A Profiling Tool to Find Opportunities for Early Property Initialization

Course Project proposed by Marija Selakovic
Program Testing and Analysis – Winter 2015/16

Introduction

The following example illustrates the typical way of creating an array object in JavaScript. At the time of the initialization, the array is populated with two elements, numbers 1 and 2. At some point during the execution, two more elements, 2 and 3, are appended to the array object by calling the built-in `push` method.

```
var array=[0,1];
....
array.push(2);
array.push(3);
```

Imagine a situation in which the array is not changed between the initialization time and the first `push` method, i.e., no operation is performed on this array and the `push` method calls are not conditioned. In such a situation, elements 2 and 3 could be initialized at the same time as elements 0 and 1, as given below:

```
var array=[0,1,2,3];
....
```

For the given example, initializing two more elements when the array is created brings at least two benefits: (i) improved code readability, and (ii) improved performance because the computation time spent in `push` calls is now saved.

Goal

The goal of this project is to design and develop a dynamic program analysis that finds potential opportunities for early initialization of object properties and array elements (at the time of an object/array creation) that are originally added later during the program execution. The analysis will keep track of every object and array creation and track all property lookups, as well as operations that change the state of an object/array. It is also important to track whether changing object/array state is conditioned or not. Situations in which properties or elements can be initialized at the time of an object/array creation, as in the example above, should be reported by the analysis.

Tasks

More specifically, the project involves the following tasks:

- Get familiar with Jalangi[1] [1], a dynamic analysis framework for JavaScript.
- Construct a set of ten test cases that will illustrate different situations when an object/array is created and updated, but between object creation and the first update, no operation is performed on an object and the update is not conditioned.
- Design and implement a dynamic analysis that finds opportunities for early property/element initialization.

[1] https://github.com/Samsung/jalangi2

- Evaluate the analysis on the manually constructed test cases.
- Run the analysis on the Octane benchmark for node.js[2] and report all the code locations where the property initialization can be moved to the object creation.
- Manually refactor the code, as in the example above, and run the refactored version of Octane. Report whether the overall performance of the benchmark programs changes for some benchmarks by comparing the benchmark scores of the original and the refactored versions.

Reading material

[1] Koushik Sen, Swaroop Kalasapur, Tasneem Brutch, and Simon Gibbs. Jalangi: A selective record-replay and dynamic analysis framework for javascript. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2013, pages 488–498, New York, NY, USA, 2013. ACM.

[2] https://github.com/dai-shi/benchmark-octane