M2 - Compilation Avancée

(COA 2016-2017)

TD3

Reconnaissance des appels MPI et manipulation du cfg

hugo.brunie.ocre@cea.fr
julien.jaeger@cea.fr
patrick.carribault@cea.fr

I Manipuler le CFG

Le but de cette partie est de reconnaître les appels MPI, notamment ceux spécifiés dans le fichier de définitions, et de modifier le CFG..

Vous pouvez partir des plugins du TP précédent pour réaliser les questions suivantes.

Q.1: En vous aidant de la dernière question du TP2, écrire une fonction prenant argument un statement gimple (gimple * stmt) et affichant uniquement le nom des fonctions appelées si celles-ci sont des fonctions MPI (fonctions préfixées par `"MPI_"`. Un fichier test2.c vous est fournit pour tester votre plugin.

Q.2: Dans le répertoire *CODE*, un fichier **MPI_collectives.def** vous est fourni. Ce fichier définit les fonctions MPI nous interessant dans le cadre de ce TD. Le code présent dans le fichier **plugin_tp3_2.cpp** permet de construire un *enum* avec les identifiants des collectives ainsi qu'un tableau contenant les noms de chaque fonction MPI présente dans le fichier de définition.
Servez-vous de ce tableau de nom de fonctions pour affichez uniquement le nom des fonctions MPI appelées présentes dans le fichier de définition.

Q.3: Modifier la fonction pour que celle-ci renvoie la valeur d'*enum* correspondant à la fonction MPI du fichier de définition reconnue par lánalyse de la question précédente.

Q.4: Les structures `basic_block` manipulées permettent de stocker des informations spécifiques à la passe courante. Trouver le champs correspondant dans la structure `basic_block`.

Q.5: Utiliser ce champs pour stocker l'identifiant de la fonction MPI reconnue pour chaque basic bloc. Cela fonctionne-t-il ? Créer une fonction permettant de remettre à zéro les champs `aux` de chaque basic bloc. Cela fonctionne-t-il ? Qu'en déduire ?

Q.6: Modifier lécriture du fichier graphviz pour introduire, dans le label d'un bloc, le nom de chaque fonction MPI reconnue dans ce bloc. Que remarquez-vous sur le fichier *test2.c* ?

Q.7: Écrire une fonction prenant en argument un pointeur de fonction (`function * fun`) et vérifiant si il existe au moins un bloc avec deux appels MPI reconnus.

Q.8: Pour simplifier l'analyse que nous allons réaliser dans les prochains TPs, nous souhaitons séparer les basic blocs contenant plusieurs appels MPI reconnus. La fonction *split_bloc* permet de séparer un bloc en deux "nouveaux" blocs. Trouvez la définition de cette fonction. Utiliser cette fonction pour séparer un bloc contenant plus d'un appel MPI en plusieurs blocs contenant chacun un seul appel de fonction MPI. Vous pouvez produire un nouveau fichier graphviz pour vérifier que votre split s'est bien déroulé.