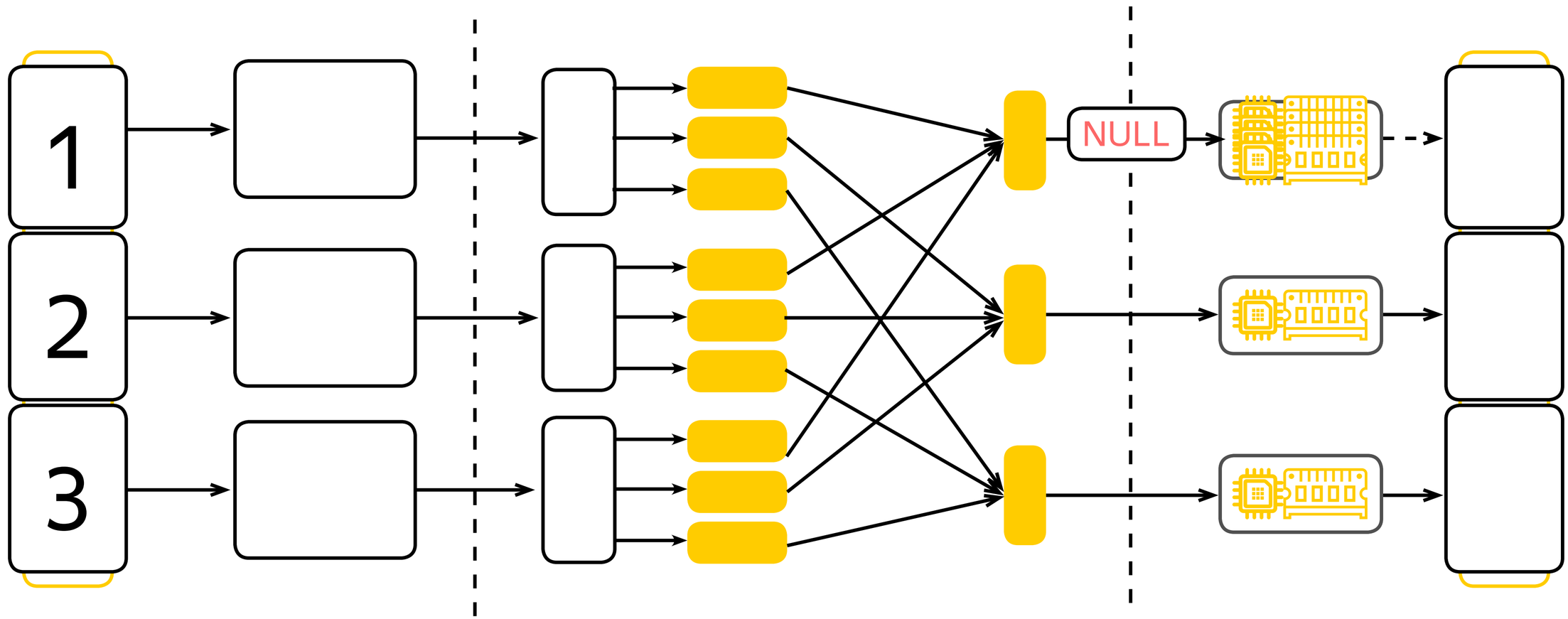
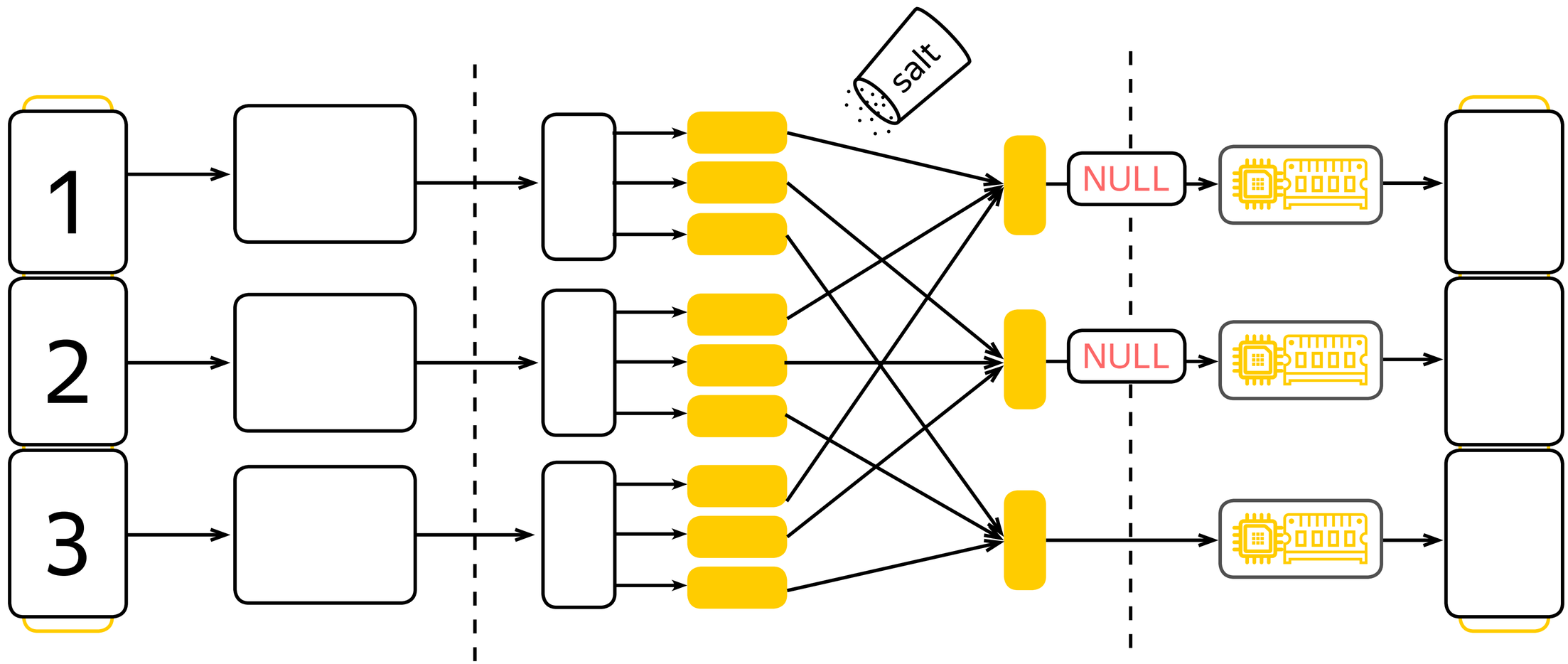
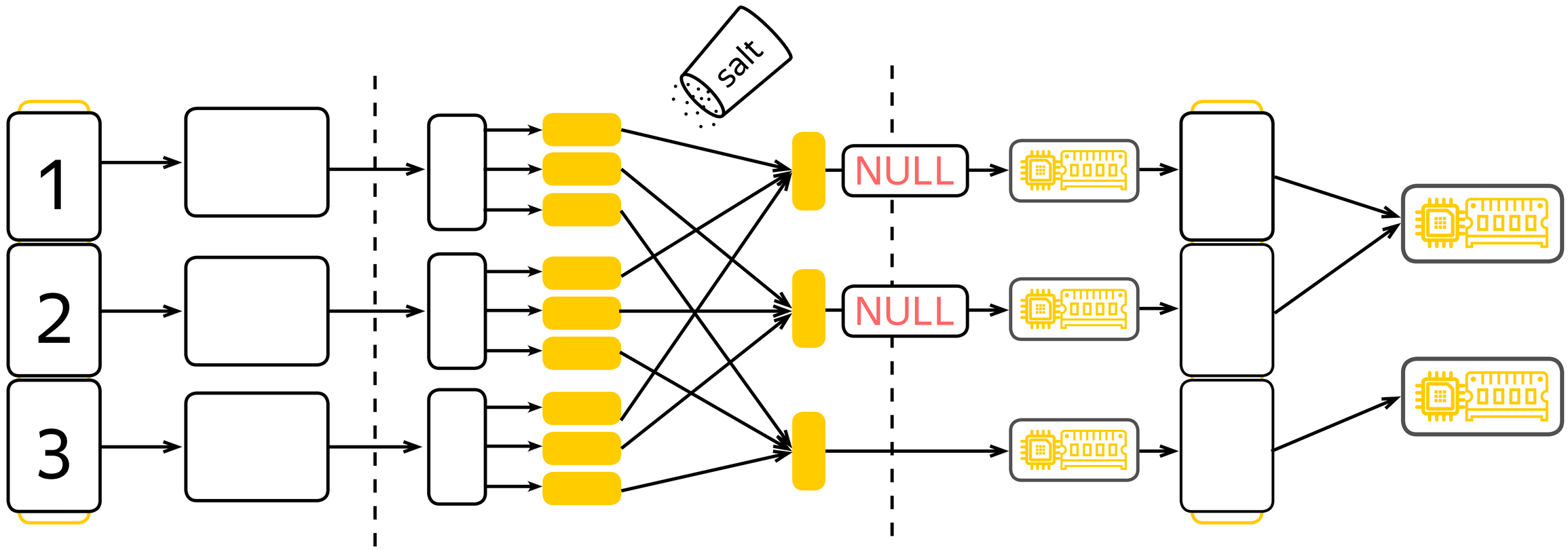
Yandex

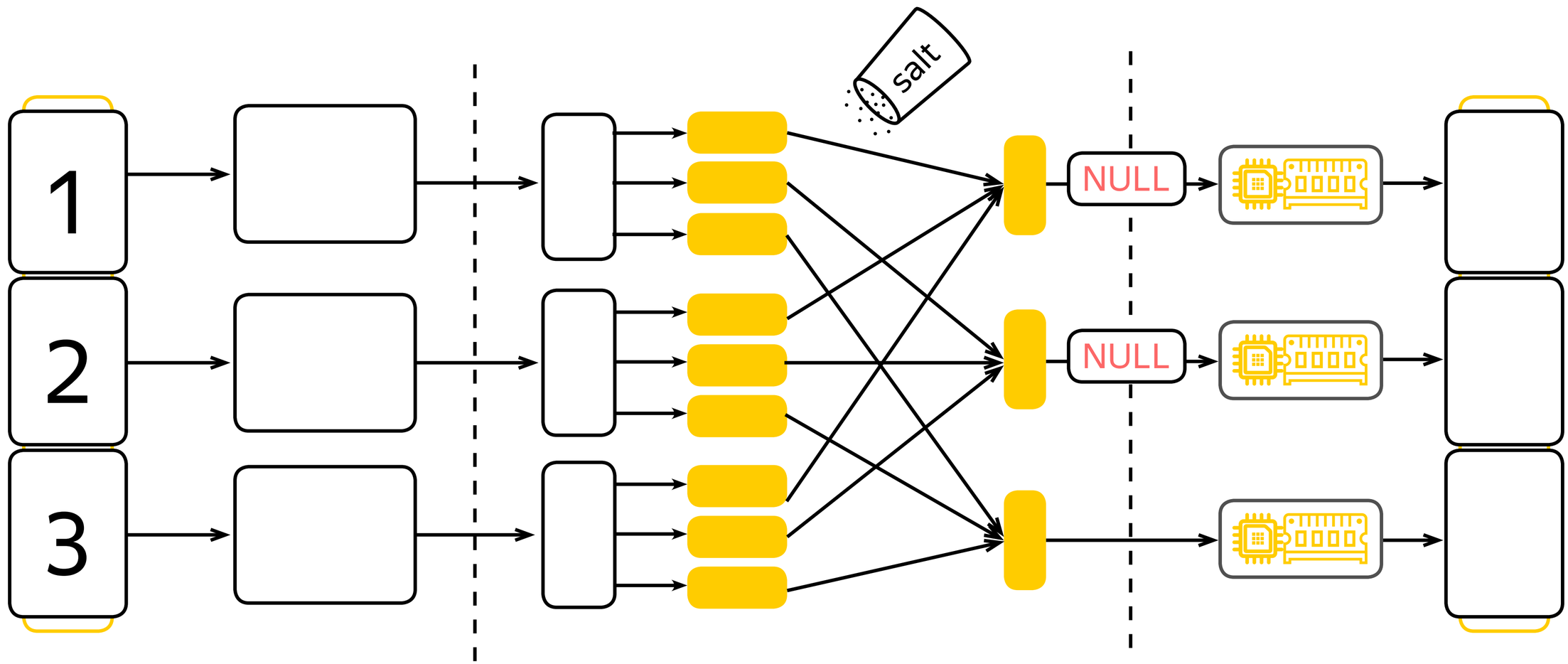
Telecommunications Analytics

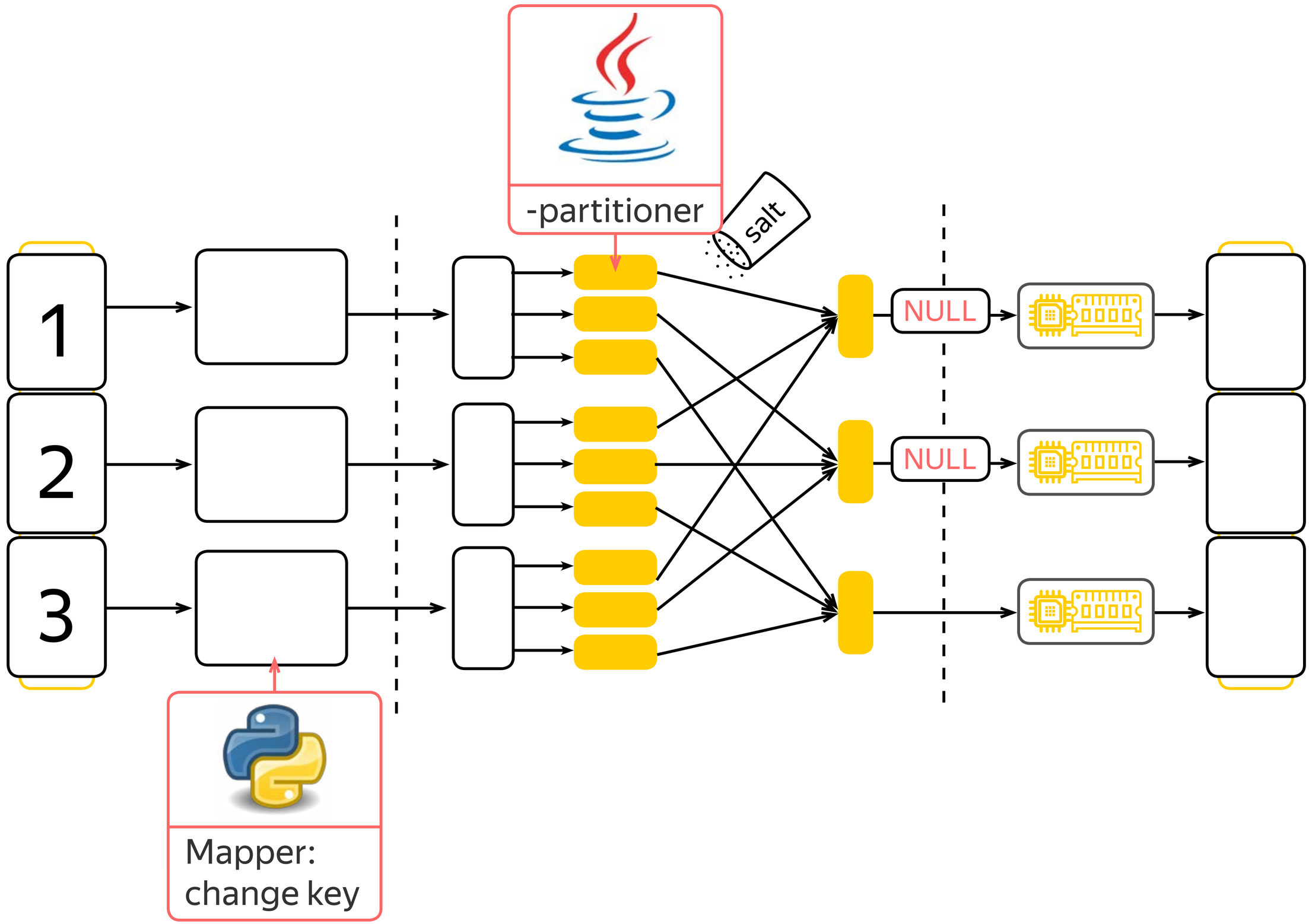
Data Skew, Salting


1
2
3
NULL


1
2
3
salt
NULL
NULL


1
2
3
salt
NULL
NULL


1
2
3
salt
NULL
NULL


-partitioner
salt
1
2
3
NULL
NULL
Mapper:
change key

```python
from random import random
geojson = json.load(open("milano-grid.geojson"))
grid = load_grid(geojson)
for line in sys.stdin:
    square_id, value_to_aggregate = line.rstrip("\n").split("\t", 1)
    square_id = int(square_id)
    grid_location = "null" if random() < 0.9 else grid[square_id]
    if value_to_aggregate:
        print(grid_location, value_to_aggregate, sep="\t")
```

```python
from random import random
geojson = json.load(open("milano-grid.geojson"))
grid = load_grid(geojson)
for line in sys.stdin:
    square_id, value_to_aggregate = line.rstrip("\n").split("\t", 1)
    square_id = int(square_id)
    grid_location = "null" if random() < 0.9 else grid[square_id]
    if value_to_aggregate:
        print(grid_location, value_to_aggregate, sep="\t")
```

```python
from random import randrange
grid_location = "null_{}".format(randrange(100)) if random() < 0.9 else grid[square_id]
```

...
null_58 40989.56529872355
null_67 40775.58025775422
null_76 42430.98650098723
null_85 41811.88806991089
null_94 41086.03092382825
...

...
null_58 40989.56529872355
null_67 40775.58025775422
null_76 42430.98650098723
null_85 41811.88806991089
null_94 41086.03092382825
...

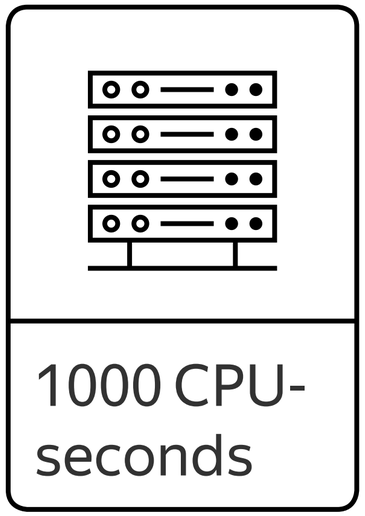
```
for line in sys.stdin:
    key, value = line.rstrip("\n").split("\t", 1)
    key = "null" if "null_" in key else key
    print("DoubleValueSum:{}".format(key), value, sep="\t")
```

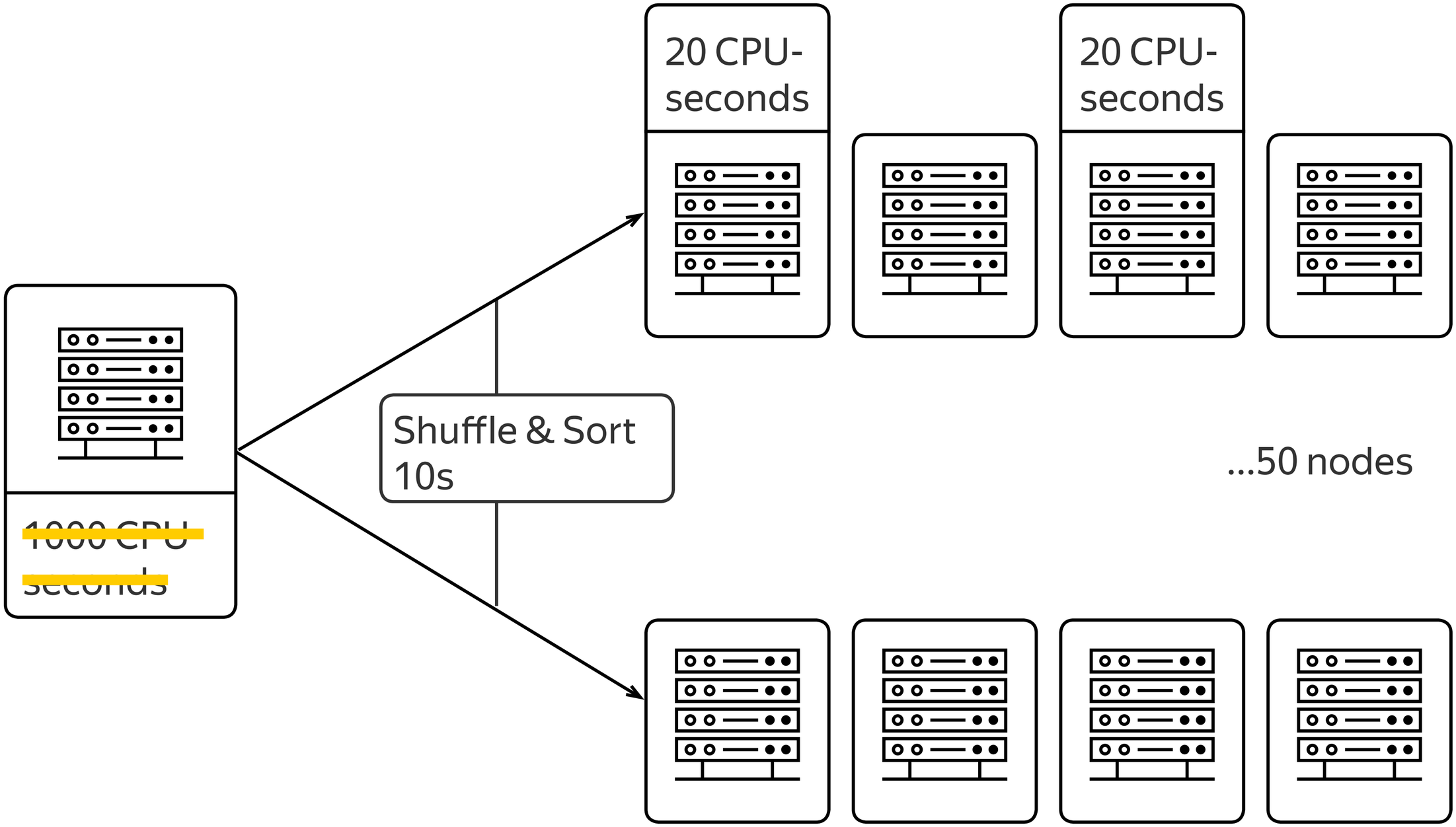
...
null_58 40989.56529872355
null_67 40775.58025775422
null_76 42430.98650098723
null_85 41811.88806991089
null_94 41086.03092382825
...

```
for line in sys.stdin:
    key, value = line.rstrip("\n").split("\t", 1)
    key = "null" if "null_" in key else key
    print("DoubleValueSum:{}".format(key), value, sep="\t")
```

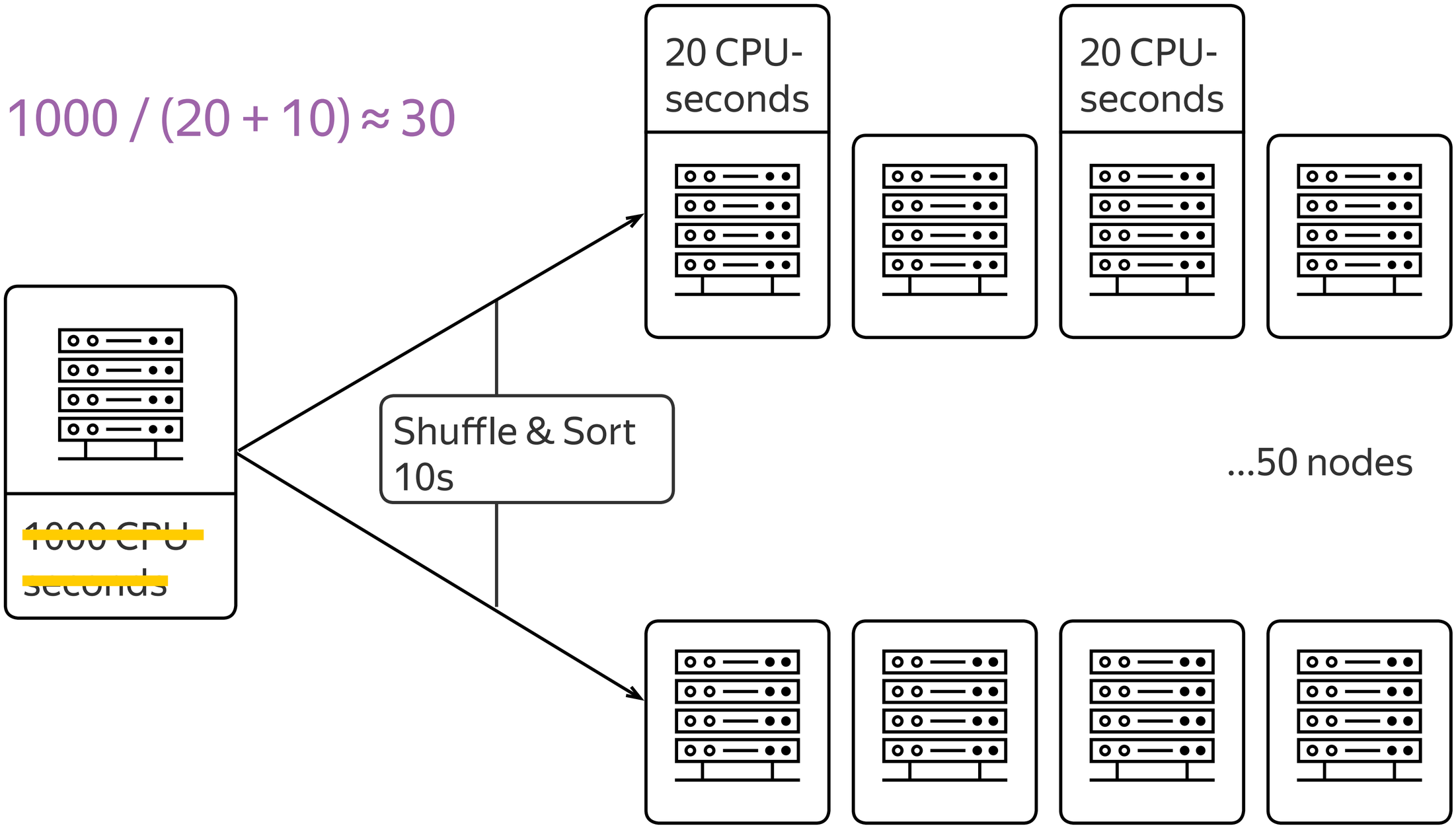
-reducer aggregate

South 164302.58197312435
null 4145425.004916422
North 296659.74407499237


1000 CPU-
seconds


20 CPU-seconds
20 CPU-seconds
Shuffle & Sort
10s
...50 nodes
~~1000 CPU seconds~~

$1000 / (20 + 10) \approx 30$



Summary

- you can process skewed dataset and tune parameters to achieve better parallelisation

BigDATAteam