

機関番号：12601

研究種目：特定領域研究

研究期間：2006～2010

課題番号：18049015

研究課題名（和文） 情報爆発に対応する高度にスケーラブルなソフトウェア構成基盤

研究課題名（英文）

Highly Scalable Software Construction Basis for Information Explosion Era

研究代表者

近山 隆 (CHIKAYAMA TAKASHI)

東京大学・大学院工学系研究科・教授

研究者番号：40272380

研究成果の概要(和文):爆発的に増加する大量の情報を効率的に扱うソフトウェアの構成には、広域に分散配置した高度な並列性を持つ情報システムを柔軟に記述できるソフトウェアの枠組が基本技術として必要となる。このためのプログラミング言語やミドルウェアのシステムと、複雑なソフトウェアの正当性を検証するためのシステムを対象に研究を進め、具体的なシステムを提案、設計、実装し、その性能を検証した。代表的成果ソフトウェアは公開している。

研究成果の概要(英文): To cope with the explosive increase of data amount, frameworks for flexible description of software for widely distributed highly parallel information systems are required. For this purpose, programming languages, middleware systems, and verification systems for highly complicated software have been investigated, and such systems have been proposed, designed, implemented and evaluated the performance. Representative resultant software systems are made open to public.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2006年度	14,500,000	0	14,500,000
2007年度	9,600,000	0	9,600,000
2008年度	13,600,000	0	13,600,000
2009年度	11,900,000	0	11,900,000
2010年度	15,000,000	0	15,000,000
総計	64,600,000	0	64,600,000

研究分野：並列・分散処理ソフトウェア

科研費の分科・細目：情報学・情報学基礎

キーワード：計算機システム、ソフトウェア学、ハイパフォーマンスコンピューティング、並列分散処理、アルゴリズム

## 1. 研究開始当初の背景

情報システムで扱うデータ量が指数的に増大する「情報爆発」を扱うための情報基盤は、これまでにない規模で広く分散した多数の情報資源を統合し、高度で適応的な協調処理、柔軟な資源の共有を行なって、さまざまな状況に対応できる弾力性の高いものとする必要がある。この実現には、高度で複雑な機能を提供するソフトウェアを、百万台以上の計算ノードやペタバイト以上の分散した

データ、それらを結合するネットワークの上を実現しなくてはならないが、既存のソフトウェア構成技術にはこの達成のために不十分な点が多い。

## 2. 研究の目的

本研究においては、新しい環境に適したソフトウェアの効率的構成のための新たなモデル、モデルを体現するプログラム言語などの記述体系やその実装技術、それらの利用を

容易化するツール群といった、高度にスケラブルな情報基盤のためのレジリエントなソフトウェア基盤の基礎技術を確立する。

### 3. 研究の方法

本研究においては(1)不均一で時間的変化の大きい超大規模環境を前提とした計算モデルとその記述体系および実現手法、(2)超大規模環境上のさまざまな計算資源を選択し、割当てる機構とその実現手法、(3)これらの研究開発を加速し、アプリケーションの開発を容易化するためのツール群、の三点に焦点を絞り、研究開発を行なった。

研究は東京大学、京都大学、早稲田大学の三拠点で実施し、(1)については東大と早大、(2)については東大と京大、(3)については東大の研究者が主として担当し、電子メール等による連絡を頻繁にとりつつ、適宜打合せの機会を持ちながら研究を進めた。

成果として得たソフトウェアについては、研究の中間段階においても積極的に公開し、本特定領域研究の他の課題の研究者のみならず、広く一般の利用に供し、利用からのフィードバックを得て改良を施していく方針とした。

研究の実施には、この課題のために調達した機器や各研究拠点に既存の機器に加え、課題「情報爆発に対応する新IT基盤研究支援プラットフォームの構築」(支援班)で整備した分散並列環境を活用した。

### 4. 研究成果

#### (1) 計算モデルと記述体系

大量のデータを扱える並列分散処理を行うソフトウェアシステムにおいては、さまざまな処理を同時並行的に遂行するシステムを適切に記述できる計算モデルが重要である。こうした計算をどのように進行させるかを記述する計算モデルについては、階層グラフ書換えモデルに基づくプログラム言語LMNtalを対象に、理論・実装の両面からの検討を加えた(①)。

ソフトウェアの大規模化・複雑化に伴い、正当性の確保はますます困難になってきている。ことに広く分散した並列性の高いソフトウェアは、人手による正当性の確認はたいへん難しい。正当性の自動検証のための有力な手法としてモデル検査があるが、複雑なソフトウェアに対して計算量が爆発的に大きくなる問題があった。この問題に対処するために、状態空間爆発の抑制、探索の並列化などを施し、さらにその性能の鍵となる充足可能性問題の求解系についても、並列処理による高速化を図り、高い性能向上を得た(②)。

情報爆発に対応する大規模情報システムにおいては、分散配置したデータへのアクセスをどのように行うかが、システムの処理性

能を決める重要なファクタである。こうしたデータアクセスの記述については、メッセージ通信による明示的なデータ交換による記述と、仮想的に共有したメモリ空間を用いる暗黙的なデータ交換による記述という、異なる手法が併用されている。これは対象とする問題の性質から、記述性と実行性能のトレードオフ点が異なることが大きな原因となっている。このため、データアクセスのモデルについては、これらの両手法について研究を進め、ミドルウェアライブラリを構築、そのソフトウェアを公開した(③④)。

#### ① 計算モデル

本研究では階層グラフ書換えに基づく並行計算モデルLMNtalを対象に、その理論上の記述力や実際上の記述性についての解析を行った。

理論上の記述力については、既存の代表的な計算モデルとの間で相互のエンコーディングが可能であることを示すことによって確認することができた。一方、記述性の面では、最適化を施した処理系を実装、記述実験を通して記述の容易性を検証した。

#### ② モデル検査器

上述のとおり、計算モデルを与えるさまざまな体系による記述からLMNtalに容易に変換できること、プログラムがそのままモデル検査器にかけることができる利点があることから、LMNtalをモデル記述言語とするモデル検査器を中心に研究を進めた。

LMNtalのデータ構造である階層グラフ構造を用いた状態空間探索は、問題の対称性をうまく吸収する特徴し、状態爆発を抑制する点を大きな特徴とする。この特性に加え、動的負荷分散機構・並列ハッシュ表の採用などの最適化による、メモリ使用量を抑制して検証可能な規模を一桁拡大し、並列処理性能の向上により30倍以上の高速化を実現した。

非決定的動作を許すモデル記述に大しては、その性質や挙動の理解は容易でない。そこで、モデル検査器にシステムの状態遷移グラフや検証結果の可視化ができる機能を付加した。

LMNtal処理系、モデル検査器、グラフィカルな編集系などを統合し、GUIを備えた開発環境LaViTを構築した。このシステムを公開し、広く一般の利用に供している。

命題論理式の充足可能性問題の求解系(SATソルバ)は、20世紀末からさまざまな技法の組合せによって劇的に性能を向上し、について、モデル検査を含む多くの探索問題の実現のための基礎技術として注目されている。しかし、高度に最適化されたSATソルバの並列実行方式や、そのスケラビリティの検討は立ち遅れていた。我々は、本研究開始時点で最新のオープンソースソルバであ

った MiniSat をベースに、種々の最適化を施した並列 SAT ソルバ c-sat を設計・構築し、31PE 規模のクラスタ上でベンチマーク課題について 23~31 倍の性能向上を得た。

c-sat がベースとしている MiniSat v1.14.1 は開発から 4 年以上が経過しているが、SAT ソルバその後も大幅な性能改善が進んでいる。SAT Competition 2009 の Application 部門で最も優秀な成績を取めた逐次ソルバ precosat は、種々の新たな技法を実装し、v2.0 以前の MiniSat が不得意であった問題を中心に大幅な逐次性能向上を達成したが、それだけに precosat の並列化によって c-sat 並みの台数効果を得ることは容易でなくなっている。そこで我々は、c-sat のアーキテクチャに基づく precosat の並列化実装 c-preco をまず構築し、それを出発点に precosat と MiniSat の双方をワークに持つ c-satw を実装した。さらに c-satw に対してサポートベクタマシンによるソルバコンフィギュレーションのパラメタ学習を追加した c-satws を作成し、有意な速度改善効果を確認した。c-satw および c-satws の並列効果は c-sat を大きく下回るものの、31PE での絶対性能は、2010 年の SATRace に出場したすべての逐次ソルバ、c-sat、および precosat を単純に複数本実行させる単純並列ソルバの性能を上回ることを確認した。

オートマトンベースの LTL モデル検査は受理サイクル探索問題を解く基本的な手法の一つであり、その並列実行は、モデル検査の状態空間爆発対策として有望なアプローチである。しかし、逐次最適とされているアルゴリズムは post-order の縦型探索に基づいており、P-完全で並列化は困難である。そのため代替りの並列アルゴリズムが提案されており、その一つである OWCTY\_reversed は十リサイクルがあるモデルに対して安定して高速に動作するアルゴリズムであるが、しかし、検証に用いるオートマトンの特性を考慮した最適化がなされているとは言えない。そこで我々は、探索する状態遷移グラフの強連結成分の特性を考え、受理サイクルを作らないと判断できる状態を早いタイミングで削除することで全体の処理量を削減するアルゴリズム SCC-OWCTY を提案した。本アルゴリズムを分散検証環境 DiVinE に実装し、性能と台数効果の両方が改善することを確認した。

### ③メッセージ通信によるアクセス記述

メッセージ通信に基づく記述には、Message Passing Interface (MPI) が事実上の標準として広く用いられている。MPI の既存の実装は、すべてのサーバが相互に直接通信する方式をとっている。これは数十台程度の規模のサーバを持つ単一のクラスタ型シ

ステムなど、比較的少数のノードが均質なネットワークにより結合されている場合には問題がない。しかしながら、非常に多数のノードが広く分散して配置されている環境においては、多数の短い通信パケットが送受されることにより、通信オーバーヘッドが大きくなる。また、多数のノードがすべて直接相互通信するためには、多数の送受信ソケットを必要とし、通信バッファのために過度なメモリを確保しようとする問題がある。さらに、ノード間の通信遅延時間がまちまちであることが考慮されていない。このため、多数のクラスタを相互接続したような大規模環境においては、性能上の問題を抱えていた。

この問題を解決するために、我々はクラスタ内外のノードの相互接続状況 (トポロジ) を効率的に自動検出する方式を考案した。そして、対象システムのトポロジに対応して中継を含む適切な通信経路を自動設定する機能を設計・実装した。これによって、接続ノード対の総数を大幅に削減し、通信オーバーヘッドを削減するとともにバッファに必要なメモリ量が過多になることを防止することを可能とした。こうした最適化を適用した MPI ライブラリの実装である MC-MPI を公開し、広い利用に供している。

### ④共有メモリによるアクセス記述

MPI のように明示的な通信を用いる方式は、分散した情報システムの物理的特性に沿った記述が可能で、効率的な処理の記述に有利な面があるが、通信を逐一記述することが煩雑になる難点がある。複数のプロセスがメモリ領域を共有するモデルは、記述の容易性の上では有利であるが、物理的に分散したメモリを共有メモリとして記述するため、データアクセスパターンについての指定なしでは、効率的な実装が困難である。

この問題を解決するために、共有メモリ方式ではあるが、どのような共有を行うかを明示的に指定するミドルウェアライブラリ Distributed Memory Interface (DMI) を、設計・実装した。このライブラリは、物理的には分散したメモリ上のデータについて、メッセージ通信を用いて共有しているかのように扱う仮想共有メモリシステムである。実装のベースには MPI を用いているが、ベンチマーキングの結果、共有メモリという直感的な記述を用いながら、MPI を直接用いて記述した場合と大差ない性能を実現できることを確認した。この DMI も公開ソフトウェアとして広い利用に供している。

### (2) 計算資源選択・割当て機構

データ量の爆発的な増加によって、もっとも重要な計算資源のひとつであるメモリ領域の自動管理機構であるごみ集め (GC) 機構

についても、従来採用されてきたさまざまな実装技術・アルゴリズムは見直しが必要となっている。

研究期間の前半においては、研究のプラットフォームとなる実行系を、Java で記述された Lisp 処理系である JAKLD を書き換えることによって開発した。この実行系は GC アルゴリズムに関する部分をマクロ化することにより、GC アルゴリズムを 150 行程度で実装できるようになっている。この上で主要な GC アルゴリズム 9 種を実装、相互の比較検討を行った。

これらの研究成果に基づき、研究期間の後半では、研究クラウド端末の応答性を向上させるために実時間 GC 技術を適用する研究を行った。個人では管理できないほど膨大な量のデータを利用・生成可能な状況下、そのような情報を効率よく管理し、いつでもどこでも必要なときに利用できる技術としてクラウドを捉えることができる。クラウドは通常の PC でももちろん利用できるが、クラウドと相性の良い携帯可能な端末を利用することにより、その価値がさらに向上する。

本研究では、代表的なクラウドサービスである Google 社が提供するクラウドサービスとの相性が良い Android 端末の応答性能を向上することを目指した。

Android のアプリケーションは基本的に Java 言語で記述され、Android 内蔵の Dalvik という仮想マシン (VM) の命令列に変換されて実行される。アプリケーションの実行には動的データ生成が不可欠であり、GC は必須の要件である。当初の Android は比較的小規模の携帯電話等を対象としていたため、GC 処理も単純なものであったが、端末性能の向上によって、ヒープメモリが大規模化する傾向にあり、GC 処理に伴う端末の応答性能が問題になりつつある。

この問題を解決するために、我々のグループが開発したいくつかの実時間 GC 技術 (アプリケーションの停止時間を短時間に抑えて応答性能を向上する技術) である、スナップショット方式と複製に基づく実時間コピー方式を Dalvik に適用した。Dalvik の枠組みをできるだけ変更せずに実時間 GC 技術を実装する方式を開発し、実際に最新版の Dalvik に組み込んだ。

### (3) 分散計算機環境のためのツール群

分散配置された計算機環境を統一的に利用してソフトウェア開発を行うための環境である InTrigger を開発した。InTrigger は多拠点に渡る分散計算機環境の構成を柔軟に変化させながら管理する機構を持ち、分散環境を使いこなすためのさまざまなツールを提供する。

実現した InTrigger 環境は、課題「情報爆

発に対応する新 IT 基盤研究支援プラットフォームの構築」(支援班) で整備した分散並列環境上に展開し、本特定領域研究の他の課題での利用に供し、研究の加速に貢献できた。

### ① 柔軟な構成変化を考慮した管理

様々な拠点に分散したクラスタ計算機の管理を行うとき、効率的に作業を行うためにはなるべく多くの作業をネットワーク経由で遠隔地から行うことができる必要がある。また、システムソフトウェアの研究では様々なソフトウェアをインストールしたり OS の設定を書き換えるような実験を行ったりする場合があるが、このようなとき、ソフトウェアパッケージをインストールするのと同じような感覚で気軽にクラスタ内の各ノードの OS を再インストールできると便利である。

これらのことを実現するためにネットワーク経由で OS のインストールを行うためのツールである Lucie4) を拡張した。元の Lucie ではすべてのノードのインストールが完了するまで待機し、自動インストールを行う設定を解除した後に、各ノードの端末を操作して再起動を指示することが必要であるが、これでは遠隔インストールには向かない。そこでインストール完了時にインストール用サーバに通知する機能を付加した。

インストールに問題があった場合には IPMI ( IntelligentPlatform Management Interface) 機能を用いてトラブルシュートを行う。IPMI を用いるとノードの電源をリセットすることができ、また、IPMI の SOL (Serial on LAN) を用いるとコンソール表示を遠隔ノードにリダイレクトすることができる。

また、元の Lucie は OS だけでなく、すべてのソフトウェアをインストールするが、ルートファイルシステムを NFS マウントした状態でのトラブルシュートは困難である。そこで、必要最低限のソフトウェア以外は、各ノードがローカルディスクから起動した状態でインストールするように変更した。インストールするソフトウェアの設定を行うスクリプトを Subversion で管理することによって、全ノードに共通のソフトウェアをインストールするようにした。

### ② GXP (Grid eXplorer)

分散環境を効率的に扱うためには、多くの計算機への素早いコマンド投入や、これらの計算機を協調動作させる分散アプリケーションを簡単に記述できることが重要になる。また、これらを複雑な設定を行わずに可能にすることが求められる。GXP はコマンドラインインタフェースの並列シェルで、複数拠点にまたがる分散環境で効率的に利用で

きるログイン機能とコマンド実行機能が実装されている。

### ③ VGXP

分散並列計算環境は、計算機の台数が増えるほど、また分散している拠点の数が増えるほど安定した状態で利用することが困難になる。不測の問題が発生したときに計算環境の状態を迅速に把握して対処を行うためには、モニタリングシステムの存在が不可欠となる。

複数の拠点に対応できる既存のモニタリングシステムでは、ノードの生死や監視項目の値があらかじめ設定した範囲にあるかどうかを判定してそれらを表示することができるが、望ましい値の範囲を静的に決定することは難しいし、設定も煩雑である。また、これらのツールは1分から数分ごとの値の変化を把握するように設計されているため、短時間のジョブについての動きを監視するには適していない。

Visual GXPは、通信量や監視の負荷を抑えつつ1秒から数秒間隔で各監視項目の値を集め、各計算機の状態変化を迅速に把握できるように設計された多拠点分散環境向けのモニタリングシステムである。収集した各計算機の情報はそれぞれを比較しやすいように可視化を行い、特別にしきい値を設定しなくとも異常の可能性を視覚的に判断し、必要に応じてさらに詳しい情報を見ることができる。

## 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 30 件)

- ① 川端 聡基、小林 史佳、上田 和紀、強連結成分の性質を用いた OWCTY モデル検査アルゴリズムの高速化、人工知能学会論文誌、査読有、26(2)、2010、341-346
- ② 綾野 貴之、堀 泰祐、岩澤 宏希、小川 誠司、上田 和紀、統合開発環境による LMNtal モデル検査、コンピュータソフトウェア、査読有、27(4)、2010、197-214
- ③ 原 健太郎、田浦 健次朗、近山 隆、DMI: 計算資源の動的な参加/脱退をサポートする大規模分散共有メモリインタフェース、情報処理学会論文誌(プログラミング)、査読有、3(1)、1-40、2010
- ④ 柴田 剛志、田浦 健次朗、トポロジ情報を用いた効率的かつ漸近安定な大容量ブロードキャスト、情報処理学会論文誌コンピュータエン지니어リングシステム(ACS)、査読有、2(3)、47-57、2009
- ⑤ 馬谷 誠二、八杉 昌宏、湯淺 太一、動的名前解決による通信先・移動先の柔軟な指

定が可能な分散アンビエントシステムの設計、コンピュータソフトウェア、査読有、27(2)、50-61、2009

- ⑥ 藤川 浩光、馬谷 誠二、八杉 昌宏、湯淺 太一、リターンバリア型実時間ごみ集めの抽象モデル検査、情報処理学会論文誌(プログラミング)、査読有、2(4)、13-32、2009
- ⑦ Kazunori Ueda, LMNtal as a Hierarchical Logic Programming Language, Theoretical Computer Science, 410(46), 査読有、4784-4800, 2009
- ⑧ Hideo Saito, Kenjiro Taura, Takashi Chikayama, Collective Operations for Wide-area Message Passing Systems Using Adaptive Spanning Trees, International Journal of High Performance Computing and Networking, 査読有、5(3), 179-188, 2008
- ⑨ Kazunori Ueda, Encoding Distributed Process Calculi into LMNtal Electronic Notes in Theoretical Computer Science, 査読有、209, 187-200, 2008
- ⑩ 弘中 健、斎藤 秀雄、高橋 慧、田浦 健次朗、複雑なグリッド環境で柔軟なプログラミングを実現するフレームワーク、情報処理学会論文誌(コンピューティングシステム)、査読有、1(2)、157-168、2008
- ⑪ 上田和紀、論理・制約プログラミングと並行計算、コンピュータソフトウェア、査読有、25(3)、59-54、2008
- ⑫ 村山 敬、工藤 晋太郎、櫻井 健、水野 謙、加藤 紀夫、上田 和紀、階層グラフ書換え言語 LMNtal の処理系、コンピュータソフトウェア、査読有、25(2)、47-77、2008
- ⑬ 鶴川 始陽、花井 亮、八杉 昌宏、湯淺 太一、マルチスレッド環境における実時間ごみ集めのためのスレッド再開バリア、コンピュータソフトウェア、査読有、25(2)、135-150、2008
- ⑭ 乾 敦行、工藤 晋太郎、原 耕司、水野 謙、加藤 紀夫、上田 和紀、階層グラフ書換えモデルに基づく統合プログラミング言語 LMNtal、コンピュータソフトウェア、査読有、25(1)、124-150、2008

[学会発表] (計 139 件)

- ① Kentaro Hara, Kenjiro Taura, A Global Address Space Framework for Irregular Applications International, ACM Symposium on High Performance Distributed Computing, 査読有、296-299
- ② Takeshi Shibata, SungJun Choi, Kenjiro Taura, File-Access Patterns of Data-Intensive Workflow Applications and their Implications to Distributed Filesystems, International Workshop on Data Intensive Distributed Computing,

- 査読有, 746-755, 2010
- ③ Tomoharu Ugawa, Hideya Iwasaki, Taiichi Yuasa, Starvation-free Heap Size for Replication-Based Incremental Compacting Garbage Collection, International Lisp Conference 2010, 査読有, 43-50, 2010
  - ④ Tomoharu Ugawa, Hideya Iwasaki, Taiichi Yuasa, ACM SIGPLAN International Symposium on Memory Management, 査読有, 73-82, 2010
  - ⑤ Kazunori Ueda, Takayuki Ayano, Taisuke Hori, Hiroki Iwasawa, Seiji Ogawa, Hierarchical Graph Rewriting as a Unifying Tool for Analyzing and Understanding Nondeterministic Systems, International Colloquium on Theoretical Aspects of Computing, 査読有, 349-355, 2009
  - ⑥ Kei Ohmura, Kazunori Ueda, c-sat: A Parallel SAT Solver for Clusters, International Conference on Theory and Applications of Satisfiability Testing, 524-537, 2009
  - ⑦ Hideo Saito, Ken Hironaka, Kenjiro Taura, A Scalable High-performance Communication Library for Wide-area Environments, 査読有, 9th IEEE/ACM International Conference on Grid Computing, 310~315, 2008
  - ⑧ Yoshikazu Kamoshida, Kenjiro Taura, Scalable Data Gathering for Real-time Monitoring Systems on Distributed Computing, IEEE International Symposium on Cluster Computing and the Grid, 査読有, 425-432, 2008
  - ⑨ Kazunori Ueda, Encoding the Pure Lambda Calculus into Hierarchical Graph Rewriting, International Conference on Rewriting Techniques and Applications, 査読有, 392-408, 2008
  - ⑩ Tomoharu Ugawa, Masahiro Yasugi, Taiichi Yuasa, Replication-Based Incremental Compaction, International Symposium on Object/component/service-oriented Real-time Distributed Computing, 査読有, 516-524, 2008

[その他：公開ソフトウェア]

- ① 田浦 健次郎 他、Intrigger プラットフォーム, <https://www.logos.ic.i.u-tokyo.ac.jp/intrigger/registration/info.php>, 情報爆発時代に向けた新しい IT 基盤技術の研究のための広域分散プラットフォーム
- ② 齋藤 秀雄、田浦健次郎 他、MC-MPI, <http://www.logos.ic.i.u-tokyo.ac.jp/>

~h\_saito/mcmapi/, MPI のマルチクラスタ向き実装

- ③ 原 健太郎、田浦 健次郎 他、Distributed Memory Interface, <http://haraken.info/index.php?tag=iapwjfnx>, 分散共有メモリシステム
- ④ 田浦 健次郎、GXP Grid/Cluster Shell, [http://www.logos.ic.i.u-tokyo.ac.jp/phoenix/gxp\\_quick\\_man\\_ja.shtml](http://www.logos.ic.i.u-tokyo.ac.jp/phoenix/gxp_quick_man_ja.shtml), 広域分散環境を使うための並列シェル
- ⑤ 上田 和紀 他、LMNtal および LMNtal モデル検査器, <http://www.ueda.info.waseda.ac.jp/lmntal/>, 非決定実行およびモデル検査機能を備えた階層グラフ書換え言語 LMNtal の統合環境

## 6. 研究組織

### (1) 研究代表者

近山 隆 (CHIKAYAMA TAKASHI)

**東大・工学系・教授**

研究者番号：40272380

### (2) 研究分担者

湯浅 太一 (YUASA TAIICHI)

京大・情報学・教授

研究者番号：60158326

上田 和紀 (UEDA KAZUNORI)

早大・理工学術院・教授

研究者番号：10257206

### (3) 連携研究者

田浦 健次郎 (TAURA KENJIRO)

東大・情報理工学系研究科・准教授

研究者番号：90282714

遠藤 敏夫 (ENDO TOSHIO)

東工大・情報理工学研究科・特任准教授

研究者番号：80396788

横山 大作 (YOKOYAMA DAISAKU)

東大・生産技術研究所・助教

研究者番号：80345272

馬谷 誠二 (UMATANI SEIJI)

京大・情報学研究科・助教

研究者番号：40378831