RSS2108-PJ7-ﾕｰｻﾞﾏﾆｭｱﾙ-001d

文部科学省次世代ＩＴ基盤構築のための研究開発
「革新的シミュレーションソフトウエアの研究開発」

RSS21 フリーソフトウエア

HEC ミドルウェア（HEC-MW）

PC クラスタ用ライブラリ型 HEC-MW

（hecmw-PC-cluster） バージョン 2.01

I0, 全体制御 マニュアル

お問い合わせ先
（公開／契約窓口） （財）生産技術研究奨励会
〒153-8505 東京都目黒区駒場４－６－１
（ソフトウェア管理元） 東京大学生産技術研究所 計算科学技術連携研究センター
〒153-8505 東京都目黒区駒場４－６－１
Fax：03-5452-6662

目　　次

1. 概要

並列計算を実施する場合，大規模かつ分散したデータを効率的に扱うことが必要である．特に並列 I/O については, 利用者が容易に大規模分散データの処理を実施できるような工夫が求められる．

HEC-MW では，大規模分散データの処理に必要と考えられる以下の並列 I/O について，容易に操作可能なインタフェースを提供する．

- 全体メッシュデータ入力
- 分散メッシュデータ入出力
- リスタートデータ入出力
- 結果データ入出力

また, 全体メッシュデータ入力においては, 以下のメッシュデータが読み込み可能である．

- HEC-MW 単一領域メッシュデータ
- GeoFEM メッシュデータ
- ABAQUS メッシュデータ（現版では未実装）
- NASTRAN メッシュデータ（現版では未実装）
- FEMAP メッシュデータ（現版では未実装）

それらの入力はただ一つのインタフェースによって行うことができる．さらに，異なるタイプのメッシュデータを同時に入力し，一つのメッシュデータとして扱う事も可能である．

本マニュアルは，HEC-MW の機能のうち，全体制御機能と並列 I/O についてその利用方法を述べたものである．

5. 分散メッシュデータ構造体の概要

分散メッシュデータ関連構造体は以下の各部分から構成されている．(*) をつけたものは実際に分散メッシュデータから読み込まれる情報を含んでいる：

(1) 制御情報 (*)
　内　容：全体的な制御に関する情報

(2) 分散メッシュ情報 (*)
　内　容：節点，要素，通信，階層構造
　構造体：hecmwST_local_mesh

(3) セクション情報 (*)
　内　容：各要素の属するセクションの情報
　構造体：hecmwST_section

(4) 材料物性情報 (*)
　内　容：各材料の物性情報，温度依存性テーブルも含む
　構造体：hecmwST_material

(5) 拘束グループ情報 (*)
　内　容：拘束グループに関する情報
　構造体：hecmwST_mpc

(6) グループ情報 (*)
　内　容：節点，要素，面グループに関する情報
　構造体：hecmwST_node_grp，hecmwST_elem_grp，hecmwST_surf_grp

5.1. 一般パラメータ設定

```
module hecmw_util
  implicit none
  include 'mpif.h'
  public

  integer(kind=4),parameter:: kint  = 4
  integer(kind=4),parameter:: kreal = 8

  integer(kind=kint),parameter :: HECMW_NAME_LEN     =   63
  integer(kind=kint),parameter :: HECMW_HEADER_LEN   =  127
  integer(kind=kint),parameter :: HECMW_MSG_LEN      =  255
  integer(kind=kint),parameter :: HECMW_FILENAME_LEN = 1023

  integer(kind=kint),parameter :: hecmw_sum              = 46801
  integer(kind=kint),parameter :: hecmw_prod             = 46802
  integer(kind=kint),parameter :: hecmw_max              = 46803
  integer(kind=kint),parameter :: hecmw_min              = 46804
  integer(kind=kint),parameter :: hecmw_integer          = 53951
  integer(kind=kint),parameter :: hecmw_single_precision = 53952
  integer(kind=kint),parameter :: hecmw_double_precision = 53953
  integer(kind=kint),parameter :: hecmw_character        = 53954
```

変数名	内容
kint	INT 変数のバイト長
kreal	REAL 変数のバイト長
HECMW_NAME_LEN	名前の最大長
HECMW_HEADER_LEN	HEADER の最大長
HECMW_MSG_LEN	ログメッセージの最大長
HECMW_FILENAME_LEN	ファイル名の最大長

5.2. 分散メッシュ情報

分散メッシュ情報について，Fortran 版を用いて説明する．C 版においても構造や変数名は同じである[1]が，配列の添字が 0 から始まることに注意しなければならない.

```
type hecmwST_local_mesh
```

(1) 全体情報

```
character(HECMW_FILENAME_LEN)          :: gridfile
character(HECMW_FILENAME_LEN),pointer :: files(:)
character(HECMW_HEADER_LEN)            :: header
integer(kind=kint) :: hecmw_flag_adapt
integer(kind=kint) :: hecmw_flag_initcon
integer(kind=kint) :: hecmw_n_file
integer(kind=kint) :: hecmw_flag_parttype
integer(kind=kint) :: hecmw_flag_partdepth
integer(kind=kint) :: hecmw_flag_version
real(kind=kreal)   :: zero_temp
```

変数名	配列要素数	内容
gridfile		グリッドファイル名
hecmw_n_file		もろもろのファイル数
files	(hecmw_n_file)	もろもろのファイル
header		ヘッダ
hecmw_flag_adapt		階層構造使用の有無（＝0：NO，＝1：YES）
hecmw_flag_initcon		初期条件使用　　（＝0：NO，＝1：YES）
hecmw_flag_parttype		領域分割形式（＝1：Node-based，＝2：Element-based）
hecmw_flag_partdepth		部分領域間袖領域のオーバーラップ深さ
hecmw_flag_version		バージョン
zero_temp		絶対零度（温度）

[1] 一部例外あり．hecmwST_local_mesh の MPI_COMM は，C 版では HECMW_COMM となる.

(2) 節点情報

```
integer(kind=kint)          :: n_node
integer(kind=kint)          :: nn_internal
integer(kind=kint)          :: n_dof
integer(kind=kint)          :: n_dof_grp
real(kind=kreal),pointer    :: node(:)
integer(kind=kint),pointer  :: node_ID(:)
integer(kind=kint),pointer  :: global_node_ID(:)
integer(kind=kint),pointer  :: node_val_index(:)
real(kind=kreal),pointer    :: node_val_item(:)
integer(kind=kint),pointer  :: node_dof_index(:)
integer(kind=kint),pointer  :: node_dof_item(:)
integer(kind=kint),pointer  :: node_init_val_index(:)
real(kind=kreal),pointer    :: node_init_val_item(:)
integer(kind=kint),pointer  :: node_internal_list(:)
```

変数名	配列要素数	内容
n_node		総節点数
nn_internal		内部節点数
n_dof		節点最大自由度数
n_dof_grp		自由度グループ数
node	3*n_node	節点座標 node(3*k-2)：節点 k の X 座標 node(3*k-1)：節点 k の Y 座標 node(3*k)：節点 k の Z 座標
node_ID	2*n_node	節点 ID（ローカル番号, 所属領域） node_id(2*i-1)：節点 i のローカル番号 node_id(2*i) ：節点 i の所属領域
global_node_ID	n_node	グローバル節点 ID
node_val_index	0:n_node	節点物理量インデックス
node_val_item	node_val_index(n_node)	節点物理量
node_dof_index	0:n_dof_grp	自由度用インデックス
node_dof_item	n_dof_grp	自由度
node_init_val_index	0: n_node	初期条件用インデックス
node_init_val_item	node_init_val_index(n_node)	初期条件
node_internal_list	nn_internal	内部節点リスト

(3) 要素情報

```
integer(kind=kint)          :: n_elem
integer(kind=kint)          :: ne_internal
integer(kind=kint)          :: n_elem_type
integer(kind=kint)          :: n_elem_mat_ID
integer(kind=kint),pointer :: elem_type_index(:)
integer(kind=kint),pointer :: elem_type_item(:)
integer(kind=kint),pointer :: elem_type(:)
integer(kind=kint),pointer :: section_ID(:)
integer(kind=kint),pointer :: elem_mat_ID_index(:)
integer(kind=kint),pointer :: elem_mat_ID_item(:)
integer(kind=kint),pointer :: elem_node_index(:)
integer(kind=kint),pointer :: elem_node_item(:)
integer(kind=kint),pointer :: elem_ID(:)
integer(kind=kint),pointer :: global_elem_ID(:)
integer(kind=kint),pointer :: elem_internal_list(:)
integer(kind=kint),pointer :: elem_mat_int_index(:)
real(kind=kreal),pointer   :: elem_mat_int_val(:)
integer(kind=kint),pointer :: elem_val_index(:)
real(kind=kreal),pointer   :: elem_val_item(:)
```

変数名	配列要素数	内容
n_elem		総要素数
ne_internal		内部要素数
n_elem_type		要素タイプの種類
n_elem_mat_ID		= elem_mat_ID_index(n_elem)
elem_type_index	0:n_elem_type	要素タイプのインデックス
elem_type_item	n_elem_type	要素タイプの内容
elem_type	n_elem	要素タイプ
section_ID	n_elem	セクション番号
elem_mat_ID_index	0:n_elem	材料物性 ID のための一次元インデックス（COMPOSITE 用）
elem_mat_ID_item	elem_mat_ID_index(n_elem)	材料物性 ID 配列（COMPOSITE 用）
elem_node_index	0:n_elem	要素コネクティビティ用一次元インデックス
elem_node_item	elem_node_index(n_elem)	要素コネクティビティ用配列
elem_ID	2*n_elem	要素 ID（ローカル番号，所属領域）
global_elem_ID	n_elem	グローバル要素 ID
elem_internal_list	ne_internal	内部要素リスト
elem_mat_int_index	0: n_elem	材料物性用一次元インデックス（内部処理用）
elem_mat_int_val	elem_mat_int_index(n_elem)	材料物性用配列（内部処理用）

elem_val_index	0:n_elem	要素物理量インデックス
elem_val_item	elem_val_index(n_elem)	要素物理量

（4） PE および通信情報

```
integer(kind=kint)          :: zero
integer(kind=kint)          :: MPI_COMM
integer(kind=kint)          :: PETOT
integer(kind=kint)          :: PEsmpTOT
integer(kind=kint)          :: my_rank
integer(kind=kint)          :: errnof
integer(kind=kint)          :: n_subdomain
integer(kind=kint)          :: n_neighbor_pe
integer(kind=kint),pointer :: neighbor_pe(:)
integer(kind=kint),pointer :: import_index(:)
integer(kind=kint),pointer :: import_item(:)
integer(kind=kint),pointer :: export_index(:)
integer(kind=kint),pointer :: export_item(:)
integer(kind=kint),pointer :: shared_index(:)
integer(kind=kint),pointer :: shared_item(:)
```

変数名	配列要素数	内容
MPI_COMM		MPI 用コミュニケータ
zero		領域番号=0 であれば「1」，それ以外は「0」
PETOT		総領域数
PEsmpTOT		SMP ノードあたりの PE 数
my_rank		プロセス番号
errnof		エラーID（FORTRAN でのみ使用）
n_subdomain		総領域数（局所分散データからの読み込み）
n_neighbor_pe		隣接領域数
neighbor_pe	n_neighbor_pe	隣接領域 ID
import_index	0: n_neighbor_pe	受信テーブル用一次元インデックス
import_item	import_index(n_neighbor_pe)	受信テーブル用配列
export_index	0: n_neighbor_pe	送信テーブル用一次元インデックス
export_item	export_index(n_neighbor_pe)	送信テーブル用配列
shared_index	0: n_neighbor_pe	送受信テーブル用一次元インデックス
shared_item	shared_index(n_neighbor_pe)	送受信テーブル用配列

(5) 階層構造

```
        integer(kind=kint)         :: coarse_grid_level
        integer(kind=kint)         :: n_adapt
        integer(kind=kint),pointer :: when_i_was_refined_node(:)
        integer(kind=kint),pointer :: when_i_was_refined_elem(:)
        integer(kind=kint),pointer :: adapt_parent_type(:)
        integer(kind=kint),pointer :: adapt_type (:)
        integer(kind=kint),pointer :: adapt_level(:)
        integer(kind=kint),pointer :: adapt_parent(:)
        integer(kind=kint),pointer :: adapt_children_index(:)
        integer(kind=kint),pointer :: adapt_children_item(:)
```

変数名	配列要素数	内容
coarse_grid_level		最も粗いメッシュのレベル
n_adapt		Refinement された回数
when_i_was_refined_node	n_node	各節点の生成された Refinement レベル
when_i_was_refined_elem	n_elem	各要素の生成された Refinement レベル
adapt_parent_type	n_elem	各要素の親要素の分割タイプ
adapt_type	n_elem	各要素の分割タイプ
adapt_level	n_elem	各要素の分割レベル
adapt_parent	2*n_elem	各要素の親要素 ID（ローカル番号，所属領域）
adapt_children_index	0:n_elem	各要素の子要素用一次元インデックス
adapt_children_item	2*adapt_children_index(n_elem)	各要素の子要素 ID（ローカル番号，所属領域）

(6) 下部構造

```
        type (hecmwST_section)   :: section
        type (hecmwST_material)  :: material
        type (hecmwST_mpc)       :: mpc
        type (hecmwST_amplitude) :: amp
        type (hecmwST_node_grp)  :: node_group
        type (hecmwST_elem_grp)  :: elem_group
        type (hecmwST_surf_grp)  :: surf_group
```

```
      end type hecmwST_local_mesh
```

5.3. セクション情報

```
type hecmwST_section
  integer(kind=kint)          :: n_sect
  integer(kind=kint),pointer :: sect_type(:)
  integer(kind=kint),pointer :: sect_opt(:)
  integer(kind=kint),pointer :: sect_mat_ID_index(:)
  integer(kind=kint),pointer :: sect_mat_ID_item(:)
  integer(kind=kint),pointer :: sect_I_index(:)
  integer(kind=kint),pointer :: sect_I_item(:)
  integer(kind=kint),pointer :: sect_R_index(:)
  real(kind=kreal),pointer   :: sect_R_item(:)
end type hecmwST_section
```

変数名	配列要素数	内容
n_sect		セクション数
sect_type	n_sect	セクションタイプ （SOLID：1，SHELL：2，BEAM：3，INTERFACE:4）
sect_opt	n_sect	SECOPT
sect_mat_ID_index	0:n_sect	材料物性 ID のための一次元インデックス（COMPOSITE 用）
sect_mat_ID_item	sect_mat_ID_item(n_sect)	材料物性 ID
sect_I_index	0:n_sect	セクション値用一次元インデックス（整数）
sect_I_item	sect_I_index(n_sect)	セクション値配列（整数）
sect_R_index	0:n_sect	セクション値用一次元インデックス（実数）
sect_R_item	sect_R_index(n_sect)	セクション値配列（実数）

5.4. 材料物性情報

```
type hecmwST_ material
  integer(kind=kint)          :: n_mat
  integer(kind=kint)          :: n_mat_item
  integer(kind=kint)          :: n_mat_subitem
  integer(kind=kint)          :: n_mat_table
  character(HECMW_NAME_LEN),pointer :: mat_name(:)
  integer(kind=kint),pointer :: mat_item_index(:)
  integer(kind=kint),pointer :: mat_subitem_index(:)
  integer(kind=kint),pointer :: mat_table_index(:)
  real(kind=kreal),pointer   :: mat_val(:)
  real(kind=kreal),pointer   :: mat_temp(:)
end type hecmwST_material
```

変数名	配列要素数	内容
n_mat		材料数
n_mat_item		材料物性総数
n_mat_subitem		材料物性サブ項目総数
n_mat_table		材料物性テーブル総数
mat_name	n_mat	材料物性名
mat_item_index	0:n_mat	材料数一次元インデックス
mat_subitem_index	0:n_mat_item	材料物性数一次元インデックス
mat_table_index	0:n_mat_subitem	材料物性テーブル一次元インデックス
mat_val	mat_table_index(n_mat_subitem)	材料物性
mat_temp	mat_table_index(n_mat_subitem)	温度依存テーブル

5.5. 拘束グループ情報

```
type hecmwST_mpc
  integer(kind=kint)          :: n_mpc
  integer(kind=kint),pointer :: mpc_index(:)
  integer(kind=kint),pointer :: mpc_item(:)
  integer(kind=kint),pointer :: mpc_dof(:)
  real(kind=kreal),pointer   :: mpc_val(:)
end type hecmwST_mpc
```

変数名	配列要素数	内容
n_mpc		拘束グループ数
mpc_index	0:n_mpc	拘束グループ用一次元インデックス
mpc_item	mpc_index(n_mpc)	拘束グループ節点番号
mpc_dof	mpc_index(n_mpc)	拘束自由度情報
mpc_val	mpc_index(n_mpc)	拘束自由度係数

5.6. AMPLITUDE 情報

```
type hecmwST_amplitude
  integer(kind=kint)          :: n_amp
  character(len=HECMW_NAME_LEN),pointer :: amp_name(:)
  integer(kind=kint),pointer :: amp_type_definition(:)
  integer(kind=kint),pointer :: amp_type_time(:)
  integer(kind=kint),pointer :: amp_type_value(:)
  integer(kind=kint),pointer :: amp_index(:)
  real(kind=kreal),pointer   :: amp_val(:)
  real(kind=kreal),pointer   :: amp_table(:)
end type hecmwST_amplitude
```

変数名	配列要素数	内容
n_amp		AMPLITUDE グループ数
amp_name	n_amp	AMPLITUDE グループ名
amp_type_definition	n_amp	AMPLITUDE タイプ（TABLAR：１）
amp_type_time	n_amp	AMPLITUDE タイプ（STEP TIME：１）
amp_type_value	n_amp	AMPLITUDE タイプ （RELATIVE：１，　ABSOLUTE：２）
amp_index	0:n_amp	AMPLITUDE インデックス
amp_val	amp_index(n_amp)	AMPLITUDE 値
amp_table	amp_index(n_amp)	AMPLITUDE 時間

5.7. グループ情報（節点）

```
type hecmwST_node_grp
  integer(kind=kint)          :: n_grp
  integer(kind=kint)          :: n_bc
  character(HECMW_NAME_LEN),pointer :: grp_name(:)
  integer(kind=kint),pointer :: grp_index(:)
  integer(kind=kint),pointer :: grp_item(:)
  integer(kind=kint),pointer :: bc_grp_ID(:)
  integer(kind=kint),pointer :: bc_grp_type(:)
  integer(kind=kint),pointer :: bc_grp_index(:)
  integer(kind=kint),pointer :: bc_grp_dof(:)
  real(kind=kreal),pointer   :: bc_grp_val(:)
end type hecmwST_node_grp
```

変数名	配列要素数	内容
n_grp		グループ数
grp_name	n_grp	グループ名
grp_index	0:n_grp	グループ要素用一次元インデックス
grp_item	grp_index(n_grp)	グループ要素用配列
n_bc		境界条件設定節点数（自由度数）
bc_grp_ID	n_bc	所属節点グループ番号
bc_grp_type	n_bc	境界条件タイプ, =1：変位, =2：Flux 負の場合はユーザーサブルーチン
bc_grp_index	n_bc	境界条件設定節点番号
bc_grp_dof	n_bc	境界条件設定自由度
bc_grp_val	n_bc	境界条件値

5.8. グループ情報（要素）

```
type hecmwST_elem_grp
  integer(kind=kint)          :: n_grp
  integer(kind=kint)          :: n_bc
  character(HECMW_NAME_LEN),pointer :: grp_name(:)
  integer(kind=kint),pointer :: grp_index(:)
  integer(kind=kint),pointer :: grp_item(:)
  integer(kind=kint),pointer :: bc_grp_ID(:)
  integer(kind=kint),pointer :: bc_grp_type(:)
  integer(kind=kint),pointer :: bc_grp_index(:)
  real(kind=kreal),pointer   :: bc_grp_val(:)
end type hecmwST_elem_grp
```

変数名	配列要素数	内容
n_grp		グループ数
grp_name	n_grp	グループ名
grp_index	0:n_grp	グループ要素用一次元インデックス
grp_item	grp_index(n_grp)	グループ要素用配列
n_bc		境界条件設定要素数
bc_grp_ID	n_bc	所属要素グループ番号
bc_grp_type	n_bc	境界条件タイプ, =1：変位，=2：Flux 負の場合はユーザーサブルーチン
bc_grp_index	n_bc	境界条件設定要素番号
bc_grp_val	n_bc	境界条件値

5.9. グループ情報（面）

```
type hecmwST_surf_grp
  integer(kind=kint)          :: n_grp
  integer(kind=kint)          :: n_bc
  character(HECMW_NAME_LEN),pointer:: grp_name(:)
  integer(kind=kint),pointer :: grp_index(:)
  integer(kind=kint),pointer :: grp_item(:)
  integer(kind=kint),pointer :: bc_grp_ID(:)
  integer(kind=kint),pointer :: bc_grp_type(:)
  integer(kind=kint),pointer :: bc_grp_index(:)
  real(kind=kreal),pointer   :: bc_grp_val(:)
end type hecmwST surf_grp
```

変数名	配列要素数	内容
n_grp		面グループ数
grp_name	n_grp	面グループ名
grp_index	0:n_grp	面グループ要素用一次元インデックス
grp_item	2*grp_index(n_grp)	面グループ要素用配列（要素，局所面番号） grp_item(2*k-1)：要素番号 grp_item(2*k　)：局所面番号
n_bc		境界条件設定要素数（自由度数）
bc_grp_ID	n_bc	所属要素グループ番号
bc_grp_type	n_bc	境界条件タイプ, =1：変位，=2： Flux 負の場合はユーザーサブルーチン
bc_grp_index	2*n_bc	境界条件設定面番号（要素，局所面番号） bc_grp_index(2*k-1)：要素番号 bc_grp_index(2*k　)：局所面番号
bc_grp_val	n_bc	境界条件値

```
end module hecmw_ util
```

6. 結果データ

6.1. 結果データ構造体

```
type hecmwST_result_data
    integer(kind=kint) :: nn_component
    integer(kind=kint) :: ne_component
    integer(kind=kint),pointer :: nn_dof(:)
    integer(kind=kint),pointer :: ne_dof(:)
    character(len=HECMW_NAME_LEN),pointer :: node_label(:)
    character(len=HECMW_NAME_LEN),pointer :: elem_label(:)
    real(kind=kreal),pointer :: node_val_item(:)
    real(kind=kreal),pointer :: elem_val_item(:)
end type hecmwST_result_data
```

変数名	内容
nn_component	節点コンポーネント数
ne_component	要素コンポーネント数
nn_dof	節点自由度数
ne_dof	要素自由度数
node_label	節点ラベル
elem_label	要素ラベル
node_val_item	節点値 値の並びは，節点ごとにその節点の全コンポーネントが格納される
elem_val_item	要素値 値の並びは，要素ごとにその要素の全コンポーネントが格納される

6.2. 結果ファイルフォーマット

```
header
n_node  n_elem
nn_component  ne_component
nn_dof(1)  nn_dof(2)  ...  nn_dof(nn_comonent)
node_label(1)
node_label(2)
...
node_label(nn_component)
node_global_ID(1)
node_val_item(1)  node_val_item(2)  ...  node_val_item(sum(nn_dof))
node_global_ID(2)
...
ne_dof(1)  ne_dof(2)  ...  ne_dof(ne_comonent)
elem_label(1)
elem_label(2)
...
elem_label(ne_component)
elem_global_ID(1)
elem_val_item(1)  elem_val_item(2)  ...  elem_val_item(sum(ne_dof))
...
```

ただし,

header : hecmw_result_init API で指定した任意の文字列

n_node : 分散メッシュ構造体の節点数 n_node

n_elem : 分散メッシュ構造体の要素数 n_elem

である.