エクサバリューフォーラム25
A-6

オープンソースで
高可用性DBクラスターを構築する

2012年7月13日

開発統括本部 技術推進室
谷　文秀

Agenda

1. 講演者のプロフィール
2. 本講演内容の概要
3. Webシステムの構成
4. DBクラスターの種類
5. 検証の目的と前提
6. 機能設計
   - ① クラスター
   - ② DBデータ領域のバックアップ
   - ③ DBデータ領域のリストア
   - ④ DBのロールフォワードリカバリ
   - ⑤ リソースと障害の監視
7. 構成設計
   - ① 論理構成
   - ② ミドルウェアとハードウェア
   - ③ ネットワーク
8. 構築と検証
   - ① 構築作業
   - ② テストシナリオとテスト結果
   - ③ 見つかった不具合と解決策
9. 検証結果とその評価
10. OSS製品を使う際の注意点
11. おわりに

1. 講演者のプロフィール

谷　文秀（シニアITアーキテクト）

所属　　開発統括本部　技術推進室

経歴　　1984～ ハード/ソフト製品の開発･販売･サポート

Unixミニコンの開発･製造･販売
Mac, Unixの日本語化（日本語入力機能、フォント）
Unix WS用ドキュメント作成ソフト(DTP)の開発･販売･サポート
米国製ドキュメント管理ソフトの販売･サポート
米国製Javaアプリケーションサーバー製品の販売･サポート

1998～ Webシステムのアプリ開発･基盤構築･運用

大手家電メーカー 資材調達EDIシステムの基盤構築･アプリ開発･運用
大手旅行ポータルサイト ホテル･旅館予約システムの基盤構築
大手旅行会社　福利厚生施設利用精算システムの基盤構築
大手テーマパーク 保全管理･コスチューム管理システムの基盤構築
大手航空会社 営業分析DWHの基盤構築

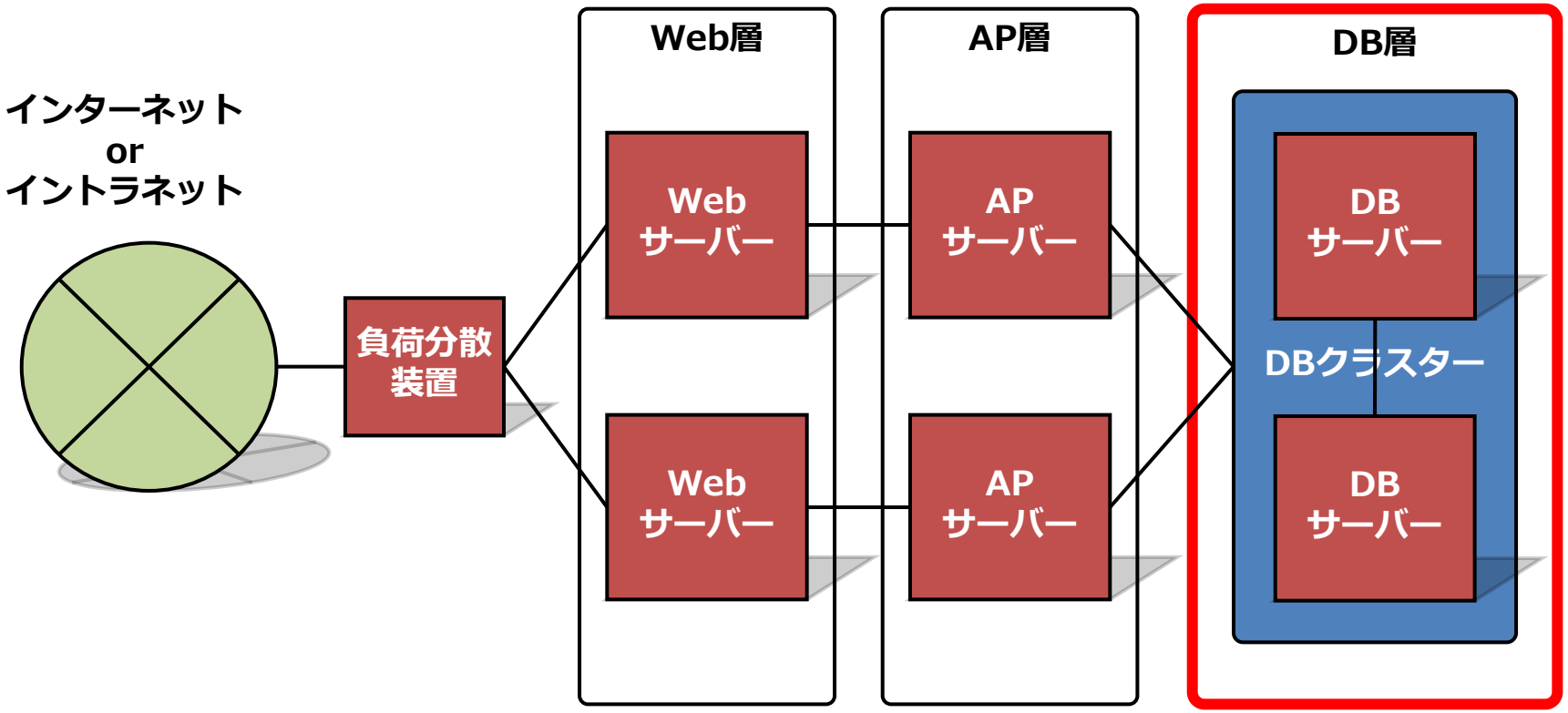
2. 本講演内容の概要

本講演内容は第50回IBMユーザー・シンポジウム発表論文の内容を要約したものである。

- 近年、Webシステムを構築する際にOSS製品のApacheやTomcatを使うケースはよく目にする。しかし、Webシステムのバックエンドに位置するDBサーバーのクラスターをOSS製品で組んでいる事例は決して多くはない。
- OSS製品は無償で使える利点があるものの、技術革新が早く、開発元のサポートもなく、OSS製品には、期待通りに動くかどうかは実際にやってみないと分からないというリスクがある。
- 本講演では、商用のWebサイトで使うことを前提に、実際にDBサーバーのクラスターをOSS製品で組んで、機能性の検証を行った事例を紹介する。構築のノウハウも併せて紹介するので、OSS製品を使ってみようという方々のお役に立てれば幸いである。

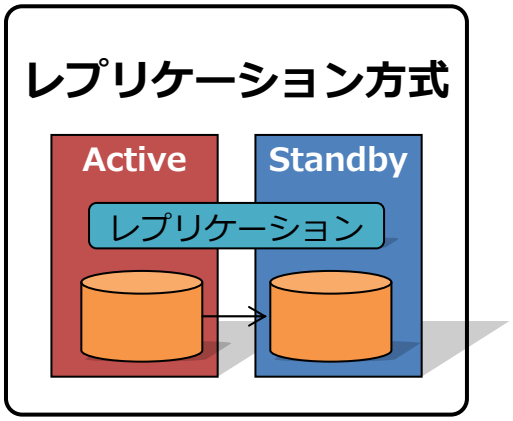
3. Webシステムの構成

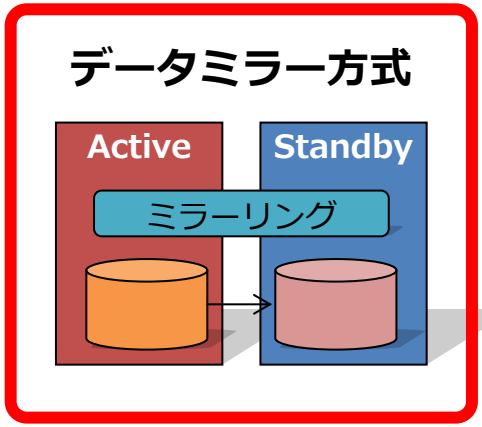
通常Webシステムは以下の3階層モデルで構成される。今回の検証はDB層のDBクラスターが対象である。



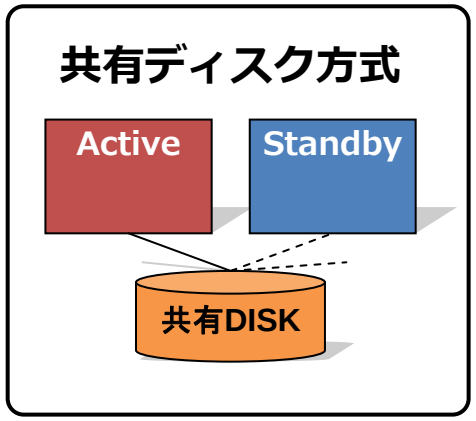
4. DBクラスターの種類

DBクラスターには大きく分けて下図のような方式がある。今回はOSS製品で実績の多いデータミラー方式で検証を行っている。

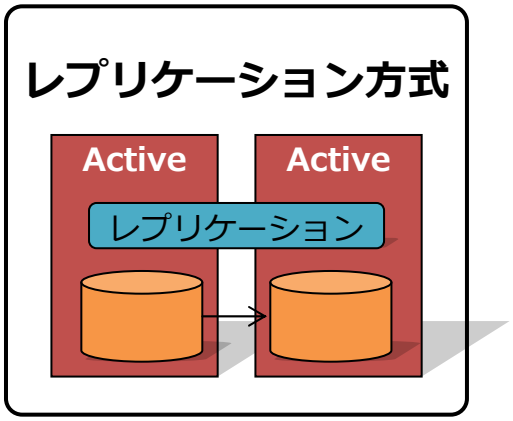


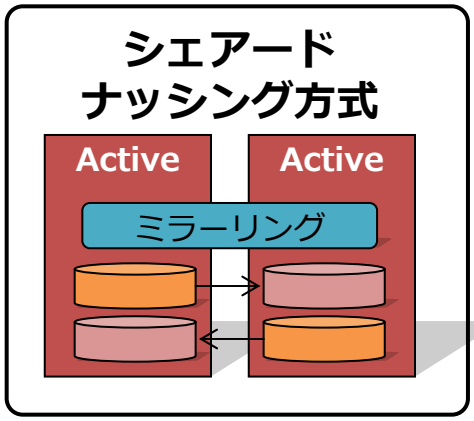


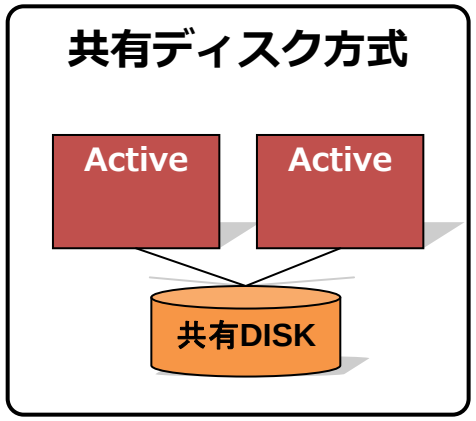
DB2 V8以前, MySQL Cluster



Oracle RAC, IBM DB2 pureScale







5. 検証の目的と前提

OSS製品だけで構築したDBクラスターが、24時間365日サービスを提供するECサイトなど、高い可用性が求められる商用Webシステムに適用できるかどうか、実際に組んでみて実用性を確かめるのが今回の検証の目的である。

＜検証の前提＞

- **すべてOSS製品**だけでDBクラスターを構築する。
- **商用製品と同等レベルのクラスター機能**を実装する。
- 検証に使うDBのデータ量は通販サイト想定で700GBとする。
- DBのバックアップは**オンラインバックアップ**とする。
- **商用製品と同等レベルの監視機能**を実装する。

6. 機能設計　①クラスター

＜設計方針＞

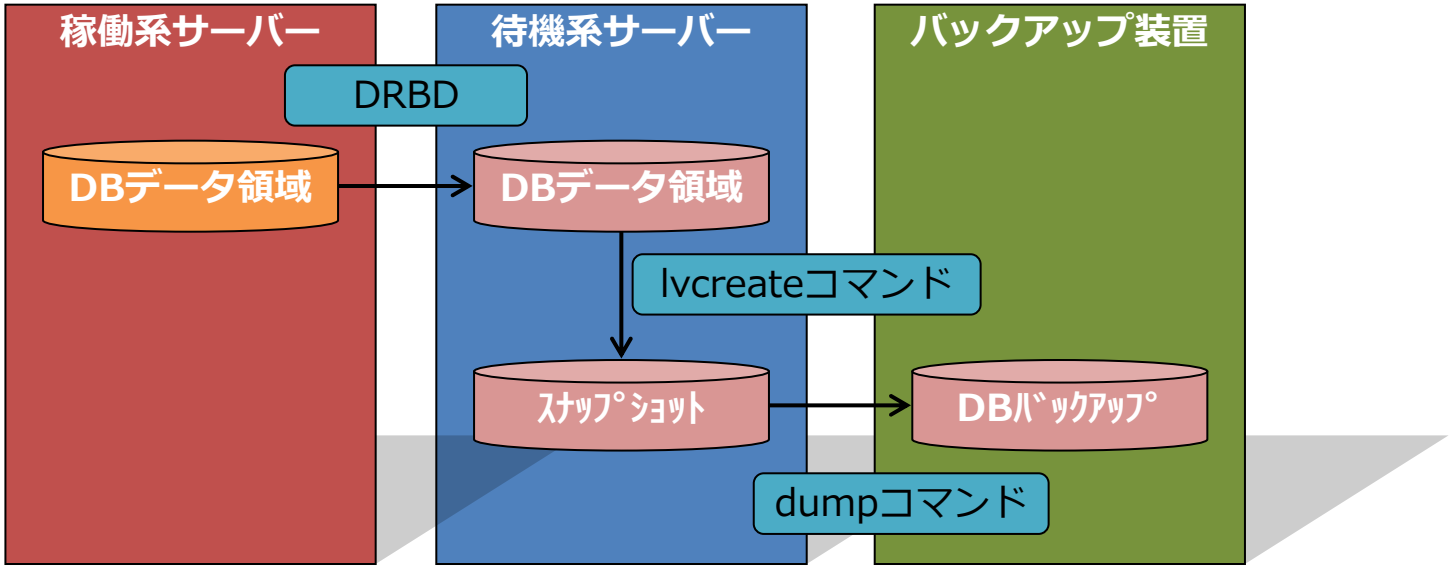
- クラスター方式はOSS製品で実績の多い**データミラー方式**とする。
- OSS製品は**Heartbeat V2**と**DRBD V8**で構成する。
- 商用製品を使った場合と同等レベルのクラスター機能を実現する。

No	稼働系サーバーにおける障害	障害検知時の動作
1	ノードダウン	待機系サーバーにフェイルオーバーさせる
2	サービスLANの通信断	同上
3	ハートビートLANの通信断	稼働系サーバーでサービスを継続させる
4	ミラーリングLANの通信断	同上
5	MySQLのプロセスダウン	待機系サーバーにフェイルオーバーさせる
6	Heartbeatのプロセスダウン	Heartbeatを再起動する

6. 機能設計　②DBデータ領域のバックアップ

＜設計方針＞

- 稼働系サーバーに負荷をかけないよう、**待機系サーバー上でオンラインバックアップを行う**。Linux標準コマンドを使用する（下図）
- バックアップを実行するシェルスクリプトを用意し、Linux標準のcronで定時に自動実行する。
- 日曜日にフルバックアップ、それ以外の曜日は差分バックアップとする。（dumpコマンドのオプションで指定）



6. 機能設計　②DBデータ領域のバックアップ

- DBデータ領域のバックアップ方法（下表）
  （DBの一貫性を保つため**更新ロック**をかける点がポイント）

No	コマンド（すべて待機系サーバーで実行）	説明
1	mysql -u db_flush -h XX.XX.XX.XX << EOF **flush tables with read lock;** quit EOF	更新ロックをかけ、テーブルをフラッシュする
2	lvcreate --snapshot --size=10G --name Snap /dev/VolGroup00/lv_mysql	スナップショットを作成する
3	mysql -u db_flush -h XX.XX.XX.XX << EOF > ./binlog-position.log flush logs; show master status; **unlock tables;** quit EOF	バイナリログをスイッチ バイナリログのファイル名を記録 更新ロックを解除する
6	mount -o ro /dev/VolGroup0/Snap /Snap	/Snapをマウント
7	dump -0uf /mnt/mysql_bak/DBFull_$TODAY.dump /Snap dump -1uf /mnt/mysql_bak/DB_$TODAY.dump /Snap	スナップショットからフルダンプまたは差分ダンプを実行する
8	umount /Snap	/Snapをアンマウント
9	lvremove -f /dev/VolGroup00/Snap	スナップショット/Snapを削除

6. 機能設計　③DBデータ領域のリストア

＜設計方針＞

- Linux標準の**restoreコマンド**を使用する。
- リストアは、稼働系・待機系サーバーともに、MySQL, Heartbeat, RDBDをすべて停止した状態で行う。
- DBデータ領域のリストア方法（下表）

No	コマンド（すべて待機系サーバーで実行）	説明
1	mount /mysql	/mysqlをマウント
2	cp mysql-bin.* /backup	バイナリログを/mysql以外に退避する
3	rm -rf /mysql/*	/mysqlの中身を削除
4	restore -rvf DBFull_XXXXXXXX.dump restore -rvf DB_YYYYYYYY.dump	バックアップからDBをリストアする
5	rm -f restoresymtable	restoresymtableファイルを削除

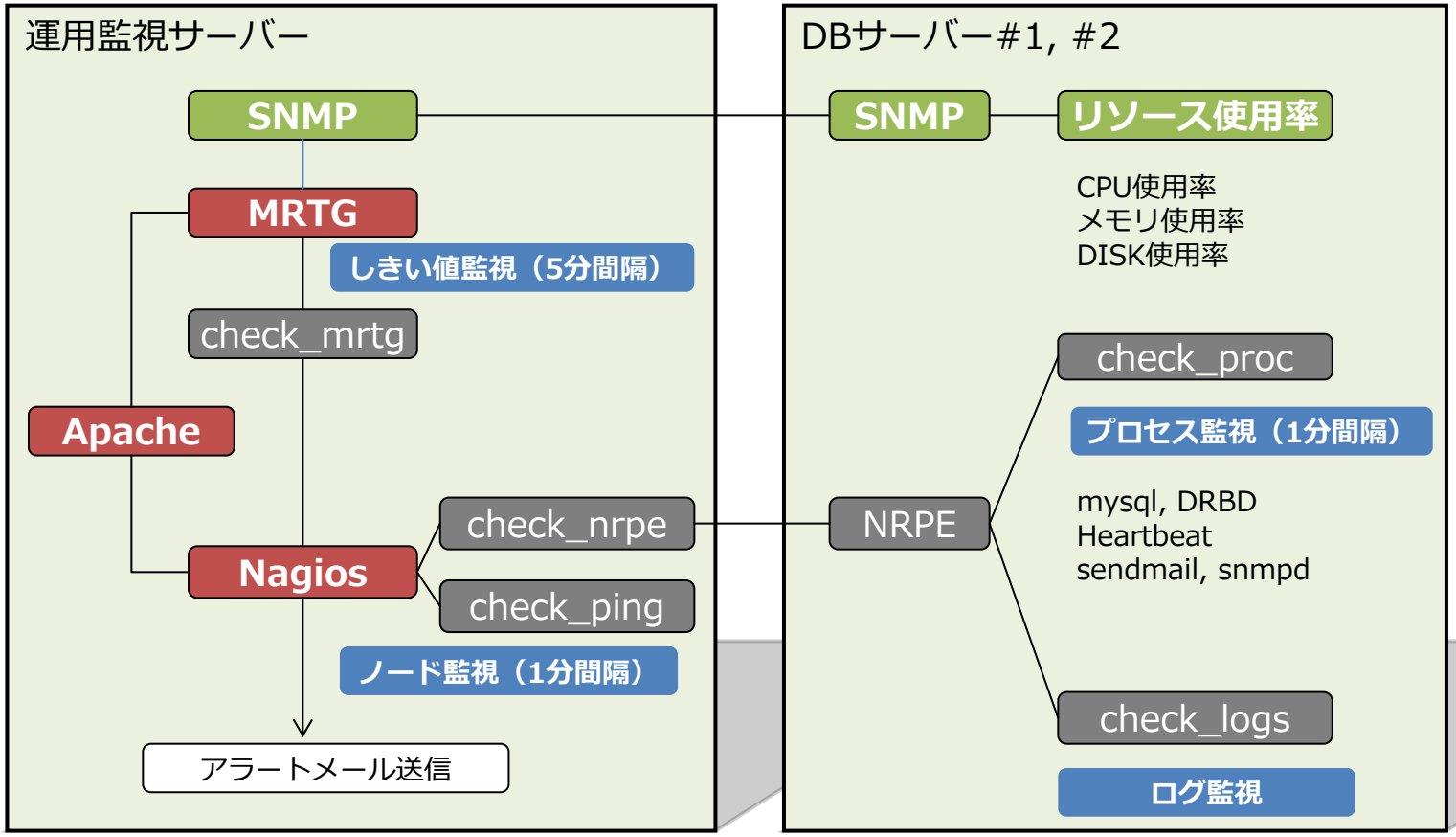
6. 機能設計　④DBのロールフォワードリカバリ

- MySQLの**バイナリログ**を使って**ロールフォワードリカバリ**を行う。
- DBのロールフォワードリカバリ方法（下表）

No	コマンド（15,16,19以外は待機系サーバーで実行）	説明
7	vi my.cnf	バイナリログ出力を無効化
8	cp /backup/mysql-bin.* .	退避したバイナリログをもとに戻す
9	cat binlog-position.log	バイナリログのファイル名を確認
10	mysqlbinlog ･･･ > recover.sql	ロールフォワード用SQLを作成する
11	mysql ･･･ < recover.sql	ロールフォワードを実行する
12	umount /mysql	/mysqlをアンマウント
13	vi my.cnf	バイナリログ出力を有効化
14	service drbd start	待機系サーバーでDRBDを起動する
15	drbdadm invalidate r0 ※稼働系サーバーで実行	稼働系サーバーのDRBDデータ領域を無効化する
16	service drbd start ※稼働系サーバーで実行	稼働系サーバーのDRBDを起動する（再同期開始）
17	service heartbeat start　※サービス再開	待機系サーバーのHeartbeatを起動する
18	cat /proc/drbd	同期完了を確認する
19	service heartbeat start ※稼働系サーバーで実行	稼働系サーバーのHeartbeatを起動する
20	hb_standby	稼働系サーバーにフェイルバックする

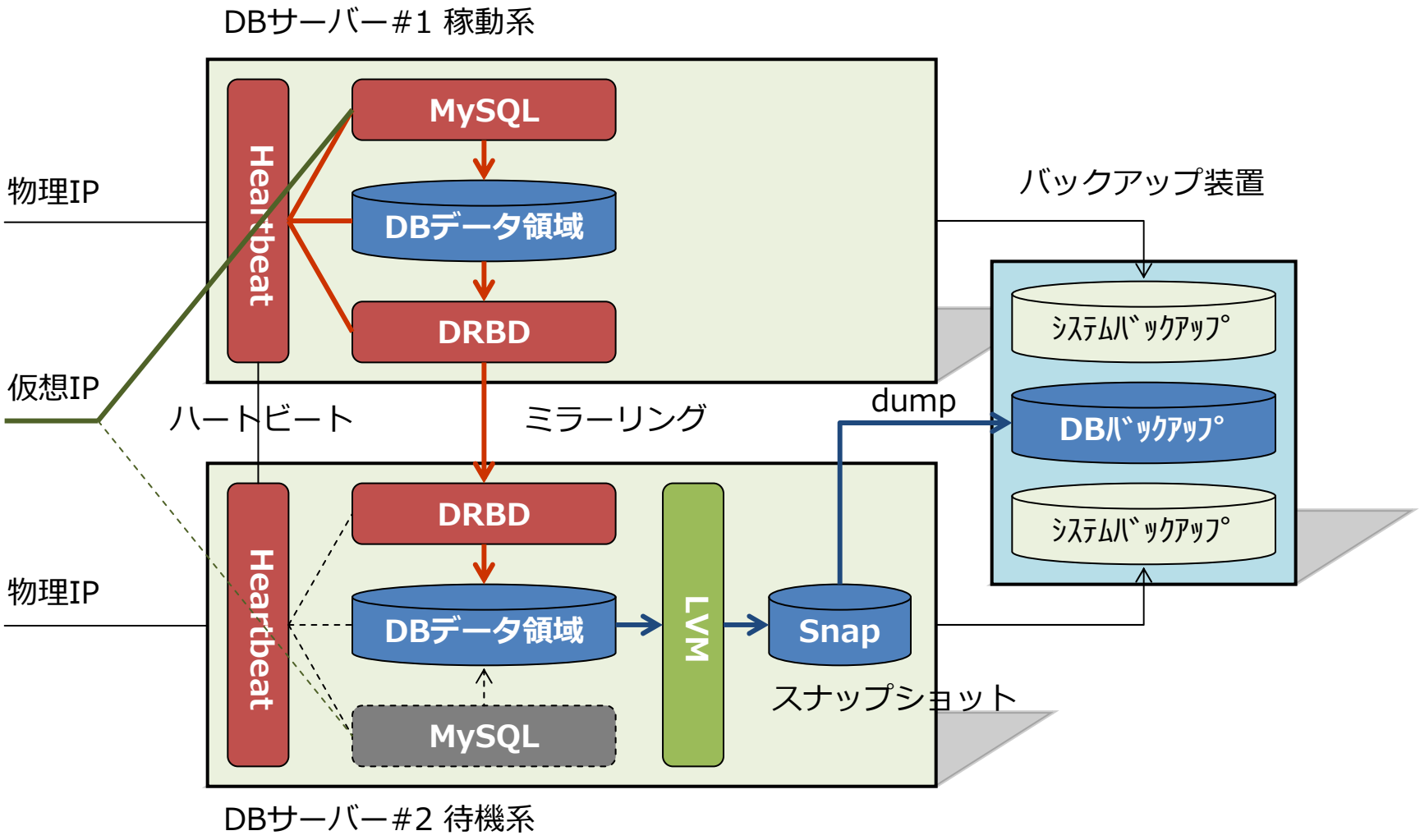
＜設計方針＞

- DBクラスターとは別に**運用監視サーバー**を立てて監視する。
- OSS製品は**Nagios**と**MRTG**で構成する。



7. 構成設計　①論理構成

データミラー方式によるDBクラスターの構成を以下に示す。



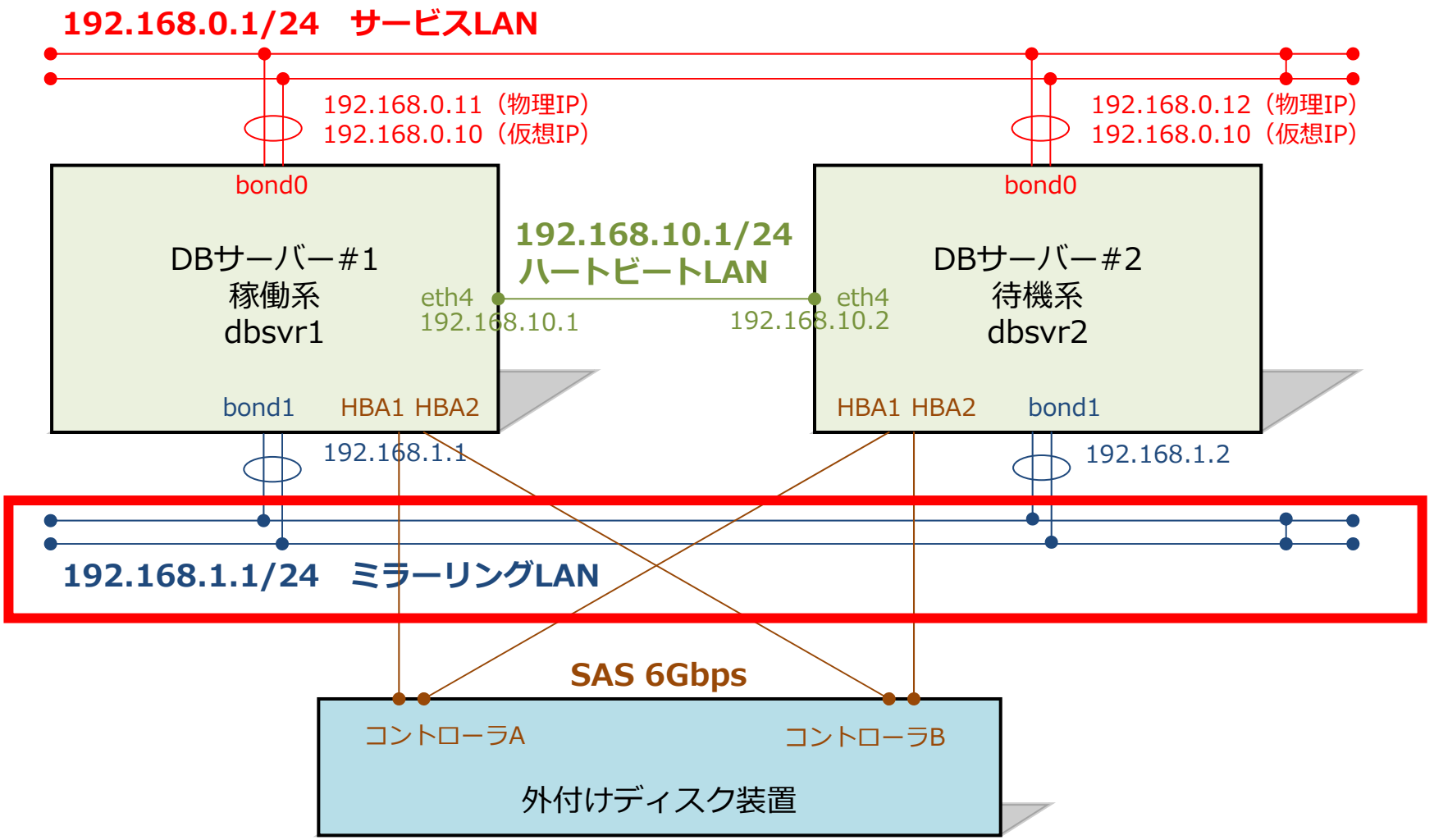
7. 構成設計　②ミドルウェアとハードウェア

ソフトウェア	プロダクト	バージョン
OS	CentOS	5.6 (Final)
データベースソフト	MySQL	5.5.15
クラスターソフト	Heartbeat	2.1.4-1
ミラーリングソフト	DRBD	8.3.8.1
リソース監視ソフト	MRTG	2.14.5
障害監視ソフト	Nagios	3.2.3

検証ではバックアップ装置として外付けのディスク装置を使っているが、安価なテープ装置で置き換え可能である。

機器	スペック	
DBサーバー ×2台	本体装置	IBM System x3650 M3
	CPU	Intel® Xeon X5650 2.66GHz x1
	メモリー	12GB
	内蔵HDD	SAS 600GB x6（RAID10） ＋ホットスペア1本
	LANインターフェース	1Gbitイーサーネット 6ポート
	拡張カード	シングルポート 6Gbps SAS HBA x2
外付けディスク装置 ×1台	本体装置	IBM System Storage DS3524
	コントローラー	デュアル･コントローラー構成
	キャッシュ	4GB
	ホストインターフェース	6Gbps SASインターフェース 4ポート
	HDD	SAS 300GB x12（RAID10） ＋ホットスペア1本

DRBD用にミラーリング専用のLANを構成している。



8. 構築と検証　①構築作業

約2ヶ月をかけて、若手SE2名とともに構築ならびに検証作業を実施した。

- 2名ともに今回使用したOSS製品を使った経験はない。
- 構築作業は、書籍やWebで公開されている事例や製品ドキュメントをもとに行った。
- 製品に対する知識不足から試行錯誤が多く、時には１つの不具合の解決に1週間以上を費やした。

8. 構築と検証　②テストシナリオとテスト結果　1/2

- 商用製品でDBクラスターを構築した場合とほぼ同じ内容のテストシナリオを用意しテストを実施した。（単体および結合テスト）

カテゴリ	テストシナリオ	期待される動作	テスト結果
起動･停止 クラスター動作 （正常系）	サーバーの起動	正常に起動	OK
	クラスターサービスの起動	正常に起動	NG→再テストでOK
	クラスターサービスの停止	正常に停止	OK
	サーバーのシャットダウン	正常にシャットダウン	OK
	クラスターの手動フェイルオーバー	正常にフェイルオーバー	OK
	クラスターの手動フェイルバック	正常にフェイルバック	OK
クラスター動作 （異常系）	稼働系OSのダウン	待機系にフェイルオーバー	OK
	待機系OSのダウン	稼働系でサービス継続	OK
	Heartbeatプロセスのダウン	稼働系でサービス継続 プロセスの再起動	OK
	稼働系MySQLプロセスのダウン	待機系にフェイルオーバー	OK
	稼働系サービスLANの通信断	待機系にフェイルオーバー	OK
	稼働系ハートビートLANの通信断	稼働系でサービス継続	OK
	稼働系ミラーリングLANの通信断	稼働系でサービス継続	OK

8. 構築と検証　②テストシナリオとテスト結果　2/2

カテゴリ	テストシナリオ	期待される動作	テスト結果
システム監視	ハードウェア障害の検知（IMM）	管理者にメールで通知	NG→再テストでOK
	CPU使用率のしきい値越え	管理者にメールで通知	OK
	メモリー使用率のしきい値越え	管理者にメールで通知	OK
	ディスク使用率のしきい値越え	管理者にメールで通知	OK
	対向ノードのPING無応答	管理者にメールで通知	OK
	DRBDプロセスのダウン	管理者にメールで通知	OK
	MySQLプロセスのダウン	管理者にメールで通知	OK
	Heartbeatプロセスのダウン	管理者にメールで通知	OK
	sendmailプロセスのダウン	管理者にメールで通知	OK
	snmpdプロセスのダウン	管理者にメールで通知	OK
バックアップとリカバリ	システム領域のバックアップ	正常にdumpファイルを作成	OK
	システム領域のリストア	正常にシステム領域を復元	OK
	DBのフルバックアップ	正常にdumpファイルを作成	OK
	DBの差分バックアップ	正常にdumpファイルを作成	OK
	DBのフルリストア	正常にDBを復元	NG→再テストでOK
	DBの部分リストア（特定テーブルのみのリストア）	正常に特定テーブルを復元	NG→再テストでOK
	DBのロールフォワード・リカバリ	正常にDBをリカバリ	OK
ログ運用	ログローテーション	正常にローテーション	OK

8. 構築と検証　③見つかった不具合と解決策

不具合が発生した原因は、製品のバグや不具合ではなく、使ったOSS製品に対する理解不足と結論付けられる。

	見つかった不具合	原因と解決策
1	クラスターサービスの起動に失敗する	Heartbeat V2.1とDRBD V8.3のバージョンの組み合わせが適切でなかったのが原因。 Heartbeat V2.1をV1互換モードで動かすことで解決。
2	IMMのメール送信に失敗する	IMMのメール送信オプションの選択ミスでメール本文が大きくなりすぎて受信メールサーバーのサイズ制限に引っ掛かってしまったのが原因。 オプションの選択項目を見直すことで解決。
3	DBのリカバリ時にクラスターのフェイルバックが失敗する	リカバリ手順の誤りでDRBDのスプリットブレイン抑止機能によりプライマリ側のDBデータ領域が読み取り専用ファイルシステムとなってしまったのが原因。 リカバリ手順を見直すことで解決。
4	DBの部分リストアに失敗する（特定テーブルの.ibdファイルだけバックアップから戻す）	MySQLのクラッシュリカバリ機能によりテーブルスペースIDが書き変わってしまったのが原因。 クラッシュリカバリを発生させないようにすることで解決。※1

※1　異なるテーブルスペースIDのテーブルをインポートする方法は見つけたが、
手順が非常に複雑であり、クラッシュリカバリを発生させないことが極めて重要である。

9. 検証結果とその評価

商用製品と比べてそん色のない機能性を確認できた。24時間サービスを提供する商用Webサイトでも十分に使えると判断する。

機能	検証結果	評価
クラスター機能	Heartbeat V2.1.4のV1互換モードで問題なく機能した。 互換モードでは不足するプロセス監視機能は簡単なシェルスクリプト（check_mysql，check_hb）を用意することで補えた。	設定はすべて定義ファイルで行うが、設定は非常に簡単である。 商用製品と比べてそん色のないクラスター機能が実現できていると評価する。
バックアップとリカバリ機能	スナップショット取得時に数秒から10秒前後の更新ロックが発生する。 700GBのバックアップ/リストアにそれぞれ約2時間（98MB/秒）。 ロールフォワード･リカバリに2～3時間。サービス再開後、1.4TBのDRBD領域の再同期に約4時間かかることがわかった。	スナップショット作成時に更新ロックをかけるため、バックアップを実行する時間帯に注意する必要がある。 今回はSAS接続の外部ストレージをバックアップ装置として使ったが、ロールフォワード･リカバリ時にはサービス再開までおおよそ4～5時間はかかる。商用製品のバックアップソフトを使っても同じぐらいの時間はかかる。
リソースと障害の監視機能	監視機能は設計通り、問題なく機能した。	多くのプラグインの設定をすべて定義ファイルで行なわなければならず、GUIベースの商用製品のほうが設定は楽である。

10. OSS製品を使う際の注意点

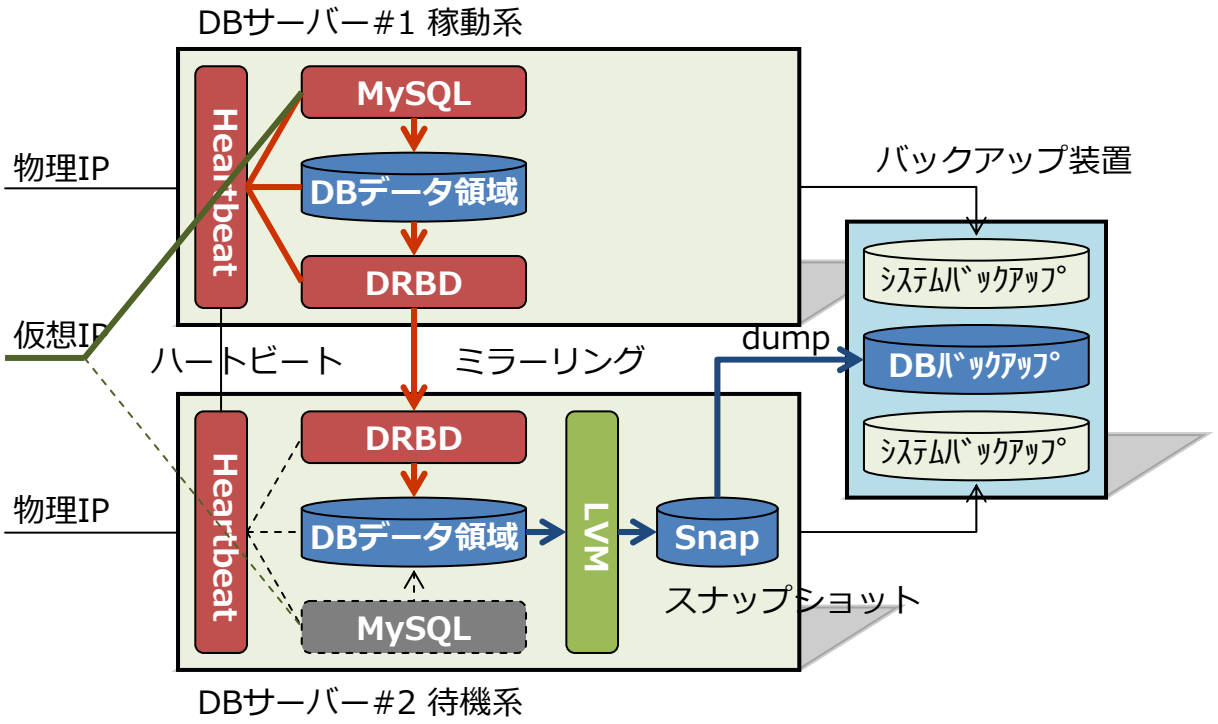
事例が豊富なOSS製品を選ぶ。事例を探すときは製品のバージョンに注意。英語のドキュメントを読む能力も時には必要である。

- ✓ 事例が多いことは利用者が多いことの裏付け。**事例の多いOSS製品を選ぶ**ことでスムーズに導入することができる。（不具合の対処方法も共有されていることが多い）
- ✓ インストール方法や設定に関する事例は製品のバージョンが大きく異なると役に立たない。使おうとしているのと**同じバージョンの事例を探すこと。**
- ✓ 動作保証のある製品の組み合わせがはっきりしないことが多いので、**事前にネット等で動作実績のある製品の組み合わせを調べる。**
- ✓ どうしても不安なときは、 有償サポートを提供しているOSSディストリビューターから製品を購入するのも選択肢の１つ。
- ✓ 商用製品でも同様に、一旦トラブルが発生すると解決にそれなりの時間がかかるので、構築に際しては時間的余裕を持つ。

11. おわりに

- DRBDを使ったDBのバックアップの取りかたは今回かなりこだわって設計したつもりである。
- OSS製品は、**商用製品とそん色のない機能を無償で使えることが最大の魅力である。**
- 試算してみたところ、今回と同様のDBクラスターをすべて商用製品で組むと、初期費用と5年間のランニングコストを合わせて最低でも750万円ぐらいになる。リスクをうまく回避すれば、**OSS製品を利用することでの、費用削減、あるいは別の目的にこの費用を使えることのメリットは大きい。**
- 今回の構築であらためて実感したのは、**OSS製品の利用で一番役に立つのはネット等で公開されている事例である**ということ。事例が豊富に見つかればOSS製品を使うリスクは小さくなる。
- もしみなさんがOSS製品でシステムを構築する機会があったら、ぜひ事例を公開してもらえれば幸いである。

ご清聴ありがとうございました



本文中の会社名・製品名・サービスネームについて

- IBMロゴは、世界の多くの国で登録されたInternational Business Machines Corp.の商標です。
- その他すべての会社名・製品名・サービスネームは、それぞれ各社の商標または登録商標もしくはサービスマークです。