Presto Unlimited

MPP Database at Scale

Wenlei Xie

17.10.2019

Introduction

Presto: Open source distributed SQL engine for Big Data

- MPP architecture (vs. MapReduce)
- Originally designed for interactive workloads

Presto SQL language as the unified computation interface

- One of the main batch engines used in Facebook

Introduction

Presto: Open source distributed SQL engine for Big Data

- MPP architecture (vs. MapReduce)
- Originally designed for interactive workloads

Presto SQL language as the unified computation interface

- One of the main batch engines used in Facebook

Scalability Challenge

- Memory Intensive Queries (multiple TBs)
- Long-running Queries (multiple hours)

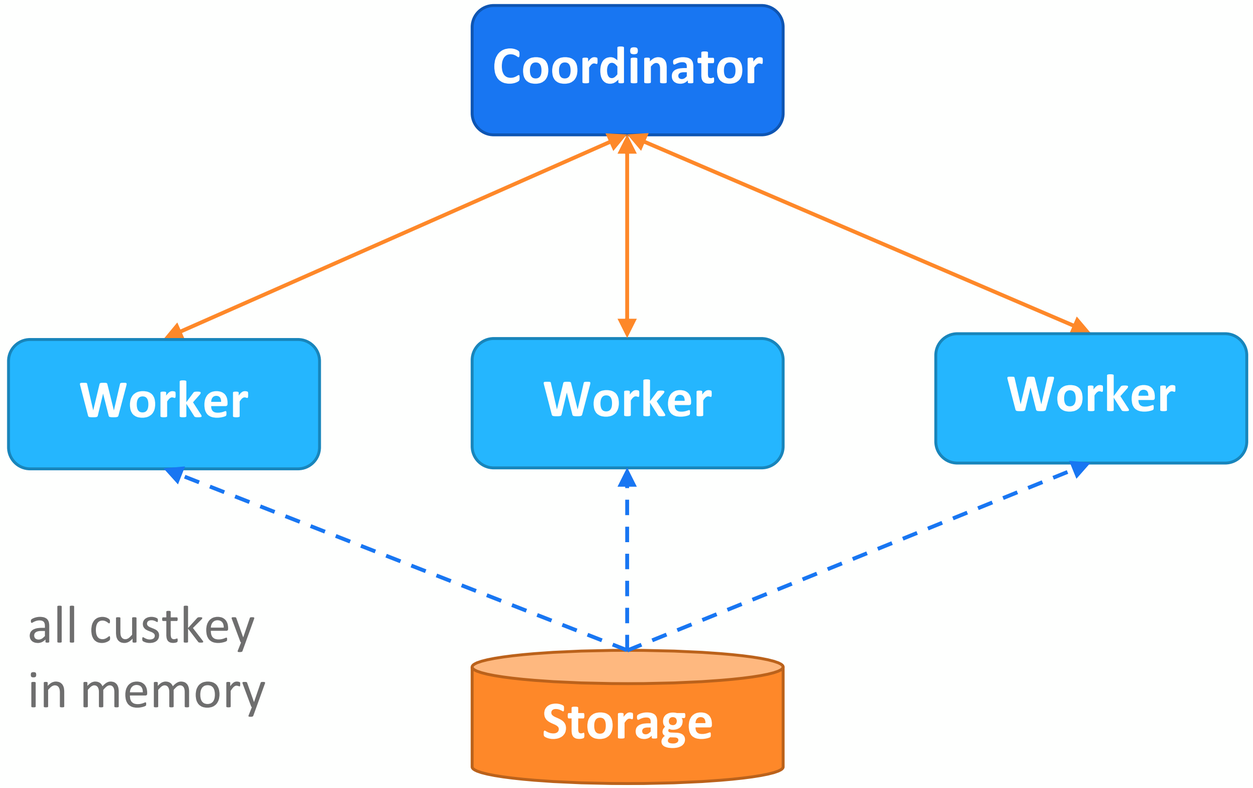
Presto Unlimited

1) Grouped Execution (#8951)
2) Lifespan Recoverability (#12124)
3) Exchange Materialization (#12387)
4) Status and Future Work

Grouped Execution

Ungrouped Execution

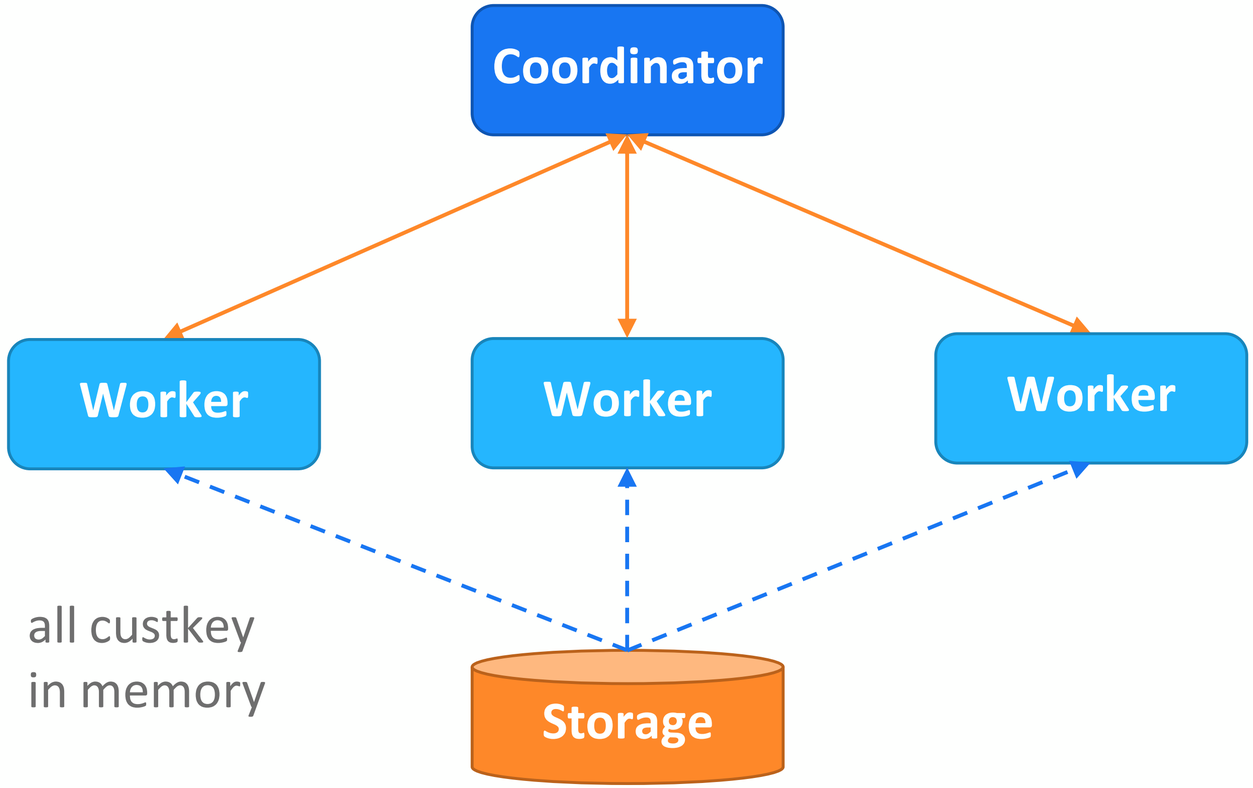
```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```



Bucketed Table

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```

What if tables are bucketed on custkey?

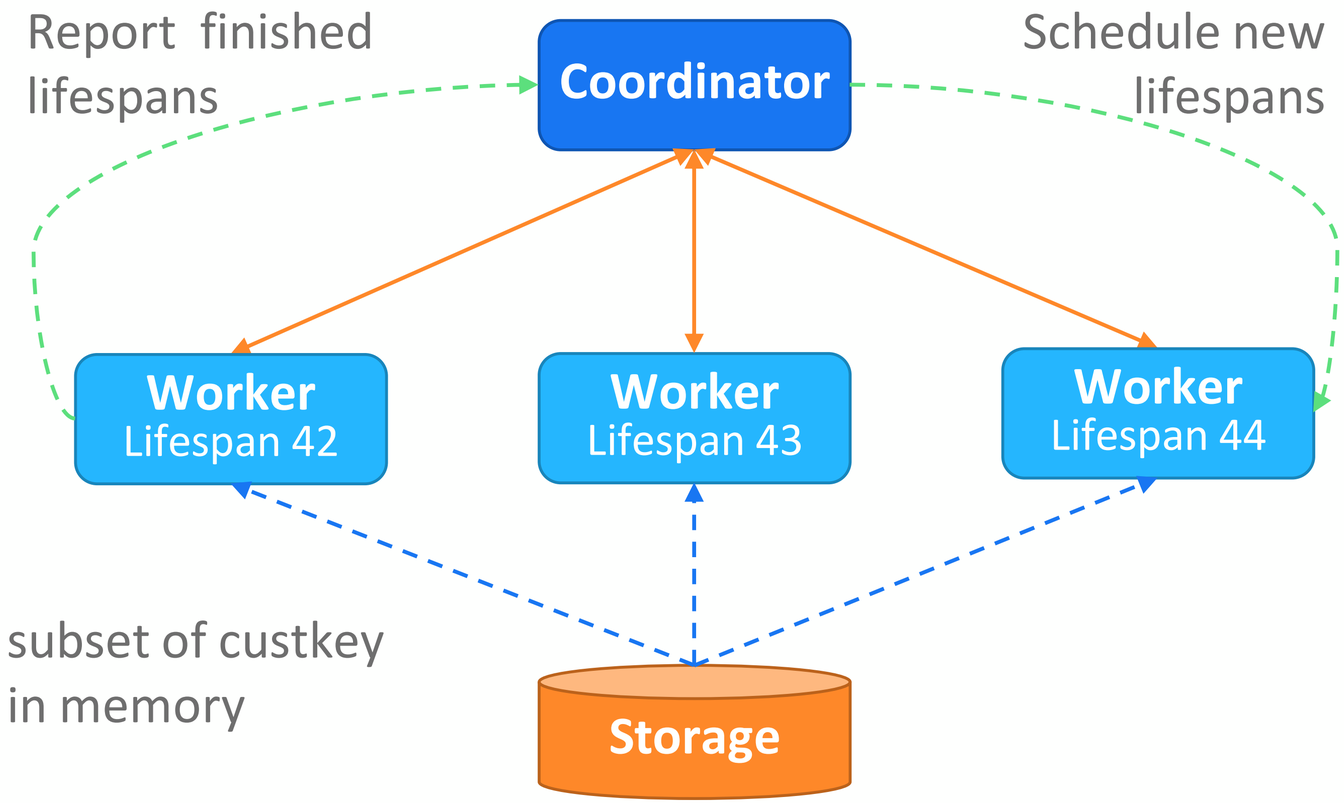


Grouped Execution

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```

What if tables are bucketed on custkey?

- Flexible schedule
- “lifespan”



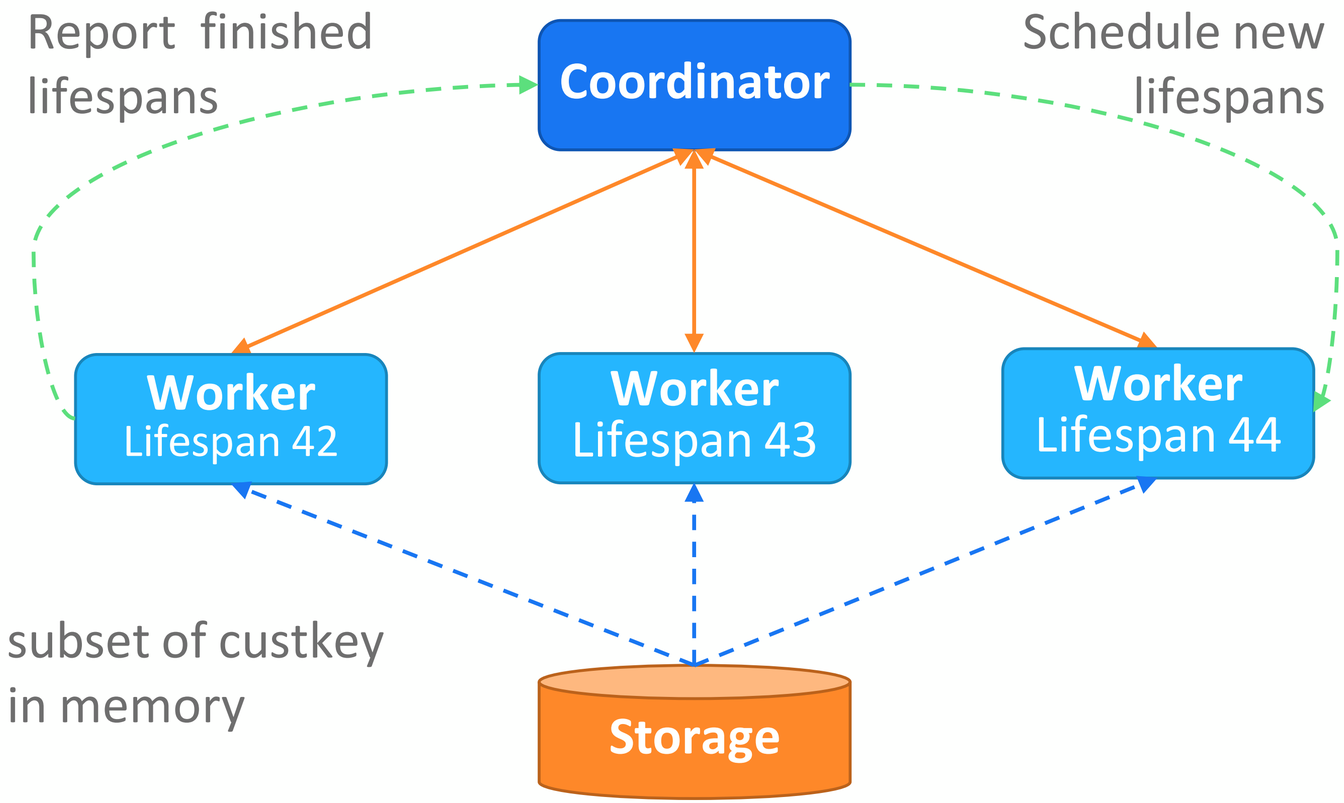
Lifespan Recoverability

Recap: Grouped Execution

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```

What if tables are bucketed on custkey?

- Flexible schedule
- “lifespan”

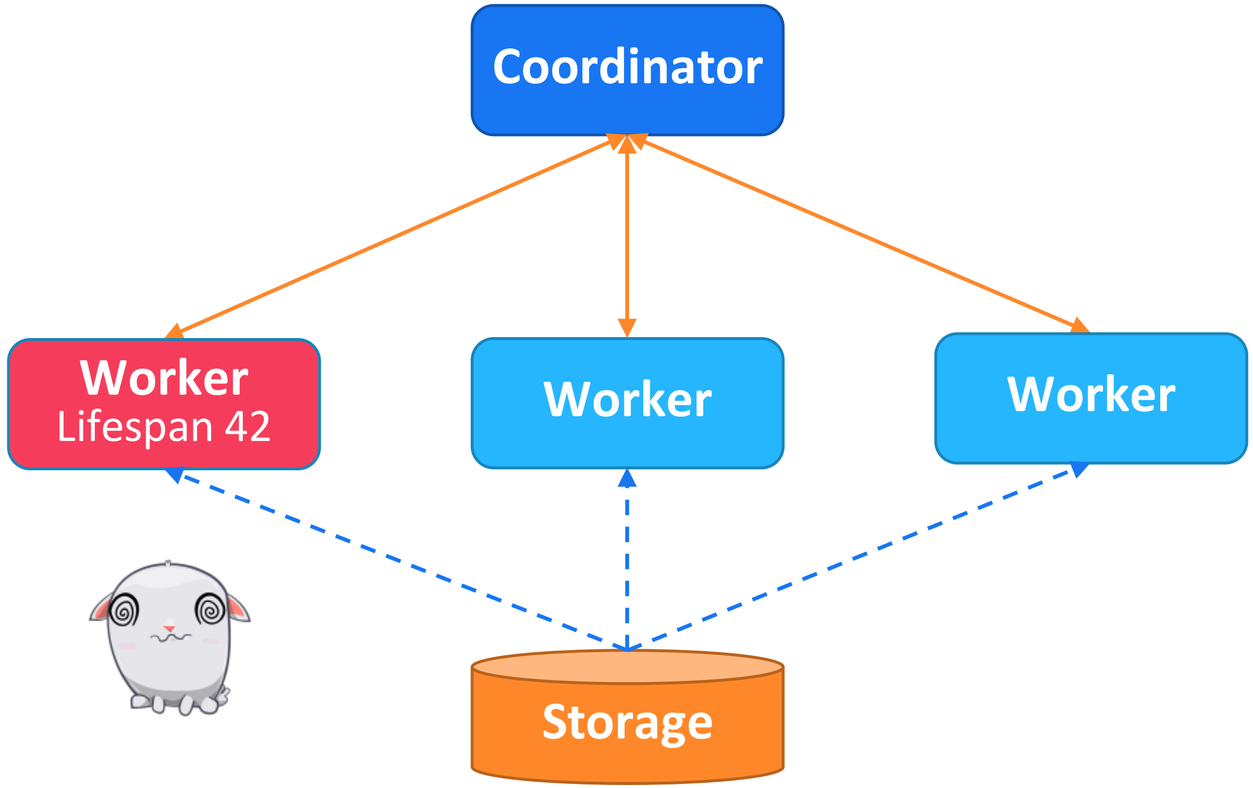


Worker Failure

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```

What if tables are bucketed on custkey?

- Flexible schedule
- “lifespan”

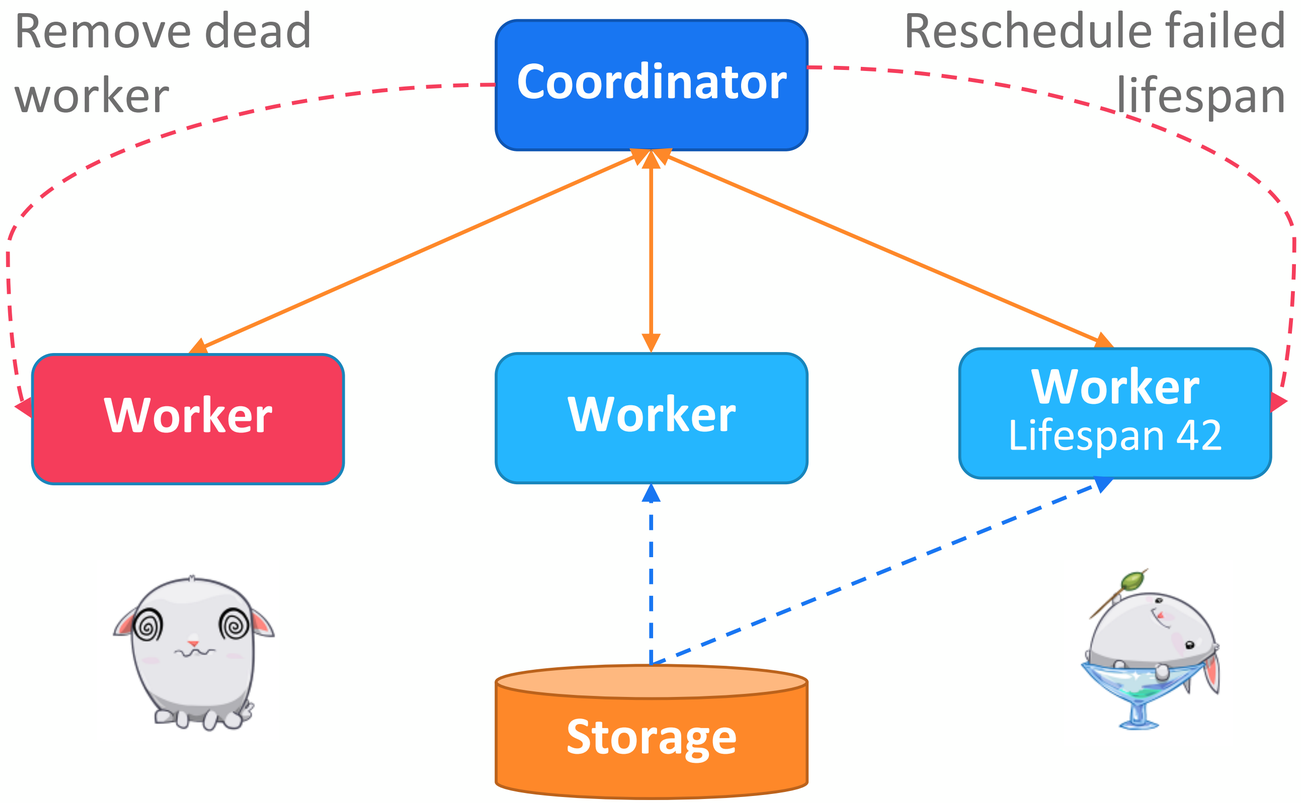


Lifespan Recoverability

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```

What if tables are bucketed on custkey?

- Flexible schedule
  - … and recoverability!
- “lifespan”



Exchange Materialization

Non-bucketed Table

Scalability Challenge

- Memory Intensive Queries
- Long-Running Queries

Grouped Execution

- Address the challenge when table is (properly) bucketed

What about Non-bucketed Table?

Non-bucketed Table

Scalability Challenge

- Memory Intensive Queries
- Long-Running Queries

Grouped Execution

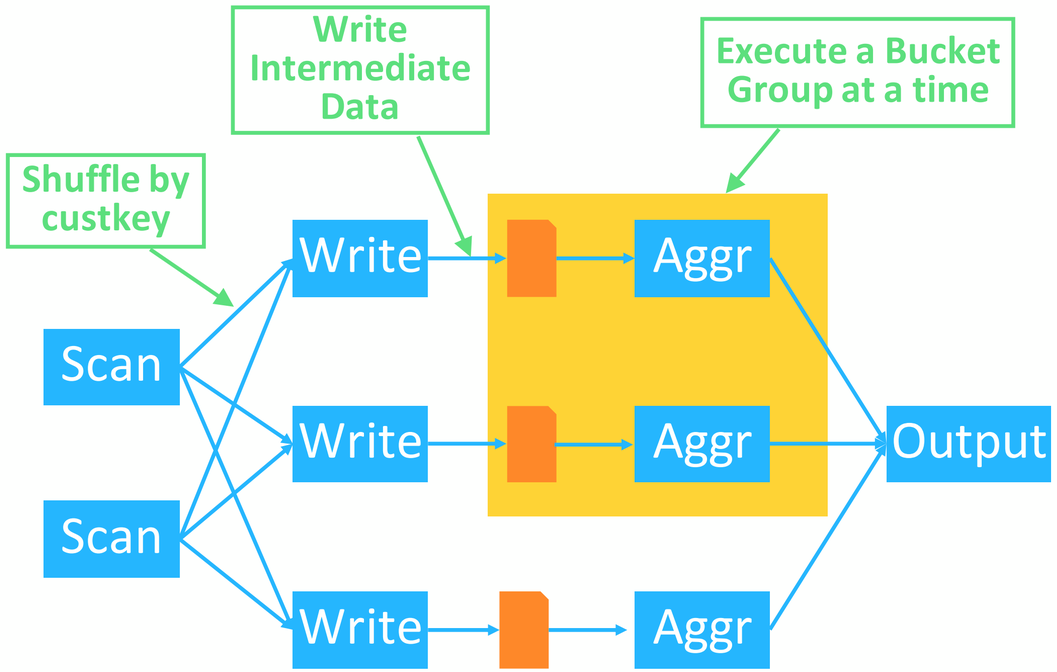
- Address the challenge when table is (properly) bucketed

What about Non-bucketed Table?

- Write temporary bucketed table as intermediate result

Exchange Materialization

```
SELECT custkey, SUM(totalprice)
FROM orders
GROUP BY custkey
```



Status and Future Work

Status

Status

- In Production!
- Initial customer onboard

Ongoing Work

- Stage Recoverability
  - Exchange materialization adds checkpoint between stages
  - See #13438 for details

- Native Shuffle Format

Future Work

Improved Batch Support

Coordinator Scalability

Resource Management

Disaggregated Shuffle Service

- Cosco, Crail, Google Cloud Dataflow Shuffle, …
- Shuffle is no longer “all or nothing”