

Plan de tests IMS



INDEXATION DU DOCUMENT

	<i>TITRE :</i> Plan de tests IMS		
<i>ACTION</i>	<i>NOM</i>	<i>DATE</i>	<i>SIGNATURE</i>
RÉDIGÉ PAR	Benjamin Isnard, Ibrahima Cisse, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, et Benjamin Depardon	25 avril 2013	

SUIVI DU DOCUMENT

INDICE	DATE	MODIFICATIONS	NOM
1	06/04/11	Version initiale	B. Isnard, D. Traoré, K. Coulomb, I. Cisse E. Capo-Chichi
2	15/06/11	Mise à jour pour la livraison du module du IMS	B. Isnard, D. Traoré, K. Coulomb, I. Cisse E. Capo-Chichi
3	14/02/12	Ajout des cas de tests pour la génération d'identifiants concernant les systèmes d'authentification : IA3-B: Define the identifier auth et IA3-E1/E2: Define the identifier auth	B. Isnard, D. Traoré, K. Coulomb, I. Cisse E. Capo-Chichi
7	21/01/13	Passage au format docbook de base. Problèmes de compilation réglés.	B. Depardon

Table des matières

1	Références	1
2	Présentation du document	2
2.1	Objectifs du document	2
2.2	Structure du document	2
2.3	Glossaire, abréviations et acronymes	2
2.3.1	Glossaire	2
2.3.2	Acronymes	2
2.3.3	Abréviations	3
3	Environnements et outils de tests	4
3.1	Environnements de tests	4
3.2	Outils de tests	4
4	Cycle de vie du test	5
5	Liste des tests	6
5.1	Tests fonctionnels	6
5.1.1	Identifiant de tests	6
5.1.2	Tests de la Catégorie 1	6
5.1.3	Tests de la catégorie 2	7
5.1.4	Tests de la catégorie 3	9
5.1.5	Tests de catégorie 4	10
5.2	Tests de stress	15
5.2.1	Présentation	15
5.2.2	Environnement des tests	15
5.2.3	Identifiants de tests	15
5.2.4	Liste des tests	15
5.3	Contrôle des performances	16
5.3.1	Présentation	16
5.3.2	Environnement de tests	16

5.3.3	Identifiants des tests	16
5.3.4	Liste des tests	16
5.4	Tests de charge	16
5.4.1	Présentation	16
5.4.2	Environnement de tests	17
5.4.3	Identifiants de tests	17
5.4.4	Liste des tests	17
5.5	Contrôle de la qualité du code	17
5.5.1	Présentation	17
5.5.2	Identifiants des tests	17
5.5.3	Liste des tests	17
5.6	Tests de compilation et d'installation	18
5.6.1	Présentation	18
5.6.2	Environnement des tests	18
5.6.3	Liste des tests	18
5.7	Contrôle de la documentation utilisateur	18
5.7.1	Présentation	18
5.7.2	Liste des tests	18

Table des figures

4.1	: Schema du test	5
-----	----------------------------	---

Chapitre 1

Références

- VISHNU_D1_1e-POL-TESTS : Politique de tests du projet VISHNU
- VISHNU_D1_1a-SPECSGEN : Spécifications générales du projet VISHNU
- VISHNU_D1_1b-STB : Spécifications Techniques des besoins
- VISHNU_D1_1g-ARCH : VISHNU Technical Architecture

Chapitre 2

Présentation du document

2.1 Objectifs du document

Les objectifs de ce document sont les suivants :

- Présenter les fonctionnalités du module IMS à tester.
- Présenter les différents types de tests permettant de valider les cas d'utilisation de chaque fonctionnalité du module IMS.
- Présenter les méthodes et outils utilisés pour ces tests.
- Donner des détails de l'environnement de test

2.2 Structure du document

Le chapitre 1 présente le document.

Le chapitre 2 présente les environnements et outils de tests.

La chapitre 3 décrit la liste des tests qui s'appliquent aux fonctionnalités à tester.

La chapitre 4 présente les outils utilisés pour ces tests.

2.3 Glossaire, abréviations et acronymes

2.3.1 Glossaire

- **Script de test** : c'est un programme qui contient les instructions permettant de réaliser les tests.
- **Certificat** : correspond à la « *session key* » définie dans [SPECSEGEN].

2.3.2 Acronymes

- STB : Spécification Technique des Besoins
- DSG : Document de Spécification générale
- DSD : Document de Spécification Détaillées
- API : Application Programming Interface (Interface de programmation d'applications)
- MPI : Message Passing Interface

2.3.3 Abréviations

- Id. Identifiant
- Réf. Référence
- Rel.# Numéro de release/version
- Resp. Responsable(s)



Chapitre 3

Environnements et outils de tests

3.1 Environnements de tests

Pour ces tests, nous allons utiliser l'environnement de tests suivants avec des systèmes dédiés aux tests:

- Système d'exploitation : c'est le système Linux décrit dans [STB] qui sera installé sur toutes les machines clientes et serveurs.
- Nombre de machines clientes : une machine cliente décrite dans [STB] pour pouvoir lancer des tests à distance (selon le cas de test).
- Nombre de machines serveurs : deux serveurs IMS décrit dans [ARCH] .
- Un serveur SysFera-DS dédié décrit dans [ARCH].
- Un serveur UMS décrit dans [ARCH].
- Un serveur TMS et FMS décrit dans [ARCH]
- Le module LogService de SysFera-DS décrit dans [STB].

3.2 Outils de tests

Les outils suivants seront utilisés pour les tests :

- Hudson (serveur d'intégration) sera utilisé pour les tests d'intégration continue (compilation, installation).
- La librairie Boost test sera utilisée pour la réalisation des tests unitaires *en C++*.
- Le framework JUnit sera utilisé pour la réalisation des tests unitaires *en JAVA*.
- Des scripts Python seront utilisés pour la réalisation des tests unitaires *en Python*.
- Des scripts SHELL BASH seront utilisés pour la réalisation des tests unitaires *de la ligne de commande*.
- Squore et Logiscope seront utilisés pour la qualité du code.
- Valgrind sera utilisé pour la détection de fuite mémoire.

Chapitre 4

Cycle de vie du test

Le schéma suivant donne une vue du plan de test sur un test à réaliser jusqu'à la réalisation du rapport de test :

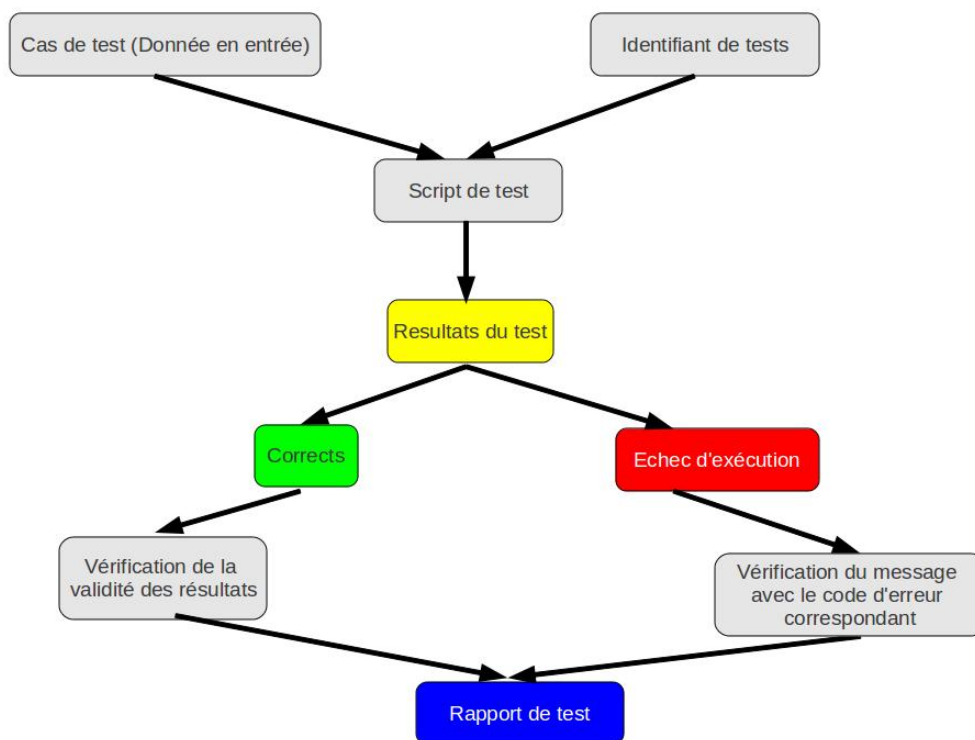


FIGURE 4.1 – : Schema du test

Chapitre 5

Liste des tests

5.1 Tests fonctionnels

Les tests fonctionnels ou tests de boîtes noires permettent à partir des données en entrée d'observer les réactions du logiciel à tester.

Les fonctionnalités sont classées par catégories de tests :

- Catégorie 1 : comprend la fonctionnalité permettant d'exporter toutes les commandes ayant été exécutées durant une session: *exportCommands*.
- Catégorie 2 : regroupe les fonctionnalités permettant d'agir physiquement sur un composant : *loadShed*, *stop*, *restart*.
- Catégorie 3 : comprend le service permettant de consulter l'ensemble des processus VISHNU : *getProcesses*.
- Catégorie 4 : regroupe les fonctionnalités permettant la consultation et modification des informations système : *getMetricCurrentValue*, *getMetricHistory*, *getSystemThreshold*, *setSystemThreshold*, *getUpdateFrequency*, *setUpdateFrequency*, *defineUserIdentifier*, *defineMachineIdentifier*, *setSystemInfo*, *getSystemInfo*, *defineJobIdentifier*, *defineTransferIdentifier*, *defineAuthIdentifier*.

Chaque test fonctionnel sera réalisé en utilisant chacune des interfaces d'accès suivantes : fonctions API/C++ et API/Python, ligne de commande utilisateur et API web services.

Nous allons dans cette section présenter les différentes stratégies de tests en utilisant des programmes exécutables ou des fonctions de l'API IMS puis en observant les résultats obtenus. Les deux conditions à vérifier sont :

- Est ce que le programme réalise les fonctions attendues en accord avec les spécifications?
- Est ce que le programme fonctionne correctement ?

5.1.1 Identifiant de tests

Les identifiants des tests sont de la forme: <ID DSG>-[BIE]:<NOM_TEST>-<INTERFACE> où 'ID DSG' est l'identifiant du cas d'utilisation dans le document [SDG] , 'B' la séquence de base (exécution du cas d'utilisation avec des paramètres valides), 'E' la séquence d'exception (comportement du cas d'utilisation en cas d'erreurs), 'NOM_TEST' le nom du test décrit dans [SDG] et 'INTERFACE' les différents codes des interfaces utilisées par VISHNU : fonctions API/C++(CPP), API/Python (PY), ligne de commande utilisateur (CLI) et API web services (WS). L'absence du code de l'interface signifie que le test s'applique à toutes les interfaces précédentes.

Il faut noter que les tests avec des paramètres corrects sont effectués dans une session active avec un mode de fermeture à la déconnexion (CLOSE_ON_DISCONNECT) et un certificat de session valide.

5.1.2 Tests de la Catégorie 1

Identifiant de tests	Actions	Résultats	Considérations particulières
I3-B: export and replay commands	<ul style="list-style-type: none"> 0. Cas de test : Les paramètres sont corrects 0. Ouvrir une session à l'aide de la commande <i>vishnu_connect</i> et enregistrer l'identifiant de la session. 0. Lancer trois commandes VISHNU. 0. Lancer la commande <i>vishnu_list_history_cmd</i> et enregistrer les commandes retournées. 0. Fermer la session 0. Ouvrir une nouvelle session 0. Lancer la commande <i>vishnu_export_commands</i> avec comme paramètre le nom du fichier dans lequel les commandes seront exportées sous le format SHELL et en utilisant l'identifiant de session de l'étape 2 0. Vérifier que le fichier contient les trois commandes lancées (ceci est valable si les commandes sont ni connect, ni changePassword, ni resetPassword de UMS) 	–	
I3-E1: export and replay commands I3-E2: export and replay commands I3-E3: export and replay commands	<ul style="list-style-type: none"> 0. Cas de test : paramètres sont incorrects 0. – Cas E1 : l'identifiant de la session est incorrect – Cas E2 : le chemin du fichier est incorrect 0. 0. Test du cas E1 et E2 : <ul style="list-style-type: none"> – Lancer la commande <i>vishnu_export_commands</i> – Vérifier que son exécution échoue 	–	

5.1.3 Tests de la catégorie 2

Identifiant de tests	Actions	Résultats	Considérations particulières
----------------------	---------	-----------	------------------------------

IA4.1-B : Hard load scheduling	<p>0. <i>Cas de test</i> : les paramètres sont corrects et l'utilisateur est un administrateur</p> <p>0. Si aucun serveur VISHNU n'a été lancé sur aucune machine se connecter sur les différentes machines et les lancer. Supposons les cas suivantes :</p> <ul style="list-style-type: none"> – Machine MA_1 : une machine sur laquelle sont lancés les serveurs suivants : <i>omniNames</i>, <i>dietAgent</i>, <i>LogCentral</i>, <i>sed UMS</i>, <i>sed TMS</i> et <i>sed IMS</i>. 0. – Machine MA_2 : une machine sur laquelle est lancé un <i>sed UMS</i> qui utilise les éléments suivants de MA_1: <i>omniNames</i>, <i>dietAgent</i> et le <i>LogCentral</i> . <p>0. Lancer la commande <i>vishnu_load_sched</i> avec comme option HARD et l'identifiant de la machine sur laquelle les processus seront stoppés (MA_2).</p> <p>0. Lancer la commande <i>vishnu_get_processes</i> et vérifier qu'aucun processus VISHNU n'est en cours d'exécution sur la machine cible.</p>	- Les processus en cours d'exécution sur la machine donnée ont été arrêtés et ne peuvent plus être redémarrés.	L'utilisateur doit être un administrateur
IA4.1-B : Soft load scheduling	<p>0. <i>Cas de test</i> : les paramètres sont corrects.</p> <p>0. Si aucun serveur VISHNU n'a été lancé sur aucune machine se connecter sur les différentes machines et les lancer. Supposons les cas suivantes :</p> <ul style="list-style-type: none"> 0. – Machine MA_1 : une machine sur laquelle sont lancés les serveurs suivants : <i>omniNames</i>, <i>dietAgent</i>, <i>LogCentral</i>, <i>sed UMS</i>, <i>sed TMS</i> et <i>sed IMS</i>. <p>0. Soumettre deux jobs à l'aide de la commande <i>vishnu_submit_job</i> sur la machine MA_1</p> <p>0. Lancer la commande <i>vishnu_load_sched</i> avec comme option SOFT et l'identifiant de la machine sur laquelle les processus seront stoppés.</p> <p>0. Lancer la commande <i>vishnu_list_jobs</i> et vérifier que le serveur TMS ne trouve pas les jobs précédents.</p>	- Les processus en cours d'exécution sur la machine donnée deviennent inactifs.	
Stop IA9 B	<p>0. Lancer la plateforme VISHNU avec un <i>sed IMS</i></p> <p>0. Appeler la fonction stop pour arrêter un process sur une machine</p> <p>0. Attendre 10 secondes et vérifier que le processus n'a pas été relancé et est arrêté</p>	–	administrateur

Stop IA9 E	<ul style="list-style-type: none"> 0. L'id de la machine est invalide, 0. Le processus n'existe pas sur la machine 0. L'appel est fait par un utilisateur est pas un administrateur 	<ul style="list-style-type: none"> – Une exception paramètre invalide est retournée – Une exception processus non valide est retournée – Une exception est renvoyée 	administrateur
Restart IA9 B	<ul style="list-style-type: none"> 0. Lancer une plateforme VISHNU avec un sed IMS 0. Arrêter (sans utiliser la fonction stop) le sed UMS 0. Vérifier que UMS tourne de nouveau, en faisant un vishnu_connect par exemple 	–	administrateur

5.1.4 Tests de la catégorie 3

Identifiant de tests	Actions	Résultats	Considérations particulières
IA1-B: Get the running processes	<ul style="list-style-type: none"> 0. <i>Cas de test</i> : Les paramètres sont corrects et l'utilisateur est un administrateur 0. Si aucun serveur VISHNU n'a été lancé sur aucune machine se connecter sur les différentes machines et les lancer. Supposons les cas suivantes : <ul style="list-style-type: none"> – Machine MA_1 : une machine sur laquelle sont lancés les seveurs suivants : <i>omniNames</i>, <i>dietAgent</i> et un <i>LogCentral</i>. 0. – Machine MA_2 : une machine sur laquelle sont lancés les seveurs suivants : un <i>sed UMS</i>, un <i>sed TMS</i> et un <i>sed IMS</i>. 0. Lancer la commande <i>vishnu_get_processes</i> avec les paramètres correspondants . 0. Vérifier que les processus VISHNU affichés correspondent bien aux processus VISHNU lancés sur les machines. 	–	L'utilisateur doit être un administrateur

IA1-E1: Get the running processes IA1-E2: Get the running processes	<p>0. Cas de test : paramètres incorrects</p> <p>0. – Cas E1 : l'identifiant de la machine est incorrect</p> <p>0. – Cas E2 : l'utilisateur n'est pas un administrateur</p> <p>0.</p> <p>0. Lancer la commande <i>vishnu_get_processes</i> avec le cas correspondant</p> <p>0. Vérifier que son exécution échoue avec le message d'erreur renvoyé.</p>	–	
--	---	---	--

5.1.5 Tests de catégorie 4

Identifiant de tests	Actions	Résultats	Considérations particulières
I2 – B: Get metric data	<p>0. Cas de test : les paramètres sont corrects, la fréquence de mise à jour de la base de données est fixée à 10 secondes et aucun enregistrement concernant la métrique <i>MEMORY</i> n'est dans la base de données.</p> <p>0. Après 20 secondes, lancer la commande <i>vishnu_get_metric_history</i> avec l'identifiant de la machine sur laquelle la métrique sera récupérée et la métrique <i>MEMORY</i> et vérifier que 2 enregistrements sont retournés.</p>	L'historique de la métrique est retourné.	la fréquence de mise à jour de la base de donnée doit être au préalable enregistrée dans la base de données et fixée à 10 secondes. Aucun enregistrement concernant la métrique <i>MEMORY</i> n'est enregistré dans la base de données.
I2 – E1: Get metric data I2 – E2: Get metric data	<p>0. Cas de test : les paramètres sont incorrects.</p> <p>– Cas E1 : l'identifiant de la machine n'est pas enregistré dans VISHNU</p> <p>0. – Cas E2 : le nom de la métrique n'est pas enregistré dans VISHNU</p> <p>0. Lancer la commande <i>vishnu_get_metric_history</i> avec le cas de test correspondant (E1 ou E2) et vérifier que son exécution échoue.</p>	–	

I4 – B: Get data on the infrastructure	<p>0. Cas de test : les paramètres sont corrects.</p> <p>0. Lancer la commande <i>vishnu_get_metric_current_value</i> avec l'identifiant de la machine sur laquelle la métrique sera récupérée et la valeur <i>ALL</i> pour la métrique et vérifier que les valeurs suivantes sont retournées: le pourcentage d'utilisation de <i>CPU</i>, le nombre de <i>CPU</i>, l'espace disque total, l'espace disque disponible, la mémoire RAM totale et l'espace RAM non utilisée .</p>	Les valeurs suivantes sont retournées :le pourcentage d'utilisation de <i>CPU</i> , le nombre de <i>CPU</i> , l'espace disque total, l'espace disque disponible, la mémoire RAM totale et l'espace RAM non utilisée.	
<p>I4 – E1: Get data on the infrastructure</p> <p>I4 – E2: Get data on the infrastructure</p>	<p>0. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> – Cas E1 : l'identifiant de la machine n'est pas enregistré dans VISHNU 0. – Cas E2 : le nom de la métrique n'est pas enregistré dans VISHNU <p>0. Lancer la commande <i>vishnu_get_metric_current_value</i> avec le cas de test correspondant (E1 ou E2) et vérifier que son exécution échoue.</p>	–	
IA2.1 – B : Get a system load threshold	<p>0. Cas de test : les paramètres sont corrects et la base de données a un seuil fixé à 2×10^6 pour l'espace disque disponible sur la machine MA_1.</p> <p>0. Lancer la commande <i>vishnu_get_system_threshold</i> sur MA_1 avec pour métrique <i>DISKSPACE</i> et vérifier que la valeur retournée est 2×10^6.</p> <p>0. Modifier le seuil pour l'espace disque disponible à 10^6 dans la base de données.</p> <p>0. Lancer la commande <i>vishnu_get_system_threshold</i> sur MA_1 avec pour métrique <i>DISKSPACE</i> et vérifier que la valeur retournée est 10^6.</p>	Le seuil correspondant est retourné.	L'utilisateur doit être un administrateur.
<p>IA2.1 – E1 : Get a system load threshold</p> <p>IA2.1 – E2 : Get a system load threshold</p> <p>IA2.1 – E3 : Get a system load threshold</p>	<p>0. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> – Cas E1 : l'identifiant de la machine n'est pas enregistré dans VISHNU – Cas E2 : le nom de la métrique n'est pas enregistré dans VISHNU 0. – Cas E3 : l'utilisateur n'est pas un administrateur <p>0. Lancer la commande <i>vishnu_get_system_threshold</i> avec le cas de test correspondant (E1, E2 ou E3) et vérifier que son exécution échoue.</p>	–	

<p>IA2 – B1 : Define a system load threshold IA2 – B2 : Define a system load threshold IA2 – B3 : Define a system load threshold</p>	<p>0. Cas de test : les paramètres sont corrects et les seuils concernant l'espace disponible, la mémoire RAM non utilisée et le pourcentage d'utilisation de la machine MA_1 sont différents des valeurs définies dans les cas qui suivent.</p> <p>0. Lancer la commande <i>vishnu_get_system_threshold</i> sur MA_1 avec les métriques <i>DISKPACE</i>, <i>FREEMORY</i> et <i>CPUUSE</i>.</p> <p>0. Lancer la commande <i>vishnu_set_system_threshold</i> sur MA_1 avec les métriques <i>DISKPACE</i>, <i>FREEMORY</i> et <i>CPUUSE</i> puis rajouter 10 à chacune des valeurs retournées à l'étape 2.</p> <p>0. Lancer la commande <i>vishnu_get_system_threshold</i> sur MA_1 avec les métriques <i>DISKPACE</i>, <i>FREEMORY</i> et <i>CPUUSE</i> et vérifier que les modifications précédents ont été prises en compte.</p>	<p>Le seuil correspondant est retourné.</p>	<p>L'utilisateur doit être un administrateur.</p>
<p>IA2 – E1 : Define a system load threshold IA2 – E2: Define a system load threshold IA2 – E3 : Define a system load threshold</p>	<p>0. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> – Cas E1 : l'identifiant de la machine n'est pas enregistré dans VISHNU – Cas E2 : le nom de la métrique n'est pas enregistré dans VISHNU <p>0. – Cas E3 : l'utilisateur n'est pas un administrateur</p> <p>0. Lancer la commande <i>vishnu_set_system_threshold</i> avec le cas de test correspondant (E1, E2 ou E3) et vérifier que son exécution échoue.</p>	<p>–</p>	
<p>I1– B : Get the update frequency</p>	<p>0. Cas de test : les paramètres sont corrects et la fréquence de mise à jour de la base de donnée est fixée à 10 secondes.</p> <p>0. Lancer la commande <i>vishnu_get_update_frequency</i> et vérifier que la valeur retournée est 10.</p>	<p>–</p>	<p>la fréquence de mise à jour de la base de donnée doit être au préalable enregistrée dans la base de données et fixée à 10 secondes.</p>

IA6– B : Set the update frequency	<p>0. Cas de test : les paramètres sont corrects.</p> <p>0. Lancer la commande <i>vishnu_get_update_frequency</i> et enregistrer la valeur retournée.</p> <p>0. Lancer la commande <i>vishnu_set_update_frequency</i> en ajoutant 10 à la valeur retournée à l'étape 2.</p> <p>0. Lancer la commande <i>vishnu_get_update_frequency</i> et vérifier que la modification de l'étape 2 a été bien prise en compte.</p>	–	L'utilisateur doit être un administrateur. La valeur de la fréquence de mise à jour avant le lancement du test doit être différente de 10 secondes.
IA6– E1 : Set the update frequency IA6– E2 : Set the update frequency	<p>0. Cas de test : les paramètres sont incorrects.</p> <p>0. – Cas E1 : la valeur de la fréquence de mise à jour de la base de données est égale à zéro</p> <p>0. – Cas E2 : la valeur de la fréquence de mise à jour de la base de données est négative</p> <p>0. Lancer la commande <i>vishnu_set_update_frequency</i> avec le cas de test correspondant (E1 ou E2) et vérifier que son exécution échoue.</p>		
IA3-B: Define the identifier : define User/Machine/Job/-FileTransfer/Auth identifier	<p>0. – Cas 1 : Définir le format d'identifiant utilisateur</p> <p>0. – Cas 2 : Définir le format d'identifiant machine.</p> <p>0. – Cas 3 : Définir le format d'identifiant job.</p> <p>0. – Cas 4 : Définir le format d'identifiant de transfert de fichier.</p> <p>0. Lancer les différentes commandes <i>vishnu_define_user_identifier</i>, <i>vishnu_defineMachineIdentifier</i>, <i>vishnu_define_job_identifier</i>, <i>vishnu_define_transfer_identifier</i>, <i>vishnu_define_auth_identifier</i> avec les formats définis respectivement dans cas de test correspondant (cas1, cas2, cas3 et cas4) .</p> <p>0. Lancer un certain nombre de fois les commandes d'ajout suivantes <i>vishnu_addUser</i>, <i>vishnu_submitJob</i>, <i>vishnu_addMachine</i>, <i>vishnu_copyFile</i> et vérifier que les identifiants générés correspondent bien aux formats attendus.</p> <p>0. L'exécution de la commande est ainsi validée.</p>	Le format d'identifiant demandé est généré.	Ces différentes commandes sont réservées aux administrateurs VISHNU

IA3-E1: Define User/Machine/Job/-FileTransfer/Auth identifier IA3-E2: Define User/Machine/Job/-FileTransfer/Auth identifier	<ul style="list-style-type: none"> 0. – Cas E1 : Le format fourni comprend une variable inconnue du système VISHNU 0. – Cas E2 : La variable représentant le compteur n'apparaît pas dans le format fourni. – Cas E3 : L'utilisateur n'est pas un administrateur 0. Lancer les différentes commandes avec le cas de test correspondant (E1 ou E2 ou E3) et vérifier que son exécution échoue. 	–	
IA5: Set system info	<ul style="list-style-type: none"> 0. Cas de test : les paramètres sont corrects. Mise à jour des données système de la machine MA_1 enregistré dans VISHNU. 0. Lancer la commande <i>vishnu_setSystemInfo</i> en lui fournissant MA_1 et les informations à mettre à jour. 0. Lancer la commande <i>vishnu_getSystemInfo</i> avec comme paramètre MA_1 et vérifier que les données ont bien été mise à jour. 0. L'exécution de la commande est ainsi validée. 	– 0. 0.	Cette commande est réservée aux administrateurs VISHNU
IA5-E1: set system info IA5-E2: set system info	<ul style="list-style-type: none"> 0. Cas de test : les paramètres sont incorrects. 0. – Cas E1 : L'identifiant machine fourni n'est pas enregistré dans VISHNU 0. – Cas E2 : Les informations systèmes fournies sont incorrectes. 0. Lancer la commande <i>vishnu_setSystemInfo</i> avec le cas de test correspondant (E1 ou E2) et vérifier que son exécution échoue. 	–	
I5-B: Get system info	<ul style="list-style-type: none"> 0. Cas de test : les paramètres sont corrects et les valeurs de mémoire et d'espace disque pour la machine MA_1 sont respectivement 256 et 10⁶. 0. Lancer la commande <i>vishnu_getSystemInfo</i> en lui fournissant MA_1. 0. Vérifier que les valeurs retournées pour la mémoire et l'espace disque de la machine MA_1 sont respectivement 256 et 10⁶. 	– 0. 0.	Avant le lancement du test les valeurs de mémoire et d'espace disque pour la machine MA_1 sont respectivement 256 et 10 ⁶ .

I5-E1: Get system info	<p>0. Cas de test : les paramètres sont incorrects.</p> <p>0. – Cas E1 : L'identifiant machine fourni n'est pas enregistré dans VISHNU</p> <p>0.</p> <p>0. Lancer la commande <i>vishnu_getSystemInfo</i> avec le cas de test E1 et vérifier que son exécution échoue.</p>	–	
------------------------	---	---	--

5.2 Tests de stress

5.2.1 Présentation

Ces tests permettent de vérifier que la cible de test peut réussir à basculer et à récupérer pour différents dysfonctionnements matériels, logiciels ou réseaux sans qu'il y ait perte de données. Ils vérifient que lorsqu'une défaillance survient un système alternatif ou un système de sauvegarde prends le relai correctement sans perte de données ou de transactions.

5.2.2 Environnement des tests

Nous utiliserons l'environnement de test 1.

5.2.3 Identifiants de tests

Les identifiants des tests sont de la forme: STR-<Element stressé>-<Type de stress>.

5.2.4 Liste des tests

Identifiant de tests	Technique	Critère de complétion
STR-SERV-DOWN	<ul style="list-style-type: none"> - Lancer la commande <i>vishnu_set_update_frequency</i> avec une valeur de fréquence de 30 secondes - Arrêter le serveur IMS après l'exécution de la commande - Redémarrer le serveur IMS - Lancer la commande <i>vishnu_get_update_frequency</i> 	<ul style="list-style-type: none"> - Terminaison normale de la commande <i>vishnu_get_update_frequency</i> avec la valeur 30.
STR-AGENT-DOWN	<ul style="list-style-type: none"> - Lancer la commande <i>vishnu_set_update_frequency</i> avec une valeur de fréquence de 30 secondes - Arrêter l'agent SysFera-DS après l'exécution de la commande - Redémarrer l'agent SysFera-DS - Lancer la commande <i>vishnu_get_update_frequency</i> 	<ul style="list-style-type: none"> - Terminaison normale de la commande <i>vishnu_get_update_frequency</i> avec la valeur 30.
STR-DB-DOWN	<ul style="list-style-type: none"> - Se connecter - Couper la base de données - Redémarrer la base de données - Faire appel à <i>vishnu_get_update_frequency</i> 	<ul style="list-style-type: none"> - Terminaison normale de la commande

5.3 Contrôle des performances

5.3.1 Présentation

Ces tests ont pour objectif de vérifier que les exigences de performance quantifiées sont effectivement satisfaites.

5.3.2 Environnement de tests

Nous utiliserons l'environnement de test 1.

5.3.3 Identifiants des tests

Les identifiants des tests sont de la forme: CDP-<ID STB>-[NomdeCommandeIdentifiantDeTest] où 'ID STB' est l'identifiant de l'exigence spécifiée dans le document [STB] au paragraphe « performances, fiabilité et robustesse ».

5.3.4 Liste des tests

Identifiant de tests	Technique	Critère de completion
CDP-3.1- <i>get_metric_current_value</i>	- Lancer la commande 'ssh <host> ps -efl' sur une machine serveur en mesurant le temps de réponse (temps de référence) - Lancer la commande 'vishnu_get_current_metric <machineId>' en mesurant le temps de réponse	temps inférieur au temps de référence (valeur moyenne sur 10 itérations)
CDP-3.1- <i>get_metric_history</i>	- Lancer une commande 'ssh <host> ps -efl' sur une machine serveur en mesurant le temps de réponse (temps de référence). - Lancer la commande 'vishnu_get_metric_history <machineId>' en mesurant le temps de réponse	temps inférieur au temps de référence (valeur moyenne sur 10 itérations)
CDP-3.2.1- <i>get_metric_current_value</i>	- Lancer la commande 'vishnu_get_metric_current_value <machineId>' 100 fois	Consommation en mémoire virtuelle sur le serveur IMS inférieure à 500 Mégaoctets.
CDP-3.2.1- <i>get_metric_history</i>	- Lancer la commande 'vishnu_get_metric_history <machineId>' 100 fois	Consommation en mémoire virtuelle sur le serveur IMS inférieure à 500 Mégaoctets.
CDP-3.2.2- <i>get_metric_current_value</i>	- Lancer la commande 'vishnu_get_metric_current_value <machineId>'	Consommation CPU inférieure à 10%.
CDP-3.2.2- <i>get_metric_history</i>	- Lancer la commande 'vishnu_get_metric_history <machineId>'	Consommation CPU inférieure à 10%.
CDP-MemServer	- Lancer le serveur IMS - Laisser le serveur fonctionner sans utiliser les services pendant 24h - Vérifier que l'espace mémoire consommé par le processus serveur n'a pas augmenté.	Consommation mémoire constante.

5.4 Tests de charge

5.4.1 Présentation

Les tests de charge ont pour objectif de vérifier que les exigences de capacité du système sont vérifiées.

5.4.2 Environnement de tests

Nous utiliserons l'environnement de test 1 avec un nombre de clients ou de serveurs tels que spécifié dans le cas de tests. Ces clients ou serveurs seront simulés sur une seule machine.

Par ailleurs, les requêtes envoyées en simultané seront en fait envoyées par une seule machine cliente avec un programme de synchronisation (MPI).

5.4.3 Identifiants de tests

Les identifiants des tests sont de la forme: LOAD-<ID STB>-[NomdeCommandeIdentifiantDeTest] où 'ID STB' est l'identifiant de l'exigence spécifiée dans le document [STB].

5.4.4 Liste des tests

Identifiant de tests	Technique	Résultats attendus
LOAD-2.1- <i>get_metric_current_value</i>	- Lancer 100 commandes 'vishnu_get_metric_current_value <machineId>' simultanées	- Terminaison normale de toutes les commandes - Etat normal du serveur IMS
LOAD-2.1- <i>get_metric_history</i>	- Lancer 100 commandes 'vishnu_get_metric_history <machineId>' simultanées	- Terminaison normale de toutes les commandes - Etat normal du serveur IMS

5.5 Contrôle de la qualité du code

5.5.1 Présentation

L'objectif de ces tests est de vérifier que le code source (C/C++ ou Java) respecte les contraintes de qualité logiciel fixées dans les spécifications et le cahier des charges.

5.5.2 Identifiants des tests

Les identifiants des tests sont de la forme: [IdentifiantDeTest]-[IdentifiantDesModules] où les identifiants sont les suivants:

- Identifiants de tests
 - CRP: Conformité aux Règles de Programmation
 - MCC: Mesure de la Complexité du Code
 - DDC: Détection de Duplication du Code
- Identifiants de modules
 - CSC : C/C++ source code
 - JSC : Java source code

5.5.3 Liste des tests

Identifiant de tests	Technique	Résultats attendus
CRP-CSC	Audit du code C/C++ avec l'outil LOGISCOPE	Le code respecte toutes les règles de codage de catégorie «REQUIRED» spécifiées pour le projet [SCL] sauf dérogation clairement documentée dans le code.
CRP-JSC	Audit du code Java avec l'outil LOGISCOPE	Le code respecte toutes les règles de codage de catégorie «REQUIRED» spécifiées pour le projet [SCL] sauf dérogation clairement documentée dans le code.

MCC-CSC	Audit du code C/C++ avec l'outil SQUORE	Pas plus de 1% de composants classés « A risque »
MCC-JSC	Audit du code Java avec l'outil SQUORE	Pas plus de 1% de composants classés « A risque »
DDC-CSC	Audit du code C/C++ avec l'outil SQUORE	Moins de 10% de code dupliqué.
DDC-JSC	Audit du code Java avec l'outil SQUORE	Moins de 10% de code dupliqué.

5.6 Tests de compilation et d'installation

5.6.1 Présentation

La compilation des différents exécutables et bibliothèques composant le module UMS sera réalisée par un serveur d'intégration continue (du type HUDSON).

Par ailleurs les tests d'installation ont pour objectif de valider la procédure d'installation de l'ensemble des différents composants d'une plateforme VISHNU (clients, serveurs, agents SysFera-DS).

5.6.2 Environnement des tests

Nous utiliserons l'environnement de test 1.

5.6.3 Liste des tests

Identifiant de tests	Technique	Résultats attendus
COMP-CSC	Compilation du code C/C++ avec l'outil GCC/CMake	Compilation correcte de tous les exécutables
INS-SERVER-PGSQL INS-SERVER-ORACLE	Installation des différents éléments serveur de la plateforme VISHNU – serveur de calcul – serveur de stockage – serveur dédié VISHNU – serveur de base de données	- Tous les processus démarrés et opérationnels - Documentation disponible (pages man)
INS-CLI	Installation du client VISHNU en ligne de commande	- Toutes les commandes installées - Documentation disponible (pages man)
INS-PGAPI	Installation des bibliothèques API VISHNU (C++ et Python)	- Bibliothèque C++ installée - Bibliothèque Python installée et pouvant être chargée dans l'interpréteur Python 2.x
INS-WSAPI	Installation des services web VISHNU sur le serveur JBoss	- Services opérationnels (contrôle via la console webservices de JBoss)

5.7 Contrôle de la documentation utilisateur

5.7.1 Présentation

L'objectif est de vérifier que la documentation utilisateur permet à l'utilisateur de comprendre et d'utiliser les fonctions spécifiées.

5.7.2 Liste des tests

Identifiant de tests	Technique	Résultats attendus
CDU-MAN	Vérifier les pages de manuel attachées à chaque commande	Toutes les pages de manuel contiennent toutes les informations nécessaires à l'utilisateur pour utiliser les commandes.
CDU-USERGUIDE	Suivre le guide de l'utilisateur	Le guide permet une prise en main rapide des fonctions du logiciel