

## Distributed Debugging

Τσώτσος Θοδωρής  
Φωλίνας Νίκος

## Δομή της παρουσίασης

- ♦ Εισαγωγικά
- ♦ Πώς γίνεται η συλλογή των καταστάσεων των διεργασιών
- ♦ Πώς η monitor αποφαίνεται ολικές συνεπείς καταστάσεις
- ♦ Πώς η monitor αποτιμά possibly φ, definitely φ σε ασύγχρονα και σύγχρονα συστήματα

## Δομή της παρουσίασης

- ♦ Εισαγωγικά
- ♦ Πώς γίνεται η συλλογή των καταστάσεων των διεργασιών
- ♦ Πώς η monitor αποφαίνεται ολικές συνεπείς καταστάσεις
- ♦ Πώς η monitor αποτιμά possibly φ, definitely φ σε ασύγχρονα και σύγχρονα συστήματα

## Εισαγωγικά

- ♦ Επιθυμούμε να μπορούμε να παρατηρούμε την εκτέλεση του προγράμματος κατά τη διάρκειά του.
- ♦ Έχουμε τη δυνατότητα να ελέγξουμε αν οι απαιτούμενες συνθήκες ασφάλειας καταστρατηγούνται.

## Παράδειγμα

- ♦ Κάθε διεργασία  $p_i$  έχει μία μεταβλητή  $x_i$  και η συνθήκη ασφάλειας είναι  $|x_i - x_j| \leq \delta$ , ακόμα και όταν κάποια διεργασία αλλάξει την τιμή της μεταβλητής της οποτεδήποτε.

## Marzullo και Neiger αλγόριθμος

- ♦ Είναι κεντρικοποιημένος
- ♦ Οι παρατηρούμενες διεργασίες στέλνουν την κατάσταση τους σε μία διεργασία που ονομάζεται monitor.
- ♦ Η διεργασία monitor παράγει ολικές συνεπείς καταστάσεις από τις καταστάσεις που λαμβάνει.
- ♦ Σκοπός μας είναι να καθορίσουμε περιπτώσεις όπου ένα δοθέν κατηγορημα ολικής κατάστασης φ είναι definitely True και περιπτώσεις όπου το κατηγορημα αυτό είναι possibly True.

## Possibly $\varphi$ – Definitely $\varphi$

- ♦ Possibly  $\varphi$ : υπάρχει μία συνεπής ολική κατάσταση  $S$  από την οποία περνά μία γραμμικότητα του  $H$ , έτσι ώστε το  $\varphi(S)=\text{TRUE}$ .
- ♦ Definitely  $\varphi$ : υπάρχει μία συνεπής ολική κατάσταση  $S$  από την οποία περνάνε όλες οι γραμμικότητες του  $H$ , έτσι ώστε το  $\varphi(S)=\text{TRUE}$ .

## Δομή της παρουσίασης

- ♦ Εισαγωγικά
- ♦ Πώς γίνεται η συλλογή των καταστάσεων των διεργασιών
- ♦ Πώς η monitor αποφαίνεται ολικές συνεπείς καταστάσεις
- ♦ Πώς η monitor αποτιμά possibly  $\varphi$ , definitely  $\varphi$  σε ασύγχρονα και σύγχρονα συστήματα

## Συλλογή της κατάστασης (1/2)

- ♦ Οι παρατηρούμενες διεργασίες  $p_i$  ( $i=1,2,\dots,N$ ) στέλνουν την αρχική τους κατάσταση στην monitor διεργασία αρχικά και στη συνέχεια κάθε φορά που απαιτείται με state messages.
- ♦ Η διεργασία monitor καταγράφει τα μηνύματα κατάστασης για κάθε διεργασία  $p_i$  σε αντίστοιχες ουρές  $Q_i$  ( $i=1,2,\dots,N$ )
- ♦ Η δραστηριότητα της δημιουργίας και αποστολής μηνυμάτων κατάστασης μπορεί να οδηγήσει σε καθυστέρηση της εκτέλεσης των διεργασιών.

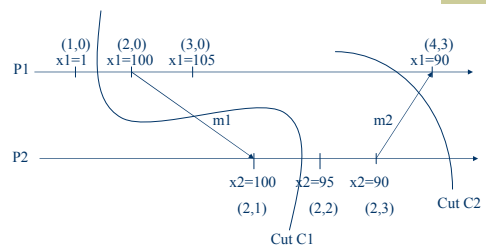
## Συλλογή της κατάστασης (2/2)

- ♦ Βελτιστοποιήσεις για μείωση της κυκλοφορίας των μηνυμάτων κατάστασης στο monitor.
- 1. Το κατηγορήμα της ολικής κατάστασης μπορεί να εξαρτάται μόνο από κάποια τμήματα των καταστάσεων της διεργασίας, π.χ. Μόνο από τις καταστάσεις κάποιων συγκεκριμένων μεταβλητών. Επομένως οι διεργασίες θα πρέπει να στείλουν μόνο τις σχετικές καταστάσεις στη διεργασία monitor.
- 2. Οι διεργασίες στέλνουν την κατάσταση τους σε χρονικές στιγμές όπου το κατηγορήμα  $\varphi$  μπορεί να γίνει TRUE ή να σταματήσει να είναι TRUE.

## Δομή της παρουσίασης

- ♦ Εισαγωγικά
- ♦ Πώς γίνεται η συλλογή των καταστάσεων των διεργασιών
- ♦ Πώς η monitor αποφαίνεται ολικές συνεπείς καταστάσεις
- ♦ Πώς η monitor αποτιμά possibly  $\varphi$ , definitely  $\varphi$  σε ασύγχρονα και σύγχρονα συστήματα

## Παρακολουθώντας τις συνεπείς ολικές καταστάσεις (1/3)



## Παρακολουθώντας τις συνεπείς ολικές καταστάσεις (2/3)

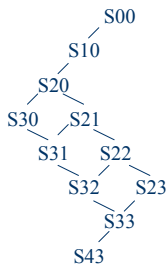
- Στο παραπάνω παράδειγμα υπάρχουν δύο διεργασίες, οι  $p_1$  και  $p_2$  με μεταβλητές  $x_1$  και  $x_2$  αντίστοιχα. Τα γεγονότα αφορούν προσαρμογές των τιμών των δύο μεταβλητών.
- Αρχικά  $x_1=x_2=0$  και ο περιορισμός είναι  $|x_1-x_2|\leq 50$ .
- Όταν κάποια διεργασία δέχεται ένα μήνυμα τροποποίησης από κάποια άλλη, θέτει την μεταβλητή της ίση με την τιμή που περιέχεται στο μήνυμα.
- Οποτεδήποτε μία από τις δύο διεργασίες  $p_1$  και  $p_2$  μεταβάλλουν την τιμή της μεταβλητής τους, στέλνουν την τιμή με ένα μήνυμα κατάσταση στη διεργασία monitor.

## Παρακολουθώντας τις συνεπείς ολικές καταστάσεις (3/3)

- Η monitor διεργασία μπορεί να διαχωρίσει τις συνεπείς ολικές καταστάσεις από τις αντίστοιχες ασυνεπείς.
- Χρησιμοποιούνται vector timestamps τα οποία στέλνονται στα μηνύματα κατάσταση.
- Μία κατάσταση  $S$  είναι μία συνεπής ολική κατάσταση αν και μόνον αν ισχύει:  
 $V(s_i)[i] \geq V(s_j)[i]$  για  $i, j=1, 2, \dots, N$  (συνθήκη CGS).
- Η συνθήκη αυτή σημαίνει ότι το πλήθος των γεγονότων της  $p_i$  που είναι γνωστά από την  $p_j$  όταν στάλθηκε το  $s_j$  δεν είναι περισσότερα από το πλήθος των γεγονότων που έχουν προκληθεί στην  $p_i$  όταν αυτή έστειλε το  $s_i$ .

## Το πλέγμα των ολικών καταστάσεων για την εκτέλεση του προηγούμενου σχήματος

Επίπεδο 0  
Επίπεδο 1  
Επίπεδο 2  
Επίπεδο 3  
Επίπεδο 4  
Επίπεδο 5  
Επίπεδο 6  
Επίπεδο 7



## Παράδειγμα

- Οι κόμβοι δηλώνουν τις ολικές καταστάσεις και οι ακμές δηλώνουν πιθανές μεταβάσεις μεταξύ καταστάσεων.
- $S_{00}$ : Και οι δύο διεργασίες στην αρχική τους κατάσταση
- $S_{10}$ : Η  $p_2$  παραμένει στην αρχική κατάσταση και η  $p_1$  είναι στην επόμενη κατάσταση.
- Το δίκτυο κατασκευάζεται με επίπεδα.

## Δομή της παρουσίασης

- Εισαγωγικά
- Πώς γίνεται η συλλογή των καταστάσεων των διεργασιών
- Πώς η monitor αποφαίνεται ολικές συνεπείς καταστάσεις
- Πώς η monitor αποτιμά possibly φ, definitely φ σε ασύγχρονα και σύγχρονα συστήματα

## Πώς η monitor αποτιμά possibly φ και definitely φ

- Possibly φ: Η διεργασία monitor αρχίζει από την αρχική κατάσταση και προχωράει μέσω όλων των συνεπών καταστάσεων που μπορεί να πάει από αυτό το σημείο, αποτιμώντας το φ σε κάθε στάδιο. Σταματάει όταν το φ αποτιμάται σε TRUE.
- Definitely φ: Η διεργασία monitor θα πρέπει να βρει ένα σετ από καταστάσεις μέσω των οποίων όλες οι γραμμικότητες θα πρέπει να περάσουν και σε κάθε μία από αυτές η αποτίμηση του φ να είναι TRUE.

## Αλγόριθμος για αποτίμηση possibly $\varphi$

```
L:=0;
States:={{s10, s20, ..., sn0}};
while( $\varphi(S)$ =False για όλες τις S ε States)
  L:=L+1;
  Reachable:={S':S' εφικτή στο H για κάθε
    S ε States και level(S')=L};
  States:=Reachable;
end while
έξοδος "possibly  $\varphi$ ";
```

## Αποτίμηση possibly $\varphi$

- Η S' είναι μία εφικτή από την S συνεπής κατάσταση στο επόμενο επίπεδο.
- Η S' διαφέρει από την S μόνο στο ότι περιέχει την επόμενη κατάσταση μιας διεργασίας.
- Η monitor βρίσκει όλες τις S' από τις ουρές των μηνυμάτων κατάστασης.
- Η S' είναι εφικτή από την S αν και μόνον αν:  
Για  $j=1,2,\dots,N$   $j \neq i$ :  $V(s_j)[j] \geq V(S'_i)[j]$ .

## Αλγόριθμος για αποτίμηση definitely $\varphi$

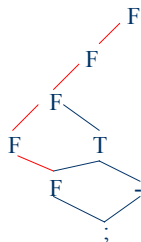
```
L:=0;
if( $\varphi(s_1^0, s_2^0, \dots, s_n^0)$ ) then States:={}
else States:={{s10, s20, ..., sn0}};
while(States  $\neq$  {})
  L:=L+1;
  Reachable:={S':S' εφικτή στο H για κάθε
    S ε States και level(S')=L};
  States:={S ε Reachable:  $\varphi(S)$ =False}
end while
έξοδος "definitely  $\varphi$ ";
```

## Αλγόριθμος για αποτίμηση definitely $\varphi$

- Η διεργασία monitor ξεκινά από μία αρχική κατάσταση  $(s_1^0, s_2^0, \dots, s_n^0)$
- Διατηρεί ένα σύνολο States που περιέχει τις καταστάσεις στο τρέχον επίπεδο που μπορούν να προσπελαστούν μέσω γραμμικότητας από την αρχική κατάσταση διασχίζοντας μόνο καταστάσεις για τις οποίες το  $\varphi$  αποτιμάται σε FALSE.
- Εώς ότου υπάρχει αυτή η γραμμικότητα δεν εισάγουμε definitely  $\varphi$ . Όταν φτάσουμε σε επίπεδο όπου δεν υπάρχει τέτοια γραμμικότητα, οδηγούμαστε σε definitely  $\varphi$ .

## Παράδειγμα αποτίμησης definitely $\varphi$

Επίπεδο 0  
Επίπεδο 1  
Επίπεδο 2  
Επίπεδο 3  
Επίπεδο 4  
Επίπεδο 5



## Παράδειγμα αποτίμησης definitely $\varphi$

- Στο επίπεδο 3 το σύνολο States αποτελείται από μία μόνο κατάσταση που είναι προσβάσιμη με γραμμικότητα όπου όλες οι καταστάσεις είναι False.
- Η μόνη κατάσταση που λαμβάνεται υπ' όψη στο επίπεδο 4 είναι αυτή που έχει το σύμβολο 'F'.
- Εάν το  $\varphi$  αποτιμάται σε TRUE στο επίπεδο 5, τότε θα έχουμε definitely  $\varphi$ . Ειδικότερα ο αλγόριθμος θα πρέπει να συνεχίσει πέρα από αυτό το επίπεδο.

## Κόστος των αλγορίθμων possibly $\phi$ , definitely $\phi$

- Με δεδομένο ότι  $N$  είναι οι διεργασίες και  $k$  το μέγιστο πλήθος από γεγονότα σε μία απλή διεργασία.
- Απαιτούνται  $O(k^N)$  συγκρίσεις.
- Απαιτείται χώρος  $O(kN)$
- Το κόστος αυτό μπορούμε να το μειώσουμε διαγράφοντας από τις ουρές «άχρηστα μηνύματα».
- Διαγράφουμε το μήνυμα που περιέχει το  $s_j$  όταν ισχύει:  
 $V(s_j^{last}[i]) > V(s_j[i])$  για  $j=1,2,\dots,N$  και  $j \neq i$ ,  
όπου  $s_j^{last}$  είναι η τελευταία κατάσταση που έλαβε η monitor από την διεργασία  $p_j$ .

## Αποτίμηση possibly $\phi$ , definitely $\phi$ σε σύγχρονα συστήματα (1/3)

- Οι διεργασίες έχουν τα φυσικά τους ρολόγια συγχρονισμένα.
- Οι παρατηρούμενες διεργασίες στέλνουν στα μηνύματα κατάσταση εκτός την κατάσταση τόσο τον vector χρόνο όσο και τον physical χρόνο.
- Όριο συγχρονισμού  $D$ :  
 $|C_i(t) - C_j(t)| < D$  για κάθε  $i, j=0,1,\dots,N$ .

## Αποτίμηση possibly $\phi$ , definitely $\phi$ σε σύγχρονα συστήματα (2/3)

- Έλεγχος για την συνέπεια της ολικής κατάστασης.
- Έλεγχος αν κάθε ζεύγος από καταστάσεις συνέβη τον ίδιο real time, δίνοντας τις τιμές των φυσικών ρολογιών.  
Για κάθε  $i, j=1,2,\dots,N$   
 $V(s_i)[i] \geq V(s_j)[i]$  και  $s_i, s_j$  έχουν συμβεί τον ίδιο real time.

## Αποτίμηση possibly $\phi$ , definitely $\phi$ σε σύγχρονα συστήματα (3/3)

- Το γεγονός ότι τα  $s_i, s_j$  έχουν συμβεί τον ίδιο real time ελέγχεται με την εξής συνθήκη:  
 $C_i(s_i) - D \leq C_j(s_j) \leq L_i(s_i) + D$ .
- Το  $L_i(s_j)$  είναι ίσο με
  - $C_i(s'_j)$ , αν η monitor, έχει λάβει ένα μήνυμα κατάστασης από την  $p_j$  για την  $s'_j$ .
  - $C_0 - \max + D$ , όπου  $C_0$  είναι η τρέχουσα τιμή του τοπικού ρολογιού του monitor και  $\max$  είναι ο μέγιστος χρόνος μεταφοράς ενός μηνύματος κατάστασης.