GPU-centric計算環境に向けて
～そしてGPUのグチ～

遠藤敏夫 (東京工業大学)

GPU・メニーコアは一時的なあだばなではない(希望的観測)

- 栄枯盛衰
  - NVIDIA Fermi
  - AMD FireStream
  - Intel SCC / Knights Ferry
  - GRAPE-DR
  - STI Cell
  - ClearSpeed
- 入れ替わりは認識しつつも、GPU・メニーコアの存在を前提としておくことが、リスクヘッジ上も重要
- GPUを計算の中心に置いてみる

疑問：待っていればx86 CPUがGPUなみのコア数になるのでしょ？

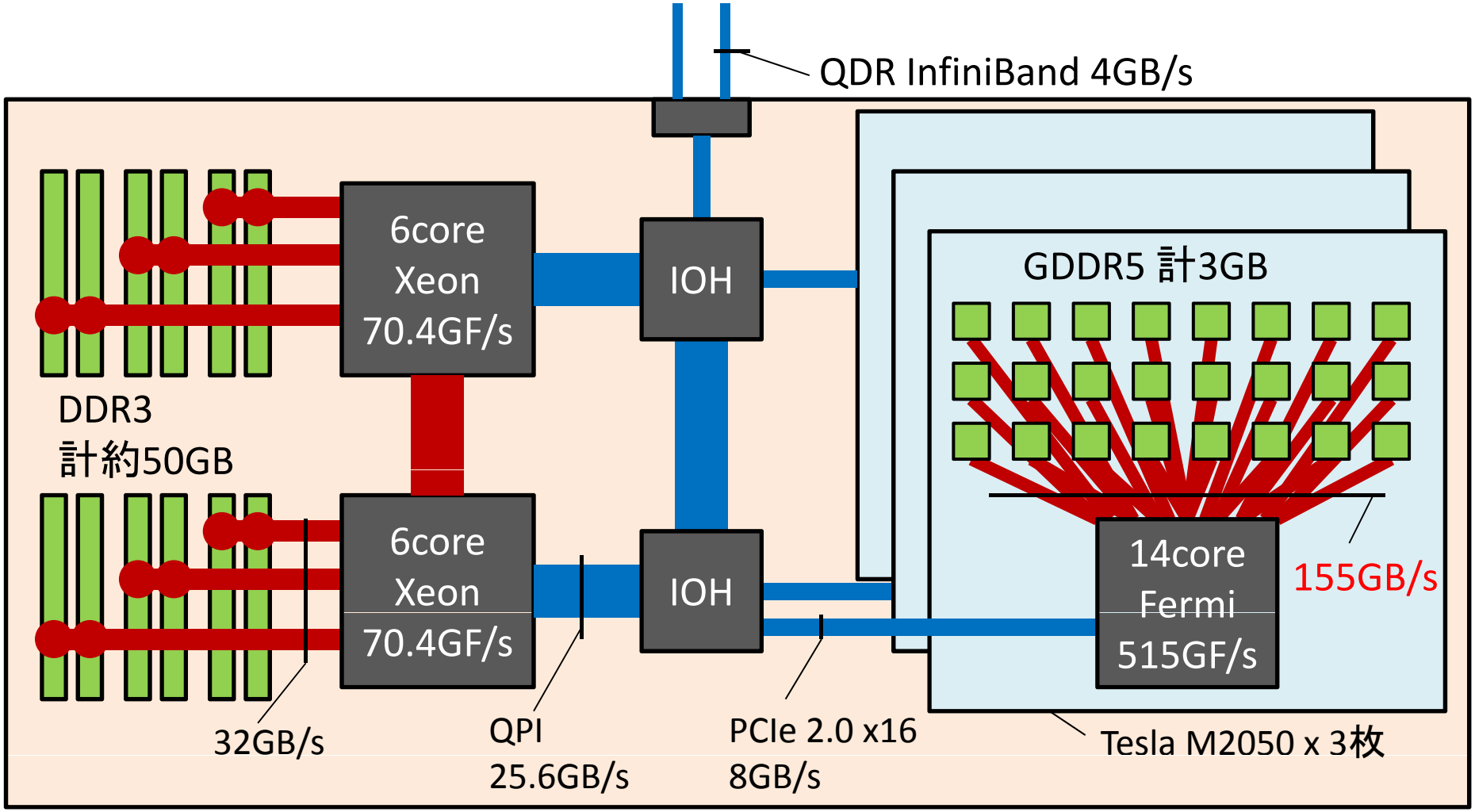
- それはそうかも。しかし・・・
  - Intel Xeon(Westmere)
    6core　x 4FP,　70.4GFlops, 32GB/s mem BW
  - NVIDIA Tesla M2050
    14core x 32FP, 515GFlops, 155GB/s mem BW

x5~7の将来

数年分の「将来」を買うことはできる ⇒ スパコンではそこが生命線になりうる

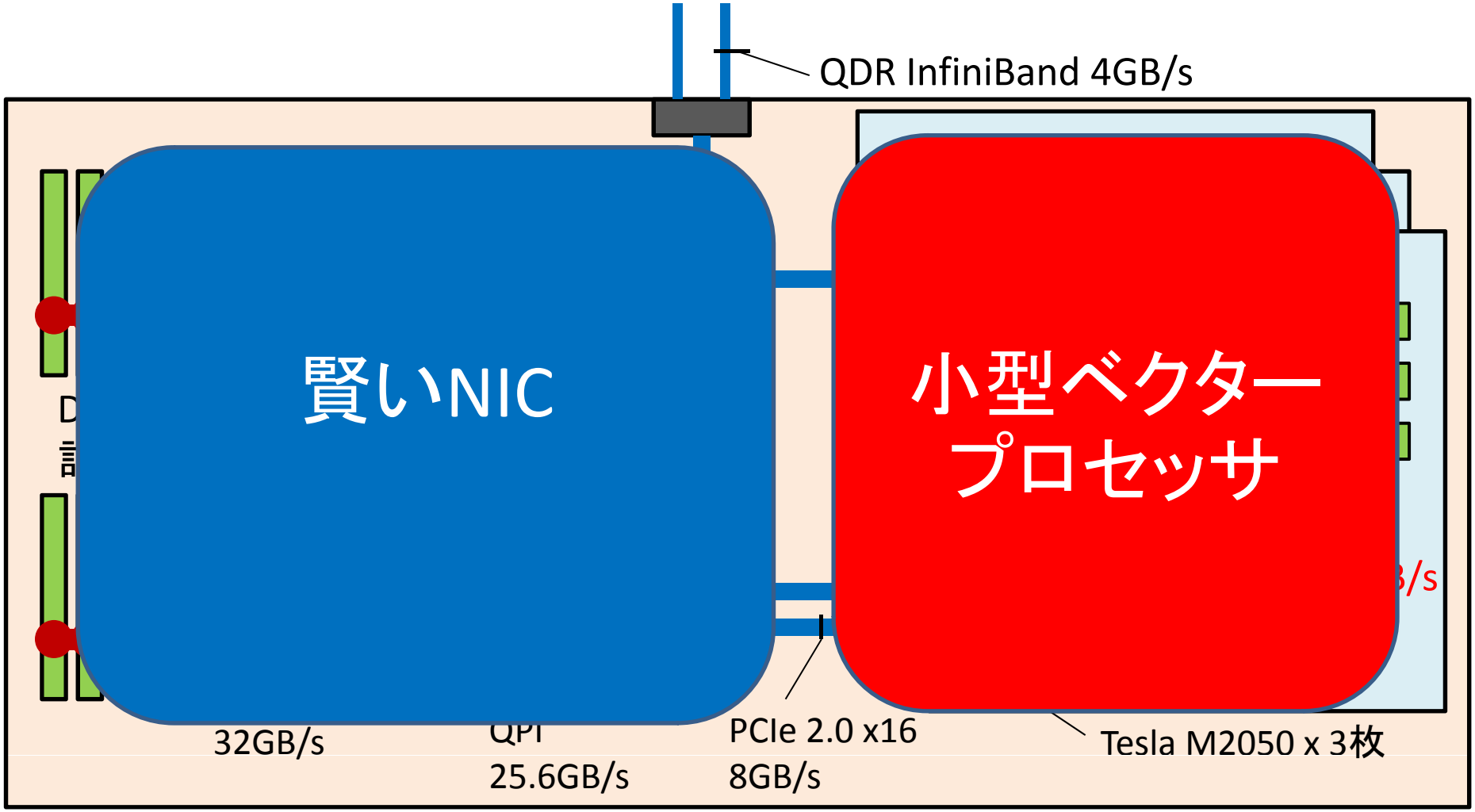
- 中国Nebula (2010/6)
  - Peak 3PFlops, C2050 4600GPUs
- TSUBAME 2.0(2010/11)
  - Peak 2.4PFlops, M2050 4200GPUs

2CPU+3GPUを持つTSUBAME2ノード



5年後にはソケット数5～6倍、各パラメータを5～6倍???

GPUを中心に置いてみる



あるべきソフトウェアスタック

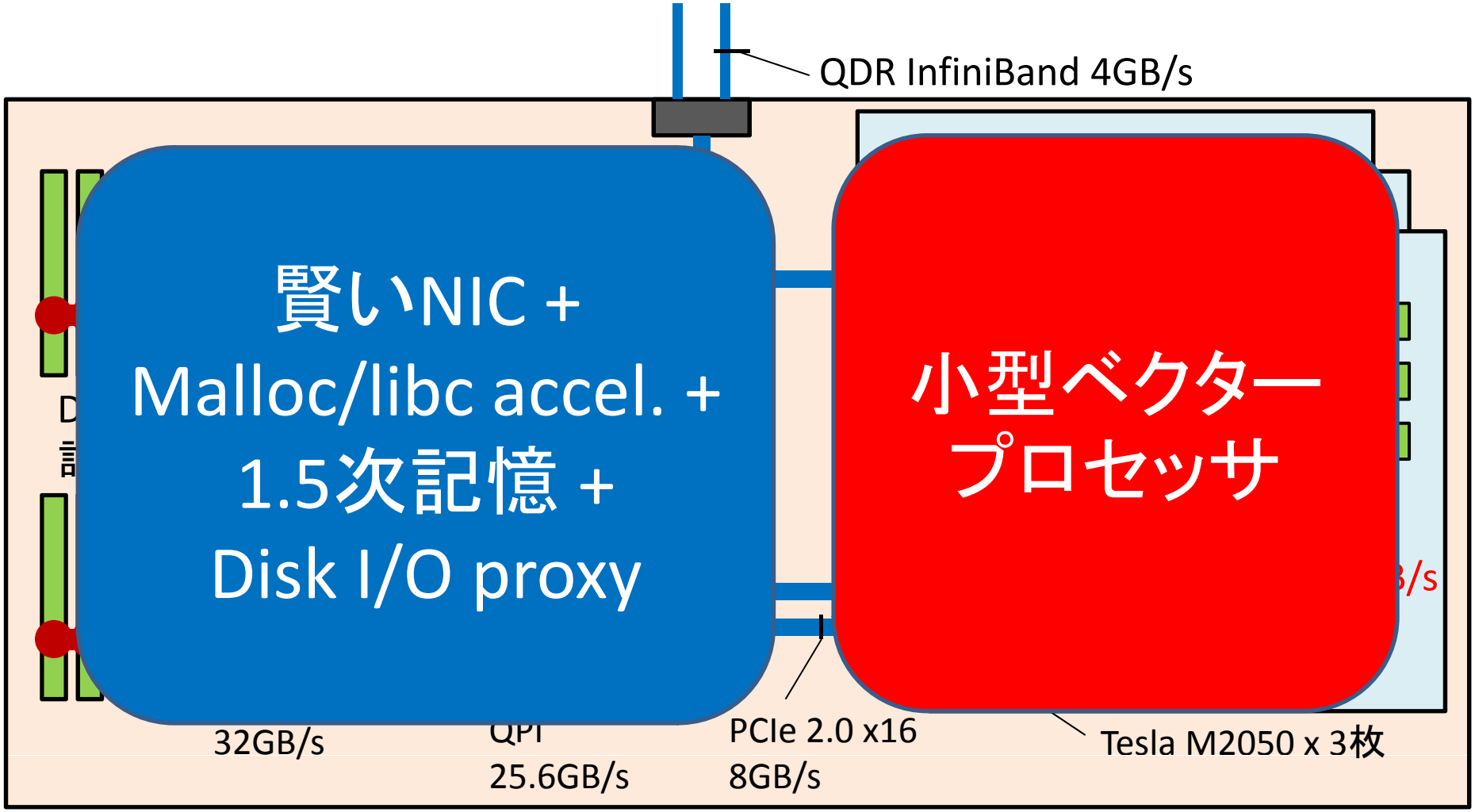
- 高局所性アルゴリズム
- プロセッサに適合する自動チューナー
- 細粒度スレッド+SIMD+通信隠ぺいコンパイラ
- Libcと同等レイヤーのライブラリ
- 1.5次記憶への高速SWAP
  - GPUメモリはやはり小さすぎ
- 高速GPU ⇔ GPU通信レイヤー
- システム最適化・マイグレーション含むスケジューラ

しかし、こんなことすらできない

- (GPUから呼べる)mallocすらない
  - Libc, I/O関係はほぼ全滅
  - 最近のCUDAでprintf可能
- コア間バリアがない、カーネル内並列度変更がない・・・
  ⇒ mallocやsafe pointにおいてCPUに制御を戻す。CPS変換コンパイラ？

- 通信はGPU⇔CPU, CPU⇔CPU, CPU⇔GPUの3hop
  - へたするとCPU上のcopyが入り5hop
  ⇒ Hop数最小限の通信ライブラリ, PGAS runtime
- 当然SWAPはなし
- 当然GPU間マイグレーションはなし
  - スケジューリング選択肢が限られる
  ⇒ GPU stateのチェックポイント研究@SWoPP
- ついでに、topコマンドもなし

CPUにいろいろ頼むとこうなる



疑問:それってNVIDIAがやるべきことでは？

- 多くはそう。しかし共同研究により世の中は進む
  - NVIDIA+PGIでアノテーテョン型コンパイラ
  - NVIDIA+Mellanoxで”GPU direct”
  - CUDA COE
  - 日本のセンターのプレゼンスを高める
- リスクヘッジ
  - TSUBAME1 ClearSpeedのためのスケジューラのworkaroundが、ほぼそのままNVIDIA GPUに使えた
  - 生き残るのは、情報開示する企業のような気も

まとまらないまとめ

- GPUはX86 CPUと違う進化過程をたどったプロセッサ
  - libcはないのにすでにマルチコア+HW スレッド
- 変わったプロセッサでも差別せずに・かつ冷静に研究しましょう
  - 区別・選別・多様性は必要