

戦略的創造研究推進事業 CREST

研究領域「情報社会を支える新しい高性能情報処理技術」

研究課題「ディペンダブル情報処理基盤」

研究終了報告書

研究期間 平成 14 年 11 月～平成 20 年 3 月

研究代表者：坂 井 修 一

東京大学大学院情報理工学系研究科 教授

1 研究実施の概要

■研究の構想（概要）

情報システムのディペンダビリティ(dependability)の確保が大きな課題となっている。ディペンダビリティは、信頼性・安全性・可用性・堅牢性・拡張性などの複合的・総合的な性質である。本研究では、超分散型情報処理環境に必須なディペンダビリティを高度に実現する情報処理基盤を研究開発した。特徴は、(1)アーキテクチャ・ソフトウェアのそれぞれでディペンダビリティ向上の要素技術開発を行うとともに、情報インフラ全体にわたる基盤技術の確立をめざす点、(2)再構成による安全性確保、メモリ操作高信頼化、効率と安全性を高度に高めた暗号処理の導入、サーバの高信頼なスケジューリングなど、アーキテクチャや基礎ソフトウェアの新技术をディペンダビリティの基本要素としている点、(3)CPUアーキテクチャにおいては、ディペンダビリティのためのアーキテクチャ基本要素をディペンダビリティマネージャが呼び出す方式によって、実行系とディペンダビリティ制御の役割を分け、処理効率を落とさずに確実にディペンダビリティを向上するようにできる点、(4)ソフトウェアにおいては、ディペンダビリティのための基本要素をミドルウェアが呼び出す方式によってプログラマとディペンダビリティ管理者の役割を分け、全体として手数少なく確実にディペンダビリティを向上するようにできる点、(5)ネットワーク侵入防止のために、ハードウェアからソフトウェアまでさまざまなレベルで侵入検知システムを提案・試作し、統合・実証する点、などである。

■研究の実施（概要）

本プロジェクトは、アーキテクチャ、サーバ用基盤ソフトウェア、アプリケーション用基盤ソフトウェアの3つの領域のそれぞれで、ディペンダビリティ向上のための基盤技術を研究開発するとともに、これらを統合したシステム技術を開発することが目的である(図1)。その目標は、情報処理の効率を落とすことなく、また、アプリケーションプログラマに負荷をかけることなく、高度なディペンダビリティを実現することであり、そのための基盤技術を研究開発した。

実施の体制として、3つのサイトが3つの領域を受け持つことを基本とした。具体的には2(次節)の通りである。これら3つのサイトの研究者が定期的に討論の場をもち、共同研究のテーマ(侵入検知システム、WWWサーバなど)を決めて協力し、最終的に個々のテーマにおいて国際的水準の成果を出すとともに、統合的・融合的成果をあげることに成功した。

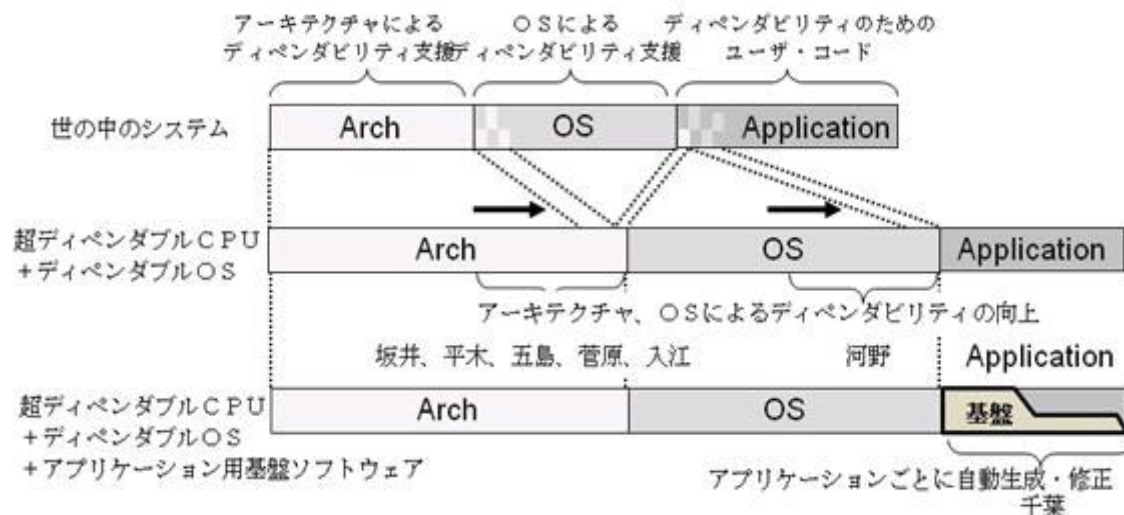


図1 ディペンダブル情報処理基盤プロジェクト

■ 研究成果（概要）

アーキテクチャに関しては、超ディペンダブルプロセッサという形で、近未来に必要となるさまざまな高信頼化・安全化技術を研究開発するとともに、ディペンダビリティマネージャを持ちてこれを統合的に管理する方式を提案・評価・検証した。ここで、要素技術として開発したものは、次のようなものである。

1) 耐タンパ技術

アドレスランダムマイズ技術、耐ソフトウェアタンパ・モード

2) プログラム監視技術

インジェクションアタック検出の高精度化、由来に基づいた情報漏洩保護技術、値範囲伝搬を利用したプログラム静的解析技術

3) 耐エラー技術

組合せ回路における Single Event Transient の検出、縦横パリティによるキャッシュアーキテクチャ、ばらつきを考慮したスペアビットの利用法、レジスタ書き込み時のタイミングエラー検出、エラー回復アーキテクチャ

4) ディペンダビリティ管理技術

メモリ上データのタグ管理方式、柔軟なタグ伝搬アーキテクチャ

また、耐永久故障アーキテクチャについては、ファンクションユニットの回路の固定故障における障害を回避するために柔軟な回路合成の可能である再構成の手法を提案し、評価し、提案手法の有効性を確認した。

高速侵入検知システムの研究では、10 ギガビット/秒を越える速度での TCP ストリーム内容検査をバックボーンネットワークで集中的に行える方式を提案

し、これを実装して有効性を検証した。高度な侵入検知のために、TCPストリームフィルタと呼ばれるソフトウェアを開発し、亜種のアタックなどより狡猾な侵入をも検知できることが示された。

サーバ用基盤ソフトウェアでは、ウェブサーバのディペンダビリティ向上のための諸技術の研究・開発を行ってきた。具体的には、ウェブサーバの性能パラメータの自動調整機構およびウェブサーバの分散ホスティングのための基盤を確立することができた。

アプリケーション用基盤ソフトウェア研究グループでは、アспект指向に代表される最新のソフトウェア技術をディペンダブルシステムの構築に応用する研究をおこなった。従来、ハードウェアやオペレーティングシステムがディペンダブルなサービスを実現する機能を提供していても、アプリケーションプログラムからそれを利用することには複雑なプログラムを書かなければならないことが多かった。これは開発の難易度を上げ、開発コストの点から、それらの機能が十分に生かされないことが問題であった。本研究グループの研究は、ディペンダビリティのための機能を利用するコード（ディペンダブル・コード）を数個のモジュールに集約し、開発者の中の少数のディペンダビリティの専門家によって開発保守ができるようにした。

以上の研究成果は、それぞれ学会・国際会議などで発表して好評を博し、同時に主要な知財について4件の特許申請（うち1件は外国出願）を行った。さらに、研究の中で開発したソフトウェアはオープンソース・ソフトウェアとして公開されており、今後の普及が大いに期待できる。

さらに、CREST シンポジウムなどの機会には、以下のような展示・デモを行った。

(1) 超ディペンダブルプロセッサ

FPGA ボードによって超ディペンダブルプロセッサを試作し、タンパ耐性、アタック耐性、故障耐性のそれぞれについて、具体的な実験・デモを行った。

(2) 高速侵入検知システム

2つの10ギガビット・イーサネットインタフェースを持つ侵入検知システムの実験装置を展示・実演した。

(3) TCPストリームフィルタ

TCPストリームフィルタの入ったPCを展示し、ウェブサーバに対して攻撃を仕掛け、その攻撃メッセージを検出したり遮断したりする様子を見せた。

(4) サーバ用基盤ソフトウェア

ウェブサーバの性能パラメータ自動調整機構について実演した。

(5) アプリケーション用基盤ソフトウェア

開発したアспект指向プログラミング言語GluonJを用いて実装した河川水量モニタシステムおよびTV番組紹介SNSを展示発表した。

2 研究構想及び実施体制

(1) 研究構想

コンピュータとインターネットを中心とする情報システムが重要な社会基盤のひとつとなるにつれ、そのディペンダビリティ(dependability)の確保が大きな課題となっている。ディペンダビリティは、信頼性・安全性・可用性・堅牢性・拡張性などの複合的・総合的な性質である。今の情報処理環境は、アドホックにできあがっている部分が多く、真にディペンダブルなシステムを形成しているとは言い難い。本研究では、近未来の超分散型情報処理環境に必須なディペンダビリティを高度に実現する情報処理基盤を研究開発する。特徴は、(1)アーキテクチャ・ソフトウェアのそれぞれでディペンダビリティ向上の要素技術開発を行うとともに、情報インフラ全体にわたる基盤技術の確立をめざす点、(2)再構成による安全性確保、メモリ操作高信頼化、効率と安全性を高度に高めた暗号処理の導入、サーバの高信頼なスケジューリングなど、アーキテクチャや基礎ソフトウェアの新技术をディペンダビリティの基本要素としている点、(3)CPUアーキテクチャにおいては、ディペンダビリティのためのアーキテクチャ基本要素をディペンダビリティマネージャが呼び出す方式によって、実行系とディペンダビリティ制御の役割を分け、処理効率を落とさずに確実にディペンダビリティを向上するようにできる点、(4)ソフトウェアにおいては、ディペンダビリティのための基本要素をミドルウェアが呼び出す方式によってプログラマとディペンダビリティ管理者の役割を分け、全体として手数少なく確実にディペンダビリティを向上するようにできる点、(5)ネットワーク侵入防止のために、ハードウェアからソフトウェアまでさまざまなレベルで侵入検知システムを提案・試作し、統合・実証する点、などである。

本研究の成果によって、ユーザが真に信頼でき、安全性・性能・機能の諸点でも満足できる情報システムの技術基盤が作られると考えられる。これが確立すれば、商取引や行政などの電子化が一気に進み、信頼性と利便性のともに高い社会をより低いコストで実現できるようになる。政府の提唱するIT国家実現には必須のことであり、医療ネットワーク、防災ネットワーク、遠隔教育ネットワークなどの実現にも必要な技術となる。また、真にディペンダブルなハードウェア・ソフトウェアの創出は、産業的には、従来のインテル/マイクロソフトの次世代のヘゲモニーを狙う可能性を秘めている。特に、利潤構造を示しにくい現在の半導体産業、ソフトウェア産業を活性化するひとつの軸となることが期待される。

以上の構想のもとで、アーキテクチャと侵入検知ハードウェア、侵入検知ソフトウェアとサーバ用基盤ソフトウェア、アプリケーション用基盤ソフトウェアの3つの領域を設定し、

それぞれを、東京大学、慶応大学（研究開始当初は電気通信大学）、東京工業大学が担当することとした。また、これらの統合・融合については、拡大運営委員会などによって定期的に議論し、実際に2つの融合テーマを実現し、成果をあげることができた。

プロジェクト期間を前半・後半にわけ、前半は主に要素技術を掘り下げる時期、後半は個々の領域の研究開発を完成させるとともに、全体統合をめざして研究を拡大していく時期と位置づけた。具体的な研究スケジュールを図2に示す。

年 技術	H14	H15	H16	H17	H18	H19
設備整備	←→			←→		
アーキテクチャ研究グループ(坂井、平木、五島、白石、入江)	← アーキテクチャ提案・シミュレーション →			←→	← アーキテクチャ改良・評価 →	←→
	← コンパイラ基本部開発 →			←→	← 超ディペンダブルチップ試作 →	←→
	← 侵入検知システムの開発・評価 →			←→	← コンパイラ最適化・コード変換 →	←→
				←→	← 超ディペンダブルIDS →	←→
				←→	← 仮想大域サーバ開発 →	←→
サーバ用基盤ソフトウェア研究グループ(河野)	← サーバ用基盤ソフト基本部開発 →			←→		
				← サーバ用基盤ソフト基本部評価 →		
				←→	← サーバ用基盤ソフト最適化・評価 →	←→
アプリケーション用基盤ソフトウェア研究グループ(千葉)	← アプリケーション用基盤ソフト基本部開発 →			←→		
	← WWWアプリケーション評価 →			←→		
				←→	← アプリケーション用基盤ソフト最適化・評価 →	←→
全体統合(全員)					← 大域サーバ技術の統合 →	←→
					← ユーザ向け技術・サーバ技術の統合 →	←→
					← 侵入検知システム統合 →	←→
					← 全体評価・まとめ →	←→

前半3年：方式検討、基本設計、実験システム構築・評価

後半2年：プロトタイプ試作と評価、要素技術の統合

図2 「ディペンダブル情報処理基盤」研究スケジュール

(2)実施体制

図3に本プロジェクトの実施体制を記す。

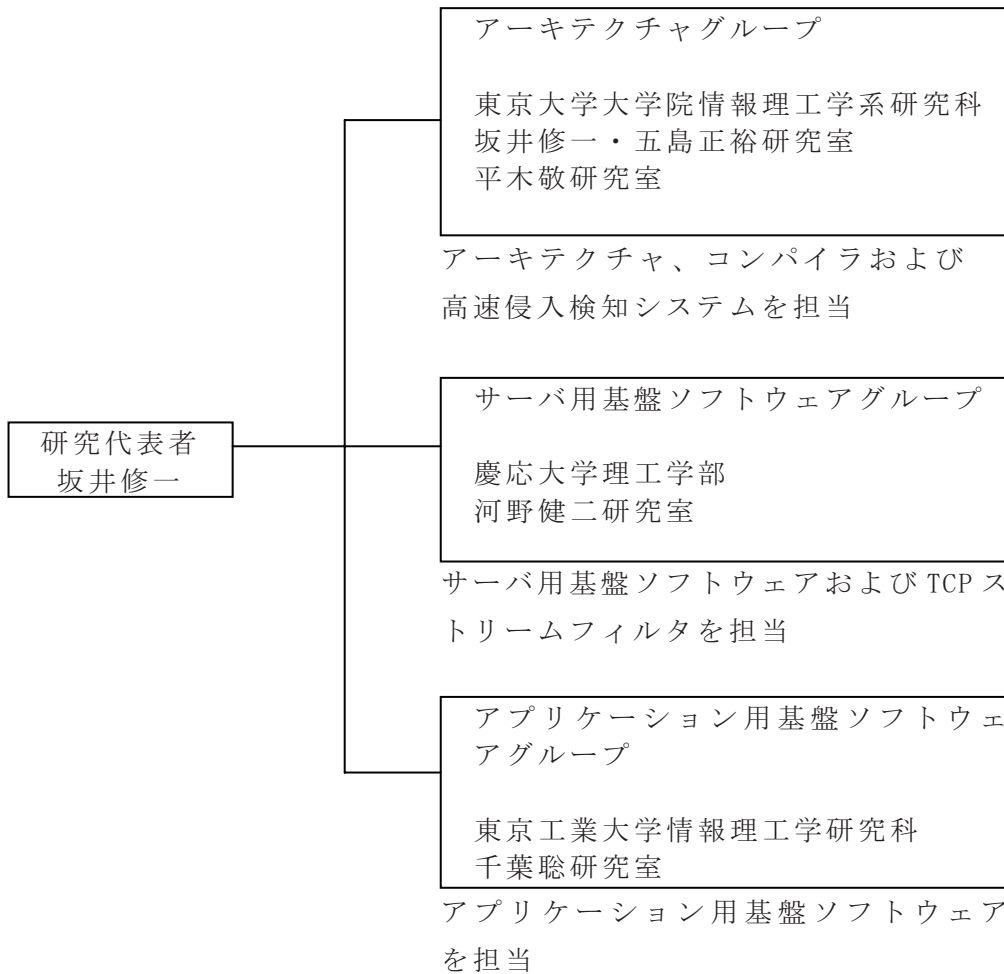


図3 「ディペンダブル情報処理基盤」実施体制

3 研究実施内容及び成果

3. 1 アーキテクチャグループ（東京大学 坂井グループ，平木グループ）

[1] 研究実施内容及び成果

■超ディペンダブルアーキテクチャ

本研究では、コンピュータシステムの中核となるマイクロプロセッサに着目し、プロセッサアーキテクチャ改良によるディペンダビリティ向上を目標とした。アーキテクチャ技術に

よって期待される効果としては、以下のものがある。

- ・プロセッサそのものの信頼性向上
- ・ハードウェアによってはじめて実現できるディペンダビリティ技術(主記憶暗号化など)
- ・ハードウェアとソフトウェアの界面からの制御による、様々な脅威への包括的な対策
- ・汎用的なディペンダビリティ操作をハードウェア化することによる高速化

○研究実施内容

1 命令毎にその実行結果の正当性をチェックしながら動作する、“超ディペンダブルアーキテクチャ”を構想し、その実現に向けた研究を行った。従来の関連研究では、個々の不具合を引き起こす原因に着目し、それぞれに対処するようなアプローチ(顕著な例ではウィルスパターンファイルなど)が主であったが、“超ディペンダブルアーキテクチャ”では、“正当性”のチェックにより、未知の脅威をも対象とする、シンプルで包括的なディペンダブルプラットフォームを提供する。

“正当性”としてチェックされる性質として、本研究では各命令の正確性、機密性、完全性に着目した。これらが守られていれば、どのような外乱や悪意のあるコードの脅威下でも、システムのディペンダビリティを維持することができる。なお、可用性に関しては問題を切り分け、耐故障アーキテクチャの研究を行った。これらは将来の統合を視野に入れている。

本研究では、“正当性”のチェックを可能とするための要素技術を多数提案した。これら要素技術は大きく3つの問題意識に切り分けられている。

- (1) 耐タンパ技術: 命令による操作以外によるサイドエフェクトの防止
- (2) プログラム監視技術: 機密性や完全性を侵す命令を検出、防止
- (3) 耐エラー技術: 命令実行時に生じる物理エラーの検出、防止

これら三要素が統合されてチェックされることにより、包括的な対策となる。

“超ディペンダブルアーキテクチャ”において、命令の実行部分と正当性チェック部分は、分離された実装が可能である。このような構成とすることにより、i) 実行のクリティカルパスを延ばさない高効率実装が可能となる、ii) 高性能追及部分と機能保証部分が別ユニットとなり設計ハザードが緩和される、iii) チェック部分が分離され柔軟なディペンダビリティポリシー実装が可能となる、などの利点が見込まれる。本研究では、このような見通しに基づき、従来のプロセッサに加えられる正当性チェック部分、“ディペンダビリティマネージャ”について提案を行った。

各研究は回路シミュレーション、サイクルシミュレーション、OS エミュレーションなど、技術に応じた様々な粒度で評価を行いながら進められた。最終評価では、エラー回復のための要素技術およびディペンダビリティマネージャは FPGA への実装を行った。また、悪

意のあるソフトウェアへの対策技術のような、OS との協調が重要な要素技術については、OS エミュレータへの機能の実装を行った。

○研究成果

(1) 耐タンパ技術

耐タンパ技術では、命令に書かれていない操作、すなわちプロセス外からのデータの参照を防止する。このようなタンパリングの手段として、外部バスを盗聴・改竄するためのハードウェアや、ルート権限を悪用して他プロセスのメモリ内容を盗聴・改竄するソフトウェアが用いられている。

タンパリングに対する技術として、バスを暗号化して盗聴・改竄を無効とする耐タンパプロセッサが既に研究されている。本研究では既存の耐タンパプロセッサを補完する技術を提案した。

1.1) アドレスランダム化技術

従来技術では行われていなかった、メモリアドレス秘匿技術を提案した。従来の耐タンパプロセッサでは、メモリ上のデータは暗号化されるが、そのアドレスは平文の場合と変化しない。しかし、アドレスのアクセスパターンは情報漏洩につながる covert channel のひとつと考えられており、例えば、暗号鍵のヒントを与えてしまうような危険性が指摘されている。

本技術では、キャッシュからメモリにライトバックする際に書き込みアドレスを変換し、アクセスパターンの特徴を低減する。変換はキャッシュラインの粒度でおこなわれる。変換前と後のアドレス対応表はツリー状に分割され、メモリ上で管理される。本手法により、キャッシュミス時のメモリアクセスレイテンシが増加することになるが、アドレス対応表の一部をチップ内にキャッシングすることにより、影響をほとんど抑えられることを確認した。

1.2) 耐ソフトウェアタンパ・モード

従来の耐タンパプロセッサでは、メモリアクセスの度に暗号化、複合化、完全性チェックが行われる。このオーバーヘッドは無視できる量ではなく、特に完全性チェックはピン帯域の使用率を数倍にまで高めてしまう。しかし、もしハードウェアによる盗聴・改竄の恐れがない場合、暗号化を用いずともメモリコントローラによる制御だけで、OS 権限を悪用したタンパリングを防げるはずである。

ここではこのような見通しの下、OS 権限を悪用したタンパリングに焦点をあてた、オーバーヘッドの少ない耐ソフトウェアタンパ・モードを提案した。バスの盗聴やチップを用いたタンパリングが高度な技術を要し、攻撃者自身のシステムに対してしか行えないことに比べ、ソフトウェアによるタンパリングは既存のツールを利用して世界中のシステムに対し行うことができる。このため、ソフトウェアタンパに特化してオーバーヘッドを削減したモードを活用する機会は多いと考えられる。

提案技術では、ページテーブルによって管理されるアクセス制限を強化し、他プロセスからの参照を禁止する属性を追加した。この属性が ON となっている場合には、プロセッサのメモリコントローラによって、ルートといえども参照することが禁止される。

また OS を悪用することによってこの制御を無効化する 3 つの経路、i) コンテキストスイッチ、ii) 二次記憶とのスワップ、iii) ページテーブル改竄に着目、それぞれの対策を行った。

提案技術では、ページテーブルを管理する secureTLB および二次記憶と主記憶のアクセスをハードウェアによって制御する secureDMA を追加した(図4)。シミュレータによる評価を行い、暗号化を用いる従来の耐タンパ技術と比較して大幅にオーバヘッドを軽減できることを確認した(図5)。

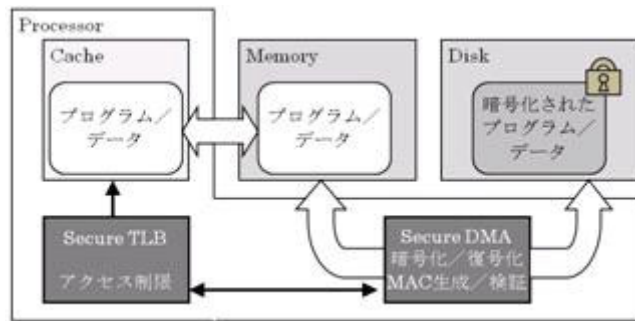


図4 耐ソフトウェアタンパ・プロセッサのブロック図

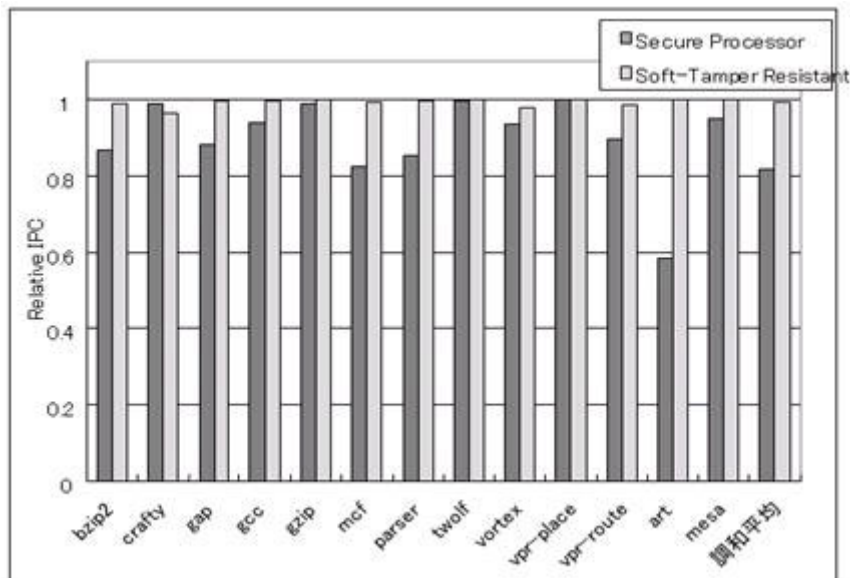


図5 従来のセキュアプロセッサとの性能比較：
(タンパ対策をしない場合の性能を1とした相対 IPC 値)

(2) プログラム監視技術

プログラム監視技術では、システムの機密性や完全性を脅かす“危険な”命令を検出して例外を発生させる。ここで想定されるのはトロイの木馬のような悪意をもって作成されたプログラムや、脆弱性を利用して侵入してきたコード、違法複製などのためにローカルユーザが起動したプログラムなどである。

一般に、与えられた命令の危険性をプロセッサ自身が判断することは困難である。この問題に対して本研究では、プログラムとは別にセキュリティポリシーを与え、ポリシーに反する実行をした場合に危険な命令として検出するアプローチに着目、大きく3つの新技術の提案・評価を行った。ポリシーはデータの由来に着目して、例えば“外部入力に由来するデータが命令として用いられてはならない”などのように定められる。データの由来を解析するため、全てのデータパスには由来をあらわす“タグ”ビットが追加される。タグはプログラムの演算に沿ってソースからディスティネーションへと伝搬し、由来を保持する。

2. 1) インジェクションアタック検出の高精度化

タグによる危険命令検出の関連研究は、主に脆弱性を利用したインジェクションアタック検出を目的としている。しかし、これらの関連研究ではタグの伝搬および検出規則は“比較命令があればサニタイズと見なしてタグを消去する”等、個々の脆弱性に基づいたヒューリスティクスによって定められており、その検出精度は低い。

本研究では、ヒューリスティクスではなく、C言語のモデルに基づき、“入力由来のデータがそのままアドレスとして用いられることはない；ただし、システム由来のアドレスの種と演算されてアドレスを生成することはありうる”というポリシーを設定し、伝搬および検出規則を提案した(図6)。OS エミュレータによる評価により、本手法は false-positive や false-negative を劇的に減少させることを確認した。

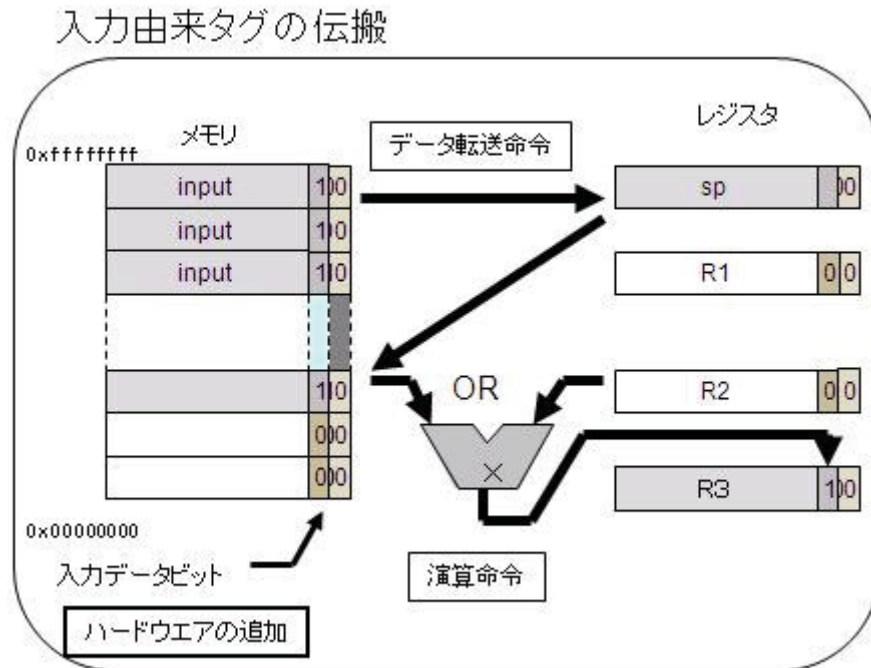
2. 2) 由来に基づいた情報漏洩保護技術

本研究では、由来に基づいた新しい情報保護技術を提案した(図7)。提案手法では、データと使用許可条件を対にした構成とした上で、タグによる情報フロー追跡を行い、出力(システムコール)時にタグからデータの使用条件を得て確認する。例えば、ネットワークに流出させたくない個人情報データに、“ネットワークI/Oへ出力しない”という使用条件を設定する。この後、誤って起動したトロイが重要データを読み込んでネットワークへ出力しようとしても、読み込んだデータにタグが付加されており、プログラム実行とともに伝搬し、出力データにタグが付加されるため、使用条件と SYSCALL が比較されて出力は失敗する。

本手法の重要な点は、データを扱うプログラムの側では情報保護に留意しなくて良いことである。これによりソフトウェア開発の敷居をさげる効果が期待できる。またネットワーク上で配布される信頼できないプログラムを安全に使用することが可能となる。

本手法における主要な技術的課題の一つは、“implicit information flow”の追跡で

ある。単純に命令のソースからディスティネーションにタグを伝搬させる方法のみの場合、攻撃者は狡猾なプログラムを書くことにより、タグを伝搬させずに情報だけを得ることができる。具体的には、タグの付加されたデータを元に分岐が行われると様々な漏洩ルートが発生する。



入力由来のデータをデータパスで追跡することにより危険を検出する。従来手法に加えてシステム由来であることを示すタグが加わっており、2ビットのパターンで検出を行う。

図6 インジェクションアタック検出手法：

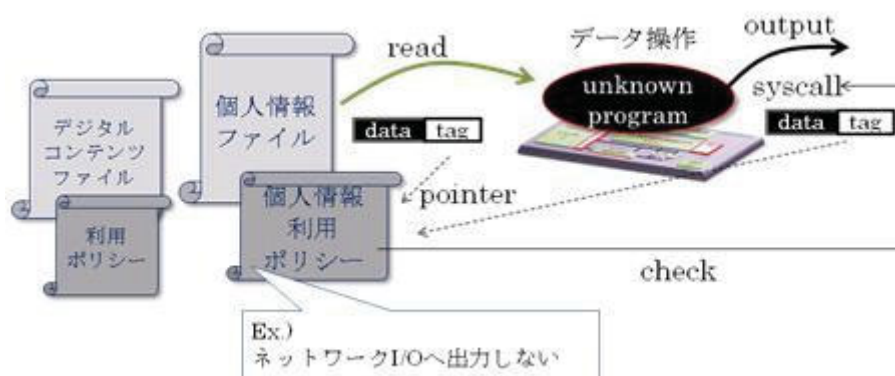


図7 由来に基づいた情報漏洩防止技術

本研究ではこの問題に取り組み、従来技術では未解決であった implicit information flow 追跡を可能とする手法を提案した。Implicit information flow による漏洩が発生する場合には例外が発生する手法(軽量だが、停止することにより情報が漏えいする)及び、ハードウェアによる巻き戻り制御を利用して implicit な漏洩ルートを塞ぐ手法の二手法の提案を行った。

2. 3) 値範囲伝搬を利用したプログラム静的解析技術

本研究では information flow を応用したデバッグ支援技術を研究した。一般に、バグはプログラムの想定と実際の挙動可能性が乖離したときに発生すると考えられる。そこで本研究では、挙動可能性 \asymp 変数の取りうる値範囲の広さと仮定し、広すぎる値範囲を持つ変数がメモリやシステムコールなどの重要操作の引数として渡される時に警告を発するデバッグ技術を検討した。変数の値範囲は、プログラム実行の流れにそって、各変数のとりうる値の最小値と最大値を伝搬させて解析する。

(3) 耐エラー技術

従来の半導体チップは比較的耐性の高い部品であった。しかし、近年の微細化によって、半導体チップのエラー脆弱性が増していることが指摘されている。本研究では、従来から研究されているソフトエラー対策に加え、新たな問題として浮上してきた製造ばらつきによるタイミングエラー対策に着目した。

特に、タイミングエラー対策技術を活用すると、従来の安全マージンを含めた動作ではなく、その時点で真に安定動作しうるぎりぎりの動作電圧/周波数に自動的に適応動作することが可能となる。現在、熱分布の偏りはプロセッサの効率を制限しており、本技術による適応動作は今後非常に重要となっていくと考えられる。

3. 1) 組み合わせ回路における Single Event Transient の検出

放射線が組み合わせ回路を直撃すると、鋭いパルスが発生する(SET)。このパルスをラッチが取り込むと誤動作に繋がってしまう。提案手法では、組み合わせ回路の出力を、異なるタイミングで動作するラッチで取り込んで一致比較することにより、SET パルスによる悪影響を検出する。

3. 2) 縦横パリティによるキャッシュアーキテクチャ

従来はシングルビットのエラーと考えられてきたソフトエラーだが、微細化によるエラーマージン縮小のため、今後は隣接したビットを巻き込んだマルチビットエラーとなる可能性が指摘されている。提案手法では縦方向と横方向のパリティをキャッシュに付加し、シングルイベント・マルチビットエラーを検出する。

3. 3) ばらつきを考慮したスペアビットの利用法

製造ばらつきの激化により、キャッシュを構成するメモリセルに不良セルが混入する確率が増加している。提案手法では、各ライン毎 BIST 時に不良ビットを検出し、その数に応じてラインの利用方法を変更する。不良セルが少ないラインは ECC によるソフトウェア回復が可能であり、通常の使い方をする。一方、不良セルの多いラインは ECC によるエラー回復が不可能であるため、もしそのラインにソフトウェアが発生した場合大きな回復ペナルティとなる。そこで、不良セルの多いラインが dirty 状態になった場合はすぐにメモリへの描き戻しを行う。ばらつきとソフトウェア頻度を見積もった評価により、このような使い分けをする方が不良セルの多いラインを disable するよりも効率的であることを確認した。

3. 4) レジスタ書き込み時のタイミングエラー検出

入力値によってアクティブとなるパスは変化し、また熱によっても遅延は変化するため、激化する製造ばらつきとも相まり、実行時にクリティカルパス部分でタイミングエラーが生じる可能性が高まっている。RAZOR に代表されるような、既存のタイミングエラー検出手法では、メモリユニットへの書き込みを検出することができない。そこで、提案手法では一度書いた値を再び読み出して比較することによって書き込みを検証する。

再読み出しのオーバーヘッドを隠ぺいするためのアーキテクチャ技術を提案(図8, 9)し、シミュレータ評価により殆ど性能低下なく実装が可能であることを確認した(図10)。

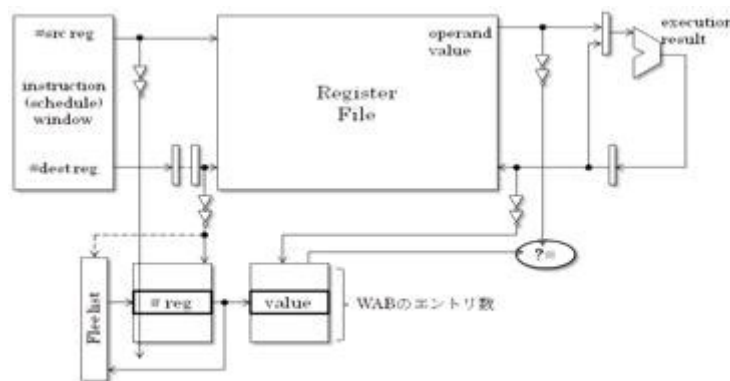


図8 書き込み保障バッファ(WAB)の実装(passive 方式)

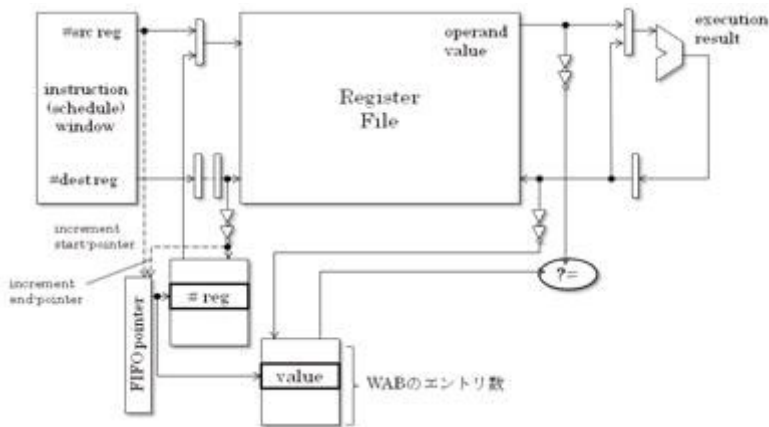
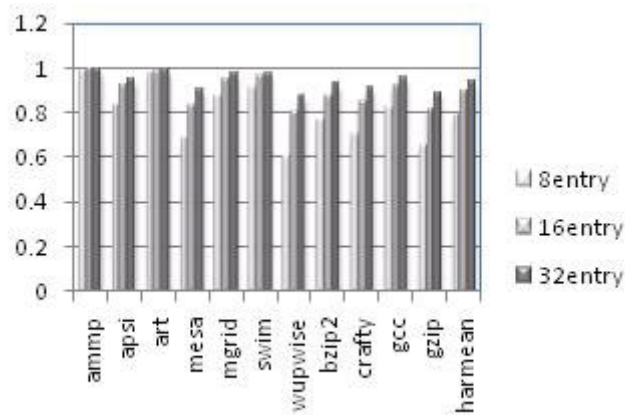
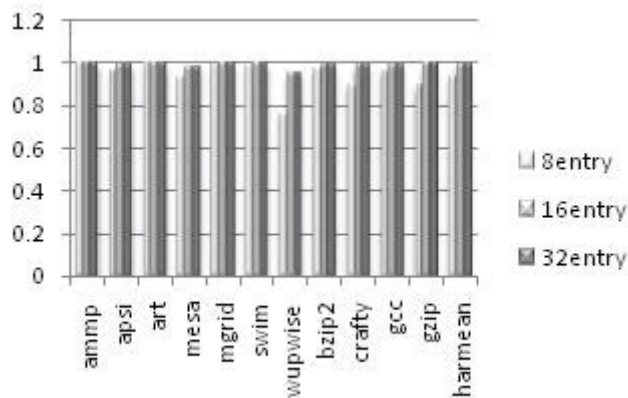


図9 書き込み保障バッファ(WAB)の実装(active方式)



(a) passive 方式



(b) active 方式

図10 書き込み保証技術適用時の性能(相対IPC)

3. 5) エラー回復アーキテクチャ

エラー検出のための様々な要素技術を利用し、タイミングエラーを検出、回復するアーキテクチャを提案した。提案手法では各ユニットのエラー検出情報を伝搬し、エラーを

生じた命令のリタイアを確実に止めることにより、アーキテクチャステートを保護する。

(4) ディペンダビリティマネージャ

以上挙げた要素技術の実装の中では、タグの管理、伝搬処理が、もっとも効率を要求され、各技術で共有される部分となる。本研究では、包括的ディペンダブルアーキテクチャとして、メモリ上データに付加されるタグの管理方式と、タグ伝搬処理を効率的に行うアーキテクチャについて提案を行った。

4.1) メモリ上データのタグ管理方式

タグが1バイトに対し1ビット付加される場合、ナイーブな実装ではメモリ容量が12.5%増加することとなる。ポリシーが複雑化あるいは複数のポリシーを適用する場合、このオーバーヘッドは更に増加するため、ポリシー実装の敷居を高くしてしまう。

そこで、本研究では階層型のテーブルを用いてメモリ上のタグ容量を圧縮管理する手法を提案した(図11)。連続領域には同じタグが並ぶと考えられ、ある階層の粒度で全て同じタグを持つ場合には、1エントリでタグが管理できることになる。本研究では動的に圧縮、縮約が可能なハードウェアと、階層アクセスのレイテンシを隠蔽するキャッシュを提案し、タグ容量の大幅な低下を実現した。

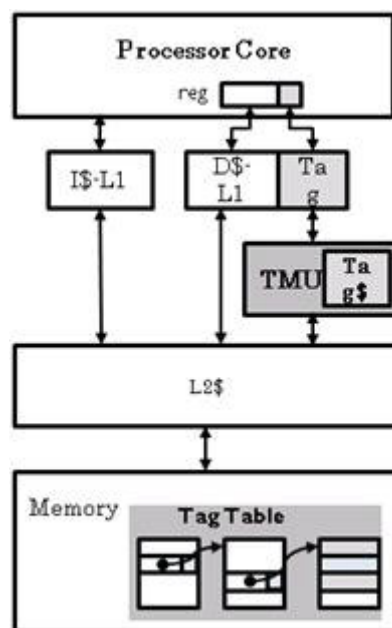


図11 メモリ上データのタグ圧縮管理方式

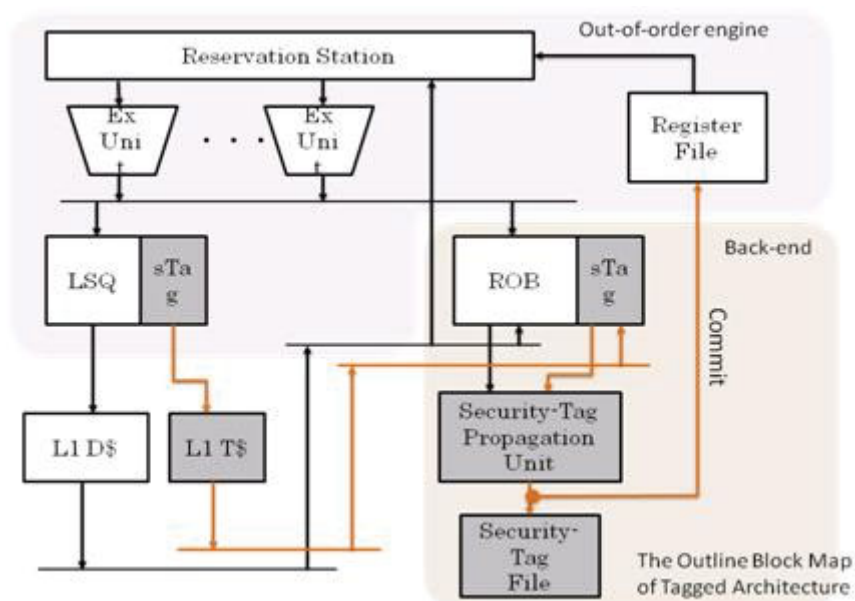
4.2) 柔軟なタグ伝搬アーキテクチャ

従来のタグ関連技術では、データパスのビット幅を拡張してタグ伝搬を行うアーキテクチャが想定されているが、この構成では柔軟さ、高速性、設計の複雑さなどに困難が予

想される。そこで本研究では実行部分とタグ伝搬・検出部分を切り離すアーキテクチャを検討した(図 12)。まず、プロトタイプとしてリタイア後に in order にタグ伝搬を行うアーキテクチャを提案・評価した。

○ 類似研究との比較

アーキテクチャの利点を活かし、包括的なディペンダビリティを目指す研究は少なく、本研究の特色と言える。各要素技術については、海外の最近の類似研究と比較しても改善されたものとなっている。概要を表1に示す。



色塗りされた区画が追加部分。実行部分とチェック部分は分離されており、特に実行のクリティカルパス部分に干渉しない

図 12 FPGA 上に実装したタグ伝搬アーキテクチャテストベッドのブロック図：

表1 類似研究との比較

	ディペンダビリティの改善	オーバーヘッドの軽減
耐タンパ技術 AEGIS(MICRO03)	アクセスパターンの秘匿	耐ソフトウェアタンパモードにより、メモリアクセス遅延とメモリ帯域を軽減
プログラム監視技術 DIFT(ASPLOS04) RIFLE(MICRO04) PTD(DSN05) Raksha(ISCA07)	のっとり検出高精度化 Implicit data flowが追跡可能	メモリに占めるタグ容量圧縮 L2hit率改善
耐エラー技術 RAZOR(MICRO03) DIVA(DSN01) Engineering OC(DSN05)	RAM書き込み時のタイミングエラー回復	

■ 耐永久故障アーキテクチャ

近年、半導体プロセスのばらつき増大、発熱の増大による劣化、エレクトロマイグレーション等の要因によりトランジスタの長期的な信頼性を確保することが徐々に困難になりつつある。このように永久故障が生じやすい状況の下でプロセッサの信頼性を確保する事が可能な耐故障化方式が求められる。本研究はその中でプロセッサの実行ステージの耐故障化を目的とする。

○ 研究実施内容

従来は実行ステージの耐故障化方式として回路の多重化、回路の代用などが提案されてきた。前者は同じ機能を持つ回路を複数実装し、そのうちいずれかで故障を検出した場合は残った正常な回路を用いて動作を続けるという手法である。実行ステージの場合は予備の演算器を持ち、使用中の演算器が故障した場合は予備の演算器に切り替えて実行を続ける。後者はある回路が壊れると別の回路を用いてその機能を代替するという手法である。例えば、OR 演算を行う回路が故障した場合はマイクロコードの差し替えにより AND と NOT 演算を用いてエミュレートする。前者の手法は回路の複製によるチップ面積の増大という欠点を持つ。キャッシュメモリなど規則的なレイ構造からなる機構では少数の予備エントリを追加すれば効果が得られるが、実行ステージは不規則な組み合わせ回路からなるために部分回路ではなく全体の複製を行うためである。後者の手法は故障した回路の機能を他の回路で代替する場合に速度性能が大幅に低下するという欠点を持つ。これらの限界を克服するため、本研究で実行ステージに再構成可能なハードウェアを用いることにより耐故障化を行う方式を提

案した。本方式によりチップ面積のオーバヘッドを少なく抑え、かつ故障発生後の性能低下を少なく抑える事を可能とする。

○研究成果

実行ステージの耐故障化を実現する方法として本研究では(1)再構成ロジックアレイを用いた実行ステージ、(2)耐故障演算器とそうでない演算器を組み合わせる再構成可能実行ステージを構成する HFU (Heterogeneous Functional Unit)、の2つの手法を提案した。アーキテクチャの研究で広く用いられている SimpleScalar シミュレータを用いて実行速度の評価を行った。また、再構成ハードウェアのゲートサイズ見積もりには実際の FPGA に対して提案するロジックを合成し評価を行った。

(1) 再構成ロジックアレイを用いた耐故障実行ステージ

本方式ではプロセッサの実行ステージを FPGA のような再構成ロジックアレイを用いて構成する。ロジックアレイを用いて実行ステージに必要な演算器のロジックを構成してプロセッサの機能を実現する。ロジックアレイの一部が壊れた時にはその部分を使わずに演算器のロジックを構成し直し実行を継続する。故障の規模が大きいため正常に動くロジックアレイ資源が減りすぎた場合は、構成する演算器ロジックを縮小して再構成を行う。演算器ロジックの縮小では、重要性が低いと判断される演算器を構成しないことによりロジックアレイ資源の使用量を減らす。この場合、再構成以前に比べ欠落する機能が存在することは必然であり、何らかの形で機能を補完する必要がある。これに関しては前述の回路の代用を用いる。そのため本手法は代用の手法を含む。

本方式の利点は演算器ロジックを自由に組み立てることによる高い柔軟性である。実行に必要な不可欠な演算器を構成する部位が故障した場合には他の重要性の低い演算器を失う代わりに機能を元通りに回復させることが出来る。故障箇所が増えるに従って演算器の規模が減少するため徐々に性能が低下するが、重要度の高い演算器を残すことにより性能低下を低く抑える事が可能である。

シミュレータを用いて提案方式の評価を行った。評価には 32 ビットの RISC アーキテクチャを用いた。命令発行幅は演算器が性能に与える影響を把握しやすいよう 1 個/サイクルとし、演算器を 1 つ設けた。演算器を含む実行ステージを耐故障化の対象とした。実行ステージについて評価を行うため、それ以外の部分は故障しないと仮定した。実行ステージは 3 入力・1 出力からなる Lookup Table (LUT) を基本単位とするロジックアレイで構成されるものとした。

再構成手法を行う際には残存するロジックアレイの個数に応じてより重要な演算器のみを選択し、そうでないものについては回路の代用の手法を用いて機能を補完する。その際に演算器を選択する基準を決める必要がある。評価では、

より多くのロジックアレイを占有している演算器から順に取り除くという戦略をとり、表 2 に示すとおりに演算器を構成した。

表 2. ロジックアレイ残存率と構成する演算器の組合せ

	ロジックアレイ残存率	構成する演算器の組み合わせ
1	100-60%	32bit Shift, 32bit add, sub, Logical
2	60-31%	16bit Shift, 32bit add, sub, Logical
3	31-20%	8bit Shift, 32bit add, sub, Logical
4	20-13%	4bit Shift, 32bit add, sub, Logical
5	13-7%	2bit Shift, 32bit add, sub, Logical
6	7-4%	2bit Shift, 16bit add, sub, Logical

本研究では提案するシステムにおける実行を追跡するためのシミュレータを作り、その上でベンチマークプログラムを実行した。各プログラムにおいて実行時間を測定した。実行ステージが性能に与える影響を評価するために以下のベンチマークプログラムを用いた。

MATRIX: 整数正方行列の積の演算である。これは和積の演算が多用される

PRIME: 素数判定の計算である。和、積、商の演算が多用される

FIBBONACCI フィボナッチ数の演算である。加算が多用される

以上の条件で評価したベンチマークプログラムの実行時間を図 13 に示す。図において、横軸は故障せずに正常に動作しているロジックアレイセルの割合を示す。縦軸は全く故障が無い場合、すなわちロジックアレイが 100%残存している場合を1とした場合のプログラムの相対実行時間である。数値が小さいほど性能が良い事を示す。例えば正常に機能しているロジックアレイセルが全体の 13%である時、Matrix と Prime のプログラム実行時間は故障が無い場合の約 2.2 倍であることを示す。

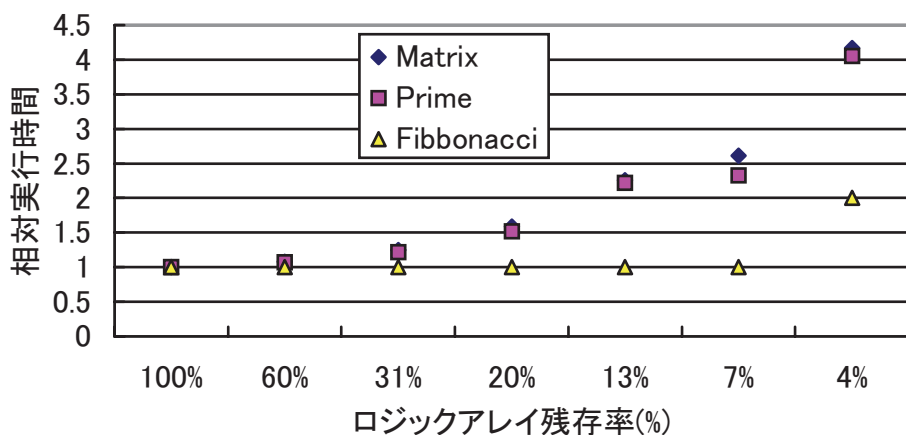


図 13 ロジックアレイ残存率と実行速度

グラフではロジックアレイの故障と共に速度性能の低下が見られる。ただし速度性能の低下の割合はプログラムに依存する。Prime、Matrix においてはロジックアレイセルが故障し失われるにつれ徐々に速度性能が低下する。しかし基本的な命令である加減算が中心であるフィボナッチ数列においては 93%のロジックセルが失われてもなお故障が無い場合と同等の速度を保つ。評価の結果では 70%のロジックセルが失われた場合の実行時間は故障が無い場合と比べ 1.5 倍以内に抑えられている。これはロジックアレイが加算、減算、論理演算、シフト演算に均等に割り振られているのではなく、それぞれを構成するために必要なロジックアレイ量が大きく異なることに由来する。この評価では使用頻度が低いシフトユニットの機能を削ることによりこのような結果が得られた。

本項目ではファンクションユニットの回路の固定故障における障害を回避するために柔軟な回路合成の可能である再構成の手法を述べた。またその障害回避に伴う速度性能の変化についてシミュレーションにより評価を行った。従来の多重化や代用などの手法に比べより低いレベルからの復帰を行うこの手法は速度の大きな低下をもたらさないという点で有用であることが示された。

(2) HFU: ヘテロニアス演算器構成による高耐故障実行ステージ

HFU(Heterogeneous Functional Unit)は耐故障演算器とそうでない演算器を併用することで速度性能・面積効率・耐故障性を同時に保つことを目的とした方式である。再構成による実行ステージの耐故障化を適用した場合、演算組み合わせ回路の伝播遅延が長いという欠点がある。これは、LUT を使う事によるゲート遅延の増大と、再構成可能なネットワークを用いることによる配線遅延の増大のためである。演算のスループットはパイプライン化により保つ事が可能であるが、レイテンシは長くなってしまふ。その結果、プログラムの実行速度が低下する。HFU では耐故障化されていないがレイテンシが短い演算器(以下、高速演算器と呼ぶ)を併用することでこの欠点を補う。本方式では、

(1)高速演算器により非故障時の演算レイテンシを短縮すること、(2)高速演算器が故障した場合にはレイテンシは増えるが耐故障演算器によりスループット(命令発行数)は維持すること、の 2 つの方針に従い制御を行う。(1)は特に逐次的なデータ依存関係を持つ命令列を含むプログラムに対して性能に与える影響が大きい。

本方式ではベースライン・アーキテクチャとして二重実行を用いたスーパースカラー方式を利用する。二重実行とは、本来の計算に加えて故障検出のための計算を別の演算器で並列に行うことである。二つの演算器の出力結果が異なる場合、故障と判定する。故障と判断された場合は該当命令以後の結果を取り消し再実行する。実行と検出演算器の出力結果が何度も異なる場合は永久故障と判定する。この場合は故障を回復するために演算器の接続を再構成し、プログラムの実行を再開する。

本方式では高速演算器と耐故障演算器を組み合わせ、二重実行をプロセッサの命令発行数分並列に行う。故障に備えて耐故障演算器に用いるハードウェアリソースは余分に用意しておく。非故障時にはレイテンシ削減のため演算のクリティカルパスになる主演算に高速演算器を優先的に割り当てる。高速演算器が故障した場合は機能を回復させることは出来ないので、その演算器全体の使用を中止して耐故障演算器で代替する。耐故障演算器が故障した場合は再構成による機能回復を行う。この制御により演算のレイテンシは増えるがスループットは維持される。演算器の接続を変更するための再構成可能な配線を持たせる。故障が増えて十分な個数の耐故障演算器を構成できなくなった場合は同時命令発行数を減らす。故障が続き最低1命令を発行できるだけの演算器が構成できなくなるまで動作を継続することが可能である。

SimpleScala3.0d シミュレータを用いて提案方式について実行速度性能の評価を行った。まず、高速演算器によるレイテンシの削減と、耐故障演算器によるスループット維持がどの程度性能に効いているかを評価する。次に、二重実行を含めた場合の実性能を評価する。ベンチマークには SPEC2000 を用いた。モデルとなるプロセッサは 4 命令を同時発行可能なスーパースカラアーキテクチャである。命令フェッチキュー、リザーベーションステーション、ロードストアキューのエントリ数はそれぞれ 16、128、64 とした。耐故障演算器は高速演算器とスループットは同じで 3 倍のレイテンシを持つとする。命令発行数に対して高速演算器が少ない場合、どの命令に高速演算器を割り当てるかトレードオフが生じる。この点については本研究では議論の範囲外とし、到着順に演算器を割り当てるものとする。高速演算器の故障状態は表 3 のように残存率 100%~25%をモデル化した。

表3 高速演算器の故障状態

FU types	高速演算器残存率			
	100%(故障なし)	75%	50%	25%
IADD	4	3	2	1
IMUL	2	2	1	1
FADD	4	3	2	1
FMUL	2	2	1	1

まず、実行ステージが耐故障化されていない場合の演算器残存率と性能の関係を図14に示す。故障が無いときの性能を1とした相対実行速度を示す。故障する演算器が減るにしたがって性能が低下している。特に演算器が残り1つになった状態では非故障状態と比較してスループットは半分程度に低下している。

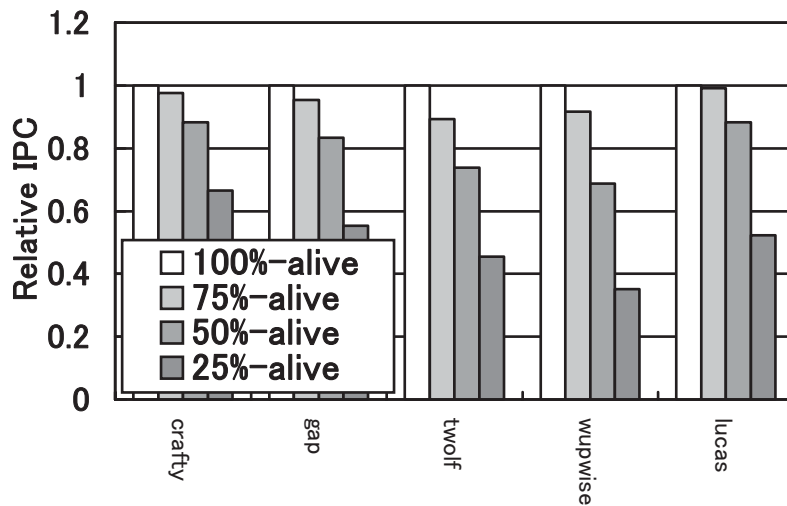


図14 演算器残存率と性能

次に、故障した高速演算器を耐故障演算器で代替した場合の性能を図15に示す。0%-alive は、全ての高速演算器が故障した状態を示す。図では、高速演算器が全て故障した場合に非故障時の約80%の性能が得られている。このことから、耐故障演算器によりスループットを保つことで性能の低下を20%程度に抑制できる事が分かった。また、高速演算器が1個残っている場合、twolf以外では性能低下を10%以下に抑えられている。耐故障演算器で代替を行わない時には性能が約50%低下していた。この結果から、耐故障演算器でスループットを保ちながら少数の高速演算器でレイテンシを削減することで性能を大幅に改善できる事が分かった。

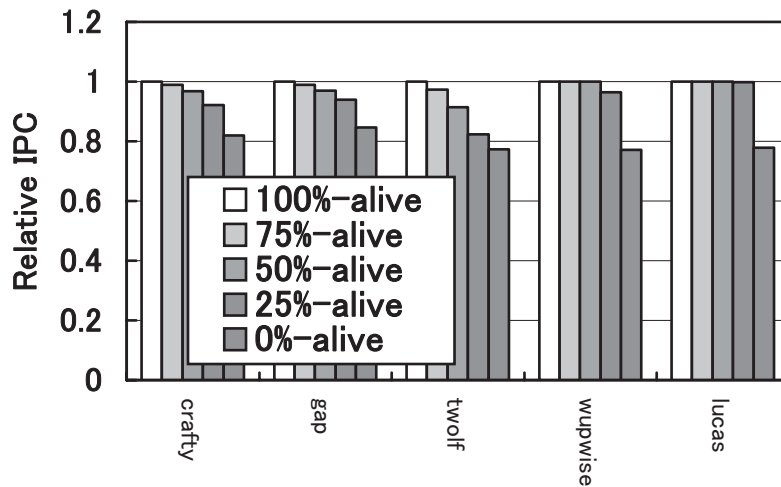


図 15 故障した高速演算器を耐故障演算器で代替した場合の性能

次に、二重実行を行う完全な実行ステージとして提案方式を組み込んだ場合の性能を評価する。評価では、SD-Fast、HFU-Det、HFU-1、SD-FT の 4 つの実行ステージ方式を比較する。SD-Fast は主演算と検出演算を全て高速演算器で実行する方法で、性能の上限を与える。HFU-Det と HFU-1 が提案方式 HFU を用いた方式である。HFU-Det は主演算の全てを高速演算器、検出演算を耐故障演算器で実行する。HFU-1 は主演算のうち 1 つのみを高速演算器で行い、残りを全て耐故障演算器で実行する。SD-FT は全ての演算を耐故障演算器で実行する方法で、性能の下限を与える。

図 16 に評価結果を示す。図より、HFU 方式を用いた場合、全ての演算を耐故障演算器で行った場合に比べて大幅に性能が向上している事が分かる。swim、wupwise、lucas など IPC が高いプログラムでこの傾向がみられる。多くのプログラムでは HFU-1 の適用により SD-Fast との性能差を 10%以内に抑制できた。この事から、耐故障演算器と少数の高速演算器を組み合わせる事によりスループットを高く保てる事が示された。

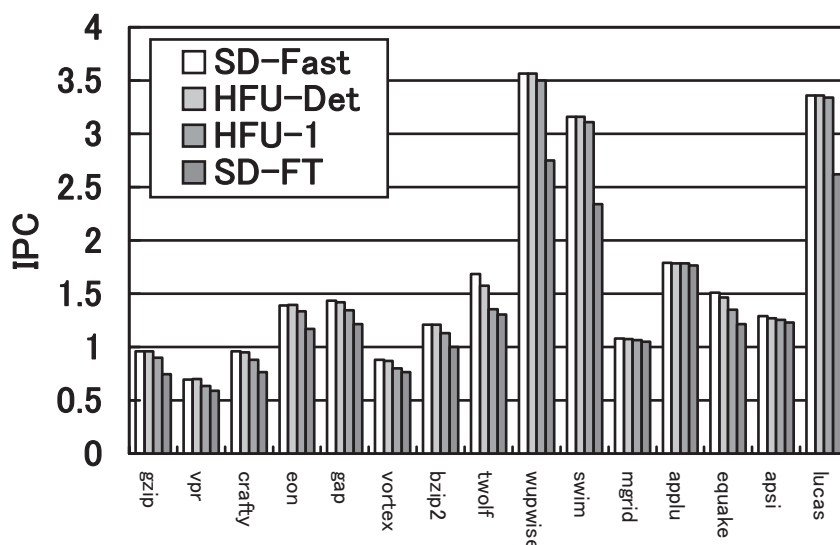


図 16 二重実行を行う完全な実行ステージとして提案方式を組み込んだ場合の性能

ここでは耐故障演算器と高速演算器を組み合わせる事によりプロセッサの実行速度を高く保つ方式 HFU を提案した。少数の高速演算器で命令実行のレイテンシを削減し、耐故障演算器で命令発行のスループットを保つことにより、高い実行性能を保ちながら実行ステージの耐故障化を実現できる事がわかった。

■高速侵入検知システム

○研究実施内容

当研究テーマでは、TCP ストリーム内容をハードウェアで高速に照合することより、バックボーンネットワークにおいて多数のトラフィックを高精度に検査する事が可能な侵入検知技術の開発を進めた。従来はアルゴリズムの限界、処理速度の限界によりバックボーンネットワークにおいて TCP ストリームの内容を検査する事は困難であった。我々は実用の再構成可能ハードウェアを用いて TCP ストリーム内容を 10 ギガビット/秒を超えるレートで検査する事を独自のアルゴリズムにより可能にした。

使用環境ごとにカスタマイズを行うために、中身のロジックを書き換えられるチップである FPGA(Field Programmable Gate Array)を用いた。実際に提案方式を FPGA 向けに実装することでロジックの実現可能性を評価した。また、パケット処理速度とメモリ使用量をサイクル・アキュレートなシミュレータを用いて評価した。さらに、FPGA を搭載した侵入検知システム実験基板を作成し、提案するシステムが実際に動作する事を確認した。

また、当初の目標に加えて、バックボーンネットワークにおける DDoS 攻撃緩和手法の検討を行った。提案方式が実用 FPGA で十分に実装可能である事を示した。また、ステートの保持に必要なメモリサイズが実用的なトラフィックに対して現実的な値に抑えられる事を示した。

○研究成果

本研究では、IDS の代表的方式の一つであるパターンマッチング方式を実現することとした。パターンマッチング方式とは、攻撃パケットのペイロード中に含まれる特定のバイト列を探すことにより攻撃を検出する手法である。この方式によりパケットヘッダ情報に基づく防御方式と比べ高精度に攻撃を検出することが可能である。しかしながら、対象となるバイト列はペイロード中の任意の場所から始まる可能性がある等の要因のため重い処理が必要となる。ソフトウェアによる処理では 10 ギガビット/秒に及ぶ今日のバックボーンネットワークへの適用は難しい事が知られている。そのため、ハードウェアを用いた処理方式が研究されている。特に、マッチさせるパターンに特化したハードウェアの生成が可能になるため、FPGA を用いた方式が多数提案されている。本研究では FPGA を用いる。

パターンマッチング方式を適用する場合、セキュリティを保つためにはパケットごとではなく TCP ストリームレベルでパターン検索を行う事が必要である。なぜなら、個々のパケット毎に独立にマッチングを行う方式では、検査対象となるパターンを複数パケットに分割して送ればマッチング機構による発見を免れることが可能であるためである。しかしながら、TCP ストリームレベルでマッチングを行う場合、以下の 3 つの課題をクリアする必要がある。

- (1) マッチングステートの退避と復帰：TCP ストリームレベルでマッチングを行う場合、ストリーム毎にマッチングユニットのステート退避と復帰が必要である。例えば、ストリーム A のパケットを照合した後にストリーム B のパケットが来た場合は、マッチングユニットからストリーム A の現在の照合ステートを退避し、ストリーム B の照合ステートを復帰させてから照合を行う必要がある。しかし、ハードウェアでの並列処理を目的として提案された既存の方式はステートサイズが大きい(ルールサイズに対して線形)のため、ステートの退避と復帰の実現が困難である。
- (2) パケット順序の逆転への対応：TCP レベルで照合を行う場合、パケット落ちなどがあるためストリームのパケットがシーケンス番号順に到着するとは限らない。パケットが逆順に到着した場合にバッファで並び替えてから照合する既存方式では、メモリ使用量が大きい。バッファでの並び替えを必要としない方式として逆順に到着したパケットを落として再送を待つ既

存方式があるが、TCP スループットが低下する。メモリ使用量が少なく、かつ、パケットをできるだけ落とさない方式が求められる。

- (3) 不正な再送パケットへの対応：TCP ではネットワークの混雑などで落ちたパケットを再送する機能がある。IDS をすり抜けるための攻撃手法として、最初に送ったデータと異なるデータを再送するという手法がある。既存のハードウェアベースでのマッチング方式の研究ではこのような不正な再送への対処がなされていない。そのような実装では、IDS でのマッチング結果が誤ったものとなり、攻撃者が IDS からの検出を逃れる事が可能となってしまう。

本研究ではこれらの限界を克服し高速な TCP ストリーム内容検査を可能とするアルゴリズムを提案した。本研究では(1)各 TCP ストリームのマッチングステートの退避と復帰が容易な「SBT(suffix based traversing)照合方式」、(2)シーケンス番号に対し逆順に届いた TCP パケットを効率よく処理するための「逆順照合法」、(3)不正な再送を防ぎ照合の完全性を保障するための「パケットフィンガープリント法」、の3つの要素技術を提案した。また、これらの要素技術を用いたマッチングハードウェアの FPGA ロジック記述を侵入検知のルール記述から自動生成するツールを作成した。さらに、生成したロジックを評価用に製作した FPGA 実験基板にプログラムし、実際に侵入検知システムとして動作させるデモを行った。以下では、これらの手法を個別に説明する。

(1) SBT 照合手法

SBT(suffix based traversing)マッチング法は、ステートサイズを小さく抑えることによりステートの退避と復帰を容易にしたマッチング方式である。SBT法は、Aho-Corasick 法というステートサイズを小さく抑えられる方法を元としている。Aho-Corasick 法ではステートサイズを理論的下限(ルール合計文字数の対数オーダー)に抑える事が可能である。SBT 法では高速化のために Aho-Corasick 法のステートを若干拡張して用いるが、ステートのサイズは理論的下限の2倍以内に抑えられる。実用上ステートサイズが32ビットを越えることはなく、高速な退避と復帰が可能である。Aho-Corasick 法では表引きによりステートマシンの動作を模倣することで文字列に対するマッチングを実現しているが、そのままでは複数文字の並列処理が難しい。そのため、SBT 方式では独自の表引き方式を導入しハードウェアで複数のオクテットを同時に照合する事を可能とした。この技術により高速なマッチングを可能とした。また、疎な表を FPGA の少ないオンチップメモリで効率よく実現するための独自の圧縮法を用いた。SBT 方式を用いたマッチング機構を実際の FPGA 向けに合成し評価を行った。評価では IDS のマッチングルールから FPGA ロジック記述を自動的に生

成するツールを開発して使用した。この事により、IDS のルールから自動生成したハードウェアが十分な性能を持つことを示した。

実 FPGA で実装したマッチング機構の packets 処理速度を図 17 に示す。横軸は並列に処理する packets のバイト数(文字数)、縦軸がスループットである。並列に処理する文字数を増やすとハードウェアサイズが増えるため、FPGA の容量制限により最大 32 文字までの並列処理とした。Set1(523 文字)~Set3(2178 文字)の 3 つのルールを用いた。評価より、最大で 30 ギガビット/秒を越える速度でのマッチングが可能である事が示された。SBT 法で独自の表引きアルゴリズムにより複数文字の並列処理を可能にしたことにより性能の向上が可能になった。

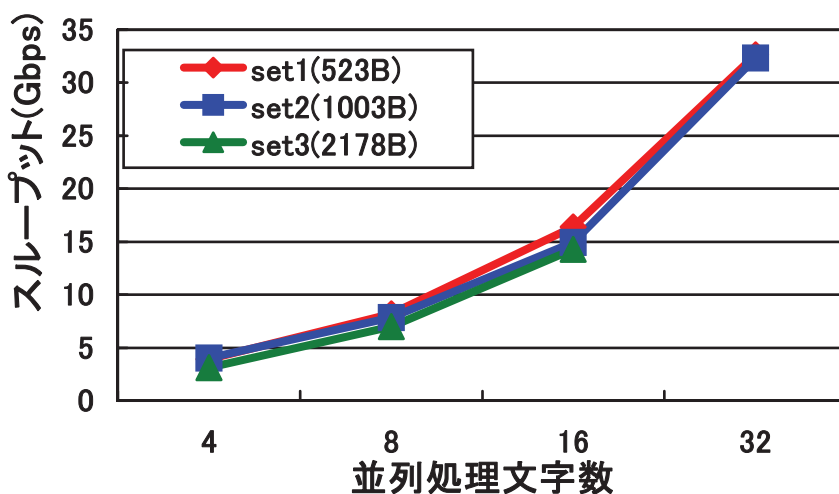


図 17 マッチング機構の packets 処理速度

また、実装したマッチング機構で退避・復帰が必要なステートビット数を図 18 に示す。比較のため、主要な既存方式である CAM、比較器、決定性オートマトン方式の値を示す。複数文字の並列処理が可能で、十分に高い速度が得られる方式のみを比較対象とした。図より、これらの既存方式ではルールサイズの増大とともにステートビット数が増大する事が分かる。Set3 では 500 ビットを越えており、このような多数の退避・復帰線を設けることは実用上困難である。一方、提案方式ではビット数は 24 以下となっており、高速な退避と復帰が十分可能である。実際、評価で実装したロジックでは退避と復帰によるスループットへのオーバーヘッドを完全に隠蔽することに成功している。

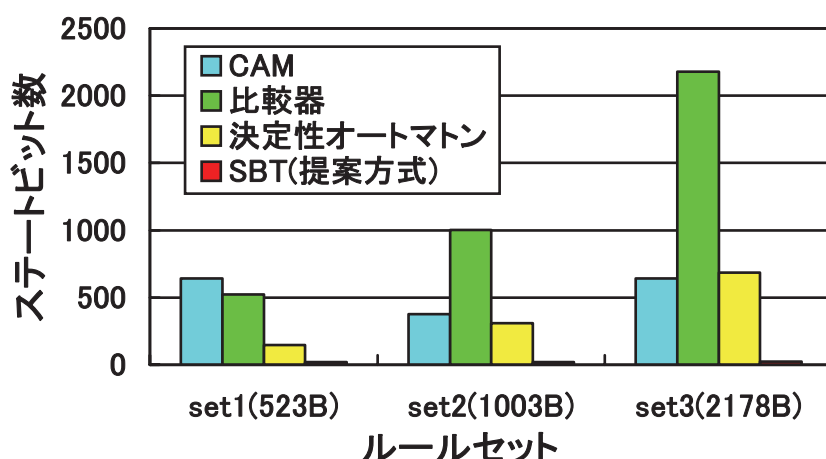


図 18 退避・復帰が必要な状態ビット数

(2) 逆順照合法

逆順照合法は、逆順に到着したパケットに対してその場でマッチングを行うことにより、並び替えに必要なバッファリングを省略できる方式である。パケットの境界部分に存在するパターンを判定するために一部の情報を残す必要はあるが、パケット自体をバッファリングする場合に比べて大幅にメモリ量を節約する事が可能である。パケット境界部分の情報の持ち方として、両端部分のオクテットをそのまま保持する 2 端保存法、後端部分については SBT 法の状態のみを保存する事によりメモリ使用量を削減した 1 端保存法の 2 つの手法を提案した。極端にサイズの小さい逆順パケットなど特殊な場合に対してはバッファリングを行うか、もしくはパケットを落として対処する。FPGA 向けに提案方式を合成し、実用的なゲート数で十分な性能が得られる事を示した。

(3) パケットフィンガープリント法

パケットフィンガープリント法は、送られた個々のパケットデータについてハッシュ値を保持する事により、一度目と違ったデータが再送された事を検出可能な方式である。このような不正な再送パケットを検出した場合は、そのパケットを落とす。この手法により、常に一貫したデータに対してマッチングを行い結果の正しさを保証する。正常な TCP の実装であれば再送パケットのデータは常に一度目と同じなので、本手法によりパケットを落とされる事は無い。最初に送られたパケットのデータそのものではなくハッシュを用いる事により、少ないメモリ量でのマッチングを可能とする。検査のためのロジックを FPGA 向けに実装し、実用的なゲート数で合成可能かつ十分な性能が得られる事を示した。

侵入検知システムの実装と評価

これらの要素技術を用いた TCP レベルでのマッチングを行う FPGA ロジックを自動生成するためのツールを作成した。このツールはマッチングの規則(ルール)を指定したファイルを入力として、その規則に基づくマッチングを行う FPGA ロジックの記述を自動生成する。ロジックはハードウェア記述言語 VHDL のソースファイルの形で出力される。このツールにより、ハードウェアの知識が無くともルール記述から侵入検知システムのロジックを自動的に生成する事を可能とした。

本テーマで作成した FPGA 搭載の侵入検知システム実験装置を用いて、提案方式を実装した侵入検知システムを構成しシンポジウムにおいてデモを行った。この装置は、10ギガビット・イーサネットのポートを2つ持つ。10ギガビット・イーサネットのバックボーンネットワークに対して直接検査を行う事が可能である。マッチングに用いたロジックは前述の FPGA ロジック自動生成ツールを用いて生成した。このデモにより、生成されたロジックが実用可能である事を示した。

また、侵入検知ロジック全体としてのパケット処理速度性能とメモリ使用量の評価を行った。評価では実装可能な限り並列処理文字数を増やして性能の変化を評価した。1 サイクルあたりの処理速度の評価は実機では難しいため、サイクルアキュレートなシミュレータを作成して評価を行った。処理速度はパケットサイズに依存する。現実的なトラフィックパターンの元での性能を評価するため、入力パターンのパケットサイズ分布は実ネットワークの観測結果をもとに決定した。

パケット処理速度の評価結果を図 19 に示す。横軸が SBT 照合機構で並列に処理される文字数、縦軸がシステム全体としてのパケット処理速度を示す。図より、rule0~rule3 の4つのルールで 10Gbps を越える処理速度を実現できることが分かる。このことから、提案システムを用いて 10ギガビット・イーサネット構成されたバックボーンネットワークに対しワイヤレートでのマッチングが可能である事が分かった。処理速度は並列処理文字数が増えるにしたがって性能の伸びが少なくなっている。これは使用している DDR-SDRAM のアクセスオーバーヘッドのためである。メモリアクセスの最適化によりオーバーヘッドを減らす事が将来の課題である。

提案方式を適用した場合のメモリ使用量を評価した(図 20)。比較のため、パケットをバッファリングして並び替える既存方式(ここではパケット保存法と呼ぶ事にする)の値を示す。図から、提案方式を適用した場合にメモリ使用量を大幅に削減できる事が分かった。特に in-flight データサイズが増加した場合にはメモリ使用量をパケット保存法の 7%程度に抑える事が可能である。評価結果より、提案方式を用いる事で少ないメモリ量で高速なネットワークにおけ

るマッチングを実現できることが分かった。

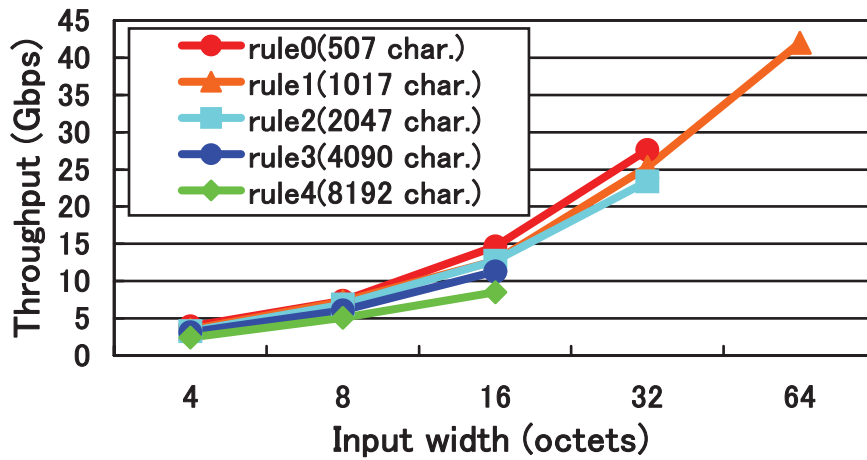


図 19 パケット処理速度

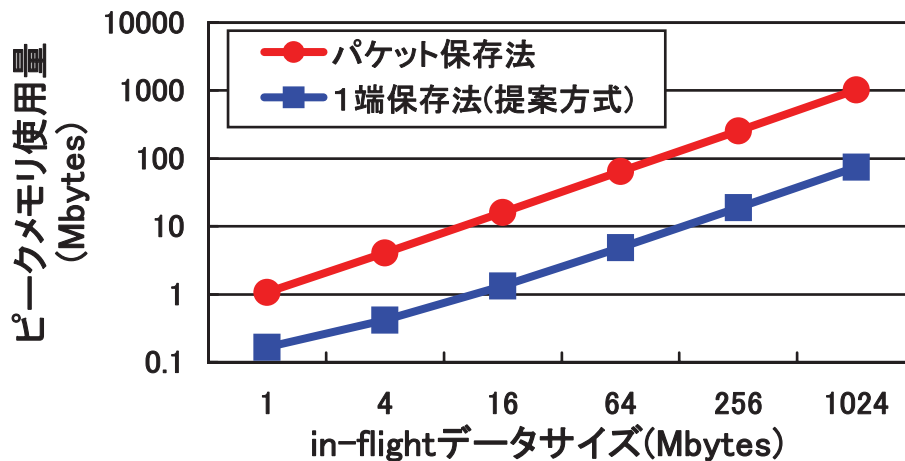


図 20 メモリ使用量

[2] 研究成果の今後期待される効果

■ 超ディペンダブルアーキテクチャ

今後の研究展開として、最適化とシステム化が挙げられる。タグ伝搬アーキテクチャや、タイミングマージンの異なる冗長実行などの新しい技術は、アーキテクチャ的な工夫により更なる効率化が可能であり、研究課題は多い。

一方、例えば、タグを付加した情報漏洩保護のような技術は、商用コンテンツの新しい配布形態や、プライバシーポリシーを付加しての個人情報入力など、新しく幅広い応用が考えられる。本研究で提案したアーキテクチャ技術を用いるシステムがどのように実装、使用されるか、OS や UI を含めた研究展開が重要であると考えられる。

ディペンダブルプロセッサの特性が活きる分野として、生活に密着した情報機器が挙げられる。携帯、ホームシアター端末、医療機器、車載コンピュータ、ロボットなどで今後用いられるプロセッサには高い信頼性や画像処理に代表される高い処理能力が要求される。更に、商用コンテンツやプライバシーに関わるデータなどに触れるため、ユーザに分かりやすく柔軟なセキュリティポリシーを運用できなければならない。このようなプラットフォームは、本研究で提案した様々な技術を応用する場として、また本研究のアプローチによる包括的なディペンダビリティを実現する場として適している。

高度な情報家電、車載コンピュータ、ロボットなどは我が国がデファクトを狙うことのできる分野である。ディペンダビリティの要求されるこれらの分野に対し、本研究の成果は有効に活かすことができると考えている。

■耐永久故障アーキテクチャ

これまでの技術では、プロセッサの実行ステージはその不規則な構造から定型的な耐故障化技術では効率の良い耐故障化が難しかった。例えば多重化方式が従来提案されていたが、基本的に実行ステージをブラックボックスとして多重化しているだけであった。そのため、チップ面積の利用効率が悪化していた。本研究は再構成という新しい手段を体系的に用いることにより、実行ステージを効率よく耐故障化することをはじめて可能とした。本研究の成果は今後の実行ステージの耐故障化研究に貢献するものと考えられる。

特に、再構成回路が持つ長いレイテンシという問題に対する現実的な解放を与えた点は今後の研究に有用であると考えている。再構成回路を用いると実行ステージのレイテンシが長くなることは実際にプロセッサとして使用する場合に性能の悪化につながる。我々が提案したHFU方式では耐故障化されていない演算器と組み合わせるといって実現可能な手段によって性能の低下を抑制することに成功した。この事から、再構成を用いた耐故障化方式の実現可能性が高まったと言える。この知見は、再構成を用いた今後の耐故障化研究に有用であると考ええる。

■構想侵入検知システム

本研究では、10ギガビット/秒を越える速度でのTCPストリーム内容検査をバックボーンネットワークで集中的に行える方式を確立した。この成果により、バックボーンネットワークで多数のトラフィックに対しTCPストリームレベルでの内容検査が可能となる。バックボーンネットワークでの侵入検知方式はホストレベルでの方式に対し少ない手間で広い範囲を守れるという利点がある。そのため、バックボーンネットワークでの侵入検知を高精度化することで、管理上の都合やコストの問題によりホストレベルでの対策が及ばないエンドホストに対してもある程度の安全性を確保する事が可能となる。

本研究で提案した方式はFPGAチップの高性能化・大容量化により性能スケールする。そのため、現在のみならず近い将来実現されるより高速なネットワーク(40/100ギガビ

ット・イーサネット)への対応が十分に可能と予想される。

本研究で提案した TCP レベルのマッチング手法では、ネットワークの中間経路でステートを持ちながら検査を行うという基本的な枠組みを構築した。今後、この枠組みを応用してバックボーンネットワークにおいてより高精度な侵入検知を実現できることが期待される。例えば、現状では TCP のステートのみを持って検査を行っているが、アプリケーションレイヤでのステートを持ちながら検査をするという拡張が考えられる。アプリケーションレイヤでのステートを再現した場合、プロトコルの規約からの逸脱を直接的に検出可能であり、高精度での攻撃検出に応用できると予想される。また、当テーマで検討を行ったが、本研究の枠組みを DoS 攻撃検出に応用することを視野に入れている。

科学技術への波及効果としては、本研究で提案した要素技術は、ネットワークの中間ノードで多数のトラフィックを扱う基礎になると考えられる。特に現実的なメモリサイズでステートを効率よく扱うための要素技術は、今後の学術研究において参考になるものとする。社会への波及効果としては、本研究はバックボーンネットワークでの侵入検知システムの発展に寄与するものと期待している。本研究では実際に動くシステムを構成しており、提案した技術の実用性は高いと考えている。実際、今までに論文を参照した企業から提案方式を用いた実製品の開発についてのオファーが来ており、関心の高さがうかがえる。

3.2 サーバ用基盤ソフトウェアグループ(慶應義塾大学河野研究グループ)

[1] 研究実施内容及び成果

■TCP ストリームフィルタ

インターネット・サーバのディペンダビリティ向上にはサーバの安定性向上だけではなく、インターネット・サーバのセキュリティ向上も重要な課題である。ここでは、エンドホスト上で動作するネットワーク侵入検知・防御システムである TCP ストリームフィルタの研究・開発を行ってきた。

TCP ストリームフィルタは、レイヤ7における通信をプロトコル規約に照らし、その正当性を検証する仕組みである。バッファあふれ攻撃などの不正攻撃を行う攻撃メッセージの多くは、厳密にはレイヤ7のプロトコル規約に違反していることが多い。そのため、通信メッセージの内容がプロトコル規約に厳密にしたがっているかどうかを検査することにより、不正攻撃の可能性のあるメッセージを事前に検出・破棄することが可能となる。TCP ストリームフィルタを用いることにより、未知の不正攻撃に対してもフィルタリングが可能となる。実際、2週間以上に渡って収集した実際のネットワーク・メッセージのログを用いて検証実験を行った結果、未知の攻撃であっても検出できるということが確認できている。また、TCP ストリームフィルタを実現するための効率的な TCP Reassembler の開発なども行った。

さらに、本手法のアイデアを拡張し、より精密なシグネチャを記述できるネットワーク侵入検知システムの開発を行った。シグネチャとは、攻撃メッセージの特徴を記述したデー

タ構造であり、典型的なシグネチャは単純なバイト列や正規表現などになっている。そのため、通信プロトコルのコンテキストを踏まえた記述は困難であり、無害なメッセージを有害なメッセージと判定してしまうことが多かった。TCP ストリームフィルタは通信内容を厳密に解析する機能を備えているため、特定のコンテキストに特定の文字列が出現した場合にのみ、攻撃メッセージとみなすといった、より精密なシグネチャの記述が可能となっている。実際に収集したネットワーク・メッセージのログを用いて、本方式を用いると検出精度の向上が図れることが確認できている。

■サーバ用基盤ソフトウェア

本研究では、インターネット上のサービスを実現するサーバのディペンダビリティ向上を狙いとし、そのための基盤ソフトウェア群について研究開発を行ってきた。特に、インターネット・サーバの中でも実用上、重要な位置づけを占めているウェブサーバを主要なターゲットとし、ウェブサーバの安定性を向上させるための、さまざまな要素技術の開発およびその統合を進めてきた。

ここでは、ウェブサーバを対象とした安定性向上のために次の3つの研究開発を行った。ひとつは、クラスタ・サーバを仮想的な高信頼計算機として抽象化するソフトウェア・レイヤの設計・開発である。負荷分散、障害検出、メンバシップ管理等を行う一連のライブラリ群を開発し、それらを統合した高信頼仮想計算機レイヤの実現を行った。さらに、このレイヤ上に高信頼ウェブサーバのプロトタイプ実装を行い、その有効性の検証を行った。この仮想化レイヤは研究期間の初期段階に行ったものであり、その後の研究を進めていく上での基盤として利用することを目指したものである。

次に、クラスタ・サーバ上で動作する個々のサーバ・ソフトウェアの安定化を図る技術について研究・開発を進めた。特に、サーバの管理者によるサーバ・ソフトウェアの管理コストの低減を目指し、サーバの挙動を定める性能パラメータの自動調整機構の研究・開発を進めた。たとえクラスタ・サーバを用いたとしても個々のPC上で動作するサーバが安定して動作しなければ、システム全体としての安定動作を保証することはできない。サーバが不安定な動作を示す要因にはさまざまな要因が考えられるが、ここでは性能パラメータを対象として研究を進めた。その理由は次のとおりである。1) 性能パラメータの設定が不適切であると、システム統合時に想定した性能を発揮することができず、サーバが過負荷状態に陥り、システムの安定性が損なわれやすい、2) 性能パラメータの設定が誤っていても、実運用時まで設定ミスを発見することが困難である。なぜなら、性能パラメータの設定が不適切でも、低負荷時には期待通りの動作を示すため設定ミスに気付きにくいからである。3) そもそも性能パラメータの設定自体が難しく、経験豊富で技術力の高い管理者であっても人手で正しい値に設定することが難しい。そのため、性能パラメータの設定を支援する機構が必要である。

本研究では実用上のインパクトを重視し、もっとも高いシェアを持つウェブサーバである Apache を対象とし、その性能を左右するといわれている Keep-Alive 時間というパラメータと MaxClients というパラメータに対象を限定した。

対象とする性能パラメータを限定したことにより、各パラメータが Apache の挙動に与える影響を詳細に分析することができるようになった。膨大な回数の実験を繰り返すことにより、Keep-Alive 時間に関しては、クライアントからのリクエストの間隔を分析すれば適切な値が割り出せること、MaxClients に関しては Apache のスループットのばらつきをヒントに適切な値が割り出せることを突き止めた。この実験結果をもとに Apache のパラメータを自動調整する機構を実現し、いくつかのワークロードに対して Keep-Alive 時間および MaxClients の値を最適な値に自動設定できるということを確認した（図 21、図 22）。

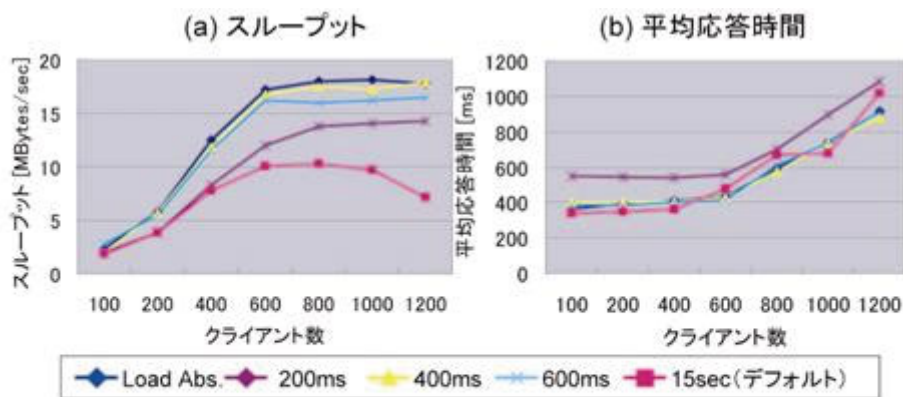
ウェブサーバの安定化のために行った 3 つ目の研究成果は、ウェブサーバそのものをインターネット上に分散配備し、インターネット上で負荷分散を図ろうという技術である。データセンター等にクラスターサーバを配備した場合、データセンターのネットワークに障害が発生すると、クラスターサーバ上のサービスにアクセスできないという問題がある。また、災害等によりデータセンターそのものが運用を停止した場合、サービスへのアクセスができなくなってしまう。従来、このような問題に対処するためにはあらかじめミラーサーバを配備しておくという方法がとられてきた。しかし、ミラーサーバを用いる手法では、配備するミラーサーバの台数、配備するインターネット上の位置などを決めるため、あらかじめ各サービスに対する需要を予測しておく必要があった。その結果、サービスへの需要変動に追従することが難しかった。インターネット上のサービスでは、急激に需要変動が起きることがしばしば指摘されており、従来の方式では安定したサービスを提供することは困難であった。

本研究では、あらかじめ、サーバ・ノードとよぶ信頼できる計算機をインターネット上に分散配備しておき、そのサーバ・ノード群上で複数のサービスを同時に提供する。サービスの需要変動に応じて、適切な分量の計算機資源を各サービスに割り当てることにより、需要変動に応じて計算機資源が配分される。その結果、サービスへの需要が急激に増大したとしても、十分な台数のサーバ・ノードが割り当てられ、サービスの品質が著しく低下するようなことはない。いわばインターネット上に分散したホスティング・サービスであるといえる。現在では、数百から数千台のサーバ・ノード上で数千以上のサービスが提供できるような基盤を目指している。現在、研究開発を進めているこの基盤のことを ExaPeer と名付けている。

ExaPeer では、オーバーレイネットワークのひとつである CAN (Content-Addressable Network) と、ネットワーク座標システムのひとつである GNP (Global Network Positioning) を組み合わせ、サーバ・ノード上に物理的なネットワーク・トポロジを考慮し

たオーバーレイネットワークを構築する。このオーバーレイネットワークを用いて各サービスに対する需要を効率的に収集し、それに応じて適切にサーバ・ノードを配置する。本手法は自律分散的に動作するため、集中サーバを必要とせず、高いスケーラビリティと信頼性を有している。

ネットワークシミュレータを用いて実験を行ったところ、ExaPeerは既存の方式よりもより適切な位置にサーバを配置可能であることが確認されている。図23のグラフに示したように、ExaPeerでは、より多くのクライアントが少ないホップ数、およびラウンドトリップ時間(RTT)で目的のサーバに到達できていることが確認できる。



- poll() の実行間隔を調整
 - スループットを平均131.9%向上
 - 平均応答時間の差は平均2.0%

図 21 パラメータ自動設定



図 22 パラメータ自動設定の効果

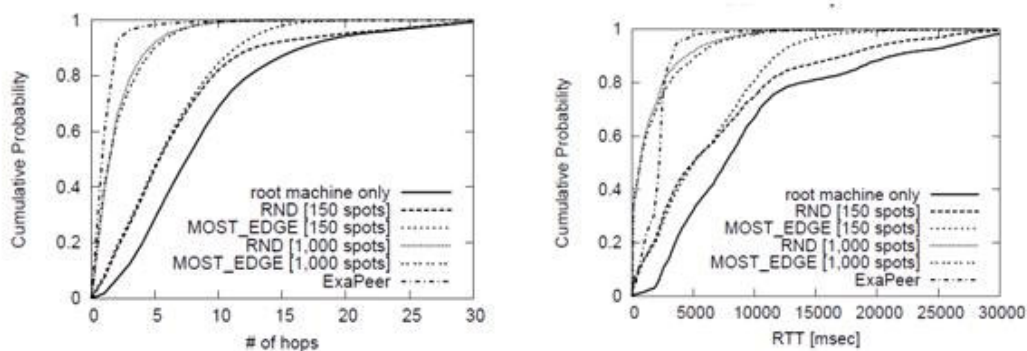


図 23 ExaPeer の効果

[2] 研究成果の今後期待される効果

■TCP ストリームフィルタ

インターネット・サーバの安全性向上のため、コアとなる部分はほぼ完成しており、実用化に至る最後の段階をつめれば実用化も可能だと考えている。今後、ますます重要性を増すであろうセキュリティ基盤技術については、国産のしっかりした技術を持っていることが大切であり、そうした技術のひとつになると考えている

■サーバ用基盤ソフトウェア

ウェブサーバを安定化させる技術のうち、性能パラメータの自動調整機構は実用化を強く意識した研究成果であり、近いうちに実用化可能な技術であると考えている。分散ホスティング・サービスの研究成果は、まだシミュレーションでの検証にとどまっており、すぐに実用化できる段階には到達していない。しかし、需要変動に応じた計算機資源の割り当てを可能とする基盤は強く希求されており、Amazon EC2 などのホスティング・サービスが出現しつつある。今後、精力的に研究を進めていけば、日本発のインターネット基盤技術となりうる可能性を秘めていると期待している。さらに、このような基盤が提供されれば、新しいサービスを安価に提供することが可能となり、インターネット上での新しいサービスの開拓にも通じる。特に、優れたサービスのアイデアを持ちながら、安定したサービスを提供できるだけの計算資源を持たないベンチャー企業等にとって大きな契機となるのではないかと考えている。

3. 3 アプリケーション用基盤ソフトウェア研究グループ（東京工業大学千葉グループ）

[1] 研究実施内容及び成果

本グループでは、ディペンダブルなソフトウェアを開発するためのプログラミング言語およびソフトウェア開発ツールの研究をおこなった。第一に、アプリケーション・ソフトウェアのプログラムを変換し、信頼性を高めるための機能を自動的に埋め込む技術を開発することをねらった。これにより、ハードウェアやオペレーティングシステム、ミドルウェアが必要な機能を提供して

もアプリケーション・ソフトウェアがその機能を使いこなせない、という従来の問題を解決する。第二に、完全な自動化が困難な接続については、高度なモジュール化技術を駆使して、接続に必要なプログラムをアプリケーション・ソフトウェア本体から分離されたプログラムとして記述できるようにすることをねらった。これにより、アプリケーション・ソフトウェアの開発者本人ではなく、ディペンダビリティの専門家が接続作業を分担できるようになる(図 24)。このような分担作業により、ディペンダビリティを飛躍的に高めることができる。本技術開発で鍵となるのがプログラム変換技術とプログラムのモジュール化技術である。本グループでは、アスペクト指向(AOP)ソフトウェア開発技術を応用し、これらの技術を開発することを目指した。

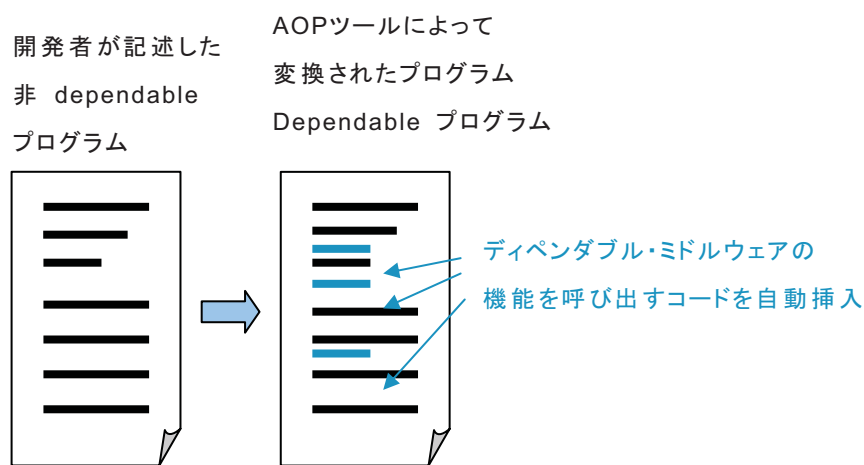


図 24 アスペクト指向によるディペンダビリティの向上

アスペクト指向ソフトウェア開発技術とは、1990年代の後半に提案され、ここ10年ほど活発に研究されている新しいモジュール化技術である。ソフトウェアを開発する際に、プログラムをいくつかのモジュールあるいはコンポーネントに分割して開発することは今や常識となっている。分割することによって、プログラムを個々のモジュールごとに個別に開発し、保守できるからである。これにより分業体制によるソフトウェアの開発や、保守を用意にすることによる品質の向上が期待できる。ところが現実のプログラムは、個々の部分が互いに密接に関連していることがほとんどで、簡単にはモジュールに分割できない。字面の上で分割できていたとしても、実際には互いに強く依存しており、モジュールごとに独立に開発、テスト、保守することが困難なことが多い。この問題を解決するために、プログラミング言語の研究分野ではさまざまなモジュール化技術が研究開発され、現実のソフトウェア開発の現場に導入されてきた。手続き(プロシージャ)や関数(ファンクション)といった概念も、ごく初期に提案された初歩的なモジュール化技術であるといえる。その後、オブジ

ェクト指向が生まれ、今や産業界でも標準的なモジュール化技術となっている。しかし、オブジェクト指向言語を用い、オブジェクト指向開発手法を導入しても、依然として問題が残っている。プログラムをどのようにモジュール（あるいはクラス）に分割すればよいかは難しく、開発者の経験に頼る部分が多いのである。この問題に対処するため、適切な分割方法を見つけるための分析手法や、デザインパターンと呼ばれる典型的な分割方法の事例集のようなものが数多く研究開発されてきた。

アスペクト指向の出発点は、一般的なプログラムについて、最適なモジュール分割が常に存在するとは限らない、という考察である。ソフトウェア開発の各局面において、適切な分割方法は異なるのである。そこで、必要に応じて異なる角度や側面（アスペクト）からプログラムを分割して扱えるようにしようと研究が始まった。これがアスペクト指向技術の研究である。世界中の研究者によるこれまでの研究で、現在、異なる側面からの分割をひとつのプログラムの中に共存させることまではできている。これを可能にする代表的な言語が AspectJ 言語である。

図 25 で、理想的には側面 A からの分割と B からの分割を開発中に切り替えられるとよいが、AspectJ を使うと少なくとも、たとえば側面 A からの分割を基本に、一部側面 B から分割したときに得られるモジュール（右下の太い枠線で囲まれたモジュール）をプログラム中に作ることができる。このモジュールのことをアスペクトという。従来のプログラミング言語で同等の分割をしようとすると、濃く塗られたモジュールとアスペクトの間の連携をとるための面倒なコードをプログラム中に追加しなければならず、かえって保守性を低下させてしまうこともあった。

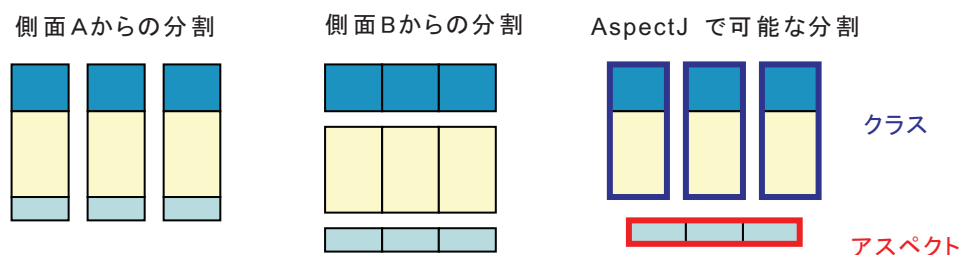


図 25 アスペクトによるプログラム分割

本グループの研究では、アプリケーションプログラム中に従来含まれていたディペンダビリティにかかわるコードを、アスペクト指向技術を使って独立したモジュール（すなわちアスペクト）にする。従来技術で作られたプログラムでは、ディペンダビリティにかかわるコードは独立したモジュールにならず、

複数のモジュールにばらばらに含まれていた（上では、「側面 A からの分割」の図の中の薄く塗られた部分）。このため保守が難しく、開発者全員がディペンダブル・コードを書く技術を持ち、注意深く各モジュールに埋め込まなければならなかった。ディペンダブル・コードを書く技術をもった開発者を多数必要とする点で、従来方式は高いコストがかかり、ディペンダブルなソフトウェアの普及の妨げになっていた。一方、アスペクト指向技術を用いると、ディペンダブル・コードは独立したモジュールになっているので、ディペンダビリティに責任をもつ専門開発者だけがディペンダブル・コードの開発と保守をおこなえばよく、他の開発者はディペンダビリティに大きな注意を払わなくてよい。ディペンダビリティにふなれな開発者が、誤ったディペンダブル・コードを書いてしまう危険性もなくなるのが期待できる。アスペクトとして書かれたディペンダブル・コードは、コンパイル時あるいはプログラムのロード時に分解され、他のモジュールに自動的に挿入される。図 25 を例にすると、開発者が見ているモジュール分割は「AspectJ で可能な分割」のようにになっているが、コンパイル時ないしはロード時に、「側面 A からの分割」のようにプログラムは自動的に変換される。したがって実行時にはアスペクト指向技術を使わずに開発したソフトウェアと同等になるので、アスペクト指向技術を使うことによってディペンダビリティが低下することは基本的にはない。

本プロジェクトでは、平成 14 年度後半から平成 15 年度にかけて、まず本グループの目的であるディペンダブル技術の研究開発のための準備として、アプリケーション・ソフトウェアのプログラム変換のための基礎技術の研究をおこなった。実施に当たっては、本グループが従来開発してきた Javassist を改良する形で研究をおこなった。Javassist は Java 言語で書かれたプログラムをコンパイルして得られるバイトコード（Java 仮想マシンの機械語）を変換するためのライブラリである。

Javassist の特徴は、Java バイトコード（機械語）の知識がなくともプログラム変換を実装できるように、抽象化をほどこしている点である（図 26）。Javassist はバイトコードの内容を Java 言語での用語に翻訳して開発者に見せる。開発者が翻訳された内容をもとにプログラム変換の手順を指示すると、Javassist がそれをバイトコードのレベルでの変換に再度翻訳して実行する。これにより Java 言語は知っていてもバイトコードのことは知らない一般の開発者が Javassist を使って、プログラム変換を容易に実装できるようになった。

Javassist のこのような特徴は、プログラム変換技術の利用を手軽なものとし、産業界への普及に貢献した。Javassist 自体も毎月 1000 件以上のペースでダウンロードされている他、多数のソフトウェアに基幹部品として組み込まれて使われている。その中でもたとえば米 Red Hat 社の J2EE サーバ JBoss は、月間 6 万から 10 万件ダウンロードされており、商用の Web アプリケーシ

ョンフレームワークとしては世界市場第3位である。また日本国内発のオープンソース・webアプリケーションフレームワークである Seasar にも組み込まれているが、このフレームワークは三菱東京 UFJ の金融ソリューションシステムに採用されている。


TOKYO INSTITUTE OF TECHNOLOGY
Source level abstraction

You don't have to learn Java bytecode for using Javassist.

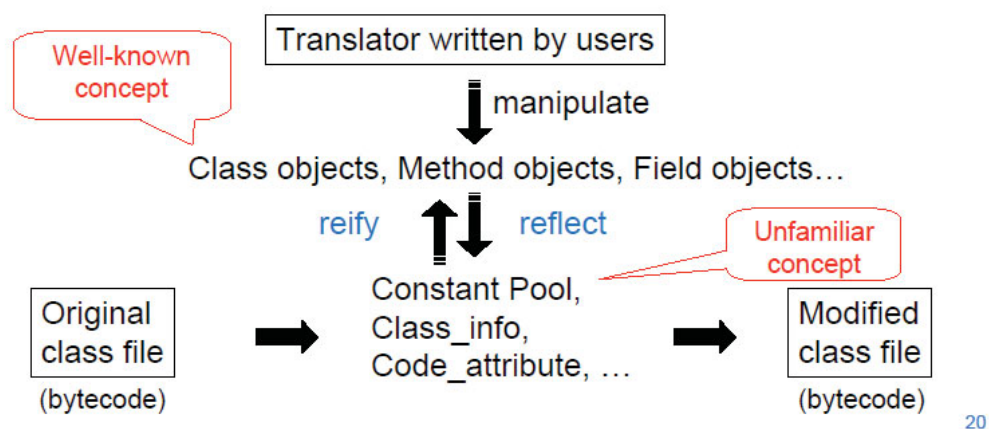


図 26 Javassist

また平成16年度には、研究の最終目標であるディペンダビリティ機能を自動的にプログラム中に埋め込む技術の開発に向け、主に本技術の応用対象である web アプリケーションサーバの性能改善の技術に取り組んだ。16年度に得られた性能改善のための知見をもとに、その後、そのような知見に基づいた技法を自動的にプログラム中に埋め込む技術を開発してゆくことになった。本研究では graceful degradation を web アプリケーションサーバで可能にするために、そのようなサーバの挙動の典型例を考察し、そのような例の下で過負荷時にも急激な性能低下・システム停止がおこらないようにする制御法を開発した(図27)。また、現実に使われているオペレーティングシステムの種類によって、過負荷時の挙動は大きく異なるため、本研究でおこなっているようなミドルウェアによる制御技術を開発し、オペレーティングシステム間の差異を吸収することが大切であることを確認した。

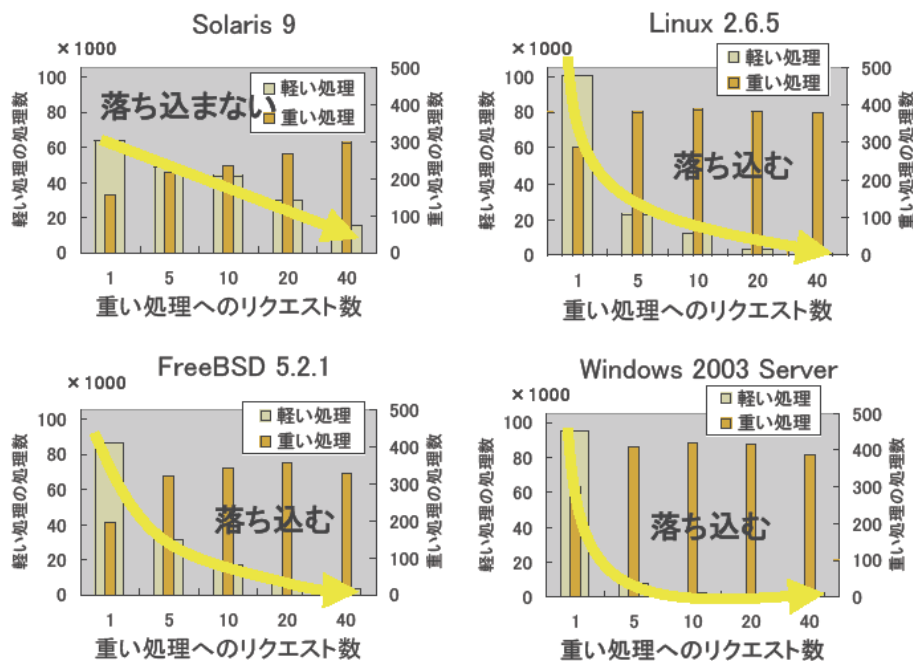


図 27 Graceful Degradation の実現

図 27 は過負荷時の各オペレーティングシステムの挙動を表す。これは Web アプリケーション上で重い処理と軽い処理の 2 種類のサービスを動かし、単位時間あたりの重い処理へのリクエストを次第に増やして過負荷状態にしたとき、単位時間あたりに受付可能な軽い処理へのリクエストの数を測定したものである。現実の web アプリケーションでは、トランザクション処理（支払い処理）へのリクエストが増えてサーバが過負荷になったときに、通常のページ閲覧（軽い処理）へのリクエストの処理能力がどの程度影響を受けるかに対応する。実験では、オペレーティングシステム以外のミドルウェアやアプリケーションプログラム、Java 仮想機械は変更していない。実験の結果、オペレーティングシステムによっては、過負荷時には重い処理を優先して軽い処理の受付数が大きく落ち込むものがある一方、軽い処理を優先して重い処理の実行が遅くなるものがあることがわかった。これは、同一アプリケーションプログラムであっても、オペレーティングシステムが代わるだけで過負荷時の挙動が大きくかわりうることを意味する。ディペンダブルなソフトウェアを実現するためには、過負荷時の挙動を安定化させることが大切だが、この実験により、安定化が容易ではないことがわかった。この知見から、本グループでは、オペレーティングシステムに依存せず、アプリケーションプログラムのレベルで過負荷時の挙動を自動的に安定化させる技術を開発することになった。

平成17年度には、アスペクト指向技術を応用してディペンダビリティ機能を自動的にプログラム中に埋め込む技術の開発に注力した。同年度はこれまでの成果をまとめる形で web アプリケーション向けにアスペクト指向システム GluonJ を開発し、これを利用したデモ・アプリケーションを試作した。このアプリケーションは河川の水位をインターネット経由で配信する web アプリケーションである。ディペンダブルにするには混雑時にも河川の水位をセンサー（作成したアプリケーションはデモ目的なので、センサーは本物ではなくシミュレータである）から最優先で取得するようにしなければならないが、Java 言語等で開発された一般的な web アプリケーションではそのようなスケジューリングは困難である。OS はスケジューリング機能を提供するが、分厚いミドルウェア層が介在するため、OS のもつスケジューリング機能を、うまくアプリケーションレベルの処理のスケジューリングに対応づけられないからである。本研究ではアスペクト指向システムを利用して、アプリケーションレベルでスケジューリングを実現するコードを対象のアプリケーションに自動挿入し、自動的にディペンダブルにする技術を、河川の水位を配信する web アプリケーションを例に開発した（図 28）。



図 28 河川水位監視システム

図 29 は、混雑時に水位センサーからどのくらいの時間でデータの取得に成功

したかを示すグラフである。デッドラインは15秒なので、これ以上時間がかかるとデータ取得失敗である。3種類の設定を比較しており、No control とあるのが何もしない通常の web アプリケーションである。Our scheduling とあるのは、スケジューリング・コードをアスペクトとして挿入して動かした web アプリケーションである。Admission control とは、アプリケーションプログラムを少し改造して、クライアントからリクエストを受け取った際に負荷を考慮して即座に処理を開始するか、あるいは少し待ってから開始するか切り替えるようにしたものである。これは従来から知られているアプリケーションレベルのスケジューリング技術を代表するものとして測定した。

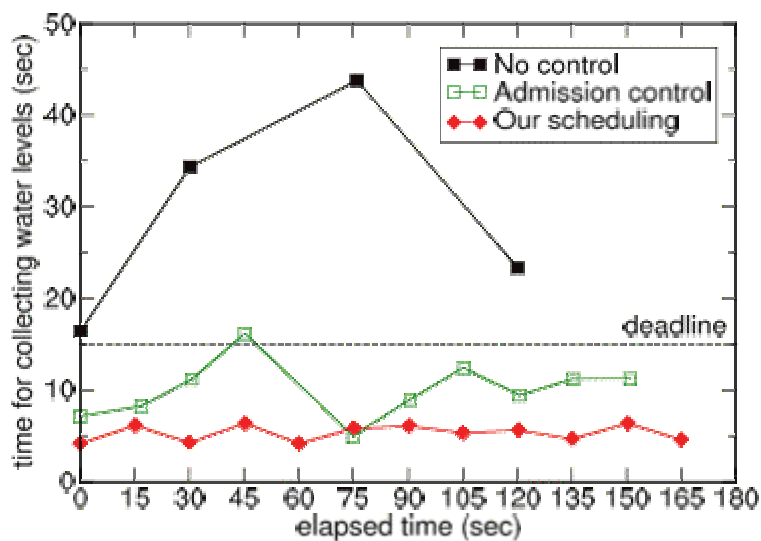


図 29 河川水位監視システムにおけるデータ取得時間

図 29 より、アスペクト指向技術を利用したものが最も性能がよいことがわかる。水位データの取得時間は最も短く、デッドラインをミスすることもなかった。一方、何もしない通常のものには性能が悪く、ほとんどのケースでデッドラインをミスした。Admission control は比較的よい性能を示したが、1度だけデッドラインをミスしている。

平成18年度には、前年度から引き続きアスペクト指向技術を応用してディペンダビリティ機能を自動的にプログラム中に埋め込む技術を開発した。前年の17年度にはこれまでの成果をまとめる形で web アプリケーション向けにアスペクト指向システム GluonJ を開発し、これを利用したデモ・アプリケーションを試作したが、18年度はこのデモ・アプリケーションで使われているアスペクト指向・ディペンダブル技術をさらに洗練し、論文としてまとめた。本研究のアスペクト指向システムを利用したアプリケーションレベルでのスケジューリングに関する技術開発について報告した論文は、アスペクト指向分野の国際会議としては最もレベルが高いといわれる AOSD にて採録・発表され、

高い評価を得た。また18年度は、さらなるディペンダビリティ技術を開発するため、そのテストベッドとしてのアプリケーション開発も始めた。

平成19年度は最終年度であるので、これまでに得られた知見をもとに、本グループで開発したアスペクト指向システム GluonJ の設計を大幅に見直し、より汎用的で使いやすいものに改善した。またこのシステムの有用性を示すデモとして、テレビ番組を共通の話題とする SNS (Social Networking System) を開発した(図 30)。SNS のような web アプリケーションは利用者の人気にシステム負荷が大きく左右される。人気がなければ小規模なシステムでもサービスを実行できるが、ひとたび人気が高まると非常に多数のアクセスが集中しがちで、サービスをディペンダブルに続行するには高性能なシステムが必要になる。しかし事前にシステム負荷を予測することはできないため、利用者の人気に応じて開発者が迅速に対応し、システムの規模を拡大・縮小してやる必要がある。典型的には負荷が増大した場合は、データベースを分散化してデータベースがボトルネックにならないようにする必要がある。ところが本デモシステムがデータベースアクセスのために利用しているミドルウェア S2Dao はデータベースの分散化に対応していない。そこで本デモでは、データベース分散化のためのコード(ディペンダブル・コード)をアスペクトで記述し、これを開発者が適宜加えることで、ミドルウェア本体のプログラムは修正せずに、自在に分散化できることを示すこととした。

応用システム: AllYouNeedsTV

- TV 番組紹介 SNS
 - TV 番組表のブラウザ機能を中心に、参加者が番組の感想を書き込める CGM (Consumer Generated Media)
 - TV 番組のデータは本物を利用。



図 30 アスペクト指向を用いた SNS

アスペクトを使ってミドルウェアを拡張することで、不必要な機能を最初からすべて盛り込むことによるミドルウェアの肥大化をさけることができる。アスペクト指向により、必要なディペンダビリティを必要なときに組み込んで利用することができるようになった。

[2] 研究成果の今後期待される効果

本研究グループでは、アスペクト指向に代表される最新のソフトウェア技術をディペンダブルシステムの構築に応用する研究をおこなった。従来、ハードウェアやオペレーティングシステムがディペンダブルなサービスを実現する機能を提供していても、アプリケーションプログラムからそれを利用することには複雑なプログラムを書かなければならないことが多かった。これは開発の難易度を上げ、開発コストの点から、それらの機能が十分に生かされないことが問題であった。本研究グループの研究は、ディペンダビリティのための機能を利用するコード(ディペンダブル・コード)を数個のモジュールに集約し、開発者の中の少数のディペンダビリティの専門家によって開発保守ができるようにした。全開発者がディペンダブルコンピューティングに詳しくなる必要もなく、ディペンダブルシステムの構築をより容易にすることが期待できる。

研究の中で開発したソフトウェアはオープンソース・ソフトウェアとして公開されており、今後の普及が期待できる。なかでも Javassist はすでに多くの採用事例があるが、プログラム変換技術を応用したミドルウェアの開発に活かされている。本研究のケーススタディから、ディペンダブル分野のミドルウェアにおいても同技術の浸透が期待できる。また研究の後半で開発した GluonJ はまだ普及しているとはいいがたいが、今後の普及し、さまざまなディペンダブルシステムの構築に活用されることが期待される。

3. 4 ディペンダブル技術の統合

高速侵入検知システムとTCPストリームフィルタは、前者がブラックリスト方式の高速ハードウェア、後者がルールベースのホワイトリスト方式のソフトウェアであり、相互補完の関係にある。これらは統合してCRESTシンポジウムなどでデモを行っている。

サーバ用基盤ソフトウェアとアプリケーション用基盤ソフトウェアも協調・補完関係にあり、制御パラメータの設定、負荷分散、故障ノードがあるときの管理など、両方のソフトウェアが協力しながら最適な方法をとることになる。CRESTシンポジウムなどで、両者の協調のデモを行っている。

4 研究参加者

研究グループ名： アーキテクチャ研究グループ（坂井修一）

氏名	所属	役職（身分）	研究項目	参加時期
坂井 修一	東京大学大学院情報 理工学系研究科 電子情報学専攻	教授	超ディペンダブル チップおよび 全体とりまとめ	H.14.11 ～ H.20.3
五島 正裕	同	准教授	超ディペンダブル チップ	H17.4 ～ H.20.3
平木 敬	同研究科 コンピュー タ科学専攻	教授	サーバアーキテ クチャおよび侵 入防止ハードウ ェア	H.14.11 ～ H.20.3
稲葉 真理	同	特任助教授	サーバアーキテ クチャ	H.14.11 ～ H.20.3
白石 知之	同 工学系研究科	助手	侵入防止システ ム	H.14.11 ～ H.16.10
岡部 淳	同 情報理工学系研 究科電子情報学専攻	博士課程退 学	リコンフィギュラ ブルアーキテクチャ	H.14.11 ～ H.15.3
森 晃平	同	修士課程退 学	リコンフィギュラ ブルアーキテクチャ	H.14.11 ～ H.15.9
岩間 智女	同	博士課程中 退	アーキテクチャ設 計法	H.15.4 ～ H.16.3
三浦 栄之	同	修士課程修 了	CMP アーキテク チャ	H14.11 ～H16.3
入江 英嗣	同	JST 研究員	超ディペンダブル CPU アーキテ クチャの研究開 発および学生指 導	H.16.4 ～ H.20.3
田代 大輔	同	博士課程修 了	コンパイラ	H.16.4 ～ H.17.3
ルオンディン フォン	同	博士課程修 了	アーキテクチャ	H.16.5 ～

				H.20.3
菅原 豊	同 コンピュータ科学 専攻	助教	アーキテクチャ	H.16.4 ～ H.20.3
中村 洋介	同	博士課程	リコンフィギュラブル アーキテクチャ に関する研究デ ータの収集、解 析	H.15.4 ～ H.20.3
高田 正法	同 情報理工学系研 究科 電子情報学専攻	修士課程修 了	アーキテクチャ	H.15.4 ～ H.17.3
櫻井 隆雄	同	修士課程修 了	アーキテクチャ	H.15.4 ～ H.17.3
柳川 善光	同 工学系研究科電 子情報学専攻、 文部科学省宇宙科学 研究所 斉藤浩文研究室	博士課程修 了	アーキテクチャ	H.15.7 ～ H.17.3
豊島 隆志	同 情報理工学系研 究科 電子情報学専攻	修士課程修 了	アーキテクチャ	H.16.4 ～ H.18.3
初田 直也	同	修士課程修 了	アーキテクチャ	H.16.4 ～ H.18.3
清水 一人	同	修士課程修 了	アーキテクチャ	H.16.4～
栗田 弘之	同	修士課程修 了	アーキテクチャ	H.16.4 ～ H.19.3
上野 裕也	同	修士課程修 了	アーキテクチャ	H.16.4 ～ H.18.3
門馬 太平	同	修士課程修 了	アーキテクチャ	H.16.4 ～ H.18.3

渡邊 翔太	同	修士課程修了	アーキテクチャ	H.16.4 ～ H.17.3
一林 宏憲	同	修士課程2年	アーキテクチャに関する研究データの収集、解析	H17.4～ H.20.3
荻野 健	同	同	アーキテクチャ	H17.4～ H.18.3
勝沼 聡	同	同	アーキテクチャ	H17.4～ H.20.3
塩谷 亮太	同	同	アーキテクチャに関する研究データの収集、解析	H17.4～ H.20.3
平井 遙	同	同	アーキテクチャ	H17.4～ H.18.3
渡辺 憲一	同	同	アーキテクチャに関する研究データの収集、解析	H17.4～ H.20.3
廣瀬 健一郎	同	同	アーキテクチャに関する研究データの収集、解析	H17.4～ H.20.3
亘理 靖展	同	同	アーキテクチャに関する研究データの収集、解析	H17.4～ H.20.3
ドウブトウ ピエール	同	同	アーキテクチャ	H17.4～ H.20.3
金 大雄	同	修士課程1年	アーキテクチャに関する研究データの収集、解析	H18.4～ H.20.3
杉本 健	同	同	アーキテクチャに関する研究データの収集、解析	H18.4～ H.20.3
樽井 翔	同	同	アーキテクチャに関する研究データの収集、解析	H18.4～ H.20.3
堀尾 一生	同	同	アーキテクチャ	H18.4～ H.20.3
梶本 尚之	同	同	アーキテクチャに関する研究データの収集、解析	H18.4～ H.20.3
山田 恭弘	同	同	アーキテクチャに関する研究データの収集、解析	H18.4～ H.20.3

安藤 徹	同	同	アーキテクチャに関する研究データの収集、解析	H19.4～ H.20.3
江口 修平	同	学部4年生	アーキテクチャ	H19.4～ H.20.3
喜多 貴信		同	アーキテクチャ	H19.4～ H.20.3
今田 俊寛		同	アーキテクチャ	H19.4～ H.20.3
顔 開		同	アーキテクチャ	H19.4～ H.20.3
横田 侑樹		同	アーキテクチャ	H19.4～ H.20.3
バルリ ニコ デムス	日本テキサス・インスツルメント	デジタルデザイン・エンジニア	アーキテクチャ	H.14.11 ～ H.17.3
八木原 晴 水	同 情報理工学系研究科・電子情報学専攻・坂井・五島研究室	研究チーム事務員	研究チームの事務処理、経理処理	H.15.4 ～ H.20.3

研究グループ名：サーバ用基盤ソフトウェア研究グループ(河野健二)

氏名	所属	役職(身分)	研究項目	参加時期
河野 健二	慶應義塾大学理工学部 情報工学科	准教授	サーバ用基盤ソフトウェア	H14.11 ～ H20.3
岩崎 英哉	電気通信大学情報工学科	教授	サーバ用基盤ソフトウェア	H17.4 ～ H19.3
揚妻 匡邦	同大学電気通信学研究科 情報工学専攻	博士後期課程修了	サーバ用基盤ソフトウェア	H14.11 ～ H17.3
杉木 章義	同	博士後期課程修了	サーバ用基盤ソフトウェアの実装	H14.11 ～ H19.3
河合 英宏	同	博士前期課程修了	サーバ用基盤ソフトウェア	H14.11 ～ H16.3

本多 孝充	同	博士前期課程 修了	サーバ用基盤ソ フトウェア	H14.11 ～ H16.3
村田 光一	同	博士前期課程 修了	サーバ用基盤ソ フトウェア	H14.11 ～ H16.3
渡辺 元晴	同	博士前期課程 修了	サーバ用基盤ソ フトウェア	H14.11 ～ H16.3
太田 尚一	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H15.4 ～ H17.3
塚田 肇	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H15.4 ～ H17.3
阿部 勝幸	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H16.4 ～ H18.3
Md. Rahat Kabir	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H16.4 ～ H18.3
渋谷 賢	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H16.4 ～ H18.3
Le Thanh Hai	同	博士前期課程 修了	サーバ用基盤ソ フトウェアの実装	H16.4 ～ H18.3
山田 浩史	慶應義塾大学大学院 理工学研究科開放環 境科学専攻	博士後期課程	サーバ用基盤ソ フトウェアの実装	H16.4 ～ H20.3
浅原 理人	同	同	サーバ用基盤ソ フトウェアの実装	H17.4 ～ H20.3
嶋村 誠	同	同	セキュリティ機構 の設計・開発	H17.4 ～ H20.3
花岡 美幸	同	同	セキュリティ機構 の設計・開発	H17.4 ～ H20.3
宮地 大輝	同	博士前期課程 修了	セキュリティ機構 の設計・開発	H17.4 ～ H19.3
山口 聖司	同	同	セキュリティ機構 の設計・開発	H17.4 ～ H19.3
小林 卓嗣	同	博士前期課程	セキュリティ機構 の設計・開発	H18.4 ～ H20.3

島田 明男	同	同	サーバ用基盤ソフトウェアの実装	H18.4 ～ H20.3
吉田 哲也	同	同	サーバ用基盤ソフトウェアの実装	H19.4 ～ H20.3
Pushkar Raj Rajkarnikar	同	同	セキュリティ機構 の設計・開発	H19.4 ～ H20.3
阿部 芳久	同	同	サーバ用基盤ソフトウェアの実装	H19.4 ～ H20.3
岩田 聡	同	同	サーバ用基盤ソフトウェアの実装	H19.4 ～ H20.3
北野 雄大	同	同	セキュリティ機構 の設計・開発	H19.4 ～ H20.3
小菅 祐史	同	同	セキュリティ機構 の設計・開発	H19.4 ～ H20.3
田上 歩	同	同	セキュリティ機構 の設計・開発	H19.4 ～ H20.3
横山 敏博	同	同	セキュリティ機構 の設計・開発	H19.4 ～ H20.3

研究グループ名： アプリケーション用基盤ソフトウェア研究グループ(千葉 滋)

氏名	所属	役職(身分)	研究項目	参加時期
千葉 滋	東京工業大学情報理工学研究科	准教授	アプリケーション用基盤ソフトウェア	H14.11 ～ H.20.3
光来健一	同	助教	アプリケーション用基盤ソフトウェア	H15.10 ～ H.20.3
横田大輔	筑波大学大学院工学研究科	博士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H14.11 ～ H15.3
佐藤芳樹	東京工業大学情報理工学研究科	博士課程退学	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H14.11 ～ H17.3

西澤無我	同	博士課程	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H14.11 ～ H16.5.31 H17.4.1～ H.20.3
栗田 亮	同	修士課程修了	アプリケーション用基盤ソフトウェア	H14.11 ～ H16.3
中川清志	同	修士課程修了	アプリケーション用基盤ソフトウェア	H14.11 ～ H16.3
宇崎央泰	同	修士課程退学	アプリケーション用基盤ソフトウェア	H14.11 ～ H17.3
柳澤佳里	同	博士課程	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H15.4 ～ H.20.3
須永 豊	同	修士課程修了	アプリケーション用基盤ソフトウェア	H15.4 ～ H17.3
松沼正浩	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H15.4 ～ H17.3
石川 零	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H16.4 ～ H18.3
薄井義行	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H16.4 ～ H18.3
日比野秀章	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H16.4 ～ H18.3
青木康博	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解	H17.4 ～

			析・実験補助	H.19.3
熊原奈津子	同	修士課程修了	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H17.4 ～ H.19.3
竹内秀行	同	修士課程	アプリケーション用基盤ソフトウェア	H17.4 ～ H.20.3
栗田洋輔	同	修士課程	アプリケーション用基盤ソフトウェア	H17.4 ～ H.20.3
滝澤裕二	同	修士課程	アプリケーション用基盤ソフトウェア	H17.4 ～ H.20.3
堀江倫大	同	修士課程	基盤ソフトウェアに関する研究データの収集・解析・実験補助	H17.4 ～ H.20.3
今吉竜之介	同	修士課程	アプリケーション用基盤ソフトウェア	H19.2 ～ H.20.3
内河綾	同	修士課程	アプリケーション用基盤ソフトウェア	H19.4 ～ H.20.3

5 招聘した研究者等

氏名(所属、役職)	招聘の目的	滞在先	滞在期間

6 成果発表等

(1) 論文(原著論文)発表

■ 国内学会論文誌・学会誌

(1) 飯塚 大介, バルリ ニコ・デムス, 坂井 修一, 田中 英彦: 値予測の軽量効率

- 化方式の提案と評価,情報処理学会 ACS 論文誌,Vol. 44, No. SIG6 (ACS 1), pp.65-75, June 2003.
- (2) 佐藤 芳樹, 千葉 滋: 効率的な Java Dynamic AOP システムを実現する Just-in-Time Weaver, 情報処理学会論文誌: プログラミング, vol. 44, no. SIG 13 (PRO18), pp.15-24, October 2003.
 - (3) 栗田 亮, 千葉 滋, 光来 健一: 感性を考慮したジョブスケジューリング, 情報処理学会論文誌: コンピューティングシステム, vol.45, no. SIG3(ACS 5), pp.29-37, Mar. 2004.
 - (4) 揚妻 匡邦, 河野 健二, 岩崎 英哉, 益田 隆司: モバイルコード技術によるアプリケーション層プロトコルのユーザ透過な配布機構, 電子情報通信学会論文誌, J86-D-1(6), pp.389-401, 2003年6月.
 - (5) 河野 健二, 金子 済, 清水 謙多郎: 資源濫用攻撃に耐性のある資源管理方式, 情報処理学会論文誌: コンピューティングシステム, Vol.44, No. SIG 11(ACS3), pp.191-200, 2003年8月.
 - (6) 品川 高廣, 河野 健二, 益田 隆司: 細粒度保護ドメインを用いた setuid プログラムの特権コード最小化, コンピュータソフトウェア, Vol.21, No.2, pp.25-31, 2004年3月.
 - (7) Niko Demus Barli, Luong Dinh Hung, Hideyuki Miura, Chitaka Iwama, Daisuke Tashiro, Shuichi Sakai, Hidehiko Tanaka: Cache Coherence Strategies for Speculative Multithreading CMPs: Characterization and Performance Study, 情報処理学会論文誌 コンピューティングシステム, vol. 45, No. SIG 11 (ACS 7), pp.119-132, Oct. 2004.
 - (8) 入江 英嗣, 服部 直也, 高田 正法, 坂井 修一, 田中 英彦: クラスタ型プロセッサのための分散投機メモリアドバンス, 情報処理学会論文誌 コンピューティングシステム(ACS 7), Vol.45, No.SIG11, pp.94-104, Oct, 2004.
 - (9) 村田 光一, 河野 健二, 岩崎 英哉, 益田 隆司: ファイル移送に基づく分散ファイルシステムの設計と実装, コンピュータソフトウェア, Vol. 21, No. 4, pp. 43-48, April 2004.
 - (10) 河野 健二(電通大), 品川 高廣(農工大), ラハト・カビル(電通大): TCPストリームに対するフィルタリングによるインターネット・サーバの安全性向上, 情報処理学会論文誌: コンピューティングシステム, Vol. 46, No. SIG4 (ACS9), pp. 33-44, March 2005.
 - (11) 葛 毅, 櫻井 隆雄, ルオン デイン フォン, 阿部 公輝, 坂井 修一: インターリーブ型剰余乗算回路の評価, 電子情報通信学会論文誌, Vol.J88-A, No.12, pp.1497-1505, Dec, 2005.
 - (12) Naoya Hatta, Niko Demus Barli, Chitaka Iwama, Luong Dinh Hung, Daisuke Tashiro, Shuichi Sakai and Hidehiko Tanaka: Bus Serialization for Reducing

- Power Consumption, 情報処理学会論文誌コンピューティングシステム(ACS 13), Vol.47, No.SIG3, Mar, 2006.
- (13) 揚妻 匡邦, 河野 健二, 岩崎 英哉, 益田 隆司: 需要変化に動的に対応する伸縮自在サーバ群の基本機構, 電子情報通信学会論文誌, Vol. J88-D1, No. 4, pp. 767-779, 2005.
- (14) 光来 健一, 千葉 滋: 仮想的な分散監視環境による安全な侵入検知アーキテクチャ, 情報処理学会論文誌: コンピューティングシステム, vol. 46, No. SIG 16 (ACS 12), pp.108-118, December, 2005.
- (15) 西澤 無我, 千葉 滋: 分散ソフトウェアのテストに適したアスペクト指向言語, 情報処理学会論文誌, 46 巻 7 号, pp.1723-1734, 2005 年 7 月.
- (16) Luong D. Hung and Shuichi Sakai: Dynamic Estimation of Task Level Parallelism with Operating System Support, 情報処理学会論文誌コンピューティングシステム(ACS 14), Vol.47, No.SIG7, pp.43-51, May, 2006.
- (17) 松沼 正浩, 光来 健一, 日比野 秀章, 佐藤 芳樹, 千葉 滋: 過負荷時の Web アプリケーションの性能劣化を改善する Session-level Queue Scheduling, コンピュータソフトウェア, vol.23, no.2, 日本ソフトウェア科学会, pp.199-210, 2006.
- (18) 三輪 忍, 福山 智久, 嶋田 創, 五島 正裕, 中島 康彦, 森 眞一郎, 富田 眞治: パス情報を用いた分岐フィルタ機構, 情報処理学会論文誌コンピューティングシステム(ACS 15), Vol.47, No.SIG12, pp.108-118, Aug, 2006.
- (19) Luong D. Hung, Masahiro Goshima and Shuichi Sakai: Zigzag-HVP: A Cost-effective Technique to Mitigate Soft Errors in Caches with Word-based Access, 情報処理学会論文誌コンピューティングシステム(ACS 16), Vol.47, No.SIG18, pp.44-54, Oct, 2006.
- (20) 松沼 正浩, 光来 健一, 日比野 秀章, 佐藤 芳樹, 千葉 滋: 過負荷時の Web アプリケーションの性能を改善する Session-aware Queue Scheduling, 日本ソフトウェア科学会 論文誌「コンピュータソフトウェア」, Vol.23, No.2, pp.199-210, 2006 年.
- (21) 塩谷 亮太, ルオン デイン フォン, 入江 英嗣, 五島 正裕, 坂井 修一: マルチコア・プロセッサの不均質共有キャッシュにおける LRU 大域置き換えアルゴリズム, 情報処理学会論文誌コンピューティングシステム(ACS 17), Vol.48, No.SIG3, pp.59-74, Feb, 2007.
- (22) 杉木 章義, 河野 健二, 岩崎 英哉: リクエスト待機間隔を考慮したウェブサーバの keep-alive 時間の自動設定, コンピュータソフトウェア, Vol. 24, No. 2, pp. 68-78, 2007.
- (23) 阿部 勝幸, 岩崎 英哉, 河野 健二: アプリケーション層プロトコルの記述に基づく拡張性に優れたプロトコル処理コード生成系, コンピュータソフトウェア, Vol. 24, No. 2, pp. 150-153, 2007.

- (24) 熊原 奈津子, 光来 健一, 千葉 滋: 例外処理のためのアスペクト指向言語, 情報処理学会論文誌:プログラミング、vol. 48, no. SIG 10 (PRO33)、pp.189-198, June 2007.
- (25) 柳澤 佳里, 光来 健一, 千葉 滋, 石川 零: OS カーネル用アスペクト指向システム KLASY 情報処理学会論文誌:プログラミング、vol. 48, no. SIG 10 (PRO33), pp.176-188, June 2007.
- (26) Miyuki Hanaoka, Makoto Shimamura, Kenji Kono: TCP Reassembler for Layer7-aware Network Intrusion Detection/Prevention Systems, IEICE Trans. on Information and Systems, Vol. E90-D, No. 12, , pp.2019-2032, Dec., 2007.

■ 国際会議 (査読付・年次論文集有)

- (1) Takahiro Shinagawa, Kenji Kono, Takashi Masuda: A Hierarchical Protection Model for Protecting against Executable Contents, International Conference on Computer Science and Technology, pp. 194-202, May 2003.
- (2) Hideyuki Miura, Luong Dinh Hung, Chitaka Iwama, Daisuke Tashiro, Niko Demus Barli, Shuichi Sakai, and Hidehiko Tanaka: Compiler-Assisted Thread Level Control Speculation, Euro-par 2003 at Klagenfurt, Austria, pp. 603-608, August, 2003.
- (3) Shigeru Chiba and Muga Nishizawa: An Easy-to-Use Toolkit for Efficient Java Bytecode Translators, Proc. of 2nd Int'l Conf. on Generative Programming and Component Engineering (GPCE '03), 22-25, September 2003.
- (4) Yoshiki Sato, Shigeru Chiba, Michiaki Tatsubori,: A Selective, Just-In-Time Aspect Weaver , Proc. of 2nd Int'l Conf. on Generative Programming and Component Engineering (GPCE '03), September 22-25, 2003.
- (5) Yoshimitsu Yanagawa, Luong Dinh Hung, Chitaka Iwama, Niko Demus Barli, Shuichi Sakai, Hidehiko Tanaka: Complexity Analysis of A Cache Controller for Speculative Multithreading Chip Multiprocessors, International Conference on High Performance Computing (HiPC) at Hyderabad in India, vol. HiPC2003, pp.393-404, Dec. 2003.
- (6) Masakuni Agetsuma, Kenji Kono, Hideya Iwasaki, Takashi Masuda: Self-configurable Mirror Servers for Automatic Adaptation to Service Demand Fluctuation, Springer, Lecture Notes in Computer Science (LNCS) 2896, Advances in Computing Science - ASIAN 2003, pp.18--32, Dec. 2003.
- (7) Takahiro Shinagawa, Kenji Kono: Implementing a Secure Setuid Program, International Conference on Parallel and Distributed Computing and Networks, pp.301-309, Feb. 2004.

- (8) Shigeru Chiba, Kiyoshi Nakagawa: Josh: An Open AspectJ-like Language, Proc. of 3rd Int'l Conf. on Aspect Oriented Software Development (AOSD '04), pp.102-111, March 22-26, 2004, Lancaster, UK.
- (9) Muga Nishizawa, Shigeru Chiba, Michiaki Tatsubori: Remote Pointcut -- A language construct for distributed AOP, Proc. of 3rd Int'l Conf. on Aspect Oriented Software Development (AOSD '04), pp. 7-15, March 22-26, 2004, Lancaster, UK.
- (10) Luong Dinh Hung, Chitaka Iwama, Niko Demus Barli, Naoya Hattori, Shuichi Sakai, Hidehiko Tanaka: Dynamic Cache Way Allocation for Power Reduction, A International Symposium on Low-Power and High-Speed Chips(COOL Chips VII), at Yokohama Joho Bunka Center(Yokohama, Japan), Vol.1, pp.268-277, Apr, 2004.
- (11) Yutaka Sugawara, Mari Inaba, Kei Hiraki: Over 10Gbps String Matching Mechanism for Multi-Stream Packet Scanning Systems, Field-Programmable Logic and its Applications (FPL2004), August 2004.
- (12) Yutaka Sugawara: Presentation on FPGA- NIC based Parallel TCP Flow Control, Super Computing 2004 (SC2004), Pittsberg, Nov. 2004.
- (13) Y. Sugawara, M. Inaba and K. Hiraki: High-speed and Memory Efficient TCP Stream Scanning using FPGA, Proc. of 15th Intl. Conf. on Field Programmable Logic and Applications(FPL '05), pp. 45--50, Aug. 2005.
- (14) Hidetsugu Irie, Naoya Hattori, Masanori Takada, Naoya Hatta, Takeshi Toyoshima, Shuichi Sakai: Distributed Speculative Memory Forwarding, IEEE Symp. on Low-Power and High-Speed Chips(COOL Chips VIII), pp.473-482, Apr, 2005.
- (15) Luong Dinh Hung, Masahiro Goshima and Shuichi Sakai: Mitigating Soft Errors in Highly Associative Cache with CAM-based Tag, IEEE International Conference on Computer Design (ICCD 2005), Vol.2005, pp.342-347, Oct, 2005.
- (16) Luong Dinh Hung and Shuichi Sakai: Dynamic Estimation of Task Level Parallelism with Operating System Support, International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN 2005), Vol.2005, pp.358-363, Dec, 2005.
- (17) A. Sugiki, K. Kono and H. Iwasaki: A Practical Approach to Automatic Parameter-Tuning of Web Servers, Springer, Lecture Notes in Computer Science (LNCS) 3818, Advances in Computing Science - ASIAN 2005, pp.146-159, 2005.
- (18) Kenichi Kourai and Shigeru Chiba: HyperSpector: Virtual Distributed

- Monitoring Environments for Secure Intrusion Detection, In Proc. of the 1st ACM/USENIX International Conference on Virtual Execution Environments (VEE'05), pp.197-207, June 2005.
- (19) Shigeru Chiba and Rei Ishikawa: Aspect-Oriented Programming beyond Dependency Injection, ECOOP 2005 -- Object-Oriented Programming, LNCS 3586, Glasgow, pp.121-143, July25-29, 2005.
- (20) Yoshiki Sato and Shigeru Chiba: Loosely-separated "Sister" Namespaces in Java, ECOOP 2005 -- Object-Oriented Programming, LNCS 3586, ECOOP 2005, Glasgow, pp.49-70, July 25-29, 2005.
- (21) Yoshisato Yanagisawa, Kenichi Kourai, Shigeru Chiba and Rei Ishikawa: "A Dynamic Aspect-oriented System for OS Kernels", Proc. of 5th Int'l Conf. on Generative Programming and Component Engineering (GPCE '06), pp.69-78, 2006.
- (22) Hiroshi Yamada, Kenji Kono: User-level disk-bandwidth control for resource-borrowing network applications, In Proc. of IEEE/IFIP Network Operations and Management Symposium, CD-ROM, April, 2006.
- (23) Makoto Shimamura, Kenji Kono: Using Attack Information to Reduce False Positives in Network IDS, In Proc. of IEEE Int'l Symp. on Computers and Communications, pp. 386-393, June, 2006.
- (24) Luong Dinh Hung, Masahiro Goshima and Shuichi Sakai: SEVA: A Soft-Error- and Variation-Aware Cache Architecture, IEEE International Symposium on Pacific Rim Dependable Computing (PRDC 2006), 於 Riverside, California, USA, pp.47-54, Dec, 2006.
- (25) Satoshi Katsunuma, Hiroyuki Kurita, Ryota Shioya, Kazuto Shimizu, Hidetsugu Irie, Masahiro Goshima, and Shuichi Sakai: Base Address Recognition with Data Flow Tracking for Injection Attack Detection, IEEE International Symposium on Pacific Rim Dependable Computing (PRDC 2006), 於 Riverside, California, USA, pp.165-172, Dec, 2006.
- (26) Kenichi Kourai, Hideaki Hibino, and Shigeru Chiba: Aspect-oriented Application-level Scheduling for J2EE Servers, Proc. of the 6th International Conference on Aspect-Oriented Software Development (AOSD '07), ACM Press, pp.1-13, 2007.
- (27) Luong Dinh Hung, Hidetsugu Irie, Masahiro Goshima, and Shuichi Sakai: Utilization of SECDED for Soft Error and Variation-Induced Defect Tolerance in Caches, Proceedings of Design, Automation and Test in Europe 2007 (DATE2007), at Acropolis, Nice, France, pp. 1134-1149, April, 2007.
- (28) Kenichiro Hirose, Yoshio Mita and Shuichi Sakai: Polarization-Transmissive

Thin-Film Solar Cell with Photodiode Nanowires, IEEE/LEOS International Conference of Optical MEMS and Nanophotonics 2007, 於 Fargroly Hotel at Hualien, Taiwan, pp.29-30, Aug. 2007.

- (29) Yousuke Nakamura, Kei Hiraki: Heterogeneous Functional Units for High Speed Fault-tolerant Execution Stage, 13th IEEE Pacific Rim Dependable Computing conference, (PRDC2007) , pp.260-263, Dec. 2007.
- (30) Kenichi Kourai and Shigeru Chiba : A Fast Rejuvenation Technique for Server Consolidation with Virtual Machines, In Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2007), pp.245-254, June 2007.
- (31) Michihiro Horie, Shigeru Chiba: AspectScope: An Outline Viewer for AspectJ Programs, In Proceedings of TOOLS EUROPE 2007: Object, Models, Components and Patterns , paper 17, June 25-27, 2007.
- (32) Hiroshi Yamada, Kenji Kono: FoxyTechnique: Tricking Operating System Policies with a Virtual Machine Monitor, ACM Int'l Conf. on Virtual Execution Environments (VEE '07), pp. 55-64, June, 2007.
- (33) Miyuki Hanaoka, Kenji Kono, Makoto Shimamura, Satoshi Yamaguchi: An Efficient TCP Reassembler Mechanism for Layer7-aware Network Intrusion Detection/Prevention Systems, IEEE Int'l Symp. on Computers and Communications (ISCC '07), pp. 79-86, July, 2007.
- (34) Masato Asahara, Akio Shimada, Hiroshi Yamada, Kenji Kono: Finding Candidate Spots for Replica-Servers based on Demand Fluctuation, IEEE Int'l Conf. on Parallel and Distributed Systems (ICPADS '07), Dec, 2007.

(2) その他の著作物

- (1) Shigeru Chiba: Javassist: Java bytecode engineering made simple, Java Developer's Journal, SYS-CON Media, Inc., pp. 26-28, volume 9, issue 1, January 8, 2004.
- (2) 千葉 滋: Javassist -- Java バイトコードを操作するクラスライブラリ -- 入門、Java Press vol. 35, pp.76-85, March 2004.
- (3) 坂井 修一: コンピュータアーキテクチャ、コロナ社、2004年3月
- (4) 千葉 滋: アスペクト指向入門 - Java・オブジェクト指向から AspectJ プログラミングへ、技術評論社、264 pages, ISBN 4774125814, 2005/11/18
- (5) Shigeru Chiba: Program Transformation with Reflection and Aspect-Oriented Programming, Generative and Transformational Techniques in Software Engineering, LNCS 4143, Springer, pp.65-94, 2006.
- (6) 河野 健二: 未知の攻撃を防御する高精度ネットワーク侵入検知・回避システム,

Cyber Security Management, Vol. 7, No. 78, pp. 83-87, April, 2006.

- (7) 坂井 修一: 違う世界をもとう, 人工知能学会誌 20 卷 1 号, pp.158-160, Jan. 2007.
- (8) 坂井 修一: ディペンダブル情報社会へ, 情報処理, Vol.48, No.7, pp.783-785, Jul. 2007.

(3) 学会発表(国際学会発表及び主要な国内学会発表)

- ① 招待講演 (国内会議 8 件、国際会議 5 件)
- (1) 坂井 修一: スピードからディペンダビリティへ: プロセッサのこれから, 情報処理学会関西支部大会2004, 於 大阪大学中之島センター, 情報処理学会関西支部大会(招待講演)論文集, Vol.H16, No.S-07, pp.143-148, Oct, 2004.
- (2) Shigeru Chiba: Is Academic Open Source Software Dependable?, WCC - 18th IFIP World Computer Congress, Toulouse - France, Aug. 2004.
- (3) Shigeru Chiba: Making Your Own AOP Tools with Javassist, International Workshop on Aspect-Oriented Software Development, 北京, Sep. 2004.
- (4) Shigeru Chiba: Remote Pointcut -- A language construct for distributed AOP, OSCAR Workshop, Santiago, Chile, Nov. 2004.
- (5) 坂井 修一: ディペンダブル情報処理基盤, 電子情報通信学会, 計算機システム研究会 招待講演, Apr. 2005.
- (6) Shigeru Chiba: Aspect-Oriented Programming beyond Dependency Injection, Asian Workshop on AOSD, Twelfth Asia-Pacific Software Engineering Conference (APSEC), not published, Dec. 2005.
- (7) 坂井 修一, 中村 宏, 五島 正裕, 松岡 聡, 橋本 幹生, 小檜山 清之, 中村 友洋: ディペンダビリティとセキュリティ - デバイス、アーキテクチャ、ソフトウェア, 電子情報通信学会技術研究報告 CPSY2006-1~12(パネル討論), 於 東京大学, Vol.106, No.3, pp.67, Apr, 2006.
- (8) 坂井 修一: ディペンダブルプロセッサ - アーキテクトの視点から -, 独立行政法人 日本学術振興会 (JSPS) シリコン超集積化システム第 165 委員会第 41 回研究会 (招待公演), 於 東京, 研究会予稿集, pp.6-20, May, 2006.
- (9) 五島 正裕: マイクロプロセッサの研究動向, 半導体理工学研究センター (STARC) 第 17 回 STARC アドバンス講座 システムアーキテクチャセミナー, 於 ホテルフロラシオン青山, pp.69-88, Jun, 2006.
- (10) 坂井 修一: 超ディペンダブルプロセッサ, 半導体理工学研究センター (STARC) 第 18 回 STARC アドバンス講座 システムアーキテクチャセミナー, 於 新横浜国際ホテル, pp.65-82, Jul, 2006.
- (11) 五島 正裕: コアの数なんてどうでもいい, 第 5 回情報科学技術フォーラム (FIT) 2006, 於 福岡大学, Sep, 2006.
- (12) 中村 宏, 京 昭倫, 佐藤 寿倫, 大津 金光, 井上 弘士, 五島 正裕, 木村 啓二, 吉瀬 謙二: これからが面白いプロセッサアーキテクチャ, 第 5 回情報科学技術フォーラム (FIT) 2006 (パネル討論), 於 福岡大学, Sep, 2006.
- (13) Shuichi Sakai, Masahiro Goshima, Hidetsugu Irie, Yutaka Sugawara, Kenji Kono, and Shigeru Chiba: Dependable Computing: From Architecture to Internet Applications, IFIP International Conference on Network and Parallel Computing (Keynote Talk), 於 The University of Tokyo, Oct, 2006.

(4) 口頭発表 (国内会議 85 件、国際会議 11 件)

■国内シンポジウム(査読付)

- (1) Niko Demus Barli, Daisuke Tashiro, Chitaka Iwama, Shuichi Sakai, and Hidehiko Tanaka: A Register Communication Mechanism for Speculative Multithreading Chip Multiprocessors , Symposium on Advanced Computing Systems and Infrastructures (SAC SIS) 於国立情報学研究所, pp.275-282, May 2003.
- (2) 杉木 章義, 河野 健二, 岩崎 英哉, 益田 隆司: ディペンダブルなインターネット・サーバを実現するクラスタ用ミドルウェアの基本設計, 情報処理学会 Symposium on Advanced Computing Systems and Infrastructures (SAC SIS), pp.267-274, 2003年5月.
- (3) Niko Demus Barli, Luong Dinh Hung, Hideyuki Miura, Chitaka Iwama, Daisuke Tashiro, Shuichi Sakai and Hidehiko Tanaka: Cache Coherence Strategies for Speculative Multithreading CMPs: Characterization and Performance Study, 先進的計算基盤システムシンポジウム2004(SAC SIS2004), 於札幌コンベンションセンター, Vol.2004, No.6, pp.111-120, May, 2004.
- (4) 入江 英嗣、服部 直也、高田 正法、坂井 修一、田中 英彦: クラスタ型プロセッサのための分散投機メモリアワードニング, 先進的計算基盤システムシンポジウム2004(SAC SIS2004), 於札幌コンベンションセンター, vol. 2004, No.6, pp.177-186, May 2004.
- (5) 河野 健二(電通大), 品川 高廣 (農工大), ラハト・カビル: TCPストリームに対するフィルタリングによるインターネット・サーバの安全性向上, 情報処理学会 コンピュータシステムシンポジウム, pp. 75-84, Nov. 2004.
- (6) 塩谷 亮太, ルオン ディン フォン, 入江 英嗣, 五島 正裕, 坂井 修一: マルチコア・プロセッサの不均質共有キャッシュにおける LRU 大域置き換えアルゴリズム, 先進的計算基盤システムシンポジウム 2006(SAC SIS2006), 於大阪国際会議場(グランキューブ大阪), Vol.2006, No.5, pp. 23-31, May, 2006.
- (7) 三輪 忍, 福山 智久, 嶋田 創, 五島 正裕, 中島 康彦, 森 眞一郎, 富田 眞治: パス情報を用いた分岐フィルタ機構, 先進的計算基盤システムシンポジウム 2006(SAC SIS2006), 於大阪国際会議場(グランキューブ大阪), Vol.2006, No.5, pp. 315-324, May, 2006.
- (8) 豊島 隆志, 入江 英嗣, 五島 正裕, 坂井 修一: レジスタ間接分岐ターゲットフォワードニング, 先進的計算基盤システムシンポジウム 2006(SAC SIS2006), 於大阪国際会議場(グランキューブ大阪), Vol.2006, No.5, pp. 325-332, May, 2006.
- (9) 品川 高廣, 河野 健二: 実行時のフェーズを考慮したセキュリティポリシー記述の簡略化, 先進的計算基盤システムシンポジウム 2006(SAC SIS2006), 於大阪国際会議場(グランキューブ大阪), Vol.2006, pp. 495-503, 2006.
- (10) 勝沼 聡, 栗田 弘之, 塩谷 亮太, 清水 一人, 入江 英嗣, 五島 正裕, 坂井

- 修一：アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出，先進的計算基盤システムシンポジウム 2006(SACSIS2006)，於 大阪国際会議場(グランキューブ大阪)，Vol.2006，No.5，pp.515-524，May，2006.
- (11) 入江 英嗣，杉本 健，五島 正裕，坂井 修一：レジスタファイルの書き込み時タイミングエラーの検出・回復手法，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol.2007，No.5，pp. 235-244，May，2007.
- (12) 三輪 忍，一林 宏憲，入江 英嗣，五島 正裕，富田 眞治：小容量RAMを用いたオペランド・バイパスの複雑さの低減手法，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol.2007，No.5，pp. 265-274，May，2007.
- (13) 堀尾 一生，平井 遥，五島 正裕，坂井 修一：ツインテール・アーキテクチャ，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol.2007，No.5，pp. 303-311，May，2007.
- (14) 脇屋 進一，五島 正裕，坂井 修一：分岐予測器を利用したホットパス検出器，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol.2007，No.5，pp. 321-328，May，2007.
- (15) 渡辺 憲一，一林 宏憲，五島 正裕，坂井 修一：プロセッサ・シミュレータ「鬼斬」の設計，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol. 2007，No. 5，pp.194-195，May 2007.
- (16) 杉本 健，入江 英嗣，五島 正裕，坂井修一：Out-of-Order スーパースカラプロセッサの FPGA への実装，先進的計算基盤システムシンポジウム 2007(SACSIS2007)，於 学術総合センター講堂・会議室(東京)，Vol. 2007，No. 5，pp.196-197，May 2007.

■国際ワークショップ

- (1) Hidetsugu Irie, Naoya Hattori, Masanori Takada, Takashi Toyoshima, Naoya Hatta, Shota Watanabe, Shuichi Sakai: Steering and Forwarding Techniques for Reducing Memory Communication on Clustered Microarchitecture, International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems 2005, vol.2005, pp.8-10, Jan.2005.
- (2) Shigeru Chiba: Generative Programming from a Post Object-Oriented Programming Viewpoint, UPP'04 (An EU-NSF workshop on Unconventional Programming Paradigms), France, Sep. 2004.
- (3) Y. Sato and S. Chiba: The Negligent Class Loaders for Software Evolution, ECOOP2004(第18回オブジェクト指向に関するヨーロッパ会議)のRAM-SE Workshop (オスロ、ノルウェー), Proceedings of RAM-SE'04 - ECOOP'04 Workshop on Reflection, AOP, and Meta-Data for Software Evolution, Oslo, 15 June (2004) 53-58, June 2004.
- (4) Y. Usui: A software development environment to enable debug code insertion by using Aspect-Oriented programming, AOSD2005(International Conference

- Aspect-Oriented Software Development (シカゴ、USA), International Conference Aspect-Oriented Software Development 2005 (AOSD.05), Student Extravaganza No.17, Mar. 2005.
- (5) Y.Yanagisawa: A Source-level Kernel Profiler based on Dynamic Aspect Orientation, AOSD2005(International Conference Aspect-Oriented Software Development (シカゴ、USA), Research Institute for Advanced Computer Science Technical Report, Proceeding of the Second Dynamic Aspects Workshop (DAW05)Robert E. Filman, Michael Haupt and Robert Hirschfeld Technical Report 05.01 pp.41-45, Mar. 2005.
 - (6) Yoshiyuki Usui, Shigeru Chiba: Bugdel: An Aspect-Oriented Debugging System, First Asian Workshop on AOSD, In Proc. of 12th Asia-Pacific Software Engineering Conference (APSEC 2005), IEEE Press, pp.790-795, Taipei, Taiwan, December 15 - 17, 2005.
 - (7) Shigeru Chiba: Generative Programming from a Post Object-Oriented Programming Viewpoint, UPP 2004, LNCS 3566, Springer-Verlag, pp.342-353, 2005.
 - (8) Hideaki Hibino, Kenichi Kourai, Shigeru Chiba: Difference of Degradation Schemes among Operating Systems--- Experimental analysis for web application servers ---, In Proc. of DSN 2005 Workshop on Dependable Software - Tools and Methods, pp.172-179, June 28 - July 1, 2005.
 - (9) Michihiro Horie and Shigeru Chiba: An Aspect-Aware Outline Viewer, 3rd ECOOP Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE'06), 2006.
 - (10) Yasuhiro Aoki and Sigeru Chiba: "Performance improvement for persistent systems by AOP", Software Engineering Properties of Languages and Aspect Technologies (SPLAT! 2007), 2007.
 - (11) Hidetsugu Irie and Ken Sugimoto and Masahiro Goshima and Shuichi Sakai: Preventing Timing Errors on Register Writes:Mechanisms of Detections and Recoveries, 2nd International Workshop on Advanced Low Power Systems, at Crowne Plaza Seattle at Seattle, WA, USA, pp. 31-38, Jun, 2007.

■研究会・全国大会

- (1) 高田 正法, 上村 明, 森 晃平, 岡部 淳, 坂井 修一, 田中 英彦: 空間分割を行う再構成可能ハードウェアにおける動的資源割り当て, 情報処理学会報告書 計算機アーキテクチャ研究会 2003-ARC-153, 於慶應大学, pp. 13-18, May 2003.
- (2) 田代 大輔, バルリ ニコ デムス, 坂井 修一, 田中 英彦: スレッド投機実行におけるエッジに着目したスレッド分割手法、第 145 回計算機アーキテクチャ研究発表会 於慶應義塾大学日吉キャンパス pp. 67-72, May 2003.

- (3) 岩間 智女、ルオン デイン フォン、バルリ ニコ デムス、坂井 修一、田中 英彦: The Design of PRESTO: A Framework For Architecture Level Power Estimation, 第 146 回計算機アーキテクチャ(ARC)研究会 (SWOPP 松江 2003), pp.103-108, Aug. 2003.
- (4) 柳澤 佳里、千葉 滋: 他のプロセスに与える影響が少ない実行時ミラーリングシステム、並列/分散/協調処理に関するサマー・ワークショップ(SWoPP '03)、第 94 回 OS 研究会、松江, pp.83-89、2003 年 8 月.
- (5) 西澤 無我、千葉 滋: 分散 Java プログラミングのためのアスペクト指向言語、並列/分散/協調処理に関するサマー・ワークショップ(SWoPP '03)、第 45 回プログラミング研究会、松江, 2003 年 8 月.
- (6) 佐藤 芳樹, 立堀 道昭, 千葉 滋: オンデマンドのアスペクト・ウィービングに基づく、Web アプリケーション開発のすすめ、オブジェクト指向最前線 2003、情報処理学会 OO2003 シンポジウム、近代科学社、pp.125-128 (short paper)、2003 年 8 月.
- (7) 杉木 章義, 河野 健二, 岩崎 英哉, 益田 隆司: ディペンダブルなインターネット・サーバを構築するためのカスタマイズ可能なクラスタ用ツールキット, 日本ソフトウェア科学会 第 20 回大会論文集 (CD-ROM), 2003 年 9 月.
- (8) 村田 光一, 河野 健二, 岩崎 英哉, 益田 隆司: ファイル移送に基づく分散ファイルシステムの設計と実装, 日本ソフトウェア科学会 第 20 回大会論文集 (CD-ROM), 2003 年 9 月.
- (9) Luong Dinh Hung, 岩間 智女, Niko Demus Barli, 坂井 修一, 田中 英彦: Way-variable Caches for Static Power Reduction、情報処理学会 計算機アーキテクチャ(ARC)研究会デザインガイア 於北九州国際会議場, vol. 2003, No.119, pp.87-92, Nov. 2003.
- (10) 渡辺 元晴, 河野 健二, 岩崎 英哉, 益田 隆司: P2P 型ファイル検索における高スループット・ピアの自動選択機構, 情報処理学会研究会報告(2004-OS-95), pp.65-72, 2004 年 2 月.
- (11) 松沼 正浩, 千葉 滋, 佐藤 芳樹, 光来 健一: 過負荷時の Web アプリケーションの性能劣化を改善する Page-level Queue Sceduling、第 7 回プログラミングおよび応用のシステムに関するワークショップ (SPA2004)、日本ソフトウェア科学会ソフトウェアシステム研究会、2004 年 3 月 1 日~3 日 (長野県 上諏訪温泉).
- (12) 西澤 無我, 千葉 滋, 立堀 道昭: 遠隔ポイントカット --- 分散アスペクト指向プログラミングのための言語機構、研究紹介、第 7 回プログラミングおよび応用のシステムに関するワークショップ (SPA2004)、日本ソフトウェア科学会ソフトウェアシステム研究会、2004 年 3 月 1 日~3 日 (長野県 上諏訪温泉).
- (13) 小倉 章嗣, 河野 健二, 松岡 聡, 中田 秀基: グリッド環境における VM 上でのジョブ実行の検討, 情報処理学会研究会報告(2004-HPC-97), pp.25-30, 2004

年 3 月.

- (14) 初田 直也(東大), バルリ ニコ デムス(日本 TI), 岩間 智女, ルオン デイン フォン, 田代 大輔, 坂井 修一(東大), 田中 英彦(情報セキュリティ大): Bus Serialization for Reducing Power Consumption, 情報処理学会報告 2004-ARC-159, 於 青森文化会館, Vol.2004, No.80, pp.163-168, Jul, 2004.
- (15) 入江 英嗣(東大/JST), 高田 正法(東大), 坂井 修一(東大): メモリ依存予測を利用したフォワーディング局所化手法, 情報処理学会報告 2004-ARC-159, 於 青森文化会館, Vol.2004, No.80, pp.49-54, Jul, 2004.
- (16) 菅原 豊, 稲葉 真理, 平木 敬: インテリジェント NIC を用いた高帯域ネットワーク向け TCP 通信方式、SWoPP 青森 2004、Aug. 2004.
- (17) Luong D. Hung, Yi Ge, Takada Masanori, Shuichi Sakai: A cost-effective Technique to Mitigate Soft Errors in Logic Circuits, 電子情報通信学会技術研究報告 VLSI 設計技術 VLD2004-49~60, 於 北九州国際会議場, Vol.104, No.477, pp.31-36, Dec, 2004.
- (18) 豊島 隆志(東大), 田代 大輔(東大), バルリ ニコ デムス(日本TI), 坂井 修一(東大): メモリ投機を支援するCMPキャッシュコヒーレンスプロトコルの検討, 情報処理学会報告 計算機アーキテクチャ 2004-ARC-160, 於 北九州国際会議場, Vol.2004, No.123, pp.47-52, Dec, 2004.
- (19) 高田 正法, 入江 英嗣, 服部 直哉, 渡辺 翔太, 清水 一人, 坂井 修一: クラスタ型プロセッサにおけるSMT実行, 情報処理学会報告 計算機アーキテクチャ 2005-ARC-162, 於 北海道大学学術交流会館小講堂, Vol.2005, No.19, pp.31-36, Mar, 2005.
- (20) 渡辺 翔太, 入江 英嗣, 高田 正法, 坂井 修一: クラスタ型スーパースカラプロセッサにおけるストア命令の早期発行手法, 情報処理学会報告 計算機アーキテクチャ 2005-ARC-162, 於 北海道大学学術交流会館小講堂, Vol.2005, No.19, pp.199-204, Mar, 2005.
- (21) 杉木 章義、河野 健二、岩崎 英哉: ロード・アブソーバを用いた高性能なインターネットサーバの実現, 第3回SPAサマワークショップ (SPA-SUMMER 2004), Aug. 2004.
- (22) 山田 浩史, 河野 健二: ユーザレベルでのディスク帯域制御機構, 日本ソフトウェア科学会 ソフトウェアシステム研究会, March 2005.
- (23) 嶋村 誠, 河野 健二: Network IDS の攻撃検知情報を利用したサーバの安全性向上, 情報処理学会 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, pp. 57 -64, Feb. 2005.
- (24) Yuya Ueno, Luong D. Hung, Masanori Takada, Daisuke Tashiro and Shuichi Sakai: Improvement of Signature-based Phase Detection and its Application to Power Reduction in Caches, 電子情報通信学会技術研究報告

- RECONF2005-1~14, Vol.105, No.42, pp.25-30, May, 2005.
- (25) 清水 一人, 高田 正法, 入江 英嗣, 坂井 修一: プログラムの振る舞い秘匿のための動的アドレス変換, 情報処理学会研究報告書 計算機アーキテクチャ研究会 2005-ARC-164, Vol.2005, No.80, pp.19-24, Aug, 2005.
- (26) 門馬 太平, ルオン デイン フォン, 田代 大輔, 坂井 修一: スレッド投機実行のためのキャッシュコヒーレンシプロトコルの検証, 情報処理学会研究報告書 計算機アーキテクチャ研究会 2005-ARC-164, Vol.2005, No.80, pp.103-108, Aug, 2005.
- (27) Luong D. Hung, Masahiro Goshima and Shuichi Sakai: Technique to Mitigate Soft Errors in Caches with CAM-based Tags, 電子情報通信学会技術研究報告 DC2005-18, Vol.105, No.227, pp.31-36, Aug, 2005.
- (28) 菅原 豊, 稲葉 真理, 平木 敬: 動的再構成を用いたアプリケーションレイヤ処理エンジンの設計, デザインガイア 2005 (リコンフィギャラブル研究会), 電子情報通信学会技術研究報告 RECONF2005-59~71, pp. 7-12, Dec. 2005.
- (29) 中村 洋介, 平木 敬: リコンフィギュレーションによるFPGAの固定故障検出, デザインガイア 2005 (リコンフィギャラブル研究会), 電子情報通信学会技術研究報告 RECONF2005-59~71, pp. 31-36, Dec. 2005.
- (30) 中村 洋介, 平木 敬: LUT とマイクロプログラムによる耐固定故障プロセッサ, SwoPP2005(ディペンダブルコンピューティング研究会), 電子情報通信学会技術研究報告 DC2005-59~71, pp. 25-30, Aug. 2005.
- (31) 杉木 章義, 河野 健二: 性能パラメータの自動調整による Web サーバの性能向上, 情報処理学会 OS 研究会報告(2005-OS-99), pp.29-36, 2005.
- (32) 花岡 美幸, 河野 健二: アプリケーション層プロトコルに対するパケット・レベルでのフィルタリング, 情報処理学会 OS 研究会報告(2005-OS-99), pp.91-98, 2005.
- (33) 山口 聖司, 河野 健二: アプリケーション層プロトコルの文脈を考慮した NIDS, 情報処理学会 OS 研究会報告(2005-OS-100), pp.9-16, 2005.
- (34) 阿部 勝幸, 岩崎 英哉, 河野 健二: アプリケーション層プロトコルの記述に基づく拡張性に優れたプロトコル処理コード生成系, 日本ソフトウェア科学会 第 22 回大会論文集, 2005. (CD-ROM のためページ番号はありません).
- (35) 杉木 章義, 河野 健二, 岩崎 英哉 : リクエスト間隔を考慮したウェブサーバの keep-alive 時間の自動設定, 日本ソフトウェア科学会 第 22 回大会論文集, 2005. (CD-ROM のためページ番号はありません).
- (36) 日比野 秀章, 光来 健一, 千葉 滋 : J2EE アプリケーションにおけるアプリケーションレベルスケジューリング, 第 3 回ディペンダブルソフトウェアワークショップ (DSW'06), 日本ソフトウェア科学会, 東京大学, 2006 年 1 月.
- (37) 光来 健一, 千葉 滋: 仮想計算機を用いたサーバ統合における高速なリポートリカバリ, 第 9 回プログラミングおよび応用のシステムに関するワークショップ (SPA

- 2006), 日本ソフトウェア科学会, 2006年3月.
- (38) 熊原 奈津子, 石川 零, 西澤 無我, 光来 健一, 千葉 滋: Recovery アドバイスをもつアスペクト指向システム, 第9回 プログラミングおよび応用のシステムに関するワークショップ(SPA 2006), 日本ソフトウェア科学会, 2006年3月.
- (39) 柳澤 佳里, 光来 健一, 千葉 滋: アスペクト指向を用いたカーネルプロファイラ, 第99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, 沖縄県 ホテルムーンビーチ, 情報処理学会研究報告 2005-OS-99, pp.149-156, 2005年5月.
- (40) 青木 康博, 千葉 滋, 佐藤 芳樹: アスペクト指向を利用した永続オブジェクト・アクセスの高速化, 日本ソフトウェア科学会第22回大会、東北大学、2005年9月.
- (41) 石川 零, 千葉 滋: アスペクト指向プログラミングと Dependency Injection の融合, 情報処理学会 第56回プログラミング研究会 (PRO-2005-3) 玉原高原, 2005年10月.
- (42) 入江 英嗣, 荻野 健, 勝沼 聡, 清水 一人, 栗田 弘之, 五島 正裕, 坂井 修一: 超ディペンダブル・プロセッサアーキテクチャの構想, 電子情報通信学会技術研究報告 CPSY2006-1~12, 於 東京大学, Vol.106, No.3, pp.49-54, Apr, 2006.
- (43) 杉木 章義, 河野 健二: 応答時間の分散を利用したウェブサーバ接続数の自動設定, 情報処理学会研究会報告(2006-OS-102), pp. 37-44, 2006年5月.
- (44) 浅原 理人, 島田 明男, 山田 浩史, 河野 健二: 需要変動に応じたコンテンツの再配置を行う Content Delivery Network, 情報処理学会研究会報告(2006-OS-102), pp. 85--92, 2006年5月.
- (45) 杉木 章義, 河野 健二: ウェブサーバの主要パラメータ自動設定機構, プログラミングおよび応用のシステムに関するワークショップ(SPAX), 日本ソフトウェア科学会, 2006年.
- (46) 堀江 倫大, 千葉 滋: AspectScope によるアスペクトとクラスのつながりの視覚化, 第10回 プログラミングおよび応用のシステムに関するワークショップ (SPA X), 日本ソフトウェア科学会・ソフトウェアシステム研究会、2006.
- (47) 滝澤 裕二, 光来 健一, 柳澤 佳里, 千葉 滋: OS が乗っ取られた場合にも機能するファイルアクセス制御システム, 第10回 プログラミングおよび応用のシステムに関するワークショップ (SPA X)、日本ソフトウェア科学会・ソフトウェアシステム研究会, 2006.
- (48) 荻野 健, 入江 英嗣, 五島 正裕, 坂井 修一: レジスタファイルの書き込み時のタイミングエラーの検出・回復手法, 情報処理学会研究報告書 計算機アーキテクチャ研究会 200-ARC-169, 於 高知商工会館(高知県高知市), Vol.2006, No.88, pp. 19-24, Aug, 2006.
- (49) 渡辺 憲一, 五島 正裕, 坂井 修一: ループを並列実行するクラスタ型アーキテ

- クチャ, 情報処理学会研究報告書 計算機アーキテクチャ研究会 200-ARC-169, 於 高知商工会館(高知県高知市), Vol.2006, No.88, pp. 25-30, Aug, 2006.
- (50) 一林 宏憲, 入江 英嗣, 五島 正裕, 坂井 修一: 逆 Dualflow アーキテクチャ, 情報処理学会研究報告書 計算機アーキテクチャ研究会 200-ARC-169, 於 高知商工会館(高知県高知市), Vol.2006, No.88, pp. 37-42, Aug, 2006.
- (51) 平井 遥, 入江 英嗣, 五島 正裕, 坂井 修一: ツインテール・アーキテクチャ, 情報処理学会研究報告書 計算機アーキテクチャ研究会 200-ARC-169, 於 高知商工会館(高知県高知市), Vol.2006, No.88, pp. 43-48, Aug, 2006.
- (52) 福山 智久, 三輪 忍, 嶋田 創, 五島 正裕, 中島 康彦, 森 眞一郎, 富田 眞治: スラック予測を用いたクラスタ型スーパースカラ・プロセッサ向け命令ステアリング, 情報処理学会研究報告書 計算機アーキテクチャ研究会 200-ARC-169, 於 高知商工会館(高知県高知市), Vol.2006, No.88, pp. 55-60, Aug, 2006.
- (53) 栗田 弘之, 塩谷 亮太, 入江 英嗣, 五島 正裕, 坂井 修一: 動的なインフォメーションフロー制御による情報漏洩防止手法, 情報処理学会報告 2007-ARC-172, 於 北海道大学学術交流会館, Vol.2007, No.17, pp.227-232, Mar, 2007.
- (54) 清水 一人, 入江 英嗣, 五島 正裕, 坂井 修一: 耐ソフトウェアタンパ・プロセッサ, 情報処理学会報告 2007-ARC-172, 於 北海道大学学術交流会館, Vol.2007, No.17, pp.239-244, Mar, 2007.
- (55) 入江 英嗣, 杉本 健, 五島 正裕, 坂井 修一: 動的タイミングエラー検出のための「書き込み保証バッファ」の評価, 電子情報通信学会研究報告 ICD2007-29, 於 富士通研究所(神奈川県川崎市), Vol.107, pp. 77-78, Jun, 2007.
- (56) 一林 宏憲, 入江 英嗣, 五島 正裕, 坂井 修一: 逆 Dualflow アーキテクチャ, 情報処理学会研究報告書 計算機アーキテクチャ研究会, 2007-ARC-174, 於大雪クリスタルホール(北海道旭川市), Vol. 2007, No.79, pp.1-6. Aug. 2007.
- (57) 亘理 靖展, 堀尾 一生, 入江 英嗣, 五島 正裕, 坂井 修一: ツインテール・アーキテクチャの改良, 情報処理学会研究報告書 計算機アーキテクチャ研究会, 2007-ARC-174, 於大雪クリスタルホール(北海道旭川市), Vol. 2007, No.79, pp.7-12, Aug. 2007.
- (58) 金 大雄, 入江 英嗣, 五島 正裕, 坂井 修一: タグ・アーキテクチャのための効率的なタグ管理機構, 電子情報通信学会技術研究報告書 コンピュータシステム研究会, CPSY2007-10, 於大雪クリスタルホール(北海道旭川市), Vol. 2007, No.175, pp.25-30, Aug.2007.
- (59) 樽井 翔, 勝沼 聡, 入江 英嗣, 五島 正裕, 坂井 修一: 脆弱性検出のための静的値範囲解析, コンピュータシステム研究会, CPSY2007-22, 於大雪クリスタルホール(北海道旭川市), Vol. 2007, No.175, pp.95-100, Aug. 2007.

- (60) 中村 洋介, 平木 敬: ヘテロジニアス演算器構成による高耐故障実行ステージ SWoPP2007, DC9 pp19-24, Aug. 2007.
- (61) 今吉 竜之介, 柳澤 佳里, 千葉 滋: アスペクト指向言語のための独立性の高いパッケージシステム, 第 11 回 プログラミングおよび応用のシステムに関するワークショップ (SPA 2007), 愛知県 三谷温泉, 2007 年 9 月.
- (62) 栗田 洋輔, 千葉 滋: C 言語を用いたマシン非依存な JIT コンパイラ作成フレームワーク, 第 11 回 プログラミングおよび応用のシステムに関するワークショップ (SPA 2007), 愛知県 三谷温泉, 2007 年 9 月.
- (63) 光来 健一, 千葉 滋: 仮想計算機を用いたサーバ統合における高速なソフトウェア若化手法, 第 5 回 ディペンダブルシステムワークショップ (DSW'07summer), pp.45-52, 2007 年 7 月.
- (64) 滝澤 裕二, 光来 健一, 柳澤 佳里, 千葉 滋: 仮想計算機を用いたファイルアクセス制御の二重化, 第 5 回 ディペンダブルシステムワークショップ (DSW'07summer), pp.75-83, 2007 年 7 月.
- (65) 柳澤 佳里, 光来 健一, 千葉 滋, 石川 零: OS カーネル用アスペクト指向システム KLASY, 第 62 回 プログラミング研究会 (PRO-2006-4), 沖縄県 メルパルク 沖縄, 2007 年 1 月.
- (66) 熊原 奈津子, 光来 健一, 千葉 滋: 例外処理のためのアスペクト指向言語, 第 62 回 プログラミング研究会 (PRO-2006-4), 沖縄県 メルパルク 沖縄, 2007 年 1 月.
- (67) 山田 浩史, 河野 健二: Foxy Technique: 仮想機械モニタによる OS の資源管理ポリシーの変更, 情報処理学会研究会報告(2007-OS-105), pp. 79-86, 2007.
- (68) 浅原 理人, 島田 明男, 山田 浩史, 河野 健二: 需要変動に応じた動的再配置が可能なミラーサーバ基盤, 情報処理学会研究会報告(2007-OS-105), pp. 123-130, 2007.

■テクニカルレポート

- (1) 石川 零, 千葉 滋: アスペクト指向プログラミングと Dependency Injection の融合, Dept. of Math. and Comp. Sciences Research Reports C-220, Tokyo Institute of Technology, February, 2006.

(5) ポスター発表 (国内会議 12 件、国際会議 6 件)

- (1) Muga Nishizawa and Shigeru Chiba: A Distributed AOP System for J2EE Applications, European Conference on Object-Oriented Programming 2005, Glasgow UK, July 2005.
- (2) Rei Ishikawa and Shigeru Chiba: GluonJ: An AOP System Dealing with Dependency among Components, European Conference on Object-Orientated

- Programming (ECOOP) 2005, Glasgow UK, July 2005.
- (3) Muga Nishizawa and Shigeru Chiba: A Distributed AOP System for J2EE Applications, 4th SPA SUMMER Workshop 2005, 2005 年 8 月.
 - (4) 石川 零, 千葉 滋: GluonJ: コンポーネント間の依存関係を取り除くための AOP システム, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (5) 薄井 義行, 千葉 滋: 統合開発環境の上に構築したアスペクト指向システム, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (6) 日比野 秀章, 千葉 滋: アプリケーションサーバにおけるパフォーマンス制御, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (7) 青木 康博, 千葉 滋, 佐藤 芳樹: アスペクト指向を利用した永続オブジェクト・アクセスの高速化, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (8) 熊原 奈津子, 千葉 滋: 例外処理をアドバースとして書けるアスペクト指向言語, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (9) 竹内 秀行, 千葉 滋: Symmetric Class Weaving: クラスの統合と操作・属性の共有, 第 4 回 SPA サマーワークショップ, 石和温泉, 2005 年 8 月.
 - (10) Muga Nishizawa and Shigeru Chiba: Preplanned Dynamic Weaving, 第 8 回プログラミングおよびプログラミング言語ワークショップ, 滋賀県大津市, 2006 年 3 月.
 - (11) 柳澤 佳里: カーネル用アスペクト指向システム KLAS の評価, 第 9 回 プログラミングおよび応用のシステムに関するワークショップ(SPA 2006), 2005 年 3 月.
 - (12) 青木 康博: アスペクト指向を利用した永続オブジェクト・アクセスの高速化, 第 9 回 プログラミングおよび応用のシステムに関するワークショップ(SPA 2006), 2005 年 3 月.
 - (13) 竹内 秀行: 振る舞いによるコンポーネントの分離と合成, 第 9 回 プログラミングおよび応用のシステムに関するワークショップ(SPA 2006), 2005 年 3 月.
 - (14) Muga Nishizawa and Shigeru Chiba: Preplanned Dynamic Weaving, Fifth International Conference on Aspect-Oriented Software Development, the University of Bonn, March 2006.
 - (15) Yoshisato Yanagisawa, Shigeru Chiba: "Performance evaluation of a kernel-level aspect-oriented system", Fifth International Conference on Aspect-Oriented Software Development, the University of Bonn, March 2006.
 - (16) Natsuko Kumahara, Rei Ishikawa, Muga Nishizawa, Kenichi Kourai, Shigeru Chiba: An Aspect-Oriented System with Recovery Advice, Fifth International Conference on Aspect-Oriented Software Development, the University of Bonn, March 2006.
 - (17) 渡辺 憲一, 一林 宏憲, 五島 正裕, 坂井 修一: プロセッサ・シミュレータ「鬼斬」の設計, 先進的計算基盤システムシンポジウム 2007(SACIS2007), 於 学術総合センター講堂・会議室(東京), Vol. 2007,

No. 5, pp. 194-195, May, 2007.

- (18) 杉本 健, 入江 英嗣, 五島 正裕, 坂井 修一: Out-of-Order スーパスカラプロセッサの FPGA への実装, 先進的計算基盤システムシンポジウム 2007(SAC SIS2007), 於 学術総合センター講堂・会議室(東京), Vol.2007, No. 5, pp. 196-197, May, 2007.

(6) 特許出願

①国内出願 (3件)

発明の名称: 高負荷時にも一部の利用者に優先的にサービスを提供できる計算機システム

発明者: 千葉 滋、松沼 正浩

出願番号:2004-29839

出願日: 平成 16 年 2 月 5 日

出願人: 科学技術振興機構

発明の名称: TCP/IP ストリームに対するパケット・レベルでのフィルタリング機構

発明者: 河野 健二、品川 高廣、Md. Rahat Kabir

出願番号:2004-162129

出願日: 平成 16 年 5 月 31 日

出願人: 科学技術振興機構

発明の名称: メモリ装置およびメモリ読み出しエラー検出方法

発明者:入江 英嗣、五島 正裕、坂井 修一

出願番号: 2006-189029

出願日: 2006 年 7 月 10 日

出願人: 東京大学 55% (独)科学技術振興機構 45%

②外国出願 : 1件

発明の名称: 中継装置、パケットフィルタリング方法及びフィルタリングプログラム

発明者: 河野 健二、品川 高廣、Md. Rahat Kabir

出願番号:PCT/JP2005/009632

出願日: 平成 17 年 5 月 31 日

出願人: (独)科学技術振興機構

(5)受賞等

■ 受賞

- (1) 優秀若手論文賞, Niko Demus Barli, Daisuke Tashiro, Chitaka Iwama, Shuichi Sakai, and Hidehiko Tanaka: A Register Communication Mechanism for Speculative Multithreading Chip Multiprocessors, 先進的計算基盤システムシンポジウム SACSIS2003, (May 2003).
- (2) 若手プレゼンテーション賞, Naoya Hatta, Niko Demus Barli, Chitaka Iwama, Luong Dinh Hung, Daisuke Tashiro, Shuichi Sakai, and Hidehiko Tanaka: Bus Serialization for Reducing Power Consumption, 情報処理学会報告書 計算機アーキテクチャ研究会 2004-ARC-159,情報処理学会 (Aug. 2004).
- (3) 第10回日本ソフトウェア科学会論文賞, 光来 健一・廣津 登志夫・佐藤 孝治・明石 修・福田 健介・菅原 俊治・千葉 滋, 日本ソフトウェア科学会, 2006.
- (4) 平成 17 年度長尾真記念特別賞, 千葉 滋, 情報処理学会, 2006.
- (5) Cell スピードチャレンジ 2007 規定課題部門 2 位, 先進的計算基盤システムシンポジウム SACSIS2007 併設企画 マルチコアプログラミングコンテスト「Cell スピードチャレンジ 2007」, Luong Dinh Hung: Implementation of a Parallel Radix Sort on Cell Processor, 於 学術総合センター講堂・会議室(東京),May, 2007.
- (5) 平成 19 年度猪瀬学術奨励賞, 財団法人電気・電子情報学術振興財団, 廣瀬 健一郎, 2007.

■新聞報道

■その他

(6)その他特記事項

7 研究期間中の主な活動

ワークショップ・シンポジウム等

年月日	名称	場所	参加人数	概要
H.15/7/30	Prof . Gurindar S. Sohi 講演会	東京大学工学部 3 号館 2階 31号講義室	120名	マイクロプロセッサの動向についての講義
15/8/6, 8/7	CREST ディペンダブル情報処理基盤拡大ミーティング	松江テレサ	21名	研究チーム内での各グループの成果発表
15/10/28	中澤喜三郎先生講演会	東京大学工学部 3 号館 2階 31号講義室	70名	Computer と共に歩んだ 49 年間

15/11/12	天野英晴先生講演会	東京大学工学部3号館1階36号講義室	50名	ダイナミックリコンフィギュラブルシステムの研究開発動向
16/1/20	内田啓一郎先生講演会	東京大学工学部3号館2階37号講義室	70名	スーパーコンピュータの原理と歴史
16/2/24	大域ディペンダブル情報基盤シンポジウム(協賛)	武田先端知ビル	100名	COE「情報科学技術戦略コア」主催 米澤明憲教授講演、坂井講演、平木講演、学生ポスター、
16/3/9	ディペンダブル情報処理基盤研究会	東京大学工学部3号館1階会議室	28名	研究チーム内での各グループの成果発表
16/8/5	吉瀬謙二先生講演会	東京大学工学10号館 4階大会議室(400号室)	30名	ディペンダブルアーキテクチャ
16/9/27	五島正裕先生講演会	東京大学工学3号館 3階31号教室	50名	Out-of-Order ILP プロセッサにおける命令スケジューリングの高速化
16/10/19	ディペンダブル情報処理基盤研究会	東京大学工学部3号館1階会議室	24名	研究チーム内での各グループの成果発表
16/12/21	荒木拓也氏講演会	東京大学工学3号館 2階37号講義室	30名	グリッドコンピューティングの最近の動向
17/3/8-9	大域知能、ディペンダブル情報処理関連の公開シンポジウム	武田先端知ビル	100名	COE「情報科学技術戦略コア」主催 坂井講演、平木講演、入江ポスター
H17/5/11	ディペンダブル情報処理基盤研究会(CREST 拡大ミーティング)	東京大学工学部3号館1階会議室	23名	研究チーム内での各グループの成果発表
H17/7/25	吉瀬謙二先生・近藤正章先生講演会	東京大学工学10号館4階大会議室(400号室)	30名	国際会議 The 32nd Annual International Symposium on Computer Architecture

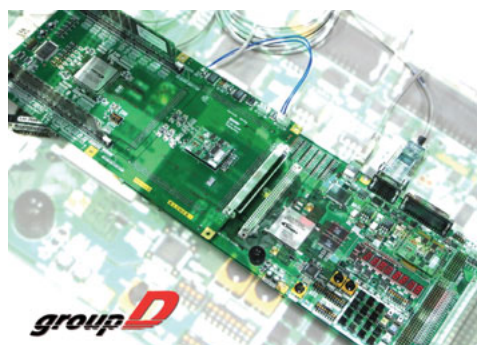
				(ISCA-32)の講演、チュートリアル、ワークショップの紹介&議論
H17/10/7	三輪忍先生講演会	東京大学工学部10号館3階 340C号室	30名	プロセッサ・シミュレータのためのキャッシュのクラス・テンプレートの設計
H17/12/20	中村友洋先生講演会	東京大学工学部3号館34号室	30名	Recovery Oriented Computing
H18/4/14	南谷崇先生講演会	武田ホール (本郷キャンパス浅野地区武田先端知ビル 5階)	80名	ディペンダブルコンピューティングの過去・現在・未来
H18/5/10	ディペンダブル情報処理基盤研究会 (CREST 拡大ミーティング)	東京大学工学部2号館N棟11階 電気系会議室5	29名	CREST「ディペンダブル情報基盤」メンバーによる発表及び討論
H18/6/6	吉瀬謙二先生講演会	東京大学2号館 N棟 4階 241号講義室	30名	タイルアーキテクチャという新しい流れ
H18/6/20	井上弘士先生講演会	東京大学2号館 N棟 4階 241号講義室	70名	マイクロプロセッサの信頼性と安全性を向上する!
H18/7/4	佐藤寿倫先生講演会	東京大学2号館 N棟 4階 241号講義室	70名	省電力とディペンダブルプロセッサ
H18/10/12	ディペンダブル情報処理基盤研究会 (CREST 拡大ミーティング)	東京大学工学部新2号館3階電気系会議室2	22名	CREST「ディペンダブル情報基盤」メンバーによる発表、討論
H18/12/12	ディペンダブル情報処理基盤研究会メンバー運営会議	東京大学工学部新2号館3階電気系会議室2	8名	CREST「ディペンダブル情報基盤」メンバーによる研究打ち合わせ
H19/5/30	ディペンダブル情報処理基盤研究会 (CREST 拡大ミーティング)	東京大学新2号館 N棟 12階電気系会議室5	20名	CREST「ディペンダブル情報基盤」メンバーによる発表、討論

H19/8/21	Derek Chiou 先生講演会	東京大学新2号館 N棟 11 階電気系会議室4	30 名	Fast, Full-System, Cycle-Accurate Computer Simulators
----------	-------------------	-------------------------	------	---

8 結び

超ディペンダブルアーキテクチャの研究では、“アーキテクチャによるディペンダビリティ”という、新しいチャレンジングな分野を切り開くことができ、大変意義があったと考えている。対外発表を通して、プロジェクト前に比べて、ディペンダビリティとアーキテクチャの関連も認知されるようになった。

前例のない技術も多く、困難な研究を進めていく状況も多かったが、チーム相互の意見交換や十分な設備・備品が大きな助けとなった。図 31、32 に、ここで開発したテストベッドと、開発に携わったメンバーの写真を記す。



(a) テストベッド第一版



(b) テストベッド第二版

図 31 超ディペンダブルプロセッサ・テストベッド



図 32 超ディペンダブルアーキテクチャ研究従事者

耐永久故障プロセッサについては、プロセッサの実行ステージに対し実行速度・チップ面積・耐故障性の3つを兼ね備えた耐故障化手法を確立するという当初の目標を達成することに成功した。本研究で得られた成果によりプロセッサの実行ステージに対する耐故障化を効率よく行う事がはじめて可能になった点で意義があると考えている。今後は、耐故障演算器を実装する場合に演算性能と耐故障性を高めるためにはどのような再構成アーキテクチャが適しているか研究を行う予定である。

侵入検知ハードウェアについては、TCP ストリームレベルでのトラフィック検査をワイヤレートで行うという当初の目標を達成する事に成功した。また、当初の目標に加えてDoS アタックへの応用までを検討することができた。本研究で得られた成果によりバックボーンネットワークにおける TCP ストリームレベルでのトラフィック検査がはじめて実用可能になったと言える。また、本研究で提案した要素技術はより高度なトラフィック検査手法などにも応用可能であると考えている。今後はバックボーンネットワークでの侵入検知システムをより高度化するための研究を展開する。一つ目の方向性はアプリケーションレイヤの規則に基づいたマッチングを行うことによる高精度化、もう一つの方向性はDoS アタックを緩和する手法への応用である。

サーバ用基盤ソフトウェアについては、5年間という限られた時間の中で、満足のいく結果を得ることができたと考えている。研究期間の初期段階から始めた、Apache の性能パラメータ調整機構、TCP ストリームフィルタなどは、研究として完成度の高いものになったと自負している。そうした研究成果を踏まえて、研究期間の後期からスタートした分散ホスティングなどのテーマは、次につながる形で発展させていくことができた。まとめるべき研究はまとめつつ、次につながるテーマを開拓していくことができたと感じており、

アプリケーション用基盤ソフトウェアに関しては、アスペクト指向の応用によって、コードを埋め込むことで飛躍的にディペンダビリティを高める技術を研究開発することに成功した。河川水位監視や SNS など、実用的な応用デモを行うことができ、実社会への展開についても十分に示せたと考える。Javassist などソフトウェアも公開され、多くのユーザを獲得した。

また、技術統合の試みとして、高速侵入検知システムとTCPストリームフィルタ、サーバ用基盤ソフトウェアとアプリケーション用基盤ソフトウェアの協調を検討し、デモを行ってその有効性を示した。今後は、提案の超ディペンダブルプロセッサ上でのサーバ用基盤ソフトウェアとアプリケーション用基盤ソフトウェアを検討・試作・評価するなど、より大きな統合が課題である。また、半導体技術やヒューマンインタフェース技術との協力も近未来の重要なテーマである。

本プロジェクトのように、広範で深い研究開発を行うためには、予算規模の大きな安定した枠組みが必須であり、CREST は最適な形態と経験的に理解した。今後とも、CREST の枠組みによって少数精鋭チームによる基盤研究を推進し、本邦発の世界に冠たる研究成果を出し続けることは、われわれ基礎研究に携わる者にとってなにより大切なことと思われる。