ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός

Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

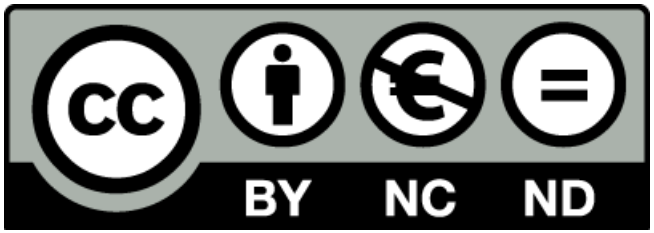
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Κατανεμημένα Συστήματα με Java», Ι. Κάβουρας, Ι. Μήλης, Γ. Ξυλωμένος, Α. Ρουκουνάκη, 3η έκδοση, 2011, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Εισαγωγή στην έννοια του πίνακα κατακερματισμού και την κατανεμημένη μορφής του.
- Εξοικείωση με τη σχεδίαση και την υλοποίηση του συστήματος Chord και του συστήματος Pastry.
- Κατανόηση της λειτουργίας του συστήματος πολυεκπομπής Scribe.

Περιεχόμενα ενότητας (1 από 2)

- Κατανεμημένος κατακερματισμός
- Το σύστημα Chord
  - Αναζήτηση στο Chord
  - Προσχώρηση στο Chord
  - Αποχώρηση από το Chord

Περιεχόμενα ενότητας (2 από 2)

- Το σύστημα Pastry
  - Δρομολόγηση στο Pastry
  - Προσχώρηση/αποχώρηση στο Pastry
  - Επίδοση του Pastry
- Το σύστημα Scribe

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κατανεμημένος κατακερματισμός

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Κατανεμημένη αναζήτηση (1 από 3)

- Πρόβλημα εντοπισμού πόρων
  - Δίνεται το όνομα ενός πόρου
  - Ζητείται η διεργασία που έχει πρόσβαση σε αυτόν
  - Εμφανίζεται σε πολλά κατανεμημένα συστήματα
- Κατανεμημένα συστήματα ονομασίας
  - Το DNS (και το X.500) παρέχει ιεραρχικά ονόματα
  - Οι εξυπηρετητές οργανώνονται και αυτοί ιεραρχικά
  - Καλή λύση για το μοντέλο πελάτη-εξυπηρετητή
  - Ακατάλληλη για το μοντέλο ομοτίμων

Κατανεμημένη αναζήτηση (2 από 3)

- Δίκτυα καταμερισμού αρχείων
  - Κάθε κόμβος διαθέτει αρχεία προς ανάκτηση
    - Τα αρχεία συνεισφέρονται από τις διεργασίες
  - Το σύστημα επιτρέπει τον εντοπισμό των αρχείων
- Κατανεμημένα συστήματα αποθήκευσης
  - Κάθε κόμβος λειτουργεί ως εξυπηρετητής αρχείων
    - Τα αρχεία συνεισφέρονται από τους πελάτες
  - Το σύστημα επιτρέπει τον εντοπισμό των αρχείων
    - Κάθε αρχείο αποθηκεύεται σε πολλούς εξυπηρετητές

Κατανεμημένη αναζήτηση (3 από 3)

- Γενικό πρόβλημα στα συστήματα ομοτίμων
  - Έστω ένα κλειδί k
    - Παράδειγμα 1: το όνομα ενός αρχείου
    - Παράδειγμα 2: όνομα και αριθμός ομάδας αρχείου
  - Ποια διεργασία διαθέτει τα στοιχεία για το k;
    - Παράδειγμα 1: ποιες διεργασίες γνωρίζουν που βρίσκεται;
    - Παράδειγμα 2: ποιες διεργασίες αποθηκεύουν το αρχείο;
  - Πώς μπορούμε να έχουμε αποδοτική αναζήτηση;

Κατακερματισμός (1 από 6)

- Ευρετήρια (πίνακες) κατακερματισμού
  - Το κλειδί k περνά από συνάρτηση h
  - Το πεδίο τιμών είναι οι ακέραιοι από 0 έως m-1
  - Τα στοιχεία του k αποθηκεύονται στην h(k)
    - Ο πίνακας κατακερματισμού έχει μέγεθος m θέσεις
  - Στόχος: ομοιόμορφη κατανομή κλειδιών
    - Κάθε θέση ιδανικά αντιστοιχεί σε ένα μόνο κλειδί

Κατακερματισμός (2 από 6)

- Κατανεμημένος πίνακας κατακερματισμού (DHT)
  - Κάθε διεργασία περιέχει ένα μέρος του πίνακα
  - Κατακερματίζουμε και τις διευθύνσεις των διεργασιών
  - Σε κάθε διεργασία αναθέτουμε τα «κοντινά» της κλειδιά
  - Ψάχνουμε τη διεργασία που βρίσκεται «κοντά» στο κλειδί
- Οι δύο έννοιες του «κοντά»
  - Τοπολογική: αναφέρεται στο υποκείμενο δίκτυο
  - Αριθμητική: αναφέρεται στο υπερκείμενο δίκτυο
  - Γενικά τοπολογική και αριθμητική απόσταση διαφέρουν

Κατακερματισμός (3 από 6)

- Παράδειγμα: οργάνωση διεργασιών σε δακτύλιο
  - Κατακερματίζουμε τις διευθύνσεις των διεργασιών
    - Ο δακτύλιος βασίζεται στις κατακερματισμένες διευθύνσεις
  - Διεργασίες κοντά στο δίκτυο είναι τοπολογικά κοντά
    - Οι διευθύνσεις τους είναι παρόμοιες
  - Στο υπερκείμενο δίκτυο δεν είναι αριθμητικά κοντά
    - Οι διευθύνσεις κατακερματίζονται σε διαφορετικές τιμές
  - Διεργασίες κοντά στο δακτύλιο είναι μακρινές

Κατακερματισμός (4 από 6)

- Στόχοι κατανεμημένου πίνακα κατακερματισμού
  - Έστω N κόμβοι και K κλειδιά
  - Σε κάθε κόμβο αντιστοιχεί 1/N του πεδίου τιμών
  - Σε κάθε κόμβο αποθηκεύονται K/N κλειδιά
    - Ισοκατανομή φόρτου στο σύστημα
  - Αποδοτική υποστήριξη αναζήτησης κλειδιών
    - Γρήγορος εντοπισμός του κατάλληλου κόμβου
    - Δρομολόγηση (της αναζήτησης) με βάση το κλειδί
    - Στο Internet έχουμε δρομολόγηση με βάση τη διεύθυνση

Κατακερματισμός (5 από 6)

- Στόχοι κατανεμημένου πίνακα κατακερματισμού
  - Σε κάθε αλλαγή μετακινούνται K/N κλειδιά
    - Προσχώρηση στο ή αποχώρηση από το σύστημα
    - Περίπου ο φόρτος που αντιστοιχεί σε έναν κόμβο
- Χρήση κρυπτογραφικών συναρτήσεων
  - Είσοδος: συμβολοσειρές αυθαίρετου μήκους
    - Διευθύνσεις IP, ονόματα DNS, ονόματα αρχείων
  - Έξοδος: φυσικοί αριθμοί
    - MD-5: έξοδος 128 bit
    - SHA-1: έξοδος 160 bit

Κατακερματισμός (6 από 6)

- Χρήση κρυπτογραφικών συναρτήσεων
  - Τα κλειδιά/αναγνωριστικά είναι επίπεδα
  - Μικρές διαφορές στην είσοδο
    - Πολύ διαφορετικά κλειδιά/αναγνωριστικά
    - Απομακρύνονται στην έξοδο
  - Ομοιόμορφη κατανομή στο πεδίο τιμών
    - Ανεξάρτητα από κατανομή στο πεδίο ορισμού
  - Ακατάλληλες για προσεγγιστικά ερωτήματα
    - Παρόμοιες είσοδοι οδηγούν σε διαφορετικές εξόδους

Το σύστημα Chord

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Βασικά στοιχεία Chord (1 από 5)

- Κόμβοι / κλειδιά οργανώνονται σε δακτύλιο
  - Κόμβοι: κατακερματισμός διευθύνσεων
  - Κλειδιά: κατακερματισμός ονομάτων
- Συνήθως χρήση της SHA-1
  - Είσοδος: συμβολοσειρά οποιουδήποτε μήκους
  - Έξοδος: αριθμός 160 bit
  - Δακτύλιος με 2^{160} αναγνωριστικά
    - Και με 2^{160} κλειδιά

Βασικά στοιχεία Chord (2 από 5)

- Ο κόμβος αποτελείται από δύο μέρη
  - Βιβλιοθήκη του Chord
    - Ανεξάρτητη από την εφαρμογή
    - Παρέχει πολύ απλή διεπαφή
  - Κώδικας εφαρμογής
    - Αξιοποιεί τη βιβλιοθήκη του Chord
    - Παρέχει πιο σύνθετη διεπαφή

Βασικά στοιχεία Chord (3 από 5)

- Διεπαφή βιβλιοθήκης του Chord
  - find_successor(k)
    - Εφαρμογή -> βιβλιοθήκη
    - Επιστρέφει πρώτο κόμβο με αναγνωριστικό >=k
    - Αυτός ο κόμβος είναι υπεύθυνος για το κλειδί k
  - redistribute_keys(k)
    - Βιβλιοθήκη -> εφαρμογή
    - Το πρώτο κλειδί του κόμβου είναι αμέσως μετά το k
    - Καλείται όταν αλλάζει η περιοχή ευθύνης του κόμβου
  - Οι εφαρμογές δομούνται με βάση αυτές τις κλήσεις

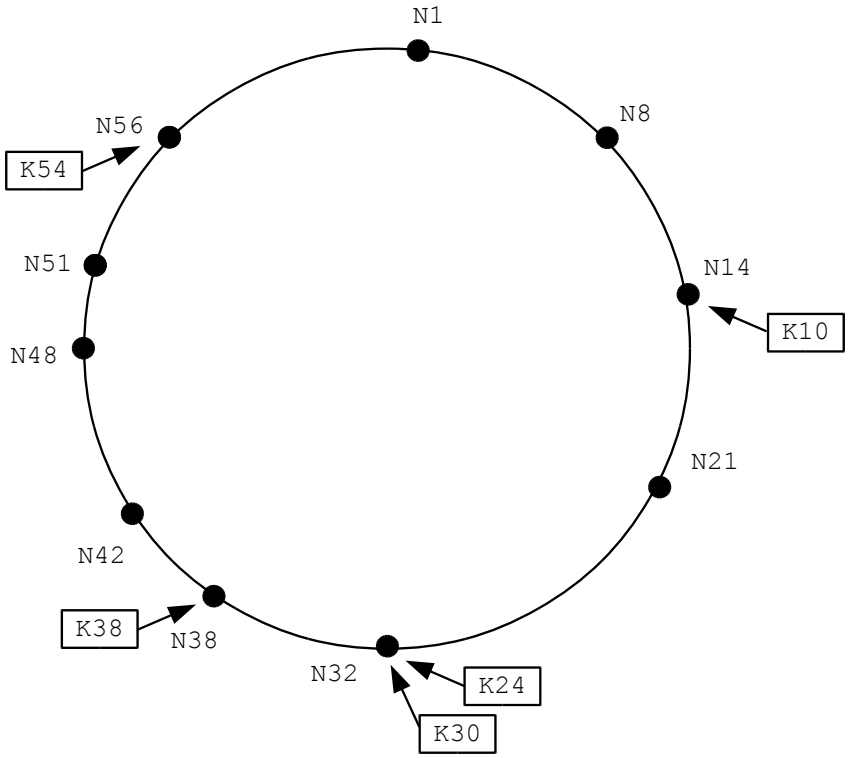
Βασικά στοιχεία Chord (4 από 5)

- Παράδειγμα: δίκτυο καταμερισμού αρχείων
  - Είσοδος κόμβου στο σύστημα
    - Κατακερματισμός των ονομάτων αρχείων σε κλειδιά
    - Εντοπισμός των κόμβων που ευθύνονται για τα κλειδιά
    - Αποθήκευση διευθύνσεων αρχείων στους κόμβους
  - Ανάκτηση αρχείου από το σύστημα
    - Κατακερματισμός ζητούμενου ονόματος αρχείου σε κλειδί
    - Εντοπισμός του κόμβου που ευθύνεται για το κλειδί
    - Ανάκτηση της διεύθυνσης του αρχείου από αυτό τον κόμβο
    - Ανάκτηση του αρχείου από τον κόμβο που το διαθέτει

Βασικά στοιχεία Chord (5 από 5)

- Παράδειγμα: δίκτυο καταμερισμού αρχείων
  - Προσχώρηση ή αποχώρηση κόμβων
    - Ενημέρωση των κόμβων που επηρεάζονται
    - Η εφαρμογή φροντίζει για τη μεταφορά των κλειδιών
- Όρια ευθύνης της βιβλιοθήκης του Chord
  - Η βιβλιοθήκη δεν ασχολείται με διαχείριση κλειδιών
    - Αυτό είναι θέμα της εφαρμογής που την αξιοποιεί
  - Η βιβλιοθήκη ασχολείται με όρια ευθύνης κόμβων
    - Αναδιοργάνωση δακτυλίου σε προσχωρήσεις/αποχωρήσεις

Κόμβοι και κλειδιά (1 από 3)



- successor(k): πρώτος κόμβος >=k
  - Γεωμετρικά, πρώτος κόμβος μετά το k (δεξιόστροφα)
  - Το κλειδί *k* αντιστοιχίζεται στον successor(k)

Κόμβοι και κλειδιά (2 από 3)

- Παράδειγμα: δίκτυο καταμερισμού αρχείων
  - 10 κόμβοι, κύκλος με 64 κλειδιά/αναγνωριστικά
  - Ο 1 θέλει να διαθέσει ένα αρχείο στο σύστημα
  - Το όνομα αρχείου κατακερματίζεται στο 10
  - Ο 1 αποθηκεύει τη διεύθυνσή του στον successor(10)
    - Μπορεί να είναι διεύθυνση IP και θύρα TCP
  - Η αναζήτηση του αρχείου γίνεται μέσω του successor(10)

Κόμβοι και κλειδιά (3 από 3)

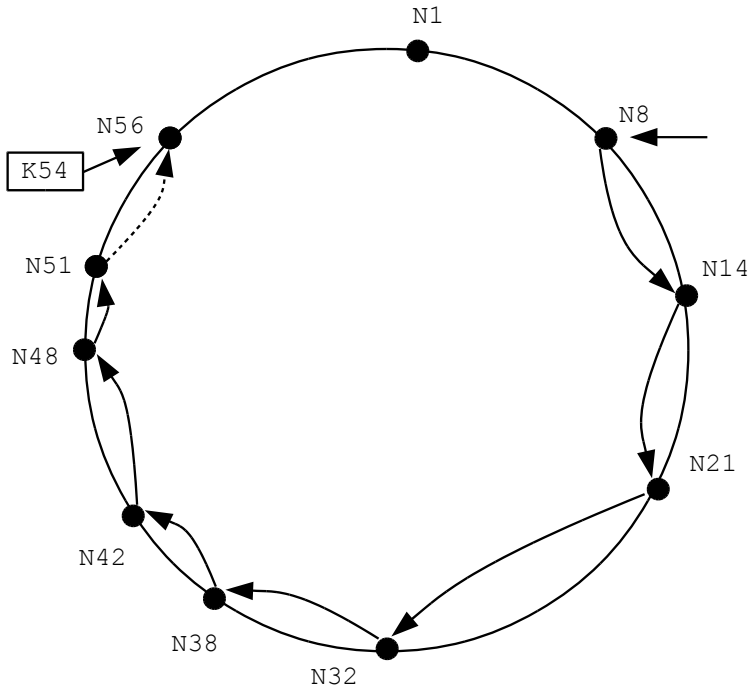
- Ανακατανομές κλειδιών
  - Ο κόμβος αναλαμβάνει περιοχή πριν από αυτόν
  - Προσχώρηση: ο κόμβος «παίρνει» κλειδιά
  - Αποχώρηση: ο κόμβος «στέλνει» κλειδιά
    - Πάντα από/προς τον επόμενο κόμβο
  - Καλείται η redistribute_keys() όταν πρέπει
  - Η μεταφορά των κλειδιών είναι θέμα εφαρμογής
    - Οι ίδιοι οι πόροι δεν μετακινούνται, μόνο τα κλειδιά

Αναζήτηση στο Chord

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Απλή αναζήτηση (1 από 3)



n.find_successor(k)

Αν το k περιέχεται στο διάστημα (n, successor]

Επίστρεψε successor

Διαφορετικά

Επίστρεψε successor.find_successor(k)

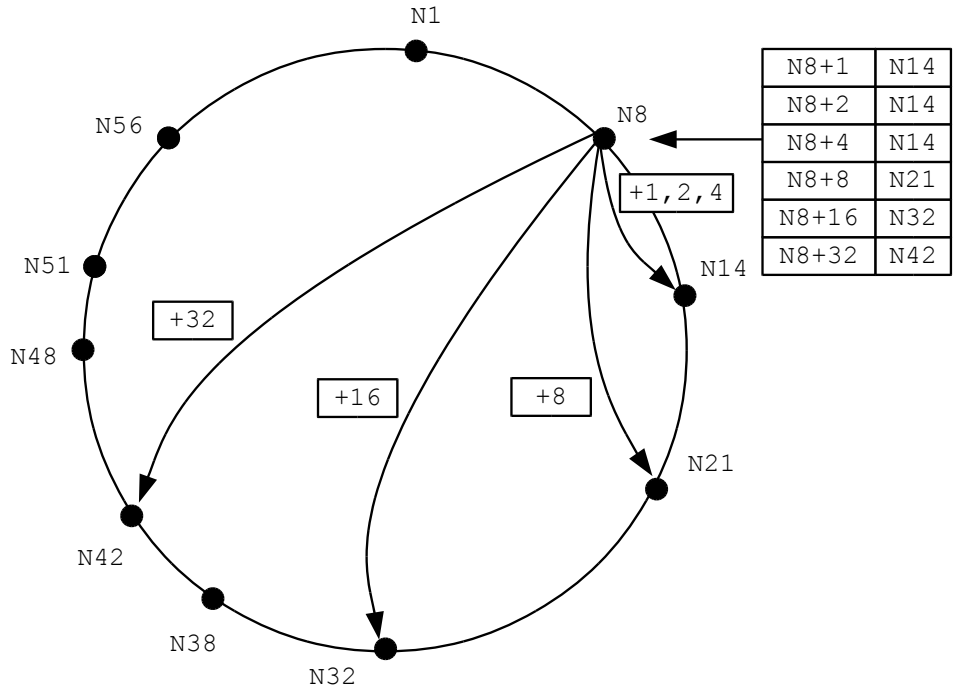
Απλή αναζήτηση (2 από 3)

- Συμβολισμοί ψευδοκώδικα
  - n.func(): κλήση μεθόδου στον κόμβο n
    - Απομακρυσμένες κλήσεις διαδικασιών (RPC)
    - Απομακρυσμένες κλήσεις μεθόδων (RMI)
  - n.x: ανάγνωση της μεταβλητής x
    - Μέσω απομακρυσμένης μεθόδου ανάγνωσης
  - Οι τοπικές κλήσεις παραλείπουν τον κόμβο
  - n: διεύθυνση και αναγνωριστικό του κόμβου

Απλή αναζήτηση (3 από 3)

- Λειτουργία απλής αναζήτησης
  - Ο κόμβος αρκεί να ξέρει τον successor του
  - Κινούμαστε από κόμβο σε κόμβο
  - Σταματάμε όταν το κλειδί ανήκει στον successor
  - Πλήθος μηνυμάτων ανάλογο του N
    - Οι τοπολογικές αποστάσεις μπορεί να είναι μεγάλες
  - Παράδειγμα: αναζήτηση του K54 από τον κόμβο N8
    - Η αναζήτηση διατρέχει το δακτύλιο μέχρι τον κόμβο N56

Σύνθετη αναζήτηση (1 από 7)



- Πίνακας δακτύλων για επιτάχυνση αναζήτησης
  - n.finger[i] = successor($n+2^{i-1}$)
    - Παρατηρούμε ότι n.finger[1] = n.successor
  - Το δάκτυλο i δείχνει στον successor($n+2^{i-1}$)

Σύνθετη αναζήτηση (2 από 7)

- Αξιοποίηση πίνακα δακτύλων
  - Εκμεταλλευόμαστε τις εκθετικές αποστάσεις
  - Σε κάθε βήμα καλύπτουμε τουλάχιστο τη μισή απόσταση

n.find_successor(k)

Αν το k περιέχεται στο διάστημα (n, successor]

Επίστρεψε successor

Διαφορετικά

n’ = closest_preceding_node(k)

Επίστρεψε n’.find_successor(k)

Σύνθετη αναζήτηση (3 από 7)

- Εντοπισμός του βέλτιστου δακτύλου

  n.closest_preceding_node(k)

  Για κάθε i από m μέχρι 1

  Αν το finger[i] περιέχεται στο διάστημα (n, k)

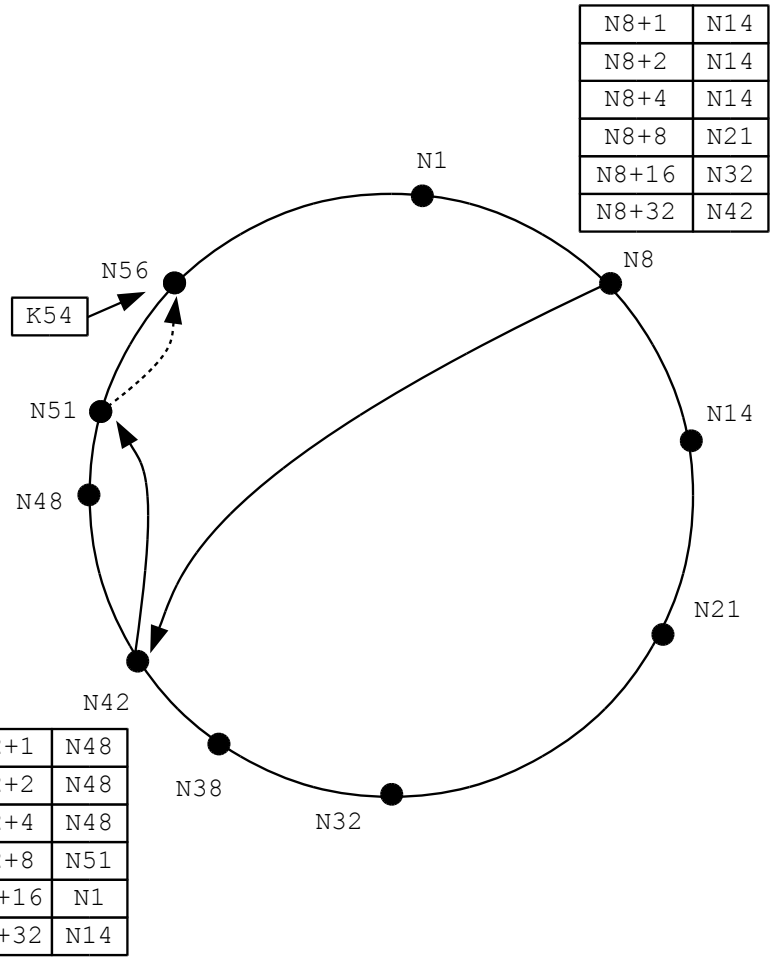
  Επίστρεψε finger[i]

  Επίστρεψε n

Σύνθετη αναζήτηση (4 από 7)

- Εντοπισμός του βέλτιστου δακτύλου
  - Πρώτο δάκτυλο που δείχνει πριν το κλειδί
    - Μπορεί να υπάρχουν κι άλλοι κόμβοι ενδιάμεσα
    - Ο πίνακας δακτύλων δεν περιέχει όλους τους κόμβους
  - Σύγκριση με την πραγματική τιμή του δακτύλου
    - Είναι ο successor της ονομαστικής τιμής
    - Μπορεί να βρίσκεται μετά την ονομαστική τιμή

Σύνθετη αναζήτηση (5 από 7)



N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

N42+1	N48
N42+2	N48
N42+4	N48
N42+8	N51
N42+16	N1
N42+32	N14

- Παράδειγμα σύνθετης αναζήτησης

Σύνθετη αναζήτηση (6 από 7)

- Βασικές επισημάνσεις
  - Δεν συγκρίνουμε με $n+2^{i-1}$
  - Συγκρίνουμε με successor($n+2^{i-1}$)
  - Τα άλματα γίνονται προς έναν προκάτοχο του k
  - Στη χειρότερη περίπτωση, μόνο ο successor
- Κόστος σύνθετης αναζήτησης
  - Έστω ότι οι πίνακες δακτύλων είναι ενημερωμένοι
    - Αλλιώς η αναζήτηση θα είναι πιο αργή
  - Κάθε άλμα μειώνει απόσταση τουλάχιστον στο μισό
  - Το πλήθος των βημάτων είναι ανάλογο του $\log_2 N$

Σύνθετη αναζήτηση (7 από 7)

- Συμπίεση πινάκων δακτύλων
  - Πολλά δάκτυλα είναι ίδια
    - Αναμενόμενο πλήθος διαφορετικών: $\log_2 N$
  - Αποθήκευση μόνο διαφορετικών δακτύλων
    - Αρκεί να τα έχουμε στη σειρά
    - Μπορούν να είναι σε πίνακα
    - Τα εξετάζουμε πάντα γραμμικά
    - Δεν χρειάζεται καν να χωράνε όλα!

Προσχώρηση στο Chord

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Προσχώρηση κόμβων (1 από 6)

- Απαιτούμενες εργασίες στην προσχώρηση
  - Τροποποίηση δακτυλίου στο σημείο εισαγωγής
    - Γίνεται άμεσα για να επιτυγχάνουν οι αναζητήσεις
    - Αλλαγές σε προηγούμενο και επόμενο κόμβο
  - Ανακατανομή κλειδιών στους νέους γείτονες
    - Μπορεί να γίνει όταν ολοκληρωθεί η τροποποίηση
    - Εξαρτάται από την εφαρμογή
    - Ενδιάμεσα, οι αναζητήσεις προωθούνται σε γείτονες

Προσχώρηση κόμβων (2 από 6)

- Απαιτούμενες εργασίες κατά την προσχώρηση
  - Ενημέρωση πινάκων δακτύλων στο δακτύλιο
    - Δεν είναι απαραίτητο να γίνει άμεσα
    - Η αναζήτηση λειτουργεί με μη ενημερωμένους πίνακες
    - Γίνεται περιοδικά αφού επηρεάζει πολλούς κόμβους
    - Κάθε κόμβος συντηρεί τον πίνακά του

Προσχώρηση κόμβων (3 από 6)

- Ξεκινάμε από έναν τυχαίο κόμβο
  - Εντοπίζουμε κάποιον κόμβο στο δακτύλιο
  - Υπάρχουν διάφορες τεχνικές εντοπισμού
  - Κεντρικός εξυπηρετητής που γράφονται οι κόμβοι
    - Παρόμοιος με tracker στο BitTorrent
  - Πολυεκπομπή προς γνωστή διεύθυνση
    - Δεν έχει κεντρικό εξυπηρετητή
    - Λειτουργεί μόνο σε τοπικό δίκτυο

Προσχώρηση κόμβων (4 από 6)

- Δημιουργία νέου δακτυλίου από τον n
  - Όταν δεν υπάρχει κανείς στο δακτύλιο
  - Ως successor ορίζουμε τον εαυτό μας
  - Η predecessor δείχνει στον προκάτοχό μας
    - Αόριστη: εκκρεμεί ανακατανομή κλειδιών

```
n.create()
   predecessor = null
   successor = n
```

Προσχώρηση κόμβων (5 από 6)

- Προσχώρηση του κόμβου n σε δακτύλιο
  - Έστω ότι εντοπίσαμε τον κόμβο n’
  - Ζητάμε από τον n’ να βρει το διάδοχό μας
  - Πρέπει να μπούμε ακριβώς πριν από αυτόν
  - Ορισμός μόνο της successor
    - Αλλά όχι της predecessor
    - Γιατί εκκρεμεί η ανακατανομή κλειδιών

Προσχώρηση κόμβων (6 από 6)

- Προσχώρηση του κόμβου n σε δακτύλιο
  - Τώρα μπορούμε να ξεκινήσουμε αναζητήσεις
  - Ακόμη δεν είμαστε πλήρως συνδεδεμένοι
    - Δεν μας γνωρίζει προηγούμενος και επόμενος
    - Δεν μπορούμε να κάνουμε ανακατανομές κλειδιών

```
n.join(n’)
  predecessor = null;
  successor = n’.find_successor(n);
```

Σταθεροποίηση (1 από 4)

- Περιοδική σταθεροποίηση δακτυλίου
  - Ολοκληρώνει την προσχώρηση
  - Αποτελείται από δύο κλήσεις
  - Έλεγχος του predecessor του successor
    - Κανονικά πρέπει να είμαστε εμείς
    - Αν είναι μετά από εμάς, τον κάνουμε επόμενο
  - Πάντα ενημερώνουμε τον successor
    - Του αναφέρουμε ότι είμαστε ο προηγούμενός του
    - Είτε έχουμε αλλάξει successor είτε όχι

Σταθεροποίηση (2 από 4)

- Περιοδική σταθεροποίηση δακτυλίου

```
n.stabilize()
    x = successor.predecessor;
    Αν το x περιέχεται στο διάστημα (n, successor)
      successor = x
    successor.notify(n);
```

Σταθεροποίηση (3 από 4)

- Ενημέρωση του successor
  - Ποιος είναι ο νέος του predecessor
  - Αν δεν είχε predecessor δέχεται τη νέα τιμή
  - Αλλιώς ελέγχει αν είναι στο σωστό διάστημα
    - Αν δεν έχουμε αλλαγή, δεν γίνεται τίποτα
  - Αν έχουμε αλλαγή, ανακατανομή κλειδών
    - Η ανακατανομή θα γίνει από την εφαρμογή
    - Αφορά τον κόμβο και τον νέο predecessor του

Σταθεροποίηση (4 από 4)

- Ενημέρωση του successor

n.notify(n’)

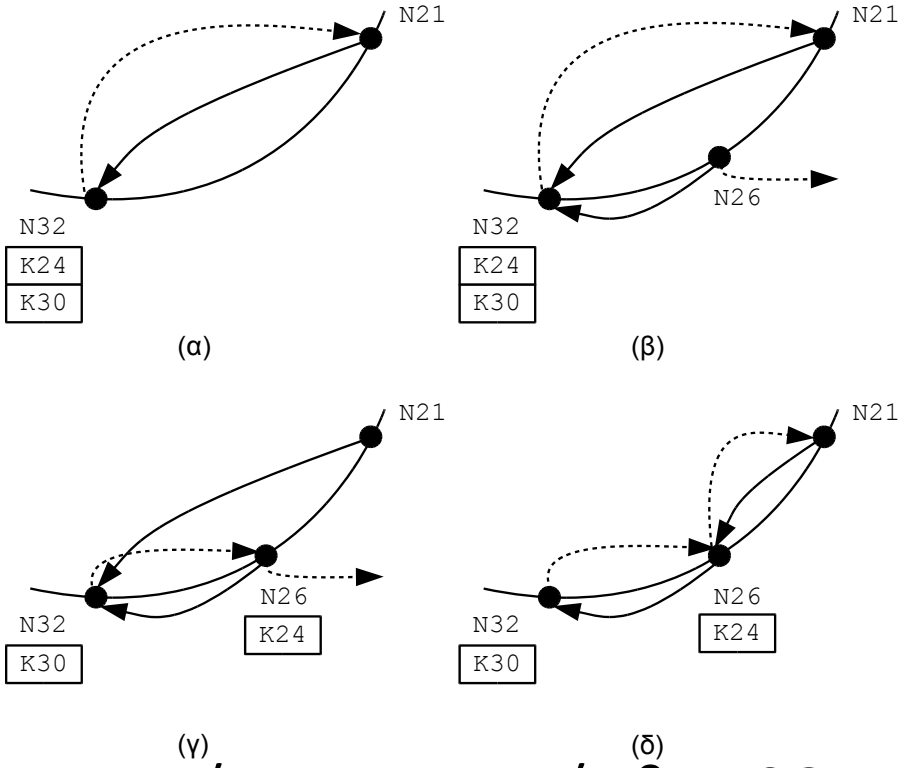
An predecessor == null ή

το n’ περιέχεται στο διάστημα (predecessor,n)

predecessor = n’

redistribute_keys(n’)

Παράδειγμα



- Παράδειγμα προσχώρησης του κόμβου 26
  - Συνδέσεις με περιοδικές κλήσεις stabilize/notify
  - Το stabilize του 26 οδηγεί σε notify στον 32
  - Το stabilize του 21 οδηγεί σε notify στον 26

Ενημέρωση δακτύλων (1 από 2)

- Ενημέρωση πίνακα δακτύλων
  - Περιοδική κλήση της fix_fingers
  - Δημιουργία νέου πίνακα/ενημέρωση υπάρχοντα
  - Κυκλική ενημέρωση των δακτύλων ενός κόμβου
  - Τα δάκτυλα διατρέχονται σε αύξουσα σειρά
    - Αξιοποιούμε το προηγούμενο για το επόμενο
    - Λειτουργεί και με λίστα αντί για πίνακα
  - Ενημέρωση μερικών δακτύλων κάθε φορά
    - Ο πίνακας δακτύλων είναι απλά βελτιστοποίηση
    - Δεν είναι απαραίτητο να είναι πάντα ενημερωμένος

Ενημέρωση δακτύλων (2 από 2)

- Ενημέρωση πίνακα δακτύλων

  n.fix_fingers()

  next = (next + 1) mod m

  finger[next] = find_successor(n + 2^{next-1});

Έλεγχος προκατόχου

- Περιοδικός έλεγχος προκατόχου
  - Για να κλείνουν τα κενά μετά από αποτυχίες
    - Αλλιώς ο διάδοχος δεν θα αλλάξει προκάτοχο!
  - Ο νέος προκάτοχος βρίσκεται με την stabilize/notify

n.check_predecessor()

Αν ο predecessor απέτυχε

predecessor = null

Αποτυχία αναζήτησης

- Αποτυχίες στην αναζήτηση
  - Η αναζήτηση επιτυγχάνει σχεδόν πάντα
    - Αρκεί οι successor είναι σωστοί
  - Αποτυχίες όταν δεν έχει κλείσει ο κύκλος
  - Οι αποτυχίες αυτές είναι παροδικές
  - Επανάληψη της αναζήτησης λίγες φορές
    - Στην επόμενη δοκιμή ίσως να έχει κλείσει ο κύκλος

Αποχώρηση από το Chord

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Αποχώρηση/αποτυχία (1 από 6)

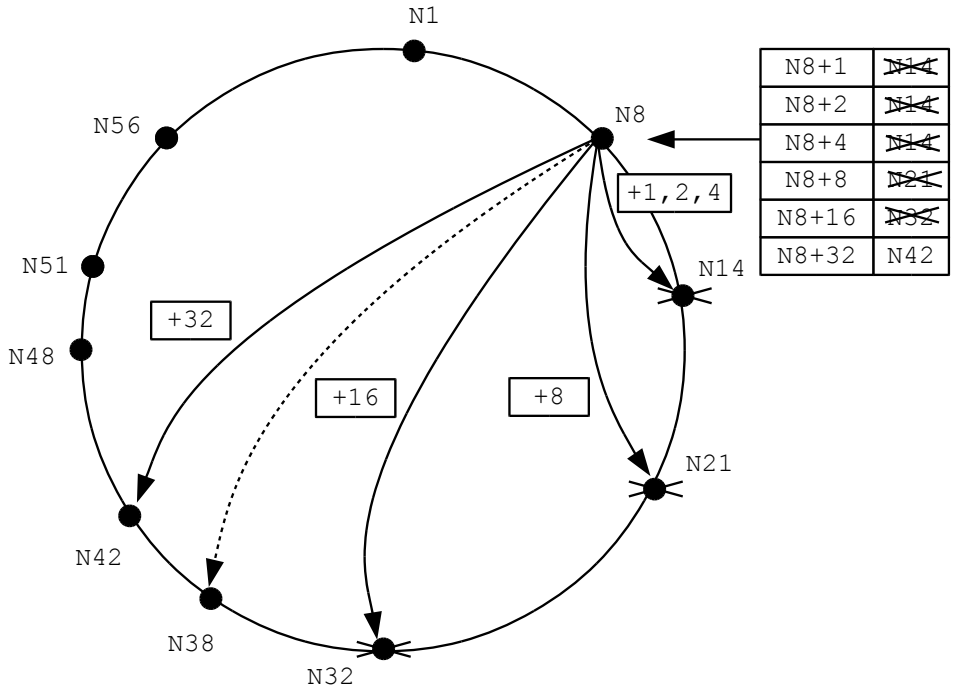
- Εντοπισμός αποτυχίας κόμβων
  - Αποτυχία των κλήσεων
  - stabilize(): αποτυχία διαδόχου
  - check_predecessor(): αποτυχία προκατόχου
  - find_successor(): αποτυχία δακτύλου
  - Το σημαντικό είναι να βρούμε νέο διάδοχο
    - Το σύστημα λειτουργεί με τους successor

Αποχώρηση/αποτυχία (2 από 6)

- Εντοπισμός διαδόχου μέσω δακτύλων
  - Διάδοχος: πρώτο δάκτυλο που δεν έχει αποτύχει

```
n.find_new_successor()
    βρέθηκε = false
    Για κάθε i από 1 έως m
      Αν ο finger[i] δεν έχει αποτύχει και !βρέθηκε
            successor = finger[i]
```

Αποχώρηση/αποτυχία (3 από 6)



- Γιατί ο πίνακας δακτύλων δεν φτάνει;
  - Μπορεί να οδηγήσει σε παράκαμψη πολλών κόμβων
  - Απαιτούνται πολλά stabilize/notify για επισκευή
    - Κάθε stabilize/notify βάζει έναν κόμβο στον δακτύλιο

Αποχώρηση/αποτυχία (4 από 6)

- Εντοπισμός διαδόχου με λίστα διαδόχων
  - Η λίστα καταγράφει τους l επόμενους κόμβους
  - Σε κάθε αποτυχία δοκιμάζουμε επόμενο στη λίστα
  - Χρειάζονται l αποτυχίες για να χαθεί ο διάδοχος
- Ιδιότητες λίστας διαδόχων
  - Ο n έχει σχεδόν την ίδια λίστα με τον διάδοχό του n’
  - Αφαιρούμε την τελευταία καταχώρηση του n’
  - Προσθέτουμε ως πρώτη καταχώρηση τον n’

Αποχώρηση/αποτυχία (5 από 6)

- Συντήρηση λίστας διαδόχων
  - Η λίστα κάθε κόμβου βασίζεται σε αυτή του διαδόχου
  - Σε αλλαγή διαδόχου ζητάμε τη λίστα του νέου διάδοχου
  - Σταδιακή κατασκευή λιστών όσο μεγαλώνει ο δακτύλιος
- Αξιοποίηση λίστας διαδόχων
  - Χρήση για να συμπληρώσει τον πίνακα δακτύλων
  - closest_preceding_node(): καλύτερος στόχος
  - find_successor(): όταν ο διάδοχος έχει αποτύχει

Αποχώρηση/αποτυχία (6 από 6)

- Αποχώρηση κόμβου: παρόμοια με αποτυχία
  - Καλύτερα να ενημερώσουμε το σύστημα όμως!
    - Αλλιώς θα έχουμε αργή ανακατασκευή
    - Στο μεσοδιάστημα αποτυγχάνουν οι αναζητήσεις
  - Ενημέρωση προκατόχου για τον νέο διάδοχο
  - Ενημέρωση διαδόχου για τον νέο προκάτοχο
  - Μεταβίβαση κλειδιών στον διάδοχο
    - Γίνεται πάντα μέσω της εφαρμογής
    - Αλλιώς τα κλειδιά θα πρέπει να καταχωρηθούν ξανά

Ανάκαμψη εφαρμογής (1 από 2)

- Ανάκαμψη εφαρμογής από αποτυχίες
  - Ο αντίκτυπος εξαρτάται από την εφαρμογή
    - Έστω (κατά)μερισμός αρχείων
    - Κάποια αρχεία δεν έχουν δείκτες
    - Κάποιοι δείκτες δεν έχουν αρχεία
  - Το Chord δεν ασχολείται με τα κλειδιά
    - Δεν μπορεί να παρέχει σχετικές επανακλήσεις
    - Μπορεί να βοηθήσει με redistribute_keys()

Ανάκαμψη εφαρμογής (2 από 2)

- Ανάκαμψη εφαρμογής από αποτυχίες
  - Περιοδικός έλεγχος κλειδιών
    - Ο κόμβος που αποθηκεύει τα κλειδιά μας απέτυχε;
    - Τότε δεν μπορεί κανείς να τα βρει
    - Αν ναι, αποθήκευση εκ νέου
  - Περιοδικός έλεγχος αρχείων
    - Ο κόμβος στον οποίο δείχνει η εγγραφή μας απέτυχε;
    - Τότε τα κλειδιά μας δεν «δείχνουν» πουθενά
    - Αν ναι, διαγραφή των κλειδιών

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Το σύστημα Pastry

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Βασικά στοιχεία Pastry (1 από 2)

- Προβλήματα επίδοσης του Chord
  - Οι διαδρομές στο δίκτυο είναι μεγάλες
    - Τουλάχιστον για τον αρχικό εντοπισμό ενός κόμβου
    - Στη συνέχεια μπορούμε να χρησιμοποιήσουμε το IP
  - Στην καλύτερη περίπτωση $\log_2 N$ βήματα
    - Όταν οι πίνακες δακτύλων είναι ενημερωμένοι
  - Κάθε βήμα μπορεί να καλύπτει όλο το δίκτυο!
    - Αριθμητική απόσταση <> τοπολογική απόσταση

Βασικά στοιχεία Pastry (2 από 2)

- Το Pastry βελτιώνει την επίδοση
  - Πίνακες δρομολόγησης με βάση και τοπολογία
  - Κάθε κόμβος επιλέγει από πολλούς γείτονες
    - Στο Chord ο πίνακας δακτύλων έχει μία επιλογή
    - Ο επόμενος μπορεί να είναι πολύ μακριά
  - Επιλέγονται οι τοπολογικά πλησιέστεροι κόμβοι
    - Από τις διάφορες δυνατές επιλογές

Διεπαφή Pastry (1 από 3)

- Διεπαφή του Pastry
  - route(msg, key): εφαρμογή -> βιβλιοθήκη
    - Στέλνει το μήνυμα msg στον πλησιέστερο κόμβο στο key
    - Ο κόμβος αυτός είναι υπεύθυνος για το κλειδί key
  - Διαφορετικές κλήσεις για τελικό / ενδιάμεσους
    - Οι ενδιάμεσοι μπορούν να επεξεργάζονται το μήνυμα
  - deliver(msg, key): βιβλιοθήκη -> εφαρμογή
    - Καλείται για να παραδοθεί το msg με κλειδί key
    - Ο παρών κόμβος είναι ο πλησιέστερος στο key

Διεπαφή Pastry (2 από 3)

- Διεπαφή του Pastry
  - forward(msg, key): βιβλιοθήκη -> εφαρμογή
    - Καλείται πριν προωθηθεί το msg με κλειδί key
    - Ο κόμβος είναι στη διαδρομή προς τον προορισμό
  - newleafs(leafset): βιβλιοθήκη -> εφαρμογή
    - Καλείται όταν αλλάζει το σύνολο φύλλων σε leafset
    - Σημαίνει ότι γίνονται αλλαγές στη γειτονιά του κόμβου
    - Μπορεί να αλλάζει και η περιοχή κλειδιών του
    - Η εφαρμογή χειρίζεται το γεγονός κατάλληλα

Διεπαφή Pastry (3 από 3)

- Διεπαφή του Pastry
  - distance(y): βιβλιοθήκη -> εφαρμογή
    - Επιστρέφει την τοπολογική απόσταση του κόμβου y
    - Αναφέρεται στο υποκείμενο δίκτυο
    - Μπορεί να χρησιμοποιεί όποια μετρική θέλει η εφαρμογή
    - Χρησιμοποιείται για την επιλογή γειτόνων
  - Η διεπαφή είναι διαφορετική από του Chord
    - Παράδοση μηνυμάτων αντί για εντοπισμός κόμβων
    - Εύκολη μετατροπή από τη μία μορφή στην άλλη

Δρομολόγηση στο Pastry

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Κόμβοι και κλειδιά

- Παρόμοια οργάνωση με το Chord
  - Κόμβοι και κλειδιά οργανώνονται σε δακτύλιο
    - Η αρχική υλοποίηση χρησιμοποιούσε m=128 bit
  - Τα αναγνωριστικά/κλειδιά ερμηνεύονται διαφορετικά
    - Πολυψήφιοι αριθμοί με βάση 2^b
    - Το b είναι παράμετρος του συστήματος, τυπικά 2 ή 4
    - Με b=4 έχουμε δεκαεξαδικά ψηφία
    - Αναγνωριστικά/κλειδιά m/b ψηφίων
  - Η περιοχή ευθύνης είναι επίσης διαφορετική
    - Ο κόμβος είναι υπεύθυνος για αριθμητικά πλησιέστερα κλειδιά

Κατάσταση κόμβων (1 από 5)

- Κατάσταση κόμβων του Pastry
  - Πίνακας δρομολόγησης
  - Σύνολο φύλλων
  - Σύνολο γειτόνων
- Πίνακας δρομολόγησης
  - Παραλλαγή πίνακα δακτύλων με δύο διαστάσεις
  - Μία γραμμή ανά ψηφίο του αναγνωριστικού
  - Μία στήλη για κάθε τιμή ψηφίου

Κατάσταση κόμβων (2 από 5)

0	0x	1x	2x	3x	4x	5x		7x	8x	9x	ax	bx	cx	dx	ex	fx
1	60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex	6fx
2	650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex	65fx
3	65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex	65afx
4	65a10x	65a11x	65a12x	65a13x	65a14x	65a15x	65a16x	65a17x	65a18x	65a19x	65a1ax	65a1bx	65a1cx	65a1dx	65a1ex	

- Πίνακας δρομολόγησης

Κατάσταση κόμβων (3 από 5)

- Πίνακας δρομολόγησης
  - Παράδειγμα: κόμβος 65a1fc με m=24 και b=4
  - Σε πλήρη μορφή 6 γραμμές και 16 στήλες
  - Η r περιέχει κόμβους τα αναγνωριστικά των οποίων
    - Στα r πρώτα ψηφία τους ταυτίζονται με τον τρέχοντα
    - Στο ψηφίο r+1 διαφοροποιούνται από τον τρέχοντα
    - Ο κόμβος στη στήλη c έχει το ψηφίο r+1 ίσο με c
    - Η στήλη που αντιστοιχεί στο ψηφίο του κόμβου είναι κενή
    - Τα επόμενα ψηφία των κόμβων δεν έχουν σημασία

Κατάσταση κόμβων (4 από 5)

- Διαφοροποίηση Pastry από Chord
  - Αν b=1 μοιάζει με τον πίνακα δακτύλων
  - Στο Chord η επιλογή κάθε γραμμής είναι μονοσήμαντη
  - Στο Pastry έχουμε πολλές επιλογές όμως
    - Επιλέγεται ο τοπολογικά πλησιέστερος κόμβος
- Πόσο μεγάλος είναι ο πίνακας δρομολόγησης;
  - Όπως στο Chord, ο πίνακας δρομολόγησης είναι αραιός
    - Τα αναγνωριστικά κόμβων είναι αραιά κατανεμημένα
  - Στις τελευταίες γραμμές θα έχουμε πολλά κενά
  - Κατά μέσο όρο έχει μόνο $\log_2{}^b N(2^b-1)$ καταχωρίσεις

Κατάσταση κόμβων (5 από 5)

- Σύνολο φύλλων
  - Περιέχει L αριθμητικά πλησιέστερους κόμβους
    - L/2 αμέσως προηγούμενοι και L/2 αμέσως επόμενοι
    - Τυπικά το L είναι 2^b ή $2x2^b$
  - Δείχνει αν ένα κλειδί είναι στην περιοχή ευθύνης
- Σύνολο γειτόνων
  - Περιέχει K τοπολογικά πλησιέστερους κόμβους
    - Τυπικά το K είναι 2^b ή $2x2^b$

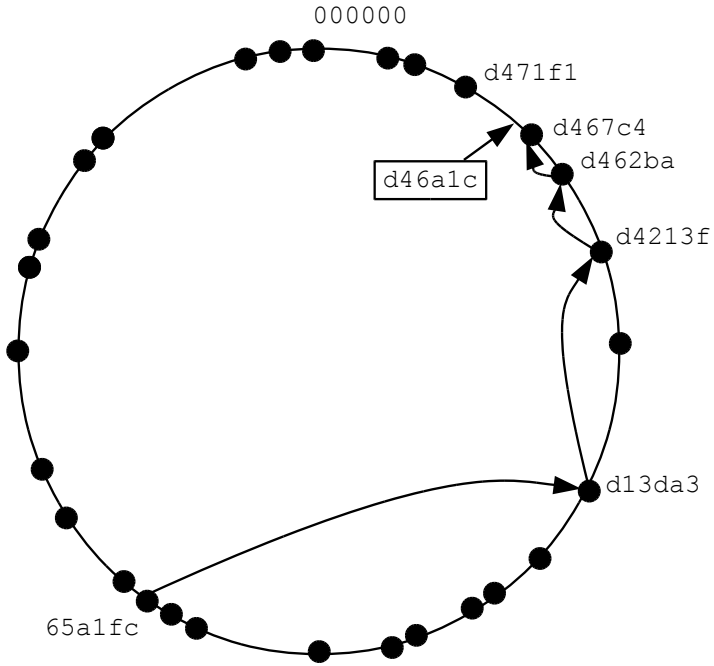
Δρομολόγηση (1 από 5)

- Δρομολόγηση μηνύματος με κλειδί key
  - Αρχικά εξετάζεται το σύνολο φύλλων
    - Το key βρίσκεται ανάμεσα στα δύο ακραία φύλλα;
    - Αν ναι, τότε εντοπίζεται ο κόμβος στον οποίο ανήκει
    - Το μήνυμα προωθείται στον κόμβο αυτό
    - Ο κόμβος αυτός μπορεί να είναι και ο τοπικός
  - Στη συνέχεια εξετάζεται ο πίνακας δρομολόγησης
    - Έστω ότι το key ταυτίζεται σε r αρχικά ψηφία με κόμβο
    - Έστω ότι το ψηφίο r+1 είναι ίσο με c
    - Το μήνυμα προωθείται στον κόμβο στη θέση (r, c)

Δρομολόγηση (2 από 5)

- Δρομολόγηση μηνύματος με κλειδί key
  - Τι γίνεται αν η θέση (r,c) είναι κενή
    - Ελέγχουμε και τον πίνακα
    - Ελέγχουμε και τα σύνολα φύλλων και γειτόνων
    - Υπάρχει αριθμητικά πλησιέστερος κόμβος
    - Που να ταυτίζεται σε r ψηφία;
    - Αν όχι, τότε ο πλησιέστερος είναι ο τρέχων κόμβος

Δρομολόγηση (3 από 5)



- Παράδειγμα: δρομολόγηση του κλειδιού d46a1c
  - Στέλνεται από τον κόμβο 65a1fc
  - Καταλήγει στον κόμβο d467c4
  - Τελικά πέφτουμε μέσα στο σύνολο φύλλων

Δρομολόγηση (4 από 5)

- Επίδοση δρομολόγησης
  - Έστω πίνακες δρομολόγησης ενημερωμένοι
  - Αναμενόμενο πλήθος βημάτων $\log_2{}^b N$
    - Σε κάθε βήμα υποδιαιρούμε τις καταχωρίσεις κατά 2^b
  - Το σύνολο φύλλων απαιτεί ένα ακόμα βήμα
  - Πολύ σπάνια χρειάζεται άλλο ένα βήμα
    - Όταν ο πίνακας δρομολόγησης έχει κενά

Δρομολόγηση (5 από 5)

- Σύγκριση με το Chord
  - Όταν b=1 Pastry και Chord απαιτούν $\log_2 N$
    - Το ίδιο ισχύει και για την κατάσταση ανά κόμβο
  - Αυξάνοντας το b μειώνουμε τα βήματα
    - Ταυτόχρονα όμως αυξάνεται η κατάσταση
  - Το b επιτρέπει να πετύχουμε συμβιβασμό
    - Πιο γρήγορη αναζήτηση ή λιγότερη κατάσταση;

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Προσχώρηση/αποχώρηση στο Pastry

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Προσχώρηση κόμβων (1 από 3)

- Προσχώρηση νέων κόμβων
  - Αρχικά πρέπει να εντοπιστεί κάποιος κόμβος
    - Με τον ίδιο τρόπο όπως και στο Chord
  - Αν εντοπιστούν πολλοί, ο τοπολογικά πλησιέστερος
  - Έστω ότι ο νέος κόμβος είναι ο X και επέλεξε τον A
  - Ο X ζητά από τον A να δρομολογήσει το μήνυμα join
    - Το join έχει ως κλειδί το X και περιέχει τη διεύθυνση του X
  - Έστω ότι ακολουθεί τη διαδρομή A,B,C,…,Z
    - Ο Z είναι ο αριθμητικά πλησιέστερος κόμβος στο X

Προσχώρηση κόμβων (2 από 3)

- Προσχώρηση νέων κόμβων
  - Κάθε κόμβος στέλνει την κατάστασή του στον Χ
    - Ο Χ χρησιμοποιεί τα στοιχεία για να φτιάξει κατάσταση
  - Το σύνολο γειτόνων αρχικοποιείται σε αυτό του Α
  - Το σύνολο φύλλων αρχικοποιείται σε αυτό του Z
    - Ο Α είναι τοπολογικά κοντά και ο Z είναι αριθμητικά κοντά
  - Ο πίνακας δρομολόγησης αρχικοποιείται
    - Ως γραμμή 0 χρησιμοποιούμε την αντίστοιχη γραμμή του Α
    - Ως γραμμή 1 χρησιμοποιούμε την αντίστοιχη γραμμή του Β

Προσχώρηση κόμβων (3 από 3)

- Προσχώρηση νέων κόμβων
  - Ο Χ στέλνει την κατάστασή του στους Α,Β,C,…,Z
    - Κάθε κόμβος διορθώνει την κατάστασή του
    - Ο Z θα πρέπει να προσθέσει τον Χ στο σύνολο φύλλων
    - Ο Α ίσως θα πρέπει να προσθέσει τον Χ στο σύνολο γειτόνων
  - Τα αρχικά μηνύματα περιέχουν χρονοσφραγίδες
    - Ο Χ βάζει την ίδια χρονοσφραγίδα στα δικά του μηνύματα
    - Αν κάποιος κόμβος έχει αλλάξει κατάσταση την ξαναστέλνει
    - Ο Χ απαντά ξανά με τη νέα χρονοσφραγίδα
    - Αντιμετώπιση ταυτόχρονων προσχωρήσεων / αποχωρήσεων

Αποχώρηση κόμβων (1 από 3)

- Αποχώρηση και αποτυχία κόμβων
  - Αντιμετωπίζονται με τον ίδιο ακριβώς τρόπο
- Επιδιόρθωση πίνακα γειτόνων
  - Περιοδικός έλεγχος όλων των γειτόνων
    - Η δρομολόγηση δεν περιλαμβάνει πίνακες γειτόνων
  - Έστω ότι δεν αποκρίνεται κάποιος γείτονας
    - Ζητάμε από τους γείτονες τα δικά τους σύνολα
    - Επιλέγουμε από αυτά έναν νέο κόμβο για το δικό μας

Αποχώρηση κόμβων (2 από 3)

- Επιδιόρθωση συνόλου φύλλων
  - Έστω ότι ο Χ παρατηρεί ότι δεν αποκρίνεται το φύλλο Υ
    - Αυτό γίνεται όταν προσπαθήσουμε να του στείλουμε κάτι
  - Αρχικά ο Χ ζητάει το σύνολο φύλλων του Z
    - Ο Z είναι το ακραίο φύλλο προς την κατεύθυνση του Υ
    - Σε μεγάλο βαθμό τα δύο σύνολα φύλλων επικαλύπτονται
  - Ο Χ επιδιορθώνει το σύνολο φύλλων με βάση αυτό του Z
    - Με τον ίδιο τρόπο ενημερώνονται όλα τα σύνολα φύλλων
  - Για αποτυχία πρέπει να χαθούν L/2 διαδοχικοί κόμβοι
    - Πρέπει να χαθεί όλη η μία πλευρά του συνόλου φύλλων

Αποχώρηση κόμβων (3 από 3)

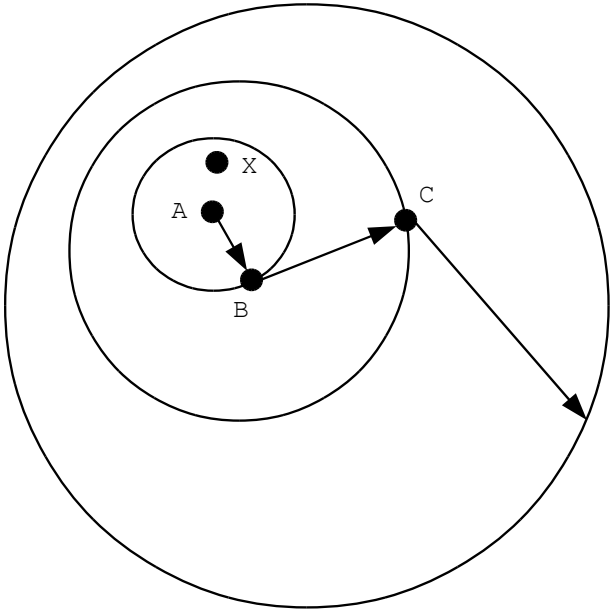
- Επιδιόρθωση πινάκων δρομολόγησης
  - Έστω ότι δεν αποκρίνεται η θέση (r,c)
    - Όταν προσπαθήσουμε να του στείλουμε κάτι
    - Προσωρινά παρακάμπτουμε τον κόμβο αυτόν
  - Επικοινωνούμε με άλλο κόμβο της γραμμής r
    - Ζητάμε τη δική του καταχώριση σε αυτή τη θέση
    - Αν δεν υπάρχει, επαναλαμβάνουμε με τη r+1

Επίδοση του Pastry

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Βελτιστοποίηση διαδρομών (1 από 4)



- Πόσο καλοί είναι οι πίνακες δρομολόγησης;
  - Η αρχική κατασκευή των πινάκων είναι αρκετά καλή
    - Οι γείτονες του A θα είναι και γείτονες του X
    - Η γραμμή 0 του A θα είναι καλή επιλογή για τον X

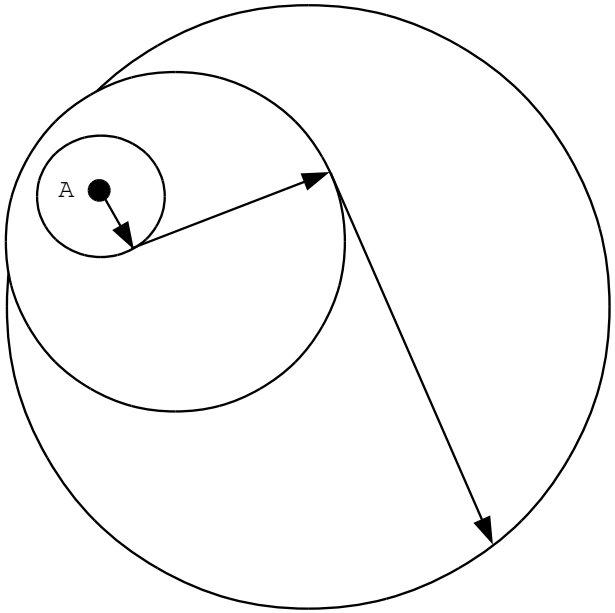
Βελτιστοποίηση διαδρομών (2 από 4)

- Πόσο καλοί είναι οι πίνακες δρομολογησης;
  - Τι γίνεται με τις υπόλοιπες γραμμές του πίνακα;
  - Ο Β είναι κοντά στον Α αλλά όχι και στον Χ
  - Όμως στη γραμμή 1 έχουμε λιγότερες επιλογές
    - Οι υποψήφιοι κόμβοι είναι πολύ λιγότεροι
  - Άρα οι κόμβοι στην 1 θα είναι γενικά πιο μακριά
  - Συνεπώς μπορεί να χρησιμοποιηθεί για τον Χ
  - Με την ίδια λογική φτιάχνονται όλες οι γραμμές

Βελτιστοποίηση διαδρομών (3 από 4)

- Σταδιακή βελτίωση των πινάκων
  - Περιοδικά κάθε κόμβος ζητάει την κατάσταση άλλων
    - Όσων είναι σε πίνακα δρομολόγησης και σύνολο γειτόνων
  - Όποτε βρει καλύτερες επιλογές τις χρησιμοποιεί
    - Για τον πίνακα δρομολόγησης και το σύνολο γειτόνων
  - Διαλέγουμε πάντα τοπολογικά πλησιέστερο κόμβο
    - Χρησιμοποιούμε την distance για να επιλέξουμε

Βελτιστοποίηση διαδρομών (4 από 4)



- Πόσο καλές είναι οι συνολικές διαδρομές;
  - Κάθε βήμα δρομολόγησης είναι καλό από μόνο του
  - Η συνολική δρομολόγηση είναι αρκετά καλή επειδή
    - Σε κάθε βήμα δεν μπορούμε να μπούμε στον προηγούμενο κύκλο
    - Κάθε βήμα δημιουργεί έναν πολύ μεγαλύτερο κύκλο

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Το σύστημα Scribe

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Εισαγωγή στο Scribe (1 από 4)

- Πολυεκπομπή στο Scribe
  - Αξιοποιεί δρομολόγηση με βάση το κλειδί
  - Σχεδιάστηκε για το Pastry
    - Διεπαφή δρομολόγησης μηνυμάτων
  - Μπορεί να χρησιμοποιηθεί με το Chord
    - Πρέπει όμως να τροποποιηθεί η διεπαφή

Εισαγωγή στο Scribe (2 από 4)

- Ομάδες πολυεκπομπής
  - Προσδιορίζονται από ένα κλειδί
    - Όλες οι λειτουργίες γίνονται με βάση το κλειδί
  - Δημιουργία ομάδας
  - Προσχώρηση και αποχώρηση κόμβων
  - Αποστολή δεδομένων στην ομάδα
    - Τα δεδομένα λαμβάνονται από όλα τα μέλη

Εισαγωγή στο Scribe (3 από 4)

- Κλειδιά ομάδων
  - Παράγονται με κατακερματισμό
    - MD5, SHA-1, κ.λπ.
  - Ίδιο πεδίο με τα κλειδιά των κόμβων
  - Αυθαίρετα ονόματα ομάδων
    - Ο κατακερματισμός τα κάνει ομοιόμορφα
    - Συνυπάρχουν διάφοροι τύποι ονομάτων
    - Παράδειγμα: ιεραρχικά, επίπεδα

Εισαγωγή στο Scribe (4 από 4)

- Scribe: βιβλιοθήκη πάνω από το Pastry
  - Τέσσερις βασικές κλήσεις
  - create(key): δημιουργία
  - join(key,handler): προσχώρηση
    - Τα μηνύματα προωθούνται στη handler
  - leave(key): αποχώρηση
  - multicast(key,msg): αποστολή
  - Κάθε κλήση παράγει μήνυμα με το ίδιο όνομα

Δημιουργία δένδρου (1 από 4)

- Σημείο ραντεβού (rendezvous) της ομάδας
  - Κόμβος με αριθμητικά πλησιέστερο κλειδί
  - Τα μηνύματα δρομολογούνται εκεί
    - create, join, leave, multicast
  - Λειτουργεί ως ρίζα δένδρου πολυεκπομπής
    - Προωθεί τα μηνύματα στους παραλήπτες
  - Μαθαίνει ότι είναι σημείο ραντεβού στο create

Δημιουργία δένδρου (2 από 4)

- Επεξεργασία μηνυμάτων
  - Τα μηνύματα στον παραλήπτη
    - Το μήνυμα multicast έχει πολλούς παραλήπτες
    - Το σημείο ραντεβού και τα μέλη της ομάδας
  - Τα join/leave και στους ενδιάμεσους
    - Χρησιμοποιούνται για την κατασκευή του δένδρου
  - Το create μόνο στο σημείο ραντεβού
    - Δημιουργεί τη ρίζα του δένδρου

Δημιουργία δένδρου (3 από 4)

- Κατάσταση κόμβων δένδρου πολυεκπομπής
  - Πίνακας προώθησης ανά ομάδα
    - Περιέχει τα παιδιά του κόμβου στο δένδρο
  - Αρχικά είναι κενός
  - Προσθήκη παιδιών: μηνύματα join
    - Τα μηνύματα join δημιουργούν το δένδρο
  - Διαγραφή παιδιών: μηνύματα leave
    - Τα μηνύματα leave καταστρέφουν το δένδρο

Δημιουργία δένδρου (4 από 4)

- Δημιουργία ομάδας
  - Μήνυμα create στο σημείο ραντεβού
    - Το επεξεργάζεται μόνο το σημείο ραντεβού
    - Δρομολόγηση μέσω του DHT
  - Το σημείο δημιουργεί πίνακα προώθησης
    - Αρχικά ο πίνακας είναι κενός
    - Θα προστεθούν παιδιά σταδιακά
  - Έλεγχος δικαιωμάτων δημιουργίας (προαιρετικά)

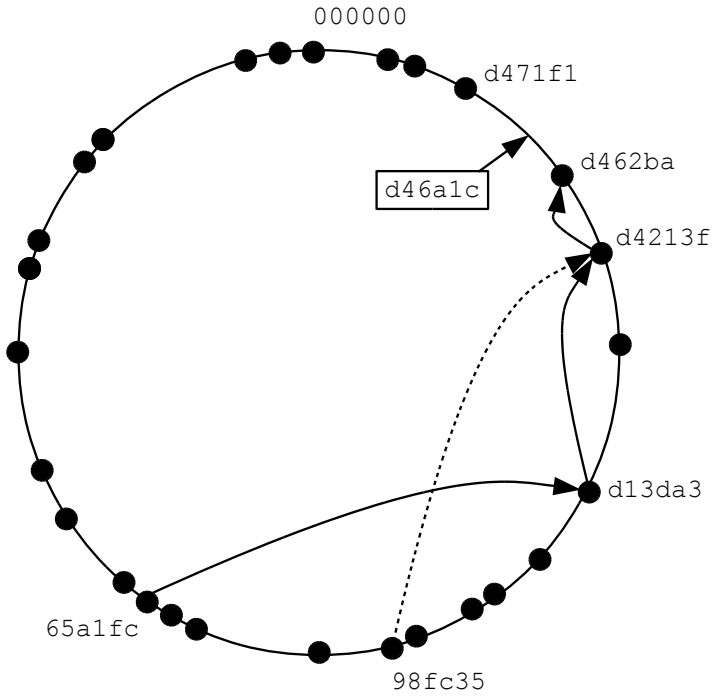
Προσχώρηση (1 από 3)

- Προσχώρηση σε ομάδα
  - Αποστολή join στο σημείο ραντεβού
    - Δρομολόγηση μέσω του DHT
  - Κάθε ενδιάμεσος κόμβος εξετάζει το μήνυμα
  - Αν έχει πίνακα προώθησης για την ομάδα
    - Προσθέτει τον αποστολέα στον πίνακα
    - Δεν προωθεί το join παραπέρα
    - Ο αποστολέας έχει συνδεθεί στο δένδρο

Προσχώρηση (2 από 3)

- Προσχώρηση σε ομάδα
  - Αν δεν έχει πίνακα προώθησης για την ομάδα
    - Δημιουργεί τον πίνακα προώθησης
    - Προσθέτει τον αποστολέα στον πίνακα
    - Αλλάζει τον αποστολέα του join στον εαυτό του
    - Προωθεί το join παραπέρα
  - Σταδιακή δημιουργία δένδρου πολυεκπομπής
    - Από τα φύλλα προς το σημείο ραντεβού

Προσχώρηση (3 από 3)



- Παράδειγμα προσχώρησης
  - Ομάδα d46a1c, προσχωρούν 65a1fc και 98fc35
  - Το δεύτερο join δεν φτάνει στο σημείο ραντεβού
    - Σταματάει στον πρώτο κόμβο που είναι ήδη στο δένδρο

Αποχώρηση

- Αποχώρηση από ομάδα
  - Αποστολή leave στο σημείο ραντεβού
  - Κάθε ενδιάμεσος κόμβος εξετάζει το μήνυμα
  - Αφαίρεση αποστολέα από πίνακα προώθησης
  - Αν ο πίνακα προώθησης δεν μένει κενός
    - Δεν προωθεί το leave παραπέρα
  - Αν ο πίνακα προώθησης μένει κενός
    - Αλλάζει τον αποστολέα του leave στον εαυτό του
    - Προωθεί το leave παραπέρα

Αποστολή μηνυμάτων

- Αποστολή μηνυμάτων στην ομάδα
  - Αποστολή multicast στο σημείο ραντεβού
  - Το σημείο ραντεβού προωθεί το μήνυμα
    - Χρήση πάντα του DHT
  - Αναδρομική προώθηση μέχρι τα φύλλα
  - Σε κάθε μέλος καλείται η μέθοδος handler
    - Ουσιαστικά είναι η μέθοδος λήψης μηνυμάτων
    - Μπορεί να περάσουμε πολλά μέλη στη διαδρομή

Απόδοση (1 από 2)

- Πόσο αποδοτική είναι η πολυεκπομπή;
  - Το δένδρο είναι ένωση μονοπατιών
  - Τα μονοπάτια είναι αντίστροφα των επιθυμητών
    - Από τα μέλη της ομάδας προς το σημείο ραντεβού
    - Μπορεί να μην είναι συμμετρικά
  - Τα μονοπάτια προς τα μέλη δεν είναι βέλτιστα
    - Ως προς το υποκείμενο δίκτυο
    - Το Pastry προσεγγίζει τα τοπολογικά βέλτιστα

Απόδοση (2 από 2)

- Πόσο αποδοτική είναι η πολυεκπομπή;
  - Το σημείο ραντεβού δεν είναι βέλτιστο
    - Τα μηνύματα κινούνται πρώτα προς σημείο ραντεβού
    - Μπορεί να απομακρύνονται από τα μέλη της ομάδας
  - Το μονοπάτι προς το ραντεβού δεν είναι βέλτιστο
    - Ο αποστολέας μπορεί να αποθηκεύσει τη διεύθυνση
    - Στη συνέχεια αποστολή μηνυμάτων μέσω IP

Αποτυχίες (1 από 2)

- Αποτυχία / αποχώρηση εσωτερικών κόμβων
  - Κάθε κόμβος στέλνει περιοδικά μηνύματα
    - Σε όλα τα παιδιά του
  - Αν ένα παιδί δεν λάβει μήνυμα για αρκετή ώρα
    - Υποθέτει ότι ο πατέρας του έχει αποτύχει
    - Στέλνει νέο join προς το σημείο ραντεβού
    - Το παιδί επανασυνδέεται με το δένδρο

Αποτυχίες (2 από 2)

- Αποτυχία ή αποχώρηση σημείου ραντεβού
  - Το σημείο ραντεβού αναπαράγει την κατάστασή του
    - Περιοδική αποστολή κατάστασης στο σύνολο φύλλων
    - Αν αποτύχει, κάποιο φύλλο θα γίνει σημείο ραντεβού
  - Τα παιδιά παρατηρούν την αποτυχία
    - Δεν λαμβάνουν μήνυμα από τον πατέρα
    - Στέλνουν νέο μήνυμα join προς το σημείο ραντεβού
    - Τα μηνύματα join καταλήγουν στο νέο σημείο ραντεβού
  - Παρόμοια αντιμετώπιση από τους αποστολείς

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τέλος Ενότητας # 9

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 9: Κατανεμημένοι πίνακες κατακερματισμού

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής