Grappe de PC pour VizieR

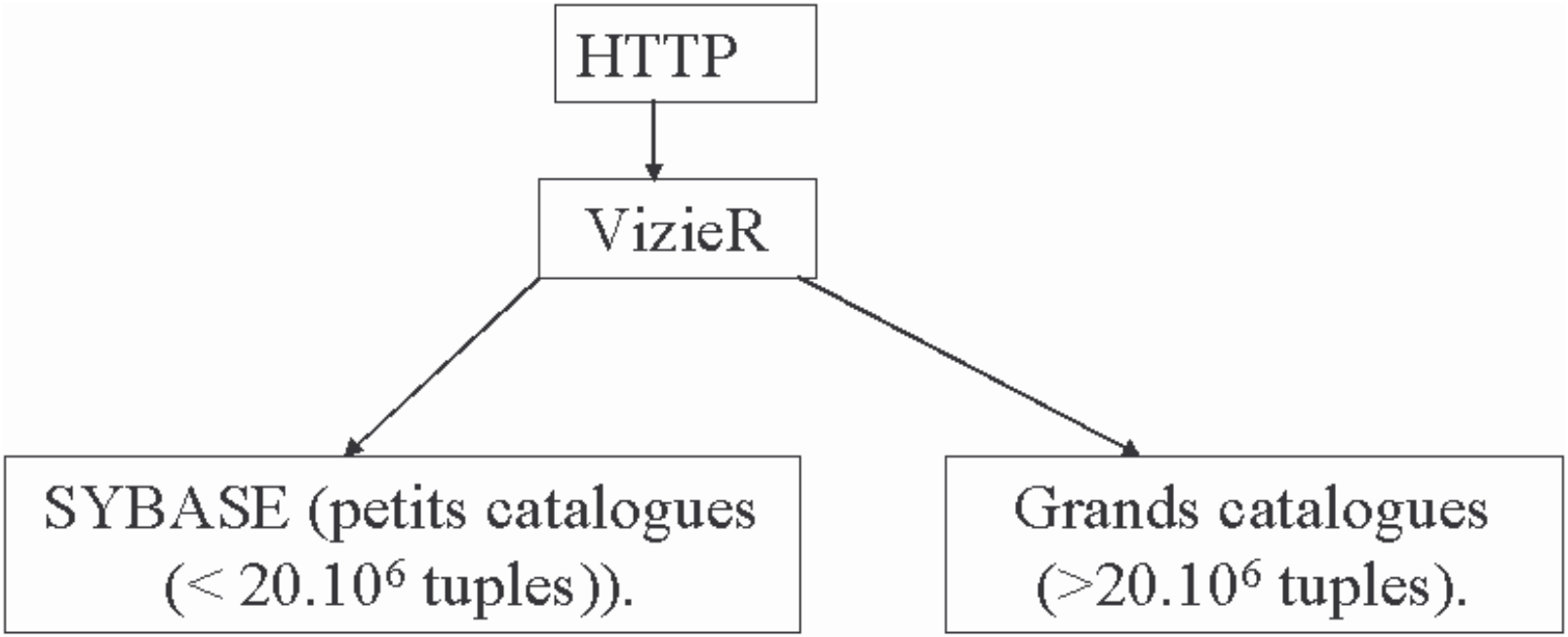
André Schaaff

Contexte (1)

- VizieR fournit un accès à la plus complète librairie de catalogues organisés, documentés et disponibles en ligne.
- Des outils d'interrogation permettent à l'utilisateur de sélectionner des tables pertinentes, d'extraire et de formater des enregistrements correspondants aux critères de recherche.
- L'accès aux très grands catalogues a été optimisé : Guide Star Catalogs, USNO-B1, 2MASS…
  - Ordre de grandeur : plusieurs centaines de millions d'objets
  - Pour chacun de ces catalogues : stockage sous forme binaire + programme associé

Contexte (2)

- Les catalogues de VizieR peuvent se classer en 2 catégories :



Futur proche : arrivée de très grand catalogues (ex : SuperCosmos)

=> Besoin en matière de données

Contexte (3)

- La base de données ainsi que les traitements sont actuellement centralisés sur un serveur Sun.
- Nous souhaitons délocaliser les plus gros catalogues (format binaire, Sybase pour les autres catalogues) ainsi que les traitements les plus coûteux en temps machine sur plusieurs serveurs.
  - Identifications croisées entre les catalogues

Quelle stragégie ?

- Serveur unique
- Grappe de PC sous Linux
  - Distribution des données
  - Clonage des données

Grappes de PC

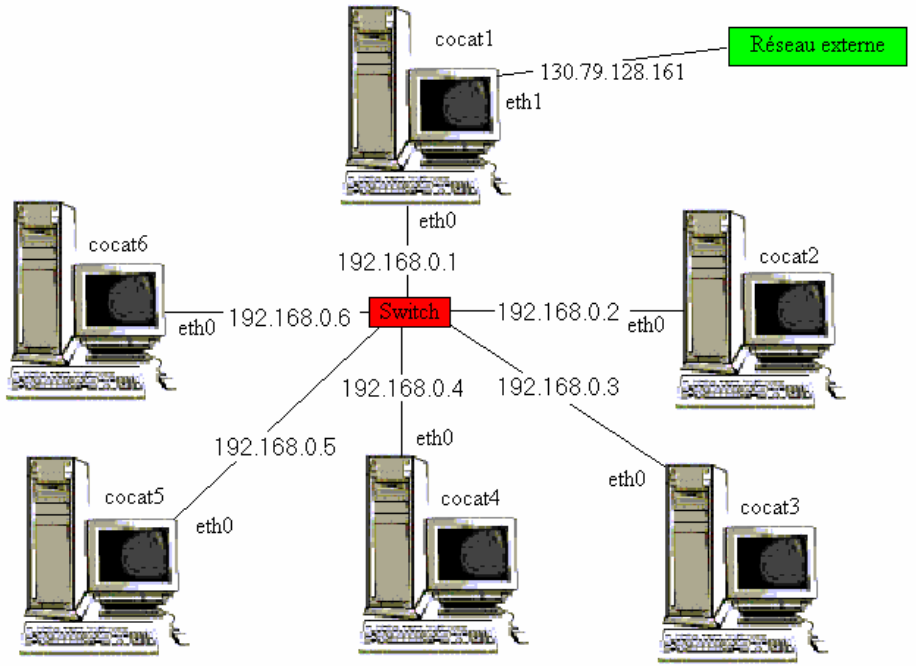
- **Différentes approches :**
  - **Utilisation d'outils de clustering**
    - Notions de serveur, Golden Node, clonage des nœuds, …
    - Les machines doivent être de préférence identiques
    - En théorie : facilité d'installation, de maintenance, …
    - Délégation du dispatching
  - **Gestion des machines comme un ensemble de ressources**
    - Les machines peuvent être hétérogènes
    - Il faut assuer le dispatching (équilibrage de charge, détection de blocages, etc.)

Notre démarche

- **Première approche : utilisation de CLIC**
  - **Partenariat MandrakeSoft, Bull et l'INPG/INRIA**
  - **Financement RNTL**
  - **Objectifs :**
    - simple et facile à installer
    - unifier l'ensemble des phases d'installation, de configuration de la couche d'interconnexion et de déploiement des applications parallélisées
  - **3 phases :**
    - développement et publication d'une distribution Linux contenant tout les outils nécessaires au déploiement rapide d'une grappe de calcul prête à l'utilisation.
    - publication d'outils spécifiques d'administration, de contrôle et d'évaluation pour la solution de clustering
    - publication d'outils et d'applications spécialisés pour le développement en environnement parallèle.

La grappe

- Une petite configuration de 6 PC (au total 6Go de RAM et 2,4 To de disque) pour la phase d'étude, de prototypage et la première version publique



La première étude

- Installation de CLIC dans le cadre d'un stage d'ingénieur
  - Clonage des données sur l'ensemble des nœuds
  - Développement d'un dispatcher MPI
- « CLIC s'installe en 2h » ?
- Premiers tests pas très concluants en terme de performances, dispatcher MPI inadapté au dispatching de requêtes simples
- Manque de souplesse
- CLIC est trop lié à Mandrake, CLIC Phase2 vient juste d'être publié mais il y a un manque de visibilité certain
- …

Autre projet

- Démarrage d'un autre projet au CDS dans le cadre d'un stage (P. Fernique, F. Bonnarel)
  - Images Aladin
  - 3 nœuds
  - Développement d'un dispatcher générique
  - …
- Partage des ressources matériels entre les projets ?
  - Mutualiser

Démarche actuelle

- Réinstallation de toutes les machines de la grappe mais conservation (dans un premier temps) du clonage des données.
- Considération des nœuds de la grappe comme des ressources indépendantes
- Test du dispatcher générique pour l'équilibrage des charges dans le cadre des requêtes sur les grands catalogues (cas le plus simple)
  - Dans le premier dispatcher cela se faisait en utilisant MPI
  - Réserver l'utilisation de MPI aux tâches de calcul (lorsque le pourcentage de code séquentiel est inférieur à un certain seuil)
- Préparation de l'arrivée des très grands catalogues (augmentation du nombre de disques pour maintenir le clonage ?, répartition des données ?, …)

Suite de la démarche ?

- Mettre les ressources matérielles des 2 projets en commun ?
- Un dispatcher capable de gérer la répartition des charges dans un environnement hétérogène
  - Distributions Linux différentes
  - Tolérance au niveau du hardware
  - Gestion des pannes ou des blocages des ressources
  - La défaillance du dispatcher doit pouvoir être détecter et celui-ci doit pouvoir être réactiver
- Un dispatcher distribuant des tâches de calcul en parallèle (avec ou sans MPI ?)

Conclusion

- En d'autres termes, une Grille « light » adaptée à des besoins spécifiques