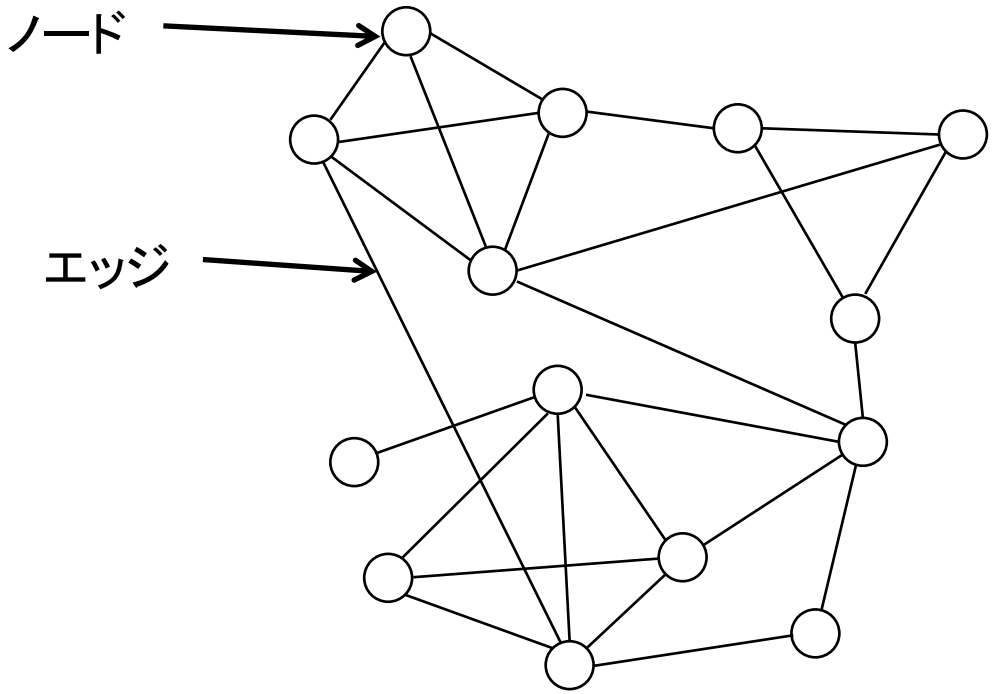
大規模グラフデータ分析入門

塩川 浩昭
筑波大学 計算科学研究センター
計算情報学研究部門 助教

第8回「学際計算科学による新たな知の発見・統合・創出」シンポジウム
2016年10月18日

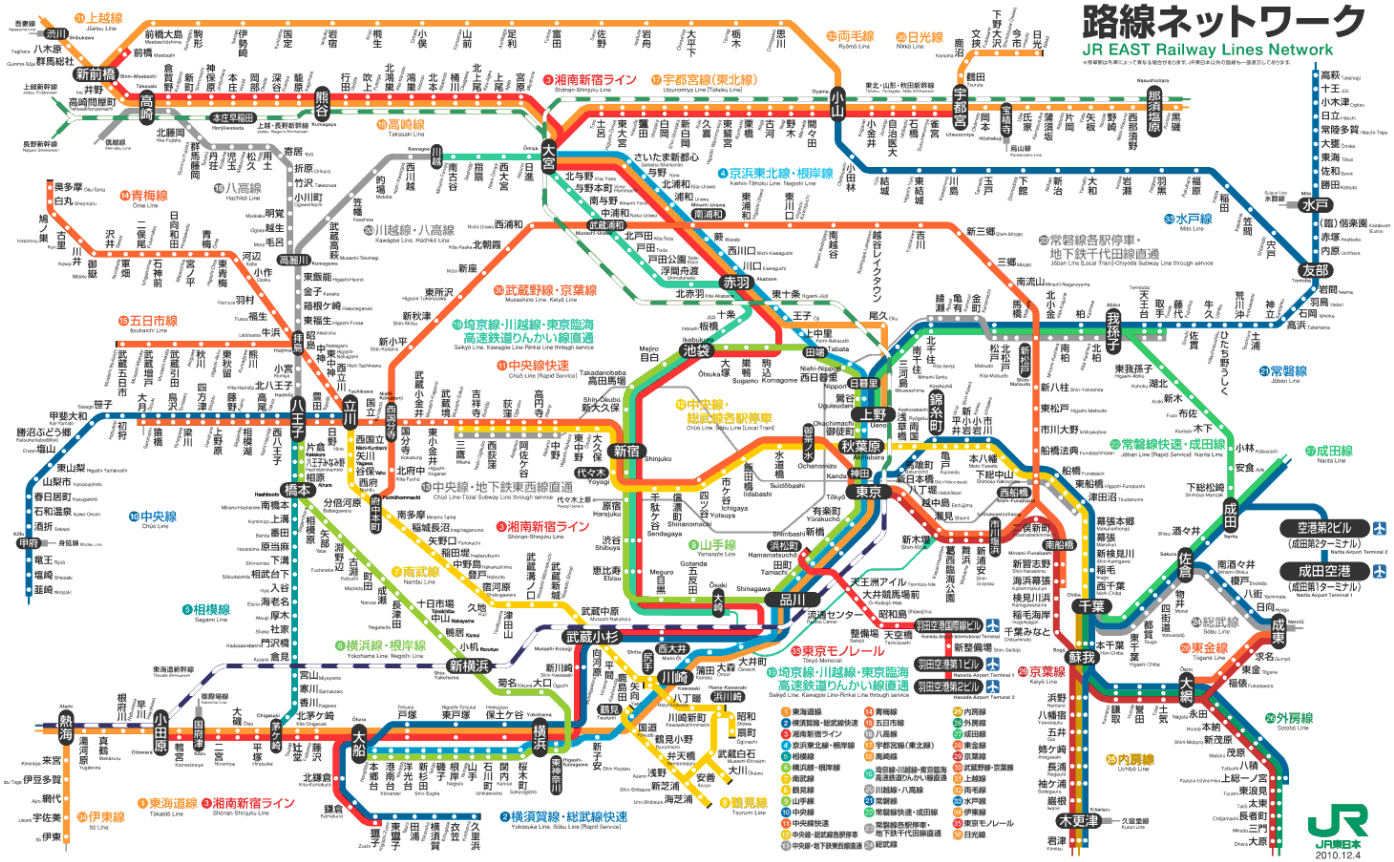
Graph is everywhere

- データエンティティと，データエンティティ間の関係性を表現した基本的なデータ構造のひとつ



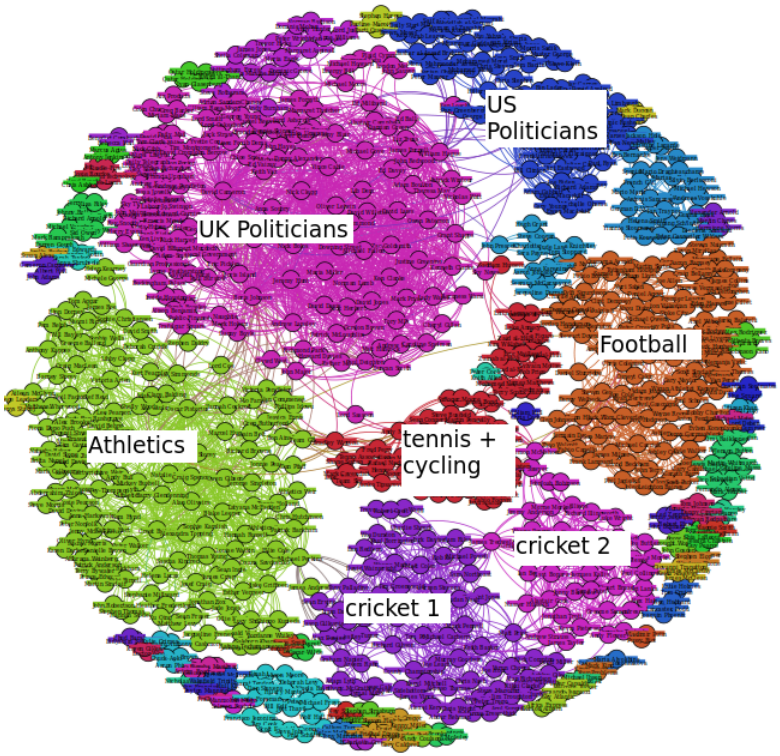
身近にあるグラフデータ

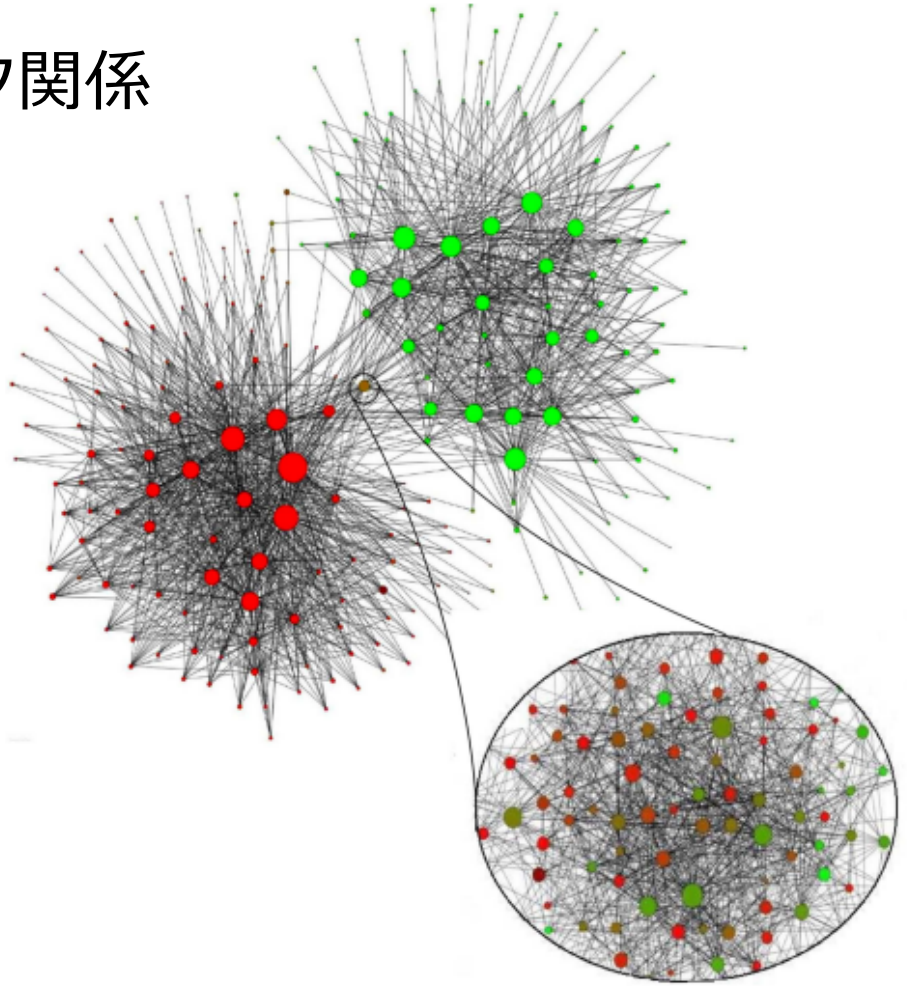
- **交通ネットワーク**
  - ノード：駅・交差点など
  - エッジ：路線・道路など



身近にあるグラフデータ

- **ソーシャルネットワーク・Webグラフ**
  - ノード：人・Webページ
  - エッジ：人間関係・リンク関係

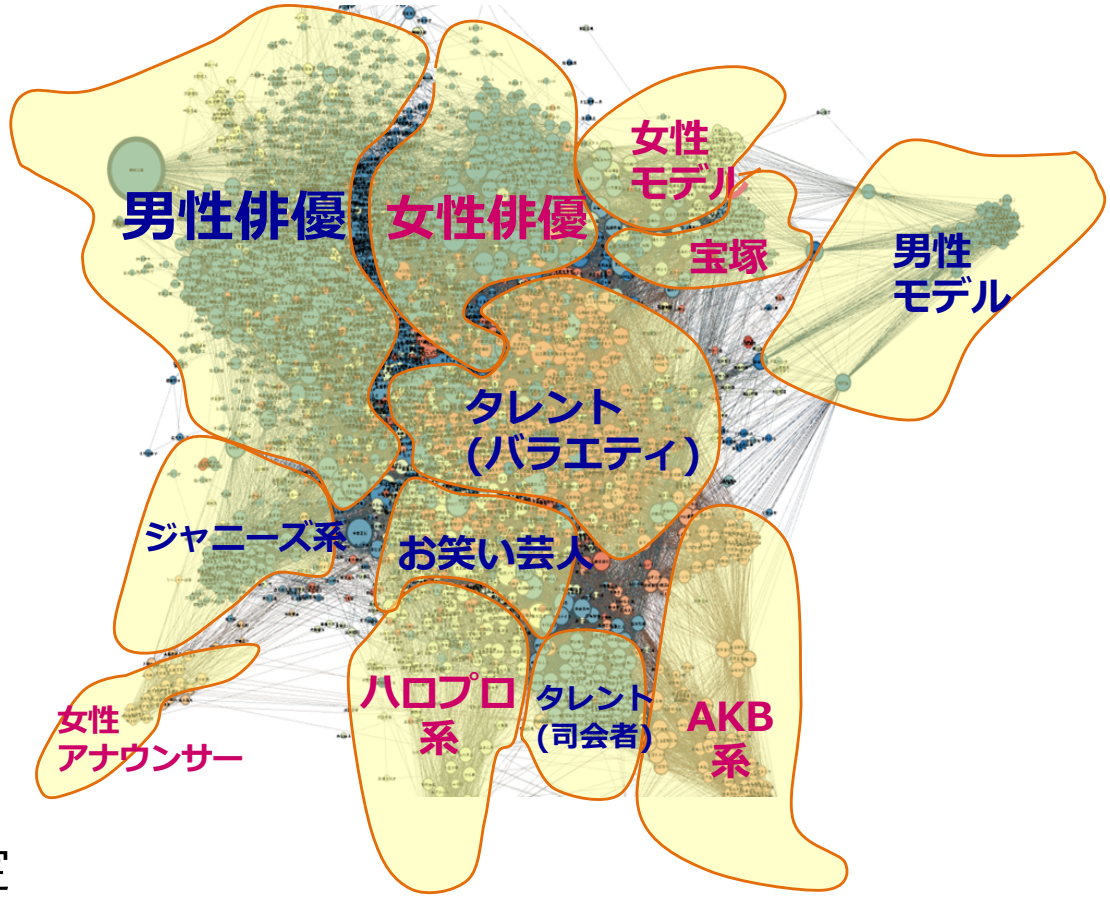




グラフデータ分析の重要性

- **グラフデータを分析することによりデータ間の関係性や類似グループなどを発見**
  - 代表的な分析技術
    - クラスタリング
    - PageRank
    - 最短経路探索 など

- Web/SNS
  - コミュニティ抽出
- 道路ネットワーク
  - 混雑予測・交通案内
- タンパク質相互作用NW
  - 代謝に関わる酵素の特定
  - 転写因子の特定



例. SNSからのコミュニティ抽出

大規模グラフデータ分析

- **大規模化に伴い分析に要する時間が爆発的に増加**

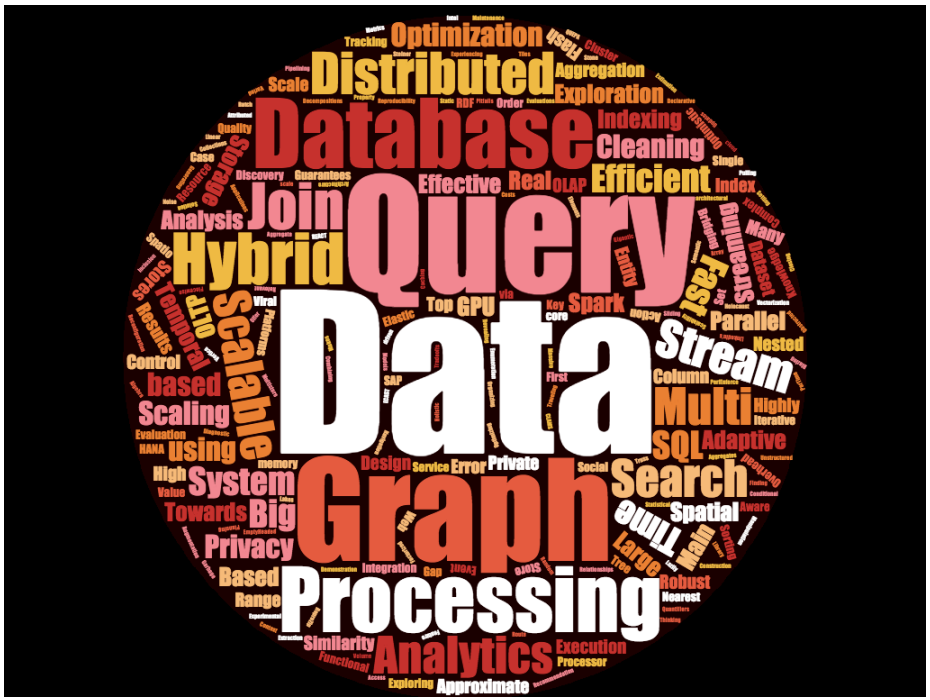
分野	種類	ノード数
OR等	交通ネットワーク	全米：2×10^7以上
Web等	ソーシャルネットワーク・ Webグラフ	Twitter：2×10^8以上 Facebook：5×10^8以上 Google：10^9以上
生物情報	タンパク質間の相互作用・ 脳神経ネットワーク	10^9以上

PFN 秋葉さんの資料より

億ノード規模以上(10^8以上)のグラフデータに対して
どのようにしたら高速かつ高精度な分析ができるか？

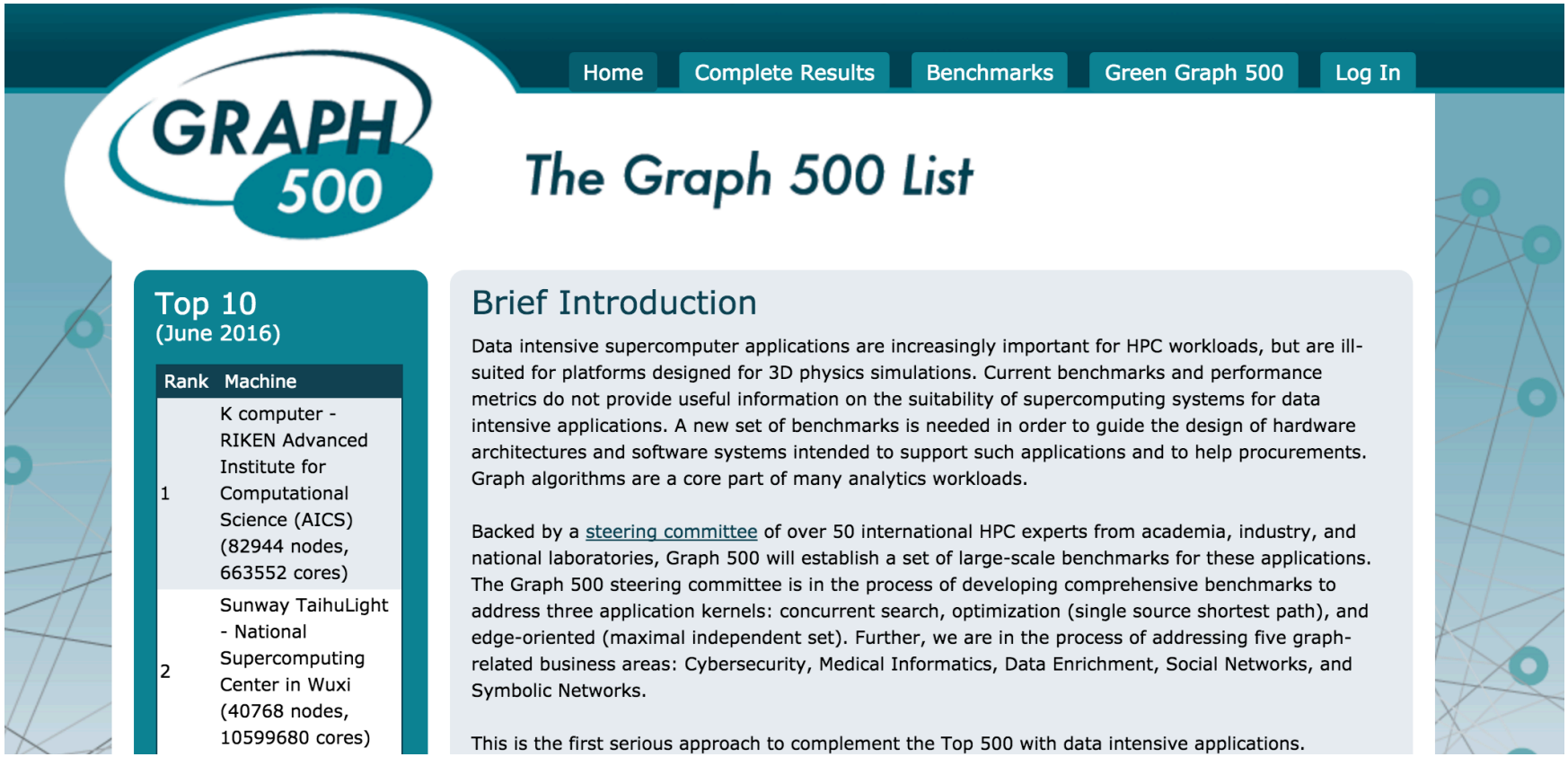
大規模グラフ分析への関心の高まり

- **SIGMOD (データベース分野のトップ会議)**
  - 2016年では ”Graph Data”セッションが4つ
  - “Query Processing”と並ぶ巨大な勢力に
- **SC（HPC分野のトップ会議）**
  - 2010年頃から毎年”Graph Algorithms”セッションが存在



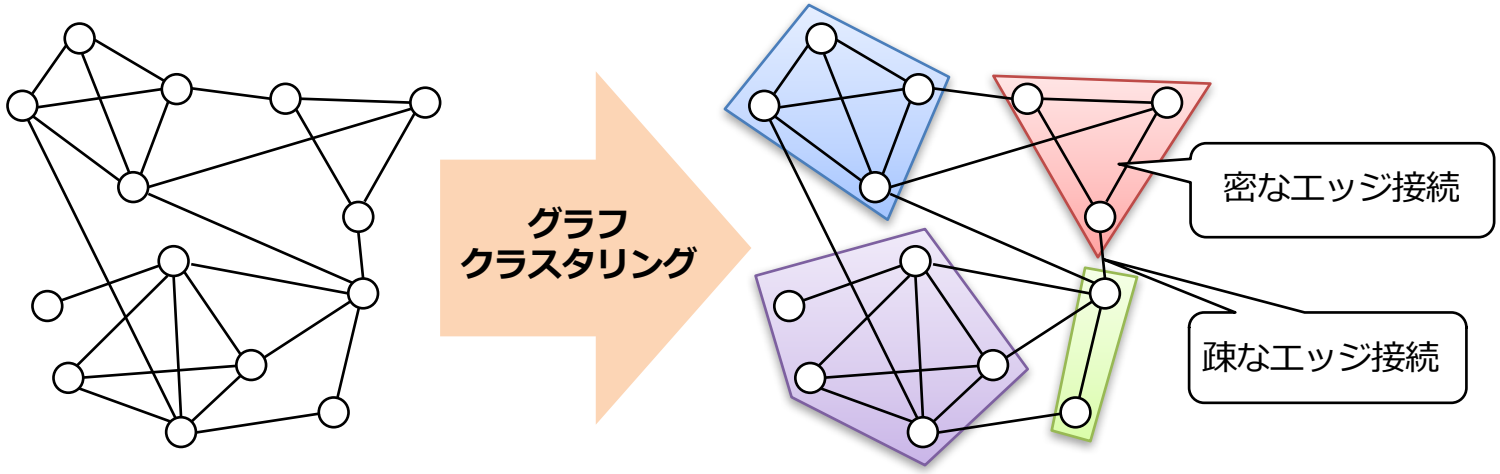
HPC分野における関心

- **Graph500**
  - Top500のあたらしい仲間
  - 高性能計算機をグラフ処理能力でランク付け
  - 2016年6月時点で京コンピュータが1位



本日の本講演の概要

- **グラフデータ分析技術を紹介**
  - グラフクラスタリング分析を題材に最新の研究動向を解説するとともに，高性能計算機を用いた今後の展望を述べる
- **グラフからコミュニティ構造（クラスタ）を抽出するグラフ分析技術のひとつ**
  - 互いに密に接続したノード集合をクラスタとして抽出
    - SNSからのコミュニティ発見 [*Zhou et al. 2010*]
    - 脳神経ネットワークからの発火部位特定 [*Yu et al. 2010*]



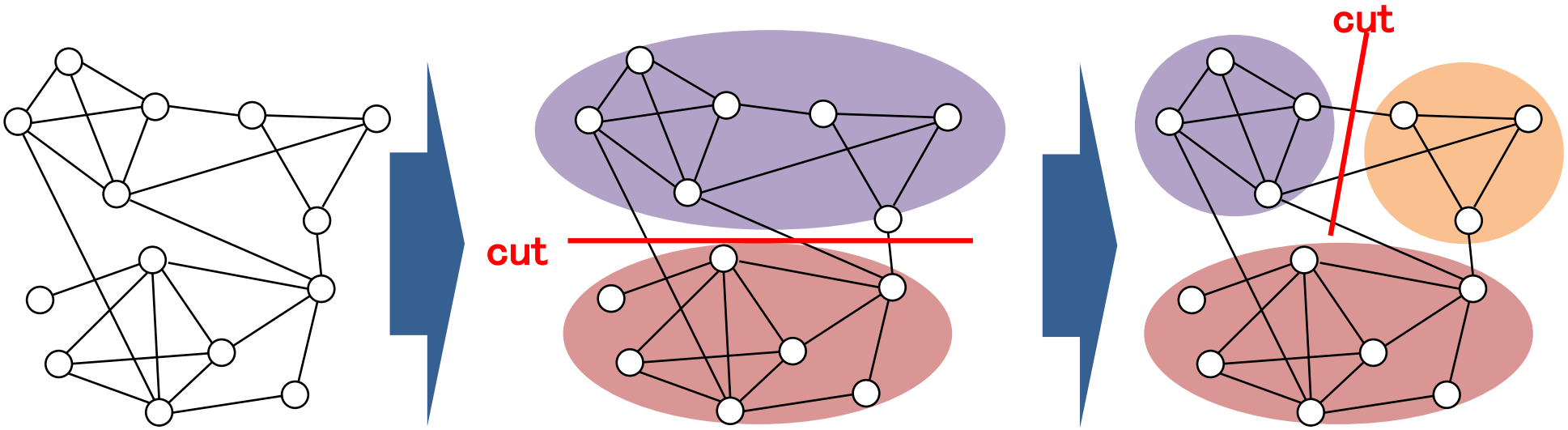
クラスタリングの2つの課題

1. どんなクラスタを定義するか？

2. どうやって高速計算するか？

Min-cut / Normalized Cut法 [Shi and Malik, TPAMI'01]

- **クラスタ間のエッジが最小・クラスタ内のエッジが最大になるようにグラフを2分割にする手法**
  - クラスタのサイズに偏りが発生してしまうため，クラスタの大きさが均等になるように調整する
  - 時間計算量は$O(N^3)$と大きい
  - クラスタの精度がイマイチ
    - 事前にクラスタの個数を決めなければならない
    - クラスタサイズは必ずしも均等ではない



Modularityクラスタリング

- **クラスタの質を表す指標 *Modularity* の最大化による クラスタリング手法**
  - M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E* 69, 026113 (2004)

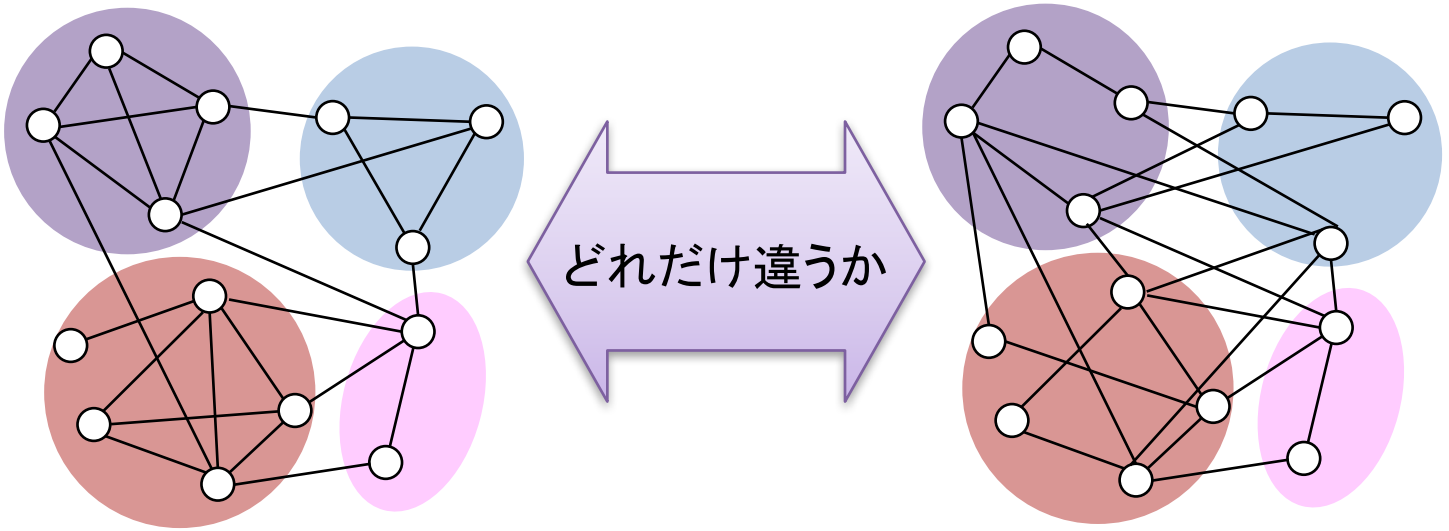
$$Q = \sum_{i \in C} \left\{ \frac{e_{ii}}{2m} - \left(\frac{a_i}{2m} \right)^2 \right\}$$

実際にクラスタ i に含まれる エッジ数の割合

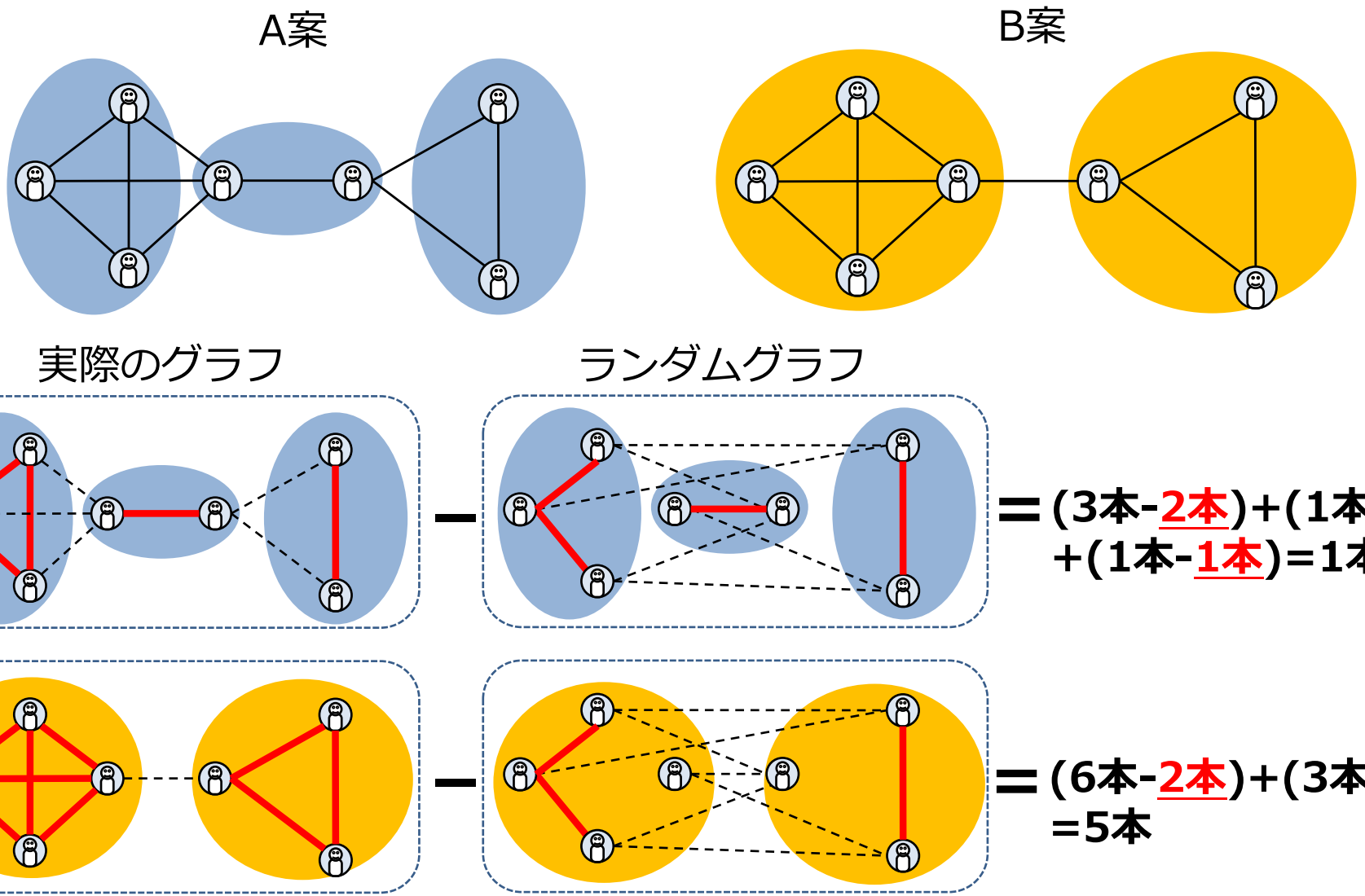
ランダムグラフにおいて クラスタ i に含まれるエッジ数の割合

実際のグラフのクラスタリング結果

ランダムグラフのクラスタリング結果



Modulraityクラスタリングの例



最も直感に従ったB案が最も高い値になっていることがわかる

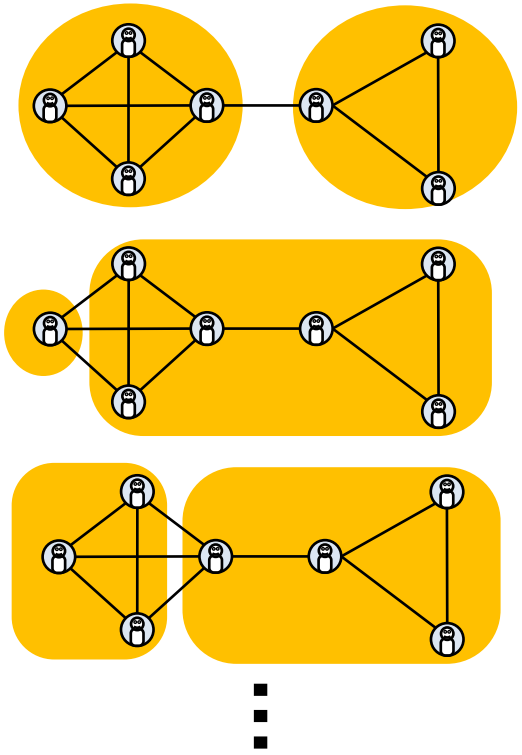
クラスタリングの2つの課題

1. どんなクラスタを定義するか？

2. どうやって高速計算するか？

モジュラリティを用いたコミュニティの抽出

- **最も単純な方法**
  - 全てのコミュニティのパターンを列挙して，モジュラリティが最も高いものをみつける



2つのコミュニティを探す場合

$$\sum_{k=1}^{7}\binom{7}{k} = 127\text{通り}$$

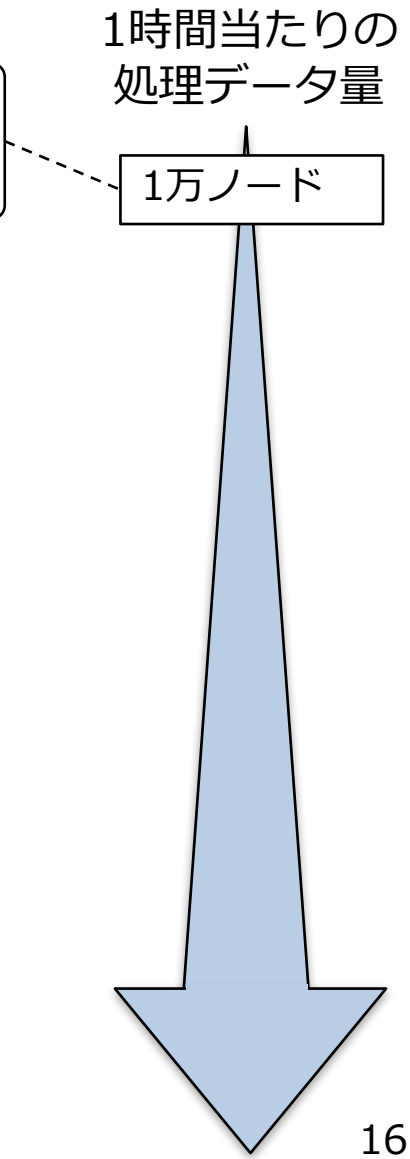
3つのコミュニティを探す場合

$$\sum_{k=1}^{7}\sum_{j=1}^{7-k}\binom{7}{k}\binom{7-k}{j} = 1932\text{通り}$$

- **ノード数やコミュニティ数が増えると組合せ数が爆発的に増加し計算が終わらない**

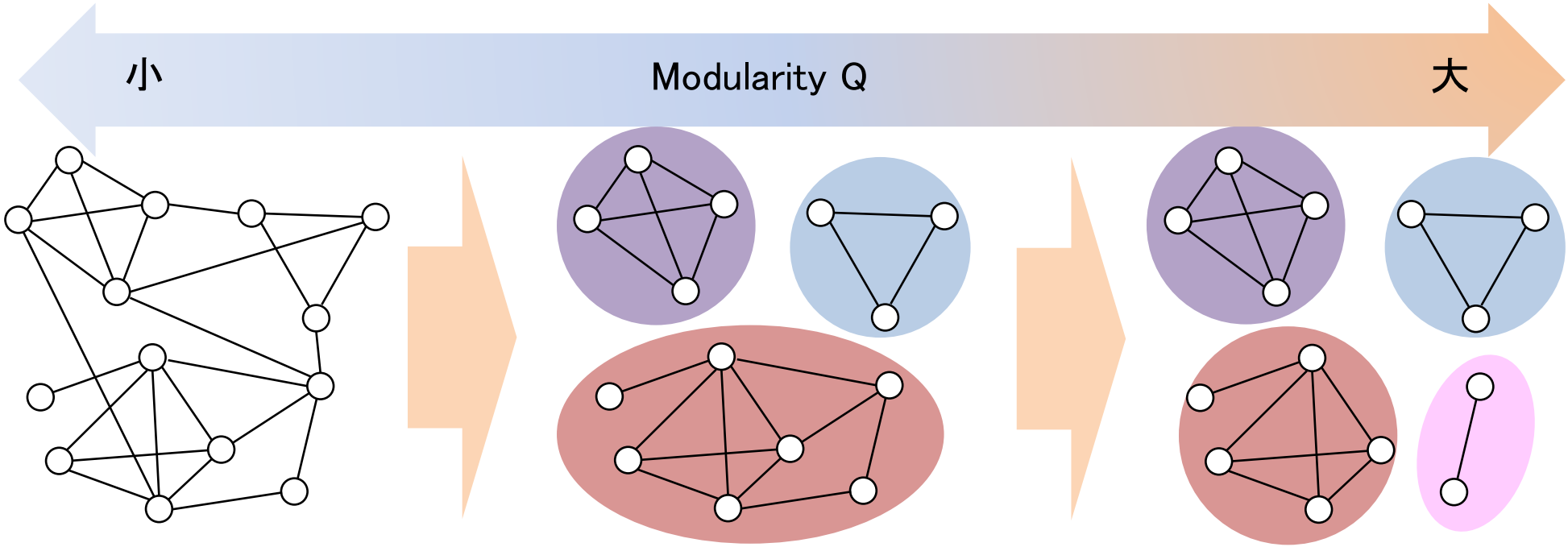
Modularityによるグラフクラスタリング

Girvan-Newman法 [*Girvan et al., 2004*]
トップダウンにクラスタに存在しそうなエッジを削除する手法



Girvan-Newman法 [Girvan et al., 2004]

- **グラフから適当にエッジを削除してその時のModularityを評価**
- **Modularityが最も大きくなったものを結果として出力**
  - Min-Max法やNormalized-cutに近いトップダウン式の手法
  - 計算量は$O(N^3)$と大きい

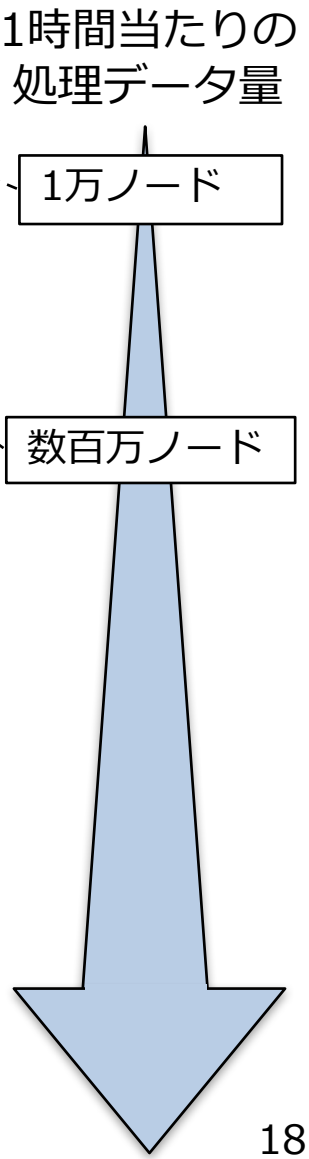


Modularityによるグラフクラスタリング

Girvan-Newman法 [*Girvan et al., 2004*]
トップダウンにクラスタに存在しそうなエッジを削除する手法

Newman法 [*Newman et al., 2004*]
貪欲法によりボトムアップからModularityを向上させる手法

CNM法[*Clauset et al., 2004*]
ヒープの導入やヒューリスティクスによるNewman法の高速化



Newman法 / CNM法

- **２つノードを同じクラスタにした時に生じるModularityの上昇量 Modularity gain ΔQを定義**

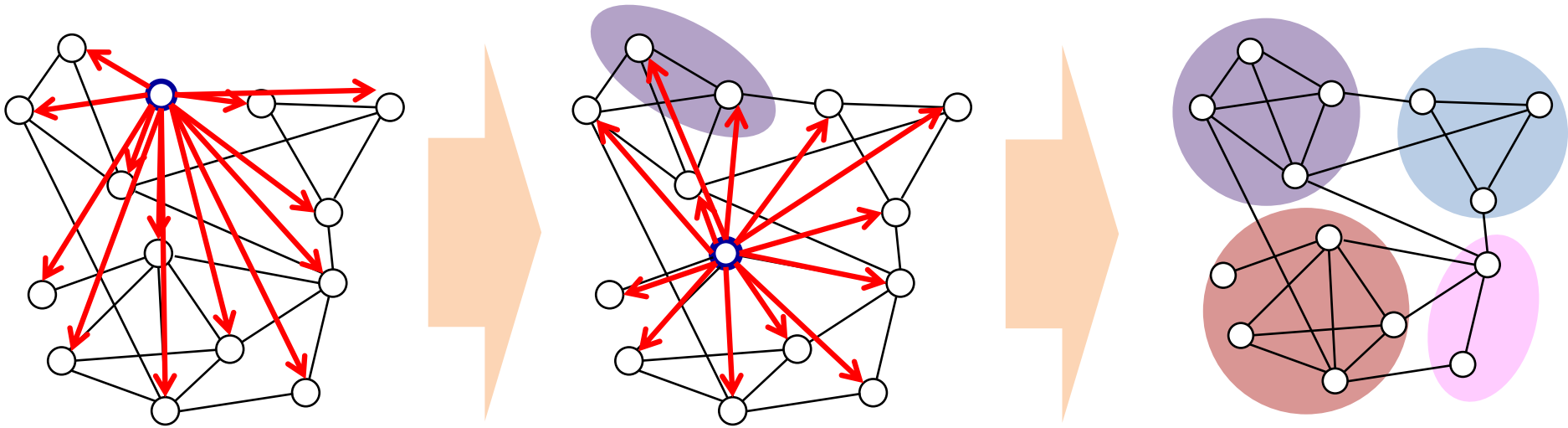
Modularity

$$Q = \sum_{i \in C} \left\{ \frac{e_{ii}}{2m} - \left(\frac{a_i}{2m} \right)^2 \right\}$$

Modularity gain

$$\Delta Q_{ij} = 2\{2me_{ij} - a_i a_j\}$$

- **ΔQを最大にするノードペアを貪欲法で探索しクラスタリング**
- **計算量は$O(N^2) \sim O(M \log N)$**



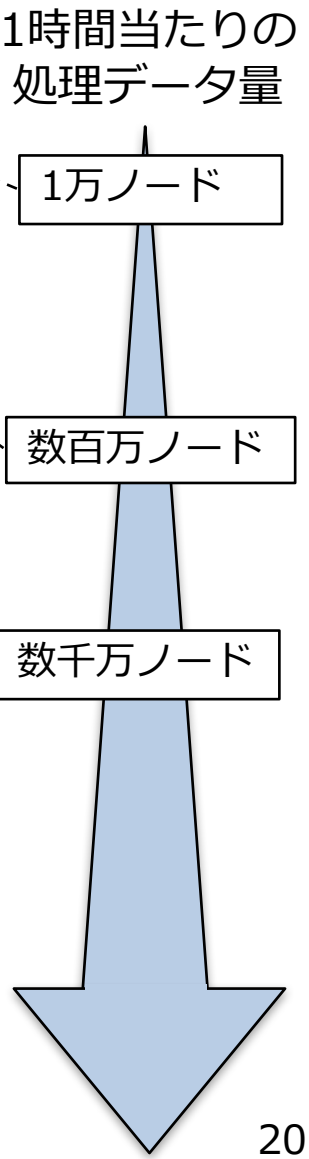
Modularityによるグラフクラスタリング

Girvan-Newman法 [*Girvan et al., 2004*]
トップダウンにクラスタに存在しそうなエッジを削除する手法

Newman法 [*Newman et al., 2004*]
貪欲法によりボトムアップからModularityを向上させる手法

CNM法[*Clauset et al., 2004*]
ヒープの導入やヒューリスティクスによるNewman法の高速化

Louvain法[*Blondel et al., 2008*]
隣接するノード同士にのみを計算する計算量の小さな手法
最速かつ最高精度の手法として広く利用されている



Louvain法

- **以下のパスをModularityが向上する限り繰り返す**

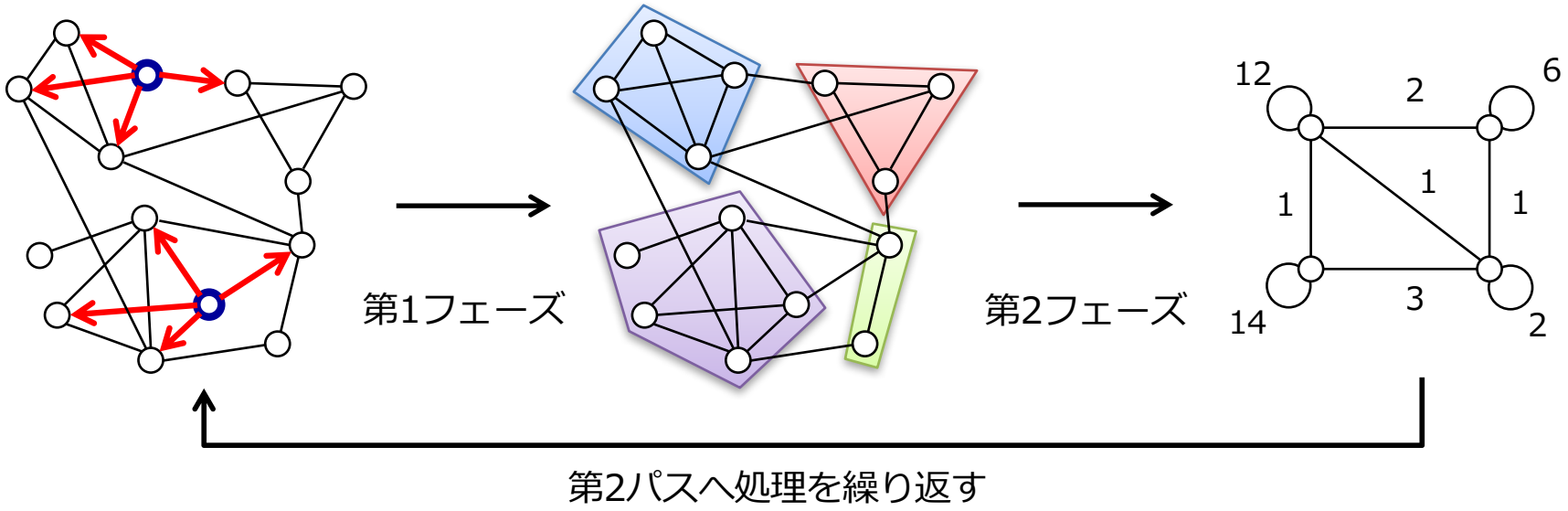
  第1フェーズ：ノードのローカルクラスタリング

  - **ランダムな順序でノードを選択**し，選択したノードをModularityが最も高くなる隣接ノードと同一クラスタとする

  第2フェーズ：クラスタに含まれるノードの一括集約

  - **同一クラスタ内のノードとエッジを一括集約し重み付きグラフに変換**

- 計算量は$O(M)$ ただし定数項が極めて大きい



Modularityによるグラフクラスタリング

Girvan-Newman法 [*Girvan et al., 2004*]
トップダウンにクラスタに存在しそうなエッジを削除する手法

Newman法 [*Newman et al., 2004*]
貪欲法によりボトムアップからModularityを向上させる手法

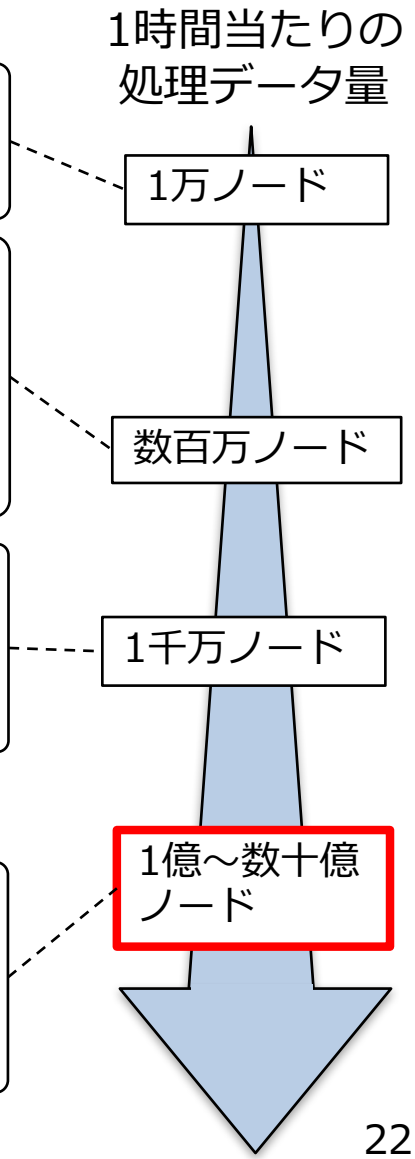
CNM法[*Clauset et al., 2004*]
ヒープの導入やヒューリスティクスによるNewman法の高速化

Louvain法[*Blondel et al., 2008*]
隣接するノード同士にのみを計算する計算量の小さな手法
最速かつ最高精度の手法として広く利用されている

Incremental Aggregation法

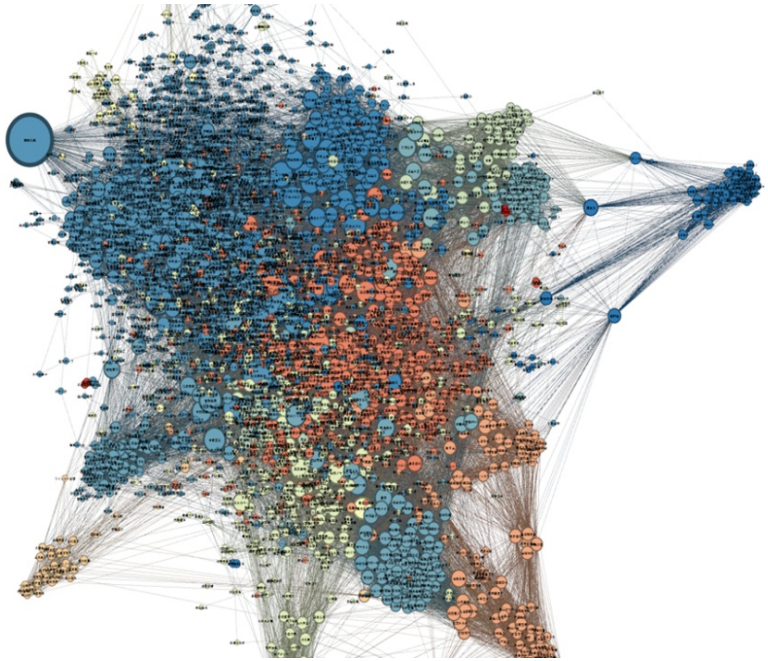
既存技術と同程度の精度を保ち10倍～20倍以上大きなデータを処理

性能目標
既存手法Louvain法より10倍以上の高速な手法の確立



Incremental Aggregation [*Shiokawa et al., AAAI'13*]

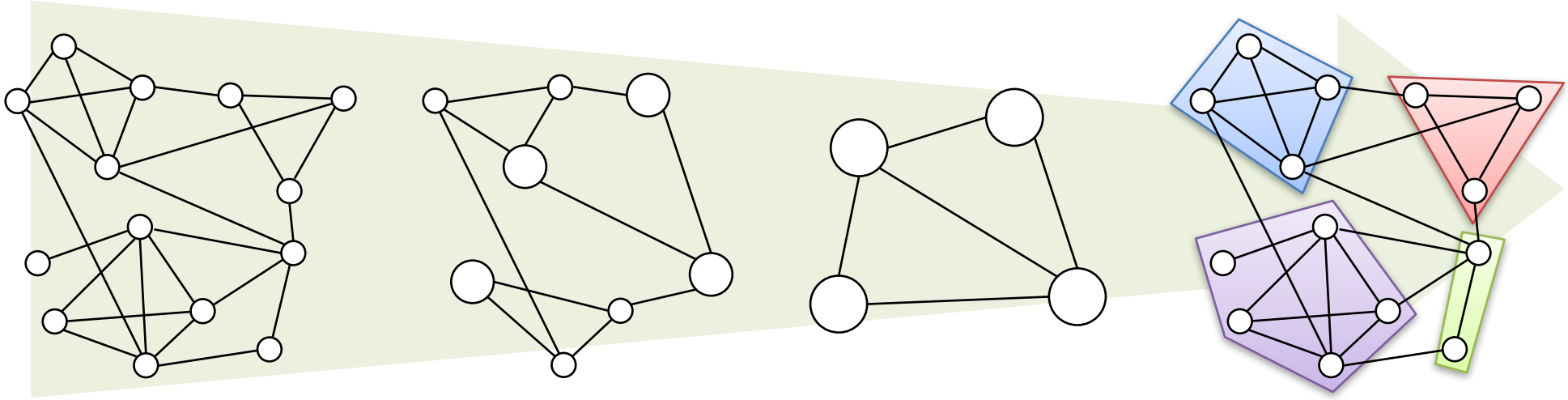
- **実世界のグラフデータが持つ構造特徴に着目**
  - **グラフのPower-Law特性**と**グラフのクラスタ性**
  - 上記の特徴に基づき不要な計算を削減する



- グラフのPower-Law特性
  大多数のノードは低次数で，
  ごく一部のノードのみ高次数となる
- グラフのクラスタ性
  隣接するノードペアは互いに多くの
  隣接ノードを共有する

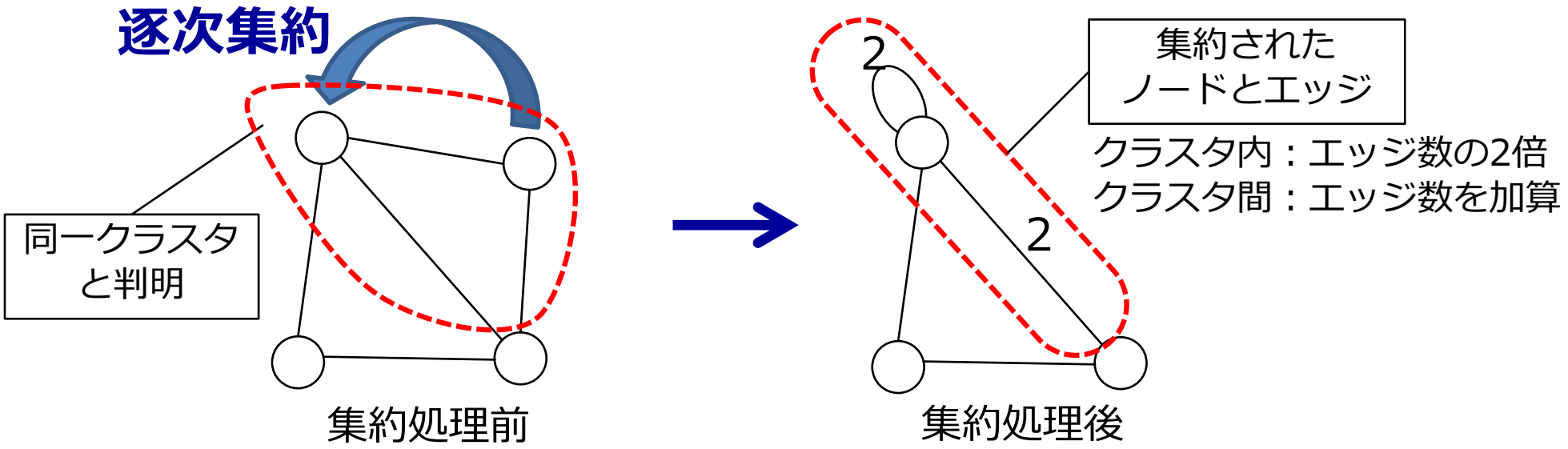
Incremental Aggregation [*Shiokawa et al., AAAI'13*]

- **逐次集約(Incremental Aggregation)の提案**
  - **グラフのクラスタ性**を利用した計算不要なエッジの集約
  - **グラフのPower-Law特性**を利用した計算エッジ数の削減



グラフのクラスタ性をとらえる

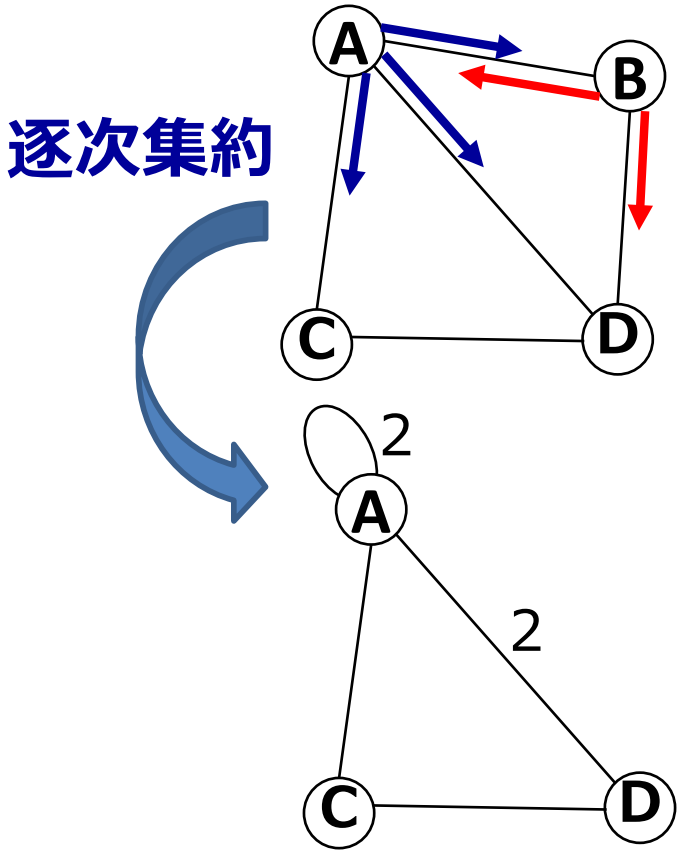
- **クラスタが決定したノードを即座に集約する**
  - 同一クラスタを1ノード，エッジを重み付きエッジに変換することで，計算対象となるノードとエッジを削減



グラフのPower-law特性を捉える

- **次数の少ない順に計算対象ノードを選択する**
  - グラフデータ中から他ノードへの次数が少ないノードを優先選択することでモジュラリティの計算回数を抑制

□ ノードAとBが同一のクラスタとなる場合



ノードAから選択
エッジ3本分計算

ノードBから選択
エッジ2本分計算

次数の少ないノードBを選択したほうが効率が良い

逐次集約による効果

ノードA・B間のエッジが集約されるため,
後続の処理で**計算が必要となるエッジ数を削減**する

実データによる評価実験

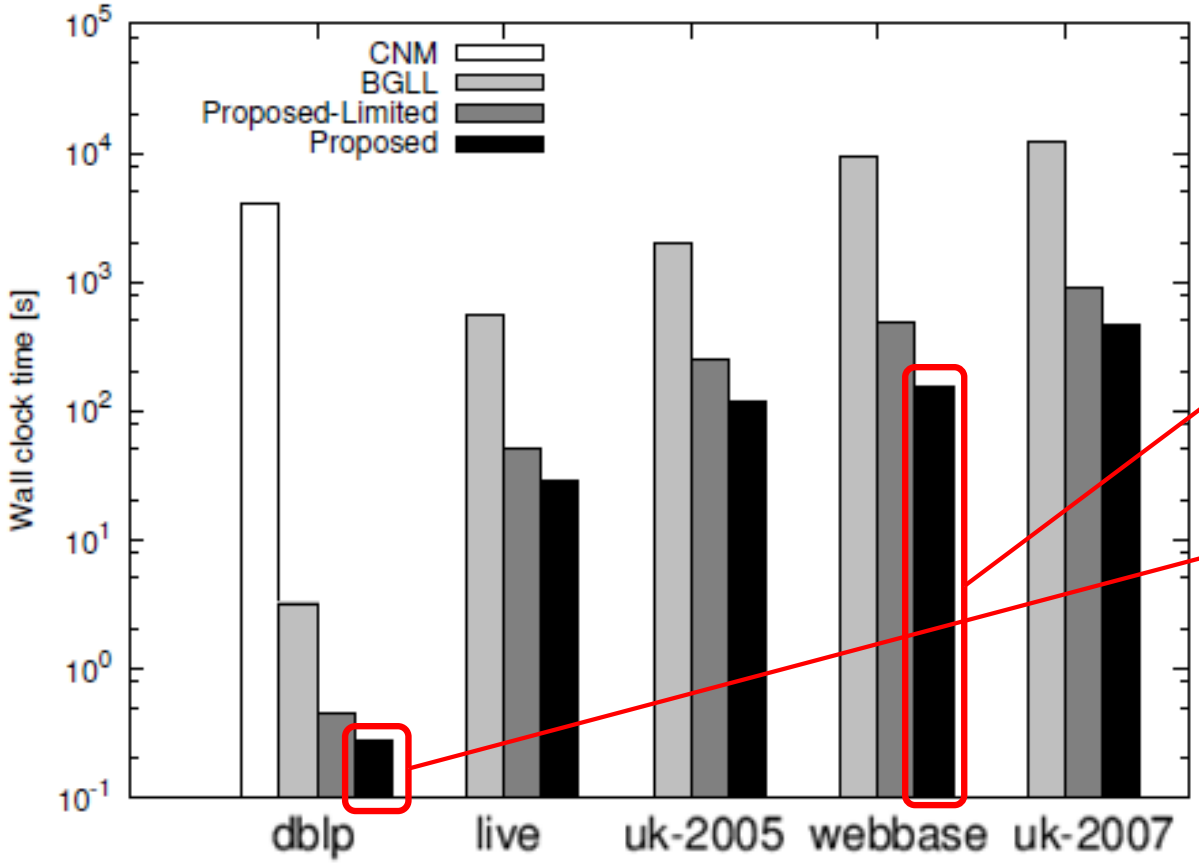
- **データセット**
  - Stanford Network Analysis ProjectおよびLaboratory of Web Algorithmsから取得した5種類のデータセットを使用

Dataset	$\|V\|$	$\|E\|$	Skewness of degree distribution
dblp	326,186	1,615,400	2.82
live	5,363,260	79,023,142	2.29
uk-2005	39,459,925	936,364,282	1.71
webbase	118,142,155	1,019,903,190	2.14
uk-2007	105,896,555	3,738,733,648	1.51

- **実験環境**
  - Linux 2.6.18 server
  - Intel Xeon CPU L5640 2.27GHz and 144GB RAM
  - State of the artの手法BGLL，CNMと比較

評価1：実行時間の比較

- **既存手法BGLLと比較して最大60倍高速**
  - 逐次集約のみ(Proposed-Limited)では約20倍の高速化
  - 次数分布の偏りが大きいグラフの高速化率が高い傾向



1億ノード/10億エッジ
を156秒でクラスタリング

3.2万ノードのグラフを
1秒未満でクラスタリング

評価2：クラスタリング結果のモジュラリティ

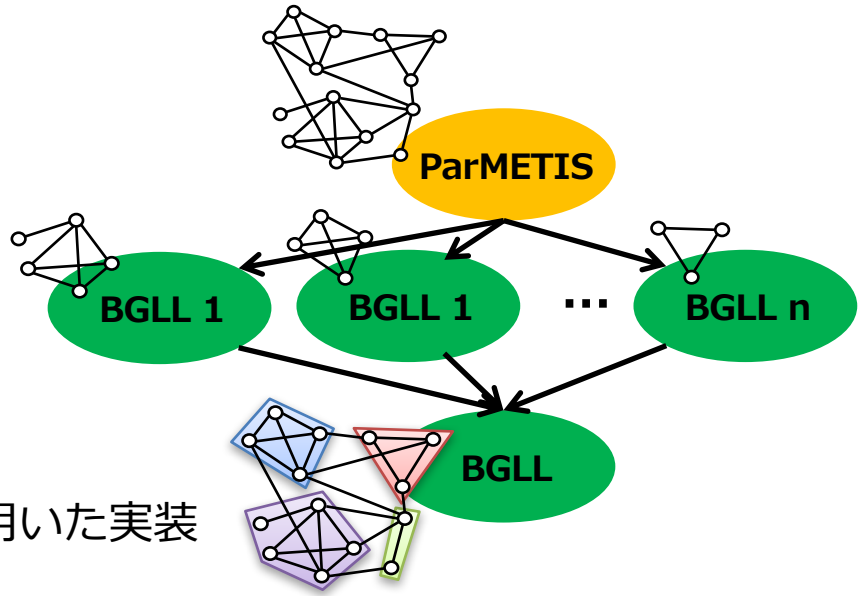
- BGLLと比較してほぼ同程度のモジュラリティ値を示すクラスタリング結果を得ることを確認

	dblp	live	uk-2005	webbase	uk-2007
Incremental aggregation	0.90	0.74	0.98	0.98	0.97
BGLL	0.88	0.74	0.97	0.96	0.97
CNM	0.82	-	-	-	-

さらなる高速化に向けて（1/2）

- **GPUを用いた並列化**
  - Single-GPU
    - [*Wu et al., 2010*]
      グラフの固有値・固有ベクトルを用いたモジュラリティクラスタリング
      べき乗法による固有値・固有ベクトルの計算をGPUで実装
  - Multi-GPU
    - [*Soman et al, 2011*]
      Label propagationをモジュラリティクラスタリングに利用した手法
      Label propagation処理部をGPUで実装

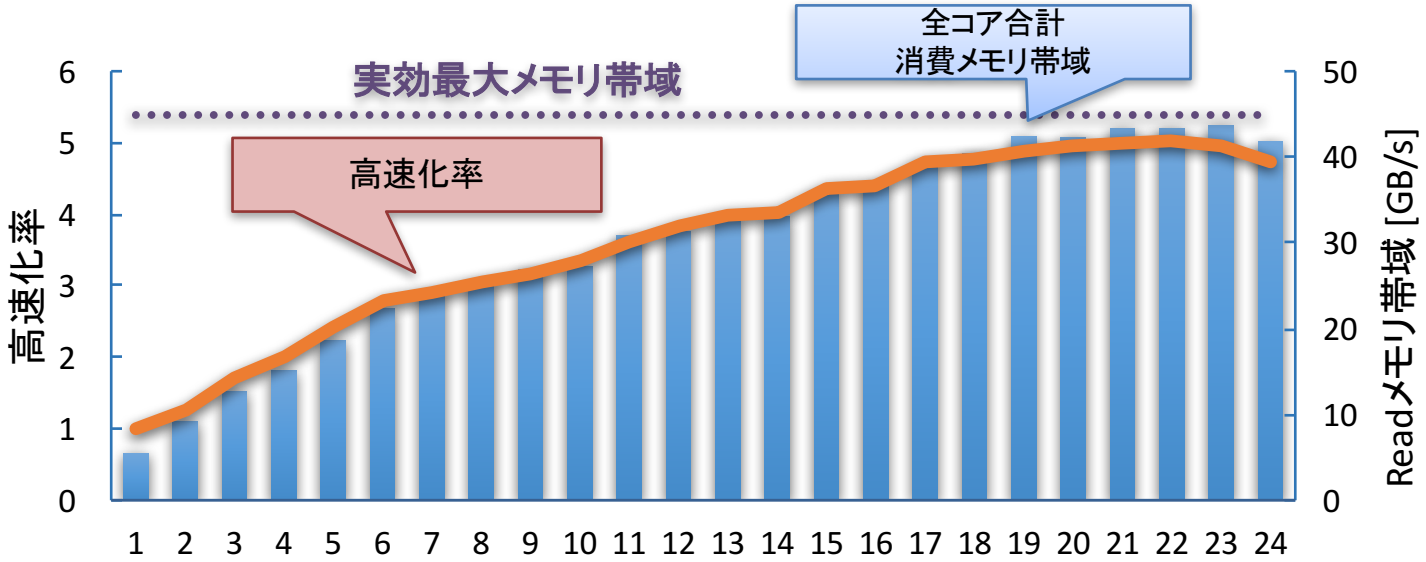
- **分散メモリ環境における並列化**
  - [*Wickramaarachchi et al., 2014*]
    グラフを最小エッジカットで事前分割し
    分割統治的にクラスタを求める
  - [Bae and Bill, SC'15]
    グラフ並列処理フレームワークGraphLabを用いた実装



さらなる高速化に向けて（2/2）

- **Multi-CPUを用いた並列化**
  - Rabbit Order [*Arai et al., IPDPS'16*]

    Incremental Aggregationクラスタリング[*Shiokawa et al, AAAI'13*]の並列化実装を利用CASを利用して分析結果の整合性を確保



- **スパコン(COMA)を用いた並列化**
  - H28年度 学際共同利用プログラム

    「46 大規模グラフ分析アルゴリズムの高速化に関する研究」

まとめ

- **本日の講演内容分析**
  - グラフクラスタリングを題材に，大規模グラフ分析アルゴリズムの研究動向を紹介した
- **Interesting Open Problems**
- スケーラブルなグラフ分析に向けたデータ構造とアルゴリズム
  - 分析処理手法に応じて大幅に変わる
  - Real-world propertyを捉えることが重要
- 分析アルゴリズムのツール化
  - Xeon Phi, GPU, etc.
- アプリケーション分野での応用
  - 生命情報，宇宙，物理，化学，医療，…