柔軟性と高性能を備えたビッグデータ・ストリーム分析プラットフォーム「Blockmon」とその使用事例

Maurizio Dusi・Nico d'Heureuse・Felipe Huici・Andrea di Pietro
Nicola Bonelli・Giuseppe Bianchi・Brian Trammell・Saverio Niccolini

要　旨

本稿は、高性能データストリーム分析を実現する、新しい構成が可能な画期的なソフトウェアベースのビッグデータ・プラットフォーム「Blockmon」について説明します。Blockmonは、幅広い分野でアプリケーションが実行できるよう設計されています。ネットワークデータ処理・監視プラットフォームとして使用した場合、1台のコモディティサーバ上で、10GB/sの連続トラフィックを最高でレイヤ7（例. Deep Packet Inspection：DPI）まで取り扱うことが可能です。また、サーバログ処理プラットフォームとして使用した場合には、Blockmon上に構築した不正検出アルゴリズムにより、マシン1台で最大70,000cps（日本全体の電話トラフィックにも対応可能な最大2億5,000万BHCA）の分析が行えます。Blockmonを商品化することにより、オペレータ・ネットワークの分析やその他の分野、特にウェブ分析、金融市場分析などのアプリケーションへの適用が期待されます。

キーワード

●データ分析 ●モジュラープラットフォーム ●ネットワーク/データ処理

1. はじめに

近年、パソコンやスマートフォンといったインターネットに接続できるノード、及びEメールやSNSなど人同士をつなぐためのアプリケーションの増加は、インターネット経由で伝送されるデータ量を常時増大、多様化させています。その量は2〜3年ごとに倍増しており、この傾向は2015年まで続くと予想されています[1]。

その結果である、交換されるデータの多様性と容量は、既存のネットワーク監視・分析メカニズムにとっての課題となっています。ネットワーク監視・分析メカニズムは、以下を実現できるよう設計されなければなりません。

(1) 新たに生み出されるアプリケーションに容易に適応できる柔軟性
(2) 近リアルタイムのデータ処理を行って機密性・サービス品質・迅速な業務計画を保証できる性能の高さ

私たちは、Blockmonが上記の要件に応えられると考えています。Blockmonは小さな離散ブロックを使った、高性能かつ目的に応じて自由な構成が可能な計測手段を提供するシステムであり、BSDライセンスの下、オープンソースで入手可能です[2]。

本稿の第2章では関連する研究について述べ、第3章ではBlockmonのアーキテクチャについて説明し、第4章ではBlockmonアプリケーションの開発方法を示します。そして第5章では、不正検出への使用事例を概説し、Blockmonを利用することで達成した性能上の利点について示します。

2. 関連する研究

Blockmonは、既存の手法で用いられている設計原理を一部採用し、それらを強化することで、より広範な監視・分析アプリケーションを実現しています。また、チューニング用メカニズムが提供されているので、最新のマルチコア・マルチキューアーキテクチャ上での動作で優れた性能を発揮します。更にBlockmonでは、ブロック間接続のランタイム再構成が許可されているので、現状のネットワーク環境に合わせたトラ

フィックのオンラインデータ分析が可能です。

Blockmonは、プログラマブル・オンライン・ネットワーク計測とコンポーザブル・ネットワーキングに関する過去の研究を踏まえて構築されています。その1つである受動的な監視システム「CoMo」[3]では監視プラグインの概念が導入されましたが、実現される監視アプリケーションは依然として、厳密な事前定義や制限の多いコールバック関数に依拠しています。

Blockmonのモジュラー原理はClickモジュラールータ[4]に着想を得ていますが、Clickはパケット処理専用であり、そのモジュールはパケット通信しか行えません。更にClickでは、TCP接続の維持、データベースとの相互動作、監視やデータ分析に関連したさまざまな動作など、先進的な処理を簡単に実行することができません。

Yahoo!の「S4」[5]とTwitterの「Storm」[6]は、近年設計されたフレームワークで、ストリーム処理のための分散型プラットフォームというBlockmonと同様の目的を持っています。これらのプラットフォームとBlockmonのパフォーマンス（性能）とフレキシビリティ（柔軟性）の詳細な比較は、今後の研究課題の1つとなるでしょう。

3. Blockmonの概要

Blockmonは、ブロック、ゲート及びメッセージという概念を踏まえて構築されています。

Blockmonではブロックと呼ばれる処理ユニットを複数種類提供します。個々のブロックは、例えばリンク上にある個々のVoIPユーザーの計数など、特定の処理を実行しています。また、ブロックは、同種の2個のブロックを別々に初期化できるよう構成することで、システムのモジュラー性と柔軟性を保証します。各ブロックには入力ゲートと出力ゲートがあり、これらを使って複数ブロックを相互接続します。ゲートの数と使用目的は、ブロックの開発者によって定義されます。メッセージとは、ブロック同士が交換している情報の単位であり、あるブロックの処理結果を回線上の次のブロックへ伝えます。

監視・データ分析アプリケーションのための相互接続されたブロックの集合はコンポジションと呼ばれ（**図1**）、XMLで定義されています。Blockmonのコアとブロック自体はC++で実現されており、システムのランタイム制御はPythonベースのコマンドライン・インタフェースを通じて行われています。

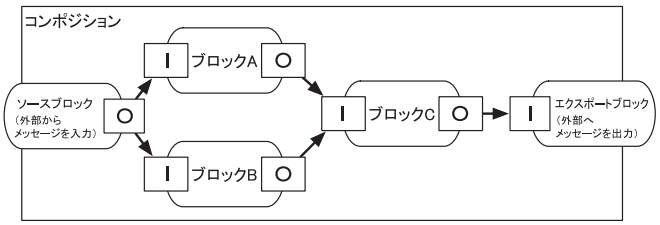

図1　Blockmon：ブロックの入力/出力ゲートを経由した相互接続による構成（コンポジション）

既に多数のブロックが開発済み・入手可能であり、これらのブロックはネットワーク監視コンポジションの設計に必要な基本的機能を網羅しています。例としては、パケット処理用の高速キャプチャーブロック（pcap、pf_ring、pfq）、フロー追跡用のブロック、統計用ブロック（Bloomフィルタとパケット・カウンタ）などがあります。ユーザーがこれ以外の追加ブロックを開発し、必要に応じて既存ブロックに接続することも可能です。BlockmonはC++で書かれているため、既存のC/C++コードで書かれた機能を、簡単にBlockmonへ移植することができます。

4. Blockmonアプリケーションの作成

(1) 開発者の視点：ブロック

Blockmon内部でアプリケーションを実行するため、ブロックはコアからBlockクラスを継承し、少なくとも下記2とおりのメソッドを実現し、適切な場合にはフレームワークからコールします。

- _configure()：コンポジションの起動時に、ブロックの構成に使用されているXMLエレメントへのリファレンスでコールされる。
- _receive_msg()：メッセージ及びそれを受信したゲートへのリファレンスでコールされる。

例えば、_receive_msg関数を多重定義してメッセージを取得し、カウンタをインクリメントするだけで、パケット計数用ブロックを制作することができます。**図2**に、パケット・カウンタブロックのためのコードを示します。

(2) ユーザーの視点：コンポジション

ユーザーがBlockmonアプリケーションを作成するには、コンポジションXMLファイルを記述し、アプリケー

```
class PacketCounter: public Block
{
 public:
  virtual void _configure(const xml_node&  n ) {};
  virtual void _receive_msg(std::shared_ptr<const Msg>&& m,
int index )
  {
    const Packet* p = static_cast<const Packet*>(m.get());
    ++m_count;
    m_byte_count += p->length();
  }
 private:
};
REGISTER_BLOCK(PacketCounter,"PacketCounter");
```

図2　Blockmonブロックの開発

```
<composition id="mysimplepfq">
  <general>
    <clock type="wall" />
  </general>

  <install>
    <threadpool id="src_thread" num_threads="1" >
      <core number="0"/>
    </threadpool>

    <block id="src" type="PFQSource" invocation="async" threadpool="src_thread">
      <params>
        <queues device="eth8"/>
      </params>
    </block>

   <block id="counter" type="PacketCounter" invocation="direct">
      <params>
      </params>
    </block>

    <connection src_block="src" src_gate="source_out" dst_block="counter" dst_gate="in_pkt"/>
  </install>
</composition>
```

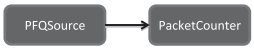

図3　Blockmonコンポジション

ションを構成するブロック及びそれらの間の相互接続を指定します。**図3**に、インタフェースに着信するパケットをリッスンするブロックと、パケットを計数するブロックで構成されたXMLコンポジションファイルの概要を示します。Blockmonでは、システムとして最良のパフォーマンスを得るため、コンポジションを指定する際に、ユーザーがブロックの動作及び相互作用を自由に指定できるようにしています。例えば、Blockmonは、ブロックを特定のスレッドへマッピングすることが可能であり、それらのスレッドを複数CPUコアへマッピングすることもできます。これによりユーザーは、どのコアがソースコードのどの部分を実行するかを決定することができ、キャッシュ効果、メモリアクセスまたはCPU利用などといった要素を最適化することが可能です。

5. 使用事例：VoIPSTREAM

Blockmonをデータ分析プラットフォームとして採用する利点を証明するため、私たちは既存の不正VoIP呼検出アプリケーション「VoIPSTREAM」をBlockmonに移植しました。その結果、VoIPSTREAMがテレマーケティングを行っている可能性のあるユーザーをリアルタイムで検出できることが実証され[7]、しかもこの処理において通常の（非テレマーケティング）ユーザーのプライバシーを保護することも確認されました。

Blockmonの性能評価にVoIPSTREAMを採用する理由は、以下の2つです。第1は、VoIPSTREAMはテキストデータで動作している点です。これによって、ネットワークパケット以外でも、Blockmonのデータ取り扱いの柔軟性が判断できます。第2は、私たちが示す実験結果は、VoIPや電話事業者の関心を得られるものだからです。

6. VoIPSTREAMのBlockmon設計

抽象化レイヤでは、アプリケーションは特徴抽出、処理、推論の3つの主要な部分に分割できます。**図4**に、VoIPSTREAMの設計を示します。

特徴抽出部は蓄積された呼ログ、または実トラフィックを読み出し、特徴を収集します。この特徴に基づいてユーザーの行動を分析します。

処理部は、各呼から抽出した特徴を利用して、長さが設定可能な時間ウィンドウ内での各ユーザーの全体の行動を追跡します。VoIPSTREAMでは、各ユーザーの全体的な動作は、各ユーザーの受信呼率、（2個の異なる時間ウィンドウ長における）確立発信呼率、新規被発呼率、合計発信呼時間、の5つの指標を計算することで得られます。これらの量は時間減衰Bloomフィルタで計算し、その結果を組み合わせてFoFiR（Fan-out Fan-in Ratio）、ACD（Average Call Duration）、URL（Unique Recipient List）の3種の時間ウィンドウ上のリスク評価モジュールにてリスクを算出します。FoFiRとは、確立した発信呼数と着信呼数の比であり、ACDとは個別の被呼者（新規被呼者）に対し確立された発信呼と総確立呼数の比であり、URLとは全体の平均呼期間とユーザーの平均呼期間の比です。各評価モジュールは、これらの比を利用して0〜1の範囲の異常度スコアを返します。

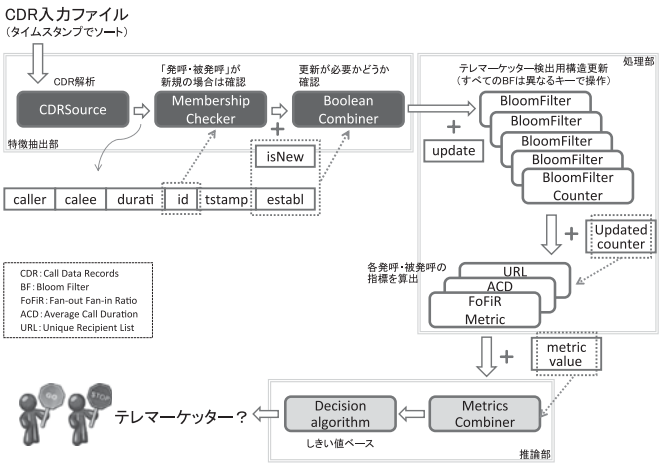

図4　VoIPSTREAMのBlockmonインプリメンテーション

最後に推論部は、上記のリスク評価モジュールを組み合わせて、ユーザーと異常度スコアとの関連付けを行います。私たちの組み合わせは、FoFiR、ACDとURLのスコアの加重和です。

ユーザーの異常度スコアに基づき、そのユーザーは所与の時間ウィンドウ内においてテレマーケッターらしく行動したかどうかを、しきい値ベースの決定アルゴリズムが判定します。

上記に説明した各部は、一連のブロックを使ってBlockmon内に実装されています。各ブロックが自己のタスクを遂行するのに必要な情報を確実に持つことができるよう、ブロック間ではメッセージが交換されます。

各ブロック及び各Bloomフィルタのパラメータは設定可能なので、コンポジションファイルによるブロックの作成中に指定することができます。例えば、各Bloomフィルタブロックのハッシュ表（これにより誤検出率が左右されます）のサイズ、決定アルゴリズムのしきい値、更には推論部の各リスク評価モジュールに指定する加重値などが指定可能です。

7. 結果

私たちは、8個の2.4GHz Xeon CPUコアと24GBのRAMを備えたx86サーバ上に、図4に示す個々のブロックを実装し、アプリケーションを実行しました。

スタンドアロン（非Blockmon）VoIPSTREAMで、1,000万の

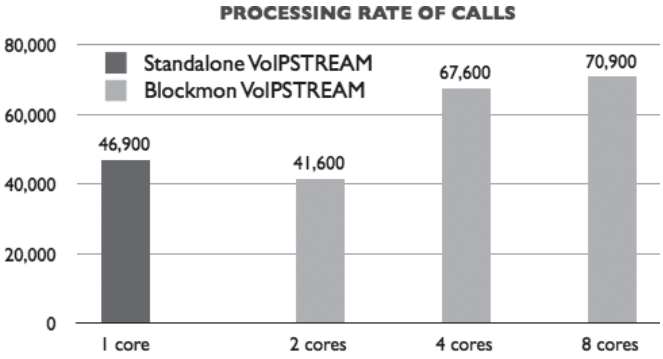

図5　VoIPSTREAMの処理速度：スタンドアロンとBlockmon

電話呼を含んだファイルの処理テストを3回行った結果、平均で46,900呼/秒の速度で処理できました。**図5** に示すように、2個のコア（1個は電話呼デコード用、1個は残りの処理用）を使用したBlockmon上のVoIPSTREAMの処理速度は41,600呼/秒でした。処理に専念するコアの数を4個（1個は電話呼デコード用、3個は将来のパイプライン用とし、計量ブロックは直接呼び出す）に増加するだけで、呼処理速度は67,600呼/秒と、スタンドアロンVoIPSTREAMと比較して44％も改善することができます。8個のコアを使用すると、呼の速度は70,900呼/秒と51％増加します。これは、スタンドアロンアプリケーションをBlockmonへ移植することでパフォーマンスが向上することを立証しています。また、オリジナルのコードをBlockmonに移植するには2週間しか掛からなかったことも言及すべき点です。

8. おわりに～将来の使用事例

Blockmonは、高い柔軟性と優れたパフォーマンスを併せ持つビッグデータ分析プラットフォームです。ネットワークトラフィック分析に使用した場合、1台のコモディティサーバで10GB/sの連続トラフィックを最高でレイヤ7（例. DPI）まで取り扱うことができます。また、サーバログ処理に使用した場合には、VoIPトラフィック中の不正を70,000cps（最大2億5,000万BHCA [*1]）で検出可能であり、これは日本全体の電話トラフィックすべてに対応可能な数字です。現在私たちは、コモディティサーバの台数に応じてパフォーマンスの拡張が可能

[*1] Busy Hour Call Attempt（最繁時呼数）

な、Blockmonの分散型バージョンの開発段階にいます。この作業の目的は、さまざまなコンテキストで使用されているその他のビッグデータ処理プラットフォーム（例えばYahoo!のS4やTwitterのStorm）とBlockmonを比較することです。この作業ではまた、VoIPSTREAMの分散型バージョン及びコンテンツの人気予測アプリケーションに加えて、Blockmonの柔軟性と高性能を実証できる、CTR（Clickthrough Rate）すなわちクリック率（ある特定のウェブサイトのため、オンライン広告キャンペーンの成功度合を計測する）と#countsすなわちハッシュタグ・カウント（Twitterで現在議論されている具体的なテーマを検出する）の2種類のウェブ分析アプリケーションの開発も行っています。

9. 謝辞

本研究の一部は、欧州委員会（EC）の第7フレームワーク・プログラム（EU FP7）が支援する研究プロジェクトであるDEMONS（契約No.257315）の支援を受けています。本稿に示されている見解及び結論は著者らのものであり、明示的または暗示的かにかかわらず、DEMONSプロジェクトまたはECの公的方針や支持を必ずしも表すものではありません。最後に、Blockmonの開発活動に関与したすべての人々に対する感謝を、ここに表明します。

参考文献

1) CISCO SYSTEMS：“Cisco Visual Networking Index: Forecast and Methodology,”http://www.cisco.com/en/US/netsol/ns827/net - working_solutions_sub_solution.html, 2011.6
2) Blockmon ソースコード https://github.com/blockmon/blockmon
3) G. Iannaccone：“Fast prototyping of network data mining appli - cations,”In Passive and Active Measurement Conference, 2006
4) R. Morris et al.：“The Click modular router,” SIGOPS Operating Systems Review. 33, 5 , pp217-231, 1999
5) L. Neumeyer et al.：“S4: Distributed Stream Computing Plat - form,”Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5691154
6) Twitter Storm ソースコード https://github.com/nathanmarz/storm
7) G. Bianchi et al.：“On-demand time-decaying bloom filters for tel - emarketer detection,” ACM SIGCOMM Computer Communica - tion Review Volume41,Number 5,pp6-12,October 2011 http://www.sigcomm.org/ccr/papers/2011/October/2043165.2043167

執筆者プロフィール

Maurizio Dusi
Research Scientist
NEC Laboratories Europe
NEC Europe Ltd.

Nico d'Heureuse
Research Scientist
NEC Laboratories Europe
NEC Europe Ltd.

Felipe Huici
Senior Researcher
NEC Laboratories Europe
NEC Europe Ltd.

Andrea di Pietro
CNIT Research Assistant
Department of Information Engineering
University of Pisa

Nicola Bonelli
CNIT Research Assistant
Department of Information Engineering
University of Pisa

Giuseppe Bianchi
Full Professor
CNIT/ University of Roma Tor Vergata, Italy

Brian Trammell
Researcher
Communications Systems Group
Swiss Federal Institute of Technology Zurich.

Saverio Niccolini
Manager
NEC Laboratories Europe
NEC Europe Ltd.

NEC 技報のご案内

NEC技報の論文をご覧いただきありがとうございます。
ご興味がありましたら、関連する他の論文もご一読ください。

NEC技報WEBサイトはこちら

NEC技報(日本語)　　NEC Technical Journal(英語)

Vol.65 No.2　ビッグデータ活用を支える基盤技術・ソリューション特集

ビッグデータ活用を支える基盤技術・ソリューション特集によせて
ビッグデータを価値に変えるNECのITインフラ

◇ 特集論文

データ管理 / 処理基盤

超高速データ分析プラットフォーム「InfoFrame DWH Appliance」
SDN技術で通信フローを制御する「UNIVERGE PFシリーズ」
大量データをリアルタイムに処理する「InfoFrame Table Access Method」
大量データを高速に処理する「InfoFrame DataBooster」
ビッグデータの活用に最適なスケールアウト型新データベース「InfoFrame Relational Store」
高い信頼性と拡張性を実現したExpress5800/スケーラブルHAサーバ
大規模データ処理に対するOSS Hadoopの活用
大容量・高信頼グリッドストレージiStorage HSシリーズ(HYDRAstor)

データ分析基盤

ファイルサーバのデータ整理・活用を支援する「Information Assessment System」
超大規模バイオメトリック認証システムとその実現
WebSAMの分析技術と応用例～インバリアント分析の特長と適用領域～

データ収集基盤

スマートな社会を実現するM2Mとビッグデータ
微小な振動を検知する超高感度振動センサ技術開発とその応用

ビッグデータ処理を支える先進技術

多次元範囲検索を可能とするキーバリューストア「MD-HBase」
高倍率・高精細を実現する事例ベースの学習型超解像方式
ビッグデータ活用のためのテキスト分析技術
ビッグデータ時代の最先端データマイニング
ジオタグ付きデータをクラウドでスケーラブルに処理するジオフェンシングシステム
柔軟性と高性能を備えたビッグデータ・ストリーム分析プラットフォーム「Blockmon」とその使用事例

◇ 普通論文

地デジTVを活用した「まちづくりコミュニティ形成支援システム」

◇ NEC Information

NEWS

スケールアウト型新データベース「InfoFrame Relational Store」が2つの賞を受賞

Vol.65 No.2
(2012年9月)

特集TOP