Capacity Planning

Konstantin Knauf, Solutions Architect

© 2019 Ververica

Preparations

Do the Math!

- Resource requirements in terms of
  - #keys, state per key
  - #records, record size
  - #state updates
- What are your SLAs?
  - latency during normal operations
  - latency during recovery after a process/machine/site-failure

Preparations

Establish a Baseline

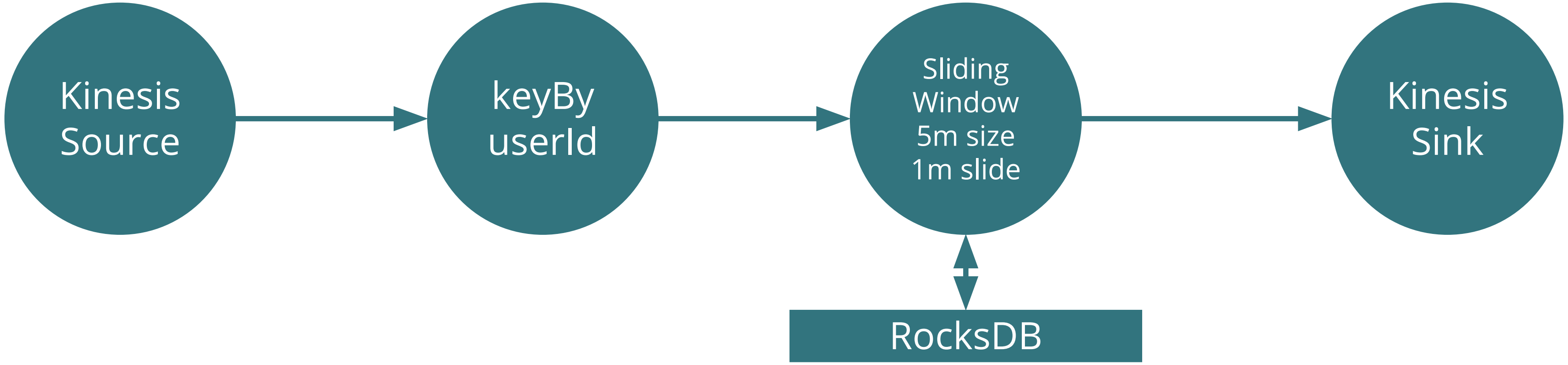
- Avoid back pressure during normal operations
- Add a margin for “catch up” during recovery
- Consider spiky load & expected growth in your application
- Consider checkpointing during capacity planning

© 2019 Ververica

Example

Data & Job

- Data
  - Message Size: 2KB
  - Throughput: 1,000,000 msg/s
  - Distinct keys: 500,000,000 (aggregation in widow: 4 longs per key)
  - Checkpoint every minute (*Result of SLAs*)
- Streaming Job



Example

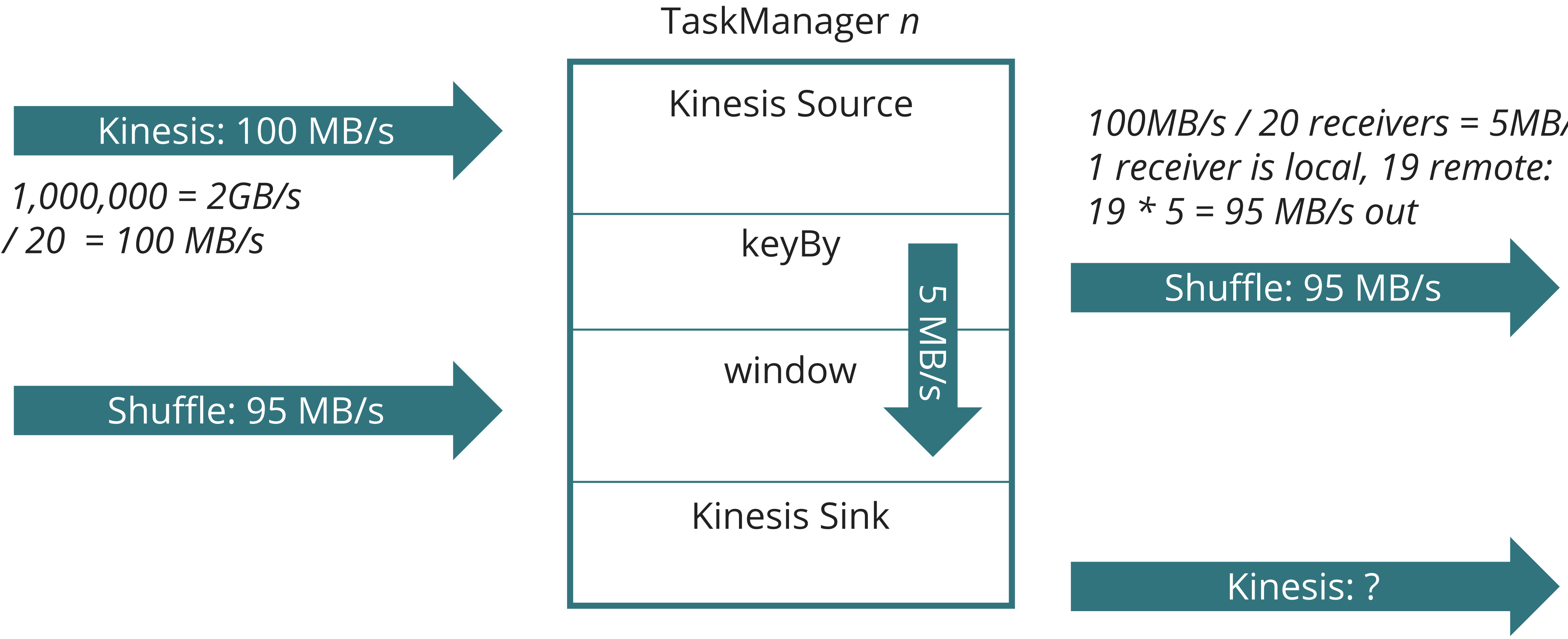
Target Deployment Environment

- EKS
- S3 for Checkpoints
- Instance Storage for local RocksDB instance
- (20 Pods)

Example

A Pod’s Perspective (20 Pods Overall)

*2 KB * 1,000,000 = 2GB/s*
2GB/s / 20 = 100 MB/s



100MB/s / 20 receivers = 5MB/s
1 receiver is local, 19 remote:
*19 * 5 = 95 MB/s out*

Example - Excursion

Window Emit

How much data is the window emitting?

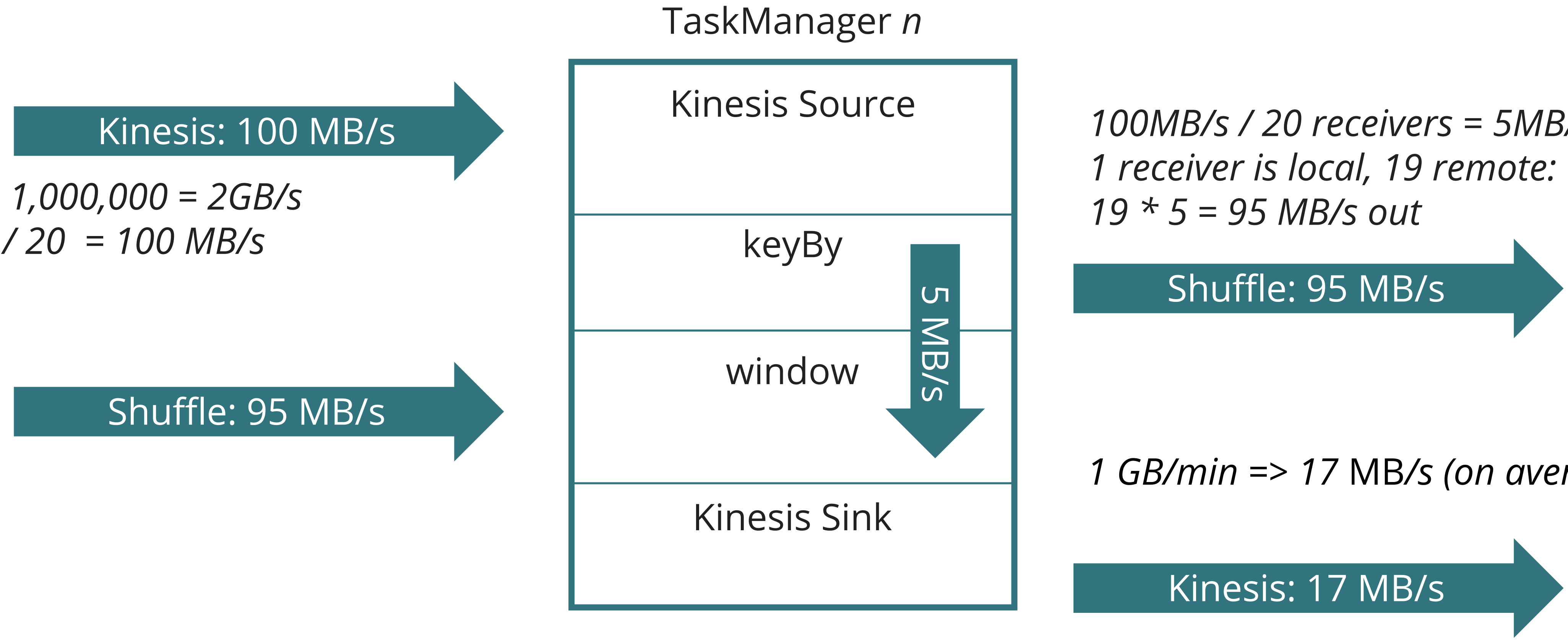
Recap: 500,000,000 unique users (4 longs per key)
Sliding window of 5 minutes, 1 minute slide

Assumption: For each user, we emit 2 ints (user_id, window_ts) and 4 longs from the aggregation = 2 * 4 bytes + 4 * 8 bytes = 40 bytes per key

25,000,000 (users) * 40 bytes = **1 GB every minute from each machine**

Example

A Pod’s Perspective (20 Pods)



TaskManager *n*

Kinesis: 100 MB/s

*2 KB * 1,000,000 = 2GB/s*
2GB/s / 20 = 100 MB/s

Shuffle: 95 MB/s

Kinesis Source

keyBy

window

5 MB/s

Kinesis Sink

100MB/s / 20 receivers = 5MB/s
1 receiver is local, 19 remote:
*19 * 5 = 95 MB/s out*

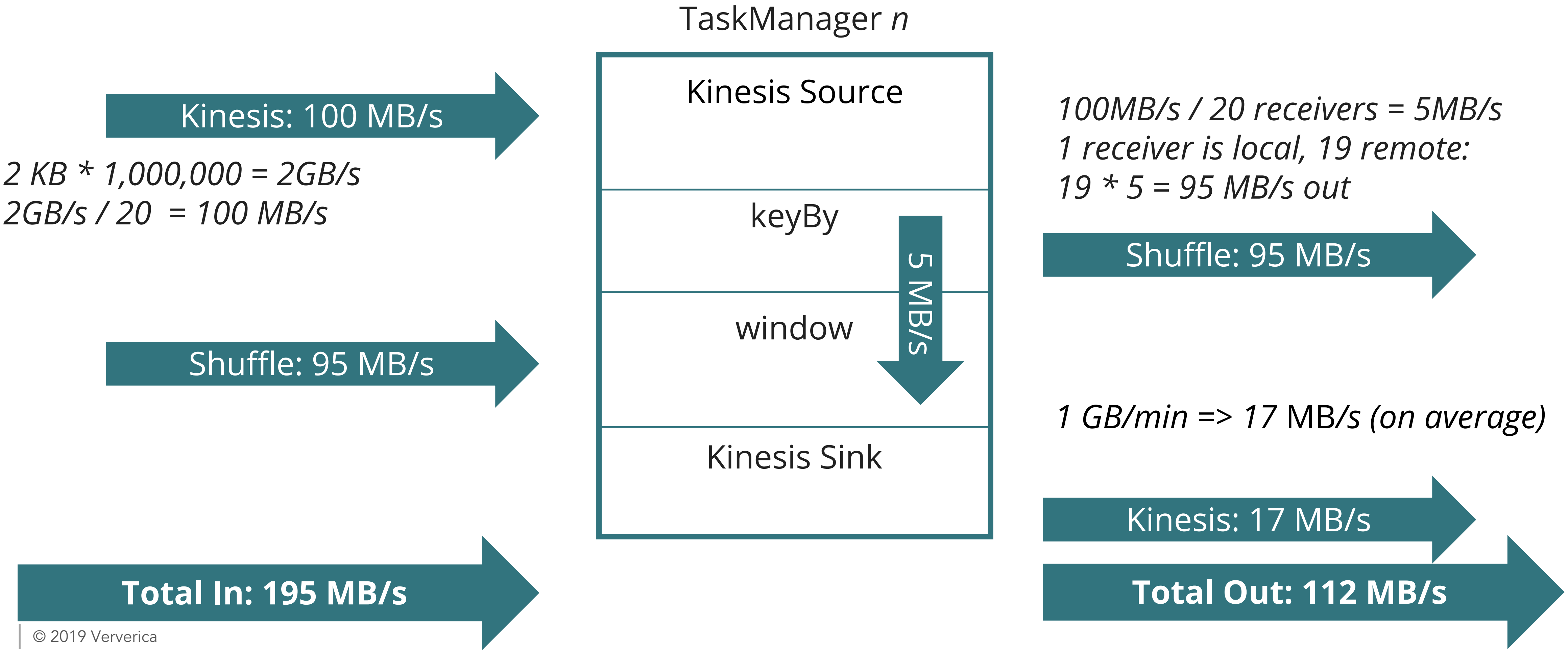
Shuffle: 95 MB/s

1 GB/min => 17 MB/s (on average)

Kinesis: 17 MB/s

Example

A Pod’s Perspective (20 Pods)



TaskManager *n*

Kinesis Source

keyBy

window

Kinesis Sink

5 MB/s

Kinesis: 100 MB/s

*2 KB * 1,000,000 = 2GB/s*
2GB/s / 20 = 100 MB/s

Shuffle: 95 MB/s

Total In: 195 MB/s

100MB/s / 20 receivers = 5MB/s
1 receiver is local, 19 remote:
*19 * 5 = 95 MB/s out*

Shuffle: 95 MB/s

1 GB/min => 17 MB/s (on average)

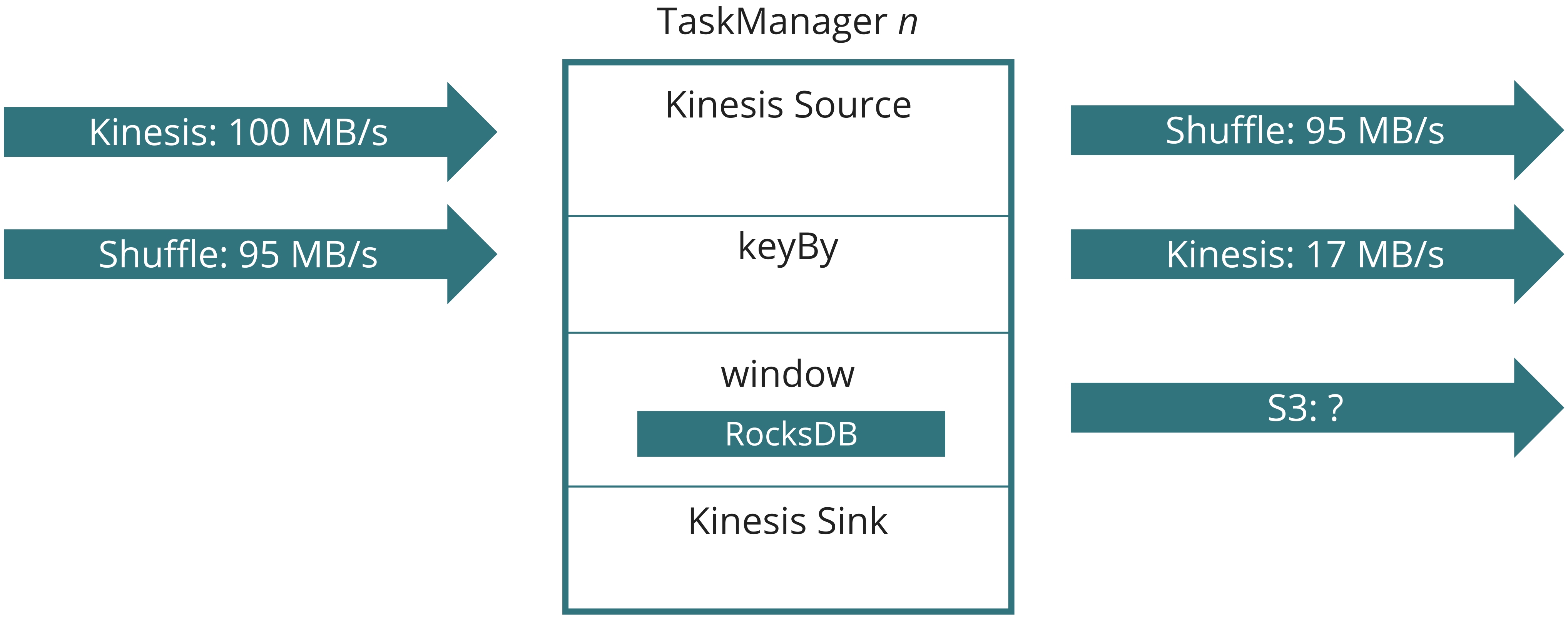
Kinesis: 17 MB/s

Total Out: 112 MB/s

© 2019 Ververica

Example

A Pod's Perspective (20 Pods Overall) - Checkpointing



© 2019 Ververica

Example - Excursion

Window State Checkpoints

How much state are we checkpointing?

Step 1: State per Pod

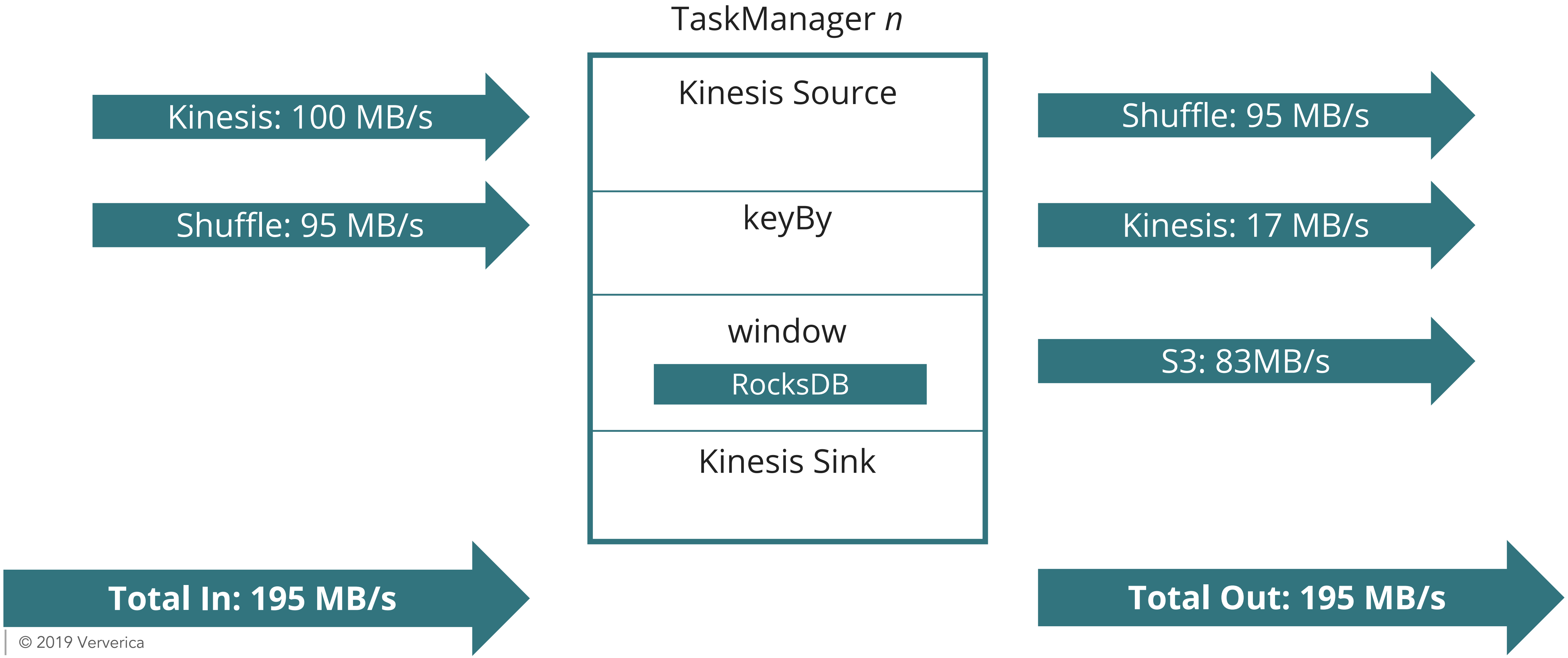
- 40 bytes * 5 windows * 25,000,000 keys = 5 GB

Step 2: Checkpointing Configuration

- Non-Incremental (Full Snapshots, (Space Amplification of RocksDB irrelevant))
- Checkpoint Interval: 1 min
- 5 GB / 60 seconds = 83 MB/s

Example

A Pod’s Perspective (20 Pods Overall)



TaskManager *n*

Kinesis Source

keyBy

window

RocksDB

Kinesis Sink

Kinesis: 100 MB/s

Shuffle: 95 MB/s

Total In: 195 MB/s

Shuffle: 95 MB/s

Kinesis: 17 MB/s

S3: 83MB/s

Total Out: 195 MB/s

Example - Final Result

Possible EKS Setup

- Assume 3 CPUs per Pod -> **2 Pods per instance**
- 10 x m5d.2xlarge
- Instance type m5d.2xlarge [1]
    - 8 CPU
    - 32 GiB RAM
    - 1 x 300 NVMe SSD attached storage
    - ~300MB/s baseline network bandwidth [2]
    - ~600MB/s average network bandwidth [2]

- Network Requirements (as derived):
    - 2x195MB/s=390MB/s (ingoing) continuously
    - 2x107MB/s=214MB/s (outgoing) continuously
    - 83MB/s (outgoing) on average for checkpointing

[1] https://aws.amazon.com/ec2/instance-types/
[2] https://docs.google.com/spreadsheets/d/1N2xQqry-zAKnK6FtW8X5zBYhMiFFnuMySMpx7f3K60s/edit#gid=533991784

Disclaimer

- This was just a “back of the napkin” calculation
- Ignored network factors
  - Protocol overheads (Ethernet, IP, TCP, …)
  - K8s Overlay Network
  - RPC (Flink‘s own RPC, K8s, checkpoint store)
  - Checkpointing causes network bursts
  - A window emission causes bursts
- CPU, memory, **disk access speed** have all been ignored

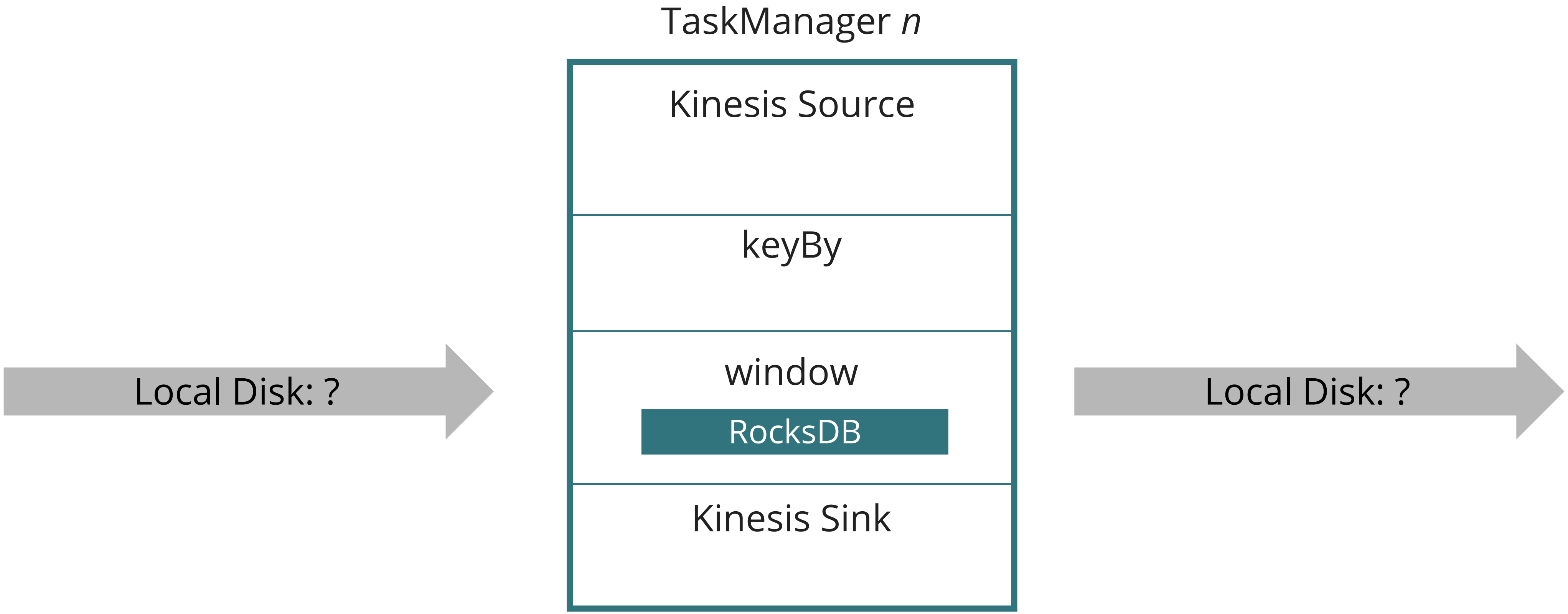
ververica

konstantin@ververica.com www.ververica.com @VervericaData

Backup: Disk Access

Example

A Pod’s Perspective (20 Pods Overall)

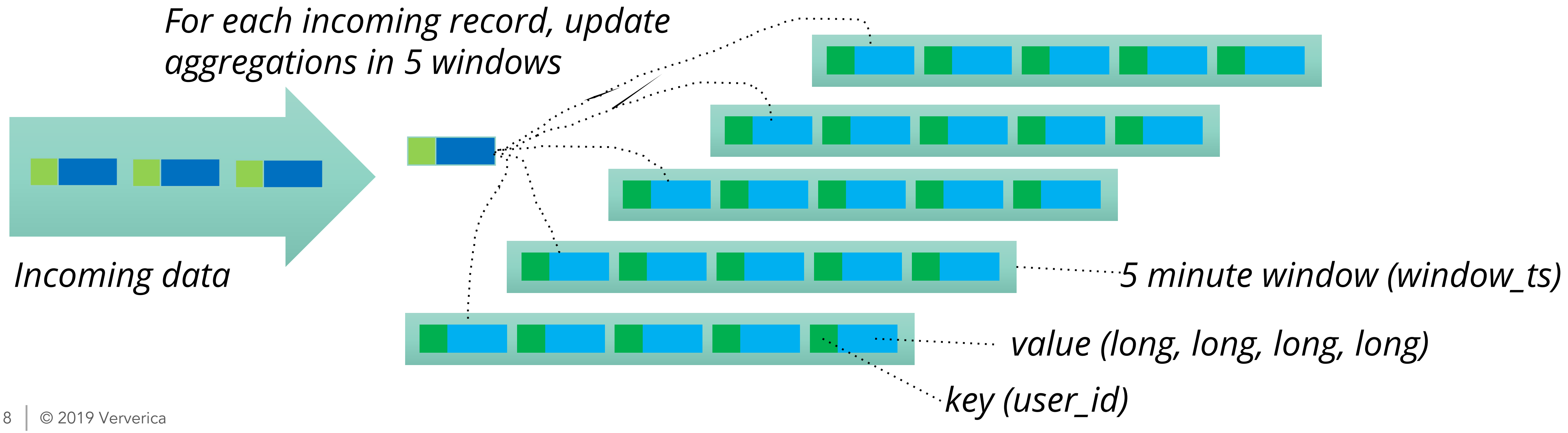


Example - Excursion

Window State Access

How is the Window operator accessing state?

Recap: 1,000,000 msg/sec. Sliding window of 5 minutes, 1 minute slide
Assumption: For each user, we store 2 ints (user_id, window_ts) and 4 longs from the aggregation = 2 * 4 bytes + 4 * 8 bytes = 40 bytes per key


For each incoming record, update aggregations in 5 windows
Incoming data
5 minute window (window_ts)
value (long, long, long, long)
key (user_id)

Example - Excursion

Window State Access

How much state is read/written from/to local RocksDB instance?

Step 1: Updates to RocksDB database

- 40 bytes * 5 windows * 50,000 msg/s = 10 MB/s

Step 2: Incorporating RockDB's disk usage

- write amplification: 15
- read amplification: 7
- Disk Write: 10 MB/s * 13 = 150 MB/s
- Disk Reads: 10 MB/s * (14 (reads during compaction)+7) = 210 MB/s

Aside: RocksDB Write/Read Amplification

Size of Data: 5 GB (see previous slides)

RocksDB Level Structure in Stable State:

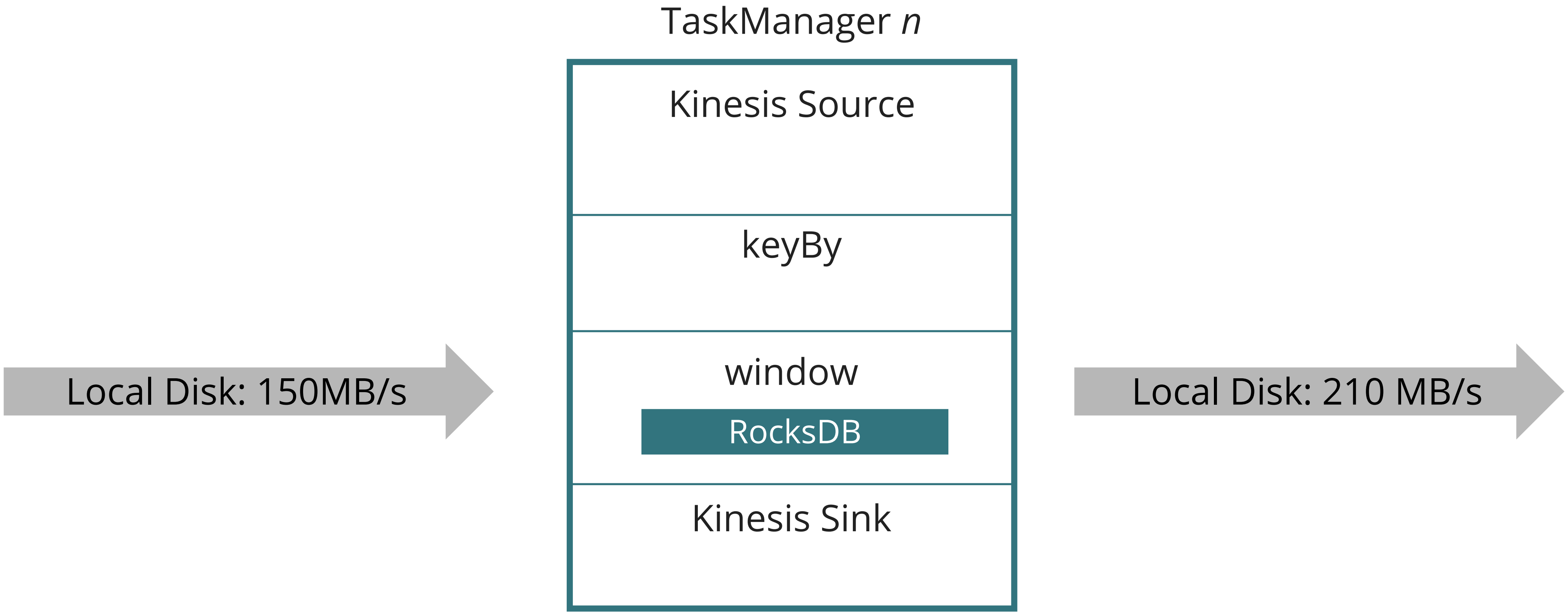
Size of L0: 256 MB
Size of L1: 256 MB
Size of L2: 2.56 GB
Size of L3: 5GB

Write Amplification:
1 (L0) + 2(L0->L1) + 10(L1->L2) + 2(L2->L3) = 15

Read Amplification:
4 (#L0 files) + 3 (#Levels) = 7

Example

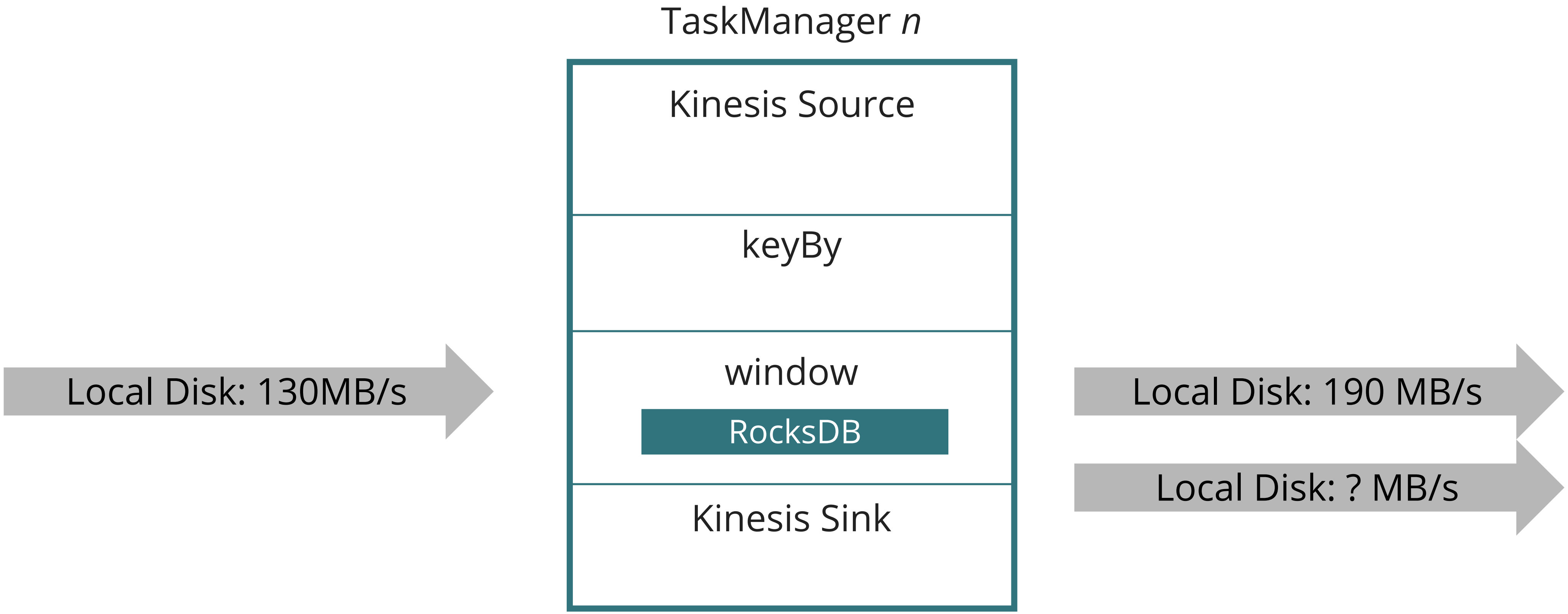
A Pod's Perspective (20 Pods Overall)



© 2019 Ververica

Example

A Pod’s Perspective (20 Pods Overall) - Checkpointing



Example - Excursion

Window State Checkpoints

How much state are we checkpointing?

Step 1: State per Pod

- 40 bytes * 5 windows * 25,000,000 keys = 5 GB

Step 2: Size of RocksDB Instance on Disk

- Database Size * Space Amplification = 5 * 1.6 = 8 GB
- 8GB/min = 125MB/s

Aside: RocksDB Space Amplification

Size of Data: 5 GB (see previous slides)

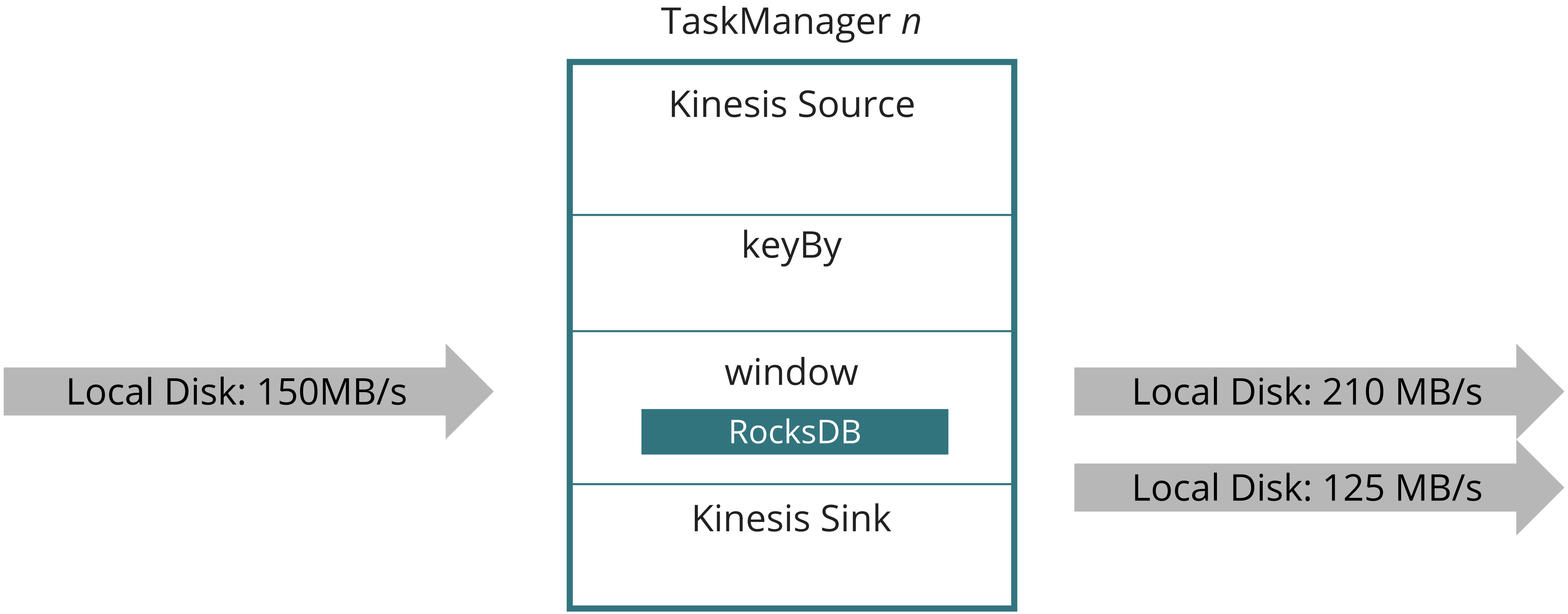
RocksDB Level Structure in Stable State:

Size of L0: 256 MB
Size of L1: 256 MB
Size of L2: 2.56 GB
Size of L3: 5GB

Space Amplification:
(256 MB + 256MB + 2.56GB + 5G) / 5G = 1.6

Example

A Pod's Perspective (20 Pods Overall) - Checkpointing



TaskManager *n*

Kinesis Source

keyBy

window

RocksDB

Kinesis Sink

Local Disk: 150MB/s

Local Disk: 210 MB/s

Local Disk: 125 MB/s

Example - Final Result

Possible EKS Setup

- 10 x m5d.2xlarge
- Instance type m5d.2xlarge [1]
  - 8 CPU
  - 32 GiB RAM
  - 1 x 300 NVMe SSD attached storage

- NVMe SSD
  - Max IOPS: ~1.1M IOPS
  - Sequential Reads: ~6.8 GB/s

- Disk IO Requirements
  - 2* (150 MB/s +210 MB/s +125 MB/s) = 2 * 485 MB/s =~ 1GB/s

[1] https://aws.amazon.com/ec2/instance-types/
[2] https://docs.google.com/spreadsheets/d/1N2xQqry-zAKnK6FtW8X5zBYhMiFFnuMySMpx7f3K60s/edit#gid=533991784