

Problème

L'analyse du comportement dynamique de programmes JavaScript ne requiert pas nécessairement un maximum de performance à l'exécution. Un compromis est donc possible pour faciliter l'instrumentation, en contrepartie d'une performance moindre.

Le développement d'une machine virtuelle (MV) et d'outils associés ne requiert pas non plus une performance maximale. Un compromis est donc possible pour faciliter le développement de la MV et d'outils associés, aussi en contrepartie d'une performance moindre. De plus, pour le développement d'outils graphiques, l'accès à un langage et des API de haut-niveau peut grandement faciliter leur réalisation.

Le développement d'optimisations de haut-niveau, comme la compilation paresseuse, l'élimination de la recherche de propriétés sur les opérations de base ou l'optimisation de la création des objets, ou la représentation des objets peuvent également être prototypés plus rapidement si le langage de destination est de haut-niveau.

Ces trois problèmes sont liés par le fait que l'implémentation d'une MV dans un langage de haut-niveau permet la réutilisation de fonctionnalités de ce langage comme les calculs à point flottant, les exceptions et la présence d'un ramasse-miette, pour faciliter ces trois tâches. De plus, l'utilisation de JavaScript comme langage de haut-niveau permet l'accès aux différents APIs des fureteurs modernes et rend la compilation du code trivial.

Objectif

L'objectif du projet est donc de faire l'implémentation d'une MV pour JavaScript en JavaScript en utilisant JavaScript comme langage de destination. À la différence du compilateur JS2JS développé par Marc, la compilation se fait vers le modèle objet développé dans le cadre du projet Photon [1]. Ce modèle objet permet aisément de faire l'intercession des opérations de base du langage pendant l'exécution du programme, ce qui n'est pas possible avec une implémentation régulière de JS ou avec JS2JS.

L'implémentation réalisée sera validée avec une tâche choisie parmi les deux catégories proposées. La catégorie outil est privilégiée, en particulier le profileur puisque le facteur limitant de Photon/Natif est la vitesse d'exécution et l'obtention d'information de profilage va permettre de cibler plus efficacement les facteurs déterminants. Cependant, une autre tâche pourraient également être choisie si les forces du stagiaire y seraient davantage mise à contribution:

- Outils
 - **Développer un profileur**
 - Développer un débogueur en JS supportant l'exécution pas-à-pas et l'inspection des trames de la pile, des variables locales et des arguments

- Développer un inspecteur
- Développer un sérialiseur vers l'assembleur
- Optimisation
 - Implémenter la compilation paresseuse, l'élimination de la recherche de propriétés et l'optimisation de la création des objets

Méthodologie

Le projet consiste d'abord à implémenter le modèle objet de Photon en JS et de fournir une implémentation des méthodes de base également en JS. Ensuite un compilateur JS vers JS (incluant les extensions au langage introduites par Photon) sera réalisé en réutilisant des éléments des différents compilateurs faits dans les dernières années ou en partant à partir de nouveaux outils (ex: Jyson). Finalement, le système sera utilisé pour réaliser une des tâches proposées dans la section objectif.

Une page web démontrant interactivement le fonctionnement du système final sera réalisée et servira à la fois de rapport et de documentation sur le projet.

Références

[1] Voir article Splash 2012