

サーバ負荷分散技術

鍋島公章

<http://www.kosho.org/books/serverload/>

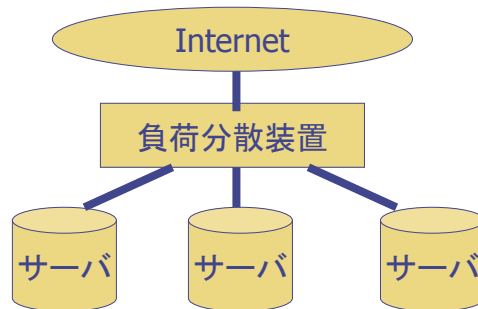
アウトライン

- ◆サーバ負荷分散入門
- ◆サーバ負荷分散システム構築

サーバ負荷分散

◆ 対象

- Webサーバ
 - ◆ 負荷の水平分散
 - ◆ ネットワーク技術としての負荷分散

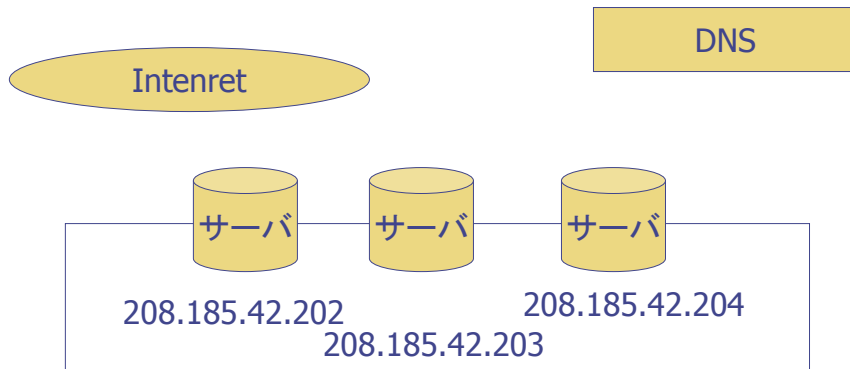


(C) kosho.org

DNSによる負荷分散

◆ 概要

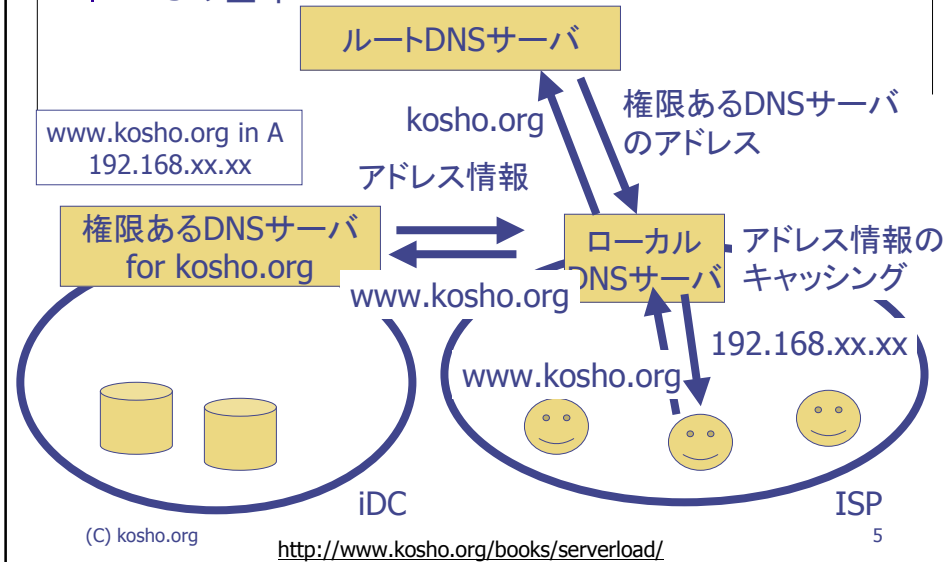
```
www.vegan.net IN A 208.185.43.202
                208.185.43.203
                208.185.43.204
```



(C) kosho.org

DNSによる負荷分散

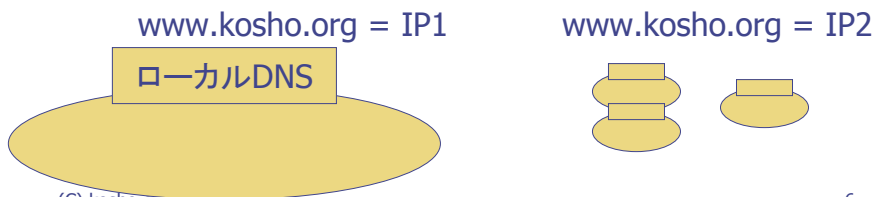
◆ DNSの基本



DNSによる負荷分散

◆ 問題点

- キャッシング問題
 - ◆ DNS情報はキャッシュされる(通常数週間)
 - ◆ サーバ障害に対応できない
- トラフィック割り振り問題
 - ◆ ローカルDNS単位にIPアドレスがキャッシュされる
 - 例) 巨大なISPのローカルDNSサーバに、あるIPアドレスがキャッシュされる



Layer 4スイッチによる負荷分散

◆ Layer 4スイッチ

- Layer4(トランスポート層)
 - ◆ コネクション単位にスイッチ
- 対比
 - ◆ Layer3(ネットワーク層)スイッチ/ルータ
 - パケット(IPアドレス)単位にスイッチ
 - ◆ Layer7(アプリケーション層)スイッチ
 - コンテンツ単位にスイッチ



(C) kosho.org

<http://www.kosho.org/books/serverload/>

7

Layer4スイッチによる負荷分散

◆ Layer4(コネクション単位)スイッチングの実現方法

- スイッチング単位
 - ◆ 従来技術(Layer3スイッチ、ルータ)
 - IPアドレス、MACアドレス
 - ◆ Layer4スイッチ
 - IPアドレス+ポート番号
 - ◆ Layer7スイッチ
 - コネクションの中身

(C) kosho.org

<http://www.kosho.org/books/serverload/>

8

Layer7スイッチング

◆コネクションの中身を見たスイッチング

- コンテンツ種別単位
 - ◆ 静的コンテンツ
 - HTML、イメージ...
 - ◆ 動的コンテンツ
 - CGI、ASP...
- クライアント種別単位
 - ◆ PCクライアント
 - ◆ I-modeクライアント

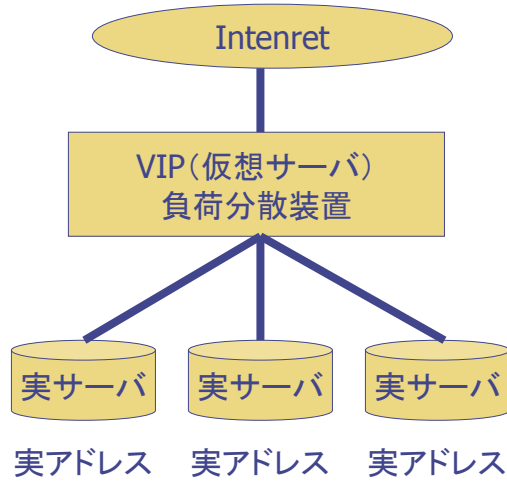
スイッチング方法

◆スイッチング方法

- 従来技術(Layer3スイッチ、ルータ)
 - ◆ パケットスイッチング+複数ポート
- Layer4、7スイッチ
 - ◆ NAT(Network Address Translation)
 - アドレス書き換え
 - ◆ MAT(Mac Address Translation)
 - 直接サーバ返答(Direct Server Return, DSR)

アドレス書き換え

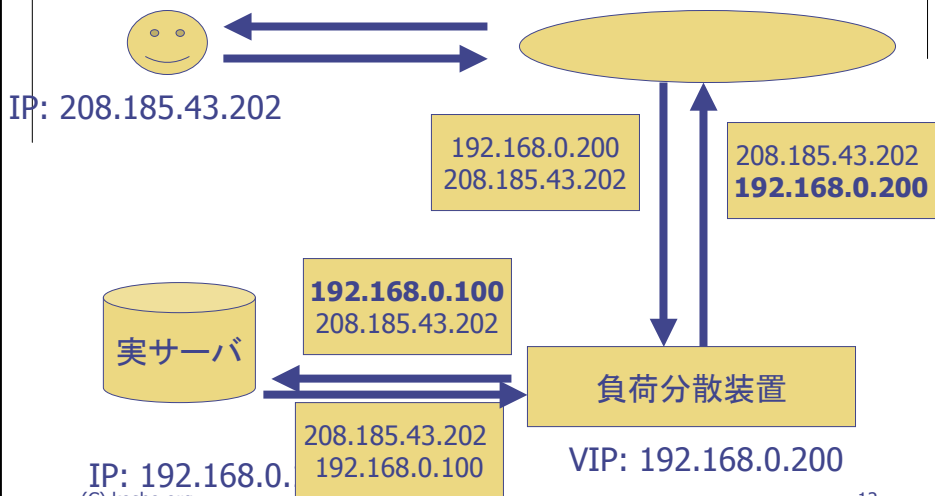
◆ NAT(Network Address Translation)



(C) kosho.org

アドレス書き換え

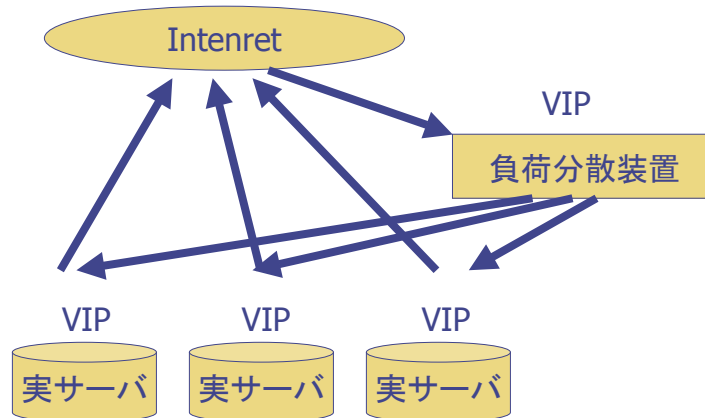
◆ パケットの旅



(C) kosho.org

直接サーバ返答

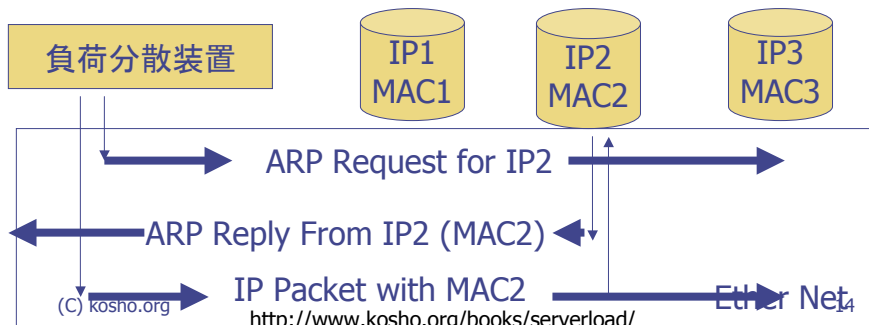
◆ MAT (MAC Address Translation)



直接サーバ返答

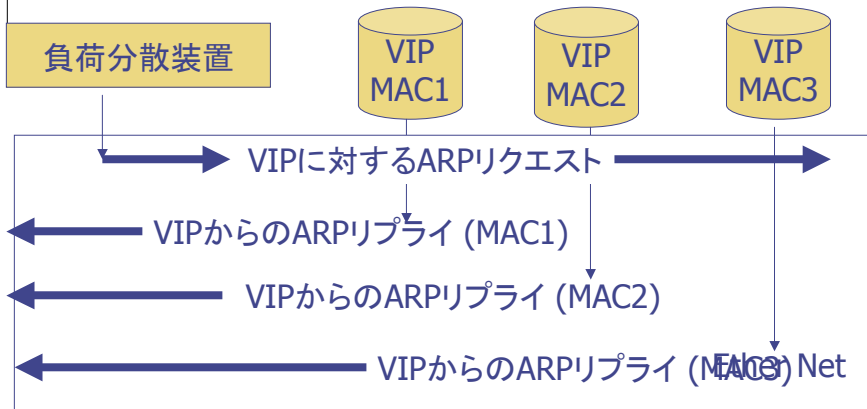
◆ EtherNet (シェアードメディア) でのパケットの受渡し

- EtherNetは汎用のLayer2デバイス
 - ◆ IPアドレスは使えない
 - ◆ MACアドレスをベースとした受け渡し
 - ◆ ARP (Address Resolution Protocol) を使い相手の送り先のMACアドレスを問い合わせ



直接サーバ返答

- ◆ 複数のサーバが同一IPアドレス (VIP) を持つ
 - ARPリプライのバッチング



(C) kosho.org

直接サーバ返答

- ◆ 回避方法
 - ループバックインターフェイスへVIPを割り振る
 - ◆ 通常は、ホスト内通信用の仮想インターフェイス (127.0.0.1)
 - ARPには答えない
 - ◆ IPエイリアスを利用
 - ARPを使わない (MAT)
 - ◆ 負荷分散装置に実サーバのMACアドレステーブルを持たせる

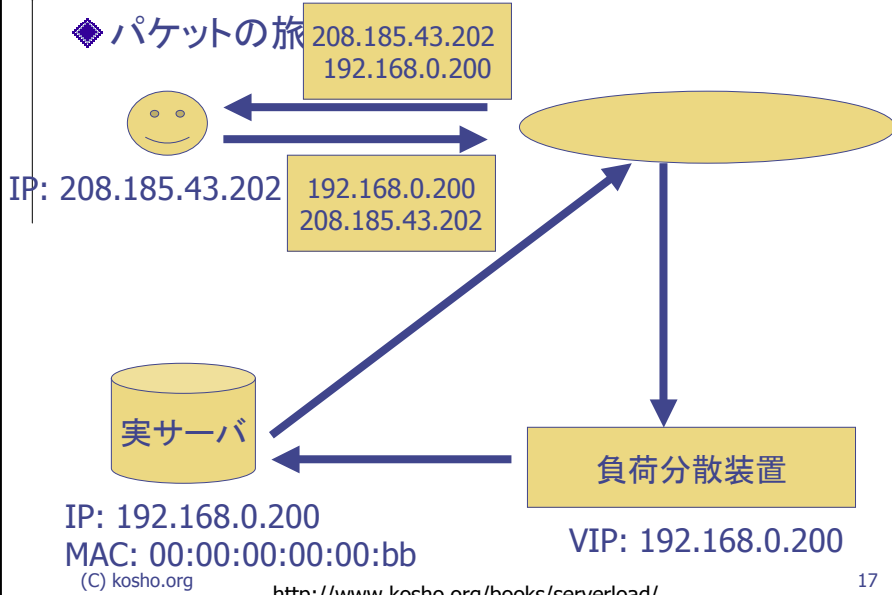
MAC1
MAC2
MAC3



(C) kosho.org

直接サーバ返答

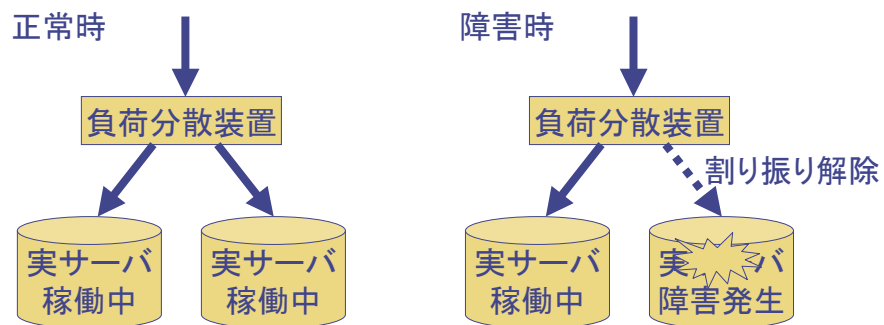
◆ パケットの旅



負荷分散の役割

◆ VIPへのクエストを複数の実サーバに割り振る

- フェイルセーフ
 - ◆ 障害サーバをリクエスト割り振りから外す
 - 回復すれば、リクエスト割り振りに戻す



負荷分散の役割(ヘルスチェック)

◆ヘルスチェックのレベル

- Pingチェック: ネットワーク(Layer3)到達性
 - ◆ サーバまでネットワークが繋がっているか?
- ポートチェック: ネットワーク(Layer4)到達性
 - ◆ TCPが張れるか?
 - 80番ポートは生きているか?
- プロトコルチェック: アプリケーションの生存性
 - ◆ HTTPが張れるか?
 - “GET / HTTP/1.0”を発行して、OKが帰るか?
 - HTTPは簡単すぎて意味がない。複雑なプロトコルでは有効
- コンテンツチェック: アプリケーションの正当性
 - ◆ 正しいコンテンツを返しているか?
 - “GET / HTTP/1.0”を発行して、特定文字列が含まれるか?

負荷分散の役割(ヘルスチェック)

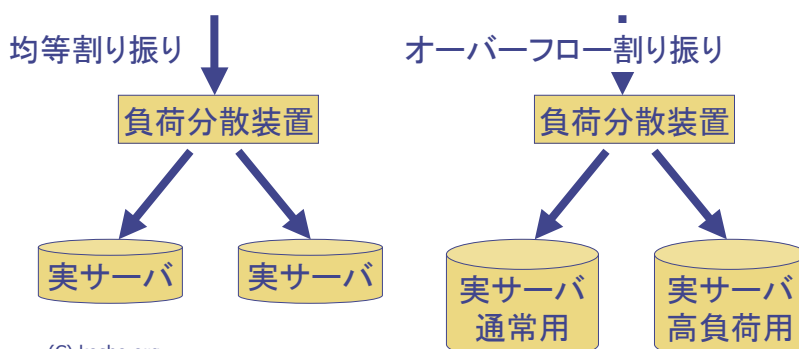
◆注意点

- 基本的には、短い間隔で発行した方がよい
 - ◆ 迅速な障害検知
- ヘルスチェックは、実際にネットワーク、アプリケーションを動かす
 - ◆ 負荷の上昇
 - ◆ アクセスログ
- 超高負荷とサーバダウン
 - ◆ 超高負荷状態をサーバダウンと誤認
 - サーバの切り離しではなく、割り振り解除だけにすべき

負荷分散の役割

◆VIPへのクエストを複数の実サーバに割り振る

- 負荷分散
 - ◆ 負荷を複数の実サーバに割り振る
 - 均等割り振り
 - オーバーフロー割り振り



(C) kosho.org

<http://www.kosho.org/books/serverload/>

21

振り分けアルゴリズム

◆ラウンドロビン

- ◆ サーバがダウンしていない限り、リクエストをサーバ群にぐるぐる割り振る
- 純粹ラウンドロビン
 - ◆ サーバの能力が同じ場合
- 重み付けラウンドロビン
 - ◆ サーバの能力が異なる場合

(C) kosho.org

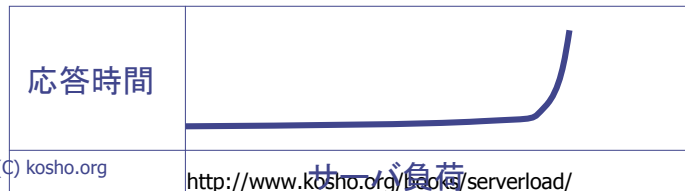
<http://www.kosho.org/books/serverload/>

22

振り分けアルゴリズム

◆ コネクション

- 同時接続数
 - ◆ 処理しているコネクションが少ないサーバにリクエストを割り振る
 - ◆ サーバの処理能力が異なる時に上手く動かない
- 応答時間
 - ◆ サーバの応答時間が短いサーバにリクエストを割り振る
 - ◆ 応答時間は急激に悪くなる
 - ◆ プローブによる応答時間計測は、数字が暴れる



(C) kosho.org

<http://www.kosho.org/books/serverload/>

23

振り分けアルゴリズム

◆ サーバ負荷

- SNMP (Simple Network Management Protocol)
- 専用デーモン
- 使える負荷分散装置が少ない

◆ 優先順位

- オーバフロー割り振り

(C) kosho.org

<http://www.kosho.org/books/serverload/>

24

負荷分散装置のパフォーマンス

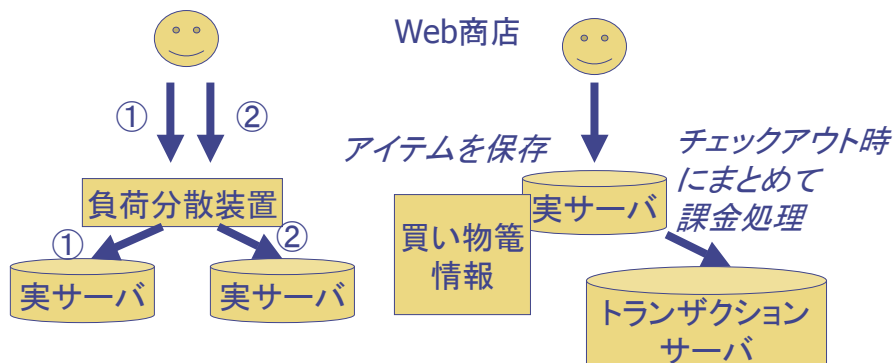
◆ 指標

- 毎秒あたりの接続数
 - ◆ 一秒あたりの、接続開始数
 - ◆ HTTPの場合、接続開始処理が重い
 - TCPのスリーウェイハンドシェイク
- 全同時接続数
 - ◆ 接続保持数
 - ◆ おもにメモリの量に依存
- スループット
 - ◆ 処理できるトラフィックの量

接続維持

◆ 負荷分散

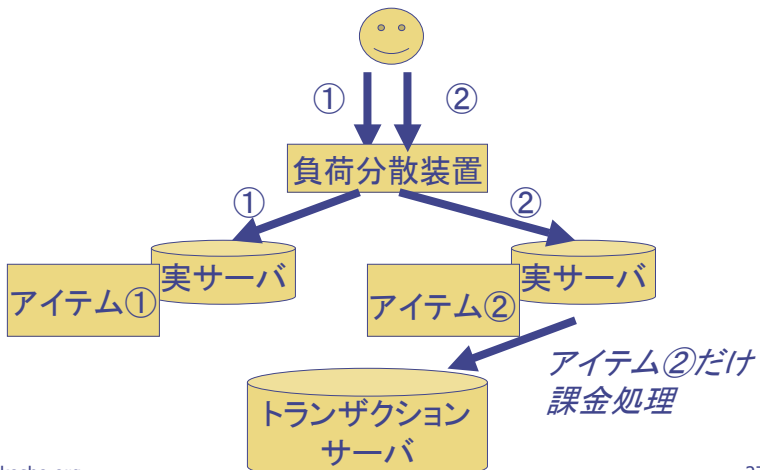
- コネクションを均等に複数のサーバに割り振る
- Web商店
 - ◆ 買い物カゴの情報をWWWサーバ単位で管理



接続維持

◆ 買い物カゴ問題

- 買い物カゴにアイテムが集まらない



(C) kosho.org

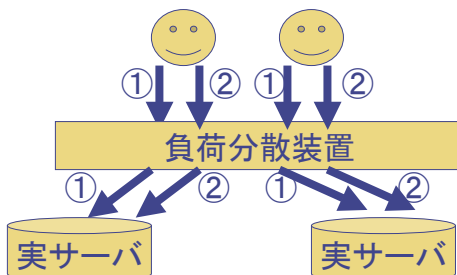
<http://www.kosho.org/books/serverload/>

27

接続維持

◆ 接続維持

- あるユーザのリクエストを一つのサーバに割り振る
 - 通常の負荷分散は、リクエスト単位でリクエストを割り振る
- ◆ ユーザ単位のリクエスト割り振り
 - ユーザの最初のリクエストが割り振られたサーバに、2回目以降のリクエストも割り振り続ける (接続維持)



(C) kosho.org

<http://www.kosho.org/books/serverload/>

28

接続維持

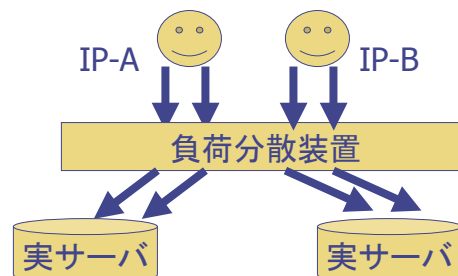
◆ ユーザ認識の方法

- 基本
 - ◆ 始点IPアドレス
 - ◆ クッキー
- SSL通信用
- 携帯電話用

接続維持

◆ 始点IPアドレス

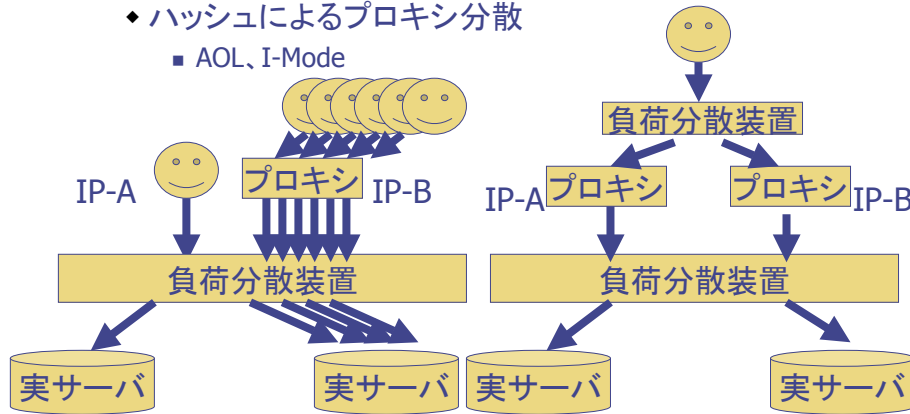
- 一番簡単、汎用性が高い
 - ◆ ユーザとIPアドレスが1対1であると上手く動く



接続維持

◆ 始点IPアドレス

- プロキシ(ゲートウェイ)サーバ問題
 - ◆ 特定のIPアドレス(ISPのプロキシ)から大量のリクエスト
 - ◆ ハッシュによるプロキシ分散
 - AOL、I-Mode



(C) kosho.org

接続維持

◆ クッキー

- クッキーの基本
 - ◆ WWWサーバ
 - ブラウザに、クッキー文字列と対応するURLを保存させる
 - ◆ ブラウザ
 - そのURLにアクセスする度に、保存されたクッキー文字列をWWWサーバに送る



(C) kosho.org

接続維持

◆ クッキー

- 付けすぎに注意
 - ◆ クッキーをつけると、キャッシュされない
 - HTMLファイルだけにクッキーを適用
- 接続維持用のクッキー
 - ◆ アプリケーション処理のためのユーザID
 - クッキーハッシュ
 - ◆ 接続維持のためのサーバID
 - インサート
 - パッシブ
 - リライト

接続維持

◆ モード

- クッキーハッシュモード
 - ◆ 既存のアプリケーションが使っているクッキーのハッシュ値をサーバIDとする



- パッシブモード
 - ◆ WebサーバがサーバIDを含むクッキー格納命令を送信

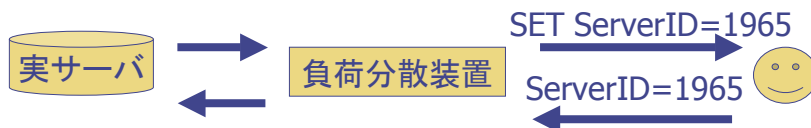


接続維持

◆モード

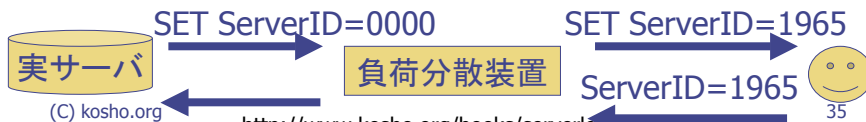
■ インサートモード

- ◆ 負荷分散装置がWebサーバからのHTTPレスポンスに、クッキー格納命令を自動挿入



■ リライトモード

- ◆ Webサーバが空白IDを含むクッキー格納命令を送信
- ◆ 負荷分散装置が、空白IDをサーバIDに書き換え



(C) kosho.org

接続維持

◆ SSL (Secure Sockets Layer)通信用

■ コネクションの暗号化

- ◆ 負荷分散装置もコネクションの中身を見れない
 - クッキーは使えず



◆ SSLセッションID

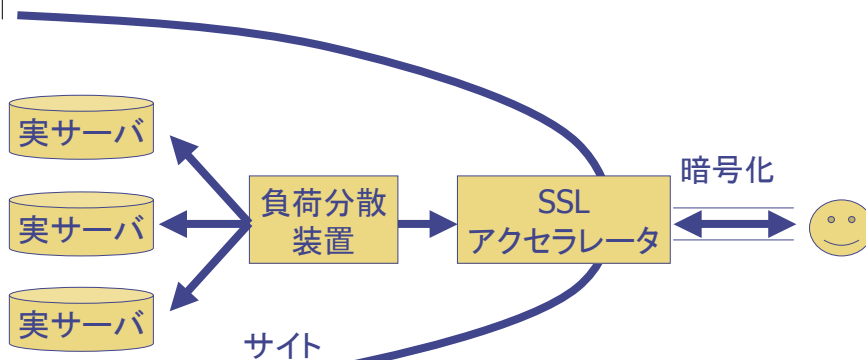
- SSLを定期的に再接続するブラウザ
- ◆ 始点IPアドレスしかない

(C) kosho.org

接続維持

◆ SSL(Secure Sockets Layer)通信用

- SSLアクセラレータの使用
 - ◆ SSLで通信を暗号化する箱
- サイト内部は平文で通信



(C) kosho.org

接続維持

◆ 携帯電話用

- 特徴
 - ◆ ゲートウェイ方式(負荷分散されている)
 - リクエスト毎にIPアドレスが変化することがある
 - ◆ 簡易HTTP
 - クッキーが使えない
- リンク先URLにサーバIDを埋め込む
 - ◆ `<a href=index.html`
 - `<a href=index.html&ServerID=10`
- 端末情報の利用
 - ◆ ゲートウェイが端末固有のIDを付ける

(C) kosho.org

Part 2

◆ 負荷分散システム構築

- 冗長性
- アーキテクチャ

冗長性

◆ サーバの冗長性

- 負荷分散装置で実現

◆ 負荷分散装置の冗長性

- ネットワークプロトコルにより実現

◆ 基本モード

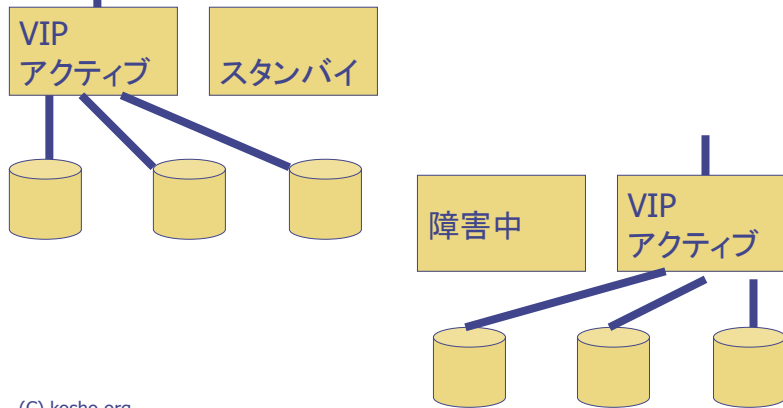
- アクティブ-スタンバイ
- アクティブ-アクティブ

冗長性

◆ 基本モード

- アクティブ-スタンバイ

- ◆ 待機マシンが稼働マシンを置き換える



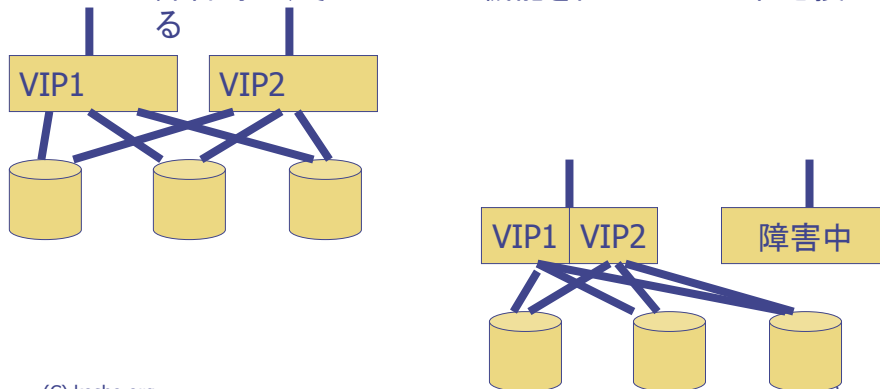
(C) kosho.org

冗長性

◆ 基本モード

- アクティブ-アクティブ

- ◆ 全てが稼働マシンとして働く
- ◆ 障害時に、そのマシンの機能を他のマシンが置き換える

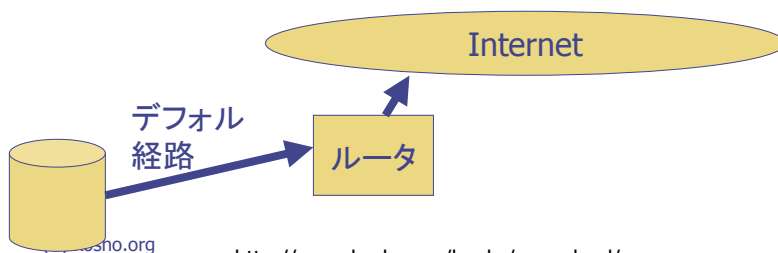


(C) kosho.org

冗長性(ルータの二重化)

◆ デフォルト経路

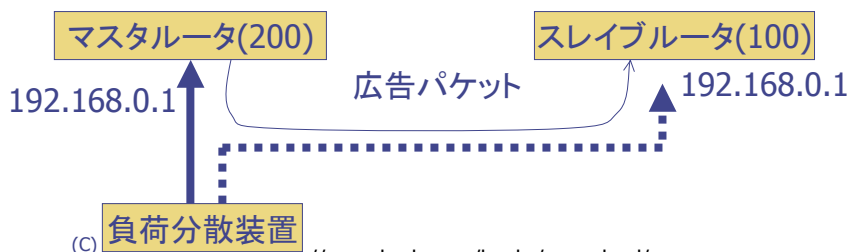
- Internetへ向かうトラフィックの一つの出口(IPアドレス)
- 一般に、サーバ、負荷分散装置では、デフォルト経路はひとつしか指定出来ない
 - ◆ 指定したルータに障害が起こるとInternetと接続できない
 - ◆ ルータ(デフォルト経路)の二重化が必須



冗長性(ルータの二重化)

◆ VRRP (Virtual Router Redundancy Protocol)

- 同一IPアドレス(デフォルト経路)を2台のルータに持たせる
- アクティブスタンバイ方式
 - ◆ VRRP優先度(高い方が優先)
 - ◆ アクティブ機は、広告パケットを送る(224.0.0.18)
 - スタンバイ機は広告パケットが届かなくなったら、アクティブ機に障害が発生したと判断



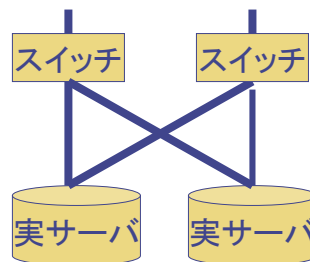
冗長性(負荷分散装置のフェイルオーバ)

- ◆ 各ベンダにより実装の方法が違う
 - フェイルオーバケーブル
 - 一般VLAN
 - ◆ VRRPと同様の方法
 - 専用VLAN

- ◆ ステイトフルフェイルオーバ
 - フェイルオーバ時に機能を引き継ぐ
 - ◆ セッション、クッキー情報等を定期的にスタンバイ機に送信

冗長性(スイッチのフェイルオーバ)

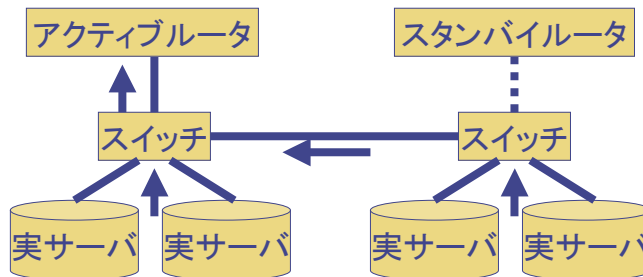
- ◆ 2つのスイッチへ接続(クロス接続)
 - 実サーバでは可能
 - ◆ ポートごとにIPアドレスを割り振る必要
 - 負荷分散装置では一般的ではない



冗長性(スイッチのフェイルオーバ)

◆トランクによる2台構成

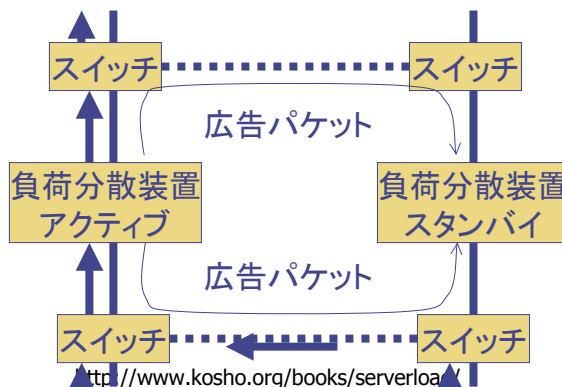
- 片方が落ちたら、それに繋がっているサーバはあきらめる
- トランク経由でトラフィックはアクティブルータに送る



冗長性(スイッチのフェイルオーバ)

◆トランクによる2台構成

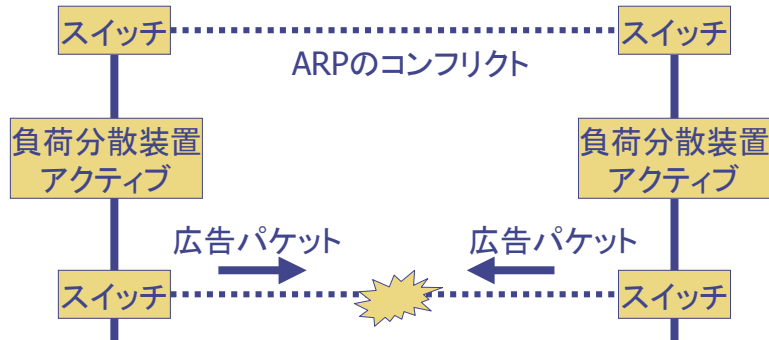
- 負荷分散装置、ルータの場合、スイッチが死んだら自身を障害状態にする
 - ◆ 入出力出来ないのに生きていてもしかたない



冗長性(ケーブルのフェイルセーフ)

◆トランクケーブルの障害(1)

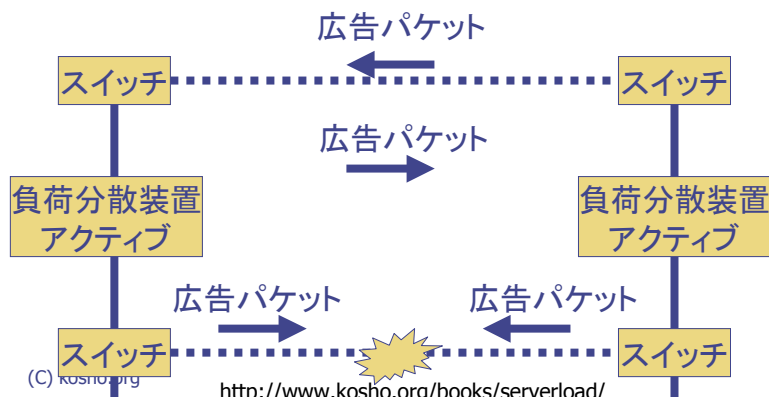
- 広告パケットが届かない
- アクティブ-アクティブになる可能性
 - ◆ ARPのコンフリクト問題が発生



冗長性(ケーブルのフェイルセーフ)

◆トランクケーブルの障害

- 複数の経路による広告パケットの送付
 - ◆ 残りの経路で広告パケットのコンフリクト
 - ◆ 旧アクティブ装置がスタンバイとなる



冗長性(ケーブルのフェイルセーフ)

◆ 負荷分散装置(スタンバイ機)のケーブル障害

- スタンバイ機がアクティブとなる
 - ◆ トランクケーブル対策のため
 - 負荷分散装置間の接続性がひとつでも切れるとスタンバイ装置がアクティブとなる
 - ◆ 実際には、ケーブル障害のため稼働できない

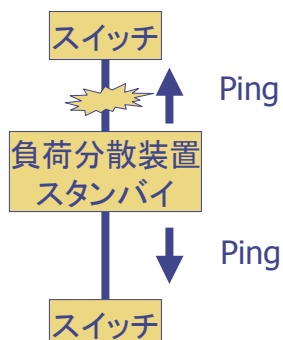


(C) kosho.org

冗長性(ケーブルのフェイルセーフ)

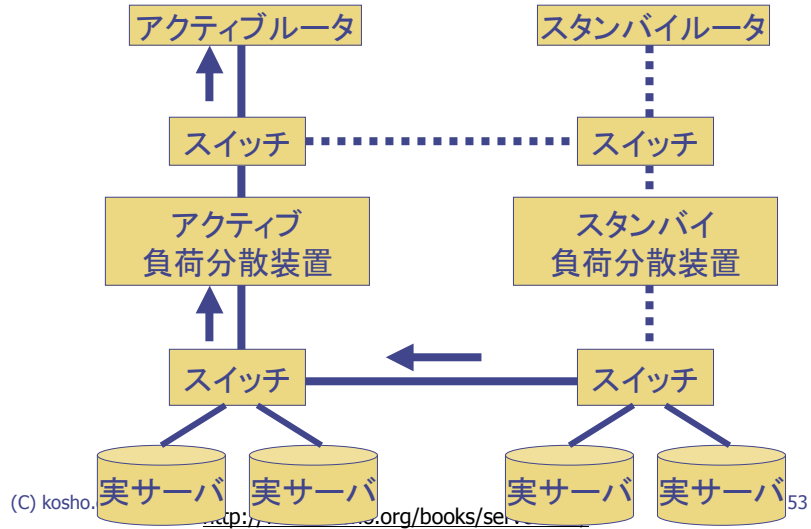
◆ 負荷分散装置(スタンバイ機)のケーブル障害

- 対策
 - ◆ リモートホストへの到達性監視(ケーブルセーフ機能)
 - ケーブル越しの特定IPアドレスへPING
 - 到達性が確認されない場合、アクティブにならない



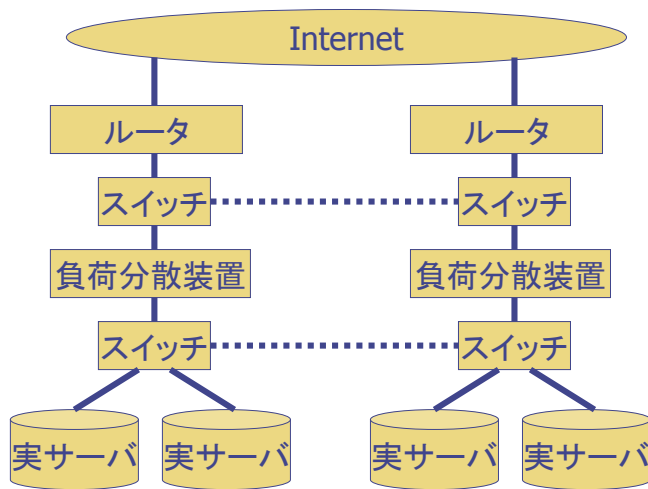
(C) kosho.org

冗長性(最終構成)



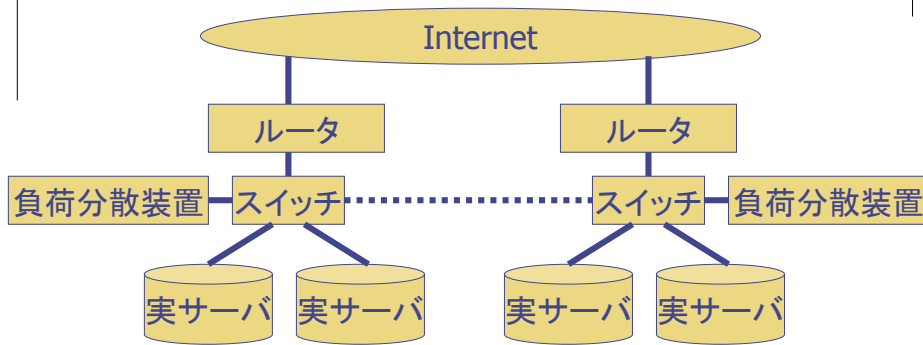
アーキテクチャ

◆ サンプル(1)



アーキテクチャ

◆ サンプル(2)



アーキテクチャ

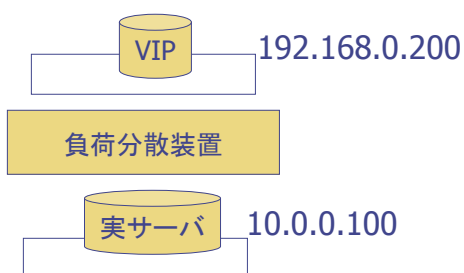
◆ 分類

- IPアドレス設定
 - ◆ VIPと実サーバのIPアドレスのネットワークアドレス
 - フラットベース、NATベース
- サーバ返答
 - ◆ 実サーバから負荷分散装置へ向かうトラフィックの処理方法
 - ◆ 負荷分散装置の役割
 - ルート、ブリッジ、直接サーバ返答
- 物理接続
 - ◆ 負荷分散装置のインターフェイス数
 - 1本腕、2本腕

アーキテクチャ

◆ NATベース

- VIPと各サーバが異なるサブネット
- トラフィックの流れが単純
- 実サーバは直接外部Internetと接続不可能
 - ◆ ある程度のセキュリティが保たれる

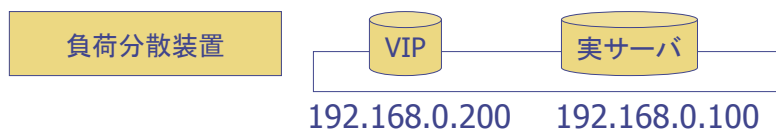


(C) kosho.org

アーキテクチャ

◆ フラットベース

- VIPと各サーバが同一サブネット
- NATを使用
- 特徴
 - ◆ 構成が単純
 - 冗長構成を取りやすい
 - ◆ 実サーバが外部Internetと直接接続可能



(C) kosho.org

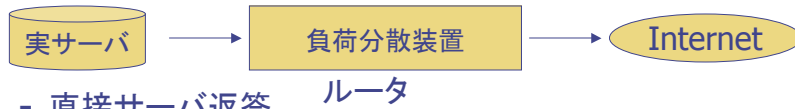
アーキテクチャ

◆ サーバ返答

■ ブリッジ構成



■ ルート構成



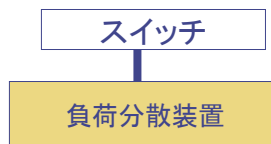
■ 直接サーバ返答



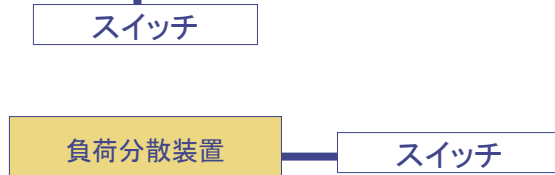
アーキテクチャ

◆ 物理接続(インターフェイス数)

■ 2本腕構成



■ 1本腕構成

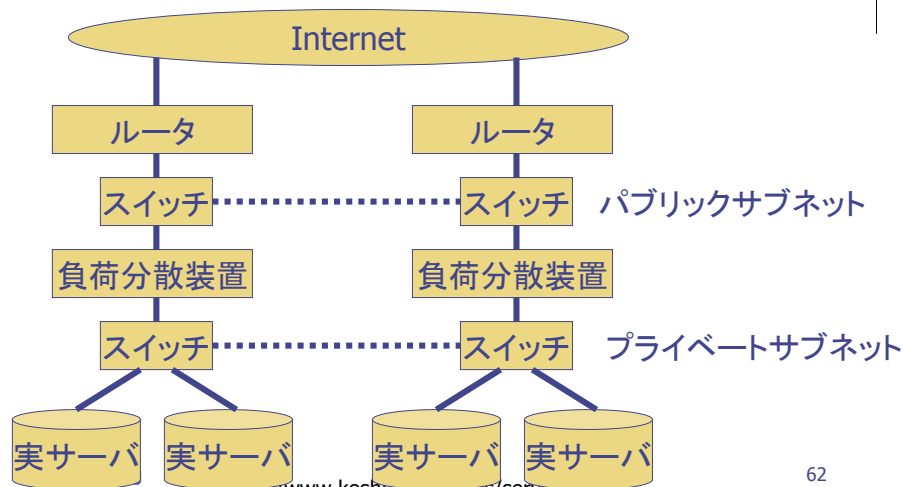


NATベース

- ◆ ルート構成
 - 2本腕構成
 - 1本腕構成
- ◆ ブリッジ構成
 - 2本腕構成
 - 1本腕構成
- ◆ 直接サーバ返答
 - 2本腕構成
 - 1本腕構成

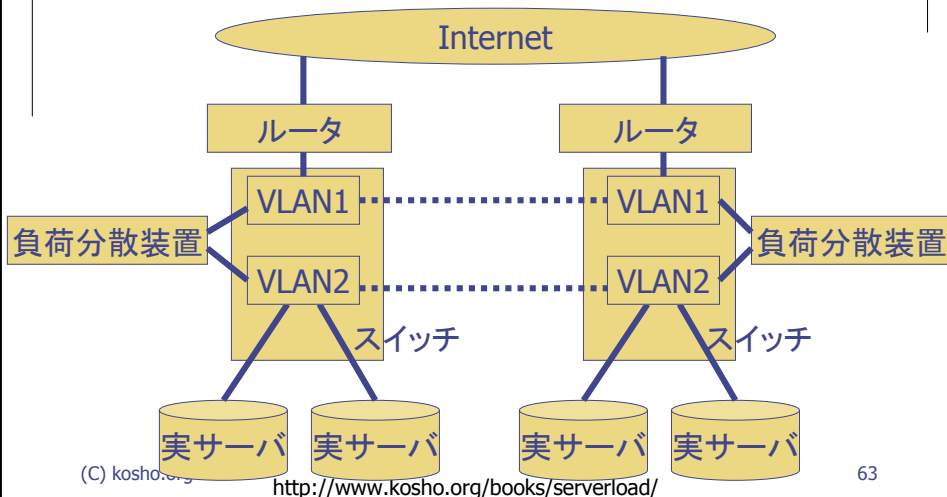
NATベース

- ◆ ルート、2本腕構成(論理構成)
 - NATベースの典型的構成



NATベース

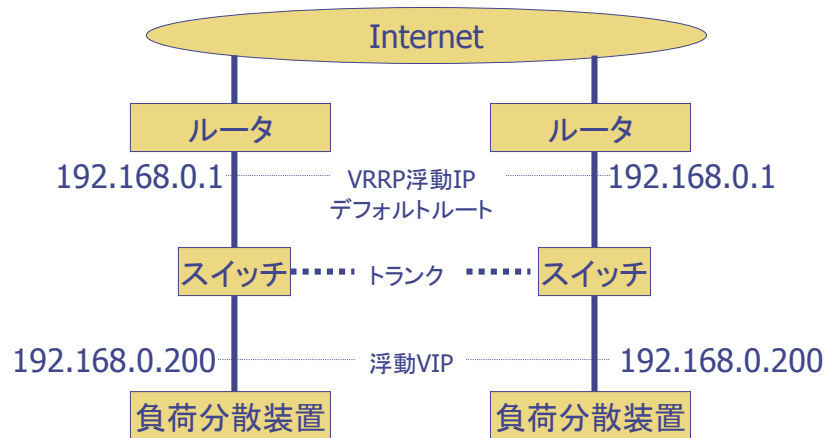
◆ ルート、2本腕構成(論理構成)



NATベース

◆ ルート、2本腕構成(論理構成)

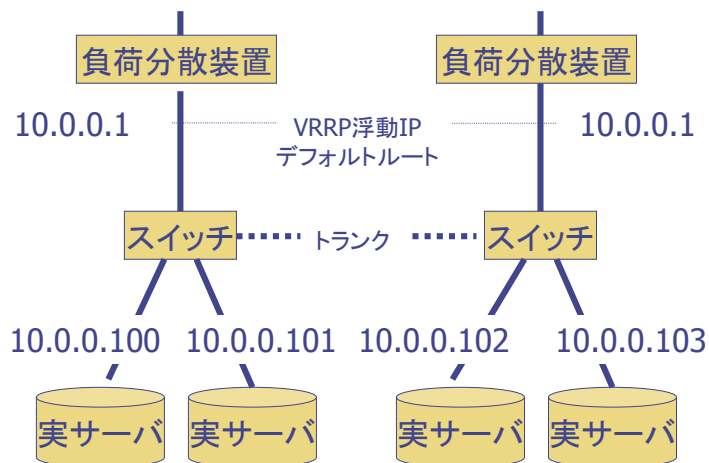
- パブリックサブネット



NATベース

◆ ルート、2本腕構成(論理構成)

- プライベートサブネット



(C) kosho.org

NATベース

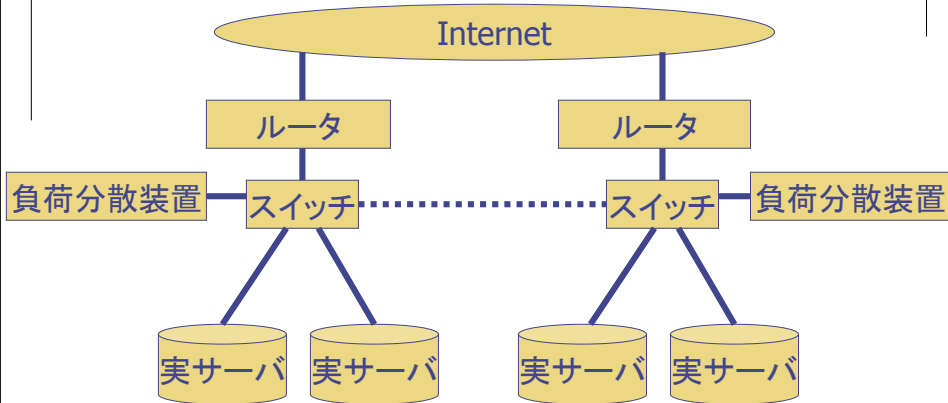
◆ ルート構成

- 2本腕
 - ◆ パブリック、プライベートネットワークの、それぞれにスイッチ(VLAN)を用意する
- 1本腕
 - ◆ パブリック、プライベートネットワークを一つのスイッチ(VLAN)上に重ねる
 - ◆ トリッキーな方法、一般には避ける

(C) kosho.org

NATベース

◆ ルート、1本腕構成



NATベース

◆ ブリッジ構成

- NATベースでは二つのサブネットを使う
 - ◆ サブネット間のパケット交換はルータモードが必要
- NATベースでは構成できず

◆ 直接サーバ返答

- 非常にトリッキーな方法

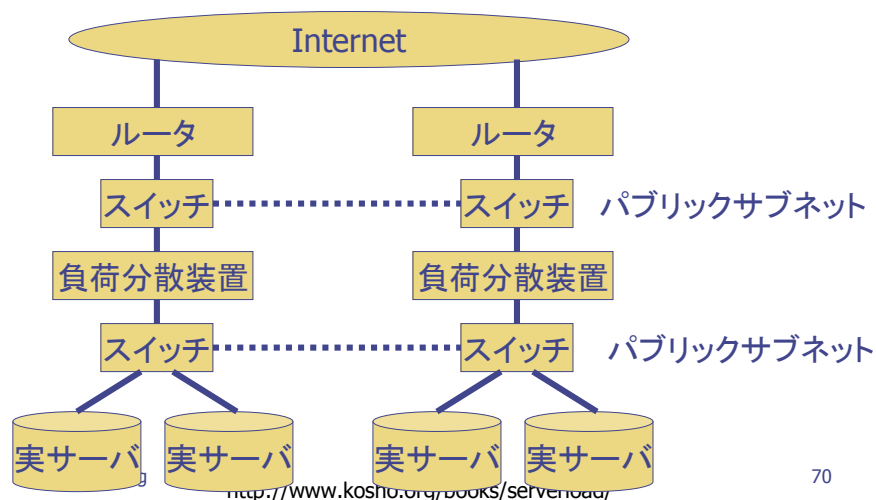
フラットベース

◆ブリッジ、2本腕構成

- 構成図は、NATベース、ルート、2本腕構成と同じ
- 相違点
 - ◆ サブネットが一つ
 - ◆ 負荷分散装置が、実サーバ群のデフォルトゲートウェイではない
 - ◆ ブリッジによるループ構成
 - ブリッジングループが発生する可能性がある
 - STPIによるループ対策が必要

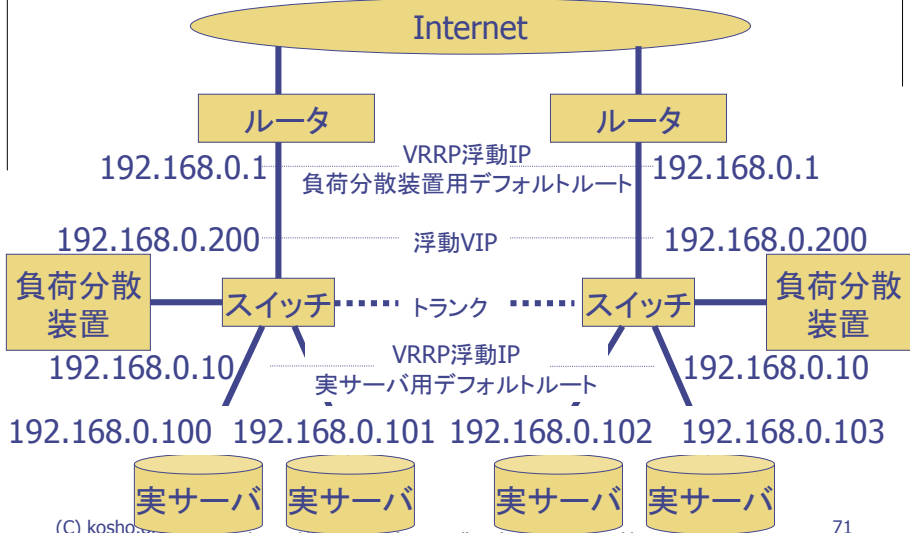
フラットベース

◆ブリッジ、2本腕構成



フラットベース

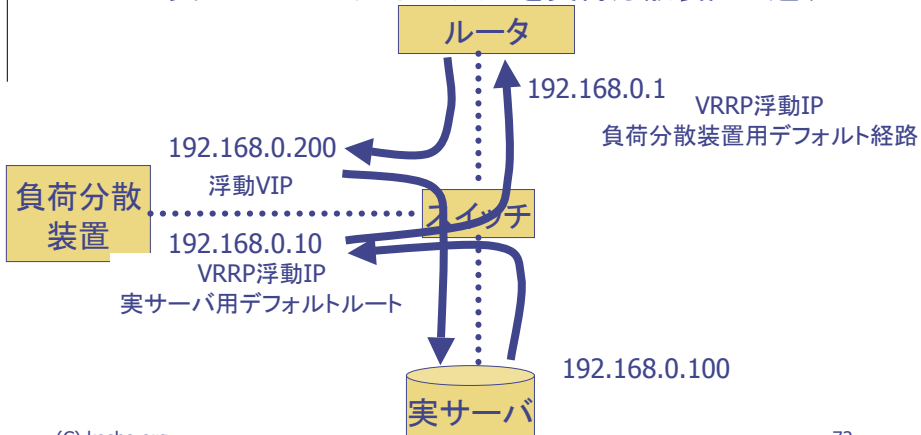
◆ ルート、1本腕構成



フラットベース

◆ ルート、1本腕構成

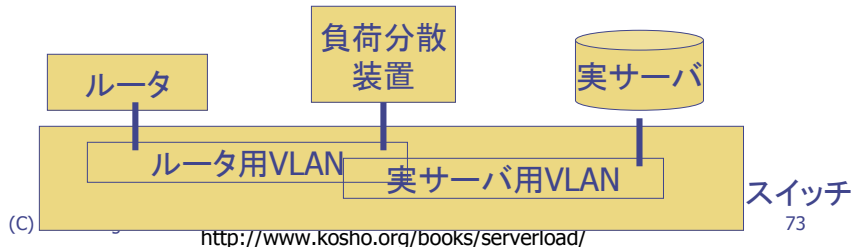
- 負荷分散装置に実サーバのデフォルトルートを向ける
 - ◆ 実サーバから出るパケットを負荷分散装置に通すため



フラットベース

◆ブリッジ、1本腕構成

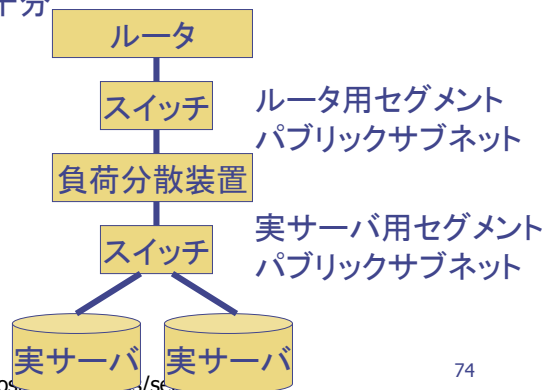
- 1本腕 (インターフェイス1本) のブリッジ
 - ◆ フラットベースではサブネットは一つ
 - 意味がない
 - ◆ 構成不可能
 - 実サーバから出るパケットを受け取れない
- 1本腕 (インターフェイス1本) で2つのVLAN間のブリッジ
 - ◆ ブリッジ2本腕構成と同じ



フラットベース

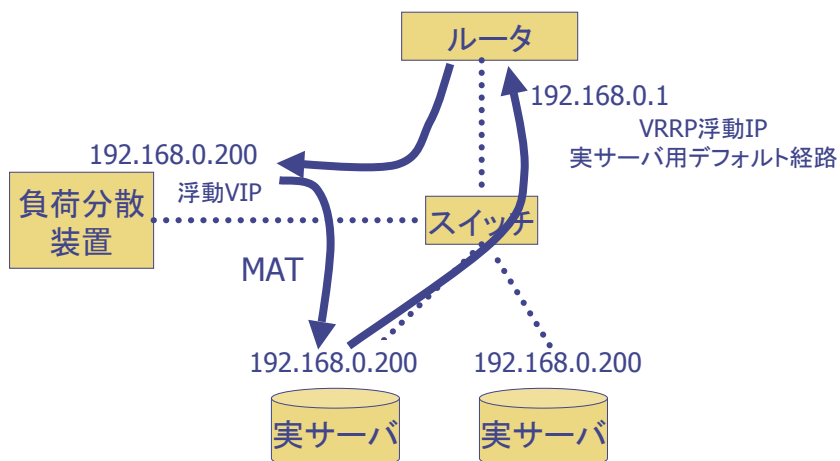
◆ルート、2本腕構成

- 構成として冗長
 - ◆ ルータ用セグメントと実サーバ用セグメントのネットワークアドレスは同一
 - ◆ ブリッジ構成で十分



フラットベース

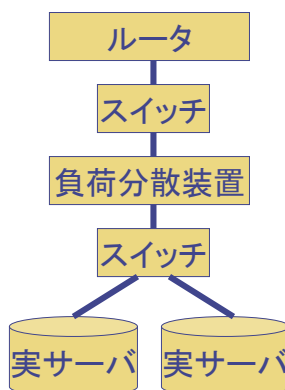
◆ 直接サーバ返答、1本腕構成



フラットベース

◆ 直接サーバ返答、2本腕構成

- 不可能
 - ◆ 実サーバから出るトラフィックが必ず負荷分散装置を通る



アーキテクチャのまとめ

NAT	ブリッジ		不可能
	ルート	1本腕	一般的でない
		2本腕	NAT構成の典型
	直接サーバ返答		トリッキーすぎる
フラット	ブリッジ	1本腕	制限強
		2本腕	ちょっと冗長
	ルート	1本腕	フラット構成の典型
		2本腕	不可能
	直接サーバ返答	1本腕	直接サーバ返答の 典型
		2本腕	不可能