Item 352
git_comments:

1. * * Context of the schema, the context can be generated from a source schema or descriptors. The * ability of converting back from Row to proto depends on the type of context.
2. * * Context the contains the full {@link ProtoDomain} and a reference to the message name. The full * domain is needed for creating Rows back to the original proto messages.
3. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
4. * * Create a new ProtoDynamicMessageSchema from a {@link ProtoDomain} and for a descriptor. The * descriptor is only used for it's name, that name will be used for a search in the domain.
5. * List of field converters for each field in the row.
6. * * Proto has a well defined way of storing maps, by having a Message with two fields, named "key" * and "value" in a repeatable field. This overlay translates between Row.map and the Protobuf * map.
7. handled by logical type case
8. Protobuf messages BYTES doesn't like empty bytes?!
9. * Enum overlay handles the conversion between a string and a ProtoBuf Enum.
10. * * This overlay handles nullable fields. If a primitive field needs to be nullable this overlay is * wrapped around the original overlay.
11. * The toRow function to convert the Message to a Row.
12. * Convert a row value and set it on a proto message.
13. * * Base converter class for converting from proto values to row values. The converter mainly works * on fields in proto messages but also has methods to convert individual elements (example, for * elements in Lists or Maps).
14. * * Specific converter for Proto Wrapper values as they are translated into nullable row values.
15. * Get a proto field and convert it into a row value.
16. * Converter for primitive proto values.
17. * * Base class for the protobuf message. Normally this is DynamicMessage, but as this schema * class is also used to decode protobuf options this can be normal Message instances.
18. * Initialize the transient fields after deserialization or construction.
19. * The fromRow function to convert the Row to a Message.
20. * Convert Proto oneOf fields into the {@link OneOfType} logical type.
21. * Convert a row value into a proto value.
22. return object;
23. * Convert a proto value into a row value.
24. * * Converter for Bytes. Protobuf Bytes are natively represented as ByteStrings that requires * special handling for byte[] of size 0.
25. * * Create a new ProtoDynamicMessageSchema from a {@link ProtoDomain} and for a message. The * message need to be in the domain and needs to be the fully qualified name.
26. * * Context that only has enough information to convert a proto message to a Row. This can be used * for arbitrary conventions, like decoding messages in proto options.
27. equality doesn't work between dynamic messages and other, so we compare string representation
28. Test map type
29. * Collection of tests for values on Protobuf Messages and Rows.
30. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
31. * Create a {@link Value} specifying which field to set and the value to set.
32. * Attach the name of the message to a map key.
33. * Return the message name for a map key.
34. * Attach the name of the message to a type.
35. * Return the message name for a type.
36. * Return the message name for a map value.
37. * Attach the name of the message to a map value.

git_commits:

1. **summary:** Merge pull request #10502: [BEAM-7274] Add DynamicMessage Schema support
   message: Merge pull request #10502: [BEAM-7274] Add DynamicMessage Schema support

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [BEAM-7274] Add DynamicMessage Schema support
   body: Add DynamicMessage schema support. This is different from generated classes as it uses the proto descriptors. It uses the ProtoDomain as an index for searching embedded messages. R: @reuvenlax ---------- Lang | SDK | Apex | Dataflow | Flink | Gearpump | Samza | Spark --- | --- | --- | --- | --- | --- | --- | --- Go | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Go/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Go/lastCompletedBuild/) | --- | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Go_VR_Flink/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Go_VR_Flink/lastCompletedBuild/) | --- | --- | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Go_VR_Spark/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Go_VR_Spark/lastCompletedBuild/) Java | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Apex/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Apex/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Dataflow/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Dataflow/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Flink/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Flink/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Flink_Batch/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Flink_Batch/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Flink_Streaming/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Flink_Streaming/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Gearpump/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Gearpump/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Samza/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Samza/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Spark/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_Spark/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Spark_Batch/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_PVR_Spark_Batch/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_SparkStructuredStreaming/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Java_ValidatesRunner_SparkStructuredStreaming/lastCompletedBuild/) Python | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Python2/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python2/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Python35/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python35/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Python36/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python36/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Python37/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python37/lastCompletedBuild/) | --- | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Py_VR_Dataflow/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Py_VR_Dataflow/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Py_ValCont/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Py_ValCont/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Python2_PVR_Flink_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Python2_PVR_Flink_Cron/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PostCommit_Python35_VR_Flink/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python35_VR_Flink/lastCompletedBuild/) | --- | --- | [![Build Status](https://builds.apache.org/job/beam_PostCommit_Python_VR_Spark/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_Python_VR_Spark/lastCompletedBuild/) XLang | --- | --- | --- | [![Build Status](https://builds.apache.org/job/beam_PostCommit_XVR_Flink/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PostCommit_XVR_Flink/lastCompletedBuild/) | --- | --- | --- Pre-Commit Tests Status (on master branch) ---------- --- |Java | Python | Go | Website --- | --- | --- | --- | --- Non-portable | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Java_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Java_Cron/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Python_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Python_Cron/lastCompletedBuild/)
[![Build Status](https://builds.apache.org/job/beam_PreCommit_PythonLint_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_PythonLint_Cron/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Go_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Go_Cron/lastCompletedBuild/) | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Website_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Website_Cron/lastCompletedBuild/) Portable | --- | [![Build Status](https://builds.apache.org/job/beam_PreCommit_Portable_Python_Cron/lastCompletedBuild/badge/icon)](https://builds.apache.org/job/beam_PreCommit_Portable_Python_Cron/lastCompletedBuild/) | --- | --- See [.test-infra/jenkins/README](https://github.com/apache/beam/blob/master/.test-infra/jenkins/README.md) for trigger phrase, status and link of all Jenkins jobs.

github_pulls_comments:

1. Run Java PreCommit

2. R: @reuvenlax this is an implementation for DynamicMessages. It takes into account the new Logical Types and also use RowWithStorage instead of going with the getters.
3. @reuvenlax this is ready for review. It has feature parity with the static compiled proto, including all logical types.
4. Run Java PreCommit
5. This branch has been updated to take the new Logical Types into account.
6. Run Java PreCommit
7. @reuvenlax @iemejia @TheNeuralBit I've tested this PR extensively on Dataflow. This is way more stable than the original DynamicMessage schema implementation. Can I get a LGTM, that I can focus on the options. Thanks. (master is failing, it was green till rebase)
8. Run Java PreCommit
9. Run Java PreCommit
10. A few comments.

github_pulls_reviews:

1. I think you can replace the lambda with this::indexDescriptor
2. you can do the same here - forEach(this::indexDescriptor)
3. I don't think we should be storing SchemaCoder objects around here. I think preferable to store schema and to/from row functions separately.
4. fixed
5. fixed
6. I've changed the accesses here, but iI still think that it's ok to use SchemaCoder as a transient container for the trio: Schema, ToRow and FromRow. If it's a blocker, I can spit it up in 3 different transient fields.
7. I would prefer 3 fields. SchemaCoder is not a container object - it just happens to contain those fields. In fact I would prefer to make SchemaCoder private (so that users of Schemas don't have to ever deal with coders at all), however that would require more major changes to Beam.
8. Removed coder as container.
9. I don't see this function being used anywhere. Remove?
10. I used it in a production pipeline, but as it's not available in the other providers I removed it, (squashed and rebased).

jira_issues:

1. **summary:** Protobuf Beam Schema support
   description: Add support for the new Beam Schema to the Protobuf extension.

jira_issues_comments:

1. [~alexvanboxel] any update on this? I'm looking to get started on BEAM-4455 which depends on this. Thanks in advance!
2. Well. picking it up again as I'm back from holidays. Hopefully we then get the pull request out in a reasonable time.
3. PR ready for review
4. Moved to 2.17.0
5. Is this resolved? Associated PR is not merged yet.
6. Reopening as new comments on PR still needs to be resolved
7. [~alexvanboxel] Seems to me this is an improvement. I am going to cut release branch for 2.20.0 tomorrow. Do you agree that we can push this Jira to 2.21.0 release?
8. Moved to 2.21
9. Moving back to 2.20 as it's merged into master