

Κατανεμημένα συστήματα

Παπαφώτη Μαρία Α.Μ: 45

17/3/03

Άσκηση: Προβλήματα από το βιβλίο των Tanenbaum και Steen, κεφάλαιο 2: 3, 6, 8, 9, 15, 22, 24, και 26

(πρόβλημα: 3) Μια αξιόπιστη multicast υπηρεσία επιτρέπει σε έναν αποστολέα να περνά αξιόπιστα μηνύματα σε ένα σύνολο από παραλήπτες. Αυτή η υπηρεσία ανήκει στο επίπεδο του middleware ή θα έπρεπε να είναι μέρος ενός χαμηλότερου επιπέδου;

Μια multicast υπηρεσία θα μπορούσε να ανήκει και στο επίπεδο μεταφοράς (χαμηλότερο επίπεδο από το middleware). Δεν θα έπρεπε όμως να βρίσκεται σε ένα τόσο χαμηλό επίπεδο γιατί ενδεχομένως τότε να είχαμε προβλήματα αξιοπιστίας. Τα προβλήματα αυτά προκύπτουν λόγω έλλειψης, στα χαμηλά επίπεδα, πληροφορίας που σχετίζεται με την εφαρμογή που χρησιμοποιεί την υπηρεσία. Επομένως τέτοιες υπηρεσίες πρέπει να υλοποιούνται στο επίπεδο middleware ή ακόμα και στο επίπεδο εφαρμογής. Σε επίπεδο middleware είναι δυνατόν να υλοποιηθούν διάφορα πρωτόκολλα επικοινωνίας (με βάση τις απαιτήσεις διαφορετικών εφαρμογών) τα οποία θα βασίζονται σε διαφορετικά πρωτοκολλά του επιπέδου μεταφοράς αλλά η πρόσβαση τους θα γίνεται από ένα κοινό για όλα interface. Σε αυτή την απάντηση θα καταλήγαμε και με βάση το end to end argument.

(πρόβλημα: 6) Ένας τρόπος για να χειριστούμε την μετατροπή των παραμέτρων σε συστήματα RPC είναι κάθε μηχανή να στέλνει τις παραμέτρους στη δική της μορφή και η άλλη να κάνει την μετατροπή αν χρειάζεται. Το σύστημα που χρησιμοποιείται μπορεί να υποδεικνύεται από ένα κωδικό στο πρώτο byte. Ωστόσο αφού ο εντοπισμός του πρώτου byte στην πρώτη λέξη είναι ακριβώς το πρόβλημα, μπορεί αυτό πραγματικά να δουλέψει;

Ο εντοπισμός του πρώτου byte στην πρώτη λέξη είναι πραγματικά το πρόβλημα. Όμως αν δουλέψουμε με βάση την τεχνική που περιγράφεται από το πρόβλημα, αν δηλαδή χειριστούμε την μετατροπή των παραμέτρων σε συστήματα RPC με το να στέλνει κάθε μηχανή τις παραμέτρους στη δική της μορφή χρησιμοποιώντας το πρώτο byte για να δηλώσει αυτή τη μορφή και η άλλη να κάνει την μετατροπή αν χρειάζεται μπορεί αυτή να λειτουργήσει αποτελεσματικά. Ο λόγος που αυτή η τεχνική είναι δυνατόν να λειτουργήσει

είναι ότι αφού το πρώτο byte έχει αποφασισθεί να δηλώνει απλά το σύστημα που χρησιμοποιεί η μηχανή που στέλνει το μήνυμα ουσιαστικά δεν συνεισφέρει στο περιεχόμενο του μηνύματος και δεν μας ενδιαφέρει η θέση του μέσα σε αυτό. Αντίθετα σε αυτή την περίπτωση καθοριστικό είναι το δεύτερο byte του μηνύματος το οποίο ουσιαστικά είναι το πρώτο που συνεισφέρει στο περιεχόμενο του, και το οποίο θα μεταφραστεί σωστά από το σύστημα το οποίο θα στηριχτεί στην πληροφορία του πρώτου byte.

(πρόβλημα: 8) Αντί να επιτρέπουμε σε έναν εξυπηρέτη να καταγράφει τον εαυτό του με έναν δαίμονα όπως γίνεται στο DCE, θα μπορούσαμε επίσης να επιλέξουμε τον προσδιορισμό πάντα του ίδιου endpoint. Αυτό το endpoint θα μπορούσε στη συνέχεια να χρησιμοποιηθεί σε αναφορές σε αντικείμενα στο χώρο διευθύνσεων του εξυπηρέτη. Ποιο είναι το βασικό μειονέκτημα αυτού του σχήματος;

Το βασικό μειονέκτημα αυτού του σχήματος είναι το γεγονός ότι συσχετίζει μόνιμα ένα endpoint με μια υπηρεσία δημιουργώντας έτσι προβλήματα έλλειψης διαφάνειας και αξιοπιστίας στο σύστημα στην περίπτωση που το συγκεκριμένο endpoint δεν λειτουργεί ή δεν προσφέρει πλέον την συγκεκριμένη υπηρεσία.

(πρόβλημα: 9) Δώστε ένα παράδειγμα υλοποίησης μιας αναφοράς που επιτρέπει στον πελάτη να δεσμεύει ένα (transient) παροδικό απομακρυσμένο αντικείμενο.

Μια τέτοια υλοποίηση είναι το FTP. Σε αυτήν το απομακρυσμένο αντικείμενο είναι το αρχείο που θέλει να κατεβάσει ο πελάτης από τον εξυπηρέτη.

(πρόβλημα: 15) Υποθέστε ότι μπορείτε να χρησιμοποιήσετε μόνο παροδικά ασύγχρονα βασικά στοιχεία επικοινωνίας, συμπεριλαμβανομένου μόνο ενός ασύγχρονου receive. Πώς θα υλοποιούσατε βασικά στοιχεία για παροδική σύγχρονη επικοινωνία;

Η διαφορά ανάμεσα στα βασικά στοιχεία της παροδικής ασύγχρονης επικοινωνίας και εκείνα της παροδικής σύγχρονης επικοινωνίας είναι ότι τα τελευταία λαμβάνουν υπ' όψιν τους την κατάσταση του άλλου άκρου της επικοινωνίας. Πιο συγκεκριμένα ένα Send το οποίο χρησιμεύει για την υλοποίηση παροδικής σύγχρονης επικοινωνίας κάνει μια αίτηση στον δέκτη και μπλοκάρει τον πομπό μέχρι να έρθει απάντηση από τον πρώτο. Αντίστοιχα

ένα Receive το οποίο χρησιμεύει για την υλοποίηση παροδικής σύγχρονης επικοινωνίας μπλοκάρει τον πομπό έως ότου ο δεκτής λάβει το μήνυμα που ζήτησε καλώντας την receive. Επομένως η υλοποίηση των synchronous\_receive και synchronous\_send δεδομένων των ασύγχρονων receive και send θα ήταν κάπως έτσι (δίνεται ψευδοκώδικας):

```
synchronous_send
{
send; /* στείλε το μήνυμα */
synchronous_receive; /* περίμενε μέχρι να λάβεις απάντηση, μην κάνεις τίποτε άλλο */
}
```

```
synchronous_receive
{
while(!receive); /*περίμενε να λάβεις ένα μήνυμα, όσο δεν το λαμβάνεις μην κάνεις
τίποτα*/
block;
}
```

(πρόβλημα: 22) Εξηγήστε γιατί η παροδική (transient) σύγχρονη επικοινωνία έχει έμφυτα προβλήματα κλιμάκωσης και πως αυτά μπορούν να λυθούν.

Η transient σύγχρονη επικοινωνία έχει έμφυτα προβλήματα κλιμάκωσης αφού τέτοια προβλήματα παρουσιάζει κάθε είδος σύγχρονης επικοινωνίας. Τα προβλήματα αυτά αφορούν κυρίως τη γεωγραφική κλιμάκωση τέτοιων συστημάτων και αυτό γιατί ενώ σε μικρά δίκτυα η επικοινωνία ανάμεσα σε δυο κόμβους του δικτύου δεν διαρκεί παρά το πολύ μερικές εκατοντάδες μsecs, όσο το δίκτυο μεγαλώνει και εξαπλώνεται αυτή η επικοινωνία απαιτεί όλο και πιο πολύ χρόνο δημιουργώντας έτσι καθυστέρηση στο σύστημα. Όσο τώρα αφορά στη κλιμάκωση στο μέγεθος του δικτύου αν αυτό αυξηθεί πολύ τότε οι πολύ μικρές καθυστερήσεις είναι ικανές όλες μαζί να δημιουργήσουν πρόβλημα. Μια λύση θα ήταν να χρησιμοποιήσουμε ένα λιγότερο ισχυρό σύγχρονο μοντέλο. Μια άλλη λύση αυτών των προβλημάτων αν δεν γίνεται να εφαρμόσουμε κανένα είδος ασύγχρονης επικοινωνίας είναι προσπαθήσουμε να μειώσουμε την ανάγκη για επικοινωνία.

Αυτό, για παράδειγμα, μπορεί να γίνει αν μεταφέρουμε στους clients ένα μεγάλο ποσοστό των υπολογισμών που κανονικά θα ζητούσαν από τους servers.

(πρόβλημα: 24) Πώς μπορείτε να εγγυηθείτε μέγιστη end-to-end καθυστέρηση σε ένα σύνολο από υπολογιστές που είναι οργανωμένοι σε (λογικό ή φυσικό) δακτύλιο;

Σε ένα σύνολο από υπολογιστές που είναι οργανωμένοι σε δακτύλιο έχουμε έναν πεπερασμένο αριθμό από υπολογιστές έστω  $N$ . Η μέγιστη end-to-end καθυστέρηση σε ένα τέτοιο δίκτυο επιτυγχάνεται αν ο κόμβος  $i$  επικοινωνεί με τον κόμβο  $j_1 = i + N/2$  αν  $j_1 \leq N$  και  $j_1 = i + N/2 - N$  διαφορετικά (παίρνουμε το ανώφλι από το  $N/2$  και στις δύο περιπτώσεις). Τη μέγιστη λοιπόν καθυστέρηση μπορούμε να εγγυηθούμε αν εξασφαλίσουμε το παραπάνω σχήμα για την επικοινωνία.

(πρόβλημα: 26) Φανταστείτε ότι έχουμε ένα token bucket specification όπου το μέγιστο μέγεθος μιας μονάδας δεδομένων είναι 1000 bytes, ο ρυθμός του token bucket είναι 10 εκατομμύρια bytes/sec, το μέγεθος του token bucket είναι 1 εκατομμύριο bytes, και ο μέγιστος ρυθμός μετάδοσης είναι 50 εκατομμύρια bytes/sec. Πόσο διαρκεί ένα burst με μέγιστη ταχύτητα;

1 εκατομμύριο bytes = 1.000 μονάδες δεδομένων/ sec

10 εκατομμύρια bytes/sec = 10.000 μονάδες δεδομένων/ sec

50 εκατομμύρια bytes/sec = 50.000 μονάδες δεδομένων/ sec

Άρα ο κάδος (bucket) γεμίζει με ρυθμό 40.000 μονάδες δεδομένων/ sec

και επομένως ο ζητούμενος χρόνος είναι  $1000/40.000\text{sec} = 1/40\text{sec} = 0.025\text{ sec}$