Fast PokeEMU

Scaling Generated Instruction Tests Using State Chaining

University of Minnesota
Driven to Discover®

Department of Computer Science and Engineering

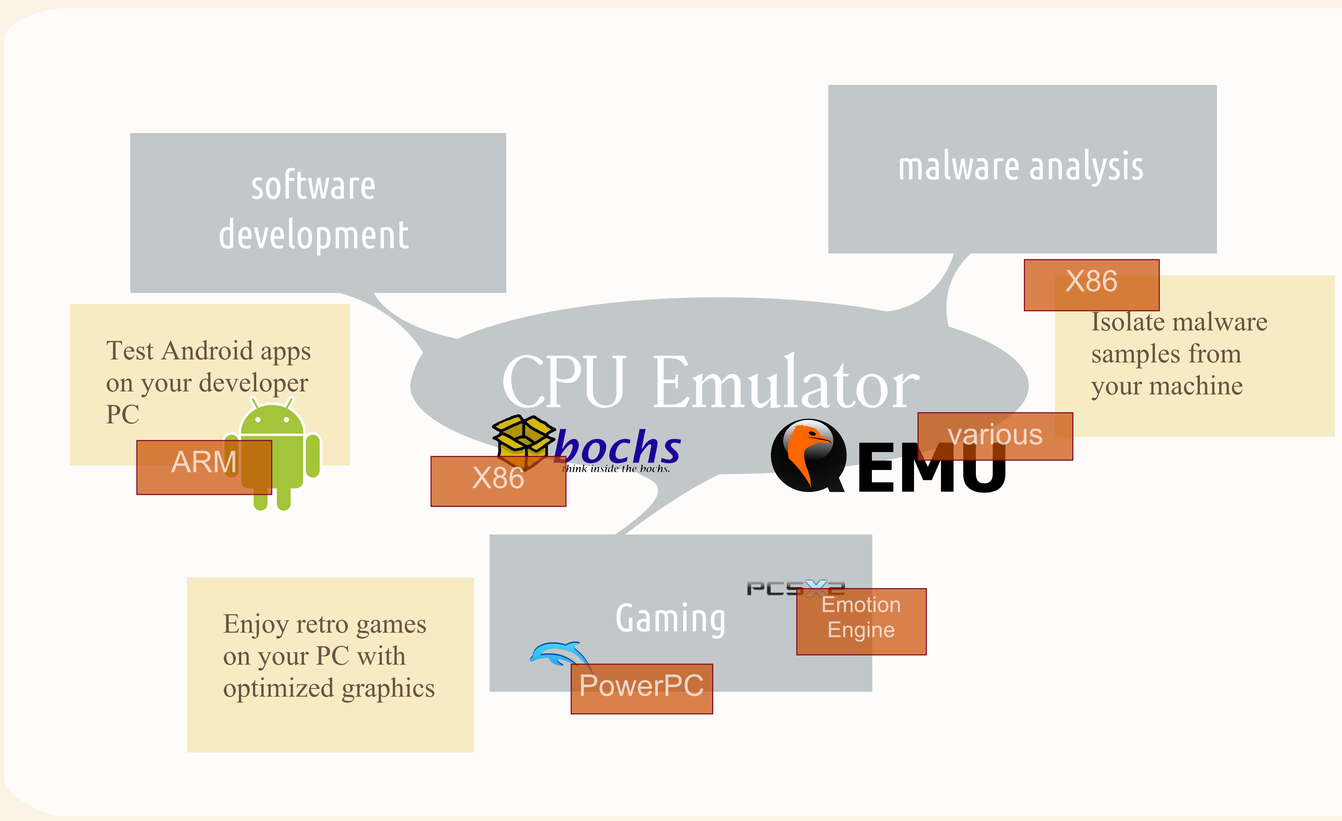
Qiuchen Yan <yanxx297@umn.edu>

Stephen McCamant <mccamant@cs.umn.edu>

Automatic Emulator Testing

Our motivation

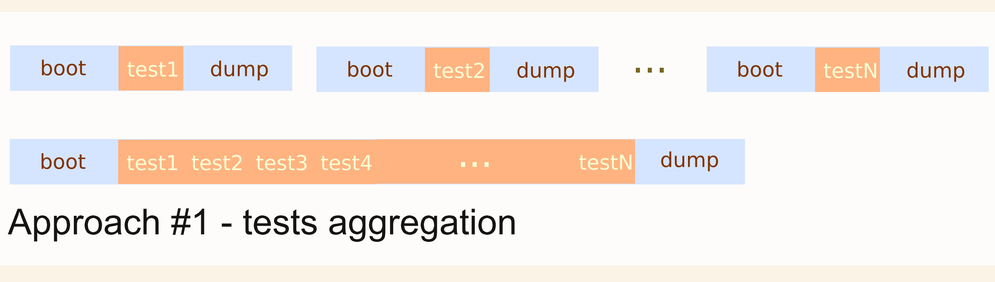
- Emulators are widely used but hard to develop
- Automate emulator testing to save more human efforts for actual development
- PokeEMU (previous auto-testing tool) limited in performance



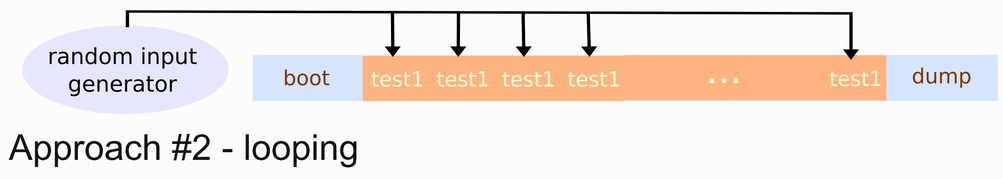
Emulators used in various fields

PokeEMU Made Faster

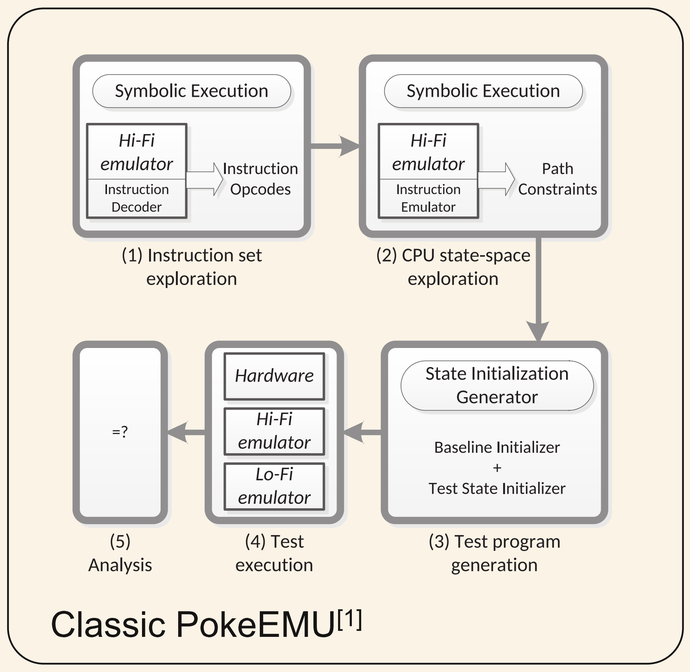
Approaches we took to improve the classic PokeEMU



Approach #1 - tests aggregation



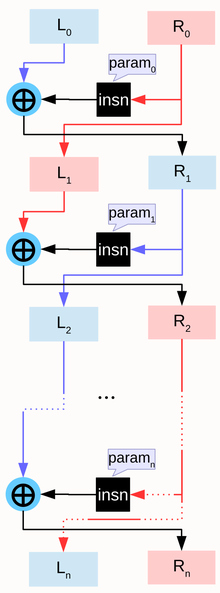
Approach #2 - looping



Classic PokeEMU[1]

Approach #3

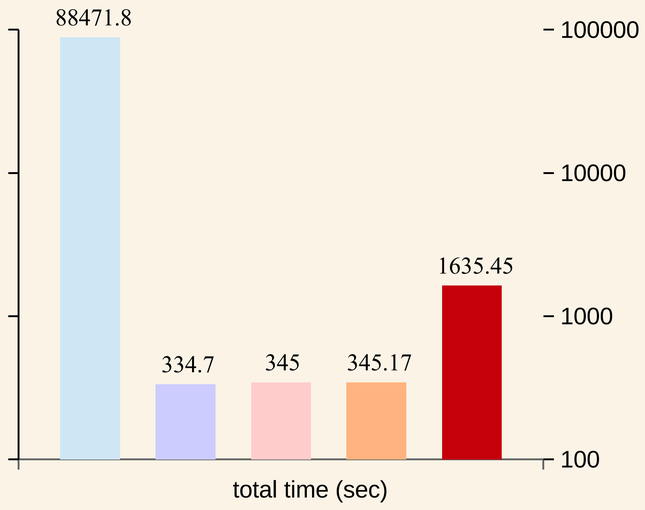
Feistel construction for memory saving

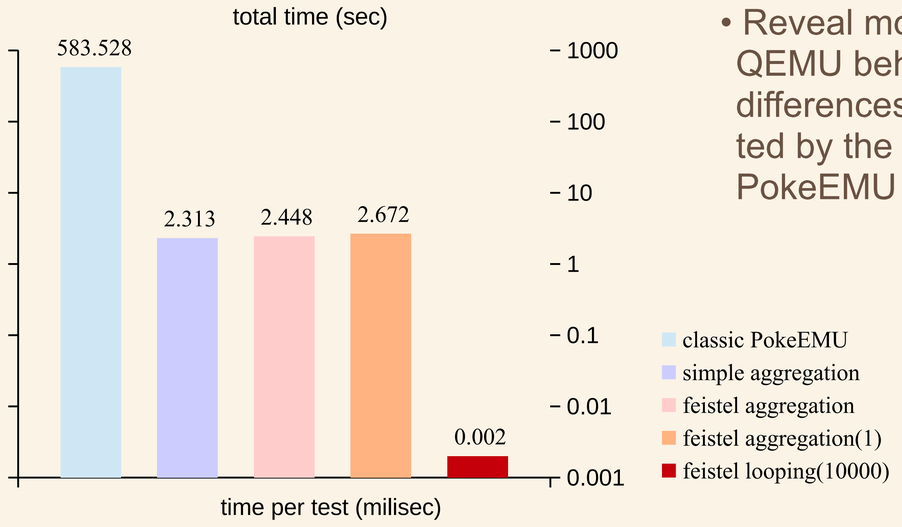


Classic vs. Fast PokeEMU

Experiment results

Experiment #1 - Performance





- Significantly increases the performance of PokeEMU when aggregation turned on
- Further decrease the average time for each test if we rerun tests for a large number of times
- Reveal most QEMU behavior differences detected by the classic PokeEMU

Experiment #2 - Effectiveness

	Separated result	Separated result with extra code	Aggregated result	# of instructions
1	Match	Match	Match	577
2	Match	Match	Mismatch	8
3	Match	Mismatch	Match	10
4	Match	Mismatch	Mismatch	28
5	Mismatch	Match	Match	25
6	Mismatch	Match	Mismatch	28
7	Mismatch	Mismatch	Match	9
8	Mismatch	Mismatch	Mismatch	273

Reference

1. Path-exploration lifting: hi-fi tests for lo-fi emulators. Lorenzo Martignoni, Stephen McCamant, Pongsin Poosankam, Dawn Song, and Petros Maniatis. 2012. In Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XVII)