

DSASのいろいろ

第9回 オープンソーステクノロジー勉強会



KLab

2007年2月2日

KLab 株式会社
Kラボラトリー
ひろせ まさあき

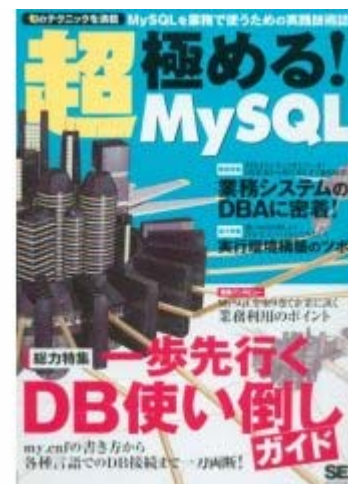
自己紹介

個人的な自己紹介

- 名前: ひろせ まさあき
- id:hirose31
で、はてダラってます
- あと...
 - 一応CPAN author (1コだけ...)
 - 一応PEAR author (これも1コだけ...)

個人的な自己紹介

- ほかに執筆活動もちょこちょこと
 - 『WEB+DB PRESS』誌にMySQLの連載
- 『超・極める! MySQL』(ムック、共著)に寄稿



勤め先

- 六本木にある
- KLab株式会社（くらぶかぶしきがいしゃ）
- （旧社名：ケイ・ラボラトリー）
- の「Kラボラトリー」という研究開発部署に所属

みじかくいうと

ろっぽんぎのくらぶ
につとめてます

たまたに

六本木のホストクラブ と誤解されます

弊社にお越しの際は
注意してください

そんなKLabの事業内容

- 主に携帯電話関連の事業
 - コンテンツ企画開発
 - コンサルティング
 - メディア事業
 - EC事業
- セキュリティ事業
 - VPN
 - 個人情報監査

ところでDSASってナニ？



DSAS とは ...

- Dynamic Server Assign Systemの略
- KLab独自のコンテンツサービス用のネットワーク/サーバインフラの呼び名
- 特徴
 - オープンソースベースのシステム
 - 単一故障点のないシステム
 - アクセスの増加に柔軟に対応できるシステム
 - ((後ほど詳しく))

で、わたしのおしごと

- そんなDSASチームで
 - ネットワーク／サーバの設計、構築、運用など
- あと、DSASチームでブログもやっています
 - <http://dsas.blog.klab.org>

DSASブログ過去の人気エントリをご紹介します

- 「こんなに簡単！ Linuxでロードバランサ」
 - 今日の話題のひとつ、LVSについて書いたもの
- 各方面から反響が！

そろそろ本題に





今日の内容

- (1) DSASの特徴の紹介
 - 設計思想、全体構成など
- (2) DSASの構成要素の紹介
 - ロードバランサ
 - LVS, keepalived
 - ネットワークブートの活用
 - 故障に強いストレージサーバ
 - DRBD
 - NICの二重化
 - bonding
 - シリアル接続 温故知新
 - サーバリソースの見える化
 - ganglia

DSASの特徴

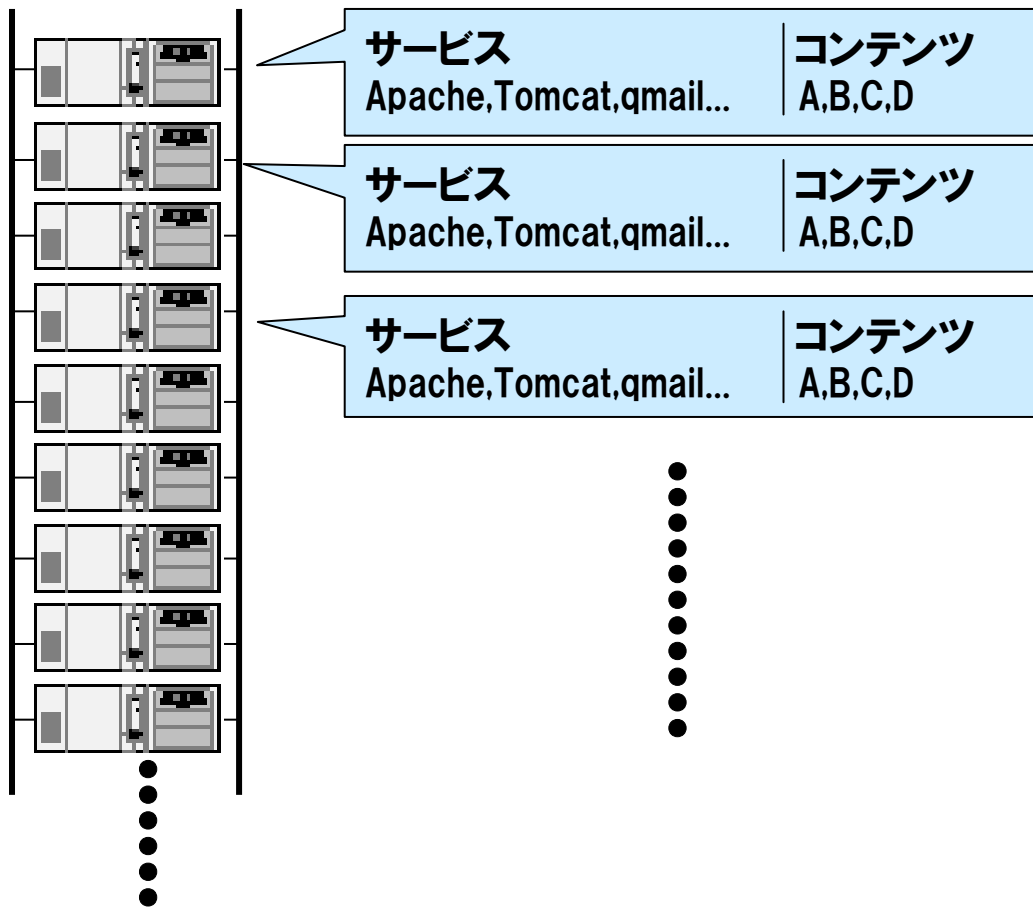
DSASの特徴(1)

オープンソーススペース

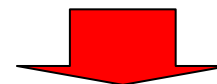
- OSはLinux
- ロードバランサはLVS (Linux)
- WebサーバはApache、thttpd
- アプリケーションサーバはTomcat、PHP
- DBサーバはMySQL
- その他もろもろも
 - qmail、djbdns、memcached、sqlrelay、stone
- ハードウェアはPCサーバ

DSASの特徴(2)

どのサーバもディスクの中身が同じ



すべてのハードディスクの内容が同じ



どのサーバも全てのコンテンツ、
全てのサービスを提供可能

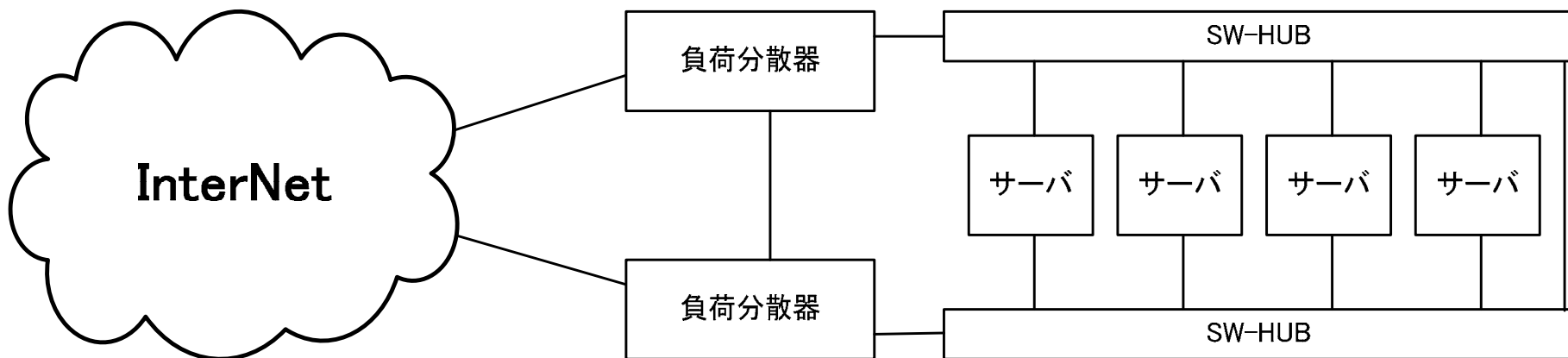


フラグをたてるだけで
各サーバの役割が早変わり

DSASの特徴(3)

単一故障点がない構成

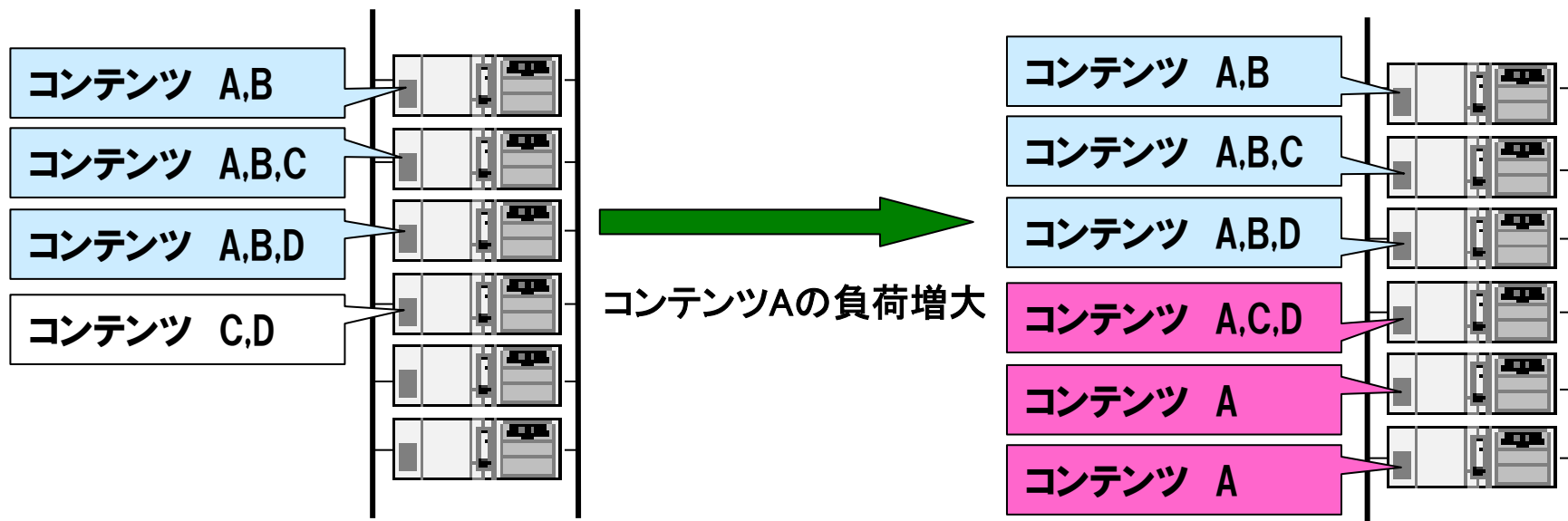
- 上流の対外線は2本引いている
- ロードバランサはアクティブ/バックアップ構成
- L2SWも冗長構成 (RSTP)
- 各サーバのNICも2個 (bonding)



DSASの効果(1)

アクセス増減に柔軟な対応が可能

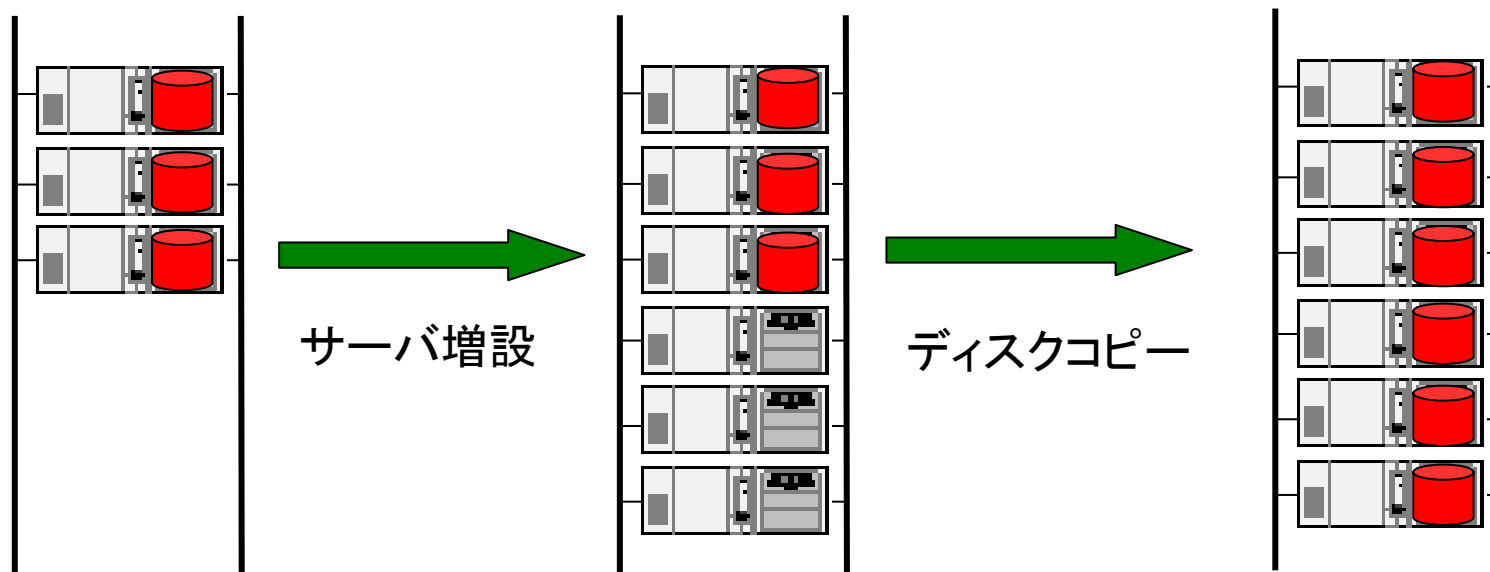
- ディスクの中身が同じなので、サーバ割り当てを操作するだけで増強完了
 - TV CMやメルマガなどの急なアクセス増加にも即時に対応可能
- 逆に、アクセスが少ないコンテンツはサーバの相乗りも可能



DSASの効果(2)

故障機の復旧やサーバの増設が容易

- ディスクをコピーするだけでサーバ増設が完了
- サーバが故障した場合も迅速な復旧が可能
- PCサーバを利用しているのでパーツの手配も容易



DSASの効果(3)

サービスインまでの期間短縮が可能

DSASを利用しない場合、サーバの機種選定や導入において多大なコストがかかります

【一般的な新規開発フロー】

1. 要件定義
2. 仕様作成
3. サーバスペック計算
4. ネットワーク設計
5. ポリシー設計
6. 機種選定
7. 機材手配
8. インストール作業
9. 設定・動作確認

- ・ヘルスチェック等も独自の実装が必要
- ・システム毎に設計がまちまちになりそう
- ・バージョン管理も困難
- ・セキュリティホールが生まれやすい
- ・壊れたときの対応が大変
(これらに気がつくのはできてからの場合が多い)



DSASを使うと3～8の作業は不要



所定の手続きだけで設定追加も簡単



SEはアプリ側の設計に専念できそう

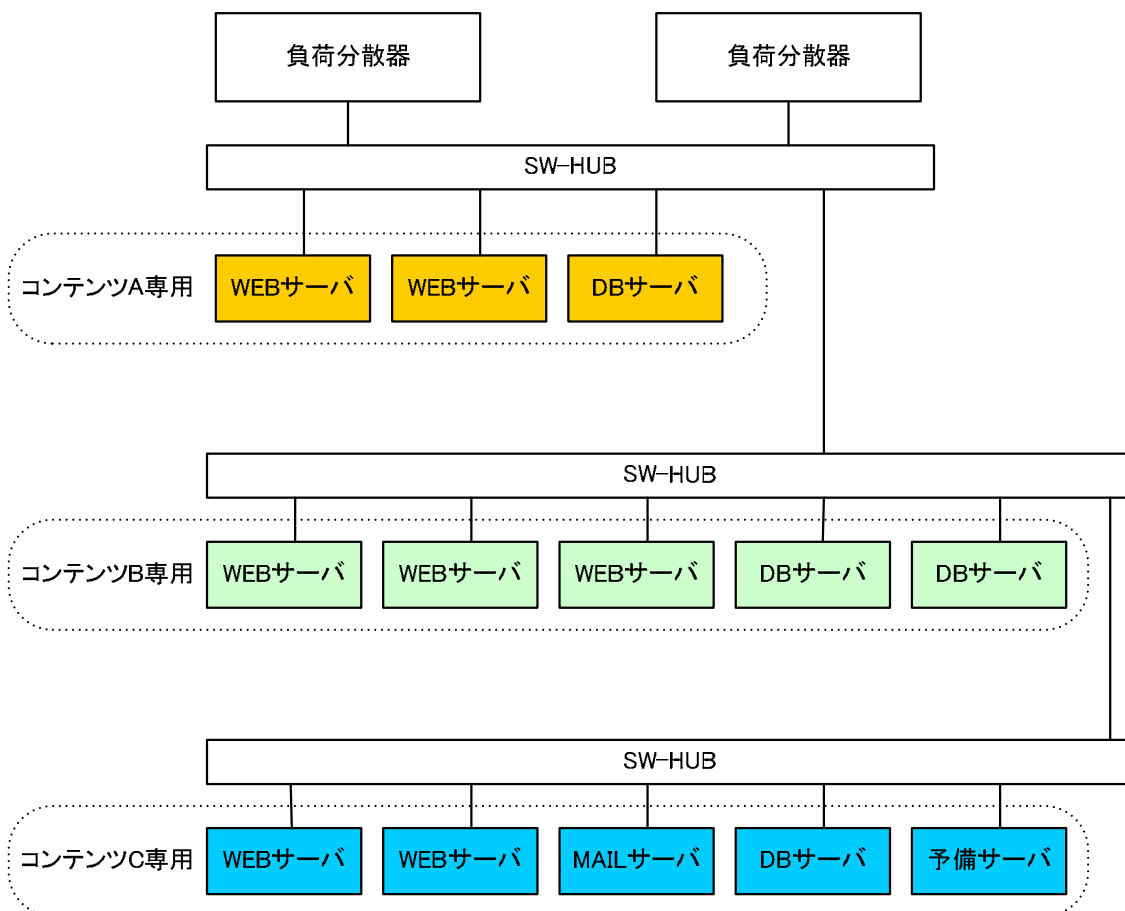


信じられない開発期間でSINできそう

DSASの効果(4)

案件担当者をインフラの悩みから解放

<一般的なサーバ構成>

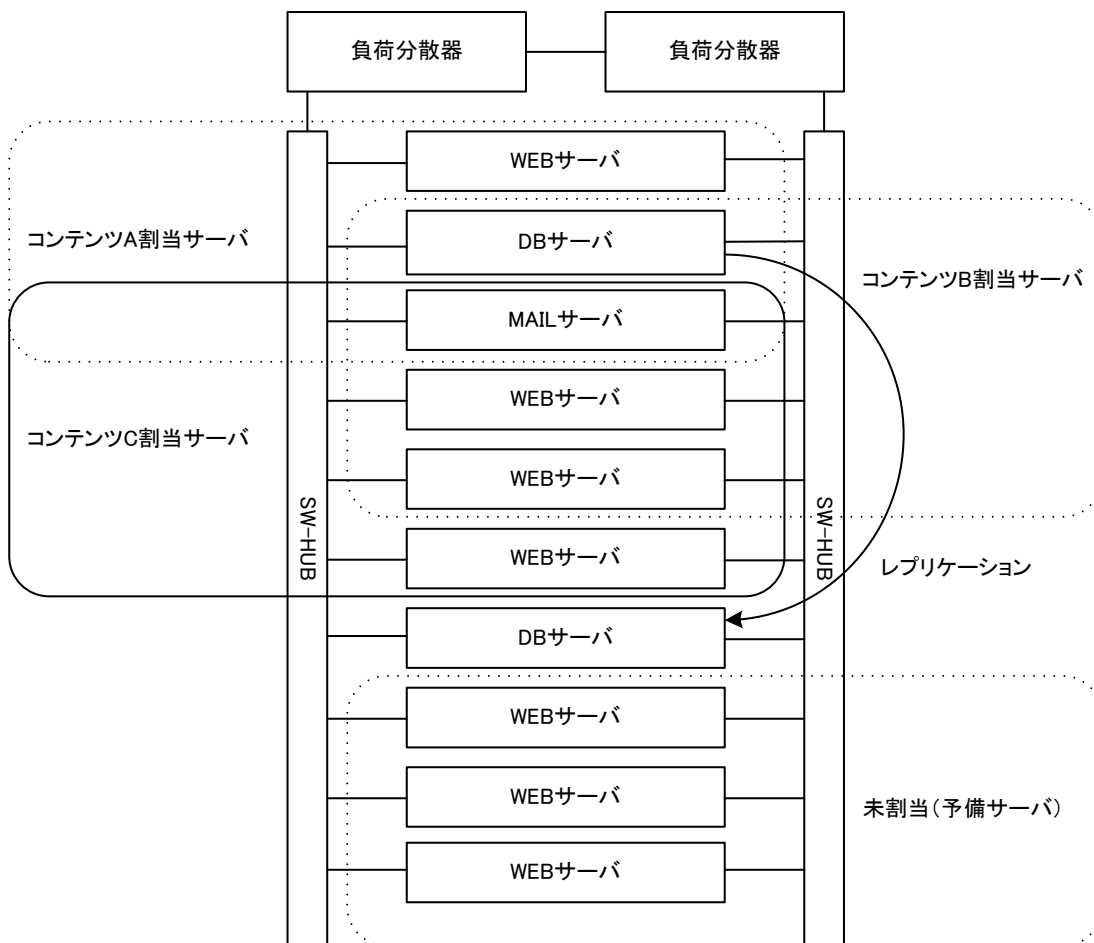


- ・コンテンツ毎に専用サーバ群が必要
- ・新規案件毎にシステム増設が必要
- ・スケーラビリティがない
- ・故障時の対応が困難かつ非効率
- ・案件毎にシステム管理者がいなきゃだめ
- ・ネットワーク構成が複雑になっていく
- ・バージョン管理がとても困難
- ・セキュリティホールがしやすい

DSASの効果(4)

案件担当者をインフラの悩みから解放

<DSASを使うと>



- ・案件担当者がサーバ増設を考えなくてもよい
- ・案件担当者がネットワーク設計しなくてもよい
- ・軽いコンテンツ同士は掛け持ちが可能
- ・スケーラブル
- ・サーバが1台壊れてもさほど困らない
- ・数人のシステム管理者でメンテナンス可能
- ・予備サーバはどのコンテンツでも提供可能
- ・一貫したセキュリティ対策が可能
- ・バージョン管理の一元化が可能

以上がDSASの特徴です
続いては
DSASの中身（構成要素）
をいくつか紹介します

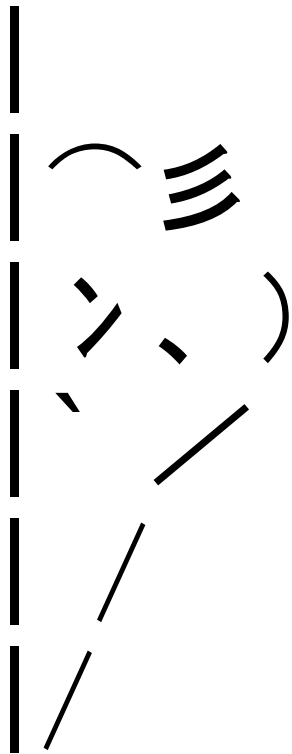
Linuxでロードバランサ

**ロードバランサって
高いですねー**



高嶺の花★ロードバランサ

- 箱物のロードバランサ
 - 1台で数百万円～数千万円
 - 二重化＋保守費用
 - コストは2倍以上
- カジュアルに導入できない
 - 金銭的成本が障壁に



それ、LVISでできるよ

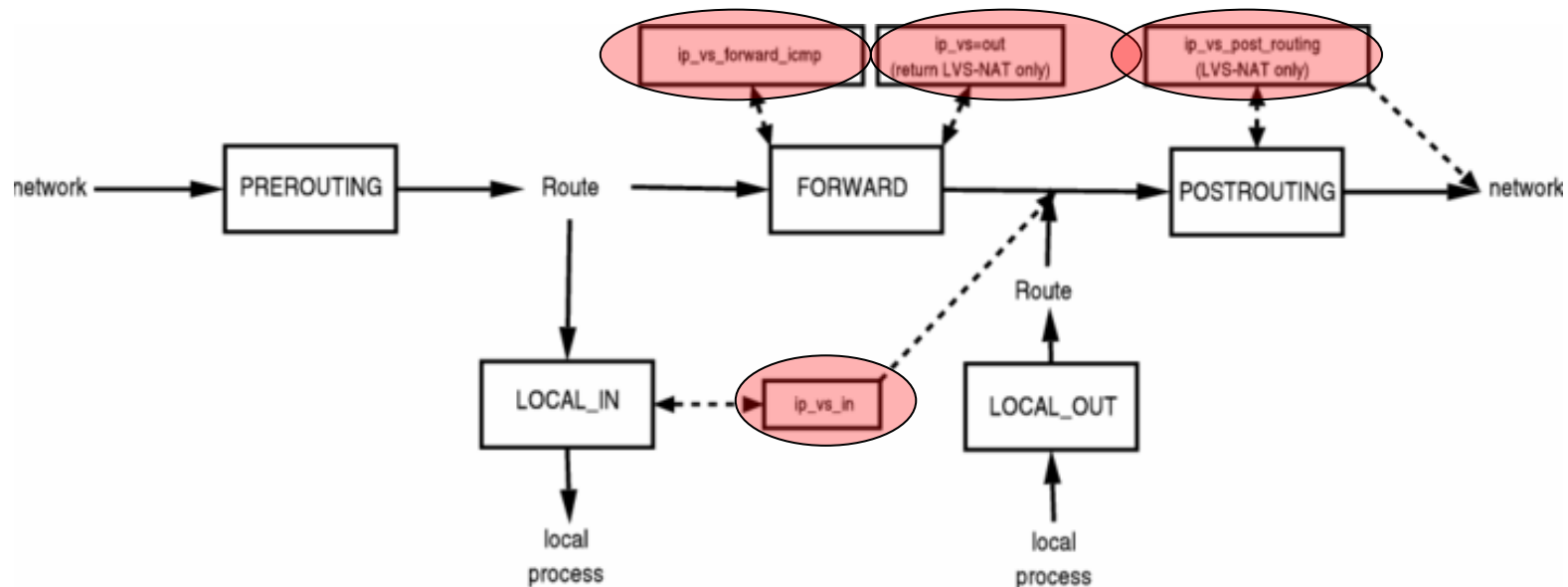


LVSとは

- LVS – Linux Virtual Server Project
 - Linuxで、高性能かつ高可用性をもったサーバシステムを作ろうというプロジェクト。
- IPVS – IP Virtual Server
 - LVSプロジェクトの成果物のひとつ
 - LinuxをL4ロードバランサに仕立て上げることができる！

IPVSとは

- Netfilterと連携して動く
- kernel module
 - カーネル空間で動くので速い



Linux Kernel Netfilter Hooks and LVS

Horms <horms@verge.net.au>, v0.1.9-1, October 2003



IPVSの設定

- 設定はipvsadmコマンドで
- www.example.org:80宛てのを ← VIP
 - 192.168.31.101
 - 192.168.31.102
- に分散するには

```
# ipvsadm -A -t www.example.org:80 -s lc
# ipvsadm -a -t www.example.org:80 -r 192.168.31.101 -m
# ipvsadm -a -t www.example.org:80 -r 192.168.31.102 -m
```

- こんだけ。(IPVSの設定は)

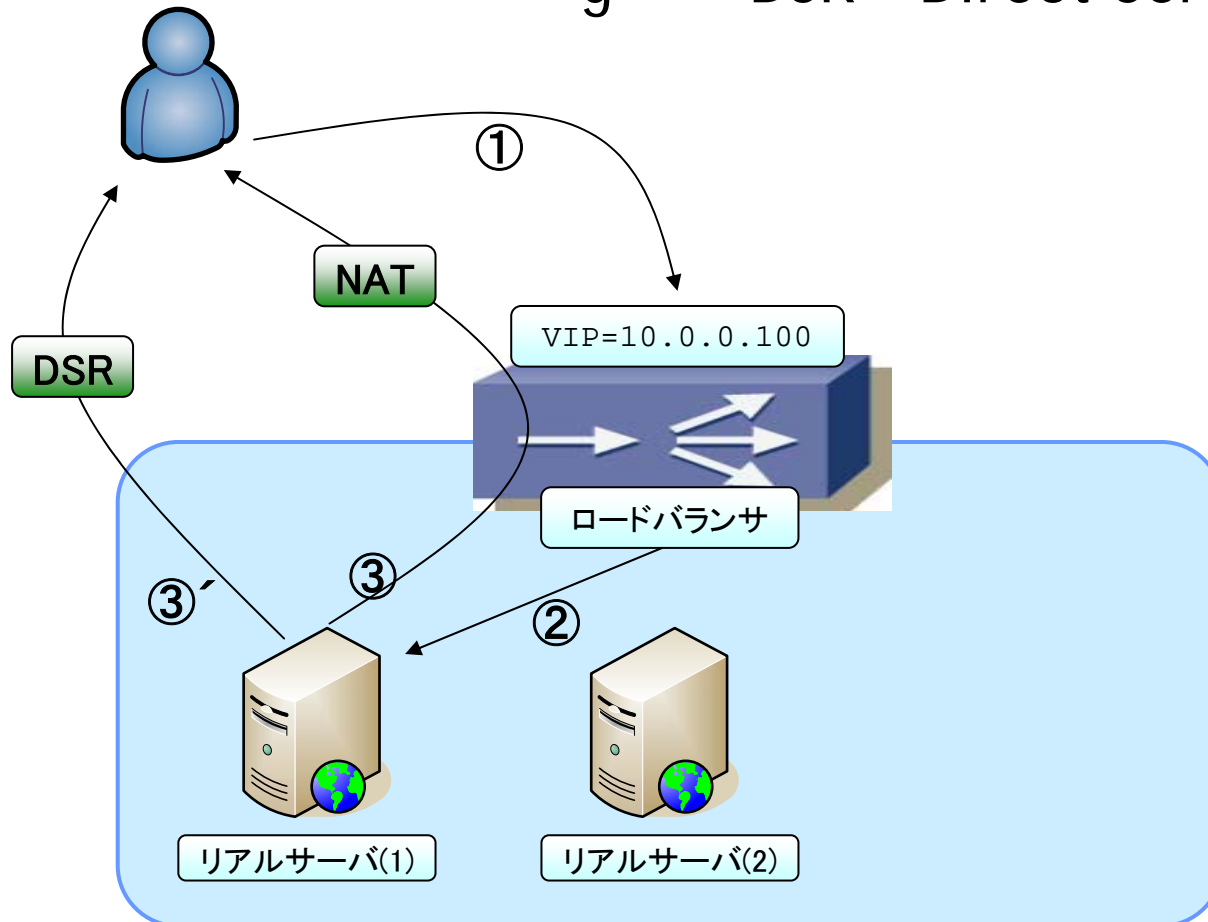


IPVSの分散方法

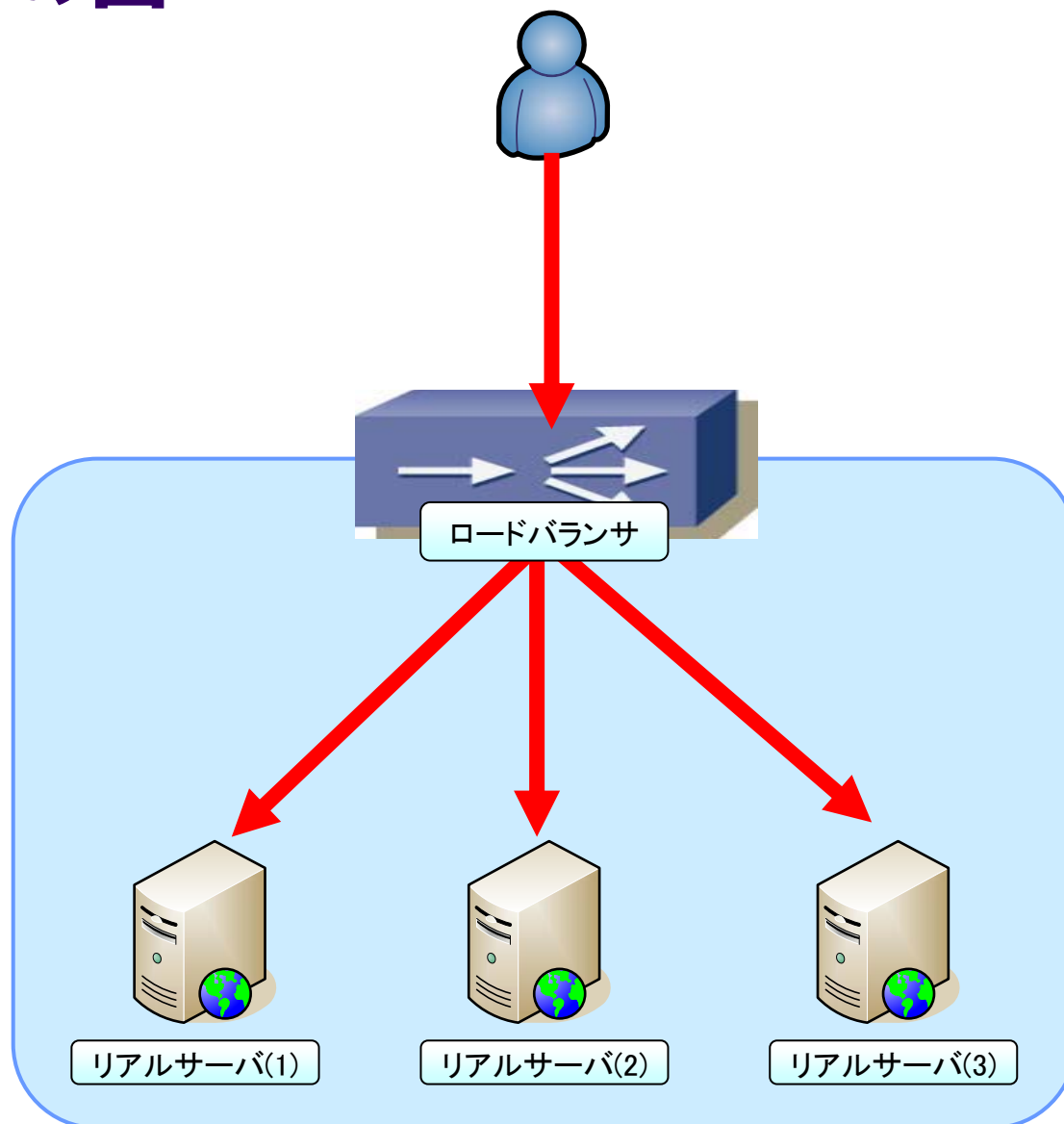
- `ipvsadm -s <scheduling_method>`
- `rr` ラウンドロビン
- `wrr` ラウンドロビン(重み付けあり)
- `lc` 最小接続
- `wlc` 最小接続(重み付けあり)
- `sh` 始点アドレスのハッシュ
- `dh` 終点アドレスのハッシュ
- ほかにもいろいろ

IPVSでDSRもできます

- -m NAT
- -g DSR - Direct Server Return



ここまでの図

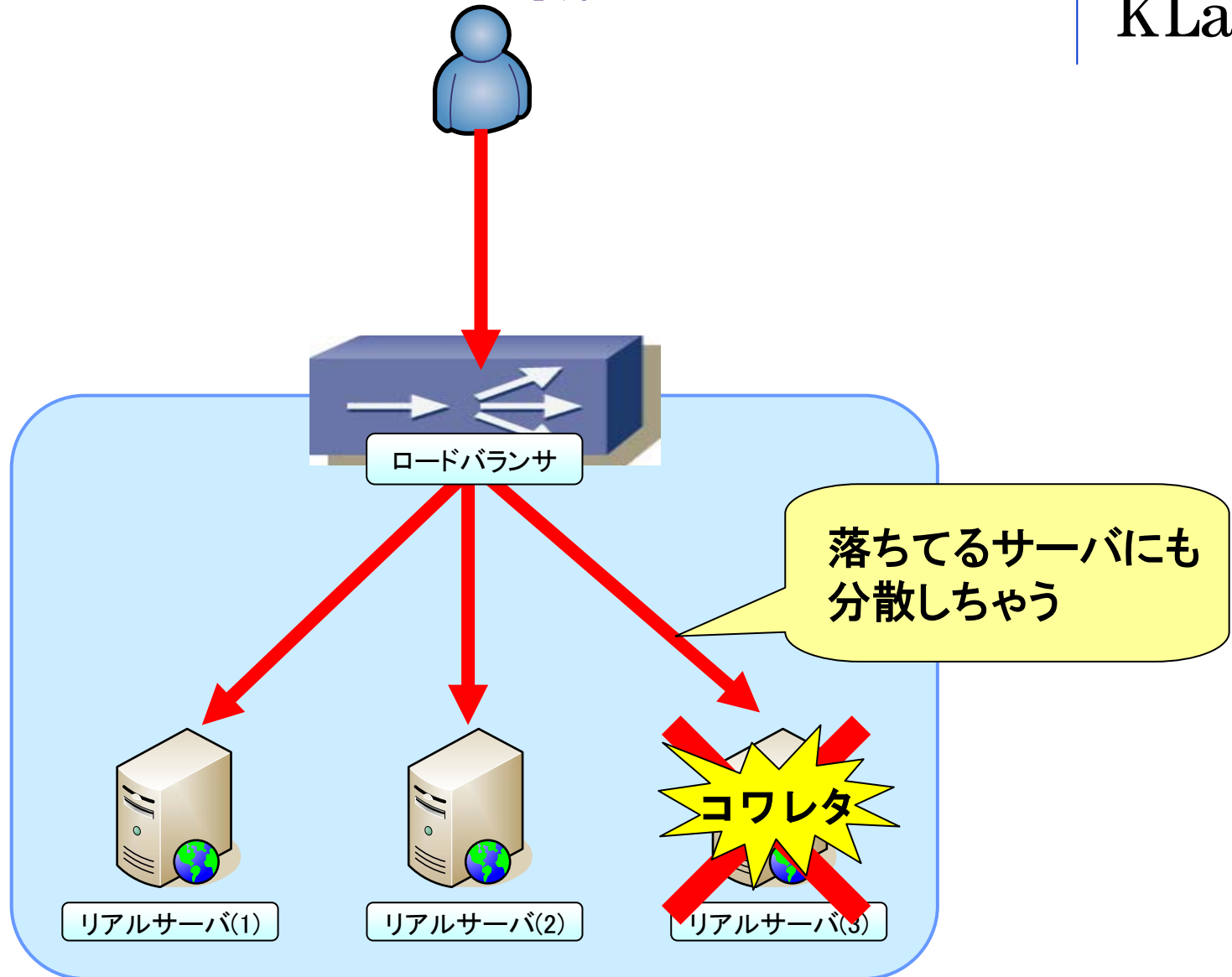


ロードバランサもあるし
これで
夜もぐっすり眠れる？

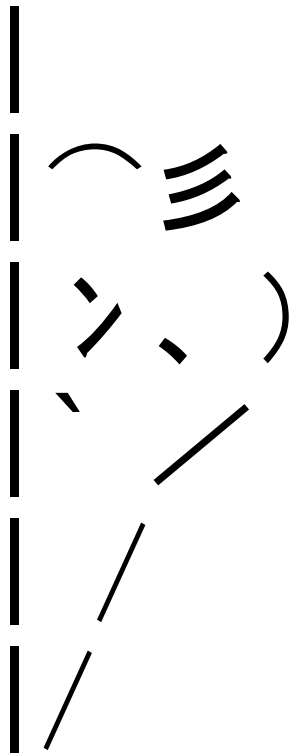
**リアルサーバが落ちたら
半停止しちゃいますねー**



IPVSにはヘルスチェック機能がない



じゃあどうするか？



それ、keepalived
でできるよ

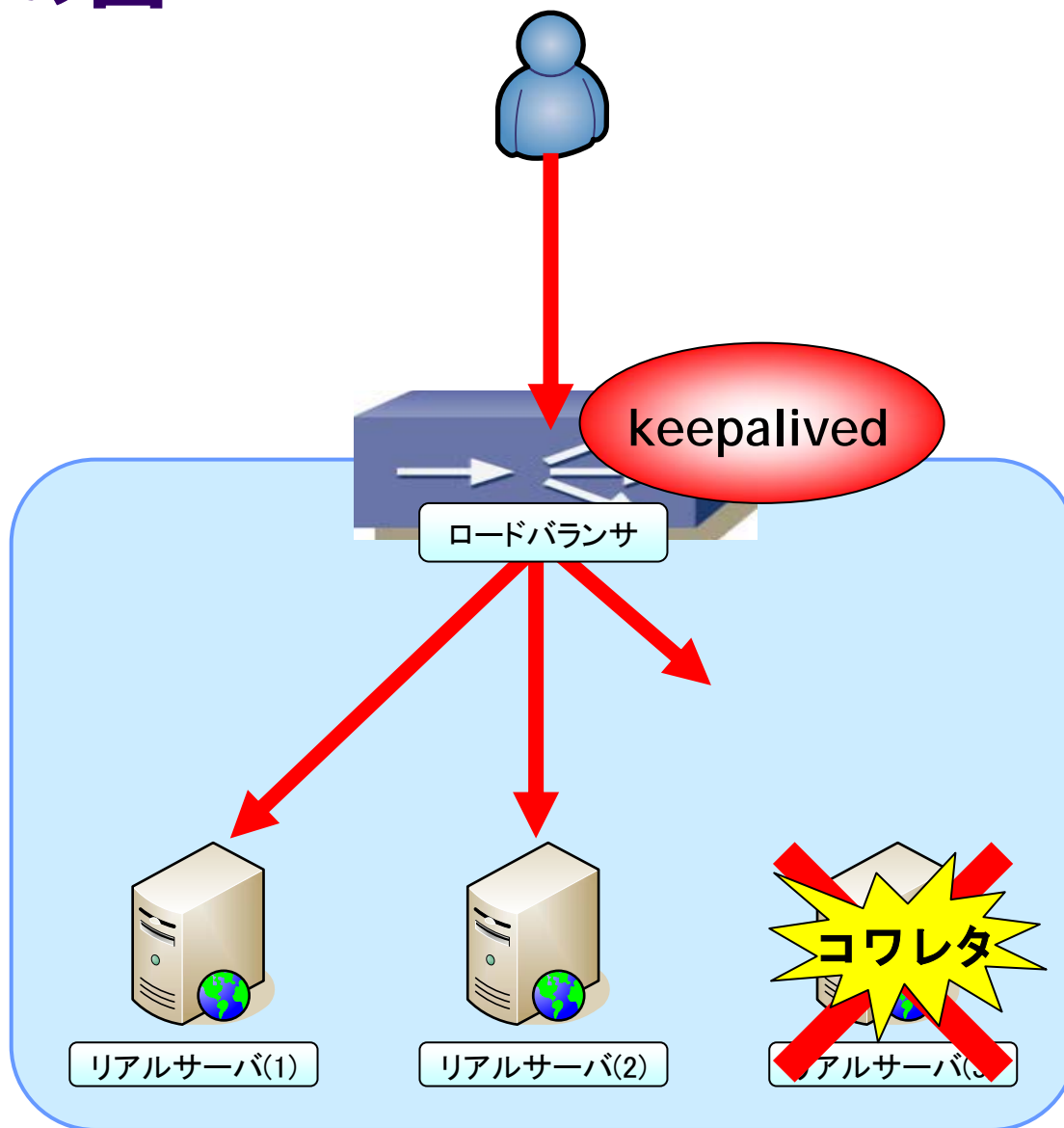
keepalivedの機能（1/2）

- 2つの機能がある
- その1つが、
 - ヘルスチェック機構＋IPVSの制御
 - 定期的にリアルサーバにヘルスチェック
 - 期待した応答が得られなければ
 - IPVS的に分散から外す
- これでリアルサーバが落ちても大丈夫

keepalivedのヘルスチェックの種類

- TCP_CHECK
- HTTP_GET
- SSL_GET
- SMTP_CHECK
- MISC_CHECK

ここまでの図

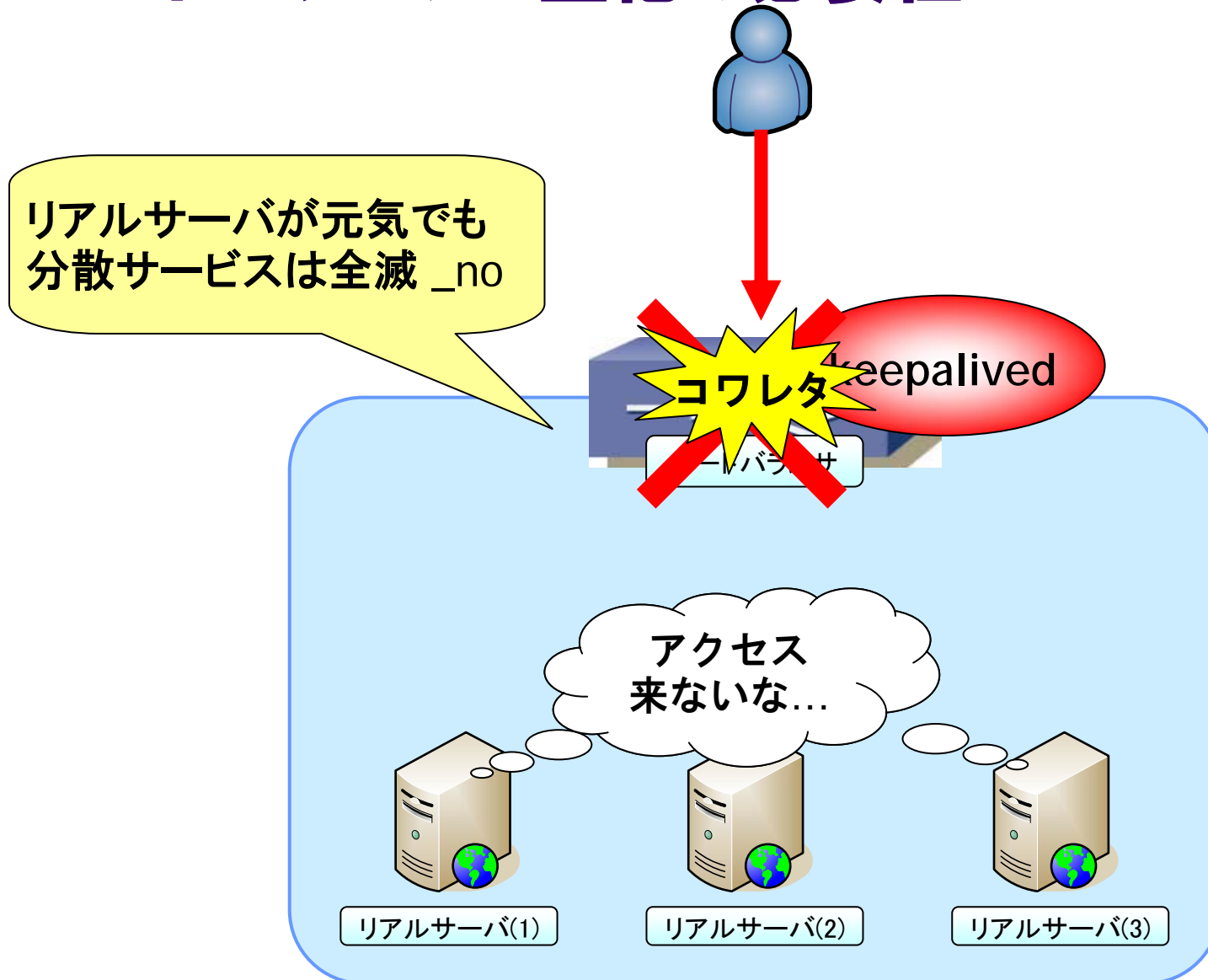


これで今度こそ
夜もぐっすり眠れる？

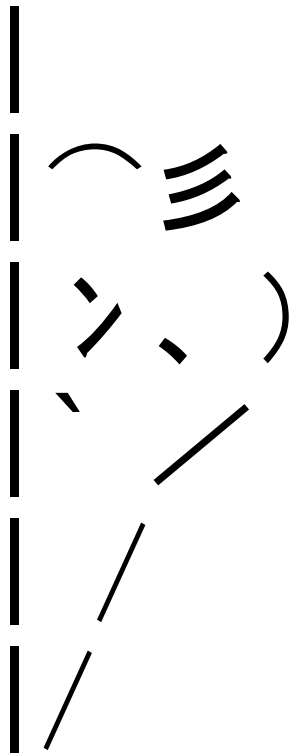
**ロードバランサが落ちたら
全停止しちゃいますねー**



ロードバランサ二重化の必要性



じゃあどうするか？

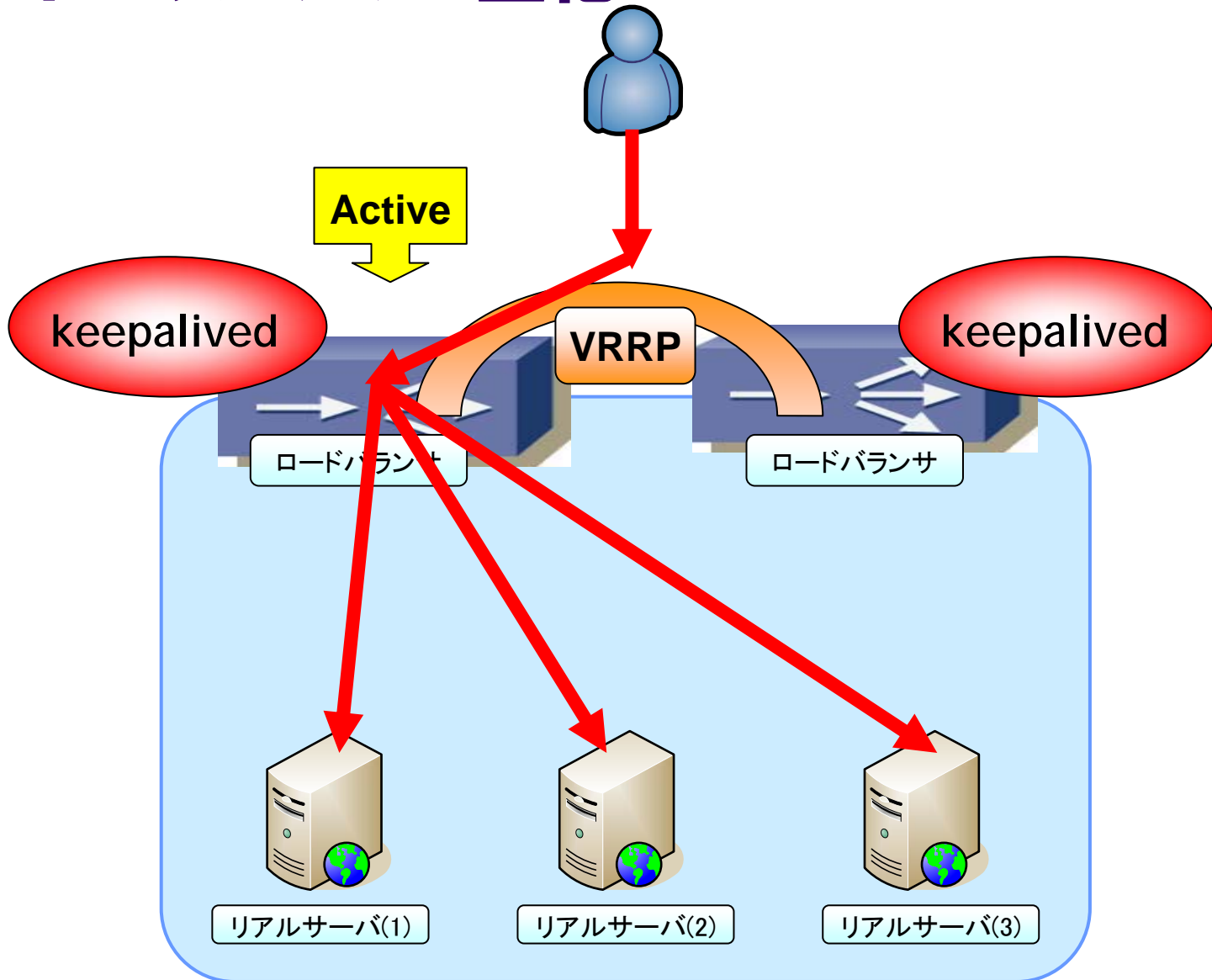


それ、keepalived
でできるよ（再）

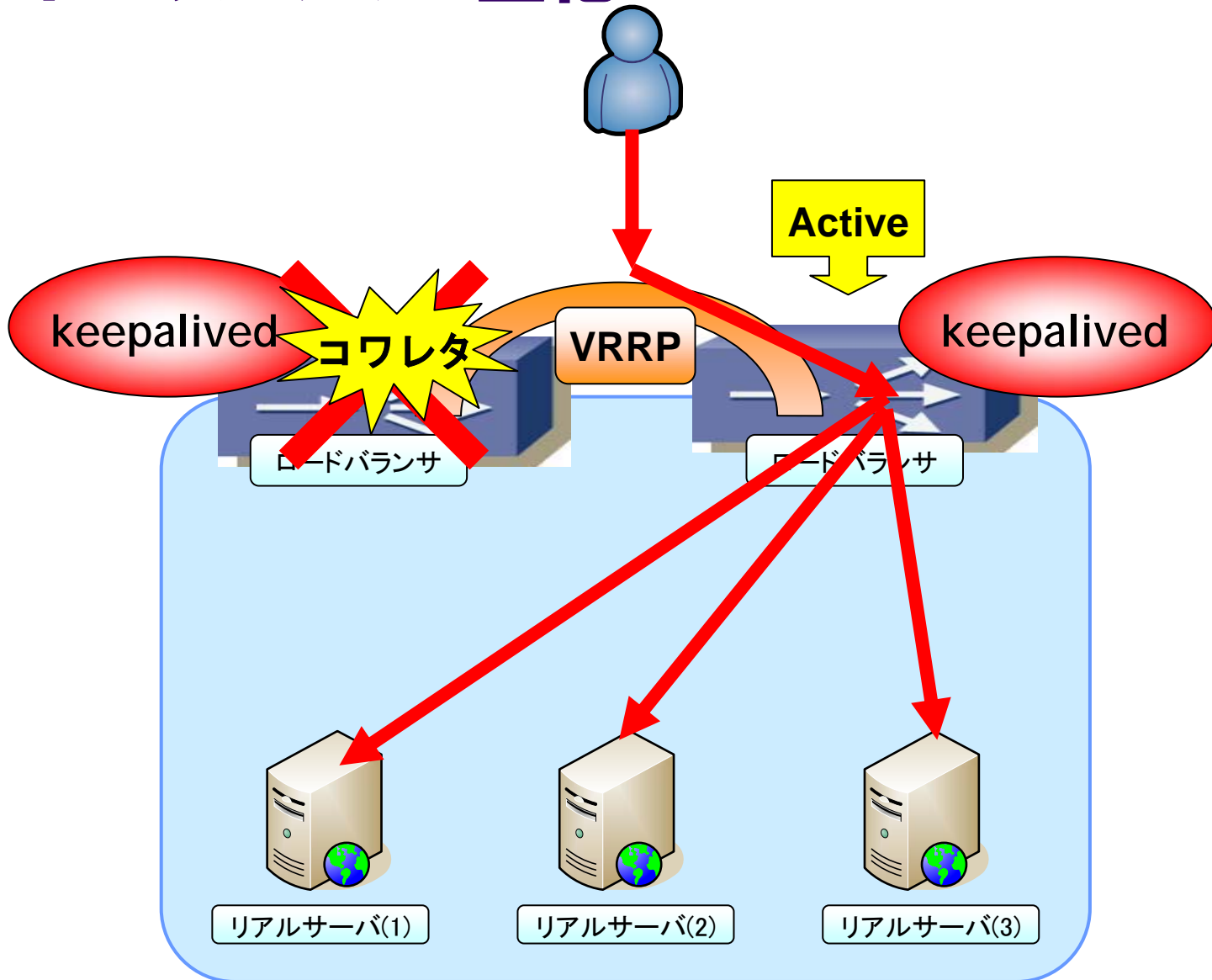
keepalivedの機能（2/2）

- もう1つの機能
- VRRP - Virtual Router Redundancy Protocol
 - ルータ(ロードバランサ)の
 - 二重化
 - フェイルオーバー
 - を実現するためのもの。
 - マルチキャストで死活確認
- これでロードバランサが落ちても大丈夫◎

ロードバランサの二重化



ロードバランサの二重化




LVS+keepalivedの 詳しい設定方法

テーマミソですが…



LVS+keepalivedの設定方法

-  Vol.37 (2月2x日発売予定)
- にこのへんの特集記事を書きましたのでそちらを！！
 - サーバ負荷分散概論
 - フルオープンソースで実現するロードバランサ
 - ロードバランサを冗長化
 - 負荷分散システム運用のコツ



小まとめ

- LVS+keepalivedでできること
 - L4負荷分散
 - リアルサーバの管理
 - リアルサーバの死活監視(ヘルスチェック)
 - ロードバランサの二重化

=マトモなL4ロードバランサがあなたの元に！

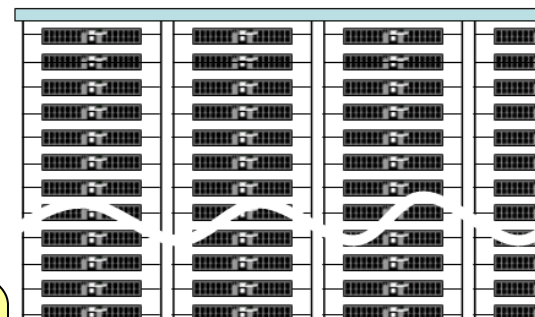
ネットワークブート

**OSのインストールって
めんどくさいですねー**



OSインストールはめんどくさい

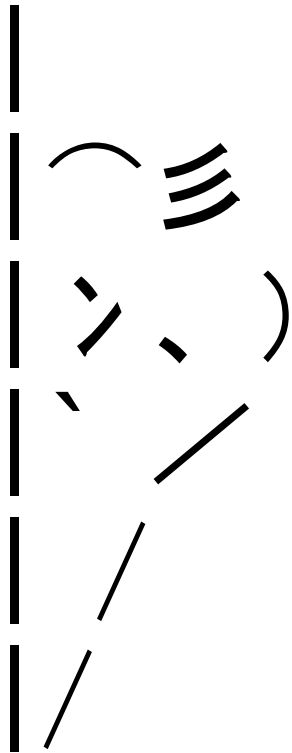
- OSインストールが発生するのは
 - 新しいマシンが来たとき
 - 壊れたマシンを復旧するとき
- まともにやろうとすると
 - 現地に行って
 - CD-ROMを入れて
 - 展開するのを待つ...



台数が
多いと...

**時間もかかるし待ち時間も長い
メンドクサイのはイヤ！！**

じゃあどうするか？



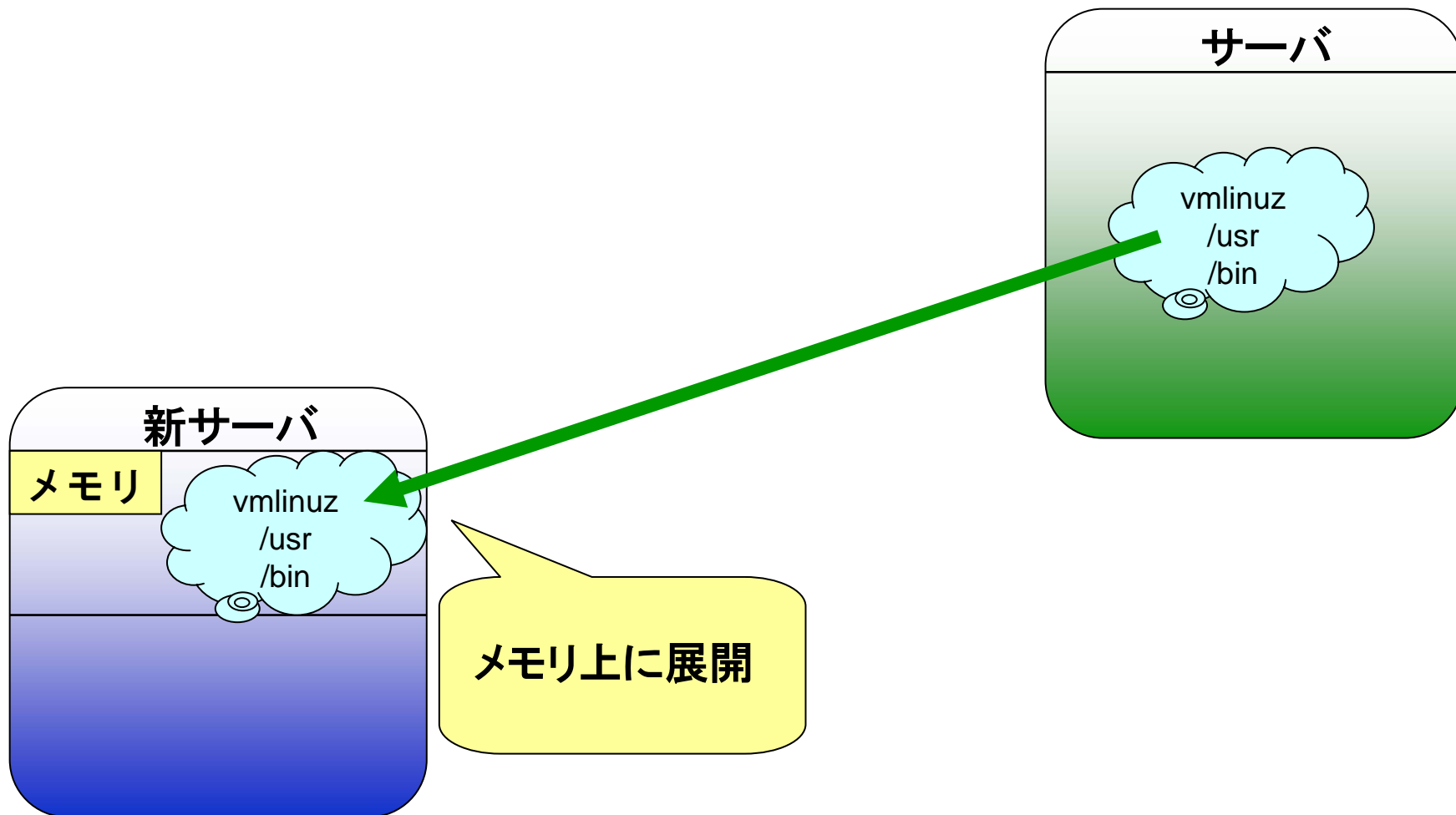
それ、ネットワークブート
でできるよ



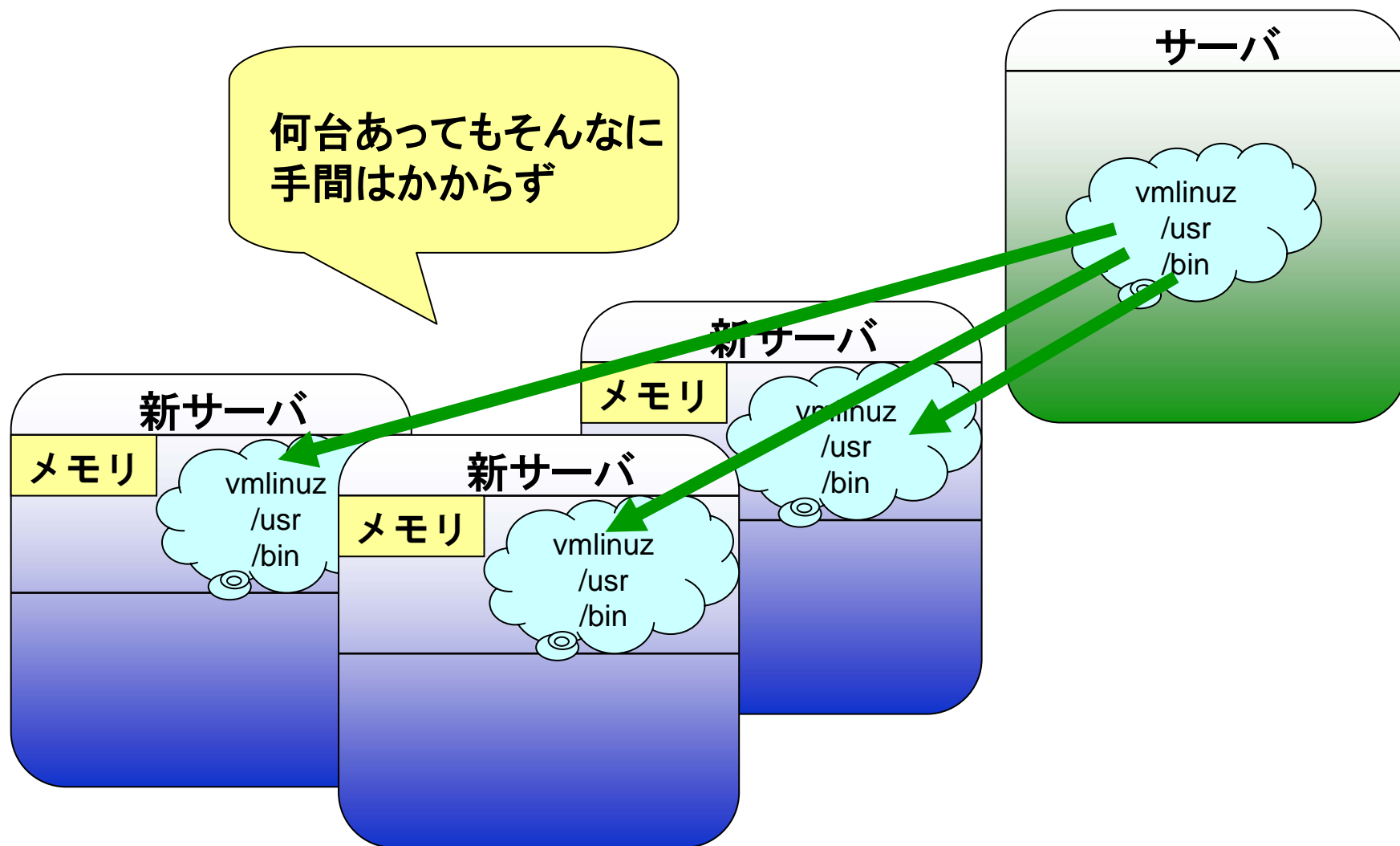
ネットワークブートとは？

- ネットワーク越しに
- 稼動に必要なもの
 - ブートローダ
 - カーネル
 - ファイルシステムなど
- を持ってきて展開して起動するしくみ

DSAS的ネットワークブートの使い方



DSAS的ネットワークブートの使い方

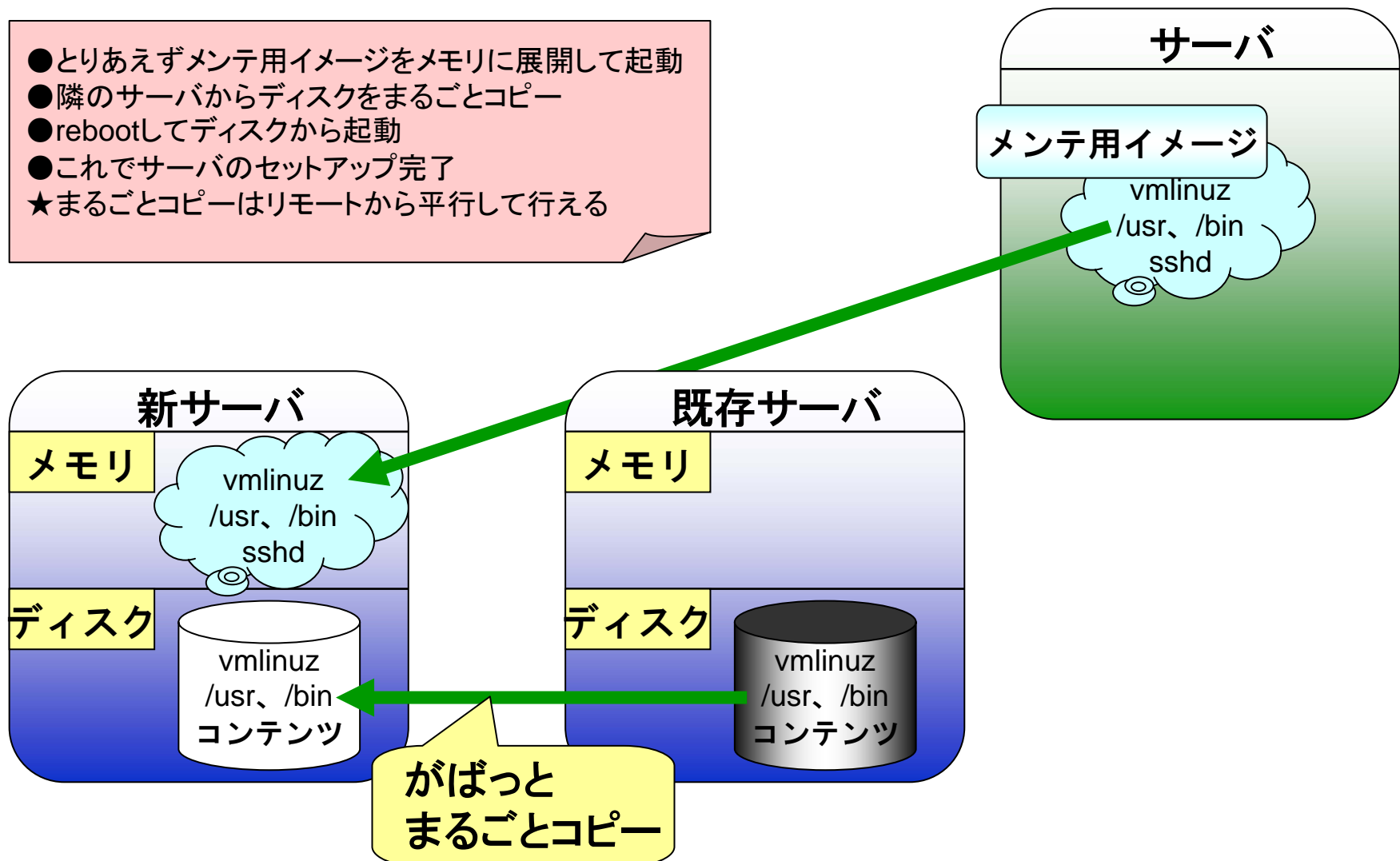


ネットブート効果

- 物理設置が終わればもう帰れる
- 大量導入時：人海戦術で一気に物理設置を済ませればそれでおしまい
- あとはリモートで

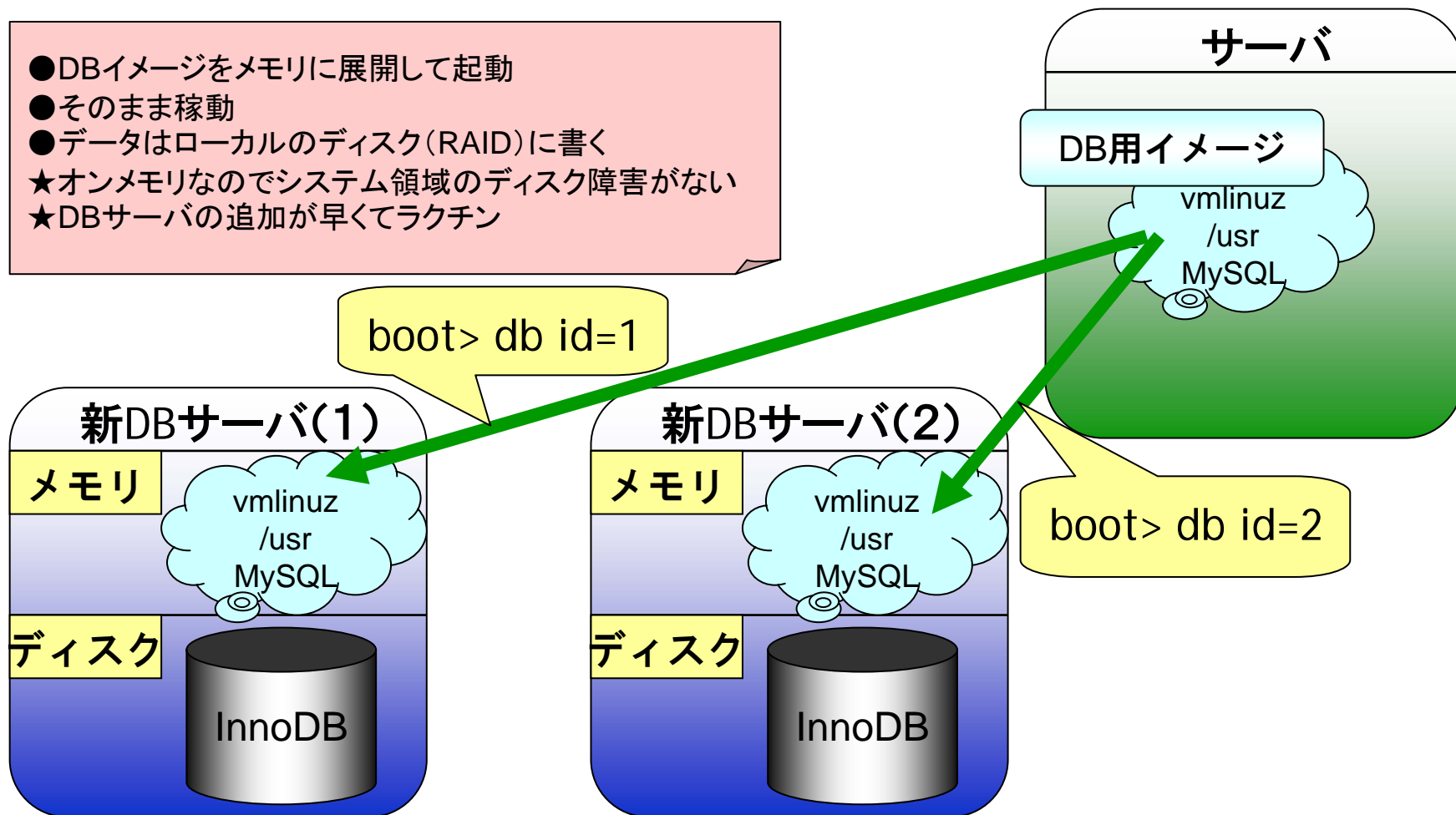
活用例 – 初期セットアップ用

- とりあえずメンテ用イメージをメモリに展開して起動
- 隣のサーバからディスクをまるごとコピー
- rebootしてディスクから起動
- これでサーバのセットアップ完了
- ★ まるごとコピーはリモートから平行して行える



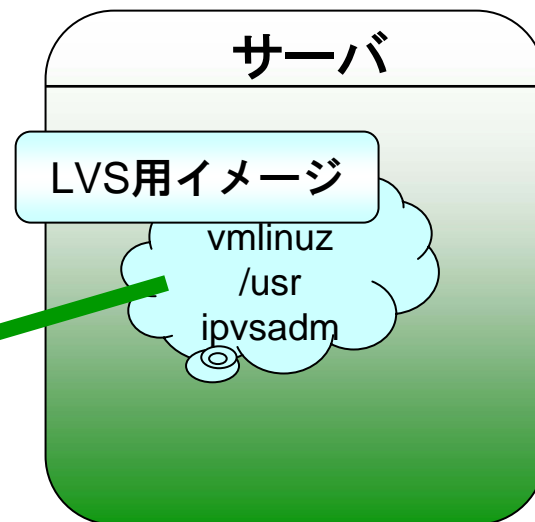
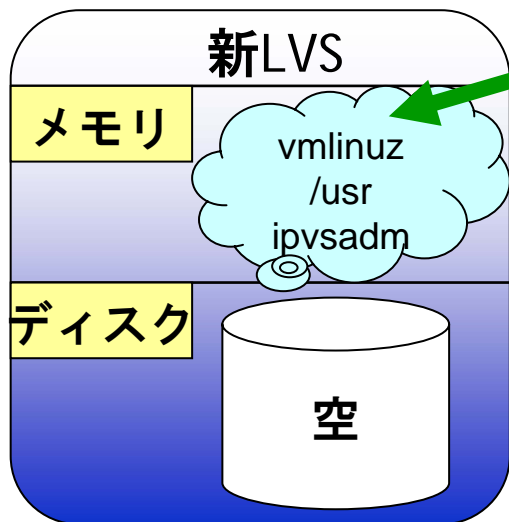
活用例 – DBサーバ

- DBイメージをメモリに展開して起動
- そのまま稼動
- データはローカルのディスク(RAID)に書く
- ★オンメモリなのでシステム領域のディスク障害がない
- ★DBサーバの追加が早くてラクチン



活用例 – LVS

- LVSイメージをメモリに展開して起動
- そのまま稼動
- ログはsyslogで飛ばす
- ★ フルオンメモリなのでディスク障害とオサラバ
- ★ もし壊れても復旧が早くてラクチン





小まとめ

- ネットワークブートの活用で、
- OSインストール要らず
 - 準備時間の短縮
 - 無駄な待ち時間からの開放
- 役割サーバの大量生産が可能
 - しかも手間がかからない

故障に強い ストレージサーバ

みなさん、
RAIDって使ってますか？

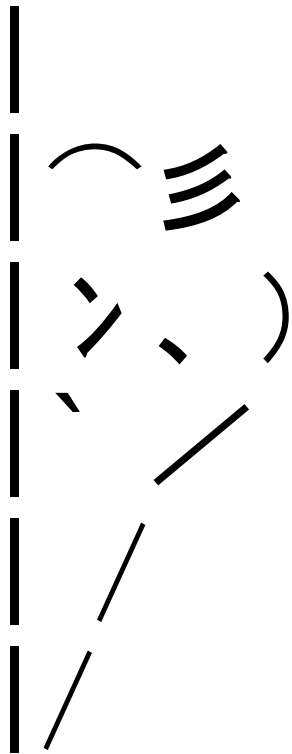
**DSASでも使ってます
でも...**

RAIDはなんのためのもの？

- ディスク故障に備えるためのもの
- ディスク故障以外のサービス停止要因
 - サーバの電源が壊れたら？
 - ネットワークが切れたら？
 - OSが落ちたら？
 - RAIDコントローラが壊れたら？

**安心して
寝られませんかー**

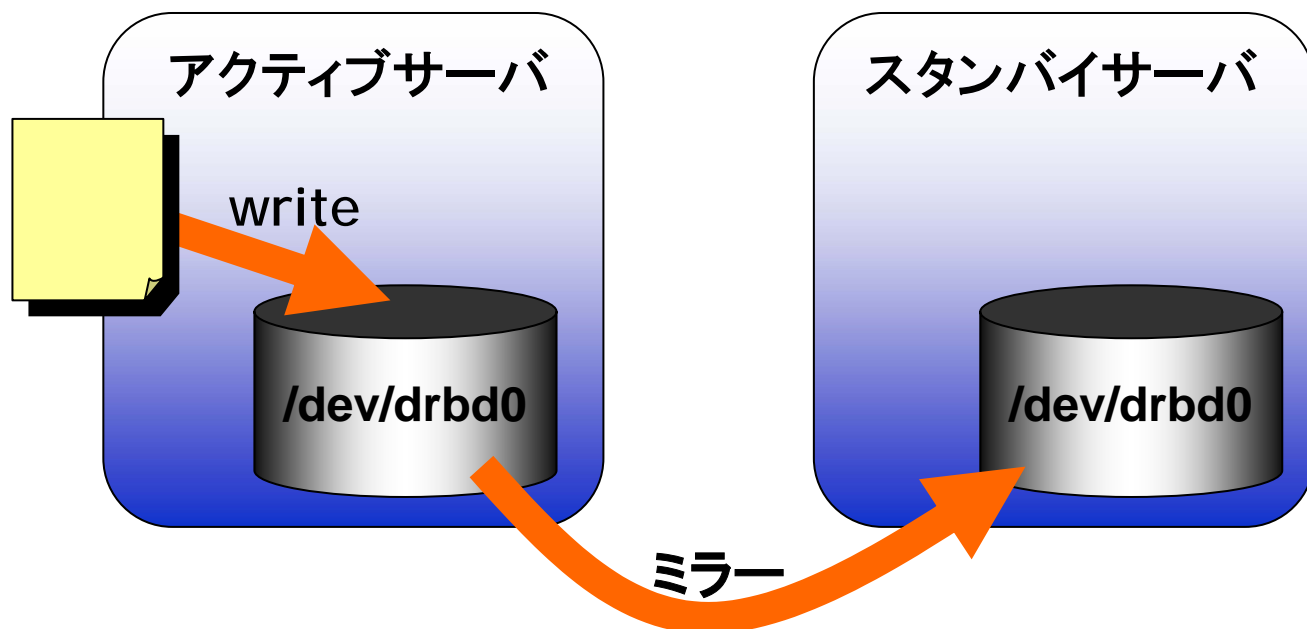




それ、DRBD
でできるよ

DRBDとは？

- DRBD - Distributed Replicated Block Device
- ネットワーク越しのサーバ to サーバのミラーリング
 - ディスクじゃなくてサーバどうしのRAID1のようなかんじもの

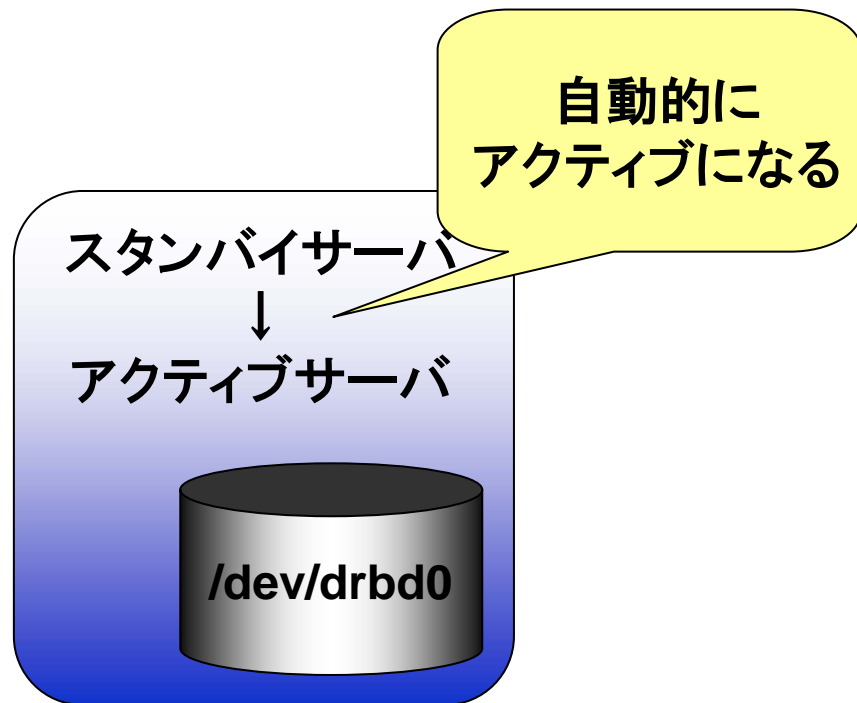


DRBDとは？

- ブロックデバイスに見える(/dev/drbd0)
 - mkfsしてmountして使う
- アクティブ/スタンバイ構成のみ
 - /dev/drbd0がマウントできるのはアクティブ側のみ
 - drbd-0.8では、GFSやOCFS2と組み合わせるとA/A構成にできる(らしい)

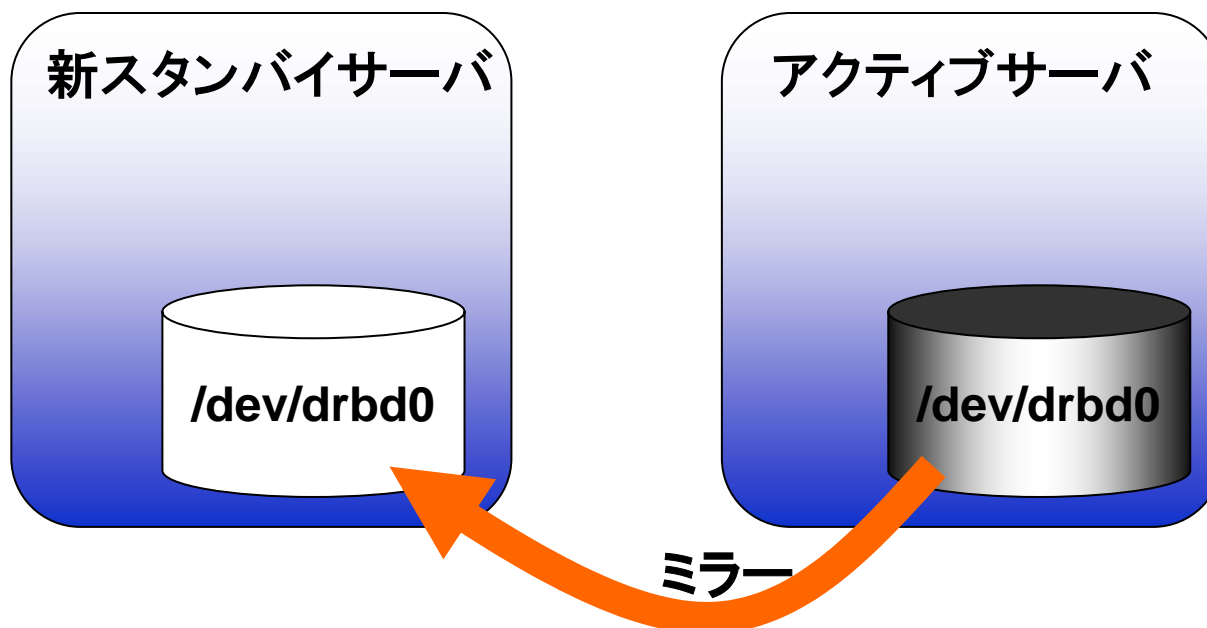
アクティブが壊れたら？

- スタンバイ機が自動的にアクティブになる
 - DRBDが勝手にやってくれる



アクティブが壊れたら？

- 新しい(空の)スタンバイ機を追加すれば、自動的にミラーをはじめて同期してくれる





DRBDの注意点

- リアルタイムのミラーリングなので
 - DRBDのオペミスなどで全部消えちゃう可能性も
 - 空のスタンバイ機を同期元としてミラーしちゃったり...
- やっぱり、コールドバックアップは必要

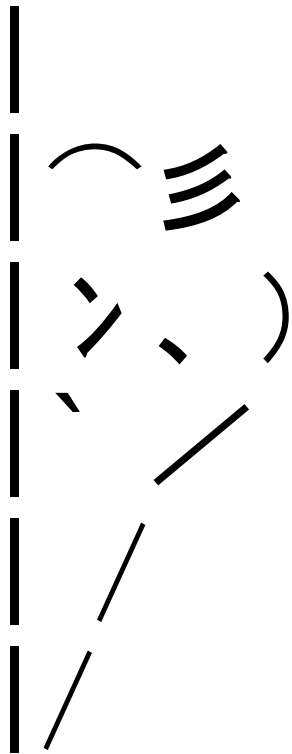
RAIDと同様

フェイルオーバーは？



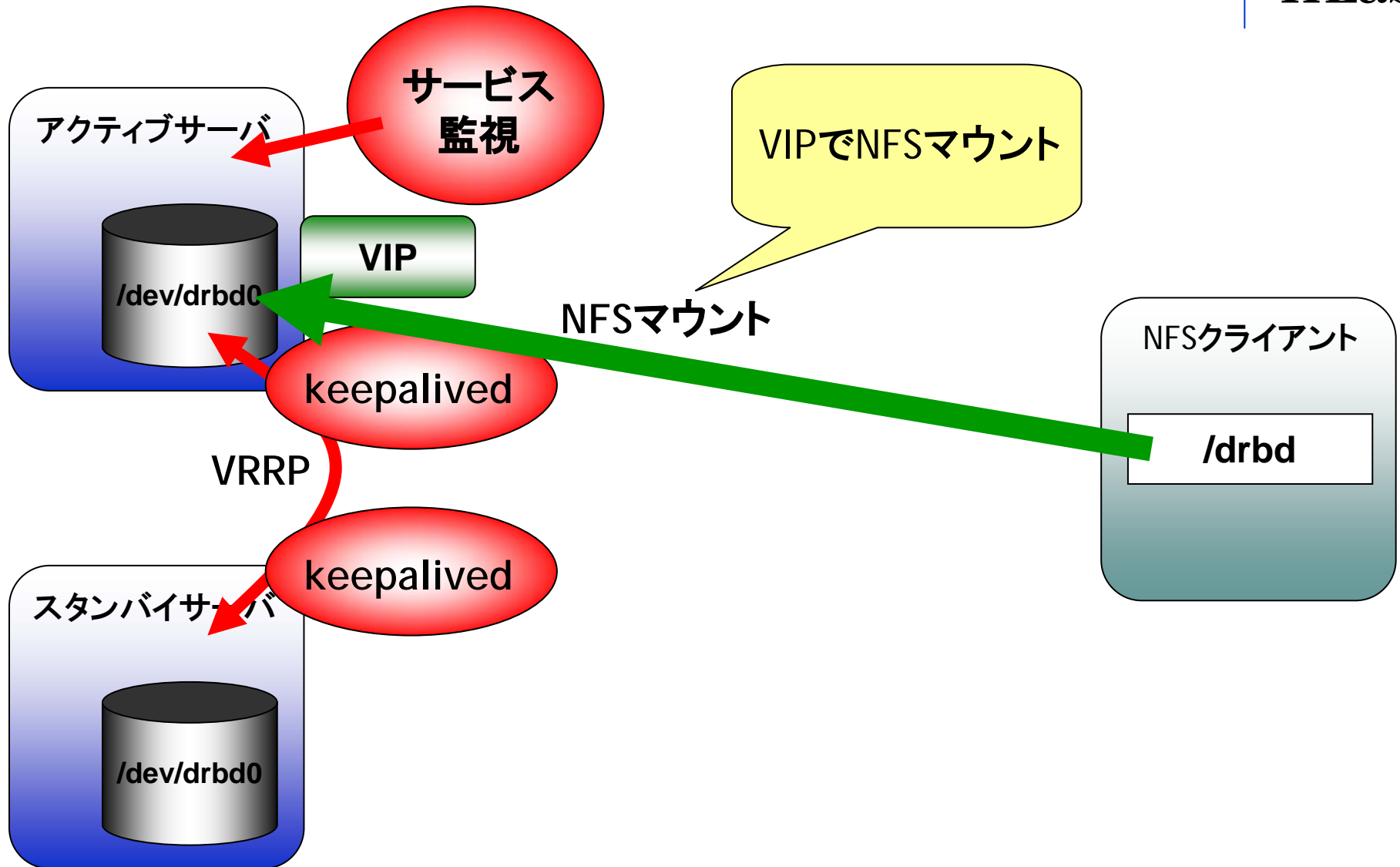
DRBDのフェイルオーバ

- ディスク障害が起こった場合
 - 実はDRBD的にアクティブ/スタンバイが切り替わるだけ
- なので、切り替わった後に
 - 新アクティブでmountが必要
 - NFSサービスなどの起動が必要
- また、
 - ネットワーク監視、サービス監視の機能はDRBDにはない

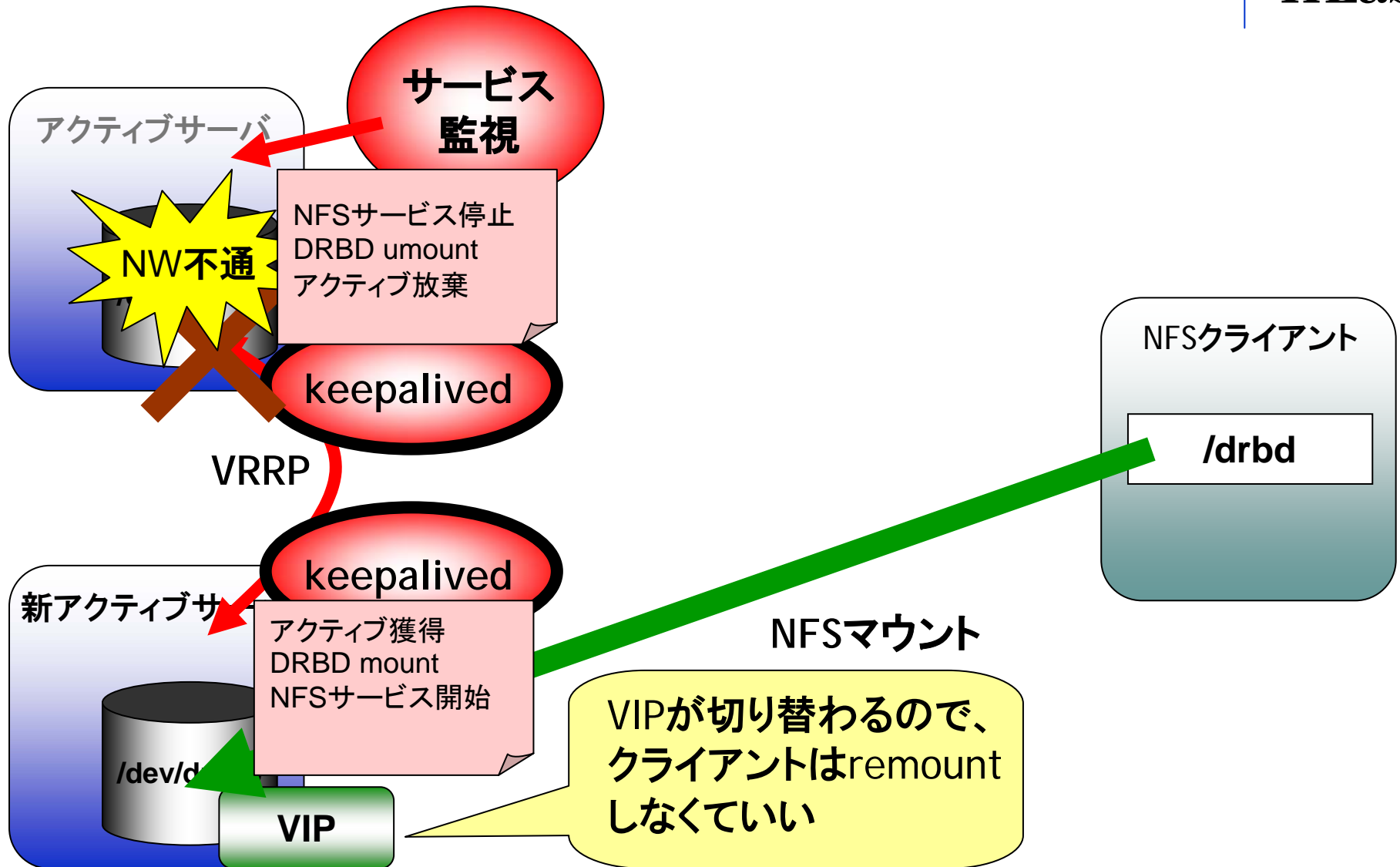


それ、keepalived
でできるよ（再々）

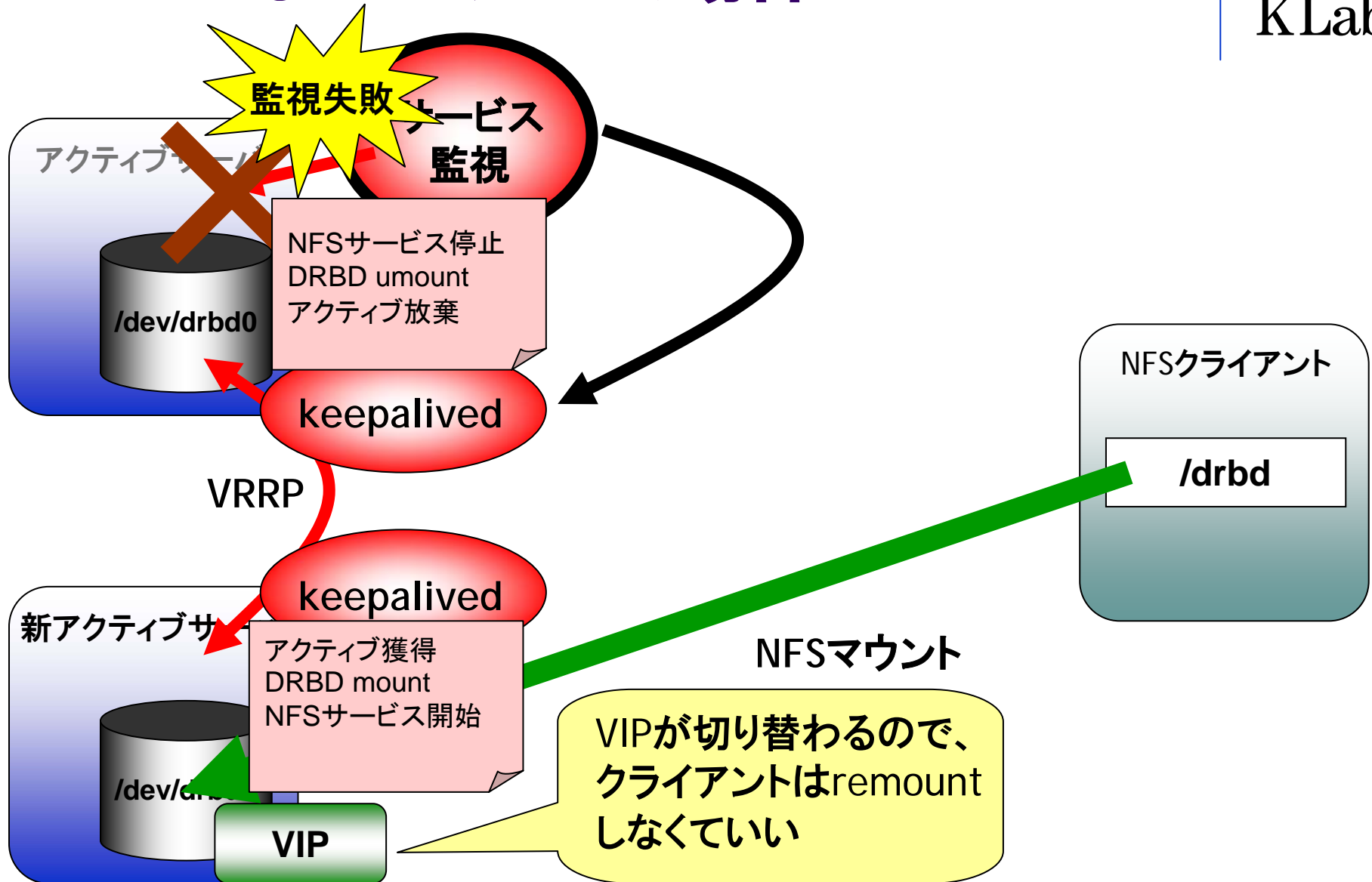
DRBDなNFSサーバの場合



DRBDなNFSサーバの場合



DRBDなNFSサーバの場合



小まとめ

- RAIDディスク＋DRBD＋keepalived
＝故障に強いストレージサーバ
- フェイルオーバーもばっちり

NICの二重化



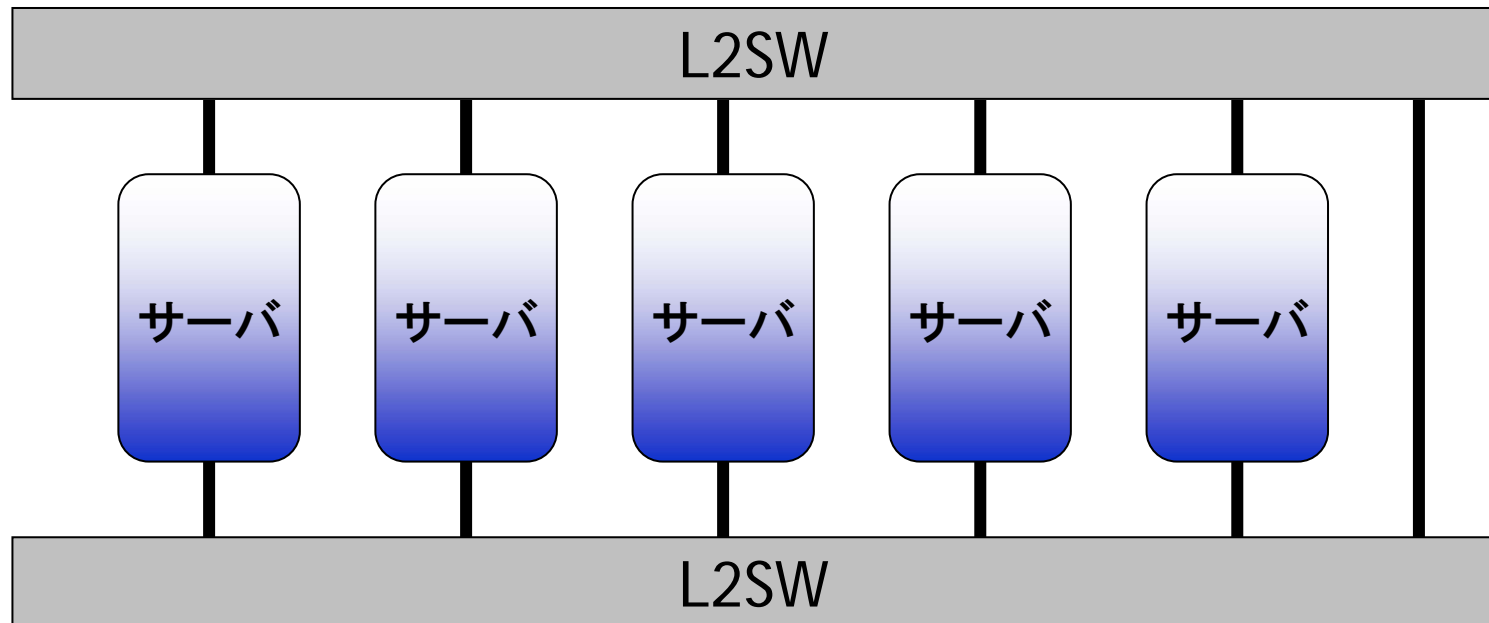
bonding

- bonding
 - 2つの物理的に異なるNICを1つの論理的NICに
 - eth0 → bond0
- 束ねるモードはいろいろある
 - active-backup
 - アクティブ/バックアップ
 - balance-tlb
 - 受信は1つのNIC、送信は2つのNICで
 - balance-alb
 - 受信も送信も2つのNICで



小まとめ

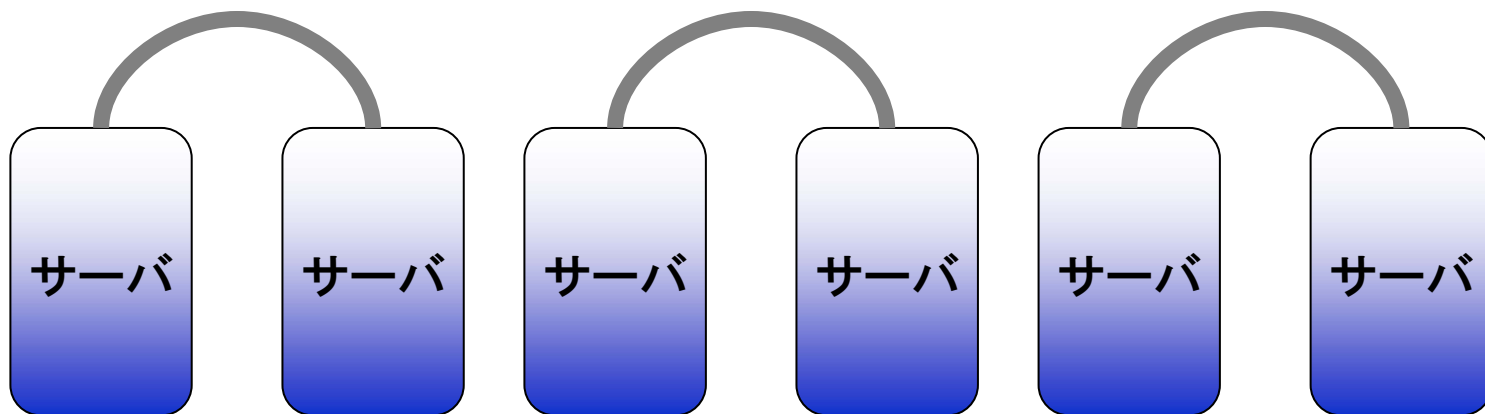
- NICの冗長化(active-backup)の利点
 - SWのポートが死んだとき
 - SWがまるごとハングったとき
 - SWのファームウェアをバージョンアップするとき
 - LANケーブルが抜けた、切れたとき



シリアル接続 温故知新

明日使える、シリアル接続

- シリアルポート、使わずに空いてませんか？
- もったいないので活用しましょう
- 2コ1でシリアル接続すれば十分
 - 超お手軽



シリアル接続でなにができるのか？

- シリアルコンソール
 - ネットワークの設定をいじるときに安心便利
 - サーバによってはBIOSもいじれる(ネットブートで必要)
- SysRqの送信
 - キーボードがなくてもシリアルで送れる
 - にっちもさっちもいかないときの非常手段
 - sync
 - read-onlyでremount
 - リセット
- データ転送
 - cu, rz, sz, zmodem

小まとめ

- 2コ1シリアル接続で、
- いざというときも安心
 - ネットワークが切れてログインもファイル転送もできない
 - ロードアベレージが高くて手がつけられない

サーバリソースの見える化

RRDToolでグラフ化

- グラフ化するもの
 - ロードアベレージ
 - CPU使用率、メモリ使用率、ディスク使用率
 - ネットワークトラフィック
- 過去のグラフも見られるので、やってみると意外と役に立つ
 - 例)先週のTVCMのときのサーバの具合を再確認したい


RRDToolのフロントエンド

- たくさんある
- たとえばCacti
 - Webブラウザでノードの追加、設定ができる
 - お手軽だけど、ノードが多いと管理が大変

DSASではgangliaを使っています

- gangliaの特徴
- クラスタ環境で使うことを想定して作られている
- 1つのコレクタ(gmetad)と、たくさんのノード(gmond)という構成
 - ノードの設定が不要
 - ノードを追加したとき、コレクタの設定変更が不要
 - →手間要らず

gangliaの画面 – クラスタ全体



Cluster Report for Fri, 02 Feb 2007 14:20:43 +0900

Get Fresh Data

Metric:

Last: - or - From: To: refresh

Sorted:

Physical View


KLab Grid > DSAS >

Overview of DSAS

CPU's Total: **156**
 Hosts up: **50**
 Hosts down: **0**

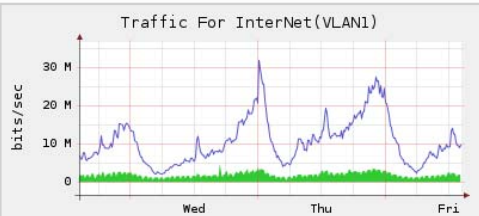
Avg Load (15, 5, 1m):
5%, 6%, 10%
 Localtime:
2007-02-02 14:20

Cluster Load Percentages



- 100+ (4.00%)
- 75-100 (2.00%)
- 50-75 (4.00%)
- 25-50 (2.00%)
- 0-25 (88.00%)

Traffic For InterNet(VLAN1)

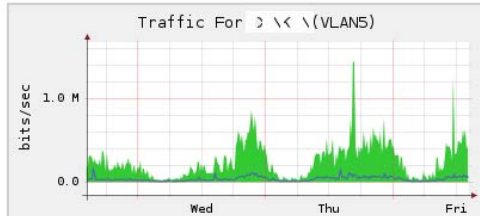


bits/sec

Wed Thu Fri

Recv min 228.52 k avg 1751.71 k max 4606.90 k
 Send min 2140.59 k avg 10666.96 k max 31983.88 k

Traffic For > \< \ (VLAN5)

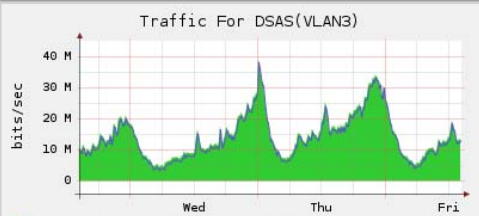


bits/sec

Wed Thu Fri

Recv min 13.30 k avg 241.41 k max 1430.99 k
 Send min 8.67 k avg 40.15 k max 166.27 k

Traffic For DSAS(VLAN3)

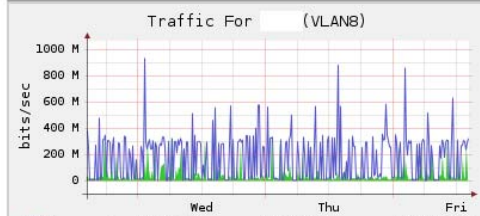


bits/sec

Wed Thu Fri

Recv min 3.99 M avg 14.60 M max 39.08 M
 Send min 3.26 M avg 13.84 M max 38.27 M

Traffic For (VLAN8)

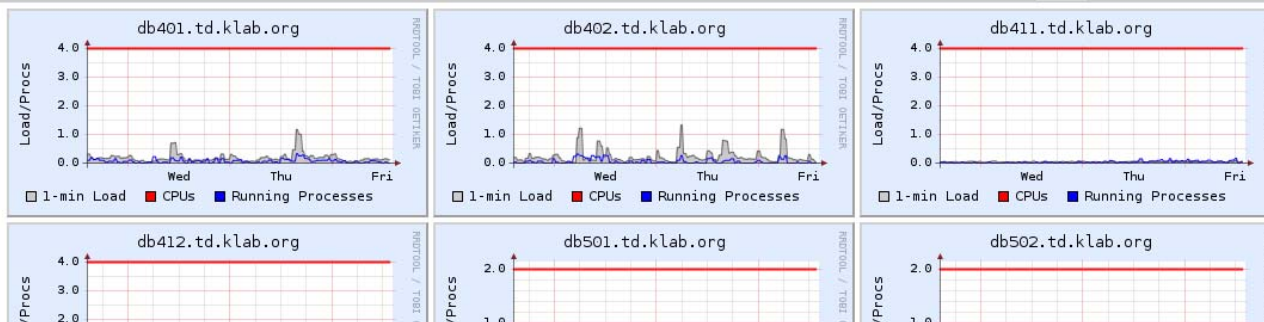


bits/sec

Wed Thu Fri

Recv min 239.96 k avg 43793.80 k max 345719.16 k
 Send min 2429.94 k avg 177105.23 k max 934731.40 k

Show Hosts: ☒ yes ☐ no | DSAS load_report last 3days sorted by hostname | Columns



gangliaの画面 – ホスト個別



Host Report for Fri, 02 Feb 2007 14:22:37 +0900

Get Fresh Data

Last - or - From - To

[Node View](#)

[KLab Grid](#) > [DSAS](#) > [w305.td.klab.org](#)

w305.td.klab.org Overview



This host is up and running.

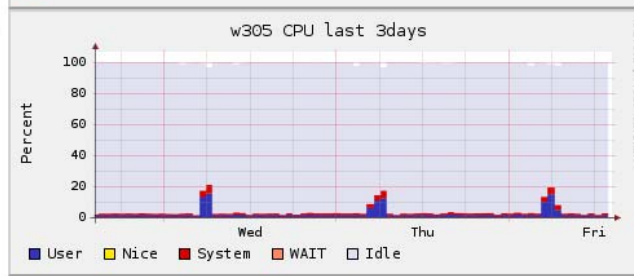
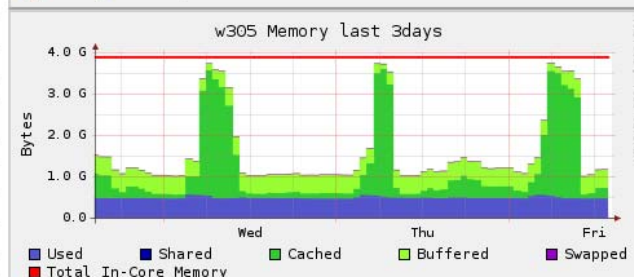
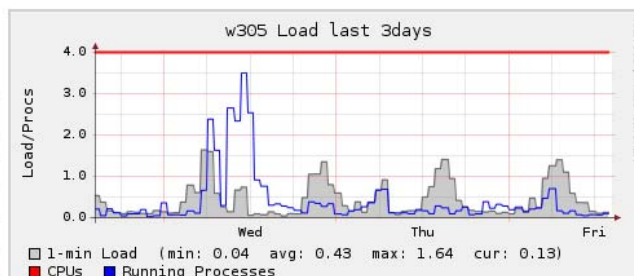
Time and String Metrics

boottime	Tue, 25 Jul 2006 21:34:12 +0900
gexec	OFF
gmond_started	Fri, 10 Nov 2006 17:56:03 +0900
last_reported	0 days, 0:00:08
machine_type	x86
os_name	Linux
os_release	2.6.16.27-01
uptime	191 days, 16:48:12

Constant Metrics

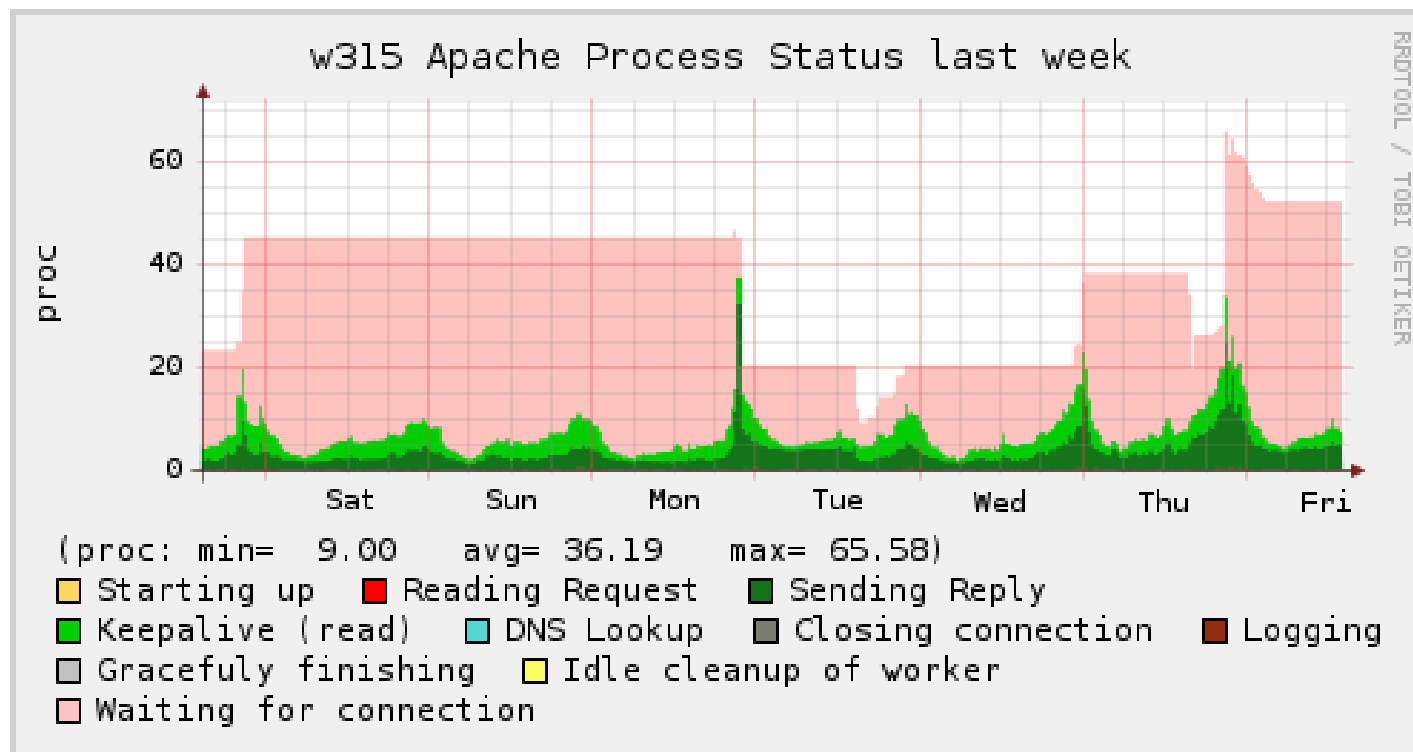
cpu_num	4 CPUs
cpu_speed	3200 MHz
mem_total	4084084 KB
swap_total	2353512 KB

Gmetrics



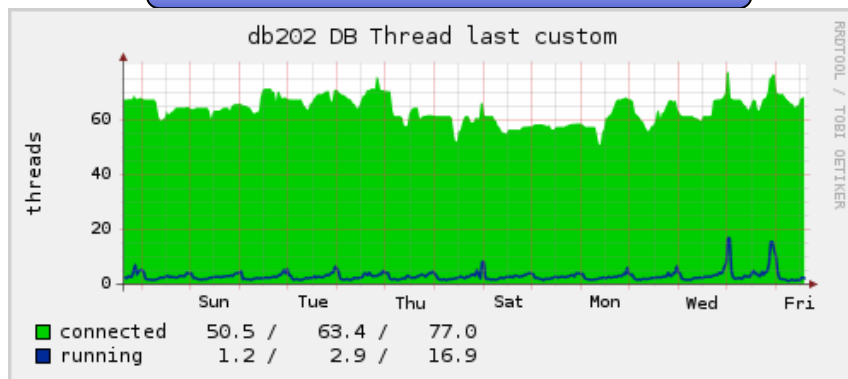
カスタムグラフ – Apacheのプロセス状態

- <http://localhost/server-status?auto>の情報を元に

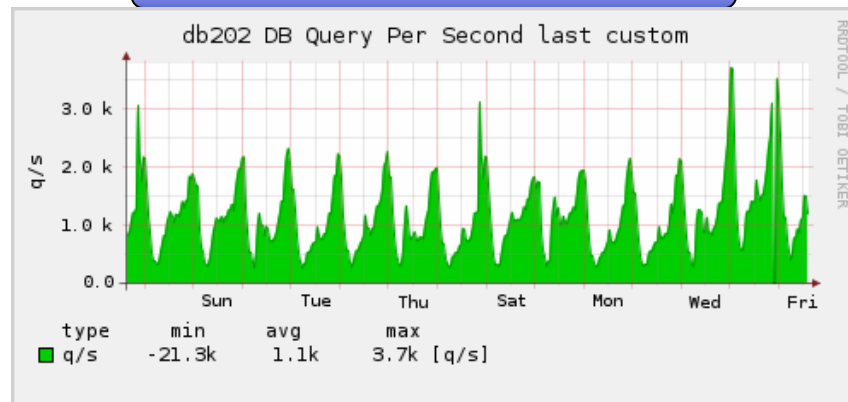


カスタムグラフ - MySQL

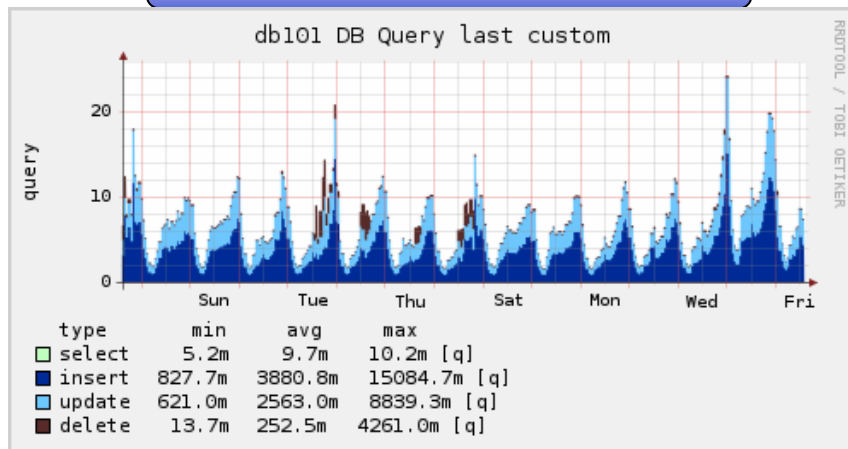
接続数



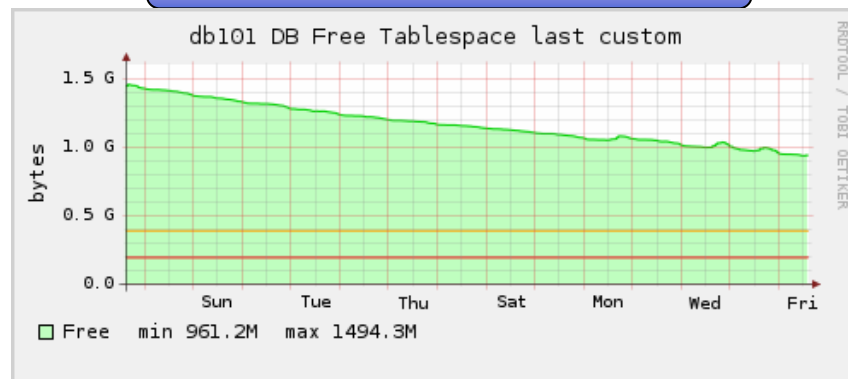
秒あたりのクエリ数



クエリ種別の割合



InnoDBの空き容量



DSASの中身のお話は ここまで

最近興味のあること



最近興味のあること

- 分散フィルシステム
 - lustre (らすたー?) <<http://www.lustre.org/>>
- F/Oできる小データ(セッションとか)共有の仕組み
 - 同期レプできるmemcachedのようなものとか
 - Terracotta <<http://www.terracotta.org/>> とか
- とかとか...
 - おもしろいネタがありましたらこの後の懇親会で！
><

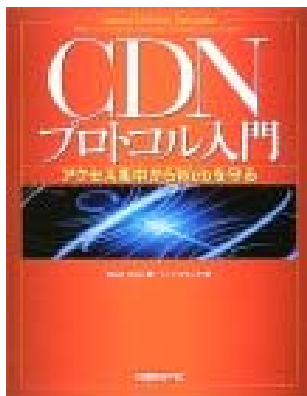
参考文献



- 『サーバ負荷分散技術』
 - トニーブルーク, Tony Bourke, 鍋島公章, 横山晴庸, 上谷一
 - オライリー・ジャパン
 - ISBN: 978-4873110653

サーバ負荷分散の概要から典型的な構成や内部の仕組みまで。

※LVSについての記述はなし



- 『CDNプロトコル入門—アクセス集中からWebを守る』
 - スコットハル, Scot Hull, トップスタジオ
 - 日経BP社
 - ISBN: 978-4822281632

サーバ負荷分散にとどまらず、その周辺の技術をイーサネットからVRRPまで幅広く扱っている。

※LVSについての記述はなし

参考文献

テーマミソですが…





● Vol.37（2月2x日発売）

- サーバ負荷分散概論
- フルオープンソースで実現するロードバランサ
- ロードバランサを冗長化
- 負荷分散システム運用のコツ

※LVSについての記述が満載！！

参考文献

さらにテマエミソですが…





- Vol.38（4月2x日発売）
 - 1年間、DSASネタで連載します！！

※LVSとかDRBDとかの記述が満載！！（予定）

今日はここまで

ご清聴、
ありがとうございました～
><

