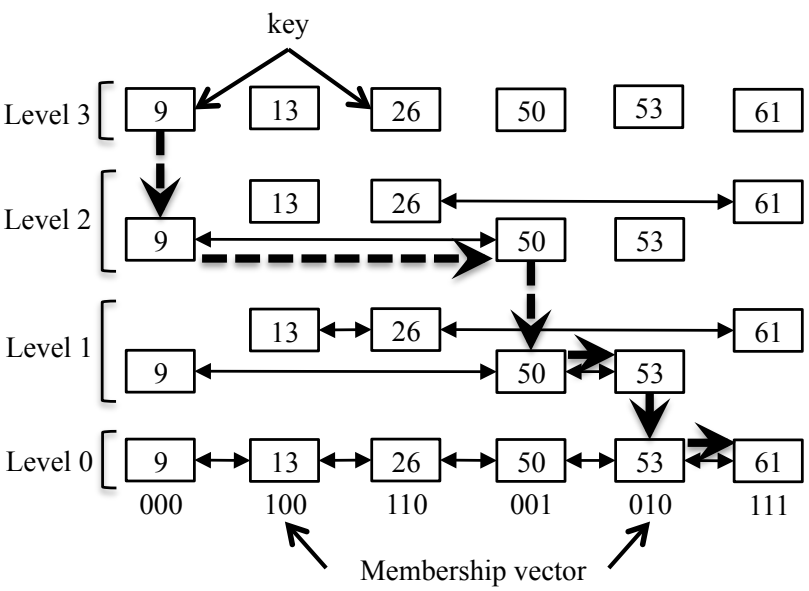
スキップグラフにおける仮想レプリカノードによるフラッシュクラウド耐性向上手法

ショウ　シュン、地引昌弘、寺西裕一、西永望

2014年11月14日

Skip Graph、およびその応用

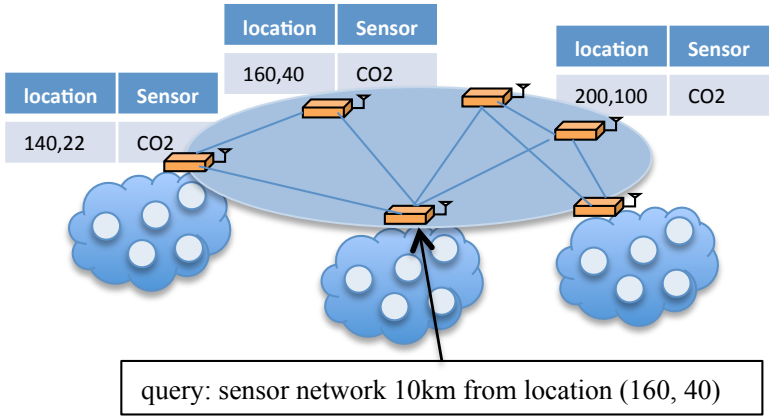
Skip Graph: key を探すための分散データストラクチャ



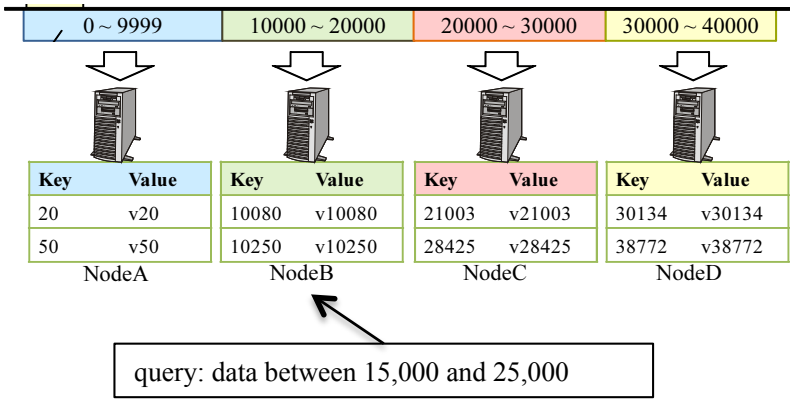
Skip Graph の最大な特徴: 範囲検索が可能. e.g. key の範囲は [a, b] のノードを探すようなことができる

Skip Graph の応用事例

例１: Range queriable sensor overlay networks[1]



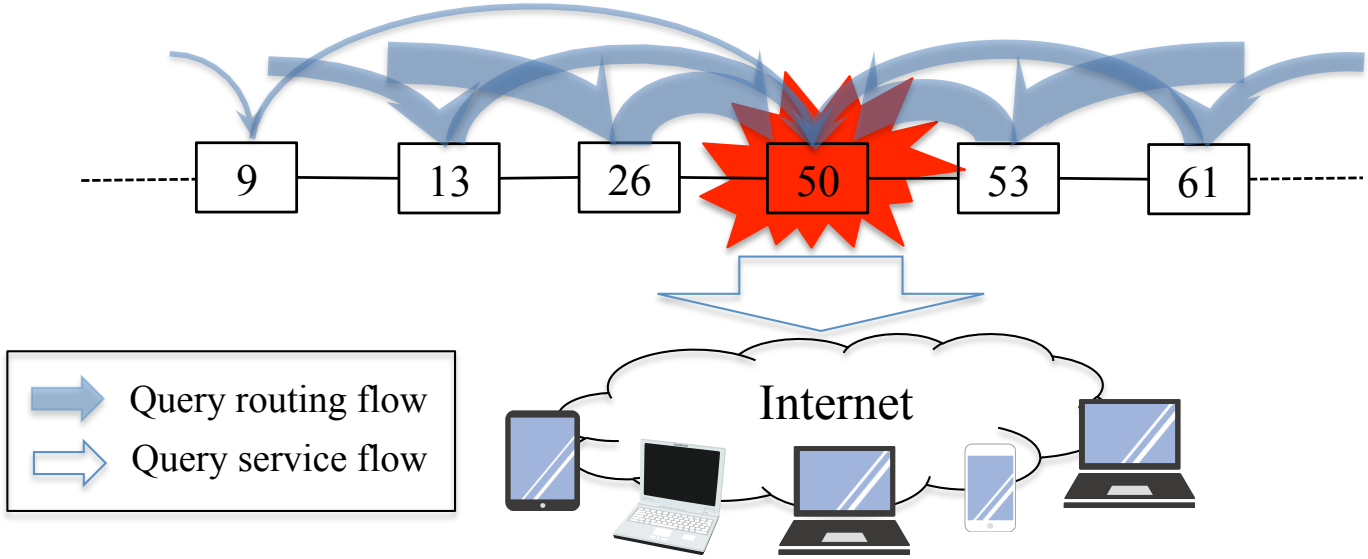
例2: Range queriable Key-value store [2]



[1] Y. Teranishi, “PIAX: Toward a Framework for Sensor Overlay Network,” in Proc. of CCNC’09, 2009
[2] Y. Ishi, Y. Teranishi, M. Yoshida, S. Takeuchi and S. Shimojo, “A Large Scale Key-Value Store Based on Range-Key Skip Graph and Its Applications,” in Proc. of DASFAA’10, 2010

Skip Graph における flash crowds 問題

Flash crowds 問題: 突発的に特定コンテンツへのリクエストが大量発生する事象
例：突発的な天候の変動により、ユーザの関心が特定のコンテンツに向かうため、特定ノード (hotspot) にクエリが集中する

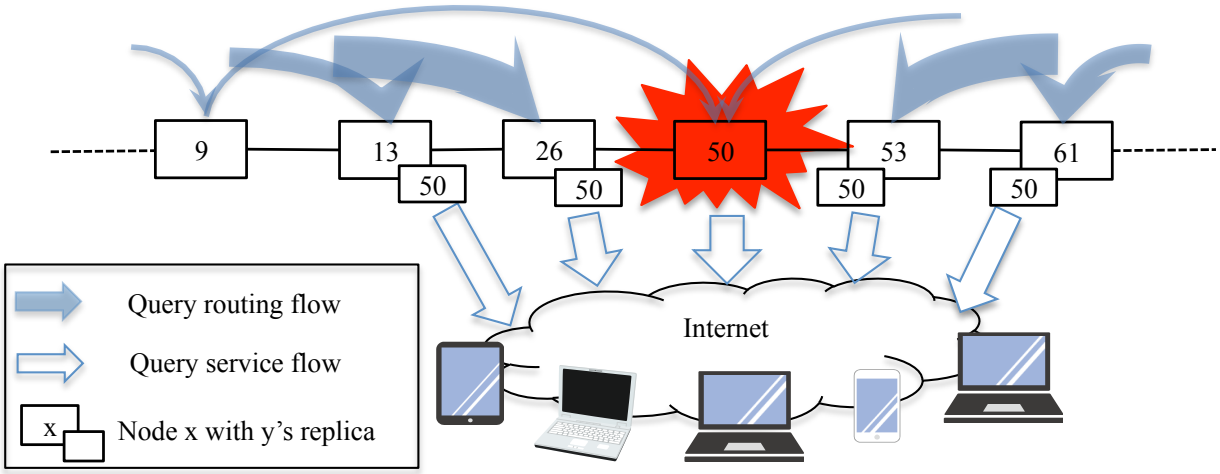


Flash crowds 問題による以下の負荷の激増

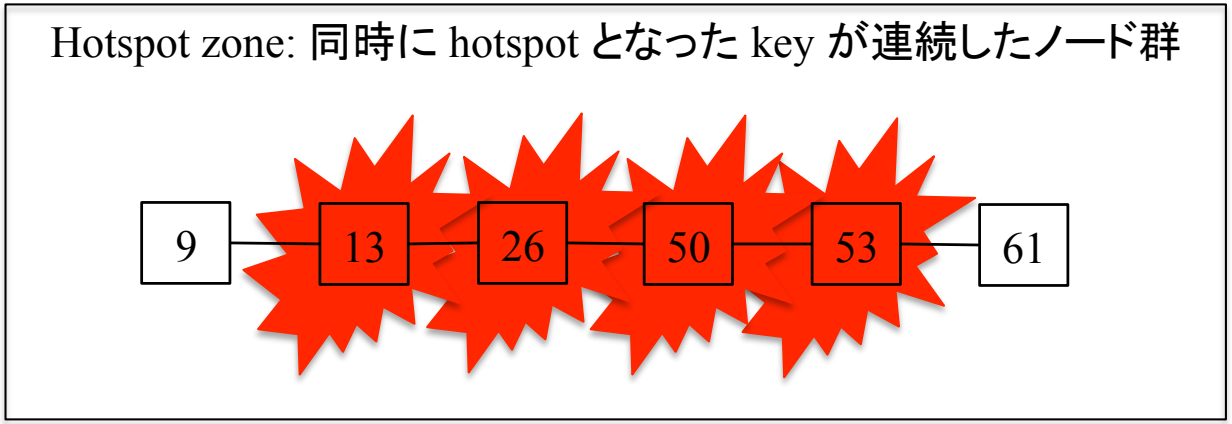
1) Query service load：Hotspot のノードが、ユーザへデータを提供する負荷
2) Query routing load：Hotspot 周辺ノードによる、hotspot へクエリを転送する負荷

既存の P2P における flash crowds を緩和させる手法

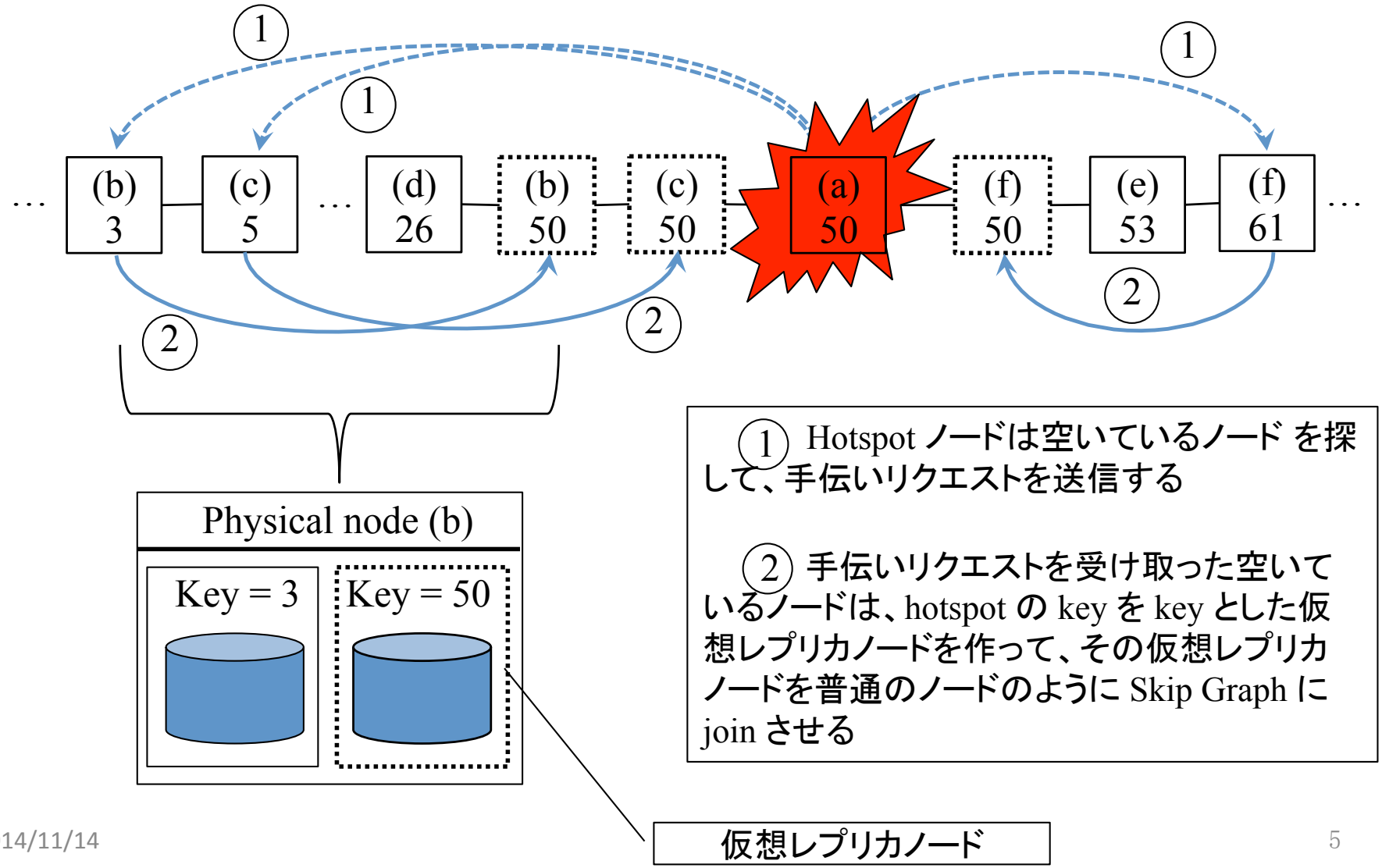
- 既存手法：Hotspot ノードは周辺ノードにレプリカを持たせることで、query service load を分散する．例：ノード 13、26、53 と 61 は、50 のレプリカを持つことで、50 と共にユーザーに要求されたデータを提供する



- 問題：Range queriable P2P (e.g. Skip Graph) によるhotspot zone には対応できない



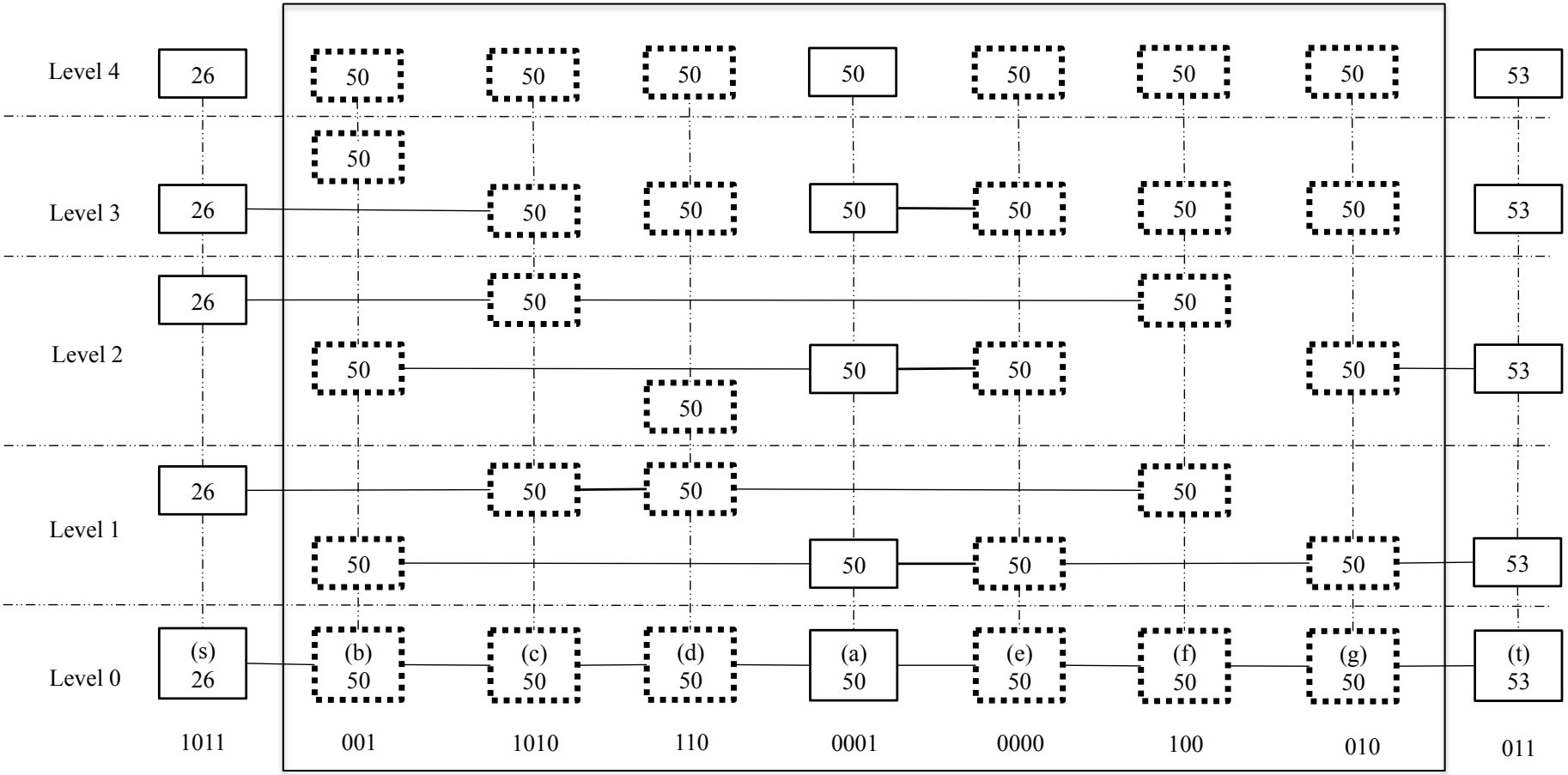
提案手法概要



① Hotspot ノードは空いているノード を探して、手伝いリクエストを送信する

② 手伝いリクエストを受け取った空いているノードは、hotspot の key を key とした仮想レプリカノードを作って、その仮想レプリカノードを普通のノードのように Skip Graph に join させる

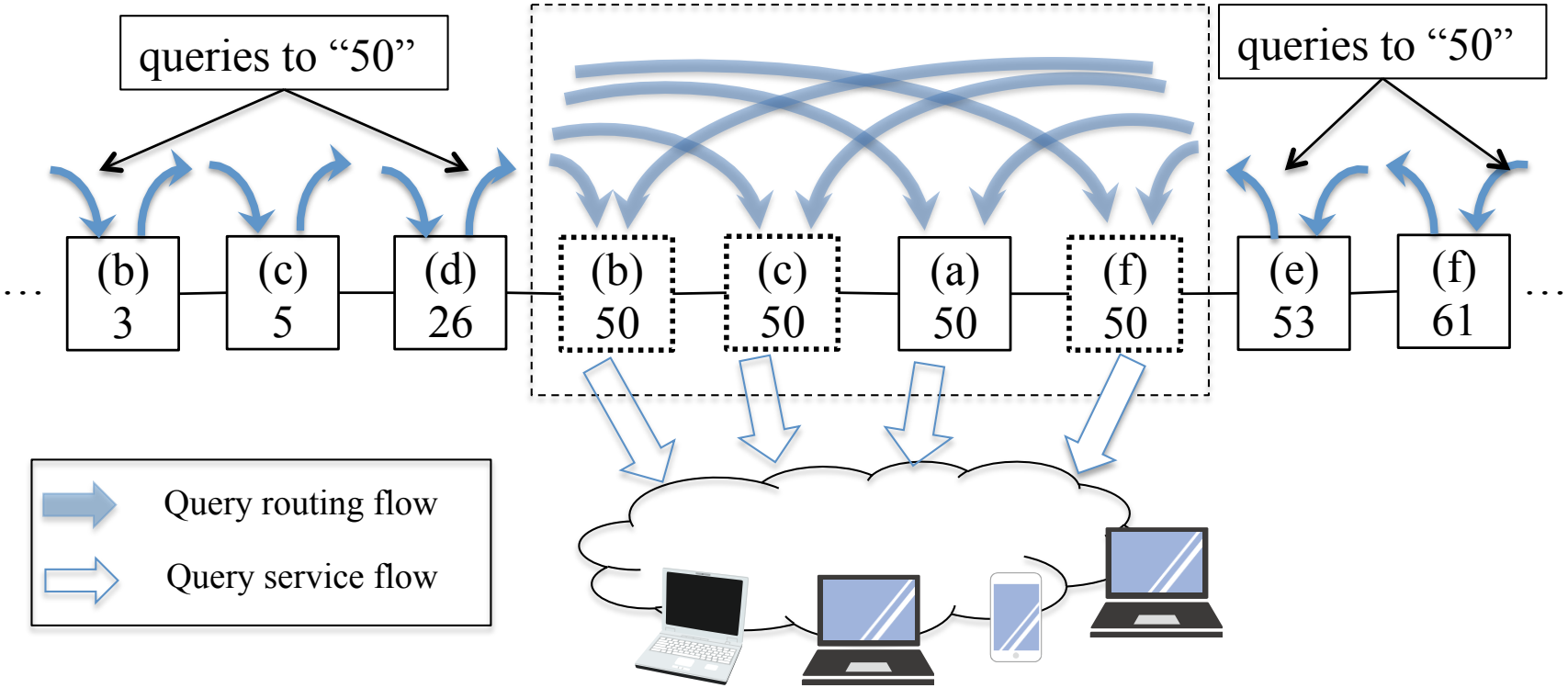
提案手法のイメージ:key は 50 の仮想レプリカノードゾーン(オリジナルノードを含む)



(s) 26 物理ノード ID が s、key が 26 の普通ノード

(a) 50 物理ノード ID が a、key が 50 の仮想レプリカノード

提案手法による負荷分散効果

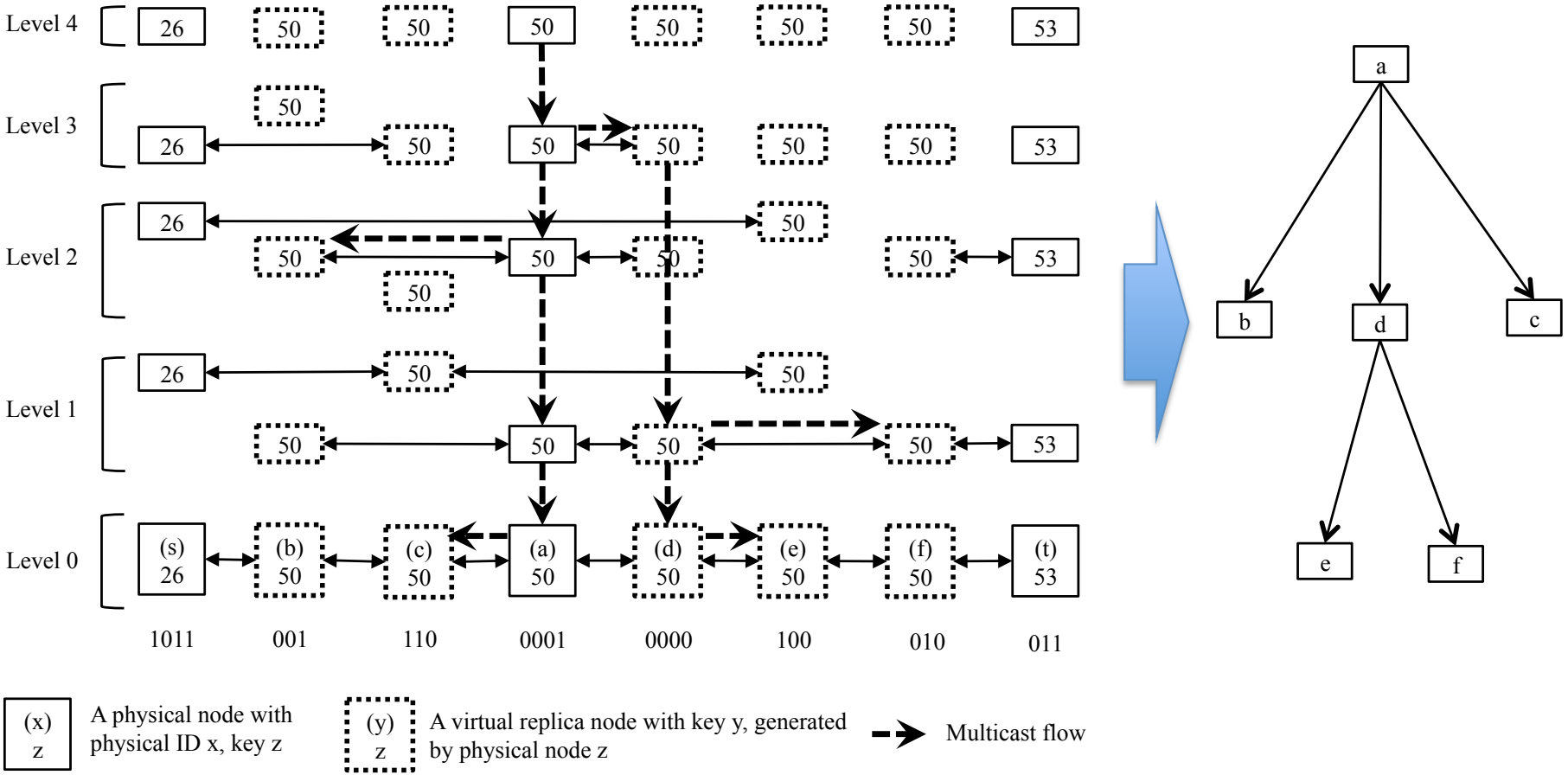


- 定理1：仮想レプリカノード数は M だとすれば、各仮想レプリカノード(hotspot 含む) が担当するquery service load の割合の期待値は：O(1/M)

- 定理2：仮想レプリカノード数は M だとすれば、仮想レプリカゾーンに距離が d のノードが仮想レプリカゾーンへのクエリを転送する確率の期待値は：O(1/(M+d))
  (提案手法でなければ、その確率の期待値は: O(1/d))

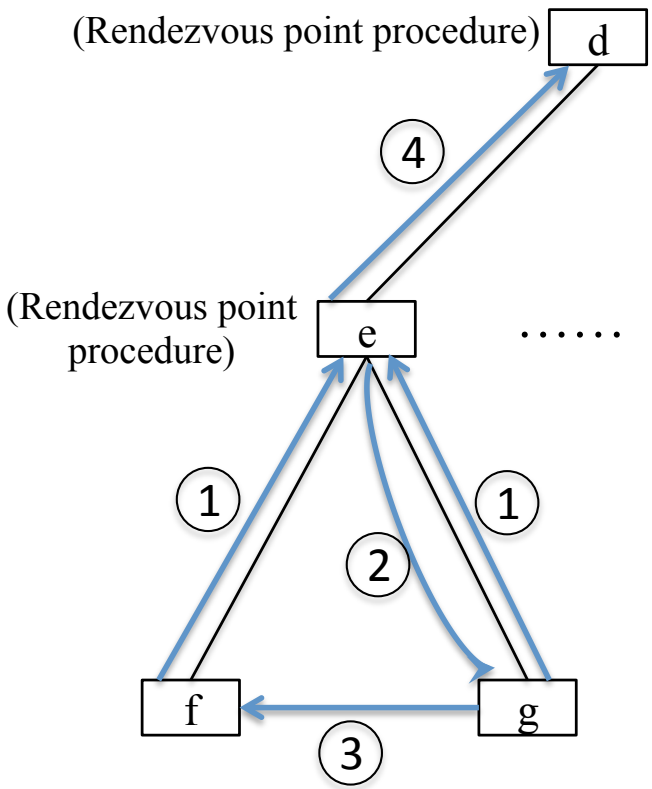
Virtual replica node Tree(VRNT) による オリジナルノードとレプリカノードの同期化

- オリジナルノードをルートとし、各レプリカノードへの multicast tree は VRNT と言い、オリジナルノードと仮想レプリカノードの同期化に用いる



VRNT による負荷の再配分

- 仮想レプリカノード間の query service load 分布の偏りの起因:
  - 仮想レプリカノード間の query service load 分散は確率的である
  - ユーザのクエリ分布には偏りがある場合がある
- 解決方法:周期的にVRNT による仮想レプリカノード間の負荷の再配分:



① 負荷過不足/過負荷の葉ノードは自分の空いているキャパシティ/過負荷の状況を親ノードにレポートする

② 親ノードは過負荷の job を負荷過不足ノードに配分する方法を決定し、過負荷ノードに送信する

③ 過負荷ノードは、親ノードの指示に従って、job の一部(全部)を指定された負荷過不足ノードに転送する

④ まだ負荷過不足/過負荷の子ノードがあれば、親ノードはその情報を自分の親にレポートする

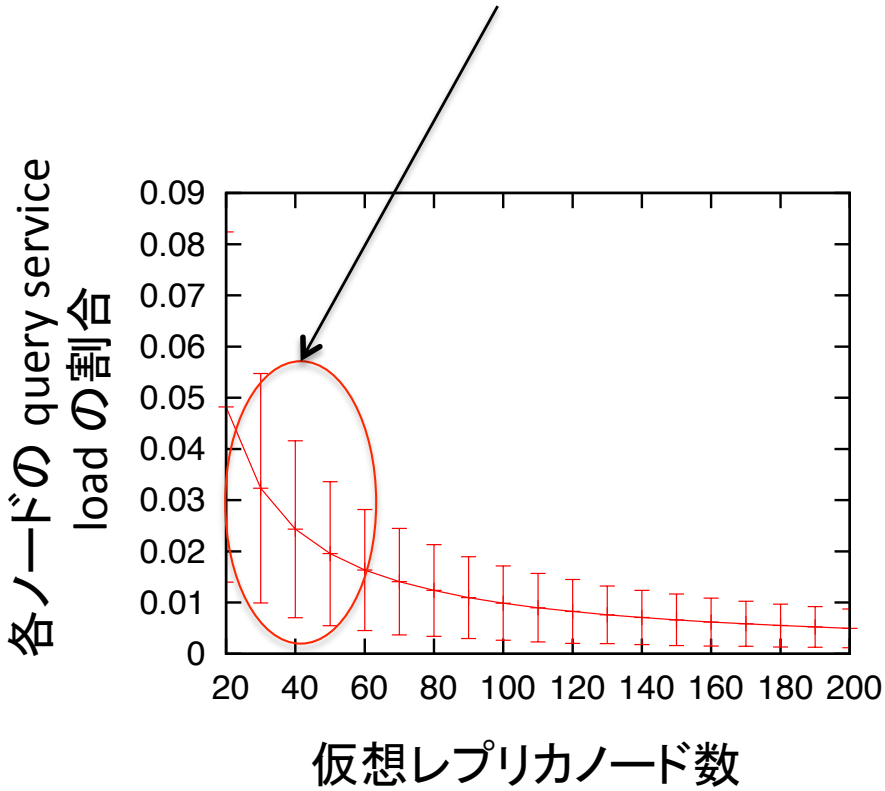
提案手法の評価

- Simulation 設定
  - Totally 10,000 nodes
  - Number of virtual replica nodes: 20 ～ 200
  - Results are from 200 randomly generated Skip Graphs

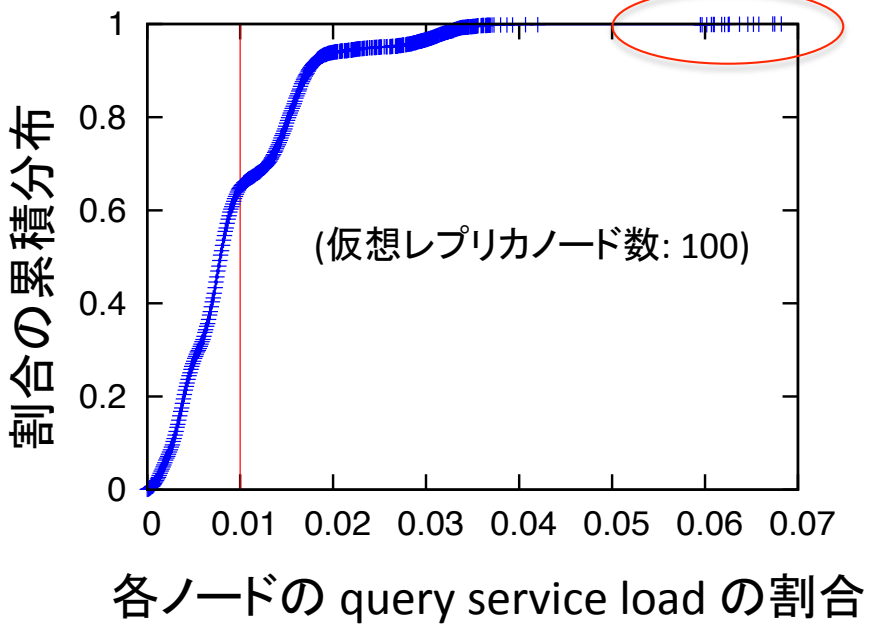
- 評価指標
  - 仮想レプリカノード間の query service load 負荷分散効果
  - Hotspot zone に近くのノードの query routing load 負荷分散効果
  - VRNT 用いた負荷再配分の効果

仮想レプリカノード間の負荷分散効果 (without VRNT による再配分)

Errorbar: 仮想レプリカノード実際の負荷の割合と理想値との標準差



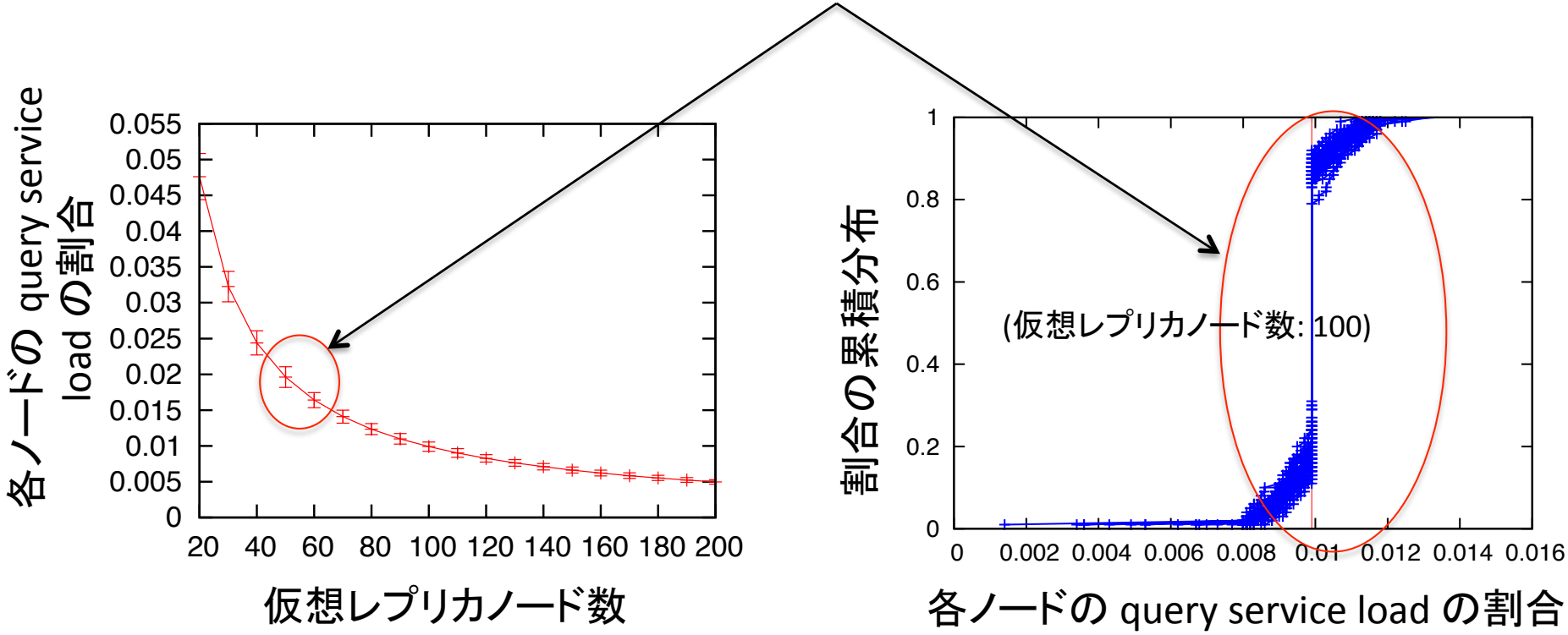
少ないながらも、負荷の割合が大きいノードが偶々出現する



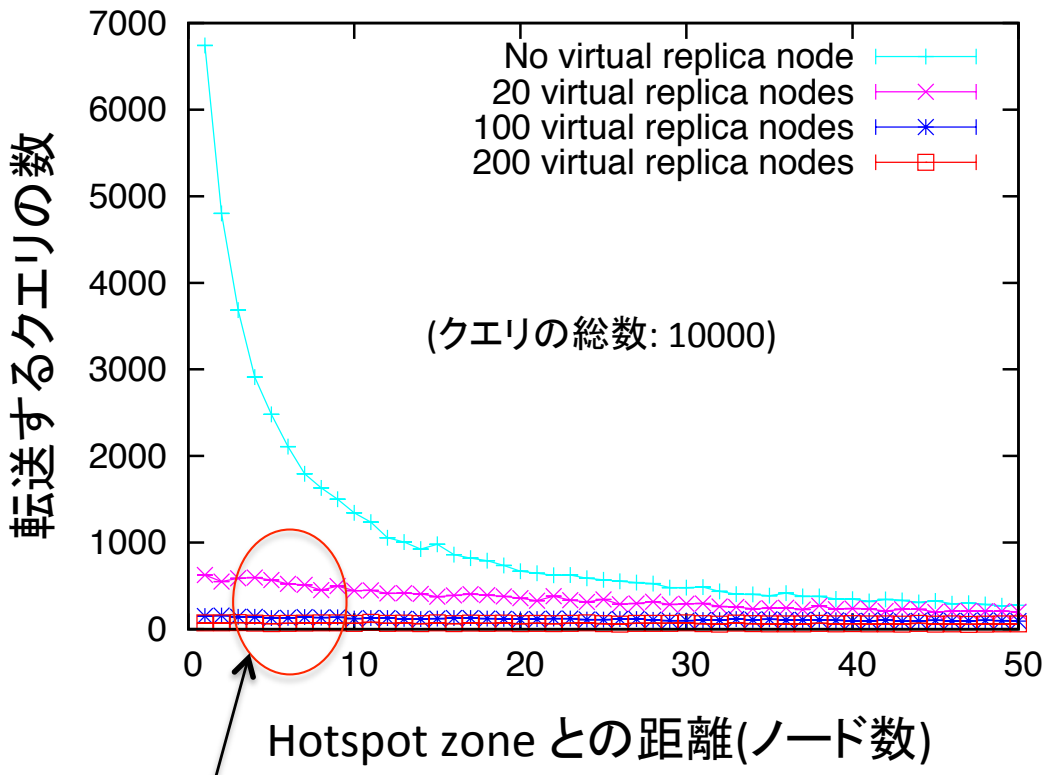
VRNT による負荷の再配分効果

負荷再配分の基準: |ノードの負荷 – 平均負荷| > 平均負荷 * 20%

- VRNT による負荷の再配分により、仮想レプリカノード間負荷の偏りを解消できた



Hotspot zone に近くのノードの query routing load 分散効果



- 仮想レプリカノードを用いることで、hotspot zone に近くのノードの query routing load を著しく軽減できた

Conclusion & Future work

- 仮想レプリカノードを用いて、Skip Graph におけるフレッシュクラウドを緩和させる手法を提案
  - 提案手法は、hotspot ノードの query service load を軽減するのみならず、hotspot の周辺ノードの query routing load を抑えることもできた
  - VRNT を用いて、低コストでオリジナルノードと仮想レプリカノード間の同期化を実現できた
  - VRNT を用いて、仮想レプリカノード間の負荷を再配分することで、負荷分布の偏りを解消できた
  - 解析とシミュレーションの結果によって、提案手法の有効性を確認した
- Future work
  - ユーザーのクエリの統計的特徴に基づいて、ユーザーに物理的に近いノードに仮想レプリカノードを持たせることで、ユーザーとの間の遅延時間を抑える方法の検討