

東京大学本郷キャンパス 工学部8号館 84講義室 (地下1階)

新機能紹介： 並列パーティショナの開発

2017年1月30日
第33回FrontISTR研究会
＜FrontISTR開発の最新動向＞

謝辞

本研究開発は、文部科学省ポスト「京」
重点課題⑧「近未来型ものづくりを先導する
革新的設計・製造プロセスの開発」の
一環として実施したものです。

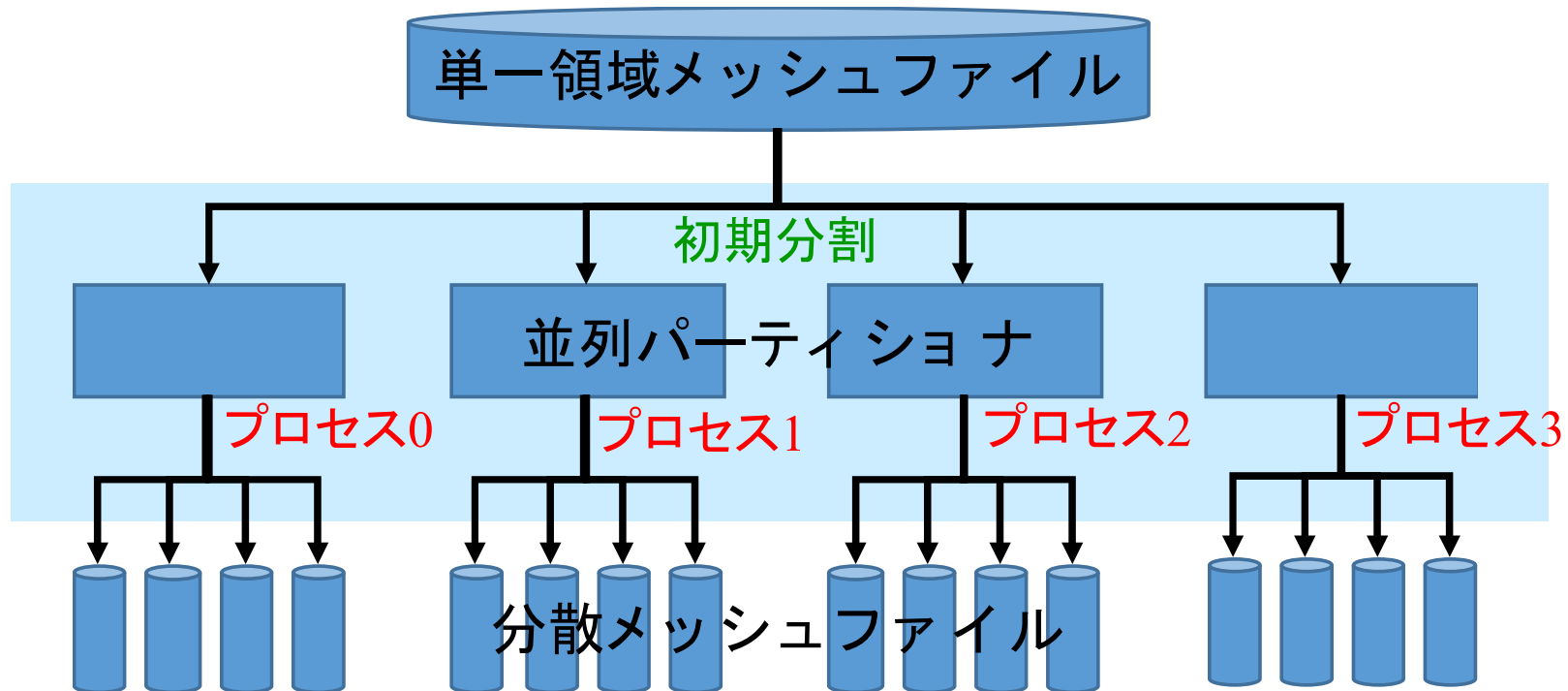
新機能紹介: 並列パーティショナの開発

発表内容

1. 並列パーティショナの基本アルゴリズム
2. 今回開発された並列パーティショナのプログラム
3. 並列パーティショナの性能検証
4. 現状のまとめと今後の課題

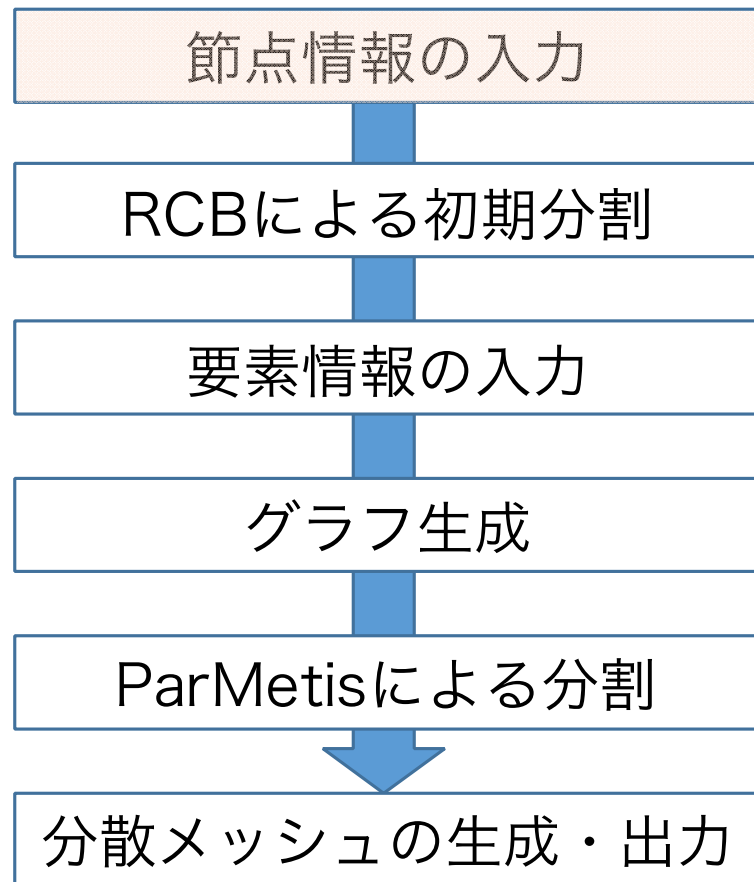
並列パーティショナの基本アルゴリズム

- 並列パーティショナは、 2^n のMPIプロセス数で実行される
- MPIプロセス数は初期分割数と一致し、
出力する分散領域メッシュの領域数とは異なる



Parallel execution with 4 MPI processes and decomposition into 16 sub-domains (distributed domains)

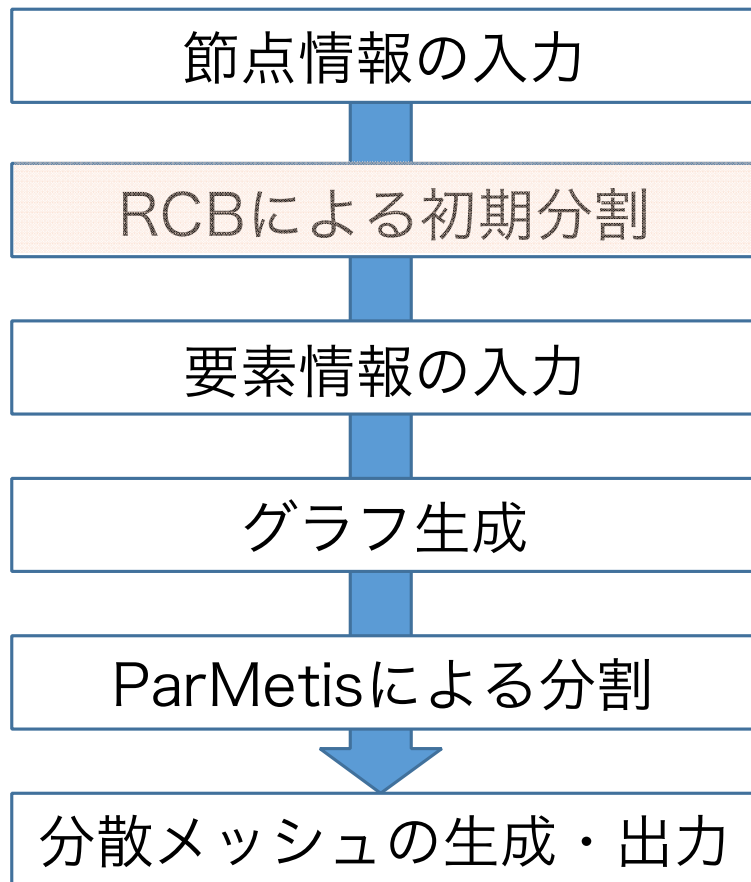
並列パーティショナの処理の流れ (1/6)



1. 全てのMPIプロセスで、節点座標データ (!NODE) を読み込み、メモリに格納する
2. 全てのMPIプロセスで、要素コネクティビティデータ (!ELEMENT) を読み込み、未使用の節点情報を削除する (要素コネクティビティデータはメモリに格納されない)

Flowchart of the whole processing

並列パーティショナの処理の流れ (2/6)

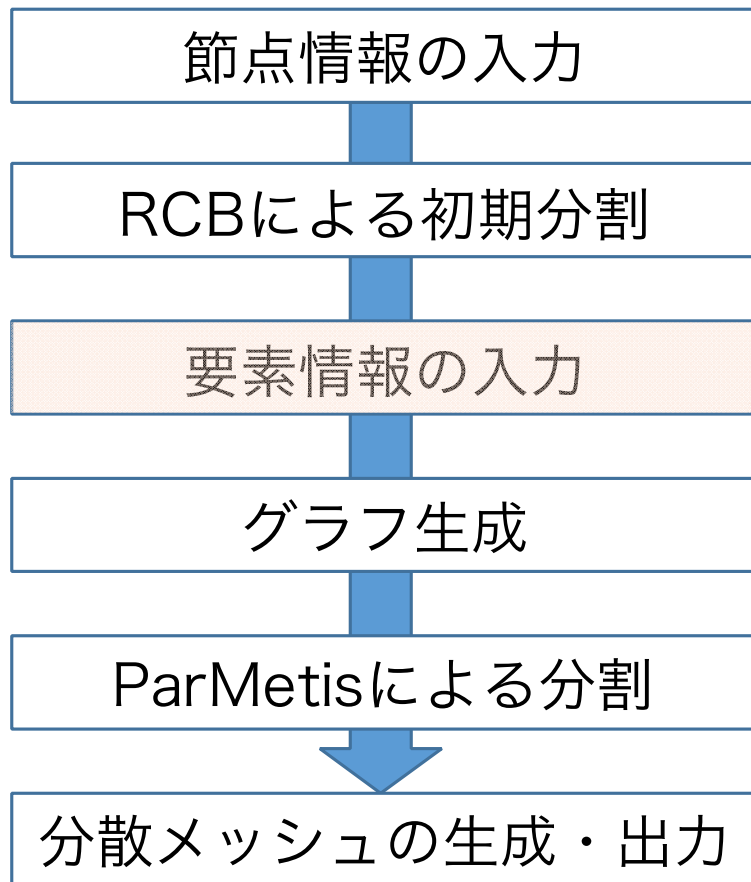


3. 全てのMPIプロセスで、節点を座標情報を用いて、RCB (Recursive Coordinate Bisection) によって並列パーティショナのMPIプロセス数の領域に分割 (以降, これを初期分割という)

※ 初期分割数 = MPIプロセス数

Flowchart of the whole processing

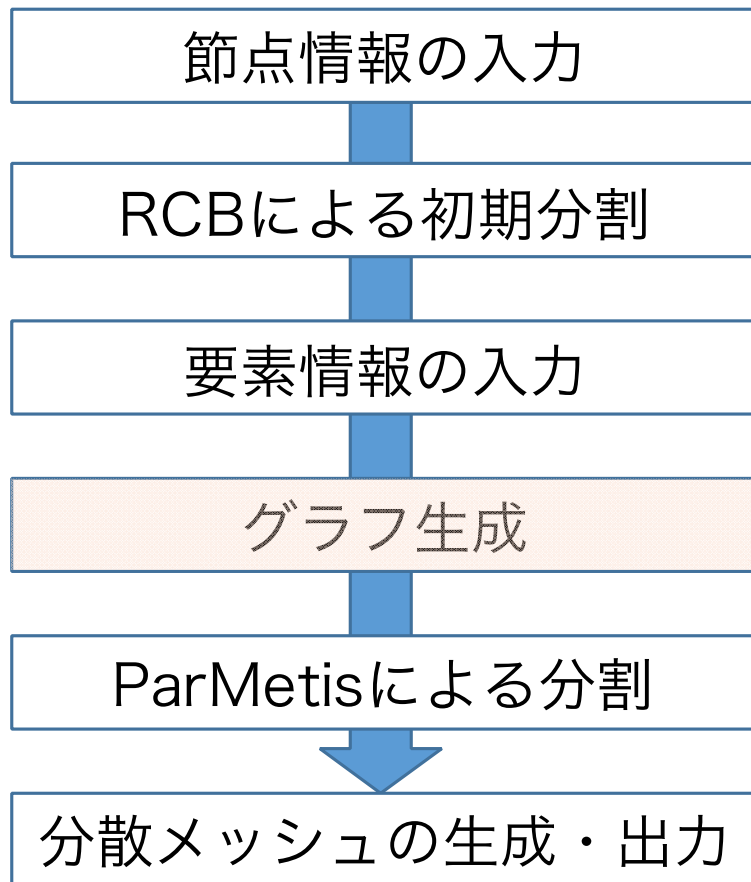
並列パーティショナの処理の流れ (3/6)



4. 各MPIプロセスで、
自分が担当する初期分割の
領域内にある節点
(部分グラフを構成するノード)
を含む要素コネクティビティ
データのみを読み込み、
メモリに格納する

Flowchart of the whole processing

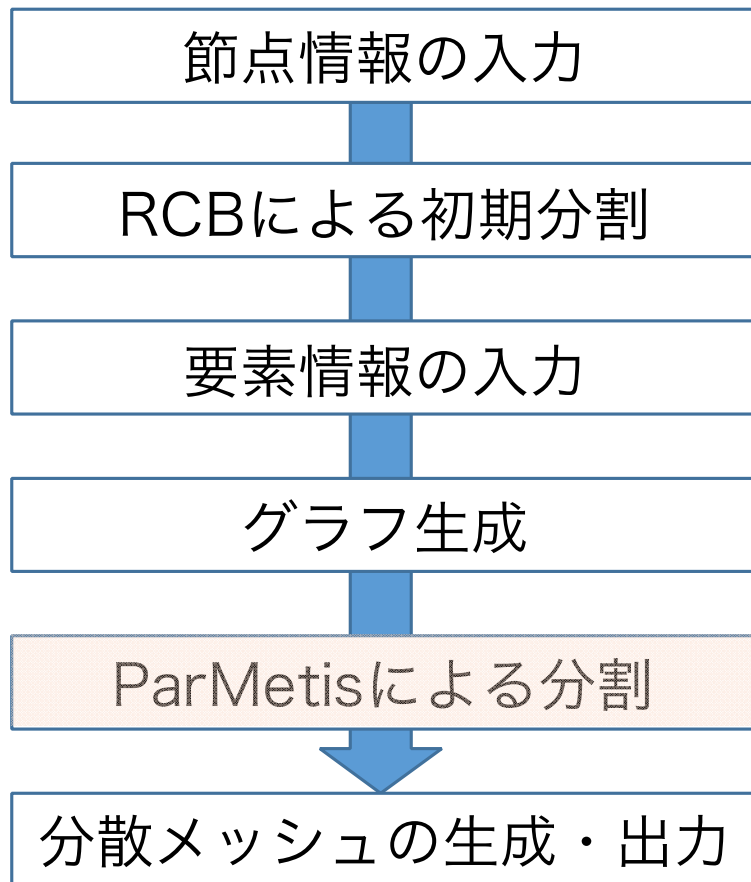
並列パーティショナの処理の流れ (4/6)



5. 各MPIプロセスで, 要素コネクティビティデータからグラフ情報を生成する
6. 各MPIプロセスで, 生成したグラフ情報をParMetisの呼び出しに適したデータ構造へ変換する

Flowchart of the whole processing

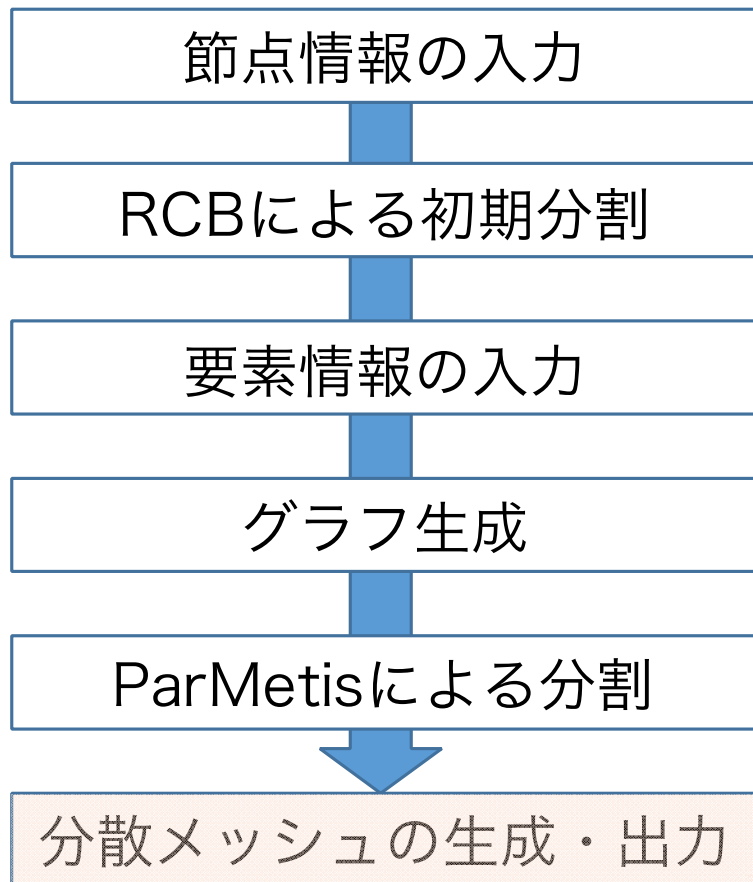
並列パーティショナの処理の流れ (5/6)



7. 各MPIプロセスで、変換したデータ構造をParMETISに渡し、領域分割を行う

Flowchart of the whole processing

並列パーティショナの処理の流れ (6/6)



Flowchart of the whole processing

8. 各MPIプロセスで,
全体メッシュを再度読み込み
ながら, 自分が担当する初期
分割に対する分散領域
メッシュを生成し, 出力する

新機能紹介: 並列パーティショナの開発

発表内容

1. 並列パーティショナの基本アルゴリズム
2. 今回開発された並列パーティショナのプログラム
3. 並列パーティショナの性能検証
4. 現状のまとめと今後の課題



並列パーティショナのプログラム

- 今回は, FrontISTRやHEC-MW本体のソースコードと独立した形で開発
 - ・・・ HEC-MWプログラム内のメッシュデータのメンバの一部を, 32ビット整数から64ビット整数に変更する必要がある, 大掛りな修正が必要となるから
- 将来的には, HEC-MWの中に取り込み, HEC-MWとの重複を排除した形で統合化
 - ・・・ 開発されたルーチンの中には, HEC-MWの中で類似の実装がなされているルーチンもあるから

入力可能なメッシュデータ

入力可能

- ・ カード

!HEADER
!NODE
!ELEMENT
!SECTION
!MATERIAL
!NGROUP
!EGROUP
!SGROUP
!AMPLITUDE
!END

入力未対応

- ・ カード

!EQUATION
!ZERO
!INITIAL CONDITION
!CONTACT PAIR

- ・ 全カードのINPUTパラメータ
- ・ !NODEのSYSTEMパラメータ
- ・ !ELEMENTで異なる自由度を持つ要素タイプの混在 (自由度が同じ要素タイプの混在は可能)
- ・ !MATERIALでの温度依存データの指定

新機能紹介: 並列パーティショナの開発

発表内容

1. 並列パーティショナの基本アルゴリズム
2. 今回開発された並列パーティショナのプログラム
3. 並列パーティショナの性能検証
4. 現状のまとめと今後の課題

性能評価に用いた例題 (1/2)

Table Analysis model

ID	Model name	Element type	The number of nodes	The number of elements	The number of DOFs
1	cube_h1_3M	8-node hexahedral element	1,030,301	1,000,000	3,090,901
2	pump	10-node tetrahedral element	36,728,129	26,289,770	110,184,387
3	cube_t2_1G	10-node tetrahedral element	344,472,101	257,250,000	1,033,416,303

性能評価に用いた例題 (2/2)

1. cube_h1_3M

- 従来のパーティショナでも領域分割できる規模
- 1辺を100分割した立方体モデル
- 比較的単純な形状
- 節点番号の空間的な飛びが少ないモデル

2. pump

- 従来のパーティショナでも領域分割できる規模
- 複数部品で構成されているような複雑な形状
- 節点番号の空間的な飛びが多いモデル

3. cube_t2_1G

- 従来のパーティショナでは領域分割できない規模
- 1辺を350分割した立方体モデル
- 比較的単純な形状
- 節点番号の空間的な飛びが多いモデル

性能評価 (節点番号順で初期分割する場合)

計算時間, プロセスあたり最大メモリ使用量, グラフ分割におけるエッジカット (64bit整数版ParMETISを使用)

ID	Model name	The number of sub-domains	#PE	Computation time [s]	Peak memory usage [GB/proc]	Edge cut
1	cube_h1_3M	8	1	17	0.41	3.7E+04
			2	14	0.28	4.1E+04
			4	12	0.16	4.1E+04
			8	12	0.11	4.3E+04
			16	12	0.08	4.4E+04
2	pump	128	1	792	18.08	2.2E+06
			2	675	29.53	2.2E+06
			4	600	27.23	2.2E+06
			8	617	26.12	2.2E+06
			16	646	23.81	2.2E+06
3	cube_t2_1G	8	1	9517	180.83	4.2E+07
			2	8435	294.56	4.2E+07
			4	Memory shortage		

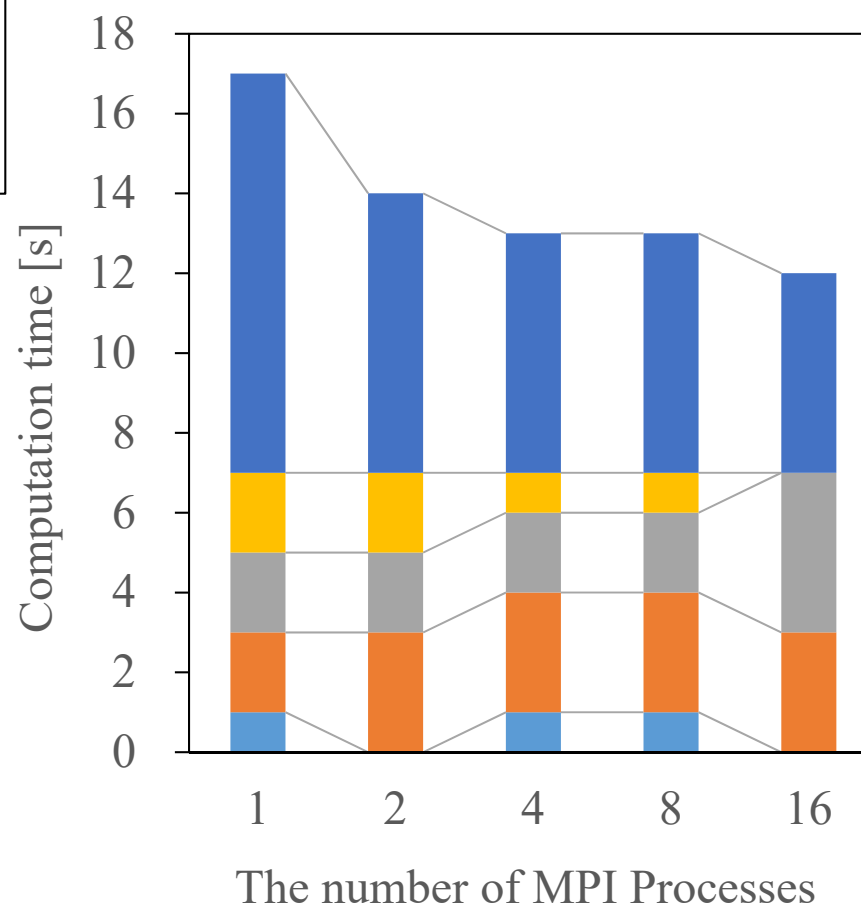
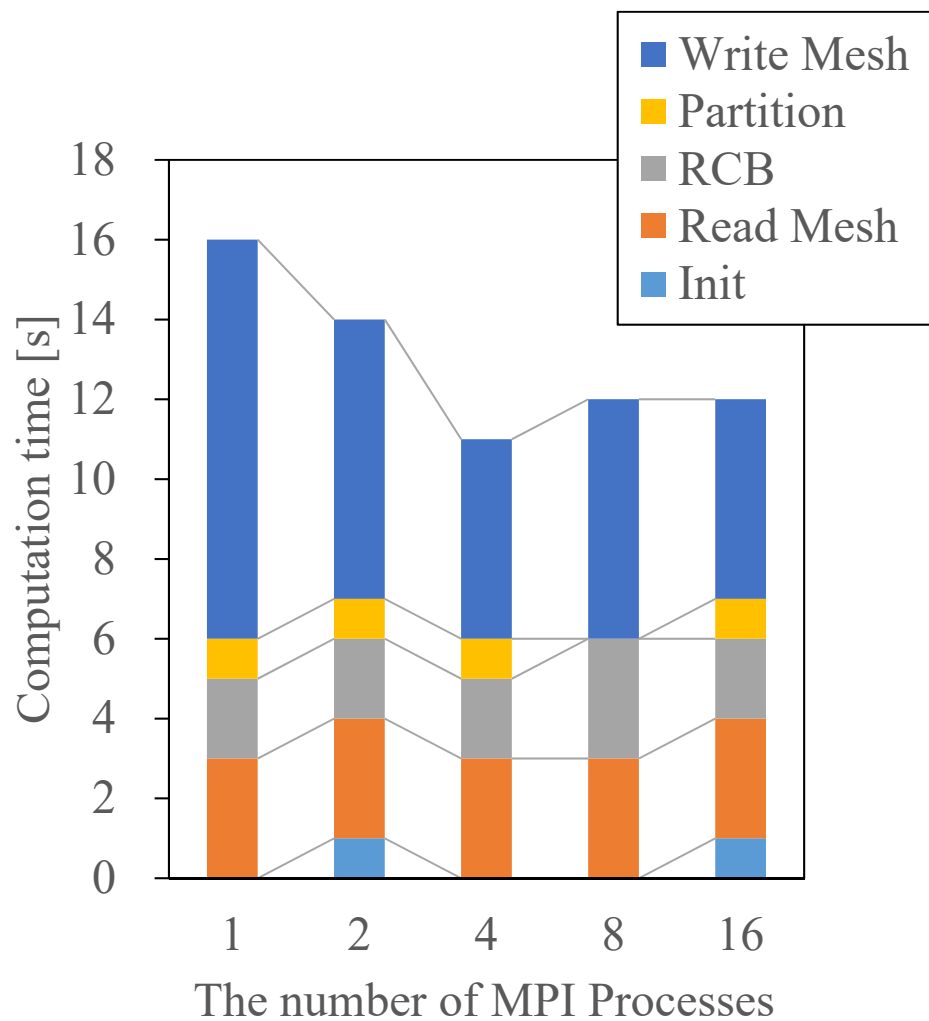
性能評価 (RCBで初期分割する場合)

計算時間, プロセスあたり最大メモリ使用量, グラフ分割におけるエッジカット (64bit整数版ParMETISを使用)

ID	Model name	The number of sub-domains	#PE	Computation time [s]	Peak memory usage [GB/proc]	Edge cut
1	cube_h1_3M	8	1	17	0.42	3.7E+04
			2	14	0.29	4.1E+04
			4	13	0.17	3.7E+04
			8	13	0.10	3.2E+04
			16	13	0.09	3.2E+04
2	pump	128	1	792	18.36	2.2E+06
			2	687	27.86	2.2E+06
			4	604	24.47	2.2E+06
			8	615	22.89	2.2E+06
			16	685	22.00	2.2E+06
3	cube_t2_1G	8	1	9517	180.83	4.2E+07
			2	7766	276.61	4.2E+07
			4	Memory shortage		

cube_h1_3Mの計算時間の詳細

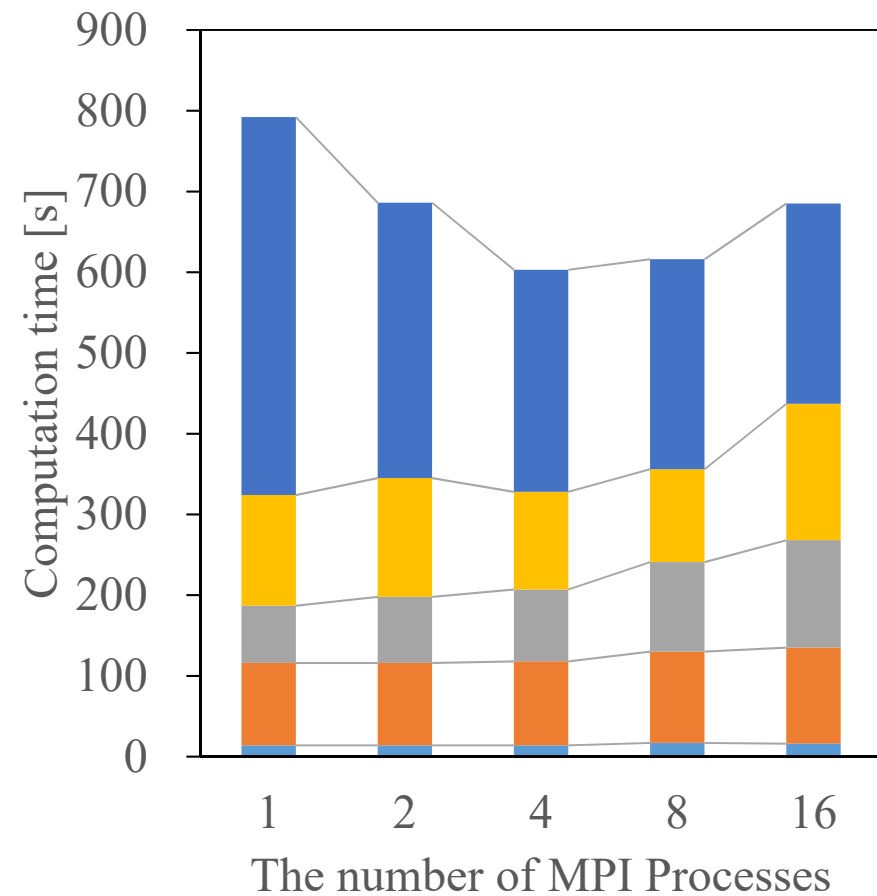
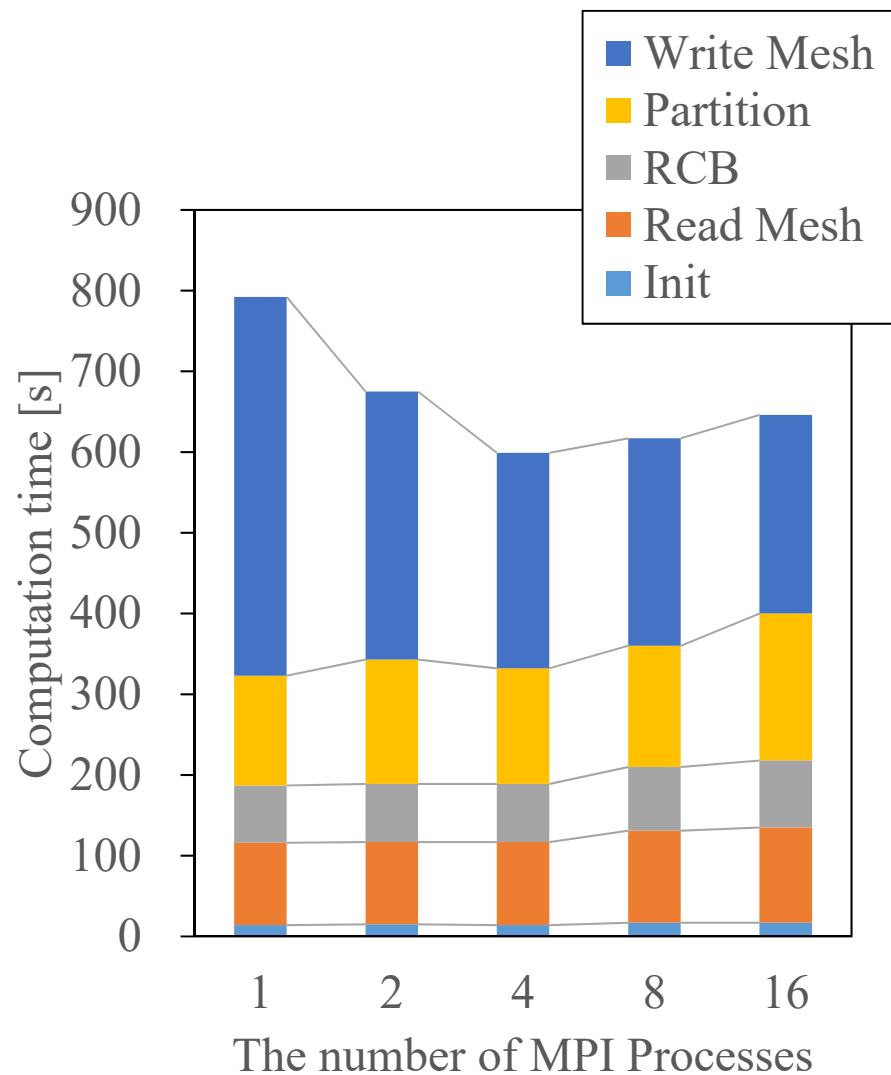
RCBプログラムには
並列化可能な部分がある



(a) 節点番号で初期分割を行う場合 (b) RCBで初期分割を行う場合

pumpの計算時間の詳細

RCBプログラムには
並列化可能な部分がある



(a) 節点番号で初期分割を行う場合 (b) RCBで初期分割を行う場合

性能評価 (従来のパーティショナ)

計算時間, プロセスあたり最大メモリ使用量, グラフ分割におけるエッジカット (32bit整数版METISを使用)

ID	Model name	The number of sub-domains	#PE	Computation time [s]	Peak memory usage [GB/proc]	Edge cut
1	cube_h1_3 M	8	1	12	0.42	3.7E+04
			2	11	0.42	3.7E+04
			4	10	0.42	3.7E+04
			8	10	0.43	3.7E+04
			16	13	0.43	3.7E+04
2	pump	128	1	602	16.13	2.2E+06
			2	533	16.13	2.2E+06
			4	487	16.13	2.2E+06
			8	517	16.13	2.2E+06
			16	545	16.13	2.2E+06

性能評価 (節点番号順で初期分割する場合)

計算時間, プロセスあたり最大メモリ使用量, グラフ分割におけるエッジカット (32bit整数版ParMETISを使用)

ID	Model name	The number of sub-domains	#PE	Computation time [s]	Peak memory usage [GB/proc]	Edge cut
1	cube_h1_3M	8	1	16	0.41	3.7E+04
			2	14	0.22	4.2E+04
			4	12	0.13	4.1E+04
			8	12	0.10	4.3E+04
			16	12	0.08	4.4E+04
2	pump	128	1	781	14.32	2.2E+06
			2	662	15.61	2.2E+06
			4	586	14.34	2.2E+06
			8	601	13.55	2.2E+06
			16	622	12.40	2.2E+06

性能評価 (RCBで初期分割する場合)

計算時間, プロセスあたり最大メモリ使用量, グラフ分割におけるエッジカット (32bit整数版ParMETISを使用)

ID	Model name	The number of sub-domains	#PE	Computation time [s]	Peak memory usage [GB/proc]	Edge cut
1	cube_h1_3M	8	1	17	0.42	3.7E+04
			2	14	0.23	4.2E+04
			4	12	0.14	3.6E+04
			8	13	0.10	3.2E+04
			16	13	0.08	3.2E+04
2	pump	128	1	782	14.60	2.2E+06
			2	669	16.50	2.2E+06
			4	589	13.84	2.2E+06
			8	600	12.56	2.2E+06
			16	658	11.87	2.2E+06

新機能紹介: 並列パーティショナの開発

発表内容

1. 並列パーティショナの基本アルゴリズム
2. 今回開発された並列パーティショナのプログラム
3. 並列パーティショナの性能検証
4. 現状のまとめと今後の課題

現状のまとめ

- FrontISTRのための並列パーティショナの**基本アルゴリズム**が開発された
- **メモリ消費量**の観点からは、節点番号に空間的な飛びがあるモデルを含め、大規模モデルの分割に有効であることが確認できた
- **計算時間**の観点からは、RCB分割の並列化や、分散メッシュ書き出し部における一時ファイルの書き出し／読み込み部分の見直しや、全体的な逐次計算性能の向上によって、高速化が可能であることがわかった

今後の課題

- 64bit整数が必要となる超大規模モデルに対する省メモリ化を行う
- RCBアルゴリズムのような未並列化部分を並列化する
- 全ての入力ファイル書式へ対応する
- HEC-MW本体のデータ形式の一部を64ビット化し、並列パーティショナのHEC-MW本体への取り込みと統合化を行う