

うちではこんな感じです

～Linuxロードバランサの活用事例

WEB+DB PRESS Tech Meeting



KLab

2007年12月20日

KLab 株式会社

Kラボラトリー

ひろせ まさあき

アジェンダ

- 自己紹介
- インフラ成長日記
 - 2001年から今日まで
- もう少し詳しくロードバランサ
 - 2つのロードバランサ
- まとめ

自己紹介

- ひろせ まさあき (id:hirose31)
- KLab株式会社
 - Kラボラトリー所属 (研究開発部門)
 - DSAS - ネットワーク・サーバインフラ
 - repcached - memcachedでデータレプリケーション
- WEB+DB PRESS
 - Vol.22: MySQL現場指向のレプリケーション詳説
 - Vol.32～37: MySQL5 開拓団
 - Vol.37: Linuxロードバランサ構築・運用ノウハウ
 - Vol.38～: スケーラブルWebシステム工房

今日、お話しすること

- LVS (Linuxロードバランサ)
 - 導入した動機、経緯
 - 活用事例
 - 使ってみての感想
 - よかった点、わるかった点

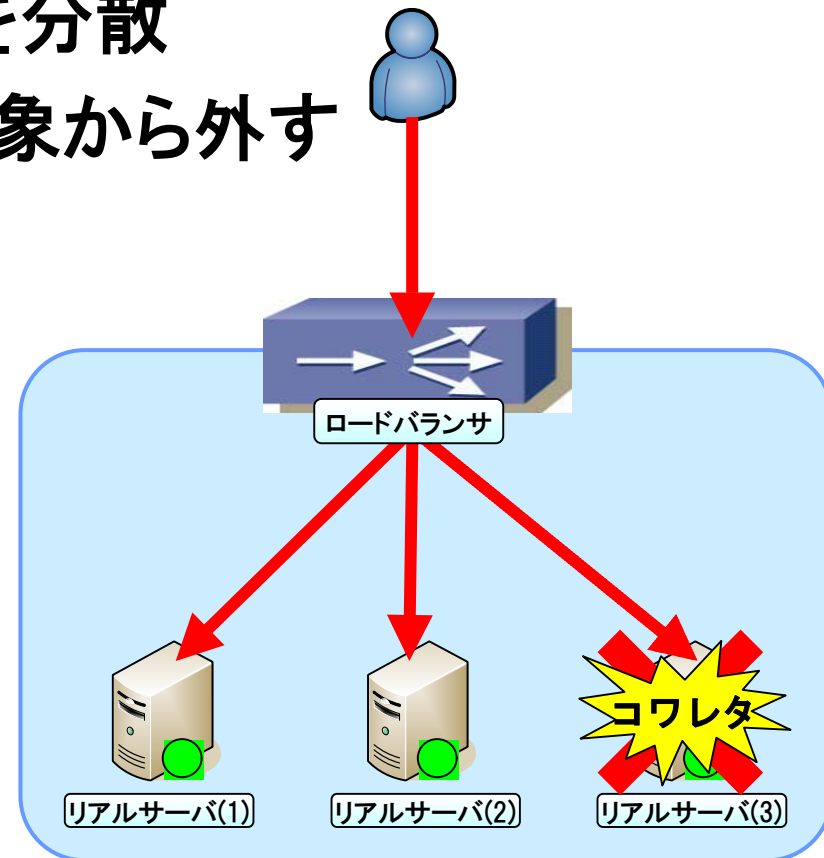
ロードバランサとは？

● 機能

- 複数のサーバにリクエストを分散
- ダウンしたサーバを分散対象から外す

● 効果

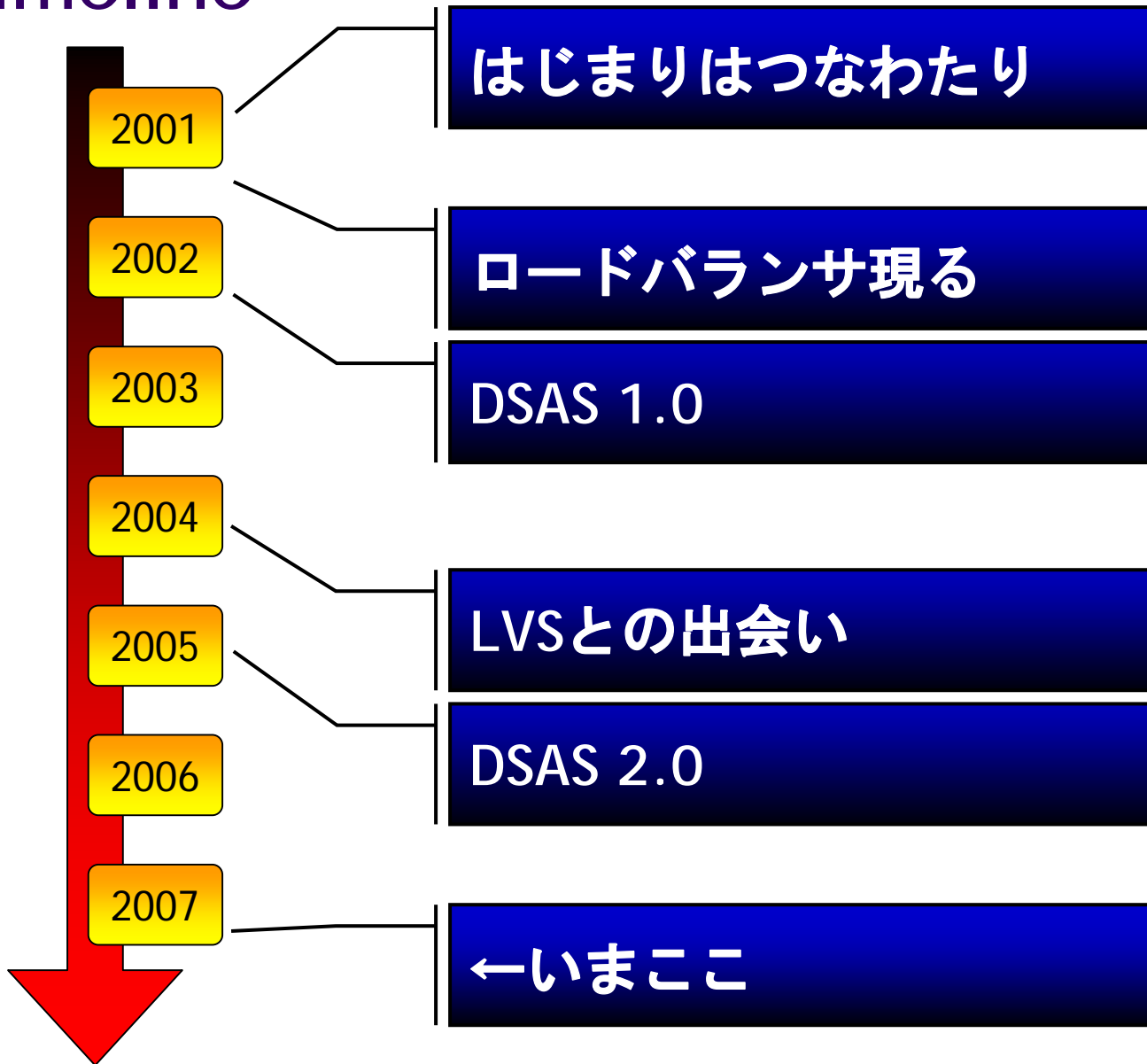
- 大量アクセスをさばける
 - スケールアウト
- サービスの無停止運用



本題 インフラ成長日記

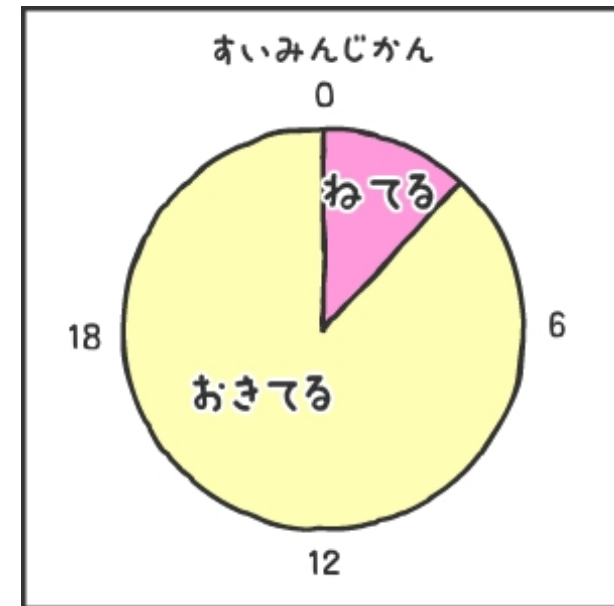
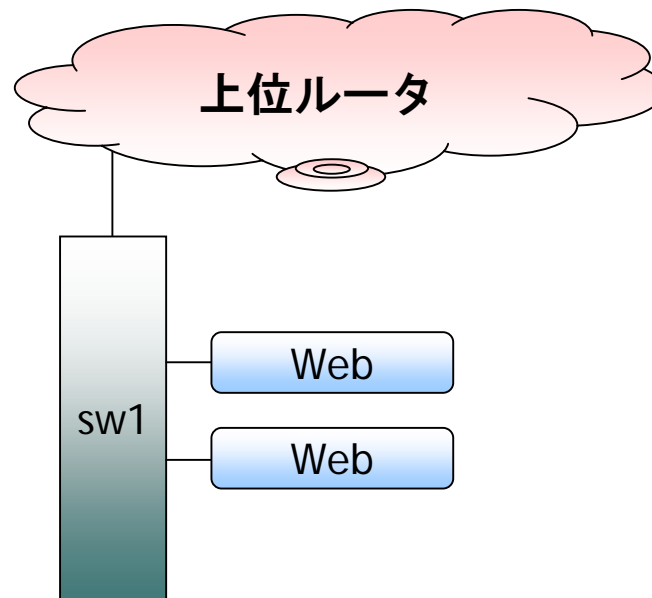


Timeline



はじまりは
つなわたり

- ロードバランサなし
- サーバが落ちたら即、サービス停止
 - 一応、コールドバックアップ機を用意していた

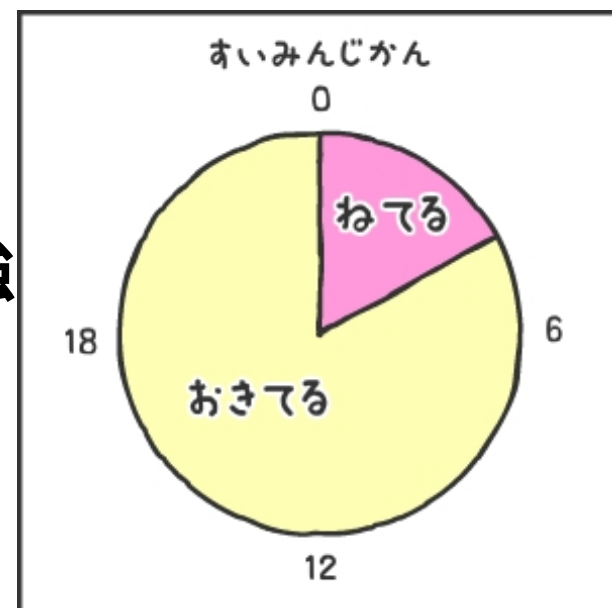


図はイメージです

ロードバランサ現る

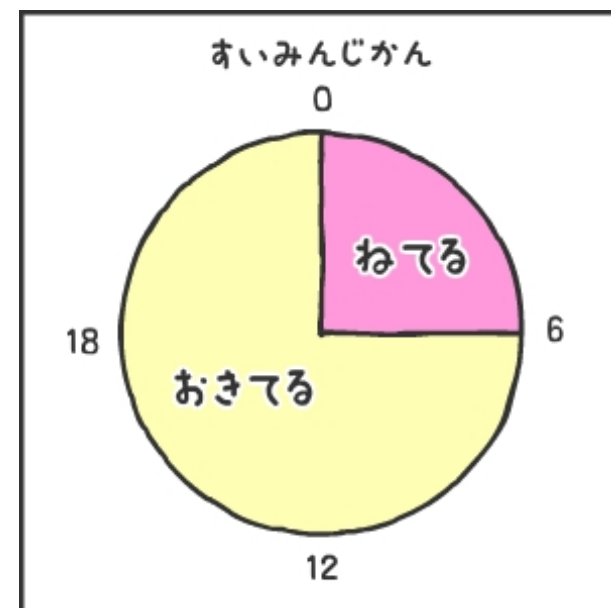
ロードバランサ導入

- **ロードバランサ導入**
 - 商用のアプリケーション製品
 - 最初は1台
 - 2002/10にもう1台導入してアクティブ/スタンバイ構成に
 - ドキュメントがしっかりしていて、サーバロードバランシングの勉強になった
 - チュートリアル、リファレンス



DSAS 1.0

- サーバ増加
- DBをMySQLに乗り換え
- ネットワーク構成の変更
 - 冗長化 - bonding
- 常に改良を続ける
 - =いじりほうだいw
- 便宜的に『DSAS』と命名



LVISとの出会い

開発用環境にLVSを導入

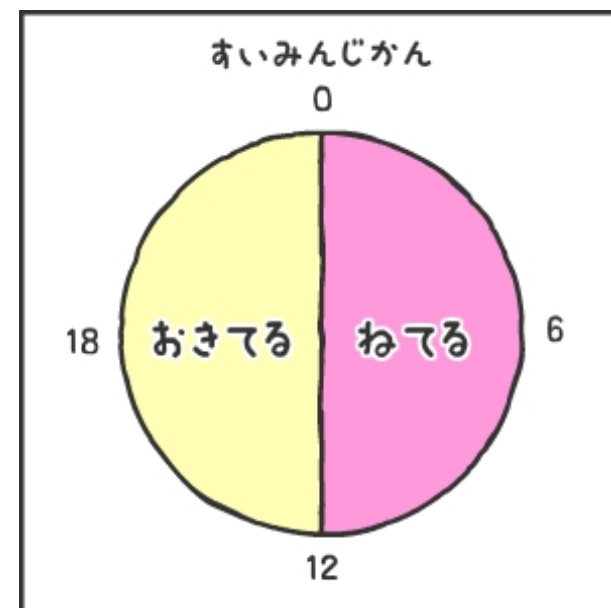
- **動機**

- Linuxでロードバランサが作れるらしいよ
- mjd!! いじってみたい!!! 11
- 2004/07ごろからLVSの調査、検証をはじめた
- 2004/12に開発用の環境にLVSを導入
 - 開発環境でも分散環境のテストができるようになった

DSAS 2.0 (笑)

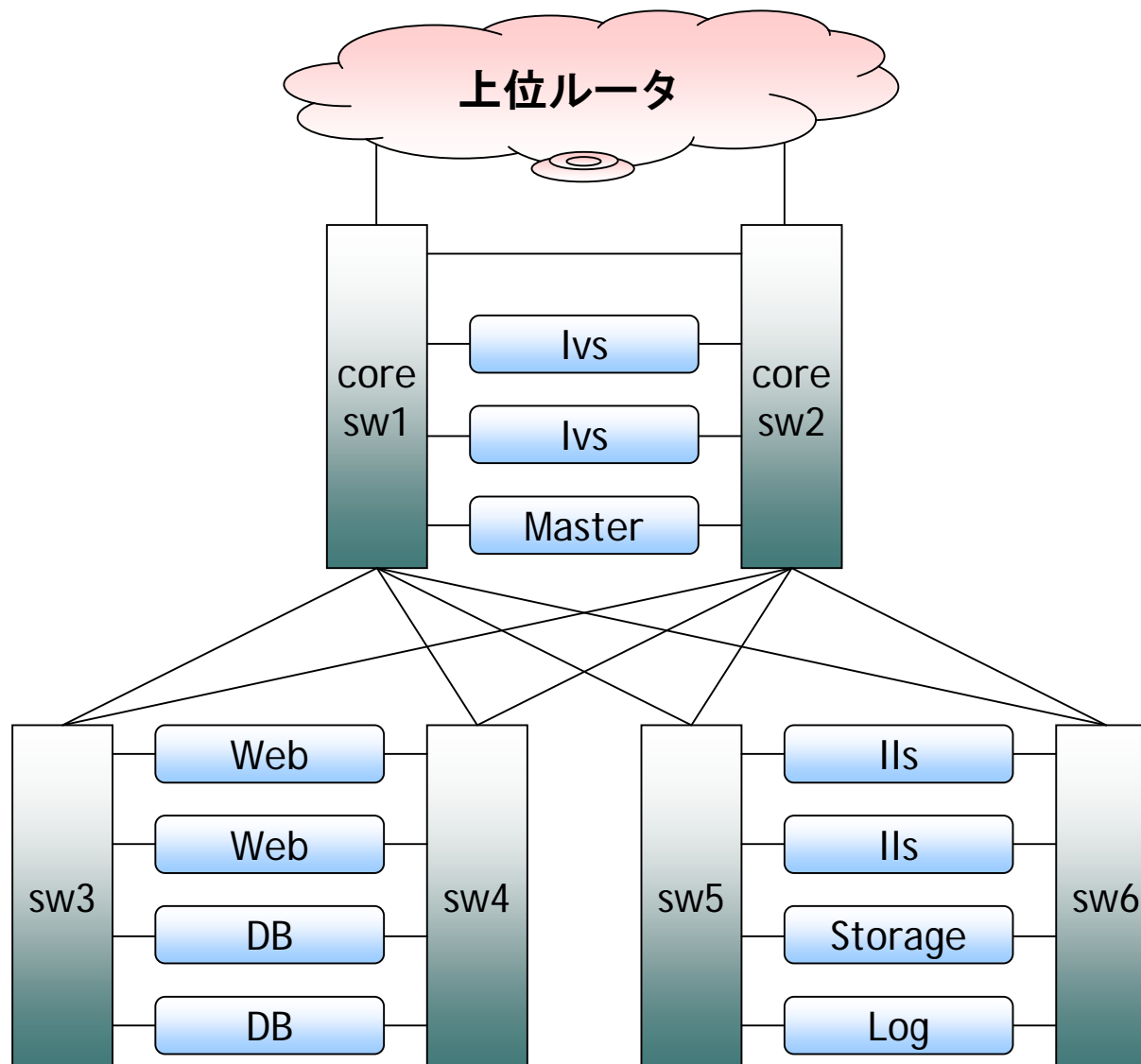
DSAS 2.0の構築開始

- もちろん、ロードバランサはLVSで
 - 本番環境で使用開始
- DSAS 1.0の特長を残しつつ新しい試みを
 - メタ情報の管理の一元化
 - サーバロール: MATRIX
 - ネットワークブート
 - ストレージサーバ
 - DRBD + keepalived
 - ネットワークの冗長構成
 - bonding、RSTP



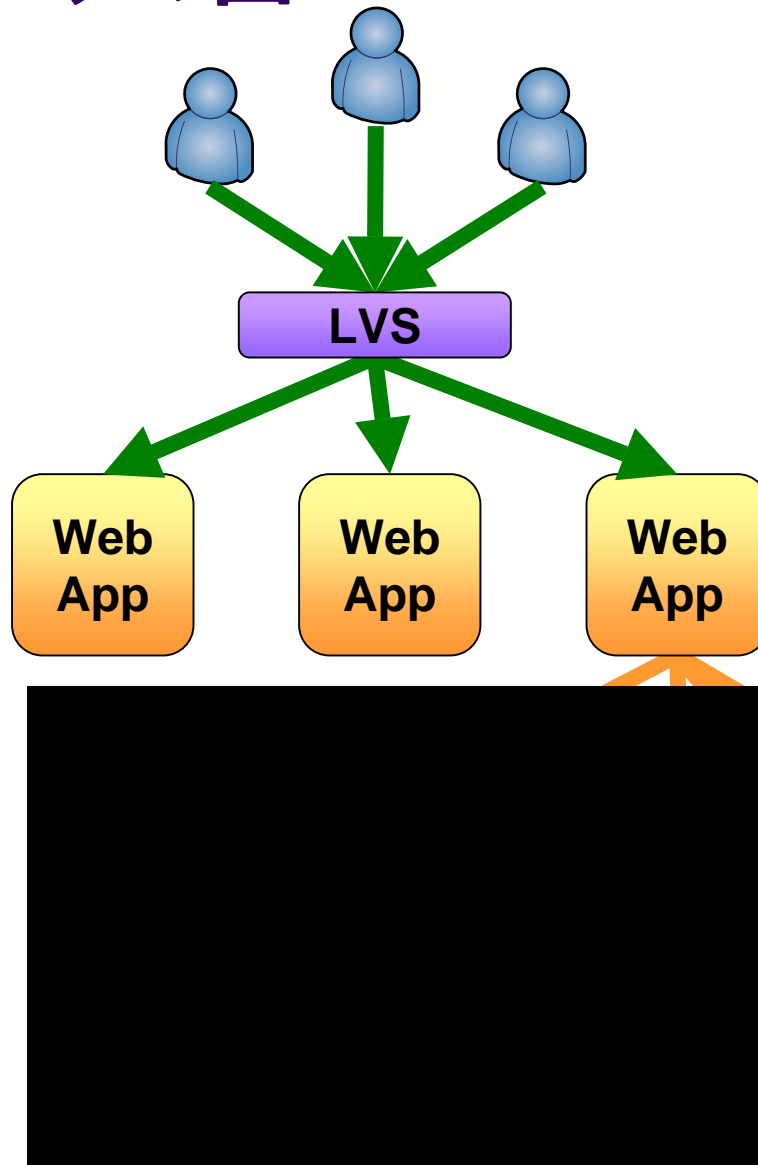
今、現在

現在の概略図



もう少し詳しく ロードバランサ

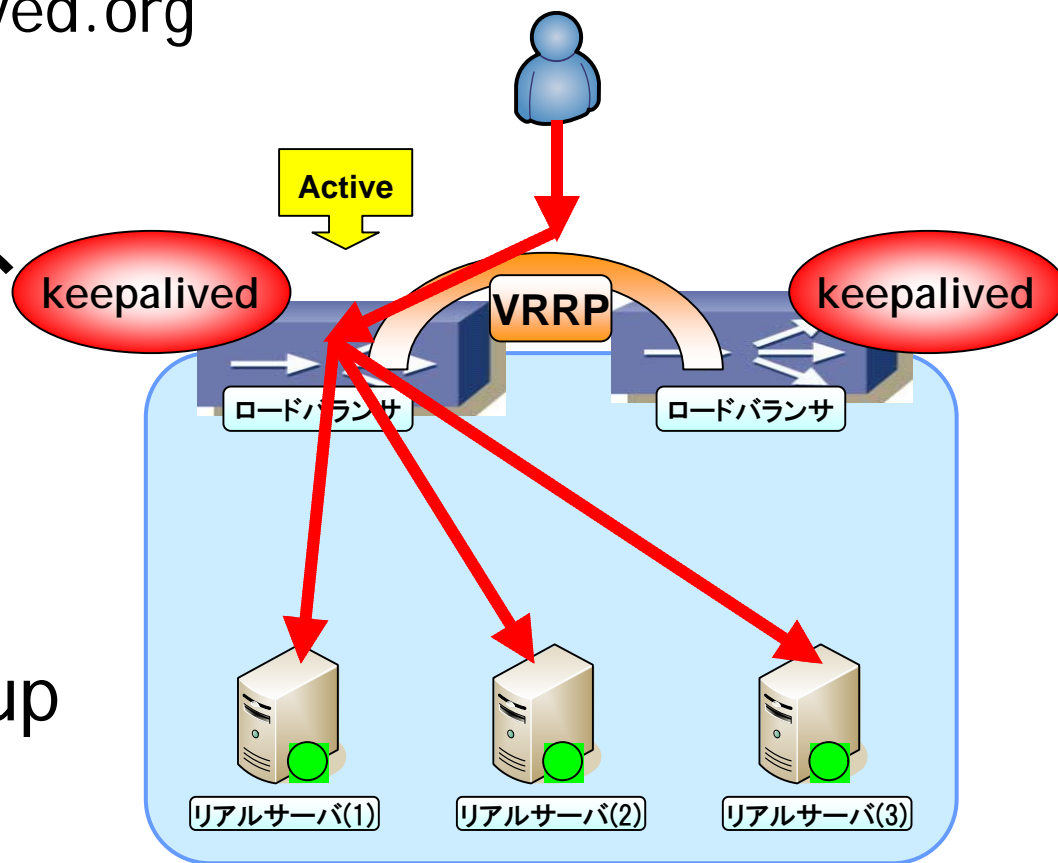
ロードバランサの図



ロードバランサ やっていること

- やっていること=ロードバランス
 - HTTP ←これがほとんど
 - SMTP
 - FTP
 - XMPP (Jabber)
 - など

- keepalivedを使用
 - <http://www.keepalived.org>
- L4のロードバランサ
 - IPVSの制御
 - サーバの死活監視
- VRRPによる冗長化
 - 2台でActive/Backup



ロードバランサ

keepalivedの設定ファイル (1/2)

- keepalived.conf
- 2500行

たくさんの
仮想サーバ

もっとたくさんの
リアルサーバ

1つ(か2つ)の
VRRPの設定

```
vrp_instance VE {
  state BACKUP
  interface eth0
  garp_master_delay 5
  virtual_router_id 1
  priority 100
  nopreempt
  advert_int 1
  authentication {
    auth_type PASS
    auth_pass himitu
  }
  virtual_ipaddress {
    10.10.31.10/24 dev eth0
    192.168.31.10/24 dev eth1
  }
}
```

```
virtual_server_group HTTP100 {
  10.0.0.100 80
}

virtual_server_group HTTP100 {
  delay_loop 3
  lvs_sched rr
  lvs_method NAT
  protocol TCP
  virtual_host example.org
  sorry_server 192.168.31.100 80
  real_server 192.168.31.101 80 {
    weight 1
    inhibit_on_failure
    HTTP_GET {
      url {
        path /health.html
        status_code 200
      }
      connect_timeout 3
    }
  }
  real_server 192.168.31.102 80 {
    weight 1
    inhibit_on_failure
    HTTP_GET {
      url {
        path /health.html
        status_code 200
      }
      connect_timeout 3
    }
  }
}
```


- 構成情報を元にkeepalived.confを生成

```
w101: usagi
w102: usagi kame
w103: [NANDEMOYA]
w104: kame
w105: kame
w106: (Iv1)
w107: (Iv2)
w108: (I11)
w109: (I12)
w110: (db101) (db100)
w111: (db102)
w112: [FREE] usagi kame
```

MATRIX

```
[% FOREACH R=REAL -%]
  real_server [% R %] [% real_port %] {
    weight 1
    inhibit_on_failure
    HTTP_GET {
      url {
        path [% HTTP_GET.path %]
        status_code [% ... status_code %]
      }
    }
    connect_port [% ... %]
    connect_timeout [% ... %]
    nb_get_retry [% ... %]
    delay_before_retry [% ... %]
  }
[% END -%]
```

テンプレート

```
vrrp_instance VE {
  state BACKUP
  interface eth0
  garp_master_delay 5
  virtual_router_id 1
  priority 100
  nopreempt
  advert_int 1
  authentication {
    auth_type PASS
    auth_pass himitu
  }
  virtual_ipaddress {
    10.10.31.10/24
  }
}

virtual_server_group ...
  10.0.0.100 80
}

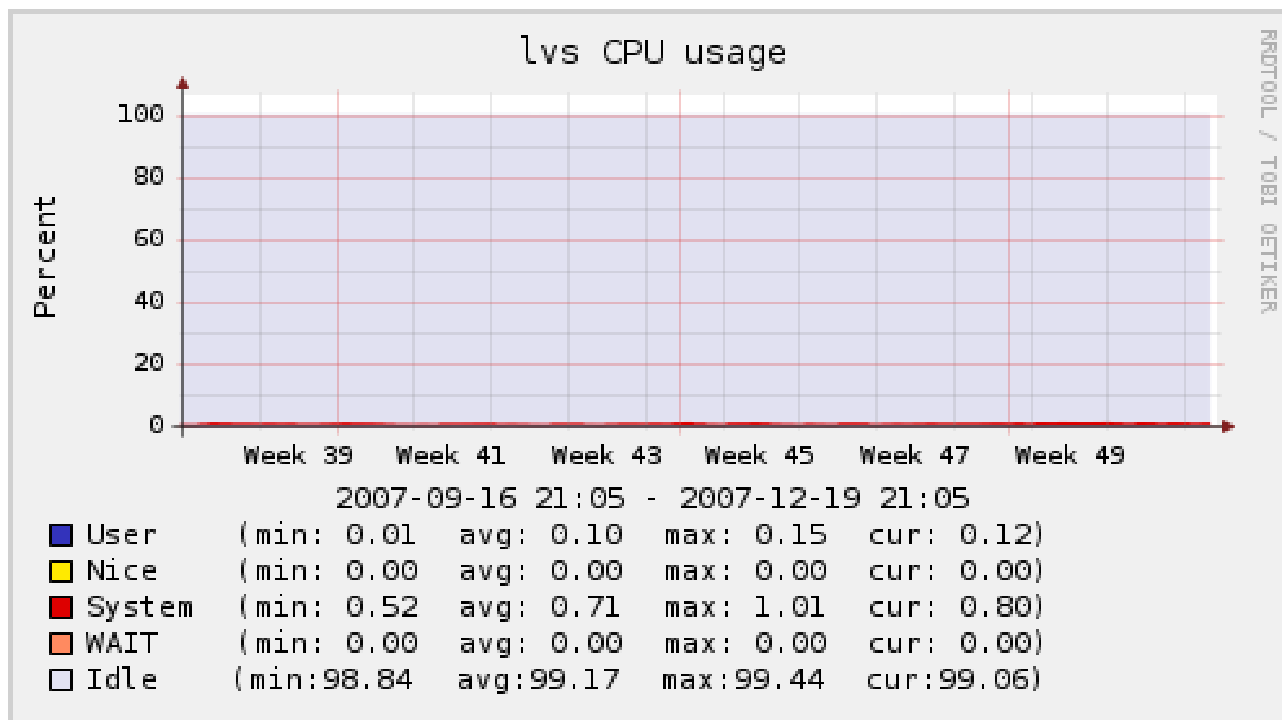
virtual_server_group ...
  delay_loop 3
  lvs_sched rr
  lvs_method NAT
  protocol TCP
  virtual_host example...
  sorry_server 192.16...
  real_server 192.16...
```

keepalived.conf

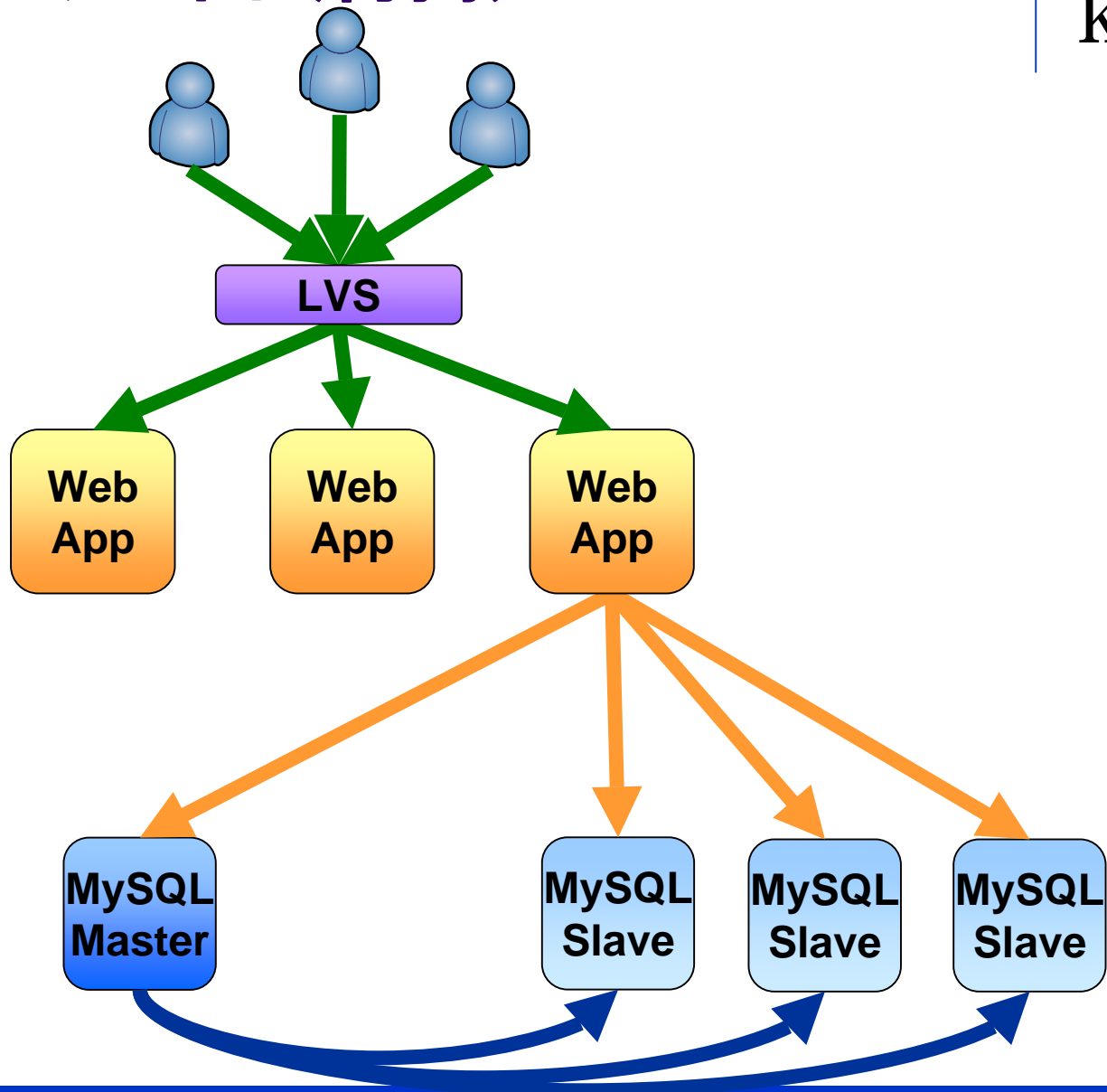


ロードバランサ LVSの負荷は？

- CPU使用率
 - Sys < 1.00%
 - User < 0.15%



ロードバランサの図 (再掲)





MySQLスレーブへの分散アクセス

- Webアプリでやらなければならないこと
 - スレーブ群のホスト名の一覧をもっている
 - 分散先を決定するロジック
 - ラウンドロビン？
 - コネクションの少ない順？
 - スレーブの死活監視
 - ダウンしているスレーブには分散しないように
- ➔ アプリはスレーブの状況を意識している必要がある

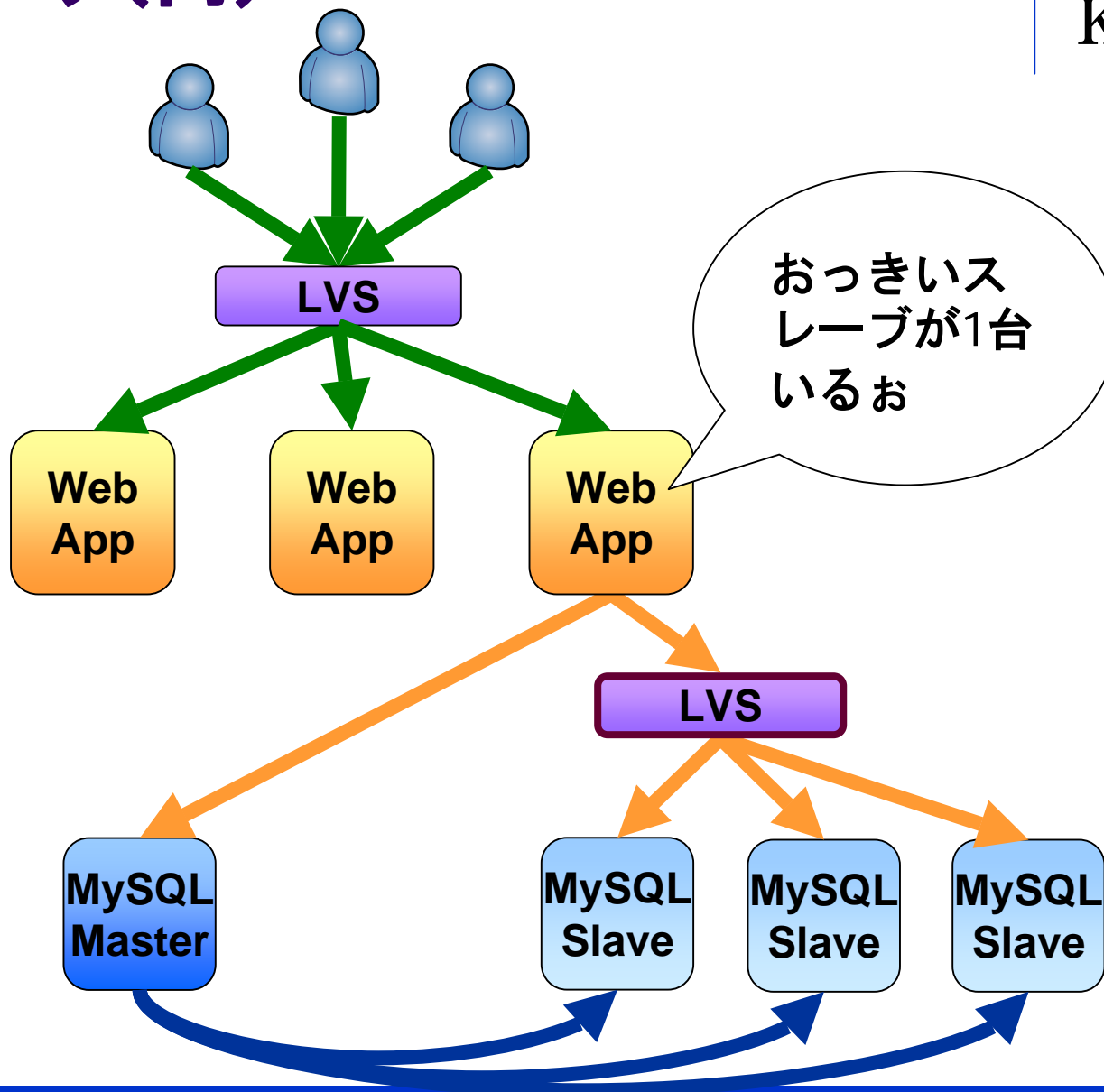
分散と
いえば...





LVSがあ
るじゃない
い!!!

ロードバランサ(内)



ロードバランサ(内)

- そのほかの使い道

- DNS
- SMTP
- Proxy (外部へのHTTPアクセスのため)

➔ 内部にもロードバランサがあるとなにかと便利！！

今日のお話は
このへんまでで...



まとめ (1/2)

- LVSのいいところ
 - 金銭的成本が低い
 - 金銭的な導入障壁が低い
 - ロードバランサをいくつも導入できる - 外部・内部、開発環境
 - 中身までいじくれる
 - 調査 - ソース読む、tcpdump
 - 足りないものは実装
 - 安定している
 - VRRPで冗長化
 - 3年ぐらい運用した限りでは、へんな落ち方をしたことはない
- LVSのいまいちなところ
 - 設定インターフェースが貧弱
 - サポートがない



まとめ (2/2)

- **なぜそこまでやるのか？**
 - ぐっすりねたいから！！
 - いじっててたのしいから！！！！
- **オープンソースでいろいろできます**
 - ロードバランス
 - 冗長構成
 - などなど...
 - **はじめは最小構成でおk**
 - **できるところから手をつけて育てればいい**

LVS、keepalivedの設定方法は...

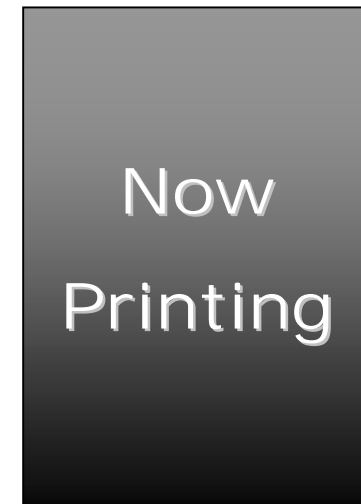
- WEB+DB PRESS Vol.37



- Linuxロードバランサ構築・運用ノウハウ
 - 1章:サーバ負荷分散概論
 - 2章:LVSで実現するロードバランサ
 - 3章:ロードバランサの冗長化
 - 4章:負荷分散システム運用のコツ

ネットワーク・サーバまわり全般の話は...

- 書籍(書名未定) 鋭意執筆中！
 - 『サーバ24時』(仮)
 - 『今夜できるサーバ』(仮)
 - 『スケラブルHP(笑)』(仮)
- 執筆陣
 - KLab/DSASチーム
 - はてな



ご清聴
ありがとうございました

