

Εισαγωγή

Τα τελευταία χρόνια αναπτύσσεται ένας σημαντικός αριθμός εφαρμογών που σχετίζονται με τη διαχείριση και την επεξεργασία μιας νέας μορφής πληροφορίας η οποία δεν έχει πλέον στατική μορφή, αλλά αποτελεί μια συνεχή ροή δεδομένων (data stream). Μια ροή δεδομένων ορίζεται ως μια ακολουθία ψηφιακά κωδικοποιημένων σημάτων που αναπαριστούν πληροφορία προς μετάδοση [5]. Πιο συγκεκριμένα μια ροή δεδομένων είναι μια συνεχής ακολουθία δεδομένων που παράγεται με πολύ υψηλό ρυθμό και έχει απεριόριστο όγκο [6],[7]. Ο ρυθμός άφιξης των δεδομένων μπορεί να μεταβάλλεται με τρόπο μη προβλέψιμο. Επιπλέον, τα δεδομένα έχουν μεταβλητά χαρακτηριστικά (όπως ο θόρυβος) και οι εφαρμογές που επεξεργάζονται ροές δεδομένων πρέπει να ανταποκρίνονται σε απαιτήσεις πραγματικού χρόνου και να πραγματοποιούν on-line ανάλυση των πληροφοριών [7]. Τα συστήματα που επεξεργάζονται ροές δεδομένων διαφέρουν από τα παραδοσιακά συστήματα διαχείρισης βάσεων δεδομένων (Database Management Systems - DBMS) και ονομάζονται συστήματα διαχείρισης ροών δεδομένων (Data Stream Management Systems - DSMS). Οι εφαρμογές που βασίζονται σε ροές δεδομένων καλύπτουν ένα ευρύ φάσμα απαιτήσεων και θεματικών περιοχών. Μερικά παραδείγματα τέτοιων εφαρμογών είναι τα εξής [8]:

➤ Διαχείριση Δικτύων

- Μέτρηση απόδοσης δικτύων
- Παρακολούθηση δικτύου
- Ανάλυση κίνησης δικτύου
- Κίνηση IP πακέτων

➤ Επιχειρηματικές εφαρμογές

- Χρηματιστηριακές εφαρμογές: π.χ. on-line ανάλυση της κίνησης μετοχών
- Μελέτη αποθεμάτων σε πραγματικό χρόνο: εξαγωγή συσχετίσεων, παρόμοιων δεδομένων – παρακολούθηση χρονικών μεταβολών

➤ Δεδομένα στο Διαδίκτυο

- Παρακολούθηση των Διαδικτυακών επισκέψεων ιστοσελίδων, του αριθμού επιλογών υπέρ-συνδέσεων, αναζήτηση περιεχομένου, εγγραφή σε web-sites κλπ.
 - Εφαρμογές προστασίας και ασφάλειας από «εισβολές» μέσω Διαδικτύου
 - On-line δημοπρασίες
 - Ανάλυση δοσοληψιών
- *Ροές ειδήσεων*
- Ροές δεδομένων από ειδήσεις – γεγονότα σε πραγματικό χρόνο
- *Δίκτυα Αισθητήρων*
- Συνεχόμενα δεδομένα προερχόμενα από ασύρματα δίκτυα αισθητήρων (παρακολούθηση καιρικών συνθηκών, εντοπισμός θέσης, κίνηση σε λεωφόρους, στρατιωτικές εφαρμογές)
- *Τηλεπικοινωνίες*
- Διαχείριση τηλεφωνικών κλήσεων
 - Καθορισμός του αριθμού των σταθμών βάσης που χρησιμοποιήθηκαν κατά τη διάρκεια μιας τηλεφωνικής κλήσης από κινητό τηλέφωνο
 - Εφαρμογές για κινητούς χρηστές με βάση τη θέση τους
- *Επιστημονικά Δεδομένα*
- Ιατρική
 - Αστρονομία

2.2 Συστήματα διαχείρισης ροών δεδομένων

Τα παραδοσιακά συστήματα βάσεων δεδομένων χρησιμοποιούνται σε εφαρμογές στις οποίες απαιτείται συνεχής αποθήκευση δεδομένων και πολύπλοκες επερωτήσεις. Μια βάση δεδομένων αποτελείται συνήθως από ένα σύνολο αντικειμένων όπου οι εισαγωγές, οι ενημερώσεις και οι διαγραφές είναι λιγότερο συχνές από τις επερωτήσεις. Οι επερωτήσεις εκτελούνται τη

στιγμή που υποβάλλονται και οι απαντήσεις απεικονίζουν την τρέχουσα κατάσταση της βάσης.

Επιπλέον, τα συστήματα διαχείρισης βάσεων δεδομένων δεν έχουν προκαθορισμένη «αντίληψη» του χρόνου, διατηρούν αποθηκευμένη την πληροφορία προς επεξεργασία και δεν είναι σε θέση να διαχειριστούν δεδομένα on-line (π.χ. αποθηκευμένα στην κύρια μνήμη και όχι αποθηκευμένα στο σκληρό δίσκο) ή σε σχεδόν πραγματικό χρόνο. Αντιθέτως, η διαχείριση ροών δεδομένων δημιουργεί έναν αριθμό νέων προκλήσεων που σχετίζονται με την προσαρμοζόμενη επεξεργασία πολλαπλών επερωτήσεων, την ανοχή σφαλμάτων, την αρχειοθέτηση και την παροχή απαντήσεων κατά προσέγγιση σε περιπτώσεις υπερφόρτωσης του συστήματος. Την τελευταία πενταετία τα συστήματα διαχείρισης ροών δεδομένων (DSMS) έχουν αναπτυχθεί για την on-line επεξεργασία ροών δεδομένων και την εκτέλεση συνεχών επερωτήσεων σε αυτές. Οι συνεχείς επερωτήσεις αξιολογούν χωρίς διακοπή κάθε νέο στοιχείο που εισέρχεται στο σύστημα. Τυπικές μορφές τελεστών που υποστηρίζονται από τα περισσότερα συστήματα DSMS είναι το φιλτράρισμα, και η απεικόνιση συναθροίσεων (aggregate) και συζεύξεων (join). Η χρήση παραθύρων είναι ιδιαίτερα χρήσιμη για επερωτήσεις συνάθροισης και σύζευξης. Μια σύγκριση των συστημάτων DBMS και DSMS φαίνεται στον Πίνακα 1. [12],[13]

Συστήματα DBMS	Συστήματα DSMS
Σταθερές σχέσεις (στατικές, αποθηκευμένες)	Βραχύβιες ροές δεδομένων (απαιτούν on-line ανάλυση)
Ομάδες πλειάδων	Ακολουθίες πλειάδων
Στιγμιαίες επερωτήσεις	Συνεχείς επερωτήσεις (Continuous Queries - CQ)
Τυχαία προσπέλαση	Σειριακή προσπέλαση
Απεριόριστος αποθηκευτικός χώρος	Περιορισμένη κύρια μνήμη σαν χώρος

στο σκληρό δίσκο	για αποθήκευση
Μόνο η τρέχουσα κατάσταση είναι σημαντική	Και τα παρελθόντα δεδομένα είναι σημαντικά
Δεν προσφέρουν υπηρεσίες πραγματικού χρόνου	Απαιτήσεις πραγματικού χρόνου
Σχετικά χαμηλός ρυθμός ενημέρωσης δεδομένων	Πιθανόν να εμφανιστούν ρυθμοί άφιξης πολλών GB
Τα δεδομένα θεωρούνται ακριβή	Τα δεδομένα ενδέχεται να μην είναι ακριβή
Μετατροπές στα δεδομένα	Προσαρτήσεις στα δεδομένα
Ακριβείς απαντήσεις στις ερωτήσεις	Απαντήσεις κατά προσέγγιση στις ερωτήσεις
Καθορισμένα σχέδια ερωτήσεων	Προσαρμοζόμενα σχέδια ερωτήσεων
Αυθαίρετη επεξεργασία ερωτήσεων	Επεξεργασία ερωτήσεων σε ένα πέρασμα
Ο τρόπος πρόσβασης στα δεδομένα καθορίζεται από τον επεξεργαστή ερωτήσεων και τον σχεδιασμό της βάσης δεδομένων	Μη προβλέψιμος/μεταβλητός ρυθμός άφιξης δεδομένων και μεταβλητά χαρακτηριστικά

Πίνακας 1: Σύγκριση συστημάτων DBMS και DSMS

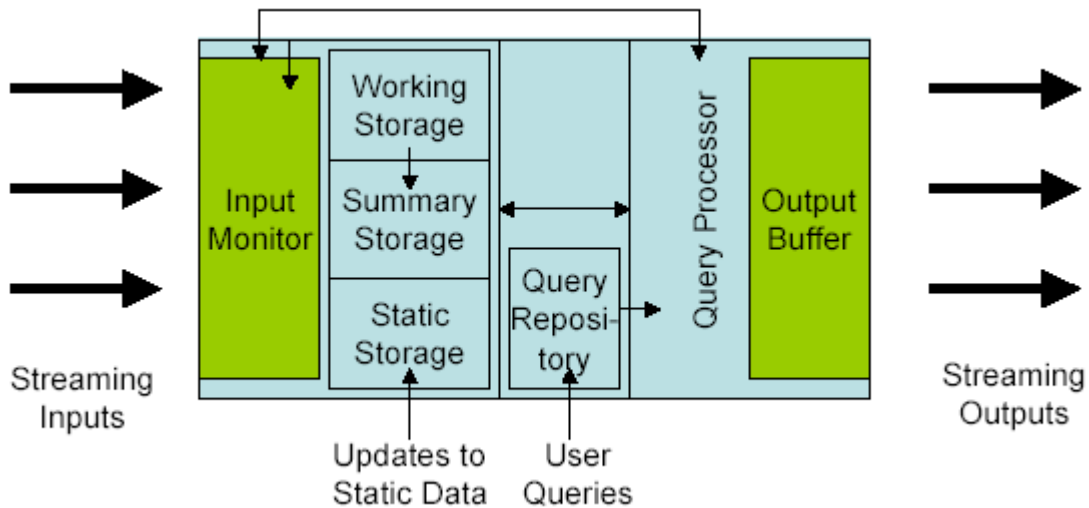
Τα μοναδικά χαρακτηριστικά των ροών δεδομένων και των συνεχών ερωτήσεων υπαγορεύουν τις ακόλουθες απαιτήσεις για τα DSMS:

- Το μοντέλο δεδομένων και η σημασιολογία των ερωτήσεων πρέπει να επιτρέπουν λειτουργίες βασισμένες στη σειρά (π.χ. οι τελευταίες εκατό

τιμές) και τον χρόνο άφιξης των δεδομένων (π.χ. επερωτήσεις με ένα κινούμενο παράθυρο πέντε λεπτών).

- Η αδυναμία αποθήκευσης μιας ροής δεδομένων στο σύνολό της οδηγεί στη χρήση προσεγγιστικών, περιληπτικών δομών όπως οι συνόψεις (synopses) ή περιλήψεις (digests). Συνεπώς, οι επερωτήσεις σε περιληπτικές δομές δεν μπορούν να επιστρέψουν ακριβείς απαντήσεις.
- Τα σχέδια επερωτήσεων σε ροές δεδομένων δεν μπορούν να χρησιμοποιήσουν ανασταλτικούς τελεστές (blocking operators) οι οποίοι πρέπει να δεσμεύσουν όλα τα δεδομένα εισόδου πριν παράγουν οποιοδήποτε αποτέλεσμα.
- Λόγω περιορισμών στην απόδοση και τον αποθηκευτικό χώρο η οπισθοδρόμηση σε μια ροή δεδομένων δεν είναι δυνατή. Οι on-line αλγόριθμοι ροών δεδομένων περιορίζονται σε ένα και μόνο πέρασμα στα δεδομένα.
- Οι εφαρμογές παρακολούθησης ροών δεδομένων σε πραγματικό χρόνο πρέπει να αντιδρούν γρήγορα σε ασυνήθιστες τιμές δεδομένων.
- Οι μακροπρόθεσμες επερωτήσεις μπορεί να συναντήσουν μεταβολές στις συνθήκες του συστήματος (π.χ. μεταβλητό ρυθμό ροής).
- Απαιτείται κοινή εκτέλεση πολλών συνεχών επερωτήσεων για τη διασφάλιση της δυνατότητας κλιμάκωσης.

Τα προτεινόμενα συστήματα διαχείρισης ροών δεδομένων ακολουθούν την αρχιτεκτονική του Σχήματος 5. Ένας ελεγκτής εισόδου (input monitor) μπορεί να σταθεροποιήσει τους ρυθμούς εισόδου, απορρίπτοντας ενδεχομένως κάποια πακέτα. Τα δεδομένα αποθηκεύονται σε τρία τμήματα: στον προσωρινό χώρο αποθήκευσης εργασίας (working storage) (π.χ. για επερωτήσεις παραθύρων), στο χώρο αποθήκευσης για τις συνόψεις των ροών δεδομένων (summary storage), και στον στατικό χώρο αποθήκευσης (static storage) για τα μετα-δεδομένα (όπως π.χ. είναι η φυσική θέση για κάθε πηγή δεδομένων).



Σχήμα 5: Αρχιτεκτονική συστήματος διαχείρισης ροών δεδομένων

Οι μακροπρόθεσμες επερωτήσεις καταχωρούνται στην αποθήκη επερωτήσεων (query repository) και τοποθετούνται σε ομάδες για κοινή επεξεργασία, ενώ μπορούν επίσης να υποβληθούν και στιγμιαίες επερωτήσεις σχετικά με την τρέχουσα κατάσταση της ροής. Ο επεξεργαστής επερωτήσεων (query processor) επικοινωνεί με τον ελεγκτή εισόδου και μπορεί να βελτιστοποιήσει εκ νέου το σχεδιασμό των επερωτήσεων ως απάντηση στη μεταβολή των ρυθμών εισόδου των δεδομένων. Τα αποτελέσματα προωθούνται στους χρήστες ή αποθηκεύονται προσωρινά. [12,14]

2.3 Υλοποιημένα συστήματα διαχείρισης ροών δεδομένων

Το συνεχώς αυξανόμενο ερευνητικό ενδιαφέρον γύρω από τη διαχείριση και επεξεργασία ροών δεδομένων έχει οδηγήσει στην υλοποίηση ενός αριθμού από συστήματα DSMS από πανεπιστημιακά ιδρύματα και εταιρείες. Τα πιο γνωστά συστήματα διαχείρισης ροών δεδομένων είναι τα εξής [13]:

- *Aurora*: Αποτελεί μια συνεργασία των πανεπιστημίων Brandeis, Brown και MIT. Είναι ένα σύστημα για εφαρμογές (πραγματικού χρόνου, αρχειοθέτησης ή συνδυασμού των δύο) παρακολούθησης αισθητήρων και ροών δεδομένων [15].

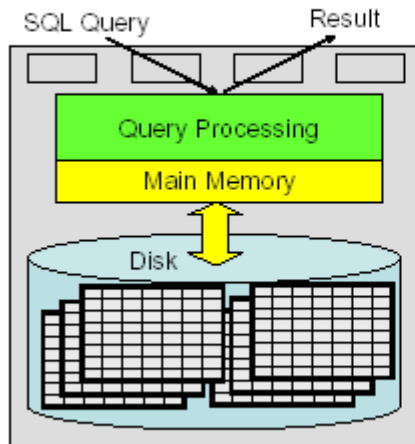
- *COUGAR*: Αναπτύχθηκε στο πανεπιστήμιο Cornell και ακολουθεί μια προσέγγιση κατανεμημένης βάσης δεδομένων για δίκτυα αισθητήρων χρησιμοποιώντας βελτιστοποιήσεις μεταξύ διαφορετικών επιπέδων [16].
- *GigaScope*: Είναι προϊόν έρευνας της εταιρείας AT&T και παρακολουθεί ταχύρυθμες εφαρμογές, ενσωματώνοντας μια βάση δεδομένων για ροές [17].
- *Hancock*: Αναπτύχθηκε στα ερευνητικά εργαστήρια της εταιρείας AT&T και είναι μια γλώσσα βασισμένη στη C που έχει ως σκοπό την εύκολη ανάγνωση, γραφή και διατήρηση προγραμμάτων που χειρίζονται μεγάλες ποσότητες δεδομένων [18].
- *NiagaraCQ*: Είναι το αποτέλεσμα της συνεργασίας των πανεπιστημίων του Wisconsin και του Oregon (OGI). Αποτελεί ένα σύστημα συνεχών επερωτήσεων για βάσεις δεδομένων στο Διαδίκτυο και μπορεί να χρησιμοποιηθεί για την συλλογή, παρακολούθηση και πραγματοποίηση επερωτήσεων σε XML δεδομένα [19].
- *OpenCQ*: Αναπτύχθηκε στο πανεπιστήμιο Georgia Tech και είναι ένα σύστημα συνεχών επερωτήσεων για την παρακολούθηση ροών web περιεχομένου [20].
- *StatStream*: Είναι ένα σύστημα παρακολούθησης πολλαπλών ροών για on-line στατιστική ανάλυση υψηλού ρυθμού, που αναπτύχθηκε στο πανεπιστήμιο της Νέας Υόρκης [21].
- *STREAM*: Αναπτύχθηκε στο πανεπιστήμιο Stanford και αποτελεί ένα σχεσιακό σύστημα διαχείρισης ροών δεδομένων και επεξεργασίας επερωτήσεων [22].
- *Streaminer*: Αποτελεί ερευνητικό πρόγραμμα του πανεπιστημίου του Illinois για την εξόρυξη δεδομένων από ροές.
- *Tapestry*: Έχει αναπτυχθεί στο ερευνητικό κέντρο που έχει ιδρύσει η εταιρεία Xerox στο Palo Alto και αποτελεί ένα σύστημα που φιλτράρει την ηλεκτρονική αλληλογραφία χρησιμοποιώντας συνεχείς επερωτήσεις [23].

- *TelegraphCQ*: Αποτελεί ένα ερευνητικό πρόγραμμα του πανεπιστημίου του Berkeley και είναι ένα προσαρμοζόμενο σύστημα διαχείρισης ροών δεδομένων και επεξεργασίας συνεχών επερωτήσεων [24].
- *Tribeca*: Έχει αναπτυχθεί στα ερευνητικά εργαστήρια Bellcore και αποτελεί ένα σύστημα διαχείρισης ροών δεδομένων για ανάλυση της κίνησης δικτύων [25].

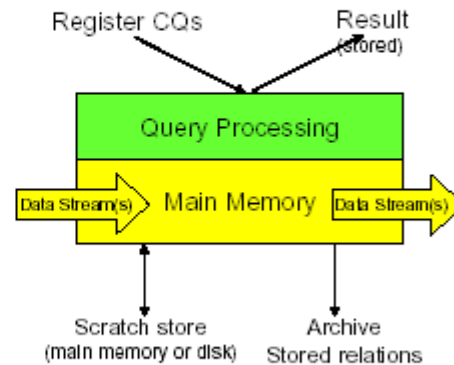
2.4 Μοντέλα ροών δεδομένων και παραθύρων

Η θεμελιώδης διαφορά ανάμεσα σε ένα κλασσικό σύστημα DBMS και σε ένα σύστημα DSMS είναι το μοντέλο ροών δεδομένων. Σύμφωνα με το συγκεκριμένο μοντέλο τα δεδομένα εισόδου (στο σύνολό τους ή ένα μέρος αυτών), τα οποία πρόκειται να επεξεργαστεί το σύστημα δεν είναι διαθέσιμα για τυχαία προσπέλαση από τον σκληρό δίσκο ή τη μνήμη του. Αντίθετα, όπως φαίνεται και στο Σχήμα 6, φτάνουν υπό τη μορφή μιας ή περισσότερων ροών δεδομένων on-line, παραμένουν για περιορισμένο χρόνο στη μνήμη και το σύστημα DSMS πρέπει να διαχειριστεί τα δεδομένα που αποθηκεύονται στον προσωρινό καταχωρητή (buffer) πριν διαγραφούν από νέα εισερχόμενα στοιχεία. Κατά συνέπεια, η προσπέλαση των δεδομένων είναι σειριακή και τα δεδομένα μπορούν να διαβαστούν μία ή λίγες φορές. Επιπλέον, το σύστημα δεν μπορεί να ελέγξει τη σειρά άφιξης των δεδομένων εντός μιας ροής ή μεταξύ περισσότερων ροών. Όταν ολοκληρωθεί η επεξεργασία ενός στοιχείου της ροής δεδομένων, το σύστημα το απορρίπτει ή το αποθηκεύει. Το στοιχείο δεν μπορεί να ανακτηθεί εύκολα, εκτός αν αποθηκευτεί στη μνήμη του συστήματος, η οποία είναι περιορισμένη σε σχέση με το μέγεθος των ροών δεδομένων [26],[27].

Traditional DBMS



DSMS



Σχήμα 6: Υλοποίηση DBMS και DSMS

Εφόσον τα στοιχεία της ροής φτάνουν κατά ριπές, μια ροή δεδομένων μπορεί να μοντελοποιηθεί σαν μια ακολουθία που αποτελείται από λίστες στοιχείων. Κάθε στοιχείο της ροής μπορεί να πάρει τη μορφή σχεσιακής πλειάδας ή στιγμιότυπου ενός αντικειμένου. Σε σχεσιακά μοντέλα (όπως π.χ. το STREAM), τα στοιχεία είναι βραχύβιες πλειάδες που αποθηκεύονται σε εικονικές σχέσεις, πιθανώς διαμοιρασμένες οριζόντια σε απομακρυσμένους κόμβους. Σε μοντέλα που βασίζονται σε αντικείμενα (όπως π.χ. το COUGAR και το Tribeca) οι πηγές και τα στοιχεία μοντελοποιούνται σαν ιεραρχικοί τύποι δεδομένων με συσχετιζόμενες μεθόδους.

Τα στοιχεία μιας ροής μπορεί να φτάσουν εκτός σειράς και/ή σε προεπεξεργασμένη μορφή περιγράφοντας ένα υποκείμενο σήμα. Ας θεωρήσουμε στην πιο απλή περίπτωση ένα σήμα a ως μια μονοδιάστατη συνάρτηση $a : [0 \dots (N-1)] \rightarrow Z^+$, δηλαδή το πεδίο ορισμού είναι διακριτό και διατεταγμένο και η συνάρτηση το αντιστοιχεί σε μη αρνητικούς ακεραίους. Έχουμε τότε την ακόλουθη λίστα πιθανών μοντέλων:

- *Μοντέλο ταμειακής μηχανής¹ με στοιχεία σε μη διατεταγμένη μορφή (unordered cash register):* Στοιχεία από διαφορετικά πεδία ορισμού φτάνουν χωρίς συγκεκριμένη σειρά και χωρίς προεπεξεργασία.
- *Μοντέλο ταμειακής μηχανής με στοιχεία σε διατεταγμένη μορφή (ordered cash register):* Μεμονωμένα στοιχεία από διαφορετικά πεδία ορισμού φτάνουν με κάποια γνωστή σειρά χωρίς προεπεξεργασία.
- *Μοντέλο συνάθροισης με στοιχεία σε μη διατεταγμένη μορφή (unordered aggregation):* Μεμονωμένα στοιχεία από το ίδιο πεδίο ορισμού έχουν υποστεί προεπεξεργασία, ώστε να φτάνει τελικά μόνο ένα στοιχείο ανά πεδίο ορισμού χωρίς συγκεκριμένη σειρά.
- *Μοντέλο συνάθροισης με στοιχεία σε διατεταγμένη μορφή (ordered aggregation):* Μεμονωμένα στοιχεία από το ίδιο πεδίο ορισμού έχουν υποστεί προεπεξεργασία, ώστε να φτάνει τελικά μόνο ένα στοιχείο ανά πεδίο ορισμού με κάποια γνωστή σειρά [14].

Για παράδειγμα ένα σήμα είναι η διάρκεια σε λεπτά των εξερχόμενων τηλεφωνικών κλήσεων από ένα τηλεφωνικό αριθμό (το πεδίο ορισμού είναι το σύνολο όλων των τηλεφωνικών αριθμών και το εύρος είναι η συνολική διάρκεια εξερχόμενων κλήσεων σε λεπτά). Η ροή δεδομένων μπορεί να περιγράψει το υποκείμενο σήμα με διαφορετικούς τρόπους ανάλογα με το μοντέλο δεδομένων που έχει επιλεγεί. Στην περίπτωση του μοντέλου ταμειακής μηχανής με στοιχεία σε μη διατεταγμένη μορφή, τα στοιχεία είναι τιμές από το πεδίο ορισμού και η συνάρτηση αναπαριστάται αθροίζοντας ρητά το σύνολο των στοιχείων με την ίδια τιμή στο πεδίο ορισμού. Στο παράδειγμα των τηλεφωνικών κλήσεων η ροή θα μπορούσε να είναι:

<8008001111,10>, <8008002222,15>, <8008003333,13>, <8008001111,23>, <8008001111,3>, ...

Το υποκείμενο σήμα θα ήταν:

<8008001111,36>, <8008002222,15>, <8008003333,13>

¹ Το μοντέλο ταμειακής μηχανής πήρε το όνομα αυτό επειδή το σύνολο των πωλήσεων μέσω μιας ταμειακής μηχανής σε ένα κατάστημα παράγει μια ροή δεδομένων που ανήκει στο συγκεκριμένο μοντέλο.

και θα μπορούσε να κατασκευαστεί αθροίζοντας τον συνολικό αριθμό της διάρκειας των εξερχόμενων κλήσεων από τους αριθμούς 8008001111, 8008002222, 8008003333 κλπ.

Στο μοντέλο συνάθροισης με στοιχεία σε μη διατεταγμένη μορφή τα στοιχεία που φτάνουν είναι οι τιμές εύρους χωρίς συγκεκριμένη σειρά και συνεπώς το σήμα μπορεί να αποδοθεί ρητά. Έτσι, στο παράδειγμα των τηλεφωνικών κλήσεων έχουμε:

<8008001111,36>, <8008003333,13>, <8008002222,15>

Στο μοντέλο ταμειακής μηχανής με στοιχεία σε διατεταγμένη μορφή η ροή θα έχει την ακόλουθη μορφή:

<8008001111,10>, <8008001111,23>, <8008001111,3>, <8008002222,15>, <8008003333,13>, ...

Τέλος, στο μοντέλο συνάθροισης με στοιχεία σε διατεταγμένη μορφή η ροή θα έχει την εξής μορφή:

<8008001111,36>, <8008002222,15>, <8008003333,13>

Οι ροές δεδομένων από διαφορετικές περιοχές ακολουθούν διαφορετικά μοντέλα. Π.χ. μια χρονική ακολουθία δεδομένων είναι πιθανό να ακολουθεί το μοντέλο συνάθροισης με στοιχεία σε διατεταγμένη μορφή, ενώ η ποσότητα των δεδομένων σε ένα δίκτυο είναι πιθανόν να ακολουθεί το μοντέλο ταμειακής μηχανής με στοιχεία σε μη διατεταγμένη μορφή. Το συγκεκριμένο μοντέλο είναι το πιο γενικό, θέτοντας τις περισσότερες προκλήσεις κατά τον σχεδιασμό αλγορίθμων επεξεργασίας δεδομένων [28].

Σε πολλές περιπτώσεις μόνο ένα τμήμα μιας ροής έχει ενδιαφέρον σε μια δεδομένη χρονική στιγμή, καθιστώντας σημαντικά τα μοντέλα παραθύρων, που μπορούν να κατηγοριοποιηθούν σύμφωνα με τα ακόλουθα τρία κριτήρια:

1. *Κατεύθυνση της κίνησης των άκρων του παραθύρου*: Δύο σταθερά άκρα ορίζουν ένα σταθερό παράθυρο (*fixed window*), δύο κινούμενα άκρα ορίζουν ένα ολισθαίνον παράθυρο (*sliding window*), ενώ ένα σταθερό και ένα κινούμενο άκρο (με κατεύθυνση προς τα εμπρός ή προς τα πίσω) ορίζουν ένα παράθυρο οριοθέτησης (*landmark window*). Στο σύνολο υπάρχουν εννιά συνδυασμοί που προκύπτουν από τη δυνατότητα για

κάθε άκρο να είναι σταθερό ή να κινείται προς μια από τις δύο κατευθύνσεις.

2. *Φυσικά έναντι λογικών παραθύρων*: Τα φυσικά (*physical*) ή βασισμένα στον χρόνο (*time-based*) παράθυρα καθορίζονται με βάση κάποιο χρονικό διάστημα, ενώ τα λογικά (*logical*) ή αλλιώς παράθυρα βασισμένα στις πλειάδες (*count-based* ή *tuple-based*) ορίζονται με βάση τον αριθμό των πλειάδων που περιλαμβάνουν.
3. *Διάστημα ενημερώσεων*: Υπάρχει η δυνατότητα ενημέρωσης του παραθύρου με την άφιξη κάθε νέας πλειάδας ή στην περίπτωση της ομαδικής επεξεργασίας προκύπτουν τα παράθυρα άλματος (*jumping windows*). Αν το διάστημα ενημέρωσης είναι μεγαλύτερο από το μέγεθος του παραθύρου το αποτέλεσμα είναι μια ακολουθία από μη επικαλυπτόμενα κυλιόμενα παράθυρα (*tumbling windows*) [14].

2.5 Τύποι επερωτήσεων

Οι επερωτήσεις σε συνεχείς ροές δεδομένων έχουν αρκετά κοινά στοιχεία με τις επερωτήσεις στα παραδοσιακά συστήματα διαχείρισης βάσεων δεδομένων (DBMS). Ωστόσο υπάρχουν δύο σημαντικοί διαχωρισμοί στους τύπους επερωτήσεων, που οφείλονται στα ιδιαίτερα χαρακτηριστικά του μοντέλου ροών δεδομένων. Ο πρώτος διαχωρισμός γίνεται ανάμεσα στις στιγμιαίες επερωτήσεις (*one-time queries*) και στις συνεχείς επερωτήσεις (*CQ-Continuous Queries*). Οι στιγμιαίες επερωτήσεις (κατηγορία που περιλαμβάνει τις παραδοσιακές επερωτήσεις στα συστήματα DBMS) υποβάλλονται μία φορά σε ένα στιγμιότυπο των δεδομένων. Οι συνεχείς επερωτήσεις υποβάλλονται κατ' επανάληψη καθώς τα δεδομένα φτάνουν διαρκώς. Οι απαντήσεις στις συνεχείς επερωτήσεις παράγονται με βάση τη ροή των δεδομένων που έχει φτάσει ως τη στιγμή υποβολής της επερώτησης. Τα αποτελέσματα αυτής της κατηγορίας επερωτήσεων μπορούν να αποθηκευτούν και να ανανεώνονται με την άφιξη καινούργιων δεδομένων ή μπορούν να αποτελέσουν νέες ροές δεδομένων. Ο δεύτερος διαχωρισμός γίνεται μεταξύ των προκαθορισμένων και των *ad hoc* επερωτήσεων. Οι προκαθορισμένες επερωτήσεις παρέχονται στο σύστημα πριν την άφιξη των

δεδομένων. Συνήθως σε αυτή την κατηγορία ανήκουν οι συνεχείς επερωτήσεις, όμως και οι στιγμιαίες επερωτήσεις είναι δυνατόν να προκαθοριστούν. Αντιθέτως, οι ad hoc επερωτήσεις υποβάλλονται on-line αφού έχει ξεκινήσει η ροή των δεδομένων. Τόσο οι στιγμιαίες όσο και οι συνεχείς επερωτήσεις μπορεί να είναι ad hoc. Ωστόσο αυτός ο τύπος επερωτήσεων καθιστά πιο πολύπλοκο τον σχεδιασμό ενός συστήματος διαχείρισης ροών δεδομένων (DSMS), αφ' ενός γιατί οι επερωτήσεις δεν είναι γνωστές εκ των προτέρων και συνεπώς δεν υπάρχει δυνατότητα βελτιστοποίησης τους και αφετέρου γιατί η απάντηση σε μια ad hoc επερώτηση ενδέχεται να απαιτεί στοιχεία από τη ροή δεδομένων που έχουν ήδη φτάσει και πιθανώς απορριφθεί από το σύστημα [13],[26].

2.6 Σημασιολογία συνεχών επερωτήσεων

Οποιαδήποτε μονότονη (monotonic) συνεχής επερώτηση που μπορεί να ενημερωθεί αυξητικά, είναι δυνατόν να υλοποιηθεί ως μια συνεχής επερώτηση σε μια παραδοσιακή βάση δεδομένων. Σε μια βάση δεδομένων μόνο για προσάρτηση (append-only database), όλες οι συζευκτικές επερωτήσεις είναι μονότονες: μόλις προστεθεί μια πλειάδα, είτε ικανοποιεί την επερώτηση ή όχι και η συνθήκη που πρέπει να ικανοποιείται δεν αλλάζει με το χρόνο. Αντίθετα, η προσθήκη άρνησης μπορεί να παραβιάσει τη μονοτονία (π.χ. η επιλογή από μια ροή μηνυμάτων ηλεκτρονικού ταχυδρομείου όλων εκείνων των μηνυμάτων που ακόμα δεν έχουν λάβει απάντηση). Ωστόσο, σε ορισμένες περιπτώσεις επερωτήσεων η άρνηση είναι δυνατόν να αποφευχθεί με κατάλληλη διατύπωση. Παρόμοια, εάν η βάση δεδομένων δεν είναι μόνο για προσάρτηση, τότε καμία ερώτηση δεν είναι μονότονη, καθώς οι ενημερωμένες πλειάδες μπορούν να πάψουν να ικανοποιούν μια δεδομένη επερώτηση.

Λιγότερο περιοριστική σημασιολογία μονότονων και μη μονότονων συνεχών επερωτήσεων σε δεδομένα έχει προταθεί [29]. Αν θεωρήσουμε για λόγους απλότητας ότι ο χρόνος αναπαριστάται σαν ένα σύνολο από φυσικούς αριθμούς και ότι όλες οι συνεχείς επερωτήσεις επαναξιολογούνται με κάθε χτύπο του ρολογιού, έστω ότι $A(Q,t)$ είναι το σύνολο απαντήσεων μιας συνεχούς επερώτησης τη χρονική στιγμή t και τ είναι ο τρέχων χρόνος που

αρχίζει από το 0. Το σύνολο απαντήσεων μιας μονότονης συνεχούς επερώτησης Q τη χρονική στιγμή τ είναι:

$$A(Q, \tau) = \bigcup_{t=1}^{\tau} (A(Q, t) - A(Q, t-1)) \cup A(Q, 0) \quad (1)$$

Δηλαδή, αρκεί η επαναξιολόγηση της επερώτησης στα νέο-αφιχθέντα στοιχεία και η προσάρτηση των πλειάδων που πληρούν τις προϋποθέσεις στο αποτέλεσμα. Αντίθετα, οι μη μονότονες επερωτήσεις μπορεί να χρειάζεται να υπολογιστούν εκ νέου κατά τη διάρκεια της επαναξιολόγησης υποστηρίζοντας την ακόλουθη σημασιολογία [14]:

$$A(Q, \tau) = \bigcup_{t=0}^{\tau} A(Q, t) \quad (2)$$

2.7 Γλώσσες επερωτήσεων

Έχουν προταθεί τρεις γλώσσες επερωτήσεων για συστήματα διαχείρισης ροών δεδομένων. Τα σχεσιακά συστήματα χρησιμοποιούν γλώσσες παρόμοιες με την SQL, για επερωτήσεις σχέσεων που έχουν χρονικές ενδείξεις (timestamps), παρέχοντας συνήθως αυξημένη υποστήριξη σε παράθυρα και ταξινόμηση. Οι γλώσσες που βασίζονται σε αντικείμενα μοιάζουν επίσης με τη γλώσσα SQL, αλλά συμπεριλαμβάνουν υποστήριξη για αφηρημένους τύπους δεδομένων σε ροές (ADT) και σχετικές μεθόδους επεξεργασίας σήματος. Τα διαδικαστικά συστήματα συνθέτουν επερωτήσεις ορίζοντας ροές δεδομένων μέσα από διάφορους τελεστές. Στη συνέχεια περιγράφονται οι τρεις ομάδες γλωσσών και δίνονται παραδείγματα επερωτήσεων για να γίνουν εμφανείς οι διαφορές τους, ενώ κάποια από τα

παραδείγματα χρησιμοποιούν απλοποιημένη σύνταξη για να γίνουν πιο κατανοητά.

2.7.1 Γλώσσες βασισμένες σε σχέσεις

Υπάρχουν τρεις γλώσσες βασισμένες στις σχέσεις (relation-based languages), η CQL, η StreaQueL και η AQuery. Η CQL (Continuous Query Language) χρησιμοποιείται στο σύστημα STREAM και περιλαμβάνει ολισθαίνοντα παράθυρα και τελεστές που μεταφράζουν σχέσεις σε ροές. Είναι δυνατή η λειτουργία PARTITION σε ένα παράθυρο με βάση ένα πεδίο καθώς και ο καθορισμός του εύρους ενός παραθύρου (π.χ. ROWS 100 ή RANGE 100 MINUTES). Για παράδειγμα αν μια ροή S περιέχει εγγραφές τηλεφωνικών κλήσεων, η ακόλουθη επερώτηση υπολογίζει το μέσο μήκος των δέκα πιο πρόσφατων υπεραστικών κλήσεων για κάθε πελάτη:

```
SELECT AVG (S.call_length) FROM S [PARTITION BY S.customer_id  
ROWS 10 WHERE S.type = 'Long Distance']
```

Οι επερωτήσεις σε ολόκληρες ροές δεδομένων μπορούν να επιλέξουν μεταξύ [UNBOUNDED] ή [NOW] στον τύπο του παραθύρου, με την τελευταία επιλογή να χρησιμοποιείται σε μονότονες επερωτήσεις (π.χ. επιλογές) που δε χρειάζεται να λαμβάνουν υπόψη προηγούμενα δείγματα. Επιπλέον, υπάρχουν τρεις τελεστές σχέσης προς ροή, που μπορούν να χρησιμοποιηθούν για να καθορίσουν ρητά τη σημασιολογία των επερωτήσεων (όπως ορίζεται στις ισότητες (1) και (2)). Ακόμη, ο ρυθμός δειγματοληψίας μπορεί να καθοριστεί ρητά ακολουθώντας την αναφορά σε μια ροή με τη δήλωση π.χ. 10% SAMPLE.

Η StreaQueL, που είναι η γλώσσα επερωτήσεων του συστήματος TelegraphCQ παρέχει εξελιγμένες δυνατότητες δημιουργίας παραθύρων. Κάθε ορισμός επερώτησης ακολουθείται από μία δομή βρόχου for με μια μεταβλητή t που επαναλαμβάνεται με τον χρόνο. Ο βρόχος περιέχει μια δήλωση Windows που καθορίζει τον τύπο και το μέγεθος του παραθύρου. Έστω μια ροή S, και ST η στιγμή εκκίνησης της επερώτησης. Για τον καθορισμό ενός ολισθαίνοντος παραθύρου στη ροή S με μέγεθος πέντε που

θα εκτελείται για πενήντα ημέρες θα προσαρτηθεί στη ροή ο ακόλουθος βρόχος for:

```
for (t=ST; t<ST+50; t++)
```

```
Windows (S, t-4, t)
```

Η μετάβαση σε ένα παράθυρο οριοθέτησης θα μπορούσε να γίνει αντικαθιστώντας το t-4 με κάποια σταθερά στη δήλωση Windows. Αλλάζοντας τη συνθήκη αύξησης σε t=t+5 θα προκαλούσε την επερώτηση να εκτελείται κάθε πέντε χρονικές μονάδες.

Η γλώσσα AQuery αποτελείται από μια άλγεβρα επερωτήσεων και μια γλώσσα βασισμένη στην SQL για ταξινομημένα δεδομένα. Οι στήλες των πινάκων θεωρούνται πίνακες (arrays), στους οποίους μπορούν να εφαρμοστούν οι βασιζόμενοι στη σειρά τελεστές όπως οι: next, prev, first και last. Για παράδειγμα, μια συνεχής επερώτηση σε μια ροή από τιμές μετοχών που αναφέρει διαδοχικές διαφορές τιμών της μετοχής π.χ. της εταιρείας IBM μπορεί να οριστεί ως εξής:

```
SELECT price - prev (price) FROM Trades ASSUMING ORDER  
timestamp WHERE company = 'IBM'
```

Η φράση ASSUMING ORDER ορίζει το πεδίο ταξινόμησης του πίνακα. Η εφαρμογή της συγκεκριμένης επερώτησης σε συμβατικές ακολουθιακές γλώσσες απαιτεί μια σύζευξη της σχέσης Trades με ένα αντίγραφο της, το οποίο έχει ολισθήσει κατά μία θέση.

2.7.2 Γλώσσες βασισμένες σε αντικείμενα

Μια προσέγγιση στην μοντελοποίηση αντικειμενοστρεφών ροών δεδομένων είναι η ταξινόμηση των περιεχομένων της ροής σύμφωνα με μια ιεραρχία τύπων. Η μέθοδος αυτή χρησιμοποιείται στο σύστημα Tribeca για την παρακολούθηση δικτύων, που υλοποιεί επίπεδα πρωτοκόλλων του Διαδικτύου σαν ιεραρχικούς τύπους δεδομένων. Μια άλλη δυνατότητα είναι η μοντελοποίηση των πηγών σαν τύπους ADT, όπως στο σύστημα COUGAR για τη διαχείριση δεδομένων από αισθητήρες. Κάθε τύπος αισθητήρα μοντελοποιείται με έναν ADT, του οποίου η διεπαφή αποτελείται από

μεθόδους επεξεργασίας σήματος που υποστηρίζει ο συγκεκριμένος τύπος δεδομένων. Η προτεινόμενη γλώσσα επερωτήσεων έχει σύνταξη που μοιάζει με την SQL και περιλαμβάνει επίσης μια δήλωση \$every() που δείχνει τη συχνότητα επανεκτέλεσης της επερώτησης. Για παράδειγμα μια επερώτηση που εκτελείται κάθε εξήντα δευτερόλεπτα και επιστρέφει τις μετρήσεις θερμοκρασίας από όλους τους αισθητήρες στον τρίτο όροφο ενός κτηρίου θα μπορούσε να οριστεί ως εξής [14]:

```
SELECT R.s.getTemperature() FROM R WHERE R.floor = 3 AND $every(60)
```

2.7.3 Διαδικαστικές γλώσσες

Μια εναλλακτική πρόταση στις δηλωτικές (declarative) γλώσσες επερωτήσεων είναι να καθορίσει ο χρήστης τον τρόπο ροής των δεδομένων στο σύστημα. Στο σύστημα Auora οι χρήστες κατασκευάζουν σχέδια επερωτήσεων μέσω μιας γραφικής διεπαφής τοποθετώντας κουτιά (που αντιστοιχούν σε τελεστές επερωτήσεων), τα οποία ενώνουν με κατευθυνόμενα τόξα για να ορίσουν τη ροή των δεδομένων παρόλο που το σύστημα μπορεί στη συνέχεια να αλλάξει τη σειρά, να προσθέσει ή να αφαιρέσει τελεστές κατά τη φάση της βελτιστοποίησης. Το σύστημα Auora περιλαμβάνει αρκετούς τελεστές που δεν ορίζονται ρητά σε άλλες γλώσσες. Ο τελεστής map εφαρμόζει μια λειτουργία σε κάθε στοιχείο, (ο τελεστής αυτός ορίζεται και στη γλώσσα AQuery, όπου ονομάζεται each). Ο τελεστής resample παρεμβάλλει τιμές στοιχείων που λείπουν σε ένα παράθυρο, ενώ ο drop απορρίπτει στοιχεία με τυχαίο τρόπο αν ο ρυθμός εισόδου είναι πολύ υψηλός [14].

2.8 Το σύστημα TelegraphCQ

Το σύστημα TelegraphCQ είναι εξέλιξη του ερευνητικού προγράμματος Telegraph το οποίο είχε ξεκινήσει στις αρχές του 2000 στο πανεπιστήμιο του Berkeley, με στόχο την ανάπτυξη μιας προσαρμοζόμενης αρχιτεκτονικής για ροές δεδομένων, η οποία θα υποστήριζε ένα ευρύ σύνολο δικτυακών εφαρμογών με έμφαση στα δεδομένα. Η πρώτη έκδοση του Telegraph

υποστήριζε συνενώσεις γεγονότων και σχημάτων (Federated Facts & Figures - FFF) και ήταν ένα σύστημα επερωτήσεων για web δεδομένα. Η αρχική αυτή έκδοση του Telegraph δεν έκανε καμία προσπάθεια να εκμεταλλευτεί τα κοινά στοιχεία μεταξύ ταυτόχρονα ενεργών επερωτήσεων. Αντίθετα, εστίαζε σε ένα σενάριο με ένα μόνο χρήστη, παρέχοντας αποδοτικά επιμέρους αποτελέσματα σε επερωτήσεις με αλληλεπιδραστικό έλεγχο από τον χρήστη.

Στη συνέχεια αναπτύχθηκαν δύο επεκτάσεις του αρχικού συστήματος με τις ονομασίες CACQ και PSOUP για την υποστήριξη κοινής επεξεργασίας σε ροές δεδομένων. Ωστόσο τα συστήματα αυτά παρουσίαζαν ορισμένους σημαντικούς περιορισμούς:

- Μπορούσαν να επεξεργαστούν μόνο δεδομένα που βρίσκονταν στη μνήμη του συστήματος.
- Δεν διερευνούσαν ζητήματα χρονοπρογραμματισμού και διαχείρισης πόρων για επερωτήσεις με μικρές ή καθόλου επικαλύψεις.
- Δεν αντιμετώπιζαν την ανάγκη για ποιότητα υπηρεσίας (QoS) με προσαρμογή στους περιορισμούς πόρων.
- Δεν ερευνούσαν δυνατότητες μεταβολής του βαθμού προσαρμοστικότητας για τη διαπραγμάτευση ευελιξίας και καθυστερήσεων.

Με βάση τα αρχικά αυτά συστήματα έχει επανασχεδιαστεί και υλοποιηθεί ένα νέο σύστημα που ονομάστηκε TelegraphCQ και το οποίο δίνει πλέον έμφαση στην υποστήριξη επεξεργασίας κοινών, συνεχών επερωτήσεων σε ροές δεδομένων. Το σύστημα αυτό μπορεί να διαχειριστεί μεγάλες ροές συνεχών επερωτήσεων σε ροές δεδομένων υψηλής μεταβλητότητας και μεγάλου όγκου. Το TelegraphCQ έχει ως σκοπό την επίλυση των προβλημάτων που αντιμετώπιζαν τα προηγούμενα συστήματα. Συγκεκριμένα εστιάζει στα εξής ζητήματα:

- Χρονοπρογραμματισμός και διαχείριση πόρων για ομάδες επερωτήσεων
- Υποστήριξη δεδομένων εκτός μνήμης
- Μεταβλητή προσαρμοστικότητα

- Δυναμική υποστήριξη QoS
- Παράλληλη επεξεργασία βασισμένη σε ομάδες και καταμερισμό εργασιών

Παρά το γεγονός ότι το σύστημα στο αρχικό ερευνητικό πρόγραμμα Telegraph είχε υλοποιηθεί στη γλώσσα προγραμματισμού Java, αποφασίστηκε για το σύστημα TelegraphCQ να χρησιμοποιηθεί το σύστημα βάσης δεδομένων ανοιχτού κώδικα (open source) PostgreSQL. Αυτό συνέβη, διότι η εμπειρία από τη χρήση της γλώσσας Java για την ανάπτυξη ενός συστήματος δεν ήταν θετική, παρόλο που η έρευνα γύρω από την προσαρμοστικότητα και τον καταμερισμό παρουσίασε σημαντικά οφέλη. Ένα σύστημα συνεχών επερωτήσεων, όπως το TelegraphCQ, διαφέρει αρκετά από έναν παραδοσιακό επεξεργαστή επερωτήσεων. Ωστόσο, ένα σημαντικό μέρος του κώδικα PostgreSQL μπορούσε εύκολα να επαναχρησιμοποιηθεί. Επιπλέον, τα χαρακτηριστικά επεκτασιμότητας του συστήματος PostgreSQL αποδείχθηκαν πολύ χρήσιμα, ιδιαίτερα η δυνατότητα να χρησιμοποιηθεί κώδικας καθορισμένος από τον χρήστη καθώς και οι πολλοί τύποι δεδομένων όπως τα διαστήματα (intervals) [30],[31].

Για τις ανάγκες της παρούσας διπλωματικής εργασίας έχει χρησιμοποιηθεί η έκδοση v0.2 της πλατφόρμας TelegraphCQ. Ωστόσο, υπάρχουν διαθέσιμες και οι εκδόσεις v2.0 και v2.1.

2.9 Σημασιολογία παραθύρων στο σύστημα TelegraphCQ

Το σύστημα TelegraphCQ υποστηρίζει συνεχείς επερωτήσεις σε ένα συνδυασμό από πίνακες και ροές δεδομένων. Για τη διαχείριση ροών δεδομένων, των οποίων το μήκος είναι απεριόριστο, ορισμένες λειτουργίες, όπως η σύζευξη, μπορούν να πραγματοποιηθούν μόνο σε πεπερασμένα παράθυρα στις ροές αυτές. Προκειμένου να υποστηρίξει ένα μεγάλο πλήθος από τύπους επερωτήσεων, το σύστημα TelegraphCQ παρέχει πλούσια παραθυρικά σχήματα, τόσο στο τμήμα της ροής που έχει ήδη φτάσει, όσο και σε τμήματα που θα φτάσουν στο μέλλον. Επιπλέον, το σύστημα προσφέρει ευέλικτους μηχανισμούς παράδοσης των παραγόμενων αποτελεσμάτων από

τις επερωτήσεις που υποβάλλονται στα παράθυρα αυτά. Δηλαδή, επιτρέπει την άμεση προώθηση στον χρήστη των αποτελεσμάτων της εκτέλεσης επερωτήσεων σε διαδοχικά παράθυρα (όπως στο σύστημα CACQ) ή την παρουσίαση των αποτελεσμάτων μόνο έπειτα από απαίτηση του ίδιου του χρήστη (όπως στο σύστημα PSOUP).

2.9.1 Παράθυρα σε ροές εισόδου

Δύο δημοφιλή σχήματα παραθύρων, στα πλαίσια της επεξεργασίας επερωτήσεων σε ροές δεδομένων, είναι τα παράθυρα οριοθέτησης και τα ολισθαίνοντα παράθυρα. Για τις επερωτήσεις με παράθυρα οριοθέτησης, το χρονικά παλαιότερο άκρο του παραθύρου είναι σταθερό, ενώ το χρονικά νεότερο άκρο του κινείται με κατεύθυνση προς τα εμπρός όταν φτάνουν νέες εγγραφές στη ροή. Αντίθετα, στις επερωτήσεις με ολισθαίνοντα παράθυρα, και τα δύο άκρα κινούνται ταυτόχρονα με κατεύθυνση προς τα εμπρός όταν φτάνουν νέες εγγραφές. Ωστόσο, ή σημασιολογία που προσφέρουν τα παράθυρα οριοθέτησης και τα ολισθαίνοντα παράθυρα, καλύπτει μόνο ένα μικρό μέρος των ενδιαφερόντων εφαρμογών σε ροές δεδομένων.

Για παράδειγμα, τα παράθυρα οριοθέτησης και τα ολισθαίνοντα παράθυρα δε μπορούν να συλλάβουν τη σημασιολογία μιας επερώτησης που εκτελείται με την άφιξη κάθε η πλειάδων, ούτε μπορούν να περιγράψουν παράθυρα με δεδομένα από το παρελθόν. Σε ένα σύστημα πλοήγησης ο χρήστης μπορεί να θελήσει να υποβάλει μια επερώτηση σχετικά με ένα τμήμα μιας ροής, το οποίο ανήκει στο παρελθόν, χρησιμοποιώντας παράθυρα που κινούνται προς τα πίσω ξεκινώντας από την παρούσα στιγμή. Τα παραδοσιακά παράθυρα οριοθέτησης και τα ολισθαίνοντα παράθυρα δεν μπορούν να χρησιμοποιηθούν σε τέτοιες εφαρμογές.

Η σημασιολογία των επερωτήσεων στο σύστημα TelegraphCQ είναι η ακόλουθη. Για κάθε χρονική στιγμή, ένα παράθυρο σε μια ροή καθορίζει ένα σύνολο πλειάδων στις οποίες θα εκτελεστεί η επερώτηση. Δεδομένου ότι κάθε εκτέλεση της επερώτησης παράγει ένα σύνολο, η έξοδος μιας επερώτησης παρουσιάζεται στον τελικό χρήστη ως μια ακολουθία συνόλων όπου κάθε σύνολο συσχετίζεται με μια χρονική στιγμή. Το σύστημα TelegraphCQ

επιτρέπει πολλαπλές ταυτόχρονες έννοιες του χρόνου, όπως μια λογική ακολουθία αριθμών ή ο φυσικός χρόνος. Προκειμένου να συμπεριληφθούν ασαφώς συγχρονισμένες κατανεμημένες πηγές δεδομένων, μεταχειριζόμαστε το χρόνο ως μια μερική διάταξη, παρά ως πλήρη διάταξη. Έχει σχεδιαστεί μια άλγεβρα που επεκτείνει τους υπάρχοντες σχεσιακούς τελεστές, ώστε να λειτουργούν σε ροές και η οποία επιτρέπει σε μια ροή που έχει οριστεί με χρήση μιας χρονικής έννοιας να μετασχηματιστεί σε μια ροή που χρησιμοποιεί μια άλλη χρονική έννοια.

Το σύστημα TelegraphCQ υποστηρίζει πολύ πιο γενικά παράθυρα από τα παράθυρα οριοθέτησης και τα ολισθαίνοντα παράθυρα. Αυτό γίνεται χρησιμοποιώντας μια δομή βρόχου `for` για τη δήλωση της ακολουθίας των παραθύρων στα οποία ο χρήστης επιθυμεί τις απαντήσεις σε μια επερώτηση. Μια μεταβλητή `t` κινείται στον χρονικό άξονα, καθώς ο βρόχος `for` επαναλαμβάνεται, ενώ το αριστερό και το δεξί άκρο κάθε παραθύρου στην ακολουθία και η συνθήκη τερματισμού για την επερώτηση μπορούν να οριστούν σε σχέση με αυτή τη μεταβλητή.

2.9.2 Επίδραση της σημασιολογίας παραθύρων στο σχεδιασμό του συστήματος TelegraphCQ

Οι διαφορετικοί τύποι παραθύρων μπορούν να θέσουν σημαντικά διαφορετικές απαιτήσεις από τον σχεδιασμό του επεξεργαστή επερωτήσεων και του υποκείμενου διαχειριστή αποθήκευσης. Ένα βασικό ζήτημα σχετίζεται με τη χρήση λογικών (δηλαδή με βάση τον αριθμό της ακολουθίας πλειάδων) έναντι φυσικών (δηλαδή με βάση τον πραγματικό χρόνο) χρονικών ενδείξεων (timestamps). Στην πρώτη περίπτωση, οι απαιτήσεις ενός παραθύρου σε μνήμη μπορούν να είναι γνωστές εκ των προτέρων, ενώ στην δεύτερη περίπτωση, οι απαιτήσεις σε μνήμη θα εξαρτηθούν από τις διακυμάνσεις στον ρυθμό άφιξης δεδομένων.

Ένα άλλο ζήτημα που σχετίζεται με τις απαιτήσεις σε μνήμη, είναι ο τύπος παραθύρου που χρησιμοποιείται σε μια επερώτηση. Ας θεωρήσουμε την περίπτωση της εκτέλεσης μιας MAX συνάθροισης σε μια ροή. Για ένα παράθυρο οριοθέτησης, είναι δυνατό να υπολογιστεί η απάντηση

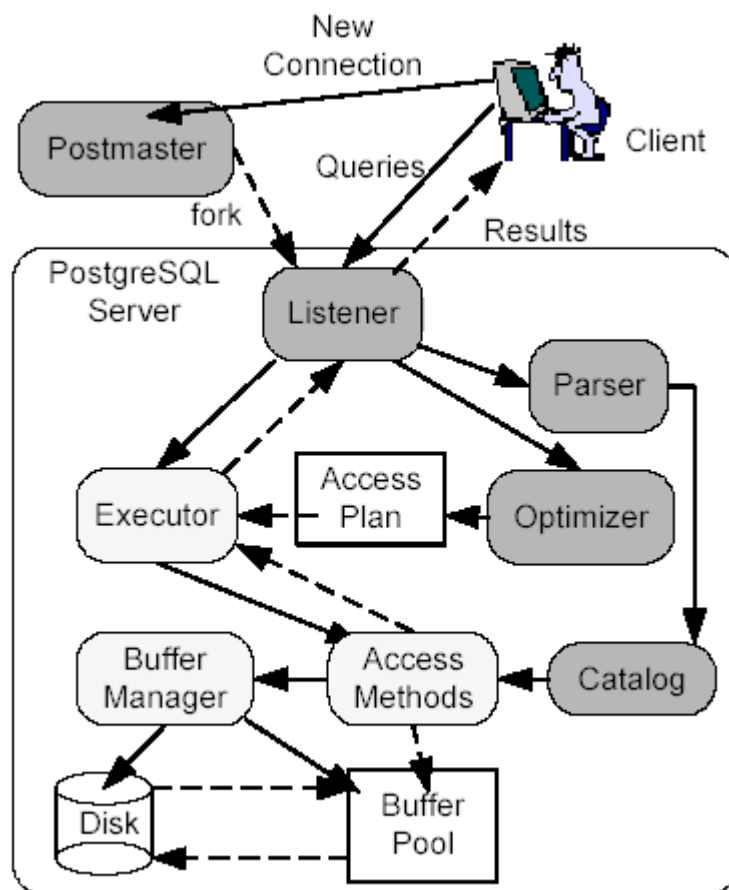
επαναληπτικά, συγκρίνοντας απλά το τρέχον μέγιστο με το νεώτερο στοιχείο καθώς το παράθυρο επεκτείνεται. Αντιθέτως, για ένα ολισθαίνον παράθυρο, ο υπολογισμός του μέγιστου απαιτεί τη διατήρηση ολόκληρου του παραθύρου.

Τέλος, η κατεύθυνση της κίνησης και το μέγεθος "άλματος" των παραθύρων (η απόσταση μεταξύ διαδοχικών παραθύρων που καθορίζεται από τον βρόχο for) έχουν επίσης σημαντική επίδραση στην εκτέλεση επερωτήσεων. Για παράδειγμα, αν το μέγεθος άλματος του παραθύρου ξεπεράσει το μέγεθος του ίδιου του παραθύρου, τότε κάποια μέρη της ροής δεν περιλαμβάνονται ποτέ στην επεξεργασία της επερώτησης [30].

2.10 Η αρχιτεκτονική του συστήματος TelegraphCQ

Στη συνέχεια περιγράφεται η αρχιτεκτονική λογισμικού του συστήματος TelegraphCQ δίνοντας αρχικά έμφαση στον τρόπο προσαρμογής της αρχιτεκτονικής του συστήματος PostgreSQL, ώστε να δοθεί η δυνατότητα κοινής επεξεργασίας των συνεχών επερωτήσεων σε πηγές ροών δεδομένων. Έπειτα περιγράφονται τα νέα στοιχεία που συνθέτουν το σύστημα TelegraphCQ.

Το σύστημα TelegraphCQ έχει υλοποιηθεί στις γλώσσες προγραμματισμού C/C++, ενώ ταυτόχρονα έχει χρησιμοποιηθεί η βάση κώδικα του ανοιχτού λογισμικού PostgreSQL. Στο Σχήμα 7 φαίνεται η βασική δομή λειτουργίας του συστήματος PostgreSQL. Το PostgreSQL χρησιμοποιεί ένα μοντέλο διεργασίας-ανά-σύνδεση. Οι δομές δεδομένων, που μοιράζονται μεταξύ πολλαπλών διεργασιών βρίσκονται στην κοινή μνήμη. Μια διεργασία Postmaster δημιουργεί νέες διεργασίες εξυπηρετητή ως απάντηση σε νέες συνδέσεις πελατών. Μεταξύ των διαφορετικών στοιχείων κάθε διεργασίας εξυπηρετητή, ο Ακροατής (Listener) είναι υπεύθυνος για την αποδοχή των αιτήσεων σε μια σύνδεση και την επιστροφή των δεδομένων στον πελάτη. Όταν μια καινούργια επερώτηση φθάσει, αναλύεται, βελτιστοποιείται, και μεταγλωττίζεται σε ένα σχέδιο προσπέλασης, ενώ στη συνέχεια την επεξεργάζεται το στοιχείο Διεκπεραιωτής επερώτησης (Executor).

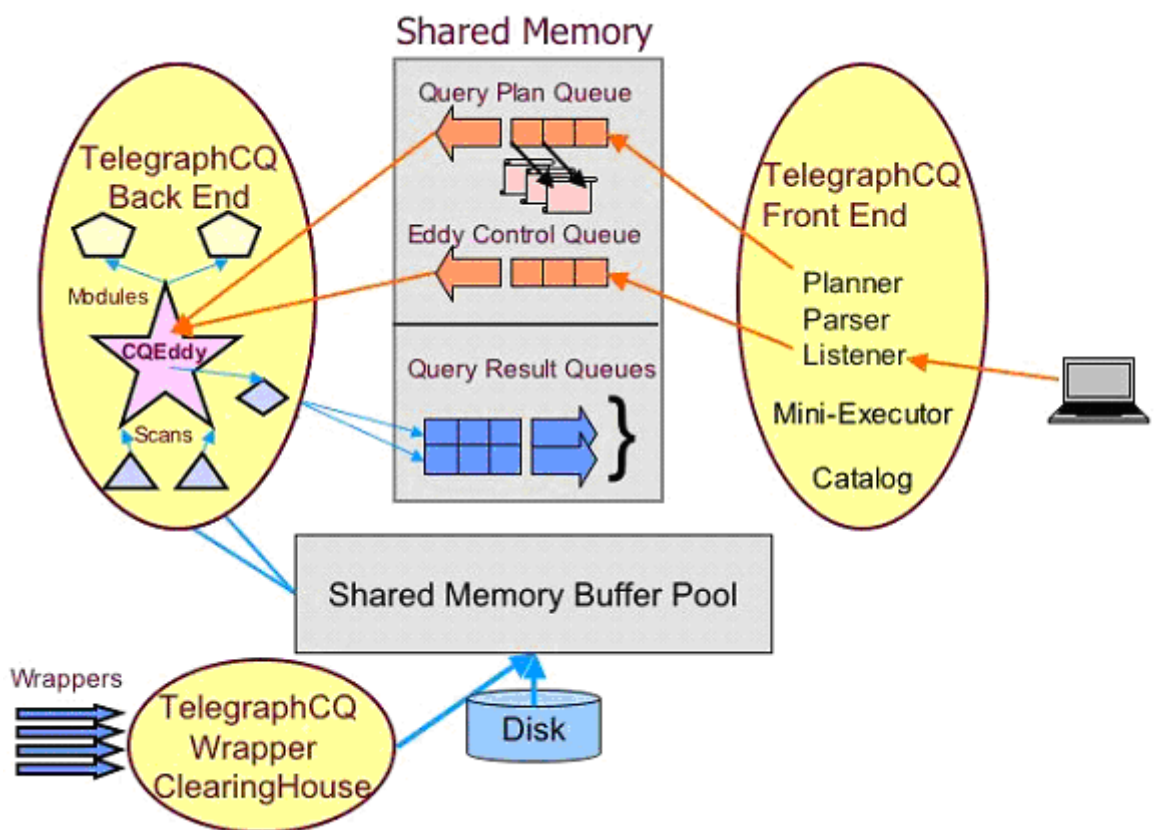


Σχήμα 7: Η αρχιτεκτονική του συστήματος PostgreSQL

Τα στοιχεία που έχουν αλλάξει ελάχιστα στο σύστημα TelegraphCQ είναι: ο Postmaster, ο Ακροατής (Listener), ο Κατάλογος Συστήματος (System Catalog), ο Αναλυτής Επερωτήσεων (Query Parser) και ο Βελτιστοποιητής Επερωτήσεων (Query Optimizer). Τα στοιχεία που έχουν υποστεί σημαντικές αλλαγές είναι: ο Διεκπεραιωτής (Executor), ο Διαχειριστής Ενδιάμεσης Μνήμης (Buffer Manager) και οι Μέθοδοι Προσπέλασης (Access Methods). Επιπλέον, με την υιοθέτηση των front-end στοιχείων του συστήματος PostgreSQL, το TelegraphCQ έχει πρόσβαση στις υλοποιήσεις διεπαφής σε επίπεδο κλήσης από την πλευρά του πελάτη (που δεν φαίνονται στο Σχήμα 7) όπως οι ODBC και JDBC.

Η κύρια πρόκληση στη χρήση του συστήματος PostgreSQL ήταν η υποστήριξη εκείνων των χαρακτηριστικών του TelegraphCQ για τα οποία δεν ήταν σχεδιασμένο όπως: οι ροές δεδομένων, οι συνεχείς επερωτήσεις, η

κοινή επεξεργασία και η προσαρμοστικότητα. Η μεγαλύτερη επίδραση των αλλαγών είναι στο μοντέλο διεργασία-ανά-σύνδεση του PostgreSQL. Στο σύστημα TelegraphCQ υπάρχει μια αποκλειστική διεργασία, που ονομάζεται Back End, για την εκτέλεση κοινών, μακροπρόθεσμων, συνεχών επερωτήσεων. Η διεργασία αυτή, μαζί με τη διεργασία ανά-σύνδεση Front End του συστήματος TelegraphCQ, δέχεται τις επερωτήσεις από τον πελάτη και επιστρέφει τα αποτελέσματα σε αυτόν. Η διεργασία Front End εξυπηρετεί επίσης μη συνεχείς επερωτήσεις και δηλώσεις DDL. Επιπλέον, το σύστημα TelegraphCQ διαθέτει μια αποκλειστική διεργασία που ονομάζεται Wrapper ClearingHouse και η οποία διασφαλίζει ότι οι λειτουργίες εισόδου των δεδομένων δεν εμποδίζουν την πρόοδο της εκτέλεσης επερωτήσεων.



Σχήμα 8: Η αρχιτεκτονική του TelegraphCQ

Το Σχήμα 8 απεικονίζει την αρχιτεκτονική του συστήματος TelegraphCQ. Με οβάλ σχήμα φαίνονται οι τρεις διεργασίες (Back End, Front End και Wrapper ClearingHouse) που περιλαμβάνει ο εξυπηρετητής του TelegraphCQ. Οι διεργασίες αυτές επικοινωνούν χρησιμοποιώντας μια υποδομή κοινής μνήμης. Η διεργασία Front End περιέχει τον ακροατή, τον κατάλογο, τον αναλυτή, τον σχεδιαστή (Planner) και τον "μίνι-διεκπαιωτή" ("mini-Executor"). Η πραγματική επεξεργασία επερωτήσεων πραγματοποιείται στην ξεχωριστή διεργασία Back End. Τέλος, η διεργασία Wrapper ClearingHouse χρησιμοποιείται για να φιλοξενήσει τους τελεστές εισόδου των δεδομένων, οι οποίοι καθιστούν τις νέες πλειάδες διαθέσιμες για επεξεργασία, αρχειοθετώντας τις αν είναι απαραίτητο. Η διεργασία Wrapper ClearingHouse είναι υπεύθυνη για τις ακόλουθες λειτουργίες:

- Αποδοχή συνδέσεων από εξωτερικές πηγές και λήψη του ονόματος της ροής που θα στείλουν.
- Φόρτωση των κατάλληλων συναρτήσεων από τους wrappers που έχει ορίσει ο χρήστης.
- Κλήση αυτών των wrapper συναρτήσεων όταν τα δεδομένα είναι διαθέσιμα σε μια σύνδεση.
- Επεξεργασία των πλειάδων που επιστρέφουν ως αποτέλεσμα από έναν wrapper σύμφωνα με τον τύπο της ροής.
- Εκκαθάριση όταν μια πηγή τερματίζει την αλληλεπίδρασή της με το σύστημα TelegraphCQ.

Όπως και στο σύστημα PostgreSQL, έτσι και στο TelegraphCQ ο Postmaster «ακούει» σε μια γνωστή θύρα και δημιουργεί μια Front End διεργασία για κάθε μια νέα σύνδεση που λαμβάνει. Ο ακροατής δέχεται εντολές από έναν πελάτη και βασισμένος σε αυτές, επιλέγει πού να τις εκτελέσει. Οι δηλώσεις DDL και οι επερωτήσεις σε πίνακες εκτελούνται στην ίδια τη διεργασία Front End. Οι συνεχείς επερωτήσεις, που περιλαμβάνουν μόνο ροές δεδομένων καθώς επίσης και εκείνες που περιλαμβάνουν ροές δεδομένων και πίνακες, "προσχεδιάζονται" και στέλνονται μέσω της ουράς Σχεδίου Επερώτησης (Query Plan) (που βρίσκεται στην κοινή μνήμη) στη διεργασία Back End. Ο διεκπαιωτής Back End βγάζει συνεχώς από την

ουρά νέες επερωτήσεις και τις προσθέτει δυναμικά στην τρέχουσα εκτελούμενη επερώτηση. Στη συνέχεια τα αποτελέσματα της επερώτησης τοποθετούνται σε ξεχωριστές ουρές Αποτελέσματος Επερώτησης (Query Result) για κάθε πελάτη. Όταν η διεργασία Front End δώσει μια επερώτηση στη διεργασία Back End, παράγει ένα ελάχιστο τοπικό σχέδιο επερώτησης, το οποίο ο μίνι-επεξεργαστής εκτελεί για να βγάξει συνεχώς αποτελέσματα από την ουρά Αποτελέσματος Επερώτησής του και να τα επιστρέφει στον συνδεδεμένο πελάτη.

Δεδομένου ότι μια συνεχής επερώτηση δεν τελειώνει ποτέ, οι πελάτες θα πρέπει να υποβάλουν τέτοιες επερωτήσεις ως τμήμα καθορισμένων δρομέων. Έχει προστεθεί ένας τρόπος λειτουργίας συνεχούς επερώτησης στον `rsq1`, τον τυπικό αλληλεπιδραστικό πελάτη του συστήματος PostgreSQL. Με τον τρόπο αυτό, οι δηλώσεις `SELECT` μετατρέπονται αυτόματα σε καθορισμένους δρομείς που μπορούν στη συνέχεια να επαναλαμβάνονται ώστε να επιστρέφουν συνεχώς αποτελέσματα [30],[31].

2.11 Χρήση του συστήματος TelegraphCQ

Εκτός από τα χαρακτηριστικά που προσφέρει το ίδιο το σύστημα PostgreSQL, η αλληλεπίδραση των χρηστών με το σύστημα TelegraphCQ περιλαμβάνει τα εξής:

- Δημιουργία αρχειοθετημένων και μη αρχειοθετημένων ροών.
- Δημιουργία πηγών που στέλνουν δεδομένα στο σύστημα TelegraphCQ.
- Δημιουργία `wrappers` καθορισμένων από τους χρήστες.
- Υποβολή συνεχών επερωτήσεων.

Οι ροές που δημιουργούνται στο σύστημα καθορίζουν αντικείμενα στα οποία μπορούν να υποβληθούν επερωτήσεις, και δεδομένα τα οποία μπορεί να επεξεργαστεί το σύστημα και πιθανώς να τα αρχειοθετήσει. Οι πηγές δεδομένων αποτελούν ανεξάρτητα προγράμματα που στέλνουν συνεχώς δεδομένα σε μια ροή με καθορισμένο όνομα στο σύστημα TelegraphCQ. Για κάθε ροή υπάρχει ένας `wrapper` ορισμένος από τον χρήστη που δηλώνεται

στο TelegraphCQ και είναι σχεδιασμένος έτσι ώστε να κατανοεί τα δεδομένα που στέλνει η πηγή. Τέλος, χρήστες και εφαρμογές μπορούν να συνδεθούν στο σύστημα ώστε να υποβάλλουν συνεχείς επερωτήσεις στις ροές αυτές [31].

2.11.1 Δημιουργία ροών

Στο σύστημα TelegraphCQ, μπορούν να δημιουργηθούν και να διαγραφούν ροές δεδομένων χρησιμοποιώντας τις νέες DDL δηλώσεις CREATE STREAM και DROP STREAM αντίστοιχα. Οι δηλώσεις αυτές είναι παρόμοιες με εκείνες που χρησιμοποιούνται για τη δημιουργία και τη διαγραφή πινάκων. Μια απαίτηση για τον ορισμό μιας ροής είναι ότι πρέπει να υπάρχει ακριβώς μια χρονική στήλη, η οποία εξυπηρετεί ως χρονική ένδειξη για τη ροή. Η στήλη αυτή ορίζεται με τον περιορισμό κλειδιού TIMESTAMPCOLUMN. Έτσι, ο καθορισμός μιας αρχειοθετημένης ροής, που χρησιμοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας για τις μετρήσεις με το TelegraphCQ γίνεται με την ακόλουθη εντολή:

```
CREATE STREAM temp.streams (Node INT, airTemp REAL, tcqtime  
TIMESTAMP TIMESTAMPCOLUMN) TYPE ARCHIVED;
```

Η παραπάνω ροή δεδομένων αποτελείται από: τον αριθμό του κόμβου αισθητήρα που πραγματοποιεί τη μέτρηση, τη θερμοκρασία αέρα που μετρά ο αισθητήρας και την ώρα μέτρησης. Πριν τη δημιουργία της ροής πρέπει να έχει δημιουργηθεί το σχήμα (schema) temp με την εντολή:

```
CREATE SCHEMA temp;
```

2.11.2 Δημιουργία πηγών

Το σύστημα TelegraphCQ περιμένει ότι μια πηγή δεδομένων θα ξεκινήσει μια σύνδεση δικτύου με αυτό και θα του δώσει το όνομα της ροής στην οποία προτίθεται να παρέχει δεδομένα. Περισσότερες από μία πηγές μπορούν να παρέχουν ταυτόχρονα δεδομένα στην ίδια ροή. Το όνομα της ροής μεταδίδεται στα πρώτα bytes που στέλνει η πηγή στο σύστημα έπειτα από την εγκαθίδρυση της σύνδεσης. Δεν υπάρχουν άλλοι περιορισμοί ως προς τη

μορφή των δεδομένων της ροής, μιας και όλες οι επακόλουθες δικτυακές λειτουργίες της σύνδεσης είναι ευθύνη της λειτουργίας του καθορισμένου από τον χρήστη wrapper, που συνδέεται με τη ροή.

Για τις ανάγκες της διπλωματικής εργασίας τροποποιήθηκε το παράδειγμα που βρίσκεται στον κατάλογο TelegraphCQ-0.2/src/test/examples/tcqsanity. Η πηγή δεδομένων που χρησιμοποιείται στο συγκεκριμένο παράδειγμα είναι το αρχείο source.pl. Ωστόσο, για να παρέχει δεδομένα στη ροή που έχουμε δημιουργήσει θα πρέπει να προσθέσουμε στο αρχείο push_streams.sh τις εξής εντολές:

```
TEMP_SCHEMA= temp
STREAMS_STREAM=streams
STREAMS_NAME="$CSV_WRAP,$TEMP_SCHEMA.$STREAMS_STREAM"
echo 'sending temp.streams'
cat streamsdata.log | ./source.pl $HOST $PORT $STREAMS_NAME
```

Όπου streamsdata.log είναι το αρχείο με τα δεδομένα που θα τροφοδοτήσουν τη ροή temp.streams.

2.11.3 Δημιουργία wrappers

Κάθε wrapper αποτελείται από τρεις συναρτήσεις ορισμένες από τον χρήστη (τις init, next και done), τις οποίες καλεί το σύστημα TelegraphCQ για να επεξεργαστεί τα δεδομένα από τις εξωτερικές πηγές. Οι συναρτήσεις δηλώνονται στο σύστημα με τη χρήση της τυπικής δήλωσης CREATE FUNCTION της PostgreSQL. Ωστόσο, εκτός από τους wrappers που μπορούν να ορίσουν οι χρήστες, στο TelegraphCQ υπάρχει ένας wrapper ο οποίος είναι διαθέσιμος για χρήση μετά την εγκατάσταση του συστήματος. Ονομάζεται csv και υποστηρίζει δεδομένα που φτάνουν οριοθετημένα, χωρίζοντας κάθε στήλη με κόμμα, ενώ οι πλειάδες χωρίζονται με αλλαγή γραμμής. Τα χαρακτηριστικά αυτά του csv wrapper μπορούν να αλλάξουν με τη χρήση των παραμέτρων ρύθμισης του wrapper. Επιπλέον, ο αριθμός στηλών σε ένα στοιχείο δεν χρειάζεται απαραίτητα να είναι ίσος με τον αριθμό των στηλών που έχει οριστεί για τη ροή. Οι επιπλέον στήλες των πλειάδων θα

περικοπούν, ενώ στην περίπτωση που οι τιμές σε κάποιες στήλες λείπουν, θα προστεθούν σε αυτές NULL τιμές. Ωστόσο, κάθε ροή δεδομένων πρέπει να έχει το ειδικό πεδίο `TIMESTAMP` ως το τελευταίο πεδίο της, το οποίο θα χρησιμοποιηθεί από τον διεκπεραιωτή του συστήματος `TelegraphCQ`. Ο `wrapper` που έχει χρησιμοποιηθεί για την επεξεργασία των ροών δεδομένων για τις ανάγκες της διπλωματικής εργασίας είναι ο `csv wrapper`. Η σύνδεσή του με τη ροή `temp.streams` γίνεται με την εντολή:

```
ALTER STREAM temp.streams ADD WRAPPER csvwrapper;
```

2.11.4 Συνεχείς επερωτήσεις

Όταν οι ροές δεδομένων έχουν δημιουργηθεί στο σύστημα, οι χρήστες μπορούν να υποβάλλουν μακροπρόθεσμες συνεχείς επερωτήσεις σε αυτές. Μια επερώτηση θεωρείται ότι είναι συνεχής εάν υποβάλλεται σε μία ή περισσότερες ροές. Τέτοιου είδους επερωτήσεις δε θα σταματήσουν αν δεν τις τερματίσει ο χρήστης. Οι επερωτήσεις μπορεί να είναι απλές SQL επερωτήσεις (χωρίς υπο-ερωτήσεις), με προαιρετική σύνταξη που να περιέχει παράθυρο (`window`). Το χρονικό διάστημα που ορίζεται σαν συμβολοσειρά (`string`) εντός της σύνταξης του παραθύρου, υπάρχει δυνατότητα να είναι οτιδήποτε μπορεί το σύστημα `PostgreSQL` να μετατρέψει σε τύπο διαστήματος (`interval`). Η σύνταξη παραθύρου εξυπηρετεί στον περιορισμό της ποσότητας των δεδομένων που συμμετέχουν στην επερώτηση με την πάροδο του χρόνου. Αυτό είναι ιδιαίτερα σημαντικό για επερωτήσεις σύζευξης και συνάθροισης, δεδομένου ότι οι ροές δεδομένων είναι απεριόριστες. Οι επερωτήσεις μπορούν να υποβληθούν στο `TelegraphCQ` με τη χρήση του αλληλεπιδραστικού πελάτη `psql`. Είναι επίσης δυνατόν να χρησιμοποιηθούν και άλλες προγραμματιστικές διεπαφές, όπως οι `ODBC` και `JDBC`, αλλά οι εφαρμογές που τις χρησιμοποιούν πρέπει να έχουν δηλώσει καθορισμένους δρομείς για τις συνεχείς επερωτήσεις. Σε διαφορετική περίπτωση η διεπαφή επιπέδου κλήσης αποθηκεύει προσωρινά τα αποτελέσματα μιας επερώτησης και σταματά μέχρι να ληφθούν όλα τα δεδομένα. Ένα παράδειγμα επερώτησης η οποία χρησιμοποιήθηκε για τις μετρήσεις απόδοσης της διπλωματικής εργασίας είναι το εξής:

```
SELECT AVG (tr.airtemp) FROM temp.streams tr GROUP BY tr.Node  
WINDOW tr ['2 minutes'];
```

Η επερώτηση αυτή υποβάλλεται στη ροή δεδομένων temp.streams και επιστρέφει τη μέση τιμή της θερμοκρασίας του αέρα σε ένα ολισθαίνον παράθυρο 2 λεπτών.