



HAL
open science

Utilisation de EGI par la communauté des modélisateurs en systèmes complexes

Romain Reuillon, Mathieu Leclaire

► To cite this version:

Romain Reuillon, Mathieu Leclaire. Utilisation de EGI par la communauté des modélisateurs en systèmes complexes. Rencontres Scientifiques France Grilles 2011, Sep 2011, Lyon, France. hal-00653020

HAL Id: hal-00653020

<https://hal.science/hal-00653020>

Submitted on 16 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Utilisation de EGI par la communauté des modélisateurs en systèmes complexes

Romain REUILLON(1), Mathieu LECLAIRE(2)

(1) *romain.reuillon@openmole.org, Institut des Systèmes Complexes - Paris*

(2) *mathieu.leclaire@openmole.org, Institut des Systèmes Complexes - Paris*

Overview:

Abstract

The complex-system models describe “multiple levels of collective structure and organization”. Understanding these systems requires to carry out large scale numerical experiments. OpenMOLE is a generic approach to explore complex-system models. It exposes a naturally parallel representation: the workflow. It makes it possible to generate a large number of model executions according to design of experiments and to execute them on several distributed execution environment such as the grid. OpenMOLE provides the complex-system community with an experimentation tool on the models, which hides the complexity of accessing EGI.

Enjeux scientifiques, besoin de la grille :

Les modèles décrivant des systèmes complexes présentent plusieurs niveaux d'organisation au travers desquels émergent des comportements globaux (Bourguin et al, 2008). Ces systèmes sont le plus souvent non-linéaires et très difficiles à manipuler analytiquement. C'est pourquoi leur étude requiert des expérimentations numériques à grande échelle aux travers de plans d'expériences, de processus d'optimisation évolutionnaires ou d'algorithmes de viabilité, ... Toutes ces expérimentations numériques présentent des temps de calcul considérables. Toutefois l'exécution répétée du modèle constitue un aspect naturellement parallèle qui peut être mis à profit sur un environnement de calcul distribué et tout particulièrement sur grille de calcul.

Développements, déploiement sur la grille :

Accéder à la grille de calcul demande des compétences méthodologiques et techniques spécifiques. Ce frein entrave son utilisation pour de nombreux projets systèmes complexes. Il est ainsi nécessaire de mettre à disposition des outils génériques adaptés aux projets de cette communauté et permettant un accès simplifié à la grille. L'approche système-complexe met en œuvre des méthodes transverses (étude de robustesse, optimisation, caractérisation de réseaux complexes...) sur des objets d'études très divers (embryogenèse, réseaux sociaux, procédés agro-alimentaires, comportements collectifs...). Une plate-forme adaptée doit refléter cette dualité méthode-objet.

De nombreux outils permettent de décrire des processus de calcul scientifique (Kepler, Triana, Taverna, MOTEUR...). Cependant, ils ne répondent pas de manière satisfaisante aux besoins de la communauté des systèmes complexes (Reuillon et al., 2011).

C'est de ce constat qu'est née la nécessité de construire une nouvelle plate-forme. Nous avons ainsi conçu OpenMOLE¹ (Open MODeL Experiment) qui expose une description des expérimentations in silico sous forme de workflow adaptée à notre communauté. Elle permet ainsi:

- l'intégration facile de modèles de simulation (exécutables, bibliothèques java, simulation NetLogo...) dans la plate-forme,
- la description de plans d'expériences complexes,
- un passage à l'échelle transparent d'un ordinateur personnel à une grille de calcul via une approche zéro déploiement (Reuillon et al., 2010),
- une indépendance des expériences numériques vis à vis des environnements,
- la description de workflows complexes utilisant des notations implicites pour générer des milliers d'exécutions d'un modèle, des dépendances cycliques entre les tâches pour raffiner itérativement les résultats et des transitions conditionnelles pour spécifier des traitements alternatifs.

Outils, difficultés rencontrées :

La difficulté majeure rencontrée lors du développement d'OpenMOLE a été de concevoir un formalisme qui permet à la fois la spécification d'algorithmes complexes sous forme de workflows tout en permettant la délégation de la charge de calcul à divers environnements de calcul distribué. Ce travail s'appuie sur JSAGA² pour l'accès aux environnements de calcul et ajoute un niveau d'abstraction en rendant transparent la soumission de jobs, le transfert des fichiers et les installations de logiciels sur les nœud de calcul. De plus OpenMOLE utilise le standard industriel OSGi, proposant ainsi une extensibilité et une modularité avancée pour étendre la plate-forme à de nouveaux environnements de calcul, de nouveaux types de tâches, et de nouveaux plans d'expériences.

En plus des difficultés conceptuelles, le développement d'OpenMOLE nous a obligé à faire face à des obstacles techniques liés à l'utilisation de la grille de calcul européenne.

Le manque de fiabilité de ce type d'environnement largement distribué, nous a amené à développer des algorithmes auto-adaptatifs et robustes aux erreurs pour la soumission de jobs et le transfert de données. Par exemple, lors de la soumission d'un job, le choix du WMS parmi ceux disponibles dans l'organisation virtuelle (V.O.) utilisé est fonction du ratio entre le nombre de jobs soumis en cours d'exécution et le nombre de jobs en attente sur un WMS pondéré par le taux d'échec de connexion. Un poids est ainsi associée à chaque WMS. Lors de la soumission d'un job le WMS est choisi au hasard avec une probabilité fonction de son poids. Cet algorithme a été mis en place pour palier à des WMS mal configurés qui sous certaines conditions « perdent » des jobs (les jobs passent en état « queued » mais ne sont en fait pas réellement soumis à un computing element, il ne passent ainsi jamais en état « running »).

Deux mécanismes additionnels complètent cet algorithme. Le premier tue et ressoumet les jobs de manière aléatoire : un job en attente a 1/100 chance d'être tué et ressoumis toutes les 5 minutes. Ce mécanisme d'agitation couplé à l'algorithme de sélection des WMS permet de faire passer à coup sûr les jobs en état d'exécution. Cependant il existe aussi des jobs qui s'exécutent sur des machines surchargées et ne se terminent pas avant un temps

¹<http://www.openmole.org/>

²<http://grid.in2p3.fr/jsaga/>

excessivement long. L'algorithme de sur-soumission pallie à ce problème et permet d'améliorer grandement les performances de l'exécution. Lorsqu'il estime que la grille est sous-utilisée, il ressoumet des calculs déjà en cours d'exécution par d'autres jobs. Lorsqu'un job se termine tous les autres jobs travaillant sur le même calcul sont tués. L'illustration 1 représente deux exécutions de 1000 jobs d'environ 1h30 sur l'organisation virtuelle biomed. Chaque trait bleu correspond à la durée d'exécution d'un job (c'est à dire la durée écoulée pour passer de l'état « Running » à l'état « Done »). La partie gauche de la figure représente une exécution sans algorithme de sur-soumission : après 24 heures le workflow n'est pas fini, il reste 30 jobs en état « Running ». La partie droite représente une exécution avec l'algorithme de sur-soumission : les 1000 jobs sont menés à bien en environ 4h30.

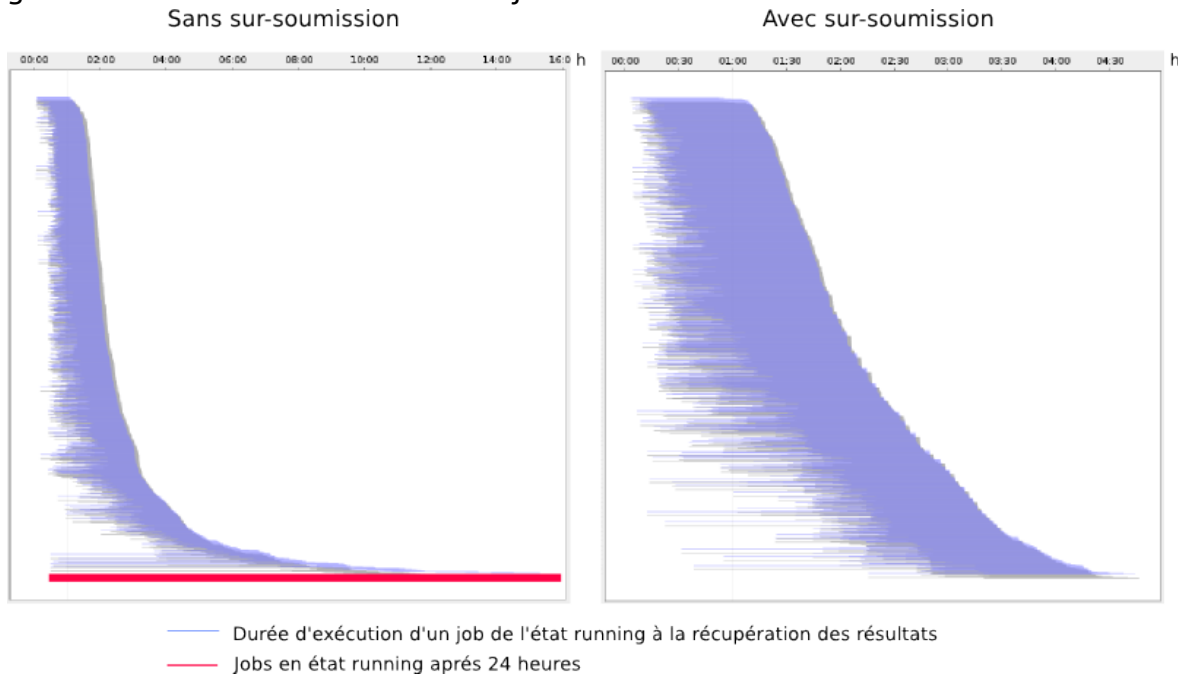


Illustration 1: Diagrammes de Gantt replantant deux exécutions de 1000 jobs sur biomed, l'une utilisant un algorithme de sur-soumission (à gauche) et l'autre non (à droite).

Ces algorithmes ont été conçus de manière empirique pour répondre à des problèmes au fur et à mesure que nous les avons rencontrés. Leur efficacité n'a pas encore fait l'objet d'étude quantitative. Cependant, de ce que nous avons pu en observer, ils font passer les chances de complétion d'une exécution d'un workflow de 0% à pratiquement 100% lors d'une utilisation classique de la grille sans recours à l'utilisation de pilot-job.

Résultats scientifiques :

OpenMOLE est une plate-forme mûre, qui permet de mettre en œuvre des algorithmes distribués de natures variées. Le workflow le plus simple rencontré en pratique permet l'exploration des paramètres d'entrée d'un modèle. L'illustration 2 montre l'interface graphique d'OpenMOLE affichant un tel workflow. La première tâche en rouge est une tâche d'exploration. Elle génère un plan d'expériences basique qui fait varier une variable i de 1 à 200 par pas de 1. La deuxième tâche en jaune utilise cette variable i comme paramètre d'entrée pour le modèle qu'elle exécute. A l'exécution du workflow, 200 instances de cette

tache sont générés. Ces instances constituent des exécutions indépendantes du modèle et peuvent être déléguées de manière entièrement déclarative à un environnement de calcul distribué (par exemple la grille de calcul européenne).

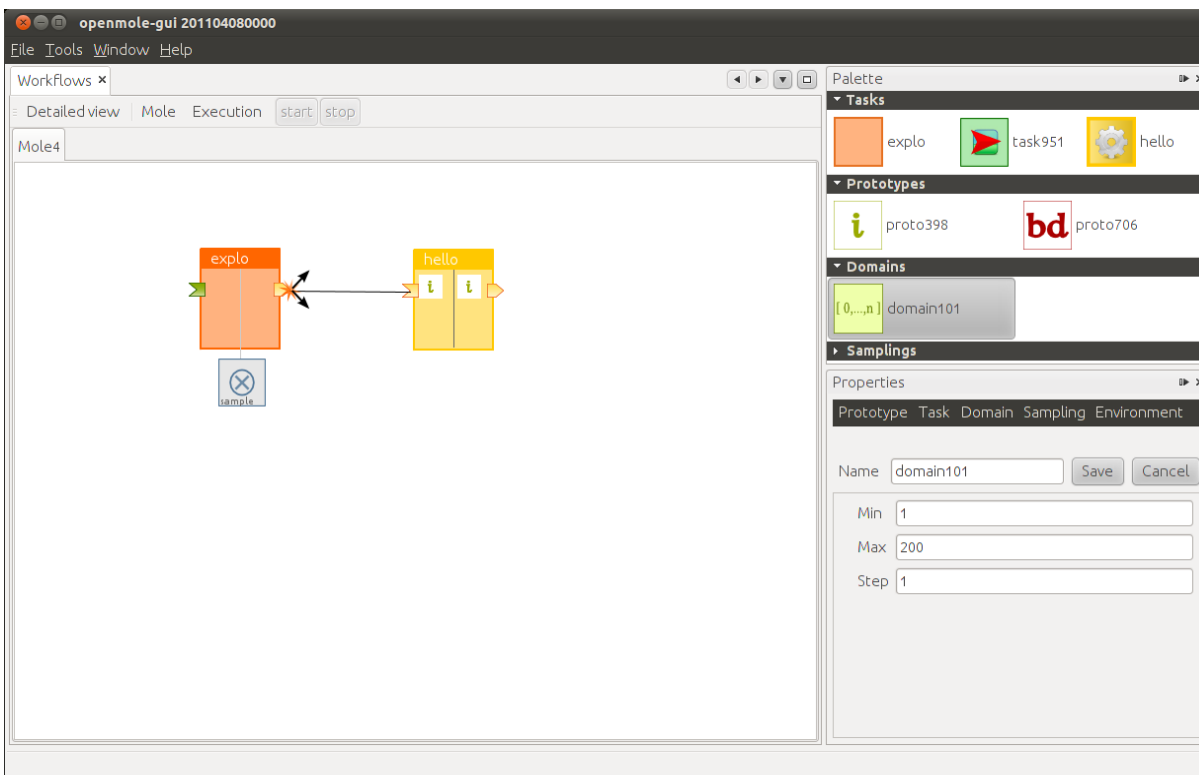


Illustration 2: Affichage d'un workflow dans l'interface graphique d'OpenMOLE

De nombreuses collaborations au sein de la communauté des systèmes complexes ont permis de faire mûrir et d'enrichir la plate-forme avec de nouvelles fonctionnalités pour répondre à des questions spécifiques sans altérer sa généralité. Voici une liste non exhaustive de nos collaborations :

- **Biologie (4):**
 - Organisation du génome, structure du chromosome – *Génoport*;
 - Étude de l'aléas dans les neurones, une approche probabiliste multi-échelle – *Institut Jaques Monod, Paris Diderot*;
 - Modélisation de la morphogénèse - *Thèse - Institut des systèmes complexes*
 - Traitement d'images de microscope pour l'étude de la morpho-dynamique des cellules du zebrafish lors de l'embryogénèse - *plateforme IBISA, Bioemergences*¹
- **Agro-alimentaire (3):**
 - Théorie de la viabilité appliquée au contrôle de procédé de fabrication: application à la fabrication du Camembert – *CEMAGREF / INRA* (Sicard et al., 2011);
 - Recherche de contrôles optimaux pour l'affinage du camembert – *CEMAGREF / INRA* (Mesmoudi et al., 2010);

¹<http://www.bioemergences.eu/>

- modélisation de la structure et la texture des gels laitiers – *INRA - GMPA (Génie et Microbiologie des Procédés Alimentaires)*;
- SHS (6)
 - Simulations multi-agent pour l'optimisation de la configuration spatiale des rames de métro - *Géographie-cité, Université Paris 1*;
 - Simulation de dynamique de croissance de peuplement, CNRS - *Géographie-cité, Université Paris 1*;
 - Simulation de mouvement de panique, *Géographie-cité, Université Paris 1*;
 - Modélisation des phénomènes de morphogenèse au sein de réseaux sociaux, *Thèse CNRS - CREA (Centre de Recherche en Épistémologie Appliquée)*;
 - Exploration de modèle d'épidémiologie - *Institut Jacques Monod, Paris Diderot*;
 - Épistémologie quantitative – calcul d'occurrence de mots clé et regroupement en clusters – *Recherches CNRS - CREA*.
- Physique (1)
 - Simulation de particules auto-propulsés - *CEA*

La puissance de calcul requises varie selon les expériences de quelque mois / CPU à 12 ans CPU distribué (de 200 à 12 000 jobs). Elles ont été exécutées sur les organisation virtuelle vo.iscpif.fr, vo.complex-systems.eu et biomed suivant la thématique, la taille du problème et la disponibilité des différentes VO. Suivant notre expérience OpenMOLE est capable de gérer l'exécution d'une centaines de milliers jobs depuis un simple ordinateur personnel avec 2Go de RAM.

Perspectives :

OpenMOLE est fonctionnel et permet à de nombreux projets d'utiliser l'infrastructure de grille de calcul. La communauté de développement est extrêmement active et a pour objectif d'atteindre le statut de plateforme de référence pour l'exploration distribuée de modèles de simulation. Pour le moment OpenMOLE propose une interface en ligne de commande, cependant une interface graphique sera proposé à partir d'Octobre 2011. Elle simplifiera encore l'utilisation de la plateforme. Parmi les autres développements à venir, un accent particulier sera mis sur:

- l'interopérabilité avec les systèmes de cloud et de cloud sur grille (StratusLab¹),
- l'interaction en ligne utilisateur-workflow,
- la mise en place de workflow génériques implémentant des méthodes génériques pour l'exploration de modèles.

Références :

P. Bourguine, P. Chavalarias, E. Perrier, F. Amblard, F. Arlabosse, et al., French roadmap for complex systems 2008-2009 arXiv:arXiv:0907.2221v1, 2008.

S. Mesmoudi, N. Perrot, R. Reuillon, P. Bourguine, E. Lutton, Optimal viable path search for a cheese ripening process using a multi-objective ea, in: ICEC 2010, International Conference on Evolutionary Computation, 2010, 24-26 oct, Valencia, Spain.

M. Sicard, N. Perrot, S. Mesmoudi, S. Martin, R. Reuillon, I. Alvarez, Development of a viability approach for reverse engineering in complex food processes: Application to a camembert

¹<http://stratuslab.eu/>

cheese ripening process. adaptation of the viability theory, Journal of Food Engineering (EFG)

submitted, 2011.

R. Reuillon, M. Leclaire, OpenMOLE: A generic approach for distributed numerical experiments on complex-system models **submitted**, 2011,

R. Reuillon, F. Chuffart, M. Leclaire, T. Faure, N. Dumoulin, D. Hill, Declarative task delegation in OpenMOLE, in: HPCS, 2010, pp. 55–62.