［共同研究成果］

MPI による数値タービンの大規模並列計算手法の開発

笹尾泰洋[1]，山本悟[2]，三宅哲[2]，岡崎健志[2]
[1]帝京大学理工学部航空宇宙工学科笹尾研究室
[2]東北大学大学院情報科学研究科山本・古澤研究室

東北大学大学院情報科学研究科山本・古澤研究室にて開発された，蒸気タービン内部流動解析プログラムである「数値タービン」コードについて，NEC の協力を得て，移動境界問題に対応した MPI による大規模並列計算手法の開発を行った．13 流路-13 並列における並列化率は 99.2%，27 流路-27 並列における並列化率は 97.3%を達成した．並列計算の逐次計算に対する加速率は，13 並列において 11.3，27 並列において 16.0 となった．

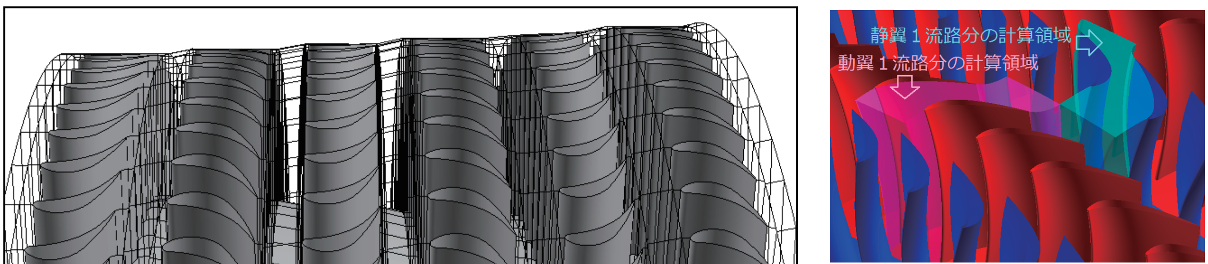
1. 緒言

蒸気タービンは，高温高圧の水蒸気から運動エネルギーを取り出すためのターボ機械の一種である．蒸気タービンの多くは，図 1 に示すような翼列と呼ばれる周方向と回転軸方向に配置された翼の集合によって構成されており，特に，周方向には周期性の強い流れ場が形成されることが知られている．そこで，ターボ機械に関する多くの解析的研究においては，翼列全体を解析対象とするのではなく，周期境界条件を活用し，周方向の一部分を対象とした流動解析が行われている．

しかし，現実の蒸気タービンにおいては，入口より供給される蒸気流量は周方向に対して不均一であり，加えて，蒸気タービンの下流に配置される排気室の形状や，失速といった様々な要因によって，周方向に著しく不均一な流れ場が形成されていることが知られている．そこで，蒸気タービンの更なる性能向上を目指し，全周を対象とした大規模流動解析による流れ場の把握に対する関心が高まっている．

蒸気タービンの大まかな流路形状は，周期的に配置された翼列によって決定される．よって，翼列間の 1 流路(ブロック)を 1 プロセッサ毎に割り当てて計算することを考えた場合，ほぼ完全な負荷分散が可能であり，MPI による並列計算においては極めて高い台数効果が期待できる．我々の研究グループにおける既往の研究では，周方向 1 流路，軸方向 6 流路から構成される計 6 流路の計算格子に周期境界条件を適用し，MPI による並列計算を行うことで，90%以上の並列化率が達成されている．しかしながら，周方向複数流路において並列計算を行う場合，動翼(図 1b の赤い翼列)は静翼(同じく青)に対して回転運動しているため，移動境界面におけるデータの送受信先は時々刻々と変化することになる．よって，ターボ機械の大規模並列計算において高い加速率と並列化率を達成するためには，移動境界におけるデータの送受信をいかに高速に行うかが課題となる．

そこで，本研究では NEC の協力を得て，MPI を用いた移動境界に適用できる数値タービンの並列化手法の開発に取り組んだ．移動境界面に属するブロック間においては，MPI_BCAST 関数を用いて，1 流路対多流路間でのデータ通信アルゴリズムを作成し，その性能と問題点について評価した．



a. 計算格子の俯瞰図　　b. 翼列と流路

図 1. 計算格子の全体像と 1 プロセッサが担当する計算領域の比較

2. 数値タービンの計算手法

2.1 基礎方程式

数値タービンにおいては，凝縮を伴う圧縮性流れの基礎方程式として，蒸気の相変化を考慮した蒸気の質量保存式，運動量保存式，エネルギー保存式，液滴の質量保存式，液滴の数密度保存式，乱流運動エネルギーおよびその比散逸率からなる式を解く．本研究で取り扱う気液二相流は液滴の質量分率が

十分に小さい均質流を仮定する($\beta <$ 0.1). 湿り蒸気の状態方程式および音速の式は石坂らにより定式化された式より算出する[1]. 凝縮による液滴の質量生成率は古典凝縮論に基づき，凝縮核生成と液滴の成長による質量増加の和で表される. 本研究ではさらに，液滴の成長を液滴の数密度を関数にした式で近似する[1]. 凝縮核生成率は Frenkel[2]の式より，液滴の成長率は，Gyarmathy[3]のモデルより算出した. 数値解法として，空間差分には Roe の流束差分離法[4]および 4 次精度 Compact MUSCL TVD スキーム[5]を用いた. 粘性項には 2 次精度中心差分を用い，乱流モデルには SST モデル[6]を用いた. 時間積分には LU-SGS 法[7]を用いた.

2.2 数値タービンのデータ構造と並列化における領域分割

ターボ機械の翼列は，回転機械の物理的な制約上，周方向に対して相似性の強い構造を採用している場合が殆どである. 数値タービンにおいては，図 2b に示すように，翼と翼に囲まれた空間(流路)を1つの計算領域(ブロック)と定義し，計算対象となる翼枚数に助走区間を加えた数だけブロックを確保する. 表 1 は逐次計算と並列計算における配列構造とループ構造の概要である. 逐次計算においては，1 つの 3 次元配列(I,J,K)を I 方向にブロック数と等しい数だけ分割し，各ブロックの計算領域として割り当てている. ループ構造からも明らかなように，基礎方程式はブロック毎に閉じており，隣接するブロックとのデータのやり取りは陽的に処理される. よって，並列計算を行う場合においても，計算領域をブロック単位で分割し，各 RANK において個別に管理・計算を行う限りにおいては，逐次計算と並列計算の間に解の不一致は生じない. なお，並列計算におけるデータ通信は，境界条件に関するサブルーチンが CALL される段階で行う.

過去の数値タービンの並列化に関する研究においては，1 ブロックを複数領域に分割し，OpenMP を用いた並列計算した例について報告している. 山本らは，領域分割計算においても陰解法の依存関係を考慮するために，パイプライン法を用いた LU-SGS 法の並列化手法を提案した[8][9]. しかし，本研究においては，移動境界における並列化アルゴリズムの複雑化を避けるために，領域分割計算は行わないものとした.

表 1. 逐次計算と並列計算における配列構造の違い


```
DO L=1, N
  DO I=IS(L), IT(L)
    DO J=JS, JT
      DO K=KS, KT
        WW(I, J, K) = W(I, J, K)
CONTINUE
```

a. 逐次計算の場合


```
LRANKO=MYRANK
DO I=IS(LRANKO), IT(LRANKO)
  DO J=JS, JT
    DO K=KS, KT
      WW(I, J, K) = W(I, J, K)
CONTINUE
```

b. MPI による並列計算の場合

2.3 数値タービンの境界条件

表 2 は数値タービンにおいて考慮されている，主な境界条件の一覧である. ターボ機械の形状は様々であり，また考慮する現象や機械的構造も大きく異なるため，表に挙げる境界条件の他にも適宜追加される境界条件が存在するが，ここでは割愛する.

これらの代表的な5つの境界条件は，他ブロックとのデータの送受信の有無によって，2 つに大別できる. 先ず，他ブロックとのデータ交換を必要としない種類の境界条件について述べる. 流入-流出境界条件は，蒸気タービンの入口と出口に適用される境界条件である. 圧縮性流れの数値解析では，計算領域の入口において全温と全圧を，出口において静圧を境界条件として固定する手法が一般的に用いられており，数値タービンにおいても同様の境界条件が適用されている. 図 2a は，静翼面を青，動翼面を赤，翼端壁

を緑，シーリング部前後のキャビティ入口を黄色と水色に着色した図である．壁面境界条件は，図 2a の赤，青および緑に着色された境界面に適用され，個体壁面上に展開された計算格子が保有すべき値は，計算条件に基づいた代数的な計算や，ブロック内部の計算格子からの外挿によって与える．壁面境界条件は，各ブロック内において完結しているため，他ブロックとのデータの送受信は必要としない．一方，蒸気タービンの内部においては，蒸気の一部は翼列を通過せず，翼列と構造体の間に施されたシーリングを通過する．この様な流れは一般に漏れ流れ(Leak Flow)と呼ばれており，図 2a における，吸込境界(水色)や吐出境界(黄色)において近似的に考慮される．シーリングを通過する蒸気は翼に対して仕事をしないため，漏れ流れを計算に取り入れることは，蒸気タービンの性能を正しく評価する上で重要である．上流のブロックにおいて発生した漏れ流れは，シーリングを通過し下流のブロックへと流入するため，本来であれば，漏れに関与するブロック間において流量やエンタルピ等に関するデータの送受信が必要となる．ただし，本研究においては，漏れ量を主流流量の 1.0%と定義し，代数的に漏れ量を算出することで，本境界条件をデータの送受信の対象外とした．漏れ流れを考慮しない解析においては，漏れ流れ境界面には壁面境界条件やすべり境界条件が適用されることが多い．

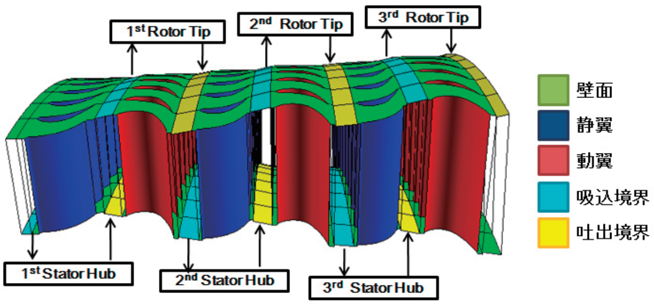
次に，他ブロックとのデータの交換を必要とする境界条件について述べる．図 2b は，計算格子番号の定義の概略図であり，数値タービンにおいては，軸方向格子番号を I，周方向格子番号を J，半径方向格子番号を K と定義している．周期境界条件は，図 2c, 2d に示すように，周方向に隣り合うブロックとの間に適用される境界条件であり，周期境界面から流出した蒸気は，隣接する周期境界面に流入するというのが基本原則である．数値タービンにおいては，周期境界面上の計算格子は，隣接するブロックの計算格子と空間的に重なり合うように作成されている．よって，それぞれのブロックの計算を担当しているプロセッサを調べ，MPI_SENDRECV を用いてシフト通信を行うことで，周期境界条件に必要なデータの送受信は完了する．周期境界条件は JK 面に適用される境界条件である．

移動境界条件は，流れ方向に隣接するブロック間に適用される境界条件であり，図 2e において着色された円盤状の領域が適用範囲となる．動翼は常に旋回しているため，隣接するブロックは時々刻々と変化する点が，周期境界条件と大きく異なる点である．また，静翼列と動翼列の翼枚数比(周方向に並んだ翼の数の比)は，蒸気タービン毎に固有の値を持ち，さらに，静動翼列の各段落によっても変化する．すなわち，データ送受信の対象となるブロックを一意に決定できない点が，並列化を行う上で課題となる．移動境界条件においては，移動境界面上に存在する格子点の情報を配列に列挙し，対面する移動境界面上の格子点に対して相対位置を評価した上で，情報の交換が行われる．しかしながら，移動境界面上の全ての格子点について検索を行うと演算量が膨大となる．そこで，移動境界面上の計算格子は，回転軸に対して同心円状に配置され，K 方向の格子番号が同じであれば，回転軸からの距離も等しくなるように設計されている．よって，周方向の位置に関してのみ検索を行えば，情報の交換対象となる隣接格子点を容易に知ることができる．

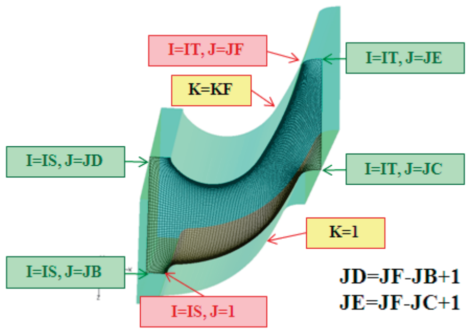
周期境界に適用するデータの送受方法に関しては次節において詳細に述べる．

表 2. 数値タービンにおいて考慮される主な境界条件

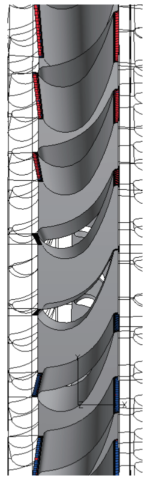
他ブロックとデータを交換しない境界条件	**1. 流入-流出境界条件(JK 面)**：蒸気タービンに流入する気体の状態や，入口-出口における圧力比などを設定する．
	2. 壁面境界条件(IK 面)：翼面および内周壁，外周壁など壁面上における流体の状態を決定する．
	3. 漏れ流れ境界条件(IJ 面)：翼端漏れ流れを代数的に近似計算する．
他ブロックとデータを交換する境界条件	**4. 周期境界条件(JK 面)**：周方向に隣り合うブロック間に適用する境界条件．周方向のブロックの相対位置は変化しないため，常に同じブロックがデータの送受信対象となる．
	5. 移動境界条件(JK 面)：軸方向(流れ方向)に隣り合うブロック間に適用する境界条件．動翼列は静翼列に対して相対運動しているため，周方向複数流路を考慮する場合には，データの送受信対象が時々刻々と変化する．



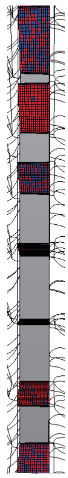
a. 壁面に適用される境界条件



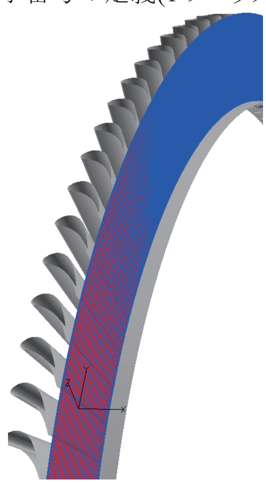
b. 計算格子番号の定義(1 ブロック)



c. 周期境界条件(翼列間)



d. 周期境界条件(助走区間)



e. 移動境界条件

図 2. 数値タービンにおいて考慮される主な境界条件の概略図

2.4 移動境界におけるデータの送受信について

動翼列は周方向に絶えず移動し，翼枚数比は蒸気タービンや段落毎に固有の値を持つため，移動境界においてデータの送受信対象となるブロックは一意に決定することはできない．そこで，図 3 に示すように，周方向に並ぶブロック(静翼列，動翼列および助走区間)をコミュニケータにおける1グループと定義し，移動境界処理を行うプロセッサ間の通信を限定することを考える．

例えば，上流側のブロックから下流側のブロックに対しデータ通信を行う場合，上流側の 1 プロセスと下流側の全プロセスとを 1 つのコミュニケータグループと定義する(図 3 右図)．次に上流側の 1 プロセスから下流側の他プロセスへ MPI_BCAST(Broad Cast)通信を行う．これによって，任意のプロセスが保有する移動境界面上の情報は，移動境界面を共有する対面側の全プロセスに対して送信される．一方，下流側から上流側へデータ通信を行う場合も，下流側の 1 プロセスから上流側の全プロセスに対し MPI_BCAST を用いた通信を行う．よって，上流側の流路数を M，下流側の流路数を N とした場合，1 つの移動境界面における MPI_BCAST は，M×N 回だけ CALL されることになる．

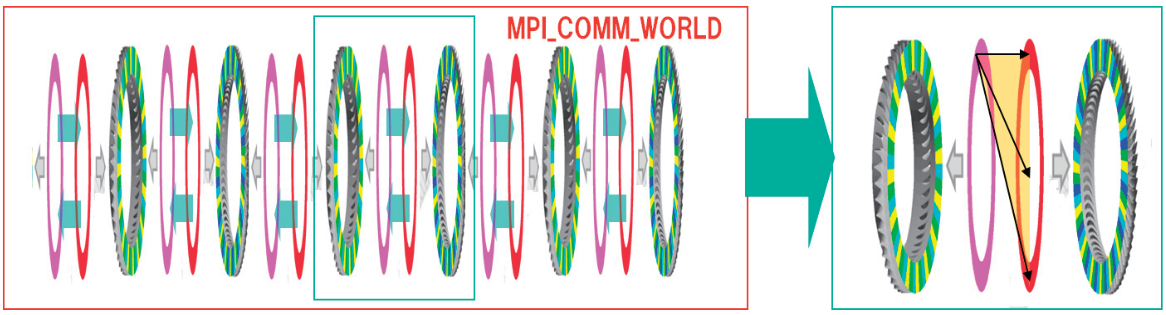


図 3. 移動境界面におけるコミュニケータのグループ分けの概略図

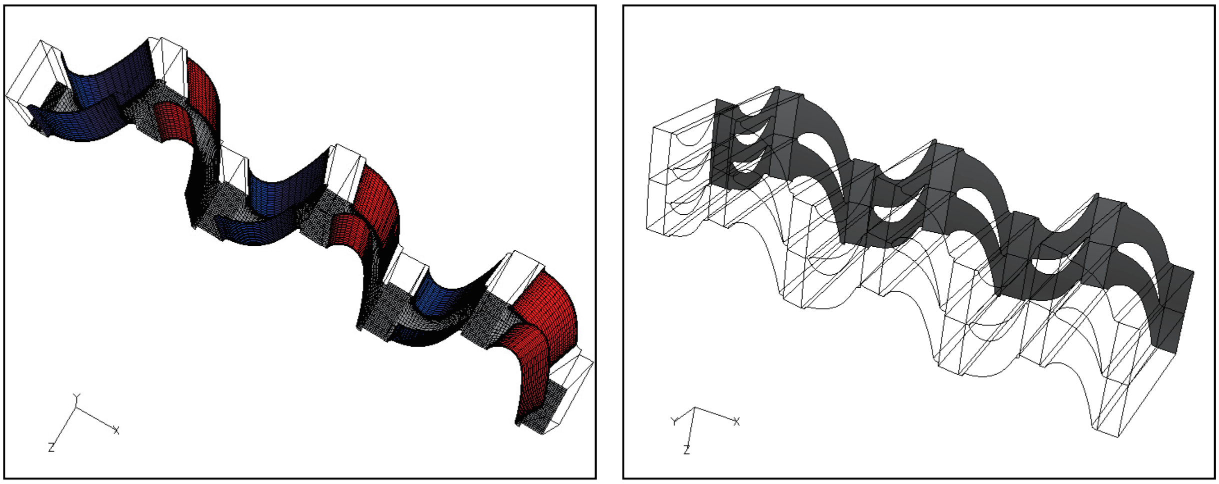
3. 計算結果

数値タービンの MPI による並列化は，逐次計算の場合とバイナリレベルで計算結果が一致することを確認しながら，段階的に行った．図 4 は本研究に用いた計算格子の俯瞰図である．図 4a は，静動翼列 3 段 6 流路と助走区間 7 流路の計 13 ブロックから構成される計算格子であり，図 4b は，初段静翼列のみ周方向 3 ブロック，他流路は周方向 2 ブロックの計 27 ブロックから構成される計算格子である．翼列間流路における格子点数が 31x31x31 点であるのに対し，助走区間の格子点数は 16x31x31 点であるため，並列計算を行う場合には，RANK 間にロード・インバランスが発生する．本共同研究における計算は，全て東北大学サイバーサイエンスセンターSX-9 上にて行った．13 ブロックを対象とした計算においては 1 node (1, 4, 7, 13 CPU)を用い，27 ブロックにおいては，1 node (1, 8, 16CPU)および 2 node (27 CPU)を用いて計算を行った．

表 3 は並列数に対する加速率の比較である．加速率は，逐次計算に必要とされた計算時間を，並列計算に必要とされた計算時間で除した値である．13 流路-13 並列の場合における加速率は 11.9，並列化率は 99.2%に達したのに対し，27 流路-27 並列の場合においては，加速率は 16.0，並列化率は 97.3%となった．また，13 流路，27 流路ともに，1 プロセッサが複数流路の計算を行った場合には，加速率の著しい悪化が見られた．

表 4 は 27 流路に対して並列計算を行った場合の，各サブルーチンのロードバランスと加速率の比較である．オレンジ色に着色されたサブルーチンは，MPI によるデータ通信を含むサブルーチンである．Explicit，Implicit2，Result，Viscous といった，計算負荷の高いサブルーチンについては，プロセッサ数に応じた高い加速率が得られたものの，データ通信に関するサブルーチンは，ほとんど加速率の向上が得られていないことがわかる．これは，移動境界における BCAST 通信に多くの時間を要したためである．

図 5 は 27 流路について並列計算した場合における，通信時間が計算時間に占める割合の比較である．全プロセッサが 2 ブロック以上の計算を行う 8 並列の場合と，各プロセッサが 1 ブロックずつ計算を行う 27 並列においては，データ通信が全計算時間に占める割合は 10%に満たないことがわかる．一方，プロセッサ間のロード・インバランスが顕著に表れる 16 並列においては，データ通信に要した時間が IDLE 時間(データ通信待ち時間)も含めて全体の 6 割以上にも及び，データ通信が処理速度のボトルネックとなっていることがわかる．以上より，BCAST 通信を頻繁に行う本並列化手法においては，プロセッサ間のロードバランスが可能な限り均一になるように，計算領域を分配することが，極めて重要であるといえる．



a. 単流路 13 ブロック　　b. 複数流路 27 ブロック

図 4. 計算格子

表 3. 並列数による加速率の比較

	逐次計算(1 並列)	4 並列	7 並列	13 並列
13 流路	1.0	3.2	6.3	11.9
	逐次計算(1 並列)	8 並列	16 並列	27 並列
27 流路	1.0	5.8	6.6	16.0

表 4. ランク 0 が要した各サブルーチンの計算時間(CPU-time [sec])と加速率の比較

subroutine	並列数 (27 流路)							
	1		8		16		27	
	[sec]	加速率	[sec]	加速率	[sec]	加速率	[sec]	加速率
MAIN_LOOP	184.37	(1.0)	31.92	(5.8)	27.76	(6.6)	11.54	(16.0)
explicit	94.29	(1.0)	15.30	(6.2)	6.76	(13.9)	6.67	(14.1)
implicit2	49.62	(1.0)	5.27	(9.4)	1.56	(31.7)	1.54	(32.2)
result	13.56	(1.0)	1.52	(8.9)	0.51	(26.4)	0.51	(26.7)
viscous	9.27	(1.0)	1.06	(8.7)	0.35	(26.6)	0.35	(26.6)
tbmdl2	8.32	(1.0)	0.93	(9.0)	0.31	(26.9)	0.31	(26.8)
mpisub_read_grid	8.04	(1.0)	5.89	(1.4)	7.35	(1.1)	5.84	(1.4)
mpisub_comm_bundperi	0.00	(0.0)	0.88	(0.0)	16.21	(0.0)	0.03	(0.0)
bundslid	7.99	(1.0)	0.64	(12.5)	0.21	(37.7)	0.21	(38.0)
main	0.59	(1.0)	0.55	(1.1)	0.52	(1.1)	0.54	(1.1)
mpisub_output_restart	0.33	(1.0)	0.34	(1.0)	0.31	(1.1)	0.29	(1.1)
bundwall	0.19	(1.0)	0.02	(8.1)	0.01	(20.7)	0.01	(20.7)
bundperi	0.12	(1.0)	0.03	(4.6)	0.01	(9.7)	0.01	(10.5)
init	0.06	(1.0)	0.01	(7.4)	0.00	(14.8)	0.00	(14.8)
metric	0.04	(1.0)	0.01	(8.0)	0.00	(20.0)	0.00	(20.0)
initmp	0.04	(1.0)	0.03	(1.2)	0.01	(2.6)	0.01	(2.6)
boundlhs	0.04	(1.0)	0.00	(8.8)	0.00	(17.5)	0.00	(17.5)
boundrhs	0.04	(1.0)	0.00	(8.8)	0.00	(17.5)	0.00	(17.5)
mpisub_read_figure	0.03	(1.0)	0.03	(1.0)	0.02	(1.6)	0.02	(1.7)
bundstrm	0.02	(1.0)	0.01	(1.4)	0.01	(2.9)	0.01	(2.9)
mpisub_final	0.02	(1.0)	0.14	(0.1)	0.24	(0.1)	0.39	(0.1)
mpisub_init	0.02	(1.0)	0.10	(0.2)	0.15	(0.1)	1.11	(0.0)
mpisub_comm_bundslid_theta	0.01	(1.0)	5.08	(0.0)	0.64	(0.0)	0.57	(0.0)
outigs	0.01	(1.0)	0.00	(6.5)	0.00	(13.0)	0.00	(13.0)
mpisub_read_init	0.01	(1.0)	0.01	(0.9)	0.01	(0.9)	0.01	(0.9)
mpisub.allocate_distributed_arrays	0.01	(1.0)	0.01	(1.4)	0.01	(0.8)	0.01	(1.1)
mpisub_comm_bundslid	0.01	(1.0)	0.01	(0.8)	0.01	(1.1)	0.01	(0.8)
bundmain	0.01	(1.0)	0.01	(1.0)	0.01	(1.0)	0.01	(1.0)

※オレンジ:データ通信を含むサブルーチン, ピンク:ファイル入出力を含むサブルーチン

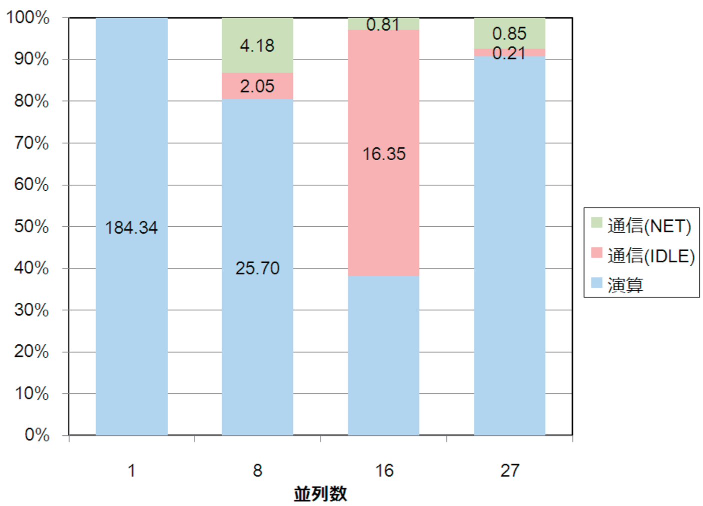


図 5. 各並列数において通信時間が計算時間に占める割合(27 流路)

4. 結言

MPI を用いて数値タービンの並列化を行い，加速率と並列化率を評価した．移動境界問題の並列化に関しては，移動境界面に属する1つの RANK から，その他の RANK に対する BCAST を用いて実装した．13 流路-13 並列の場合における加速率は 11.9，並列化率は 99.2%，27 流路-27 並列の場合においては，加速率は 16.0，並列化率は 97.3%を達成した．

謝辞

本研究における数値タービンの並列化は，NEC および東北大学サイバーサイエンスセンターの全面的な技術協力の元，行いました．ご協力頂きました団体ならびに皆様に対し，厚く御礼申し上げます．

参考文献

[1] Ishizaka, K., Ikohagi, T. and Daiguji, H., A High-Resolution Numerical Method for Transonic Non-Equilibrium Condensation Flow through a Steam Turbine Cascade, Proc. of the 6th ISCFD, 1, 479-484, 1995.

[2] Frenkel, J., Kinetic Theory of Liquids, Dover, 1955.

[3] Gyarmathy, G., Zur Wachstumsgeschwindigkeit kleiner Flussigkeitstropfen in einer ubersattigten Atmosphare, Zeitschrift fur angewandte Mathematik und Physik, 14-3, 280-293, 1963.

[4] Roe, P.L., Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, Journal of Computational Physics, 43, 357-372, 1981.

[5] Yamamoto, S. and Daiguji H., Higher-Order-Accurate Upwind Schemes for Solving the Compressible Euler and Navier-Stokes Equations, Computers and Fluids, 22-2/3, 259-270, 1993.

[6] Menter, F.R., Two-equation Eddy-viscosity Turbulence Models for Engineering Applications, AIAA Journal, 32-8, 1598-1605, 1994.

[7] Yoon, S. and Jameson, A. Lower-upper Symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations, AIAA Journal, 26, 1025-1026, 1988.

[8] Satoru Yamamoto, Yasuhiro Sasao, Shoichiro Sato and Kentaro Sano, Parallel-Implicit Computation of Three-dimensional Multistage Stator-Rotor Cascade Flows with Condensation, 18th AIAA CFD Conference-Miami, AIAA Paper 2007-4460, June, 2007.

[9] Satoru Yamamoto, Yasuhiro Sasao, Kentaro Sano, Hiroshi Satsuki, Kouichi Ishizaka and Hiroharu Ooyama, Parallel Computation of Condensate Flows through 2-D and 3-D Multistage Turbine Cascades, Int. Gas Turbine Congress 2007(IGTC'07)-Tokyo, Dec.2007, CD-ROM..