Demande n°3199 - ARED

Acronyme :VPLDA

Identification

Identification

N°	3199
Année	2023
Statut	Brouillon
Date de création	02/02/2023
Date de modification	02/02/2023
Date de soumission	02/02/2023
Acronyme	VPLDA
Intitulé du projet	Verification pratique de logiciels distribués asynchrones
Dispositif	ARED
Civilité du porteur	
Nom du porteur	Quinson
Prénom du porteur	Martin
Courriel	martin.quinson@irisa.fr

Priorité régionale

Priorité régionale

Priorité régionale	Cybersécurité
Regional priority	Cybersécurité

Stratégie régionale de recherche et d’innovation (S3)

Stratégie régionale de recherche et d’innovation (S3)

Axe transversal	Transitions numérique et industrielle Transitions sociales et citoyennes
Domaine d'innovation stratégique	Économie numérique sécurisée et responsable
Domaine d'innovation stratégique	.
Levier thématique	Cybersécurité

Organisme de tutelle

Organisme de tutelle

Etablissement	ENS RENNES
Commune	BRUZ
Civilité	Monsieur
Nom	de Dreuzy
Prénom	Jean-Raynald

Unité de recherche

Unité de recherche

Nom de l'unité	Laboratoire IRISA
Civilité	Monsieur
Nom	Gravier
Prénom	Guillaume
Courriel	directeur@irisa.fr

Données complémentaires de l'unité de recherche

Libellé	Valeur
Statut et numéro de l'unité de recherche (UMR, USR, EA?)	UMR 6074
Le cas échéant, nom de l'équipe de recherche	MYRIADS
Le cas échéant, nom du responsable de l'équipe de recherche	Guillaume Pierre
Activités de recherche et points forts de l'équipe	Performance et correction des systèmes distribués de très grande taille type Fog/Clouds/etc
Si implication dans un/plusieurs programmes européens ou internationaux, préciser	

Co-directeur de thèse

Co-directeur de thèse

Civilité	Monsieur
Nom	Jéron
Prénom	Thierry
Unité de recherche	Laboratoire IRISA
Etablissement de tutelle	INRIA

Description

Description du projet

Libellé	Valeur
Résumé synthétique du projet (2000 caractères max)	La vérification de programmes par la technique dite de Software Model Checking est une méthode relativement facile

Libellé	Valeur
	à mettre en oeuvre pour traquer les bugs d'un programme critique. Il s'agit d'une exploration exhaustive de toutes les évolutions possibles du système à vérifier. Le gros avantage de cette méthode est qu'elle est entièrement automatisable, tandis que les approches traditionnelles basées sur un assistant de preuve demandent à l'utilisateur d'inventer des objectifs intermédiaires quand l'objectif initial est hors de portée de l'outil. Des outils construits sur l'approche de software model checking (SMC) existent pour étudier la correction de systèmes électroniques ou de programmes embarqués comme ceux qu'on trouve dans les avions, mais aucun outil de ce type n'existe à ce jour pour étudier les systèmes distribués asynchrones comme ceux qu'on retrouve dans les infrastructures numériques de grande taille comme les Clouds ou le HPC. L'objectif de ce projet est d'étendre un prototype existant dans nos laboratoires afin d'en faire un outil utilisable en pratique. À long terme, nous sommes persuadés que le SMC deviendra un outil incontournable pour les analystes de sécurité, tout comme le fuzzing actuellement, qui est une méthode comparable et complémentaire largement utilisée en cybersécurité et au delà. Les deux approches explorent de manière automatique des cas limites difficiles à tester autrement. Là où le fuzzing explore les variations des données d'entrée pour chercher des bugs pouvant donner lieu à des failles de sécurité, le SMC explore les différents ordres d'exécution des entités concurrentes. Notre outil permet déjà la démonstration automatique de propriétés sur des programmes distribués asynchrones écrits en C ou Fortran. L'objectif du travail envisagé est d'améliorer fortement la vitesse de fonctionnement de l'outil, tout en augmentant son champ d'application pour permettre son usage à d'autres types d'infrastructures numériques critiques.
Porteur du projet : date d'obtention de l'HDR ou dérogation	Martin Quinson, Professeur ENS Rennes, HDR en 2013.
Hypothèses, questions posées, points de blocage, approche méthodologique, technique (4000 car. max)	La mise au point d'outils de software model checking pose de nombreuses difficultés en pratique. La première difficulté est d'ordre algorithmique. Elle réside dans l'explosion combinatoire qui résulte de l'exploration exhaustive de l'espace d'états du système : un système formé de processus séquentiels finis s'exécutant en parallèle a un graphe d'état de taille exponentielle, voire infinie si les données du système ne sont pas bornées. Cela rend souvent impossible en pratique l'exploration de systèmes réels de taille même modérée. Depuis quelques décennies, une recherche foisonnante vise à réduire cet espace d'états, en exploitant par exemple des propriétés de redondance ou de symétrie. Parmi les méthodes proposées, DPOR (dynamic partial ordering reduction) tire parti du fait que deux événements causalement indépendants n'ont aucune influence mutuelle sur la satisfaction de propriétés. Pour la détection des blocages, il suffit par exemple d'explorer un graphe d'états réduit contenant une seule exécution par classe de comportements équivalents (par commutation d'événements adjacents indépendants). Différentes variantes de cette méthode ont été proposées dans la littérature, et nous souhaitons étudier l'efficacité de ces variantes sur des applications distribuées asynchrones réelles. Orthogonalement à la réduction de l'espace d'états, le model checking dirigé accélère la recherche de fautes dans les programmes vérifiés en changeant l'ordre d'exploration. Les transitions possibles sont explorées dans un ordre donné afin d'arriver plus rapidement aux états problématiques. Une heuristique approximant la distance entre l'état courant et l'état d'erreur à atteindre est utilisée pour le guidage. Si le système est correct, l'exploration reste exhaustive, mais si une erreur est effectivement atteignable, la performance et la longueur des contre-exemples peuvent être nettement améliorées. Cette approche a très peu été explorée dans la littérature de la vérification de systèmes réels par model checking (SMC) à ce jour. Une seconde difficulté est d'ordre technique. Contrôler finement l'exécution du système vérifié demande de combiner des techniques de virtualisation, d'interception d'exécution et d'inspection de la mémoire. Un outil de vérification dynamique

	d'inspection de la mémoire. Un outil de vérification dynamique peut être vu comme une machine virtuelle capable d'inspecter le programme comme un debugger, puis d'influer automatiquement sur l'exécution. Certaines techniques sont inspirées d'attaques de sécurité pour injecter un comportement alternatif à des fonctions classiques. Il n'existe pas à l'heure actuelle d'outil générique permettant la vérification formelle de n'importe quelle application : certains outils sont dédiés à la vérification d'applications concurrentes tandis que d'autres outils (comme Mc SimGrid à l'heure actuelle) visent plutôt les applications distribuées. L'évolution actuelle des applications vers un parallélisme mixte mélangeant MPI et programmation concurrente (MPI+X) nécessite pourtant de pouvoir vérifier les deux types d'applications à la fois. Par ailleurs, le candidat à la bourse de thèse a montré pendant son stage de M1 que toutes les solutions concurrentes de Mc SimGrid pour la vérification formelle d'applications MPI manquent de robustesse et échouent sur des codes simples sortant des cas les plus basiques. Mc SimGrid tire parti de l'environnement SimGrid, qui est le standard de facto pour l'étude par simulation des performances des applications distribuées. SimGrid offre un très bon support du standard MPI, puisque la plupart des benchmarks classiques fonctionnent dans l'environnement de simulation. Les aspects formels introduits par Mc SimGrid sont moins testés que ces aspects de simulation, mais Mc SimGrid demeure plus robuste que les outils du même genre. Les différents travaux envisagés dans cette thèse s'attacheront à renforcer cet avantage technique.
Environnement scientifique, positionnement dans contexte régional/national/international (2000 car.)	
Collaborations scientifiques (nature/partenariat/pays) et partenariat socio-économique envisagé	Ce projet s'appuie sur l'outil SimGrid, co-développé par Martin Quinson en collaboration avec des collègues français et américains depuis près de 20 ans. Le projet a bénéficié de nombreux financements d'envergure en France au niveau national (deux ANR, et de nombreux financements INRIA et CNRS) et régionaux (en Bretagne, Lorraine et Rhone-Alpes) pour un montant total de subvention dépassant les deux millions d'euros, ainsi que des financements comparables par la NSF américaine ces dernières années. Il s'agit d'un standard de facto au niveau mondial pour l'étude de systèmes informatiques distribués par simulation porté principalement par l'équipe de Martin Quinson et ses proches collaborateurs français. À l'heure actuelle les aspects formels de SimGrid sur la vérification et la recherche de bug automatisée sont moins matures que les aspects de simulation de performance qui constitue le coeur historique du projet. Ce travail de thèse vise à renforcer ces aspects formels. Le travail nécessaire pour cela s'inscrit à la croisée des travaux de l'équipe Myriads sur l'étude des infrastructures numériques distribuées et ceux de l'équipe SUMO sur la vérification formelle de systèmes informatique. Ces deux équipes de l'IRISA jouissent d'une reconnaissance nationale et internationale dans leurs domaines respectifs, et les deux co-directeurs ont l'habitude de collaborer après une thèse soutenue et plusieurs stages de niveau master co-encadrés.
Adéquation du projet avec le DIS de rattachement (3000 caractères maximum)	Il n'est pas habituel d'appliquer le software model checking (SMC) pour vérifier en pratique que les applications sont sûres, car la plupart des outils de SMC échouent à vérifier des codes réels. L'objectif de cette thèse est d'améliorer l'un des outils les plus solides au niveau mondial pour permettre la vérification d'infrastructures numériques de calcul distribuées et asynchrones (HPC voire Cloud) par cette approche, ce qui serait une première mondiale. À long terme, nous sommes persuadés que le SMC deviendra un outil incontournable pour les analystes de sécurité, tout comme le fuzzing actuellement. En effet, les avantages du SMC sont très comparables à ceux du fuzzing dans ce contexte : le SMC explore des cas limites difficiles à tester autrement, de façon entièrement automatisée sans supervision humaine, et les outils donnent des traces de rejeu prêtes à l'emploi pour chaque problème détecté. Là où le fuzzing explore les variations sur les données fournies en entrée, le SMC explore quant à lui les variations sur l'ordre d'exécution entre les entités concurrentes, ce qui augmente encore la complémentarité entre les deux approches. Les contributions techniques attendues pour la vérification d'applications MPI+X visent à rendre observables les

	opérations multithreads, de façon à permettre à l'outil d'explorer différents ordres sur ces opérations. Cela permettra à terme d'élargir le spectre des applications vérifiables pour couvrir les applications concurrentes classiques. Les contributions algorithmiques attendues dans cette thèse visent quant à elles à accélérer la vérification, ce qui est indispensable pour un usage de cette approche sur des systèmes de plus grande taille.
Si priorité régionale, préciser (200 caractères maximum)	L'un des objectifs de la thèse est de permettre la vérification en pratique d'infrastructures numériques distribuées, qui constituent une brique indispensable de l'internet du futur.
Cotutelle internationale (indiquer "oui" ou "non"; si oui, préciser le pays et l'établissement)	non
Nom(s) du(des) cofinanceur(s) envisagé(s) ou confirmé(s)	ENS Rennes
Si cofinancement industriel, justifier le choix du dispositif ARED au lieu du dispositif CIFRE	n/a
État du cofinancement (indiquer "acquis" ou "demandé")	acquis
Si demandé, date prévisionnelle de réponse du cofinanceur	n/a
Si cofinancement refusé, autre(s) source(s) de cofinancement identifiée(s)	n/a
Si reversement prévu, préciser (partenaire, montant)	n/a

Candidat

Candidat

Genre	Masculin
Nationalité	Française

Collaboration internationale

Collaboration internationale

Pays	non
Etablissement partenaire à l'étranger	

Pièces jointes

Date de mi	Type de pièce	Commentaire	Document	Valide	Commentaire
02/02/2023	ARED - Dépôt - CV du porteur	Obligatoire - Le CV doit préciser si le porteur est titulaire de l'HDR et, le cas échéant, depuis quand, et mentionner la liste des doctorant·e·s déjà encadré·e·s et des publications associées.	CV-mquinson.pdf	Non traité	
02/02/2023	ARED - Dépôt - Avis directeur·rice d'unité	15 lignes maximum		Non traité	
02/02/2023	ARED - Dépôt - Justificatif cofinancement	Facultatif - Dépôt de demande, justificatif d'obtention, courrier du cofinanceur pressenti...		Non traité	

02/02/2023	ARED - Dépôt - Lettre partenaire étranger	Facultatif - Lettre d'engagement de l'établissement partenaire étranger en cas de cotutelle internationale		Non traité	
02/02/2023	ARED - Dépôt - Pièce complémentaire	Facultatif		Non traité	
02/02/2023	ARED - Dépôt - Pièce complémentaire	Facultatif		Non traité	
02/02/2023	ARED - Instruction - Lettre de saisine	Lettre de l'établissement à transmettre à la Région (utiliser le modèle disponible dans la rubrique "Documents à télécharger")		Non traité	
02/02/2023	ARED - Versement - Convention de partenariat	Le cas échéant (si autorisation de reversement)		Non traité	
02/02/2023	ARED - Versement 1/2 - CV doctorant.e	Pour versement tranche 1 (cohortes jusqu'en 2018) Pour versement tranche 2 (cohortes à partir de 2019)	Laurent_Mathieu_CV.pdf	Non traité	
02/02/2023	ARED - Versement 1/2 - Copie du contrat doctoral (ou CDD ou attestation))	Pour versement tranche 1 (cohortes jusqu'en 2018) Pour versement tranche 2 (cohortes à partir de 2019)		Non traité	
02/02/2023	ARED - Versement 1 - Lettre d'approbation école doctorale	Pour versement tranche 1 (cohortes jusqu'en 2018)		Non traité	
02/02/2023	ARED - Versement 2 - Attestation bon déroulement travaux année 1	Pour versement tranche 2		Non traité	
02/02/2023	ARED - Versement 2 - Bilan financier année 1	Pour versement tranche 2 ARED		Non traité	
02/02/2023	ARED - Versement 3 - Attestation bon déroulement travaux année 2	Pour versement 50% tranche 3 ARED		Non traité	
02/02/2023	ARED - Versement 3 - Bilan financier année 2	Pour versement 50% tranche 3 ARED		Non traité	
02/02/2023	ARED - Versement 4 - Bilan financier année 3	Pour versement solde année 3 ARED		Non traité	
02/02/2023	ARED - Versement 4 - Certificat administratif précisant statut thèse	Pour versement du solde de la tranche 3 (à accompagner des pièces justificatives correspondantes pour les cohortes jusqu?en 2018 : avis du jury, attestation de réinscription en ED, justificatif de prolongation ou d'abandon)		Non traité	
02/02/2023	ARED - Versement 4 - Avis du jury si thèse soutenue	Pour versement solde tranche 3 ARED (cohortes jusqu'en 2018)		Non traité	
02/02/2023	ARED - Versement 4 - Attestation de réinscription en ED	Pour versement solde tranche 3 ARED (cohortes jusqu'en 2018)		Non traité	
02/02/2023	ARED - Versement 4 - Justificatif de prolongation	Pour versement solde tranche 3 ARED (cohortes jusqu'en 2018)		Non traité	
	ARED - Versement 4 - Justificatif d'abandon de thèse	Pour versement solde tranche 3 ARED (cohortes jusqu'en 2018)		Non traité	