

VISHNU D1.1b - Spécifications techniques des besoins



COLLABORATORS

| | TITLE : VISHNU D1.1b - Spécifications techniques des besoins | | |
|---------------|--|----------------|------------------|
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | Benjamin Isnard, Daouda Traoré, and Eugène Pamba Capo-Chichi | April 25, 2013 | |

REVISION HISTORY

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------------|--|---------|
| 01 | 07/12/2010 | Exemple pour validation du format | SysFera |
| 02 | 14/01/2011 | Version du livrable | SysFera |
| 03 | 18/01/2011 | Ajout de SWIG,Python,Java dans §3.1. | SysFera |
| 04 | 22/07/2011 | Prise en compte de SLURM. | SysFera |
| 05 | 18/01/2012 | Mise à jour de versions manquantes. | SysFera |
| 06 | 14/02/2012 | Prise en compte du support de LDAP. | SysFera |
| 07 | 28/02/2012 | Prise en compte du support de LSF. | SysFera |
| 08 | 26/03/2012 | Ajout des caractéristiques de circulation du mot de passe. | SysFera |
| 09 | 11/04/2012 | Prise en compte du support de Grid Engine. | SysFera |
| 10 | 31/08/2012 | Support SLURM 2.3 et 2.4, new cmake version. | SysFera |
| 10 | 22/01/2013 | Clean corba/sysfera-ds. | SysFera |



Contents

| | | |
|----------|--|----------|
| 1 | Présentation du document | 1 |
| 1.1 | Objectifs | 1 |
| 1.2 | Structure du document | 1 |
| 1.3 | Références | 1 |
| 1.4 | Glossaire | 1 |
| 2 | Sécurité du système et des données | 2 |
| 2.1 | Tableau des Exigences | 2 |
| 2.2 | Glossaire | 4 |
| 3 | Environnement matériel et logiciel | 5 |
| 3.1 | Tableau des Exigences | 5 |
| 3.2 | Glossaire | 7 |
| 4 | Performances, fiabilité et robustesse | 8 |
| 4.1 | Tableau des Exigences | 8 |
| 4.2 | Glossaire | 10 |

Chapter 1

Présentation du document

1.1 Objectifs

Ce document présente les spécifications techniques des besoins pour le système Vishnu. Elles sont basées sur les documents de Cahier des charges du système Vishnu (voir Références) et sur la proposition technique de la société Sysfera. Ces spécifications incluent les contraintes d'implémentation du logiciel. L'objectif principal de ces spécifications est de valider l'adéquation de la solution proposée par rapport aux contraintes de l'environnement d'utilisation.

Ce document pourra également contenir des prévisions sur les évolutions futures du logiciel et préciser quelles fonctions devraient être faciles à ajouter ou supprimer.

1.2 Structure du document

Le document est composé de 3 parties correspondant à des domaines techniques différents:

- Besoins liés à la sécurité du système et des données
- Besoins liés à l'environnement matériel et logiciel
- Besoins en performance, fiabilité et robustesse

Chaque partie contient un tableau des exigences qui fait l'inventaire des tous les besoins techniques concernant le domaine spécifié. Ce tableau est suivi d'un dictionnaire des termes techniques employés afin de les préciser le cas échéant.

1.3 Références

- Cahier des charges pour les services de base de l'intergiciel réparti VISHNU [EDF R&D, Juillet 2010]
- Compléments au cahier des charges pour les services de base de l'intergiciel réparti VISHNU [EDF R&D, Août 2010]

1.4 Glossaire

- UMS ("User Management System"): nom du module Vishnu de gestion des sessions et des utilisateurs
 - TMS ("Tasks Management System"): nom du module Vishnu de gestion des tâches
 - FMS ("Files Management System"): nom du module Vishnu de gestion des transferts de fichiers
 - IMS ("Information Management System"): nom du module Vishnu de gestion des informations
-

Chapter 2

Sécurité du système et des données

2.1 Tableau des Exigences

| ID | Name | Text |
|-------|--|---|
| 1 | Sessions et authentification | |
| 1.1 | Types d'utilisateurs | Les utilisateurs de l'intergiciel réparti VISHNU seront de deux types : utilisateurs ou administrateurs. Les administrateurs sont des utilisateurs avec des droits supplémentaires. L'identification dans le système d'un utilisateur se fera à l'aide d'un login unique et d'un mot de passe qui sera crypté. |
| 1.2 | Format des identifiants et mots de passe | Le login et le mot de passe respecteront respectivement les expressions régulières suivantes: ^[A-Za-z0-9_]{LOGIN_MIN_SIZE,LOGIN_MAX_SIZE}\$ et ^[^\s]{PASSWORD_MIN_SIZE,PASSWORD_MAX_SIZE}\$ |
| 1.3 | Caractéristique d'une authentification | Une authentification se fait à l'aide d'un login et d'un mot de passe qui doivent être au préalable enregistrés dans la base de données pour vérification. Le login est unique et le mot de passe enregistré est crypté en utilisant la fonction crypt de la librairie libcrypt de Linux. La procédure d'authentification est donc sécurisée en ce sens que le login est unique dans la base de données et que le mot de passe est crypté par un algorithme efficace SHA-512. Avant tout ajout d'un nouvel utilisateur via vishnu dans la base de données le système vérifie que le login n'est pas déjà utilisé. |
| 1.4 | Authentification unique | Une fois authentifié dans l'intergiciel, l'utilisateur n'aura plus à s'authentifier pour utiliser les services de l'intergiciel avec son propre compte. Un identifiant de session sera créé par le système et retourné à l'utilisateur. Cet identifiant est ensuite enregistré dans une variable VISHNU_SESSION_ID pour éviter d'avoir à fournir cette information pour chaque requête (dans le cadre du client shell Unix). La complexité de cet identifiant garantit son unicité mais surtout un niveau de sécurité pour une utilisation du système sans authentification systématique. |
| 1.5 | Fermeture de session | |
| 1.5.1 | Fermeture manuelle de session | L'utilisateur peut fermer manuellement la session dans laquelle il se trouve. Avant la fermeture, le système vérifie la présence de commandes en cours d'exécution dans la session. Si il existe une ou plusieurs commandes en cours d'exécution le système rend impossible la fermeture de session sinon la session est fermée. Cette fermeture est possible en utilisant l'identifiant de session défini dans la variable VISHNU_SESSION_ID. |

| ID | Name | Text |
|-------|--|--|
| 1.5.2 | Choix de l'option de fermeture automatique de session | L'utilisateur peut choisir entre deux options de fermeture automatique de session: soit l'option de fermeture à la déconnexion du terminal, soit l'option de fermeture après expiration du délai d'inactivité (période sans aucun appel aux services de l'intergiciel). Le choix est enregistré dans la configuration permanente de l'utilisateur et peut également être modifié pour une session donnée. Le choix de fermeture à la déconnexion du terminal n'est possible que dans le cas d'une session de type "shell unix". Dans les deux cas, la présence de commandes en cours d'exécution rend impossible la fermeture de session. Par contre, pour l'option de fermeture après expiration du délai d'inactivité, il y a une réinitialisation du délai d'inactivité lorsqu'il y a des commandes en cours d'exécution. |
| 1.5.3 | Vérification des commandes en cours avant fermeture de session | Le système vérifie que toutes les commandes lancées pendant la session sont terminées avant de fermer la session. Dans le cas où l'utilisateur s'est déconnecté de son terminal et que la session doit être fermée automatiquement, la session ne sera pas fermée avant la fin de l'exécution de toutes les commandes en cours, et elle sera fermée après expiration du délai d'inactivité. |
| 1.6 | Reconnection à une session | Un utilisateur peut se connecter à une session déjà ouverte sur la même machine cliente en fournissant un identifiant unique de session ainsi que son login et mot de passe. S'il se trouve déjà dans une session ouverte, le système lui propose de fermer cette session ou bien de la conserver ouverte avec une déconnexion automatique au bout du temps maximum d'inactivité. |
| 1.7 | Accès aux informations sur les sessions | Pour un utilisateur normal, l'affichage des sessions (actives ou fermées) n'est possible qu'après une authentification à partir du login et du mot de passe de l'utilisateur. Un administrateur peut par contre accéder à toutes les informations de sessions de tous les utilisateurs. |
| 1.8 | Accès aux machines | |
| 1.8.1 | Création et modification d'accès | L'accès à une machine au travers de l'intergiciel Vishnu se fera à l'aide d'un couple clé SSH privée/publique spécifique à chaque couple (machine, utilisateur). L'utilisateur du système Vishnu devra ajouter la clef publique aux clefs autorisées sur son compte Unix. |
| 1.8.2 | Révocation d'un accès au compte unix par un utilisateur | L'utilisateur pour supprimer l'accès au travers de l'intergiciel Vishnu à son compte Unix sur une machine donnée en supprimant simplement la clef publique de la liste des clefs autorisées sur son compte. |
| 1.8.3 | Respect des autorisations d'accès | L'accès au compte de l'utilisateur sur une machine et aux informations y étant stockées se faisant via SSH avec le compte de l'utilisateur, tous les droits d'accès système seront respectés. |
| 1.8.4 | Utilisation de comptes individuels | Un utilisateur Vishnu ne peut enregistrer un compte Unix sur une machine donnée que si ce compte n'est pas déjà utilisé par un autre utilisateur Vishnu. Si c'est le cas, le système Vishnu doit refuser la création (ou modification) de l'accès à la machine. |
| 1.9 | Changement de mot de passe | Un utilisateur peut modifier son mot de passe lorsqu'il a déjà ouvert une session. Le nouveau mot de passe est actif immédiatement c'est-à-dire que toute nouvelle authentification se fera avec le nouveau mot de passe. |
| 2 | Sécurité des communications | |
| 2.1 | Données des requêtes au système | Les requêtes au système vishnu sont authentifiées mais non cryptées. |
| 2.2 | Communications à travers un firewall | Les communications entre machines du système Vishnu pourront être configurées pour traverser un firewall permettant les communications cryptées. |

| ID | Name | Text |
|-----|--|--|
| 2.3 | Fichiers des utilisateurs | Les fichiers des utilisateurs sont transférés en utilisant le cryptage SSH avec les clefs de l'utilisateur. Pour qu'un transfert puisse avoir lieu entre deux machines l'utilisateur doit avoir configuré ses clefs SSH (indépendamment de Vishnu) pour permettre une communication entre ces deux machines. |
| 2.4 | Informations stockées par le système | Le système Vishnu stockera toutes les informations relatives aux utilisateurs, aux tâches, transferts de fichiers, mesures de performance et trace des requêtes dans une base de données PostgreSQL sans être cryptées. L'accès du système Vishnu à la base de données sera authentifié par login/mot de passe dont l'envoi au travers de la connexion sera réalisé après hachage MD5. Les communications entre le système Vishnu et la base de données pourront être cryptées si nécessaires par le protocole SSL en configurant le système Vishnu et la base de données. |
| 3 | Sécurité des fonctions administrateurs | |
| 3.1 | Configuration de l'option par défaut de fin de session | Un administrateur peut définir quelle est l'option par défaut pour la fermeture automatique d'une session utilisateur: soit terminaison en cas de déconnexion de l'utilisateur du client, soit terminaison après expiration du délai d'inactivité |
| 3.2 | Utilisation d'un compte tiers | Un administrateur de l'intergiciel pourra ouvrir une session VISHNU avec le compte d'un utilisateur n'étant pas administrateur lui-même. |

2.2 Glossaire

- **Firewall (ou pare-feu en français):** est un matériel ou un logiciel qui impose une politique de sécurité pour protéger un réseau en constituant une barrière (un filtre) avec d'autres réseaux.
- **MD5:** est un acronyme de Message Digest 5. C'est une fonction de hachage cryptographique.
- **SHA-512:** SHA-512 est un acronyme de Secure Hash Algorithm avec une empreinte de 512 bits. C'est une fonction de hachage cryptographique dérivée de SHA-1.
- **SSH:** à la fois programme exécutable et protocole de communication sécurisé utilisant un échange de clés de chiffrement en début de connexion.

Chapter 3

Environnement matériel et logiciel

3.1 Tableau des Exigences

| ID | Name | Text |
|---------|--|---|
| 1. | Plateformes supportées | |
| 1.1 | Machines clientes | |
| 1.1.1 | Système d'exploitation | Debian Linux v5 (Lenny) ou ultérieure |
| 1.1.2 | Architecture matérielle | Intel x86 |
| 1.1.3 | Nombre et vitesse des processeurs | Monoprocasseur ou multiprocasseur, 2GHz minimum |
| 1.1.4 | Mémoire vive | Minimum 1Go |
| 1.1.5 | Espace disque | Minimum 1Go disponible |
| 1.1.6 | Logiciels requis | SysFera DS, SSH (client/serveur), RSync |
| 1.1.7 | Variables d'environnement | La clé RSA ou DSA du serveur SSH (chemin complet sur la clé publique) afin d'identifier le client |
| 1.1.8 | Interface réseau | Adresse IP dynamique et nom DNS dynamique |
| 1.2 | Serveurs de calcul ou de stockage | |
| 1.2.1 | Système d'exploitation | Debian Linux v5 (Lenny) ou ultérieure |
| 1.2.2 | Architecture matérielle | Intel x86 ou IBM BlueGene/P® (PPC 32-bits) |
| 1.2.3 | Nombre et vitesse des processeurs | Monoprocasseur ou multiprocasseur, 2GHz minimum |
| 1.2.4 | Mémoire vive | Minimum 1Go |
| 1.2.5 | Espace disque | Minimum 10 Go disponible |
| 1.2.6 | Logiciels de gestion de jobs batch supportés | <ul style="list-style-type: none"> - Torque 2.3.6 - IBM LoadLeveler 3.5.1.14 - SLURM 2.2.1 (ou 2.2.7) - SLURM 2.3 ou 2.4 - LSF 7.0.6.134609 - Grid Engine 2011.11 |
| 1.2.7 | Logiciels requis | SSH (client/serveur), RSync, Client database (MySQL ou PostgreSQL), ZMQ |
| 1.2.8 | Ports pour tunnel(s) SSH | 1 port réseau disponible pour un tunnel SSH si serveur placé derrière un firewall |
| 1.2.9 | Interface réseau | Adresse IP et nom DNS statiques |
| 1.3 | Serveurs dédiés VISHNU | |
| 1.3.1 | Nombre de serveurs | Au minimum 2 serveurs dédiés. Un ratio de 1 serveur supplémentaire pour 5 serveurs de calcul ou stockage est conseillé. |
| 1.3.2 | Plateforme | |
| 1.3.2.1 | Système d'exploitation | Debian Linux v5 (Lenny) ou ultérieure |

| ID | Name | Text |
|---------|---|--|
| 1.3.2.2 | Architecture matérielle | Intel x86 |
| 1.3.2.3 | Nombre et vitesse des processeurs | Monoprocasseur ou multiprocasseur, 2GHz minimum |
| 1.3.2.4 | Mémoire vive | Minimum 2 Go |
| 1.3.2.5 | Espace disque | Minimum 10 Go disponibles |
| 1.3.2.6 | Ports pour tunnel(s) SSH | N ports réseau disponibles pour un/des tunnel(s) SSH où N est le nombre de serveurs de calcul ou de stockage placés derrière un firewall. |
| 1.3.2.7 | Interface réseau | Adresse IP et nom DNS statiques |
| 1.4 | Serveur Web (pour API Web Services) | |
| 1.4.1 | Plateforme | |
| 1.4.1.1 | Système d'exploitation | Debian Linux v5 (Lenny) ou ultérieure |
| 1.4.1.2 | Architecture matérielle | Intel x86 |
| 1.4.1.3 | Nombre et vitesse des processeurs | Monoprocasseur ou multiprocasseur, 2GHz minimum |
| 1.4.1.4 | Mémoire vive | Minimum 2Go |
| 1.4.1.5 | Espace disque | Minimum 500 Go (espace nécessaire au stockage de fichiers temporaires) |
| 1.4.1.6 | Interface réseau | Adresse IP et nom DNS statiques |
| 1.4.2 | Machine virtuelle Java | Java SDK v1.6.0 |
| 1.4.3 | Autres logiciels requis | ZMQ(version 2), SSH (client/serveur), RSync, supervisord (version 3.0a12), xmlrpc (version 2.1.19) |
| 1.5 | Serveur de Base de données | |
| 1.5.1 | Plateforme | |
| 1.5.1.1 | Système d'exploitation | Debian Linux v5 (Lenny) ou ultérieure |
| 1.5.1.2 | Architecture matérielle | Intel x86 |
| 1.5.1.3 | Mémoire vive | Minimum 2Go |
| 1.5.1.4 | Nombre et vitesse des processeurs | Monoprocasseur ou multiprocasseur, 2GHz minimum |
| 1.5.1.5 | Espace disque | Minimum 500 Go |
| 1.5.1.6 | Ports pour tunnel(s) SSH | N ports réseau disponibles pour un/des tunnel(s) SSH où N est le nombre de serveurs de calcul ou de stockage placés derrière un firewall. |
| 1.5.1.7 | Interface réseau | Adresse IP et nom DNS statiques |
| 1.5.2 | SGBD supportés | L'un des deux SGBD suivantes doit être installé: - PostgreSQL version 8.0 ou ultérieure - MySQL (v5.1.49) |
| 1.5.3 | Réplication de la base de données | Afin de garantir un temps de réponse maximum aux requêtes des machines de la plate-forme VISHNU, la base de donnée pourra être répliquée selon une architecture maître-esclave (écriture sur un seul serveur, lecture sur plusieurs serveurs). |
| 1.6 | Infrastructure réseau | |
| 1.6.1 | Connexion entre serveurs (calcul/stockage/VISHNU) et serveur de base de données | |
| 1.6.1.1 | Débit | Minimum 100Mb/s |
| 1.6.1.2 | Temps de réponse | Maximum 100ms |
| 1.6.1.3 | Protocoles autorisés | TCP SSH (port 22) minimum |
| 1.6.2 | Connexion entre clients (ou serveur web) et serveurs (calcul/stockage/VISHNU) | La connexion entre le client et le serveur se fait de manière non cryptée, ainsi les informations telles les mots de passe circulent en clair |
| 1.6.2.1 | Débit | Minimum 10Mb/s |
| 1.6.2.2 | Temps de réponse | Maximum 200ms |

| ID | Name | Text |
|---------|---|--|
| 1.6.2.3 | Protocoles autorisés | TCP SSH (port 22) minimum |
| 1.6.3 | Connexion vers un serveur de messagerie | Les serveurs de calcul et de stockage de la plateforme VISHNU doivent pouvoir se connecter sur un serveur de messagerie via le protocole SMTP. |
| 1.6.3.1 | Compte de messagerie | Un compte de messagerie sera utilisé par le système VISHNU pour envoyer des messages email aux utilisateurs. |
| 2. | Versions des logiciels requis | |
| 2.1 | SSH | Serveur ssh et client ssh - version openSSH 4.2 |
| 2.1.1 | SSL | OpenSSL 0.9.8k |
| 2.3 | Librairie BOOST C++ | v1.46.1 |
| 2.4 | Librairies des Batch schedulers | Torque API, LoadLeveler API, SLURM API, LSF API et Grid Engine API |
| 2.5 | Logiciels de transferts de fichiers | RSync v3.0 |
| 2.8 | Client database | - PostgreSQL client v8.0 - MySQL v5.1.49 |
| 2.9 | LDAP | OpenLDAP version 2.4 |
| 3. | Outils de compilation | |
| 3.1 | CMake | CMake v2.8.5 |
| 3.2 | GCC | GCC v4.4.3 (Janvier 2010) |
| 3.3 | SWIG | SWIG v1.3.40 |
| 3.4 | Python | Python v2.5 ou v2.6 |
| 3.5 | Java | JDK 1.6 |
| 3.6 | GLIBC | Version 2.7 pour un support natif du cryptage |

3.2 Glossaire

- **SGBD**: (Système de gestion de bases de données, en anglais DBMS)

Chapter 4

Performances, fiabilité et robustesse

4.1 Tableau des Exigences

| ID | Name | Text |
|-----|---------------------------------|---|
| 1 | Fiabilité | |
| 1.1 | Réponse aux requêtes | Toute requête lancée à travers le système Vishnu conduit à une réponse. Cette réponse peut être soit le résultat prévu, soit un message d'erreur. Pour chaque message d'erreur, le système vishnu associe un code d'erreur (type entier) correspondant. Un code égal à zero correspondra à un résultat correct. Les interfaces de programmation de Vishnu fourniront un mécanisme d'exceptions permettant à l'utilisateur de traiter séparément les erreurs selon leur cause. |
| 1.2 | Enregistrement des requêtes | Le système Vishnu conserve la trace de chaque requête utilisateur dans une base de données. Cette trace comprend les identifiants de l'utilisateur et de la machine cliente, le nom de la commande ayant permis le lancement de la requête et les paramètres de la requête. Dans le cas d'une requête soumise via l'API Web Services, la machine cliente sera le serveur web lui-même. |
| 1.3 | Reproductibilité des requêtes | Le système Vishnu peut ré-exécuter une ou plusieurs requêtes à partir des traces enregistrées. Pour chaque requête, les paramètres de la requête seront les mêmes que lors de la soumission initiale à l'exception de l'utilisateur et de la machine cliente qui proviendront de la session courante (remarque: un administrateur peut ouvrir une session en tant qu'un autre utilisateur). |
| 1.4 | Erreurs dues à l'infrastructure | Lorsqu'une machine de l'infrastructure est arrêtée ou injoignable le système répond aux requêtes des utilisateurs concernant cette machine avec un message d'erreur indiquant que la machine n'est pas disponible. |
| 1.5 | Erreurs d'utilisation | Une requête lancée avec des paramètres incorrects sera rejetée par le système Vishnu, et pour chaque paramètre incorrect, le système Vishnu indiquera le motif du rejet. Le système Vishnu indiquera à l'utilisateur l'utilisation correcte de la requête avec les bons paramètres. |

| ID | Name | Text |
|-------|--|---|
| 1.6 | Erreurs dues à l'intergiciel | Lorsqu'un processus serveur fournissant un service de l'intergiciel Vishnu crashe pendant l'exécution de ce service le système Vishnu informe l'utilisateur de l'arrêt de ce service par un message d'erreur ou une exception. Dans le cas d'un appel de service asynchrone, après que la requête utilisateur a été lancée, un arrêt du service ne sera pas signalé à l'utilisateur. Si le service est redémarré avant une requête d'interrogation de l'utilisateur alors l'utilisateur recevra des informations sur l'état de ses requêtes sans être informé de l'arrêt temporaire du service. |
| 1.7 | Erreurs dues aux droits d'accès | L'accès à certaines informations nécessite des permissions en lecture ou écriture. Si un utilisateur n'a pas les droits nécessaires pour accéder à ces informations, le système Vishnu l'en informera par un message indiquant la nature de la permission manquante et l'objet concerné. |
| 2 | Capacité | |
| 2.1 | Nombre de requêtes simultanées | Le système pourra supporter au minimum 100 requêtes simultanées. |
| 2.2 | Nombre d'utilisateurs | Le système pourra supporter au minimum 1000 utilisateurs enregistrés. |
| 2.3 | Nombre de machines serveur | Le système pourra supporter au minimum 10 machines serveurs (supercalculateurs ou frontales de cluster) |
| 2.4 | Nombre de machines clientes | Le système pourra supporter au minimum 1000 machines clientes (postes scientifiques) |
| 3 | Performance | |
| 3.1 | Latence | En fonctionnement normal (ressources des machines non saturées), avec l'interface d'utilisation en ligne de commande, la latence sera inférieure ou égale à celle obtenue par une commande ssh ayant le même effet en termes de soumission de job ou transfert de fichiers. |
| 3.2 | Consommation de ressources | |
| 3.2.1 | Consommation mémoire moyenne des processus VISHNU | La consommation moyenne des processus VISHNU ne devra pas dépasser 500Mb de mémoire virtuelle |
| 3.2.2 | Consommation CPU moyenne des processus VISHNU | La consommation CPU des processus VISHNU ne devra pas dépasser 10% lorsque le système n'est pas soumis à des requêtes utilisateur (la consommation étant essentiellement due aux processus de supervision) |
| 3.2.3 | Consommation ponctuelle pour l'exécution d'une requête utilisateur | L'utilisation de Vishnu pour exécuter des requêtes sur un gestionnaire de jobs (soumission de job ou interrogation) ne devra pas consommer en moyenne davantage de ressources sur le serveur que l'utilisation d'un client en ligne de commande (client "lourd" du gestionnaire de jobs). |
| 4 | Robustesse | |
| 4.1 | Absence de plantage intempestif | Les processus du système ne doivent pas s'interrompre en fonctionnement "de croisière" (c'est-à-dire hors de toute défaillance logicielle ou matérielle du système hôte). |
| 4.2 | Résistance aux défaillances | Le système doit rester disponible en cas de défaillance de l'infrastructure n'affectant pas la connection de l'utilisateur aux point(s) d'entrée du système. Plusieurs points d'entrée pourront être configurés mais le nombre de processus lancés par le système sur les machines sera alors augmenté. |
| 4.3 | Délestage | Le délestage permettra à un administrateur de réduire de manière immédiate la charge générée par l'intergiciel sur une machine de l'infrastructure. Deux niveaux de délestage seront possible. |
| 4.3.1 | Délestage "soft" | Au niveau "soft", le délestage interrompra les activités en cours sur la machine concernée (activités contrôlées par le système Vishnu uniquement). Ces activités pourront être relancées par le système lors de la commande de fin de délestage exécutée par l'administrateur. |

| ID | Name | Text |
|-------|------------------|---|
| 4.3.2 | Délestage "hard" | Au niveau "hard", le délestage interrompra tous les processus de l'intergiciel sur la machine concernée sans effectuer d'actions éventuellement nécessaires à leur redémarrage. Le redémarrage de Vishnu sur la machine utilisera la même procédure qu'un démarrage "à froid" standard. |

4.2 Glossaire

- Aucun terme à définir