ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΟΥΣΙΑΣΗ / ΕΞΕΤΑΣΗ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ξανθάκης Γεώργιος
Μεταπτυχιακός Φοιτητής

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Επόπτης Μεταπτυχιακής Εργασίας: Καθηγητής, Α. Μπίλας

Παρασκευή, 2 Ιουλίου 2021, ώρα 10:00 π.μ.

Join Zoom Meeting

https://zoom.us/j/91906158786

" Ισορροπώντας το κόστος της ανάκτησης χώρου και του αυξημένου I/O σε συστήματα κλειδιού-τιμής βασισμένα στο LSM δέντρο με την υβριδική τοποθέτηση ζευγαριών κλειδιού-τιμής "

Περίληψη

Η τεχνική της διαχώρισης κλειδιού-τιμής εισάγει τυχαιότητα στα μοτίβα πρόσβασης ώστε να μειώσει το επιπλέον I/O στα συστήματα κλειδιού-τιμής που είναι βασισμένα στο LSM δέντρο και στοχεύουν τις γρήγορες συσκευές αποθήκευσης (NVMe). Η τεχνική της διαχώρισης κλειδιού-τιμής αποθηκεύει τα ζευγάρια σε ένα log και αποθηκεύει κάποια μεταπληροφορία για την ανάκτηση των δεδομένων στο ευρετήριο του συστήματος. Παρ' όλα αυτά η τεχνική της διαχώρισης κλειδιού-τιμής έχει ένα σημαντικό μειονέκτημα που τη κάνει λιγότερο ελκυστική. Για τις λειτουργίες διαγραφής και ενημέρωσης που είναι σημαντικές για τα workloads που εμφανίζονται στη βιομηχανία προκαλούν συχνά την ακριβή λειτουργία ανάκτησης χώρου (Garbage Collection) στο log

του συστήματος και καθιστά τα συστήματα που χρησιμοποιούν αυτή τη τεχνική να μην είναι πρακτικά.

Σε αυτή την εργασία, σχεδιάζουμε και υλοποιούμε το Parallax, το οποίο προτείνει μια υβριδική τοποθέτηση των ζευγαριών κλειδιού-τιμής που μειώνει το κόστος της ανάκτησης χώρου σημαντικά και μεγιστοποιεί το κέρδος όταν αποθηκεύουμε τις τιμές σε ένα log. Πρώτα μοντελοποιούμε τα κέρδη της τεχνικής διαχώρισης κλειδιού-τιμής για διάφορα μεγέθη. Χρησιμοποιούμε το μοντέλο για να κατηγοριοποιήσουμε τα ζευγάρια κλειδιού-τιμής σε τρεις κατηγορίες στη μικρή, τη μεσαία, και τη μεγάλη.

Στη συνέχεια, το Parallax χρησιμοποιεί διαφορετικές προσεγγίσεις για τη κάθε κατηγορία: Τοποθετεί τα μεγάλα ζευγάρια πάντα σε ένα log και τα μικρά ζευγάρια πάντα στο ευρετήριο. Για την μεσαία κατηγορία χρησιμοποιεί μια μεικτή στρατηγική που συνδυάζει τα κέρδη του log για όλα εκτός από τα τελευταία επίπεδα (συνήθως το τελευταίο ή προτελευταίο) στην LSM δομή, όπου πραγματοποιεί μια πλήρη σάρωση στο log και τοποθετεί τις τιμές μέσα στο ευρετήριο. Μετά τη σάρωση ανακτά το χώρο του μεσαίου log χωρίς να χρειάζεται η λειτουργία της ανάκτησης χώρου (Garbage Collection).

Στην πειραματική μας ανάλυση συγκρίνουμε το Parallax με τη RocksDB που τοποθετεί όλες τις τιμές μέσα στο ευρετήριο της και τη BlobDB που τοποθετεί όλες τις τιμές σε ένα log. Καταλήγουμε ότι το Parallax αυξάνει την απόδοση από 12.4 μέχρι 17.83x, μειώνει το επιπλέον I/O από 26 μέχρι 27.1x και αυξάνει την αποτελεσματικότητα του επεξεργαστή από 18.7 μέχρι 28x αντίστοιχα για όλα τα workloads του YCSB εκτός από αυτά που βασίζονται σε scans.

University of Crete

Computer Science Department

M.Sc. Thesis presentation / examination

Xanthakis Georgios

Master's Thesis Supervisor: Professor, A. Bilas

Friday, 2 July 2021, 10:00 a.m.

Join Zoom Meeting

https://zoom.us/j/91906158786

“Balancing Garbage Collection vs I/O Amplification using hybrid Key-Value Placement in LSM-based Key-Value Stores”

Abstract

Key-value (KV) separation is a technique that introduces randomness in the I/O access patterns to reduce I/O amplification in LSM-based key-value stores for fast storage devices (NVMe). KV separation has a significant drawback that makes it less attractive: Delete and especially update operations that are important in modern workloads result in frequent and expensive garbage collection (GC) in the value log.

In this thesis, we design and implement Parallax, which proposes hybrid KV placement that reduces GC overhead significantly and maximizes the benefits of using a log. We first model the benefits of KV separation for different KV pair sizes. We use this model to classify KV pairs in three categories small, medium, and large. Then, Parallax uses different approaches for each KV category: It always places large values in a log and small values in place. For medium values it uses a mixed strategy that combines the benefits of using a log and eliminates GC overhead as follows: It places medium values in a log for all but the last few (typically one or two) levels in the LSM structure, where it performs a full compaction, merges values in place, and reclaims log space without the need for GC.

We evaluate Parallax against RocksDB that places all values in place and BlobDB that always performs KV separation. We find that Parallax increases throughput by up to 12.4x and 17.83x, decreases I/O amplification by up to 27.1x and 26x, and increases CPU efficiency by up to 18.7x and 28x respectively, for all but scan-based YCSB workloads.