

非順序実行型データベースにより多次元配列データを管理するマテリアルズ・インフォマティクス・プラットフォームの開発

浅原 彰規¹ 森田 秀和¹ 海野 英一郎² 池本 理³ 小野 寛太⁴

¹(株) 日立製作所 ²(株) 日立ソリューションズ ³サクシード(株) ⁴高エネルギー加速器研究機構

材料科学分野でITによる研究開発効率化の取り組みが進む中、筆者らはこれを支えるMI (Materials Informatics) プラットフォームを開発した。本システムは実験データや数値シミュレーションデータなどの多次元配列を非順序実行原理にもとづくRDBMSにより管理しており、それをWebブラウザを用いて可視化分析する機能を持つ。この設計により、従来、材料科学の研究者が2日ほどかけて開発した150億件のデータに対する処理プログラムと同等の処理をわずか1文で書け、かつ遜色ない速度で実行できた。これにより材料科学の専門家の作業負担なくさまざまな分析ができるようになった。

1. はじめに

材料科学分野で、近年「マテリアルズ・インフォマティクス」(以降、MI)というITを用い研究開発を効率化する取り組みが活発化してきている[1]。MIは、従来、実験の試行錯誤によって発見されてきた材料科学分野の知見を、情報通信技術(ICT)を用いることでより短期間に導き出そうという試みである。

MIにおいて最も主要な役割を占めるものの1つは、物理法則をあらわす方程式などを数値的に解いて物理現象を再現する数値シミュレーションの技術である。元来、新材料の開発においては、たとえばある金属に何をどのくらい添加すればより良い電気伝導性や塑性などを改善できるかを実験的に調べるなど、試行錯誤によって材料の探索が行われていたが、これを数値シミュレーションで置き換えれば時間的にも費用的にも効率が改善できる。今後、近年のGPGPU等による並列計算技術の発展にともないMIが普及し、多くの材料科学研究者が数値シミュレーションを行うようになると、生成されるデータは膨大になると予測される。また、従来よりも網羅的な実験を行うことで数値計算結果の検証や機械学習の入力に使うデータにするなど、新たなデータの利用形態が生まれることも想定される。

そこで筆者らは、この材料科学分野、特に金属材料を対象にMIのためのデータ管理活用を推進する情報システムを開発した[2][3][4]。本論文ではこれを「MIプラットフォーム」と呼ぶ。MIプラットフォームは材料科学にまつわる実験や数値シミュレーションのデータを大規模データベ

ースシステムに保管し、高度な分析機能やデータの再利用を促すための機能を提供する。これにより、材料開発者や研究者の間でデータの共有が円滑化され、横断的なデータ分析や予測等のデータ分析ができるようになる。

MIプラットフォームの中核となるのは、実験データや数値シミュレーションデータなどを管理するデータベースシステムである。材料科学のデータには事後的にさまざまな分析処理を加えて用いることが多く、このデータベースシステムにも分析処理を容易に実装できるような機能が求められる。というのも、分析のたびにデータを逐一他のシステムへ移行していたのではあまりにも利便性を損なうためである。一般的にこの目的でよく用いられるのが、SQLで処理を受け付けるリレーショナルデータベースシステム（RDBMS）である。SQLは非常に汎用的で平易であり、MIプラットフォームの利用者である材料科学の専門家にも学習しやすく、一度覚えれば多種の用途で活用可能な言語である。また学習のための書籍類が入手しやすいなど、ユーザ層を拡大しやすい利点もある。ところが、MIプラットフォームが扱うべきデータには画像や時系列信号等の表形式に変換しにくいデータが含まれており、一般にこのようなデータをRDBMSで扱うと処理効率が悪いという課題が知られている。処理効率を改善するためのデータの種類の専用管理システムが提案されている[5]が、それは利用者に対してより多くの学習負荷を強いることであり、材料開発という観点で見ればかえって効率を悪化させることになりかねない。つまり、特別な工夫のないSQLを用いても高い処理効率が発揮できることが望ましい。

本論文では、MIで用いられるデータの用途について整理し、要件を満たす表形式に変換し汎用的なRDBMSを用い管理するシステムについてプラクティスを述べる。このシステムは高速検索にすぐれた非順序実行原理[6][7]にもとづくデータベースシステムを採用している点が特徴である。この方式の高速性により、処理性能の問題が回避でき、利用者にとっての利便性とのトレードオフが解消できると期待される。また、実際の金属材料のシミュレーションデータを対象に、そのフィージビリティを評価した結果についても報告する。

2. MIプラットフォームの設計

2.1 対象分野

図1に材料科学の対象についてまとめたものを示す。材料科学の対象は多岐に渡っているが、大きくはプラスチック、ゴム、繊維等の有機材料と、セラミック、金属等の無機材料に分類される。有機材料は炭素Cを中心に、水素Hや窒素Nが組み合わせられてできる材料の総称である。無機材料は有機物でない材料全般を指し、金属材料とセラミックやガラスなどのSiO₂を主原料とする非金属材料に分類されることが多い。特に金属材料は電気、電子部品や自動車航空機船舶等の機体など機械類の材料として広く用いられており、材料科学分野でも大きな位置を占める。

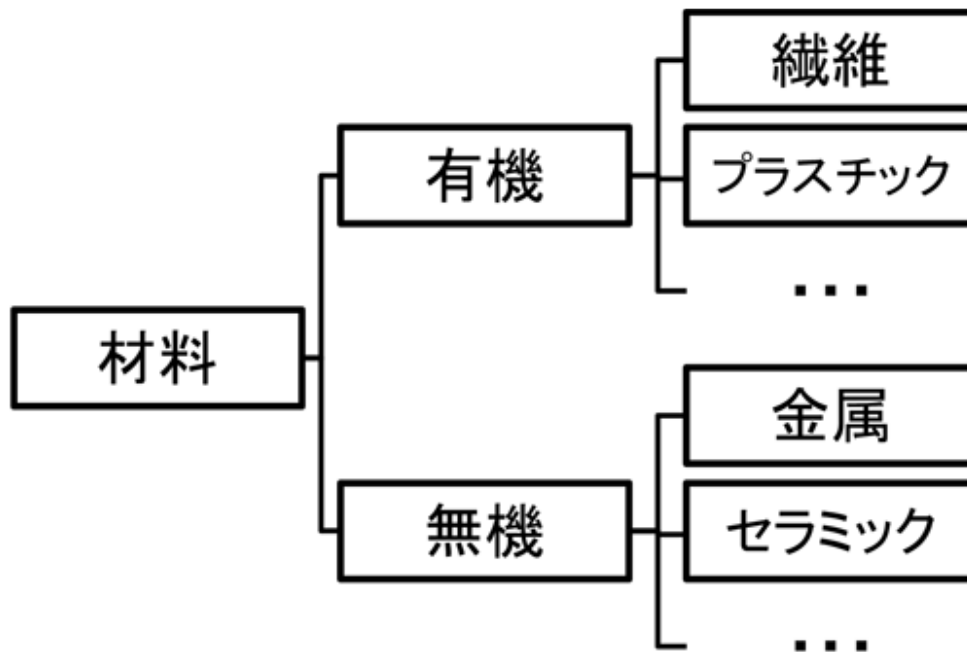


図1 材料の分類

金属材料の開発では、金属材料の組成（使用されている金属元素）や加工のプロセス（加熱処理など）を決定して実際にサンプルを作成して、材料特性の評価が行われている。評価すべき材料特性は力学的なもの、電気的なものや化学的なものなど種々あるが、知られている限りにおいては万能な金属というものは存在しない。用途ごとにどの種の金属が適切かが知られており、それにもとづく試行錯誤により材料特性の改善が図られる。この試行錯誤の過程では、時として1つに数百から数千時間を要するケースもあり（たとえば、金属の経年劣化、疲労特性を評価するには相応の時間をかける必要がある）、試行錯誤の回数を少しでも削減することが材料開発の効率化につながるとされている。

2.2 材料科学研究におけるデータ

材料開発の試行錯誤は理論や経験則によりできる限り無駄がないように計画される。しかしそれでも1つ1つの試行には多くのコストや時間がかかるため、その効率化の要求は高い。そこで数値シミュレーション等を用いてその回数を低減するのがMIの取り組みである。ただし、実は試行錯誤の回数自体は減っておらず一部を仮想的に実行できたのみである、という点に注意を要する。すなわち、試行錯誤の回数を低減する方法は別に検討が必要である。そこでたとえば、他の研究者が同様の計算を過去に行った事例がないかを検索できるようにするなど、データの活用により試行錯誤を低減することが考えられる。すると管理対象とすべきデータは以下のように整理できる。

数値シミュレーション

数値シミュレーションはあるモデルにもとづき物理現象を再現したものであり、その出力はそのモデルに沿った形式である。基本的な物理法則の多くは時空間座標や運動量を用いた偏微分方程式として記述されるため、時空間を有限の微小な要素に分け、近似的に偏微分方程式を解く方法などが用いられる（有限要素法）。それ以外にも材料に関する経験式などは大半が偏微分方程式などで記述され、差分法など数値的解法が用いられる。

計測結果

材料の計測データは大まかには、試料の状態の計測と、試料の材料特性の評価の2種類に

分類できる。試料の状態の計測として代表的なものとしては試料表面を電子顕微鏡で観測した画像や、X線等を試料に照射して得られる散乱パターンの画像が挙げられる。他方の材料特性の評価は、試料の物理的な特性を計測したもので、電気伝導率などを直接的に計測したり、試料を加熱したり、圧力を加えたりして、その特性を計測した結果である。

MIプラットフォームにはこれら数値シミュレーションや計測結果に対して、分析を容易に行える機能が求められる。そしてまた、データの分析が可能であるためには、数値シミュレーションのデータモデルをできる限りそのまま扱うことが求められる。たとえば、Schrödinger方程式の解は時刻 t における波動関数 $\psi(t, \mathbf{x})$ の分布であり（なお、 $\mathbf{x}=(x,y,z)$ の3次元位置ベクトル）、6次元の配列 $(t,x,y,z, \text{Re } \psi, \text{Im } \psi)$ で記述される（ $\psi(t, \mathbf{x})$ は複素関数）。あるいは、たとえば粒子の運動は座標の時間変化 $\mathbf{x}(t)$ で記述でき、 (t,x,y,z) の4次元配列となる。このように数値シミュレーションの結果は通常多次元の配列として記述でき、NetCDF[8]等の配列データ向けのデータ形式が用いられる。また、計測結果についても同様に、材料の組成、材料特性などを列挙すると、各材料ごとの数値が並んだ配列として扱える。したがって、この多次元の数値データを管理するシステムが必要となる。以降では、このような多次元の配列データを扱うデータ管理システムを多次元配列データベースと呼ぶ。現状、多次元配列データベースとしてRDBMSなどのデータ管理システムはほとんど活用されていないのが実情である。

2.3 MIプラットフォームの構成

図2に今回開発したMIプラットフォームの構成概略を示す。MIプラットフォームは、前節にて述べた多次元データを保存管理し、可視化、集計、共有等がしやすい機能を提供する。多次元配列のデータは一般的にデータ量が多く、集計等かける前に個々のデータに適したノイズ除去等の前処理を要する場合が多い。また現状では、データ形式も事例ごとに異なっており、一律の扱いは困難である。そこで本システムでは、データの前処理をする部分と可視化や分析処理をする部分が多次元配列データベースを通じて疎結合するシステム設計とし、その機能を大まかに2つの機能群に分類した。1つ目の機能群はデータの加工、取り込みを行うためのデータ管理機能であり、もう1つはそれをWebブラウザ等を通じて可視化する可視化分析機能である。両者の中間には多次元配列データベースが存在しており、データ管理機能で加工されたデータが多次元配列データベースに格納されると、可視化分析機能を通じWebブラウザ上で容易に可視化できる。この設計により、たとえば新たなデータ形式に対応せざるを得なくなった場合のシステム改修の範囲をデータ管理機能の部分のみに限定することができる。また、多次元配列データベースとしてできるだけ汎用的なデータベースシステムを採用すれば、可視化分析機能については既存のソフトウェア資産を活用しやすいという利点もある。たとえば近年のビッグデータの急速な発展の要因ともなるオープンソースソフトウェア（OSS）は高頻度に機能拡張されるが、その最新機能を迅速に導入できるのである。

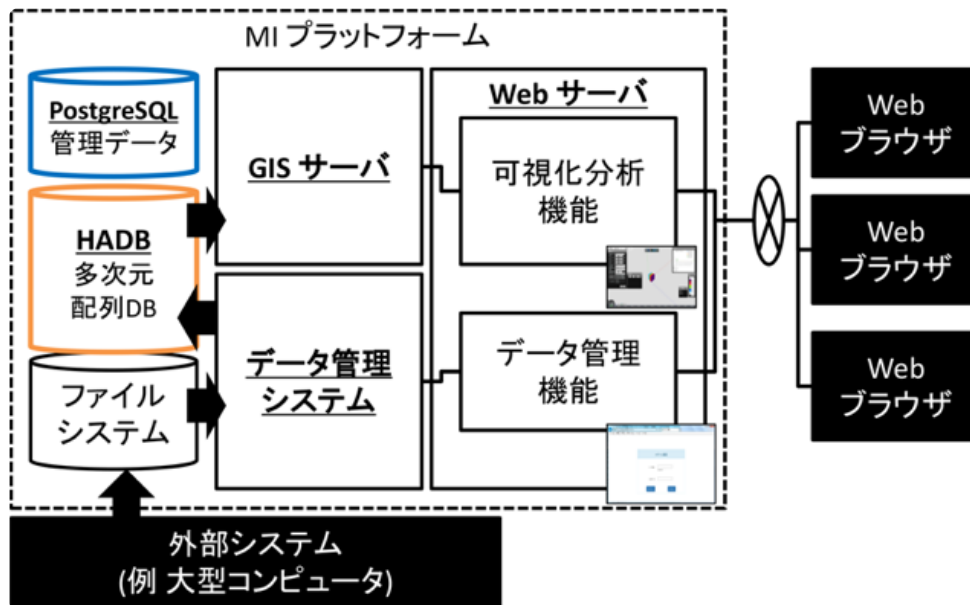


図2 MIプラットフォームの概略

2.3.1 データ管理機能

表1にデータ管理機能の一覧を示す。データ管理機能は、実験や数値シミュレーションの多次元配列データを取り込み、多次元配列データベースに格納できる形に変換、再構成する機能である。

表1 データ管理機能

Function name	Detail
Format conversion	Converts into standard formats
Data cleansing	Noise reduction, interpolation etc.
DB import	Registers the data into RDBMS

多次元配列データベースに格納されるデータは大きく、特に大型コンピュータで計算された数値シミュレーションの結果等であるため、FTP等の一般的なファイル転送を用い、図2中のファイルシステムにファイルとして置かれる。このファイルが本システムがサポートしているデータ形式であれば、データ管理システムはこのデータを読み取り、コマンドラインで指示されたとおりにノイズ除去等のデータの加工ができる。この加工はWebサーバ上のサーブレットとして実装されたデータ管理機能のユーザインタフェースを通じてWebブラウザからも指定できるようになっている。ここで加工可能な範囲を超えたデータの前処理には、一般的にETL（Extract, Transform, Loadの略。データ抽出や変換など）ツールと呼ばれるデータ統合ツールの活用が考えられる。

この処理の結果は、DBインポート機能により、一度標準的なCSV形式で出力されて多次元配列データベースにインポートされる。これは、データベースシステムは一般的にSQL文によるデータ投入よりもCSV形式などのファイルから一括読み取りしたほうが処理が高速なためである。

2.3.2 多次元配列データベース

多次元配列データベースはデータ管理機能でインポートされたデータを保管し、SQL文を受け付けて結果を返すシステムである。多次元配列データベースシステムの実装としては、多次元配列データを扱うための実装が存在し[9][10][11]、多次元データを扱うためのSQL拡張[12]も提案されている。また、KVS (Key Value Store) などのいわゆるビッグデータ向けのデータ管理システムの適用も考えられるとともに、画像データ等の非構造データを対象にした分散処理プラットフォームなども存在する。ただし、それらに対応しようとする、各ソフトウェアに多種多様なデータ形式に対応するための改変が必要になってしまい、拡張性が著しく失われてしまう。また、材料科学の研究者が直接多次元配列データベースにアクセスして集計等を行うことを想定すると、学習にかかる負荷を最低限にすることが求められる。その際には分析言語としてのSQLの機能は有益であり、仮にRDBMSを用いないとしても、SQLでの操作をサポートすることが望ましい。

そこで本システムでは一般的なSQL文を受け付けるRDBMSを用い、多次元データの要素1つ1つをレコードとして管理することで多次元配列データベースを実装する。これによりRDBMSに対応した種々のソフトウェアを容易に使用できるようになり、SQLによる分析処理も一般的な文法のみで簡単に記述できるようになる。ただし、このデータ構造では処理時間等が非常に大きくなるという点に課題がある。この点については2.4節にて詳しく後述する。

2.3.3 可視化分析機能

表2に可視化分析機能の一覧を示す。今回、特に取り扱う多次元データが位置と時間の情報で関連付けられている点から、地理空間情報システム (GIS) を活用して可視化分析が行いやすいようにした。GISは通常、地図情報などの広域 (数km以上) の空間に分布するデータや図形を管理するために用いられているソフトウェアである。本システムの扱う材料科学のデータは、nm単位と広さこそ違えど、時空間的な情報である点は共通しており、多くの機能がそのまま転用できる。またGISで気象や災害等の物理現象に関する数値シミュレーションが管理されている事例[13]がある点も、本システムの目的との類似性が見られる。また、地理空間情報分野の国際標準仕様に対応することで機能拡張性を高めることもできる。

表2 可視化分析機能

Function name	Detail
2D view	Visualizes 2D spatial data
3D view	Visualizes 3D spatial data
Timeseries view	Animate along time
Attribute view	Displays quantities at specified point
Graph view	Shows statistics of the data
Attribute search	Shows data satisfying specified conditions
Downloader	Downloads shown data in specified format

Web経由で動作可能なGISはさまざまなものが知られているが[14][15]、今回、日立ソリューションズ製品「GeoMation®」とOSSのCesium[16]、three.js[17]を組み合わせ用いた。図3にその画面のスクリーンショットを示す。ここで中央に表示されているのが多次元データである。多次元データの表面の物理量はGISにより画像化されており、その画像がWeb3Dで描かれ

た直方体の3Dオブジェクト上にテクスチャとして貼り付け表示されている。この構成により、Webブラウザで扱うデータが削減でき、スムーズに3次元空間上のデータを表示できる。また、物理量についてのヒストグラムを表示するヒストグラムダイアログや、画面上でマウスによってクリックした地点の物理量を表示する属性値ダイアログ、マウス位置情報ダイアログなどのユーザインタフェースを備えており、数値として物理量が把握できるようになっている。また、タイムラインツールやコマンドダイアログでは、画面に表示する時刻や空間的な範囲を指定できるようになっている。これを変更すると、本システムは多次元配列データベースから時空間的な断面の情報を取得し、画面を更新するようになっている。したがって、多次元配列データベースの断面検索の性能は非常に重要である。

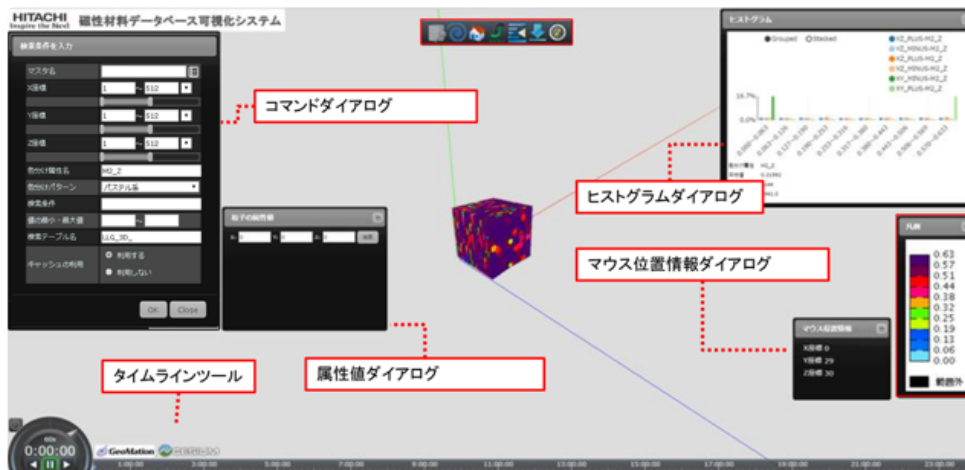


図3 可視化の画面

以上の通り、本システムはWebブラウザを通じて多くの可視化の機能を提供している。これにより、Webブラウザを備えたコンピュータから簡単にデータが確認できるため、材料科学の研究者間での情報共有が円滑化されると期待される。

ここで用いた製品やOSSはRDBMSのプログラミングインタフェースであるJDBCやODBCに対応しているため、多次元配列データベースシステムとして汎用的なRDBMSを採用していれば結合が容易であり、拡張性を備えることとなる。また、研究者が仔細に分析をする場合には、コマンドラインやRやPythonプログラミングを用いた処理、あるいは直接SQL文を書いて実行可能である。このように、標準的なRDBMSを用いることで、既存あるいは最新のソフトウェアを活用可能になったといえる。

2.4 多次元配列データベース管理の課題

前述の通り、多次元配列データベースの管理するデータは大きいのが、特に問題となるのはレコード数の多さである。一般に多次元の配列の要素数は、次元数に対して指数関数的に増大してしまう。たとえば三次元座標と時刻のそれぞれ100分割した数値シミュレーションだと1つでも総数が 10^8 、つまり1億件にもなる。

また、配列に対して行うべき分析等の処理内容は、材料科学的な新発見を目指す分析なので、画一的ではなく試行錯誤的なものが多い。よってデータ管理システム内でさまざまな分析の処理を実行できるようにすべきである。たとえば、物質のエネルギーを詳細に計測しようとしても実験的には物質全体のエネルギー総和の形でしか計測できないとき、その物質内部を微細な要素に

分けて数値シミュレーションすることで、物質内部のエネルギーの由来を理解しようとするところがある。このとき要素の種類ごとの平均エネルギーや要素間の分散、時系列変動など、実験で計測できない物理量でも、数値シミュレーションならば確認できるため、材料特性の発現メカニズムの理解につながり、新材料設計の指針となるのである。このように、微細な要素の挙動の物理法則が判明しているとき、それが全体の物理量にどう影響するかを調べる目的で数値シミュレーションが用いられる。よってデータ管理システムは種々の集計演算ができるよう設計されなければならない。

実際には、数値シミュレーションを行う時点で単純なモデルへの帰着がなされている。上述の通り、数値シミュレーションの多くは局所的な物理法則から全体の挙動を再現するために実行される。そのため物質を微細な要素に分けたモデルを立て、その要素の局所的な挙動の組合せで全体の複雑な挙動を再現しようとする。たとえば空間を微小なセルに分割して隣接セルとの相互作用を扱うようにしたり、要素間を接続してネットワーク構造をつくり物理量を伝播させたりする。つまり、物理法則がシンプルに表現できるような要素のモデルがすでに定義されているのである。したがって、多次元配列データベースシステムは数値シミュレーションの結果をできる限りありのまま扱えば、複雑な物理現象を表現したデータであっても精度良く扱えるものと考えられる。ただし、この方策は、数値シミュレーションにおける課題を同様に内包する。すなわち、処理時間に課題がある。

そこで本システムでは、並列データベースを用いることによってそれを解決する。並列データベースは複数のハードウェアを並列的に利用するデータベースシステムである[18]。上述の通り、本システムの多次元配列データベースが処理すべき配列の次元数は多いが、配列の要素ごとに並列計算が可能であれば並列実行による高速化が期待できる。配列に対する分析処理には並列化が困難な計算も含まれ得るが、一般的に時空間的な分割による有限要素法の計算結果では、時空間的に遠いものは関連性が低くなるという性質があり、分析計算も時空間的に近傍のデータとの関連性しか求めないことが多く、並列性が高くなりがちであり、結果的に並列データベースがその性能を発揮しやすいことが多いと想定される。そのため、SQLのインタフェースで並列処理が可能なデータベースシステムであれば、分析の工夫をあまりしなくとも、十分な高速化が期待できるのである。

今回のシステムでは、並列データベースとして、高速処理可能な非順序実行原理にもとづくRDBMSを用いることで解決した。具体的には、同原理を採用したソフトウェア製品であるHitachi Advanced Data Binder^{☆1}[20] (HADB) を用いて、数値シミュレーション等の内部のデータ構造そのままのテーブル設計（上述の例でいえば、空間のセルやネットワークのノードをレコードとするテーブル）でデータ管理するようにした。これにより、各レコードが内部のデータ構造と同等の情報を持つようになるため、数値シミュレーションと同等の柔軟性を持って集計処理等ができ、その高速性が発揮されれば処理時間にもほとんど支障がなくなると期待される。

3. 多次元配列データベースシステムの性能評価

3.1 評価の目的と評価対象データ

上述のとおり、非順序型原理にもとづくデータベースを使えば解析の利便性と処理速度のトレードオフが解消できると期待されるものの、依然として多次元配列データベースには性能上の懸念が大きい。そこで実際に設計の妥当性を検証するための性能評価を行った。

今回の評価では、磁性体の理論Landau-Liftshitz-Gilbertequations (LLG) の数値シミュレーション結果のデータを対象とした。図4にその概念図を示す。LLGシミュレーションは3次元空間の格子に配された要素によって表現される、有限要素法の数値シミュレーションである。整数で指定される空間座標値 (x,y,z) に物理量がそれぞれ付与されており、それが時間tとともに変化していく。このシミュレーションは、本来は大型計算機を用いて長時間計算する必要がある。しかし、本システムの開発以前に実際の材料研究の一環として行われた既存データがあった点、および空間的に分布する物理量が時間発展する有限要素法の典型的な形となっていて一般性が高い点から、本システムの評価に用いることとした。前述の通り、数値シミュレーションや計測の結果に対しては集計演算を行うことが基本であり、データのインポート後はRDBMSの機能にあるupdate文等を実行することはほとんどないため、このデータに対するselect文によって性能評価する。

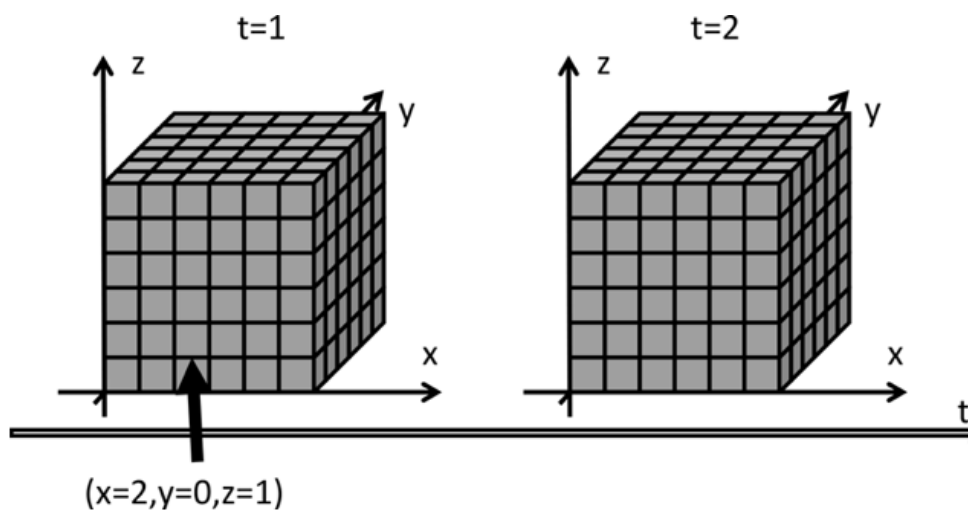


図4 LLG データのモデル

データベースシステムは数値シミュレーションのモデルをそのまま引き継ぐべきである、という観点から表3に示すとおりスキーマとした。空間座標はx,y,zのそれぞれ512分割したセルで表現されており、3次元なので3乗、134,217,728の格子点がある。各格子点はいずれかの粒子（材料を構成する塊）に所属しており、その粒子を一意に識別する番号がgrain_idに格納されている。また、それぞれの粒子には時間変化する物理量に関連づいており、各レコードは各格子点についてstart_timeからend_timeまでの時間の物理量を表現している。時刻のステップ数は総合で116ステップあるため、データの総数としては15,569,256,448件（およそ150億件）レコードとなる。

表3 LLG 時空間データテーブル

Name	Data type	contents
st_dataid	integer	primary key
exp_id	integer	ID of experiments
start_time	integer	Start of time range
end_time	integer	End of time range
x	integer	x coordinate
y	integer	y coordinate
z	integer	z coordinate
hx	double precision	x of magnetic field
hy	double precision	y of magnetic field
hz	double precision	z of magnetic field
m1_x	double precision	x magnetization of Elem.1
m1_y	double precision	y magnetization of Elem.1
m1_z	double precision	z magnetization of Elem.1
m1_theta	double precision	θ magnetization of Elem.1
m1_phi	double precision	ϕ magnetization of Elem.1
m2_x	double precision	x magnetization of Elem.2
m2_y	double precision	y magnetization of Elem.2
m2_z	double precision	z magnetization of Elem.2
m2_theta	double precision	θ magnetization of Elem.2
m2_phi	double precision	ϕ magnetization of Elem.2
a_x	double precision	uniaxial anisotropy x-axis
a_y	double precision	uniaxial anisotropy y-axis
a_z	double precision	uniaxial anisotropy z-axis
a_theta	double precision	uniaxial anisotropy θ -axis
a_phi	double precision	uniaxial anisotropy ϕ -axis
E_tot	double precision	Total energy
E_zee	double precision	Zeeman energy
E_exc	double precision	Excitation energy
E_ani	double precision	Anisotropy energy
E_dip	double precision	Dipole energy
grain_id	integer	ID of grains
ms	double precision	milliseconds for each step

今回、表4のハードウェアにHitachi Advanced DataBinder 03-03をインストールし、CSV形式に変換したLLGのデータをインポートし評価環境とした。なお、空間検索の高速化のため、(exp_id,grid_id)、(x,y,z,exp_id,grid_id)、(y,z,x,exp_id,grid_id)、(z,x,y,exp_id,grid_id)、(exp_id,grid_id,t)にマルチカラムB-treeインデックスを構築してある。以降、この環境をHADB環境と呼ぶ。

表4 HADB 環境のハードウェア構成

Computing system	
Type	HP DL580 Gen9
CPU	Intel(R) Xeon(R) Processor E7-8860 v3 2.2 GHz 16core cache L1:512KB, L2:4MB, L3:40MB
memory	512GB
OS	RHEL6.7
compiler	Intel(R) Fortran Compiler V16.0.0
Storage	
Type	Hitachi HUS110
HDD	SAS 1.2TB 10krpm Total 76.8TB
RAID	RAID6 : (8D+1P) × 8RG +4S
Path	Fibre Channel (8Gbps)

3.2 可視化の性能

まず最も単純で基礎的なクエリの1つについて、一般的なRDBMSであるPostgreSQLとの性能比較を行った。この基礎的なクエリとは、あるx軸に垂直なある平面に関する断面を取得するクエリであり、本システムでは3D可視化機能において、図3の立体表面のテクスチャ生成で用いるものである。以下にSQL文例を示す。

```
SELECT x, y, z, m1_z FROM v_data
WHERE exp_id = 2 AND t = 0 AND x = 1;
```

このSQL文は、t=0で表面のm1_zを取得するもので、たとえば断面画像の生成に用いることが想定される。また、同様にy軸、z軸についての断面も生成できる。本報告ではx軸、y軸、z軸それぞれの断面を生成するクエリ、つまりx=0の条件のクエリ1、y=0の条件のクエリ2、z=0の条件のクエリ3の3つについて性能評価を行った。いずれのクエリも得られるレコード数は262,144件である。

本評価では、HADB環境における性能との比較対象として、一般的なRDBMSであるPostgreSQL9.2.15を用いた。計測環境はHADB環境と同一、すなわち表4のComputing systemのハードウェアを用いたが、ストレージとしては表4のStorageはHADB環境に使用していたため、別途用意された表5に示すストレージを用いた。このPostgreSQLで用いたハードウェアのランダムリードの速度については、表4のStorageにくらべて倍程度のアクセス速度が期待されるものである。

表5 PostgreSQL のストレージ

Storage	
Type	Hitachi VSP G800
HDD	SAS 6TB 7.2krpm Total 576TB
RAID	RAID6 : (12D+2P) × 8RG+8S
Path	Fibre Channel (8Gbps)

HADB環境は大規模なデータであっても短時間で並列インポートできる機能があるが、PostgreSQLでは処理時間が少なくとも1週間程度かかると見込まれ評価作業が困難であったので、今回はデータを時刻4ステップ分に限定した。そのためデータの総件数としては格子点数134,217,728の4倍で536,870,912件となる。このデータ数では、インポートはPostgreSQLでも5時間程度、HADBでも15分程度で可能であった。なお各軸のデータに偏りが生じないようにランダムに順序をシャッフルしたCSVファイルを生成してからインポートした。PostgreSQLでのテーブルサイズは141GB、インデックスサイズ30GBであった。各レコードのデータサイズは大きくないため、データサイズはさほどでもないがレコード数は極端に大きい。なお、HADB環境も同様に4ステップに限定したところ、テーブルサイズ182GB、インデックスサイズ29GBとなった。

図5に各クエリの処理に要した時間を示す。クエリ1, 2, 3はそれぞれの空間軸での検索であるが、いずれもほぼ均等な処理性能であった。PostgreSQLでは各検索におよそ800~850秒、HADBでは10~12秒程度で検索を終了した。つまり、PostgreSQLを用いた実装では図3の画面では6面の画像が必要なので、設定を変えてから1時間ほど待たねばならないことになり、実用的ではない。HADB環境であれば待ち時間は1分程度ですむため、多少の不便こそあれ実用に足る性能である。

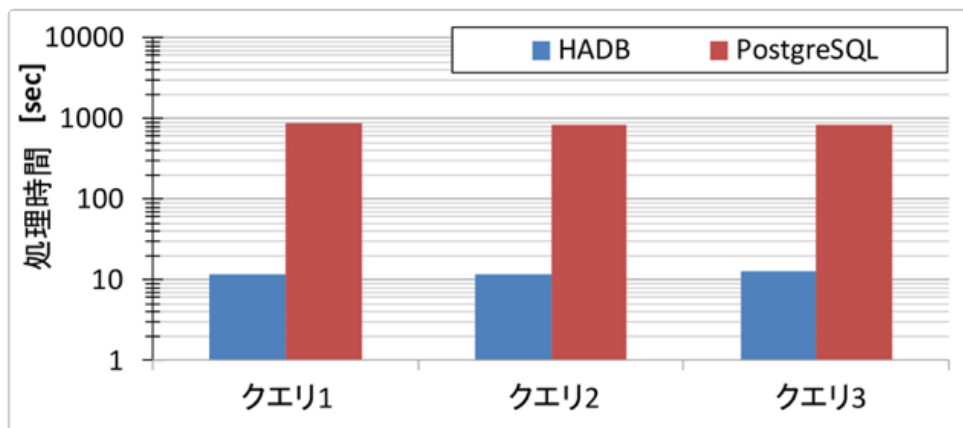


図5 検索処理の時間

3.3 従来集計の代替

次に、本システムがデータの分析に適するかを評価するため、従来、材料研究者が行っている計算をHADB環境上で再現実行し、どの程度の差があるかを性能を評価した。その処理内容の概念図を図6に示す。この処理は、LLGの要素それぞれに付加されている物理量である磁化 m_1 に関連して、各粒子（Grain）ごとの物理量の統計量を求め、時系列的に評価するというものである。今回の計算では、磁化 m_1 のz成分の向きが異なる要素が混じっているか（多磁区）そうでないか（単磁区）をしらべた。結果としては304,152件の出力が得られる。この処理は従来の計算が116ステップすべてのデータを対象にしていたため、HADB環境も116ステップすべてを対象としてインポートした。その結果、インポートにはおよそ7時間程度要し、テーブルサイズは4.0TB、インデックスサイズは6.7TBとかなり大きなものとなった。なお、前述の通りPostgreSQLで116ステップのデータを取り込むのは困難であったので評価は見送った。

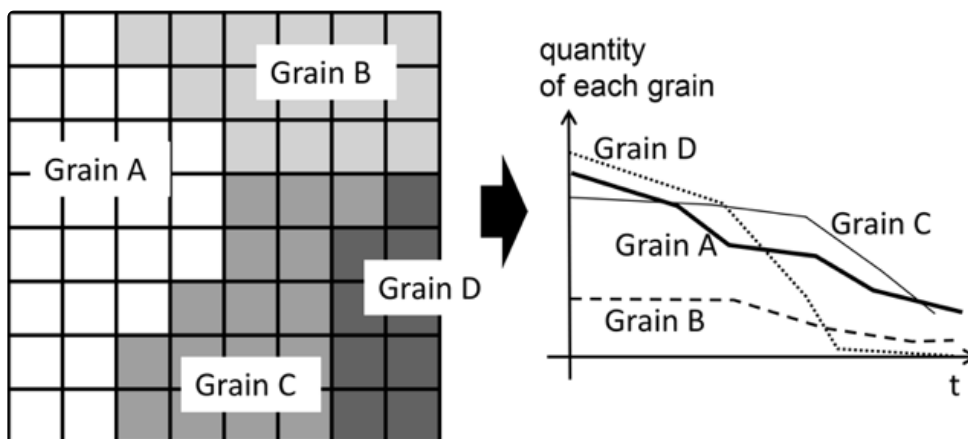


図6 集計処理の概念図

この処理を実行するために、従来は材料科学の研究者がFortranでプログラムを組んで処理を行っていた。そのアルゴリズムを表6に示す。この手順はある時刻に対してのもので、これを時間ステップ分実行し目的の分析結果が得られる。このプログラムはステップ数にして575ステップからなる分析専用のプログラムであり、開発に2日を要したとのことであった。なお表6では省略しているが、データを部分的に読み出して計算して逐次メモリを開放するなど、このプログラムにはデータの規模の大きさに対応するためのメモリ管理などのためのコードを多く含む。これは本評価とは別に、通常材料研究の一環として開発されたものであり、既存のプログラムの流

用等の効率化は図った上での開発期間である。この開発期間は分析内容や研究者のスキルなどによって異なるとは考えられるが、この研究者の場合は、新たに分析を行う際にはまた別のプログラムを数日かけて開発することを目安としており、本例は平均的な開発期間よりやや短いくらいの時間で開発していたとのことであった。

表6 Algorithm1

Algorithm 1: The algorithm of the existing Fortran code

Data: indexfile: grain index for each position, magentfile: magnetization for each position

Result: 3-dimensional array containing + for Single domain and - for Multi domain

```

1 begin
2   indexdata ← read indexfile
3   Prepare 3-dimensional array re_cell initialized by 0
4   magdata ← read magentfile
5   for all position x satisfying re_cell[x]=0 do
6     t_index,t_m_z ← indexdata[x],magdata[x]
7     for all position y satisfying t_index=indexdata[y]
8       do
9         flag←t_m_z×magdata[y]
10        if flag ≤ 0 then
11          End this loop
12        for all position z satisfying t_index=indexdata[z]
13          do
14            re_cell[z] ← flag
15      return re_cell

```

ここでRDBMSを導入して同様の処理をSQLで表現すると、以下のようになる。

```

SELECT grain_id , t,
count ( distinct SIGN ( m1_z ) ) FROM v_data
WHERE exp_id = 2 GROUP BY grain_id , t;

```

この記述はステップ数としては1文であり、簡潔である。作成に要する時間が非常に少なく（おそらく1時間ばかりかからない）、他の分析でも同様にして簡単に作成可能と推定される。これはSQLは宣言型言語であり集計の目的を直接的に記述する言語である点が高い。CやFORTRANで同様の処理を実行するには処理手順を書かねばならず、データの読み書きなどコンピュータの操作を意識せずには実装ができない。材料科学の研究者にとっては、コンピュータを

意識せずにしたい集計処理が直接記述できるほうが馴染みやすく、またメモリ管理等をせずとも動作するため、意識すべき事柄がきわめて少ないのである。反面、従来の専用プログラムに比べると長い処理時間を要し得る。そこで、実際にFortran言語で実際に用いられていたプログラムとHADB環境の性能を比較した。図7にこの処理時間の比較を示す。従来のFortran90を用いて作成された575ステップのプログラムでは、1プロセスで処理を行った場合で処理時間は33時間を要した。Fortranプログラムは、複数プロセスを用いた並列処理によって速度を増加させることができるが、8プロセス並列にした場合で4.8時間、32プロセス並列にした場合で1.6時間、40プロセス並列にした場合で3.9時間となった。このとき1プロセスの処理では7GB、8プロセスの場合で56GB、32プロセスでは224GBのメモリを要しており、40プロセスでは280GBのメモリを要したがメモリが不足してしまいストレージとのスワップが発生して性能低下が起きていた。一方、SQL文1つをHADB環境に入力した結果はおよそ0.9時間を要した。HADB環境は40スレッドが並列動作するように設定されており、使用メモリは260GBに固定されていた。単純な比較では、予想に反しHADB環境は従来よりも早く、プログラム開発の分量も1/575に低減できており、効果的であった。データのインポートに7時間程度要していたことは考慮すべきではあるものの、HADBによって従来よりも早いか同等程度の処理時間で実行が完了できるので、繰り返し行う分析に関しては、効率化の効果は著しく高いといえる。

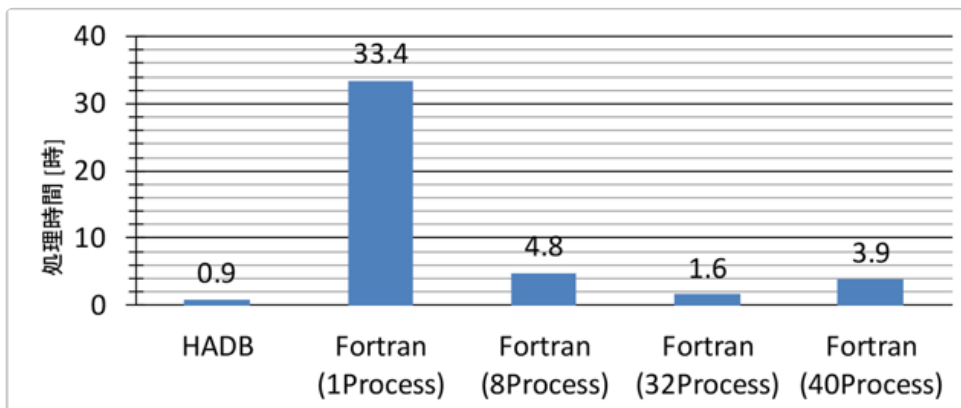


図7 集計処理の時間

3.4 本システム開発におけるプラクティス

3.4.1 多次元配列データベースの開発について

レコード数が極端に多くなり取り扱いが困難な多次元配列データであるが、実際に可視化の性能評価を見るに、少なくともLLGデータでは通常のRDBMSを適用するのは現実的ではなかった。その一方でHADB環境は実用的な性能を達成していた。また、HADB環境は従来の集計の代替実験でも、Fortranで実装された専用プログラムと大差ない時間で実行できていた。

この多次元配列データベースは時間とともに変化する物理量に対する分析処理を対象としているが、表3の多次元配列データには`grain_id`のように時間変化しない量も含まれている。実際には本システムの設計にあたり、当初いわゆるデータベースの正規形を想定して外部キー参照等での実装を検討していた。しかし、RDBMSが内部的に一時表を生成するようなクエリになってしまいがちなテーブルスキーマ設計では、一時表にボトルネックが生じて並列データベースの並列計算が活かせず、処理時間が極度にかかったりメモリ不足を引き起こすような結果になった。大規模な多変量数値シミュレーションは並列計算を想定したデータ構造でデータを扱っており、なるべくそのままの構造を維持するように努めたほうが並列化が効果を発揮しやすいようであっ

た。また、よくある集計演算の結果を別途実行しておくことで実質的な集計を回避する方策をとれば一般的なRDBMSを用いても対応できるのでは、という可能性についても検討したが、対応すべき集計演算が多様すぎて事前計算が困難であった。この点からも、非順序実行原理を用いたRDBMSなどの並列データベースが、MIプラットフォームに適していたといえる。

また、処理の高速さを優先して、RDBMSでない並列データベースを用いる方策も考えられる。RDBMSとした第一の理由としてはSQLが使えることが挙げられるが、他の理由として他システムとの連携のしやすさも挙げられる。RDBMSのインターフェースはODBCやJDBCといった標準的なものがあり、サポートしているソフトウェアは多い。実際、本システムでもOSSのBIツールであるPentahoCE[21]と連携させ動作することができた。また、MIプラットフォーム上でPythonやR向けのODBCドライバを用いることで多次元配列データベースからデータを取得し分析する機能が容易に実装できることが確認できた。ただし、並列データベースはPostgreSQLなどの一般的なRDBMSに比べると成熟度が低く、一部の機能が未サポートであったり、クエリに長時間かかったり、データの更新が想定外に遅かったりして、動作が困難なことも多い。そのため、従来のRDBMSのすべてを並列データベースで置換するような設計は避け、性能上の問題がない場合はPostgreSQL等を用いるようにするほうが、本稿の執筆時点では望ましいものと考えられる。

本例ではLLGシミュレーションの結果を対象としたが、多次元配列のデータ構造は局所的な自然法則（たとえば微分方程式）に関するデータであれば広く適用可能と期待される。たとえば、数十万パターン想定される結晶構造に対応したX線回折輝度分布のデータベースや、第一原理計算による電子状態の波動関数分布のデータベースなどを同様に扱うことが想定できる。その他、空間分布でないものとして、たとえばタンパク質などの分子動力学の計算結果は分子のID、時刻、座標値、保有する物理量などからなり、多次元配列のデータ構造として捉えることができる。このように一般に物理法則は物質は近くにあるものと相互作用が強く、その振る舞いが局所的であるため、それを表すモデルのデータ構造や計測結果および集計、分析処理も同様に局所性をもつことが多い。つまり、物質中のある部分の計算結果は、物質内で遠くに位置する別の部分に影響しにくく、並列計算に適しやすいためである。なお、材料科学とは異なるが、たとえば海洋や気象などの流体力学的な数値計算でもほぼ同様であり、転用できる可能性は高い。

3.4.2 材料開発の加速にむけて

材料科学の解析は、材料科学の知見に基づき探索的に行うことが多い。そのため、従来プログラムの開発に2日程度かかっていたものをわずか1行のコードで実行できたことは非常に大きな意味を持つ。なぜならば、従来方式のプログラム開発はそれが実行可能な人材が必須であるのに対し、HADBを用いた方式では単純なSQLだけ覚えればよく、ほとんどすべての材料科学研究者で実行可能だからである。このことは以下の2つの利点に繋がる。

第一の利点は拘束時間の解消である。従来方式とSQLでは処理時間そのものは大きな差がなくとも、その作業に拘束される時間には天と地ほどの差がある。実際にはプログラム開発ができる材料科学の研究者は希少であり、そのような人材には多数の開発依頼が舞い込み、それに拘束されてしまいがちになる。その場合でも、SQLを使うことで開発時間が短縮できれば実質的な拘束時間を小さくできる。すると、より材料科学の研究開発に注力できる時間が増し、新たな発見に繋がる可能性が高まる。第二の利点は分析の一般化である。従来方式に比べればSQLは実装が容易であるためさまざまな試行が簡単にできる。また、これまでプログラミングスキルがなくな

かなか手を出せなかった分析に対し、SQL文だけの作成であれば実行できる、というケースもあり得る。するとこれまでシミュレーションに現れていたが見落とされていた事実の再発見に繋がる可能性がある。

本システムで採用したSQLは宣言型言語であるため、シンプルにデータの集計操作を記述でき、SQLを利用するユーザはSQL文とその結果のみを意識すればよく、ファイルシステムなどのアーキテクチャを気にする必要がない点で、材料科学の研究者にとって受け入れやすく、学習がしやすい。また並列データベースを用いたことで、日々の測定データ追加により扱うデータ量が増えても対応でき、通常のプログラムでは改変が必要なところ、SQL文は通常ほとんど変更しなくてよいという利点もある。本システムの導入後、実例としてX線回折等のデータベース作製等が行われたが、Pythonプログラムではデータハンドリングのミスが多かったところ、SQLではそのような問題が少なく記述できたとのことであった。一方でシンプルなSQLでも全件処理を要する場合などで計算に数日かかってしまうこともあり、研究者にとって想定外に長時間待たされることに関することもあり、プログラミングの容易さと処理時間の不整合があることは周知しておく必要がある。

一般に、「MIは材料開発の効率化には繋がるが材料科学の新発見に繋げるのは難しい」という言説がある。確かに、MIの分析がそれだけで新発見を生み出すかということそれは困難であろう。ただ上述のように、従来できなかったこと、あるいは、できたとしても作業負荷が大きいことを容易化することが、結果的に新発見を促すことは十分あり得る。いいかえれば、非順序実行原理を用いたRDBMSにより生まれた余裕を、材料科学の研究者が創造的な活動をするために費やせたということである。同様のことがMIプラットフォームの他の機能についてもいえ、本システムが材料科学のさらなる発展に寄与すると期待したい。

4. 関連研究

多次元配列のデータを管理する配列志向のデータベースとしてはSciDB[11]が知られている。SciDBはデータを並列的に格納管理するアプローチをとっており、Rのインタフェースにより高速に統計処理が行える点が強みである。また、rasdaman[9]は地理空間情報を主体にした配列指向のデータベースであり、基盤として他のRDBMSを使うことも可能であり、同じRDBMSを用いて分布ではないデータを管理することもできるという点で、汎用性は高い。ただしこれらは程度の差こそあれいずれも標準的なSQLからは逸脱したインタフェースを使用しなければならない。そのような独特のインタフェースは技術の発展に伴い改廃される可能性が高いため、材料科学の専門家にとって「せっかく学習したのに廃れてしまう」というリスクを感じさせる。それに比べると、本システムは標準的なSQLを使っても十分な性能が得られているのが1つの利点である。

標準的なSQLは関連書籍類が入手しやすく、実績もあり、当面廃れる可能性も薄く、非専門家にとって受け入れられやすいのである。ただし、この点については材料科学研究者のリテラシーや環境の変化にかかる部分もあり、上記の専用システムのインタフェース、たとえばR言語やPythonがデファクト・スタンダードとなるようであれば、将来的にはそちらを採用していくことも考えられる。なお、本システムでは多次元配列データベースを中心に扱ったが、それに類するものとして画像や時系列信号のデータも用いられる。これらも多次元配列であるため、多次元

配列データベースに格納可能である。ただし、これらにはフーリエ変換などの集計演算とは異なる複雑な演算を求められることも多い。その場合、上述のR言語やPythonを用いて処理を行うことになるが、その場合でもデータはRDBMSから取得する形にでき、併用が可能である。

5. おわりに

本稿では、材料開発効率化のために筆者らが開発したWebブラウザを用いた簡易的な可視化分析ができるMIプラットフォームについて示した。本システムの特徴としては多次元配列データベースに非順序実行原理にもとづくRDBMSを採用した点が挙げられる。実験によると、本システムは材料科学の研究者が2日ほどかけて開発した500ステップ以上の従来プログラムと同等の処理がわずか1文で記述でき、実行速度も遜色なかった。これにより材料科学の専門家の手をさほど煩わせずにさまざまな分析ができる。

今後の課題としては、さらにデータが増加した場合や分析が複雑化した場合の高速化や実験データ向けの、計測誤差等を含む不確定性を加味した検索、集計を容易にするためのユーザインタフェース（BIツールを用いた可視化、集計の機能強化）などMIプラットフォームに種々のアナリティクスの機能を追加していくことが考えられる。

謝辞 本研究の一部は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の未来開拓研究プロジェクト「次世代自動車向け高効率モーター用磁性材料技術開発」の支援を受けて行われた。

参考文献

- 1) 寺倉清之：マテリアルズインフォマティクスの現状と将来：趣旨説明，日本物理学会講演概要集，Vol.70, No.1, p.3110 (2015).
- 2) (株)日立製作所：KEK 向けに、磁性材料の磁気構造に関するシミュレーションデータや量子ビーム実験データを解析・可視化するシステムの開発を支援，<http://www.hitachi.co.jp/New/cnews/month/2017/04/0417.html> (2017)
- 3) 浅原彰規，森田秀和，林 秀樹，海野英一郎，小野寛太：マテリアルズ・インフォマティクスのための大規模多次元データベースシステムの提案，情報処理学会研究報告データベースシステム（DBS），Vol.2016-DBS-163, No. 11, pp.1-6 (2016).
- 4) Ono, K., Tsukahara, H., Ishikawa, T., Asahara, A., Morita, H., Hayashi, H. and Umino, E. : Big Data Analytics for Ultra-Large-Scale Micromagnetic Simulation of Permanent magnets, IEEE International Magnetism Conference (INTERMAG), Vol.FQ-14, pp.1-4 (2017).
- 5) 柏野邦夫，黒住隆行，村瀬 洋：ヒストグラム特徴を用いた音や映像の高速AND/OR探索，電子情報通信学会論文誌D, Vol.J83-D2, No.12, pp.2735-2744(2000).
- 6) 合田和生，豊田正史，喜連川優：アウトオブオーダー型データベースエンジンOoODEの試作とその実行挙動，第5回データ工学と情報マネジメントに関するフォーラム,F3-1, pp.1-6 (2013).
- 7) 合田和生，豊田正史，喜連川優：アウトオブオーダー型データベースエンジンOoODE の試作実装と小規模実験環境におけるソフトウェア実行挙動の観測，日本データベース学会論文誌，Vol.12, No.1, pp.25-30(オンライン)，<http://ci.nii.ac.jp/naid/40019725685/> (2013)
- 8) Open Geospatial Consortium : OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0 (10-090r3),
<http://www.opengis.net/standards/netcdf>
- 9) Rasdaman Community Open-Source Project : Rasdaman ,
<http://www.rasdaman.org/>
- 10) Stonebraker, M., Duggan, J., Battle, L. and Papaemmanouil, O. : SciDB DBMS

Research at M.I.T, IEEEData Eng. Bull., Vol.36, No.4, pp.21-30 (2013).

11) Inc., P. : SciDB, <http://scidb.org/>

12) ISO : ISO/IEC CD 9075-15 Information Technology - Database Languages — SQL — Part 15 : Multi Dimensionalarrays,
[http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?
csnumber=67382](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67382)

13) 林 秀樹, 浅原彰規, 菅谷奈津子, 小川祐一, 富田仁志ほか : 災害時の被害推定のための時空間類似シナリオ検索手法, 第8回Webとデータベースに関するフォーラム論文集, Vol.2015, pp.102-109 (2015).

14) Open Source Geospatial Foundation : MapServer,
<https://www.esri.com/products/arcgis/> (2018)

15) ESRI ジャパン (株) : ArcGIS, <https://www.esri.com/products/arcgis/>

16) Cesium Consortium : Cesium, <https://cesiumjs.org/> (2017)

17) Three.js Authors : Threejs.org, <http://threejs.org/> (2017)

18) Apache Spark SQL : <https://spark.apache.org/sql/>

19) 日立製作所 : 高速データアクセス基盤 Hitachi Advanced Data Binder プラットフォーム, <http://www.hitachi.co.jp/products/it/bigdata/platform/data-binder/> (2014)

20) 清水 晃, 茂木和彦, 合田和生, 喜連川優 : 非順序型実行原理に基づく超高速データベースエンジンの詳細分析処理における性能評価, 日立評論, Vol.7, p.8 (2014).

21) Pentaho : <https://sourceforge.net/projects/pentaho/> (2018)

脚注

☆1 Hitachi Advanced Data Binder プラットフォーム[19] は, 内閣府の最先端研究開発支援プログラム「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的サービスの実証・評価」(中心研究者: 喜連川東大教授/国立情報学研究所所長) の成果を利用しています.

浅原 彰規 (正会員) akinori.asahara.bq@hitachi.co.jp

2002年北海道大学理学部物理学卒業. 2004年北海道大学大学院理学研究科物理学専攻修士課程修了. 同年(株)日立製作所入社, 以来, 研究開発グループにて空間情報システムの研究に従事.

森田 秀和 (非会員)

2002年東京大学大学院理学系研究科物理学専攻修士課程修了. 同年(株)日立製作所入社, 以来, 公共システム事業部にて学術情報システムの開発に従事.

海野 英一郎 (非会員)

1999年日立ソフトウェアエンジニアリング(株)(2010年(株)日立ソリューションズに社名変更)入社. 2004年Hitachi Software Engineering France S.A.S.へ赴任. 2007年に帰任. 現在, クロスインダストリソリューション事業部, 空間情報ソリューション本部GIS部所属. 地理情報システムの設計・開発・拡販に従事. 技術士(情報工学部門).

池本 理 (非会員)

2013年東京大学理学部数学科卒業, 同年サクシード(株)入社. 以来, システム開発本部にてエンジニアとして従事.

小野 寛太（非会員）

1996年東京大学大学院理学系研究科博士課程修了。同年東京大学大学院工学系研究科助手，2003年高エネルギー加速器研究機構助教授を経て現職。量子ビームを用いた磁性材料の研究に従事。博士（理学）

投稿受付：2018年3月30日

採録決定：2018年8月2日

編集担当：上條浩一（日本アイ・ビー・エム（株））