

Binary Analysis for Missed Vectorization Opportunities Detection

AMOS HERZ and ALESSANDRO LEGNANI, ETH Zürich, Switzerland

ACM Reference Format:

Amos Herz and Alessandro Legnani. 2024. Binary Analysis for Missed Vectorization Opportunities Detection. 1, 1 (March 2024), 1 page. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Modern compilers are often able to automatically vectorize code using SIMD instructions. Take, as an example, the following code snippet:

```
1 void copy(long *restrict a, long *restrict b, unsigned long n) {
2     for (unsigned long i = 0ul; i < n; i++) {
3         a[i] = b[i];
4     }
5 }
```

We can compile it with the following set of compiler flags

- `-O3`: tells the compiler to use the highest level of optimization available.
- `-fno-tree-loop-distribute-patterns`: prevents replacing the loop with a call to `memcpy`
- `-fno-tree-vectorize`: prevents vectorization

Which will produce the following assembly code:

```
1 .L3:
2     movq    (%rsi,%rax,8), %rcx
3     movq    %rcx, (%rdi,%rax,8)
4     addq    $1, %rax
5     cmpq    %rax, %rdx
6     jne     .L3
```

However, by compiling without the `-fno-tree-vectorize` flag, the compiler will produce the following vectorized code (note the use of wider instructions and registers):

```
1 .L4:
2     movdqu  (%rsi,%rax), %xmm0
3     movups  %xmm0, (%rdi,%rax)
4     addq    $16, %rax
5     cmpq    %rcx, %rax
6     jne     .L4
```

REFERENCES

Authors' address: Amos Herz, amherz@ethz.ch; Alessandro Legnani, alegnani@ethz.ch, ETH Zürich, Zürich, Switzerland, 8092.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM XXXX-XXXX/2024/3-ART

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>