SUPV003 - Test du superviseur

Résumé :

Ce test n'a aucune signification physique, c'est uniquement un test informatique.
Il permet de vérifier la non régression de fonctionnalités de bas niveau comme des enrobages Python à routines fortran.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des matières

1 Problème de référence....3
2 Solution de référence....3
3 Modélisation A....3
3.1 Caractéristiques de la modélisation....3
4 Modélisation B....3
4.1 Caractéristiques de la modélisation....3
5 Modélisation D....3
5.1 Caractéristiques de la modélisation....3
6 Modélisation E....3
6.1 Caractéristiques de la modélisation....3
7 Synthèse des résultats....4

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Problème de référence

Sans objet.

2 Solution de référence

Il s'agit d'un test de fonctionnement. Il n'y a pas d'incertitude sur le comportement attendu.

3 Modélisation A

3.1 Caractéristiques de la modélisation

On teste des fonctions de base accessibles dans l'espace Python :

- `aster.get_option_dependency()` : Cette fonction fournit les options qui doivent être connues au préalable pour calculer une option donnée (c'est à dire les dépendances de cette option).
- Internationalisation des messages : On vérifie que la fonction de traduction de `gettext` (fonction « `_()` ») est disponible, et qu'elle retourne la traduction attendue pour une chaîne simple. Le résultat dépend de `LANG='EN'` dans `DEBUT`, de la bonne installation du fichier `share/locale/aster/en/LC_MESSAGES/aster_messages.mo` et du bon chargement de ce fichier par le module `i18n.py`.

4 Modélisation B

4.1 Caractéristiques de la modélisation

Cette modélisation vérifie que le bilan des alarmes émises est correctement envoyé vers le processeur #0.
Dans `FIN`, chaque processeur envoie au processeur #0 le nombre d'alarmes émises durant l'exécution. Ici, le processeur #1 émet 2 alarmes.
On enrobe la fin d'exécution pour déclencher un erreur si on ne trouve pas les deux alarmes attendues.

5 Modélisation D

5.1 Caractéristiques de la modélisation

Cette modélisation vérifie le bon fonctionnement des fonctions du module Python de Code_Aster. Essentiellement, ce sont les fonctions qui permettent de récupérer une copie des valeurs de certains concepts (fonctions, listes de réels, vecteurs et collections Jeveux) dans l'espace Python.
On crée des concepts triviaux dont on récupère les valeurs ensuite.
On vérifie également que des objets Python de base sont bien transmis en poursuite.

6 Modélisation E

6.1 Caractéristiques de la modélisation

On teste dans cette modélisation le fonctionnement les enrobages Python de fonctions MPI comme la fonction utilitaire pour récupérer le nombre de processeurs et le rang ou bien la communication d'une chaîne de caractères.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

7 Synthèse des résultats

Sans objet pour un test de fonctionement.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)