シングルサインオン導入プロジェクトテンプレート
方式設計書(冗長構成版)

Ver 1.0.1 （2013/09/02）

（株）野村総合研究所
オープンソースソリューション推進室

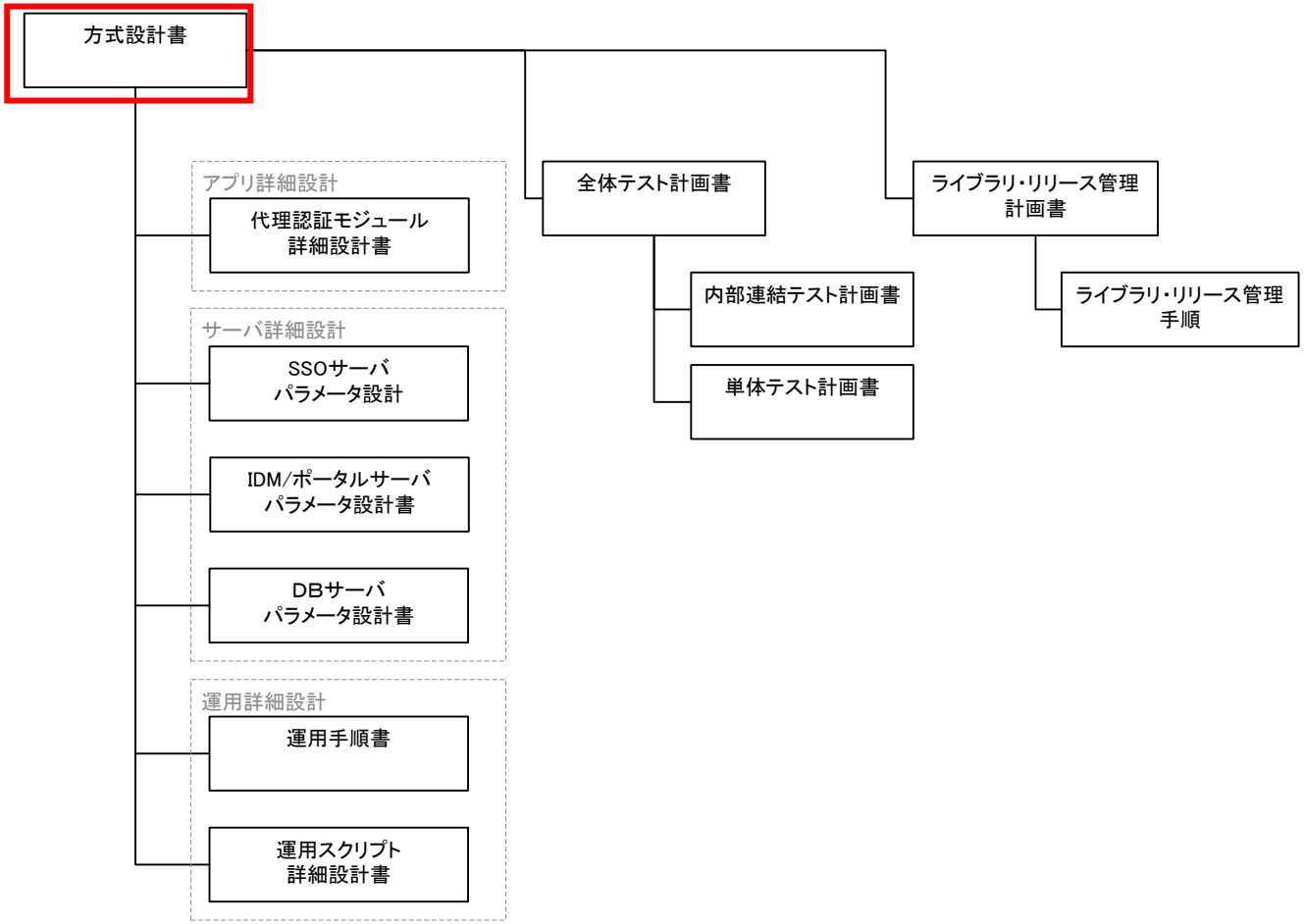
変更履歴

版	日付	担当者	変更内容	備考
1.0	2013/03/22	渡部	初版作成	
1.0.1	2013/09/02	和田	タイトル修正	

目次

1. はじめに

1.1. 本ドキュメントについて



1.2. 本ドキュメントの構成

2章にてシステムの要件を定義し、3章以降で要件を満たすシステムの実現方式を記載する。

システムの要件と実現方式記載箇所の対応一覧

要件			実現方式
機能要件(2.6章)		→	処理方式(**エラー! 参照元が見つかりません。** 章)
非機能要件(2.7章)	運用要件	→	運用設計(**エラー! 参照元が見つかりません。** 章)
	セキュリティ要件	→	セキュリティ設計(6章)
	耐障害性要件	→	耐障害設計(**エラー! 参照元が見つかりません。** 章)
	性能要件	→	性能設計(サイジング)(8章)
	拡張要件	→	拡張設計(9章)
	移行要件	→	移行設計(10章)
	端末要件	→	端末設計(11章)
	維持管理・サポート要件	→	維持管理・サポート方針(12章)
	開発要件	→	開発方針(13章)

2. 要件定義

2.1. プロジェクトの目的／方針

- AシステムおよびBシステムのシングルサインオンを実現する。
- 管理者が、ID管理を行うことができる Web システムを構築する。

2.2. スコープ

- 連携先システムは対象外

2.3. 前提

- (特になし)

2.4. 制約事項

- ActiveX は対応不可
- ログイン画面のリクエストに対して、HTTP 301 not modified などの body を含まないリクエストを返却するシステムへのSSO連携は対応不可

- バックアップ中は OpenLDAP を更新する作業をすると、バックアップが不正になる可能性があります。
- LB は次の機能を有する必要があります。
  - SSL デコード機能（必須）
  - TCP(L4)レベルでの負荷分散機能（必須）
  - HTTP(L7)でのヘルスチェック。HTTP の応答がない場合に NG になるだけでなく、HTTP の応答ステータスコードが異常(503 等)の場合にも NG になる機能が必要。
  - クッキーが同じならば同じサーバにリクエストを振り分ける機能（クッキースティッキネス）（できれば）
  - クッキースティッキネス機能がない場合は、IP アドレススティッキネス機能（必須）
  - 外部からの通信と内部の通信で利用するプールを分ける機能（できれば）

2.5. スケジュール

- 2012 年 12 月末までにリリース完了

2.6. 機能要件

2.6.1. アクター一覧

＜シングル構成と同じ＞

2.6.2. 機能一覧

＜シングル構成と同じ＞

2.7. 非機能要件

2.7.1. 運用要件

項番	大分類	中分類	内容	備考
1	定常運用	データバックアップ	過失によるデータ消失などの事態を防ぐため、バックアップ/リストアできること。バックアップ対象は顧客属性情報と認証先情報とする。 毎日バックアップを取得し、1 週間分バックアップを取得する。	
2		ログ出力	システムの利用状況の把握、障害やその予兆の検知、障害発生時の問題解決のため、各ミドルウェアのログを出力すること。 また、顧客が準備した監視サーバからログ監視ができるようにすること。	
3		ログ保管	ログを 18 カ月分保管すること。	

4		ログ参照	以下のログについて、ポータルから簡単に参照できるようにすること。 ・Web サーバアクセスログ ・認証・認可ログ	
5		稼働統計情報取得	以下の統計情報を取得し、1 週間分保管する事。 ・CPU 使用率 ・メモリ使用率 ・IO 負荷状況 ・ネットワークコネクション数	
6		時刻同期	全てのサーバで時刻をそろえること	
7		サービス時間	24 時間 × 365 日サービスを継続できること。	
8	監視運用	死活監視	顧客が準備した監視サーバから、以下の項目を監視できるようにすること。 ・ICMP,TCP ポート,プロセス,URL	
9		ログ監視	顧客が準備した監視サーバから、障害を検知できるように、監視サーバから以下のログを監視できるようにすること。 ・Apache, Tomcat, OpenAM, MySQL レプリケーションログ, OpenLDAP レプリケーションログ	
10		ジョブ稼働監視	顧客が準備した監視サーバが、cron のジョブの失敗を検知できるように、ジョブの失敗をログに出力し、監視できるようにすること。	
11		リソース監視	顧客が準備したから、CPU、メモリ、ディスク使用量を監視できるようにすること。	
12	障害時運用	プロセス起動・停止	障害時に、手動でプロセスの起動・停止ができるように、手順を準備すること。	
13		MySQL・OpenLDAP データリストア	データ障害時に、バックアップデータからデータを復旧できるように、手順を準備する事。	
14		DB サーバフェイルバック	DB サーバ正系障害が発生して、フェイルオーバした時に、フェイルバックできるように、手順を準備すること。	
15	特別時運用	SSO サーバ・IDM/ポータルサーバ切り離し・切り戻し	SSO サーバ、IDM/ポータルサーバにリリースする際に、片系ずつリリースできるように、ノードの切り離し・切り戻し手順を準備する事。	
16		ソフトウェアバージョンアップ	各ソフトウェアのバージョンアップ手順を準備する事	今回は対象外
17		リリース	コンテンツのリリース等、定型的なリリースの手順を準備する事	今回は対象外

2.7.2. セキュリティ要件

＜シングル構成と同じ＞

2.7.3. 性能要件

＜シングル構成と同じ＞

2.7.4. 耐障害性要件

項番	大分類	中分類	内容	備考
1	耐仮想マシンホスト障害		複数の仮想マシンホストが利用できる場合は、複数の仮想マシンホスト上に、システムを分散して構築し、一つの仮想マシンホストが障害になった場合でも、システム停止しないこと。	※今回は複数の仮想マシンホストを利用できないため、対象外
2	耐サーバ(仮想マシン)障害		仮想マシンの冗長化を行い、片方の仮想マシンが障害になっても、機能を停止しないこと。	1 号機 2 号機の同時障害は考慮しない。また、正系・副系の同時障害も考慮しない。
3	耐ソフトウェア障害	OS	OSの冗長化を行い、片方のOSが障害になっても、機能を停止しないこと。	二重障害時は考慮しない。
		プロセス	プロセスの冗長化を行い、片方のプロセスが障害になっても、機能を停止しないこと。	二重障害時は考慮しない。
4	耐データ障害	バックアップリストア	復旧手順に従い作業を行うことにより、データ復旧できること。 また、データ復旧した時に、全営業日の状態に戻れること。	
5	許容する障害対応時間(MTTR)		1 分	

2.7.5. 拡張要件

＜シングル構成と同じ＞

2.7.6. 移行要件

＜シングル構成と同じ＞

2.7.7. 端末要件

＜シングル構成と同じ＞

2.7.8. 維持管理・サポート要件

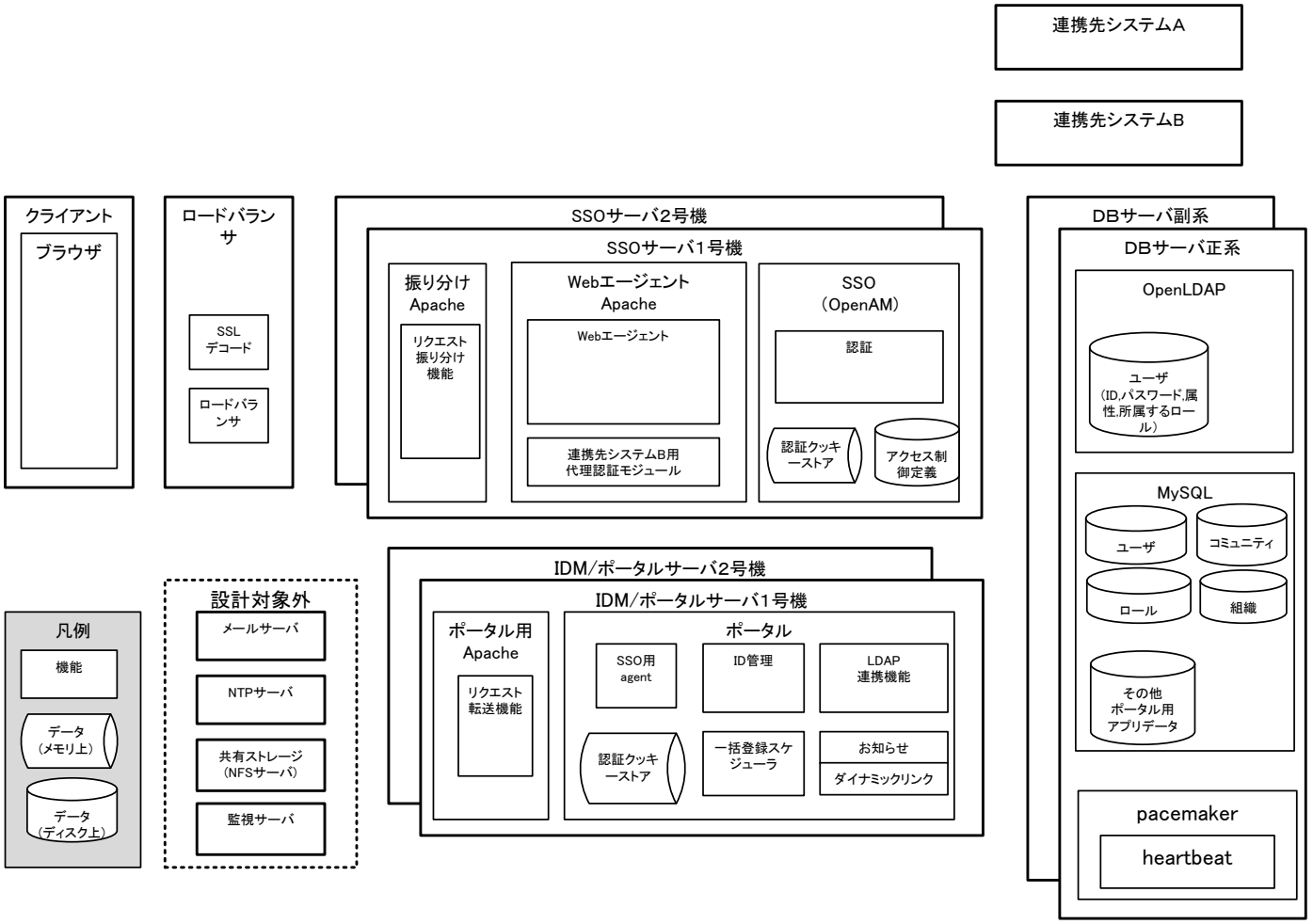
＜シングル構成と同じ＞

2.7.9. 開発要件

＜シングル構成と同じ＞

3. システム構成図

3.1. 全体構成図



設計意図

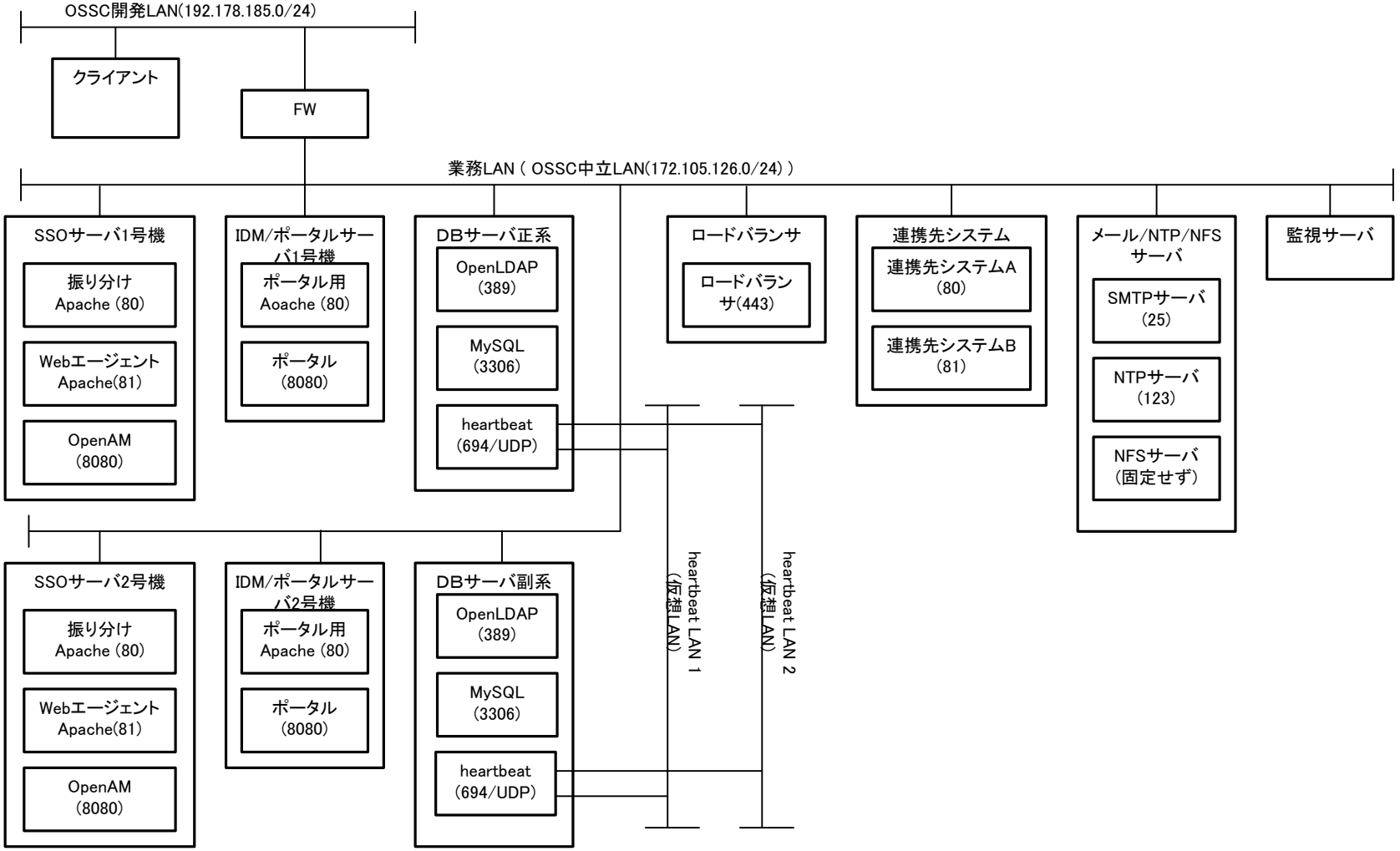
- 永続化が必要な情報を持たない SSO サーバと IDM/ポータルサーバはスケールアウト構成とする。
- 永続化が必要な情報を持つ DB サーバはクラスタ構成とする。クラスタは pacemaker+heartbeat で行う。
- IDM/ポータルサーバにはポータル用 Apache を前に置く。これにより、サイト閉塞等の運用を柔軟に素早く行うことができるようにする。

- ロードバランサには SSL デコード機能を持たせる。
- 共有ストレージは、SSO サーバなどが動作する仮想マシンホストとは、物理的に別のものにすること。

3.2. ネットワーク構成

3.2.1. ネットワーク構成図

※括弧内はポート番号



設計意図

- heartbeat 通信の帯域を確保するために、業務リクエストを通信する業務 LAN とは別に、heartbeat 専用の LAN を設ける。

- heartbeatLAN がダウンすると、クラスタの状態に不整合が発生するため、極力 heartbeatLAN は二重化する。

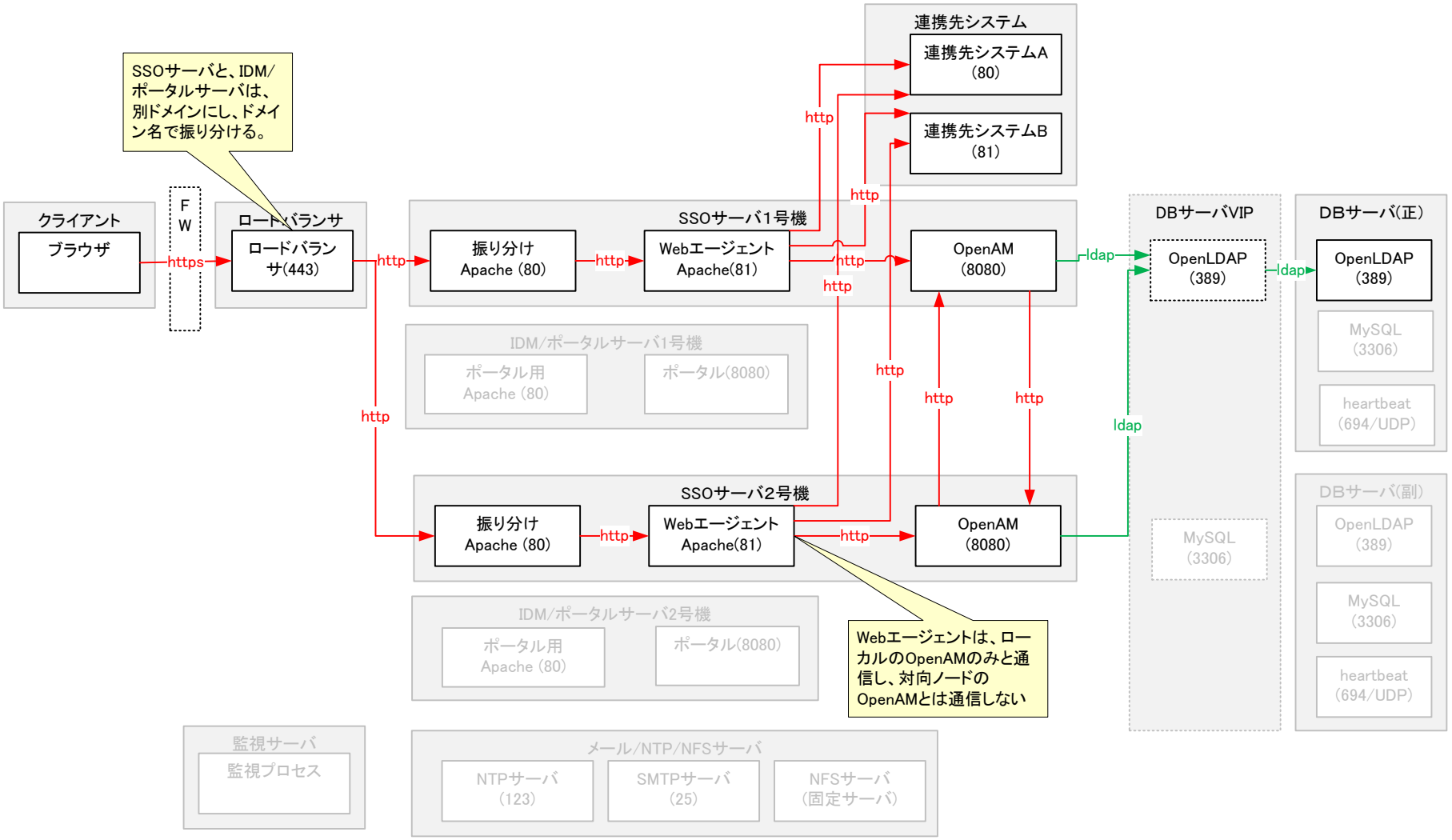
3.2.2. 仮想ホストの構成

本システムのサーバやネットワークは、VMWareESX 等の仮想マシンホスト上で動作することを想定している。
そのため、仮想マシンホスト上に配置する際には以下の構成になるように配置する。この構成が取れない場合、仮想マシンホストの障害により、システムの全面停止になる恐れがある。

- SSO サーバ、IDM/ポータルサーバ、DB サーバは、各ペアのマシンは別の仮想マシンホストにて稼働させる。
- Heartbeat LAN 1 と Heartbeat LAN 2 は仮想マシンホストの障害時に同時にダウンしないように、仮想ネットワークの割り当てに注意する。

3.2.3. 通信経路一覧

（１） シングルサインオン

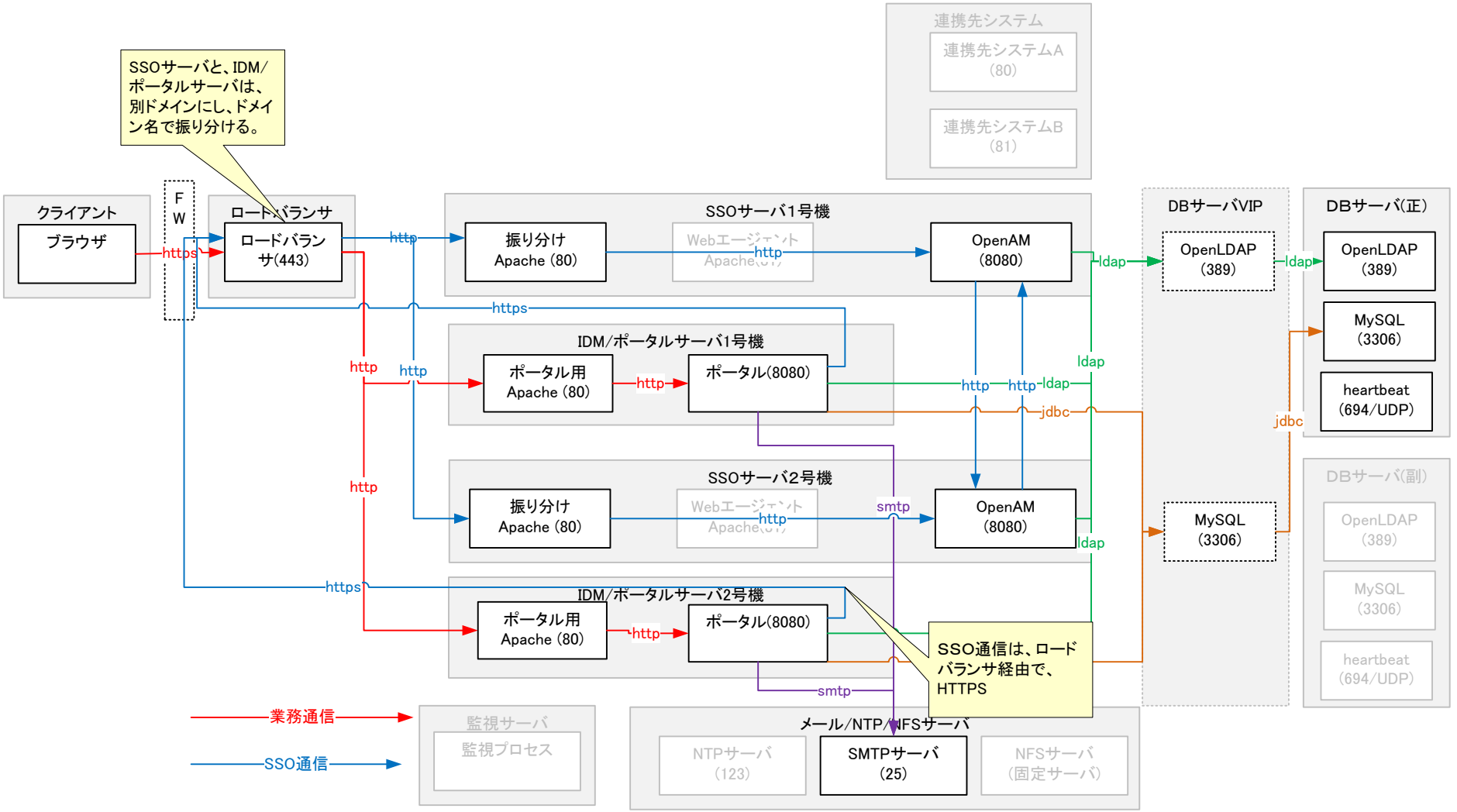


設計意図

- SSO サーバと IDM/ポータルサーバは別ドメインにする。
  - ロードバランサにコンテキストパスでリクエストを振り分ける機能がない事が想定されるため（クラウドのロードバランサはほとんどこの機能がない）
- Web エージェントは、ローカルの OpenAM のみを登録し、対向ノードの OpenAM とは通信しない。OpenAM の標準構成は、対向ノードと通信する方式であるが採

用しない。
  - 障害発生時の影響範囲をサーバ内にとどめ、障害解析を容易にするため
  - システム全体の流量を少なくするため
- DB サーバのクラスタは VIP を一つだけ持つ構成とする（MySQL と OpenLDAP で VIP を分けない）。
  - メンテナンスがしやすいため。
- ロードバランサから振り分け Apache に対する通信は、処理性能を上げるために、ユーザ毎に振り分けられるサーバが決まるようなロードバランス方式にする。
  - 同一ユーザからのリクエストは、同一サーバで処理するほうが、OpenAM 間のセッション照合処理がなく高速であるため。
  - クライアントの IP アドレスが十分にばらつく場合は、L4LB の IP スティッキネスでよいが、クライアントがプロキシを通してアクセスする等により IP アドレスに偏りがある場合は L7LB のクッキーによるスティッキネスを利用すべきである。

（２） IDM/ポータル利用

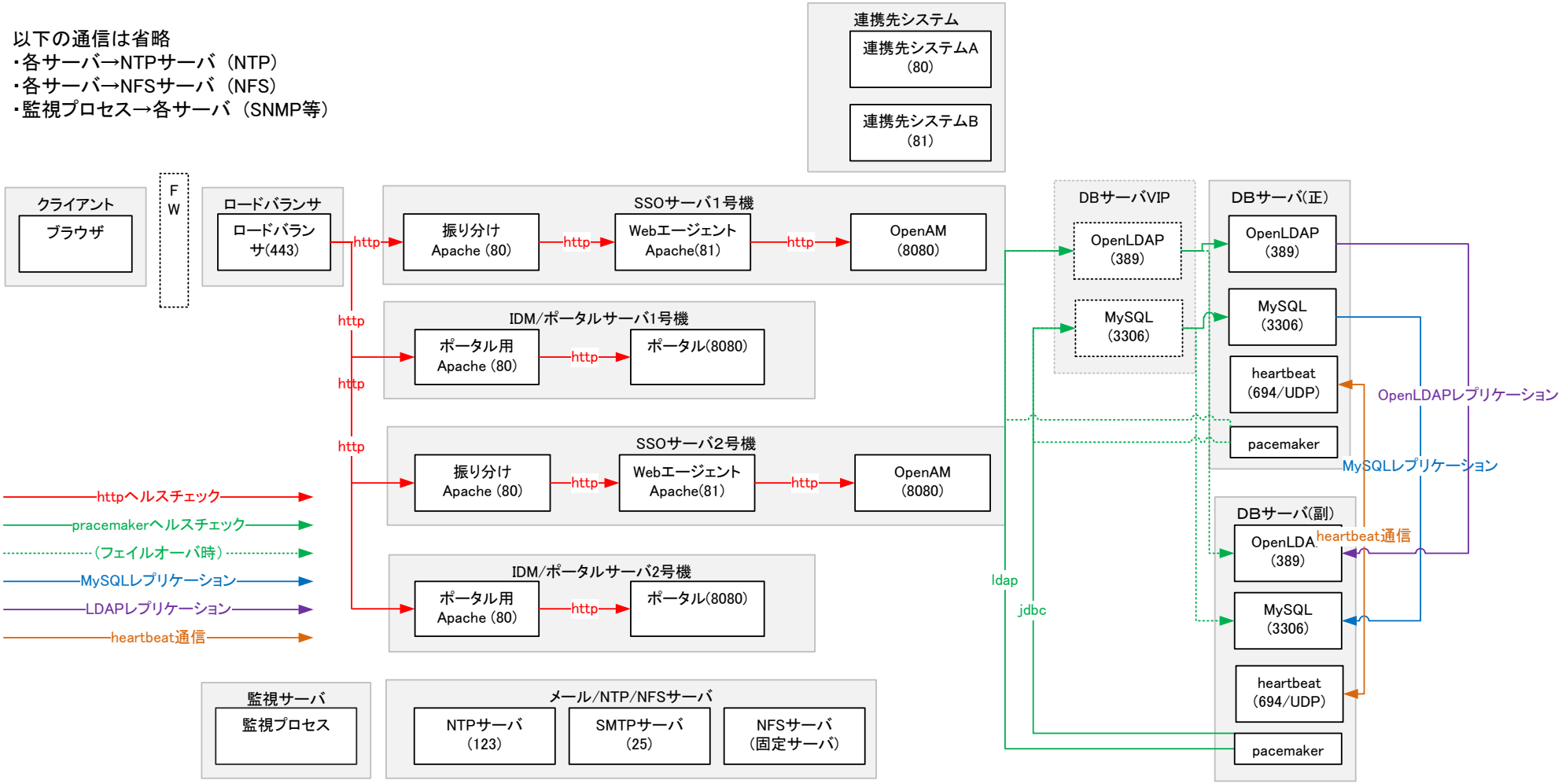


設計意図

- ロードバランサからポータル用 Apache に対する通信は、ユーザ毎に振り分けられるサーバが決まる負荷分散方式にする。
  - 同一ユーザからのリクエストは、同一サーバで処理するほうが高速である。OpenAM 間のセッション照合処理がないため。
  - クライアントの IP アドレスが十分にばらつく場合は、L4LB の IP スティッキネスでよいが、クライアントがプロキシを通してアクセスする等により IP アドレスに偏りがある場合は L7LB のクッキーによるスティッキネスを利用すべきである。

- IDM/ポータルサーバは、SSO サーバとは別ドメインにする。また、IDM/ポータルサーバから SSO サーバへの通信は、ロードバランサ経由にする。
  - SSO サーバと IDM/ポータルサーバは、お互いに相手が障害かどうかを気にする必要がなくなり、耐障害性が向上するため。
- IDM/ポータルサーバからロードバランサへの通信は、オンラインリクエストと同じプールを用いる。つまり HTTPS の通信になる。
  - 本来は、ロードバランサに、内部通信用のプールを作り HTTP で通信すべきだが、ロードバランサに内部通信用プール作成機能がない可能性があるため、標準化では採用しない。

（３） その他



項番	プロトコル	通信元		方向	通信先		説明
		説明	ポート		説明	ポート	
1	HTTPS	ブラウザ	any	→	振り分け Apache	443	
2	HTTP	振り分け Apache	any	→	Web エージェント Apache	81	リクエスト振り分け
3	HTTP	Web エージェント Apache	any	→	OpenAM	8080	認可
4	ldap	OpenAM	any	→	OpenLDAP	389	ユーザデータ参照
5	HTTP	Web エージェント Apache	any	→	連携先システム A	80	
6	HTTP	Web エージェント Apache	any	→	連携先システム B	81	
7	HTTP	振り分け Apache	any	→	ポータル	8080	リクエスト振り分け
8	HTTP	ポータル	any	→	OpenAM	8080	認可
9	ldap	ポータル	any	→	OpenLDAP	389	ユーザの認証データ CRUD
10	jdbc	ポータル	any	→	MySQL	3306	ポータルのデータ CRUD
11	HTTP	ブラウザ	any	→	ポータル	8180	管理コンソール直接利用
12	HTTP	振り分け Apache	any	→	OpenAM	8080	管理コンソール直接利用
13	SMTP	ポータル	any	→	SMTP サーバ	25	メール送信
14	NTP	SSO サーバ	any	→	NTP サーバ	123	時刻同期
15	NTP	IDM/ポータルサーバ	any	→	NTP サーバ	123	時刻同期
16	NTP	DB サーバ	any	→	NTP サーバ	123	時刻同期
17	SSH	ブラウザ	any	→	SSO サーバ	22	サーバ管理用
18	SSH	ブラウザ	any	→	IDM/ポータルサーバ	22	サーバ管理用
19	SSH	ブラウザ	any	→	DB サーバ	22	サーバ管理用
20	NFS	SSO サーバ	any	→	NFS サーバ	未固定	NSF マウント
21	NFS	IDM/ポータルサーバ	any	→	NFS サーバ	未固定	NSF マウント
22	NFS	DB サーバ	any	→	NFS サーバ	未固定	NSF マウント

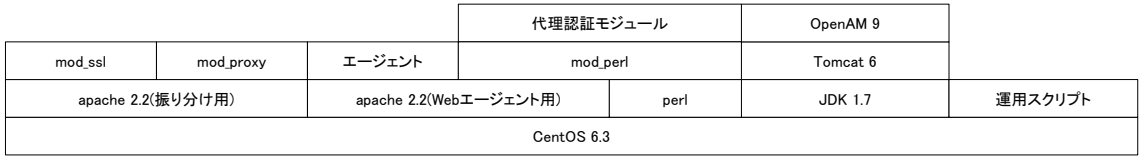
3.3. ハードウェア構成

項番	サーバ	HW 型番	CPU	メモリ	ディスク容量	IP アドレス	備考
1	SSO サーバ(1 号機)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.104	
2	SSO サーバ(2 号機)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.105	
3	IDM/ポータルサーバ(1 号機)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.106	
4	IDM/ポータルサーバ(2 号機)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.107	

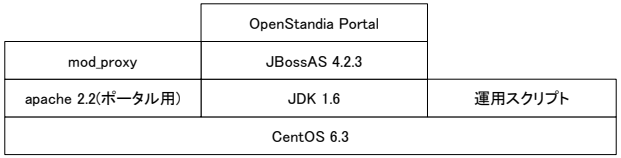
5	DB サーバ(正)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.108	
6	DB サーバ(副)	仮想マシン	Intel Corei7 2core	4G	20G	172.105.126.109	
7	連携先システム					172.105.126.100	
8	メール/NTP/NFS サーバ					172.105.126.110	設計対象外

3.4. ソフトウェア構成

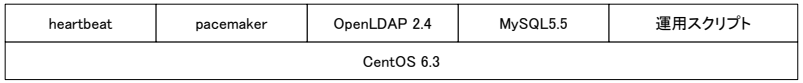
SSOサーバ



IDM/ポータルサーバ



DBサーバ



3.4.1. ソフトウェア選定の考え方

顧客の独自要件がない限り、以下のバージョンを使用する事。

3.4.2. ソフトウェアー覧

項番	サーバ	ソフトウェア	バージョン	備考
1	SSO サーバ 1 号機、2 号機	Red Hat Enterprise Linux	6.3	
2		JDK	1.7	
3		Apache httpd	2.2.24	振り分け用。OSSC 独自でソースからコンパイル
4		Apache httpd	2.2.24	エージェント用。OSSC 独自でソースからコンパイル

5		Tomcat	6.0.35	
6		OpenAM	9.5.5	
7		Web Policy エージェント	3.0.4	
8		perl	5.10.1	OS 付属のバージョン
9		mod_perl	–	代理認証モジュールに付属
10		mod_ssl	–	Apache httpd に付属
11		ruby	1.9.3	運用スクリプト実行用
12		rbatch	–	Ruby ベースの運用スクリプトフレームワーク
13	IDM/ポータルサーバ 1 号機、2 号機	Red Hat Enterprise Linux	6.3	
14		JDK	1.7	
15		JBoss AS	4.2.3	
16		OpenStandia Portal	5.2.3	
17		Apache httpd	2.2.24	振り分け用。OSSC 独自でソースからコンパイル
18		ruby	1.9.3	運用スクリプト実行用
19		rbatch	–	Ruby ベースの運用スクリプトフレームワーク
20	DB サーバ正系、副系	Red Hat Enterprise Linux	6.3	
21		OpenLDAP	2.4.25-42.el6	
22		MySQL	5.5.30	
23		heartbeat	3.0.5-1.1	
24		pacemaker	1.0.12-1	
25		ruby	1.9.3	運用スクリプト実行用
26		rbatch	–	Ruby ベースの運用スクリプトフレームワーク

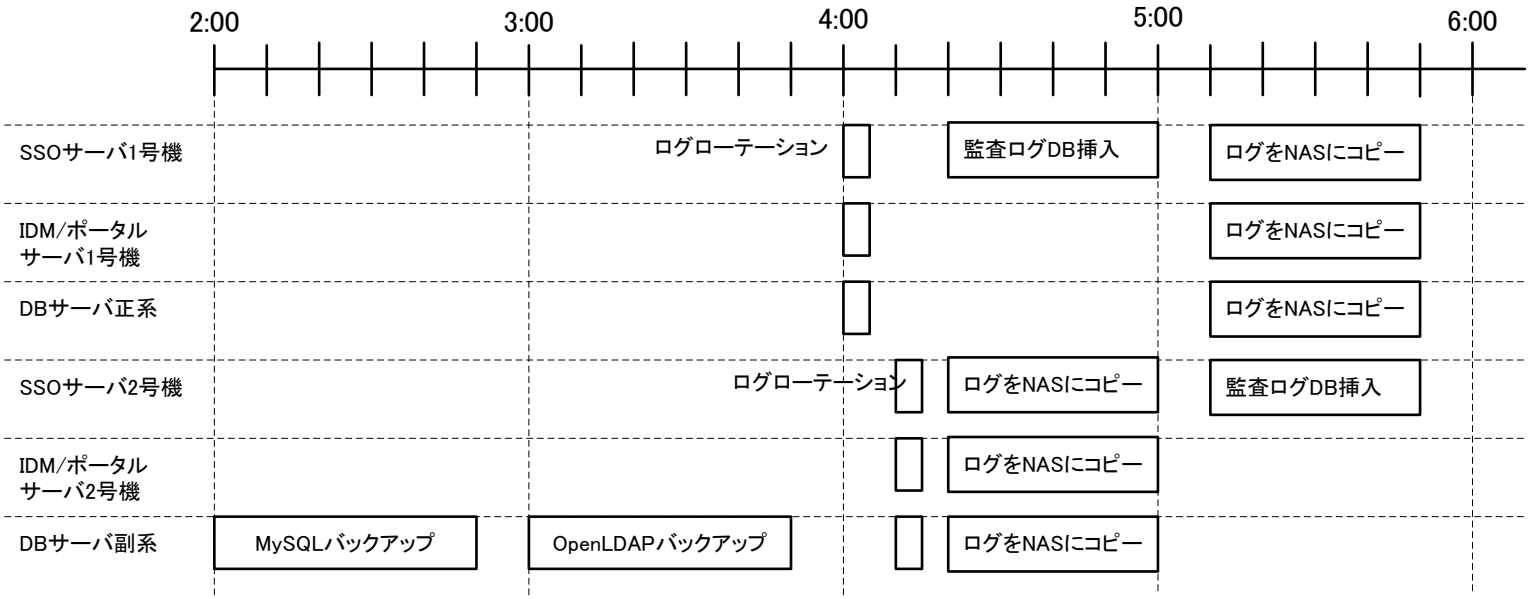
4. 処理方式

＜シングル構成と同じ＞

5. 運用設計

5.1. 定常運用

運用フローチャートは以下の通り



5.1.1. データバックアップ

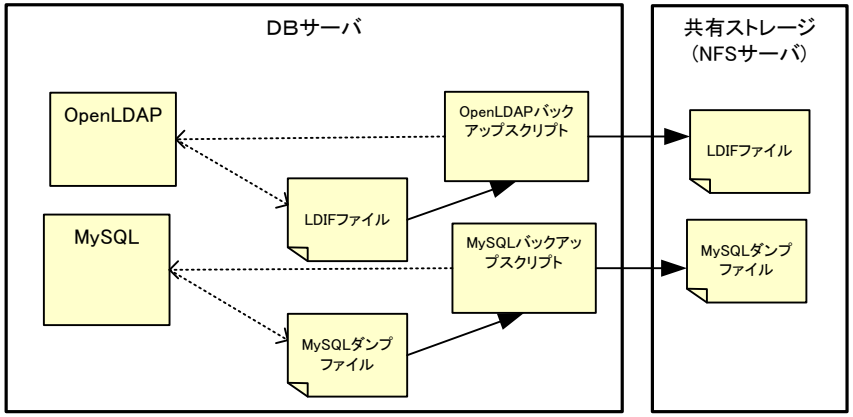
DB サーバをバックアップ対象とする。ただし、クラスタ構成であるため、状況に合わせてバックアップを取得するサーバを変える。
各状況に合わせたバックアップ実施ノードは以下の通り。

No	状況	バックアップ実施ノード	方法	理由	備考
1	正常時	DB サーバ副系	日次で自動実行	副系ノードの方が負荷が少ないため	
2	正系障害時	DB サーバ副系	日次で自動実行	正系のデータが正しい状態でない可能	

				性があるため	
3	副系障害時	DB サーバ正系	手動で正系のバックアップ運用を開始する		

バックアップの対象

サーバ	バックアップ対象プロダクト	バックアップ対象	バックアップ先	バックアップ取得方法	タイミング	保管期間	備考
DB サーバ	OpenLDAP	OpenLDAP の全データ	共有ストレージのバックアップディレクトリ	slapcat コマンドによりテキストファイルにデータを書き出し、共有ストレージにコピーする	日次 午前 2 時	7 日分	注意）更新がかかっている最中に、バックアップが行われると、論理的に不整合なバックアップになる可能性がある。OpenLDAP の slapcat コマンドにトランザクションの概念がないため。
	MySQL	全テーブル	共有ストレージのバックアップディレクトリ	mysqldump コマンドにてテキストファイルにデータを書きだし、共有ストレージにコピーする	日次 午前 3 時	7 日分	



5.1.2. ログ出力

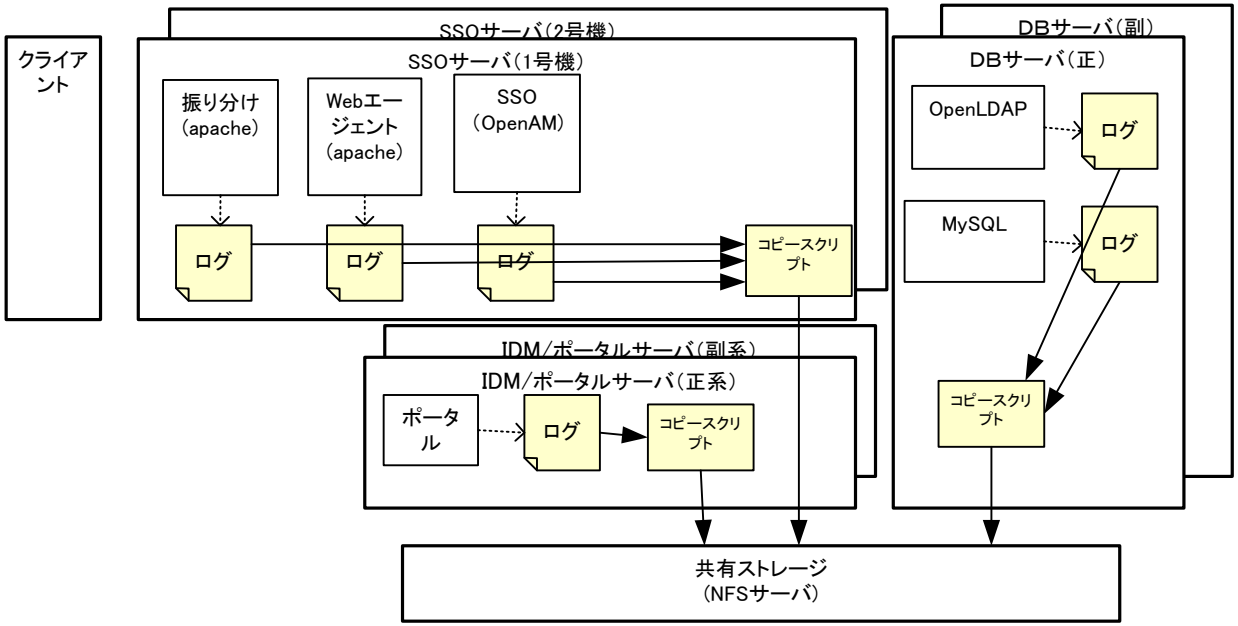
- 以下のミドルのログを取得する
  - Apache
  - Tomcat
  - OpenAM
  - OpenLDAP
  - MySQL
- ログローテーションは以下の通り実施する
  - 1週間以上前のものは削除する。
  - 毎日午前 4:10 にローテーションする

- 初回のローテーションでは圧縮は行わず、その次のローテーションでログを圧縮する
- ローテーションする場所は、ログファイルと同じディレクトリとする
- ログファイルが存在しない場合にエラーを出力しない

- Apache のログローテーションは、Apache を瞬断し行う。
  - 理由はログローテーション中にアクセスがあった場合は、監査ログが失われるため。

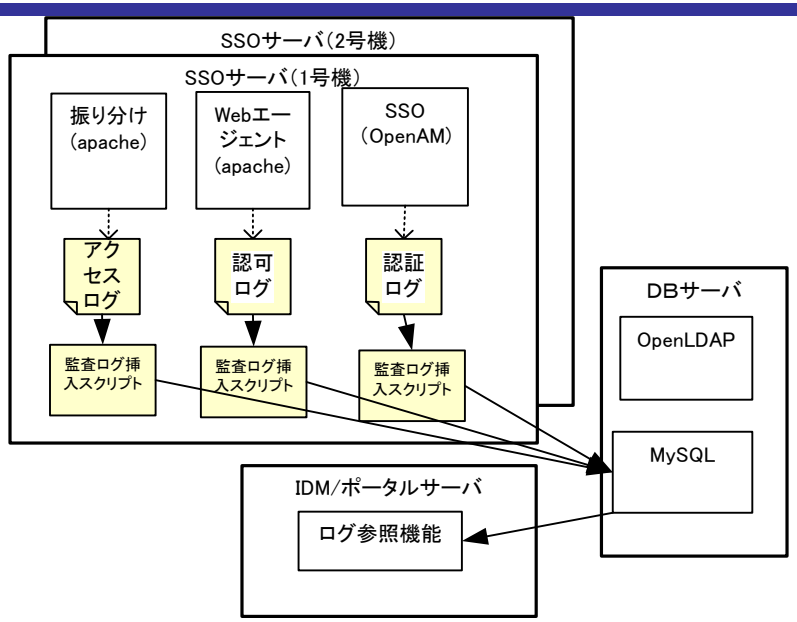
5.1.3. ログ保管

- ログの共有ストレージへのコピー
  - 毎日午前 4:20 にログファイルを共有ストレージへコピーし、18 カ月分保管する。



5.1.4. ログ参照

- SSO サーバの Web アクセスログ、ログを MySQL に挿入
  - ポータルから監査ログを見るために、一部のログを MySQL に挿入する。
  - 5:00 に MySQL への挿入を行う。
  - MySQL へ挿入するログの一覧は、**エラー！ 参照元が見つかりません。**参照



5.1.5. 稼働統計情報取得

- sar コマンドにより OS の CPU 使用率、メモリ使用率、IO 使用率を定期的に取得する
- netstat コマンドの出力結果を定期的に保存する。

5.1.6. 時刻同期

- 各サーバに NTP デーモンを起動させ、NTP サーバと時刻同期を行う。

5.1.7. 常時サービス提供

- すべてのサーバは 24 時間 365 日動作させ続ける。

5.2. 障害時運用

以下の運用手順書を準備する。

- サーバ再起動手順
- プロセス起動・停止手順
- MySQL データバックアップ・リストア手順
- OpenLDAP データバックアップ・リストア手順
- DB サーバフェイルバック手順

6. セキュリティ設計

＜シングル構成と同じ＞

※冗長構成の場合は、IDM ポータルサーバの Apache ログ(アクセスログ)も追加で監査対象ログとなる。

7. 耐障害設計

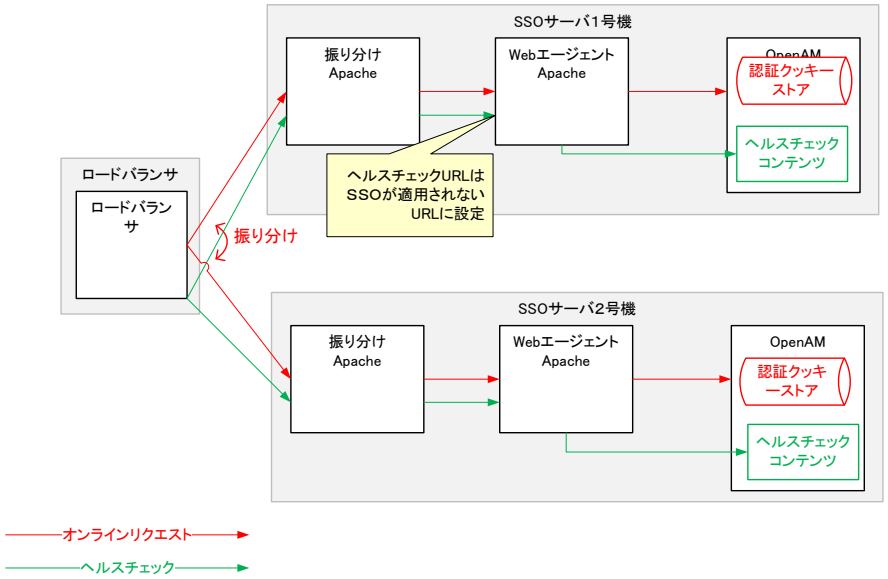
7.1. 冗長化

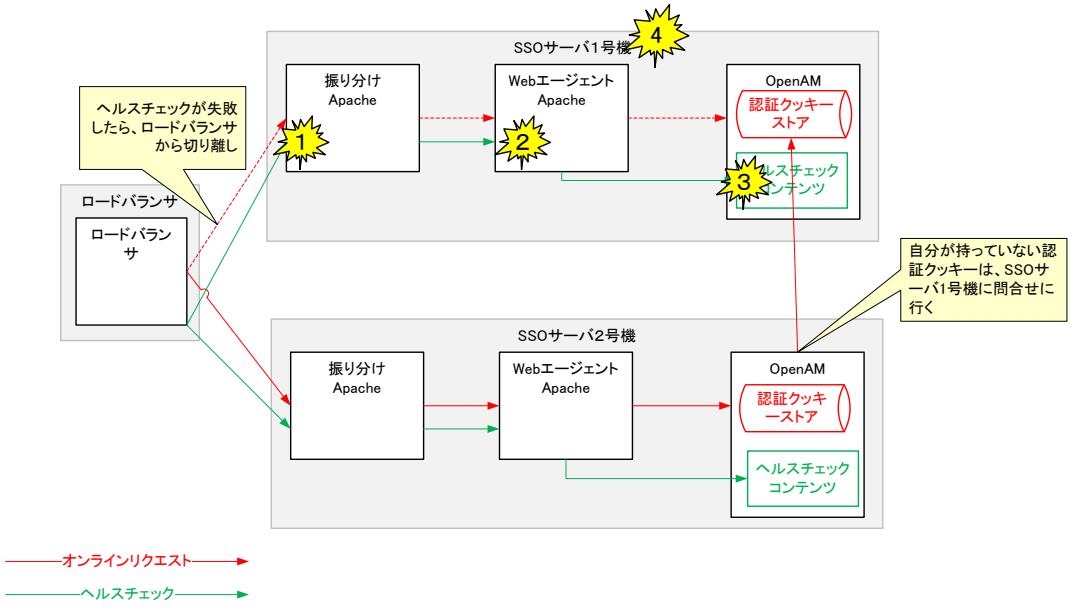
7.1.1. 基本方針

永続化が必要な情報を持たない SSO サーバと IDM/ポータルサーバはスケールアウト構成とし、永続化が必要な情報を持つ DB サーバはクラスタ構成とする。

7.2. 障害検知方法とその場合の動作

7.2.1. SSO サーバ

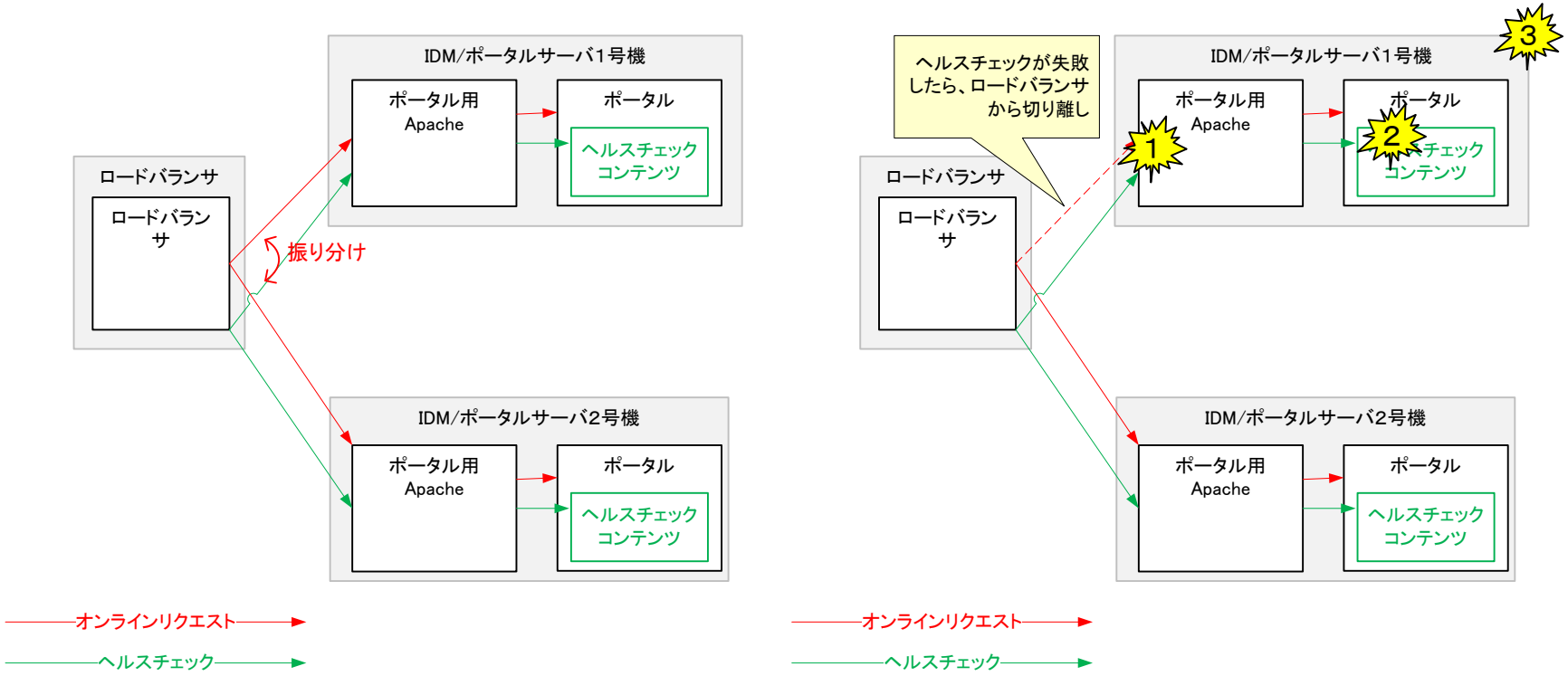




No	障害発生個所	障害検知方法	動作	影響	備考
1	振り分け Apache プロセス	ヘルスチェック失敗	ロードバランサから障害となったサーバを切り離し、以降は 1 台のサーバでリクエストを処理する。	障害発生から切り離しまでの間、一部のリクエストがエラーになるが、その後はリトライにより継続利用可能(再ログイン必要なし)。	
2	Web エージェント Apache プロセス				
3	OpenAM プロセス			障害発生から切り離しまでの間、一部のリクエスト	

4	OS			がエラーになるが、その後は再ログインが必要。	

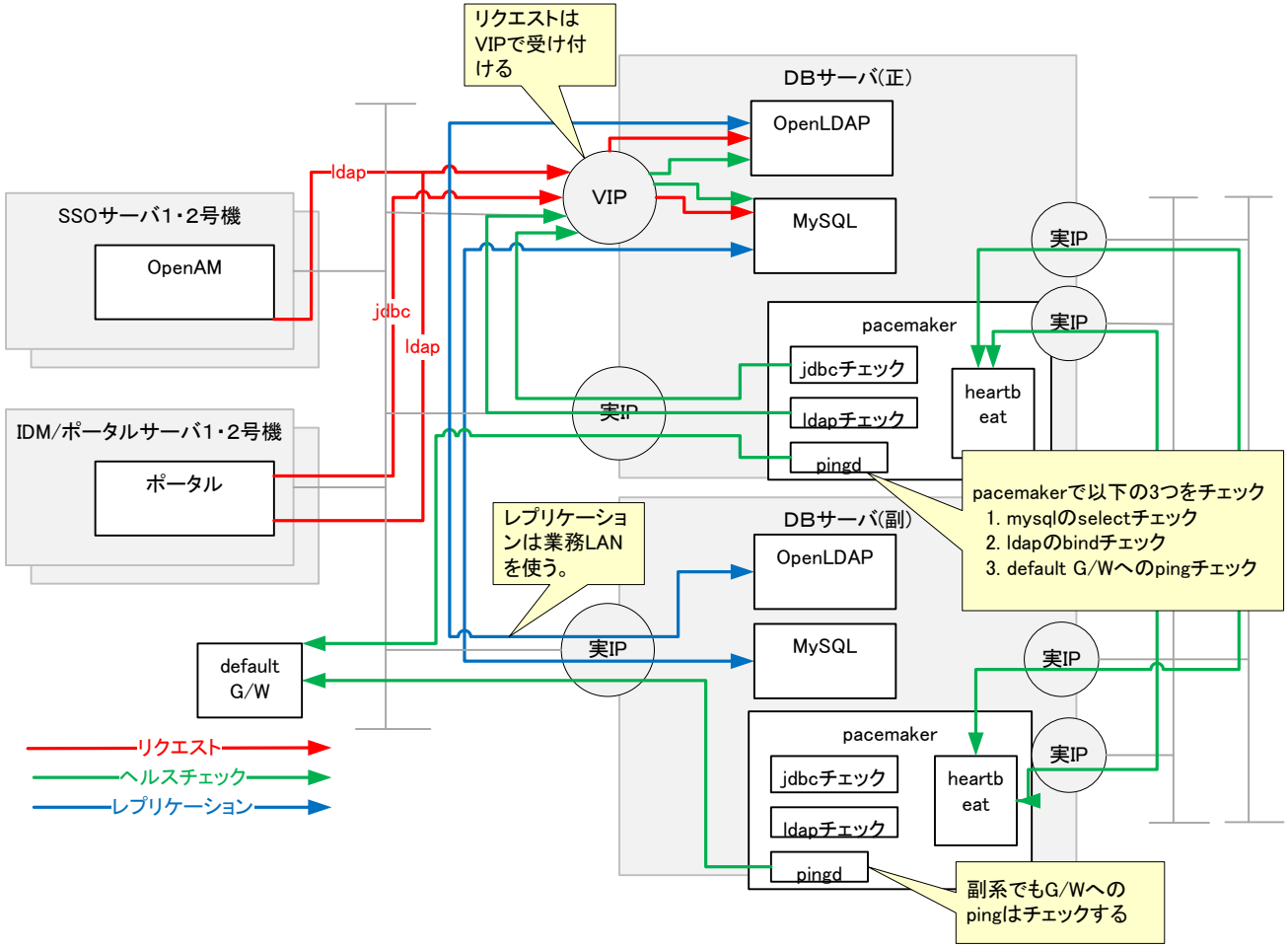
7.2.2. IDM/ポータルサーバ



No	障害発生個所	障害検知方法	動作	影響	備考
1	ポータル用 Apache	ヘルスチェック失敗	ロードバランサから障害となったサーバを切り離し、以降は 1 台のサーバでリクエストを処理する。	障害発生から切り離しまでの間、一部のリクエストがエラーになる。 ポータルからはログアウトするが、SSO にはログイン状態であるため、リトライすれば再ログインは必要ないが、ポータルの作業情報(wiki の編集画面など)はなくなる。	
2	ポータル				
3	OS				

7.2.3. DB サーバ

（１） 通常時



動作説明

- Pacemaker に自前の OCF スクリプトを用意して、MySQL と OpenLDAP の死活監視を行う。加えて、ネットワークの正常性監視として、defaultG/W への ping チェックを行う。
- Pacemaker のチェック間隔は 10 秒。チェックに失敗したら即リトライ。2 回連続で失敗したらフェイルオーバを実施。
- 副系ノードでは、mysql や OpenLDAP の監視は行わないが、default G/W への ping チェックは行う。
- チェック失敗時には、heartbeat のプロセスが syslog にエラーを書き込む。このエラーを運用監視ツールで監視することにより、システム管理者はクラスタ状態の異常

に気付くことができる。

設計意図

- MySQL および OpenLDAP のレプリケーションは、マルチマスタの双方向レプリケーションとする。
  - マルチマスタレプリケーションでは、データは共有せずに各々のノードが管理している。そのため、スプリットブレイン発生時に一つのデータを複数のプロセスで書き込んで壊してしまうような事態を避けることができる。また、フェイルバック時にも障害復旧が容易である。
  - 一方、レプリケーションが非同期であるため、正系にデータを書き込んでから副系にデータを書き込む間に、障害が発生するとデータロストする可能性がある。厳密に書き込みの一貫性を保証したい場合は、データを共有ディスクに置く構成などを検討する事。また、OSと共有ディスクの接続は信頼性の面でNFSは避けるべきである。

（２） 障害発生時パターン

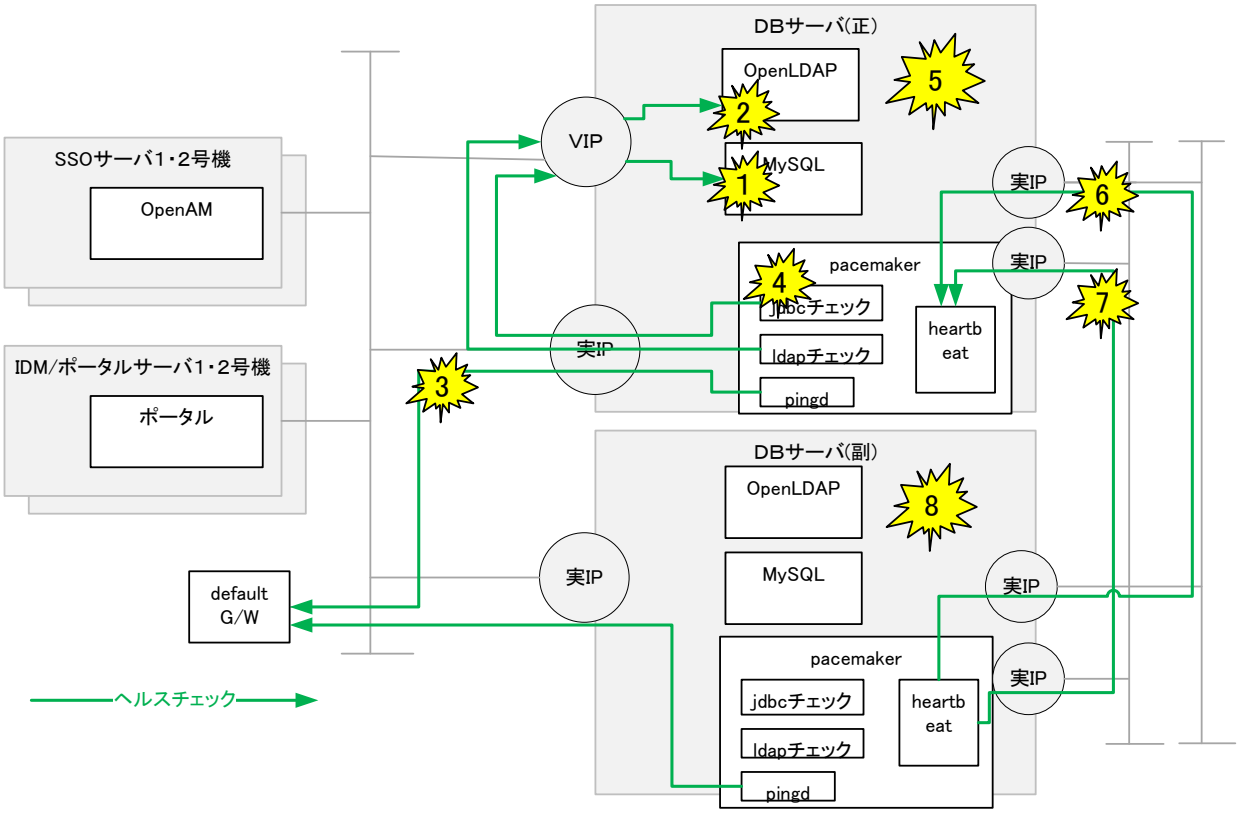
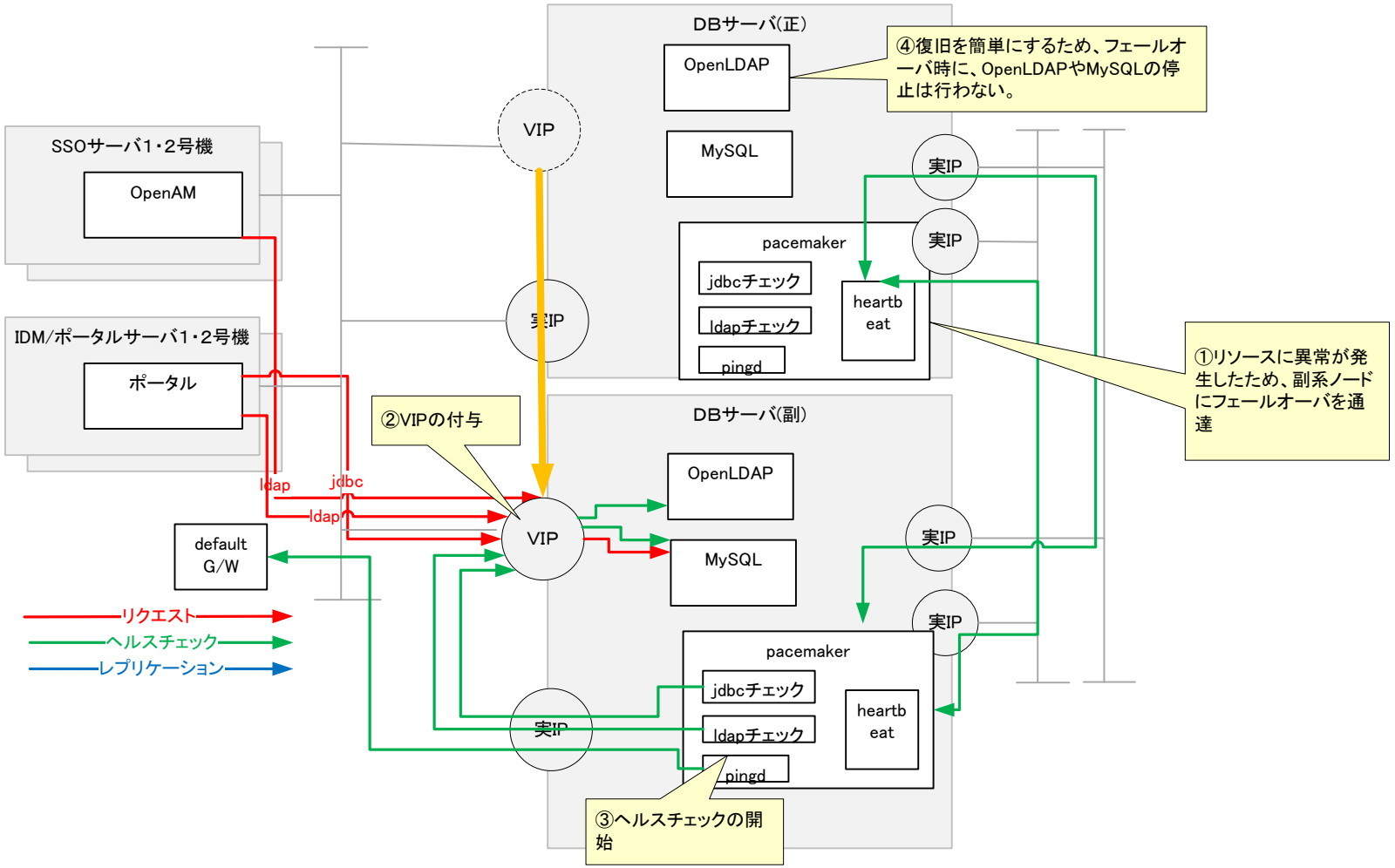


表 1 障害発生パターン一覧と対応方法

No	障害発生個所	障害検知方法	動作	影響	備考
1	正系 MyAQL プロセス	Pacemaker のリソース異常（SQL の実行失敗）	・副系にフェイルオーバ ・VIP を副系に付与する。 ・副系にてリソースのチェックを開始しする。	障害発生からフェイルオーバ完了までの間、リクエストが返ってこない、もしくは失敗する。 MySQL と OpenLDAP のレプリケーションは非同期であるため、場合によってはトランザクションのロストが発生する。	
2	正系 OpenLDAP プロセス	Pacemaker のリソース異常（Ldap bind の失敗）			
3	正系業務 LAN の障害(NIC/ルータ/スイッチ)	Pacemaker のリソース異常（Default G/W への ping 失敗）			

4	正系 pacemaker / heartbeat プロセス	Heartbeat 失敗			
5	正系 OS	Heartbeat 失敗			
6	heartbeatLAN 1 (NIC/ルータ/スイッチ)	Heartbeat にてエラー検知	何もしない	影響なし	
7	heartbeatLAN 2 (NIC/ルータ/スイッチ)	Heartbeat にてエラー検知	何もしない	影響なし	
8	副系全般	Heartbeat 失敗	何もしない	影響なし	

（３） フェイルオーバ時の動作



動作説明

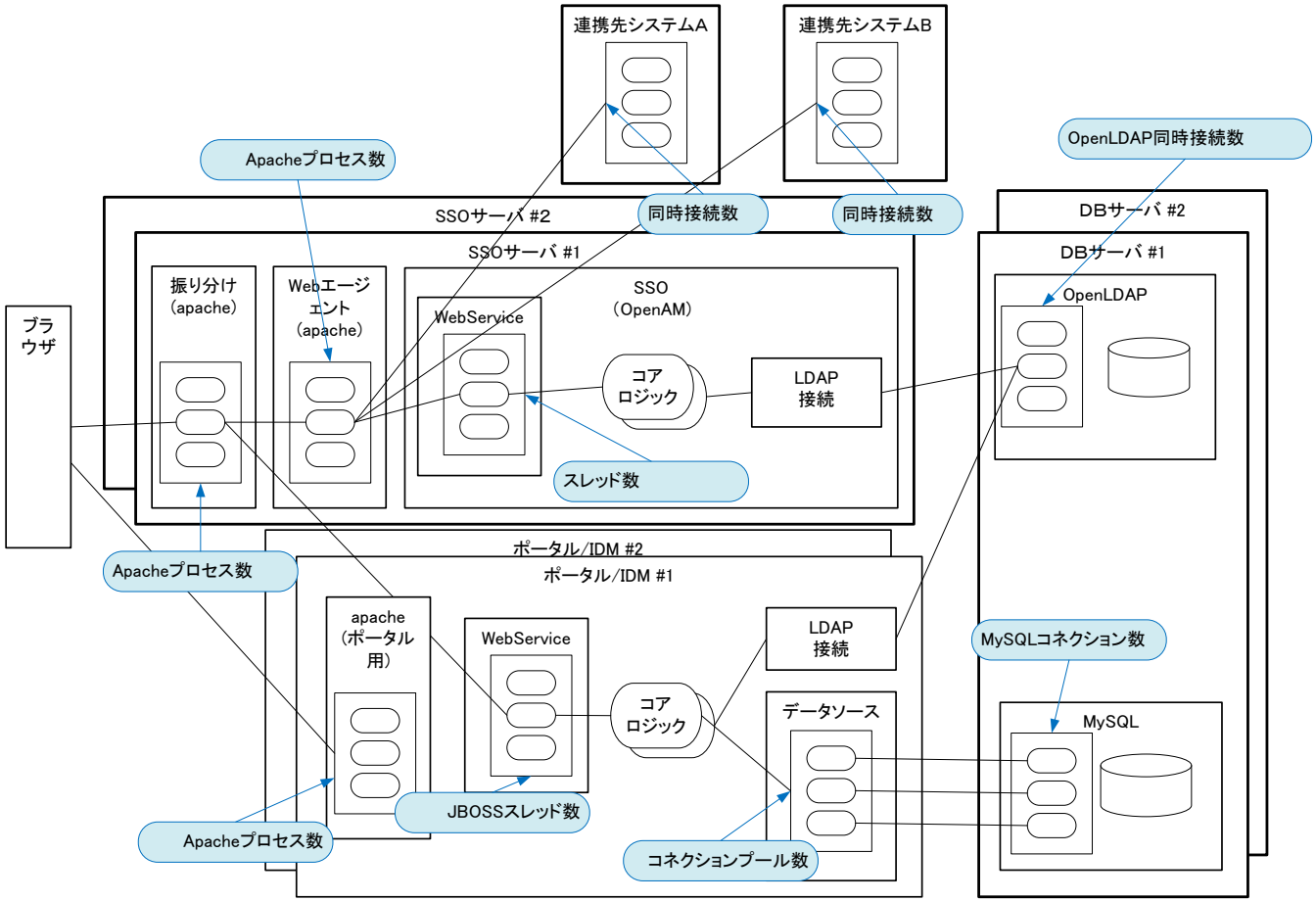
- 自動でフェイルバック(VIP のつけ戻し)は行わない。
- フェイルオーバ・フェイルバック時に MySQL や OpenLDAP のプロセスの起動停止の制御は行わない。
  - 基本的にプロセスは常時起動させておき、VIP の付け替えのみでアクティブノードを切り替える。
  - 理由は、特に MySQL は自分で勝手に起動する仕組みがあり、pacemaker で MySQL の起動停止を制御しにくいため。

8. 性能設計（サイジング）

8.1. 性能目標

- CPU 使用率 40%未満、メモリ使用率 40%未満で、全ての性能要件を満たせること。
  - 理由は、障害発生時の片系運用時に負荷が倍になったとしても、CPU 使用率 80%未満、メモリ使用率 80%未満で処理できるため。

8.2. サイジング



No.	サーバ	設定箇所	設定値		備考
1	SSO サーバ#1/#2	振り分け(Apache)プロセス数	80		性能要件と過去の実績値により設定。
2		Web エージェント(Apache)プロセス数	80		振り分け(Apache)の MaxClients と合わせる。
3		Tomcat スレッド数	80		振り分け(Apache)の MaxClients と合わせる。
4	ポータルサーバ#1/#2	ポータル用 Apache プロセス数	80		性能要件と過去の実績値により設定。
5		JBoss スレッド数(8080 ポート)	20		性能要件と過去の実績値により設定。
6		JBoss スレッド数(8180 ポート)	10		管理者用ポートであり同時アクセス数は少ないため、8080ポートの半分に設定。
7		コネクションプール数(合計)	36	30	lportal コネクションプール数(Liferay) ※JBoss スレッド数と合わせる。
8				5	org001 コネクションプール数(Liferay) ※グループウェアは現時点では使用しないためデフォルトの 5 のままとする。
9				1	lportal コネクションプール数(quartz) ※スケジューラからの DB 接続用に 1 本確保する。
10	DB サーバ#1/#2	MySQL コネクション数	100		ポータルサーバ 2 台同時接続することを考慮しつつ、最大想定数+α。 バックアップバッチ、heartbeat からの監視用などの余裕を持たせる。
11		OpenLDAP 同時接続数	1024		ulimit の値となる

9. 拡張設計

＜シングル構成と同じ＞

10. 移行設計

移行要件なし

11. 端末設計

特に記載する内容は無い

12. 維持管理・サポート方針

ソフトウェアの技術的なサポートについては、OpenStandia の OSS 年間サポートに加入する。
トラブル発生時には、OpenStandia の OSS 年間サポートに問い合わせるが、その際顧客主体でログの取得やプロセスのダウンアップができるように、手順書等を準備する。

13. 開発方針

以下の3つの環境を用意する。

項番		SSO	IDM/ポータル	DB	連携先システムA	連携先システムB	ロードバランサ	メール	NTP	共有ディスク	監視サーバ	備考
1	本番環境											
2	ステージング環境	本番環境と同じソフトウェア構成にする。			本番環境とインターフェースをそろえる。実装方法は問わない。						用意しない。	今回はテンプレートの開発なので準備せず
3	開発環境	本番環境と同じソフトウェア構成にする			ダミーシステムを用いる。		用意しない。	ベンダのメールサーバを用いる	ベンダのメールサーバを用いる	用意しない。ローカルディスクを用いる	用意しない。	今回はテンプレートの開発なので準備せず

14. 付録

＜シングル構成と同じ＞