

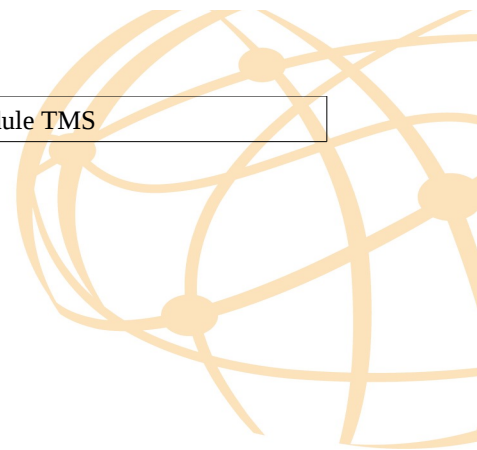
PROJET VISHNU
DOCUMENTS DE REFERENCE

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

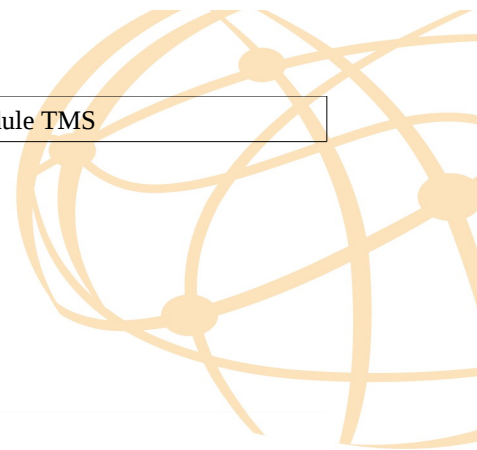
Version	Date	Modifications	Auteur
V1.0	21/03/11	Version initiale	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi
V1.1	22/07/11	Prise en compte de SLURM	Daouda Traoré
V1.2	14/12/11	Prise en compte de l'expression des besoins d'évolution de VISHNU : - ajout des cas de tests <i>T1.1-B2:submit a job</i> et <i>T1.1-B3:submit a job</i>	Daouda Traoré
V1.3	10/01/12	Amélioration du module TMS : - ajout des cas de tests <i>T1.2-B: automatic submit a job</i> , <i>T2.8-B: list jobs on all machines</i> , et <i>T2.4-B2:list jobs</i>	Daouda Traoré
V1.4	28/02/12	Prise en compte du support de LSF	Daouda Traoré
V1.5	12/04/2012	Prise en compte du support de Grid Engine	Amine Bsila
V1.6	25/04/12	Tests de submit job avec paramètres textuels et fichiers.	Kevin Coulomb
V1.7	22/05/12	Tests de submit job avec un script exploitant la fonctionnalité du répertoire de sortie auto-généré	Rodrigue Chakode



VALIDATION DU DOCUMENT

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_1g-ARCH	VISHNU Technical Architecture



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

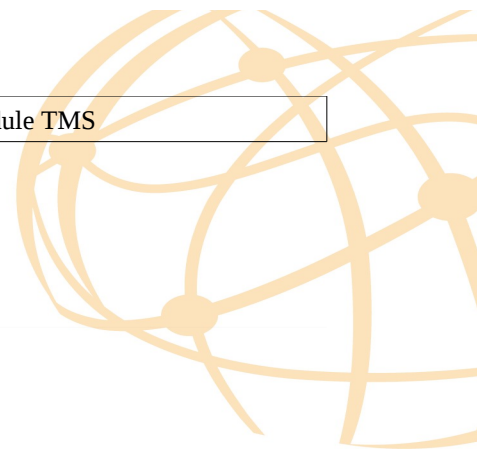
- **Script de test** : c'est un programme qui contient les instructions permettant de réaliser les tests.
- **Certificat** : correspond à la « *session key* » définie dans [SPECSEGEN].

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 les différents environnements de tests suivants avec des systèmes dédiés aux tests:

Environnement 1 :

- 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 : un serveur TMS sur une machine sur laquelle est installée un gestionnaire de ressources Torque [ARCH] .
- Un serveur SysFera-DS dédié décrit dans [ARCH].

Environnement 2 :

- 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 : un serveur TMS sur une machine sur laquelle est installée un gestionnaire de ressources LoadLeveler [ARCH].
- Un serveur SysFera-DS dédié décrit dans [ARCH].

Environnement 3 :

- 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 : un serveur TMS sur une machine sur laquelle est installée un gestionnaire de ressources SLURM [ARCH].
- Un serveur SysFera-DS dédié décrit dans [ARCH].

Environnement 4 :

- 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 : un serveur TMS sur une machine sur laquelle est installée un gestionnaire de ressources LSF [ARCH].
- Un serveur SysFera-DS dédié décrit dans [ARCH].

Environnement 5 :

- 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 : un serveur TMS sur une machine sur laquelle est installée un gestionnaire de ressources Grid Engine [ARCH].
- Un serveur SysFera-DS dédié décrit dans [ARCH].

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).
- 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.

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 :

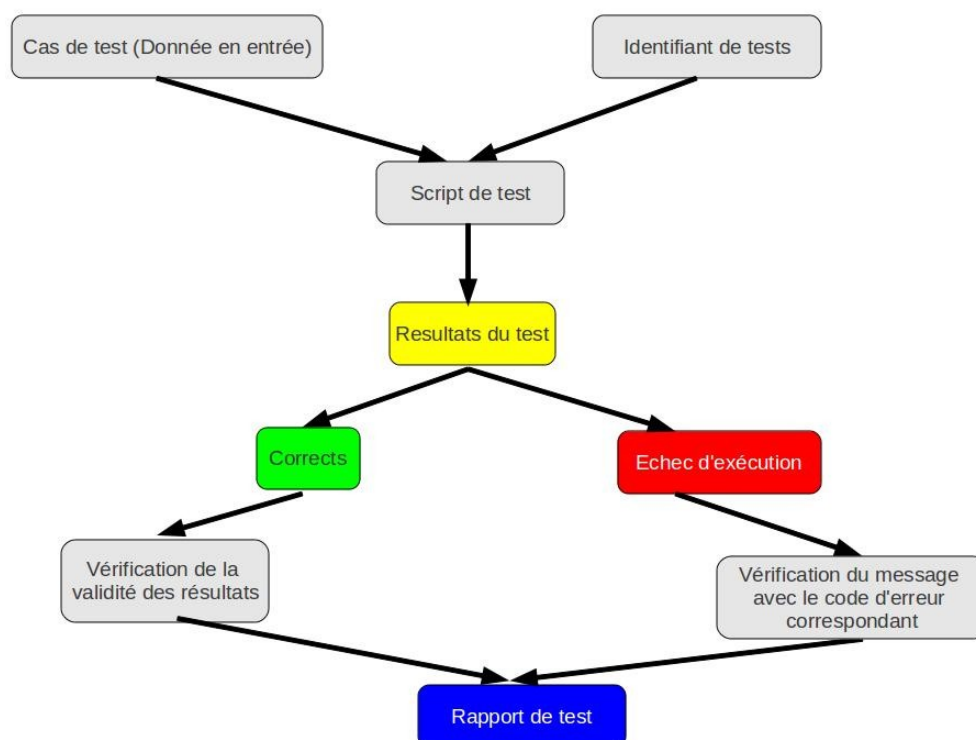


Illustration 1: Schema du test

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 sont classées par catégories de tests :

- Catégorie 1 : regroupe les fonctionnalités permettant de soumettre une action au serveur TMS: *vishnu_submit_job*, *vishnu_cancel_job*.
- 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_list_queues*, *vishnu_list_jobs*.

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 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]:<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.

4.1.2 Tests de la Catégorie 1

Identifiant de tests	Actions	Résultats	Considérations particulières
T1.1-B1: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. Lancer la commande vishnu_list_jobs et vérifier que 	- 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.

Identifiant de tests	Actions	Résultats	Considérations particulières
	l'identifiant du job est bien présent dans la liste.		
T1.1-B2:submit a job	<ol style="list-style-type: none"> Cas de test : les paramètres sont corrects et les variables d'environnements définies par VISHNU sont utilisées dans le script à exécuter. 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 et les variables d'environnement VISHNU. Si l'identifiant de machine et le script sont valides, un job sera créé et son identifiant retourné. Attendre la fin de l'exécution du job. Lancer la commande vishnu_get_job_output avec l'identifiant de la machine sur laquelle le job a été soumis et l'identifiant du job. Vérifier que : <ul style="list-style-type: none"> ● \$VISHNU_BATCHJOB_ID : correspond à l'identifiant donné au job par le gestionnaire de ressources sous-jacent ● \$VISHNU_BATCHJOB_NAME : correspond bien au nom donné au job ● \$VISHNU_SUBMIT_MACHINE_NAME : correspond bien au nom de la machine sur laquelle le job a été soumis ● \$VISHNU_BATCHJOB_NODEFILE : contient les noeuds alloués pour l'exécution du job ● \$VISHNU_BATCHJOB_NUM_NODES : contient le nombre de noeuds alloués pour l'exécution du job 		
T1.1-B3: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, le chemin vers le script contenant les caractéristiques, et avec chaque option associée à une valeur. Si l'identifiant de machine et le script sont valides, un identifiant du job généré par VISHNU. Lancer la commande vishnu_list_jobs et vérifier que l'identifiant du job est bien présent dans la liste, et que les valeurs données aux options dans l'étape 2 sont bien exactes. 		
T1.1-B4:submit a job	<ol style="list-style-type: none"> Cas de test : les paramètres sont corrects et le script à exécuter nécessite des paramètres (textes ou fichiers) qui sont passés lors de la soumission du job. Lancer la commande vishnu_submit_job avec l'identifiant de la machine sur laquelle le job sera soumis, 		

Identifiant de tests	Actions	Résultats	Considérations particulières
	<p>le chemin vers le script contenant les caractéristiques tout en spécifiant les paramètres d'exécutions (textuel et fichier).</p> <p>3. Si l'identifiant de machine et le script sont valides, un identifiant du job généré par VISHNU.</p> <p>4. Lancer la commande vishnu_list_jobs et vérifier que l'identifiant du job est bien présent dans la liste, et récupérer l'output du job pour voir si les valeurs des paramètres utilisés correspondent à ceux donnés.</p>		
T1.1-B5:submit a job	<p>1. Cas de test : les paramètres sont corrects et le script à exécuter fait référence à la variable d'environnement \$VISHNU_OUTPUT_DIR pour stocker des fichiers générés pendant l'exécution.</p> <p>2. Lancer la commande vishnu_list_job en spécifiant un identifiant de machine ainsi que le script à exécuter. Aussi bien l'identifiant que le script doivent être valides.</p> <p>3. Un job sera créé et son identifiant retourné.</p> <p>4. Lancer la commande vishnu_list_jobs et vérifier que l'identifiant du job est bien présent la liste retournée.</p> <p>5. Lancer la commande vishnu_get_job_info en spécifiant l'identifiant de la machine et l'identifiant du job et vérifier que le répertoire de sortie est bien généré et listé dans la sortie de la commande.</p> <p>6. A la fin de l'exécution du job, lancer la commande vishnu_get_job_output en spécifiant l'identifiant de la machine et l'identifiant du job et vérifier que les fichiers générés par le script seront bien téléchargés.</p>		
T1.1-E1:submit a job T1.1-E2 : submit a job T1.1-E3 : submit a job T1.1-E4 : submit a job	<p>1. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E3 : une option inconnue ou une erreur de syntaxe est introduite dans le script de soumission ● Cas E4 : le chemin jusqu'au script est incorrect <p>2. Lancer la commande vishnu_submit_job avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message correspondant.</p>	- Un message correspondant au cas d'erreur est affiché.	
T1.2-B: automatic submit job	<p>1. Cas de test : les paramètres sont corrects.</p> <ul style="list-style-type: none"> ● Pour la validité ce test, on suppose que le serveur TMS est lancé sur deux machines ayant respectivement pour identifiant MA_1 et MA_2. Les gestionnaires de ressources de ces deux machines ne 	- Un job est automatiquement soumis au Batch scheduler et un identifiant de job est	Deux SeDs TMS doivent être lancés sur deux

Identifiant de tests	Actions	Résultats	Considérations particulières
	<p>doivent avoir aucun job en attente.</p> <ol style="list-style-type: none"> Lancer les commandes vishnu_cancel_job MA_1 all et vishnu_cancel_job MA_2 all. Lancer la commande vishnu_submit_job avec comme paramètres MA_1 et un script contenant la directive permettant de donner la date de commencement du job (par exemple 1h après la date de lancement de la commande vishnu_submit_job). Le choix de cette directive à pour but de forcer l'état en attente (<i>waiting</i>). Lancer la commande vishnu_submit_job avec comme paramètre l'identifiant de la machine égale à autom 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. Lancer la commande vishnu_list_jobs et vérifier que le job a été bien lancé sur la machine MA_2. 	généré par VISHNU.	<p>machines.</p> <p>Le script à soumettre doit être générique et doit respecter les directives du Batch scheduler et les directives du script générique de vishnu</p>
T1.2-E1: automatic submit job	<ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : l'utilisateur n'a pas de compte sur aucune des machines Lancer la commande vishnu_submit_job avec l'identifiant de la machine égale à autom et le cas de test E1 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	<ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU Cas E3 : l'identifiant de job n'est pas enregistré dans VISHNU Cas E4 : la commande est lancée par un utilisateur autre que celui qui a soumis le job et qui n'est pas 	1.- Le message d'erreur correspondant au cas de test est affiché	

Identifiant de tests	Actions	Résultats	Considérations particulières
	<p>administrateur.</p> <p>3. Lancer la commande vishnu_cancel_job avec le cas de test (E1, E2, E3 ou E4) et vérifier que son exécution échoue avec le message d'erreur correspondant.</p>		

4.1.3 Tests de la catégorie 2

Identifiant de tests	Actions	Résultats	Considérations particulières
T2.1-B:get job information	<p>4. Cas de test : les paramètres sont corrects.</p> <p>5. 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.</p> <p>6. 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.</p> <p>7. Comparer les résultats des commandes précédentes et vérifier qu'ils sont identiques.</p>	- les informations correspondant au job sont retournées.	Les commandes doivent être lancées avant la fin d'exécution du job.
T2.1-E1:get job information T2.1-E2:get job information T2.1-E3:get job information	<p>1. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E3 : l'identifiant du job n'est pas enregistré dans VISHNU <p>2. Lancer la commande vishnu_get_job_info avec le cas de test (E1, E2 ou E3) et vérifier que son exécution échoue avec le message d'erreur correspondant.</p>	- Un message d'erreur correspondant au cas d'erreur est affiché	
T2.5-B:get jobs progression	<p>Cas de test : les paramètres sont corrects.</p> <p>1. 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 puis conserver les identifiants de job retournés par le système.</p> <p>2. 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.</p> <p>3. Répéter la commande précédente jusqu'à ce que la progression</p>	- La progression du job est retournée par le système	

	<p>du job soit positive.</p> <p>4. Faire <i>vishnu_get_job_out_put</i> 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>		
<i>T2.5-E1:get jobs progression</i> <i>T2.5-E2:get jobs progression</i> <i>T2.5-E3:get jobs progression</i>	<p>1. Cas de test : les paramètres sont incorrects</p> <ul style="list-style-type: none"> ● Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E3 : l'identifiant du job n'est pas enregistré dans VISHNU <p>2. Lancer <i>vishnu_get_job_progress</i> 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	
<i>T2.6-B:get one job output</i>	<p>1. Cas de test : les paramètres sont corrects.</p> <p>2. Lancer <i>vishnu_submit_job</i> 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. Attendre pendant une durée égale au temps d'exécution du job défini dans le script.</p> <p>4. Faire <i>vishnu_get_job_out_put</i> avec l'identifiant de la machine sur laquelle le job a été soumis et l'identifiant du job.</p> <p>A l'issu de la commande, vérifier que le fichier de sortie standard, le fichier d'erreurs standard, et éventuellement les fichiers générés par le job et stockés via la variable d'environnement \$VISHNU_OUTPUT_DIR ont été bien rapatriés. Ces fichiers seront téléchargés dans un répertoire auto-généré ou le cas échéant, dans un répertoire spécifié par l'utilisateur (voir l'option fournie à cet effet).</p>	- Les résultats de job sont récupérés.	
<i>T2.6-E1:get one job output</i> <i>T2.6-E2:get one job output</i> <i>T2.6-E3:get one job output</i>	<p>1. Cas de test : les paramètres sont incorrects</p> <ul style="list-style-type: none"> ● Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU ● Cas E3 : la progression du job n'est pas de 100% <p>2. Lancer le programme <i>vishnu_get_job_out_put</i> 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	
<i>T2.7-B : Get all completed jobs outputs</i>	<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 <i>vishnu_cancel_job</i>.</p>	- Les résultats des deux jobs soumis sont rapatriés.	La période de rafraichissement est d'une minute.

	<ol style="list-style-type: none"> 3. Lancer deux fois 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 dont les chemins « <i>outputPath</i> » et « <i>errorPath</i> » et le temps d'exécution de chacun des jobs . Dans le script du premier job, le temps d'exécution estimé est d'une minute et dans le script du second job, il est de deux minutes. 4. Lancer la commande vishnu_get_all_jobs_out_put 5. Une minute après le lancement de la commande vishnu_get_all_jobs_out_put vérifier que les résultats du premier job ont été bien rapatriés dans les chemins spécifiés dans le script de soumission. 6. Deux minutes après le lancement de la commande vishnu_get_all_jobs_out_put vérifier que les résultats du second job ont été bien rapatriés. 		
T2.7-E1 :Get all completed jobs outputs T2.7-E2 : Get all completed jobs outputs	<ol style="list-style-type: none"> 1. Cas de test : les paramètres sont incorrects <ul style="list-style-type: none"> ● Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré ● Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU 2. Lancer la commande vishnu_get_all_jobs_out_put avec le cas de test (E1 ou E2) et vérifier que son exécution échoue avec le message d'erreur correspondant. 	- Le message d'erreur correspondant au cas d'erreur est affiché	
T2.3-B:list job queues	<ol style="list-style-type: none"> 1. Cas de test : les paramètres sont corrects. 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. 3. Se connecter au Batch scheduler associé à la machine précédente. <p>Cas 1 : le Batch scheduler est Torque - Lancer la commande qstat -Q et vérifier que la liste des queues obtenue est la même que celle qui précède.</p> <p>Cas 2 : le Batch scheduler est LoadLeveler - Lancer la commande llclass et vérifier que la liste des queues obtenue est la même que celle qui précède.</p> <p>Cas 3 : le Batch scheduler est SLURM - Lancer la commande scontrol show partition et vérifier que la liste des queues obtenue est la même que celle qui précède.</p> <p>Cas 4 : le Batch scheduler est LSF - Lancer la commande bqueues et vérifier que la liste des queues obtenue est la même que celle qui précède.</p> <p>Cas 5 : le Batch scheduler est Grid Engine - Lancer la commande qconf -sql et vérifier que la liste des queues obtenue est la même que celle qui précède.</p>	- Les queues du Batch scheduler sont listées.	

T2.3-E1:list job queues T2.3-E2:list job queues	<ol style="list-style-type: none"> Cas de test : paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU Lancer la commande vishnu_list_queues avec le cas de test (E1 ou E2) et vérifier que son exécution échoue avec le message d'erreur correspondant. 	- Un message correspondant au cas d'erreur est affiché	
T2.4-B:list jobs	<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 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 identifiants des jobs. Lancer la commande vishnu_list_jobs et vérifier que les cinq jobs précédents sont affichés. 	Les jobs soumis sont affichés.	
T2.4-B2:list jobs	<ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. Lancer la soumission d'un job avec le script en entrée de la commande choisie ci-dessous selon les cas suivants : <ul style="list-style-type: none"> Cas 1 : Le Batch scheduler Torque est installé sur la machine cible <ol style="list-style-type: none"> Lancer la commande 'ssh <host> qsub' Cas 2 : Le Batch scheduler LoadLeveler est installé sur la machine cible <ol style="list-style-type: none"> Lancer la commande 'ssh <host> llsubmit' Cas 3 : Le Batch scheduler SLURM est installé sur la machine cible <ol style="list-style-type: none"> Lancer la commande 'ssh <host> sbatch' Cas 4 : Le Batch scheduler LSF est installé sur la machine cible <ol style="list-style-type: none"> Lancer la commande 'ssh <host> bsub' Lancer la commande vishnu_list_jobs avec l'identifiant de la machine sur laquelle le job a été soumis et avec l'option batch (-b pour la ligne de commande). Vérifier à l'aide de la commande (qstat pour TORQUE, squeue pour SLURM, llq pour LoadLeveler et bjobs pour LSF) du batch sous-jacent sur la machine que les jobs soumis précédemment sont bien affichés. 	Les jobs soumis sont bien retournés avec les identifiants donnés par le batch sous-jacent.	L'option batch (-b pour la ligne de commande) n'est pas prise en compte avec le Batch scheduler Grid Engine.
T2.4-E1:list jobs T2.4-E2:list jobs	<ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Cas E2 : l'identifiant de la machine n'est pas enregistré dans VISHNU 	- Un message correspondant au cas d'erreur est affiché	

	2. Lancer la commande vishnu_list_jobs avec le cas de test (E1 ou E2) et vérifier que son exécution échoue avec le message d'erreur correspondant.		
T2.8-B:list jobs on all machines	<ol style="list-style-type: none"> Cas de test : les paramètres sont corrects. <ul style="list-style-type: none"> Pour la validité de ce test, on suppose que le serveur TMS est lancé sur deux machines. On suppose que les deux machines ont pour identifiants MA_1 et MA_2. Vérifier sur les deux machines qu'aucun job n'est soumis ou en cours d'exécution sinon les annuler à l'aide de vishnu_cancel_job. Lancer deux fois vishnu_submit_job avec l'identifiant MA_1 et le chemin vers le script contenant les caractéristiques du job puis conserver les identifiants des jobs. Lancer trois fois vishnu_submit_job avec l'identifiant MA_2 et le chemin vers le script contenant les caractéristiques du job puis conserver les identifiants des jobs. Lancer la commande vishnu_list_jobs avec l'identifiant de la machine égale à all et vérifier que les cinq jobs précédents sont affichés, et qu'il y a deux jobs qui ont été soumis sur MA_1 et trois jobs sur MA_2. 	Les jobs soumis sont affichés.	
T2.8-E1:list jobs on all machines	<ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : le certificat de session n'est pas enregistré dans VISHNU ou il a expiré Lancer la commande vishnu_list_jobs avec l'identifiant de la machine égale à all et le cas de test E1 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 1 décrit dans le chapitre 2.

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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Arrêter le Batch scheduler - Lancer la commande <i>vishnu_submit_job</i> 	- Une erreur due à une indisponibilité 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 1 décrit dans le chapitre 2.

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</p> <ul style="list-style-type: none"> - Lancer une commande '<i>ssh <host> qsub</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence) <p>Cas 2 :Batch scheduler LoadLeveler</p> <ul style="list-style-type: none"> Lancer une commande '<i>ssh <host> llsbmit</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence) 	temps inférieur au temps de référence (valeur moyenne sur 10 itérations)

	<p>Cas 3 :Batch scheduler SLURM Lancer une commande '<i>ssh <host> sbatch</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 4 :Batch scheduler LSF Lancer une commande '<i>ssh <host> bsub</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 5:Batch scheduler Grid Engine - Lancer une commande '<i>ssh <host> qsub</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence) - Lancer la commande <i>vishnu_submit_job</i> en mesurant le temps de réponse</p>	
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>Cas 3 :Batch scheduler SLURM Lancer une commande '<i>ssh <host> scancel</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 4 :Batch scheduler LSF Lancer une commande '<i>ssh <host> bkill</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 5 :Batch scheduler Grid Engine - Lancer une commande '<i>ssh <host> qdel</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence) - 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>' 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>Cas 3 :Batch scheduler SLURM</p>	temps inférieur au temps de référence (valeur moyenne sur 10 itérations)

	<p>Lancer une commande '<i>ssh <host> squeue</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 4 :Batch scheduler LSF Lancer une commande '<i>ssh <host> bjobs</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 5 :Batch scheduler Grid Engine - Lancer une commande '<i>ssh <host> qstat</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>	
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>Cas 3 :Batch scheduler SLURM Lancer une commande '<i>ssh <host> scontrol show partition</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 4 :Batch scheduler LSF Lancer une commande '<i>ssh <host> bqueues</i>' sur une machine serveur en mesurant le temps de réponse (temps de référence)</p> <p>Cas 5 :Batch scheduler Grid Engine - Lancer une commande '<i>ssh <host> qconf -sql</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%.
CDP-MemServer	<p>- Lancer le serveur TMS</p> <p>- Laisser le serveur fonctionner sans utiliser les services pendant 24h</p> <p>- Vérifier que l'espace mémoire consommé par le processus serveur n'a pas augmenté.</p>	Consommation mémoire constante.

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 1 et 3 décrit dans le chapitre 2.

Par ailleurs, les requêtes envoyées en simultané 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 <i>vishnu_submit_job</i> simultanées	- Terminaison normale de toutes les 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.
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 	<ul style="list-style-type: none"> - Tous les processus démarrés et opérationnels - Documentation disponible (pages man)

	<ul style="list-style-type: none"> • serveur dédié VISHNU • serveur de base de données 	
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 librairies API VISHNU (C++ et Python)	<ul style="list-style-type: none"> - Librairie C++ installée - Librairie 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 operationnels (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

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