



PROJET VISHNU
DOCUMENTS DE REFERENCE

PLAN DE TESTS

Ref. document VISHNU_D4_1b-TMS-PlanTests
Objet du document Référentiel du plan de tests du module TMS du projet Vishnu

TABLE DES REVISIONS

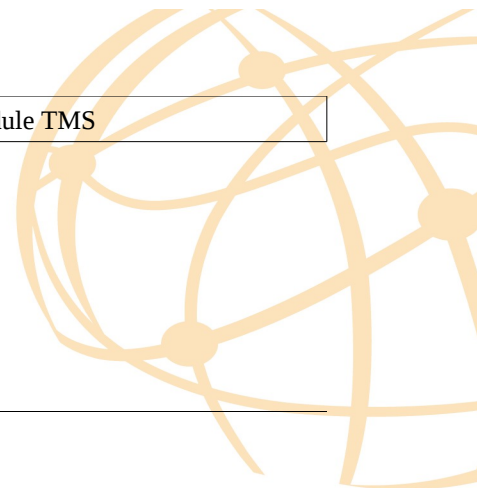
| Version | Date | Modifications | Auteur |
|---------|----------|------------------|--|
| V1.0 | 21/03/11 | Version initiale | Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi |

VALIDATION DU DOCUMENT

| Date | Personne |
|------|----------|
| | |
| | |

REFERENCES

| Ref. document | Description |
|------------------------|---|
| 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_1a-VTMSPD | VISHNU Task Management Service Package Design |



1 Présentation du document

1.1 Objectifs du document

Les objectifs de ce document sont les suivants :

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

1.2 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.

1.3 Glossaire, abréviations et acronymes

1.3.1 Glossaire

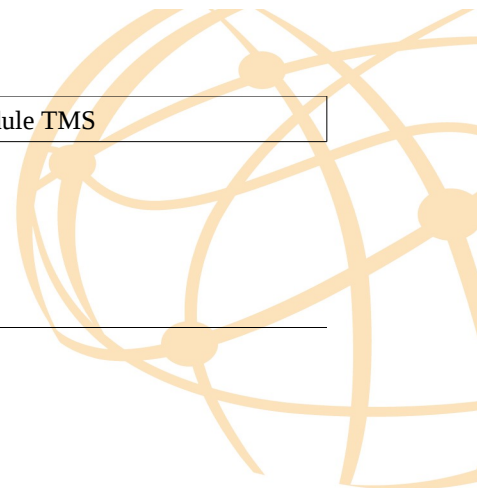
- **Script de test** : c'est un programme qui contient les instructions permettant de réaliser les tests.
- **Certificat** : correspond à la clé de session (identifiant de session crypté)

1.3.2 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
- Batch Scheduler : outil logiciel effectuant la gestion des ressources informatiques dans des supercalculateurs ou dans des grappes de serveurs informatiques. Ces outils permettent aux utilisateurs d'effectuer des réservations de ces ressources pour un temps donné.

1.3.3 Abréviations

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



2 Environnements et outils de tests

2.1 Environnements de tests

Pour ces tests nous allons utiliser l'environnement Lsuivant :

- Système d'exploitation : c'est le système Linux qui sera installé sur toutes les machines clientes et serveurs.
- Nombre de machines clientes : au moins 1 machine cliente décrite dans [STB] pour pouvoir lancer des tests à distance (selon le cas de test).
- Un serveur TMS par Batch Scheduler [VTMSPD] (Torque ou LoadLeveler).
- Un serveur SysFera-DS dédié décrit dans [STB].

2.2 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).
- Boost test sera utilisé pour la réalisation des tests unitaires.
- Squore et Logiscope seront utilisés régulièrement pour la qualité du code.
- Valgrind sera utilisé pour la détection de fuite mémoire.

3 Cycle de vie du test

Le schéma suivant donne une vue d'ensemble du plan de test sur des cas de tests à réaliser jusqu'à la rédaction du rapport de tests :

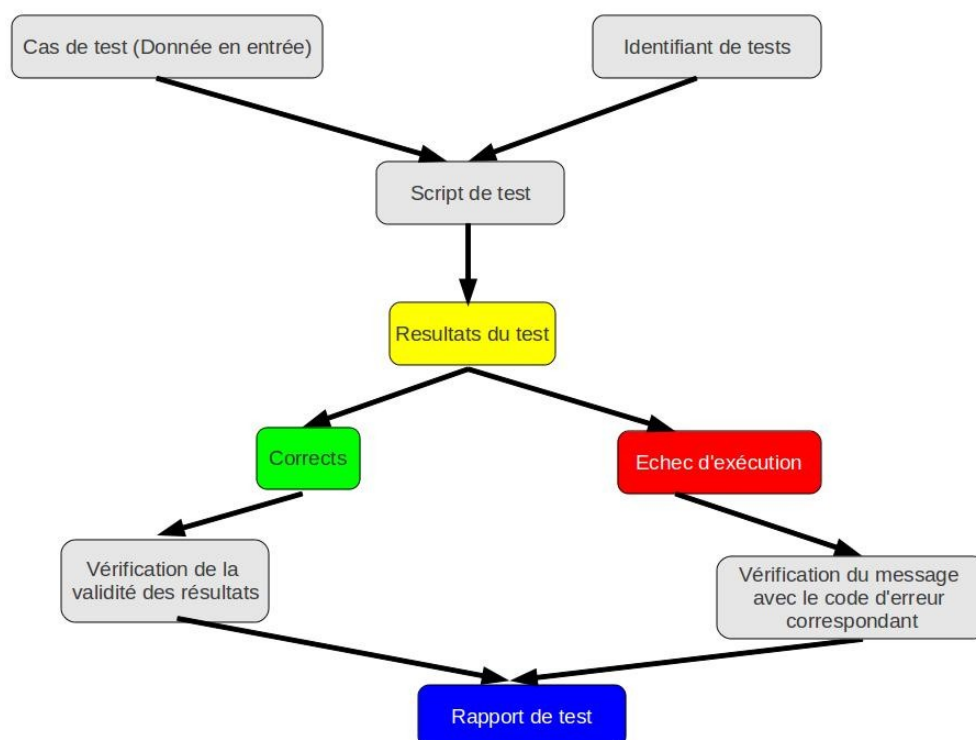


Illustration 1: Schema du test

4 Liste des tests

4.1 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 (*fonctions API/C++ et API/Python, ligne de commande utilisateur, API web service*) sont classées par catégories de tests. A chaque catégorie correspond un ensemble de fonctionnalités. Les catégories de tests sont les suivantes :

- Catégorie 1 : regroupe les fonctionnalités permettant de soumettre une action au serveur TMS: *vishnu_submit_job, vishnu_cancel_job, vishnu_set_machine_refresh_period.*
- Catégorie 2 : regroupe les fonctionnalités permettant de solliciter des informations du serveur TMS: *vishnu_get_job_info, vishnu_get_job_progress, vishnu_get_job_out_put, vishnu_get_all_jobs_out_put, vishnu_get_machine_refresh_period, vishnu_list_queues, vishnu_list_jobs.*

Nous allons dans cette section présenter les différentes stratégies de tests en utilisant des programmes exécutables ou des fonctions de l'API TMS 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 ?

4.1.1 Identifiant de tests

Les identifiants des tests sont de la forme: <ID DSG>-[B|E] 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) et 'E' la séquence d'exécution (comportement du cas d'utilisation en cas d'erreurs).

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 valide.

4.1.2 Tests de la Catégorie 1

| Identifiant de tests | Actions | Résultats | Considérations particulières |
|--|--|--|--|
| T1.1-B:submit a job | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Lancer la commande vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis et le chemin vers le script contenant les caractéristiques du job. Si l'identifiant de machine et le script sont valides, un identifiant du job généré par VISHNU et le chemin jusqu'au résultats du job sont retournés. Lancer la commande vishnu_list_jobs et vérifier que l'identifiant du job est bien présent dans la liste avec la bonne date de soumission. | - Un job est soumis au Batch scheduler et un identifiant de job est généré par VISHNU. | Le script soumis doit respecter les directives du Batch scheduler utilisé. |
| T1.1-E1:submit a job T1.1-E2 : submit a job T1.1-E3 : submit a job T1.1-E4 : submit a job T1.1-E5 : submit a job T1.1-E6 : submit a job | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : les options du script sont incorrects Cas E5 : le chemin jusqu'au script est incorrect Cas E6 : le Batch scheduler est indisponible Lancer la commande vishnu_submit_job avec le cas de test (E1, E2, E3, E4, E5 ou E6) et vérifier que son exécution échoue avec le message correspondant. | - Un message correspondant au cas d'erreur est affiché. | |
| T2.2-B:cancel a job | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Lancer la commande vishnu_cancel_job avec l'identifiant de la machine sur laquelle le job sera annulé et l'identifiant du job à annuler. Lancer la commande vishnu_get_job_info avec les paramètres précédents (l'identifiant de la machine sur laquelle le job a été annulé et l'identifiant du job annulé) et vérifier que son exécution échoue. | Le job est annulé. | |

| | | | |
|--|--|--|---|
| T2.2-E1:cancel a job T2.2-E2:cancel a job T2.2-E3:cancel a job T2.2-E4:cancel a job T2.2-E5:cancel a job T2.2-E6:cancel a job | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : l'identifiant de job n'est pas enregistré dans VISHNU Cas E5 : la commande est lancée par un utilisateur autre que celui qui a soumis le job et en plus qui n'est pas administrateur. Cas E6 : le Batch scheduler est indisponible Lancer la commande vishnu_cancel_job avec le cas de test (E1, E2, E3, E4, E5 ou E6) et vérifier que son exécution échoue avec le message d'erreur correspondant. | 1.- Le message d'erreur correspondant au cas de test est affiché | |
| TA1-B:set machine refresh period | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects et l'utilisateur est un administrateur du système VISHNU. Lancer la commande vishnu_get_machine_refresh_period avec l'identifiant de la machine dont l'administrateur souhaite obtenir la période de rafraichissement. Lancer la commande vishnu_set_machine_refresh_period avec l'identifiant de la machine sur laquelle la période de rafraichissement sera appliquée et le nouveau délai de rafraichissement. Faire vishnu_get_machine_refresh_period avec l'identifiant de la machine dont la période a été modifiée et vérifier que le nouveau délai de rafraichissement a été bien pris en compte. | | L'utilisateur qui exécute les commandes doit être un administrateur du système VISHNU |
| TA1-E1:set machine refresh period TA1-E2:set machine refresh period TA1-E3:set machine refresh period TA1-E4:set machine refresh period TA1-E5:set machine refresh period | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : le délai de rafraichissement est négatif Cas E5 : l'utilisateur n'est pas un administrateur Lancer vishnu_set_machine_refresh_period avec le cas de test (E1, E2, E3, E4 ou E5) et vérifier que son exécution échoue avec le message d'erreur correspondant. | - Un message correspondant au cas d'erreur est affiché | |

4.1.3 Tests de la catégorie 2

| Identifiant de tests | Actions | Résultats | Considérations particulières |
|--|--|---|------------------------------|
| T2.1-B:get job information | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Lancer la commande vishnu_list_jobs avec l'option -i suivie de l'identifiant du job généré par VISHNU et l'identifiant de la machine sur laquelle le job a été soumis. Lancer la commande vishnu_get_job_info avec l'identifiant de la machine sur laquelle le job a été soumis et l'identifiant du job généré par VISHNU. Comparer les résultats des commandes précédentes et vérifier qu'ils sont identiques. | - les informations correspondant au job sont retournés. | |
| T2.1-E1:get job information T2.1-E2:get job information T2.1-E3:get job information T2.1-E4:get job information T2.1-E5:get job information | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : l'identifiant du job n'est pas enregistré dans VISHNU Cas E5 : le Batch scheduler est indisponible Lancer le commande vishnu_get_job_info avec le cas de test (E1, E2, E3, E4 ou E5) et vérifier que son exécution échoue avec le message d'erreur correspondant. | - Un message d'erreur correspondant au cas d'erreur est affiché | |
| T2.5-B1:get jobs progression T2.5-B2:get jobs progression | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. <ul style="list-style-type: none"> Cas B1 - get jobs progression : la commande vishnu_get_job_progress est utilisée avec l'identifiant de la machine sur laquelle le job a été soumis Cas B2 - get jobs progression : la commande vishnu_get_job_progress est utilisée avec l'option -i suivie de l'identifiant du job généré par VISHNU et l'identifiant de la machine sur laquelle le job a été soumis. Vérifier qu'aucun job n'est soumis ou en cours d'exécution sinon les annuler à l'aide de vishnu_cancel_job. Lancer successivement vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis et le chemin vers le script contenant les caractéristiques du job puis conserver les identifiants de job retournés par le système. | - La progression du job est retournée par le système | |

| | | | |
|---|---|---|--|
| | <p>Pour le cas B1 :</p> <ol style="list-style-type: none"> Lancer la commande vishnu_get_job_progress avec le même identifiant de machine que précédemment. Vérifier que les progressions des deux jobs lancés précédemment sont retournées par VISHNU. Répéter la commande précédente jusqu'à ce que la progression soit de 100% pour chacun des jobs Faire vishnu_get_job_out_put avec l'identifiant de la machine sur laquelle les deux jobs ont été soumis et l'identifiant de chaque job et vérifier que le résultat de chacun des jobs est bien retourné. <p>Pour le cas B2 :</p> <ol style="list-style-type: none"> Lancer la commande vishnu_get_job_progress avec l'option -i suivie d'un identifiant de job généré par VISHNU et l'identifiant de la machine sur laquelle le job a été soumis. Répéter la commande précédente jusqu'à ce que la progression du job soit de 100% . Faire vishnu_get_job_out_put avec l'identifiant de la machine sur laquelle le job a été soumis et l'identifiant du job et vérifier que le résultat job est bien retourné. | | |
| <p>T2.5-E1:get jobs progression T2.5-E2:get jobs progression T2.5-E3:get jobs progression T2.5-E4:get jobs progression</p> | <p>1. Cas de test : les paramètres sont incorrects</p> <p>Pour le cas B1 et B2:</p> <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>Pour le cas B2 :</p> <ul style="list-style-type: none"> Cas E4 : l'identifiant du job n'est pas enregistré dans VISHNU <p>2. Lancer vishnu_get_job_progress avec les paramètres correspondant à chaque cas et vérifier que l'exécution de la commande échoue avec le message d'erreur correspondant.</p> | - Un message correspondant au code d'erreur est affiché | |
| <p>T2.6-B:get job outputs synchronously</p> | <p>1. Cas de test : les paramètres sont corrects.</p> <p>2. Lancer vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis et le chemin vers le script contenant les caractéristiques du job et conserver l'identifiant du job retourné par le système.</p> <p>3. Lancer la commande vishnu_get_job_progress avec l'option -i suivie de l'identifiant du job et de la machine précédents.</p> <p>4. Répéter la commande précédente jusqu'à ce que la progression du job soit de 100%.</p> <p>5. Faire vishnu_get_job_out_put avec l'identifiant de la machine sur laquelle le job a été soumis et l'identifiant du job et vérifier</p> | - Les résultats de job sont récupérés. | |

| | | | |
|--|---|---|--|
| | que le résultat job est bien retourné. | | |
| T2.6-E1: get job outputs synchronously T2.6-E2: get job outputs synchronously T2.6-E3: get job outputs synchronously T2.6-E4: get job outputs synchronously | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : le Batch scheduler est indisponible La progression du job n'est pas de 100% Lancer le programme vishnu_get_job_out_put avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message d'erreur correspondant. | - Un message correspondant au cas d'erreur est affiché | |
| T1.2-B : get job outputs asynchronously | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Vérifier qu'aucun job n'est soumis ou en cours d'exécution sinon les annuler à l'aide de vishnu_cancel_job. Lancer successivement vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis et le chemin vers le script contenant les caractéristiques du job dont le outputPath et le errorPath puis conserver les deux identifiants de job retournés par le système et les chemins outputPath et le errorPath enregistrés dans le script pour chacun des jobs. Répéter la commande vishnu_get_job_progress avec l'identifiant de la machine précédente jusqu'à ce que la progression des deux jobs soumis soit de 100%. Lancer la commande vishnu_get_all_jobs_out_put et vérifier que les résultats des jobs sont bien rapatriés | - Tous les jobs soumis sont rapatriés. | |
| T1.2-E1 : get job outputs asynchronously T1.2-E2 : get job outputs asynchronously T1.2-E3 : get job outputs asynchronously T1.2-E4 : get job outputs asynchronously | <ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects <ul style="list-style-type: none"> Cas E1 : le serveur TMS est indisponible Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E4 : le Batch scheduler est indisponible Lancer la commande vishnu_get_all_jobs_out_put avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message d'erreur correspondant. | - Le message d'erreur correspondant au cas d'erreur est affiché | |
| T2.7-B: get machine refresh period | <ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Lancer la commande vishnu_set_machine_refresh_period avec l'identifiant de la machine sur laquelle la période de rafraichissement sera appliquée et le nouveau délai de rafraichissement. Lancer la commande vishnu_get_machine_refresh_period | - Le délai de rafraichissement est retourné. | |

| | | | |
|---|---|--|--|
| | <p>avec l'identifiant de la machine utilisée précédemment et vérifier que la valeur est bien celle qui a été attribuée.</p> <p>4. Répéter 1 avec un délai de rafraichissement différent de celui qui précède puis faire 2.</p> | | |
| <p>T2.7-E1: get machine refresh period</p> <p>T2.7-E2: get machine refresh period</p> <p>T2.7-E3: get machine refresh period</p> | <p>1. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : le serveur TMS est indisponible ● Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>2. Lancer la commande vishnu_get_machine_refresh_period avec le cas de test (E1, E2 ou E3) et vérifier que son exécution échoue avec le message d'erreur correspondant.</p> | - Un message correspondant au cas d'erreur est affiché | |
| T2.3-B: list job queues | <p>1. Cas de test : les paramètres sont corrects.</p> <p>2. Lancer la commande vishnu_list_queues avec l'identifiant de la machine rattachée au Batch scheduler dont les queues seront listées et conserver la liste des queues.</p> <p>3. Se connecter au Batch scheduler associé à la machine précédente.</p> <p>Cas 1 : le Batch scheduler est Torque</p> <ul style="list-style-type: none"> - Lancer la commande qstat -Q et vérifier que la liste des queues obtenue est la même que celle qui précède. <p>Cas 2 : le Batch scheduler est LoadLeveler</p> <ul style="list-style-type: none"> - Lancer la commande llclass et vérifier que la liste des queues obtenue est la même que celle qui précède. | - Les queues du Batch scheduler sont listées. | |
| <p>T2.3-E1: list job queues</p> <p>T2.3-E2: list job queues</p> <p>T2.3-E3: list job queues</p> <p>T2.3-E4: list job queues</p> | <p>1. Cas de test : paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : le serveur TMS est indisponible ● Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E4 : le Batch scheduler est indisponible <p>2. Lancer la commande vishnu_list_queues avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message d'erreur correspondant.</p> | - Un message correspondant au cas d'erreur est affiché | |
| T2.4-B: list jobs | <p>1. Cas de test : les paramètres sont corrects.</p> <p>2. Vérifier qu'aucun job n'est soumis ou en cours d'exécution sinon les annuler à l'aide de vishnu_cancel_job.</p> <p>3. Lancer cinq fois vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis et le chemin vers le script contenant les caractéristiques du job puis conserver les</p> | Les jobs soumis sont affichés. | |

| | | | |
|--|---|--|--|
| | identifiants des jobs. 4. Lancer la commande <i>vishnu_list_jobs</i> et vérifier que les cinq jobs précédents sont affichés. | | |
| T2.4-E1:list jobs T2.4-E2:list jobs T2.4-E3:list jobs T2.4-E4:list jobs | 1. Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> ● Cas E1 : le serveur TMS est indisponible ● Cas E2 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E3 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E4 : le Batch scheduler est indisponible 2. Lancer la commande <i>vishnu_list_jobs</i> avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message d'erreur correspondant. | - Un message correspondant au cas d'erreur est affiché | |

4.2 Tests de stress

4.2.1 Présentation

Ces tests permettent de vérifier que la cible de test peut réussir à basculer et être récupérée 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.

4.2.2 Environnement des tests

Nous utiliserons l'environnement de test décrit dans le chapitre 2 dans un espace isolé (réseau et systèmes non partagés).

4.2.3 Identifiants de tests

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

4.2.4 Liste des tests

| Identifiant de tests | Technique | Critère de complétion |
|-----------------------|--|--|
| STR-SERV-DOWN | - Lancer la commande <i>vishnu_submit_job</i> - Arrêter le serveur TMS après l'exécution de la commande - Redémarrer le serveur TMS - Lancer la commande <i>vishnu_cancel_job</i> | - Terminaison normale de la commande <i>vishnu_cancel_job</i> |
| STR-AGENT-DOWN | - Lancer la commande <i>vishnu_submit_job</i> - Arrêter l'agent SysFera-DS après l'exécution de la commande - Redémarrer l'agent SysFera-DS - Lancer la commande <i>vishnu_cancel_job</i> | - Terminaison normale de la commande <i>vishnu_cancel_job</i> |
| STR-BATCH-DOWN | - Arrêter le Batch scheduler | - Une erreur due à une indisponibilité |

- Lancer la commande ***vishnu_submit_job***

du Batch scheduler est retournée par le système.

4.3 Contrôle des performances

4.3.1 Présentation

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

4.3.2 Environnement de tests

Nous utiliserons l'environnement de test décrit dans le chapitre 2 dans un espace isolé (réseau et systèmes non partagés).

4.3.3 Identifiants des tests

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

4.3.4 Liste des tests

| Identifiant de tests | Technique | Critère de completion |
|-----------------------------|--|--|
| CDP-3.1-submit a job | <p>Cas 1 :Batch scheduler Torque - Lancer une commande '<i>ssh <host> qsub</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 2 :Batch scheduler LoadLeveler Lancer une commande '<i>ssh <host> llsbmit</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>- Lancer la commande <i>vishnu_submit_job</i> en mesurant le temps de réponse</p> | temps inférieur au temps de référence (valeur moyenne sur 10 itérations) |
| CDP-3.1-cancel a job | <p>Cas 1 :Batch scheduler Torque - Lancer une commande '<i>ssh <host> qdel</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 2 :Batch scheduler LoadLeveler Lancer une commande '<i>ssh <host> llcancel</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>- Lancer la commande <i>vishnu_cancel_job</i> en mesurant le temps de réponse</p> | temps inférieur au temps de référence (valeur moyenne sur 10 itérations) |
| CDP-3.1-list jobs | <p>Cas 1 :Batch scheduler Torque - Lancer une commande '<i>ssh <host> qstat</i>'</p> | temps inférieur au temps de référence (valeur moyenne sur 10 itérations) |

| | | |
|----------------------------------|---|--|
| | <p>sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 2 :Batch scheduler LoadLeveler Lancer une commande '<i>ssh <host> llq</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>- Lancer la commande <i>vishnu_list_jobs</i> en mesurant le temps de réponse temps de réponse (temps de référence).</p> | itérations) |
| CDP-3.2.1-list job queues | <p>Cas 1 :Batch scheduler Torque - Lancer une commande '<i>ssh <host> qstat -Q</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 2 :Batch scheduler LoadLeveler Lancer une commande '<i>ssh <host> llclass</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>- Lancer la commande <i>vishnu_list_queues</i> en mesurant le temps de réponse</p> | temps inférieur au temps de référence (valeur moyenne sur 10 itérations) |
| CDP-3.2.2-submit a job | - Lancer le serveur TMS | Consommation CPU inférieure à 10%. |

4.4 Tests de charge

4.4.1 Présentation

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

4.4.2 Environnement de tests

Nous utiliserons l'environnement de test décrit dans le chapitre 2 dans un espace isolé (réseau et systèmes non partagés).

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

4.4.3 Identifiants de tests

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

4.4.4 Liste des tests

| Identifiant de tests | Technique | Résultats attendus |
|------------------------------|------------------------|-------------------------------------|
| LOAD-2.1-submit a job | - Lancer 100 commandes | - Terminaison normale de toutes les |

| | |
|----------------------------|----------------------------------|
| VISHNU_D4_1b-TMS-PlanTests | Politique de tests du module TMS |
|----------------------------|----------------------------------|

| | | |
|--|--|---|
| | <i>vishnu_submit_job</i> simultanées | commandes - Etat normal du serveur TMS |
| LOAD-2.1-list jobs, LOAD-2.1-list job queues, | - Lancer 100 commandes <i>vishnu_list_jobs/ vishnu_list_queues</i> simultanées | - Terminaison normale de toutes les commandes - Etat normal du serveur TMS |

4.5 Contrôle de la qualité du code

4.5.1 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.

4.5.2 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

4.5.3 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. |

| | |
|----------------------------|----------------------------------|
| VISHNU_D4_1b-TMS-PlanTests | Politique de tests du module TMS |
|----------------------------|----------------------------------|

| | | |
|----------------|--|--------------------------------|
| DDC-JSC | Audit du code Java avec l'outil SQUORE | Moins de 10% de code dupliqué. |
|----------------|--|--------------------------------|

4.6 Tests de compilation et d'installation

4.6.1 Présentation

La compilation des différents exécutables et bibliothèques composant le module TMS 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).

4.6.2 Environnement des tests

Nous utiliserons l'environnement de test décrit dans le chapitre 2.

4.6.3 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 <ul style="list-style-type: none"> • serveur de calcul • serveur de stockage • serveur dédié VISHNU • serveur de base de données | <ul style="list-style-type: none"> - Tous les processus démarrés et opérationnels - Documentation disponible (pages man) |
| INS-CLI | Installation du client VISHNU en ligne de commande | <ul style="list-style-type: none"> - Toutes les commandes installées - Documentation disponible (pages man) |
| INS-PGAPI | Installation des bibliothèques API VISHNU (C++ et Python) | <ul style="list-style-type: none"> - 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 | <ul style="list-style-type: none"> - Services opérationnels (contrôle via la console webservices de JBoss) |

4.7 Contrôle de la documentation utilisateur

4.7.1 Présentation

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

4.7.2 Liste des tests

| | |
|----------------------------|----------------------------------|
| VISHNU_D4_1b-TMS-PlanTests | Politique de tests du module TMS |
|----------------------------|----------------------------------|

| 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-USERSGUIDE | Suivre le guide de l'utilisateur | Le guide permet une prise en main rapide des fonctions du logiciel |