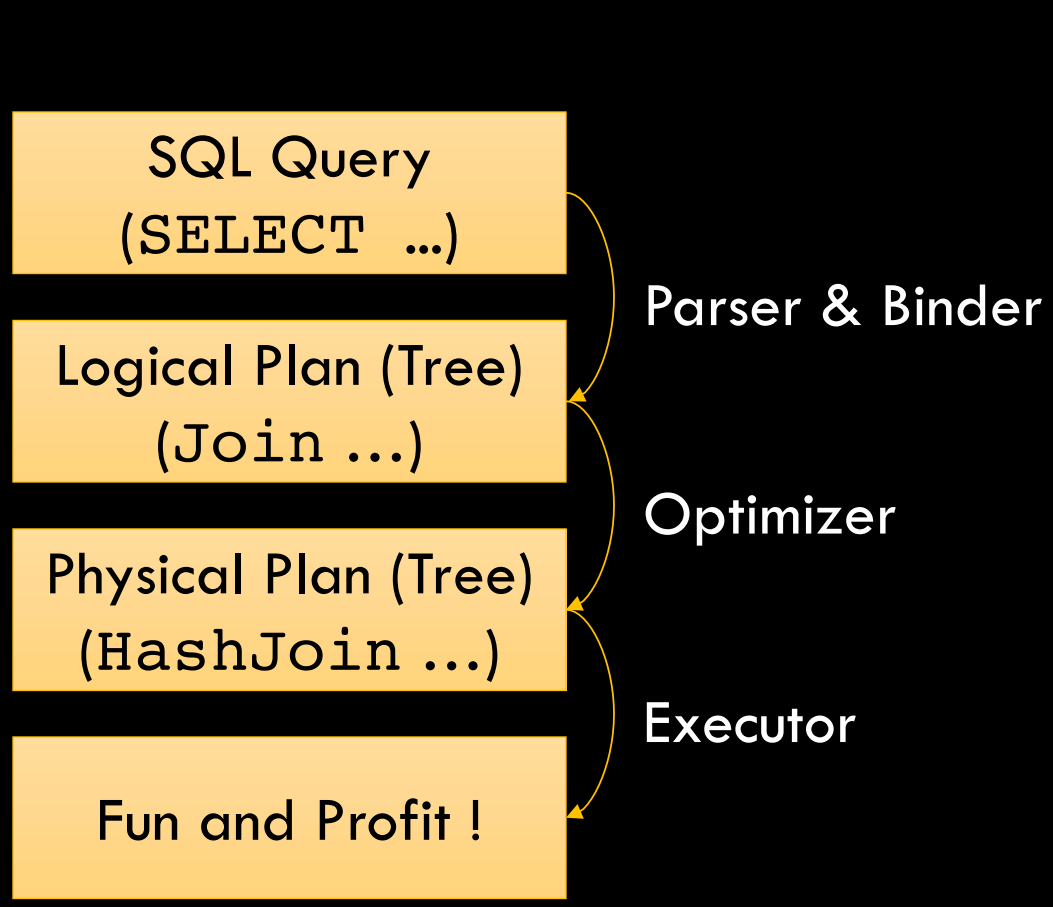
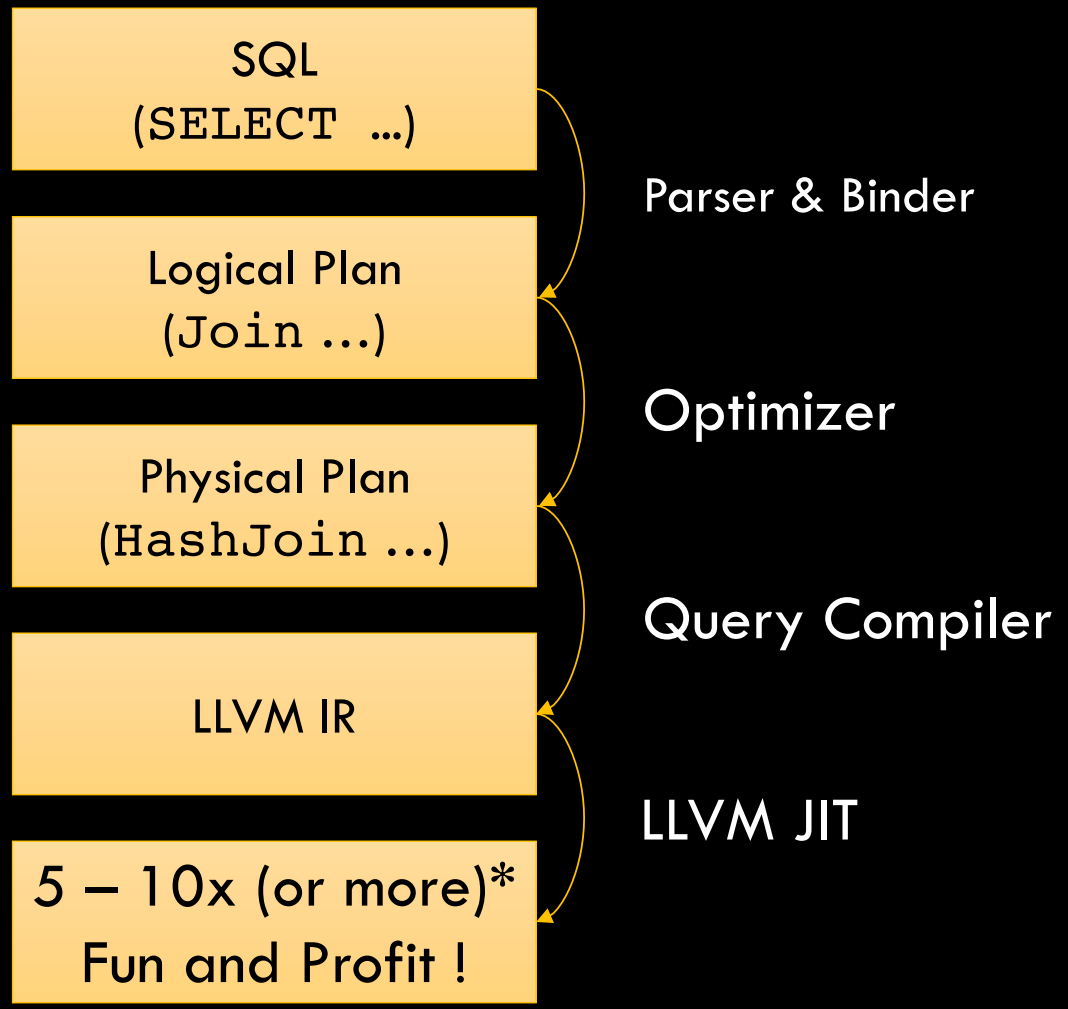
LLVM QUERY COMPILATION

15-721 Database Systems

Zhixun Tan
Shuyao Bi
Xinlyu Huang
Wei Cui

Executor → Compiler





Goals

- Peloton already has compilation for `Select`

- 75% and 100% Goals ✓
  - Support `Insert`, `Update`, `Delete`
  - Plan Caching
  - Parameterization

- 125% Goals ✗
  - Raw Memory Writing for Insert & Update (w/o `DataTable::*`)

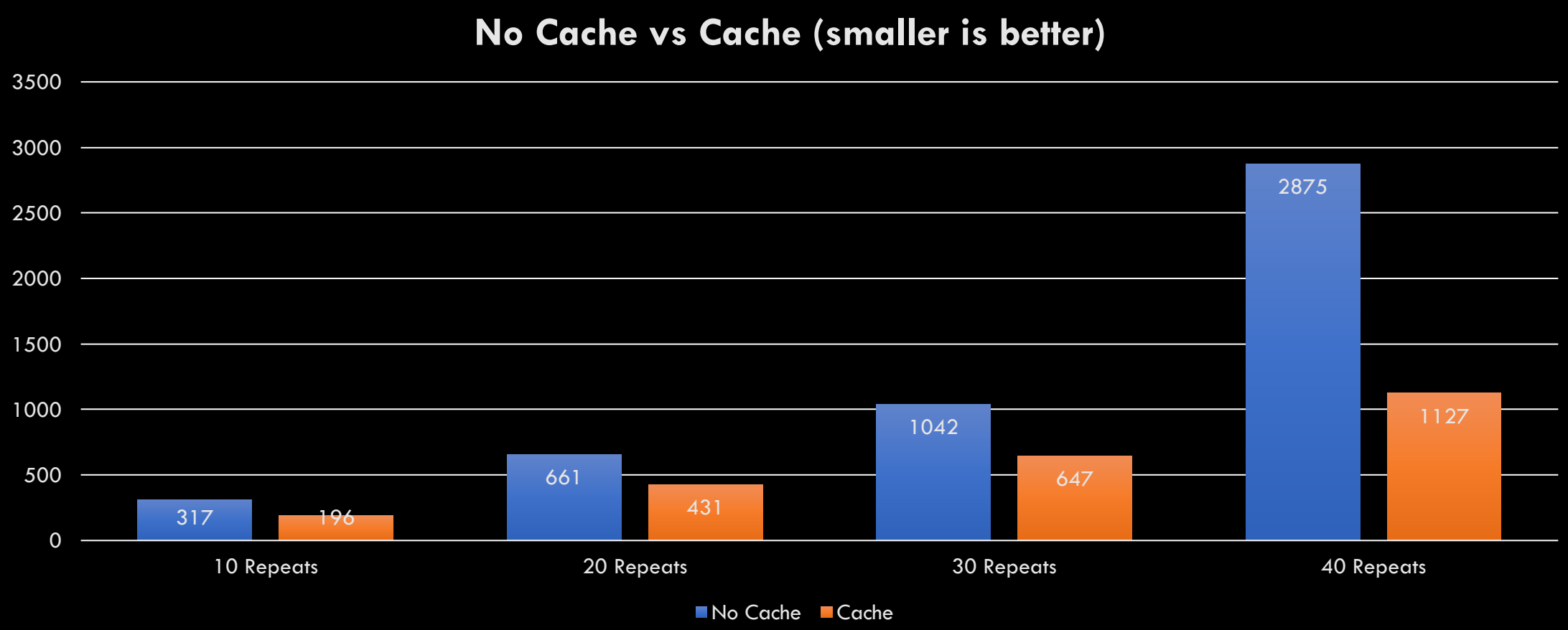
Task 1. Plan Caching

- Avoid re-compilation of the same plan
- LRU Cache

```
std::map<std::shared_ptr<AbstractPlan>, // physical
         std::shared_ptr<Query>>        // compiled
```

- No default `std::less<AbstractPlan>` → Custom Comparator

Benchmark - Cache



Task 2. Parameterization

```
SELECT *
FROM   table
WHERE  x < 10
```

```
void execute(output_t* output):
  for tuple in table:
    if tuple.x < 10:
      output->push(tuple)
```

```
SELECT *
FROM   table
WHERE  x < 11
```

```
void execute(output_t* output):
  for tuple in table:
    if tuple.x < 11:
      output->push(tuple)
```

```
void execute(output_t* output, int param1):
  for tuple in table:
    if tuple.x < param1:
      output->push(tuple)
```

Insert, Update, Delete

Existing Codegen: Scan

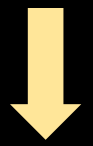
```
foreach tuple : table { // table scan
    if pred(tuple) {    // expr eval
        <your code here>
    }
}
```

Task 3. Delete

```
foreach tuple : table { // table scan
    if pred(tuple) {    // expr eval
        DataTable::PerformDelete(...)
    }
}
```

Tip: Calling C++ functions from C/assembly

```
class DataTable {
  ...
  size_t DataTable::GetTileGroupCount() const;
};
```



```
struct DataTable {
  ...
};
size_t GetTileGroupCount(const DataTable* this);
```

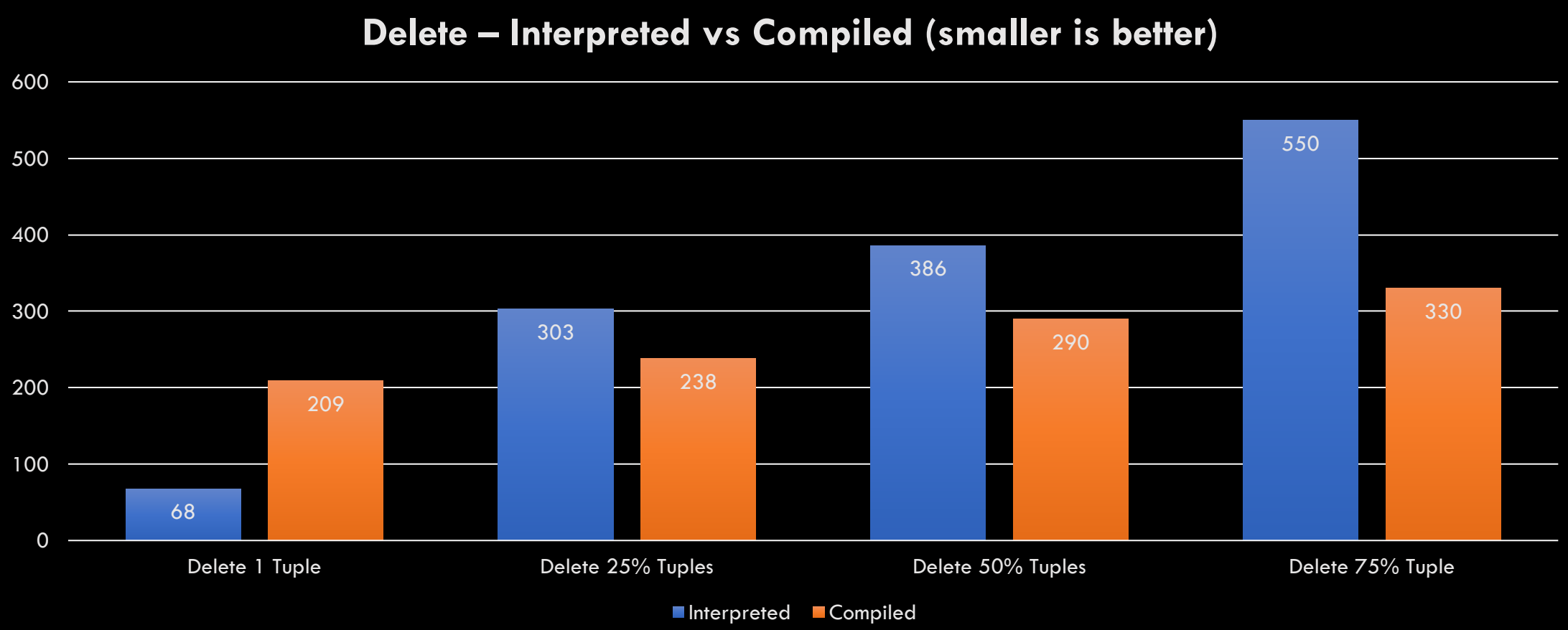
`class` to `struct`

Explicit `this` Pointer

```
size_t _ZNK7peloton7storage9DataTable17GetTileGroupCountEvy(const DataTable* this);
```

Name Mangling

Benchmark - Delete



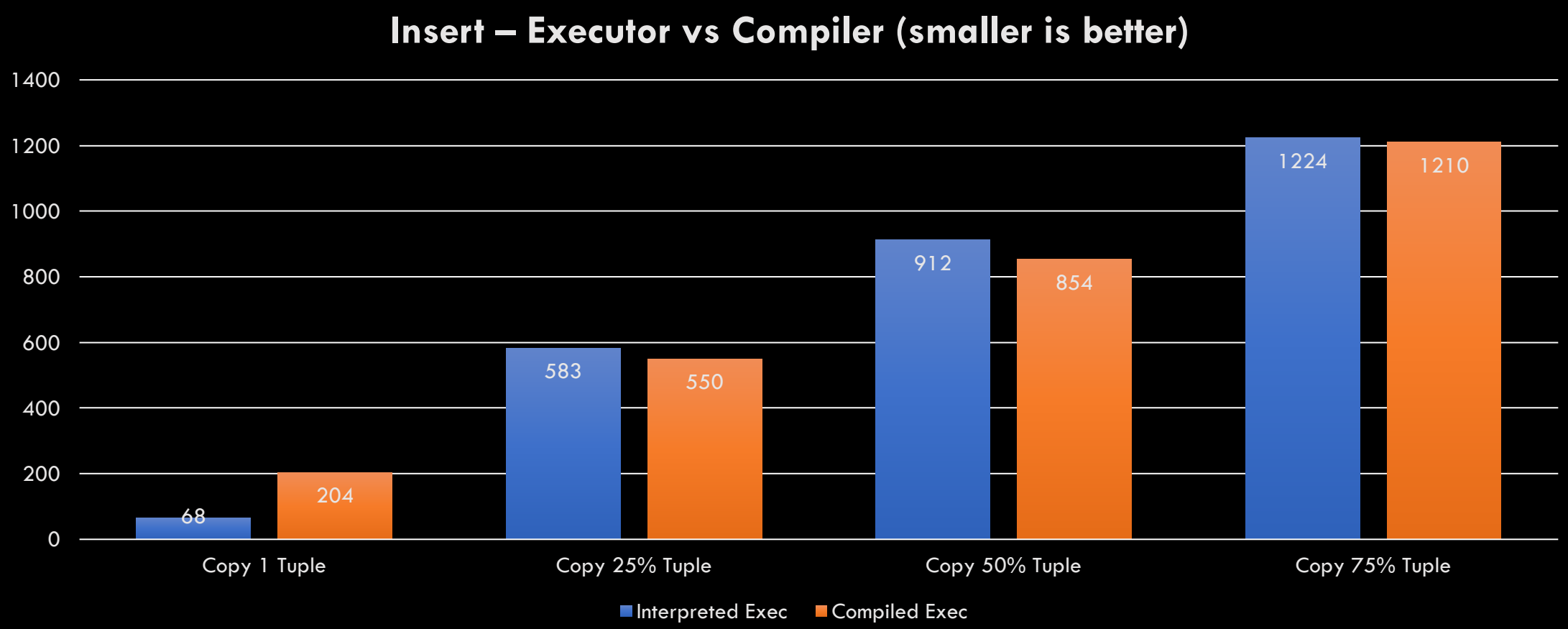
Task 4. Insert-Scan

```
foreach tuple : table { // table scan
    if pred(tuple) {    // expr eval
        materialize tuple       // LLVM
        DataTable::Insert(...) // C++
    }
}
```

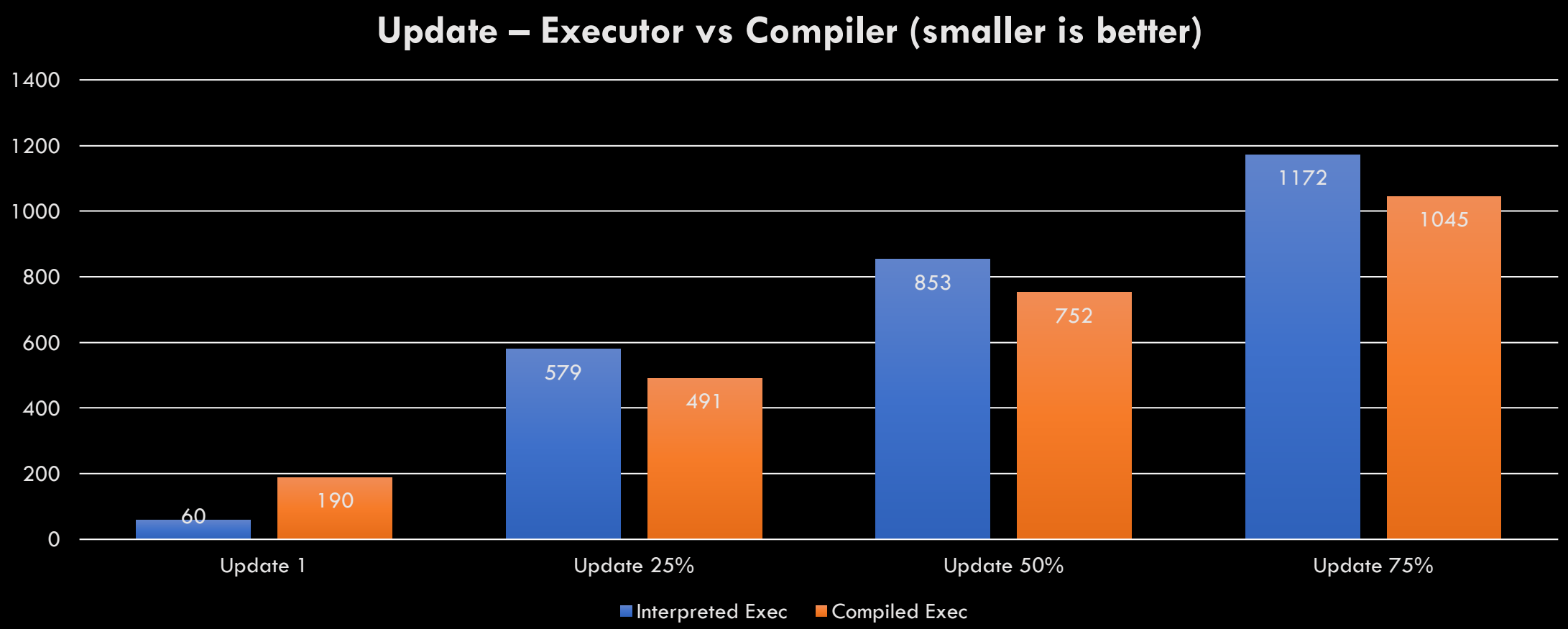
Task 5. Update

```
foreach tuple : table { // table scan
    if pred(tuple) {    // expr eval
        transform tuple                  // LLVM
        DataTable::InsertVersion(...) // C++
    }
}
```

Benchmark – Insert-Scan



Benchmark – Update



Future Work

- Indexed Scan
- Raw Memory Insert/Update