CSC 400 - ITGR Miniproposal
Parallel Programming Extension in Dynamic Language

Fangzhou Liu
September 12, 2017

1 Project Summary

With the growing popularity of data mining and machine learning, the demand for faster programs grows simultaneously. However, adding cores on processors almost reach its boundary, the fundamental for concurrency now turns towards software level[1], which introducing the term Parallel Programming. This raise another problem that current parallel programming language is either totally new, like OpenMP, or well support for C/C++ only, which is quite user unfriendly to new programmers. Meanwhile, dynamic languages, like Python and Ruby, though recently becomes an first choice programming language taught for newly introduced students in school because of its high modularity feature and simple grammar rules, is hard to parallelize because of the complexity of the interpreter and dynamic nature of script code[2]. This project tries to integrate the parallel programming feature into the dynamic language, here is Ruby.

Chen et al. ‘s research built an prototype interface for Ruby called Ruby BOP(Behavior of Parallelization), which running the loop speculatively, that is executing the code in parallel by default and re-run it sequentially if conflict detected. It introduced two blocks $PPR$ and $ORD$[2], which $PPR$ contains the code running in parallel, running in different process, and $ORD$ does the dependence detection using a write and read set. Another process called $UNDERSTUDY$ is introduced to do the correctness checking.

```ruby
PPR1 { 
  a = 1 #write a
}
PPR2 { 
  b = a + 1 #read a
}
```

Listing 1: Ruby BOP Example code
Listing 1 shows the example of how current Ruby BOP works. PPR1 and PPR2 represent two blocks running in parallel. In PPR1, there is a write of a and in PPR2 a read. If we run this code, PPR2 will be abort thus there is an dependence between two PPRs.

However, this research has its limitations, the architecture of Ruby BOP only guarantees the semantics for parallelization execution in arrays only, when it comes to other more complex data structure like HashMap, this framework does not works. So, our project aims to do the full extension of Ruby BOP, which support parallelization for all kind of common data structures. With this language extension, learning parallel programming becomes trivial and it provides the best tools for parallel programming education.

2 Project Description

2.1 Research Plan

- First, go through the code of Ruby BOP and cbop, a process-based speculative parallel programming library in C, understands how Ruby BOP works, its memory allocation methods for and garbage collection methods. Identify the function for each API given in cbop.

- Design the advanced version of parallel behavior module for Ruby, letting it support more complex data structures. And modify the Ruby interpreter based on newly designed memory allocation and garbage collection mechanism.

- Do performance analysis and semantics verification for each given benchmarks by comparing the run-time and result for the same piece of code running in both sequential order and parallel respectively.

2.2 Intellectual Merit

This project tries to add Parallel Programming Module into Ruby interpreter using speculative way and verify its correctness and performance by running specific benchmarks. Further inspection of this project may help find better solutions for parallelization mechanism within the modern compiler or interpreters and improve the executing rate for programs.

2.3 Broader Impact

The outcome of this project can be used in education level mostly. This tools can help students or new programmers, who does not quite familiar
with C/C++, learn the features of parallel programming quickly as well as help them build the clear understanding of parallel programming. As in the near future, these students equipped with strong parallel programming skills, will finally walk into the industry, this research can definitely help improve the software performance for both academic and industrial area, and bring valuable profit to the entire country.

References


