DATA+AI SUMMIT 2022

DBA Perspective

Optimizing Performance Table-by-Table

Douglas Moore

Senior Specialist Solutions Architect, Databricks

ORGANIZED BY databricks

6 Ways to know if you are a DBA:

1. You have a
   - ➢ Data Lakehouse
   - ➢ Data Lake
   - ➢ Data Warehouse

2.Data performance & service levels

3.Sizing & costs

4.Data access

5.Get excited about how the Lakehouse works?

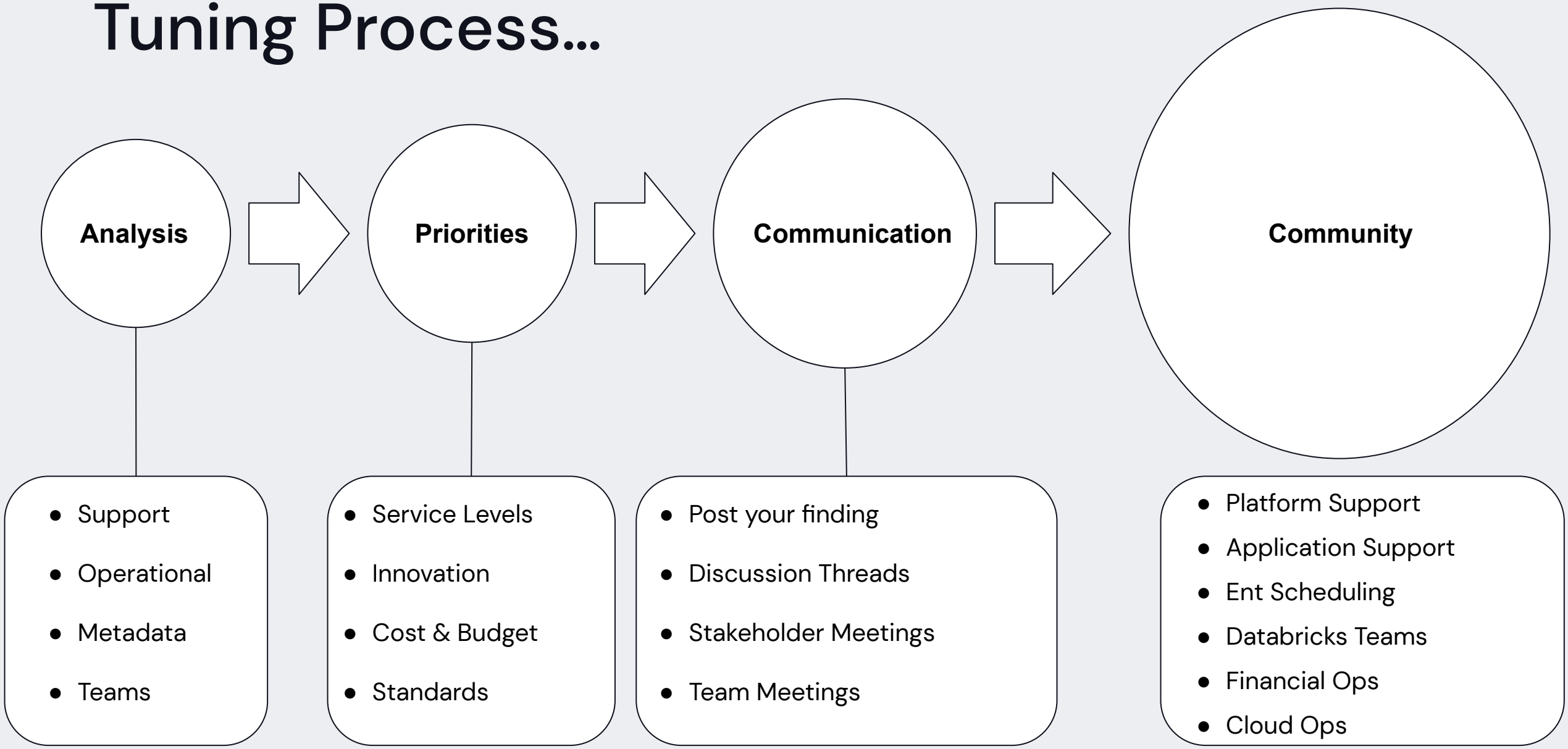

6. Complaint Department

There are lots of great talks about performance tuning, one job at a time.

How does a DBA scale?

Scalable Performance Tuning Process

Tuning Process…



Analyze

Data Inputs for priorities, trends, outliers

Support Channels

Service Now

Direct E-mail

Discussion Groups

“Executive Feedback”

Metadata

Table Details

File Metadata

Transaction Log

Users & Groups & Grants

Operational Logs

Job History

Enterprise Scheduler

Cluster Logs

Audit Logs

Usage Logs

Let’s gather and **analyze** the numbers

Analyze

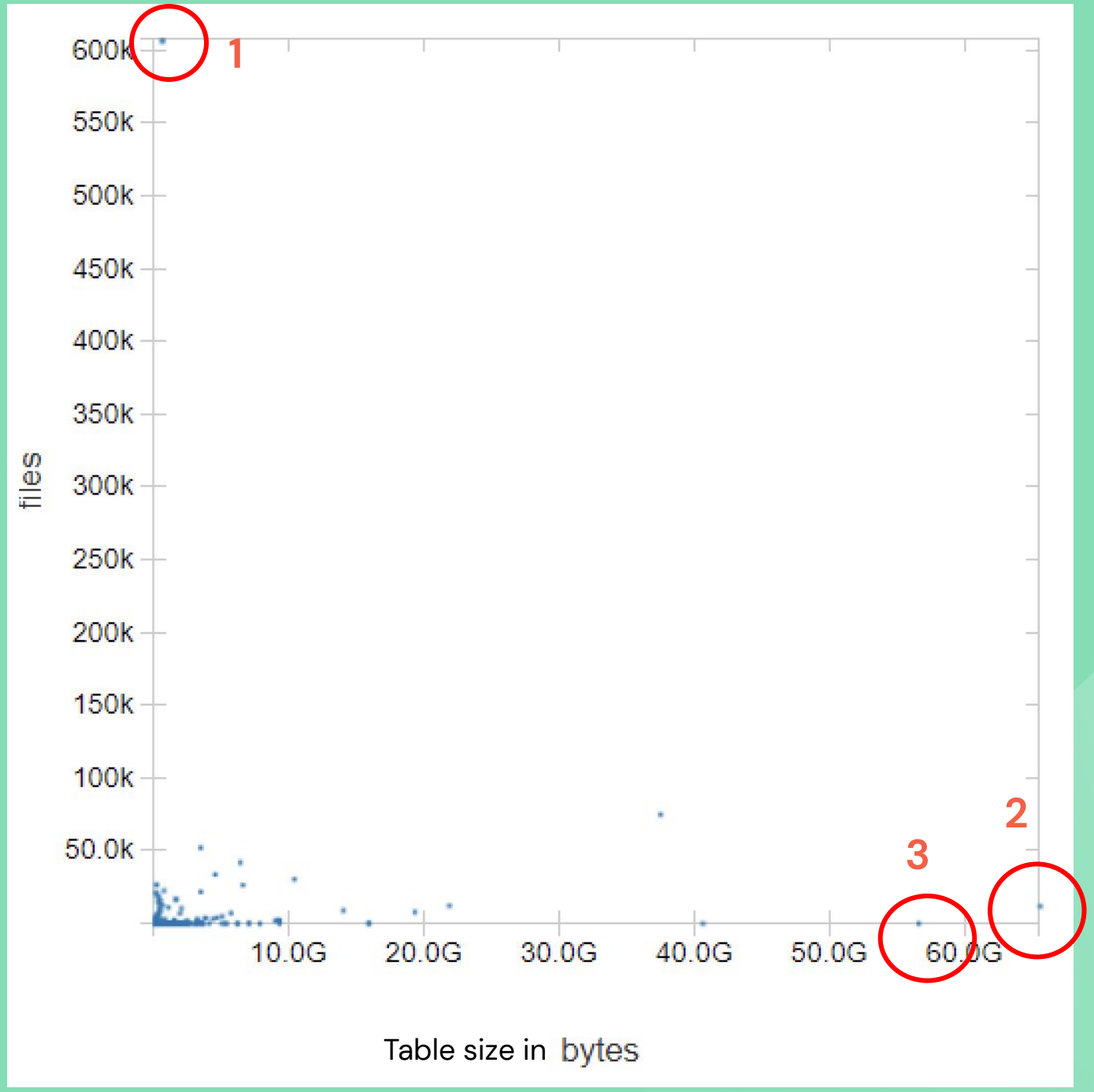
Run DESCRIBE DETAIL in parallel

github.com/dmoore247/DBA-Helper/

```
1  def table_detail_func(row) -> None:
2    """Capture DESCRIBE DETAIL metadata"""
3    if row['isTemporary'] != 'true':
4      (spark.sql(F"DESCRIBE DETAIL {row['database']}.{row['tableName']}")
5              .write.format('delta')
6              .mode('append').option('mergeSchema','true')
7              .saveAsTable("dba_helper.table_details"))
8
9
10 run_parallel(table_detail_func, "show tables in default")
```

DESCRIBE DETAILS metadata

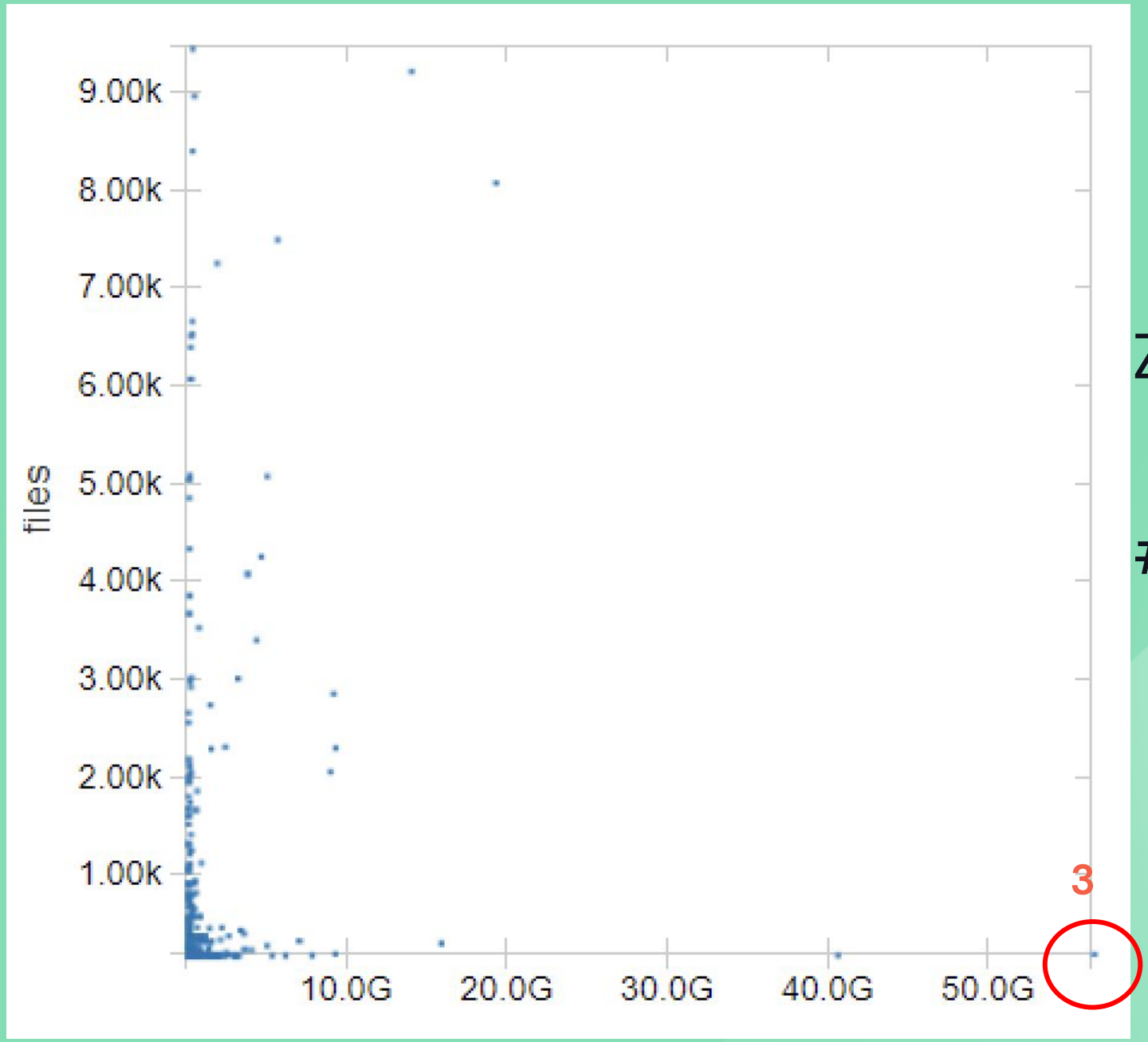
```
%sql
CREATE OR REPLACE TABLE ${c.database_name}.table_details (
  databaseName STRING,
  tableName STRING,
  format STRING,
  id STRING,
  name STRING,
  description STRING,
  location STRING,
  createdAt TIMESTAMP,
  lastModified TIMESTAMP,
  partitionColumns ARRAY<STRING>,
  numFiles BIGINT,
  sizeInBytes BIGINT,
  properties MAP<STRING, STRING>,
  minReaderVersion INT,
  minWriterVersion INT
  )
USING delta
```



#1 too many small files

#2 too few files?

#3 definitely too few files

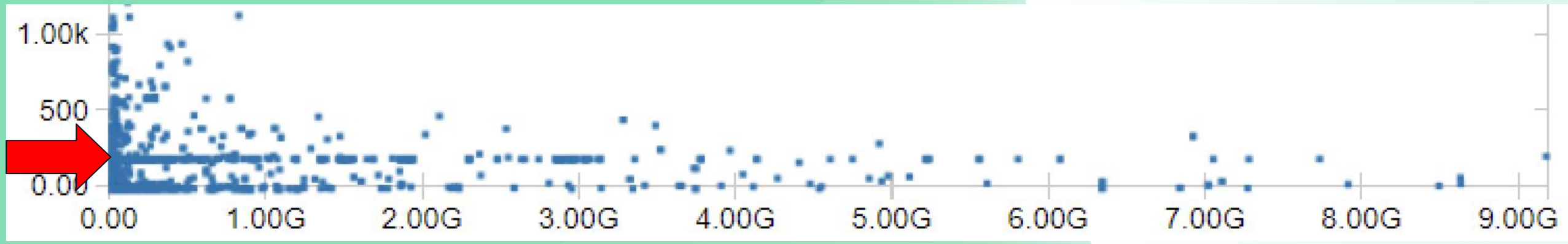


Zoom in...

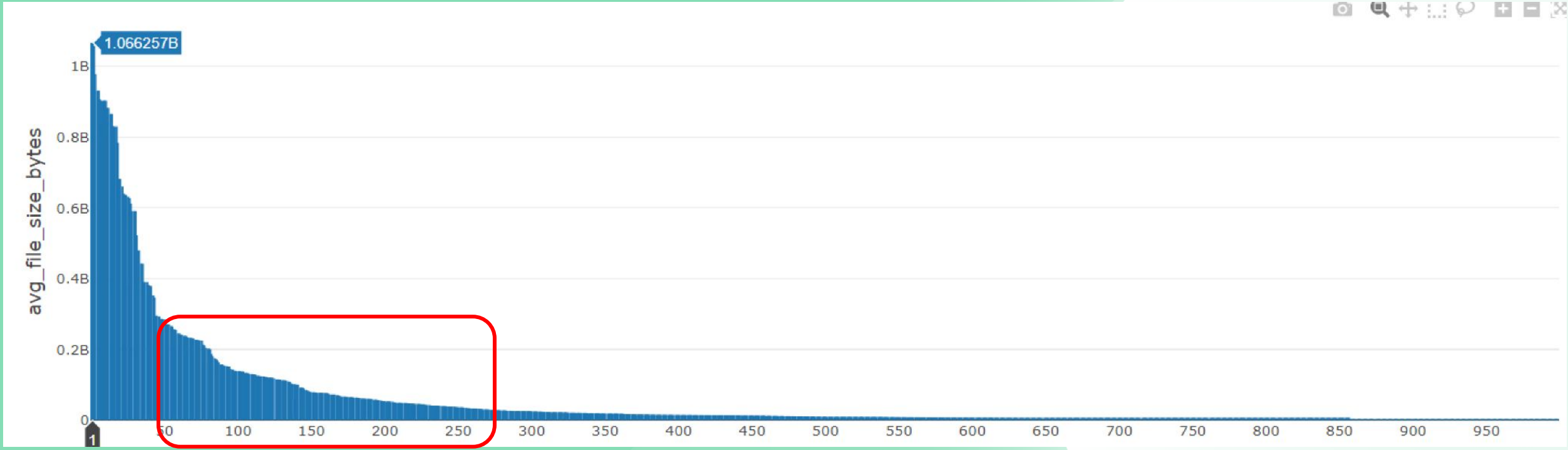
#3 definitely too few files

Curious pattern at

#files == 200

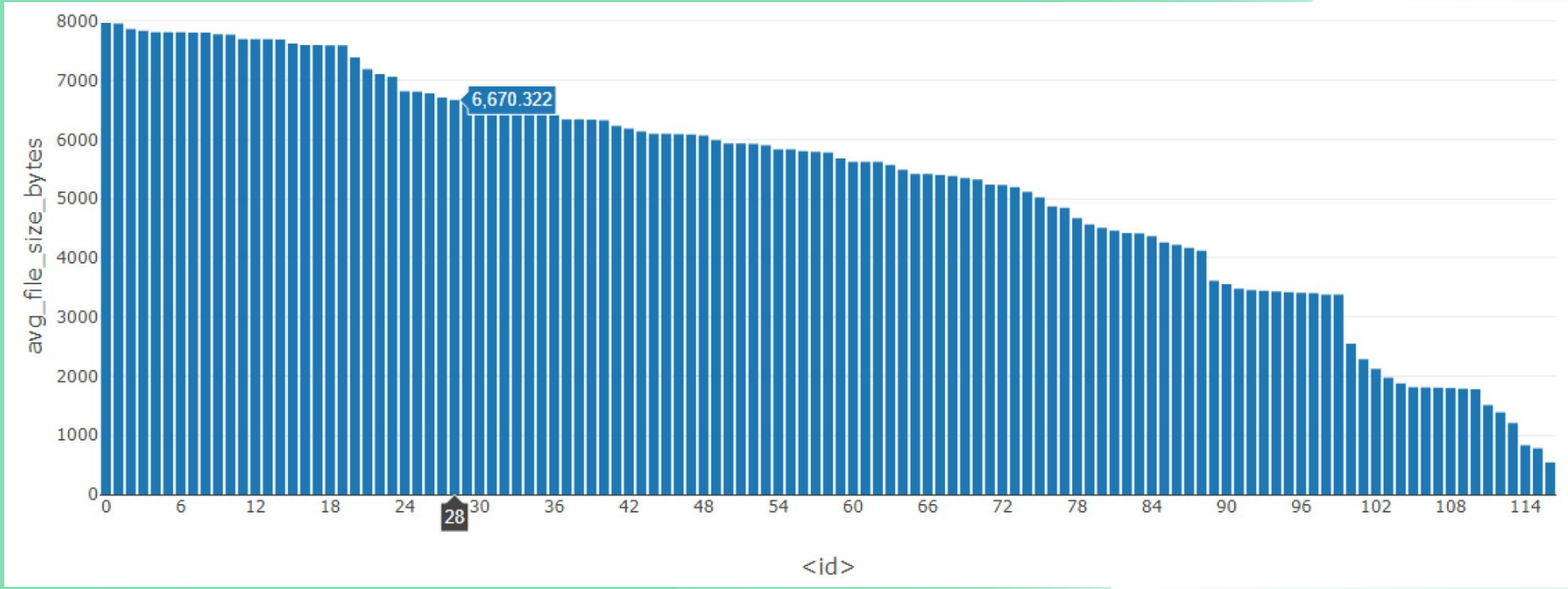


Average File Size

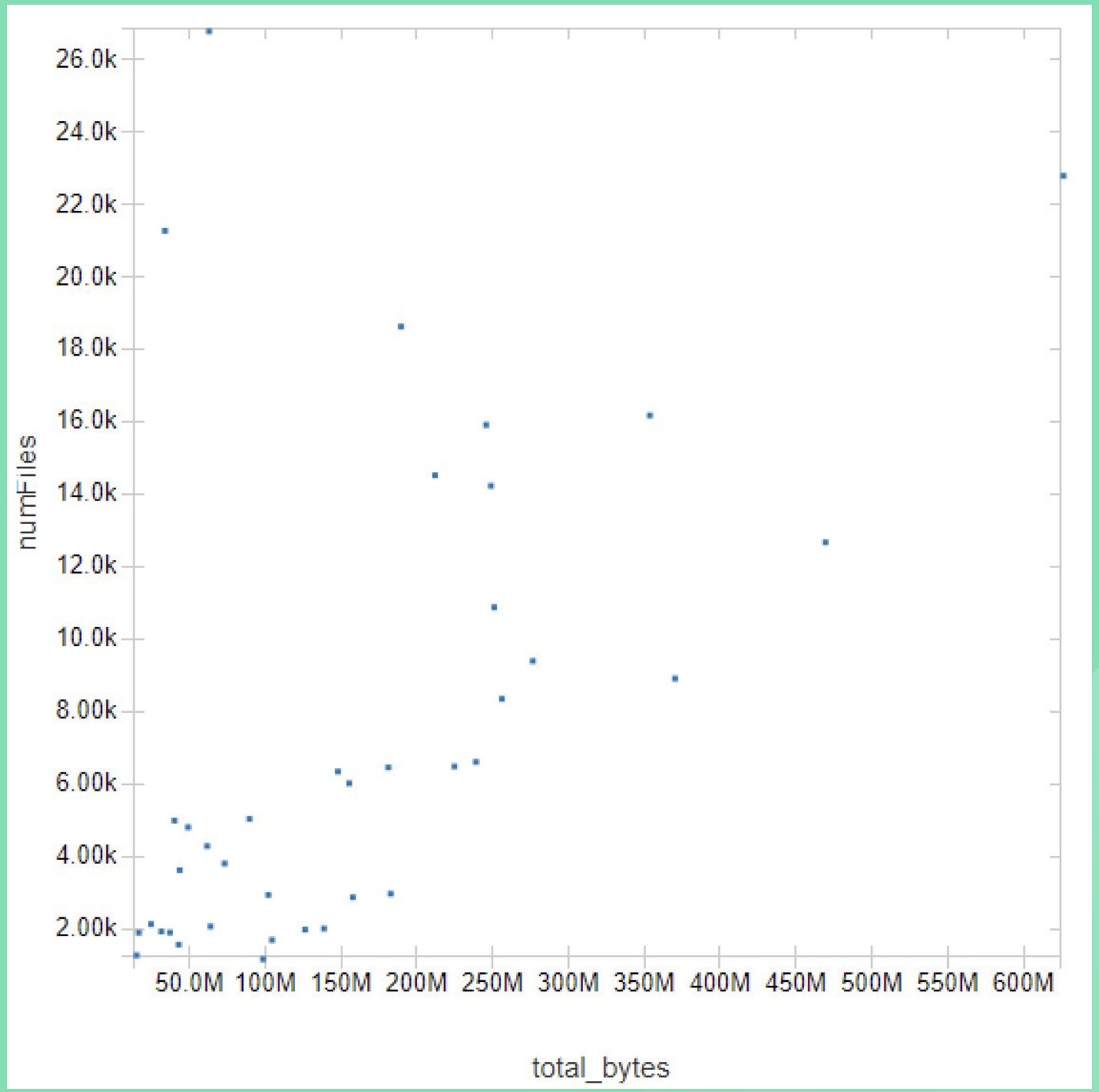


Sweet spot

Average File Size…



Very long tail of small tables with very small files



Small tables

Many files

Solutions

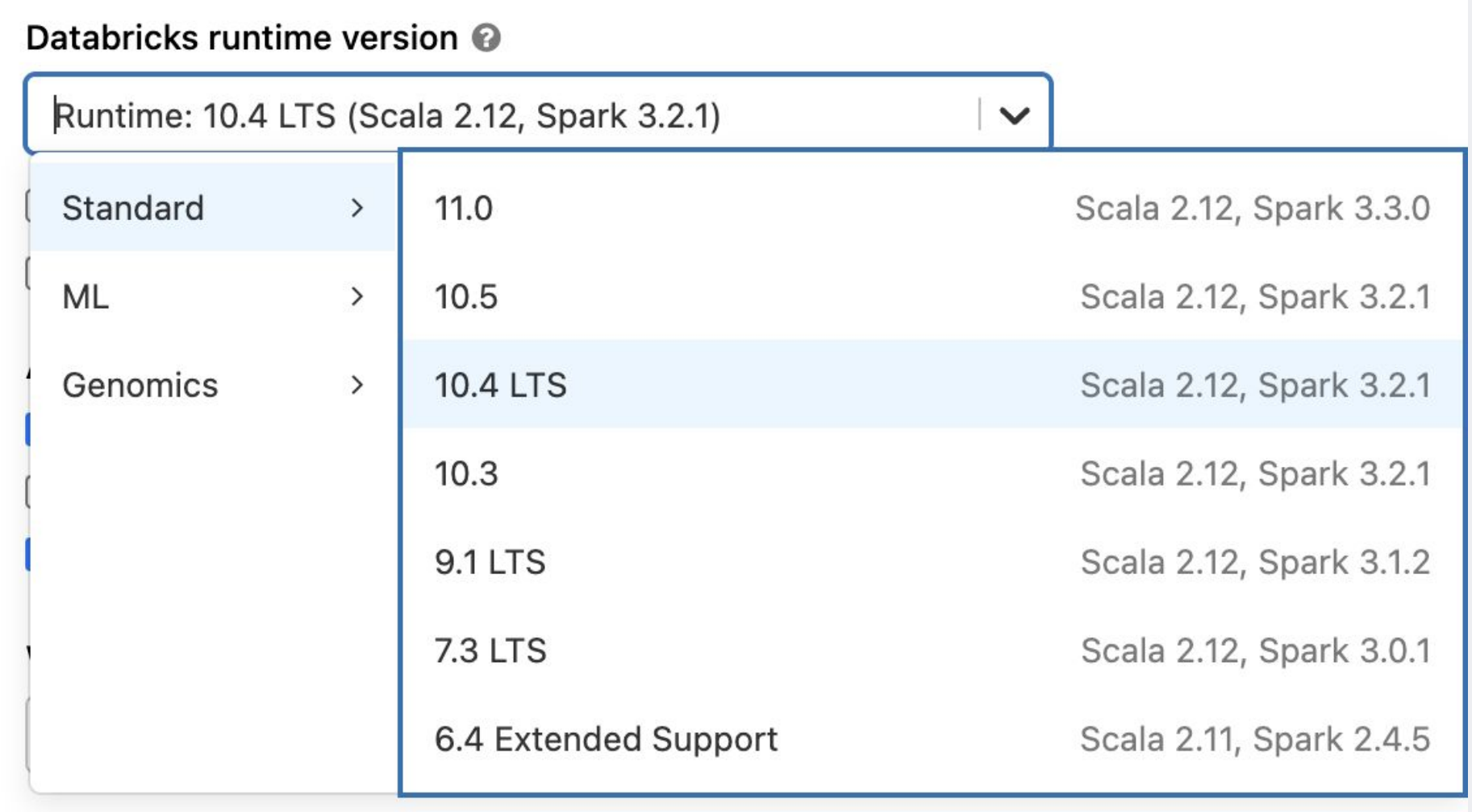
1. Upgrade!

DBR 10.4LTS is amazing!

Performance is more automatic with fewer knobs to turn

Compute

Keep your runtime versions up to date



2. Remove ‘old’ overrides
(they may be no longer useful)

Remove ‘old’ settings

Just comment them out!

```
1   -- AQE works wonders in determining the right # partitions
2   -- SET spark.sql.shuffle.partitions = 200
3
4   # df.repartition(32)
5   # df.coalesce(32)
6
7   # df.cache()
8
9   -- PARTITIONED BY
10
```

3. OPTIMIZE Your tables

Configure

Configure your your medium to large tables

```
--one time
ALTER TABLE <table_name>
SET TBLPROPERTIES (
  delta.autoOptimize.optimizeWrite = true,
  delta.autoOptimize.autoCompact = true,      -- 'auto' if > DBR 10.1
  delta.targetFileSize = 33554432,            -- optimize your queries,joins,merges
  delta.tuneFileSizesForRewrites = true,      -- If merge heavy table
  delta.dataSkippingNumIndexedCols = 5,       -- optimize your writes
  delta.deletedFileRetentionDuration = "interval 7 days",
  delta.logRetentionDuration          = "interval 60 days"
);
```


https://docs.databricks.com/delta/optimizations/file-mgmt.html

OPTIMIZE – ZORDER BY

ZORDER BY - build N dimensional clustered index

```
1
2  OPTIMIZE <table_name> [WHERE predicate]
3    [ZORDER BY (<join_key> [, ...], <predicate_col> ) ] -- up to 5 cols
4
5
6
7
8
9
10
```

Run `OPTIMIZE` daily basis when spot prices are low

Use compute optimized instances

ANALYZE

Compute statistics

```
1  --periodically
2
3
4  -- Optimizes that first query plan before AQE kicks in on later stages
5  ANALYZE TABLE <table_name> COMPUTE STATISTICS FOR ALL COLUMNS
6
7
8
9
10
```

VACUUM

Clean up unused files

```
--run periodically to reduce cloud storage costs, boost GDPR compliance


VACUUM <table_name> [RETAIN num HOURS] [DRY RUN]


```

Summary

Setup a maintenance schedule

```
1   -- Summary
2
3   -- 1. ALTER TABLE      -- Once
4   -- 2. OPTIMIZE     -- Periodically
5   -- 3. ANALYZE TABLE    -- Periodically
6   -- 4. VACUUM           -- Periodically
7
8
9
10
```

4.
Build Community

Building Community

Scale yourself through others

Home Page

Discussion Groups

Learning Series

“Wins” board

User Training

Champions

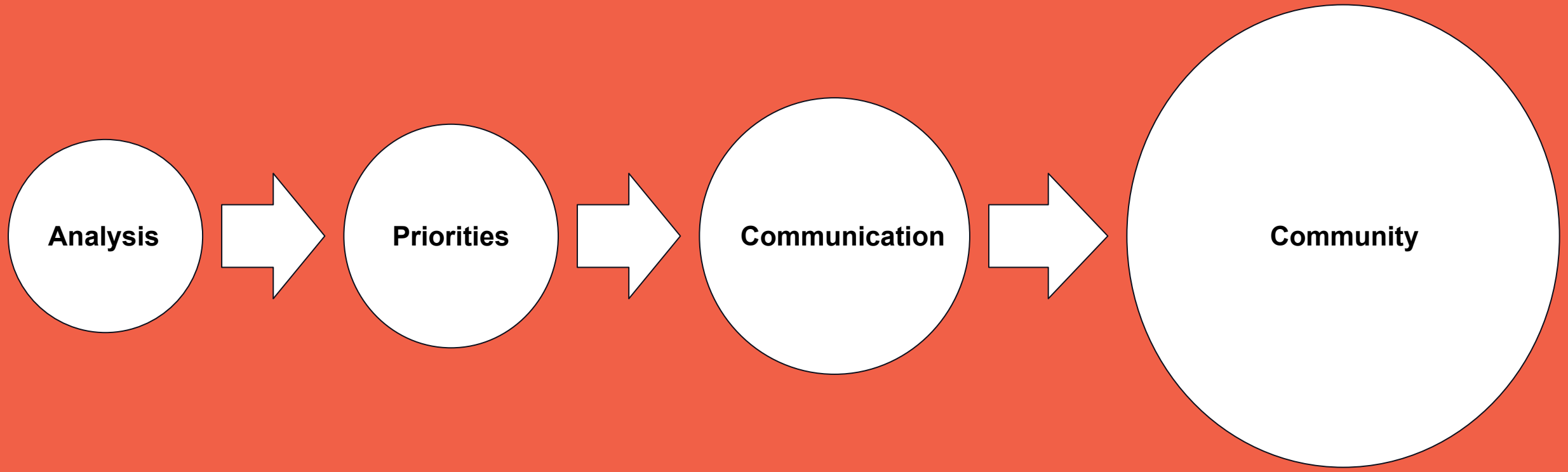
Solution Patterns

Solution Reviews

Monthly Operations Review with Stakeholders

5. Summary

As a DBA, it's all about scaling yourself



Bonus

Run SQL commands in parallel

```
1  from multiprocessing.pool import ThreadPool
2  import multiprocessing as mp
3
4  def run_parallel(func, list_query) -> None:
5    lst = spark.sql(list_query).collect()
6    if len(lst) > 0:
7      cpus = mp.cpu_count()
8      with ThreadPool(cpus) as p:
9        p.map(func, lst)
10
```

Catalog your data

- Make it ready for volumetric analysis

Cmd 8

```
# Gather file metadata from all files under current_path
def catalog(current_path:str):
  return (spark.read.format("binaryFile")
        .option("recursiveFileLookup", "true")
        .option("pathGlobFilter", "*.csv")
        .load(current_path)
        .drop('content'))

df = catalog('s3a://databricks-corp-training/common/')
display(df)
```

▸ (5) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [path: string, modificationTime: timestamp ... 1 more fi

Table | Data Profile

	path	modificationTime	length
1	s3a://databricks-corp-training/common/asa/flights/all.csv	2017-09-30T00:21:13.000+0000	12037151361
2	s3a://databricks-corp-training/common/asa/flights/1990-1999.csv	2017-09-30T00:14:57.000+0000	5181408817
3	s3a://databricks-corp-training/common/EDGAR-Log-20170329/EDGAR-Log-20170329.csv	2018-01-26T19:19:34.000+0000	3454449785
4	s3a://databricks-corp-training/common/asa/flights/2000-2004.csv	2017-09-30T00:19:17.000+0000	3005636811
5	s3a://databricks-corp-training/common/asa/flights/2005-2008.csv	2017-09-30T00:15:14.000+0000	2735385998

Bonus

Collect file metadata, analyze it for:

- Files / hour
- Bytes / hour
- Memory Constraints
- Large / Small file issues
- Seasonality
- Capacity requirements
- Cluster sizing
- Partition imbalance

DATA+AI SUMMIT 2022

DBA Perspective

Optimizing Performance Table-by-Table

github.com/dmoore247/DBA-Helper/

Douglas Moore

Senior Specialist Solutions Architect, Databricks