

AOBA-A
プログラム開発・実行環境
利用手順書

日本電気株式会社

2023年8月23日

2.0版

<< 改版履歴 >>

版数	改版日	改版理由	改版者
第一版	2020.10.17	第一版	NEC
第二版	2023.8.23	AOBA-AとAOBA-Sの互換性について記載を追加 環境変数“VE_OMP_NUM_THREADS=8”のデフォルト 設定化に伴い、スクリプト例の “PBS -v OMP_NUM_THREADS=8”を “export VE_OMP_NUM_THREADS=8”に変更 venode 既定値設定に合わせて mpirun の--venode 記 載を削除	NEC

目次

1. AOBA-A 実行プログラム開発・実行環境.....	3
1) AOBA-A と AOBA-S の互換性	3
2. プログラム開発・実行	3
1) プログラム のコンパイル.....	3
2) ジョブスクリプト作成と実行	5
3) 利用上の注意.....	7
3. NLC ライブラリの利用	9

1. AOBA-A 実行プログラム開発・実行環境

AOBA-A は、NEC 社製ベクトルエンジン(VE)を 8 基搭載したノード 72 台で構成しています。
V 当り 8 コアを搭載しており、ノード内並列では、8 共有メモリ並列までのプログラム実行および、複数ノードを使用した MPI 並列プログラム実行が行えます。

AOBA-A で実行するプログラムの開発・実行には、VE の持つベクトルアーキテクチャの特性、高 Core 性能、高メモリバンド幅を発揮するための専用の環境を利用します。

開発環境	ノード内並列	MPI 実行環境	科学技術計算ライブラリ	備考
NEC SDK	自動並列 OpenMP	NEC MPI	NEC NLC	

1) AOBA-A と AOBA-S の互換性

AOBA-S 用にコンパイルされた実行プログラムは AOBA-A では実行できません。AOBA-A 用にコンパイルされた実行プログラムは AOBA-S でも実行可能です。ただし、AOBA-S の環境に合わせて最適化されていないので、AOBA-S で実行するプログラムは、AOBA-S でコンパイルすることを強く推奨します。

2. プログラム開発・実行

1) プログラム のコンパイル

① ノード内実行プログラム

コンパイラコマンド : nfort(Fortran プログラム)
 ncc(C プログラム)
 nc++(C++プログラム)

実行形式 : 実行オブジェクト a.out を、カレントディレクトリに作成する場合

<コンパイラコマンド> <コンパイラオプション> <プログラムソースファイル名> 例) \$ nfort -O3 sample.f

コンパイラオプション例 :

-O _n	最適化レベルを指定する
4	Fortran 言語仕様を逸脱した副作用を伴う最大限の最適化・自動ベクトル化を適用する

3	副作用を伴う最適化・自動ベクトル化、および、多重ループの最適化を適用する
2	(既定値) 副作用を伴う最適化・自動ベクトル化を適用する
1	副作用を伴わない最適化・自動ベクトル化を適用する
0	最適化、自動ベクトル化、並列化、インライン展開を適用しない
-mparallel	自動並列化を利用する
-fopenmp	OpenMP 並列化を利用する
-mno-parallel-omp-routine	-mparallel と-fopenmp が同時に指定された時、OpenMP ディレクティブを含むルーチンを自動並列化しない
-finline-functions	自動インライン展開を利用する
-fbounds-check	バウンズチェック(配列の上下限のチェック)を行う
-ftrace	性能解析 ftrace 機能用の実行ファイルを作成する
-report-diagnostics	ベクトル化診断リスト(ソースファイル名.L)を出力する
-fdiag-vector=2	詳細なベクトル化診断メッセージを出力する (既定値は、-fdiag-vector=1)
-report-format	ベクトル化、並列化などの最適化情報がソース行とともに出力された編集リスト(ソースファイル名.L)を出力する

NEC SDK のコンパイルオプションの詳細は以下を参照下さい。

<https://www.hpc.nec/documentation> (NEC SDK)

②AOBA-A と AOBA-S の互換性

・ハードウェアアーキテクチャー

-march=ve1(規定値 AOBA-A 用のハードウェアアーキテクチャー指定)

-march=ve3(AOBA-S 用のハードウェアアーキテクチャー指定)

・半精度浮動小数点

AOBA-A では半精度浮動小数点を利用できません

③ MPI 実行プログラム

コンパイラコマンド： mpinfort(Fortran プログラム)

mpincc(Cプログラム)
mpinc++(C++プログラム)

実行形式：実行オブジェクト a.out を、カレントディレクトリに作成する場合

```
<コンパイラコマンド> <コンパイラオプション> <プログラムソースファイル名>  
例)  
$ mpinfort -O3 sample.f
```

コンパイラオプション例：

ノード内実行プログラム作成時と同じオプションが利用できます。

2) ジョブスクリプト作成と実行

作成した実行プログラムは、バッチ処理環境(NQSV：NEC Network Queuing System V)に対し、バッチリクエストという形式で実行を依頼します。

バッチリクエストで実行を依頼するためには、プログラムの実行を記述した、ジョブスクリプトを作成します。

① ノード内実行プログラム用ジョブスクリプト

バッチリクエストを処理するための、実行時間などのパラメータを定義して、実行プログラムの実行を記述します。

例(sh形式の場合)：

```
#!/bin/sh  
#PBS -q sx ..... (a)  
#PBS -l elapstim_req=2:00:00 ..... (b)  
#PBS --venode 1 ..... (c)  
#PBS -A kadai1 ..... (d)  
export VE_OMP_NUM_THREADS=8 ..... (e)  
cd $PBS_O_WORKDIR ..... (f)  
./a.out ..... (g)
```

記述詳細：

- (a) : リクエスト投入先キューとしてAOBA-A用キューを指定
- (b) : 最大実行経過時間を指定(hh:mm:ss形式)
- (c) : 使用VE数を指定、ノード内実行プログラムの場合には1固定
- (d) : 課金先のプロジェクトコードを指定(任意)、未指定の場合はデフォルトのプロジェクト

ェクトコードに課金

- (e) : 自動並列/OpenMP 並列実行プログラムの場合、並列数を指定
未指定の場合、デフォルト値として 8 を設定
- (f) : カレントディレクトリへ移動
- (g) : 実行プログラム名を指定

② MPI 実行プログラム用ジョブスクリプト

MPI 実行プログラムは、mpirun コマンドで実行プログラムを実行します。

バッチリクエストを処理するための、実行時間などのパラメータを定義して、mpirun コマンドで実行プログラムの実行を記述します。

例(sh 形式の場合) :

```
#!/bin/sh
#PBS -q sx                ..... (a)
#PBS -T necmpi            ..... (b)
#PBS -l elapstim_req=4:00:00 ..... (c)
#PBS -A kakin1           ..... (d)
cd $PBS_O_WORKDIR        ..... (e)
mpirun -np 256 ./a.out    ..... (f)
```

記述詳細 :

- (a) : リクエスト投入先キューとして AOBA-A 用キューを指定
- (b) : MPI 実行環境を指定, AOCC 開発環境では、openmpi 固定
- (c) : 最大実行経過時間を指定(hh:mm:ss 形式)
- (d) : 課金先のプロジェクトコードを指定(任意), 未指定の場合はデフォルトのプロジェクトコードに課金
- (e) : カレントディレクトリへ移動
- (f) : mpirun コマンドで実行プログラムを指定
-np オプションで MPI 総プロセス数を指定

NEC MPI のプログラム実行方法の詳細は以下を参照下さい。

<https://www.hpc.nec/documentation>

(NEC MPI ユーザーズガイド)

③ 自動並列/OpenMP 並列、MPI 併用実行プログラム用ジョブスクリプト

自動並列/OpenMP 並列と MPI を併用した実行プログラムは、mpirun コマンドで実行プログラムを実行します。

バッチリクエストを処理するための、実行時間などのパラメータに加え、自動並列/OpenMP 並列実行を定義して、mpirun コマンドで実行プログラムの実行を記述します。

例(sh 形式の場合) :

```
#!/bin/sh
#PBS -q sx                ..... (a)
#PBS -T necmpi            ..... (b)
#PBS -l elapstim_req=4:00:00 ..... (c)
#PBS -A kakin1           ..... (d)
export VE_OMP_NUM_THREADS=8 ..... (e)
cd $PBS_O_WORKDIR        ..... (f)
mpirun -np 32 ./a.out    ..... (g)
```

記述詳細 :

- (a) : リクエスト投入先キューとして AOBA-A 用キューを指定
- (b) : MPI 実行環境を指定, AOCC 開発環境では、openmpi 固定
- (c) : 最大実行経過時間を指定(hh:mm:ss 形式)
- (d) : 自動並列/OpenMP 並列実行プログラムの場合、並列数を指定
- (e) : 課金先のプロジェクトコードを指定(任意), 未指定の場合はデフォルトのプロジェクトコードに課金
- (f) : 自動並列/OpenMP 並列実行プログラムの場合、並列数を指定
未指定の場合、デフォルト値として 8 を設定
- (f) : カレントディレクトリへ移動
- (g) : mpirun コマンドで実行プログラムを指定
-np オプションで MPI 総プロセス数を指定

作成したジョブスクリプトを、バッチ処理環境(NQSV)に qsub コマンドを使って投入します。

```
qsub <ジョブスクリプトファイル名>
例)
$ qsub run.sh
```

3) 利用上の注意

- ① 旧システムの MPI プログラム実行ジョブスクリプトは、そのままでは実行はできません。ジョブスクリプトを、上記形式への修正をして実行して下さい。

② 旧システムの実行モジュールは実行できません。プログラムのコンパイルを行い AOBA-A 用の実行モジュールを作成して下さい。

③ MPI プログラム実行の標準的な実行方法(上記例)では MPI プロセスは、NQSV により割り当てられたノードの先頭ノードの先頭 VE のコアから順次割り当てられます。

例) 96 MPI プロセスを実行

```
#!/bin/sh
#PBS -T necmpi
#PBS -l elapstim_req=4:00:00
cd $PBS_O_WORKDIR
mpirun -np 96 ./a.out
```

⇒ 先頭ノードの 8 つの VE に 64 プロセス (8 コア/VE×8VE) 割り当て、次ノードの先頭 4 つの VE に 32 プロセス割り当てられる。

④ MPI 実行プログラムでは、各 MPI プロセスの出力が、標準出力、標準エラー出力ファイルのそれぞれ一つのファイルに混在して記録されます。各 MPI プロセス毎に出力を分離したい場合には、分離するスクリプトファイルを、実行プログラムの前に指定して実行します。

```
mpirun -np 256 /opt/nec/ve/bin/mpisep.sh ./a.out
```

ファイルの分離は、複数の方式があり、環境変数 NMPI_SEPSELECT で指定します。

NMPI_SEPSELECT :

- 1 : 各 MPI プロセスの標準出力だけを stdout.uuu:rrr へ保存
- 2 : (既定値) 各 MPI プロセスの標準エラー出力だけを stderr.uuu:rrr へ保存
- 3 : 各 MPI プロセスの標準出力 および 標準エラー出力を, それぞれ stdout.uuu:rrr および stderr.uuu:rrr に保存
- 4 : 各 MPI プロセスの標準出力 および 標準エラー出力を, 同一のファイル std.uuu:rrr に保存

例)

```
#!/bin/sh
#PBS -q sx
#PBS -T necmpi
#PBS -l elapstim_req=4:00:00
#PBS -A kakin1
cd $PBS_O_WORKDIR
```

```
export NMPI_SEPSELECT=3
mpirun -np 256 /opt/nec/ve/bin/mpisep.sh ./a.out
```

3. NLC ライブラリの利用

AOBA-A では、AOBA-A のアーキテクチャに対応した科学技術計算ライブラリ NLC(NEC Numeric Library Collection)を利用できます。

NLC は、広範な分野の数値シミュレーションプログラムの作成を強かに支援する数学ライブラリのコレクションであり、AOBA-A のベクトルエンジンに対応しています。NLC を用いることにより、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な科学技術計算プログラムを作成することができ、数値シミュレーションプログラム開発の生産性を大幅に改善することができます。

NLC の詳細および利用方法は、以下を参照下さい。

<https://www.hpc.nec/documentation>

(NLC (NEC Numeric Library Collection) ユーザーズガイド)