# 12.05.2025 Week 12 exercises: Stream processing

**Exercise 1:**

(a) What is a Publish/Subscribe (pub/sub) system?

Pub/Sub systems carry out a communication paradigm that decouples producers and consumers of data items in terms of time, space and synchronization.

(b) Explain its basic functioning.

Subscribers submit their interest in receiving certain kinds of events (a subscription). Publishers generate such events and publish them. In between the two, a mediator (or broker) combines them and routes publications to matching subscribers, which finally consume the data.

(c) What are the main advantages of pub/sub systems in comparison to direct messaging?

They do not require the endpoints of communication channels to be online at the same time and can quickly adapt to dynamic environments.

**Exercise 2:**

Concerning Kafka, answer the following questions:

(a) List and briefly explain the main roles in Kafka.

- Topic: a stream of messages belonging to the same type.
- Producer: can publish messages to a topic.
- Consumer: subscribes to topics and pulls data from brokers.
- Brokers: a set of servers where publications are stored.

(b) What are topics and partitions? What ordering guarantees can a Kafka user expect?

Topics are logical collections of partitions that are ordered, append-only, and immutable. Messages sent by a producer to a particular topic partition will be appended in the order they are sent, so that Consumers see messages in this same order. Partitions of a topic may be stored across different brokers, but the order is only guaranteed within a single partition.

(c) What is the purpose of the offset?

Ordered messages within partitions are assigned a monotonically increasing number that is called offset. Its purpose is to uniquely identify every message within the partition.

(d) What is the role of ZooKeeper?

Kafka uses Zookeeper to detect the addition and removal of brokers and consumers. Additionally, it is used to keep track of the consumed offset in each partition.

(e) What are the responsibilities of Leaders and Followers?

Every partition in Kafka has one server which plays the role of a Leader, and zero or more servers that act as Followers. The Leader performs all reads and writes for the partition. Followers, in turn, passively replicate the Leader for fault tolerance. In the event of a failing Leader, one of the Followers will assume the role of Leader.

**Exercise 3:**

(a) What is is Windowing?

Window is a buffer associated with a stream input that groups tuples for processing.

(b) Give a few examples on how to determine windows.

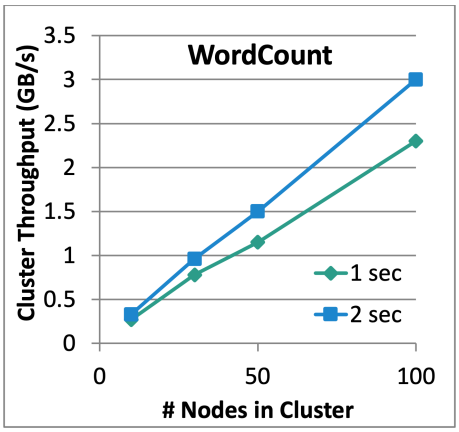
- Tumbling window: fixed length, *e.g.*, all events that happen in a timespan of 10 minutes are grouped in the same window. After that, a new non-overlapping window is created to group the events in the next period.
- Hopping window: like tumbling window, but allows a fixed length overlap (the hop size).
- Sliding window: groups events within an interval that continually moves (*i.e.*, slides), *e.g.*, all events in the last 10 minutes.
- Session window: groups together events that occur closely in time with no fixed duration, *e.g.*, the clicks of a user in some Web page during a session.

**Exercise 4:**

What is the difference between task and data parallelism?

Data parallelism applies the same computation to different partitions of data. Task parallelism applies distinct computation to partitions of data (possibly the same data).

**Exercise 5:**




The graph above shows the cluster throughput as a function of nodes in the cluster for a streaming word count example. Explain the impact of window size on the system latency. Explain why we achieve less throughput with a smaller window size.

- Increasing the window duration, increases the end-to-end latency.
- Smaller window size incurs more communication and system overheads. There are throughput penalties, but latency benefits.

**Exercise 6:**

True or False?

( F ) In ~~Topic-~~ Content-based pub/sub systems, a subscriber can specify filters on key/value attribute pairs of events.

( F ) Kafka ~~Producers~~ Consumers manifest their interest in some topics and pull the corresponding data from brokers.

( T ) Kafka users need to check for duplicates, as the system only guarantees at-least-once delivery.

( F ) ~~Continuous processing-based~~ Micro-batch systems collect data in small groups and take action on each of them.

( T ) A processing element (PE) operates on input tuples by applying a function on them and outputting other tuples.

( T ) Pipelined parallelism consists of sequential stages that concurrently execute a computation on distinct data items.