スーパーコンピュータ「富岳」における Graph500ベンチマークの幅優先探索の性能評価

中尾 昌広[1]、上野 晃司[2]、藤澤 克樹[3]、児玉 祐悦[1]、佐藤 三久[1]

1. 理化学研究所 計算科学研究センター
2. 株式会社フィックスターズ
3. 九州大学 マス・フォア・インダストリ研究所

第175回HPC研究会（オンライン）、2020年7月31日

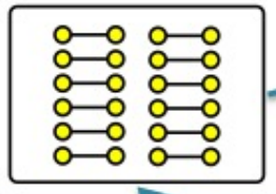
もくじ

大規模グラフ解析の利用方法と応用分野

大規模グラフ解析の応用分野

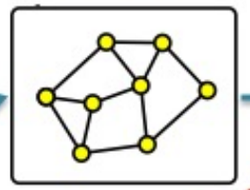
交通ネットワーク
ソーシャルネットワーク
サイバーセキュリティ
バイオインフォマティクス
脳科学
防災計画策定

事象同士の関係
(Relationships)



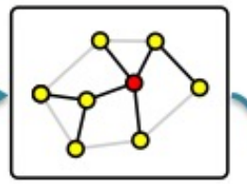
Step1

グラフ



Step2
グラフ解析

解析結果



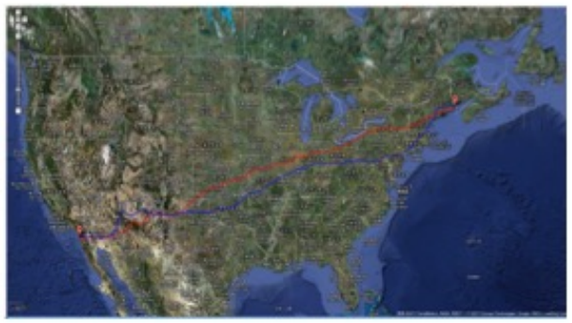
Step3
分析と理解

- 並列グラフ探索 (幅優先探索)
- 最適化 (最短路, 最大フロー, 最小費用フロー)
- クラスタリング (グラフ分割, コミュニティ抽出)

Twitter

ソーシャルネットワーク
6,160万点 & 14億7千万枝

全米道路ネットワーク
2,400万点 & 5800万枝



サイバーセキュリティ
150億/日 のアクセスログ

ニューラル・ネットワーク 890億点 & 100 兆枝

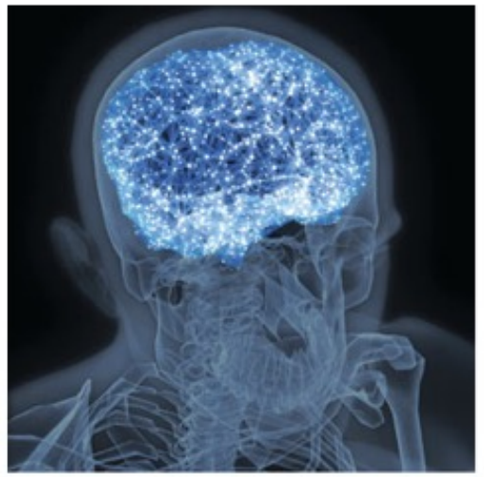


Image: Illustration by Mirko Ilic

参照：http://opt.imi.kyushu-u.ac.jp/lab/jp/activities.html

Graph500の誕生 https://graph500.org

- 大規模グラフの処理性能を評価するランキング

- 2010年から年に2回（6月と11月）にリストが更新される
  - **幅優先探索（BFS：Breadth-First Search）**
  - 単一始点最短経路（SSSP：Single-Source Shortest Path）

- クロネッカーグラフという人工グラフ
  - 一部の頂点が他の多くの頂点と繋がっている一方、他の多くの頂点はごく僅かな頂点としか繋がっていない（スケールフリー性）
  - SNSのデータも同じような性質を持つことが知られている

我々の取り組み

- 様々な工夫をBFSに対して行ってきた[1-3]

[1] 上野ら，大規模分散メモリ環境におけるハイブリッドBFSの最適化，Vol.2014-HPC-146 No.21

[2] 上野ら，Bitmap Hybrid BFS の改良と「京」における性能評価，Vol.2016-HPC-153 No.10

[3] Ueno et al, Efficient Breadth-First Search on Massively Parallel and Distributed-Memory Machines, DOI 10.1007/s41019-016-0024-y, 2017

スーパーコンピュータ「京」を用いた結果により、
2014年6月と2015年6月～2019年6月の
Graph500において、通算10期の世界1位を獲得

そして「京」は2019年8月に運用終了。

2020年6月のGraph500で1位を獲得

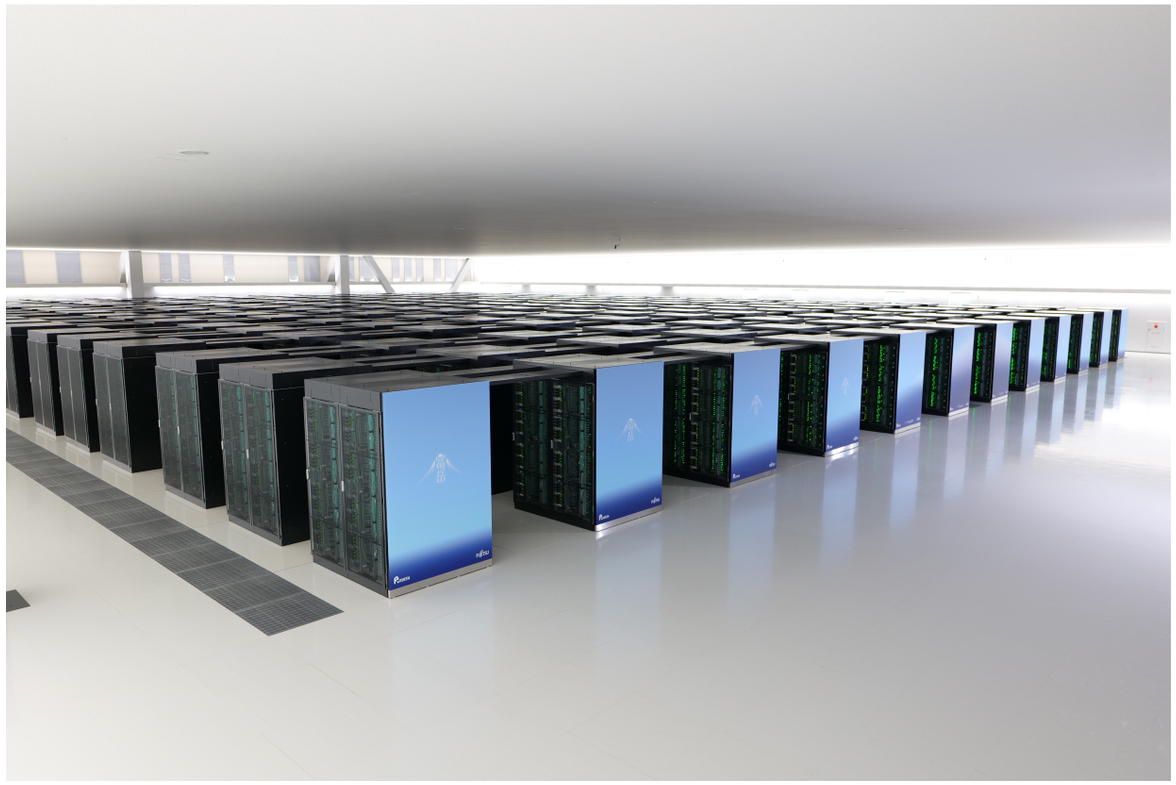
- **「京」（82,944ノード）**の後継機の**「富岳」の一部（92,160ノード）**を用いた（「富岳」の全系は158,976ノード）
- Performance Metric: TEPS (Traversed Edges Per Second)
  - 1秒間に探索できるエッジの数
  - 1GTEPS = 1秒間に10億本のエッジを探索できる

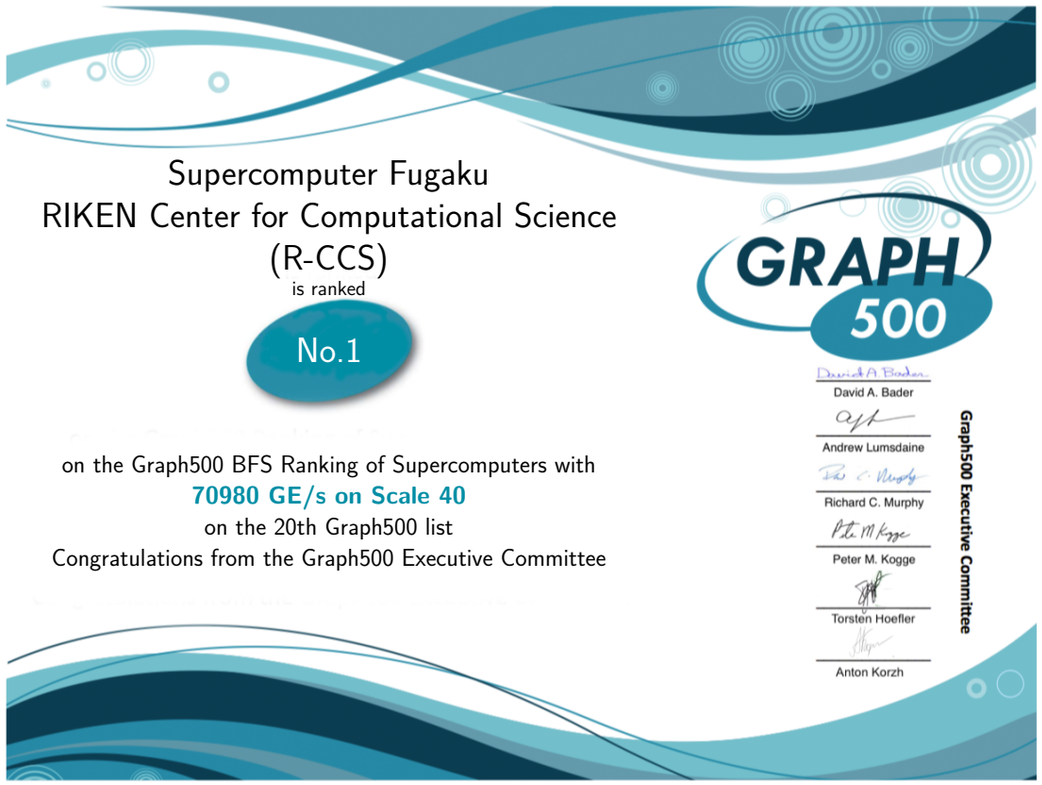
	June 2019		November 2019		June 2020	
	NAME	GTEPS	NAME	GTEPS	NAME	GTEPS
1st	**K computer**	**31,302**	Sunway TaihuLight	23,756	**Supercomputer Fugaku**	**70,980**
2nd	Sunway TaihuLight	23,756	Sequoia	23,751	Sunway TaihuLight	23,756
3rd	Sequoia	23,751	Mira	14,982	Summit	7,666
4th	Mira	14,982	Summit	7,666	SuperMUC-NG	6,279
5th	SuperMUC-NG	6,279	SuperMUC-NG	6,279	Cori	2,562

- **「京」**の結果は31,302GTEPSに対し、**「富岳」**の結果は70,980GTEPS
  - 2.27倍の性能向上を達成

本発表の目的

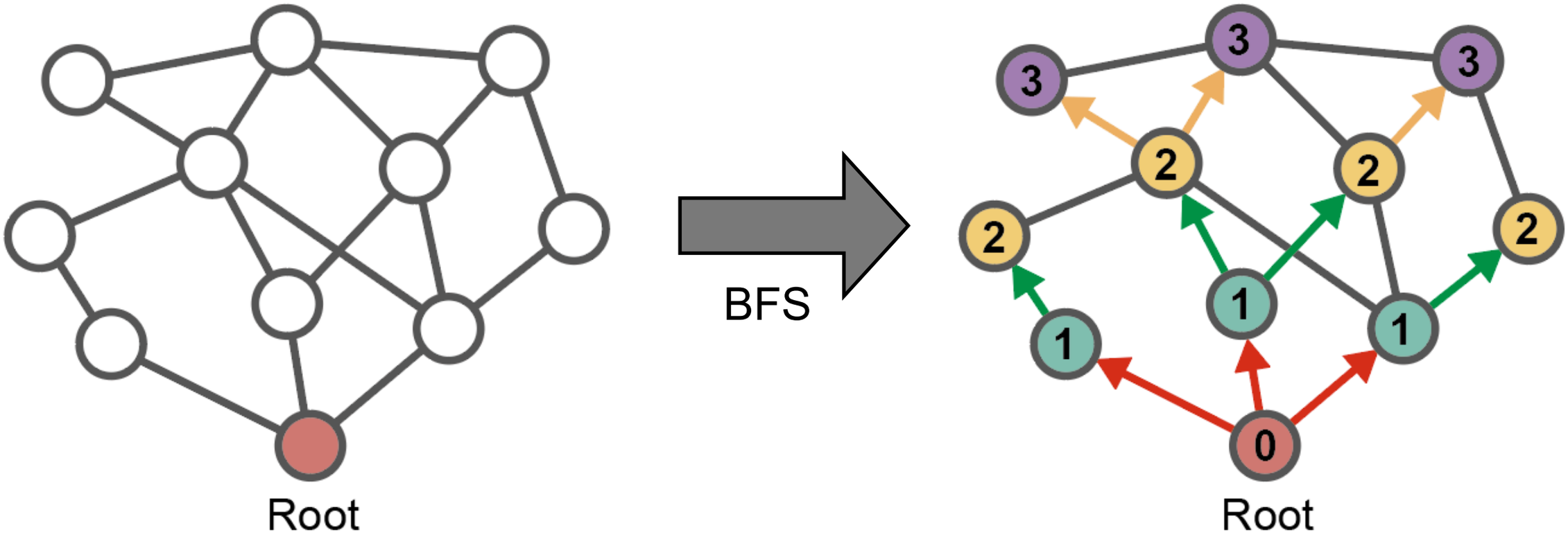
- Graph500の投稿に用いたBFSと、「富岳」におけるBFSの性能チューニングについて述べる





もくじ

Graph500におけるBFS



Input：グラフと出発点（Root）

Output：BFSツリー

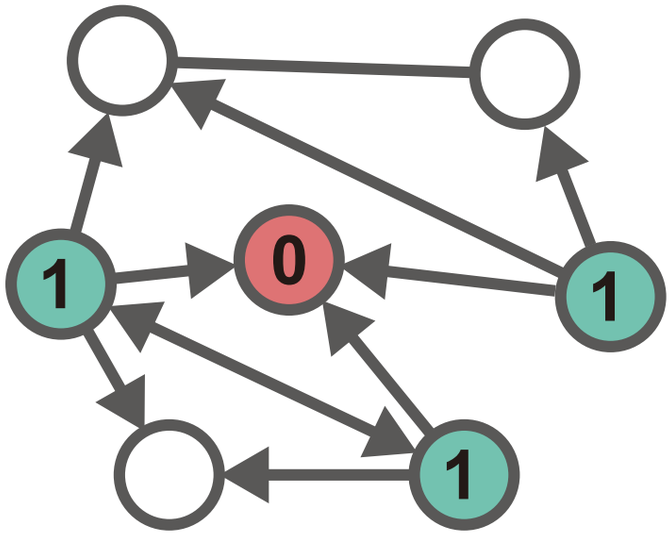
- データ構造やBFSのアルゴリズムは自由

Hybrid-BFS

[Beamer, 2012] Scott Beamer et al. Direction-optimizing breadth-first search, SC '12

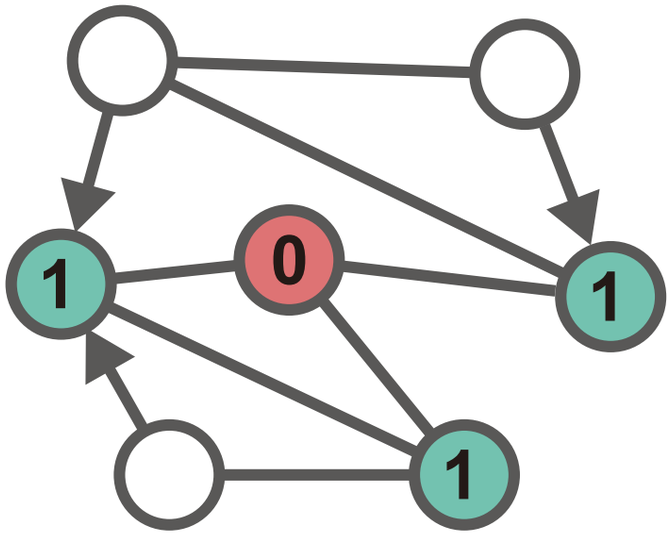
- Graph500で扱うような直径が小さいグラフに向いている
- Top-downとBottom-upを切り替えながらBFSを行う
  - Graph500のBFSの中盤では、**探索中の頂点**が爆発的に増えるため、Top-downでは効率が悪いから

Top-down



探索中の頂点から、
未探索の頂点を探す

Bottom-up



未探索の頂点から、
探索中の頂点を探す

チェックの回数（SCALE=26）

- 問題サイズ：SCALE
  - 2^{SCALE} 個の頂点と、頂点数x16本のエッジを持つグラフ
  - SCALE=26の時、約6711万個の頂点と約11億本のエッジ

	Top-down	Bottom-up	Hybrid-BFS
0	**2**	2,103,840,895	**2**
1	**66,206**	1,766,587,029	**66,206**
2	346,918,235	**52,667,691**	**52,667,691**
3	1,727,195,615	**12,820,854**	**12,820,854**
4	29,557,400	**103,184**	**103,184**
5	82,357	**21,467**	**21,467**
6	**221**	21,240	**221**
Total	2,103,820,036	3,936,062,360	**65,679,625**
Rate	100.00%	187.09%	**3.12%**

Credit: Yasui

2D Hybrid-BFS

[Beamer, 2013] Scott Beamer, et. al. Distributed Memory Breadth-First Search Revisited: Enabling Bottom-Up Search. IPDPSW '13.

- 2次元プロセスグリッド（R x C）に隣接行列を分散

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,C} \\ \vdots & \ddots & \vdots \\ A_{R,1} & \cdots & A_{R,C} \end{pmatrix}$$

- 列プロセス内および行プロセス内のみに通信が発行される
  - 列プロセス：Allgatherv
  - 行プロセス：Alltoallvとsend/recv
  - RとCの値は近いほど、総通信量は小さい

2D Hybrid-BFSの改良

[Ueno, 2017] Koji Ueno et al. Efficient breadth-first search on massively parallel and distributed-memory machines. Data Science and Engineering, Vol. 2, pp. 22–35

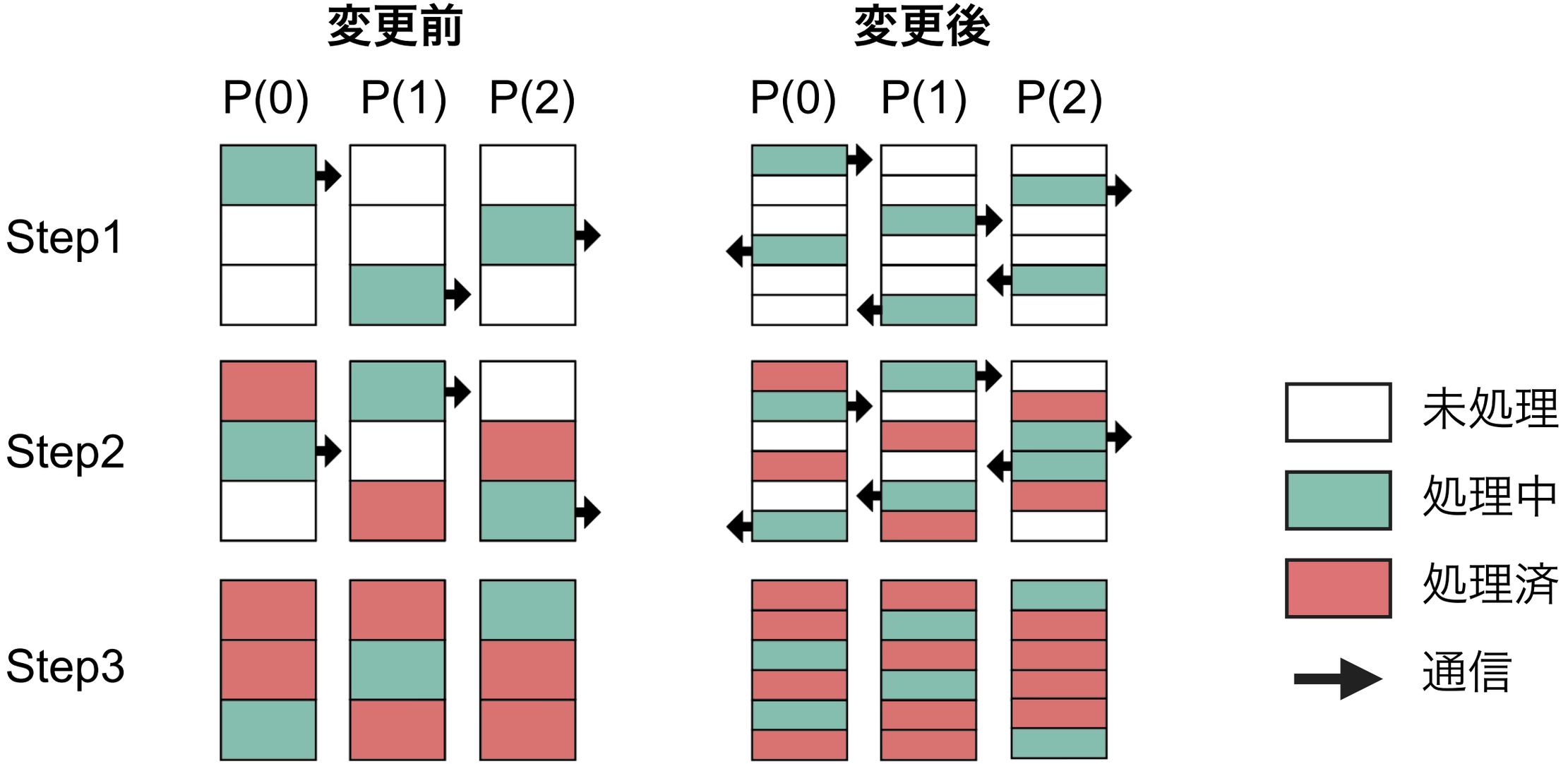
- 2D Hybrid-BFSをベースに改良（詳細は予稿集と上論文を参照）

  https://github.com/suzumura/graph500

- ローカル計算最適化
  - 省メモリかつ効率良く情報を取り出せる疎行列表現の適用
  - メモリの局所性を利用するための頂点番号の並べ替え
  - スレッド並列におけるロードバランス最適化
- 通信最適化
  - 通信の一部を省くことができる隣接行列における分割方法の変更
  - 通信量削減のための頂点濃度によるデータ構造の切り替え
  - 非同期send/recvの一部を集合通信に変更
  - 2方向同時通信の実装と通信と計算のオーバラップ

2方向同時通信の実装と通信と計算のオーバラップ

- 「富岳」やBlueGene/Qなどで用いられているトーラス形状の直接網を有効利用するため、非同期send/recvの隣接通信を2方向同時に行う
- 処理を分割することで、通信と計算のオーバラップを促進する

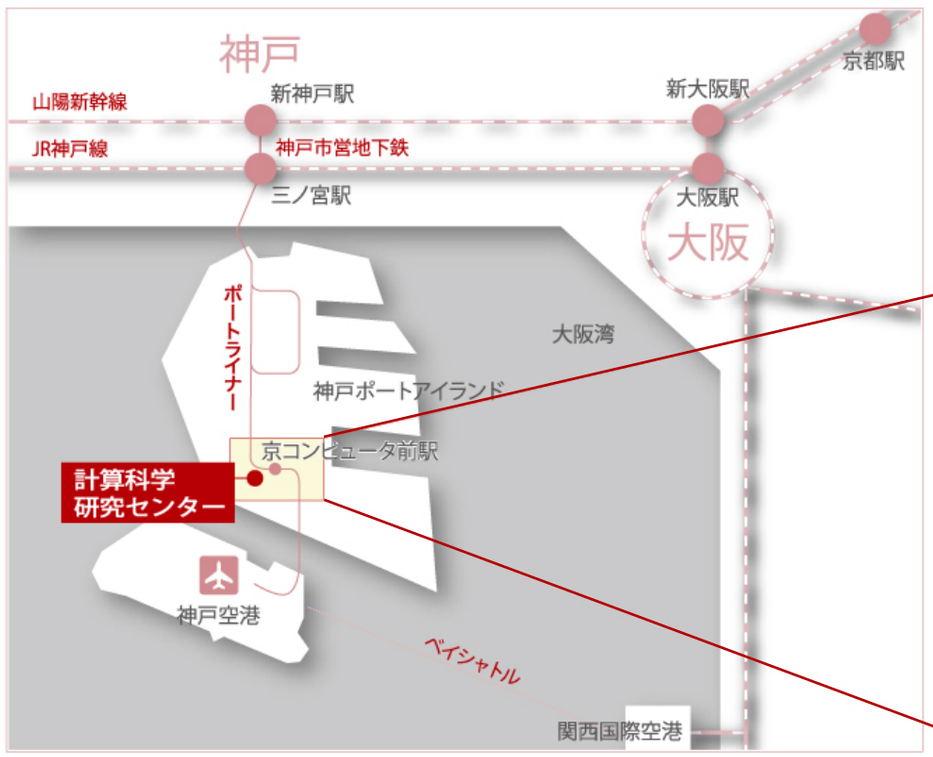


もくじ

- 背景
- 幅優先探索（BFS）の概要
- 「富岳」におけるBFSの性能チューニング
- まとめと今後の課題

スーパーコンピュータ「富岳」

- 神戸の理化学研究所 計算科学研究センターに設置
- 158,976台の計算ノード
- 2021年度に共用開始予定。現在は共用前評価環境であるため、
  本発表の結果は共用開始時の性能を保証しないことにご注意ください

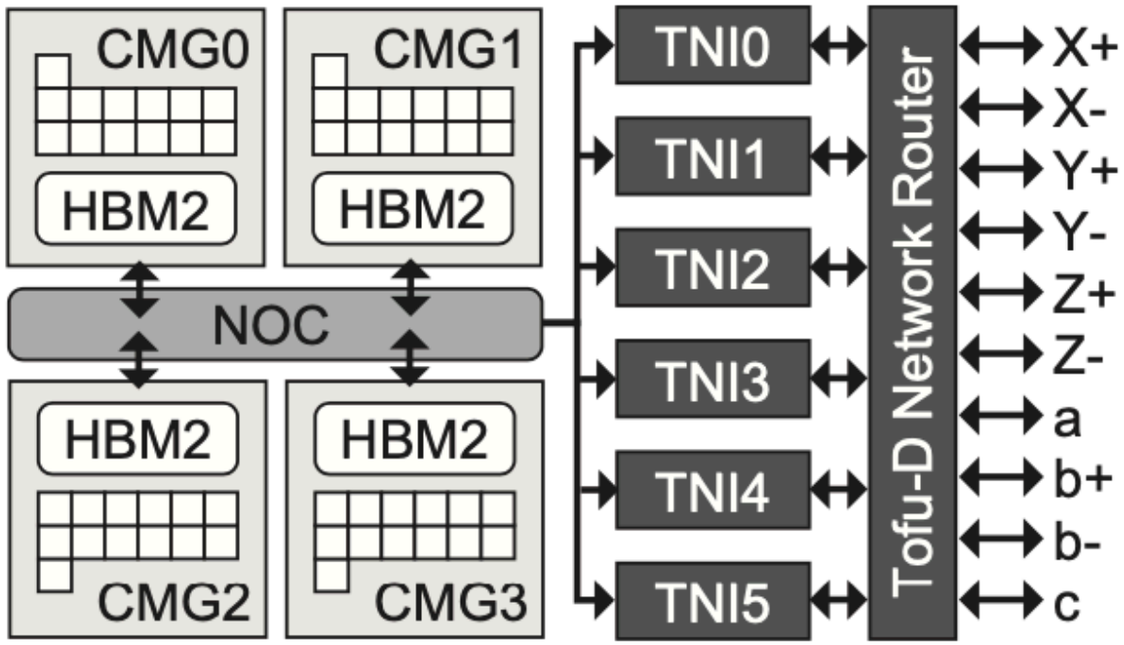


「富岳」の計算ノード

計算ノードのスペック

CPU	A64FX, 48+2/4 cores, 2.0/2.2GHz, 3,072/3,379GFlops(DP)
Memory	HBM2 32 GiB, 1,024GB/s
Interconnect	Tofu-D, 6-dimensional mesh/torus 28.05Gbps × 2 lane ×10 ports

A64FXプロセッサのブロック図

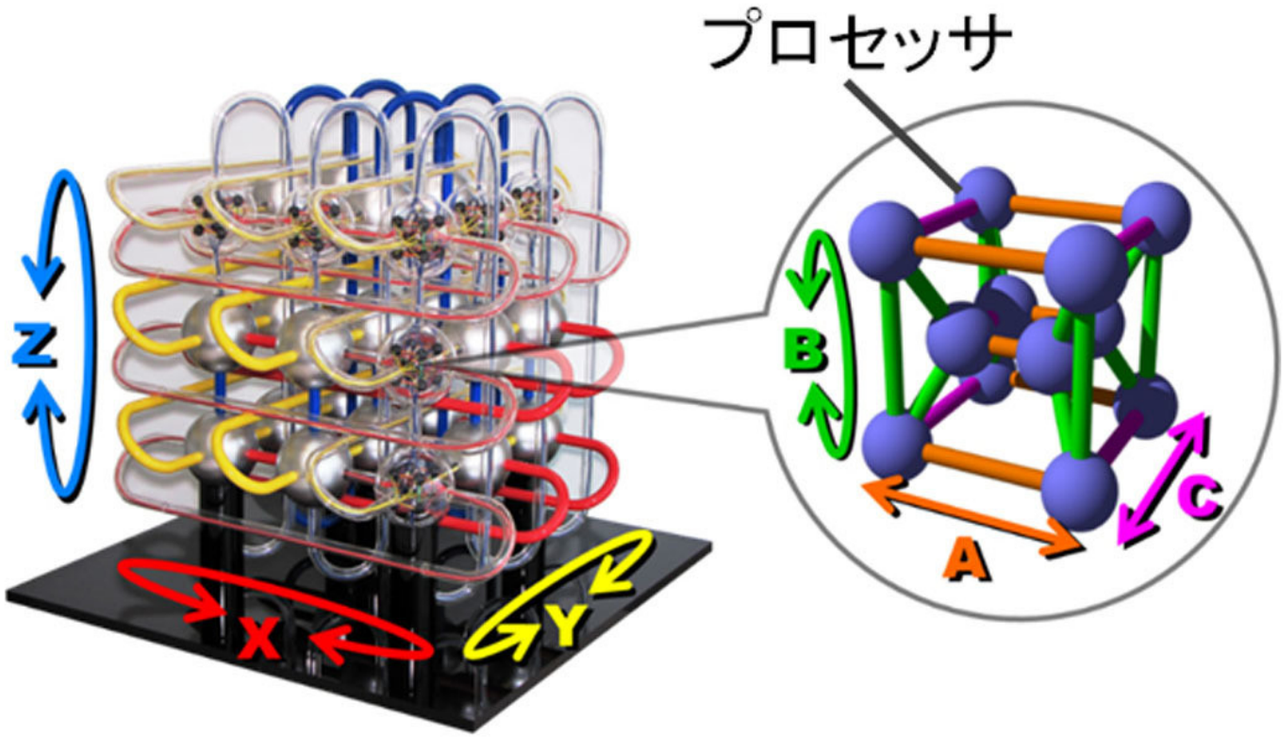


CMG : Core Memory Group
NOC : Network on Chip
TNI: Tofu Network Interface

- 48個の計算コア
- 2/4個のアシスタントコア
  - OSや通信などの割り込みを処理する
- **クロック数は2.0 or 2.2 GHzを選択可**
- 12+1個のコアと8GiBのHBM2で構成されるCMGが4つで構成されている
  - **ノードあたりのプロセス数は4の約数 or 倍数が良い**
- Tofu Interconnect D（Tofu-D）
  - 6次元メッシュトーラス
  - XYZabc軸がある
  - 10本のケーブル、6個の同時通信可能

「富岳」のネットワーク

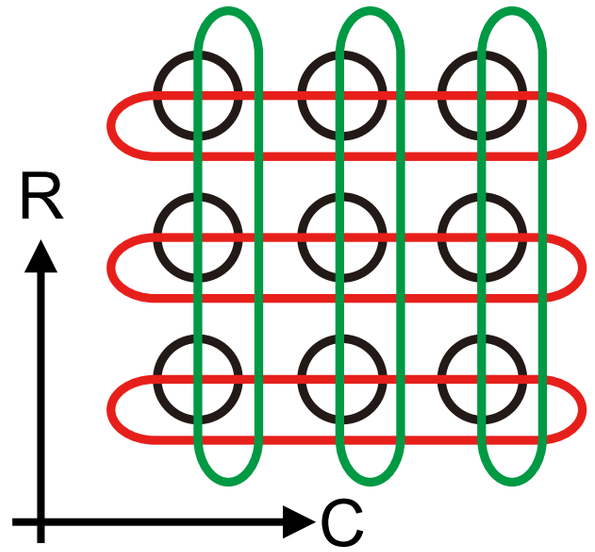
- 6次元メッシュトーラス：XYZabc
  - XYZ：大きさはシステム依存
  - abc：大きさは固定。(a, b, c) = (2, 3, 2)
  - 「富岳」は(X, Y, Z) = (24, 23, 24)なので、24*23*24*2*3*2 = 158,976ノード



https://pr.fujitsu.com/jp/news/2020/04/28.html

- プロセスのマッピング
  - 離散割り当て
  - 1次元トーラス or メッシュ
  - **2次元トーラス** or メッシュ
  - 3次元トーラス or メッシュ

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,C} \\ \vdots & \ddots & \vdots \\ A_{R,1} & \cdots & A_{R,C} \end{pmatrix}$$

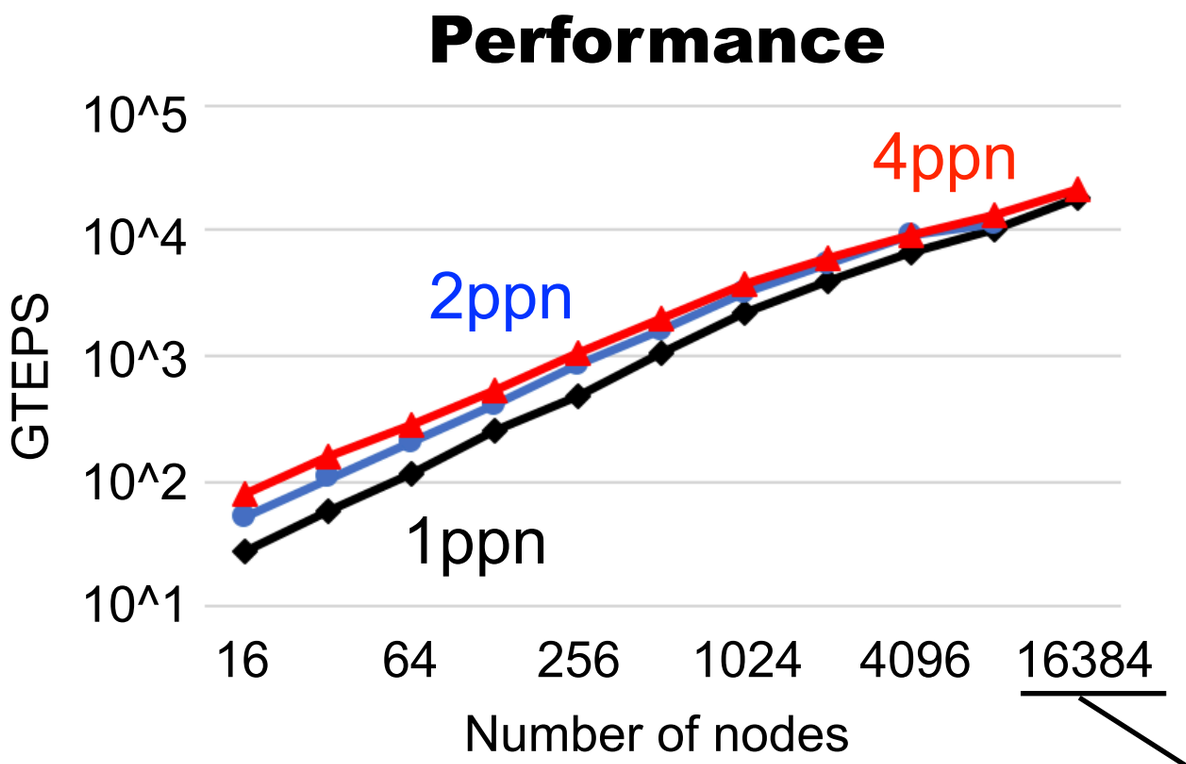


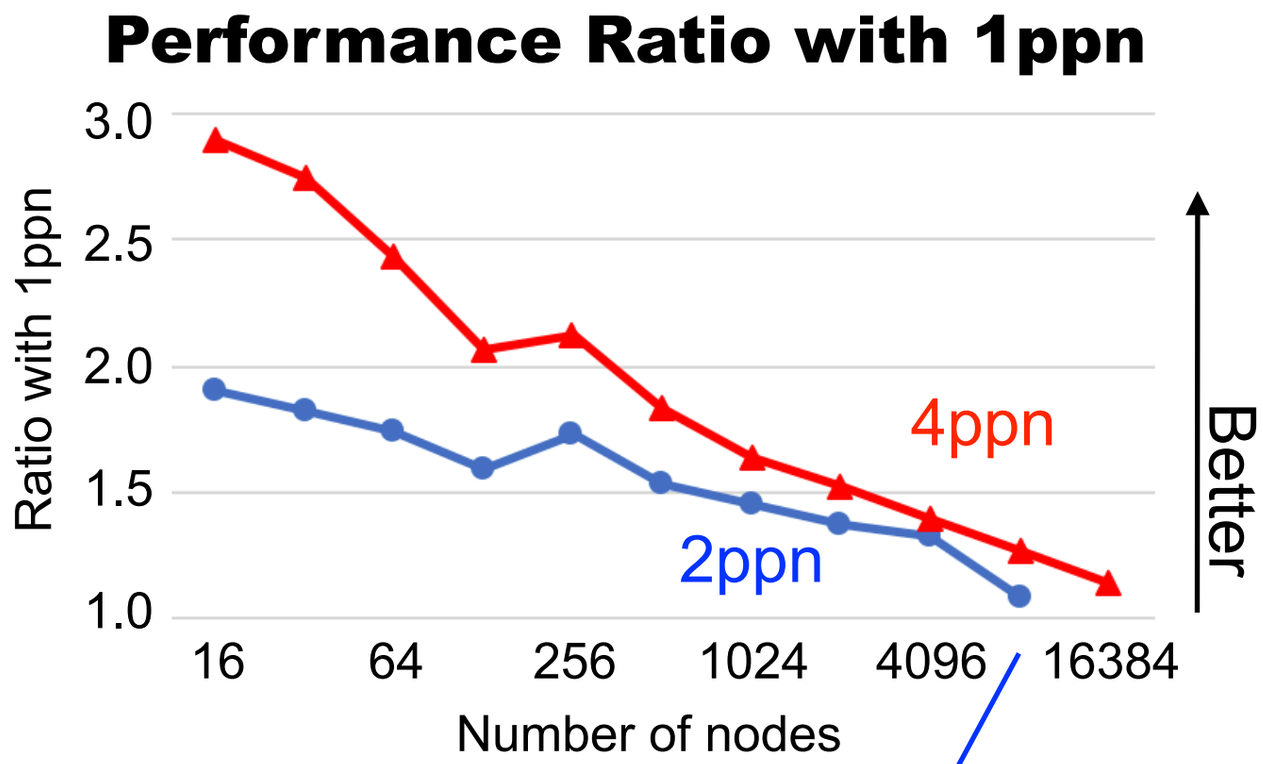
実験の手順

- 最大16,384ノードを使ってパラメータチューニングする（2020年3月〜5月）
- 92,160ノードを使ってGraph500に投稿するデータを取得する（2020年5月）

ノードあたりの最適なプロセス数（1/2）

- Process per node (ppn)
  - 1プロセス48スレッド（1ppn）
  - 2プロセス24スレッド（2ppn）
  - 4プロセス12スレッド（4ppn）
- ノードあたりSCALE=24（約1678万頂点）
- Weak scaling



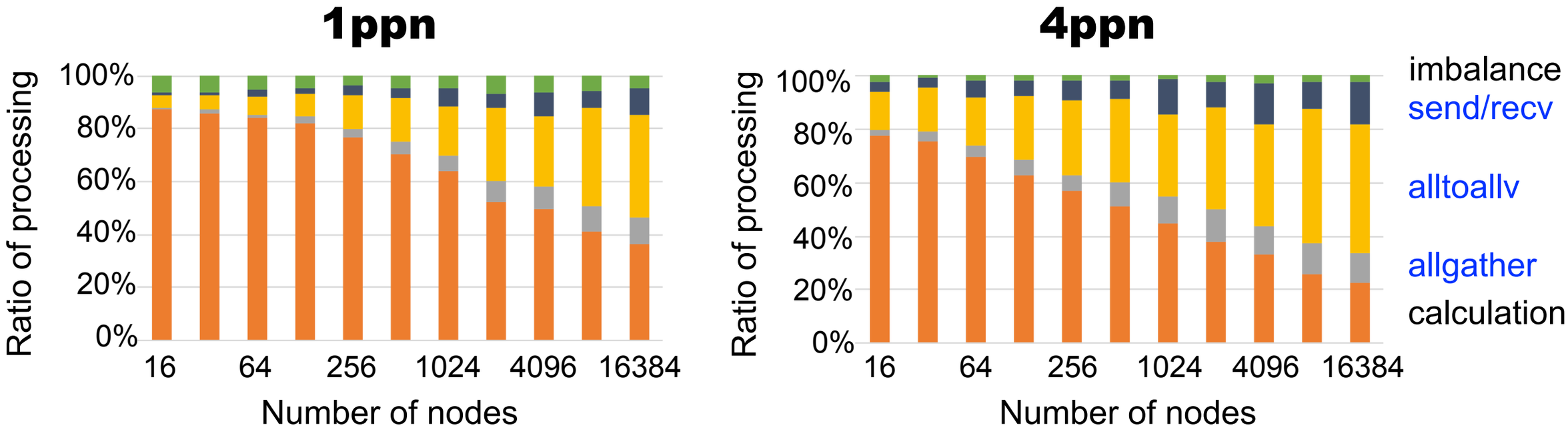


1ppn : R x C = 128 x 128
2ppn : R x C = 256 x 128
4ppn : R x C = 256 x 256

2ppnの16384ノードの結果は
システム不具合で取得できず

- ノード数が多いほど、それぞれの性能差は小さくなることがわかる

ノードあたりの最適なプロセス数（2/2）

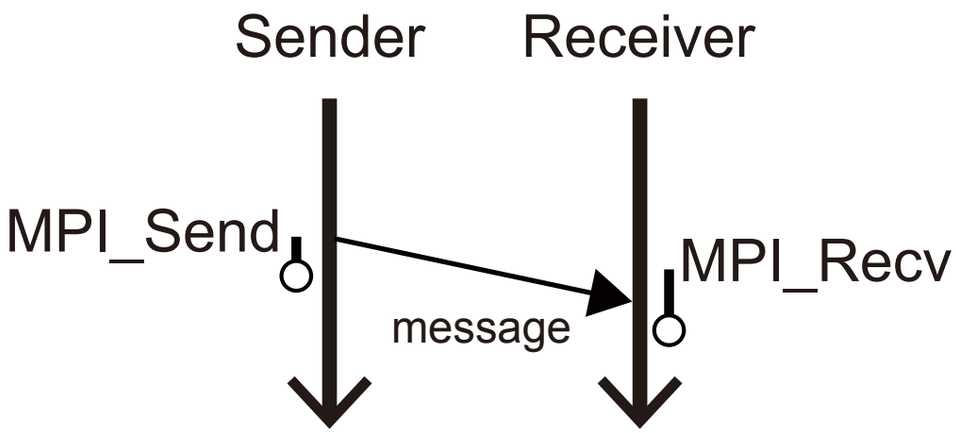


- 1ppnの方が4ppnよりも**通信の割合**は小さい
- 今回の実験では、最大16,384ノードまでだったが、ノード数をこれ以上に増やすと、より通信の割合が大きくなると考えられる
- **通信性能をフルに引き出せる1ppnで計測することにする**
- 1ppnの16,384ノードの結果は**18,450 GTEPS**であった

Eager通信方式の利用（1/3）

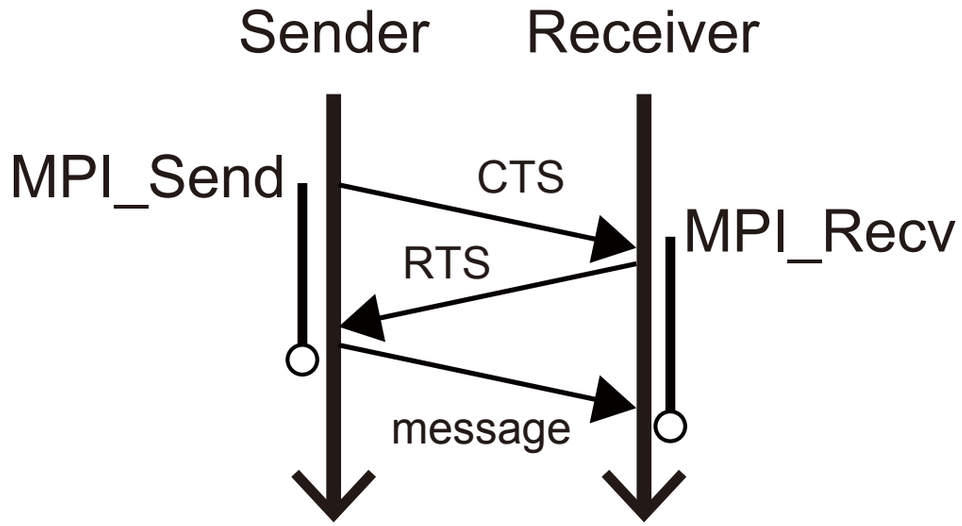
- 主要なMPIライブラリは、メッセージの送受信においてEagerとRendezvousという2つの通信プロトコルをサポートしている
  - 転送サイズが小さい場合はEager、大きい場合はRendezvousが自動的に選択される

Eager



送信/受信プロセスの状態に関わらず、メッセージの送信処理を開始/終了することができる非同期通信

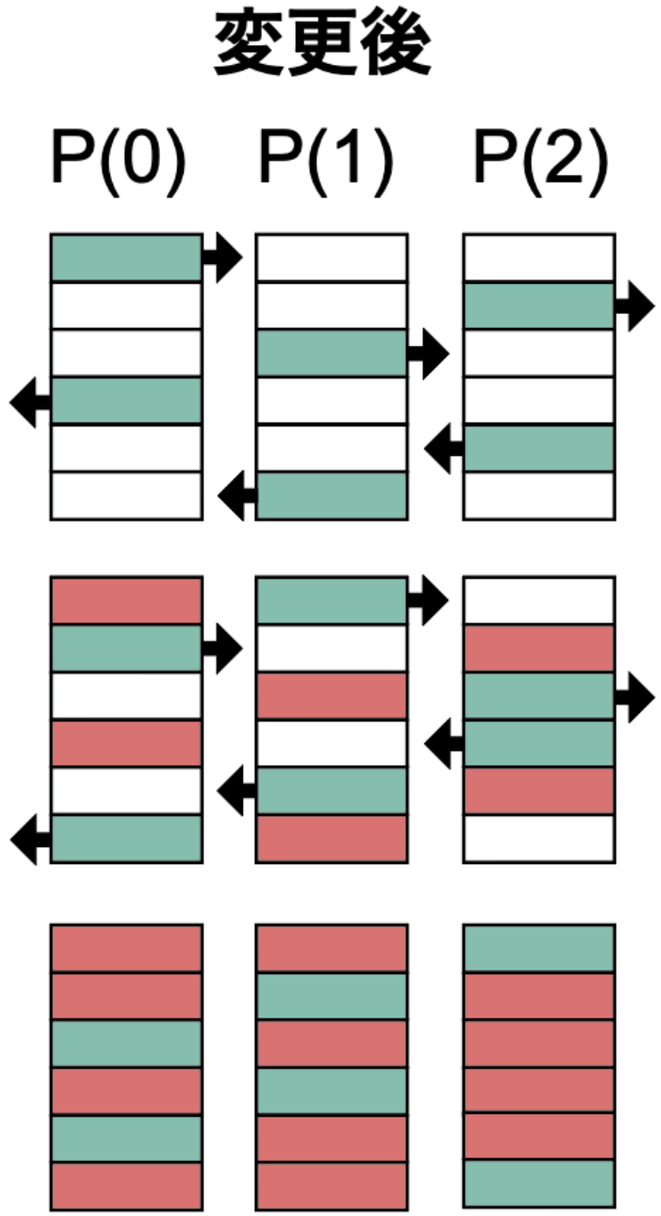
Rendezvous

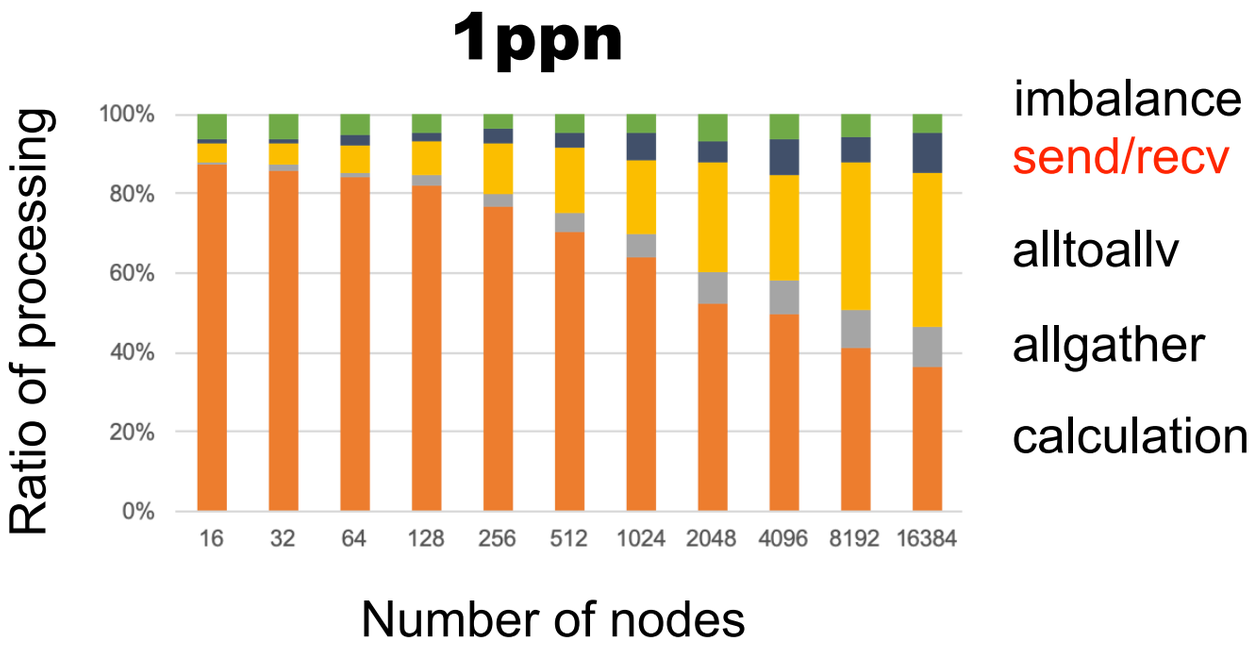


通信を行う双方のMPIプロセスの準備が完了してからメッセージの送受信を行う同期通信

Eager通信方式の利用（2/3）

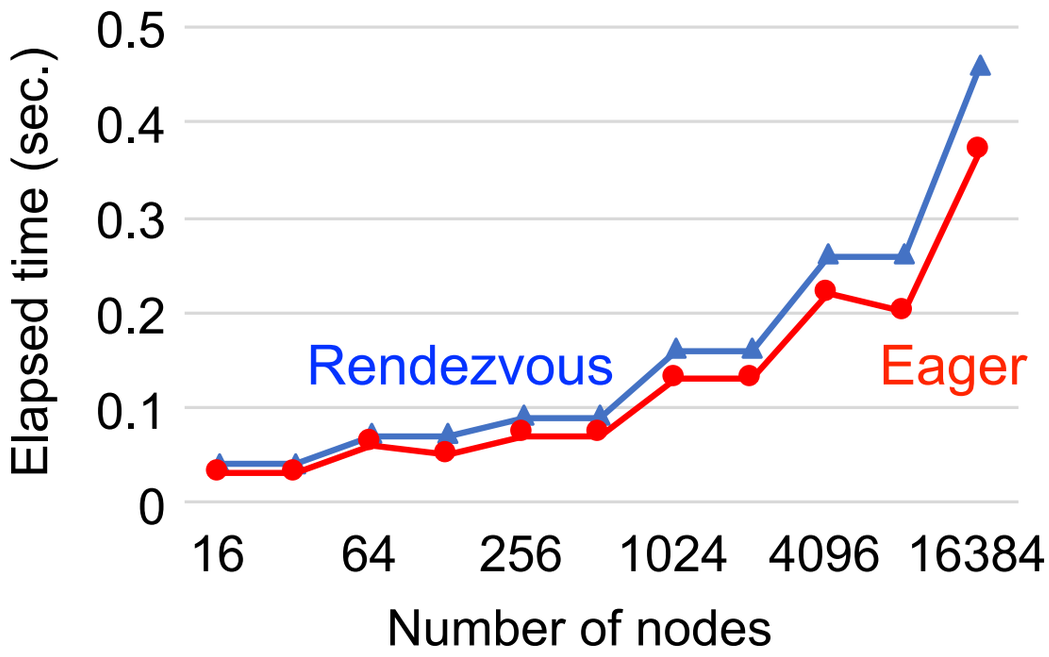
- 前実験で使われたsend/recvの通信方式は、すべてRendezvousだった
- 「富岳」の富士通MPIライブラリでは、
  メモリに余裕があり、非同期通信を
  促進したい場合、mpiexecにパラメータを
  渡すことで、Eager通信方式の利用率を
  高めることができる



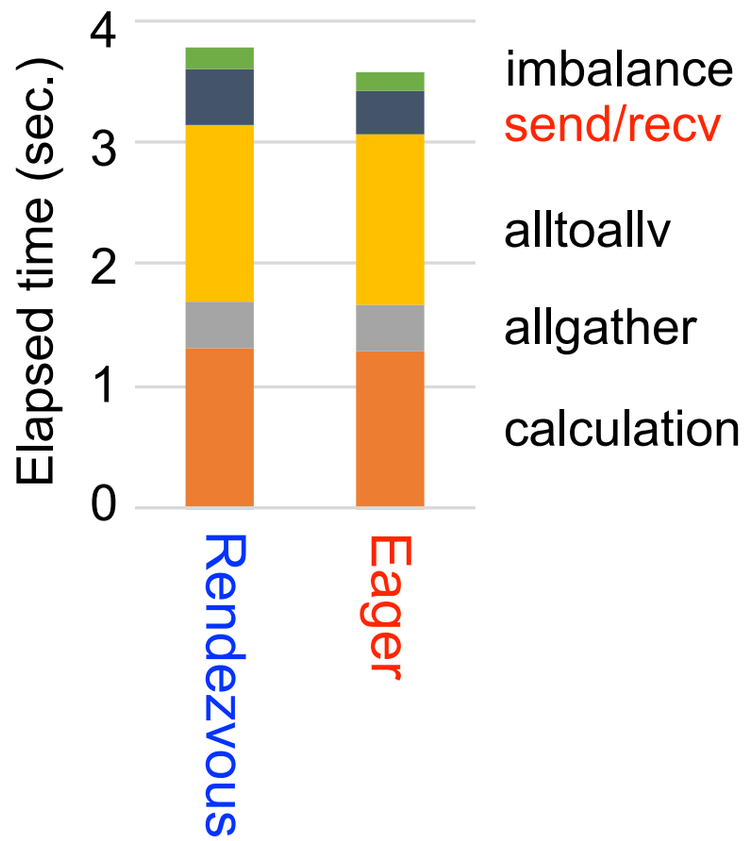


Eager通信方式の利用（3/3）

- send/recvの通信時間



- 16,384ノード時の各時間



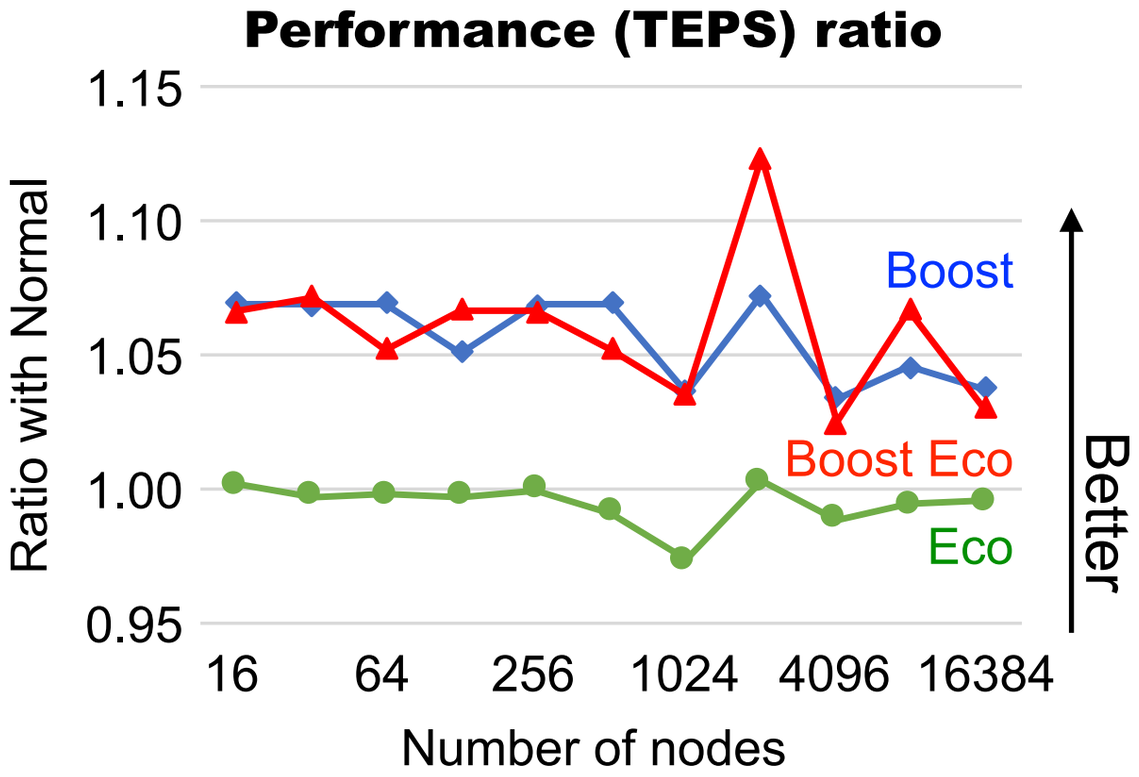
- **Eager通信方式**の16,384ノードの結果は**19,496 GTEPS**であり、**Rendezvous通信方式**の結果（18,450 GTEPS）よりも 5.7%性能向上した
- 以降の実験では、すべて**Eager通信方式**で実行する

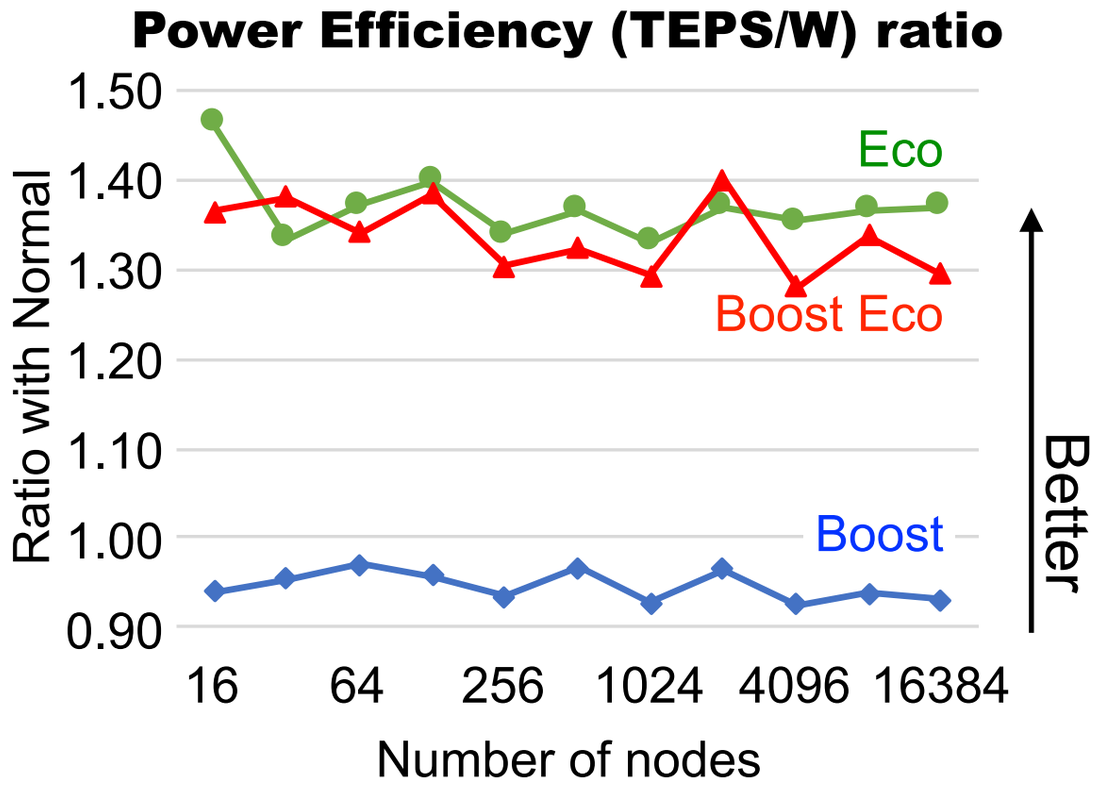
ブーストモードとエコモード(1/2)

- ユーザがジョブ毎にCPUの周波数を変更できる
  - Normal mode：**2.0** GHz
  - Boost mode：**2.2** GHz
- Eco mode：**2**本の浮動小数点演算パイプラインが**1**本に制限され、その際の最大電力に合せた電力制御が行われる
  - BFSでは、浮動小数点演算は基本的にないため、性能に影響を与えずに電力削減を行えることが期待できる
  - なお、電力測定はユーザが行う方法と施設側で行う方法がある
- 4つの組合せで性能と電力について調べる
  - Normal：　**2.0** GHz、浮動小数点演算パイプライン **2**本（これまでの設定）
  - Boost：　**2.2** GHz、浮動小数点演算パイプライン **2**本
  - Normal Eco：**2.0** GHz、浮動小数点演算パイプライン **1**本
  - Boost Eco：**2.2** GHz、浮動小数点演算パイプライン **1**本

ブーストモードとエコモード(2/2)

- Normal： **2.0** GHz、浮動小数点演算パイプライン **2**本（これまでの設定）
- Boost： **2.2** GHz、浮動小数点演算パイプライン **2**本
- Normal Eco： **2.0** GHz、浮動小数点演算パイプライン **1**本
- Boost Eco： **2.2** GHz、浮動小数点演算パイプライン **1**本





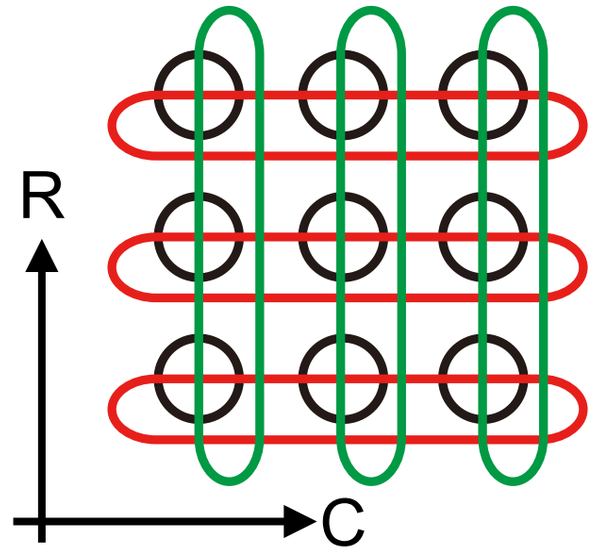
- Normalの結果が1.00。Boostにすると性能が高く、Ecoにすると電力効率が良い
- **Boost Eco**が性能と電力効率のバランスが良い。16,384ノードの結果は **20,098 GTEPS**であり、前の結果（19,496 GTEPS）より 3.1%性能向上した

6次元マッピング（1/3）

- 「富岳」のネットワークのトポロジー
  - 全系の6次元の軸の大きさは、(X, Y, Z, a, b, c) = (23, 24, 23, 2, 3, 2)
  - ジョブスケジューラの設定でプロセスマッピングを行った場合の各軸の最大値は YZc × Xab = 1,104 x 114 = R x C
  - しかしながら、RとCの値は近い方が望ましい

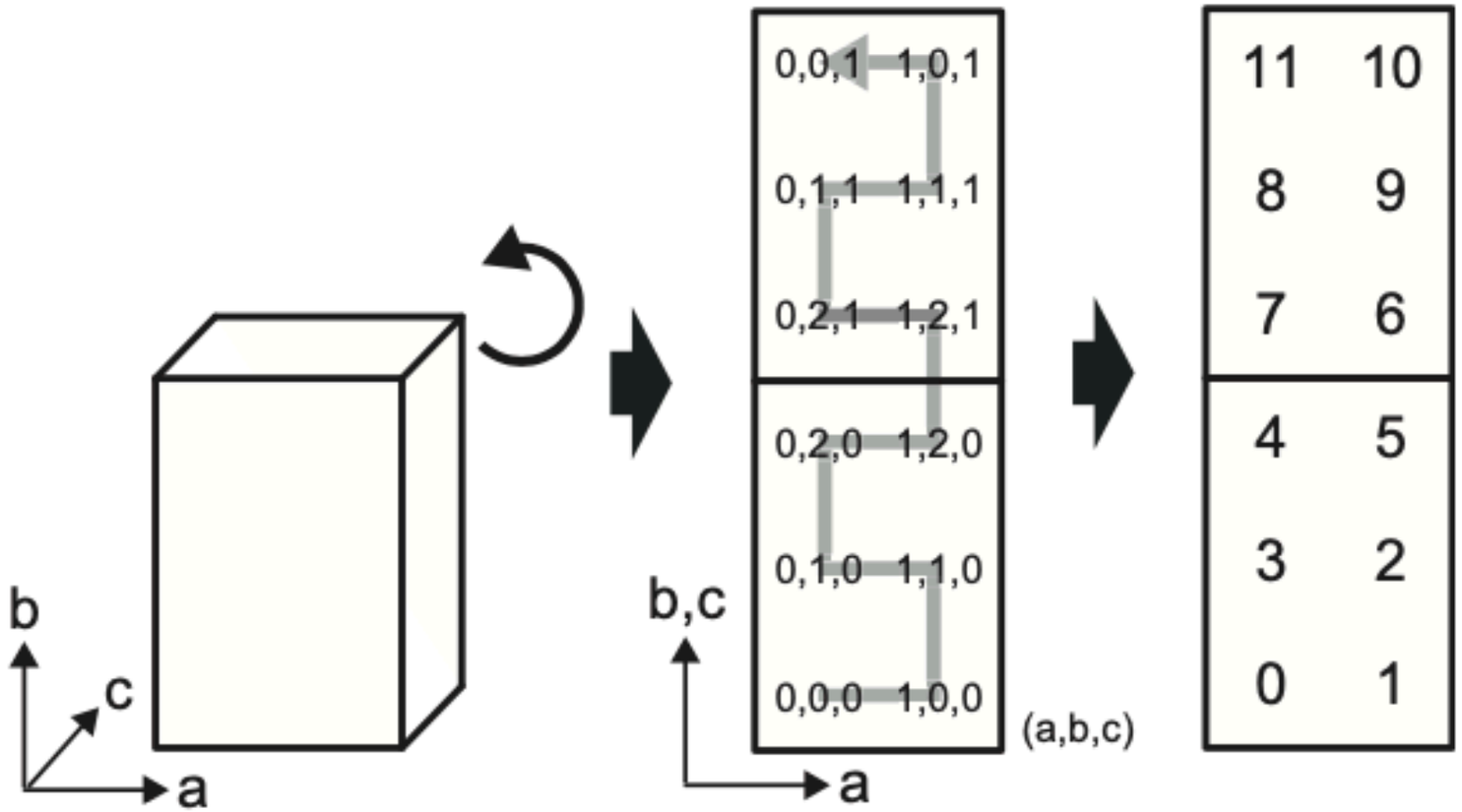
- 6次元の任意の軸にマッピング
  - RとCの値が最も近い組み合わせは、XY × Zabc = 552 x 288 = R x C
  - 今回は共用前評価環境なので、92,160ノードを用いた評価を行う
  - (X, Y, Z, a, b, c) = (20, 16, 24, 2, 3, 2)
  - XY × Zabc = 320 x 288 = R x Cが最適

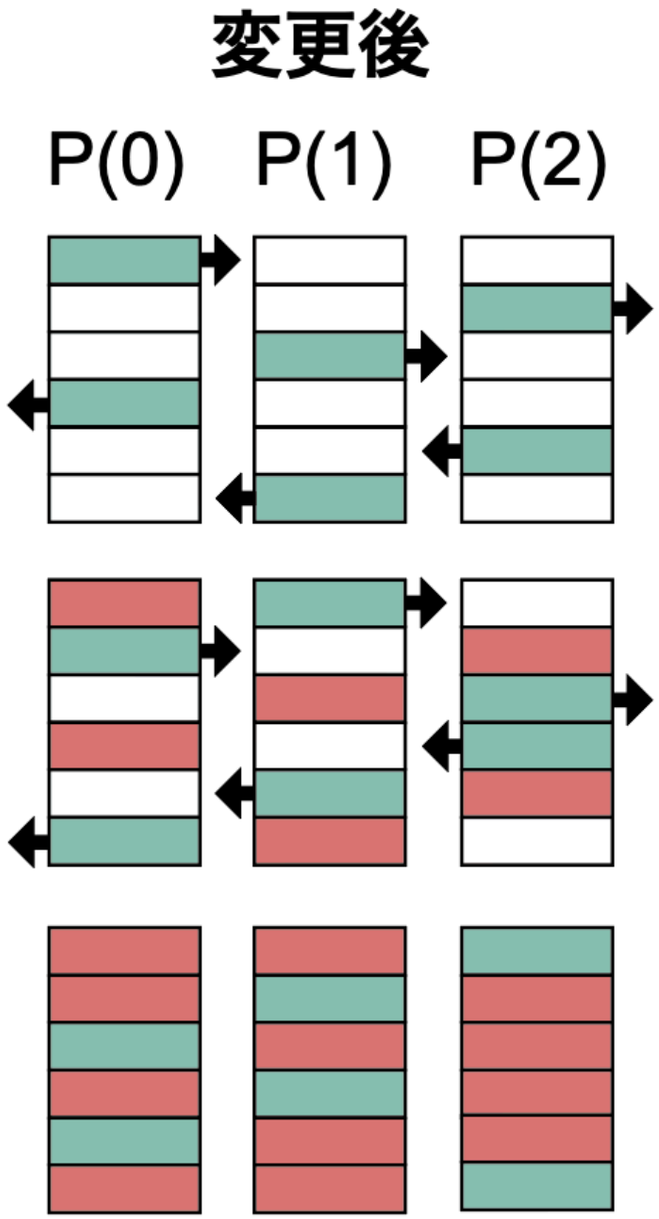
$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,C} \\ \vdots & \ddots & \vdots \\ A_{R,1} & \cdots & A_{R,C} \end{pmatrix}$$



6次元マッピング（2/3）

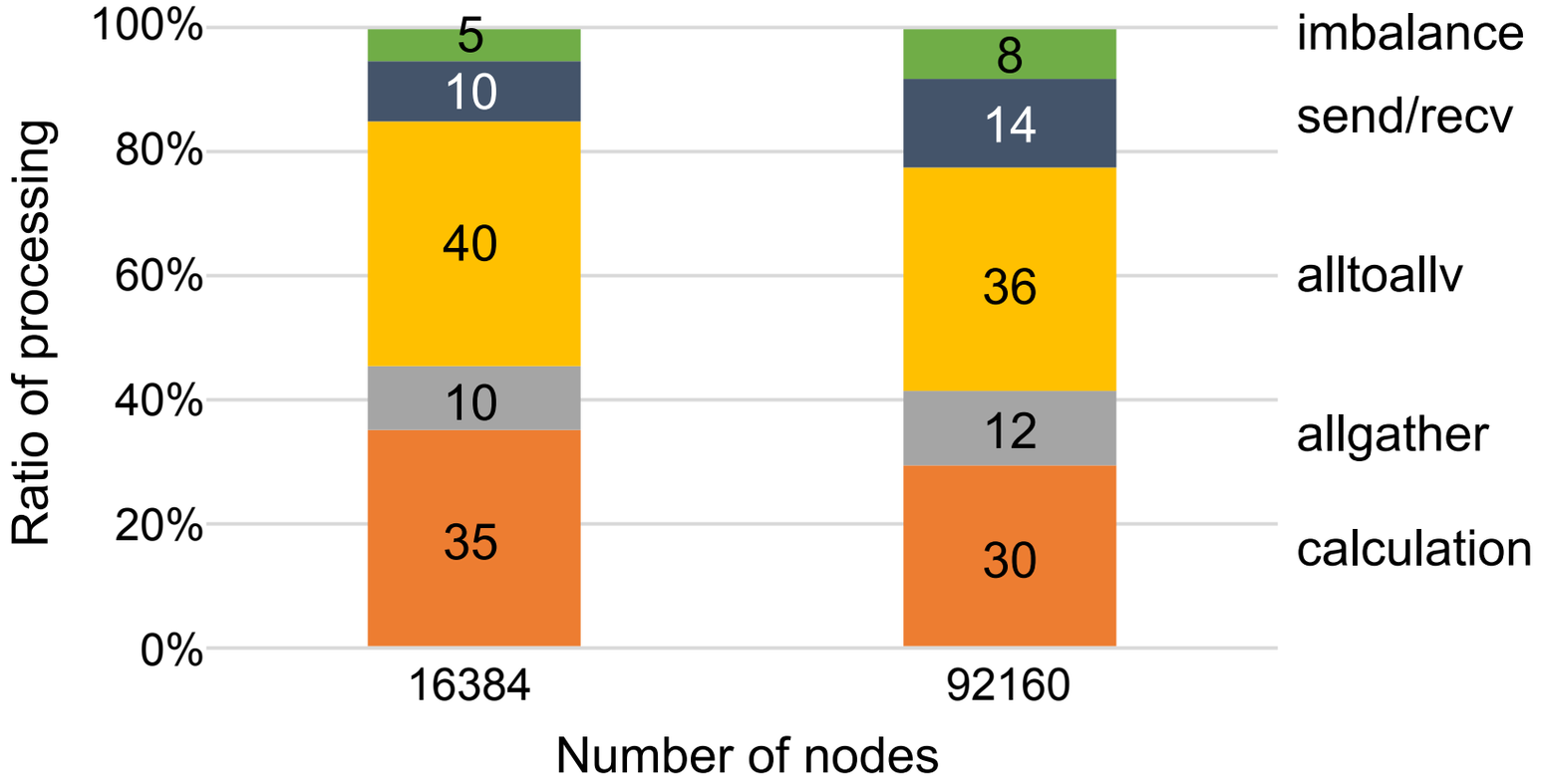
- Cはすべてのノードが隣接すると性能が高い
- Cにabc軸（2 x 3 x 2）を割り当てる例
  - 最初の次元を横軸、残りの次元を縦軸に分解
  - 下記のように、最初と最後のランク（0と11）がネットワーク的に隣り合わせになるようにする





6次元マッピング（3/3）

- SCALE=40、92160ノード（XY × Zabc = 320 x 288 = R x C）で計測した結果、性能は 70,980 GTEPS、消費電力は 8,300 kW、電力効率は 8.55 MTEPS/W
  （この電力測定は施設側で行った）
  - 性能は「京」（82,944ノード）の2.27倍、電力効率はMiraの1.93倍であった



- 92,160ノードの方が通信の割合が大きい

もくじ

まとめと今後の課題

- まとめ
  - 「富岳」の一部（92,160ノード）を用いた2D Hybrid-BFSの性能チューニングおよび評価を行った
  - SCALE=40（約1.1兆個の頂点と約17.6兆本のエッジから構成される大規模グラフ）を用いて性能評価を行った結果、70,980 GTEPSを達成し、2020年6月のGraph500で1位を獲得した
  - この結果は、不規則な計算が大半を占めるBFSにおいても、「富岳」が高い能力を持っていることを実証した
- 今後の課題
  - 「富岳」の全系（158,976ノード）を用いた評価を行う
  - 様々なグラフ処理を行うコード（SSSPも含む）を開発し、実データのグラフ処理を行うために「富岳」を活用していく