学際大規模情報基盤共同利用・共同研究拠点　萌芽型共同研究　採択課題
EX18317（東京大学情報基盤センター推薦課題）
10th Symposium
Joint Usage / Research Center for Interdisciplinary Large-scale Information Infrastructures

高柳雅俊（山梨大学総合研究部）

ハイブリッドクラスタシステムにおけるタイルQR分解のタイルサイズチューニング

目的：CPU/GPUクラスタシステム向けの高速なQR分解ルーチンの実装

CPU/GPUクラスタシステム

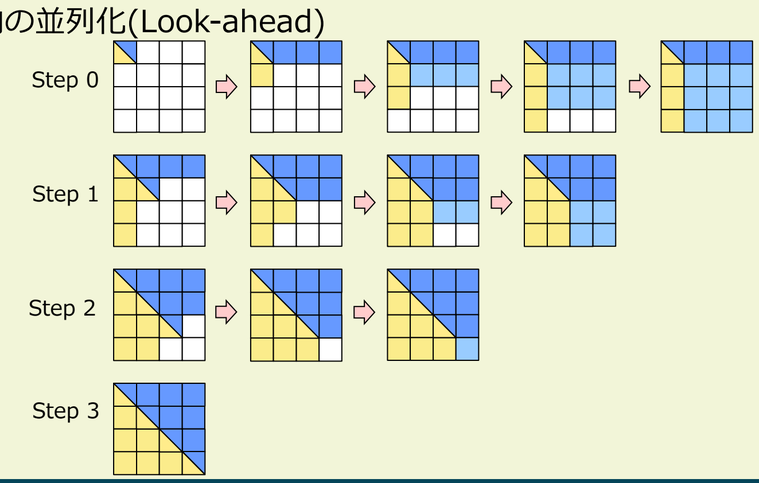
- GPUを搭載したクラスタシステム → 今後の主流
  - 高い電力性能比
- しかし、CPU/GPUクラスタシステムの数値線形代数ライブラリは存在しない
  - ScaLAPACK ← マルチコアCPUクラスタ向け
- GPUメモリはホストメモリと比べ小さい→大規模な行列に対応できない
- GPU間は高速な通信路が用意されているが、ノード間の通信が遅い

November 2017 The Top500

Top 500 Rank	Rmax (TFLOPS)	Total Power	Computer
1	93,014.6	15,371	Sunway TaihuLight, Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway
2	33,862.7	17,808	Tianhe-2 (MilkyWay-2), TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P
3	19,590.0	2,272	Piz Daint, Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100
4	19,135.8	1,350	Gyoukou, ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz
5	17,590.0	8,209	Titan, Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x

タイルアルゴリズムの導入

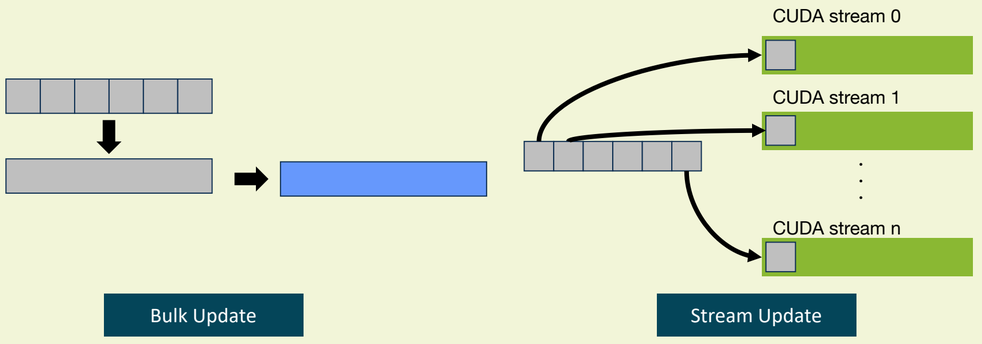
- **タイルアルゴリズム**
  - 行列を小行列（タイル）に分割
  - タイルごとにタスクを実行 → 大量の細粒度タスク生成可能
- ホストメモリ上に行列データを保持
  →大規模行列に対応，GPUメモリ使用量削減
- OpenMP 4.0 task構文depend節による動的タスクスケジューリング
  →Step方向の並列化(Look-ahead)



これまでの研究成果

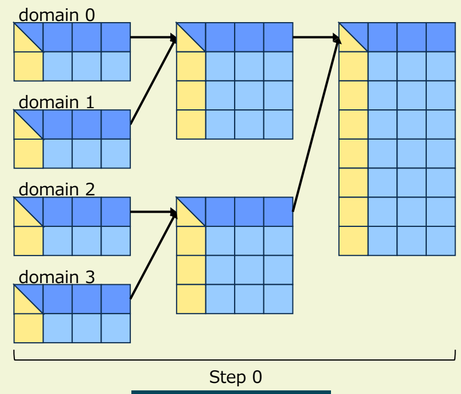
CPU/GPUハイブリッド実装

- 分解タスク:逐次性強，memory-bound→CPU側で処理
- 更新タスク:並列性高，compute intensive→GPU側で処理
- 2種類の更新手法
  - **Bulk Update**:大きいデータを与え，単一GPUカーネル内で並列実行
  - **Stream Update**: 複数のGPUタスクを同時実行させ稼働状態を維持



通信削減

- Communication Avoiding tile QR
  - タイルをドメインに分割しQR分解を1Step実行
  - 各ドメインの最上タイル行をマージ
  - 縦方向がドメインごとの処理→並列性の向上
  - マージ処理時の通信のみ→通信の削減

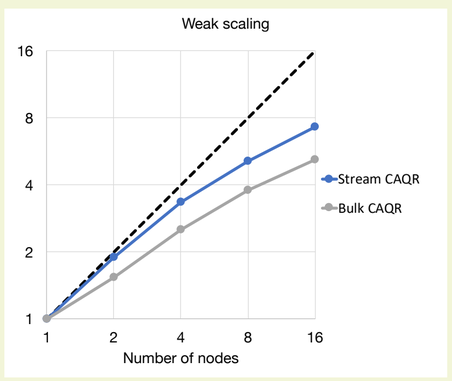


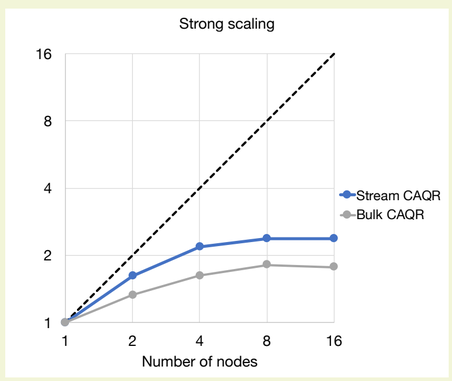
通信削減QR分解

これまでの研究成果っづき

速度測定結果

- Reedbush-H@東京大学情報基盤センターで性能測定
- Weak Scaling:1ノードあたりの行列サイズ81920x81920
  - 8ノードまでは良スケールが得られた
  - ノード数が増加すると効率が落ちる
    →16ノードでは理論性能の約50%
- Strong Scaling:行列サイズ102400x102400固定
  - 16ノードで約2倍の並列化効率
  - 縦方向の依存性のため並列性能が得にくい

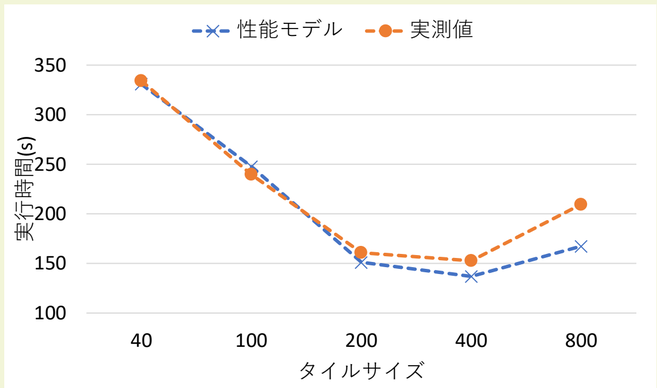




今後の研究計画

CPU/GPUクラスタシステム実装のタイルサイズチューニング

- 選択した**タイルサイズによって性能が大きく変化**
- タイルサイズチューニングのための**性能モデル構築**
  - CPU・GPUで異なる最適タイルサイズ
    CPU：タイルサイズ小 ⇔ GPU：タイルサイズ大
  - 性能モデルの構築は**困難**
    - 動的スケジューリングによるタスクの非同期実行
    - タスクと通信のオーバーラップ



京コンピュータ上の実装の性能モデルと
実測値(64ノード使用時)

UNIVERSITY OF YAMANASHI