情報爆発に対応する高度にスケーラブルなモニタリングアーキテクチャ

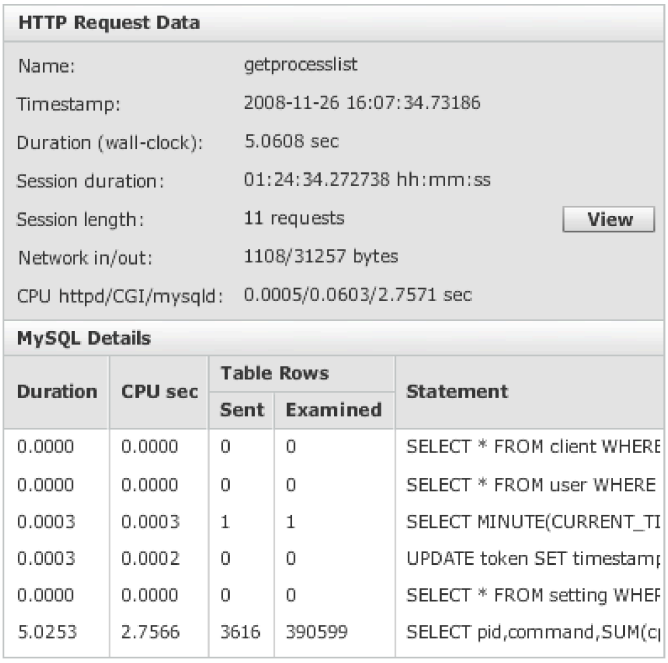
研究代表者	中島　達夫	早稲田大学・理工学術院・教授
研究連携者	村岡　洋一	早稲田大学・理工学術院・教授
	後藤　滋樹	早稲田大学・理工学術院・教授
	山名　早人	早稲田大学・理工学術院・教授
	甲藤　二郎	早稲田大学・理工学術院・教授
	追川　修一	筑波大学・システム情報工学研究科・準教授
	秋岡　明香	電気通信大学大学院・システム情報学研究科・助教

１．研究概要

大規模な分散システムを安定して動作させるためにはシステムが置かれた状況を理解することを可能とする必要がある．本研究では，そのためのインフラストラクチャとして様々な実時間に生成された情報を収集，分析することを可能とするためのモニタリングアーキテクチャに関する研究をおこなう．

２．モニタリングアーキテクチャ

本年度は２つのモニタリングシステムに関する研究をおこなった．１つ目のシステムは mBrace である．mBrace は Web サービスの性能解析ツールであり，トランザクション毎の詳細な性能評価を可能とする．図１に示すように，各トランザクションの性能結果を可視化することが可能であり,それにより管理者はどのトランザクションが問題であるかを容易に評価することが可能である．mBrace は Web アプリケーションを変更せずに，ミドルウエアの変更のみで，詳細な性能評価を可能とする．本年度は，mBrace の有効性を大規模なベンチマークを利用することにより示した．ベンチマークは簡単な Web ベースのショッピングサイトをエミュレーションするプログラムであり，mBrace は大きなオーバヘッドを生じずに詳細な性能データを集め，容易にボトルネックを検出出来ることを示した．また，mBrace が生成する情報を利用して，各トランザクションをマルチコアプロセッサ上でスケジューリングする方式に関して検討をおこなった．過去のモニタリング情報を利用して，各トランザクションを最適なコアに割り当てる方式に関する検討をおこなった．また，実際に提案する方式を利用して，Web サービスの性能を改善出来ることを示した．



HTTP Request Data

Name:	getprocesslist
Timestamp:	2008-11-26 16:07:34.73186
Duration (wall-clock):	5.0608 sec
Session duration:	01:24:34.272738 hh:mm:ss
Session length:	11 requests　View
Network in/out:	1108/31257 bytes
CPU httpd/CGI/mysqld:	0.0005/0.0603/2.7571 sec

MySQL Details

Duration	CPU sec	Table Rows		Statement
		Sent	Examined	
0.0000	0.0000	0	0	SELECT * FROM client WHERE
0.0000	0.0000	0	0	SELECT * FROM user WHERE
0.0003	0.0003	1	1	SELECT MINUTE(CURRENT_TI
0.0003	0.0002	0	0	UPDATE token SET timestamp
0.0000	0.0000	0	0	SELECT * FROM setting WHEF
5.0253	2.7566	3616	390599	SELECT pid,command,SUM(c

図１　mBrace 性能可視化ツール

現在の組込みシステムは単体として用いられるのではなく，ネットワークに接続され，大規模システムの一部として機能することが多い.ハードウェアの高性能化やアプリケーションへの機能要求の増大により，ソフトウェアのコード量は増大し，これらの要因によりシステム障害はより複雑化している．形式検証手法を利用した場合でも，数百万行ある全てのコードに対して検証をおこなうことは困難である．また，十分なテストをおこなったとしても，障害を発生する恐れやセキュリティホールなどが残る可能性が高い．こうした現状に対し，本研究では，実時間でアプリケーションの振る舞い

の異常を検出するロギングサービスを提案した.ロギングサービスは出荷時にテストや検証により予知できなかった異常を運用時に検出し，即座に上管理者に通知するシステムである(図２).

本システムは，稼働時にシステム内で発生する様々な異常を検出するためのシステムとしてDEOSに貢献する.例えば，異常を検出してハードウエア仮想化層状に複数のOSを動作させることにより，監視用OSを介して異常を発生したOSを再起動することが可能となる．以下に，本提案で構築したロギングサービスの特徴を述べる.

(1) 低オーバーヘッド

組込みシステムではコスト制約もありハードウェア資源が十分でないケースが多い．本研究では,運用時ロギングと解析のオーバーヘッドが 1.2%で異常検出の付加価値が提案できることを示した．低オーバーヘッドの実現のために，本研究ではシステムコールなどのイベントログではなく，プロセス毎の資源利用率のログを用いた．プロセスが呼び出すシステムコールのトレースを使用する場合は，プロセスが呼び出したシステムコールの全ての実行ログが必要であるが，資源利用率を用いる場合は，周期毎にプロセスの資源利用量を統計値として取得するコストのみがオーバヘッドとなるので，ログの取得・解析にかかるコストが小さい．また，検出精度を上げるために，プロセス毎の資源利用率を用いた.

(2) ブラックボックスアプローチ

組込みシステムでは,様々な製品やサービスを考慮しないといけないので,使用するプログラミング言語も一つではなく，様々な言語が使用されることを想定する必要がある．本提案では，特定の言語に依存しない資源情報を入力情報として用いた．資源使用率を考慮して異常を検出する際，CPUのみの資源では，例えば攻撃時に生じるメモリ量の変化やネットワークの使用率の変化が重要な場合にプロセスの振る舞いを解析することが困難となる.プロセスの動作を詳細にモデル化するためには，3 つの資源(CPU，メモリ，ネットワーク)を対象とした分析が必要であると考えた．また，資源利用率は確率的な状態遷移によりモデル化できると考え，HMM（Hidden Markov Model）を用いた．これらにより，アプリケーションには一切手を入れずに精度の高いモデルを作成することができた.

(3) オンラインでの検出

組込みシステムは，B2Bのサーバマシンと異なりコンシューマーが利用する機器がターゲットである．通常マシンの状況を熟知した管理者が存在し，異常発生時の問題解析や通知をおこなうことを前提とすることができない．そのため，組込み機器がネットワークに接続されている場合は,ネットワークを介して異常をオペレータが監視しているサーバ側のノードに通知するものとした．また，被害が拡大しないよう，異常が検知されたプロセスの資源利用量を制限することによりシステムの負荷が予測不可能に増大しないように制御することを可能とした.

(4) 未知の異常の検出

実際の製品の開発では，出荷前に想定できる範囲内でのテストをおこなうため，既知の問題は出荷前に対処する．そのため，運用時に発生するのは，想定外の攻撃やタイミングに関わる問題などである．こうした想定外の問題を効率よく検出するためにロギングサービスでは，正常状態を学習し，稼働時は，正常状態と異なる状態を検出した場合に，異常として報告する．これにより，運用時に未知の異常を検出することを可能とした.

以下に検証結果を示す.ロギング時のシステム全体にかかるオーバーヘッドの計測では，ロギングシステムを動作しない状態，学習中，運用中時に,動作させたプログラムの実行に必要とした時間を計測し比較をおこなったところ,学習時のオーバーヘッドは1.1%，運用時のオーバーヘッドは1.2%と，非常に低い結果が得られた.

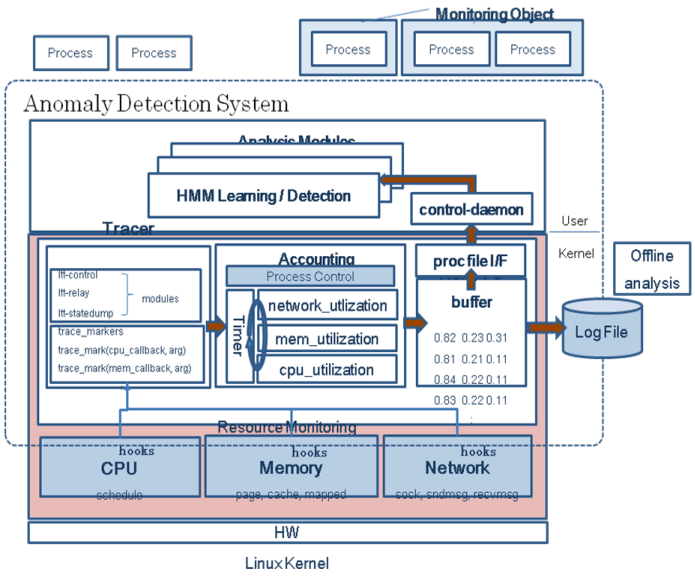


図２ アーキテクチャ全体

また，異常検出精度に関する検証では，サーバクライアントシステムを構築し，クライアントからサーバに対して正常なアクセスと攻撃が含まれたアクセスをおこない，正常に攻撃が検出されることを示した（図３，図４）．

比較対象としては，SQL インジェクションとバッファオーバーランの検出について，本手法（HMM）による方式，閾値，移動平均，増加率を用いた各手法を利用した場合の異常検出精度を比較した．結果より， HMM を利用した場合の検出精度は，誤認識はあるものの，検出精度が高いことがわかった．

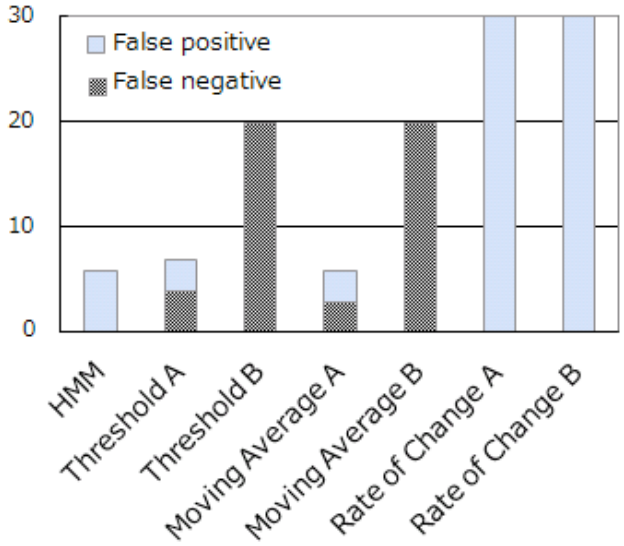


図３ SQLインジェクションの検出

図４ バッファオーバーランの検出

以下，今後の検討課題に関して述べる．

１） 検出精度の向上

今回の評価実験を進める中で，検出精度に影響を与えるパラメータには，検出時の学習モデルの判定スレッシュホルド，検出周期，検出プログラムと検出対象アプリケーションとの優先度などがあることがわかった．今後，実験を続け，これらの項目や相関関係を明確にし，より精度の高いモデル化やパラメータの最適化により検出精度を向上させる．

２） 検出する異常の明確化

現状のロギングサービスでは，異常の例としてセキュリティの問題を題材にした評価をおこなった．しかし，提案するロギングサービスはセキュリティ面で生じる異常だけではなく，ソフトウエアのバグ等により生じる様々な異常の検出にも利用可能であると思われる．中間評価以降も調査を続け，本システムにより検出可能な異常がどのようなものであるかを明確にしていく．

３） 予測機能の検討

現在のロギングサービスはシステムの正常動作モデルを HMM の学習により作成し，それを利用した異常を検出している．この手法は異常モデルを全てリストアップする必要がないので，軽量で即効性がある．しかし，異常を事前に予測することはできないという制限がある．中間評価以降は，予測を可能にするため異常動作モデルの学習をおこない，その情報を利用して，事前にその兆候を予測し，障害の発生を未然に防ぐ方法を検討する．

また，グループのメンバーを中心に，First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009) の開催をおこない，情報爆発セッションを設置することにより現状の研究成果の宣伝をおこなった．ワークショップには内外から１００名程度の参加者が集まり，海外からの参加者が６０％程度であった．ワークショップでは，個別技術の他に，グループの研究の概要を紹介する以下の論文の発表をおこなった．プロシーディングは IEEE により出版した．

３．マルウェア検出技術

マルウェアは、一見無害なプログラムを装って侵入するプログラムである。今年度はマルウェア対策として、

・ハニーポットを使って収集したボットについてその振る舞いの解析を行った

・ マルウェアの解析技術の開発を行った．

以下にそれぞれの成果について報告する。

（１） ボットの振る舞いの解析

マルウェアによる脅威は、 高度に複雑化され、脅威となるマルウェアやその配布元サイト、 攻撃者の存在が隠蔽されている。このようなマルウェアの感染は、 Webアプリケーションとクライアントアプリケーションの脆弱性が合わせて悪用されていることが多い。

マルウェアの中でも、 ボットネットへの接続による被害が拡大している。 ボットネットとは、 ボット

などのウィルスによって外部の人間によりコントロールされるようになった複数のコンピュータをつなぐネットワークのことを指す。 ボットは従来のワームやウィルスのように自動的に感染を拡大せず、 Harderと呼ばれる攻撃者からの指令を受けて活動するため、 その実態の把握が難しいといわれている。
本年度は、 ハニーポットを利用したシステムを構築し、 得られた通信トラフィックのキャプチャ（研究データセット CCC DATASet 2009）に対して時系列分析を行った。 時系列に対し、 挙動を可視化することにより、 ボットネットの実態や挙動を観測し、 時間と攻撃元・攻撃先 IP アドレスの関連性を評価できる。 得られた結果を評価することで、 ボットの実態、 動向の観測を行った。

得られた結果のいくつかを紹介する。攻撃通信データの中でも TCP-syn と UDP のパケットに注目した。TCP-syn のパケットを攻撃パケットと見なすことで、 さまざまなパターンに分けて挙動を抽出することができた。 ボットからの通信のプロット図からボットから外部端末への継続的な感染活動が把握できた。 ボット感染端末への通信のプロット図では様々な国からアクセスが網羅的にきていることを見ることができた。また、 UDP のパケットに注目し、 解析した結果、ボット端末の周りに攻撃したり、 特定の IP を継続的に攻撃していることが分かった。

以上より、 TCP-syn と UDP のパケットによる解析結果を比較すると、 TCP-syn パケットはアドレススキャン型の攻撃を観測し、UDP パケットは特定のアドレスに対する攻撃の傾向が多く観測することができた。

（2）マルウェア解析技術

マルウェアは単純なオブジェクト形式ではなく、パラメータによる自己展開などを行うため、これまでのパターンマッチングなどの技術のみでは検出は不可能になっている。これに対応するために、プログラムの命令系列をトレースして不審な振る舞いをする可能性の有無を検出する技術として、分岐命令等に頼らず確率的に命令かデータかを判断する逆アセンブル手法を提案した。これまでの方法に比べて高い正解率が得られた。

昨今の多くのマルウェアは、一般的なソフトウェアと同様、バグ改修や機能改良といった工程を円滑に実施するために、C やC++といった高級言語で開発されている。こうして、効率的なマルウェアの亜種の開発が行われている。加えて、マルウェアはウィルス対策ソフトによる検出から逃れるために、パッカーと呼ばれる一種の圧縮ツールにより、同じ機能を持ちながらも外観が異なる形式で大量に生産されている。このような一連の開発サイクルにより、近年のマルウェア数は増加の一途を辿り、それら全てに対策を打つことはおろか、対策の優先度を決めることさえ困難になっている。

こうした背景を踏まえ、マルウェアが備える脅威を解明する方法として機械語命令列間の類似度算出手法を提案した。これは2 つの機械語命令列の最長一致部分列を算出することで、それらの長さから類似度を算出する手法である。これにより新たなマルウェアが出現した際に、過去に収集されたマルウェアとの近さを算出できるとともに、過去のマルウェアと共通の命令列および実際に変更のあった箇所を推定することも可能になる。

次にこれまで提案してきたアンパッキング手法と逆アセンブル手法に、機械語命令列類似度算出に関する提案手法を組み合わせ、自動マルウェア分類システムを構築した。また実験として、3 グループのマルウェア検体を本システムにより分類した。その結果から、約3000 検体のSHA1 ハッシュ値が異なるマルウェアに

関して、約50%がConficker、約25%がW32。Rahack。H およびW32。Rahack。Wであることが分かった。つまりこのデータセットに関しては、わずか数種類のマルウェアを解析することで、全体の75%程度のマルウェアの機能を把握できることになる。

4.自己組織型セキュリティミドルウェア

自己組織型セキュリティミドルウェアに関係する研究課題として、P2P型データ配信技術に関する検討を進めており、まずメッシュ型配信とツリー型配信を組み合わせた配信方式の新方式の提案を行ない、配信効率の改善と障害時の復旧時間の改善を実現した。また、理論的な観点から、P2P型データ配信におけるオーバーレイ構造とノード接続方式に関する検討を行い、配信遅延と総トラヒックの低減を同時に満足するノード配置アルゴリズムを明らかにした。また、センサーネットワークを絡めた大規模画像データベースの活用を想定して、より効率的なオブジェクトの同定・認識方式に関する検討も行った。

またネットワーク測定技術の面から、ルータの内部情報を可視化する技術の検討を進めた。インターネットのルータはパケットを単に素通しにするのではなく、種々の情報を保持しながら稼働している。この情報をエンド(end)ノードにおいて活用することができれば、ネットワークを効率的に使うことができる。この研究は産業技術総合研究所との共同研究として遂行した。早稲田大学側は可視化を可能としたルータ(i-Pathと名付けた)を使用することにより、通信経路にそってセキュリティの情報を提供することができることを示した。また昨年度から研究を継続しているNAT越え(traversal)の技術においてi-Pathルータを活用すればアルゴリズムが容易になることを示した。

以下に具体的に述べる。

メッシュ型配信とツリー型配信を組み合わせたP2P型データ配信方式においては、基本的に、高性能サーバ群を上位層でメッシュ構造に従って接続し、性能の劣るサーバ群は下位層でツリー構造に従って接続する方式の提案を行った。既存方式として、通常はツリー構造で配信木を決定し、障害発生時のみメッシュ構造をアクティブにする方式や、本提案とは逆に、上位層はツリー構造で構成し、下位のクライアント群はメッシュ構造で構成する方式が知られているが、提案方式によって、配信効率の改善、障害復旧時間の低減、ならびに、制御オーバーヘッドの削減が実現できることを明らかにした。さらに、具体的な応用例として、地理的に分散したデータセンター網を想定し、提案方式が障害耐性を提供することも検討を行った。

P2P型データ配信の理論解析に関しては、これまでCDN(コンテンツ配信ネットワーク)に関して進めてきたコンテンツの人気度に応じた複製配置方式兼負荷分散方式を応用し、人気度をサーバ性能に置換え、理論的な観点から、サーバ性能に応じたP2P型データ配信網の構成方式を明らかにした。これによって、従来の方式に対して、データの配信遅延と総トラヒック量を共に削減するノード配置アルゴリズムを明らかにし、かつ、上記で行ったメッシュ型配信とツリー型配信のハイブリッド方式の有効性に関する理論面からのサポートを行った。

大規模画像データベースを活用したオブジェクト認識方式に関しては、カメラを配したセンサーネットワークを想定し、撮像画像に対して、同定・認識に不要と思われる領域を排除することで、オブジェクトの認識性能を改善する方式の提案を行った。具体的には、シームと呼ばれる変化の少ない地帯を逐次削減する方式と、人間の目に顕著に知覚される領域を積極的に残す方式を組み合わせ、結果として、認識に有効と思わ

れる領域のみを残す処理を実現した。この方式のネットワーク応用を考えた場合、有効領域に限定することによる転送情報量の削減と、物体認識の性能改善の効果が同時に期待されることになる。

ネットワークの測定技術に関しては、ルータの内部情報を可視化する技術と、その応用に重点を置いて研究を進めた。ルータはインタフェース毎の特性、利用帯域、個々のパケットだけではなくフローとしての特性などの情報を保持して、内部のキュー(queue)を制御している。この情報は通常は管理者が見るだけであり、利用者はルータの内部の情報を見ることができない。我々はルータの内部情報の「見える化」をはかった新技術(i-Path)の応用を探求するべく産業技術総合研究所との共同研究を行った。

これまでに産業技術総合研究所において、通信の経路に沿ってリンクの帯域情報を収集すること、実際の利用帯域を収集すること、さらにルータの地理的情報を収集してルータが設置されている場所を地図上に表示する応用が実現されていた。早稲田大学のチームは、これまでのネットワーク計測に基づくセキュリティの研究の実績を活用して、各ルータにおけるフローの特性をi-Pathの情報として収集すれば、通信の経路に沿って有用な情報が提示できることを示した。その一例はDoS (サービス妨害攻撃)の検知法の提案である。

さらに昨年度に実現したNAT (Network Address Transformation)を越える技術にi-Pathの技術を応用する検討を進めた。従来のNAT越えの技術は単純なNAT、例えばcone NATには有効である。我々の先行研究では、symmetric NATのような複雑なNATの構成に対しても有効なtraversalの技術を提案していた。その技法においてはNATの構成を推定するために多数のパケットを観測する必要がある。i-Pathの技術を用いれば推測が不要となり、NAT越えのアルゴリズムが簡素となり、成功の確率が向上する。NATの構成法には数種類があるが、i-Pathの情報として提示することは容易である。

上記のi-Patの情報提供に関しては、必ずしもルータが保持する全部の情報を提示する必要はない。この点はi-Pathの仕組みの中で、情報開示法についての考慮をしている。

今後の展望は下記のとおりである。

・P2P型データ配信に関しては、データセンター応用等の検討に加え、悪意のあるノードが存在する場合のセキュリティ対策の検討やその理論検討を予定している。センサーネットワーク応用に関しては、上記P2P型データ配信との融合や、監視ネットワークへの応用を通じたセキュリティ拡張に関する検討を進めていく予定である。

ネットワーク測定に関しては、測定結果の対象をi-Pathの提示する内部情報に拡大できた。今後は、その情報を活用してP2Pネットワークのトラフィックを制御する研究を進めていきたい。また提案する手法を実証できるネットワーク環境の実現に留意をしていきたい。

5. 分散処理フレームワーク・ストレージボトルネック軽減の研究

本グループの目的は、分散処理の効率化を実現するために、ストレージやネットワークのボトルネックを軽減することである。近年、クラウドサービスの台頭により個人や中小企業でも大量の計算資源を利用し、情報爆発時代の大規模データに対応できるようになってきている。分散処理を補助するため Apache Hadoop のような分散フレームワークの開発が盛んであるが、MapReduce 型フレームワークは I/O ボトルネックが大きいことや、ストリーム処理型のアプリケーションが作成しにくいといった問題点が指摘されている。本グループでは、Key-Value ストレージを利用することで I/O ボトルネックを軽減した MapReduce 型分散フレー

ムワーク Gridool、ストリーム処理型のアプリケーションに適した分散フレームワーク QueueLinker の開発を進めている。また、将来のメニーコア CPU 環境で深刻になると予想される排他制御によるボトルネックを軽減するため、ロックフリーなキャッシュアルゴリズム Nb-GCLOCK を実現した。

分散フレームワークの開発

分散フレームワークとして Google MapReduce のオープンソース実装である Apache Hadoop が代表的であるが、Hadoop が普及すると共に、既存の MapReduce 型フレームワークの問題点が指摘されるようになってきた。そのひとつに Map と Reduce ごとにデータがストレージに書きこまれるため、I/O ボトルネックが大きいことがある。これは Map が終了するまで Reduce が、Reduce が終了するまで次の Map 処理が開始できないということであり、複数の MapReduce 処理をパイプライン的に実行してパフォーマンスを向上できないということである。これはストリーム的にデータを処理するアプリケーションにとって致命的なオーバーヘッドとなる。

本年度は、Key-Value ストレージを用いてボトルネックを軽減した MapReduce 型フレームワーク Gridool、ならびにストリーム型処理に適したフレームワークである QueueLinker の開発を進めた。MapReduce 型のフレームワークでは Reduce を実行する前に同一の Key ごとに Value をまとめ上げる Shuffle フェーズを実行する必要がある。Shuffle フェーズではソートのためにディスクに対する I/O を行うためストレージボトルネックが発生する。我々のグループで開発している Gridool は P2P の技術である Key-Value ストレージを用いることで、データの分散と同時に Key ごとのまとめ上げを実現するため、ボトルネックを軽減することが可能である。現在、最終的な開発を進めており、2010 年前半には初期リリースを行う予定である[2]。

我々が開発しているもう一つのフレームワークである QueueLinker は、ストリーム処理型のアプリケーションに適した分散フレームワークである。QueueLinker は Producer-Consumer 型のプログラミングモデルを採用し、モジュール間がキューで通信することで分散処理を実現する。プログラマは各モジュールを作成し、モジュール間のデータの流れを指定する。QueueLinker は指定されたデータの流れに基づいて、利用可能な計算機にモジュールを配置し自動実行する。どの計算機にどのモジュールを配置するかといった分散戦略はプログラマが自由に制御することが可能であり、分散アプリケーションのチューニング作業を容易に行うことができる。プログラマの負担をより軽減するために、モニタリングシステムで取得した負荷情報を活用して、自動スケジューリングを行う機構の開発を進めている。

キャッシュアルゴリズムのロックフリー化

前述のようにストレージボトルネックは分散処理においても重大な問題である。しかし、メニーコア CPU が普及するにつれ、ハードディスクや SSD のような物理的なストレージだけでなく、キャッシュアルゴリズムそのものがボトルネックとなり得る。OS は空きメモリ上にファイルキャッシュを確保しているが、一般的にキャッシュアルゴリズムは、どのデータを破棄するかを決定するために、ファイルにアクセスがある度にデータ構造を更新する必要がある。この際、データ構造に対して排他制御が必要になるため、データベースのように並列アクセスを行うアプリケーションにおいては、ロック獲得のための待ち時間が発生する。例えば、LRU を用いる場合、ファイルに対するアクセスがある度に LRU リストの排他ロックを取得し、リスト

を並び替える必要がある。このような排他的動作は今後 CPU のコアが増えるにしたがって致命的なボトルネックとなり得る。

そこで本研究では、ロックフリーのキャッシュアルゴリズムである Nb-GCLOCK を開発した。ロックフリーアルゴリズムでは CPU の atomic 命令を利用することで、スレッドの進行を止めることなく排他的にデータの更新を実現できる。既存のキャッシュアルゴリズムでは 16 プロセッサ以上の場合にスケーラビリティを示せないのに対し、Nb-GCLOCK は 64 プロセッサまでほぼ線形のスケーラビリティを示すことを確認できた。

最終年度への展望

最終年度ではこれまでの研究成果をまとめ、外部にソフトウェアを公開していくことで社会的な貢献を目指す。具体的には開発した分散フレームワークをオープンソースとして公開し、広く利用を促進することを目指している。Nb-GCLOCK は分散フレームワークのキャッシュアルゴリズムとして採用する。QueueLinker の開発においては、本グループで作成したモニタリング機構から得た情報を利用することで分散処理のスケジューリング性能を向上させることを目指している。

７．研究成果リスト

著書、論文、国際会議

1) Andrej van der Zee, Alexandre Courbot, Tatsuo Nakajima: mBrace: Action-Based Performance Monitoring of Multi-tier Web Applications. The 7th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp166-173, 2009.

2) Midori Sugaya, Yuki Ohno, and Tatsuo Nakajima, Lightweight Anomaly Detection System with HMM Resource Modeling, International Journal of Security and Its Applications, pp35-54, Vol. 3, No. 3, July, 2009

3) Sayaka Akioka, Junichi Ikeda, Takanori Ueda, Yuki Ohno, Midori Sugaya, Yu Hirate, Jiro Katto, Shigeki Goto, Yoichi Muraoka, Hayato Yamana, and Tatsuo Nakajima, ″A Scalable Monitoring System for Distributed Environments”, First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009)

4) Yuki Ohno, Midori Sugaya, Andrej van der Zee, and Tatsuo Nakajima, ″Anomaly Detection System using Resource Pattern Learning″, First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009)

5) Midori Sugaya, Yuki Ohno, Andrej van der Zee and Tatsuo Nakajima, ″A Lightweight Anomaly Detection System for Information Appliances″ , 12th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2009) Tokyo, Japan, March 17-20, 2009.

6) 池田、岩村、秋岡、村岡：通信トラフィックの時系列分析によるポット活動の可視化と特長検出、MWS2009

岩村、村岡： 機械語命令列の類似性に基づく自動マルウェア分類システム、情報処理学会論文誌（投稿中）
7) 船越裕介，松川達哉，吉野秀明，後藤滋樹，“通信ネットワークの保全度向上のための故障修理時間分布の特性分析”，電子情報通信学会論文誌 B，Vol.J92-B，No.7，pp.1153-1163，July，2009.
8) Makoto Iguchi and Shigeki Goto，“Privacy-conscious P2P data sharing scheme with bogus profile distribution”，Web Intelligence and Agent Systems: An International Journal，Vol.7，pp.209-222，2009.
9) S.Awiphan，Z.Su and J.Katto，“ToMo: A Two-layer Mesh/Tree Structure for Live Streaming in P2P Overlay Network”，IEEE CCNC 2010，Jan.2010.
10) Z.Su，S.Awiphan，K.Ogura and J.Katto，“Hybrid Application Layer Multicast with Hierarchically Distributed Nodes”，IEEE CCNC 2010，Jan.2010.
11) M.Sato and J.Katto，“Performance Improvement of Generic Object Recognition by using Seam Carving and Saliency Map”，IWAIT 2010，Jan.2010.
12) Dai Mochinaga，Katsushi Koyabashi，Shigeki Goto，Akihiro Shimoda and Ichiro Murase，“Collecting inside information to visualize network status”，APAN Network Research Workshop，pp.1-4，July，2009.
13) Trung-Tuan Luong，Bu-Sung Lee，Chai-Kiat Yeo，Ming-Shiunn Wong and Shigeki Goto，“Algorithms to Minimize Channel Interference in Multiple Channels Multiple Interfaces Environments”，IEEE 34th Conference on Local Computer Networks (LCN 2009)，pp.61-68，October，2009.
14) 木佐森幸太，下田晃弘，森達哉，後藤滋樹，“TCPフィンガープリントによる悪意のある通信の分析”，コンピュータセキュリティシンポジウム2009，pp.553-558，October，2009.
15) Makoto Yui，Jun Miyazaki，Shunsuke Uemura，Hayato Yamana，“Nb-GCLOCK: A Non-blocking Buffer Management Based on the Generalized CLOCK,” In *Proc. of ICDE*, Long Beach，California，Mar. 2010 (To Appear).
16) 油井誠，宮崎純，植村俊亮，加藤博一，山名早人，“ロックフリーGCLOCK ページ置換アルゴリズム,” 情報処理学会論文誌 データベース(TOD)，vol.2，no.4，pp.32-48，Dec. 2009.

招待ポスター

[1] 上田高徳，片瀬弘晶，森本浩介，打田 研二，山名早人，“QueueLinker: Distributed Producer/Consumer Queue Framework,” WebDB Forum（招待ポスター），Nov. 2009.

その他の成果

[2] Gridool: An Infrastructure of Parallel Job Execution on Grid，http://code.google.com/p/gridool/