REPLICATION AND CACHING

Νικόλαος Καλλιμάνης
Κώστας Λίλλης
Σπυρίδων Δημήτριος Αγάθος

1. Γενικά

Τα συστήματα ομοτίμων (peer to peer) έχουν χρησιμοποιηθεί ευρέως για το διαμοιρασμό αρχείων μεταξύ χρηστών. Τα εν' λόγω συστήματα είναι δυναμικά, στα οποία επιτρέπεται νέοι κόμβοι να συμμετέχουν, άλλοι να αποχωρούν, αλλά και μερικοί κόμβοι να αποτυγχάνουν. Η αποτυχία ενός κόμβου σε ένα σύστημα ομοτίμων σηματοδοτεί τη μη διαθεσιμότητα των δεδομένων που είναι αποθηκευμένα σε αυτόν. Έτσι γεννάται η ανάγκη ύπαρξης πολλαπλών και διασκορπισμένων αντιγράφων των δεδομένων στο σύστημα (replication), ώστε αυτά να είναι διαθέσιμα ακόμα και αν αρκετοί κόμβοι του συστήματος έχουν αποτύχει. Επίσης αξίζει να ειπωθεί ότι η τεχνική του replication χρησιμοποιείται σε ορισμένες περιπτώσεις για την αποσυμφόρηση κόμβων, οι οποίοι κατέχουν δημοφιλή αρχεία και έτσι διεκπεραιώνουν μεγάλο αριθμό αιτημάτων.

Ένα άλλο σημαντικό ζήτημα στη σχεδίαση συστημάτων ομοτίμων είναι η ταχεία διεκπεραίωση αιτημάτων (queries) για δεδομένα. Ακόμη ένα άλλο βασικό ζήτημα είναι η αποφυγή της υπερφόρτωσης του συστήματος από αιτήματα που αφορούν δημοφιλή δεδομένα. Τα δύο προαναφερθέντα ζητήματα αντιμετωπίζονται μέσω της τεχνικής αποθήκευσης των δημοφιλών προορισμών σε τοπική λανθάνουσα μνήμη (caching).

Στην παρούσα εργασία αναφέρονται και προτείνονται διάφορες τεχνικές για replication και caching, καθώς και η εφαρμογή τους στα δομημένα συστήματα CAN, CHORD, BATON και SKIP NET.

2. Replication

Παρακάτω θα αναφερθούν διάφορες τεχνικές ώστε να επιτυγχάνουμε replication στα συστήματα CAN, CHORD, BATON και SKIP NET. Σε όλες τις τεχνικές δημιουργούμε r αντίγραφα στα δεδομένα σε r διαφορετικούς κόμβους, αυξάνοντας τη διαθεσιμότητα τους. Έτσι αν η πιθανότητα να αποτύχει ένας κόμβος είναι p με $0 < p < 1$, τότε η πιθανότητα να αποτύχουν και οι r κόμβοι είναι p^r. Όσο αυξάνουμε το r τόσο αυξάνεται η διαθεσιμότητα της πληροφορίας, αλλά παράλληλα αυξάνεται η πληροφορία που πρέπει να αποθηκεύεται σε κάθε κόμβο, ενώ το σύστημα επιβαρύνεται διατηρώντας τη συνέπεια των αντιγράφων.

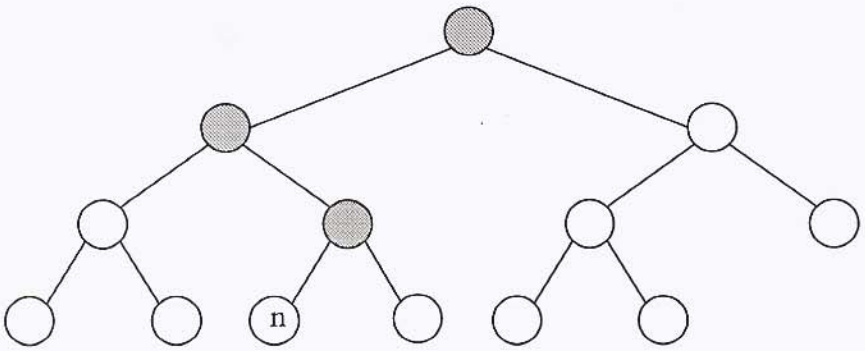
Μία δημοφιλής τεχνική για replication σε δομημένα συστήματα είναι η δημιουργία αντιγράφων σε επιλεγμένους κόμβους (συνήθως σε γειτονικούς κόμβους) στο σύστημα. Η τεχνική αυτή εφαρμόζεται στο CHORD και στο CAN, ενώ όπως θα ειπωθεί μπορεί να εφαρμοστεί στο BATON και στο SKIP NET.

Στο CHORD ένας κόμβος δύναται να χρησιμοποιεί τη λίστα, στην οποία είναι αποθηκευμένοι οι r επόμενοι successors, ώστε να αποθηκεύει r αντίγραφα των δεδομένων του σε αυτούς. Με αυτή την τεχνική τα δεδομένα ενός κόμβου που αποτυγχάνει παραμένουν διαθέσιμα στο σύστημα. Για να μην είναι διαθέσιμα τα δεδομένα ενός κόμβου πρέπει και οι r successors να αποτύχουν. Επίσης είναι δυνατό να χρησιμοποιήσουμε την παραπάνω τεχνική ώστε να αποφευχθεί η υπερφόρτωση κόμβων, οι οποίοι περιέχουν εξαιρετικά δημοφι-

λή δεδομένα. Έτσι ένας υπερφορτωμένος κόμβος που δέχεται ένα αίτημα, επιλέγει από τη λίστα με τους r successors τυχαία έναν κόμβο και του αναθέτει την εξυπηρέτηση του αιτήματος.

Στο CAN είναι δυνατό r κόμβοι να μοιράζονται την ίδια περιοχή συντεταγμένων (overloading coordinate zones), δηλαδή κάθε κόμβος έχει ένα αντίγραφο των δεδομένων της περιοχής. Η πιθανότητα να αποτύχουν όλοι οι κόμβοι που διαμοιράζονται μία περιοχή είναι μικρή, αυξάνοντας τη διαθεσιμότητα των δεδομένων. Όταν για κάθε περιοχή είναι υπεύθυνος ένας κόμβος είναι δυνατό να αντιγράφει τα δεδομένα του στους γειτονές του. Στην περίπτωση που ένας κόμβος αποτύχει, κάποιος γειτονικός κόμβος θα αναλάβει την περιοχή του, οπότε και τα δεδομένα αυτού θα παραμένουν διαθέσιμα. Επίσης στην περίπτωση υπερφόρτωσης ένας κόμβος μπορεί να αντιγράφει τα δεδομένα του στους γειτονικούς του κόμβους, κατανέμοντας το φορτίο εξυπηρέτησης σε αυτούς.

Στην περίπτωση του BATON η εφαρμογή της τεχνικής replication έχει νόημα μόνο για τη διαθεσιμότητα των δεδομένων αφού το BATON κάνει ικανοποιητική αυτόματη κατανομή φορτίου (load balance). Μία λύση για replication στο BATON είναι η αντιγραφή των δεδομένων ενός κόμβου στους r αμέσως προηγούμενους πρόγονους του, αφού κάποιος από αυτούς θα είναι υπεύθυνος για τη διαχείριση της αποτυχίας του απογόνου. Έτσι ένας εσωτερικός κόμβος θα κρατάει το πολύ 2^{r+1}-2 αντίγραφα από απογόνους[†]. Στο παρακάτω παράδειγμα φαίνεται αναλυτικά η προαναφερθείσα τεχνική, με κόκκινο χρώμα σημειώνονται οι κόμβοι οι οποίοι θα κρατούν αντίγραφα των δεδομένων του κόμβου n. Στο εν' λόγω παράδειγμα το r=3.



Σχήμα 1: Παράδειγμα replication στο BATON.

Στο SKIP NET κάθε κόμβος διατηρεί μία λίστα με δείκτες σε r γειτονικούς κόμβους στο root δακτύλιο. Έτσι είναι δυνατό να αποθηκεύσουμε r αντίγραφα της πληροφορίας ενός κόμβου στους r γείτονες. Έτσι όταν ένας κόμβος αποτύχει, τα δεδομένα του θα υπάρχουν στους r γείτονές του. Επίσης είναι δυνατό να χρησιμοποιήσουμε την παραπάνω τεχνική ώστε να αποφευχθεί η υπερφόρτωση κόμβων, οι οποίοι περιέχουν εξαιρετικά δημοφιλή δεδομένα. Έτσι ένας υπερφορτωμένος κόμβος που δέχεται ένα αίτημα, επιλέγει από τη λίστα με τους r γείτονες τυχαία έναν κόμβο και του αναθέτει την εξυπηρέτηση του αιτήματος.

[†] Επειδή το πλήθος των αντιγράφων που κρατάει ένας κόμβος είναι της τάξης 2^r, το r πρέπει να είναι μικρός αριθμός.

Μία άλλη τεχνική για replication σε δομημένα συστήματα είναι η δημιουργία πολλαπλών πραγματικοτήτων στο σύστημα (realities). Η τεχνική αυτή εφαρμόζεται στο CAN, ενώ όπως θα ειπωθεί μπορεί να εφαρμοστεί και στο CHORD.

Στο CAN μπορούμε να υποστηρίξουμε πολλούς ανεξάρτητους χώρους συντεταγμένων. Κάθε κόμβος συμμετέχει σε όλους τους χώρους συντεταγμένων, αλλά σε κάθε χώρο συντεταγμένων αναλαμβάνει μία διαφορετική περιοχή (zone). Άρα για ένα σύστημα CAN με r πραγματικότητες κάθε κόμβος αναλαμβάνει r περιοχές. Τα δεδομένα κάθε περιοχής αντιγράφονται σε κάθε πραγματικότητα αυξάνοντας έτσι τη διαθεσιμότητα τους.

Στο CHORD είναι δυνατό να έχουμε r ισομεγέθεις κύκλους[‡], τους οποίους καταχρηστικά τους ονομάζουμε πραγματικότητες. Κάθε κόμβος καταλαμβάνει διαφορετική θέση σε καθένα από τους r κύκλους. Για την αντιστοίχηση των κόμβων σε μία θέση σε έναν από τους r κύκλους χρησιμοποιούνται r διαφορετικές συναρτήσεις κατακερματισμού. Εν' γένει κάθε κλειδί έχει ως successor διαφορετικό κόμβο σε κάθε κύκλο (reality). Αυτό προφανώς αυξάνει τη διαθεσιμότητα, αφού αν ο υπεύθυνος για ένα κλειδί σε ένα κύκλο έχει αποτύχει, τότε το κλειδί θα είναι διαθέσιμο από έναν άλλο κόμβο σε ένα άλλο κύκλο.

Μία άλλη τεχνική για replication σε δομημένα συστήματα είναι η χρησιμοποίηση πολλών συναρτήσεων κατακερματισμού. Η τεχνική αυτή εφαρμόζεται στο CAN, ενώ όπως θα ειπωθεί μπορεί να εφαρμοστεί και στο CHORD.

Στο CAN μπορούμε να χρησιμοποιήσουμε k διαφορετικές συναρτήσεις κατακερματισμού για την αντιστοίχηση ενός κλειδιού σε k διαφορετικά σημεία του χώρου συντεταγμένων. Τα δεδομένα που αντιστοιχούν σε ένα κλειδί αντιγράφονται στις k διαφορετικές περιοχές, όπου αντιστοιχούν τα σημεία.

Στο CHORD μπορεί να χρησιμοποιηθεί η προαναφερθείσα τεχνική με παρόμοιο τρόπο. Έτσι ένα κλειδί αντιστοιχίζεται σε k θέσεις του κύκλου χρησιμοποιώντας k συναρτήσεις κατακερματισμού. Ομοίως τα δεδομένα του αντιγράφονται στις k θέσεις του κύκλου.

Στο SKIP NET μπορεί να χρησιμοποιηθεί ο μηχανισμός CLB (που χρησιμοποιείται για εξισορρόπηση φορτίου) ώστε να επιτύχουμε replication. Ειδικότερα ο CLB μηχανισμός χρησιμοποιεί μία συνάρτηση κατακερματισμού για την αντιστοίχηση ενός αρχείου σε έναν κόμβο ενός συσχετισμένου συνόλου κόμβων (domain). Έτσι δύναται να χρησιμοποιηθούν r διαφορετικές συναρτήσεις κατακερματισμού, ώστε να έχουμε r αντίγραφα της πληροφορίας σε ένα domain.

Παρακάτω θα περιγραφεί μία τεχνική, η οποία μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σύστημα ομοτίμων, ανεξαρτήτως σχεδίασης. Ουσιαστικά με αυτή την τεχνική μπορούμε να θεωρήσουμε ένα σύστημα ομοτίμων ως ένα «μαύρο κουτί». Έστω ότι θέλουμε για κάθε κλειδί να δημιουργήσουμε r αντίγραφα της πληροφορίας του. Επιλέγουμε ένα σύνολο από r συναρτήσεις χαρτογράφησης, οι οποίες δοθέντος ενός κλειδιού key μας δίνουν ως έξοδο το σύνολο $K=\{key_1, key_2, \ldots, key_r\}$, το οποίο περιέχει r κλειδιά διαφορετικά μεταξύ τους. Αν θέλουμε να κάνουμε εισαγωγή του key κάνουμε εισαγωγή στο σύστημα όλων των κλειδιών του K. Αν επιθυμούμε να κάνουμε αναζήτηση του key στο σύστημα επιλέγου-

‡ Κάθε κύκλος περιέχει τον ίδιο αριθμό θέσεων.

με τυχαία ένα στοιχείο του K και πραγματοποιούμε το αίτημα. Αν θέλουμε να κάνουμε διαγραφή του key διαγράφουμε από το σύστημα κάθε στοιχείο του K.

3. Caching

Αυτή η τεχνική είναι χρήσιμη για την ταχεία διεκπεραίωση αιτημάτων προς το σύστημα, καθώς και για αποφυγή της υπερφόρτωσης του συστήματος από αιτήματα που αφορούν δημοφιλή δεδομένα. Σε όλα τα συστήματα ομοτίμων ένας κόμβος μπορεί να διατηρήσει μια λανθάνουσα μνήμη, στην οποία αποθηκεύονται προσωρινά οι απαντήσεις των πιο πρόσφατων αιτημάτων που επιτέλεσε. Έτσι ένας κόμβος πριν προωθήσει ένα αίτημα, το οποίο μπορεί να είναι είτε τοπικό αίτημα είτε να είναι ένα αίτημα που να προέρχεται από έναν άλλο κόμβο, ελέγχει αν η απάντηση του αιτήματος υπάρχει στην τοπική, λανθάνουσα μνήμη του και αν τη βρει, αποδίδει ο ίδιος την απάντηση χωρίς περαιτέρω καθυστέρηση. Αξίζει να σημειωθεί ότι στη λανθάνουσα μνήμη ένας κόμβος μπορεί να αποθηκεύει τα δημοφιλέστερα αιτήματα και όχι τα πιο πρόσφατα. Γενικά είναι δυνατό να χρησιμοποιηθούν πολλοί ευρετικοί αλγόριθμοι, για την πλήρωση της λανθάνουσας μνήμης με απαντήσεις αιτημάτων.