様式 C-19

科学研究費補助金研究成果報告書

平成 21 年　4 月　1 日現在

研究種目：基盤研究（C）
研究期間：2006～2008
課題番号：18500044
研究課題名（和文）　低消費電力コンパクトクラスタに関する研究

研究課題名（英文）　Development of Low Power Compact Cluster

研究代表者
早川　潔
大阪府立工業高等専門学校・総合工学システム学科・電子情報コース・准教授
研究者番号：20325575

研究成果の概要：低消費電力コンパクトクラスタのプロトタイプとして，9 台の PentiumM クラスタの構築し，それを現存のクラスタ 27 台と結合した低消費電力ヘテロクラスタシステムを構築した．また，そのクラスタ上での Intra および Inter　Chassis　Network を実装し，その上でのコミュニケーションフレームワークを提案した．性能評価として，分子軌道計算を行い，並列化効率，消費電力などを計測し，低消費電力ヘテロクラスタシステムの有効性を検証した．

交付額

（金額単位：円）

	直接経費	間接経費	合　計
2006 年度	**2200,000**	**0**	**2200,000**
2007 年度	**800,000**	**240,000**	**1040,000**
2008 年度	**400,000**	**120,000**	**520,000**
年度			
年度			
総　計	**3400,000**	**360,000**	**3760,000**

研究分野：総合領域
科研費の分科・細目：情報学・計算機システム・ネットワーク
キーワード：並列計算，ネットワークインターフェース

１．研究開始当初の背景

（１）CPU の低消費電力化にともない，クラスタの低消費電力化およびコンパクト化が重要になりつつある．汎用部品（パソコンのマザーボードやインテルの CPU など）で構成された PC クラスタシステムは,そのシステム構築が比較的安価でかつ容易なため,数十～数百台規模のシステムに膨らんできている．また,市販マイクロプロセッサの性能が急激に向上しており,そのプロセッサを使用する PC クラスタシステムは，より高速な並列処理を可能にしている．

（２）しかし,近年，インテルなどの CPU ベンダーは，CPU の単体性能を上げる技術開発よりも CPU の消費電力を抑えるような技術開発に重点をおいている．それは，性能とともに消費電力が増大し，ファンなどのプロセッサの熱を逃がす装置が巨大化し，それによりコンパクトな筐体設計が難しくなるということも原因の１つとして挙げられる．

（３）クラスタシステムでは，CPU が複数集まっているので，熱放出を考えた筐体設計はより難しくなる．１つの CPU がほんの数%消費電力が上がっても，クラスタシステム全体では数十～数百倍に跳ね上がる．消費電力増大によるファンなどの熱放出装置の巨大化

によって，筐体はより巨大になっていく．PC クラスタシステムはある程度小規模な企業・研究グループで使用されることが多いので，できるだけコンパクトで低消費電力なクラスタシステムが望まれる．
（4）クラスタシステムを長期間運用するときの問題として故障部品の供給停止が挙げられる．一般市場における CPU のライフサイクルは 2〜3 年であり，故障した時期が遅れるほど，市場で入手できにくくなる．入手できたとしても，性能の高い CPU よりも高価になっている場合が多く，その場合，性能の高い CPU に買い換えたほうが得策である．また，新たにノード台数を増やす場合にも，性能の高い CPU を増設するほうが安価になる場合が多い．
（5）そのような CPU 交換またはノード増設方法を行った場合に問題となるのが，ノードの性能にアンバランスが生じ，ノードの利用効率が悪化することである．この問題に対処するため，様々な負荷分散方式が提案されている．

２．研究の目的

（1）低消費電力ヘテロコンパクトクラスタを構築し，その上で電力消費量などの性能評価を行う．
（2）コンパクト実装を活かしたネットワークを構築し，その上でのコミュニケーションフレームワークを提案する．
（3）低消費電力クラスタ上の実行環境フレームワークを提案し，そのフレームワークを利用したアプリケーション実行を行う．
（4）ヘテロクラスタに考慮した負荷分散方式を提案し，実装する．
（5）低消費電力ヘテロクラスタを電力消費量．並列化効率という観点で性能評価する．

３．研究の方法

（1）低消費電力クラスタとして，EMDC（Embedded Middle Density Cluster）（図１参照）を構築した．EMDC システムでは，FA などで使用する組み込み機器のマザーボードを採用し，そのマザーボードにインテルが供給している Embedded CPU(長期間入手可能な CPU)を搭載することで長期間運用を可能にする．組み込み機器のマザーボードにも Pentium M などのノート PC で使用される低消費電力型 CPU が搭載できる製品もでてきており，より低消費電力なクラスタが実現可能である．
（2）本システムでは，PentiumM(2.0GHz)搭載のノードが 9 台，PentiumIII 搭載のノード 27 台，およびホストコンピュータで構成されている．PentiumIII 搭載のノードでは，3 ノードのうち 1 ノードが動作周波数 600MHz の CPU で残りの 2 ノードが 700MHz の CPU である．PentiumIII や PentiumM などの比較的低動作周波数ではあるが低消費電力である CPU を利用して，コンパクトなクラスタを目指している．
（3）ネットワーク構成は，Intra-Chassis Network および Inter-Chassis Network の２つのネットワークで構成されている．Intra-Chassis Network は，ギガビットイーサーボードを使用して構築されている．PentiumIII ノードの Inter-Chassis Network は，100Base-TX の Ethernet で，PentiumM ノードの Inter-Chassis Network は，Gigabit Ethernet で構築されている．

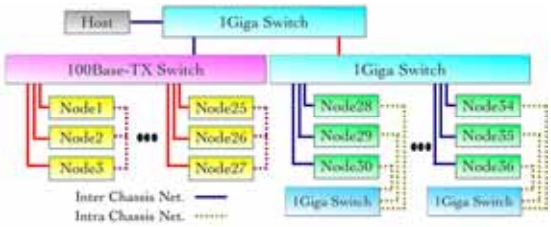

図 1. EMDC クラスタ

図 2. EMDC クラスタの外観

（4）EMDC システムにおけるネットワークフレームワークとして，仮想 Tree ネットワークを考える（図３参照）．Intra および Inter-Chassis Network をうまく利用することにより，仮想的に Tree ネットワークを構成する．シャーシ内の通信を Intra-Chassis Network で密に行うことができるので，1 シャーシで 3 つのネットワークポート（Inter-Chassis Network のポート）があると仮想的に考え，それらのポートを使って，Binary Tree を構成する．

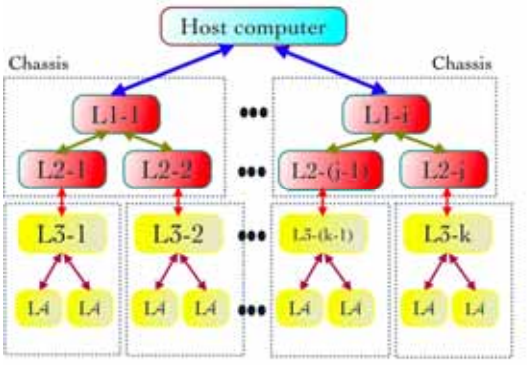

図 3　仮想 Tree ネットワーク

（5）アプリケーションフレームワークの例を図４に示す．アプリケーションフレームワークは，Appli_Control 処理部，Communication Library 処理部，Communication Bridge 処理部，および FMC 処理部で構成される．Appli_Control 処理部では，計算ノードの計算データを生成，各計算ノードに分配，および計算ノードが処理した計算結果の収集処理などアプリケーション全体のコントロールが行われる．Communication Library 処理部は後述する通信ライブラリ（H_comm および L_comm) の処理を行う．Communication Bridge 処理部は，通信のリンクレイヤ層が異なる場合に，その間のインターフェースの役割を行い，コレクティブ通信を制御する役割も行う．FMC 処理部は，Appli_Control 処理部から送られてきたデータを使用して計算を行う．実行モデルとしては，master-worker 型の実行モデルとする．Appli_Control 処理部が Master になり，FMC 処理部にコマンドを送ることにより，Worker として動作させる．

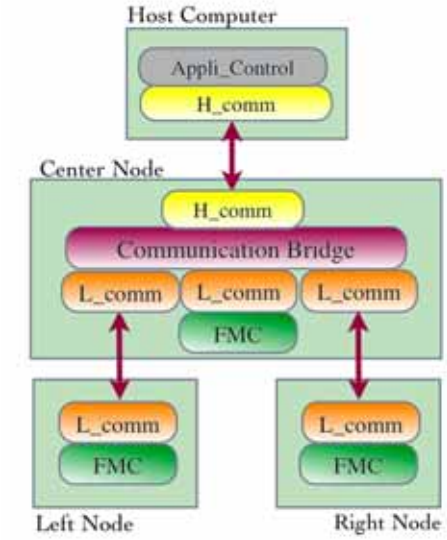

図 4　アプリケーションフレームワーク

（6）通信フレームワークの構成例を図 5 に示す．

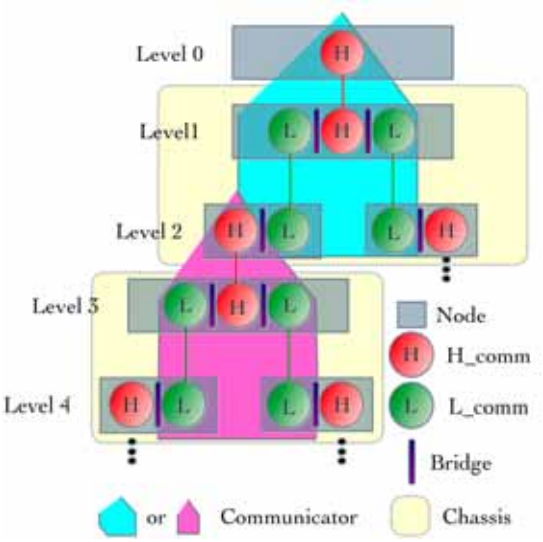

図 5　通信フレームワーク

3 つの連続するレベルに属しているノード集合が，通信を形成する 1 つの集合（MPI でいう Communicator)として構成される．それらの集合をレベル 3n からレベル 3n+2(n=0,2,4...:2 の倍数）に属しているノード集合，レベル 3n+2 からレベル 3n+4 に属しているノード集合，およびレベル 3n+4 からレベル 3(n+1)に属しているノード集合というように，ある集合の最下位レベルのノードがその下のレベルのノード集合の最上位レベルのノードと重なるように集合を構成する．重なったレベルに Communication Bridge を挿入し，上位のレベルのノード集合と下位レベルのノード集合を結合させる．このことにより，より深いレベルのノード間通信を可能にする．例えば，図 5 において，Level 0 から Level 2 までの 3 つのノードの集まりを 1 つのノード集合，Level 2 から Level 4 までのノードの集まりを 1 つのノード集合として考え，Level 2 の L_comm と H_comm の間に Communication Bridge を挿入することにより，Level 0 から Level 4 までの通信を可能にする．

（7）通信ライブラリのレイヤ構成を図６に示す．通信ライブラリは，Intra Chassis Network 側のレイヤおよび Inter Chassis Network のレイヤ，それを結ぶコレクティブ通信レイヤで構成される．Intra および Inter Chassis Network の Phy.および Link レイヤは，Ethernet とし，それに対応する H_comm および L_comm を実装した．H_comm および L_comm では，send/recieve などの atomic な命令を実装し，それを利用して Communication Bridge レイヤが通信を行う．H_comm および L_comm は，基本的に，1 つ上（または 1 つ下）のレベルへのノード間

通信のみを実現する．1つより上（または下）のレベルへのノード間通信は，Communication Bridge レイヤを介して行われる．コレクティブ通信も Communication Bridge レイヤが担当する．

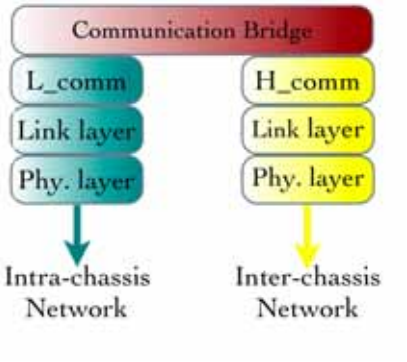

図 6　通信ライブラリの構成図

４．研究成果

（１）分子軌道計算ソフト GAMESS を EMDC 用に並列化したプログラムを使用し．積算電力量および並列化効率を測定した．本プログラムを 1 つのネットワークの場合と 2 つのネットワークにおける並列化効率を比較した．使用した分子は，C 端末を OH キャップしたグリシンの 5 量体である．

（２）EMDC システムを PentiumIII ノードと PentiumM ノードに分け，それぞれで分子軌道計算を実行させた．実行結果を図 7 に示す．

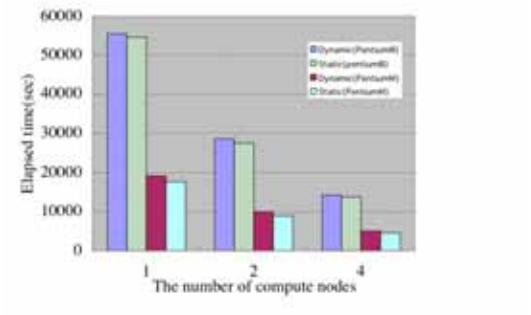

図 7　PentiumIII および PentiumM の性能比較

図 7 において，（Static）は静的負荷分散における分子軌道計算全体の実行時間であり，（dynamic）は動的負荷分散における分子軌道計算全体の実行時間である．静的負荷分散とは，実行前に各計算ノードの計算量を決め，実行させる方式であり，動的負荷分散は，全体の計算を細かなタスクに分け，実行中に，そのタスクを各計算ノードにスケジュールする方式である．PentiumM ノードの処理速度は，PentiumIII に比べて，静的負荷分散では 3.05 倍，動的負荷分散では 2.83 倍高速であった．したがって，この数値を用いて PenitumM の性能を PentiumIII の台数に換算し，並列化効率を算出する．この数値は，動作周波数の差とほぼ一致する．分子軌道計算では，メモリアクセスが少なく，浮動小数点処理部の性能が動作時間に大きく影響するので，PentiumM ノードに実装されているデュアルチャネルメモリなどの高速化技術が実行時間に影響を与えられなかったと思われる． PentiumM のシャーシ（２ノード実行）の消費電力（静的負荷分散）は 127W であり，PentiumIII のそれは 103W であった．積算電力（消費電力×計算時間）で比較すると，PentiumM シャーシは PentiumIII シャーシより，4.91 倍よい結果となった．また， Pentium4 (3.0GHz : Northwood, Hyper-threading, FSB800MHz)搭載のパソコンで同様の計算を行い，積算電力を計測した．その場合，PentiumM 筐体は Pentium4 パソコンより 1.96 倍よい結果となった．

（３）分子軌道計算における Intra-Chassis network のみを使用した場合（以後，1net と略す）と Intra および Inter-Chassis Network の両ネットワークを使用した場合（以後，2net と略す）の並列化効率を比較した．静的負荷分散方式における 1net と 2net の比較を図 8 に示し，動的負荷分散方式のそれを図 9 に示す．

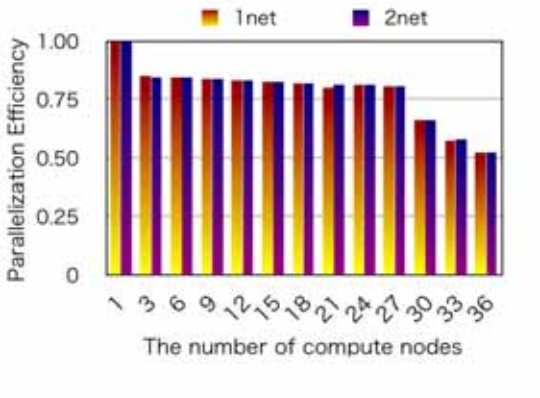

図 8　2net 化に伴う並列化効率の比較 (静的負荷分散方式)

図 8 において，1 から 27 ノードまでは，pentiumIII ノードのみでクラスタを構成し，30 ノード以降は，PentiumIII および PentiumM ノードで（ヘテロ）クラスタを構成した．なお，30 ノード以降の並列化効率は，PentiumM ノードの性能を PentiumIII ノードの性能に変換して計算した．図 8 からわかるように，静的負荷分散方式では，1net と 2net とではほとんど変化が見られなかった．これは，静

的負荷分散方式では，ほとんどの時間を計算時間に費やし，通信はほとんどしていないからである．今回計算した分子では，36 ノードの場合，少なくても 150 秒程度の計算と 1 秒も満たない通信を 13 回繰り返しただけだったので，2net 化にしたメリットが活かしきれなかった．

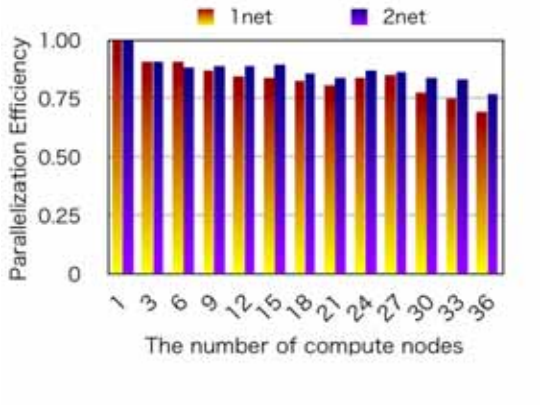

図 9　2net 化に伴う並列化効率の比較 (動的負荷分散方式)

動的負荷分散方式の場合，Intra-Chassis および Inter-Chassis Network を利用したネットワークは，Inter-Chassis Network のみと比べて，最大で約 10\%の性能改善が得られた．計算中，頻繁に通信を行うので，6 計算ノードあたりから 2net のほうが有利になったまた，1 から 27 台までの PentiumIII の時より，30 台から 36 台までの PentiumM が加わってからのほうが，2net による並列化効率の向上が顕著になった．これは，PentimM の場合，NIC を内蔵しているということおよび 2net 化による衝突回避が効いたためである．PentiumM の処理が早いため，PentiumIII より多くの計算タスクが割当られた．それらの通信の衝突回避が 1net に比べて多くなった．

（4）静的負荷分散では，PentiumM ノードの負荷が軽いため，並列化効率が低下した．そこで，PentiumM に FMC プロセスを複数実行させることにより，負荷バランスの改善を計った．PentiumM ノードで複数の FMC プロセスを並列実行させた場合の並列化効率を図 10 および図 11 に示す．図 10 は，分子軌道計算の中の Fock 行列生成に費やした時間(つまり，各 FMC プロセスの SCF をさせてから Barrier が終わるまで）である．一方，図 11 は，Fock 行列生成を行うための準備や Fock 行列生成処理の後処理など，並列処理していない部分も含めた全体の時間である．図 10 において，30 台，33 台において，4 プロセス実行の時に，並列化効率のピークパフォーマンスが得られ，36 台では，3 プロセス実行の時に，ピークパフォーマンスが得られた．PentiumM は PentiumIII に比べて，静的負荷分散方式では，2.83 倍速い．単純に考えれば,FMC プロセスを 3 に増やしたときに，並列化効率は低下するはずである．しかし，FMC プロセスを 4 にしても，並列化効率が低下しなかった．これは，PentiumM の 1 サイクルあたりの発行命令数にあると考えられる．PentiumM は 1 サイクルあたり最大 3 命令発行可能である．1 プロセスでは，命令発行スロットに空きがあり，最大命令発行ができず，FMC プロセスを増やすことによって，命令発行スロットが埋まり，最大発行が可能となり，FMC プロセスを 2 倍に増やしても，計算時間は，2 倍より小さい数値になったのではないかと推測される．

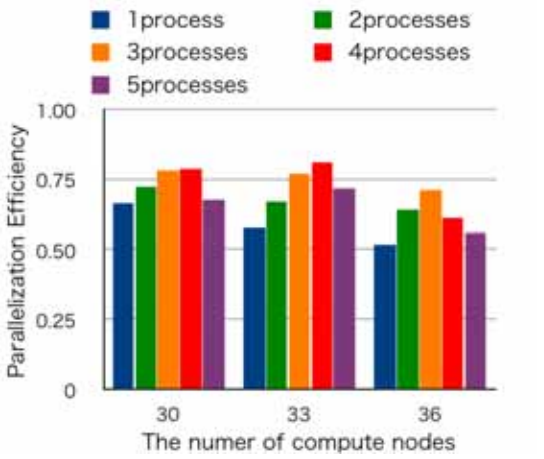

図 10　FMC プロセス数と並列化効率の関係 (Fock 行列生成処理時間のみ)

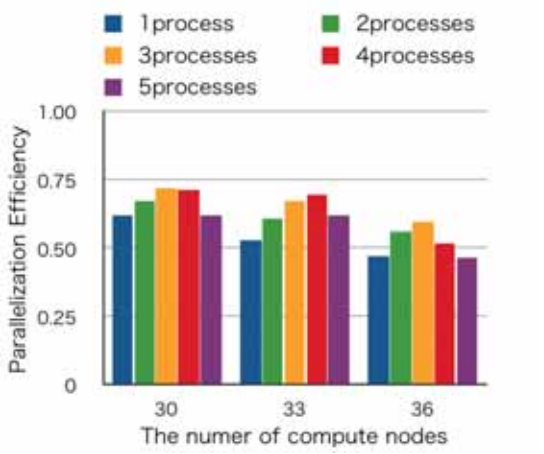

図 11　FMC プロセス数と並列化効率の関係 (全体の実行時間)

36 台の場合は，3FMC プロセスまでは，並列化効率が向上していったが，4FMC プロセスのときに，並列化効率が低下した．この原因としては，計算タスクの不均一さが原因である．今回の実装では，FMC プロセスの処理として，

分子軌道に影響を与えない計算は省いている．したがって，厳密には，同じ計算量のタスクでも計算時間に多少の差が生じる．FMC プロセスが少ない場合，計算タスクが大きいので，計算時間の差は現れにくいが，FMC プロセスが増加するにつれて，計算時間の差が現れて，並列化効率の低下をもたらしたと考えられる．図 11 において，30 台および 36 台の時に，4FMC プロセスを実行した場合，並列化効率の低下がみられた．FMC プロセスを増加させることにより，計算タスク分割やその転送などの時間がかかるため，トータルな計算時間では，30 台でも微小であるが，並列化効率の低下がみられ，36 台では，Fock 行列生成よりも急激に並列化効率が低下した．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 4 件）

① Kiyoshi Hayakawa, Tohru Sasaki, Hiroaki Umeda, Umpei Nagashima,「Evaluations of Load Balancing Methods for Molecular Orbital Calculations on a Hetero-Cluster System」， 20th IASTED International conference Parallel and Distributed Computing and Systems, pp.285-290, 2008.（査読あり）
② Kiyoshi Hayakawa, Tohru Sasaki, Hiroaki Umeda, Umpei Nagashima,「MOLECULAR ORBITAL CALCULATIONS ON EMBEDDED MIDDLE DENSITY CLUSTER SYSTEM」, 19th IASTED International conference Parallel and Distributed Computing and Systems, pp.1-6，2007.(査読あり)
③ Kiyoshi Hayakawa,「SCMD-CLUSTER : Design, Implementation, and Primary Evaluation」，IEEE 8th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp.171-172，2007.（査読あり）
④ 早川　潔,「低消費電力コンパクトクラスタの開発」，大阪府立工業高等専門学校研究紀要，第 42 巻，pp.35-42, 2007.（査読あり）

〔学会発表〕（計 4 件）

① 早川　潔，佐々木　徹，梅田　宏明，長嶋　雲兵,「低消費電力コンパクトクラスタによる分子軌道計算」，高槻産学連携フォーラム，pp.11(2008)．
② 早川　潔，佐々木　徹，梅田　宏明，長嶋　雲兵,「分子軌道計算におけるヘテロクラスタ上での効果的な負荷分散方式」，情報処理学会研究報告，2008-HPC-108，pp.79-84(2008)．
③ 松島　達哉，早川　潔,「低消費電力コンパクトクラスタにおけるコレクティブ通信ライブラリの開発」，日本高専学会第 14 回年会講演会論文集，pp.157-158(2008)．
④ 早川　潔，佐々木　徹，梅田　宏明，長嶋　雲兵,「コンパクトクラスタを利用した分子軌道計算の性能評価」,情報処理学会研究報告，2007-HPC-109，pp.1-6(2007)．

〔図書〕（計 1 件）

① Kiyoshi Hayakawa, Tohru Sasaki, Hiroaki Umeda, Umpei Nagashima,「A Communication Method for Orbital Calculations on a Compact Embedded Cluster」,Systems Modeling and Simulation, pp.357-361, Springer, ISBN-4431490213, 2006.

６．研究組織
(1)研究代表者

早川　潔（HAYAKAWA　KIYOSHI）
大阪府立工業高等専門学校・総合工学システム学科・電子情報コース・准教授
研究者番号：**20325575**