



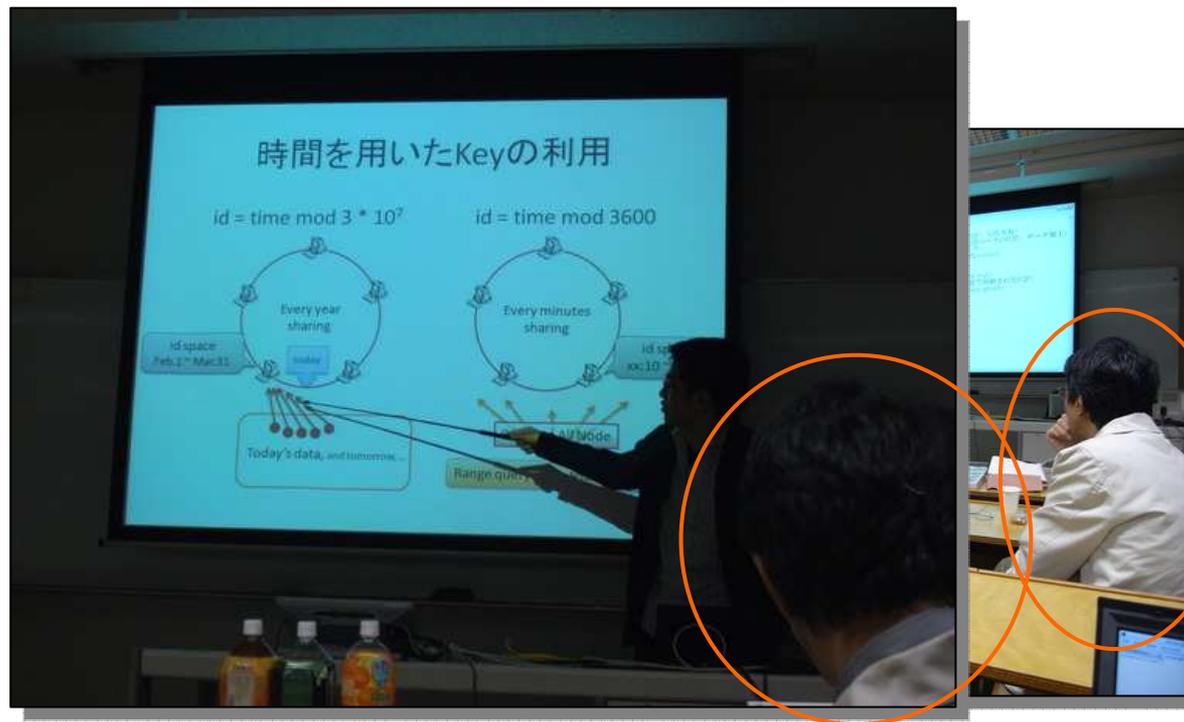
Skip Graph と Chord 等の 共通フレームワーク

首藤 一幸

Y氏の口ぐせ

- 「それは Skip Graph でやった方が」
「それも Skip Graph の方がいい」
「それ Skip Graph で」

「それ Ski」 cf. それ Pla



この人

研究交流会
@奈良先端
2008/4/14

DIS りたくなる

Skip Graph がなんぼの (ry

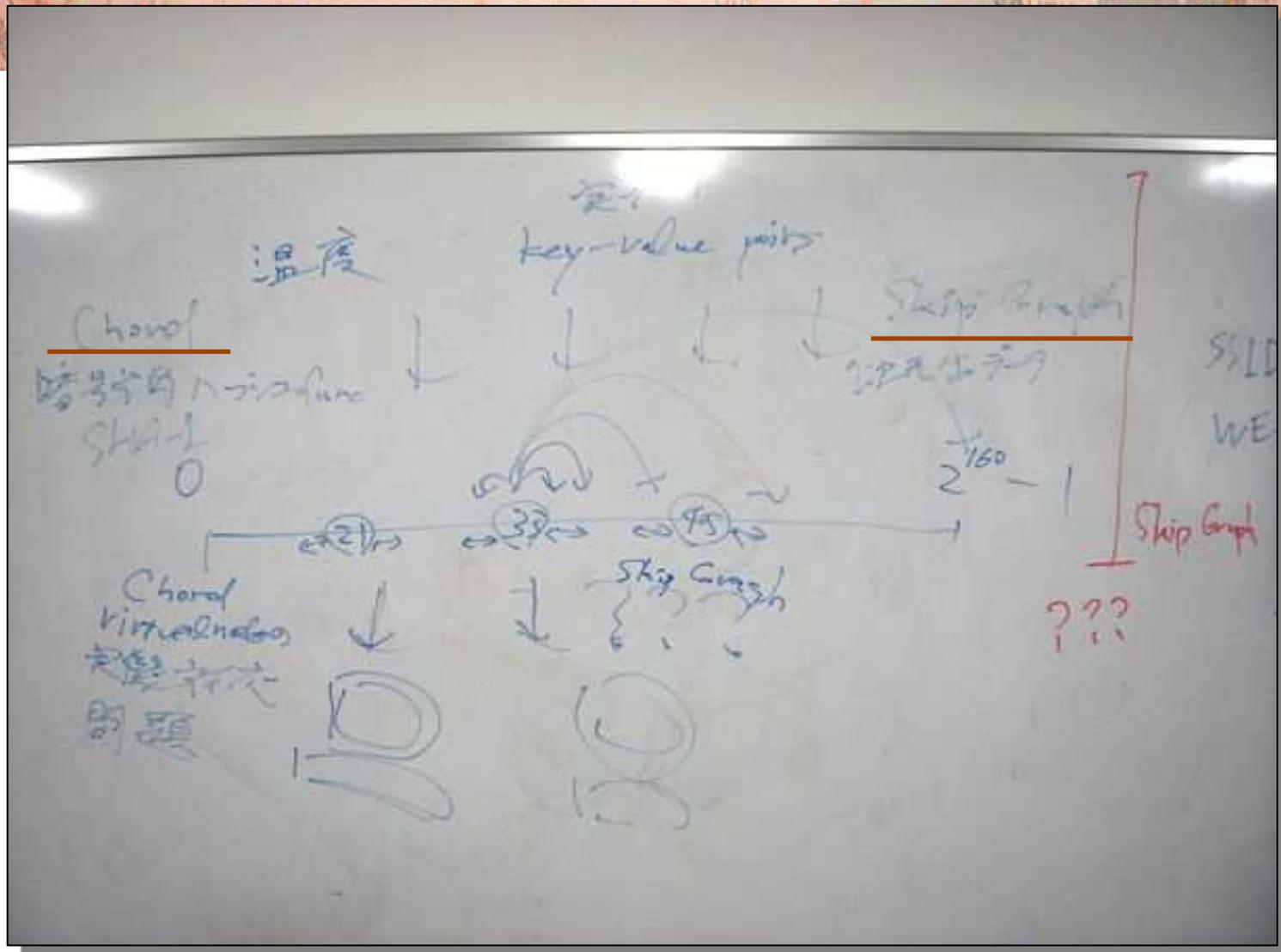
並行 join も大変なくせに！

stabilization (安定化) のアルゴリズムも
持たないくせに！



Skip Graph の強み (だと思われている点) について
よくよく考えてみる

考えてみた



Chord 等 多くの DHT / 構造化オーバレイと Skip Graph の対応付け

Chord 等

Skip Graph

“shudo”

データ (キー)

“shudo”

ID (数値) に変換:
暗号的
ハッシュ関数
(SHA-1 とか)

気温とか氏名とか...

○ 負荷分散
× 順序の喪失

0xfd2...

ID

1 対 1

担当ノード決め:
consistent
hashing

0 ~ 2¹⁶⁰ - 1 の値など

ショートカットリンクがあって...
というリンク構造は、どちらも同じ。

ノードと実機の
対応付け

ノード

負荷分散は
ここで何とかする

基本:
1 対 1

virtual node:
多対 1

Skip Graph 自体は、
負荷分散を扱ってない。

? 論文の
スコープ外!



実機



実装する人が何か工夫しなければならない!
multi-key Skip Graph (阪大) は、きっと何か工夫してる。

実装時の「負荷分散」と 「範囲検索」のトレードオフ

- トレードオフのどちらを採っているか
 - Chord: 暗号学的ハッシュ関数で負荷分散 → 範囲検索 困難
 - Skip Graph: 負荷分散はスコープ外 → 範囲検索 可能
- Chord で、ハッシュ関数を使わずに、範囲検索を可能にしたっていい。
 - その代わりに、(Skip Graph と同様に) ノードの実機への対応付けの際に、負荷分散を考えざるを得ない。

Further discussion

- Skip Graph での DHT (membership vector 使用)、を、どう位置づけるか？
- Skip Graph の設計を構造化オーバレイの他の方式で採用したら、どうなるか？
- 設計空間の整理 → より自由な設計へ
 - ハッシュ関数使用の有無
 - ショートカットリンクの選択: skip list, finger table, Plaxton mesh, k-buckets, ...

Acknowledgments

- PIAX, 阪大グループ
 - PIAX の開発や、Skip Graph の布教 ☺
 - 吉田さんのご発表
- SIProp / P2P SIP 勉強会, ML
 - 議論してくださる皆様
 - 主宰 今村さん