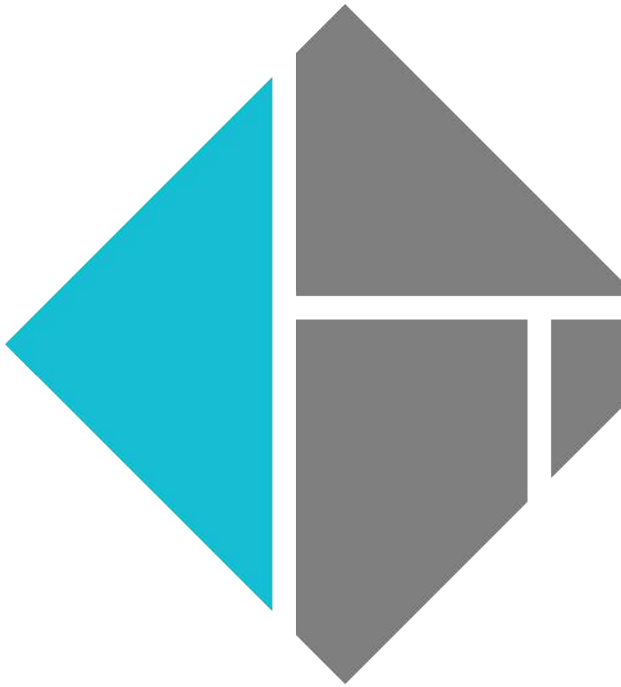
TIS
IT Holdings Group
Go Beyond

HAクラスタの構成要素として DRBDを選択するポイント



TIS株式会社
中西 剛紀

自己紹介

- 氏名：中西 剛紀 (なかにし よしのり)
- 所属：TIS株式会社　OSS推進室
- 仕事：
  - OSSのサポート, 技術支援
- 得意分野：PostgreSQL全般
  - 日本PostgreSQLユーザ会
    勉強会でたまに講演しています
    http://www.slideshare.net/naka24nori/jpug25
  - PostgreSQLエンタープライズコンソーシアム
    WGの主査としてセミナー講演してみたり
    http://itpro.nikkeibp.co.jp/atcl/column/15/052800134/052900004/?ST=oss&a

AGENDA

- HAとDRBD
- HAシステムの実現に不可欠な
  データベースの冗長化
- OSSによるHAシステムを使いこなす
  - ISHIGAKI Template のご紹介 -

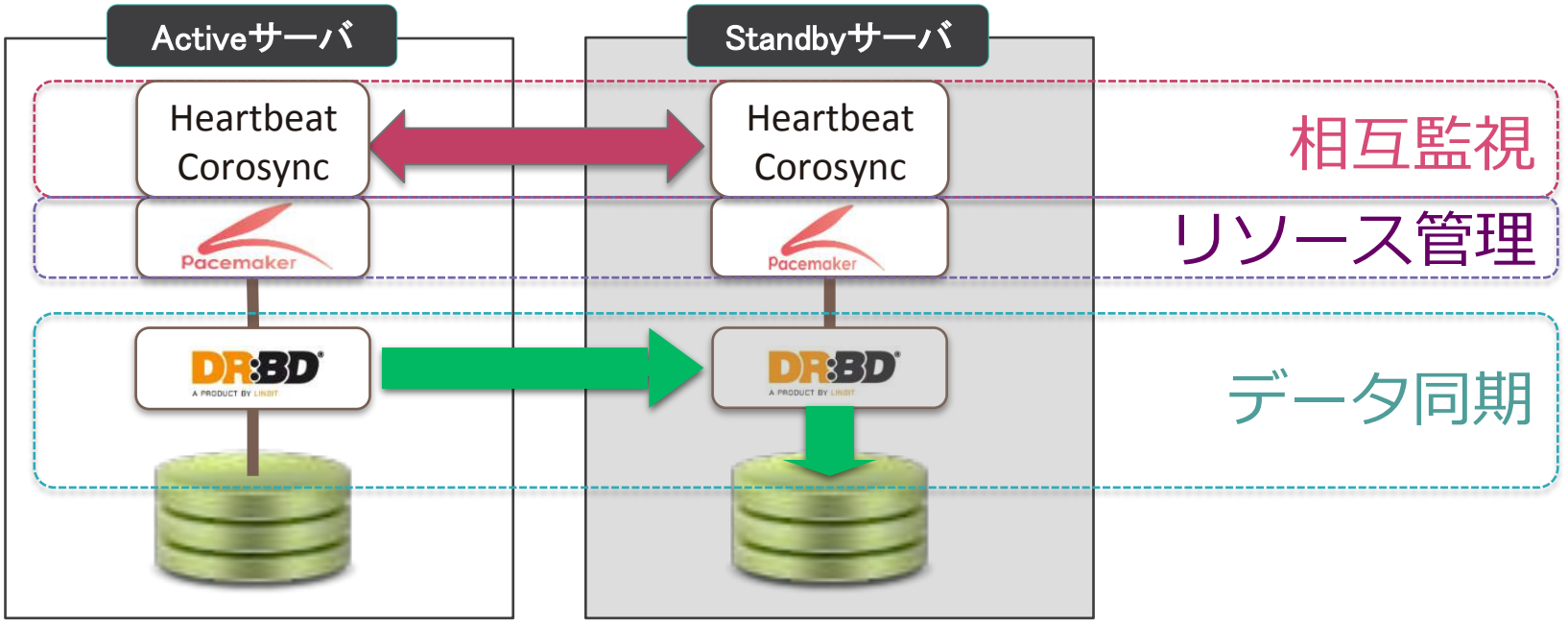
HAとDRBD

HAとは

- 企業システムにはサービスの継続性が求められている。
- High Availability（高可用性）
  - 稼働率 99.99%　⇒ 年間停止時間は53分
  - 稼働率 99.999% ⇒ 年間停止時間は　5分
- ダウンタイムを最小化
  - 冗長化
  - バックアップ
  - 障害の検知と切替を自動化

Linux-HAクラスタスタック

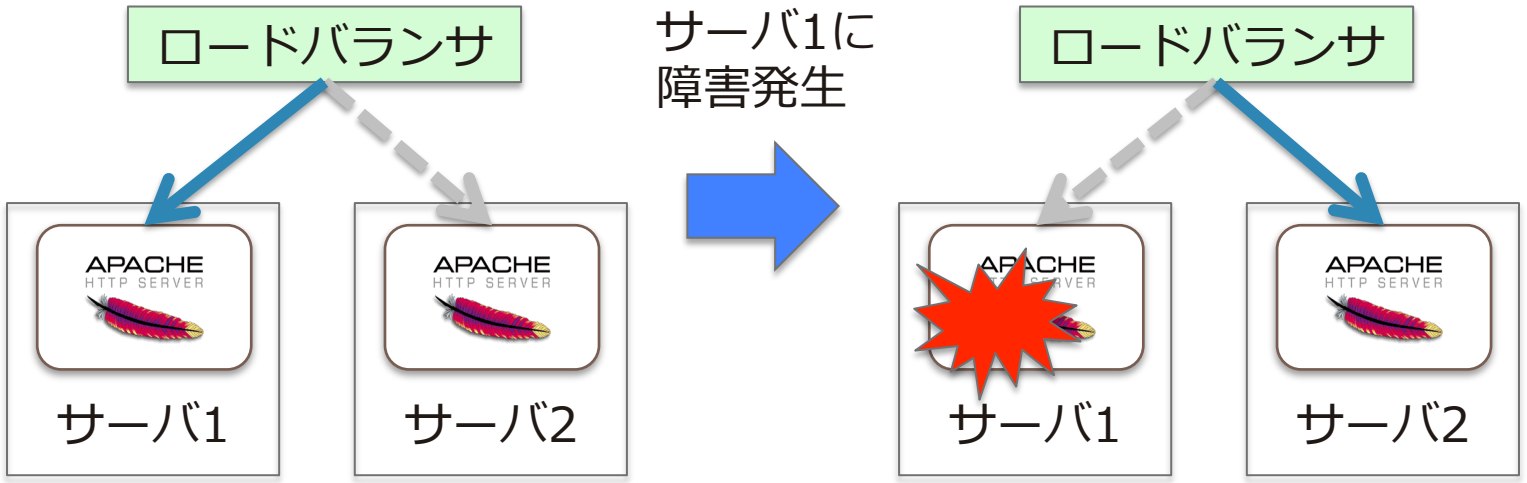
- OSSでHAクラスタを実現する組合せ
  - 相互監視：Heartbeat, Corosync
  - リソース管理：Pacemaker
  - データ同期：DRBD



HAシステムの実現に不可欠なデータベースの冗長化

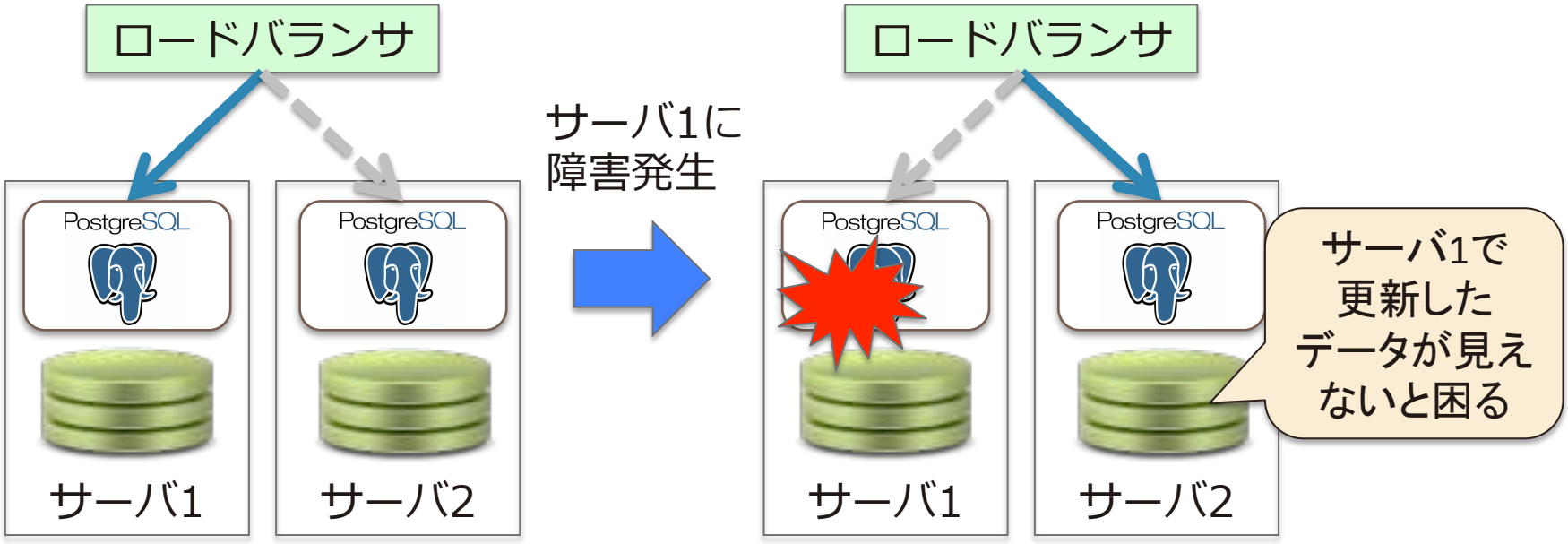
冗長化によるHAシステムの実現

- データを持たないサーバの冗長化
  - サーバを複数用意し、障害時に切り替える。



冗長化によるHAシステムの実現

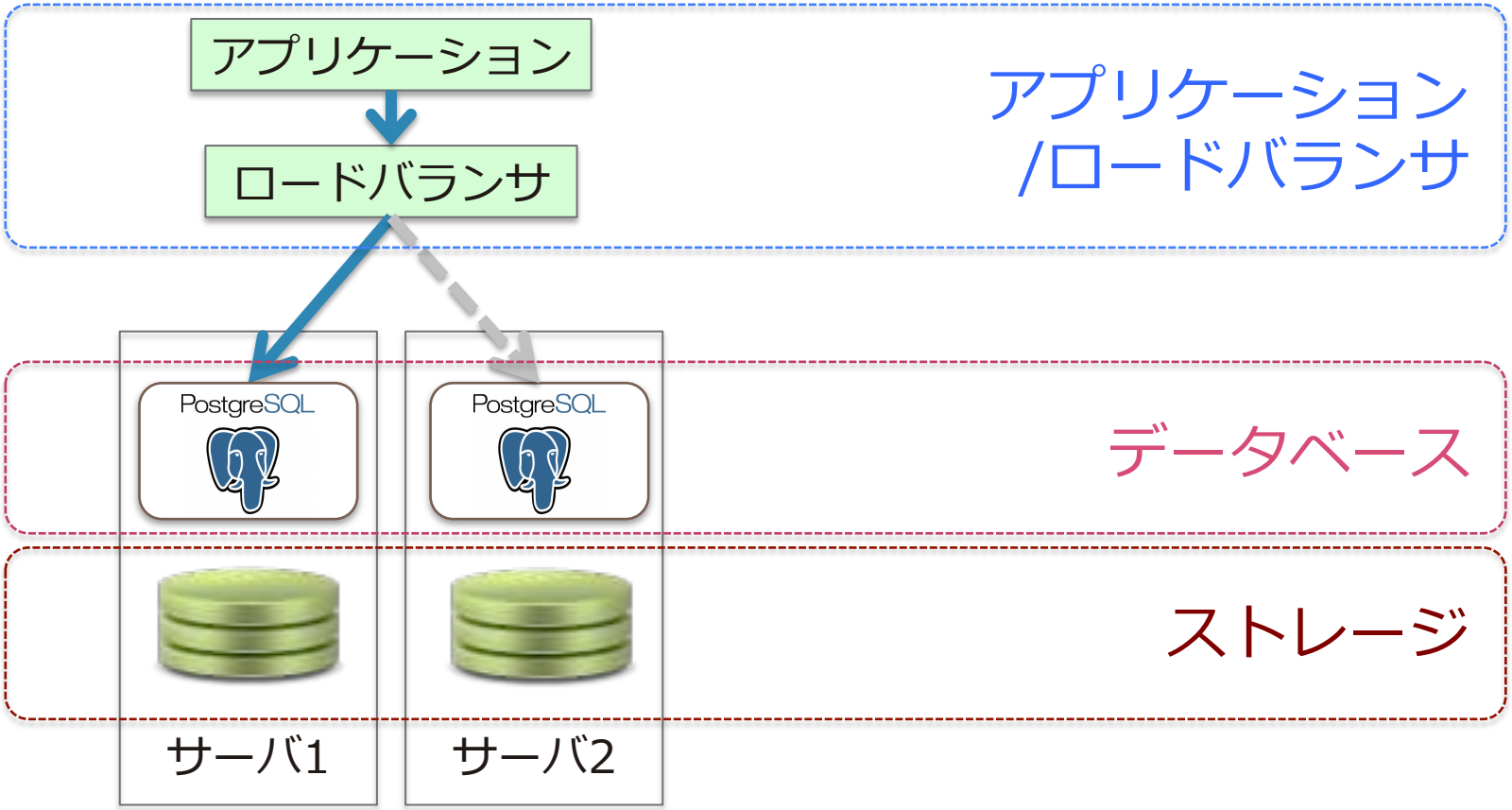
- データを持つサーバの冗長化
  - 複数のサーバ間で整合性を担保する必要あり



データを保持するデータベースを
どのように冗長化するかがポイント

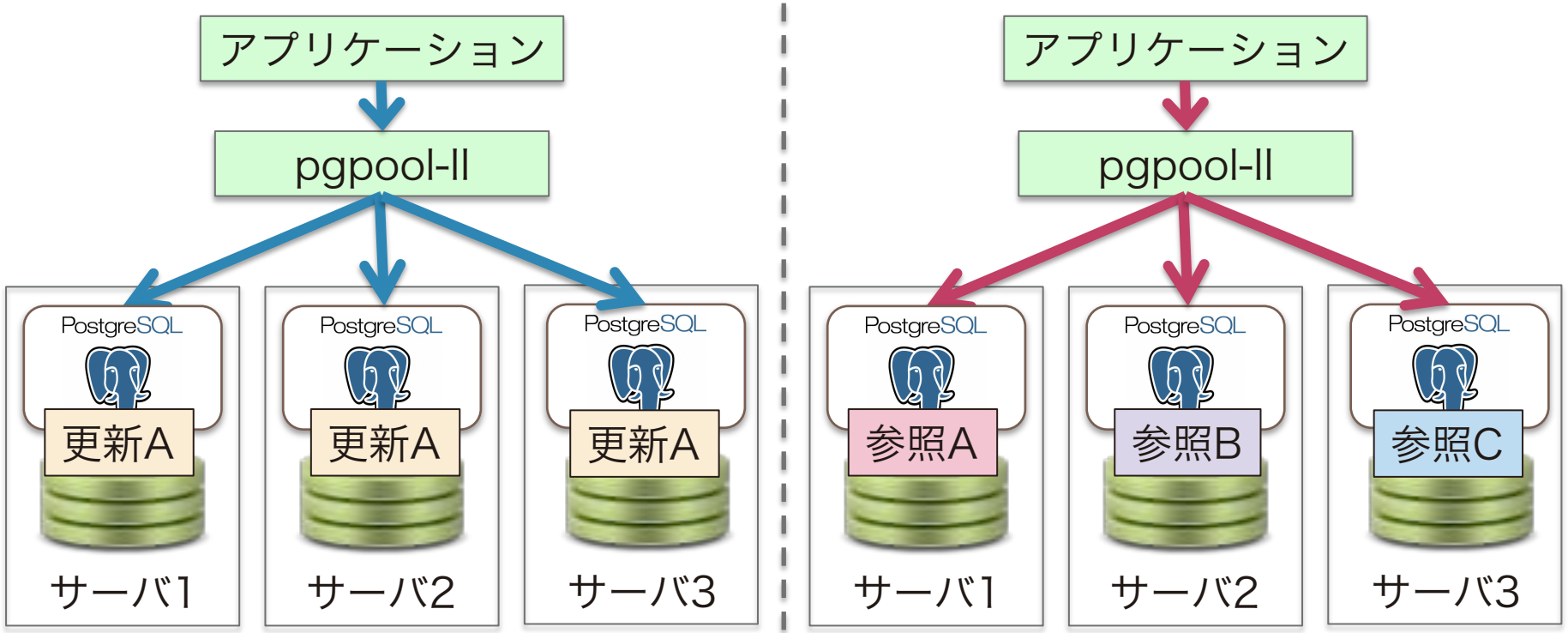
PostgreSQLの冗長化方式

- データ同期をどのレイヤで行うか？



データベースより上のレイヤで同期

- pgpool-II レプリケーションモード
  - pgpool-IIがデータ同期と参照負荷分散を担当



更新は全てのサーバで実行

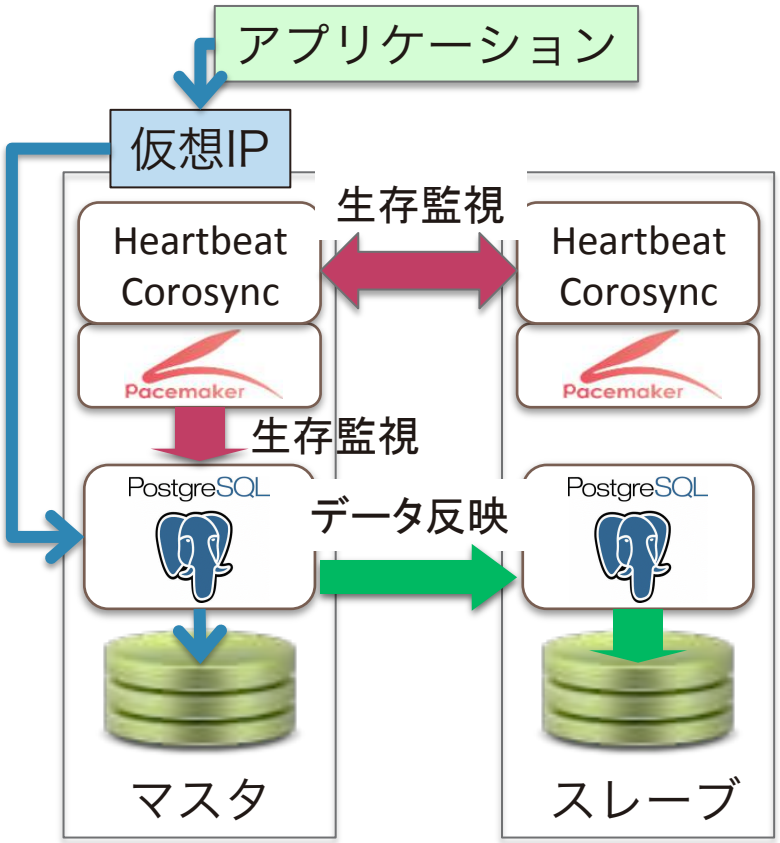
参照はどれか1つサーバで実行

pgpool-II レプリケーションモード

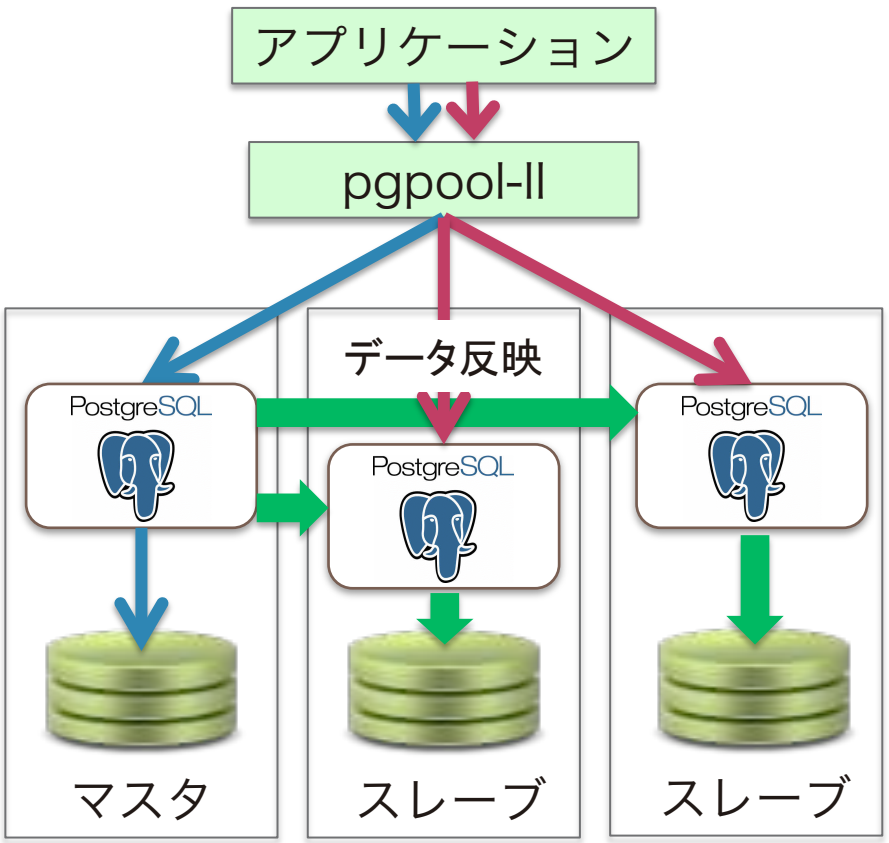
- メリット
  - pgpool-IIのみで様々な機能を実現
    - データ同期
    - 障害サーバの切り離し、リカバリ
    - 参照負荷分散
    - pgpool-II自体の冗長化(watchdog)
  - マルチマスタで運用が容易
  - 同期レプリケーション
- デメリット
  - データ更新処理の遅延リスク
  - 使用できるSQLに制限あり
  - データの整合性を守る仕組みが弱い

データベースのレイヤで同期

- ストリーミングレプリケーション
  - PostgreSQLの標準機能でデータ同期



Pacemakerとの組合せ



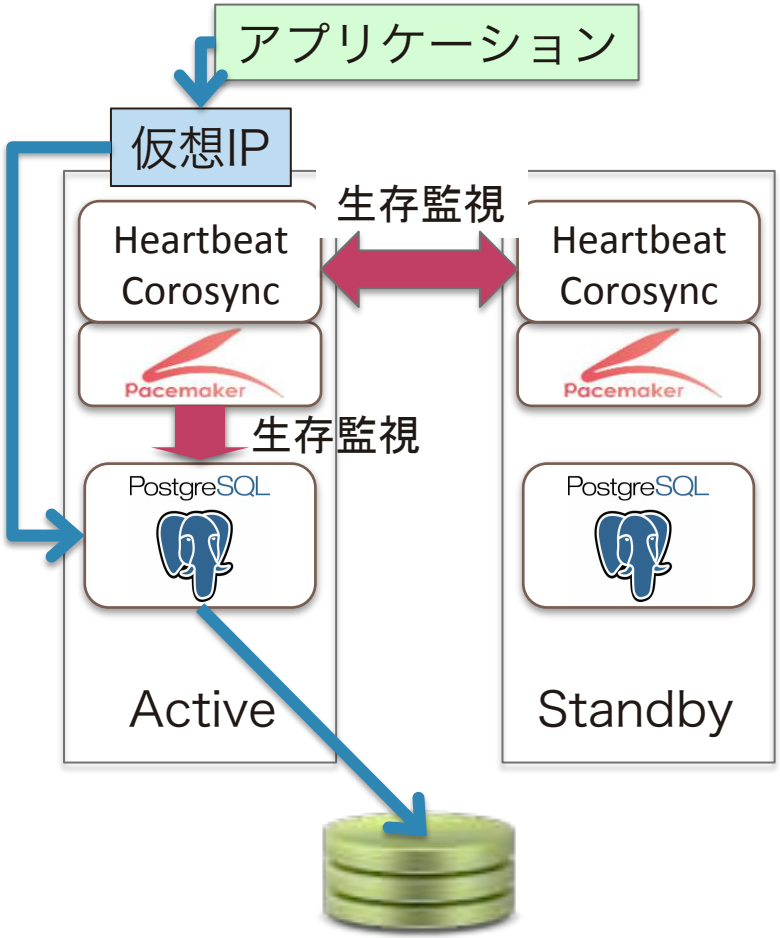
pgpool-IIとの組合せ

ストリーミングレプリケーション

- メリット
  - DB機能によりデータの整合性を担保
  - スレーブを活用した参照負荷分散も可能
- デメリット
  - シングルマスタの障害時運用は若干複雑
  - 完全な同期レプリケーションは不可能
  - PostgreSQLのレプリケーション機能は発展途上な部分もある。

ストレージのレイヤで同期

- 共有ディスク

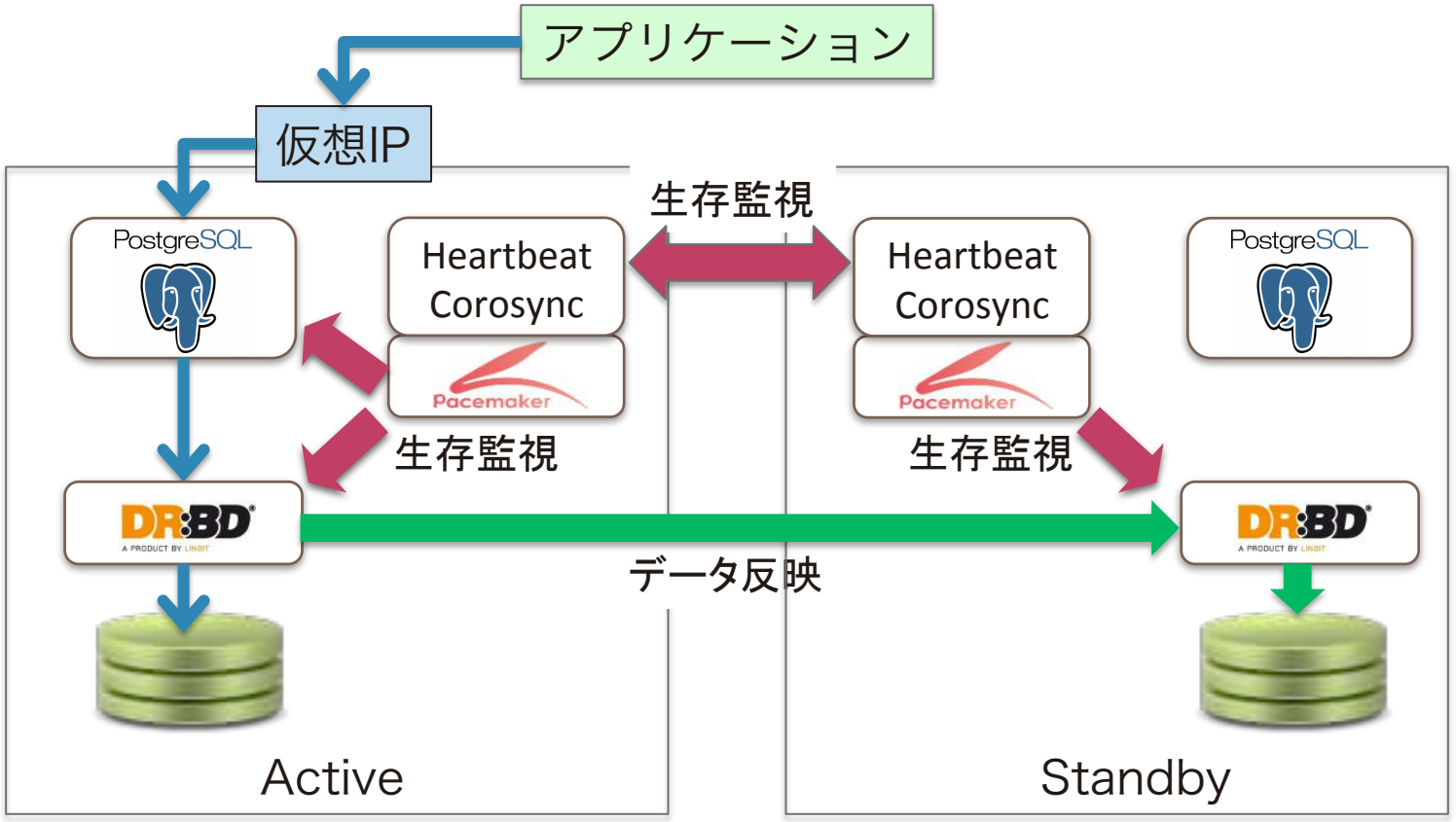


共有ディスク

- メリット
  - DBの使い勝手はシングルサーバと同様
  - データの整合性は確実に担保
  - シンプルな構成で運用負荷が小さい
  - 古くから実現できていた方式で実績多数
- デメリット
  - 共有ディスク自体の障害への対策が必要
  - 共有ディスクのコスト

ストレージのレイヤで同期

- DRBD
  - PostgreSQLのデータパーティションを同期



DRBD

- メリット
  - DBの使い勝手はシングルサーバと同様
  - DB以外のデータも含めて冗長化
  - 完全な同期レプリケーションを実現
  - データの整合性も確保
- デメリット
  - 待機系を活用した参照負荷分散は難しい
  - レプリケーションの負荷
  - データ更新処理の遅延リスク

方式の使い分け方

- 特徴を理解して使い分けよう
- pgpool-IIレプリケーションモード
  - リアルタイム同期、シンプルな運用
- ストリーミングレプリケーション
  - データ整合性、更新性能維持、参照負荷分散
- DRBD
  - データ整合性、リアルタイム同期、
    サーバ全体の冗長化

OSSによるHAシステムを使いこなす
- ISHIGAKI Templateのご紹介 -

HAシステムを活用する上での課題

- HAシステムを構築する上で、
  様々なミドルウェアの知識が求められる。
- ミドルウェアを組合せて正常動作するか、
  性能や可用性の要求に応えられるか、
  をあらかじめ検証する必要がある。

HAシステムを実際に活用するハードルは高い

ISHIGAKIのコンセプト

- OSSの利用促進を目的に開発をスタート
- OSSを利用する際に投げかけられる不安
  - OSSを本番業務に使って大丈夫？
  - 非機能要求（性能や可用性）への適合性は机上調査では（やってみないと）わからない

検証ケースを積み上げて洗練した推奨構成を提供

- 実機検証から生み出されたノウハウを実装したOSSの推奨パッケージ

ISHIGAKIを構成するOSS

ISHIGAKIの構成パターン

TIS IT Holdings Group | Go Beyond

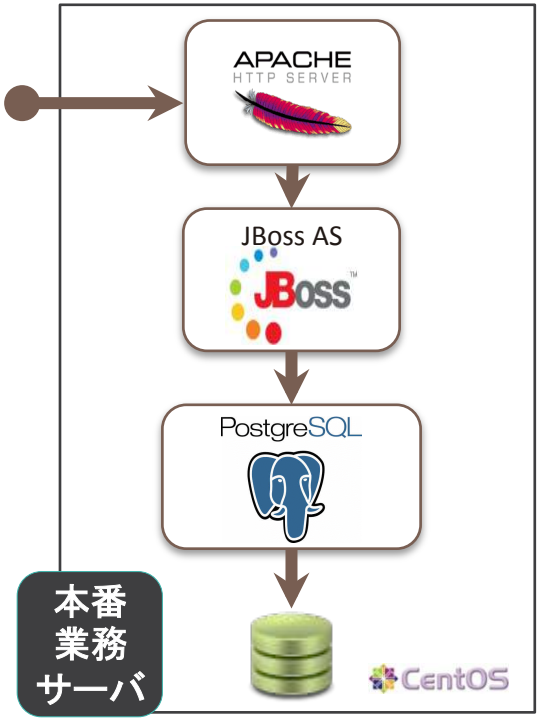
Single Edition

HA Edition

Cluster Edition

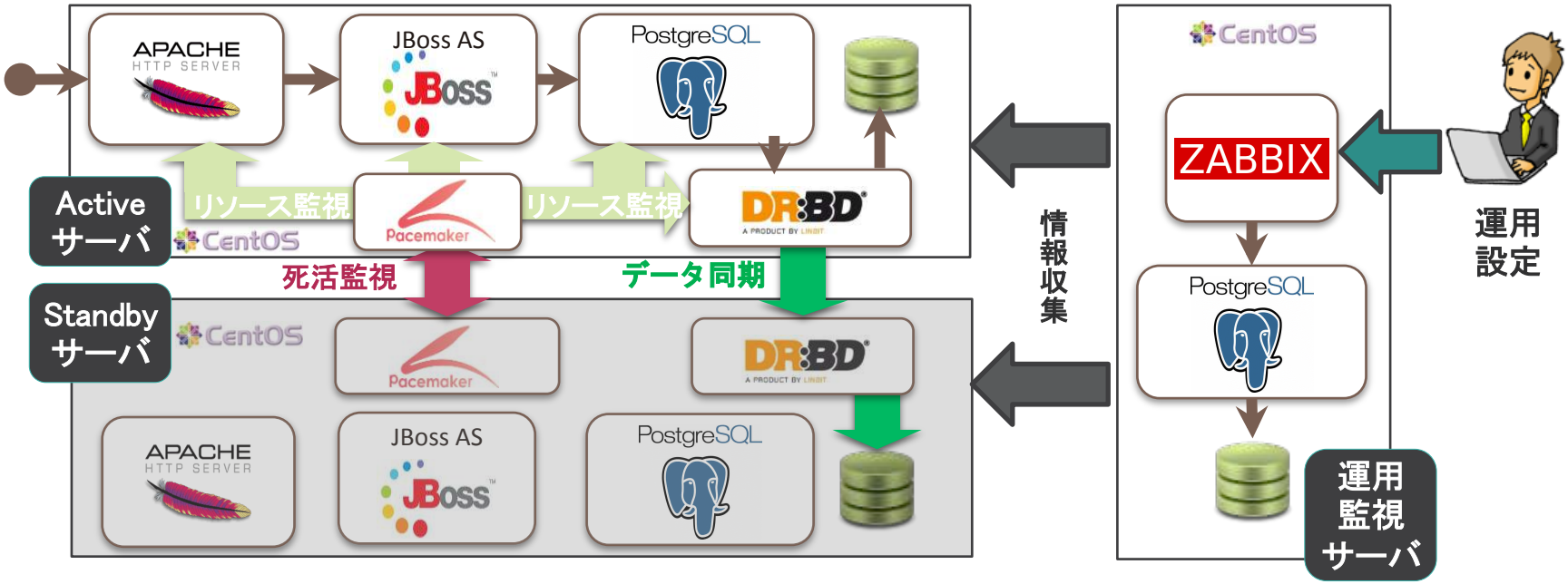
Single Edition

- OSSベースのアプリケーション基盤を容易に構築するシングルサーバ



HA(High-Availability) Edition

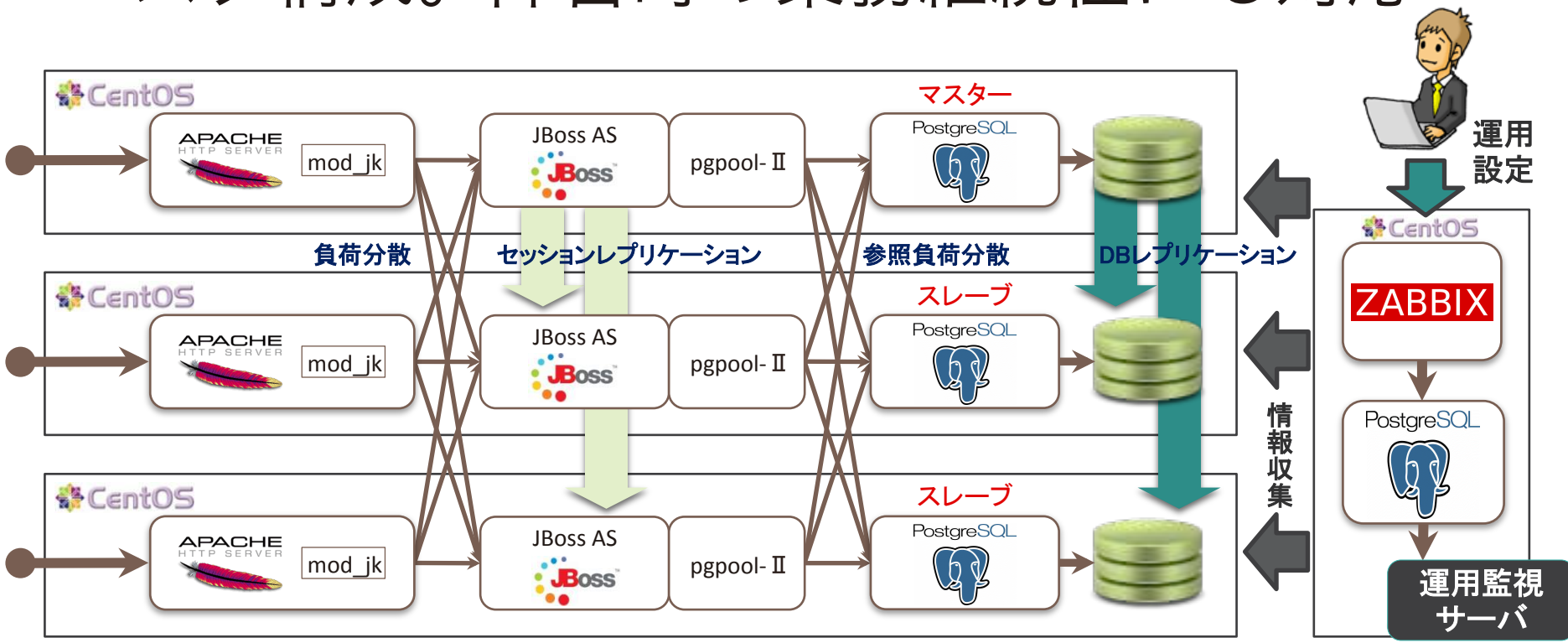
- 長時間の業務停止を回避するHAクラスタ



サーバをActive/Standby型で冗長化し、可用性を向上
DRBDにより、安価なディスクで共有ディスクを代替

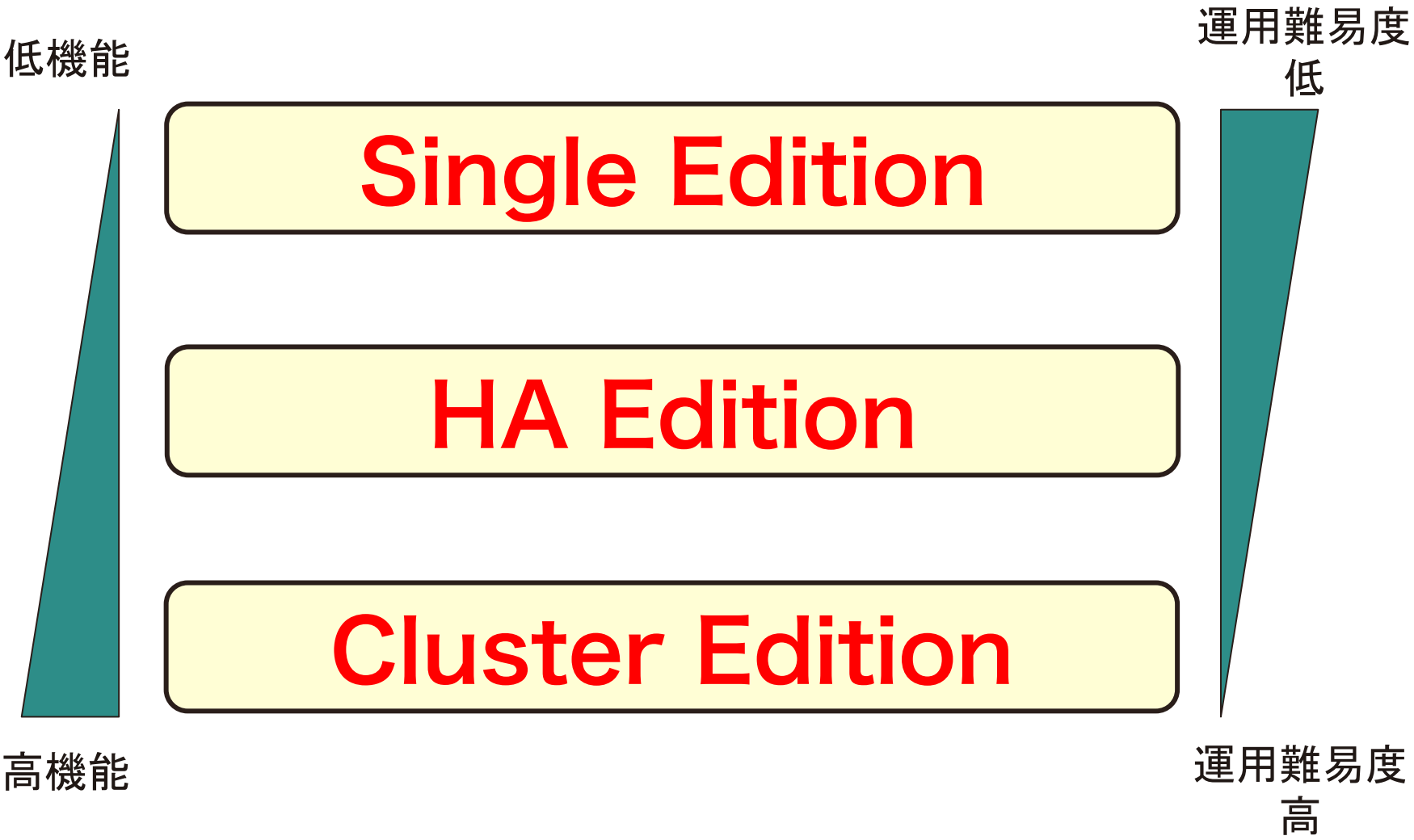
Cluster Edition

- 複数台で分散して処理を行う高性能クラスタ構成。障害時の業務継続性にも対応



サーバーをActive/Active型で冗長化し可用性を向上
処理量増加に応じたスケールアウト型での性能向上を実現

構成パターンの使い分け



ISHIGAKIを利用するメリット

TIS IT Holdings Group | Go Beyond

- 実機での事前検証による信頼性

実例：HA Edition における検証ケース例

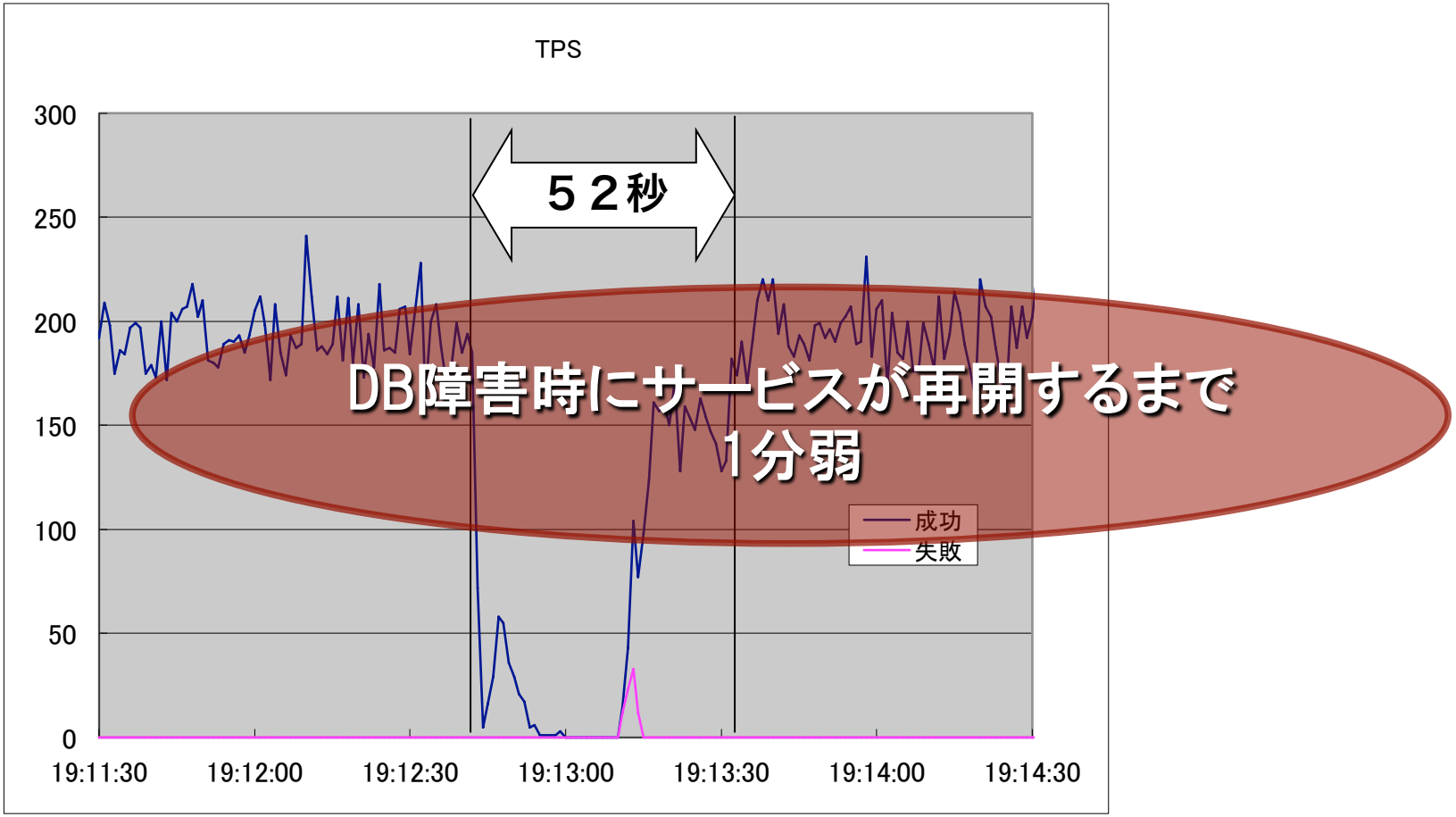
カテゴリ	テストケース	実行する操作	想定される結果
可用性（プロセス停止時のフェイルオーバー）	監視対象プロセスが1回ダウンした場合（Apache, JBoss, PostgreSQL）	kill -9 コマンドで対象プロセスを停止	Active機でプロセス再起動
	監視対象プロセスが2回ダウンした場合（Apache, JBoss, PostgreSQL）	kill -9 コマンドで対象プロセスを停止し、Pacemaker が再起動させた後に、再度 kill -9 コマンドで対象プロセスを停止	Standby機にフェイルオーバー
	Pacemaker のプロセスがダウンした場合	kill -9 コマンドで Pacemaker プロセスを停止	Standby機にフェイルオーバー
可用性（HW停止時フェイルオーバー）	Active側仮想マシンが電源断した場合	Active機仮想マシンの電源を強制終了	Standby機にフェイルオーバー
	Standby側仮想マシンが電源断した場合	Standby機仮想マシンの電源を強制終了	Active機の継続稼動
	サービスLANが断線した場合	Active機のサービスLAN接続NICを停止	Standby機にフェイルオーバー
	インターコネクトLANが断線した場合	Active機のインターコネクトLAN接続NICを停止	Active機の継続稼動
	DRBDデータ転送LANが断線した場合	Active機のDRBDデータ転送LAN接続NICを停止	Active機の継続稼動
	インターコネクト/DRBDデータ転送LANが断線した場合	Active機のインターコネクトLAN接続NICとDRBDデータ転送LAN接続NICを停止	Active機の継続稼動
可用性（フェイルバック）	復旧機をStandbyとしてクラスタに再加入	復旧機に環境構築し、Pacemaker を起動	Active機は継続稼動 Standby機がHAクラスタに加入
	Standby機をActive機に昇格させる場合	Pacemaker で migrate コマンドを実行	Standby機がActive機に昇格
性能	Starter へ高負荷をかけた時のスループット、リソース使用状況	CPU使用率が70%程度になるよう負荷をかける	（Standard評価の基準値）
	Standard へ高負荷をかけた時のスループット、リソース使用状況	CPU使用率が70%程度になるよう負荷をかける	Starterと同程度のスループット Starterよりリソース使用量大
	高負荷環境下のPacemaker の挙動	高い負荷を掛ける	Active機の継続稼動

実際に組合せた構成で性能、可用性観点の検証を実施済

ISHIGAKIを利用するメリット

- 実機での事前検証による信頼性

実例：Cluster EditionにおけるDB障害時のサービス継続状態



ISHIGAKIを利用するメリット

TIS IT Holdings Group | Go Beyond

- 検証によるノウハウを活かした推奨設定

実例：HA Editionにおける事前設定済の設定項目（一部）

Apache
- •マルチスレッド化
- •インターネットからの性能・セキュリティ
- •JBossへの接続
- •PostgreSQL管理ページへのアクセス許可

JBoss
- •デフォルトデータソース変更
- •アクセスログ
- •HTTPヘッダのセキュリティ強化
- •Webコネクタの性能・セキュリティ強化
- •JVMの起動オプション変更
- •ログの変更、ローテーションの設定
- •サーバ設定とアプリケーションの分離
- •アプリケーション用データソースの コネクションプール・暗号化パスワードへの対応

PostgreSQL
- •コネクション数の拡大
- •共有バッファサイズの拡大
- •メディアリカバリへの対応
- •アーカイブログの出力先の指定
- •高速な検索方式の利用頻度UP

Pacemaker
- •インターリンク通信詳細
- •フェイルオーバ動作
- •監視間隔・タイムアウト
- •各ミドルウェアへの監視設定

DRBD
- •タイムアウト
- •データ同期速度

170か所以上を設定済

ISHIGAKIを利用するメリット

TIS IT Holdings Group | Go Beyond

- すぐに稼働できる形でパッケージング
  - Apache, Tomcat, JBoss, PostgreSQL, Pacemaker, DRBD, Zabbix
- 複雑なクラスタシステムも速やかに構築
  - Chefを採用し、導入手順をコード化

CHEF™

ISHIGAKIを利用するメリット

TIS IT Holdings Group | Go Beyond

- 導入、運用者向けのドキュメントを提供
  - 各エディションが前提とするシステム構成
  - 導入時の作業、導入後の動作確認手順
  - 障害からの復旧手順
  - データベースのバックアップ、リカバリ手順
- 稼働後の運用基盤を速やかに構築
  - システム監視を行うZabbixサーバを自動構築
  - 各エディションのシステム構成に合わせた監視設定を自動化

ISHIGAKIの提供形態、サービス

TIS IT Holdings Group | Go Beyond

- 弊社のインテグレーション案件で
  ISHIGAKI Templateをご利用いただけます。
- TISエンタープライズOSSサポートサービス
  https://www.tis.jp/service_solution/oss/
  - OSS技術コンサルティング
  - OSSプロダクトサポート

まとめ

- クラウドでも高可用なシステムを
  - ミドルウェアを活用したHAシステムの実現
  - Linux-HAは有効なソリューション
- 冗長化のポイントはデータ整合性確保
  - 方式の特徴を見極めて選択
  - データ整合性を重視するシステムでは
    これからもDRBDは有効なソリューション
- HAシステムのご利用は計画的に
  - 事前に十分な検証とノウハウの蓄積を
  - ISHIGAKI Templateもご検討ください


TIS
IT Holdings Group
Go Beyond