp/container>

- コンテナサーバでは、40種のアプリを用意しており、自由に組み合わせてSingularityコンテナイメージを生成することができる。



Generate SIF file

Processing Status

The task is still waiting to start.

Elapsed time : 00:00

コンテナに取り込むアプリを選択する  
(複数選択可能)

必要なアプリを選択後、押下すると、  
コンテナ生成処理が起動する

# コンテナのダウンロード [画面遷移②]

画面遷移② : コンテナ生成処理が起動中の場合

前ページの「Generate SIF file」を押下すると、該当するコンテナを検索し、コンテナが存在しない場合は、コンテナ生成処理が起動され、以下の画面に切り替わる

該当するコンテナが既に存在する場合は、コンテナ生成処理は起動されず、「コンテナのダウンロード[画面遷移④]」の画面に切り替わる



The screenshot shows a web browser window with the address bar displaying `virtual.fugaku.r-ccs.riken.jp/container/`. The page content is as follows:

**Processing Status**

The following applications will be included in the SIF file:

-----  
GROMACS  
-----

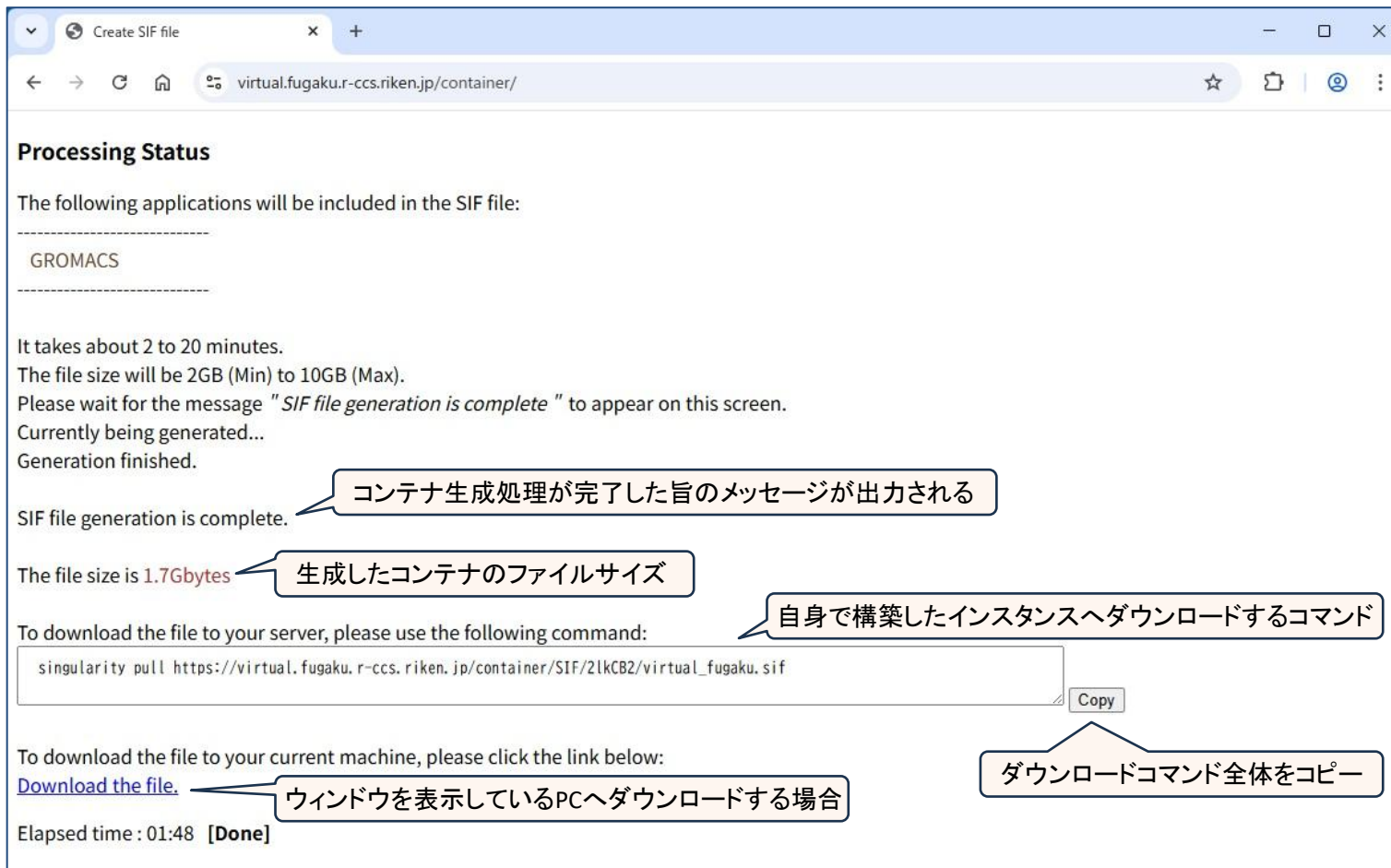
It takes about 2 to 20 minutes.  
The file size will be 2GB (Min) to 10GB (Max).  
Please wait for the message " SIF file generation is complete " to appear on this screen.  
Currently being generated...

Elapsed time : 00:10

コンテナ生成処理が完了するまで、経過時間がカウントされる  
生成処理が完了すると、次ページに示す画面が表示される

# コンテナのダウンロード [画面遷移③]

画面遷移③ : コンテナ生成処理が完了し、ダウンロード可能になった場合



**Processing Status**

The following applications will be included in the SIF file:

-----  
GROMACS  
-----

It takes about 2 to 20 minutes.  
The file size will be 2GB (Min) to 10GB (Max).  
Please wait for the message "SIF file generation is complete" to appear on this screen.  
Currently being generated...  
Generation finished.

SIF file generation is complete.

The file size is 1.7Gbytes

To download the file to your server, please use the following command:

```
singularity pull https://virtual.fugaku.r-ccs.riken.jp/container/SIF/2lkCB2/virtual_fugaku.sif
```

[Download the file.](#)

Elapsed time : 01:48 [Done]

コンテナ生成処理が完了した旨のメッセージが出力される

生成したコンテナのファイルサイズ

自身で構築したインスタンスへダウンロードするコマンド

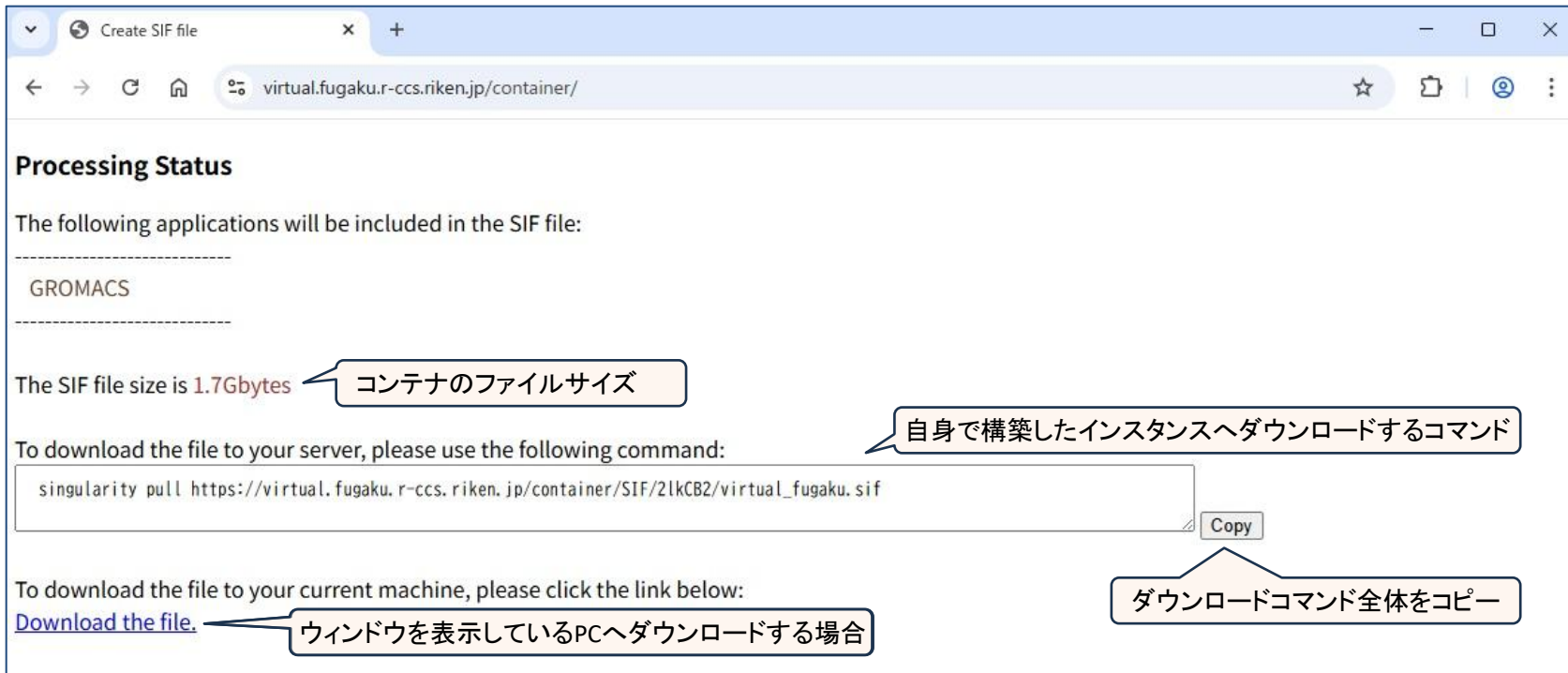
ダウンロードコマンド全体をコピー

ウインドウを表示しているPCへダウンロードする場合

# コンテナのダウンロード [画面遷移④]

画面遷移④ : サーバ上にコンテナが存在しており、ダウンロード可能な場合

「画面遷移①」で選択したアプリのコンテナがサーバ上に存在している場合、瞬時に本画面に遷移する。(画面遷移①→④)  
生成したコンテナは、いつでもダウンロードできるようにサーバ上に保持されており、画面遷移①から本画面(④)へ移行する。



**Processing Status**

The following applications will be included in the SIF file:

-----  
GROMACS  
-----

The SIF file size is 1.7Gbytes

To download the file to your server, please use the following command:

```
singularity pull https://virtual.fugaku.r-ccs.riken.jp/container/SIF/21kCB2/virtual_fugaku.sif
```

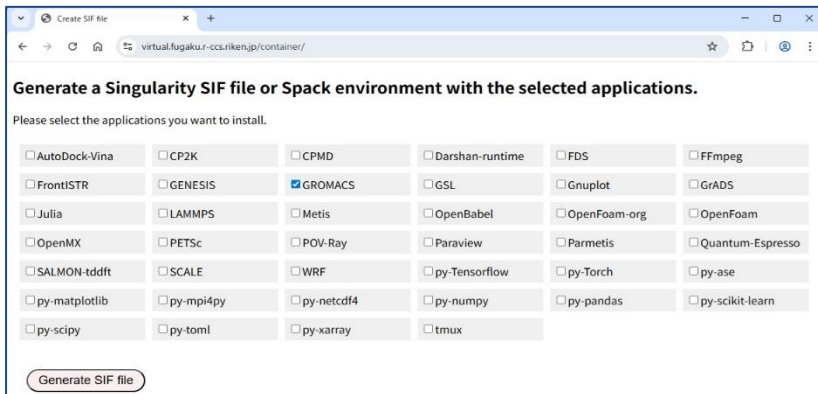
To download the file to your current machine, please click the link below:  
[Download the file.](#)

Annotations:

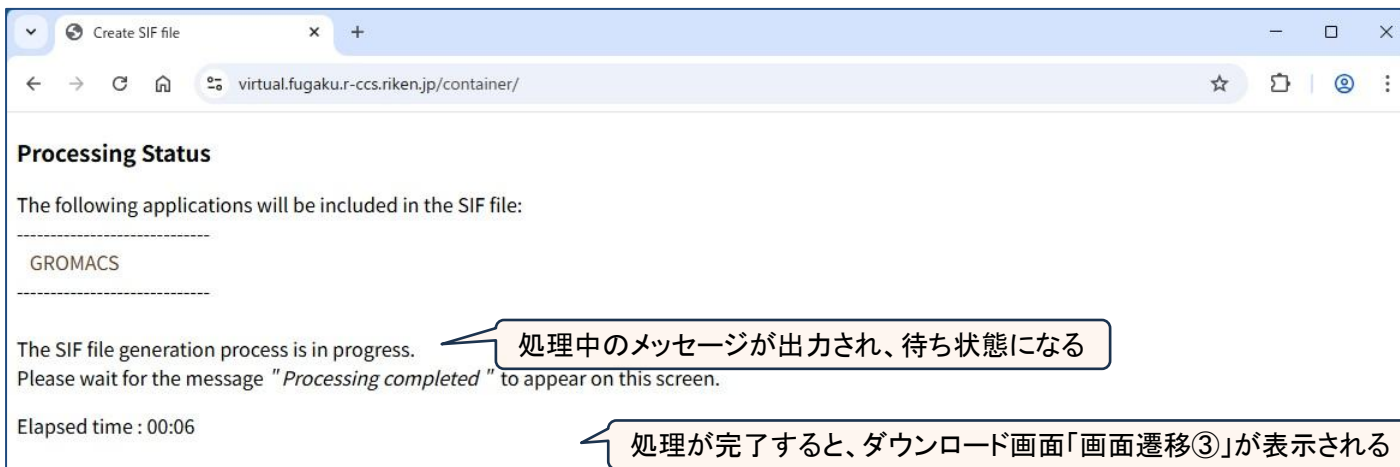
- コンテナのファイルサイズ
- 自身で構築したインスタンスへダウンロードするコマンド
- Copy
- ダウンロードコマンド全体をコピー
- ウィンドウを表示しているPCへダウンロードする場合

# コンテナのダウンロード [画面を閉じた場合]

コンテナ生成処理が起動され、「画面遷移②」で待機中に、ブラウザを閉じた場合でも、サーバ上でコンテナ生成処理は継続している。よって、ブラウザを閉じた直後に再度、アプリ選択画面(画面遷移①)で対象のアプリを選択すると、処理を継続している旨の画面が表示される。処理が完了すると、画面遷移③と同等の表示になり、ダウンロード可能になる。なお、再度アクセスしたときに、コンテナ生成処理が完了していた場合は、「画面遷移④」が表示される。



gromacsを選択し、「Generate SIF file」を押下後、ブラウザを閉じて、再度、ブラウザを立ち上げて、gromacsを選択した状態



処理中のメッセージが出力され、待ち状態になる

処理が完了すると、ダウンロード画面「画面遷移③」が表示される

# アプリの実行例1 : genesis

- 入力データ

[https://github.com/genesis-release-r-ccs/genesis\\_benchmark\\_input](https://github.com/genesis-release-r-ccs/genesis_benchmark_input)

- 実行スクリプト

```
#!/bin/bash
#SBATCH -p satf01
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=8
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH -J test_genesis

SIFFILE=~/.virtual_fugaku.sif

export SINGULARITY_BIND=${PWD},/opt/amazon,/usr/lib64/libbfa.so.1,/usr/lib64/libibverbs.so.1
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}

cd npt/genesis1.6_2.5fs/jac_amber
mpiexec -n ${SLURM_NTASKS} singularity run ${SIFFILE} spdyn p${SLURM_NTASKS}.inp
```

- 実行結果

```
Output_Time> Averaged timer profile (Min, Max)
total time      =    21.991
  setup         =     0.852
  dynamics      =    21.138
```

【以降省略】

# アプリの実行例2 : gromacs

- 入力データ

[https://ftp.gromacs.org/pub/benchmarks/ADH\\_bench\\_systems.tar.gz](https://ftp.gromacs.org/pub/benchmarks/ADH_bench_systems.tar.gz)

- 実行スクリプト

```
#!/bin/bash
#SBATCH -p satf01
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=4
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH -J test_gromacs

SIFFILE=~virtual_fugaku.sif

export SINGULARITY_BIND=/opt/amazon,/usr/lib64/libefa.so.1,/usr/lib64/libibverbs.so.1

mpiexec -n 1 singularity run ${SIFFILE} gmx_mpi grompp -f pme_verlet.mdp -c conf.gro -p topol.top -o ions.tpr
mpiexec -n ${SLURM_NTASKS} singularity run ${SIFFILE} gmx_mpi mdrun -ntomp ${SLURM_CPUS_PER_TASK} -s ions.tpr
```

- 実行結果

	Core t (s)	Wall t (s)	(%)
Time:	1825.922	57.060	3200.0
	(ns/day)	(hour/ns)	
Performance:	30.287	0.792	

# アプリの実行例3 : scale

- 入力データ

<https://scale.riken.jp/archives/scale-5.4.5.tar.gz>

- 実行スクリプト

```
#!/bin/bash
#SBATCH -p satf01
#SBATCH --ntasks=2
#SBATCH --cpus-per-task=8
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH -J test_scale

SIFFILE=~/.virtual_fugaku.sif

export SINGULARITY_BIND=${PWD},/opt/amazon,/usr/lib64/libbfa.so.1,/usr/lib64/libibverbs.so.1

cp -pr ../scale-5.4.5/scale-rm/test/tutorial/ideal/* .

#初期値の作成
cp sample/init_R20kmDX500m.conf ./init_R20kmDX500m.conf
mpirun -n ${SLURM_NTASKS} singularity run ${SIFFILE} scale-rm_init init_R20kmDX500m.conf

#シミュレーションの実行
cp sample/run_R20kmDX500m.conf ./run_R20kmDX500m.conf
mpirun -n ${SLURM_NTASKS} singularity run ${SIFFILE} scale-rm run_R20kmDX500m.conf

#後処理
cp sample/net2g_R20kmDX500m.conf ./net2g_R20kmDX500m.conf
mpirun -n ${SLURM_NTASKS} singularity run ${SIFFILE} net2g net2g_R20kmDX500m.conf
```

- 実行結果

