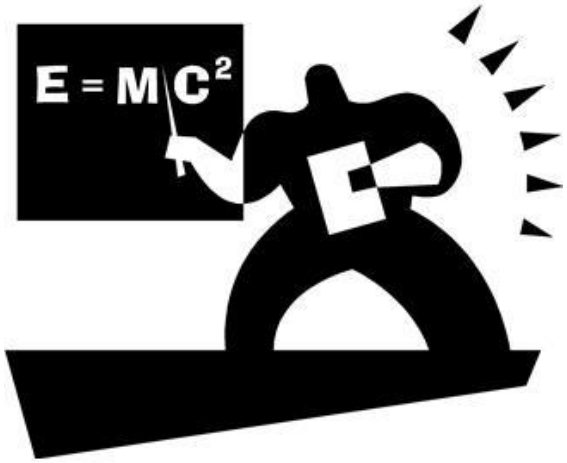
PostgreSQL高可用性構成の選択肢とトレンド

PostgreSQL高可用性構成最新動向セミナー

2013年2月12日　14:10～15:00

SRA OSS, Inc. 日本支社 PostgreSQL技術グループ 高塚 遥

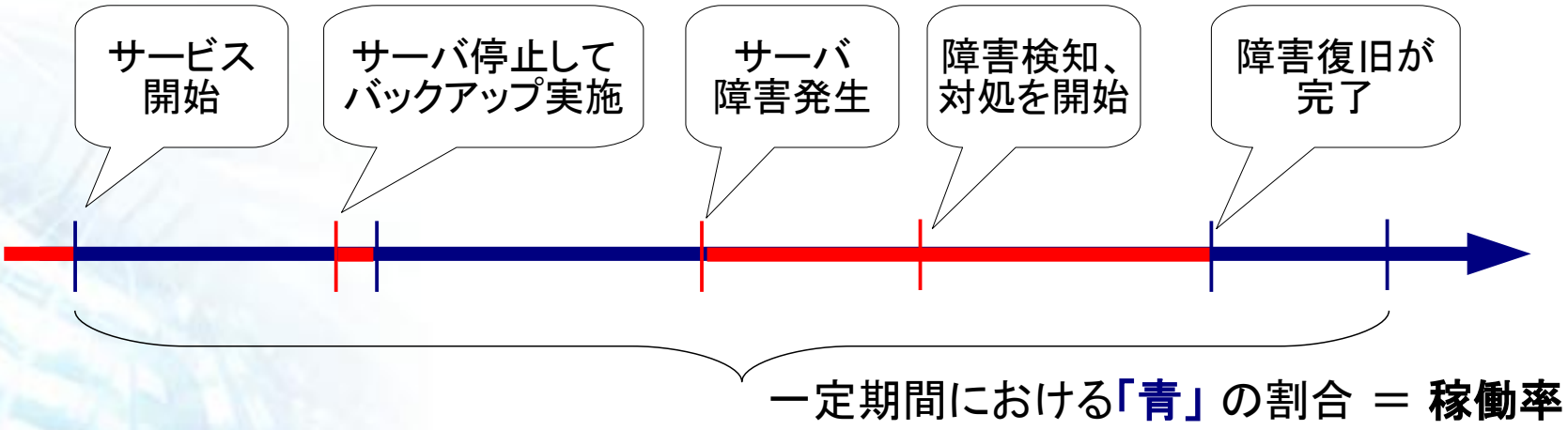
高可用性の基本概念



可用性、信頼性の考え方について、
基本概念を説明いたします。

高可用性構成とは？（１）

- 高可用性とは？
  - サービスが提供できない時間の割合が小さいこと
  - 「絶対に止まらないよう」にはできない



高可用性構成とは？（２）

- 「とにかく絶対に止まらないようにしてほしい」
  「どのくらいの稼働率を想定されていますか？」

※「ご予算は…」
という聞き方もあるが
本来的ではない

稼働率	年間停止時間
90%	36.5 日
99%	3.65 日
99.9%	8.7 時間
99.99%	52 分
99.999%	5 分

高可用性構成とは？（３）

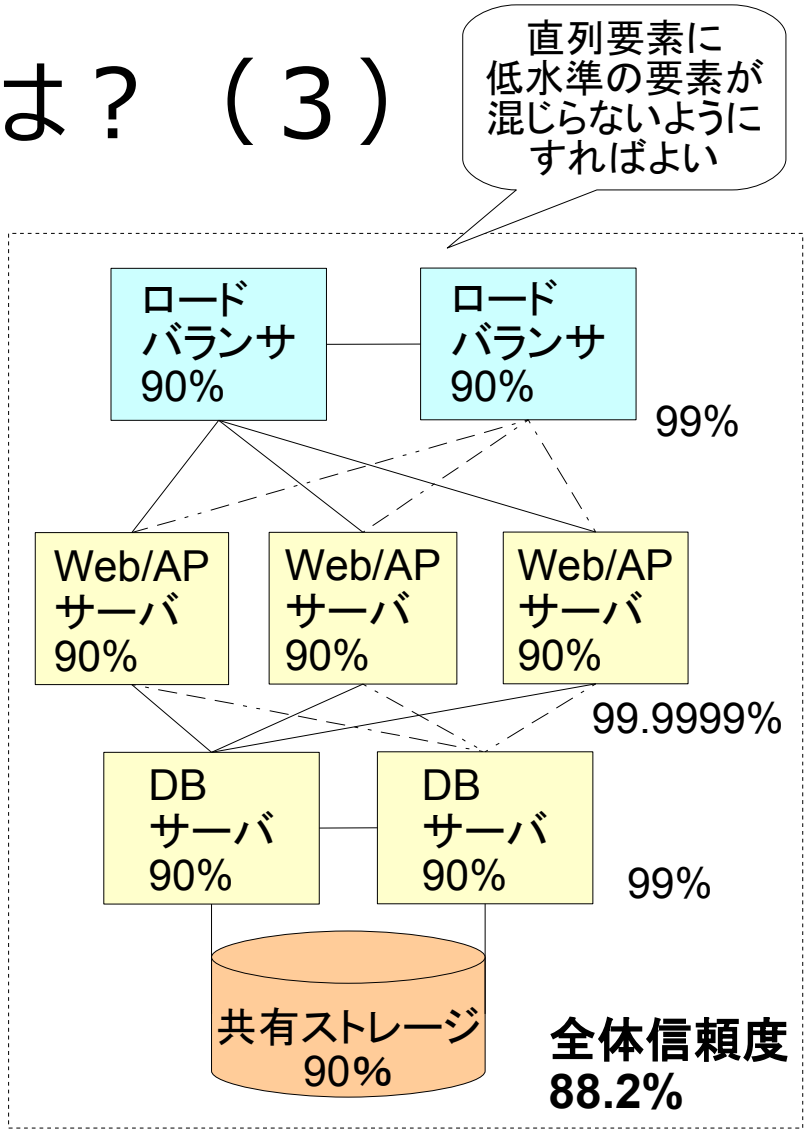
- 信頼度の計算
  - MTBF（平均故障間隔）

    カタログ記載～メーカ表明
  - 故障率 λ ＝ １ / MTBF

    単位期間での故障数の期待値
  - 信頼度 R=exp(-λ)

    単位期間に障害が起きない確率

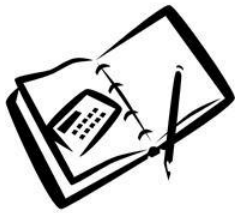
    MTBF 8～9万時間で 90%
  - 組み合わせた信頼度 →



高可用性構成とは？（４）

経年劣化には、別途の考慮がいることに注意

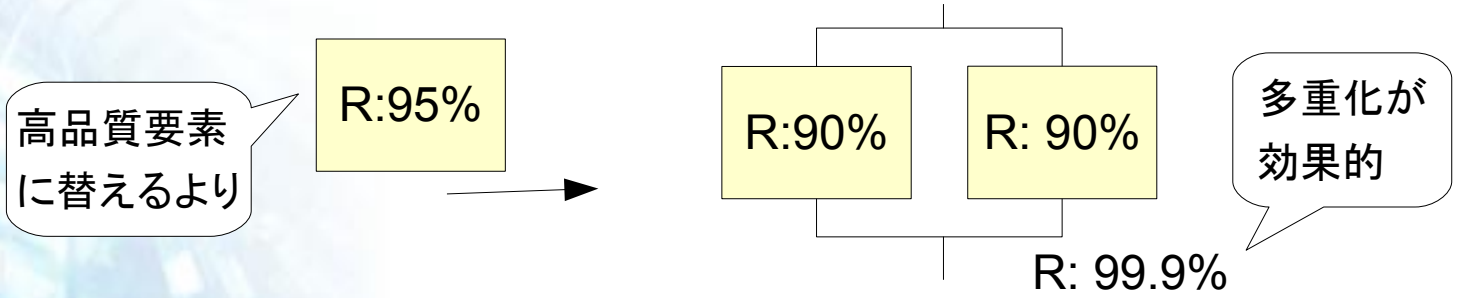
- 稼働率の見積もり
  - 年間信頼度 88.2%（＝故障率 0.13／年）のシステム
  - 見積もりした MTTR（平均復旧時間） 20 時間
    - 平均検知時間 15時間
      - 平日日中のみなら 30分で検知、さもなくば翌平日 9:00検知
    - 平均リカバリ対応 5時間
      - マシン予備機はあるものと想定して見積もり
  - 年間停止 0.13 x 20 = 2.6 時間
  - 実現できる稼働率 99.9% くらい



※障害が1回起きても停止はこの程度、という観点が求められることも

高可用性構成とは？（５）

- 高可用性の実現方法
  - 各要素において障害を起きにくくする
    - ハウジング環境選定・ハードウェア選定は品質優先
    - OS、ミドルウェアは安定バージョンを使う
    - 自社ソフトは十分な品質工程をふむ
    - 基本ではあるが、一定水準を超えて上げていくのは難しい



高可用性構成とは？（６）

- 高可用性の実現方法
  - メンテナンス停止時間を短くする
    - 停止しない、オンラインバックアップを使う
      - pg_dump (pg_dumpall)、PITR によるバックアップ
    - VACUUM FULL 運用をやめる
    - 各種サーバ保守作業の間に代わりに動作するサーバを用意

ここは特に PostgreSQL の場合

  - 障害を検知し対処開始するまでの時間を短くする
  - 障害から復旧するまでの時間を短くする

高可用性構成とは？（７）

- 高可用性の実現方法
  - メンテナンス停止時間を短くする
  - 障害を検知し対処開始するまでの時間を短くする
    - 自動で動作を監視するソフトウェアを導入
    - 障害検知で担当者がすぐアクションできる体制を作る
    - 検知から復旧動作を自動開始させるソフトウェア等を導入
  - 障害から復旧するまでの時間を短くする

高可用性構成とは？（８）

- 高可用性の実現方法
  - メンテナンス停止時間を短くする
  - 障害を検知するまでの時間を短くする
  - 障害から復旧するまでの時間を短くする
    - バックアップからのリストア所要時間を短縮する
      - 予備のマシンを用意しておく
    - 直ちに切り替え可能な待機サーバを用意する
    - 障害検知を受けて自動「切り替わり」「切り離し」させるHAクラスタソフトウェア等を導入し、多重化を実現

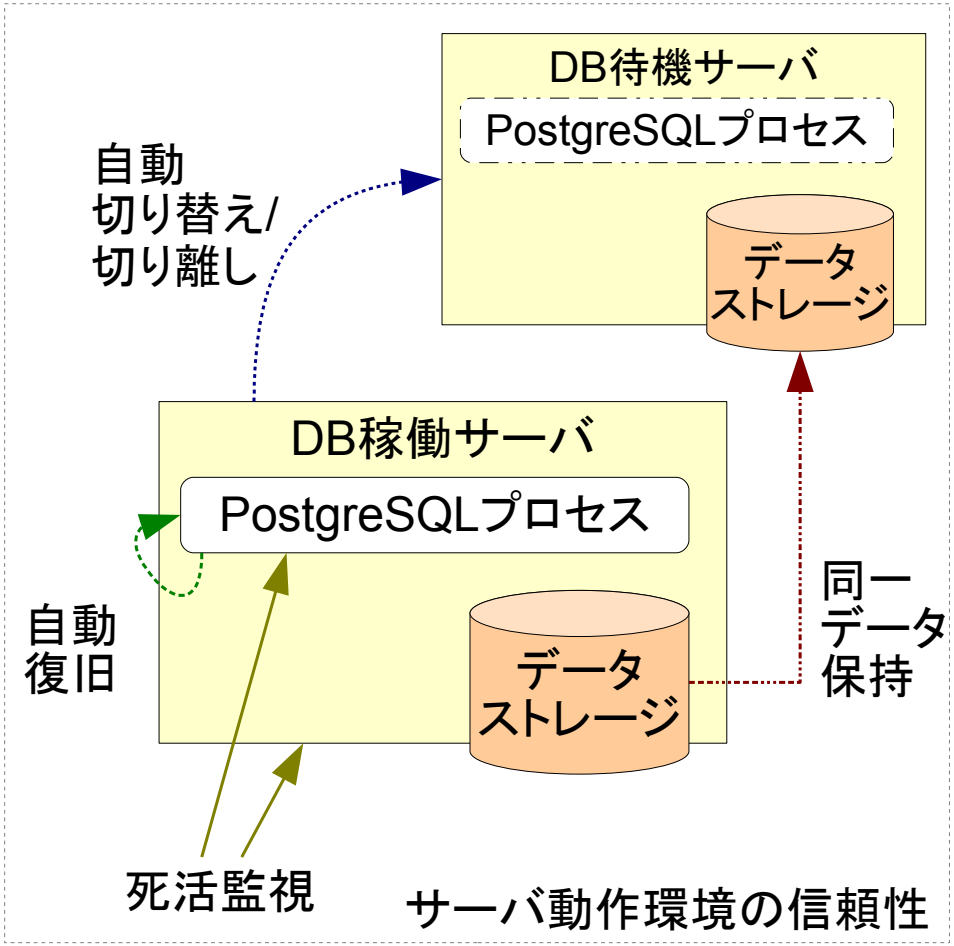
高可用性構成とは？（９）

- PostgreSQL高可用性の実現方法めやす

稼働率	年間停止時間	実現方法
90%	36.5 日	バックアップ～リストアだけで十分。オンラインバックアップ取得を実施。
99%	3.65 日	オンプレミスなら予備マシンが必要。大データならバックアップのリストア所要時間を把握しておく。
99.9%	8.7 時間	保守停電のないクラウド～ハウジングが必要。平日日中のみ障害検知だとむずかしい。
99.99%	52 分	バックアップのリストアがほぼ不可能。レプリケーション（データ同期）された待機サーバが必要。
99.999%	5 分	HAクラスタソフトウェア等が必要。5分は自動切り替え１～２回分。

高可用性構成の要素

- 高可用性構成の要素
  - 機器の信頼性
    - 動作環境の信頼性
  - 多重化の仕組み
    - 死活監視
    - 自動復旧
    - 自動切り替え/切り離し
      - 同一データ保持
    - 仕組み自体の信頼性

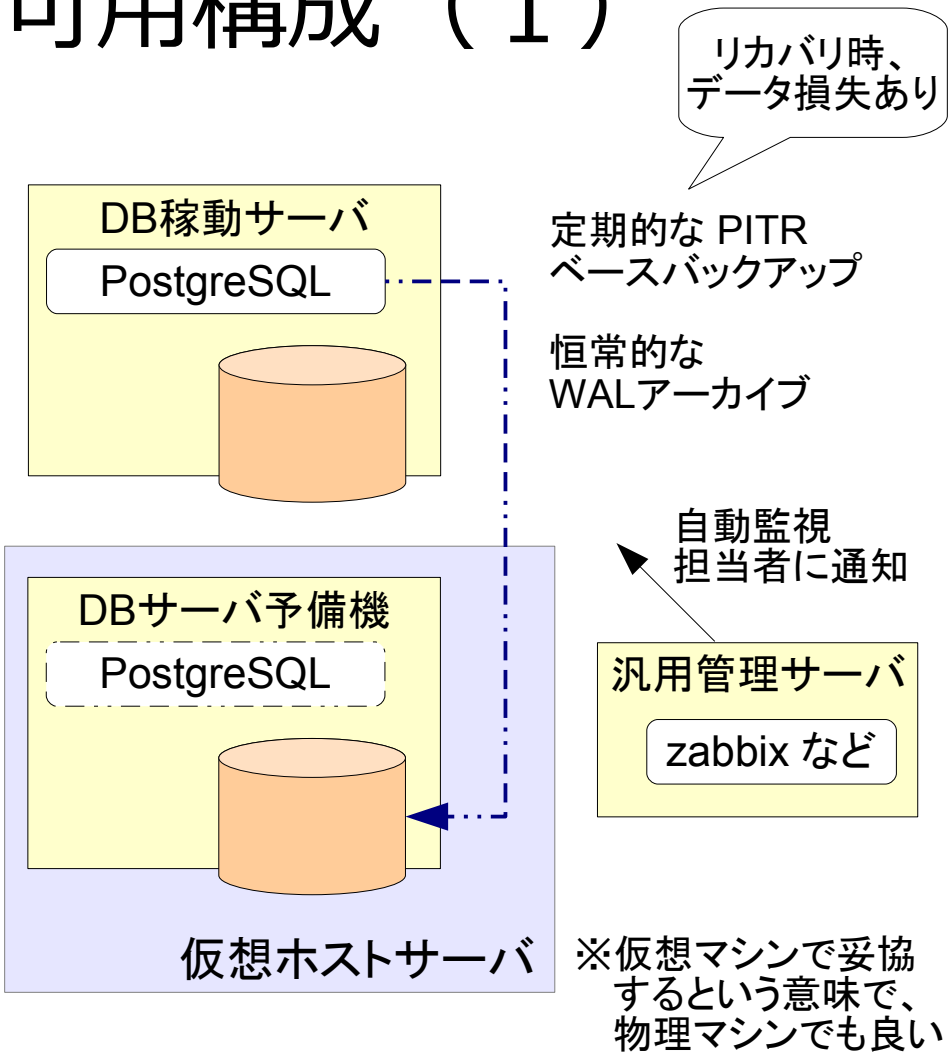


PostgreSQLむけ
実際の
高可用構成

具体的な構成例を紹介します。
全て何らかの形で実例があるものです。

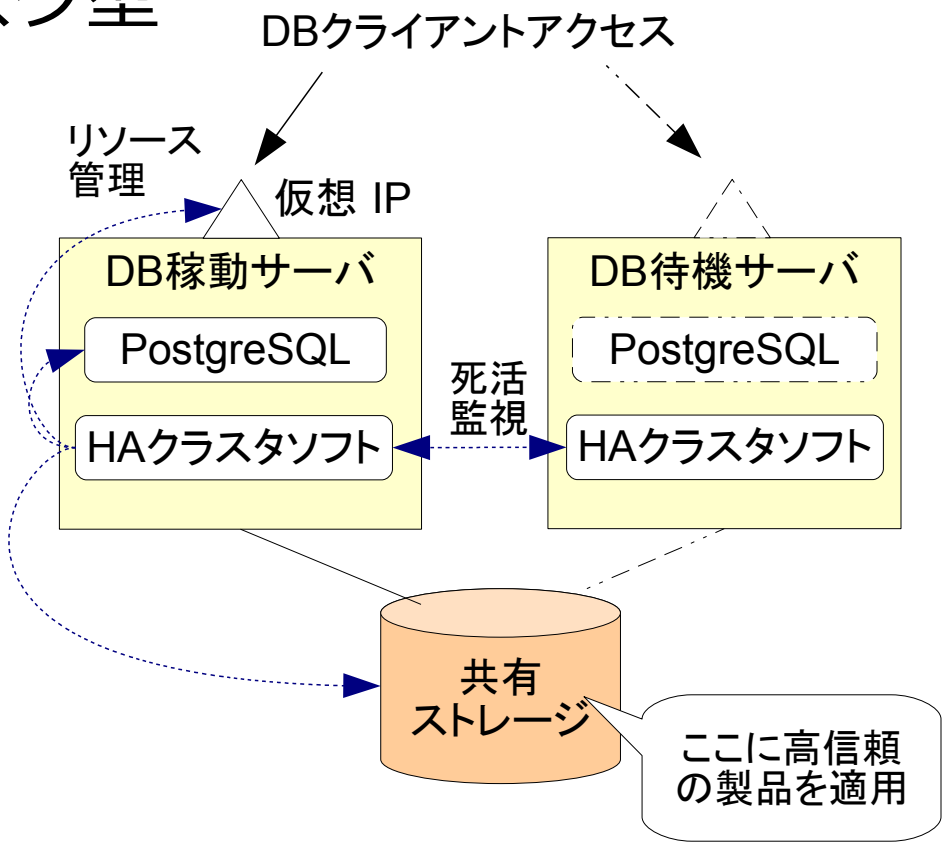
PostgreSQL高可用構成（１）

- 予備機とバックアップ
  - 稼働率 95～99%
  - バックアップ先でデータベース稼働できる体制を作っておけばよい
  - 検知～対応開始の時間に合わせて担当者体制を
  - リストア所要時間にあわせてベースバックアップ頻度を調整



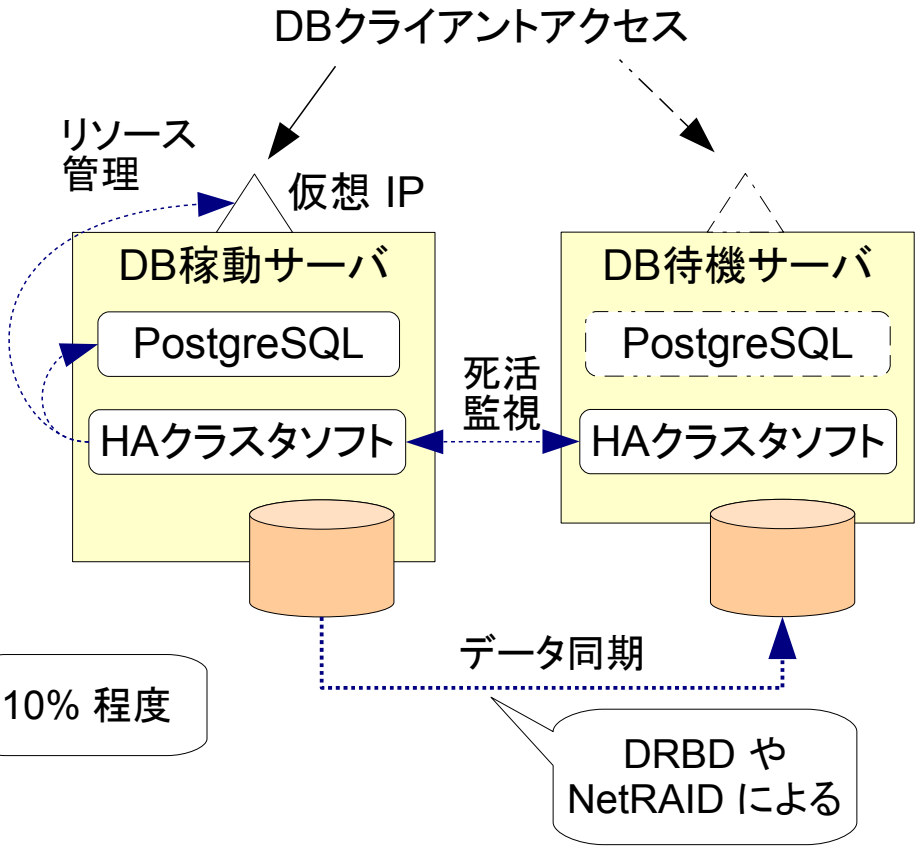
PostgreSQL高可用構成（２）

- HAクラスタ 共有ディスク型
  - 稼動率 ～99.999%
  - 共有ストレージが弱点
  - 待機サーバは待機だけ
  - HAクラスタソフト:
    - LifeKeeper
    - CLUSTERPRO
    - Pacemaker



PostgreSQL高可用構成（３）

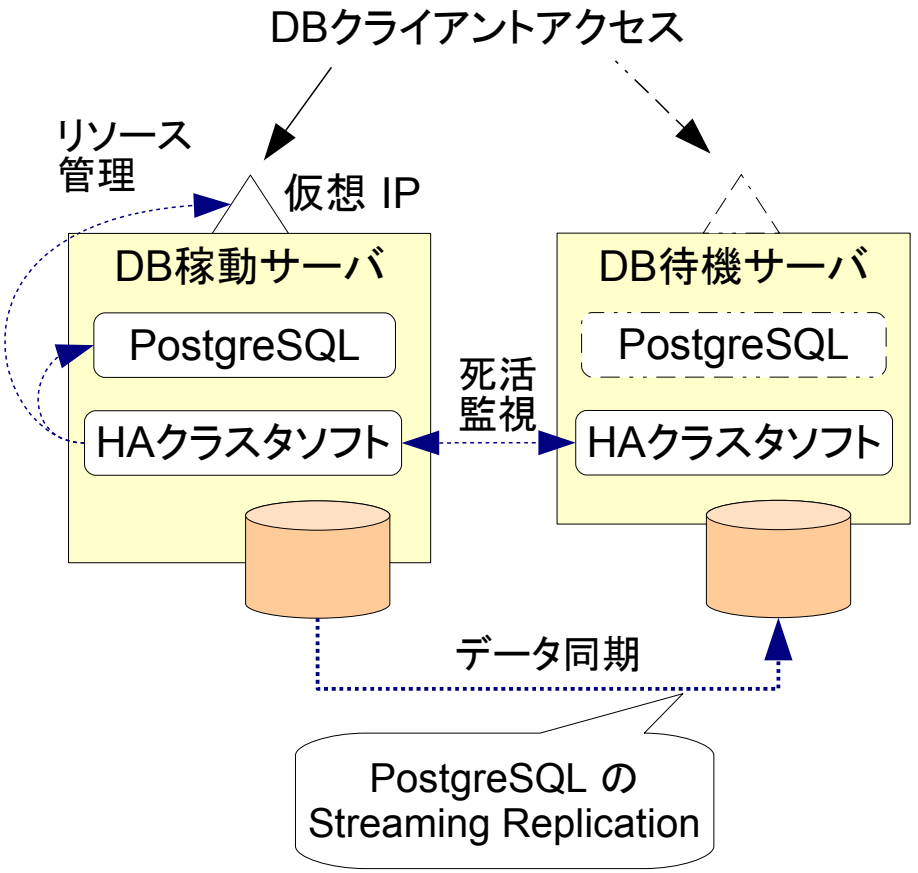
- HAクラスタ データレプリケーション型
  - ブロックデバイス～ファイルシステムレベルでのデータ同期機能を使う
  - HAクラスタソフトに機能統合されている
  - 書き込み性能劣化あり



PostgreSQL高可用構成（４）

- HAクラスタ SR型
  - PostgreSQL のレプリケーション機能をデータ同期に使う
    - Linux-HA Japan Pacemakerリポジトリ 1.0.12-1.2 以上 (2012年7月～)
    - PostgreSQL 9.1 以上
  - 書き込み性能劣化あり
  - 運用性やや難

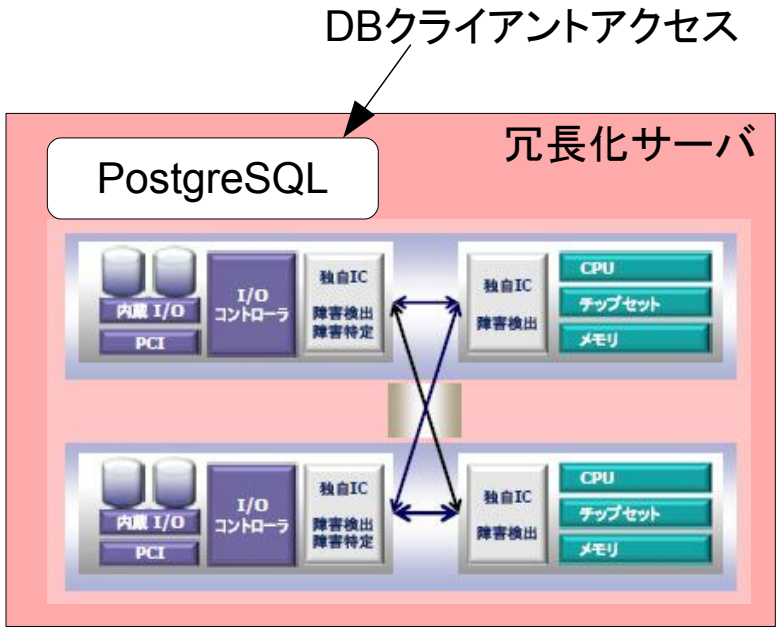
(3)よりは少し軽い



PostgreSQL高可用構成（５）

- 無停止型サーバ
  - 稼働率 99.999%以上も
  - ハードウェアにて冗長化されたサーバマシン
  - HAソフトウェアのセットアップが不要
  - アプリ障害には弱い
    - 単体用HAソフトを
  - Stratus ftServer、NEC Express5800/ft など

CLUSTERPRO SSS monit などの プロセス監視ツール

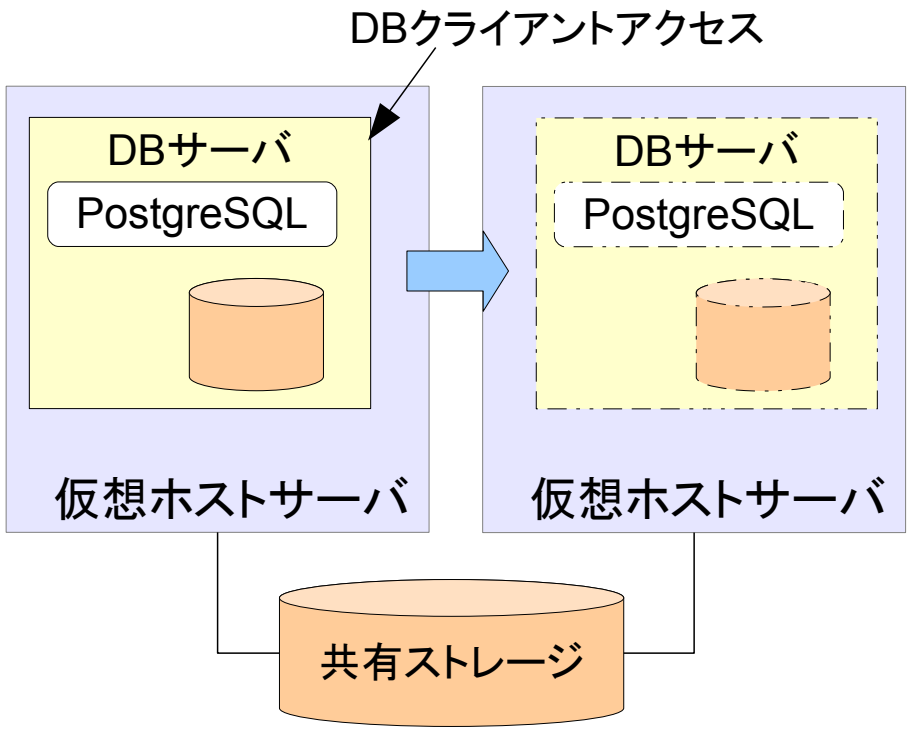


※図はftServer 資料より

PostgreSQL高可用構成（６）

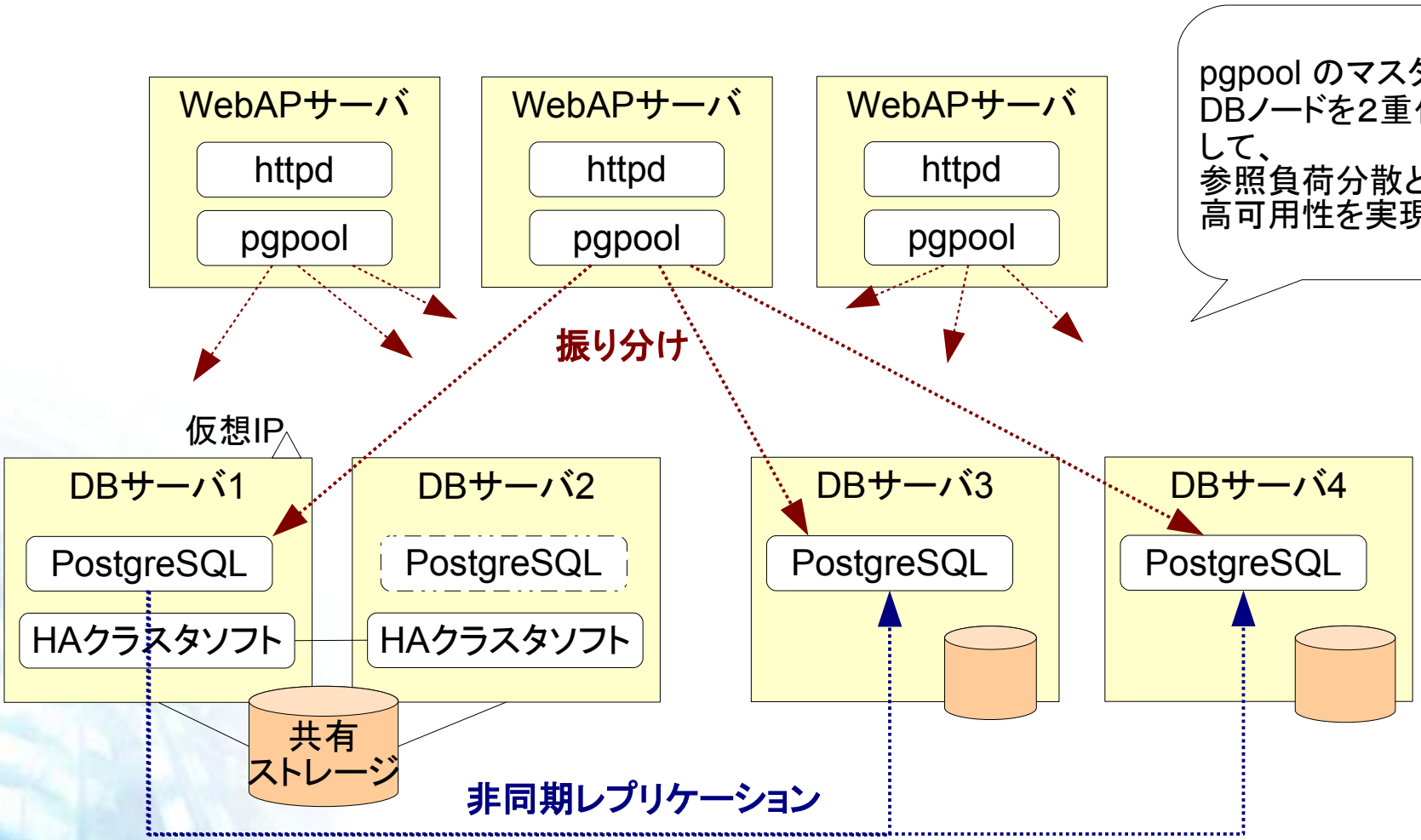
- 仮想基盤の高可用機能
  - VMware HA など
    - 仮想ホスト上で仮想マシンごとフェイルオーバー
    - OSレベル障害、マシンレベル障害に対応
    - アプリ障害には弱い
      - 単体用HAソフトを

CLUSTERPRO SSS
monit などの
プロセス監視ツール

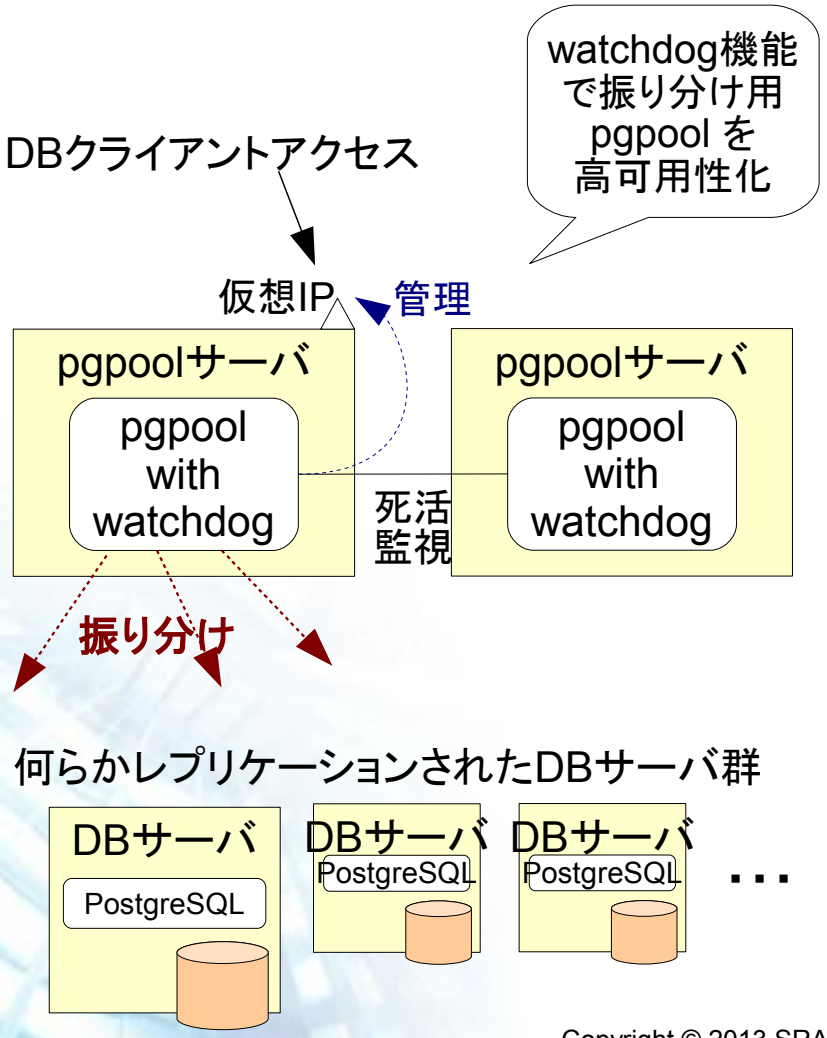


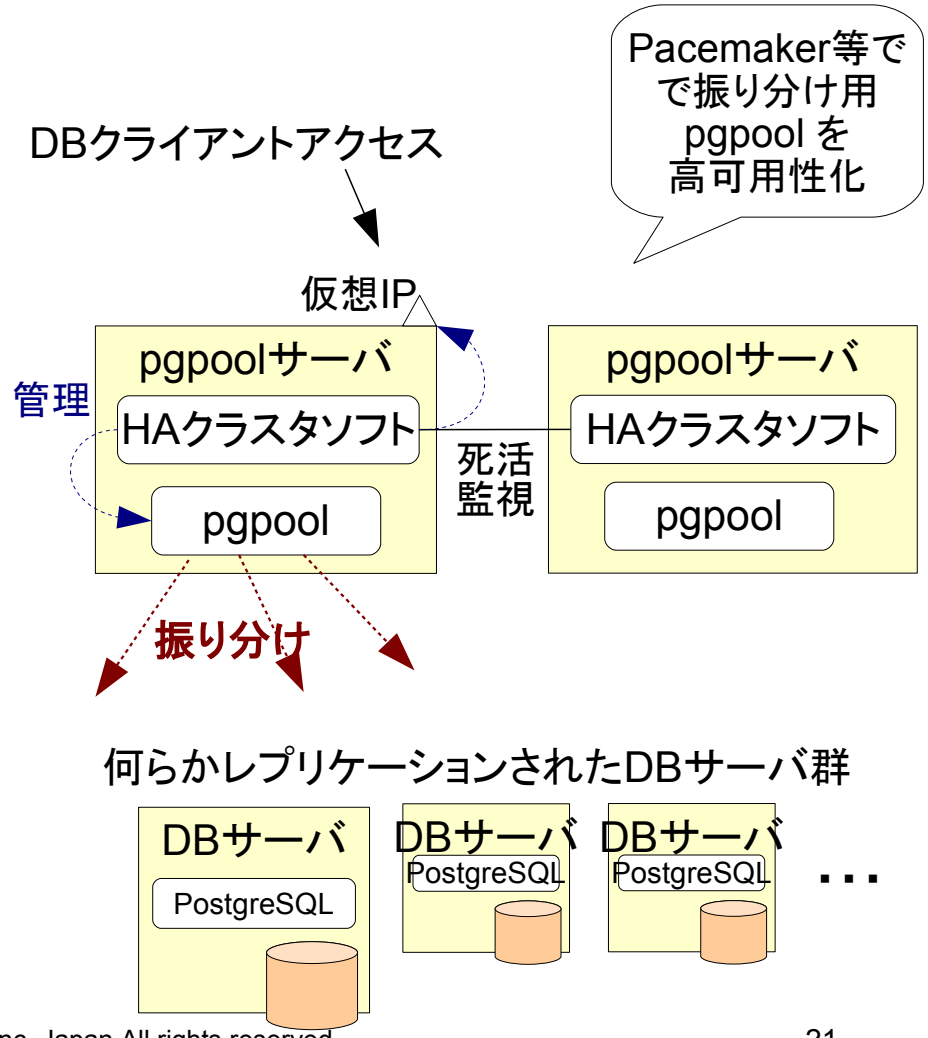
※データレプリケーション型もある

負荷分散クラスタとの統合（１）



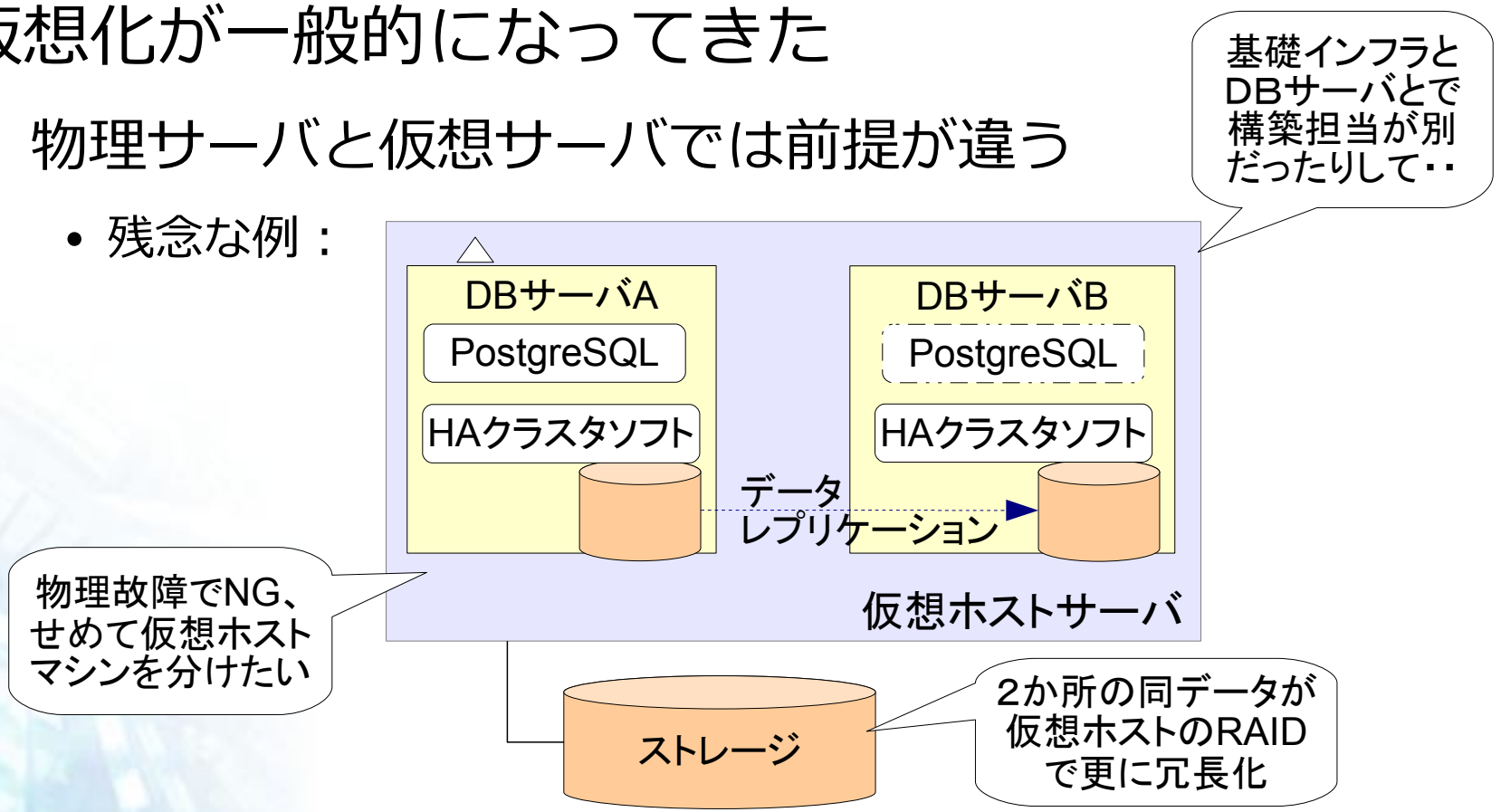
負荷分散クラスタとの統合（２）





近年の傾向（１）

- 仮想化が一般的になってきた
  - 物理サーバと仮想サーバでは前提が違う
    - 残念な例：



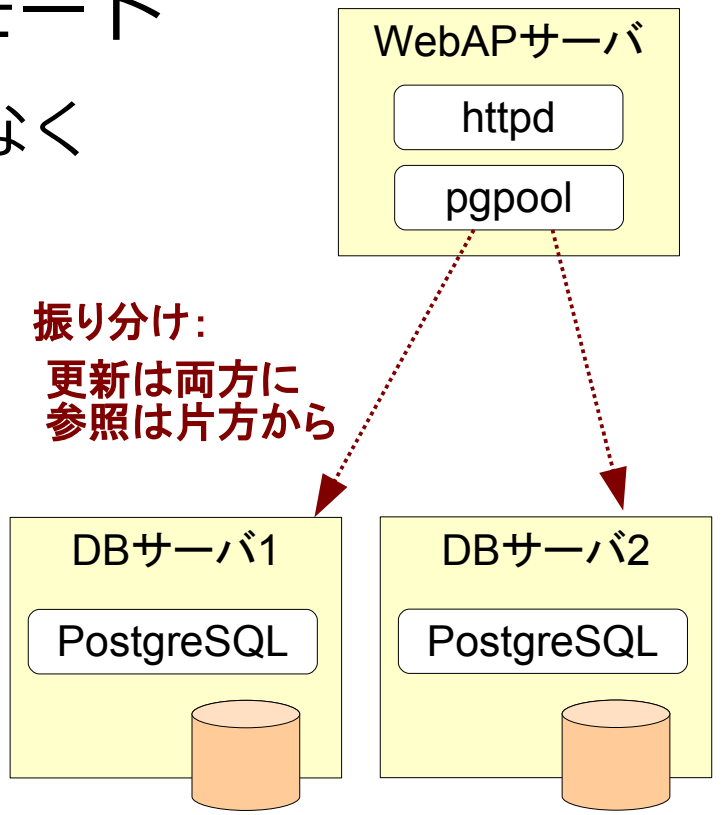
近年の傾向（２）

- クラウド上でどう可用性を実現するの？
  - vmware HA や 無停止型サーバと同じ考え方で対応
    - マシン、OSレベルの障害対策についてクラウド業者が表明しているスペックを参照して選択
      - 必要に応じて可用性を付加するサービスメニューを使う
    - ミドルウェア、アプリケーションレベルの障害対策につては単体保護の仕組みを導入
    - ホスティング業者の中には物理サーバ２台組でクラスタを組ませてくれるところもある

近年の傾向（３）

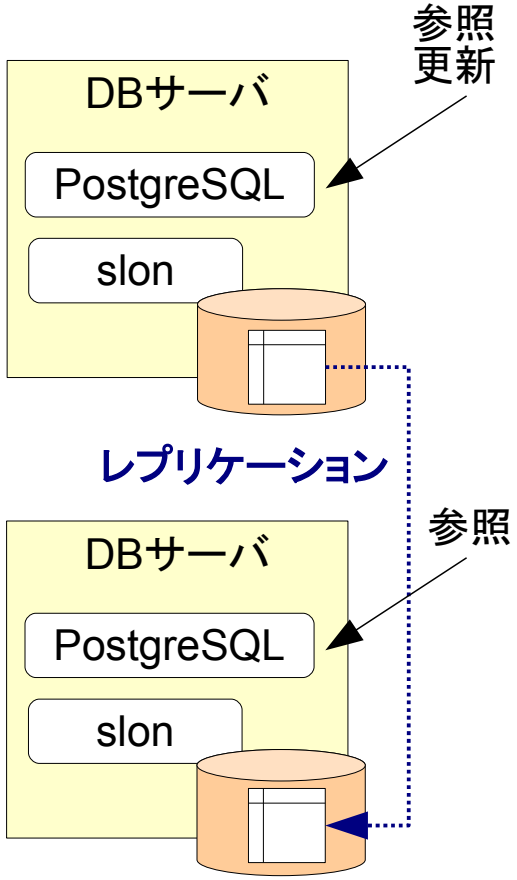
- pgpool-II レプリケーションモード
  - 可用性目的ではあまり採用されなくなってきている
    - 何らか pgpool自体の保護が必要
    - 同期型レプリケーション
    - フェイルオーバーは「切り離し」なので速い
    - 同時実行トランザクション設計にそれなりに神経を使う

二相コミットではない



近年の傾向（４）

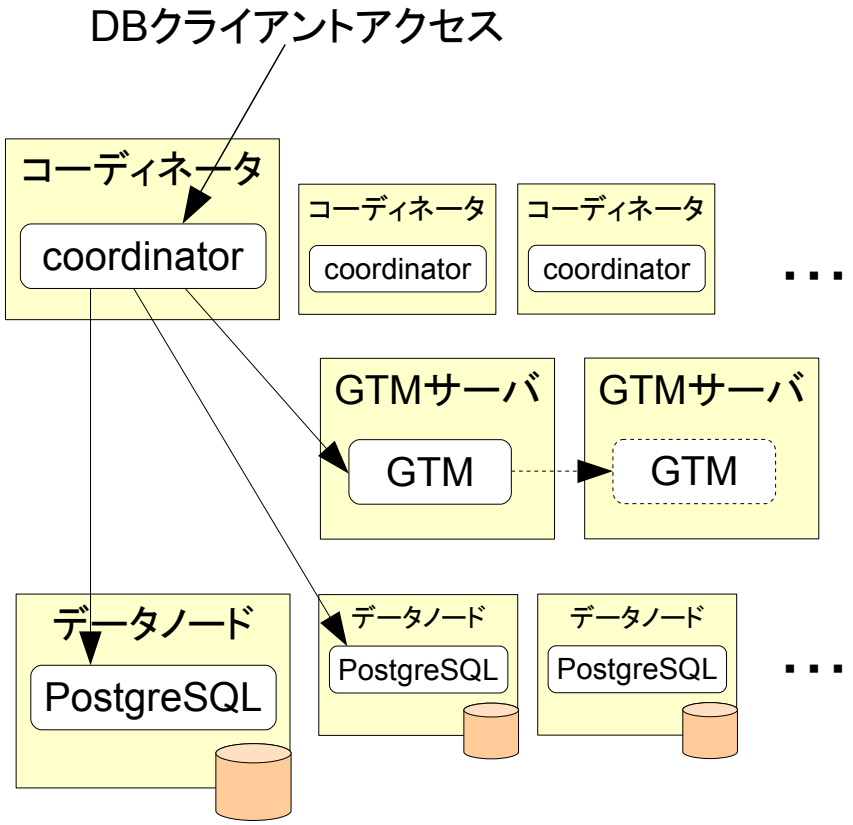
- Slony-I
  - 主役から脇役になりつつあるが、高速化や操作性向上など開発継続
    - テーブル単位のレプリケーションツール
    - データベース毎 slon プロセス
      - 別途 slon プロセス保護も必要
    - ストリーミングレプリケーションと比べると動作が遅い
    - 概念が独特で技術習得コストあり



近年の傾向（５）

- Postgres-XC 登場
  - 負荷分散クラスタ
    - データ分散格納
    - 分散MVCCで整合性維持
    - GTMサーバの多重化支援機能を持つ
    - コーディネータは多数配置可能
    - v1.0　2012年6月、
      v1.0.1 2012年9月

まだ若く
実戦投入
例は乏しい



高可用性構成を導入するには？

- 高可用性の目標水準を決めましょう
- 実現手段を選択しましょう
  - 実現できる可用性水準は
  - 構築コストは？
    - 自分たちでセットアップできる？ 購入するものは？
  - 運用コストは？
    - 運用手順作成できる？ 運用の手間の大小は？
  - 外部支援は得られる？