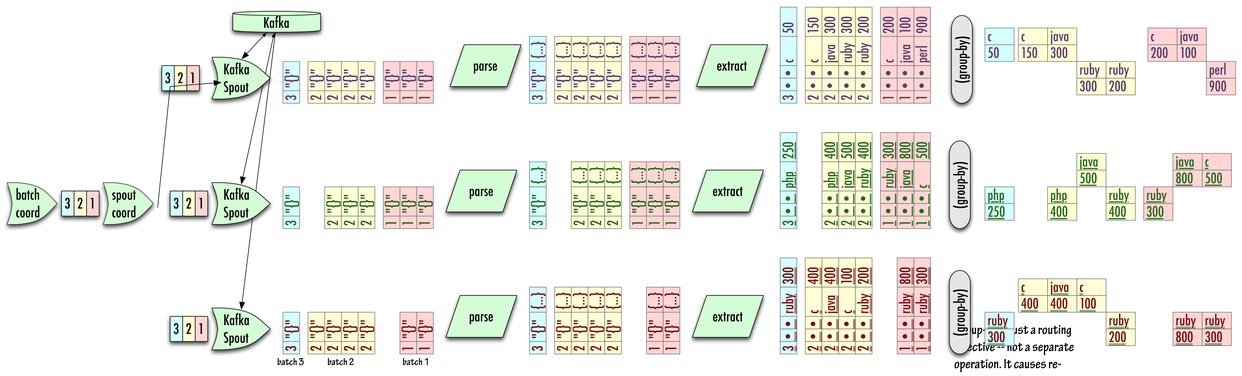


MBC dispenses tuples

spout coord assigns ranges

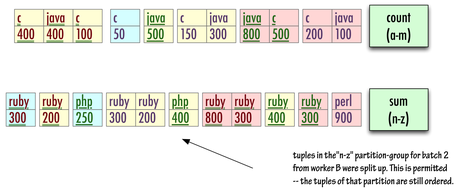
each range becomes a trident batch partition: here, 6 partitions

output fields are "anchored" on their ancestor tuple

executors process tuples in serial: partial order is preserved

up: just a routing ective - not a separate operation. It causes re-partitioning. There are now up to three partitions per batch

(Question: does Trident do a projection to just the relevant field for you, if it's not a combiner? Or does it ship the whole tuple across the network?)



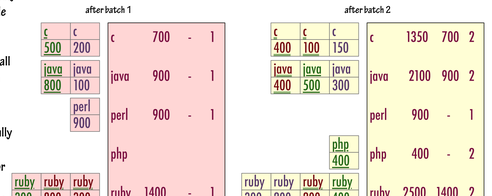
tuples in the "n-z" partition-group for batch 2 from worker B were split up. This is permitted -- the tuples of that partition are still ordered.

As each partition is completely received, sends `coord` tuple to MBcoordinator

Once coord tuples are in for all partitions in batch, sends a `commit` tuple to all

As each partition successfully finishes, sends a `success` tuple back to MBcoordinator

Once success tuples are in for all partitions in batch, MBcoord acks, satisfying the original batch tupletree



Each write records:
* current value
* prior value
* current txID

"a" now has curr value 4, prev value 1, and txID 2. "m" is unchanged: curr value 2, prev null, txID 1.

it's OK that "m" had a txID more than one behind: the batch guarantee still holds