

# From Message to Cluster

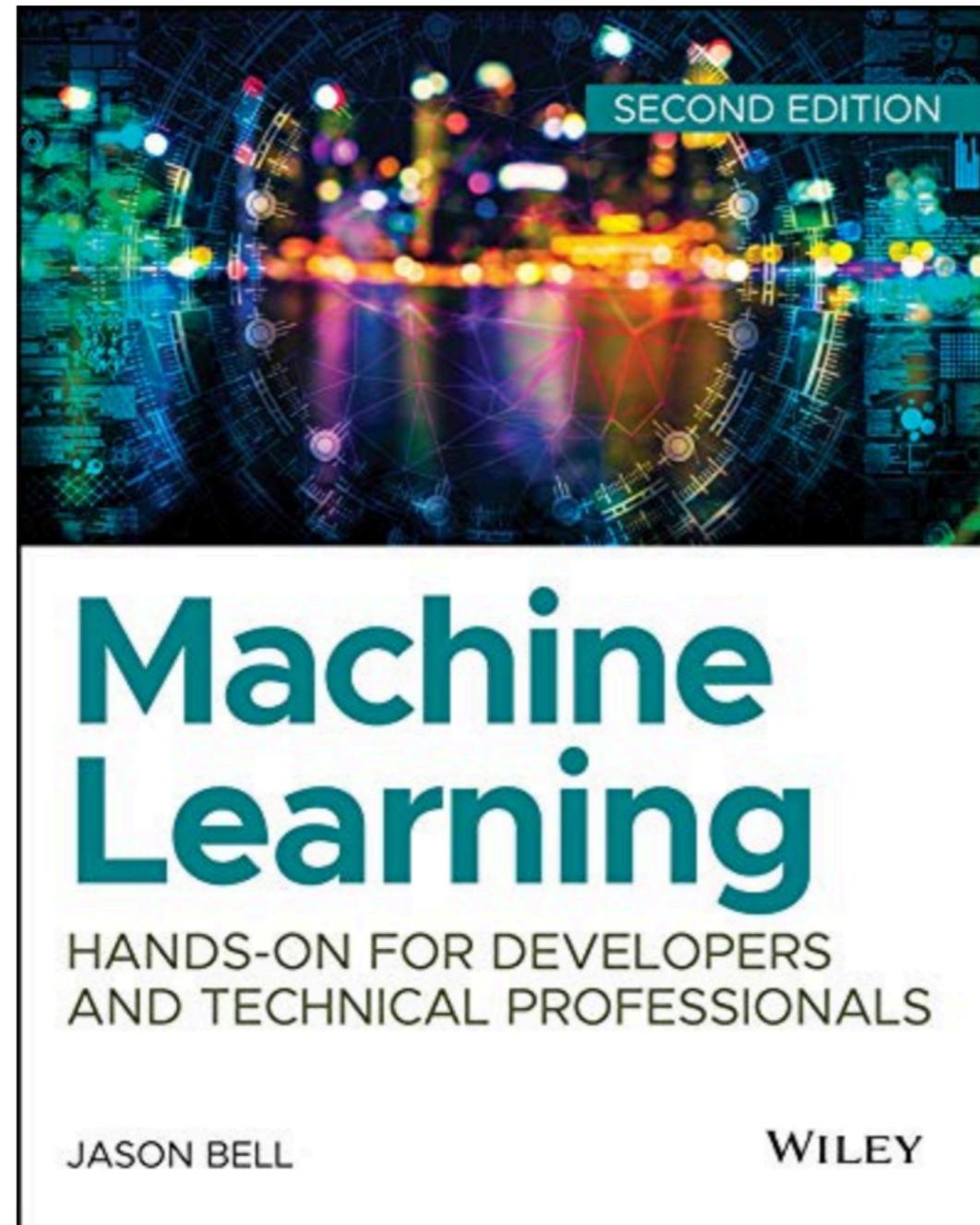
A Realworld Introduction to Kafka Capacity Planning.

**Jason “Jase” Bell - @jasonbelldata**



digitalis.io

**<https://digitalis.io>**



# **MeetupCat is my spirit animal.**





## Flight Mode is ON! You may.....

- **Heckle.**
- **Ask Questions.**
- **Heckle More.**
- **Talk about steak.**
- **Heckle again.**

A screenshot of a social media profile for Jase Bell. The profile picture shows a man with glasses and a beard, smiling, surrounded by guitars. The bio reads: "Kafka Ops @digitalis\_io, author of Machine Learning:Hands on for Devs and Tech Profs. Founder of @DeskHoppa and Chapman Stick player." It includes location information ("Limavady, Northern Ireland, UK"), a website link ("itsallabet.com"), and birth date ("Born October 7, 1971"). The profile also indicates he joined in March 2013 and has 317 following and 1,310 followers.

**Jase Bell**  
@jasonbelldata  
Kafka Ops @digitalis\_io, author of Machine Learning:Hands on for Devs and Tech Profs. Founder of @DeskHoppa and Chapman Stick player.  
Limavady, Northern Ireland, UK [itsallabet.com](#) Born October 7, 1971  
Joined March 2013  
317 Following 1,310 Followers

**What I'm Going To  
Cover**

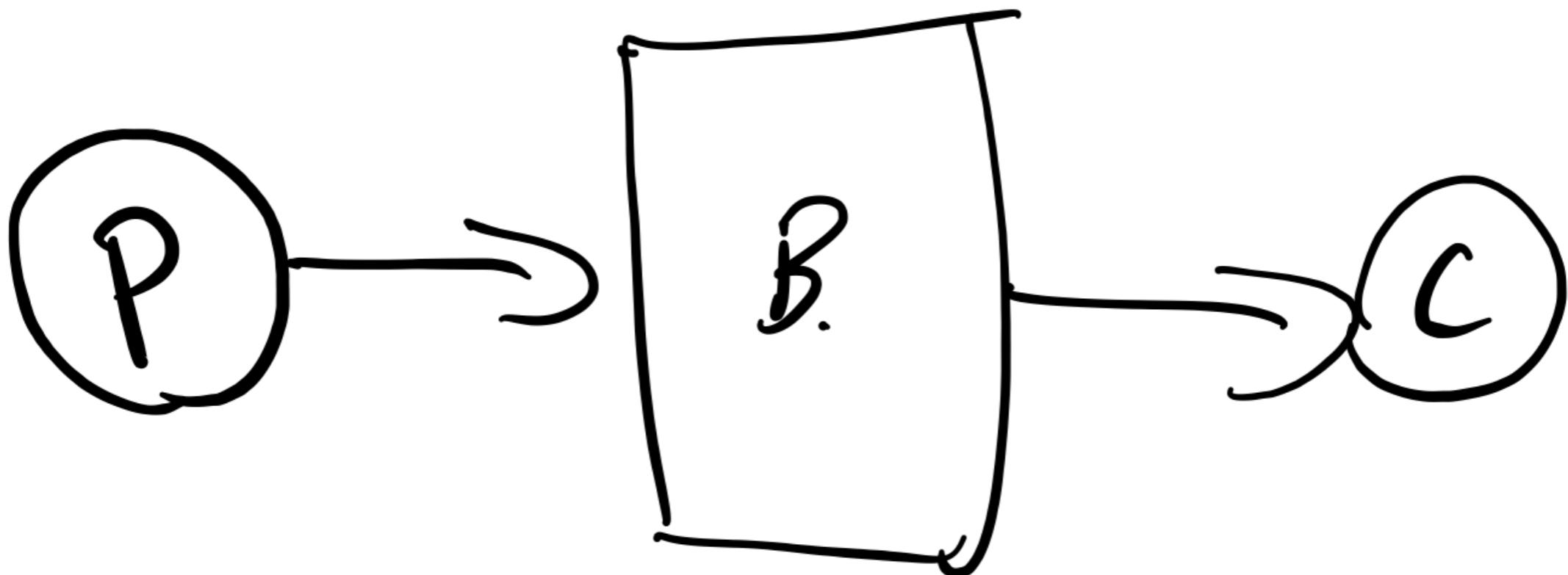
What I'm Going To  
Cover

The Set List.

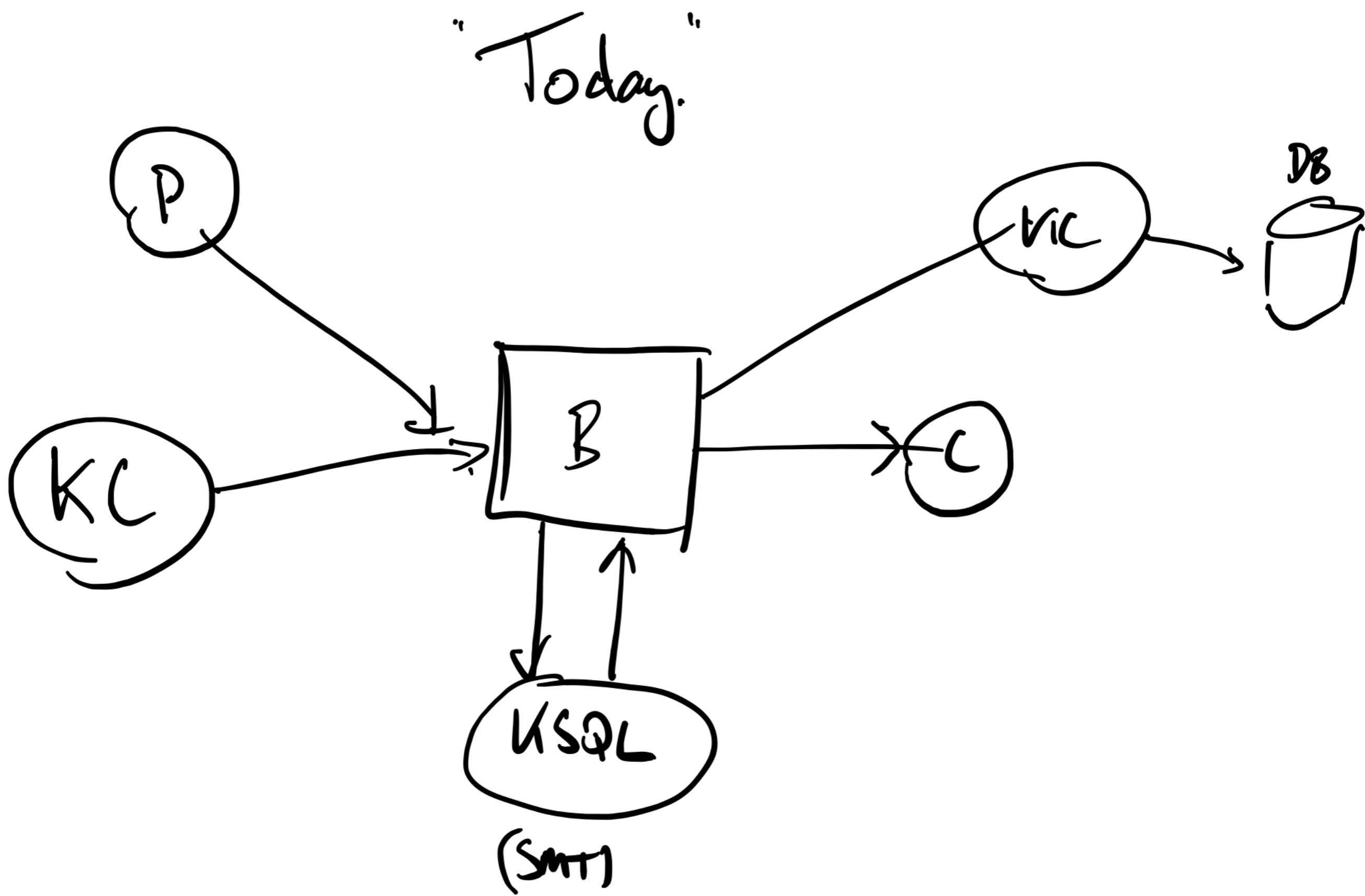
- The Old Days.
- The Now Times.
- The Stuff We Don't Talk About
- The Message
- What I Usually Ask For
- Retention
- Estimated Capacity
- Compression
- Stress Testing
- Network and Disk Throughput
- Topic Partitions
- Kafka Connect
- KSQL
- Replicator
- Parting Thoughts.....
- -----
- Rapturous Applause
- Encore (Probably Eye of the Tiger.....)

# The Old Days

The Old Days

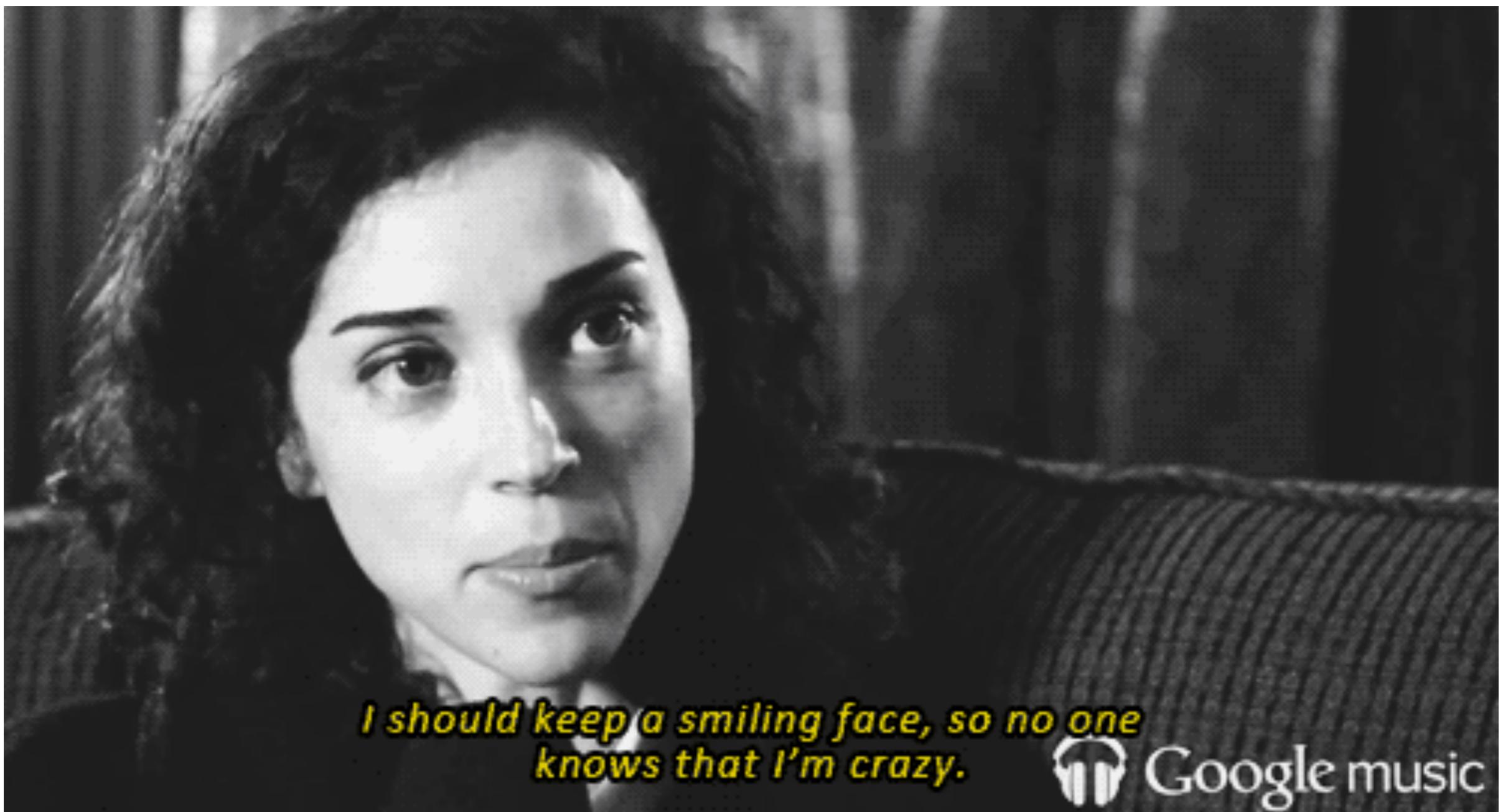


# **The Now Times**



# **The Stuff We Don't Talk About**

We *think* we know what  
we need from our Kafka  
Cluster

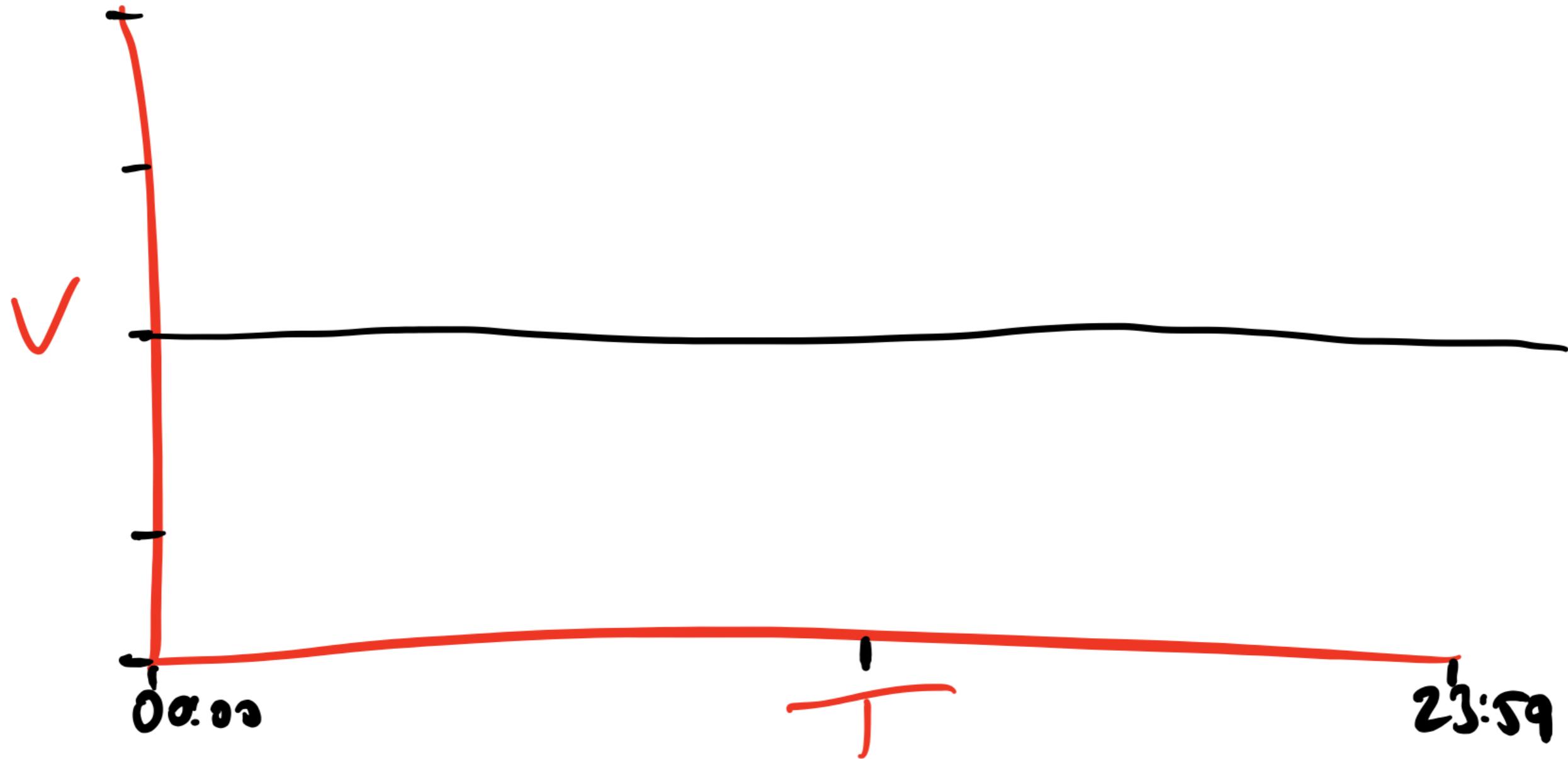


*I should keep a smiling face, so no one  
knows that I'm crazy.*

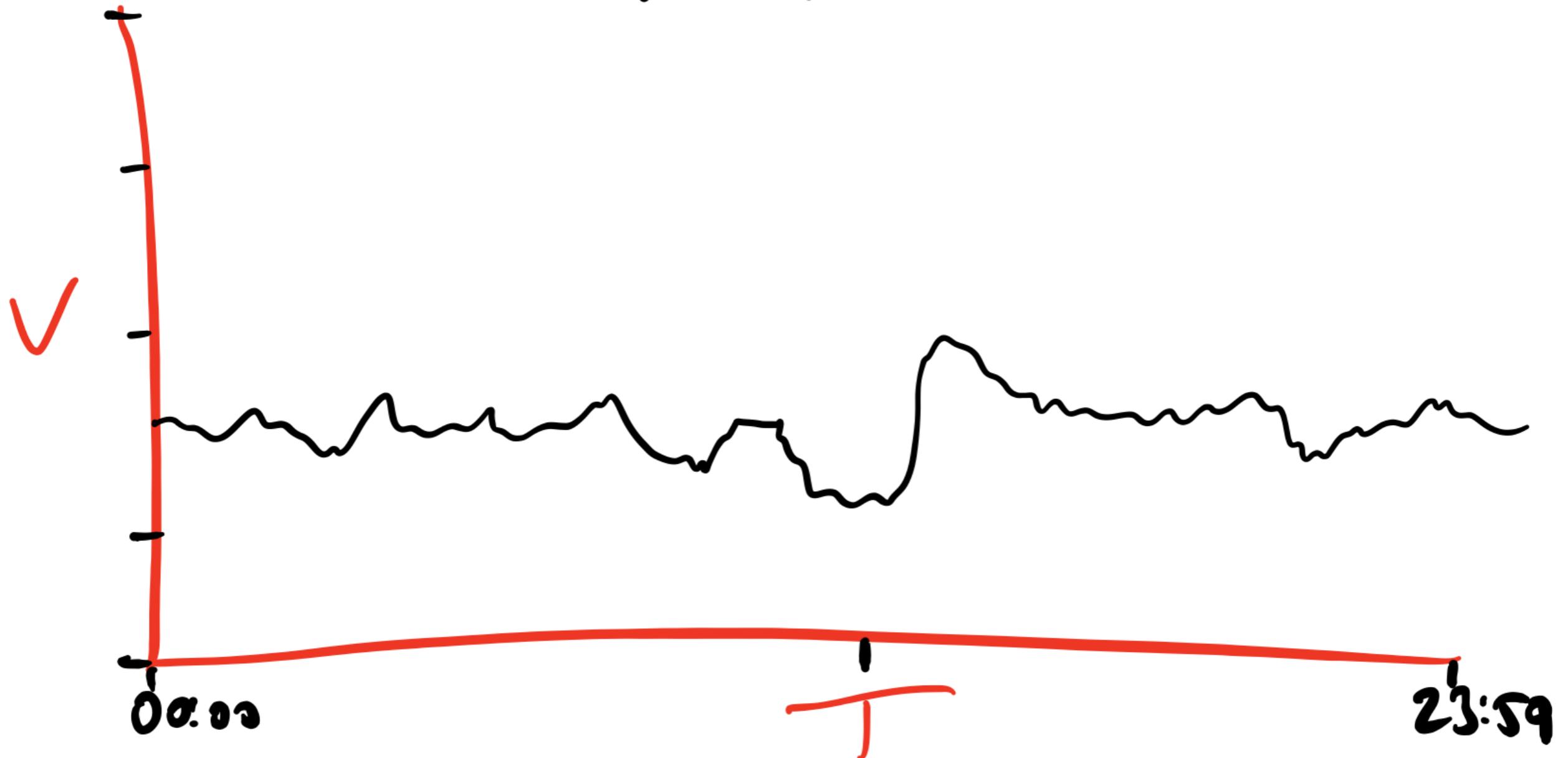


Google music

"On Yeah, Right."



"Realtime"

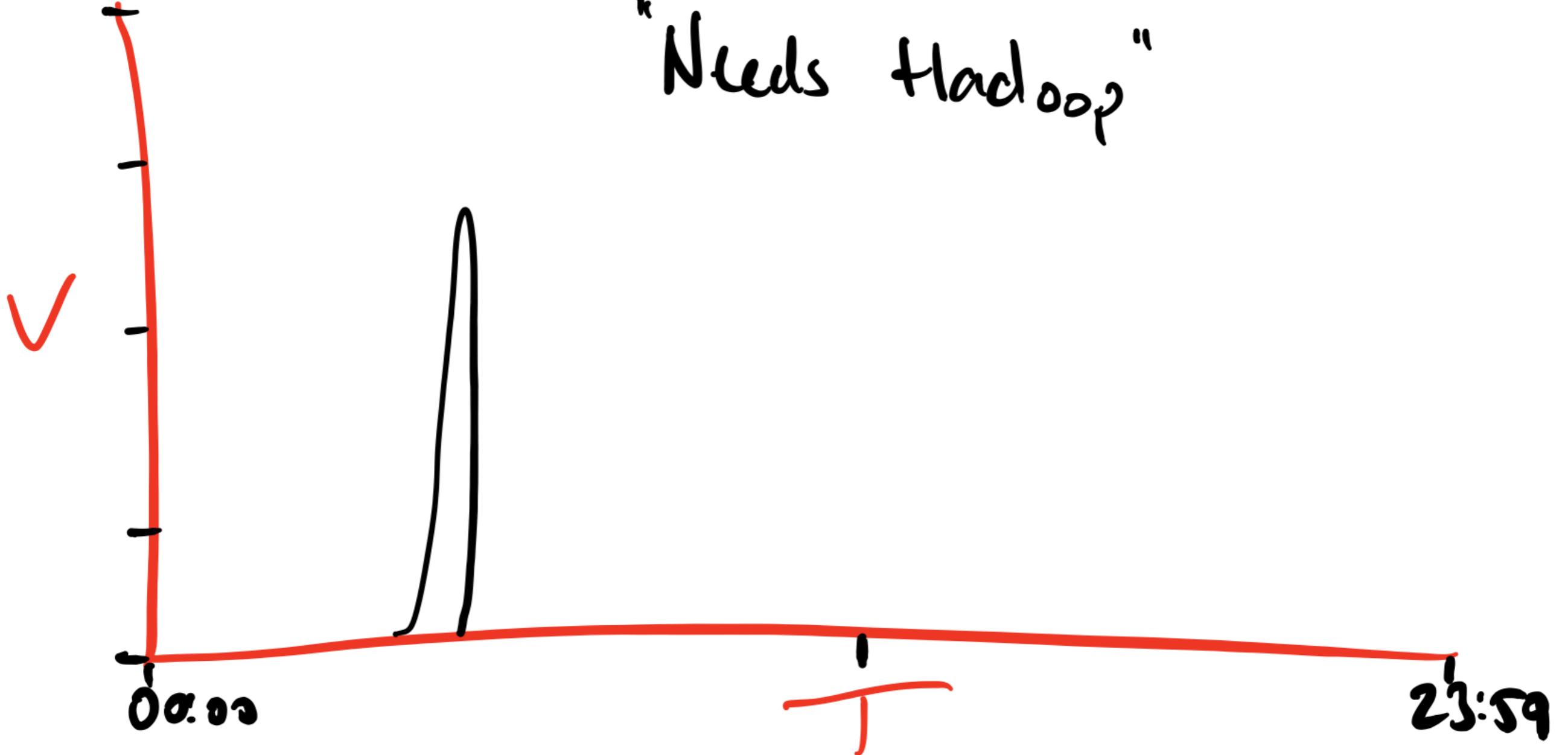


"Retail"



"Batch(?)"







# The Message



```
{  
  "text": "RT @PostGradProblem: In preparation for the NFL lockout, I will be spending twice as much time analyzing my fantasy baseball team during ...",  
  "truncated": true,  
  "in_reply_to_user_id": null,  
  "in_reply_to_status_id": null,  
  "favorited": false,  
  "source": "<a href=\"http://twitter.com/\" rel=\"nofollow\">Twitter for iPhone</a>",  
  "in_reply_to_screen_name": null,  
  "in_reply_to_status_id_str": null,  
  "id_str": "54691802283900928",  
  "entities": {  
    "user_mentions": [  
      {  
        "indices": [  
          3,  
          19  
        ],  
        "screen_name": "PostGradProblem",  
        "id_str": "271572434",  
        "name": "PostGradProblems",  
        "id": 271572434  
      }  
    ],  
    "urls": [ ],  
    "hashtags": [ ]  
  },  
  "contributors": null,  
  "retweeted": false,  
  "in_reply_to_user_id_str": null,  
  "place": null,  
  "retweet_count": 4,  
  "created_at": "Sun Apr 03 23:48:36 +0000 2011",  
  "retweeted_status": {  
    "text": "In preparation for the NFL lockout, I will be spending twice as much time analyzing my fantasy baseball team during company time. #PGP",  
    "truncated": false,  
    "in_reply_to_user_id": null,  
    "in_reply_to_status_id": null,  
    "favorited": false,  
    "source": "<a href=\"http://www.hootsuite.com/\" rel=\"nofollow\">HootSuite</a>",  
    "in_reply_to_screen_name": null,  
    "in_reply_to_status_id_str": null,  
    "id_str": "54640519019642881",  
    "entities": {  
      "user_mentions": [ ],  
      "urls": [ ],  
      "hashtags": [ ]  
    }  
  }  
}
```

# Twitter JSON Payload ~6kb

**What I Usually Ask  
For**

# **What I'll Ask Team For...**

- Average Message Size**

# **What I'll Ask Team For...**

- Average Message Size**
- Estimated Daily Quantity**

# **What I'll Ask Team For...**

- Average Message Size
- Estimated Daily Quantity
- Any Peak Per Hour Quantity

# **What I'll Ask Team For...**

- Average Message Size
- Estimated Daily Quantity
- Any Peak Per Hour Quantity
- Desired Replication Factor

# **What I'll Ask Team For...**

- Average Message Size
- Estimated Daily Quantity
- Any Peak Per Hour Quantity
- Desired Replication Factor
- Desired Partitions

# **What I'll Ask Team For...**

- Average Message Size
- Estimated Daily Quantity
- Any Peak Per Hour Quantity
- Desired Replication Factor
- Desired Partitions
- Minimum In-sync Replicas

# What I'll Ask Team For...

- Average Message Size - (6 KB)
- Estimated Daily Quantity - (10,000,000/d)
- Any Peak Per Hour Quantity - (1,250,000)
- Desired Replication Factor - (4)
- Desired Partitions - (10)
- Minimum In-sync Replicas - (2)

# **Estimated Capacity**

# **Estimated Capacity**

**(Message size x 3) x Daily Qty  
x 1.4 (add 40%)  
= Volume per replicated broker.**

# Estimated Capacity

**(6KB x 3) x 10,0000,000 = 184,320,000 KB  
x 1.4 (add 40%)  
= 258,048,000 KB  
= 248.09 GB**

**Roughly translates to 2.940 MB/sec**

# Estimated Capacity

The x3 gives me a payload size with key, header, timestamp and the value. It's just a rough calculation.

# Estimated Capacity

The x3 gives me a payload size with key, header, timestamp and the value. It's just a rough calculation.

Adding 40% overhead will give you some breathing space when someone does a stress test and doesn't tell you.....

# Retention

**(6KB x 3) x 10,0000,000 = 184,320,000 KB  
x 1.4 (add 40%)  
= 258,048,000 KB  
= 248.09 GB**

**248.09 GB/day x 14 days retention  
= 3.4 TB per broker.**

# **Estimated Capacity**

**df -h**

**Is your friend.....**

# **Estimated Capacity**

**d u -H .**

**Is also your friend.....**

DRINK SENSIBLY  
GROW UP  
USE A STRAW



# **Compression**

**Producer configuration compression.type defaults to “none”.**

**Options are gzip, snappy, lz4 and zstd.**

**Expect ~20%-40% message compression depending on the algorithm used.**

# **Stress Testing**

```
kafka-producer-perf-test --topic TOPIC --record-size SIZE_IN_BYTES
```

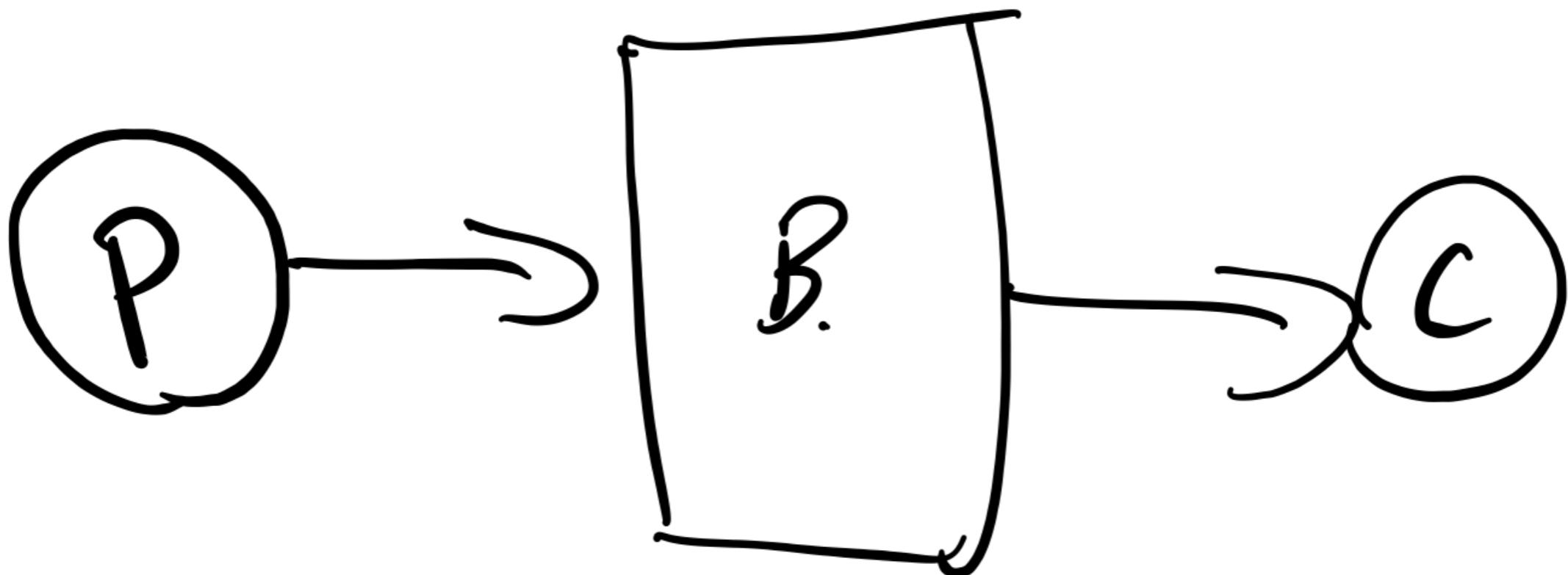
```
$ bin/kafka-producer-perf-test --topic testtopic --record-size 1000 --num-records 10000 --throughput 1000 --producer-props bootstrap.servers=localhost:9092
```

5003 records sent, 1000.4 records/sec (0.95 MB/sec), 1.6 ms avg latency, 182.0 ms max latency.

10000 records sent, 998.801438 records/sec (0.95 MB/sec), 1.12 ms avg latency, 182.00 ms max latency, 1 ms 50th, 2 ms 95th, 19 ms 99th, 23 ms 99.9th.

```
kafka-consumer-perf-test --broker-list host1:port1,host2:port2 --topic  
TOPIC
```

The Old Days



# **Network and Disk Throughput**

- D - Data to be written (**MB/sec**)
- R - **Replication Factor**
- C - **Number of Consumer Groups (readers for each write)**

**The Volume of Writes: (D \* R)**

**Reads happen internally by the replicas, this gives us:**

**The Volume of Reads within Replication:  $((R-1) * D)$**

**Reads happen internally by the replicas, this gives us:**

**The Volume of Reads within Replication:  $((R - 1) * D)$**

**Adding the consumers we end up with:**

**The Volume of Reads within Replication:  $((((R + C) - 1) * D)$**

**We have memory! We have Caching!**

**$M/(D * R)$  = seconds of writes cached.**

**We have memory! We have Caching!**

**$M/(D * R)$  = seconds of writes cached.**

**We have to assume that consumers might drop from the cache, consumers are running slower than expected or even that replicas might restart due to failure, patching or rolling restarts.**

**Lagging Readers  $L = R + C - 1$**

**Disk Throughput:**  $D * R + L * D$

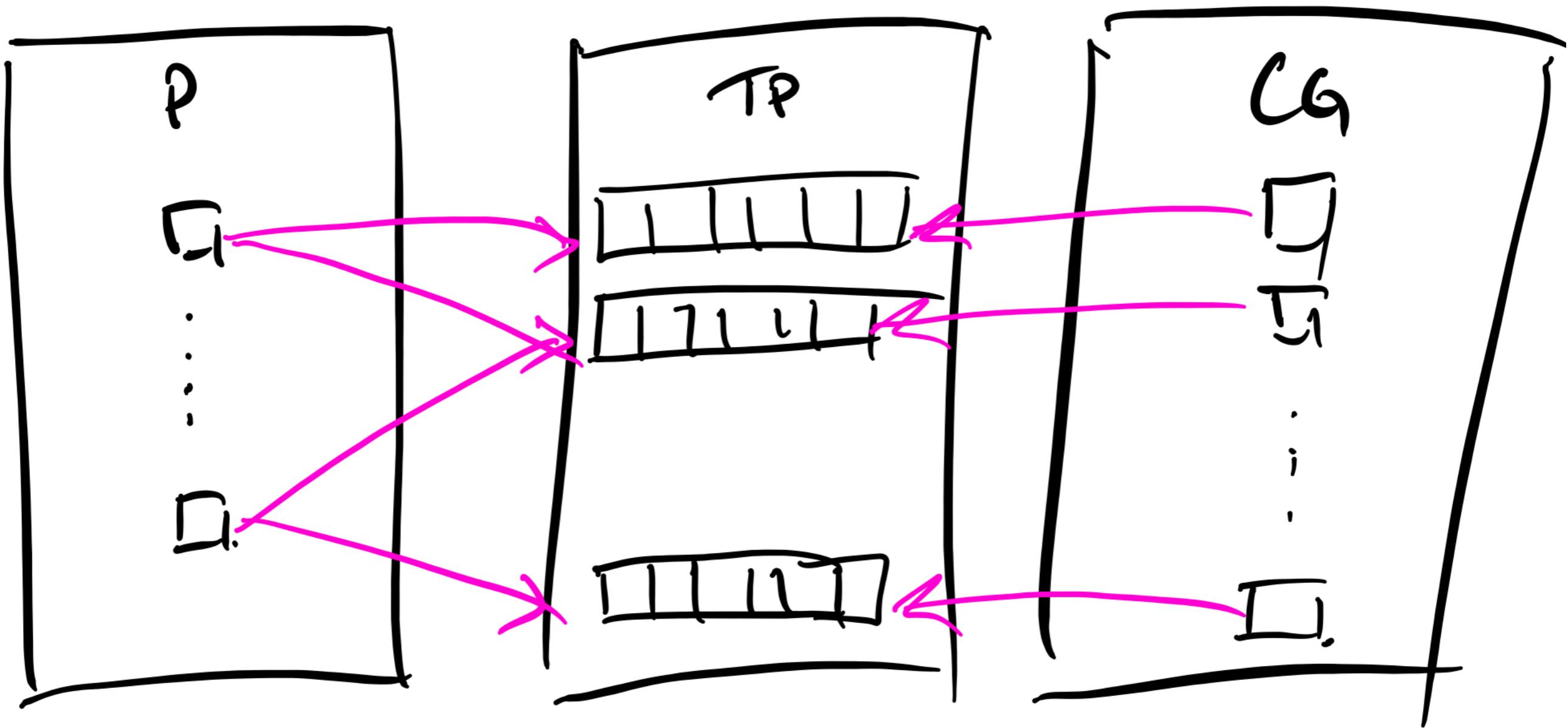
**Network (reads) Throughput:**  $((R + C - 1) * D)$

**Network (writes) Throughput:**  $D * R$

# Topic Partitions

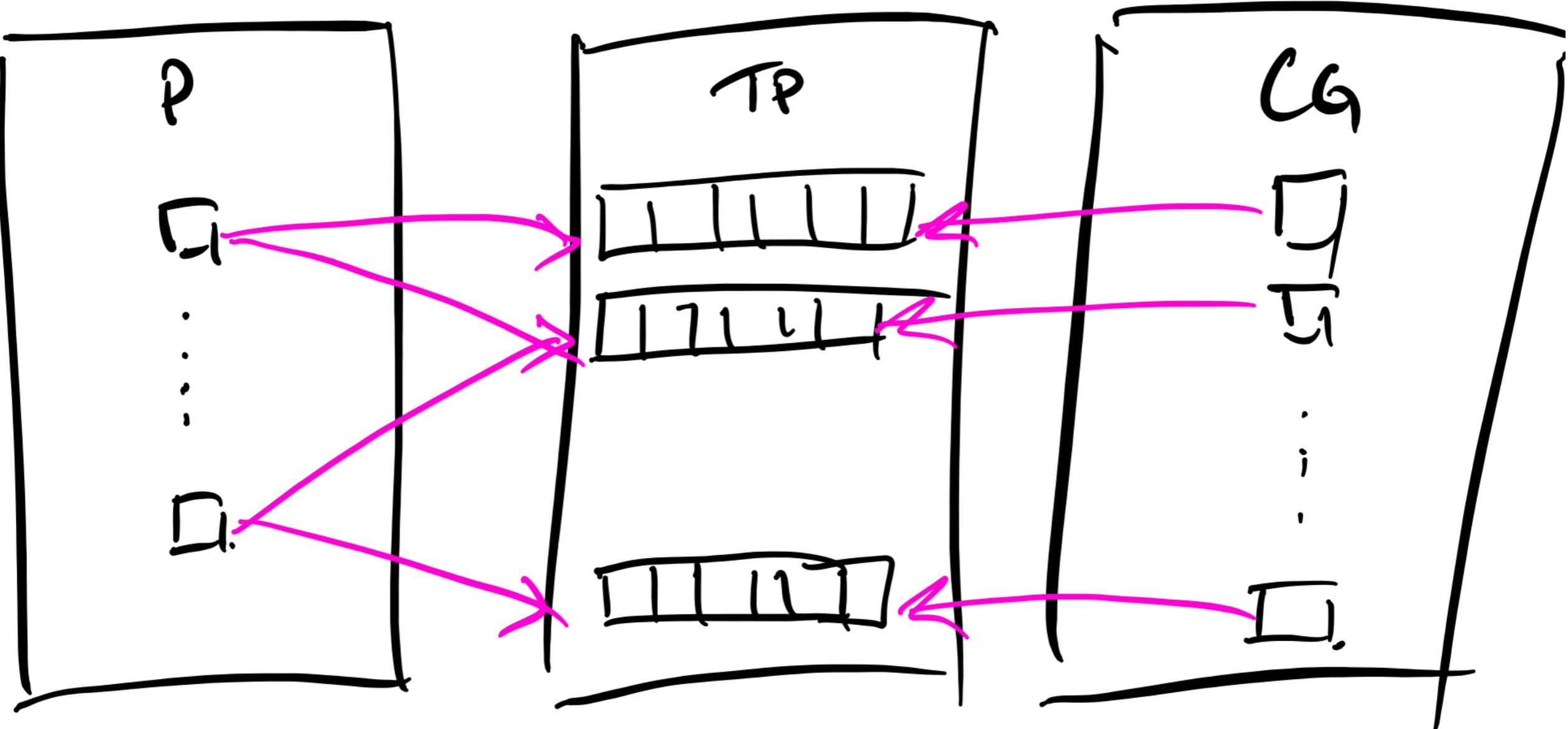
A woman with dark hair tied back is sitting at a table, looking down at a calculator. She is wearing a dark long-sleeved shirt. On the table in front of her is a book with a colorful cover and a calculator. In the background, there is a lamp with a yellow shade and some papers pinned to a wall.

then he got out  
the calculator.



Per Consumer Reads @ 50ms/sec

$$16GB \div 50ms = 20$$



Producer Writes C 100 MB/sec.

$$16 \text{ GB} \div 100 = 10$$

Partitions

Handy Notes

**You can set partitions either creating  
the topic (-partitions n) or afterwards.**

**Having a large number of partitions will have effects on Zookeeper znodes.**

- More network requests
- If leader or broker goes down it may affect startup time as the broker returns to the cluster.

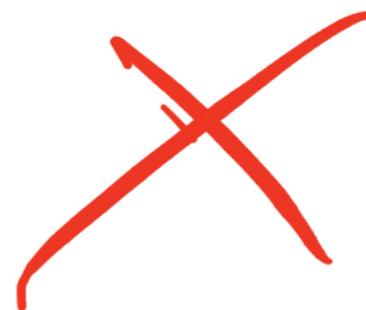
Yes, I know

KIR-SOO!!!!

Partitions ++



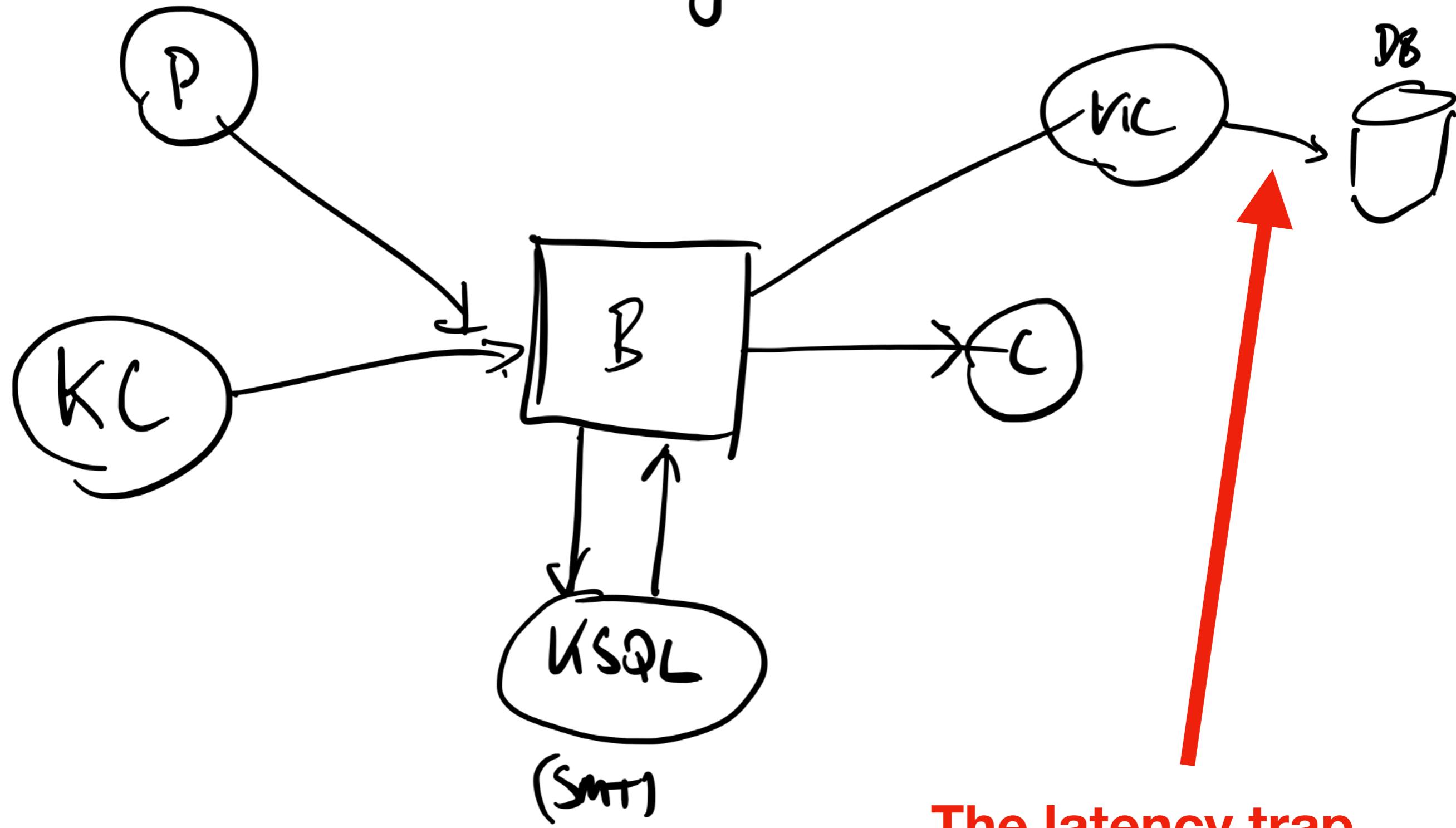
Partitions --



If you need to reduce partitions create a new topic and reduce the partition count.

# Kafka Connect

"Today."



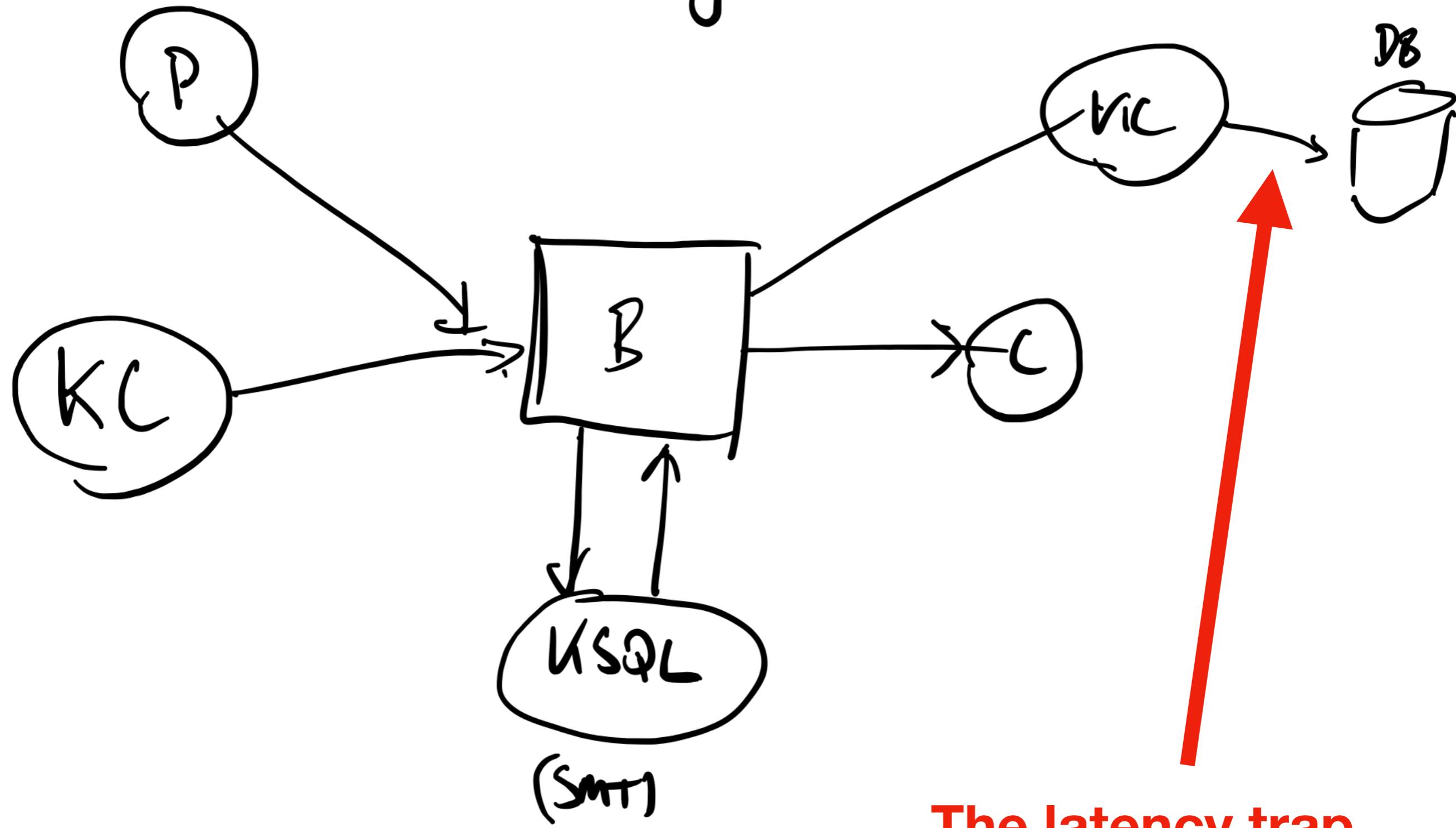
**The latency trap.....**

DLQ's

(More Topics . . . .  
More Consumers . . . .  
More throughput . . . .)

**Think about second and third order  
consequences if a connector would fail.  
What is the impact?**

"Today."



The latency trap.....

# KSQL

# ksqldb

## Baseline Server Requirements

- 4 CPU Cores
- 32GB RAM
- 100GB SSD Disk
- 1Gbit Network



## **Default Outbound Topic Assumptions**

- **Partition Count of 4**
- **Replication Factor of 1**

**(These settings can be modified within your CREATE query)**



## Default Outbound Topic Assumptions

**Some queries will require repartitioning and intermediate topics for certain operations, taking all available records.**



## **Default Outbound Topic Assumptions**

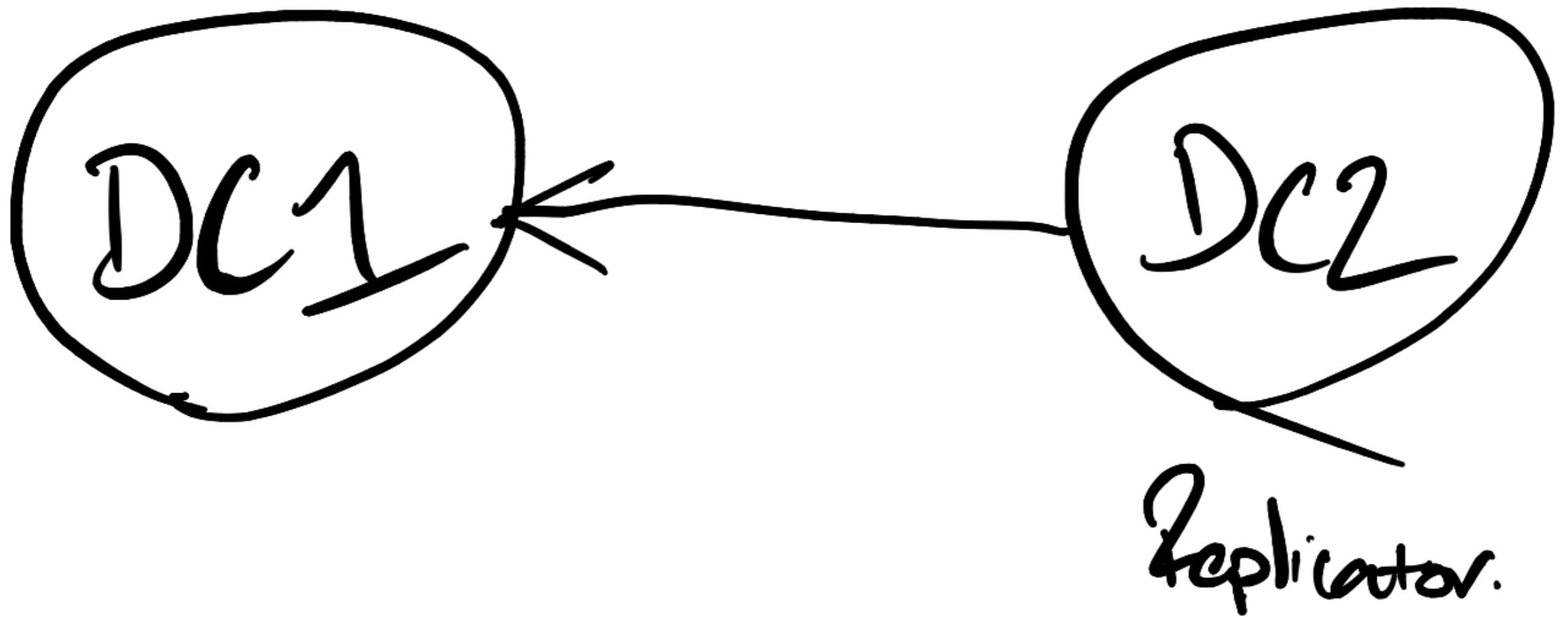
**Processing Small Message/Many Columns  
= CPU Saturation**



## **Default Outbound Topic Assumptions**

**Processing Large Message/Small Columns  
= Network Saturation**

# **Replicator**



**Data Centre to Data Centre is going to lead to increased network latency.**

On producers and consumers, use `send.buffer.bytes` and `receive.buffer.bytes`.

On brokers, use `socket.send.buffer.bytes` and `socket.receive.buffer.bytes`.

No more than:

150 partitions per task

20 tasks per worker.

Too many tasks?

Add more nodes!

# **Parting Thoughts**

# 1

**Consumer Group Lag Reports are your guiding light.  
(If you have Rundeck setup a scheduled job to email  
you the log output)**

```
kafka-consumer-groups --bootstrap-server BROKER_ADDRESS --describe  
--group CONSUMER_GROUP --new-consumer
```

# 2

**Kafka is about trade offs, from the producer right the way through to the consumer (and beyond).**

**There's no right or wrong answer, just experimentation, monitoring and learning.**

# 3

**While securing Kafka is important there is also a cost as certificates are verified and take up CPU resources.**

**Your throughput will be affected.**

# 4

**The Kafka Ecosystem has increased in features over the last few years. This has lead to increased topic and disk space usages that need to be factored in to capacity planning calculations.**

Pfad



Topic<sup>①</sup>



KSQL

— T<sub>1</sub> — T<sub>2</sub> — T<sub>3</sub> —

T<sub>4</sub>

Connect

② ⑤ ⑥

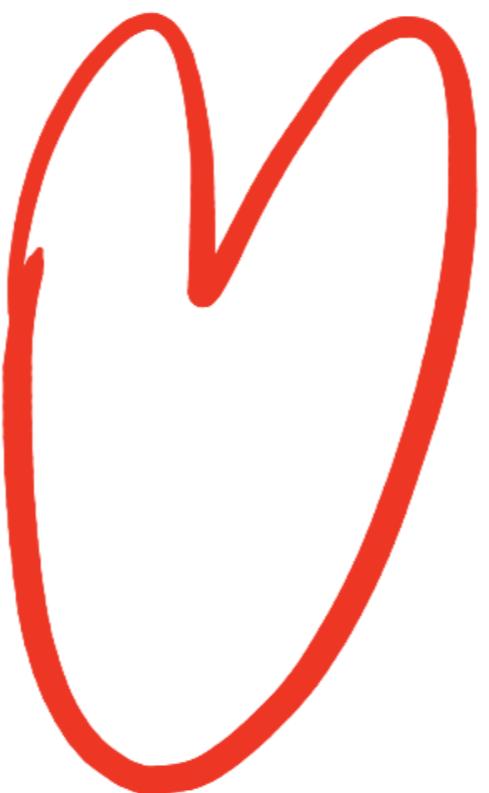
Consumer

One Topic Becomes

Six.

"Can you create me a topic please?"

Thank You !



**Thank you.**

**Many thanks to Shay and David for organising, everyone who attended and sent kind wishes. Lastly, a huge thank you to MeetupCat.**



**Photo supplied by @jbfletch\_**