Consensus and related problems

ΟΜΑΔΑ: Ιωάννα Ζέλιου Α.Μ.: 55
Μελισσόβας Σπύρος Α.Μ.: 21
Παπαδόπουλος Φίλιππος Α.Μ.: 60

Τι θα δούμε…

- Consensus
- Byzantine generals
- Interactive consistency

Agreement Problems

- Imposibility in a asynchronous system

Το μοντέλο του συστήματος

- Συλλογή από διεργασίες p_i (i = 1, 2, …, N)
- Οι διεργασίες επικοινωνούν με μηνύματα.
- Υποθέτουμε ότι η επικοινωνία είναι αξιόπιστη.
- Η κάθε διεργασία μπορεί να είναι:
  - Λανθασμένη (Faulty): μία διεργασία στέλνει λάθος μήνυμα ή δεν απαντά καθόλου
  - Σωστή (Correct): οι υπόλοιπες
- Θεωρούμε ότι δε χρησιμοποιούνται ψηφιακές υπογραφές (αν και θα ήταν καλό…)

Ορισμός του consensus problem

- Αρχικά, κάθε διεργασία είναι στην κατάσταση *undecided*
- Έπειτα, κάθε διεργασία προτείνει μία τιμή $v_i \in D$ (i = 1, 2, …, N)
- Οι διεργασίες επικοινωνούν και ανταλλάσσουν μηνύματα
- Κάθε διεργασία επιλέγει μία τιμή για το *decision variable* d_i και μπαίνει στην κατάσταση *decided*.
- Από τη στιγμή που μία διεργασία μπει στην κατάσταση decided δε μπορεί πλέον να αλλάξει την τιμή του d_i

Ορισμός του consensus problem

Για έναν consensus αλγόριθμο πρέπει να ικανοποιούνται οι εξής συνθήκες:

- **Termination:** τελικά, κάθε διεργασία αποφασίζει για την decision variable της.
- **Agreement**: όλες οι σωστές διεργασίες πρέπει να έχουν την ίδια τιμή για τη decision variable.
- **Integrity:** Αν όλες οι σωστές διεργασίες έχουν προτείνει την ίδια τιμή, τότε κάθε σωστή διεργασία, που είναι στην κατάσταση decided,πρέπει να έχει επιλέξει αυτή την τιμή.

Παράδειγμα

Έστω μια ομάδα διεργασιών

- Κάθε διεργασία κάνει multicast την τιμή που προτείνει στα μέλη της ομάδας
- Κάθε διεργασία περιμένει έως ότου συλλέξει N τιμές
- Κάθε διεργασία εφαρμόζει τη συνάρτηση majority(v_1, v_2, …, v_N) στις τιμές που έλαβε.

Ορισμός του byzantine generals problem

- Τρεις ή περισσότεροι στρατηγοί θέλουν να αποφασίσουν για επίθεση ή υποχώρηση.
- Ένας από αυτούς είναι ο αρχηγός (commander) και δίνει το αρχικό «πρόσταγμα».
- Ο αρχηγός μπορεί να είναι προδότης ή όχι.
- Οι υπόλοιποι στρατηγοί μπορεί να προδότες ή όχι.

Ορισμός του byzantine generals problem

Για έναν byzantine generals αλγόριθμο πρέπει να ικανοποιούνται οι εξής συνθήκες:

- **Termination**
- **Agreement**
- **Integrity:** Αν ο αρχηγός είναι σωστός τότε όλες οι σωστές διεργασίες θα πρέπει να έχουν την τιμή που αποφάσισε ο αρχηγός.

➡ Η διαφορά με το consensus είναι ότι στο Consensus κάθε διεργασία προτείνει τη δική της ξεχωριστή τιμή.

Ορισμός του Interactive consistency problem

- Κάθε διεργασία προτείνει μια τιμή.
- Σκοπός είναι όλες οι σωστές διεργασίες να συμφωνήσουν για ένα πίνακα τιμών (decision vector)
- Ο πίνακας περιέχει μία τιμή για κάθε διεργασία.

Ορισμός του Interactive consistency problem

Για έναν byzantine generals αλγόριθμο πρέπει να ικανοποιούνται οι εξής συνθήκες:

- **Termination**
- **Agreement:** Το decision vector είναι ίδιο για όλες τις διεργασίες
- **Integrity:** Αν η διεργασία p_i είναι σωστή, τότε όλες οι σωστές διεργασίες αποφάσισαν την τιμή v_i για την i-οστή θέση του πίνακα

Σύνδεση του consensus με άλλα προβλήματα

- Μπορεί να δοθεί λύση σε κάποιο πρόβλημα μέσω της λύσης που έχει δοθεί σε κάποιο άλλο παρόμοιο πρόβλημα.
- Για παράδειγμα με απλές μεθόδους μπορούμε να εξάγουμε λύση:
  - για το πρόβλημα του Interactive Consistency (IC) από τη λύση του Byzantine Generals (BG).
  - για το πρόβλημα του Consensus (C) από τη λύση του Interactive Consistency (IC).
  - για το πρόβλημα του Byzantine Generals (BG) από τη λύση του Consensus (C).

Consensus σε ένα σύγχρονο σύστημα

- Ο αλγόριθμος υποθέτει μόνο crash failures και όχι arbitrary failures
- Λύση του προβλήματος με ένα μόνο multicast protocol.
- Από τις N διεργασίες το πολύ f διεργασίες μπορεί να παρουσιάσουν crash failures.
- f+1 γύροι
- Σε κάθε γύρο γίνεται multicast των τιμών ανάμεσα σε όλες τις διεργασίες.

Consensus σε ένα σύγχρονο σύστημα Αλγόριθμος

Algorithm for process $p_i \in g$; algorithm proceeds in $f + 1$ rounds

On initialization
  $Values_i^1 := \{v_i\}$; $Values_i^0 = \{\}$;
In round r $(1 \le r \le f + 1)$
  B-multicast$(g, Values_i^r - Values_i^{r-1})$; // Send only values that have not been sent
  $Values_i^{r+1} := Values_i^r$;
  while (in round r)
  {
    On B-deliver(V_j) *from some* p_j
      $Values_i^{r+1} := Values_i^{r+1} \cup V_j$;
  }
After $(f + 1)$ *rounds*
  Assign $d_i = minimum(Values_i^{f+1})$;

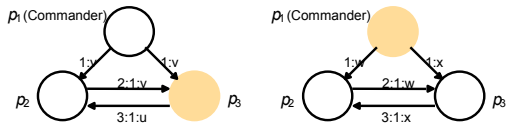
Consensus σε ένα σύγχρονο σύστημα

- Ο προηγούμενος αλγόριθμος ικανοποιεί το κριτήριο του termination λόγω της σύγχρονης επικοινωνίας.
- Ορθότητα: αρκεί να ελέγξουμε ότι στον τελευταίο γύρο κάθε διεργασία έχει το ίδιο σύνολο τιμών (αποδεικνύεται με εις άτοπον απαγωγή).
- Τα κριτήρια του agreement και του integrity είναι επακόλουθα της ορθότητας.

Το Byzantine generals problem σε ένα σύγχρονο σύστημα

- Οι διεργασίες μπορούν να παρουσιάσουν και crash failures και arbitrary failures.
- Από τις N διεργασίες το πολύ f διεργασίες μπορεί να παρουσιάσουν failures.
- Οι σωστές διεργασίες μπορούν να έχουν timeouts για να ελέγχουν αν τα μηνύματα φτάνουν έγκαιρα.
- Τα κανάλια επικοινωνίας είναι private

Impossibility για τρεις βυζαντινούς στρατηγούς

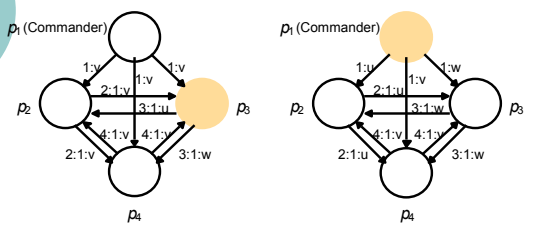


Faulty processes are shown shaded

Impossibility για N ≤ 3f

- Έστω N διεργασίες
- Το πολύ f από αυτές μπορεί να είναι λανθασμένες
- Αποδεικνύεται ότι δεν υπάρχει λύση αν N ≤ 3f (ομαδοποίηση στρατηγών, αναγωγή στο προηγούμενο)

Λύση για μία λανθασμένη διεργασία



Faulty processes are shown shaded

Μετρώντας την αποτελεσματικότητα…

- Πόσοι γύροι μηνυμάτων απαιτούνται;
- Πόσα μηνύματα και τι μεγέθους στέλνονται;

Μετρώντας την αποτελεσματικότητα…

- Στη γενική περίπτωση (f ≥ 1) ο αλγόριθμος του Lamport, για μηνύματα χωρίς ψηφιακές υπογραφές, απαιτεί
  - f+1 γύρους και
  - σε κάθε γύρο στέλνονται μηνύματα της τάξης O(N). Άρα συνολικά στέλνονται $O(N^{f+1})$ μηνύματα.
- Οι Fischer και Lynch απέδειξαν ότι οποιαδήποτε λύση στο πρόβλημα των Byzantine generals, απαιτεί τουλάχιστον f+1 γύρους μηνυμάτων.
- Άρα κανένας αλγόριθμος δε μπορεί να είναι πιο γρήγορος από του Lamport.

Impossibility σε ασύγχρονα συστήματα

- Ο Fischer απέδειξε ότι:

  κανένας αλγόριθμος δεν μπορεί να εγγυηθεί τη λύση του προβλήματος του consensus σε ένα ασύγχρονο σύστημα, ακόμη και αν μία μόνο διεργασία παρουσιάσει crash failure

- Κατά συνέπεια, σε ένα ασύγχρονο σύστημα δεν υπάρχει εγγυημένη λύση και στα εξής προβλήματα:
  - στο πρόβλημα των Byzantine generals
  - στο πρόβλημα του Interactive consistency
  - στο πρόβληματου totally ordered and reliable multicast.

Λύσεις για τα ασύγχρονα συστήματα

- Μία λύση είναι η χρήση *partially synchronous systems*
- Τρεις άλλες βασικές τεχνικές είναι:
  - Masking faults
  - Consensus using failure detectors
  - Consensus using randomization

Masking faults

- Κάλυψη οποιασδήποτε αποτυχίας
- Π.χ.
  Τα συστήματα δοσοληψιών χρησιμοποιούν δευτερεύοντα μέσα μόνιμης αποθήκευσης για να επιβιώνουν από τα crash failures

Consensus using failure detectors

- Οι διεργασίες θεωρούν ότι μία άλλη διεργασία έχει αποτύχει αν δεν απαντήσει μέσα σε κάποιο χρονικό διάστημα.
- Ακόμη κι αν η διεργασία αυτή απαντήσει κάποια στιγμή, η απάντηση δε λαμβάνεται υπόψη
- Ασύγχρονο σύστημα → σύγχρονο σύστημα

Consensus using failure detectors

- Γενικά, δεν επιλύουν πλήρως το πρόβλημα.
- *Imperfect detector:* Ειδική κατηγορία detectors που αγνοούν επιλεκτικά «ύποπτες» διεργασίες.
- Eventually weak failure detector: αποτελούν τον ασθενέστερο τύπο detectors και έχουν τα εξής χαρακτηριστικά:
  - *Eventually weakly complete:* κάποιες σωστές διεργασίες υποπτεύονται μόνιμα κάθε λανθασμένη διεργασία
  - *Eventually weakly accurate:* από κάποιο σημείο και μετά, μία τουλάχιστον σωστή διεργασία δε θεωρείται ύποπτη από καμία άλλη.

Consensus using randomization

Επιτρέπει στις διεργασίες να φτάσουν σε συμφωνία μέσα σε ένα πεπερασμένο, αναμενόμενο χρονικό διάστημα.

ΤΕΛΟΣ