

Άσκηση 3

Ένα τέτοιο service θα ήταν προτιμότερο να τοποθετηθεί στο middleware layer. Ο λόγος είναι ότι για τη μετάδοση πληροφορίας σε πολλούς διαφορετικούς χρήστες πρέπει να ληφθούν υπόψιν οι απαιτήσεις της εφαρμογής. Έτσι με την τοποθέτηση του service στο middleware layer δίνεται η δυνατότητα να χρησιμοποιείται διαφορετικό πρωτόκολλο μεταφοράς (ανάλογα με την εφαρμογή) ακόμα και αν χρησιμοποιείται το ίδιο interface.

Επιπλέον το κατώτερο επίπεδο δεν εξασφαλίζει ότι τα δεδομένα θα παραδοθούν σε όλους και χωρίς λάθη.

Άσκηση 6

Αν στείλουμε ένα πακέτο που έχει 1byte που δηλώνει τον τύπο κωδικοποίησης και τα υπόλοιπα τρία bytes μηδέν τότε μπορούμε να εντοπίσουμε το τύπο της κωδικοποίησης διαβάζοντας το μη-μηδενικό byte.

Άσκηση 8

Το κύριο μειονέκτημα αυτού του σχήματος είναι ότι δεσμεύεται ένα endpoint αποκλειστικά από μια διεργασία (server). Έτσι αν για κάποιο χρονικό διάστημα ο server είναι εκτός λειτουργίας θα δεσμεύεται χωρίς λόγο αυτό το port ενώ θα μπορούσε να χρησιμοποιηθεί από κάποια άλλη διεργασία.

Μπορεί το συγκεκριμένο endpoint να μην είναι πάντα διαθέσιμο από το λειτουργικό σύστημα ή από κάποια άλλη εφαρμογή.

Το σύστημα δεν είναι ευέλικτο σε αλλαγές (πάντα θα είναι στην ίδια port η εφαρμογή)

Δεν μπορούμε να έχουμε παραπάνω από έναν server του ίδιου τύπου να τρέχουν (θα πρέπει να τρέχουν στο ίδιο port).

Επιπλέον υπάρχει πεπερασμένος αριθμός από διαθέσιμα endpoints και αν δοθούν στατικά σε μία εφαρμογή τότε δεν επιτυγχάνουμε scalability.

Άσκηση 9

Ένα παράδειγμα εφαρμογής που επιτρέπει στον client να κάνει bind ένα transient απομακρυσμένο αντικείμενο είναι το ftp, στο οποίο αν κάποιος ζητήσει κάποιο απομακρυσμένο αρχείο χωρίς ο ftp server να είναι σε λειτουργία τότε στέλνεται πίσω μήνυμα λάθους

Άσκηση 15

Σε ψευδοκώδικα ο αποστολέας και ο παραλήπτης θα εκτελούνται με τον ακόλουθο τρόπο:

Sender

```
MPI_irecv()
while(1)
    MPI_irecv()
    if(received ACK)
        break
```

Receiver

```
While(1)
    MPI_irecv()
    if(reveived message)
        MPI_isend(ACK)
        break
```

Άσκηση 22

Το πρόβλημα προκύπτει από την σύγχρονη επικοινωνία η οποία απαιτεί από το αποστολέα να μπλοκάρει μέχρι το μήνυμα να φτάσει στον παραλήπτη και από τον παραλήπτη να μπλοκάρει μέχρι να φτάσει το μήνυμα που έστειλε ο αποστολέας. Όταν λοιπόν αναφερόμαστε σε μεγάλο δίκτυο με μεγάλες γεωγραφικές αποστάσεις, ο χρόνος αποστολής κάθε μηνύματος μπορεί να γίνει πολύ μεγάλος με αποτέλεσμα να είναι μη αποδοτική η transient synchronous επικοινωνία., αφού θα αναγκάζονται οι διεργασίες να μπλοκάρουν για μεγάλα χρονικά διαστήματα. Επομένως αυτό εμποδίζει τη γεωγραφική κλιμάκωση.

Επίσης εμποδίζει την κλιμάκωση ως προς τους χρήστες και τους πόρους. Όταν υπάρχει ισχυρή synchronous επικοινωνία τότε κάποιος client θα μπλοκάρει έως ότου ο server επεξεργαστεί όλα τα μηνύματα των clients που περιμένουν. Έτσι όσο πιο πολλοί είναι οι clients τόσο μεγαλύτερες είναι αυτές οι καθυστερήσεις.

Λύσεις είναι:

Να μεταφέρουμε όσο το δυνατό περισσότερη δουλειά στους clients.

Να χρησιμοποιήσουμε ένα μη ισχυρό synchronous transient communication πρωτόκολλο.

Άσκηση 24

Ο μέγιστος χρόνος που χρειάζεται είναι αυτός που απαιτείται να κάνει το μήνυμα έναν πλήρη κύκλο. Αυτό μπορεί να το εγγυηθεί κανείς με βάση την οργάνωση του

δακτυλίου, ο οποίος εξασφαλίζει ότι ο μέγιστος αριθμός βημάτων που μπορεί να γίνουν για την αποστολή ενός μηνύματος από έναν κόμβο σε ένα άλλο είναι $n-1$, όπου n ο αριθμός των κόμβων του δακτυλίου.

Άσκηση 26

Κάθε στιγμή έρχονται $50.000.000/1000 = 50.000$ data units/sec

Στέλνονται από τον κουβά $10.000.000/1000 = 10.000$ data units/sec

Άρα συνολικά γεμίζει ο κουβάς με ρυθμό 40.000 data units /sec

Ο κουβάς χωράει $1.000.000/1.000 = 1.000$ data units

Άρα θα γεμίσει σε $1.000/40.000 = 1/40$ sec.