Python Bindings for IR2Vec

Nishant Sachdeva
S. VenkataKeerthy

Prof. Ramakrishna Upadrasta
Scalable Compilers for Heterogeneous Architectures Group, IIT Hyderabad

EuroLLVM 2026

Background

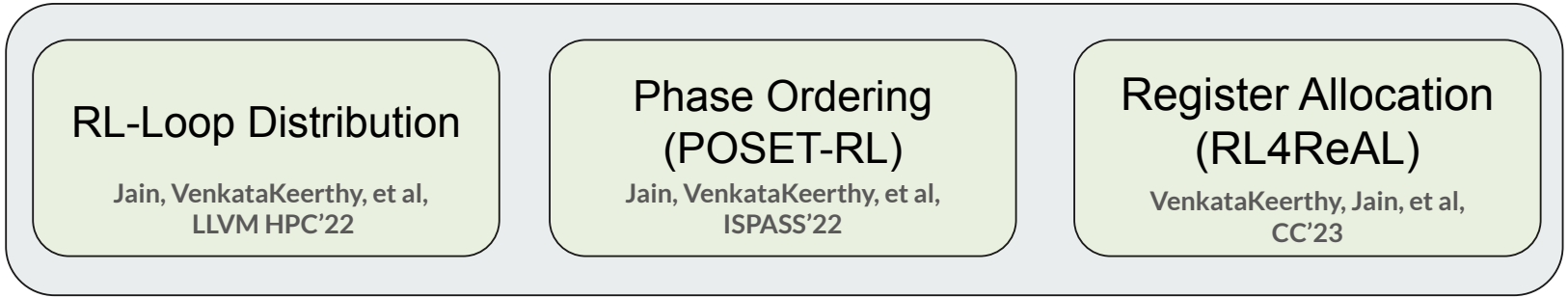
- ML guided optimizations in LLVM
- MLGO - Inlining for size, Register Allocation

Features	Embeddings
Handcrafted (counts, depths, CFG metrics)	Learned automatically
Task-specific	General-purpose, reusable

- https://llvm.org/docs/MLGO.html

IR2Vec

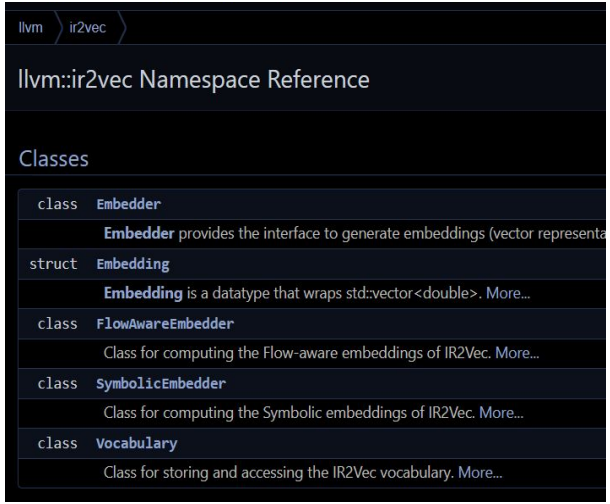
- LLVM IR based embeddings
- IR2Vec with MLGO for inlining yields up to 5% code size reduction
  - over -Os *
- Shown effectiveness on different ML-driven optimizations



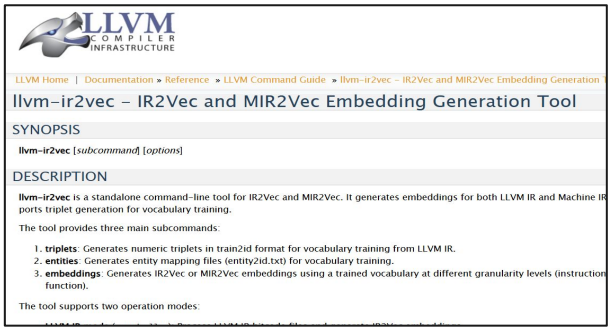
- https://discourse.llvm.org/t/rfc-enhancing-mlgo-inlining-with-ir2vec-embeddings/86250

Current ecosystem

- IR2Vec in `llvm/Analysis`
- MIR2Vec in `llvm/CodeGen`
- `llvm-ir2vec` standalone tool in `llvm/tools`
  - Supports embedding generation




```
proc = subprocess.run([
        "llvm-ir2vec", "embeddings",
        "--ir2vec-vocab-path=vocab.json",
        "--ir2vec-kind=flow-aware",
        ll_file
    ], capture_output=True, text=True
 )
embeddings = parse_ir2vec_output(proc.stdout)
```



Proposed `ir2vec` python package

- Uses `nanobind`
- Available upstream
- Currently, wheels built and published downstream to testPypi as `llvm-ir2vec`

```
pip install -i https://test.pypi.org/simple/ llvm-ir2vec
```

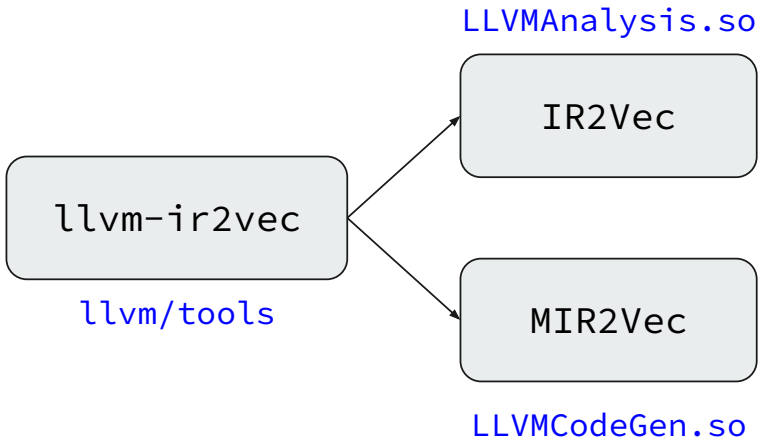
- Wheels come bundled with pre-trained vocabulary files
  - 75D, 100D, 300D

`ir2vec` Python API

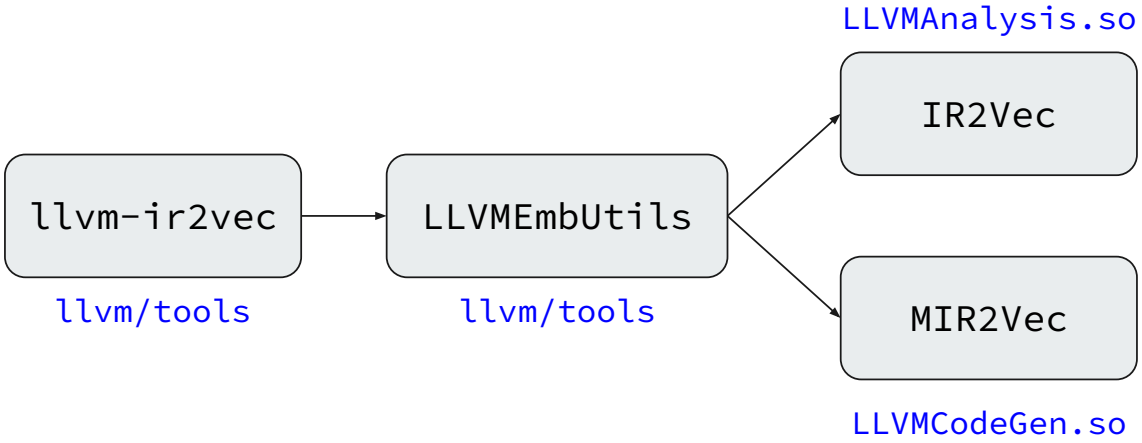
```
import ir2vec

vocabObj = ir2vec.loadVocab(
    ir2vec.vocab.seedEmbedding75D
)

emb = ir2vec.initEmbedding (
    filename="file_path.ll",
    mode=ir2vec.IR2VecKind.FlowAware,
    vocab=vocabObj
)
```

```
func_names = emb.getFuncNames()
func_emb_map = emb.getFuncEmbMap()

# for an IR function "foo"
func_emb = emb.getFuncEmb("foo")
bb_map = emb.getBBEmbMap("foo")
inst_map = emb.getInstEmbMap("foo")
```

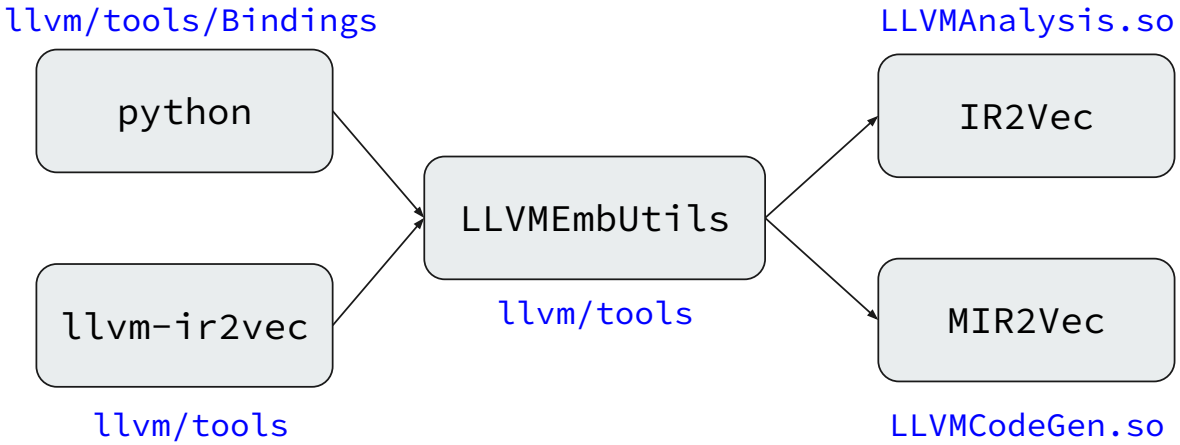
Implementation specifics - Python Bindings



Implementation specifics - Python Bindings



Implementation specifics - Python Bindings



Wins ? - Performance

- Scalability stress test
  - Given a dataset
  - Collect function embedding maps and record the time-taken

```
@timed()
def benchmark(ll_files):
    . . .
    for ll_file in ll_files:
        proc = subprocess.run(...)
        embeddings = parse_output(proc.stdout)
        embedding_dict[ll_file] = embeddings
```

llvm-ir2vec

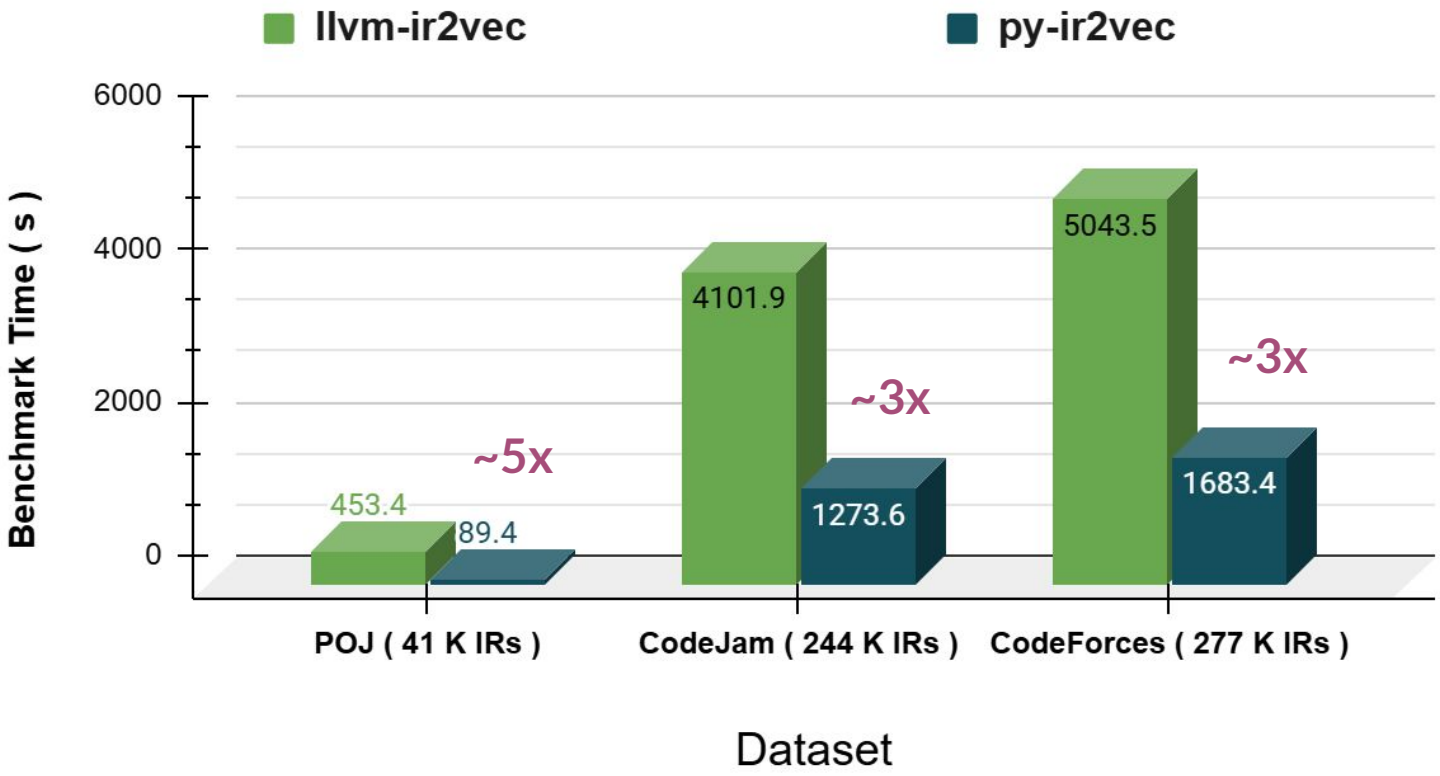
Wins ? - Performance

- Scalability stress test
  - Given a dataset
  - Collect function embedding maps and record the time-taken

```
@timed()
def benchmark(ll_files):
    . . .
    vocabObj = ir2vec.loadVocab(...)
    for ll_file in ll_files:
        tool = ir2vec.initEmbedding(filename, mode, vocab)
        embeddings = tool.getFuncEmbMap()
        embedding_dict[ll_file] = embeddings
```

py-ir2vec

Benchmark Function Embedding collection from a dataset



Pain points addressed

- Subprocesses removed
- Output parsing bypassed
- Easy Error Handling
- Speed enhancements

Next Steps

- Currently, a downstream repo builds and publishes `llvm-ir2vec` wheels to TestPyPi
- Distribution
  - `ir2vec` wheels built and published as part of LLVM
    - No separate repo, No maintenance drift

Artifacts for this work are available at

- Technical Report - HERE
- Colab Demo - HERE

Acknowledgements

Scalable Compilers for Heterogeneous Architectures Group, IIT Hyderabad

- S. VenkataKeerthy
- Prof. Ramakrishna Upadrasta

Thank You