AArch64 Support for llvm-exegesis

Lakshay Kumar, Rahul Shinde, Sjoerd Meijer

Is AArch64 (fully) supported?

Our First Commit to llvm-exegesis

commit 6720ce75f61a306a3ed26b2205f09a7099e978e7

Author: Sjoerd Meijer <smeijer@nvidia.com>

Date: Thu Nov 7 10:48:52 2024 +0000

[Docs][llvm-exegesis] Clarify AArch64 support (#114989)

Claiming AArch64 support for llvm-exegesis is a bit of a stretch in my opinion as only a couple of opcodes with GPR64 operands will work for snippet benchmarking, so I propose to clarify that AArch64 support is very experimental. Also added some clarifications about its libpfm4 dependency.

llvm-exegesis

- A benchmarking tool, a test case generator, to measure instruction latency and throughput:
  - Generates a test case, compiles it, runs it, and evaluates different metrics

```
mode:          inverse_throughput
key:
  instructions:
    - 'ADDVv16i8v B8 Q5'
  config:        ''
  register_initial_values:
    - 'Q5=0x0'
cpu_name:       neoverse-v2
llvm_triple:    aarch64-unknown-linux-gnu
min_instructions: 10000
measurements:
  - { key: inverse_throughput, value: 1.3749, per_snippet_value: 1.3749, validation_counters: {} }
```

- Why are we interested?
  - Software Optimisation Guide advertises best case numbers: compare measured vs. advertised numbers
  - Correlation of simulators: run exegesis or test cases within simulation environment, compare with SWOG / HW
  - Longer term: can it help with auto-generating scheduler models?

AArch64 Support ”BEFORE”

-mode=latency

#6045: Total Opcodes

#112 : Working out of the box

#2825 : Working with warnings

#3098 : Errors, not running

- [1339] Uninitialized operands by the snippet generator
- [921] isPseudo/usesCustomInserter
- [607] Segmentation fault
- [307] No serial execution strategy
- [15] Illegal instruction
- [15] isBranch/isIndirectBranch
- [13] isCall/isReturn
- [18] Targets with target-specific operands should implement

AFTER

-mode=latency

#6045: Total Opcodes

#4297 : Working

#370 : No strategy found to make the execution serial

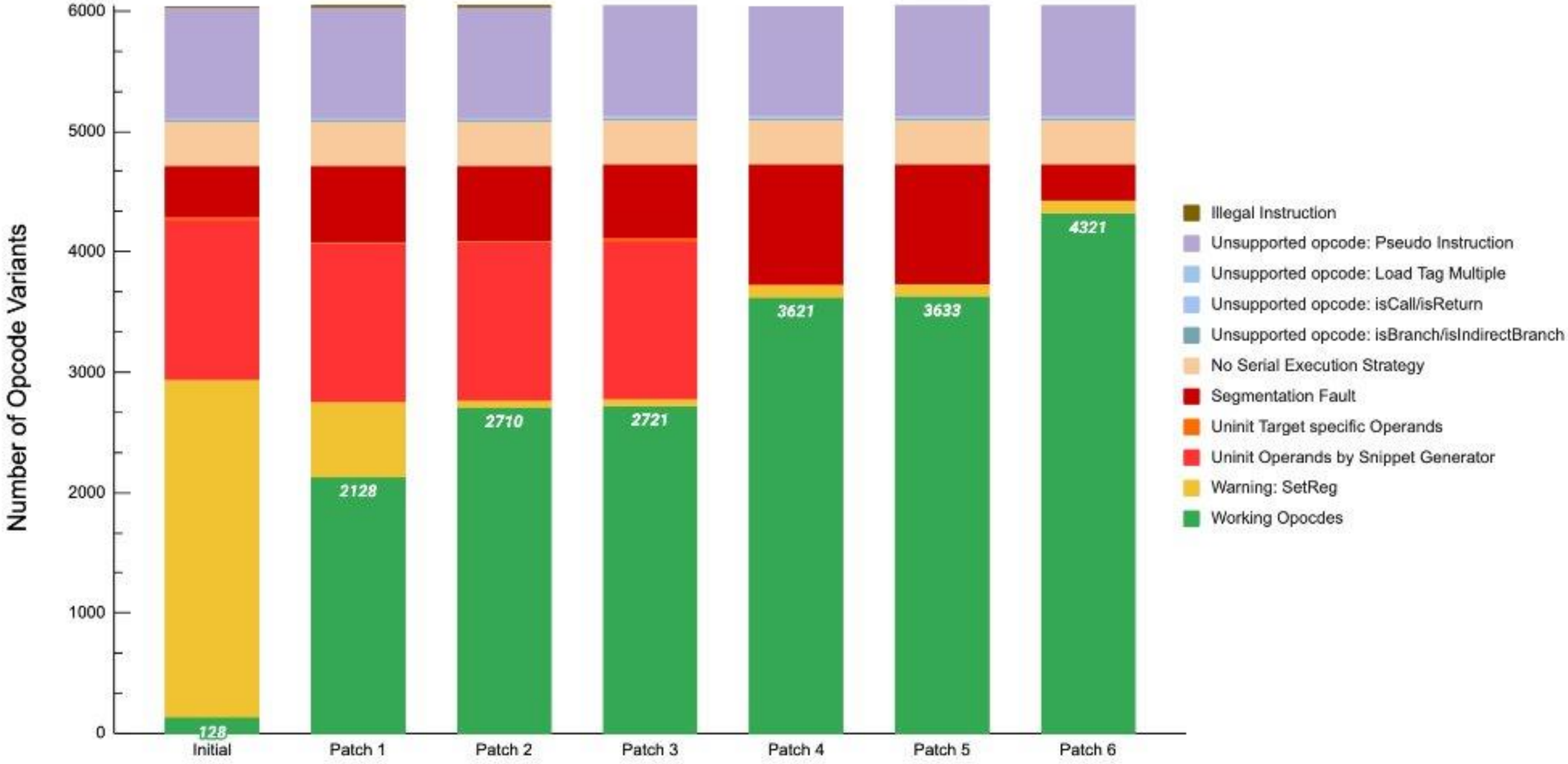
#405 : Segmentation Fault (adr 0 (#386) ffffffc0000 (#14))

#919 : Unsupported opcode: Pseudo Instruction

#15 : Unsupported opcode: isBranch/isIndirectBranch

#13 : Unsupported opcode: isCall/isReturn

Progress we made



Contributions, Next Steps and Conclusions

- Some of our contributions include:
  - Disabling instructions that cannot be easily measured: avoids lots of crashes.
  - Quite some work on initialization code: remove warnings, make results reliable.
  - Features to print the snippets: useful for debugging snippets
  - Added support for loop mode:
  - Various other smaller fixes

- Currently working on load / store instructions:
  - Proven to be quite difficult:
    - Understanding the flow, as quite some setup code is required,
    - And there are quite a few X86 assumptions here and there.

- We have only looked at latency, not so much yet at throughput.

- We ❤️ llvm-exegesis
  - An easy to use tool to measure instruction characteristics is (surprisingly) powerful
  - Thank you contributors, and thank you reviewers!