LUCIA,
l'outil d'équilibrage de charge pour un système couplé OASIS

Projet PULSATION, ANR-11-MONU-010-02
Livrable D12
Janvier 2014

Sommaire

Ce document décrit la façon d'optimiser les performances d'un **système couplé** utilisant OASIS3-MCT, en affectant de la façon la plus efficace possible les ressources (cœurs de calcul) à chacun des modèles (composants) d'une configuration couplée.

L'utilisateur dispose pour cela d'un outil (LUCIA, pour « Load-balancing Utility and Coupling Implementation Appraisal ») qui extrait d'une simulation couplée toutes les informations utiles à l'analyse d'équilibre de charge, à savoir le temps d'attente et le temps de calcul effectif pour chacune des composantes du système.

Nous décrivons ici le mode de fonctionnement d'un système couplé ainsi que les les méthodes pour l'optimiser grâce aux informations données par l'outil LUCIA.

1. Description d'un système couplé utilisant OASIS3-MCT

1.1. Qu'est ce qu'un « système couplé » ?

Un modèle numérique de climat est un ensemble d'équations qui représentent mathématiquement une composante du système climatique. Un « système couplé » climatique est l'assemblage de plusieurs de ces modèles numériques qui peuvent être développés indépendamment les uns des autres, chacun représentant un milieu différent (océan, atmosphère,...). On appelle couplage la représentation des échanges qui ont lieu entre les différentes composantes. Ces informations échangées, appelées champs de couplage, sont discrétisées sur des grilles (ou maillages) qui peuvent être différentes d'un modèle à l'autre : il est donc parfois nécessaire d'interpoler une information de la grille du modèle source vers la grille du modèle cible. Les échanges entre les différents modèles d'un système couplé se font avec une périodicité appelée « **pas de temps de couplage** ». Un modèle utilise de l'information provenant d'un autre modèle pour faire ses calculs : à la fin d'un pas de temps de couplage, le modèle « recevant » va donc attendre l'information dont il a besoin pour reprendre ses calculs. L'objectif du travail qui suit est de proposer une méthode permettant de diminuer au maximum ce temps d'attente et ainsi accélérer la réalisation de la simulation couplée toute entière.

L'outil OASIS[1] – OASIS3-MCT, pour sa dernière version[2]- permet de coupler entre eux des modèles numériques climatiques. Ces modèles peuvent avoir été développés indépendamment les uns des autres. Pour mettre en place un couplage entre deux modèles, le développeur ajoute dans les différents codes sources des modèles les appels aux sous-routines de la librairie OASIS. Ces sous-routines permettent la réalisation des différents étapes du couplage : l'initialisation, la définition des champs de couplage, la réception et l'envoi des champs de couplage, la finalisation. Un champ de couplage est envoyé à un autre modèle au moyen d'un appel à l'instruction OASIS_put et reçu d'un autre modèle au moyen d'un appel à l'instruction OASIS_get. Les informations sur les champs de couplage à échanger au cours d'une simulation entre les modèles sont définies dans un fichier de paramètres appelé « namcouple ». L'échange des tableaux numériques contenant les champs de couplage se fait grâce à la librairie de communication MPI (Message Passing Interface).

1 Sophie Valcke, Tony Craig, Laure Coquart, 2013: OASIS3-MCT User Guide, OASIS3-MCT 2.0, Technical Report, TR/CMGC/13/17, CERFACS/CNRS SUC URA No 1875, Toulouse, France

2 Ce document décrit le fonctionnement d'un couplage utilisant la version 2.0 du coupleur OASIS3-MCT

Il existe deux façons de coupler entre eux des modèles interfacés avec OASIS: le mode **séquentiel** et le mode **concurrent**. D'une manière générale, à un pas de temps de couplage donné, un modèle à besoin des résultats des calculs de l'autre modèle pour commencer ses propres calculs. Si pendant qu'un modèle fait ses calculs, l'autre modèle attend (et réciproquement), c'est le mode séquentiel. Le mode concurrent permet en revanche aux deux modèles de tourner simultanément, chacun des deux modèles utilisant les résultats calculés au pas de temps précédent par l'autre modèle. A noter que les deux modes peuvent coexister dans un système comprenant plus de 2 modèles.

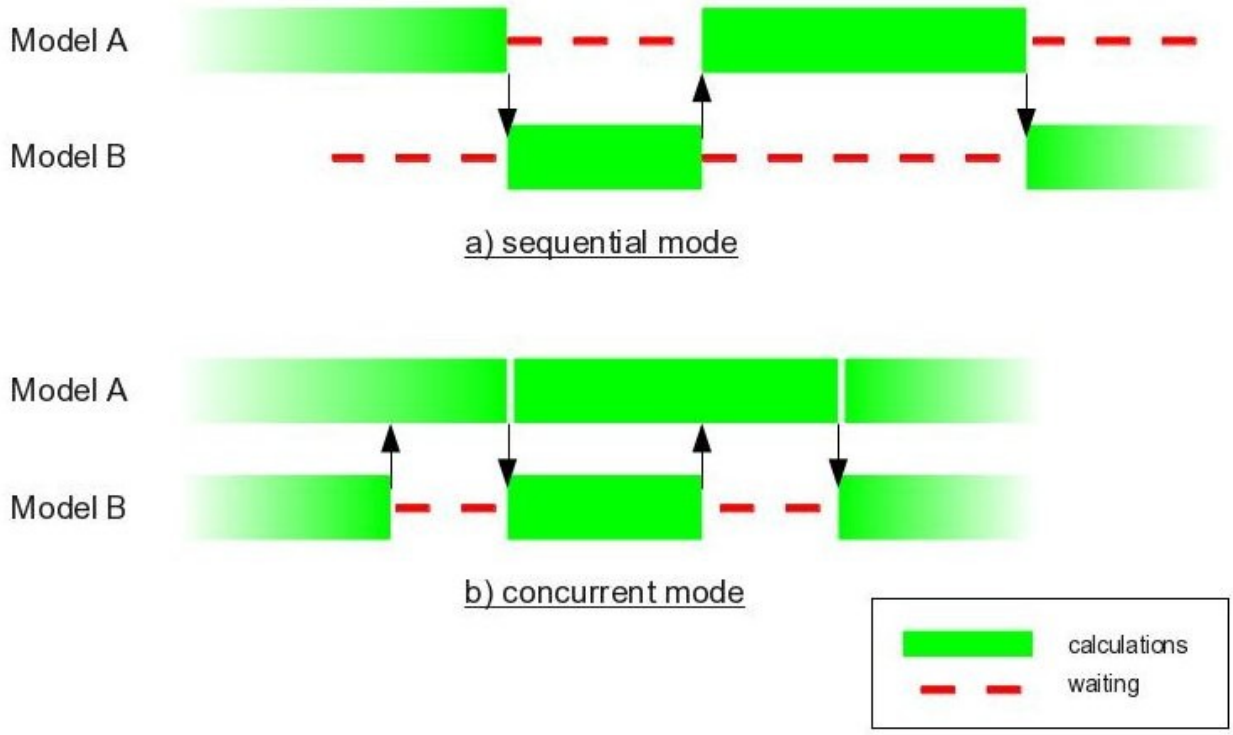

Fig 1 : Mode séquentiel, mode concurrent
dans le cas d'un système couplé composé de deux modèles

1.2. Les différentes phases d'un couplage

Un modèle fournit à OASIS des variables de couplage qui sont destinées à être utilisées par un (ou plusieurs) autre(s) modèle(s). Les grilles sur lesquelles sont discrétisées ces variables peuvent être différentes d'un modèle à l'autre : une interpolation de l'information de la grille du modèle source vers la grille du modèle cible est donc parfois nécessaire. Le cas échéant, le coupleur, utilisé comme une librairie par les modèles, va effectuer cette opération d'interpolation. Deux choix sont possibles :

- soit c'est le modèle source qui, avant d'envoyer sa variable de couplage au modèle cible, effectue l'interpolation.
- soit c'est le modèle cible qui, une fois la variable de couplage reçue, l'interpole sur sa propre grille afin de pouvoir l'utiliser.

Ce choix se spécifie dans le fichier de configuration d'OASIS (« namcouple ») pour chacun

des champs de couplage (mots clé « src » ou « dst » de l'instruction « MAPPING »)

Un pas de temps de couplage pour un modèle particulier se décompose en plusieurs étapes (pas nécessairement dans l'ordre donné) :

- le modèle fait ses calculs.
- le modèle reçoit les variables de couplage provenant d'un autre modèle (au moyen des des communications MPI réalisées via les appels à l'instruction Oasis_get).
- le modèle envoie les variables de couplage qu'il a produites (au moyen des communications MPI réalisées via les appels à l'instruction Oasis_put).
- le modèle fait les interpolations nécessaires avant l'envoi MPI (cas « src ») ou après la réception MPI (cas « dst »).

1.3. L'outil de mesure de performance LUCIA

L'outil LUCIA permet de mesurer le temps passé dans chacune des étapes listées ci-dessus pour chacun des modèles. Des mesures de temps d'horloge sont réalisées au cours de la simulation, sauvées dans des fichiers d'historique d'événements, puis post-traitées afin de fournir à l'utilisateur des informations interprétables et utilisables. Les prises de temps sont implémentées dans OASIS avant et après chaque échange de champ (reçu ou envoyé par un modèle), et avant et après chaque interpolation, pour chaque processus MPI (impliqué dans le couplage) de chacun des modèles: le temps est mesuré via la fonction MPI_Wtime. Un contrôle en début de simulation permet de s'abstraire d'un possible décalage entre les temps d'horloge des processus MPI. Les mesures sont prises sur un échantillon significatif de la simulation (hors phase d'initialisation et de finalisation).

L'utilisation de LUCIA n'est possible qu'à partir de la version 3.0 d'OASIS3-MCT, (ou dans la version TRUNK, à partir de la révision 934, mise à disposition sur demande). Le déclenchement de l'écriture des mesures dans des fichiers d'historique d'événement (nommés «lucia.MM.PPPPPP», avec MM numéro du modèle dans le système couplé, et PPPPPP rang MPI du processus dans le communicateur local du modèle) se fait en spécifiant la valeur -1 comme 2ème argument pour le mot clé $NLOGPRT dans le fichier namcouple.

Lorsque la simulation s'est terminée[3], le script « lucia » doit être lancé manuellement depuis le répertoire où se trouvent les fichiers lucia* produits. Ce script appelle un programme FORTRAN (compilable avec l'option -c du script « lucia ») qui relit les fichiers d'historique, calcule et affiche (dans la sortie standard et dans un fichier (info.dat) lisible par exemple grâce au logiciel gnuplot[4]) les quantités suivantes:

- **temps *En*** : temps passé par un modèle (n) à envoyer ou recevoir des messages MPI (nous parlons ici des messages MPI impliqués dans l'échange des variables de couplage et hors communications internes à chacun des modèles). Plus précisément ce temps *En* est le temps passé, pour l'ensemble des champs de couplage, entre le début et la fin d'un envoi de champ et entre le début et la fin d'une réception de champ. Ce temps rassemble donc tous les temps de **communication** (envoi et réception de chacun des champs de couplage). Étant

3 L'analyse LUCIA peut aussi se faire pendant la simulation, mais ne produira logiquement son analyse que sur le nombre de pas de temps de couplage déjà réalisés par le système couplé au moment de l'analyse.

4 http://www.gnuplot.info/

donné qu'OASIS utilise le mode non bloquant de la fonction d'envoi MPI (MPI_WAITALL + MPI_ISEND), le temps d'envoi représente le temps nécessaire à l'écriture en tampon du message MPI envoyant le champs de couplage[5]. Le temps de réception comprend, en plus du temps de lecture tampon du message MPI, l'éventuel **déséquilibre de charge**[6] entre les modèles: un modèle peut avoir à attendre que l'autre modèle termine ses calculs avant qu'il puisse envoyer les champs de couplage demandés. Ce temps d'attente est souvent d'un ordre de grandeur supérieur aux temps de communications MPI : ce temps *En* peut donc être appelé **temps d'attente**. Dans le cas où le modèle est parallélisé sur plusieurs processus MPI, ce temps *En* est le temps mesuré sur les envois et les réceptions les plus tardives entre tous les processus d'un modèle, et ceci pour chacun des champs de couplage. Nous avons fait le choix d'utiliser ces valeurs extrêmes plutôt que des moyennes pour inclure le jitter dans le temps de calcul (voir paragraphe suivant).

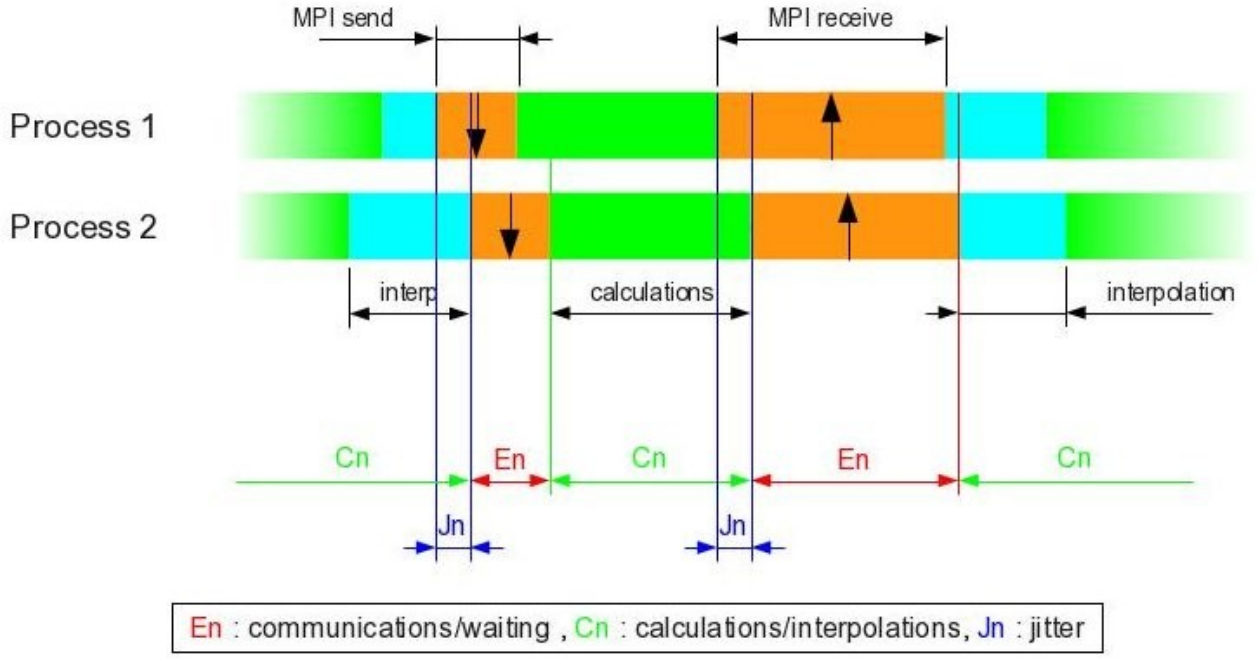

Figure 2 : Définition des quantités En, Cn et Jn (jitter) lors des opérations de réception et d'envoi de champs de couplage, dans le cas d'un modèle (n) réparti sur 2 processus

– **temps *Cn* :** temps passé par un modèle à faire ses propres **calculs** et les **interpolations** OASIS**.** Ce temps est le complément du temps *En* et la somme *Cn* +*En* doit être égale au temps de restitution de la part de la simulation qui a été utilisé pour établir les prises de temps[7]. *Cn* rend donc compte à la fois du temps passé par chaque modèle à faire des calculs mais également, dans le cas où le modèle est parallélisé, de l'éventuel déséquilibre interne du modèle (« jitter »). Ce **jitter** est en fait le temps nécessaire pour que tous les processus MPI du modèle

5 L'instruction MPI_WAITALL se trouve en fait devant MPI_ISEND, ce qui signifie qu'un message n'est envoyé que si le message précédent a été lu.

6 La notion de déséquilibre de charge s'applique ici à des modèles (un modèle est plus rapide ou plus lent qu'un autre) et non pas, comme cela est plus courant dans le domaine du calcul parallèle, aux différents processus d'un exécutable MPI (un processus MPI est plus rapide ou plus lent qu'un autre).

7 Le premier ou les deux premiers et le dernier pas de temps de couplage sont exclus de la mesure, pour ne pas tenir compte des éventuels décalages dus aux phases d'initialisation et de clôture.

soient prêts à envoyer ou recevoir une variable de couplage (temps entre le premier et le dernier envoi (ou réception)). Il peut provenir du modèle lui même ou bien plus probablement des interpolations OASIS :

- d'une part parce que certains processus peuvent avoir plus de sous-domaines que d'autres à interpoler (ce qui accroit le déséquilibre de charge entre processus d'un modèle et donc le jitter)
- d'autre part à cause de la présence de synchronisations dans la parallélisation interne des modèles, qui ont tendance à réduire ce jitter.

Il est à noter que ce temps de jitter est inclus dans le temps *Cn*, même s'il est fortement dépendant de l'optimisation du couplage et donc directement lié au temps *En*.

Outre ces deux quantités En et Cn, le script LUCIA donne les valeurs du temps de jitter et les valeurs (moyennes sur tous les processus) des temps d'interpolation d'OASIS, ceci pour chaque modèle.

La figure 3a donne un exemple de sortie graphique des quantités *Cn* (en vert) et *En* (en rouge) pour le système couplé PULSATION[8] comprenant le modèle d'atmosphère WRF et le modèle d'océan NEMO, avec un couplage en mode concurrent. On note que le modèle qui met le plus de temps à terminer ses calculs (NEMO, 195 s) a un temps d'attente quasi nul. Ceci s'explique par le fait que le modèle WRF (130s),plus rapide que NEMO, propose ses champs de couplage avant que NEMO ne les demande. Dans cette configuration, des ressources de WRF pourraient être ré-allouées à NEMO pour accélérer ce dernier et ainsi le système complet.

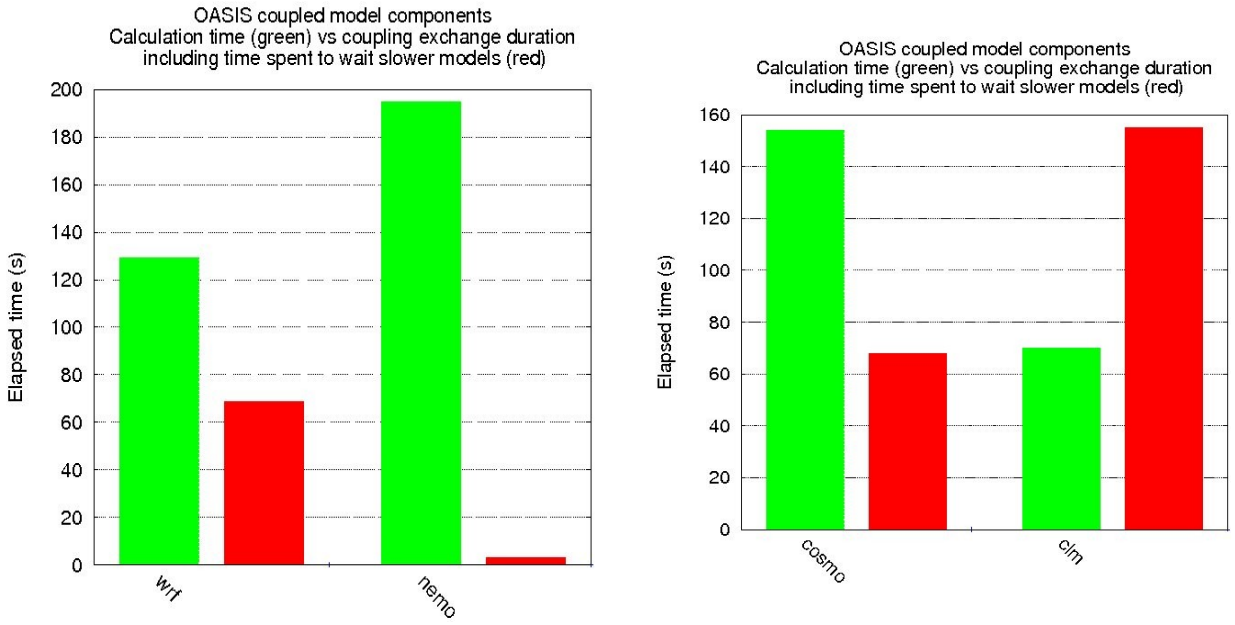

Figure 3 : Analyse d'équilibre de charge LUCIA du système couplé en mode concurrent WRF-NEMO (a) et du système couplé en mode séquentiel COSMO-CLM (b)

8 Masson, S., Hourdin, C., Benshila, R., Maisonnave, E., Meurdesoif, Y., Mazauric, C., Samson, G., Colas, F., Madec, G., Bourdallé-Badie, R., Valcke, S., Coquart, L., 2012: Tropical Channel NEMO-OASIS-WRF Coupled simulations at very high resolution , 11.4. 13th WRF Users' Workshop – 25-29 June 2012, Boulder, CO

Sur la figure 3b, ce sont les modèles d'atmosphère COSMO et de surface terrestre CLM qui forment le système couplé mais le mode de couplage est ici séquentiel. On voit dans ce cas que

$$C_1 = E_2 \text{ et } C_2 = E_1$$

ce qui s'explique par le fait qu'en mode séquentiel, chacun des deux modèles attend les résultats des calculs de l'autre modèle pour reprendre les siens (cf. figure 1).

2. Méthodes d'analyse et d'optimisation

L'objectif d'une analyse avec l'outil LUCIA est de permettre à un utilisateur d'affecter de façon optimale les ressources dont il dispose (cœurs de calcul) à chacun des modèles (composants) de sa configuration couplée. L'analyse et l'optimisation de son système peuvent être faites de deux façons différentes :

- soit en cherchant à effectuer la simulation du système couplé la plus rapide possible (temps de restitution le plus faible) pour un nombre de ressources fixé. Cette optimisation revient à équilibrer entre eux les modèles du système.
- soit en allouant modèle par modèle un nombre de ressources à définir tirant partie au mieux des scalabilités (ou efficacités) de chacun des modèles du système couplé.

Ces deux méthodes sont expliquées dans la suite de ce document, pour les deux modes séquentiel et concurrent du couplage.

2.1. Équilibrage des modèles entre eux, optimisation pour un nombre de ressources fixé.

Le point de départ de cette 1ère méthode est de fixer un nombre de cœurs de calcul pour faire tourner un système couplé. Ce nombre de cœurs fixé peut correspondre soit au nombre maximal de cœurs dont on dispose sur une machine donnée, soit au nombre de cœurs qui permette de faire plusieurs tests rapidement,soit enfin au nombre de cœurs que l'on voudra utiliser pour faire tourner des simulations du système couplé en mode production (simulations longues,...).

Dans le cas du mode séquentiel, la notion d'équilibrage des modèles n'intervient pas dans l'optimisation du système: en effet, les modèles tournant les uns à la suite des autres (cf. figure 1), l'objectif ici est donc tout simplement de réduire au maximum les temps de calcul de chacun d'entre eux et donc d'utiliser la méthode décrite au paragraphe 2.2.a. Nous nous intéresserons donc ici au seul mode concurrent.

Le point de départ de la méthode est de lancer une première simulation sur un nombre choisi de ressources et d'analyser les informations *Cn* et *En* obtenues (Cn_1 et En_1 pour le modèle 1, Cn_2 et En_2 pour le modèle 2).

Si le modèle 1 est plus rapide en temps de calcul que le modèle 2, on obtient :

$$Cn_1 < Cn_2$$
$$En_2 = 0$$
$$En_1 > 0$$

L'objectif dans ce cas est alors de diminuer au maximum le temps d'attente En_1 du modèle le plus rapide qui attend les données provenant du modèle le plus lent : ceci conduit donc à accélérer le modèle le plus lent (à nombre de cœurs fixé pour l'ensemble du système, l'accélération de l'un implique forcément le ralentissement de l'autre). Pour cela, il existe principalement deux possibilités :

- assigner au modèle le plus lent certains cœurs de calcul utilisés jusqu'alors par le modèle le plus rapide. Et on renouvelle l'opération jusqu'à obtenir $Cn_1 \sim Cn_2$ et En_1 et En_2 les plus petits possibles. La configuration optimale, du point de vue de l'équilibrage des modèles et de l'utilisation des ressources, obtenue au final est également la plus rapide.
- assigner le calcul des interpolations des champs de couplage au modèle le plus rapide (utilisation des mots clés « src » et « dst ») si ce n'est pas déjà le cas. Le gain possible dans ce cas-là est plus faible que la 1ère possibilité, les temps d'interpolation étant dans la plupart des cas assez faibles[9] par rapport aux temps des autres calculs des modèles. L'effet de l'interpolation peut cependant jouer sur le « jitter » du modèle et ainsi perturber l'équilibre général du système.

Cas particuliers

Option de namcouple : EXPOUT

Il est préférable de ne pas effectuer d'analyse LUCIA avec le mode « EXPOUT » (sauvegarde dans un fichier NETCDF des champs échangés à chaque pas de temps de couplage): le temps passé dans l'écriture est compté dans le temps de calcul *Cn* du modèle. L'écriture des champs de couplage dans OASIS3-MCT n'est pas parallèle, ce qui signifie que seul le processus maître de chaque modèle écrit dans les fichiers de sortie les données rassemblées. Ce travail n'étant effectué que par un seul des processus du modèle, il est probable que l'utilisation du mode « EXPOUT », en plus d'allonger le temps de restitution de la simulation, perturbe l'équilibre interne d'un modèle (« jitter » plus grand).

Option de namcouple : OUTPUT

Cette option permet d'utiliser la librairie OASIS comme une interface d'écriture NETCDF sur disque. Notre outil d'analyse ne peut pas traiter des champs de couplage s'ils ne sont pas reçus par un modèle. En conséquence, l'option de namcouple « OUTPUT » n'est pas compatible avec LUCIA.

Champs échangés avec des pas de temps de couplage différents

LUCIA analyse l'échange de champs entre deux modèles, y compris si les champs d'un même modèle sont échangés à des fréquences différentes.

Système couplé de + de 2 modèles (avec ou sans serveur d'IO)

LUCIA calcule les valeurs *En* et *Cn* pour un système couplé composé de 2 modèles ou

9 Les sorties complémentaires de l'analyse LUCIA permettent de se faire une idée de ces temps d'interpolations (moyennes sur l'ensemble des processus MPI d'un modèle)

plus. Attention, car il se peut que la fréquence de couplage des champs soit différente entre les différents modèles. Par exemple, dans le cas présenté en figure 4, 3 modèles et 1 serveur d'IO(xios.x) composent le système couplé. Il n'y a pas d'analyse sur le serveur d'IO (qui n'échange pas de données via OASIS). La fréquence de couplage pour tous les échanges ARPEGE/NEMO est égale à 3 heures. Pour tous les échanges ARPEGE/TRIP et TRIP/NEMO, elle est égale à 1 jour. La durée totale de la simulation étant de 4 jours, l'échantillonnage des calculs dans ARPEGE et NEMO se fait sur 3,75 jours, mais sur 3 jours seulement dans TRIP, ce qui explique que dans cet exemple, la somme *En+Cn* ne soit pas la même pour toutes les composantes n du système.

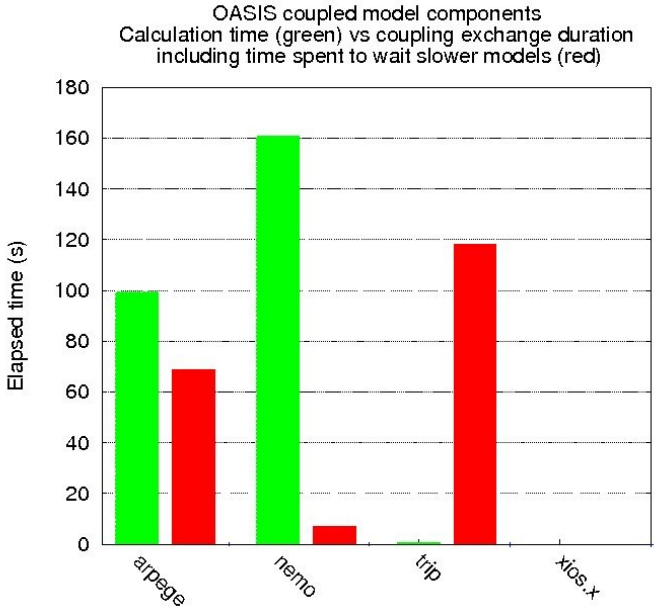

Figure 4 : Analyse d'équilibre de charge LUCIA
du modèle couplé ARPEGE-TRIP-NEMO-XIOS

2.2. Allocation d'un nombre optimal de ressources (rapidité ou scalabilité / efficacité des modèles)

Alors qu'on détermine une configuration optimale pour un nombre de ressources fixé dans la 1ère méthode, cette seconde approche consiste à trouver le nombre total de ressources nécessaires pour faire tourner le système couplé le plus rapidement possible (1er cas), ou porter le système à la scalabilité (ou efficacité) optimale pour chacun des modèles (2ème cas).

Dans les deux cas, il s'agit de faire varier le nombre de cœurs utilisés pour chaque modèle et d'obtenir un temps *Cn* correspondant à une configuration donnée pour un modèle: l'ensemble des temps *Cn* obtenus nous permettent ainsi de tracer les courbes de scalabilité et d'efficacité pour chacun des modèles. Par exemple, on peut partir d'un nombre de cœurs minimum et doubler ensuite les ressources pour chaque modèle à chaque nouvelle simulation. On s'arrête quand on atteint la limite de scalabilité de chaque modèle, c'est à dire le point à partir duquel le temps de restitution ne diminue plus malgré l'augmentation du nombre de ressources. Dans l'exemple donné en figure 5 (ressources données en échelle logarithmique), le modèle NEMO atteint cette limite aux alentours de 2.000 cœurs de calculs, tandis que pour le modèle WRF c'est autour de 16.000 cœurs de

calcul.

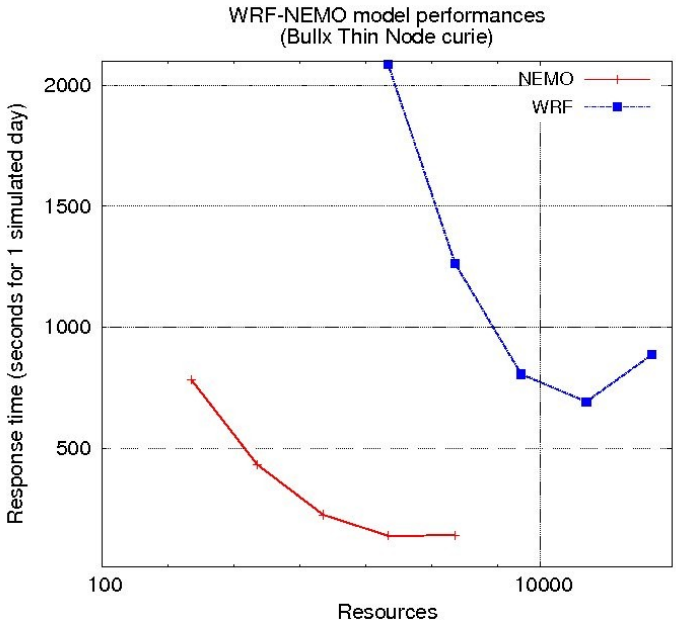

Figure 5 : Courbes de scalabilité des modèles WRF et NEMO en mode couplé

A noter que l'on mesure ici la scalabilité d'un modèle en tant que composante d'une configuration couplée particulière. Cette scalabilité peut être différente de celle obtenue en mode forcé ou bien dans une autre configuration couplée. L'intérêt de la méthode que nous proposons ici est double : elle permet d'avoir une idée des scalabilités de tous les modèles composant le système couplé en une seule analyse et la scalabilité de chaque modèle est bien celle des modèles qui seront utilisés en production dans le système couplé.

2.2.a. Rapidité

En mode séquentiel (chaque modèle attend le résultat des calculs de l'autre modèle), les courbes de scalabilité des modèles permettent de choisir pour chacun d'eux les configurations les plus rapides et ainsi obtenir les ressources nécessaires à l'ensemble du système.

En mode concurrent, la vitesse maximale du système couplé sera du même ordre que celle du modèle « le plus lent ». Une fois déterminées les limites de scalabilité de tous les modèles, on considère comme modèle le plus lent celui dont le temps de restitution à la limite de scalabilité est le plus long. On alloue à ce modèle le nombre nécessaire de ressources pour qu'il atteigne cette limite de scalabilité. On alloue ensuite aux autres modèles le nombre nécessaire de ressources pour aller aussi vite (ou légèrement plus vite[10]) que le modèle le plus lent, un nombre plus élevé de ressources pour ces composantes ne pouvant pas diminuer les temps de restitution du système couplé. Ainsi, notre système couplé peut effectuer ses calculs au maximum de sa vitesse, en utilisant le

10 Lorsque deux modèles d'un système couplé accomplissent un pas de temps de couplage à la même vitesse, les opérations de couplage en fin de pas de temps peuvent interférer entre elles et provoquer un léger ralentissement de telle sorte que la vitesse du système couplé sera légèrement inférieure à la vitesse des modèles mesurée dans une configuration couplée déséquilibrée (une configuration où une composante va plus vite ou plus lentement que les autres).

nombre strictement nécessaire de ressources pour chaque composante.

D'après la figure 5, on voit que le modèle WRF (sur 16 000 cœurs) va plus lentement que le modèle NEMO à leurs limites de scalabilités respectives. Pour aller à la même vitesse, NEMO aura besoin d'environ 512 cœurs de calcul.

2.2.b. Scalabilité / efficacité parallèle

Il arrive assez souvent que travailler à la limite de scalabilité d'un modèle conduise à un gaspillage de ressources. En effet, cette limite est généralement atteinte loin de la courbe de scalabilité parfaite : rapidement, l'augmentation du nombre alloué de ressources ne conduit pas à une accélération proportionnelle de la simulation mais à une accélération bien moindre. Il peut donc être plus économique de s'en tenir à une vitesse un peu moins élevée que celle de la limite de scalabilité,utilisant ainsi un nombre bien moins grand de ressources.

Pour fixer aisément cette nouvelle limite, on trace la dérivée des courbes de scalabilité, dite d'efficacité parallèle. Ce n'est plus le temps de restitution *T* qui est tracé en fonction du nombre de ressources *R* mais l'efficacité parallèle *E* avec :

$$E = (T_1 * R_1) / (R * T)$$

T_1 étant le temps de restitution pour le nombre minimal de ressources R_1 (idéalement, 1) avec lesquelles le modèle peut fonctionner.

Par convention, on considère comme acceptable d'allouer un nombre de ressources qui permet de ne pas abaisser l'efficacité parallèle en dessous de ½.

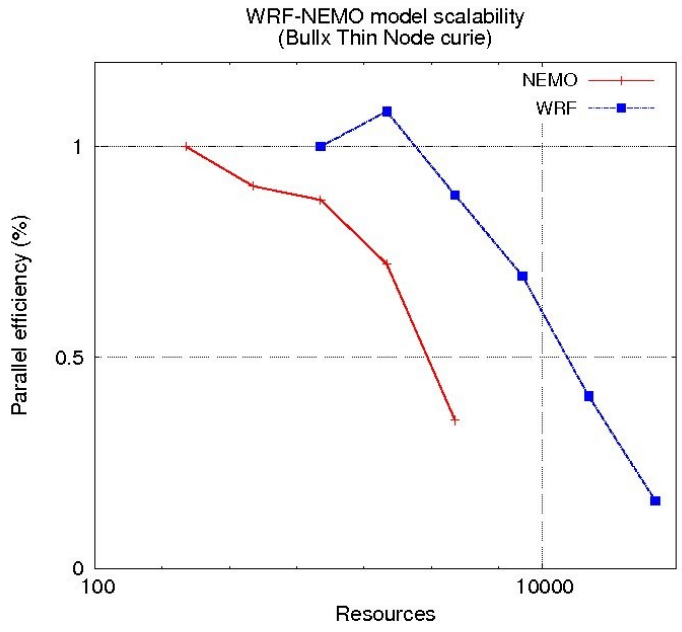

Figure 6 : Efficacité parallèle des modèles WRF et NEMO en mode couplé

Lorsque le système couplé est de type séquentiel, on utilise les courbes de scalabilité (ou

d'efficacité parallèle) pour choisir indépendamment pour chacun des modèles les ressources correspondant aux limites de scalabilité (ou d'efficacité parallèle) que l'utilisateur s'est fixées.

Dans le cas du mode concurrent, plutôt que la limite de scalabilité, on utilise le critère d'efficacité parallèle (par exemple ½). On prend pour modèle « le plus lent » celui dont le temps de restitution pour une efficacité parallèle de ½ sera le plus long et on attribue en conséquence les ressources aux différentes composantes du système couplé.

En regardant les efficacités parallèles des modèles WRF/NEMO du couplé PULSATION (cf. figure 6) on voit que la valeur de ½ est atteinte par le modèle le plus lent (WRF) après 10.000 cœurs. Pour faire avancer NEMO à la même vitesse, il ne nous faudra dans ce cas que 256 cœurs (on tire cette information de la courbe produite en figure 5).

3. Conclusion

OASIS possède désormais un outil de mesure de performance intégré (LUCIA). Il permet de délivrer une information synthétique sur les temps de calcul et d'attente des modèles les uns par rapport aux autres dans un système couplé.

LUCIA ne se substitue pas à des outils de profiling plus détaillés (de type « vampirtrace », « paraver », « vtune », etc.) qui permettent, grâce à une étude exhaustive de toutes les communications MPI du système, d'identifier plus précisément les causes d'éventuels ralentissements. En revanche, nous disposons avec LUCIA d'un outil :

- facile d'utilisation : il n'est pas nécessaire d'instrumenter les modèles couplés et une simple option de namelist suffit à déclencher les mesures.
- robuste : le mode MPMD de MPI utilisé par OASIS n'est pas toujours compatible avec l'utilisation standard des outils de profiling.
- synthétique : l'analyse rend compte de 4 quantités, intégrées sur tous les processus d'un modèle.

L'utilisateur trouvera avec LUCIA les moyens d'optimiser facilement et rapidement les performances de son système couplé, gagnant ainsi en temps de restitution et/ou en ressources. Les informations délivrées pourront également s'avérer utile pour des optimisations plus poussées, comme par exemple le placement des différents processus sur les différents cœurs et nœuds de calcul de la machine.

Annexe 1 : mode d'emploi

```
=======================================================

This tool was developped by Eric Maisonnave (CERFACS),
Uwe Fladrich, Martin Evaldsson (SMHI) and Arnaud Caubel (IPSL)
to perform an analysis of the coupled components load balance

        v1.0 :              12/2013

=======================================================

1. Compilation
--------------

In $(OASIS_DIR)/oasis3-mct/util/lucia, compile main-lucia.F90 :
* compile using the command : lucia -c
* if your compiler is not automatically detected,
  specify your compiler by modifying "F90=my_compiler" in
  "lucia" script file
* executable file "lucia.exe" is created in the same directory


2. Simulation set up
--------------------

Before launching your OASIS coupled model, modify your "namcouple"
file: the second number on the line below $NLOGPRT must be set to
-1. This option enables the production of OASIS-LUCIA log files,
named "lucia.MM.PPPPP", with "MM" executable number and "PPPPP"
MPI process number in local communicator. It is not possible to
produce timer log files at the same time that lucia log files.


3. Post processing
------------------

In your results directory (where executable and model output are
located), post-treat the files produced by OASIS-LUCIA log files:
$(OASIS_DIR)/oasis3-mct/util/lucia/lucia

This command will post-treat the OASIS-LUCIA log files located in
the directory using lucia.exe


4. Analysis
-----------

Several information related to the coupled simulation are provided
on standard output:
```

* Name and number of "lucia.MM.PPPPP" processed
For performance reasons (ASCII file reading), LUCIA do not process all OASIS-LUCIA log files, but only a subset, displayed below the comment line "Computed log files for model MM"

* Coupling field names + model exchanging them, ordered by exchange date
This information is displayed below the comment line "Exchanged fields (based on first exchange)"

* Load balance
For each model, LUCIA gives the total time spent during calculations, the total time spent to wait information from OASIS and the number of coupling time step used to calculate those values.
Time is in seconds. Information is displayed below the comment line "Component - Calculations - Waiting time (s) - # cpl step "

* Additional information
LUCIA also provides, for each model, the total time spent to perform OASIS interpolations during simulation, and total process jitter measured at each OASIS send/receive steps.
This information is displayed below the comment line "Additional informations".

Load balance information is also provided on graphical format (using gnuplot, if available) in oasis_balance.eps file.