

# ビッグデータのための基盤ソフトウェア

## － IBMのビッグデータ・ソフトウェア －

日々大量に生成され、蓄積されていくデータを効率的に分析するには、データの容量 (Volume)、スピード (Velocity)、種類 (Variety) に応じて、最適なシステムを構築する必要があります。IBM では、大規模分散分析システムをサポートする IBM InfoSphere BigInsights (以下、InfoSphere BigInsights) を用いて Volume と Velocity に、リアルタイムの分析をサポートする IBM InfoSphere Streams (以下、InfoSphere Streams) を用いて Velocity と Variety に対応し、さまざまな分析に対して、効率的な分析環境の構築を実現します。本稿では、根底にあるビッグデータの分析のための思想から、今後の展望を含め、IBM のビッグデータに対する取り組みを紹介していきます。

### ① ビッグデータとその分析に対する課題

Web データ、遺伝子データ、自動車機器のセンサー・データなど、分析することでさまざまな知見やサービスを創出する可能性のある大量のデータは「Big data (ビッグデータ)」[1] と呼ばれ、現在 IT 業界のみならず通信、製造、バイオテクノロジーなど、さまざまな業種から大きな注目を集めています。

従来、データ分析はデータ・ウェアハウスに格納し多角的に分析する手法が主流でしたが、ビッグデータは以下のデータ性質や処理要件から従来の分析手法の適用が困難となっています。

#### ・容量 (Volume)

数百ギガ、数ペタバイトという規模の膨大なデータ量となるビッグデータは、今後も増加が予想されています [2]。Social Network Service (SNS) や Consumer Generated Media (CGM) の普及により爆発的に増え続ける Web データ、携帯電話や防犯カメラから刻一刻と提供されるセンサー・データなど、現在さまざまなビッグデータが存在しています。ビッグデータのデータ量とその処理量は一般的な規模のデータ・ウェアハウスで賄うのは困難な規模で、必然的に大規模な分散システム・分散処理が必要となります。また大規模な分散システムの導入・管理・運用は容易ではなく、分散アプリケーションの開発には、スケーラビリティを考慮する専門的な知識が必要となります。

## Infrastructure Software for Big Data

### - IBM's Big Data Software -

It has been observed that data is being generated and stored at an unprecedented scale and rate. In order to derive business insights from such vast amounts of data, we need to take separate yet integrated approaches depending on the volume, velocity and variety of data. IBM provides InfoSphere BigInsights, which can analyze Internet-scale volumes of data in many different formats, and InfoSphere Streams, which facilitates the continuous analysis of such data sets. This article introduces Big data and big analytics, presents current and future trends, and shows how IBM helps build best-of-breed platforms for the analysis of Big data.

#### ・スピード (Velocity)

ビッグデータの分析に対する処理速度も重要です。Web のオンライン・ショッピングでは、レスポンスに 2 秒以上かかると、ユーザーが離れるといわれています [3]、さらに電子商取引などのアルゴリズム・トレードや、センサー・データを対象にした分析では、数 100 ミリ、数 10 ミリ秒といった、より処理時間の短い分析が必要とされます。例えば、地震計のデータを分析する緊急地震速報の送信システムでは、センサー・データを受信後、50 ミリ秒以内で分析データを配信します。これらの処理は、センサー・データをデータベースに保存して、ほかのデータを照会しながら分析していく方法では時間的な要件を満たせない可能性があります。

#### ・種類 (Variety)

ビッグデータはデータ・ウェアハウスに単純に取り込める構造化されたデータだけではありません。音声、テキスト、動画といったさまざまな非構造化データが多く含まれます。一般的に企業におけるデータのうち、80% 以上は非構造化データになるといわれています [4]。これらのデータの分析には従来の Online Analytical Processing (OLAP) のような画一的な分析手法を利用できるとは限らず、データの種類や分析の目的により、臨機応変に分析手法を選択する必要があります。

このようにビッグデータの分析には、「容量 (Volume)」 「ス

ピード (Velocity)」「種類 (Variety)」への対応が重大な課題となります。現在、多くのベンダー、ユーザー企業がこれらの課題に対する解決策を模索していますが、本稿では IBM のビッグデータ分析の根本的な考え方と、分析のための基盤技術を中心に紹介していきます。

## ② ビッグデータと分析手法の分類

数百ギガバイト、数ペタバイトの膨大なビッグデータを瞬時にデータ・ウェアハウスに取り込み、数ミリ秒で分析を行うことは、現在のコンピューター・システムでは困難です。2011年8月時点で提供されている IBM System x において、サーバー1台に搭載できる最大のメモリー・サイズは eX5 (第5世代エンタープライズ X-Architecture) の3テラバイトであり、1ペタバイトのデータをメモリーに配置するだけでも、300台以上の eX5 が必要となります。データ量に応じて分析に要する処理量も増え、サーバー台数に応じて分析に要する通信量も増えるため、数ミリ秒で1つの分析を行うことは現実的には不可能です。そのため、ビッグデータの分析には、3つの課題、Volume、Velocity、Variety の観点からデータを分類し、分析を行っていく必要があります。

### 2.1 Data in Motion と Data at Rest

ビッグデータは、日々生成され、蓄積されています。例えば携帯電話会社では、携帯端末数千万台によって日々生成される電波・通話データから不正利用の検知を行い、過去の膨大な携帯利用状況から顧客の行動傾向の分析を行います。また、証券取引会社では、過去の膨大な証券データを分析することで、最適な投資対象を選定するアルゴリズムを計算し、最新のニュースや気象情報に応じて日々のトレーディングを行っています。

これらの分析において、特徴的なのは、「Velocity が重視されるデータ」と「Volume が重視されるデータ」が存在することです。携帯電話会社を例に挙げると、不正利用の検知に利用されるデータは Velocity が重視され、行動傾向の分析では Volume が重視されます。

Velocity が重視されるデータは、生成され、流入してくるスピードが速く、迅速に分析しなければならないデータです。一度に生成されるデータ・サイズは、数キロバイトから数ギガバイトの比較的小さいものもありますが、数十ミリ秒、場合によっては数マイクロ秒で、データを利用可能な状態にする必要があります。IBM のリサーチ部門の技術予測である Global Technology Outlook (以下、GTO) 2011 の“Petascale Analytics, Appliances and Ecosystems” [5] では、本データを Data in Motion (流入するデータ) と表しています。

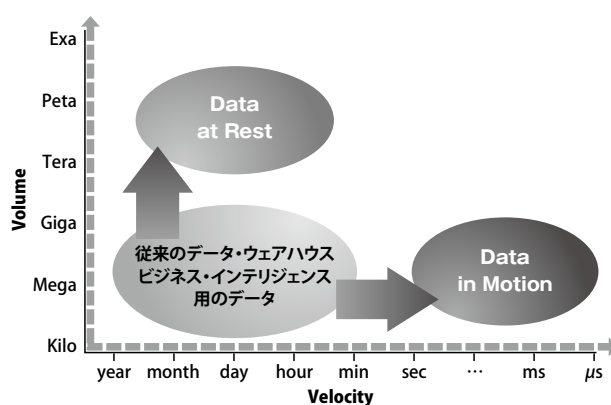


図1. ビッグデータの分類

Volume が重視されるデータは、迅速に分析される必要はないものの、そのデータ・サイズはテラバイトを超え、ペタバイト、あるいはそれ以上の大きなサイズのデータとなり得ます。このようなデータは、特定の分析を前提とせず保管されている場合が多く、さまざまな利用用途が考えられる反面、冗長なデータが多くなり、1台のサーバーやストレージ・システムでは管理が困難な規模になります。GTO 2011 では、本データを Data at Rest (蓄積しているデータ) と表し、これらの関係性を図1のようにまとめています。

Data in Motion や Data at Rest は、それぞれ別の分析手法を用いることで、従来の手法では困難だった分析処理が実現可能となります。

### 2.2 Reactive Analytics と Deep Analytics

GTO 2011 では、Data in Motion を対象にする分析を Reactive Analytics と呼び、Data at Rest を対象にする分析を Deep Analytics と呼んでいます。

対象のデータを磁気ディスク上のデータベースに格納し、SQL のようなクエリー言語を利用してデータベースからデータを取得する方法では、ミリ秒、マイクロ秒単位の速度で処理を行うことは困難です。そこで Reactive Analytics では対象データをデータベースに格納せず、常にメモリー上に確保しながら分析処理を行います。

また、Reactive Analytics では高い並列性も重要視されます。数キロバイト程度のデータであれば、逐次的な処理でも迅速な分析は実現可能です。しかし、数百メガバイトのデータになると、データ量の増加に伴って分析に必要な処理量も増加するため、分析処理を分割し、それらを複数の CPU で計算することで分析全体の処理速度を向上する必要があります。その際、通信時間が多大なオーバーヘッドとならない1台のサーバー内で大量の CPU を駆使する並列処理が、Reactive Analytics には有効です。

一方 Deep Analytics では、数百テラバイト、数ペタバイトのデータを分割し、複数のサーバーに配置して分析処理

を行います。現在のサーバーは数十テラバイト程度のディスクを構成可能なため、数百テラバイトのビッグデータでも、現実的なサーバー台数に格納することが可能です。大規模なクラスターを構成すれば、ペタバイトのデータも、分析可能となります。データ量が増えた場合、サーバーを増やすことで対応できることもクラウド時代では重要な特性です。

Deep Analytics では迅速な処理速度が分析に求められないとはいえ、データ量が膨大なため Reactive Analytics と同様に高い並列性が重要となります。Deep Analytics においては、データが複数サーバーにわたって配置されるため、必然的に分散的な並列処理を行う上での並列性が求められます。各サーバーが配置されたデータを読み込み、部分的な分析を行い、その分析結果の集約を繰り返していく並列処理が有効です。

### 2.3 ビッグデータ分析の Variety への対応

ビッグデータは、構造化されたデータだけではなく、音声やテキストといった、非構造化データである可能性があります。そのため、Reactive Analytics、Deep Analytics の2つの分析手法はさまざまなデータ形式に対応できる柔軟な分析手法である必要があります。

複雑なデータ形式に対応し多様な分析を行うには、高い記述力のあるプログラミング言語が必要です。Velocity や Volume が必要とする高い並列性を考慮することなく、より容易に分析処理の実現に集中できることが、求められます。

また、Reactive Analytics、Deep Analytics は、既存の分析手法と容易に融合することも必要となります。現在までに UIMA (Unstructured Information Management Architecture [6]) フレームワークを利用したテキスト解析や、Web 上で公開されている自動翻訳 API、IBM のマイニング・ツールの TAKMI [7] をはじめとした評判分析システムなど、有用で実績のあるライブラリーやサービスが存在しています。これら既存ライブラリー、サービスを利用した分析手法とビッグデータ分析とを柔軟に組み合わせることで、さらに応用力のある分析処理の実現が期待されています。

### 2.4 Reactive Analytics と Deep Analytics の融合

Reactive Analytics と Deep Analytics を融合することで、ビッグデータの膨大なデータ量に対しても、分析処理をさらに効果的に行うことが可能になります。

2つの Analytics の代表的な融合例は、2つの分析間で予測モデルを共有する手法です。この手法では、Deep Analytics で予測モデル (PMML [8] など) を生成し、そのモデルに基づいた分析を Reactive Analytics で実現します。再度予測モデルの修正が必要と判断した場合は、適切なフィードバックを行うとともに、Deep Analytics を再実

行します。

予測モデルの生成には、一般的に大量のデータを分析する必要があります。例えば、携帯電話上のアプリケーションの異常な通信パターンを検出するには、全ユーザーのメール情報を分析する必要がありますし、SNS から交友関係を分析するには、全ユーザーのコネクションやインタラクションを分析する必要があります。これらの分析に Deep Analytics を用いることで、アプリケーションの異常パターンを導き出したり、交友関係の中の重要人物を特定することができ、Reactive Analytics による効果的なアクセス制御や、キャンペーン活動を行うことが可能になります。

また、Reactive Analytics で、予測モデルと実際のデータを照合し、その乖離<sup>かいり</sup>が顕著になった際に、新たなモデルの生成を行うために、Deep Analytics を呼び出すといった融合も考えられます。このようにすることで、Reactive Analytics の精度も向上し、効果的な Deep Analytics が実現し、さまざまな状況に適用可能なソリューションが構築できます。

このように、2つの Analytics に対して、同一の予測モデルを生成・利用し、互いにフィードバックすることで、高い適用性を持った、統合した1つのソリューションを提供可能になります。

## ③ ビッグデータ分析のための基盤技術

IBM では、Reactive Analytics、Deep Analytics の実現に向けた、数多くのソフトウェア製品を提供しています。これらの製品を利用することで、ビッグデータに対して生産性の高い分析アプリケーションを開発することが可能となります。特に、IBM の Information Management に関する最も大きな展示会である Information On Demand (IOD) では、2011 年において、図 2 のように Data in Motion、Data at Rest の分析に対応する製品として、InfoSphere Streams と InfoSphere BigInsights を紹介しています。

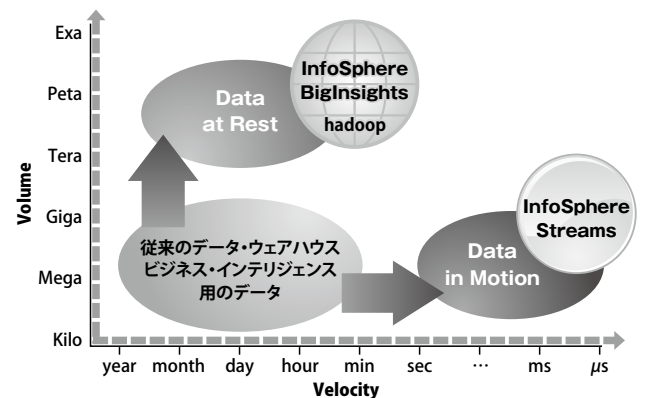


図2. InfoSphere BigInsightsとInfoSphere Streams

### 3.1 InfoSphere Streams

InfoSphere Streams は、2010年3月に製品発表された、Reactive Analyticsに有用な製品です。プログラム言語 Stream Processing Language [9] (以下、SPL) を用いて多様なデータ入力に応じて複雑な並列処理を簡潔に記述することが可能です。InfoSphere Streams は、SPL の記述やシステムの構成に応じてシステム全体の通信コスト・計算コストが最小になるように最適な並列処理方法を選択し、分析処理を実行します。

SPL では、1つの処理単位を表すオペレーターとオペレーターを連結するストリームを定義することで分析処理を記述します。オペレーターには入力と出力が存在し、1つのオペレーターからの出力を複製・分散し、複数のオペレーターで処理することで並列のストリームとして処理することが可能です。SPL の複製や分散の記述は高度なレベルで抽象化されており、システム全体の通信・計算コストを最小化するために、プロファイル結果を用いて、効率的に並列処理を行うようコンパイル・実行されます。

例えば、図3では、Beacon オペレーターで取り込まれたデータが RRSplit オペレーターにて処理後、ストリームが分岐され、2つの Functor オペレーターで並列処理されます。Functor オペレーターの処理結果は Pair オペレーターで集約され、Writer オペレーターにて外部へ結果が出力されます。これらのオペレーターは、異なるサーバーで実行される場合もありますし、すべて1台のサーバーの1スレッドで実行される場合もあります。InfoSphere Streams ではオペレーターを処理する最適なサーバーやスレッドを、プロファイル結果を基に、SPL のコンパイル時に決定し、効率的な処理を実現します。

各オペレーターでは、Java や C++ のライブラリーを呼び出すことが可能です。言語処理用のライブラリー (形態素分析など) や、統計分析用のライブラリー (IBM SPSS など) の既存のツールを組み合わせることで、複雑なデータに対しても、容易に分析することが可能となります。

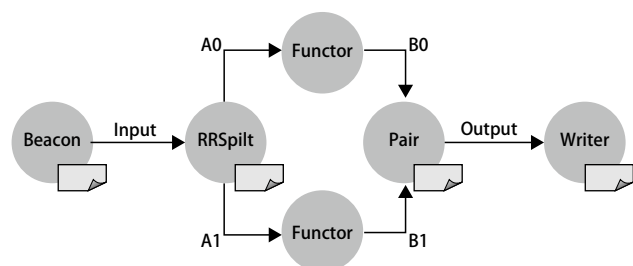


図3. SPLのプログラム例

### 3.2 InfoSphere BigInsights

InfoSphere BigInsights は、2011年5月に製品発表された、Deep Analyticsに有用な製品です。データの分散配置、並列分散処理を行う Hadoop をベースにソフトウェアが構築されており、Hive、Pig、Zookeeper、Hbase、Avro Hadoop など、Hadoop 上で稼働するさまざまなオープンソースが利用可能になっています。

InfoSphere BigInsights の大きな特長の1つは、プログラミング言語 Jaql (a query language desired for JSON) が利用可能であることです。Jaql は、IBM 主導でオープンソースとして開発されているクエリー言語で、JSON (JavaScript Object Notation) 形式のデータを処理し、Hadoop 上のデータを簡単に分析することを可能とします。Jaql は、Hadoop の並列分散処理フレームワークである MapReduce に変換され実行されるため、プログラマーは、耐障害技術などの Hadoop の有用な技術を利用しながらも、MapReduce を意識することなく、分析用プログラムを記述することが可能になります。また、データは軽量なデータ記述言語である JSON を利用して表現されるため、容易なデータ操作が可能です。SPL と同様、Jaql も Java のライブラリーを処理内で呼び出すことが可能なため、既存のライブラリーを有効活用できることも大きな利点です。

例えば、図4のようなデータが、employees.json という Hadoop のファイル・システムである HDFS (Hadoop Distributed File System) 上のファイルとして、分散配置されていたとします。なお、JSON では、{ } で囲まれたデータを、1つのオブジェクトとして扱います。ここでは、オブジェクトごとにサーバーに配置されていることを想定します。

```
{name: "Jon Doe", income: 20000, mgr: false, country: "Japan"},  
{name: "Vince Wayne", income: 32500, mgr: false, country: "USA"},  
{name: "Jane Dean", income: 72000, mgr: true, country: "Germany"},  
{name: "Alex Smith", income: 25000, mgr: false, country: "Japan"},
```

図4. Jaqlのサンプル・データ

このファイルは、図5のように読み込み、mgr 要素が true である値のみを抽出することが可能になります。なお、Jaql では “->” という演算子で処理を連結していきます。

```
read(hdfs('employees.json')) -> filter $.mgr;
```

図5. Jaqlのフィルター処理

「データを読み込み、フィルターを掛ける」という処理は、一見簡単な処理ですが、対象がビッグデータとなると、非常に難しい処理になります。データが複数サーバー上に配置されているため、プログラムを全サーバーに配置して処理し、全処理の終了後に、その結果を読み込めるようにしなくては

なりません。InfoSphere BigInsights はこれらの難しい処理を容易に行います。

「国ごとの income 要素の平均を取る」という処理を上記結果に追加すると、図 6 のようになります。ここでは、フィルタリングされた結果のうち country 要素でグループ化し、その income 要素の平均を計算しています。

```
read(hdfs('employees.json')) |> filter $.mgr;
-> group by $each = $.country info{$each, avg($[*].income)};
```

図6. Jaqlのグループ処理

上記処理も一見簡単な処理ですが、実際には各サーバーで計算されたフィルタリング結果を、country 要素ごとにまとめ、まとめた結果の中で income 要素の平均を取るという処理になります。フィルタリング結果は、各サーバーに散在していますので、country 要素ごとにまとめるには、フィルタリング結果を生成したサーバーから、平均を取るサーバーに通信する必要があります。Jaql を利用することで、高い並列性を保ちながら、これらの処理を容易に実装することが可能となります。

このように、Jaql を利用することで、SQL と同等以上の表現のクエリーを記述でき、意識することなく並列処理を行うことが可能になります。さらに、Java のライブラリーの呼び出しや既存のソフトウェアと統合することによる、幅広い応用も期待されます。

### 3.3 製品間の連携

InfoSphere Streams や InfoSphere BigInsights は、Java のような多言語のライブラリーを呼び出すことが可能なため、ほかの製品との連携が容易です。また、Java からの呼び出しも容易なため、WebSphere などの既存の製品と連携したソリューションも多く普及していくことが予想されます。

InfoSphere BigInsights での分析を容易にするためには、データを分析しやすい形式に変換し、大量に生成される分析結果を効率的に転送する必要があります。これらの処理は、ETL (Extract/Transform/Load) と呼ばれ、InfoSphere BigInsights とのデータの相互運用性が高まる予定の IBM InfoSphere DataStage (以下、InfoSphere DataStage) の最新バージョンを用いることで、効果的に実現することが可能です。

また、スケーラビリティと開発生産性に優れている InfoSphere BigInsights は、ほかのミドルウェアの基盤にもなり得ます。機械学習や評判分析など、今後ビッグデータを対象にするミドルウェアの多くは、InfoSphere BigInsights 上で実現されていくことも考えられます。

## 4 ビッグデータ分析基盤における技術革新の方向性

SPL、Jaql の原型は、IBM の研究開発部門により開発されました。現在でも言語の拡張や、ツールの研究・開発、言語処理系の最適化など、各種の機能拡張が IBM の研究開発部門によって行われています (図 7)。

SPL では、Reactive Analytics が行われることが想定されますが、言語や処理システムとしては Deep Analytics に応用することも可能です (図 7 の「Deep Analytics の SPL 応用」)。例えば、IBM 東京基礎研究所では、InfoSphere Streams を利用して、熱・速度といったセンサーデータをを用いて、機械の故障判定や、エラーの検出処理技術を研究しています。従来、通常数時間かかっていた分析の処理時間を、InfoSphere Streams の高い並列性を用いて、大幅に改善する成果も上がっており、SPL が、より汎用的な分析に適用可能であることを示しています。ほかにも、SPL を用いて、複雑なデータ変換を高速に処理する研究や、さまざまな分析用ライブラリーの開発も進められており、今後、InfoSphere Streams の幅広い活用が期待されます。

また Jaql をはじめとした InfoSphere BigInsights の処理系も、現在広範囲で研究・開発がなされています。ファイル・システムとして HDFS より高速な入出力を行える IBM General Parallel File System (GPFS: 本誌 58 ページ以下: 解説 ④ 参照) に対するサポートは、InfoSphere BigInsights の今後のリリースで行われる予定です。また、Jaql から MapReduce への変換の最適化、MapReduce 自体の高速化など、Deep Analytics の処理時間を高速化する研究も広く行われています (図 7 の「InfoSphere BigInsights 処理系の最適化」)。特に、Hadoop を処理するハードウェア、OS、JVM、Java の一体となった効率化・高速化は、IBM の研究開発部門の得意とする分野であり、お客様へのより高い付加価値の提供が期待されます。

これらの InfoSphere Streams の汎用化、InfoSphere

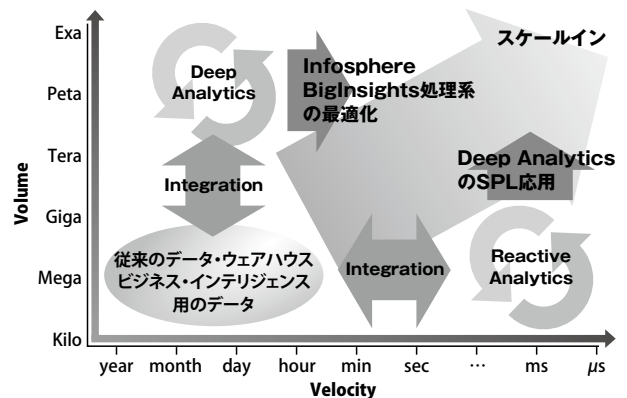


図7. 今後の技術革新の方向性

BigInsights の高速化の動向は、「VelocityとVolumeを併せ持つデータ」に対して Analytics を行う方向性を示しており、より汎用的なデータ分析手法の出現を目標としています。

さらに、これから起こるハードウェアの技術革新も、この動向を後押ししています。現在、IBM では、次世代の高速記憶デバイスである SCM (Storage Class Memory) や、高速な通信デバイスである光インターコネク、CPU とストレージが直接つながる技術の開発が進んでおり、データの読み書きや、デバイス間の通信速度が劇的に向上してきています (図7の「スケールイン」)。これらの技術は、GTO 2011 ではスケールインの技術として紹介されており、将来的には、ペタバイト級の分析を1台のアプリアンスで提供可能とすることを目標としています。

## 5 まとめ

ビッグデータの分析には、「容量 (Volume)」「スピード (Velocity)」「種類 (Variety)」という課題が伴います。IBM では「Velocity」「Variety」に対応する Reactive Analytics と、「Volume」「Variety」に対応する Deep Analytics を組み合わせることで、ビッグデータの分析手法を示し、それぞれ InfoSphere Streams、InfoSphere BigInsights の開発を進めています。これらのソフトウェアは、今後、ビッグデータの分析の基盤となり、ハードウェアやソフトウェアの研究・開発によって「より大きく」「より複雑な」ビッグデータの分析を「より速く」実現できるようになることを目指しています。

ビッグデータの分析は、Smarter Planet [10] の実現に有用なソリューションです。本文中で言及した携帯電話会社における分析、Web データの分析の事例のほかにも、数万台から提供される車載 GPS データと自動車の行動モデルに基づいたスマートな交通システムや、電力消費量と仮想マシンのモニタリングに基づいた最適なシステム構成を実現するスマートな IT システムなど、現在、世界中でさまざまな適用がなされており、その可能性はさらに大きなものになっていくでしょう。

### [参考文献]

- [1] Lynch C: "Big data: How Do Your Data Grow?," Nature 455: 28-29, (2008.9).
- [2] Grantz J.: "The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011," An IDC White Paper - sponsored by EMC, (2008.3).
- [3] Forrester Consulting: "eCommerce Web Site Performance Today: An Updated Look At Consumer Reaction To A Poor Online Shopping Experience," White Paper of Akamai Technologies Inc., (2009.8).

- [4] Grantz J.: "The Expanding Digital Universe," An IDC White Paper - sponsored by EMC, (2007.3).
- [5] IBM Research: "Petascale Analytics, Appliances and Ecosystems," Global Technology Outlook 2011 in IBM, [http://www.ibm.com/ch/ibm\\_sap\\_portfolio/pdf/Global\\_Technology\\_Outlook\\_2011.pdf](http://www.ibm.com/ch/ibm_sap_portfolio/pdf/Global_Technology_Outlook_2011.pdf) (2011.1).
- [6] Ferrucci, D., and Lally, A.: "UIMA: An architectural approach to unstructured information processing in the corporate research environment," Natural Language Engineering, (2004.6).
- [7] IBM Research: "Text Mining System -TAKMI-," [http://www.trl.ibm.com/projects/s7710/tm/takmi/takmi\\_e.htm](http://www.trl.ibm.com/projects/s7710/tm/takmi/takmi_e.htm)
- [8] Guazzelli A., Zeller M., Chen W., and Williams G.: "PMML: An Open Standard for Sharing Models," The R Journal, Vol. 1, No. 1, (2009).
- [9] InfoSphere Streams Information Center: "IBM Streams Processing Language," <http://publib.boulder.ibm.com/infocenter/streams/v2r0/index.jsp?topic=/com.ibm.swg.im.infosphere.streams.whats-new.doc/doc/ibminfospherestreams-whats-new-spl.html>
- [10] A SMARTER PLANET, <http://www.ibm.com/innovation/jp/smarterplanet/>



日本アイ・ピー・エム株式会社  
東京基礎研究所  
インフラストラクチャー・ソフトウェア  
スタッフ・リサーチャー

堀井 洋 Hiroshi Horii

### [プロフィール]

2004年、日本IBM入社。以来、同社東京基礎研究所にて、トランザクション処理システムおよび分散処理システムの研究に従事。工学博士、情報処理学会会員。  
<http://www.trl.ibm.com/people/horii/>



日本アイ・ピー・エム株式会社  
東京基礎研究所  
インフラストラクチャー・ソフトウェア担当  
シニア・テクニカル・スタッフ・メンバー

小野寺 民也 Tamiya Onodera

### [プロフィール]

1988年、日本IBM入社。以来、同社東京基礎研究所にて、オブジェクト指向言語の設計および実装の研究などに従事し、2008年より同研究所インフラストラクチャー・ソフトウェア担当部長。同社シニア・テクニカル・スタッフ・メンバー、理学博士、ACM Senior Member。情報処理学会、日本ソフトウェア科学会、各会員。  
<http://www.trl.ibm.com/people/onodera/>