

ハンズオンで学ぶ計算物理化学の基礎 Linux入門・スパコン体験

2025年 9月 24日
公益財団法人計算科学振興財団



1. 公益財団法人 計算科学振興財団(FOCUS)の概要
2. スパコンとは
3. Linuxについて
4. FOCUSスパコンについて
5. 富岳、HPCI計算資源について
6. FOCUSスパコンの利用方法 (実習)
7. LAMMPSの実行 (実習)



1. 公益財団法人 計算科学振興財団 (FOCUS) の概要



理化学研究所
計算科学研究センター(R-CCS)
スーパーコンピュータ「富岳」

FOCUSによる
富岳活用支援サービス



(産業界の意見集約・提言、HPCI)

(広報、セミナー、出展、見学)

産業利用への
環境整備

普及・啓発活動

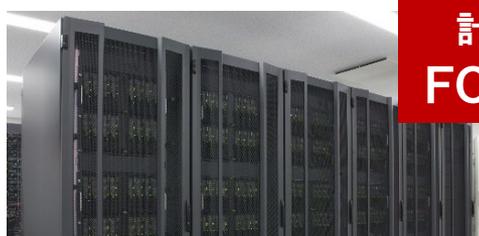


人材開発・研究活動

(講習会、新分野利用開拓)

シミュレーション技術の
普及による産業活性化

(コンサルテーション、企業訪問、FOCUSスパコン)



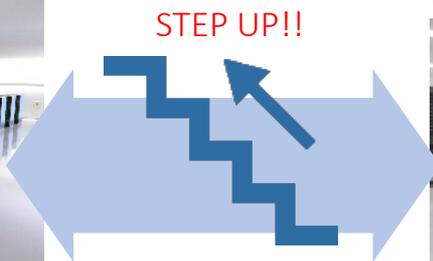
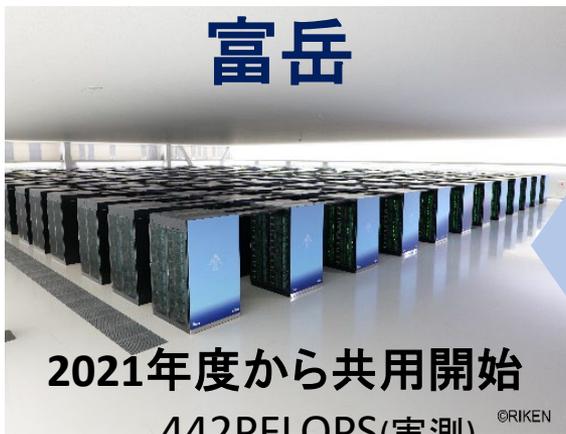
計算科学振興財団
FOCUSスパコン





1. 「富岳」とFOCUSスパコン

総計算資源の10〜15%程度を産業界が利用



FOCUSスパコン



2011年4月から稼働
653TFLOPS (理論)



理化学研究所「富岳」施設
計算科学研究センター R-CCS

高度計算科学研究支援センター
計算科学振興財団 **FOCUS**

日本で数少ない産業利用向け公的スパコン



2. スパコンとは

スパコンとは

- 巨大並列計算機

```
do i = 1, 1000000  
  c(i) = a(i) + b(i)  
enddo
```

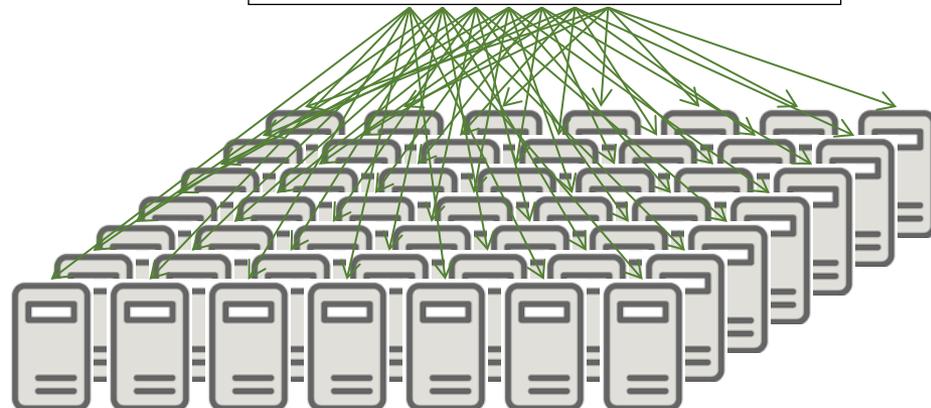


パソコン
計算を1台で処理



大量の計算には多くの時間が必要

```
do i = 1, 1000000  
  c(i) = a(i) + b(i)  
enddo
```



スパコン (巨大並列計算機)
計算をたくさんのパソコン (ノード)
に分散して処理



大量の計算でも短時間で処理可能



2. スパコンとは(2)

例えば「富岳」コンピュータ(2021年3月共用開始)

- ・ 1ノード性能 48コア, 3.3792GFLOPS, 32GBメモリ
- ・ ノード数 158,976
- ・ 総理論演算性能 **537ペタFLOPS (53.7京回/秒)**
- ・ 総メモリ容量 4.85 ペタByte
- ・ インターコネクトTofu (6次元メッシュ/トーラス)

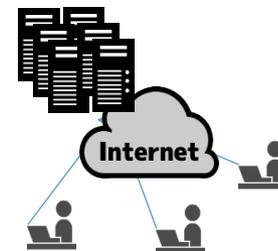
コンピュータの処理性能を
FLOPS (フロップス)
という単位で表します



量子コン連携
(2025.6月～研究開始)

スパコン利用

- みんなで使う（マルチタスク、マルチユーザ）



ジョブを実行する（*batch*処理と対話処理）

batch処理 ファイル等に一連の処理を記述しておき、一括で指示を送り、コンピュータを操作
対話処理 ディスプレイ等に表示しつつコンピュータを対話的に操作

みんなでジョブを実行する（ジョブ管理システム）

複数ジョブの起動や終了を一括管理
たくさんのユーザが使うので、資源（ノード（CPU, メモリなど））の割り当てや
計算の順番待ちの制御を行う

SLURM, LSF, SGE, PBS/Torque 等

※FOCUSスパコンはSLURMを採用しています

① 大規模な系の計算

PC/ワークステーションでは不可能な規模の計算が可能

② 高速計算

PC/ワークステーションでは現実的な時間での計算が不可能

例) PC/ワークステーションで3日 -> スパコン利用で3時間

③ 複数計算の一括処理

パラメータを変えた複数ケースの計算

例) PC/ワークステーションで1ケース3時間

-> スパコン利用で100ケース3時間

④ 自社計算機資源の代替

繁忙期等に補助計算機として活用

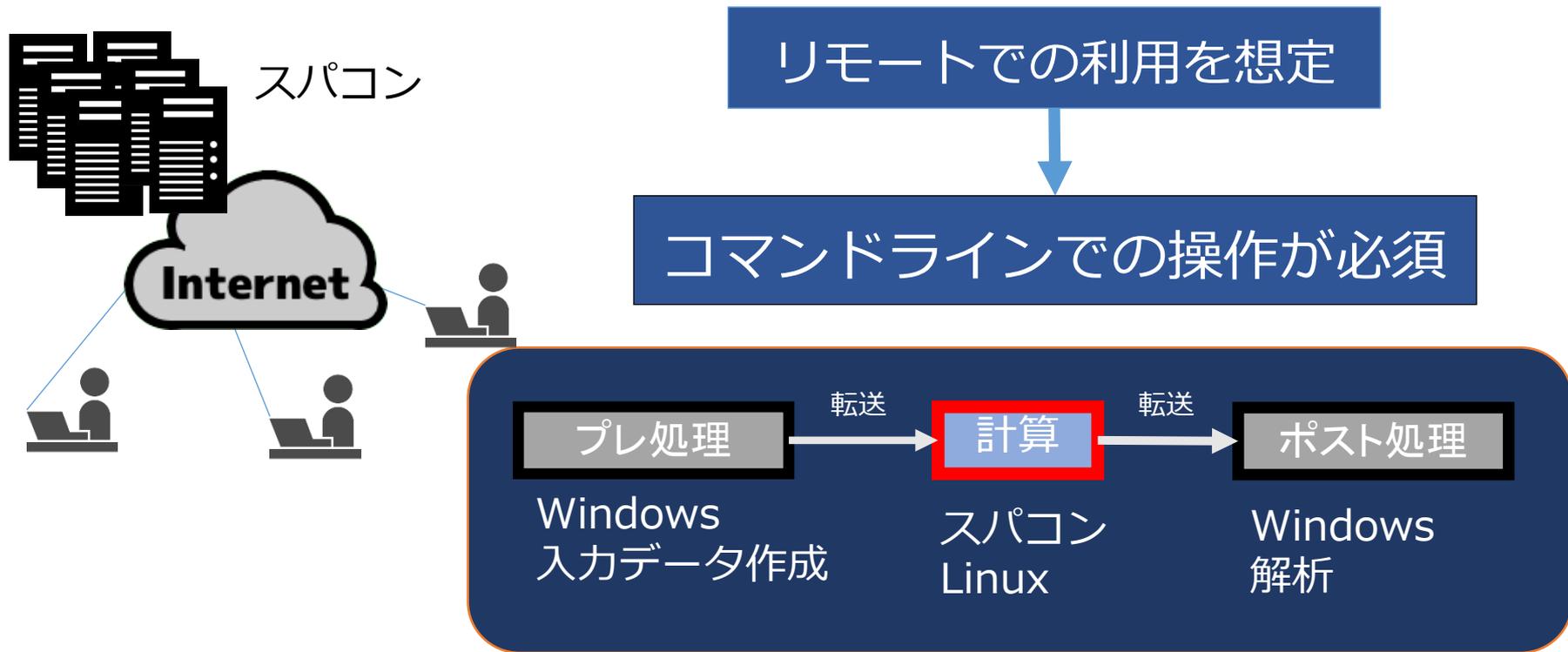
Linuxで動作するソフトウェアと、ある程度のLinuxスキルは必要

そもそも、使用するプログラムが、

1. Linuxで動作すること
2. C/C++, Fortranで書かれたプログラムであること
3. 並列化対応（MPI、OpenMP等）されていること
4. 利用者は最低限のLinux操作のスキルが必要
->CCNV、Winmostar等GUI操作可能なものもある



3. Linuxについて



ほとんどのスパコンは *Linux* ベース
スパコン利活用のため
Linuxコマンドの習得が必要



3. Linuxについて(2)

• Unix

1969年 米国AT&Tのベル研究所においてに開発開始

1971年 First Edition 公開開始



• GNU

1983年 開発開始

GNU's Not Unix

フリーソフトウェアのみによる「完全なUnix互換ソフトウェアシステム」

• Linux

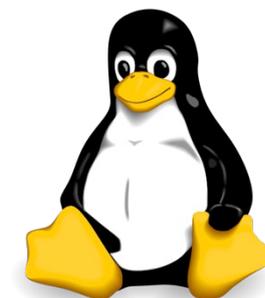
1991年 ヘルシンキ大学@フィンランド 開発開始

開発者 リーナス・ベネディクト・トーバルズ (Linus Benedict Torvalds)

本来 Linux とは Linuxカーネルを指す

ライセンス形態 General Public License (GPL)

→パソコンから大型のスパコンまでUNIXの機能がfreeで利用可能に





3.1 様々なディストリビューション

RedHat系

SUSE系

debian系

ARCH系

<https://distrowatch.com/>



Windows Subsystem for Linux

LinuxをWindows上で実行するための仕組み

マイクロソフトが無料提供

Microsoft Store

ubuntu

スクリーンショット

Ubuntu
Canonical Group Limited

開く

4.2 ★ 55
平均 評価

Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications...

開発者ツール

説明

Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux without leaving Windows.

Key features:

- Efficient command line utilities including bash, ssh, git, apt, npm, pip and many more
- Manage Docker containers with improved performance and startup times
- Leverage GPU acceleration for AI/ML workloads with NVIDIA CUDA
- A consistent development to deployment workflow when using Ubuntu in the cloud
- 5 years of security patching with Ubuntu Long Term Support (LTS) releases

表示数を増やす

評価およびレビュー

4.2 ★ 55 件の評価

Microsoft Store

almalinux

スクリーンショット

AlmaLinux 9
AlmaLinux OS Foundation

入手

5.0 ★ 7
平均 評価

AlmaLinux 9 for WSL all your Linux needs right within Windows. Leverage the power of the EL ecosystem and power all your development an...

開発者ツール

説明

AlmaLinux 9 for WSL all your Linux needs right within Windows. Leverage the power of the EL ecosystem and power all your development an...

AlmaLinux is the Community Owned and Governed Enterprise-Grade Linux OS and is 1:1 RHEL

評価およびレビュー

5.0 ★ 7 件の評価

すべて表示

このバージョンでの新機能

Microsoft Storeから入手可能



4. FOCUSスパコンについて（特長）

利用制度

- 利用は **産業利用** に特化（アカデミアの利用も産業利用に限る）
- 申請受付は随時、申請受理後 通常 **3業務日** でアカウント発行
- 利用成果の公開は **不要**（組織名は公開）・知財権は利用者に帰属

仕様・ 利用形態

- 種類豊富な **12種** の演算ノード群、実行性の高い **Linux OS** を採用
- コンパイラ、並列APIからユーティリティまでを備えた **快適環境**
- 利用者による **アプリ導入が可能**（商用・オープンソース・自社コードなど）

従量利用

- 演算ノードに空きがあれば、**随時利用** いただけます
- 利用料は、**使った分だけ課金** されます（ノード時間単価×ノード数×時間）
- **最長168時間使えるキュー** を設定（最長実行時間はシステムごとに異なります）

料金体系

| | | | | | | |
|---|---|--|---|---|---|---|
| <p>アカウント 利用料</p> <p>1アカウントあたり ¥10,000 × アカウント数 (年度あたり)</p> | + | <p>システム利用料 (演算ノード)</p> <p>従量制 利用単価×ノード数 ×時間</p> <p>期間占有 (1日、1月、1年)</p> | + | <p>ストレージ 利用料</p> <p>【ホーム領域】 1課題につき、 年度あたり 10GB無償 追加は有償</p> | + | <p>フロントエンド サーバ利用料</p> <p>無償 専用フロントエンド サーバは有償</p> |
|---|---|--|---|---|---|---|



4.1 種類豊富な演算ノード群 (全12システム)

| | ノード数 | ホストCPU、デバイス | コア数 | メモリ (コアあたり) | ローカル ディスク | ノード間 通信 | 理論演算性能 (GFLOPS) | 備考 |
|---------------|--------------|---|-----|------------------|----------------|-------------------------------|--------------------|---------------------------------------|
| Fシステム | 60 | Intel Xeon E5-2698v4 2.2GHz × 2基 | 40 | 128GB (3.2GB) | 2000GB ※HDD | 56Gbps InfiniBand FDR | 1152 | ノード内並列・ノード間 高並列 兼用 |
| Zシステム | 24 | Intel Xeon Gold 6230 2.1GHz (Turbo Boost時最大 3.9GHz) × 2基 | 40 | 192GB (4.8GB) | 2000GB ※HDD | 100Gbps InfiniBand EDR | 1408 | ノード間高並列演算指向 |
| Hシステム | 136 | Intel Xeon D-1541 2.1GHz × 1基 | 8 | 64GB (8GB) | 512GB ※SSD | 10Gbps × 4 Ethernet | 205 | ノード間高並列演算志向 |
| Qシステム | 1 | AMD EPYC 7713P 2.0GHz × 1基 | 64 | 512GB (8GB) | 1600GB ※SSD | 10Gbps Ethernet | 2048 | メニーコアプロセッサ (AMD EPYC) 搭載 |
| Rシステム | 10 | AMD EPYC 7543P 2.8GHz × 1基 | 32 | 256GB (8GB) | 1600GB ※SSD | 10Gbps Ethernet | 1434 | メニーコアプロセッサ (AMD EPYC) 搭載 |
| Sシステム | 32 ※1 | AMD EPYC 9654 2.4GHz × 2基 | 192 | 768GB | 1600GB | 100Gbps Ethernet | 7373 | AMD EPYC搭載 |
| 8コアVM | 48(VM) ※2 | 上記CPUから 1VMあたり8コアを割り当て | 8 | 24GB (3GB) | 50GB ※SSD | 仮想ネットワーク インターフェース (NIC) | 1229 | 仮想計算機VM運用 ノード(VM)内並列・小規模 ノード間並列 |
| 32コアVM | 80(VM) ※2 | 上記CPUから 1VMあたり32コアを割り当て | 32 | 128GB (4GB) | 200GB ※SSD | 仮想ネットワーク インターフェース (NIC) | 1229 | 仮想計算機VM運用 ノード(VM)内並列・小規模 ノード間並列 |
| 192コア (物理) | 16 ※2 | 上記同様 | 192 | 768GB (4GB) | 1600GB | 100Gbps Ethernet | 7373 | AMD EPYC搭載 |

※1 Sシステムは全32ノードを、物理ノード(192コア)とVM環境(32コアVMと92コアVM)に分割して提供します。

※2 Sシステムの物理ノード提供数・VM提供数は利用状況に応じ適宜変更しています。最新情報は渉外担当者まで。



4.1 種類豊富な演算ノード群 (アクセラレータ)

| | ノード数 | ホストCPU、デバイス | コア数 | メモリ (コアあたり) | ローカルディスク | ノード間通信 | 理論演算性能 (GFLOPS) | 備考 |
|----------------|------|-------------------------------------|-----|-----------------|--------------|---------------------------|-----------------|----------------------|
| GPU | 2 | Intel Xeon E5-2698v4 2.2GHz × 2基 | 40 | 128GB (3.2GB) | 2000GB ※HDD | 56Gbps InfiniBand FDR | 1152 | 1ノードあたりGPGPU1基搭載 |
| | | NVIDIA Tesla P100/PCI × 1基 | — | 16GB | — | | 4700 | |
| MN-Core | 1 | Intel Xeon Platinum 8480+ × 2基 | 112 | 1024GB (9.1GB) | 45900GB ※SSD | 100Gbps Ethernet | 7168 | MN-Core2 8基搭載 |
| | | MN-Core2 × 8基 | — | 0 | — | | 96000 | |
| GPU | 2 | AMD EPYC 9354P 3.25GHz × 1基 | 32 | 384GB (12GB) | 3840GB ※SSD | 100Gbps Ethernet | 1664 | 1ノードあたりGPGPU1基搭載 |
| | | NVIDIA H100 NVL 94GiB × 1基 | — | 94GB | — | | 34000 | |
| GPU | 1 | AMD EPYC 7543 2.8GHz × 2基 | 64 | 512GB (8GB) | 1600GB ※SSD | 10Gbps Ethernet | 2867 | 1ノードあたりGPGPU2基搭載 |
| | | NVIDIA A100 × 2基 | — | 80GB×2 | — | | 19400 | |
| Vector | 2 | Intel Xeon Gold 6148 2.4GHz × 1基 | 20 | 96GB (4.8GB) | 240GB ※SSD | 56Gbps InfiniBand FDR | 1024 | 1ノードあたりベクトルエンジン1基搭載 |
| | | NEC SX-Aurora TSUBASA Type 10B × 1基 | 8 | 48GB (6GB) | — | | 2150 | |
| Vector | 2 | Intel Xeon Gold 6148 2.4GHz × 2基 | 40 | 192GB (4.8GB) | 240GB ※SSD | 100Gbps InfiniBand EDR | 2048 | ベクトルプロセッサ搭載 |
| | | NEC SX-Aurora TSUBASA Type 10B × 8基 | 64 | 48 (0.8GB) | — | | 17203 | |
| 三富岳 | 6 | Fujitsu A64FX 1.8GHz × 1基 | 48 | 32GB (0.7GB) | 512GB ※SSD | 100Gbps InfiniBand EDR | 2765 | 「富岳」商用機FX700 A64FX搭載 |
| PrePost | 1 | Intel Xeon Silver 4112 2.6GHz × 2基 | 8 | 1,536GB (192GB) | 1200GB ※HDD | 56Gbps InfiniBand FDR | 282 | プリポスト、大容量メモリ処理用 |
| | | NVIDIA Quadro P4000 × 2基 | — | — | — | | — | |



4.2 利用料金

従量利用制 利用時間・ノード数に応じて、利用分だけ支払い

| | 従量利用単価 | 並列割引適用 |
|---------------|---------------|-------------------------------------|
| Fシステム | 300円/ノード・時間 | 4ノード毎に30円減額(5~20ノード)、min150円/ノード・時間 |
| Hシステム | 60円/ノード・時間 | 4ノード毎に5円減額(5~20ノード)、min35円/ノード・時間 |
| Mシステム | 無償 | (設定無し) |
| Nシステム | 1,000円/ノード・時間 | (設定無し) |
| Qシステム | 450円/ノード・時間 | (設定無し) |
| Rシステム | 300円/ノード・時間 | 4ノード毎に30円減額(5~8ノード)、min240円/ノード・時間 |
| Sシステム(8コアVM) | 60円/ノード・時間 | 4ノード毎に5円減額(5~20ノード)、min35円/ノード・時間 |
| Sシステム(32コアVM) | 320円/ノード・時間 | 4ノード毎に32円減額(5~8ノード)、min160円/ノード・時間 |
| Sシステム(192コア) | 1,200円/ノード・時間 | 1ノード毎に60円減額(2~6ノード)、min840円/ノード・時間 |
| Vシステム | 250円/ノード・時間 | (設定無し) |
| Wシステム | 500円/ノード・時間 | (設定無し) |
| Xシステム | 100円/ノード・時間 | (設定無し) |
| Zシステム | 300円/ノード・時間 | 4ノード毎に30円減額(5~20ノード)、min150円/ノード・時間 |

期間占有利用(日単位/月単位/年度単位予約)

日単位: 利用単価×ノード数×24時間

月単位/年度単位: 1ノードあたり20~60%程度の割引率を設定



4.3 FOCUSスパコンでのソフトウェア利用

- **OS、開発環境、ジョブ管理システム等は無償**
(アカウント発行料を含む)
- **商用ソフトウェアは原則利用者がベンダーから直接ライセンスを取得しFOCUSスパコンにインストールして利用**
(FOCUSスパコン上での利用形態、料金、実行方法等はベンダーに直接お問い合わせください)
 - **Gaussian 09/16**、MIZUHO BioStation は
FOCUSがライセンスを取得しており、時間従量制にて利用可
(要・利用申請、OKBizよりご連絡ください)
- **オープンソース(OSS)等で要望が多いものは共用領域にインストール**
- **自社開発ソフトやOSSをユーザ自身がホーム領域にインストールすることも可能**



4.4 CCNVマルチサーバCAE実行システム

■ マウス操作でFOCUSスパコンを利用

- ファイル転送、ジョブ実行、ジョブ実行管理、リモートファイル編集、Linuxコマンド実行といったスパコン利用に必要な機能を全て実装

■ FOCUSスパコンユーザーは**無償**で利用可能

CUIでの実行

```
$ cd $workdir
$ cat run.sh
#!/bin/bash
#SBATCH -p g001h Lec
#SBATCH -N 2
#SBATCH -n 2
#SBATCH --ntasks-per-node=1
#SBATCH -c 12
#SBATCH -J pi_hyb
#SBATCH -o output_hyb%J.o
#SBATCH -e output_hyb%J.e

export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}

#mpirun -np $SLURM_NTASKS ./pi_hyb.out
mpirun -np $SLURM_NTASKS ./pi_hyb.exe
```



GUIで選択・設定してジョブ投入



Job

Submit Control License

Use template

Class: f024h_p100 【Fシステム(w/GPU)(最長 24時間)】XeonE5-...

Library: intel2021

Dependency:

JOB名: pi_hyb

ノード数: 2

プロセス数: 2

スレッド数: 12

実行時間制限:

Working directory: /home1/gism/uism0001/demo

Settings Script

Auto Interval time: 60



4.5 各種ソフトウェア利用講習会

・講習会情報

各種ソフトウェア利用講習会を財団主催/外部団体主催で実施しています。

【講習会情報】 https://www.j-focus.jp/lectures/currently_accepting/seminarandevent/

| スパコン利用に関する講習会 | | |
|------------------------------|----|----------------|
| スパコン利用講習会 | 無料 | 月1回（神戸、東京で各1回） |
| スパコン利用のためのLinux初級講習会 | 有料 | 月1回 スパコンユーザ無料 |
| 計算化学に関するアプリケーション講習会 | | |
| Gaussian & GaussView利用講習会 | 有料 | 年3～4回程度 |
| Quantum ESPRESSO利用講習会 | 有料 | 年2回 |
| LAMMPS利用講習会 基礎 | 有料 | 年4回 |
| LAMMPS利用講習会 応用（有機物・ポリマー） | 有料 | 年2回 |
| LAMMPS利用講習会 応用（界面・無機物） | 有料 | 年2回 |
| 可視化ソフトウェアに関する講習会 | | |
| ParaView利用講習会 | 無料 | 年2回 |
| ANSYS EnSight利用講習会 | 無料 | 年2回 |
| 「富岳」、HPCI活用に向けた講習会 | | |
| HPCIアクセスポイント解説セミナー（RIST共同開催） | 無料 | 年4回 |
| ハンズオンで学ぶ計算物理化学の基礎（3日間コース） | 無料 | 年2回 |
| 火災シミュレーションソフトウェアFDS利用講習会 | 無料 | 年2回 |
| 電磁界シミュレーター OpenFDTD 利用講習会 | 無料 | 年2回 |
| | | |

※スパコンによるハンズオンには、FOCUSスパコンを利用（一部アプリケーションでは「富岳」）

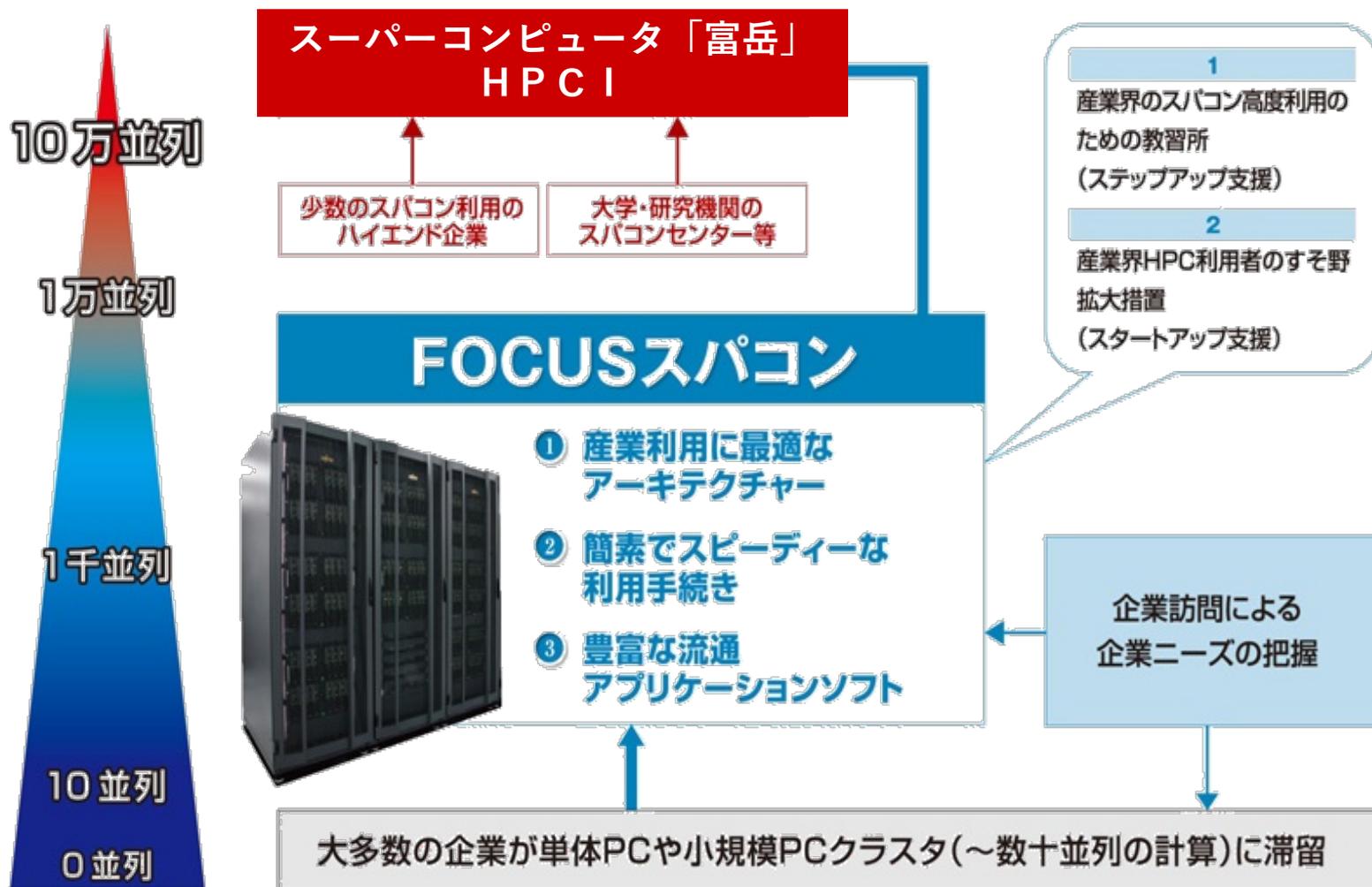
■講習会編成には追加・変更があります。また不定期開催の企画あり(AI 関連など)。

■ほか外部団体連携による講習会は随時開催します(アプリケーション関係など)。



5. 富岳、HPCI計算資源について

「富岳」/HPCIへのステップアップとスパコン利用のすそ野拡大に寄与



HPCIシステムとは？

「富岳」を中核として国内の高性能なコンピュータを、学術的な研究から産業界の研究開発まで、広範な用途に利用できるように整備している計算基盤システム
(富岳、北大、東北大、筑波大、東大、東工大、名大、京大、阪大、九大、産総研、JAMSTC)

- ・ 募集中の課題に申請し、採択されると利用可能
- ・ 応募資格
 - ・ 学術界：日本国内外の研究機関及び大学に所属する方
 - ・ 産業界：原則として、日本国内に登録された企業に所属する方



詳しくは、下記URLから各課題の募集要項を参照

https://www.hpci-office.jp/pages/project_categories?tab=industrial

ヘルプデスク

メール：helpdesk@hpci-office.jp

電話：078-940-5795

Webフォーム

<https://www.hpci-office.jp/pages/helpdesk/>

Webフォーム、メールによる受付：24時間

電話による受付：平日9:00～12:00/13:00～17:30





5.2 富岳(HPCI)、FOCUSスパコンの特徴、利用用途

| | 特長 | 対象 |
|---------------|---|--------------------|
| 富岳 (HPCI) | <ul style="list-style-type: none">・ 大規模計算機資源を無償/安価に利用できる・ 富岳をはじめ、最新の計算機資源 (GPU、IntelXeon、AMD、SX等) 利用可能・ さまざまな募集課題 ・ 利用には審査を経て採択される必要あり・ 利用報告書 | 研究課題向け 大規模計算が必要 |
| FOCUS スパコン | <ul style="list-style-type: none">・ 利用報告義務なし・ 有償アプリの利用が容易 ・ 計算資源的に、3000並列程度の利用が目安 | スパコン入門 小-中規模計算 |

富岳産業利用申請マニュアル

<https://www.r-ccs.riken.jp/outreach/topics/20211203-1/>



5.3 FOCUSの富岳活用支援サービス（検証準備中）

サービスの特長

対象:有償(産業)利用・従量制

ポイント1

FOCUSが利用申請や報告等の事務処理を代行するため、ユーザーはシミュレーション業務に集中できます

ポイント2

ユーザー直接の課題申請より安価に 富岳を利用できます

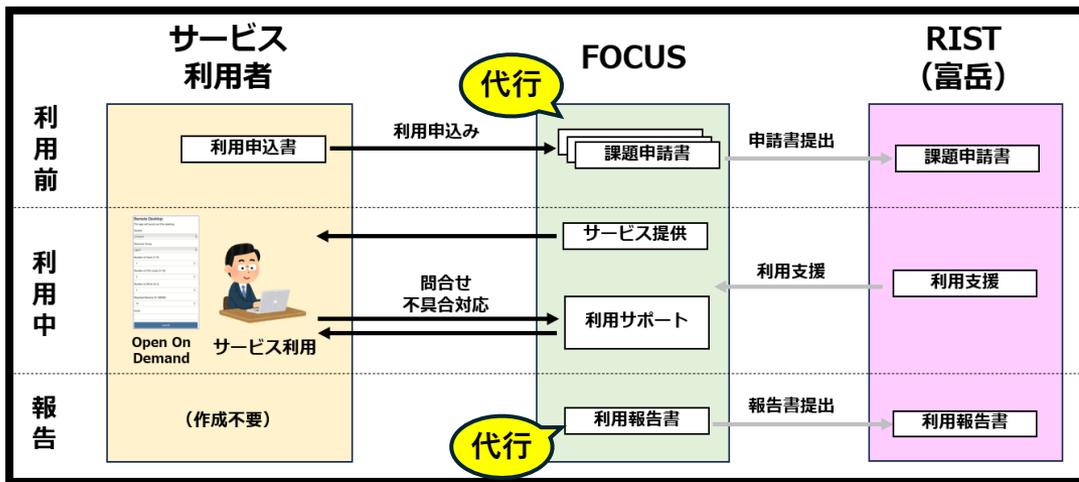
(ご参考)
 [直接利用] ¥98/NH ⇒
 [FOCUS経由] 検討中

ポイント3

ジョブの優先実行

直接利用ジョブより優先的に処理されます

サービス利用の流れ



お問合せ先

※富岳活用支援サービスに関するFOCUS窓口
 E-mail : fugaku-cloud@j-focus.or.jp

※FOCUSスパコンサービス全般については
 E-mail : fukyu@j-focus.or.jp 普及促進グループ



6.1 ログイン経路

6.2 VPNによる接続

6.3 WinSCPによるファイル転送



6.1 ログイン経路

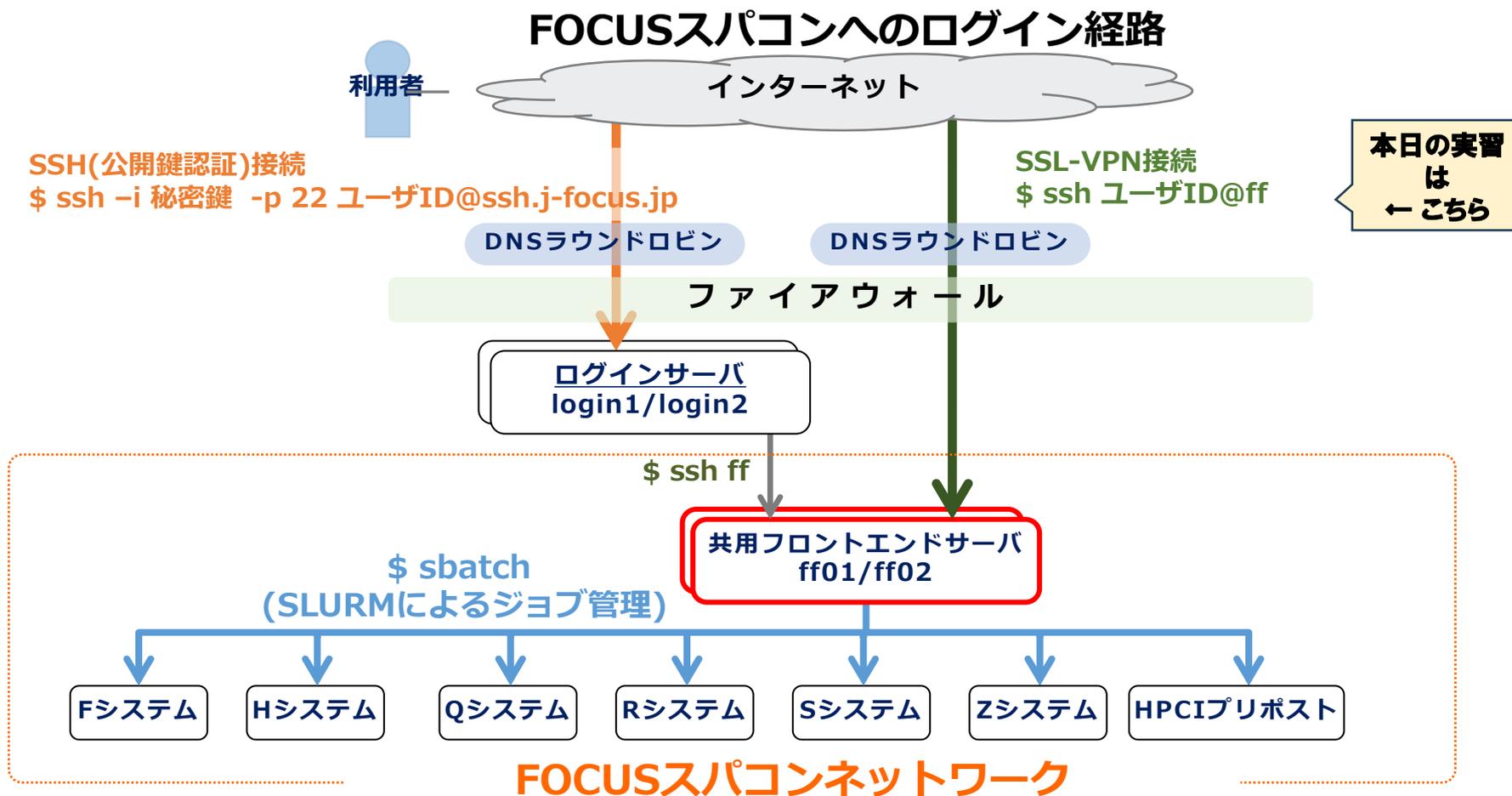
FOCUSスパコンにログインするには、以下2つの経路があります。

2-1 SSH(公開鍵認証)による接続

(ローカルPC→ログインサーバ→フロントエンド)

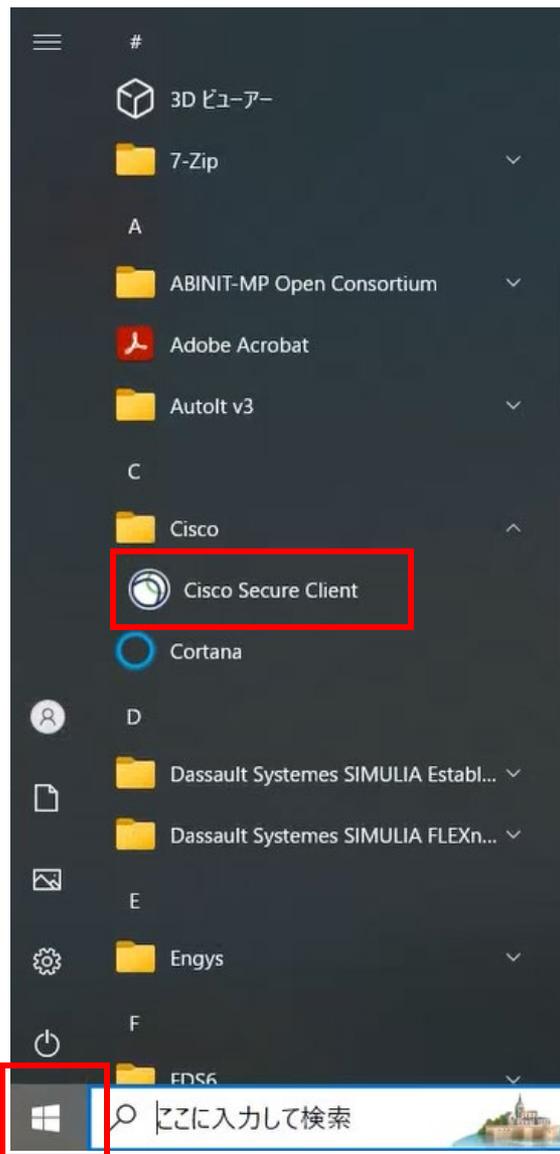
2-2 SSL-VPNによる接続

(ローカルPC→フロントエンド)





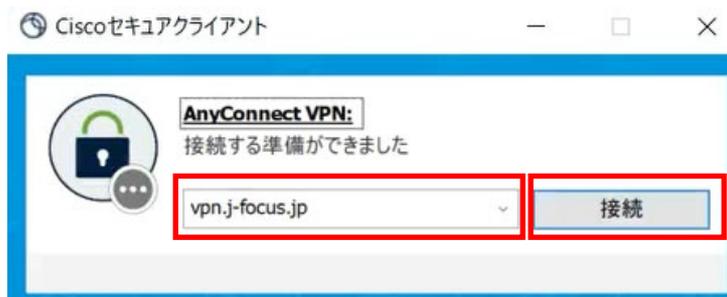
6.2 SSL-VPNによる接続【実習】 (1)



1) Cisco Secure Client Any Connect VPNの起動

2) SSL-VPNサーバへの接続

接続先が「vpn.j-focus.jp」になっていることを確認し、「接続」をクリック。



接続先 : vpn.j-focus.jp



3) 接続設定の入力

Ciscoセキュアクライアント | vpn.j-focus.jp ×

 ユーザー名とパスワードを入力してください。

グループ:

ユーザー名:

パスワード:

Group : 「Focus」であることを確認
Username : (お手元のアカウント情報)
Password : (お手元のアカウント情報)



6.2 SSL-VPNによる接続【実習】(3)



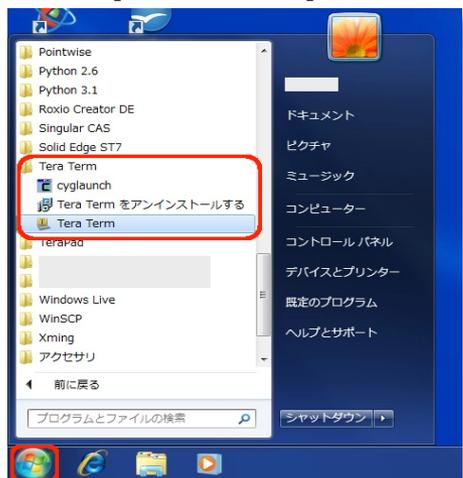
接続の確認



6.2 SSL-VPNによる接続【実習】 (4)

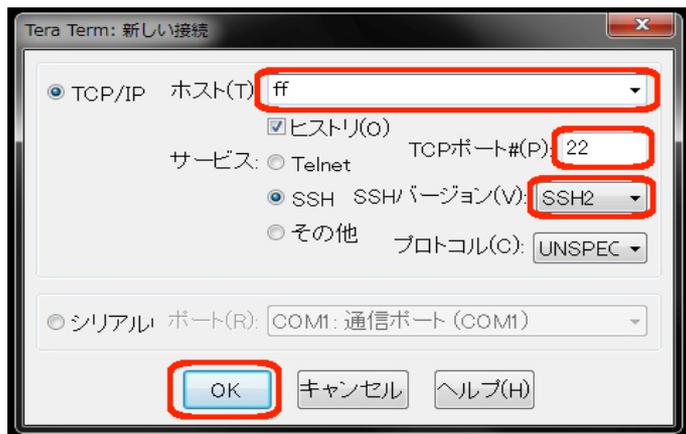
5) ログイン (TeraTermを使つてのログイン)

i) 端末機(Windows)にて tera term を起動



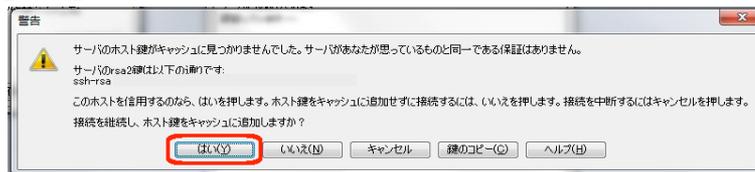
[スタートメニュー]
→[すべてのプログラム]
→[Tera Term]
→[Tera Term]

ii) 接続設定を入力



ホスト名 : ff (共用フロントエンド)
TCPポート# : 22
SSH バージョン : SSH2

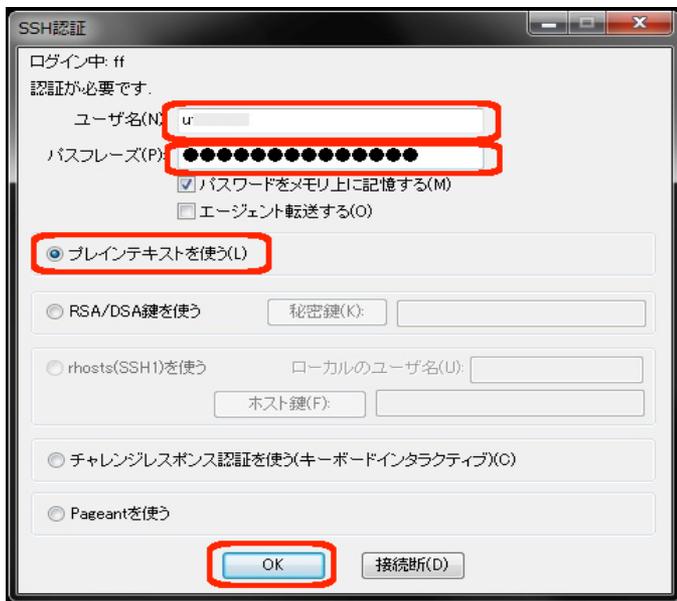
初回ログイン時に以下の表示がでます。
[OK]をクリックしてください。





6.2 SSL-VPNによる接続【実習】 (5)

iii) アカウント情報を入力



ユーザ名 : (お手元のアカウント情報)
パスワード : (お手元のアカウント情報)
プレーンテキストを使う : ON

iv) ログイン完了



6) ログアウト手順 (exitコマンド)

```
$ exit
```



6.3 WinSCP (SSL-VPN接続) によるファイル転送【実習】

ローカルPC → フロントエンドサーバ へ転送します。

※ /home1/gxxx はログインサーバおよびフロントエンドサーバの両方にマウントされています。

【参考】・ FOCUSスパコン利用の手引

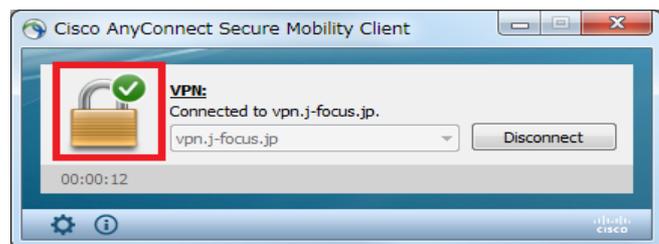
2.1.2.2.SSL-VPN接続によるSCPファイル転送 (WinSCP使用)

https://www.j-focus.jp/user_guide/ug0002010000/#ug0002010202

・ SSL-VPN利用の手引

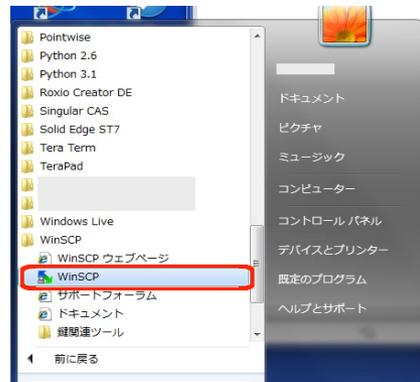
<https://www.j-focus.jp/sslvpn/>

1) SSL-VPN接続を確立



接続状態 (グリーンの「チェックマーク」) であることを確認

2) WinSCPを起動

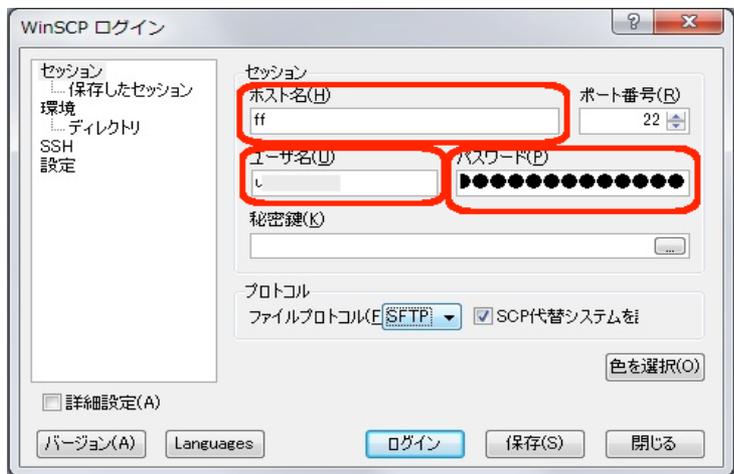


[スタートメニュー]
→ [すべてのプログラム]
→ [WinSCP]
→ [WinSCP]



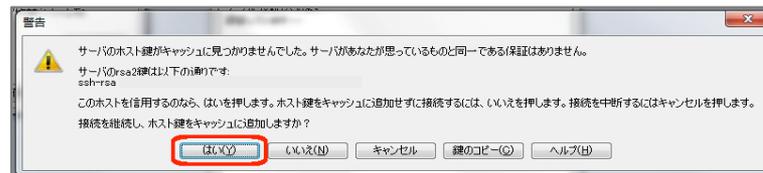
6.3 WinSCP (SSL-VPN接続) によるファイル転送 **【実習】**

3) ログイン情報を入力し、接続

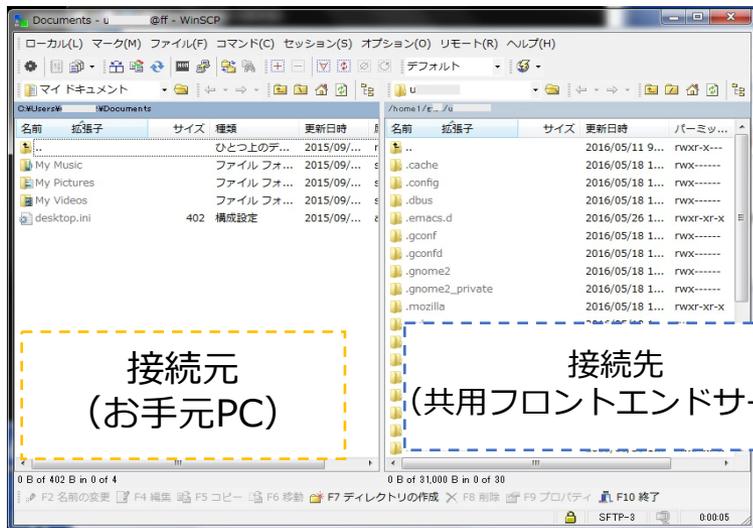


ホスト名 : ff (共用フロントエンド)
ユーザ名 : (お手元のアカウント情報)
パスワード : (お手元のアカウント情報)

初回ログイン時に以下の表示がでます。
[OK]をクリックしてください。



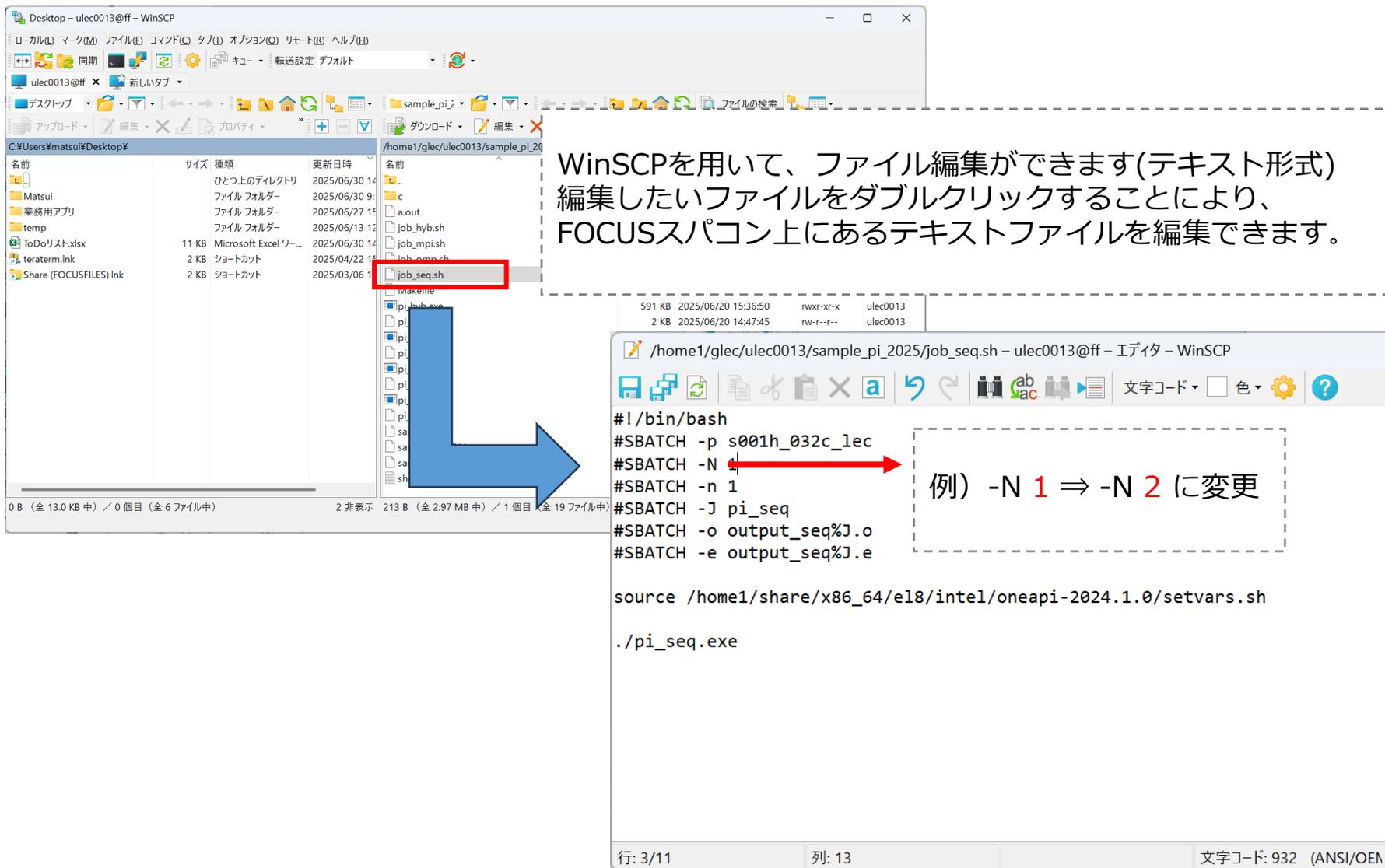
4) ファイル転送が可能になります



転送したいドキュメントを選択し、
転送先にドラッグ&ドロップすることにより転送
できます。



(参考) ファイル編集 (WinSCP編)



WinSCPを用いて、ファイル編集ができます(テキスト形式)
編集したいファイルをダブルクリックすることにより、
FOCUSスパコン上にあるテキストファイルを編集できます。

例) `-N 1` ⇒ `-N 2` に変更

```
#!/bin/bash
#SBATCH -p s001h_032c_lec
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -J pi_seq
#SBATCH -o output_seq%J.o
#SBATCH -e output_seq%J.e

source /home1/share/x86_64/e18/intel/oneapi-2024.1.0/setvars.sh

./pi_seq.exe
```



(参考) SSHターミナルソフトウェア



Teraterm
定番



Tabby
かっこいい



PowerShell
Windows標準搭載



rlogin
ファイル転送ツールあり



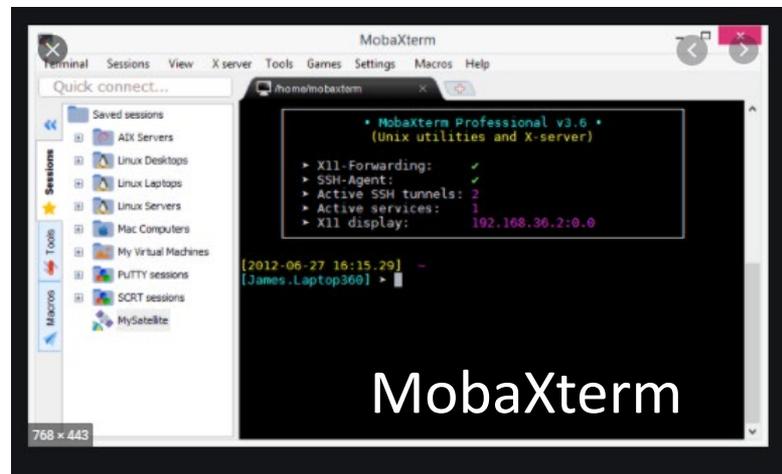
MobaXterm

MobaXterm
高機能



PuTTY

Putty
シンプル、人気



7.1 Linuxのコマンド実行

7.2 計算実行

7.3 ジョブのキャンセル

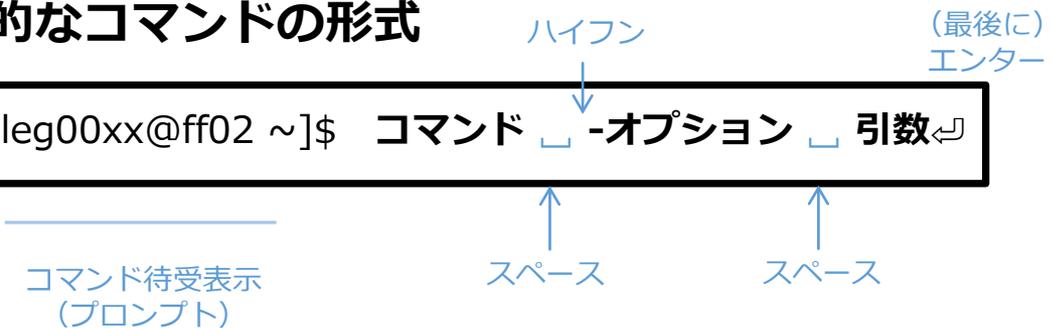


(参考) 知っておきたい Linuxコマンド

| コマンド | 主なオプション | 引数 | 用途 | コマンド名の由来 |
|----------------|---------|------------------------------|--|--|
| ls | a, l | ファイル/ディレクトリ名 | ファイル一覧を表示する | <u>l</u> ist |
| pwd | | | カレントディレクトリを絶対パスで表示 | <u>print</u> <u>working</u> <u>directory</u> |
| cd | | 移動先のパス | カレントディレクトリを移動する | <u>change</u> <u>directory</u> |
| mkdir rmdir | | ディレクトリ名 ディレクトリ名 | ディレクトリの新規作成 空ディレクトリの削除 | <u>make</u> <u>directory</u> <u>remove</u> <u>directory</u> |
| cp | i, r | コピー元 コピー先 | ファイルやディレクトリをコピーする | <u>copy</u> |
| mv | i | 移動元 移動先 | ファイルやディレクトリの移動または名前変更 | <u>move</u> |
| rm | i, r | ファイル名 | ファイルの削除 | <u>remove</u> |
| cat | | ファイル名 | ファイル内容の標準出力する | <u>catenate</u> (連結する) |
| head tail | n n | ファイル名 ファイル名 | ファイルの最初の数行を標準出力する ファイルの最後の数行を標準出力する | |
| less more | | ファイル名 ファイル名 | ファイルを表示する(上下スクロール可) ファイルを表示する(画面送りのみ) | |
| grep | i, v | 検索ワード ファイル名 | ファイル内の文字列を検索、抽出 | <u>globally</u> <u>search</u> a <u>regular</u> <u>expression</u> and <u>print</u> |
| chmod chown | R R | ディレクトリ/ファイル名 ディレクトリ/ファイル名 | ファイルのアクセス権限の変更 ファイル所有者の変更 | <u>change</u> <u>mode</u> <u>change</u> <u>owner</u> |
| printenv | | | 環境変数の表示 | <u>print</u> <u>environment</u> |
| echo | | 表示内容 | 文字列や環境変数の表示 | |
| man | | コマンド | コマンドのマニュアル表示 | <u>manual</u> |

一般的なコマンドの形式

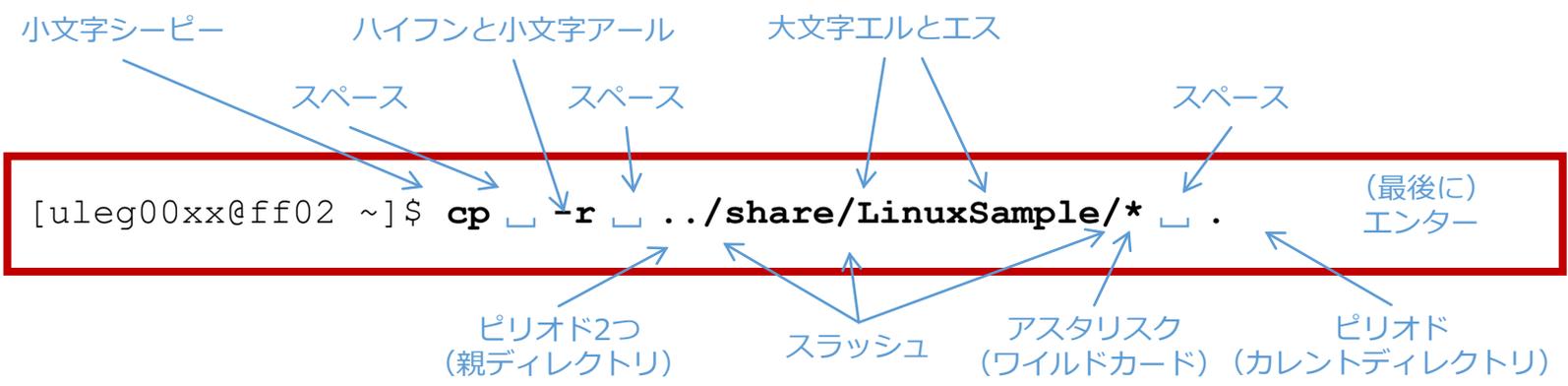
```
[uleg00xx@ff02 ~]$ コマンド -オプション 引数 ↵
```



- 小文字/大文字 は区別されます
- 区切りのスペースはひとつ以上入力（連続で2つ入力されてもOK）
- オプションは1つとは限りません（なし、2つ以上など・・・）
- 引数は1つとは限りません（なし、2つ以上など・・・）
- 本講義では登場しませんがサブコマンドを持つコマンドもあります

●まずは、サンプルファイルの準備のため、以下のコマンドを入力してみましょう。

```
[uleg00xx@ff02 ~]$ cp -r ../share/LinuxSample/* .
```





ファイルやディレクトリの一覧表示

```
[uleg00xx@ff02 ~]$ ls  
memo.txt  memo1.txt  sample  testdir
```

```
ulec0008@ff02:~  
[ulec0008@ff02 ~]$ ls  
memo.txt  memo1.txt  sample  testdir  
[ulec0008@ff02 ~]$
```

色付きがディレクトリ
白色はファイル
(ソフトウェアによって異なる)

Linuxは視覚的にディレクトリ階層、ディレクトリ内部ファイルが分からない
常にlsで確認しましょう。



ディレクトリ確認, 移動

```
[uleg00xx@ff02 ~]$ pwd
```

```
/home1/glec/ulec00xx
```

カレントディレクトリを表示

```
[uleg00xx@ff02 ~]$ cd testdir
```

```
[uleg00xx@ff02 ~]$ pwd
```

```
/home1/glec/ulec00xx/testdir
```

testdirディレクトリに移動

```
[uleg00xx@ff02 ~]$ cd ..
```

```
[uleg00xx@ff02 ~]$ ls
```

```
memo.txt memo1.txt sample testdir
```

一つ前のディレクトリに移動

- 引数なしの[cd]はホームディレクトリに移動
- [..]は1つ前を表す
- ls、pwdで常にどこにいるか、何があるか把握する（推奨）



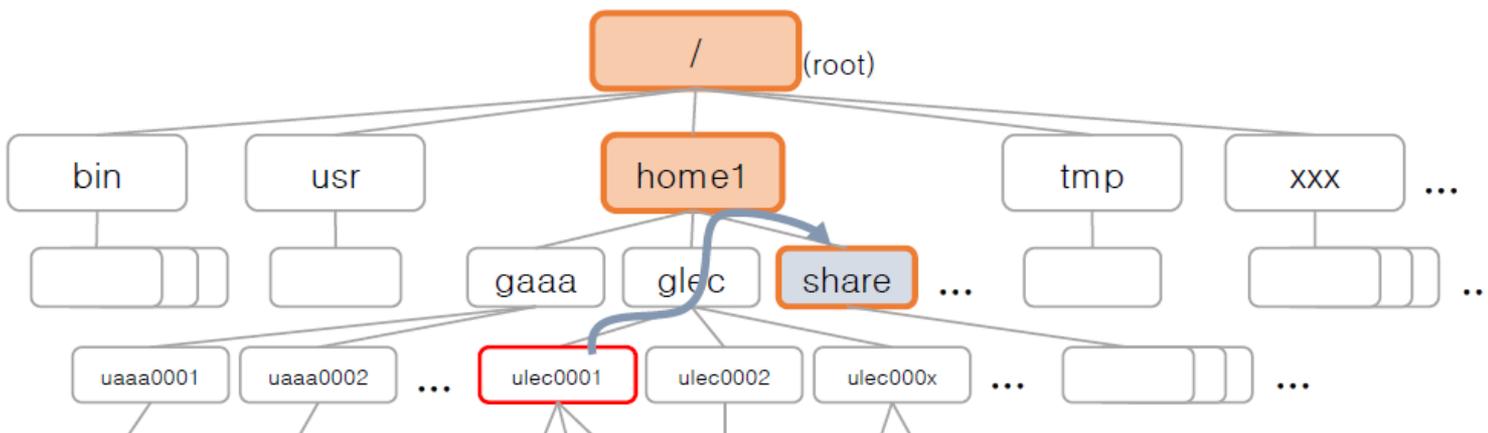
(参考) Linux ディレクトリ構成 (絶対パス/相対パス)

Windowsの場合 ⇒ C:¥Users¥glec¥...

Linuxの場合

例) 絶対パス : /home1/share

相対パス : ../../share (/home1/glec/ulec0001 からの相対パス)





ファイル表示

```
[uleg00xx@ff02 ~]$ cat memo.txt
```

memo.txtの中身を確認

```
[uleg00xx@ff02 ~]$ cd testdir
```

testdirディレクトリに移動

```
[uleg00xx@ff02 ~]$ pwd
```

```
/home1/glec/ulec00xx/testdir
```

一つ前のディレクトリに移動

```
[uleg00xx@ff02 ~]$ cd ..
```

```
[uleg00xx@ff02 ~]$ ls
```

```
memo.txt  memo1.txt  sample  testdir
```

- 引数なしの[cd]はホームディレクトリに移動
- [..]は1つ前を表す
- ls、pwdで常にどこにいるか、何があるか把握する（推奨）

- LAMMPSのサンプルデータfrictionを実行。

```
# 2d friction simulation

dimension          2
boundary           p s p

atom_style         atomic
neighbor           0.3 bin
neigh_modify       delay 5

# create geometry

lattice            hex 0.9
region             box block 0 50 0 22 -0.25 0.25
create_box         4 box

<略>
# Run

timestep           0.0025
thermo             1000
thermo_modify      temp new

#dump              1 all atom 500 dump.friction

#dump              2 all image 500 image.*.jpg type type &
#                  zoom 1.6 adiam 1.5
#dump_modify       2 pad 5

#dump              3 all movie 500 movie.mpg type type &
#                  zoom 1.6 adiam 1.5
#dump_modify       3 pad 5

run                20000
```



7.2.1 サンプルデータのコピー 【実習】

```
$ cd
$ cp -r ../share/sample_LAMMPS2025 .
$ cd sample_LAMMPS2025
$ ls
in.friction  job02.sh
```

サンプルデータをコピー

「sample」ディレクトリへ移動

ディレクトリ内の一覧を表示

ジョブ実行に必要なこと

- ①プログラムの準備
- ②入力データのアップロード
- ③バッチスクリプト作成

バッチスクリプト例 (job02.sh)

```
$ cat job02.sh
```

```
#!/bin/bash
#SBATCH -p s001h_032c_lec      # キュー名指定
#SBATCH -n 24                 # プロセス数指定
#SBATCH -J lmp_fric           # ジョブ名指定
#SBATCH -o lmp_fric.o%J      # SLURM標準出力ファイル名
#SBATCH -e lmp_fric.e%J      # SLURM標準エラー出力ファイル名
# %JにはジョブIDが補完されます

source /home1/share/x86_64/el8/intel/oneapi-2024.1.0/setvars.sh #intel環境設定読込

#LAMMPS実行コマンドの設定
#export EXE=/home1/share/LAMMPS/lammps-16Mar18/src/lmp_intel_cpu_intelmpi
#export EXE=/home1/share/x86_64/el8/LAMMPS/lammps-2Aug2023/src/lmp_oneapi
#export EXE=/home1/share/LAMMPS/lammps-16Feb16/src/lmp_mpi
#export EXE=/home1/share/LAMMPS/lammps-28Jun14/src/lmp_linux

mpirun -np ${SLURM_NTASKS} $EXE < in.friction #LAMMPS実行
```



7.2.3 空き状況を確認 (squeues)

【実習】

・ キュー毎の利用状況確認 (squeues コマンド) ※一般アカウントでの表示例です。

```

$ sqqueues
QUEUE_NAME      TIMELIMIT      STATUS  MAXNODES  NNODES  DEPEND  PEND  RUN  FREE
a006m           6:00           up      2          0        0       0    0    2
a024h           1-00:00:00    up     194         1        0       0    1   193
a168h           7-00:00:00    up      31          1        0       0    1    30
b006m           6:00           up      2          0        0       0    0    2
b024h           1-00:00:00    up      2          0        0       0    0    2
b168h           7-00:00:00    up      1          0        0       0    0    1
f006m           6:00           up      2          0        0       0    0    2
f024h           1-00:00:00    up     53         12        4       0    8   45
f072h           3-00:00:00    up     31          8        0       0    8   23
f072h_p100     3-00:00:00    up      2          0        0       0    0    2
g006m           6:00           up      4          0        0       0    0    4
h006m           6:00           up      4          0        0       0    0    4
h024h           1-00:00:00    up    136         33        0       0   33  103
h072h           3-00:00:00    up    102         33        0       0   33   69
p072h           3-00:00:00
v006m           6:00
v024h           1-00:00:00
v072h           3-00:00:00

```

<出力説明>

| | |
|------------|---------------------------------|
| QUEUE_NAME | キュー名 (パーティション名) |
| TIMELIMIT | 最大実行時間 |
| STATUS | キューの状態 (up or inact) |
| MAXNODES | 最大ノード数 |
| NNODES | 実行中及び実行待ちのジョブが要求しているノード数 |
| DEPEND | 実行待ち(Dependency)のジョブが要求しているノード数 |
| PEND | 実行待ちのジョブが要求しているノード数 |
| RUN | 実行中のノード数 |
| FREE | 空きノード数 |

FREEが要求ノード数より多く、
PEND がゼロであるキューを指定すると
ジョブを即時実行できます。



7.2.4 ジョブの実行

\$ queues

`-p` で指定したキュー (g001h_lec) の空き状況を確認

| QUEUE_NAME | TIMELIMIT | STATUS | MAXNODES | SNNODES | DEPEND | PEND | RUN | FREE |
|----------------|-----------|--------|----------|---------|--------|------|-----|------|
| a001h_lec | 1:00:00 | up | 91 | 5 | 0 | 0 | 5 | 86 |
| f001h_lec | 1:00:00 | up | 16 | 15 | 0 | 0 | 15 | 1 |
| f001h_p100_lec | 1:00:00 | up | 2 | 0 | 0 | 0 | 0 | 2 |
| z001h_lec | 1:00:00 | up | 4 | 0 | 0 | 0 | 0 | 4 |
| z006m | 6:00 | up | 4 | 0 | 0 | 0 | 0 | 4 |
| h001h_lec | 1:00:00 | up | 68 | 36 | 0 | 0 | 36 | 32 |

FREE (空きノード) : 86

PEND (待ち状態) : 0

即時実行可能

\$ sbatch job2.sh

実行スクリプトを投入

Submitted batch job XXXXXX

\$ queue

ジョブの実行状況を確認

| JOBID | PARTITION | NAME | USER | ST | TIME | NODES | NODELIST(REASON) |
|-------|-----------|------|------|----|------|-------|------------------|
|-------|-----------|------|------|----|------|-------|------------------|

<主な ST(ステータス)フィールド 出力説明>

PD (PENDING) 資源の割り当て待ち

R (RUNNING) 実行中

CG (COMPLETING) 終了手続き中

ファイルが出力されているか確認

\$ ls

```

dump.friction.lamppstrj  job02.sh          lmp_fric.o2221080
in.friction              lmp_fric.e2221080 log.lammps

```



7.2.5 WinSCPで計算結果を転送

「sample_LAMMPS2025」フォルダをクリック

The screenshot shows the WinSCP interface with two panes. The left pane shows the local desktop (C:\Users\sakakura\Desktop) and the right pane shows the remote directory (/home1/gfsc/afsc0049/aaa/). The folder 'sample_LAMMPS2021' is highlighted with a red box in the right pane.

| 名前 | サイズ | 種類 | 更新日時 |
|----------------------------|-----------|---------------------|---------------------|
| .. | | ひとつ上のディレクトリ | 2023/06/23 16:55:30 |
| tmp | | ファイル フォルダ | 2023/06/23 10:13:15 |
| ★HPCI受託 (アプリ&講習会).lnk | 3 KB | ショートカット | 2022/03/30 13:25:02 |
| dump.friction.lammprj | 1,914 KB | LAMMPSSTRJ ファイル | 2023/06/22 11:17:16 |
| FOCUSスパコン紹介_20230621.pptx | 1,774 KB | Microsoft PowerP... | 2023/06/21 11:39:59 |
| FOCUSスパコン紹介_20230622.pptx | 1,771 KB | Microsoft PowerP... | 2023/06/22 11:58:45 |
| gvim.exe.lnk | 2 KB | ショートカット | 2020/03/03 10:20:35 |
| risa.url | 1 KB | インターネット ショート... | 2022/10/13 11:47:44 |
| RLogin.lnk | 2 KB | ショートカット | 2022/04/05 9:19:39 |
| rview-pc-agent-windows.exe | 66,633 KB | アプリケーション | 2023/04/24 12:45:52 |
| Share (FOCUSFILES).lnk | 2 KB | ショートカット | 2020/03/03 9:50:44 |
| Tabby Terminal.lnk | 3 KB | ショートカット | 2023/04/05 16:51:48 |
| teraterm-5.0-beta1.exe | 8,051 KB | アプリケーション | 2023/05/09 10:46:53 |
| Ubuntu 16.04 LTS.lnk | 2 KB | ショートカット | 2022/04/18 12:02:39 |
| 物理化学_スパコン体験_rev01.pptx | 24,054 KB | Microsoft PowerP... | 2023/06/22 18:06:52 |

| 名前 | サイズ | 更新日時 | パーミッション |
|-------------------|-----|---------------------|-----------|
| .. | | 2023/06/23 16:58:52 | rwx----- |
| sample_LAMMPS2021 | | 2023/06/23 16:45:22 | rwxr-xr-x |



7.2.5 WinSCFで計算結果を転送(2)

計算結果「dump.friction.lammpstrj」を
ドラッグ&ドロップでデスクトップ上（任意フォルダ）に転送

The screenshot shows the WinSCP interface with two panes. The left pane shows the local desktop (C:\Users\sakakura\Desktop) and the right pane shows the remote directory (/home/1/gfsc/afsc0049/aaa/sample_LAMMPS2021/). The file 'dump.friction.lammpstrj' is highlighted in the remote view, and a red arrow points from it to the desktop view.

| 名前 | サイズ | 種類 | 更新日時 |
|-----------------------------|-----------|---------------------|---------------------|
| .. | | ひとつ上のディレクトリ | 2023/06/23 16:55:30 |
| tmp | | ファイル フォルダ | 2023/06/23 10:13:15 |
| ★HPCI受託 (アプリ&講習会) .lnk | 3 KB | ショートカット | 2022/03/30 13:25:02 |
| dump.friction.lammpstrj | 1,914 KB | LAMMPSTRU ファイル | 2023/06/22 11:17:16 |
| FOCUSスポン紹介_20230621.pptx | 1,774 KB | Microsoft PowerP... | 2023/06/21 11:39:59 |
| FOCUSスポン紹介_20230622.pptx | 1,771 KB | Microsoft PowerP... | 2023/06/22 11:58:45 |
| gvim.exe.lnk | 2 KB | ショートカット | 2020/03/03 10:20:35 |
| risa.url | 1 KB | インターネット ショート... | 2022/10/13 11:47:44 |
| RLogin.lnk | 2 KB | ショートカット | 2022/04/05 9:19:39 |
| rvview-pc-agent-windows.exe | 66,633 KB | アプリケーション | 2023/04/24 12:45:52 |
| Share (FOCUSFILES).lnk | 2 KB | ショートカット | 2020/03/03 9:50:44 |
| Tabby Terminal.lnk | 3 KB | ショートカット | 2023/04/05 16:51:48 |
| teraterm-5.0-beta1.exe | 8,051 KB | アプリケーション | 2023/05/09 10:46:53 |
| Ubuntu 16.04 LTS.lnk | 2 KB | ショートカット | 2022/04/18 12:02:39 |
| 物理化学_スポン体験_rev01.pptx | 24,054 KB | Microsoft PowerP... | 2023/06/22 18:06:52 |

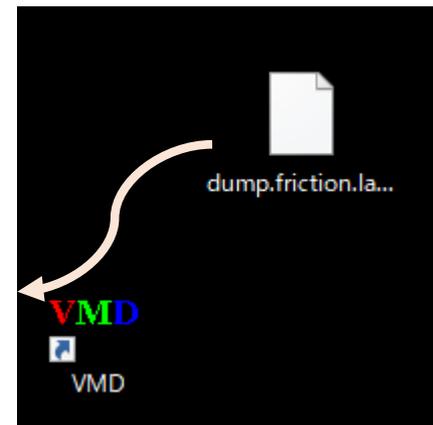
| 名前 | サイズ | 更新日時 | パーミッション |
|-------------------------|----------|---------------------|------------|
| .. | | 2023/06/23 16:59:27 | rxwxr-xr-x |
| dump.friction.lammpstrj | 1,914 KB | 2023/06/23 16:30:30 | rw-r--r-- |
| in.friction | 2 KB | 2023/06/22 10:35:32 | rw-r--r-- |
| job02.sh | 1 KB | 2023/06/22 10:35:32 | rw-r--r-- |
| Imp_fric.e1802652 | 0 KB | 2023/06/23 16:30:26 | rw-r--r-- |
| Imp_fric.o1802652 | 5 KB | 2023/06/23 16:30:31 | rw-r--r-- |
| log.lammps | 7 KB | 2023/06/23 16:30:30 | rw-r--r-- |



7.2.6 VMDで可視化

計算結果dump.friction.lammpstrjをVMDで開く。

VMD (Virtual Molecular Dynamics)
フリーの分子構造可視化ツール



Info) -----
 Info) Multithreading available, 12 CPUs, ISA dispat
 Info) CPU features: SSE2 SSE4.1 AVX AVX2 FMA F16
 Info) Detected VM or hypervisor execution environ
 Info) Free system memory: 1090MB (14%)
 Info) No CUDA accelerator devices available.
 Info) OpenGL renderer: Intel(R) UHD Graphics
 Info) Features: STENCIL MDE CVA MTX NPOT PP PS GL
 Info) Full GLSL rendering mode is available.
 Info) Textures: 2-D (16384x16384), 3-D (512x512x51
 Info) No joysticks found. Joystick interface disab
 Info) Dynamically loaded 76 plugins in directory:
 Info) C:/Program Files/VMD/plugins/WIN64/molfile
 Info) File loading in progress, please wait.
 Info) Using plugin lammpstrj for structure file C:\
 lammpsplugin) New style dump with 5 data fields. Co
 lammpsplugin) Reconstructing atomic coordinates fro
 Info) Using plugin lammpstrj for coordinates from fi
 Info) Determining bond structure from distance sear
 Info) Finished with coordinate file C:\Users\matsui
 Info) Analyzing structure ...
 Info) Atoms: 1722
 Info) Bonds: 4886

| ID | T | A | D | F | Molecule | Atoms | Frames | Vol |
|----|---|---|---|---|-------------------------|-------|--------|-----|
| 0 | T | A | D | F | C:\Users\matsui\Desktop | 1722 | 41 | 0 |



1. 逐次実行

- ・ 並列化せず、1プロセスで実行する
- ・ (FOCUSスパコンの場合、ノード単位での課金となるので、逐次実行は非効率的)

2. スレッド(**OpenMP**)並列【ノード内並列/共有メモリ】

- ・ スレッドを複数個生成し、それぞれのスレッドに別々のCPUコアを割り当てて並列処理を行う。
- ・ **各スレッド間ではメモリ空間を共有**して処理を行う。
- ・ 単一プロセス内で並列処理するため、複数のノードにまたがって並列処理することはできない。(ノード間並列不可)
- ・ 並列API: OpenMP 等

3. プロセス(**MPI**)並列【ノード間並列/分散メモリ】

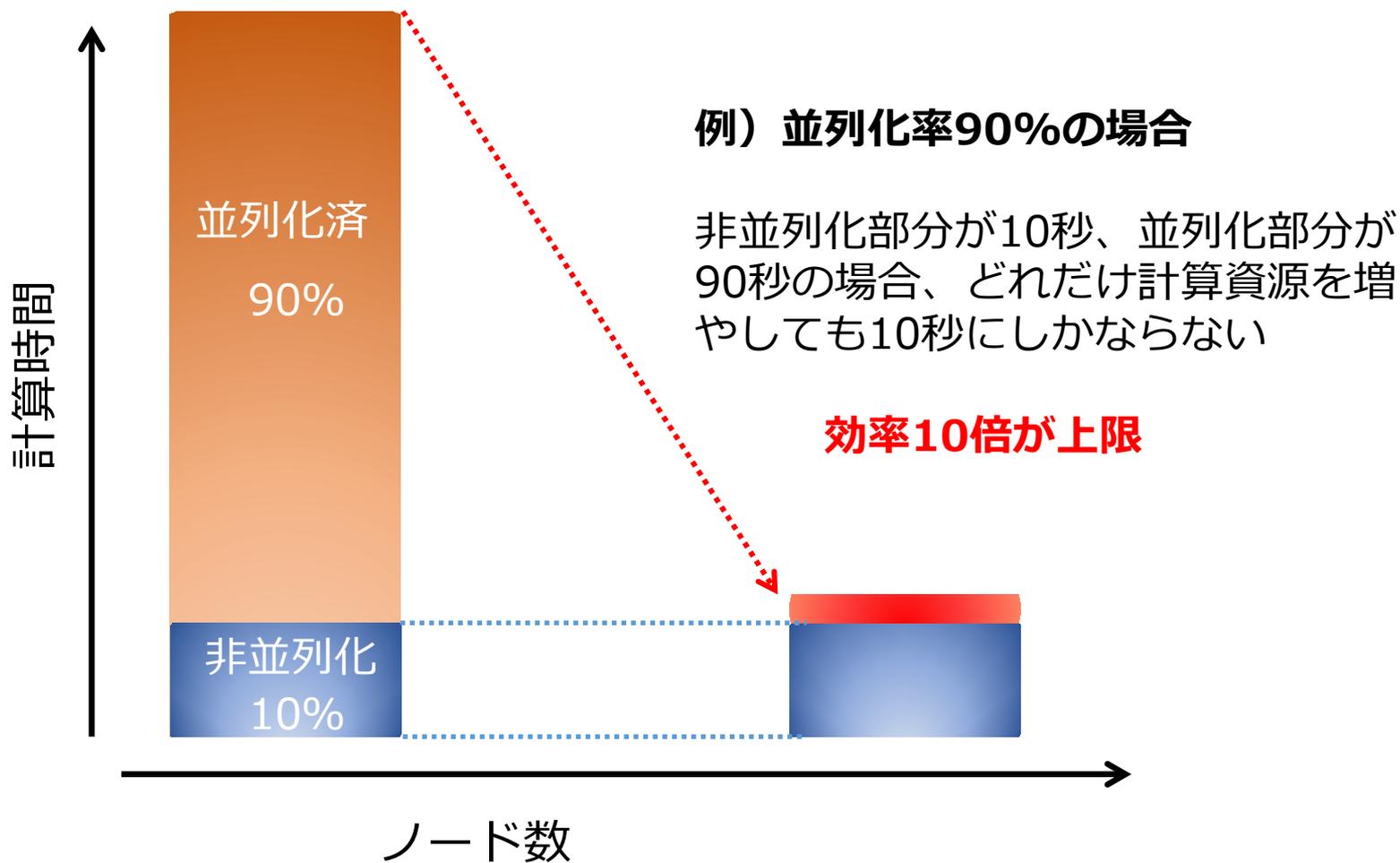
- ・ 複数のプロセスを起動して、それぞれのプロセスに別々のCPUコアを割り当てて並列処理を行う。
- ・ **各プロセスは、独立したメモリ空間を使用**して処理を行う。
- ・ 複数のノードにまたがって並列処理を行うことができる。(ノード間並列可能)
- ・ 並列API: MPI 等

4. ハイブリッド並列【ノード内/ノード間並列】

- ・ スレッド並列とプロセス並列を組み合わせることで並列処理を行う。
- ・ プロセス並列で並列処理を行い、個々のプロセス内でさらにスレッド並列による並列処理を行う。
- ・ 並列API: MPI/OpenMP 等

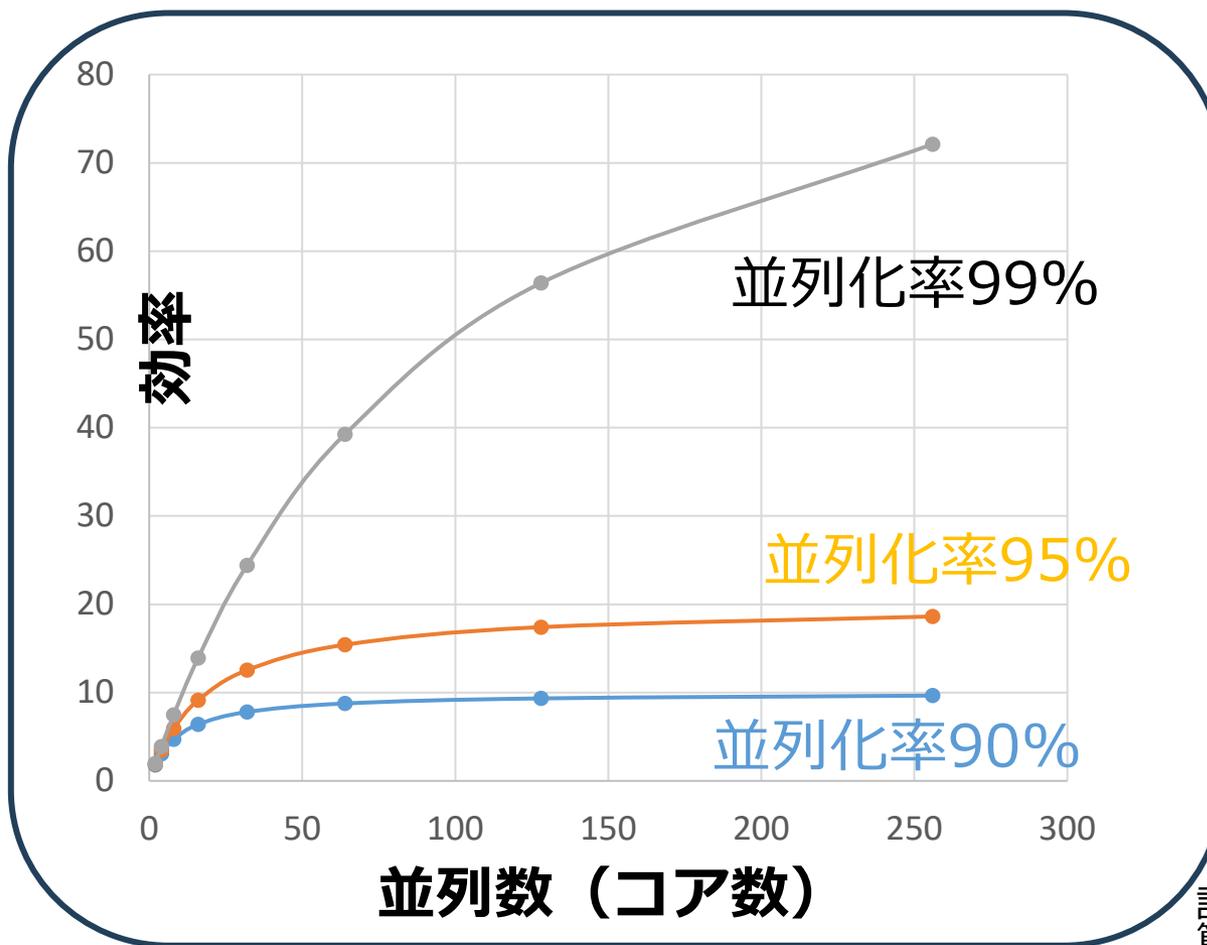


(参考) 並列ジョブの特性 並列化率と並列化効率

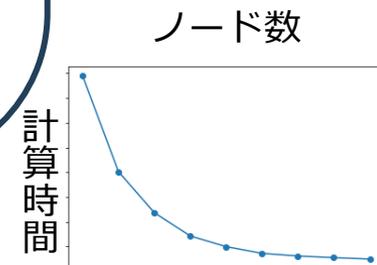




(参考) 並列ジョブの特性 並列化率と並列化効率(2)



ノード数をN倍にしても計算時間はN倍速くならない



並列性能特性はプログラム、データ、計算手法によって異なります。
★並列特性（どこで性能向上が止まるか）を見極めることが重要です。



アカウント

- ▶ 2週間有効です。

利用QUE

- ▶ 原則s001h_032c_lecを利用してください。

講習会内容等の質問

- ▶ lecture@j-focus.or.jpにお願いします。

接続環境

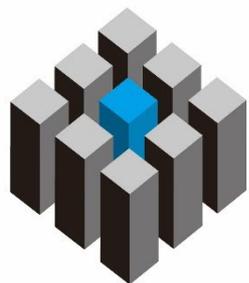
- ▶ RemoteViewの引き続きの利用はできません。
- ▶ ご自身の端末にVPNソフト、Teraterm等をご用意ください。
<https://vpn.j-focus.jp/>

講習会アンケートのお願い

- ▶ Zoom退出後、自動的にアンケート画面が立ち上がります。



「**計算機シミュレーション**」の第一歩は



FOCUS から !!

FOCUSスパコンは平成23年4月より共用を開始し、
「**産業利用向けスパコン**」として活用されています。



連絡先) 公益財団法人 計算科学振興財団
普及促進グループ

E-mail : fukyu@j-focus.or.jp

TEL: 078-599-5024

本資料に掲載いたしました価格等を含む情報につきましては、予告なく変更することがあります。