キーバリューストアRedisの概要とPostgreSQLとの連携

2022年11月18日
db tech showcase 2022

SRA OSS LLC
佐藤 友章
sato@sraoss.co.jp

自己紹介

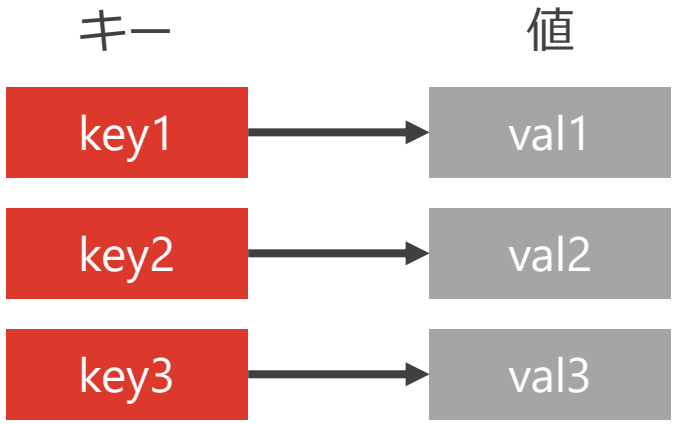
- 佐藤 友章 (さとう ともあき)
- SRA OSSでデータベースおよび技術部門の責任者を務める
- PostgreSQLのサポートやトレーニング、コンサルなど、
  今なおエンジニアとしても活動
- 最近はPostgreSQL以外にNoSQLも手がける
- 趣味はお酒と海外旅行

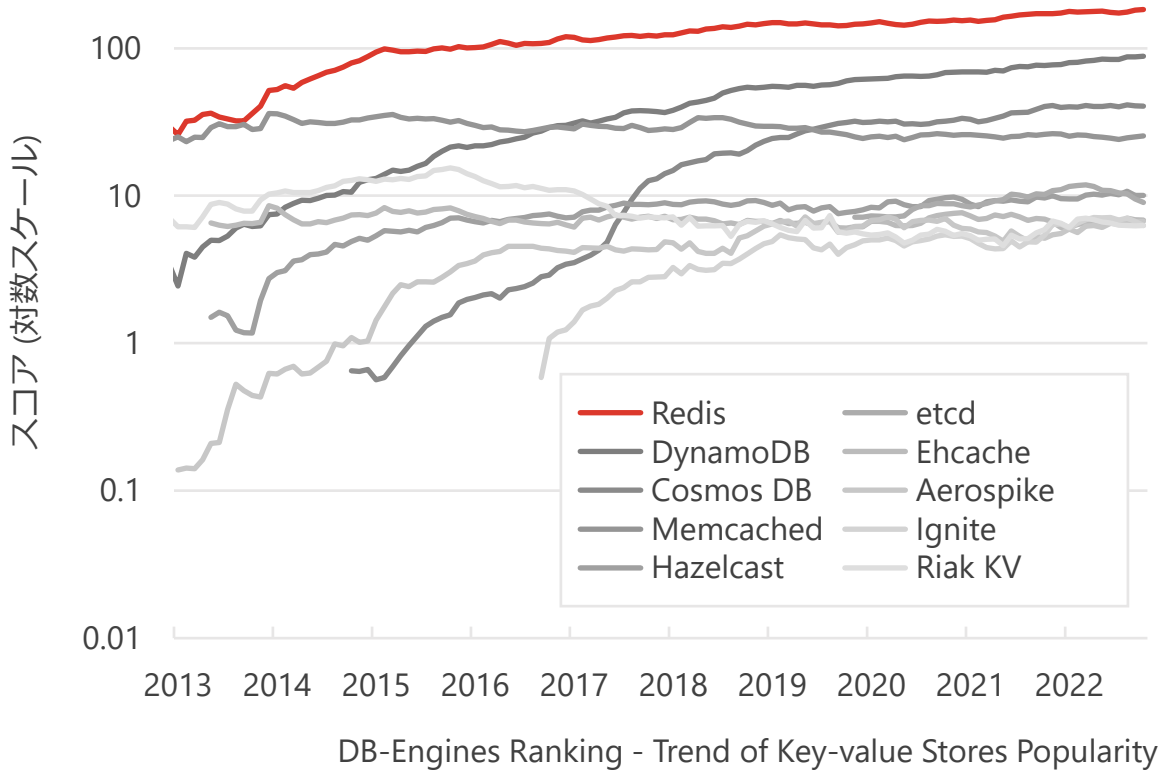
キーバリューストアRedisとは

What's Key-Value Store, Redis

キーバリューストア (KVS) とは

- キーと値のペアを格納する単純な形式のDBMS
- KVSの人気ランキングでRedisは65種中1位

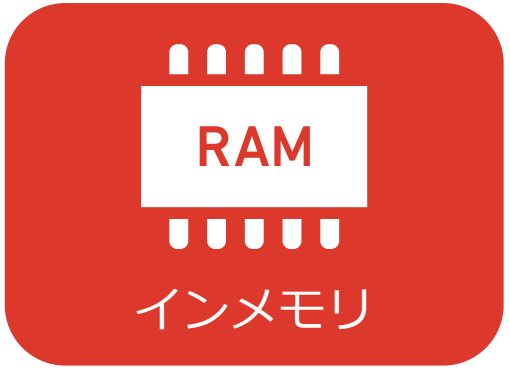


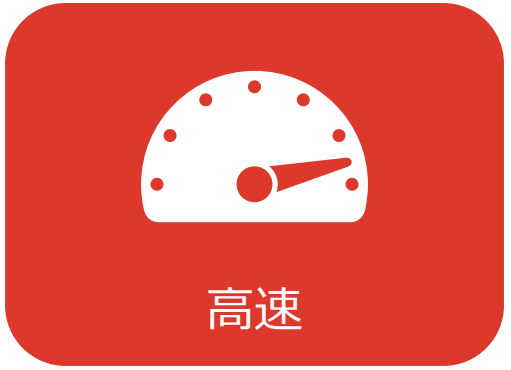


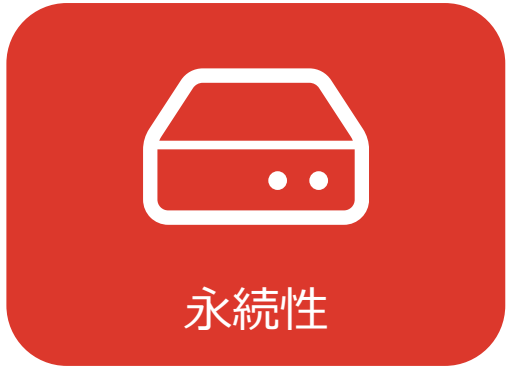
Redis (Remote Dictionary Server) とは

- レディス (red + ~~k~~iss) と読む
- KVSの一種
- もとはSalvatore Sanfilippo氏が開発
- 現在は米国Redis社 (元Redis Labs) が開発を継続
- 最初のバージョンは2009年5月にリリース
- リポジトリはGitHubで公開
- 記述言語はC
- ライセンスは修正BSD

Redisの特徴

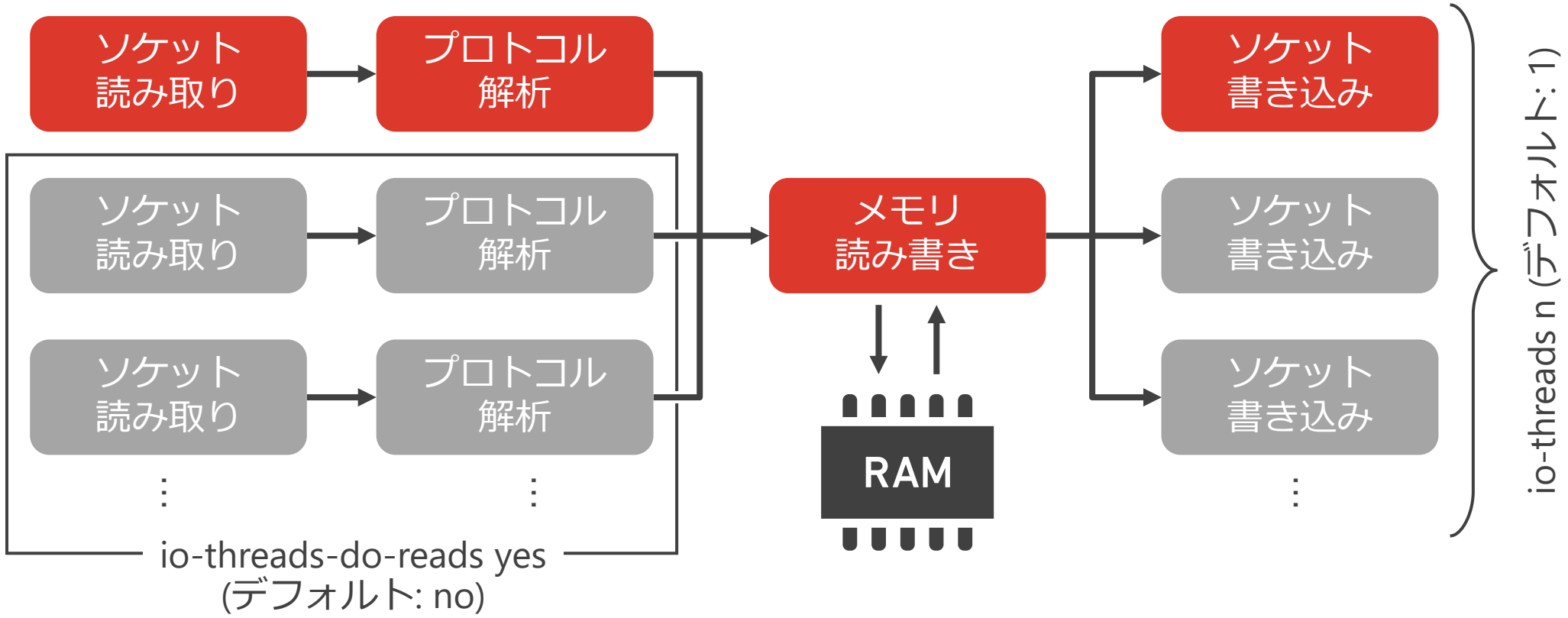






Redisのアーキテクチャ

- メモリ上で動作
- 基本はシングルスレッド

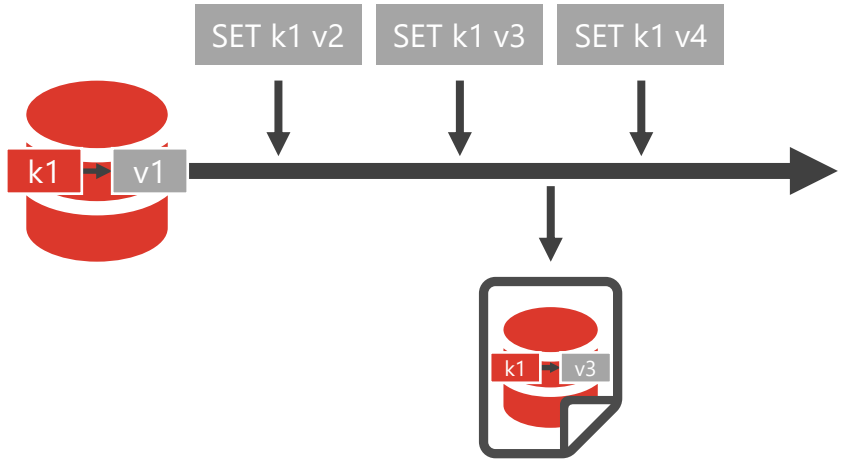


Redisの永続性

メモリ上のデータはサーバ停止時に消失
データを保持するにはディスク書き込みが必要

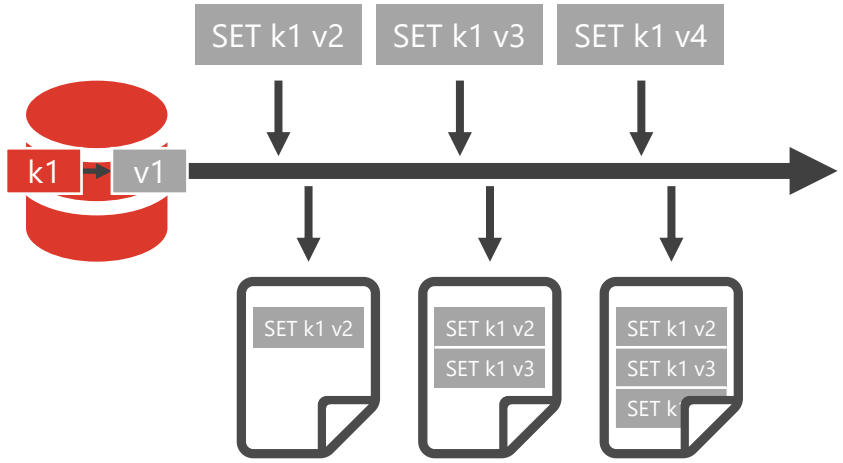
RDB (Redis Database)

スナップショット = ある時点の全データを保存



AOF (Append Only File)

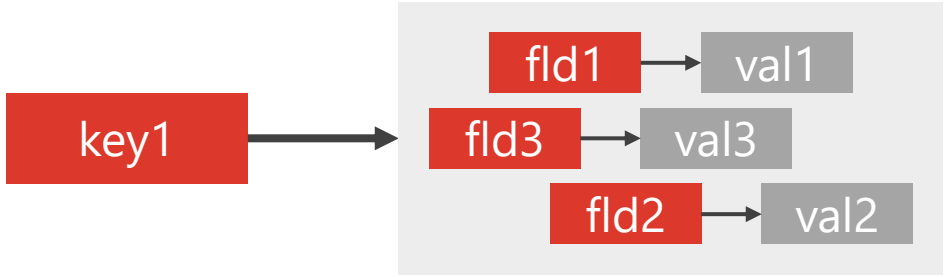
書き込み操作を追記で保存、サーバ起動時に再生



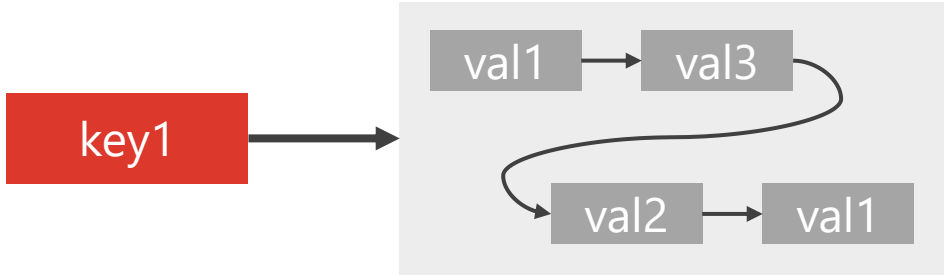
Redisのデータ型

文字列

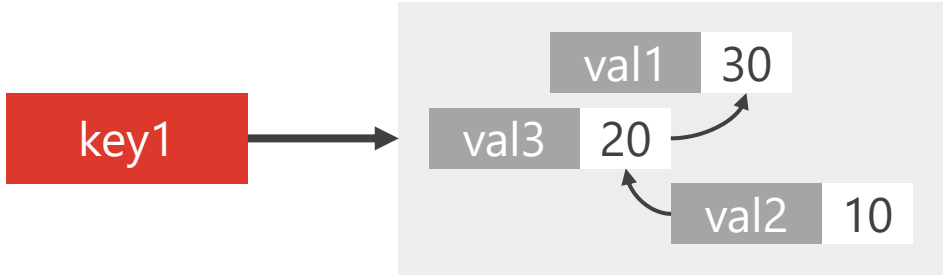
ハッシュ



リスト



ソート済みセット



セット

その他

- ストリーム
- 地理空間インデックス
- ビットマップ
- ビットフィールド
- HyperLogLog

Redisの可用性とスケーラビリティ

レプリケーション

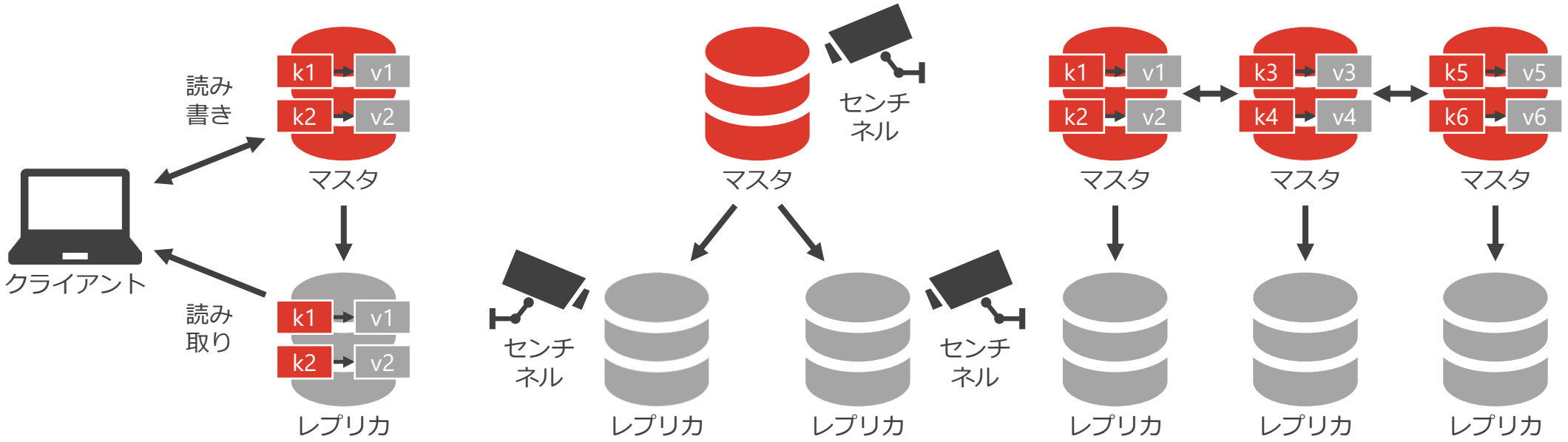
非同期にデータを複製、レプリカは読み取り専用

センチネル

死活監視、多数決で自動フェイルオーバを行う

クラスタ

複数ノードにデータを分散 (シャーディング)



Redisの基本的な使い方

Getting Started with Redis

Redisのインストール

- Linux、 macOSに対応
- Windowsは公式には非対応、 WSL2上で開発用に利用可
- RHEL8の場合

```
$ sudo dnf -qy install redis                    # パッケージをインストール

Installed:
  redis-5.0.3-5.module+el8.5.0+657+2674830e.x86_64

$ sudo systemctl enable redis.service           # サーバの自動起動を有効化
Created symlink /etc/systemd/system/multi-user.target.wants/redis.service → /us
r/lib/systemd/system/redis.service.
$ sudo systemctl start redis.service            # サーバを起動
```

redis-cliコマンド

- Redisのコマンドラインインタフェース
- コマンドは引数に指定するか、対話的に入力して実行

```
redis-cli [オプション] [コマンド [引数 [引数 …]]]
```

```
$ redis-cli SET key1 val1                    # コマンドを引数に指定して実行
OK
$ redis-cli                                  # 対話モードで起動
> GET key1
"val1"
> DEL key1
(integer) 1
> QUIT
```

おもなオプション
- -h ホスト名　デフォルト: 127.0.0.1
- -p ポート　デフォルト: 6379
- -a パスワード　パスワード認証が有効な場合に指定
- -d DB番号　0～15、デフォルト: 0

Redisのおもなデータ操作コマンド (1/3)

共通

コマンド	説明
KEYS パターン	パターンに一致する全キーを取得
EXISTS キー …	キーが存在するかを取得
DEL キー …	キーを削除
RENAME キー 新キー	キーを別名に変更
TYPE キー	値のデータ型を取得

データ型に応じて
コマンドの使い分けが必要

文字列

コマンド	説明
SET キー 値	指定したキーの値を設定
SETNX キー 値	キーが存在しない場合に値を設定
GET キー	指定したキーの値を取得
MGET キー …	指定した複数のキーの値を取得
GETSET キー 値	指定したキーの値を設定し、古い値を取得
INCR キー	値が整数の場合に1増やす
INCRBY キー 数	値が整数の場合に指定した数増やす
SUBSTR キー 開始 終了	値の部分文字列を取得

Redisのおもなデータ操作コマンド (2/3)

リスト

コマンド	説明
LPUSH キー 要素 …	リストの先頭に要素を追加
LPOP キー	リストの先頭から要素を削除し、取得
LSET キー 位置 要素	リストの指定した位置の要素を設定
LINDEX キー 位置	リストの指定した位置から要素を取得
LRANGE キー 開始 終了	リストから指定した範囲の要素を取得
LREM キー 数 要素	リストから要素を削除
LLEN キー	リストの長さを取得

セット

コマンド	説明
SADD キー メンバ …	セットにメンバを追加
SRANDMEMBER キー	セットからランダムにメンバを取得
SMEMBERS キー	セットから全メンバを取得
SISMEMBER キー メンバ	セットにメンバが含まれるかを取得
SREM キー メンバ …	セットからメンバを削除
SCARD キー	セットのメンバ数を取得
SINTER キー …	複数セットに共通するメンバを取得

Redisのおもなデータ操作コマンド (3/3)

ハッシュ

コマンド	説明
HSET キー フィールド 値 …	ハッシュのフィールドに値を設定
HGET キー フィールド	ハッシュからフィールドの値を取得
HKEYS キー	ハッシュから全フィールドを取得
HEXISTS キー フィールド	ハッシュにフィールドが存在するかを取得
HDEL キー フィールド …	ハッシュからフィールドを削除
HLEN キー	ハッシュのフィールド数を取得

ソート済みセット

コマンド	説明
ZADD キー スコア メンバ …	ソート済みセットにメンバを追加
ZRANGE キー 開始 終了	ソート済みセットから指定した範囲のメンバを取得
ZREM キー メンバ …	ソート済みセットからメンバを削除
ZCARD キー	ソート済みセットのメンバ数を取得
ZSCORE キー メンバ	ソート済みセットのメンバのスコアを取得
ZINCRBY キー 数 メンバ	ソート済みセットのメンバのスコアを指定した数増やす

Redisのトランザクション

- 複数コマンドを後でまとめて実行
- 実行中に他コマンドをはさまず、すべて成功か、すべて失敗かのいずれかが保証される

```
> MULTI
OK
> SET key2 val2
QUEUED
> SET key3 val3
QUEUED
> EXEC
1) OK
2) OK
```

コマンド	説明
MULTI	トランザクションを開始
EXEC	トランザクション内の全コマンドを実行
DISCARD	トランザクション内の全コマンドを破棄
WATCH キー …	指定したキーを監視し、変更されたらEXECを失敗にする
UNWATCH	キーの監視を解除

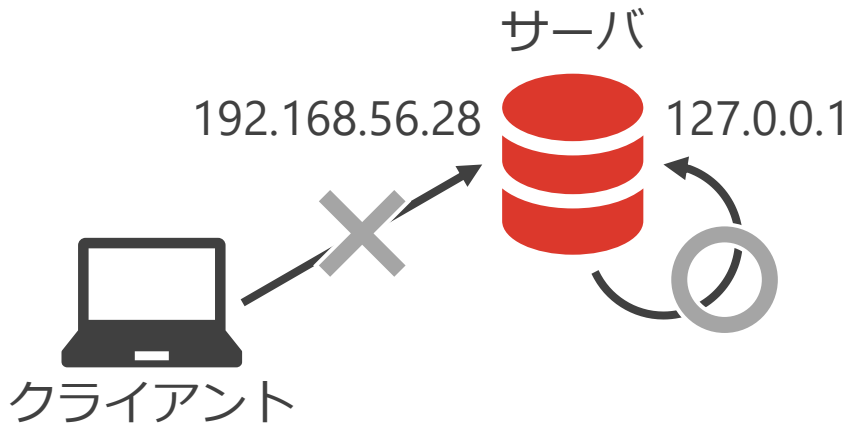
リモートホストからRedisへの接続設定

- デフォルトではローカルホストからのみ接続可
- リモートホストからの接続には設定が必要

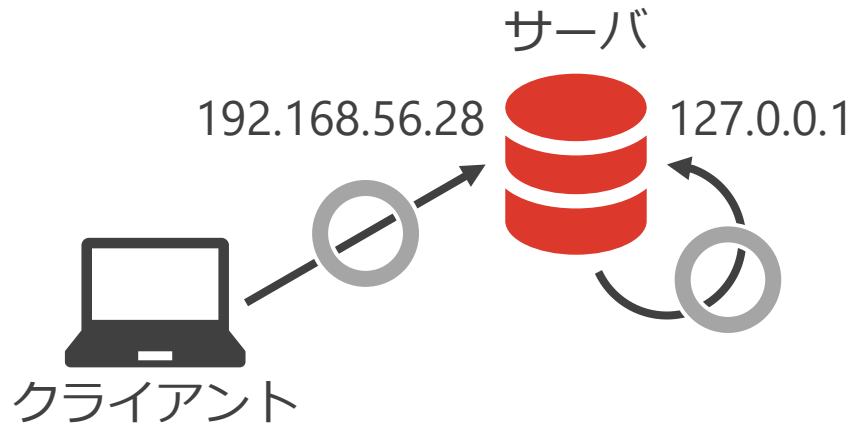
/etc/redis.conf

```
bind 127.0.0.1 192.168.56.28      # 接続受付IPアドレス
requirepass P@$5w0rd              # パスワード
```

bind 127.0.0.1



bind 127.0.0.1 192.168.56.28

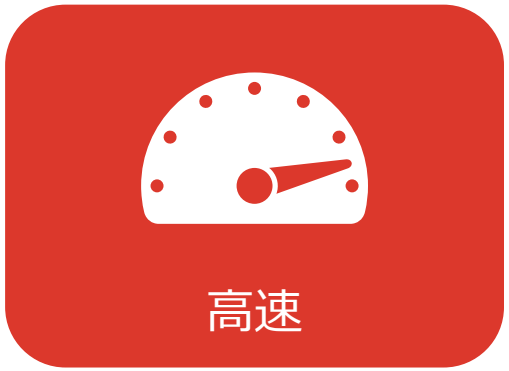


RedisとPostgreSQLの連携

Accessing from PostgreSQL to Redis

RedisとPostgreSQLの連携

Redis (KVS)

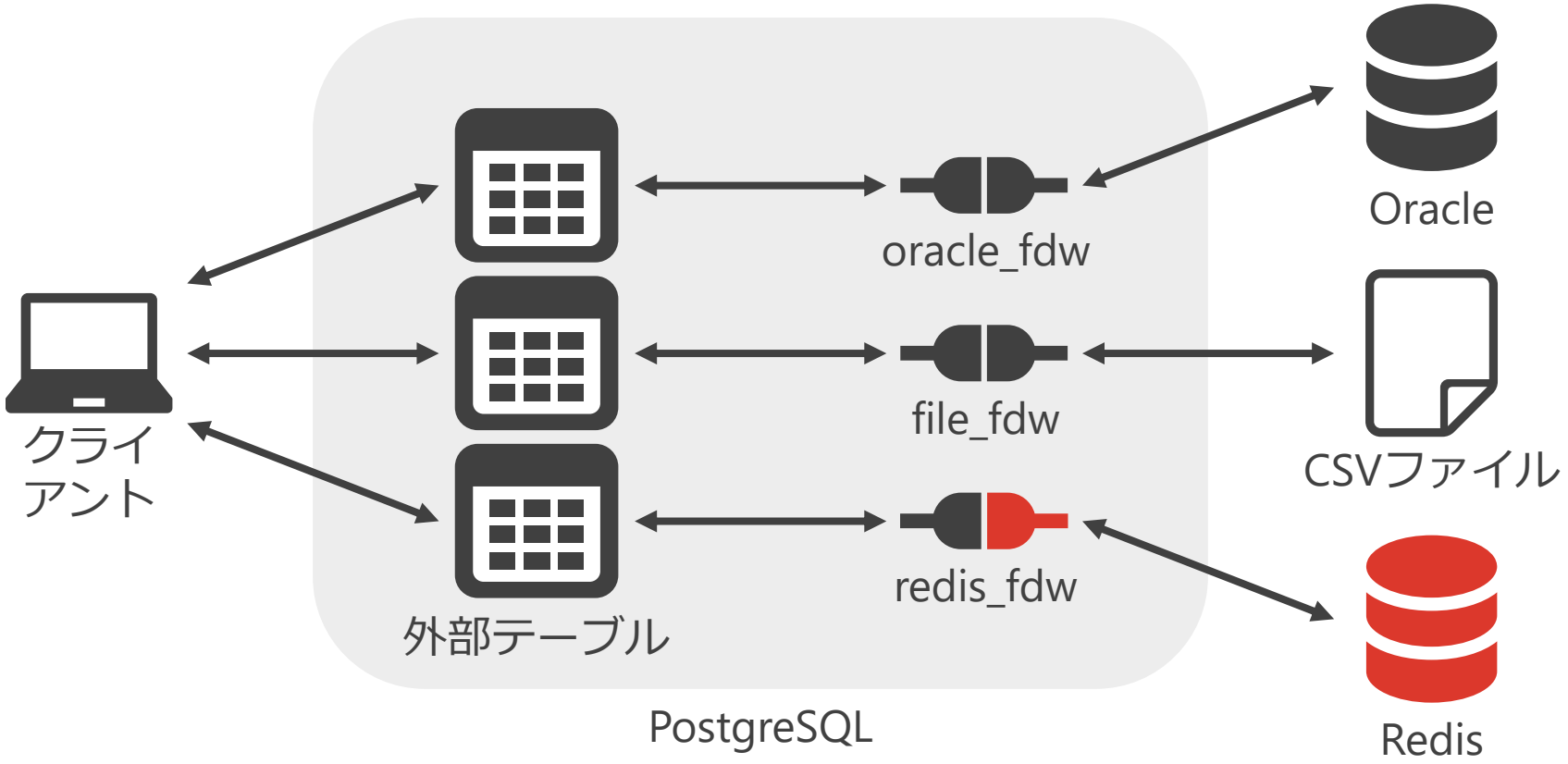


互いの弱点を
補い合う

PostgreSQL (RDB)

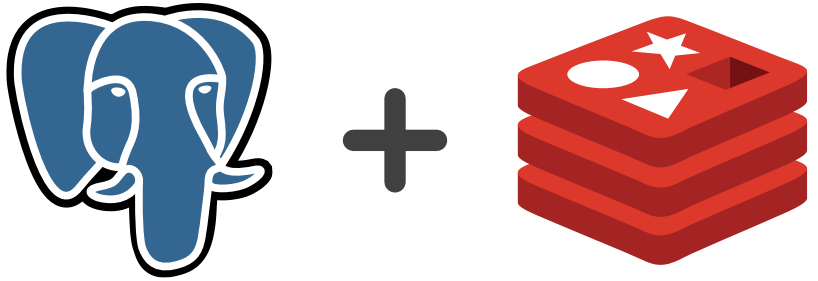
外部データラッパ (Foreign Data Wrapper, FDW) とは

- データベースから外部データにアクセスする機能
- SQL標準のSQL/MEDに基づく



redis_fdwとは

- Redis用FDW
- もとはDave Page氏が開発
- 現在はAndrew Dunstan氏が開発を継続
- リポジトリはGitHubで公開
- 記述言語はC
- ライセンスはPostgreSQLライセンス



PostgreSQLのインストール

- PostgreSQLがまだならインストール
- RHEL8の場合

```
$ sudo dnf -qy install https://download.postgresql.org/pub/repos/yum/reporpms/EL-$(rpm -E %rhel-%_arch)/pgdg-redhat-repo-latest.noarch.rpm
                                                    # PGDGリポジトリをインストール
$ sudo dnf -qy module disable postgresql            # モジュールを無効化
$ sudo dnf -qy install postgresql15-server          # パッケージをインストール
$ PGSETUP_INITDB_OPTIONS="-E UTF8 --no-locale" ¥    # データベースクラスタを作成
  sudo -E /usr/pgsql-15/bin/postgresql-15-setup initdb
Initializing database ... OK

$ sudo systemctl enable postgresql-15.service       # サーバの自動起動を有効化
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql-15.service → /usr/lib/systemd/system/postgresql-15.service.
$ sudo systemctl start postgresql-15.service        # サーバを起動
```

redis_fdwのインストール

- バイナリ提供はない
- ソースコードからビルドが必要

```
$ sudo dnf -qy install epel-release  # EPELリポジトリをインストール
$ sudo dnf -qy install make redhat-rpm-config postgresql15-devel hiredis-devel
                                     # ビルドに必要なパッケージをインストール
$ curl -LOs https://github.com/pg-redis-fdw/redis_fdw/archive/refs/heads/REL_15
_STABLE.tar.gz                       # ソースコードをダウンロード
$ tar -xzf REL_15_STABLE.tar.gz      # ソースコードを展開
$ cd redis_fdw-REL_15_STABLE         # ソースコードの展開先に移動
$ make PG_CONFIG=/usr/pgsql-15/bin/pg_config
                                     # ソースコードをビルド
$ sudo make install PG_CONFIG=/usr/pgsql-15/bin/pg_config
                                     # プログラムをインストール
```

外部テーブルを通じたRedisへのアクセス (1/4)

- データベースを作成し、拡張をインストール

```
$ createdb redis_db
$ psql redis_db
=# CREATE EXTENSION redis_fdw;
CREATE EXTENSION
```

- 外部サーバを作成

```
=# CREATE SERVER redis_server
     FOREIGN DATA WRAPPER redis_fdw OPTIONS (address '127.0.0.1', port '6379');
CREATE SERVER
```

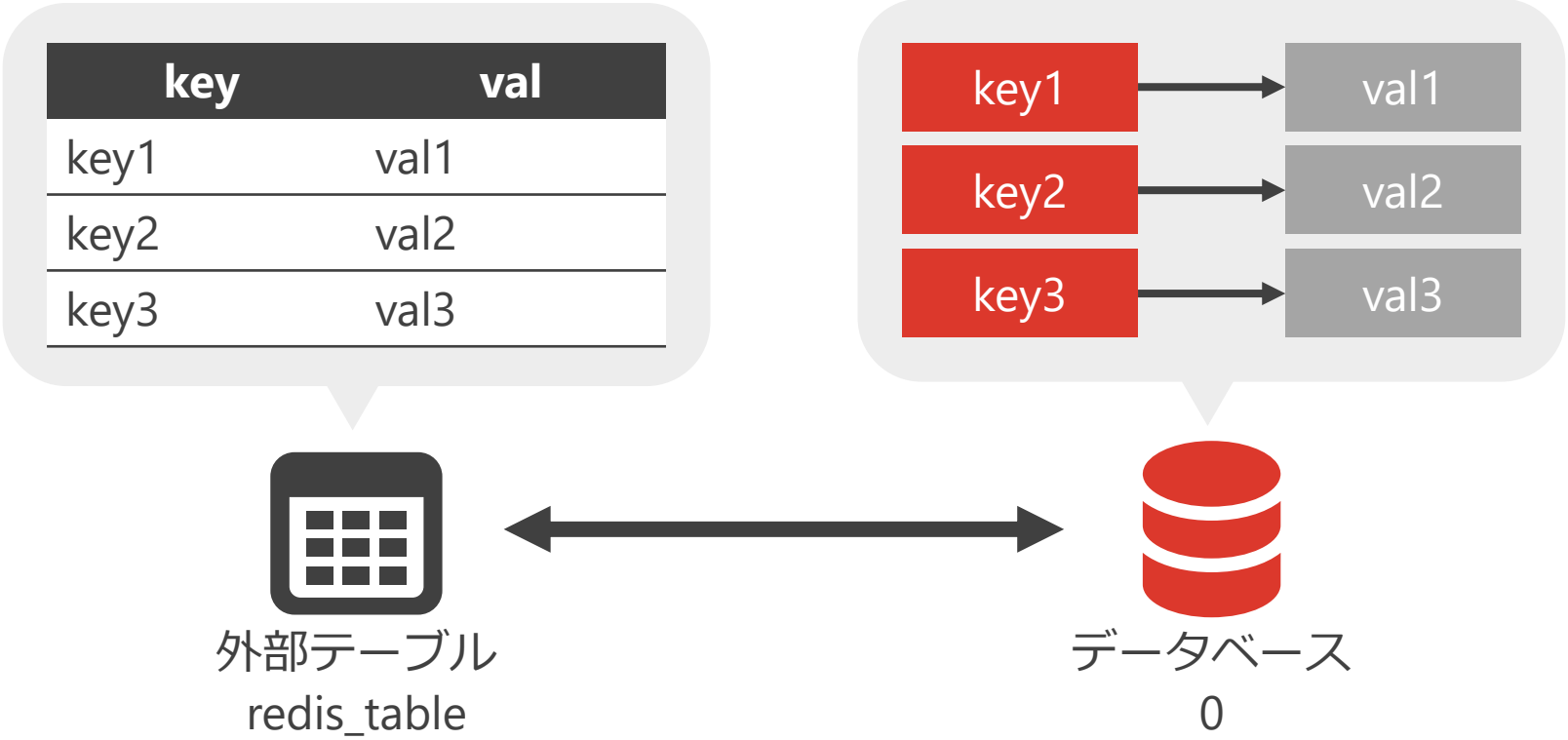
- 外部サーバのユーザマッピングを作成

```
=# CREATE USER MAPPING FOR PUBLIC
     SERVER redis_server OPTIONS (password 'P@s5w0rd');
CREATE USER MAPPING
```

外部テーブルを通じたRedisへのアクセス (2/4)

- 外部テーブルを作成

```
=# CREATE FOREIGN TABLE redis_table (key text, val text)
     SERVER redis_server OPTIONS (database '0');
CREATE FOREIGN TABLE
```



外部テーブルを通じたRedisへのアクセス (3/4)

- 外部テーブルのデータを参照し、挿入

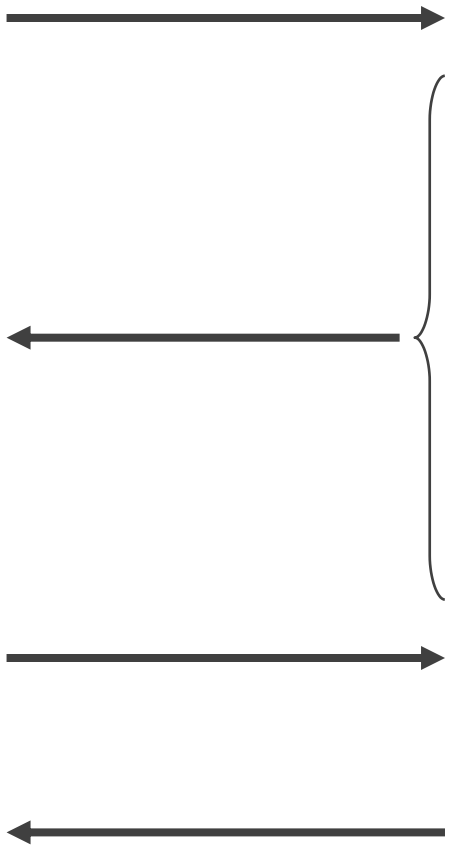
PostgreSQL

```
=# SELECT * FROM redis_table;


 key  | val
------+------
 key1 | val1
 key2 | val2
 key3 | val3
(3 行)


=# INSERT INTO redis_table
     VALUES ('key4', 'val4');


INSERT 0 1
```

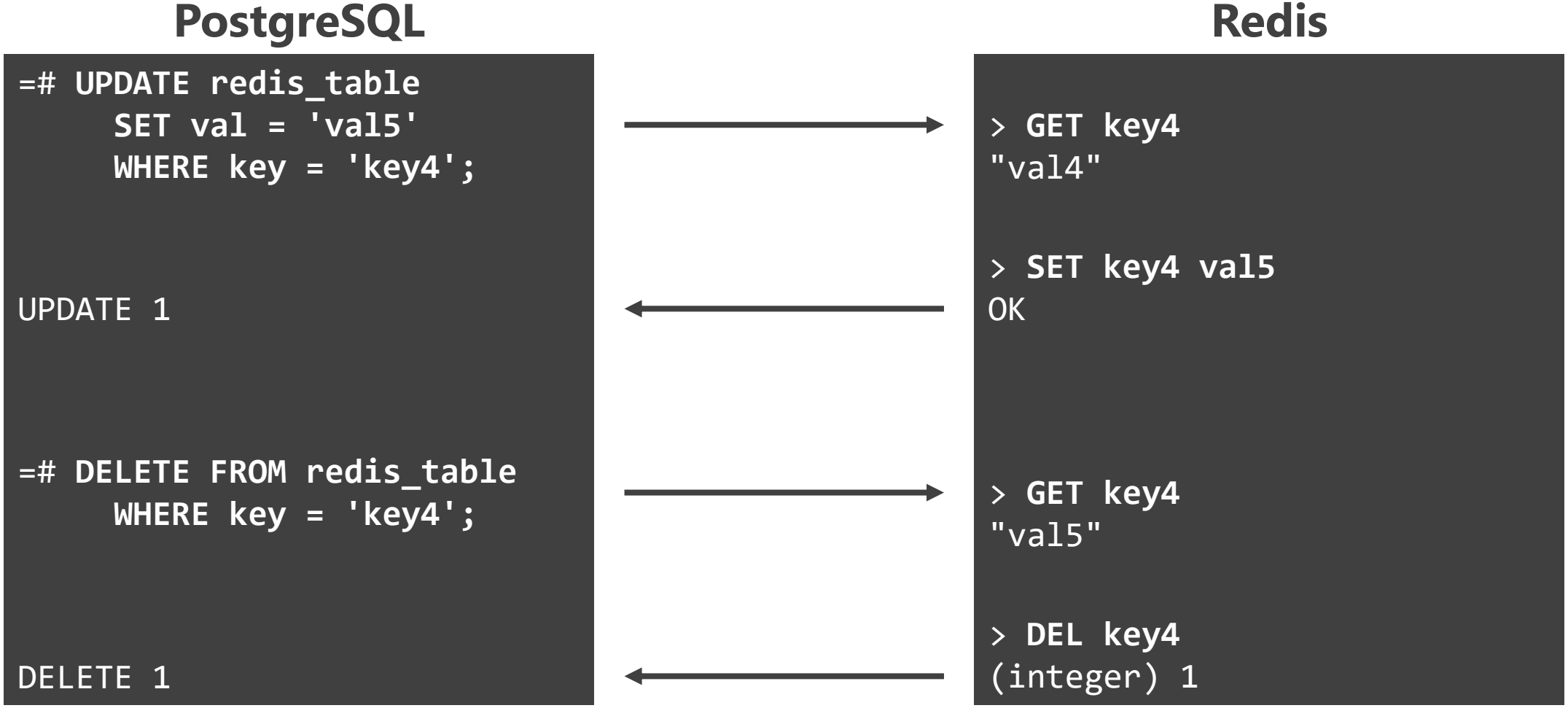


Redis

```
> SCAN 0 COUNT 1000
1) "0"
2) 1) "key1"
   2) "key2"
   3) "key3"
> GET key1
"val1"
> GET key2
"val2"
> GET key3
"val3"
> EXISTS key4
(integer) 0
> SET key4 val4
OK
```

外部テーブルを通じたRedisへのアクセス (4/4)

- 外部テーブルのデータを更新し、削除



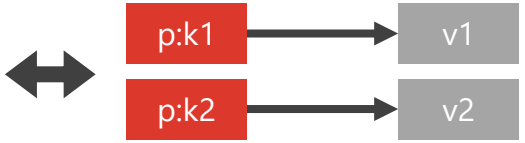
redis_fdw外部テーブルのオプション

オプション	説明
database	データベース番号。0～15を指定可。デフォルトは0
tabletype	データ型。list、set、hash、zset (ソート済みセット) を指定可。デフォルトは未指定で文字列
tablekeyprefix	指定した接頭辞をもつキーでデータを絞り込む
tablekeyset	指定したキーのセットに含まれるキーでデータを絞り込む
singleton_key	指定したキーの値を展開し、データとする

※tablekeyprefix、tablekeyset、singleton_keyはいずれか1つのみ指定可

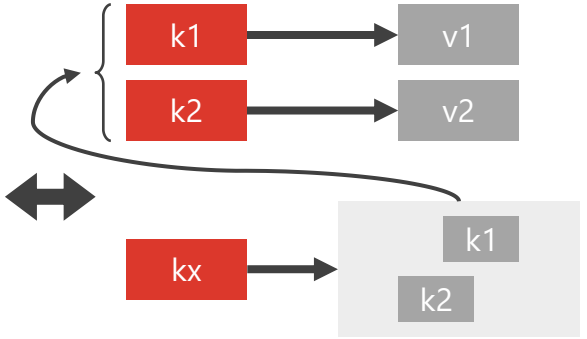
tablekeyprefix 'p:'

key	val
p:k1	v1
p:k2	v2



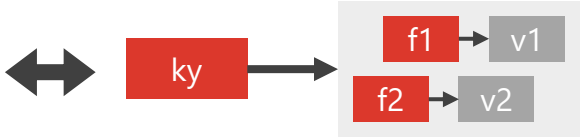
tablekeyset 'kx'

key	val
k1	v1
k2	v2



tabletype 'hash', singleton_key 'ky'

key	val
f1	v2
f2	v2



まとめ

- Redisはインメモリで高速なKVS
- データはRDBやAOFで永続化
- 豊富なデータ型に対応
- 自動フェイルオーバやシャーディング機能を備える
- redis_fdwでPostgreSQLからRedisにアクセス可
- KVSの高速さを活かしつつ、SQLで複雑な問い合わせも可

SRA OSSでは、PostgreSQLはもちろん、Redisやredis_fdwのサポートを提供
Redisに関して何かありましたら、お気軽にお問い合わせください

参考URL

- Redis
  - https://redis.io/
- redis_fdw
  - https://github.com/pg-redis-fdw/redis_fdw
- SRA OSS
  - Tech Blog - Redis: Redisを含むOSSの技術情報を公開
    - https://www.sraoss.co.jp/tech-blog/category/redis/
  - OSSサポート: Redisもサポート対象
    - https://www.sraoss.co.jp/prod_serv/support/oss-support/
  - PostgreSQLサポート&保守: redis_fdwもオプションでサポート可
    - https://www.sraoss.co.jp/prod_serv/support/pgsql-mainte/

sraoss redis

SRA OSS

https://www.sraoss.co.jp/ sales@sraoss.co.jp 03-5979-2701