

Présentation générale de la conception des modules UMS,TMS,FMS et IMS

Mai 2011

Conception commune aux différents modules



- Interfaces
 - **Services** fournis via une API C++
 - **Wrappers** pour ligne de commande, Python, WS
- Communications internes
 - Utilisation de **DIET** : appels vers les SeDs
 - Utilisation de **SSH** : accès compte user, appels non-SeDs
- Modèles de données
 - Modèle objet **EMF** utilisé dans l'API C++ et en interne
 - Modèle relationnel utilisé pour la persistance

- API C++
 - fonctions indépendantes
 - données: classes de données pures (génération EMF)
 - exceptions Vishnu (System/Module)
- Wrappers:
 - ligne de commande (C++)
 - Python (génération par SWIG)
 - Java (génération par SWIG)
 - Web Services (utilise l'API Java)

Communications internes



- Utilisation de DIET : appels client / SeDs
- Utilisation de SSH : accès compte user, appels non-SeDs, transferts de fichiers
- Communication avec la base de données (protocole lié au type de base de données)

Modèles de données



- Modèle objet EMF utilisé dans l'API C++ et en interne
- Modèle relationnel utilisé pour la persistance

Fonctions communes aux modules



- Accès à la base de donnée
 - couche d'abstraction des API psql/mysql
- Gestion des exceptions
 - continuité des exceptions du serveur DIET aux API externes
- Gestion de l'historique des appels Vishnu
 - modèle générique pour toutes les fonctions de l'API Vishnu

Module UMS

Conception générale (1)



- Contrôle les aspects d'authentification et autorisations des utilisateurs pour tous les modules VISHNU.
- La clé de session est un *hash code* qui constitue le lien entre tous les modules en évitant une authentification systématique par login/password.
 - L'obtention d'une clé de session pour utiliser VISHNU n'est possible qu'après une authentification par login/password.
 - L'utilisation de la clé de session en ligne de commande est transparente à l'utilisateur en utilisant un cache.

Module UMS

Conception générale (2)



- Les Processus daemons
 - Un processus daemon coté serveur gère la durée de vie des sessions en mode *CLOSE ON TIMEOUT*.
 - En ligne de commande:
 - Un processus daemon coté client gère la fermeture des sessions ouvertes en mode *CLOSE ON DISCONNECT* dans des terminaux.

Module FMS

Conception générale



- Permet la gestion de fichiers situés sur une machine distante.
 - Manipulations des propriétés des fichiers
 - Transferts de fichiers de manière synchrone ou asynchrone
 - Depuis une machine locale vers une machine distante,
 - depuis une machine distante vers une machine locale,
 - entre deux machines distantes.
- Toutes les requêtes sont exécutées par un unique SeD via des envois de commandes distantes (ssh).



- Le suivi et l'historique des transferts est réalisé via un cache dans la base de données
 - Une estimation de la progression des transferts sera disponible
 - Un état des transferts soumis sera disponible.

Module IMS

Conception générale



- Interaction publisher/subscriber avec le Log Service
- Création d'un cache dans la base de données :
 - Des états des machines monitorées
 - Des états des serveurs VISHNU
- Serveur multi-thread:
 - Requêtes DIET
 - Rafraichissement du cache
- Un serveur IMS donné supervise:
 - Les processus VISHNU locaux
 - Les autres serveurs IMS

Module IMS

Conception spécifique aux services



- Interdépendance avec les autres modules:
 - Délestage (FMS, TMS)
 - Export des commandes (UMS, FMS, TMS)
- Export, depuis un format générique, dans un fichier script shell
 - Connection avec password non incluse par sécurité
- Stop et délestage “HARD”: Serveur non redémarré automatiquement

Module TMS

Conception spécifique aux services



- Gestion des tâches des utilisateurs VISHNU.
- Uniformisation de la gestion des tâches pour différents gestionnaires de ressources:
 - Création d'un identifiant de job unique
 - Ecriture d'un script générique indépendant de la plate-forme de soumission
 - Permet de soumettre un job sur différents types de plate-formes sur lesquelles sont installés des gestionnaires de ressources différents (ex: TORQUE ou LOADLEVELER)
- Processus esclave
 - Permet de soumettre ou d'annuler un job sous le nom de l'utilisateur nominatif de la machine de soumission
 - Utilisé par deux services : submitJob et cancelJob

Module TMS

Conception spécifique aux services



- Création d'un cache dans la base de données :
 - Informations sur les tâches soumises
- Processus daemon
 - Permet de mettre à jour les états des jobs dans la base de données
 - Permet de récupérer la date de commencement de l'exécution de chaque job (start time) par le gestionnaire de ressources sous-jacent