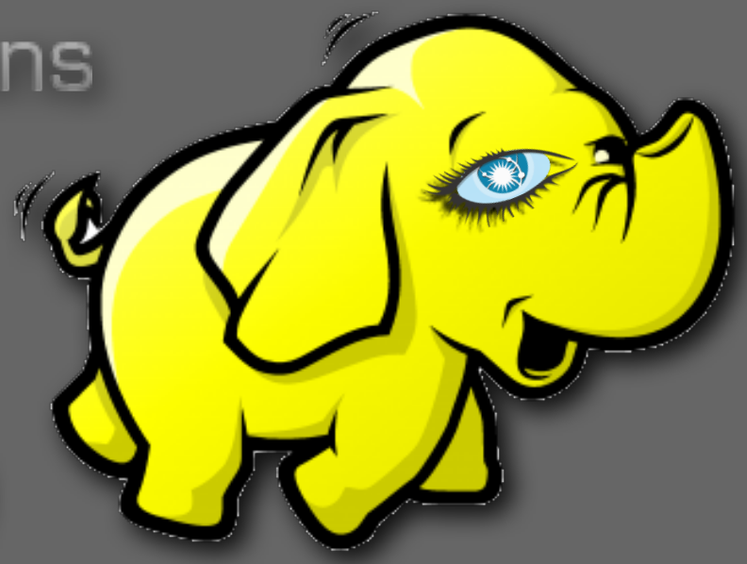
Real-Time Analytics with Cassandra and Hadoop

Patricia Gorla

opensource connections



Download code: bit.ly/1aB8Jy8 (12KB)

About Me

- Solr
- Cassandra
- Datastax MVP

Download code: bit.ly/1aB8Jy8 (12KB)

Outline

- Introduction to Cassandra *+ 2 labs*

  15m Break ~ 14:30

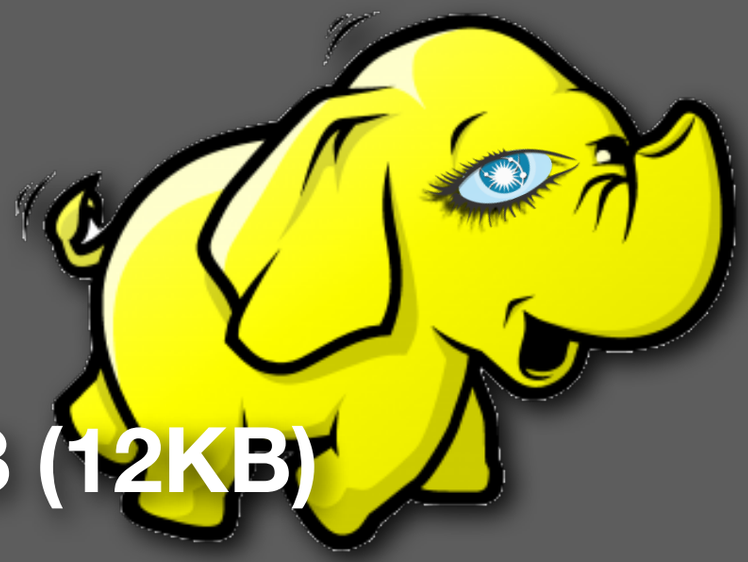
- Analytics *+ 1 labs*

  15m Break ~ 16:30

- Extra Credit

Download code: bit.ly/1aB8Jy8 (12KB)

Introduction



Download code: bit.ly/1aB8Jy8 (12KB)

Getting Started

Architecture

Data Modeling

Download code: bit.ly/1aB8Jy8 (12KB)

History

- Powered inbox search at Facebook
- Open-sourced in 2008

Why Cassandra?

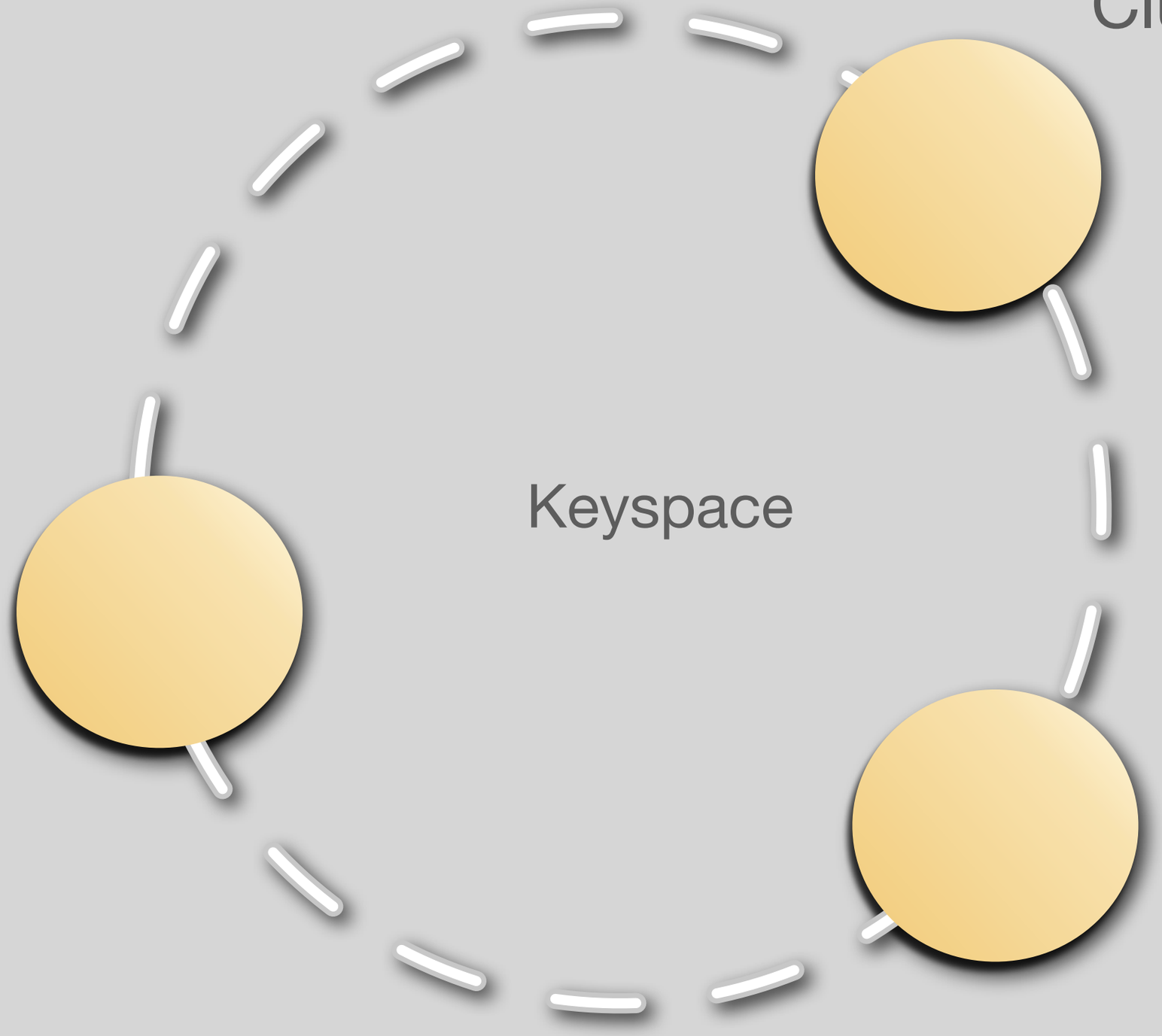
- Linear scalability
- Availability
- Set it and forget it

...

Many companies use Cassandra.

What is Cassandra?

- Dynamo distributed cluster (no vector clocks)
- Bigtable data model
- No SPOF
- Tuneably consistent



Architecture

Keyspace

Column Family 1

Column Family 2

Keyspac

Column Family 1

Column Family 2

row1: {col1:val1,time,TTL; … }

Lab

`introduction/1-getting-started.md`

Download code: bit.ly/1aB8Jy8 (12KB)

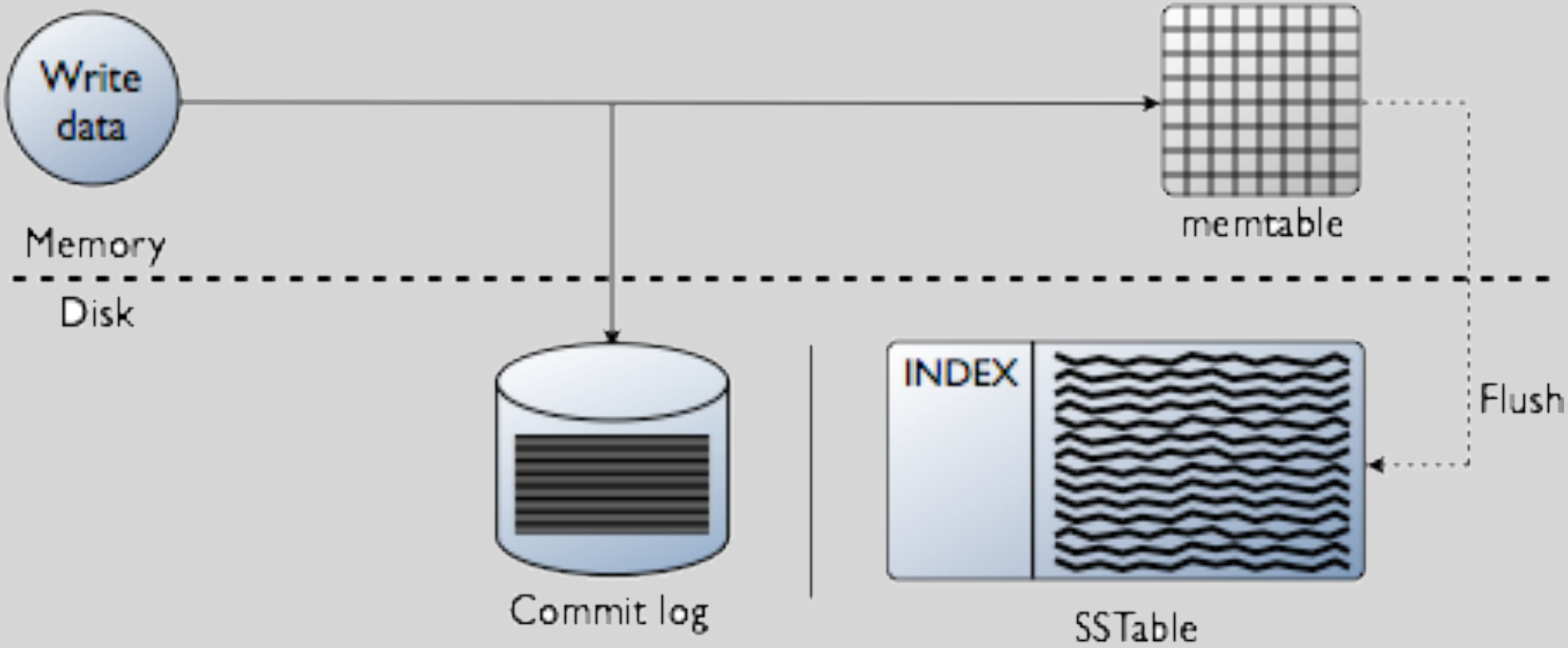
Getting Started

Architecture

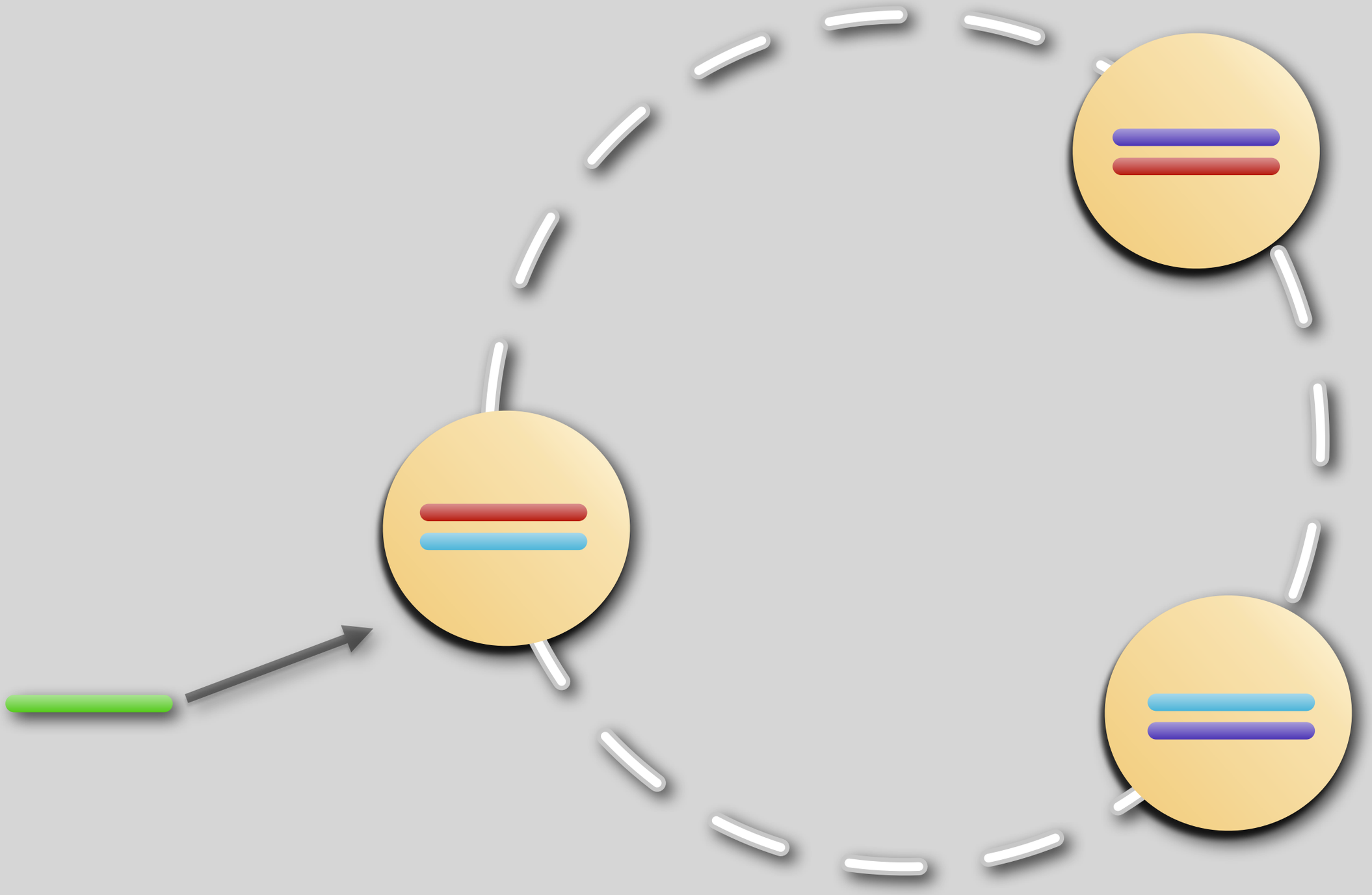
Data Modeling

Writes

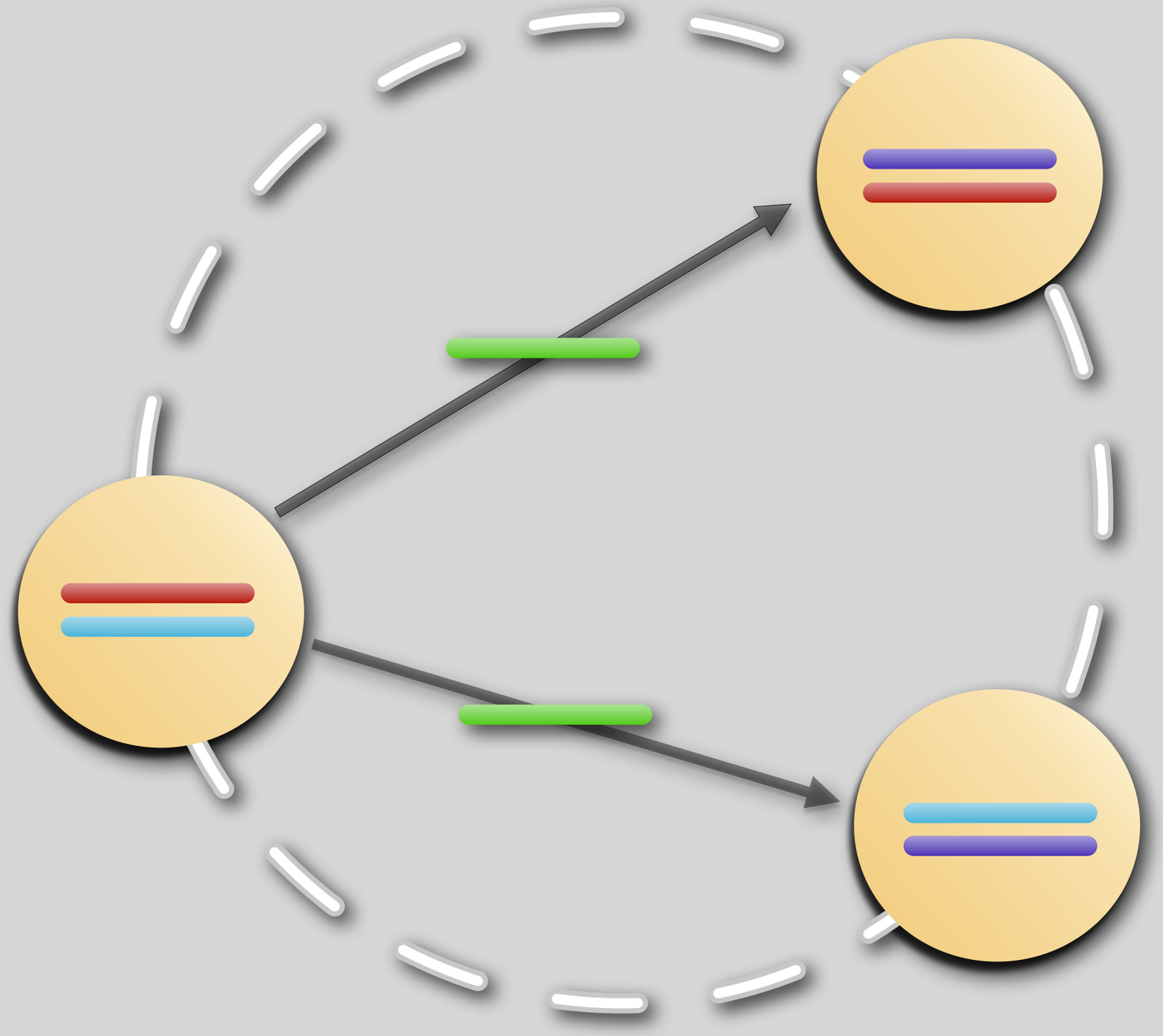
Commit Log -> Memtable -> SSTables

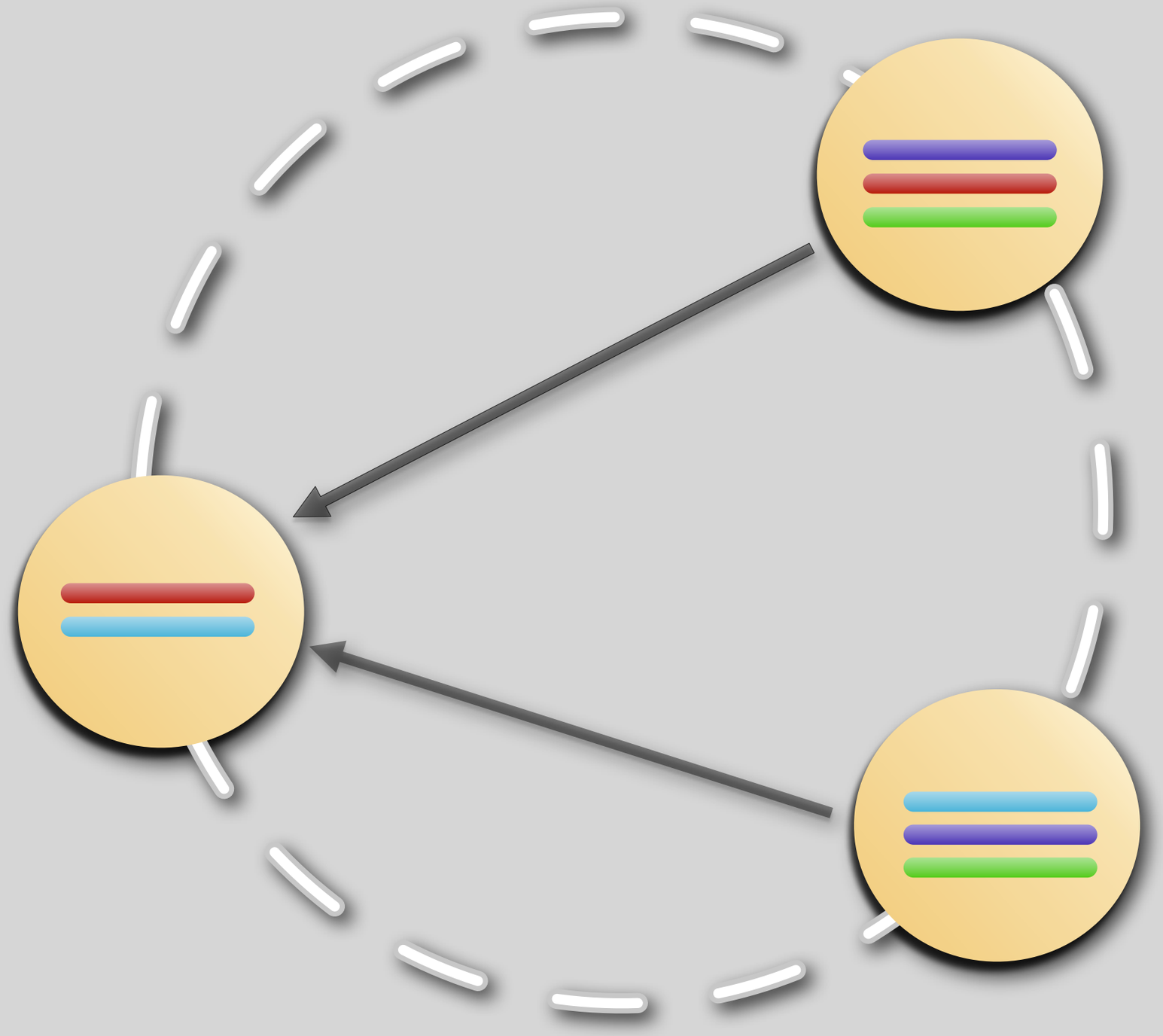


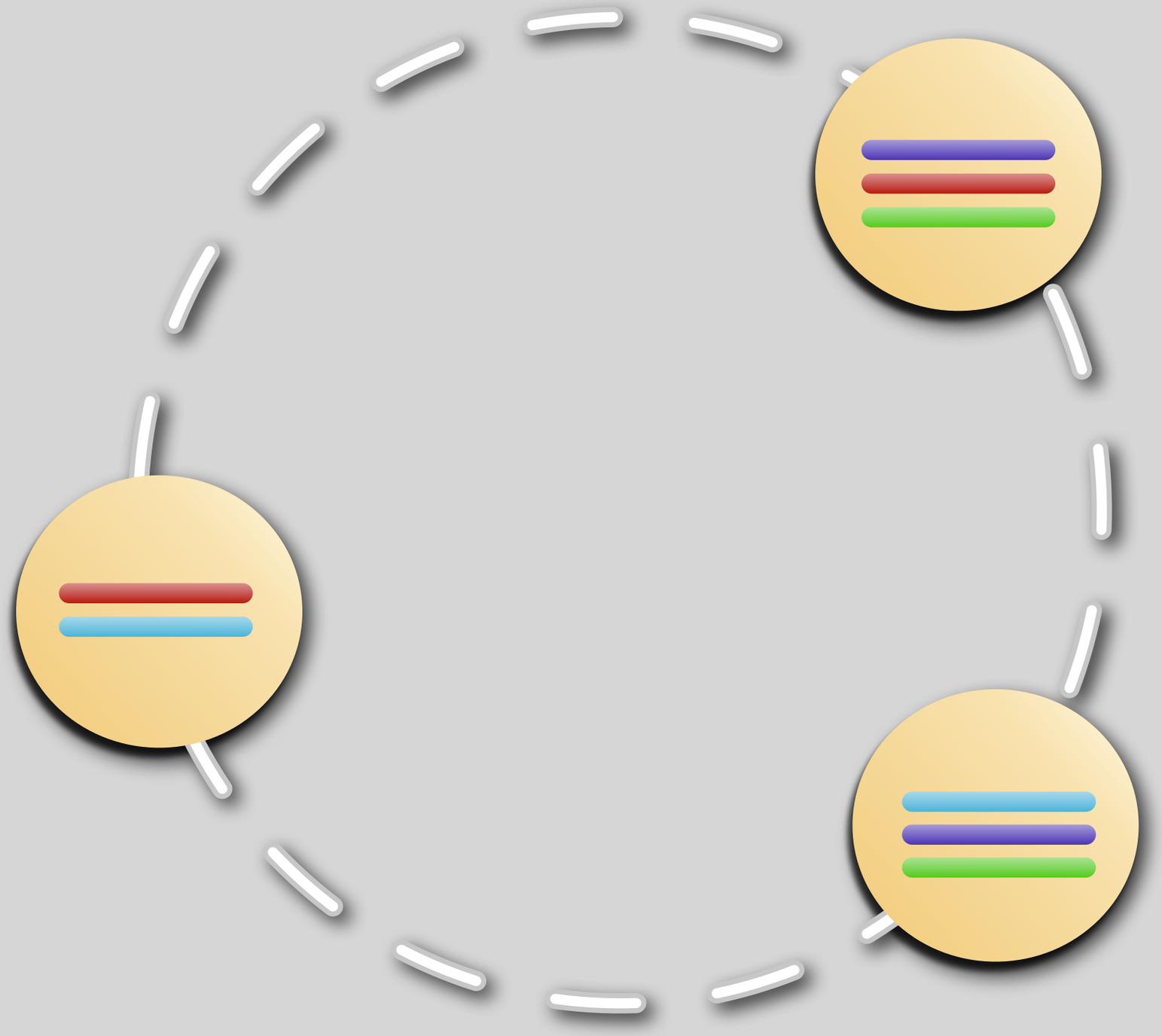
Source: datastax.com



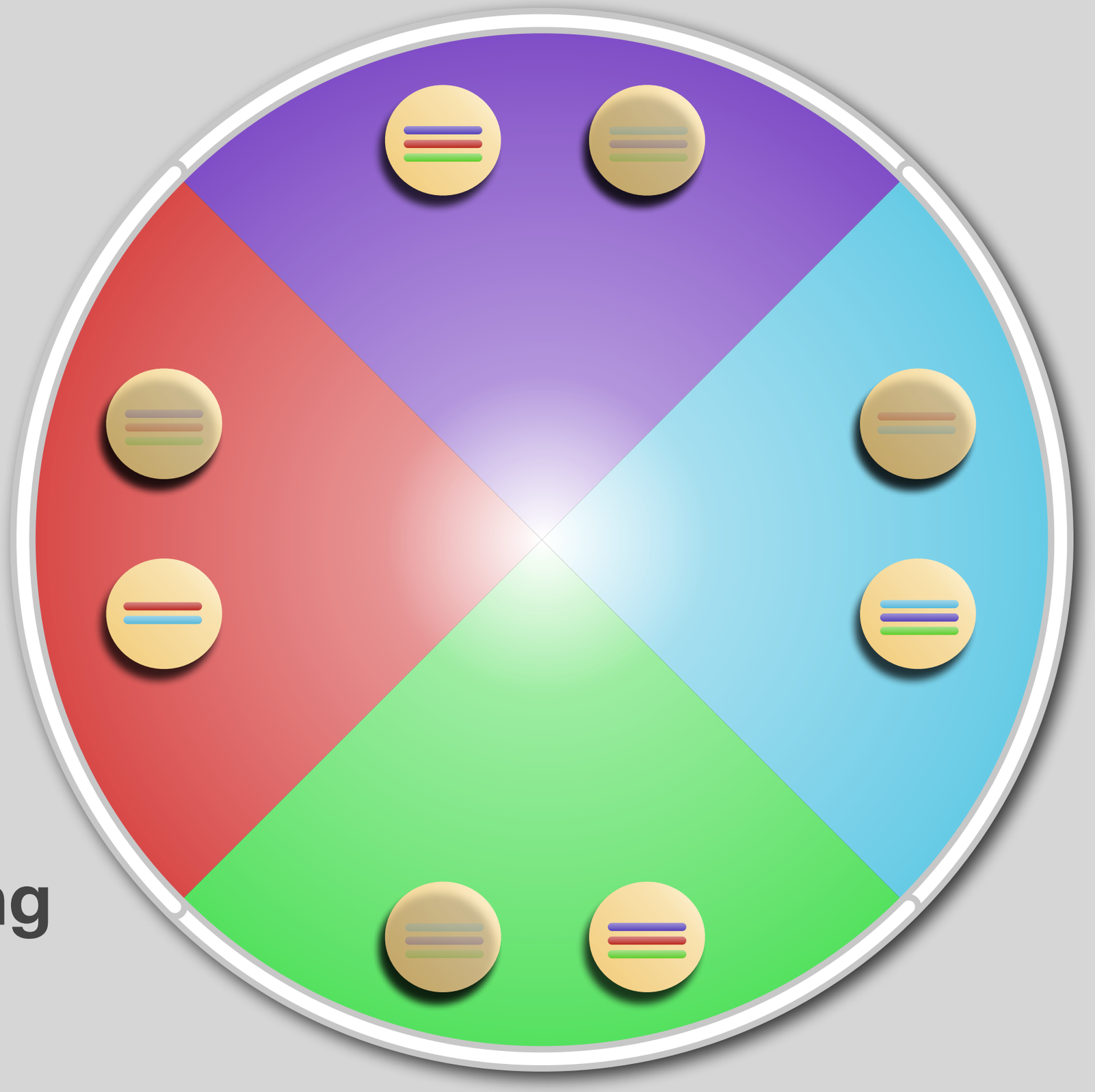
Incoming write to cluster.



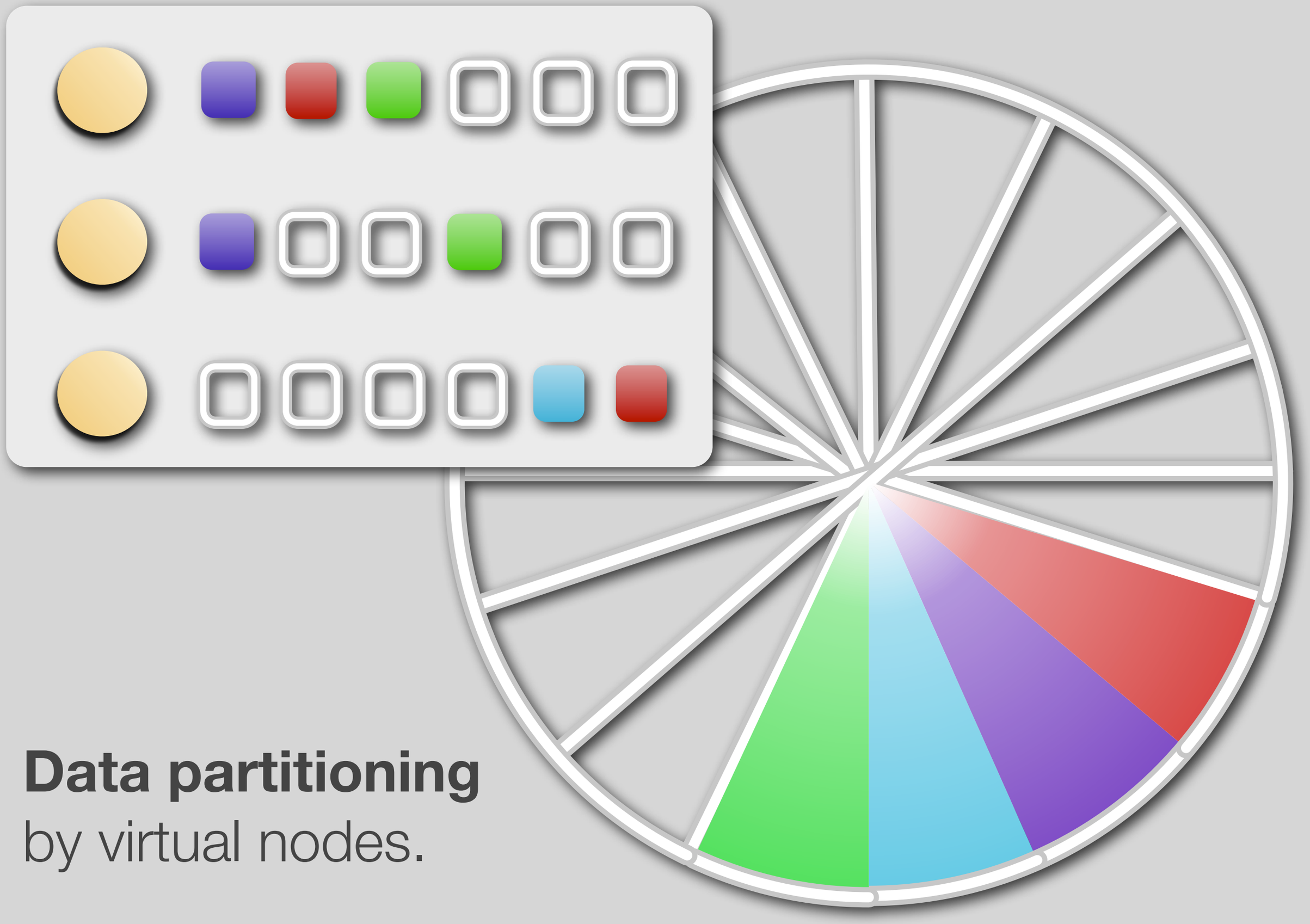




Data replicated to replicants.

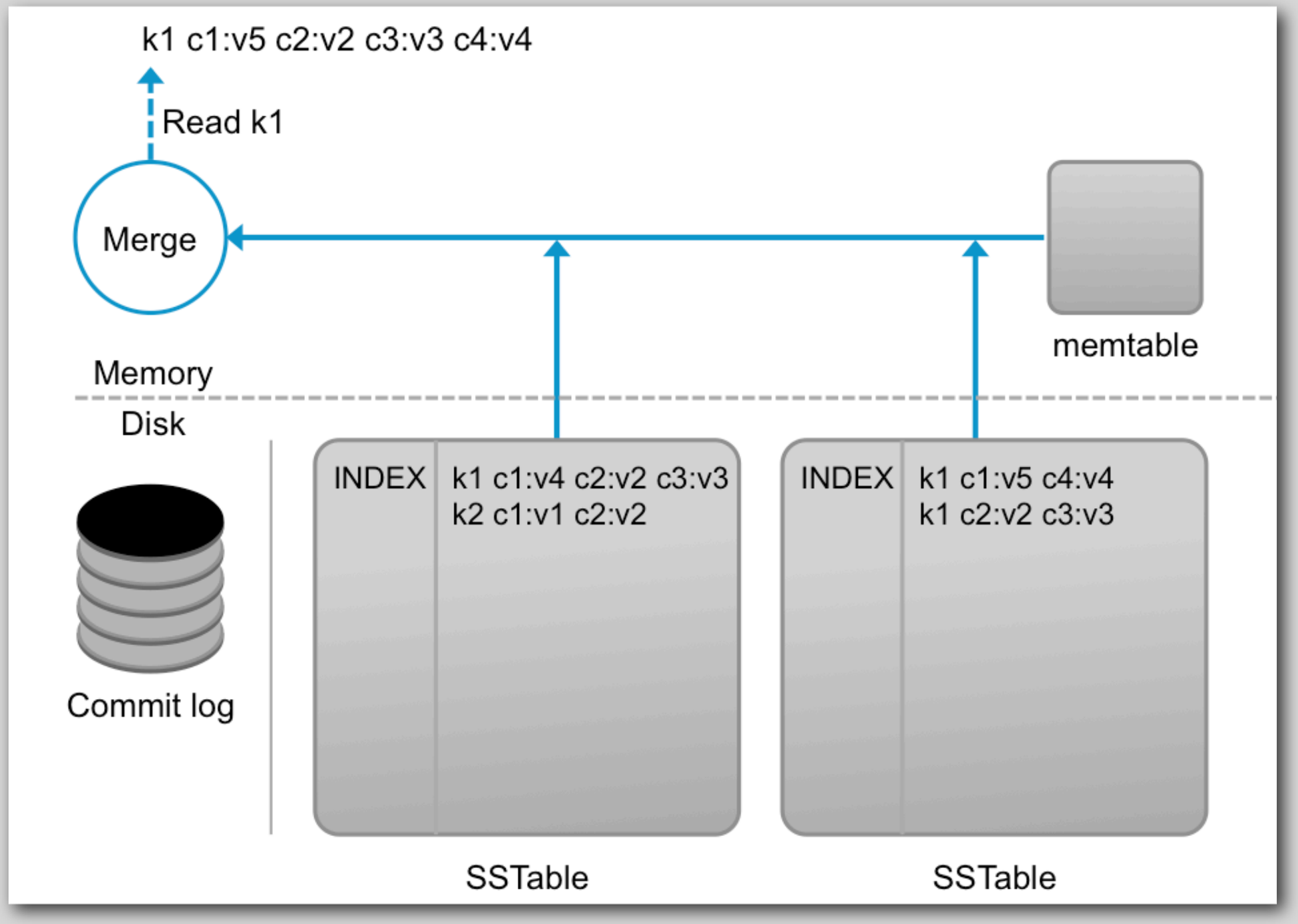


Data partitioning
by token ranges.



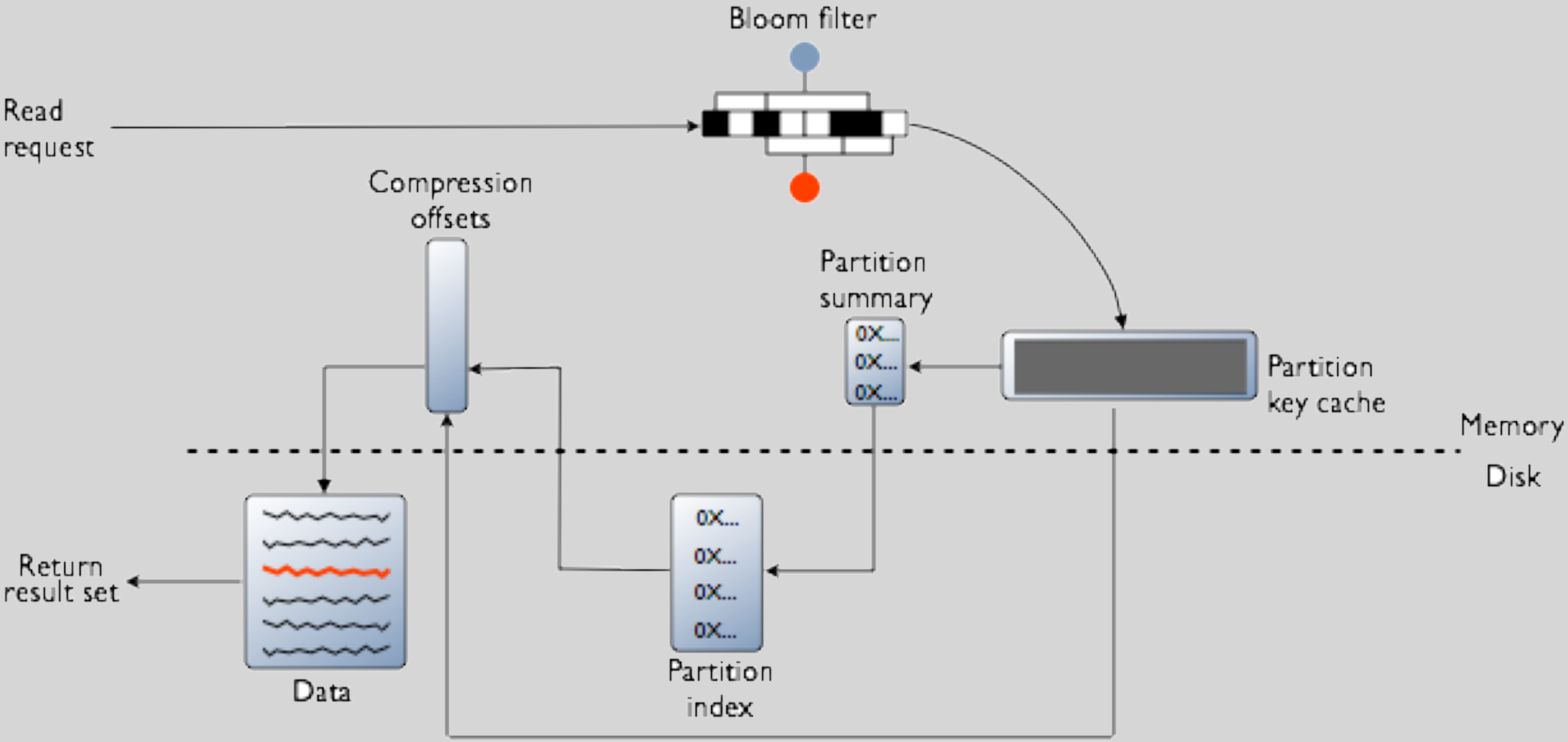
Data partitioning
by virtual nodes.

Reads

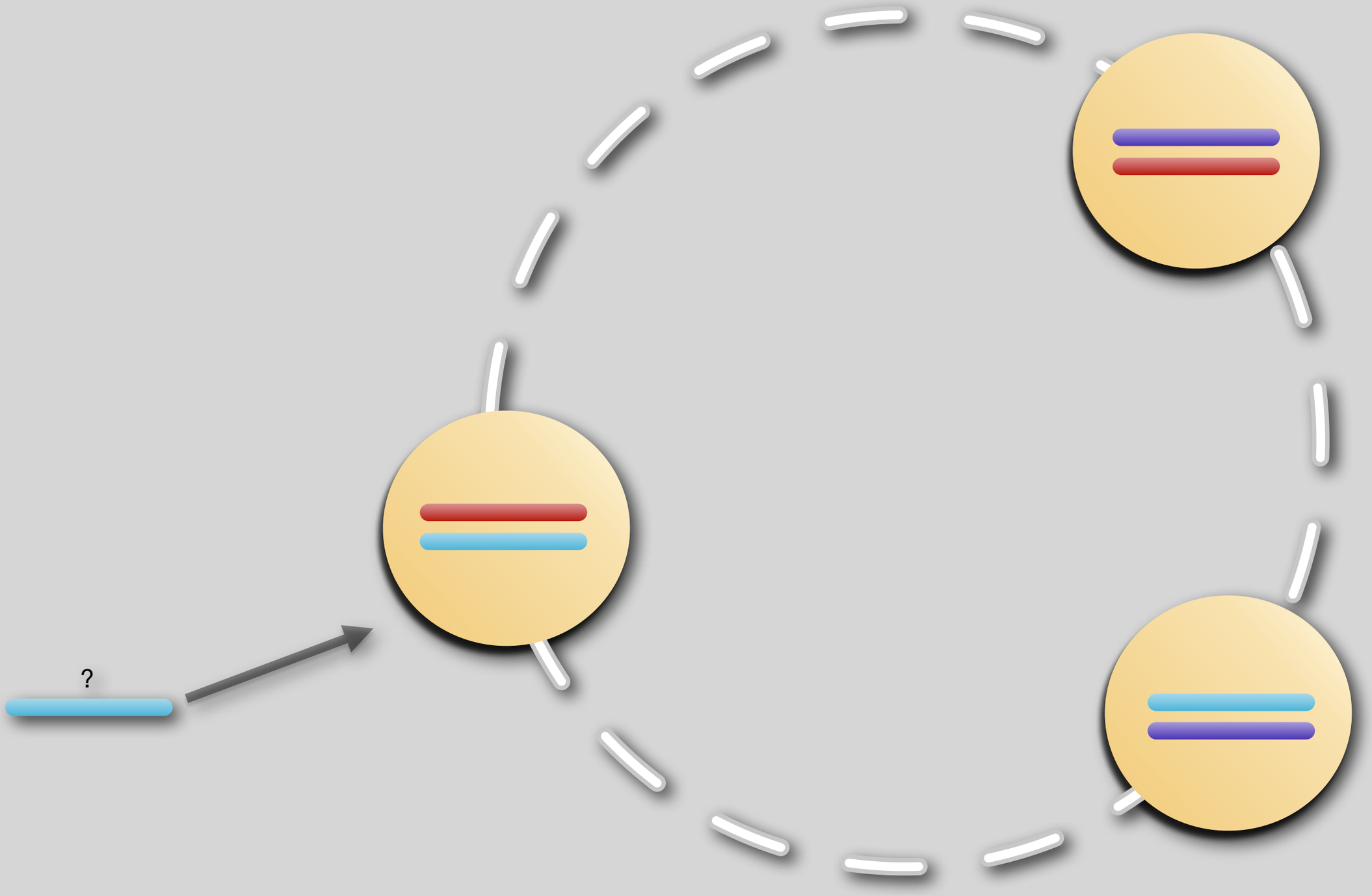


High-level overview of reads.

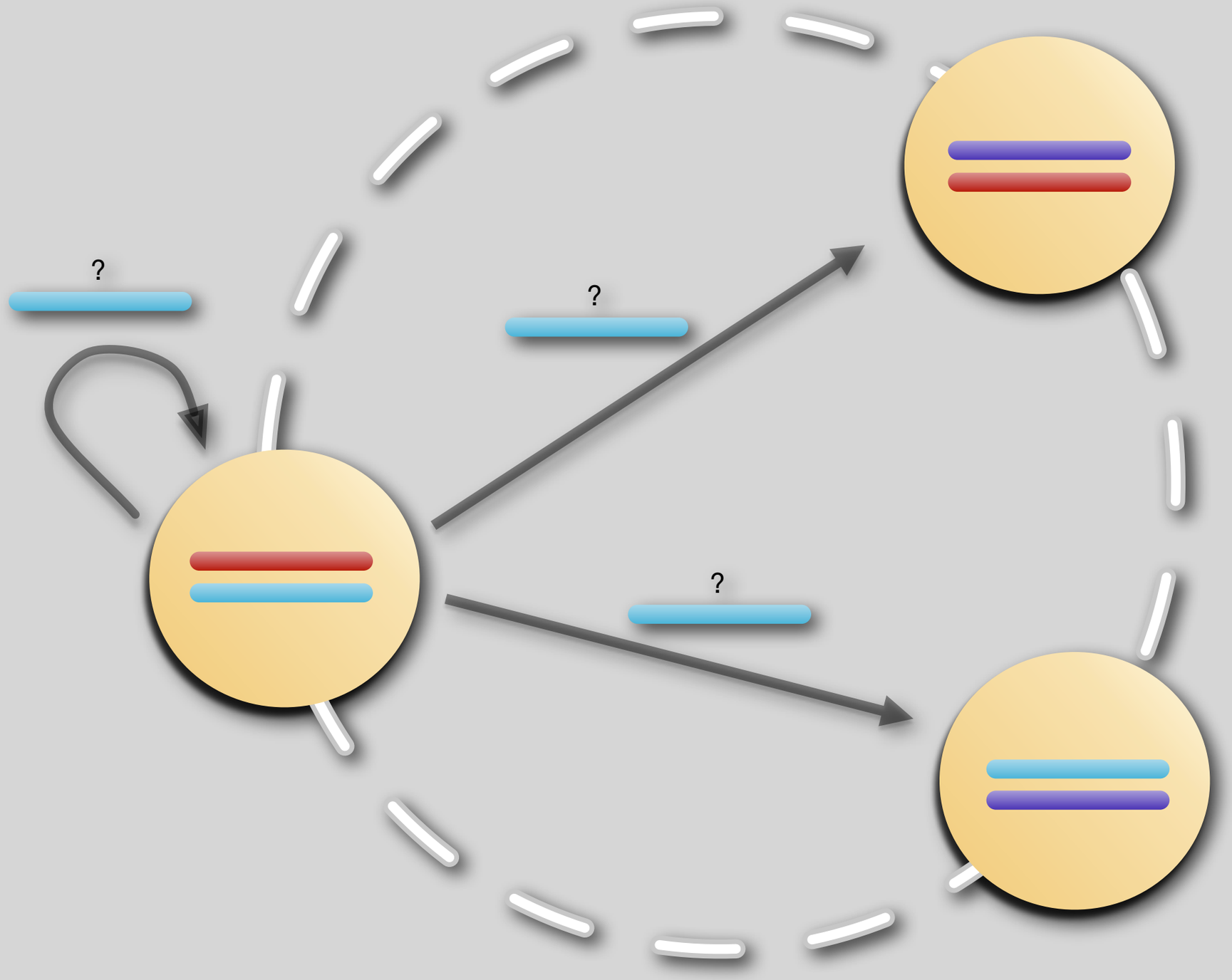
Source: fusionio.com



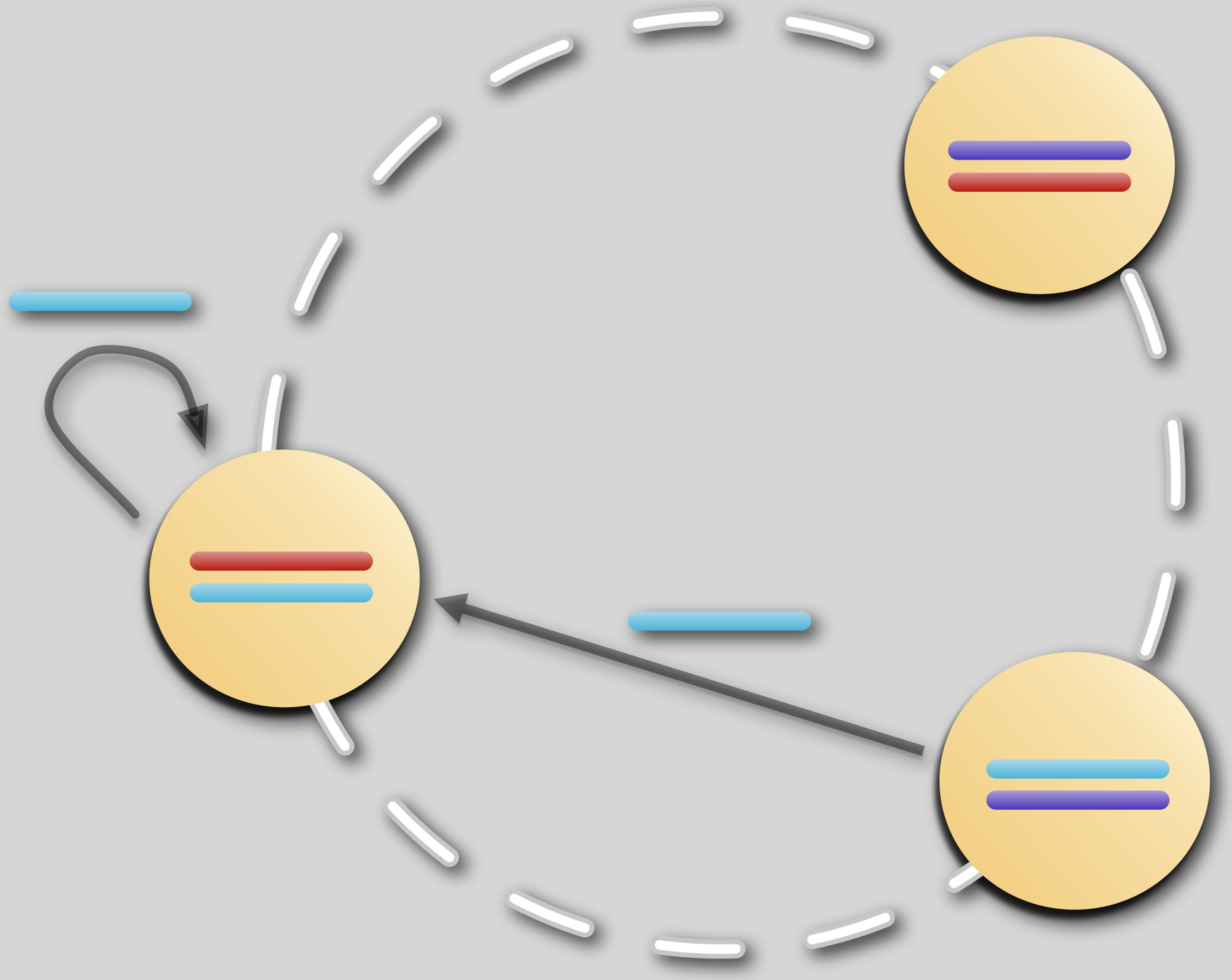
Source: datastax.com



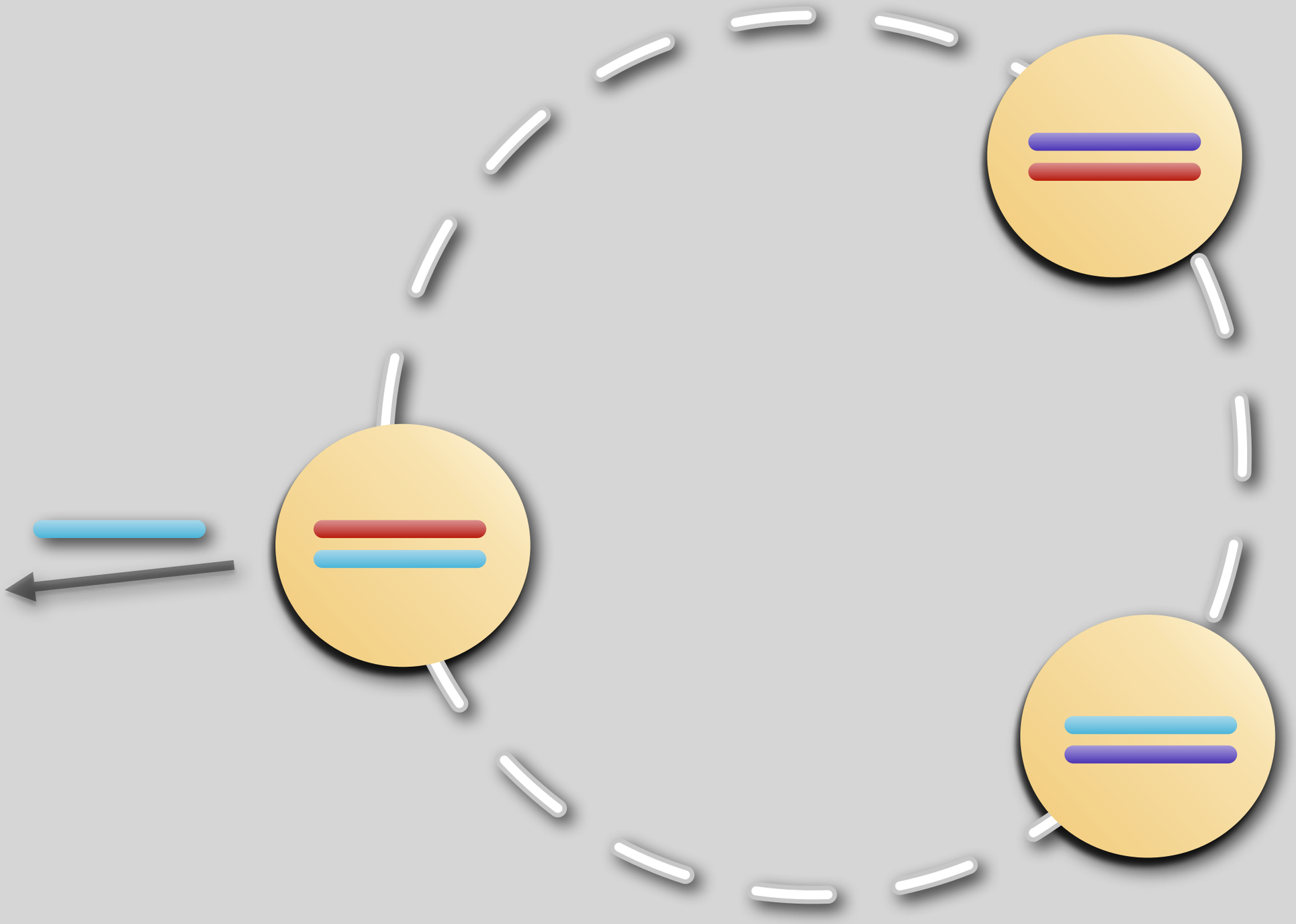
Reading from cluster.



Reading from cluster.

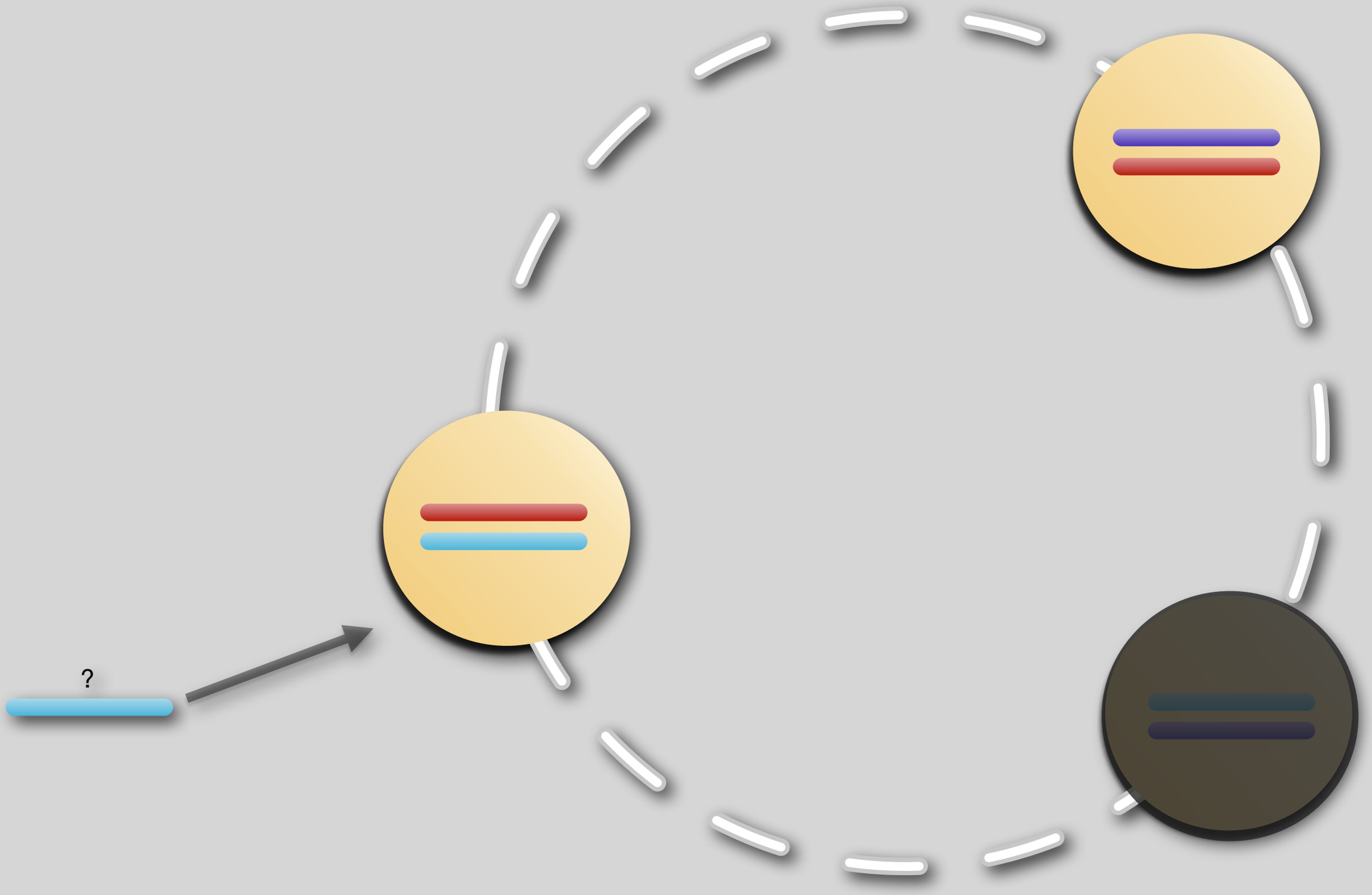


Reading from cluster.

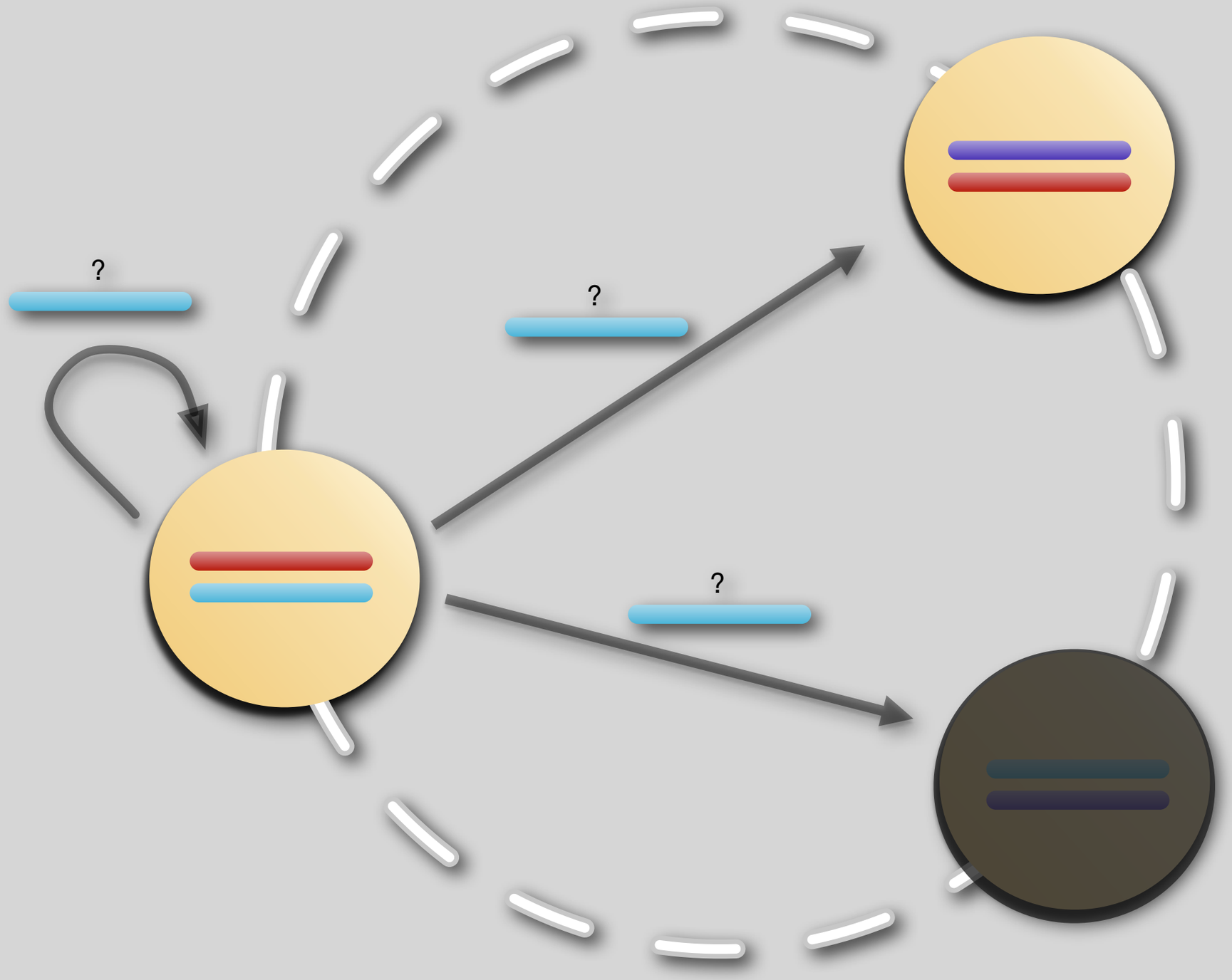


Reading from cluster.

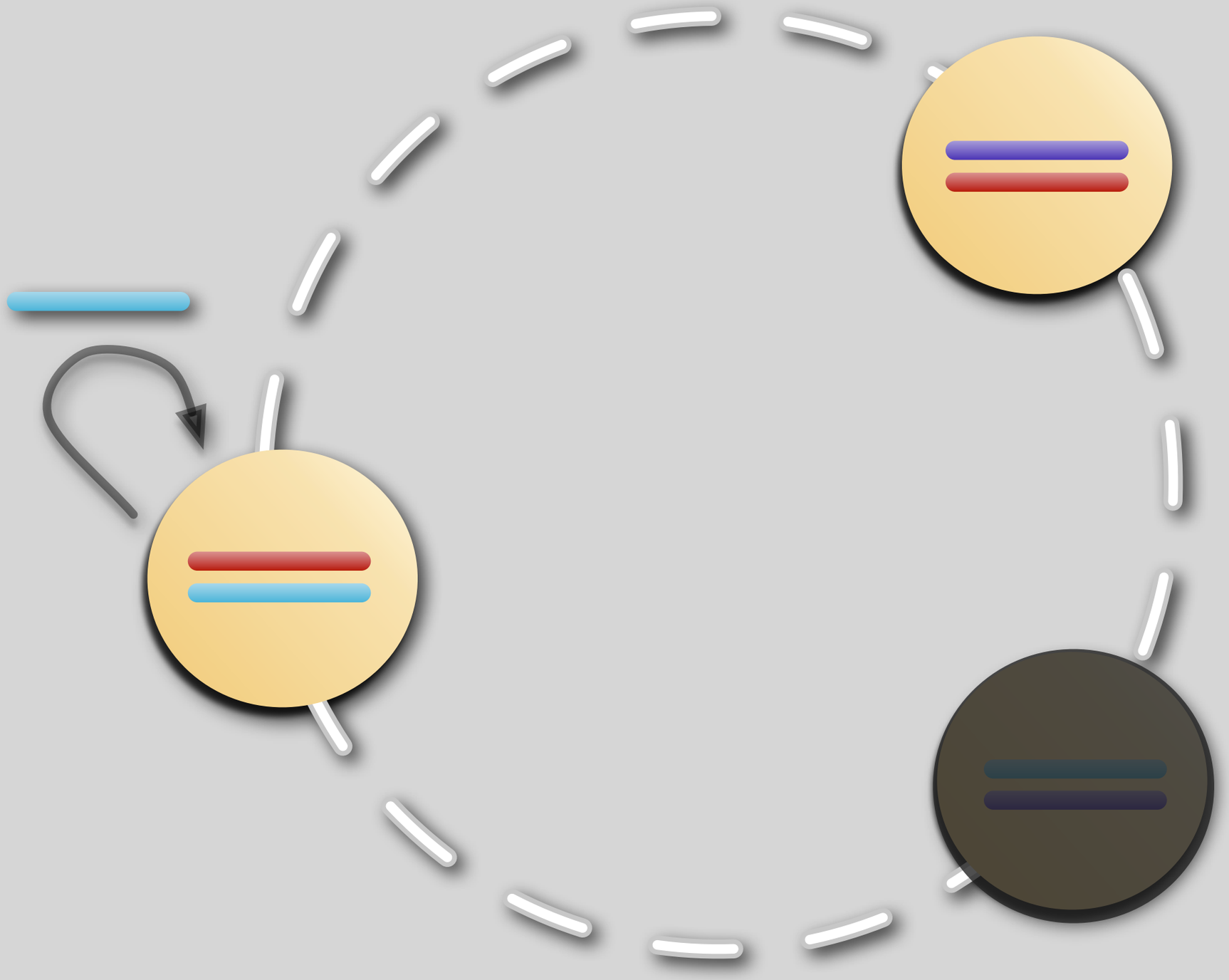
Fault tolerance



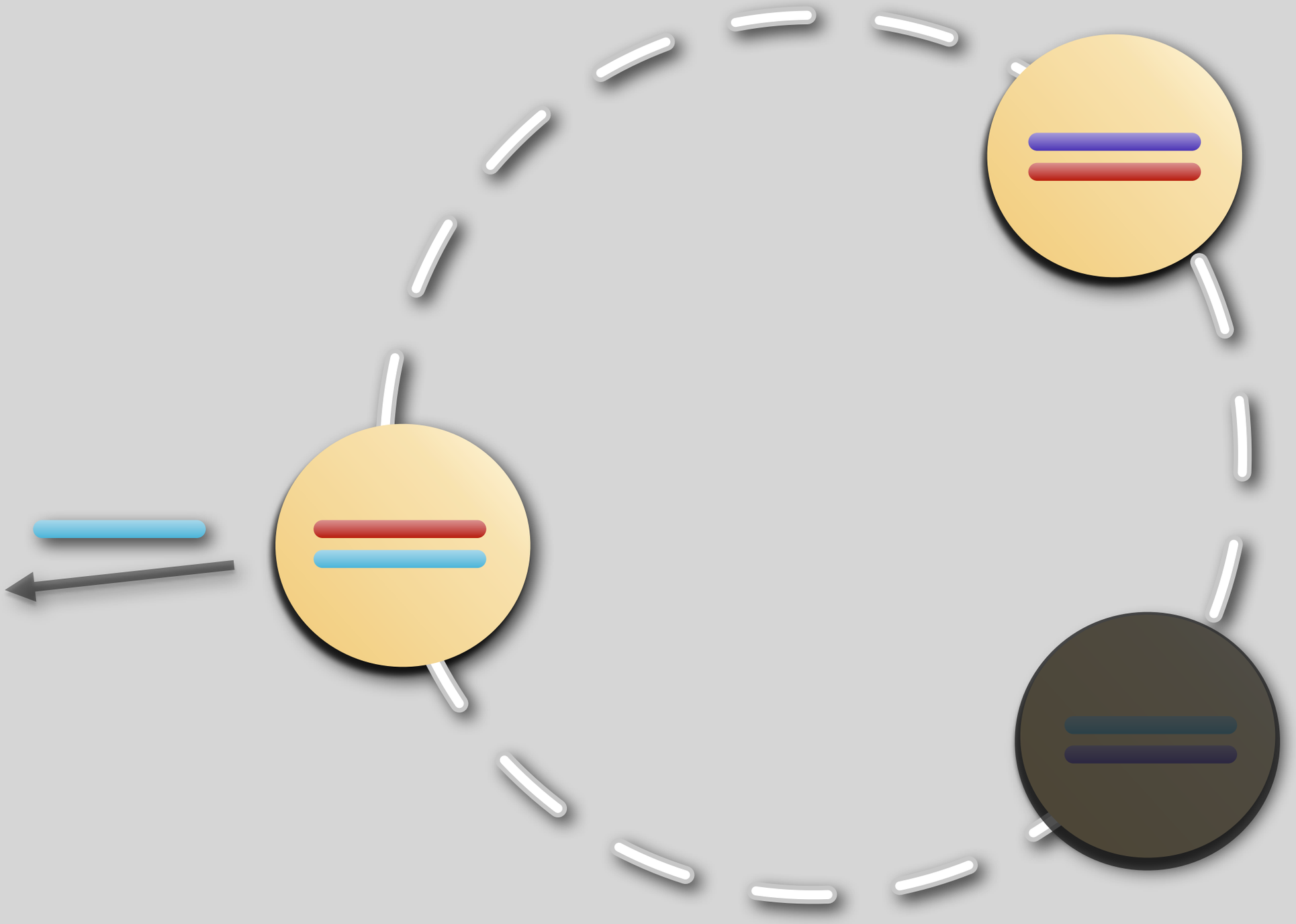
Reading from cluster.



Reading from cluster.



Reading from cluster.



Reading from cluster.

Deletes

- Distributed deletes are tricky
- Tombstones may not be propagated
- Don’t rely on a delete-heavy system

Getting Started

Architecture

Data Modeling

Protocols

Thrift

- Thrift, CQL
- Synchronous

Binary

- CQL
- Asynchronous

Why CQL?

- Familiar syntax
- Flexible data model over Cassandra

CQL: Creating a Keyspace

```
create KEYSPACE “Patisserie”
  with replication =
  {‘class’: ‘SimpleStrategy’,
   ‘replication_factor’: 1 } ;

use “Patisserie”;
```

CQL: Creating a Column Family

```
create TABLE “customers”
  (customer text,
   age int,
   PRIMARY KEY (customer) ) ;
```

CQL Schema

customer	age
Yves Laurent	77
Coco Chanel	130
Pierre Cardin	

CQL: Creating a Column Family

```
create TABLE “customers”
  (customer text,
   age int,
   PRIMARY KEY (customer) ) ;
```

Physical Representation

```
”Yves Laurent”:  {“age”:77}
“Coco Chanel”:   {“age”:130}
“Pierre Cardin”: {}
```

CQL: Composite Columns

```
create TABLE “customer_purchases”
    (customer text,
    day text,
    item text,
    PRIMARY KEY (customer,day) ) ;
```

customer	day	item
ylaurent	M	rivoli
ylaurent	T	mille feuille
cchanel	M	pain au chocolat
pcardin	W	mille feuille
pcardin	F	croissant

CQL: Composite Columns

```
create TABLE “customer_purchases”
    (customer text,
    day text,
    item text,
    PRIMARY KEY (customer,day) ) ;
```

```
”ylaurent”: { “M:item”: “rivoli”, “T:item”: “mille feuille” }

“cchanel”:  { “M:item”: “pain au chocolat” }

“pcardin”:  { “W:item”: “mille feuille”, “F:item”: croissant }
```

CQL: Composite Primary Keys

```
create TABLE “daily_sales_by_item”
   (day text,
    customer text,
    hour timestamp,
item text,
PRIMARY KEY ((day,customer), hour) ) ;
```

day	customer	hour	item
M	cchanel	13	rivoli
M	cchanel	15	mille feuille
M	ylaurent	4	rivoli
T	cchanel	17	mille feuille
W	pcardin	20	croissant

CQL: Composite Primary Keys

```
create TABLE “daily_sales_by_item”
   (day text,
    customer text,
    hour timestamp,
          item text,
          PRIMARY KEY ((day,customer), hour) ) ;
```

```
”M:cchanel”:  { “13:item”: “rivoli”, “15:item”: “mille feuille” }

“M:ylaurent”: { “4:item”: “rivoli” }

“T:cchanel”:  { “17:item”: “mille feuille" }

“W:pcardin”:  { “20”item”: “croissant” }
```

CQL: Collections

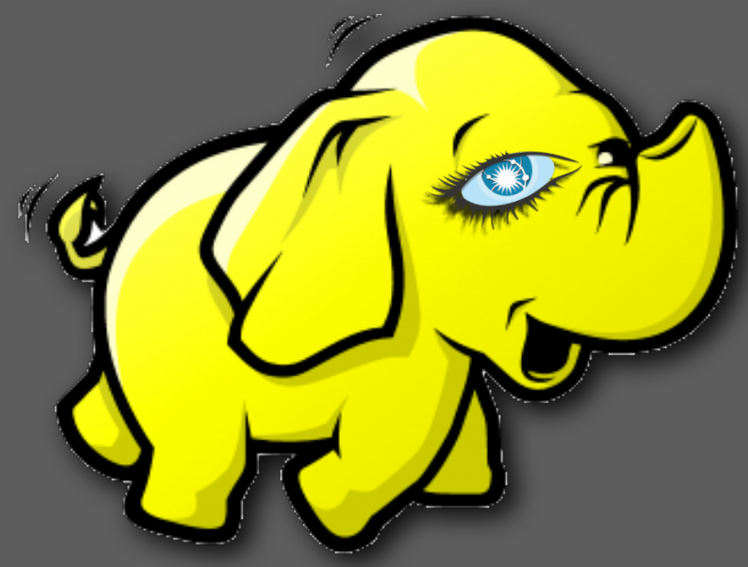
```
create TABLE “customer_purchases”
    (customer text,
    day text,
    item list<text>,
    PRIMARY KEY (customer,day) ) ;
```

customer	day	item
ylaurent	M	[‘rivoli’, ‘rivoli’, ‘javanais’]
cchanel	M	[‘pain au chocolat’]
pcardin	W	[‘mille feuille’, ‘croissant’]
pcardin	F	[‘croissant’]

Data Modeling Lab

`introduction/2-data-modeling.md`

Analytics



Cassandra and Analytics

Adapting the Data Model

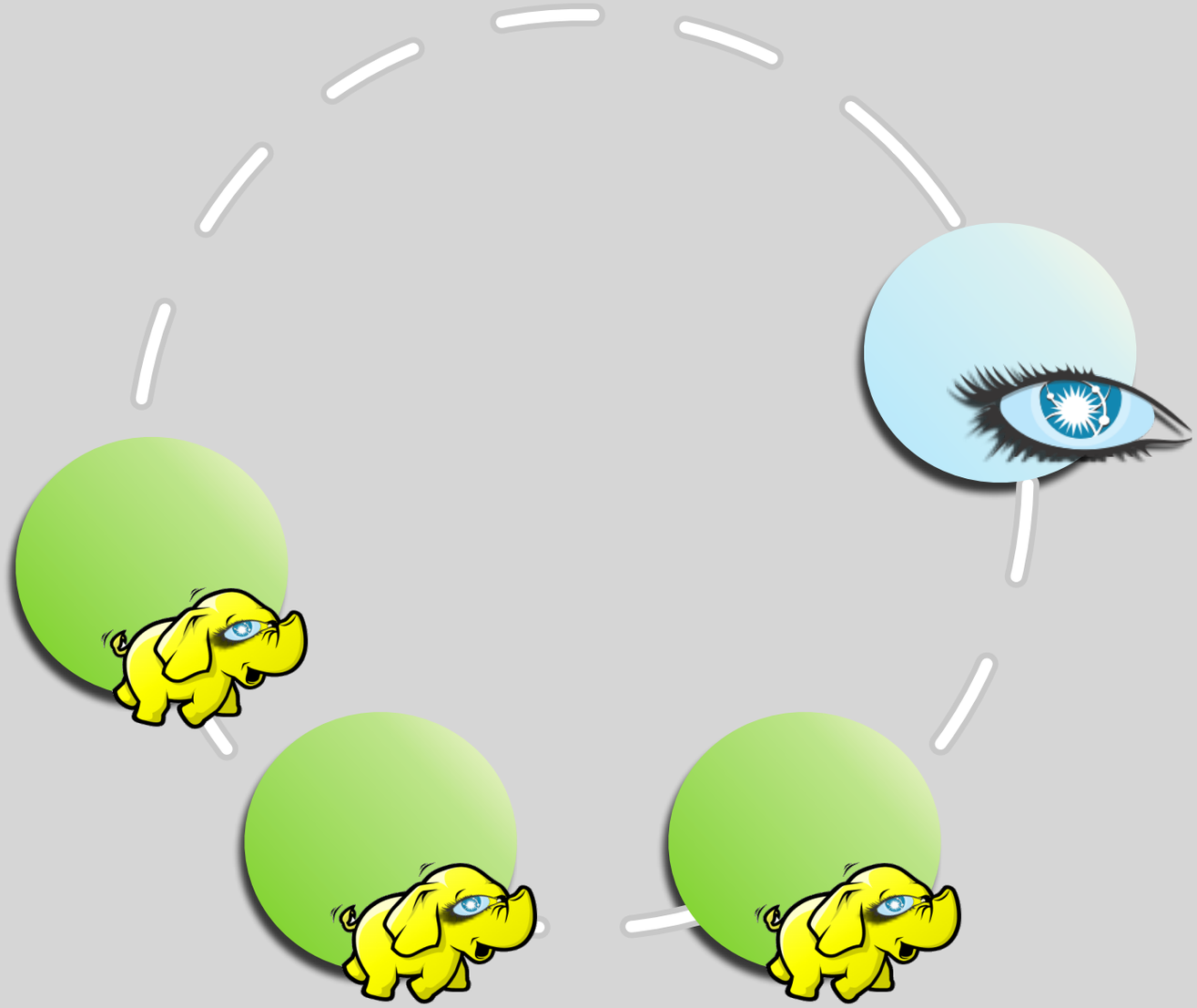
MapReduce Paradigms

An Unlikely Union

- Batch processing analytics and real-time data store
- MapReduce, Hive, Pig, Sqoop, Mahout

Why Cassandra and Hadoop?

- Unified workload
- Availability
- Simpler deployment

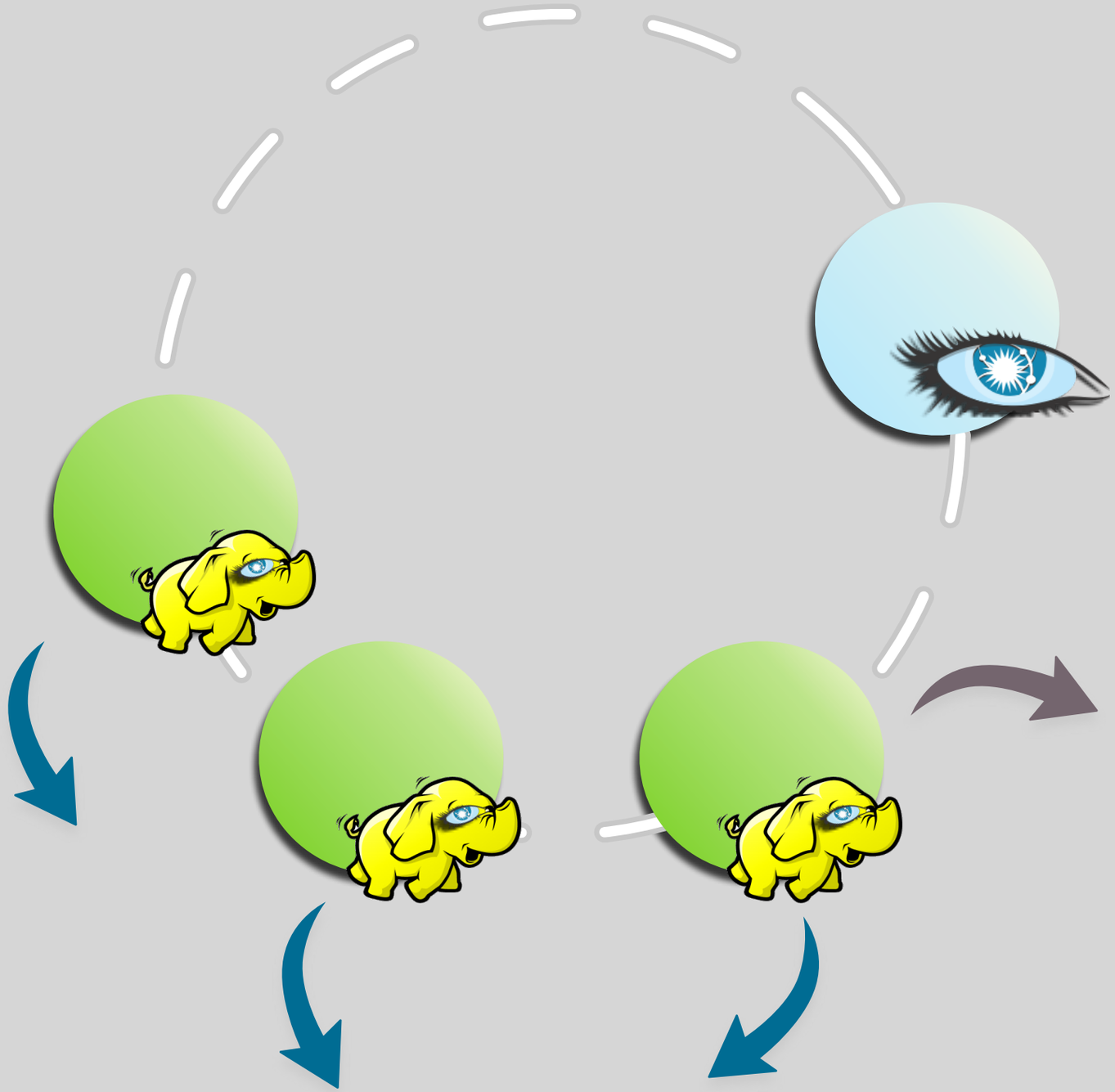


Data Locality

Data Locality

Data Locality

Datastax Enterprise



Datastax Enterprise

MapReduce

CFS

Writing in / out is passed through the CassandraFS layer

Starting Analytics Node

```
$ bin/dse cassandra -t -j

# Starts task tracker and job tracker on
# node
```

Hello, Wordcount

```
$ bin/dse hadoop fs -put wikipedia /

$ bin/dse hadoop jar wordcount.jar /wikipedia
  wc-output
```

Cassandra and Hadoop

Adapting the Data Model

MapReduce Paradigms

Hive

- SQL-like MapReduce abstraction
- Data types
- Efficient JOINs, GROUP BY

Cassandra and Hive

- Hive still has to have separate tables.
- DSE stores them in a separate keyspace.
  - 1:1 mapping to Cassandra CFs
  - Schemas must match or columns will be inaccessible.

MapReduce

CFS

Hive Metastore is persisted in Cassandra layer

Hive: Creating a DB

```
hive> CREATE EXTERNAL TABLE customers (
   id string, name string, age int
)
STORED BY
   ‘o.a.h.h.cassandra.CassandraStorageHandler’
TBLPROPERTIES (
   “cassandra.ks.name” = “Oberweis”,
   “cassandra.ks.repfactor” = “2”,
   “cassandra.ks.strategy” = “o.a.c.l.SimpleStrategy”
);
```

Hive: Multiple Data Centers

```
hive> CREATE EXTERNAL TABLE customers (
   id string, name string, age int
)
STORED BY
   ‘o.a.h.h.cassandra.CassandraStorageHandler’
TBLPROPERTIES (
   “cassandra.ks.name” = “Oberweis”,
   “cassandra.ks.stratOptions” = “DC1:3, DC2:1”,
   “cassandra.ks.strategy” = “o.a.c.l.NTStrategy”
);
```

Hive

- What about composite columns?
- Must be retrieved as binary data, and then use UDF to deserialize it.

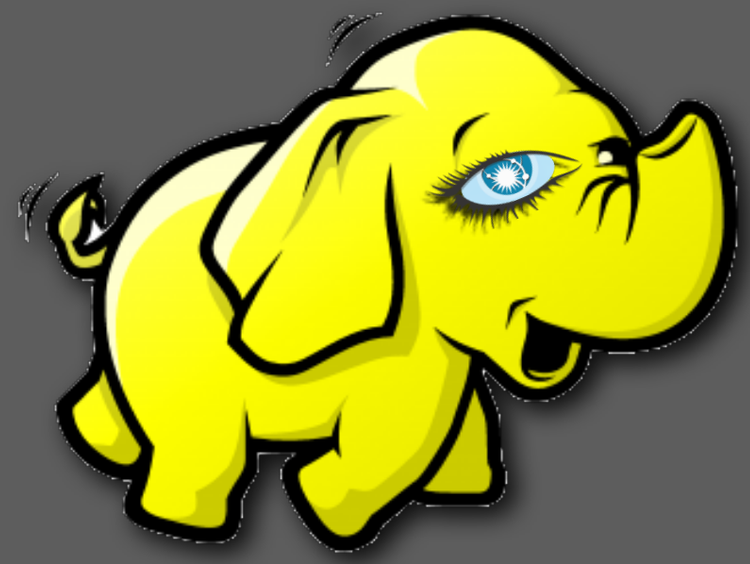
Hive: Lab

- For each person, calculate how many pastries (and of what kind) they purchased.

Hive: Multiple Data Centers

```
hive> SELECT
  b.name, a.item, sum(a.amount)
FROM Oberweis.daily_purchases a
JOIN Oberweis.person b ON (a.person = b.id)
GROUP BY b.name, a.item;
```

Extra Credit



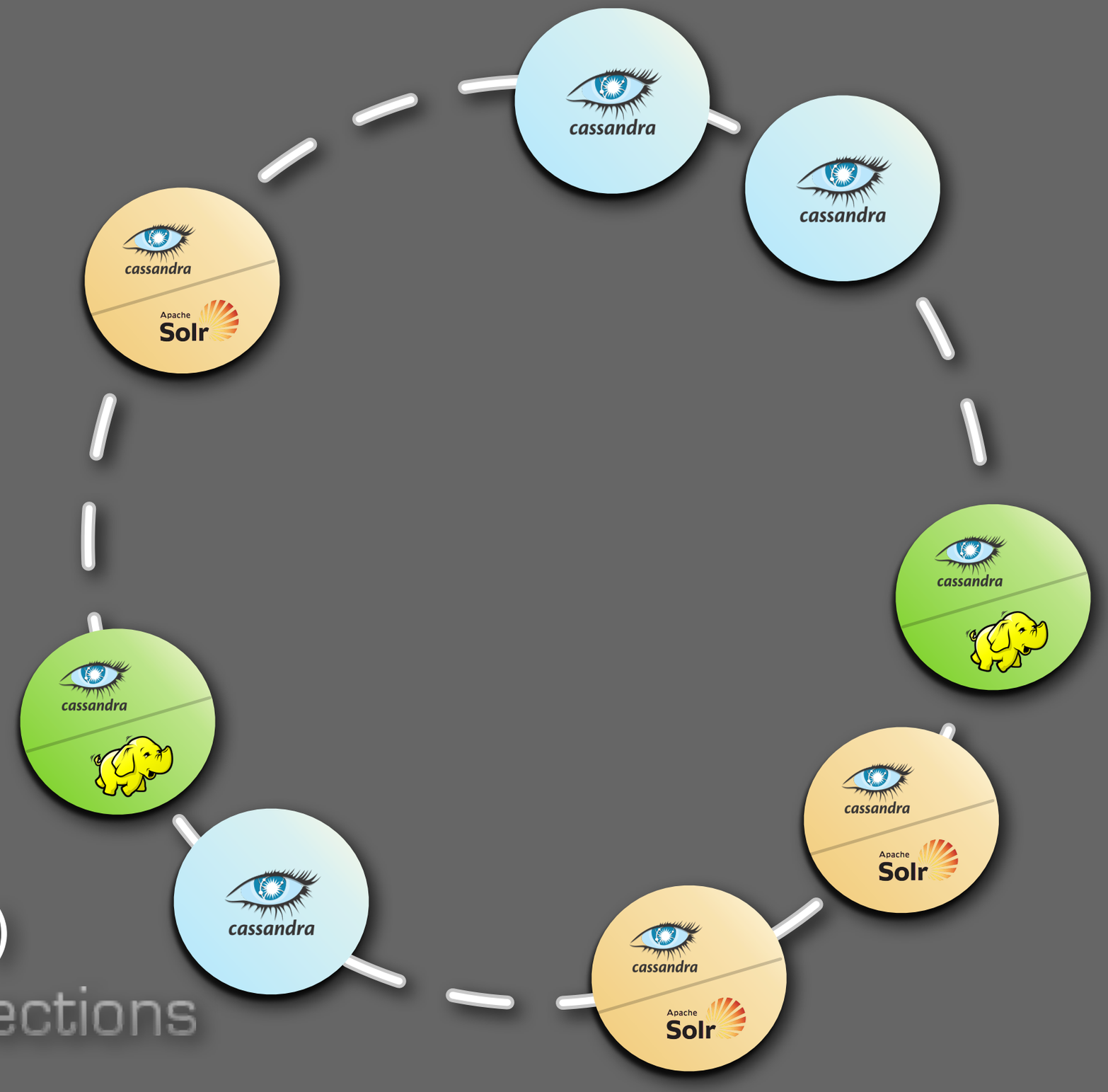
Real Time Considerations

- What about real time?
- Neither Hadoop nor Hive are built for real-time
- Cassandra provides you with data locality

Cassandra 2.0

- Transactions
- Triggers
- Prepared Statements

Q&A



@patriciagorla
pgorla@o19s.com
pgorla on IRC
(#cassandra, #python)

opensource connections