UNIVERSITÉ TOULOUSE III PAUL SABATIER

Millian Poquet
Maître de conférences
Univ. Toulouse III, IRIT, équipe Sepia
IRIT2-469, 120 cours Rose Dieng-Kuntz, 31400 Toulouse
millian.poquet@univ-tlse3.fr

Exécution distribuée contrôlée de workflows d'expériences

Contexte

De nombreuses campagnes d'expériences scientifiques sur ressources de calcul informatique peuvent être résumées à l'exécution d'un ensemble d'applications et au déplacement d'un ensemble de données. Les différentes exécutions et les différents transferts de données sont souvent liés par une relation de dépendance. Par conséquent, de nombreuses expériences peuvent être décrites par un workflow qui décrit ce que doit réaliser l'expérience : c'est un graphe orienté acyclique (*DAG*) dont les nœuds sont des exécutions d'applications et les arcs des mouvements de données.

Cette formalisation possède le très fort avantage de permettre d'exécuter l'expérience sur des plate-formes de calcul très différentes, puisque l'exécution du workflow est dédiée à un environnement d'exécution (*runtime*) responsable d'utiliser les ressources de calcul au mieux tout en respectant les diverses contraintes de l'expérience et de l'infrastructure de calcul. Cela peut également grandement simplifier la tolérance aux pannes d'expériences à grande échelle, puisque ce formalisme permet de réexécuter des sous-parties du workflow au besoin.

De nombreux outils ont été développés pour exécuter des workflows. Ces développements sont nés dans des communautés scientifiques ou industrielles très éloignées les unes des autres, ce qui fait que la plupart de ces outils sont incompatibles entre eux et utilisent leurs propres formats de données pour représenter le workflow. Les workflows eux-mêmes peuvent être définis de manière incompatibles (pas le même type de graphe). Les technologies sous-jacentes qu'utilisent ces outils peuvent également être très variées et avoir un jeu de contraintes différent les unes des autres. Afin d'aider à la mise en commun des différents efforts qui ont été réalisés autour de l'exécution de workflows, le langage CWL a été conçu pour servir de standard de description de workflows.

Objectifs du projet

Le but de ce projet est de réaliser une veille technologique des implémentations de *runtimes* d'exécution de workflows existantes qui supportent CWL. On s'intéressera ici au cas d'utilisation d'expériences de campagnes d'expériences scientifiques simples (génération déterministe de données d'entrée, exécution déterministe d'un ensemble d'instances de simulations déterministes, analyse déterministe de données), où l'on souhaite se servir d'une infrastructure de calcul distribuée comme Grid'5000, Jean Zay ou Calmip. On souhaitera étudier les fonctionnalités offertes par les différents *runtimes* (est-ce que l'utilisation de ressources distribuée est possible/efficace/tolérante aux pannes ?), ainsi que les propriétés de traçabilité et d'isolation/contrôle des tâches qu'elles permettent. Dans un second temps, il est probable qu'on contribue à certains des *runtimes* choisis pendant le projet pour y ajouter un support de l'exécution de tâches dans des environnements contrôlés décrits par des *Nix flakes*.