Offre n°2024-07921

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

Type de contrat : CDD

Niveau de diplôme exigé : Bac + 5 ou équivalent

Fonction : Doctorant

A propos du centre ou de la direction fonctionnelle

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...

Contexte et atouts du poste

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.

Mission confiée

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).

Principales activités

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

Axe 1 : Cre□ation d'un jeu de donne□es

Un jeu de donne□es de haute qualite□ est une condition ne□cessaire pour cre□er des mode□les pre□cis. Pour cre□er notre jeu de donne□es, nous nous ap- puierons sur deux sources comple□mentaires contenant des codes corrects et incorrects. Dans un premier temps, nous exploiterons la base de donne□es git d'EasyPAP [1], une plateforme qui enseigne la programmation paralle□le. Bien que limite□ en taille, le code soumis par les e□tudiants est repre□sentatif des erreurs que font les de□butants. Nous explorerons ensuite Github via son API inte□gre□ pour collecter des codes re□els et plus conse□quents en taille. Les projets seront se□lectionne□s selon les issues, pull requests et descriptions des commits. Nous re□cupe□rerons le code avant et apre□s les commits pertinents.

Axe 2 : Labellisation

Une fois le jeu de donne□es cre□e□, l'e□tape cruciale est d'e□tiqueter les programmes, c'est-a□-dire d'associer chaque programme avec un label (erreur pre□sente dans le code ou corrige□e). Pour cela, on utilisera des techniques de NLP. Les descriptions des commits et toute me□ta-information associe□e (e.g., CI) seront analyse□es avec TF-IDF (ou optionnellement des textes d'embeddings a□ la word2vec). Les vecteurs obtenus seront traite□s avec NMF [2] pour en extraire les diffe□rentes classes d'erreurs que nous e□tudierons. En paralle□le, nous pourrons e□galement directement utiliser des LLMs (e.g., ChatGPT) sur les commit pour les grouper. De plus, nous analyserons l'embedding des codes avant et apre□s chaque commit [3] : les vecteurs obtenus seront clusterise□s pour grouper des changements similaires. A terme, nous unifirons les deux classifications pour cre□er un processus de labellisation plus ge□ne□ral.

Axe 3 : Entrai□nement des mode□les

Nous visons deux types de mode□les. Nous commencerons par cre□er des mode□les supervise□s (Code2Error) qui prennent le code source (ou une repre□sentation du compilateur, e.g., LLVM IR) d'un programme et pre□disent la cate□gorie d'erreur associe□e au programme (base□e sur la labellisation). Ces mode□les permettront de classer les codes incorrects et d'enrichir les descriptions des proble□mes. En de□tail, Code2Error utilisera un embedding (e.g., ir2vec, code2vec) pour ge□ne□rer des vecteurs, a□ partir des codes, auxquels nous appliquerons un mode□le supervise□ (e.g., arbre de de□cision) pour decider du label. Les codes avant et apre□s le commit serviront a□ donner a□ l'arbre la version incorrecte et sa correction. Nous avons de□ja□ valide□ une version pre□liminaire (ir2vec & arbre de decision) sur 2000 codes tests de□die□s pour la ve□rification MPI et souhaitons passer ce mode□le a□ l'e□chelle sur de vrais codes.

Ensuite, nous utiliserons des LLMs (Code2Fix). Pour chaque erreur (et donc groupe de commits associe□s), nous entrai□nerons un LLM specialise□. Ce LLM recevra les codes corrects et incorrects associe□s a□ une certaine erreur. Nous utiliserons ici les codes sources (car plus utile pour l'utilisateur) et entrai□nerons (fine tuning) le LLM pour passer de la version errone□e a□ 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 associe□ a□ l'erreur pour essayer de la re□soudre. Notre intuition est qu'un LLM spe□cialise□ par erreur sera plus efficace. Enfin, on pourra explorer la granularite□ du code pour Code2Error & Code2Fix. De petites granularite□s seront faciles a□ ge□rer pour le mode□le et donc pour trouver la localisation de l'erreur au moment de la correction mais pourront manquer de contexte pour traiter certaines erreurs complique□es. Ce sera un compromis a□ explorer.

Axe 4 : Disse□mination

Les diffe□rents mode□les (Code2Fix, Code2Error) seront applique□s a□ des projets existants pour chercher et corriger des erreurs existantes. Nous validerons e□galement nos mode□les sur des erreurs que nous aurons exclues du jeu de donne□es pour l'apprentissage afin de mettre en avant la ge□ne□ralisation de notre me□thode et estimer a□ quel point deux erreurs sont similaires (si nous pouvons pre□dire une erreur avec des informations provenant d'une autre erreur, il est probable qu'elles soient lie□es). Les experts en outils de ve□rification pourront utiliser cette information pour de□finir de nouvelles topologies d'erreurs. Enfin, nous envisageons d'e□tendre notre ensemble de donne□es avec de nouveaux codes ge□ne□re□s 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.

Compétences

- 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

Avantages

- 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

Rémunération

Montant salaire brut 1e année : 2100€

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

Informations générales

- **Thème/Domaine :** Calcul distribué et à haute performance
  Calcul Scientifique (BAP E)
- **Ville :** Talence
- **Centre Inria :** Centre Inria de l'université de Bordeaux
- **Date de prise de fonction souhaitée :** 2024-10-01
- **Durée de contrat :** 3 ans
- **Date limite pour postuler :** 2024-07-31

Contacts

- **Équipe Inria :** STORM
- **Directeur de thèse :**
  Saillard Emmanuelle / emmanuelle.saillard@inria.fr

A propos d'Inria

Inria est l'institut national de recherche dédié aux sciences et technologies du numérique. Il emploie 2600 personnes. Ses 215 équipes-projets agiles, en général communes avec des partenaires académiques, impliquent plus de 3900 scientifiques pour relever les défis du numérique, souvent à l'interface d'autres disciplines. L'institut fait appel à de nombreux talents dans plus d'une quarantaine de métiers différents. 900 personnels d'appui à la recherche et à l'innovation contribuent à faire émerger et grandir des projets scientifiques ou entrepreneuriaux qui impactent le monde. Inria travaille avec de nombreuses entreprises et a accompagné la création de plus de 200 start-up. L'institut s'efforce ainsi de répondre aux enjeux de la transformation numérique de la science, de la société et de l'économie.

L'essentiel pour réussir

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.

Attention: Les candidatures doivent être déposées en ligne sur le site Inria. Le traitement des candidatures adressées par d'autres canaux n'est pas garanti.

Consignes pour postuler

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

Sécurité défense :
Ce poste est susceptible d'être affecté dans une zone à régime restrictif (ZRR), telle que définie dans le décret n°2011-1425 relatif à la protection du potentiel scientifique et technique de la nation (PPST). L'autorisation d'accès à une zone est délivrée par le chef d'établissement, après avis ministériel favorable, tel que défini dans l'arrêté du 03 juillet 2012, relatif à la PPST. Un avis ministériel défavorable pour un poste affecté dans une ZRR aurait pour conséquence l'annulation du recrutement.

Politique de recrutement :
Dans le cadre de sa politique diversité, tous les postes Inria sont accessibles aux personnes en situation de handicap.