Advanced Programming in Spark

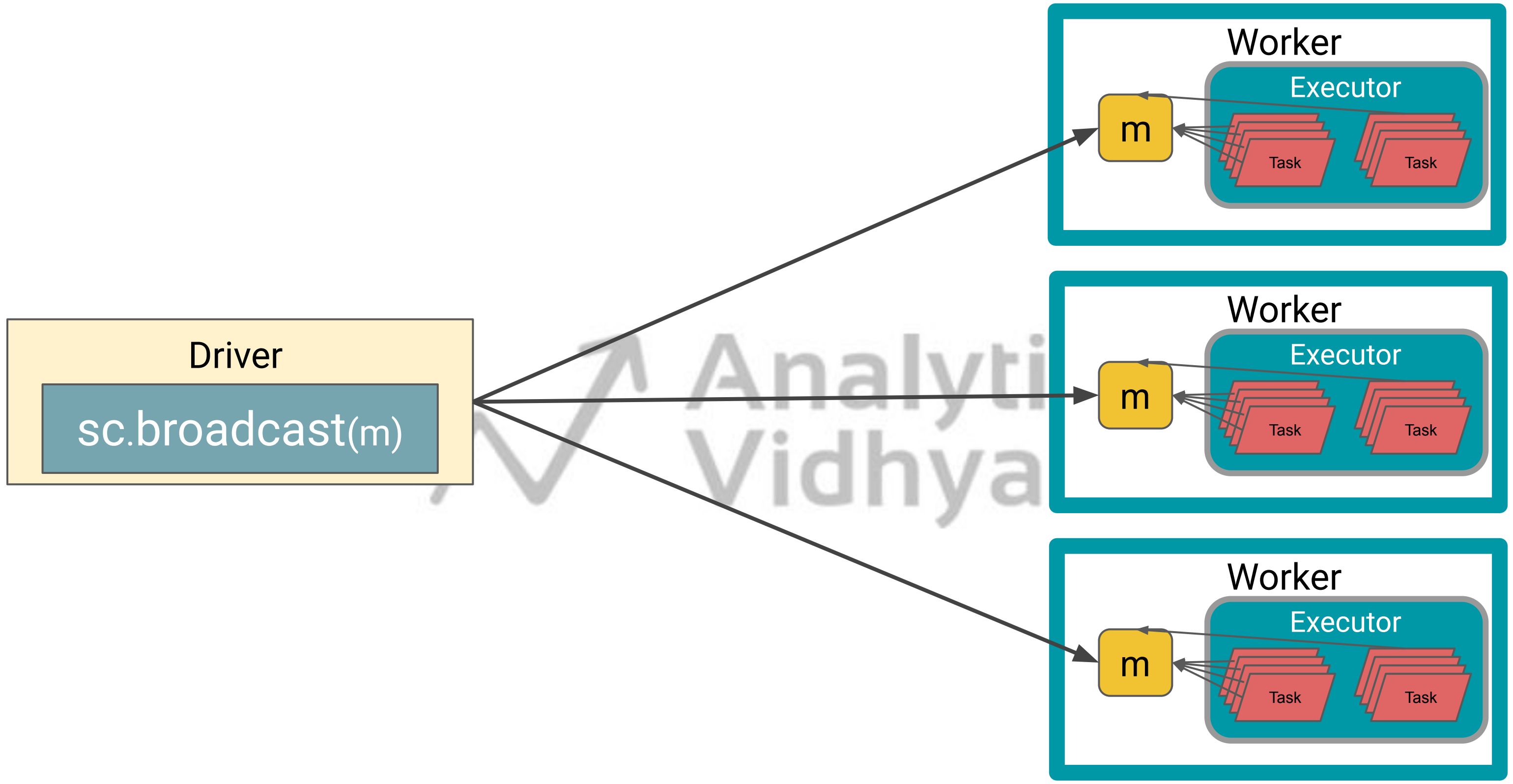
Shared Variables

Shared Variables

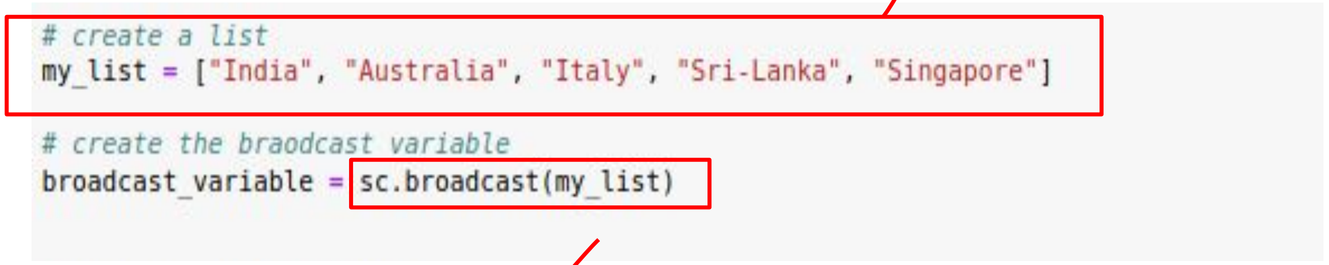
- Broadcast Variables
- Accumulators

Broadcast Variable

- Read-only variable cached on each node rather than shipping a copy of it with tasks
- They give every node a copy of a large input dataset in an efficient manner
- Attempts to distribute broadcast variables using efficient broadcast algorithms to reduce communication cost


Driver
sc.broadcast(m)
Worker
Executor
m
Task
Task
Worker
Executor
m
Task
Task
Worker
Executor
m
Task
Task

How to create Broadcast Variables?



```
# create a list
my_list = ["India", "Australia", "Italy", "Sri-Lanka", "Singapore"]

# create the braodcast variable
broadcast_variable = sc.broadcast(my_list)
```

Variable to be broadcasted

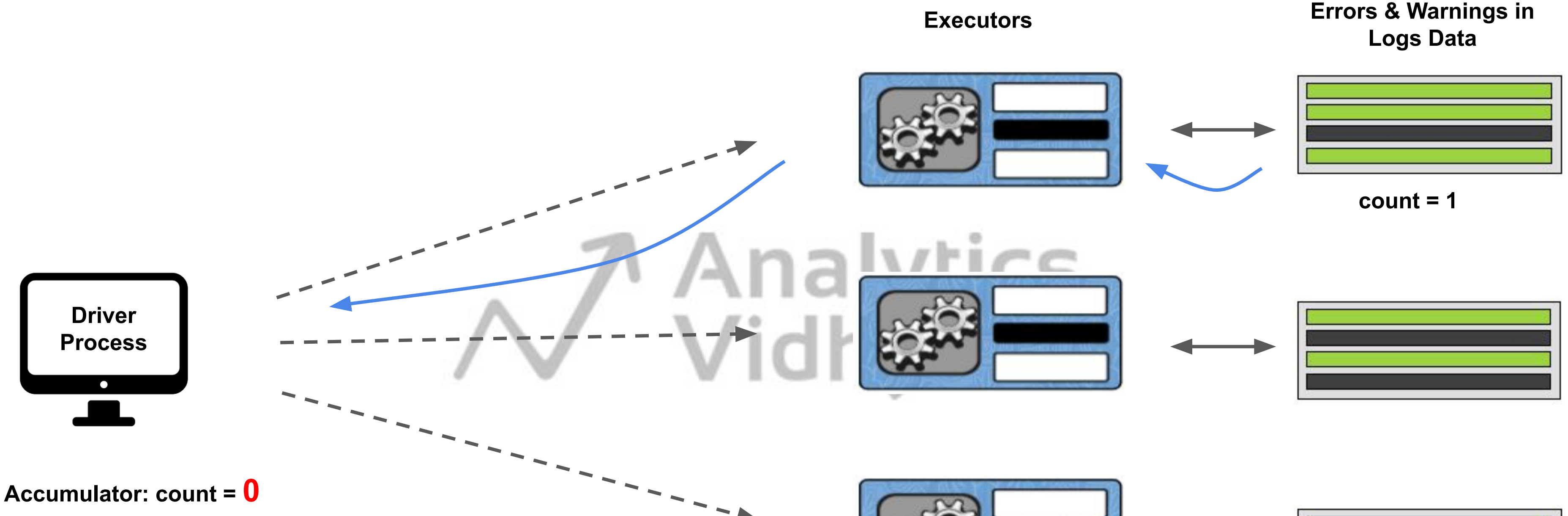
broadcast keyword with Spark Context

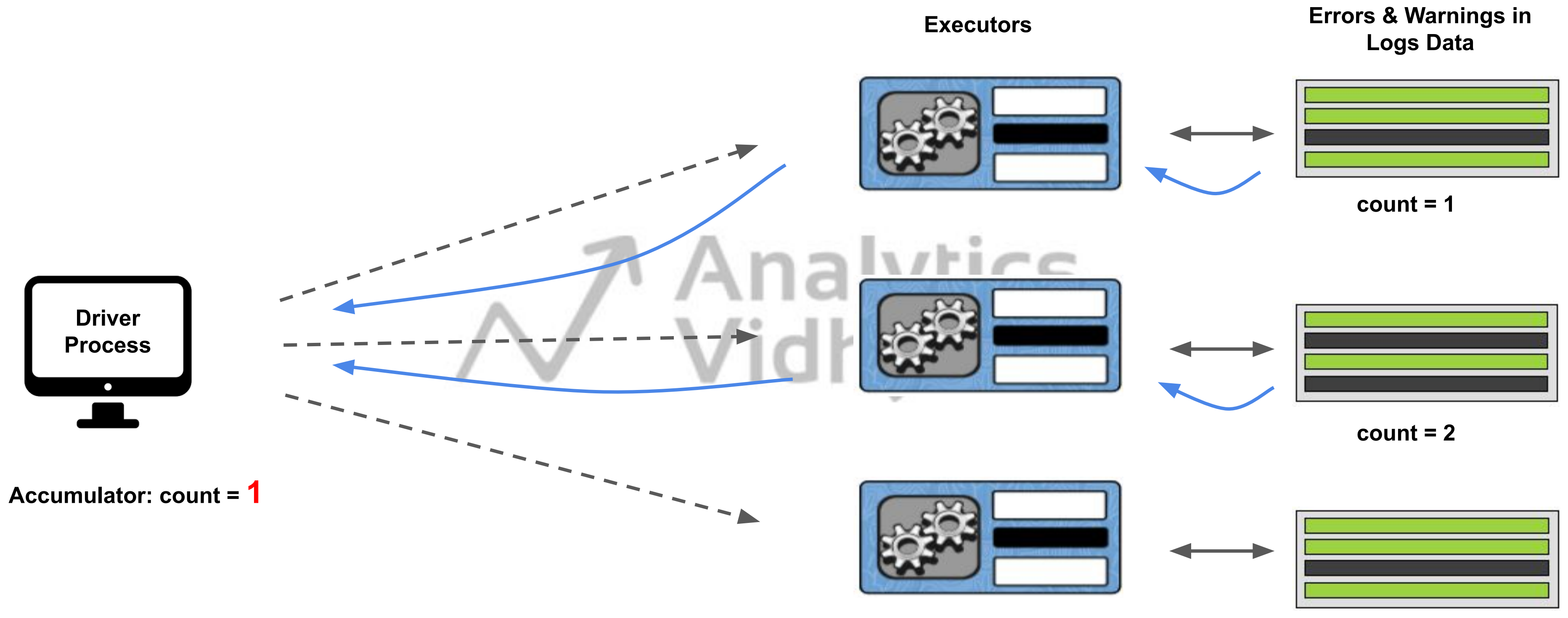
When to create Broadcast Variables?

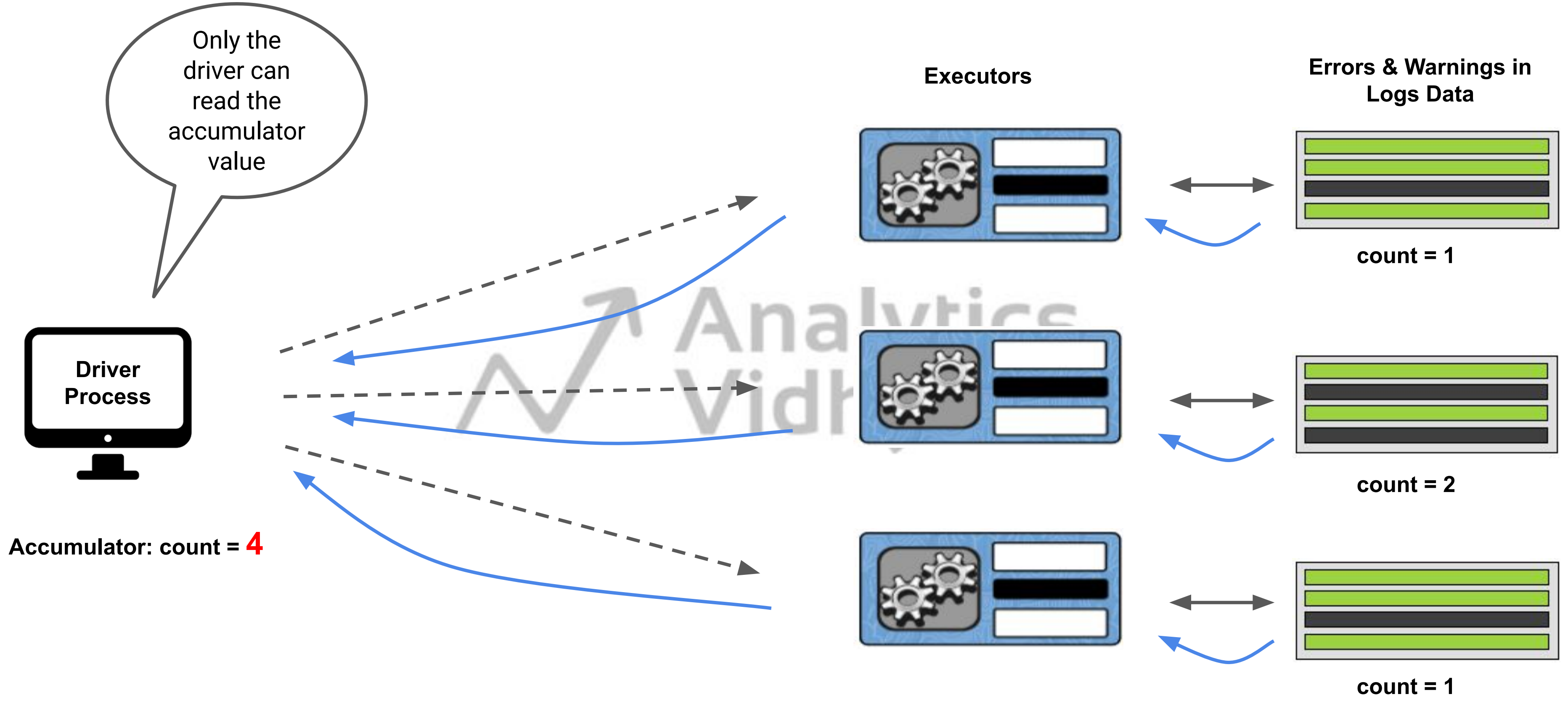
- When tasks across multiple stages need the same data.
- When caching the data in deserialized form is important.

Accumulators

- Are only created through an associative or a commutative operation.
- Can be used to implement counters or sums.
- Spark natively supports accumulators of numeric types, and programmers can add support for new types

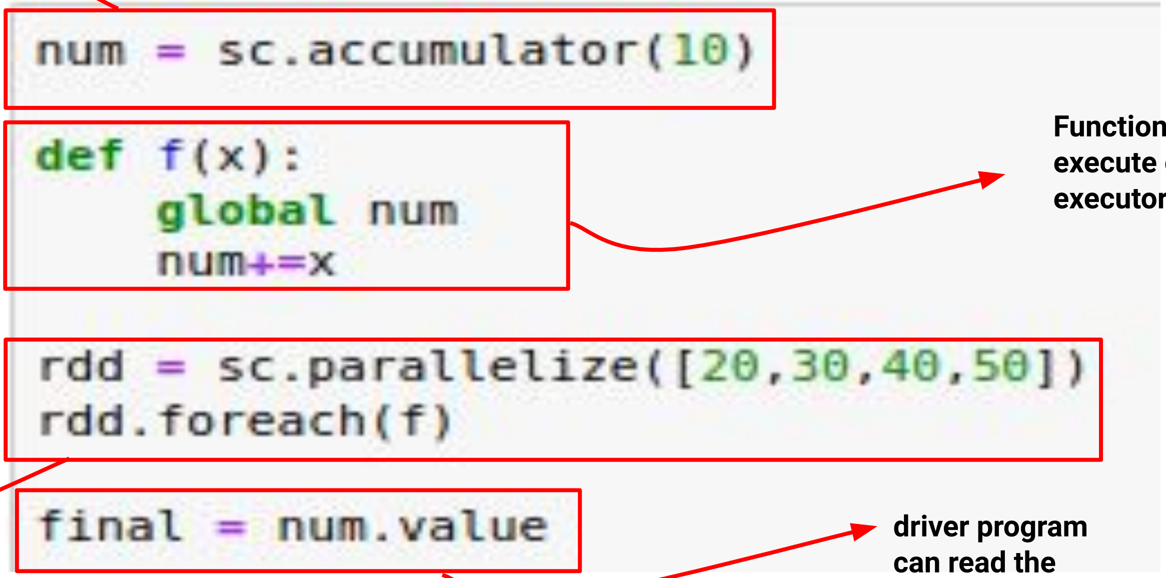

Executors
Errors & Warnings in Logs Data
count = 1
Driver Process
Accumulator: count = 0


Executors
Errors & Warnings in Logs Data
Driver Process
count = 1
count = 2
Accumulator: count = 1


Only the driver can read the accumulator value
Executors
Errors & Warnings in Logs Data
Driver Process
count = 1
count = 2
count = 1
Accumulator: count = 4

How to create Accumulators?

Accumulator keyword with Spark Context



```
num = sc.accumulator(10)
def f(x):
    global num
    num+=x

rdd = sc.parallelize([20,30,40,50])
rdd.foreach(f)
final = num.value
```

Function that will execute on each executor.

foreach function

driver program can read the accumulator value.

Analytics Vidhya

Thank You!!