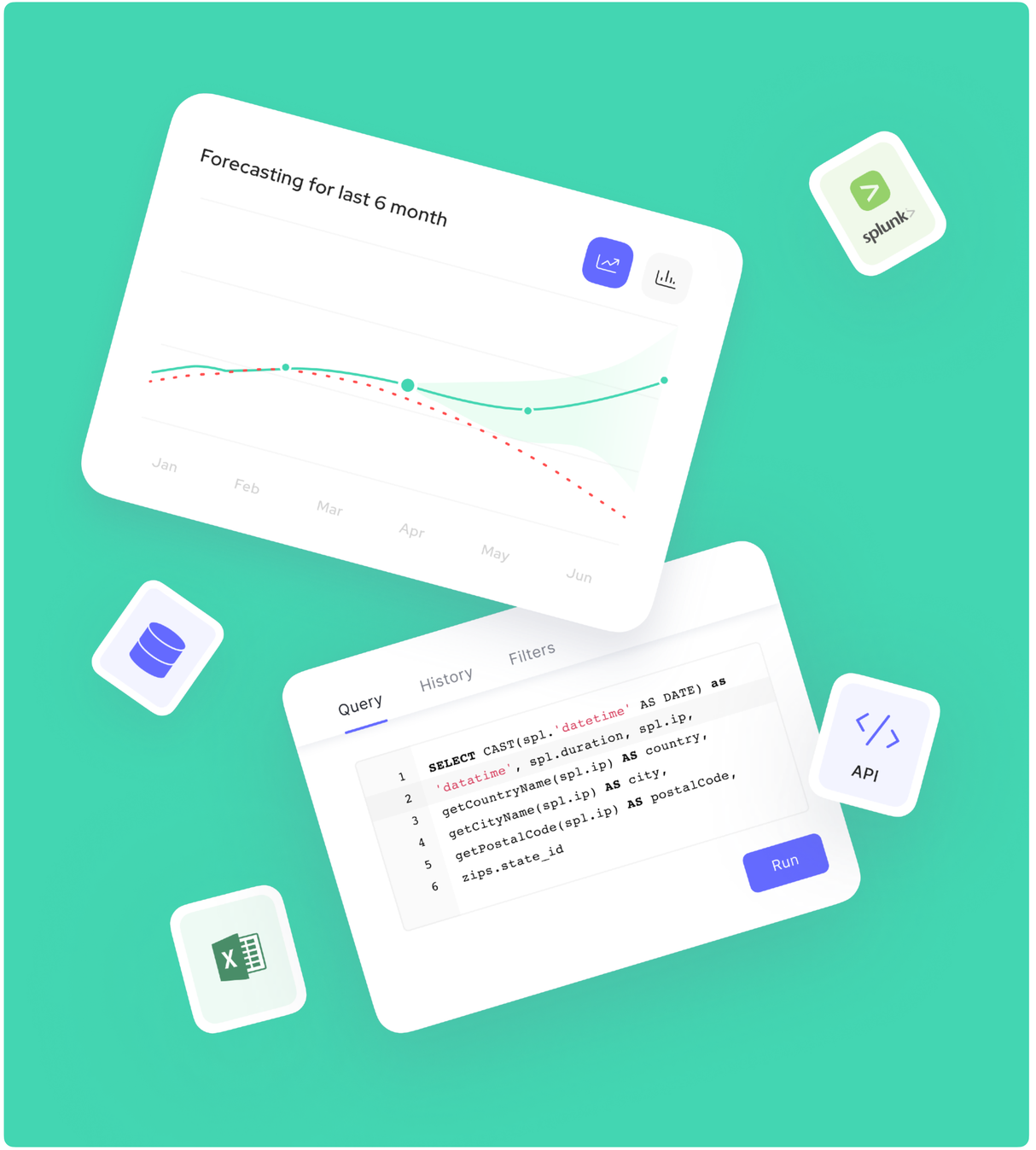
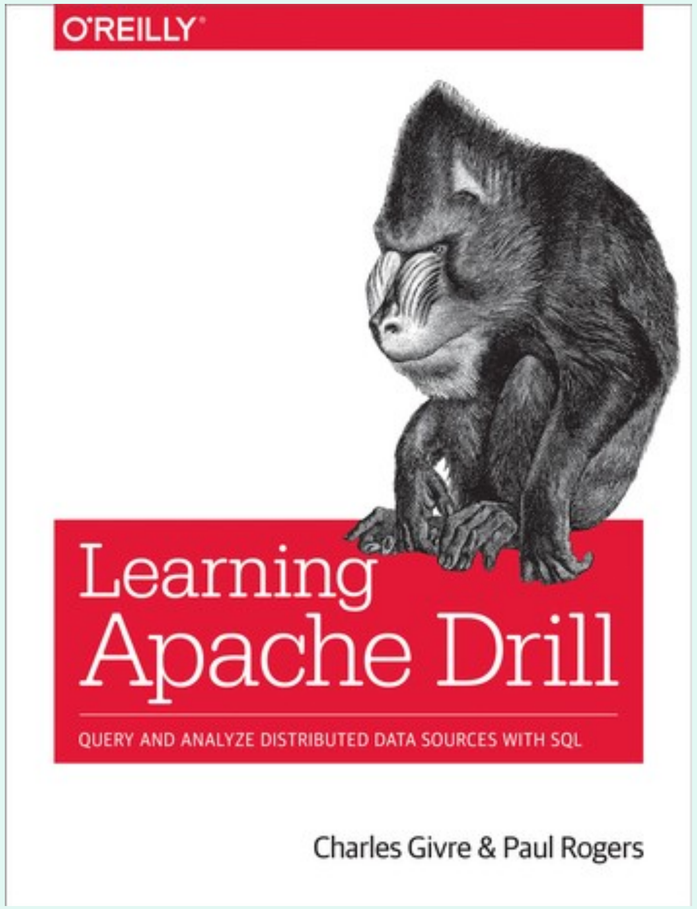
DataDistillr

HDF5 & SQL: Together at Last!



About Me

- Former CIA/NSA Cyber Security Professional
- Speaker at BlackHat, Strata and O'Reilly Author
- Apache Drill Contributor and PMC Chair
- Literally Wrote the Book on Drill
- Launched DataDistillr in October 2020



National Security Agency

Central Intelligence Agency

Deutsche Bank

black hat

JPMorgan Chase & Co.

Apache Software Foundation 20th Anniversary

Booz Allen

Why SQL?

- SQL is a standard language for data definition and data manipulation
- It is in widespread use in many database and a growing number of big data platforms
- SQL is relatively easy to use and very powerful.

What is Drill?

Apache Drill is a MPP Query Layer for big data

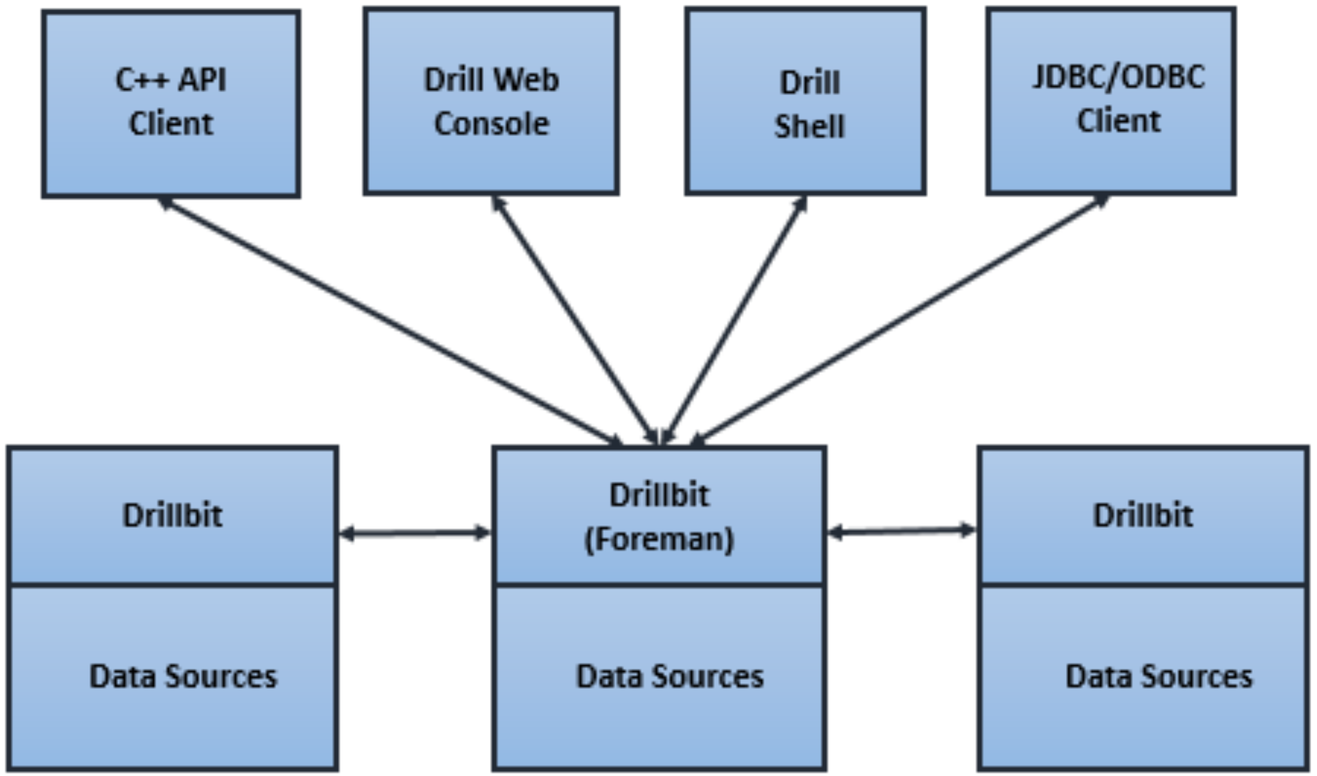
Apache Drill Does Not Require Schema Definitions

Drill lets you query data where it is

Drill can query many different file types and systems

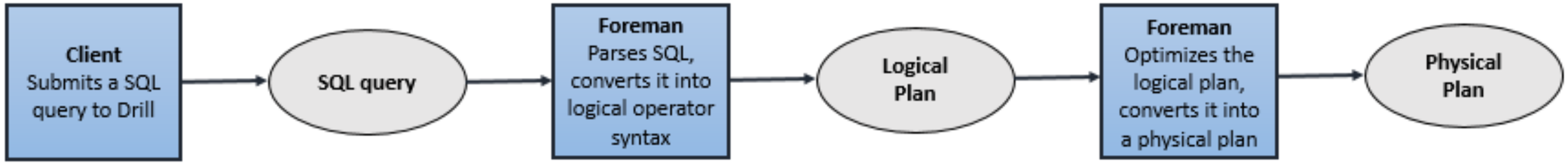
Anything you can query, you can join

How Drill Works



https://drill.apache.org/docs/drill-query-execution/

How Drill Works



https://drill.apache.org/docs/drill-query-execution/

Drill and HDF5

- As of Drill 1.18.0, Drill can natively query HDF5 files.
- Since HDF5 files are "a filesystem within a file" the way it works is a little tricky... Let's take a look.

Configuring Drill for HDF5

- Drill allows you to connect to various file systems such as HDFS, S3 and other object stores, etc.
- You can read about how to connect to the various file systems which Drill supports here: https://drill.apache.org/docs/querying-a-file-system-introduction/
- The file system configuration allows you to configure what file types are supported, workspaces etc. The sample config below demonstrates how to configure a file system to query HDF5.

```
"hdf5": {
      "type": "hdf5",
      "extensions": [
        "h5"
      ],
      "defaultPath": null,
      "showPreview": false
    }
```

Metadata Queries

- Drill has two query modes for HDF5, Metadata and Dataset queries. A metadata query should be used for exploring files and the dataset queries should be used to query actual datasets
- You can create a dataset query by setting the defaultPath variable to a dataset in the HDF5 file.

```
SELECT * FROM dfs.test.`dset.h5`;
```

```
apache drill> select * from dfs.test.`dset.h5`;
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
| path  | data_type | file_name | data_size | element_count | is_timestamp | is_time_duration | dataset_data_type | dimensions |                                        int_data                                         |
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
| /dset | DATASET   | dset.h5   | 96        | 24            | false        | false           | INTEGER          | [4, 6]     | [[1,2,3,4,5,6],[7,8,9,10,11,12],[13,14,15,16,17,18],[19,20,21,22,23,24]] |
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
```

https://github.com/apache/drill/tree/master/contrib/format-hdf5

Metadata Queries

- Drill has two query modes for HDF5, Metadata and Dataset queries. A metadata query should be used for exploring files and the dataset queries should be used to query actual datasets
- You can create a dataset query by setting the defaultPath variable to a dataset in the HDF5 file.

```
SELECT * FROM dfs.test.`dset.h5`;
```

```
apache drill> select * from dfs.test.`dset.h5`;
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
| path  | data_type | file_name | data_size | element_count | is_timestamp | is_time_duration | dataset_data_type | dimensions |                                        int_data                                         |
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
| /dset | DATASET   | dset.h5   | 96        | 24            | false        | false           | INTEGER          | [4, 6]     | [[1,2,3,4,5,6],[7,8,9,10,11,12],[13,14,15,16,17,18],[19,20,21,22,23,24]] |
+-------+-----------+-----------+-----------+---------------+--------------+-----------------+------------------+------------+-----------------------------------------------------------------------------------------+
```

https://github.com/apache/drill/tree/master/contrib/format-hdf5

Dataset Queries

- By supplying a value for the `defaultPath` variable, Drill will parse only that dataset in the HDF5 file and return the results as a queryable table.
- It is important to use the dataset query when actually working with datasets as the performance will be significantly better than viewing the data via the Metadata view.

```
SELECT *
FROM table(dfs.test.`dset.h5` (type => 'hdf5', defaultPath => '/dset'))
```

int_col_0	int_col_1	int_col_2	int_col_3	int_col_4	int_col_5
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24

https://github.com/apache/drill/tree/master/contrib/format-hdf5

Drill's Limitations

- Drill cannot read unsigned 64 bit integers
- While Drill can read nested data of arbitrary dimensions, the current implementation of the HDF5 reader for Drill does not support datasets of greater than 2 dimensions. Any datasets with more than 2 dimensions are flattened.
- Drill's implementation of HDF5 compound data types is somewhat limited

https://github.com/apache/drill/tree/master/contrib/format-hdf5

Future Improvements

- Improved pushdown capabilities to enable faster reads
- Streaming reader to improve performance on object stores and distributed file systems
- Improvements to compound data type parsing
- Possible parallelization of reads

Questions?

Charles Givre: charles@datadistillr.com

Thank You!