SAKURA Internet

KVSのマルチテナント化について

さくらインターネット研究所
大久保 修一 <ohkubo@sakura.ad.jp>

2010/12/16

Agenda

SAKURA Internet

- 自己紹介
- NoSQLとKVSについて
- KVSのホスティング
- KVSのマルチテナント化
- KVSテストサービスの紹介
- 応用事例紹介
- テストサービスを運用してみて
- まとめ

自己紹介

SAKURA Internet

- さくらインターネット研究所　所属
- 研究所の目的
  - インターネット技術に関する基礎研究および応用研究を行い、成果の発信と利用に努めることにより、会社とその事業の発展に寄与する。
  - 数年後のサービスのネタになりそうな技術の評価等
- トピック
  - IPv4枯渇（IPv6、トランスレータ、自動トンネル技術）
  - クラウド（仮想化、NoSQL、分散ストレージ）

NoSQLの広まり

SAKURA Internet

- 高速
- スケールアウト
- 冗長性
- 単純なデータ構造

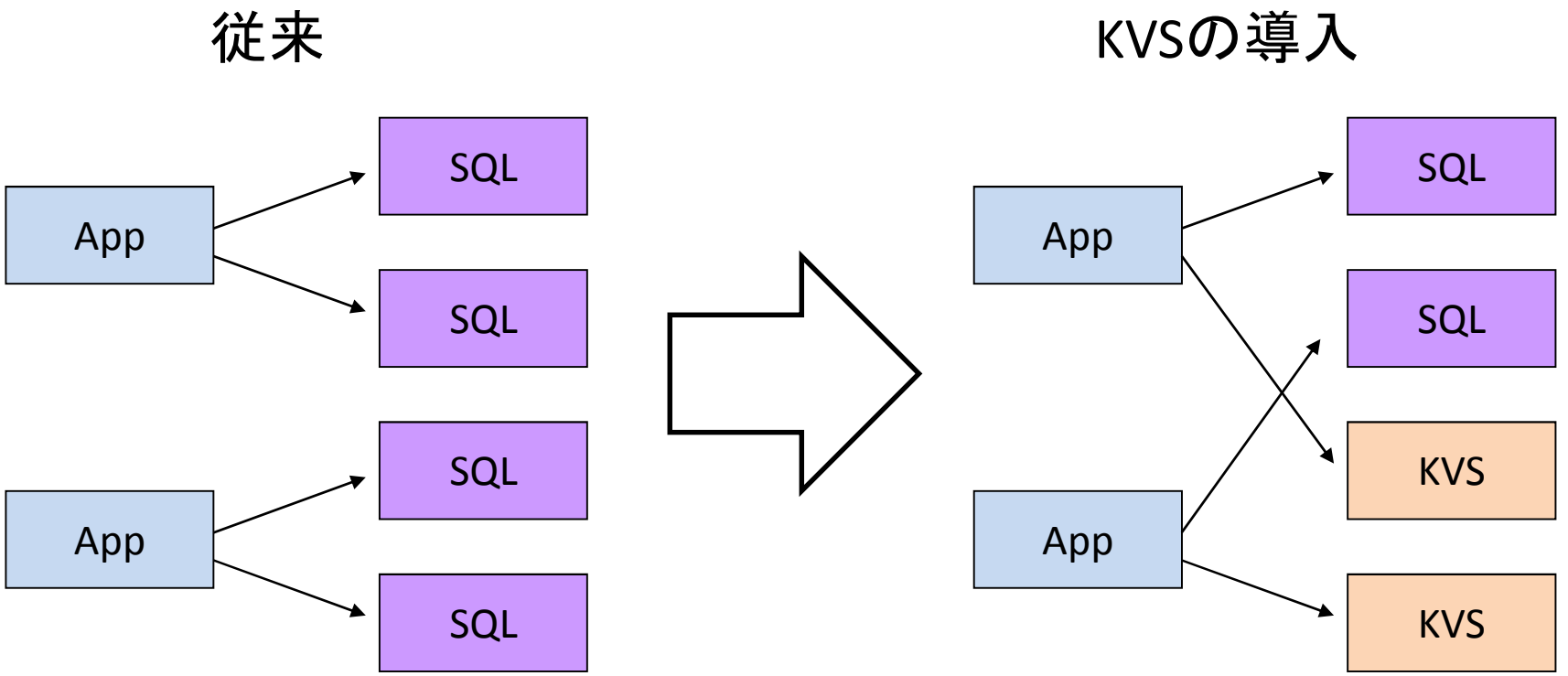
  SQLほどの高度なデータ構造や検索は必要ない
- KVSはNoSQLの1つ

KVSとは？

SAKURA Internet

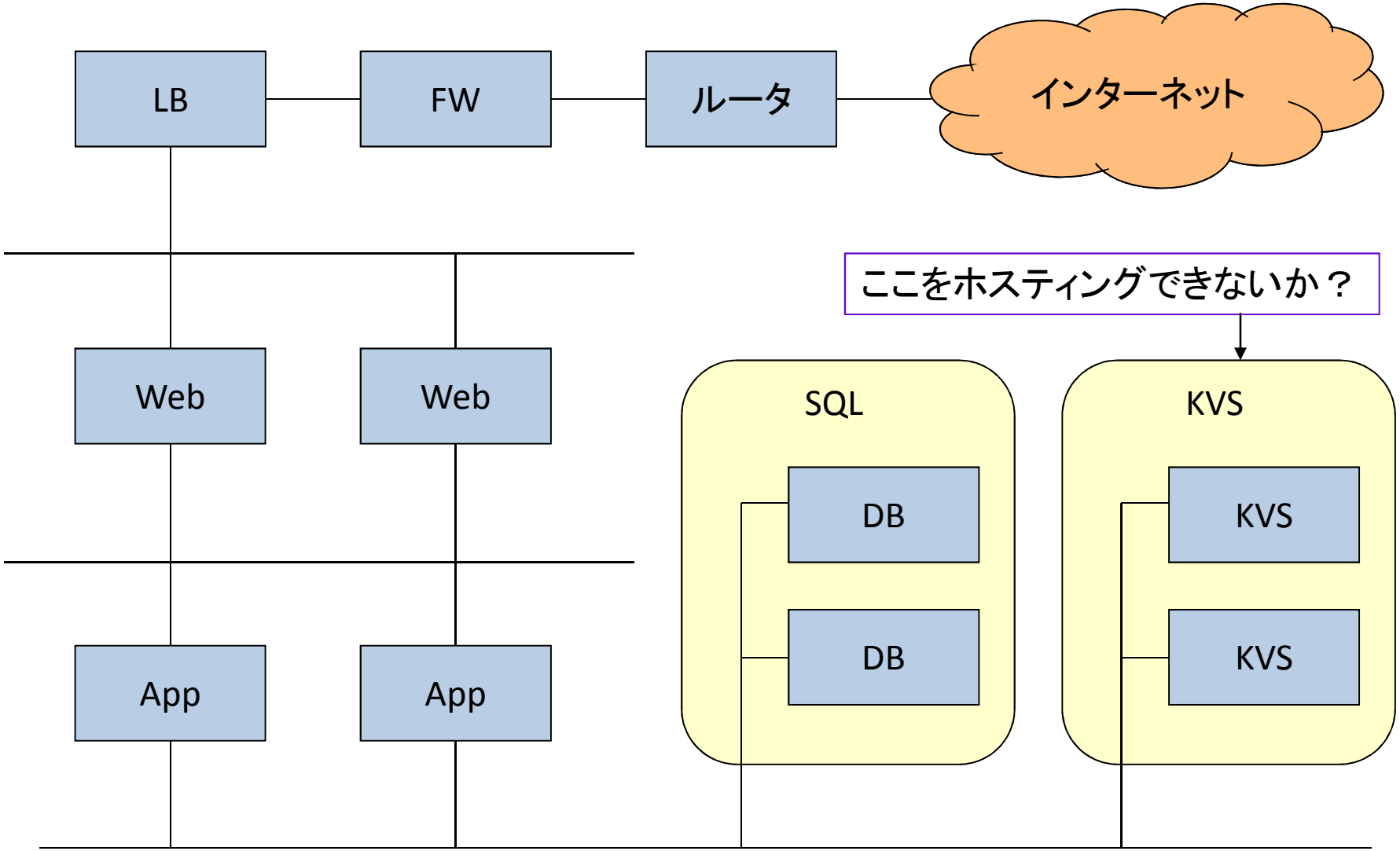
- KVS(=Key Value Store)
- キーと値でのみ表現されるシンプルなデータベース
  - データ構造は連想配列に近い
  - name →ohkubo
  - age →　30
  - home →tokyo
- memcachedプロトコルを用いて、ネットワーク経由で利用できる。
- ストレージサーバの分散が可能なKVSもある。
- 大量のデータを高速に扱える。

SQLからKVSへの移行

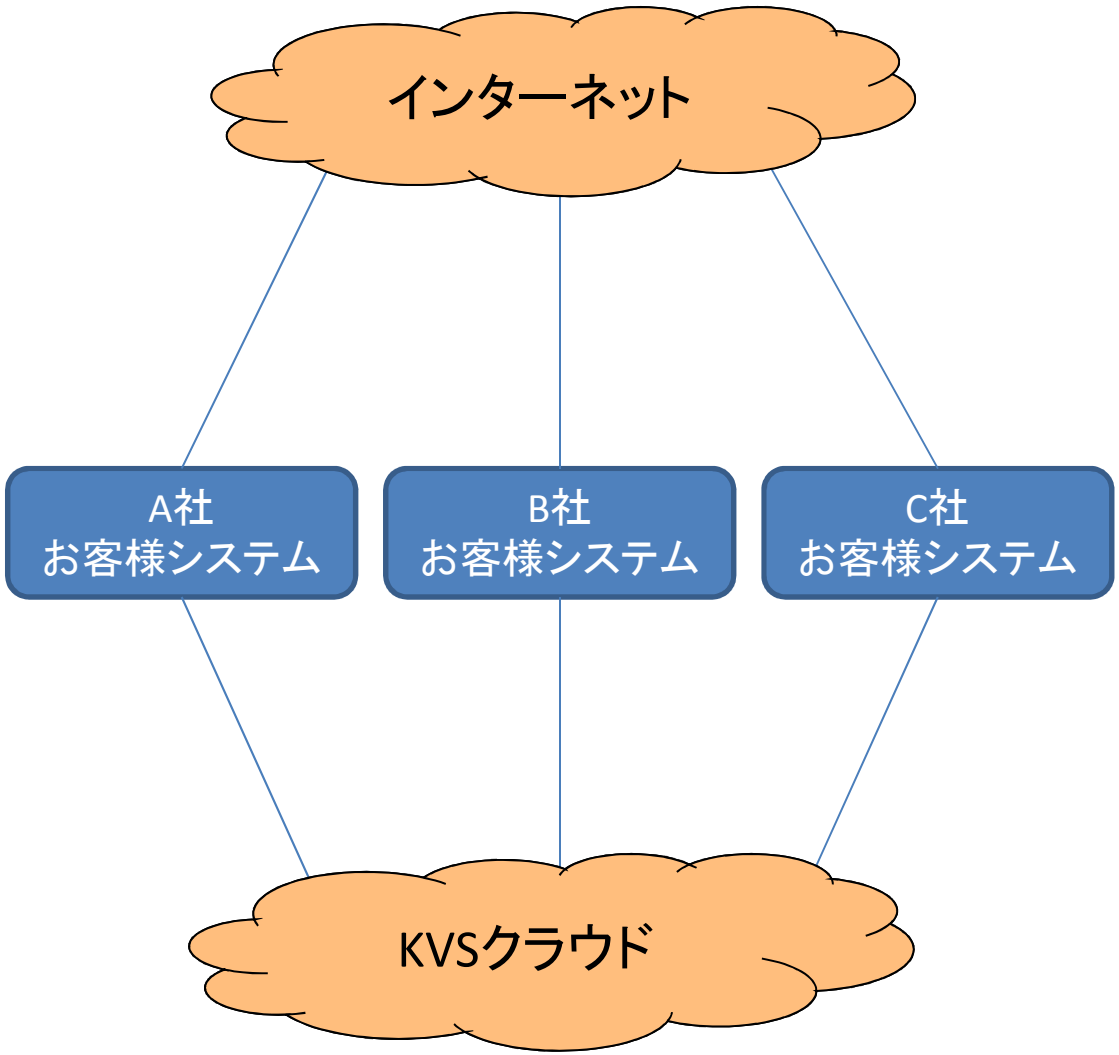


速度が求められる一部のデータをKVSに移行

一般的なユースケース



KVSホスティングのイメージ



利用者のメリット

- サーバの初期投資
コスト不要
- サーバ増設、
メンテナンス不要
- データバックアップ
作業不要

⇩

- 開発に集中
できる！

KVSホスティングの機能要件

SAKURA Internet

- 課金情報取得
  - 課金に必要な統計情報(容量/アクセス回数)の取得
- スケールアウト
  - サーバの単純な追加で簡単に
- データのレプリケーション
  - 格納されたデータが自動的に複数のサーバにコピー

一般的なKVSソフトウェアには実装されている

- マルチテナント機能
  - 複数の顧客のデータを同じサーバ群に格納し、論理的に分離
- 認証機能
  - 顧客ごとの接続認証

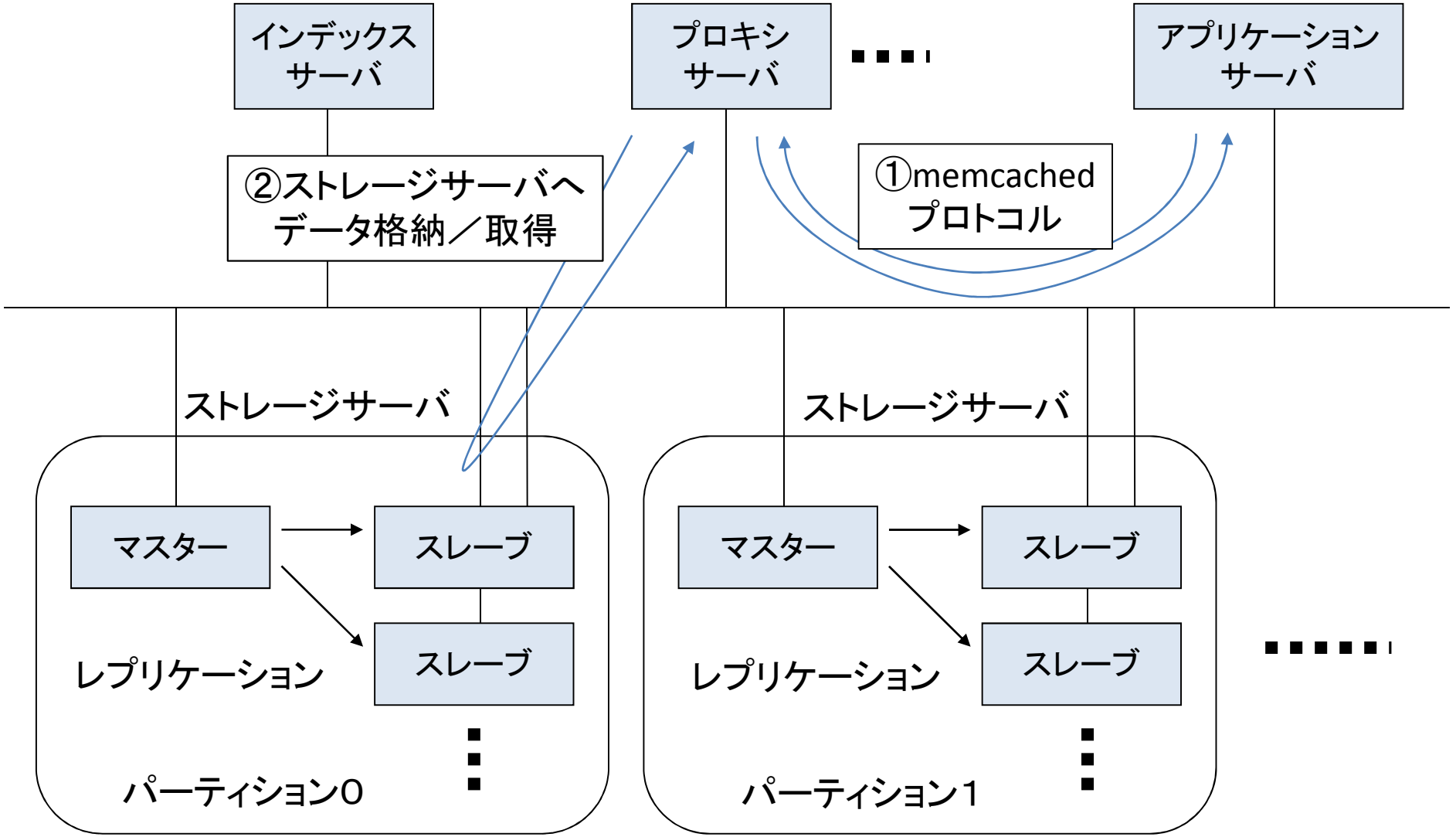
要検討

マルチテナント化の検討

SAKURA Internet

- 一面の物理サーバ群、物理ネットワークの上に、お客様毎に論理分割されたKVSクラスタを重畳したい。
- 今回はグリーさんのFlareを用いて検討

Flareのサーバ構成

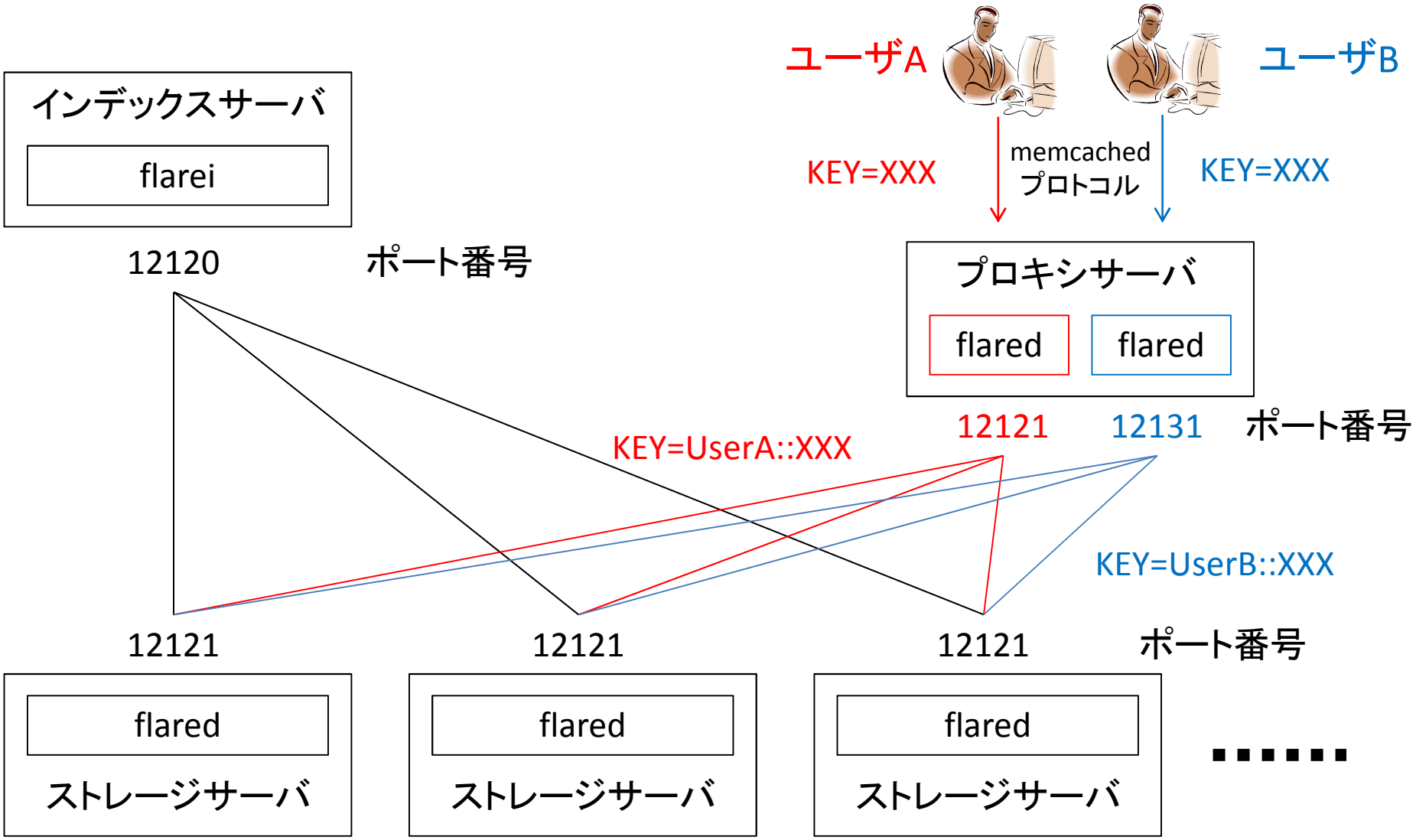


マルチテナント方式

SAKURA Internet

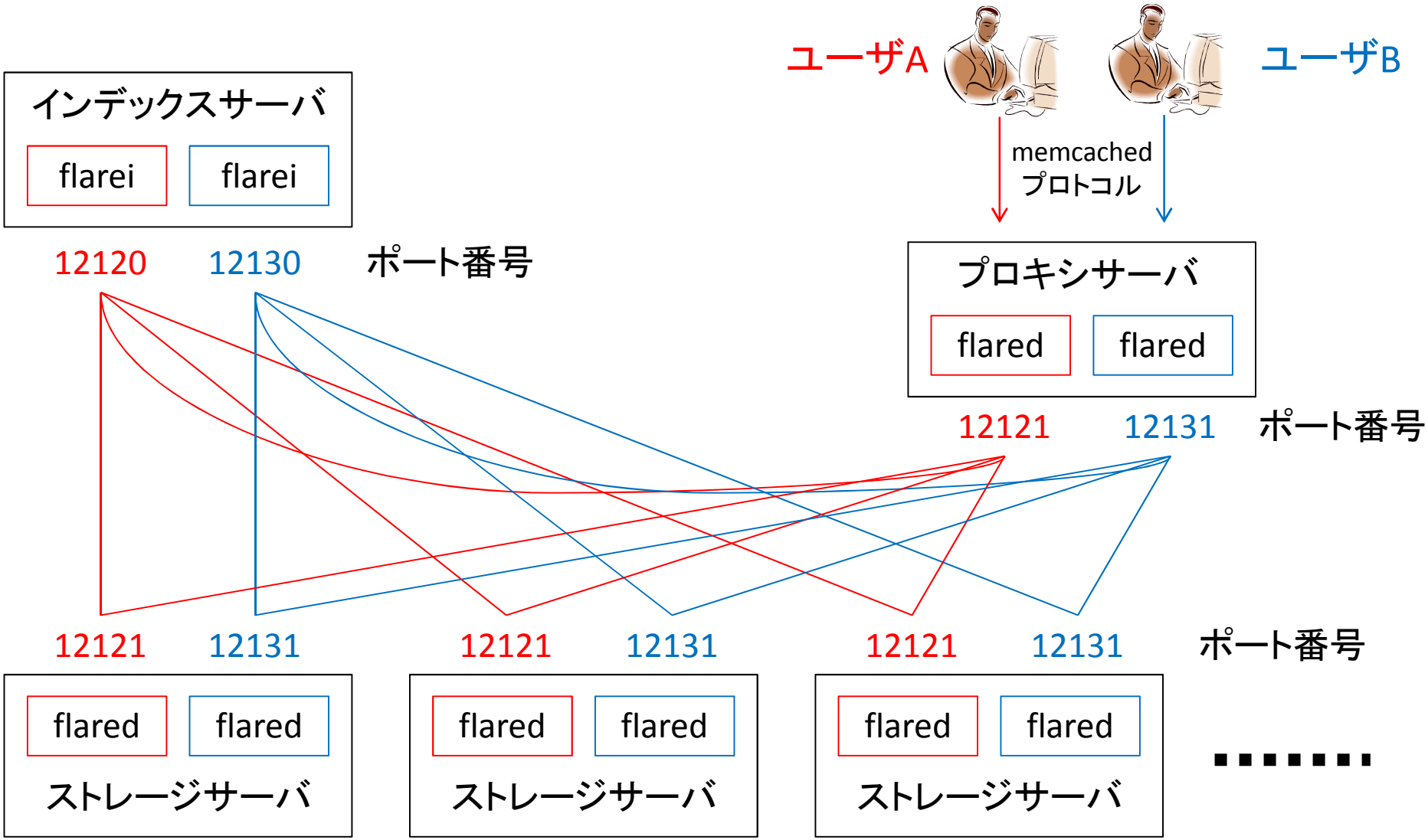
1. キーの名前空間で分離
2. ポート番号を分けて複数プロセス起動
3. 仮想マシン、VLANで分割

キーの名前空間で分離

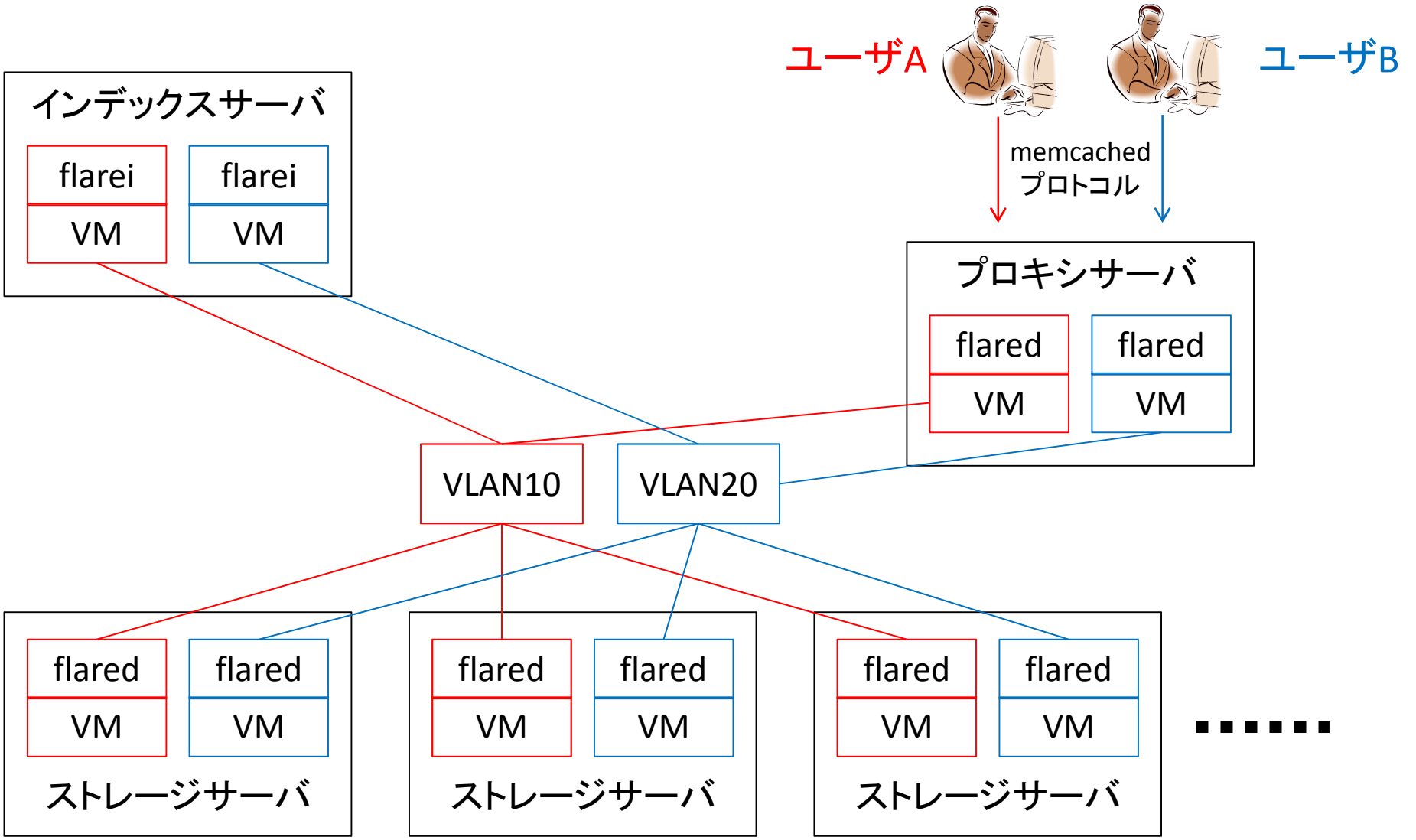


ポート番号を分けて複数プロセス起動

SAKURA Internet



仮想マシン、VLANで分割



各方式の比較

SAKURA Internet

	低オーバーヘッド	隔離性	標準実装
名前空間で分離	◎	△	×
複数プロセス起動	○	○	○
仮想マシン、VLAN	×	◎	○

→ このあたりが妥当なラインか？

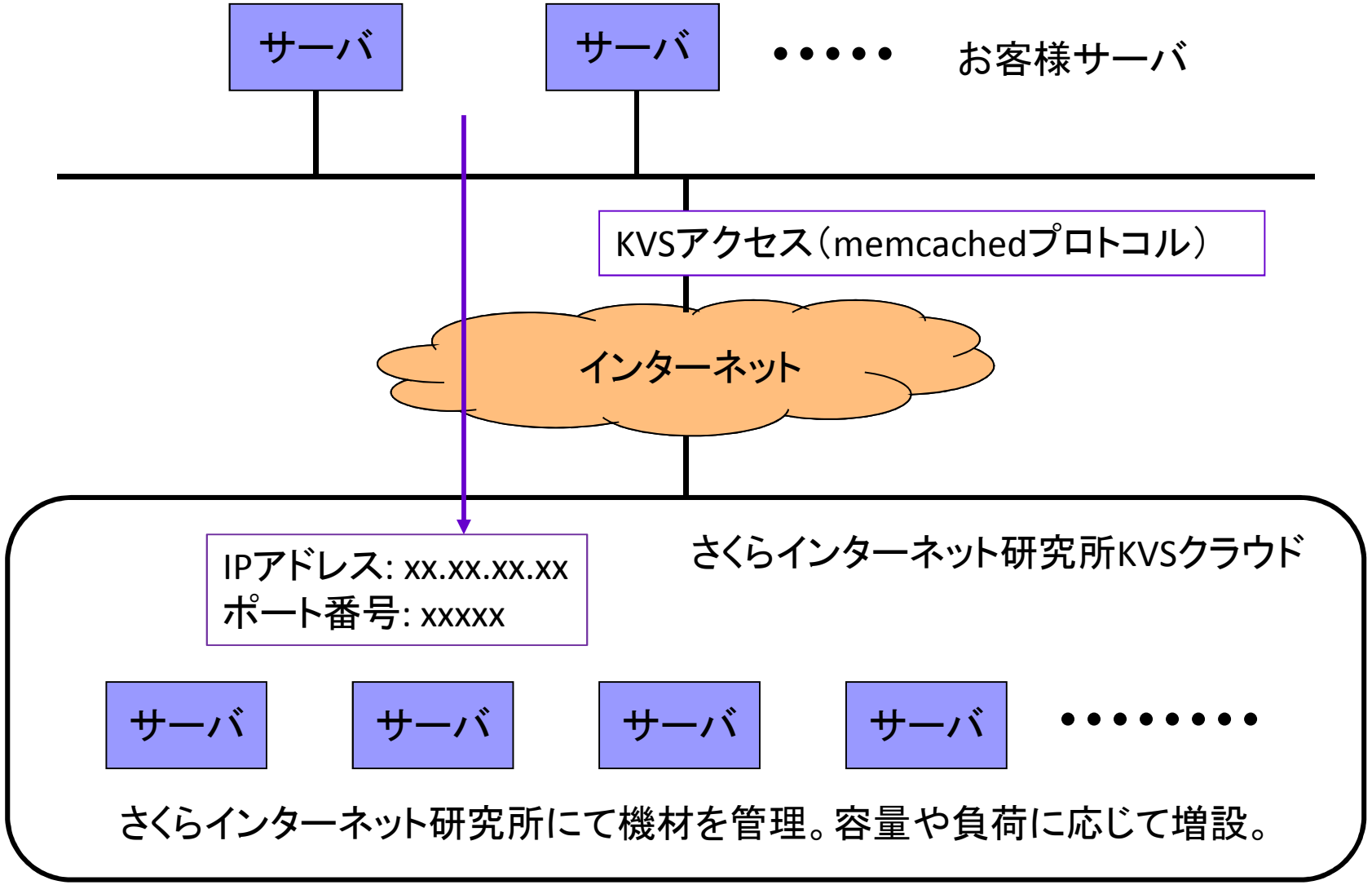
ソフトウェアの改変が必要 ←

KVSテストサービスの紹介

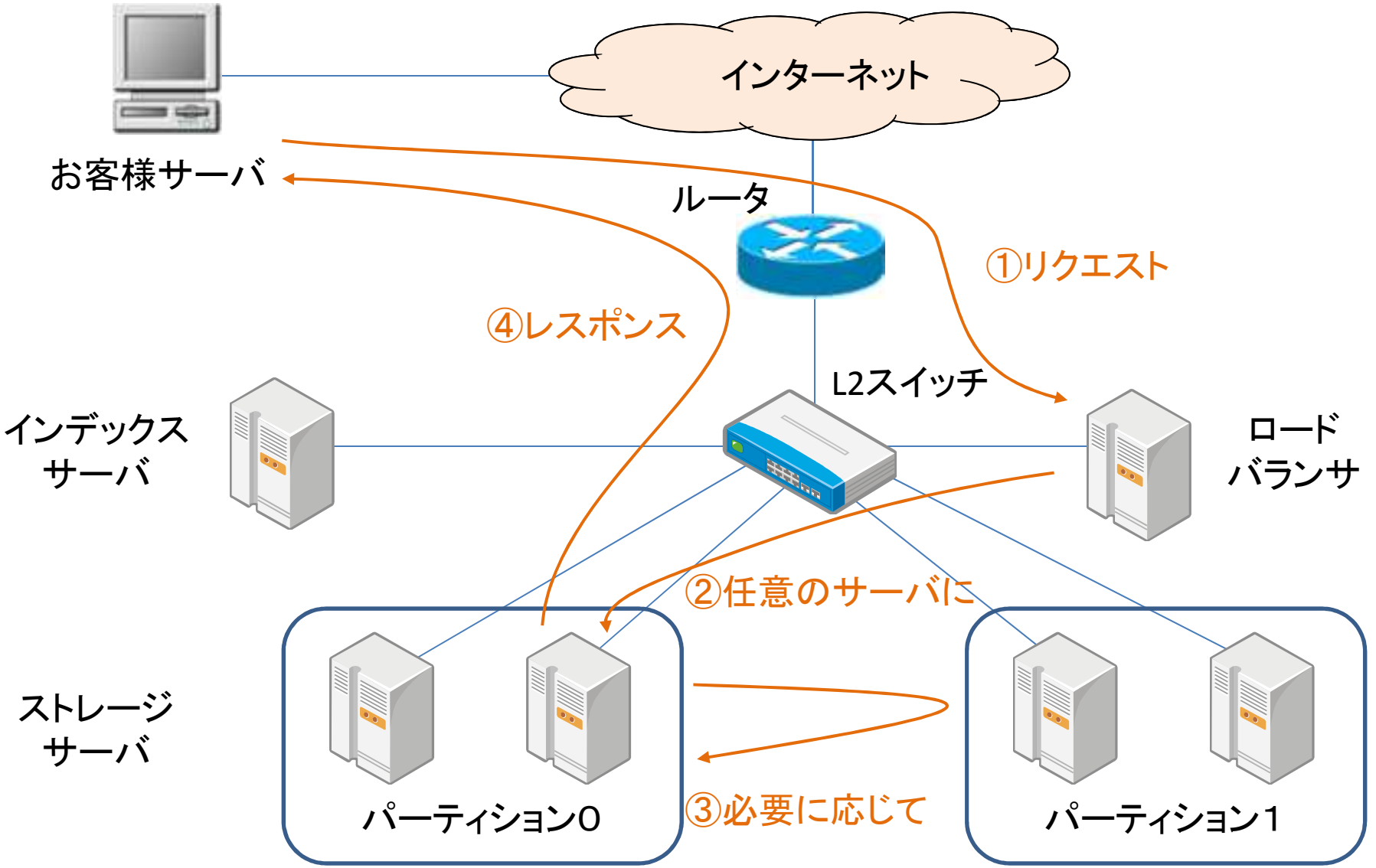
SAKURA Internet

- さくらインターネット研究所にて、2010/7/1より開始（2011/3/31まで）。
- KVSのホスティングサービス。
- インターネット経由でmemcachedプロトコルを用いてアクセス。
- バックエンドとしてグリーさんのFlareを使用。
- お客様毎にポート番号を分けて複数プロセス起動。

KVSテストサービスの利用方法



システム構成



現在のステータス

SAKURA Internet

- 2010/12/9現在
- 16ユーザ
- データ量
  - レコード数：約600万
  - サイズ：約94GB
  - ※ レプリケーションにより、実際はこの2倍

- 現在もテストユーザを募集しています！
  http://research.sakura.ad.jp/kvs-alpha/

応用事例

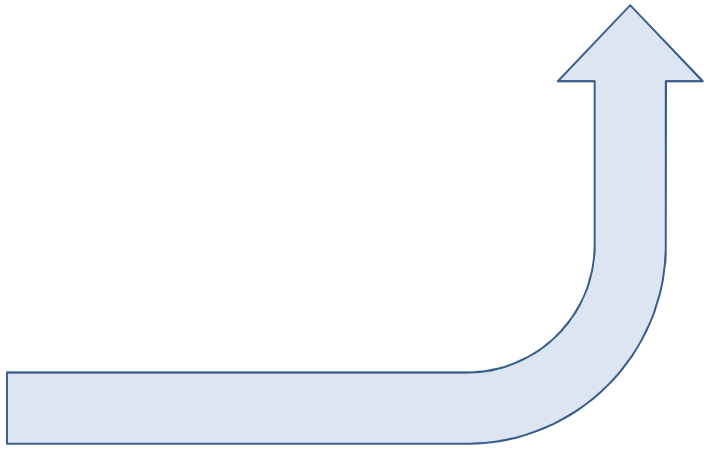
SAKURA Internet

- とある櫻花の画像生成（ジェネレーター）
  http://to-a.ru/
- URL短縮サービス「douj.in」
  http://douj.in/

とある櫻花の画像生成

SAKURA Internet

http://to-a.ru/



URL短縮サービス「douj.in」

SAKURA Internet

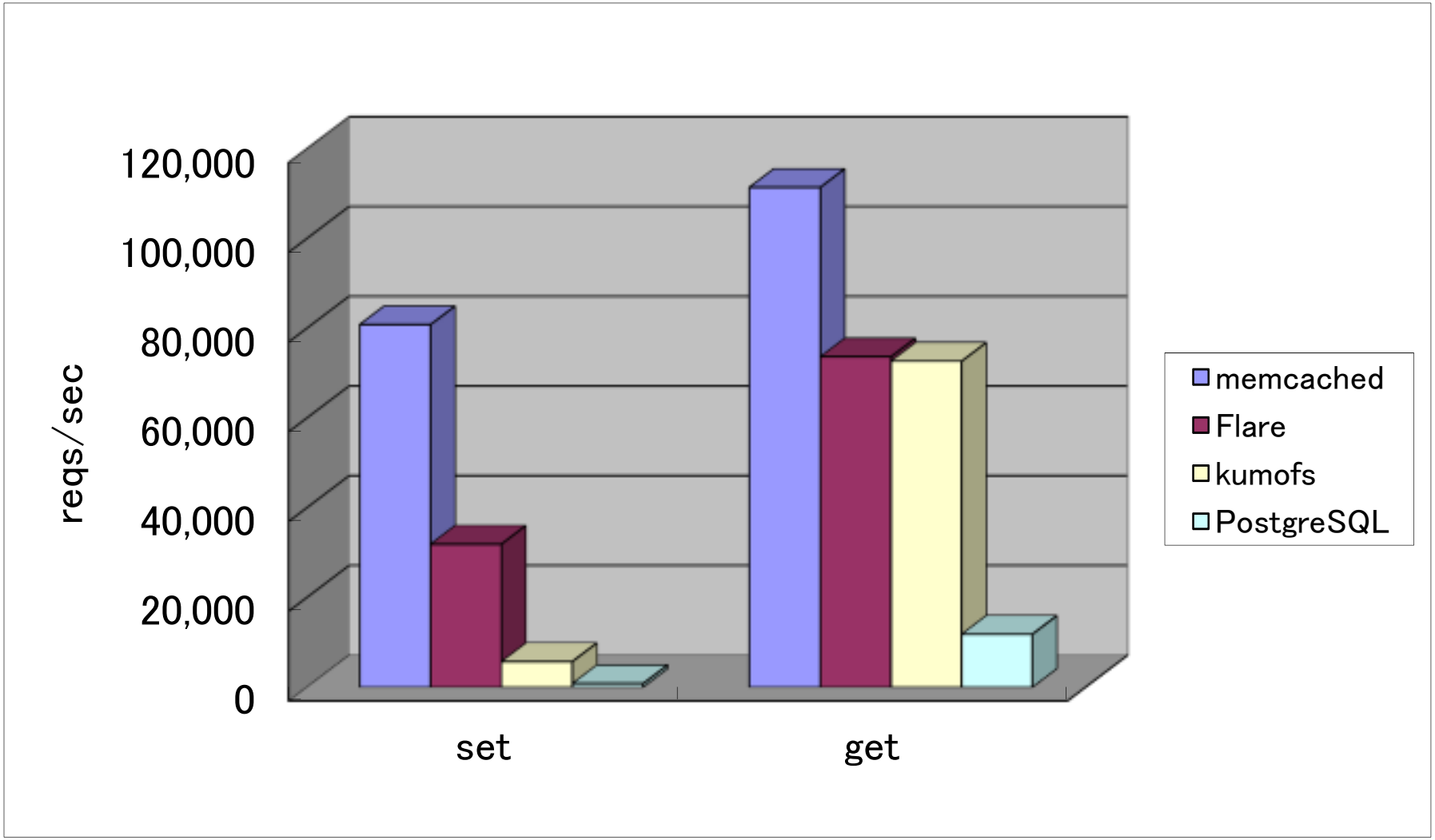
http://douj.in/

他のKVSと比較してFlareの良いところ

SAKURA Internet

- 高速
  - set/get/delete、リビルド時
- SetのflagオプションやCASコマンドも普通に対応
- シンプルなアーキテクチャ
  - パーティショニング＋レプリケーション
  - スレーブサーバ追加、ダウン時の影響が局所的
- サーバ間の通信もmemcachedプロトコル。
  - わかりやすく、デバッグしやすい。
- インストール、設定が簡単。
- 大量のデータ、高負荷でも安定稼働。
  - 当方では1億レコード、200GB程度まで確認。

性能比較

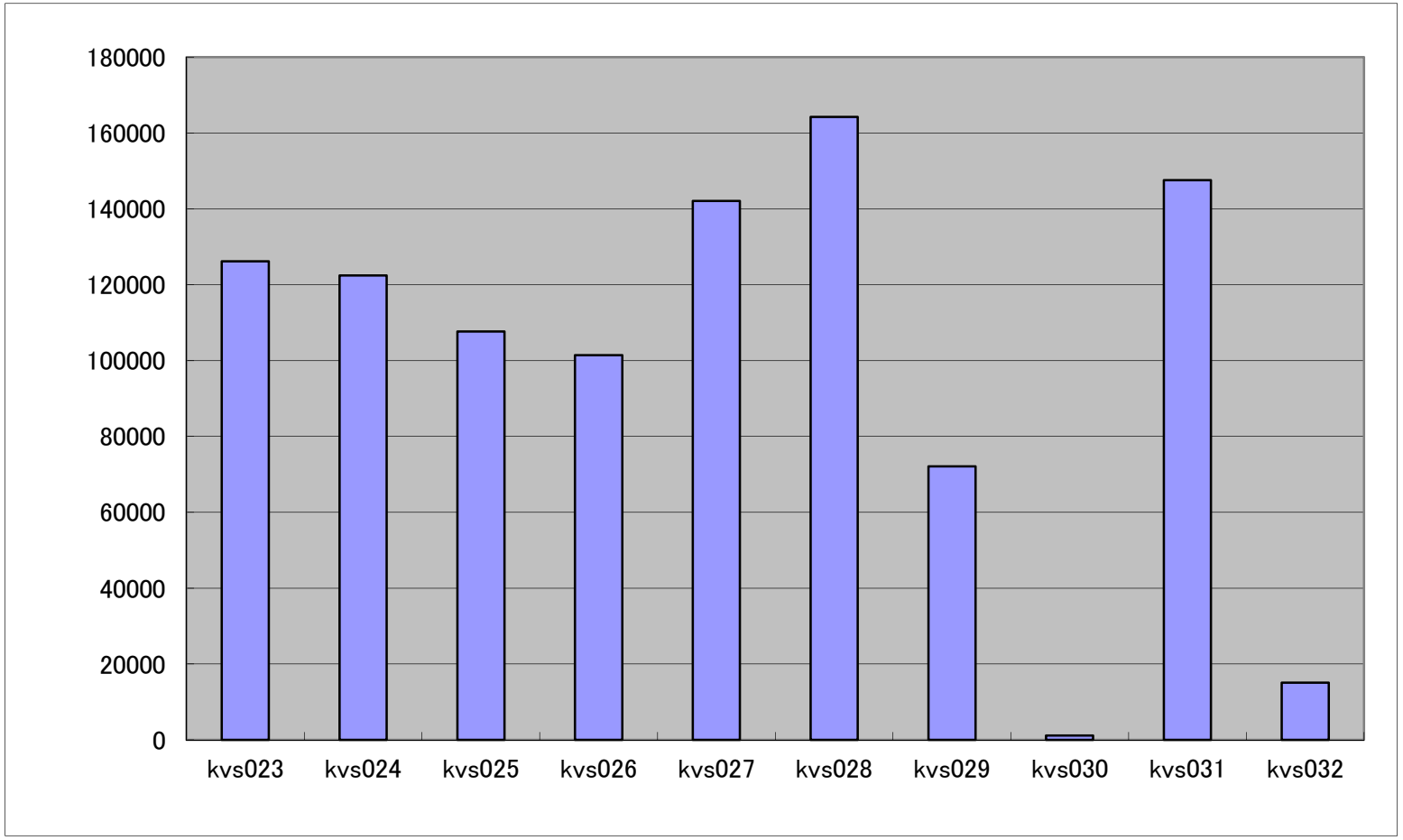


Flareの注意点

SAKURA Internet

- setが完了しても、ディスクには書かれていない。
  - サーバの突然のダウンやディスク溢れ時に、setが完了してもデータがロストすることも。
- レプリケーションが完了する前に、setコマンドが完了する。
  - setが完了しても、直後にgetすると古い値が返ることがある。
- ※ setコマンドのsyncオプションで回避可能
  - 例: set hogehoge 0 0 3 sync
  - 応答時間は若干伸びるが、スループットは変わらない。
  - サーバ側オプションで動作を変更できるとうれしい。
- 単純なキー名（連番など）を使うと、データがうまく分散しない。
  - ハッシュアルゴリズムの問題？

レコードの分散度合測定結果



キー名：test00000000～test00999999（100万レコード）
パーティション数：10

Flareの注意点

SAKURA Internet

- インデックスサーバの冗長化ができない。
- flush_allコマンドがレプリケーションされない。
  - データの不整合が発生する。
  - KVSテストサービスでは、flush_allコマンドを無効にしている。
- 操作インターフェイスがtelnetのみ。
- TokyoCabinetの仕様により、デフォルトで1データベースファイルの容量が64GBまで。
- memcachedバイナリプロトコルには対応していない。
  - ※ テキストプロトコルでも、バイナリデータの格納はできる。

まとめ

SAKURA Internet

- 将来、クラウド基盤に乗っかるPaaSとして、NoSQLを検討している。
- その足がかりとして、現在KVSをマルチテナント化した、KVSのホスティングテストサービスを提供中。
- Flareは結構良い。
- 今後、KVS以外のNoSQLも検討。

付録

Flareのデータ分散方式

SAKURA Internet