Scalable Activity Stream

Marcos Rebelo
oleber@gmail.com

Famous Quote

“Well done is better than well said.”

by Benjamin Franklin

Talk is cheap. Talking about a project won't get it completed. Taking action and seeing the task through to completion is the only way to get the job done.

Some terms

Activity: Something made by a Source, and that will be seen by Consumers.

Source: Someone that generate Activities.

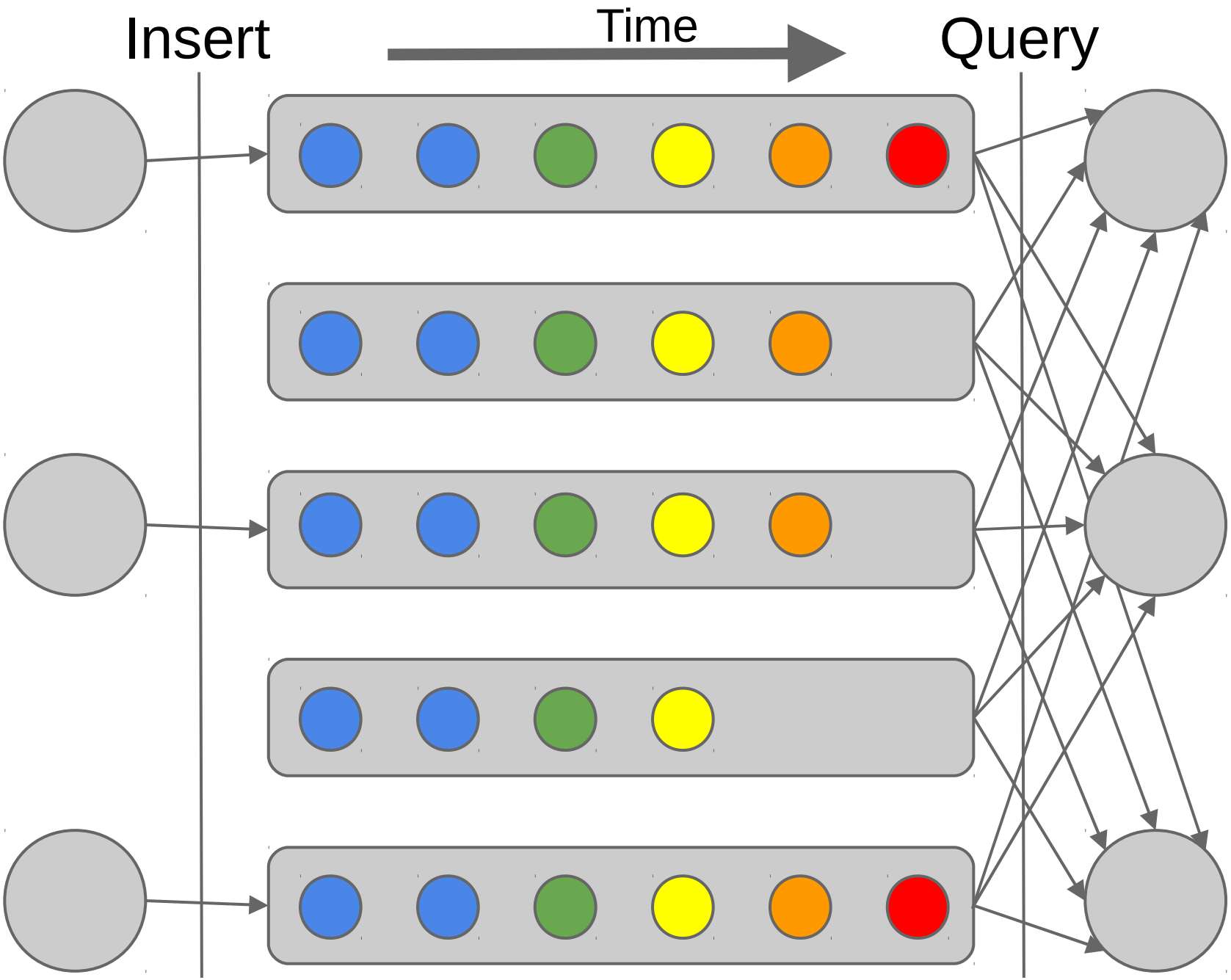
Consumer: Someone that follow 1 or more Source Activities.

Some solutions Solutions

Organize the data by:

- Source: the Consumers will search for all the Sources when requesting for the data.
- Consumer: copies of the data will be spread for all the consumers upfront.
- Date: to be discussed after.

Organize by Source


Insert
Time
Query

Organize by Source

Index by 'source_id' => 'time'

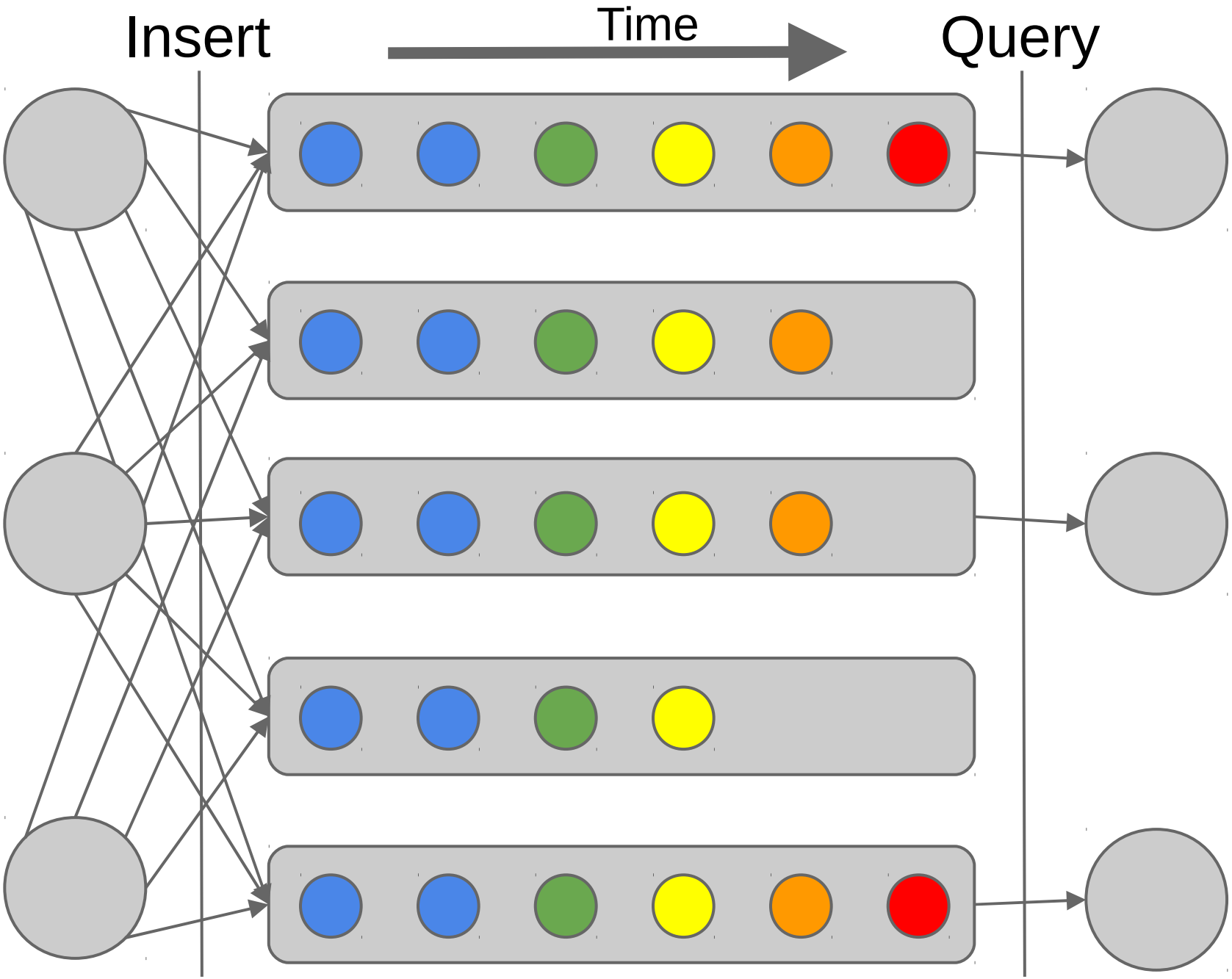
Advantages:

- The data isn't repeated.

Disadvantages:

- Querying each Source every time the data is needed, even inactive Sources.
- Old data is maintained in memory (index) for inactive Sources.
- Disastrous for Consumers following many Sources.

Organize by Consumer


Insert
Time
Query

Organize by Consumer

Insert by 'consumer_id' => 'time'

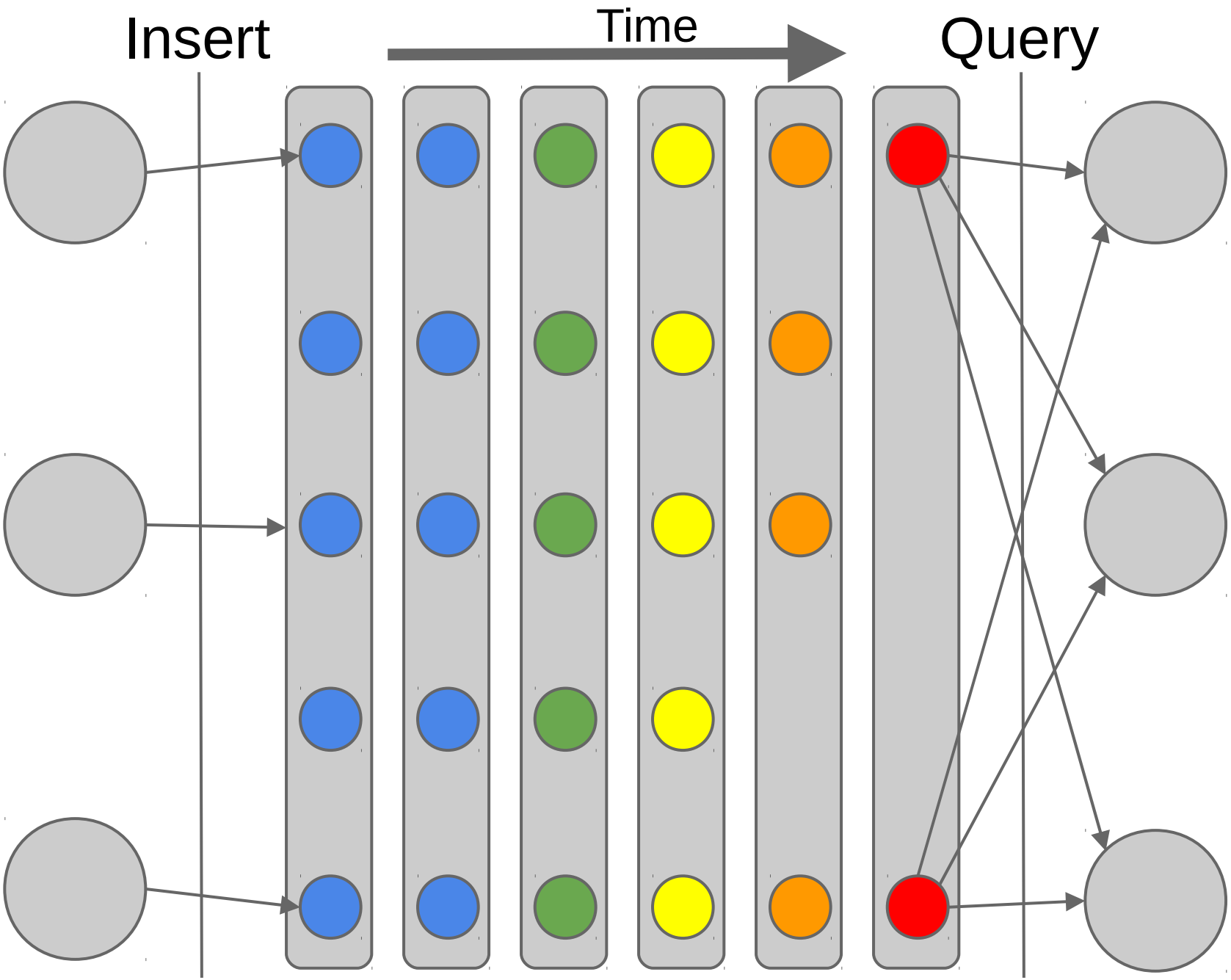
Advantages:

- Simple query to get the data.

Disadvantages:

- Copied information in each Consumer.
- Heavy Insert/Update/Delete of the Activities. Write upfront is disastrously slow for Sources with many Consumers.
- Complex maintenance of the Consumers following sources.

Organize by Time


Insert
Time
Query

Organize by Time

Index by 'time' => 'source_id'

Advantages:

- The data isn't repeated.
- Just actual data on memory (index).
- Old Activity from inactive Sources is ignored.

Disadvantages:

- Querying each Source every time the data is needed but not so bad Consumers following many Sources.

Managing time

- The idea is to Cluster information by time.
- Usually time has a granularity of a second, this means that in a day 86400 buckets.
- To resolve this unmanageable number of buckets, it is better to add a new field grouping multiple buckets.
- For example add the field 'day' or 'week' and index by 'day' => 'source_id'.

Caching Results

Cache Results

To avoid loading multiple times data from disk for active Consumers, it is possible to aggregate results previous requests.

The problem is detecting Source changes.

Solution:

- Add a 'status' field on each Source Cluster that changes every time that data changes.
- On the Consumer Cluster, keep the last Source Cluster 'status' for that period.
- Add the 'status' to the Source Cluster index.

Q. A.