Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ανατολικής Μακεδονίας και Θράκης

ΤΜΗΜΑ:

Μηχανικών Πληροφορικής

ΤΙΤΛΟΣ ΕΡΓΑΣΙΑΣ:

Δημιουργία ψευδό-τυχαίων αριθμών σε GPU

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ:

Παράλληλος και Κατανεμημένος Υπολογισμός

ΔΙΔΑΣΚΩΝ ΚΑΘΗΓΗΤΗΣ:

Παπακώστας Γεώργιος

ΟΝΟ/ΜΟ – ΑΕΜ :

Παπαρούνας Φώτης 3792

Τοροσιάν Δημήτριος 3824

Περιεχόμενα:

- Τι είναι η Γεννήτρια Ψευδοτυχαίων Αριθμών;
- Τι είναι η Προσομοίωση;
- Που χρησιμοποιούνται οι Ψευδοτυχαίοι Αριθμοί;
- Μεγάλη Απόδοση με μικρό χρηματικό κόστος
- Προβλήματα Παραλληλοποίησης Γεννήτριας Ψευδοτυχαίων Αριθμών
- Αξίζει να παραλληλοποιήσουμε ένα PRNG;
- Πάνω σε τι επικεντρώνεται το επιστημονικό άρθρο
- 2 Τρόποι Υλοποίησης PRNG σε GPU με χρήση βιβλιοθηκών
  - CURAND
  - Trust::Random
- Απαιτήσεις για την αποτελεσματική εκτέλεση του PRNG σε GP-GPU
- Ειδικές απαιτήσεις για την αποτελεσματική εκτέλεση του PRNG σε GP-GPU
- Οι κλασικές τεχνικές δημιουργία τυχαίων αριθμών ταιριάζουν καλά στη GP-GPU;
- Συμπεράσματα

Τι είναι η Γεννήτρια Ψευδοτυχαίων Αριθμών;
PRNG (Pseudorandom number generator)

➢ Η **γεννήτρια ψευδοτυχαίων αριθμών** χρησιμοποιεί υπολογιστικούς αλγόριθμους που μπορούν να παράγουν μεγάλες σειρές φαινομενικά τυχαίων αποτελεσμάτων, τα οποία είναι στην πραγματικότητα πλήρως προκαθορισμένα από μια μικρότερη αρχική τιμή, που είναι γνωστή ως σπορά ή κλειδί. [1]

➢Αυτοί οι αλγόριθμοι ονομάζονται γεννήτριες ψευδοτυχαίων αριθμών. Αυτοί οι τύποι των γεννητριών συνήθως δεν βασίζονται σε φυσικές πηγές, αν και μπορεί να χρησιμοποιήσουν κάποια φυσική πηγή για αρχικοποίηση της σποράς έτσι ώστε τα αποτελέσματα να είναι φαινομενικά απρόβλεπτα. [2]

Τι είναι η Προσομοίωση;

➢ Η μίμηση της λειτουργίας μιας εγκατάστασης ή μιας διαδικασίας, συνήθως με τη χρήση Η/Υ

➢Η προσομοίωση δεν είναι κάποιο είδος μοντέλου.

➢Τα μοντέλα γενικά αναπαριστούν την πραγματικότητα, ενώ η προσομοίωση την μιμείται.

Η προσομοίωση συνιστάται για:

- Τη μελέτη πολύπλοκων συστημάτων, δηλ. συστήματα για τα οποία η αναλυτικές λύσεις είναι εφικτές
- Για τη σύγκριση εναλλακτικών σχεδίων για ένα σύστημα που δεν υπάρχει ακόμα
- Για την μελέτη πιθανών μεταβολών σε ένα υπάρχων σύστημα
- Για την επαλήθευση αναλυτικών λύσεων [3]

Που χρησιμοποιούνται οι Ψευδοτυχαίοι Αριθμοί;

➢ Οι Ψευδοτυχαίοι Αριθμοί χρησιμοποιούνται για Κρυπτογραφία δημοσίου κλειδιού, Αποκρυπτογράφηση κ.α. αλλά κυρίως σε στοχευμένες προσομοιώσεις.

➢Διότι αποτελεί την βάση οποιασδήποτε στοχευόμενης προσομοίωσης.

➢ Αν αναφερόμαστε σε διαδοχικές προσομοιώσεις τότε μπορούμε να πούμε ότι λειτουργούν με αξιοπιστία. Αλλά όταν έχουμε να κάνουμε με παράλληλες προσομοιώσεις δεν είναι τόσο αξιόπιστη η χρήση τους.

➢ Γι’ αυτό παράλληλες πλατφόρμες εκτέλεσης (πχ. GPU) προσθέτουν περιορισμούς στην παραγωγή ψευδοτυχαίων αριθμών.

➢Οπότε αυτό έχει σαν αποτέλεσμα την πτώση της απόδοσης όταν ο παραλληλισμός δεν πραγματοποιείται.

Μεγάλη Απόδοση με μικρό χρηματικό κόστος

➢ Άλλο λόγος του χαμηλού κόστους και της μεγάλης υπολογιστικής ισχύς που προσφέρουν οι GPUs σε σχέση με τους υπερυπολογιστές, τα τελευταία χρόνια έχει αυξηθεί το ενδιαφέρων των developers ώστε να αναπτύξουν αλγορίθμους προσομοιώσεις αλλά και εργαλεία που να δουλεύουν απροβλημάτιστα και σε παράλληλα συστήματα.

Προβλήματα Παραλληλοποίησης Γεννήτριας Ψευδοτυχαίων Αριθμών

➢ Πρέπει πριν γίνει η επιλογή ποιος αλγόριθμος θα υλοποιηθεί σε παραλληλία να έχει τεσταριστή σε ένα μεμονωμένο επεξεργαστή οπότε να διαπιστωθεί πιο εύκολα η απόδοση του.

➢Πώς θα χωρίσουμε τυχαίες ροές μεταξύ των παραλλήλων στοιχείων επεξεργασίας.

➢Πώς θα εξασφαλίσουμε την ανεξαρτησία μεταξύ των παράλληλων ροών για να αποτρέψουμε την συμμετοχή της προσομοίωσης από την παραγωγή προκατειλημμένων αποτελεσμάτων

➢Αξιοποιώντας την ισχύ μιας συσκευής απαιτεί καλή γνώση στις GPUs. Αν και με τα πρόσφατα πλαίσια προγραμματισμού όπως το CUDA ή OpenCL, σχεδόν οποιοσδήποτε μπορεί να αναπτύξει εφαρμογές για GPUs.

Αξίζει να παραλληλοποιήσουμε ένα PRNG;

➢ Τα τρέχοντα PRNG που λειτουργούν με CPU εμφανίζουν αρκετά καλές επιδόσεις χάρη στις εξατομικευμένες βελτιστοποιήσεις του μεταγλωττιστή.

➢Παράδειγμα, η Mersenne Twister για επεξεργαστές γραφικών (MTGP), με πρόσφατη εφαρμογή της GPU της γνωστής Mersenne Twister, ανακοινώνεται ως 6 φορές ταχύτερη από την CPU αναφορά

➢Μερικές μελέτες δείχνουν ότι ο χρόνος που δαπανάται για τη δημιουργία ψευδοτυχαίων αριθμών καταναλώνεται το πολύ το 30% του χρόνου της CPU για μερικές "εντατικές στοχευμένες" πυρηνικές προσομοιώσεις, αλλά αυτές οι περιπτώσεις είναι πολύ σπάνιες.

➢ Λόγω του μικρού μέρους της εκτέλεσης στο χρόνο που χρησιμοποιούν οι περισσότερες από τις στοχαστικές προσομοιώσεις που παράγουν τυχαίους αριθμούς, δεν αξίζει η χρήση των GPU σε διεργασία παραγωγής.

Πάνω σε τι επικεντρώνεται το επιστημονικό άρθρο

➢ Το άρθρο δεν έχει σκοπό να παραθέτει παραδείγματα προσομοιώσεων που θα λειτουργούν αποτελεσματικά σε GPU.

➢Εστιάζει στις τεχνικές παραλληλισμού των ψευδοτυχαίων ροών που χρησιμοποιούνται για την άμεση τροφοδοσία προγραμμάτων παράλληλης προσομοίωσης που εκτελούνται σε GPU.

➢Η μελέτη αυτή:

- Προτείνει συγκεκριμένα κριτήρια GP-GPU για το σχεδιασμό PRNGs.
- Προτείνει απαιτήσεις για τεχνικές παραλληλισμού τυχαίων ροών στη GP-GPU.
- Μελετά, σύμφωνα με τις απαιτήσεις που έχουν εισαχθεί προηγουμένως, την καταλληλότητα των γνωστών PRNG και των τεχνικών παραλληλοποίησης για τις αρχιτεκτονικές GP-GPU.

2 Τρόποι Υλοποίησης PRNG σε GPU με χρήση βιβλιοθηκών

- CURAND

- Thrust::random

- Και τα δύο έχουν σκοπό να παρέχουν μια απλή διεπαφή για τη δημιουργία τυχαίων αριθμών στη GPU.

1ος Τρόπος Υλοποίησης PRNG σε GPU με χρήση του CURAND

➢ Το CURAND σχεδιάστηκε να παράγει τυχαίους αριθμούς με απλό τρόπο σε GPU με δυνατότητα CUDA.

➢Το κύριο πλεονέκτημα του CURAND είναι ότι είναι σε θέση να παράγει τόσο σχεδόν τυχαίες όσο και ψευδο-τυχαίες αλληλουχίες, είτε σε GPU είτε στην CPU.

➢Η διεπαφή προγραμματισμού (API) της βιβλιοθήκης παραμένει η ίδια, ανεξάρτητα από το είδος του PRNG και την πλατφόρμα στην οποία τρέχει η εφαρμογή.

2ος Τρόπος Υλοποίησης PRNG σε GPU με χρήση του Thrust :: random

➢Το Thrust :: random είναι μέρος μιας βιβλιοθήκης γενικής χρήσης με δυνατότητα GPU που ονομάζεται Thrust.

➢Αυτό το πρότζεκτ ανοιχτού κώδικα σκοπεύει να παρέχει ισοδύναμη βιβλιοθήκη με δυνατότητα GPU σε τυπικές βιβλιοθήκες C ++ γενικής χρήσης

➢Το Thrust::random υλοποιεί 3 PRNG, το καθένα ανήκει σε διαφορετική κατηγορία.

- Γραμμική Συμπαγή Γεννήτρια (LCG)
- Γραμμική Ανταλλαγή σχολίων (LFS)
- Αφαίρεση με Δανεισμό (SWB)

Απαιτήσεις για την αποτελεσματική εκτέλεση του PRNG σε GP-GPU

Συνοπτικά:

- Οι αριθμοί κινητής υποδιαστολής απλής ακρίβειας πρέπει να προτιμάται σε όλο τον αλγόριθμο παραγωγής τυχαίων αριθμών στη GP-GPU.

- Ο αλγόριθμος θα πρέπει να έχει σχεδιαστεί κατά τρόπο τέτοιο που να αποφεύγεται η πρόσβαση στη καθολική μνήμη.

Ειδικές απαιτήσεις για την αποτελεσματική εκτέλεση του PRNG σε GP-GPU

Συνοπτικά:

- Κάθε νήμα θα πρέπει να διαθέτει τη δική του τυχαία ακολουθία δηλ. κάθε νήμα να είναι σε θέση να παράξει την δικιά του τυχαία ροή αριθμών.

- Πρέπει να μπορεί να χρησιμοποιηθεί για οποιοδήποτε αριθμό νημάτων της GP-GPU ο αλγόριθμος PRNG.

- Οι παράλληλες τυχαίες ροές που παράγονται δεν πρέπει να είναι συνδεδεμένες

- Η έξοδος ενός PRNG δεν θα πρέπει να εξαρτάται από τον αριθμό των επεξεργαστών που χρησιμοποιούνται.

- Η ακολουθία των τυχαίων αριθμών που παράγονται για το συγκεκριμένο νήμα πρέπει να είναι το ίδιο, ανεξάρτητα από το αριθμό των νημάτων.

Οι κλασικές τεχνικές δημιουργία τυχαίων αριθμών ταιριάζουν καλά στη GP-GPU;

- **Leap Frog**

Αυτή η τεχνική δεν είναι αρκετά προσαρμοσμένη στη διάσπαση τυχαίων ροών στη GP-GPU δηλ. αν αλλάξει ο αριθμός των νημάτων, επηρεάζεται η υποσειρά σε κάθε νήμα και θα είναι διαφορετικό.

Λύση: θα ήταν η εφαρμογή του PRNG σε ένα επίπεδο με σταθερό αριθμό νημάτων.

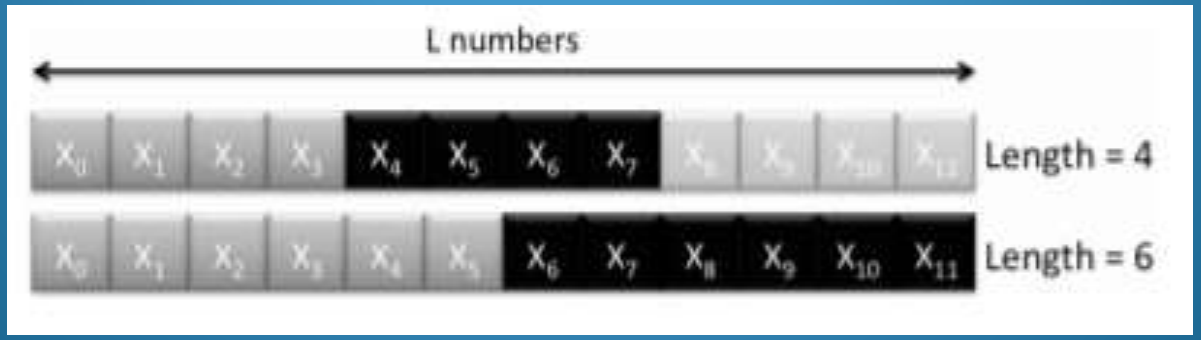
Διαφορετικοί αριθμοί νημάτων που οδηγούν σε διαφορετικές τυχαίες υπορροές

Οι κλασικές τεχνικές δημιουργία τυχαίων αριθμών ταιριάζουν καλά στη GP-GPU;

- **Ακολουθία Διαίρεσης**

Μια αποτελεσματική ακολουθία διαίρεσης βασίζεται σε ένα συγκεκριμένο χαρακτηριστικό του PRNG που ονομάζεται Jump Ahead ή Skip Ahead. Αν ένας εγγενές αλγόριθμος Jump Ahead είναι διαθέσιμος για το εμπλεκόμενο PRNG, η ακολουθία διαίρεσης είναι μια πολύ καλή προσέγγιση για GP-GPU. Σπόρος λέγεται η ρύθμιση μιας γεννήτριας σε μια προκαθορισμένη κατάσταση.

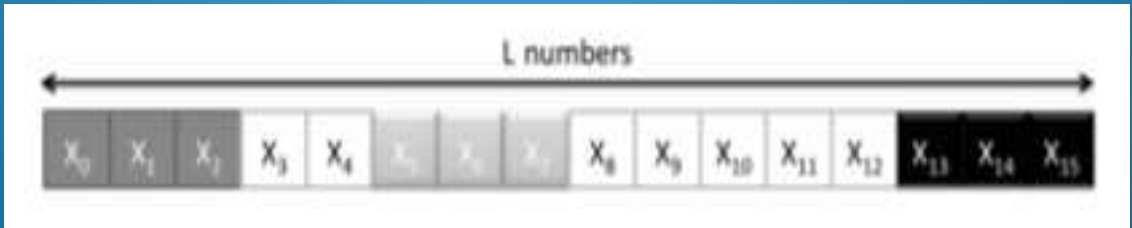
Λύση: Να υπολογίσουμε εκ των προτέρων υπορροές στην πλευρά του host (CPU), αποθηκεύουμε την κατάσταση των σπόρων στο κάθε σημείο εκκίνησης υπορροών και στη συνέχεια να μεταφέρουμε όλες αυτές τις καταστάσεις στη συσκευή (device).



Δύο παράλληλες ροές τυχαίων αριθμών που βασίζονται στην ακολουθία διαίρεση με δύο διαφορετικά μήκη δευτερευόντων ακολουθιών

Οι κλασικές τεχνικές δημιουργία τυχαίων αριθμών ταιριάζουν καλά στη GP-GPU;

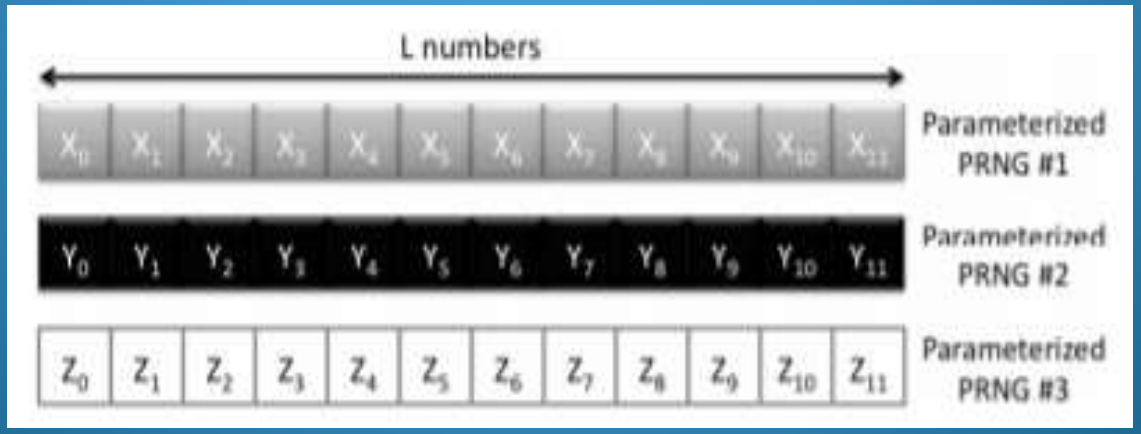
- **Random Spacing**
  - Προσέγγιση που βασίζεται στις πιθανότητες επιλέγοντας τυχαίες καταστάσεις στη βασική ροή και να τις ορίσουμε ως αρχικές καταστάσεις για την προκύπτοντα υπορροή.
  - Πρόβλημα: Υπάρχει μεγάλη πιθανότητα αλληλεπικάλυψης τυχαίων υπορροών.
  - Συμπέρασμα, αυτή η τεχνική ταιριάζει καλά με GP-GPUs, αλλά ο κίνδυνος αλληλεπικάλυψης μεταξύ υπο-ακολουθιών μεταβάλλεται ανάλογα με τον αριθμό και τη διάρκεια των υπο-ακολουθιών και στην περίοδο του χρησιμοποιούμενου PRNG. Αν επιλέξουμε γεννήτριες με μεγάλες περιόδους, όπως WELLs και Mersenne Twister, ο κίνδυνος αυτός είναι αμελητέος.



Τυχαία δημιουργία διαστήματος δημιουργίας τριών υπο-ακολουθιών ίσου μήκος

Οι κλασικές τεχνικές δημιουργία τυχαίων αριθμών ταιριάζουν καλά στη GP-GPU;

- Παραμετροποιημένη κατάσταση
- Οι τεχνικές που παρουσιάστηκαν μέχρι τώρα προσπάθησαν να χωρίσουν μία ροή σε αρκετές δευτερεύουσες υπορροές. Μια άλλη προσέγγιση αποτελείται από: κάθε μια γεννήτρια έχει την ίδια δομή και μηχανισμό παραγωγής με ένα μοναδικό σύνολο παραμέτρων.
- Αυτή η τεχνική εμφανίζει ορισμένους περιορισμούς που δυσκολεύουν τη μεταφορά στις μονάδες GPU. Όπως η αποθήκευση των καταστάσεων σπόρων είναι ήδη προβληματική, διότι χρειάζεται να ξοδεύουμε τεράστια ποσά μνήμης για να αποθηκεύσουμε μια παραμετροποιημένη κατάσταση ανά νήμα.



Τρία παραμετροποιημένα PRNG που παράγουν τρεις υψηλά ανεξάρτητες τυχαίες ακολουθίες

Συμπεράσματα

Έχουμε δει ότι όλο και περισσότερες εφαρμογές, και ιδιαίτερα για στοχευμένες προσομοιώσεις, τείνουν να εκμεταλλεύονται τις τελευταίες αρχιτεκτονικές της GP-GPU, προκειμένου να βελτιωθεί η εκτέλεση τους. Ωστόσο, οι υπολογισμοί GPU πρέπει να προσφέρει τα ίδια εργαλεία με άλλες πλατφόρμες. Καλής ποιότητας PRNGs ανήκουν σε αυτή την κατηγορία. Αυτή η παρουσίαση έδειξε πόσο δύσκολο θα μπορούσε να είναι η παραγωγή ψευδοτυχαίων αριθμών καλής ποιότητας σε GP-GPU.

Βιβλιογραφία

[1] https://el.wikipedia.org/wiki/Γεννήτρια_Τυχαίων_Αριθμών

[2] A.Toles “BBS Γεννήτριες Ψευδοτυχαίων Αριθμών” σελ.3-5, 2014
https://slideplayer.gr/slide/2004729/

[3] Έ.Σαμαράς ”Προσομοίωση” σελ.3-7, 2015
https://slideplayer.gr/slide/6239509/

[4] J. Passerat-Palmbach, C.Mazel, D. R.C. Hill “Pseudo-Random Number Generation on GP-GPU” σελ.1-7, 2011

Ευχαριστούμε για τον χρόνο σας!!!