

Oracle DBA & Developer Days 2011

日本オラクル、今年最大の技術トレーニングイベント

2011年11月9日(水)～11月11日(金) シェラトン都ホテル東京



ORACLE®

**オラクルコンサルが語る！超高速！Oracle Coherenceの
高可用性を支える多様な障害検知アーキテクチャ**

日本オラクル テクノロジーソリューションコンサルティング統括本部
矢形拓也

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

Coherenceが採用される理由

Why Coherence?

ORACLE

Oracle Coherenceが多くのシステムで採用される理由#1

まず、障害検知の仕組みという本題に入る前に...

- Oracle Coherenceがミッションクリティカルな分野で多く利用される理由を考えて見ました
- Coherenceコンサルとして支援させていただいたお客様がプロジェクト当初に必ず口にされるキーワードは

性能

(高トランザクション、低レイテンシ)

スケールアウト性

高可用性

Oracle Coherenceが多くのシステムで 採用される理由#2

性能

(高トランザクション、低レイテンシ)

スケールアウト性

高可用性

他の分散KVS製品でも、
Key, Valueのストア先が
メモリであればある程度実現できる

KVSの特性上大規模なクラスタを
組むことは他の製品でも可能

障害発生時にデータを失うことなく
自律的かつ高速に復旧する仕組みは
優れているという意見を多く聞く

アジェンダ

- 可用性を高く保つためのフェイルオーバ処理概要
 - 検知・昇格・再配置
- Coherenceはどのように障害を検知するか
 - プロセス障害
 - マシン障害
 - JVMハング
- Coherence障害検知アーキテクチャのポイント

可用性を高く保つための フェイルオーバー処理概要

Failover sequence...

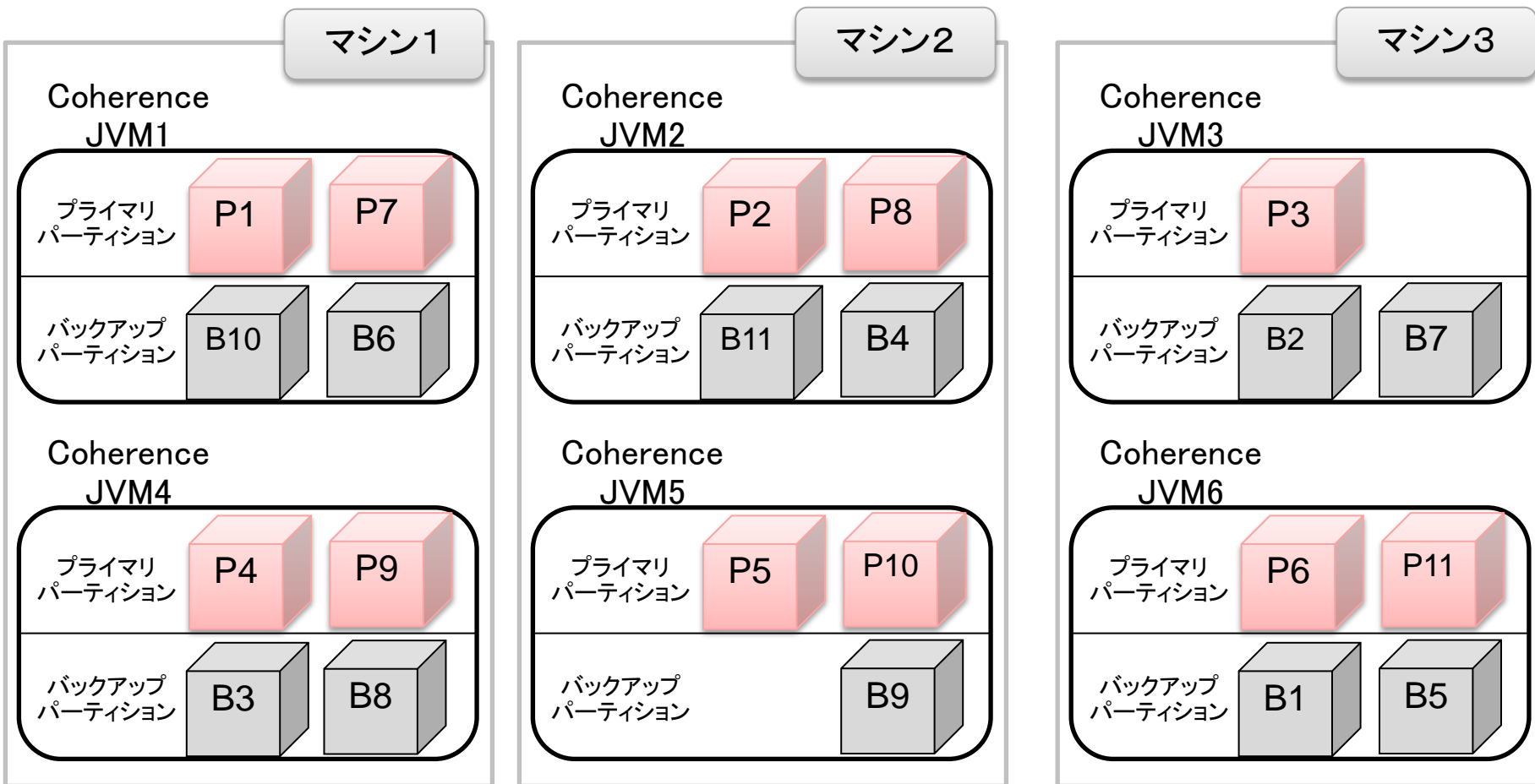
可用性を高く保つためのフェイルオーバー処理概要

- Coherenceは複数のマシンに配置したJVMにデータを保持する
- 障害が発生した場合においてもデータを失わないように、データ作成時にバックアップを作成する
- バックアップは、オブジェクトごとではなくある程度のオブジェクトをまとめたパーティション単位で行われる
- パーティション数は格納されるデータ量やJVM数によって適切な値に設定する
- バックアップはCoherence用のマシンが複数台存在する場合はマシンが停止しても復旧可能とするため、基本的には必ず別のマシンに配置される

Coherenceのプロセスとパーティション

～通常状態でのパーティション配置～

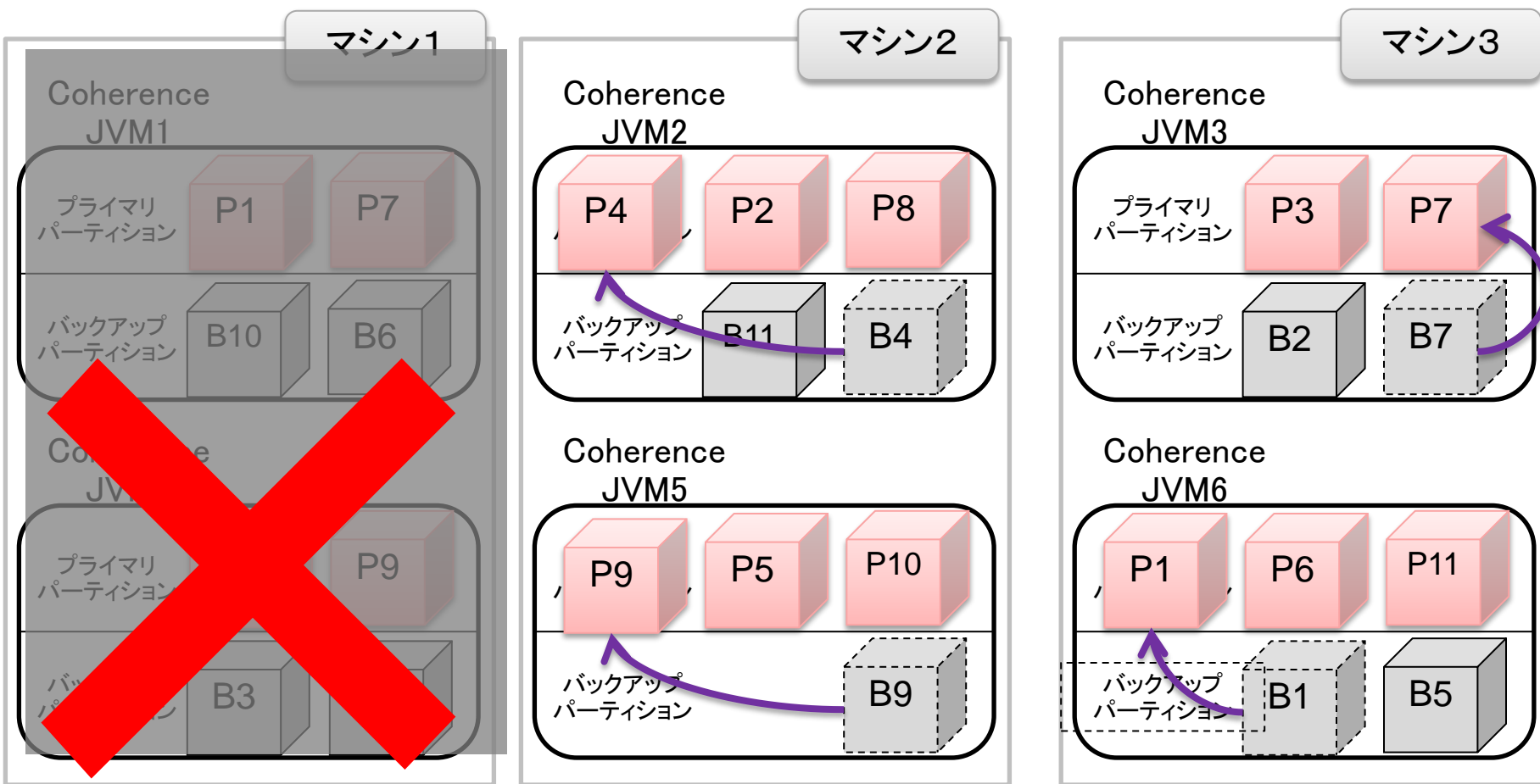
- マシン3台、Coherence JVM 6台、パーティション11個の例



Coherenceのプロセスとパーティション

～障害発生時の昇格処理～

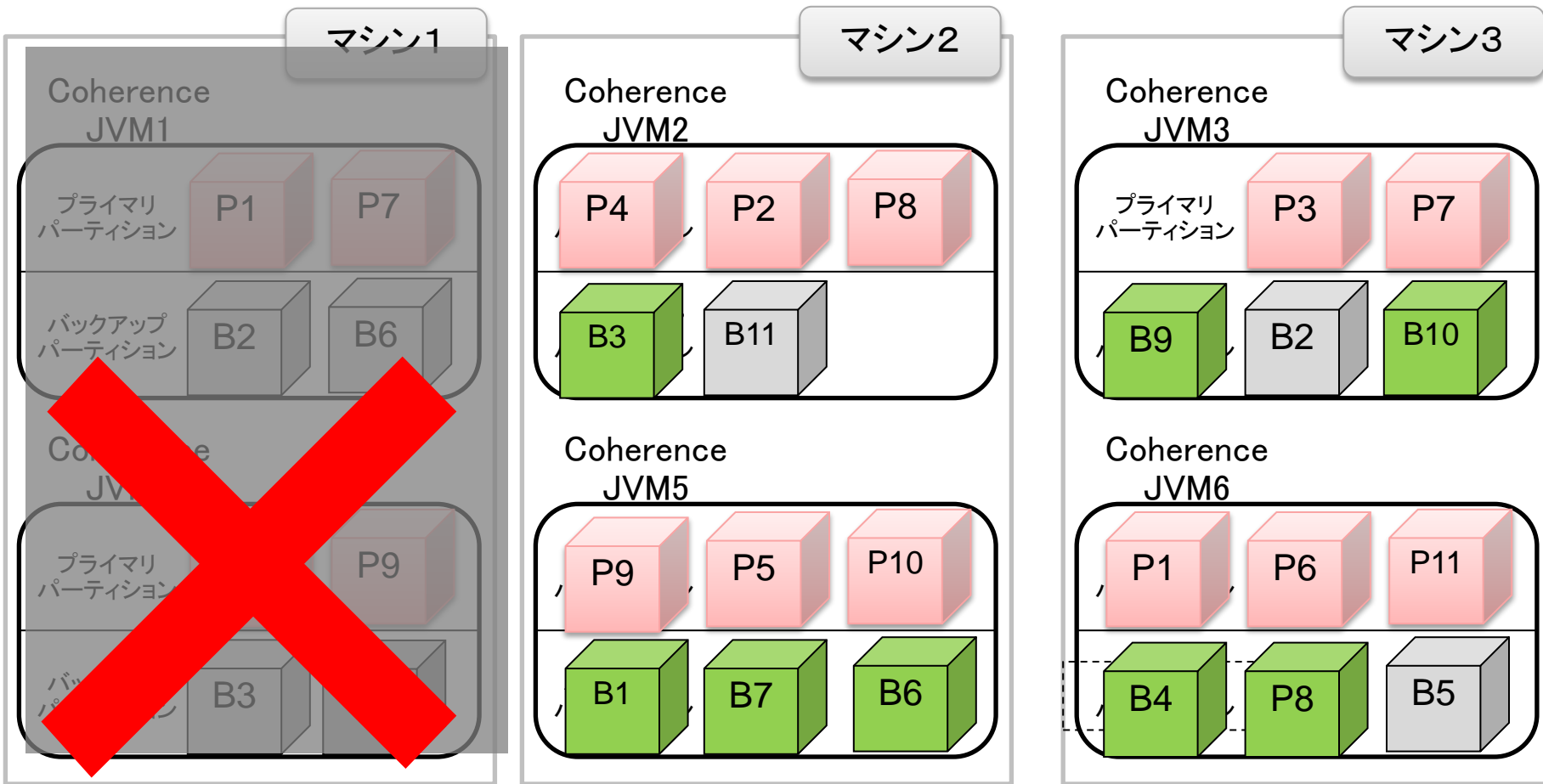
- 障害(マシン1が停止)が発生するとバックアップが昇格



Coherenceのプロセスとパーティション

～パーティション再配置～

- 失ったバックアップの作成および、同一マシンにできてしまったバックアップ、偏ったプライマリパーティションの再配置を実施する



Coherenceのフェイルオーバー動作

- Coherenceのフェイルオーバーは以下の3フェーズに分かれている
 - 障害検知フェーズ
 - JVMとの通信ができなくなり、残されたCoherence JVMが障害箇所を切り離す判断をする
 - バックアップ昇格フェーズ
 - 失われたバックアップを昇格する
 - パーティション再配置フェーズ
 - プライマリ、バックアップパーティションを移動させ偏りを無くす
 - 失われたバックアップの再作成を行う

障害検知
フェーズ

バックアップ
昇格
フェーズ

パーティション
再配置
フェーズ

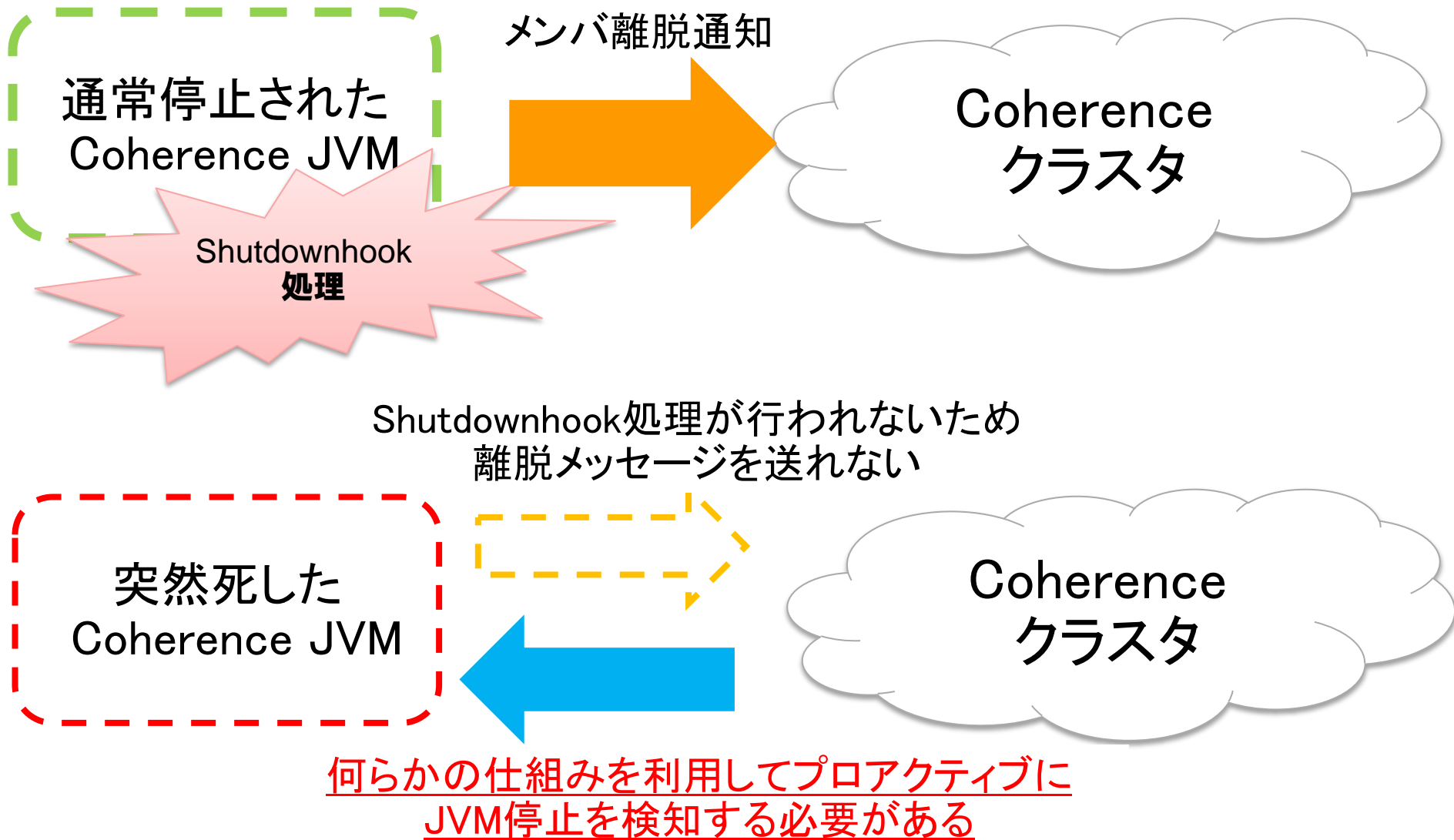
Coherenceはどのように障害を検知するか

Failure Detection...

CoherenceJVMの正常停止と異常停止#1

- CoherenceJVMの正常停止、異常停止の判断基準はJVM停止時に離脱メッセージを投げることができるかが基準となります
- CoherenceJVMの停止には専用のコマンドがあるわけではなく、プロセス停止のコマンドをOSから発行する
 - Linuxの場合はkillコマンドを利用
- プロセス停止のコマンドを利用してJVMを正常に停止した際にはJVM停止時にshutdown hookが呼ばれ、離脱メッセージをクラスタに投げる
- JVMプロセスがさまざまな要因により停止し、shutdown hookが呼ばれない場合には、離脱メッセージが送れないため、外部のJVMが検知する必要があり、“障害”とみなす
- 正常停止の場合であっても、フェイルオーバー処理は実施される

Coherence JVMの正常停止と異常停止#2



Coherenceの障害ケースと検知機構#1

- Coherenceはネットワークのみを利用して、障害の発生を検知し、切り離しを行う自律型のクラスタ
- いかに早く検知するかを突き詰めて考えた結果、障害ケースごと別々の検知機構を用意している
- Coherenceは複数のプロトコルを駆使して、可能な限り高速に障害を検知する

Coherenceの障害ケースと検知機構#2

- ・ 検知機構には以下の3つが用意されている

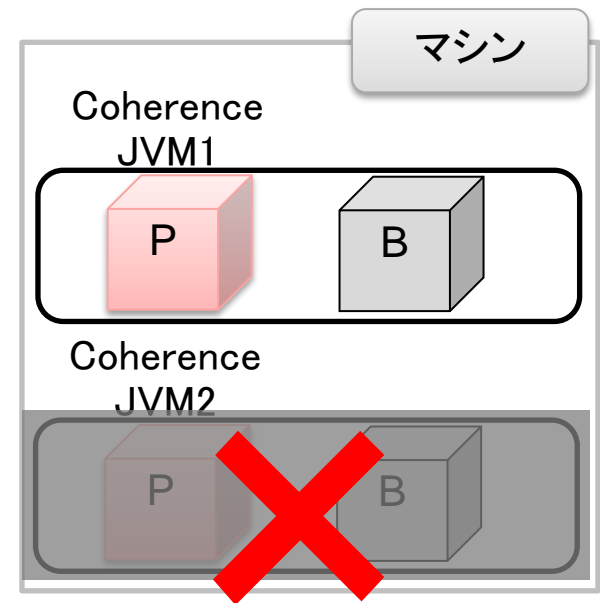
プロセス障害に即時対応するTCP Ring

マシン障害、ネットワーク障害に対応する
IP Monitor

JVMハングに対応するパケットデリバリ
タイムアウト

プロセス障害に即時対応するTCP Ring#1

- プロセス障害を高速に検知する機構
 - CTRL+C、Kill -9
 - 内部的なバグによるJVM異常終了
- クラスタ内のCoherence JVMがRing上にTCPコネクションを確立する。
- 重要なのはコネクションを利用した通信ではなくTCPのコネクション
 - メインリング
 - 補助リング
- プロセスが停止するとOSがTCPコネクションを切断し、切断を検知したCoherence JVMがプロセス障害による離脱を検知する
- 障害検知までの時間は数ミリ秒単位と非常に高速

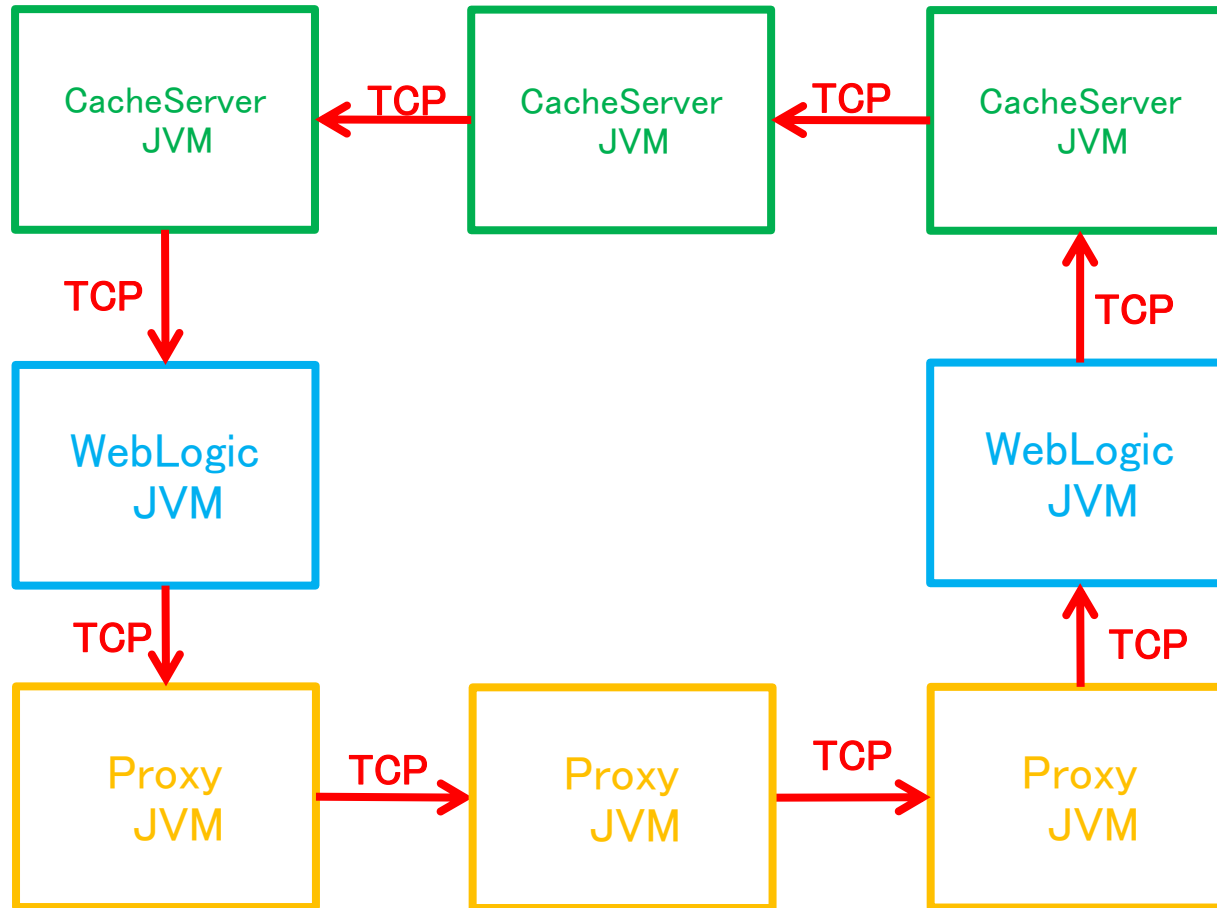


プロセス障害に即時対応するTCP Ring#2

- TCP Ringを説明するにあたり、Coherence JVMの識別情報であるMember IdentityのRoleに関する理解が必要
- Coherenceクラスタには様々なプロセスが参加するため、これらを種別ごとに分けるため利用されるのがMember Identity
- Member Identityの中でTcpRingの動作に関連する情報がRole
- Roleはその名の通り、プロセスの役割を記載するもので一般的にWebLoic、CacheServer、Proxy程度の大きなくりで設定する

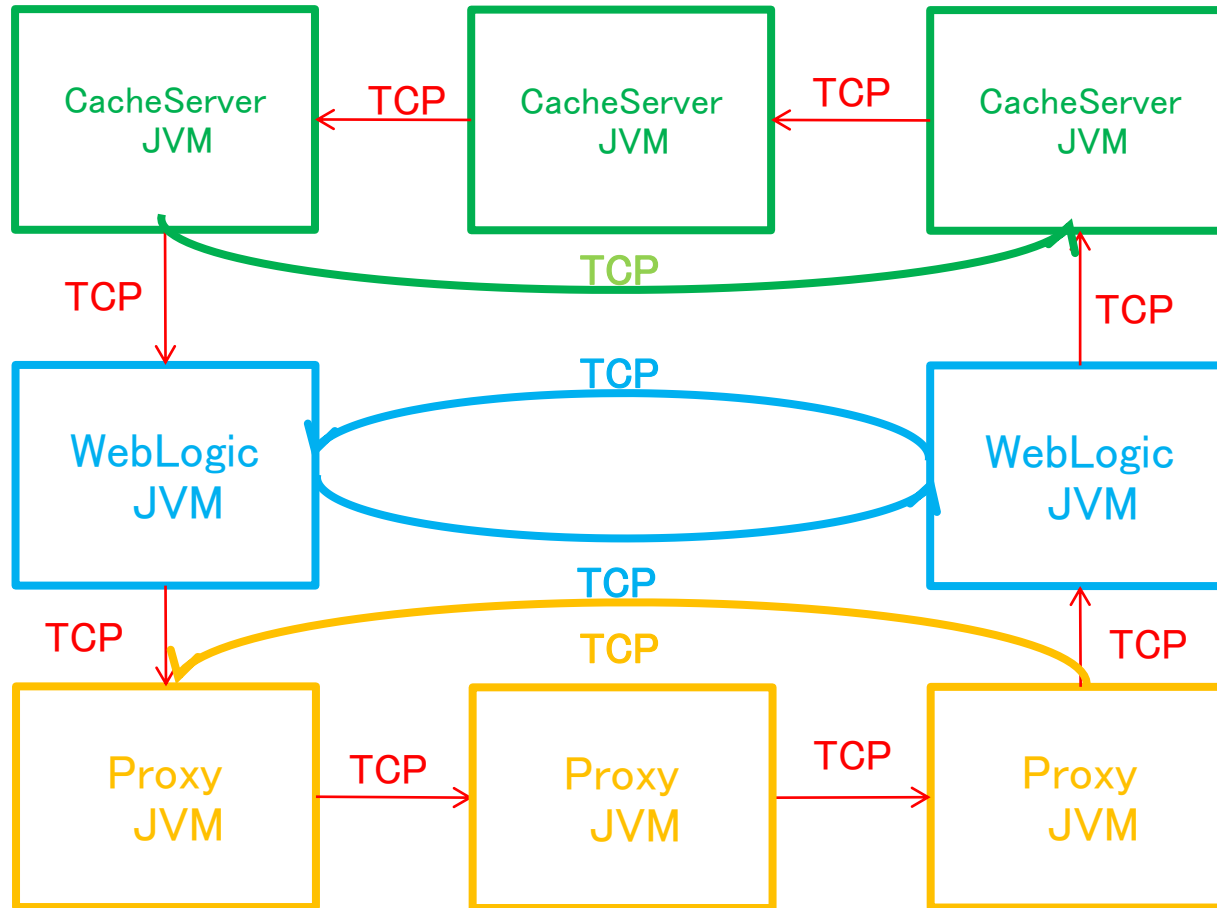
プロセス障害に即時対応するTCP Ring#3

- メインリングのつながり方は、すべてのプロセスを一方向に循環する形でコネクションが形成される



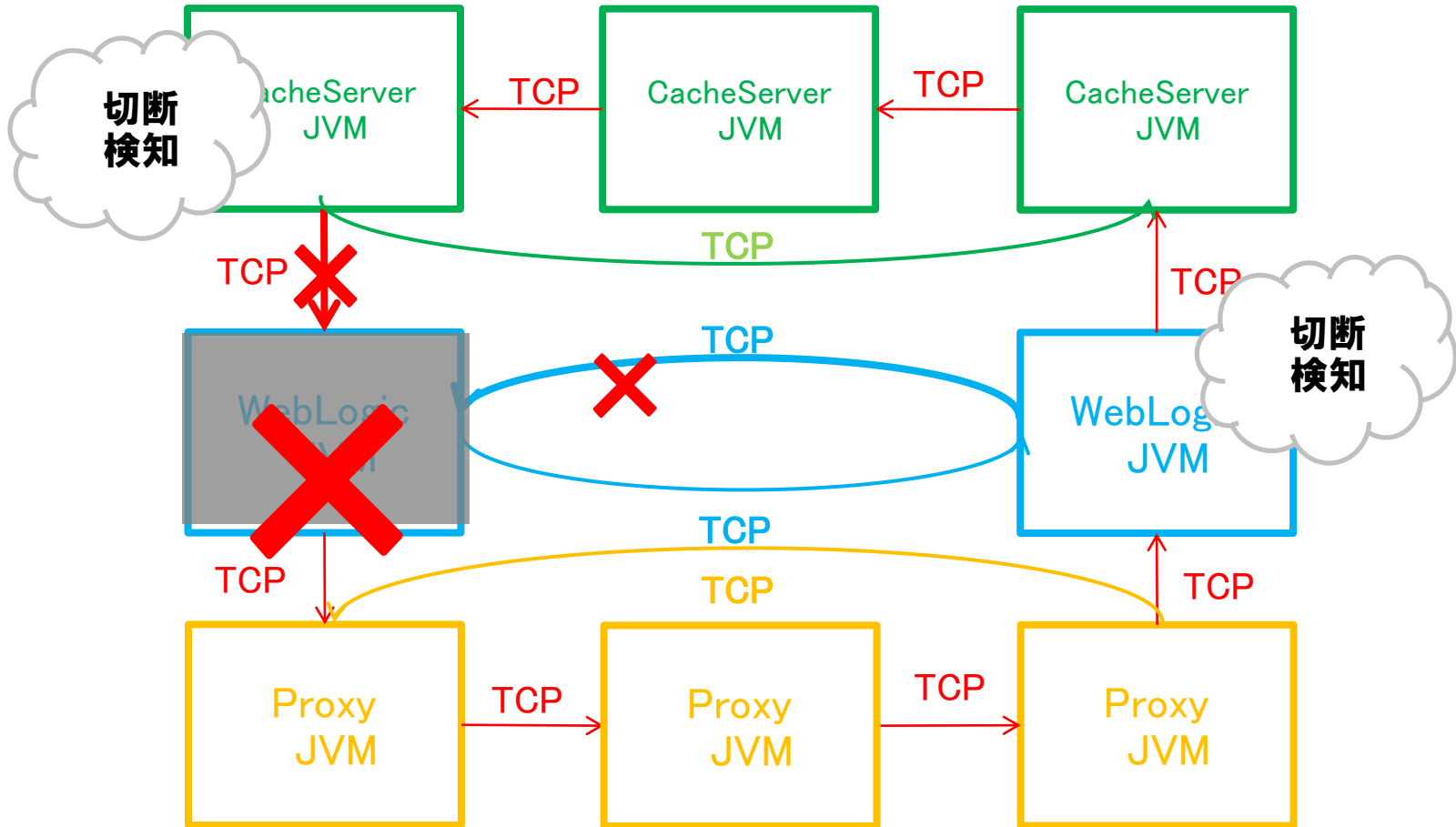
プロセス障害に即時対応するTCP Ring#4

- 補助リングのつながり方は、クラスタ内の同一ロールを一方行に循環するリングを作成する



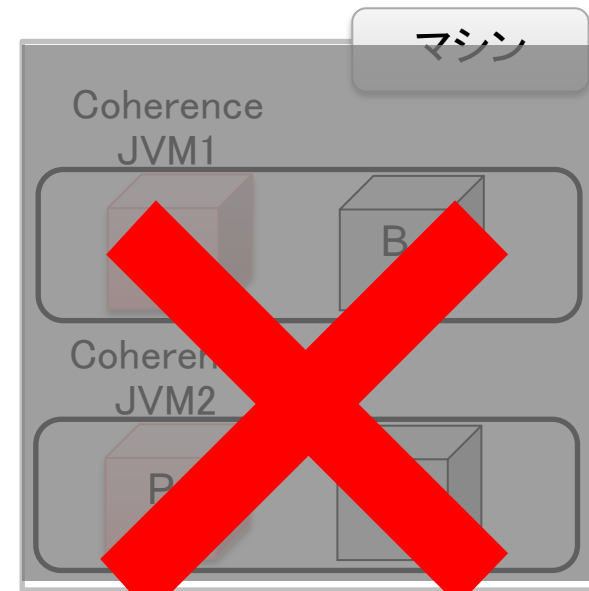
プロセス障害に即時対応するTCP Ring#5

- プロセス障害発生時には接続の切断が発生し、障害が検知されクラスタ全体に離脱が通知される



マシン障害、ネットワーク障害に対応するIP Monitor#1

- マシン障害、ネットワーク障害を高速に検知する機構
- プロセス障害と異なり、TCPコネクション切断によるイベント検知ができない。
 - マシン障害では、OSがTCPコネクションの切断処理を実施せず停止する
 - ネットワーク障害も同様に、通信ができなくなり明示的なTCPコネクションの切断処理ができない
- プロセスではなく、OSに対しpingを行い返答がなければ障害が発生したとみなし切り離す
 - pingはJDK1.5より組み込まれた `InetAddress#isReachable()` を利用
 - クラスタ情報より、マシンのリストを作成し、試行間隔ごとにマシンリストからランダムに抽出しpingを送る
- デフォルトの死活監視設定は15秒間の通信断絶検知し障害とみなす
 - タイムアウト時のリトライ3回
 - タイムアウト5秒
 - 試行間隔1秒

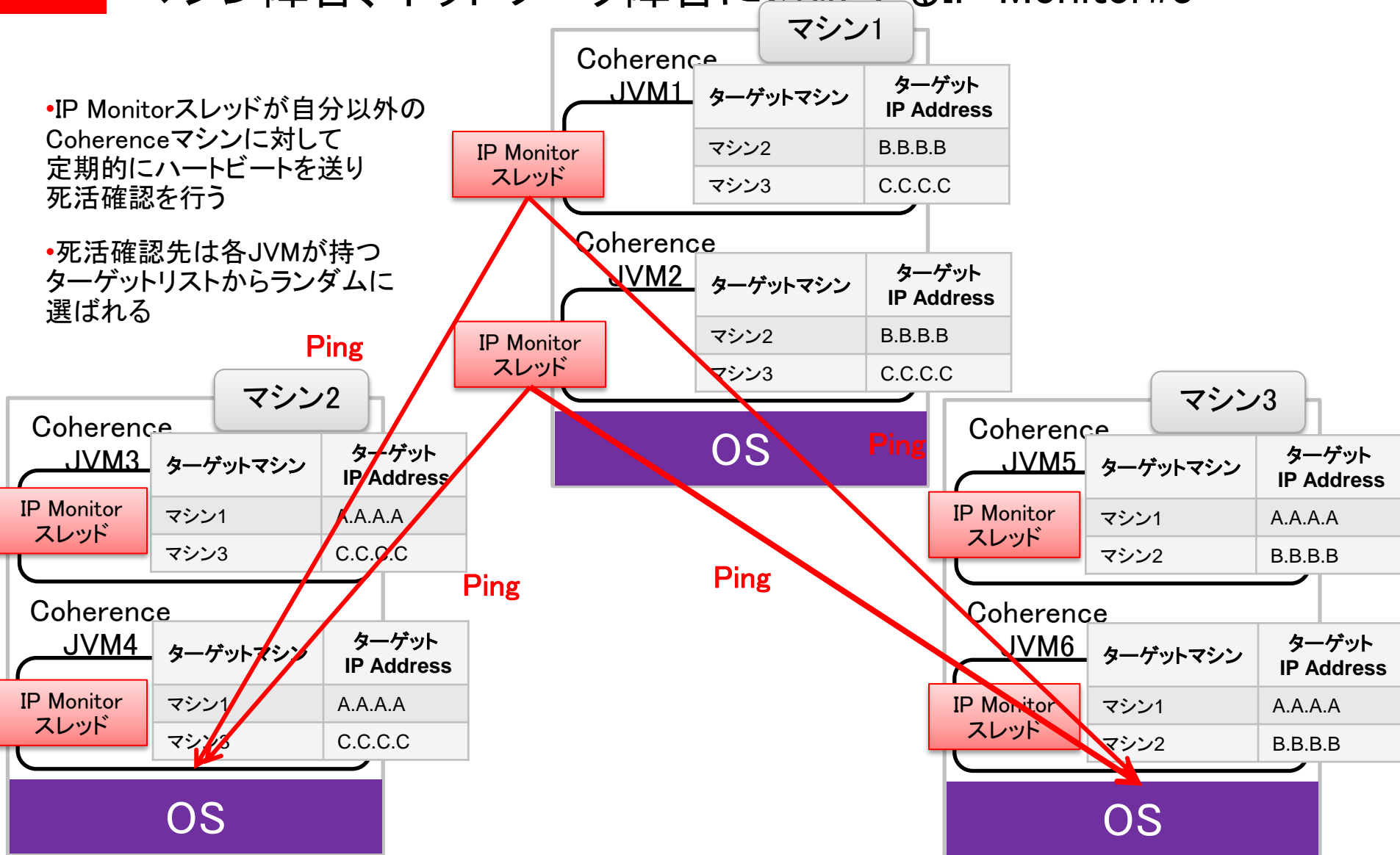


マシン障害、ネットワーク障害に対応するIP Monitor#2

- Pingの仕組みはJDKのisReachable()に依存することに注意が必要
- JVMを起動しているユーザの権限によって到達確認の仕組みが異なる
 - 特権ユーザ(root)→icmpパケットを利用した到達確認
 - 一般ユーザ→tcpのechoリクエストを利用した到達確認
- Coherence開発者に問い合わせたところ、到達確認方式の違いによる検知時間やOSに対しての到達確認という目的自体が変わるわけではないため、影響はないという認識
- ただし、死活確認のプロトコルが異なることからグリッドネットワークのセキュリティ設定などにより、死活確認が動作しない可能性があるため理解しておくことは重要

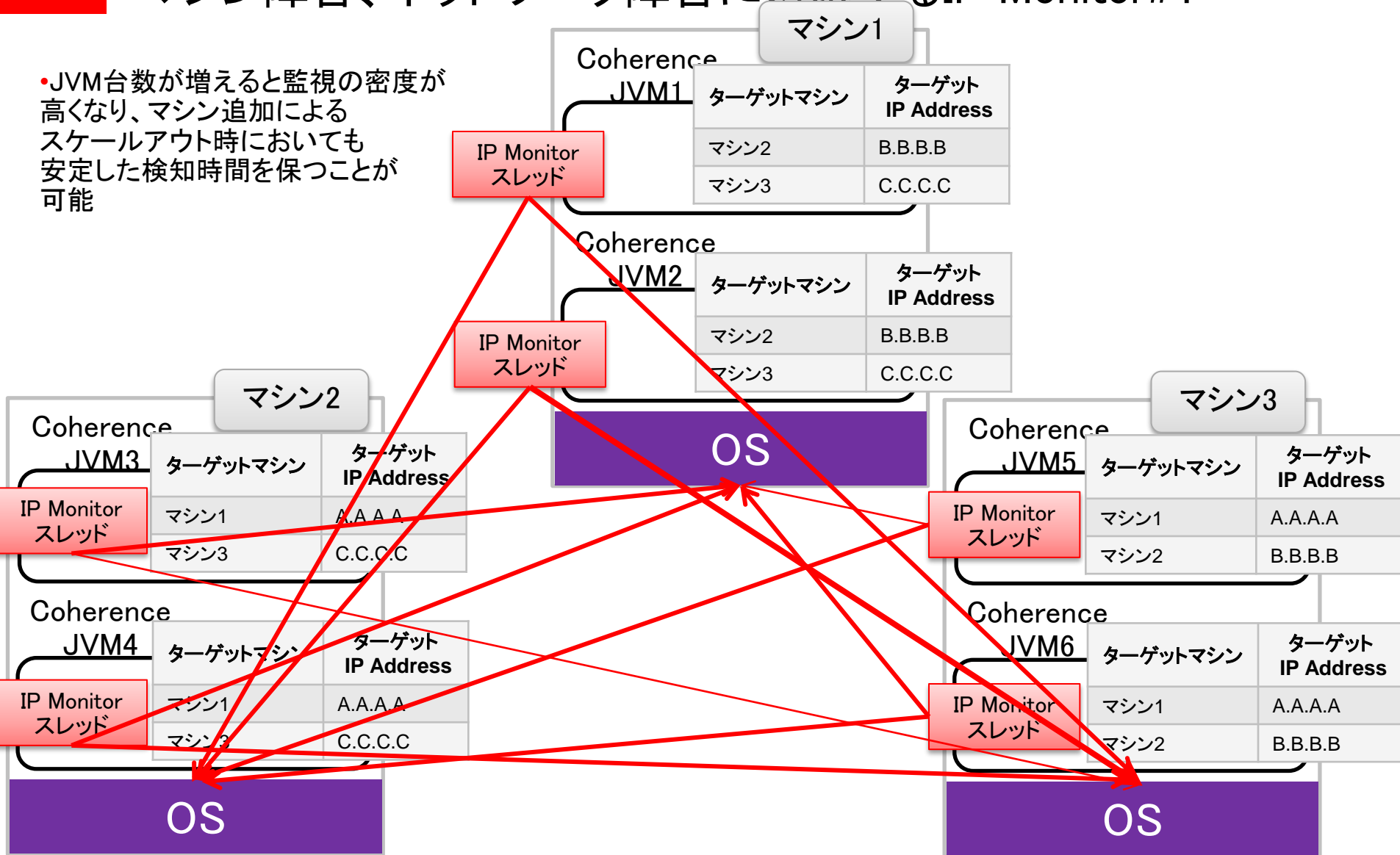
マシン障害、ネットワーク障害に対応するIP Monitor#3

- IP Monitorスレッドが自分以外のCoherenceマシンに対して定期的にハートビートを送り死活確認を行う
- 死活確認先は各JVMが持つターゲットリストからランダムに選ばれる



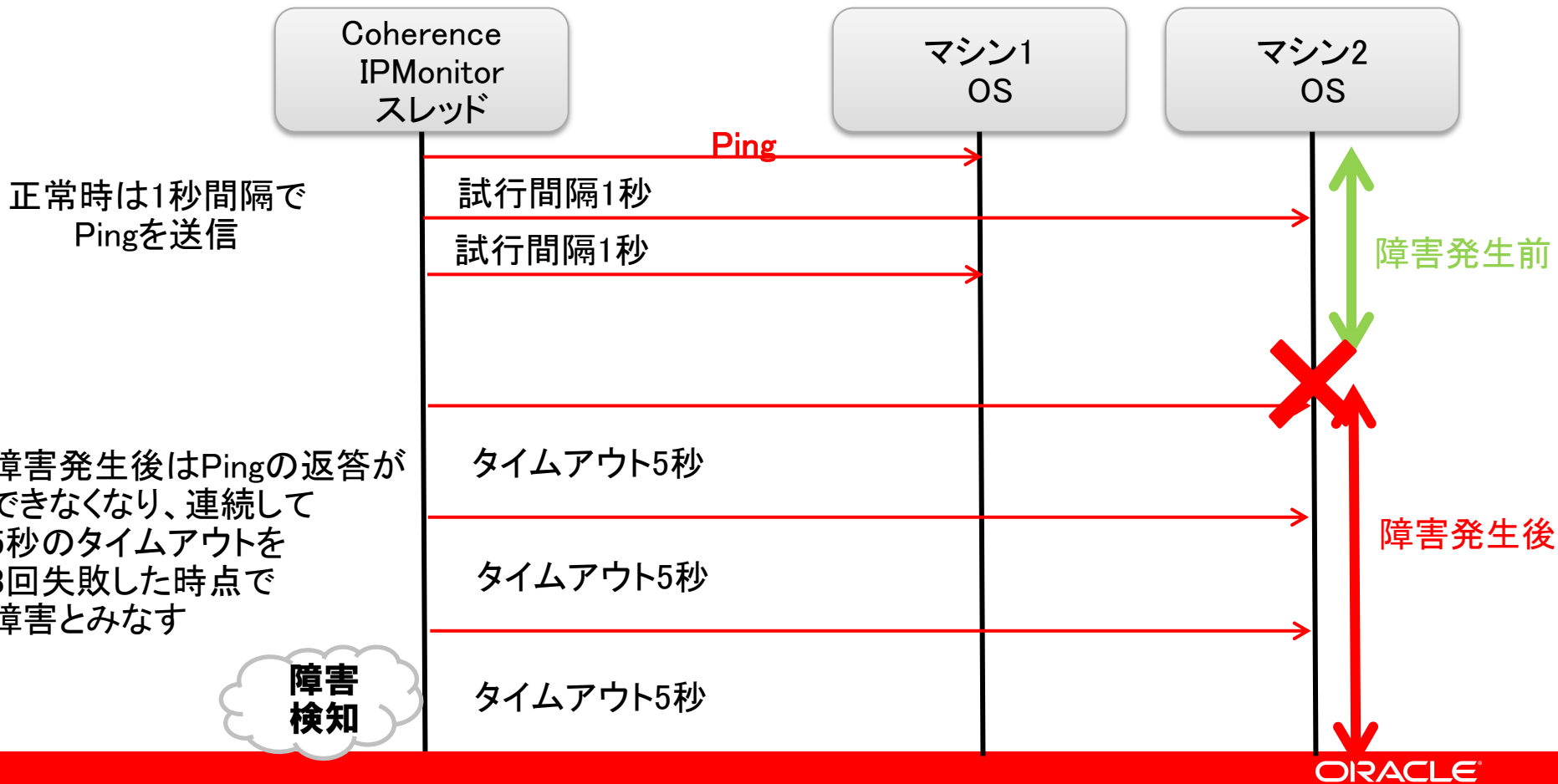
マシン障害、ネットワーク障害に対応するIP Monitor#4

- JVM台数が増えると監視の密度が高くなり、マシン追加によるスケールアウト時においても安定した検知時間を保つことが可能



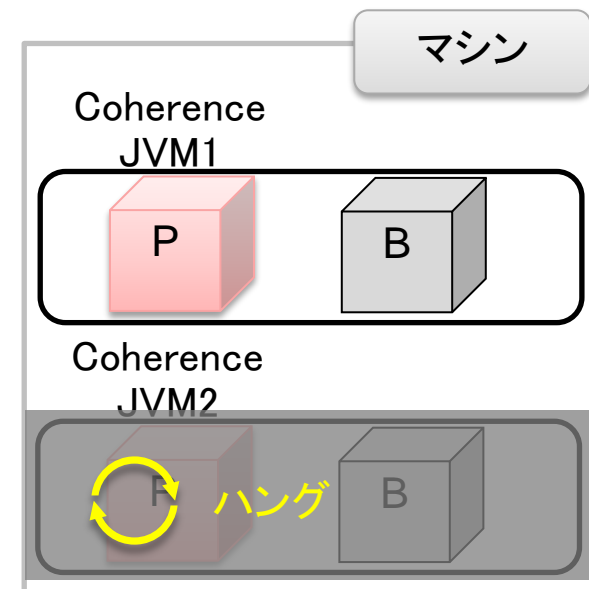
マシン障害、ネットワーク障害に対応するIP Monitor#5

- マシン障害を検知するまでの死活監視動作
 - タイムアウト時のリトライ3回
 - タイムアウト5秒
 - 試行間隔1秒

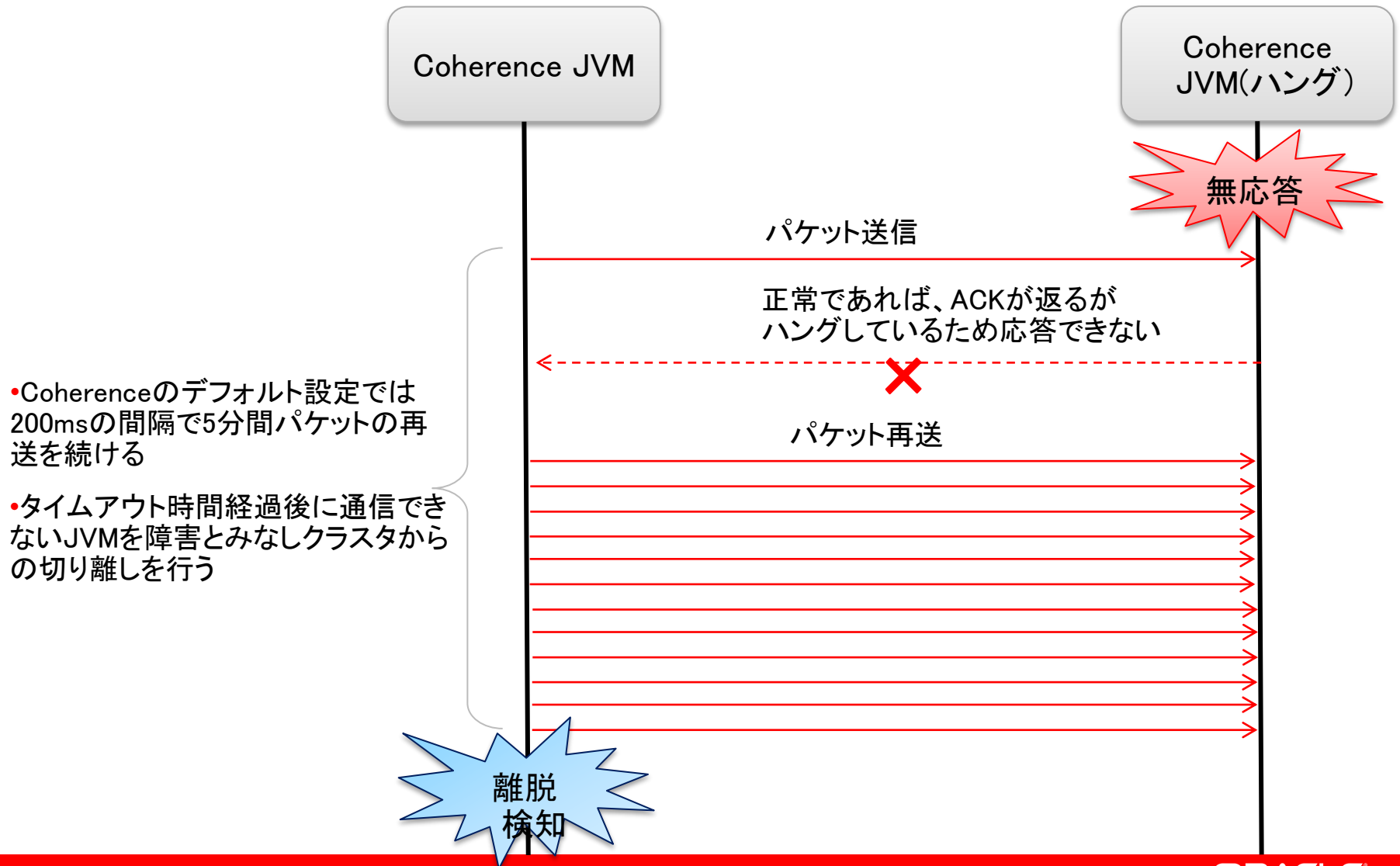


JVMハングに対応するパケットデリバリタイムアウト#1

- プロセス、マシンともに正常だが、JVMが応答を返せないハング状態を検知しクラスタから切り離す機構
- Coherenceはクラスタ通信を行うための専用プロトコルであるTCMP(Tangosol Coherence Management Protocol)を利用する
 - UDPベース
 - ユニキャスト、マルチキャストを状況によって使い分ける
 - パケットのAcknowledgementを独自実装
- 通信が一定時間途切れると対象のJVMをに異常が発生したとみなし、クラスタから切り離す
 - パケットデリバリタイムアウトで設定
 - デフォルトは5分



JVMハングに対応するパケットデリバリティタイムアウト#2



Coherence障害検知アーキテクチャまとめ

Conclusion...

Coherence障害検知アーキテクチャまとめ

- 障害ケースを網羅し、可能な限り早く検知するための検知機構を複数を持っている
- TCP Ringはプロセス停止にともなうTCPコネクション切断を検知し数ミリ秒単位での障害検知を行う
- マシン障害、ネットワーク障害に対してはIP Monitorを利用しPingを利用した死活監視を実施している
- マシン生存、プロセス生存、無応答という特殊なハング状況に陥ったとしても独自プロトコルによる切り離しが可能
- Oracle Coherenceはフェイルオーバー時間を可能な限り短くとどめ、サービスダウンタイムを最小化するための設計がなされている

OTNセミナーオンデマンド

コンテンツに対する
ご意見・ご感想を是非お寄せください。

OTNオンデマンド 感想



http://blogs.oracle.com/oracle4engineer/entry/otn_ondemand_questionnaire

上記に簡単なアンケート入力フォームをご用意しております。

セミナー講師/資料作成者にフィードバックし、
コンテンツのより一層の改善に役立てさせていただきます。

是非ご協力をよろしくお願いいたします。

OTNセミナーオンデマンド

日本オラクルのエンジニアが作成したセミナー資料・動画ダウンロードサイト

掲載コンテンツカテゴリ(一部抜粋)

Database 基礎

Database 現場テクニック

Database スペシャリストが語る

Java

WebLogic Server/アプリケーション・グリッド

EPM/BI 技術情報

サーバー

ストレージ



超入門! Oracle データベースって何
再生時間: 60分

100以上のコンテンツをログイン不要でダウンロードし放題

データベースからハードウェアまで充実のラインナップ

毎月、旬なトピックの新作コンテンツが続々登場

例えばこんな使い方

- 製品概要を効率的につかむ
- 基礎を体系的に学ぶ/学ばせる
- 時間や場所を選ばず(オンデマンド)に受講
- スマートフォンで通勤中にも受講可能



毎月チェック!



コンテンツ一覧 はこちら

<http://www.oracle.com/technetwork/jp/ondemand/index.html>

新作&おすすめコンテンツ情報 はこちら

<http://oracletech.jp/seminar/recommended/000073.html>

OTNオンデマンド



オラクルエンジニア通信

オラクル製品に関わるエンジニアの方のための技術情報サイト

オラクルエンジニア通信 - 技術資料、マニュアル、セミナー

Oracleエンジニアのための技術情報サイト by Oracle Japan

新着情報を知りたい

技術資料を探したい

セミナーを受けたい

About

Oracleエンジニアの方がスキルアップしていただくために、厳選した情報をお届けしています

技術資料	<p>インストールガイド・設定チュートリアルetc. 欲しい資料への最短ルート</p>	アクセスランキング	<p>他のエンジニアは何を見ているのか？人気資料のランキングは毎月更新</p>
特集テーマ Pick UP	<p>性能管理やチューニングなど月間テーマを掘り下げて詳細にご説明</p>	技術コラム	<p>SQLスクリプト、索引メンテナンスetc. 当たり前運用/機能が見違える!?</p>

<http://blogs.oracle.com/oracle4engineer/>

オラクルエンジニア通信



The screenshot shows the top navigation bar of the oracletech.jp website. It features the 'oracletech.jp' logo in red and black, with the tagline '好奇心が、エンジニア人生を豊かにする。' below it. To the right is the 'ORACLE' logo, a search bar, and social media icons for Twitter, Facebook, LinkedIn, YouTube, and RSS. Below these are five red navigation buttons: '製品/技術情報', 'スキルアップ', 'セミナー', 'キャンペーン', and 'ちょっと一息'.

製品/技術
情報



Oracle Databaseっていくら？オプション機能も見積れる簡単ツールが大活躍

セミナー



基礎から最新技術までお勧めセミナーで自分にあった学習方法が見つかる

スキルアップ



ORACLE MASTER ! 試験頻出分野の模擬問題と解説を好評連載中

Viva!
Developer



全国で活躍しているエンジニアにスポットライト。きらりと輝くスキルと視点を盗もう

<http://oracletech.jp/>

oracletech



あなたにいちばん近いオラクル



Oracle Direct

まずはお問合せください

Oracle Direct



システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。
システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。
http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28

※フォームの入力にはログインが必要となります。
※こちらから詳細確認のお電話を差し上げる場合がありますので
ご登録の連絡先が最新のものになっているかご確認下さい。

フリーダイヤル

0120-155-096

※月曜～金曜
9:00～12:00、13:00～18:00
(祝日および年末年始除く)

ORACLE

Hardware and Software **Engineered to Work Together**

ORACLE®