「超高性能メールフィルターの研究開発と商品化」

1. はじめに

我が国においてはインターネットが爆発的な普及をみせており、インターネットのユーザは増加の一途を辿っている。ユーザのほとんどは電子メールを利用しており、安価なメッセージ伝達手段である電子メールはビジネス・個人向け問わず急速に拡大してきた。特に最近では携帯電話からの電子メール利用が急速に拡大している。

一方電子メールの普及に伴い、各種のセキュリティ問題が顕在化した。例えば、無作為に送られてくるさまざまな迷惑メール（スパムメール）は、受信者の社会活動の時間浪費や生産性非効率化の大きな要因になっている。個人の側では、利用者の望まない情報に対する課金の発生（携帯メールでは受信メール数に応じて課金される）、私的生活領域への侵害等が引き起こされている。ISP（internet service provider）やキャリア側では、数多くの宛先不明メール（迷惑メールの宛先は無作為に選ばれている）がメールサーバの処理を圧迫し、迷惑メールのために設備増設を行う状況が定常化した。またメール全体の配信遅延、迷惑メールを理由とした利用者の解約等も増加している。NTTドコモの調査によると、2001年10月の1日当たりの平均値として、NTTドコモのメールセンターに届いた約9.5億通の電子メールのうち、約8億通が実在しない宛先への電子メールであった。またコンテンツ･プロバイダ側では、電子メールによるマーケティングに対する信頼が損なわれ、正当なマーケティングが悪影響を被っている。(1)

迷惑メールが受信者、電気通信事業者等に生じさせている被害や問題を解決することが、インターネット関連ビジネスの健全な発達を図るためには必要不可欠である。われわれは本問題の解決に向け、既存のメールフィルタリング性能を大幅に向上させる研究開発に挑戦したので報告する。本研究開発の成果は、2003年12月の弊社プレスリリース、新聞各社の記事にて既に公表済みである。(2)

2. 研究開発の背景

増加の一途を辿る迷惑メールやウィルスに対して、設備コストを増加し続けることなく、かつ健全なメール利用を促進していくためには、メールサーバの前段にそれらのフィルタリングを行うシステムが必要である。しかも、全メールサーバにおけるフィルタリング処理を代替するためには、既存の代表的なシステムである「汎用コンピュータ（サーバ、ＰＣ）＋オペレーティングシステム＋ソフトウェア」の性能を遙かに上回る必要がある。そこでわれわれは、従来システムの性能面のボトルネックを分析し、負荷の重いソフトウェア処理のハードウェア化と新規アーキテクチャによる大幅な性能向上に取り組んだ。

2.1　通信処理に関する課題

現状のメール転送における機能分担を図 1 に示す。

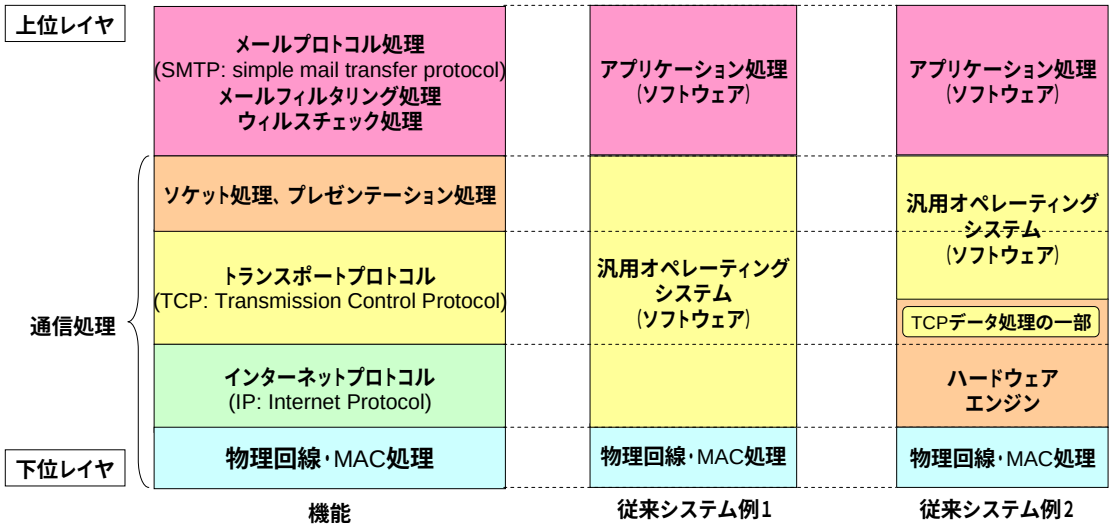

図 1 メール転送の機能分担

通信処理の性能向上に向けて、オペレーティングシステムの TCP 処理や IP 処理のハードウェア化が商品化されている。しかしながら部分的なハードウェア化であり、アプリケーションに近い領域は依然としてオペレーティングシステムに依存している。例えば従来の TCP ハードウェア化はデータ転送部分のみであり、プロトコル本体の制御やネットワーク輻輳時のデータ再構築はオペレーティングシステムに依存している。その結果として、多数の通信コネクションが持続する場合や、瞬間的に大量の通信コネクションが発生する場合は、オペレーティングシステムの性能がボトルネックとなって、全体の性能が上がらない。ちなみに既存システム単体の最大性能（新規通信コネクション速度）は数 100 コネクション/秒である。しかしながらモバイルキャリアにおける携帯メール量や、大手 ISP のメール量を想定すると、大量の通信コネクションが繁忙時に発生しており、少なくとも 1 桁～2 桁以上の性能向上が必須である。

2.2　コンテンツチェック処理に関する課題

メールフィルタリングは、メールプロトコル上のコマンド内容や本文、送信元の IP アドレス等のさまざまなコンテンツ情報を元に迷惑メールをフィルタリングする処理である。またウィルスチェックは、メールの添付ファイルにおいて数千ものシグネチャーと呼ばれる特有のパターンを検出する処理である。これらのコンテンツチェック処理は極めて負荷が高い処理であり、大型サーバにおいても数 10～100 メール/秒の性能が限

界である。汎用のパソコンの場合は、メール 1 通のウィルスチェックを行うだけでも人間が感じられるレベルの遅延が発生する。モバイルキャリアにおける携帯メールや大手 ISP のインターネットメールを想定すると、数桁上の大幅な性能向上が必須である。

2.3 新たなソフトウェア、ハードウェアの結合アーキテクチャに関する課題

ハードウェア処理の利点は高速性である。負荷の重いソフトウェア処理をハードウェア化するにあたり、ハードウェア処理に適した新規のアルゴリズムが考案できれば、数桁の性能向上が可能になる。一方ソフトウェア処理の利点は、機能追加の柔軟性や、機能記述の簡易性である。従って高機能高性能な処理システムを構築するためには、汎用的に利用可能なソフトウェア処理（前述した通信処理やコンテンツチェック処理）はハードウェアエンジン化し、アプリケーション単位の特有な処理はソフトウェアに分担させるシステムが有効である。しかしながら従来、前述した通信処理はオペレーティングシステムの存在無くして実現不可能であり、コンテンツチェック処理もアプリケーションレイヤのソフトウェア処理無くして実現不可能であった。

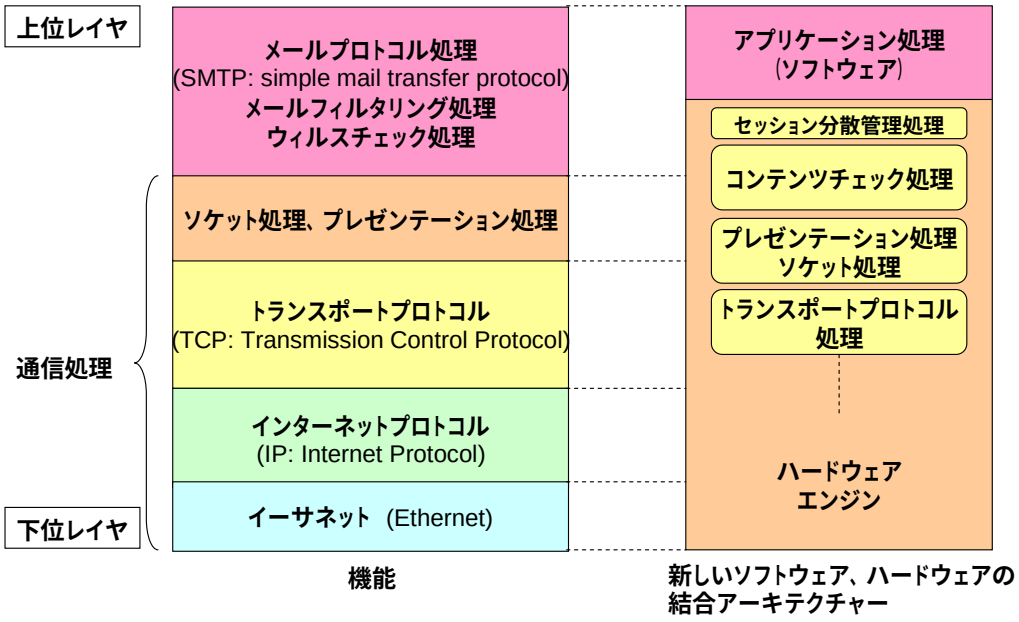

図 2 高速高性能が期待されるアーキテクチャー

4. 超高性能メールフィルターへの挑戦

本挑戦は、迷惑メールフィルター、ウィルススキャン等への特殊用途を前提とした、言わばハードウェアによるオペレーティングシステムの実現やハードウェアによる複雑なアプリケーションソフトウェア（コンテンツチェック）の実現であり、従来に無い研究

開発テーマである。それだけに実現の敷居も非常に高い技術であったが、われわれは従来比 1000 倍以上の性能を実現するシステムを実現した。適用例として、秒間 1000 通以上の高速スパムメールフィルタが可能になった（図 3）。

図 3　SLIMIT 1000 （Spam Mail Limitter ; 2003/12 プレスリリース）

次に前章で挙げた技術課題に関し、どのような技術で解決していったのかを紹介する。

5.　トランスポートコントロールプロトコル、ソケット、プレゼンテーション処理のハードウェアエンジン化

オペレーティングシステムにて実現されているトランスポートプロトコル処理、ソケット処理、プレゼンテーション処理のハードウェア化を行い、従来システム「汎用プロセッサ＋オペレーティングシステム＋アプリケーションソフトウェア」の 1000 倍に及ぶ性能向上を実現した。

そもそも逐次処理前提のソフトウェアと、処理タイミングが厳密に規定されまた並列処理を基本とするハードウェアの動作は異なる。従って、ハードウェアの高速性を生かすには、ソフトウェアプログラムの機能を十分解釈した上で、根本的に異なるハードウェア処理方式を開発する必要がある。TCP 処理に関して、われわれは実績のある BSD4.4 Lite のプログラムソース 数 10KL の解析を元に、すべてのプロトコル処理のハードウェア化を実施した。結果として、特に新規 TCP コネクション処理の能力は 160,000 セッション/sec を達成しており、従来システムの 1000 倍近い向上である。またソケット･プレゼンテーション処理に関しても、BSD4.4 Lite の機能を参考にしつつ、独自の処理方式により大幅な性能向上を達成した。以下にハードウェアエンジンの構成および性能を示す。

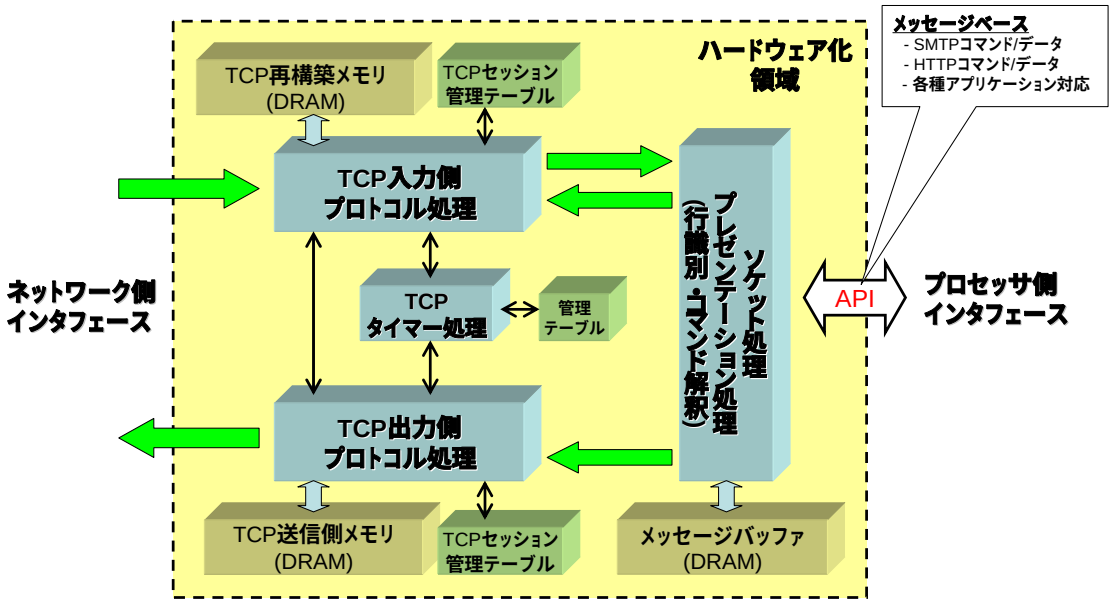

図 4　TCP、ソケット、プレゼンテーション処理のハードウェアエンジン構成

項目	内容
処理速度	GbE回線レート
新規TCPコネクション性能	160,000コネクション/秒
同時最大TCPコネクション数	メモリ(DRAM)増設により増加可能。
TCP基本機能	TCP Reno プロトコルフルセット実装
ソケット処理、プレゼンテーション処理	SMTP, HTTP, etcに対応。 10M文字/秒の処理性能。

図 5　性能諸元

本ハードウェアエンジンとアプリケーションを実施するプロセッサは、図 6 のように結合する。プロセッサは性能ボトルネックとなる通信系処理を行わず、付加価値の高いアプリケーション処理に専念することにより、高性能かつ高機能なシステムを構築することができる。

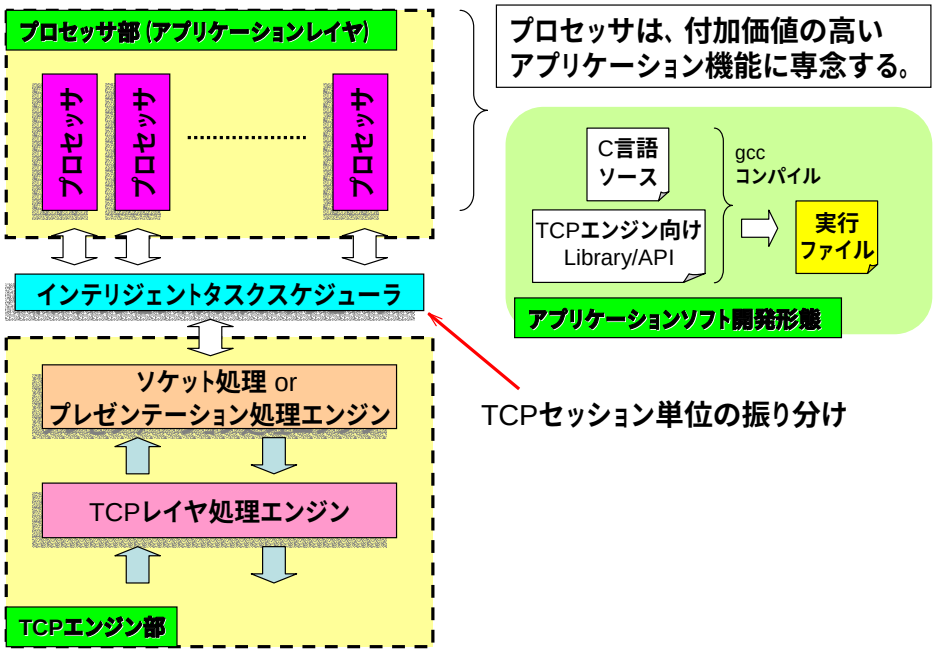

図 6 ハードウェアエンジンとアプリケーションプロセッサの結合

6. コンテンツチェック処理のハードウェアエンジン化（Programmable Pattern Search Engine）

迷惑メールは、メールの本文、Subject、メール経路、メールコマンド、送信元 or 宛先（IP レイヤ）の情報と、関連するデータベース（過去の履歴、ブラックリスト等）を元に判断される。われわれは、本処理の高速化を行うための共通機能を分析した。その結果として、2 種類の機能がハードウェアエンジンとして実現できれば、全体の処理を極めて高速にすることができるとの結論に至った。

まず、メールが届いた時、本メールに関する多様な情報を高速に収集する必要がある。それには、何らかのキーワードが必要である。そこで、キーワードの指定として複数の正規表現を指定し、左記正規表現に基づいて高速に該当メール内のキーワード情報を集めるハードウェアエンジン（Field Extract 部）を定義した。次に収集したキーワード情報に関して、データベース（過去の履歴、ブラックリスト等）から関連する情報を検索する必要がある。そこで取得済みのキーワード情報に関し、自動的にデータベースを高速検索するハードウェアエンジン（Table Search 部）を定義した。本エンジン間の関係を図7に示す。

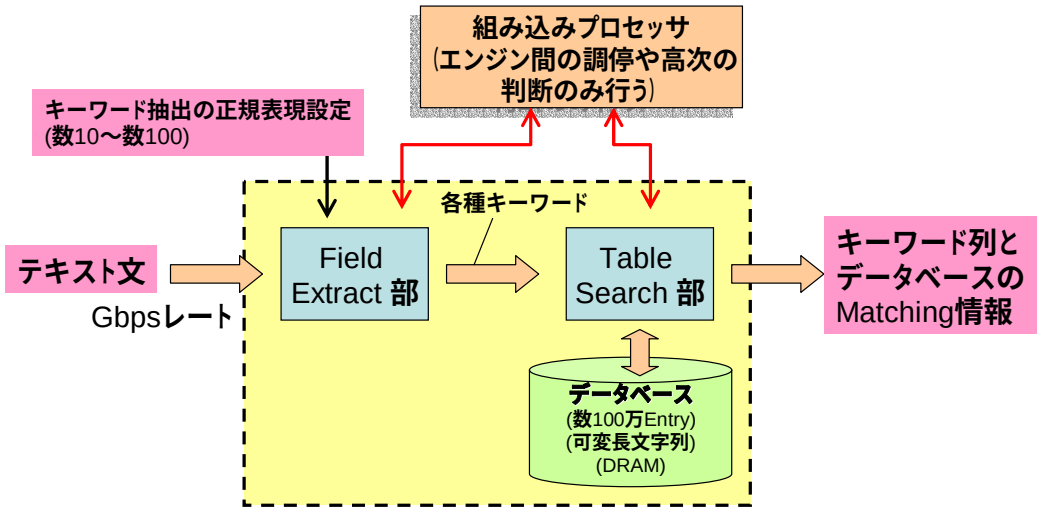

図 7　Programmable Pattern Search Engine

われわれは上記の高速化により、迷惑メールフィルタリングにおける「キーワード収集=>関連データベースの検索」に関し最大で 100 万件検索/秒の性能を実現した。これはメール当たり平均 10 個のキーワードが存在した場合でも 100,000 通/秒のメールを処理できる性能に該当し、既存ソリューションの 1000 倍以上の性能向上に該当する。また本方式は迷惑メール対策のみならず、広く利用されている WWW 通信におけるフィルタリング処理や通信コマンド内容のチェックにも有効である。

6.1　高速キーワード抽出ハードウェアエンジン (Field Extract 部)

Field Extract 部は、入力するテキスト文 (メール、ファイル、etc) 内のキーワードを検索する。柔軟なキーワード検索を可能にするために、キーワードの指定は正規表現である。また、個々の正規表現照合を行なう処理ブロックは内部スイッチ機構を利用して自由に接続可能なので、複数の正規表現を組み合わせたキーワード抽出が可能になる。例えば電子メールにおいて、ヘッダ文から各種キーワード (Subject, ID, 経路情報, 送信者のメールアドレス, IP アドレス) を取り出したり、また本文から各種キーワード (http リンク、電話情報、住所) を取り出したり、あるいは上記の組み合わせにすべて合致する条件でキーワードを抽出することが可能である。

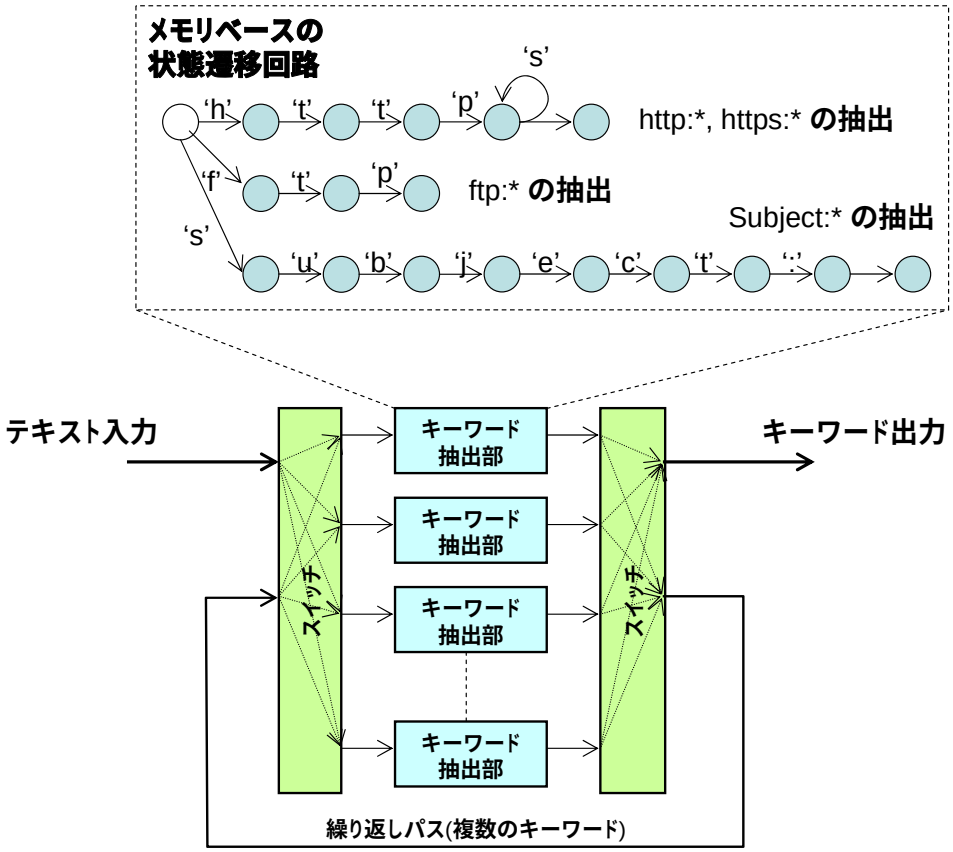

図 8　Field Extract 部

6.2　高速データベース検索ハードウェアエンジン (Table Search 部)

Table Search 部は、Field Extract 部で収集したキーワード情報を元に関連するデータベースを高速に検索する。検索においては、キーワードが最も長くマッチするエントリを選択する (Longest Prefix Match) 。例えば図 9 において、入力された検索キーが"ABCDEFGHIJ"で、かつデータベースに"ABCD"で始まるパターンが 3 つ登録されている場合、検索キーと最も長く一致する"ABCDEFGH"を検索結果として返す。Longest Prefix Match を用いる理由は、元のキーワード情報と最も長く一致するデータベース情報の方が関連性が高いと考えられるからである。Table Search 部では、データベースにおける参照番号として複数のハッシュキーを利用する。ハッシュとは、可変長の入力情報を固定長の短い参照番号にランダムに変換する技術である。ハッシュを用いることにより、様々な長さを有するキーワード情報を固定長の短い参照番号に変換できる。

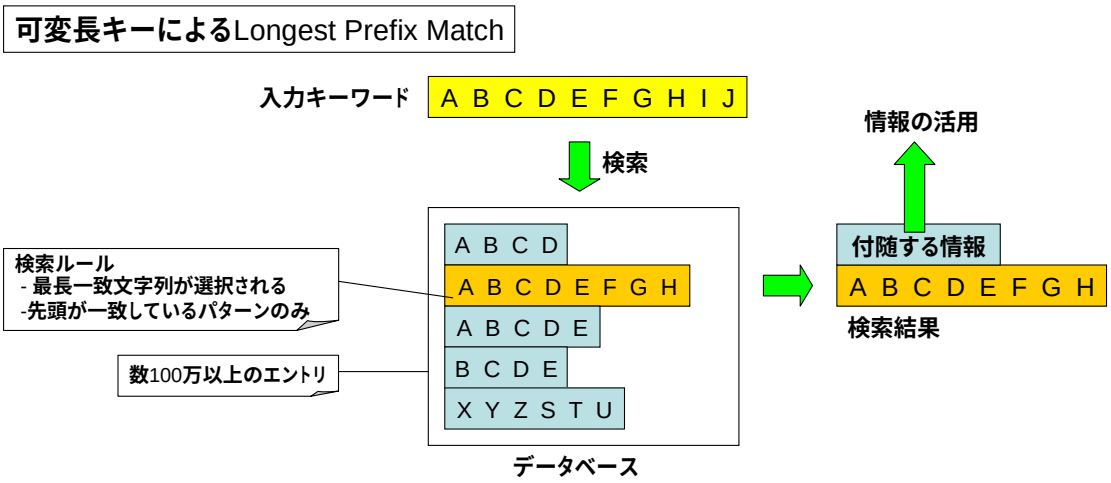

図 9　可変長キーによる Longest Prefix Match 検索

キーワード情報が最も長くマッチするエントリを検索するために、同じキーワード情報に対して、複数のハッシュキーを作成する。複数のハッシュキーは、入力キーワードの先頭からの長さ N byte、2N byte、3N byte・・・のストリング部分に対応して計算される。ハッシュキーを元にデータベース参照を行い、ハッシュ値が一致するエントリが見つかった場合は、メモリ（DRAM）に格納されている完全なパターンと照合を行う。複数の一致が存在する場合は、最も長いパターンを検索結果として採用することになる。

図 10 の例では、入力された検索キーから長さ N byte、2N byte、3N byte の 3 種類の先頭ストリングを取り出し、ハッシュを計算し、各々Hash_1/2/3 を得ている。この 3 種類のハッシュ値でハッシュ・テーブルを参照し、Hash_3＝Hash_A であったとすると、一致した Hash_A について DRAM から詳細なパターン情報を取得し一致検索を行う。同じハッシュを有する場合は、該当ハッシュに関してさらに複数回のリニア検索を行う必要があるが、十分大きなハッシュ関数を用意することにより、リニア検索の発生頻度を抑えている。

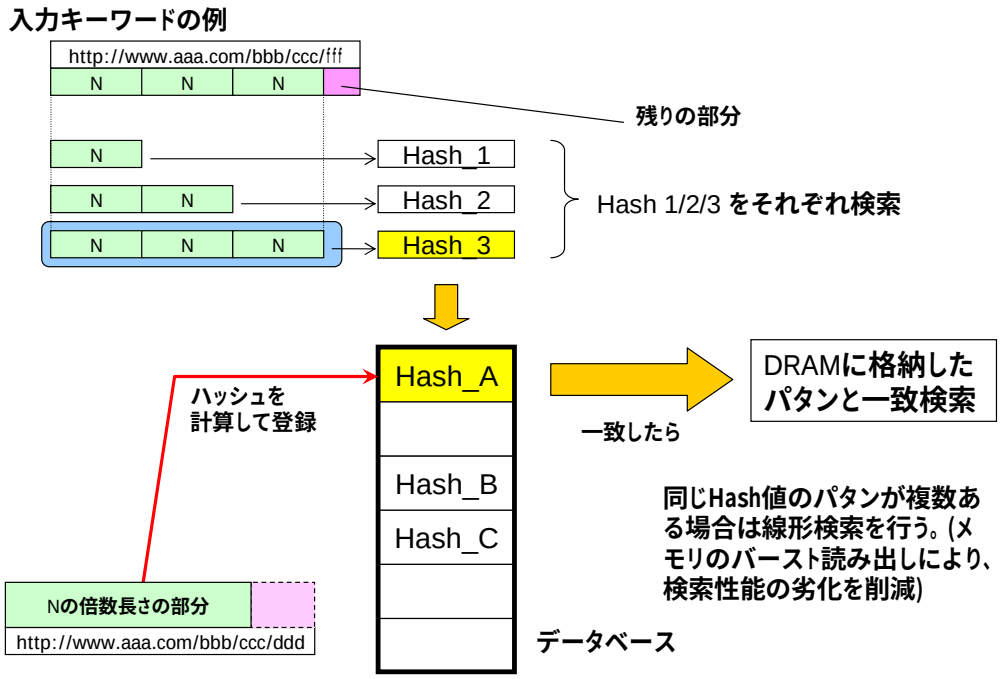

図 10　複数ハッシュキーによる Longest Prefix Match 検索

7.　コンテンツチェック処理のハードウェアエンジン化 (Stream Check Engine)

メールの添付ファイルに対するウィルススキャンは、シグネチャーと呼ばれるパターンの検索処理である。シグネチャーは特定のウィルスを示す可変長のパターンである。長さは数 Byte～数 10K Byte に渡り、最大 7000 種類のパターンが存在する。本処理も、ソフトウェアで実装する場合は非常にプロセッサ負荷の高い処理である。われわれは本処理の高速化にあたり、既に知られている Aho-Corasick 法 (3) と呼ばれるパターンマッチ検索アルゴリズムを改良して、新たに Multi-Byte Aho-Corasick 法を考案した。本方式により、高速化に効果的なハードウェアのパイプライン処理が可能になる。結果として、1 億文字/秒で入力されるデータストリームに対して、同時に 10000 種類のパターンを検索し続ける性能を有している。メールの平均サイズが 10Kbyte とすると、本性能は 10,000 通/秒のウィルスチェック処理に相当しており、ソフトウェアベースの既存ソリューションに比べ最大 1000 倍程度の性能向上に該当する。

Aho-Corasick 法は有限オートマトンの状態遷移処理であり、single character 単位の処理を前提としている。そのままハードウェア処理化すると、状態遷移表 (メモリ) を参照するたびにアクセス遅延が発生し性能が減少する。Multi-Byte Aho Corasick 法は、Aho Corasick の single character 単位の有限オートマトンでは無く Multi-Byte (N byte ： N character) 単位の有限オートマトンとして動作する。これによりハードウェアパイプライン動作が可能になり、アクセス遅延の影響を回避できる。Multi-Byte 単位の有限オートマトンの動作例を図 11 に示す。個々の状態遷移において、必ず 4 byte

を単位として判断する。4 byte 単位にする一方、後述するハードウェアエンジンでは 4 段のパイプラインを取る。個々のパイプラインは、入力するストリームにおいて常に 4 byte 周期のストリングを取り出して上記のオートマトン動作を行なう。N byte 単位の状態遷移と N 個のパイプラインの組み合わせにより、single byte 単位のパターンマッチと同様にすべての位相でマルチパターン検索することが可能になる。本アルゴリズムに関連して、われわれはウィルスパターン（約 10000 種類）を Multi-byte Aho-Corasick アルゴリズムの有限オートマトンに変換するソフトウェアも独自に開発した。

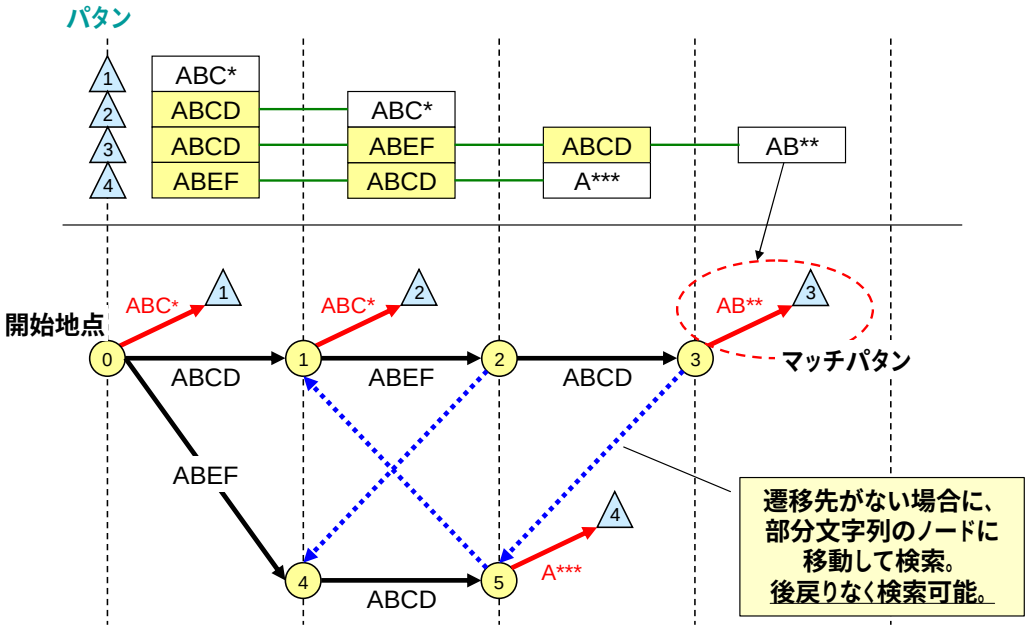

図 11　Multi Byte Aho Corasick 法のオートマトン動作

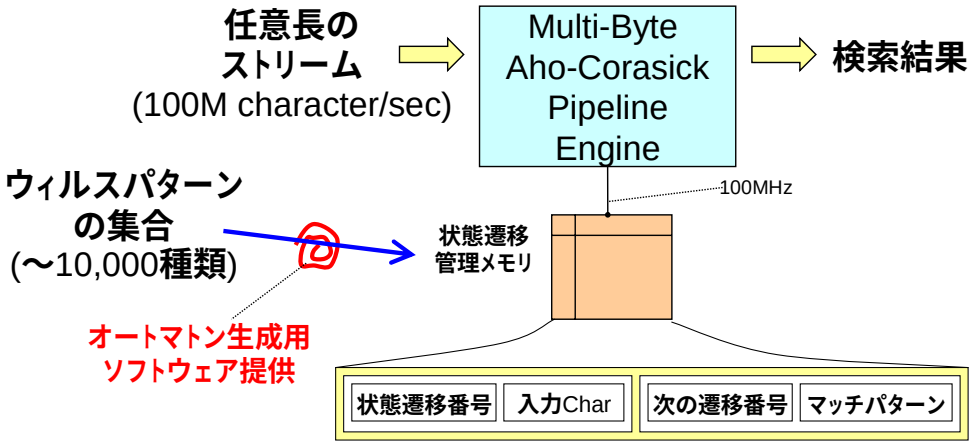

図 12　Multi Byte Aho Corasick 法のパイプライン処理

7. 新しいソフトウェア、ハードウェアの結合アーキテクチャの実装

迷惑メールフィルター、ウィルススキャン等への用途を前提に、従来オペレーティングシステムで実現されていた多くの負荷の重い処理のハードウェアエンジン化を行い、またアプリケーションソフトウェアにおいても負荷の重いコンテンツチェック処理のハードウェアエンジン化を行った。プロセッサ上のソフトウェアは付加価値の高いアプリケーション処理や高次の判断に専念することができ、システム全体で既存モデルの数桁上の性能向上が可能である。各種ハードウェアエンジン、プロセッサ間の通信はインテリジェントタスクスケジューラと呼ばれる機構を介しているが、迷惑メールフィルタ、ウィルススキャン等のアプリケーションでは TCP セッション単位の振り分け機能のみで良い。図 13 に新しいソフトウェア、ハードウェアの結合アーキテクチャの概要と、図 14 に上記アーキテクチャを実現したエンジンボード（略称：Hacone）を紹介する。本エンジンボードは前述した商品（図 3：SLMIT1000）に実装されている。

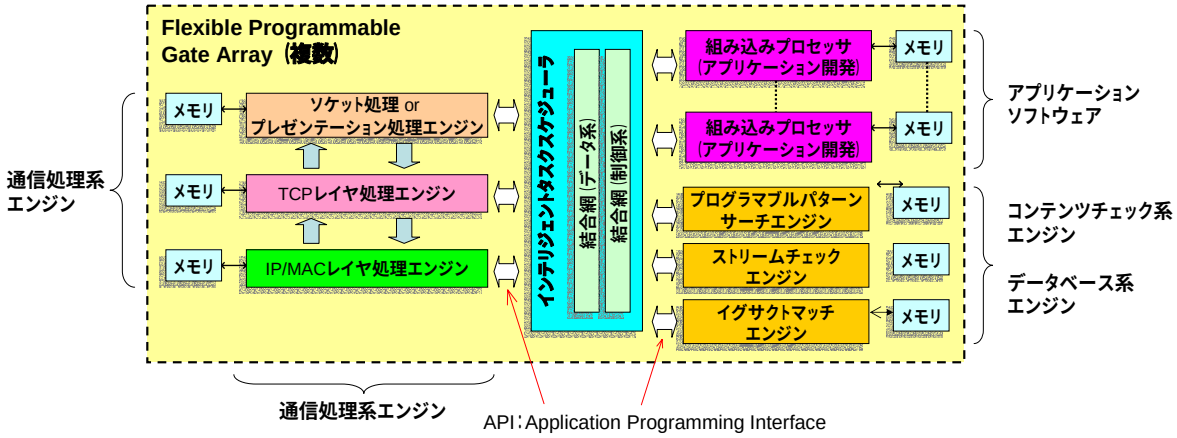

図 13　新規のソフトウェア、ハードウェア結合アーキテクチャ

図 14　エンジンボード概観

8. 将来展望

「汎用プロセッサ＋オペレーティングシステム＋汎用ソフトウェア」という誰もが前提としている既存のアーキテクチャを脱し、新たなハードウェア技術とソフトウェア技術の融合を図る研究開発を進めた。これにより、迷惑メール等の急務の課題に関して、既存の技術では達成できない性能を実現することができた。本性能は、セキュリティ保護における有効な武器であり、前述した SLIMIT 製品は ISP、キャリアのインフラ、企業の Front End 領域において、将来的な負荷増大に際しても基幹業務を守る性能を有している。また余剰性能を生かして、様々なアプリケーションへの取り組みも可能である。今後は新規のエンジン追加開発を進めるとともに、個々の API (Application Programming Interface) を整備することにより広範囲なアプリケーション開発の環境を提供していく。

＜関連特許＞　8 件

＜参考文献＞

1)　総務省、迷惑メールへの対応の在り方に関する研究会、中間報告 (2002.1.24)
2)　2003/12/3 NEC プレスリリース, 日本経済新聞、日経産業新聞、電波新聞
3)　A.V. Aho and M.J. Corasick, Efficient string matching, C.ACM, 18(6):333, 1975