

On Another Level: How to Debug Compiling Query Engines

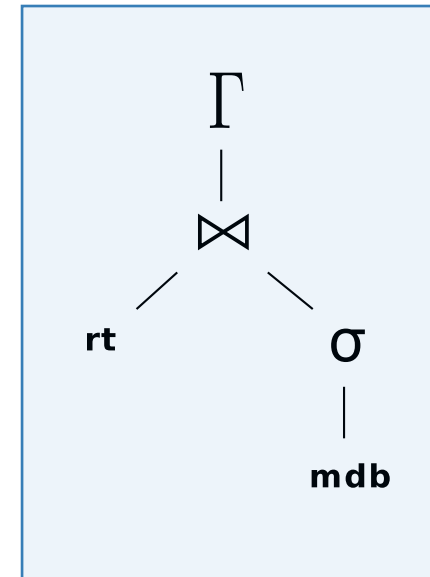
Timo Kersten and Thomas Neumann

Technical University of Munich

Debugging

Example Problem

```
1 select count(*)
2 from
3   RotatingTomatoes rt,
4   MovieDatabase mdb
5 where
6   rt.name = mdb.name and
7   rt.rating = mdb.rating and
8   mdb.reviews > 10;
```



Problem: Result count is zero, but there are two matching tuples!

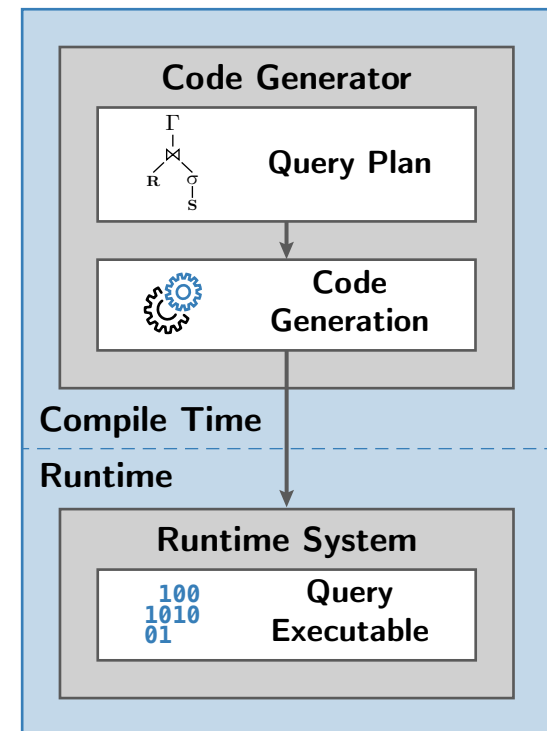
Code Generation for Query Execution

Compile Time:

Generate code

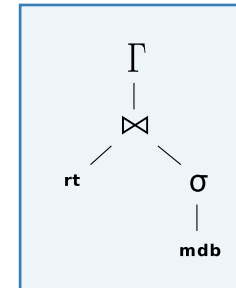
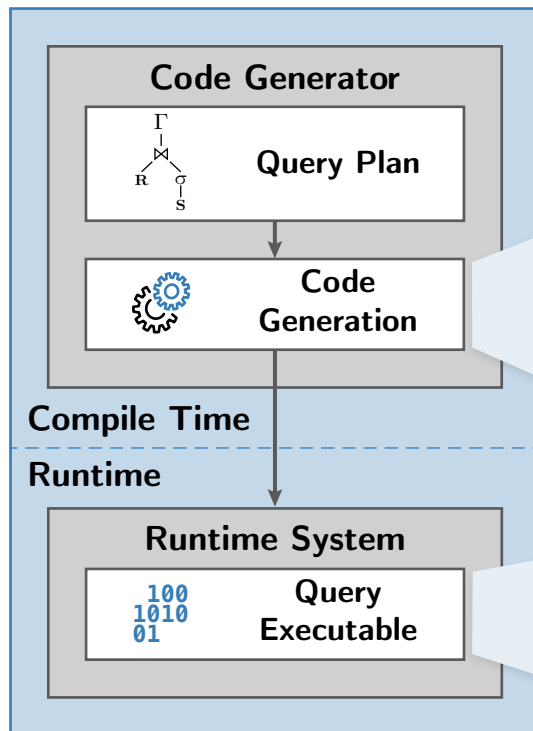
Runtime:

Run generated code



Example Problem

Code Generation



```
void JoinOperator::consume(ConsumerScope scope)
# probe side consume

hashTable.find(keys, scope, entry -> {
    ConsumerScope nestedScope(scope)
    unpack(leftValues, entry, nestedScope)
    parent.consume(nestedScope)
})
```


```
%9 = xor i64 %6, %8
%10 = call i64 @TextRuntime::hash(%4924, %9)
%11 = call ptr @HashTable::lookup(%ht, %10)
%12 = isnotnull ptr %11
condbr %12 %block4 %block1
```

Debugging

Code Generation

Debug at *compile time*:

 Bug does not show here



```
void JoinOperator::consume(ConsumerScope scope)
# probe side consume

    hashTable.find(keys, scope, entry -> {
        ConsumerScope nestedScope(scope)
        unpack(leftValues, entry, nestedScope)
        parent.consume(nestedScope)
    })
```

Debug at *runtime*:

 Code is very low-level



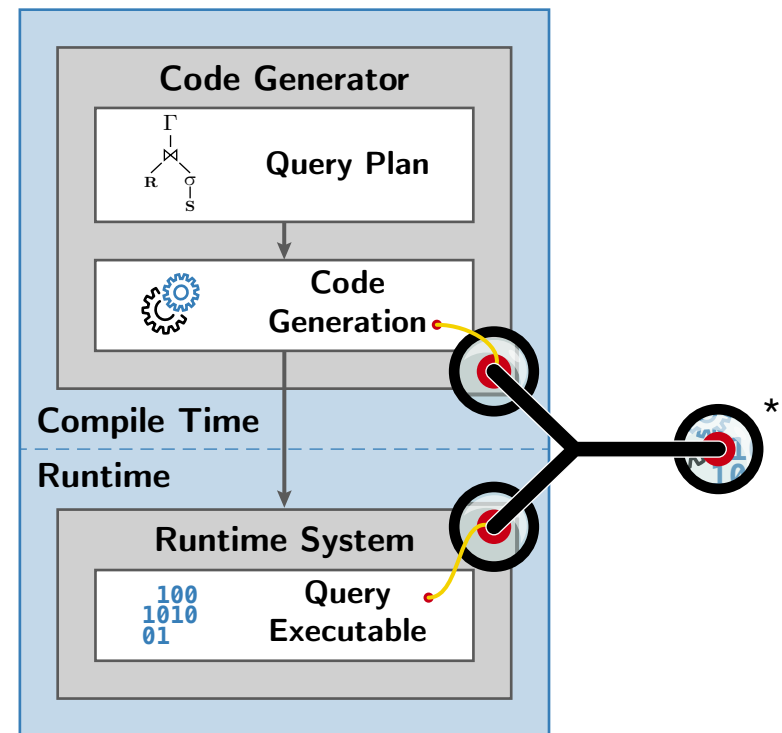
```
%9 = xor i64 %6, %8
%10 = call i64 @TextRuntime::hash(%4924, %9)
%11 = call ptr @HashTable::lookup(%ht, %10)
%12 = isnonnull ptr %11
condbr %12 %block4 %block1
```

Multi-Level Debugger

Context Included

Debug compile time
and runtime *simultaneously!*

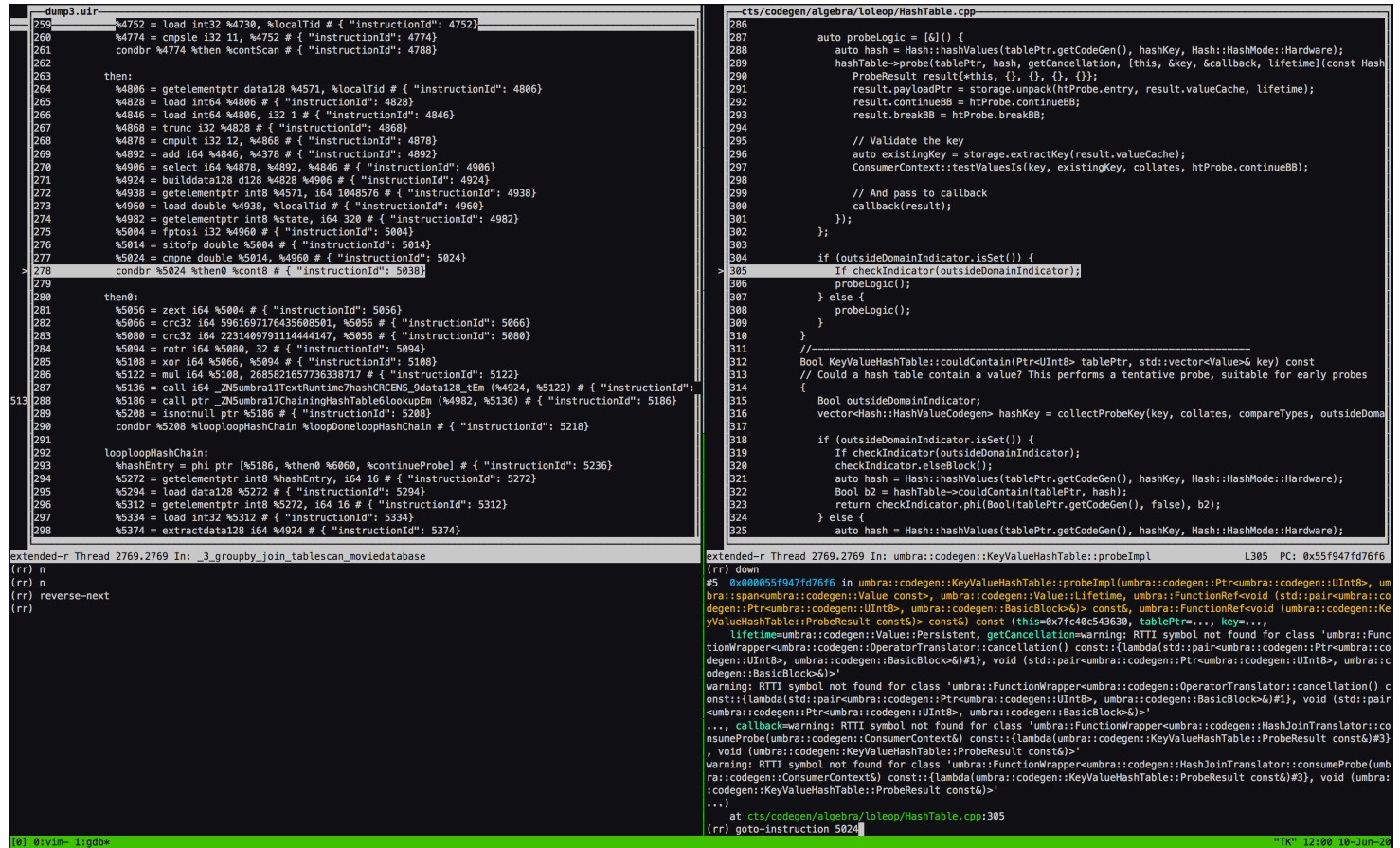
Time-travel debugger *
+ unique instruction identifiers
= Step through execution,
context from compile time ✓



* thanks *rr-project.org!*

Multi-Level Debugger

Demo



Add it to Your System

Core GDB command:

```
> tbreak IRProgram.cpp:972 if instr == 5038
```

Implementation effort:

- Reuse of debugger components
- Extensions to GDB with Python: ~ 70 lines

Evaluation

Runtime overhead:

- RR recording of TPC-H Q1 scale factor 1
- 470MB of recorded data
- Runtime 10 seconds (vs. 1 second without RR)

Multi-Level Debugger

Questions?

