

Άλκης Γεωργόπουλος      Α.Μ.: 39  
Αναστάσιος Κοντογιώργης      Α.Μ.: 43

## Remote procedure calls vs. Java Remote Method Invocation

### Remote procedure calls

Το βασικότερο χαρακτηριστικό ενός RPC συστήματος είναι η ετερογένεια (heterogeneity). Αυτό σημαίνει ότι οι RPC μηχανισμοί (όπως CORBA και DCOM) λειτουργούν με την υπόθεση ότι το καταναμημένο σύστημα στο οποίο τρέχουν περιλαμβάνει διαφορετικές μηχανές οι οποίες μπορεί να διαφέρουν στο υλικό ή στο λογισμικό. Πρωταρχικός στόχος των RPC μηχανισμών είναι να εξαλείψουν την ετερογένεια.

Για την αντιμετώπιση αυτού του προβλήματος, τα RPC συστήματα χρησιμοποιούν proxies σε κάθε κόμβο του καταναμημένου συστήματος. Στα σημεία αυτά γίνεται επεξεργασία της μεταφερόμενης πληροφορίας, έτσι ώστε να μετατραπεί από τη μορφή που χρησιμοποιεί ο client σε αυτή του server και αντίστροφα. Το proxy στη μηχανή του client συνήθως αποκαλείται *stub*, ενώ αυτό στη μηχανή του server *skeleton*. Τα stubs μεταγλωττίζονται μαζί με τον κώδικα κλήσης και μετατρέπουν κάθε κλήση σε μία αναπαράσταση δεδομένων ανεξαρτήτου μηχανής (*marshaling*), η οποία στέλνεται στο skeleton του server. Στη συνέχεια το skeleton μεταφράζει την πληροφορία που έλαβε στη κατάλληλη μορφή για τη μηχανή που τρέχει (*unmarshaling*).

Το μεγαλύτερο μειονέκτημα των RPC συστημάτων είναι η στατική φύση της πληροφορίας που μπορεί να μεταφερθεί στο δίκτυο. Δηλαδή ο server που δέχεται τα δεδομένα, θα πρέπει να ξέρει ακριβώς τι του στέλνει ο client, ώστε να τα ερμηνεύσει κατάλληλα. Δεν επιτρέπεται δηλαδή πολυμορφισμός στα δεδομένα.

### Java Remote Method Invocation

Στα RMI συστήματα δεν υπάρχει το πρόβλημα της ετερογένειας, λόγω του ότι client και server ουσιαστικά είναι Java classes που τρέχουν σε μια εικονική Java μηχανή, συνθέτοντας έτσι ένα ενιαίο δίκτυο από εικονικές μηχανές Java. Λόγω αυτής της επαγόμενης ομοιογένειας τα RMI συστήματα δεν χρειάζονται IDL γλώσσα.

Υπάρχει δυνατότητα δυναμικής φόρτωσης, αλλά και μετακίνησης (migration) του κώδικα, κάθε στιγμή κατά τη διάρκεια της επικοινωνίας, σε αντίθεση με τα RPC συστήματα όπου ο προς εκτέλεση κώδικας θα πρέπει να είναι διαθέσιμος πριν από την έναρξη της επικοινωνίας.

Ο αντικειμενοστραφής σχεδιασμός του RMI προσφέρει πολυμορφισμό στη μεταφορά των δεδομένων. Στα RPC συστήματα για το κώδικα του stub υπεύθυνος είναι ο client που τον καλεί, ενώ στα RMI ο κώδικας για ένα αντικείμενο αρχικοποιείται μαζί με το αντικείμενο και μπορεί να είναι διαφορετικός ακόμα και για δύο αντικείμενα του ίδιου τύπου.

Η βασικότερη διαφορά μεταξύ RPC και RMI είναι ο μηχανισμός με τον οποίο μεταφέρονται τα δεδομένα στο δίκτυο. Στα RPC συστήματα το IDL interface παράγει ένα στατικό πρωτόκολλο το οποίο ορίζει τον τρόπο που θα επικοινωνήσουν το stub με το skeleton. Στα RMI συστήματα η αλληλεπίδραση μεταξύ client και server ορίζεται μέσω του Java interface για κάποιο αντικείμενο και φορτώνεται δυναμικά όποτε αυτό χρειαστεί.