



Kolloquium Qualitätssicherung IV Visualizer

Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB, Josua Benjamin Eyl, Lukas Friedrich, Max Bretschneider, Nathaniel Till Hartmann, Robin Köchel

Betreut von: Mickael Cormier M.Sc., Stefan Wolf M.Sc



Arbeitsweise

Vorgehen:

- Verwenden von Feature-Banches
- Issues in GitLab
- Abarbeiten von funktionalen Anforderung und optionalen Anforderungen
- Testen des Programmcodes
 - Unit-Tests, Google Tests, etc.



Arbeitsweise

CI/CD Pipeline:

- Frontend
 - Der Build-Prozess , überprüft ob der Code ohne Fehler des Compilers erstellbar ist
 - die Dokumentation wird von Doxygen erzeugt
- Backend
 - Der Build-Prozess , überprüft ob der Code ohne Fehler des Compilers erstellbar ist
 - durch Unit-Tests wird die Funktionalität einzelner Komponenten überprüft
 - die Dokumentation wird von Doxygen erzeugt



Arbeitsweise

Statische Codeanalyse

- Doxygen
 - Dokumentation des Programmcodes
- CLOC
 - zählt alle Zeilen an Code zusammen und unterteilt diese in Leer-, Kommentar- und Codezeilen
 - zählt auch die Anzahl der benutzten Dateien



Tools

Google Test Framework

Icov

GDB

Valgrind



gtest

```
TEST_F(DBManagerTest, Null_InsertAndGetBoundingBoxes) {
    QVector<std::shared_ptr<iv_annotation::BoundingBox>> bboxes;
    QVector<std::shared_ptr<iv_annotation::RegionOfInterest>> rois;
    uint64_t timestamp = std::chrono::system_clock::now().time_since_epoch().count();

    ASSERT_TRUE(dbManager->insertBoundingBoxes(bboxes, rois, 30, timestamp));

    auto [retrievedAnnotations, retrievedRois, fps] = dbManager->getBoundingBoxes(timestamp, timestamp + 1);
    ASSERT_EQ(retrievedAnnotations.size(), 1);

    auto retrievedBboxes = retrievedAnnotations[0];
    ASSERT_TRUE(retrievedBboxes.empty());
}
```



The screenshot shows a terminal window with a dark background. The left pane displays a tree view of test results, and the right pane shows the command and output of the test execution.

Left Pane (Test Results):

- ✓ IV_VISUALIZER_test
 - ✓ DBManagerTest
 - ✓ Random_InsertAndGetROI
 - ✓ Zero_InsertAndGetROI
 - ✓ Max_InsertAndGetROI
 - ✓ Random_InsertAndGetBoundingBoxes
 - ✓ Zero_InsertAndGetBoundingBoxes
 - ✓ Null_InsertAndGetBoundingBoxes
 - ✓ Max_InsertAndGetBoundingBoxes
 - ✓ GetTimeRangeTuple
 - ✓ GetTimeRange
 - > ✓ DataManagerTest
 - > ✓ FSMManagerTest
 - > ✓ BoundingBoxTest
 - > ✓ MessageHandlerTest
 - > ✓ TimeRangeTest

Right Pane (Test Output):

```
✓ Tests passed: 16 of 16 tests – 3 sec 552 ms

/workspaces/iv_visualizer/cmake-build-debug/test/IV_VISUALIZER_test --gtest_filter=* --gtest_color=no
Testing started at 6:48 AM ...
==== Using optimized driver!!! ====
Process finished with exit code 0
```



```
test:
  stage: test
  services:
    - name: scylladb/scylla:5.4
      alias: scylla
  variables:
    SCYLLA_IP: "scylla"
  script:
    - cd ${BUILD_DIR}/test
    - ./IV_VISUALIZER_test
  dependencies:
    - build
```


lcov

Filename	Line Coverage 	Functions 
DBManager.cpp	 92.8 % 374 / 403	96.0 % 24 / 25

```
while (true) {  
    connectionFuture = cass_session_connect(session, cluster);  
    cassError = cass_future_error_code(connectionFuture);  
    if (handleErrors(cassError)) {  
        break;  
    } else {  
        cass_future_free(connectionFuture);  
        std::cerr << "Retrying to connect to ScyllaDB in 5 seconds...\n";  
        sleep(5);  
    }  
}
```

GDB

Console GDB Memory View

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

```
> this = {DBManagerTest_Random_InsertAndGetBoundingBoxes_Test *const} 0x7ffcc001aa0
> bboxes = {QVector}
> bboxesCopy = {QVector}
  > QListSpecialMethods<std::shared_ptr<iv_annotation::BoundingBox> > = {QListSpecialMethods<std::shared_ptr<iv_annotation::BoundingBox>>}
    ⓘ QListSpecialMethodsBase<std::shared_ptr<iv_annotation::BoundingBox> > = {QListSpecialMethodsBase<std::shared_ptr<iv_annotation::BoundingBox>>}
  > d = {QList<std::shared_ptr<iv_annotation::BoundingBox>>::DataPointer}
    > d = {QArrayDataPointer<std::shared_ptr<iv_annotation::BoundingBox>>::Data *} 0x5555570e9280
  > ptr = {std::shared_ptr<iv_annotation::BoundingBox> *} 0x5555570e9290
    > get() = {iv_annotation::BoundingBox *} 0x5555570a15d0
      > m_bbox = {std::vector<int>} std::vector of length 4, capacity 4
        ⓘ m_rotation = {int} 259
        ⓘ m_confidence = {float} 0.880000055
      > m_label = {QString}
      > m_image_wh = {cv::Size2i}
    ⓘ size = {qsize_t} 1000
```

Valgrind

```

  Leak_DefinitelyLost 2 warnings
    DBManager.cpp 2 warnings
      > 21,288 (9,720 direct, 11,568 indirect) bytes in 243 blocks are definitely lost in loss record 2,069 of 2,138 1 warning
      > 335,568 (302,784 direct, 32,784 indirect) bytes in 228 blocks are definitely lost in loss record 2,112 of 2,138 1 warning
        0x4848899 malloc
        0x6D5A274 datastax::internal::Memory::malloc
        0x6D61B71 datastax::internal::Allocated::operator new
        0x6DFA8B cass_data_type_new_udt
        0xEC4766 db::DBManager::getEmptyCassBoundingBox DBManager.cpp:709
        0xEC577C db::DBManager::insertBoundingBoxes DBManager.cpp:163
        0xEBE109 DataManager::addDataFrame DataManager.cpp:70
        0x239934 MessageHandler::startIncomingStream MessageHandler.cpp:191
        0x2B764F operator() ivbackend_grpc.pb.cc:284
        0x2B764F __invoke_impl<...> invoke.h:61
        0x2B764F __invoke_r<...> invoke.h:116
        0x2B764F std::_Function_handler::_M_invoke std_function.h:291
        0x2D0842 operator() std_function.h:590
        0x2D0842 operator() method_handler.h:115
        0x2D0842 grpc::internal::CatchingFunctionHandler<...> method_handler.h:43
        0x2D542F grpc::internal::RpcMethodHandler::RunHandler method_handler.h:112
        0x27C2F5 grpc::Server::SyncRequest::ContinueRunAfterInterception
  Leak_PossiblyLost 65 warnings
```

Preview Editor Frame Information

Instruction Pointer: 0xEC4766


Object: /workspaces/iv_visualizer/cmake-build-debug/ivbackend

Function: db::DBManager::getEmptyCassBoundingBox()

Directory: /workspaces/iv_visualizer/source/backend/model

File: DBManager.cpp

Line: 709



```
CassUserType *DBManager::getEmptyCassBoundingBox() {
    CassDataType *boundingBox = cass_data_type_new_udt(6);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "coords", CASS_VALUE_TYPE_LIST);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "confidence", CASS_VALUE_TYPE_FLOAT);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "label", CASS_VALUE_TYPE_TEXT);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "imagesize", CASS_VALUE_TYPE_TUPLE);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "format", CASS_VALUE_TYPE_INT);
    cass_data_type_add_sub_value_type_by_name(boundingBox, "rotation", CASS_VALUE_TYPE_INT);
    return cass_user_type_new_from_data_type(boundingBox);
}
```



Schwierigkeiten beim Testen

- Grpc
- Frontend Module nur schwierig mit Unit Tests testbar
- Dummy
- zu Beginn noch keine stehende Verbindung zwischen Front und Backend
- Hardware Limitation



Fortschritte & behobene Fehler

- Viele kleine Issues & Fehler
- Integrationstests
- Ausgiebigere Fehlerbehandlung
- Beständige Fehler besser dokumentiert
- **Besseres Zusammenspiel Backend, Frontend und Datenbank**



Memory Management

- Unterschiedliche Behandlung von grundlegendem Datentyp
- Hat zu Segfaults geführt
- Memory Rework im Backend
- Hat auch zu allgemein besserer Performance geführt



Dummy Extension

- Dummy wurde erweitert mit Hilfe vom Fraunhofer-Team
- Echte Daten
- Kleine Fehler in der Generierung der Daten behoben



Kleine Issues

- Datenverlust der Annotationen im Backend
- Konvertierung zwischen den Proto Messages
- Unterschiedliche Proto Versionen
- Update von verfügbaren Zeiten nicht rechtzeitig
- Allgemein Container Issues/Config wurde zur falschen URL geschickt
- Probleme mit Libraries, wie z.B. Qt



Demo



Danke für Ihre Aufmerksamkeit!

- Fragen?