Yandex

# Broadcast variables

# Broadcast variable

- Broadcast variable is a read-only variable that is efficiently shared among tasks

- Distribution is done by a torrent-like protocol (extremely fast!)

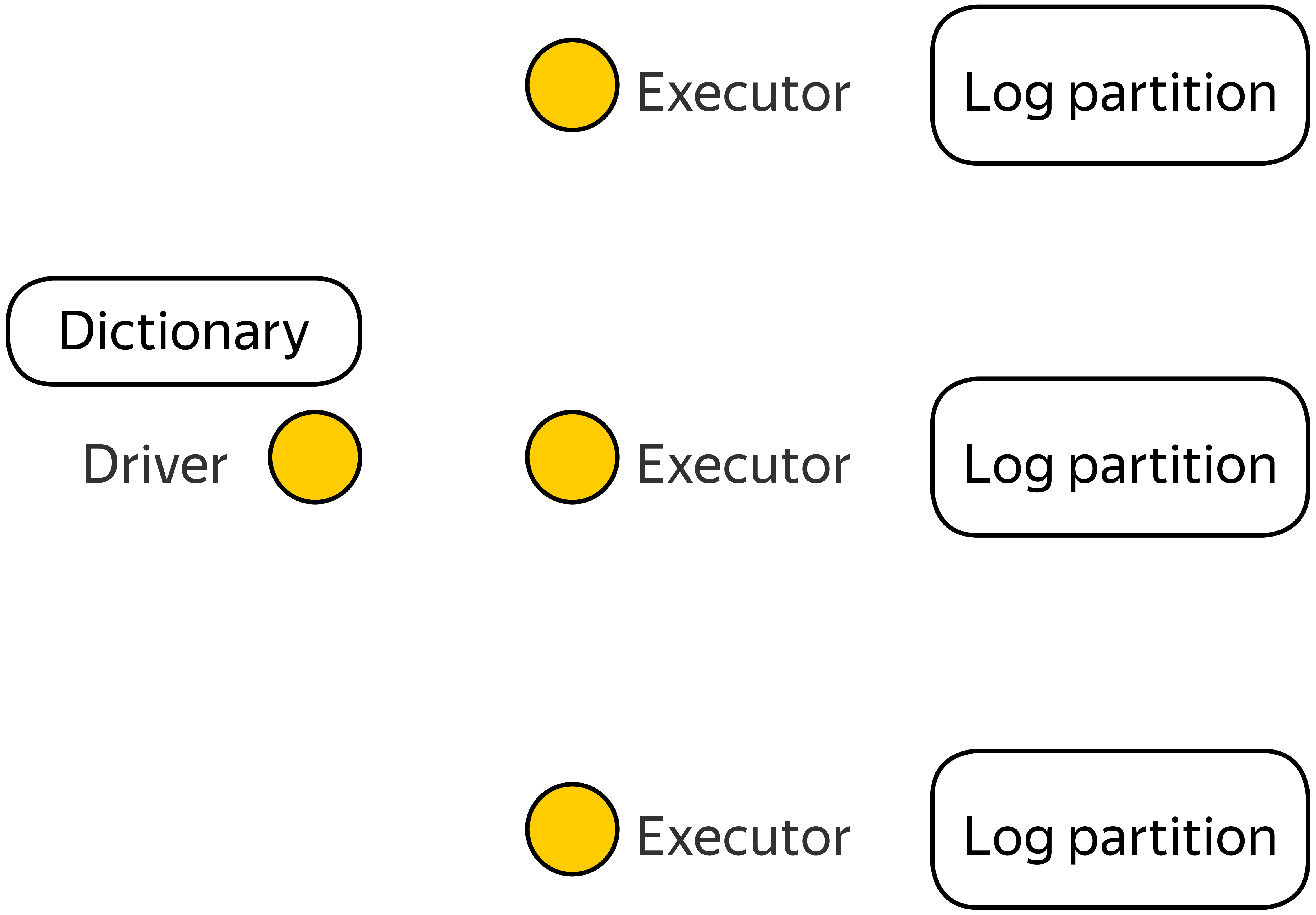
- Distributed efficiently compared to captured variables

# Motivating example

# Motivating example

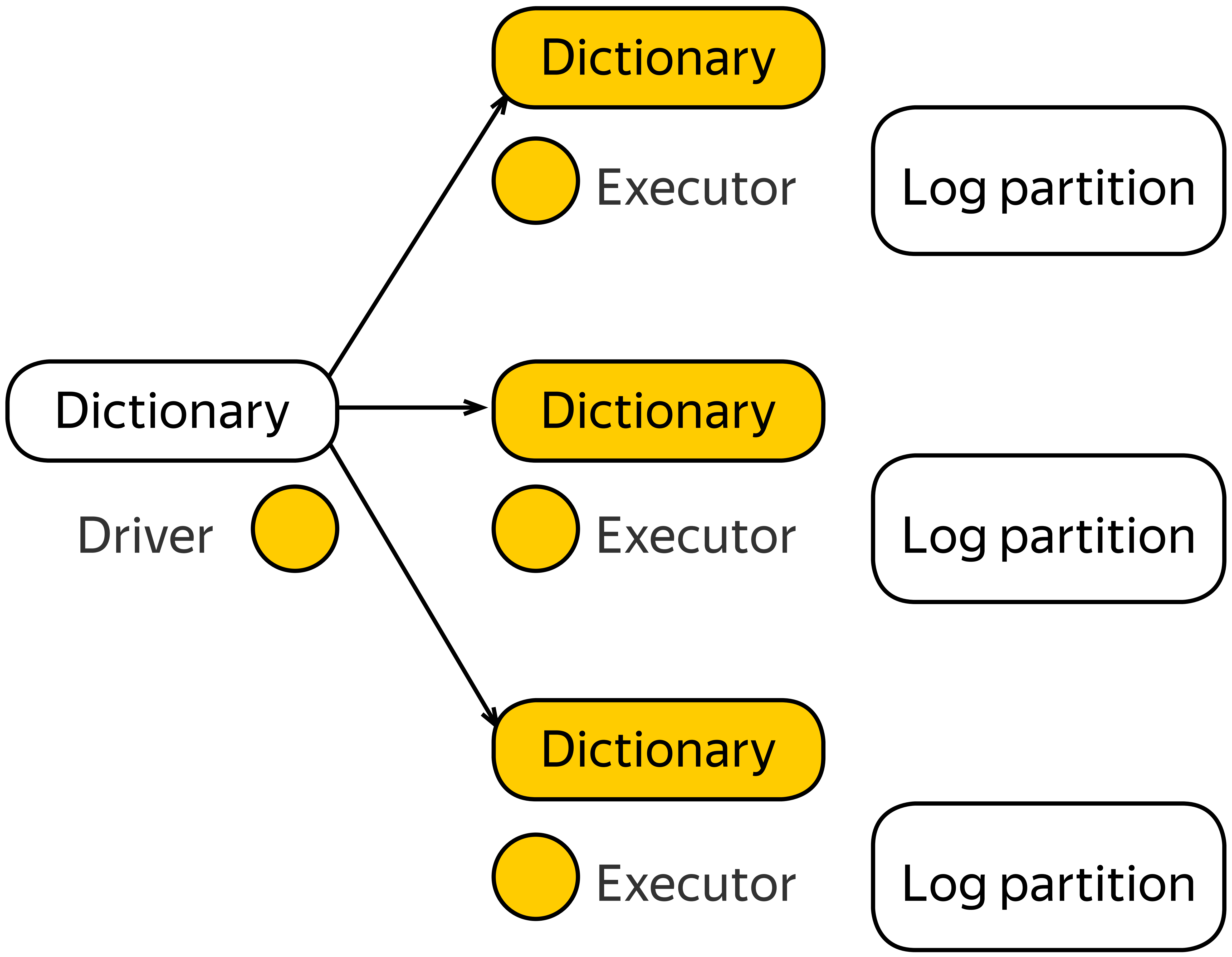
- Input:
  1TB partitioned log, 1GB IP dictionary

- Task:
  resolve IP addresses



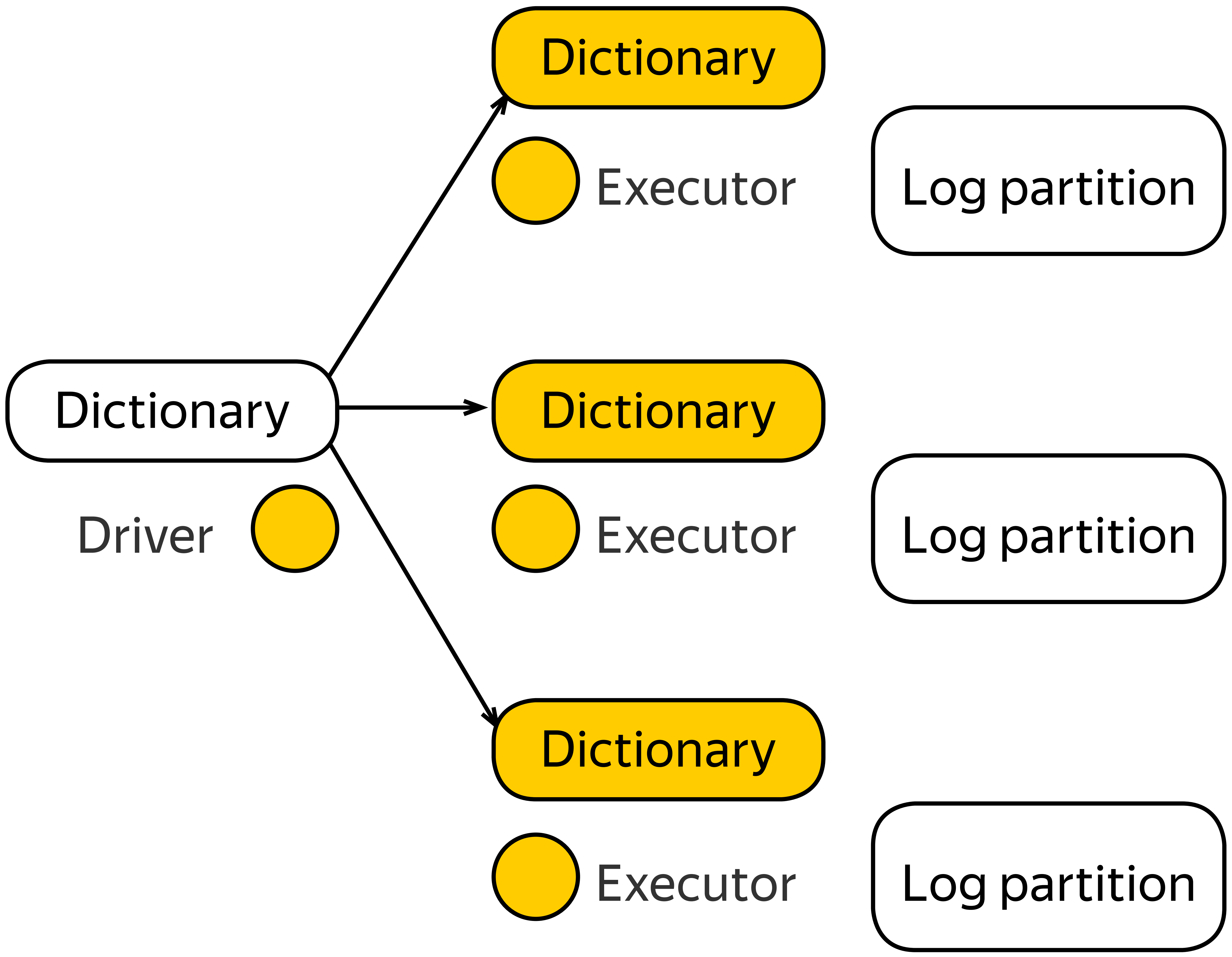
# Motivating example

- Input:
  1TB partitioned log, 1GB IP dictionary
- Task:
  resolve IP addresses
- Idea:
  distribute the dictionary
  query it locally



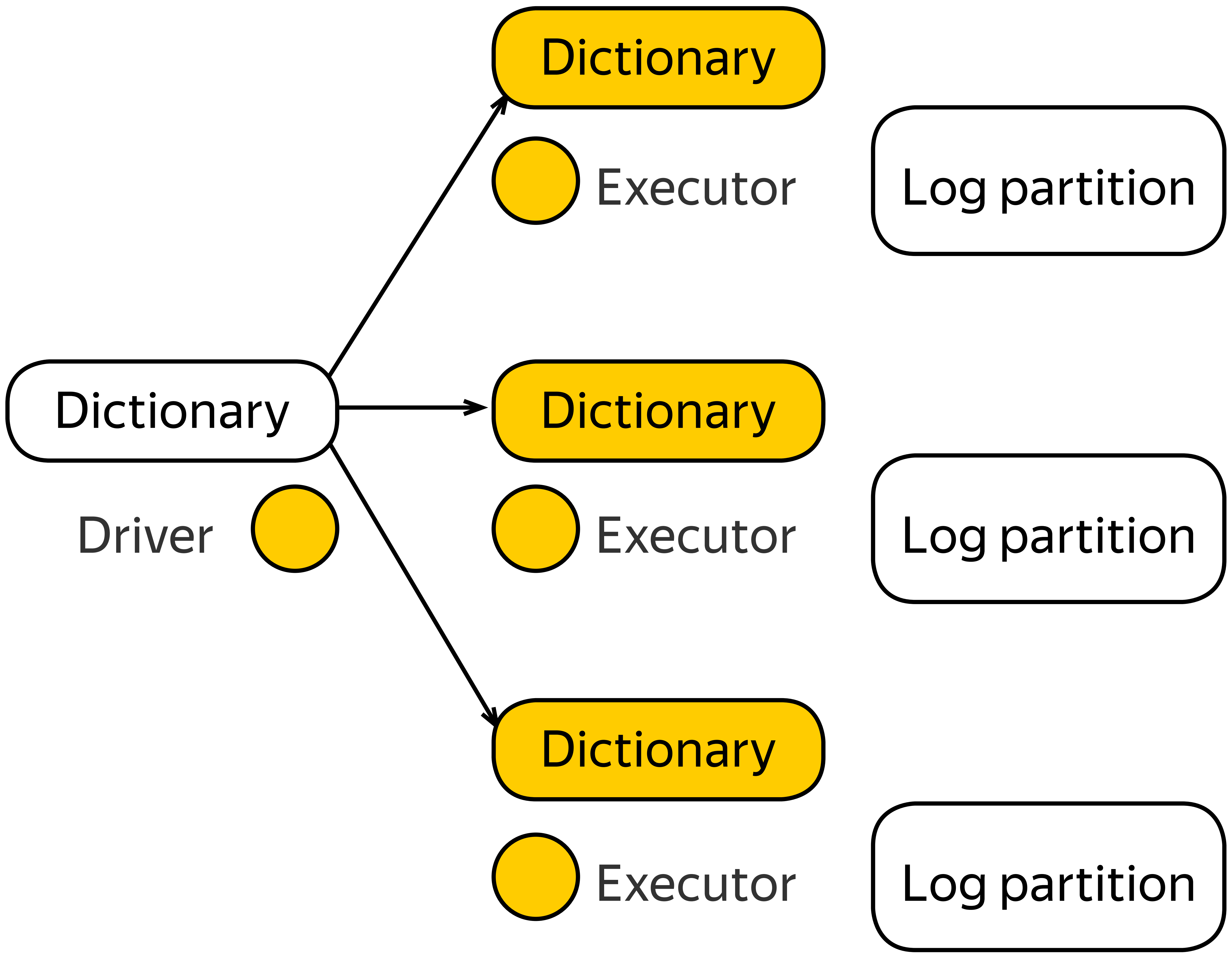
# Motivating example

Serial distribution via the closure
(from the driver to every executor)
~1000 (tasks) * 1GB = 1TB of traffic



# Motivating example

Serial distribution via the closure
(from the driver to every executor)
~1000 (tasks) * 1GB = 1TB of traffic



Parallel distribution via
the broadcast variable
(torrent-like)
~1-2 GB of traffic

Faster!

# Motivating example 2

```
sc = SparkContext(conf=...)

# compute the dictionary
my_dict_rdd = sc.textFile(...).map(...).filter(...)
my_dict_data = my_dict_rdd.collect()

# distributed the dictionary via the broadcast variable
broadcast_var = sc.broadcast(my_dict_data)

# use the broadcast variable within the task
my_data_rdd = sc.textFile(...).filter(
            lambda x: x in broadcast_var.value)
```

# Motivating example 2

```
sc = SparkContext(conf=...)

# compute the dictionary
my_dict_rdd = sc.textFile(...).map(...).filter(...)
my_dict_data = my_dict_rdd.collect()

# distributed the dictionary via the broadcast variable
broadcast_var = sc.broadcast(my_dict_data)

# use the broadcast variable within the task
my_data_rdd = sc.textFile(...).filter(
        lambda x: x in broadcast_var.value)
```

# Motivating example 2

```
sc = SparkContext(conf=…)

# compute the dictionary
my_dict_rdd = sc.textFile(…).map(…).filter(…)
my_dict_data = my_dict_rdd.collect()

# distributed the dictionary via the broadcast variable
broadcast_var = sc.broadcast(my_dict_data)

# use the broadcast variable within the task
my_data_rdd = sc.textFile(…).filter(
            lambda x: x in broadcast_var.value)
```

# Motivating example 2

```
sc = SparkContext(conf=...)

# compute the dictionary
my_dict_rdd = sc.textFile(...).map(...).filter(...)
my_dict_data = my_dict_rdd.collect()

# distributed the dictionary via the broadcast variable
broadcast_var = sc.broadcast(my_dict_data)

# use the broadcast variable within the task
my_data_rdd = sc.textFile(...).filter(
            lambda x: x in broadcast_var.value)
```

# Summary

- Broadcast variables are read-only shared variables with effective sharing mechanism

- Useful to share dictionaries, models

**Big**DATAteam