ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΕΛΕΤΗ ΚΑΙ ΣΥΓΚΡΙΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΕΞΟΡΥΞΗΣ ΓΝΩΣΗΣ ΑΠΟ ΔΕΔΟΜΕΝΑ ΡΟΗΣ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ – Η ΠΕΡΙΠΤΩΣΗ ΤΗΣ STREAMING SQL

Γράβας Χριστόφορος

Streams

Τι είναι τα streams;

- Άπειρες ακολουθίες δεδομένων που διατίθενται στη ροή του χρόνου.
- Δυνατότητα αλληλεπίδρασης σε αυτά για τη διεξαγωγή συμπερασμάτων.

Data

Δυο ειδών δεδομένα

Πεπερασμένου μεγέθους (Bounded Data)

Απεριόριστου μεγέθους (Unbounded Data)

Id	Name	Birthday	Address	Time
18	mr. vke	23/12/2037	Street. IFH 91	21:37:00
28	mr. paj	23/12/2037	Street. TWX 2	06:05:45
37	mr. zpf	23/12/2037	Street. LAE 1	01:04:13
59	mr. xse	23/12/2037	Street. WOZ 15	06:14:59
91	mr. ptc	23/12/2037	Street. YKE 20	01:24:44
110	mr. kpp	23/12/2037	Street. SXU 51	19:28:02
115	mr. klq	23/12/2037	Street. DNH 4	14:01:11
117	mr. qhf	23/12/2037	Street. NLZ 79	22:36:17
119	mr. oes	23/12/2037	Street. KUZ 62	02:44:50
135	mr. tkt	23/12/2037	Street. XRE 47	06:43:53
Id	Name	12/2037	Address	Time

Showing 1 to 10 of 245 entries (filtered from 5,000 total entries) ← Previous

Πίνακες

Ολιστική προβολή συνόλου δεδομένων που έχουν προκύψει από χρονικά στιγμιότυπα τα οποία χειριζόμαστε με τη βοήθεια της SQL.

Ροές δεδομένων

Προβολή στοιχείων δεδομένων που εξελίσσονται στην πάροδο του χρόνου.

Tables are data at rest.
Streams are data in motion.

Streams

1. Δομημένα (bounded data)

Στοιχεία τα οποία ακολουθούν μια συγκεκριμένη μορφή η οποία μας επιτρέπει περεταίρω μοντελοποίηση.

2. Μη δομημένα (unbounded data)

αυθαίρετο περιεχόμενο που προκύπτει συχνά από το συνδυασμό ροών από πολλαπλές πηγές

Timestamps

- Χρονική στιγμή δημιουργίας του stream
- Εγκυρότητα του περιεχομένου
- Διαθεσιμότητα επεξεργασίας

Κατηγορίες δομημένων streams

1. **Περιστροφικό μοντέλο (turnstile model)**
   - Το πιο γενικό μοντέλο.
   - Αναπαριστά διάνυσμα από στοιχεία όπου το κάθε στοιχείο S_i στη ροή είναι μία αύξηση ή μείωση σε ένα στοιχείο του υποκείμενου διανύσματος .
   - Το μέγεθος του διανύσματος αποτελεί το domain του stream.
   - Χρήση σε database systems στα οποία εφαρμόζεται το CRUD*.

*CRUD: *Create, Read, Update, Delete*

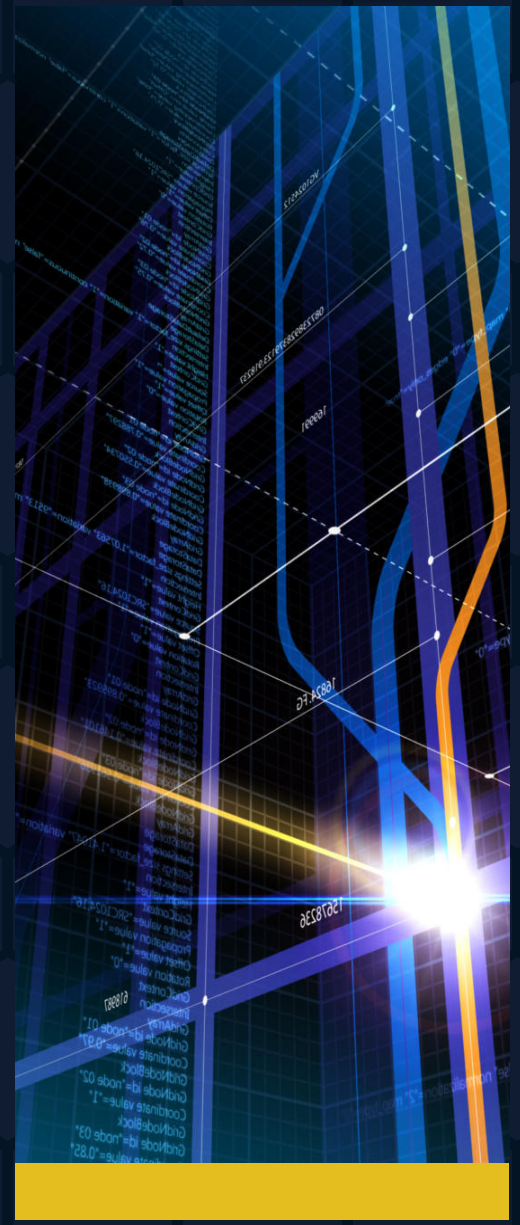
Κατηγορίες δομημένων streams

2. Cush register

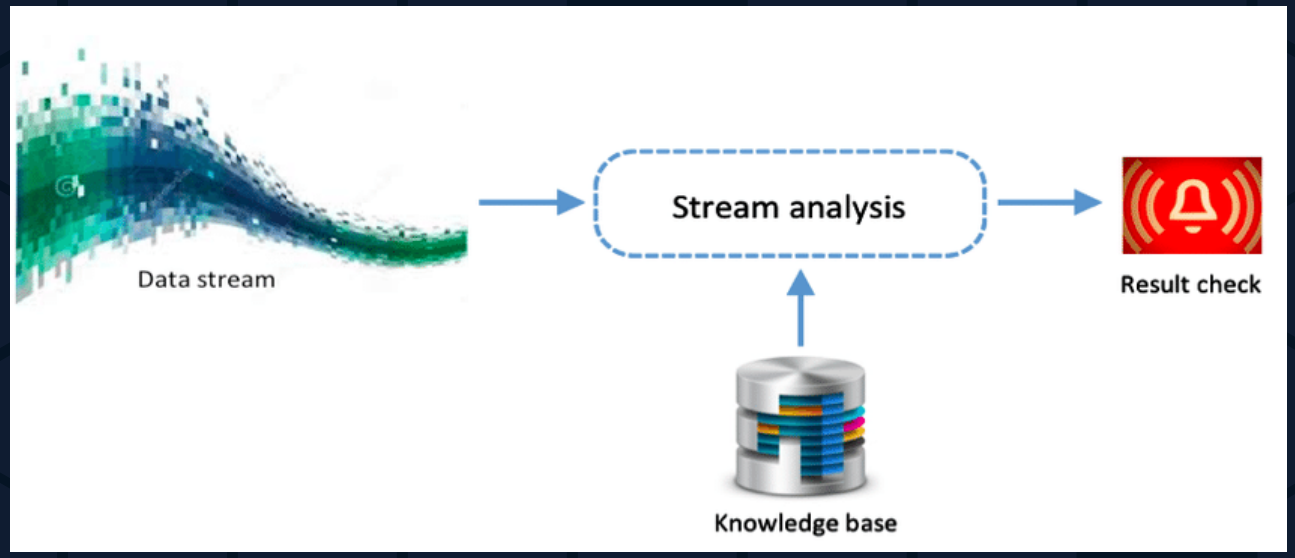
- Τα στοιχεία της ροής δεδομένων είναι προσθήκες στο υποκείμενο διάνυσμα αλλά τα στοιχεία δεν φεύγουν ποτέ από αυτό.
- Πανομοιότυπη λειτουργία με την καταγραφή ιστορικού σχέσεων βάσεων δεδομένων.

3. Timeline model

- Συμπεριφέρεται σε κάθε στοιχείο S_i σαν να είναι ένα καινούριο και ανεξάρτητο διάνυσμα εισαγωγής τιμών.
- Αυξάνεται συνεχώς χωρίς κανένα περιορισμό.
- Εφαρμόζεται σε μηχανές επεξεργασίας ροής δεδομένων λόγω του ότι κάθε στοιχείο μπορεί να επεξεργαστεί ως μια ξεχωριστή οντότητα.



Streaming Processing



Τι είναι;

- Επεξεργασία δεδομένων ροής σε μεγάλους όγκους δεδομένων.
- Υποβολή ερωτημάτων .
- Ανίχνευση συνθηκών σε μικρό χρόνο.

Streaming Processing

Προκλήσεις και λύσεις

- Επεξεργασία τεράστιων ποσοτήτων συμβάντων ροής.
- Ανταπόκριση σε πραγματικό χρόνο στις μεταβαλλόμενες συνθήκες της αγοράς.
- Απόδοση και επεκτασιμότητα καθώς οι όγκοι δεδομένων αυξάνουν σε μέγεθος και πολυπλοκότητα.
- Ταχεία ενοποίηση με υπάρχουσες υποδομές και πηγές δεδομένων.

Streaming Processing

Προκλήσεις και λύσεις

- Μεγαλύτερη παραγωγικότητα προγραμματιστών σε όλα τα στάδια του κύκλου ζωής ανάπτυξης εφαρμογών προσφέροντας καλή υποστήριξη εργαλείων και ευέλικτη ανάπτυξη.
- Ανάλυση και παρακολούθηση δεδομένων ροής.
- Συνεχής επεξεργασία ερωτημάτων.
- Οπτικοποίηση δεδομένων.

Streaming Processing

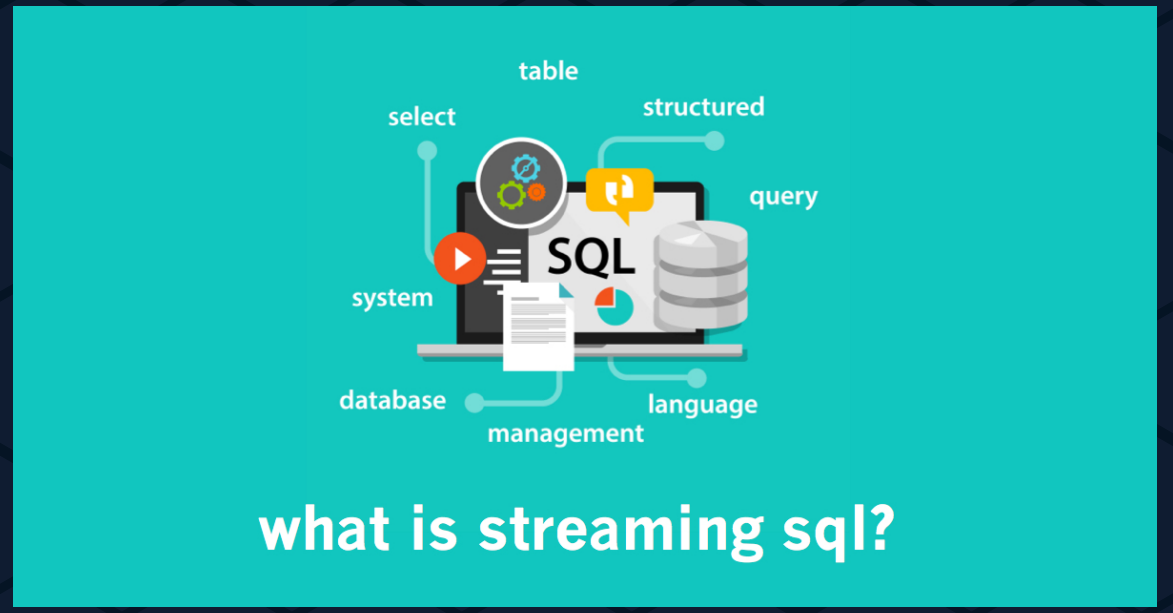
Επιπλέον λόγοι χρήσης

1. Ατέρμονες ροές δεδομένων (Never-ending streams).
2. Εξοικονόμηση υπολογιστικών πόρων.
3. Λιγότερη αναγκαιότητα αποθήκευσης δεδομένων.
4. Εφαρμογή σε μεγάλη γκάμα δεδομένων.

SQL και Stream SQL

Stream SQL

- Γλώσσα που εφαρμόζει ερωτήματα και επεκτείνει την SQL με δυνατότητα επεξεργασίας δεδομένων ροής σε πραγματικό χρόνο.
- Προσθέτει τη δυνατότητα χειρισμού ροών δεδομένων, οι οποίες αποτελούν ατέρμονες πλειάδες δεδομένων που δεν είναι όλες διαθέσιμες το ίδιο χρονικό διάστημα.



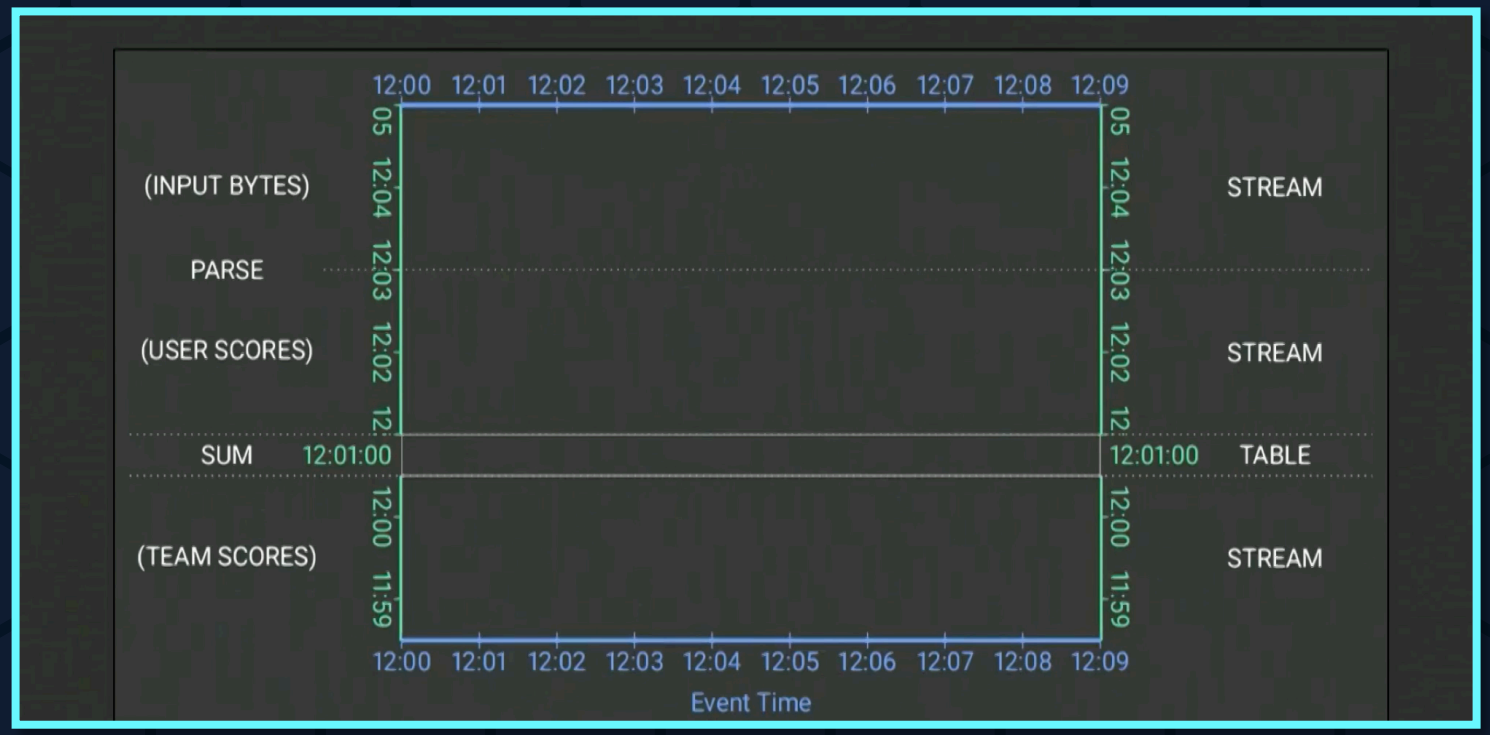
Μερικές Εντολές Stream SQL

Select

Join

Where

Group By



Stream SQL

```
12:01> SELECT STREAM Name,
            SUM(Score) as Total,
             MAX(Time) as Time
             Sys.Undo as Undo
             FROM UserScores GROUP BY Name;
```

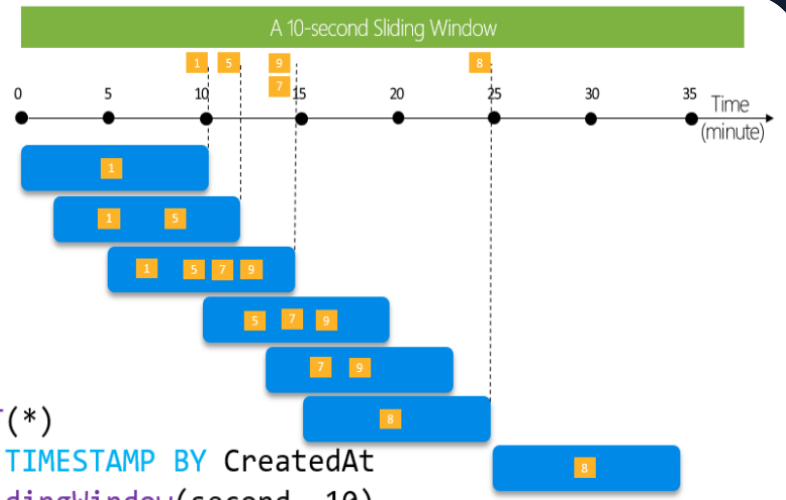
Name	Total	Time	Undo
Yannis	7	12:01	
Paul	3	12:03	
Yannis	7	12:03	undo
Yannis	8	12:03	
Yannis	8	12:07	undo
Yannis	12	12:07	

..... [12:01, 12:07] .....

Stream SQL

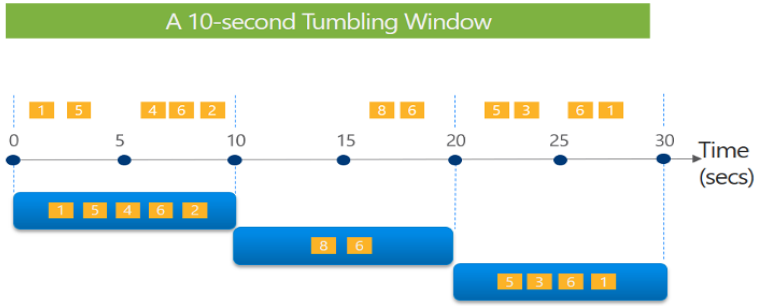
Χαρακτηριστικά Stream SQL

Give me the count of tweets for all topics which are tweeted more than 10 times in the last 10 seconds




```
SELECT Topic, COUNT(*)
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY Topic, SlidingWindow(second, 10)
HAVING COUNT(*) > 10
```

Tell me the count of tweets per time zone every 10 seconds



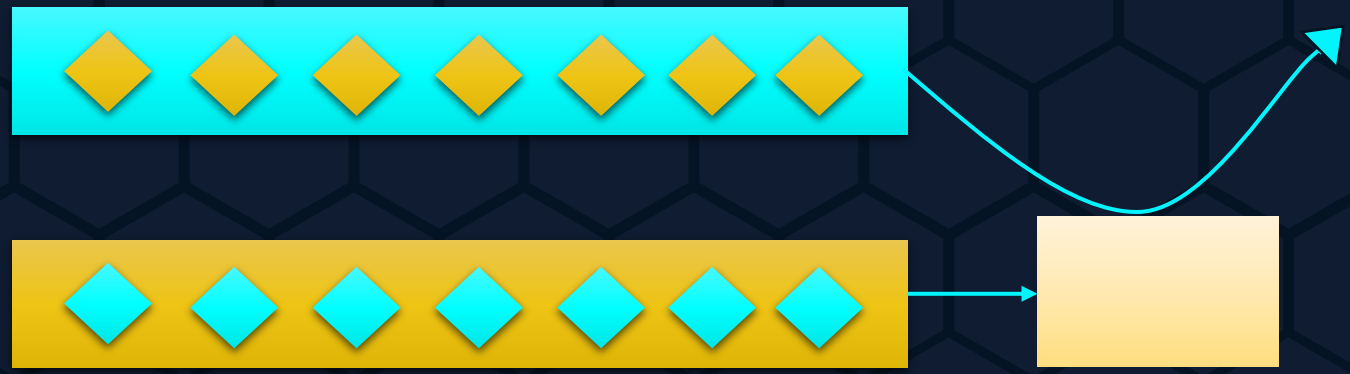

```
SELECT TimeZone, COUNT(*) AS Count
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TimeZone, TumblingWindow(second,10)
```

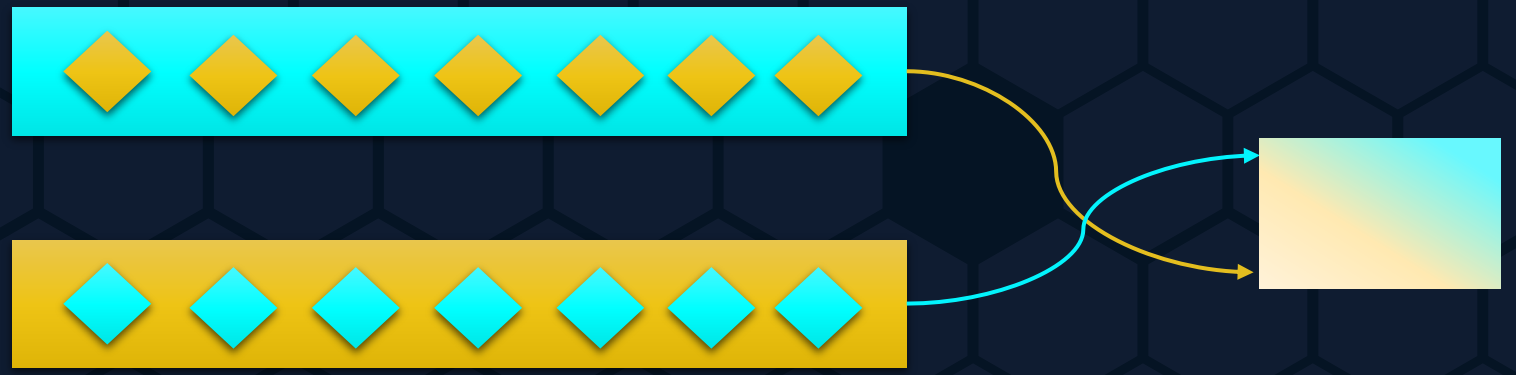
Windows

Stream SQL

Χαρακτηριστικά Stream SQL

Ενοποίηση ενός παραθύρου





Ενοποίηση δύο παραθύρων

Join

Πλατφόρμες ροής δεδομένων

Spark Streaming

1. Υψηλή συνδεσιμότητα με άλλες πηγές

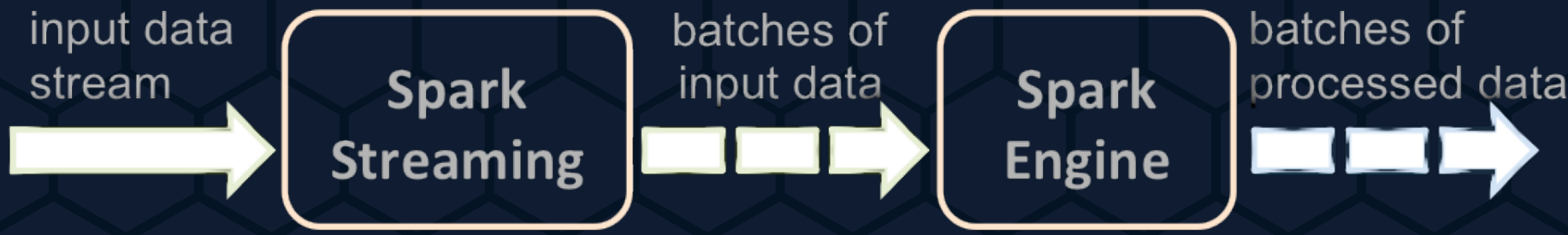
Kafka, Flume, Kinesis ή θύρες TCP

2. Λειτουργίες Υψηλού Επιπέδου

- MapReduce, Connect , Window
- Πλήρης εκμετάλλευση λειτουργιών Lambda Architecture
- Spark SQL

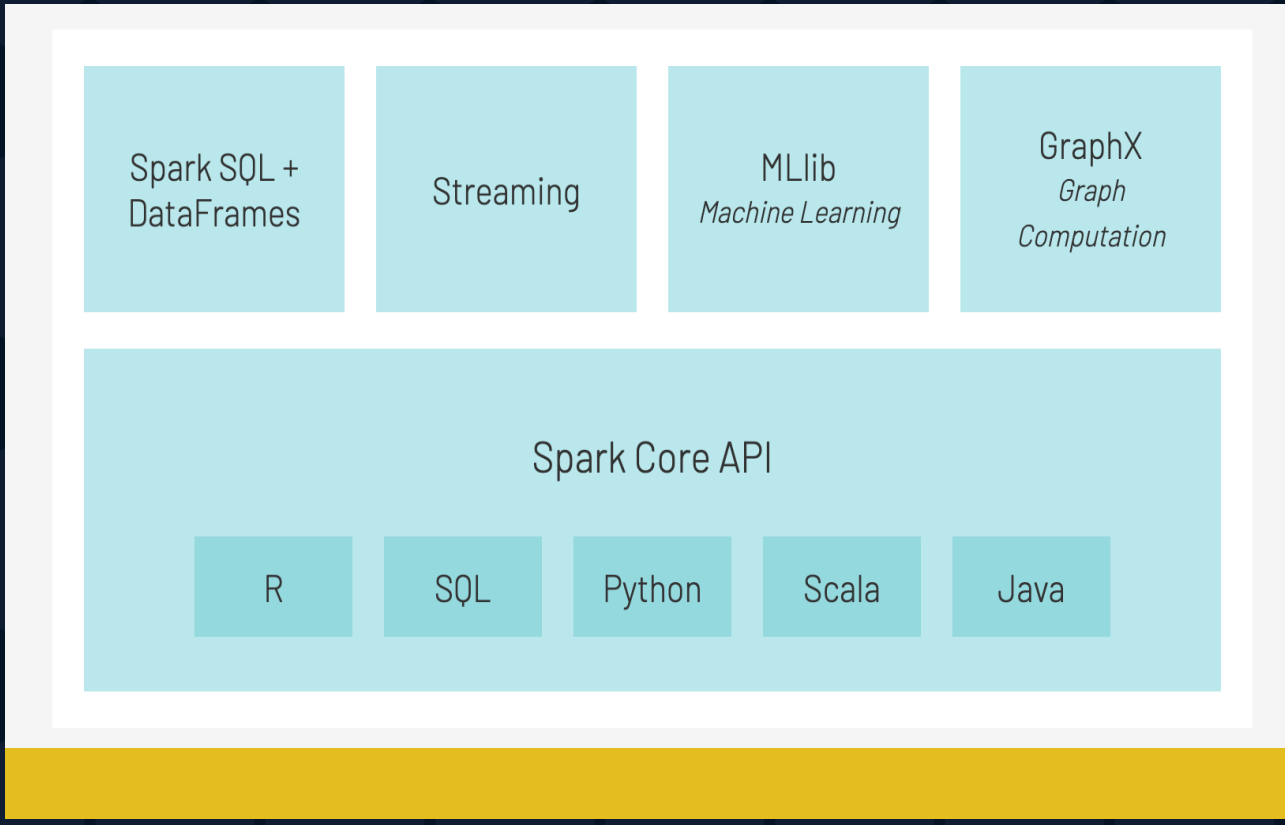
3. DStreams

- Αφαίρεση υψηλού επιπέδου η οποία αντιπροσωπεύει μια συνεχή ροή δεδομένων.
- Δημιουργούνται είτε από ροές δεδομένων εισόδου από πηγές , είτε εφαρμόζοντας λειτουργίες υψηλού επιπέδου σε άλλα DStreams



Βασικές λειτουργίες

- Διακριτές ροές (DStreams)
- Εισερχόμενα DStreams και Αποδέκτες
- Μετασχηματισμοί σε DStreams
- Δεδομένα εξόδου σε DStreams
- DataFrame και Λειτουργίες SQL
- Λειτουργίες MLlib
- Caching / Προσωρινή αποθήκευση
- Σημεία ελέγχου
- Συσσωρευτές, μεταβλητές εκπομπής



Πλεονεκτήματα

- Υποστηρίζει την αρχιτεκτονική Lambda και παρέχεται δωρεάν με το Apache Spark.
- Δίνει τη δυνατότητα υψηλών αποδόσεων και αποτελεί καλή λύση για πολλές περιπτώσεις χρήσης όπου δεν απαιτείται καθυστέρηση.
- Έχει υψηλή ανοχή σφαλμάτων
- Απλό στη χρήση API υψηλότερου επιπέδου
- Αναφέρεται σε μεγάλο ποσοστό χρηστών και επιδέχεται ραγδαίων βελτιώσεων

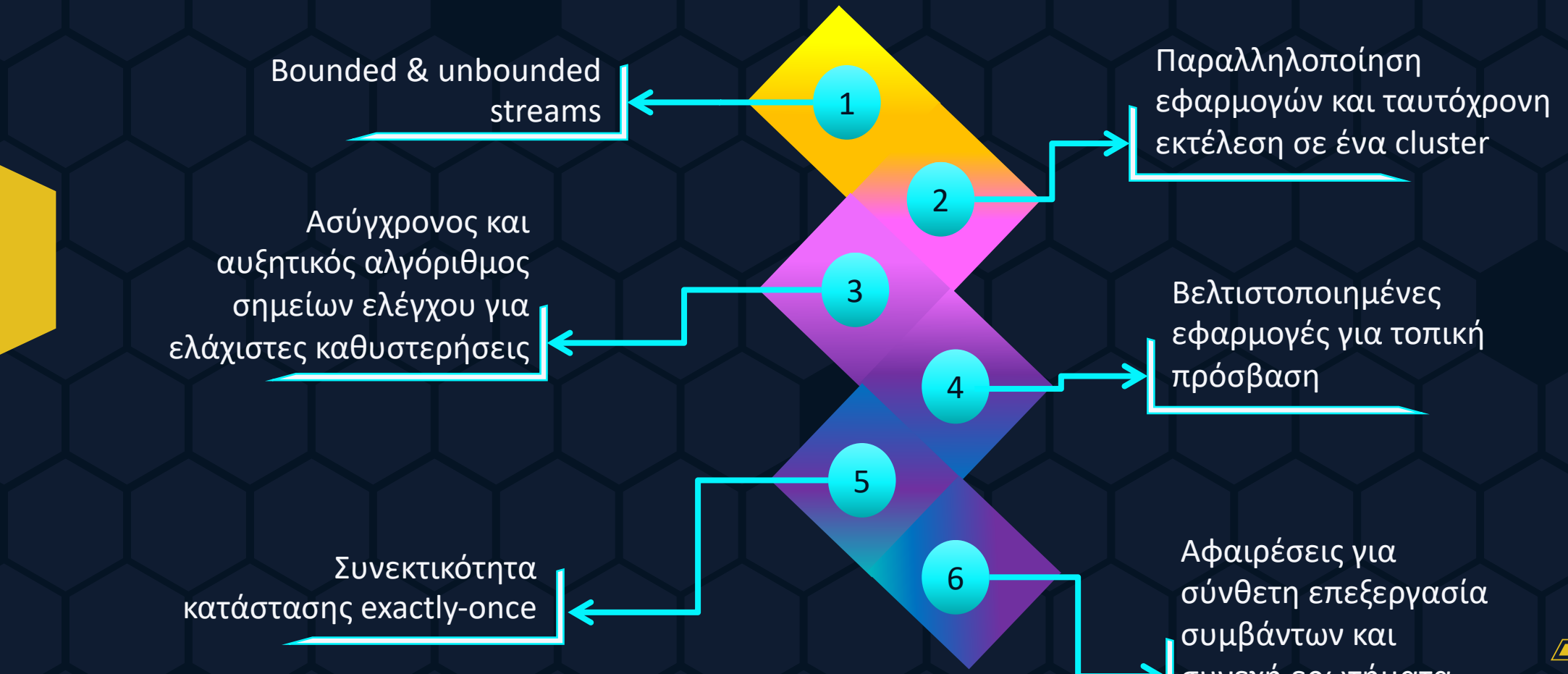
Μειονεκτήματα

- Δεν υποστηρίζει πραγματική ροή δεδομένων, δεν είναι κατάλληλο για απαιτήσεις χαμηλού λανθάνοντος χρόνου.
- Αποτελείται από πάρα πολλές παραμέτρους και είναι δύσκολο στη ρύθμισή του.
- Υστερεί αρκετά συγκριτικά με το Apache Flink το οποίο περιέχει προηγμένα χαρακτηριστικά.

Flink

Χαρακτηριστικά του

Apache Flink



Σύνολο βασικών λειτουργιών του

Apache Flink

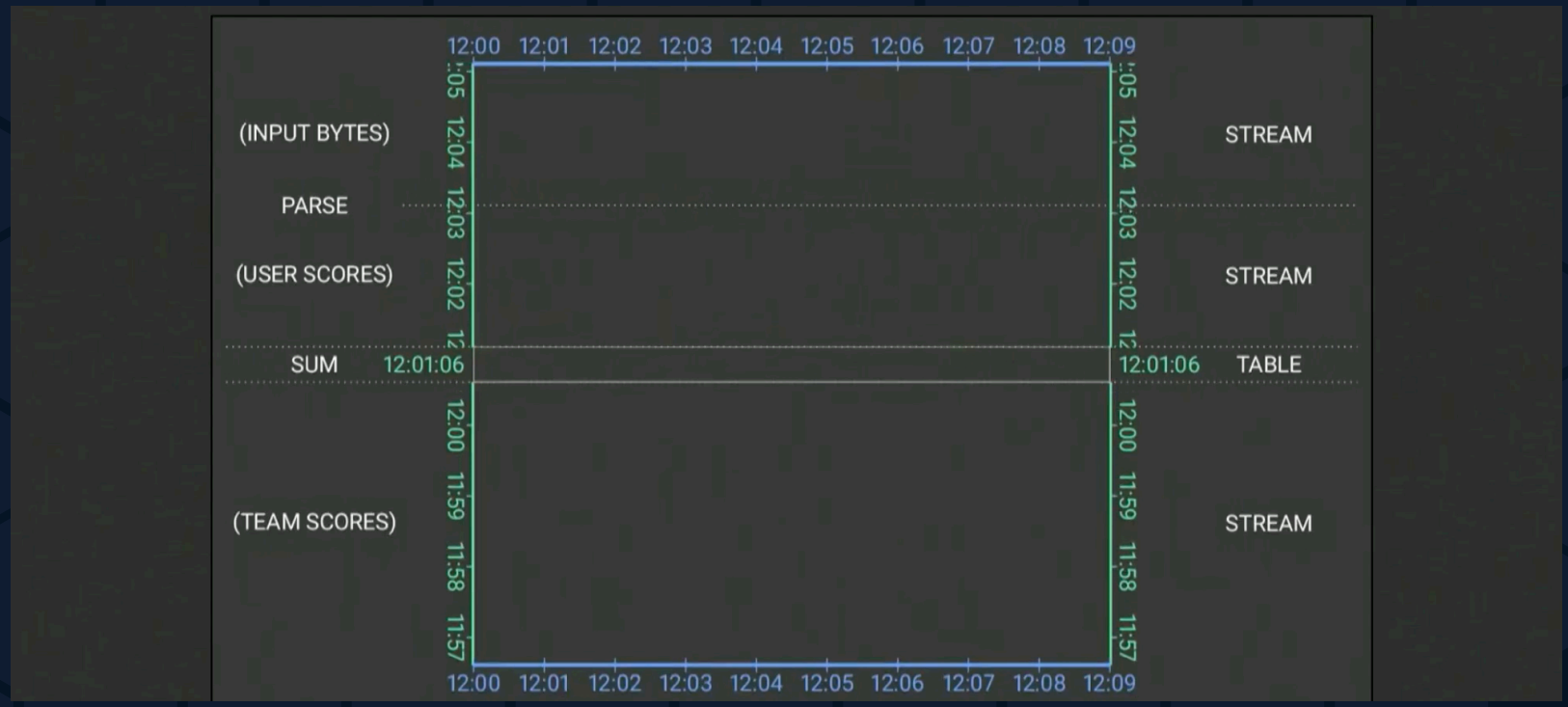
1. **Λειτουργία χρόνου συμβάντος**

Επιτρέπει ακριβή και συνεπή αποτελέσματα ανεξάρτητα από το εάν υποβάλλονται σε επεξεργασία τα γεγονότα που έχουν καταγραφεί ή σε πραγματικό χρόνο

2. **Υδατογράφημα**

Ευέλικτος μηχανισμός για την αντιστάθμιση της καθυστέρησης και της πληρότητας των αποτελεσμάτων

Flink



Flink

Apache Flink

Watermark

```
PCollection<KV<User, Score>> input = IO.read(...)
        .apply(ParDo.of(new ParseFn());
        .apply(Window.into(FixedWindows.of(Duration.standardMinutes(2))
                .triggering(AtWatermark())
        .apply(Sum.integersPerKey());
```

Apache Flink APIs

1. Process Function

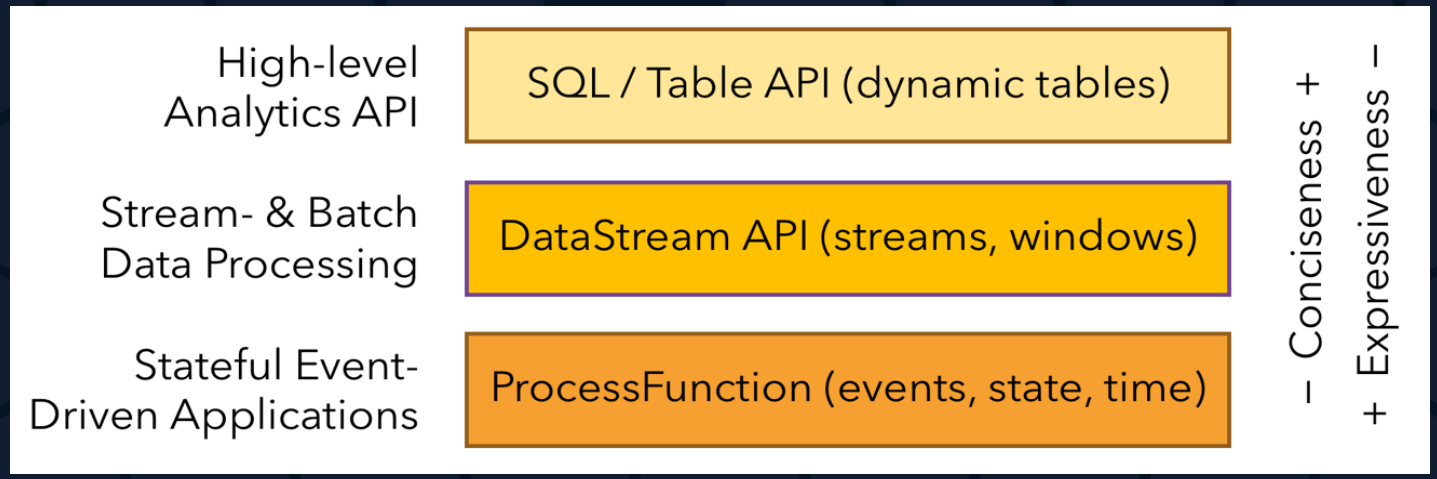
   παρέχει λεπτομερή έλεγχο του χρόνου και της κατάστασης

2. DataStream API

   λειτουργία παραθύρου, μετασχηματισμούς καταγραφής χρόνου και εμπλουτισμό συμβάντων με ερωτήματα σε εξωτερικούς χώρους αποθήκευσης δεδομένων. Βασίζεται σε λειτουργίες, όπως map(), reduce() και aggregate()

3. **SQL & Table API**

   επικύρωση και βελτιστοποίηση ερωτημάτων σε μορφή πίνακα



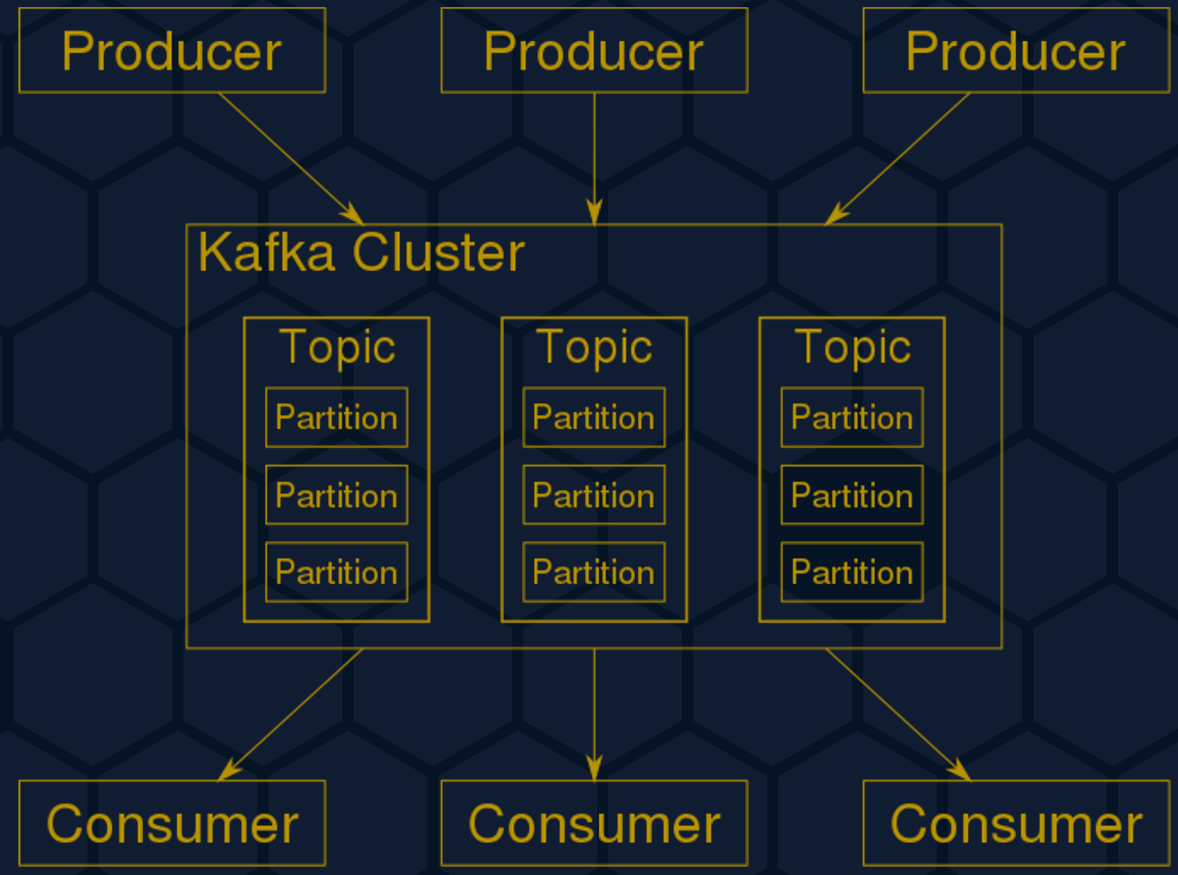
Apache Kafka

- πλατφόρμα επεξεργασίας ροής ανοιχτού κώδικα
- επιτρέπει να δημοσιεύσουμε και να εγγραφούμε σε ροές εγγραφών (publish and subscribe)
- να αποθηκεύσουμε ροές εγγραφών με τρόπο ανεκτικό σε σφάλματα
- να επεξεργαστούμε ροές εγγραφών κατά τη στιγμή που συμβαίνουν

APACHE kafka®
A distributed streaming platform

Apache Kafka

- πολύτιμο για εταιρικές λειτουργίες ώστε να επεξεργάζονται ροές δεδομένων εκμεταλλευόμενες το γρήγορο, επεκτάσιμο και ανθεκτικό χαρακτήρα του.
- εκτελείται ως cluster σε έναν ή περισσότερους διακομιστές που μπορούν να εκτείνονται σε πολλά κέντρα δεδομένων.
- αποθηκεύει ροές αρχείων σε κατηγορίες που ονομάζονται topics. Κάθε εγγραφή αποτελείται από ένα κλειδί, μια τιμή και μια χρονική σήμανση.



APACHE kafka
A distributed streaming platform

Apache Kafka APIs

1. Producer API

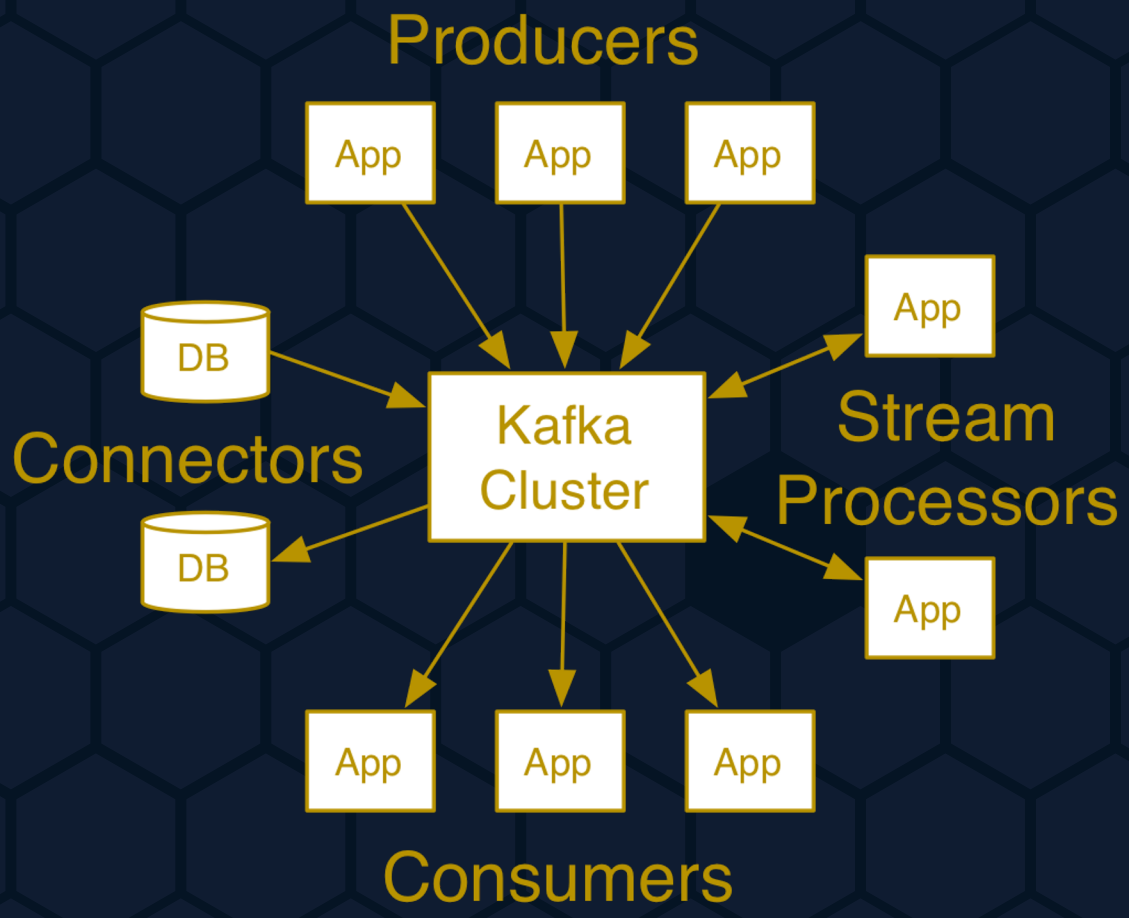
επιτρέπει σε μια εφαρμογή να δημοσιεύει μια ροή αρχείων σε ένα ή περισσότερα topics του Kafka

2. Consumer API

επιτρέπει σε μια εφαρμογή να εγγραφεί σε ένα ή περισσότερα topics και να επεξεργάζεται τη ροή των εγγραφών που παράγονται σε αυτά

3. Streams API

επιτρέπει σε μια εφαρμογή να ενεργεί ως επεξεργαστής ροής μετατρέποντας αποτελεσματικά τις ροές εισόδου σε ροές εξόδου



APACHE kafka
A distributed streaming platform

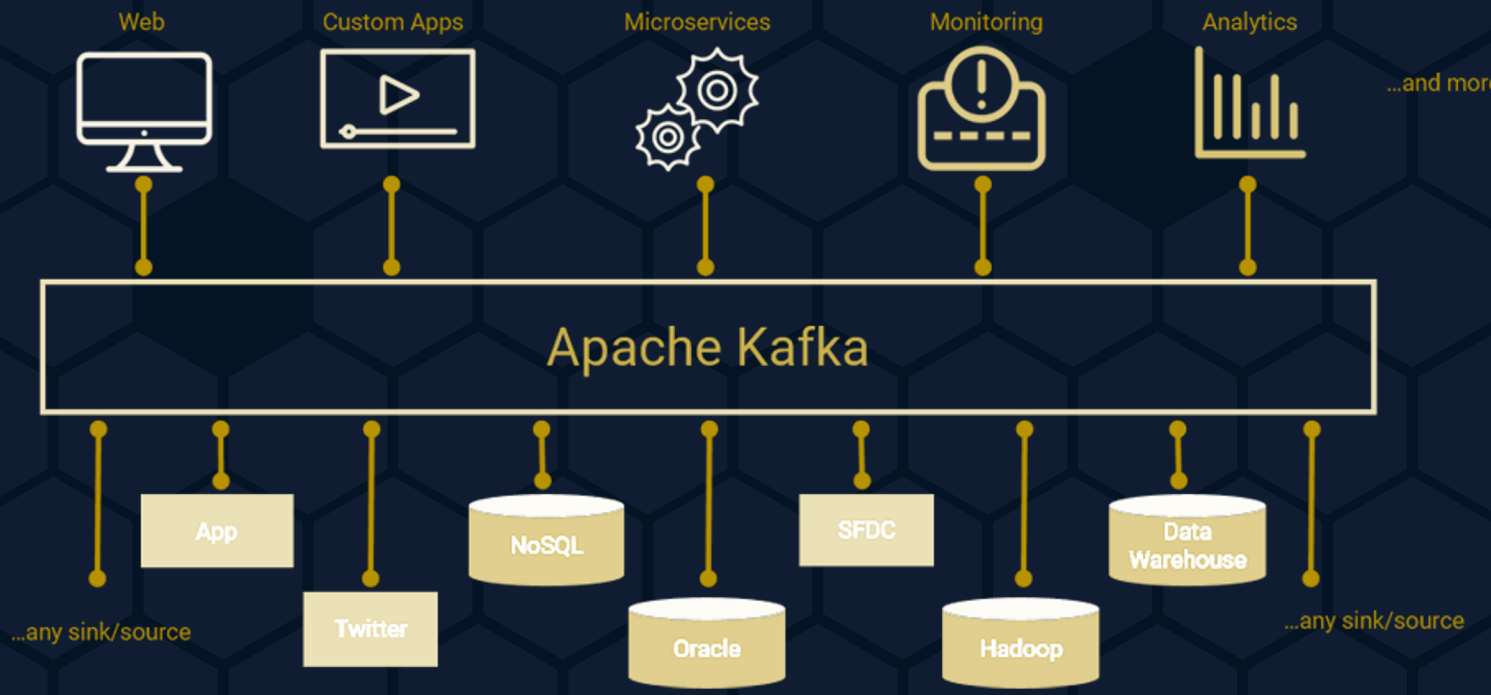
Apache Kafka APIs

4. Connector API

επιτρέπει την κατασκευή και λειτουργία επαναχρησιμοποιήσιμων παραγωγών ή καταναλωτών που συνδέουν Kafka topics με υπάρχουσες εφαρμογές ή συστήματα δεδομένων

5. Admin API

επιτρέπει τη διαχείριση και τον έλεγχο topic, και άλλων αντικειμένων Kafka



APACHE kafka
A distributed streaming platform

Πλεονεκτήματα

- Πολλαπλοί παραγωγοί (producers).
- Πολλαπλοί καταναλωτές (consumers).
- Διατήρηση δεδομένων βάσει δίσκου.
- Χαμηλή καθυστέρηση.
- Ανοχή σε σφάλματα,
  ανθεκτικό σε αστοχίες κόμβου /μηχανήματος εντός cluster
- Διάρκεια,
  τα μηνύματα δεν χάνονται ποτέ.

Μειονεκτήματα

- Δε διαθέτει ένα πλήρες πακέτο εργαλείων διαχείρισης και παρακολούθησης, γεγονός που δημιουργεί ανασφάλεια στις επιχειρήσεις που θέλουν να το επιλέξουν.
- Χαμηλή απόδοση συστήματος σε περίπτωση τροποποίησης μηνύματος
- Πλήρης ταυτοποίηση του ονόματος του topic σε περίπτωση αναζήτησης

APACHE kafka
A distributed streaming platform

Apache Samza

- framework ανοιχτού κώδικα για κατανεμημένη επεξεργασία ροών συμβάντων μεγάλου όγκου.
- Υποστηρίζει υψηλή απόδοση για ένα ευρύ φάσμα μοτίβων επεξεργασίας, παρέχοντας παράλληλα λειτουργική ευρωστία σε μαζική κλίμακα που απαιτείται από εταιρείες Διαδικτύου.

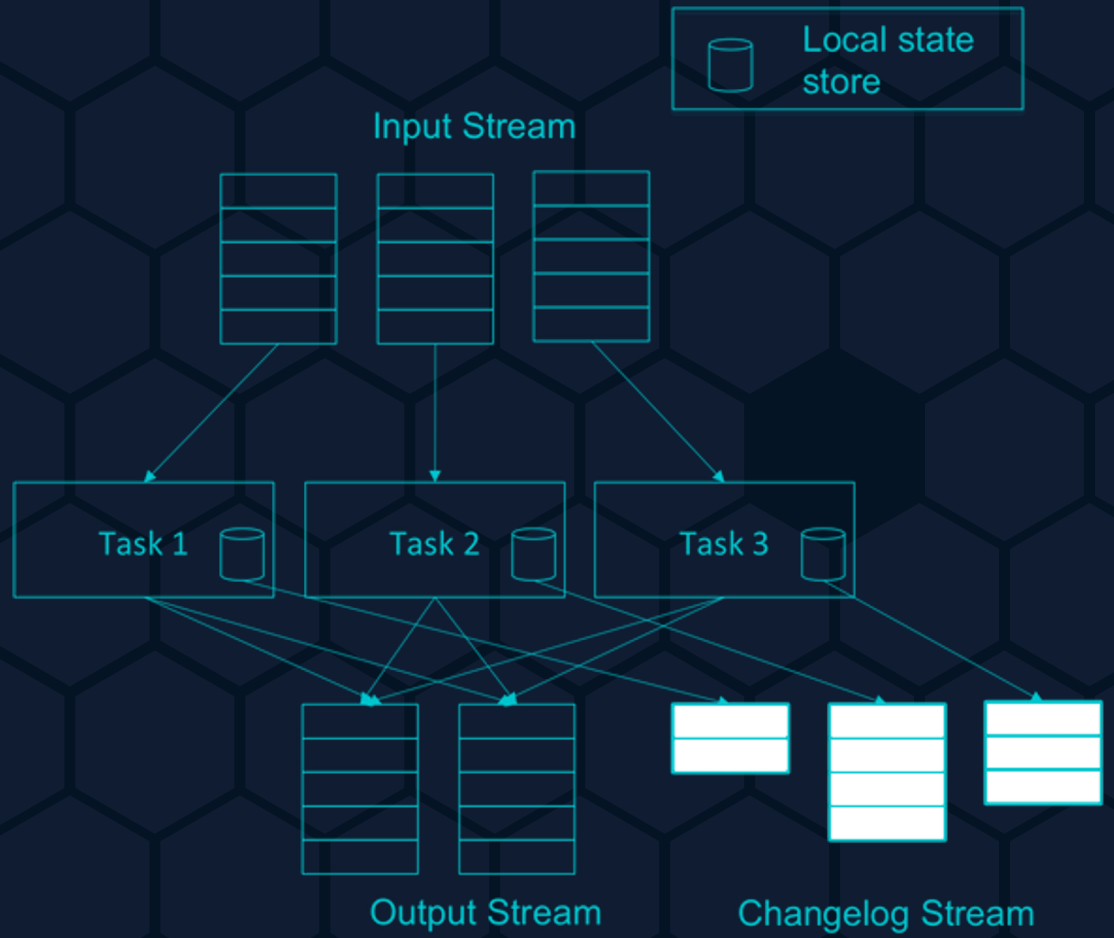
Σημαντικά του σημεία για την υψηλή επεκτασιμότητα και ευρωστία του

1. Επεξεργασία κατατμημένων αρχείων καταγραφής
2. Τοπική κατάσταση ανεκτική σε σφάλματα.
3. Προγραμματισμός εργασιών βάσει cluster

Λειτουργία του

Apache Samza

- Χρήση μηνυμάτων.
- Οι ροές δεδομένων χωρίζονται σε διαμερίσματα και κάθε ένα αποτελεί μια ταξινομημένη ακολουθία μηνυμάτων μόνο για ανάγνωση με κάθε μήνυμα να έχει ένα μοναδικό αναγνωριστικό (**offset**) .
- Κατανάλωση πολλών μηνυμάτων από το ίδιο διαμέρισμα ροής διαδοχικά .
- Συνήθως χρησιμοποιείται σε συνδυασμό με το Apache Kafka.



Λειτουργία του

Apache Samza

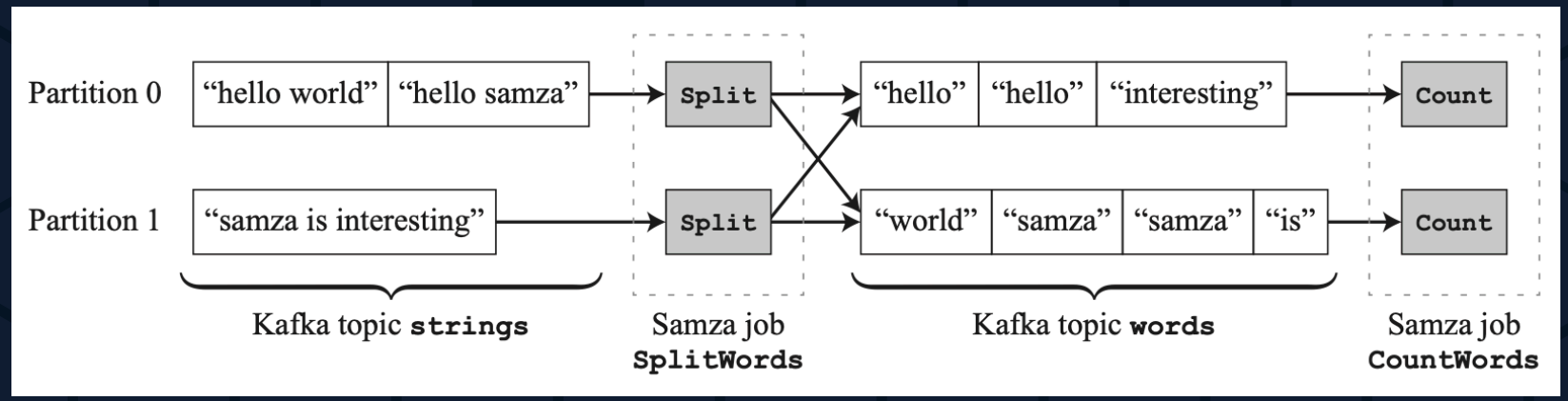
- Κάθε εργασία Samza SQL αποτελείται από μία ή περισσότερες δηλώσεις Samza SQL.
- Κάθε δήλωση αντιπροσωπεύει έναν αγωγό συνεχούς ροής.
- Μια εργασία του αποτελείται από ένα σύνολο στιγμιότυπων Java Virtual Machine (JVM) που περιέχουν το Samza framework.

```
class SplitWords implements StreamTask {

  static final SystemStream WORD_STREAM =
    new SystemStream("kafka", "words");

  public void process(
        IncomingMessageEnvelope in,
        MessageCollector out,
        TaskCoordinator _) {

    String str = (String) in.getMessage();

    for (String word : str.split(" ")) {
      out.send(
        new OutgoingMessageEnvelope(
          WORD_STREAM, word, 1));
    }
  }
}
```

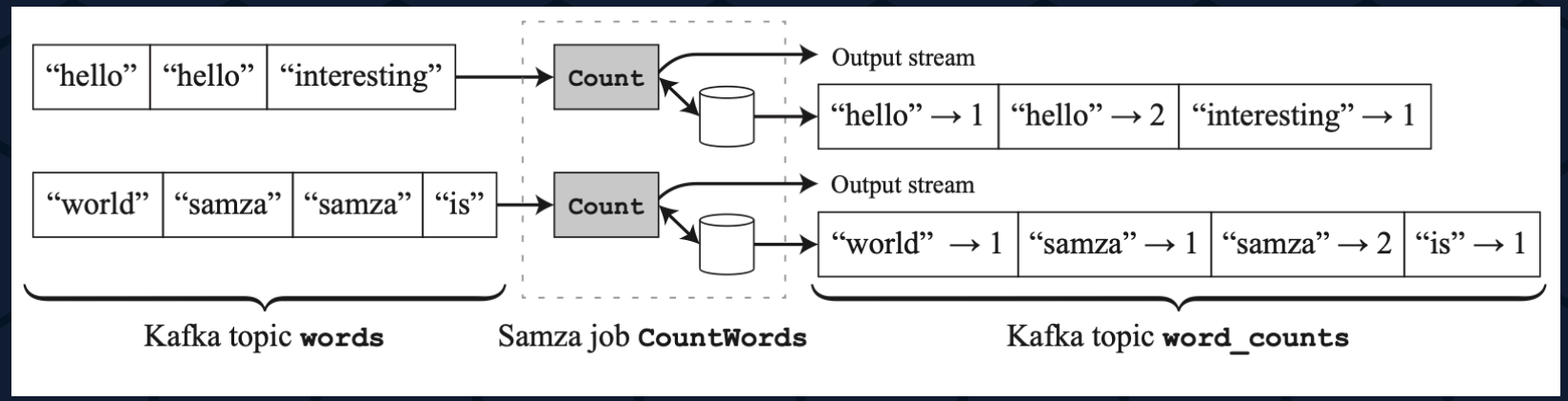
```
class CountWords implements StreamTask,
                            InitableTask {

  private KeyValueStore<String, Integer> store;

  public void init(Config config,
                   TaskContext context) {
    store = (KeyValueStore<String, Integer>)
      context.getStore("word-counts");
  }

  public void process(
        IncomingMessageEnvelope in,
        MessageCollector out,
        TaskCoordinator _) {

    String word = (String) in.getKey();
    Integer inc = (Integer) in.getMessage();

    Integer count = store.get(word);
    if (count == null) count = 0;
    store.put(word, count + inc);
  }
}
```

Λειτουργία του

Apache Samza



Κατανάλωση δεδομένου εισόδου ενός διαμερίσματος και αποστολή εξόδου σε οποιοδήποτε διαμέρισμα



Διεργασία τοπικού διαμερίσματος η οποία καθίσταται διαρκής εκπέμποντας ένα log file αλλαγών στο Kafka

Συνεργασία

Apache Samza & Kafka

Το Apache Samza δεν εφαρμόζει το δικό του επίπεδο μεταφοράς μηνυμάτων για την παράδοση μηνυμάτων μεταξύ των χειριστών ροής αλλά κάνει χρήση του Apache Kafka.

Πλεονεκτήματα αυτή της χρήσης

- Εκμετάλλευση τοπικής αποθήκευσης Kafka για γρήγορη προσπέλαση προβλημάτων κατά τη ζωντανή επεξεργασία περιορίζοντας καθυστερήσεις και λάθη.
- Κάθε ροή μηνυμάτων στο σύστημα είναι προσβάσιμη για εντοπισμό σφαλμάτων και παρακολούθηση σε οποιοδήποτε σημείο

Πλεονεκτήματα

- Πολύ καλό στη διατήρηση μεγάλων ποσοτήτων πληροφορίας (καλό για χρήση σε περιπτώσεις σύνδεσης ροών – joining streams) χρησιμοποιώντας το αρχείο καταγραφής RockDb και Kafka.
- Είναι ανεκτικό σε σφάλματα και παρέχει υψηλές αποδόσεις χρησιμοποιώντας τις ιδιότητες του Apache Kafka.
- Χαμηλός λανθάνων χρόνος.

Μειονεκτήματα

- Στενά συνδεδεμένο με το Kafka.
- Δύσκολη χρήση του Apache Kafka.
- Υποστηρίζει μόνο γλώσσες Java Virual Machine (JVM).
- Δεν υποστηρίζει πολύ χαμηλό λανθάνοντα χρόνο.
- Δεν παρέχει “exactly-once semantics”

samza

Αρχιτεκτονική Λάμδα

- Αρχιτεκτονική επεξεργασίας δεδομένων που έχει σχεδιαστεί για να χειρίζεται τεράστιες ποσότητες δεδομένων, εκμεταλλευόμενη τις μεθόδους επεξεργασίας δεδομένων δέσμης (batch data) και δεδομένων ροής (stream data).
- Εξισορροπεί τον λανθάνοντα χρόνο, την απόδοση και την ανοχή σφαλμάτων.
- Παρέχει προβολή παρουσιάσεις δεδομένων δέσμης και χρησιμοποιεί επεξεργασία δεδομένων ροής σε πραγματικό χρόνο για την παροχή παρουσίασης δεδομένων στο διαδίκτυο.
- Μετριάζει τις καθυστερήσεις του MapReduce.

Επίπεδα

Αρχιτεκτονικής Λάμδα

- **Επίπεδο δέσμης (Apache Hadoop)**

✓ προ-επεξεργάζεται τα αποτελέσματα χρησιμοποιώντας ένα κατανεμημένο σύστημα επεξεργασίας που μπορεί να χειριστεί πολύ μεγάλες ποσότητες δεδομένων.

✓ Στοχεύει στην απόλυτη ακρίβεια με δυνατότητα επεξεργασίας όλων των διαθέσιμων ειδών δεδομένων κατά τη δημιουργία παρουσίασης τους

✓ Η έξοδος συνήθως αποθηκεύεται σε μια βάση δεδομένων μόνο για ανάγνωση.

- **Επίπεδο γρήγορης επεξεργασίας (Apache Storm, Apache Spark κ.α.)**

✓ στοχεύει στην ελαχιστοποίηση της καθυστέρησης παρέχοντας παρουσιάσεις δεδομένων σε πραγματικό χρόνο στα πιο πρόσφατα δεδομένα .

- **Επίπεδο εξυπηρέτησης**

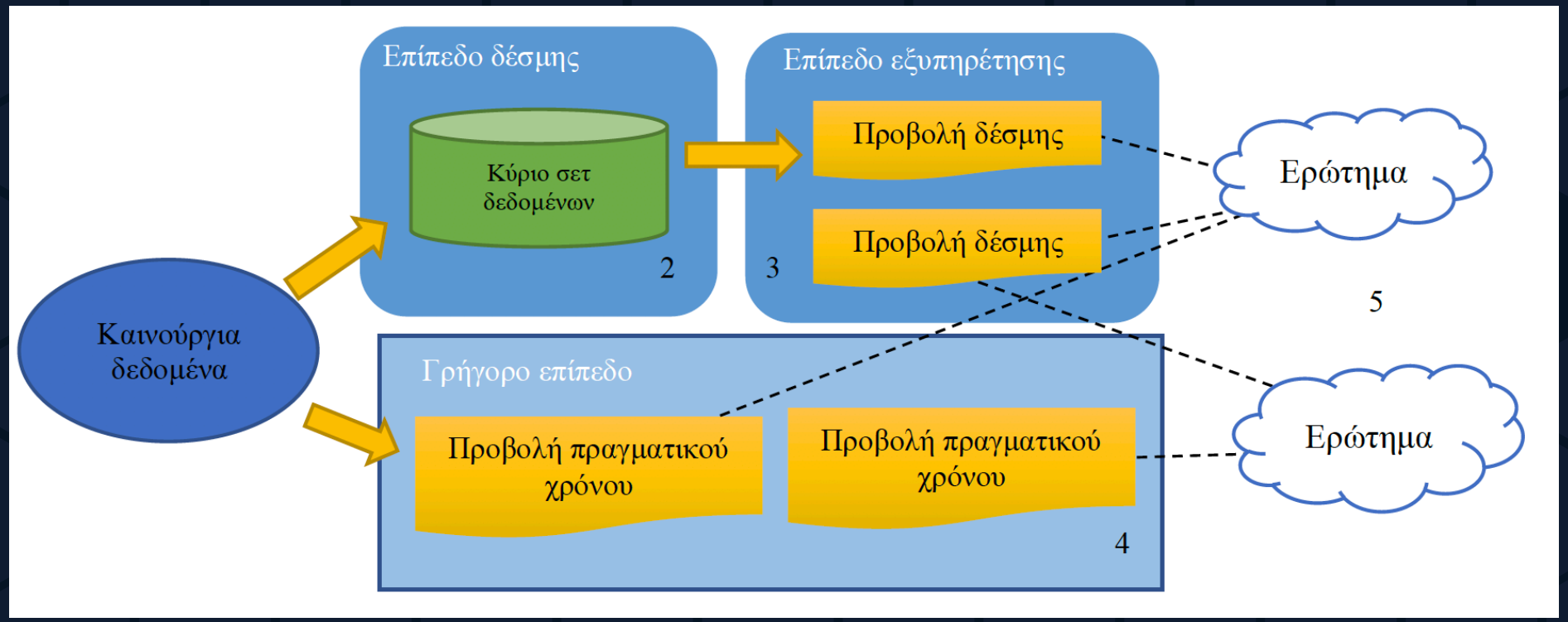
✓ Αποθήκευση αποτελεσμάτων των δυο προηγούμενων επιπέδων.

✓ Εφαρμογή ερωτημάτων

✓ Δημιουργία παρουσίασης αποτελεσμάτων

Λειτουργία

Αρχιτεκτονικής Λάμδα



Πλεονεκτήματα

- Διατηρεί τα δεδομένα εισόδου αμετάβλητα
- Η πειθαρχημένη μοντελοποίηση στο μετασχηματισμό δεδομένων μας επιτρέπει τον εντοπισμό σφαλμάτων σε κάθε στάδιο ανεξάρτητα.
- γίνεται αντιληπτό το πρόβλημα της επανεπεξεργασίας των δεδομένων ροής
- Διαθέτει έξτρα στρώμα επεξεργασίας υψηλής ακρίβειας
- Παρέχει δεξαμενή ανεπεξέργαστων δεδομένων

Μειονεκτήματα

- Η συντήρηση του κώδικα
- Περίπλοκος προγραμματισμός για το Storm και το Hadoop.
- ο κώδικας καταλήγει να δημιουργείται ειδικά για το framework στο οποίο λειτουργεί.
- το σύστημα επεξεργασίας δεδομένων δεν μπορεί να εξελιχθεί.

Συγκριτική ανάλυση frameworks

Λειτουργία Παραθύρου (Windowing)	Apache Samza	Apache Flink	Apache Storm	Apache Spark
Fixed Windows	NAI	NAI	NAI	NAI
Sliding Windows	NAI	NAI	NAI	NAI
Session Windows	NAI	NAI	OXI	OXI
Global Windows	NAI	NAI	NAI	OXI

Μηχανισμός Παραθύρου - Windowing

Συγκριτική ανάλυση frameworks

Μηχανές streaming	Τρόποι διαχείρισης κατάστασης
Apache Samza	Key-value , αποθήκευση τοπικά, RocksDB
Apache Flink	Key-value , τοπική καθώς και εξωτερική αποθήκευση, RocksDB
Apache Storm	Key-value, χρήση μνήμης, τοπική αποθήκευση,
Apache Spark	Εξωτερική αποθήκευση , state-store

Διαχείριση κατάστασης δεδομένων - State management

Συγκριτική ανάλυση frameworks

Μηχανές streaming	Εγγυήσεις Επεξεργασίας
Apache Samza	Ακριβώς μια (Exactly-Once)
Apache Flink	Ακριβώς μια (Exactly-Once)
Apache Storm	Τουλάχιστον μια και Ακριβώς μια (Exactly-Once) με τη χρήση Trident
Apache Spark	Ακριβώς μια (Exactly-Once)

Εγγύηση επεξεργασίας δεδομένων

Συγκριτική ανάλυση frameworks

Μηχανές streaming	Ανεκτικότητα σε σφάλματα
Apache Samza	Change log , Host-affinity
Apache Flink	Checkpoint , Επανεκτέλεση της ροής
Apache Storm	Checkpoint , Επανεκτέλεση της ροής
Apache Spark	Checkpoint , Write-ahead-log

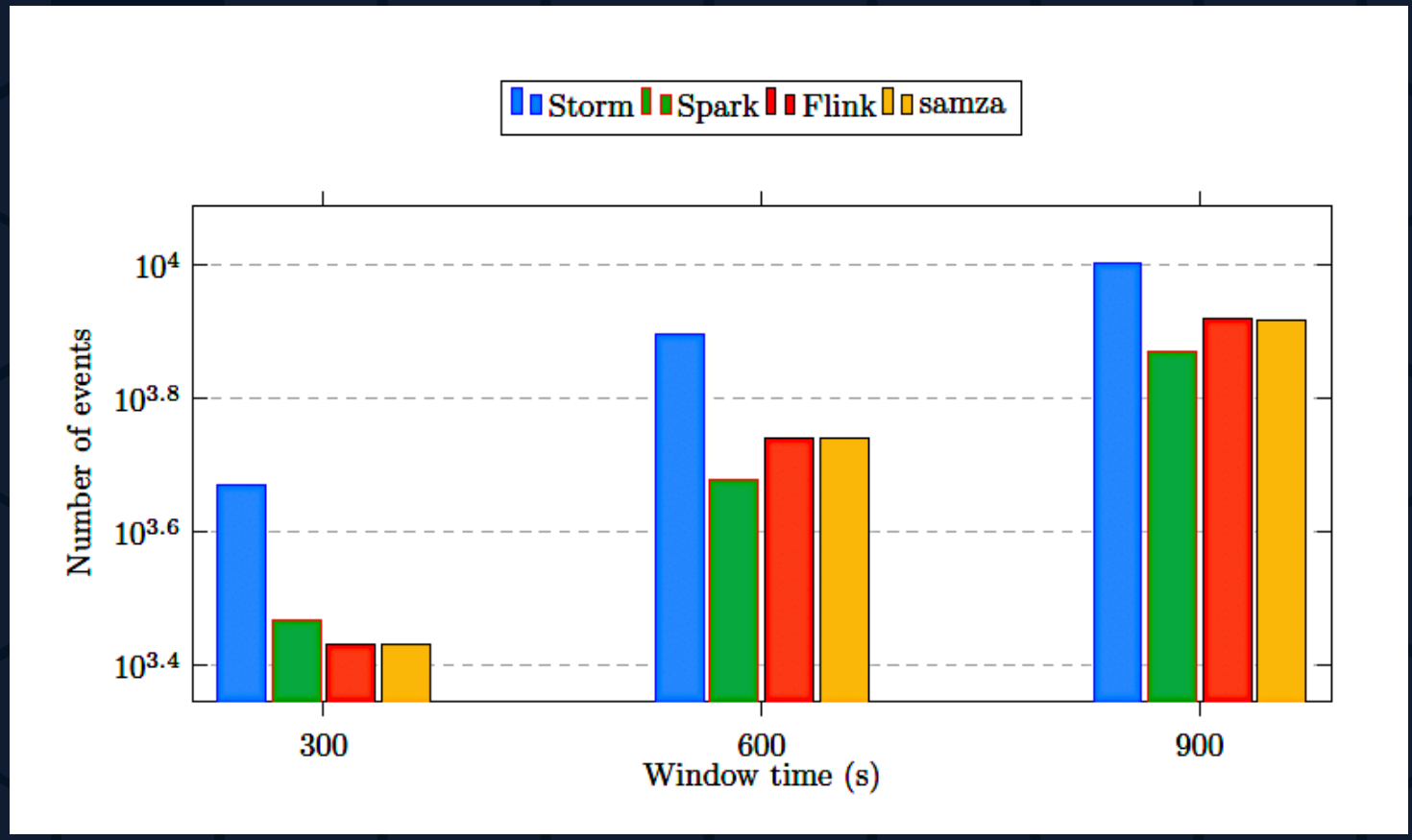
Ανεκτικότητα σε σφάλματα - Fault tolerance

Πειραματική μελέτη

Σε αυτή τη διαδικασία μελετήθηκαν:

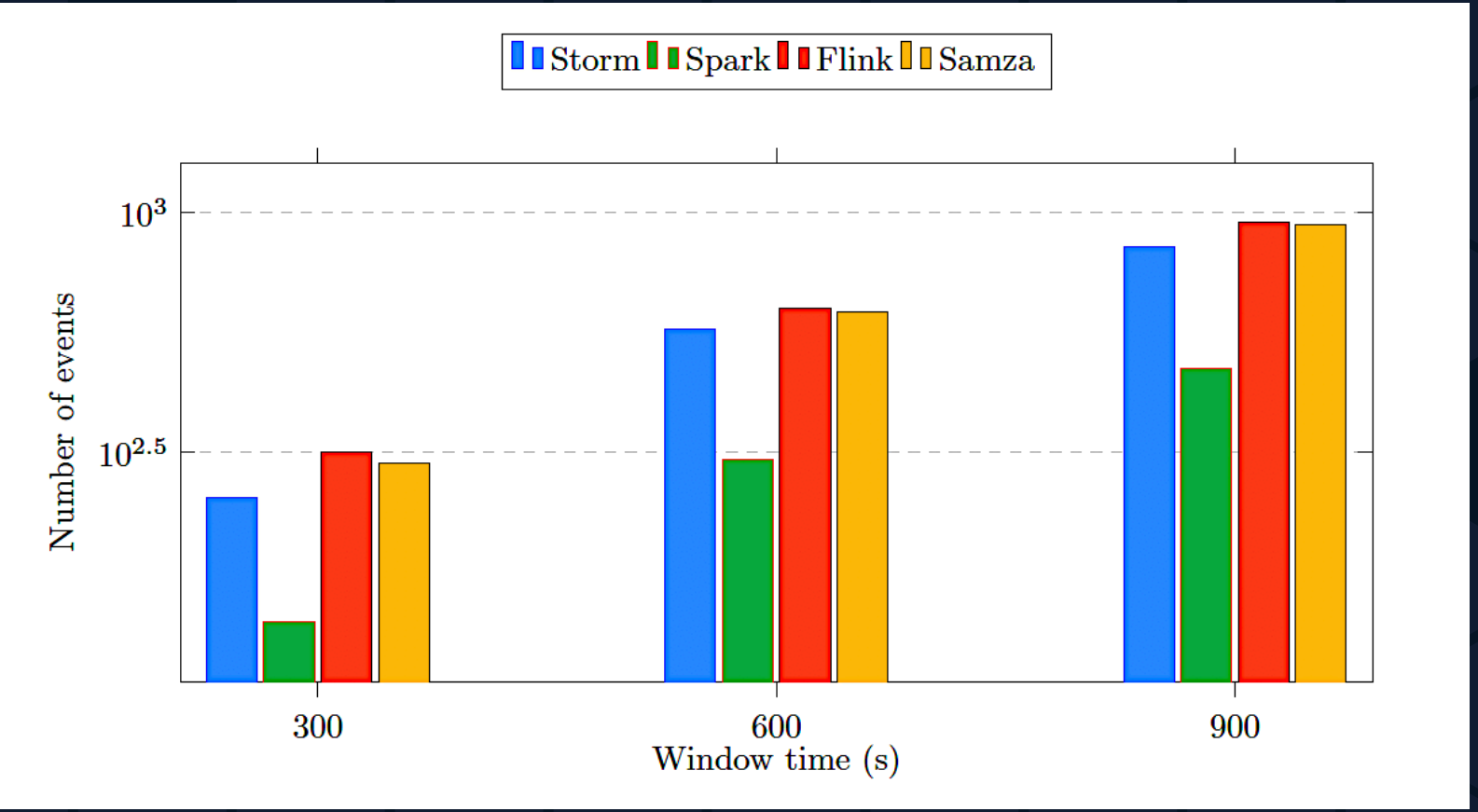
1. ο αριθμός των μηνυμάτων που υποβάλλονται σε επεξεργασία από κάθε πλαίσιο σε μια δεδομένη περίοδο,
2. το αντίκτυπο του μεγέθους του μηνύματος στον αριθμό των επεξεργασμένων μηνυμάτων και
3. η κατανάλωση πόρων από τα διάφορα frameworks που εφαρμόστηκαν.

Πειραματική μελέτη



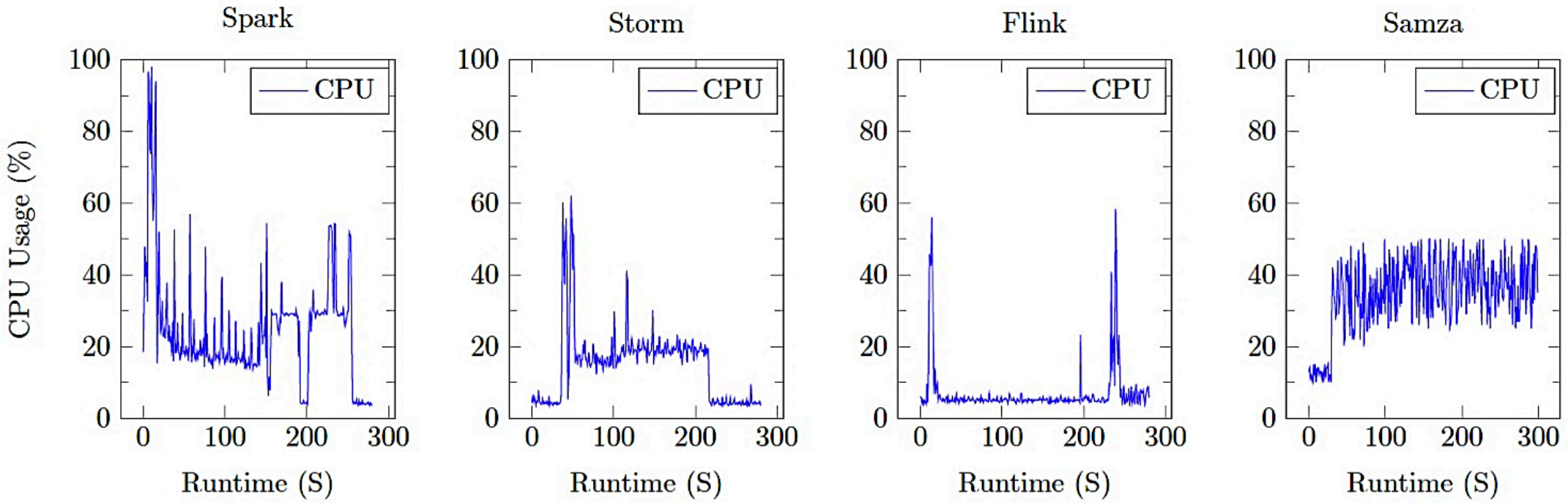
Αντίκτυπο χρόνου παραθύρου στο χρόνο συμβάντων προς επεξεργασία (100 KB ανά μήνυμα)

Πειραματική μελέτη



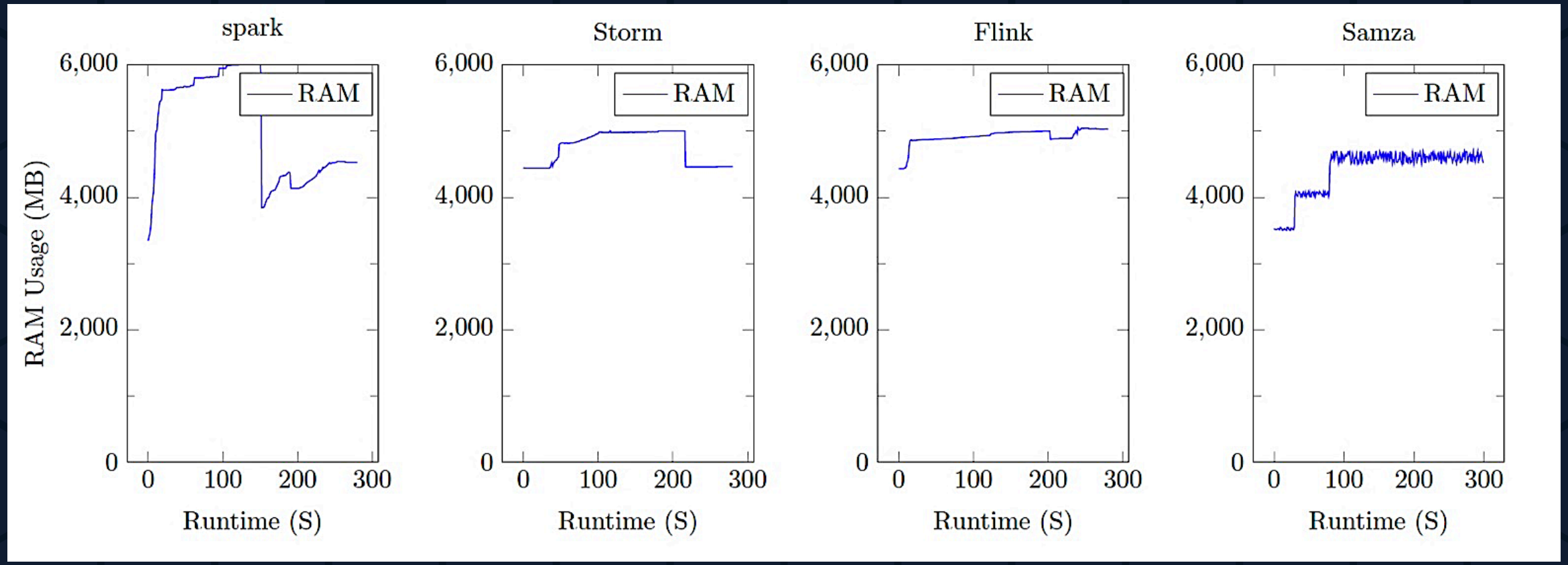
Αντίκτυπο χρόνου παραθύρου στο χρόνο συμβάντων προς επεξεργασία (500 KB ανά μήνυμα)

Πειραματική μελέτη



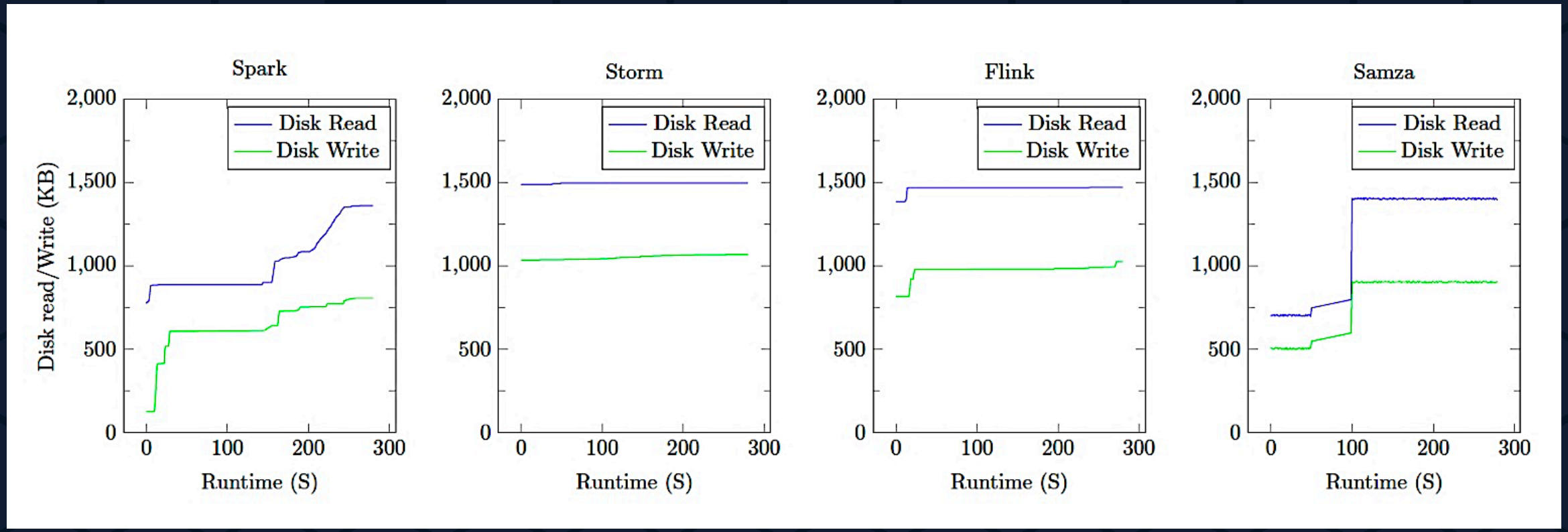
Κατανάλωση CPU

Πειραματική μελέτη



Κατανάλωση RAM

Πειραματική μελέτη



Κατανάλωση δίσκου για εργασίες Read/Write

Σας ευχαριστώ

Γράβας Χριστόφορος
18064