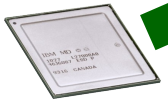
Blue Gene/Qの高速化手法とライフサイエンスへの応用

土井 淳
doichan@jp.ibm.com
東京基礎研究所：日本アイ・ビー・エム株式会社
19 Oct. 2012

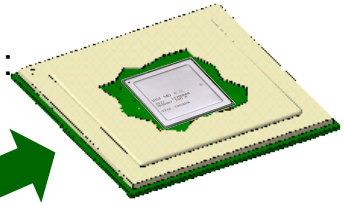
Blue Gene/Qの概要

Blue Gene/Q

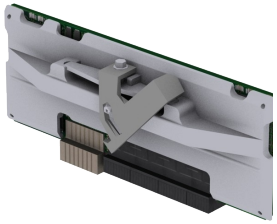
1. 計算チップ:
16 コア



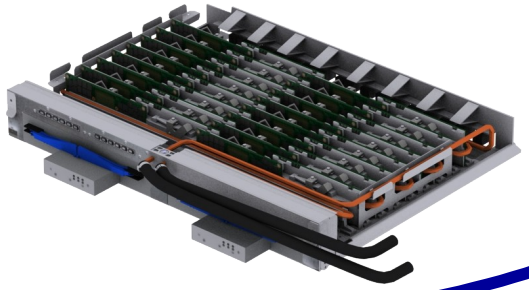
2. チップモジュール



3. 計算ノード:
1チップモジュールと
16 GB DDR3 メモリを搭載, 204.8 GF/s



4. ノードカード:
32枚の計算ノード,
トーラスネットワーク通信モジュール

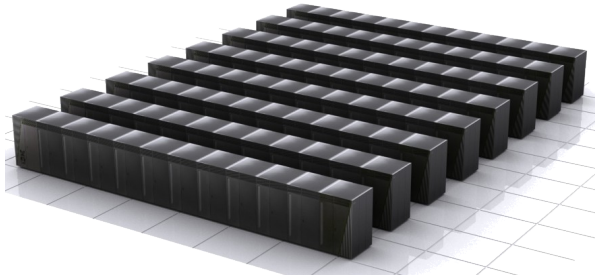


5a. ミッドプレーン（1/2筐体）:
16枚のノードカード

5b. IOドロワー:
最大8枚のIOノード
8 PCIe Gen2 x8 スロット

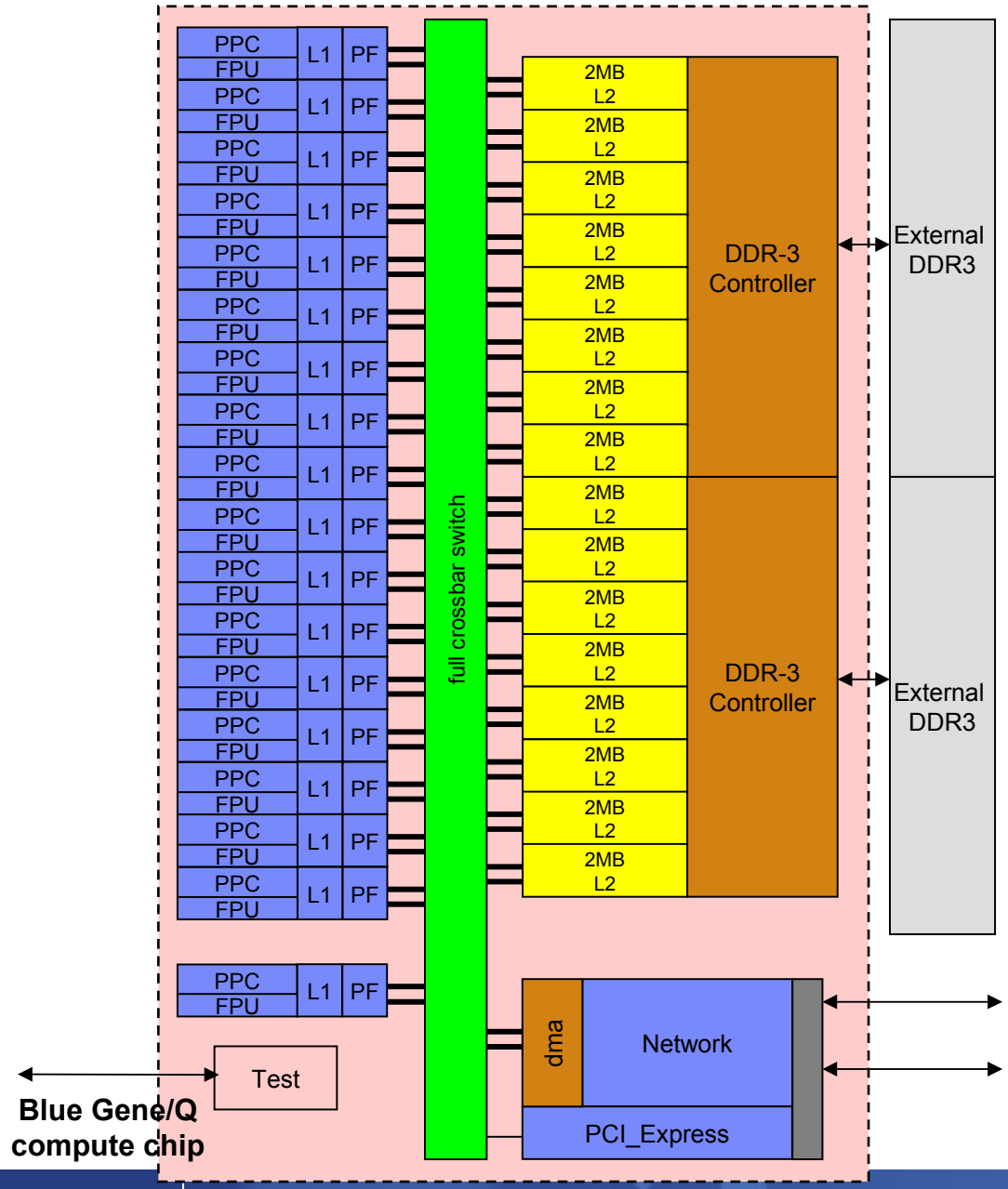
6. Blue Gene/Q筐体,
209.8 TF/s

7. システム:
20PF/s (96筐体)



Blue Geneファミリー仕様比較

Property		BG/L	BG/P	BG/Q
Node Properties	Processors per chip	2* 440 PowerPC	4* 450 PowerPC	16* A2
	Processor Frequency	0.7GHz	0.85GHz	1.6 GHz
	Processor Voltage	1.5V	1.2 V	0.7 V
	Coherency	Software managed	SMP	SMP
	L2/L3 Cache size (shared)	4 MB	8 MB	32 MB
	Main Store/processor	512MB/1GB	2GB/4GB	16GB
	Main Store Bandwidth	5.6GB/s (16B wide)	13.6 GB/s (2*16B wide)	42.6 GB/s (2*16B wide)
	Peak Performance	5.6GF/node	13.6 GF/node	204.8 GF/node
Torus Network	Bandwidth Off-box latency: hw/MPI (µs)	6*2*175MB/s=2.1GB/s 0.5/4	6*2*435MB/s=5.2GB/s 0.5/3	10*2*2GB/s=40GB/s 0.3/2.5
System Properties	Peak Performance	596TF (104k)	1PF (72k)	20PF (96k)
	Total Power	2.3MW	2.3MW	8MW



Blue Gene/Qチップアーキテクチャ

- 16+1 コアSMP（1コアはシステムが使用）
  コアあたり4ハードウェアスレッド（SMT）
- 動作周波数，1.6 GHz
- コアあたり4浮動小数点数演算のSIMD演算器
  ノードあたり　204.8 GF
- 563 GB/s の共通L2メモリ間バイセクションバンド幅
- 32 MBの共通L2キャッシュメモリ
- 42.6 GB/s のDDR3メモリバンド幅 (1.333 GHz DDR3)
- 2.0 GB/sのバンド幅の10本のプロセッサ間リンク
- 2.0 GB/sのバンド幅の1本のI/Oリンク
- ノードあたり16 GB のメモリ
- ノードあたり~30 Wの消費電力

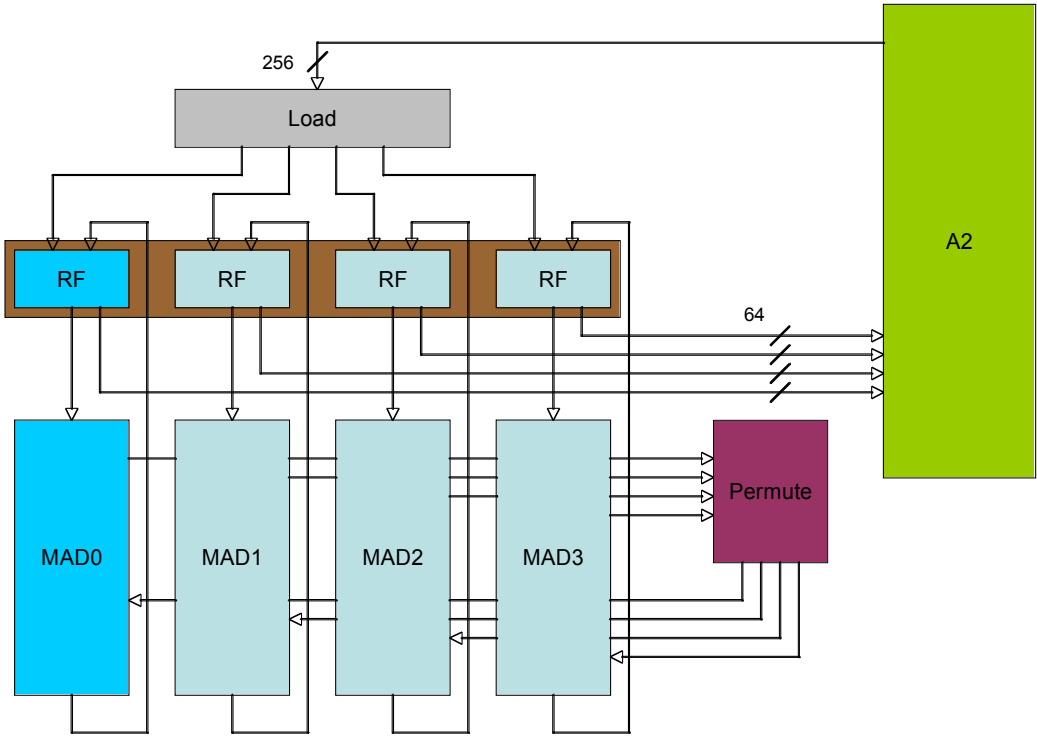
2 GB/s I/O link (to I/O subsystem)

10*2GB/s intra-rack & inter-rack (5-D torus)

note: chip I/O shares function with PCI_Express

Quad FPU：4−wayのSIMD演算器

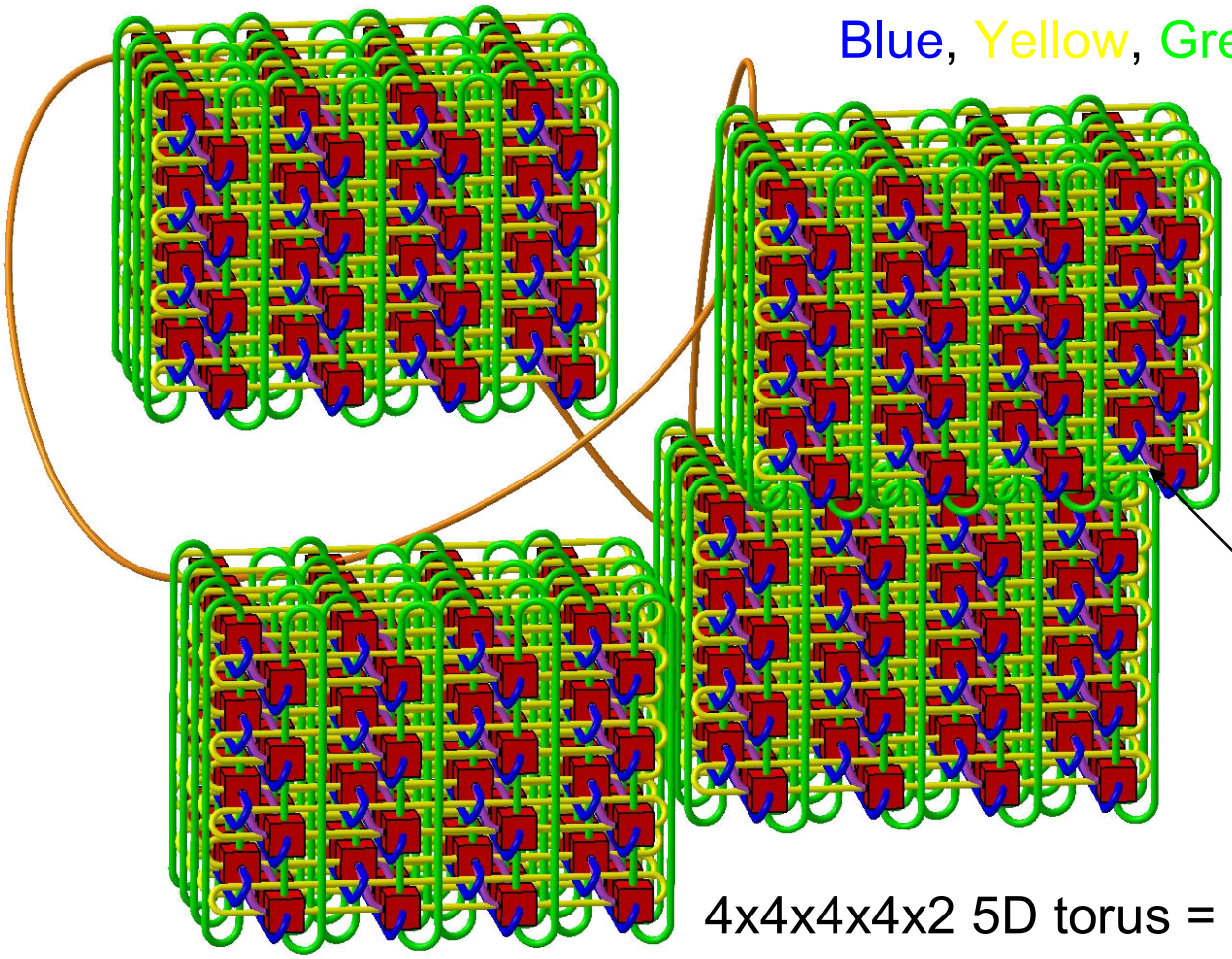
- 4-way 浮動小数点演算
  - BG/L,Pは 2-way
  - 倍精度，単精度ともに
- 2-way　複素数演算
  - BG/L,Pは1-way
- Multiply-add演算のサポート
  - 最大でサイクルあたり8演算
- A2コアのAXUポートに接続
  - サイクルあたり1命令発行
- 4本のレジスタファイル
  - スレッドあたり32x256ビット　（4x倍精度浮動小数点数）
- L1キャッシュと256ビットの読み書きが可能



5次元トーラス通信網

1～4次元目

Blue, Yellow, Green and Orange links



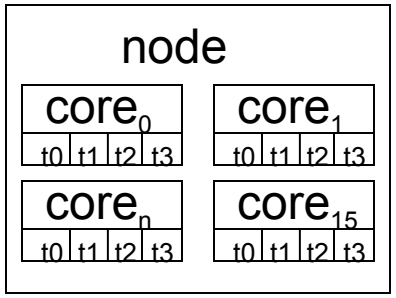
5th dim. length=2

(fixed)

4x4x4x4x2 5D torus = 512 nodes (midplane)

1 rack: 8x4x4x4x2, 2racks: 8x8x4x4x2, 3 racks: 12x8x4x4x2, ...

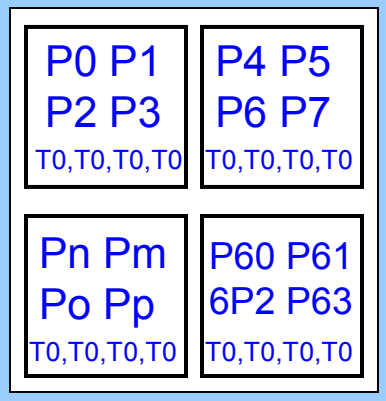
Blue Gene/Qの実行モード



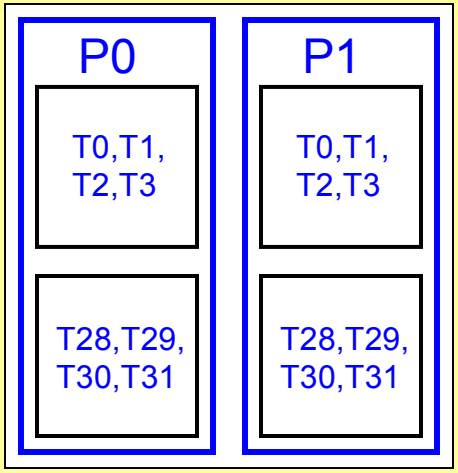
1ノードあたりのプロセス数を1～64の範囲で選択

ノードあたりのプロセス数＊プロセスあたりのスレッド数≦64

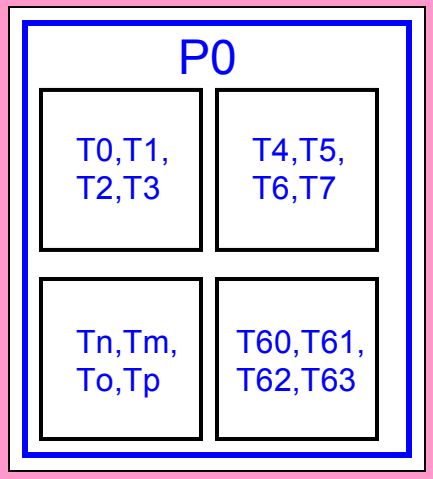
フルflat-MPIモード
64 プロセス
1 スレッド/プロセス



ミックスモード
2,4,8,16,32 プロセス
32,16,8,4,2 スレッド



フルSMPモード
1 プロセス
64 スレッド



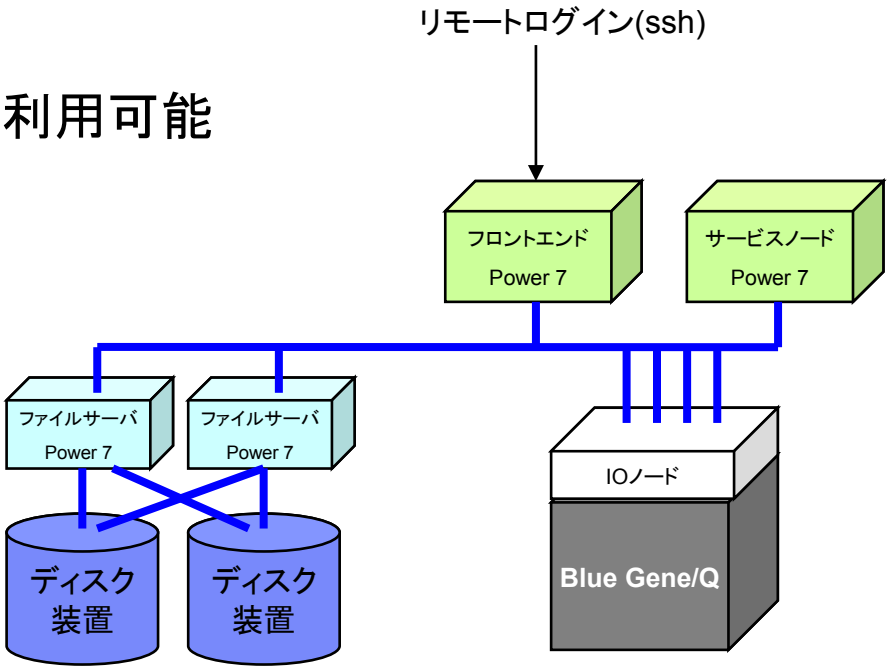
Blue Gene/Qの開発環境とアプリケーションの移植

移植可能なアプリケーション

- **使用言語**
  - C/C++
  - Fortran
- **並列化**
  - MPI（MPICH2相当）
  - OpenMP
  - pThread
    - マスタースレッドとあわせてノードあたり最大64スレッド生成可能
- **メモリ空間**
  - 64ビット
  - ノードあたり16GB（仮想メモリ＝実メモリなので16GBが上限）

Blue Gene/Qの開発環境

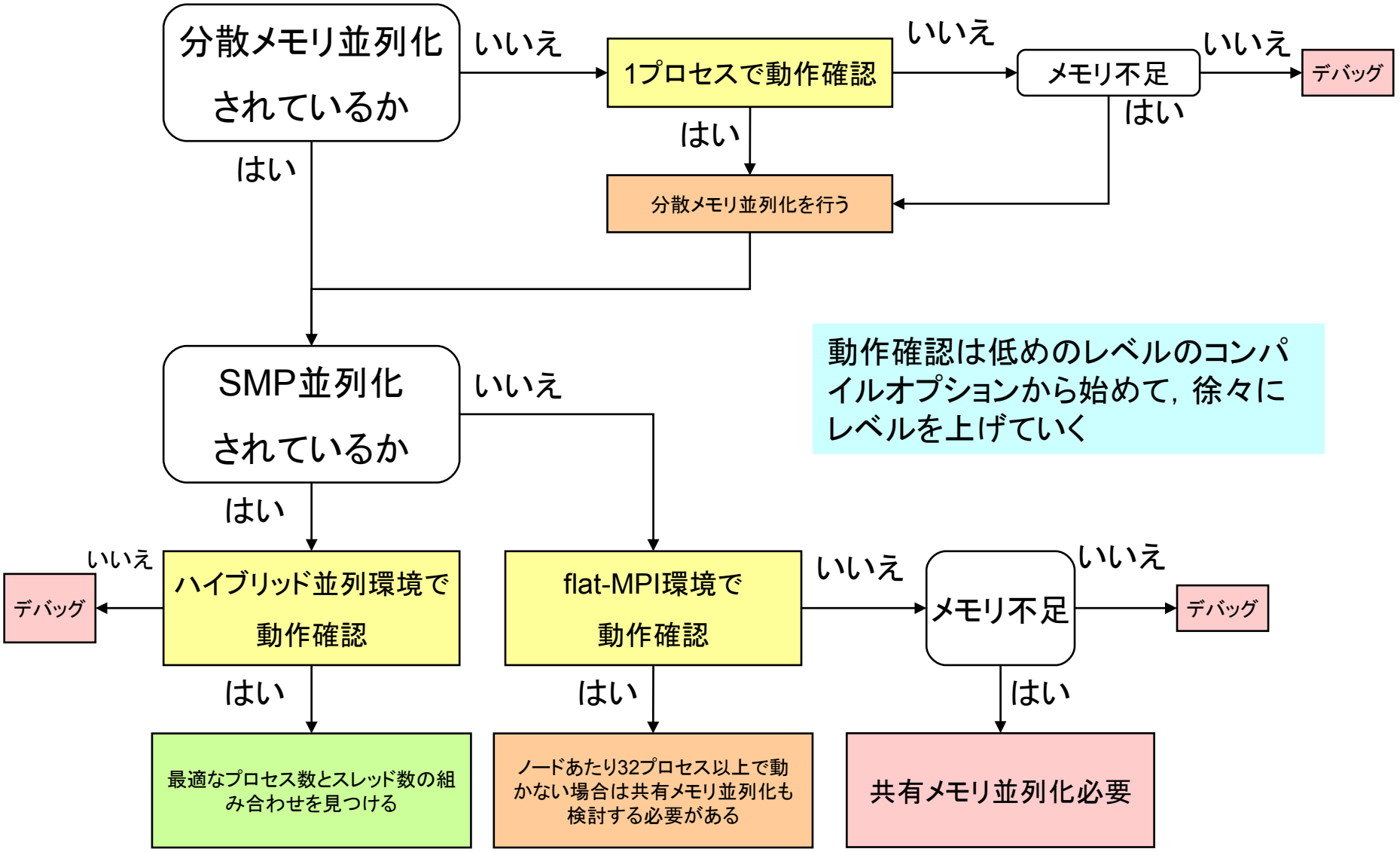
- **フロントエンドノード**
  - ログインノード，作業ノード
    - クロスコンパイル，アプリケーションやデータの管理や可視化など
    - Blue Gene/Qへのジョブの投入
      - LoadLeveler等により資源割り当て管理可能
  - Linux環境
    - 通常のほとんどのLinuxツールが利用可能
  - Power7プロセッサ
- **ファイルサーバー**
  - GPFSファイルシステム
- **Blue Gene/Q**
  - ユーザログイン不可



Blue Gene/Qの開発環境（続き）

- **クロスコンパイラ**
  - XL C/C++コンパイラー，XL Fortranコンパイラー
  - gnuコンパイラ：gcc, g++, gfortran
- **数値計算ライブラリ**
  - MASS,MASSV：算術演算ライブラリ(-lmの最適化版)
  - ESSL：数値計算ライブラリ（BLAS,LAPACK,FFT）
- **プロファイラ**
  - gprof
- **デバッガ**
  - gdb
  - 今後各種市販ツールも利用可能予定

Blue Gene/Qへのアプリケーション移植



動作確認は低めのレベルのコンパイルオプションから始めて，徐々にレベルを上げていく

Blue Gene/Qにおけるアプリケーション最適化

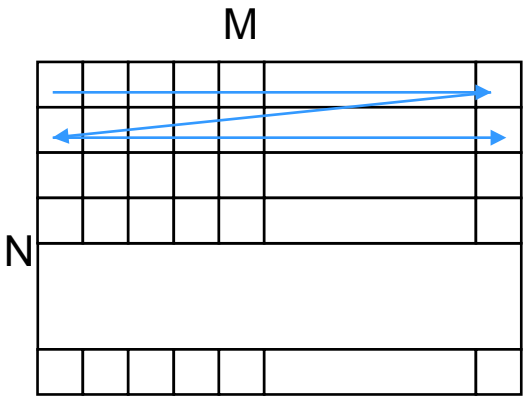
アプリケーションの最適化

- **メモリアクセスの最適化**
  - キャッシュメモリの最適化
    - ループ最適化，キャッシュブロッキング等
  - データレイアウト，データアクセスの最適化
- **Quad FPUによるSIMD演算**
  - コンパイラによるSIMD化
  - SIMDに向いたコードへの書き換え
  - ハンドチューニングによるSIMD化
- **共有メモリ並列化による最適化**
  - ノードあたりのプロセス数と，プロセスあたりのスレッド数の選択
- **分散メモリ並列化の最適化**
  - 問題の分割とノードマッピング
  - 計算と通信のオーバーラップ

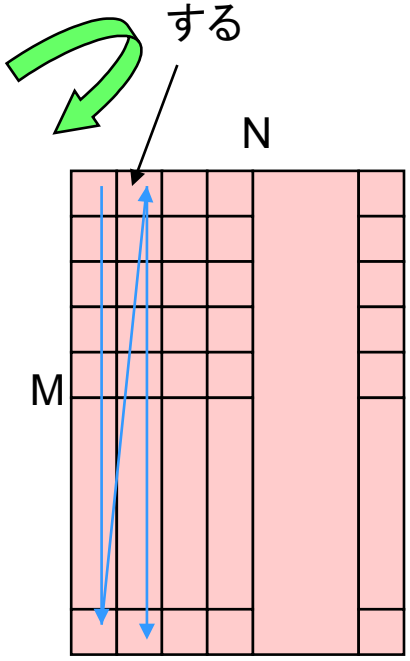
L1 キャッシュブロッキング

行列の転置の例

```
COMPLEX*16 A(N,M), B(M,N)

DO I=1,N
  DO J=1,M
    A(I,J) = B(J,I)
  ENDDO
ENDDO
```

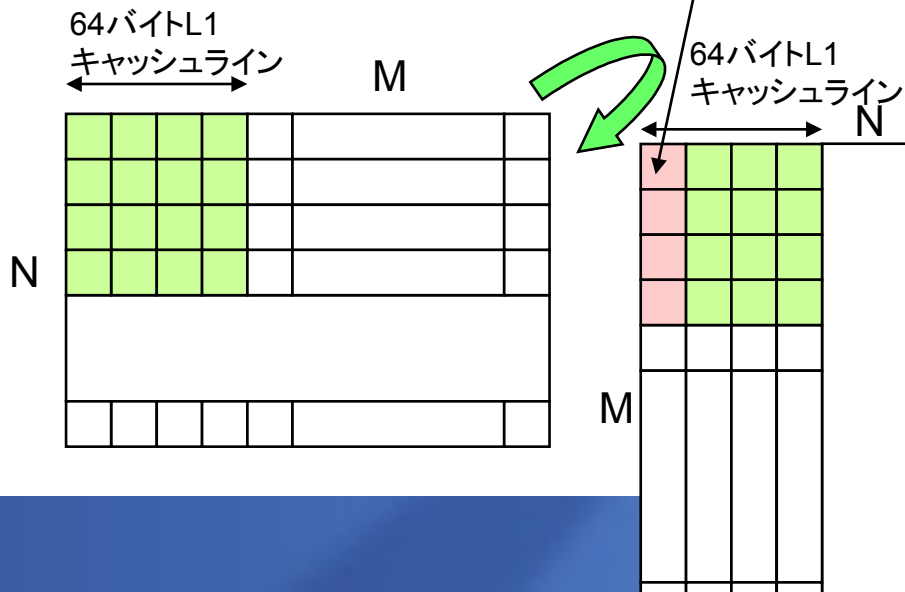


データストア時に
毎回キャッシュミス
する



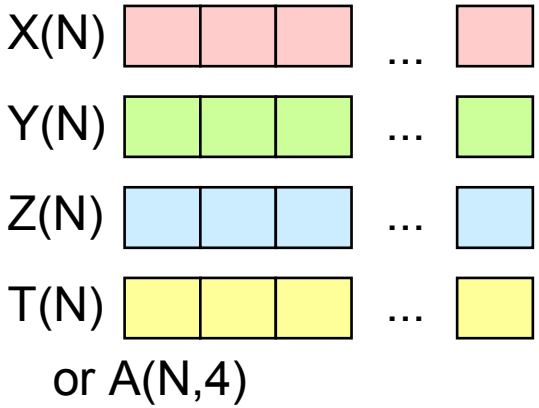

```
COMPLEX*16 A(N,M), B(M,N)

DO I=1,N,4
  DO IC=1,4
    DO J=1,M,4
      A(I+IC,J)   = B(J   ,I+IC)
      A(I+IC,J+1) = B(J+1,I+IC)
      A(I+IC,J+2) = B(J+2,I+IC)
      A(I+IC,J+3) = B(J+3,I+IC)
    ENDDO
  ENDDO
ENDDO
```

最初の1回は
キャッシュミスする
が後の3回はヒット



データレイアウトの考察: Structure of Arrays (SoA) vs Array of Structure (AoS)

Structure of Arrays

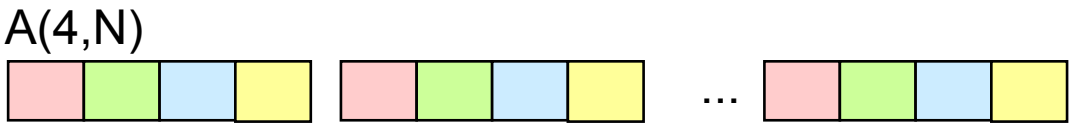


一般的にSIMDやベクトル化に向いている

多くの要素を扱う場合にデータストリームが多くなる

多次元配列の袖領域へのアクセス時にキャッシュミスが頻発するケースがある

Array of Structure



キャッシュの利用に有利

こちらのほうがSIMDに適している場合もある(要素数や計算による)

要素数によってはSIMD化できない

(例えば, 3次元ベクトルは, 4-way SIMDには向かない)

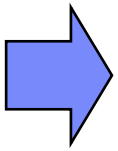
Quad FPUによるSIMD化

- **コンパイラによる自動SIMD化**
  - コンパイラが自動的にSIMD化しやすいコードにする必要あり
  - ディレクティブによりコンパイラを助ける
- **SIMD化しやすいコードに書き換える**
  - 連続アクセス
    - ループアンローリングにより簡単にSIMD化できるようなコード
  - 直接アクセス
    - テーブル参照等で非連続アクセスの場合自動SIMD化は困難
  - 単純な計算の長いループ

連続アクセス，直接アクセスへの書き換えの例

アドレスが計算で求まるような場合

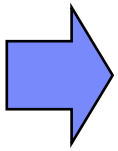
```
COMPLEX*16 A(*),B(*)
i=Nx+1
DO y=1,Ny
  i=i+1
  DO x=1,Nx
    i=i+1
    B(i) = (A(xup(i))+A(xdn(i))+A(yup(i))+A(ydn(i)))/4.0D0
  ENDDO
  i=i+1
ENDDO
```



```
COMPLEX*16 A(*),B(*)
i=Nx+1
DO y=1,Ny
  i=i+1
  DO x=1,Nx
    i=i+1
    B(i) = (A(i+1)+A(i-1)+A(i+(Nx+2))+A(i-(Nx+2)))/4.0D0
  ENDDO
  i=i+1
ENDDO
```

テーブルの中身がわからない場合

```
REAL*8 V(*),U(*)
D=0.0D0
DO i=1,Nidx
  D=D+V(idx(i))*U(i)
ENDDO
```



```
REAL*8 V(*),U(*)
REAL*8 VT(*)
D=0.0D0
DO i=1,Nidx
  VT(i)=V(idx(i))
ENDDO
DO i=1,Nidx
  D=D+VT(i)*U(i)
ENDDO
```

一度作業用の配列にコピーしてしまう

2個目のループは連続アクセスなので
SIMD化可能

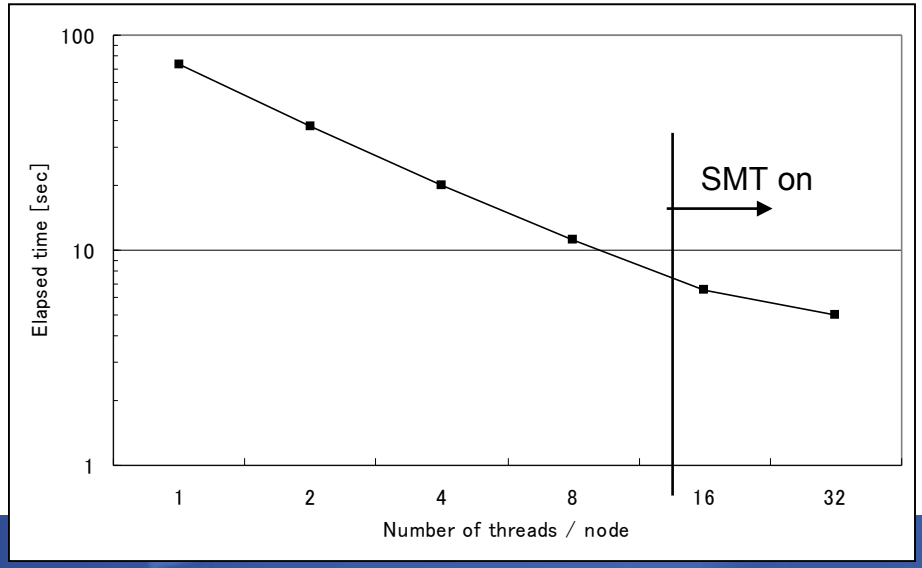
SMPによる並列化に関する考察

- **ノードあたりのプロセス数を増やすと**
  - プロセスあたりのメモリ量，メモリと通信のバンド幅が減る
    - メモリ量についてはシステムリソースもプロセス分必要なので，相対的にプロセスが増えるとユーザメモリが小さくなる
- **ノードあたりのプロセス数を減らすと**
  - 効率の良い共有メモリ並列化が必要
    - プログラム内に共有メモリ並列化できる十分な並列度が必要
  - スレッド間同期等のオーバーヘッド

SMTを使用する

- コアあたり2以上のハードウェアスレッドを利用する
  - 1サイクルあたり浮動小数点演算と整数演算を1つずつ同時に実行できる
  - しかし，1スレッドあたり1サイクルに1つの命令のみしか発行できない
  - コアあたり2スレッド以上であればそれぞれのスレッドから浮動小数点演算と整数演算を同時に発行/実行できる
- SMTにより命令レイテンシを隠す
  - 1つのスレッドがストールしても他のスレッドが動ける可能性がある

ノードあたりのスレッド数を変えた場合の実行時間の比較



プロセス数とスレッド数の組み合わせの考察

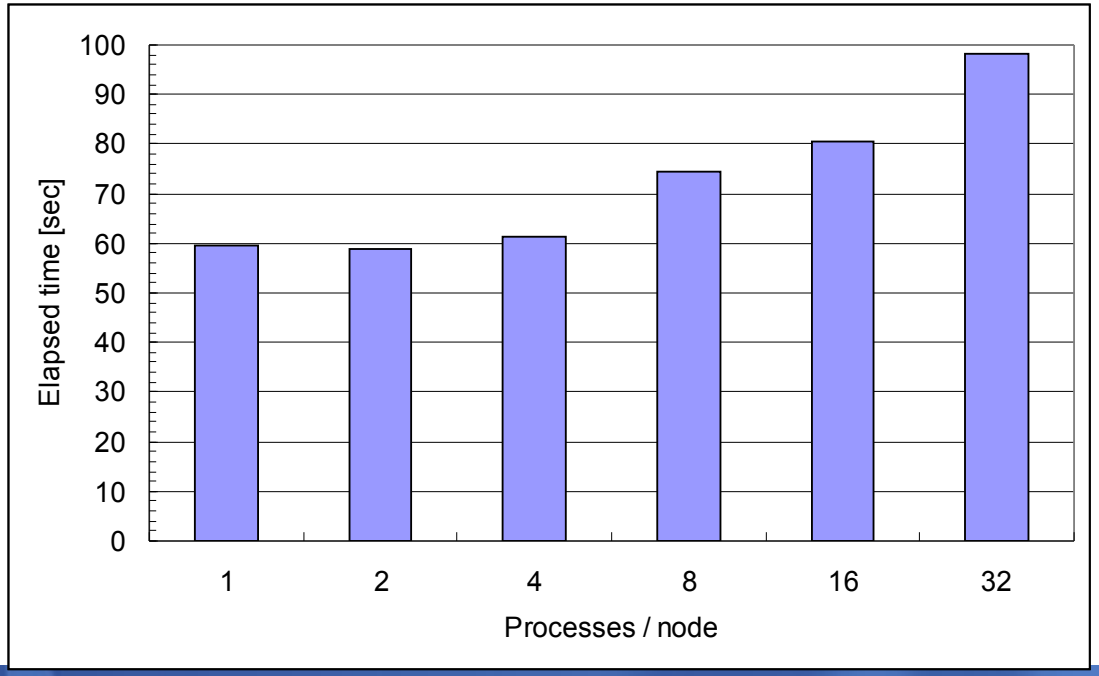
- **プロセス数とスレッド数がうまくバランスの取れる組み合わせを見つける**
  - メモリアクセス，通信パターンなどに依存する
  - 一般的に，プロセス数を少なめにした方が，性能が良い傾向
- **アプリケーション毎に試す必要がある**
  - 可能な限り，MPI+OpenMPのハイブリッド並列が望まれる

格子QCDの例

ノードあたりの問題サイズを一定にし，ノードあたりの全スレッド数が32になるような組み合わせで実行した場合の実行時間の比較

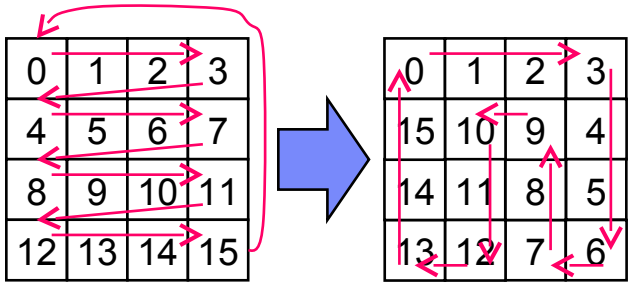
(プロセス数*プロセスあたりスレッド数 = 32)

通信を計算の裏に隠すような最適化を施したコードのため，ノードあたりのプロセス数を増やすと，通信バンド幅が下がることで，通信時間を隠しきれなくなっている



問題の分割とノードマッピング

- **分散メモリ並列化ではどのように問題を分割するかが重要**
  - 通信量，通信パターン，アルゴリズム等を考慮
  - Blue Geneの場合多くの並列度を稼げる分割方法が必要
    - 格子状のデータ構造であればできるだけ多次元分割する必要がある
- **5次元トーラスに問題をマッピングする**
  - 差分法のような問題の場合隣接格子との通信が重要
  - 隣接格子が物理的に隣の計算ノードになるようなマッピング
    - トーラスは5次元なので，問題の次元数（分割した次元数）に合わせた仮想的なトーラスを形成する
    - 512ノードを3次元トーラストして使うには
      - 4x4x4x4x2　>>>　(4x4)x4x(4x2) = 16x4x8
  - 実行時にノードマッピングを変えることが可能



Blue Gene/Qにおける格子QCDの最適化事例

格子QCDのカーネルルーチン

Wilson-Diracオペレーター

3x3 ゲージ場行列

4 スピノル（3色の物理量を持つ）

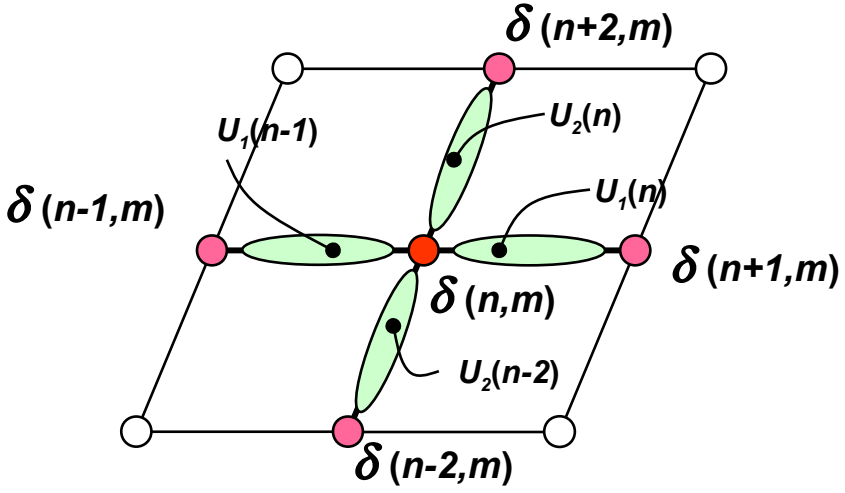
$$D(n,m) = \delta(n,m) - \kappa \sum_{\mu=1}^{4} \left\{ (1-\gamma_\mu) U_\mu(n) \delta(n+\mu,m) + (1+\gamma_\mu) U_\mu^t(n-\mu) \delta(n-\mu,m) \right\}$$

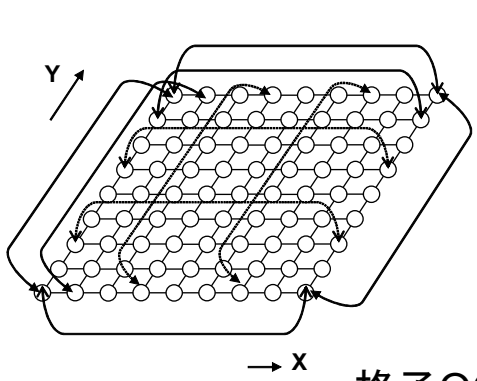
前方　後方

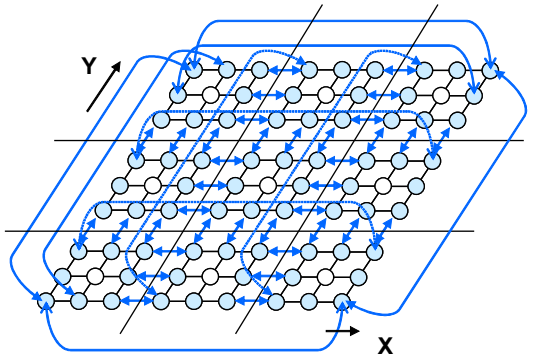
4次元分加算

4つの3次元ベクトルと3x3行列の乗算を8隣接(XYZT,+-)について行うステンシル計算

（全て複素数）







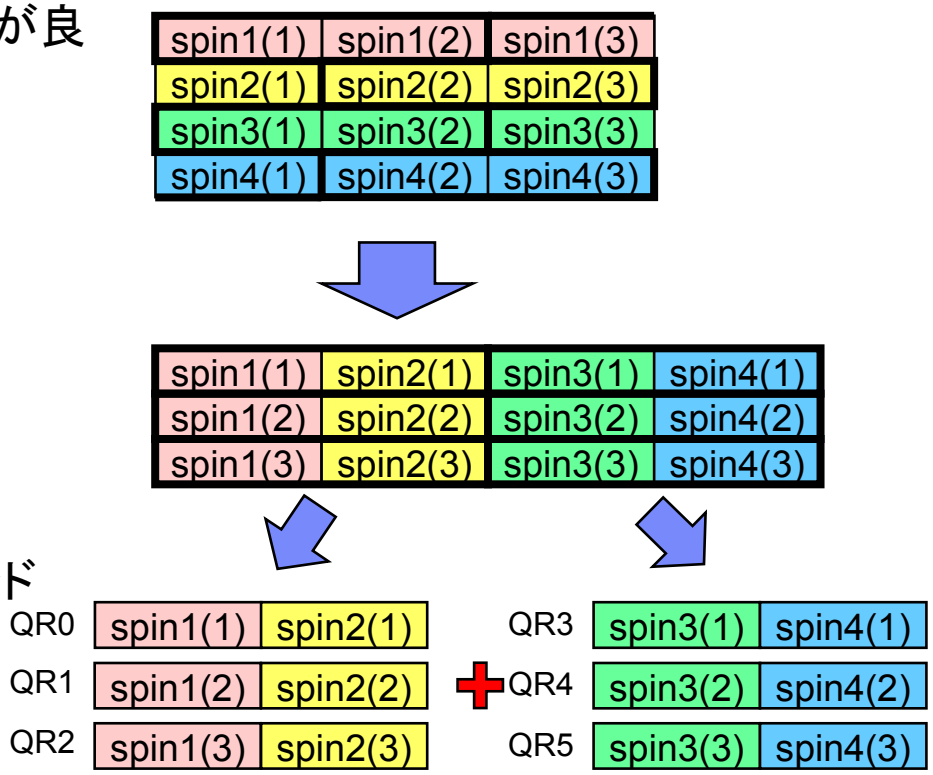
格子QCDの並列化

Wilson-Diracオペレーターの Quad FPUによる最適化

- **Quad FPUによる複素数演算**
  - Quad FPUは2つの複素数演算を同時に実行
  - 2つのベクトルと行列乗算を同時に実行

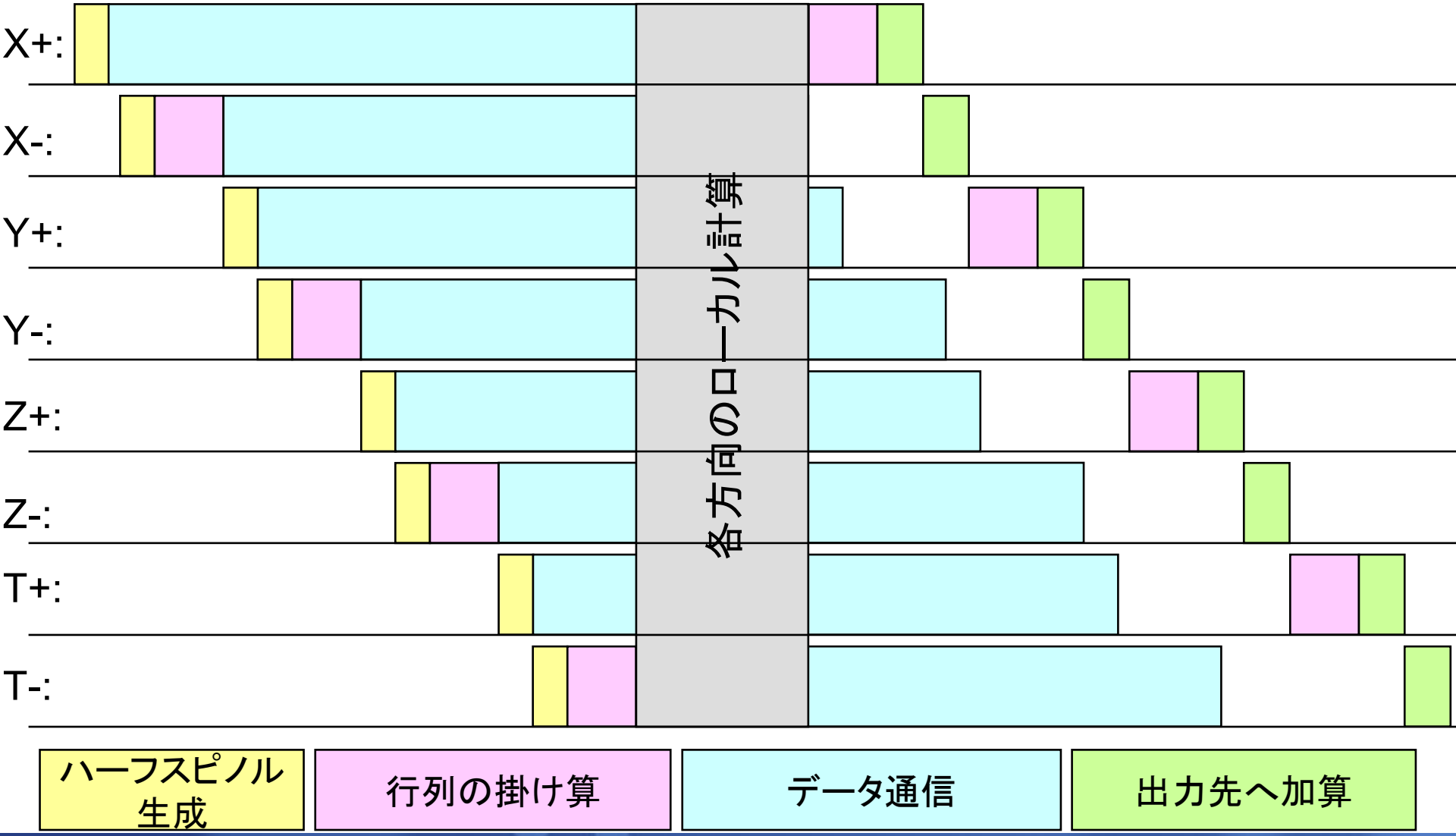
spinA(1)	spinB(1)	x	m(1,1)	m(2,1)	m(3,1)
spinA(2)	spinB(2)		m(1,2)	m(2,2)	m(3,2)
spinA(3)	spinB(3)		m(1,3)	m(2,3)	m(3,3)

- **Quad FPUに向いたデータ配置**
  - 格子QCDでは，3x4の形のスピノル構造が良く用いられる
    - COMPLEX*16 W(3,4,Nx,Ny,Nz,Nt)
    - このままだと非常にSIMD化しづらい
  - 4x3の形に書き直す
    - COMPLEX*16 W(4,3,Nx,Ny,Nz,Nt)
    - きれいにSIMD化できる
- **Quad FPU命令の最適化**
  - 今回はインラインアセンブラを用いてハンドコーディング



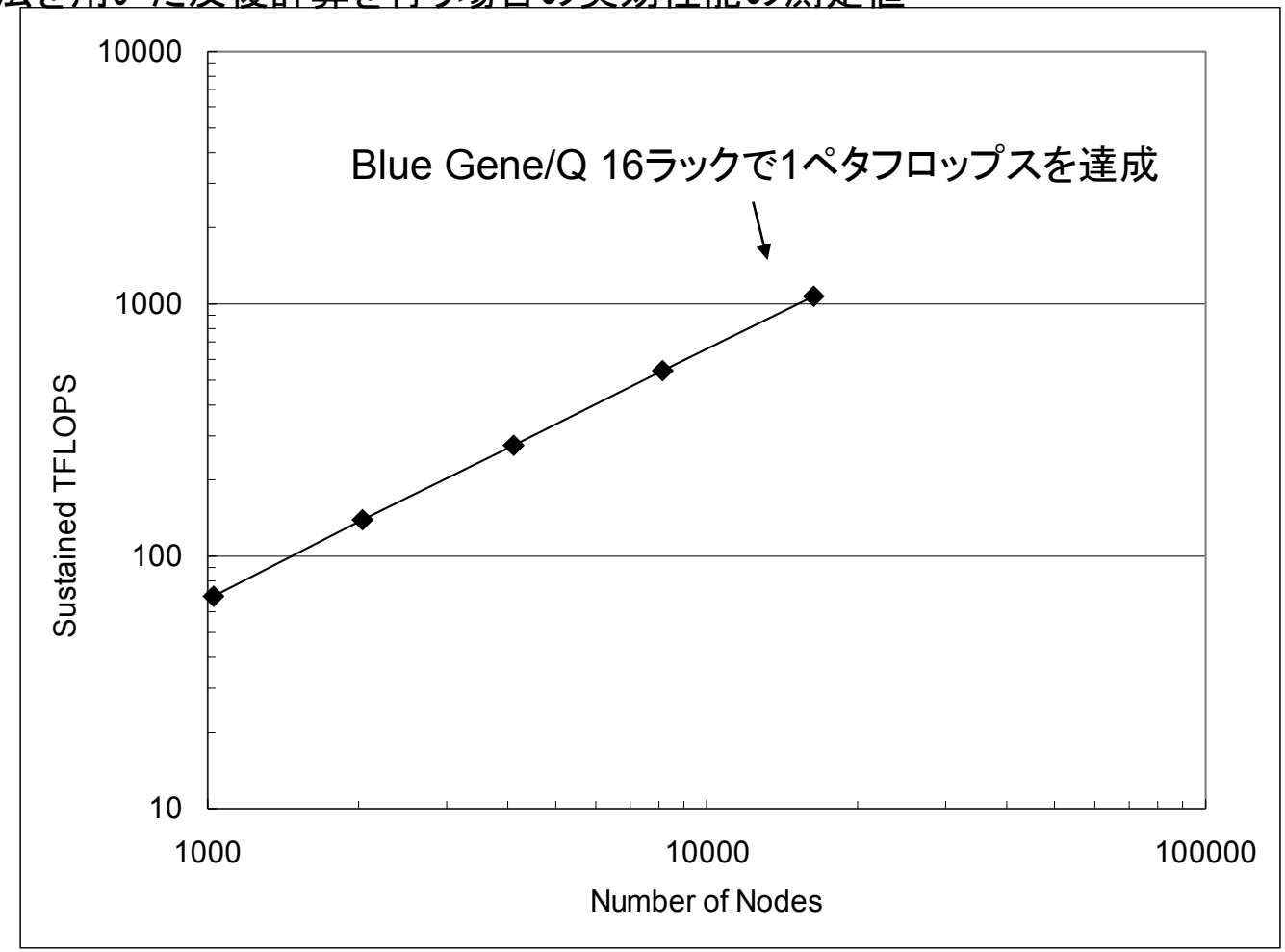
通信と計算のオーバーラップ

8方向それぞれについてパイプライン的に処理する



Blue Gene/Qにおける格子QCDのスケーリング

BiCGStab法を用いた反復計算を行う場合の実効性能の測定値



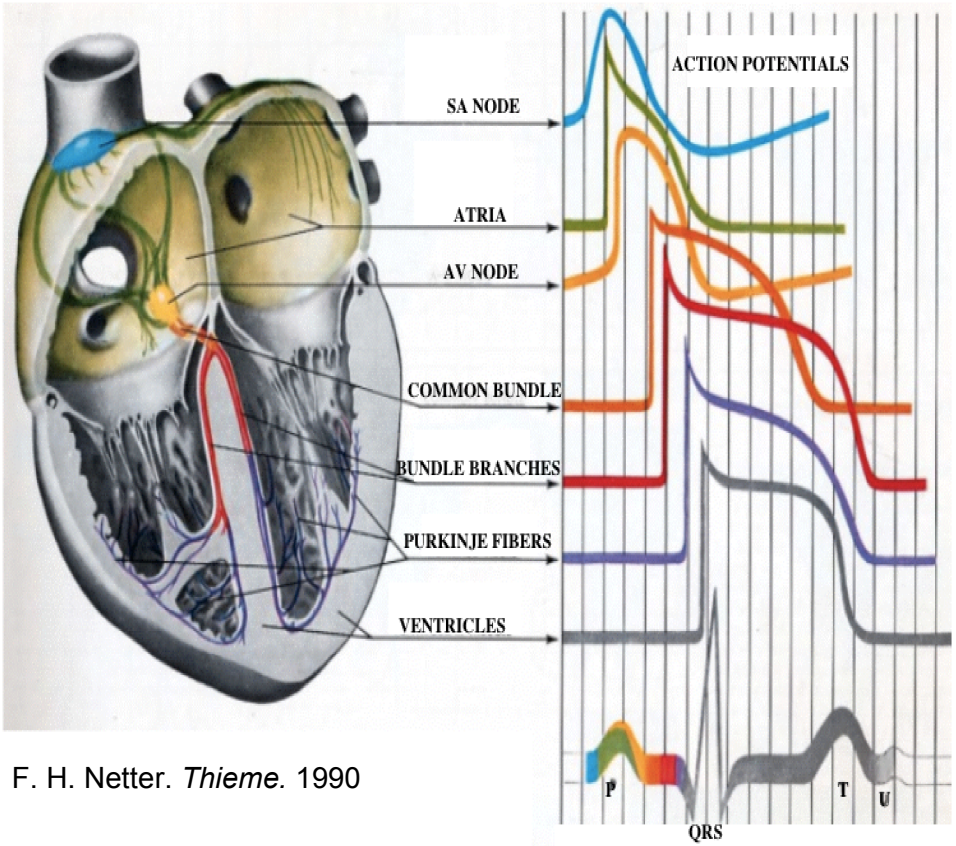
詳細は11月にソルトレイクシティで開催されるSC12にて発表予定

Blue Gene/Qにおける心臓シミュレーションの事例

Matthias Reumann

IBM Research – Australia

Cardiovascular disease (CVD)

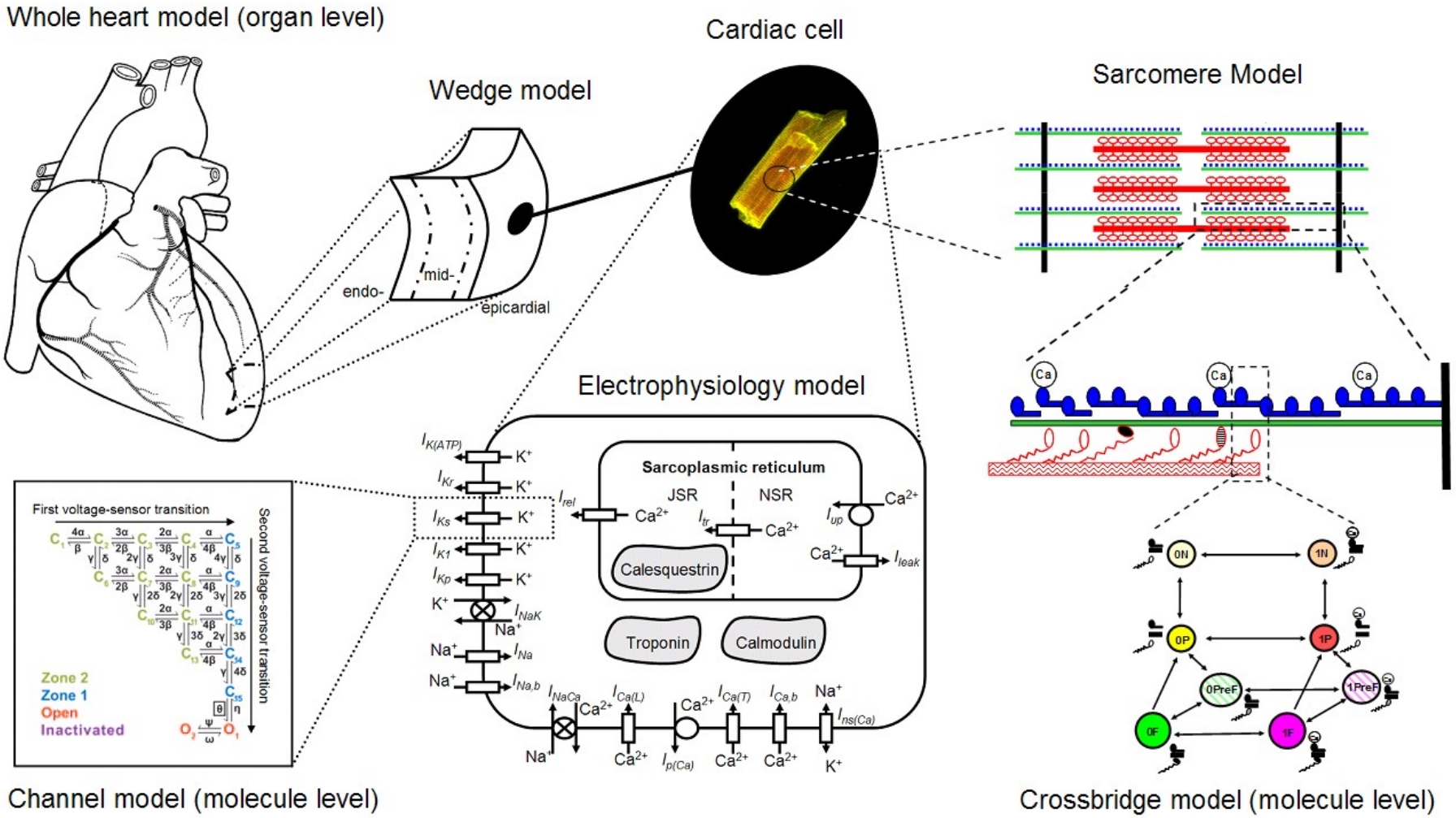


F. H. Netter. *Thieme.* 1990

- **Leading causes of death**
  (WHO report Sept. 2011)
- Research reduces deaths
- Cardiac models help understand mechanisms of CVD in a way that cannot be done neither in vivo nor in vitro.

Challenge: multi-scale cardiac simulations to model and understand the progression of disease

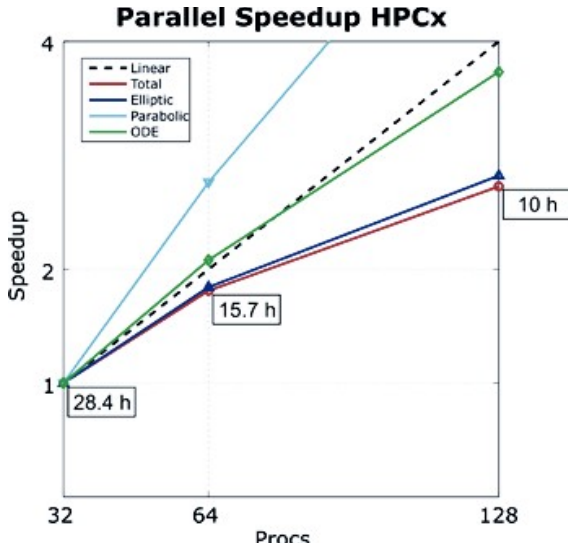
Multi-scale cardiac models – Molecules to Cells to Organ



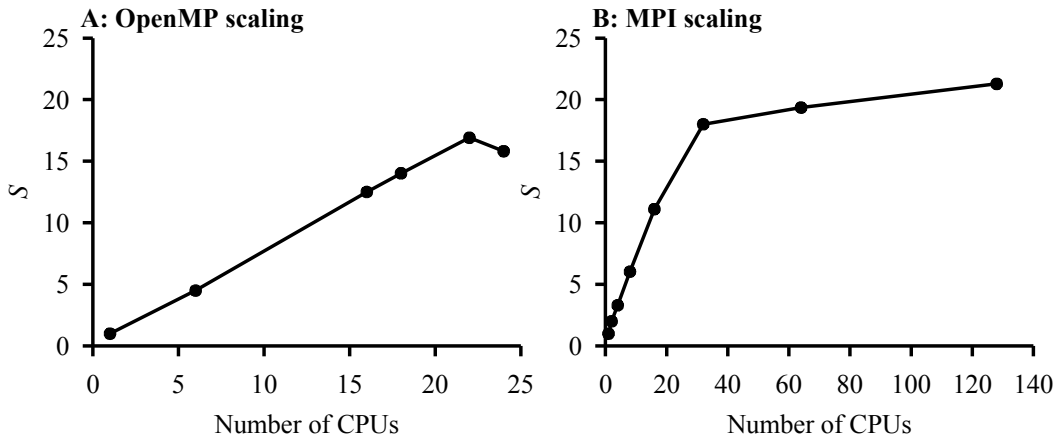
Silva and Rudy. 2005

Clancy and Rudy. 2005

5 years ago: Existing codes did not scale



Plank et al. Capability Computing. 2006



Kharche et al. Concurrency and Computation: Practice and Experience. 2008

First Author	Year	Elements	CPU used	CPU available	Simulation time	Run time	Run time [h]
Pavarino	2004	600 x 600 x 100	128	512	30 x 0.05 ms	163.6	1.36
Colli Franzone	**2007**	**200 x 200 x 50**	**36**	**92**		**21 - 24 h**	
Murillo	2004	512 x 512	128	128	200 ms	519 s	4.325
Saleheen	1998	156 x 256 x 32			0.004 ms	2.217 s/time step	
Potse	**2006**	**567 x 501 x 711**	**32**	**64**	**600 ms**	**39 h/51 h**	
Weber dos Santos	2004	0.64 x 0.64 x 2.56 cm	16	16	10000 steps	28254	235.45
Plank	**2007**	**862515**	**64**	**128**	**500 ms**	**6.4 h**	

3 – 4 years ago…

Full heart models require very long run times

World's Biggest Heart Model Simulated At Universite De Montreal

Main Category: Cardiovascular / Cardiology
Article Date: 21 Jan 2008 - 1:00 PST

email to a friend | printer friendly | view / write opinions | rate article | newsletters

Researchers from the Université de Montréal has used a supercomputer to conduct the largest-ever mathematical simulation of the electrical activity of a human heart - a 2 billion element model - to provide new insight into cardiac and other illnesses. Until recently, the world's largest simulated hearts had a few million elements at most. The UdeM simulation was up to 1,000 times more detailed than previous models and will enable new scientific discoveries that would never have been possible otherwise.

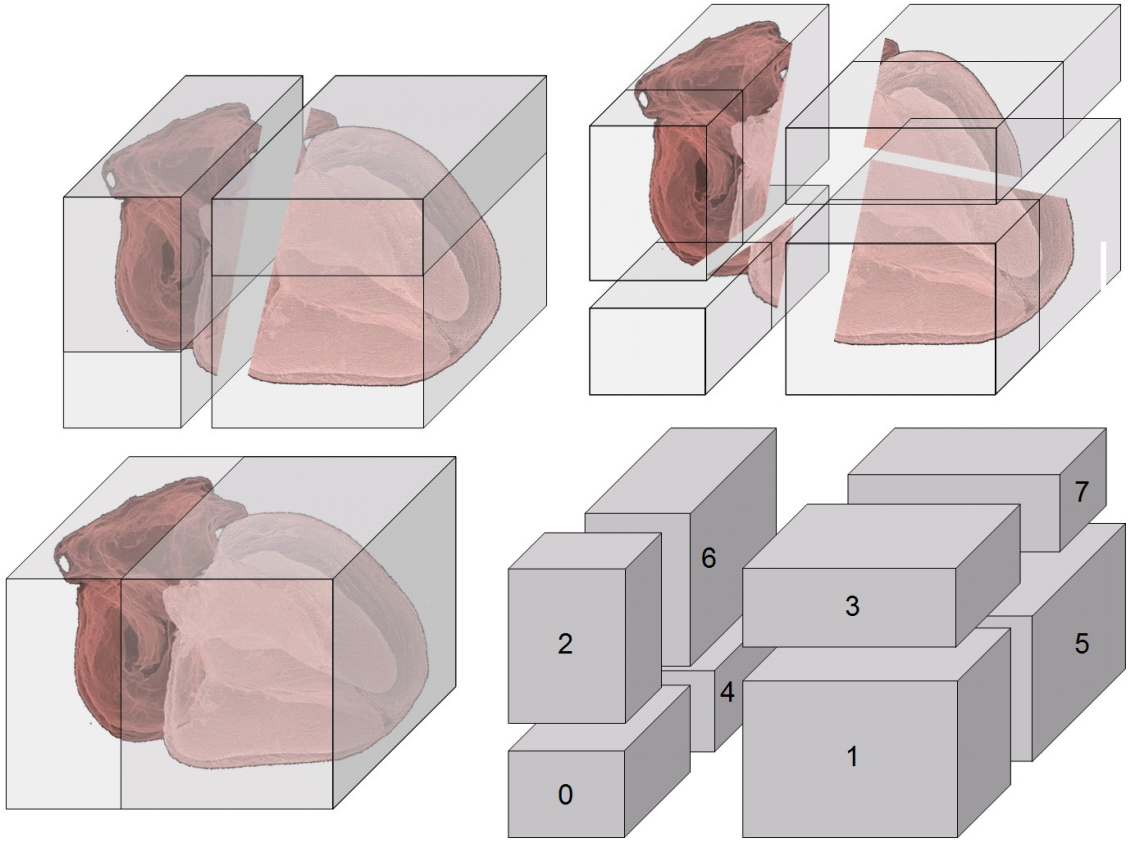
The computer on which the simulation was performed, a 768-processor SGI Altix 4700, is the largest shared-memory computing system in Canada. Operated by the Réseau québécois de calcul de haute performance (RQCHP), it is used by hundreds of Canadian researchers. Mark Potse and Alain Vinet, of the UdeM's Institute of Biomedical Engineering, routinely use 60 to 100 of these processors to run their simulations of the human heart. In late October, Potse and Vinet had the opportunity to use the entire SGI Altix system and its1.2TB of shared memory to solve the largest, most detailed heart model ever.

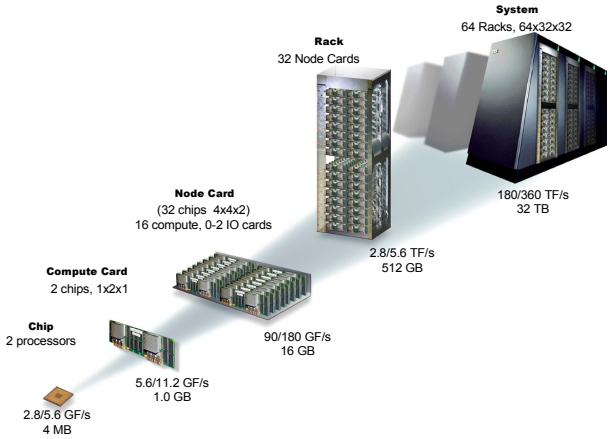
Two weeks to simulate full heartbeat

The researchers simulated five milliseconds of activation in a tissue block that included some properties of a real heart, such as fibre running in different directions. The simulation solved a system of two billion equations a dozen times. The test took two hours. A full heartbeat. Potse says. would take two weeks to

Methods – Orthogonal Recursive Bisection (ORB)

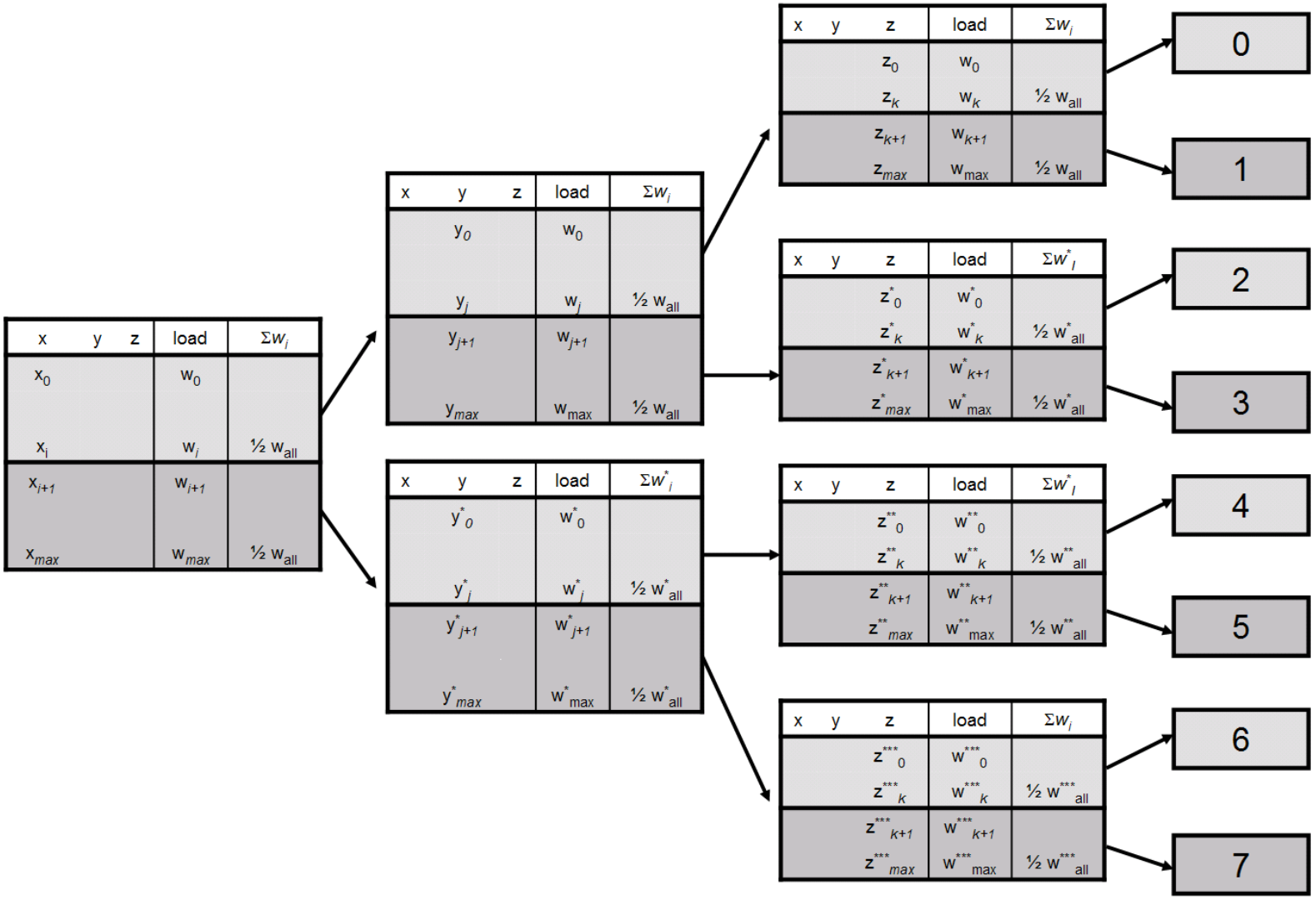
(Data decomposition adapted from molecular dynamics)





Reumann et al. In Proc IEEE EMBC 2008
Reumann et al. Biomed Tech 2011

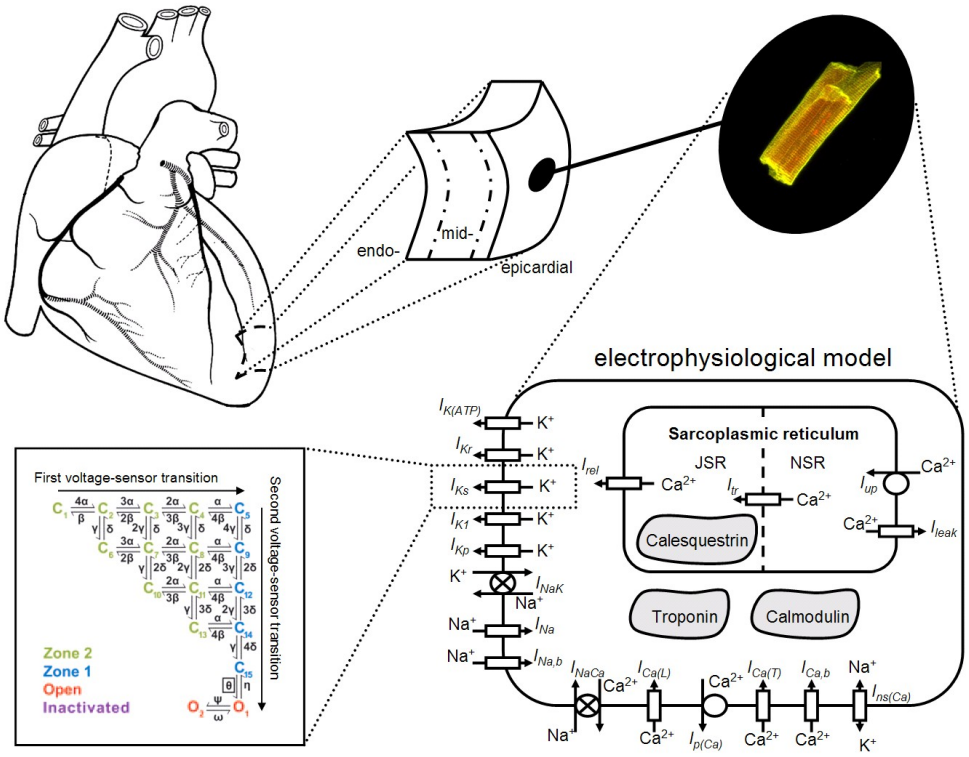
ORB Tree



Reumann et al. Biomed Tech 2011

Multi-scale, reaction – diffusion cardiac models

Reumann, Gurev, Rice. Personalized Medicine 2009



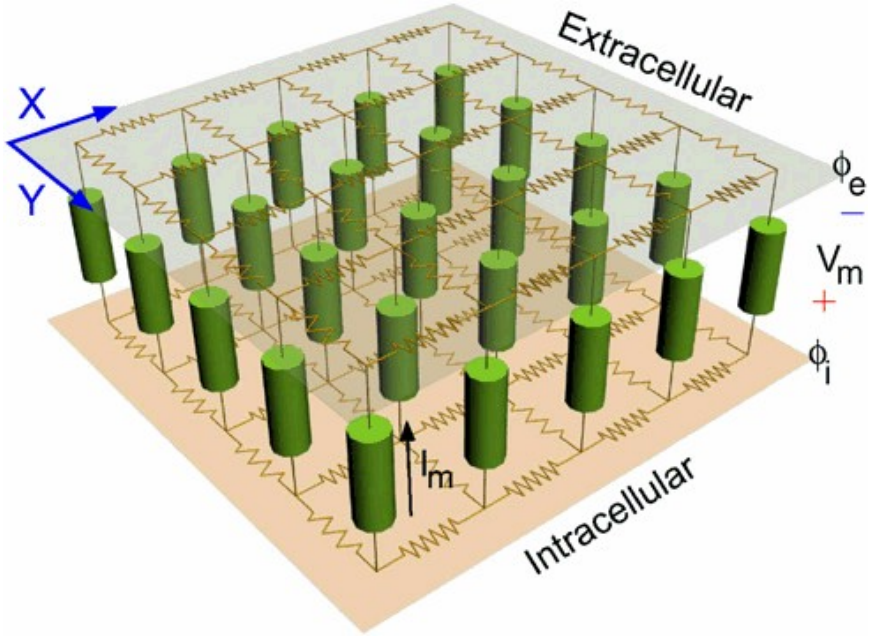
- **Non-linear stiff ODEs**
- **3-100 state variable per cell depending on model detail**
- **370 Mio tissue cells (0.1 mm cubic)**

$$\nabla \cdot (\sigma \nabla \phi) = -8\left(\sigma_i^{(1)} g^{1i} + \sigma_i^{(2)} g^{2i}\right)\phi_{(i)(j)} + \left(-\sigma_{i,j}^{(1)} g^{ji} + \sigma_i^{(1)}\overline{\Gamma^i} + 4\sigma_i^{(1)} g^{1i}\right)\phi_{(i-1)(j)} + \left(\sigma_{i,j}^{(1)} g^{ji} - \sigma_i^{(1)}\overline{\Gamma^i} + 4\sigma_i^{(1)} g^{1i}\right)\phi_{(i+1)(j)} + \left(-\sigma_{i,j}^{(2)} g^{ji} + \sigma_i^{(2)}\overline{\Gamma^i} + 4\sigma_i^{(2)} g^{2i}\right)\phi_{(i)(j-1)} + \left(\sigma_{i,j}^{(2)} g^{ji} - \sigma_i^{(2)}\overline{\Gamma^i} + 4\sigma_i^{(2)} g^{2i}\right)\phi_{(i)(j+1)} + \left(\sigma_i^{(2)} g^{1i} + \sigma_i^{(1)} g^{2i}\right)\phi_{(i+1)(j+1)} - \left(\sigma_i^{(2)} g^{1i} + \sigma_i^{(1)} g^{2i}\right)\phi_{(i+1)(j-1)} - \left(\sigma_i^{(2)} g^{1i} + \sigma_i^{(1)} g^{2i}\right)\phi_{(i-1)(j+1)} + \left(\sigma_i^{(2)} g^{1i} + \sigma_i^{(1)} g^{2i}\right)\phi_{(i-1)(j-1)} \quad (4.98)$$

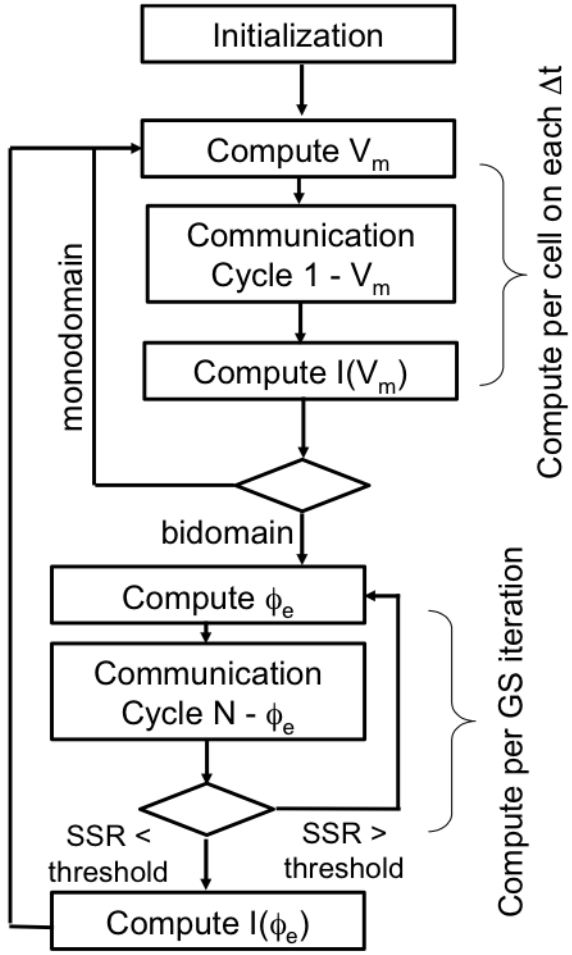
$$V_m^{t+\Delta t} = V_m^t + \frac{\Delta t}{A_m C_m}\left[I\left(V_m\right)^t + I\left(\phi_e\right)^t + I_{s2}\right] - \frac{\Delta t}{C_m} I_{ion}$$

Pullan, 2006

Methods - Solution of reaction-diffusion equations



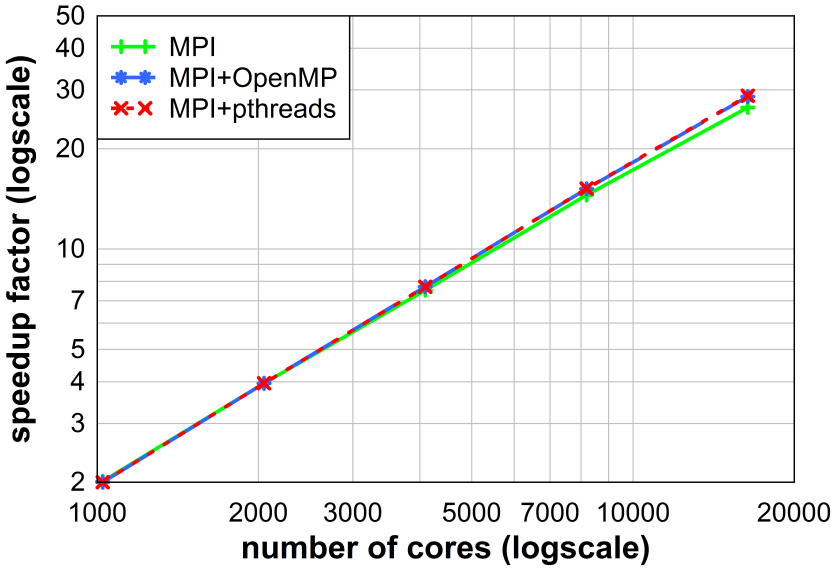
Vigmond et al. Progress in Biophysics and Molecular Biology. 2008

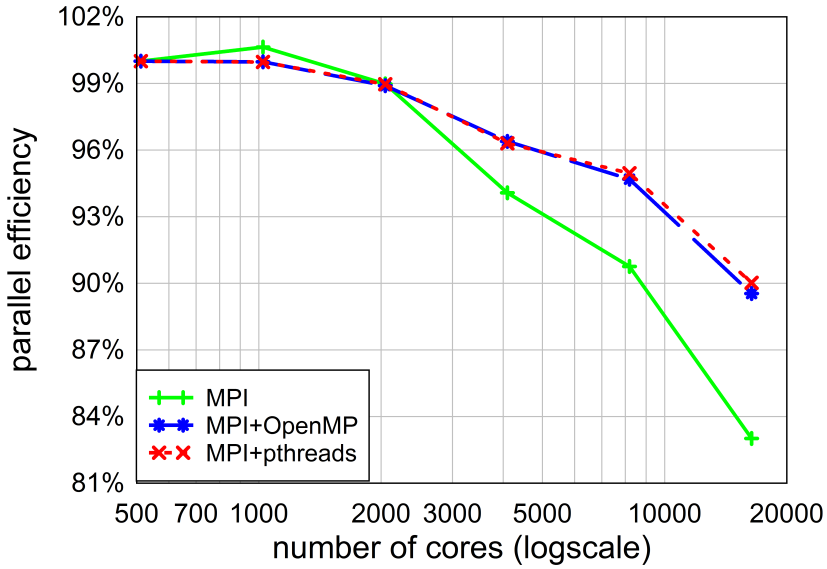


Reumann et al.
In Proc Computers in Cardiology. 2008

- **Monodomain - iso-potential extracellular space**
- **Bidomain - compute extracellular potential**

Scaling & Parallel Efficiency





Pope et al. IEEE TBME 2011

- A fast cardiac model of 1 second simulation time can run in 215 seconds wall clock time (Fitzhugh-Nagumo monodomain model with ~32.5M tissue cells on a 4 rack IBM Blue Gene/P supercomputer)
- 10 times faster on the next generation IBM Blue Gene/Q supercomputer

詳細は11月にソルトレイクシティで開催されるSC12にてゴードンベル賞のファイナリスト講演で

Arthur A. Mirin et al. , Toward Real-Time Modeling of Human Heart Ventricles at Cellular Resolution: Simulation of Drug-Induced Arrhythmias

まとめ

- **Blue Gene/Qは特別なスパコンではない**
  - 16GBのメモリとSMP，SMTのサポートによる高い互換性
  - 一般的なスパコンやPCクラスタと同等の開発環境
    - Linux開発環境
    - MPI+OpenMP
  - 一般的な分散メモリ並列コンピュータの最適化手法
- **ペタスケールシミュレーション**
  - 多くの分野でBlue Gene/Qの利用が始まっている
  - 未踏分野への応用の可能性