

高可用システムの 構築って何？

Linux-HA Japan Project
今崎 充智





本日のアジェンダ

- RASって知っています？
- 可用性向上の方策
- ロードバランサーって？
- クラスタリング
- Pacemakerの歴史
- Linux-HA Japanの紹介



可用性(Availability)とは

- ◎ システムが継続して稼働できる能力
 - ◎ 利用者から見て「使用できる」度合い
 - 信頼性 (Reliability)
 - 保守性 (Serviceability)
- と併せて、RASと呼ばれ、システムの性能を総合的に示す指標の一つである。

稼働率とは

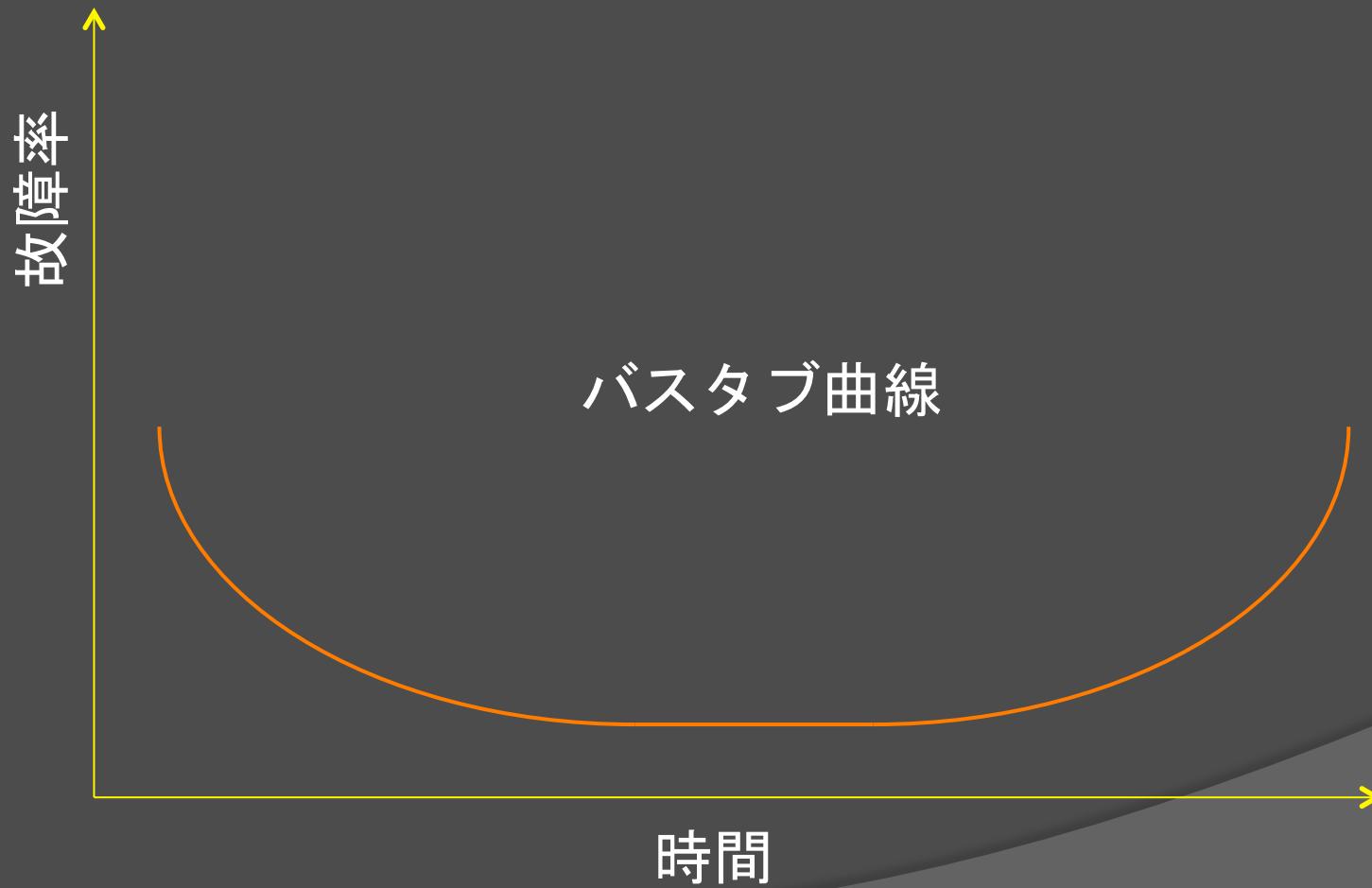
- 稼働率とは、修理可能な系・機器・部品などが、ある特定の瞬間に機能を維持している確率（瞬間稼働率）、または規定の時間で機能を維持している確率（平均稼働率）のこと。
- 稼働率Aと平均故障間隔（MTBF）、平均復旧時間（MTTR）の関係は以下のようになる。

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

MTBF: 信頼性
MTTR: 保守性



故障率曲線





稼働率と停止可能時間

稼働率	停止可能時間／年
99%	3日15時間36分
99.9%	8時間46分
99.99% (フォーナイン)	52分34秒
99.999% (ファイブナイン)	5分15秒
99.9999% (シックスナイン)	32秒

たとえば

- ◎ とある銀行のオンラインバンキング
 - ・ 毎週日曜日21時から翌月曜日7時は停止。
→10/168は停止。 稼働率94%

- ◎ とあるインターネットショッピング
 - ・ 毎週日曜日2時～9時は停止。
→7/168は停止。 稼働率96%

情報システムの信頼性向上に関するガイドライン

<http://www.meti.go.jp/press/20060615002/20060615002.html>



情報システムの信頼性向上に関するガイドライン概要

基本的考え方

- 情報システム利用者及び情報システム供給者の**双方が応分の責務**を担う
 - ・利用者：業務・サービスの企画発意・機能維持
 - ・供給者：システム供給に對し最大限の努力と多面的な恒常的取組
- 利用者及び供給者の**経営層は、説明責任を認識**し、必要な経営資源の投入等に対して責務を担う。
- 未然防止と事後対策の**両側面からの対策**が必要。

(注)求められる信頼性・安全性の水準に応じ、情報システムを3段階に分類
(A) 重要インフラ等システム
(B) 企業基幹システム
(C) その他のシステム

具体的な対策

1. 企画・開発及び保守・運用全体における事項

- 【企画・開発】利用者・供給者双方は、信頼性・安全性の水準を検討し、仕様に取り込む。
- 【保守・運用】情報システム障害発生時の対応手順を文書化し、合意。障害の内容・原因等を記録。
- 【全体横断】定量的手法を取り入れたプロジェクトマネジメントを実行。等

2. 技術に関する事項

- 【手法・ツール活用】人手による誤りの排除等のためにモデル化言語、形式手法等を活用。
- 【基本構造の確立】将来の拡張性、障害の影響の最小化等を考慮。等

3. 人・組織に関する事項

- 【人材育成】情報処理技術者試験及びITスキル標準等を活用。
- 【組織整備】障害発生時の経営層まで含めた緊急体制を整備。等

4. 商慣行・契約・法的要素に関する事項

- 【契約】利用者・供給者双方の役割分担・責任関係を合意し、契約において明記。
- 【契約】情報システム構築の分業時の役割分担・責任関係を合意し、契約において明記。等

実効性に関する担保措置

1. モデル契約の策定・活用

- ・利用者団体・供給者団体が協力して、本ガイドラインの考え方を反映させた標準的な契約のあり方を検討。

2. 政府調達における活用

- ・経済産業省は、本ガイドラインの内容を積極的に調達に活用。
- ・政府調達における本ガイドラインの活用方策を検討。

3. 診断(ベンチマーク)方法の整備

- ・経済産業省及びIPAは、本ガイドラインの内容に沿って、利用者及び供給者両者に対する情報システムの開発及び運用状況の診断システム(ベンチマー킹)の方法を整備。

情報システムの信頼性向上に関するガイドライン

<http://www.meti.go.jp/press/20060615002/20060615002.html>



信頼性・安全性向上に向けた方向性

システム障害・影響拡大の原因及び背景

直接的原因	要件の誤り
	ソフトウェアの誤り
	ハードウェア故障・性能低下等
	製品間インターフェイスの誤り
	性能・容量等の不足
	運用方法・手順等の誤り
	障害発生時の対応の誤り・遅れ
間接的原因／背景	工数・工期・コスト見積りミス
	プロジェクト管理ミス
	開発・運用体制不備
	緊急時の体制不備
	人員のスキル不足

改善すべき要素

システム障害・影響拡大の原因は、複数の要素に分けて分析が可能

全般的配慮事項

企画・開発及び保守・運用全体における事項

技術に関する事項

人・組織に関する事項

商慣行・契約・法的要素に関する事項

各要素を改善・見直しすることで、障害の防止を狙う



システムが停止する要因

- HW故障
- メンテナンス
- バッチ処理
- AP不具合
- 自然災害
- 人為故障



信頼性を高める工夫

◎ 装置単体の信頼性の向上

- メモリ、IF、電源等の冗長化
- RAID

◎ 複数の装置による信頼性の向上

- 負荷分散
- クラスタリング

フォールトレジストリシステム

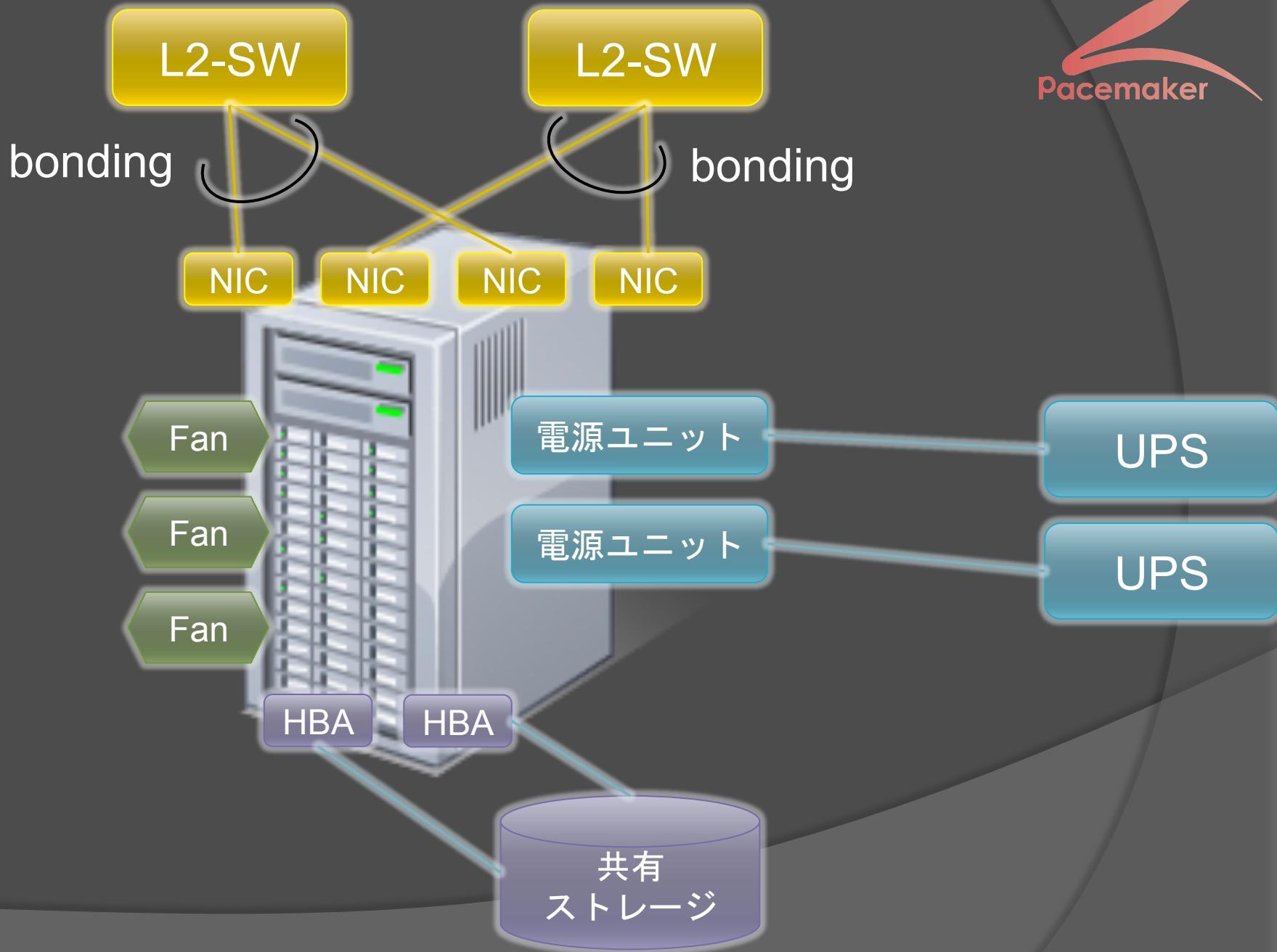


- ◎ 装置内部で二重化等をすることにより、信頼性を高めているシステム。
(デュアルシステム)
 - メインフレーム
 - ストラタスftサーバー
 - HP Non Stop Server (旧タンデム)
- ◎ 故障時に内部で切替、部品交換を行う。



パートの冗長化

- ◎ HDD／メモリ／NW-IF／電源／ファン
 - HDD等 RAID、ミラー
 - メモリ ミラー、チップキル、ECC
 - NW-IF 2重化
 - ストレージIF 2重化
 - 電源 N+1
 - ファン N+1





RAIDとは？

- 複数台の記憶媒体を組み合わせて、仮想的な1台の記憶媒体として運用する技術
- ハードウェアRAIDとソフトウェアRAIDに大別。
- RAIDレベルにより機能が異なる。性能向上、対障害性など。

RAID0



RAID1



RAID0+1



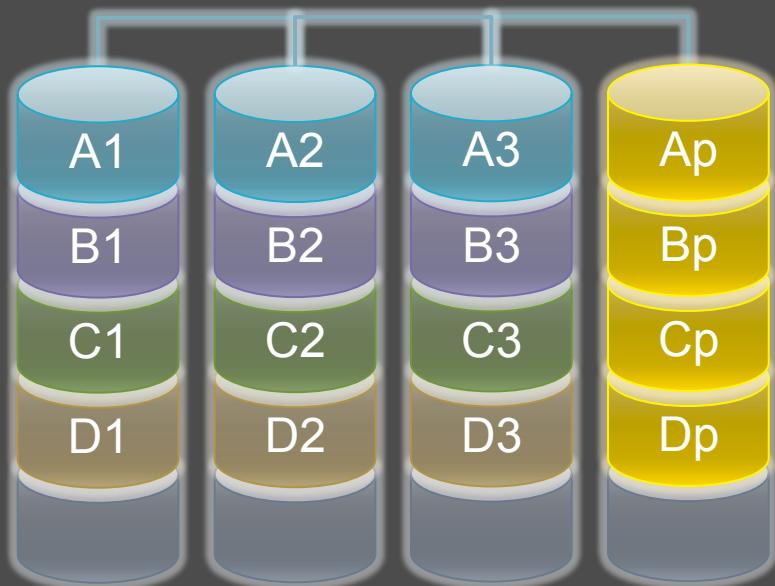
RAID1+0 (RAID10)



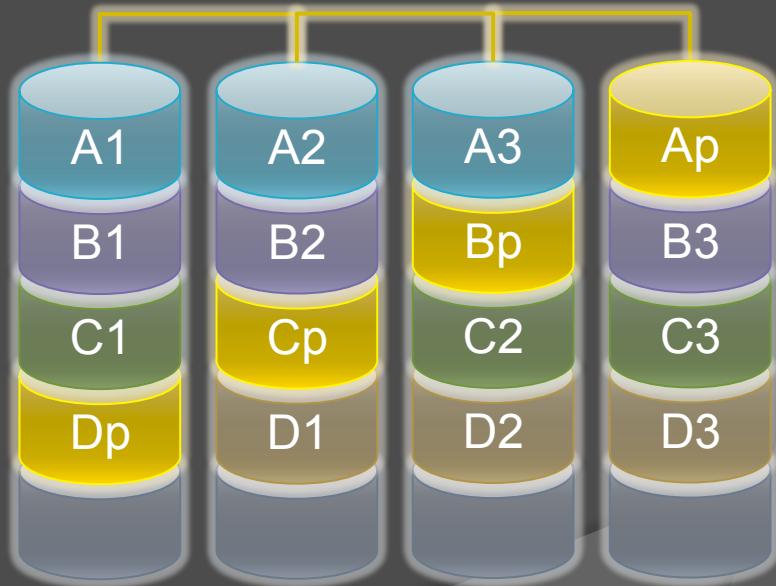
2台故障時のデータ破壊の可能性 2/3

2台故障時のデータ破壊の可能性 1/3

RAID4

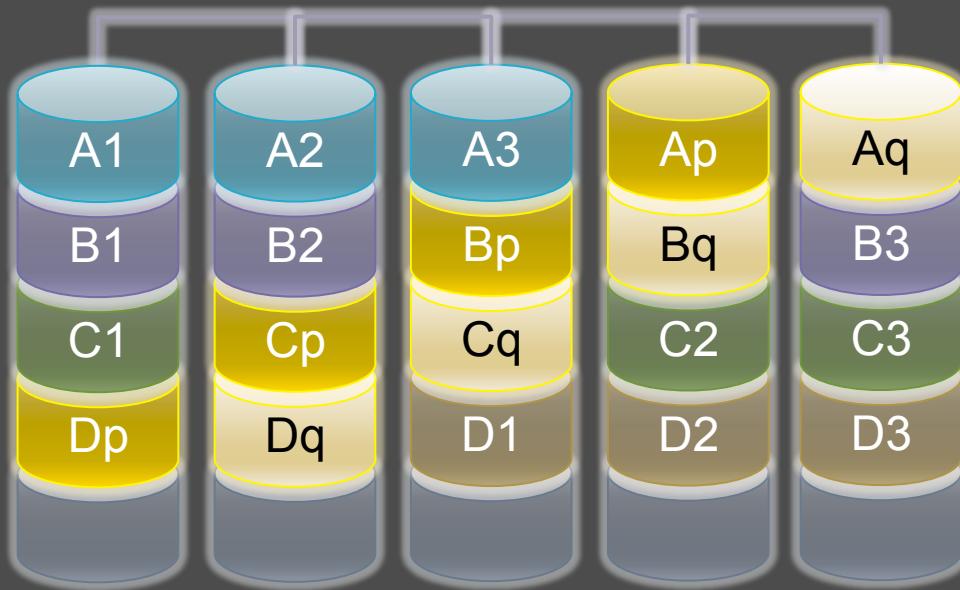


RAID5



RAID 2,3は省略

RAID6



RAID-Zは省略するのだ



RAID まとめ

RAIDレベル	信頼性	書込速度	容量効率
RAID0	×	○	6/6
RAID1	◎	○	3/6
RAID10(1+0)	◎	○	3/6
RAID0+1	△	○	3/6
RAID4	○	△	5/6
RAID5	○	△	5/6
RAID6	◎	△	4/6

LVMの仕組み

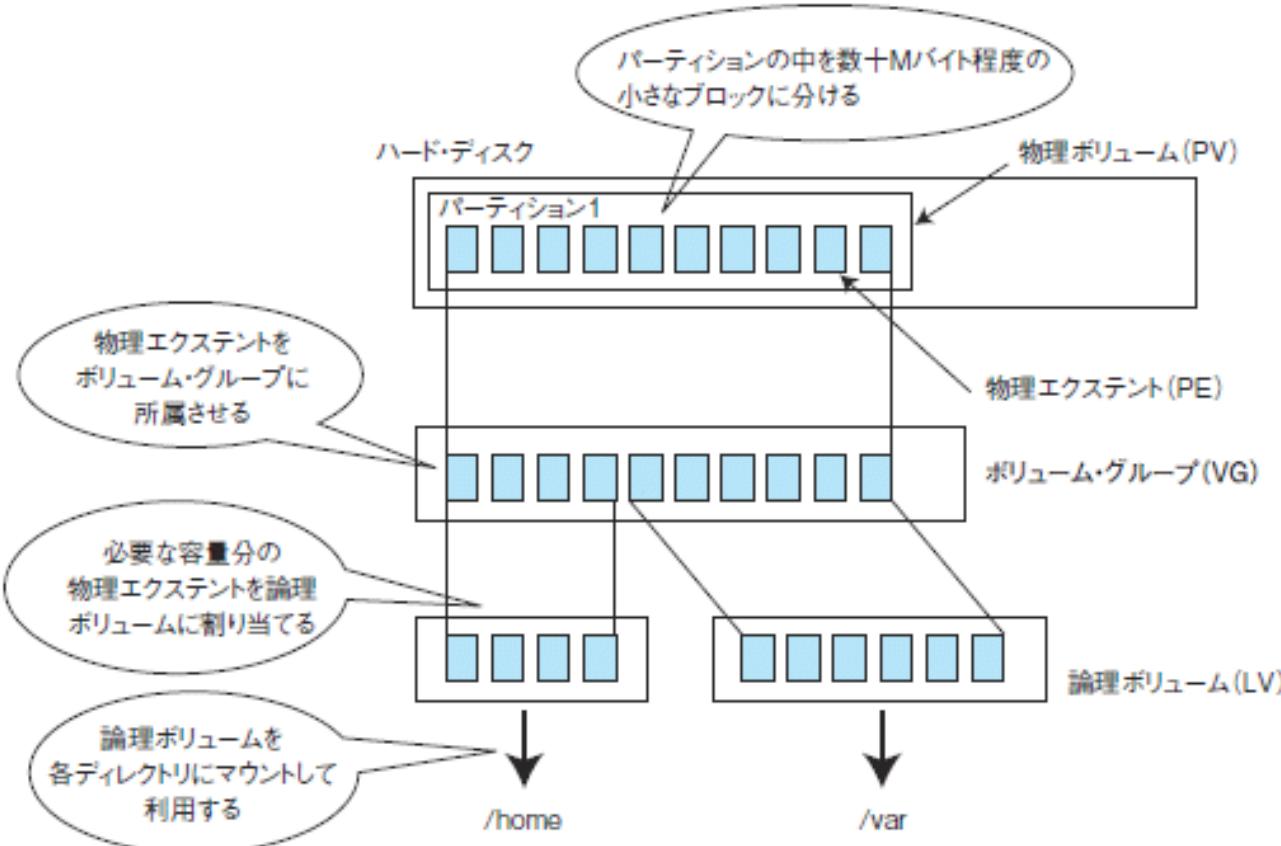
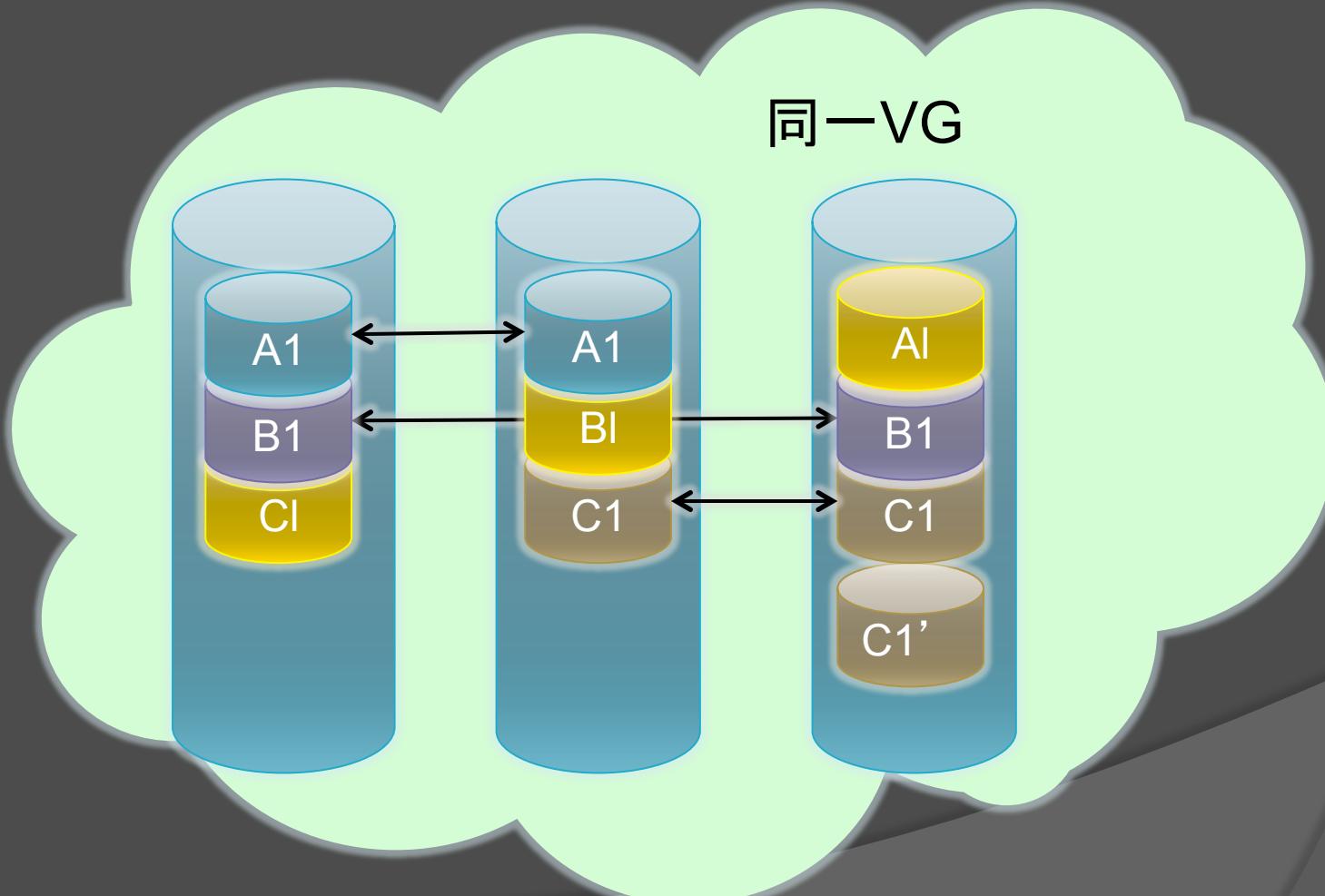


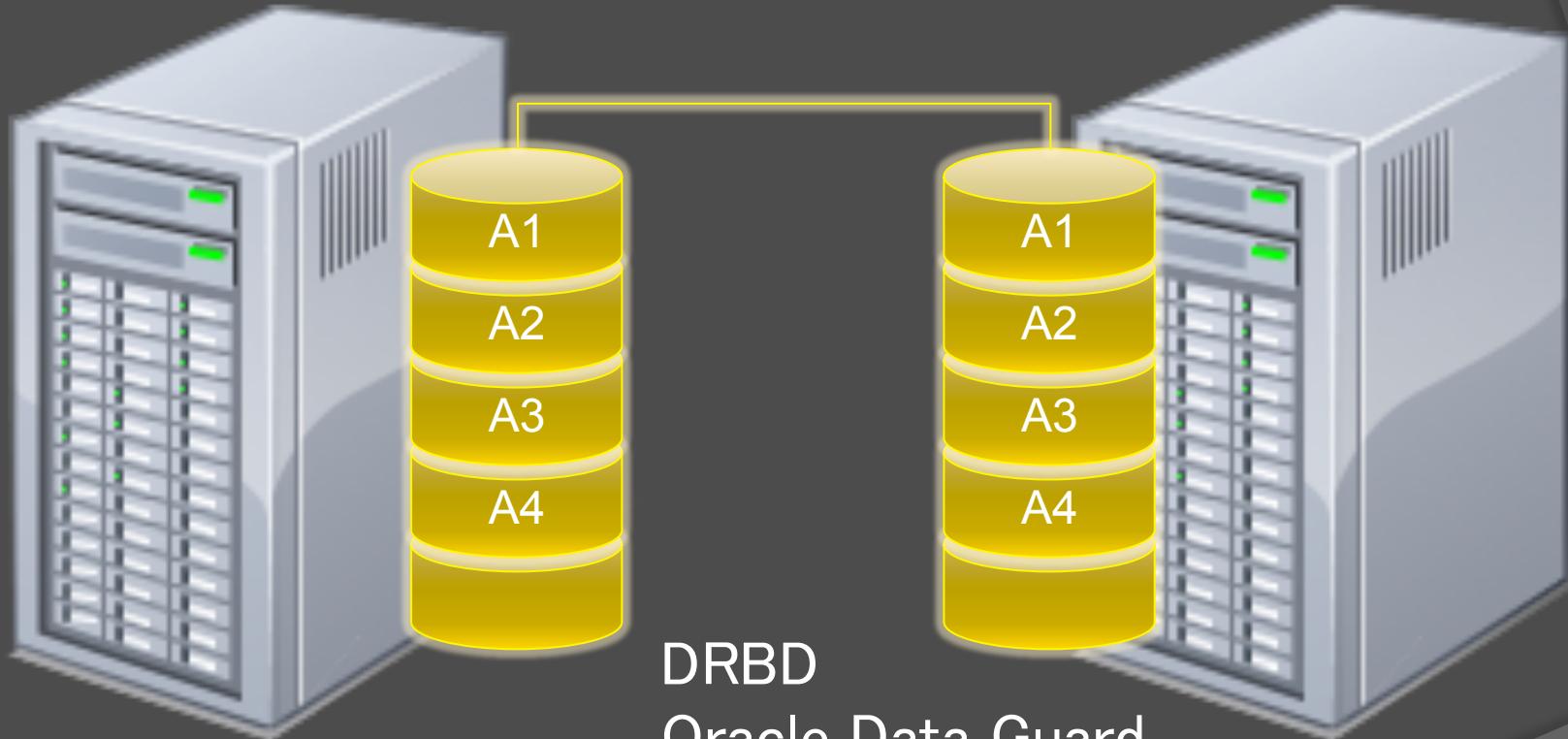
図 LVMの仕組み

「物理エクステント」，「物理ボリューム」，「ボリューム・グループ」，「論理ボリューム」という単位でディスクを管理する。

LVMミラー、スナップショット



複数にまたがるレプリケーション



DRBD

Oracle Data Guard

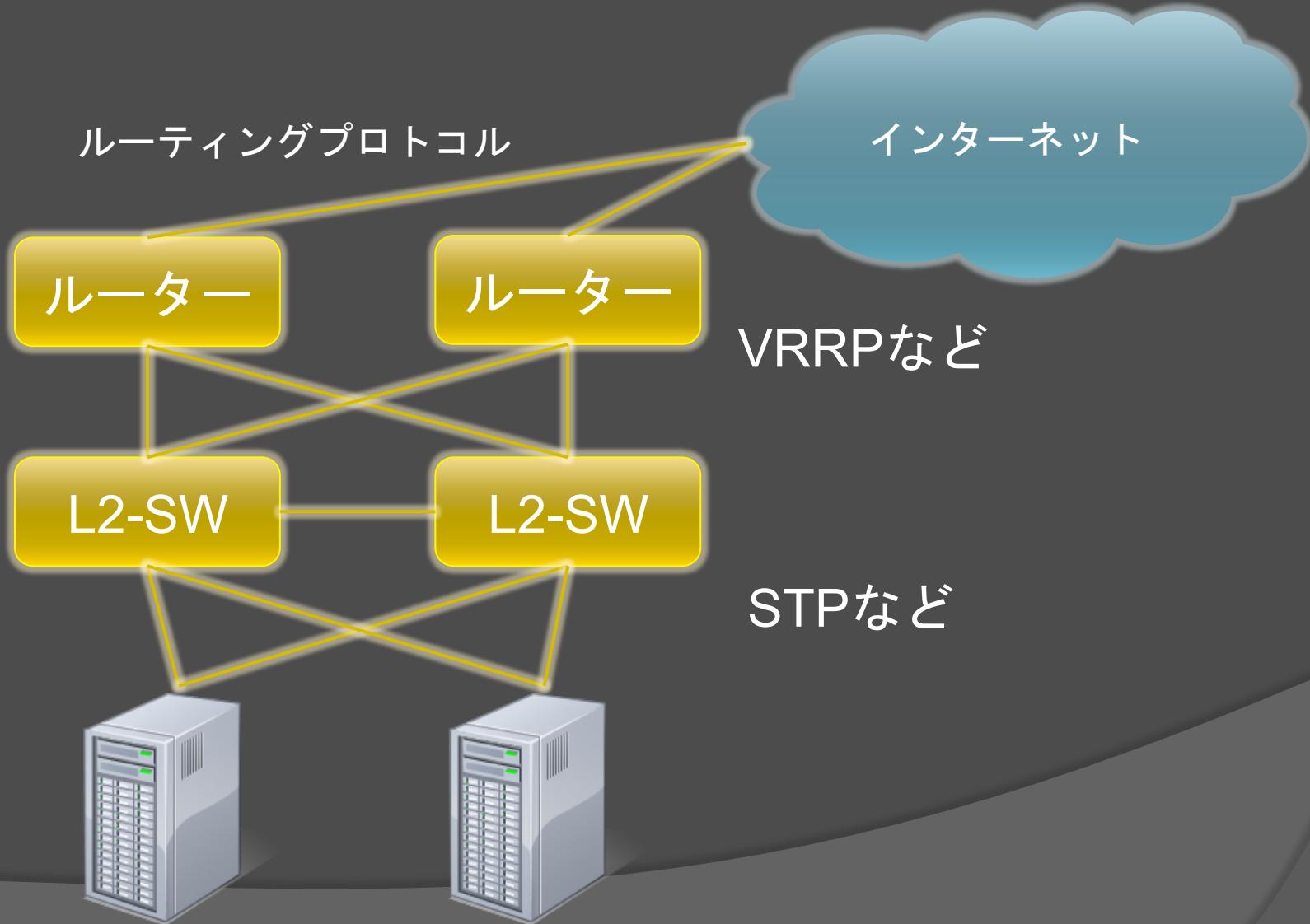
Veritas Volume Replicator

SteelEye Data Keeper

Cluster Pro X Replicator

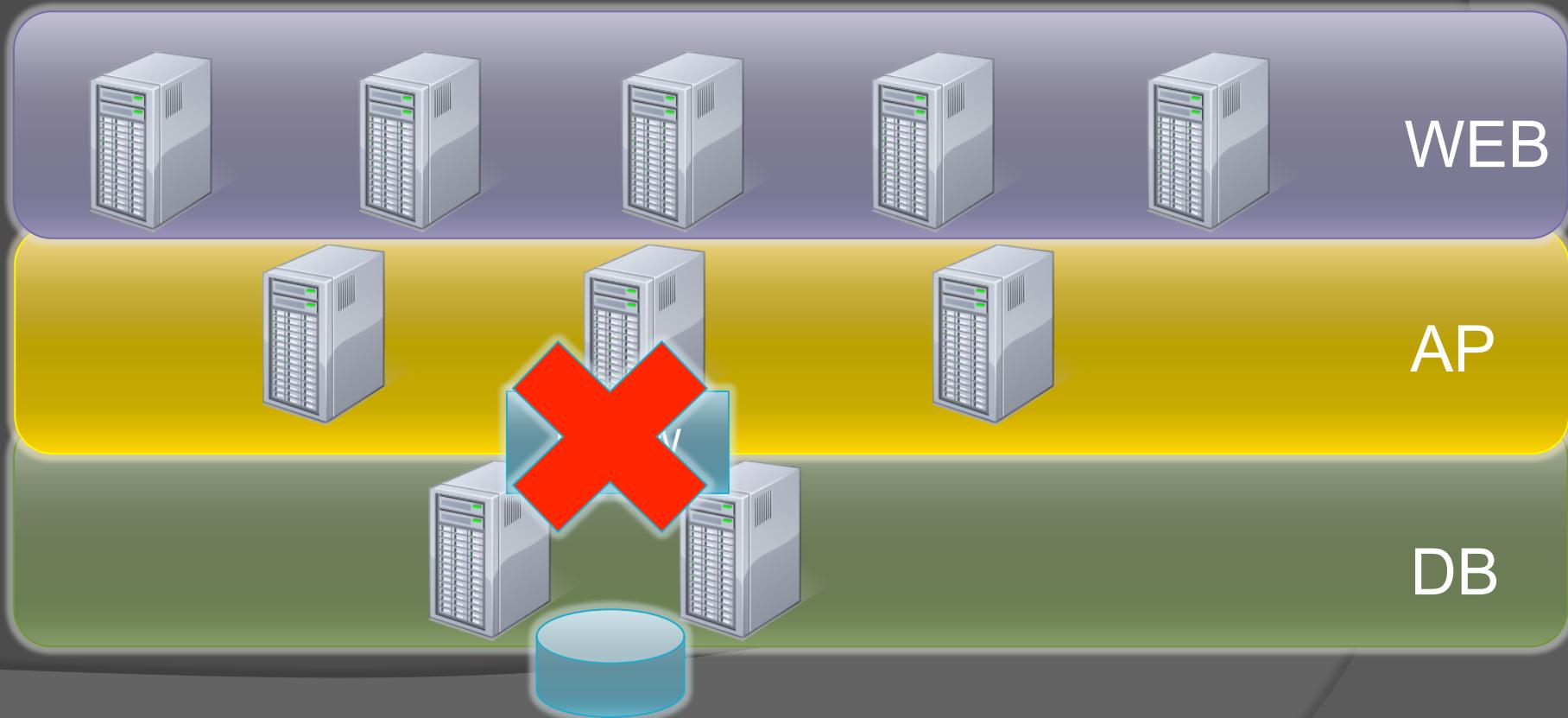
...

ネットワークの冗長



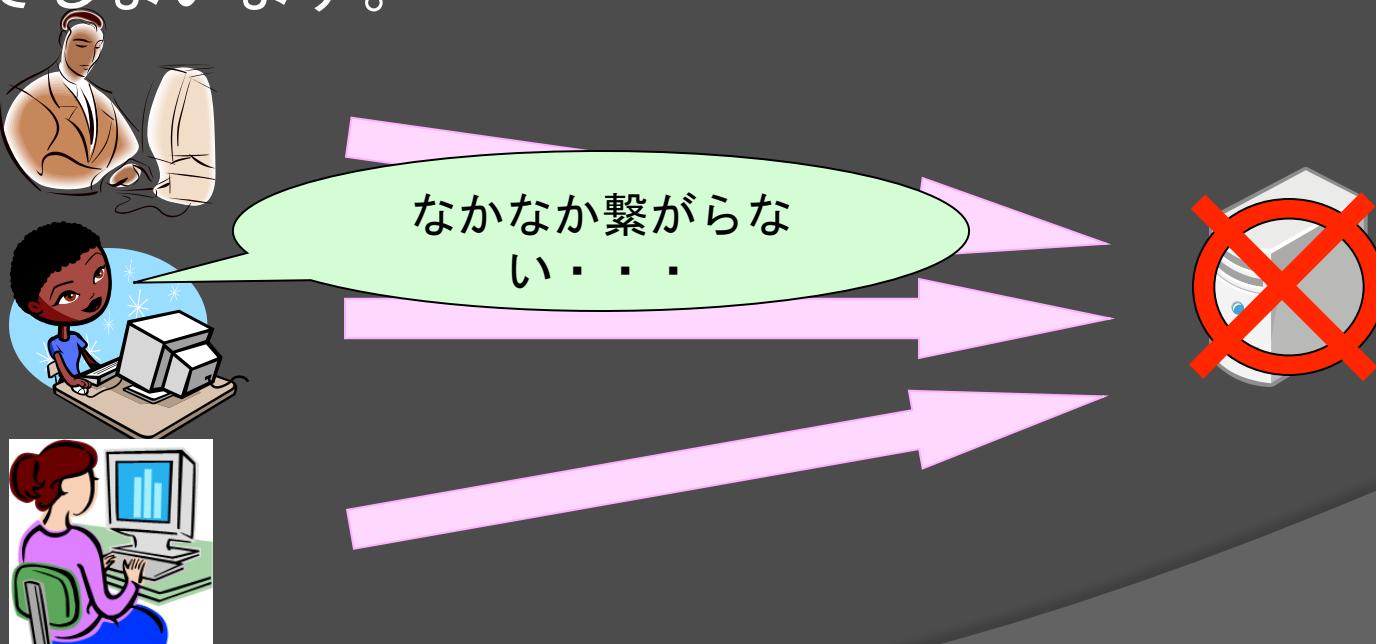
SPOFとは？

- ◎ Single Point of failure(单一故障点)



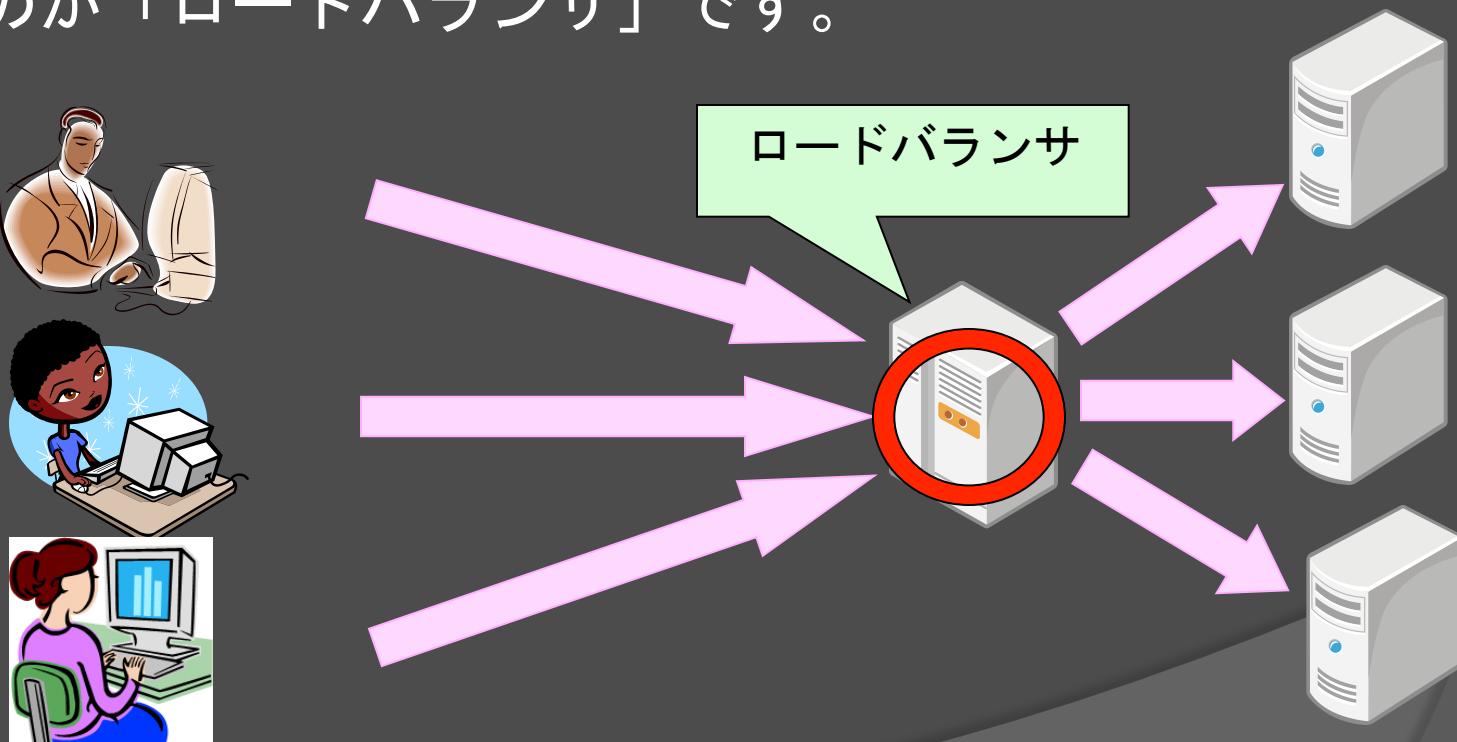
ロードバランサって何？

- たくさんのお客様が同じWebサイトを閲覧し、リクエストが予想以上に増加すると、Webサイトの応答が遅くなったり、接続しにくくなったりして、サービスが低下してしまいます。



ロードバランサって何？

- このような事態を防ぐため、サーバの負荷を適切に分散させ、リクエストが増えても対応できる仕組みを提供するのが「ロードバランサ」です。





ロードバランサーとは？

- ◎ Webサーバーなどに対するアクセス要求を管理し、同等の機能を持つ複数のサーバーにアクセスを分散する働きを持つ。
- ◎ ダウンしたサーバーへのリクエストを停止する役割もあり、システムの可用性向上のために用いられることがある。



主なロードバランサ製品

◎ ハードウェア

- F5 BIG-IP
- A10ネットワーク SoftAX
- CISCO

など

◎ ソフトウェア

- LVS / UltraMonkey
- Pound / HA Proxy / Piranha
- Apache / nginx
- UltraMonkey-L7

Ultra Monkey

Load Balancing and High Availability Solution

[English](#) | [Japanese](#)



[Top](#) | [About](#) | [Mirrors](#) | [History](#) | [Contacts](#) | [Ultra Monkey 3](#) | [Ultra Monkey 2.0.1 \(Old\)](#)
[News Archive](#) | [Papers](#) | [Ultra Monkey L7 \[Japanese\]](#)

About

Ultra Monkey is a project to create load balanced and highly available network services. For example a cluster of web servers that appear as a single web server to end-users. The service may be for end-users across the world connected via the internet, or for enterprise users connected via an intranet.

Ultra Monkey makes use of the Linux operating system to provide a flexible solution that can be tailored to a wide range of needs. From small clusters of only two nodes to large systems serving thousands of connections per second.

SOURCEFORGE.JP オープンソース・ソフトウェアの開発とダウンロード

ログイン アカウ

“起動がものすごく速い。何を検索していてもぜんぶ速い。”
プログラマー 28歳
グーグルがつくった速いブラウザ Google™クローム

MY SF.JP ソフトウェアを探す Magazine キャリア ビジネス 開発 ブログ

今後のLotus Notesのアプリ運営どうしますか？クラウドに移行するなら今

SourceForge.JP > ソフトウェアを探す > UltraMonkey-L7 > 概要

UltraMonkey-L7

UltraMonkey-L7は、OSI7階層モデルの第4層（Layer4）までの情報に基づいた負荷分散ソリューションである従来のUltraMonkeyを応用して、第7層（Layer7）までの情報に基づいた負荷分散機能を実現するためのプロジェクトです。ユーザの利便等を考慮し、本家UltraMonkeyプロジェクトのリーダであるSimon Horman氏ご了解のもとUltraMonkey(L4)パッケージも合わせて提供しております。

[UltraMonkey-L7の詳細情報へ](#)

[UltraMonkey-L7 のインストール方法](#)

[UltraMonkey-L7 の使い方](#)

最終更新日: 2011-06-30 11:44

開発メンバー: [hibari](#), [jsugiura](#), [kondoh86](#),
[sktateish](#), [shin_kusanagi](#),
[suigintoh](#), [tanuma](#), [toreno4257](#),
[fukushima](#), [h-okada](#), [hiroakinakano](#),
[mkurebayashi](#), 他4名 [[一覧](#)]

[その他の情報](#)

開発者向けページ [Home](#)

No Image Available

[他の画像を見る]

このプロジェクトは星いくつ?



[ツイートする](#)

5

[いいね!](#)

+1

0

UltraMonkeyは
Ldirectord (Linux-HA)と
LVSから構成される
L4LBです。

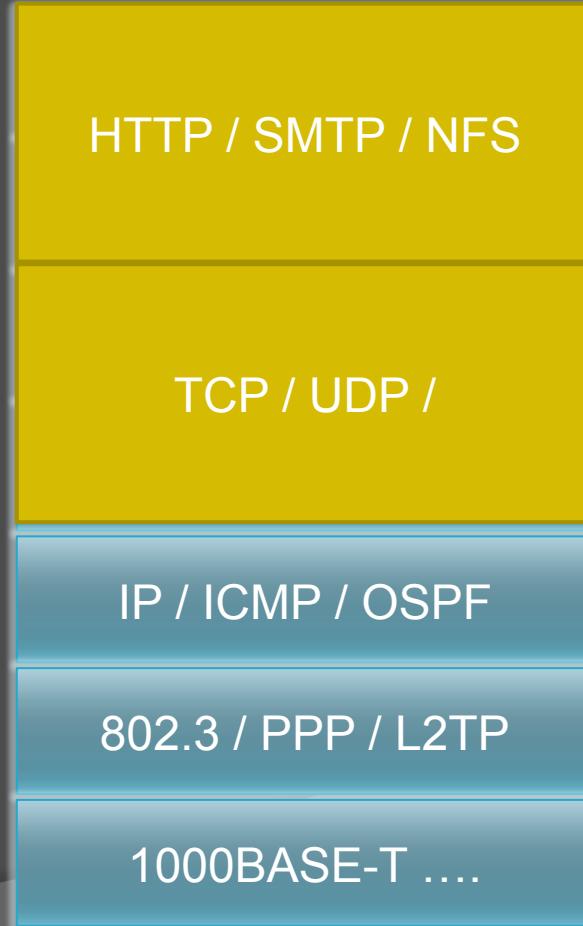


閑話休題

OSI参照モデル



IP Suite





閑話休題2

クッキー・パーシステンスって特許あるの 知ってましたか？

米国特許番号6,473,802

HTTPクッキーでロードバランシング
情報を格納する手法とシステム
F5ネットワークス

Array Networks

Radware

Netscaler

→ クロスパテントや特許料支払い

IBM

→ クロスパテント契約

2003年8月21日

F5ネットワークス社とNetScaler社、クロスライセンス契約締結により、特許論争に終止符

2003年8月21日、シアトル - セキュア・アプリケーション・トラフィック管理製品で業界をリードするF5ネットワークス社(ナスダック:FFIV、同日本法人F5ネットワークスジャパン株式会社 代表取締役社長 ティム・グッドウィン)と、アプリケーション・インテリジェント・ネットワークを取り扱うNetScaler, Inc.は、本日、トラフィック管理やロードバランス製品にとって不可欠なF5ネットワークスの「クッキー・パーシステンス」の特許に関する論争が解決したことを共同発表しました。合意内容としては、NetScalerがF5のクッキー・パーシステンス特許のライセンスを取得、F5はNetScalerの特許を取得し、NetScalerはF5に対しライセンス料金を払う(金額は非公開)という、クロスライセンス契約に調印することが含まれています。

F5社 社長兼CEO、John McAdamは次のように述べています。「NetScalerがF5のクッキー・パーシステンス特許をライセンスすることに关心を示すということは、F5が生み出したこの特許に対する業界での技術評価の高さが証明されています。今回のF5とNetScalerによる合意はF5ネットワークスの知的財産を守る役割を果たすでしょう。」

2002年10月、F5は米国にて特許番号6,473,802「HTTPクッキーでロードバランシング情報を格納する手法とシステム」として、あらゆるトラフィック管理製品、そしてロードバランシング製品の核となっている、クッキー・パーシステンス技術の特許を認可されました。F5の特許製品、クッキー・パーシステンス技術は、ユーザーが前に一度アクセスしたウェブサイトに再びアクセスする際、同じサーバに再接続できるよう、ユーザーのコンピュータに格納されているHTTP クッキーを使用します。たとえば、クッキー・パーシステンスはトラフィック管理デバイスが、インターネット上の買い物を可能にするため、個人のショッピングカード情報が記憶されたサーバに誘導できるようにしています。クッキー・パーシステンスがない場合、トラフィック管理デバイスはお客様の個人情報や、ショッピングカードが記憶されていない別のサーバに導いてしまうことが起こります。

今回のクロスライセンス契約の合意内容としてF5は、サーバを負荷プロセスから解放し、ネットワークインフラのパフォーマンスを向上させる、アメリカで2002年6月に認可されたNetScalerの特許「インターネット・クライアント・サーバ・マルチブレクサー」、特許番号6,411,986のライセンスを取得しました。

DRとは

- disaster recovery（ディザスタリカバリ）
- 拠点崩壊に対する対策を打つことを言う。
- やり方、レベルは千差万別



東京



大阪

コンピュータの世界で

クラスタというと



複数のコンピュータを結合し、
果実・花などの房のように
ひとまとまりとしたシステムのこと

(Wikipediaより)

HAクラスタ
HPC並列クラスタ
負荷分散クラスタ
・・・・





HAクラスタとは

- 一台のコンピュータでは得られない高い信頼性を狙うために、複数のコンピュータを結合し、ひとまとまりとしたシステムのこと
- デュプレックスシステム
- Shared Diskで使うことが多いが
Shared Nothingでも使用。



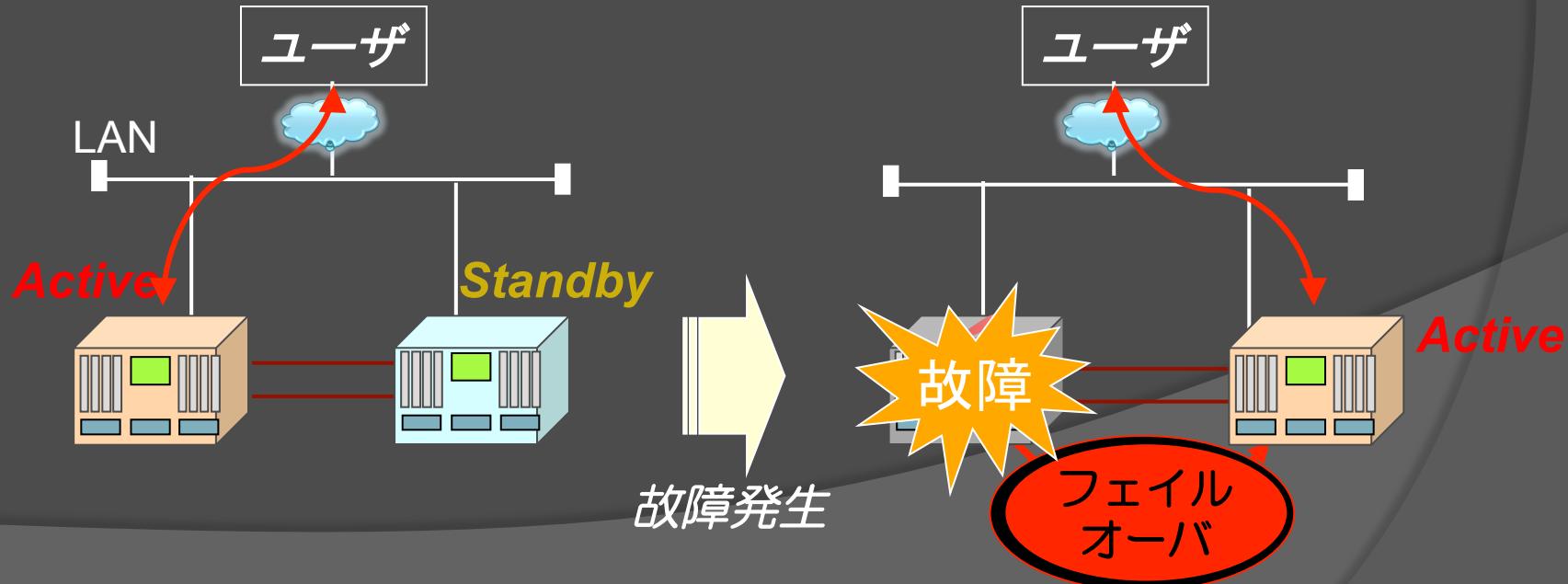
主なHA製品

- HACMP → PowerHA
- MC/ServiceGuard
- Veritas Cluster Server
- MSCS
- ClusterPro
- Lifekeeper
- RHCS
- Heartbeat / Pacemaker

基本構成

Active/Standby(1+1)構成

- 通常はActiveノードと呼ばれるサーバでサービスを提供します。
- Activeノードが故障した場合は、StandbyノードがActiveになりサービスを引き継ぎます。これをフェイルオーバと呼びます。





Keepalived

- Linux上でVRRPを実装し、VIPを持たせることができる。
- あくまで、ネットワークの二重化用途。

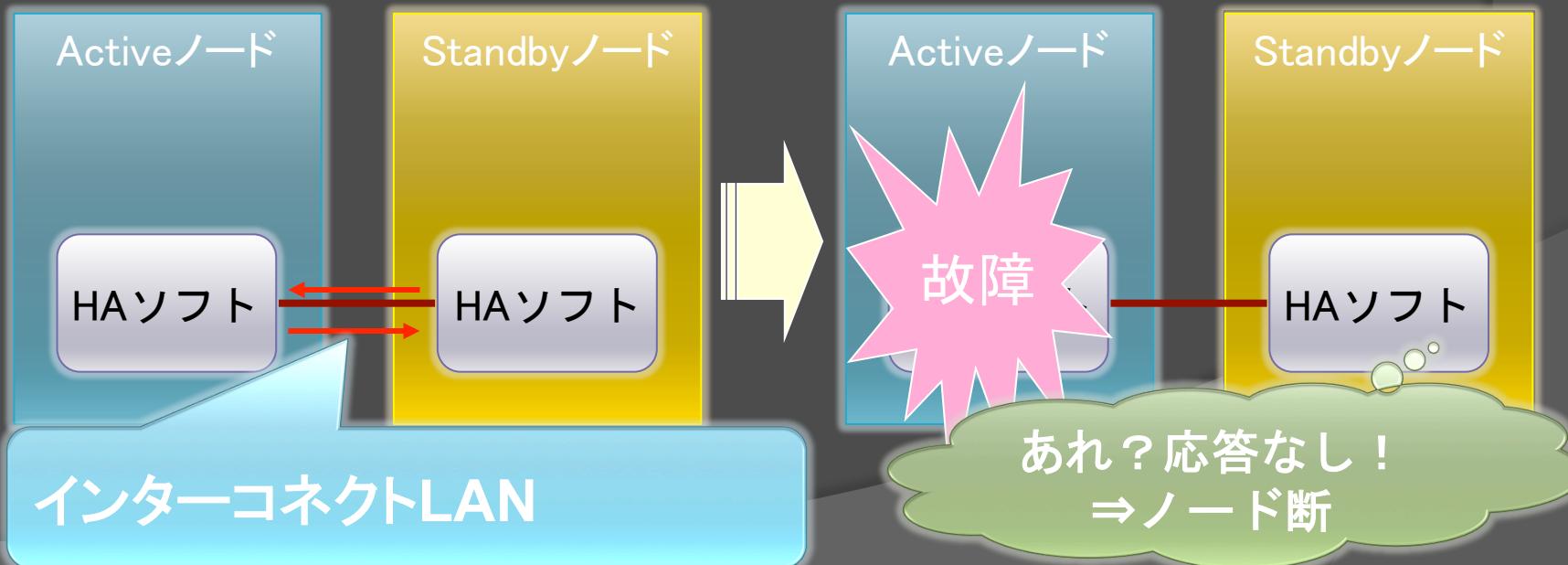
注：

- VRRPはCISCOが持つHSRPの特許に抵触？
- VRRPもいろんな会社が特許出してますよ。
→VRRPの特許を抜いたプロトコル CARP

Linux-HA Japanでは、Keepalivedの使用は推奨していません。

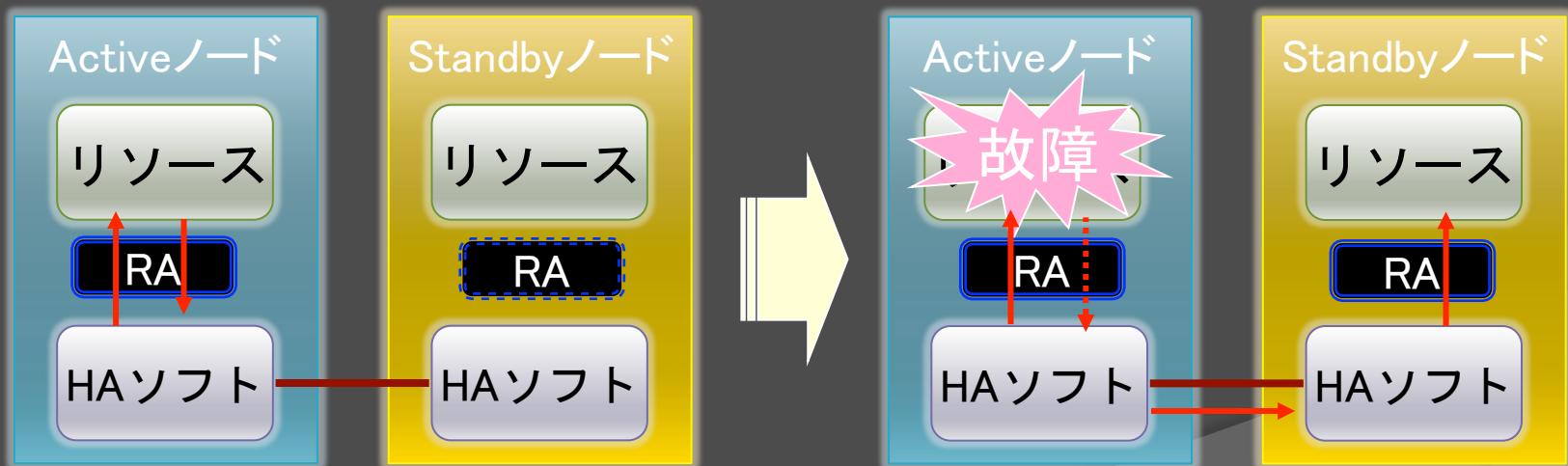
基本動作1：ノード監視

- 相手ノードの生死を確認するために、一定間隔で相手ノードを監視します。
- 相手ノードと通信できなくなった場合に、相手はダウンしたと判断し、フェイルオーバなどのクラスタ制御の処理を行います。



基本動作2: リソース制御

- リソースと呼ばれる物をリソースエージェント(RA)を介して起動(start)、停止(stop)、監視(monitor)します。
- リソースが故障した場合にはフェイルオーバといったリソース制御の処理を行います。



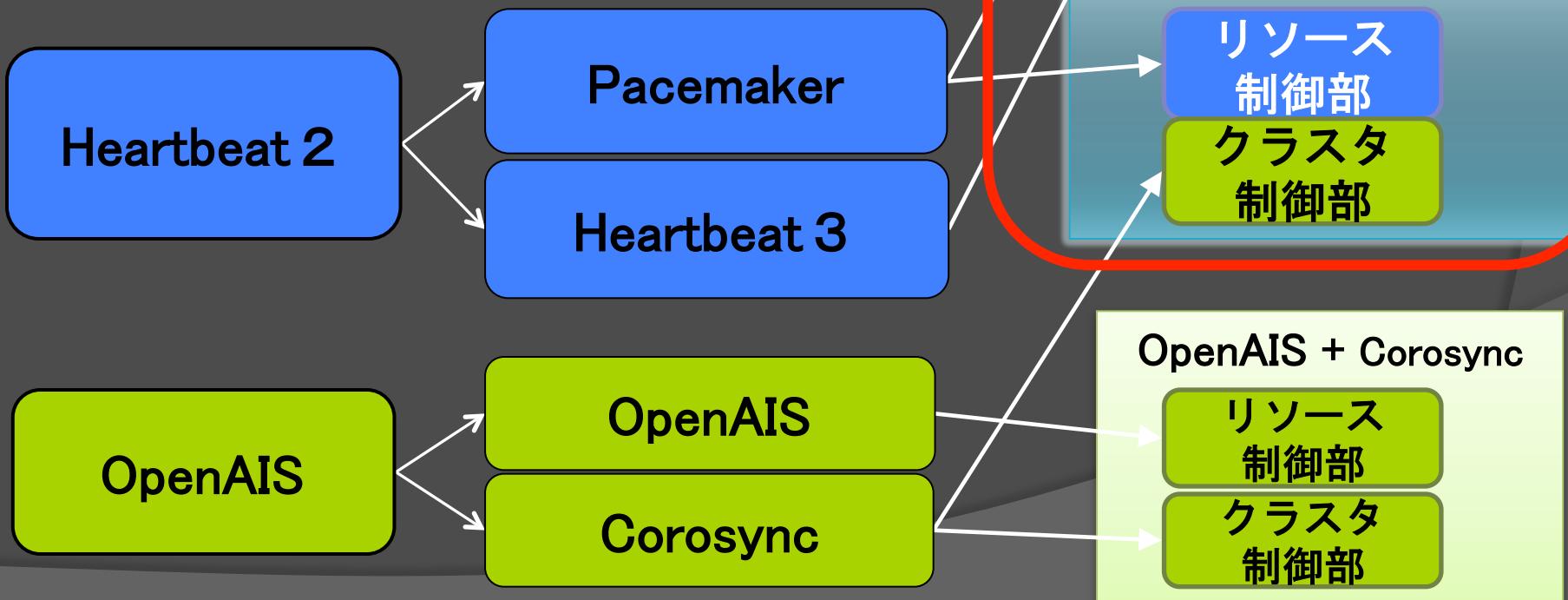


Pacemakerとは

- OpenSourceで開発されている
HAクラスタソフト
- Heartbeat2の後継
- 複数のコミュニティの製品をまとめて動作
- 主要Linuxディストリビューションで
動作可能
- 有償で保守サポートする会社あり。

コンポーネント

オープンソースのHAクラスタはこのように複数のコンポーネントの組み合わせとして提供されるようになりました。





Linux-HA Japanとは

『Heartbeat(ハートビート)』の日本における更なる普及展開を目的として、2007年10月5日「Linux-HA (Heartbeat) 日本語サイト」を設立

その後、日本でのLinux-HAコミュニティ活動として、Heartbeat2のrpmバイナリと、オリジナルのHeartbeat機能追加用パッケージを提供



Linux-HA Japan URL

<http://linux-ha.sourceforge.jp/>



情報の公開用として
新しい一般向けウェブサイト
を2010年6月25日にオープン

随時情報を更新中

Linux-HA Japan メーリングリスト



日本におけるHAクラスタについての活発な意見交換の場として「Linux-HA Japan 日本語メーリングリスト」も開設しています。

Linux-HA-Japan MLでは、Pacemaker、Heartbeat3、Corosync DRBDなど、HAクラスタに関連する話題は歓迎！

- ML登録用URL

<http://linux-ha.sourceforge.jp/>

の「メーリングリスト」をクリック



- MLアドレス

linux-ha-japan@lists.sourceforge.jp

※スパム防止のために、登録者以外の投稿は許可制です



本家Pacemakerサイト clusterlabs.org

<http://clusterlabs.org/>

Fedora, openSUSE,
EPEL(CentOS/
RHEL)のrpmがダウ
ンロード可能です。

The screenshot shows the official Pacemaker website at <http://clusterlabs.org/>. The page features a large blue header with the Pacemaker logo and the text "A scalable High-Availability cluster resource manager". Below the header, there are navigation links for "The Team", "Overview", "Features", "FAQ", and "Explore". A central "Overview" box contains text about Pacemaker's role in managing clusters and ensuring high availability through shared failover. To the right, there is a diagram titled "Shared Failover" illustrating how multiple services (Mail, Web Site, DB, Files, Storage) are managed by Pacemaker across four hosts, with CoroSync providing synchronization.

Get Pacemaker

Get Pacemaker

http://clusterlabs.org/

Norton Confidential

clusterlabs menu

A scalable High-Availability cluster resource manager

Pacemaker

The Team Overview Features FAQ Explore

Deployment Examples

Shared Failover

Find out more about Pacemaker on our wiki

Send site feedback to the project mailing list or maintainer: Andrew Beekhof

まとめ

- 高信頼システム構築は、様々な要件を確認した上で、適切に構築する必要がある。
- 特にSPOFを作らないように設計する。
- ハード以外の要因によるシステム停止を意識して設計する。
- 必要に応じてHAや負荷分散構成を用いる。