
NIMS スパコンを外部計算器資源として利用した報告書

リリース **1.0**

SIP-MI(A4)

2022 年 08 月 03 日

目次

第 1 章	概要	1
第 2 章	NIMS のスパコンシステムを外部計算対象とするには	2
第 3 章	外部計算の流れ	3
3.1	通常の形態	3
3.2	NIMS スパコン用の形態	4
第 4 章	準備	5
4.1	アカウント開設	5
4.1.1	中間計算機に関して	5
4.1.2	ポーリングスクリプト	6
4.1.3	API トークン	6
4.1.4	識別子	6
4.2	ワークフローの準備	6
4.3	NIMS スパコン用に用意するもの	6
4.3.1	事前処理プログラム	6
4.3.2	事後処理プログラム	7
4.3.3	qsub スクリプト	7
4.3.4	環境変数	7
4.4	ポーリングスクリプトの実行	8
第 5 章	ランについて	9

第 1 章

概要

本書は、NIMS スパコンシステムを外部計算機として MInt システムのワークフローから利用した場合の検証結果と、実際に利用する場合の準備、用意するなどをまとめたものである。

第 2 章

NIMS のスパコンシステムを外部計算対象とするには

NIMS のスパコンでは長時間のログインまたはプログラムの実行が禁止されている。他方、外部計算は MInt システムから SSH によるアクセスか、外部計算機からの API による MInt システムへのアクセスのどちらかしか利用方法がないが、前者は MInt システムが DMZ にあるため、直接 SSH できない。後者は長時間プログラム実行が禁止されているため、利用できない。

このため、中間に API で MInt システムにアクセスでき、SSH で NIMS スパコンへアクセスできる計算機を用意し、ここでユーザーごとに API ポーリングを行い、NIMS スパコンへは SSH で事前処理、qsub 処理、事後処理をおこなう方法をとることとした。

注釈: 事前処理、qsub 処理、事後処理は予め利用ユーザーがシェルスクリプトの形で用意する。

注釈: 事前処理、事後処理はオプションであり、必要なけれども良い。

第 3 章

外部計算の流れ

外部計算のうち API ポーリングを利用した利用の流れを示す。

3.1 通常の形態

標準的な外部計算実行のフローを 図 1 に示す。

- ワークフロー → モジュール指定のプログラム（A）の実行
- モジュール指定のプログラムは外部計算 API アクセススクリプトの起動 → API（B）へ
- 外部計算機は外部計算機 API アクセスプログラム（C）を起動 → API（B）へ（ポーリング）
- 外部計算機は外部計算機 API アクセスプログラム（C） → 外部計算機の特定のプログラム（D）を実行（API 経由外部計算の実行）

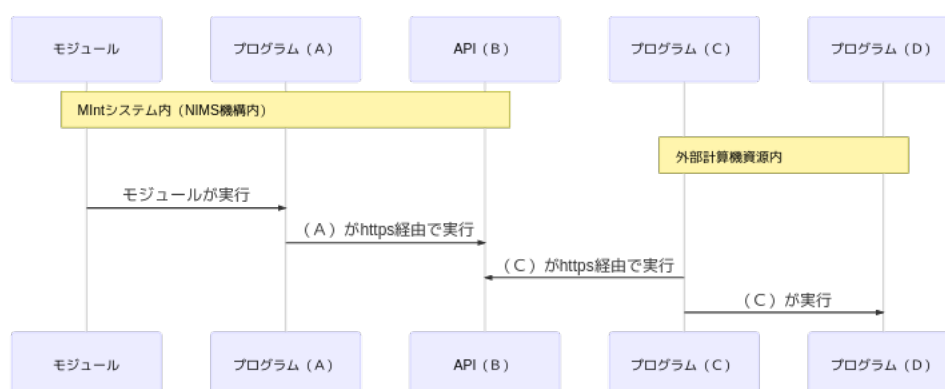


図 1 標準的な外部計算実行のフロー

3.2 NIMS スパコン用の形態

NIMS スパコン向けの外部計算実行のフローを 図 2 に示す。

- ワークフロー → モジュール指定のプログラム (A) の実行
- モジュール指定のプログラム (A) は外部計算 API アクセススクリプトの起動 → API (B) へ
- 外部計算用中間サーバー機は外部計算機 API アクセスプログラム (C) を起動 → API (B) へ (ポーリング)
- 外部計算用中間サーバー機は外部計算機 API アクセスプログラム (C) → 中間プログラム (D) を実行 (API 経由外部計算の実行)
- 中間プログラム (D) → 事前処理、qsub 処理、事後処理 (E) を実行

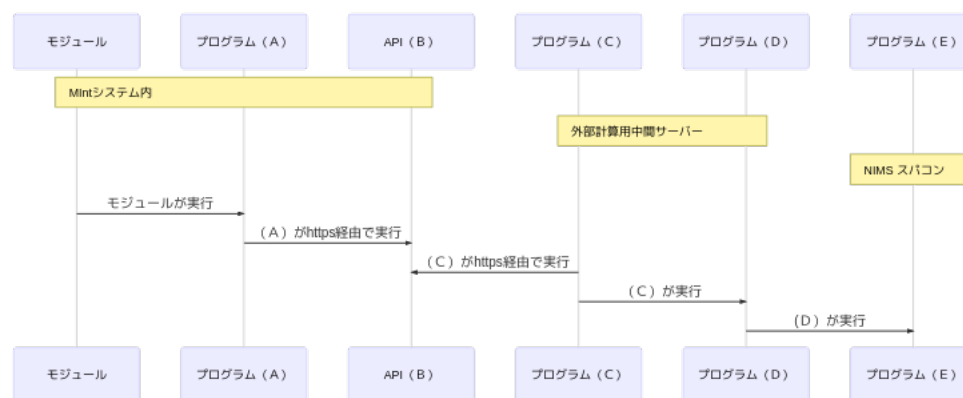


図 2 NIMS スパコン向け外部計算実行のフロー

注釈: 黄色い四角の中のシステム、サーバーおよび計算機はすべて NIMS 機構内にある。

第 4 章

準備

準備は以下の順番で進める。概ね、通常の外部計算機利用のマニュアルに準じるが、NIMS スパコンの場合はその制限からいくつか違いがある。

- アカウント開設
- ワークフロー作成申請を A4 班へ実装調査書の作成と送付
- NIMS スパコンで実行する必要があるプログラム類は、事前処理、qsub 処理、事後処理に分けて利用者が用意する。
- 準備ができれば、ポーリングスクリプトを実行し、待機する。

4.1 アカウント開設

アカウントなどの準備は以下のように進めるが、NIMS スパコンと A4 班が用意する中間計算機が対象である。

- NIMS スパコンアカウントの開設。
- スパコン利用方法に則り利用者が行う。
- A4 班が用意する中間計算機にアカウントを開設する。
- 利用者が A4 班に申請し、A4 班側でアカウントを作成する。以降は利用者が実施する。
 - ログインは公開鍵暗号方式とする。
 - 公開鍵を作成、申請時に A4 班へ伝える。
 - API ポーリングには専用の識別子が必要なので利用者側で決定し、申請時に A4 班へ伝える。
 - 利用者の API トークンを A4 班へ伝え、設定を行う。
- ここにポーリングスクリプトを配置する必要がある。
 - このスクリプトは A4 班が運用する Github リポジトリがあるので、そこからダウンロードする。

4.1.1 中間計算機に関して

サーバー名は、carta04.nims.go.jp である。アカウントは A4 班へ申請し、作成は A4 班が実施する。

4.1.2 ポーリングスクリプト

ポーリングスクリプトは、Github から以下のように git コマンドを使って clone する。

```
$ cd <ポーリングスクリプトを配置したい場所> $ git clone https://github.com/  
↪materialsintegration/misrc_distributed_computing_assist_remote_side.git``
```

4.1.3 API トークン

このワークフローを利用するユーザーで、MInt システムにログインし、ユーザープロフィール管理画面から取得できる。これを A4 班へ伝える。利用するユーザーが複数いる場合はそれぞれ伝えること。

4.1.4 識別子

ポーリングシステムでは識別子で MInt システムと外部計算機とを関連付ける。他にもいくつかあるので重複しないものであれば何でも良い。多くの場合企業名や部署名などをアルファベットで記述したものとすることが多い。本検証では、nims-supercom-a4 としている。

4.2 ワークフローの準備

ワークフローおよび MInt システム側のプログラム類は A4 班側で用意する。実装調査書を予め作成依頼するので、作成し、A4 班へ提出する。

ワークフローの準備ができれば、ワークフロー ID を知らせるので、ポーリングスクリプトを実行開始し、お知らせしたワークフロー ID のワークフローを実行する。

4.3 NIMS スパコン用に用意するもの

用意するのは以下の通り。

- 事前処理プログラム（オプション）
- 事後処理プログラム（オプション）
- qsub スクリプト（必須）
- 環境変数 **REMOTE_HOSTNAME** の設定

注釈: 用意するファイル名は A4 班で指定したもの以外は A4 班に知らせる必要はない。各プログラム内で整合性が取れていれば問題ない。

注釈: ダウンロードしたポーリングスクリプトに **resources** ディレクトリがあるので、ここに事前処理、事後処理、qsub スクリプトを用意する。（利用者）

4.3.1 事前処理プログラム

外部計算機で実行される。qsub スクリプト実行前の処理を別にする必要がある時に利用する。必要なければファイルを作成しない。ファイル名は、**pre_process.sh** である。ダウンロードしたディレクトリの下に、**resources** ディレクトリがあるので、ここへ配置する。

4.3.2 事後処理プログラム

外部計算機で実行される。qsub スクリプト実行後の処理を別にする必要がある時に利用する。必要なければファイルを作成しない。ファイル名は、post_process.sh である。ダウンロードしたディレクトリの下に、resources ディレクトリがあるので、ここへ配置する。

4.3.3 qsub スクリプト

このスクリプトファイルは必須である。qsub 実行のためのスクリプトを記述する。ファイル名は、qsub_process.sh である。ダウンロードしたディレクトリの下に、resources ディレクトリがあるので、ここへ配置する。

4.3.4 環境変数

qsub 実行するためのサーバー名を環境変数名 **REMOTE_HOSTNAME** で指定する。通常 NIMS Super Computer へアクセスする時のホスト名でも可能であるが、**~/ssh/config** などに設定したホスト名でも構わない。公開鍵暗号を指定したホスト名にしたい場合は通常指定するように、それも含めて設定しておく。設定場所は、**~/bashrc** など。

注釈: ポーリングスクリプトはこの環境変数を設定した後に動作させる。

4.4 ポーリングスクリプトの実行

ポーリングは以下のようにして中間計算機にて実行する。

```
$ python3.6 mi-system-remote.py <識別子> https://nims.mintsys.jp <token> calc_base:~/
↳remote_calculation
```

- 実際のポーリング動作
- MInt システムから外部計算への切り替え
- パラメータ送信
- パラメータ保存
- 外部計算開始
- 計算終了後、必要なファイルの受信
- MInt システムへ結果の送信
- MInt システムへ終了の通知

実際にポーリング動作中の画面のイメージを 図 3 に示す。

```
code = 0300 / message = サイト ID(nims-supercom-a4)の accept_id(None)の情報はありません。
There is calc in MI-system by accept_id(66fb3db9-a22a-436d-91d7-2a9a2d2d76fa)
計算情報を取得します。
2022/07/25 18:36:25:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/calc-params?accept_id=66fb3db9-a22a-436d-91d7-2a9a2d2d76fa&site_id=nims-supercom-a4)を送信
計算情報が取得できませんでした。
2022/07/25 18:36:35:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/calc-params-complete)を送信
code = 200 / message = ok
計算を開始します。
2022/07/25 18:36:45:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/calc-start)を送信
code = 200 / message = ok
ファイル名(calccond.txt)をバイナリファイルとして取り出し、保存しました
ファイル名(hazmeshccv.in)をバイナリファイルとして取り出し、保存しました
ファイル名(matpara.xml)をバイナリファイルとして取り出し、保存しました
ファイル名(property.xml)をバイナリファイルとして取り出し、保存しました
計算中...(/home/misystem/remote_calculation/66fb3db9-a22a-436d-91d7-2a9a2d2d76fa)
実行結果 (0)
計算終了。計算終了を通知します
2022/07/25 21:17:08:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/calc-end)を送信
code = 200 / message = ok
結果をアップロードします。
filename(huddleston.dat) is b'text/plain; charset=us-ascii\n'
filename(huddleston.dat) size is 12
filename(計算標準エラー出力.txt) is b'text/plain; charset=us-ascii\n'
filename(計算標準エラー出力.txt) size is 4
filename(計算標準出力.txt) is b'text/plain; charset=utf-8\n'
filename(計算標準出力.txt) size is 1808
2022/07/25 21:17:18:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/send-results)を送信
code = 200 / message = ok
アップロード終了
2022/07/25 21:17:28:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/end-send)を送信
code = 200 / message = ok
全行程が終了したので、待ち受け状態に遷移します。
2022/07/25 21:17:58:リクエスト (https://nims.mintsys.jp:50443/mi-distcomp-api/calc-request?site_id=nims-supercom-a4)を送信
code = 0300 / message = サイト ID(nims-supercom-a4)の accept_id(None)の情報はありません。
```

図 3 API ポーリングの実行イメージ

第 5 章

ランについて

ランは通常の MInt システムの実行によるものと変わらない。ラン詳細からの実行状況表示および完了後のダウンロードも通常のランと全く同じように扱える。

注釈: 外部計算中の状況は中間計算機のポーリングスクリプトに「計算中」としか表示できない。

計算中のログは中間計算機の `~/remote-calculation` 以下に `UUID` でできたディレクトリが作られるので、そこに「計算標準出力.txt」と「計算標準エラー出力.txt」というファイルで作成される。

以上