Yandex

Accumulator variables

Accumulator variable

› Accumulator variable is a read-write variable that is shared among tasks

› Writes are restricted to increments!
  › i. e.: var += delta
  › addition may be replaced by any associate, commutative operation

Accumulator variable

› Accumulator variable is a read-write variable that is shared among tasks

› Writes are restricted to increments!
  › i. e.: var += delta
  › addition may be replaced by any associate, commutative operation

› Reads are allowed only by the driver program!

Example (similar to the previous video)
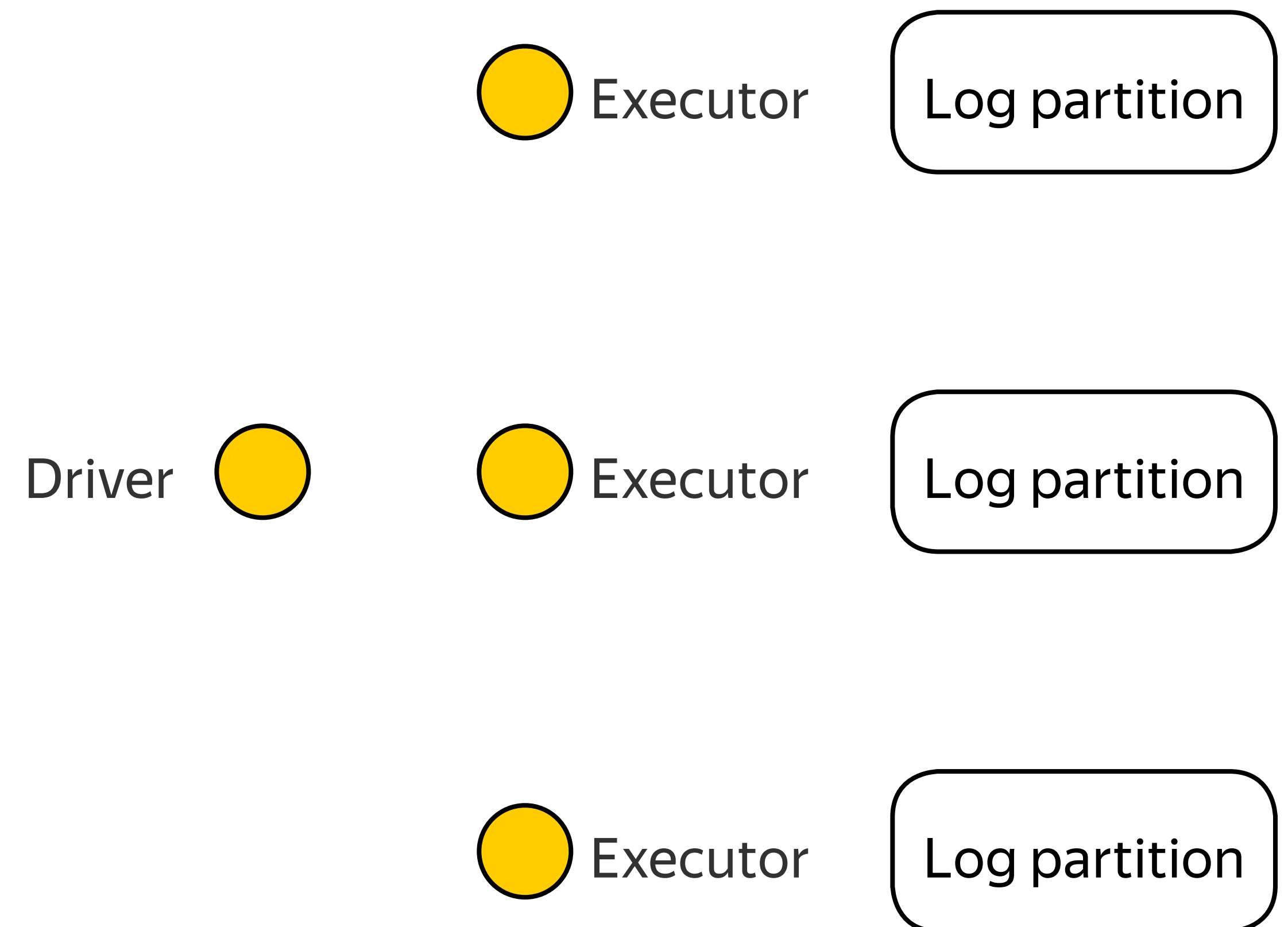
› <u>Input:</u>
1TB partitioned log

› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

Executor   Log partition

Driver   Executor   Log partition

Executor   Log partition

Example (similar to the previous video)

› <u>Input:</u>
1TB partitioned log

› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

VALID += 42

Executor   Log partition

Driver   Executor   Log partition

Executor   Log partition
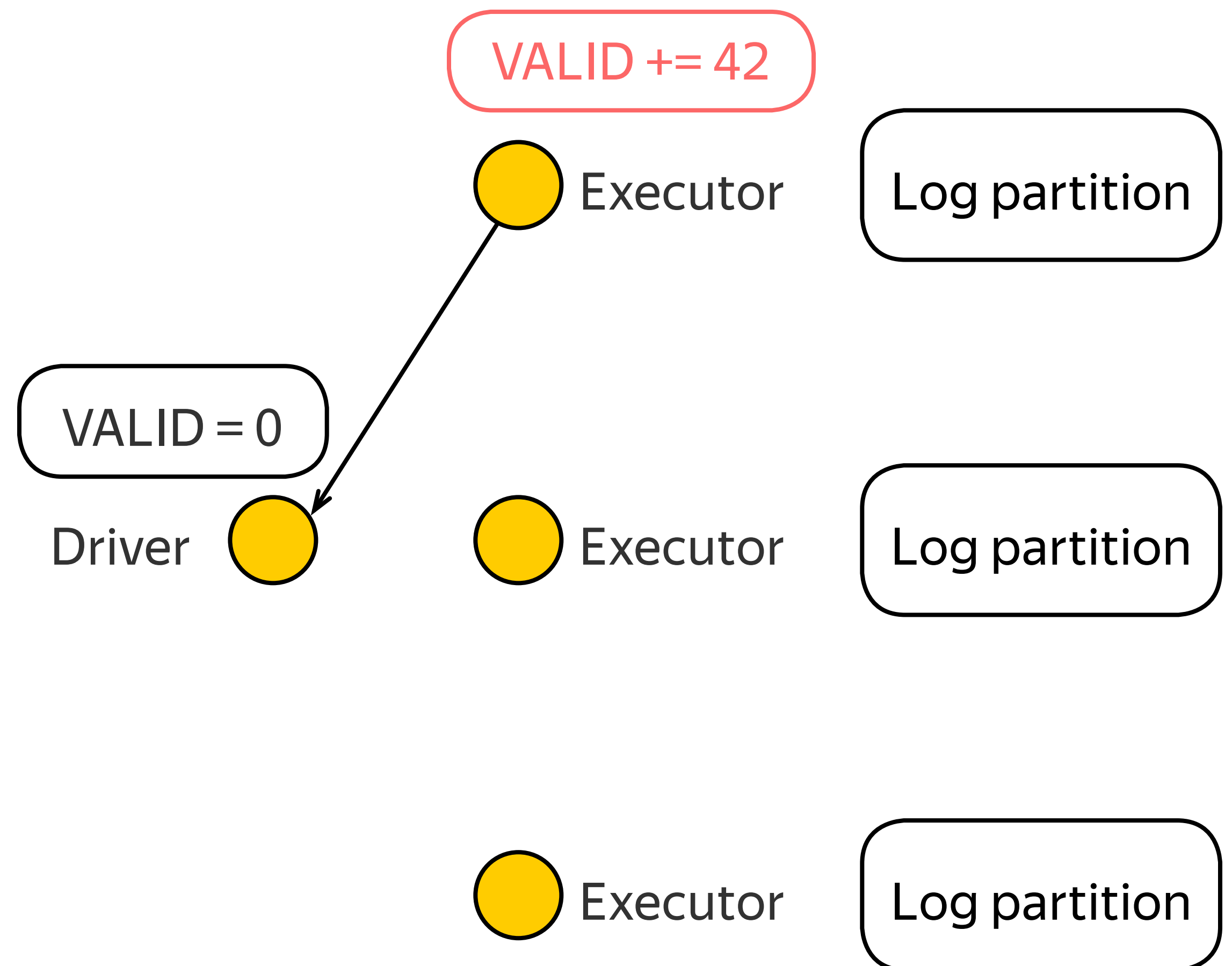
Example (similar to the previous video)

› <u>Input:</u>
1TB partitioned log

› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

VALID += 42

● Executor   Log partition

VALID = 0

Driver ●   ● Executor   Log partition

● Executor   Log partition

Example (similar to the previous video)

› <u>Input:</u>
1TB partitioned log
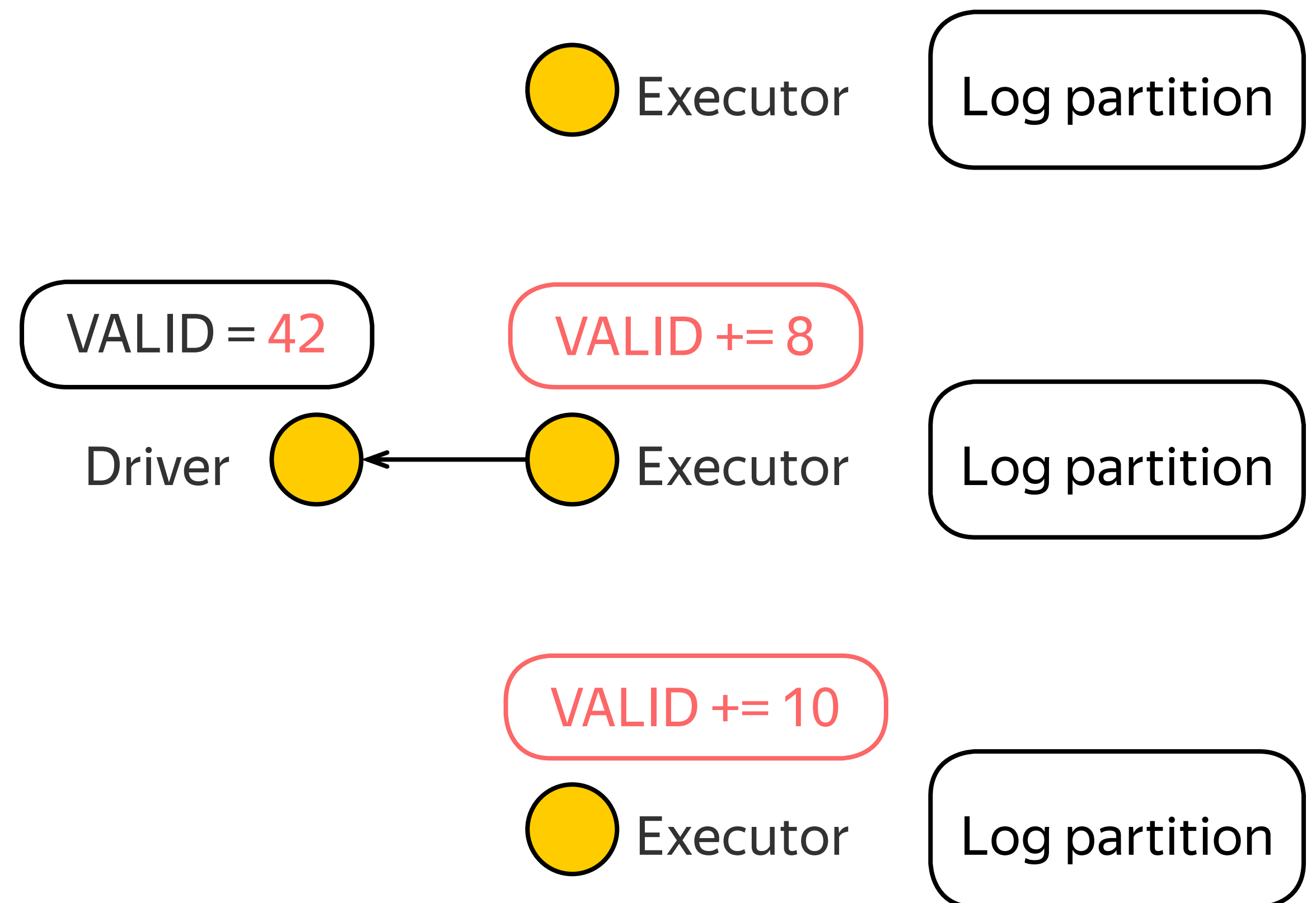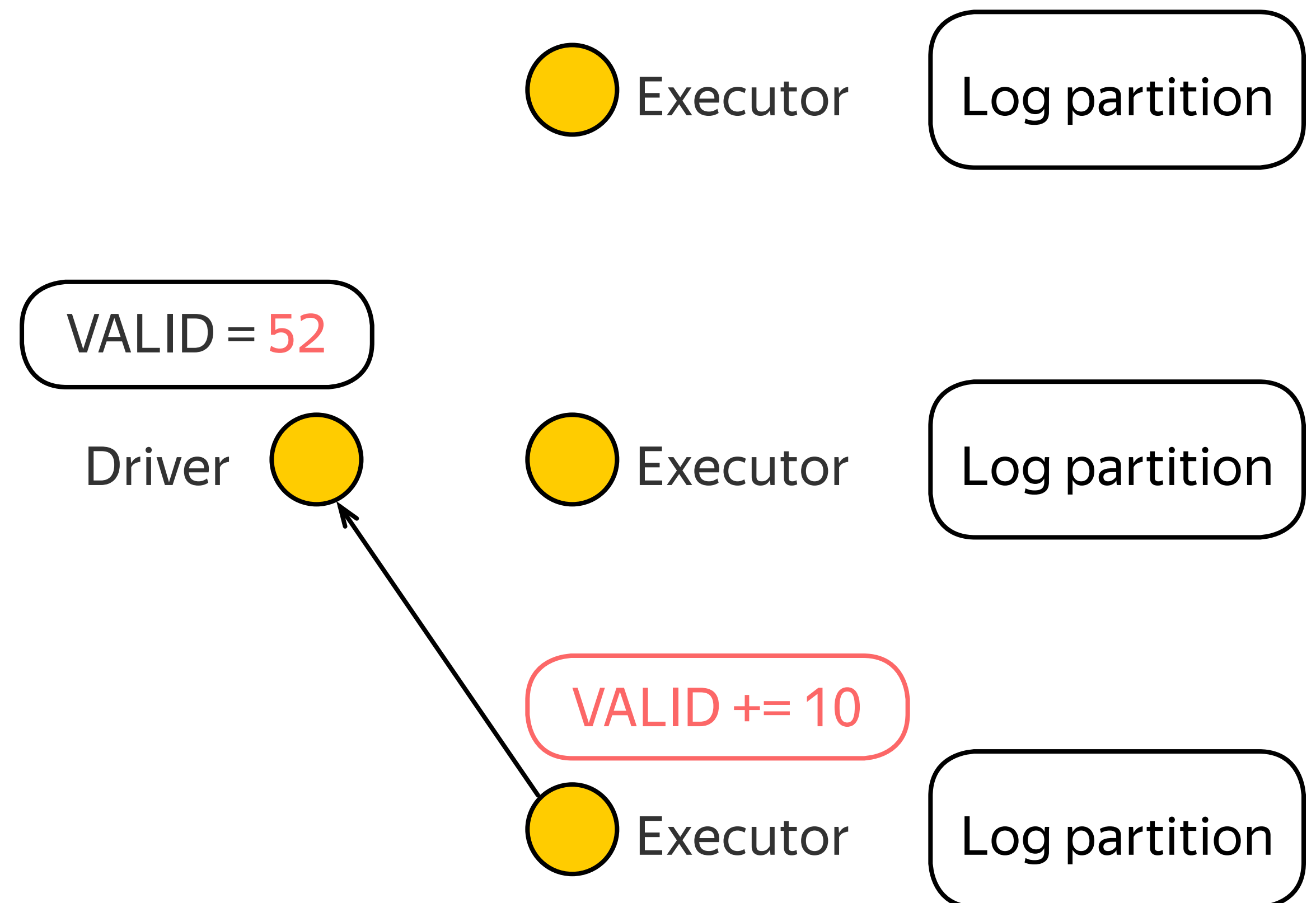
› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

Example (similar to the previous video)
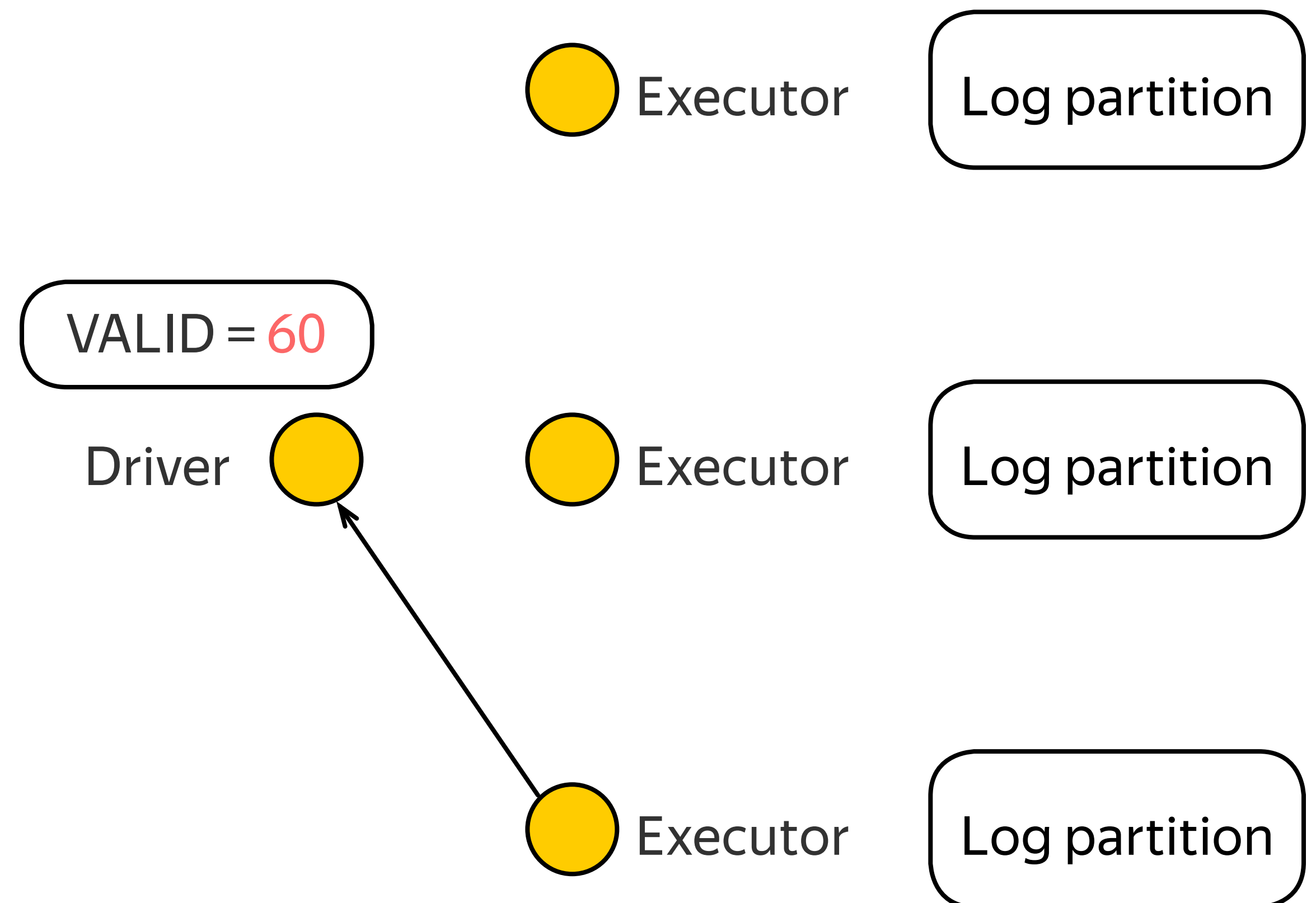
› <u>Input:</u>
1TB partitioned log

› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

Example (similar to the previous video)

› <u>Input:</u>
1TB partitioned log

› <u>Task:</u>
resolve IP addresses
AND
collect metrics:
# of valid records

Executor ⬤ | Log partition |

VALID = 60

Driver ⬤ | ⬤ Executor | Log partition |

⬤ Executor | Log partition |

Guarantees on the updates

› In actions updates are applied exactly once

› In transformations there are no guarantees as the transformation code may be re-executed

Use cases

› Performance counters
  › # of processed records, total elapsed time, total error and so on and so forth

› Simple control flow
  › conditionals: stop on reaching a threshold for corrupted records
  › loops: decide whether to run the next iteration of an algorithm or not

› Monitoring
  › export values to the monitoring system

› Profiling & debugging

Summary

› Accumulators are read-write shared variables with restricted updates
  › increments only
  › can use custom associative, commutative operation for the updates
  › can read the total value only in the driver

› Useful for the control flow, monitoring, profiling & debugging

BigDATAteam