

On Another Level: How to Debug Compiling Query Engines

Timo Kersten and Thomas Neumann
Technical University of Munich



Debugging

Example Problem

```
select count(*)
from
RotatingTomatoes rt,
MovieDatabase mdb
where
rt.name = mdb.name and
rt.rating = mdb.rating and
mdb
mdb
```

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



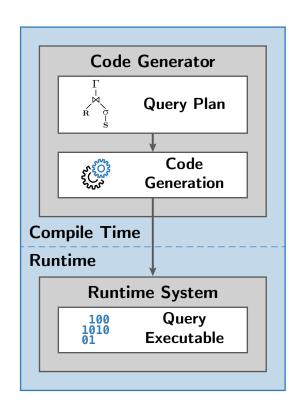
Code Generation for Query Execution

Compile Time:

Generate code

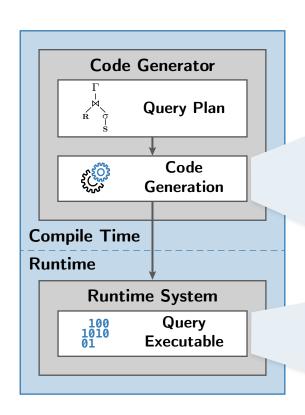
Runtime:

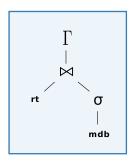
Run generated code





Example ProblemCode 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:

X Bug does not show here

Debug at *runtime*:

Code is very low-level

```
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
```

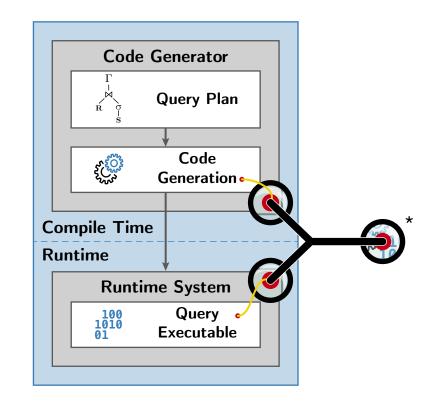


Multi-Level Debugger Context Included

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

```
-cts/codegen/algebra/loleop/HashTable.cpp-
                                            %4752 = load int32 %4730, %localTid # { "instructionId": 4752}
%4774 = cmpsle i32 11, %4752 # { "instructionId": 4774}
condbr %4774 %then %contScan # { "instructionId": 4788}
                                                                                                                                                                                                                                                                                                                                                                    auto probeLogic = [&]() {
                                                                                                                                                                                                                                                                                                                             288
289
290
291
292
293
294
295
296
297
298
299
301
302
303
304
305
306
307
308
310
311
312
313
                                                                                                                                                                                                                                                                                                                                                                           auto hash = Hash::hashValues(tablePtr.getCodeGen(), hashKey, Hash::HashMode::Hardware);
                                                                                                                                                                                                                                                                                                                                                                             hashTable->probe(tablePtr, hash, getCancellation, [this, &key, &callback, lifetime](const Hash ProbeResult result{*this, {}, {}, {}, {}}, {}};
                                            %4806 = getelementptr data128 %4571, %localTid # { "instructionId": 4806}
                                                                                                                                                                                                                                                                                                                                                                                   result.payloadPtr = storage.unpack(htProbe.entry, result.valueCache, lifetime);
                                            %4828 = load int64 %4806 # { "instructionId": 4828}
%4846 = load int64 %4806, i32 1 # { "instructionId": 4846}
                                                                                                                                                                                                                                                                                                                                                                                   result.continueBB = htProbe.continueBB:
                                                                                                                                                                                                                                                                                                                                                                                   result.breakBB = htProbe.breakBB:
                                             %4868 = trunc i32 %4828 # { "instructionId": 4868}
                                          %4868 = trunc 132 %4828 # { "instructionId": 4868} 

M878 = cmpul 132 12, %4868 # { "instructionId": 4878} 

%4892 = add 164 %4846, %4378 # { "instructionId": 4892} 

%4966 = select 164 %4878, %4892, %4864 # { "instructionId": 4906} 

%4924 = buildata128 d128 %4828 %4906 # { "instructionId": 4924} 

%4938 = getelementpt in18 %4571, 164 1048576 # { "instructionId": 4938} 

%4960 = load double %4938, %localTid # { "instructionId": 4960}
                                                                                                                                                                                                                                                                                                                                                                                   // Validate the key
                                                                                                                                                                                                                                                                                                                                                                                   auto existingKey = storage.extractKey(result.valueCache);
                                                                                                                                                                                                                                                                                                                                                                                   ConsumerContext::testValuesIs(key, existingKey, collates, htProbe.continueBB);
                                                                                                                                                                                                                                                                                                                                                                                   // And pass to callback
                                                                                                                                                                                                                                                                                                                                                                                  callback(result):
                                             %4982 = getelementptr int8 %state, i64 320 # { "instructionId": 4982}
                                             %5004 = fptosi i32 %4960 # { "instructionId": 5004}
                                            %5014 = sitofp double %5004 # { "instructionId": 5014}
%5024 = cmpne double %5014, %4960 # { "instructionId": 5024}
                                                                                                                                                                                                                                                                                                                                                                     if (outsideDomainIndicator.isSet()) {
                                            condbr %5024 %then0 %cont8 # { "instructionId": 5038}
                                                                                                                                                                                                                                                                                                                                                                         If checkIndicator(outsideDomainIndicator):
                                                                                                                                                                                                                                                                                                                                                                            probeLogic():
                                                                                                                                                                                                                                                                                                                                                                   } else {
                                             %5056 = zext i64 %5004 # { "instructionId": 5056}
                                                                                                                                                                                                                                                                                                                                                                           probeLogic():
                                            %5066 = crc32 i64 5961697176435608501, %5056 # { "instructionId": 5066} %5080 = crc32 i64 2231409791114444147, %5056 # { "instructionId": 5080}
                                          %5080 = crc32 164 2231409791114444147, %5055 # ("instructionId": 5080}
%5094 = rotr 164 %5080, 32 # ("instructionId": 5080)
%5192 = mul 164 %5100, 368521657736339717 # ("instructionId": 5102}
%5122 = mul 164 %5100, 368521657736339717 # ("instructionId": 5122}
%5136 = call i64 25100, 368521657736339717 # ("instructionId": $122}
%5136 = call i64 25100, 368521657736339717 # ("instructionId": $1080}
%5136 = call ptr_2NSumbralTextRuntimePhashCRCENS_904s128_tEm (%4924, %5136) # ("instructionId": $1080}
%5280 = isnotnull ptr %5186 # ("instructionId": $208)
                                                                                                                                                                                                                                                                                                                                                             Bool KeyValueHashTable::couldContain(Ptr<UInt8> tablePtr, std::vector<Value>& key) const
                                                                                                                                                                                                                                                                                                                                                             // Could a hash table contain a value? This performs a tentative probe, suitable for early probes
                                                                                                                                                                                                                                                                                                                             315
316
317
318
319
320
321
                                                                                                                                                                                                                                                                                                                                                                   Bool outsideDomainIndicator;
                                                                                                                                                                                                                                                                                                                                                                     vector<Hash::HashValueCodegen> hashKey = collectProbeKey(key, collates, compareTypes, outsideDoma
                                             condbr %5208 %looploopHashChain %loopDoneloopHashChain # { "instructionId": 5218}
                                                                                                                                                                                                                                                                                                                                                                    if (outsideDomainIndicator.isSet()) {
                                                                                                                                                                                                                                                                                                                                                                           If checkIndicator(outsideDomainIndicator);
                                           uopicupmasiciain:
**MashEntry = phi ptr [%5186, %then0 %6060, %continueProbe] # { "instructionId": 5236}
%5272 = getelementptr int8 %hashEntry, 164 16 # { "instructionId": 5272}
%5294 = load data128 %5272 # { "instructionId": 5294}
%5312 = getelementptr int8 %5272, 164 16 # { "instructionId": 5312}
                                                                                                                                                                                                                                                                                                                                                                           checkIndicator.elseBlock():
                                                                                                                                                                                                                                                                                                                                                                            auto hash = Hash::hashValues(tablePtr.getCodeGen(), hashKey, Hash::HashMode::Hardware);
        295
                                                                                                                                                                                                                                                                                                                                                                           Bool b2 = hashTable->couldContain(tablePtr, hash);
                                                                                                                                                                                                                                                                                                                                                                            return checkIndicator.phi(Bool(tablePtr.getCodeGen(), false), b2);
                                            %5334 = load int32 %5312 # { "instructionId": 5334}
%5374 = extractdata128 i64 %4924 # { "instructionId": 5374}
                                                                                                                                                                                                                                                                                                                                                                            auto hash = Hash::hashValues(tablePtr.getCodeGen(), hashKey, Hash::HashMode::Hardware);
extended-r Thread 2769.2769 In: _3_groupby_join_tablescan_moviedatabase
                                                                                                                                                                                                                                                                                                                    extended-r Thread 2769.2769 In: umbra::codegen::KeyValueHashTable::probeImpl
(rr) n
(rr) n
                                                                                                                                                                                                                                                                                                                 (Irr down
fy 8.0000555947fd76f6 in umbra::codegen::KeyValueHashTable::probeImpl(umbra::codegen::Ptr<umbra::codegen::UInt8>,
bra::span-cumbra::codegen::Value const>, umbra::codegen::Value::Lifetime, umbra::FunctionRef<void (std::pair<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::Ptr<umbra::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::degen::
 rr) reverse-next
                                                                                                                                                                                                                                                                                                                  lifetime=umbra::codegen::Value::Persistent, getCancellation=warning: RTTI symbol not found for class 'umbra::Func
tionWrapper<umbra::codegen::OperatorTranslator::cancellation() const::{lambda(std::pair<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Ptr<umbra::codegen::Codegen::Ptr<umbra::codegen::Codegen::Codegen::Codegen::Ptr<umbra::codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen::Codegen
                                                                                                                                                                                                                                                                                                                   degen::UInt8>, umbra::codegen::BasicBlock>&)#1}, void (std::pair<umbra::codegen::Ptr<umbra::codegen::UInt8>, umbra::
                                                                                                                                                                                                                                                                                                                   odegen::BasicBlock>&)>'
                                                                                                                                                                                                                                                                                                                   warning: RTTI symbol not found for class 'umbra::FunctionWrapper<umbra::codegen::OperatorTranslator::cancellation()
                                                                                                                                                                                                                                                                                                                   onst::{lambda(std::pair<umbra::codegen::Ptr<umbra::codegen::UInt8>, umbra::codegen::BasicBlock>&)#1}, void (std::pai
                                                                                                                                                                                                                                                                                                                 cumbra::codegen::Ptr-cumbra::codegen::Unt8>, umbra::codegen::RasicElocks6)>-
..., callback=warning: RTI symbol not found for class 'umbra::FunctionWrapper<umbra::codegen::HashJoinTranslator::co
sumeProbe(umbra::codegen::ConsumerCnotext6) const: (lambda(umbra::codegen::KeyValueHashTable::ProbeResult const6)#3)
, void (umbra::codegen::KeyValueHashTable::ProbeResult const6)>'
                                                                                                                                                                                                                                                                                                                   warning: RTTI symbol not found for class 'umbra::FunctionWrapper<umbra::codegen::HashJoinTranslator::consumeProbe(umbra::codegen::ConsumerContext&) const::{lambda(umbra::codegen::KeyValueHashTable::ProbeResult const&)#3}, void (umbra:
                                                                                                                                                                                                                                                                                                                   :codegen::KeyValueHashTable::ProbeResult const&)>'
                                                                                                                                                                                                                                                                                                                          at cts/codegen/algebra/loleop/HashTable.cpp:305
                                                                                                                                                                                                                                                                                                                   (rr) goto-instruction 5024
```



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

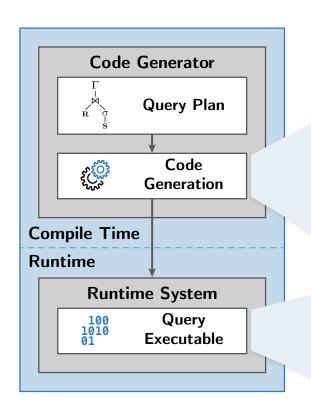
Runtime overhead:

- 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?



```
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
```