PhD Position F/M [DOCT2024] Compiling Dynamic Languages to WebAssembly

Le descriptif de l'offre ci-dessous est en Anglais

Type de contrat : CDD

Niveau de diplôme exigé : Bac + 5 ou équivalent

Autre diplôme apprécié : Master thesis

Fonction : Doctorant

A propos du centre ou de la direction fonctionnelle

The Inria centre at Université Côte d'Azur includes 37 research teams and 8 support services. The centre's staff (about 500 people) is made up of scientists of different nationalities, engineers, technicians and administrative staff. The teams are mainly located on the university campuses of Sophia Antipolis and Nice as well as Montpellier, in close collaboration with research and higher education laboratories and establishments (Université Côte d'Azur, CNRS, INRAE, INSERM ...), but also with the regiona economic players.

With a presence in the fields of computational neuroscience and biology, data science and modeling, software engineering and certification, as well as collaborative robotics, the Inria Centre at Université Côte d'Azur is a major player in terms of scientific excellence through its results and collaborations at both European and international levels.

Contexte et atouts du poste

Research in the Inria Splits team.

Mission confiée

As most dynamic languages, JavaScript is very difficult to implement efficiently because most of its expressions have all sorts of different meanings that involve all sorts of different executions. To give a single prominent example, in JavaScript the expression `obj.prop` might denote five radically different operations. It might fetch the property `prop` from `obj`. It might scan the linked list of `obj`'s prototype chain and access the property `prop` of one of these objects along the way. It might involve calling a user defined function if `prop` is an accessor. It might involve allocating an object if `obj` is a primitive value. At last, it might involve jumping through another component of the standard runtime system if `obj` is a proxy object. No syntactic element nor type information let the execution engine discover in advance which evaluation schema to use. Checking all the possible interpretations in sequence and then execute literally, that is following the prescription of the language specification, the proper one would deliver very slow performances. All mainstream implementations use alternative strategies. Amongst all the possible interpretations, they favor the one that corresponds to the most frequent situation for which they elaborate a faster execution plan, and, as importantly, for which they elaborate a fast guard that let them decide as quickly as possible, if the faster schema would preserve the original semantics or not. Typically, that what "inline caches" and "hidden classes" achieve. Using a single test, the comparison of the object's hidden class with the inline cache, the guard checks if the offset where to read the property is known or not. If it is, the property is directly fetched using a single indexed memory read. Otherwise, the slower execution path checking all the possible situations, for instance, the situation where the property is an accessor, is used.

The common intuition is that dynamic compilers (jit) are in a better position to use dynamic heuristics-based strategies as they have the program and the data at hand where they have to decide which code to generate. We disagree with this intuition and we have shown in several publications that compiling at runtime is not a significant advantage.
because in most common situations simple guesses let static compilers, nowadays called ahead-of-time compilers (AOT), infer the most likely types and at the cost of larger codes, they can generate several versions of the program and select the one to execute using guards similar to those jit compilers would use. Based on this general idea, we have build a full-fledged JavaScript AOT compiler. It is called hopc. It is distributed as an open source software. It can be downloaded at the URL: http://hop.inria.fr.

WASM (WebAssembly) is a new popular execution platform that brings the promise of safe, fast, and portable executions. Considering the industrial effort invested in promoting WASM, it might be a central platform for tomorrow's applications.

In spite of its name, WASM is a general-purpose virtual machine, not particularly tailored for the web, although most web browsers now implement it. As of today, despite its growing popularity, only a small set of languages can be compiled to WASM, probably because until very recently, it was lacking important constructs that enables efficient implementations of high level languages. The recent inclusion of mechanisms for dealing with automatic memory management and exceptions opened new opportunities for compiling new languages to WASM. Exploring the compilation of dynamic languages to WASM is yet an unexplored domain. This is the purpose of this PhD thesis proposal.

WASM does not expose any particular mechanisms for JIT-ted executions so generating WASM code requires AOT techniques. Fortunately, most of the techniques we have developed for the static compilation of JavaScript could be applied to WASM. The starting point of the thesis will then consist in adapting the existing static compilation techniques to generate WASM code. Once this preliminary prototype built and operational, dedicated techniques for exploiting the unique mechanisms WASM provides (tail-call elimination, stack walking, memory safety, etc.) will be explored. Undoubtedly, WASM will also demand ad-hoc optimizations for good performance. Studying these new implementation techniques and optimization will be an important part of the thesis.

The thesis will take place at Inria Sophia-Antipolis, under the supervision of Manuel Serrano.

Principales activités

- Conduct research
- Development of open-source code
- Write and submit research papers.

Compétences

- Familiarities with compilation technics
- Programming skills in C and in one dynamic programming language (JavaScript prefered).

Avantages

- Subsidized meals
- Partial reimbursement of public transport costs
- Leave: 7 weeks of annual leave + 10 extra days off due to RTT (statutory reduction in working hours) + possibility of exceptional leave (sick children, moving home, etc.)
- Possibility of teleworking (after 6 months of employment) and flexible organization of working hours
- Professional equipment available (videoconferencing, loan of computer equipment, etc.)
- Social, cultural and sports events and activities
- Access to vocational training
- Social security coverage

Rémunération

Duration: 36 months
Location: Sophia Antipolis, France
Gross Salary per month: 2100€ brut per month (year 1 & 2) and 2190€ brut per month (year 3)

Informations générales

- Thème/Domaine : Architecture, langages et compilation
- Ingénierie logicielle (BAP E)
- Ville : Sophia Antipolis
- Centre Inria : Centre Inria d'Université Côte d'Azur
Date de prise de fonction souhaitée: 2024-11-01
Durée de contrat: 3 ans
Date limite pour postuler: 2024-04-28

Contacts
- Équipe Inria: SPLITS
- Directeur de thèse: Serrano Manuel / Manuel.Serrano@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.

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
Before applying, it is strongly recommended that you contact the Scientific manager beforehand.

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.