Linux-HA を利用した Zabbix2.0 の高信頼クラスタの構築、検証報告

ミラクル・リナックス（株）

吉田

2013/2/22

アジェンダ

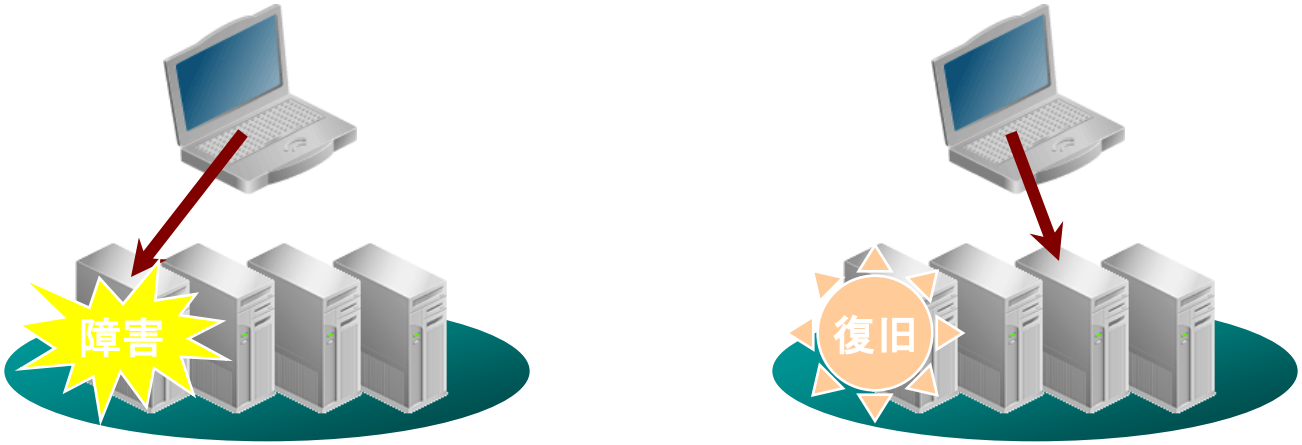
- クラスタリングとは
- 今回の構成
- DRBD、Zabbix等の構成ソフトウェア
- Zabbix2.0新機能
- 構築方法
- バックアップについて
- 参考資料

クラスタリングとは

- HAクラスタとは
- ディスクミラー型クラスタの構成

HAクラスタとは

- **High Availabilityクラスタ**
  - サーバを冗長化し、システムの停止時間を最小限に抑える
  - 業務の可用性（availability）を向上させるクラスタシステム



- **クラスタシステムが求められる主なサーバ**
  - 基幹システムの データベースサーバ
  - 社内インフラとして頻繁に利用する ファイルサーバ　メールサーバ
  - 社内サイト、社外サイトを提供する　Webサーバ

なぜクラスタ化するのか

- 監視サーバが止まると監視になりません
  - 監視サーバが落ちている時に障害が起きたら…
- 稼働率と年間停止時間

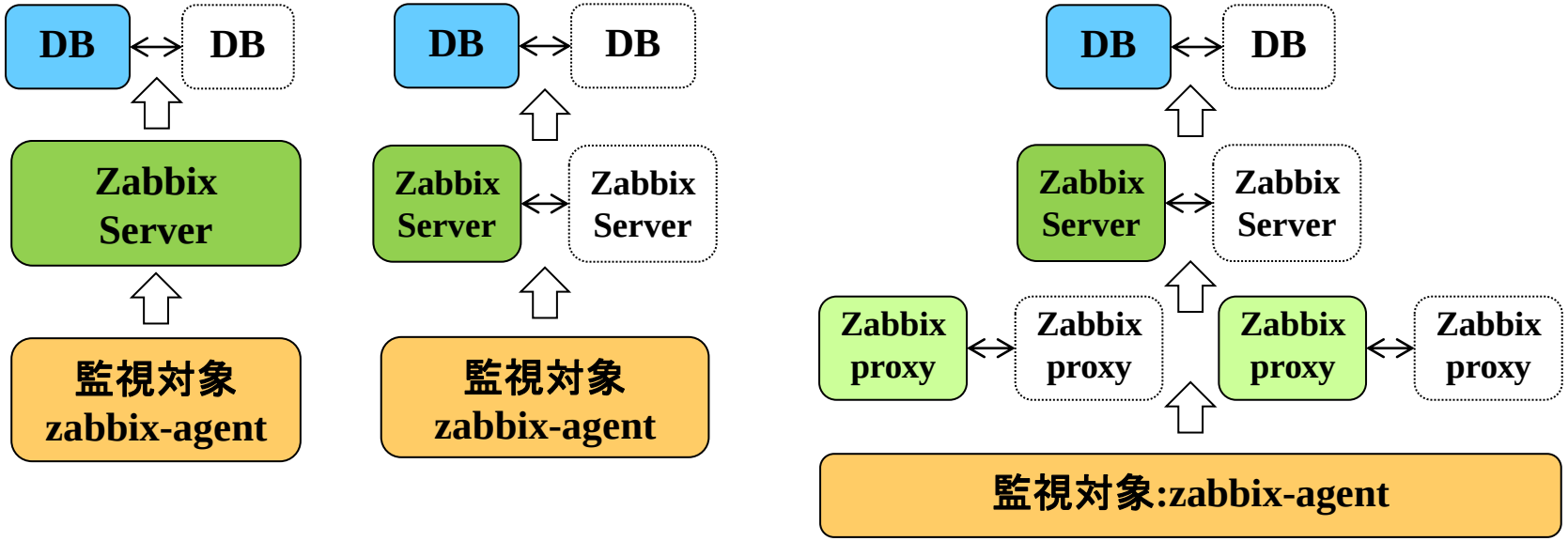
稼働率	年間停止時間
99%	3日15時間36分
99.9%	8時間46分
99.99%	52分34秒
99.999%	5分15秒
99.9999%	32秒

今回の構成

- Zabbix冗長化構成の選択枝
- 構成図
- 構成ソフトウェア

Zabbix冗長化構成の選択

- DBのみ冗長化
- DBとZabbixサーバを冗長化
- Zabbix-proxyも使用して全て冗長化
- 1台に同居or専用DBサーバ&専用Zabbixサーバ



対象規模のサイズ例

小規模

監視対象:
30台または3000監視項目まで

CPU: Intel Atom N270 1.6GHz
メモリ: 2GB
HDD: 160GB
MIRACLE ZBX2100

小～中規模

監視対象:
100台または10000監視項目まで

CPU: Intel Atom D510 1.6GHz
メモリ: 2GB
HDD: 500GB
MIRACLE ZBX3100

大規模

監視対象:
1000台または100000監視項目まで

CPU: Intel Xeon E3-1270 3.40GHz
メモリ: 8GB
HDD: 450GB x2（SAS/RAID1）
MIRACLE ZBX8000a

※監視間隔5分、ログ/SNMPトラップ監視を含まない場合の目安

Zabbixベンチマーク

- Zabbixベンチマーク

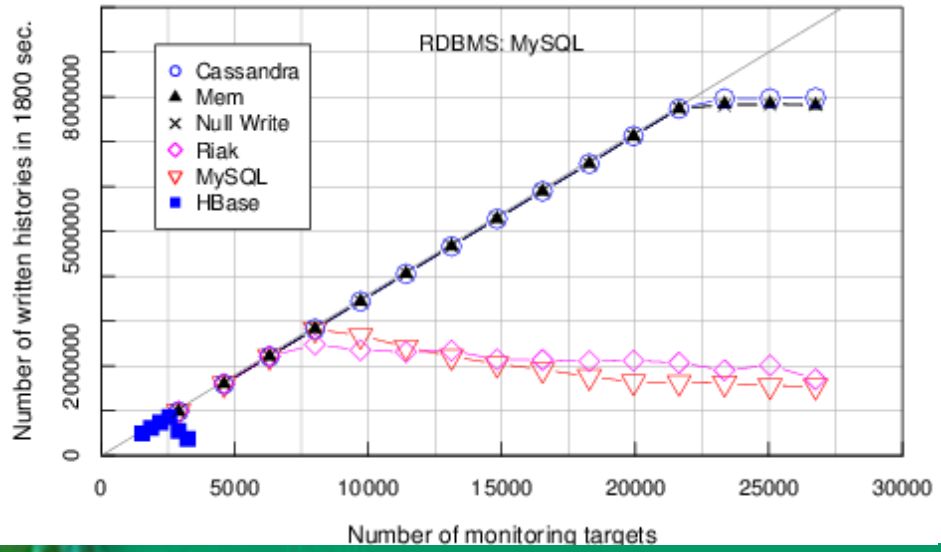
  https://github.com/miraclelinux/zabbix-benchmark

- NoSQL版Zabbix

  https://github.com/miraclelinux/MIRACLE-ZBX-2.0.3-NoSQL

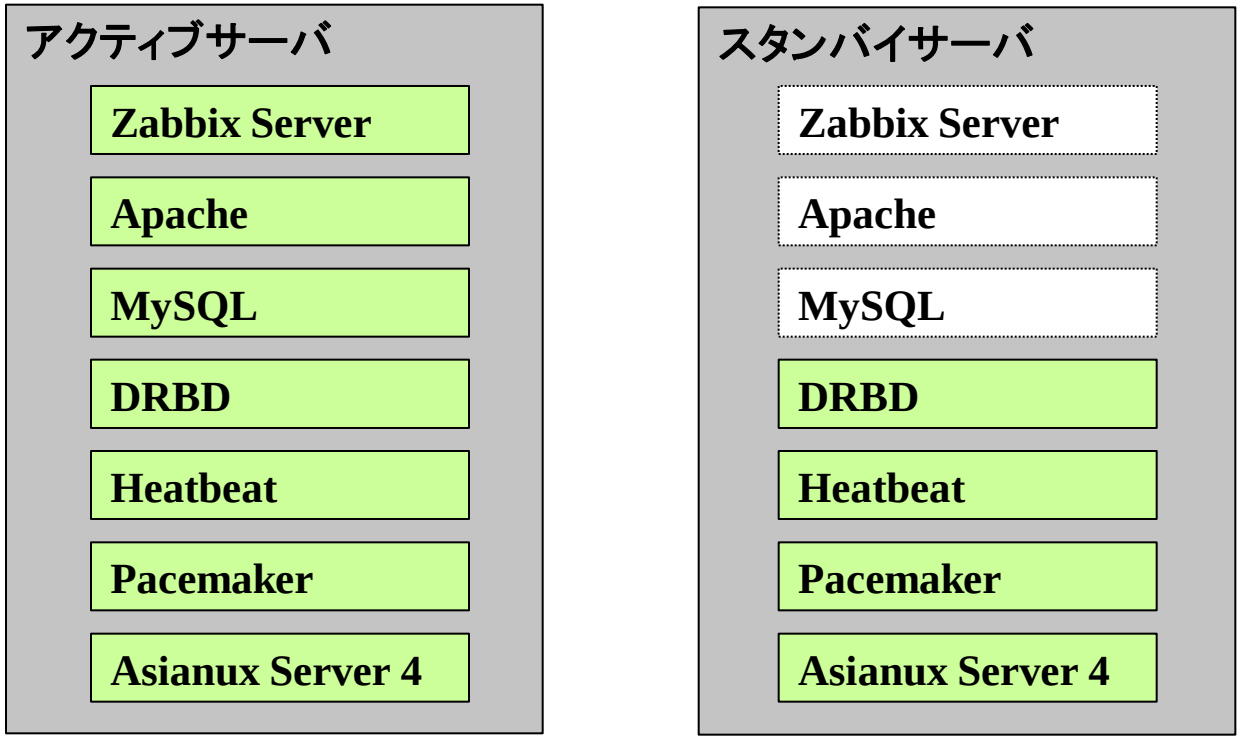
- NoSQL を使ったZabbixの高速化

  http://cloud.watch.impress.co.jp/docs/news/20130214_587738.html



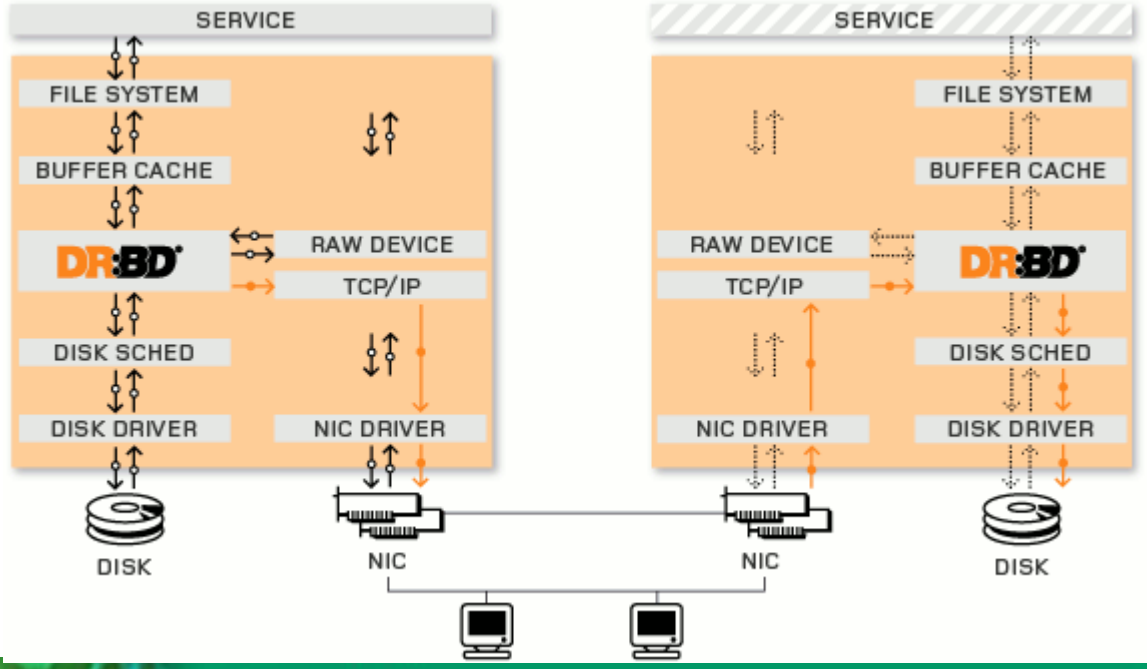
今回の例(最小構成)

- サーバ2台でDBとZabbixサーバを冗長化
- DBのデータファイル部分をDRBDでミラーリング
- Zabbixサーバは障害時フェイルオーバー



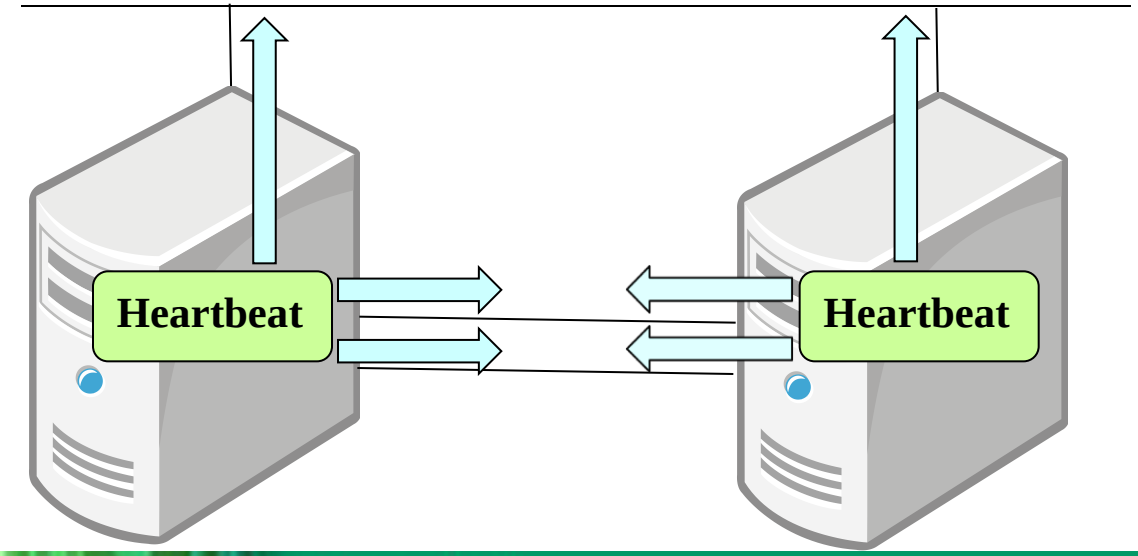
DRBD

- DRBD（Distributed Replicated Block Device）は、ネットワークを通じてハードディスク（ブロックデバイス）をリアルタイムに複製（同時複製）するソフトウェアです。大切なデータを失わ ないためのバックアップや、サービスの冗長化に役立つソフトウェアとして広く使用されています。
- http://www.drbd.jp/



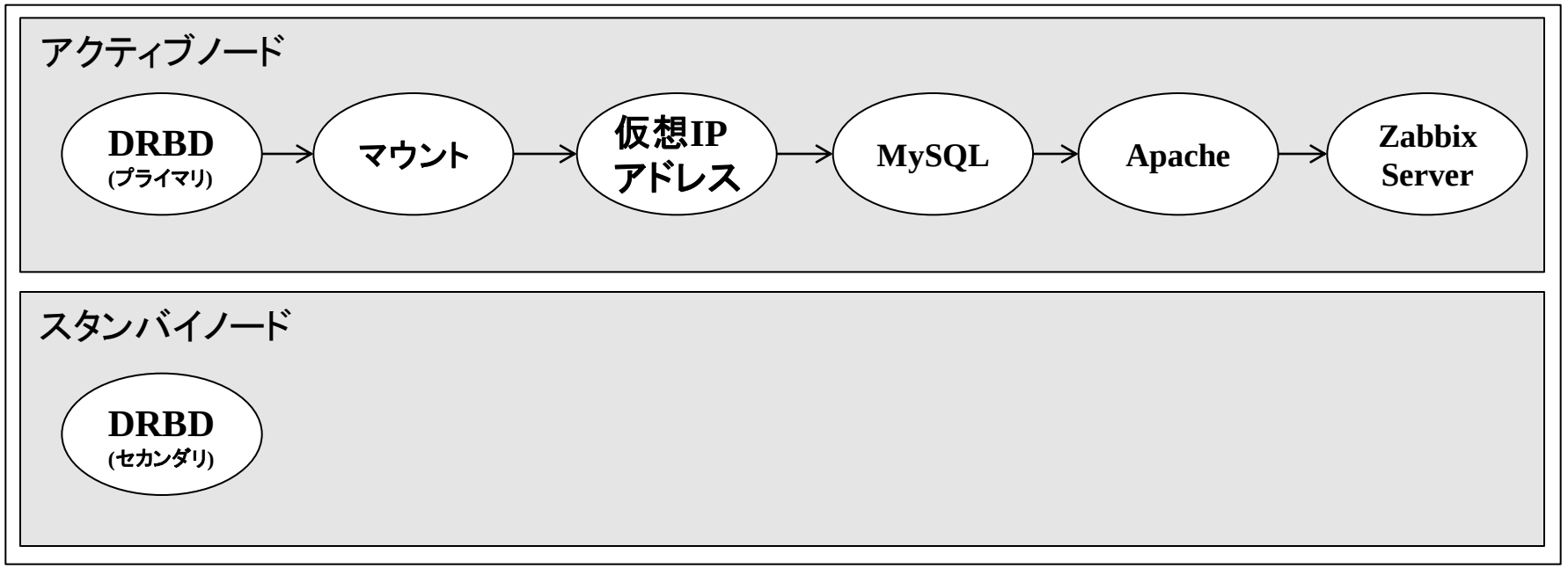
Heartbeat

- 各サーバは「ハートビート」パケットを他のノードに向けて定期的に送信する。他のサーバは、応答を返す。
  - 一定時間以上応答がないノードは、クラスタから取り除かれる。
- HeartbeatはPacemakerが必要とする通信を仲介
  - HeartbeatとPacemakerの実行ログを管理する。
- Heartbeat経路も冗長化推奨



Pacemaker

- Pacemakerは「クラスタリソース管理システム」
- リソース(サービス等)の開始、終了、死活監視を実行



Zabbix＝統合監視ツール

Zabbixの主な特徴

- オープンソース・ソフトウェア
- Webインターフェースから監視設定、情報表示が可能
- リアルタイムなマップ、グラフ表示
- 監視データをデータベースに長期間蓄積
- 監視テンプレートによる監視設定の一元管理
- 大規模システム向けの監視機能を標準で搭載
- マルチプラットフォーム対応
- 豊富な監視機能

Zabbix2.0変更点

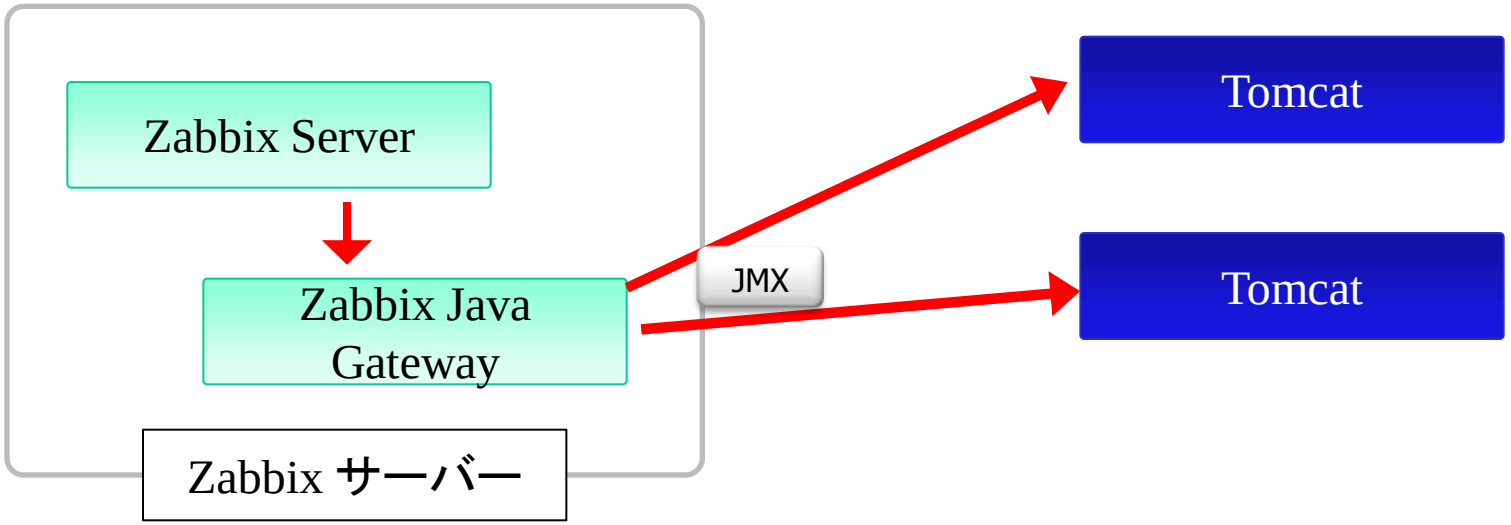
代表的なものを紹介

Zabbix APIの正式実装

- 1.8 において試験的に実装されていたZabbix APIを正式に実装
- JSON-RPCを利用したAPIにより、API経由での監視結果の取得や監視設定の追加・変更・削除が実現可能
- コミュニティーにおいて、すでにRubyによるコマンドラインツールや各種プログラム向けのライブラリが存在

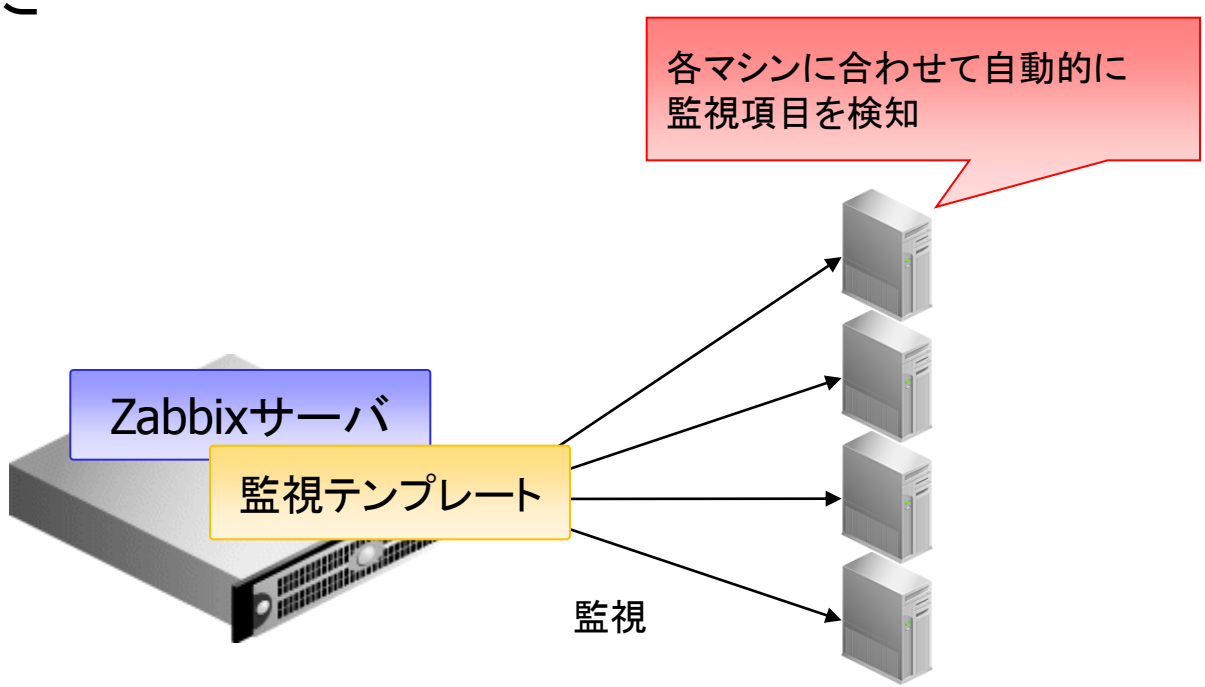
JMXを利用したJavaアプリケーション監視

- 以前はzapcatを利用し実現していたJavaアプリケーションの監視機能をZabbix自体に実装
  - ZabbixサーバーやJavaアプリとは別にZabbix Java Gatewayを経由してデータを取得
  - Zapcatと比べ、記法は多少変更
- JBoss, WebLogic, WEBSphere, Tomcat などに対応



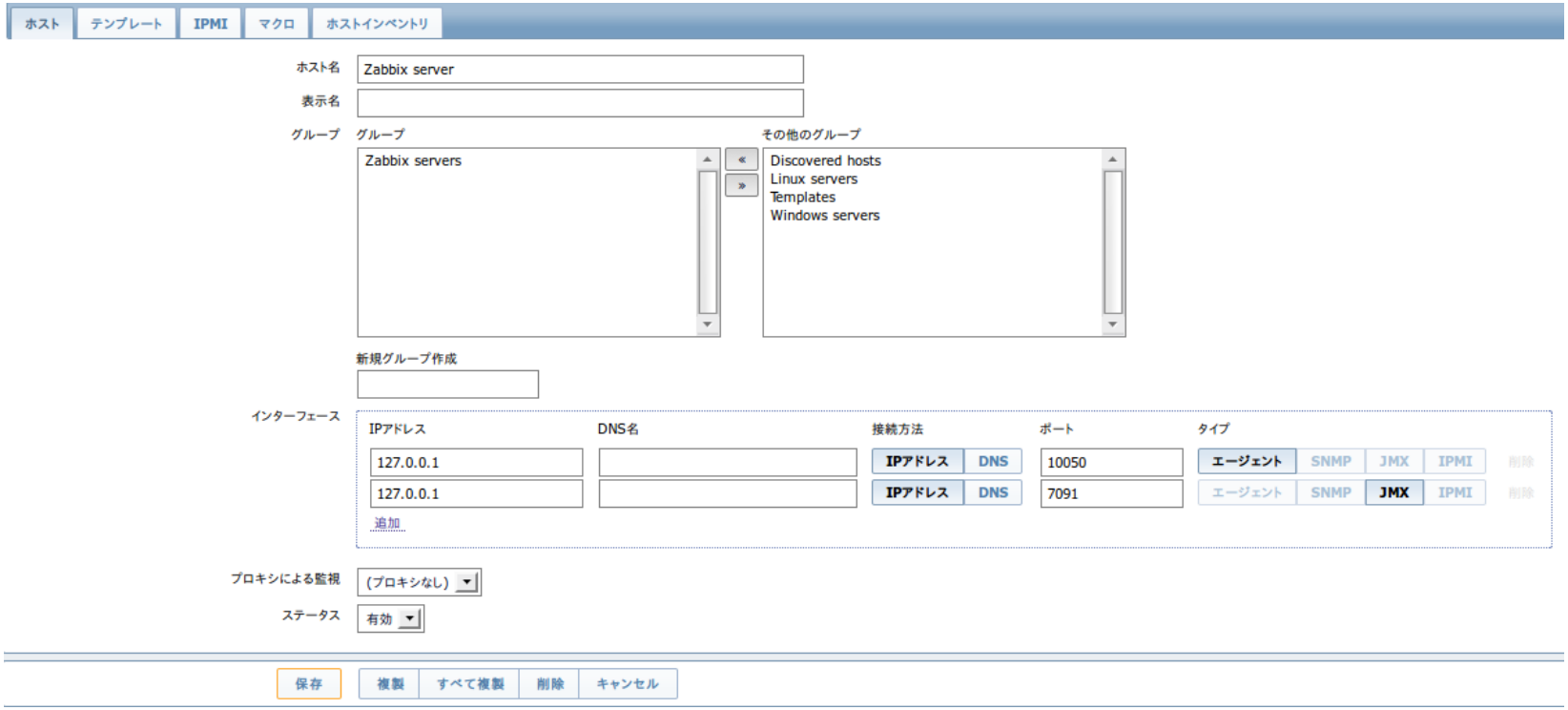
監視対象の自動検知(Low-level iscovery)

- ネットワークデバイス名、ディスクデバイスの構成などを検知し、自動的に監視対象に追加
- 自動検知により、細かな構成の違いによるテンプレートの分割が不要に



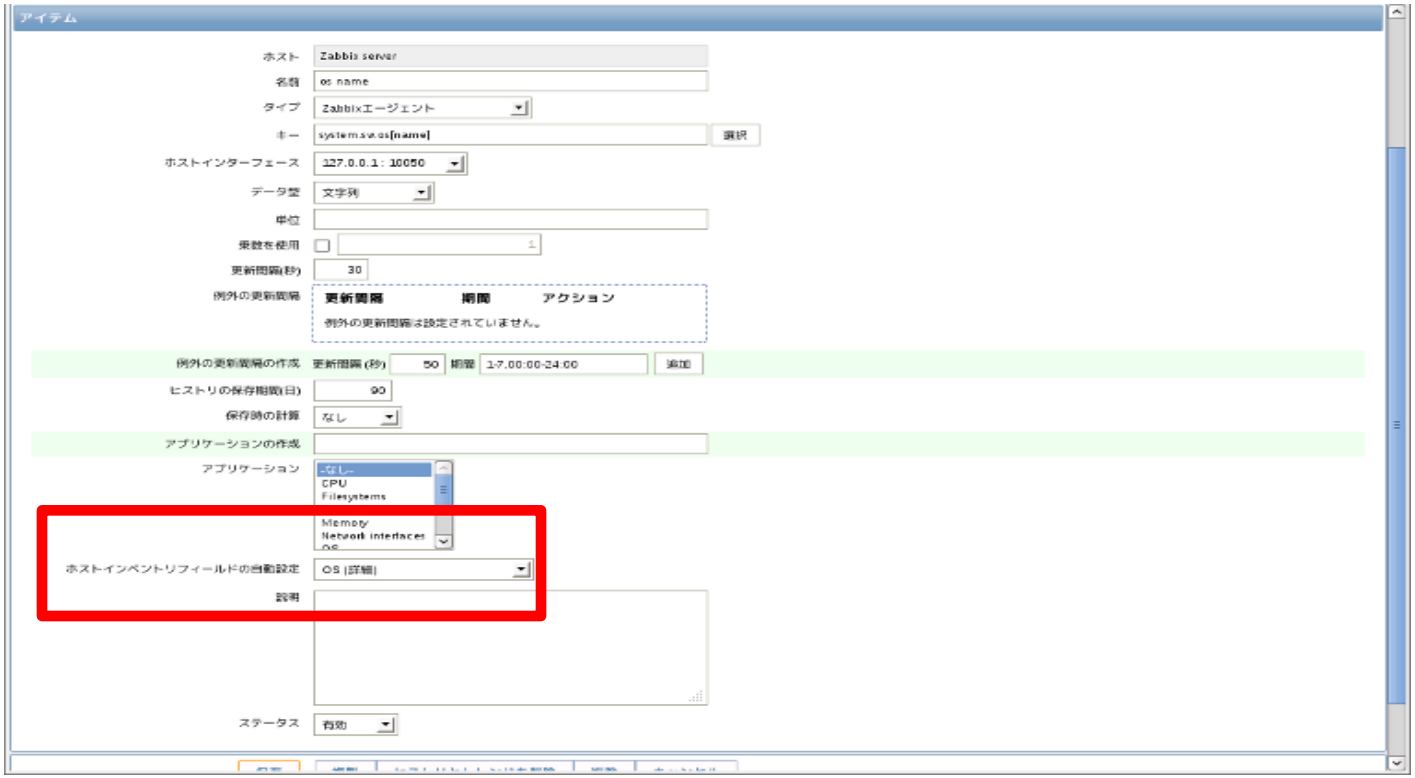
Webインターフェイスのデザインの改善

- 設定画面へのタブの採用
- AJAXによる、より柔軟な操作の実現
- ダッシュボードのレイアウトがカスタマイズ可能に



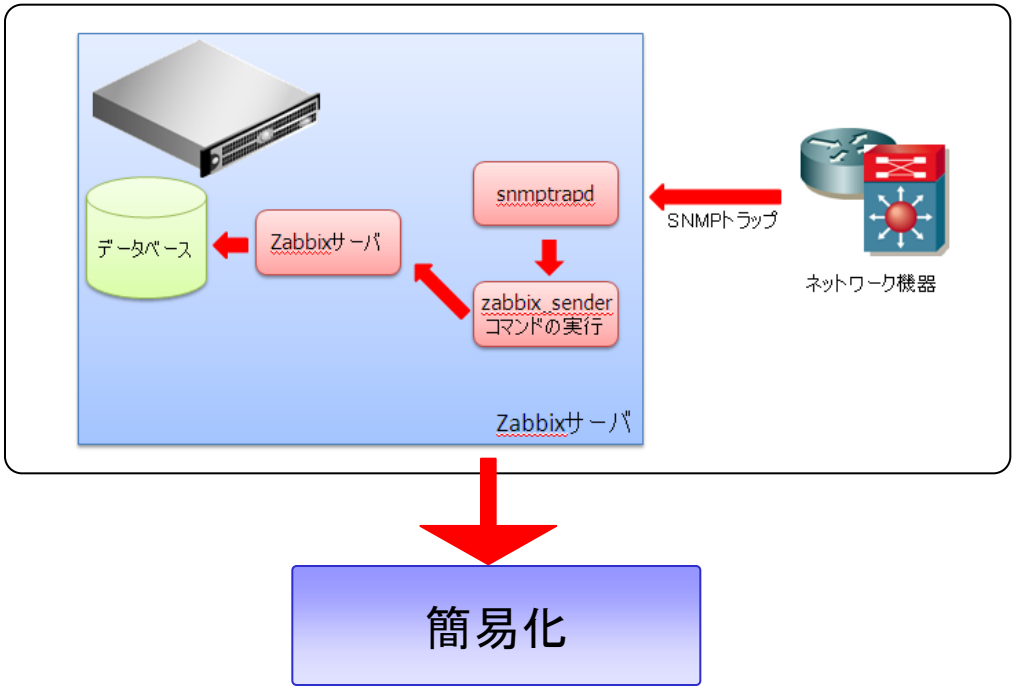
ハードウェア構成情報の自動収集

- OS、MACアドレス、CPUアーキテクチャなどを収集し、インベントリ情報として自動登録可能
- 監視アイテムからインベントリへの取込みが設定可能



SNMP Trap機能の強化

- 煩雑であったSNMP Trap監視設定を容易に設定可能
- snmptrap[正規表現]、snmptrap.fallback というキーが新規追加



Pacemakerに対応させる

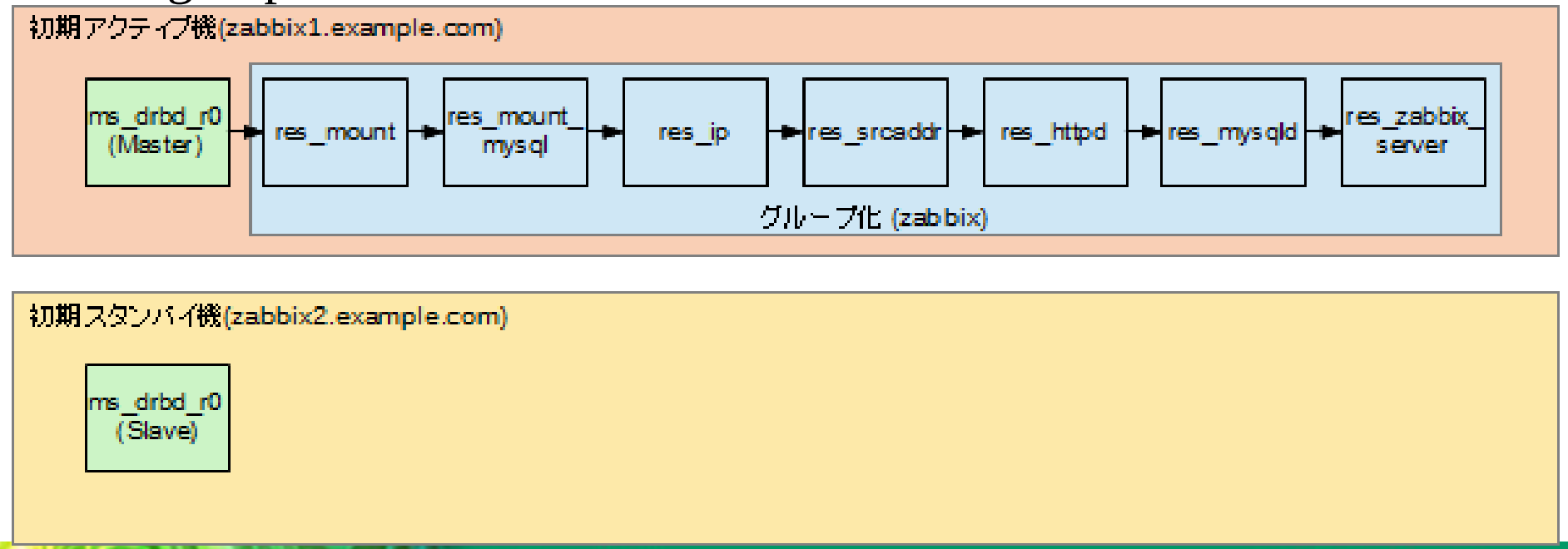
- アプリケーションのクラスタ対応実装方法
- 注意点

アプリケーションのクラスタ対応実装方法(1)

- アプリケーションをリソースとして実装する
  - データの引き継ぎに対応する(drbd Master/Slaveリソース)
  - ファイルシステムのマウント(Filesystemリソース)
  - IP アドレスの引き継ぎに対応する(IPaddr2/IPsrcaddrリソース)
    - IPaddr2:リスン用/IPsrcaddr:Zabbixソースアドレス指定用
  - Apache起動終了に対応する(apacheリソース)
  - Mysql起動終了に対応する(mysqlリソース)
  - Zabbix起動終了に対応する(zabbixリソース)

アプリケーションのクラスタ対応実装方法(2)

- リソースをまとめてグループ、location、orderを定義
- colocaton:同居制御
- location:配置先ノードの制御
- order:起動・終了制御
- gruopはcolocaton +order



詳細構築手順

- 下記資料をご参照ください
- DRBD、Heartbeat、Pacemaker によるZabbix サーバの HA クラスタ構築
  - http://www.miraclelinux.com/jp/online-service/download/docs-products-service/zabbix_and_linux-ha
  - http://www.3ware.co.jp/download.html

DRBDの設定

- metadisk領域は専用パーティションの確保をおすすめ(後述)
- **/etc/drbd.d/r0.res**

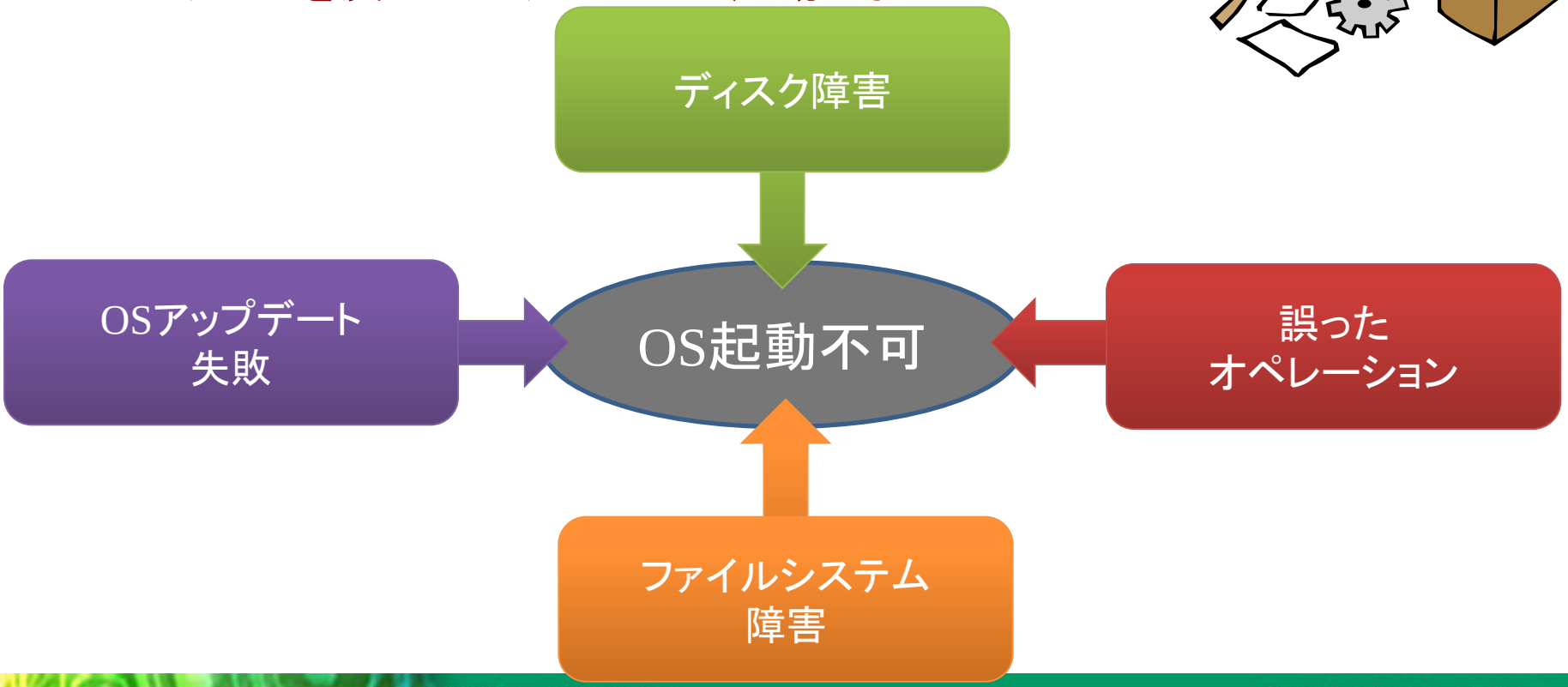
```
        on zabbix1.example.com {
                address 172.16.2.1:7788;
                meta-disk /dev/sda5 [0];
                }
        on zabbix2.example.com {
                address 172.16.2.2:7788;
                meta-disk /dev/sda5 [0];
        }
}
```

バックアップについて

- システムバックアップの必要性
- 障害から復旧までの流れ
- OSの復旧

システムバックアップの必要性

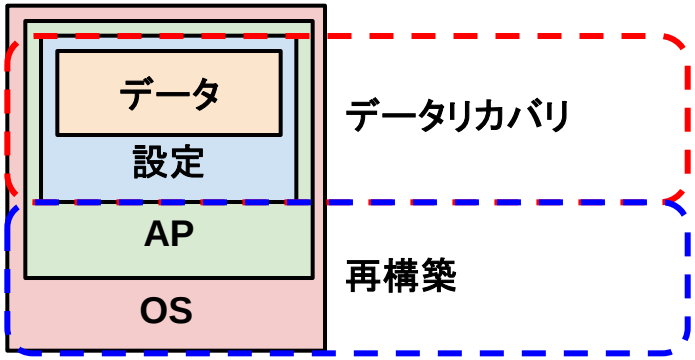
- データのバックアップだけで十分？
  - RAIDを組んでいてもデータ障害は防ぎきれない
  - ファイルがひとつ壊れただけでもOSは起動しなくなる
  - ファイルを戻しただけではOSは起動しない



そのシステムは大丈夫？

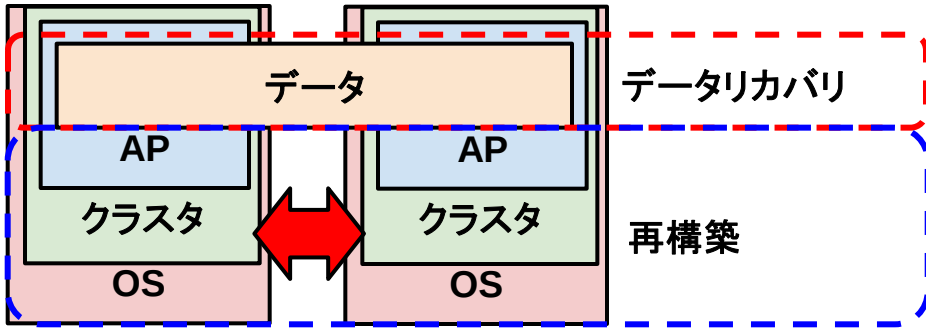
バックアップしてるから大丈夫？

- OSをインストールしただけではデータ復旧できない
- OSの設定項目は膨大にある
- どこまでバージョンアップしていたか知るのは困難
- 最新でない状態までパッチを当てるのは困難

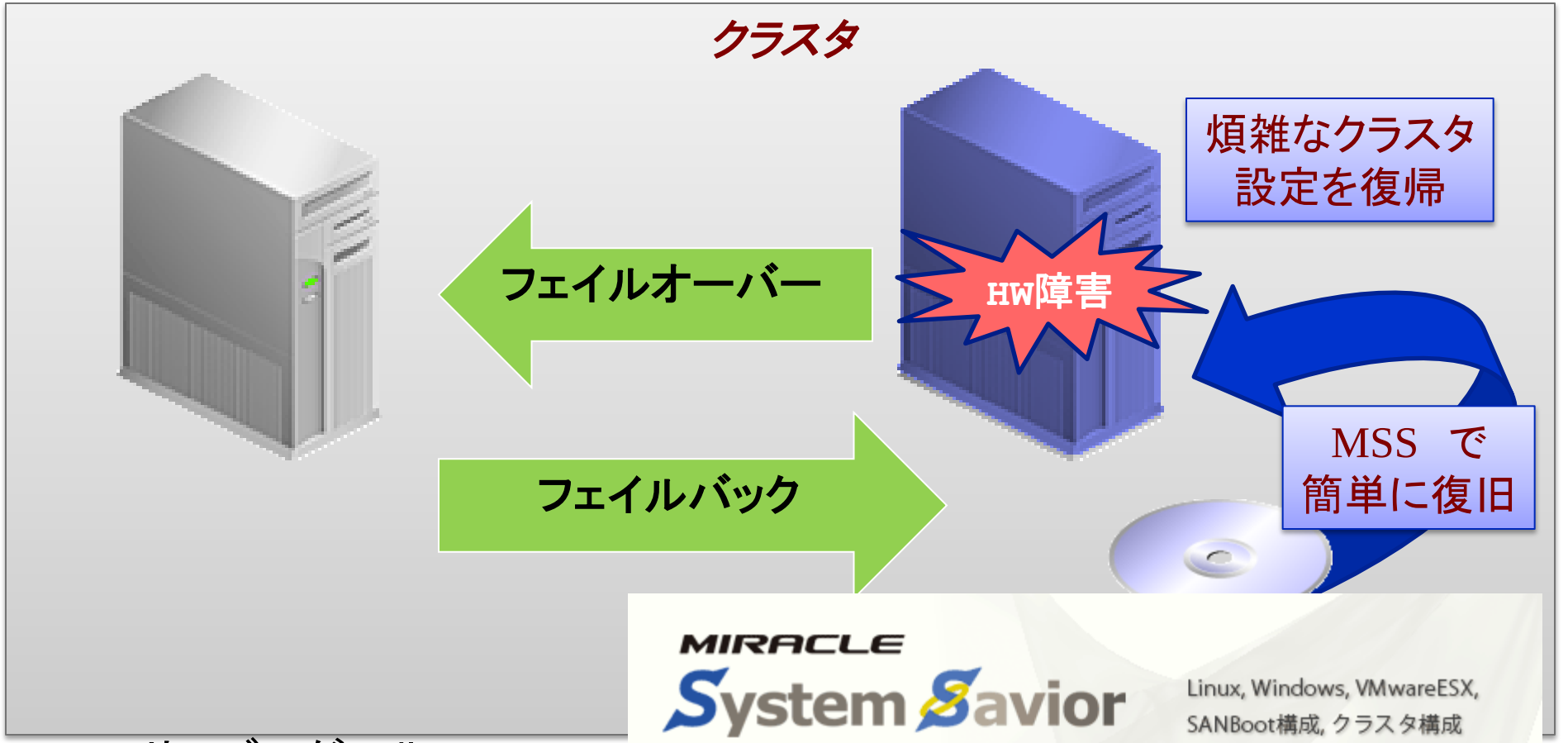


クラスタリングしてるから大丈夫？

- 縮退運転中に障害が発生してもフェイルオーバできない
- クラスタ間で同一設定にしておかないと問題が出る
- 障害試験、フェイルオーバ試験が十分にできない
- クラスタ間でパッチの適用状態合わせる必要がある



システムリストア(イメージバックアップツール MIRACLE System Savior等)



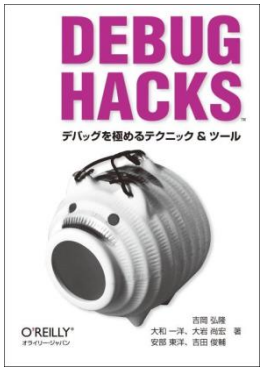
1. サービスがFailover
2. MSSによるシステム復旧
3. 適切なタイミングで元のサーバへフェイルバック

DRBD領域をイメージバックアップする時の注意

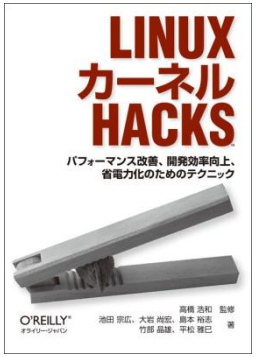
DRBDでミラーディスクを構築する場合、metadiskを指定する。
metadiskの指定方法は2通り

- 専用パーティション
  - 専用にmetadisk用のパーティションを作成(複数のミラーディスクでインデックス番号[0],[1]等を使用して共用可能)
  - MSS使用時はmetadiskに専用パーティションの割り当てを推奨
    - 例:meta-disk /dev/sda7 [0];
  - MSSは専用パーティションに作られたmetadiskをディスクイメージでバックアップ/リストア
  - リストア後、そのまま運用可能
- Internal
  - データパーティションの末尾を使用(その分fs用のサイズが減る)
  - MSSはファイルシステム部分のみを認識してバックアップ/リストア
  - リストア後、メタデータの再作成が必要

エンジニア/インターン募集中



- MLの製品一覧と関係するOSS
  - Asianux(Linux Kernel/gcc/gdb/Python等)
  - MIRACLESystemSavior (CloneZilla等)
  - ML ZBX(Zabbix/Apache/MySQL/Cassandra等)
  - MVS(X11/GTK+/gstreamer/WebKit等)
- 開発、パッチ作成、パッケージメンテナンス、サポート
- 興味ある製品やOSSがあれば、担当している社員から話を聞けます。
  - インターン受付中、興味あれば声かけてください
  - または:https://www.miraclelinux.com/jp/company/recruit



参考資料

- DRBD、Heartbeat、Pacemaker によるZabbix サーバの HA クラスタ構築
  - http://www.miraclelinux.com/jp/online-service/download/docs-products-service/zabbix_and_linux-ha
  - http://www.3ware.co.jp/download.html
- DRBDユーザーズガイド
  - http://www.drbd.jp/users-guide/users-guide.html
- **Linux-HAユーザーズガイド**
  - http://linux-ha.sourceforge.jp/wp/manual/linux-haユーザーズガイド
- NoSQL を使ったZabbixの高速化
  - http://www.miraclelinux.com/jp/online-service/labs/lab01
- 今回の資料は抜粋となりますので、詳細は上記をご参照ください。