Prendre en charge des opérations chronophages depuis Shiny avec Shinybatch

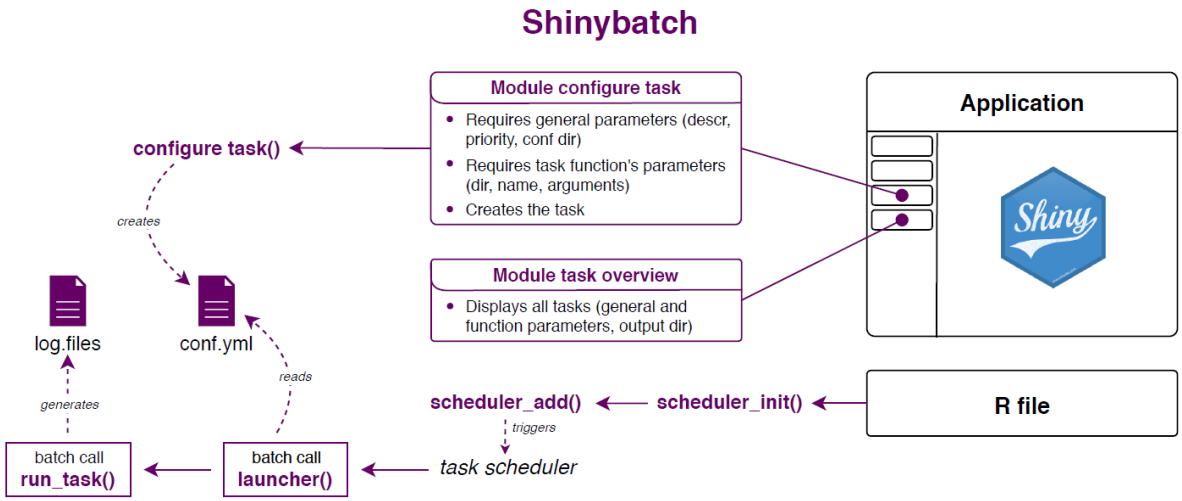
Benoît Thieurmel* Thibaut Dubois†

Résumé

Il est peu pratique de réaliser de longues opérations dans une application Shiny, car l'application est indisponible pendant tout le temps de la résolution. Pour certains usages cependant, la possibilité de lancer des taches depuis l'application est un atout.
Nous proposons une solution packagée comprenant deux modules Shiny. Le premier module permet de paramétrer des opérations qui seront ensuite lancées en tâche de fond par batch. Le second module consiste ensuite en une interface qui permet de suivre la résolution des opérations et facilite la récupération des résultats.
Mots-clefs : Shiny, opérations chronophages, monitoring

Développement

Parmi les solutions existantes pour faciliter le déclenchement d'opérations sur une application et la gestion simultanée de plusieurs sessions, on peut citer Shiny asynchrone qui repose notamment sur les packages *future* et *promises*. L'idée derrière Shiny asynchrone est d'allouer des ressources supplémentaires partagées entre les utilisateurs pour les calculs, de sorte que lorsque l'un d'entre eu déclenche une opération, l'application continue de tourner pour les autres. Cela fonctionne pour les opérations relativement courtes, en revanche si une opération doit durer plusieurs heures/jours, il n'est pas souhaitable de bloquer la session de l'utilisateur qui la déclenche pendant si longtemps. C'est là que Shinybatch entre en jeu, en déportant les calculs sur des ressources indépendantes de l'application.



Le package Shinybatch contient un ensemble de fonctions qui permettent la configuration et le déclenchement des opérations sous Windows et Linux. Le déclenchement se fait à intervalle régulier à partir du planificateur de taches, qui nécessite deux dépendances distinctes pour Windows (*taskScheduleR*) et Linux (*CronR*).

* Datastorm, Benoit.thieurmel@datastorm.fr

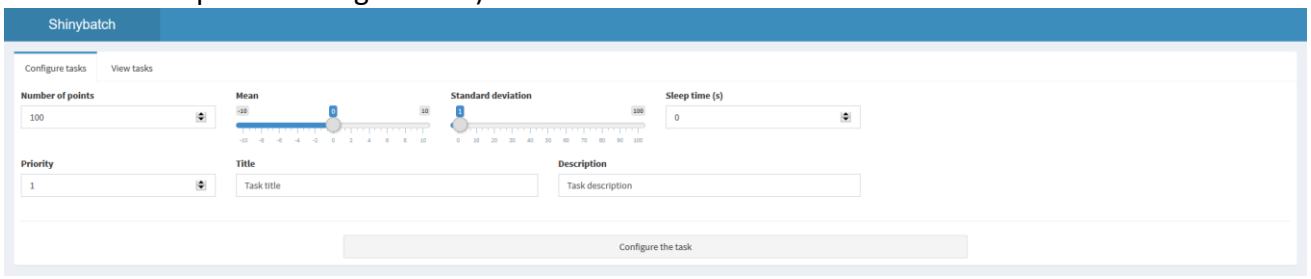
† Datastorm, Thibaut.dubois@datastorm.fr

L'ensemble des fonctions vise à faciliter au maximum la prise en main du package, qui demande seulement quelques connaissance en Shiny. En résumé, les principales étapes pour l'utiliser dans une application sont les suivantes :

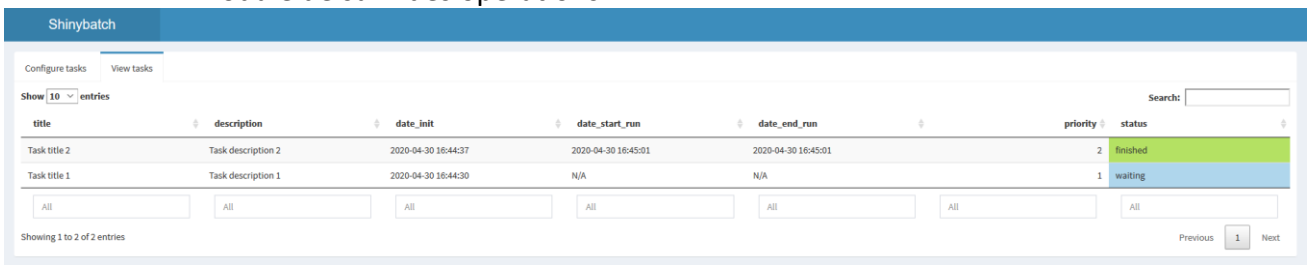
1. **Définir la tache**
    - Quels sont les paramètres de l'application (inputs) à brancher au module de configuration ?
    - Quelle est la fonction à appeler (et où la sourcer) pour réaliser la tache ?
    - Où sauvegarder les résultats (répertoire) ?
    - (opt) Quelques champs descriptifs (nom, fonction de la tache)
    - (opt) La priorité associée à la tache

2. **Configurer le Planificateur de tache**
    - Où trouver le script qui va lancer la tache (répertoire)
        i. La fonction scheduler_init() permet de le créer
    - Où trouver la configuration (répertoire, cf 1.)
    - Divers arguments
        i. Nombre de taches simultanées
        ii. Ordre de (re)lancement des taches en fonction des statuts ('running', 'finished', 'error')

3. **Lancer l'application**
    - Paramètres du module de configuration (paramètres de la fonction et paramètres généraux)



    - Module de suivi des opérations



    - En cas d'erreur pendant la résolution d'une opération, des logs sont enregistrés dans le répertoire de la configuration.

Références

Joe Cheng, RStudio. Promises: Abstractions for Promise-Based Asynchronous Programming. R package, https://cran.r-project.org/web/packages/promises/index.html.

Henrik Bengtsson. future: Unified Parallel and Distributed Processing in R for Everyone. R package, https://cran.r-project.org/web/packages/future/index.html.

Jan Wijffels, Oliver Belmans. taskscheduleR: Schedule R Scripts and Processes with the Windows Task Schedule. R package, https://cran.r-project.org/web/packages/taskscheduleR/index.html.

Jan Wijffels, Kevin Ushey. cronR: Schedule R Scripts and Processes with the 'cron' Job Scheduler. R package, https://cran.r-project.org/web/packages/cronR/index.html.

Thibaut Dubois, Benoît Thieurmel. Shinybatch: Optimize your Computation Session in Shiny, R package, https://www.datastorm.fr/optimiser-ses-temps-de-calcul-avec-shinybatch/.