Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

連成・連携計算によるデータ量削減の評価

押川雄大[†1] 小林泰三[†2] 森江善之[†2]
高見利也[†2] 青柳睦[†2]

「京」コンピュータや TSUBAME 2.0 などのように，計算環境は飛躍的に大規模化し，且つ，GPU などの加速機構がついたヘテロジニアスな環境も注目されてきている．このように大規模化している並列計算環境を効率よく利用する計算方法として，計算科学で需要が拡大しつつある連成・連携計算がある．並列計算環境として GPU と CPU を設定し，連成・連携計算の対象として融点付近における金属クラスターのシミュレーションを行った．GPU で生成される一次データを CPU で平行にデータ解析を行うことで，ストレージに書き出すデータ量の削減とその評価をした．

Coupled and Cooparated Simulation for Reduction of Data Amount

Yuta Oshikawa ,[†1] Taizo Kobayashi ,[†2]
Yoshiyuki Morie [†2] and Mutsumi Aoyagi [†2]

Large-scale computing environments like “K” computers and TSUBAME 2.0,are progressing dramatically and being watched heterogeneous with such as GPU. There is coupled and cooparated simulation, which is growing demand for computational science, as a calculation method in order to efficiently utilize large heterogeneous computing environments.

The simulation of the metal cluster in near a melting point was performed as an object of coupled and cooperated calculation. By CPU analyzing the data which is generated by GPU simultaneously, and the reduction and evaluation of data amount which are written out to storage were carried out.

†1 九州大学大学院
Kyushu University graduate school
†2 九州大学情報基盤研究開発センター
Research Institute for Information Technology

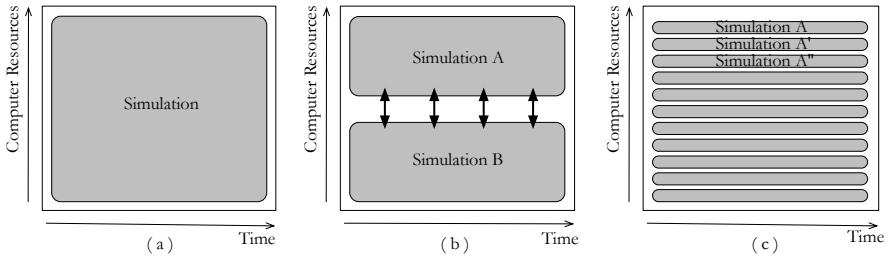

図 1 これまでの並列計算機使用例：
(a) 単独シミュレーション，(b) 連成計算，(c)Embarrassingly Parallel 計算

1. はじめに

近年では計算機の性能向上は CPU コア自体の性能向上よりも，「京」コンピュータに代表されるようにコア数やノード数を増加させる並列計算の方向に向かっている．この傾向は今後も暫く続くと予想され，ハードウェアではコア間やノード間通信の効率的な構造を探る研究がなされ，ソフトウェアではアプリケーションである数値計算の並列化の向上やノード間通信の効率化などの研究が盛んに行われている．また，TSUBAME2.0 の様な GPU を始めとした何らかの加速機構を搭載したヘテロジニアスな構成を持つ計算機の発展も注目され，同様に計算をいかに効率よく並列化するかが専らの研究対象となっている．図 **1** に現状の研究対象になっている計算の構造を示す．計算の構造は巨大な単独ジョブであったり連成計算や Embarrassingly Parallel 型など様々にある． これらの計算タイプの中でも特に注力されているのが単独シミュレーションである．このタイプの計算は，計算環境が大規模化するに伴い，計算全体の計算効率を高める事が困難になりつつある．例えば，並列度の増大が通信コストを相対的に引き上げる状況を作るので，数万ノードを対象にする現場ではジッターレヴェルの精度で通信の効率化をはかる必要が出てきている．また，ハードウェア面でもノード数の増加はシステム全体の平均故障間隔（MTBF）を引き下げるので，問題サイズの大きな単一ジョブは実質的に実実行時間を大きくとれなくなってきている．さらには，問題サイズの増大，或はシミュレーションステップ数の増大はハンドリングするデータ量の増大とセットになる為に，計算そのものの他にもプレ・ポスト処理の負荷が非常に重くなってきている．

このような計算環境を効率よく利用するための新しい計算方法として，連成・連携計算が

Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

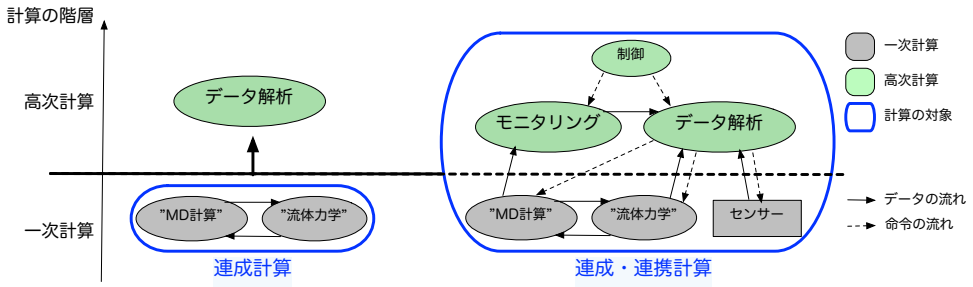

図 2　連成・連携計算による計算対象の拡大

提案されている[1]．連成・連携計算とは，一言で言えば，異なった種類の複数の計算を組み合わせる計算である．これまでの連成計算との違いは，連成計算が数値シミュレーションのみを対象にしているのに対し，連成・連携計算は可視化やデータ解析などのプレ・ポスト処理からセンサーからのデータ入力とセンサーの制御まで，凡そ数値計算を利用する研究ひとセットを丸ごと対象にするところにある．例えば，連成計算は，水溶液中のタンパク質の動きを計算する場合に水溶液部分とタンパク質部分とでそれぞれ異なる計算を行い，かつ互いの計算結果の影響も加味しながら計算を進めるといったものである．この例の場合は，水和タンパク質の構造を求めるものであるので，安定構造を目指して収束させる計算になり，最終的な計算結果がそのまま直接その研究で求められる結果になる．しかし，非平衡緩和過程や化学反応などの過渡現象を研究対象にする場合は，最初の計算で求めるものは系の時系列データであり，研究はその時系列データを解析して進められていく．つまり，連成計算を含めた多くの科学計算には生成されたデータをもとにした解析を行う必要が有る．その場合には，研究を進める為に複数段の数値計算が階層的に必要になる．連成・連携計算が対象にするのはこのような階層的な数値計算全体である．(図 **2**)

ここで術語を定義する．シミュレーションの最初の計算結果を生成する部分を一次計算と呼び，一次計算の結果をもとにして何らかの解析を行う部分を二次計算，二次計算の結果を基にして一次計算やセンサーの制御をしたり，二次計算の結果をさらに解析する部分を三次計算，と順次帰納的に定義する．また，二次計算以降のものをまとめて高次計算と呼ぶことにする．

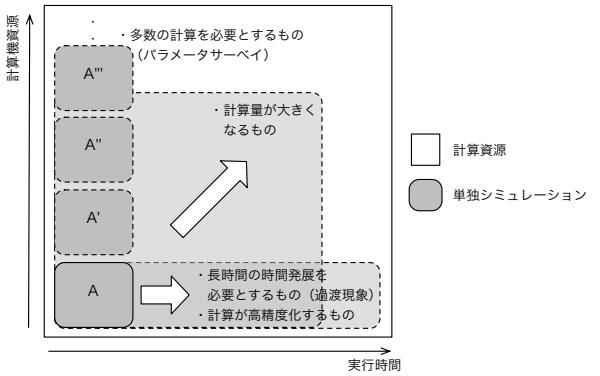

図 3　膨大な一次データを生成する計算の類別

2. データ爆発への応用

前節で整理したように，現状の大規模計算には様々な課題があるが，その中でも本論文ではデータ爆発に連成・連携計算が有効である事を議論する．これまでの研究成果により，大規模な数値計算の実行効率はかなり高まっており，問題サイズの大きな系でも計算時間は短くなってきている．例えば，著者の一人である小林が研究している流体音[2] の計算では，0.05 秒の時系列生成に FUJITSU PRIMERGY RX200S6 72 node を利用した場合で一次計算は 20 時間程度で終了する．しかし問題はその後のポスト処理であり，出力データは gzip 圧縮を施した状態で 3TB を超える．この一次データをポスト処理するのに一週間から十日程の時間がかかる状況である．このように現状では，二次計算以降の計算結果のみを必要とする場合にも全ての一次計算結果を保持しておく必要があり，計算の高精度化，長期化，計算量の増加に伴い膨大な一次データが生成されてしまうといったデータ爆発が問題となってきている．(図 **3**)

さて，具体的な例として一次計算結果を格納する為に必要な記憶容量について考えよう．過渡現象を計算する場合には長時間の時間発展が必要になる．本論文での評価対象に選んだナノメートルサイズの金属微粒子（以下クラスター）の物性を研究する為の分子動力学（MD）計算を例にして，生成データ量を見積もってみる.MD の時間刻みは系の固有振動数の逆数の数十分の一に設定するので，ミリ秒まで計算すると 10^{12} ステップの計算が必要に

Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

なる．クラスターを構成する粒子数を 1000 個に設定したときの全データ量を試算すると，出力として三次元空間における各原子の座標と速度を倍精度にした場合，総データ量は

$8[\text{byte}] \times 1000 \times 6 \times 10^{12} \simeq 50\text{PB}$

になる. 各ステップの時系列データの保存を断念して，1000 ステップ の平均値を保存する事にしても 50 TB の容量が必要になる．しかもこれは一本の時系列のデータ量であり，過渡現象を研究するには，時系列のアンサンブルが必要であり，さらにそれにパラメータサーベイがおぶさってくる. アンサンブルで一桁，パラメータサーベイでは二桁程度の上乗せがされるので，結局 PB のデータ量になる．

2.1 連成・連成計算の適用

これまでに述べた一次計算と二次計算に対して連成・連携計算の枠組みを当てはめることで，一次計算で起こる可能性のあるデータ爆発に対処する．クラスター物性の研究に必要であるのはデータ解析した結果であり，粒子配置の時系列データではない. 二次計算としてのデータ解析の詳細は次節で述べるが，一般論として二次データは抽象化された物理量の時系列である事が多く，解析を終了した一次データは不要になるため，一次計算とデータ解析を同時に行うことにより一次データを最後まで保持する必要性をなくすことができる．これで単純に系の自由度分だけデータ量を削減できる. さらに，解析結果のデータは系にイベントが起こらないと値が変わらないので，gzip などでの圧縮効率が非常に高くなる．

本来は二次計算側からデータ解析をもとにした何らかの命令を渡すことで初めて連成・連携計算になるが，今回はデータ量の削減を目標としているため，二次計算側では一次データを受け取り処理を行うだけにとどめている. 本論文では，一次計算と高次計算を同時に実行する事により，膨大な一次データを外部記憶装置に保存する必要性を排除する事を示し，データ爆発に対する連成・連携計算の有効性を評価する．

3. 実　　装

データ爆発が起こる例として長時間の時間発展を必要とする過渡現象のシミュレーションを考える．扱う問題としては，融点付近におけるナノメートルサイズの金属微粒子クラスターの動きを計算する．ここでは初期配置を図 4 のように設定し，計算を行った．

まず，一次計算としてクラスター内の原子に対して MD 計算を行い，原子配置を更新していく．また，二次計算では一次計算の結果を元にデータ解析を行う．以下に具体的な計算内容と実装環境，処理の流れについて説明する．

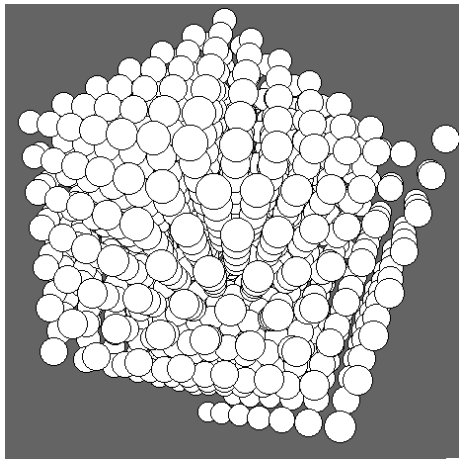

図 4　１０００原子のクラスターの初期配置

3.1 一 次 計 算

まず，シミュレーションの一次計算部分についての詳細を述べる．今回の物理シミュレーションを行う為，始めにある原子 i と他の全ての原子との距離から相互作用を計算し，それを足し合わせることで原子 i に働く力を求める．その際，モデルポテンシャルとして Morse ポテンシャルを用いる．

$$V(r) = \epsilon \left\{ e^{-2\beta(r-\sigma^c)} - 2e^{-\beta(r-\sigma^c)} \right\} \tag{1}$$

ここで σ は平均原子間距離，β はポテンシャルの有効距離，ϵ はポテンシャルの深さを各々決定する係数である．これらのパラメータは銅の値を採用し，値は $\beta = 1.3588[\mathring{A}^{-1}]$, $\epsilon = 0.3429[eV]$, $\sigma = 2.866[\mathring{A}]$ である．

全ての原子についての計算を終えた後，求めた値から各原子の加速度と速度を計算する．MD の時間積分法には二次のシンプレクティック積分法である速度 Verlet 法を用い，$0.01[ps]$ の時間刻みで実行する．

粒子 i が他の粒子から受ける力 $\mathbf{F}_i$ は

$$\mathbf{F}_i(r_{ij}) = \sum_{j=1}^{N} (-\nabla U_{ij}) \tag{2}$$

Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

になる．

ここで，質量 m_i の粒子 i の加速度 $\mathbf{a}_i$ は，

$$\mathbf{a}_i = \frac{\mathbf{F}_i}{m_i} \tag{3}$$

力 $\mathbf{F}_i$ を 3 次元ベクトル方向 x, y, z に分け，それぞれの加速度 a_{xi}, a_{yi}, a_{zi} を求める．x 軸方向では次のようになる．

$$a_{xi}(r_{ij}) = \sum_{j=1}^{N} \left[\frac{2\beta\epsilon x_{ij}}{m_i r_{ij}} \epsilon^{-\beta(r_{ij}-r_{ij}^c)} \{\epsilon^{-\beta(r_{ij}-r_{ij}^c)} - 1\} \right] \tag{4}$$

このようにして求めた $\mathbf{a}_i$ をもとに，時間ステップ t での粒子 i の速度 $\mathbf{v}_i$ と座標 $\mathbf{r}_i$ を $t-1$ ステップでの速度と座標をもとに計算する．

$$\mathbf{v}_i(t) = \mathbf{v}_i(t-\Delta t) + \mathbf{a}_i(t-\Delta t)\Delta t \tag{5}$$

$$\mathbf{r}_i(t) = \mathbf{r}_i(t-\Delta t) + \mathbf{v}_i(t-\Delta t)\Delta t \tag{6}$$

本研究では毎ステップの計算結果を観察するのではなく，少なくともクラスター固有振動より長いタイムスケールで平均化された計算結果についての観察を必要とする．そのため，クラスター固有振動が $0.5[ps]$ であることと 1 ステップが $0.01[ps]$ であることから，１０００ステップ分（$10[ps]$）の計算結果を平均したものについて解析を行うこととする．

3.2 二次計算

二次計算では「クラスター重心と原子間平均距離」,「最近接原子数」,「隣接原子の組み替え頻度数」といった 3 つの物理量を設定し，それぞれの値を求めることとした．次に物理量の詳細を述べる．

3.2.1 クラスター重心と原子間平均距離 $R(t)$

これは原子とクラスターの重心との距離の平均を出すもので，初期配置時にクラスターの外側に位置している原子に対して計算する．以下の式で定義され，原子がクラスター内部へ潜り込む様子を抽出する．

$$R(t) = \frac{1}{N_B} \sum_{i \in outer\ atoms} |\overline{\mathbf{q}}_i(t)| \tag{7}$$

$\mathbf{q}_i(t)$ は i 番目の原子のクラスター重心からの距離である．クラスターの構成原子は格子振動[⋆1]をしているのでその影響を排除するために t から $\Delta t \sim 10[ps] \gg \omega_D^{-1}$ の時間にわたって $\overline{\mathbf{q}}(t) \equiv \int_{t-\Delta t}^{t} \mathbf{q}(t')dt'/\Delta t$ と平滑化している．

3.2.2 最近接原子数 $nB(t)$

ある原子の最近傍にいる原子の数を原子一個あたりに換算した量である．初期配置時にクラスターの外側に位置している原子に対して計算する．

$$n_B(t) = \frac{1}{N_B} \sum_{i \in outer\ atoms} N_{A(i)}(t) \tag{8}$$

ここで N_A はクラスターを構成する原子，N_B はクラスターの外側に位置している原子数であり，$N_{A(i)}$ は N_B に隣接している N_A の個数である．シミュレーション中に原子がクラスターから離脱した場合，$N_{A(i)} = 0$ となる．原子の系への親和度を表現する指標の一つで，ほぼポテンシャルエネルギー変化に追随する．

3.2.3 隣接原子の組み替え頻度数 $S(t)$

これは原子が組み替わる活性度を図る物理量で，最近接原子の組み合わせが替わる度にカウントされる指標である．クラスターの全ての原子に対して計算を行う．この量を時間の観点から観れば系が活性状態か沈静状態かを判断でき，クラスター内部の位置を区切って観ればクラスターのどの部位が活性であるのかが分かる．最近接原子の組の算定には距離の指数が用いられる．距離指数は隣接行列 $A(t)$ から導かれる．隣接行列 $A(t)$ は $N \times N$ の対称行列で，その要素は

$$A_{ij}(t) = \begin{cases} 1: \ \overline{r}(t)_{ij} \leq \sqrt{2}\sigma \\ 0: \ \overline{r}(t)_{ij} > \sqrt{2}\sigma \end{cases} \tag{9}$$

で表される．ここで $\overline{r}(t)_{ij} = |\overline{\mathbf{q}_i}(t) - \overline{\mathbf{q}_j}(t)|$ は i と j 原子間の距離の短時間平均である．i 番目の原子の距離指数 $d_i(t)$ は

⋆1 大体 Debye frequency 程度で $\omega_D^{-1} \sim 0.1ps$ である

Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

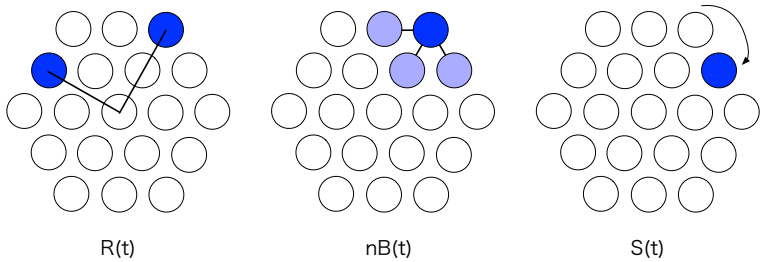

図 5　求める物理量の概念図

$$d_i(t) = \sqrt{\sum_j |A_{ij}(t+\Delta t) - A_{ij}(t)|^2} \tag{10}$$

と定義される．これは，i 番目の原子に関する組み替え自称を量っていることになり，各々の原子の移動度，あるいは活性度を特徴づける．1 原当たりの累積距離指数 $S(t)$ は

$$S(t) = \sum_{n=1}^{t/\Delta t} \left\{ \sum_{i=1}^{N} \frac{d_i(n\Delta t)}{N} \right\} \tag{11}$$

で定義される．

3.3 実装環境

本実装では並列計算環境として GPU を搭載した計算機を設定し，CPU と GPU でそれぞれ計算を実行する．一次計算の特徴として，互いに独立した MD 計算をクラスターを構成する原子数分行う必要が有る．そのため多数の演算コアを搭載し，SIMD 型での命令実行を行う GPU 側に一次計算を担当させ，二次計算以降を CPU 側に担当させる．

実験では CPU に intel Xeon X5650,GPU に NVIDIA Tesla C2050 をそれぞれ使用し，実行環境として CUDA を用いた．

3.4 処理の流れ

具体的な処理内容と流れにつては次のようになる（図 **6**）

(1)　与えられた原子数に対応したクラスターの初期配置を CPU 上で設定し，GPU 側へデータを渡す．

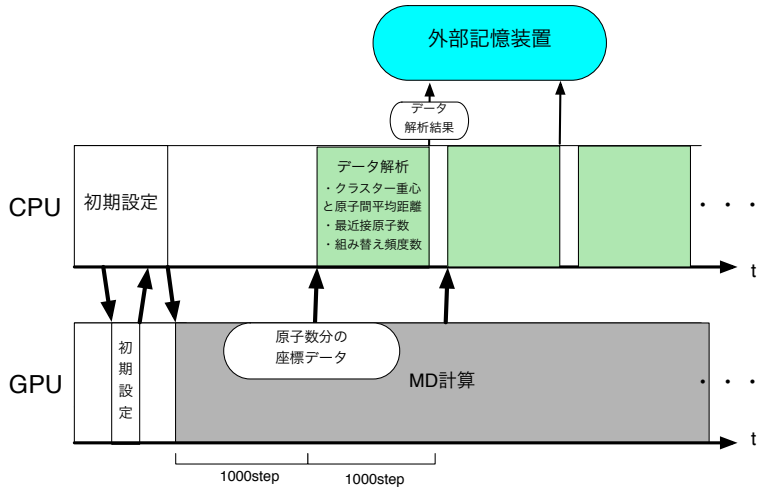

図 6　CPU と GPU での処理の流れ

(2)　次に最急降下法を用いてクラスターを所望のエネルギー値にする．その際の計算は GPU 上で行い，終了時の判断は CPU が行う．

(3)　CPU からの命令を受け，GPU 上で各原子に対する MD 計算を開始する．ここで毎ステップの計算結果を順に足し合わせていき，１０００ステップ分の計算終了後に，合計値を一次データとして CPU 側へ渡す．

(4)　CPU 側で受け取った一次データの平均化と解析を行う．設定した物理量「クラスターの重心と原子間平均距離 $R(t)$」,「最近接原子数 $nB(t)$」,「組み替え頻度数 $S(t)$」を求め，結果を外部記憶装置に書き出す．この間，GPU 側では引き続き次の１０００ステップ分の MD 計算を実行する．

4. 評　　価

ここで本論文の主題である連成・連携計算によるデータ量削減について議論する．今回の実装で行った一次計算を愚直に実行しようした場合，

$8[byte] \times 10^3[bodys] \times 3[dimension] \times 10^9[step] \simeq 21.83[TB]$

ものデータ量を外部記憶装置に書き込む必要がある．ここに連成・連携計算の枠組みを与

Vol.2011-ARC-197 No.4
Vol.2011-HPC-132 No.4
2011/11/28

え，計算の対象を一次計算から高次計算まで拡張することで一次データの生成と解析を平行に行う．そのため，一次データを計算終了まで保持しておく必要がなくなり，必要な記憶容量を最小限に押さえることが出来る．最終的に書き出す３つの物理量のデータの総量は
$8[byte] \times 3[kinds] \times 10^6[step] \simeq 22.89[MB]$
となり，一次データを書き出す場合に比べて大幅な削減が可能になった．

5. 展　　望

本研究では，連成・連携計算の考え方から一次計算で起こる可能性のあるデータ爆発に対処する方法を議論した．現在の仕様では，データの流れは一次計算側からの結果を二次計算側が受け取るだけに留まっており，今後は更に二次計算側での解析結果から一次計算側へ何らかのフィードバックを行う機構を設計していくことで，本来の連成・連携計算が完成する．今回の計算例の中では，長時間のクラスターの動きを観察することを目的とした場合に粒子がクラスターから離脱してしまうと不都合がある．そのため二次計算側で粒子の離脱が解析により確認されたときに，前の結果に離脱が起こらないような処理を加えたものから再び計算を開始するといった事が計算中に可能になる．

この計算方法を発展させるために，既存の優秀なソルバーをできる限り利用した連成・連携計算機構の設計，そのミドルウェアとして実装を目指す．既存のソルバーをまとめる役割としてのメディエイターは，既存の連成・連携計算スキームでは仲介役に留まっているが，ミドルウェア化してメタソルバーの位置づけにすることにより，各ソルバーが計算を続ける系のモニタリングからそのリアルタイムなデータマイニングとソルバーの制御などを行うフレームワークを構築することが重要である．

メタソルバーの役割は，ソルバーが実行する一次計算をモニタリングしてその結果をデータマイニングなどで解析し，必要なデータ意味変換を行った後に適切なソルバーにフィードバックする．それと同時に必要なソルバーを起動したり，不要なソルバーを停止させたりする．メタソルバーを，ソルバーとみなして従える事も可能にする．

このメタソルバーの構造は，サイトを管理する計算機グリッドの管理機構[9]と相似であり，本研究の成果は計算科学だけでなく複数のサイトを跨ぐインタークラウドの管理運用などの計算機科学への貢献も期待される．

参　考　文　献

1) 押川雄大，小林泰三，森江善之，高見利也，青柳睦,「ヘテロジニアスな並列計算環境を応用した連成・連係計算の提案」, SWoPP 2011
2) Yasunori ITO, Taizo KOBAYASHI, Kin'ya TAKAHASHI, Toshiya TAKAMI, Akira NISHIDA, Mutsumi AOYAGI, "Reproduction of Transitions among Notes on an Air-reed Musical Instrument with Compressible LES Combined with Moving Boundary Technique", OSCIC2011
3) Supercomputing Sites:http://www.top500.org/ TOP500
4) CUDA ZONE:http://www.nvidia.co.jp/object/cuda_home_new_jp.html
5) OpenCL:http://www.khronos.org/opencl/
6) 岡崎進，コンピュータシミュレーションの基礎，化学同人，ISBN4-7598-0856-6
7) 清水寧，池田研介，澤田信一，2 元金属クラスターにおける自発的合金化現象とハミルトン系モデル（複雑系５），CiNii.http://ci.nii.ac.jp/naid/110006452093
8) 小林泰三，金属微粒子に於ける原子拡散，ー自発的合金化現象から金属微粒子の普遍的物性解明を目指してー，2002
9) 小林泰三, 天野浩文, 青柳睦, 合田憲人,「大学間連携グリッド基盤の運用」情報処理学会誌 Vol.51, No.2 2010 年 2 月