

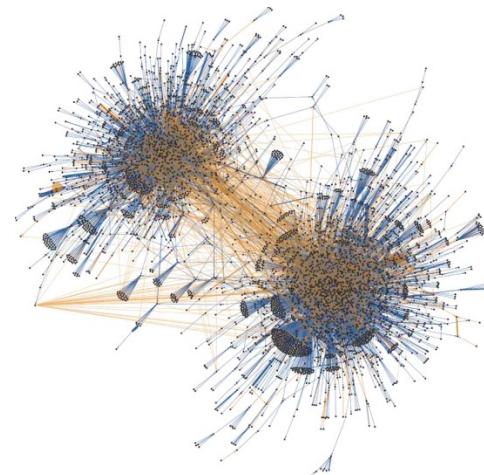
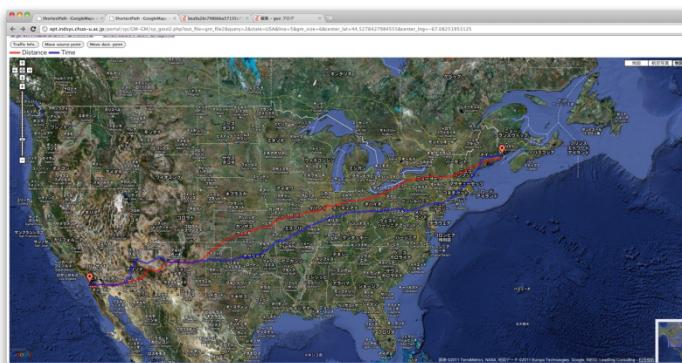
「高速かつ省電力なグラフ解析とその実応用」

藤澤 克樹

九州大学マス・フォア・インダストリ研究所

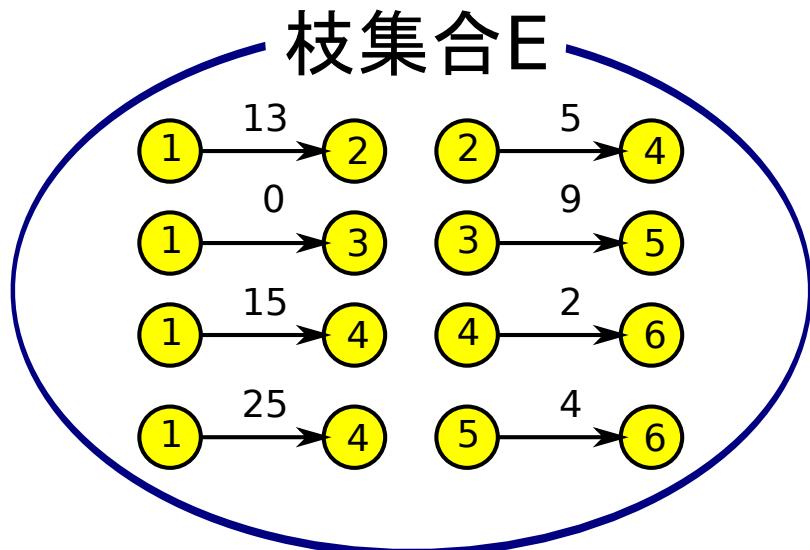
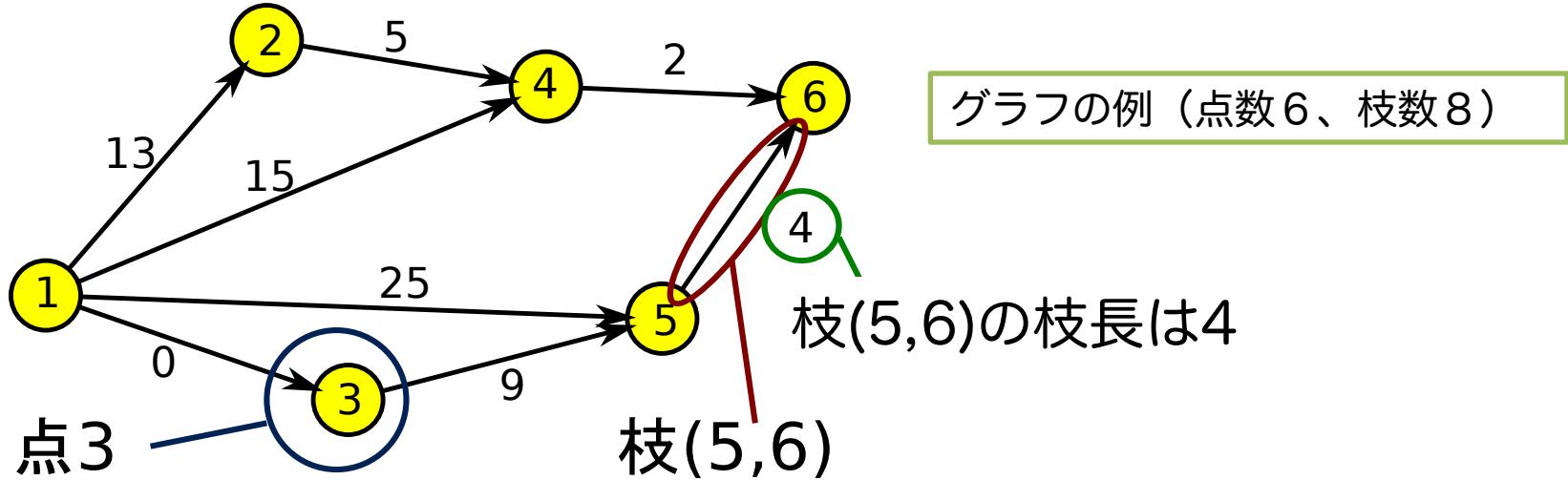
ACSI 2016: チュートリアル

2016年1月19日



グラフとは？

- グラフは点と枝で構成される



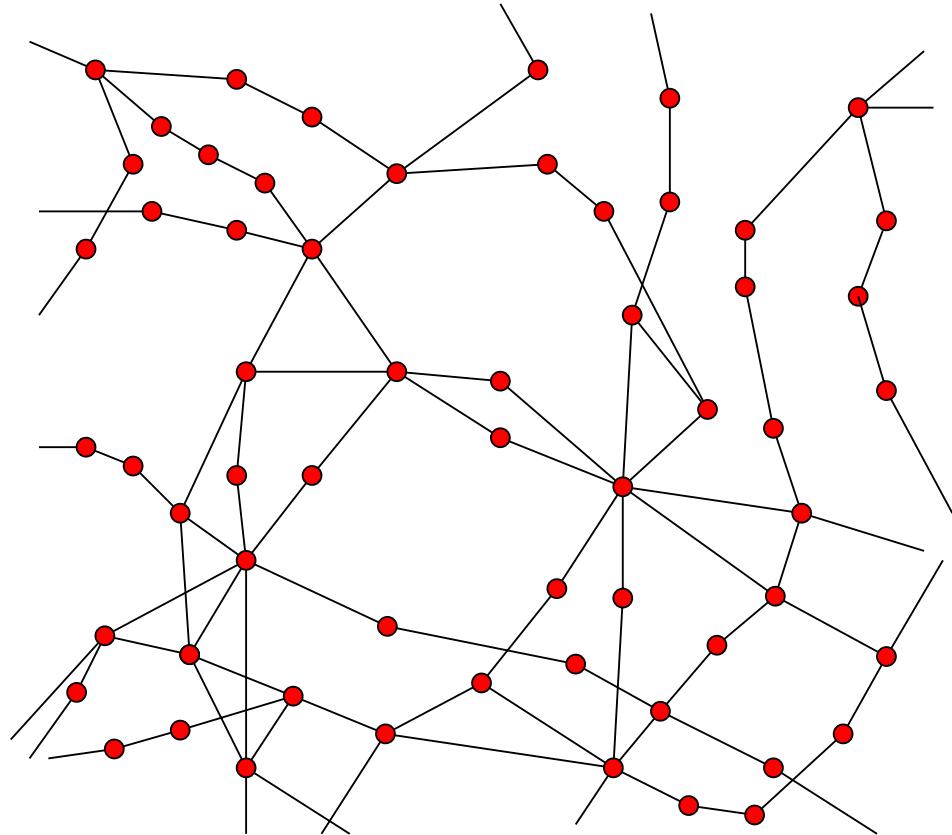
実社会ネットワークのグラフ化

- 道路ネットワーク
 - NY のジョージワシントン橋周辺



実社会ネットワークのグラフ化

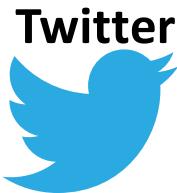
- 鉄道ネットワーク
- 東京メトロの路線図



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

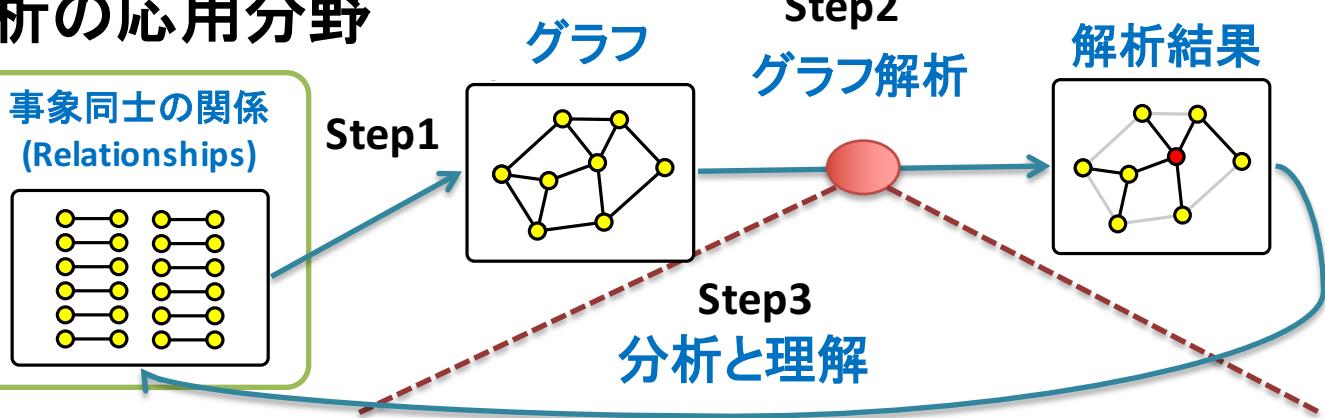
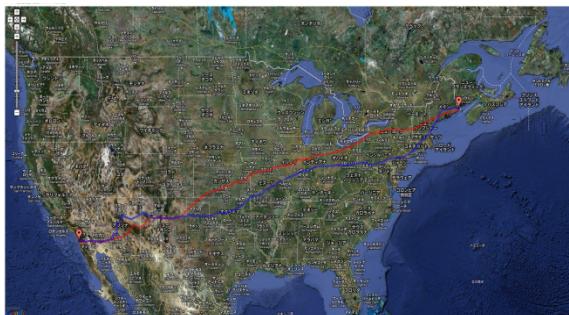
事象同士の関係
(Relationships)



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



- 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)

ニューラル・ネットワーク
890億点 & 100兆枝

サイバーセキュリティ
150億/日 のアクセスログ

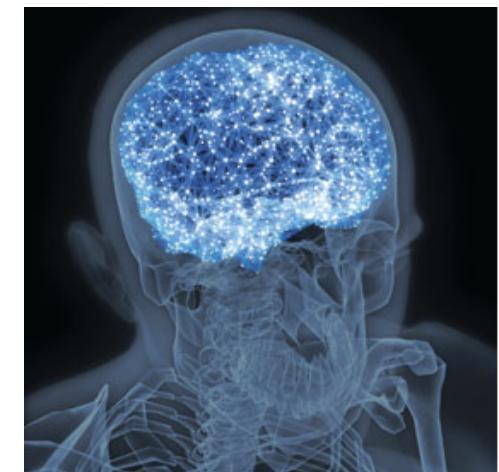
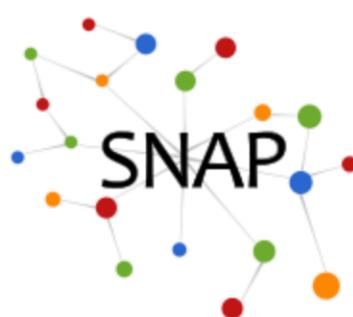


Image: Illustration by Mirko Ilic



[SNAP for C++](#) ▶
[SNAP for Python](#) ▶
[SNAP Datasets](#) ▶
[What's new](#)
[People](#)
[Papers](#)
[Citing SNAP](#)
[Links](#)
[About](#)
[Contact us](#)

Open positions

We have filled all the positions for this quarter.
[More info.](#)

Stanford Large Network Dataset Collection

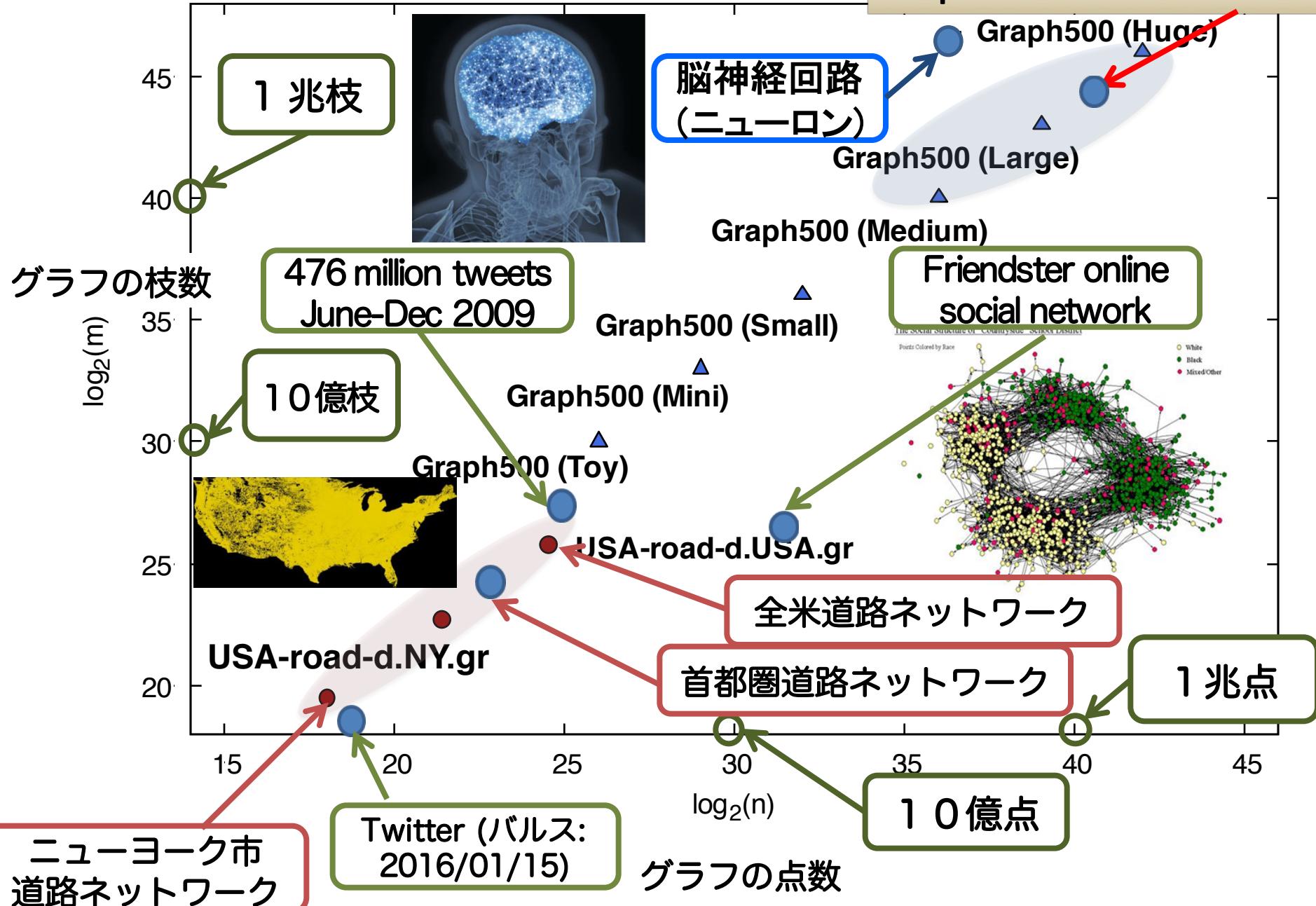
- [Social networks](#) : online social networks, edges represent interactions between people
- [Networks with ground-truth communities](#) : ground-truth network communities in social and information networks
- [Communication networks](#) : email communication networks with edges representing communication
- [Citation networks](#) : nodes represent papers, edges represent citations
- [Collaboration networks](#) : nodes represent scientists, edges represent collaborations (co-authoring a paper)
- [Web graphs](#) : nodes represent webpages and edges are hyperlinks
- [Amazon networks](#) : nodes represent products and edges link commonly co-purchased products
- [Internet networks](#) : nodes represent computers and edges communication
- [Road networks](#) : nodes represent intersections and edges roads connecting the intersections
- [Autonomous systems](#) : graphs of the internet
- [Signed networks](#) : networks with positive and negative edges (friend/foe, trust/distrust)
- [Location-based online social networks](#) : Social networks with geographic check-ins
- [Wikipedia networks and metadata](#) : Talk, editing and voting data from Wikipedia
- [Twitter and Memetracker](#) : Memetracker phrases, links and 467 million Tweets
- [Online communities](#) : Data from online communities such as Reddit and Flickr
- [Online reviews](#) : Data from online review systems such as BeerAdvocate and Amazon

SNAP networks are also available from [UF Sparse Matrix collection](#). [Visualizations of SNAP networks](#) by Tim Davis.

• Social networks

Name	Type	Nodes	Edges	Description
ego-Facebook	Undirected	4,039	88,234	Social circles from Facebook (anonymized)
ego-Gplus	Directed	107,614	13,673,453	Social circles from Google+
ego-Twitter	Directed	81,306	1,768,149	Social circles from Twitter
soc-Epinions1	Directed	75,879	508,837	Who-trusts-whom network of Epinions.com
soc-LiveJournal1	Directed	4,847,571	68,993,773	LiveJournal online social network
soc-Pokec	Directed	1,632,803	30,622,564	Pokec online social network
soc-Slashdot0811	Directed	77,360	905,468	Slashdot social network from November 2008
soc-Slashdot0922	Directed	82,168	948,464	Slashdot social network from February 2009

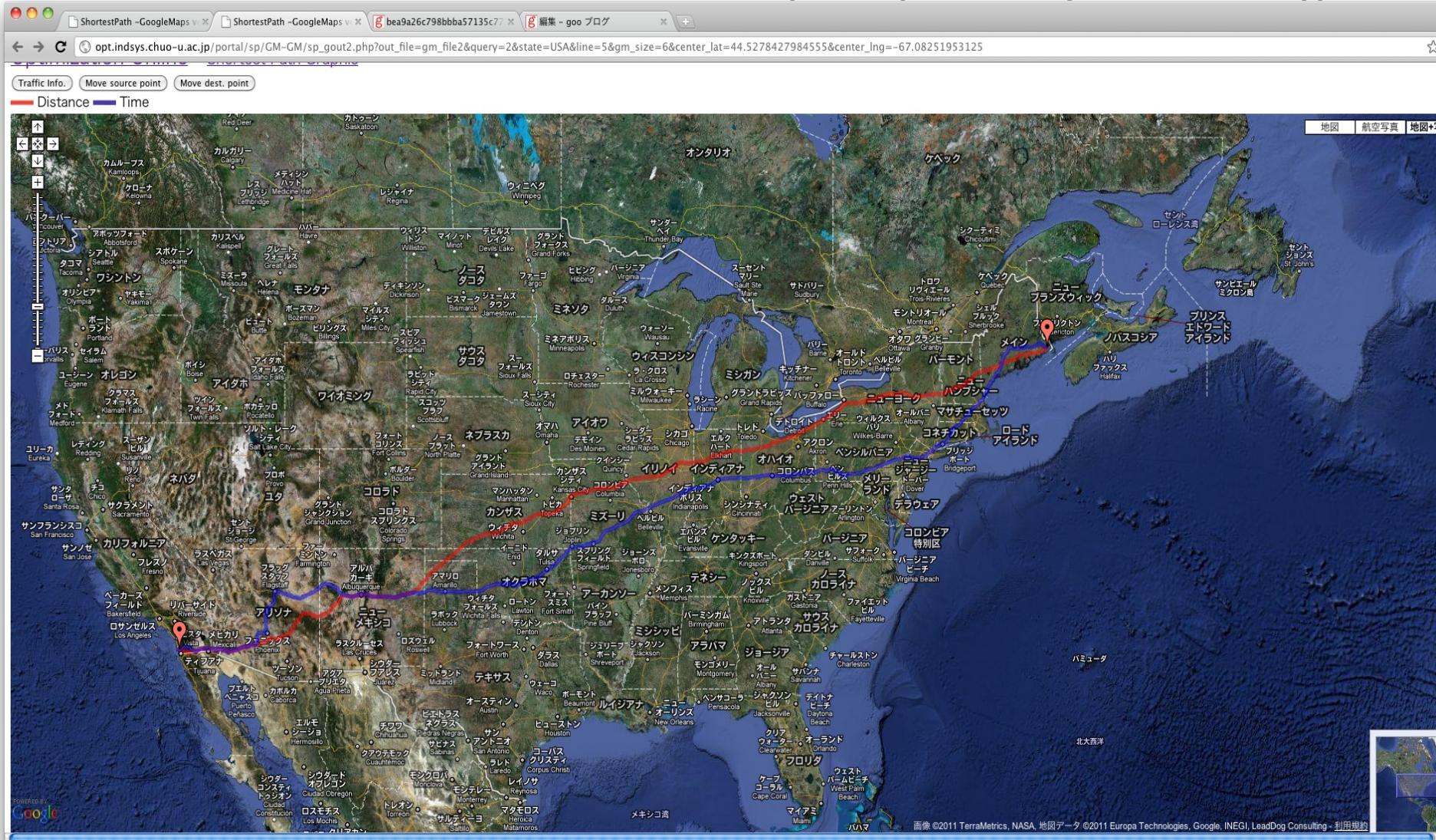
京コンピュータ: 82944ノード
Graph500: 38621 GTEPS



全米横断の最短路: 2400万点, 5800万枝 → 約2秒で計算(3W)

サンディエゴから北東部のメイン州までの最短路

(赤: 最短距離、青: 最短時間): <http://opt.imi.kyushu-u.ac.jp/>

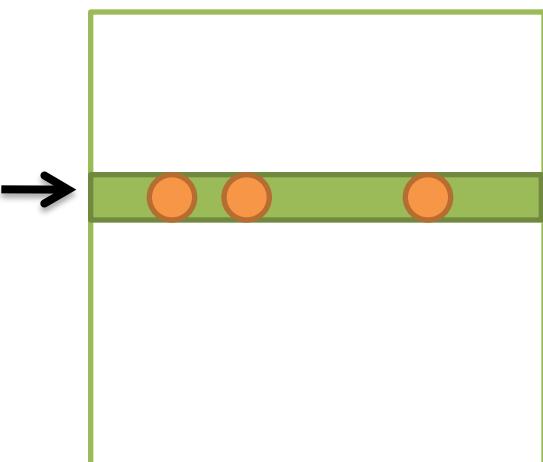


グラフ表現 (全米グラフ表現時のメモリ量)

■ 距離行列 ($2.03\text{ PB} (2^{50})$)

かなり密向き。

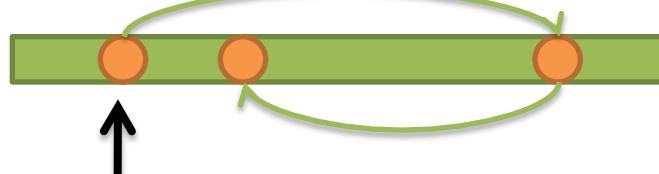
疎なグラフでは、無駄な領域が大きい。



■ 線形リスト ($0.98\text{GB} (2^{30})$)

疎向き。

枝数分の領域しか必要としない。
構築コストがほとんどなし。



■ フォワード・スター ($0.54\text{GB} (2^{30})$)

疎向き。

枝数分の領域しか必要としない。

構築時にソートが必要。



		終点			
		1	2	3	4
始点	1	0	5	1	2
	2	3	0	∞	4
	3	∞	∞	0	3
	4	2	∞	∞	0

始点	ポインタ	終点	枝長
0	-	2	5
1	0	3	1
2	3	4	2
3	5	1	3
4	6	4	4
5	7	4	3
6		1	2

Forward-Star グラフ表現

0. 元ファイル

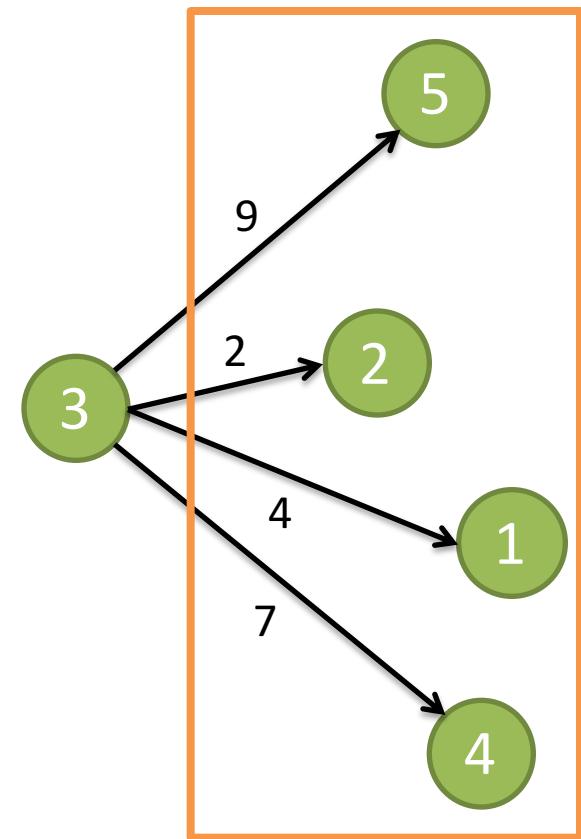
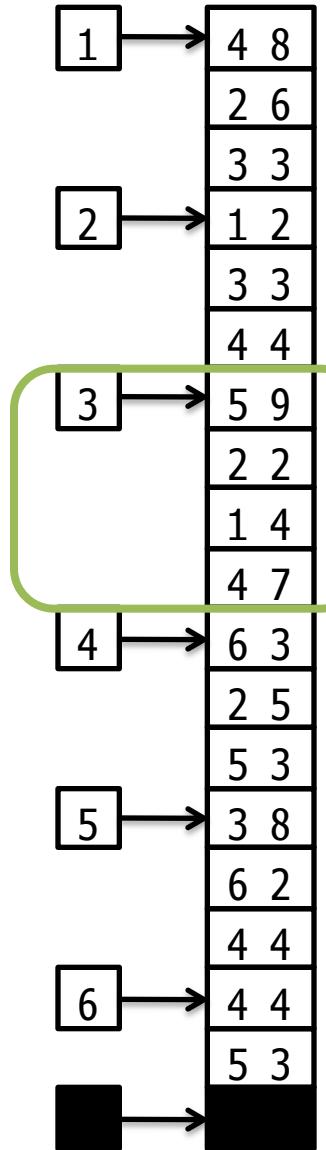
```
p sp 6 18
a 3 2 2
a 1 2 6
a 2 1 2
a 5 6 2
a 6 4 4
a 1 3 3
a 3 5 9
a 4 2 5
a 1 4 8
a 2 3 3
a 4 6 3
a 5 3 8
a 2 4 4
a 3 1 4
a 3 4 7
a 4 5 3
a 5 4 4
a 6 5 3
```

1. 読込み & 始点でソート

1	4	8
1	2	6
1	3	3
2	1	2
2	3	3
2	4	4
3	5	9
3	2	2
3	1	4
3	4	7
4	6	3
4	2	5
4	5	3
5	3	8
5	6	2
5	4	4
6	4	4
6	5	3

a “始点ID” “終点ID” “枝の重み”

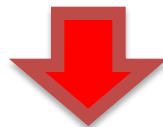
2. 始点をまとめる



●連続的に配置
→キャッシュ・メモリの
ヒット率向上

グラフ解析の予備知識（1）

- **PageRank** : Webの検索システムにおける検索結果の表示順の判断基準の1つとして使用される → 重要なWebページは、たくさんのページからリンクされる
- **MapReduce** : 巨大なデータセットに対する分散コンピューティングを支援する目的で、Googleによって2004年に導入されたプログラミングモデル
- **Hadoop** : MapReduceとHDFSで構成 → HDFSは、MapReduceで処理するデータを扱う分散ストレージで、複数のマシンを1つのストレージとして扱うことができる → ところがグラフ問題に使いにくい一面がある



- **Pregel** はグラフ問題に特化した解決策としてグーグル社で開発された → **BSP (Bulk Synchronous Parallel)** の採用

グラフ解析の予備知識（2）

Pregel における **BSP (Bulk Synchronous Parallel)**

1. Concurrent Computation

タスクの計算を行う所。ノード間で並行に行われる。

2. Communication

各ノード間で通信をする所。

3. Synchronization

全ての処理を待ち受ける所（同期処理）。全てのタスクが合流する。

- Pregel に関するオープンソース

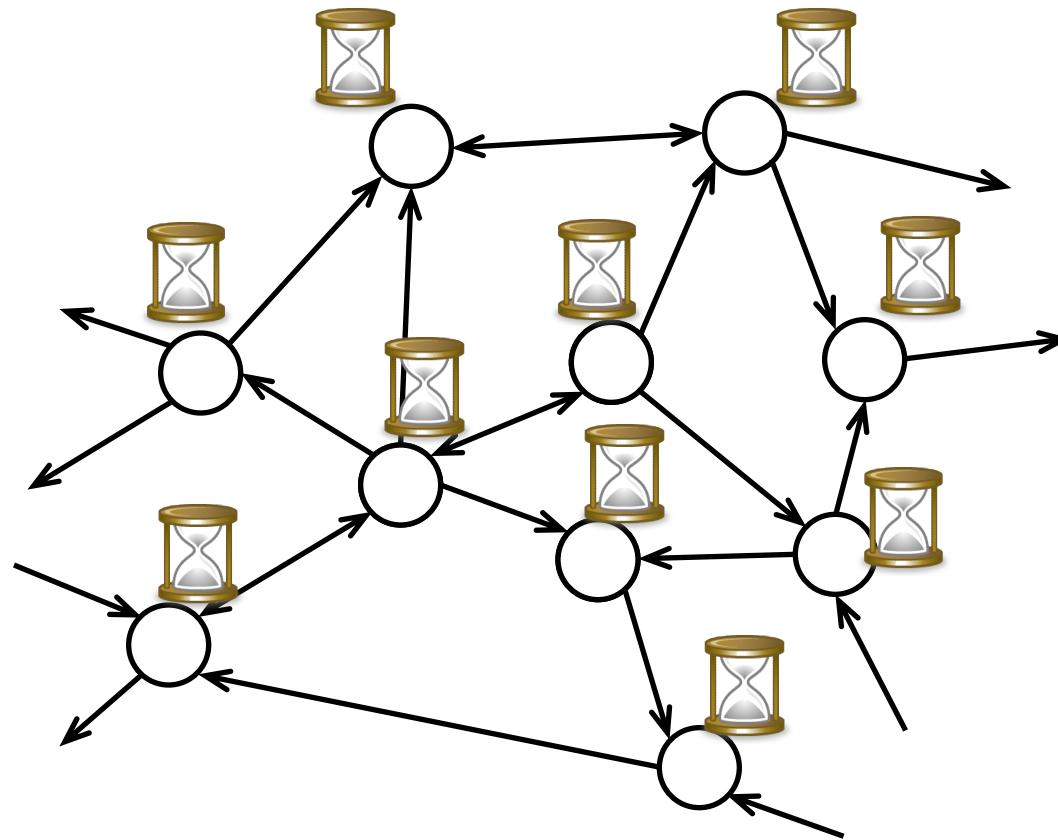
1. Giraph : Hadoop環境上で稼働する

2. Jpregel : 独自の分散環境で稼働する

3. ScaleGraph : X10 上で稼働する

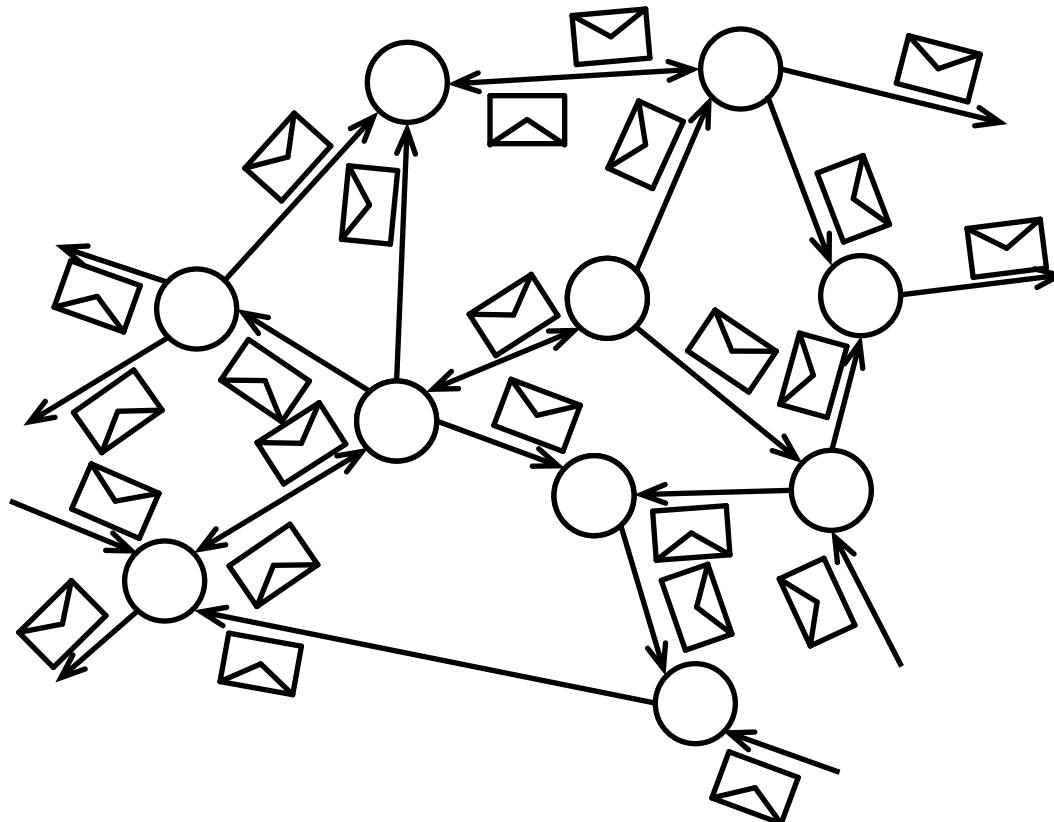
Pregel プログラミングモデル (1)

- 各点での状態を初期化する : Each vertex initialize its state.



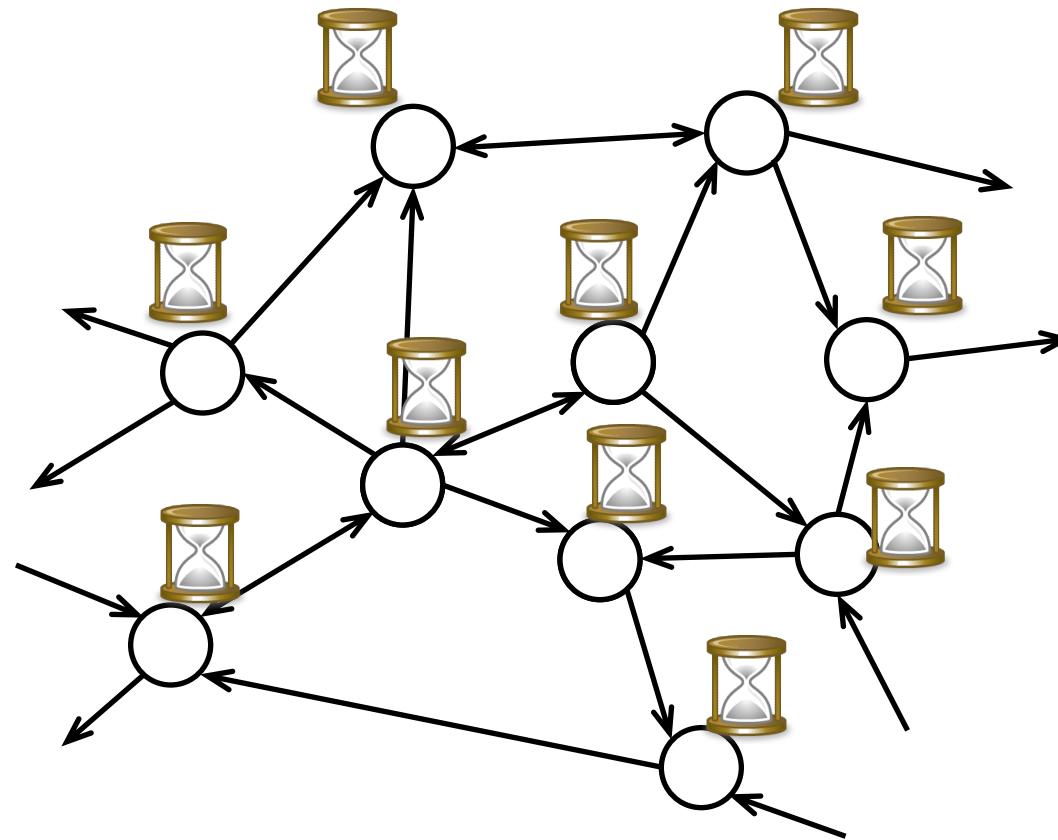
Pregel プログラミングモデル（2）

- 各点は他の点にメッセージを送信する
- Each vertex send messages to other vertices.



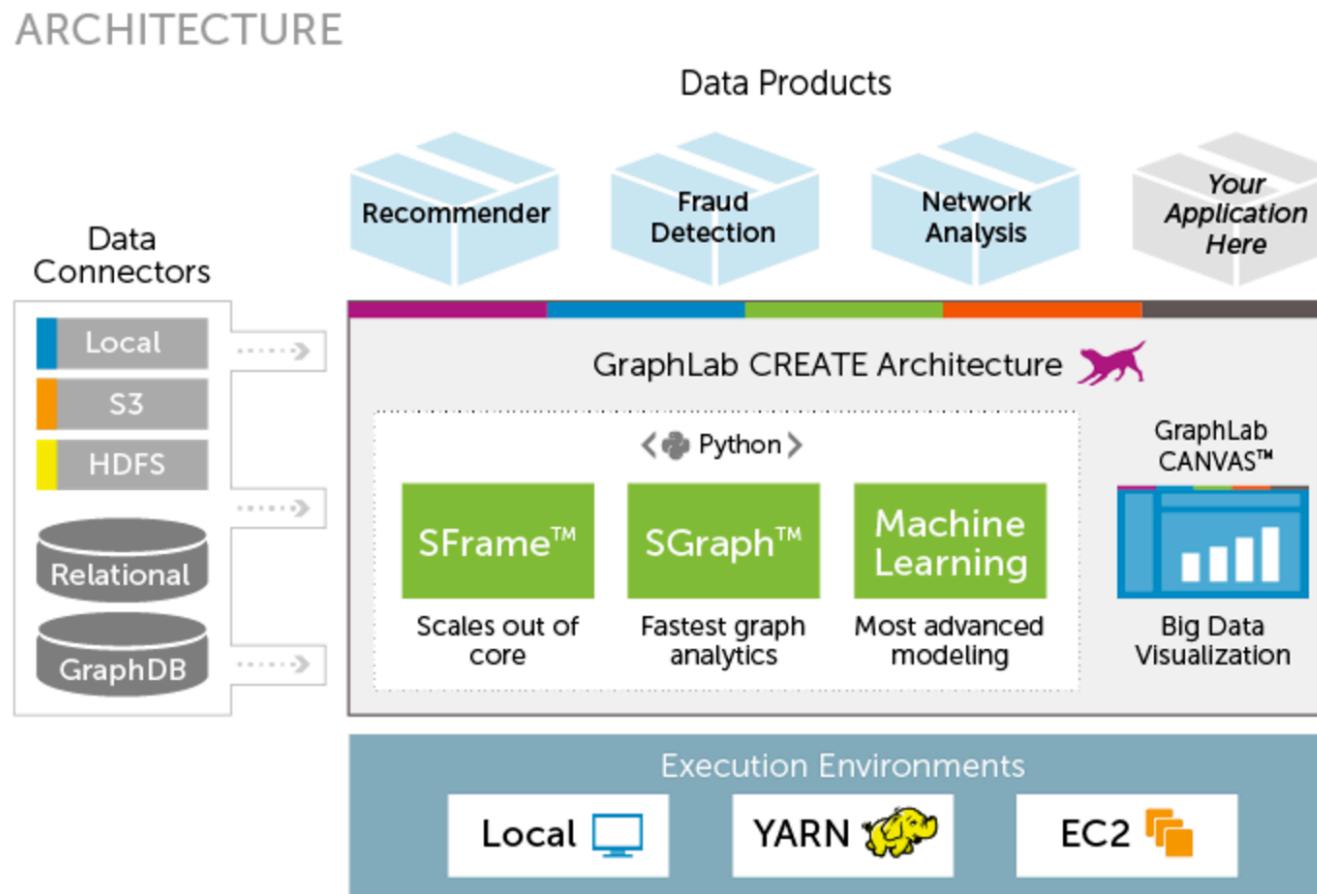
Pregel プログラミングモデル（3）

- 各点は受信したメッセージを処理して、状態を更新する
- Each vertex process received messages and update its state.



Graph解析処理系の紹介(1)

- **GraphLab CREATE** (<https://dato.com/products/create/>)
- 主に機械学習用のグラフ解析ツール



- — GiraphとGraphLabに関してはIBM developerWorksに詳しい
- <http://www.ibm.com/developerworks/jp/opensource/library/os-giraph/>

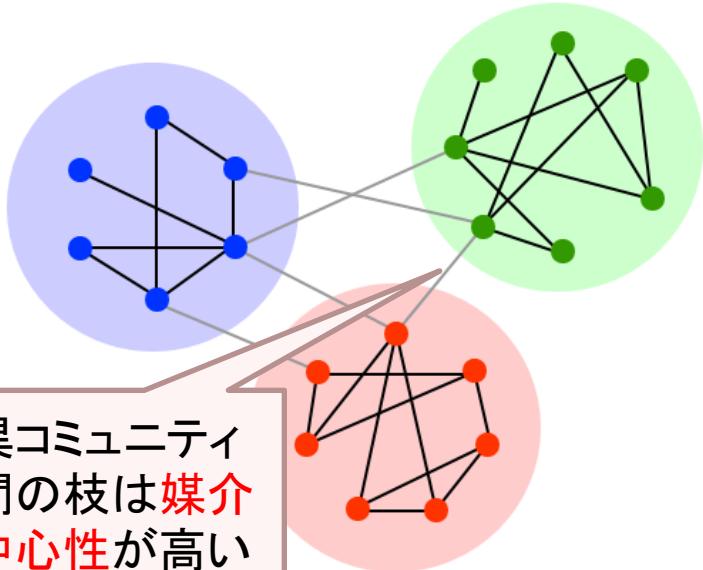
最新の Graph解析処理系の紹介（2）

- **GraphX**
 - Sparkと呼ばれるMapReduce実装で動作するグラフ処理系
 - GraphX: A Resilient Distributed Graph System on Spark
 - Apache Spark のコンポーネントのひとつ
 - 大容量のグラフデータを並列分散環境で処理するためのフレームワーク
- <https://amplab.cs.berkeley.edu/publication/graphx-grades/>
- **X-stream**
- <http://labos.epfl.ch/x-stream>
- **Neo4J**
- <http://www.neo4j.org>
- - Javaで記述されたグラフデータベース
- - オライリーから本が出版されている (<http://graphdatabases.com>)

ネットワークにおけるコミュニティ検出

ネットワークのコミュニティ分割

コミュニティ：密な部分ネットワーク



異コミュニティ
間の枝は媒介
中心性が高い



盛んに研究されツールも多数

モジュラリティ

コミュニティ分割の“良さ”を測る代表的な指標

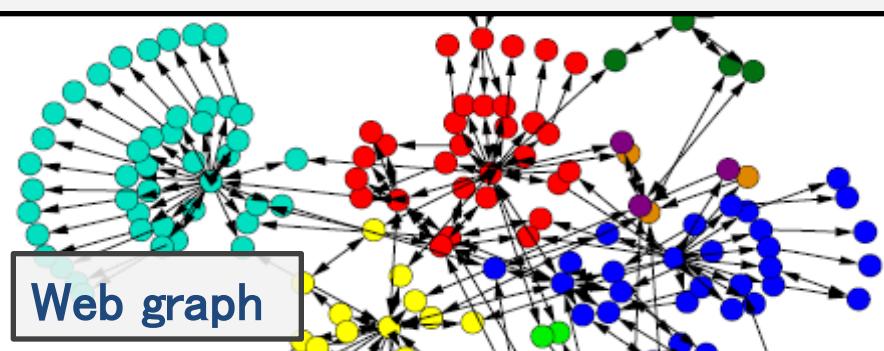
$$Q := \sum_i \{e_{ii} - a_i^2\}, \quad a_i := \sum_i e_{ij}.$$

e_{ij} はコミュニティ i とコミュニティ j を結ぶ枝が全ての枝に対して占める割合

様々なヒューリスティックスが提案

(※モジュラリティ最大化はNP困難)

- GN法 … 媒介中心性が高い枝を繰り返し削除
- Newman法… コミュニティを逐次結合する局所探索
- CNM法… Newman法のデータ構造を改良
- Louvain法… コミュニティ全体が1つのノードとなるようにグラフを縮約しながら局所探索



Web graph

中心性の定義

v-t 間最短路長

- Closeness Centrality

$$C_C(v) = \frac{1}{\sum_{t \in V} d_G(v, t)}$$

- Graph Centrality

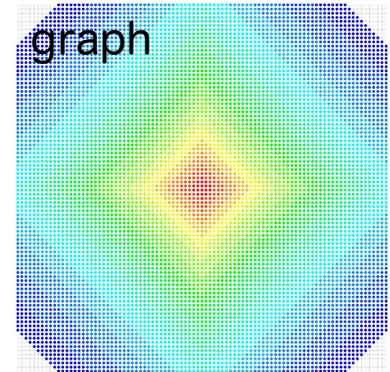
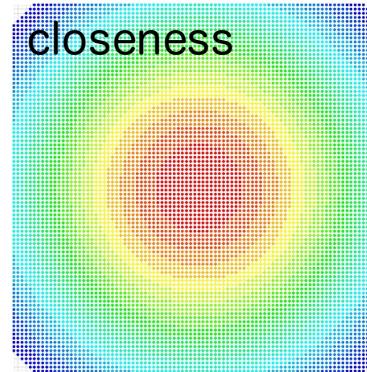
$$C_G(v) = \frac{1}{\max_{t \in V} d_G(v, t)}$$

- Stress Centrality

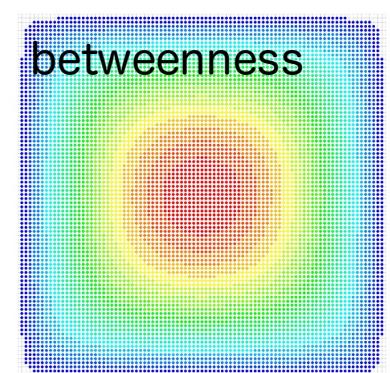
$$C_S(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v)$$

- Betweenness Centrality

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$



v を通る s-t 間最短路数



Brandes's Algorithm

始点数に応じて、厳密解(全点) \Leftrightarrow 近似解(始点を Random Sampling)

始点数分の
繰り返し

1. 始点を1つ選ぶ
2. 1で決定した点を始点とする
 - 幅優先探索 (BFS) … 全ての枝長が1
 - 最短路問題 (SSSP) … 枝長を考慮
3. 各点の最短路長、最短路数から **中心性** を計算

Betweenness Centrality

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

- (s, t) 間の最短路長 : $d_G(s, t), (s, t \in V)$
 - $d_G(s, s) = 0 (s \in V)$
 - $d_G(s, t) = d_G(t, s), (s, t \in V)$
- (s, t) 間の最短路数 : $\sigma_{st}, (s, t \in V)$
 - $\sigma_{ss} = 1, (s \in V)$
 - $\sigma_{st} = \sigma_{ts}, (s, t \in V)$

```

1:  $C_B[v] \leftarrow 0, v \in V$ 
2: for  $s \in V$  do
3:    $\sigma, S, P \leftarrow \text{BFS}^*(s)$ 
4:    $\delta[v] \leftarrow 0, v \in V$ 
5:    $S \leftarrow S \setminus s$ 
6:   while  $S \neq \emptyset$  do
7:      $w \leftarrow \text{pop}(S)$ 
8:     for  $v \in P[w]$  do
9:        $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]} \cdot (1 + \delta[w])$ 
10:       $C_B[w] \leftarrow C_B[w] + \delta[w]$ 
11:    end for
12:  end while
13: end for
14: return  $C_B$ 

```

NETAL (NETwork Analysis Library)

- 高速な最短路計算・中心性指標計算
 - NUMA の特徴を自動的に検出しアフィニティ設定を行う
 - 全米道路ネットワーク (2500万点) に対する全体全最短路を 7.75 日で計算可能で、9th DIMACS の参考実装に比べて 230 倍高速
 - 複数の指標 Closeness, Degree, Graph, Stress, Betweenness (BC), edge-Betweenness などに対応し、GraphCT (BCのみ) より 13 – 26 倍高速、SSCA#2 (BCのみ) より 2.4 – 3.8倍高速

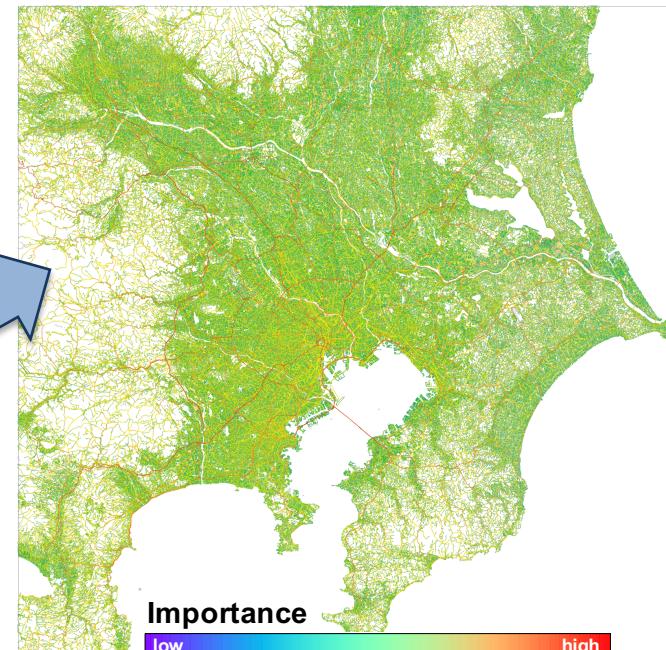
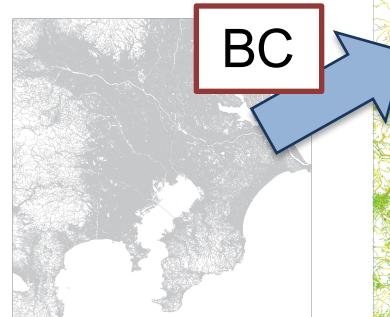
Closeness [1]
$$C_C(v) = \frac{1}{\sum_{t \in V} d_G(v, t)}$$

Graph [2]
$$C_G(v) = \frac{1}{\max_{t \in V} d_G(v, t)}$$

Stress [3]
$$C_S(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v)$$

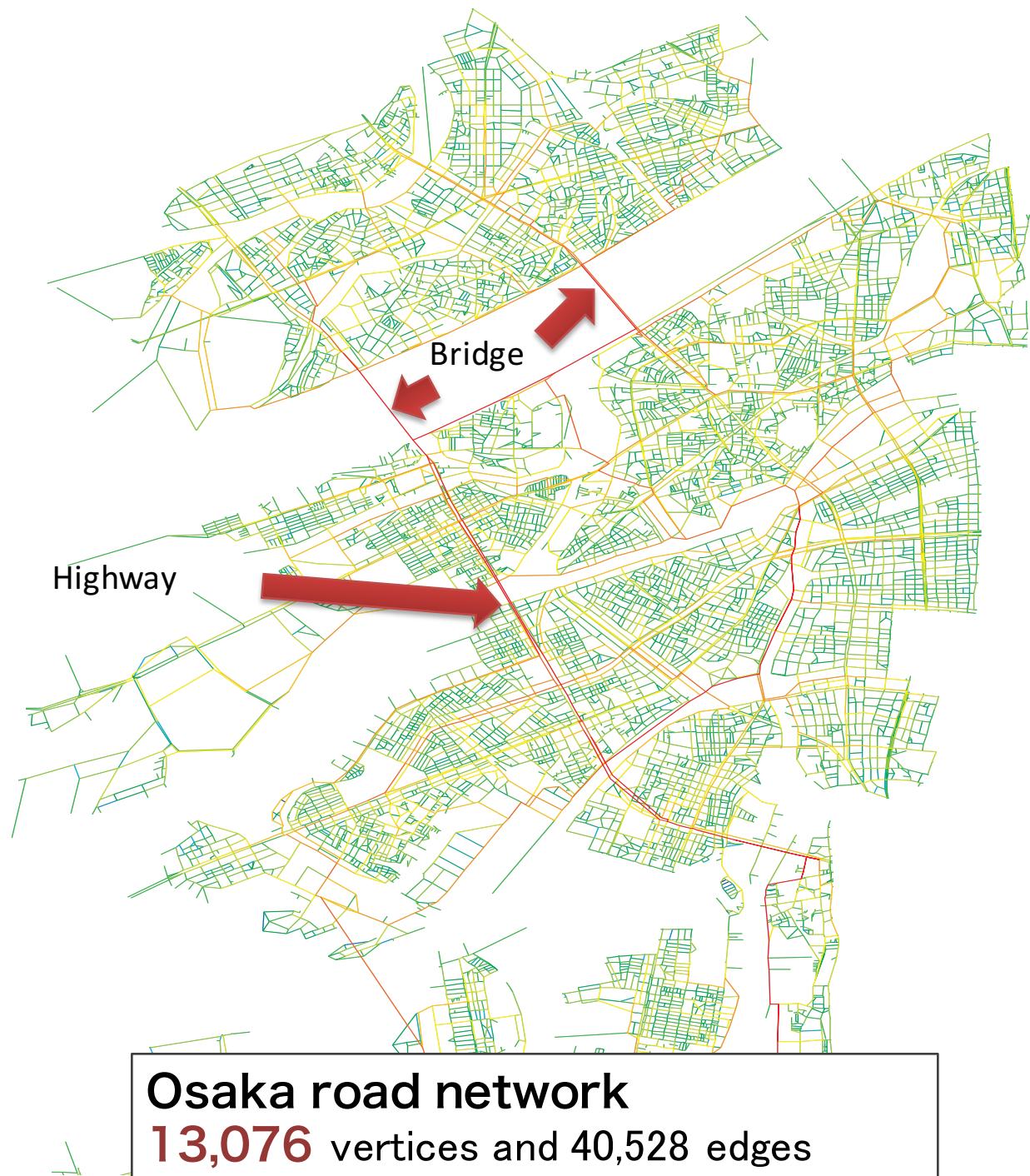
Betweenness [4]
$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

関東周辺
交通ネットワーク
650万点/1446万枝



点の重要性

- グラフの特徴を示す指標
 - 中心性、PageRank …
- 中心性指標が広く利用されている
 - 近年は中心性が盛んに研究されている
- 中心性指標にもいくつか種類がある
 - 各点に接続されている枝数 (Degree Centrality)
 - **各点が最短路に含まれる回数 (Betweenness Centrality)**
 - 最も遠い点までの最短距離 (Graph Centrality)
 - 各点までの最短距離 (Closeness Centrality)

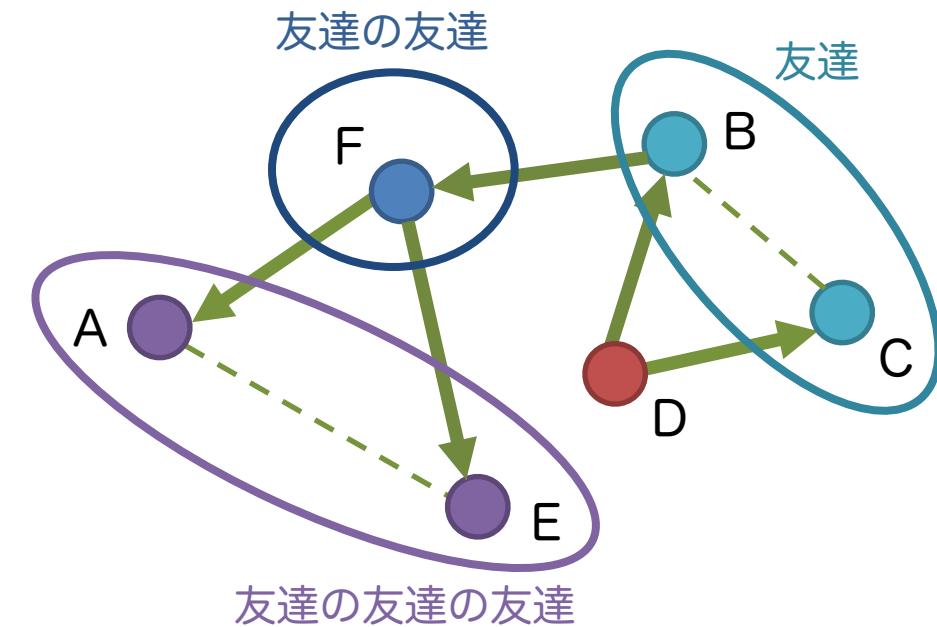
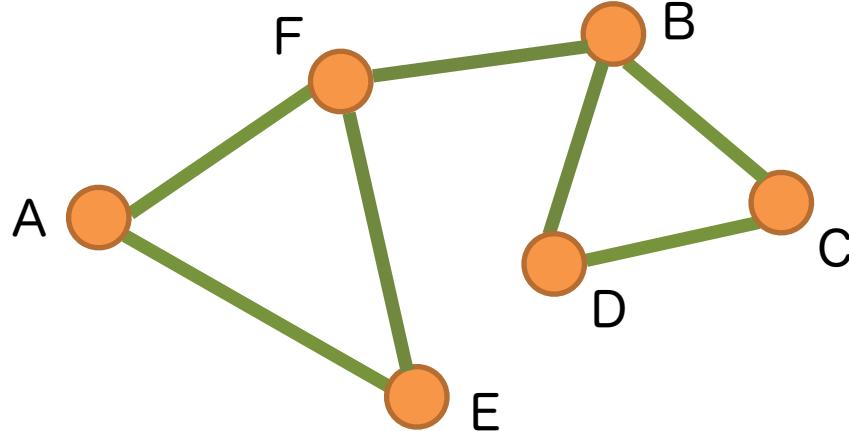


ソーシャルネットワーク

- 問題

- A 君の友達は E さんと F 君
- B 君の友達は C さんと D さんと F 君
- C 君の友達は D さん
- E 君の友達は F 君
- さて D さんの友達の友達は誰？

答) F 君

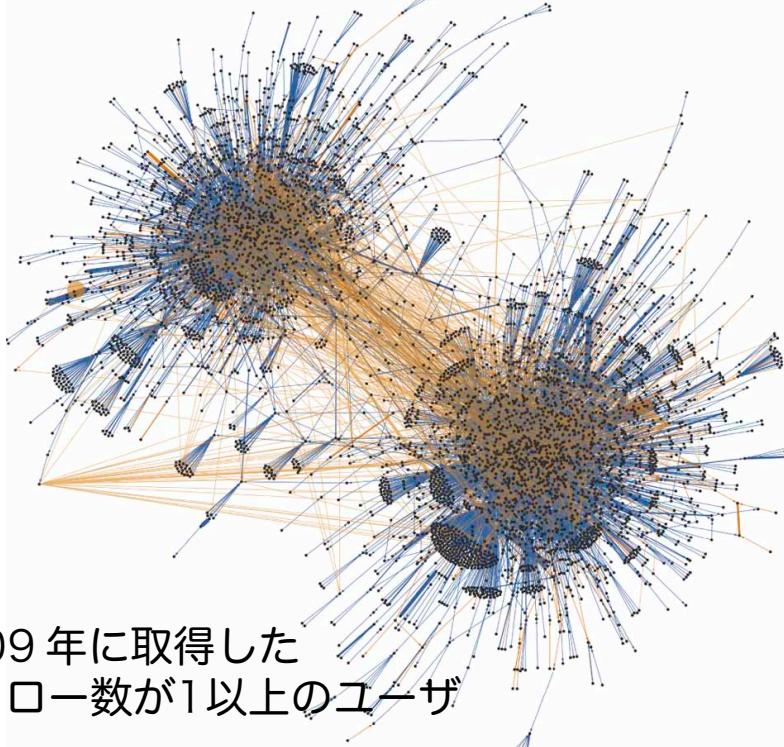




Graph500技術の応用 Twitter ネットワークの解析

• フォロー・ネットワーク

- ユーザ数 (点数) 41,652,230
- フォロー関係 (枝数) 2,405,026,092



2009年に取得した
フォロー数が1以上のユーザ

• Graph500 ベンチマーク

- 幅優先探索の性能「**1秒間に通過した枝数 TEPS**」を用いて、コンピュータの性能を比較する



ユーザ 21,804,357 からの幅優先探索の結果

ホップ数	ユーザ数	割合 (%)	累積割合 (%)
0	1	0.00	0.00
1	7	0.00	0.00
2	6,188	0.01	0.01
3	510,515	1.23	1.24
4	29,526,508	70.89	72.13
5	11,314,238	27.16	99.29
6	282,456	0.68	99.97
7	11536	0.03	100.00
8	673	0.00	100.00
9	68	0.00	100.00
10	19	0.00	100.00
11	10	0.00	100.00
12	5	0.00	100.00
13	2	0.00	100.00
14	2	0.00	100.00
15	2	0.00	100.00
合計	41,652,230	100.00	-

0.069 秒で探索可能
 $\Rightarrow 21.28 \text{ GTEPS} (10^9 \text{ TEPS})$



Wikipedia ネットワーク

- 記事ネットワーク
 - 記事間はハイパーリンクによって参照し合っている
 - 任意の記事から 6 回リンクをたどると、どの記事にもいける（らしい）
 - 6次の隔たりとして有名



- Wikitter
 - 指定された 2 記事間の距離を計算（何回リンクを辿ったかを計測）
 - 距離が遠い（繋がりの薄い）記事を探してみようというゲーム

<http://wikitter.info>

wikitter
ういきったー

2013年8月20日のお題は「弓矢」

本日のお題
弓矢
挑戦する！
本日の達成件数：4件
本日のお題とは？

弓矢 → 高校野球
登録されています。
探索開始！
※テキストボックスには記事ページ名を入力ください。

経路数：3経路
弓矢
↓
スポーツ
↓
野球
↓
高校野球

探索時間
探索時間：0.093秒

弓矢から高校野球までのリンクの辿り方



【演習】6 経路の事柄を探そう

【発展】得られた経路を観察し、事柄の関係性について考察してみよう

経路数	例題	1	2	3	4	5
0	弓矢	電子励起爆薬				
1	スポーツ	TNT換算				
2	野球	1945年				
3	高校野球	大正				
4		日本の歴史				
5		院政期文化				
6		かえる				
7						
経路数	3	6				

Graph500 · Green Graph500

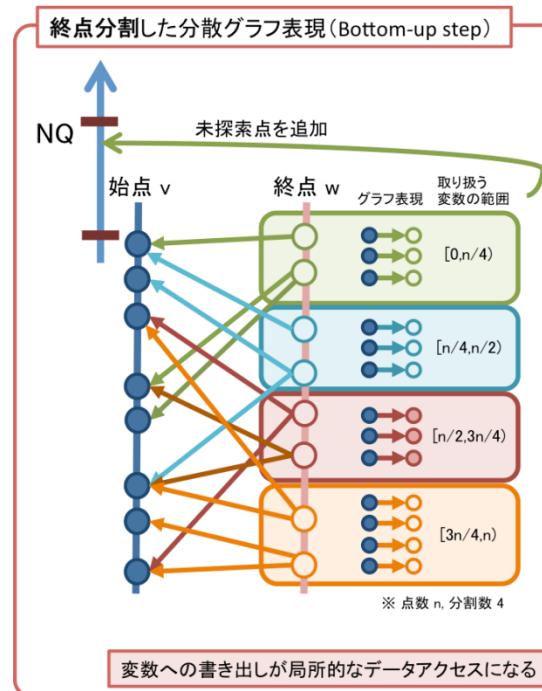
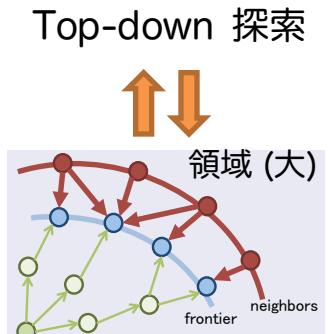
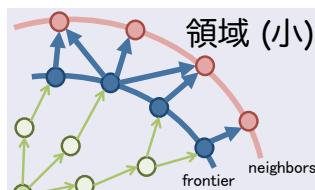


○ Graph500 · Green Graph500 ベンチマーク

- パラメータ **SCALE** と **edgefactor** (=16) から、点数 $n=2^{\text{SCALE}}$, 枝数 $m=\text{edgefactor} \cdot n$ となる Kronecker graph を生成
- 幅優先探索 (BFS) での**1秒辺りの探索枝数 TEPS** により、**Graph500** リストを作成
- 消費電力あたりの TEPS (TEPS/W) により、**Green Graph500** リストを作成

○ 高速な幅優先探索の実装 BFS

- 計算機性能を引き出す汎用的な高速化



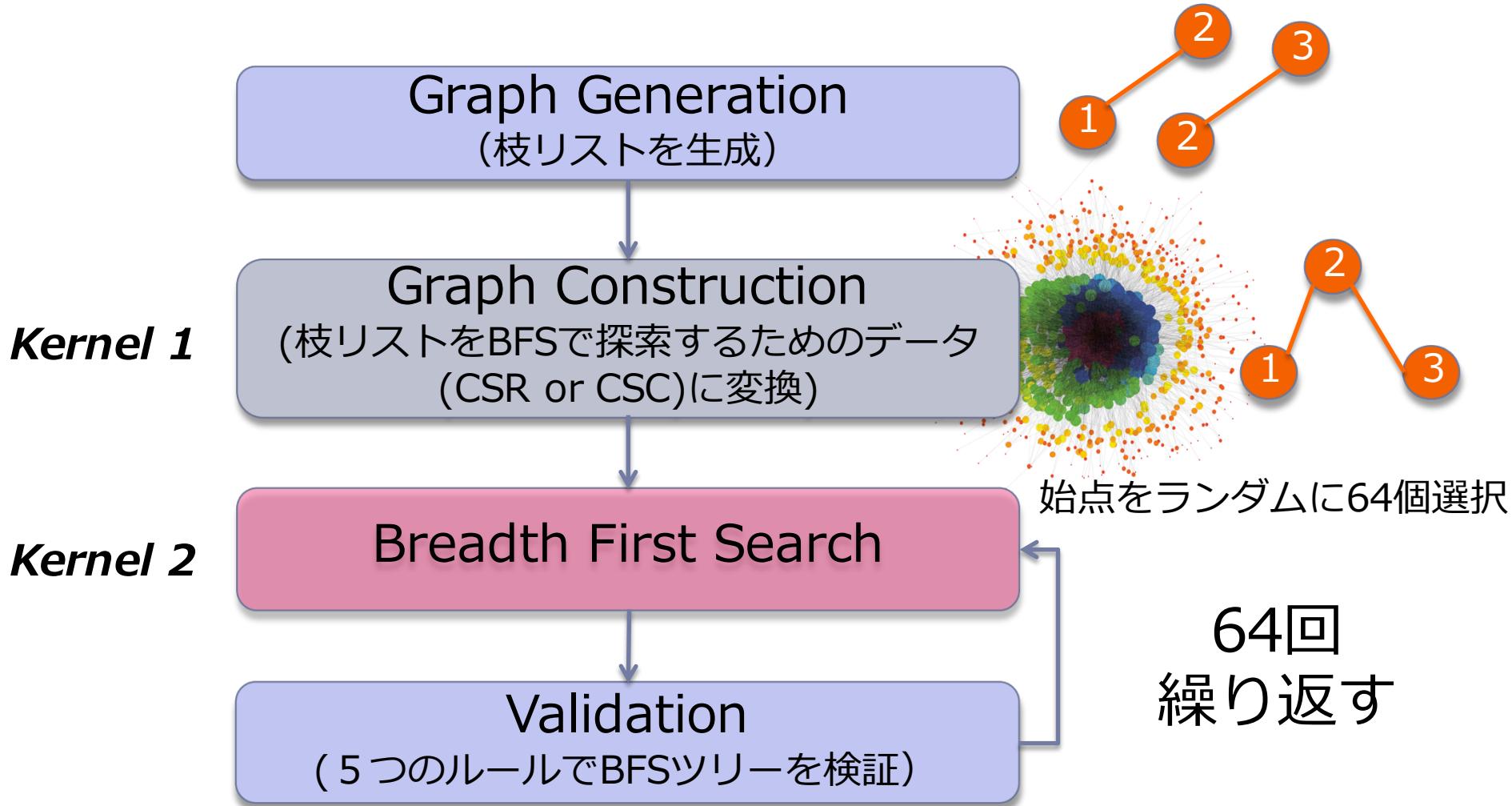
○ Green Graph500 (June 2013)



Rank	MTEPS/W	Site	Machine	G500 scale	GTEPS	Nodes
1	64.12	Chuo University	GraphCREST-Tegra	132	20	0.153885
2	53.82	Chuo University	GraphCREST-Intel-NUC	110	23	1.08175
3	53.47	Chuo University	GraphCREST-Mac-mini	90	24	1.94104
4	52.02	Chuo University	GraphCREST-MBA13	105	23	1.22761
5	51.62	Chuo University	GraphCREST-Reli615	89	24	1.98735
6	39.29	Changsha, China	TH-LV-FEP/C	-	26	9.74402
7	32.25	Chuo University	GraphCREST-NEXUS10	133	20	0.118688
8	30.41	Tokyo Institute of Technology	Uchi-2012: ultrabook+GPU	76	24	5.12372
9	20.58	Chuo University	GraphCREST-SBeep0	72	25	6.14248
10	18.68	Chuo University	GraphCREST-SBeep2.9	67	25	6.77685
11	17.39	Chuo University	GraphCREST-4waysBep2.4	57	26	11.1148
12	10.95	Chuo University	GraphCREST-Wex40	58	26	11.061
13	2.43	The University of British Columbia	Aikiindi	106	28	1.20576
14	1.89	University of Luxembourg	Virdis HPC@Unilu	-	21	0.389643
15	0.20	Swiss National Supercomputing Center	Todai	49	29	15.5983



Graph500ベンチマークの実行順

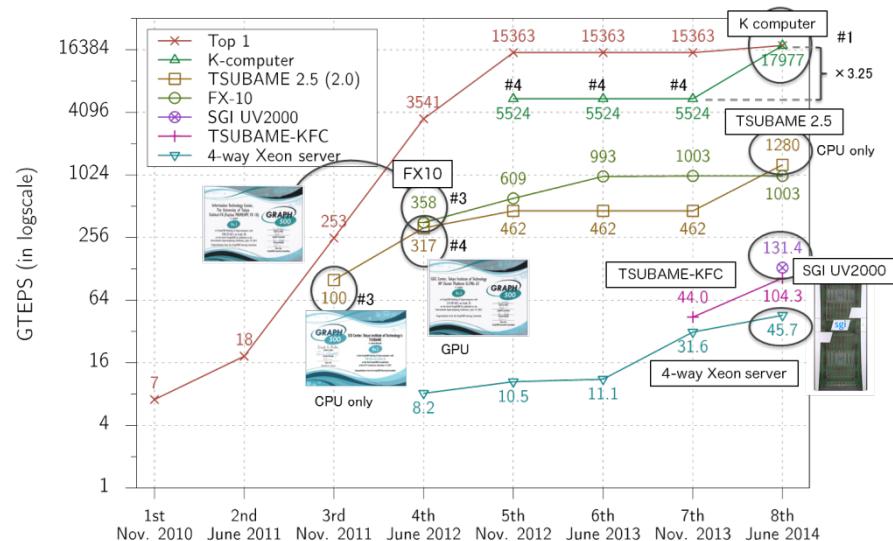
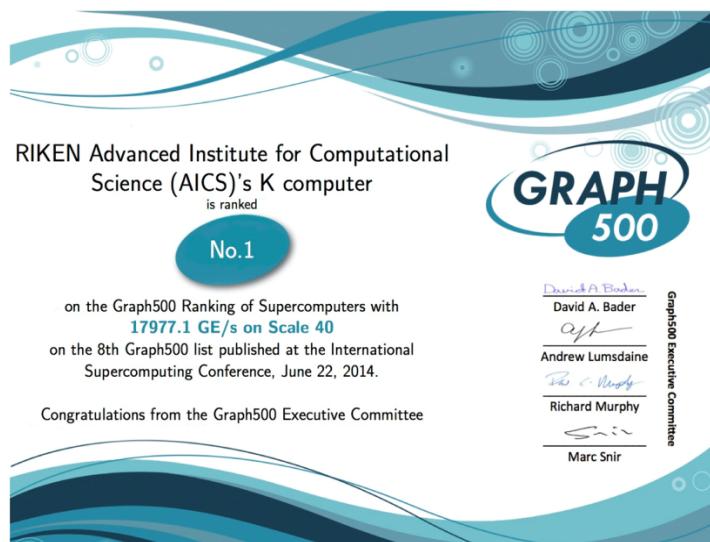


※CSR: Compressed Sparse Row, CSC: Compressed Sparse Column

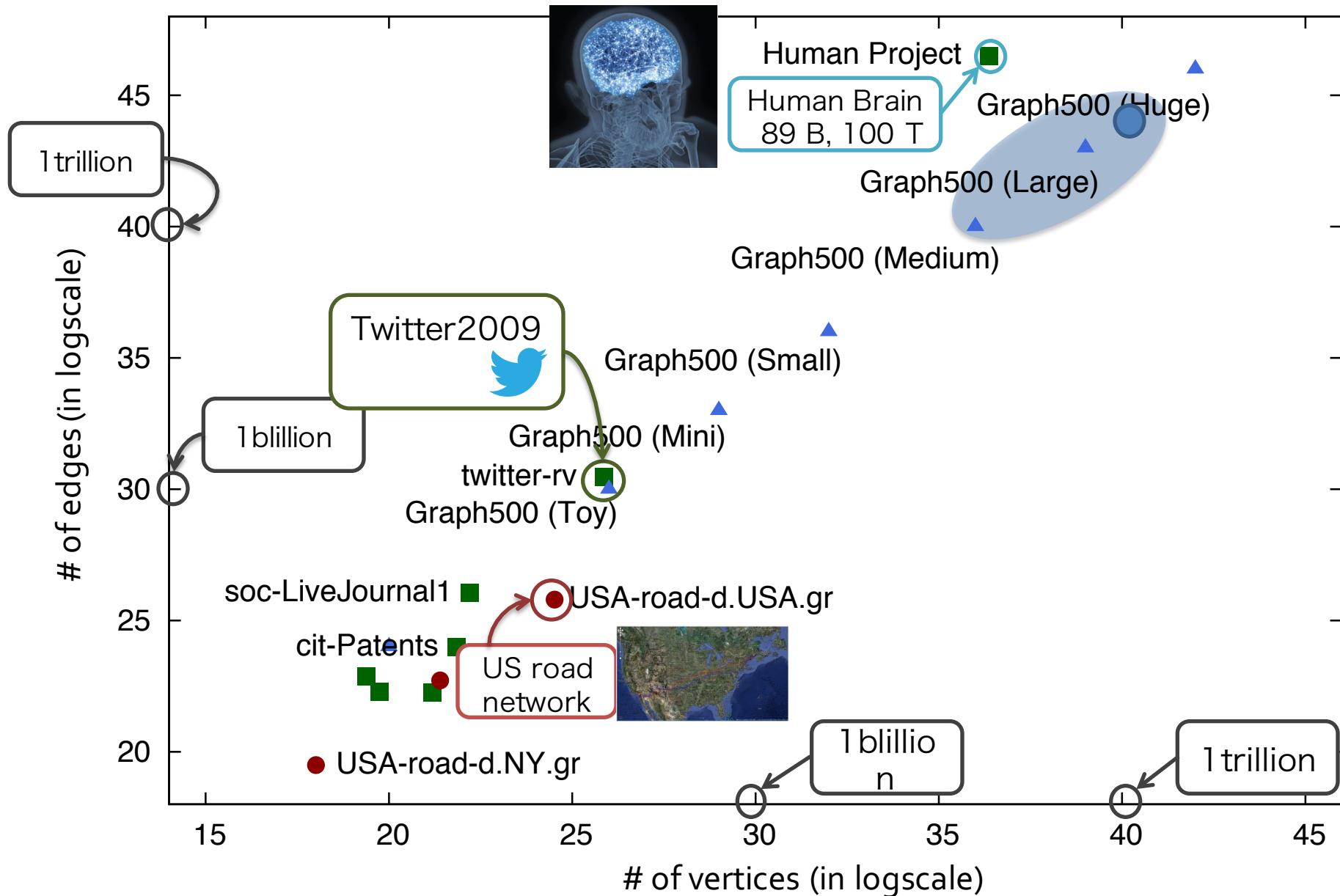
図5－6／スーパーコンピュータ「京」の概要と主な成果

1. 概要

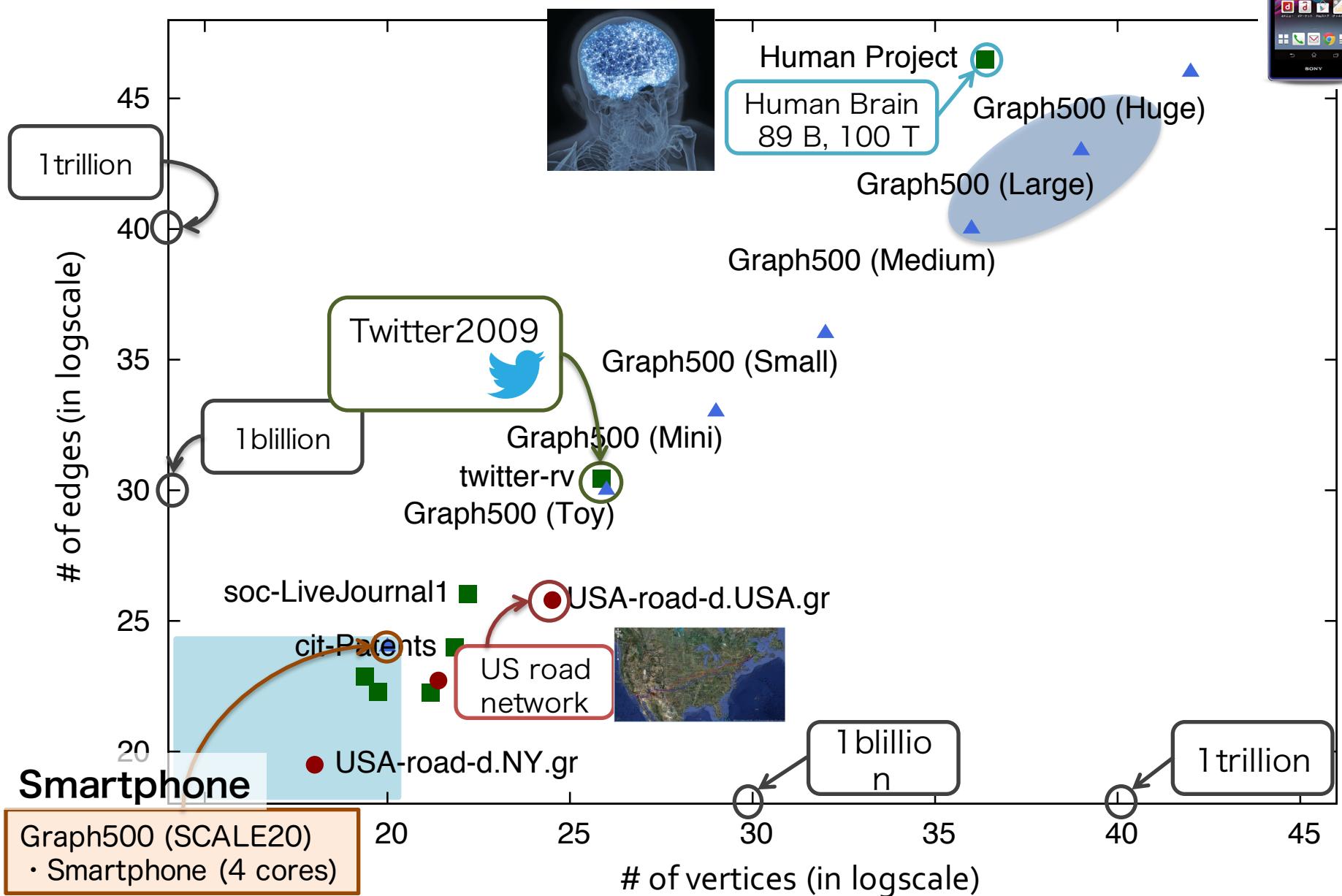
- 平成18年4月に国家基幹技術として、プロジェクトを開始。富士通と理化学研究所が共同開発。
- 平成23年6月、11月と連続で世界スパコン性能ランキング(TOP500)において1位を獲得。
- 平成24年9月28日に供用開始。
- 分野で最高の賞である「ゴードン・ベル賞」を2年連続(平成23、24年度)で受賞。
- 実用に近い総合的な性能を評価する「HPCチャレンジ賞」を4年連続(平成23～26年度)で受賞。
- 平成26年6月にビックデータの解析性能を評価するランキング(Graph500)において1位を獲得。
- 平成26年7月に「京」のネットワーク技術が(公財)発明協会から最も優れた発明として恩賜発明賞を受賞。
- プロジェクト経費:約1,110億円(平成18年度～平成24年度)



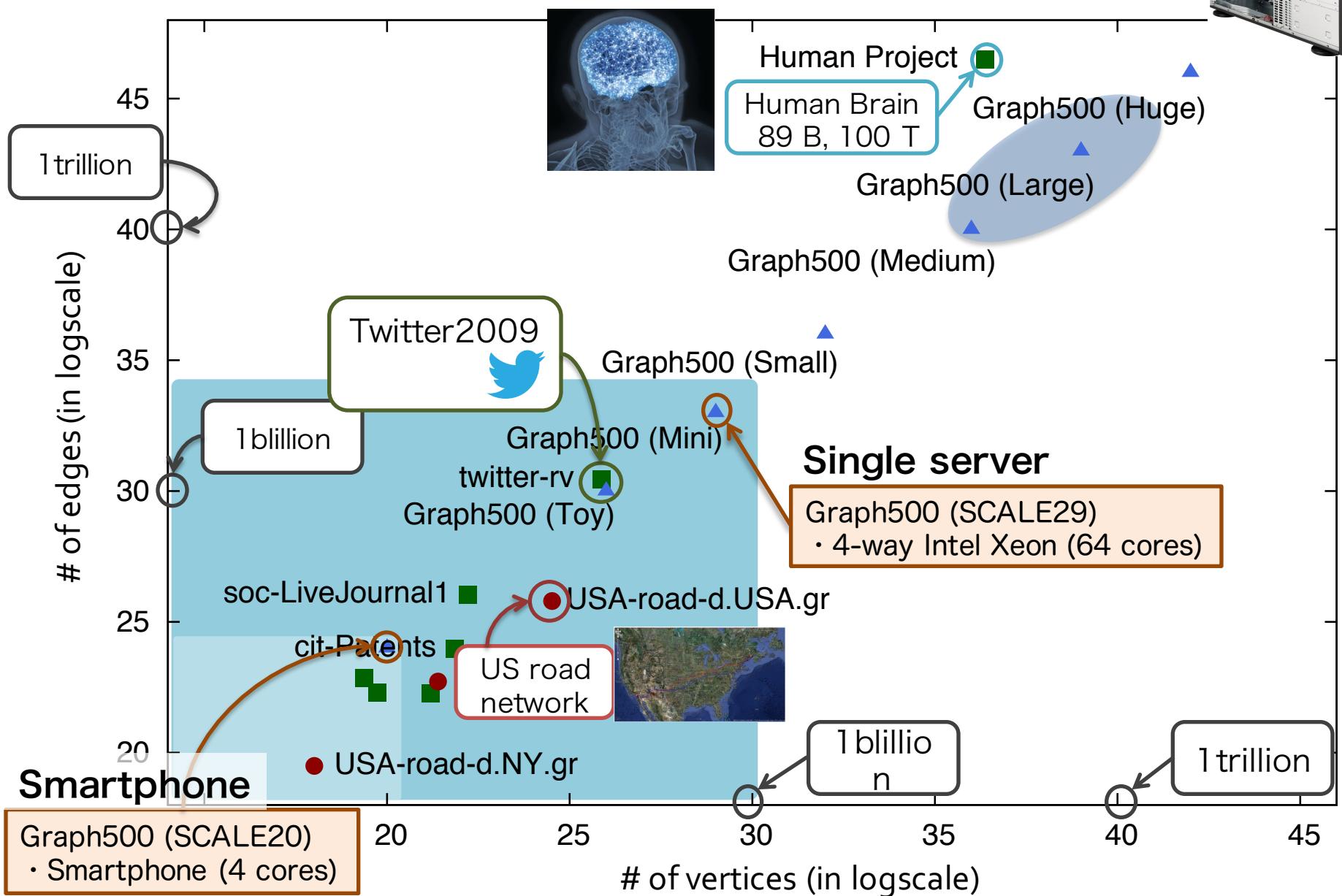
Target networks



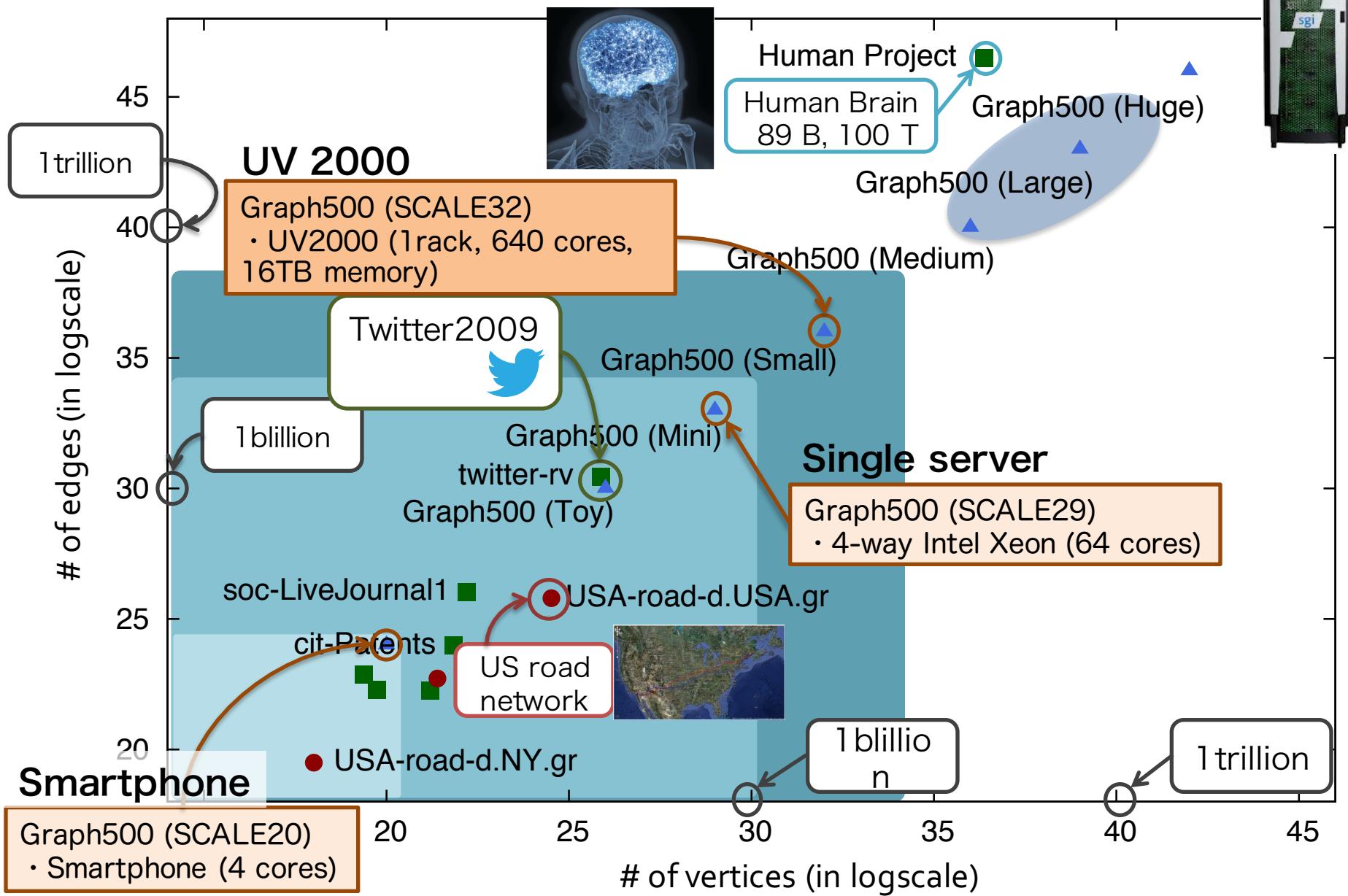
Target networks on Smartphone



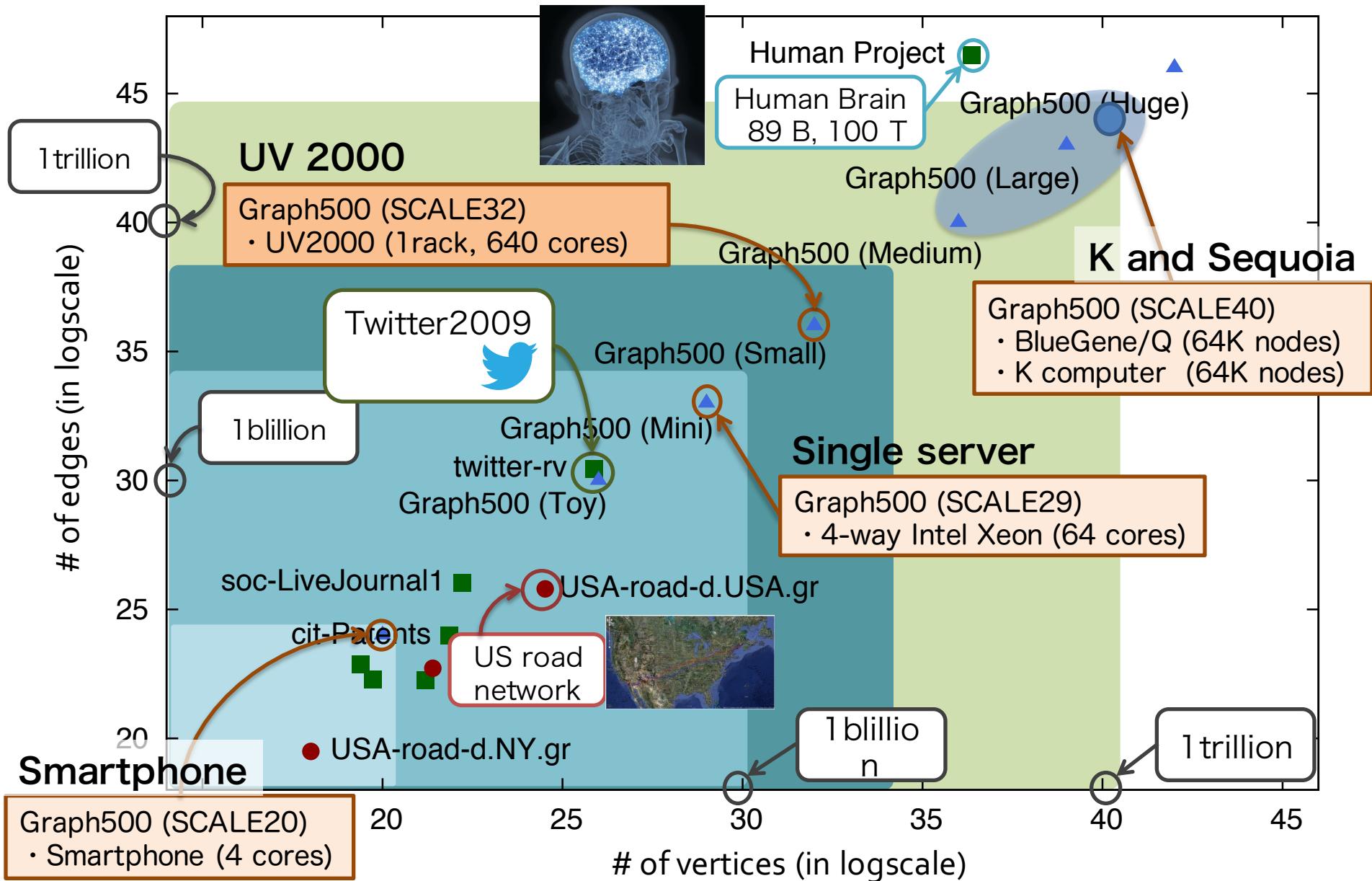
Target networks on Single-server



Target networks on UV2000

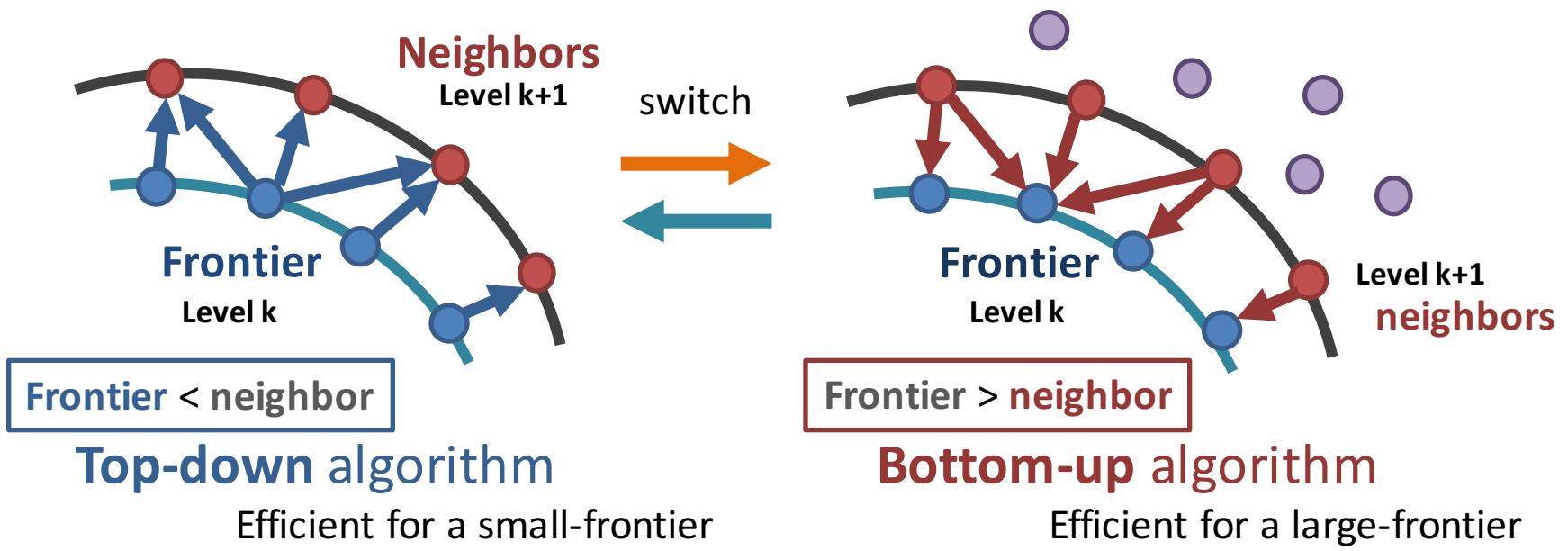


Target networks on Supercomputer



Hybrid BFS for low-diameter graph

- Efficient for Low-diameter graph [Beamer2011, 2012]
 - scale-free and/or small-world property such as social network.
- At higher ranks in Graph500 benchmark
- Hybrid algorithm
 - combines top-down algorithm and bottom-up algorithm
 - reduces unnecessary edge traversal



Direction-optimizing BFS

Chooses direction from Top-down or Bottom-up
for small frontier for large frontier

Beamer @ SC12

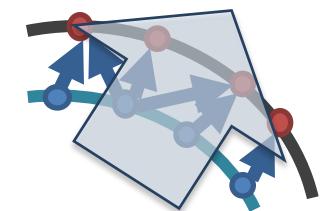
of traversal edges of Kronecker graph with SCALE 26

$$|V| = 2^{26}, |E| = 2^{30}$$

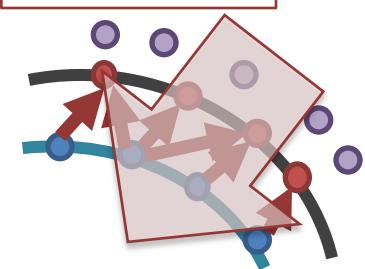
Level	Distance from source		
	Top-down	Bottom-up	Hybrid
Large frontier	2	2,103,840,895	2
	66,206	1,766,587,029	66,206
	346,918,235	52,677,691	52,677,691
	1,727,195,615	12,820,854	12,820,854
	29,557,400	103,184	103,184
	82,357	21,467	21,467
	221	21,240	227
Total	2,103,820,036	3,936,072,360	65,689,631
Ratio	100.00%	187.09%	3.12%

= $|E|$

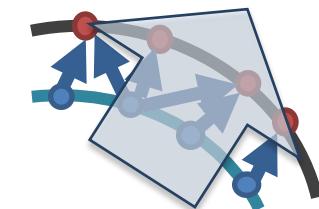
Top-down



Bottom-up



Top-down



Hybrid-BFS reduces
unnecessary edge traversals

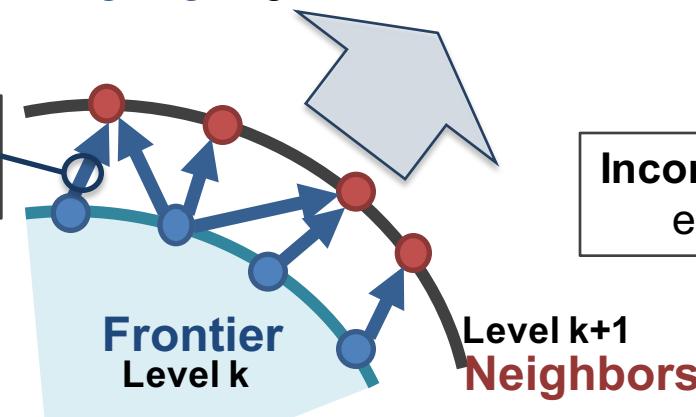
Direction-optimizing BFS

Chooses **direction** from *Top-down* or *Bottom-up*

Beamer2012 @ SC2012

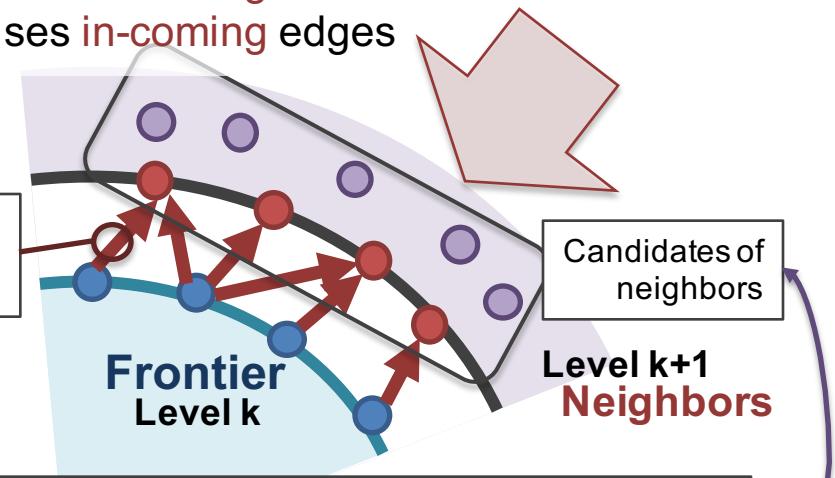
Top-down direction

- Efficient for **small-frontier**
- Uses **out-going edges**



Bottom-up direction

- Efficient for **large-frontier**
- Uses **in-coming edges**



Input : Directed graph $G = (V, A^F)$, Queue Q^F
Data : Queue Q^N , visited, Tree $\pi(v)$

```

 $Q^N \leftarrow \emptyset$ 
for  $v \in Q^F$  in parallel do
  for  $w \in A^F(v)$  do
    if  $w \notin \text{visited}$  atomic then
       $\pi(w) \leftarrow v$ 
       $\text{visited} \leftarrow \text{visited} \cup \{w\}$ 
       $Q^N \leftarrow Q^N \cup \{w\}$ 
   $Q^F \leftarrow Q^N$ 

```

Current frontier

Unvisited neighbors

Input : Directed graph $G = (V, A^B)$, Queue Q^F
Data : Queue Q^N , visited, Tree $\pi(v)$

```

 $Q^N \leftarrow \emptyset$ 
for  $w \in V \setminus \text{visited}$  in parallel do
  for  $v \in A^B(w)$  do
    if  $v \in Q^F$  then
       $\pi(w) \leftarrow v$ 
       $\text{visited} \leftarrow \text{visited} \cup \{w\}$ 
       $Q^N \leftarrow Q^N \cup \{w\}$ 
    break
   $Q^F \leftarrow Q^N$ 

```

Candidates of neighbors

Current frontier

Skips unnecessary edge traversal

NUMA architecture

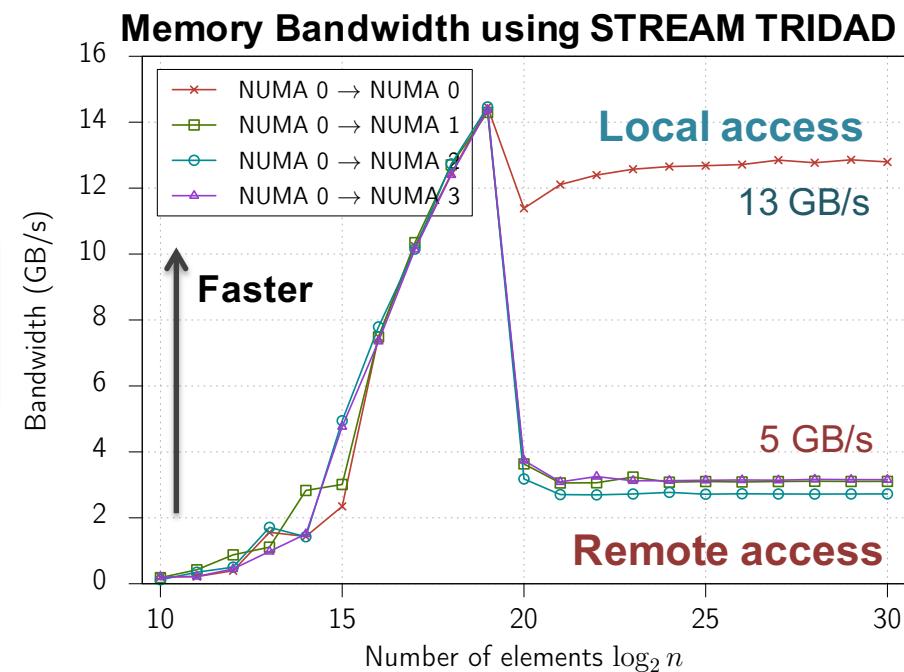
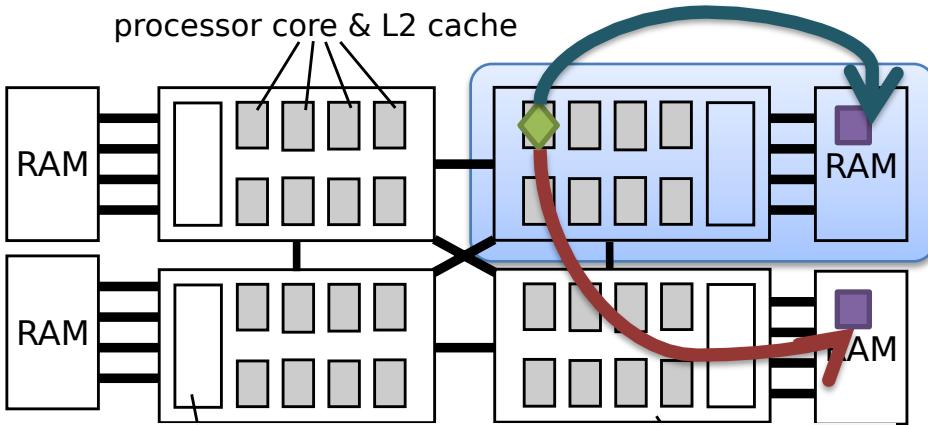
- 4-way Intel Xeon E5-4640 (Sandybridge-EP)

- 4 (# of CPU sockets)
- 8 (# of physical cores per socket)
- 2 (# of threads per core)

Max.
 $4 \times 8 \times 2 = 64$ threads

NUMA node

Memory access for Local RAM (Fast)

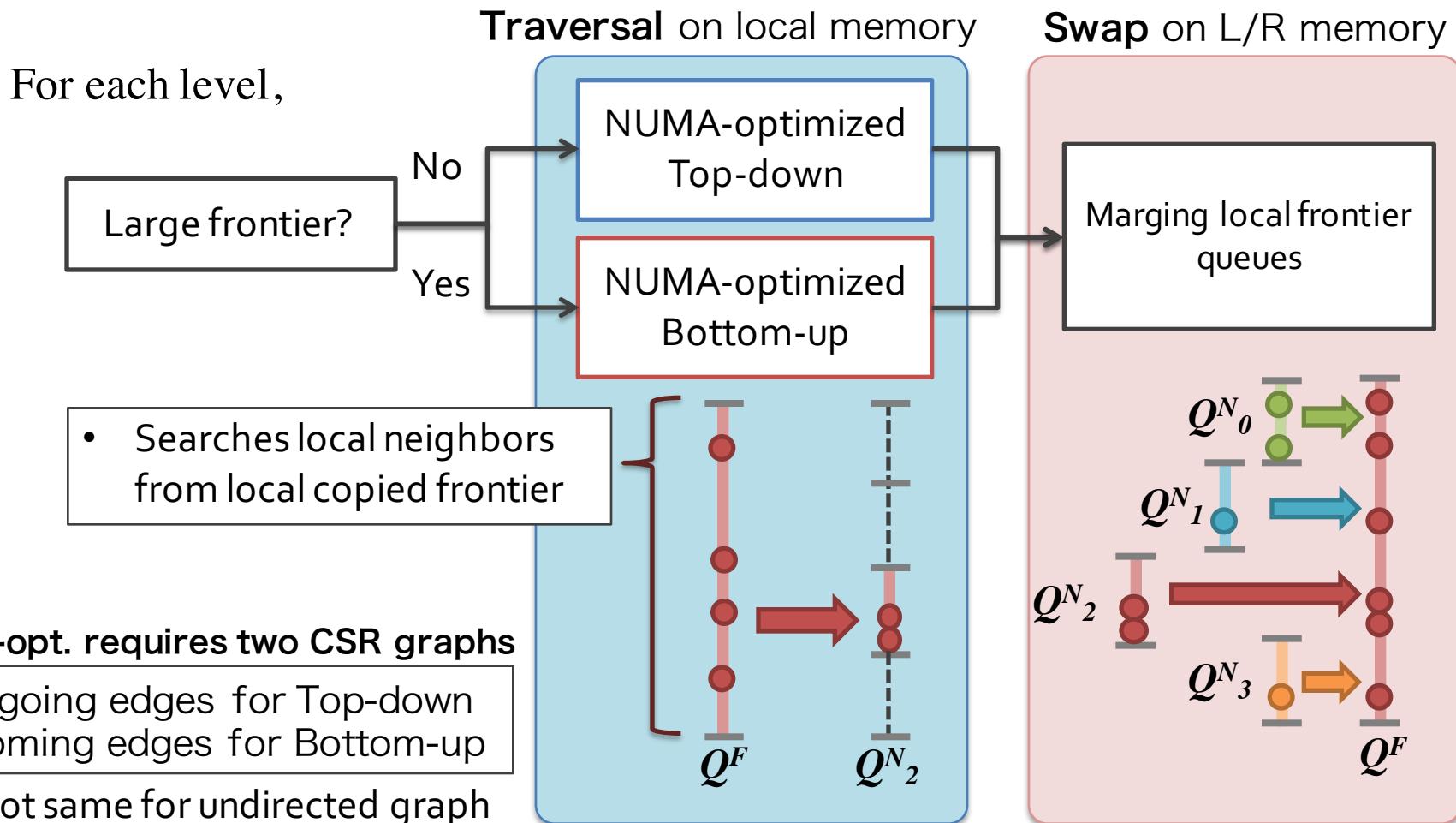


NUMA-aware (optimized) computation

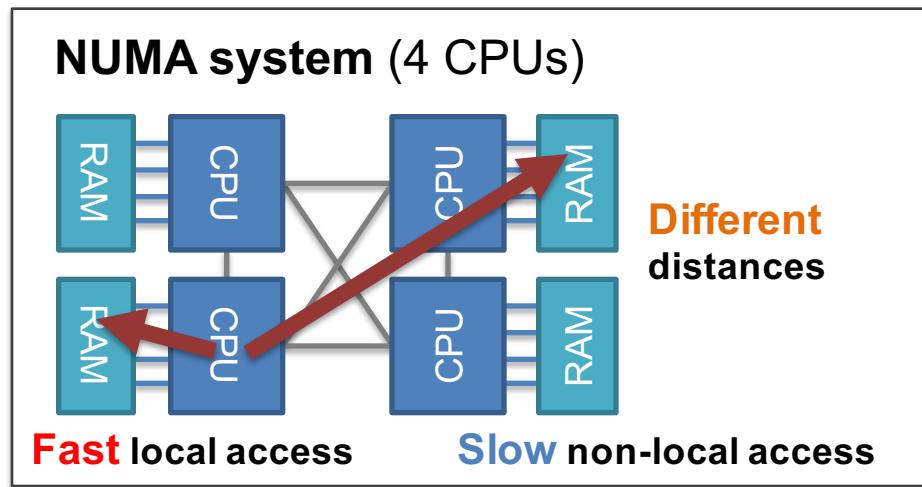
- Reduces and avoids memory accesses for **Remote RAM**

NUMA-optimized BFS

- Clearly separated to accessing for local and remote memory
 - Edge traversal on Local RAM
 - All-gather of local queues and bitmaps for Remote RAM

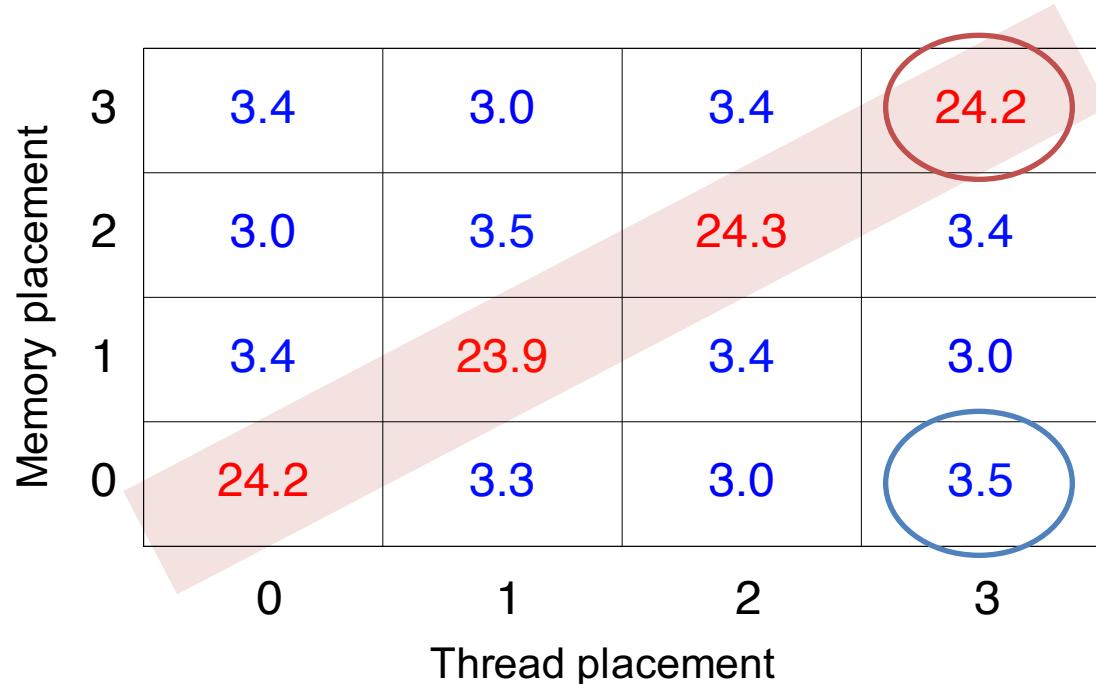


NUMA (Non-uniform memory access) system

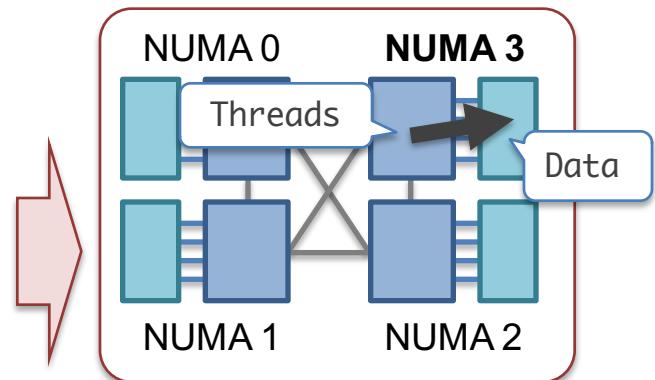


(Example) 4-socket Xeon system

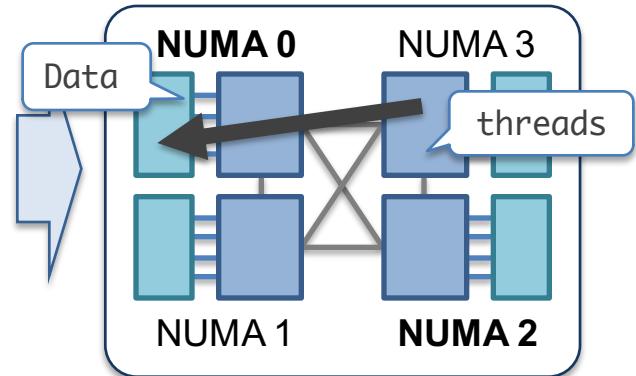
- 4 (# of CPU sockets)
- 8 (# of physical cores per socket)
- 2 (# of threads per core)



Local access: 24 GB/s



Remote access: 3 GB/s

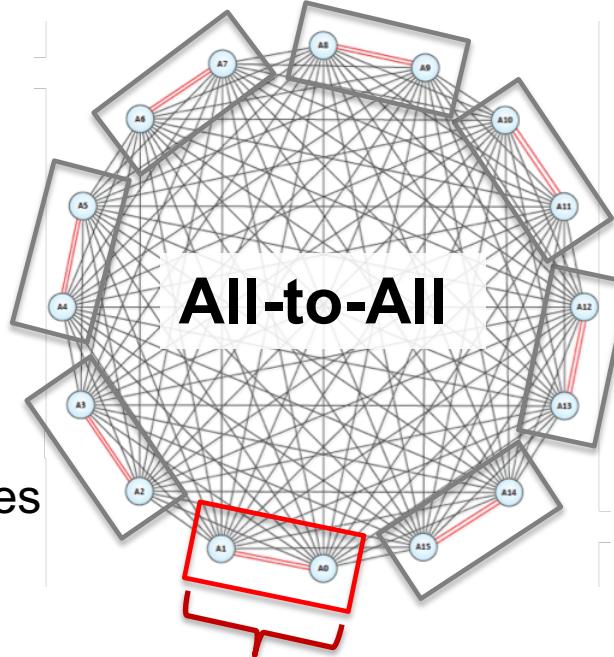


SGI UV 300 Specification

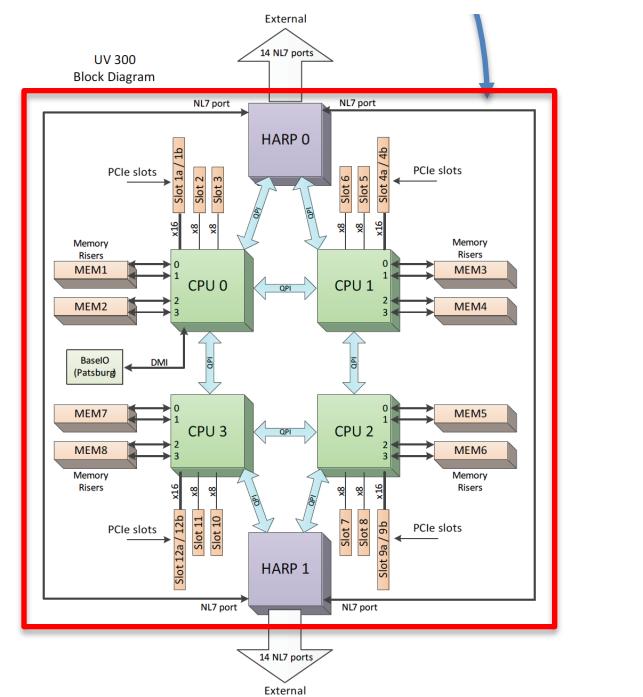
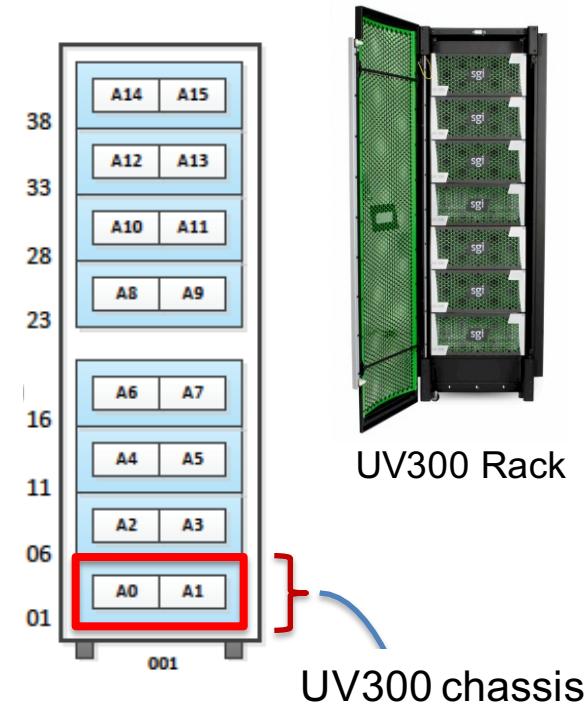
- UV300 one rack
 - All-to-all communication
 - SUSE Linux Enterprise Server 11 (x86_64)
 - Up to **1152 threads** (= 32 sockets x 18 x 2)
 - **16 TB RAM**

UV300 rack
= 8 UV300 chassis

Each chassis has
two HARP interfaces



- UV300 chassis
 - 144 threads (4 sockets x 18 cores x 2 SMT)
 - Intel Xeon E7-8867 v3 2.50GHz (Haswell)
 - 2TB RAM (512 GB per NUMA node)



Bandwidths on UV300 (HT, THP, Local-mode)

Measured by STREAM Benchmark TRIAD operation

1. NUMA local

– **56 GB/s**

2. NUMA remote

– **12–14 GB/s**

3. NUMA Link7

– **6 GB/s**

12–14 GB/s — **56 GB/s**



Node 1 (4-socket)

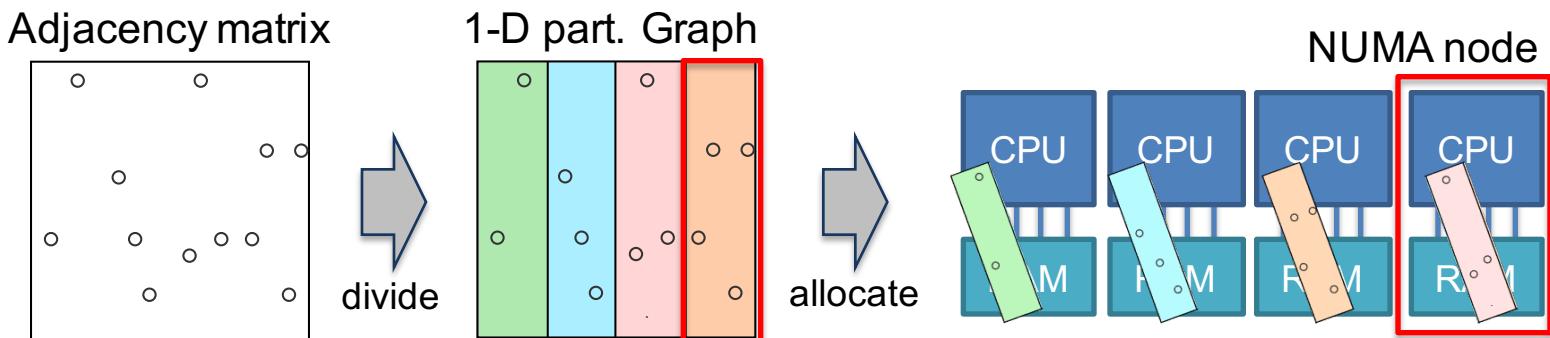
Memory placement

31	5.9	5.8	5.8	5.9	5.8	5.8	5.9	5.8	5.8	5.9	5.9	6.0	5.9	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	6.0	5.9	5.9	6.0	6.0	13.9	12.3	14.6	56.5			
30	5.8	5.8	5.9	5.9	5.8	5.8	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	6.0	6.0	6.0	12.5	14.5	56.5	14.1				
29	5.8	5.8	5.7	5.8	5.8	5.8	5.8	5.8	5.9	5.9	5.9	5.8	5.9	5.8	5.8	5.9	5.9	5.9	5.9	5.8	5.9	5.9	5.9	5.8	5.9	14.5	56.3	13.7	12.1				
28	5.9	5.8	5.8	5.9	5.9	5.9	5.8	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	6.0	6.0	5.9	6.0	6.0	6.0	56.4	14.1	12.5	13.8				
27	5.8	5.8	5.8	5.9	5.8	5.8	5.9	5.8	5.8	5.9	5.9	5.8	5.8	5.9	5.9	5.8	5.9	5.9	5.8	5.8	5.9	5.9	5.9	5.9	5.9	13.8	12.3	14.6	56.5				
26	5.8	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	6.0	6.0	5.9	5.9	6.0	12.5	14.5	56.6	14.1				
25	5.8	5.8	5.8	5.8	5.8	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	14.5	56.5	13.8	12.1				
24	5.8	5.9	5.8	5.9	5.9	5.8	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	56.5	14.1	12.6	13.9				
23	5.8	5.8	5.9	6.0	5.8	5.8	5.9	5.9	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	13.8	12.3	14.6	56.5				
22	5.9	5.9	5.9	5.9	6.0	5.9	5.9	6.0	5.9	5.9	6.0	5.9	5.9	6.0	6.1	5.9	5.9	6.0	12.6	14.6	56.7	14.1	5.9	5.9	5.9	6.0	5.9	5.9	6.0	6.0			
21	5.9	5.9	5.8	5.9	5.9	5.9	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	14.5	56.3	13.8	12.1				
20	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	6.0	5.9	6.0	5.9	6.0	5.9	6.0	5.9	6.0	5.9	6.0	5.9	6.0	6.0	6.0	56.5	14.1	12.6	13.9				
19	5.8	5.8	5.9	5.8	5.8	5.8	5.9	5.8	5.9	6.0	5.9	5.8	5.9	5.9	13.9	12.4	14.6	56.6	5.9	5.9	5.9	5.9	5.8	5.8	5.9	5.8	5.8	5.8	5.9	5.9	5.9	5.9	
18	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.9	6.0	6.0	5.9	5.9	6.0	12.5	14.6	56.6	14.1	5.9	6.0	6.0	5.9	5.9	6.0	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9
17	5.8	5.9	5.8	5.9	5.8	5.8	5.9	5.9	5.8	5.9	5.9	5.9	5.9	5.9	14.6	56.4	13.8	12.2	5.9	5.9	5.8	5.9	5.9	5.9	5.8	5.8	5.8	5.9	5.8	5.9	5.9	5.9	5.9
16	5.9	6.0	5.9	6.0	5.9	5.9	5.8	5.9	6.0	6.0	5.9	6.0	5.9	6.0	12.5	14.2	12.6	13.9	6.0	6.0	5.9	6.0	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9
15	5.8	5.8	5.8	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	13.8	12.3	14.6	56.6	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.8	5.8	5.8	6.0	5.8	5.8	5.8	5.9	5.9	
14	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	6.0	12.5	14.5	56.7	14.1	6.0	5.9	6.0	6.1	5.9	6.0	6.0	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.9	
13	5.8	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	14.5	56.4	13.7	12.1	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.8	5.8	5.9	5.8	5.8	5.9	5.9	5.9	5.9	
12	5.9	5.9	5.9	6.0	5.9	5.9	6.0	5.9	5.9	5.9	6.0	12.5	14.1	12.6	13.8	5.9	5.9	5.9	6.0	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	6.0	
11	5.8	5.9	6.0	5.8	5.8	5.9	6.0	13.9	12.3	14.6	56.8	5.9	5.8	5.9	5.8	5.9	5.8	5.9	5.8	5.9	5.9	5.9	5.9	5.8	5.8	5.9	5.8	5.8	5.9	5.9	5.9	5.9	
10	5.9	6.0	6.0	5.9	5.9	6.0	6.0	12.5	14.5	56.9	14.1	5.9	6.0	6.0	6.1	5.9	5.9	6.0	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.8	5.9	5.8	5.9	5.9	5.9	5.9	5.9
9	5.9	5.9	5.8	5.9	5.8	5.9	5.8	14.5	56.5	13.7	12.1	5.9	5.9	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.8	5.9	5.8	5.9	5.9	5.9	5.9	5.9	
8	6.0	6.0	5.9	6.0	6.0	6.0	6.0	13.8	12.3	14.6	56.6	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
7	5.8	5.9	5.9	6.0	13.8	12.3	14.6	56.6	5.8	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.8	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
6	5.9	5.9	6.0	6.1	12.5	14.5	56.5	14.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.8	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
5	5.9	5.9	5.9	6.0	14.5	56.4	13.7	12.1	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
4	5.9	5.9	5.9	6.0	14.5	56.4	13.7	12.1	5.9	5.9	5.9	6.0	6.0	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
3	13.8	12.3	14.6	56.4	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.8	5.9	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.8	5.9	5.8	5.9	5.9	
2	12.5	14.4	56.5	14.1	5.9	5.9	6.0	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	
1	14.5	51.0	13.7	12.1	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.9	5.9	5.8	5.9	5.9	5.8	5.9	5.9	5.8	5.7	5.8	5.8	5.7	5.8	5.8	5.7	5.8	5.9	
0	56.6	14.1	12.6	13.9	6.0	6.0	5.9	6.0	6.0	6.0	5.9	6.0	5.9	5.9	6.0	5.9	5.9	6.0	5.9	5.9	5.9	6.0	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	

Thread placement

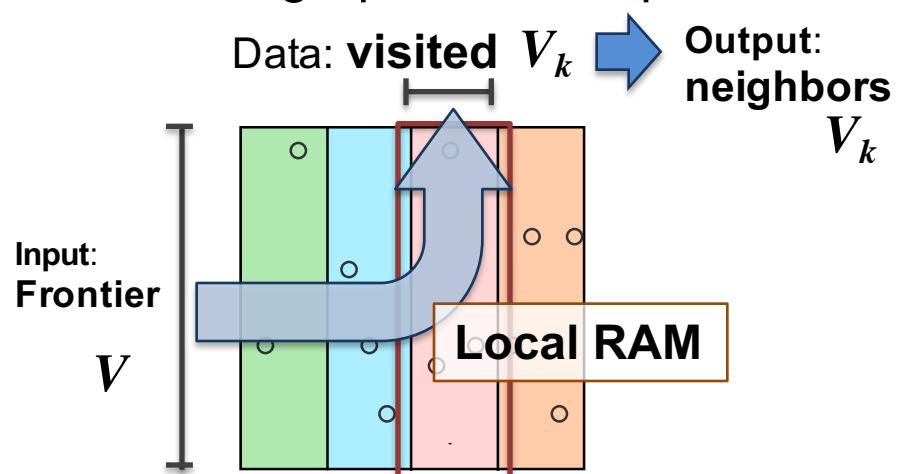
Ours: NUMA-aware 1-D partitioned graph rep.

- Divides sub-graphs and allocates on each NUMA node

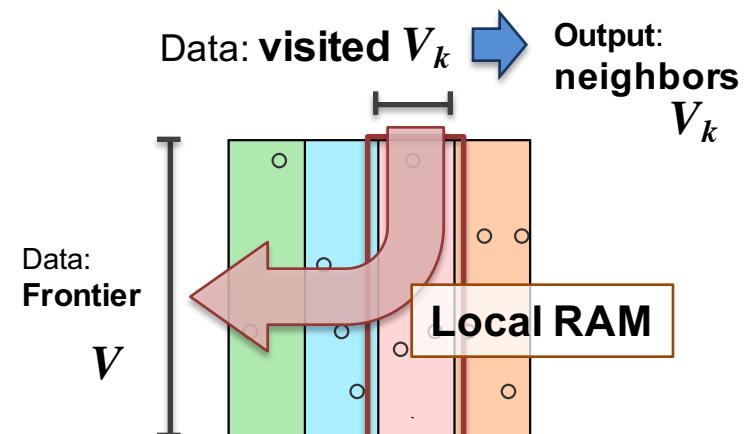


- Avoids remote memory access at traversal phase

- Forward graph G^F for Top-down



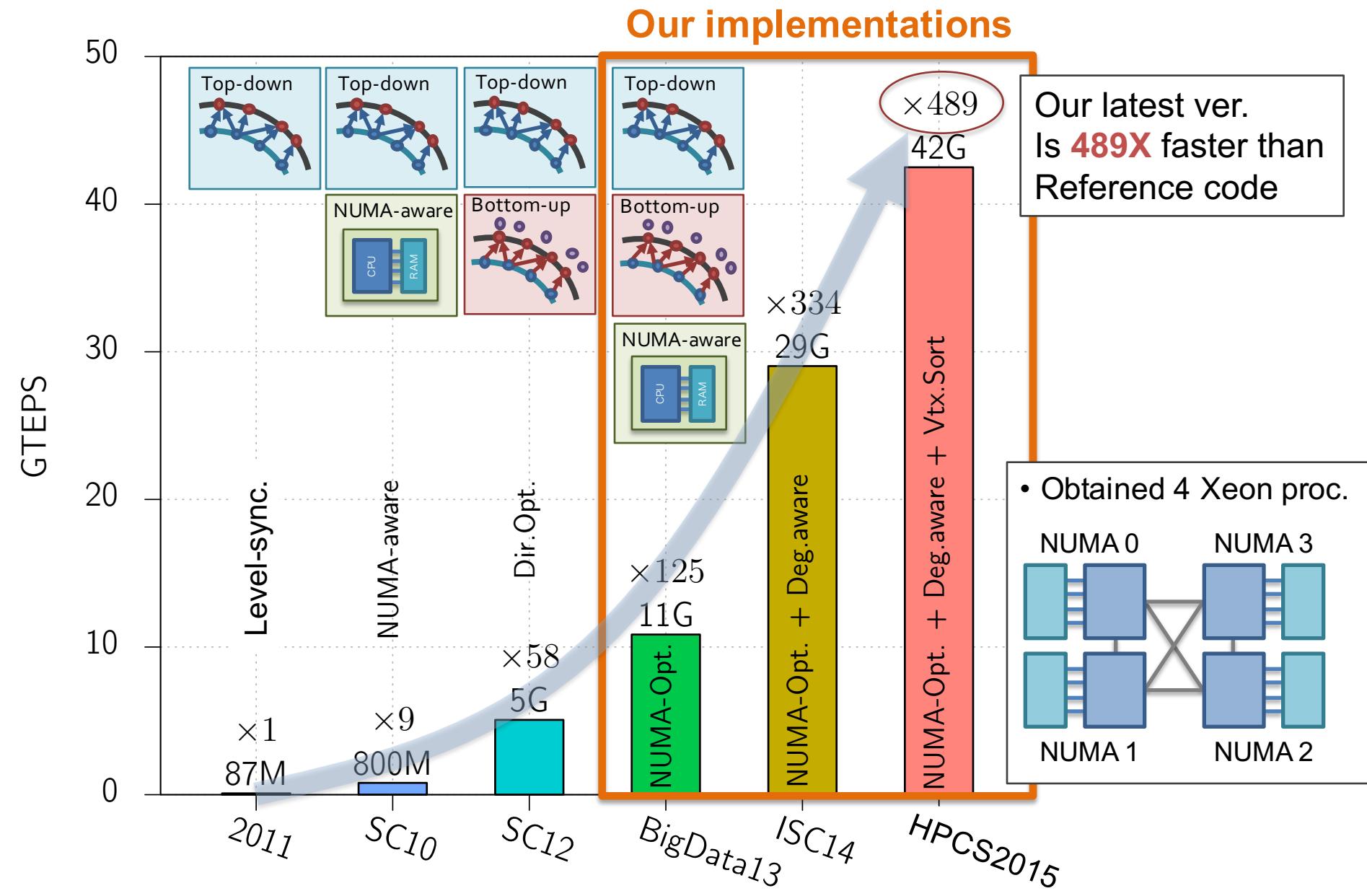
- Backward graph G^B for Bottom-up



Using subset
of vertex

$$V_k = \left\{ v_j \in V \right\} \quad j \in \left[\frac{k}{\ell} \cdot n, \frac{(k+1)}{\ell} \cdot n \right)$$

Ours: NUMA-optimized + Dir. Opt. BFS



Our implementations for Graph500 and Green Graph500

- Multiple nodes and multiple processes
 - Multiple CPUs (**K computer**, TSUBAME 2.0, Fujitsu FX10, TSUBAME KFC)
 - **38621.4GTEPS**, 1st position in 10th Graph500 list
 - Multiple CPUs and GPUs (**TSUBAME 2.0** & **TSUBAME KFC**)
 - **317GTEPS**, 4th position in 4th Graph500 list
 - **6.72GTEPS/kW**, 1st position in 2nd Green Graph500 list (Big)
- Single node and single process
 - NUMA optimized and multiple threads (**SGI Altix UV2000**, 4-way Intel Xeon server)
 - **174.7GTEPS**, 40th position in 10th Graph500 list (world fastest implementation for single node)
 - Multiple threads and power efficient (Android tablet, Android based smart phone)
 - **153.2 GTEPS/kW**, 1st position in 2nd Green Graph500 list (Small)

2D Hybrid BFS [Beamer, '13]

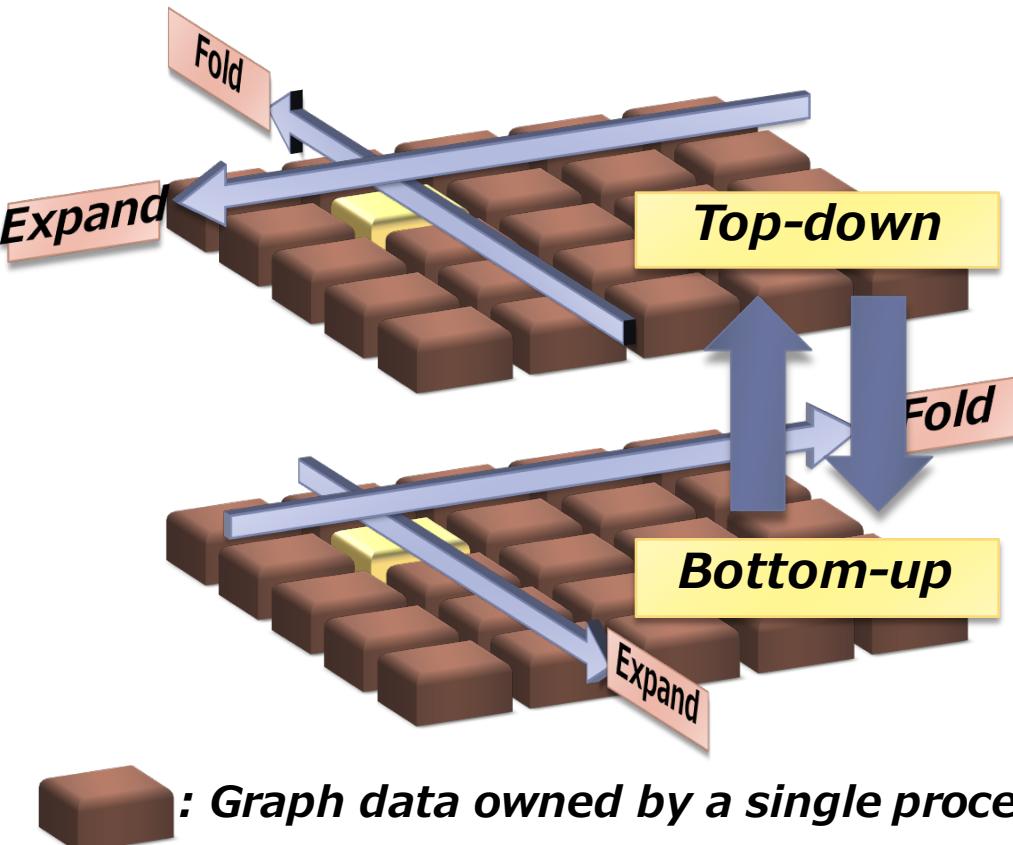
- ▶ **2D partitioning of the adjacency matrix for the graph**

$$A = \left(\begin{array}{c|c|c} A_{1,1} & \cdots & A_{1,C} \\ \hline \vdots & \ddots & \vdots \\ \hline A_{R,1} & \cdots & A_{R,C} \end{array} \right)$$

- ▶ **Each partitioned region is assigned to a processor places on the 2D mesh.**
 - ▶ **Locality is better in 2D partitioning**
 - ▶ **1D partitioning requires all-to-all communication among all the processes.**
- ▶ **Our implementation is based on this method.**

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

Efficient Hybrid Search with 2D Partitioning



*Sharing the same graph data
between two direction search.*

- ▶ Our approach is based on the Top-down and bottom-up hybrid search BFS.
[Beamer2011, 2012]
- ▶ We realized the hybrid search without any increase of memory footprint.
 - ▶ Graph data is shared between two search directions.
- ▶ Overlapped communication with computation.
 - ▶ Both top-down and bottom-up utilize overlapping communication.

The Graph500

K Computer and TSUBAME 2.0 & 2.5

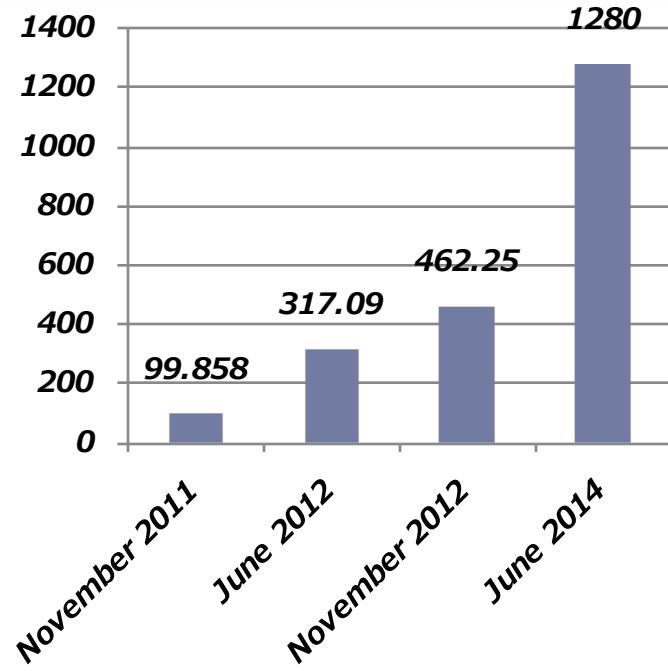
Graph500 ranking history for TSUBAME2.0 and 2.5

List	Rank	GTEPS	Implementation
November 2011	4	99.858	Top-down only
June 2012	4	317.09	CPU+ GPU
November 2012	20	462.25	CPU + GPU
July 2015	12	1345	<u>Efficient hybrid</u>

*Every score is obtained using TSUBAME2.0 1366 nodes or TSUBAME 2.5 1024 nodes

Graph500 ranking history for K Computer

BFS performance on TSUBAME2.0 and 2.5



List	Rank	GTEPS	Implementation
November 2013	4	5524.12	Top-down only
June 2014	1	17977.05	Efficient hybrid
July, November 2015	1	38632.4	<u>Efficient hybrid & Zero-degree suppression & Vertex sorting</u>

スパコン「京」がグラフ解析性能を競うGraph500で世界1位を奪還

2015年7月13日(日本時間14日)

Safari ファイル 編集 表示 履歴 ブックマーク ウィンドウ ヘルプ 5.33 GB Free 100% 月 17:44 藤澤克樹

news.mynavi.jp 急上昇ワード [武井咲 マスク wifi 事業承継 長]

マイナビ ニュース テクノロジー

半導体デバイス 次世代半導体技術 カーエレクトロニクス 産業機器/ロボット 組み込み/IoT スパコン/HPC 計測機器
エネルギー 医療/バイオ サイエンス ものづくり セイコーライ・インフォテック ビジネス会員

記事種別 特集 レポート レビュー ハウツー インタビュー 連載 コラム

ニューストップ > テクノロジー > スパコン/HPC

スパコン「京」がグラフ解析性能を競うGraph500で世界1位を奪還

[2015/07/14] [PR]

今人気の動画もマイナビニュースで楽しもう☆動画コーナーはここをクリック！
13.3型 (2560x1440) + 500GB SSD搭載ノートPCがこの価格？
マイナビニュースでお買い物もできちゃう！！ショッピングチャンネルはこちらから♪
OSSデータベースのセキュリティ対策に応えた暗号化ソリューション【MyDiamo】

理化学研究所、東京工業大学、九州大学、富士通、アイルランドのユニバーシティ・カレッジ・ダブリンらの国際研究グループは7月14日、ビッグデータ処理に関するスーパーコンピューターの国際的な性能ランキングであるGraph500において、スーパーコンピューター「京」が2014年6月以来となる1位を獲得したと発表した。



How Successful People Start Their Day
Whether you're a morning person or a night owl, we all seem to start our day at some point. And we all seem to start it differently. Some of us hop online ...

特別企画 [PR]

日本が経済成長する上の障壁

知らないと損するITセミナー



スパコン「京」がグラフ解析性能を競うGraph500で連覇(世界1位) 2015年11月17日(日本時間18日)

トピックス

← 前の記事

↑ 一覧へ戻る

→ 次の記事

2015年11月18日

f いいね！ 48

ツイート 55



理化学研究所

東京工業大学

ユニバーシティ・カレッジ・ダブリン

九州大学

富士通株式会社

科学技術振興機構

2期連続でスーパーコンピュータ「京」がGraph500で世界第1位を獲得 —ビッグデータの処理で重要なグラフ解析でも最高レベルの評価—

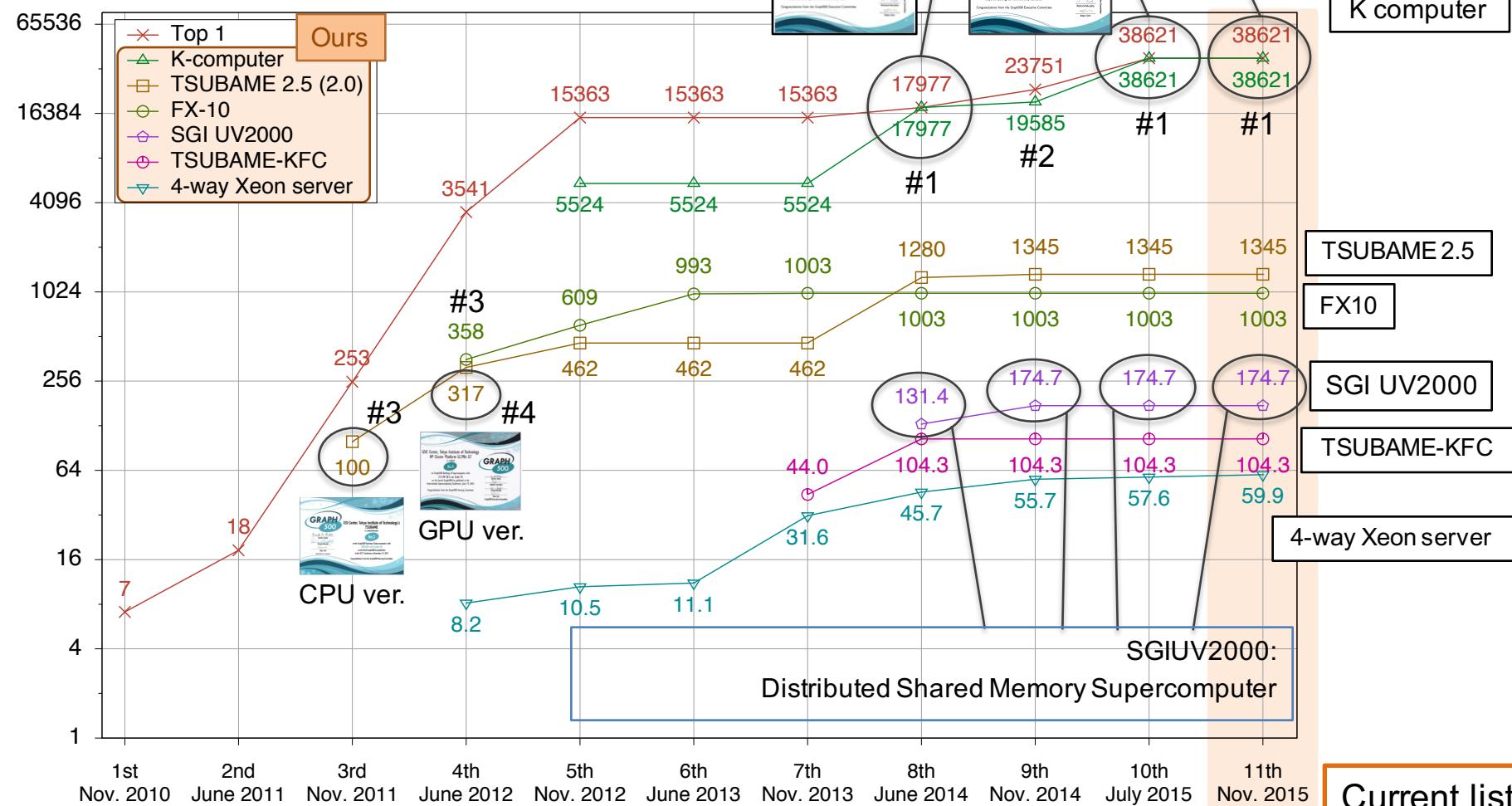
理化学研究所（理研）と東京工業大学、アイルランドのユニバーシティ・カレッジ・ダブリン、九州大学、富士通株式会社による国際共同研究グループは、ビッグデータ処理（大規模グラフ解析）に関するスーパーコンピュータの国際的な性能ランキングであるGraph500※において、[スーパーコンピュータ「京（けい）」](#)^[1]による解析結果で、2015年7月に続き第1位を獲得しました。これは、東京工業大学博士課程（理研研修生）上野晃司氏による成果です。

大規模グラフ解析の性能は、大規模かつ複雑なデータ処理が求められるビッグデータの解析において重要なもので、今回のランキング結果は、「京」がビッグデータ解析に関する高い能力を有することを実証するものです。

本研究の一部は、科学技術振興機構（JST）戦略的創造研究推進事業CREST「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」（研究総括：佐藤 三久 理研計算科学研究機構）における研究課題「ポストペタスケールシステムにおける超大規模グラフ最適化基盤」（研究代表者：藤澤 克樹 九州大学、拠点代表者：鈴村 豊太郎 ユニバーシティ・カレッジ・ダブリン）および「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」（研究総括：喜連川 優 国立情報学研究所）における研究課題「EBD：次世代の年ヨットバイト処理に向けたエクストリームビッグデータの基盤技術」（研究代表者：松岡 聰 東京工業大学）の一環として行われました。

Graph500 ベンチマークにおける 当研究チームの結果

The 8th, 10th, and 11th Graph500 において 京コンピュータが 1 位



The 10th and 11th Graph500 Lists : The winner is K computer

Scale 40 : 38621.4 GTEPS (July and November 2015)



RIKEN Advanced Institute for Computational
Science (AICS)'s K computer
is ranked
No.1

on the Graph500 Ranking of Supercomputers with
38621.4 GE/s on Scale 40

for High Performance Computing, Networking, Storage, and Analysis
(SC'15), November 16, 2015.

Congratulations from the Graph500 Executive Committee



David A. Bader

David A. Bader

Andrew Lumsdaine

Richard Murphy

Marc Snir

Graph500 Executive Committee



The 10th and 11th Graph500 Lists : The winner is K computer **Scale 40 : 38621.4 GTEPS** **(July and November 2015)**

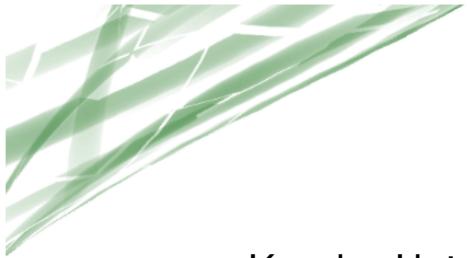


===== Result =====

SCALE:	40
edgefactor:	16
NBFS:	64
graph_generation:	197.379
num_mpi_processes:	82944
construction_time:	609.395
min_time:	0.395321105141
firstquartile_time:	0.409624118358
median_time:	0.455501377815
thirdquartile_time:	0.566996186739
max_time:	1.95167534612
mean_time:	0.562004323256
stddev_time:	0.311699826145
min_nedge:	1.7592103987e+13
firstquartile_nedge:	1.7592103987e+13
median_nedge:	1.7592103987e+13
thirdquartile_nedge:	1.7592103987e+13
max_nedge:	1.7592103987e+13
mean_nedge:	1.7592103987e+13
stddev_nedge:	0
min_TEPS:	9.01384752422e+12
firstquartile_TEPS:	3.10268470903e+13
median_TEPS:	3.86214067477e+13
thirdquartile_TEPS:	4.29469437914e+13
max_TEPS:	4.45007963348e+13
harmonic_mean_TEPS:	3.1302435335e+13
harmonic_stddev_TEPS:	2.18728188393e+12
min_validate:	43.201660905
firstquartile_validate:	43.4925568579
median_validate:	44.3293765394
thirdquartile_validate:	45.4055157886
max_validate:	50.040661654
mean_validate:	44.6539914012
stddev_validate:	1.39768976422

The 1st to 6th GreenGraph500 Lists : We keep the 1st place

Scale 30 : 62.93MTEPS/W (November 2015)



Kyushu University's
GraphCREST-SandybridgeEP-2.4GHz
is ranked

No. 1

in the **Big Data** category of the Green Graph 500
Ranking of Supercomputers with
62.93 MTEPS/W on Scale 30

on the sixth Green Graph 500 list published at the
Supercomputing Conference, November 17, 2015.

Congratulations from the Green Graph 500 Chair




Torsten Hoefler

GreenGraph500 Chair

グラフ解析と最適化の今後

九州大学
「センター・オブ・イノベーション(COI)プログラム」
共進化社会システム創成拠点



CESS

Center for Co-Evolutional Social System

共進化社会システム創成拠点



KYUSHU UNIVERSITY

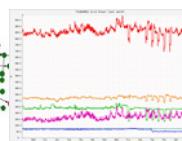
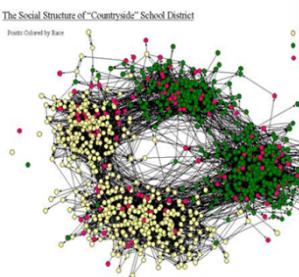


学内機関及び民間企業との連携による社会実装の必要性

1. 最先端理論(Algorithm Theory) + 大規模実データ(Big Data) + 最新計算技術(Computation)による超大規模問題の解決
2. 適用が期待される多くの分野
 - 防災計画策定、交通・災害復興・避難・ロジスティクス
 - エネルギー資源の確保や省エネルギーと最適供給(スマートグリッド)
 - 今後注目の分野：疫病の拡散、人口の増減、経済動向等の分析。生命科学系（創薬、遺伝子）。ビジネス系（金融、データマイニング）
3. 米国や欧州等では社会問題解決のビッグデータ処理が大規模プロジェクトとして開始
4. ビッグデータ解析、数理最適化、AI(人工知能)、HPC(高性能計算)及びサイバーセキュリティなどの最新技術の組合せや融合による社会実装(学内機関及び民間企業との連携)



超大規模センター
• 観測データ
• スマートグリッド
• 交通・輸送
• SNS (Twitter)



社会システムの事前解析、デザイン及び問題解決

交通、災害復興、避難、ロジスティクス、都市計画

エネルギー、省電力

物理、化学、生物などの科学

医学、バイオ
経済、経営、金融

ビッグデータ解析、数理最適化、AI(人工知能)、HPC(高性能計算)及びサイバーセキュリティ

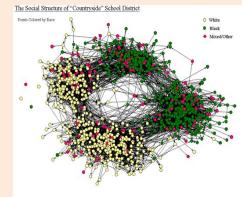
次世代スパコン(高密度演算装置+階層型データストア)

CPSとしての都市OS

実世界 実世界のデータ化



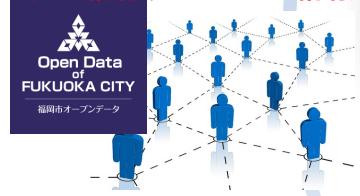
サイバー空間 最適化計算、 シミュレーション



実世界 実世界へ反映、制御



行政情報、ソーシャル情報



水素インフラセンシング



交通センシング



ソーシャルシステム ITS、管制、EMS



スマホ、PC
ナビゲーション

ウェアラブルデバイス パーソナルデバイス



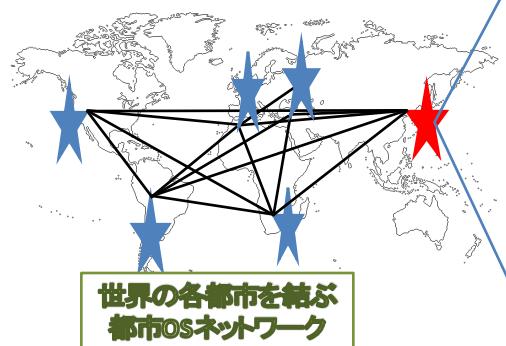
デジタルサイネージ



都市OSの機能

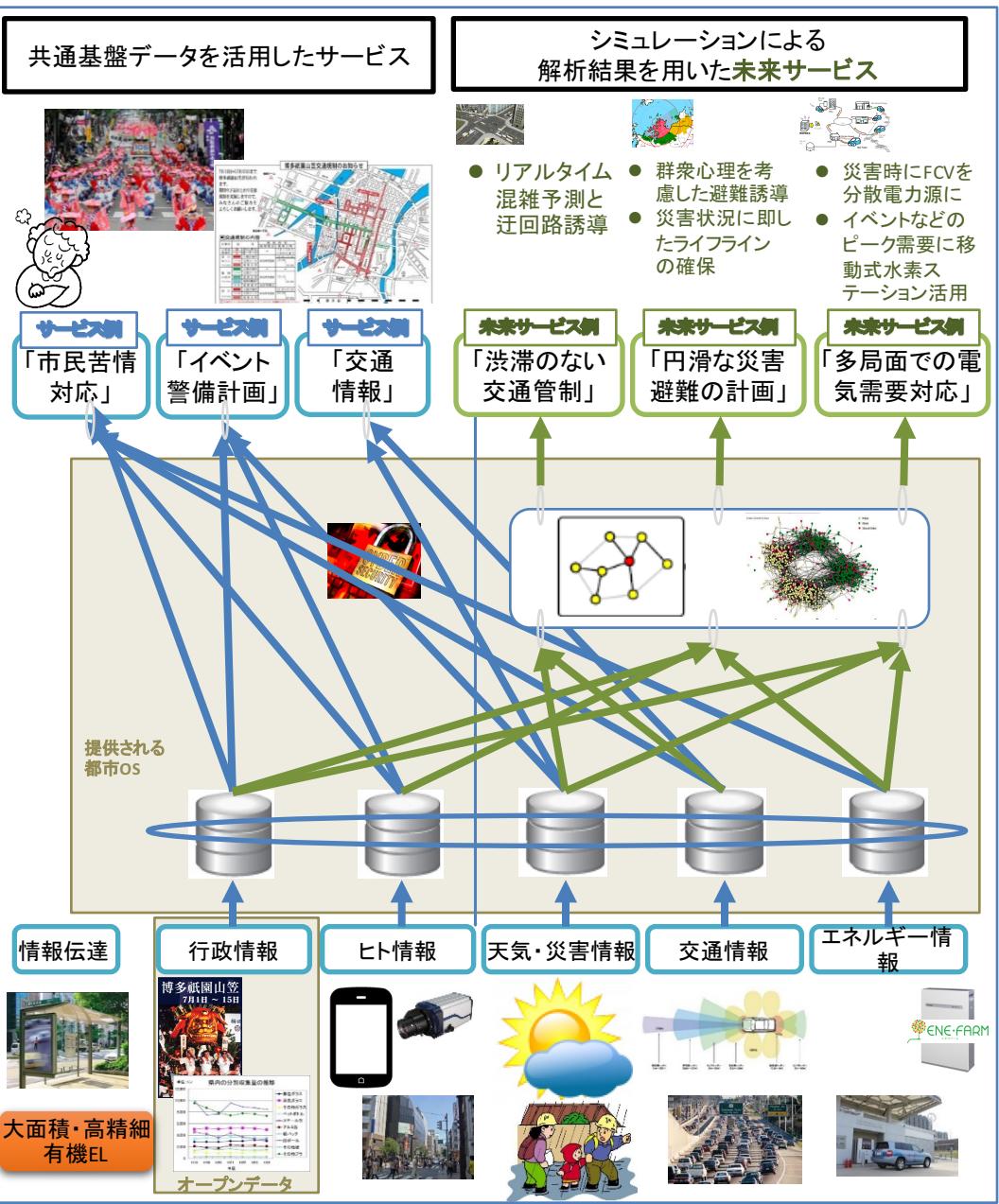
都市OSの展開

複数の都市への展開
複数の都市でデータ共有



■ 都市OSの機能

- 新しい都市サービスの共通基盤の提供
- 都市間のデータやサービス共有の仕組み
- 都市規模に応じたスケーラビリティの担保
- 災害時などの緊急時サービスの他都市での代替
- 新しいサービス・アプリケーションのための抽象化



アプリケーション・サービス

共通データを
様々な市民サービスへ
展開

- 新規民間サービス支援のしくみ
- エネルギー可視化システム
- 最適なHEMSと制度
- スマート＆マルチモーダル交通

起業 創業

最適化・分析

自動で最適化、制御
ボトルネック検出

- リアルタイムシミュレーション
- セキュリティ

データの格納

いろいろな情報を共通データ基盤に吸い上げ、
必要に応じて取り出し

- 都市OSコアソフトウェア
- データ格納リポジトリ

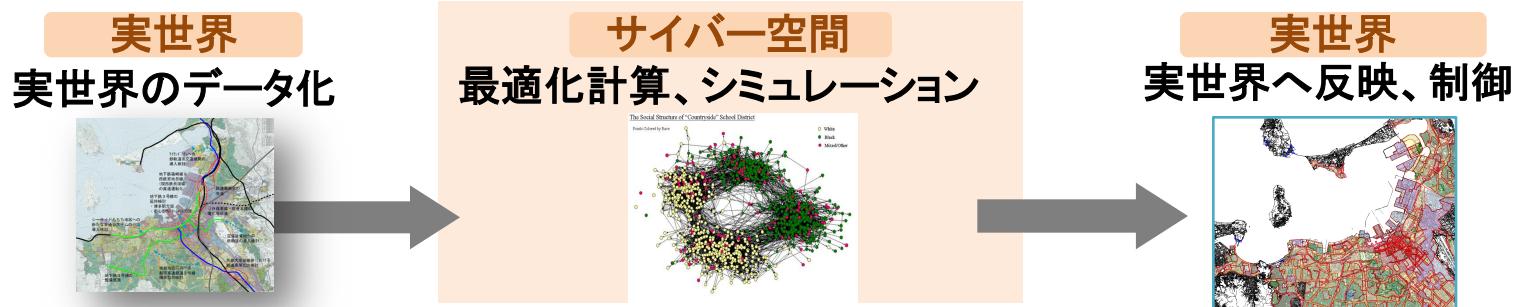
データ

オープンデータ
センサーネットワーク

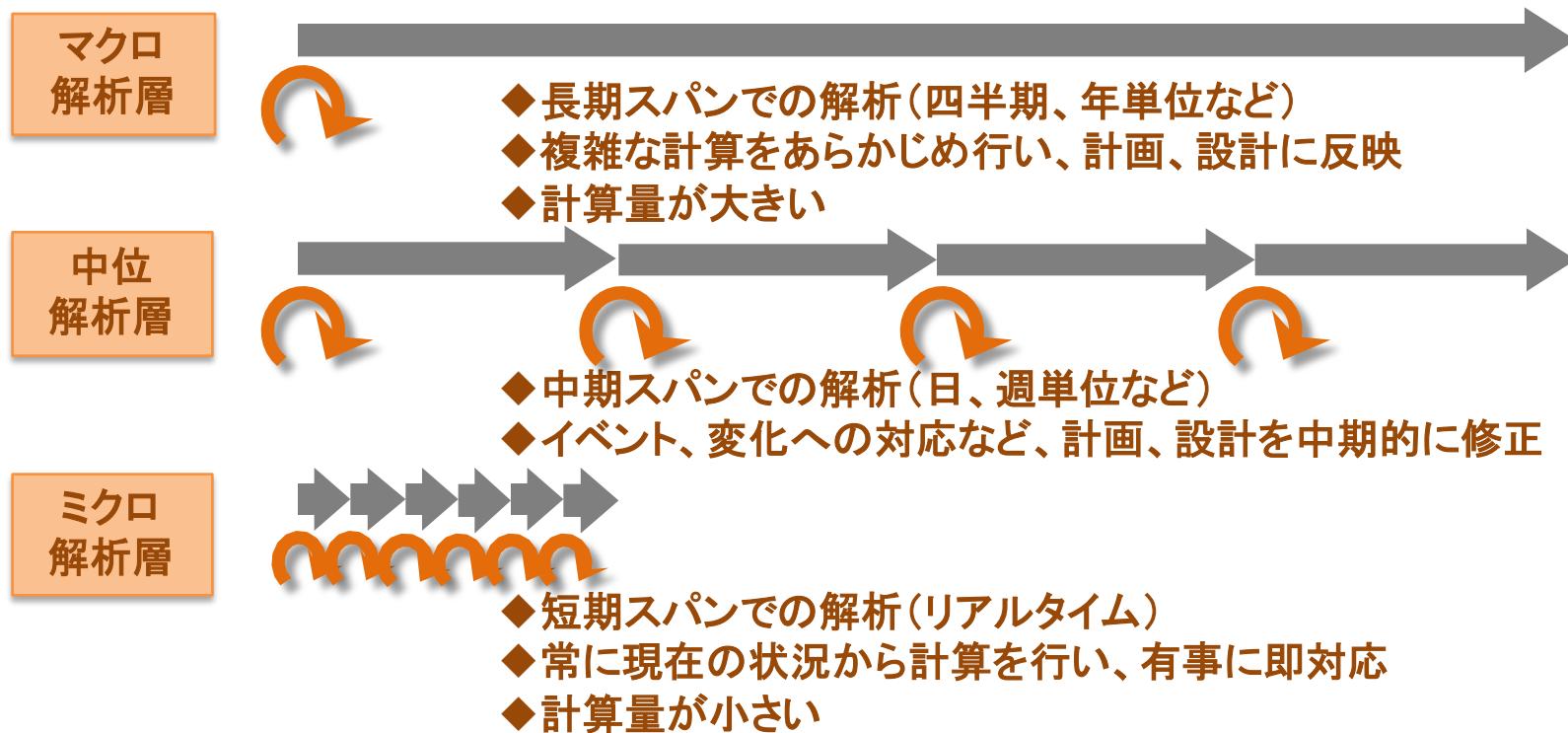
- 経済的・快適な燃料電池車
- 経済的・エコな燃料電池

都市OS解析層モデル

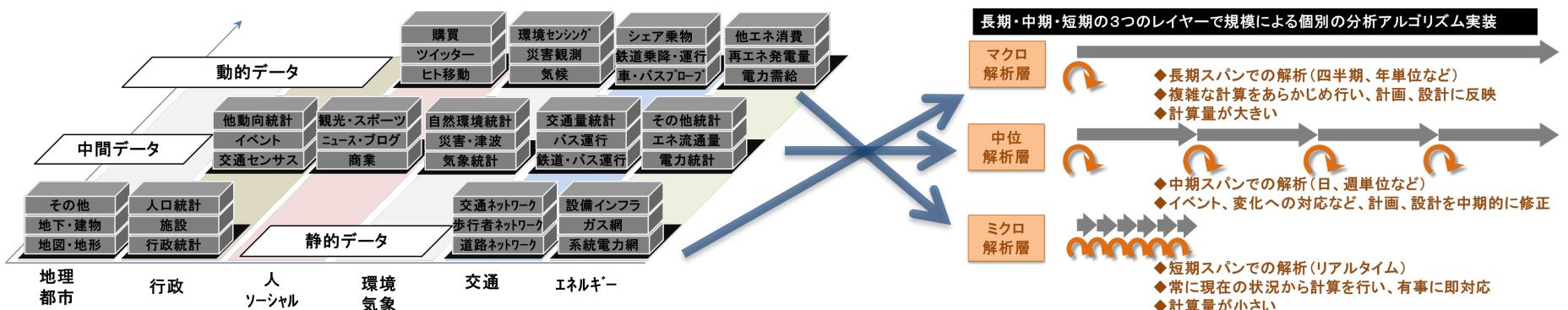
実世界の様々な事象をモデリング。サイバー空間にて解析し、実世界にフィードバック。



長期・中期・短期の3つのレイヤーで規模による個別の分析アルゴリズム実装



- マクロ解析層には、静的データを活用
 - 変動データは統計化。動向として活用可能
- 中位解析層には、中間データを活用
 - 静的な情報はベースデータとして活用
 - 変動の大きなデータは統計化。動向として活用可能
- ミクロ解析層には、動的データを活用
 - 静的・中間データはベースデータとして活用

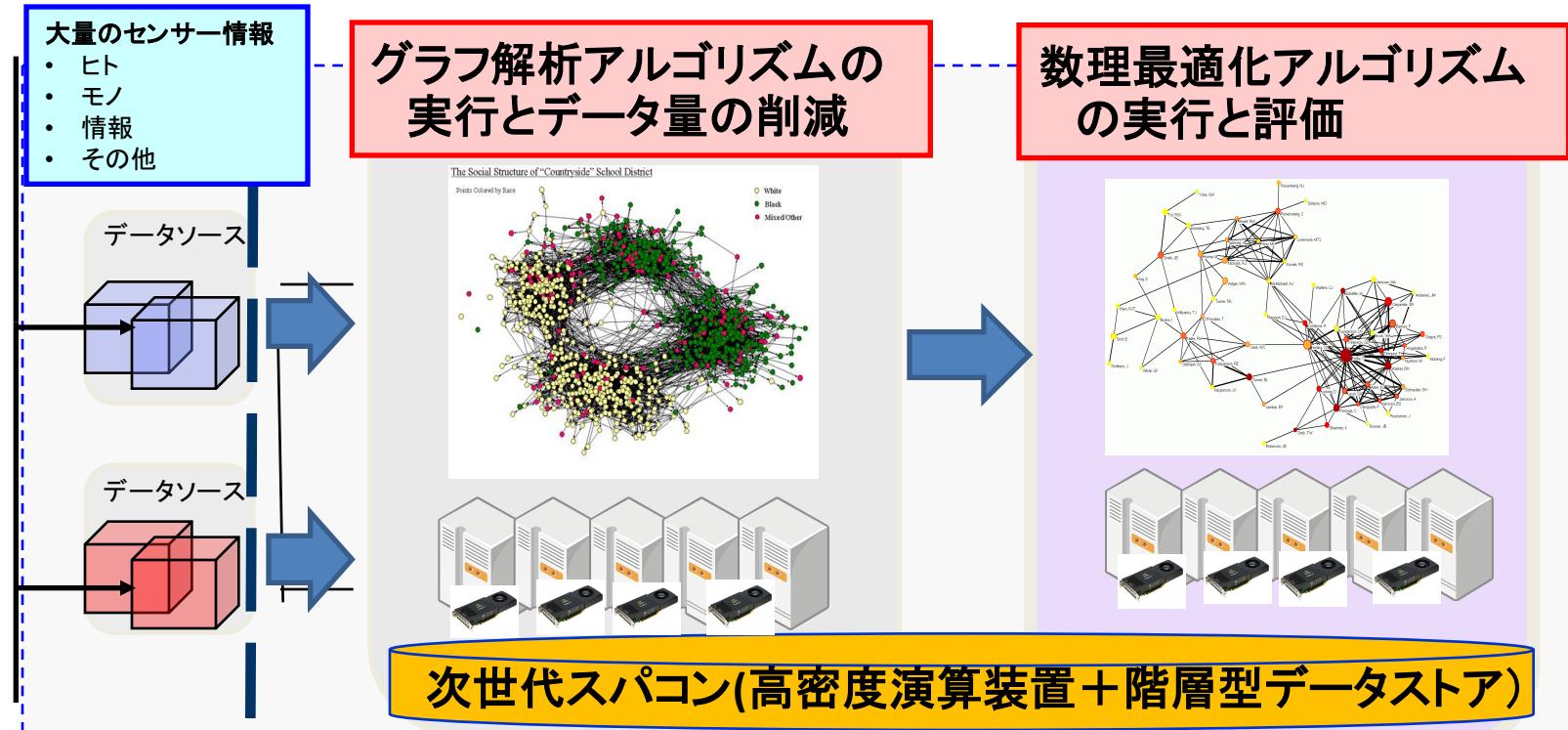


都市OS解析層が支える社会～交通



グラフ探索及び数理最適化ライブラリによる大規模グラフ処理基盤

- 大規模センサーから到着するストリーミングデータに対して精緻な解析を実現する**大規模グラフ処理基盤**を開発する
- 大規模グラフ処理基盤は以下の処理系から構成される
 - グラフ解析アルゴリズムの実行**: 最短路計算、ネットワーク内の各点の重要度を推定。各点の周辺、及び広域内における影響（情報の伝播力）を計算 → 重要な要素を失うことなくデータ量を削減する
 - 数理最適化アルゴリズムの実行**: 施設配置問題、集合被覆（分割）問題、スケジューリング、配送計画問題などの数理最適化問題 → モビリティに関する改善提案
 - 大規模なセンサーデータを**高速かつ重要性を失うことなく縮約すること**によって、精緻な解析を実現
- 計算及びデータ蓄積の基盤となる次世代スパコン
 - 並列数の爆発的増大、不均質化、高密度化 & 記憶装置の多階層化・大容量化



ネットワーク・フロー

アルゴリズム

最大流・最小費用流

- 分散したものを最適に集約する

応用

交通計画、電力網、輸送計画、避難計画

応用例

普遍的最速流による避難計画

群衆を避難所に避難させる

- 避難者の行動をフローとして捉える
- ネットワークフローベースのアルゴリズム
- どの時点でも到達避難者数が最大
- 避難完了が最速

ミクロ解析層



短期



今の群衆と設備情報をデータ化

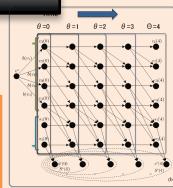


PC
Micro

最速フロー計算



リアルタイム計算



計算結果から各人の避難経路を誘導



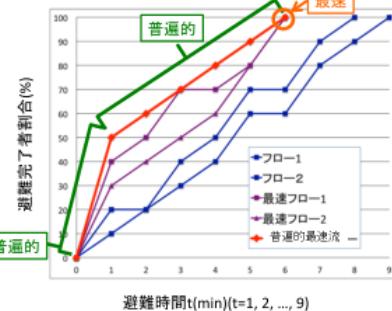
混雑度対応型リアルタイム避難誘導

普遍的最速流(Universally Quickest Flow : UQF)

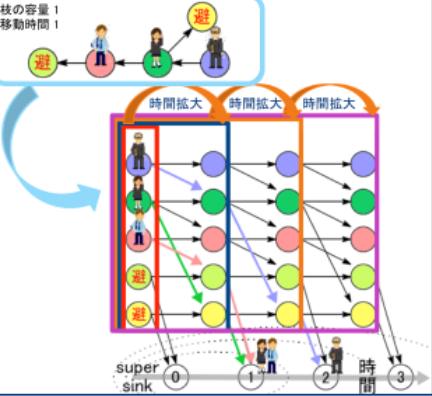
N. Kamiyama , "Studies on Quickest Flow Problems in Dynamic Networks and Arborescence Problems in Directed Graphs -A Theoretical Approach to Evacuation Planning in Urban Areas-", 2009

UQFの特性

- 各時刻において避難所に到達する避難者数が最大。(普遍的)
- 避難者全体の避難時間が最速。(最速)



UQFのアルゴリズム



- ネットワークの時間拡大。
- 最大流量を求める。
- 各容量の更新。
- 全員が避難を完了するまで1~3を繰り返す。

天神地下街における リアルタイム避難計画策定

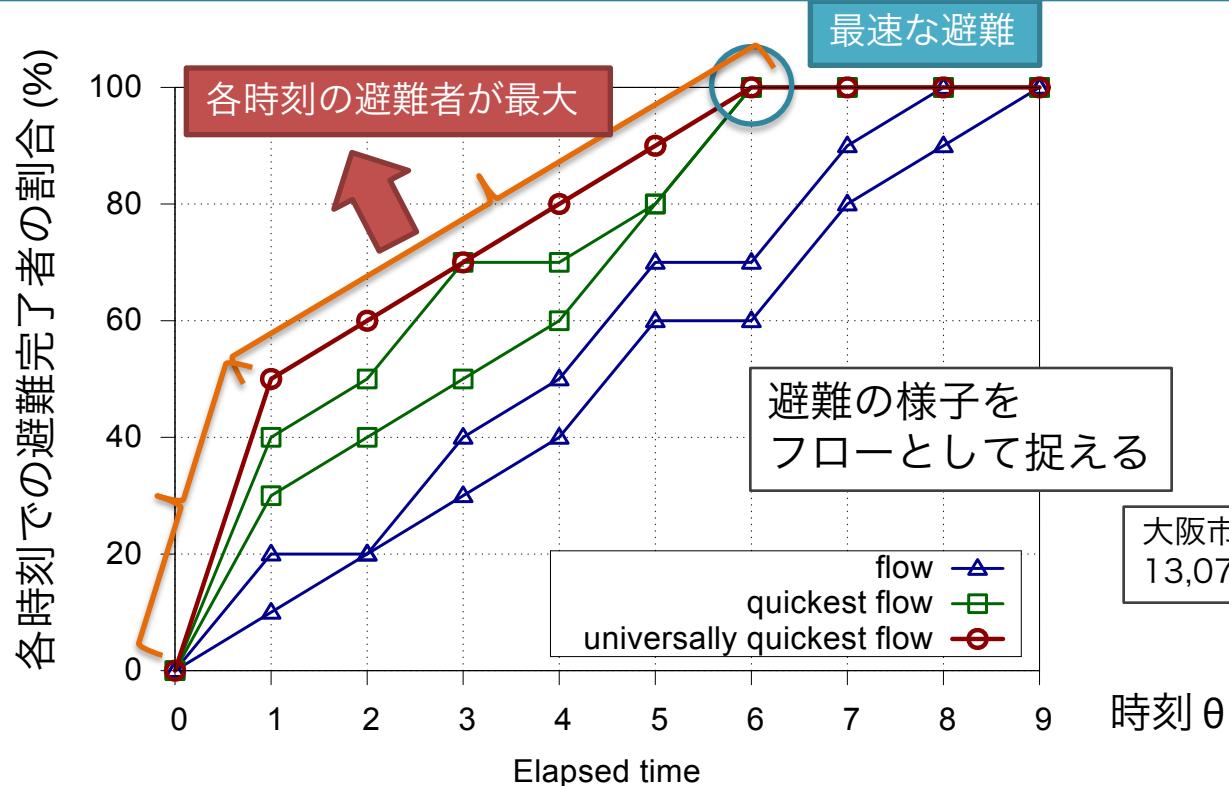
- 九州最大の繁華街・天神の渡辺通り直下（天神橋口交差点から渡辺通4丁目交差点の間）を南北に走る地下街
- 1番街から12番街まで
- 長さは約600メートル。
- 地下2階には計421台収容の駐車場
- 店舗面積は11,400m²、SC面積は28,000m²、テナント数は153
- 1日の歩行者通行量は約40万人

混雑度対応型リアルタイム避難誘導計画の策定
○は避難所:□は避難者(避難者が存在する枝)
混雑度:赤(大) ←→ 青(小)



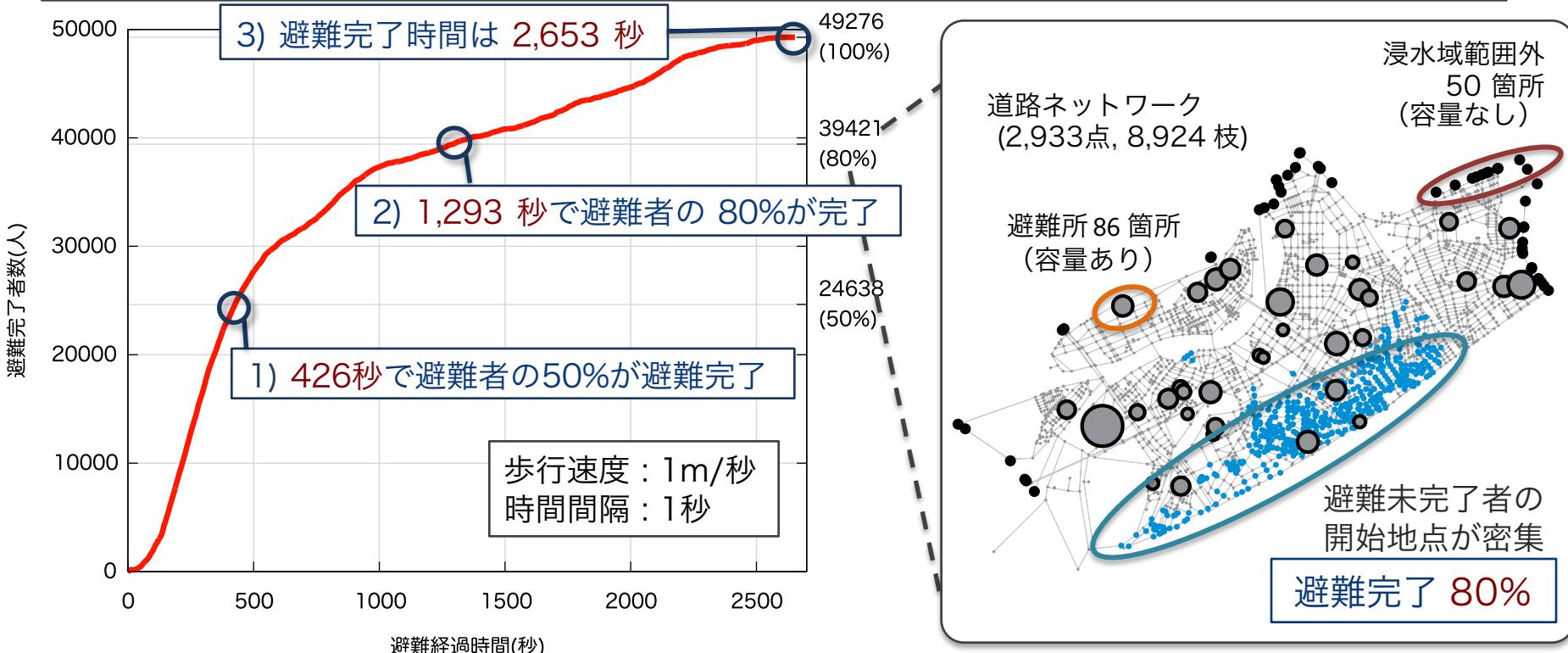
普遍的最速流を用いた避難計画の解析

- 大規模津波災害では、迅速に1次避難所への避難が重要
 - 非常に短い避難時間での高台や津波避難ビルへの避難が求められる
- 普遍的最速流を用いた津波災害の評価
 - 普遍的最速流は、各時間で最大の流量となる最速流



普遍的最速流を用いた津波災害の評価

- 淀川区、西淀川区周辺（避難者数 49,276 人に比べ、避難所容量 36,549 人が小さい）
 - 開始わずかで避難者の 50% が避難完了
 - 避難完了時間の半分は全体の 20% の避難者に費やされる
 - 避難時間を多く要する避難者の開始地点はある特定の地域に密集
- 計算時間：12.4 時間（逐次計算）
 - C++ で実装し、Max-Flow 計算には Push-relabel 法の改良版を用いた
 - 2,653 回の時間拡大により、最大で 778 万点、3087 万枝の巨大グラフに



中心性探索

アルゴリズム

重要度・重心・最短路

- グラフ化されたネットワークの中心性

応用

交通網解析、電力網解析、
SNS解析、脳解析

応用例

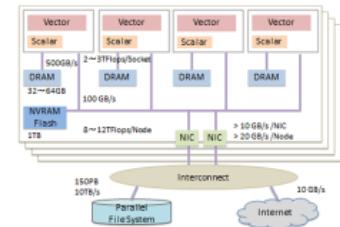
道路の重要度解析

よく使われる道路を道路網の
トポロジーから計算

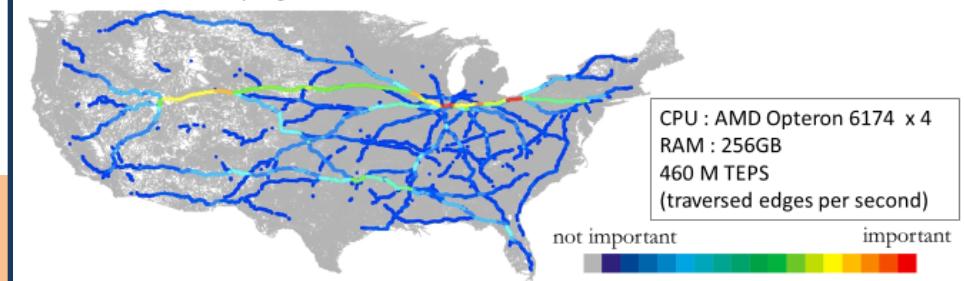
- 道路のトポロジー情報を使いグラフ化
- 全対全の経路で使用頻度の高い道路を探索
- 高頻度の道路がボトルネック
- ボトルネック解消の施策を都市計画に反映

Fast and Exact Computation of Betweenness Centrality
on Post peta-scale Supercomputer

- Exact computation → All pairs shortest paths
- Ex. USA (23,947,347 nodes, 58,333,344 edges)
 - 1 thread → over 35 days
 - 1M threads → only 3 minutes → real-time analysis
- Many applications
 - Analyzing massive social networks
 - Optimizing smart grid networks
 - Health care (disease spread)
 - Understanding complex life system



10,000 - Random Sampling 15 minutes : 48 cores



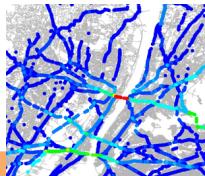
中位解析層



中期

道路、交通網、ヒト、クルマを
データ化ボトルネック解析、最適化
計算

ワークステーション



オンデマンド計算

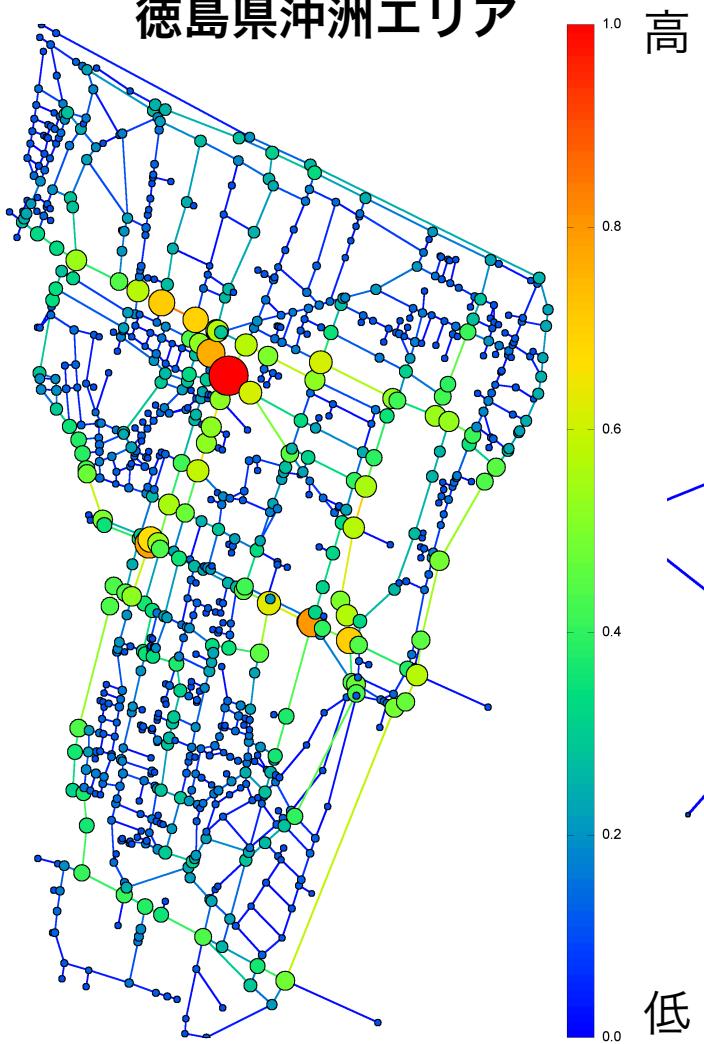
最適化データを実世界に
展開イベントに応じた中期交通
スケジュール

Mid

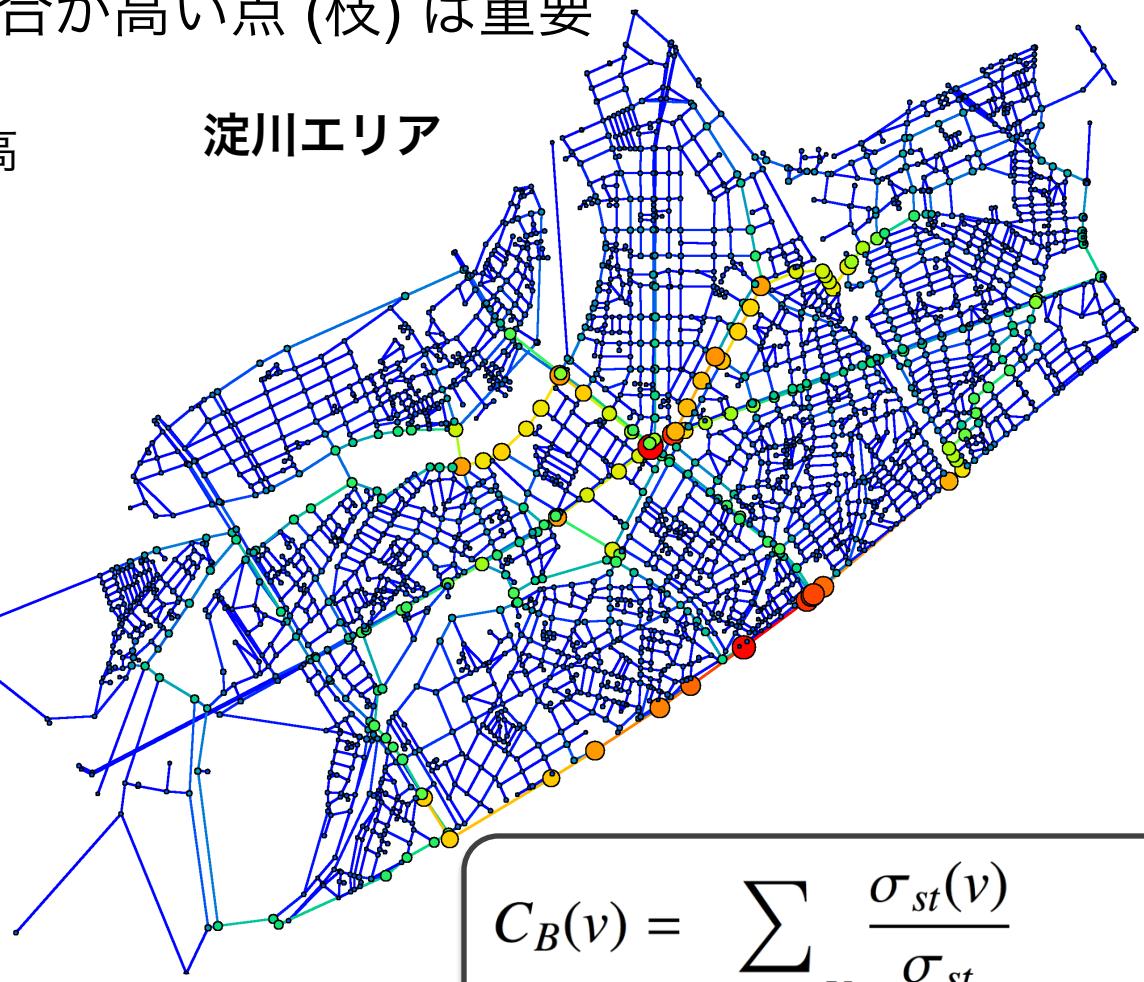
媒介中心性 (Betweenness centrality; BC)

- 最短路に含まれる割合が高い点(枝)は重要

徳島県沖洲エリア



淀川エリア



$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

σ_{st} : (s,t)-最短路の数

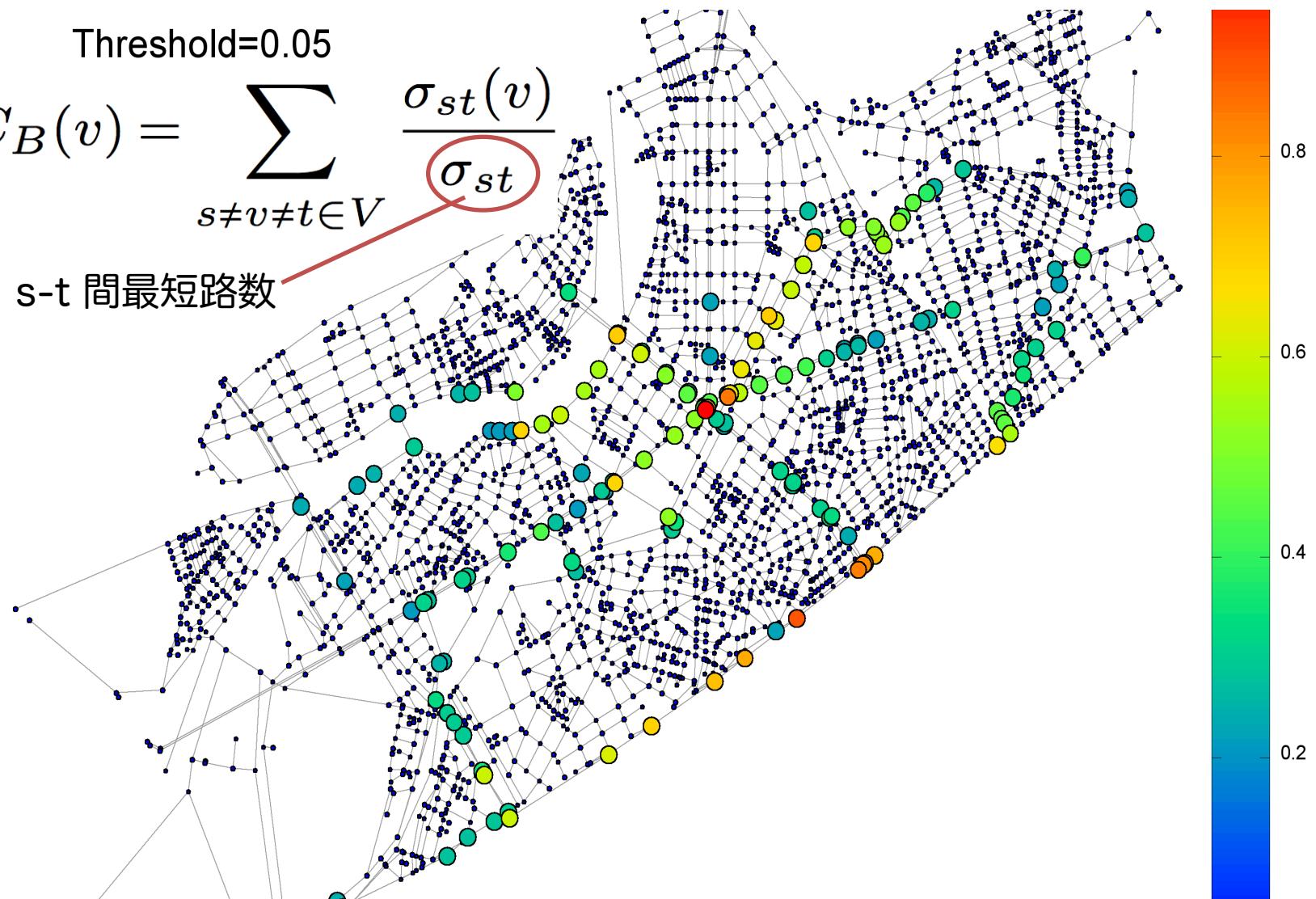
$\sigma_{st}(v)$: v を通る(s,t)-最短路数

媒介中心性 (Betweenness centrality; BC)

Threshold=0.05

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

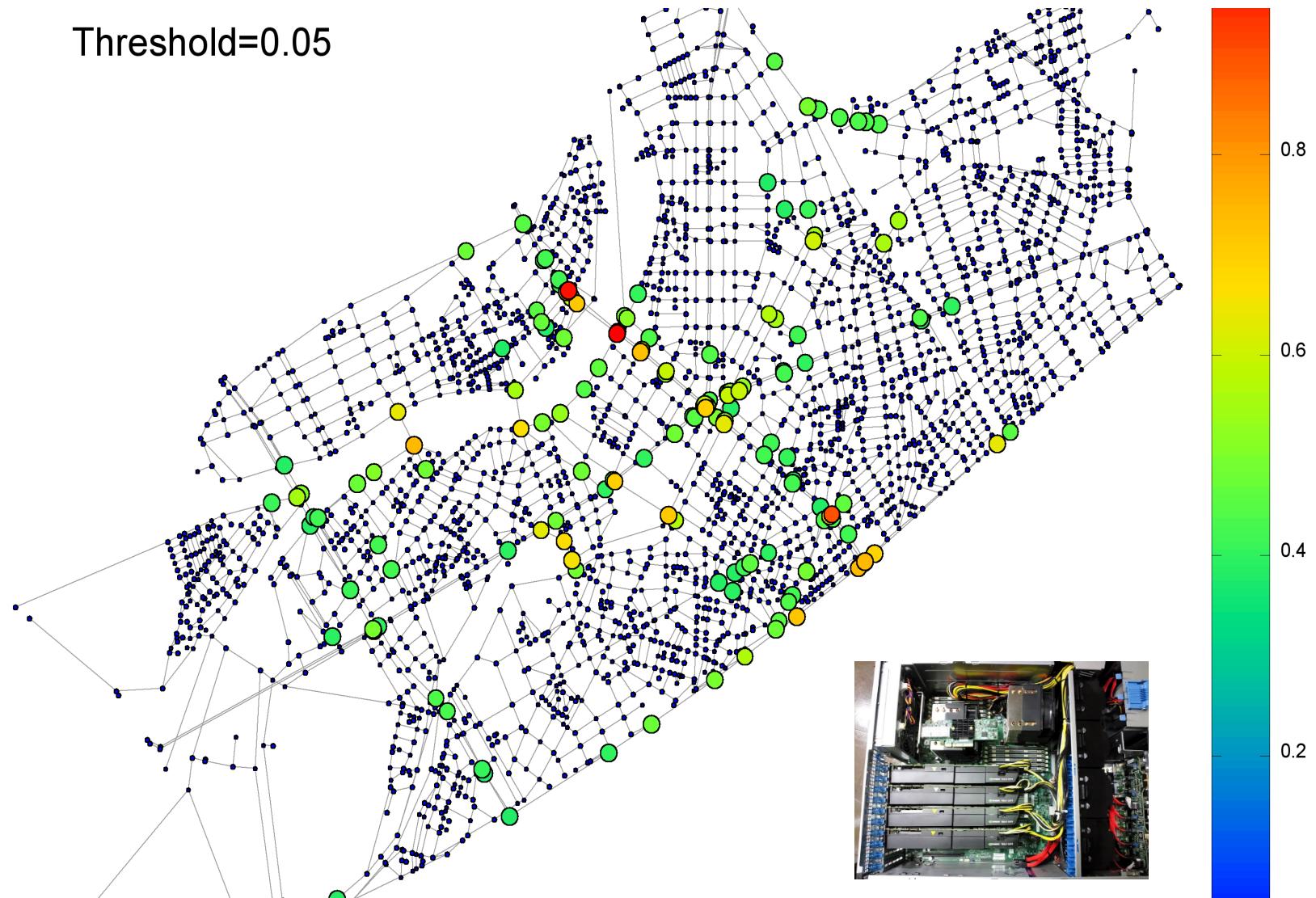
s-t 間最短路数



Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass through v :

Current_flow_betweenness_centrality

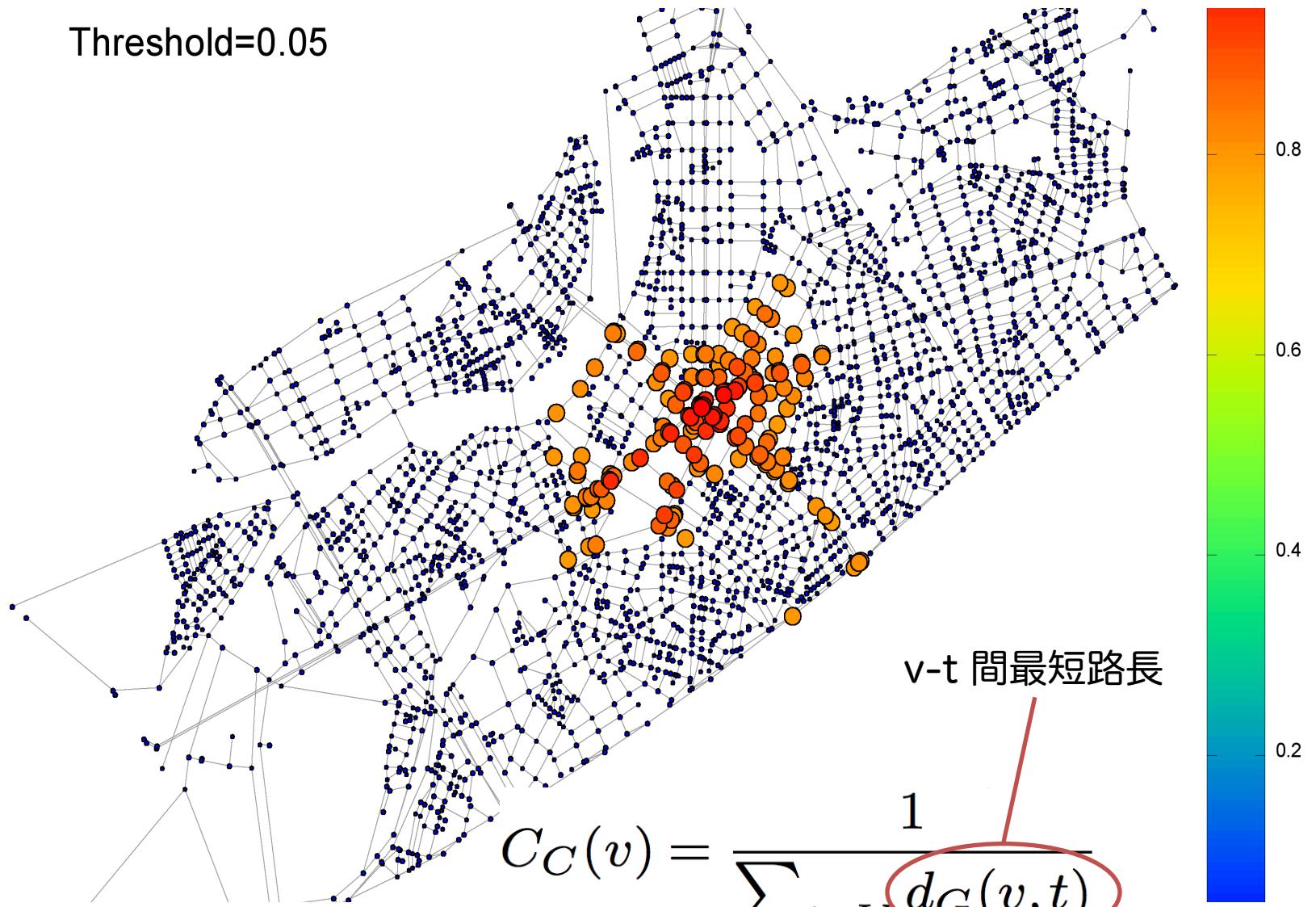
Threshold=0.05



Current-flow betweenness centrality uses an electrical current model for information spreading in contrast to betweenness centrality which uses shortest paths.

closeness_centrality

Threshold=0.05



$v-t$ 間最短路長

Closeness centrality at a node is 1/average distance to all other nodes.

communicability_centrality

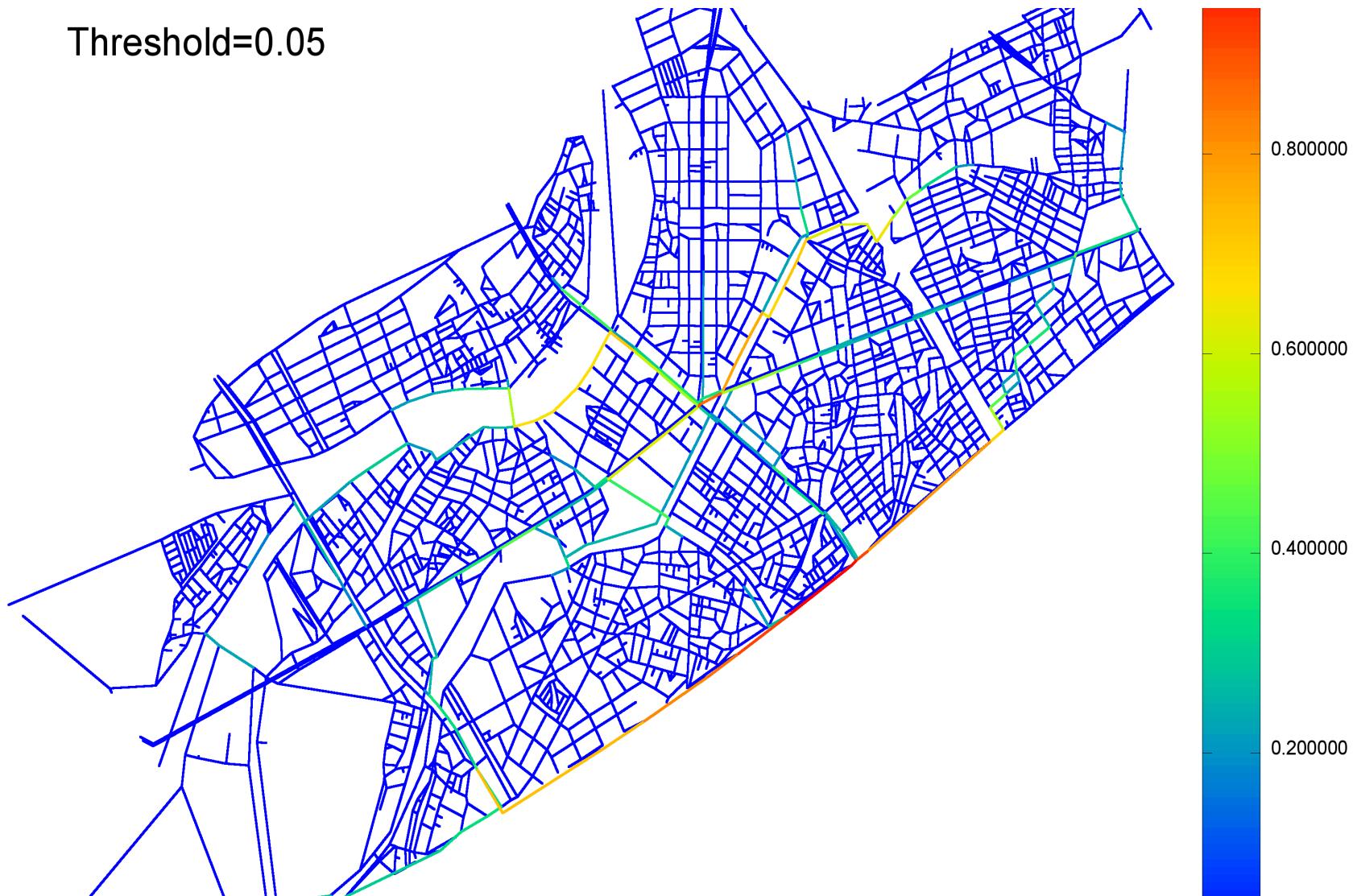
Threshold=0.05



Communicability centrality, also called subgraph centrality, of a node n is the sum of closed walks of all lengths starting and ending at node n .

edge_betweenness_centrality

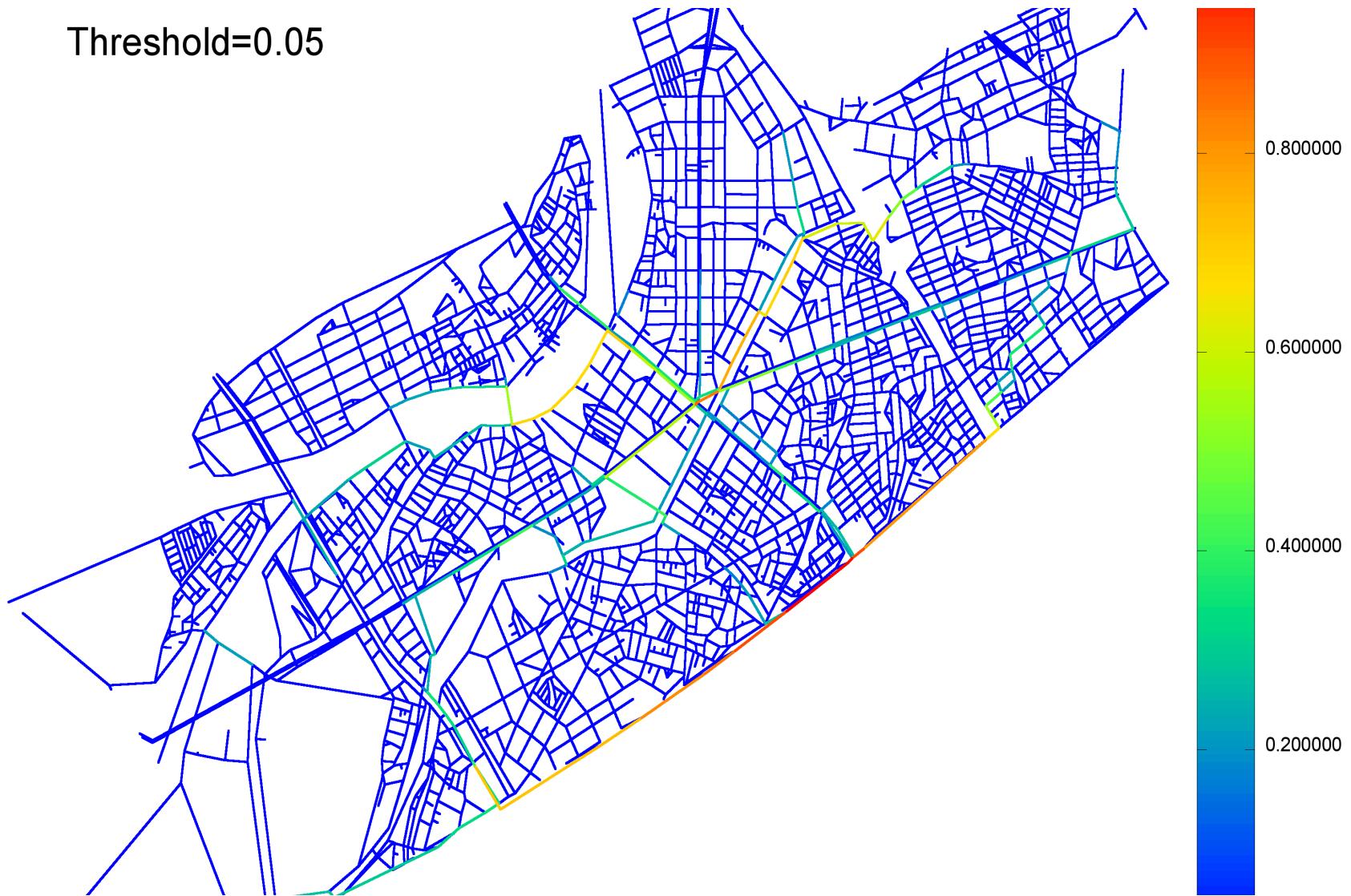
Threshold=0.05



Betweenness centrality of an edge e is the sum of the fraction of all-pairs shortest paths that pass through e :

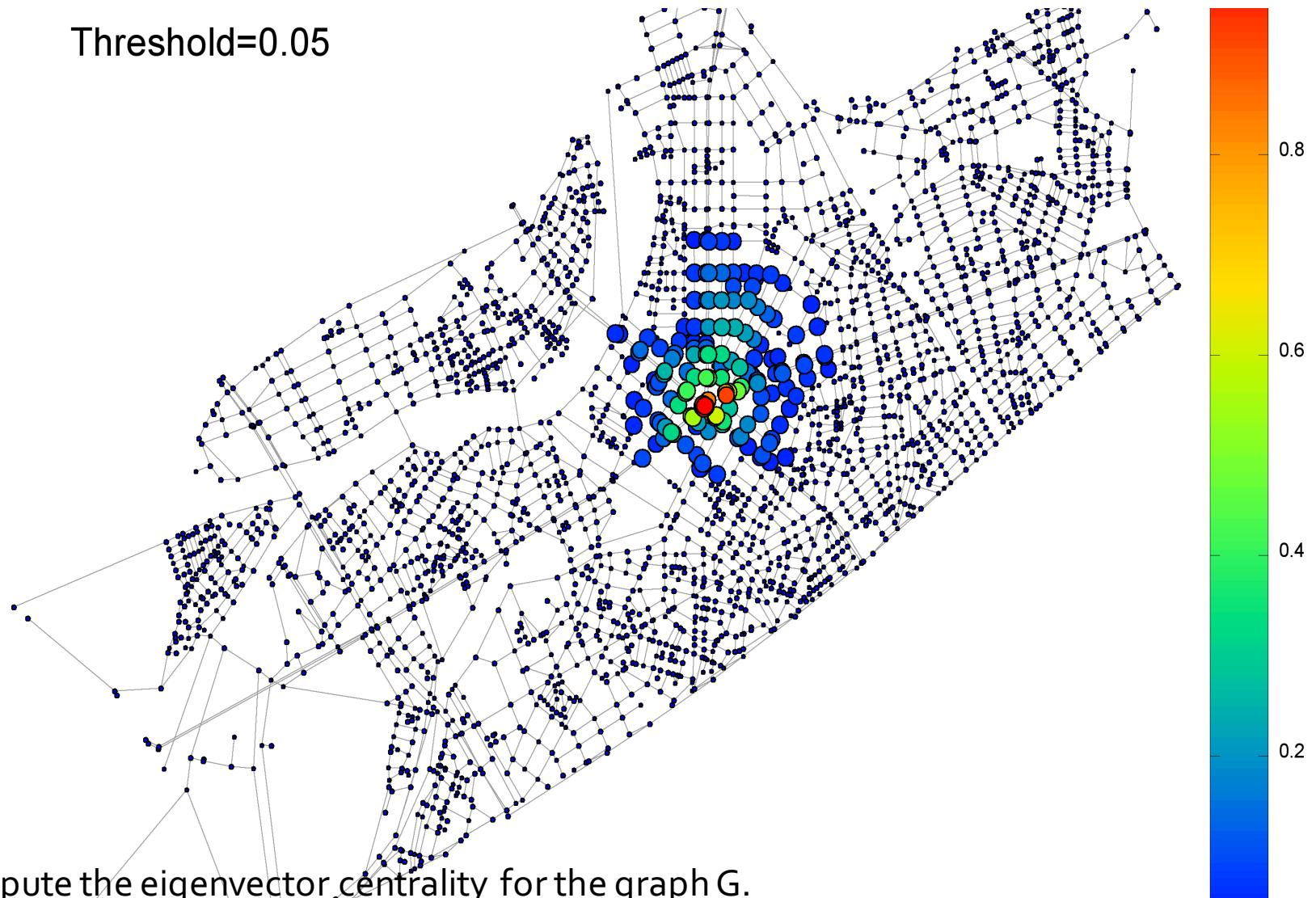
edge_load

Threshold=0.05



eigenvector_centerality

Threshold=0.05

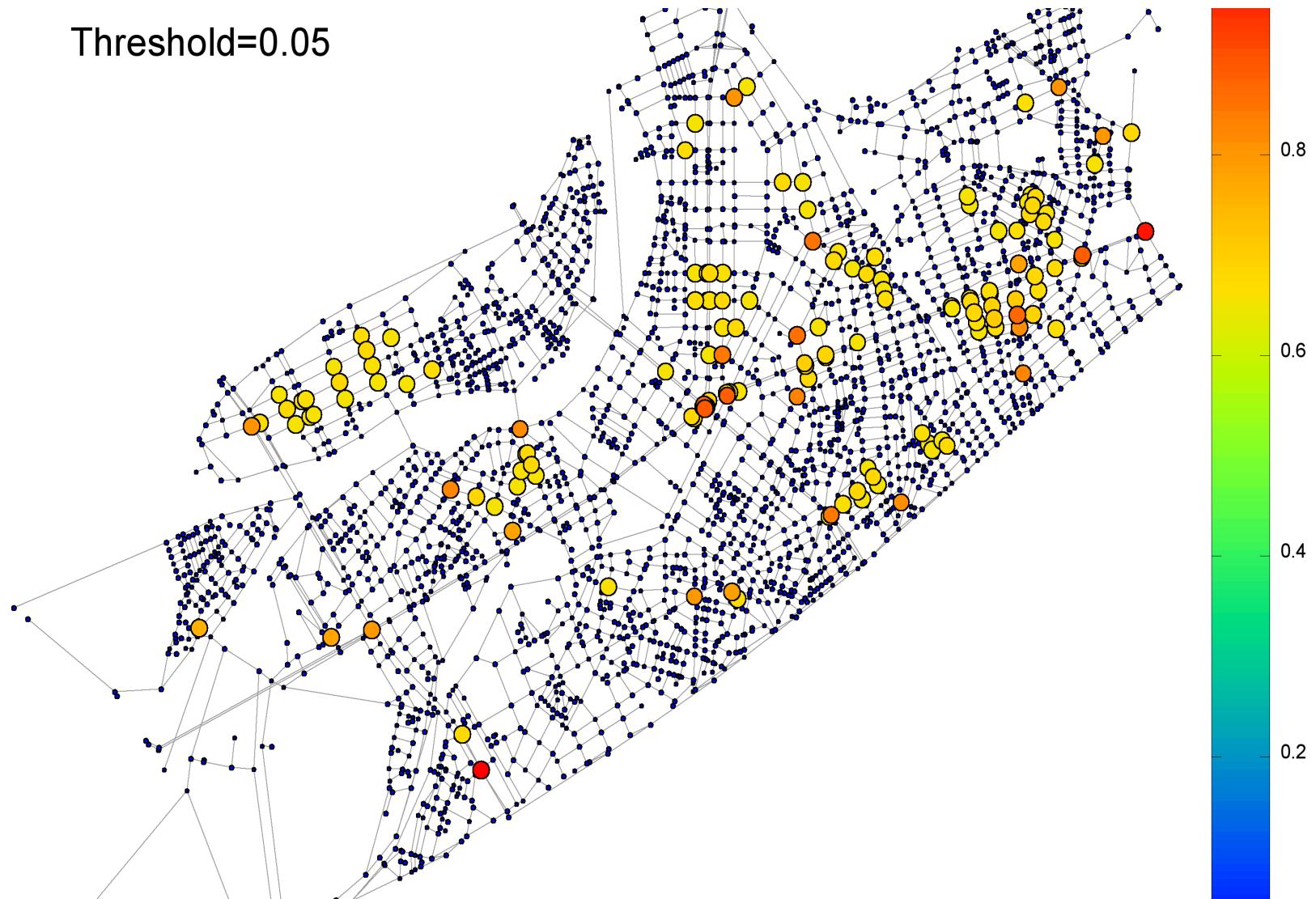


Compute the eigenvector_centerality for the graph G.

Uses the power method to find the eigenvector for the largest eigenvalue of the adjacency matrix of G. (隣接行列Aの最大固有値に対する固有ベクトル)

katz_centrality

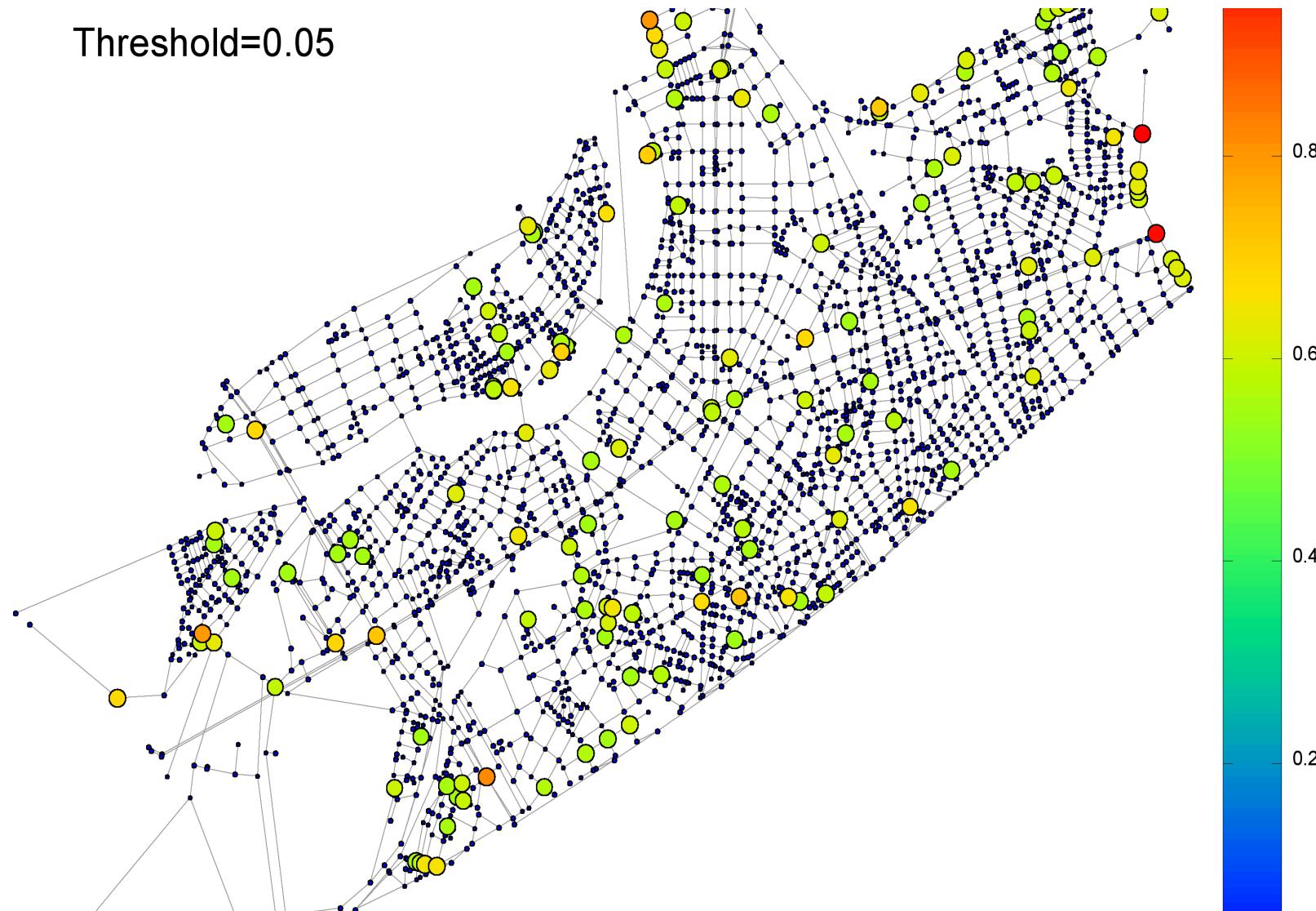
Threshold=0.05



Katz centrality computes the relative influence of a node within a network by measuring the number of the immediate neighbors (first degree nodes).

pagerank

Threshold=0.05



福岡市交通ネットワーク

Betweenness centrality

枝長を考慮しない
厳密解: 2m 30s

HP ProLiant m710

サーバーカートリッジ



有向グラフ

点数 314,571
枝数 694,906



東京都周辺交通ネットワーク

東京都周辺交通ネットワーク

Open Street Map

<https://mapzen.com/metro-extracts>

有向グラフ

点数 6,509,809

枝数 14,460,834

Huawei RH5885H V3

CPU : Intel Xeon E7-4890 x 4

Memory : 2.0TB (32GB

LRDIMM x 64 DIMMs)



Betweenness centrality

東京都周辺交通ネットワーク

Open Street Map

<https://mapzen.com/metro-extracts>

有向グラフ

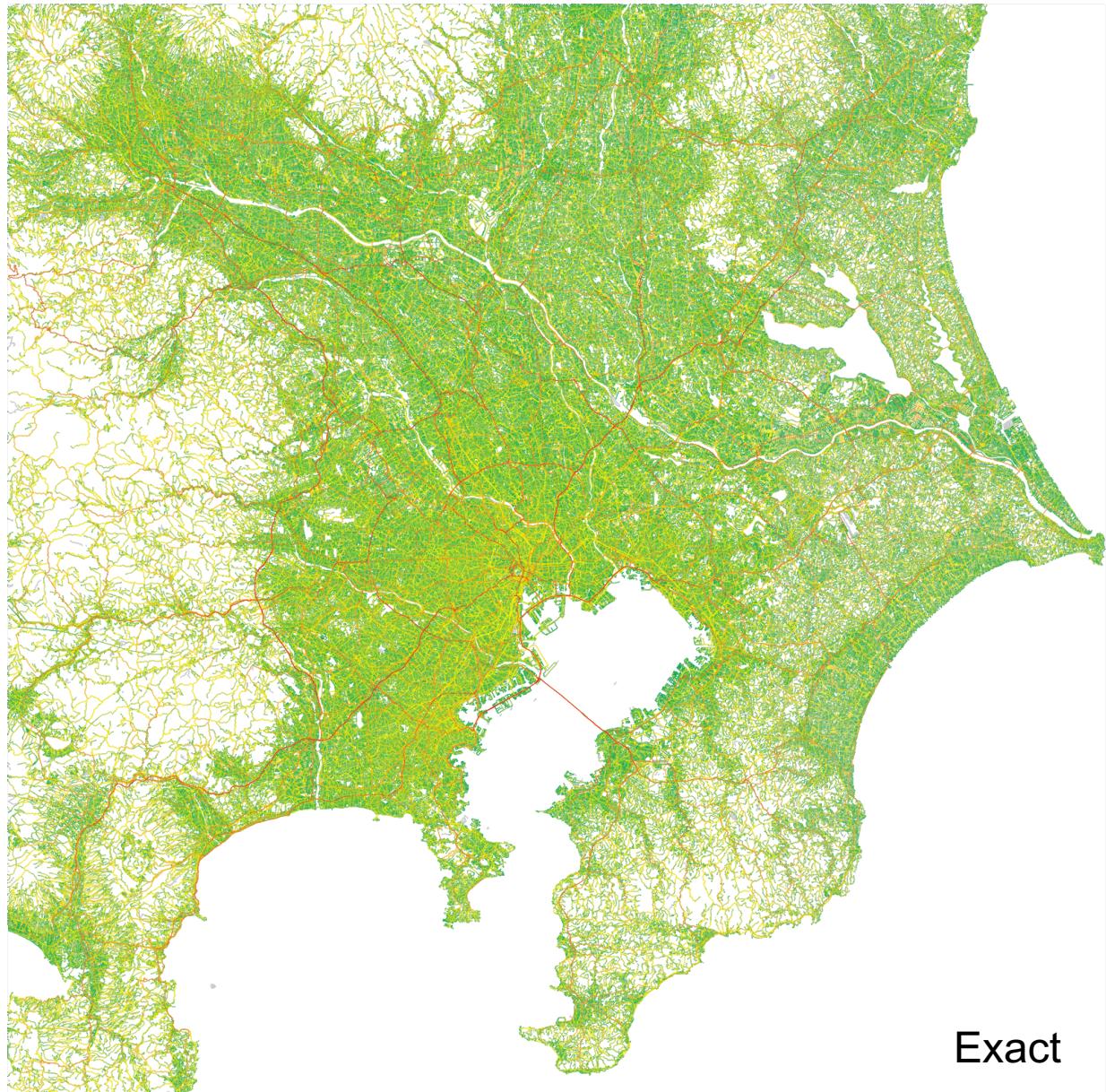
点数 6,509,809

枝数 14,460,834

Betweenness centrality

枝長を考慮しない

厳密解: 98h 27m 37s



ランダムサンプリングによる近似

東京都周辺交通ネットワーク

Open Street Map

<https://mapzen.com/metro-extracts>

有向グラフ

点数 6,509,809

枝数 14,460,834

Betweenness centrality

枝長を考慮しない

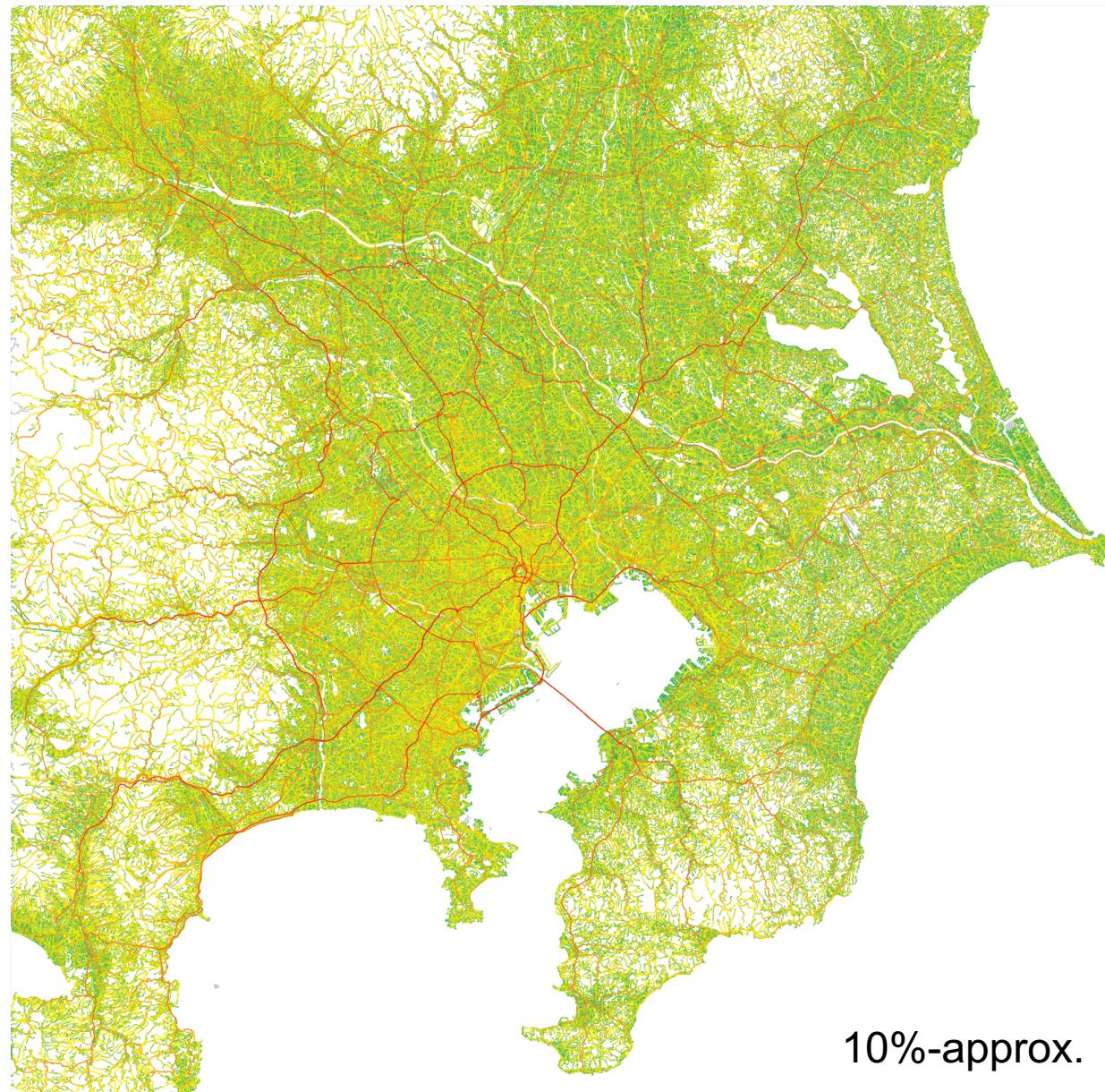
厳密解: 98h 27m 37s



枝長を考慮しない

10%近似解: 9h 51m 11s

10%なら実行時間はほぼ1/10に



数理計画問題

アルゴリズム

凸及び非凸整数計画問題

- 膨大な組合せ条件の中から最適解の算出

応用

施設配置問題、集合被覆（分割）問題、スケジューリング、配送計画問題などの数理最適化問題

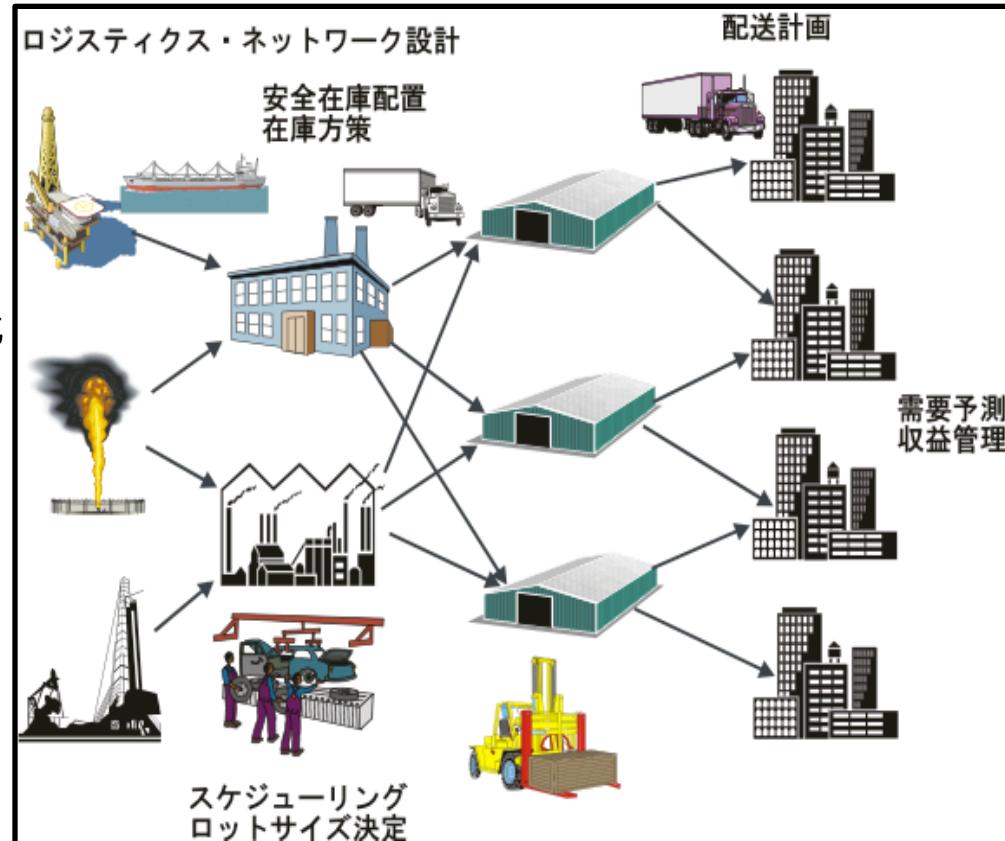
特徴

実世界



サイバー空間

実世界（この世） のデータから**サイバー空間（あの世）** を次世代スーパーコンピュータ上に再現（構築）。最終的には**実世界（この世）** の都市計画策定やネットワーク構築などに反映させる。
例：福岡市 152万市民の朝から夜まで一日分のモビリティの把握及び移動＆行動モデルの構築



マクロ解析層



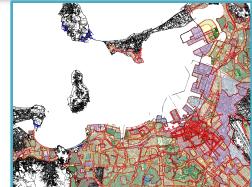
交通網、施設の分布状態をデータ化



ボトルネック解析、最適化計算



最適化データを都市計画に反映



Macro

Deep Calculation

Optimal Urban Planning

交通分野で扱うデータ

連携企業データ・購入データ

キャリアデータ

Agoop

WiFiデータ

福岡市

九州フリー

ヒト移動
情報
気象情報

Person Trip

福岡市

交通センサスデータ

国土交通省

気象データ

気象庁

アメダス

九大収集データ

Bluetoothビーコン

Bluetoothセンサー

センサー
データ
@伊都C

屋内レーザースキナ

ネットワークカメラ

P-Sen

B-Sen

交通情報

鉄道ネットワーク

駅データjp

西鉄

JR九州

運行データ

西鉄

JR九州

地下鉄路線

福岡市営

乗降数データ

福岡市営

バスネットワーク

西鉄

バスプローブ

西鉄

クルマプローブ

トヨタ

日産

ホンダ

タクシープローブ

Uber

VICS

JARTIC

地理情報
統計情報
行政情報

人口統計データ

Mapple

行政オープンデータ

福岡市

デジタル地図データ

Mapple

地下・建物3D

OSM

伊都C

道路ネットワーク

Mapple

歩行者ネットワーク

Mapple

福岡市都心部



交通分野への展開

伊都キャンパス

- センシング
- 屋外: P-Sen
- 屋内: B-Sen



センサー解析



最適化・快適化

都市・交通の最適化

- ✓ 道路ネットワーク解析によるボトルネック解消
- ✓ 交通機関、クルマ、ヒトの移動解析による混雑緩和

インフラ

交通

- 鉄道ネットワーク
- バスネットワーク
- 通信ネットワーク

地理・行政

- マップデータ
- 道路ネットワーク
- 歩行者ネットワーク
- 建物・地下3D
- 統計データ
- 行政オープンデータ



プローブ・運行

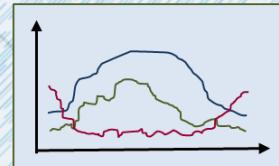
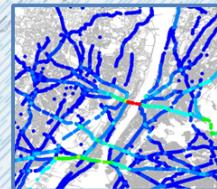
- クルマプローブ
- バスプローブ
- 鉄道運行
- バス運行

ヒト移動

- キャリア・WiFi
- 交通センサス
- 鉄道運行

グラフ解析

福岡市都心部



気象・環境

- イベント
- 気象
- 災害



相関分析

- 様々なデータを収集・蓄積するオープンシステムとしての都市OS
 - ビッグデータ / オープンデータ / センサーデータ / RDFデータ
- 数理を生かした最先端の最適化・解析を提供するプラットフォームとしての都市OS
- 様々なデータの掛け合わせによる活用と適切な最適化・解析エンジンによるアプリケーション開発プラットフォームとしての都市OS
 - 起業・創業を支援