

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

中央大学理工学部
JST CREST
藤澤克樹



大規模グラフ最適化(解析)の重要性

1: 実社会における本質的課題の解決に必須な手法

- 大規模災害等で突発的に発生してリアルタイムに状況が変化し早急な解決が望まれる問題
- 最先端理論(Algorithm Theory) + 大規模実データ(Practice) + 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

- 交通データに対する経路探索: 動的に変化する交通量等から高速な重要度判定を行うことによって、交通管制等に活用する
- ソーシャルネットワークデータに対する解析: 動的な重要度、影響度の判定。各点の周辺、及び広域内における影響(情報の伝播力)を推定する
- その他: 疫病の拡散、人口の増減、経済動向等の分析。ライフライン等の基盤計画(電力、水、食料)。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)。安全保障分野(組織構成の解明、事件事故の事前予測)。

3: HPC分野における注目のアプリケーション

- Graph500 などの新しいベンチマークテストが提案。各国ともグラフ探索等のプログラム開発とスパコンでの実行を重視



JST CREST

応募研究領域: 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

研究課題名: 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

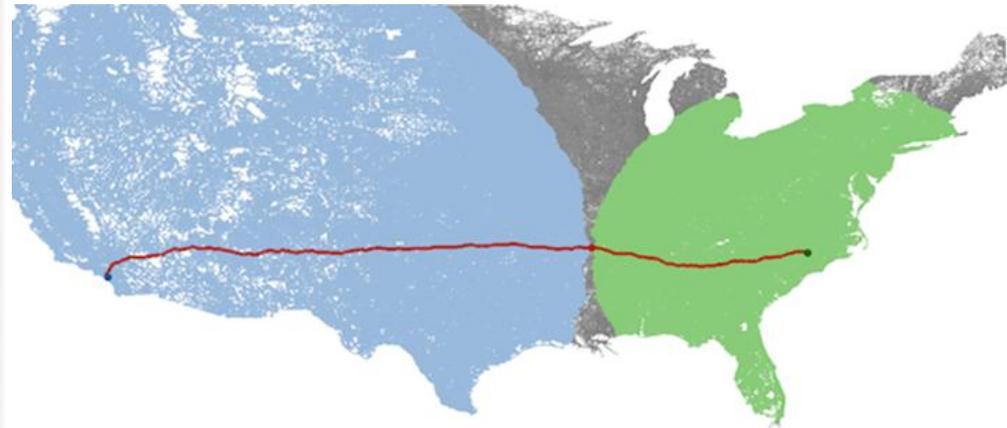
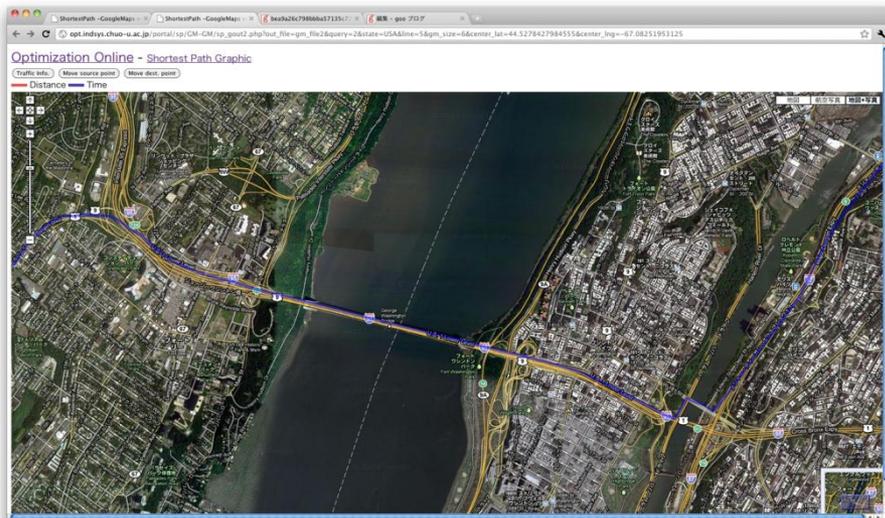
(研究代表者 藤澤克樹(中央大学))

研究計画の概要

- 実社会では解決困難だが、早急に解決すべき問題が多数存在する
 - 大規模災害等で突発的に発生してリアルタイムに状況が変化し早急な解決が望まれる問題等
- **グラフ解析システム**がこれらの問題解決に非常に重要 → **ポストペタスパコンでの実行**が必要 → アルゴリズム的、システムの的に様々な**解決すべき課題と困難が存在する**
- 最適化とHPC系研究者のポストペタスパコン上での**Co-design**による**解決**
- ポストペタスパコン上での基盤ソフトの整備に貢献すると共に安心安全な社会の実現を目指す

超大規模ネットワークにおける高速探索技術の開発

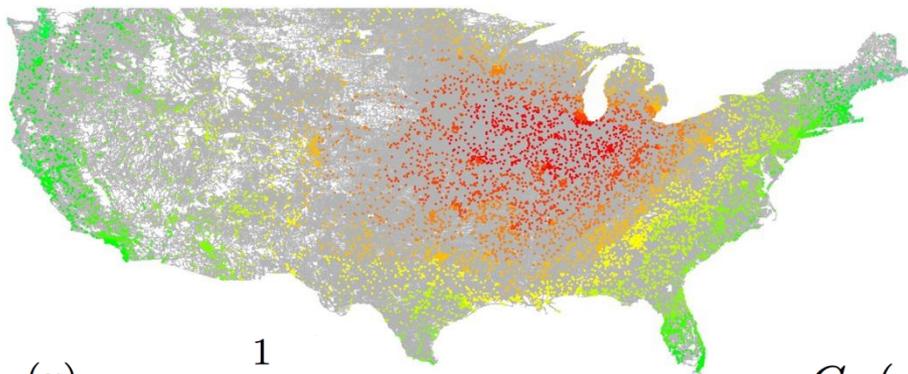
- 超大規模ネットワークデータ (点数10億以上)が対象
- 最短路(1対全), グラフ探索(深さ優先、幅優先等)
- グラフ探索の局所的な評価では優先キュー(ヒープ木)を用いる⇒実行時間、メモリ消費量が安定的
- 実行時間(1対全最短路計算: CPU : Intel Xeon 5670 2.93GHz)
 - 全米グラフ 3秒 (点数 $n = 23,947,347$ 、枝数 $m = 58,333,344$)
 - 超大規模グラフ 870秒 (点数 $n = 10$ 億、枝数 $m = 20$ 億)
- 多くの応用(交通ネットワーク, e-コミュニティのネットワーク、ロジスティクスネットワーク)



全米グラフに対する重要性(中心性)の計算

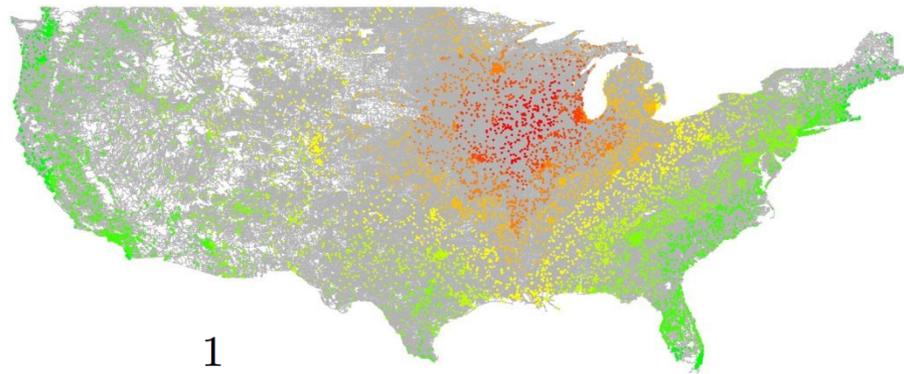
- ネットワーク内での各点の**重要度**を推定
- **各点の周辺、及び広域内における影響**(情報の伝播力)を推定する
- 超大規模ネットワークデータ上での短時間に大量のグラフ探索を行う必要がある

Closeness



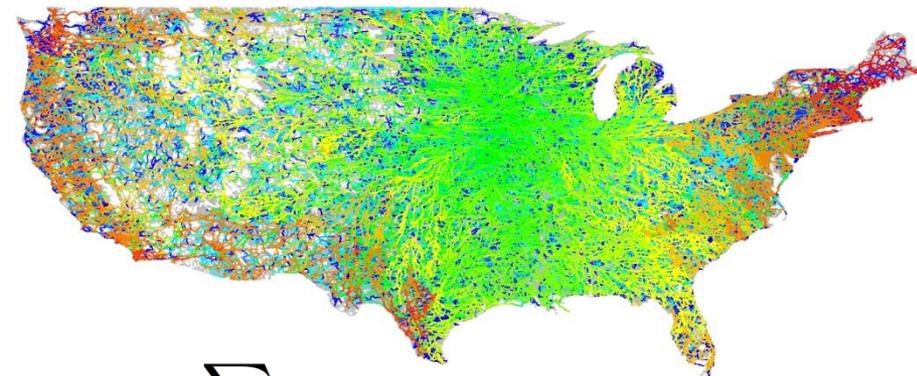
$$C_C(v) = \frac{1}{\sum_{t \in V} d_G(v, t)}$$

Graph



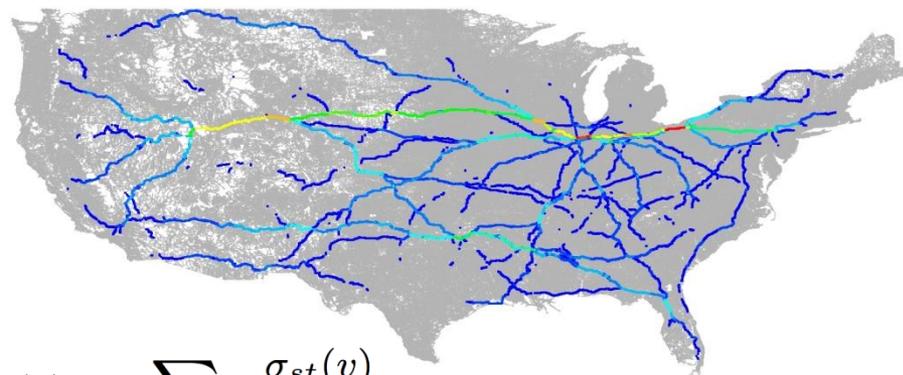
$$C_G(v) = \frac{1}{\max_{t \in V} d_G(v, t)}$$

Stress(対数)



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

Betweenness



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

The U.S. Electric Grid

<http://www.npr.org/templates/story/story.php?storyId=110997398>

About This Map »

Click on the links below to switch layers on and off.

EXISTING LINES

- 345-499 kV
- 500-699 kV
- 700-799 kV
- 1,000 kV (DC)

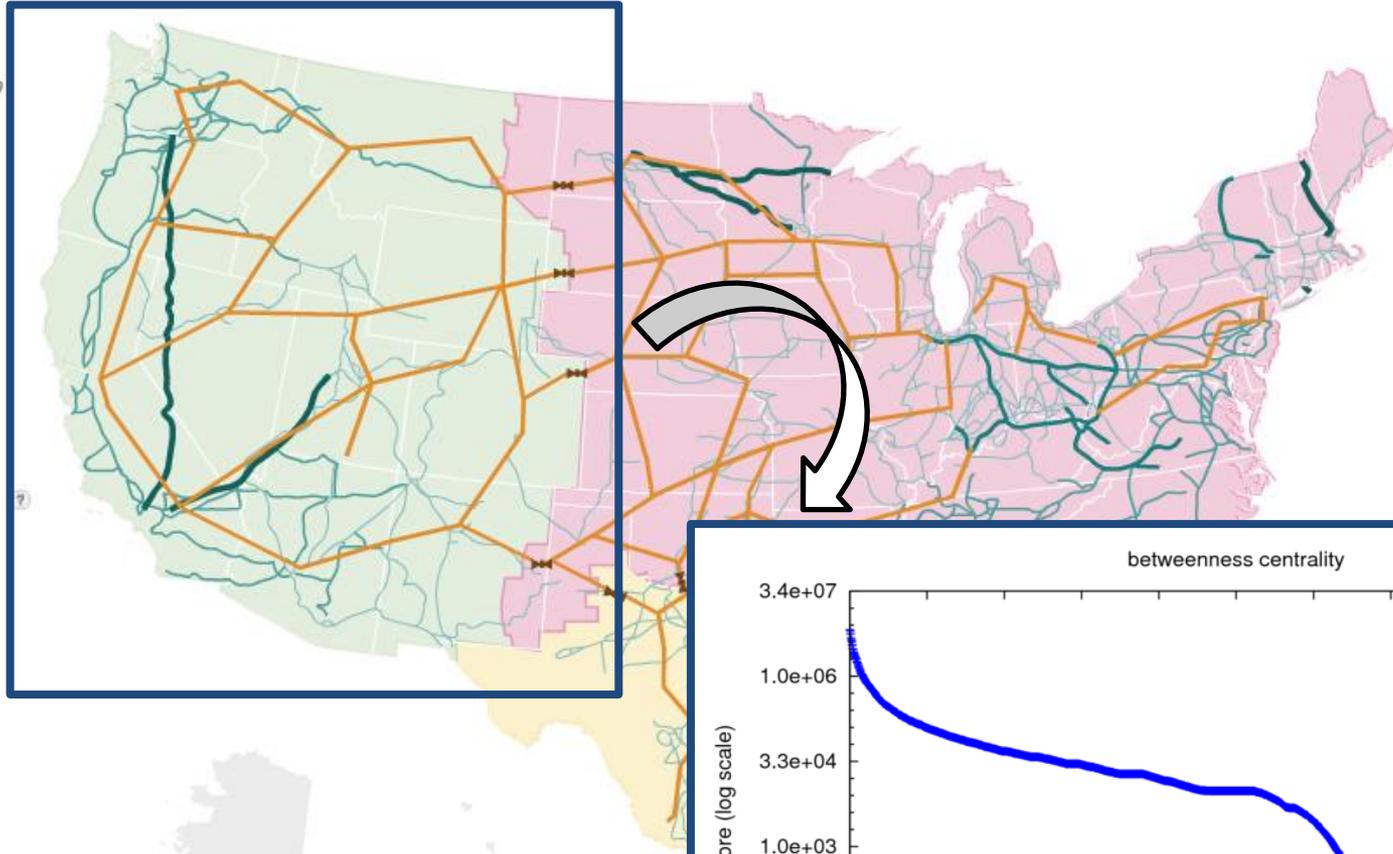
PROPOSED LINES

- New 765 kV
- AC-DC-AC Links

INTERCONNECTIONS

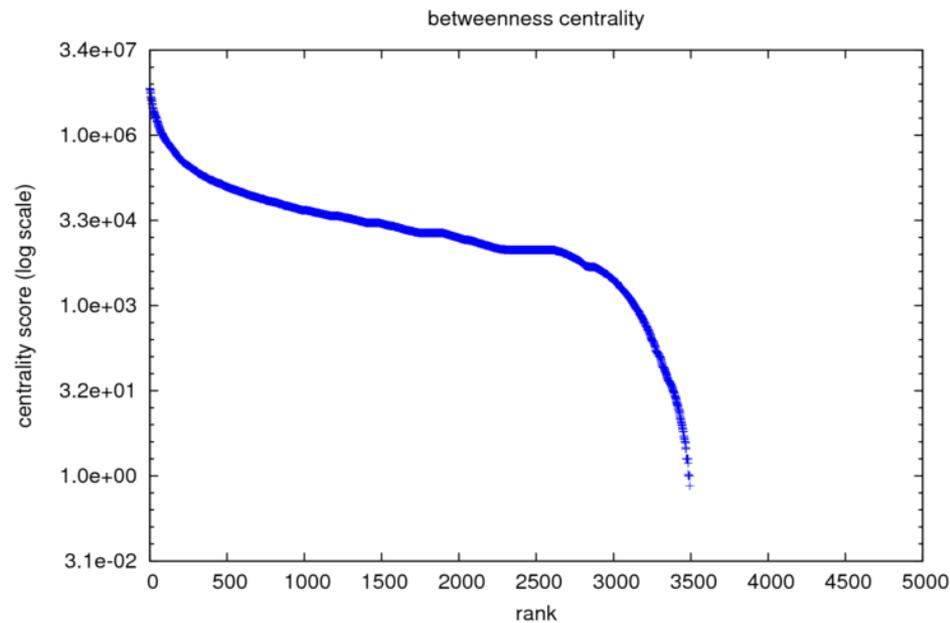
Major sectors of the U.S. electrical grid

- Eastern
- Western
- Texas (ERCOT)



[Electrical Centrality Measures for Electric Power Grid Vulnerability Analysis](#)

Wang, Z.; Scaglione, A.; Thomas, R. J. 2010





Graph500 (<http://www.graph500.org/>)

超大規模グラフの探索能力で計算機を評価する新しいベンチマーク

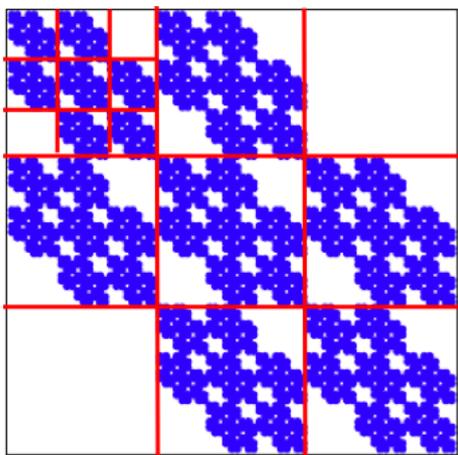
Kronecker graph

$$\arg \max_{\Theta} P(\text{Green Grid} \mid \text{Blue Grid} \xleftarrow{\text{Kronecker}} \Theta)$$

A: 0.57, B: 0.19
C: 0.19, D: 0.05

1	1	0
1	1	1
0	1	1

G_1



G_4 adjacency matrix

- 現在の指標 TEPS (traversed edges per second)
- 三つのカーネル
 - concurrent search,
 - optimization (single source shortest path)
 - edge-oriented (maximal independent set)
- 超大規模グラフへの適用

- Reference Code
 - Kronecker graph の生成
 - 1-BFS を全プロセスで並列に計算
 - 点数 2^{SCALE} , 枝数 $2^{\text{SCALE}+4}$

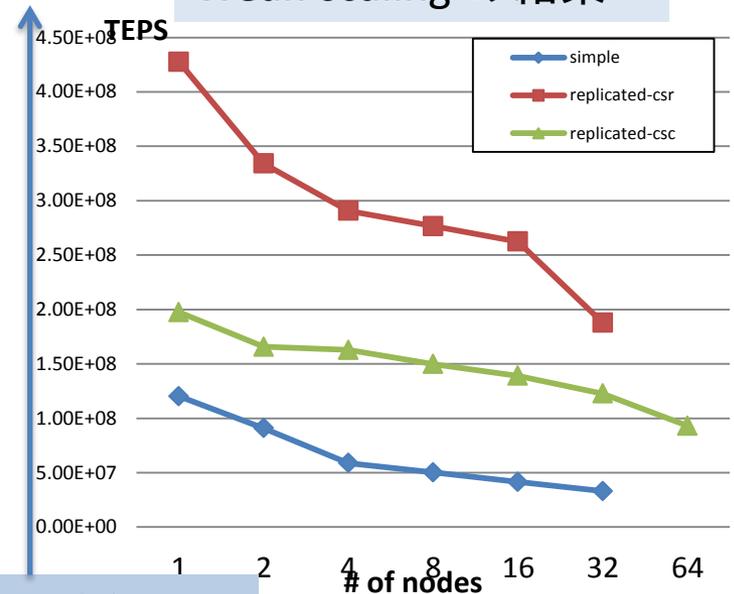
Complete Results - June 2011

Rank	Machine	Owner	Problem Size	TEPS
1	Intrepid (IBM Blue Gene/P, 32,768 nodes / 131,072 cores)	ANL	38	18,508,000,000
2	Jugene (IBM Blue Gene/P, 32,768 nodes / 131,072 cores)	Forschungszentrum Jülich	38	18,416,700,000
3	Lomonosov (MPP, 4096 nodes / 8192 cores)	Moscow State University	37	43,471,500,000
4	Hopper (Cray XE6, 1800 nodes / 43,200 cores)	LBL	37	25,075,200,000
5	Franklin (Cray XT4, 4000 nodes / 16,000 cores)	LBL	36	19,955,100,000

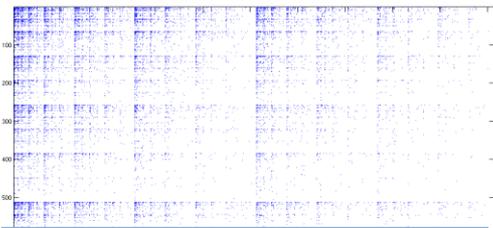
Graph500のTSUBAME2.0における 性能特性解析

- スパコンの新規ランキング指標 Graph500をTSUBAME2.0上で実行
- Weak-scaling (1ノードにつき Scale 26)により1ノードから64ノード(1536コア)まで性能特性解析
- Graph500が提供する参照実装では通信及び同期コストによりスケーラビリティが得られないことを確認
- 現在、2次元分割及びスケールフリー性を利用した頂点配置を用いた最適化アルゴリズムを実装中

Weak-scaling の結果



Upper is better

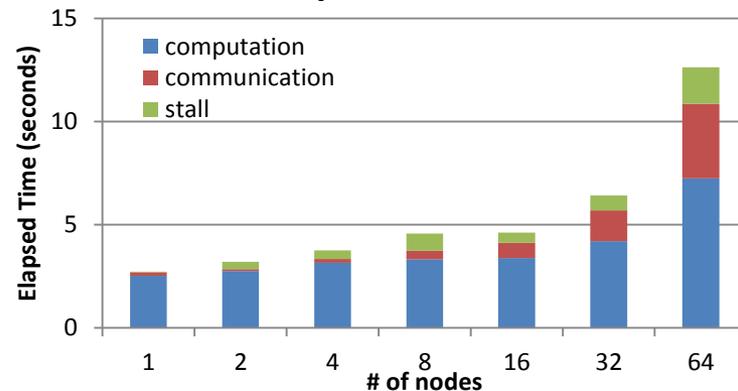


クロネッカーグラフのスケールフリー性を活用した最適化

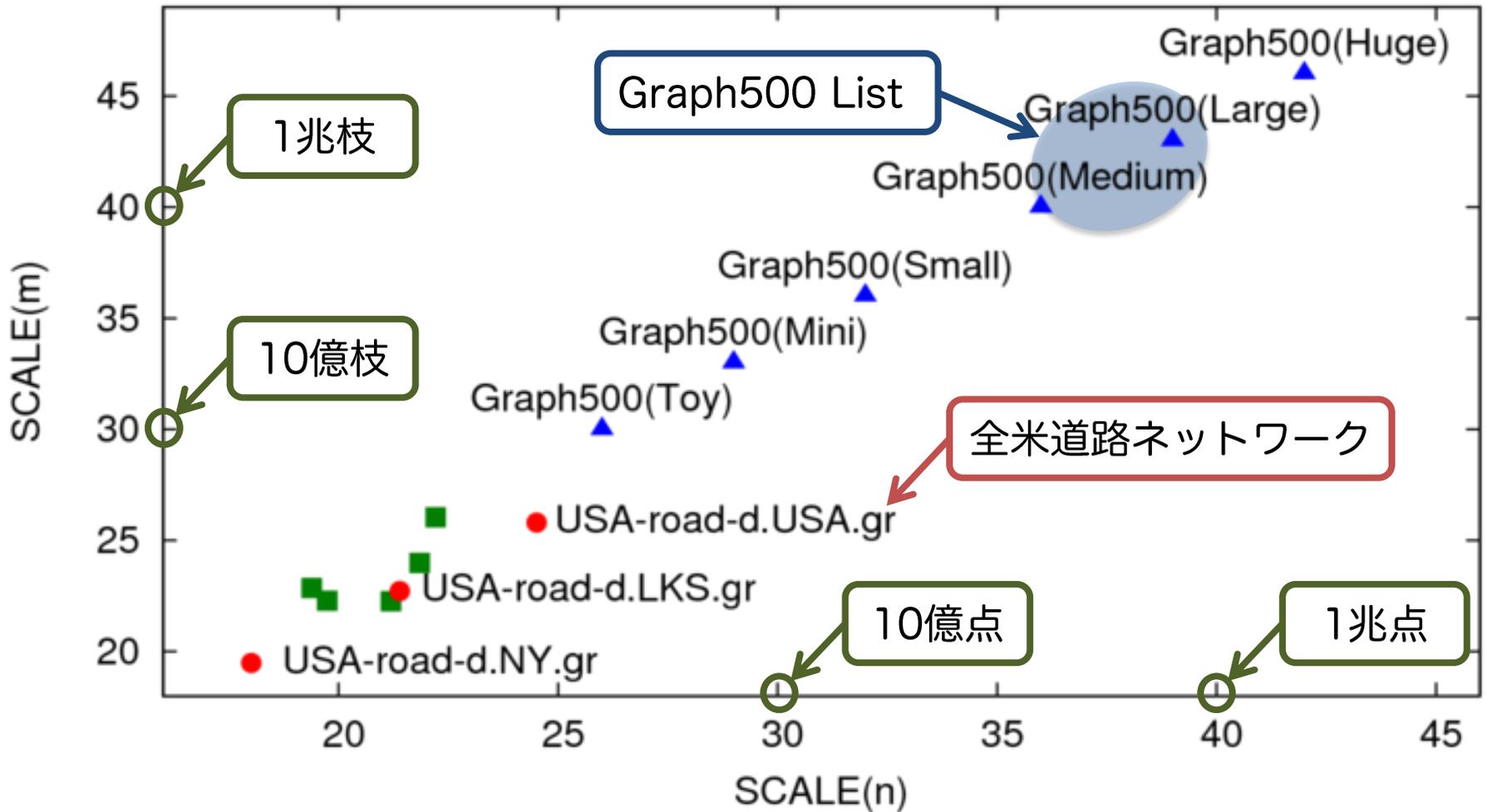
$A_{1,1}^{(1)}$	$A_{1,2}^{(1)}$...	$A_{1,C}^{(1)}$
$A_{2,1}^{(1)}$	$A_{2,2}^{(1)}$...	$A_{2,C}^{(1)}$
⋮	⋮	⋮	⋮
$A_{R,1}^{(1)}$	$A_{R,2}^{(1)}$...	$A_{R,C}^{(1)}$
⋮			
$A_{1,1}^{(C)}$	$A_{1,2}^{(C)}$...	$A_{1,C}^{(C)}$
$A_{2,1}^{(C)}$	$A_{2,2}^{(C)}$...	$A_{2,C}^{(C)}$

2次元分割による最適化

Replicated-csr

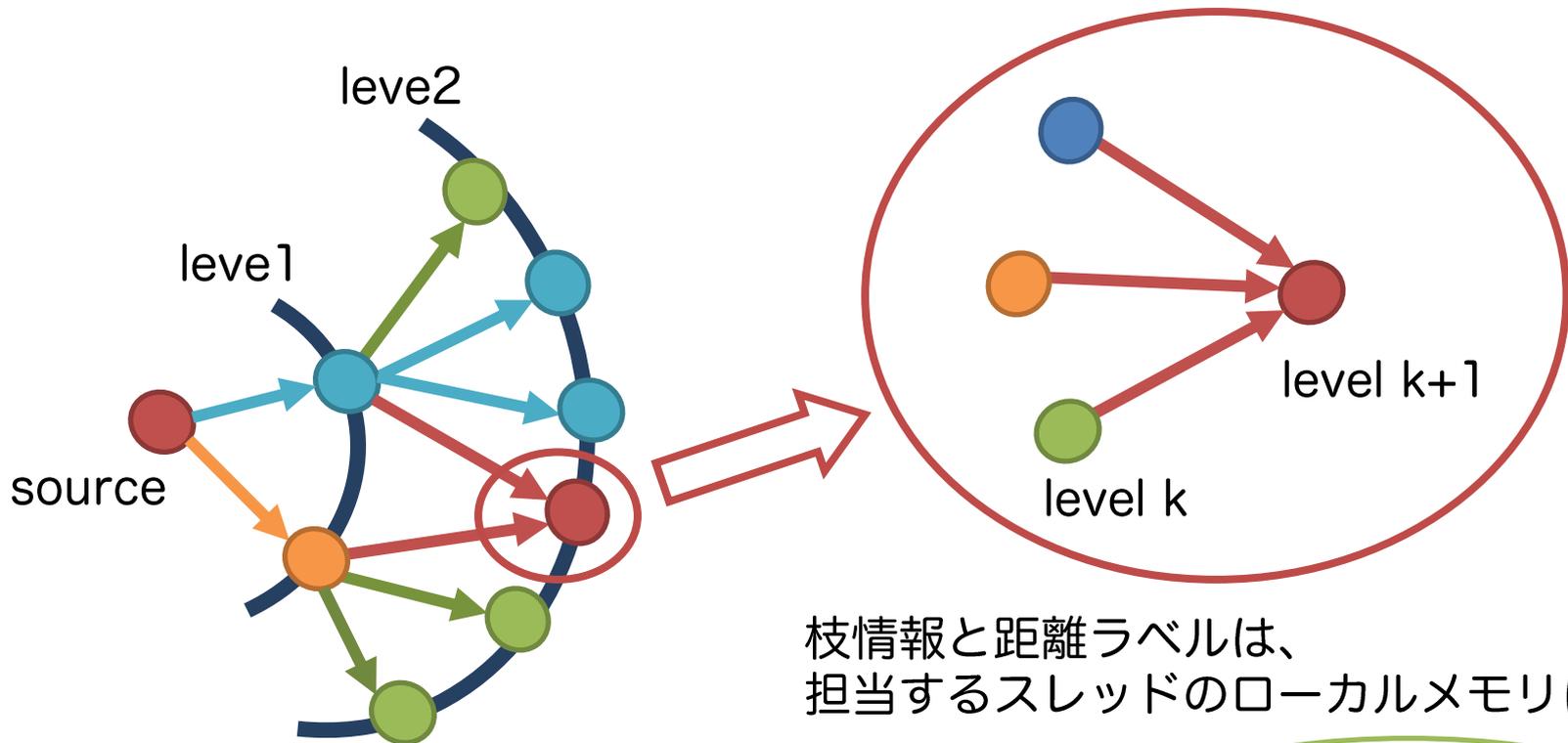


Graph500 List



Parallel BFS frontier extension

- Level Synchronous Parallel w/o relaxation lock



枝情報と距離ラベルは、
担当するスレッドのローカルメモリに配置

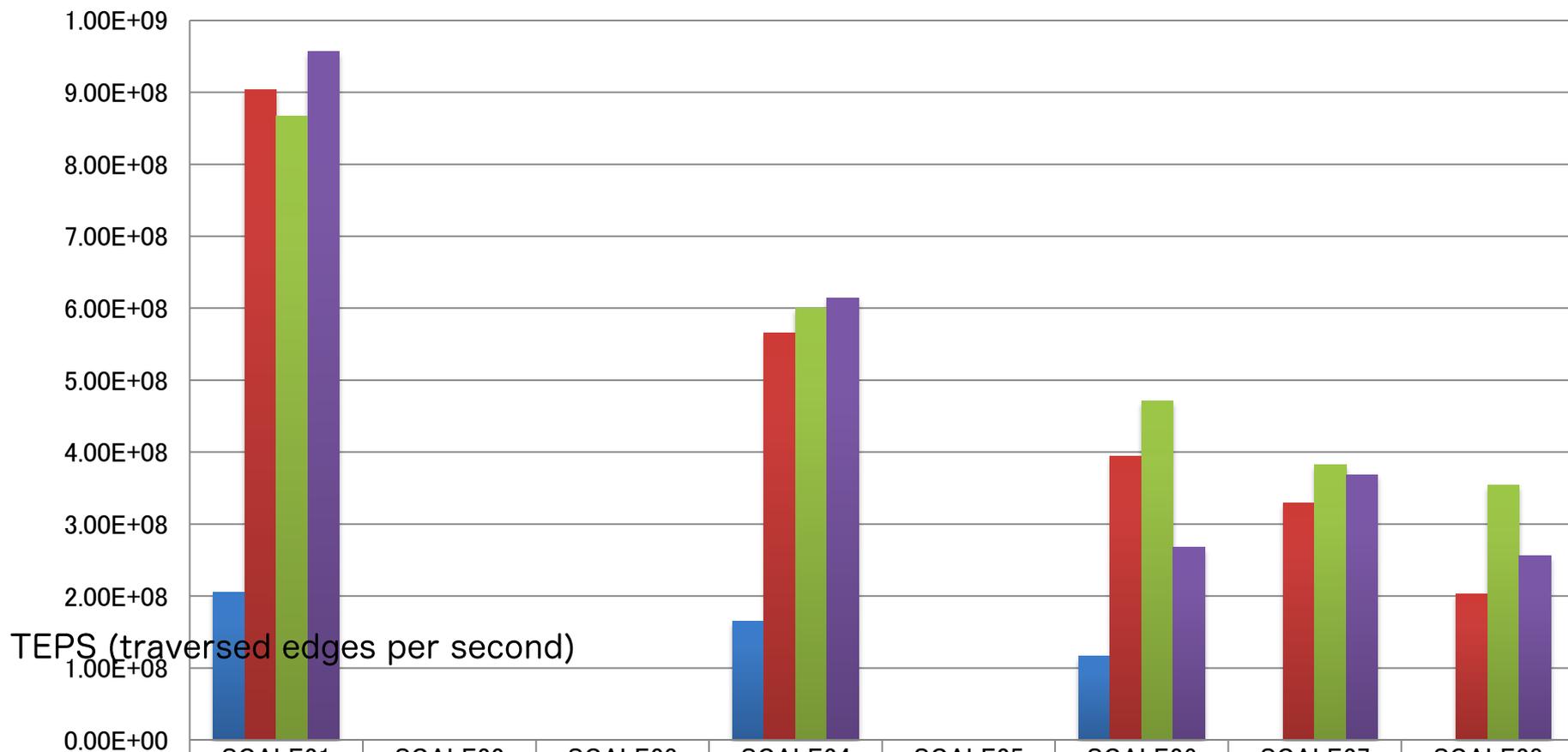
Level 毎に barrier 同期をかける必要あり



※ Level Synchronous BFS において、
距離ラベルは Level によって与えられるため、始点情報は必要ない

SCALE21 ~ SCALE28

- directed graph として処理 → $n=2^{\text{SCALE}}$, $m=2^{\text{SCALE}+5}$



graph500(p=48)	2.04E+08			1.64E+08		1.16E+08		
our bfs (p=36)	9.04E+08			5.66E+08		3.94E+08	3.29E+08	2.03E+08
our bfs (p=42)	8.67E+08			6.00E+08		4.71E+08	3.82E+08	3.54E+08
our bfs (p=48)	9.57E+08			6.14E+08		2.68E+08	3.69E+08	2.55E+08

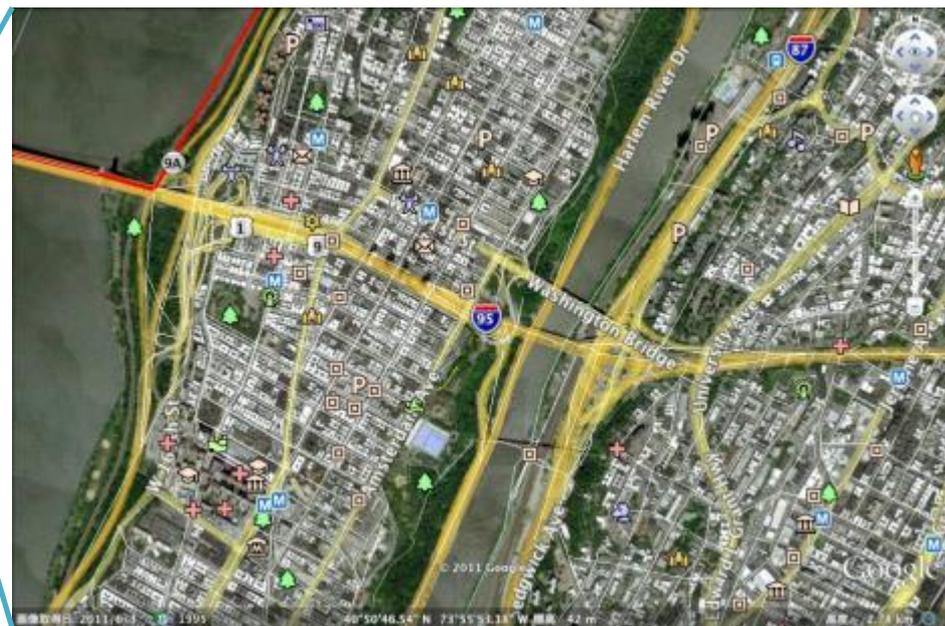
全米道路ネットワークに対する 全対全最短路問題

4-way Opteron 6174 2.3 GHz (12 cores x 4), 256 GB, GCC-4.6.0

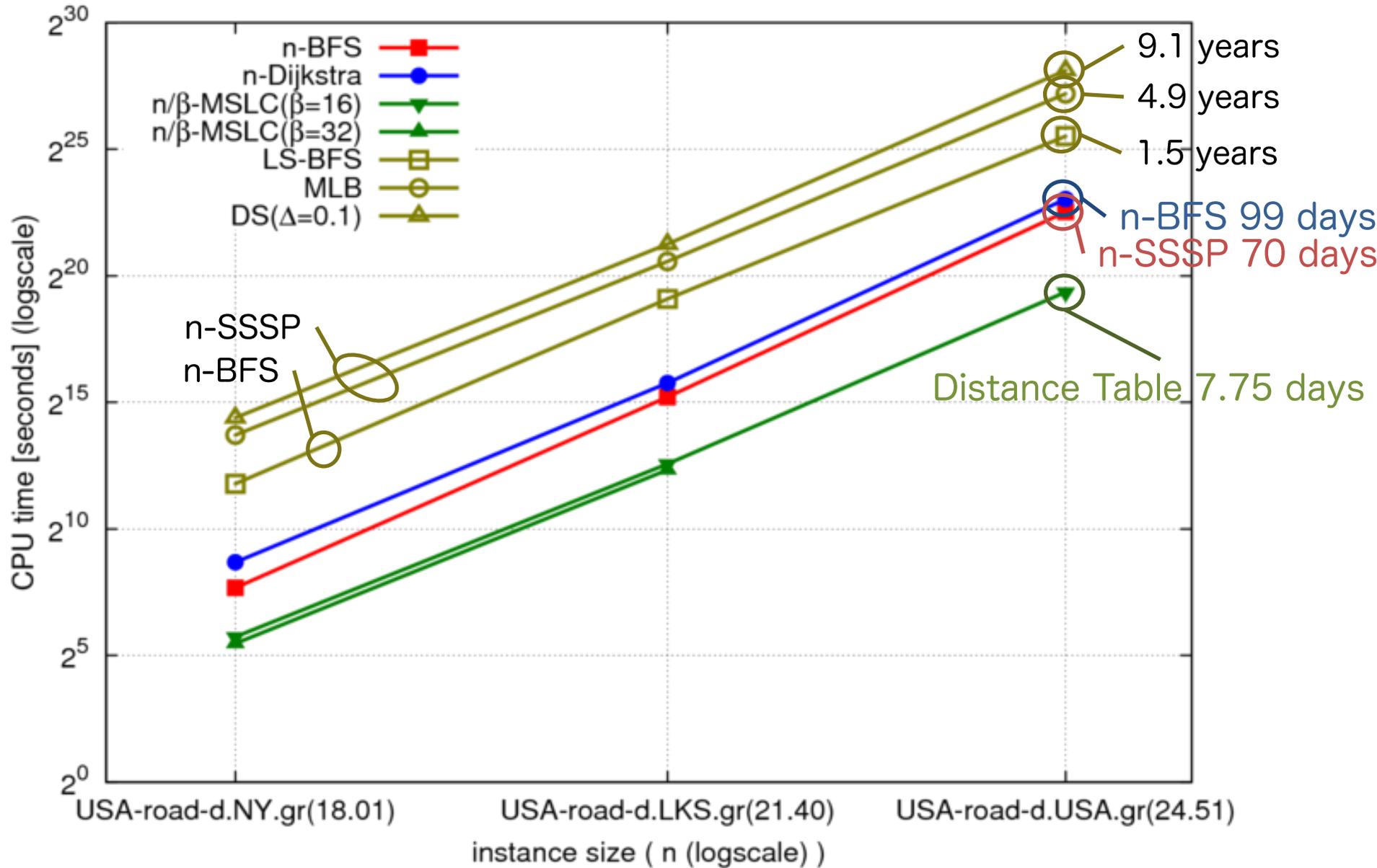
- これまでの方法だと
 1. Δ -stepping algorithm で 9.1 年
 2. Dijkstra's algorithm with Multi-Level Buckets で 4.9 年
- 我々の実装
 - 厳密な distance-table を 7.75 日で求めた
 - MSLC-algorithm を高速化



USA (23,947,347 nodes 58,333,344 arcs)

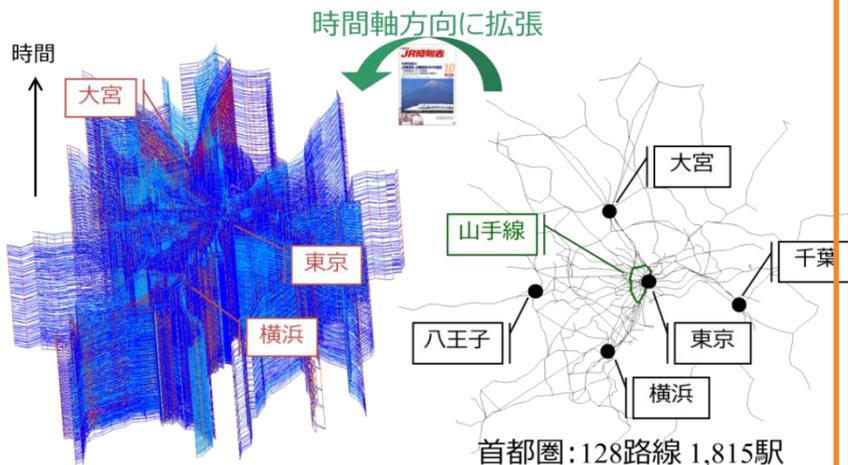


all-pairs shortest paths



実問題に対する大規模グラフ解析

- 超大規模ネットワークに対する探索アルゴリズムとクラスタリングアルゴリズムの高速実装
- 数百万頂点～数兆頂点、数億枝～数百兆枝からなる超大規模なグラフ解析を必要とする。
 - 数百万人の被災者の避難経路の計算では数千万頂点のグラフ(1スレッドあたりのメモリ要求量1Gbytes)に対して、同時に数百万スレッド単位で各被災者毎の最短路計算と各点の重要度判定
- 防災計画の策定、災害時の避難と誘導および情報収集と解析、復興計画の策定、スマートグリッドによる高度かつ安定な電力供給
 - 交通データに対する経路探索動的に変化する交通量等から高速な重要度判定を行い、交通管制等に活用
 - ソーシャルネットワークデータに対する動的な重要度、影響度の判定各点の周辺、および広域内における影響(情報の伝播力)を推定



- 災害時における大規模な物流網および輸送計画の高速デザイン
- 首都国鉄道ネットワーク
 - 時空間ネットワークによる詳細な表現
 - 渋滞・事故情報を考慮した動的データ更新
 - パーソントリップ調査等による帰宅困難者の推定

時空間ネットワークの規模	
ノード	: 746,871
リンク	: 3,053,930
走行リンク	: 507,513
着発間リンク	: 197,863
待ちリンク	: 533,737
待ち合わせリンク	: 20,973
乗換リンク	: 1,793,844

1. 解析対象

twitter	Wikipedia	facebook
2億ユーザ 2億呟き/日	4.7億更新 19更新/頁	7.5億ユーザ 130 friends 300億記事

2. 高次元グラフィック

大規模グラフデータ	10億頂点, 200辺/頂点
クラスタリング > 次元圧縮	1階層, 5億次元 > 3階層, 40,000次元
高次元レイアウトデータ	1.2PB

3. 対話的グラフフィルタリング

検索キーワード

原発 推進 反対

ノード

- 記事
- 議論
- 紹介

クラスタ

関係性

- 同期的編集関係
- 参照関係
- 共著関係

検索対象期間

2011.3.8 ~

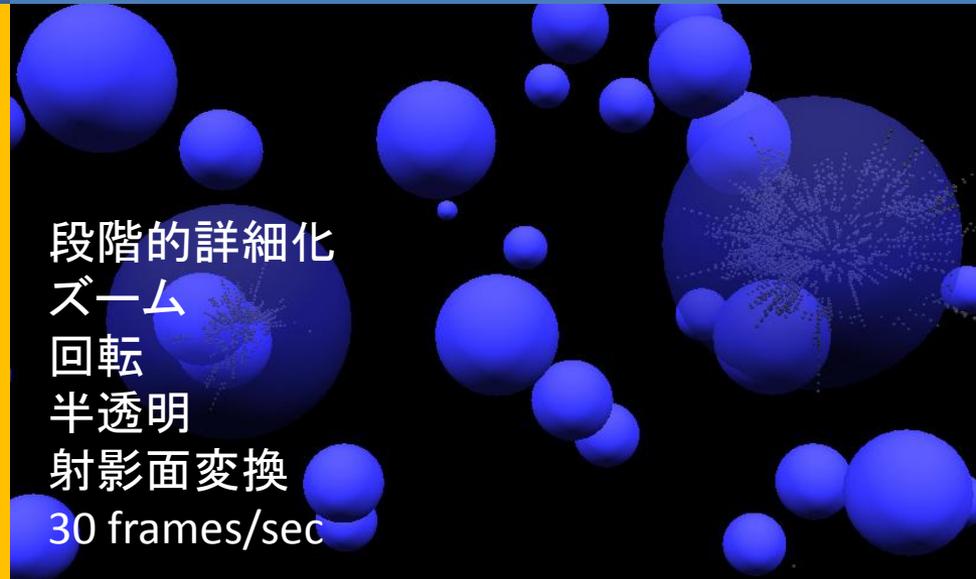
2011.3.25

10億頂点
200辺/頂点

数100万頂点
数10辺/頂点

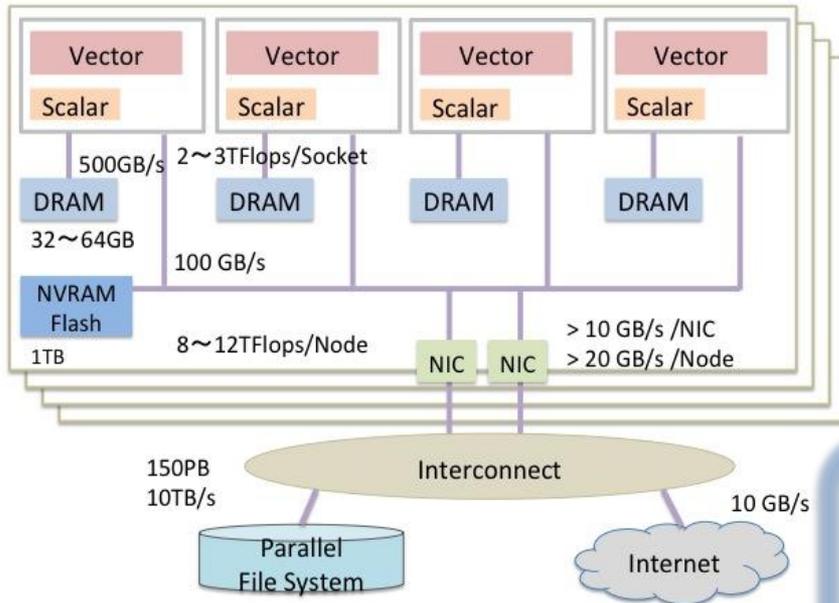
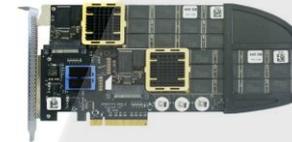
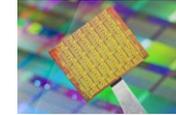
4. 対話処理: 射影演算

段階的詳細化
ズーム
回転
半透明
射影面変換
30 frames/sec



大規模計算機システムの動向

- 並列数の爆発的増大、不均質化、高密度化
 - プロセッサのマルチコア・メニーコア化
 - ヘテロジニアスプロセッサ(ベクトル・スカラ)の台頭
- 記憶装置の多階層化・大容量化
 - 次世代メモリ、Flash/NVRAM, 並列FS
 - ノード内外のデータ転送性能バランスの逆転



100ペタフロップス級スパコンの
想定アーキテクチャ

技術的課題

数千万並列規模の
スケーラビリティ

並列アルゴリズム

局所性の活用

生産性

通信最適化

ストレージの
階層性の深化

不均質性

耐故障

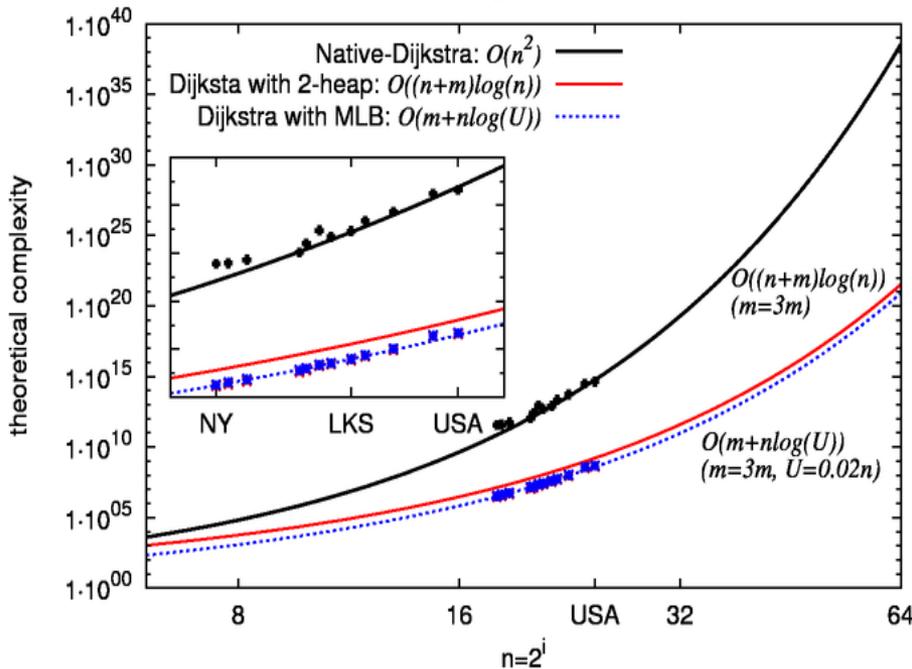
省電力

大規模データI/O

超大規模データに対するグラフ探索アルゴリズム

- ポストペタスケールの超大規模データに対して、最適化やデータ解析等の処理を行う際に、百万コアを用いたとしても、計算量の高いアルゴリズムを適用することは不可能
 - 10億点以上のグラフに対しては、 **$O(n \log n)$ 程度が計算量の上限**となる(下図参照) ただし $\log n$ の部分はヒープ(2分木)に関するものなので、実際に全ての点が同時にヒープに入ることはなく、実際の計算量は小さい
 - 今後はCPUのコア数は増えるが1コアあたりのメモリ量やバンド幅は減少する。アルゴリズムとデータ構造の工夫で**メモリバンド幅の要求量を減らして百万規模のスレッド並列**での高速処理の達成

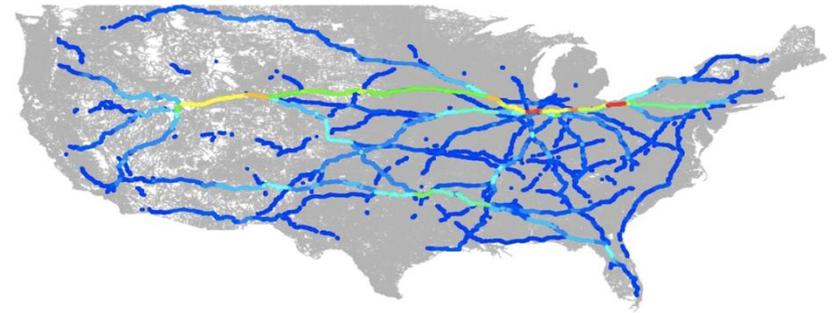
Theoretical Complexity of Single-Source Shortest Path Problems



n = 数千万点の場合:

TEPS(traversed edges per second)値は **1コアで 10^7 = 1000万程度**。よって、全米データ(2400万点, 5800万枝)に対する最短経路計算は5秒程度で終了する。
全対全最短経路問題:

- **1スレッド → 35日以上**
- **百万スレッド → 3分弱 → 実時間解析**



ポストペタスパコン上で超大規模グラフ最適化システム を実現するための技術的課題

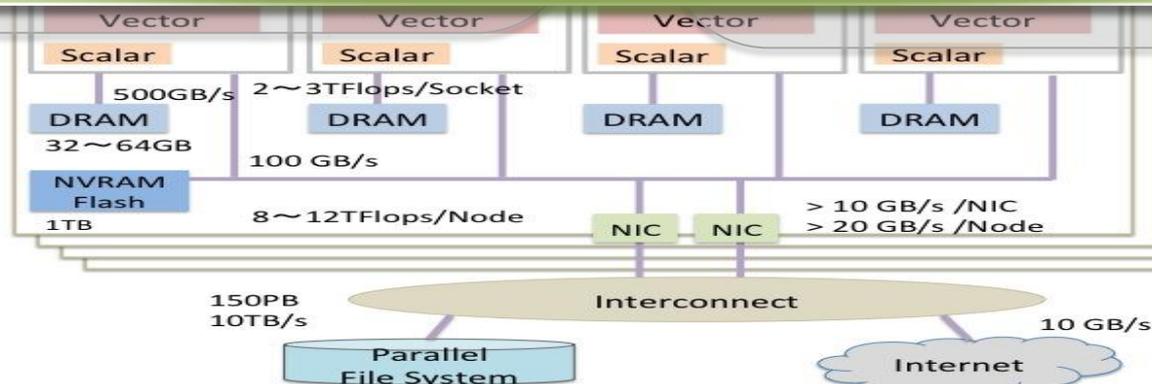
計算量とデータ量の規模が大きく従来手法では解決困難

大規模グラフをどのように可視化すべきか？

大規模グラフストリームをリアルタイム
に処理するには？

ポストペタスケールシステム上で
大規模なグラフ処理を高性能かつ
効率的に実現するには？

超大規模グラフをどのように格納すべきか？



超大規模グラフ最適化システムの提案

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・省電力

ソーシャル
ネットワーク解析

超大規模グラフ最適化システム

大規模グラフ可視化

リアルタイム大規模グラフ
ストリーム処理

グラフ最適化ライブラリ
による大規模グラフ処理

中心性
解析

最短経路
問題

最速
フロー問題

Indexing

クラスタ
リング

半正定値
計画問題

混合整数
計画問題

リアルタイム
ストリーム処理系

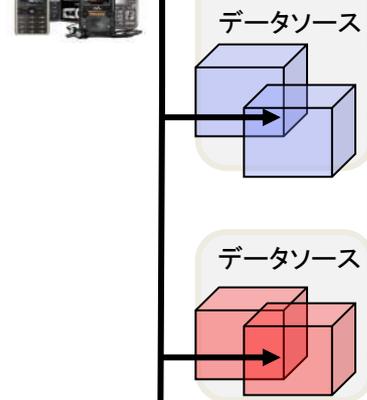
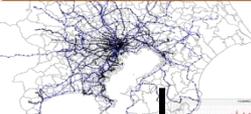
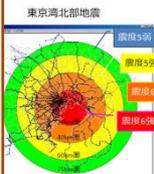
X10言語処理系

100PF級ヘテロジニアススパコン

大規模グラフストア

超大規模センサー

- 観測データ
- スマートグリッド
- 交通・輸送
- SNS (Twitter)



ポストペタスパコン上で超大規模グラフ最適化システム を実現するための技術的課題

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・省電力

ソーシャル
ネットワーク解析

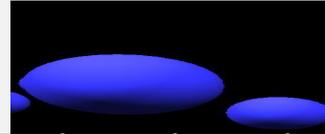
超大規模グラフ最適化システム

大規模グラフ可視化

- 一般的に時間的・空間的計算量大
- 動的変化を伴う大規模グラフの可視化
- ペタバイト級の高次元レイアウト配置

超大規模センサー
• 観測データ

東京湾北部地震
震度5弱



大規模

大規模グラフ最適化アルゴリズム

- 数十億点以上の超巨大ネットワークデータに対するグラフ探索(最短路計算、幅優先探索等)を高速に実現するアルゴリズム → $O(n \log n)$ 程度が計算量の上限
- グラフ探索のみならず半正定値計画問題, 混合整数計画問題による高度なグラフ最適化技術が不可欠
- ベクトル・スカラープロセッサから成る不均質かつ分散共有メモリから成る大規模並列環境において高性能なグラフ最適化を実現する必要あり

大規模グラフ処理基盤

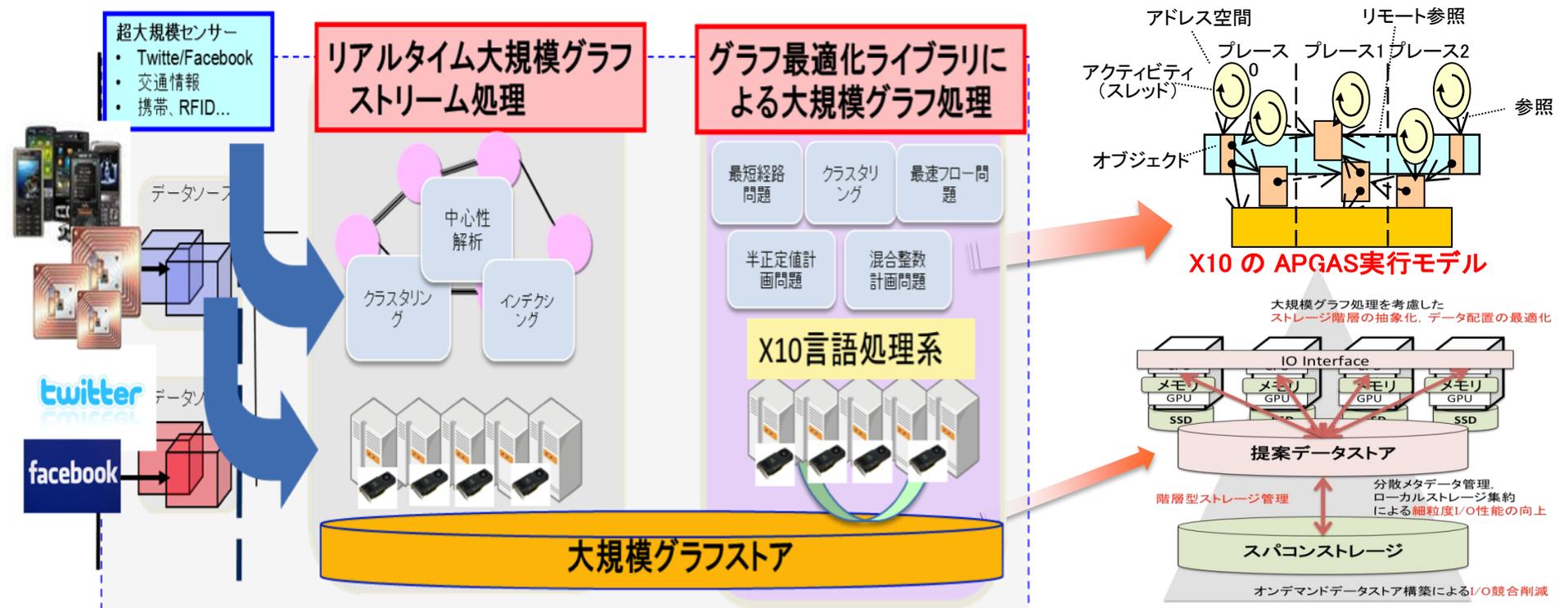
- ポストペタスパコン上における汎用的かつ高速グラフ処理を提供するグラフ処理ライブラリが必要
- 大規模センサーからのグラフストリームに対してスケラブルなリアルタイム処理を実現するシステムソフトウェアが必要

大規模グラフ格納基盤

- ペタバイト級の大規模グラフデータを格納
- 高速なグラフ解析に不可欠なランダムアクセスや細粒度I/O処理を提供
- 階層的なストレージ(次世代メモリ, Flash/NVRAM, 並列FS)を活用した高速I/O
→ 既存の分散ファイルシステムでは実現困難

ポストペタスケールシステムにおける大規模グラフ処理・格納基盤

1. **大規模グラフ最適化ライブラリ**: ベクトル・スカラープロセッサから成る不均質でかつ分散共有メモリから成る大規模並列環境における大規模グラフ処理を PGAS 言語 X10 を用いて一元的に記述。ポストペタスケールシステムにおいてスケーラブルなグラフ処理を実現するため、アルゴリズム・データ構造の改良と共に X10処理系の拡張・性能最適化 も行い、数百万スレッドまでスケールさせる
2. **リアルタイム大規模グラフストリーム処理**: 超並列分散環境上において通信と計算のバランスを考慮したデータフローの分割、ベクトル・スカラープロセッサの活用、効率的なグラフストリーム処理アルゴリズム などの手法を用いて、数百万～数十億のセンサーデータ に対して 数百万スレッド を用いて リアルタイム処理を実現
3. **大規模グラフストア**: 階層型ストレージ を活用した高速かつ 巨大グラフデータ(数ペタ～数百ペタバイト) を格納し得る方式を実現。オンデマンドデータストア構築による高速アクセスの実現 (I/O競合の削減)



X10(エックステン)

X10は、IBM Researchが開発している

新しい並列分散プログラミング言語です

- **並列・分散アプリケーションの生産性向上が目標**

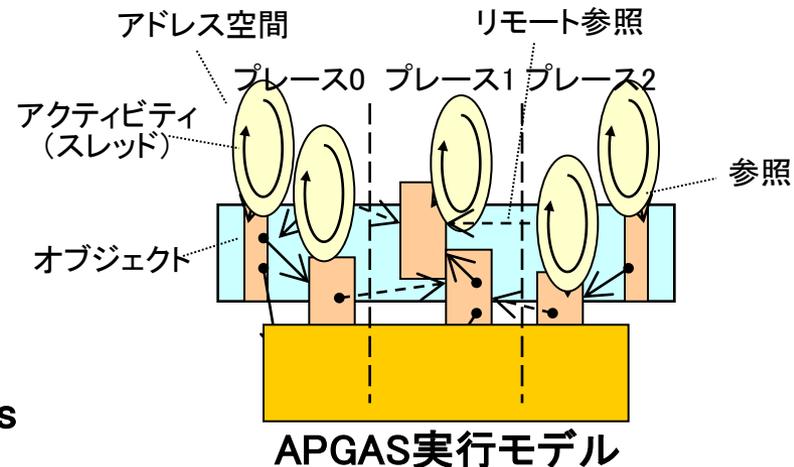
- ・ マルチコアSMPチップやGPGPUなどが相互接続された並列分散環境に対する、スケーラブルなプログラミングを可能にします

- **APGAS (Asynchronous Partitioned Global Address Space) 実行モデルにもとづくプログラミング**

- ・ 複数のマシンを、分割されたグローバルメモリ空間 (Places)として統一的に管理可能
- ・ 軽量の非同期アクティビティを生成可能
- ・ リモートPlaceへのアクティビティやオブジェクトの生成および参照をサポート

- **多様な実行環境をサポート**

- ・ Java環境およびネイティブ環境上で実行可能
- ・ Eclipseに統合された開発ツール群を提供



```
class MyHello {  
    public static def main(Array[String]) {  
        finish for (pl in Place.places()) {  
            async at (pl) Console.OUT.println(  
                "Hello World from place " +  
                here.id);  
        }  
    }  
}
```

```
$ x10c MyHello.x10 # コンパイル  
$ x10 MyHello # 実行  
Hello World from place 3  
Hello World from place 0  
Hello World from place 1  
Hello World from place 2
```

X10によるHello Worldと、その実行例

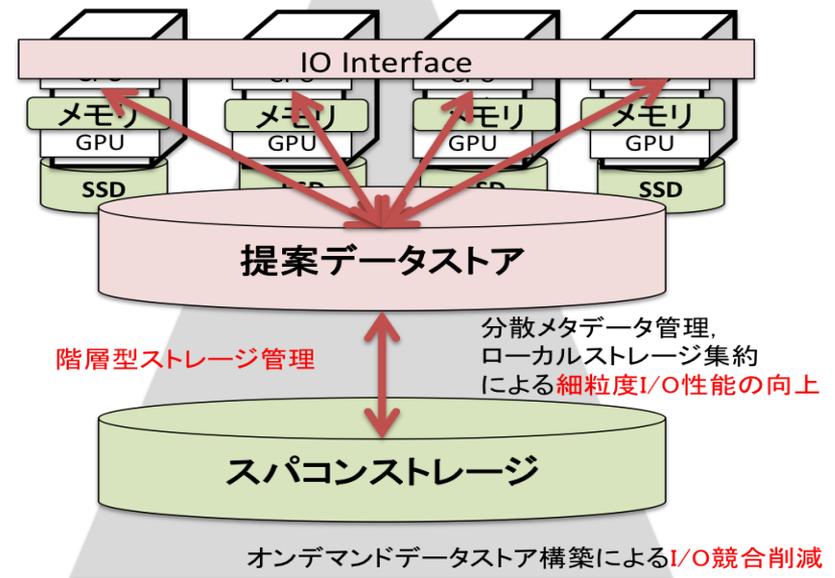
大規模グラフ処理向け階層型データストア

- 大規模グラフに特化したI/Oインターフェース・DB
 - 頂点・辺情報への高速アクセス
 - 隣接頂点・辺情報の**ダイレクトポインタ**
 - アルゴリズムに応じた**データ配置の最適化**
 - ストレージ階層の**抽象化**
 - 次世代メモリ, Flash/NVRAM, 並列FS (Lustre, GPFS, etc.)
- 階層型ストレージ管理
 - スパコンスケジューラと協調し**オンデマンド**にデータストアを構築
 - ローカルストレージ (Flash/NVRAM) の集約
 - 参照データのみローカルストレージに保持
 - I/Oの分離によるスパコンFS全体に対する**I/O競合を解消**
- 細粒度I/O性能の向上
 - 分散メタデータ管理による**IOPS向上**, **メタデータアクセス遅延の削減**
 - ノード内Flash/NVRAMの集約による**スループットの向上**

オンデマンドにノードを集約し
合算により性能向上
(10Kノード規模)

- メモリ帯域を最大限活用
(ノードあたり**0.2TB/s~1TB/s**)
- ファイルのストアはローカルFlashへ
(ノードあたり**1M~1G IOPS**)

大規模グラフ処理を考慮した
ストレージ階層の抽象化, データ配置の最適化



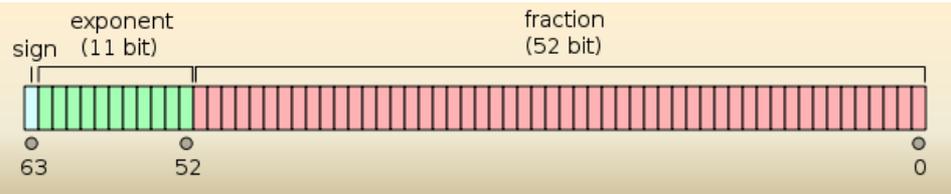
- 2⁴²頂点**, **数PB以上のワーキングデータセット**
- 現実的には属性情報や複数グラフデータを格納するので**数十~数百PB規模のストアが必要**

超大規模最適化問題に対する高速計算システムの構築と評価

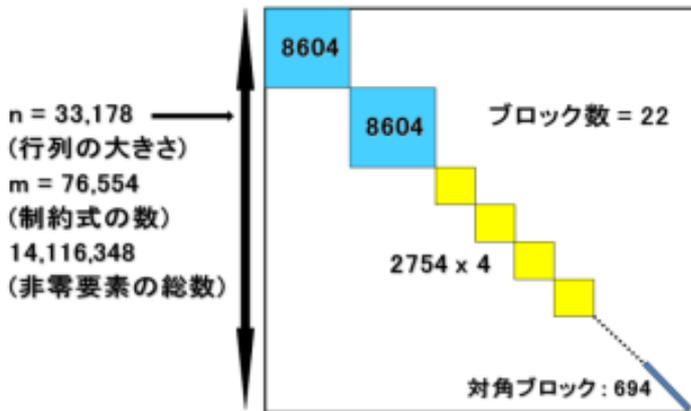
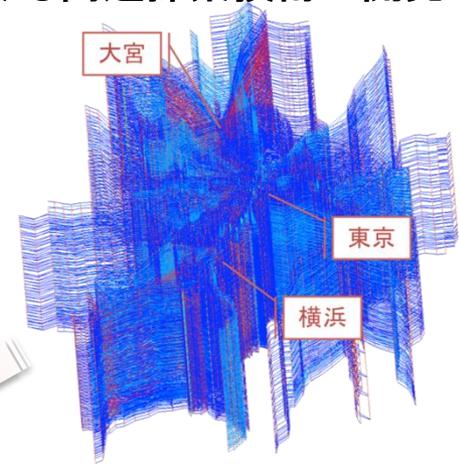
- **世界最高レベルの性能を持つ最適化ソルバー**(グラフ探索(最短路、幅優先等), 半正定値計画問題(SDP), 混合整数計画問題(MIP)) の開発
- グラフ問題等の良質な近似解を高速に探索する高性能最適化基盤の開発
- 高精度線形代数演算ライブラリ: 線形代数演算は問題規模が大きくなれば数値的に不安となるので, 高精度ライブラリをアクセラレータにより高速化する

高精度線形演算ライブラリの開発と高速化

超大規模ネットワークにおける高速探索技術の開発



数千万頂点のグラフ
(1スレッドあたりのメモリ要求量1Gbytes)
に対して、同時に数百万スレッド単位での最短路計算

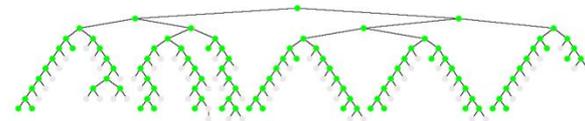


巨大 SDP のデータ構造

疎性の追求と前処理さらに並列計算の適用
(大規模スレッド並列 & CPU + GPU による高速化)変数と制約条件 100 万以上

スーパーコンピュータ
上の大規模グラフ処理
基盤での実行

大規模グラフデータ
数理計画問題
並列化支援
高精度量子化学計算



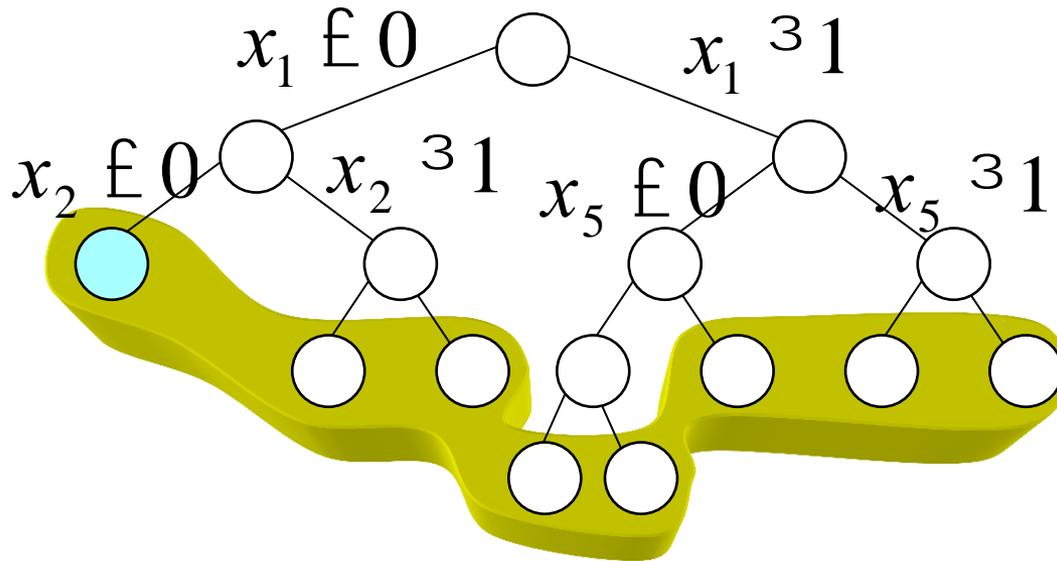
前処理による変数の削除と並列計算の適用
(大規模スレッド並列) 整数変数10万以上

IP に対する分枝操作

スーパーコンピュータ上での最適化問題の実行

- どのような最適化問題（アルゴリズム）がスーパーコンピュータでの実行に適しているのか？
 1. 解くべき問題の規模が非常に大きいこと
 2. アルゴリズムの演算量がある程度大きいこと
 3. 演算量データ移動量の比が十分程度大きいこと
 4. 並列計算数に合わせて演算量やデータ移動量がほぼ均等になるように各プロセスに割り振ることができること
 5. 並列に動作しているプロセス同士が高速かつ高い信頼性で頻繁に通信を行ったり、同期を取る必要がある場合
- 上記の条件を（ほぼ）満たす最適化問題
 - SDP, MIP, TSP など
- 計算量とデータ移動量の正確な推定を行う。またデータの特性(疎性, サイズ)と性能値の関係を見極める

MIP solvers: LP based Branch-and-cut algorithm



Subproblem
(sub-MIP)

Subproblems (sub-MIPs) can be processed independently

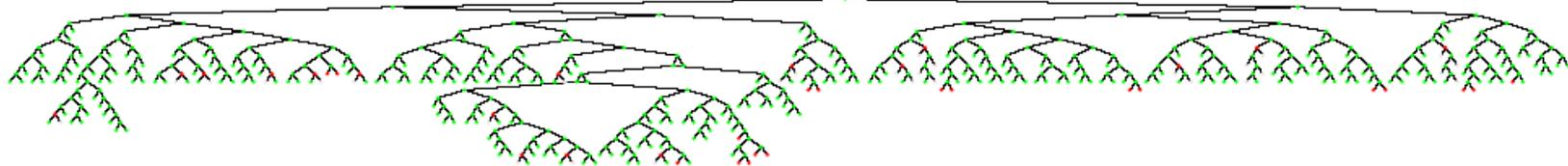
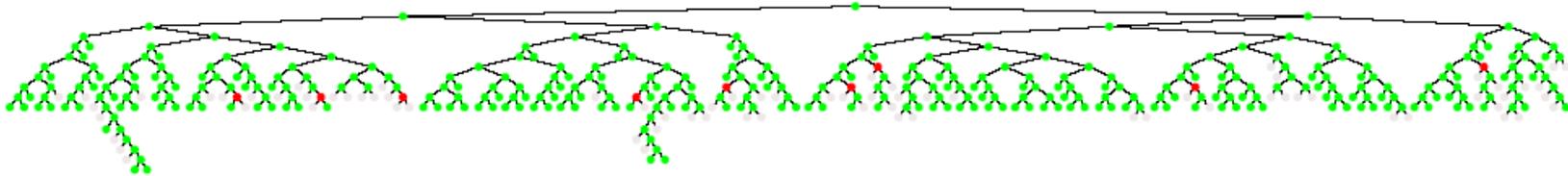
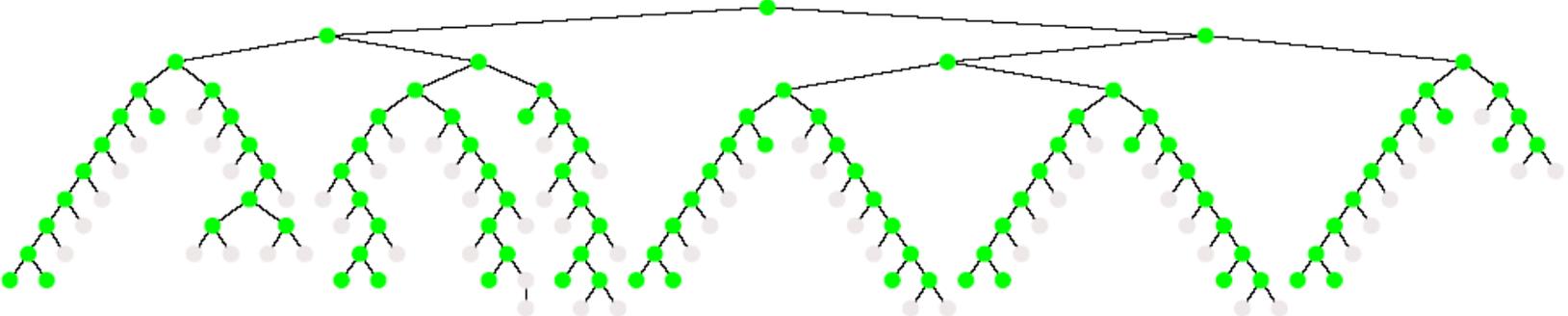


B&C is suitable for parallelization

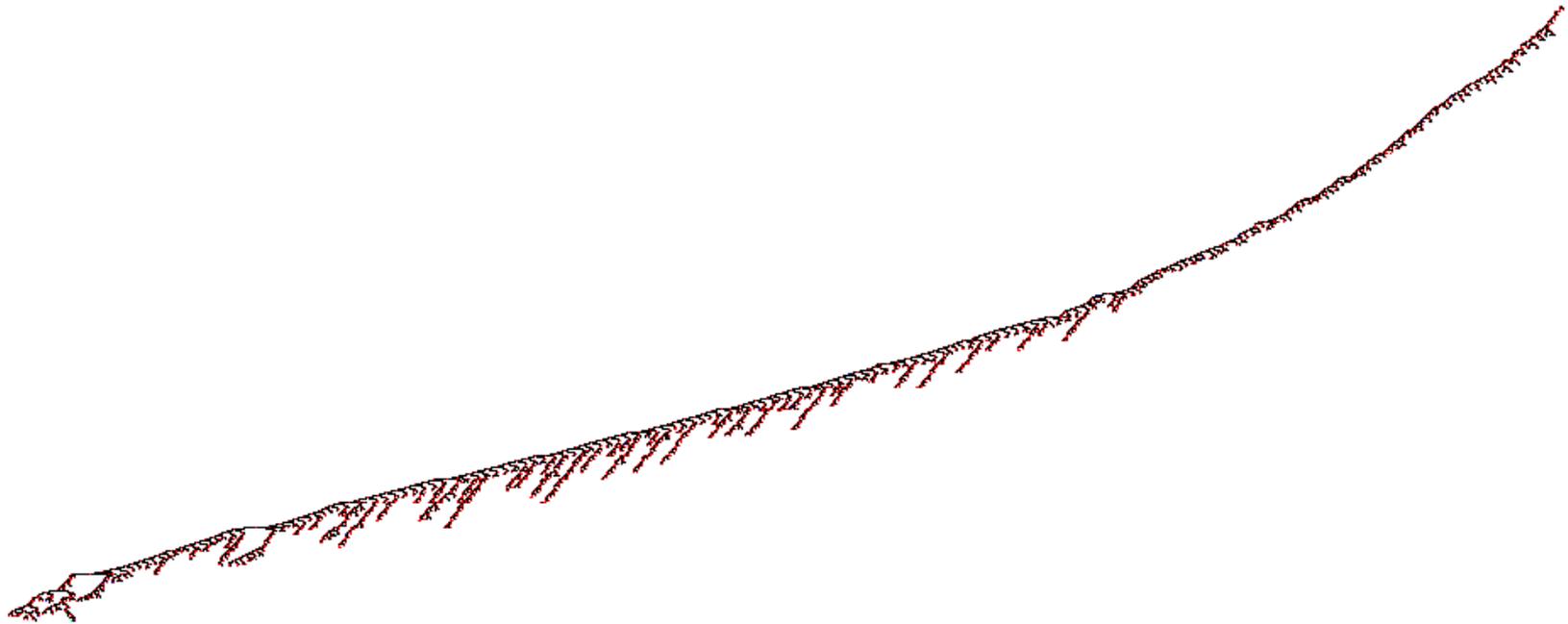
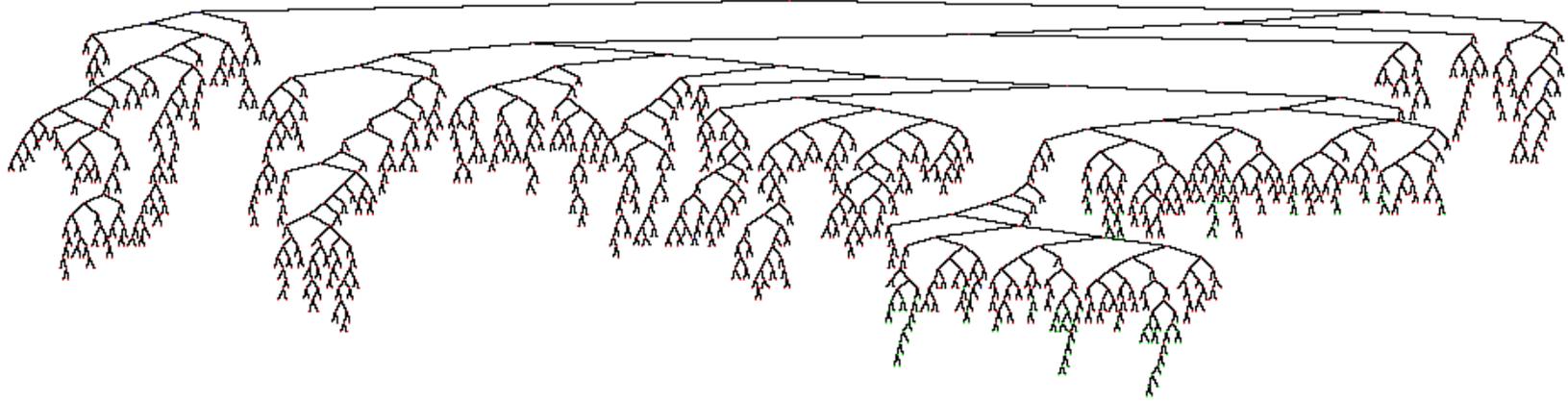


Utilize a large number of processors
for solving hard problem instances

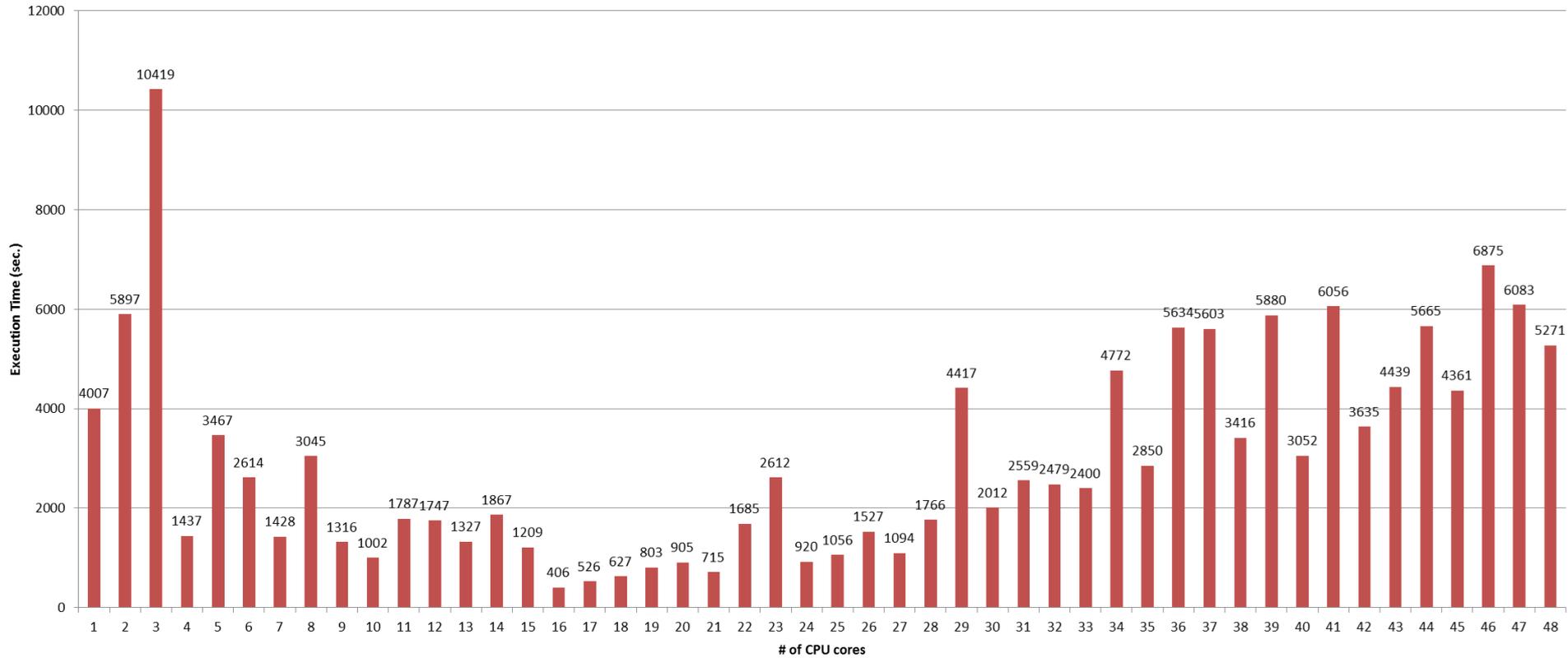
Ramp-up: Branching Trees



Ramp-up: Branching Trees



roll3000.mps : MIPLIB2010



CPU : AMD Opteron 6174(2.2GHz) x 4

→ 48 コア

RAM : 256GB

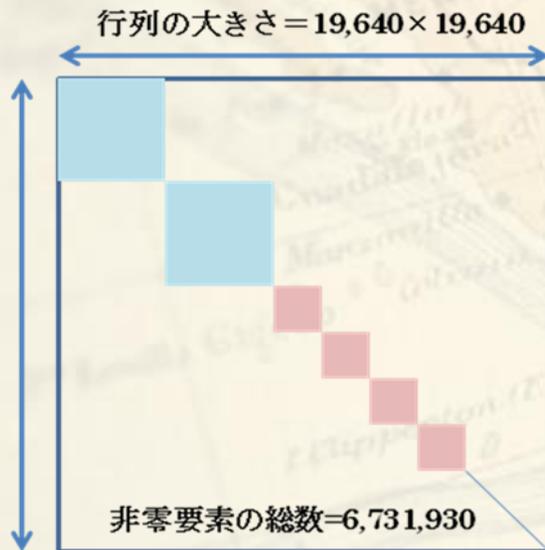
OS : Fedora 15

京大T₂K スパコンでの SDPARA の大規模実行 (量子化学)



CPU	AMD Opteron 8356 2.3GHz x 4
メモリ	32GByte
NIC	GbE x 2 & Infiniband x 4

京大 T2K オープンスパコンとスペック



量子化学分野の超大規模SDPIに対する計算時間

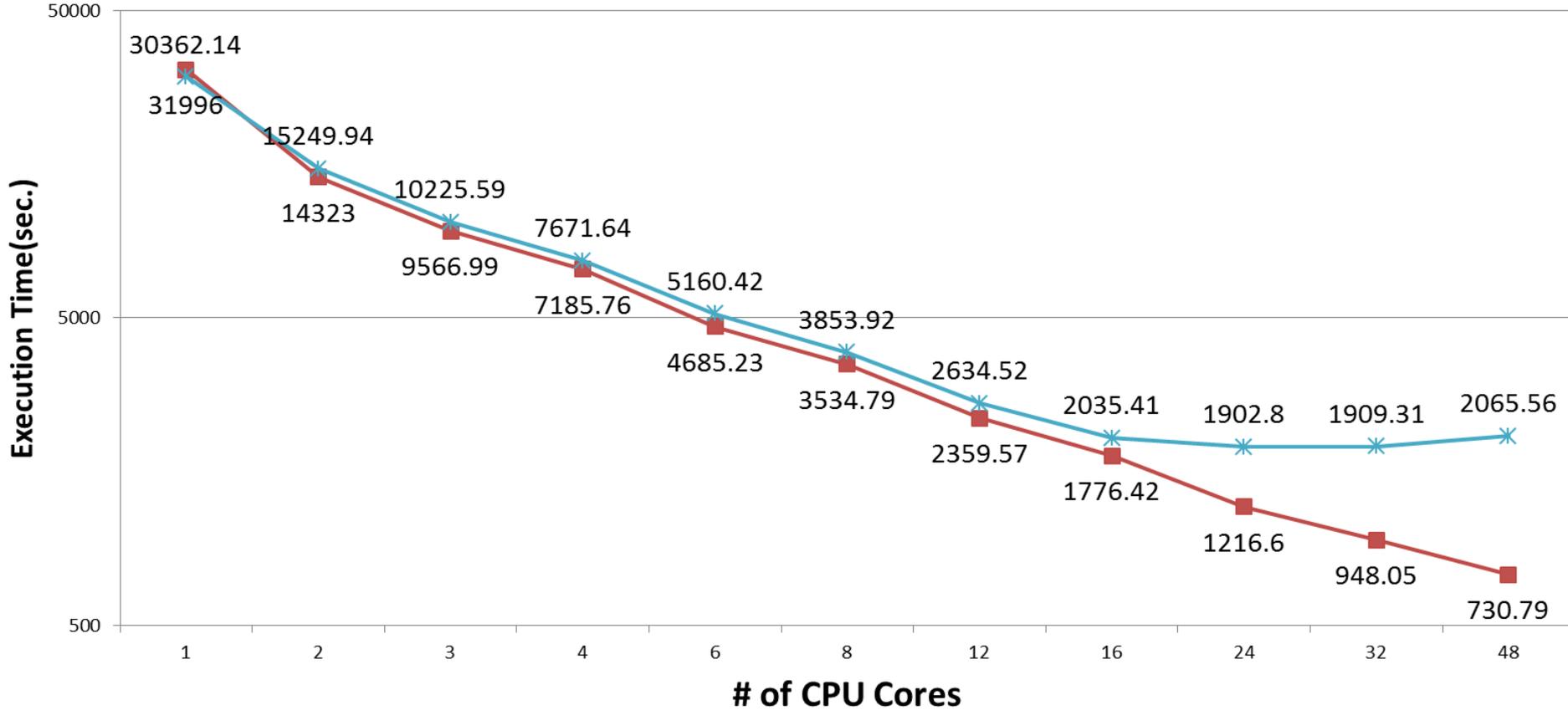
問題名	総計算時間(秒)
H ₂ O	27,523.8秒 = 7.6時間
CH ₃	68,593.4秒 = 19時間
NH ₃	72,025.6秒 = 20時間
O ₂	5,943.1秒 = 1.6時間

128ノード × 16 コア = 2048コアによる
超大規模計算

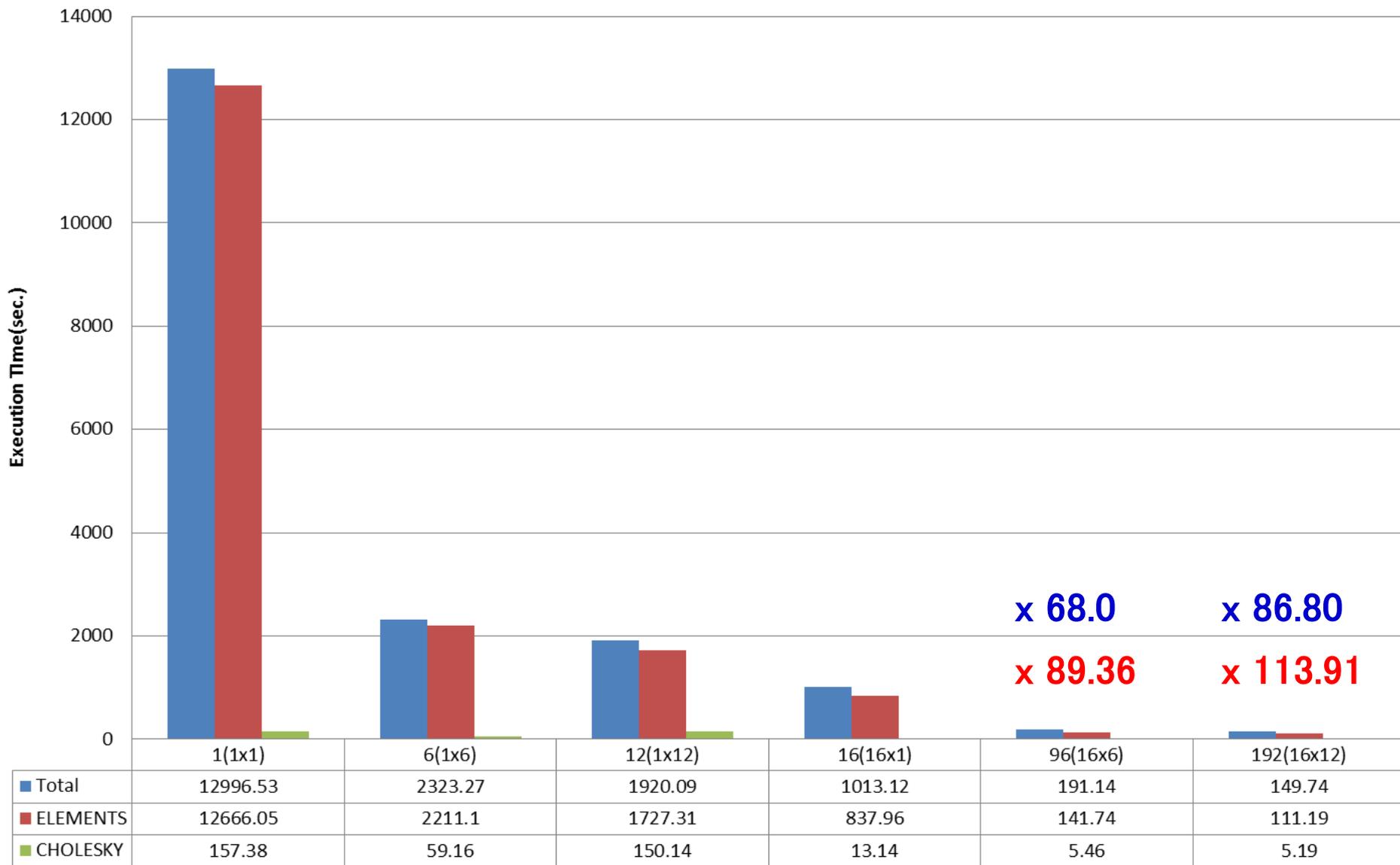
今回計算に成功した超巨大なSDP

Total

■ SDPA * CSDP



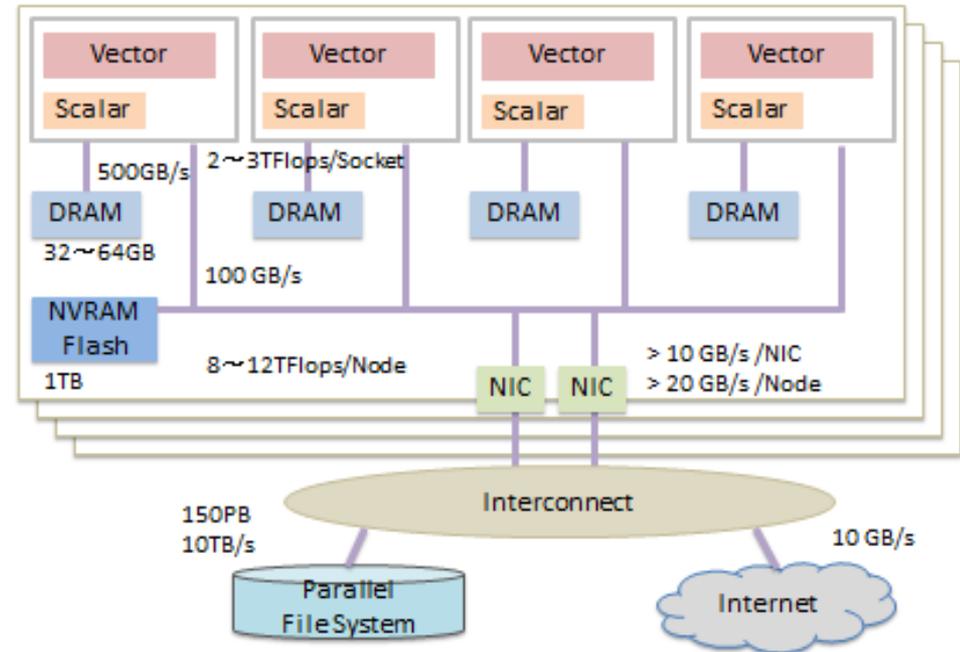
SDPARA 7.3.3 on Intel Xeon X5670 (16node x 12cores)



Problem: Be.1S.SV.pqgt1t2p.dat-s

Post peta-scale supercomputer

- Each CPU has **Vector** and **Scalar** Processing Units
- 10,000 ~ 25,000 nodes in one supercomputer
- Total performance 100 ~ 300 Pflops
- SDPARA on peta-scale supercomputer
 - Generating SCM : **Scalar**
 - Cholesky Factorization of SCM : **Scalar** and **Vector**
 - We can solve very large-scale SDPs:
 - n and $m \gg 1$ million



計算量

大



小

上層: NP 困難な最適化アルゴリズムの実行

1. MIP(混合整数計画問題)の場合: 0-1 整数変数の数 = n で計算量はおよそ $O(2^n)$
2. 前処理による変数の削除と並列計算の適用 (CPUコア中心の大規模スレッド並列: 分枝カット法の適用)
3. データ量は 10^5 以下 (整数変数の数)
4. 施設配置問題、集合被覆 (分割) 問題、スケジューリング問題などの最適化問題

中層: 多項式時間最適化アルゴリズムの実行

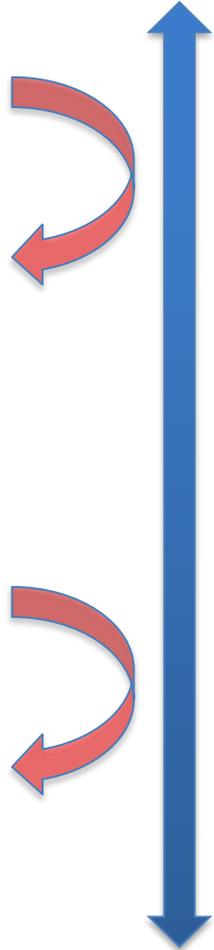
1. SDP(半正定値計画問題)の場合: n = 行列の大きさ, m = 制約式の数で計算量はおよそ $O(n^3 + m^3)$
2. 疎性の追求と前処理さらに並列計算の適用 (CPUコア中心の大規模スレッド並列が中心だが、今後は CPU + GPU による高速化)
3. データ量は 10^8 以下 (非負変数の数)
4. グラフ分割、センサーネットワーク、サポートベクターマシンなどの最適化問題

下層: グラフ解析アルゴリズムの実行を想定

1. ダイクストラ法 (1対全最短経路問題: ヒープ付き) の場合: n = 点数, m = 枝数で計算量はおよそ $O((n + m) \log n)$
2. グラフ探索の局所的な評価では優先キュー (ヒープ木) を用いる \Rightarrow 実行時間、メモリ消費量が安定的 (CPUコア中心の大規模スレッド並列が中心だが、高速ストレージ技術による超大規模グラフ処理)
3. データ量は 10^{11} 以下 (グラフの点数と枝数)
4. 最短経路計算、ネットワーク内で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

データ量

小

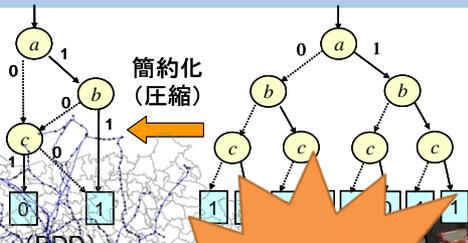


大

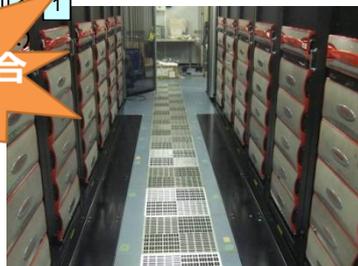
まとめ

- ポストペタスケールスパコン上での高性能な**超大規模グラフ処理技術**の達成
 - リアルタイム大規模グラフストリーム処理基盤
 - 高速グラフ最適化ライブラリ・ソルバー
 - 大規模グラフストア
 - 大規模グラフストリームデータの対話型閲覧システム
 - **最重要カーネル**のひとつである超大規模グラフ処理の研究開発を通じて**ポストペタ・エクサスケールスーパーコンピューティング**に向けた要素技術の実現に大きく貢献
 - 数千万規模の並列性・不均質性の克服
 - 計算量・データ移動の最適バランスの実現
- ⇒ 研究開発を通じて得られた知見は他のカーネルへ適用・応用

最先端理論(Algorithm Theory)



有機的な融合



超大規模実データ(Practice)

最新計算技術(Computation)

HPCによる安心安全な社会基盤の実現

- 応用例
 - ✓ 防災計画の策定
 - ✓ 災害時避難と誘導及び情報収集と解析
 - ✓ スマートグリッドによる高度かつ安定な電力供給
 - ✓ ソーシャルネットワーク解析
- 新しいスパコン利用法
 - ✓ 災害時に稼働可能(免震、独立電源)な状態で整備
 - ✓ 緊急時には災害対策プログラムが稼働 → 国民の生命財産の保護