

「京」による ダークマターハローシミュレーション

牧野淳一郎

東京工業大学 地球生命研究所

理化学研究所 計算科学研究機構

概要

- 宇宙の始まり、はて、終わりはどうなってるのか？
- 何がまだわかっていないのか？
- シミュレーションでわかることは？
- 「京」でどんな計算をしようとしているか？
- さらにその先は？
- 計算機の話: エクサスケールに向けて

宇宙の始まり、はて、終わり

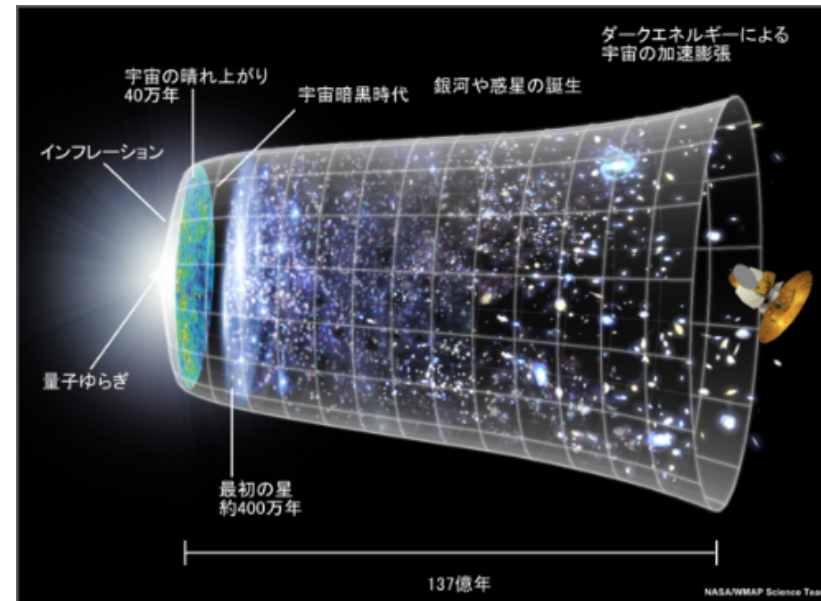
世界の始まり、はて、終わりがどうなっているか？

- 人類にとって根源的ななにか？
 - 大抵の神話、宗教は「世界の始まり」と「終り」についての物語を含む
 - キリスト教、ギリシャ神話、その他多数の例
- 素粒子物理学と天文学の進歩により、「科学の対象」になってきた
 - なにか役にたつか？と言われるとただちに役に立つわけではないが、、
 - 科学の限界を広げる、という意味でも重要

宇宙の始まり、はて、終わり

まず、現在の我々の理解を簡単に。

- 始まり: 本当の最初がなんだったかは良くわからないが、ものすごく密度・温度が高い状態があって、そこから現在まで膨張してきた。



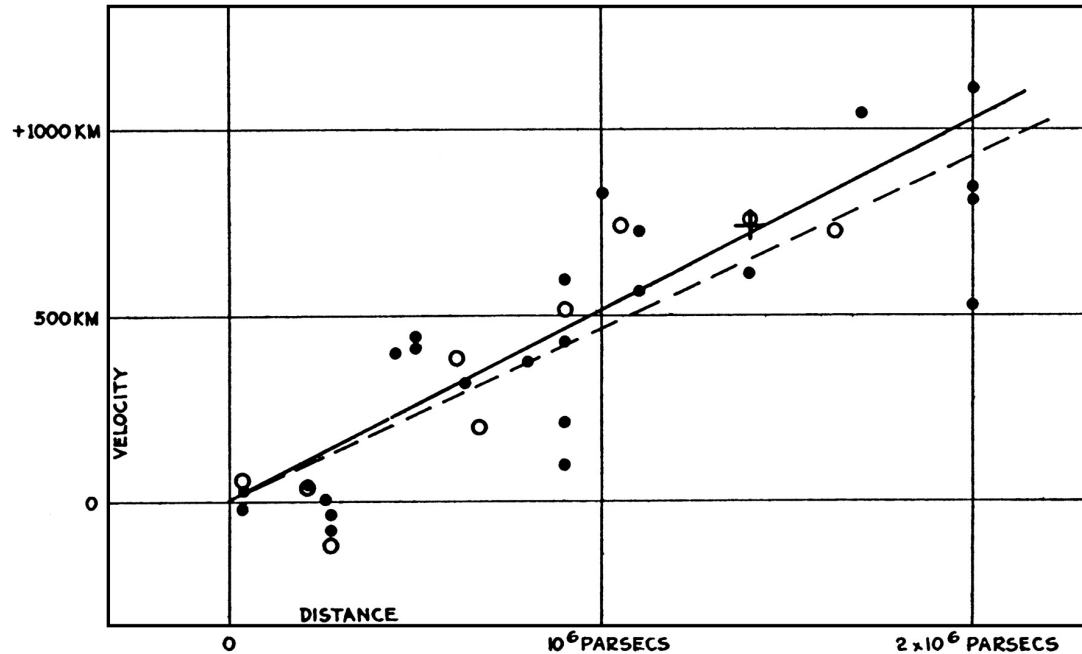
- はて: 我々が知る (観測する) ことのできる宇宙にははてがある (宇宙ができてから現在まで光が走る距離)。インフレーション理論では、その外側にもかなり広く同じような宇宙が広がっている。
- 終わり: 膨張が速くなっていき、無限にひろがって最終的にはほぼ「ダークエネルギー」だけの宇宙になる。

キーワード: 「宇宙膨張」 「ダークエネルギー」

それぞれについてもうちょっと詳しく。

宇宙膨張

20世紀初め: E. ハッブル



遠くの銀河ほど速く
我々の銀河系から遠
ざかっている

「宇宙膨張」

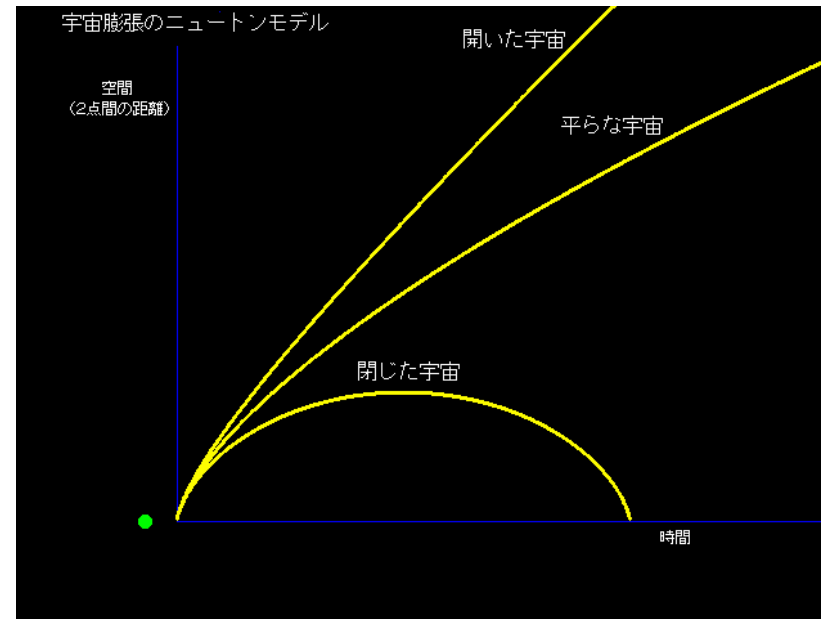
我々の宇宙は「ビッグ
バン」から始
まった

宇宙が膨張するって？

- 一応正しいんだけどあんまりわかった気がしない説明:

アインシュタインの一般相対性理論の方程式を、「宇宙が空間的に一様」として解くと、「静止している」という解はなくて「膨張している」か「収縮している」である

謎な定数をいれて静止解も出すことはできるが



- もうちょっと感覚的な説明:

宇宙に物質があれば、必ず重力があって、お互いにひきあう。なので、「止まっている」解はない。全体として膨張、全体として収縮、はありうる。

重力のため、段々膨張がゆっくりになる。

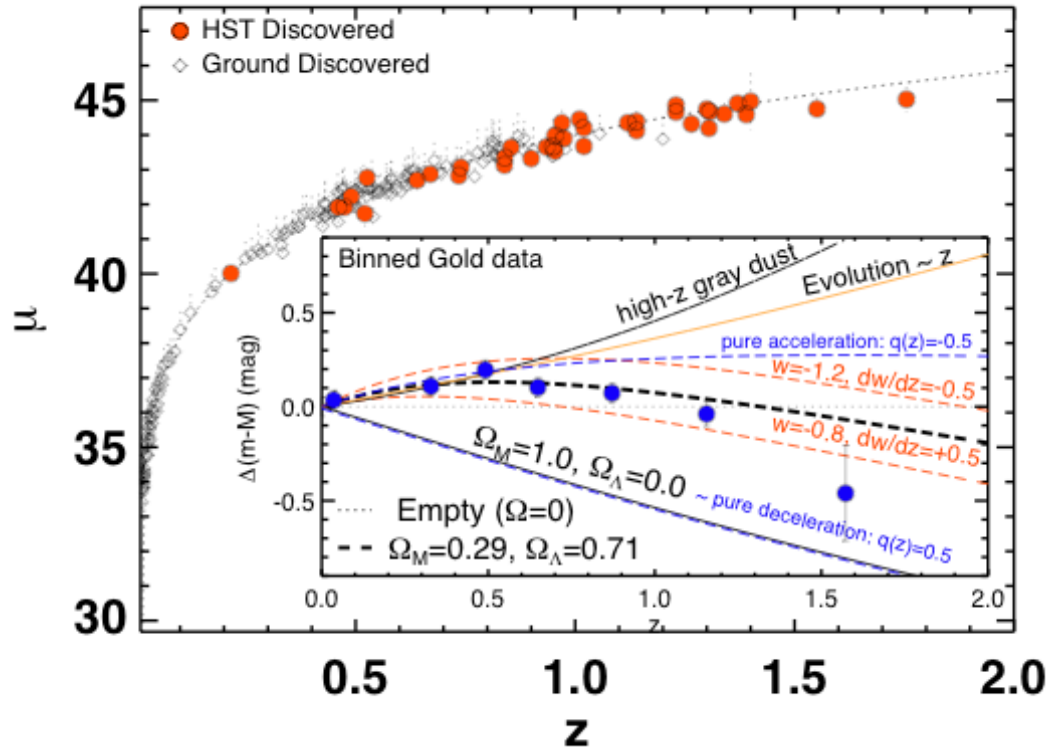
どんなふうによっくりになるか？

- 現代の宇宙物理学の基本問題だった。2000年代はじめまでほぼ1世紀に渡る論争
- 15年くらい前までの支配的な考え:(意味はちょっとおいといて)「平坦な宇宙」
 - 無限の未来に膨張速度がゼロに近づく
- 最近の観測からの示唆:実はよっくりにならない。無限の未来に無限に速くなる

非常に予想外な発見。

宇宙膨張の加速

遠方の超新星の明るさを観測する: 同じ「赤方変移」でも膨張のしかたで距離、従って明るさが違う



- 普通に平坦な宇宙:
明るい
- 物質が少ない宇宙:
暗い
- 膨張が加速している
宇宙: もっと暗い
これが我々の宇宙

2011 年ノーベル物理学賞

膨張を加速しているなにか=ダークエネルギー

では「物質」のほうは？

- 観測の示唆: $\text{ダークエネルギー} + \text{物質} = \text{「1」}$
- ダークエネルギー: 68.3%, 「**ダークマター**」: 26.8%, 普通の物質: 4.9%
- 普通の物質: 陽子、電子、中性子からなる普通の元素。それぞれクォークからできている。
- ダークマター: 普通の物質「ではない」なにか。現在の宇宙ではほぼ重力しか働いていない

ダークマターは何か？

大きくわけて 2 つの理論：

- Hot dark matter 質量をもったニュートリノが大量にあって、それが宇宙の物質のほとんどを占めている。
→ 銀河がまだできていないはず、という問題、、、
- Cold dark matter 未知の素粒子があってそれが宇宙の物質のほとんどを占めている。
→ こちらが有力。但し全く正体不明

ダークマターの正体 ???

- というわけで、現在のところダークマターの正体は「未知の素粒子」
- 有力な候補、と考えられているもの: 「超対称性理論」で予言されている粒子 (どういう理論でどういう粒子かはあまり聞かないで)
- 名前: 「ニュートラリーノ」、質量: 陽子の100倍くらい?
- 普通の物質や他のダークマター粒子と、全く相互作用しないわけではない。
 - 1秒に1億個くらいのダークマター粒子が我々の体を通り抜けている
 - ダークマター粒子が私の体の原子とぶつかる: 1000年-1億年に1度くらい?
 - ダークマター粒子同士の衝突、というのもある。

ダークマター探査

2つの方針:

- 直接検出: 検出器を通り抜けるダークマター粒子が普通の物質とぶつかり、はね飛ばすのを検出 (日本の XMASS、アメリカの CDMS-II など) CDMS-II は「発見したかも」と4月に発表したか???
- 間接検出: 宇宙の中でダークマター粒子が集まっているところでの対消滅からでてくるなにか (線? 電子? 陽電子?) を人工衛星で観測 (Fermi 望遠鏡の天体の中にないか? AMS 実験:ISS 上で反粒子を観測) AMS も「発見したかも」と4月に発表したか???

もちろんまだ見えてないので、どこにどれだけあるのかよくわからない

シミュレーション: どう分布しているかを予測。直接検出、間接検出について、どんなふうに見えるかはいえるはず。

ダークマター構造形成シミュレーション

計算の1例 戦略分野5 石山さん
できた構造の1つ
ここでやっていること：

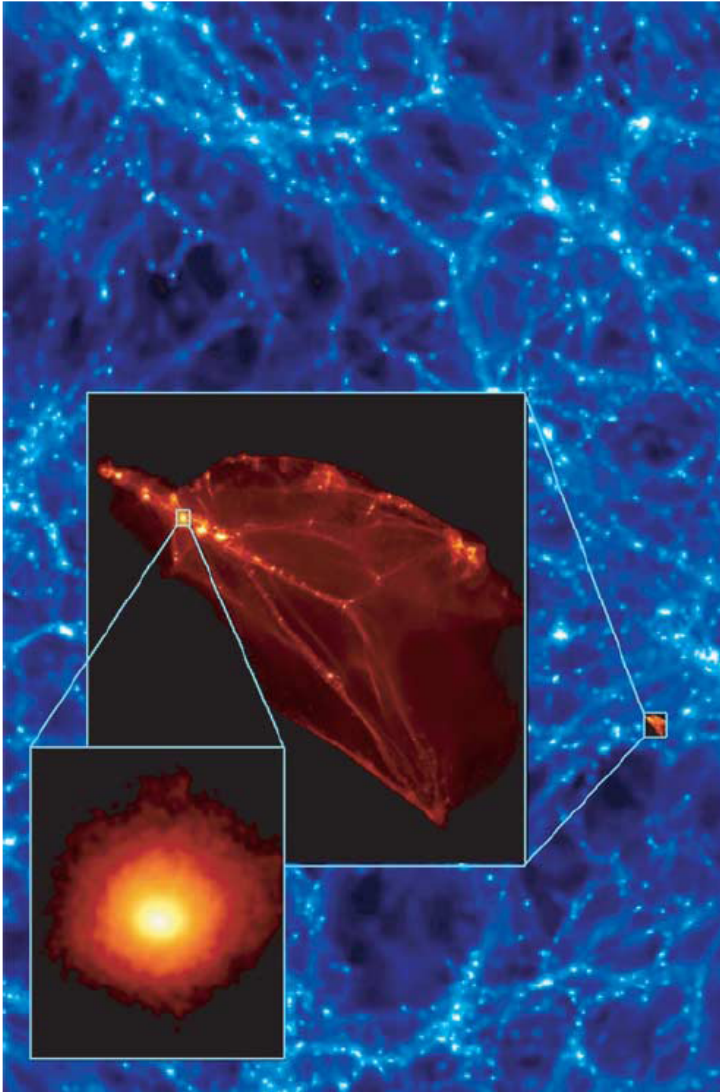
- 基本的には「一様」な宇宙を、なるべく沢山の粒子で表現する
- 理論的に「こう」と思われる揺らぎを与える
- 理論的に「こう」と思われる初期の膨張速度を与える
- あとは各粒子の軌道を数値的に積分していく。

計算について

- 最初に密度が高いところは、他に比べて相対的に密度がどんどん大きくなっていく。これは重力で集まるから。
- 最初に小さいものが沢山できて、それらがだんだん集まって大きなものになる。
- この計算では、銀河くらいの天体の形成を計算
- ダークマター粒子検出で興味: **銀河の中でのダークマター粒子の分布**
- 理論的には地球質量くらいのものが最初に沢山できる。それが集まったり、一部生き残ったりする。

「京」を使ってこの過程を調べたい

従来の研究



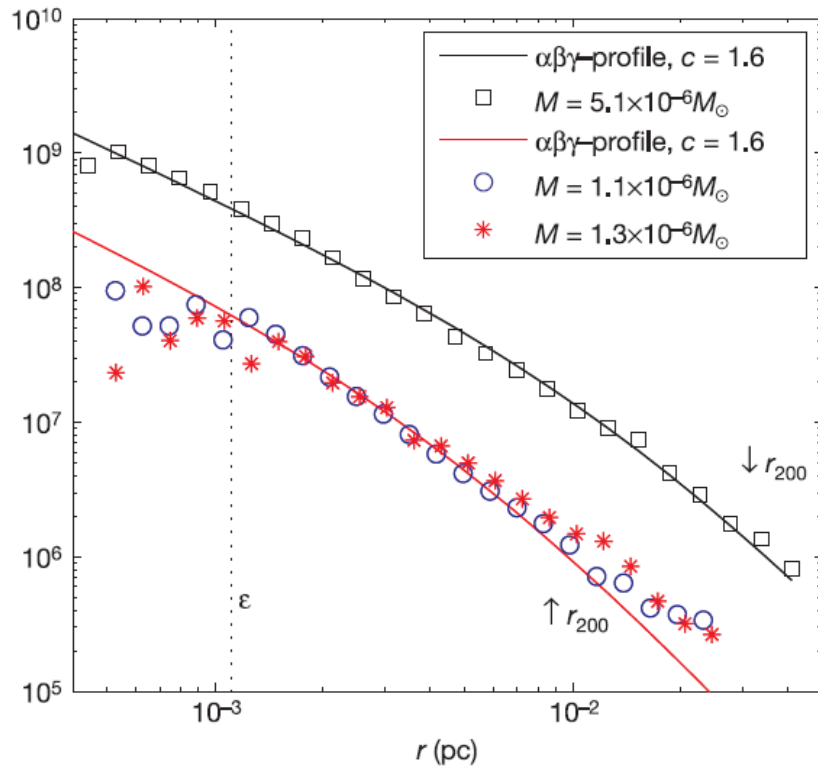
Diemand et al. 2005, Nature
433, 389

地球質量スケールに1万粒子くらい。全体で1億粒子程度？

それなりに大規模な計算だが、分解能は全く不足

分解能の問題

- 中心部の密度構造が分解能で決まってしまう



- 対消滅の確率: 中心密度がどのように上がるかで決まる。

- 半径の -1.5 乗以上で上がると中心で確率発散 (非常に明るくなる可能性あり)

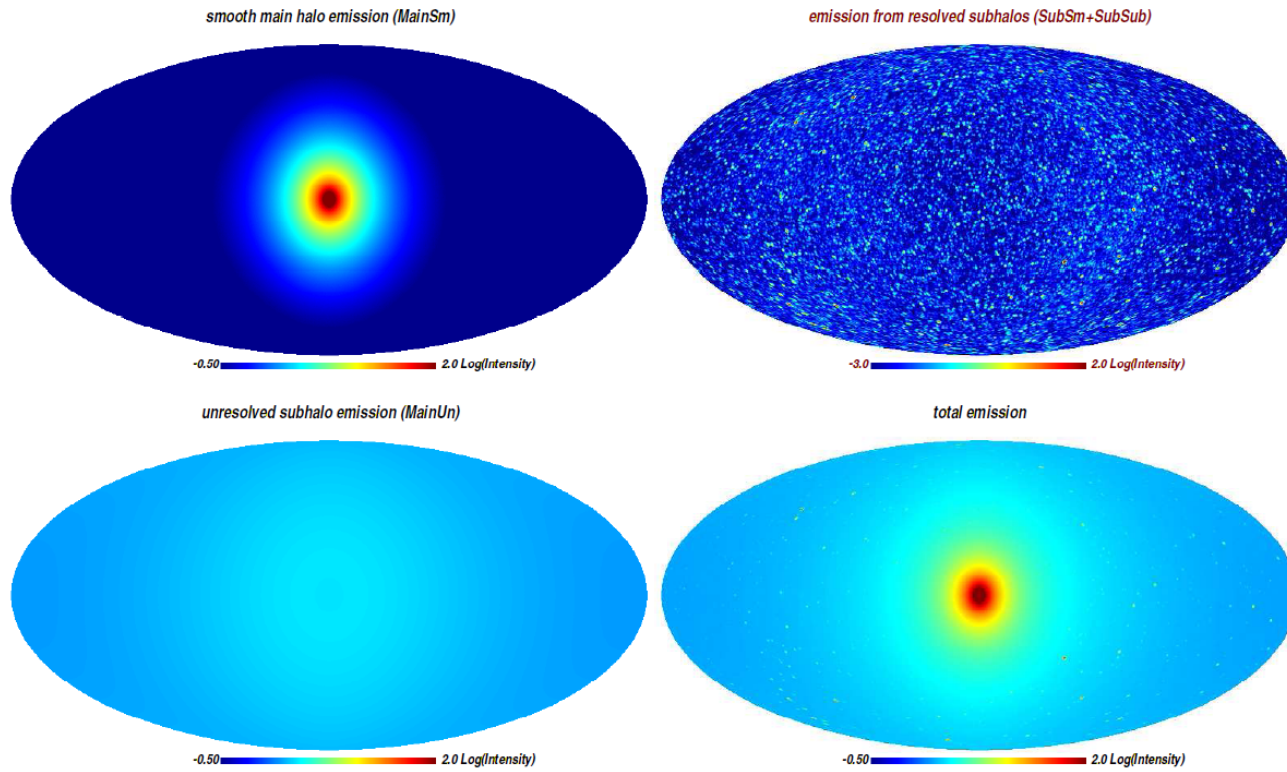
論争

- もしも現在まで生き残っていれば、地球質量スケールハローは対消滅 γ 線の主要なソースである
- しかし、
 - 他のハローとの合体成長
 - 大きなハローからの潮汐力
 - 恒星との遭遇

で破壊される可能性あり。

下の2つは、この論文の密度プロファイルが本当なら有効だが、本当か？

Springel et al 2008



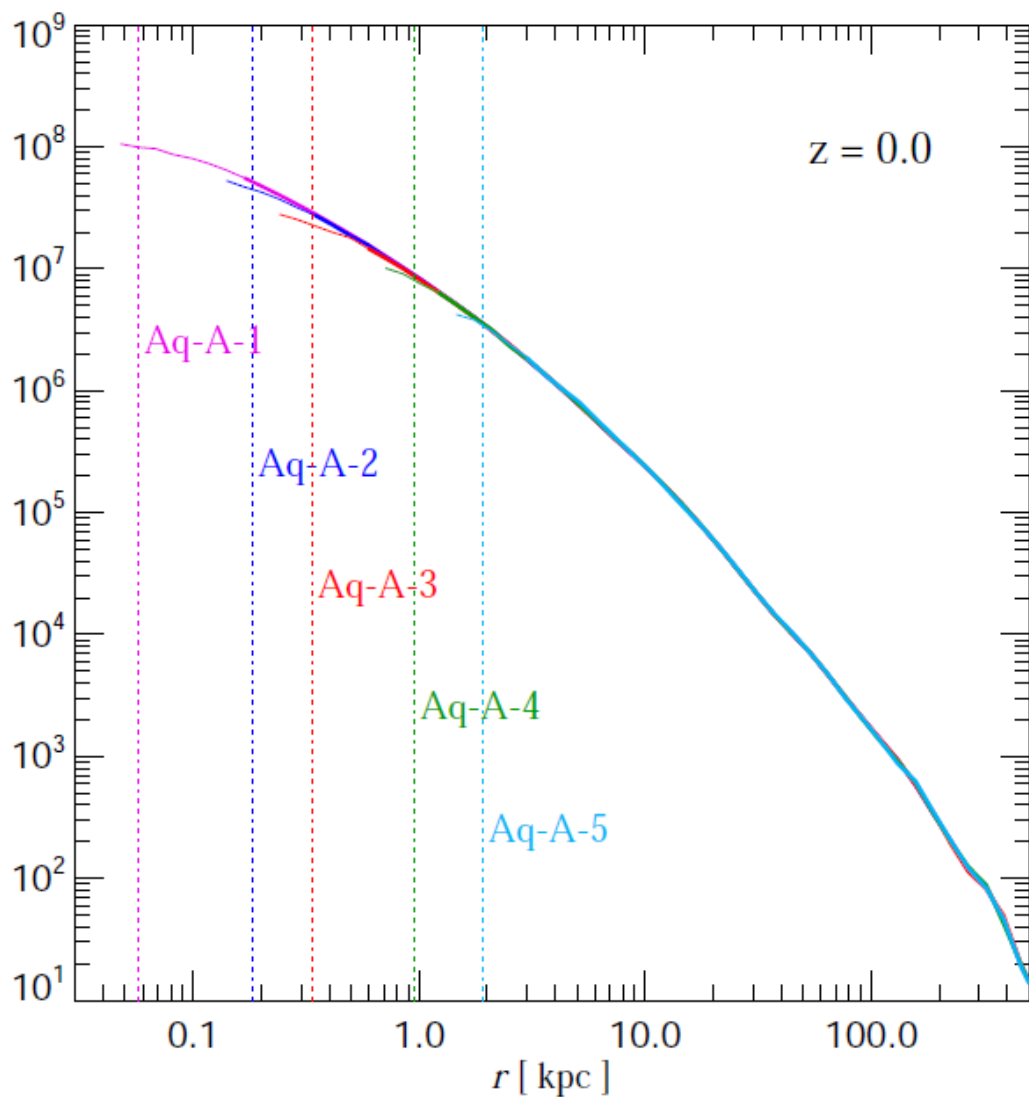
「小さなハローは残らない」という結果

低分解能計算の問題

- 2体緩和: 中心部分を加熱 (外側のほうが温度高いため)、フラットなコアにしてしまう。
- 重力をソフトポテンシャルに変えている ($(r^2 + \epsilon^2)^{-1/2}$)。これで空間分解能決まる

典型的には、半径 r での構造がそれらしく求まるためには、その内側に1万粒子以上必要。

最も高解像度のダークハロー計算

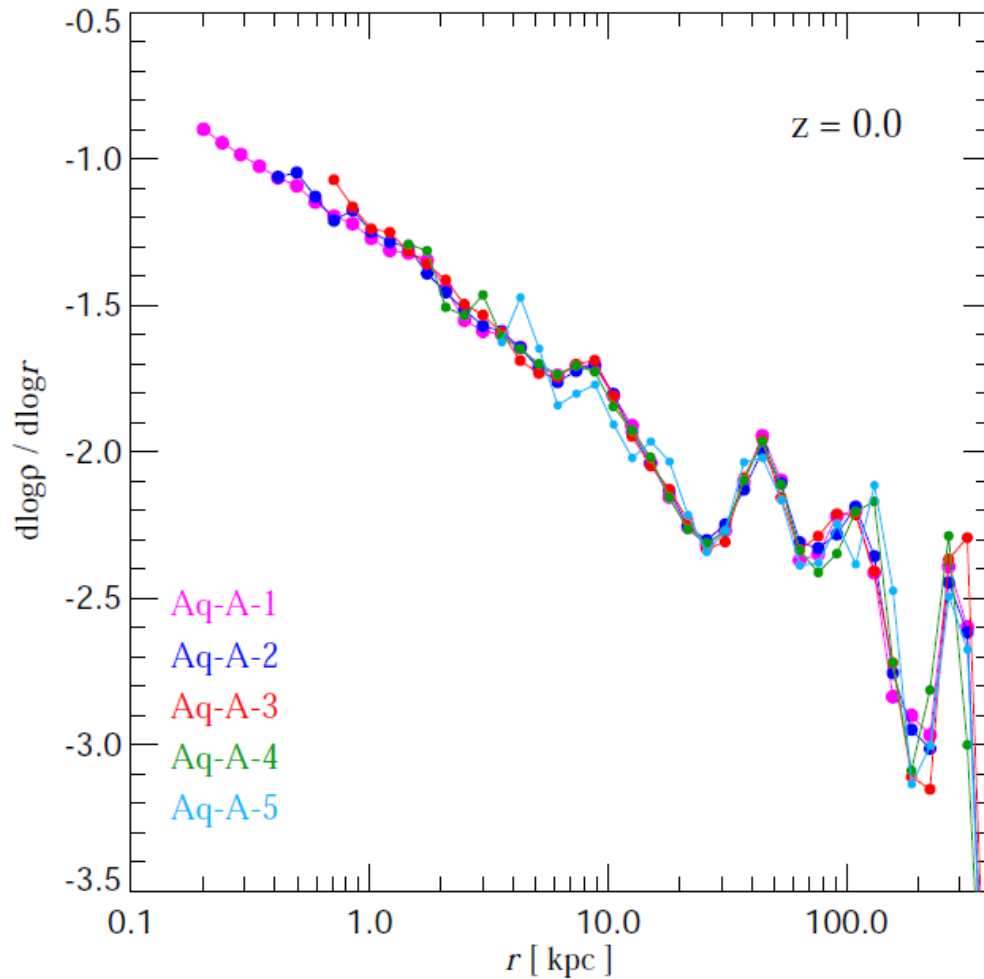


今のところ最大規模 Springel et al (2008)

N を3桁変えた

収束しているようにはみえる。

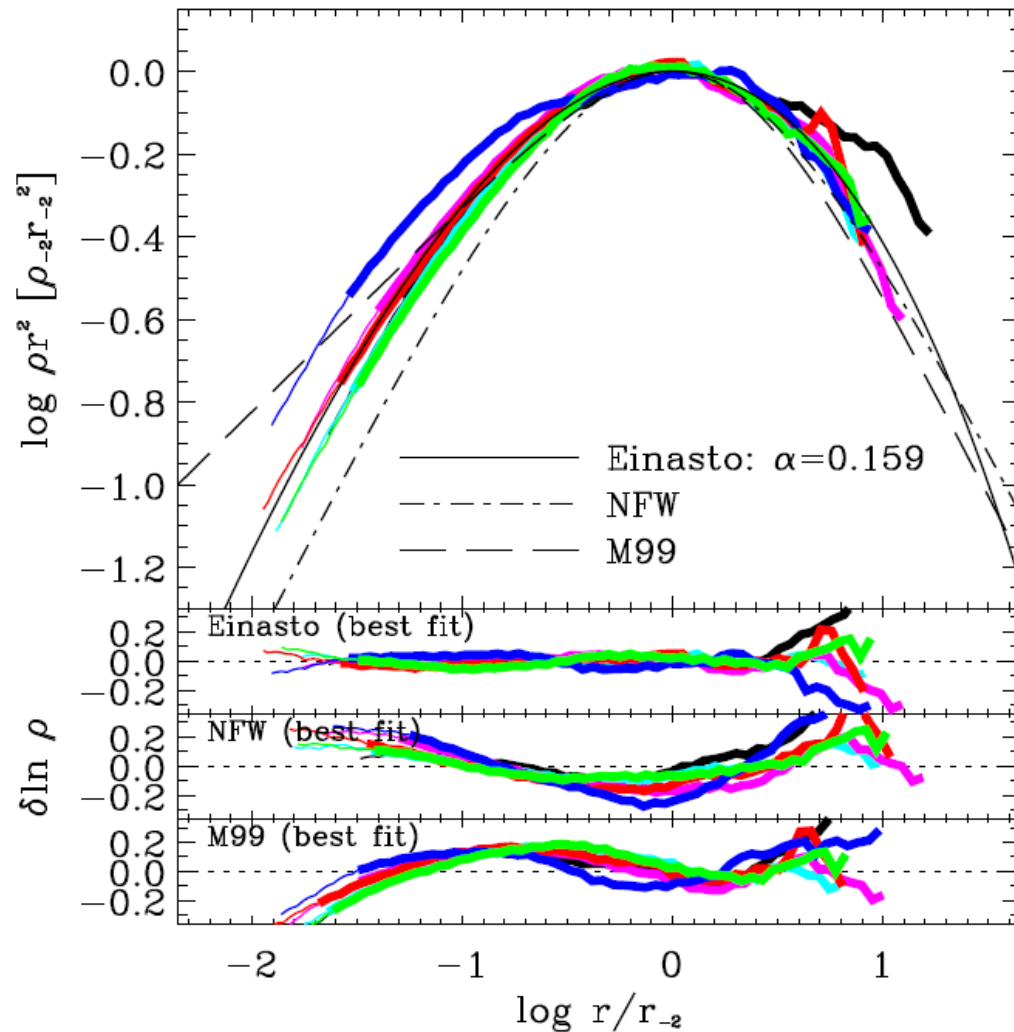
密度プロファイルのべき指数



中心に行くほど浅くなる

単一のべきになってない

モデル(理屈があるわけじゃない)との比較



Navarro et al. 2008

$$\text{NFW: } \frac{1}{r(1+r)^2}$$

Moore99:

$$\frac{1}{r^{1.5}(1+r^{1.5})}$$

Einasto:

$$\exp[(-2/\alpha)(r^\alpha - 1)]$$

ダークハローシミュレーションの現状

- 銀河サイズハローの計算では、数値計算結果では密度べきが中心にいくほど浅くなる
- 理屈は不明
- 地球サイズは大規模計算がない

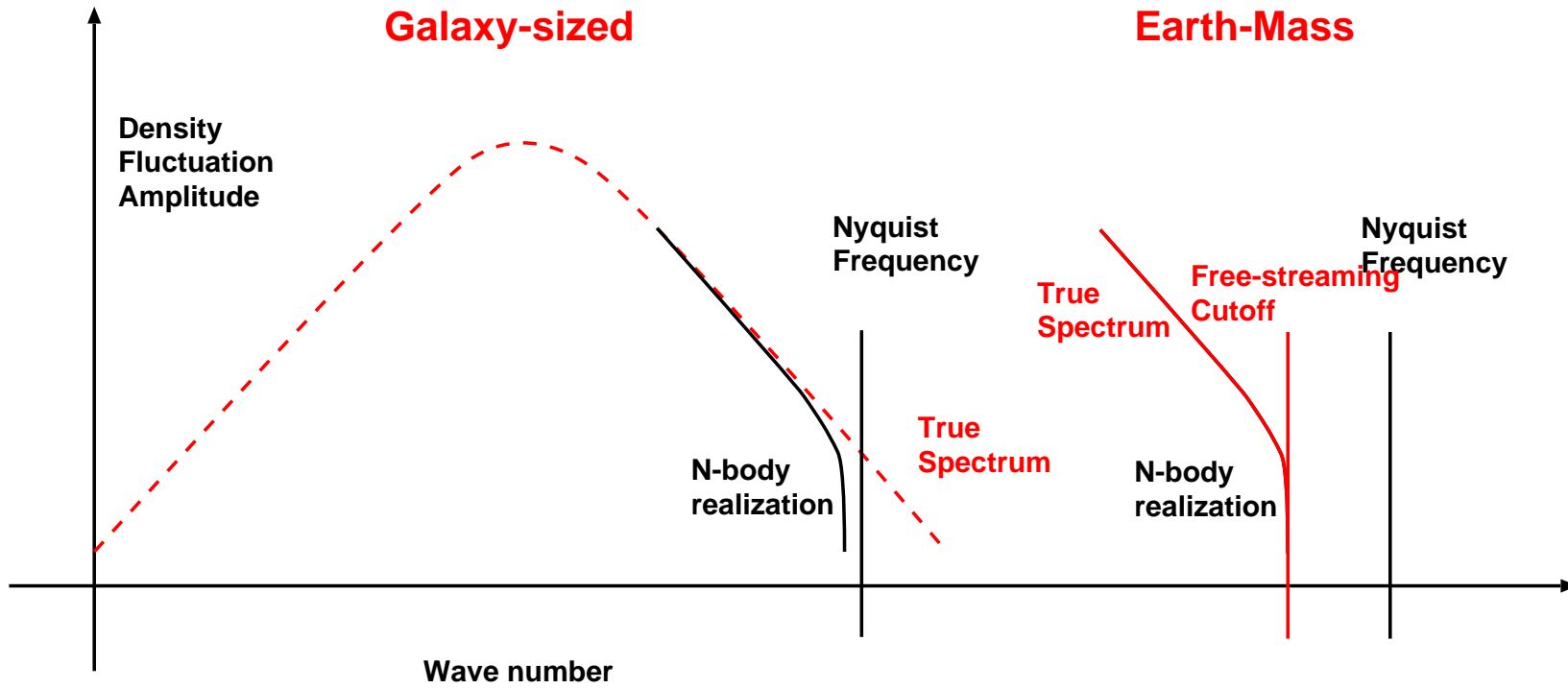
地球サイズと銀河サイズの違い

- 銀河サイズ: もっと小さい波長のゆらぎが中に一杯ある:
サブ構造あり
- 地球サイズ: 定義によりそういうサブ構造のない、最小構造 (Free-streaming cutoff: No substructures)

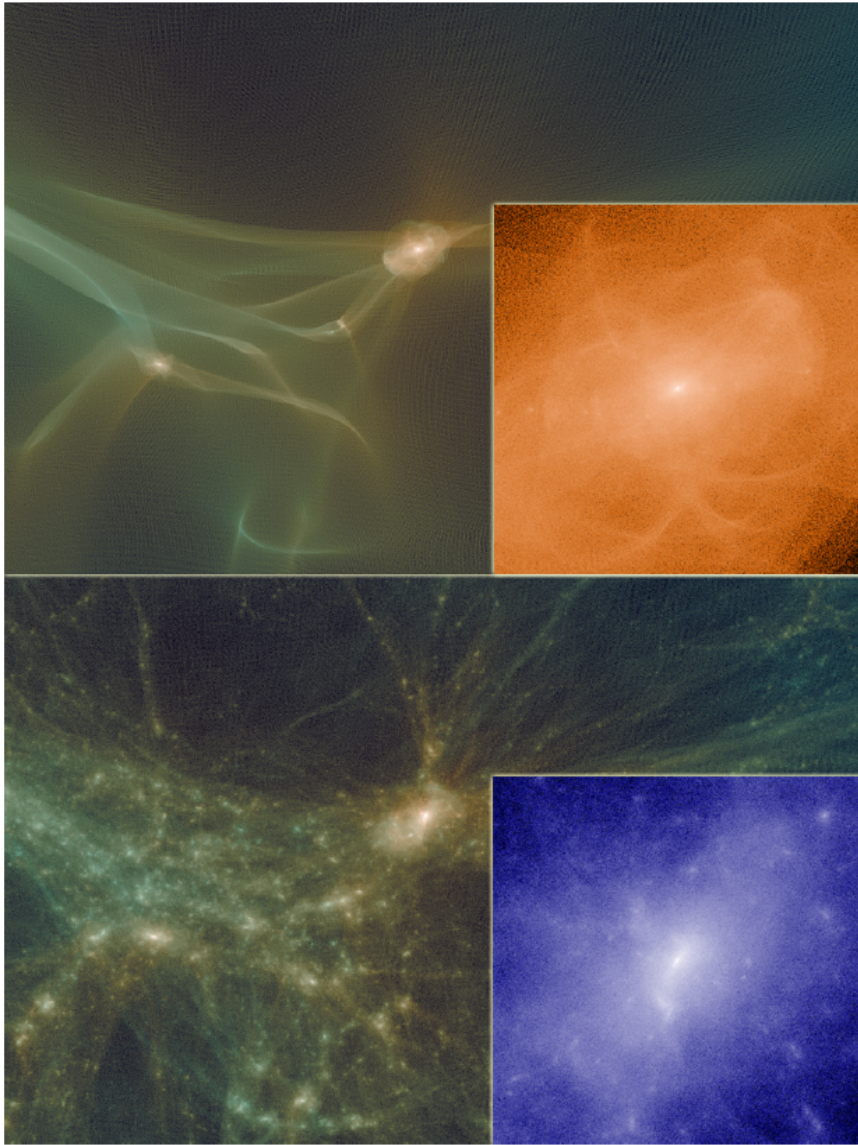
問題:

- 地球サイズのハローは本当はどんな構造か？
- それが成長していくとどうなるか？

Initial condition



我々の(ちょっと前の)計算



Ishiyama et al., 2010

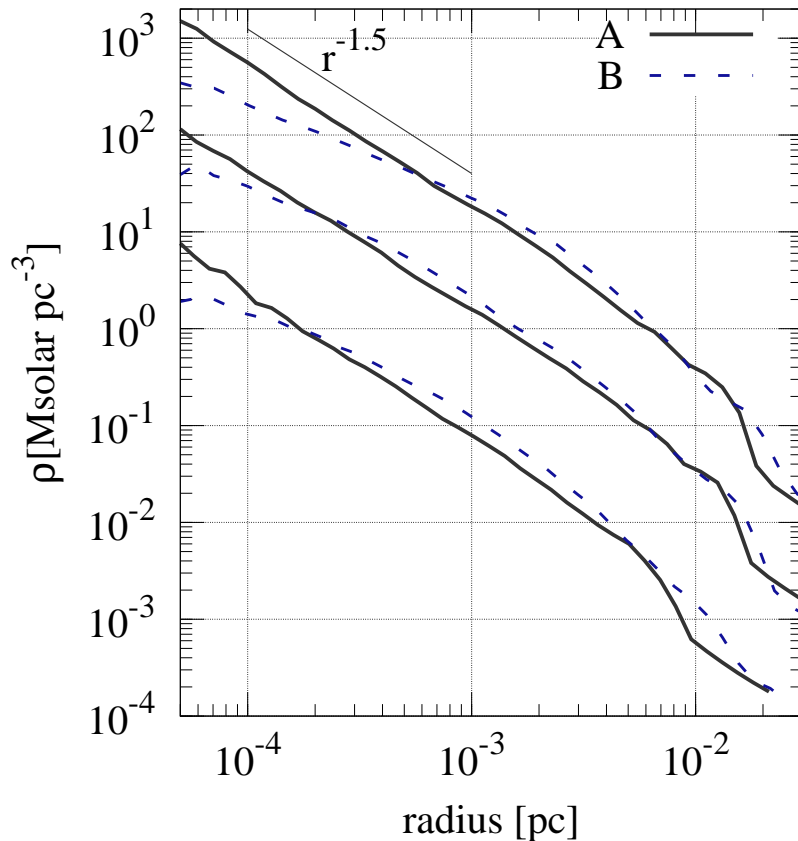
国立天文台の Cray XT4。

「京」の 1/500 の能力

10億粒子くらい

- 上: 地球質量くらいに対応する計算。
- 下: もっと大きなサイズに対応する計算。

密度分布



実線: 地球質量に対応する計算。中心部で -1.5 乗の密度べき

波線: もっと大きなスケールに対応。中心密度低い

密度が中心で高い
=ダークマター粒子同士の衝突が起こりやすい

地球質量八ローの中心部が 線で光っている?

「京」でやろうとしていること

- 理想的には、地球質量のダークマター天体が集まって銀河くらいになるところまで
- これはまだ無理だが、可能な限り大規模な計算をいくつかやって、もっと先を予測。



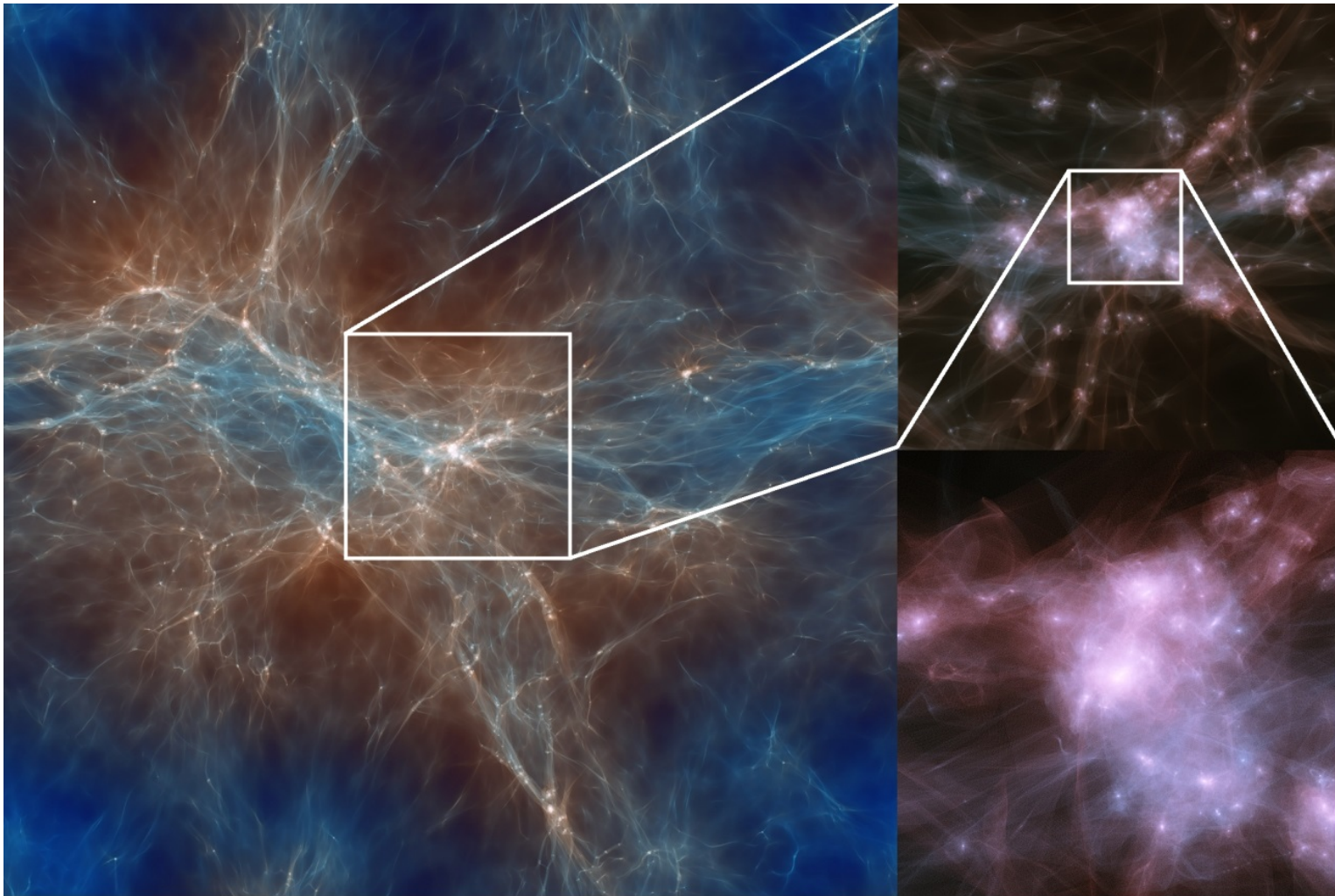
我々の周りのダークマターの分布について、理論に基づいた定量的な予言

計算例: 168億粒子の計算

アニメーション

「京」の全能力を使うとあと100倍程度大きな計算

640億粒子の計算



640億粒子の計算:密度プロファイル

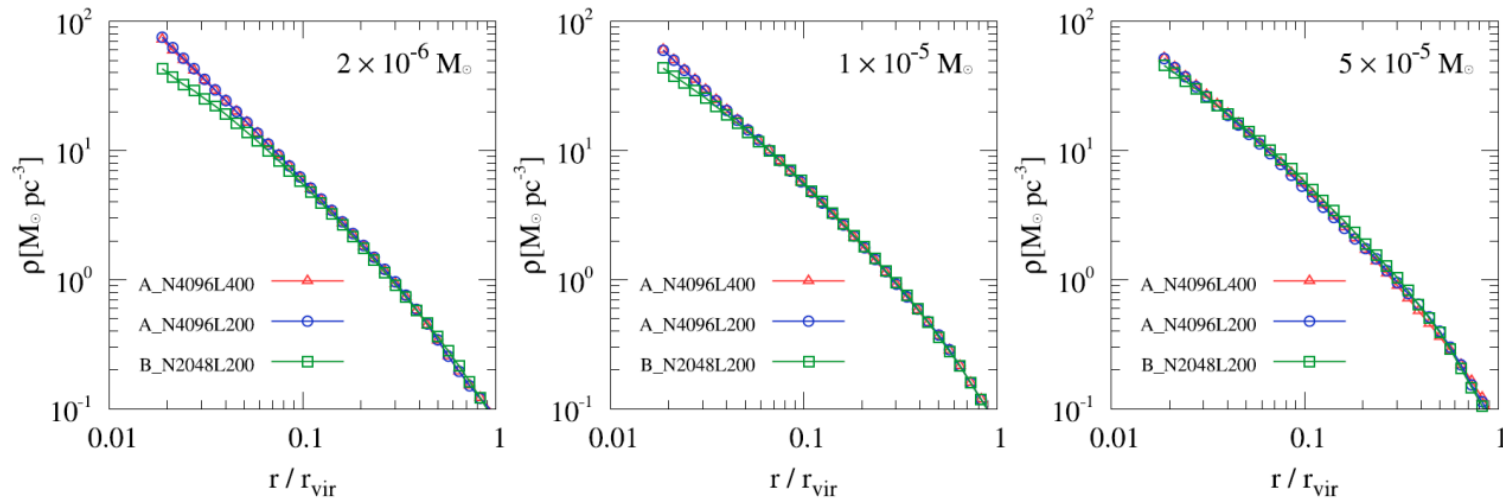
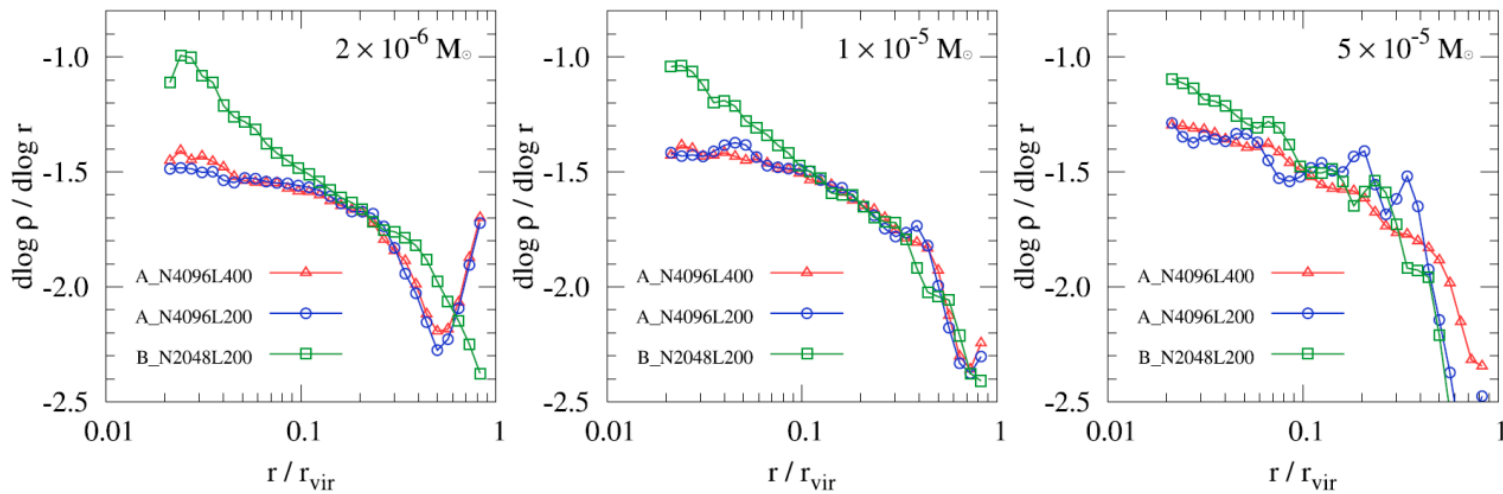


Figure 3. Stacked radial density profiles at $z = 32$ for three simulations as a function of the normalized radius by the virial radius.



結果の解釈

- 何がおこっていきそうか？
 - ハロー質量が大きいほど中心の密度カスプは浅くなる
- そうなる理屈は？
 - まだよくわからない。
 - 定性的には「当然」だが、、、

何故「当然」と考えるか？

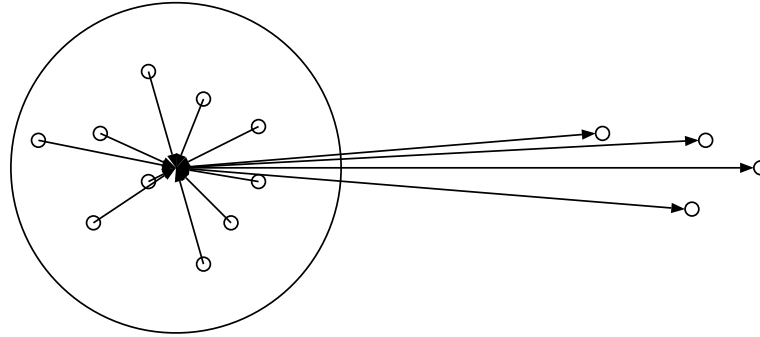
- 初期条件のパワースペクトルから：できたハローの平均密度は質量にあまり依存しない
- 中心部の最大密度、最小半径 (その辺から密度一定になるところ)：ダークマター粒子の質量でほぼ決まり、ハロー質量によらない

しかし

- その間がどうつながっているかはまだよくわからない

計算方法等について少しだけ

遠距離重力の評価:ツリー法

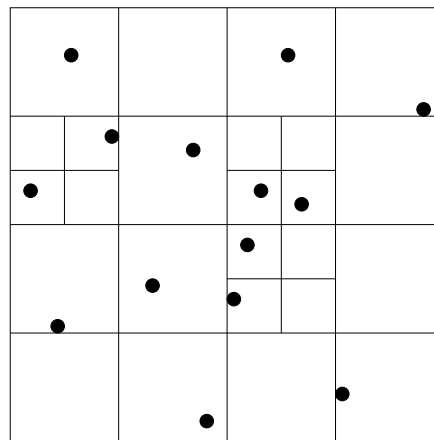


遠くの粒子
からの力は
弱い

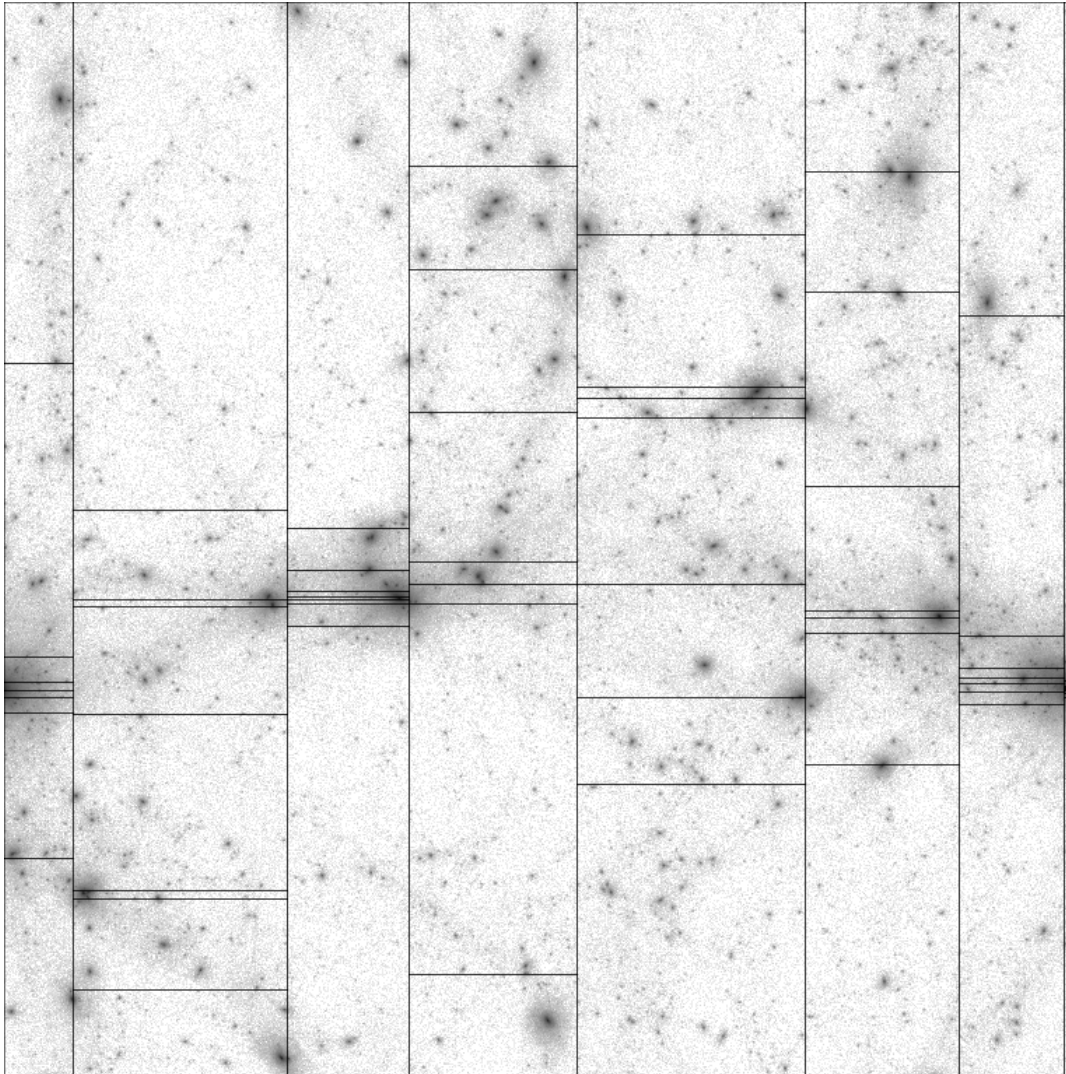
ツリー構造:



まとめて
計算



どうやって「京」を使うか？



空間分割して計算
ノードに割り当て
Recursive Multi-
section (JM 2004)

「京」の Tofu ネット
ワークに適した方法

計算時間が均等にな
るよう領域サイズ
調整 (石山他 2009、
2012)

チューニング

「京」を使いこなすには色々な努力も必要

- 粒子間相互作用計算カーネルの最適化。
 - 区分多項式の SIMD 向け変形、ビルトイン関数で SIMD データ型を直接利用
 - 最適な命令列ができるようなコードをスクリプトで生成 (アンロール回数等)
 - コンパイラによるアンロールも適用
 - メモリアクセスを減らすため、1 粒子から複数粒子への力を並列計算する形にアルゴリズム変更 (これは従来から行われている)
- 通信の工夫

実現できた性能

- 「京」全ノード使った 2兆粒子の計算で、5.67 ペタフロップス (ピークの 55%)
- 1秒に 400億粒子アップデート (2兆粒子 1ステップ 50秒)
- 20PF の BG/Q での同様な計算 (Habib et al. 2012): 14ペタフロップスだけど 1秒に 160億粒子アップデート (2兆粒子 1ステップ 120秒)
- 我々の「京」での計算のほうが、実質 2.4倍速い

2012年ゴードンベル賞獲得!!

2011年に続いて「京」での計算は2年連続でゴードンベル賞

何故こんなに性能差が？

- 理由ははっきりしている：BG/Q のグループのプログラムはしなくてもいい計算をしている
- 良くわからないこと：何故しなくてもいいことをしているのか？
 - 我々が(私じゃなくて実際にプログラム開発した石山さん、似鳥さんが)偉かった？
 - BG/Q に対しては彼らのがベスト？
 - それ以外のなにか？

多分このへんは日本が進んでいる、といってもいいのではないかと思う

今後の方向

- ダークマターだけでなく、普通の物質も計算にいれて銀河の形成・進化を
- さらに、以下のような方向も
 - 恒星の形成過程、惑星形成過程の解明
 - 生命が発生できる惑星はできるか？
- 計算機的能力向上に期待
- 計算方法の改良も必要

他分野への応用

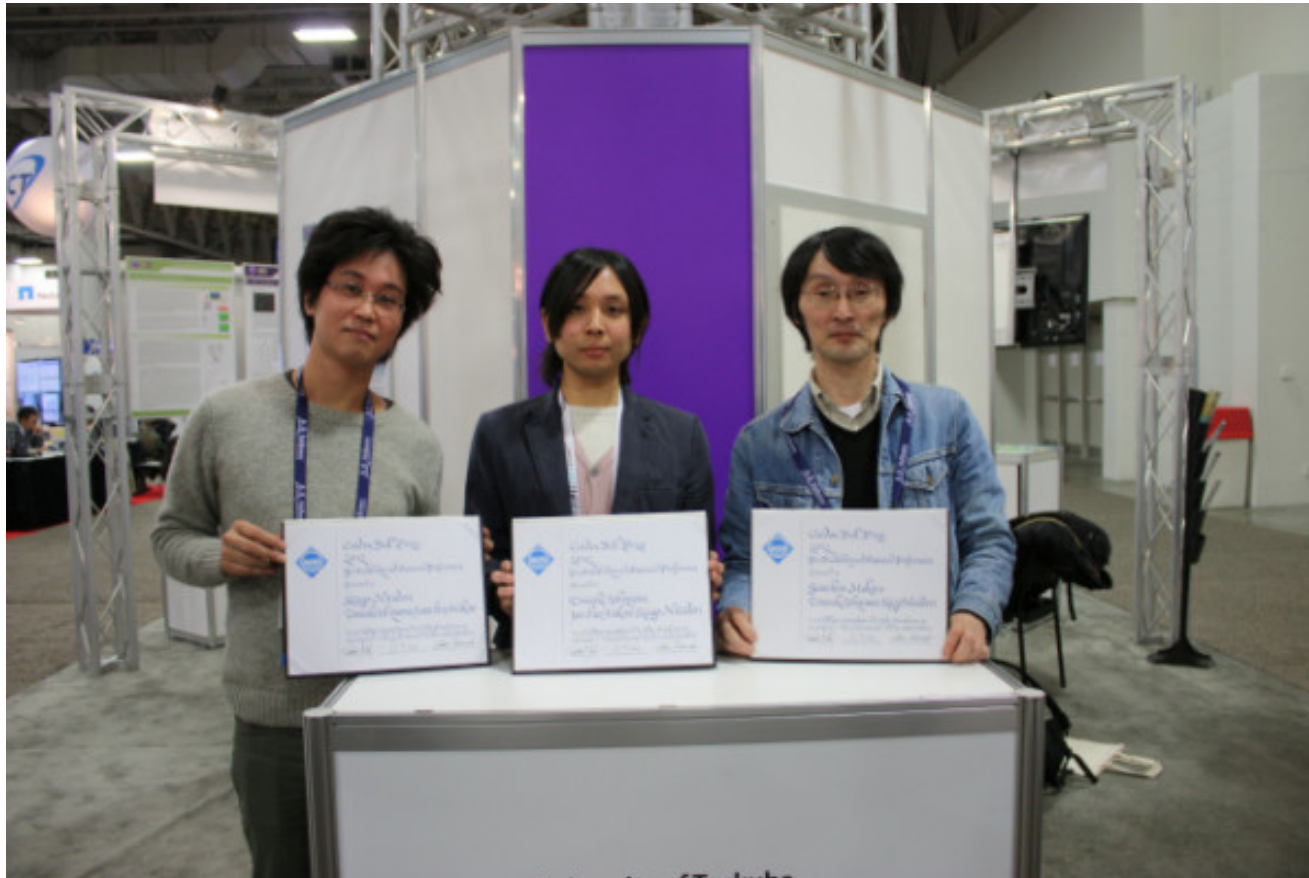
- 現在のコードで並列化、ロードバランスに使っているアルゴリズムは重力でなくても、粒子系なら汎用に近い。
- 分子動力学、粒子による流体や連続体力学、エージェントシミュレーション等にも応用可能？
- そのための汎用並列化プラットフォーム開発も始めている (AICS 粒子シミュレータ開発チーム)

まとめ

- 我々は、「京」を使って大規模シミュレーションで、銀河系の中のダークマター粒子の分布、どのように観測されるかを予言することを目指している
- このため、「京」の全能力を発揮できる、非常に高性能な計算コードを開発した
- その結果、演算性能では「京」の2倍ある BG/Q での米国のグループの計算の 2.4 倍の性能を実現し、2012年ゴードンベル賞も獲得できた
- 日本はソフトウェアが弱い、と言われるがそうでもないところもある
- 「京」でのサイエンス、このソフトウェア技術の他の分野への応用にも期待して欲しい

最後に

「京」での実際にのコード開発、計算は、石山さん(筑波大)、似鳥さん(AICS) によるものです。



似鳥さん、石山さん、牧野 @SC12 つくばCCSブース

エクサスケールシステムに向けて — 分散メモリ超並列アーキテクチャの復活

牧野淳一郎

東工大 地球生命研究所
理研 計算科学研究機構 粒子系シミュレータチーム

話の構成

- スパコンの進化:: 1950-2010
- 現状の問題
 - － 消費電力
 - － 並列処理オーバーヘッド
 - － ソフトウェアの開発/メンテ
- 解決？
- 日本の「ポスト「京」」

スパコンの進化: 1940-2000

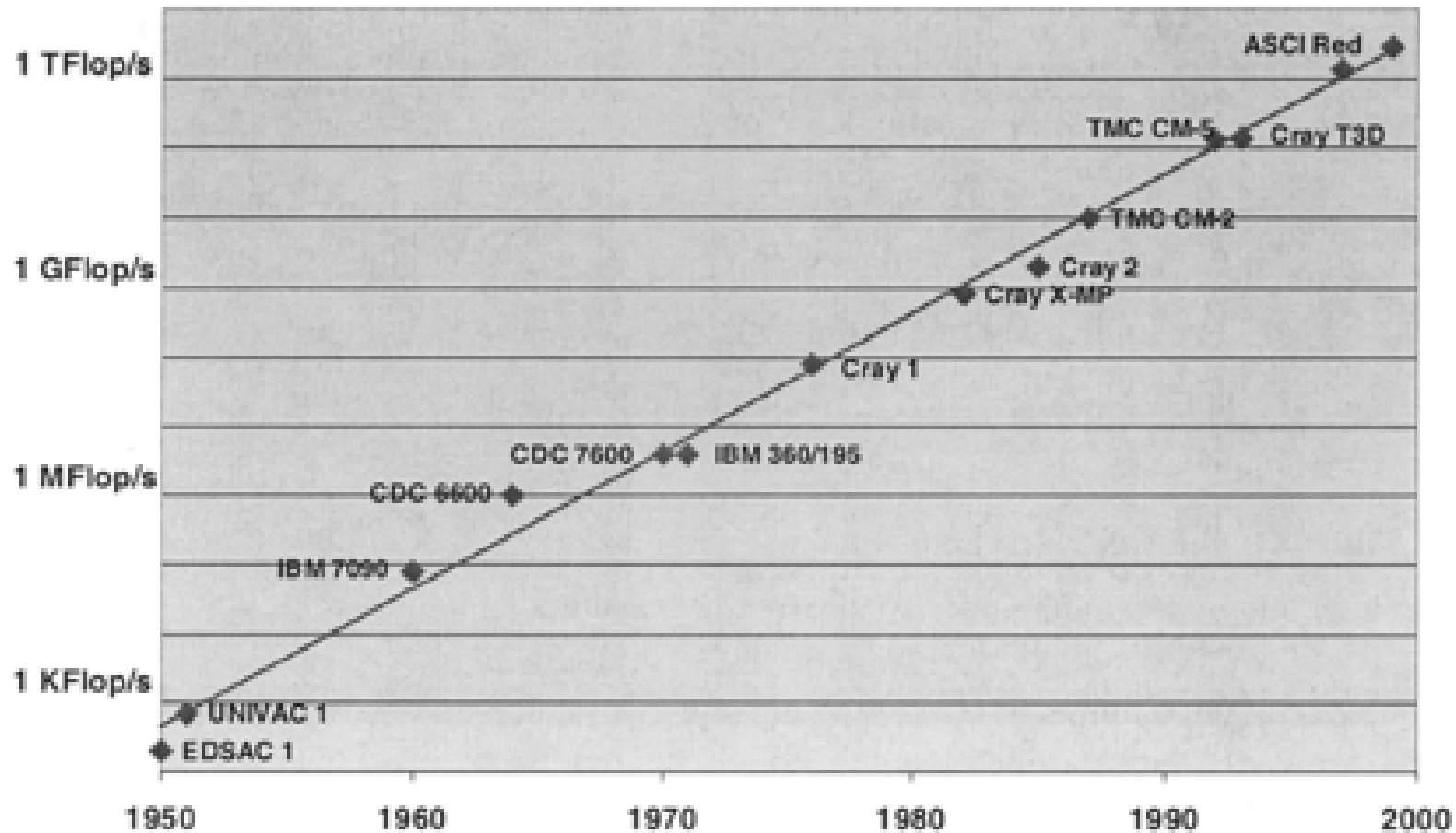
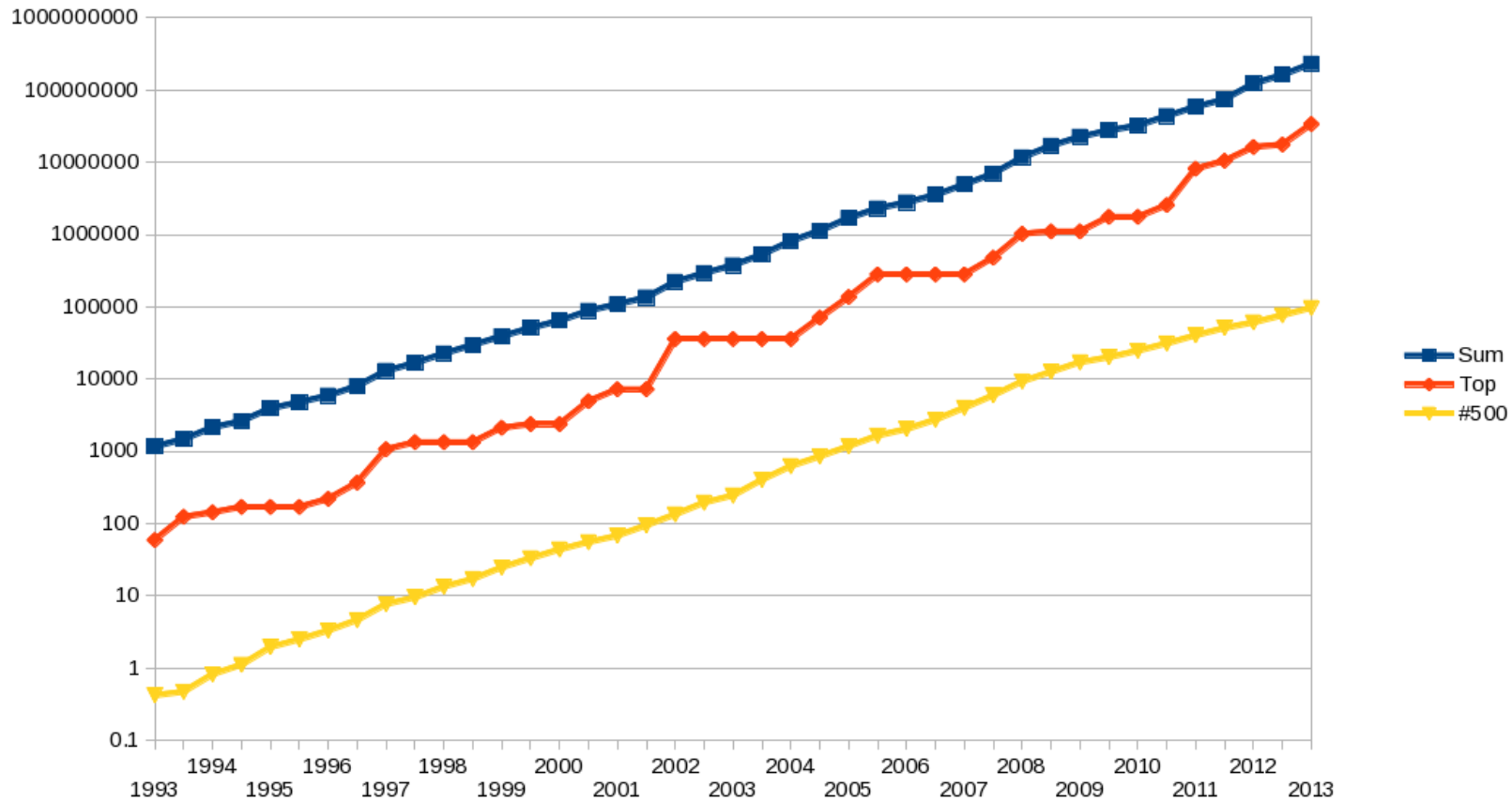


Figure 1. Moore's law and past performance of various "supercomputers" over time.

1940-2000: 10年に100倍

スパコンの進化: 1993-2013

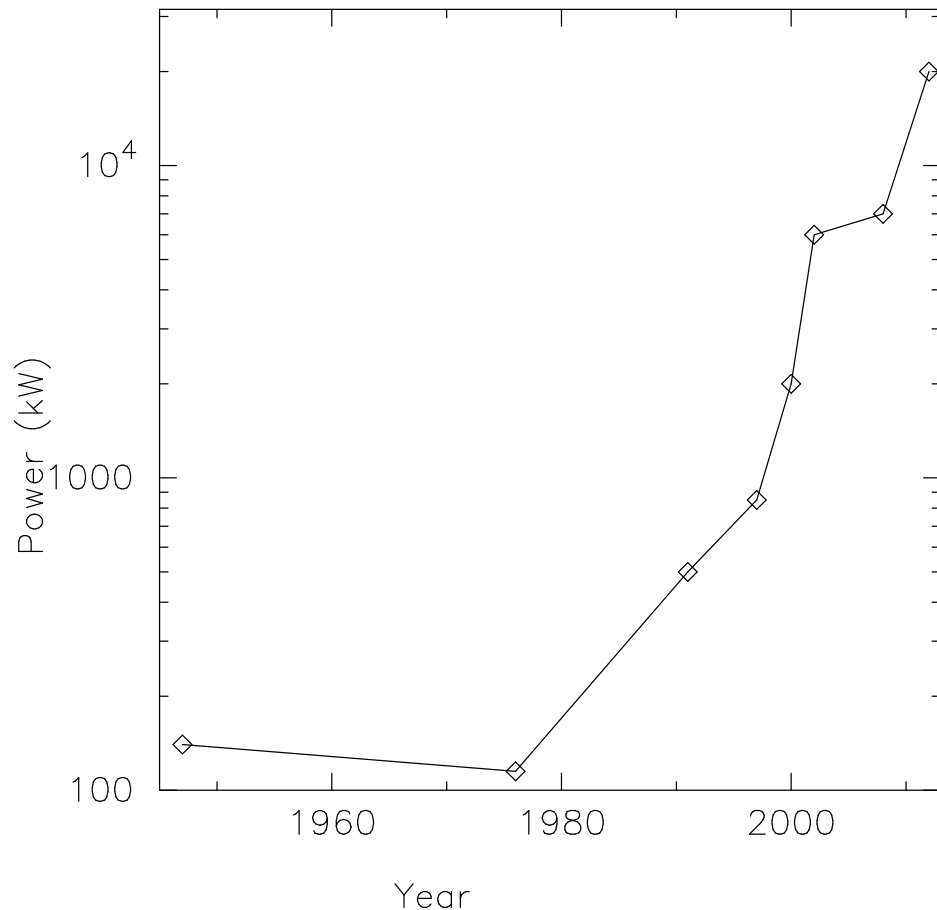


1993-2013: 10年に500倍!?

問題1: 消費電力

ENIAC	1947	140kW
Cray-1	1976	115kW
Cray C90	1991	500kW
ASCI Red	1997	850kW
ASCI White	2000	2MW
ES	2002	6MW
ORNL XT5	2008	7MW
「京」	2012	20MW

グラフにすると.....



ENIACから Cray-1 ま
ではたいして変わらない

1975-95 の20年間に
10倍

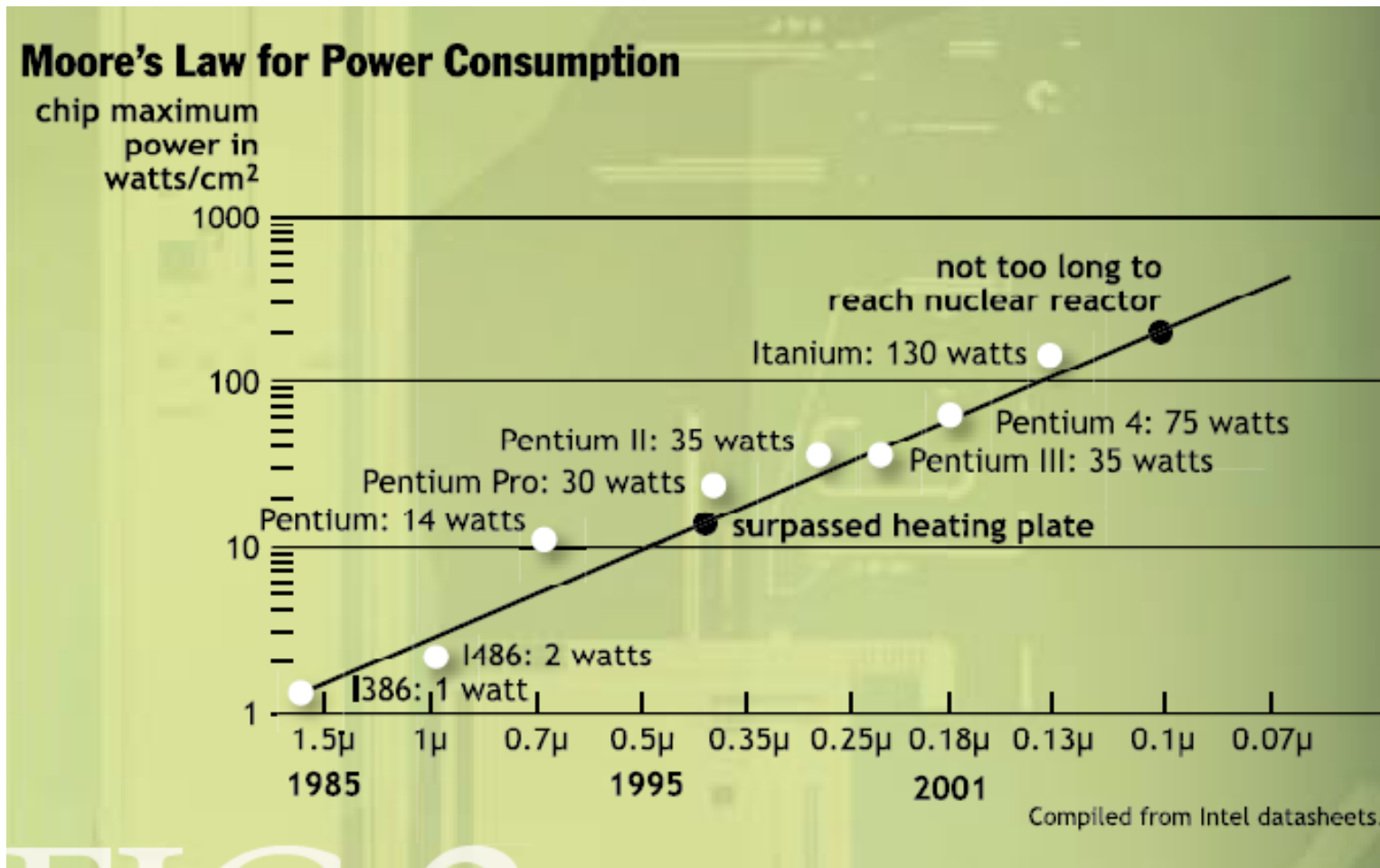
1995-2012 の17年間に
30倍

指数関数より速い増加 (有限時間でクラッシュ)

何故？

- 沢山お金掛けるようになった:: ASCII Red: 50億, 「京」
×百億
- チップあたり(ないし面積あたり)の消費電力増加
- チップ面積あたりの価格低下

シリコン面積あたりの消費電力



2003 年に限界に到達、それ以上増えてない (BTX の失敗)

問題 2: 並列化オーバーヘッド

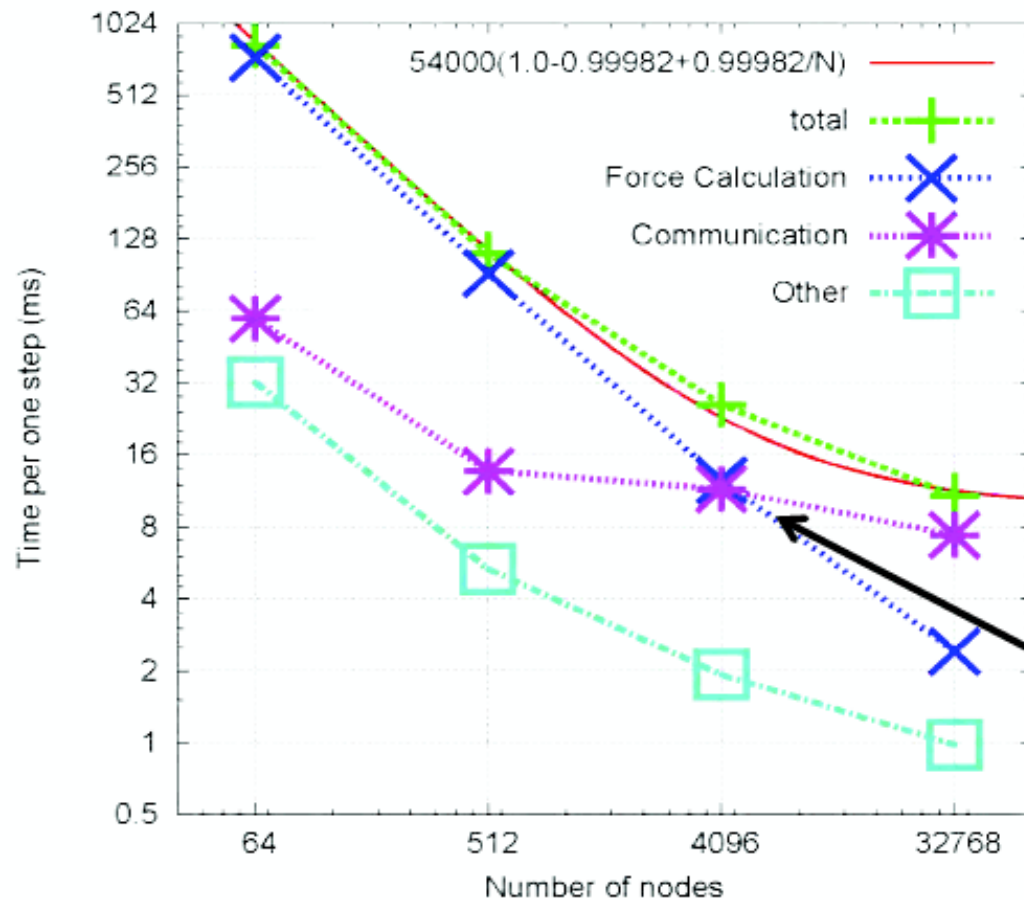
浮動小数点演算器の数 (乗算+加算で1つ)

Cray-1	1976	1
Cray C90	1991	16
ASCI White	2000	16,384
地球シミュレータ	2002	40,960
「京」	2012	2,820,096

「京」は、大自由度系の短時間計算はできて、小(といっても100万とか、、)自由度系の長時間計算にはむかない

性能スケーラビリティの例

Strong Scaling (内訳)



Cutoff 28Å,
3,349,656 atom ,
calculate energy
every 4 step

Cross over at
817 atom/node

「京」での分子動力学計算

- 1 タイムステップが 5ms を切らない
- 通信オーバーヘッドが問題

5ms で十分速いか？

- タンパクとかだとマイクロ秒くらいしか計算できない
- 専用機 ANTON は 100 倍以上速い

現在のアーキテクチャの延長では大きな改善は難しい。

並列化オーバーヘッドの起源

演算器の数自体が問題なのではない

- 「通信時間」のほとんどは、メモリ読み書きのオーバーヘッド
- CPU キャッシュ メモリ NIC NIC メモリ キャッシュ CPU
- 同期や総和になるともっと大変なことに

問題 3: どうやってプログラム書くの？

- MPI
- OpenMP
- SIMD 拡張
- Cache の有効利用
- アクセラレータ
- ...
- ...

解決の方向は？

- 消費電力と通信オーバーヘッドの両方を減らしたい
- メモリはそんなにいらぬ(という問題も多い)。100万原子:100MB。1兆になっても 100TB。タイムステップが少し多いと1兆粒子はエクサでも終わらぬ。

一つの方向:

- 「小さな」オンチップメモリしかもたぬプロセッサチップ(「小さなといっても256MB とか、、、)
- 単純なコアを多数 SIMD で動かすことで、同期、通信のオーバーヘッドを減らす。

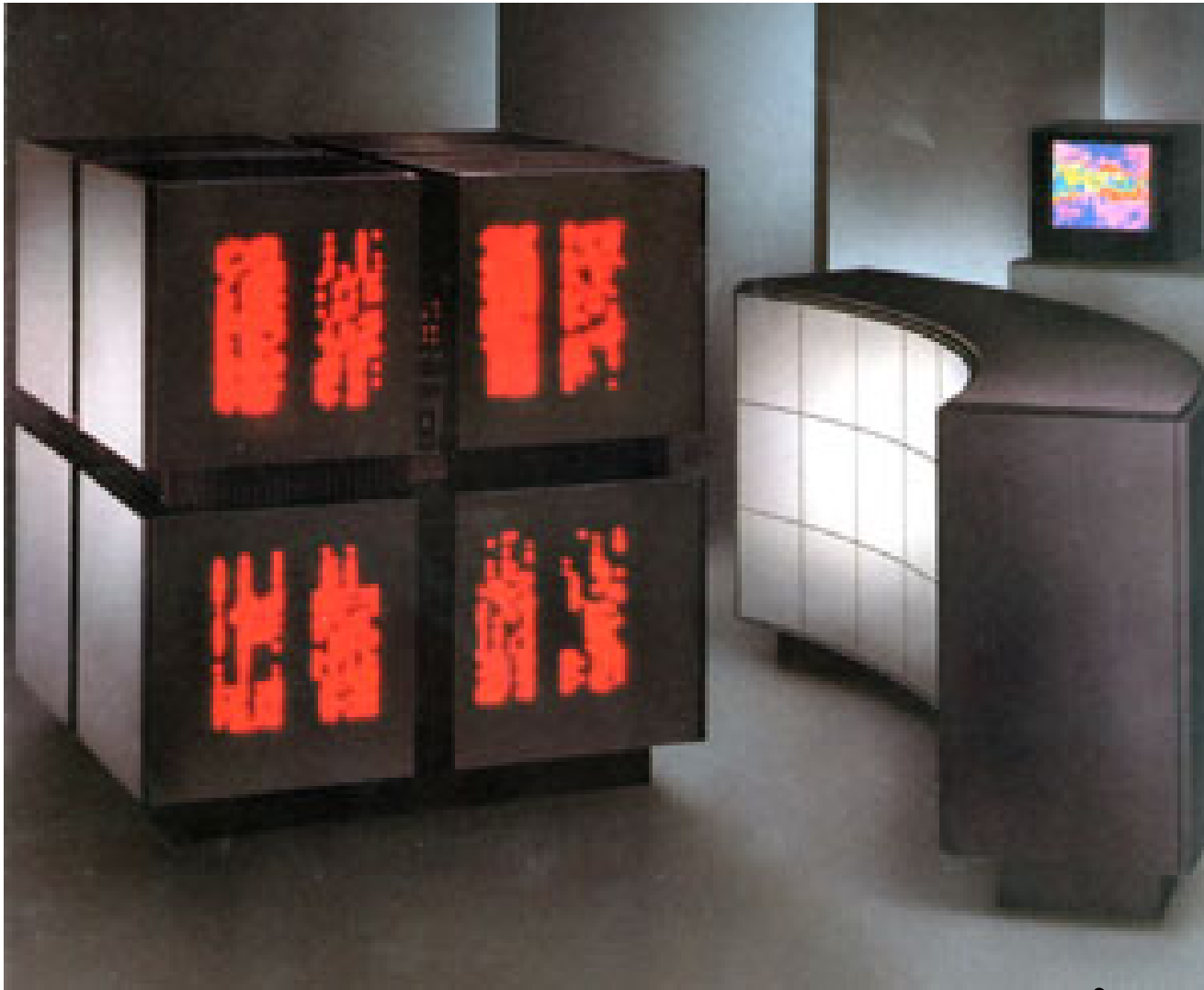
超並列SIMD計算機

— A lost technology —

- Goodyear MPP (1970s)
- ICL DAP (Late 1970s)
- Thinking Machines Connection Machine-1/2 (Late 1980s)
- Maspar MP-1/2 (Early 1990s)

CM-2 はそれなりに成功だった

TMC CM-2



2048 個の Weitek 浮動小数点演算チップセットを SIMD で動かす

TMC CM-2

- 64k個の 1-bit プロセッサ。メモリ量 64kビット
- 2048 個の浮動小数点演算器。1つが32プロセッサにシェアされる
- 12次元ハイパーキューブネットワーク (16プロセッサが1チップ)

現代 (というかちょっと未来) の半導体技術なら、4-8台くらいの CM-2 を1チップに置いて、100倍のクロックで動作させ、10-20 Tflops くらいを消費電力100W 以下で実現可能

CM-2 ソフトウェア

- *Lisp: データ並列 Lisp
- C*: C++ で実装したデータ並列 C
- CM-Fortran (ほぼ HPF)

もちろん「天才」 Guy Steele がいたからできた話ではある。

私のハードディスクから発掘された C* コードの残骸

```
typedef domain node_domain {
    EXTERN int index;          /* index for particle/node */
    EXTERN REAL mass;         /* mass of the node */
    EXTERN REAL position[NDIM]; /* position */
    EXTERN REAL velocity[NDIM]; /* velocity */
    EXTERN REAL acc[NDIM];     /* acceleration */
    EXTERN REAL acc_old[NDIM]; /* acceleration of previous step*/
    EXTERN REAL potential;     /* current potential of the particle */
    EXTERN REAL out_potential;
    EXTERN REAL r_work[NDIM+1];
} NODE_DOM, *NODE_PTR;
```

C* コードの残骸の続き

```
void node_domain::calc_accel()
{
    int myindex;
    mono int i,k;
    REAL dx[NDIM];
    acc[0] = acc[1] = acc[2] = 0.0;
    potential = 0.0;
    myindex = (int)this - (int)&node[0];
    if(this && this < &node[nbody]){
        for(i = 0; i < nbody; i++){
            REAL rsq;
            REAL pot, rsqinv, rinv;
            if(myindex != i){
                rsq = eps2;
                for (k = 0; k < NDIM; k++){
                    dx[k]=node[i].position[k]
                        -position[k];
                    rsq += dx[k]*dx[k];
                }
                rsqinv = 1.0/rsq;
                rinv = sqrt(rsqinv);
                pot=node[i].mass *rinv;
                potential+=pot;
                pot *=rsqinv;
                for (k=0; k<NDIM; k++) {
                    acc[k]+=pot*dx[k];
                }
            }
        }
    }
}
```

いいとこ

- 「データ並列」を表現。通信/プロセッサ内データの切り分け、データレイアウトその他はコンパイラ+ランタイムが面倒みてくれる
- 通信は単に配列への間接アクセスで書ける
- 実際に相当複雑な処理が書ける。Barnes-Hut ツリー:構築、粒子毎にバラバラにツリー探索、といったことも。

汎用コアの SIMD との違い: メモリが独立、独立な範囲内ではランダムアクセスできる。

実際には

コンパイラがバグばかりだったのでこんなの書く羽目に

```
for(k=0; k<4; k++){
    host_work[k] = CM_u_read_from_processor_1L(&node[i],
        &r_work[k], REAL_LEN);
    CM_u_move_constant_1L(&r_work2[k], host_work[k], REAL_LEN);
}
rsq = eps2;
for (k = 0; k < NDIM; k++){
    CM_f_subtract_3_1L(&dx[k], &r_work2[k], &position[k],
        SIG_LEN, EXP_LEN);
    CM_f_mult_add_1L(&rsq, &dx[k], &dx[k], &rsq,
        SIG_LEN, EXP_LEN);
}
CM_f_divinto_constant_3_1L(&rsqinv, &rsq, 1.0,
    SIG_LEN, EXP_LEN);
CM_f_sqrt(&rinv, &rsqinv, SIG_LEN, EXP_LEN);
CM_move(&pot, &r_work2[NDIM], REAL_LEN);
CM_f_multiply(&pot, &rinv, SIG_LEN, EXP_LEN);
```

```
CM_f_add(&potential, &pot, SIG_LEN, EXP_LEN);
CM_f_multiply(&pot, &rsqinv, SIG_LEN, EXP_LEN);
for (k=0; k<NDIM; k++) {
    CM_f_mult_add_1L(&acc[k], &dx[k], &pot, &acc[k],
        SIG_LEN, EXP_LEN);
}
}
```


非構造格子？

- 非構造疎行列だって書くのは難しくない。全節点でデータ並列動作。ポインタでアクセスすればPE間通信でデータ取ってくる。
- 性能は実は結構でる： 実はほとんどのアクセスは PE メモリ内ですむ。(ように節点番号ふって欲しいな) メモリバンド幅は高い(しアクセス粒度も小さい)から。

汎用スカラー並列に比べてどう改善？

- 消費電力:
 - データ移動が減る。キャッシュ階層がなく、チップ外メモリも(第一義的には)ない
 - 命令フェッチ・デコードのコストも減っている。1チップに1ユニット。
- 通信オーバーヘッド
 - データ移動が減る。外部メモリ、キャッシュ階層がない
 - ハンドシェイク、同期のオーバーヘッドも減る。SIMDで動いている範囲では始めから同期しているので同期操作不要。細粒度通信が可能。
 - チップ間はまだちょっと考える必要がある。データフローマシンの動作をさせたい

SIMD 超並列機のネットワーク

ポスト「京」の FS で一応「検討中」あんまりまだ考えてないですが、、、以下は検討中の1例

- チップ内: 64 コア程度をクロスバー結合したものが基本ユニット。ユニット間は多段ネットワーク
- チップ間: 16 チップ程度をルータにつなぐ。ルータ間は色々できるように作るがポスト「京」むけは 4D トーラスの通信パターンで性能できるように、、、
- このネットワークでつなぐのは 2048-4096 チップ程度。
- ルータの upward link は総バンド幅で 400GB/s 程度

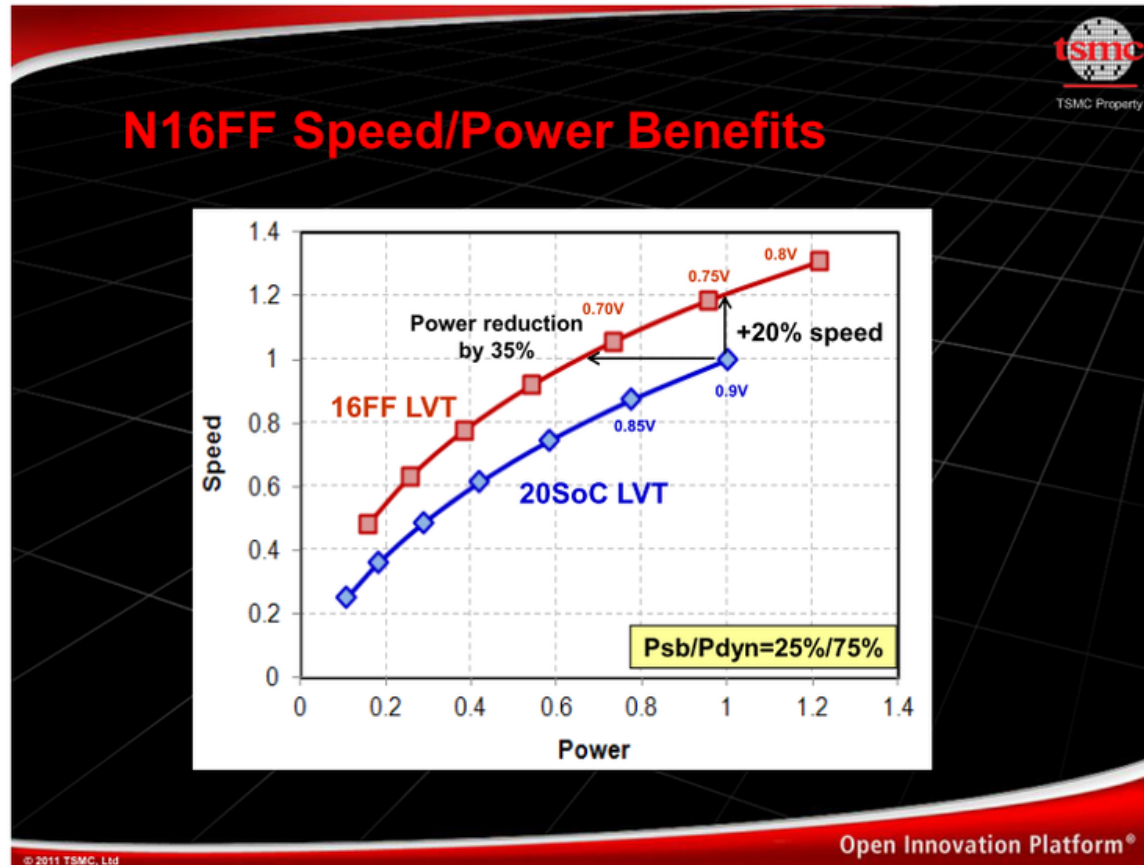
脳の話だと、、、

- ニューロン 10^{11} 個
- シナプス 10^{15} 個
- シナプスの結合情報を全部オンチップでもつのはちょっと無理
- リアルタイムくらいの速さでいいなら、総バンド幅 10PB/s くらいでいいはず。これは大したことない。
- ネットワークは問題。ランダム結合なら少なくとも (データ量/バイセクションバンド幅)*平均ホップ数 くらいの時間かかる？
- ある程度でも階層化されてるなら、ネットワークはほぼモジュール毎に考えるのでいいはず。

(楽観的な) 評価例

- 昨日の五十嵐さんの例: 神経細胞 17 億。
- 1 ステップの通信は秒オーダー?
- 原理的には、1 チップが受け取るデータは 100GB くらい? オーダーとしては 1 秒スケールで受け取ることができる。
- シナプス演算を例えば 8 ビット加算 1 つ程度で表現「できるなら」、消費電力は倍精度浮動小数点演算の $1/500$ くらい。50Tops/W くらいはできる。
- 結合データを外部メモリに置くと、その読出しの電力消費ははるかに大きい。

低電圧動作 (TSMC側の主張)



低電圧動作 (TSMC側の主張)

- 16nm 0.75V に比べて 0.5V だと電力 1/6、性能 4割。電力あたり性能は2倍以上。
- これ使わない手はない、、、

ポスト「京」

The screenshot shows a Mozilla Firefox browser window displaying the NHK News Web page. The address bar shows the URL: www3.nhk.or.jp/news/html/20130508/k10014434551000.html. The page title is "京の100倍 新スパコン開発 NHKニュース". The main article is titled "京の100倍 新スパコン開発" and is dated May 8, 2013, at 17:55. The article text states that the Ministry of Education, Culture, Sports, Science and Technology aims to develop a supercomputer named 'K' that is 100 times faster than the current 'K' by next year. It also mentions that the current 'K' is the world's fastest supercomputer, but it is being surpassed by the American 'Titan'.

文部科学省は、おとし計算速度の世界一を達成したスーパーコンピューター「京」の100倍の性能を持つ、新型のスパコンの開発を来年度から始めることになりました。

日本のスーパーコンピューターは、理化学研究所などが開発した「京」がおとし計算速度で世界一となりましたが、その後はアメリカのスパコンに相次いで抜かれ、現在は世界で3番目となっています。こうしたなか、文部科学省は、来年度から「京」の100倍の処理能力を持つ新型のスパコンの開発に乗り出すことを決め、8日開か

文部科学省
2020年ごろの完成目指す

主要ニュース

- 首相 抑止力の観点で敵基地攻撃議論
- トヨタ 営業利益増加を発表
- 川口委員長解任決議案 可決の公算
- 東京 最後の同潤会アパートを解体へ
- 長嶋・松井両氏 首相と夕食会
- アジア最大規模 東京でバイオ燃料
- 大阪 「ふたりっ子」舞台の閉鎖

Facebookページはこちらか
気になるニュースは いいね! をクリック
※クリックするとNHKサイトを離れます

WEB特集

- ネット報道の新潮流(1)「まとめ」で読
- 5月7日(火)
- 米軍制敵組トップ 同行取材記
- 5月2日(木)
- 世界遺産登録へ 明暗の謎は?
- 5月1日(水)
- 鳥インフル感染確認1か月 対策は



今後のHPCI計画推進のあり方に関する検討ワーキンググループ (牧野もはいつてる)

現在の計画

- 汎用スカラーで系の後継: 2020年でもエクサには 60-80MW。
いくらなんでも大き過ぎる。
- SIMD 「加速部」をつけたす。
 - 電力を $1/2-1/3$ に
 - (加速部ネットワーク内では) 通信オーバーヘッドを1桁以上削減
 - (加速部オンチップメモリにのるなら) $B/F=4$ を実現

まとめ

- 現代の大規模スカラ-並列機は「小さい」問題にむかない
(現在の「小さい」=自由度 10^7 、2020年だと 10^9)
- この問題の解決の一つの方向: SIMD 超並列
- SIMD 超並列による加速部が今のところポスト「京」プロジェクトにはいっている。