Heterogeneous Multi-Computer System における重力効果を含む宇宙輻射流体計算

朴 泰祐[†1] 牧野 淳一郎[†2] 須佐 元[†3]
梅村 雅之[†4] 福重 俊幸[†5] 宇川 彰[†4]

Heterogeneous Multi-Computer System（HMCS）は，マルチスケールな計算物理シミュレーションを行うために開発された，連続体向け超並列計算機と多粒子系向け超並列計算機を融合した新しい計算機システムである．我々は HMCS のプラットフォームとして，科学技術計算用超並列計算機 CP-PACS と，重力計算専用計算機 GRAPE-6 を，並列コモディティネットワークによって結合したシステムを開発した．HMCS では，GRAPE-6 で行われるミクロスコピックな重力多体計算と，CP-PACS で行われるマクロスコピックな輻射流体計算を並行して行い，非常に詳細な計算宇宙物理シミュレーションを実現する．両システムは PIO（Parallel I/O System）と呼ばれる，コモディティネットワークを用いた並列入出力システムで結合され，各計算タイムステップごとに，計算対象となる全粒子データを交換し合う．本論文では，HMCS の全体構想・代表的アプリケーション・実装・プロトタイプシステムの性能評価に関して述べる．

Space Radiative Transfer and Hydrodynamics Calculation with Self-gravity on Heterogeneous Multi-Computer System

Taisuke Boku,[†1] Junichiro Makino,[†2] Hajime Susa,[†4]
Masayuki Umemura,[†4] Toshiyuki Fukushige[†5]
and Akira Ukawa[†4]

HMCS (Heterogeneous Multi-Computer System) is a new parallel processing platform combining massively parallel processors for continuum simulation and particle simulation to realize multi-scale computational physics simulations. We constructed a prototype system of HMCS with a general purpose scientific parallel processor CP-PACS and a gravity calculation parallel processor GRAPE-6 connecting them via commodity-base parallel network. On the prototype of HMCS, a micro-scopic gravity calculation on GRAPE-6 and a macro-scopic radiative-transfer hydrodynamics calculation on CP-PACS are performed simultaneously to realize detailed simulation on computational astrophysics. Both systems are connected via parallel network controlling system named PIO (Parallel I/O System). On each time step, all data of particles are exchanged between two systems hiding communication latency with a special algorithm and buffering effect by PIO. In this paper, we describe the overall concept of HMCS, its application and system implementation, and the performance evaluation of a prototype system.

†1 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba
†2 東京大学理学系研究科天文学専攻
Department of Astronomy, Graduate School of Science, The University of Tokyo
†3 立教大学理学部物理学科
Department of Physics, College of Science, Rikkyo University
†4 筑波大学物理学系
Institute of Physics, University of Tsukuba
†5 東京大学総合文化研究科広域科学専攻
Multi-Disciplinary Sciences, Graduate School of Arts and Sciences, The University of Tokyo

1. はじめに

計算物理学における大規模シミュレーションには，並列ベクトル計算機・超並列型スカラー計算機・専用並列計算機等，様々な HPC 向け並列計算プラットフォームが用いられる．従来，これらのプラットフォームは単体で用いられることがほとんどであり，最近になって，Grid 環境下でスーパーコンピュータ間の接続を行おうという機運も生じているが，実用研究としてはまだ時間がかかる．このような状況下では，当然，対象となる計算もそのプラットフォームに合わせざるをえなくなり，より広汎な物理計算を行う際に実効性能・

問題規模等の点で制約条件とならざるをえない．

このような背景の下，我々は次世代の大規模シミュレーションプラットフォームとして HMCS（Heterogeneous Multi-Computer System）を提案する．HMCS は異なるアーキテクチャを持つ（並列）計算環境の融合体の総称であり，たとえば超並列計算機と専用並列計算機を高速・大容量ネットワークによって有機的に結合し，互いを相補的に用いることにより，従来実現不可能であった新しいパラダイムによる物理シミュレーションを可能とするものである．このようなシステムを実現するにあたり，ボトルネックとなるのは，いうまでもなく複数のシステム間を結合するネットワークである．現在，コモディティネットワークの性能は日々増大しつつあり，Gbps クラスのネットワークが PC にも導入できる時代になっている．しかし，実際の超並列システム等にこれを用いる場合，入出力プロセッサにおける処理の集中・絶対的バンド幅の不足等の問題が生じる．これに加え，1 世代前のプラットフォームにおいてこれらの最先端ネットワークインタフェースが必ずしも利用できない等の問題もある．

我々は，これらの問題を解決する手法として，筑波大学において研究開発された，コモディティネットワークに基づく並列入出力システム PIO（Parallel I/O System）を用い，超並列計算機と重力多体問題専用並列計算機を結合した HMCS プラットフォームを開発した．本論文ではこの HMCS のコンセプト，アプリケーション，システムの概要と実装，および性能評価について述べる．

2. HMCS プロトタイプ

2.1 ヘテロな計算プラットフォームの必要性

宇宙には，星や銀河，銀河団等様々な階層の構造が存在する．しかし，それらを支配する基礎物理過程の観点から見ると，流体過程，重力過程，輻射過程の 3 つに集約される．しかし，これらをすべて採り入れた自己重力輻射流体力学の実現はこれまで困難であった．たとえば，銀河形成過程は，紫外線輻射で満たされた宇宙の中で起こったことが観測により分かっているが，その物理過程はまだ解明されていない．この問題では，銀河の元になるバリオン物質，重力場を支配するダークマター，そして紫外線輻射輸送を同時に取り扱う必要がある[1),2)]．

このような複雑な計算を従来の超並列計算機またはベクトル並列機，あるいはクラスタ等の均質（homogeneous）な単体計算機で処理することは，実効性能の点できわめて難しい．特に，重力過程に関しては粒子数 N に対し，その計算コストが N^2 のオーダになる点で，（超並列機を含む）汎用計算機システムでも，ある程度以上の N についてはこの部分が計算を支配してしまい，大規模化が難しい．

しかし，重力計算部分のみに関しては，現実的な解として，専用計算機による高速処理が可能である．東京大学で研究開発が続けられている GRAPE はその代表であり，最新バージョンの GRAPE-6[3)] はプロセッサボードあたり約 1 TFLOPS の実効性能で重力多体問題の処理を行うことが可能である．この GRAPE-6 システムを，柔軟で複雑な処理を行うことができる汎用超並列計算機と結合し，高速だが柔軟性に欠ける GRAPE-6 と，それに比べ低速だが柔軟性に優れている汎用計算機との融合により，上述のような問題に対応可能なシステムが実現できる．

我々は，このような heterogeneous なシステムによって，複数の過程を含む詳細な計算物理シミュレーションを行うことを目指し，HMCS の開発を行っている．HMCS の計算能力を高め，数百万体に対する計算が可能となれば，宇宙における様々な階層の天体形成史の解明に大きな進展をもたらすことが期待される．さらに，heterogeneous なシステムのより理想的な構成を探り，適用可能な分野を広げることも本研究の重要なテーマである．

2.2 HMCS プロトタイプのシステム概要

我々が開発中の HMCS プラットフォーム（以下，便宜上，単に HMCS と呼ぶ）では，汎用超並列計算機 CP-PACS[4)] と，重力計算専用並列処理システム GRAPE-6[3)] を用いる．CP-PACS は 2,048 台の計算専用プロセッサ（PU: Processing Unit）と，128 台の入出力プロセッサ（IOU: I/O Unit）を持つ．また，GRAPE-6 システムは 32 チップの重力計算専用プロセッサを 1 つのボードに実装したものであり，これをさらに複数用いることにより，プロセッサ数に比例した超高性能な重力計算エンジンが提供される．

CP-PACS には 128 台の IOU のうち 16 台に対し，100 base-TX イーサネットのインタフェース（NIC）が実装されている．このほかに汎用インタフェースとしての 10 base-T イーサネットと，フロントエンドコンピュータとの間の大容量データ通信のための HIPPI（ピーク性能 100 MB/s）も別途用意されているが，今回の HMCS 設計では後述の GRAPE-6 ホストシステムとの並列接続によりネットワークバンド幅を確保をするため，16 本の 100 base-TX イーサネットを用いる．2,048 台のすべての PU は，どの IOU を経由して

も通信を行える．

GRAPE-6ボードそのものはスタンドアロンタイプの計算機システムではなく，インタフェースを通じてホスト計算機と接続し，制御される．現在のインタフェースは32 bit PCIであり，Alpha系およびPentium系のプロセッサを持つPCと接続可能である．GRAPE-6のPCIインタフェースには制御用プロセッサが実装されており，ホストからはPCIバスを通じて命令およびデータのやりとりを行う．GRAPE-6ではN個の粒子に対する N^2 の総当たりの重力計算を行う．このため，まずN個の全粒子データ（質量・位置・速度・前ステップにおける加速度等）をGRAPE-6上の高速メモリに格納しておき（これを j 粒子と呼ぶ），その後でN個の粒子データを改めて内部のパイプライン演算器のレジスタにセットして一気に処理する．GRAPE-6チップでは，1回のパイプライン演算でセットできる粒子（これを i 粒子と呼ぶ）数は48個である．したがって，i 粒子に関してはN/48回の演算を行うことになる．

GRAPE-6ボードを多数用いれば，i 粒子を分散させることにより，処理性能がボード数に比例して向上する．GRAPE-6では，複数のボードの入出力を束ねるツリー型通信ボードも別途用意されているが，HMCSでは以下の2つの理由によりこれを用いない．まず第1に，ホスト計算機にあまり多数の粒子を集中させると，GRAPE-6の演算性能に対し，ホスト上での汎用処理および通信処理を行う際にこちらがボトルネックとなってしまう．第2に，CP-PACSとの通信において，並列ネットワークを用いたバンド幅向上を狙う都合上，ホスト計算機も並列構成になっていた方が良い．以上の理由により，HMCSではGRAPE-6のホスト計算機を，複数のPCによるクラスタ構成とし，ノードPCがGRAPE-6ボード1枚ずつを担当し，必要に応じてノード間通信も行いながら，重力計算部を構成する．

図1にHMCSの全体構成を示す．現在の構成では，8枚のGRAPE-6ボードを用いるため，クラスタのノードも8台になっている．CP-PACSからの16本の100 base-TXイーサネットリンクと，クラスタの8本のそれは，2台のSwitching Hubを介して接続される．本システムで用いるPIO（Parallel I/O System）は，ネットワークが複数のHubを用いたサブネット構成になっている場合でも，静的および動的負荷分散機能を用い，これに対応するように設計されている[5]．本システムの場合，総イーサネットリンク数は24本となるため，1台のHubでも収容可能ではあるが，これ以外に周辺機器として，ファイルサーバであるSGI Origin-2000（8プロセッサ，100 base-TXイーサネット8本）およびグラフィックサーバであるSGI Onyx2（4プロセッサ，100 base-TXイーサネット4本）も収容しなければならないため，実際のリンク数は30本以上になる．よって，PIOのサブネット機構を積極的に利用し，総バンド幅を広くするためにこのような構成をとった．

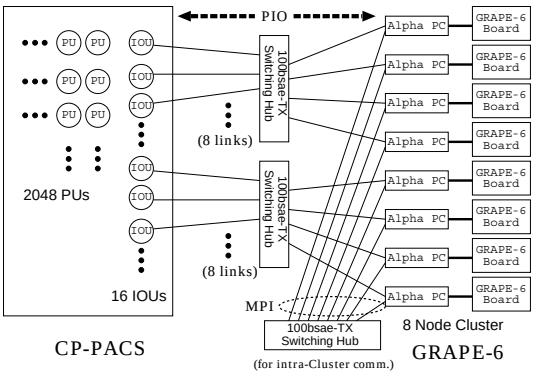

図1 HMCSの全体構成図
Fig. 1 Block diagram of HMCS.

3. 基本アルゴリズムと性能

3.1 対象問題と基本アルゴリズム

GRAPE-6を用いたHMCSでは，重力多体計算をベースにした様々な物理シミュレーションを扱える．現在，我々は重力効果を採り入れた，SPH（Smoothed Particle Hydrodynamics）に基づく宇宙輻射流体計算[6]を最初のアプリケーションとして考えている．

地上における流体力学と宇宙における流体力学の大きな違いは，重力の振る舞い方にある．地上では地球が作る重力場を外場として扱い流体計算を行えばよいが，宇宙流体は流体の各部分に重力が働く“自己重力系”であり，流体の運動に従って重力場も変化する．自己重力系の重力場の変化は，ポアソン方程式によって記述されるが，coarse-grainedな系として扱えば，N体粒子系の重力で置きかえることが可能である．SPHは，連続体である流体を広がりを持った流体粒子でcoarse-grained化し，流体粒子の重ね合わせで流体を表現する方法である．よって，SPHの自己重力は，N体系として計算することができる．しかしながら，N体系の精度の良い重力計算は，N^2 の計算コストがかかる．このため，銀河形成等の計算に必要な数10万流体粒子系の計算では，汎用スーパーコンピュータを用いてもこれまでは近似的な自己重力の取扱いしかできなかった．

HMCS では，流体部分を汎用スーパーコンピュータで，自己重力部分を GRAPE-6 で分担して計算することにより，これまで実現しえなかった高精度の自己重力流体計算を可能にする．もう 1 つこれまで近似的にしか扱われてこなかった要素に “輻射” がある．輻射は，宇宙現象においてエネルギー収支を司る主要な物理過程である．これまでの SPH 計算では，輻射はいったん放射されれば速やかに系から脱出するという仮定や，ほとんど系の中に閉じ込められて拡散的に伝播していくといった仮定が用いられてきた．しかし，本来輻射はその輸送過程を記述する “輻射輸送方程式” によって支配されており，上記の仮定はしばしば本質的に誤った結果を生み出す．

後述するように，輻射輸送方程式は基本的に光子のボルツマン方程式であり，その高い次元数のため正確な取扱いは長い間困難とされてきた．我々は，輻射輸送方程式を SPH の枠組みの中で高速かつ正確に解き上げるアルゴリズムを開発し，これを CP-PACS に実装することに成功した．これによって，宇宙におけるエネルギー収支を正確に取り扱うことのできる輻射流体計算を可能にした．HMCS は，宇宙現象の解明に不可欠な自己重力輻射流体力学を可能にする現在唯一のプラットホームである．

流体計算に SPH 法を用いる大きな利点は，ガスの運動に応じて流体粒子の大きさを変化させることにより，3 次元のラグランジ流体計算を可能にするという点である．自己重力が働く宇宙現象では，密度が何桁にもわたって劇的に変化する現象がしばしば起こる．このような大きなダイナミックレンジの現象は，空間に一様に張り巡らした格子による流体計算では扱うことができない．格子の大きさを動的に変化させていく適合格子法等も提案されているが，アルゴリズムが極端に複雑化しているばかりでなく，並列化効率はきわめて低い．SPH 法は，アルゴリズムの柔軟性に富むばかりでなく，高い並列化効率を実現でき，大きなダイナミックレンジを扱える．これらの点で，SPH は大規模な宇宙流体力学計算にきわめて適した方法であるといえる．

また，銀河形成論において無視できない重要な存在としてダークマターがある．ダークマター自体は上述の SPH 法における通常物質を表す粒子（これを SPH 粒子と呼ぶ）に対し，流体計算上の影響は与えないが，重力計算には大きな影響をもたらす．我々は冷たいダークマターと宇宙項を含む宇宙モデルを想定し，ダークマターは弱い相互作用の無衝突粒子として扱うことにした．したがって，ダークマターは GRAPE-6 上での計算にのみ用いられ，GRAPE-6 には SPH 粒子とダークマター粒子のすべてが渡されることにより重力計算が行われる．現在の我々のモデルでは，総粒子数の半分が SPH 粒子であり，残り半分がダークマターとなっている．

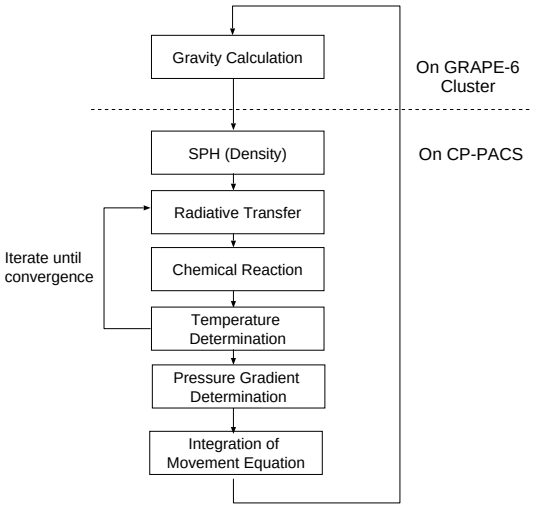

図 2 重力効果を含む SPH の基本アルゴリズム
Fig. 2 Basic algorithm of SPH with self-gravity.

HMCS における基本的計算アルゴリズムの概略を**図 2** に示す．詳細は以下のとおりである．

(1) 全粒子の位置・速度，場の温度・密度・化学種の量を初期条件として与える．
(2) CP-PACS から GRAPE-6 に粒子データ（SPH 粒子，ダークマター）を送り，自己重力を評価する．
(3) SPH により密度・圧力・圧力勾配を計算する．
(4) 速度・力から時間刻み幅を決定する．
(5) SPH 粒子の大きさを変更する．
(6) 輻射輸送計算・化学反応計算・SPH エネルギー方程式の計算を，温度・化学種が収束するまで繰り返す．必要によってタイムステップをさらに短く調整する．
(7) SPH 運動方程式により粒子の運動を積分する．
(8) ステップ (2)〜(7) を必要タイムステップだけ繰り返す．

3.2 計算と通信の見積り

まず，計算量のオーダは以下のとおりである．以下，N は総粒子数，N_{neib} は，ある粒子の近傍の粒子数，N_{src} は輻射の源となる粒子数，f_i は計算にかかるファクタを表す．

- 重力（GRAPE-6）：
  $f_1 N^2, f_1 \sim O(10^2)$

$\Rightarrow 1Tflop(N/10^5)^2/step$

- 流体（CP-PACS）:
  $f_2 NN_{neib}, f_2 \sim O(10^3), N_{neib} \sim O(10^2)$
  $\Rightarrow 10Gflop(N/10^5)/step$
- 輻射輸送 + 化学反応（CP-PACS）:
  原始組成のネットワークで調べる限り，化学反応に要する時間は輻射輸送に比べ無視できる．
  $f_3 N^{4/3} N_{neib} N_{src}, f_3 \sim O(10^4)$
  $\Rightarrow 100GflopN_{src}(N/10^5)/step$

したがって，演算量だけを評価すると，10万体程度の計算を行う場合，CP-PACSとGRAPE-6での計算がバランスするには，

$$N_{src} \sim 10 \left(\frac{N}{10^5}\right) \frac{\text{[CP-PACS flops]}}{\text{[GRAPE flops]}}$$

程度の N_{src} を扱うことになり，その下で $10^{4\sim5}$ ステップの計算を行う．

一方，通信性能に関する評価は以下のとおりである．GRAPE-6は重力多体計算の副産物として，各粒子に対する近傍粒子リストを生成することができる．しかし，現在のHMCSでは両システムの計算パワーは非常に強力であるため，通信コストを極力抑える必要がある．このため，近傍粒子リストは転送せず，CP-PACS側で直接生成するものとする．この場合のCP-PACSからGRAPE-6へのデータ転送では，各粒子に対して質量・3次元座標・1ステップ前の加速度のファクタ・1ステップ前のポテンシャルの合計6つの値を送る．倍精度浮動小数点を用いるため，総転送量は

$$N \times 6 \times 8/step = 48\,Mbyte \left(\frac{N}{10^5}\right)/step$$

となる．逆に，GRAPE-6からCP-PACSに戻されるデータは，各粒子に対して3次元加速度・ポテンシャルの合計4つの値である．よって，総転送量は

$$N \times 4 \times 8/step = 32\,Mbyte \left(\frac{N}{10^5}\right)/step$$

となる．

これらのパラメータを，予備実験のCP-PACS flops値とGRAPE-6 flops値，およびPIOの性能[5]と組み合わせた結果，10万体程度を扱う場合，1ステップあたりCP-PACS側で約16秒，GRAPE-6側で1秒未満，通信時間も1秒未満となり，CP-PACS側でのSPH計算が支配的となることが予測される．しかし，実際に重力効果をCP-PACSで処理した場合，全粒子データをレジスタはもちろん，キャッシュに保持しておくことも不可能なため，$O(N^2)$ の演算を数十万以上の粒子に対して行うことは現実的でない．したがって，GRAPE-6をHMCSに導入する意義は十分に高い．

4. プロトタイプの実装

4.1 CP-PACS

本論文はHMCS自体に関する説明を主とするため，CP-PACS上におけるSPH計算の詳細については省略し，並列化の要点と，GRAPE-6とのインタフェースに関して説明する．

CP-PACS上で動作するSPH計算プログラムでは，1000 PU規模の超並列アルゴリズムを実装している．ここでは，全粒子を座標に従って空間的に分割し，それらを並列処理する．ただし，粒子は全PUにおいてその数が均等になるように分割し，必ずしも直交空間のセルに従って厳密に分割するわけではない．すなわち，粒子処理における負荷分散が主目的であり，なるべく近い粒子を1つのPUに収めるようにすることは，近傍粒子計算を効率的に行うためのものである．この分割手法は，ORB（Orthogonal Recursive Bisection）[7]として知られているものと基本的に同じであり，粒子数は厳密に各PUに均等分割され，1 PUに割り当てられる粒子は空間的に近接している．

1つの計算ステップでは，GRAPE-6から返された現在の粒子配置における加速度より，粒子の位置を決定し，上述のPU間分割を行う．その後，全粒子に対し輻射輸送 + 化学反応の計算を行う．前章で述べたとおり，途中の計算では化学過程の収束のための反復計算をともなう．計算終了後，全粒子の質量・座標・加速度のファクタ・力のポテンシャルをGRAPE-6に送り，次の計算ステップに移行する．なお，今回の実装では各粒子の質量は変化しないが，後述のGRAPE-6クラスタAPIに合わせるため，毎回転送している．また，1000以上のPUからいっせいに，個別にデータ転送を行うのは通信回数と通信粒度の点であまりに非効率的である．よって，適当な数の入出力とりまとめPUを設け，これらがエージェントのような形でGRAPE-6クラスタと通信する．エージェントの数は必ずしもGRAPE-6クラスタのノード数と一致はしていない．

4.2 GRAPE-6クラスタ

HMCSプロトタイプでは，8枚のGRAPE-6ボードを用いている．これらを制御するため，8ノードのPCクラスタを用意し，各ノードのPCIスロットにGRAPE-6インタフェースボードを実装している．各ノードの主な仕事は，CP-PACSから並列ネットワーク経由で送られてくる粒子データをGRAPE-6ボードに送り，結果を再びCP-PACSに返すことであるが，これに付随して，クラスタ内での全粒子データの交換

作業も行う．

CP-PACS から送られてくる N 粒子のデータは，ネットワークにおける通信量を最小化するため，8 台のノードに $N/8$ 粒子ずつ送られる．しかし，GRAPE-6 ノードでは，全ボードに対し，全 j 粒子をセットし，そのうえで i 粒子を順次セットしながら演算を進めていく．したがって，各ノードは GRAPE-6 ボードをアクセスするにあたり，全粒子のデータを持っている必要がある．このため，ノード間で全対全のデータ交換を行う．GRAPE-6 クラスタのこれらの並列制御は MPI プログラミングによって行われている．クラスタ間の通信には，CP-PACS との通信において用いられているネットワークとは別のネットワークが用いられる．これは，CP-PACS との通信用並列ネットワークは複数のサブネットに分割されているため，クラスタ全体を束ねるフラットなネットワークが別途必要なためと，MPI による通信が CP-PACS との間の通信を邪魔しないように考慮してのことである．

クラスタのノードには Samsung UP1100 マザーボードを用いた．ここに CPU 周波数 600 MHz の Alpha 21264A を載せ，768 MB の SDRAM メモリで運用する．OS は RedHat6.2J for Alpha で，Linux カーネルは 2.2.16 である．UP1100 には on-board Ethernet として DEC Tulip ベースのものが載っているが，上述の用途のため Intel EthernetPro100 を別途実装している．

クラスタの各ノードでは，通信およびデータ交換を行うための g6pioserv と呼ばれるサーバプログラムが 1 つずつ走る．1 つの g6pioserv は CP-PACS 上の複数の入出力エージェントと通信するようプログラムされている．GRAPE-6 インタフェースの制御には，公開されている GRAPE-6 API[8]を用いる．これは GRAPE-6 ボードに対する一連のアクセスルーティンのライブラリである．また，PIO を用いた通信を行うため，PIO-API[5]を用いる．

4.3 PIO ネットワーク

両システムを結合する 100 base-TX ベースのネットワークは，並列入出力システム PAVEMENT/PIO[5]（以下，PIO）によって制御される．PIO は並列に張られたコモディティネットワークにおいて，バンド幅の増大・大容量バッファリングによる通信時間隠蔽・並列ネットワーク間の自動負荷分散制御等を目的とした，ユーザレベルの超並列計算機向け入出力システムである．PIO は本来，超並列計算機や大規模クラスタが持つ，多数の入出力プロセッサ（クラスタの場合は演算用プロセッサがこれを兼ねる）間を，100 base あるいは 1000 base Ethernet のようなコモディティ・ネットワークで並列に接続し，安価で高い総バンド幅を確保するとともに，ユーザ側に対してはこの並列接続を意識しないプログラミングを可能にするインタフェースを提供するシステムである．PIO では超並列計算機・共有メモリワークステーション・PC クラスタ等，あらゆる並列処理システムに適応する高い移植性を目指しているため，その実装は TCP/IP をベースとしたユーザレベルのライブラリおよびデーモンからなる．この上で並列プロセス間の並列通信や，分散ファイルシステムライブラリといったアプリケーションを提供する．

HMCS においては，この PIO を CP-PACS と GRAPE-6 クラスタ間の接続に利用する．CP-PACS は完成後 6 年以上経過している計算機であり，ネットワークインタフェースとしては 100 MB/s の HIPPI または 100 base-TX までの Ethernet にしか対応していないため，外部装置との転送バンド幅を確保する手段として PIO を用いている．また，PIO を用いることにより，アプリケーション（この場合は輻射輸送+SPH 計算を行う CP-PACS 上のプログラム）から見た GRAPE-6 クラスタへのデータ転送時間は，PIO のデーモンによってバッファリングされるため，アルゴリズムを工夫することにより通信時間の隠蔽を行うことが可能となる．我々は 3.1 節で述べた基本アルゴリズムを修正し，GRAPE-6 に最新の粒子データを与えると同時に，SPH 粒子の大きさの決定や時間刻み幅の決定等の処理を，1 ステップ前の粒子情報を元に計算することにより，GRAPE-6 クラスタでの処理と CP-PACS 上での処理をオーバラップさせることに成功した．GRAPE-6 の裏で走る CP-PACS 側の処理は，ある程度の誤差を折り込んで計算することにより，1 ステップ分のずれを吸収するように処理している．

PIO には並列ネットワークへの並列データ流の分散に関し，プロセスとネットワークの関係をつねに静的に固定する方法，ユーザがプログラム上で明示的にネットワークを指定する方法，動的にデータ流量を観察しデーモンが負荷分散を行う方法，の 3 通りが用意されている．今回の HMCS の実装では，CP-PACS 側のエージェントと，GRAPE-6 クラスタ上の g6pioserv との間で PIO による通信が行われる．エージェントあたりの送受信粒子データ数は均一化され，またデータ送受信タイミングもアプリケーションによりほぼ同期しているため，PIO がデフォルトで提供する静的なデータ流分散を用いることにした．

CP-PACS には合計 16 個の 100 base-TX NIC が装

表 1 GRAPE-6 クラスタアクセス API
Table 1 API to access GRAPE-6 cluster.

ルーティン名	機能
g6cpp_start	通信モードをセットし処理を開始する
g6cpp_unit	粒子数をセットし座標と時間刻みの値をセットする
g6cpp_calc	担当粒子データをセットし計算を開始する
g6cpp_wait	GRAPE-6 による演算結果を待ち加速度等のデータを返す

表 2 各種問題サイズにおける実行時間（秒）
Table 2 Execution time for various problem size.

内訳	問題サイズ (n)		
	15	16	17
particle data trans.	5.613	10.090	17.998
all-to-all data trans.	0.309	0.476	0.681
set j-particles	0.231	0.362	0.628
calculation	0.064	0.169	0.504
TOTAL	6.217	11.097	19.811

備されているが，これらは CP-PACS 上のどのプロセスからもアクセス可能である．図 1 に示したように，これらは 2 系統の FastEthernet スイッチを経由して 8 台の GRAPE-6 クラスタノードに接続される．CP-PACS は全 2048 ノードで運用する以外にも，数百ノード単位のパーティションに分割してジョブを実行することが可能であり，通常はこの形で運用されている．このため，HMCS では 8 台の GRAPE-6 クラスタを必ずしもすべて利用するのではなく，必要に応じて分割できるように g6pioserv および g6cpplib（後述）を設計・実装した．たとえば，CP-PACS を 1024 ノード ×2 に分割し，各々に PIO を通じて 4 ノードの GRAPE-6 クラスタを割り当て，2 種類の独立した計算を同時並行に進めることも可能となっている．また，CP-PACS のノード数だけでなく，実際に計算する粒子数に応じて GRAPE-6 クラスタノード数も選べるようになっている．このような実装により，HMCS を CP-PACS と GRAPE-6 クラスタの間だけで利用するのではなく，将来的に CP-PACS の代わりに別のスーパーコンピュータや大規模クラスタを利用し，GRAPE-6 クラスタを効率的に利用ことを可能とする．実際，我々は現在の CP-PACS と GRAPE-6 による HMCS のほかに，日立 SR8000 や COMPAQ Alpha CPU クラスタ（GRAPE-6 クラスタとは別）を CP-PACS の代わりに用いる実験も行った．

エージェントでのプログラミングを容易にするため，CP-PACS 側の任意のノードから，GRAPE-6 クラスタをアクセスするための g6cpplib と呼ばれる API を新たに設けることにした．この API ライブラリは，g6pioserv との間のプロトコルに基づき，PIO-API を用いた通信を行う．このライブラリを用いることにより，今後，CP-PACS から自由に GRAPE-6 クラスタをアクセスすることが可能となる．**表 1** に API の概略を示す．

5. 性能評価

4.3 節で述べたように，HMCS は CP-PACS 上の任意のパーティションと，GRAPE-6 クラスタのサブシステムを組み合わせて運用することができるようになっている．本章では，CP-PACS の 512 台または 1,024 台の PU と，4 ノードの GRAPE-6 サブクラスタを用いた性能評価について述べる．

5.1 問題サイズに対するスケーラビリティ

CP-PACS の 512 PU を用いたシステムにおける，各種問題サイズに対する性能評価結果を**表 2** に示す．表中の数値は，1 ステップのシミュレーション実行における処理時間の内訳を秒単位で表したものである．ここでは粒子数 $N = 2^n$ とし，n を 15 から 17 まで変化させている（$N = 32768, 65536, 131072$）．なお，N の半分は SPH 粒子，残り半分がダークマターであるため，CP-PACS における SPH 計算の対象粒子数は $N_{sph} = 2^{14}$〜2^{16} 個である．また，図 2 に示したように，SPH の処理においては，化学過程の収束を待つための反復計算が含まれ，この計算回数は初期段階では小さく，ある程度反応が進まないと正しい時間計測ができないことが分かっている．このため，長時間の実行履歴を参照し，この回数が安定した部分に関して計測を行った．実際に測定したのは 400 ステップから 800 ステップの間の 400 ステップ分である．

表中，“particle data trans.” は GRAPE-6 クラスタ（実際には g6pioserv）から見て，「前の回の加速度データを CP-PACS に返し終わり，次の粒子データの受信を完了するまで」の時間である．PIO が通信経路上でのバッファリングを行うため，CP-PACS 側と GRAPE-6 側では非同期通信が行われることになる．これを利用して CP-PACS 側での SPH 処理をオーバラップさせているため，この部分について厳密に通信と計算を切り分けることはできない．すなわち，GRAPE-6 から見ると，通信時間および CP-PACS 側での SPH 処理時間の総和がこの時間として見えていることになる．“all-to-all data trans.” は，GRAPE-6 クラスタの各ノードがデータを並列受信した後，全対全通信によりデータ交換を行うのに要する時間である．“set j-particles” は各ノードが GRAPE-6 ボード上のローカルメモリに対し，彼演算対象となる全粒子データを格納するまでに要する時間，そして “calculation”

は演算対象粒子をパイプラインに流し，全演算結果を引き上げるまでの時間である．

まず全実行時間に関しては，粒子数を倍に増やした際，いずれの場合も実行時間は 1.78 倍程度になっている．これは，粒子データ転送や GRAPE-6 に対する演算立ち上げのオーバヘッドが，粒子数の増加にともなって緩和されている影響と考えられる．したがって，粒子数に対する演算性能のスケーラビリティは確保さていると考えてよい．正確にいえば，SPH 演算の処理量 $O(N)$☆に対し，重力計算の方は $O(N^2)$ である．実際に “calculation” の部分は非線型に増加している（$n = 15$ からの 4 倍増加に対して約 7.8 倍）．しかし，この程度の粒子数では結局計算の絶対時間のオーダが異なるため，総実行時間に対してはほとんど影響が見られない．

今回用いたプログラム構成では，エージェントの数を 8 に固定している．したがって，PIO 通信にかかわる CP-PACS 上の並列プロセスは 8 つであり，これらと 4 ノードの GRAPE-6 クラスタが通信することになる．すなわち，GRAPE-6 側の個々の g6pioserv はそれぞれ 2 つのエージェントの世話をしている．今回，実験時間の制約上，GRAPE-6 クラスタのノード数をさらに削減したり，PIO 通信の並列ネットワーク接続数を変化させたりする実験を行うことはできなかった．しかし，今回の測定結果を見ると，GRAPE-6 における処理時間は全実行時間のオーダから見ると非常に小さく，ノード数を削減できる可能性がある．ただし，このノード数を減らすと PIO によって並列転送されるデータが逐次化されることになり，通信時間の増加を招く恐れもある．

この問題に関しては，たとえば GRAPE-6 側と CP-PACS 側のタイムチャートの刷り合わせ等により，通信時間の推定を行う必要があるが，両システムが共通クロックを持っているわけではないため，精密な測定は困難である．現状では，HMCS 内の GRAPE-6 クラスタをいくつかの構成のために分割可能として運用しているため，実際のアプリケーションの進み具合を見定め，適用ノード数を制御するのが現実的である．

5.2 PU 台数に対するスケーラビリティ

現在我々が行っているプロダクトランの最大構成である，$n = 17$（すなわち $N = 131072$ で SPH 粒子とダークマターが半分ずつ）の場合について，CP-PACS の PU を 512 台用いた場合と 1,024 台用いた場合について比較した結果を**表 3** に示す．GRAPE-6 クラスタはどちらも 4 ノードである．表の見方は表 2 と同様である．

表 3 CP-PACS の PU 台数による比較（秒）
Table 3 Comparison between various PU numbers on CP-PACS.

内訳	PU 台数	
	512	1024
particle data trans.	17.998	10.594
all-to-all data trans.	0.681	0.639
set j-particles	0.628	0.609
calculation	0.504	0.503
TOTAL	19.811	12.345

PU 台数を 2 倍にした結果，総実行時間で見て約 1.6 倍の性能向上となった．通信時間を含めた CP-PACS 側の SPH 処理時間では約 1.7 倍の性能向上がみられる．GRAPE-6 クラスタ側の条件はまったく同じである．“calculation” 以外の時間で若干差がみられるが，これらの処理の途中では，時間計測を厳密にするためのバリア同期等をとっていないため，多少の測定の乱れはやむをえないと思われる．

CP-PACS 側での処理が PU 台数に対して線形に速度向上しない理由はいくつか考えられる．その 1 つは，我々のアルゴリズムでは CP-PACS に戻された加速度データに基づき，粒子位置の計算とともに「近傍リスト」を作成している．これは，個々の粒子の近傍の粒子を調べ，SPH における流体計算において PU 間通信すべきデータを抽出するための前作業であり，PU 間で完全並列化することは難しい．3.1 節で述べたように，GRAPE-6 自体にもこの近傍リストを計算の副産物として生成しホストに返す機能が備わっているが，このリストの大きさは個々の粒子に対して 100 程度のオーダになるため，加速度データよりはるかに大きくなってしまい，通信時間を増大させる可能性がある．このため，我々は近傍リストを CP-PACS 内で作成することにした．しかし，もしこの部分がボトルネックとなっているのであれば，GRAPE-6 による近傍リスト作成の場合との比較を，前述した通信時間の推定とともに再評価する必要がある．

もう 1 つの可能性は，PIO が対象としている並列プロセス数の増加にともなう，管理テーブルの増大やキャッシュミスの増加である．我々のプログラムではエージェントの数をパラメータとして設定可能になっているが，今回はこれを 8 に固定して測定した．しかし，PIO 自体はシステム上のどのプロセスからの通信も許すため，その受け入れのための通信バッファを最低限のサイズだけは用意する．対象プロセス数が増加

☆ さらに厳密にいえば前述の化学過程の反復回数の問題があるため完全に線形ではないが，定常状態ではほぼ安定と考えてよい．

すると，このバッファサイズが縮小されるため，結果的に通信効率が低下する可能性がある．これも前述のように通信時間の厳密な測定を行わないと解析できないが，PIO システム自体に性能測定のための計測機構を組み込む等して調査する必要がある．

これらの課題は今後解決していくべき問題である．いずれにしても，CP-PACS の最大構成である 2048 PU を用いた場合でも，このスケーラビリティは保たれるかあるいはさらに若干落ちる可能性がある．$N = 2^{17}$ 規模に関しては，ステップあたり 8 秒程度の総実行時間が予想される．

5.3 実計算結果の例

1024 PU の CP-PACS を用い，$N = 2^{17}$ 粒子を対象として行った銀河形成シミュレーションの実行結果を**図 3** に示す．これは合計 25,000 ステップのシミュレーションを実行し，その過程における系の中のガス流体の密度分布を 3 次元ボリュームレンダリング処理によってイメージ化したものである．計算の進行に従い，当初ほぼ均等な密度だったものが粗密を持つ構造に変化していく様子がとらえられている．

現在のシステムではこの程度の問題を約 4 日でシミュレーションすることが可能である．今後，この規模のシミュレーションを重ね，とらえるべき問題領域を絞りながら，より大きな系に対するシミュレーションを行うことを検討している．また，先述したように，GRAPE-6 クラスタというリソースを効率的に利用するために，重力計算リクエストを出す汎用計算機を CP-PACS に限らず，他の並列計算機やクラスタからも利用できるようシステムを拡張中である．

6. おわりに

本論文では，マルチスケールな複合物理シミュレーションを行うための新しい計算プラットフォームとして，HMCS（Heterogeneous Multi-Computer System）を提案し，そのプロトタイプとして，汎用超並列計算機 CP-PACS と重力計算専用並列システム GRAPE-6 を，並列コモディティネットワーク制御システム PIO によって結合した実装例を示した．その結果，SPH 粒子とダークマターを半分ずつ含む 13 万粒子の系に対し，1024 台の CP-PACS の PU と 4 ノードの GRAPE-6 クラスタを用いて，1 ステップあたり約 12 秒で，自己重力効果を含む輻射流体計算による銀河形成シミュレーションが実現可能なシステムが構築された．

今後の課題として，HMCS の構成要素である，g6pioserv サーバプログラムと g6cpplib API ライブラ

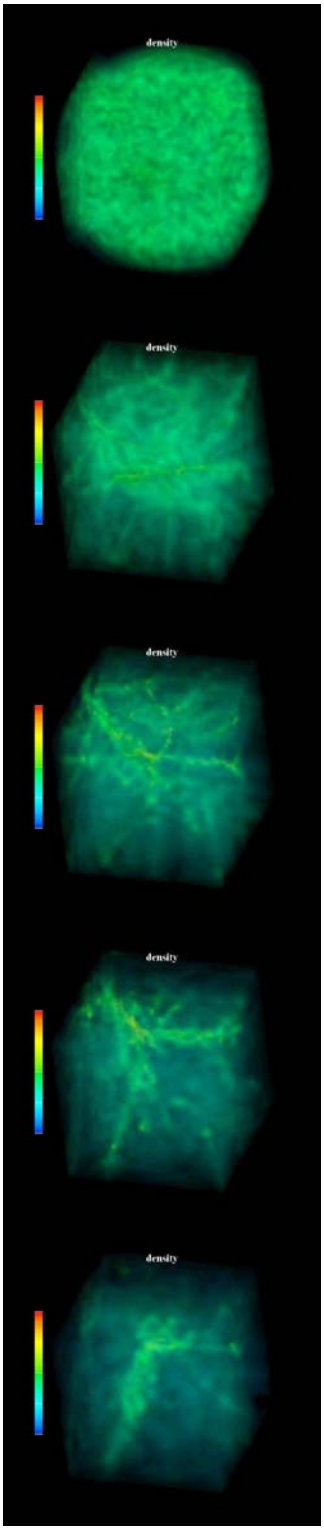

図 3 HMCS による計算結果の例
Fig. 3 A sequence of simulation results by HMCS.

リのチューニングを行うことが必要である．特に PIO を用いたプロトコルでは，ユーザデータの配列の制約上，10 万粒子程度では通信粒度が比較的小さく，より効率の良い実装を考える必要がある．また，現在の g6cpplib では GRAPE-6 が提供する機能のうち一部のみを用いており，より広汎なシミュレーションを行うため，API としての充実化を図る必要がある．

現在，HMCS は筑波大学計算物理学研究センター[9]で稼働中であるが，HMCS をさらに拡張し，このような閉じた環境だけでなく，高速な広域ネットワーク上でのグリッド環境における適用も視野に入れ，改良を行っている．さらにその先の大きな課題としては，HMCS そのものの枠組みの見直しがある．現在の HMCS は基本的に 2 つの高性能並列処理システムを結合した形をとっているが，より柔軟で高い性能を求めるならば，両者の結合ネットワークがボトルネックになる．たとえば，並列システムのノード内において，汎用プロセッサの横にコプロセッサのような形で GRAPE チップを設け，並列処理システムが持つ高いネットワークバンド幅をそのまま活かした通信を行うような,「汎用・専用ノード一体型」の HMCS を実現するのが理想であろう．

謝辞 本研究を行うにあたり，ご意見･ご協力をいただいた筑波大学計算物理学研究センター未来開拓プロジェクト関係各位に感謝いたします．なお，本研究の一部は日本学術振興会未来開拓学術研究推進事業「計算科学」プロジェクト（No.JSPS-RFTF 97P01102）および科学研究費補助（基盤研究（C）課題番号 14580360）によるものである．

参 考 文 献

1) McLeod, K.K. and Rieke, G.H.: Luminous Quasars in Luminous Early-Type Host Galaxies, *Astrophysical Journal*, No.454, pp.L77–L80 (1995).
2) Bahcall, J.N., Kirhakos, S. and Saxe, D.H.: Hubble Space Telescope Images of a Sample of 20 Nearby Luminous Quasars, *Astrophysical Journal*, No.479, pp.642–658 (1997).
3) Makino, J., et al.: A 1.349 Tflops simulation of black hole in a galactic center on GRAPE-6, *Proc. SC2000* (CD-ROM), IEEE, Los Alamitos (2000).
4) Nakazawa, K., et al.: CP-PACS: A massively parallel processor at the University of Tsukuba, *Parallel Computing*, Vol.25, pp.1635–1661 (1999).
5) 松原正純ほか：分散メモリ型超並列計算機における並列入出力，情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム，Vol.41, No.SIG 5(HPS1), pp.58–69 (2000).
6) Umemura, M.: Three-dimensional hydrodynamical calculations on the fragmentation of pancakes and Galaxy formation, *Astrophysical Journal*, No.406, pp.361–382 (1993).
7) Culler, D.E., Singh, J.P. and Gupta, W.: *Parallel Computer Architecture: A Hardware/Software Approach*, p.170, Morgan Kaufmann Publishers, ISBN 1-55860-343-3 (1998).
8) GRAPE-6 documents (on WWW). http://grape.astron.s.u-tokyo.ac.jp/pub/people/makino/softwares/GRAPE6/
9) http://www.rccp.tsukuba.ac.jp/

(平成 14 年 2 月 5 日受付)
(平成 14 年 5 月 22 日採録)

朴 泰祐（正会員）

1984 年慶應義塾大学工学部電気工学科卒業．1990 年同大学大学院理工学研究科電気工学専攻後期博士課程修了．工学博士．1988 年慶應義塾大学理工学部物理学科助手．1992 年筑波大学電子・情報工学系講師，1995 年同助教授，現在に至る．超並列処理ネットワーク，超並列計算機アーキテクチャ，ハイパフォーマンスコ ンピューティング，並列処理システム性能評価の研究に従事．電子情報通信学会，日本応用数理学会，IEEE 各会員．

牧野淳一郎

1985 年東京大学卒業．1990 年同大学大学院総合文化研究科博士課程修了，博士号取得．東京大学教養学部助手，同助教授を経て，現在東京大学大学院理学系研究科天文学専攻助教授．専門は理論天文学，特に恒星系力学．主に興味があるのは球状星団，銀河中心等の高密度恒星系の熱力学的進化．1995，1996 年および 1999–2001 年ゴードンベル賞受賞．1998 年日本天文学科林忠四郎賞受賞．

須佐　元

1994 年京都大学理学部卒業後，1997 年京都大学理学研究科物理学宇宙物理学専攻で理学博士取得．2000 年 4 月より筑波大学で助手，2002 年 4 月より立教大学理学部物理学科専任講師．専門は宇宙物理学の理論的研究．特に輻射流体の問題としての銀河および第 1 世代天体の形成を研究課題としている．最近は隕石中のコンドリュールという構造の形成に関しても研究を始めている．

梅村　雅之

1957 年生．1982 年北海道大学物理学科卒業．1987 年同大学大学院理学研究科物理学専攻博士課程修了．理学博士．1986，1987 年日本学術振興会特別研究員．1988 年京都大学基礎物理学研究所非常勤講師．同年，国立天文台助手．1992 年米国プリンストン大学客員研究員．1993 年筑波大学物理学系（計算物理学研究センター）助教授．同年，国立天文台客員助教授．2002 年から筑波大学物理学系教授．専門は理論宇宙物理学．特に宇宙輻射流体力学による銀河形成，宇宙構造形成の研究に従事．1989〜1990 年日本天文学会理事．日本天文学会，米国天文学会各会員．国立天文台理論計算機共通専門委員会台外委員．

福重　俊幸

1991 年東京大学教養学部卒業後，1996 年東京大学大学院総合文化研究科広域科学専攻で学術博士取得．1996 年 4 月より 6 月まで日本学術振興会特別研究員．1996 年 6 月より東京大学大学院総合文化研究科助手，現在に至る．専門は宇宙物理学の数値シミュレーションを使った研究，そのための専用計算機の開発等．1996 年および 1999–2001 年ゴードンベル賞受賞．

宇川　彰

1949 年生．1972 年東京大学理学部物理学科卒業．1976 年同大学大学院理学系研究科単位取得退学．1977 年理学博士．Cornell 大学，CERN，Princeton 大学を経て，1981 年東京大学原子核研究所助教授．1984 年筑波大学物理学系助教授．1990 年同教授．1998 年より筑波大学計算物理学研究センター長．素粒子を記述する場の理論，特に格子 QCD の数値シミュレーションによる研究を専門とし，CP-PACS 計画以来計算機開発にも従事．日本物理学会会員．