

Virtualisation et HA

Virtualisation et Haute Disponibilité par Alixen

Introduction

- **Architecture SAN : investissement important**
- **Cluster HA Alixen**
 - Accessible
 - Évolution graduelle
- **Cibles :**
 - Plate-formes hébergées (SAAS, ASP,...)
 - PME
 - Systèmes critiques
- **Open source**

Plan

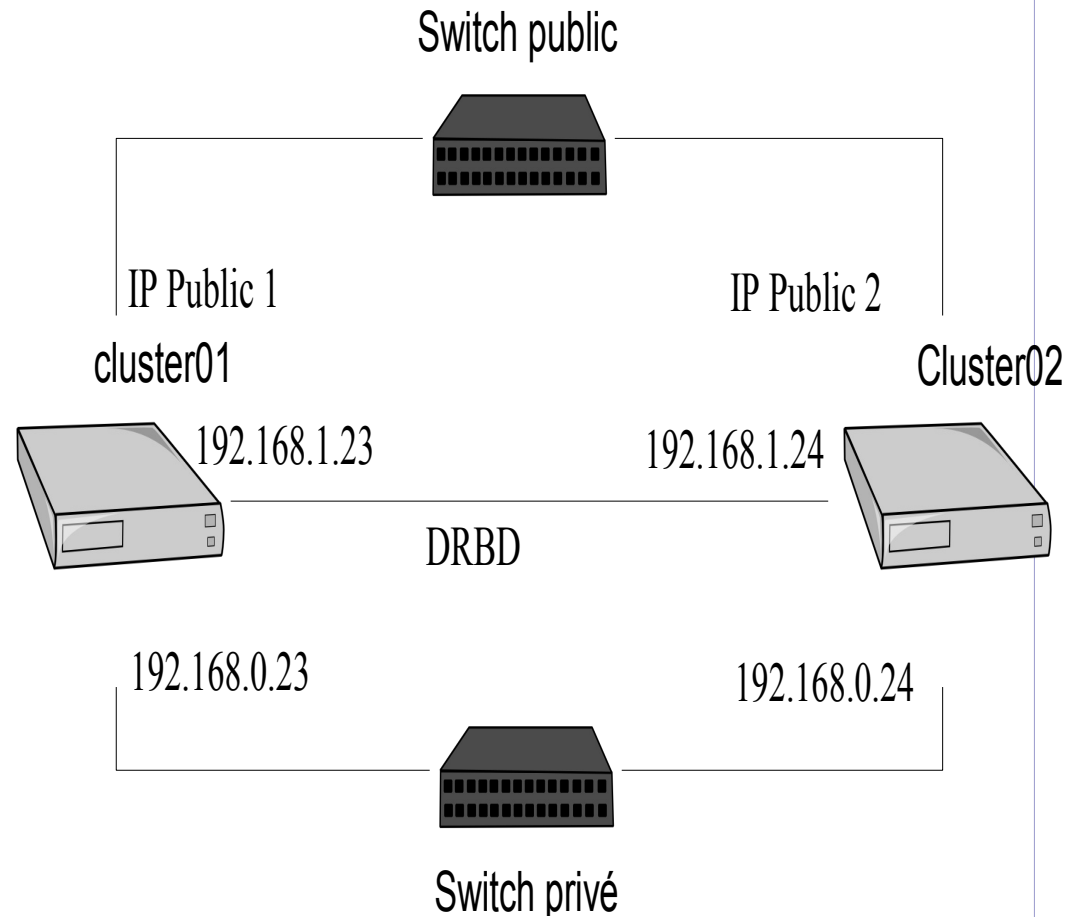
- ❖ **Objectifs**
- ❖ **Réponses apportées**
 - Architecture matérielle et logicielle
 - Administration du cluster
 - Bilan
- ❖ **Pérennité des composants logiciels**
- ❖ **Bonnes pratiques**
- ❖ **Exemples de déploiement**

Les Objectifs

- **Cluster haute disponibilité**
 - Redondance des services et données
 - Isolation des services
- **Performance**
- **Réduction des coûts**
 - Matériel
 - Administration des serveurs

Architecture cible

- 2 serveurs standards
- Réplication des données temps réel
- Haute disponibilité
- Virtualisation



Virtualisation : les grandes familles (1)

• Émulation

- Totalité du matériel simulé (CPU compris)
- Très lent

• Virtualisation

- Périphériques simulés, CPU natif
- Possibilité d'utiliser les extensions CPU

Virtualisation : les grandes familles (2)

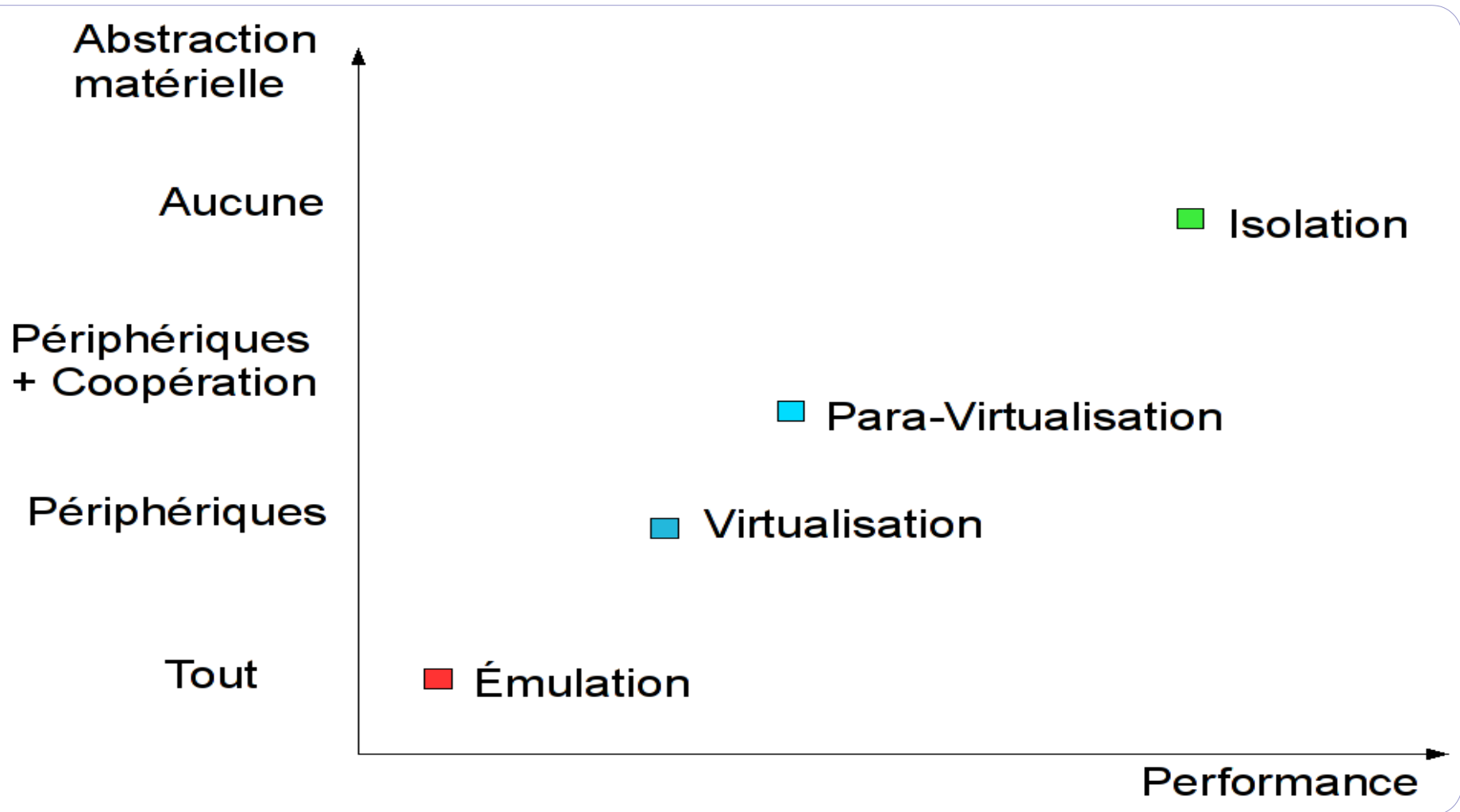
• Paravirtualisation

- Principes identiques à la virtualisation
- Coopération entre l'invité et l'hôte

• Isolation

- Un seul noyau pour tous les systèmes
- Degrés d'isolation variables
- Performance quasi native

Les technologies de virtualisation : Résumé



Plan : Virtualisation et haute disponibilité

Objectifs

Réponses apportées

- Architecture matérielle et logicielle
- Administration du cluster
- Bilan

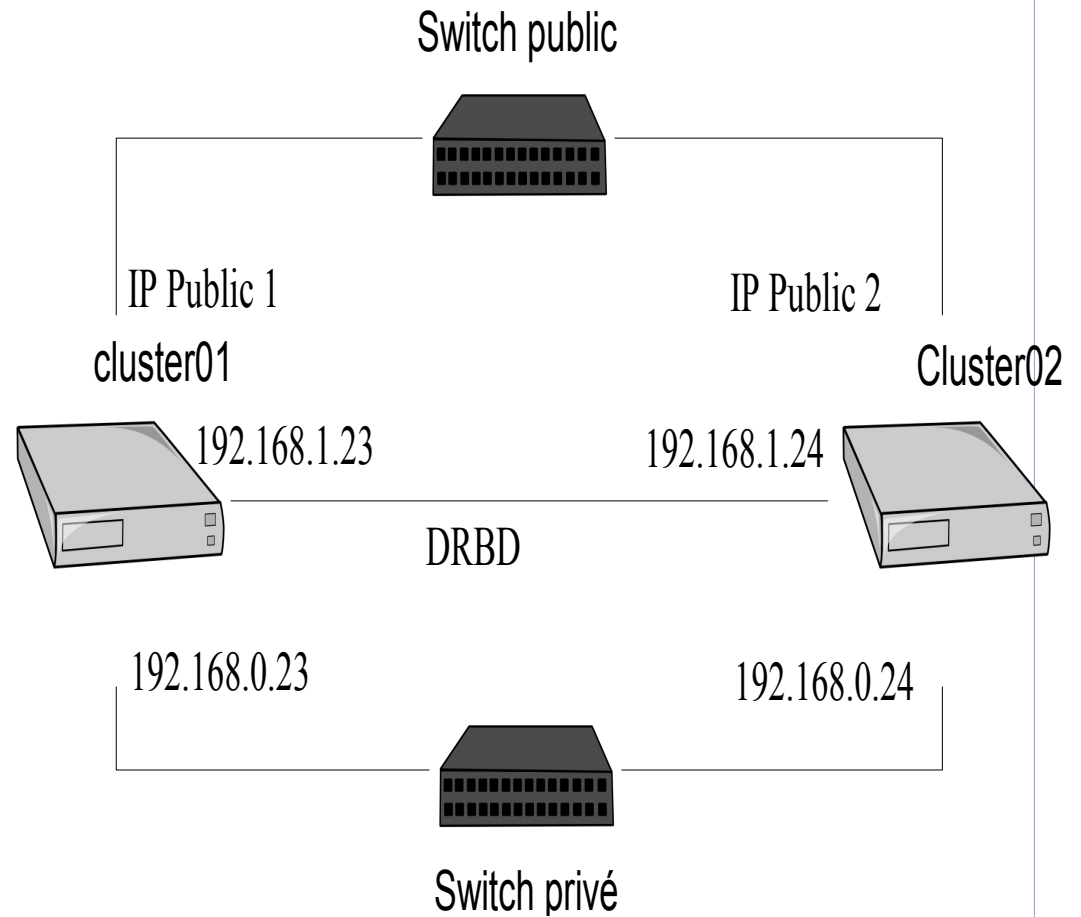
Pérennité des composants logiciels

Bonnes pratiques

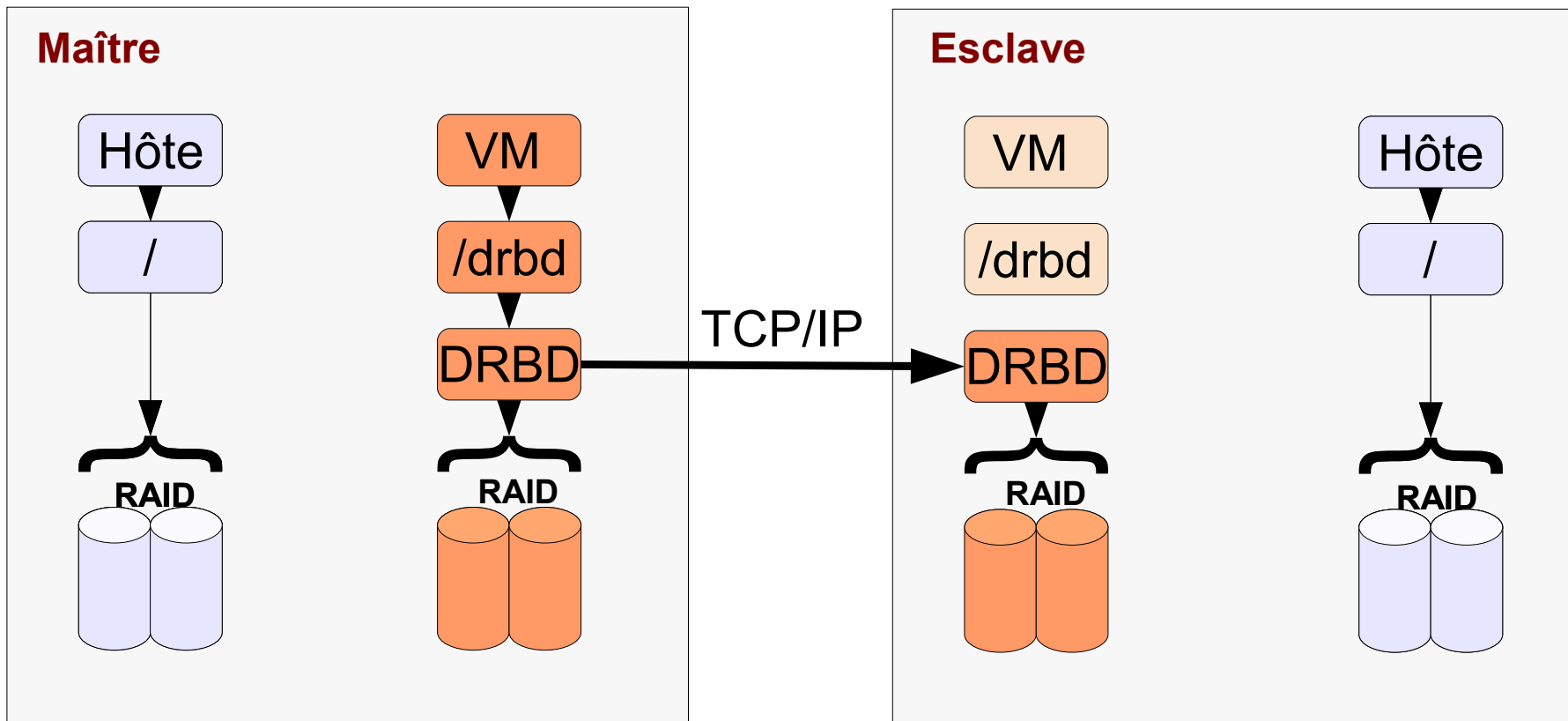
Exemples de déploiement

Architecture cible

- 2 serveurs standards
- Réplication des données
temps réel : DRBD
- Haute disponibilité :
Heartbeat
- Virtualisation : OpenVZ



DRBD : RAID 1 via le réseau



HeartBeat

- **Chef d'orchestre**
 - Gestion des ressources, services, et basculement
- **Au départ : basculement de services**
 - Complexe – configuration des systèmes hôtes
- **Apport de la virtualisation/isolation**
 - Basculement de systèmes entiers

Virtualisation - Isolation

- **Un service = une machine virtuelle (VM)**
- **Limitation des ressources par VM**
- **Administration : gestion indépendante des services**

OpenVZ : Pourquoi ?

☛ Performance

- Technologie d'isolation => overhead faible
- IO disque : identique au système hôte

☛ Gestion des ressources

- Allocation mémoire et ressources CPU

☛ Environnement homogène (Plate-forme Linux)

Variantes : choix de la technologie

	Performance (IO disque)	Environnement hétérogène
OpenVZ	Native	Non
Xen	Dégradation	Oui

Sauvegarde

• HotBackup

- Dump mémoire (registre, sockets ouvertes, ...)
- Snapshots LVM
- Indépendant de l'applicatif

• Downtime minimisé

- OpenVZ : Suspension de 1 à 3 secondes

• PRA simplifié

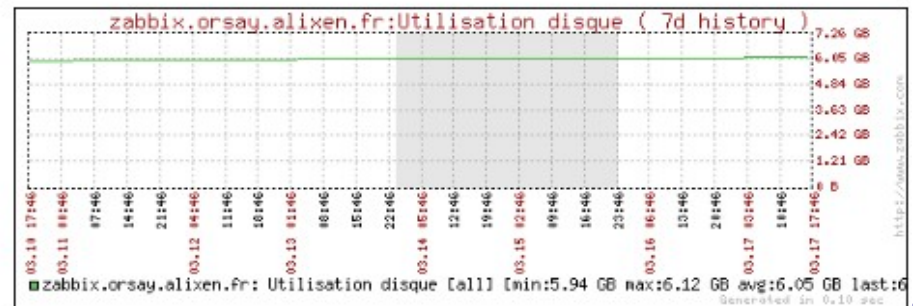
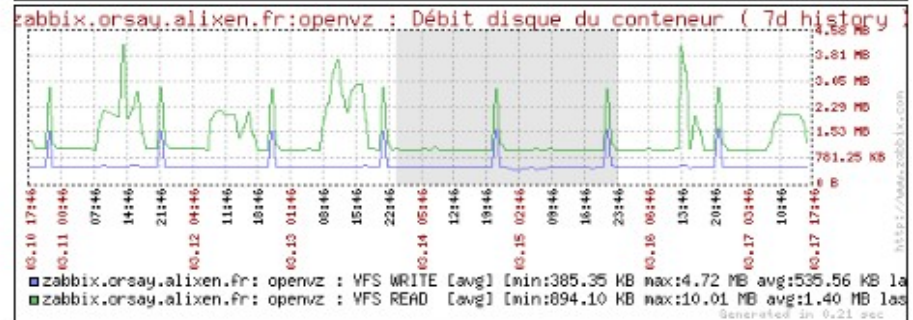
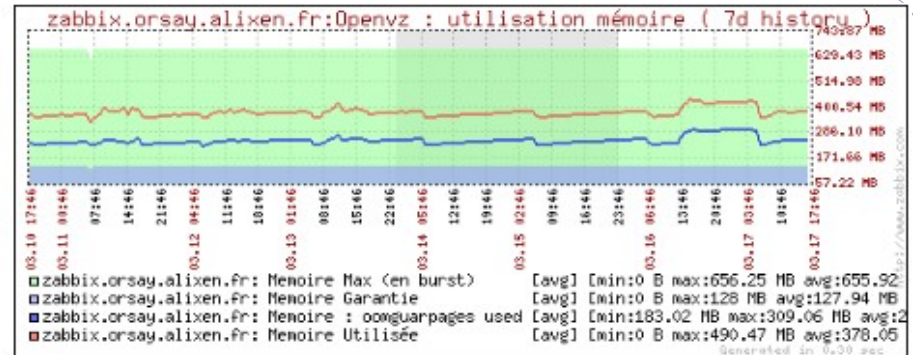
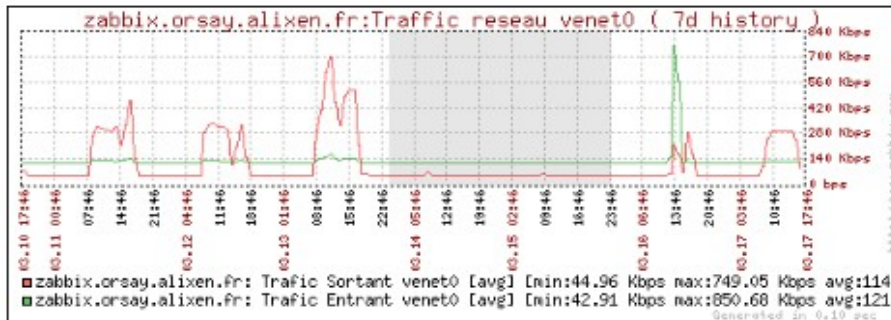
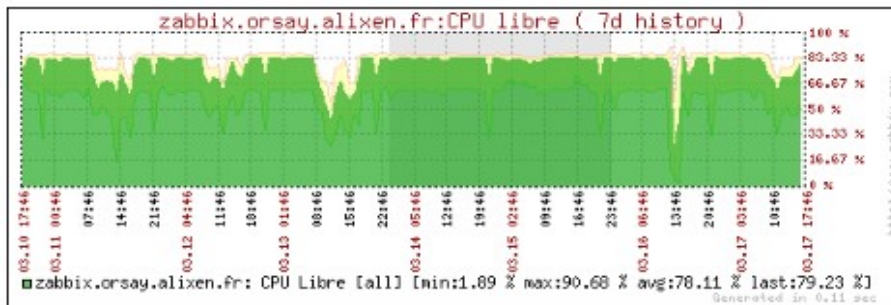
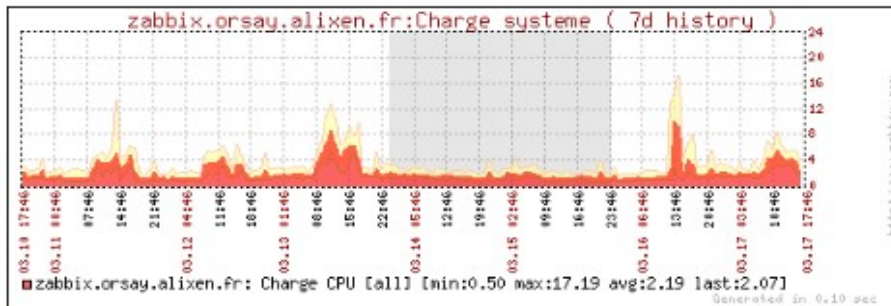
Administration du cluster

- **Ligne de commandes**
 - Outils openVZ
- **GUI à l'étude**
 - Supervision
 - Gestion des conteneurs
 - Sauvegarde et restauration

Supervision

- **Supervision du système hôte et des VMs**
 - Outils de supervision Zabbix
 - Paramètres systèmes
 - Surveillance applicative
- **Supervision spécifique**
 - État du DRBD
 - Consommation des ressources par conteneur

Supervision



OpenVZ : types de réseau

• VENET

- Simplicité, Performance, Sécurité
- Fonctionnalités : pseudo réseau
- Exemples : services web

• VETH

- Bridge
- Broadcast, adresses MAC, etc..
- DHCPD, Samba

Bilan (1)

• Haute Disponibilité

- Redondance des données
- Redondance des services

• Performance

- Proche de systèmes natifs

• Administration des services facilitée

- Un service = une machine virtuelle
- Limitation des dépendances applicatives

Bilan (2)

• Réduction des coûts

- Matériel standard
- Faible coût de stockage (pas de SAN)
- Support matériel (prix H+4 vs J+1)
- OpenSource (aucune gestion de licences)

• Evolution progressive du SI

- Par brique

Plan : Virtualisation et haute disponibilité

• Objectifs

• Réponses apportées

- Architecture matérielle et logicielle
- Administration du cluster
- Bilan

• Pérennité des composants logiciels

• Bonnes pratiques

• Exemples de déploiement

Pérennité : DRBD et heartbeat

- **Plus de 10 ans d'existence**
- **Disponible sur de nombreuses plate-formes**
 - Debian, Redhat Enterprise, ...
- **Communauté active**
 - Développement
 - Forum, MailingList
- **Sortie régulière de versions**

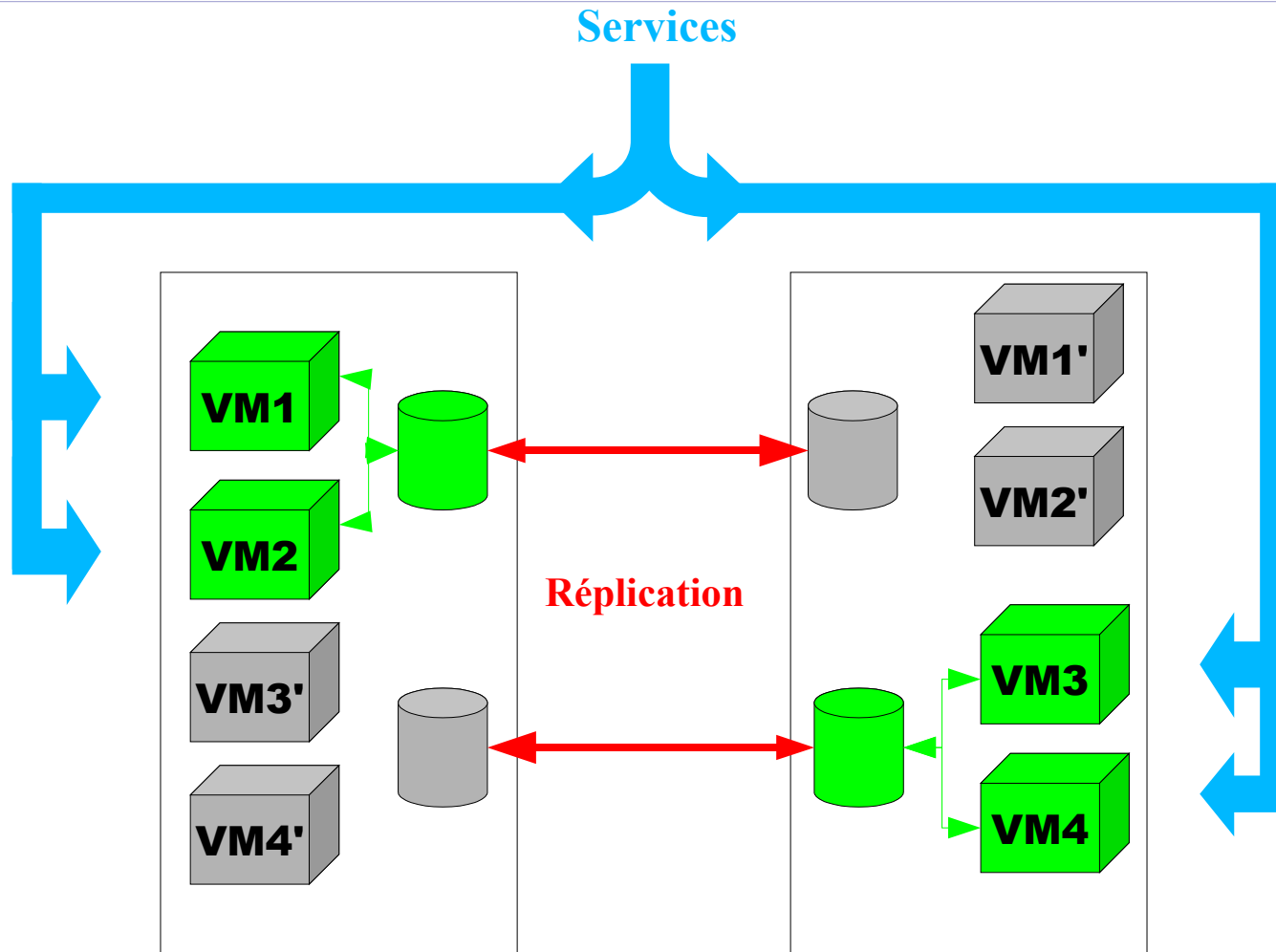
Pérennité OpenVZ

- **Première version stable : 2005**
- **Inclusion dans le noyau Linux**
 - Contrôle mémoire, réseau, etc...
- **Disponible sur de nombreuses plate-formes**
 - Debian, Redhat Enterprise, ...
- **Communauté active**
 - Développement
 - Forum, MailingList

Bonnes pratiques (1)

- **Mise en place d'actif/passif croisé ?**
 - Services répartis sur deux serveurs
- **Actif/Passif croisé sur une seule chaîne de disques**
 - DRBD : IO disque x2
- **Plusieurs grappes de disques**
 - Une grappe = un drbd
 - Performance native des disques

Actif / Passif croisé



Bonnes pratiques (2)

- **Services fonctionnant sur une seule machine**
 - Performance réduite en mode dégradé ?
- **OpenVZ : dimensionnement des conteneurs**
 - Mémoire garantie
 - Mémoire maximale
 - Paramètres secondaires : Buffers TCP, etc...
- **Mémoire**
 - « Plus il y en a, mieux c'est »

Dimensionnement des conteneurs : Méthodologie

- **Relevé de paramètres**
 - IO accounting : utilisation disque
 - Consommation CPU et mémoire
 - Nombre de sockets, fichiers, ... ouverts
- **Réalisation d'un tableau récapitulatif**
 - CPU, RAM, DISQUE, etc...
 - Ressources consommées par services
- **Déduction des besoins et du paramétrage**

Exemples de déploiement (1)

- **Serveurs d'infrastructure hébergés**
 - Services critiques pour la société
 - MX, DNS, ...
 - Inutilité d'un SAN (espace disque nécessaire faible)
- **Apport du cluster HA**
 - Sécurisation des données
 - Disponibilité des services
 - Utilisation maximale des ressources matérielles

Exemples de déploiement (2)

• Services hébergés pour des clients

- Fourniture de services SAAS
- Applications critiques
- Économique

• Services web

- Modification d'un site = pas d'impact sur d'autres
- Sécurité : un site web = une VM
- Pas de conflits de dépendances logicielles

Conclusion

- **Sécurité des données**
 - Réplication des données sur deux serveurs
- **Disponibilité des services**
- **Réduction des coûts**
 - Utilisation efficiente des ressources
 - Administration des services facilitée
 - Aucune licence
- **Évolution progressive du SI**

Virtualisation et Haute Disponibilité

Des questions ?

<http://www.alixen.fr>