

計算機とシミュレーション

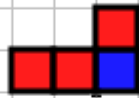
分散平行処理研究室

中里直人

LIFE GAME

LIFE GAME

- 4個のルール
 - 生きる
 - 人が2,3人いる
 - 消える
 - 人が1人以下しかいない
 - 人が4人以上いる
 - 生まれる
 - 人が3人いる



LIFE GAME

ライフゲームとシミュレーション

- 簡単なルール＝法則から複雑なモノ
- 計算機に頼ることで脳力の限界を超える
 - そして世界はひろがる
 - なんらかの世界を作り出すことができる

法則によりから複雑な現象をまねる

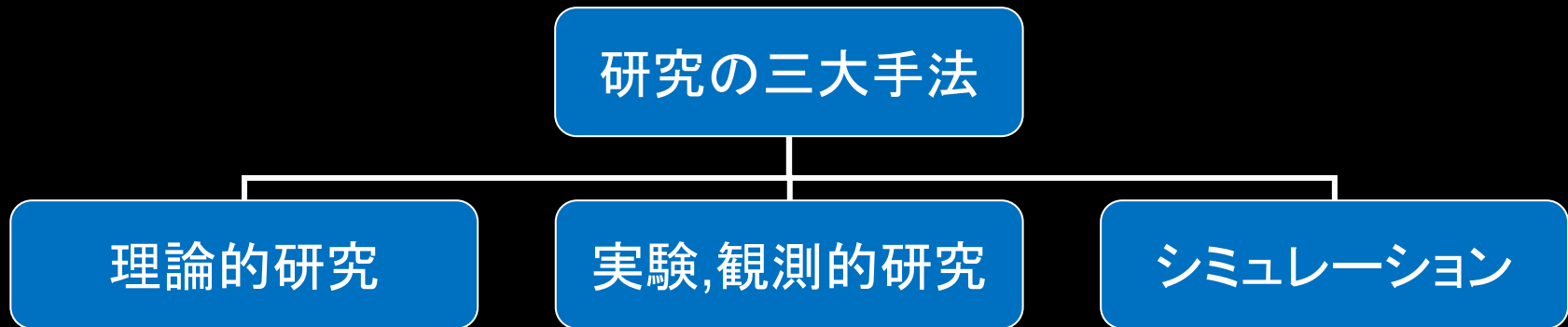
(計算機)シミュレーション

- のお仕事とは

- 世界にはどのような複雑な現象があるのだろうか？
- どういう法則が必要なのだろうか？
- その法則に基づいてなにをどう計算するのだろうか？
- シミュレーション結果は何を意味するのだろうか？

シミュレーション科学とは

- 複雑な現象を計算機内部に再現することで研究するという、科学研究の一手法
- 計算機の急速な発展により確立



リチャードソンの夢

- 1920年頃：英国人リチャードソン
 - 6時間後の大気の変化を計算した
 - 失敗1：計算に一ヶ月以上かかった
 - 失敗2：答えが完全に間違っていた



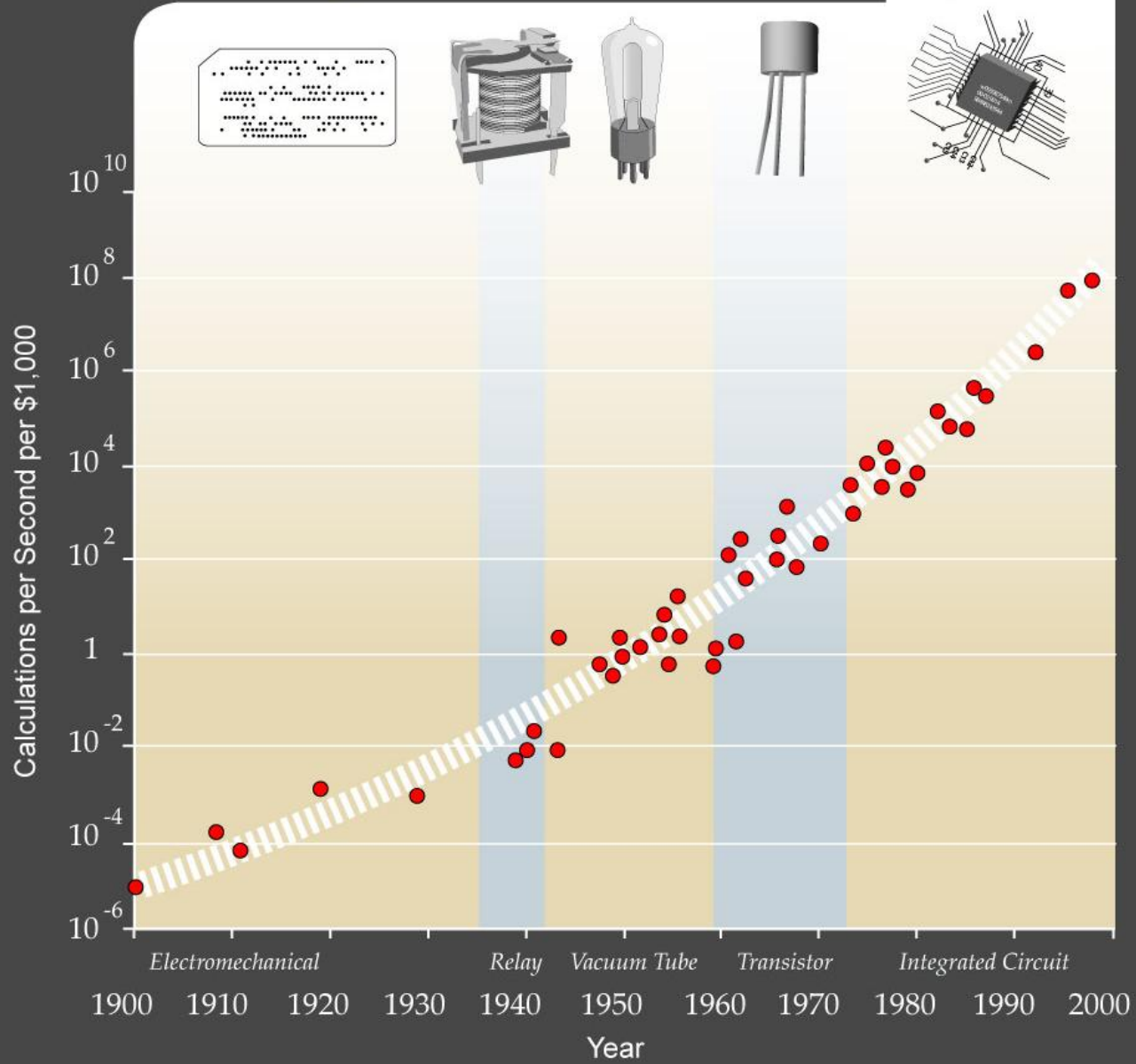
*Perhaps some day in the dim future it will be possible to advance the computations faster than the weather advances and at a cost less than the saving to mankind due to the information gained. But **that is a dream.***

Lewis Fry Richardson, 1921

Moore's Law

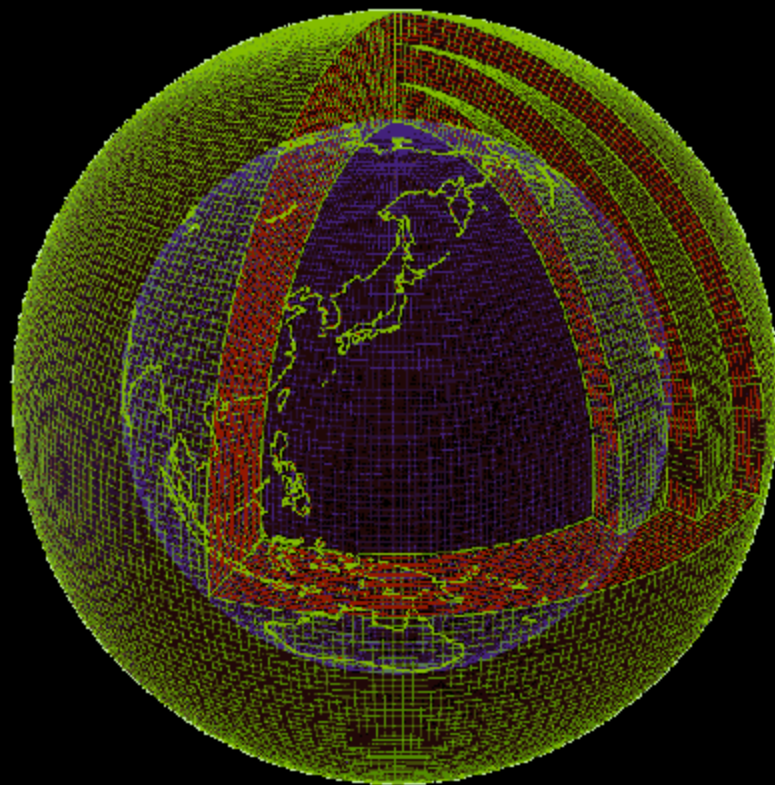
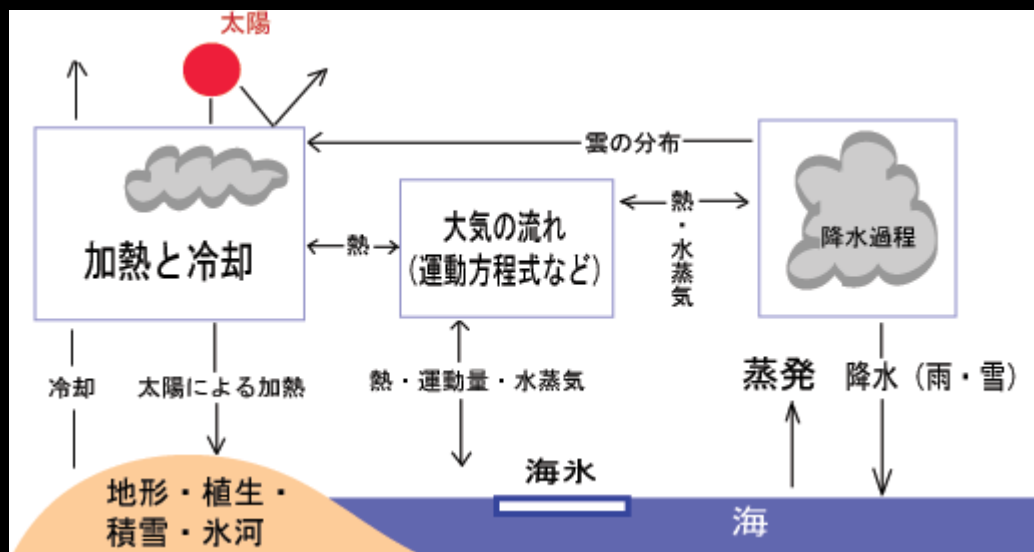
The Fifth Paradigm

Logarithmic Plot

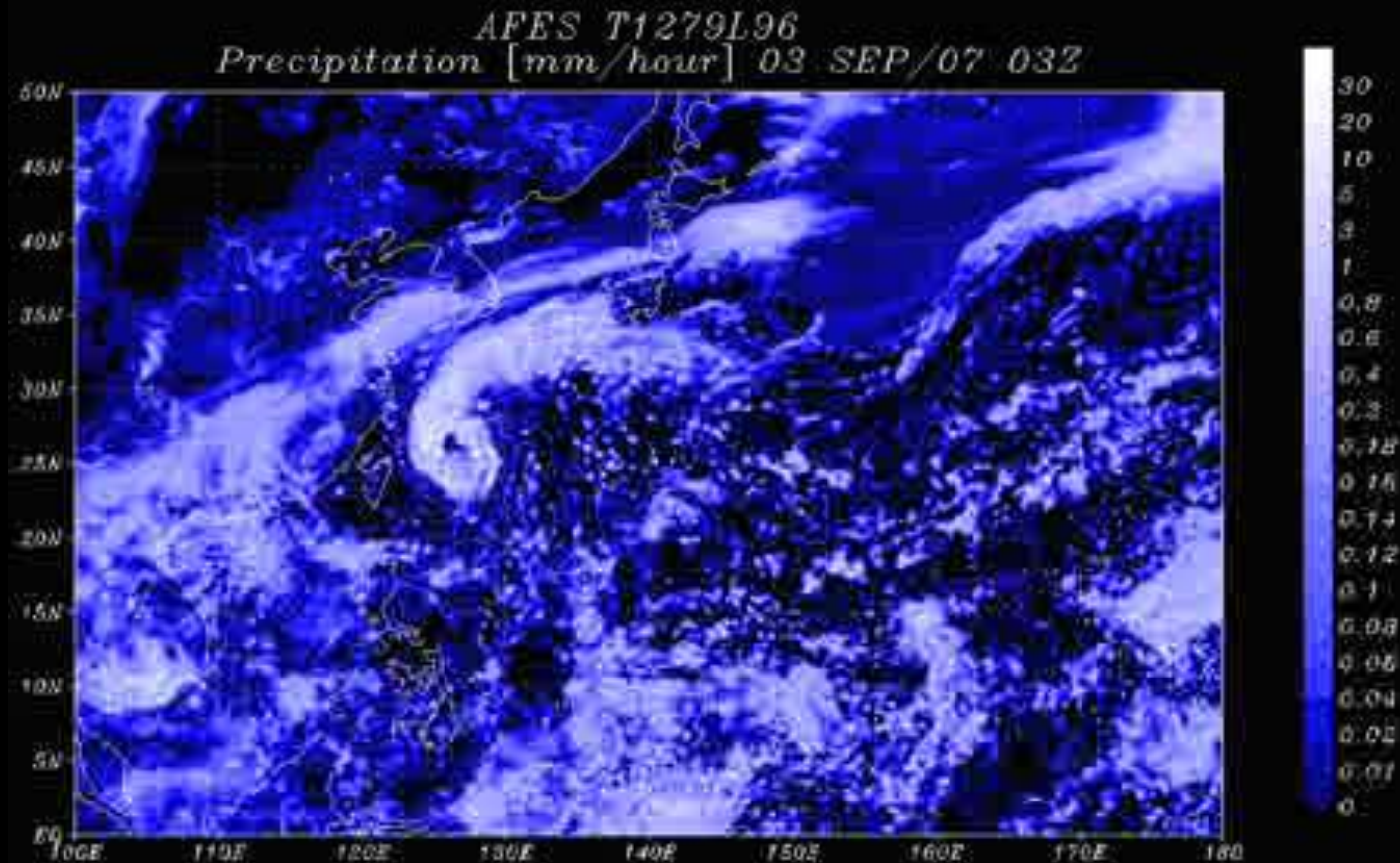


数値予報

- 気象庁のWebサイトより

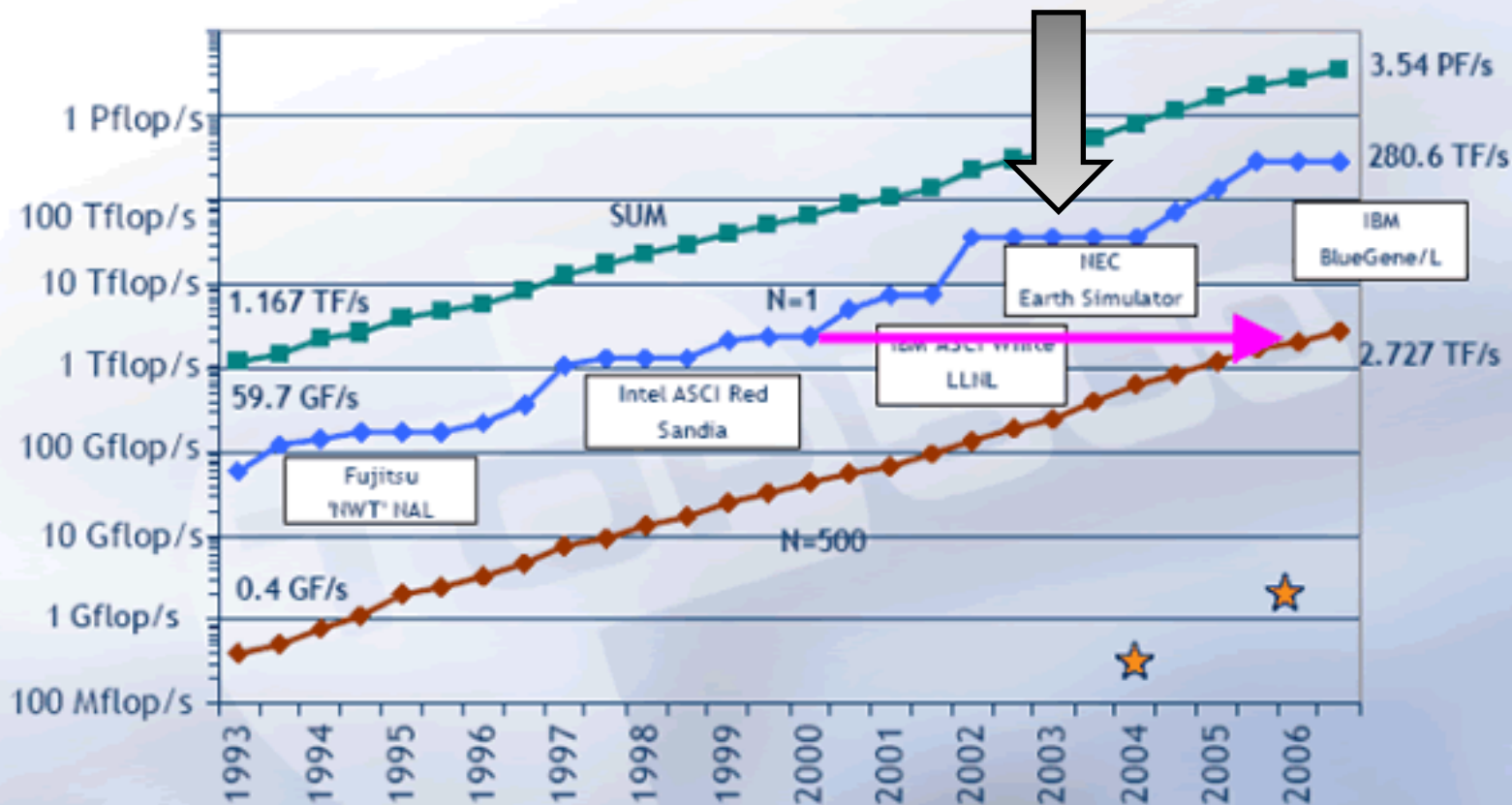


地球シミュレーターによる計算例



地球シミュレーター：日本にあるスーパーコンピューター

Performance Development



Richardson's Forecast Factory

- 何が悪かったのか後知恵で考えると
 - 世界にはどのような複雑な現象があるのだろうか？
 - 大気の状態という複雑な現象
 - どういう法則が必要なのだろうか？
 - 大気の状態の進化の法則化＝方程式(モデル)化
 - 法則に基づいてなにをどう計算するのだろうか？
 - 微分方程式をどのように数値的に解くのか
 - シミュレーション結果は何を意味するのだろうか？
 - 計算時間がかかりすぎたので検証できず

Richardson's Forecast Factory



それから90年経ちました



©JAMSTEC



Image courtesy of the National Center for Computational Sciences, Oak Ridge National Laboratory

それから90年経ちました

(機械式計算機 + 脳力)の集まり

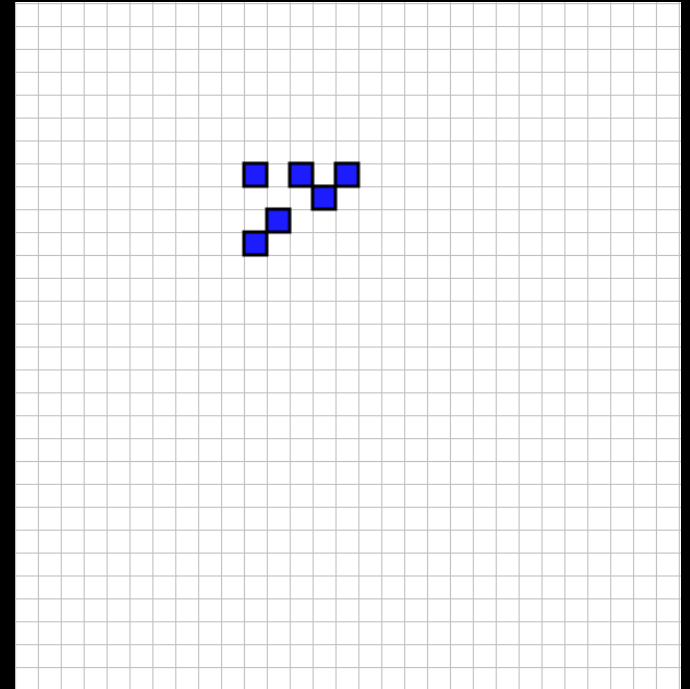
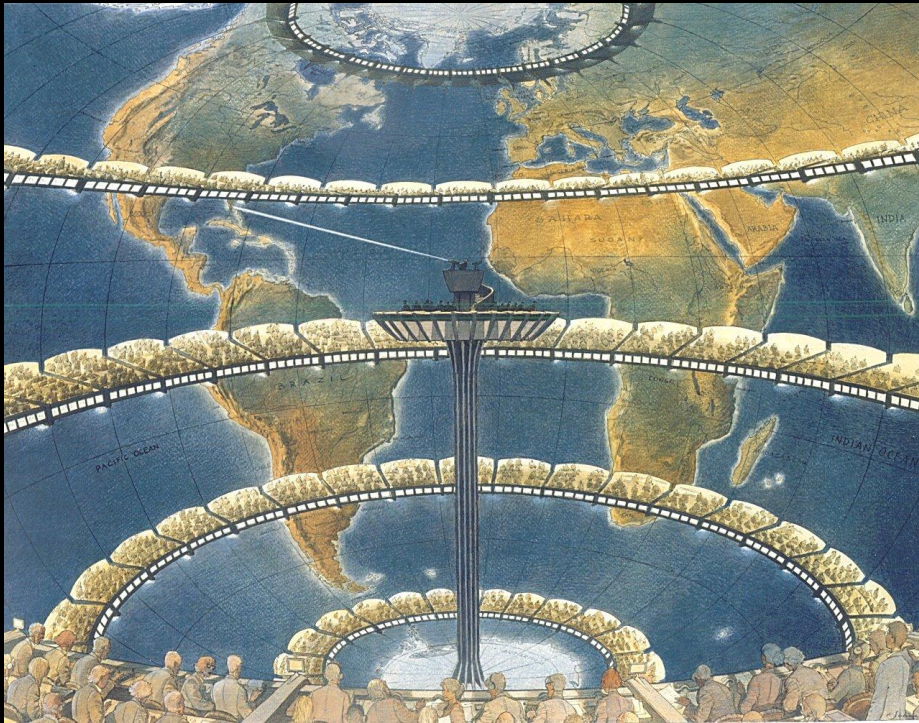


電子計算機の集まり

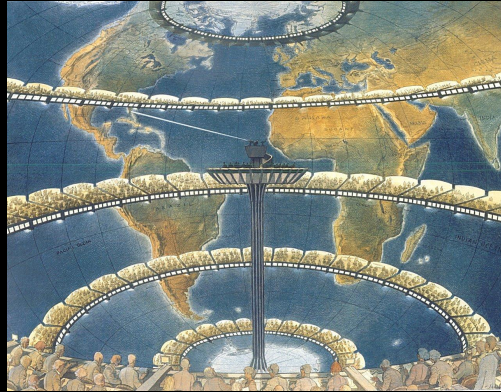
Image courtesy of the National Center for Computational Sciences, Oak Ridge National Laboratory

Cell

- 自然：無限(に近い)
- 計算機：有限 → 離散化

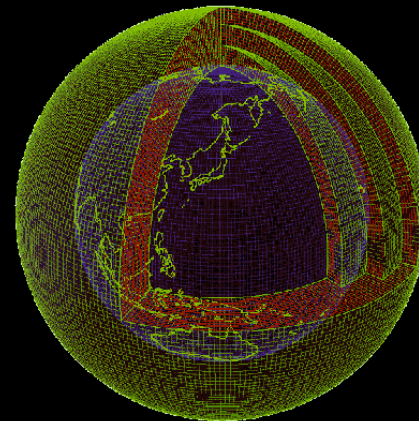
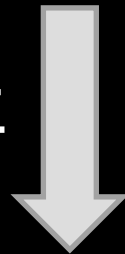


それから90年経ちました



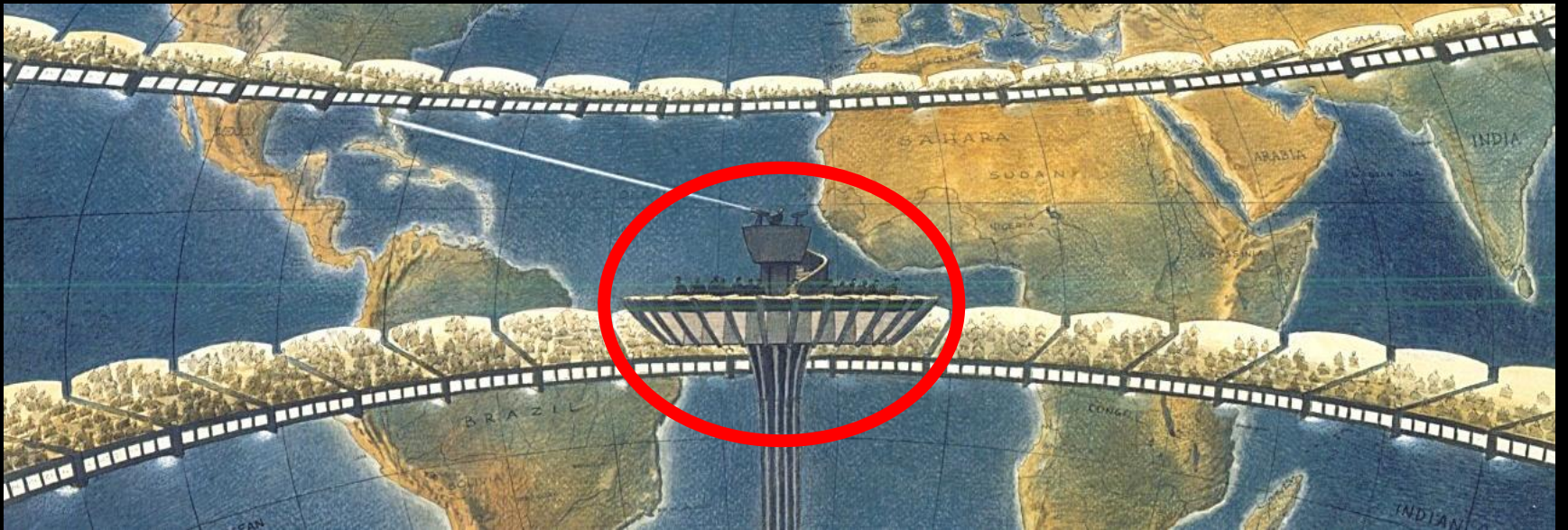
10¹²倍 = 1 T倍高速に

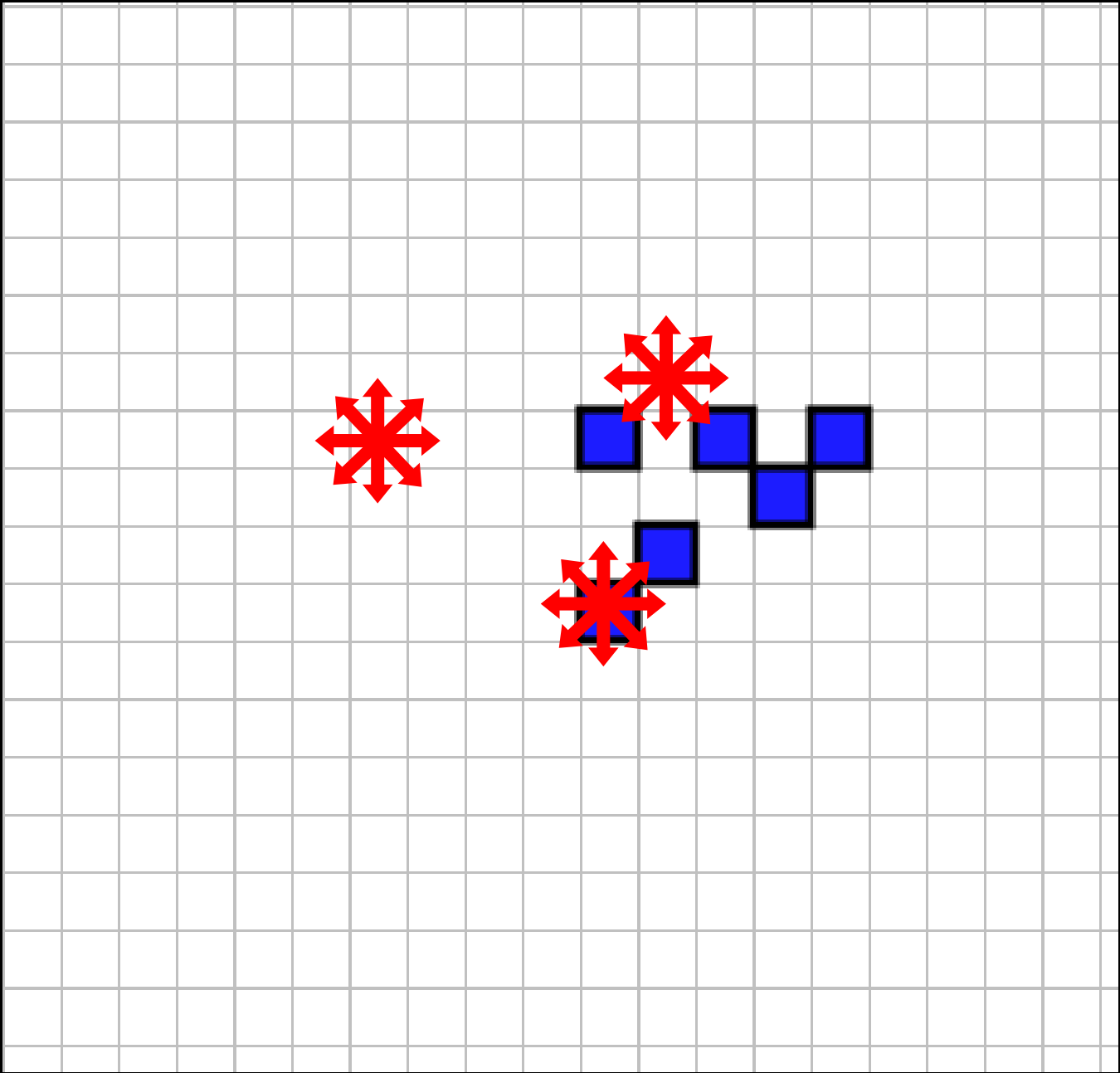
精密なモデル 計算手法



電子計算機の集まり

- 集まりということ
 - 烏合の衆, 船頭多くして... では意味がない
 - 計算機の集合(並列計算機)も同じ





リーダー, 指導者, 指揮者

- 並列計算機にも必要不可欠

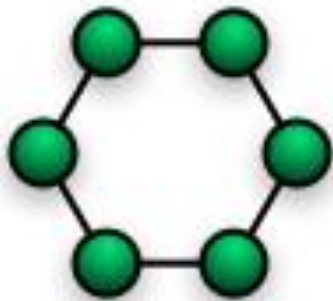
- ソフトウェア的には

- そもそも人間：研究者や利用者
 - プログラム自体
 - プログラムの構成(クライアントvs.サーバー)

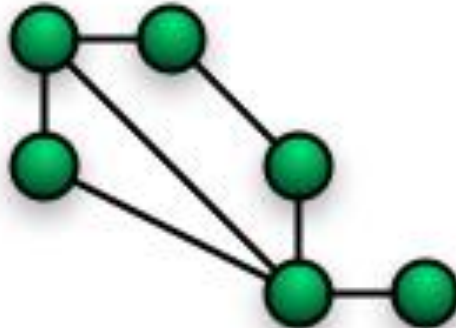
- ハードウェア的には

- 計算機自体の制御部分
 - ネットワークの構成

ネットワークトポロジー



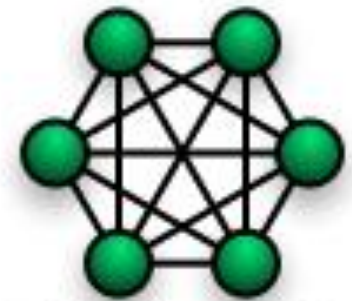
Ring



Mesh



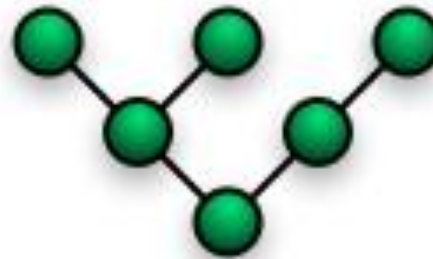
Star



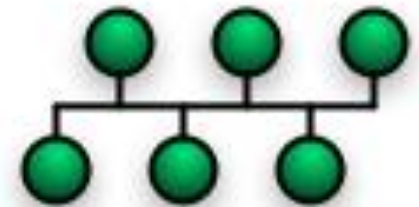
Fully Connected



Line



Tree

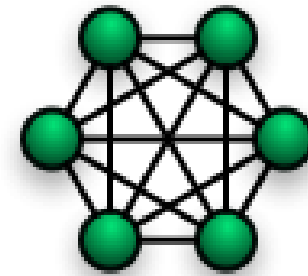


Bus

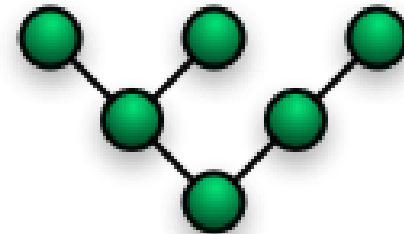
実例



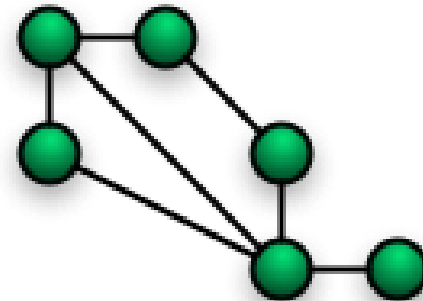
次世代スーパーコンピュータ「京」



Fully Connected



Tree



Mesh

並列計算機なんて私に関係ないし

- などということはありません

- 携帯電話 : 1 – 2? 並列な計算機
- パソコン : 2 – 6 並列な計算機
- ゲーム機 : 3 – 6 並列な計算機
- GPU : ~ 100 – 1000 並列な計算機
- スパコン : ~ 1000 -100000 並列な計算機
- Google : ~ 10000 並列な計算機
- インターネット : ~ 100000000 並列な計算機

計算機シミュレーション

- のお仕事とは
 - 世界にはどのような複雑な現象があるのだろうか？
 - 全分野(哲学, 文学, 理学)
 - **どういう法則が必要なのだろうか？**
 - 理学, 工学
 - **その法則に基づいてなにをどう計算するのだろうか？**
 - 数学, 工学
 - シミュレーション結果は何を意味するのだろうか？
 - 全分野

より詳細に

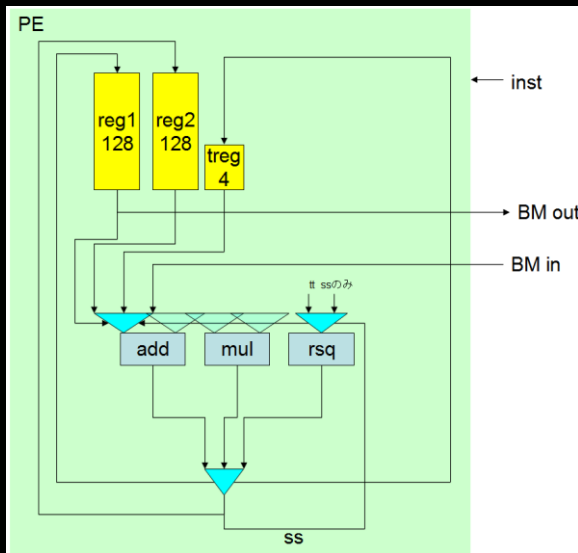
- **どういう法則が必要なのだろうか？**
 - 様々な現象の理解, 推測: 理学, 工学
- **その法則に基づいてどう計算するのだろうか？**
 - どう計算するのか? : 数学
 - どうやって計算機を使うのか: 計算機科学
 - どう計算機を実現するのか: 計算機工学
- **シミュレーション結果は何を意味するのだろうか？**
 - 計算科学 (理学, 工学)

私の興味

- 計算機を設計する実現する
 - 効率のよい計算機アーキテクチャ
- 計算機を使う
 - アルゴリズムの考案や作成
 - プログラミングによるアルゴリズムの実装
- 計算機結果について調べる考える
 - 将来の予測
 - 未知の現象の理解

計算機を設計する実現する

- 商用の計算機に刃向かうには？
 - 大量生産のため安い
 - なんでもできる
- 問題を特化した効率の良い計算機



```
s0 : swap_128 port map ( f => dx(11), x => x, y => y, xs => xx, ys => yy);

with dx(11) select
diff <= dy when '1',
dx when others;

e0 : extract_128_116_11 port map ( x => xx, s => sx, m => mx0, e => ex);
e1 : extract_128_116_11 port map ( x => yy, s => sy, m => my0, e => ey);

with ex select
mx <= (others => '0') when "000000000000",
mx0 when others;

with ey select
my <= (others => '0') when "000000000000",
my0 when others;

qx <= mx;

ff <= diff(10) or diff(9) or diff(8) or diff(7);
with ff select
shift <= "11111111" when '1',
diff(6 downto 0) when others;

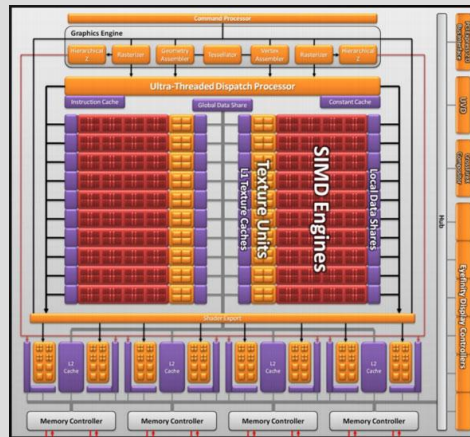
rs : int_rshift_V port map ( c => shft, i => my, o => qy0, s => rf1);
qy(0) <= qy0(0) or rf1; -- force1 rounding
qy(116 downto 1) <= qy0(116 downto 1);
```



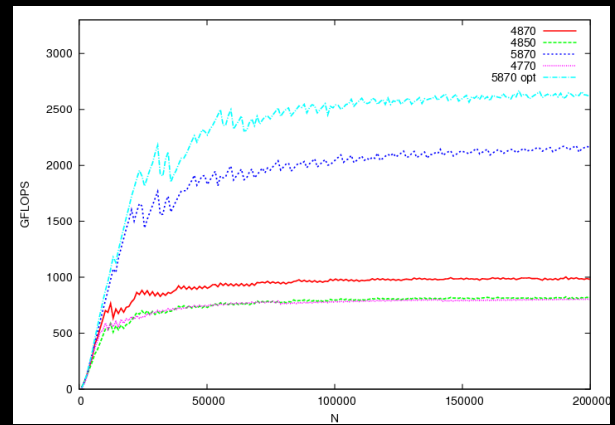
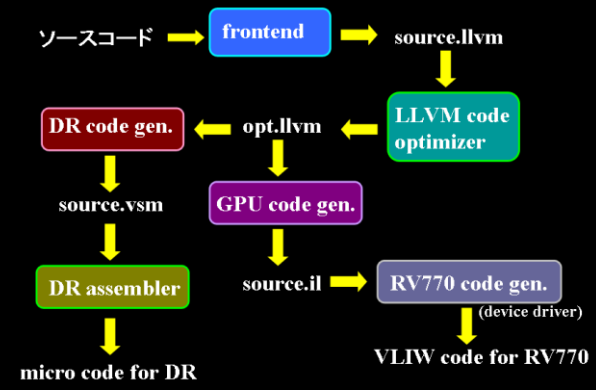
計算機を使う

- GPUという並列計算機のプログラミング

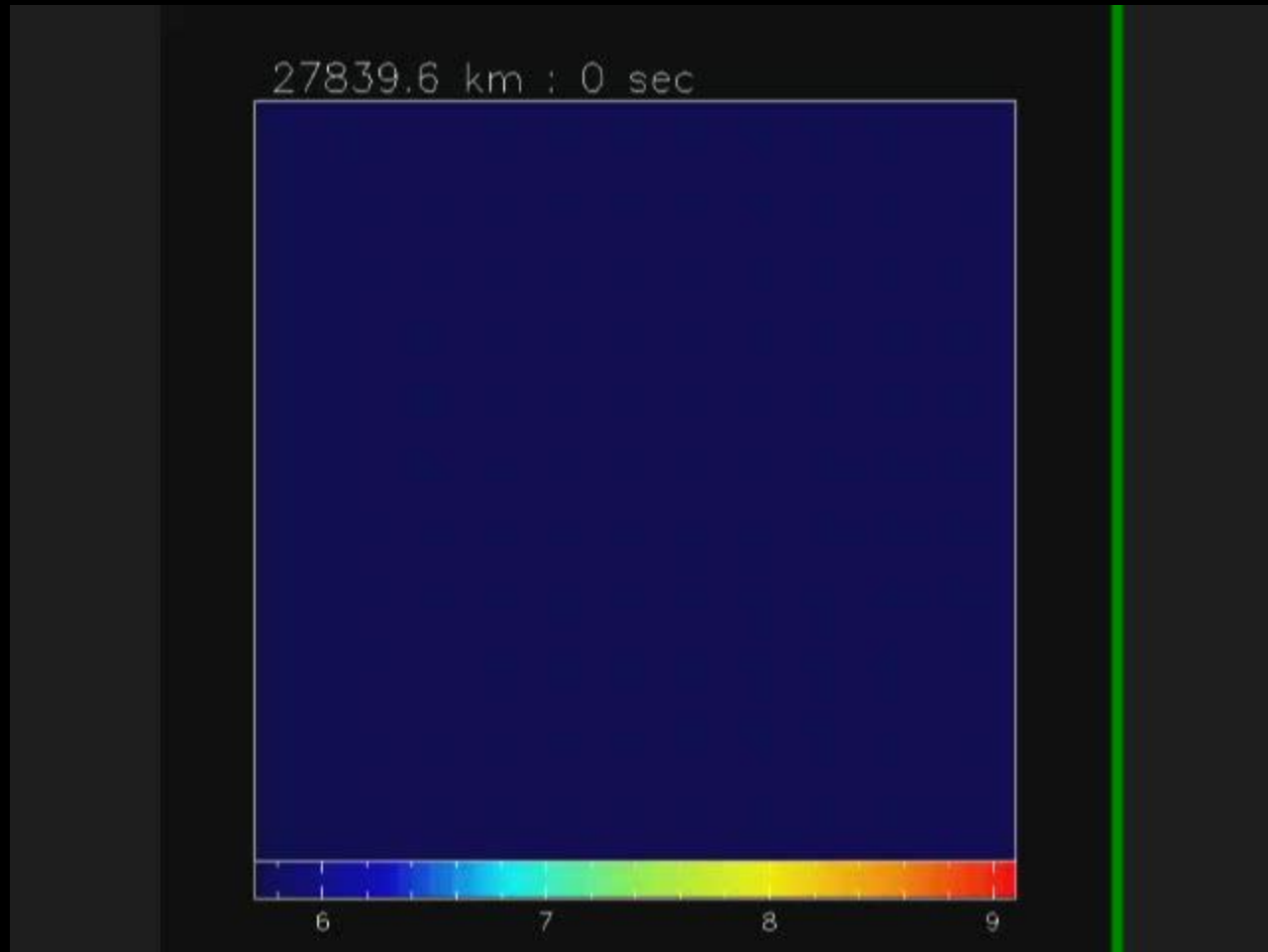
No.22



コンパイラフロー



計算機結果について調べる考える



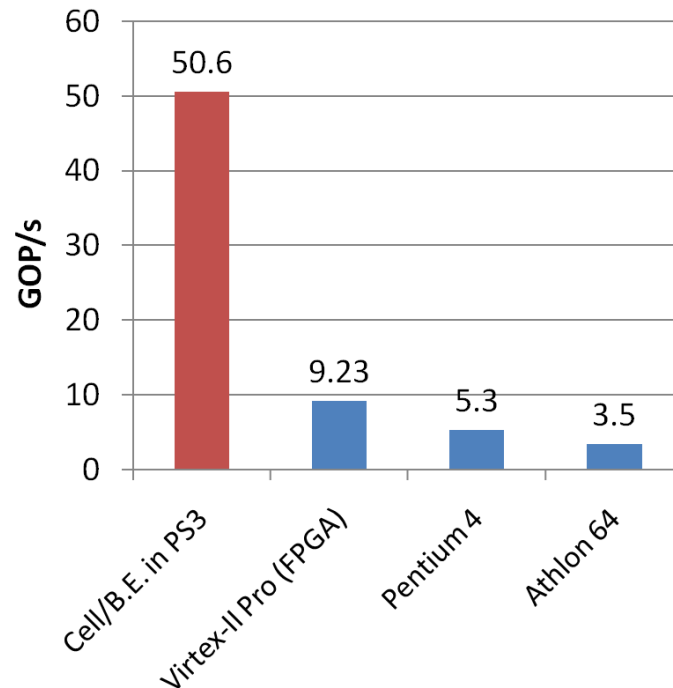
卒業研究の例 (1)

All-Pairs Shortest Paths Problem on PlayStation 3

- All-pairs shortest paths problem (APSP) is to find the minimum distance path between any two nodes:

$$d(i, j) = \min_{p \in \text{paths}(i, j)} \sum_{(u, v) \in E} w(u, v).$$

- APSP appears in
 - Network routing
 - Bioinformatics
 - LSI design
 - ...



卒業研究の例 (2)

Matrix Inversion on Cell/B.E.

- Matrix Inversion:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

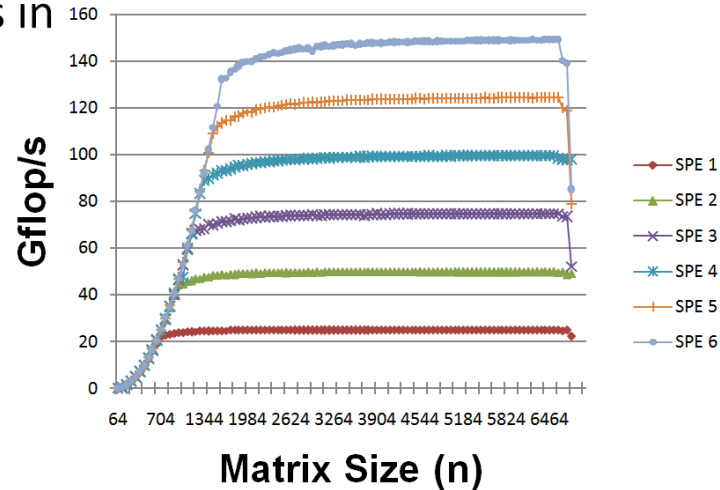
$\det |\mathbf{A}| \neq 0$, non - singular

- Matrix Inversion appears in

- Material Science
- Control Theory
- Computer Graphics
- System of Equations

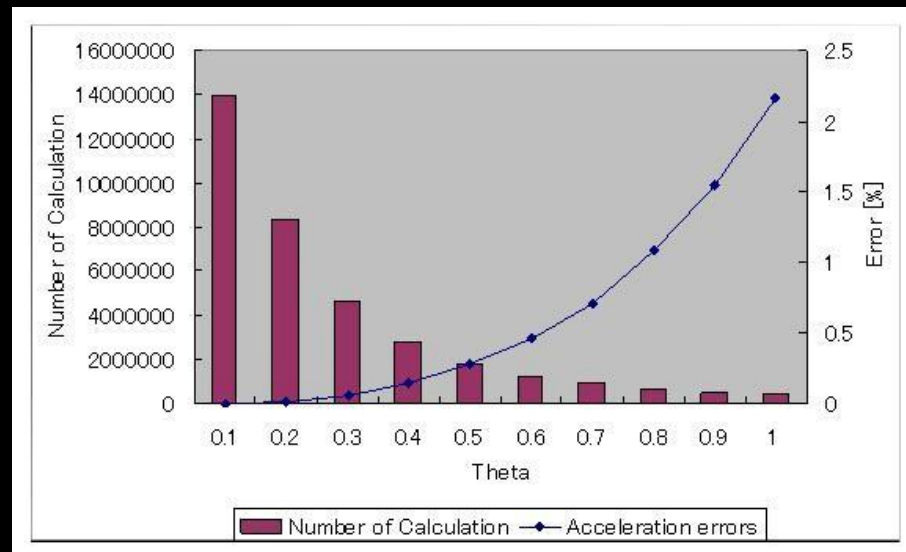
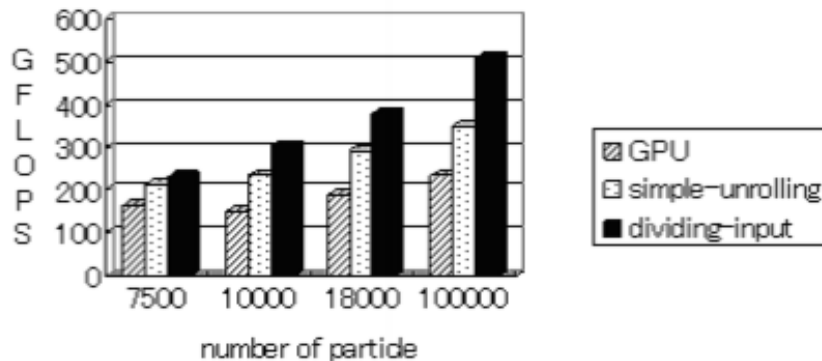
- Platform is Cell

- Heterogeneous
- 6 cores



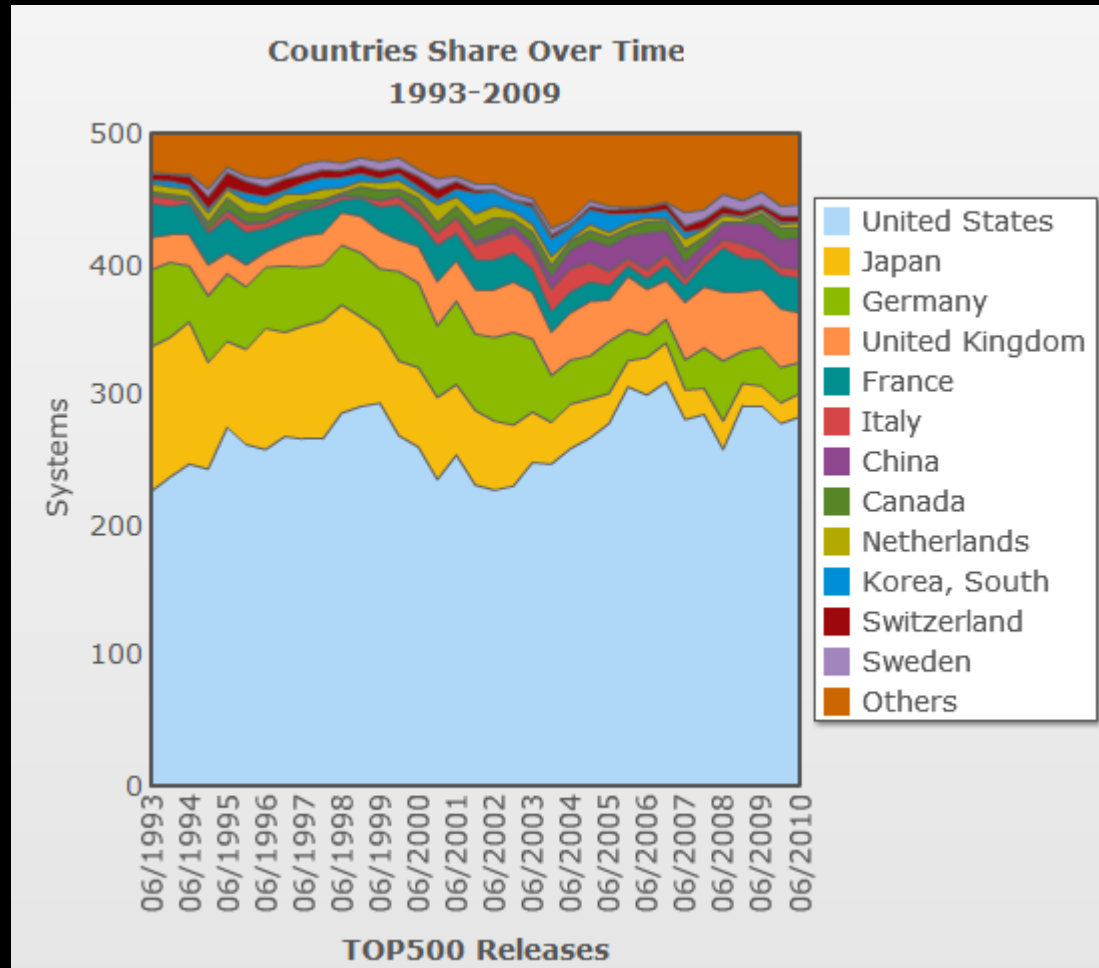
卒業研究の例 (3)

- GPUによる天体シミュレーションの高速化
 - CPU ~ 10^{10} 演算
 - GPU ~ 10^{12} 演算 : 特別なプログラミングが必要



2008年度 藤原さん, 佐藤さん

今日本では



- 日本発の高速な計算機のシェアが減少

野心的な未来

- **次世代スーパーコンピュータ「京」**
 - 10 Pflopsの演算性能をめざして
 - CPU, ネットワークの設計実装は日本産
 - 2012年に稼働予定

http://www.nsc.riken.jp/index_j.html

- **TSUBAME 2.0 (東京工業大学)**
 - GPUを大規模に活用する並列計算機
 - 2010年11月に稼働予定

<http://www.gsic.titech.ac.jp/>

消費電力の問題

- 携帯電話, ゲーム機, ノートパソコン, デスクトップPC
 - 消費電力が小さくて高速であればうれしい
 - 1週間充電のいらない携帯電話
 - 10時間連続でプレゼンのできるノートパソコン
 - 部屋の暑くならないゲーム機
- 並列計算機, データセンターも同じ
 - Power Usage Effectiveness

効率の良い計算機

- GRAPE-DR (東京大学・国立天文台)
 - 消費電力あたりで性能のよい並列計算機

日本のスパコン、省エネ世界一 市販PC部品を駆使

2010年7月8日16時28分



電力効率世界一になったスパコン
「GRAPE-DR」=東京大提供

ブ
リ
め
の
1
用

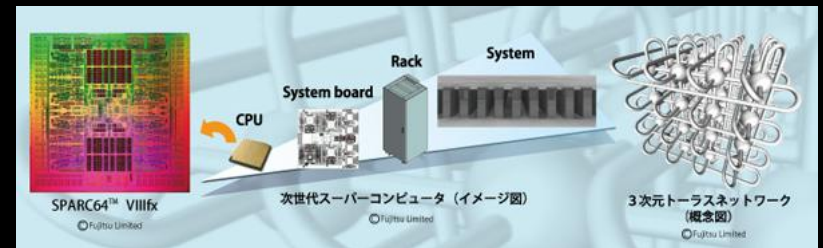


コンピューター
開する省エネ
日本のスパコ

は1秒間に8
長研究所のス
力率は3倍と

宣伝

- 9月に研究会を会津大学で開催します
 - 著名な研究者をお招きし高速な計算機について発表、議論をおこないます



- 10月以降にもスーパーコンピュータに関する講演会を計画中です

リーダー, 指導者, 指揮者

