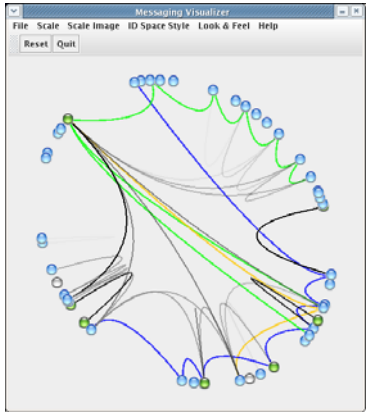
分散システムの大規模シミュレーション

首藤 一幸，華井 雅俊，杉野 好宏

東京工業大学

分散システムの大規模シミュレーション

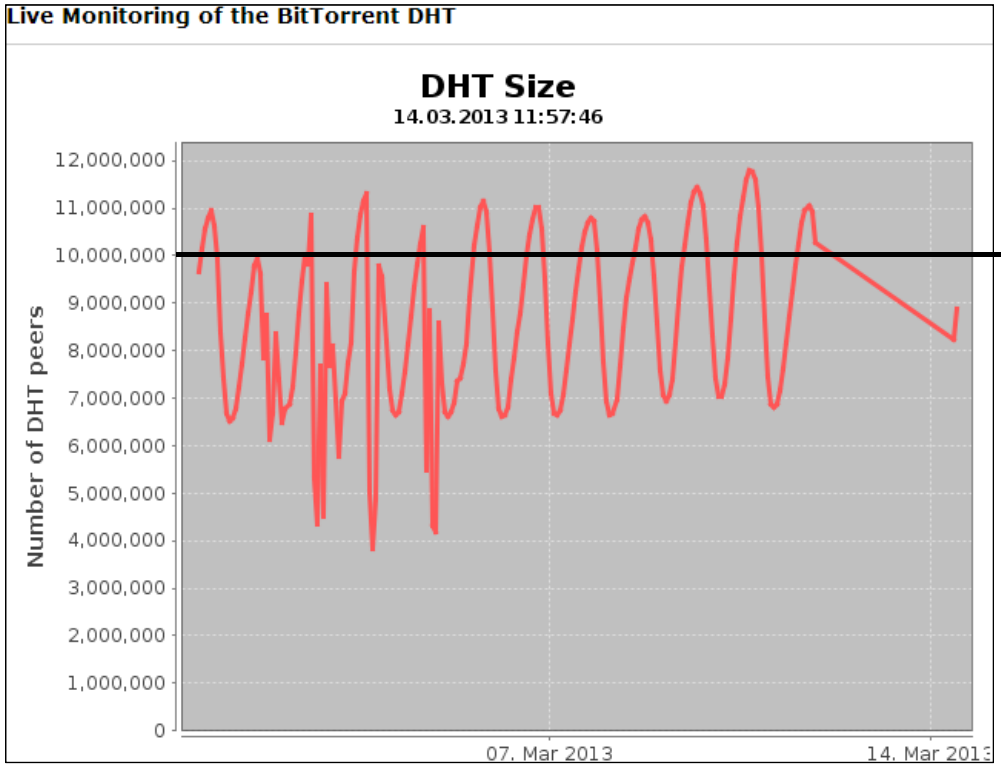
- 自律分散システム / peer-to-peer を **数十億**ノード規模でシミュレートする。
  - 貢献：シミュレータ構成法
- これまで
  - PC 1台で 200万ノード (Chord などの DHT アルゴリズム)
    - SimGrid (INRIA), Overlay Weaver (首藤)
- 現時点
  - PC 10台で 1 億ノード (Gnutella の flooding)
    - 我々のシミュレータ (杉野, 華井, 首藤)



Overlay Weaver

研究対象の規模

- 今日の分散システム：１千万ノード



BitTorrent DHT

１千万

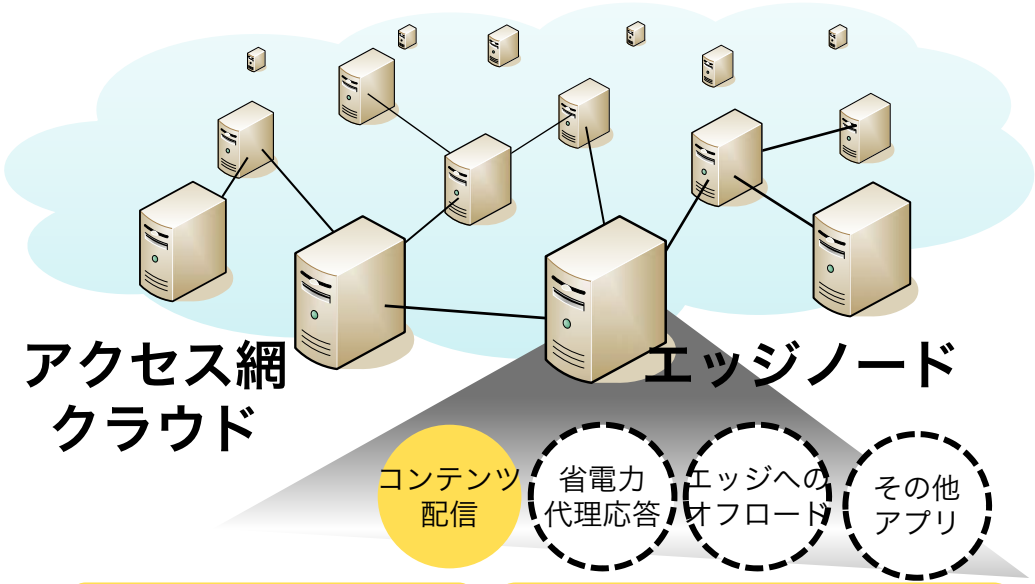
- 実験できない！ → ネットは大丈夫？
- 今後： Internet of Things, M2M で数百億

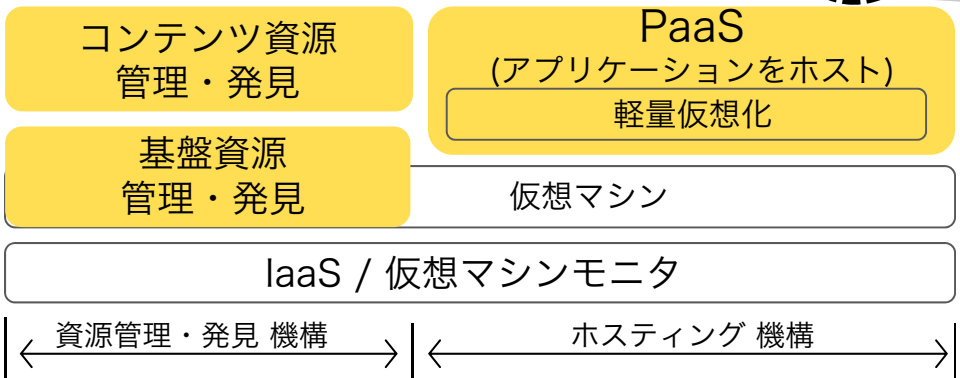
研究対象の規模 (cont'd)

- DYPOP: Datacenter beside Your telco-POP
  - 2011〜2012年度 NICT 受託研究。理研 (小林), IIJ, 東工大 (首藤)。
  - アクセス回線を集約する局舎にサーバを置く。DCを分散化。

光のアクセス回線数
２千万

コンテナ DC（写真：あきみちさん）

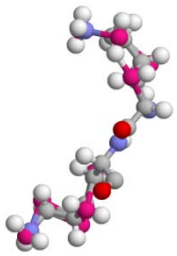




シミュレーション

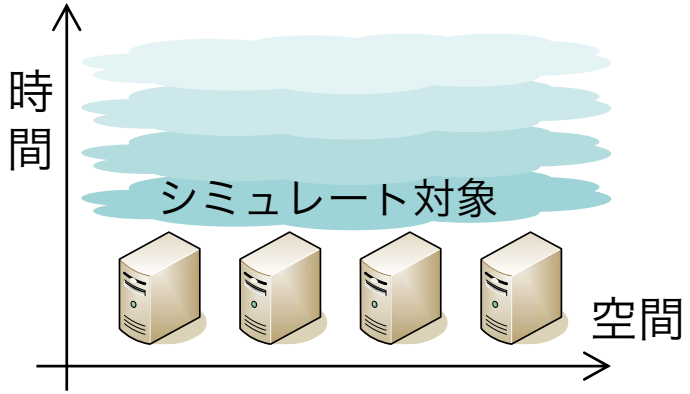
- 分子, 天体, 流体, 気象, 交通, …, 分散システム, …
- （空間的）シミュレーション単位間の相互作用を計算し、時間発展させていく
  - 単位：　　粒子, 分割された空間の代表点, ノード
  - 相互作用：　力, 車の出入り, 通信

- 単位時間あたりの影響範囲を限定しない。汎用。大変。
  - オーバレイ / P2P
    → 全ノード間で通信



大規模化

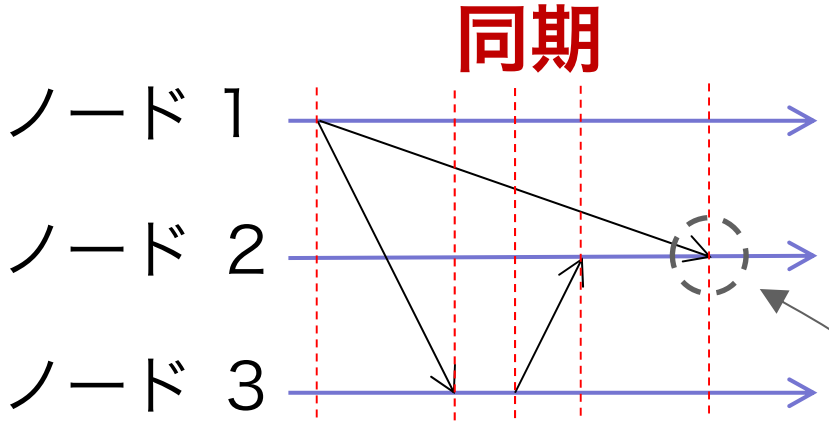
- 前提：イベント駆動 (event driven) シミュレーション
  - 規模の制約は、メモリ容量
  - １台の限界を越えようとするなら…
- 規模の向上：分散化 空間
- かつ、進行速度を一定以上に保つ 時間
  - 対象が分散システムなら、実時間より1〜2桁遅い程度まで？
  - 分散化すると、非常に遅くなる！
    - dPeerSim：47秒 → 1時間 6分, Chord 32万ノード
  - 同期のせい。



分散化の方針 (1)：時刻管理

- 楽観的な時刻管理：Time Warp [TOPLAS 1985]
  - ⇔ 悲観的

悲観的：イベントのたびに同期が要る



楽観的（Time Warp）

- まあいいや、と投機的に受信処理してしまう。
  - 同期回数が大幅減。
- 後でなんとかする：rollback, anti-message, global sync.

分散化の方針 (2)：ソフトの構成

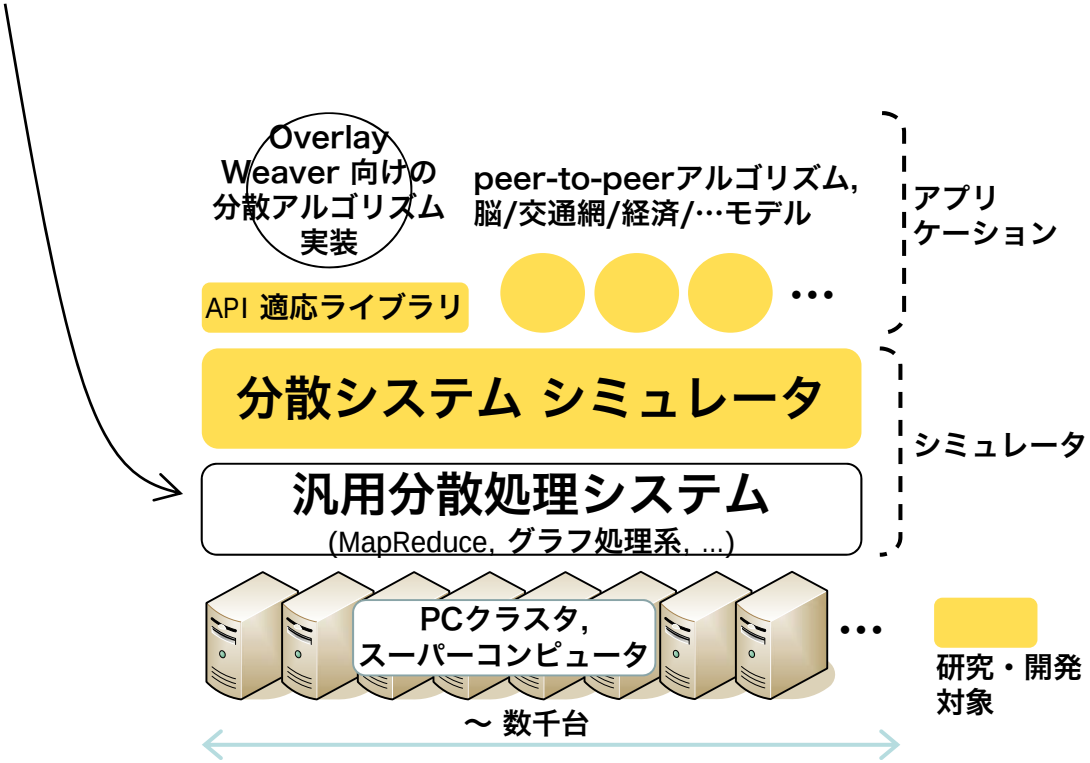
- スケールさせるために、耐故障性が欲しい。
  - 数十台 ～ 数千台 → 故障が日常茶飯事
    - 例：MTBF 15,000時間の PC が 24時間稼働
      10台 98.4%, 100台 85.3%, 500台 44.9%, …
- 自分達で作り込むのは、非常に大変。

- 汎用の分散処理システムを活用
  - Hadoop MapReduce 
  - 分散グラフ処理系

- 耐故障の機能
  （再実行等）
- 動作実績
  - 世界中
  - 4,500台
- ソフトウェアの成熟

分散化の方針 (2)：ソフトの構成

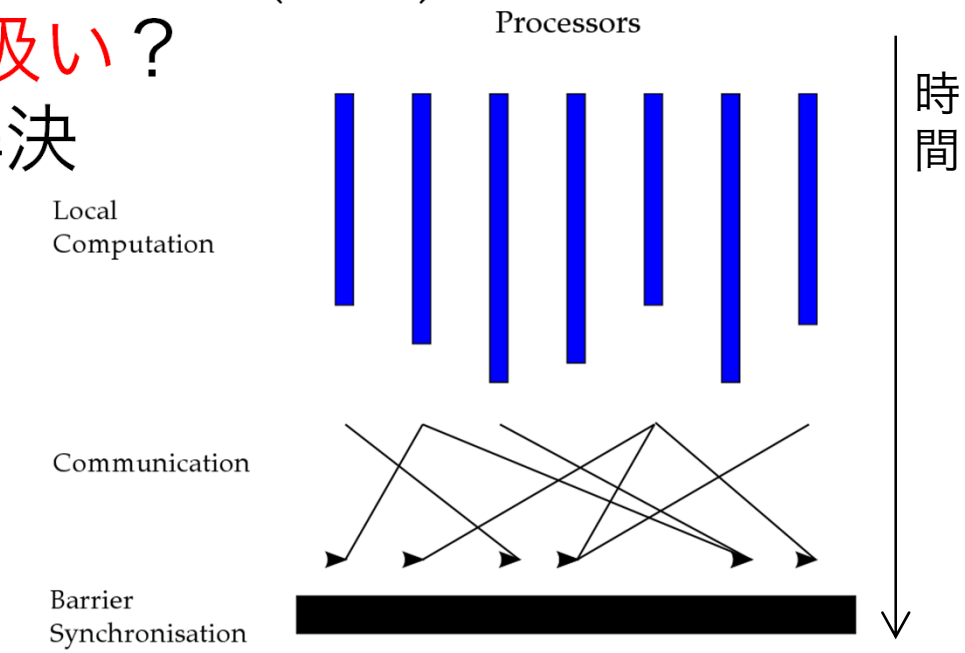
- 汎用の分散処理システムを活用
  - MapReduce 処理系, 分散グラフ処理系, …



分散化の方針 (2)：ソフトの構成

- 汎用の分散システム上での**課題** & **解決**
  1. 分散システム（ノードと通信）のモデル化？
     → ひと工夫で、できた (省略)

  2. Bulk Synchronous Parallel (BSP) 並列処理モデルでの時刻の取り扱い？
     → Time Warp で解決
     意外とマッチ



BSP モデル
(画像：Wikipedia より)

シミュレータの現状

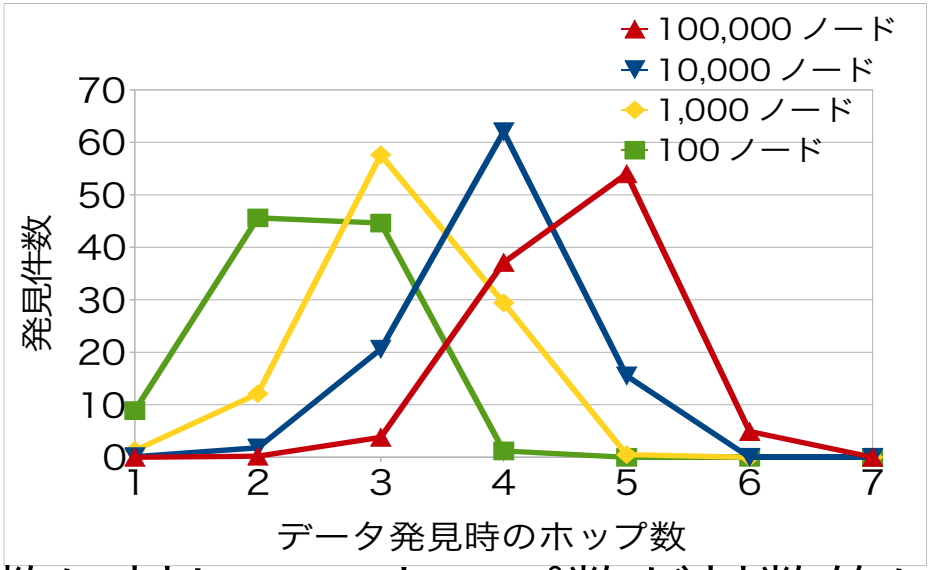
- MapReduce 版　by 杉野
  - MapReduce： Apache Hadoop を使用
  - Gnutella の flooding をシミュレート
    10台で 1億ノード

- 分散グラフ処理系 版　by 華井
  - グラフ処理系： Pregel [SIGMOD 2010] にならった自前実装
  - DHT アルゴリズム Chord をシミュレート
    10台で 10万ノード

  - シミュレーションシナリオを読み込み、実行。
  - アプリ実装のための API を提供。主に送受信。

シミュレータの動作

- MapReduce 版：
  複雑ネットワーク上の flooding

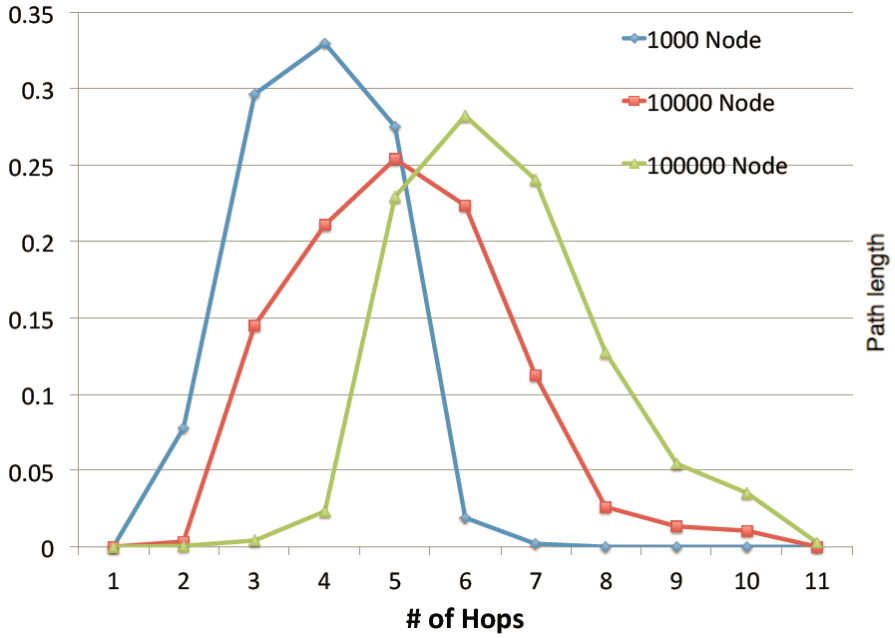


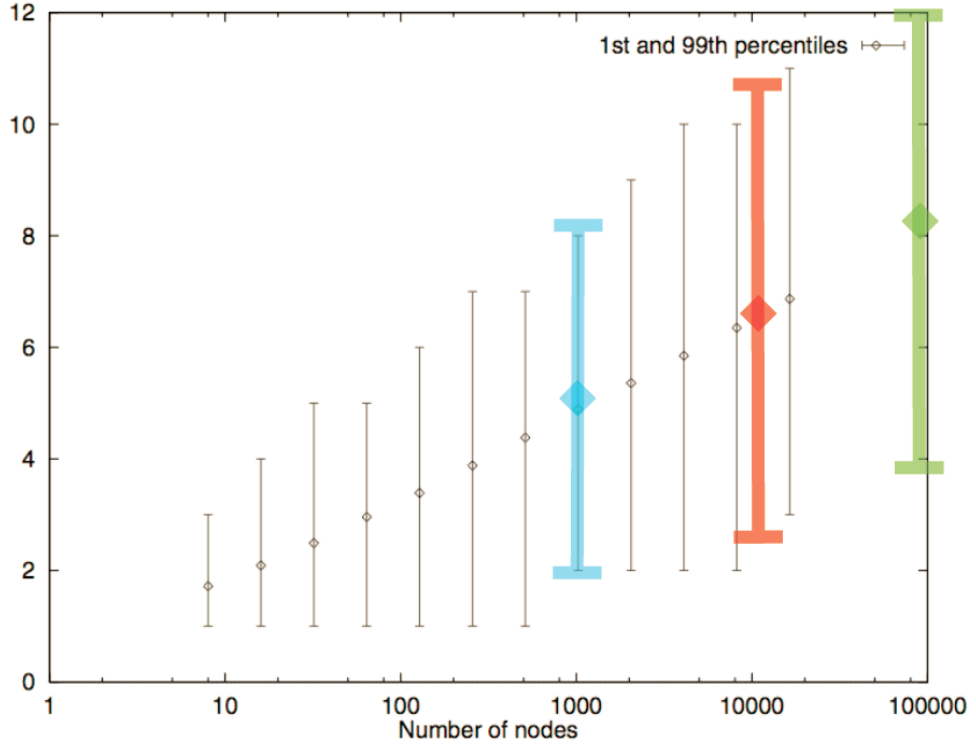
  - ノード数に対して、ホップ数が対数的に増加
  - すべての探索が成功

スモールワールド性
妥当な シミュレーション結果

シミュレータの動作

- 分散グラフ処理系 版：
  DHT アルゴリズム Chord でのホップ数



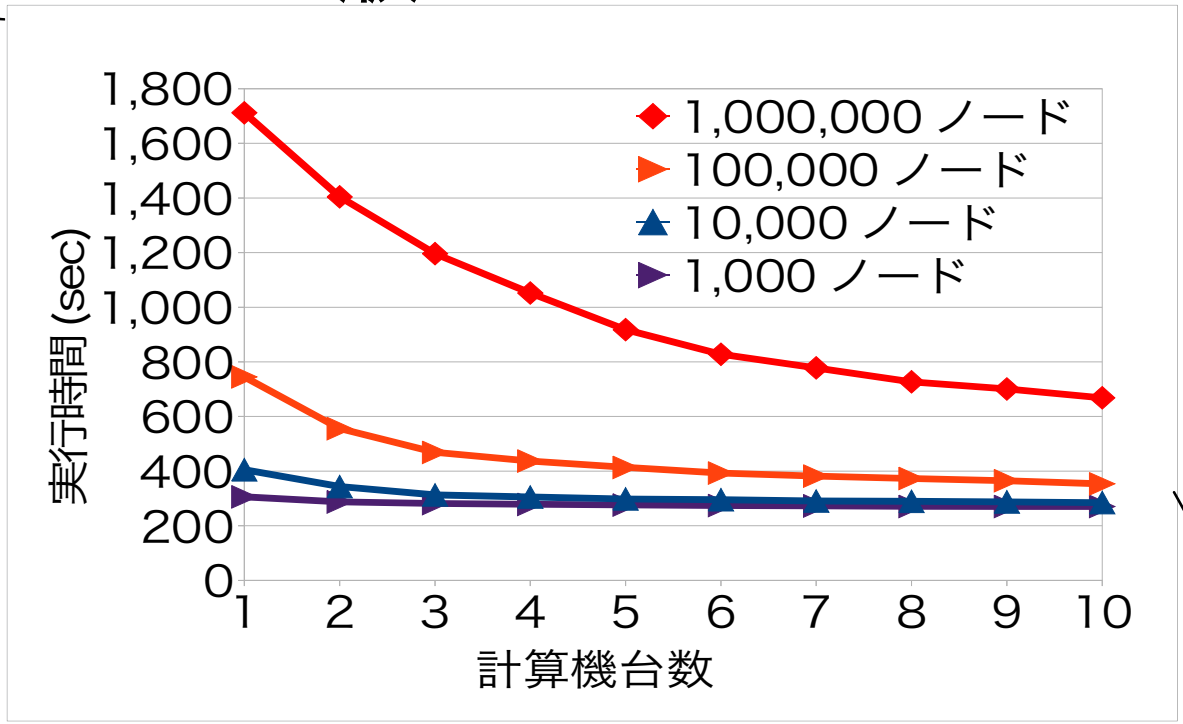


Chord 論文の図の上にプロット

妥当な シミュレーション結果

シミュレーション性能

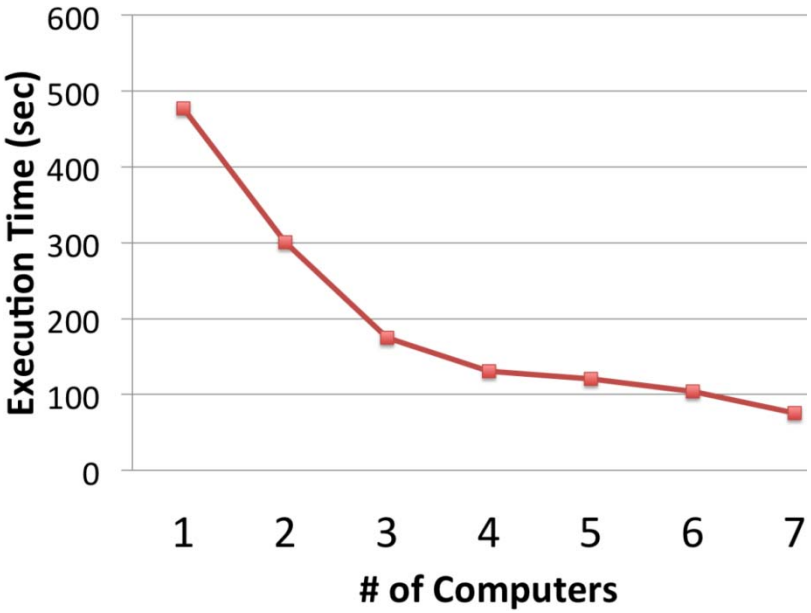
- MapReduce 版



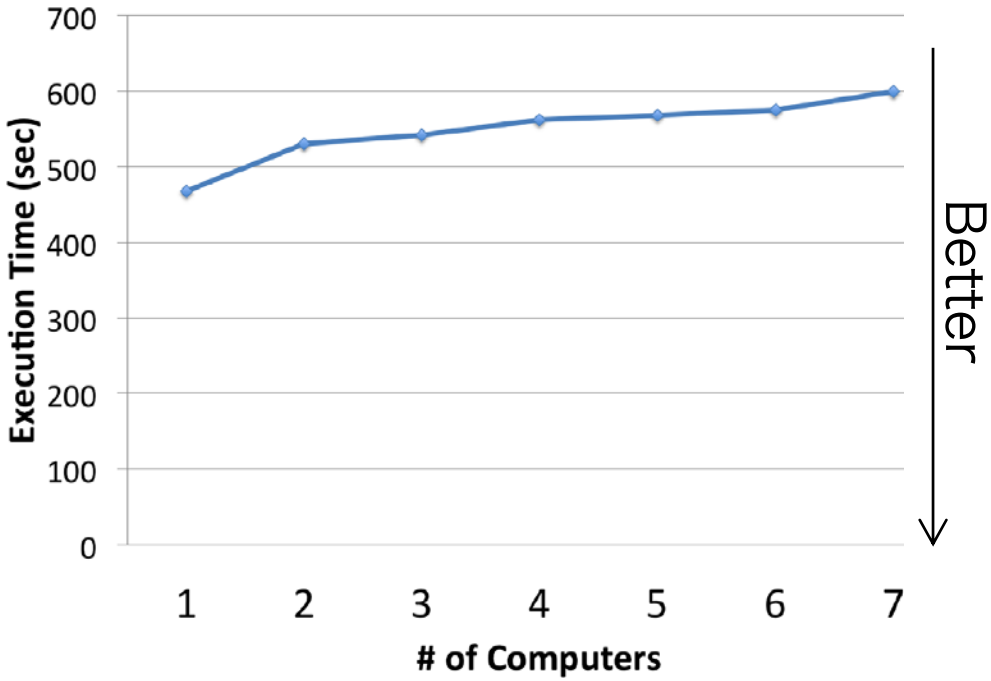
  - 台数に応じて、時間短縮
    - この性能はどうなのか？ これ以上の規模は？
  - Hadoop 起因のオーバヘッド → 高速化の余地

シミュレーション性能

- 分散グラフ処理系 版



Strong Scaling
10,000ノード / 1〜7台



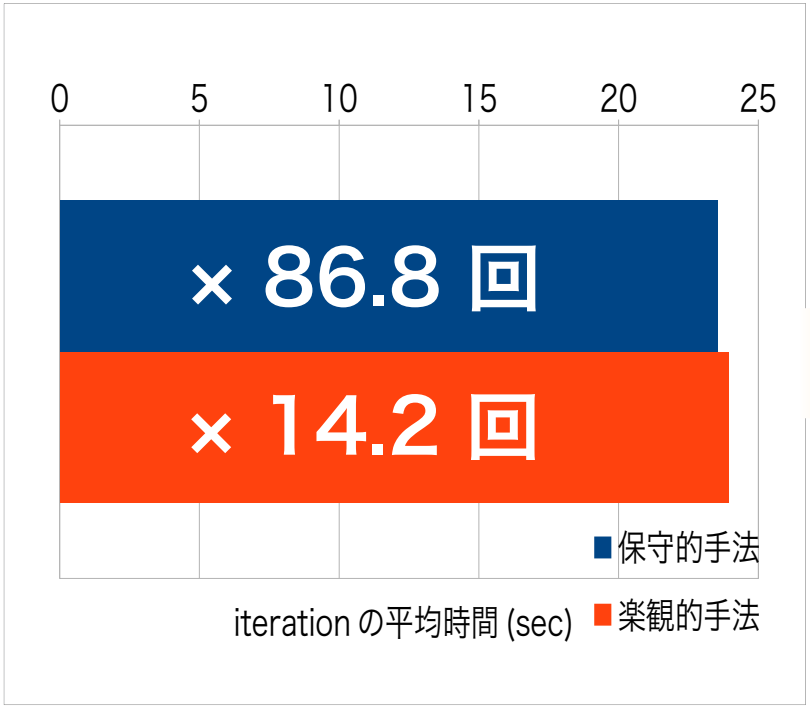
Weak Scaling
10,000ノード / 台

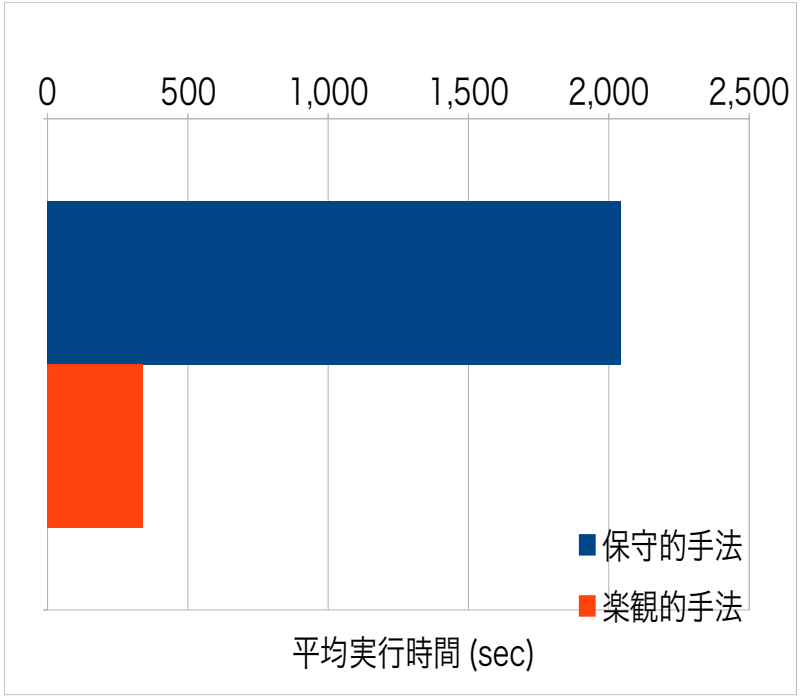
– 良好にスケール

- この性能はどうなのか？ これ以上の規模は？

時刻管理アルゴリズム

- 楽観的な時刻管理で、同期回数が大幅に減





大幅な実行時間 減

分散システムの大規模シミュレーション

- 自律分散システム / peer-to-peer を数十億ノード規模でシミュレートしたい
- １億ノードのシミュレート (PC 10台) ができた
  - 従来の２桁上
  - うまくいきそう
- 今後
  - 高速化, 大規模化
  - 他の応用
  - ネットワークの綿密なシミュレート