PERF016 – Performances d'INFO_MODE/ MACRO_MODE_MECA parallèles

Résumé :

L'objectif de ce cas-test est de mesurer les performances du cumul de deux stratégies de parallélisation dans INFO_MODE et MACRO_MODE_MECA. Les gains en temps et en mémoire sont intéressants et ces accélérations (jusqu'à 2 fois en pic mémoire, 30/40 fois en temps sur une soixantaine de processeurs) peuvent réellement faciliter de nombreuses études dynamiques sur base modale.

1 Problème de référence

Le cas-test perf013c sert de problème de référence.

Les gros calculs modaux, en taille de problème et/ou en nombre de modes recherchés, sont effectués *via* l'opérateur `MACRO_MODE_MECA`. Cet opérateur découpe la bande fréquentielle de recherche en plusieurs sous-bandes contiguës de manière à réduire les consommations en temps et en mémoire du calcul. Cette recherche par sous-bandes permet aussi d'améliorer la robustesse et la précision des résultats.

Pour limiter les déséquilibres de charge dans le calcul modal, on les pré-calibre *via* un appel à `INFO_MODE`. Ce dernier opérateur active les mêmes deux niveaux de parallélisme que ceux de `MACRO_MODE_MECA`.

La décomposition en sous-bandes procure enfin un dernier avantage, elle aménage un niveau de parallélisme explicite et très efficace, en permettant la distribution de chaque sous-bande sur 1 ou plusieurs processeurs. Ainsi, les calculs modaux opérés au sein de chaque sous-bande se déroulent concurremment. Si plusieurs processeurs sont disponibles pour chaque sous-bande, on peut même appeler un deuxième niveau de parallélisme *via* le solveur linéaire direct MUMPS. Ces deux niveaux de parallélisme conjugués permettent de gagner beaucoup en temps et un peu en pic mémoire RAM (mot-clé `NIVEAU_PARALLELISME='COMPLET'`, valeur par défaut).

Pour axer les gains principalement sur le pic mémoire ou parce que le nombre de processeurs disponibles est insuffisant par rapport au nombre de sous-bandes, on peut aussi limiter le parallélisme au seul deuxième niveau, celui de MUMPS (`NIVEAU_PARALLELISME='PARTIEL'`).

L'objectif de ce cas-test est de mesurer les performances de ces deux stratégies de parallélisation dans `INFO_MODE` et `MACRO_MODE_MECA`. Les gains en temps et en mémoire sont intéressants et ces accélérations (jusqu'à 2 fois en pic mémoire, 30/40 fois en temps sur une soixantaine de processeurs) peuvent réellement faciliter de nombreuses études dynamiques sur base modale.

Le calcul modal utilisé reprend la modélisation C du cas-test *perf013*: plaque carrée maillée en éléments de coques ($N=4M$ degrés de liberté), éléments finis linéaires, recherche de 50 modes propres en 4 sous-bandes avec le solveur modal par défaut et le solveur linéaire MUMPS. On teste la configuration séquentielle (modélisation A), ainsi que les deux stratégies parallèles sur 4 et 16 processeurs (modélisations B/C et D/E).

Pour les calculs `MACRO_MODE_MECA`, on vérifie différentes valeurs de fréquence et différentes composantes des masses modales effectives unitaires¹.

Pour les calculs `INFO_MODE`, on teste les valeurs minimales, maximales et la somme des composantes '`NB_MODE`' de la table générée.

2 Solution de référence

Voir le cas-test perf013c.

1 Aux bornes des sous-bandes pour détecter d'éventuels problèmes dans les communications MPI.

3 Modélisation A

3.1 Caractéristiques de la modélisation A

Nombre de processeur : 1

On utilise la modélisation C du cas-test perf0013: plaque carrée maillée en éléments de coques, éléments finis linéaires, recherche de 50 modes propres *via* l'opérateur `MACRO_MODE_MECA`. On utilise le solveur modal et la valeur du test de Sturm par défaut.

On pré-calibre le calcul modal par un appel à `INFO_MODE` sur les sous-bandes de fréquences: $[0.1\text{GHz}, 1.1\text{GHz}]$, $[1.1\text{GHz}, 2.0\text{Hz}]$, $[2.0\text{GHz}, 3.0\text{GHz}]$ et $[3.0\text{GHz}, 4.0\text{GHz}]$.

On paramètre la brique solveur linéaire avec le meilleur outil disponible actuellement dans *Code Aster* pour ce type de calcul: le solveur MUMPS (`METHODE='MUMPS'`) couplé au renuméroteur QAMD (`RENUM='QAMD'`) et appelé en mémoire (pour accélérer les nombreuses descente-remontées, `GESTION_MEMOIRE='IN_CORE'`).

Caractéristiques du maillage : 667 489 `NOEUD`, 3 264 `SEG2` et 665 856 `QUAD4`.

Nombre de degrés de liberté : 4 024 530.

3.2 Résultats

Grandeur	Référence
<i>FREQ</i> n°1	47.605
<code>MASS_EFFE_UN_DZ</code> n°1	0.420180
<i>FREQ</i> n°10	1546.457
<code>MASS_EFFE_UN_DZ</code> n°10	$3.423 \cdot 10^{-3}$
<i>FREQ</i> n°21	3978.453
<code>MASS_EFFE_UN_DZ</code> n°21	$1.467 \cdot 10^{-3}$

4 Modélisation B

4.1 Caractéristiques de la modélisation B

Nombre de processeurs: 4.

Identique à la modélisation A mais le calcul est effectué ici sur 4 processeurs en privilégiant le premier niveau de parallélisme (celui par défaut *via* `NIVEAU_PARALLELISME='COMPLET'`).

Chaque sous-bande est confiée à un processeur. Les occurrences MUMPS, appelées concurremment par chacune des sous-bandes, ne travaillent donc que sur 1 processeur.

4.2 Résultats

Identique à la modélisation A.

5 Modélisation C

5.1 Caractéristiques de la modélisation C

Nombre de processeurs: 16.

Identique à la modélisation A mais le calcul est effectué ici sur 16 processeurs en utilisant deux niveaux de parallélisme (fonctionnement par défaut *via* NIVEAU_PARALLELISME='COMPLET').

Chaque sous-bande est confiée à quatre processeurs. Les occurrences MUMPS, appelées concurremment par chacune des sous-bandes, travaillent donc chacune sur 4 processeurs.

5.2 Résultats

Identique à la modélisation A.

6 Modélisation D

6.1 Caractéristiques de la modélisation D

Nombre de processeur : 4.

Identique à la modélisation A mais le calcul est effectué ici sur 4 processeurs en privilégiant le second niveau de parallélisme (NIVEAU_PARALLELISME='PARTIEL'). Toutes les sous-bandes sont traitées les unes après les autres.

Seules les occurrences MUMPS, appelées pour les factorisations numériques et les descente-remontées requises par le solveur modal, travaillent en parallèle sur 4 processeurs.

6.2 Résultats

Identique à la modélisation A.

7 Modélisation E

7.1 Caractéristiques de la modélisation E

Identique à la modélisation D mais le calcul est effectué ici sur 16 processeurs.

7.2 Résultats

Identique à la modélisation A.

8 Synthèse des résultats

Ces résultats ont été obtenus en version 11.3.9 sur la machine IVANOE avec 1 processus MPI par noeud. Pour rappel, les performances en version 11.2 (paramètres par défaut) étaient: 9296 s et 23.5 Go (pour MACRO_MODE_MECA) et 2016 s et 17.0 Go (pour INFO_MODE).

V11.3.9 INFO_MODE/ MACRO_MODE_MECA	A	B	C	D	E
Type de parallélisme	1×1	4×1	4×4	1×4	1×16
Temps Elapsed	1008 s 3544 s	387 s 1123 s	170 s 636 s	422 s 1853 s	253 s 1432 s
Vmpeak	7.8 Go 17.2 Go	7.8 Go 19.5 Go	7.8 Go 13.8 Go	7.8 Go 10.6 Go	7.8 Go 9.5 Go