

# Infrastructure VO au SIO

ALBERT SHIH<sup>1</sup>

<sup>1</sup>Observatoire de Paris - Meudon

12 mars 2008

# Buts

- Mise en place d'une infrastructure pour les services VO.
- Le meilleur compromis  
qualité de service/sécurité/facilité/prix.

# Buts

- Mise en place d'une infrastructure pour les services VO.
- Le meilleur compromis  
qualité de service/sécurité/facilité/prix.

# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . .).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.

# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . . ).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.

# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . .).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.

# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . .).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.

# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . .).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.



# Sécurité

- Chaque service VO a des besoins propres.
- Configuration particulière (bibliothèques, outils spécifiques. . .).
- Pas toujours compatible avec celui d'un autre service
- Problèmes de sécurité.
- Chaque service doit être indépendante des autres.
- → virtualisation.

# Ce que c'est

- **Serveurs de plus en plus puissant.**
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. ...).
- De nombreuses techniques

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. . . ).
- De nombreuses techniques

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. . . ).
- De nombreuses techniques

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. . . ).
- De nombreuses techniques
  - Virtualisation totale via l'OS : VMware, VirtualPC, Parallels ...
  - Virtualisation au niveau CPU : Xen, KVM ...

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. ...).
- De nombreuses techniques
  - Virtualisation totale via l'OS : VMware, VirtualPC, Parallels ...
  - Virtualisation au niveau CPU : Xen, KVM ...
  - Chrooting : jail (FreeBSD), vserver (Linux), zones (Solaris).

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. ...).
- De nombreuses techniques
  - Virtualisation totale via l'OS : VMware, VirtualPC, Parallels ...
  - Virtualisation au niveau CPU : Xen, KVM ...
  - Chrooting : jail (FreeBSD), vserver (Linux), zones (Solaris).

# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. ...).
- De nombreuses techniques
  - Virtualisation totale via l'OS : VMware, VirtualPC, Parallels ...
  - Virtualisation au niveau CPU : Xen, KVM ...
  - Chrooting : jail (FreeBSD), vserver (Linux), zones (Solaris).



# Ce que c'est

- Serveurs de plus en plus puissant.
- Les services ne consomment pas beaucoup plus de CPU qu'avant.
- Héberger plusieurs machines logiques sur une machine physique.
- Réduction du coût infrastructure (électricité, climatisation etc. ...).
- De nombreuses techniques
  - Virtualisation totale via l'OS : VMware, VirtualPC, Parallels ...
  - Virtualisation au niveau CPU : Xen, KVM ...
  - Chrooting : jail (FreeBSD), vserver (Linux), zones (Solaris).

# Jail sous FreeBSD

- **Natif au système.**
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme



# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme

# Jail sous FreeBSD

- Natif au système.
- Avec un même noyau on peut avoir de nombreuses machines logiques.
- Surcoût au niveau ressource négligeable.
- Étanchéité des OS «guest».
- Configuration propre à chaque services.
- Plusieurs machines logiques sur une machine physique  
→ consolidation → redondance.
- Deux serveurs identiques.
- Possibilité de migrer une machine logique depuis une machine physique sur une autre (à «froid»).
- Migration automatique à terme.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.

# Accès aux machines

- L'accès aux machines ne se fait qu'au travers d'outils comme subversion.
- Pas d'accès `shell` à priori sur les serveurs pour les développeurs.
- Meilleurs sécurités.
- Conservation de différentes versions.
- Accès concurrents pour plusieurs développeurs aux codes sources.
- Possibilité de site test.



# Questions

?