超並列計算への挑戦
～宇宙物理学的多次元輻射輸送問題における一例～

中本　泰史
筑波大学物理学系／計算物理学研究センター（当時）
東京工業大学理工学研究科（現在）

1. はじまり

私が CP-PACS プロジェクトに参加させていただいたのは，宇宙物理グループの立ち上げとほぼ同時の 1994 年です．当時は実機の製作段階にあり，計算物理学研究センターの新しい建物の工事も始まろうとしていた頃でした．私たちの任務は，大規模数値シミュレーションを用いた宇宙物理学の推進でした．その頃の天文学・宇宙物理学における数値シミュレーションと言えば，圧縮性気体の流体力学(あるいは電磁流体力学)シミュレーションと，重力多体系のシミュレーションが主でした．そのような中，CP-PACS というせっかくの大型計算機を生かしてどんなシミュレーションをやろうか，宇宙物理グループの梅村さんと検討しました．そして得た答えは，多次元の輻射輸送計算でした．

輻射というのは，ひらたく言えば光とか電波とかのことです．天文学・宇宙物理学において輻射は非常に重要な役割を担っています．まず第一に，天文学的観測によって得られる情報のほとんどは輻射によってもたらされています．望遠鏡で天体を観測するというのは，天体からやってくる輻射を望遠鏡で集めて見るということです．ところが日常経験でも分かるように，遠くにあるものの真の姿を，見た目だけで判断することは実は容易ではありません．霧でかすんでいるかもしれないし，手前と奥の前後関係の判定すら怪しいことがあります．遠方のものの実体を見た目から知るためには，実体と見た目との間の関係が理解できればいいのですが，日常生活で私たちはこのことを一瞬のうちに頭の中でシミュレーションしているわけです．ところが宇宙の現象では地上での経験や直感が成り立たないことがあります．したがって観測天体の見た目から実体を理解するために，輻射の伝

播の仕方をシミュレーションすることが必要になるのです. さて, 輻射は宇宙物理学においてもっと物理学的な意義も持っています. 輻射は一般にエネルギーを運びます. 日の光を浴びれば暖かく感じるのは, エネルギーが太陽から輻射によって運ばれてきているからです. 輻射のエネルギー輸送機能により, 天体は冷えたり暖まったりします. 天体の温度というのはその天体の物理的性質を特徴づける極めて重要な意味を持っていますので, その温度を左右する輻射というのもまた, 宇宙物理学において極めて重要な役割を担っているのです. さらに輻射は運動量も運びます. 地上ではこれは完全に無視できますが, 天体現象においては輻射圧が無視できない場合も多数存在するのです. したがって輻射の運動量輸送機能もまた, 宇宙物理学における輻射の重要性を裏付ける根拠の一つとなっています.

さて, このように重要な輻射ですが, それまでの宇宙物理学では2次元・3 次元空間内の輻射輸送シミュレーションはほとんど存在しませんでした. そのような計算が困難だったからです. 困難の最大の原因は, まともに扱うと計算量が膨大になるということでした. しかし, CP-PACS のような大型計算機を用いればこの困難は克服できるかもしれない. 私たちはそう考え, 多次元輻射輸送シミュレーションを CP-PACS を用いた宇宙物理学シミュレーションの柱とすることにしました.

2. どうやって計算しようか

目標を定めたまでは良かったのですが, いざそれを実現しようという段になると, いろいろと問題が出てきました.

私たちが計算しようとしていた問題では, 空間内の全ての点が他の点と輻射を介して相互に関係し合っています. 南の端のある点が明るくなったという影響は, 北の端の点にも及びます. しかもそれは, 途中の伝播の仕方にもよります. 光源に近い上流から下流に向かって順に伝播を計算して初めて, 北の端における南の端の変化の影響を知ることができるのです. さらには, 北端からの輻射が南進して南端にも影響を及ぼします. つまり, 全空間点同士の輻射を介した相互作用を, 上流・下流関係を考慮しながら, 全て計算する必要があるのです. これをCP-PACSという超並列計算機

を使ってどのように計算したらよいのか，効率的な並列計算のアルゴリズムがすぐには思いつきませんでした．

もうひとつの問題は先の問題と関連しますが，輻射輸送計算問題をどのように分割して並列化するかということでした．これは，CP-PACSのメモリサイズとも関係しています．CP-PACSの各PUに付随するローカルメモリでユーザが利用できるのは 40MB 程度でした．扱う問題をこのサイズ以下に分割する必要があったわけです．ところが輻射輸送問題を単純にプログラミングすると，メモリを大量に消費してしまいます．輻射を表す物理量である輻射強度というものが，時間以外に6つの独立変数に依存する量だからです．6つの独立変数というのは，空間の座標(x, y, z)と光子の伝播方向（θ，ϕ），それに光子の振動数νです．これら全ての次元を適当な格子数，たとえば 100 ずつに分割したとすると， 総格子数は $100^6 = 10^{12}$ 個にもなります．これをたとえ 2048 個の PU に分配したとしても，輻射強度という1変数のためだけに必要な 1 PU あたりのメモリは 3.2 GB にもなってしまいます．したがって，何らかの方法によって使用メモリを減らすことが必要でした．

問題は見極められたのですが，その解決策はなかなか見つかりませんでした．ああだこうだと机上テストで悩むうちにどんどん時間が過ぎていきました．状況によっては当初かかげた目標問題を，計算の実行が可能になるまで縮小する必要があるとも考えました．しかし，pilot-3 や CP-PACS の実機を利用して並列計算のテストを始めると，少しずつですが並列計算の特徴が実感できるようになってきて並列計算の考え方に頭が慣れ，並列アルゴリズムも思い描けるようになっていきました．そしてようやく，問題の解決策が見えてきました．私の場合は，対象問題の特徴はこう，CP-PACSの特性はこう，だからアルゴリズムはこうあるべき，という理詰めで解決策を見出したわけではありません．それぞれの特徴を頭の中に入れ，ガチャガチャとかき混ぜていろんな角度から眺めているうちに何となくひらめいたアイディアを拾い上げてみたらうまくいった，という感じでした．

3. 問題の解決

問題となっていたうちの 2 番目の方，すなわち，使用メモリが大きくなっ

てしまうという問題は比較的簡単に解決できました. 私たちが取り組んでいた輻射輸送問題では, 実は輻射強度という物理量そのものを全空間格子点上で保持しておく必要はありませんでした. 必要なのは輻射強度を方向と振動数に関し積分した量だけだったのです. そこでまず, 輻射輸送計算に必要な最小限の輻射強度だけを保持しておき, それ以外のものは各点での方向積分に繰り込んだ後はすぐに捨ててしまうことにしました. これによって使用メモリサイズを, 先の例で言うと100^2分の1に縮小できました. さらには振動数積分も, 輻射強度の絶対値が必要な部分とそれ以外の部分に切り分け, それ以外の部分を予め計算しておいてテーブルとして参照することにより, 積分時に必要な振動数格子点数を 6 (要求精度が低い場合は 3) にまで縮小できる新しい振動数積分法を考案しました(6 点近似法と呼ぶことにしました). これによりさらに使用メモリを減らせたと同時に, うれしい副産物ですが, 振動数積分の精度を上げることもできました.

もう一方の問題, すなわち, 空間の上流・下流関係を保って全空間格子点同士の相互作用を計算するという問題は, もう少しやっかいでした. 私たちの対象はメモリの問題もあり, 並列化の際には方向・振動数次元ではなく, 位置の 3 次元空間を分割して並列化せざるを得ませんでした. その場合, 下流側の空間を担当する PU では上流側の PU での計算の終了を待ってから計算を始める必要があります. さらには, 輻射の伝播方向は上下・左右・前後・斜めすべて(全立体角方向)ですから, 方向毎に上流・下流の関係が変わります. このような状況を超並列計算機で効率よく並列計算するにはどうしたらよいか. なかなか難しい問題でした. 解決策は, 並列計算の際の PU の動作の様子を見ていて思い付きました.

実際の対象は空間 3 次元ですが, 簡単のために 2 次元平面問題で説明しましょう. 計算領域の南西の端にある点から出た北(からほんの少しだけ東)向きの輻射の伝播を計算しようとすると, まず最初に南西端の領域を担当する PU で計算し, その後, それに隣接する北側と東側の PU で計算し, 以下それを北東端に達するまで繰り返す, となります. こうすることによって, 上流・下流の計算順序を守ることができます. 次に同じ点から出るもう少し東向きの輻射, 例えば北北東向きの輻射の伝播を計算しようとする

と，PU の動作は北向きの場合と全く同じになることがわかります．つまり，最初のステップでは南西端担当の PU が動作し，次のステップではそれに接する北隣と東隣の PU が動作するわけです．物理的には北向きの輻射と北北東向きの輻射は別物なのですが，空間分割された領域が割り付けられた PU 空間内での稼働 PU 群に色づけしたら，その色の動きは全く同じになるのです．調べてみると，2 次元平面問題の場合には稼働 PU 群の動作パターンは 8 つに，3 次元問題では 24 になることがわかりました．私たちは対象の 3 次元問題を扱うために物理的方向を 100^2 もの格子に分けていますが，PU の動作パターンはわずか 24 しかないのです．ですから，400 〜 500 ($\fallingdotseq 100^2/24$)もの方向格子が同じ PU 動作パターンで並列計算を行うのです．このことに気づいたことが，問題解決の糸口になりました．

再び 2 次元平面問題で説明します．北向き(よりほんの少し東向き)の輻射の伝播を計算する場合，まず最初のステップで南西端の PU での計算のみを行い，その計算終了後，その北隣と東隣の PU にデータが送られ，2 ステップ目でそれら 2 つの PU での計算が行われる，というふうに計算が進みます．このとき，1 ステップ目の稼働 PU 数は 1，2 ステップ目でのそれは 2 です．3 ステップ目では，さらにそれぞれの北隣と東隣の PU での計算が行われ，このときの稼働 PU 数は 3 となります．南北，東西，各方向に並べられた PU 数を N_{PU} とすると(この場合，全 PU 数は N_{PU}^2 です)，北向き輻射の計算は 2 N_{PU} − 1 ステップで北東端に達し完了します．ある瞬間に計算をしている PU 群を PU 空間内で色づけしたとすると，北西から南東に向かう 1 本の線上に並ぶことが分かります．そして計算ステップを進めるとこの線が南西から北東に伝播し，それによって上流から下流に向かう計算順序を守りつつ全空間での計算が行われることになります．

しかし，このままでは並列化率が極めて低いことがわかります．ある瞬間における稼働 PU 数の全 PU 数に対する割合は，最大でも $1/N_{PU}$ 程度にしかなりません．1 ステップ目などではわずかに $1/N_{PU}^2$ です．しかし実は，先に述べたように私たちの問題ではたくさんの方向格子に対する計算が同じ PU 動作パターンでできるので，このことを用いて稼働 PU 数を増やすことができます．北向き方向の計算の 2 ステップ目の時，それより少し東向きの

別の方向の計算を南西端の PU で始めればよいのです. この方向は北向きとは別の方向ですが, PU 空間内における稼働 PU 群のパターンは北向きの場合と同じなので, このようなことができるのです. さらには, 最初の北向き方向の計算が 3 ステップ目に入ったとき, 2 つ目の方向の計算を 2 ステップ目に進め, さらに 3 つ目の方向の計算の 1 ステップ目を南西端 PU で始めるのです. これを繰り返すと, 最初の計算を始めてから 2 N_{PU} − 1 ステップ目では北東端の PU で北向き方向の計算をしていることになりますが, その南隣と西隣の 2 つの PU では 2 つめの方向を, それらのさらに南隣と西隣の 3 つの PU では 3 つ目の方向を計算しています. そしてそのとき南西端の PU では 2 N_{PU} − 1 番目の方向の計算を行っています. つまり, 2 N_{PU} − 1 個の方向が同時に計算されている状態になるのです. ここで, 2 N_{PU} − 1 の値はせいぜい 80 程度(3 次元の場合は 3 N_{PU} − 2 となって, せいぜい 40 程度)であり, 先に求めた PU の 1 動作パターンあたりの方向格子数(400 〜 500)に比べればずっと小さいということに注意すると, これからしばらくの間はそれぞれの PU は別々の方向を計算しながらも, 全ての PU が稼働し続けることがわかります. こうして長い時間にわたって全 PU が稼働する状態が続くため, 問題全体として高い並列化率が得られました.

2 次元平面問題の場合, ある瞬間にある方向を計算している PU 群を PU 空間内で色づけしたとすると, 2 次元 PU 空間全体は全部で 2 N_{PU} − 1 本の線に色分けされて埋め尽くされます. 3 次元問題の場合は PU 空間も 3 次元となり, 同じ方向を計算している PU 群は平面を形成します. そして 2 次元の場合と同様に, 3 N_{PU} − 2 つの平面で 3 次元 PU 空間を埋め尽くします. 計算ステップを進めると, この面が順に下流に伝播していきます. この計算法では PU 空間を複数(Multiple)の波面(Wave Front)が伝播しつつ埋め尽くすことによって, 上流・下流の計算順序を守りつつ, 高い並列化率が得られています. そこで私はこの計算法を, Multiple Wave Front 法と呼ぶことにしました. PU 空間内を 1 つの波面を伝播させることによって上流・下流の計算順序問題は解決できたわけですが, さらに複数の波面を同時に存在させて PU 空間を埋め尽くすことができることに気づいたことで, 並列化効率を上げることができたわけです.

余談になりますがMultiple Wave Front法の原理を思いついたあと，プログラミングして実装し，テストを行って有効性を確認するのに，ある年の夏まるまる1ヶ月をかけました．このとき私はまだ独身でしたが，ほとんど家に帰らず研究室で寝泊まりし，根を詰めて作業していました．プログラムが物理の実問題に応用できるようになったのはさらに数ヶ月後でしたが，プログラム開発にとってはあの夏休みの集中的作業が最も重要であり，鍵になったと思います．その後，私自身はこのようなやり方でのプログラム開発は，体力的，精神的，社会的な理由により，なかなかできなくなりました．

4. その後

なんとか問題が解決し，当初の目標であった3次元輻射輸送シミュレーションを実行することができました．CP-PACSを利用した結果，シミュレーションの規模は最大 $128^3 \times 128^2 \times 6$ 格子にもなりました．この数字は自分たちで言うのもなんですが，驚異的な値です．宇宙物理分野で輻射輸送シミュレーションを行っている人に言うと，今でもびっくりされます．この計算の規模は，世界の数年先を行っていたと自負しています．

さて，振動数積分に関する6点近似法も並列化効率を上げるためのMultiple Wave Front法も，CP-PACSを有効に使って大規模計算をしようという必要に迫られて考え出されたものです．（またまた余談ですが，宇宙物理グループの人達は格子QCDグループの人々と違い，計算機を効率良く使い尽くそうという厳しさがあまりありませんでした．実際，計算機を少し無駄に使っていて，お叱りを受けたことがありました．そういうことも，私たちの新アルゴリズム開発の強い動機になっていたように思います．）しかし，ここで考案した手法はCP-PACS以外の計算機にも応用することができました．また，宇宙物理の対象問題が変わっても類似の多くの問題に応用できます．実際，その後現在に至るまで，利用計算機が変わっても私たちはこれらの手法を利用し続けています．さらには，外国のグループもこれらの手法を取り入れるようになってきました．このような状況を見るにつけ私は，並列プログラミングと格闘したあの夏のことを思い出しつつ，ちょっとした喜びに浸っています．一方で，あのアルゴリズムはもっと改良できそうだなあ，

改良してもっといろんな問題に応用したいなあ，とも考えています．実際には，なかなか思うようには動き出せないのですが．

5. 御礼

最後になりましたが，計算物理学研究センターでの宇宙物理シミュレーションの機会を私に与えていただき，陰に陽にご支援をいただいた，岩崎先生をはじめとする CP-PACS プロジェクトの皆様，日立製作所等関係企業の皆様，計算物理学研究センターの皆様，筑波大学の関係者の皆様，天文学関係者の皆様，そして宇宙物理グループの梅村雅之先生と須佐元氏(現 甲南大)に，この場をお借りして心よりの謝意を表したいと思います．ありがとうございました．