Nested Scribble

BENITO ECHARREN SERRANO, Imperial College London, UK
NOBUKO YOSHIDA, Imperial College London, UK
DAVID CASTRO-PÉREZ, Imperial College London, UK

Abstract

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Benito Echarren Serrano, Nobuko Yoshida, and David Castro-Pérez. 2018. Nested Scribble. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Introduction

2 EVALUATION

We evaluate our framework in terms of it's runtime performance (§ 2.1), using a machine with [...], running [...] and Go version `go1.14.6`.

2.1 Run-time Overheads of Generated APIs

We evaluate the overheads introduced by our framework during session execution due to the function calls to the user-defined callbacks which are interleaved in the implementation and the overheads associated with the process of sending and receiving invitations. We present seven benchmarks to measure these overheads in the implementation of protocols with different communication patterns and workloads.

We implement each benchmark by two methods. **(1) Scribble-Go**: we specify the communication interactions of each program as a protocol in our extended Scribble framework and implement the behaviour of the protocol through the generated callbacks. **(2) Go base cases**: We compare the performance of each Scribble-Go program against a handwritten Go implementation designed to carry out the same behaviour. Each benchmark kind is parametrised by a parameter to be able to observe the difference in performance for increasing input sizes.

We measure the *execution time* from the start of a session (before the goroutines and the channels have been created) until the end of the session, when the result of the protocol has been returned. Since the execution time of a single instance of a benchmark can be very small (in the order of microseconds), we repeatedly run the benchmark over various iterations. The number of iterations is not fixed, rather, we set a minimum of 20 iterations and a minimum of 10 seconds that the benchmark must execute for. To ensure the results are reliable, we also monitor the standard deviation of the execution times. We require that the standard deviation is less than 5% of the

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

average execution time. However, when the execution times are very short, the standard deviation may not converge to such a small number, so we relax this restriction by 1% for every 1000 iterations that the benchmark has executed.