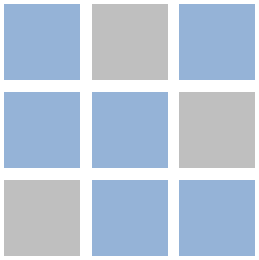
A Stable Broadcast Algorithm for Distributed Environments and Its Implementation

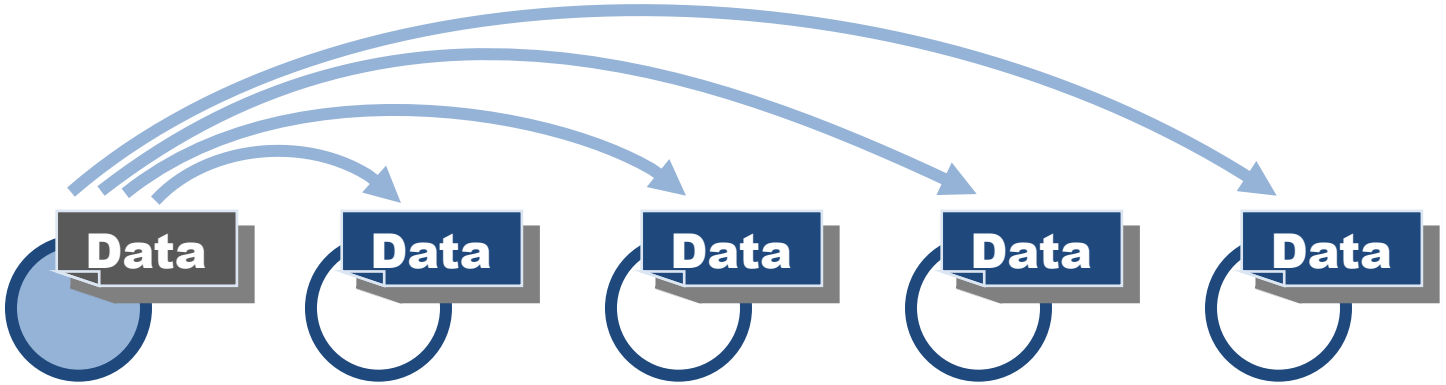
分散環境でのStableなブロードキャストアルゴリズムの提案と実装



近山・田浦研究室　高橋 慧
(56428)

ブロードキャストとは:

- 多数のノードへ同じデータを転送すること
  - 転送元 (Source)から転送先(Destination)へ
- 一般に広く用いられる
  - ファイルの転送・ストリーミングなど



大きなデータのブロードキャスト

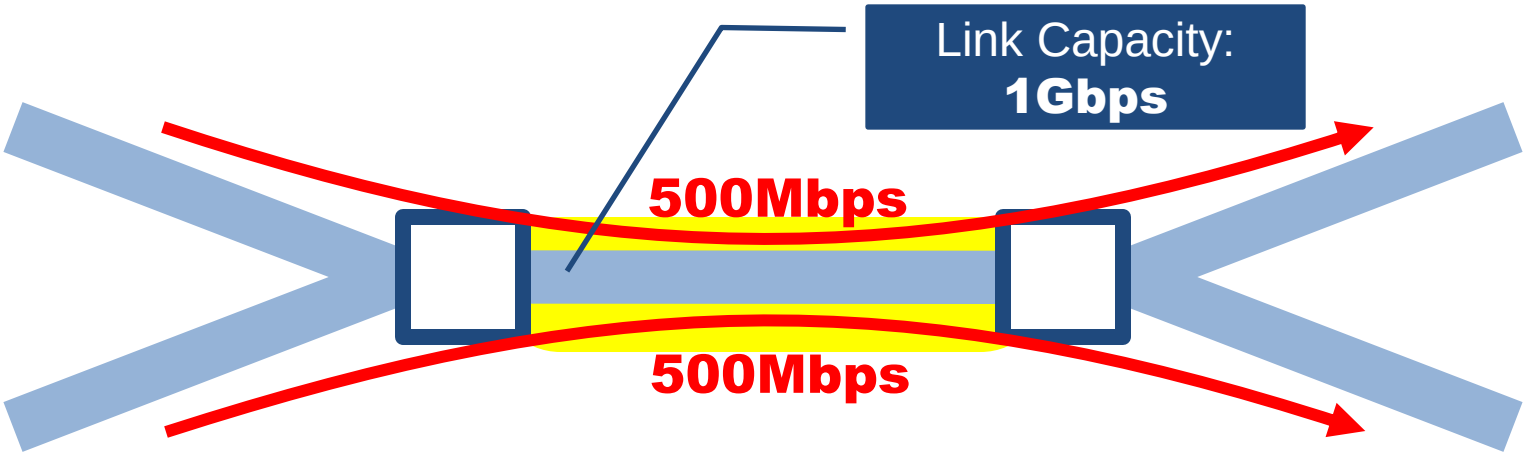
- バンド幅が重要になる
- パイプライン転送により性能が改善される
  - 転送先ノードがデータを中継

並列計算でのブロードキャスト

- クラスタ内や分散環境でも広く用いられる
  - 同じデータに対し異なる処理
  - 大きな辞書をローカルディスクにコピー
- 単純な手法では様々な問題が発生する
  - リンク共有による性能低下
  - 遅いノードによる性能低下

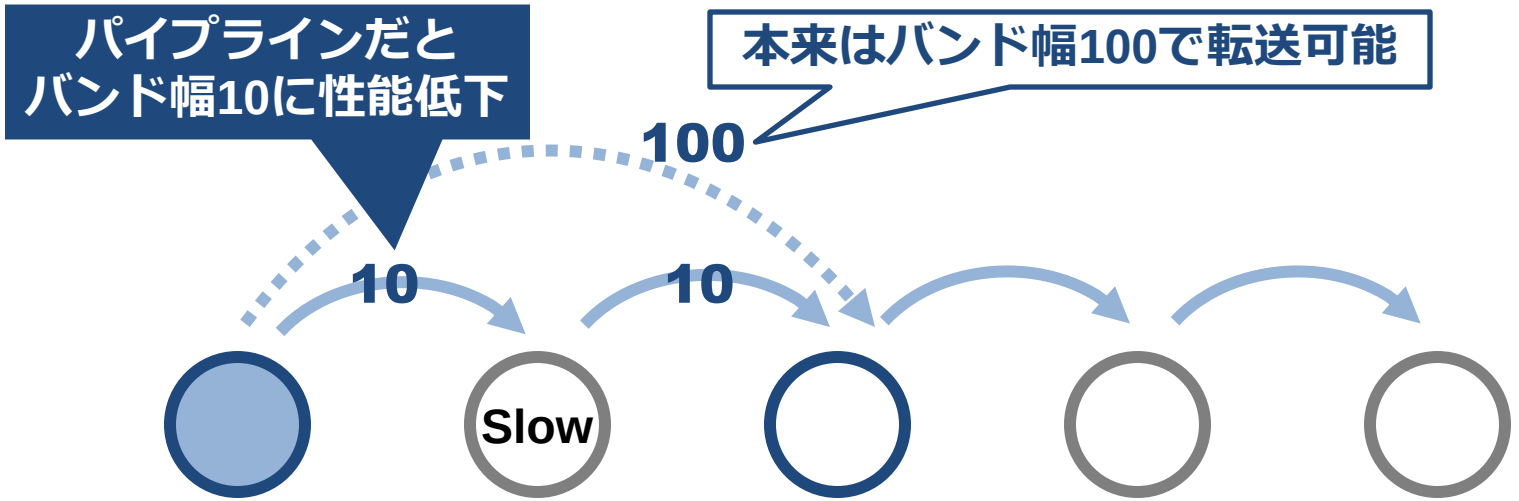
問題1:リンク共有による影響

- N本の転送がリンクを共有すると、転送のバンド幅は最悪(1/N)になってしまう
  - 1種類のデータ転送については、同じリンクを何度も通らないことが望ましい



問題2:「遅い」ノードによる影響

- バンド幅が小さな(遅い)ノードがパイプラインに参加すると、全体の性能が低下してしまう
  - バンド幅の小さなノードは他のノードの邪魔をしないことが望ましい



本研究の貢献

- 「**Stable**なブロードキャスト」という概念を提案
  - 遅いノードが速いノードの邪魔をしない
  - 全ノードが可能な最大バンド幅でデータを受信
- 木構造ネットワークで実際にStableなブロードキャストアルゴリズムを提案
  - Stableであることを数学的に証明
  - 一般のグラフネットワークでも性能改善
- 実験の結果、不均質な環境で既存手法に比べ2.5倍の合計バンド幅を達成

発表の流れ

- 研究の概要
- 問題設定
  - 大きなメッセージのブロードキャスト
  - Stabilityの定義
- 関連研究
- 提案するアルゴリズム
- 実験・評価
- 結論

問題設定

- 対象: 大きなメッセージのブロードキャスト
  - ノード間での一対一転送を組み合わせて実現
- ネットワークトポロジ・バンド幅は既知とする
  - トポロジは高速に推定可能
    - 4クラスタ・ 256ノードの推定に16秒

      Shirai et al. A Fast Topology Inference
      – A building block for network-aware parallel computing. (HPDC 2007)
  - リンクのバンド幅も高速に測定可能

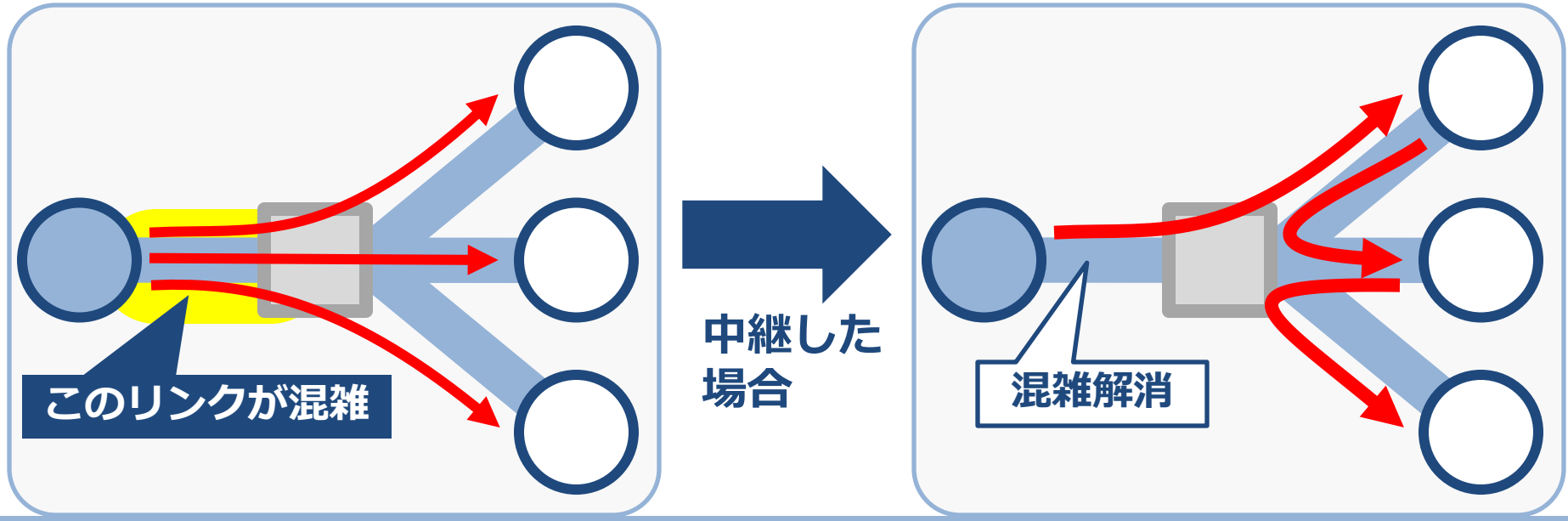
    長沼他. ネットワークトポロジを考慮したバンド幅推定の高速化手法.
    (情報処理学会学生セッション, 2008年3月 発表予定)

バンド幅と遅延

- メッセージ伝達時間は遅延とバンド幅で決まる
- メッセージが大きい場合、到達時間においてバンド幅のみが重要になる
  - $(\text{到達時間}) = (\text{遅延}) + \dfrac{(\text{メッセージの大きさ})}{(\text{バンド幅})}$
  - メッセージサイズ1GB・バンド幅1Gbps ・ 遅延50msだと、到達時間の99%以上がバンド幅の項

中継の必要性

- 中継を用いず、転送元ノードが直接全ノードに転送すると、転送元ノードがボトルネックになる
- 転送先ノードが中継を行うと、各ノードがより多くのバンド幅でデータを受け取ることができる
  （パイプライン転送）



非同期的なブロードキャスト

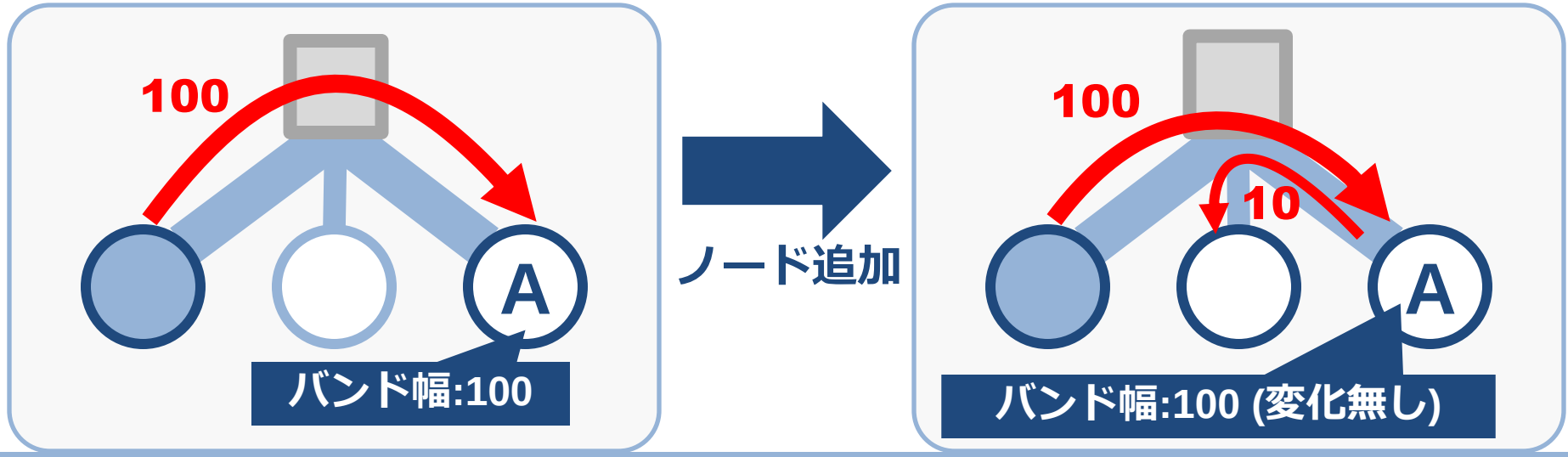
- 多くの既存アルゴリズムは、全ノードに同じバンド幅でデータが届くのを前提としていた
- 実際には大きなバンド幅で接続されたノードは、より早くデータを受け取ることが望ましい
  - データが到着したら独立に処理を開始可能

ブロードキャストの性能評価

- 転送時間による評価:
  - 最も遅いノードにデータが到着する時間で評価
  - ノード間の同期が不要な場合には適さない
- 合計バンド幅による評価:
  - 各ノードが受け取るバンド幅の合計で評価
  - 全ノードが同量のデータを受け取るのであれば、転送時間と同じ結果となる
  - 非同期的なブロードキャスト(各ノードが異なった量のデータを受け取る場合)の性能評価に適切である

Stableなブロードキャストとは:

- 各ノードに届くバンド幅が、バンド幅の少ないノードを追加しても減らないこと、と定義する
- ブロードキャストがStableなら以下も満たされる
  - 合計バンド幅最大
  - 転送時間最小

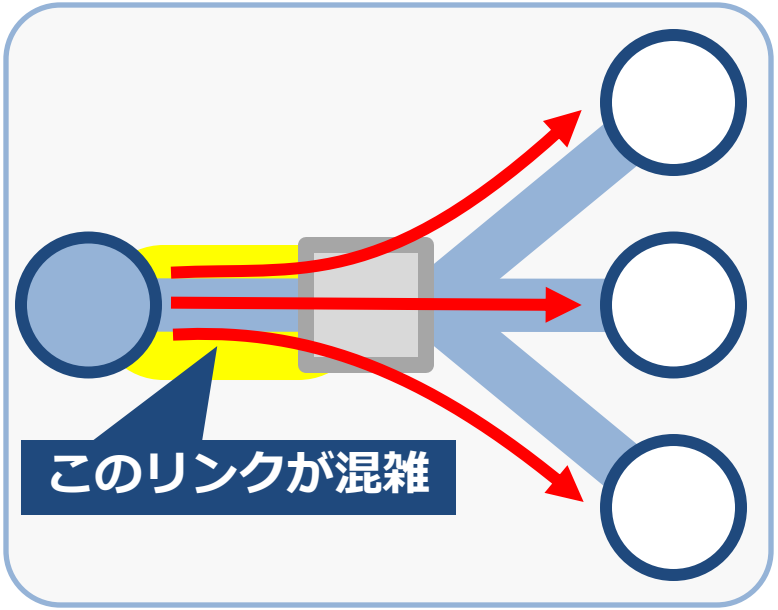


発表の流れ

- 研究の概要
- 問題設定
- 関連研究
- 提案するアルゴリズム
- 実験・評価
- 結論

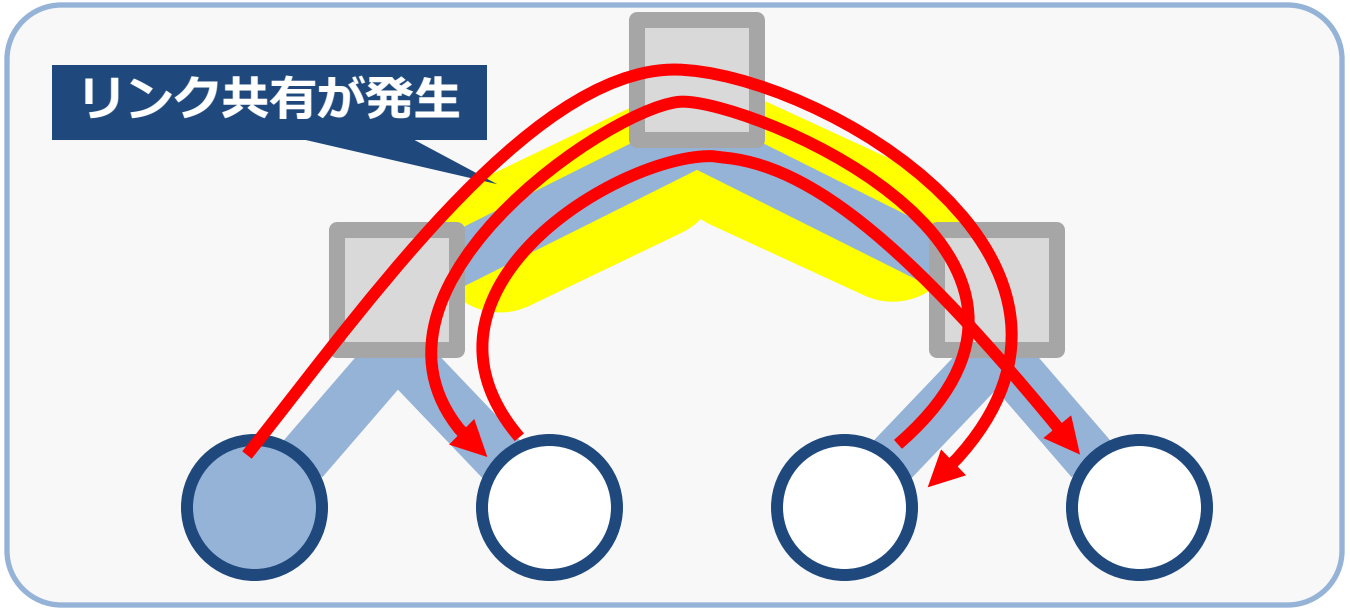
FlatTreeアルゴリズム

- 転送元ノードから直接多数のノードに転送
  - 実装が容易
  - ノード数に比例した時間が必要
  - ノード数が増えると性能が大きく低下



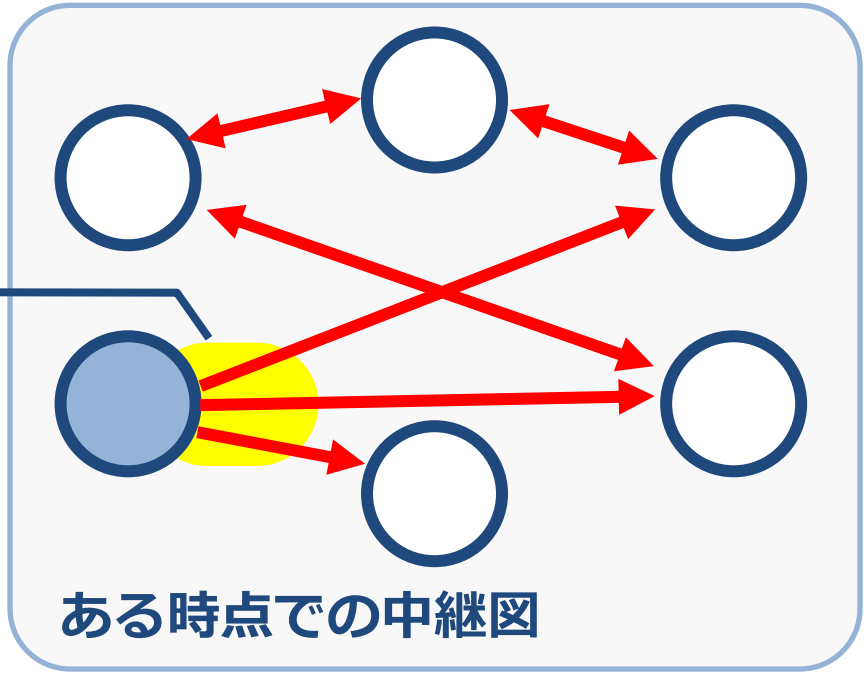
トポロジを考慮しないパイプライン

- 転送元ノードから転送先ノードを順番に辿る
  - 転送元のボトルネックを解消
  - リンク共有が発生し性能低下



BitTorrentでのブロードキャスト[†]

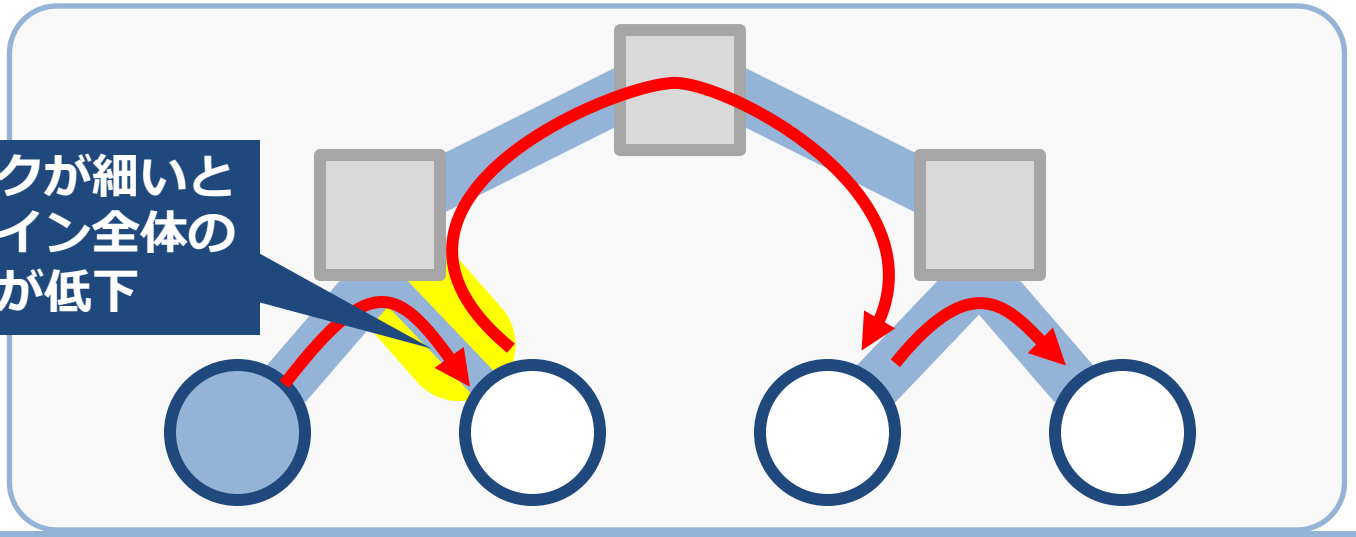
- BitTorrentは、転送時のバンド幅から中継構造を変化させ、合計バンド幅を順次向上させる
- 中継に枝分かれが多く、またリンク共有も回避されないため、クラスタ内でのブロードキャストには向かない



[†] Wei et al. Scheduling Independent Tasks Sharing Large Data Distributed with BitTorrent. (In GRID '05)

深さ優先パイプライン[†]

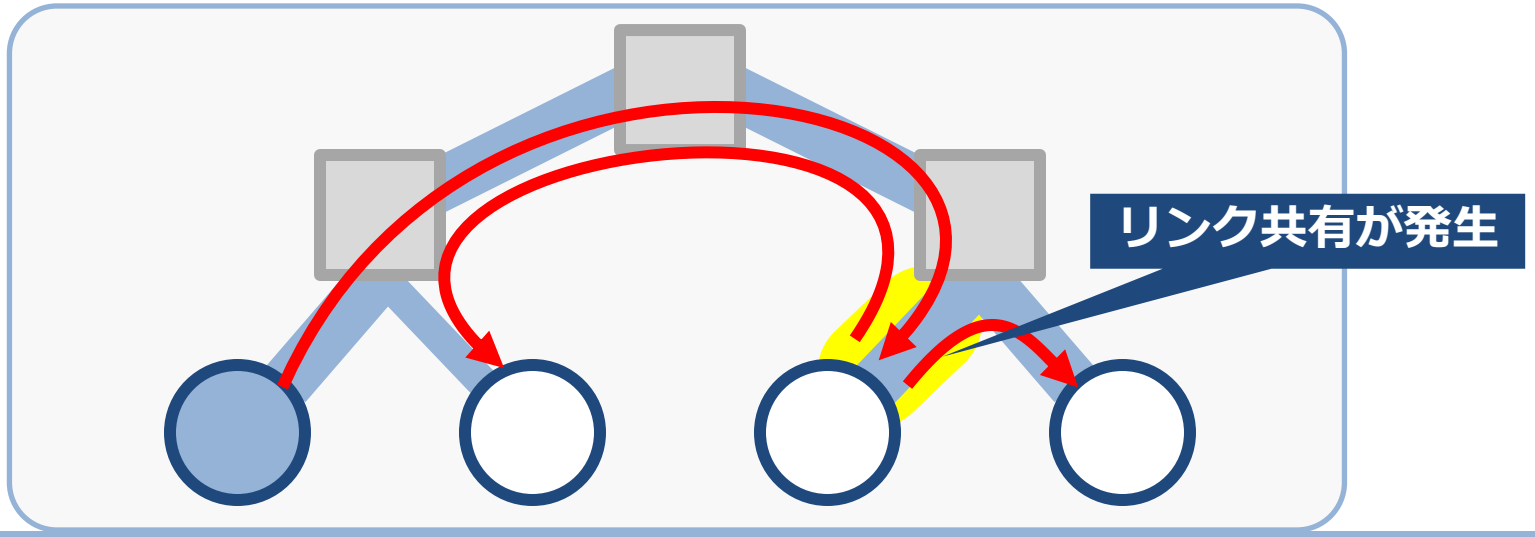
- トポロジ情報を用い、転送元から深さ優先にパイプラインを構築
  - リンク共有が起こらない
  - 木構造だと最も遅いノードにとって最短で受信完了
- 遅いノードによる性能低下が発生



[†] Shirai et al. A Fast Topology Inference - A building block for network-aware parallel computing. (HPDC 2007)

Dijkstra法によるツリー構築[†]

- バンド幅付きトポロジを用い、現在の転送ツリーからのバンド幅が最大であるノードを順次追加
  - 遅いノードによる性能低下を緩和
- リンク共有が発生する



[†] Wang et al. A novel data grid coherence protocol using pipeline-based aggressive copy method. (GPC, pages 484–495, 2007)

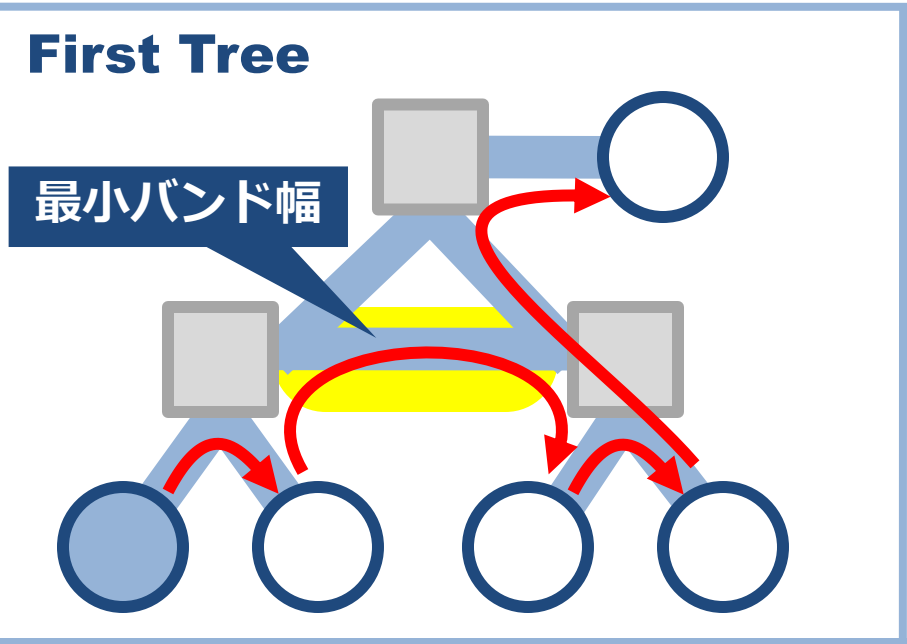
FPFRアルゴリズム[†]

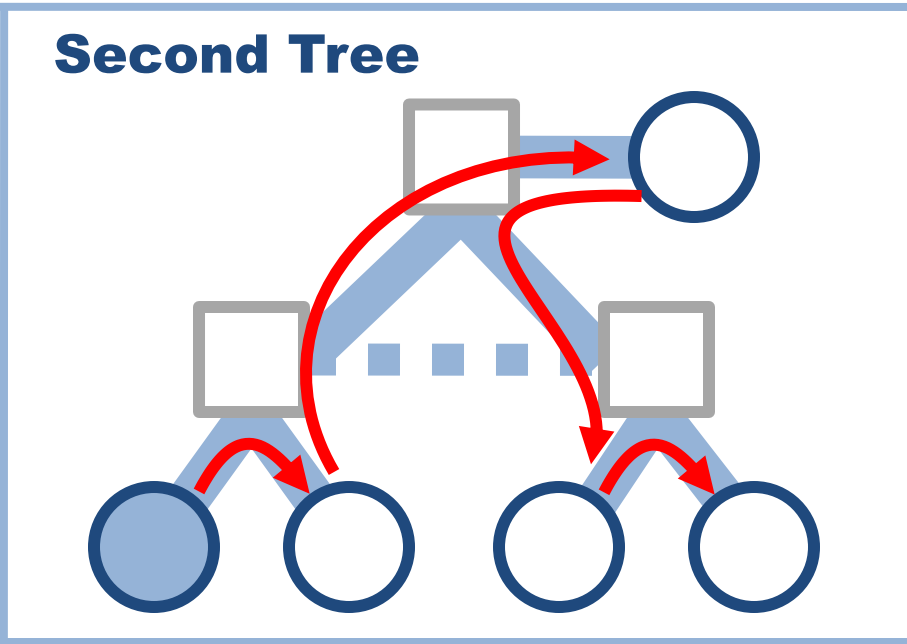
- FPFR: Fast Parallel File Replication
  - バンド幅付きトポロジ情報を用いる
  - 複数のストリームを用いることで、深さ優先パイプラインより合計バンド幅を向上
- アルゴリズム
  - 複数のツリーを深さ優先で構築
  - ツリー群を同時に用いてデータ転送

[†] Izmailov et al. Fast Parallel File Replication in Data Grid. (GGF-10, March 2004.)

FPFRにおけるツリー群の構築

- 深さ優先にツリーを構築し、最小バンド幅をスループットとする
- さらに深さ優先に次の木を構築する
  - 以前のツリーが用いるバンド幅を差し引いて構築する
  - バンド幅が0のリンクは連結されていないと見なす
- 全ての転送先を辿れなくなったら、ツリー構築を終了する

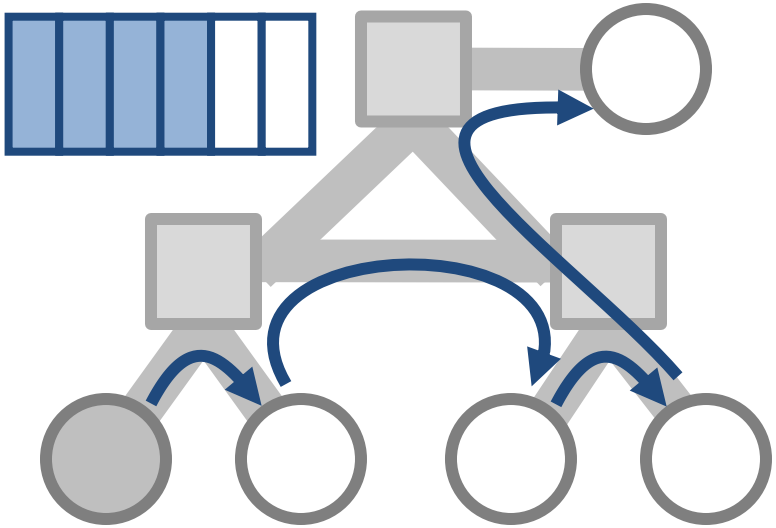




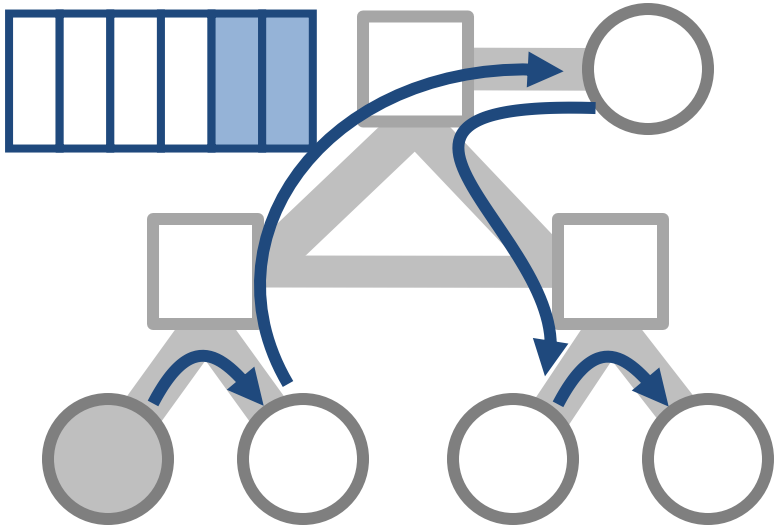
FPFRにおけるデータ転送

- ファイルを断片に分け、各ツリーが異なる断片を並列に送信する
  - 各ツリーのスループットに比例した量を送る

First Tree: データ前半を送信

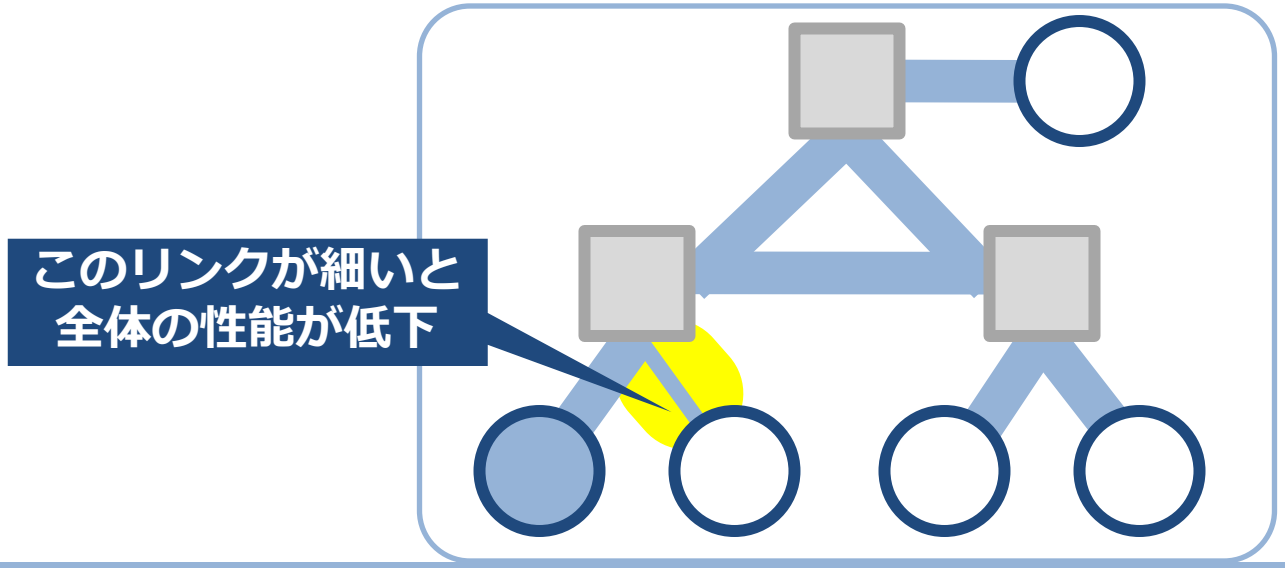


Second Tree: データ後半を送信



FPFRの問題点

- FPFRアルゴリズムは既存手法より合計バンド幅を改善したが、Stableではない
  - 遅いノードが他のノードに影響してしまう
- トポロジが木構造の場合は深さ優先パイプラインと全く同じ計画となり、性能向上されない



発表の流れ

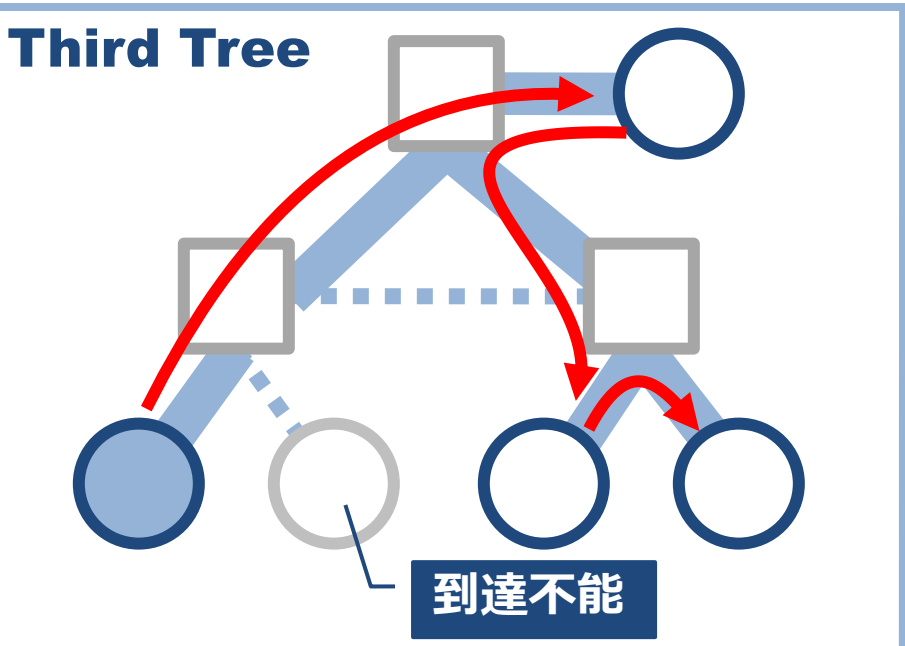
- 研究の概要
- 問題設定
- 関連研究
- 提案するアルゴリズム
  - ツリー構築
  - 転送の実行
  - ブロードキャストツールの実装
- 実験・評価
- 結論

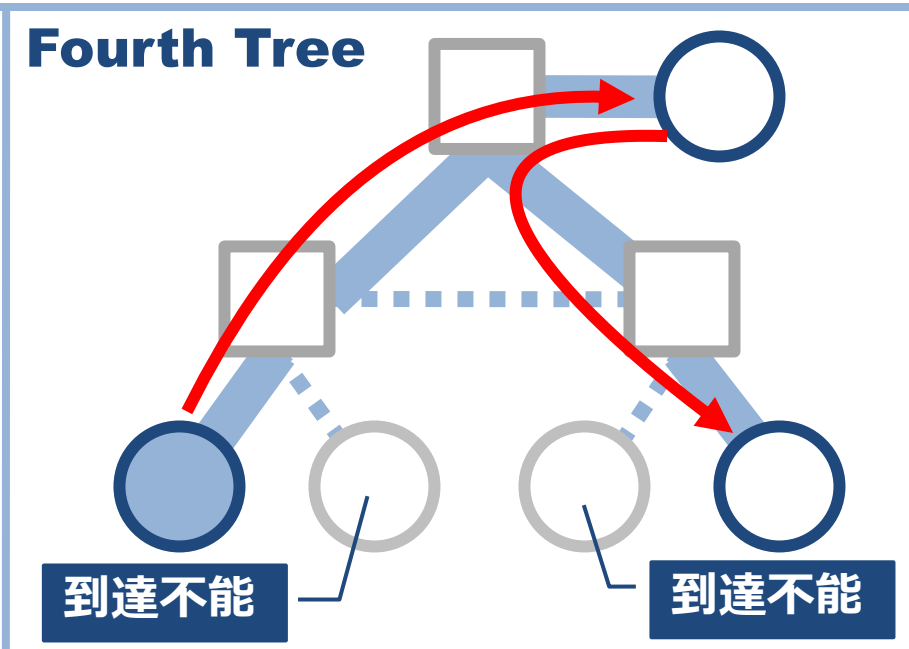
提案するアルゴリズム

- FPFRを改良
  - トポロジを用い、深さ優先にツリー群を構築する
  - 一部の転送先のみ結ぶツリーも用いる
- 性能的な改善点
  - 各リンクのバンド幅が上下対称な木構造トポロジにおいては**Stable**であることを証明した
    - 合計バンド幅最大かつ転送時間最短
  - 任意のトポロジに対し合計バンド幅を改善

ツリー群の構築

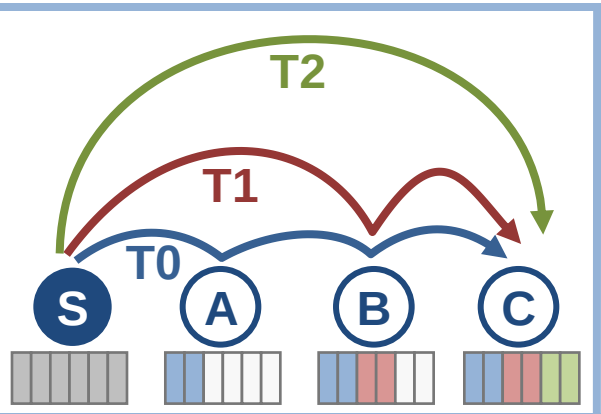
- FPFRと同様に深さ優先にツリーを構築
  - 全てのDestinationをたどれなくなっても、引き続き一部のDestinationを結ぶツリーを作る
- データ送信は全てのツリーを平行に用いる

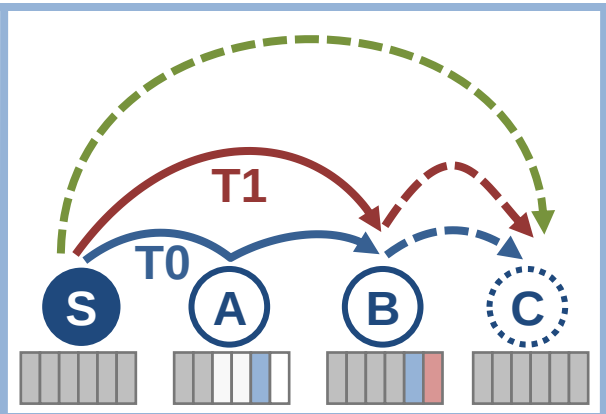


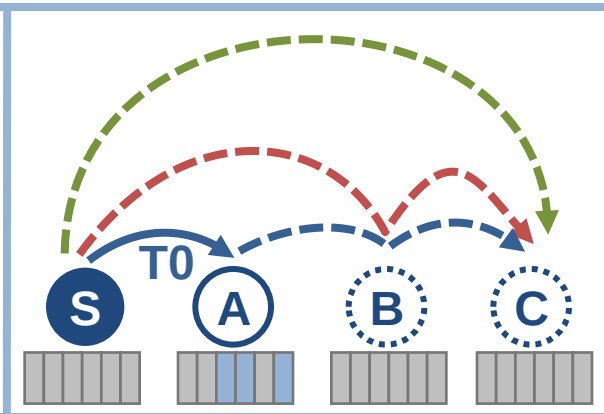


データ送信の例

- T0, T1, T2の3本のストリームを用いて転送
- Stage1: T0, T1, T2全てを用いて転送
  - Cがデータ全体を受け取ったら終了
- Stage2: T0, T1を用いて転送
  - 以前T2が送信した断片を転送
  - Bがデータ全体を受け取ったら終了
- Stage3: T0を用いて転送
  - 以前T1が送信した断片を転送







提案アルゴリズムの特徴

- 木構造(かつリンクの上下バンド幅が対称)ではStable
  - 全ノードが可能な最大バンド幅でデータを受信
  - 合計バンド幅最大
  - 完了時間最短 (最も遅いノードが最短で同期される)
- 任意のトポロジに対し合計バンド幅を改善
  - FPFR(及びそれに類する手法)が有効に利用ではないバンド幅を用いることができるため
  - 遅いノードの影響が小さい

ブロードキャストツールの実装

- 先に提案したアルゴリズムを利用した、ブロードキャストツール(ucp)を実装した
  - 効率的な中継計画を立てる
    - Stableなブロードキャストアルゴリズム
  - NAT・Firewall下で動作する
    - 逆向きの接続・中継を利用
  - 対話的に使え、使い勝手が良い
    - scpやrsyncに類似した書式
    - 正規表現によるホスト指定

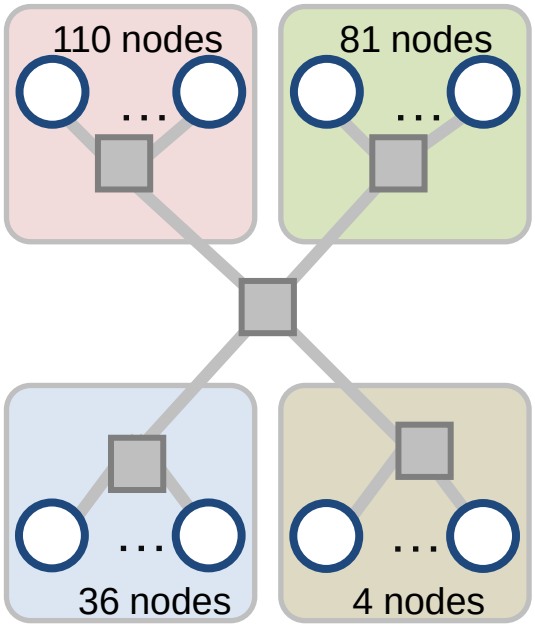
```
$ ucp hongo000:/source/ptn hongo:/destination/ptn
```

発表の流れ

- 研究の概要
- 問題設定
- 関連研究
- 提案するアルゴリズム
- 実験・評価
  - シミュレーション
  - 実機評価
- 結論

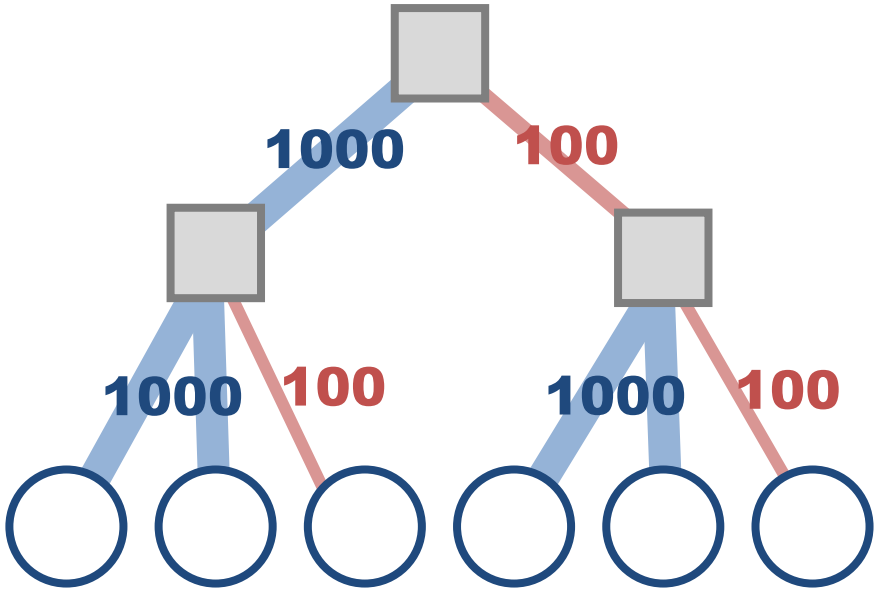
(1) シミュレーションによる評価

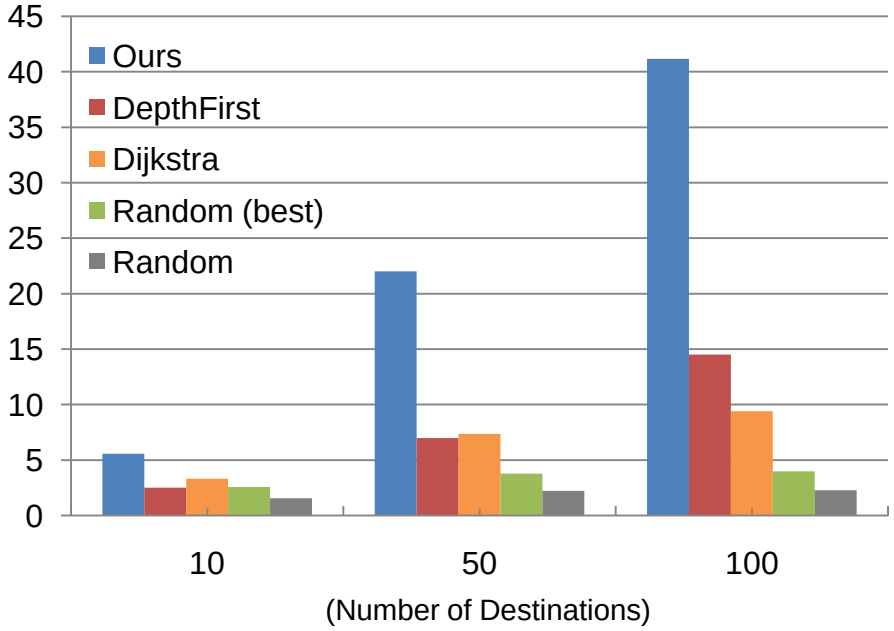
- 4クラスタの実機のトポロジを用い、以下の4手法を比較するシミュレータを実装した
  - 提案手法
  - 深さ優先 (FPFR)
  - Dijkstra
  - ランダム
    (トポロジを用いないパイプライン)
- 各手法の合計バンド幅を計算した
  - 10, 50, 100ノードのブロードキャスト
  - 転送元・転送先を変え、10回ずつシミュレート
  - 様々なバンド幅の条件を試行した
    (リンクの上下のバンド幅が非対称な条件を含む)



シミュレーション・条件1

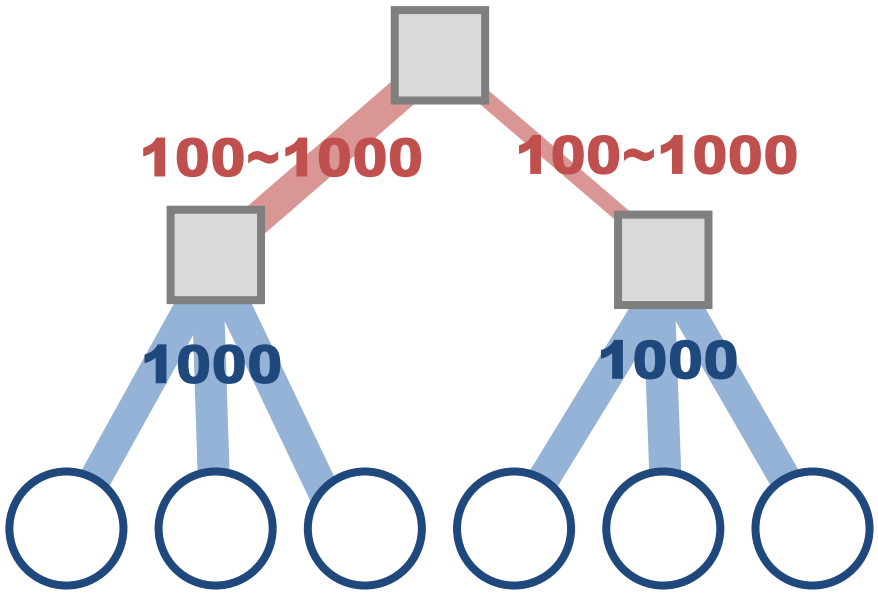
- 100Mbpsと1Gbpsのリンクをランダムに混在
  - FlatTreeに対する性能向上を縦軸として表示
  - 100ノードでFlatTreeの40倍、深さ優先(FPFR)の3.0倍
  - 速いノードが遅いノードに影響されないため

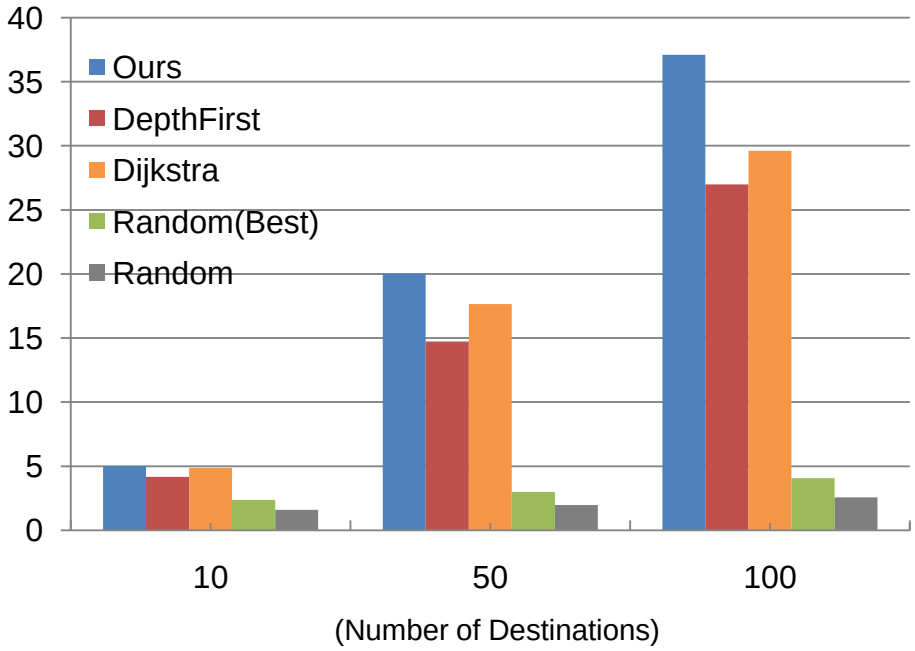




シミュレーション・条件2

- スイッチ間が100M-1Gbpsに一様分布
  - FlatTreeに対する性能向上を縦軸として表示
  - 100ノードでFlatTreeの36倍、深さ優先(FPFR)の1.2倍





シミュレーションのまとめ

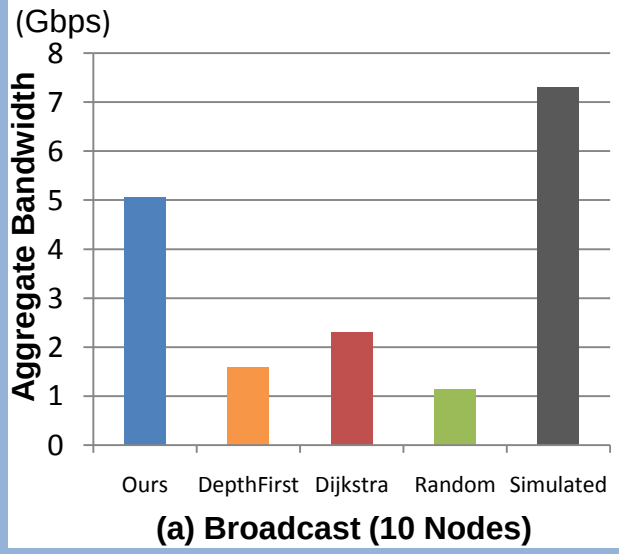
- 合計8種類のバンド幅分布で実験した
  - 500–1000Mbpsの間に一様分布
  - 100–1000Mbpsの間に一様分布
  - 100Mbpsと1000Mbpsを混在
  - スイッチ間を100–1000Mbpsの間に一様分布
- 1条件を除き、提案手法が最も高い合計バンド幅を達成した
  - 特にバンド幅の分散が大きい場合に大きく性能向上
  - 上下で異なるバンド幅を100–1000Mbps間に一様分布させた場合のみ、Dijkstraの98%の性能に留まった

(2) 実機での評価

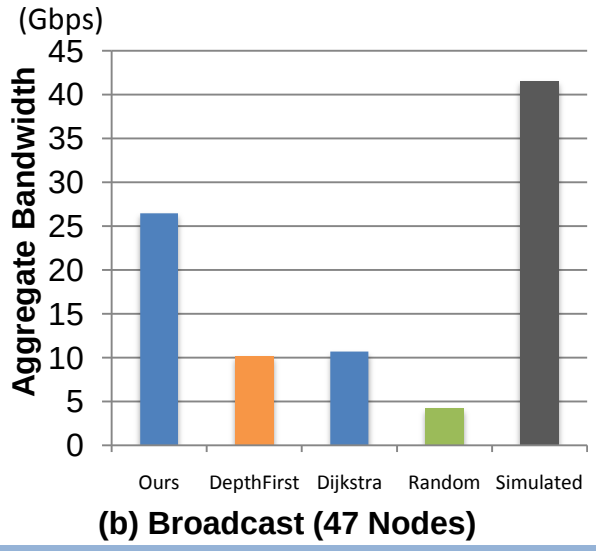
- 4クラスタ10, 47, 105ノードでブロードキャスト
  - ノード間のバンド幅は1Gbpsから10Mbpsの間に分散
- 4手法の合計バンド幅を比較した
  - 提案手法
  - 深さ優先 (FPFR)
  - Dijkstra
  - ランダム(100回のうち最高のスケジュール)
- さらに、理論上の最大値とも比較した
  - 各ノードが転送元から独立にデータを受け取った場合の、バンド幅の合計値

合計バンド幅の評価

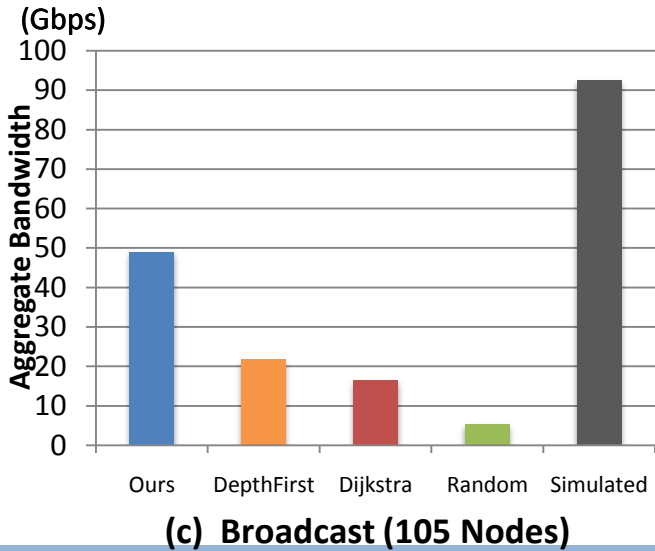
- 既存手法に比べ2.5倍以上の性能
- 理論最大値に比べると0.5−0.7倍に留まった
  - 用いたノードが上下のネットワークをフルに用いることができないため（例えば一方向の送信では900Mbps処理できるが、受信と送信を同時に行うと合計750Mbpsに低下してしまう）



(a) Broadcast (10 Nodes)



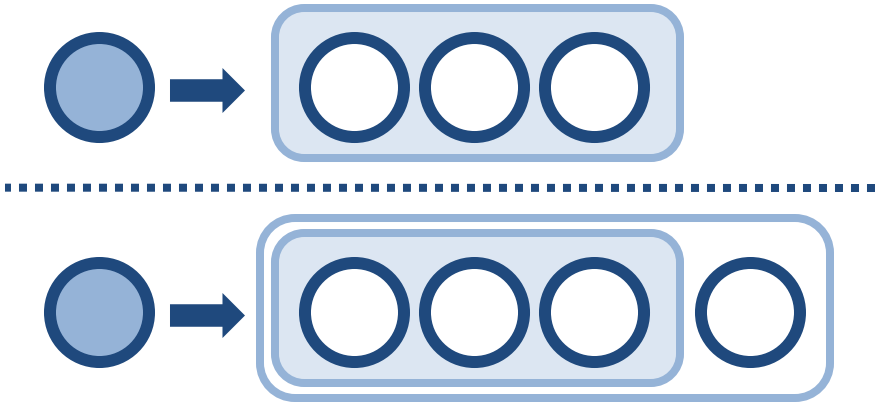
(b) Broadcast (47 Nodes)

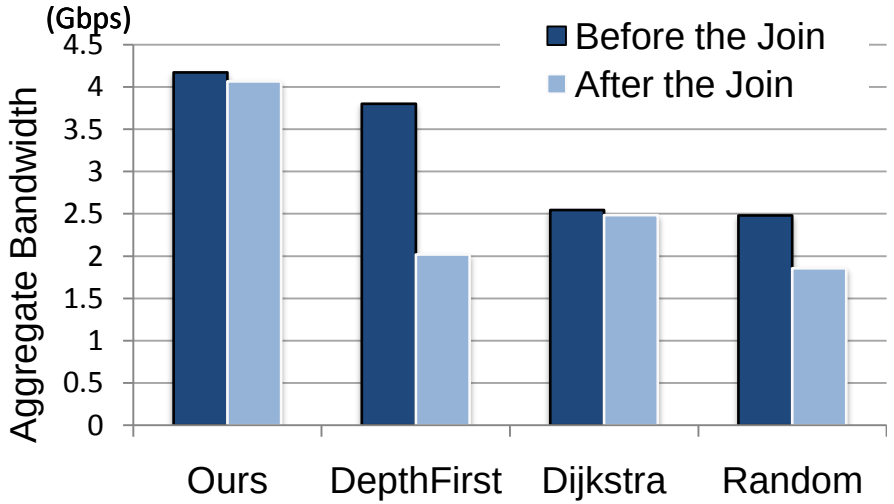


(c) Broadcast (105 Nodes)

Stabilityの評価実験

- 9ノードでのブロードキャストにバンド幅の小さな1ノードを追加した際、既存ノードの合計バンド幅の変化を観測
- 考察:
  - 深さ優先(FPFR)に比べ、遅いノードを追加しても既存ノードのバンド幅が変化していない
  - Dijkstraに比べ、合計バンド幅が1.6倍





発表の流れ

- 研究の概要
- 問題設定
- 関連研究
- 提案するアルゴリズム
- 実装・評価
- 結論

本発表のまとめ

- Stableなブロードキャストという概念を提案
- 実際に木構造でStableなアルゴリズムを提案
  - 数学的に証明
  - 実機において既存手法の2.5倍の合計バンド幅
  - シミュレーションで様々な条件での性能向上を確認
  - バンド幅が小さなノードを追加しても性能低下しない
- それを用いたブロードキャストツールを実装

今後の課題

- グラフ構造トポロジにおいて合計バンド幅を最大化するアルゴリズム
  - 常にStableなアルゴリズムは存在しない
- バンド幅変化を検知し転送構造を変化させる、転送の再構築アルゴリズム
- タスクスケジューラーとの連携

学会発表

- 国際会議

  Kei Takahashi, Hideo Saito, Takeshi Shibata and Kenjiro Taura.
  A Stable Broadcast Algorithm.
  To appear in the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid2008), May 2008

- 研究会報告

  高橋慧, 田浦健次朗, 近山隆.
  トポロジを考慮しソース選択を行うデータ転送スケジュラー.
  並列/分散/協調処理に関するサマーワークショップ(SWoPP2007),
  旭川, 2007年8月.

  高橋慧, 田浦健次朗, 近山隆.
  マイグレーションを支援する分散集合オブジェクト.
  並列/分散/協調処理に関するサマーワークショップ(SWoPP2005),
  武雄, 2005年8月.

- ポスター発表

  高橋慧, 田浦健次朗, 近山隆.
  マイグレーションを支援する分散集合オブジェクト.
  先進的計算基盤システムシンポジウム(SACSIS2005), 筑波. 2005 年5月.