Architecting a Low-Latency Schemaless SQL Engine

Igor Canadi, Rockset

[ROCKSET]

About

Rockset

- Search and analytics engine
- Enables data-driven applications

Igor

- Rockset
- Facebook
- RocksDB
- GraphQL

Overview

- Hardware and people efficiency
- Designing systems for people efficiency
  - Schemaless SQL
  - Converged indexing
  - Serverless architecture

Hardware Efficiency

TigerGraph Announces Free Developer Edition of the World's Fastest Graph Database

World's Fastest & Most Advanced GPU Database

Brytlyt combines the power of GPUs with patent pending IP and is built on PostgreSQL.

VoltDB Unveils v7.0, The World's Fastest Enterprise-Class Database for Powering Critical Data-Driven Actions Across Industries, in Real Time

EXASOL: BUILDING THE FASTEST DATABASE IN THE WORLD

ScyllaDB Announces General Availability of World's Fastest NoSQL Database

ROCKSET

Aerospike Unveils New Features to World's Fastest Database Powering Next-Gen Apps

SQL Server 2017: Fast, faster, and the fastest database everywhere you need it

MemSQL Launches Community Edition: World's Fastest In-Memory Database Now Available to All

IBM Delivers World's Fastest Database

ROCKSET

Hardware Efficiency

- Faster databases ~= less hardware
- How much hardware do I need?
- Important, but not the only thing that matters

People Efficiency

People Efficiency

- How many people do I need?
- How much time do I need?

People Efficiency - Configuration

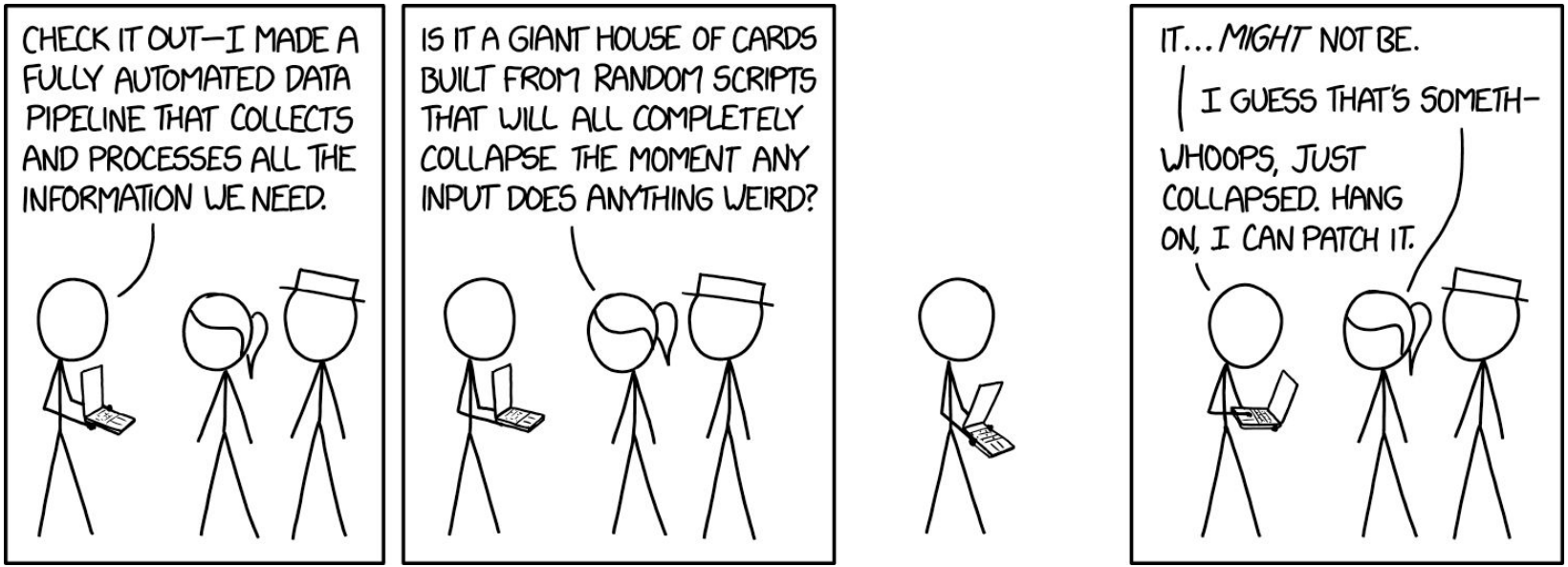
“My query is slow”

“Do you have an index?”
“What’s your partition key?”
“What’s your buffer size?”
“You should hire a DBA”

People Efficiency - Organizational Friction

- Pre-cloud era: Application developers blocked on provisioning
- Data scientists blocked on data engineers

People Efficiency - Pipelines



Hardware vs. People Efficiency

- Hardware is frequently cheaper than people
- Increase hardware efficiency - spend less money
- Increase people efficiency - spark creativity

Designing Systems for People Efficiency

[ROCKSET]

Rockset

- Search and analytics engine
- “Shortest path from data to applications”
- Connect to data sources or streams
- Execute fast queries

Schemaless SQL

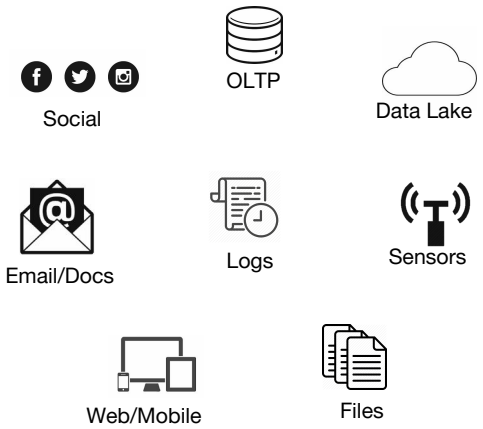
Choosing the Query Language

- SQL is the obvious choice
- Maximize usefulness
- Existing knowledge
- Ecosystem of tools

Querying existing data sources

ROCKSET

Querying existing data sources



SQL

...but first, let me define a schema

ROCKSET

SQL Schema

- Drag on people efficiency
- Messy data
- Complex ETL jobs

Schemaless SQL

- “Smart schema”
- Frictionless data onboarding
- Data scientists no longer blocked on data engineers
- Performance overhead?

https://rockset.com/blog/using-smart-schema-to-accelerate-insights-from-nested-json/

Schemaless SQL - Storage

ROCKSET

Schemaless SQL - Query Execution

Rows

Strict schema

1	10	7	4	5
a	b	c	d	e

Columns

Schemaless

I 1	I 10	I 7	I 4	I 5
S a	S b	I 3	I 5	S e

Columns

Schemaless
(with type hoisting)

I	1	10	7	4	5
M	S a	S b	I 3	I 5	S e

Columns

ROCKSET

Schemaless SQL

- Superior user experience
- Field interning reduces storage overhead
- Type hoisting reduces query execution overhead

Converged indexing

Converged Indexing

- “Query is slow because of the missing index”

Converged Indexing

- “Query is slow because of the missing index”

Index all the fields!

Background on Indexing

- Columnar storage
- Search indexing

Columnar Storage

- Store each column separately
- Great compression
- Only fetch columns the query needs

Columnar Storage

- Store each column separately
- Great compression
- Only fetch columns the query needs

```
<doc 0>
{
  "name": "Igor",
  "interests": ["databases", "snowboarding"],
  "last_active": 2019/3/15
}

<doc 1>
{
  "name": "Dhruba",
  "interests": ["cars", "databases"],
  "last_active": 2019/3/22
}
```

"name"

0	Igor
1	Dhruba

"interests"

0.0	databases
0.1	snowboarding
1.0	cars
1.1	databases

"last_active"

0	2019/3/15
1	2019/3/22

Columnar Storage

Advantages

- Cost effective
- Narrow queries, wide tables
- Scan queries
- Analytical queries

Disadvantages

- High write latency
- High minimum read latency
- Not suitable for online applications

Search Indexing

- For each value, store documents containing that value (posting list)
- Quickly retrieve a list of document IDs that match a predicate

Search Indexing

- For each value, store documents containing that value (posting list)
- Quickly retrieve a list of document IDs that match a predicate

```
<doc 0>
{
  "name": "Igor",
  "interests": ["databases", "snowboarding"],
  "last_active": 2019/3/15
}

<doc 1>
{
  "name": "Dhruba",
  "interests": ["cars", "databases"],
  "last_active": 2019/3/22
}
```

"name"

Dhruba	1
Igor	0

"interests"

databases	0.0; 1.1
cars	1.0
snowboarding	0.1

"last_active"

2019/3/15	0
2019/3/22	1

Search Indexing

Advantages

- High selectivity queries
- Low latency queries
- Suitable for online applications

Disadvantages

- Slower analytical queries

Converged Indexing

- Columnar and search indexes in the same system
- Built on top of key-value store abstraction
- Each document maps to many key-value pairs

Converged Indexing

- Columnar and search indexes in the same system
- Built on top of key-value store abstraction
- Each document maps to many key-value pairs

```
<doc 0>
{
  "name": "Igor"
}

<doc 1>
{
  "name": "Dhruba"
}
```

Key	Value	
R.0.name	Igor	**Row Store**
R.1.name	Dhruba	
C.name.0	Igor	**Column Store**
C.name.1	Dhruba	
S.name.Dhruba.1		**Search index**
S.name.Igor.0		

Converged Indexing - Queries

- Fast analytical queries + fast search queries
- Optimizer picks between columnar store or search index

Converged Indexing - Queries

- Fast analytical queries + fast search queries
- Optimizer picks between columnar store or search index

```
SELECT *
FROM search_logs
WHERE keyword = 'datacouncil'
AND locale = 'en'
```

Search index

```
SELECT keyword, count(*)
FROM search_logs
GROUP BY keyword
ORDER BY count(*) DESC
```

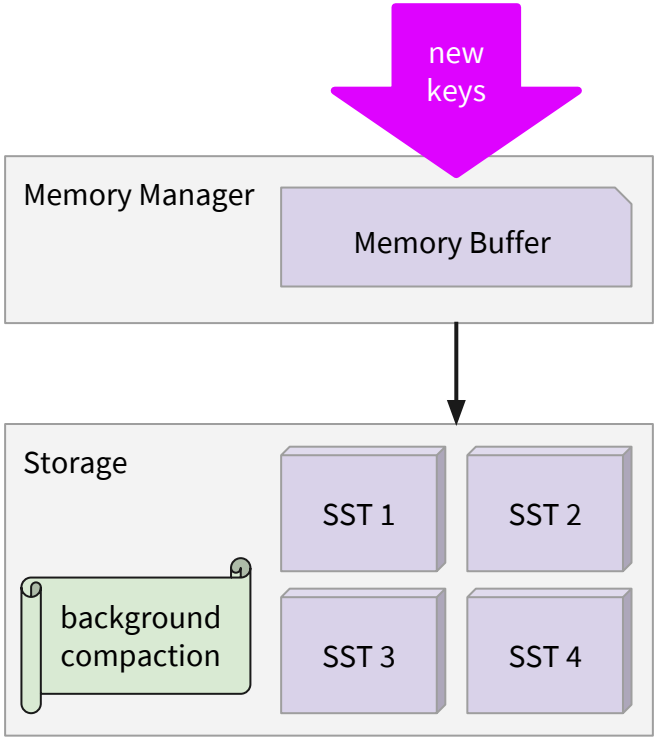
Columnar store

Converged Indexing - Writes

- One document write results in many key-value store writes
- Use write-optimized key-value store - RocksDB

Converged Indexing - Writes

- One document write results in many key-value store writes
- Use write-optimized key-value store - RocksDB



Converged Indexing

- Fast queries out of the box
- Real-time index writes

More efficient

- Database configuration
- Queries

Less efficient

- Storage
- Writes

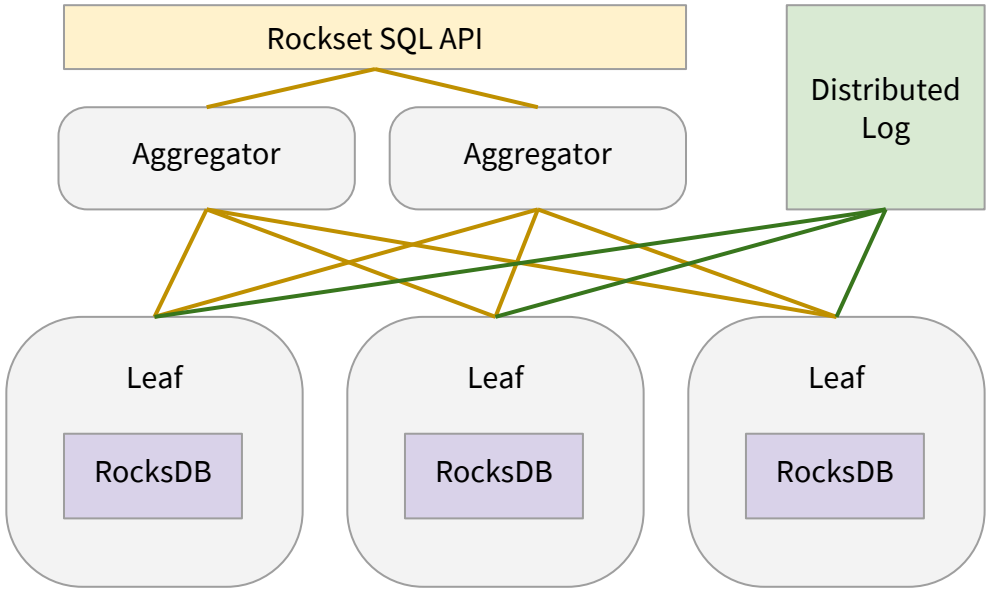
Serverless Architecture

Serverless Architecture

- Rockset is a cloud service
- No need to manage hardware
- Seamless autoscale

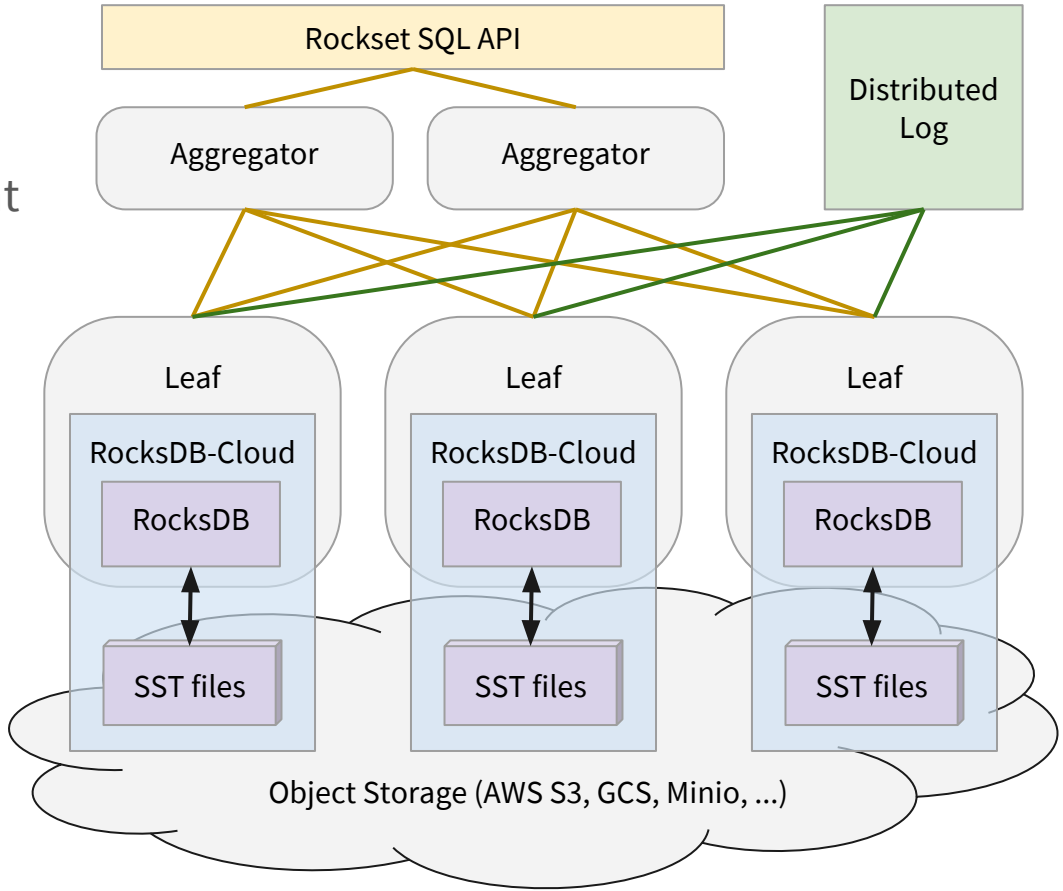
Storage in the Cloud

- Data is sharded across leaves



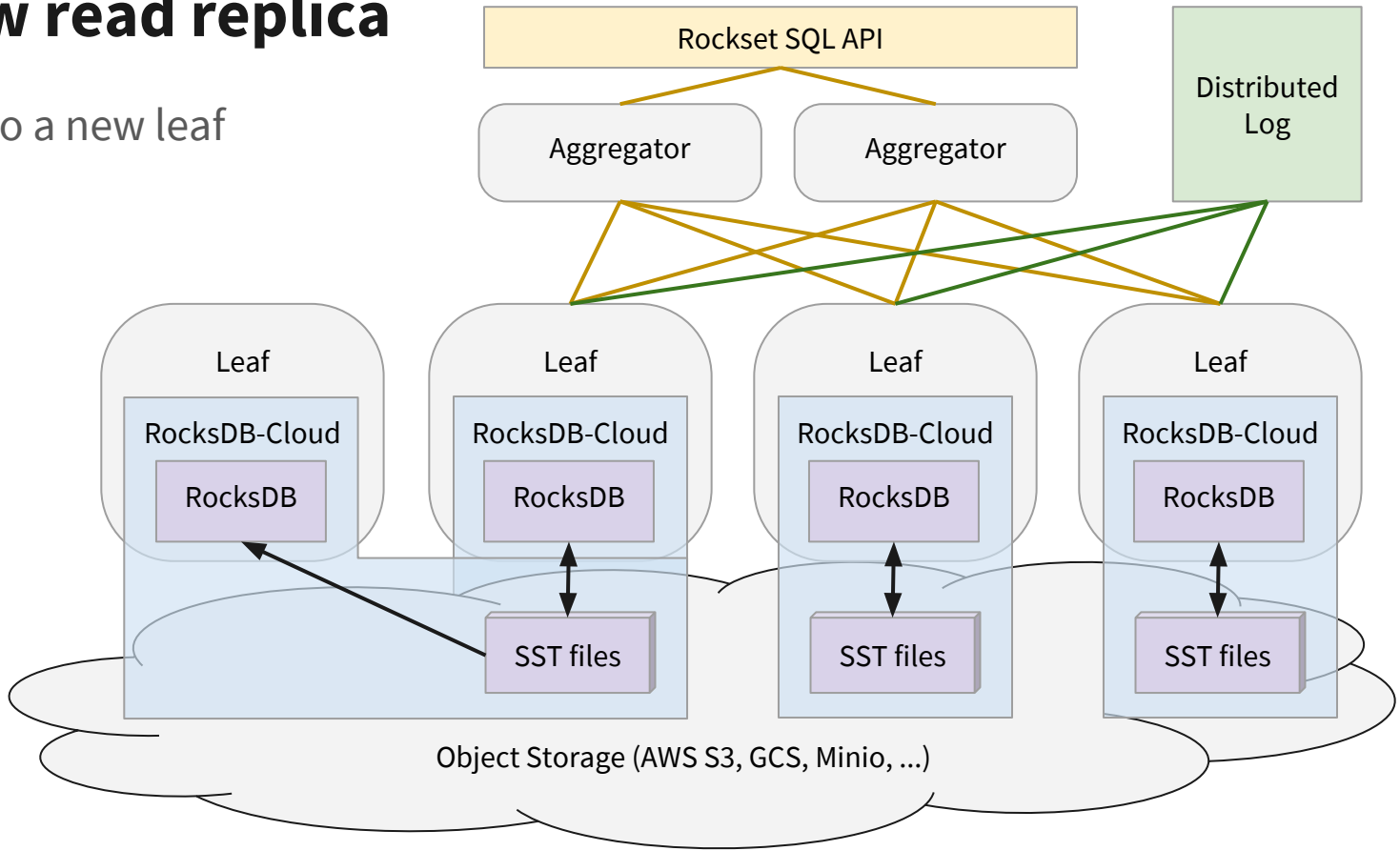
Storage in the Cloud

- Data is sharded across leaves
- RocksDB-Cloud keeps consistent copy in cloud object storage



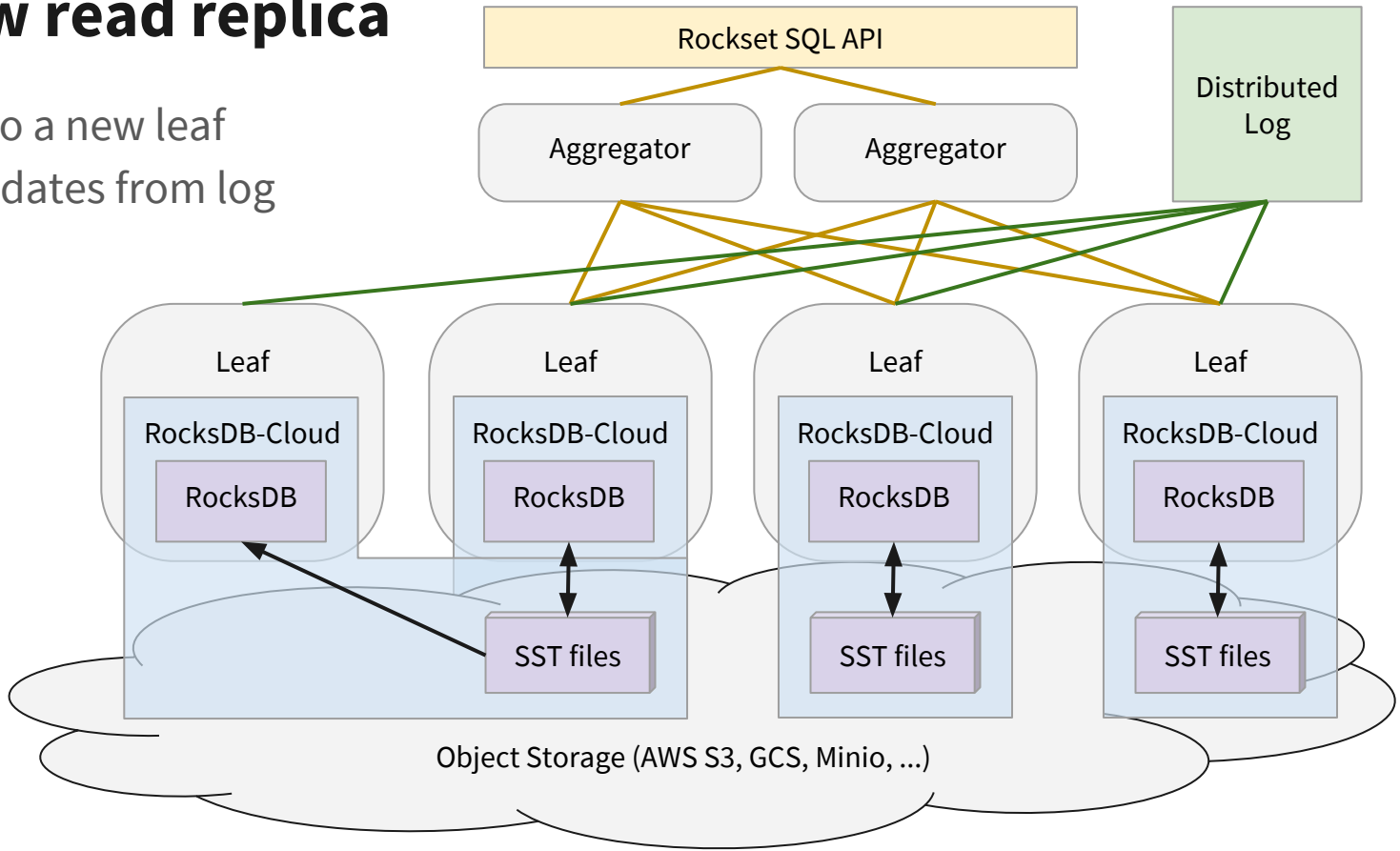
Adding new read replica

- Copy data to a new leaf



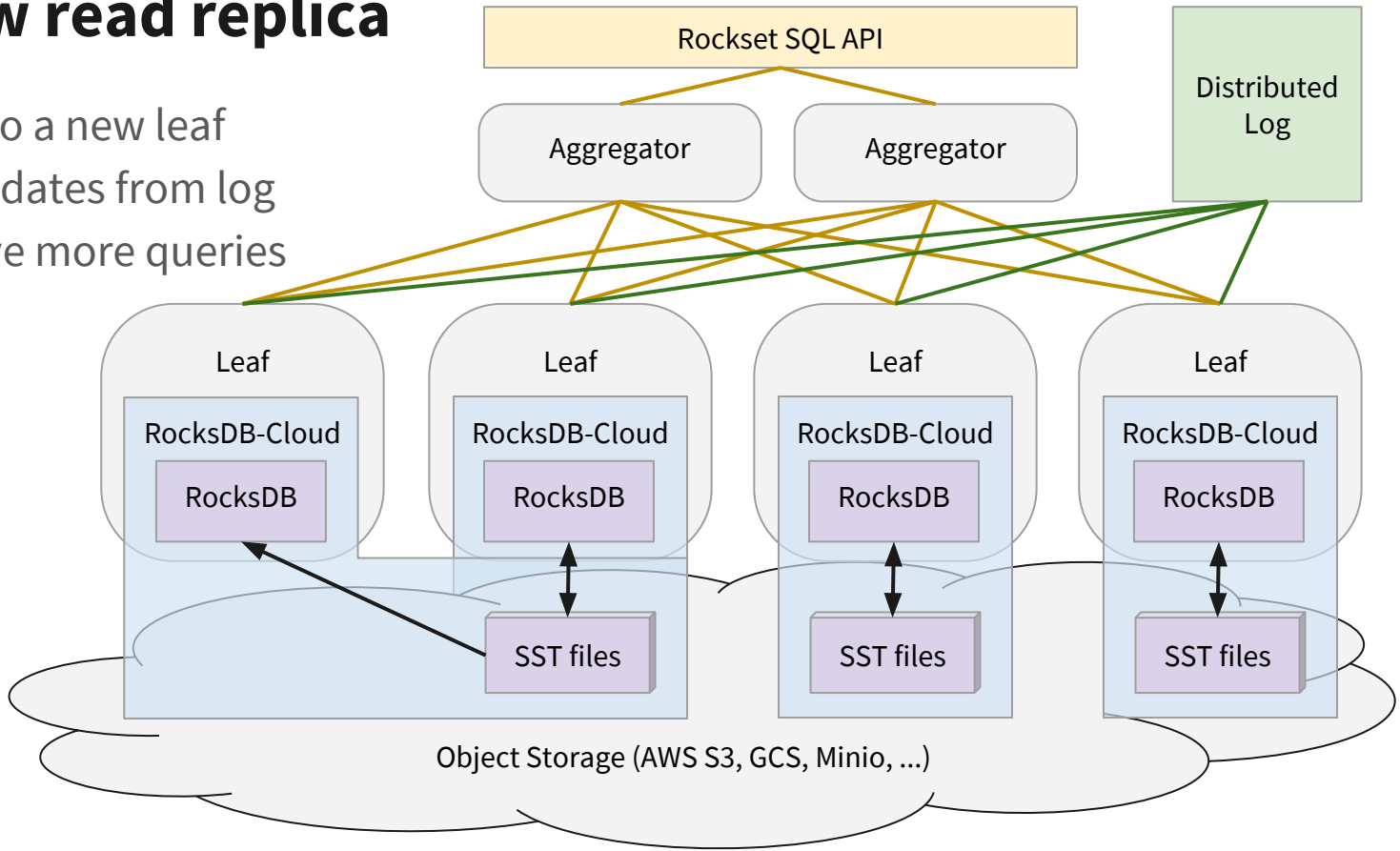
Adding new read replica

- Copy data to a new leaf
- Tail new updates from log



Adding new read replica

- Copy data to a new leaf
- Tail new updates from log
- Able to serve more queries



Conclusion

Conclusion

- Schemaless SQL
- Converged indexing
- Serverless architecture

No need to configure...

...schema

...indexes

...servers

Conclusion

- Rockset - “shortest path from data to applications”
- Making workflows easy catalyzes creativity

Thank you

[ROCKSET]