

TReqS : État d'avancement et Perspectives

Jonathan Schaeffer



1 Déploiement

- Déploiement matériel
- Déploiement logiciel

2 Bilan

- Bilan des tests
- Bilan de l'exploitation
- Bilan de l'utilisation

3 Le pain, la planche et nous

- Idées d'amélioration
- C'est en cours ...



Déploiement

- Besoins en ressources théoriquement modestes



Déploiement

- Besoins en ressources théoriquement modestes
- Installation sur une machine virtuelle ccgridvml10



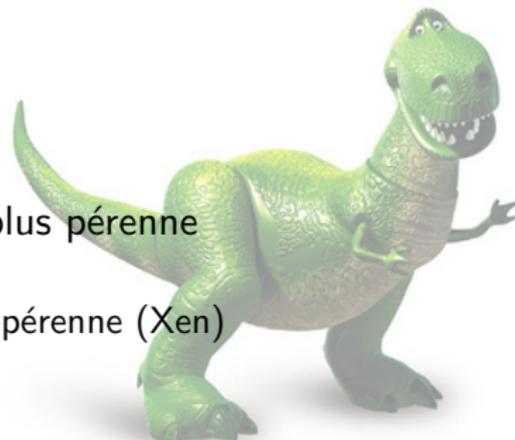
Déploiement

- Besoins en ressources théoriquement modestes
- Installation sur une machine virtuelle ccgridvml10
- Base de donnée MySQL locale



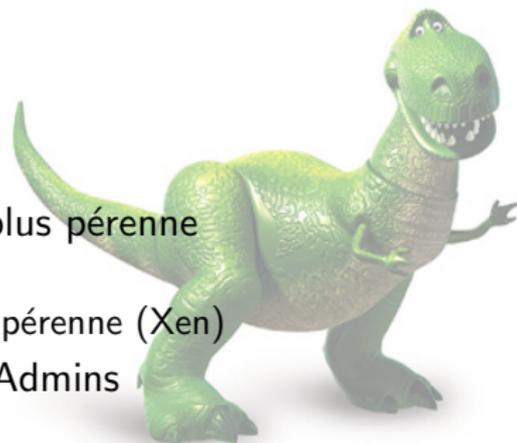
Perspectives d'évolution

- Rester sur une machine virtuelle mais plus pérenne
 - Augmenter les ressources mémoire
 - Basculer un jour vers une virtualisation pérenne (Xen)



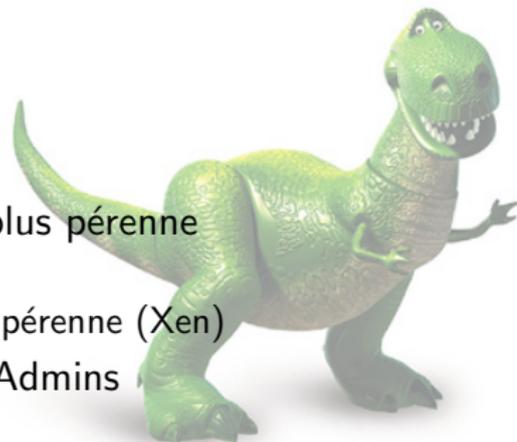
Perspectives d'évolution

- Rester sur une machine virtuelle mais plus pérenne
 - Augmenter les ressources mémoire
 - Basculer un jour vers une virtualisation pérenne (Xen)
- Sous-traiter la base de donnée aux DBAdmins
 - Bénéficier de leur expérience
 - Obtenir une meilleur qualité de service



Perspectives d'évolution

- Rester sur une machine virtuelle mais plus pérenne
 - Augmenter les ressources mémoire
 - Basculer un jour vers une virtualisation pérenne (Xen)
- Sous-traiter la base de donnée aux DBAdmins
 - Bénéficier de leur expérience
 - Obtenir une meilleur qualité de service
 - ⇒ Les DBAdmins Préfèrent l'utilisation d'Oracle

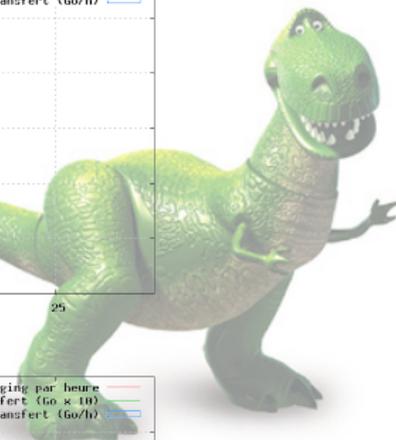
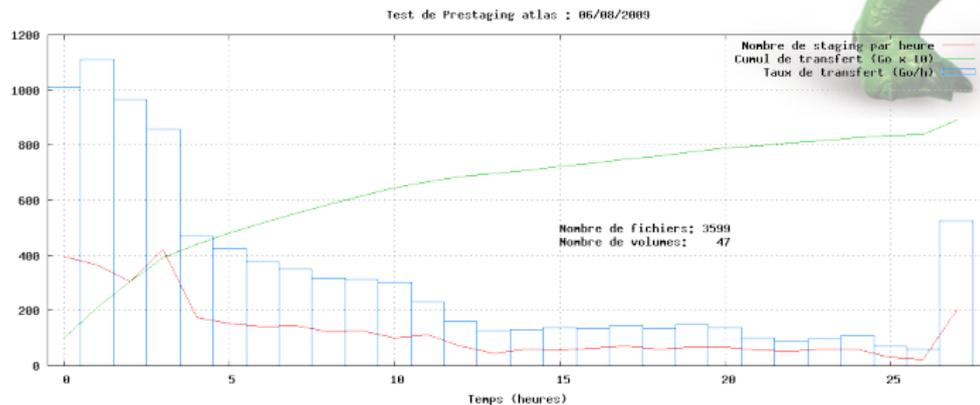
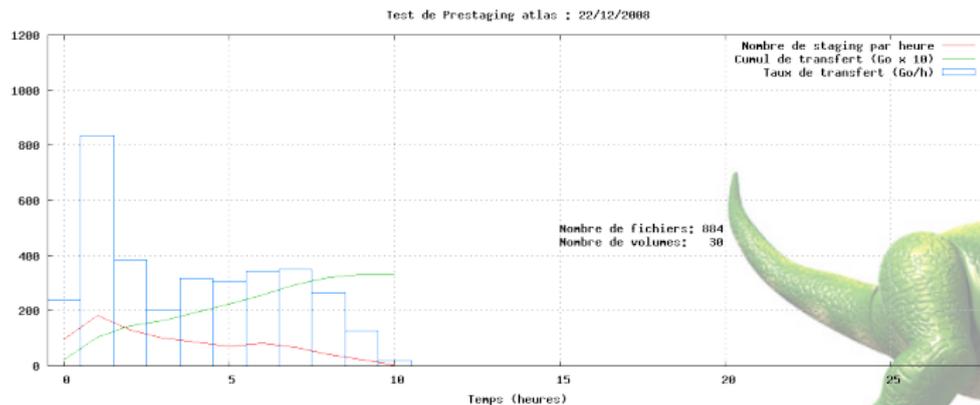


Versions et Déploiement

- La version actuelle est une alpha-release.
- Elle permet de valider l'idée
- Les prochains déploiements auront un numéro de version **X.Y.Z** avec les règles suivantes :
 - Z Z est incrémenté en cas de correction de bugs, pas de changement de structure.
 - Y Y est incrémenté lorsqu'il y a eu de modifications de structures internes. Mais la compatibilité est garantie
 - X Si $X < 1$, version considérée comme non stable. X est incrémenté lors de changement conceptuels important
- Le client TReqS suivra la même numérotation



Bilan des tests



Questions en suspens



- Utilisation dans un environnement plus bridé (moins de drives disponibles)
- Utilisation concurrente de plusieurs expériences

Les plus

- Meilleure exploitation des ressources
 - moins de "mount wait"
 - partage souple des drives entre expériences
 - moins de Slow Jobs



Les plus

- Meilleure exploitation des ressources
 - moins de "mount wait"
 - partage souple des drives entre expériences
 - moins de Slow Jobs
- Exploitation HPSS et dCache plus sereine



Les plus

- Meilleure exploitation des ressources
 - moins de "mount wait"
 - partage souple des drives entre expériences
 - moins de Slow Jobs
- Exploitation HPSS et dCache plus sereine
- Meilleure visibilité de l'activité de staging générée par dCache



Les moins

- L'exploitation de TReqS manque de paramétrage dynamique
- Les statistiques de monitoring ne sont pas fiables

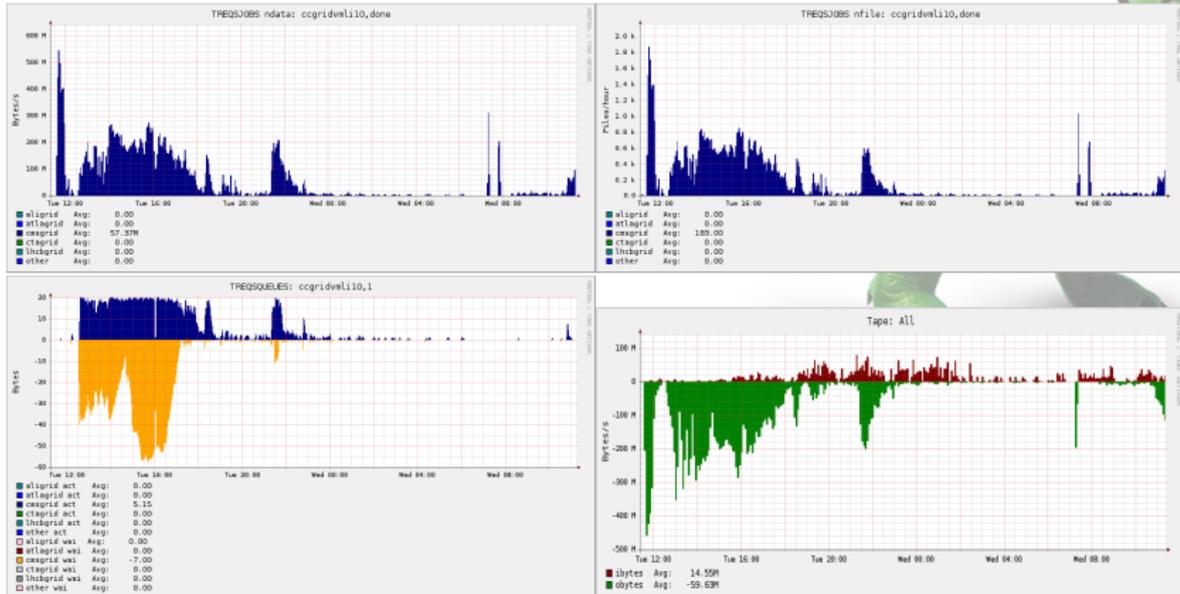


Outils d'exploitation

- Page Web
- RLS
- Documentation du service



Métriques



Prestaging Massif

- Pas beaucoup d'activité de prestaging explicite
- Voir la config de Phedex avec CMS pour éventuellement utiliser directement le client TReqS



Staging dCache

- CTA : Staging de beaucoup de fichiers (dont des petits) séquentiellement
- CMS : Staging de beaucoup de fichiers en bloc (3000 requêtes d'un coup)
- ATLAS : Exercice de reprocessing à deux reprises



Et les bonnes pratiques ?

Il faut rester vigilant aux cas d'utilisation d'HPSS. TReqS est efficace et masque, dans une certaine mesure, les mauvaises pratiques :

- Lecture de petits fichiers
- Mise en exécution de jobs dont les fichiers ne sont pas disponibles

Fonctionnalités

- éviter les retours de tête de lecture
- améliorer la comptabilité



Surveillance

- Utilisation de RLS côté serveur



Exploitation

- proposer des fonctionnalités d'exploitation à chaud
- utiliser une base de donnée des services DBAdmins



Tests

- Déclencher davantage de tests conjoints entre les expériences
- Jouer sur quelques paramètres de l'ordonnanceur pour optimiser les algorithmes



Réécriture du code

- Conception théorique poussée (Andres)
- Création d'un projet dans la forge de l'IN2P3

