Offer #2024-07921

Doctorant F/H Des Grands Modèles de Language pour la détection et la correction des erreurs dans les applications HPC

The offer description below is in French

Contract type : Fixed-term contract

Level of qualifications required : Graduate degree or equivalent

Fonction : PhD Position

About the research centre or Inria department

Le centre Inria de l'université de Bordeaux est un des neuf centres d'Inria en France et compte une vingtaine d'équipes de recherche. Le centre Inria est un acteur majeur et reconnu dans le domaine des sciences numériques. Il est au cœur d'un riche écosystème de R&D et d'innovation : PME fortement innovantes, grands groupes industriels, pôles de compétitivité, acteurs de la recherche et de l'enseignement supérieur, laboratoires d'excellence, institut de recherche technologique...

Context

Nous proposons un contrat de thèse sur une durée de 3 ans dans l'équipe Storm (https://team.inria.fr/storm/) du centre Inria de l'Université de Bordeaux.

Assignment

Afin de re□soudre les plus grands proble□mes scientifiques en un temps raisonnable, les applications sont paralle□lise□es et lance□es sur des supercalculateurs. Cependant, ces supercalculateurs sont de plus en plus complexes et puissants, ce qui entraine une e□volution des applications (ex., nouveaux algorithmes pour le passage a□ l'e□chelle, combinaison de mode□les de programmation paralle□le). Cette e□volution le□ve de nombreux de□fis de programmation et un re□el besoin d'outils et techniques pour aider les de□veloppeurs a□ utiliser au mieux les diffe□rentes machines et architectures a□ leur disposition. En effet, a□ grande e□chelle, les de□veloppeurs d'applications font face a□ de nouvelles erreurs, lie□es au paralle□lisme, souvent difficiles a□ analyser et corriger. Aujourd'hui, s'assurer que les applications paralle□les s'exe□cutent correctement devient aussi important que d'obtenir de bonnes performances.

Les grands mode□les de langage (LLMs) sont un sujet de recherche en pleine e□volution. En particulier, leurs re□cents succe□s pour ge□ne□rer du texte pertinent et re□pondre a□ des questions en font des candidats attrayants dans le domaine de la ve□rification.

Objectif:

L' objectif de cette thèse est d'exploiter et d'adapter les Grands Mode□les de Langage pour identifier et corriger les erreurs dans les programmes paralle□les. Pour cela, nous proposons d'entrai□ner des mode□les sur des ensembles de donne□es soigneusement ge□ne□re□s et e□tiquete□s gra□ce a□ une combinaison de techniques d'apprentissage et de traitement du langage naturel.

Collaboration :
La personne recrutée sera sous la direction d'Emmanuelle Saillard et Mihail Popov. Elle sera également en lien avec Pablo Oliveira (Université de Versailles) et Eric Petit (Intel).

Main activities

Le programme de recherche est découpé en 4 axes d'exploration.

Axe 1 : Création d'un jeu de données

Un jeu de données de haute qualité est une condition nécessaire pour créer des modèles précis. Pour créer notre jeu de données, nous nous ap- puierons sur deux sources complémentaires contenant des codes corrects et incorrects. Dans un premier temps, nous exploiterons la base de données git d'EasyPAP [1], une plateforme qui enseigne la programmation parallèle. Bien que limité en taille, le code soumis par les étudiants est représentatif des erreurs que font les débutants. Nous explorerons ensuite Github via son API intégré pour collecter des codes réels et plus conséquents en taille. Les projets seront sélectionnés selon les issues, pull requests et descriptions des commits. Nous récupérerons le code avant et après les commits pertinents.

Axe 2 : Labellisation

Une fois le jeu de données créé, l'étape cruciale est d'étiqueter les programmes, c'est-à-dire d'associer chaque programme avec un label (erreur présente dans le code ou corrigée). Pour cela, on utilisera des techniques de NLP. Les descriptions des commits et toute méta-information associée (e.g., CI) seront analysées avec TF-IDF (ou optionnellement des textes d'embeddings à la word2vec). Les vecteurs obtenus seront traités avec NMF [2] pour en extraire les différentes classes d'erreurs que nous étudierons. En parallèle, nous pourrons également directement utiliser des LLMs (e.g., ChatGPT) sur les commit pour les grouper. De plus, nous analyserons l'embedding des codes avant et après chaque commit [3] : les vecteurs obtenus seront clusterisés pour grouper des changements similaires. A terme, nous unifirons les deux classifications pour créer un processus de labellisation plus général.

Axe 3 : Entraînement des modèles

Nous visons deux types de modèles. Nous commencerons par créer des modèles supervisés (Code2Error) qui prennent le code source (ou une représentation du compilateur, e.g., LLVM IR) d'un programme et prédisent la catégorie d'erreur associée au programme (basée sur la labellisation). Ces modèles permettront de classer les codes incorrects et d'enrichir les descriptions des problèmes. En détail, Code2Error utilisera un embedding (e.g., ir2vec, code2vec) pour générer des vecteurs, à partir des codes, auxquels nous appliquerons un modèle supervisé (e.g., arbre de décision) pour decider du label. Les codes avant et après le commit serviront à donner à l'arbre la version incorrecte et sa correction. Nous avons déjà validé une version préliminaire (ir2vec & arbre de decision) sur 2000 codes tests dédiés pour la vérification MPI et souhaitons passer ce modèle à l'échelle sur de vrais codes.

Ensuite, nous utiliserons des LLMs (Code2Fix). Pour chaque erreur (et donc groupe de commits associés), nous entraînerons un LLM specialisé. Ce LLM recevra les codes corrects et incorrects associés à une certaine erreur. Nous utiliserons ici les codes sources (car plus utile pour l'utilisateur) et entraînerons (fine tuning) le LLM pour passer de la version erronée à la version correcte. Nous pourrons appliquer Code2Error sur un programme inconnu pour identifier le type d'erreur qu'il contient, et appeller le LLM Code2Fix associé à l'erreur pour essayer de la résoudre. Notre intuition est qu'un LLM spécialisé par erreur sera plus efficace. Enfin, on pourra explorer la granularité du code pour Code2Error & Code2Fix. De petites granularités seront faciles à gérer pour le modèle et donc pour trouver la localisation de l'erreur au moment de la correction mais pourront manquer de contexte pour traiter certaines erreurs compliquées. Ce sera un compromis à explorer.

Axe 4 : Dissémination

Les différents modèles (Code2Fix, Code2Error) seront appliqués à des projets existants pour chercher et corriger des erreurs existantes. Nous validerons également nos modèles sur des erreurs que nous aurons exclues du jeu de données pour l'apprentissage afin de mettre en avant la généralisation de notre méthode et estimer à quel point deux erreurs sont similaires (si nous pouvons prédire une erreur avec des informations provenant d'une autre erreur, il est probable qu'elles soient liées). Les experts en outils de vérification pourront utiliser cette information pour définir de nouvelles topologies d'erreurs. Enfin, nous envisageons d'étendre notre ensemble de données avec de nouveaux codes générés automatiquement par le biais des LLMs (Dataset2Code).

Références :

[1] A. Lasserre, R. Namyst, and P.-A. Wacrenier. Easypap : a framework for learning parallel programming. In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pages 276– 283, 2020.

[2] S. Heldens, P. Hijma, B. Werkhoven, J. Maassen, A. Belloum, and R. Van Nieuwpoort. The landscape of exascale research : A data-driven literature analysis. ACM Computing Surveys, 53(2) :1–43, Mar. 2020.

[3] H. Wang, G. Ye, Z. Tang, S. H. Tan, S. Huang, D. Fang, Y. Feng, L. Bian, and Z. Wang. Combining graph-based learning with automated data collection for code vulnerability detection. Trans. Info. For. Sec., 16 :1943–1958, jan 2021.

Skills

- Motivation

- Curiosite et capacite a apprendre de nouveaux concepts
- Expe rience avec l'e criture de scripts (ex., Python)
- Mai trise des bases Linux
- Des connaissances en ML est un plus

Benefits package

- Restauration subventionnée
- Transports publics remboursés partiellement
- Congés: 7 semaines de congés annuels + 10 jours de RTT (base temps plein) + possibilité d'autorisations d'absence exceptionnelle (ex : enfants malades, déménagement)
- Possibilité de télétravail et aménagement du temps de travail
- Équipements professionnels à disposition (visioconférence, prêts de matériels informatiques, etc.)
- Prestations sociales, culturelles et sportives (Association de gestion des œuvres sociales d'Inria)
- Accès à la formation professionnelle
- Sécurité sociale

Remuneration

Montant salaire brut 1e année : 2100€

Montant salaire brut 2e et 3e année : 2190€

General Information

- **Theme/Domain :** Distributed and High Performance Computing
  Scientific computing (BAP E)
- **Town/city :** Talence
- **Inria Center :** Centre Inria de l'université de Bordeaux
- **Starting date :** 2024-10-01
- **Duration of contract :** 3 years
- **Deadline to apply :** 2024-07-31

Contacts

- **Inria Team :** STORM
- **PhD Supervisor :**
  Saillard Emmanuelle / emmanuelle.saillard@inria.fr

About Inria

Inria is the French national research institute dedicated to digital science and technology. It employs 2,600 people. Its 200 agile project teams, generally run jointly with academic partners, include more than 3,500 scientists and engineers working to meet the challenges of digital technology, often at the interface with other disciplines. The Institute also employs numerous talents in over forty different professions. 900 research support staff contribute to the preparation and development of scientific and entrepreneurial projects that have a worldwide impact.

The keys to success

Le ou la canditate doit avoir un bon niveau de programmation.

De plus, la personne recrutée devra relire de la bibliographie scientifique, écrire des rapports/articles et présenter ses travaux devant la communauté. De ce fait, un bon niveau de communication en anglais sera fortement apprécié.

Warning : you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.

Instruction to apply

Si vous êtes intéressés, merci de bien vouloir candidater via le site jobs.inria avec les documents suivants :

- CV
- lettre de motivation
- notes master
- lettre de recommandation le cas échéant

Defence Security :
This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST).Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

Recruitment Policy :
As part of its diversity policy, all Inria positions are accessible to people with disabilities.