様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月 １１ 日現在

機関番号：１２６０１
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５０００３１

研究課題名（和文）準パススルー型仮想マシンモニタの研究

研究課題名（英文）A study on parapass-through virtual machine monitors

研究代表者
品川　高廣（SHINAGAWA, Takahiro）

東京大学・情報基盤センター・准教授

研究者番号：４０３６１７４５

交付決定額（研究期間全体）：（直接経費）　　4,100,000円

研究成果の概要（和文）：本研究の目的は，申請者が研究開発してきた仮想マシンモニタ(VMM)「BitVisor」を発展させ，複雑なオペレーティングシステム(OS)の機能を補って安全性や機能・性能を向上させるための，安全・軽量なプラットフォームを実現することである．BitVisor は，準パススルー型という申請者が考案した新しいアーキテクチャのVMMであり，ゲストOS の機能を部分的に活用することで，従来のVMM と比べて軽量・小型でありながら高い安全性を実現している．本研究では，安全性向上に加えて，機能追加や性能向上への応用可能性を探り，準パススルー型アーキテクチャの有用性を向上させるための研究開発をおこなった。

研究成果の概要（英文）：The purpose of this study is to extend our original virtual machine monitor (VMM), called BitVisor, and making a secure and lightweight platform for improving functionality and performance of existing complex operating systems (OSs). BitVisor is a VMM with a new architecture, called parapass-through, that achieves high security while are lightweight and compact compared to conventional VMMs by partially exploiting the functions of guest OSs. In this research, we developed new features for improving functions and performance, in addition to security, to improve the usefulness of the parapass-through architecture.

研究分野：オペレーティングシステム

キーワード：仮想化技術　BitVisor

１．研究開始当初の背景

(1) 仮想マシンモニタによるシステム管理：近年､多くの IaaS 事業者は仮想マシンを貸し出している。仮想化技術を用いることで､IaaS 事業者はクラウドコンピューティングに必要不可欠な要素を提供することが出来る。様々な研究により、仮想化のオーバーヘッドを削減することが出来るようになってきたが、最先端の性能や機能、セキュリティなどが必要とされる環境においては、依然として仮想化のオーバーヘッドが問題になっている。例えば、ビッグデータや高性能計算、データベース・サーバなどにおいては定常的に安定した高性能を出し続けるために、仮想化による予測不可能な性能低下を避ける必要がある。また、GPU や RAID,SSD といった高性能ハードウェアの性能を引き出す必要があるケースや、セキュリティ上同一マシン上で複数のテナントの仮想マシンが動作することが望ましくないケースなどにおいても、仮想化による性能・機能低下やセキュリティ低下が問題となっている。これらの問題を解決するために、ベアメタル・クラウドと呼ばれる物理マシンを貸し出すサービスが近年 IBM をはじめとして多くの事業者により提供されるようになってきている。しかし、ベアメタル・クラウドでは、OS を最初に導入する際に、OS 依存の作業が必要になるほか導入に非常に長い時間がかかるといった問題点がある。OS を導入する最も基本的な手法はイメージコピーと呼ばれるものであり、OS 全体のイメージをサーバからネットワーク経由でコピーしてきて、ローカルのストレージに書き込む手法である。この手法では、コピーが完了するまでに十数分の時間がかかる。また、書き込み後に再起動をする際にも、特にサーバマシンにおいてはファームウェアの初期化などを行うために非常に長い時間がかかる。OS の導入に時間がかかると、クラウドの利点である俊敏性や拡張性といった重要な性質を実現することが難しくなる。

(2) 仮想マシンモニタによる物理マシン上での OS ライブマイグレーション：ベアメタル・クラウドは仮想マシンの代わりに物理マシンを貸し出すものである。OS からハードウェアへのアクセスに対して、仮想化のレイヤが介在することなく直接アクセスすることを許容するため、ハードウェア・プラットフォームの最大限の性能を安定して提供することが可能である。従って、ベアメタルインスタンスはビッグデータや高性能計算など、仮想化のオーバーヘッドが無視できない非常に高性能な計算が必要な場合に適している。ライブマイグレーションは、IaaS クラウドの重要な性質の一つである。例えば、IaaS ベンダーは予防的メンテナンスのために、ライブマイグレーションを使用する。これは、システムの信頼性やセキュリティ、パフォーマンスを維持するために必要不可欠な作業である。電源などのメンテナンスには電源切断が必須であるが、ライブマイグレーションを用いると IaaS ベンダーはサービスを停止させることなくメンテナンスを実施することが可能になる。しかし、ライブマイグレーションはベアメタルインスタンスではサポートされていない。従って、ベアメタル・クラウドではハードウェアメンテナンスの際にダウンタイムが避けがたい。ライブマイグレーションは、以下の理由から、OS からは透過的に実現されるべきである。まず､IaaS 事業者は通常は顧客の OS を直接は管理していない。従って、ライブマイグレーションが OS に依存すると、顧客自身が必要な時にライブマイグレーションの作業を実施する必要が生じる。顧客がいつでもライブマイグレーションを実施できるように待機していることは現実的ではない。また、追加のソフトウェアやデバイスドライバを導入するのは面倒な作業である。さらに、顧客は独自の OS を使いたいかもしれない。そのような場合、特殊なソフトウェアやデバイスドライバは入手できないかもしれない。

(3) 仮想マシンモニタによるシステムイメージの完全性保護：近年、OS のカーネル権限を不正に取得する攻撃が増加している。攻撃者は、カーネルルートキットと呼ばれるソフトウェアを使うことで OS のカーネル権限を取得し、セキュリティソフトウェアを無効にするなどして、カーネル権限での不正アクセスを継続的におこなうことが出来る。攻撃者は、OS が再起動されても不正アクセスを継続するためには、カーネルルートキットをハードディスク等にインストールして永続化する必要がある。永続化が、システムの基幹部分のファイルであるシステムイメージを改ざんすることでおこなわれた場合、ファイルが隠蔽されたりセキュリティソフトが無効にされたりするため、その検出や除去は非常に困難になる。カーネル権限を取得する攻撃を防止する手法としては、仮想マシンモニタを用いて OS の外部から監視する手法が研究されている。しかし、OS のファイルシステムのレベルで完全性保護をおこなうためには、仮想マシンモニタからファイルシステムを解釈する必要があり、セマンティックギャップと呼ばれる問題を解決する必要がある。

(4) 仮想マシンモニタによるボランティアコンピューティング基盤の実現：ボランティアコンピューティングとは､一般の PC

ユーザがボランティアとして計算資源の余剰分を提供し、企業や研究機関（組織）がその計算資源を用いて計算処理をおこなう仕組みである。組織側は高価なコンピュータを持たずに大規模計算をおこなえるという利点がある。ボランティアコンピューティングでは、組織側及びボランティア側のそれぞれに問題が存在する。まず組織側にとっては、悪意のあるボランティアが組織側の計算結果などを漏洩・改竄してしまう恐れがある。例えば、重要な計算結果が外部に流出する危険性があるほか、そもそも計算結果が正しいことを保障することが困難である。ボランティア側にとっては、組織側の計算コードがボランティアの PC 環境を壊してしまう恐れがある。ボランティアコンピューティングでは基本的には組織は信頼できることを想定しているが、組織側の計算コードのバグによってマシン環境が壊されてしまう可能性は否定できない。

(5) 仮想マシンモニタによる物理デバイスドライバ検証環境の実現：Windows や Linux のなど多くの OS では、デバイスドライバがシステムクラッシュの原因の多くを占める。例えば、Linux カーネルではデバイスドライバのコードが約 70%を占め、その不具合の密度はデバイスドライバ以外のコードに比べて、3～7 倍であることが知られている。そのため、OS の安定性を維持するためには、デバイスドライバの検証が重要である。しかし、特定のまれな条件やタイミングでのみ起こるような不具合は、再現することが難しい。例えば、デバイスがハードウェアエラーを起こした時のエラー処理などは、実際に試すためには意図的にハードウェアエラーを引き起こす必要があるが、これは一般には用意には実現できない。そのため、このような場合のエラー処理は十分に検証をおこなえない可能性がある。

(6) 仮想マシンモニタによるストレージクラスメモリへのアクセスの捕捉：近年、DRAM に近い性能が出る不揮発性ストレージであるストレージクラス・メモリ（SCM）が登場している。SCM はメモリバスに直接接続され、通常のメモリ・アクセス命令でアクセスすることができる。SCM では、従来のストレージのようにデバイスドライバやファイルシステムなどのソフトウェアスタックを用いると、レイテンシに対するオーバーヘッドが非常に大きくなってしまうという問題点がある。従って、近年 OS のストレージスタックを最小化する研究が多くなされている。我々は、仮想マシンモニタのレイヤでストレージの暗号化を強制的におこなうシステムの研究をしてきた。このシステムでは、仮想マシンモニタは I/O 命令を捕捉して、ブロック単位で転送されるデータを暗号化する。しかし、SCM ではデータにアクセスする際に I/O 命令は使われないので、この手法を適用することは出来ない。更に、SCM では通常のロード／ストア命令を用いてバイト単位でのアクセスがおこなわれるため、単純にこの命令を捕捉して暗号化をおこなおうとすると、仮想マシンモニタによる捕捉のオーバーヘッドが著しく大きくなってしまう。

２．研究の目的

(1) 仮想マシンモニタによるシステム管理：本研究では、古典的な IaaS クラウドの様々な利点をベアメタル・クラウドに導入することを目的としている。具体的には、俊敏性、拡張性、OS 透過性である。

(2) 仮想マシンモニタによる物理マシン上での OS ライブマイグレーション：本研究では、OS を修正することなくベアメタルインスタンスのライブマイグレーションを実現するシステムを実現することを目的としている。

(3) 仮想マシンモニタによるシステムイメージの完全性保護：本研究では、軽量な仮想マシンモニタを用いて、システムイメージの完全性を保護する手法を実現することを目的としている。

(4) 仮想マシンモニタによるボランティアコンピューティング基盤の実現：本研究では、信頼できる仮想マシンモニタにより、ボランティアコンピューティングにおける組織側及びボランティア側の双方のセキュリティを実現する手法を実現することを目的としている。

(5) 仮想マシンモニタによる物理デバイスドライバ検証環境の実現：本研究では、仮想マシンモニタを用いて、OS やハードウェア、デバイスドライバのソースコードに依存せずに、一般的なハードウェア上で実デバイスドライバの検証を行う手法を実現することを目的としている。

(6) 仮想マシンモニタによるストレージクラスメモリへのアクセスの捕捉：本研究では、仮想マシンモニタにより SCM へのストレージアクセスを仲介するシステムを、大きなオーバーヘッドを導入することなく実現することを目的としている。

３．研究の方法

(1) 仮想マシンモニタによるシステム管理：本手法の実現にあたっては、ベアメタル・クラウドの性能面での利点を損なわないことが重要である。これを実現するために、本研究では脱仮想化が可能な専用の仮想マシンモニタを用いて、OS 透過で高速なベアメタル・マシンのインスタンスの立ち上げを可能にするシステムを提案する。本システムでは、OS ストリーミング・デプロイメントを仮想マシ

ンモニタのレイヤでおこなうことによって、OS 透過性を維持しつつも高速なインスタンスの立ち上げを実現する。一方で、OS からはハードウェアに対して直接アクセスすることを許可することによって、仮想化のオーバーヘッドを最小限に抑える。OS の導入が終了した後には、仮想化をオフにして OS の下からいなくなることによって、仮想化のオーバーヘッドをゼロにする。この手法で最も難しいのは、２つの相反する目標を実現することである。すなわち、OS ストリーミング･デプロイメントのためにゲスト OS と仮想マシンモニタでデバイスを共有しつつ、最終的にベアメタル・マシンの性能を実現するためにシームレスに脱仮想化を実現することである。仮想デバイスを用いると、デバイスの共有は容易になるが、脱仮想化が難しくなる。デバイスを直接ゲスト OS に見せると、脱仮想化は容易になるが、デバイスの共有が難しくなる。この問題に対処するため、本研究では、デバイス・インターフェイスのレベルで I/O を仲介するデバイス・メディエータという概念を導入する。デバイス・メディエータは、I/O を注意深く監視、捕捉、変換、挿入することにより、デバイスの共有とシームレスな脱仮想化の両方を実現する。

(2) 仮想マシンモニタによる物理マシン上での OS ライブマイグレーション：ライブマイグレーションを実現しつつベアメタルインスタンスと同等の実行環境を実現するために、準パススルー型の仮想マシンモニタを採用する。ライブマイグレーションをおこなうために、仮想マシンモニタは移動元の CPU やメモリ、I/O デバイスなどのハードウェアの状態を取得し、移動先に転送する。移動先のマシンでは、その状態を適切に適用する。この手法で最も難しいのは、I/O デバイスの状態を取得・設定することである。仮想化環境においては、仮想マシンモニタは仮想デバイスの状態を簡単に取得・設定することができるが、物理環境においては、I/O デバイスの内部状態を取得したり設定することは簡単ではない。例えば、リードオンリーのレジスタやライトオンリーのレジスタなどが存在している。これらの状態を取得・設定するために、仮想マシンモニタは、ゲスト OS からのライトオンリーのレジスタに対するアクセスを監視・捕捉して、その状態を内部に保存しておく。また、リードオンリーレジスタの状態を設定するために、デバイスを意図的に操作することで間接的に状態を変更させる。これらの手法により、I/O をデバイスの内部状態をマイグレーションすることが可能になる。

(3) 仮想マシンモニタによるシステムイメージの完全性保護：ファイルシステムの仕様に基づいて、事前にファイルのデータとメタデータのストレージデバイス内での位置を正確に把握し、書き込み禁止箇所をバイト単位で指定したセキュリティポリシーを作成する。このセキュリティポリシーに基づいて、仮想マシンモニタによりストレージへのアクセスを捕捉してシステムイメージへの書き込みを確実に防止する。仮想マシンモニタの TCB(Trusted Computing Base)のサイズを小さく抑えて安全性を向上させるために、仮想マシンモニタはディスクに対して能動的にはアクセスせず、受動的に書き込みを検査するだけでシステムイメージの完全性を保護できる手法を実現した。

(4) 仮想マシンモニタによるボランティアコンピューティング基盤の実現：まず、組織が用意した計算結果などの漏洩・改竄を防止するために、計算処理をボランティアの PC 上で動作する仮想マシンモニタの内部でおこなう。仮想マシンモニタは OS よりも高い特権で動作しており、ボランティアの OS 側から VMM 内の計算コードやデータにアクセスすることを防止できる。組織側は、仮想マシンモニタと暗号通信をおこなうほか、仮想マシンモニタ自身が改竄されていないことを検証する。また、ボランティア側の PC 環境が壊されてしまうことを防止するために、組織側の計算コードは、仮想マシンモニタ内部の保護ドメインでアクセス権が制限して実行する。保護ドメイン内では、あらかじめ割り当てられた資源へのアクセスのみが可能であり、ボランティアのマシン環境を保護する事が出来る。また、準パススルー型アーキテクチャを活用して、ゲスト OS からハードウェアへのアクセスの大部分をパススルーとしつつ、計算コードに最低限必要な CPU・メモリなどの計算資源の確保と、計算コードの実行環境に対する保護のみを仮想マシンモニタで実現することにより、オーバーヘッドを削減するほか、既存のインストール済みの環境に対する導入を容易にする。

(5) 仮想マシンモニタによる物理デバイスドライバ検証環境の実現：まず、デバイスとデバイスドライバの状態を把握するために、対象となるデバイスドライバが動作する OS を、準パススルー型仮想マシンモニタ上で動作させる。これにより、ゲスト OS のデバイスドライバに物理デバイスを直接見せて制御させつつ、デバイスドライバから物理デバイスへの I/O を監視する。また、検証対象となるデバイスの状態遷移を監視するために、仮想マシンモニタ内でデバイスの仕様に基づいて状態遷移を監視するオートマトンを

動作させる。特定のまれな条件やタイミングで不具合が発生しないことを検証するために、仮想マシンモニタ で指定したタイミングで I/O やレジスタの値を強制的に変更する。これにより、仕様では規定されているものの通常はめったに発生しない状態遷移を意図的に引き起こし、デバイスドライバが正しく動作することを検証する。

(6) 仮想マシンモニタによるストレージクラスメモリへのアクセスの捕捉：最初のステップとして、SCM の暗号化をおこなうシステムを提案する。このシステムでは、OS から透過的に SCM への読み書きを捕捉して、暗号化や復号をおこなう。オーバーヘッドを大きくしないようにしながら SCM へのアクセスを捕捉するために、仮想マシンモニタはバイト単位ではなくページ単位での捕捉をおこなって暗号化や復号をおこなう。障害発生時の一貫性を維持するために、永続ページバッファという概念を導入して、ページの内容や暗号化の進捗状況などのデータを保持するようにする。

４．研究成果

(1) 仮想マシンモニタによるシステム管理：本研究では、BitVisor 1.4 をベースとして仮想マシンモニタの実装をおこなった。BitVisor のコア部分の変更点は 3,576 行であった。また、IDE や AHCI, Intel PRO/1000, X540, Realtek 816x, Broadcom NetXtreme などのデバイスに対する実装をおこなった。また、ATA over Ethernet (AoE)プロトコルを拡張したネットワーク・ストレージプロトコルの実装もおこなった。実験の結果、Windows や Linux を修正なしで導入できることを確認した。提案手法では、イメージコピーを用いた場合とくらべて 8.6 倍速く導入することが可能であったほか、データベースの性能を測る実験では、KVM に最新の研究成果である Exit-less Interrupt(ELI)を導入したものと比べても、同等の性能を達成できることを確認した。脱仮想化のあとは、ゼロ・オーバーヘッドを達成できていることも確認した。研究成果は、著名な国際会議 ASPLOS 2015 で発表した。

(2) 仮想マシンモニタによる物理マシン上での OS ライブマイグレーション：本研究は BitVisor をベースに実装をおこなった。我々の実装では、Programmable Interrupt Controller (PIC) や Programmable Interval Timer (PIT)、Realtek 8169 などの最小限のハードウェアを装備したマシン上において、最小構成の Linux をマイグレーションすることを実現した。Netperf を用いた実験結果により、スループットやレイテンシに対するオーバーヘッドは無視できるほど小さいことがわかった。研究成果は、国際ワークショップ APSys 2014 において、ポスター発表を行った。

(3) 仮想マシンモニタによるシステムイメージの完全性保護：提案手法に基づいて、仮想マシンモニタとして BitVisor を使用して、OS から透過的にバイト粒度でのストレージ保護を実現するシステムを実装した。また、実際に FAT32 上で動作する WindowsXP を対象として、セキュリティポリシーの作成をおこない、実際にシステムイメージの完全性が保護されていることを確認した。研究成果は、情報処理学会論文誌に論文として掲載された。

(4) 仮想マシンモニタによるボランティアコンピューティング基盤の実現：提案手法に基づいてボランティアコンピューティング基盤のプロトタイプの設計と実装をおこなった。また、組織側の計算コードのために専用 CPU コアを割り当てる方式を実装した。CPU のホットプラグ機能を利用して、ゲスト OS から CPU コアを一つ hot remove して、組織側の計算のために使用する。専用コアを割り当てることにより、スケジューリングなどシステムの実装が簡便になるほか、ボランティア側と組織側で相互に性能が影響しあうこともなくなるという利点がある。実装したプロトタイプを用いて予備的な実験をおこなった。モンテカルロ法により円周率計算をおこなうプログラムを実装し、仮想マシンモニタに制御が戻って来るたびに一定回数の計算をおこなうようにした。実験の結果、通常のシステム上で計算した場合には 23.79 秒かかったのに対し、提案システム上では 16.88 秒しかかからなかった。これは、CPU コアを専用に割り当てることで割り込みなどの影響が発生せずに、計算コードのみを全力で実行するようになったためと考えられる。研究成果は、国際会議 APSys や国内会議 DSW 2013 でポスター発表をおこなった。

(5) 仮想マシンモニタによる物理デバイスドライバ検証環境の実現：Linux の既知のデバイスドライバのバグとして、ATA デバイスドライバの不具合を用いて、実際にバグが再現するかどうかの実験をおこなった。このバグは、デバイスがビジー状態であるかどうかを判定するコードに関係するものである。具体的には、特定のコントローラの busy 状態を表す値が、他の種類のコントローラにおいても busy 状態として処理されているために、デバイスのリセット時に ATA デバイスが初期化出来なくなるというものである。このバグが再現するように準パススルー型仮想マシンモニタである BitVisor をベースとして、デバイスの状態遷移を変更するシステムを実装して実験をおこなった。その結果、バグのあるカーネルではバグ

が再現することが確認された。また、他のバグがないバージョンや Windows や FreeBSD など他の OS でも試してみたところ、同様のバグは存在していないことがわかった。研究成果は、国内会議 DSW 2013 でポスター発表をおこなった。

(6) 仮想マシンモニタによるストレージクラスメモリへのアクセスの捕捉：本システムは、BitVisor をベースに実装をおこなった。実験にはオープンソースのメモリファイルシステムである PMFS を用いて、SCM の動作をメモリで模倣することでおこなった。我々のページ単位での暗号化をおこなうと、４KB レコードの読み書きのスループットは 5%から 50%程度低下することがわかった。研究成果は、SWoPP 新潟 2014 で発表をおこなった。

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１件）

① 忠鉢 洋輔，表 祐志，品川 高廣，加藤 和彦．軽量ハイパバイザによるシステムイメージの完全性保護．情報処理学会論文誌，第 54 巻，第 12 号，2402-2412 頁，2013 年 12 月．http://id.nii.ac.jp/1001/00096737/

〔学会発表〕（計７件）

① Yushi Omote, Takahiro Shinagawa, Kazuhiko Kato. Improving Agility and Elasticity in Bare-metal Clouds. In Proceedings of the 20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2015), Istanbul TURKEY, pp. 145-159, Mar. 16 2015.

② 表 祐志，品川 高廣，加藤 和彦．Hypervisor-based Interposition Framework for Storage-Class Memory．2014 年並列／分散／協調処理に関する『新潟』サマー・ワークショップ（SWoPP 新潟 2014）．情報処理学会研究報告，第 2013-OS-130 巻，情報処理学会，朱鷺メッセ（新潟市中央区），2014 年 7 月 28 日．

③ 品川 高廣，榮樂 英樹，松原 克弥．iMac での Windows のスケジュール起動．大学 ICT 推進協議会 2013 年度年次大会，幕張メッセ国際会議場（千葉県千葉市），2013 年 12 月 18 日．

④ 深井 貴明，表 祐志，品川 高廣，加藤 和彦．物理マシン間のライブマイグレーション手法の提案．第 127 回システムソフトウェアとオペレーティング・システム研究会．情報処理学会研究報告，第 2012-OS-127 巻，情報処理学会，芝浦工業大学 豊洲キャンパス（東京都江東区），2013 年 12 月 3 日．

⑤ 忠鉢 洋輔，品川 高廣，加藤 和彦．直感的ポリシー設定を可能にする動的な資源隔離機構．第 124 回システムソフトウェアとオペレーティング・システム研究会．情報処理学会研究報告，岡山コンベンションセンター（岡山県岡山市），第 2012-OS-124 巻，情報処理学会，2013 年 3 月 1 日．

⑥ 芹川 大地，表 祐志，品川 高廣，加藤 和彦．VMM による軽量かつセキュアなボランティアコンピューティング基盤の実現．2012 年並列／分散／協調処理に関する『鳥取』サマー・ワークショップ（SWoPP 鳥取 2012）．情報処理学会研究報告，第 2012-OS-122 巻，第 12 号，情報処理学会，とりぎん文化会舘（鳥取県鳥取市），2012 年 8 月 1 日．

⑦ 島田 恭平，表 祐志，品川 高廣，加藤 和彦．VMM を用いた実デバイスドライバの検証環境の設計．2012 年並列／分散／協調処理に関する『鳥取』サマー・ワークショップ（SWoPP 鳥取 2012）．情報処理学会研究報告，第 2012-OS-122 巻，第 12 号，情報処理学会，とりぎん文化会舘（鳥取県鳥取市），2012 年 8 月 1 日．

〔図書〕（計０件）

〔産業財産権〕

○出願状況（計１件）

名称：情報処理装置及びプログラム
発明者：品川 高廣，表 祐志，北村 朋宏，榮樂 英樹，松原 克弥
権利者：東京大学、イーゲル
種類：特許
番号：特許出願 2015-010521
出願年月日：2015 年 1 月 22 日
国内外の別：国内

○取得状況（計０件）

〔その他〕

ホームページ等
http://www.os.ecc.u-tokyo.ac.jp/

６．研究組織

(1)研究代表者

品川　高廣（SHINAGAWA, Takahiro）
東京大学・情報基盤センター・准教授
研究者番号：40361745

(2)研究分担者

加藤　和彦（KATO, Kazuhiko）
筑波大学・システム情報工学研究科・教授
研究者番号：90224493