



### Ordre du jour :

- Mise en place d'un cycle de réunions régulier
- Utilisation d'un bug tracker pour communiquer sur les demandes et les évènements entre ISTEEX et IPROD
- Point sur l'installation du R630
- Discussion autour de l'architecture de données actuelle

**Présents :** Claude Niederlender, Stanislas Perrin, Roland Lambert, Martial Luc, Patrice Ringot, Philippe Scheffer, Angel Turri, Michel Villaume.

**Durée de la réunion 1h30 / Rédacteur du CR :** Patrice Ringot

### **Mise en place d'un cycle de réunions régulier**

Il est convenu de se réunir régulièrement, chaque semaine tant que cela sera nécessaire pour résoudre les problèmes liés à l'architecture de données d'ISTEEX.

### **Utilisation d'un bug tracker pour communiquer sur les demandes et évènements entre ISTEEX et IPROD**

Les participants sont d'accord pour communiquer via un bug tracker, en l'occurrence un repository gitbucket que l'on nommera **istexsvp**. Y figureront les demandes de travaux ISTEEX et la suite qui leur est donnée. Y figureront également des éléments de communication des uns et des autres (ex : « attention chargement de masse ce week-end » ; « attention vmotion fréquents à prévoir cette semaine pour travail sur ESXI »).

A l'occasion de cette discussion, des problèmes sont évoqués par ISTEEX :

- Durée des pings entre VM > 3 ou 4ms, 7ms des fois.
- Performances variables au niveau de l'API (use case : recherche biblio)

IPROD propose de collaborer à l'élaboration d'un test de performance de l'API en utilisant la plateforme Gatling qui a démontré son intérêt dans le cadre de tests effectués autour de Panist. Cela nous permettra d'avoir un point de référence, sur lequel on pourra s'appuyer pour mesurer nos progrès (ou nos régressions).

Toujours à propos de performance, ISTEEX exprime le souhait de bénéficier de relevés SNMP (mesures système) en complément de leur vision VSphere (mesures hyperviseur) pour mieux comprendre les montées en charge. IPROD est OK.

### **Point sur l'installation du R630**

Le R630 a été installé ainsi que les extensions mémoire. A cette occasion il y a eu pas mal de vmotion (mouvements de VM d'un ESXI à l'autre).

Situation actuelle : 3 ESXI à 384Go de mémoire chacun.

Les ESXI qui sont en veille sur le cluster ISTEEX ne le sont que pour des raisons d'économie d'énergie dès lors qu'ils ne sont pas utilisés (leur mise en veille et leur réactivation est automatique).

La quantité de mémoire à présent disponible incite l'équipe ISTEEX à optimiser ses traitements pour en profiter.

### **Discussion autour de l'architecture de données actuelle**

Claude nous fait un schéma qui indique les masses de données significatives autour de la VP-LI

- Les /data/EDITEURS, qui même sur la VP-LI doivent être en Read/Only (sauf éventuellement pendant une période de traitement initiaux)
  - o Ces FS contiennent actuellement 13To de données effectifs
  - o 6 éditeurs en prod, cela va de quelques To à 100Go selon l'éditeur
  - o Une vingtaine d'éditeurs supplémentaires à prendre en compte (peut-être moins volumiques, plus dans un profil de niche)
  - o Ce FS est accédé en NFS R/O à partir des \*-API, \*-WEB, et ISTEEX-R
  - o L'accès \*-WEB s'explique par le besoin de lier les documents au tableau de bord sur le fonds qui est mis à disposition d'une population restreinte (ABES/INIST).
- Le corpusOutput de 16To qui contient le résultat des traitements d'ingestion
  - o Monté en NFS sur les \*-API pour enrichissement des résultats ElasticSearch, récupération des chemins d'accès aux "fichiers documentaires" (originaux + ceux générés par LoadISTEEX, accès aux fichiers générés par LoadISTEEX (ex : TEI et MODS)
  - o Ce FS contient actuellement 11 To de données sur les 16To alloués. Très grand nombre d'inodes (157 754 829)
  - o Contient une arborescence organisée par éditeur puis par un identifiant documentaire (se référant à un chemin dans le /data/EDITEURS) et dont les 3 premiers caractères permettent une répartition acceptable d'un grand nombre de fichiers (ISTEEX à un nom pour cette répartition : « A/O/F »).
  - o Deux idées pour casser ce FS trop gros :
    - a) Faire un FS par éditeur. Dans ce cas les masses seront déséquilibrées. Rien à faire du côté applicatif
    - b) Faire un FS par premier niveau de la répartition A/O/F, et dans ce cas supprimer le niveau éditeur (en cas de pb sur un FS, tous les éditeurs sont impactés à des degrés divers le temps du rétablissement). Equilibre

les masses. Nécessite une modification dans le code dont la portée reste à évaluer.

- Le li2 de 1To qui contient les résultats intermédiaire de chaque module de la chaine de traitement
  - o Monté en NFS sur les \*-API (on verra pourquoi à l'occasion d'une prochaine réunion)

On se repose la question de la possibilité d'offrir à plusieurs VM en montage local un FS d'un même datastore. IPROD répond que cela n'est pas possible.

## **Conclusions**

Plusieurs pistes d'amélioration ont été évoquées lors de cette réunion

- Découpage du corpusOutput : ISTEEX réfléchit à la solution b) (suppression du niveau éditeur) qui permettra de couper le FS actuel en 16 FS plus « gérables »
- Voir si le NAS ISTEEX (Synology RS814), mieux positionné sur le réseau par rapport aux serveurs, et doté de plus de surface disques, est en mesure de soulager la charge sur la partie production ou API (ex : utilisation pour ISTEEX-R et travaux préparatoires)
- Il est urgent de faire une réunion « sauvegarde » qui permettrait de diminuer la sollicitation des ressources (détection de cas de sauvegardes inutiles ou trop fréquentes).
- Mise en place d'un test de perf Gatling sur l'API

## **Prochaine réunion (n+1)**

- Dédiée essentiellement à la sauvegarde
- ISTEEX nous donne si possible dans le temps imparti son avis sur la solution b)

## **Prochaine réunion (n+X)**

- Présentation visuelle de l'API
- Les ISTEEX et les IPROD échangent sur l'opportunité d'utiliser des FS distribués/répliqués (motivations, cas d'utilisation, pistes techniques envisagées, alternatives).