Πετράκης Κωνσταντίνος *ΑΜ: 1041589* *ΥΔΑ*

Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual

Η παρούσα εργασία συγκρίνει μία SQL βάση δεδομένων, την PostgreSQL, με δύο NoSQL βάσεις, την Cassandra και την MongoDB, ως προς την απόδοση τους στην διαχείριση δεδομένων που έχουν προκύψει από αισθητήρες (sensor data). Οι βάσεις δεδομένων συγκρίνονται ως προς την απόδοση τους σε 4 λειτουργίες: single write (όπου μία μέτρηση εισάγεται στη βάση), single read (μια μέτρηση διαβάζεται από τη βάση), multiple writes (1000 μετρήσεις εισάγονται στη βάση) και multiple reads (1000 μετρήσεις διαβάζονται από την βάση), ενώ γίνεται σύγκριση και ως προς την λειτουργία αυτών των βάσεων σε πραγματικούς φυσικούς διακομιστές (physical server) και εικονικές μηχανές (virtual machine).

Αρχικά επειδή παρατηρείται πως η χρήση ευρετηρίων για τις βάσεις MongoDB και PostgreSQL έχει θεαματικά αποτελέσματα στην απόδοση των βάσεων όσο αφορά το διάβασμα δεδομένων και ελάχιστη επιβάρυνση στο γράψιμο, υιοθετείται από τους συγγραφείς η χρήση τους σε όλα τα πειράματα.

Στην περίπτωση όπου ένας client εκτελεί ένα γράψιμο η MongoDB έχει με διαφορά την καλύτερη απόδοση, η Cassandra έχει μέτρια απόδοση και η PostgreSQL πολύ χαμηλή απόδοση. Όταν ένας client εκτελεί ένα διάβασμα η διάταξη ως προς την απόδοση των βάσεων παραμένει ίδια αλλά οι διαφορές είναι τώρα πολύ μικρότερες. Η χρήση εικονικής μηχανής επηρεάζει όμοια όλες τις βάσεις όσον αφορά το διάβασμα ενώ στα γραψίματα η εικονική μηχανή είναι γρηγορότερη από τον φυσικό server μόνο στη περίπτωση της PostgreSQL.

Όταν ένας client εκτελεί πολλαπλά γραψίματα η Cassandra έχει με διαφορά την καλύτερη απόδοση με τις MongoDB και PostgreSQL να ακολουθούν. Στην περίπτωση όπου ένας client εκτελεί πολλαπλά διαβάσματα η PostgreSQL αποκτά προβάδισμα με την MongoDB να είναι πολύ κοντά και την Cassandra να αποδίδει πολύ άσχημα όσο ο αριθμός των λειτουργιών αυξάνεται. Εδώ η διαφορά μεταξύ φυσικού server και εικονικής μηχανής είναι ελάχιστη, με την Cassandra να φαίνεται πως επηρεάζεται λίγο περισσότερο από το virtualization.

Όταν πολλαπλοί client εκτελούν από μία εγγραφή η MongoDB είναι πολύ γρήγορη για ένα πελάτη αλλά επιβραδύνει αργά όσο προστίθενται πελάτες. Η Cassandra αρχικά επωφελείται από την αύξηση των clients αλλά στην συνέχεια η απόδοση της πέφτει όταν προστεθούν πολλοί clients ενώ η PostgreSQL επωφελείται ελάχιστα από την αύξηση των clients. Όταν πολλαπλοί clients εκτελούν από ένα διάβασμα και οι 3 βάσεις επωφελούνται από την αύξηση των clients μέχρι το πλήθος τους να φτάσει στο 16. Η διαφορά στην απόδοση μεταξύ φυσικού server και εικονικής μηχανής είναι αξιοσημείωτη και για τις δυο λειτουργίες, διάβασμα και γράψιμο Ο φυσικός server κερδίζει στην περίπτωση γραψίματος στις MongoDB και Cassandra αλλά η εικονική μηχανή είναι γρηγορότερη για το γράψιμο στη PostgreSQL. Η επίδραση στο διάβασμα είναι τεράστια για τις MongoDB και Cassandra με την εικονική μηχανή να είναι 10 φορές πιο αργή από τον φυσικό server.

Όταν πολλαπλοί clients εκτελούν πολλαπλές εγγραφές μόνο η PostgreSQL επωφελείται από την αύξηση των clients με την απόδοση των MongoDB και Cassandra να παραμένει σχετικά σταθερή καθώς το πλήθος των client αυξάνει. Στην περίπτωση που πολλαπλοί clients εκτελούν πολλαπλά διαβάσματα, η αύξηση των clients ωφελεί σημαντικά τις Cassandra και PostgreSQL, ενώ από την άλλη η απόδοση της MongoDB παραμένει σταθερή. Η επίδραση του virtualization στο γράψιμο είναι χαμηλή ενώ για το διάβασμα έχει την ίδια αρνητική επίδραση στις PostgreSQL και MongoDB αλλά πολύ θετική επίδραση στην απόδοση της Cassandra.

Τέλος, υπό το πρίσμα πως η χρήση αισθητήρων απαιτεί πολλαπλές εγγραφές μικρών τμημάτων δεδομένων και διάβασμα μεγάλων τμημάτων δεδομένων, οι συγγραφείς καταλήγουν πως καμία από τις βάσεις που εξετάστηκαν δεν υπερισχύει και ως προς τις 2 αυτές απαιτήσεις.

The Extensibility Framework in Microsoft StreamInsight

Η παρούσα εργασία περιγράφει ένα πλαίσιο επέκτασης στο StreamInsight με στόχο την ενσωμάτωση οριζόμενων από το χρήστη λειτουργιών σε ένα σύστημα επεξεργασίας ροών (stream), ώστε να δοθεί η δυνατότητα στους domain experts να επεκτείνουν και να εφαρμόζουν το σύστημα σε πολλαπλά περιβάλλοντα. Το StreamInsight αποτελεί μια πλατφόρμα της Microsoft για την ανάπτυξη streaming εφαρμογών οι οποίες απαιτούν την εκτέλεση συνεχόμενων queries (CQ) σε ροές υψηλού ρυθμού δεδομένων εισερχόμενων συμβάντων (events). Στην εργασία αναλύονται τα προβλήματα που προέκυψαν και οι αλλαγές που έγιναν εξαιτίας της επέκτασης από τρεις οπτικές: από τη σκοπιά του συντάκτη των ερωτημάτων (query writer), από τη σκοπιά του συντάκτη του οριζόμενου από το χρήστη προγράμματος (user defined module ή αλλιώς UDM) και από τη σκοπιά του εσωτερικών λειτουργιών του συστήματος.

Η αρχιτεκτονική του framework επέκτασης έχει σχεδιαστεί έτσι ώστε να ικανοποιεί αυτές τις αρχές σχεδιασμού που θα συμβάλουν στην ευελιξία, την αποτελεσματικότητα και την απρόσκοπτη ενσωμάτωση ενός ευρέος φάσματος καθοριζόμενων από το χρήστη modules στον αγωγό συνεχούς επεξεργασίας ερωτημάτων. Πιο συγκεκριμένα απαιτείται ευελεξία και προσαρμοστικότητα, ευκολία χρήσης, φορητότητα, συμβατότητα, αποδοτικότητα, σπάσιμο των ορίων βελτιστοποίηση και ‘ζωντάνια’ (liveliness).

Αρχικά αναφορικά με τον συντάκτη του query επειδή ένα μόνο UDM μπορεί να κληθεί από εκατοντάδες queries, ο ρόλος του συντάκτη του query σχεδιάστηκε να είναι τόσο απλός όσο μια κλήση μεθόδου και τόσο ευέλικτος όσο η εκτέλεση του ίδιου UDM κάτω από διαφορετικές προδιαγραφές παραθύρων και διαφορετικές πολιτικές εισόδου/εξόδου. Κύριοι στόχοι του συντάκτη του query παραμένουν η ευκολία στη χρήση, δηλαδή όλες οι λεπτομέρειες υλοποίησης του UDM να παραμένουν κρυφές, και η ευελιξία, δηλαδή ο έλεγχος της συμπεριφοράς του UDM από τον συντάκτη του query μέσω των παραμέτρων προσδιορισμού των παραθύρων και της πολιτικής χρονικής σήμανσης εισόδου/εξόδου.

Στην συνέχεια όσον αφορά τον συντάκτη του UDM, στους απλούς χρήστες παρέχεται η δυνατότητα να γράφουν ισχυρά UDMs χωρίς να ανησυχούν για τους ειδικούς για stream τύπους συμβάντων (π.χ. insertion, retractions). Ειδικότερα οι συντάκτες των UDM έχουν την δυνατότητα να μεταφέρουν βιβλιοθήκες που έχουν δημιουργήσει για παραδοσιακά συστήματα βάσεων δεδομένων με ελάχιστο κόπο. Αυτό επιτυγχάνεται με την διατήρηση της σχεσιακής προβολής των δεδομένων, την διαχείριση της χρονικής διάσταση εκ μέρους του συντάκτη του UDM και τον χειρισμό παράδοσης ατελών συμβάντων που αντιπροσωπεύονται από αργοπορημένες αφίξεις συμβάντων από την μεριά του συντάκτη του UDM. Από την άλλη στους προηγμένους συντάκτες UDM παρέχεται η ευελιξία να διαχειρίζονται την χρονική πτυχή των συμβάντων εισόδου και εξόδου στον κώδικα τους. Για το σκοπό αυτό παρέχεται εξουσιοδότηση στα UDM να διαβάζουν την χρονική διάσταση των συμβάντων εισόδου (δηλ. τα όρια των events LE και RE), δίνεται στα UDM η δυνατότητα χρονικής σήμανσης των συμβάντων εξόδου τους και παρέχεται υποστήριξη της σταδιακής αξιολόγησης των αποτελεσμάτων τους.

Τέλος σχετικά με τις εσωτερικές λειτουργίες του συστήματος οι συγγραφείς αναφέρονται στον τρόπο με τον οποίο το σύστημα φιλοξενεί ένα UDM, στον τρόπο με τον οποίο το σύστημα απαλλάσσει τον συντάκτη ενός UDM από λειτουργίες ειδικά για stream και πως το σύστημα πραγματοποιεί την διαχείριση του χρόνου και τον χειρισμό ελαττωματικών συμβάντων εκ μέρους του συντάκτη του UDM. Επίσης γίνεται αναφορά και στους τρόπους με τους οποίους το σύστημα ‘εκμεταλλεύεται’ τις διάφορες ευκαιρίες για βελτιστοποίηση ενώ εκτελεί τον κώδικα του χρήστη.

Research on Data Mining Models for the Internet of Things

Στην παρούσα εργασία προτείνονται 4 μοντέλα εξόρυξης δεδομένων για το Internet of Things (IoT): το μοντέλο εξόρυξης πολλαπλών επιπέδων (multi-layer model), το κατανεμημένο μοντέλο, το μοντέλο βασισμένο σε Grid και ένα μοντέλο από την οπτική της ενσωμάτωσης πολλαπλών τεχνολογιών (multi-technology integration).

Το μοντέλο πολλαπλών επιπέδων χωρίζεται σε 4 επίπεδα: Ένα επίπεδο συλλογής δεδομένων το οποίο είναι υπεύθυνο για τον τρόπο συλλογής των δεδομένων και παρέχει λύσεις σε προβλήματα όπως φιλτράρισμα των δεδομένων, ανοχή σε λάθη, αποδοτικότητα ενέργειας κλπ. Ένα επίπεδο διαχείρισης δεδομένων για την εφαρμογή μιας κεντρικής ή κατανεμημένης βάσης δεδομένων για την αποθήκευση και συμπίεση των δεδομένων. Ένα επίπεδο επεξεργασίας συμβάντων για την αποδοτική διαχείριση και φιλτράρισμα των συμβάντων, όπου με τον όρο συμβάν (event) εννοείται ένας μηχανισμός υψηλότερου επιπέδου για την επεξεργασία των δεδομένων του IoT που επιτρέπει την συγκέντρωση, οργάνωση και ανάλυση αυτών των δεδομένων. Ένα επίπεδο υπηρεσιών εξόρυξης δεδομένων το οποίο βασίζεται στα επίπεδα επεξεργασίας συμβάντων και διαχείρισης δεδομένων και εκτελεί τις διάφορες εργασίες εξόρυξης όπως κατηγοριοποίηση, ομαδοποίηση, πρόβλεψη, ανίχνευση ακραίων τιμών, κλπ.

Το κατανεμημένο μοντέλο αποτελεί λύση στο γεγονός πως τα δεδομένα του IoT είναι μεγάλου όγκου και συνήθως αποθηκεύονται σε διαφορετικούς ιστοτόπους (sites) ενώ παράλληλα απαιτούν επεξεργασία σε πραγματικό χρόνο, με αποτέλεσμα κεντρικοποιημένες (centralized) αρχιτεκτονικές να μην μπορούν να παρέχουν ασφάλεια, ιδιωτικότητα και ανοχή σε σφάλματα. Στις περισσότερες περιπτώσεις οι κεντρικοί κόμβοι δεν χρειάζονται όλα τα δεδομένα παρά μόνο λίγες παραμέτρους. Στο κατανεμημένο μοντέλο λοιπόν υπάρχει ένας κόμβος ελέγχου ο οποίος επιλέγει τον αλγόριθμο εξόρυξης και τα δεδομένα στα οποία θα εφαρμοστεί και στη συνέχεια προσπελάζει τους υπό-κόμβους που περιέχουν τα δεδομένα. Αυτοί οι υπό-κόμβοι λαμβάνουν δεδομένα, τα οποία προ-επεξεργάζονται, συμπιέζουν και αποθηκεύουν τοπικά, από διάφορες έξυπνες συσκευές. Οι υπό-κόμβοι ανταλλάσσουν δεδομένα αντικειμένων, δεδομένα επεξεργασίας και γνώση υπό τον έλεγχο ενός συνεργατικού μηχανισμού διαχείρισης που βασίζεται σε πολλούς παράγοντες.

Το Grid computing είναι μια νέα υπολογιστική υποδομή η οποία είναι ικανή να υλοποιεί ετερογενή, μεγάλης κλίμακας και υψηλής απόδοσης εφαρμογές. Η βασική ιδέα εδώ είναι η θεώρηση των διάφορων έξυπνων αντικειμένων του IoT σαν ένα είδος πόρων για υπολογισμούς στο Grid και στη συνέχεια η χρήση των υπηρεσιών εξόρυξης δεδομένων του Grid για την υλοποίηση των λειτουργιών εξόρυξης δεδομένων για το IoT.

Το μοντέλο από την οπτική της ενσωμάτωσης πολλαπλών τεχνολογιών περιγράφει το αντίστοιχο πλαίσιο για το μελλοντικό Διαδίκτυο. Σε αυτό το μοντέλο τα δεδομένα προέρχονται σαν αποτέλεσμα της ιδιότητας context-awareness, από έξυπνα αντικείμενα ή το περιβάλλον. Υιοθετούνται διευθύνσεις IPV6 128-bit και παρέχεται μια ποικιλία διαφορετικών τρόπων για την πρόσβαση στο μελλοντικό διαδίκτυο. Το έμπιστο επίπεδο ελέγχου είναι σε θέση να διασφαλίσει την αξιοπιστία και τον πλήρη έλεγχο στην μετάδοση δεδομένων. Πάνω σε αυτή τη βάση εκτελούνται οι διάφοροι αλγόριθμοι εξόρυξης δεδομένων.

Επιπλέον οι συγγραφείς σχολιάζουν κάποια σημαντικά θέματα στην εξόρυξη δεδομένων του IoT, όπως η συλλογή δεδομένων και οι ειδικές ανάγκες των έξυπνων αντικειμένων (π.χ. κλιμακωσιμότητα και ανοχή σε λάθη) που πρέπει να ληφθούν υπόψιν. Ακόμη όσον αφορά την διαχείριση των δεδομένων IoT που πολλές φορές είναι ανακριβή, σχετίζονται με το χρόνο και τη τοποθεσία και έχουν υπονοούμενη σημασία ανοιχτά παραμένουν ζητήματα όπως η αναγνώριση και διευθυνσιοδότηση των αντικείμενων, η συμπίεση, η αφαιρετικοποίηση, η δεικτοδότηση και η κλιμακωσιμότητα των δεδομένων, η διαλειτουργικότητα και σημασιολογική κατανόηση, η συγκέντρωση δεδομένων σε επίπεδο συμβάντων και θέματα σχετικά με την ιδιωτικότητα και την ασφάλεια. Τέλος θέματα περαιτέρω ανάλυσης αποτελούν η ανίχνευση και το φιλτράρισμα των συμβάντων, η κατάλληλη επιλογή μεταξύ κεντρικοποιημένων και κατανεμημένων βάσεων δεδομένων, η έρευνα σε αλγορίθμους εξόρυξης ειδικά για το IoT και η σύνδεση αυτών των αλγορίθμων με τις τεχνολογίες νέας γενιάς του Διαδικτύου.