

情報視覚化手法「データ宝石箱」のハイパフォーマンス計算技術への応用

伊藤貴之 山口裕美

1. はじめに

情報視覚化とは、形状や空間的構造をもたない情報に、形状を与えて提示することにより、人間による視覚的な理解を支援する技術である。情報視覚化技術が対象とするデータ構造は一般的に、1次元データ、2次元データ、3次元データ、n次元データ、時系列データ、階層型データ、リンクデータ、の7種類に大別される、と言われている。

これらの7種類のデータ構造のうち、階層型データを対象とした情報視覚化手法は、木構造を表現する手法、画面空間を再帰的に分割して表現する手法、透明な3次元物体の入れ子構造で表現する手法、など非常に多く提案されている。著者らの提案している階層型データ視覚化手法「データ宝石箱」は、長方形の2次元的な入れ子構造により、大規模な階層型データ全体を一画面に表現することを目的としている。このような情報視覚化手法は非常に用途が広く、実際に著者らは

- ウェブサイトのアクセス傾向分析[3]
- ネットワーク不正アクセスの監視[4]
- 企業組織コミュニケーションの分析[5]
- 計算生物学シミュレーションのパラメタ最適化
- 創薬用化合物データの分類結果表示
- テキストマイニング結果の視覚化

筆者紹介



伊藤貴之：1968年生。1992年早稲田大学大学院理工学研究科電気工学専攻修士課程修了。同年日本アイ・ピー・エム(株)入社。1997年博士(工学)。現在同社東京基礎研究所勤務。2003年より京都大学情報学研究科 COE 研究員(客員助教授相当)を兼任。CG, CAD, CAE, 情報視覚化, 分散処理システム, ネットワークセキュリティなどの研究に従事。E-mail: itot@computer.org



山口裕美：1977年生。2001年お茶の水大学大学院人間文化研究科数理・情報科学専攻博士前期課程修了。同年日本アイ・ピー・エム(株)入社。現在同社東京基礎研究所勤務。情報視覚化、Webサービスの研究開発に従事。

- ウェブ検索エンジン出力結果の視覚化といった広い目的に適用を進めている。

本稿では、「データ宝石箱」のハイパフォーマンス計算技術への応用例として、分散計算環境の負荷分布監視への適用事例[6]を報告する。本事例では、OGSA (Open Grid Services Architecture)[7]を分散計算環境の一例として、その上で稼働されるジョブ分布を階層型データとして管理し、その遷移を「データ宝石箱」で視覚化した。この視覚化が有効であるアプリケーション例として著者らは、大規模 CAD データに対する四角メッシュ生成[8]を用いた。

2. データ宝石箱

「データ宝石箱」[1][2]による階層型データの視覚化例を図1に示す。「データ宝石箱」は階層型データを構成する葉ノードを色のついたアイコンで、枝ノードを入れ子状に配置された長方形の枠で表現する。「データ宝石箱」は、数千、数万もの葉ノードで構成される大規模な階層型データの全貌を、1枚の画像で表現することを目的としている。階層型データは日常生活にも広く存在するデータであることから、「データ宝石箱」は非常に広い分野の大規模データの視覚化に適用が可能であると考えられる。

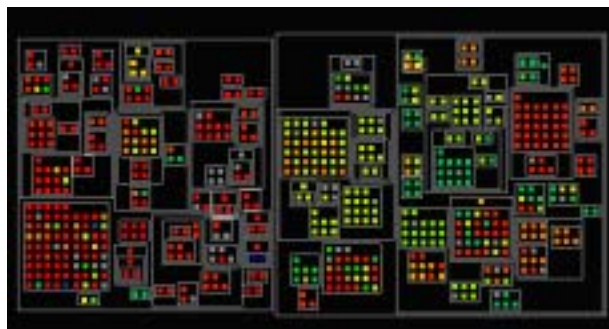


図1 「データ宝石箱」による階層型データの視覚化例。

3. 分散計算環境の負荷分布監視技術の有用性

近年のグリッド計算技術の普及に伴い、分散計算環境における計算機資源(CPU, RAM, HDD など)のユーティリティ(有効活用度)の向上に関する研究がますます活発になっている。著者らは文献[6]にて、OGSA[7]上のジョブ分布を「データ宝石箱」で視覚化する事例を報告している。

OGSA は、グリッド計算技術の事実上の標準実装である Globus Toolkit にも採用されたアーキテクチャである。

OGSA の最大の特徴は、グリッド計算機上のアプリケーションのサービスプログラム化を可能にする、という点である。この特徴は、任意のユーザにアプリケーションを公開可能にするだけでなく、多様な開発言語・OS・ハードウェアが混在するヘテロな分散計算環境において、ユーティリティやインターオペラビリティ(相互運用性)の向上を実現できる。また OGSA は、任意の計算機上でのアプリケーション起動のためのファクトリーまでも、サービスプログラムとして公開可能にする。この特徴により、さらに自在に、さらに容易に、ユーザやプログラムは計算機資源活用を活用できる。

このようなグリッド計算技術の普及により、計算機資源活用の可能性はますます自在になると考えられる。しかしその反面、分散計算環境の管理者が予測不能な計算機資源利用状況が発生する状況も予想できる。この問題を支援する一手段として、分散計算環境にて稼働されるジョブ分布を、情報視覚化技術を用いてリアルタイムに監視し、分散計算環境の負荷分布を迅速に理解することが考えられる。

「データ宝石箱」は以下の特徴を有することから、分散計算環境の負荷分布監視に適した手法であるといえる。

Reactivity: 分散計算環境の負荷分布の変化に瞬時に対応して、最新状況を提示する高速な視覚化手法。

Scalability: 数千～数万ものジョブを運用する大規模分散計算環境の全貌を、一画面に表示する視覚化手法。

Similarity: 時間変化に応じて微小変化する負荷分布を、シームレスに表現する類似性重視の視覚化手法。

Hierarchy: 負荷分布を階層型データとして表現することにより、詳細度制御を可能にする視覚化手法。

4. 「データ宝石箱」による負荷分布監視

本稿が想定する分散計算環境のアーキテクチャ例を、図 2 に示す。このアーキテクチャ例では、まずクライアントユーザがポータルにジョブの実行を要求すると、ポータルはサーバーアプリケーションの親サーバーに要求を転送する。親サーバーは子サーバーにジョブの実行を割り振り、続いて子サーバーから各ジョブの回答を収集する。それと同時に親サーバーは、各ジョブの進捗状況を視覚化サーバーに随時転送する。視覚化サーバーは「データ宝石箱」を用いてジョブ分布を視覚化し、その結果を SVG 形式画像として蓄積する。Web ブラウザから視覚化サーバー上の SVG 形式画像にアクセスすることにより、ユーザは任意の計算機からジョブ分布を視覚化できる。

ここで視覚化サーバーは、各ジョブのアプリケーション名、子サーバー名、クライアントユーザ名、計算時間、そ

の他測定結果、を表形式データとして格納し、親サーバーからのデータ転送のたびに表形式データを更新する。

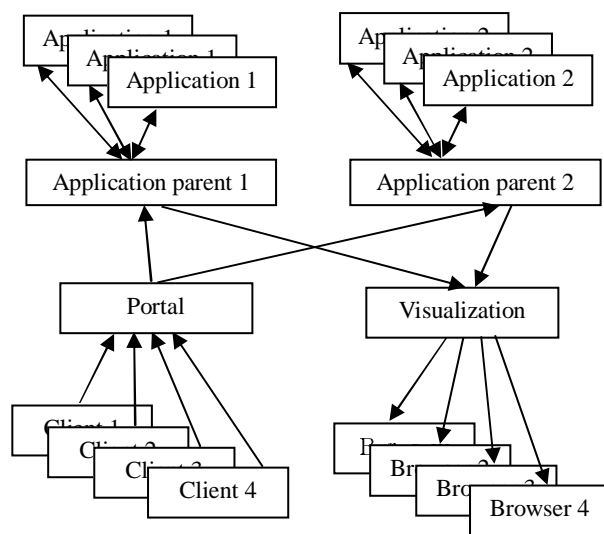
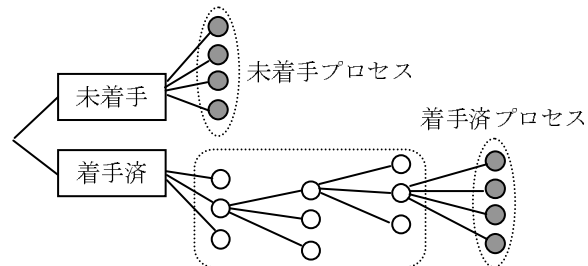


図 2 本稿が想定するアーキテクチャ例。

本手法では、表形式データに記述されたジョブ群を、以下の手順で階層型データ(図 3 参照)に変換する。

1. 未着手ジョブと着手済ジョブに分類する。
2. 着手済ジョブをアプリケーション名、子サーバー名、クライアントユーザ名の 3 指標で順次分類する。この 3 指標の分類順を変えることで、同一の表形式データから異なる階層型データを得ることができる。



アプリケーション名、子サーバー名、クライアントユーザ名、の各指標による 3 段階の分類

図 3 本手法で構築する階層型データの例。

本手法では、図 3 に示す階層型データを「データ宝石箱」を用いて視覚化することにより、ジョブ分布の全貌を一画面に表示する。このとき本手法では、各ジョブに相当する葉ノード(図 3 の黒丸)を棒グラフで、枝ノード(図 3 の白丸)を入れ子状の長方形の枠で表現する。なお本稿で示す視覚化結果では、棒グラフの高さを各ジョブの計算時間から算出する。

以下、ジョブ群の分類順を変えることにより、同一のジ

ジョブ群データから異なる視覚化結果が得られる例を示す。

まず、クライアントユーザ名、アプリケーション名、子サーバー名、の順にジョブ群を分類した結果を図4に示す。この視覚化結果から、多数のサーバーを同時使用しているユーザ、逆に単一の計算機に極端な負荷をかけているユーザ、また計算時間の大きいジョブを多発しているユーザ、などを視覚的に発見することができる。

続いて、アプリケーション名、子サーバー名、クライアントユーザ名、の順にジョブ群を分類した結果を図5に示す。この視覚化結果から、多数のユーザに利用されているアプリケーション、逆に単一のユーザに独占利用されているアプリケーション、また計算時間にばらつきがあるなどの問題点をもつアプリケーション、などを視覚的に発見することができる。

最後に、子サーバー名、クライアントユーザ名、アプリケーション名、の順にジョブ群を分類した結果を図6に示す。この視覚化結果から、負荷の大きい子サーバー、逆に負荷の小さい子サーバー、また多数のユーザに利用されている子サーバーや、多数のソフトウェアを稼動している子サーバー、などを視覚的に発見することができる。

5. アプリケーション例

著者らは、大規模曲面 CAD データに対する四角メッシュ生成を並列化し、そのジョブ分布を「データ宝石箱」を用いて視覚化した。図7に曲面 CAD データおよび四角メッシュ生成結果の例を示す。

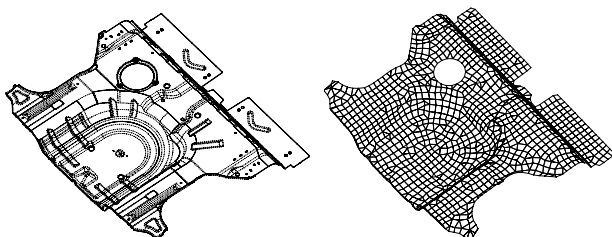


図7 (左) 曲面 CAD データの例。(右) 四角メッシュ生成結果の例。(文献[9]より引用)

著者らは分子動力学モデルを適用した四角メッシュ生成手法[8]を提案している。この四角メッシュ生成手法は、メッシュ生成工程の自動化と高品質化を実現しているが、曲面1個あたり数秒~数十秒程度の計算時間を要する。自動車や航空機などの大規模な工業製品の CAD データは、数千~数万枚もの膨大な数の曲面で構成されている。これらの曲面すべてに四角メッシュ生成手法を適用すると、数万~数十万もの膨大な数のメッシュ要素を生成することが想定される。このような大規模 CAD データに上述の四角メッシュ生成手法を適用すると、数時間~数日の計算時

間を要することが予想される。

ところで、曲面 CAD データに対するメッシュ生成では、隣接曲面間の共有稜線上の節点さえ共有されていれば、個々の曲面内部のメッシュ生成処理は独立に実行することができる。このことから、曲面内部のメッシュ生成処理を並列処理することで、大規模曲面 CAD データのメッシュ生成に要する時間を大幅に短縮することができる。

図8は、80枚の曲面で構成される CAD データを4台の計算機でメッシュ生成したときに、各計算機へのジョブ分布の時間変化を本手法で視覚化した例である。各画像の左側には未着手ジョブが、右側には4台の計算機に割り当てられたジョブが表示される。棒グラフの色は割り当てられた計算機を表し、高さは各ジョブの計算時間を表している。この視覚化結果から、本手法が分散計算環境の負荷分布の時間変化をシームレスに表現していることがわかる。

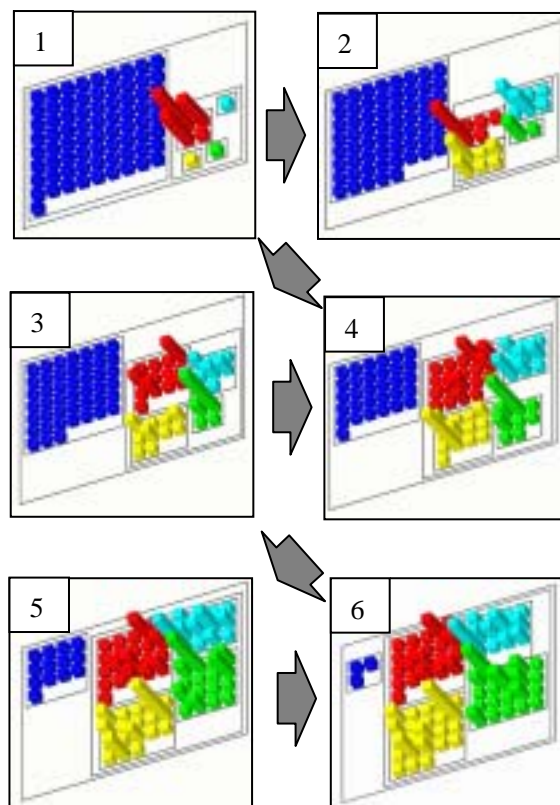


図8 曲面 CAD データのメッシュ生成を並列処理したときの負荷分布の時間変化の表示結果。

6. まとめ

本稿では、情報視覚化手法「データ宝石箱」をハイパフォーマンス計算技術に適用する一例として、分散計算環境の一種である OGSA 上のジョブ群の分布を視覚化する手法を報告した。本手法は分散計算

環境の負荷分布を、ユーザごと、アプリケーションごと、サーバーごと、という多角的な視点から視覚化できるという点で、分散計算環境のユーティリティを監視する目的に適しているといえる。

また本稿ではアプリケーション例として、大規模曲面 CAD データからの四角メッシュ生成技術を示し、このジョブ分布の時間変化を視覚化した例を示した。本手法はジョブ分布のなだらかな時間変化をシームレスに表現できるという点においても、分散計算環境を継続的に監視する目的に適している。

今後の課題として、実際に業務で運用されている分散計算環境にて本手法を運用し、運用上の問題点を発見することがあげられる。

参考文献

- [1] Itoh T., Yamaguchi Y., Ikehata Y., Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 3, pp. 302-313, 2004.
- [2] 山口, 伊藤, 長方形の入れ子構造を用いた階層型データ視覚化手法の拡張, 情報処理学会論文誌, Vol. 44, No. 10, pp. 2469-2477, 2003.
- [3] 山口, 伊藤, 池端, 梶永, 階層型データ視覚化手法「データ宝石箱」とウェブサイトの視覚化, 画像電子学会論文誌 Visual Computing 特集号, Vol. 32, No. 4, pp. 407-417, 2003.
- [4] 伊藤, 高倉, 沢田, 小山田, ネットワーク不正侵入監視のための視覚化の一手法, 情報処理学会第9回分散システム/インターネット運用技術シンポジウム, 2004.
- [5] 中村, 水田, 企業組織をコミュニケーションから評価する, 情報処理, Vol. 45, No. 9, pp. 950-955, 2004.
- [6] Yamaguchi Y., Itoh T., Visualization of Distributed Processes Using "Data Jewelry Box" Algorithm, CG International 2003, pp. 162-169, 2003.
- [7] Foster I., Kesselman C., Mick J. M., Tuecke S., The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, <http://www.globus.org/research/papers/ogsa.pdf>
- [8] 伊藤, 自動四角メッシュ生成手法の検討, シミュレーション学会誌, Vol. 18, No. 2, pp. 19-25, 1999.
- [9] Inoue K., Itoh T., Yamada A., Furuhashi T., Shimada K., Face Clustering of a Large-scale CAD Model for Surface Mesh Generation, Computer-Aided Design, Vol. 33, No. 3, pp. 251-261, 2001.



図4 クライアントユーザ名、アプリケーション名、子サーバー名、の順にジョブ群を分類した視覚化結果。

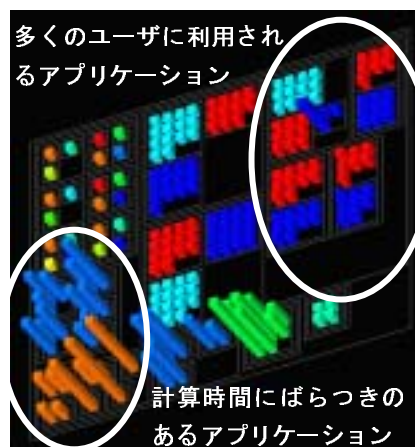


図5 アプリケーション名、子サーバー名、クライアントユーザ名、の順にジョブ群を分類した視覚化結果。



図6 子サーバー名、クライアントユーザ名、アプリケーション名、の順にジョブ群を分類した視覚化結果。