



ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ

Προσομοιώσεις Monte Carlo σε GPU

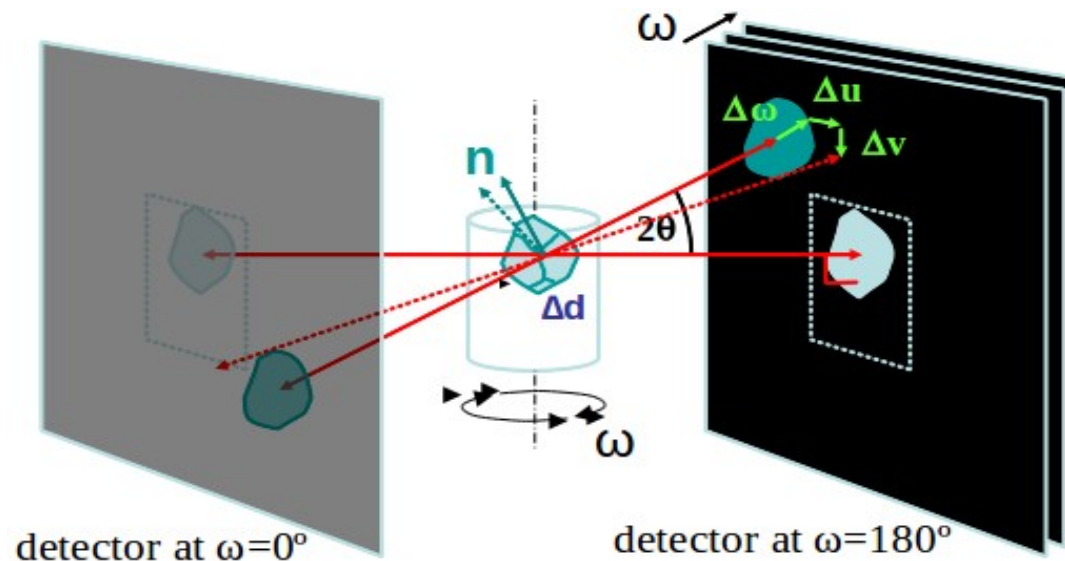
Δημήτρης Καρκούλης

Επιβλέπων: Κ. Αναγνωστόπουλος

15/07/2010

Discrete Contrast Tomography

• Χρήση ακτίνων Χ (απορρόφηση ή διάθλαση) όπου το αρχικό δείγμα (3D) προβάλλεται σε ένα επίπεδο 2 διαστάσεων (αισθητήρας CCD) υπό διακριτό εύρος γωνιών. Στόχος, από τα δεδομένα που “βλέπουμε” στον αισθητήρα, να ανακατασκευάσουμε το αρχικό δείγμα.



- Iterative Αλγεβρικοί Αλγόριθμοι ART
- Μέθοδοι: Back Projection, Filtered Back Projection, Forward Projection
- Εφαρμογές: Ιατρική, Παλαιοντολογία, Επιστήμη Υλικών κ.α.

Πρακτική στο European Synchrotron Radiation Facility

Σκοπός της πρακτικής: Ανάπτυξη ενός γρήγορου αλγορίθμου forward projection, βελτίωση ενός νέου πειραματικού αλγορίθμου, παράλληλη ανακατασκευή πολλών γειτονικών domains και εισαγωγή τοπολογικών περιορισμών.

- Τυπικός χρόνος ανακατασκευής ενός domain ~3 ώρες (200 iterations).
- Εύρεση του πιο αργού τμήματος του αλγορίθμου (profiling) και μεταφορά στην GPU. Μείωση του χρόνου ανακατασκευής σε ~20 λεπτά (200 iterations).
- Πλέον σε μία ημέρα μπορούν να δοκιμαστούν πολλές αλλαγές στον αλγόριθμο (Οι αλγόριθμοι ART σε αρκετό βαθμό βασίζονται σε εμπειρικό tweaking).
- Παράλληλη ανακατασκευή και τοπολογικός περιορισμός με χρήση MPI και 4 κάρτες γραφικών.
- Νέο bottleneck, μη βελτιστοποιημένη επικοινωνία.

Προσομειώσεις Monte Carlo σε GPU

- Σκοπός: Ανάπτυξη παράλληλων αλγορίθμων Monte Carlo για το 2D πρότυπο Ising.
Χρήση καρτών γραφικών (GPU) για την επιτάχυνση αυτών των αλγορίθμων.

Γιατί παράλληλοι αλγόριθμοι;

- Η ταχύτητα ενός πυρήνα αυξάνεται όλο και με μικρότερο ρυθμό. Αντίθετα, έχουμε ήδη περάσει από 2 σε 6 πυρήνες ανά επεξεργαστή.
- Πλέον σε κάθε κόμβο ενός υπερυπολογιστή βρίσκουμε 8 έως 12 πυρήνες ενώ σε έναν επιτραπέζιο υπολογιστή 2 έως 4.
- Επίσης, υπάρχουν βιβλιοθήκες που καθιστούν τον προγραμματισμό σε πολλαπλούς πυρήνες σχετικά απλή υπόθεση.

Γιατί GPU;

- Οι GPU έχουν εντελώς διαφορετική φιλοσοφία από τους επεξεργαστές.
- Είναι εξειδικευμένοι στην μαζική παράλληλη επεξεργασία δεδομένων και επομένως είναι ιδανικοί για συγκεκριμένα προβλήματα και ακατάλληλοι για αρκετά άλλα.
- Στηρίζονται σε πάρα πολλούς μικρούς επεξεργαστικούς πυρήνες (έως και λίγες χιλιάδες) για να προσφέρουν ταχύτητες ως και 5 TFLOPs ανά κάρτα στις σύγχρονες υλοποιήσεις.
- Όπως και οι επεξεργαστές, είναι ευρέως διαδεδομένες.

Πρότυπο Ising

- Κάθε μαγνητική ορμή περιγράφεται από ένα Ising spin σε έναν πλεγματοειδή χώρο.
- Κάθε Ising spin μπορεί να πάρει τιμή +1 ή -1.
- Αλληλεπιδρούν μεταξύ τους μέσω του όρου αλληλεπίδρασης J.
- Αλληλεπιδρούν με ένα εξωτερικό μαγνητικό πεδίο B.
- Η Hamiltonian του συστήματος είναι:

$$H = -J \sum_{(i,j)} s_i s_j - B \sum_i s_i$$

- Ο Όρος αλληλεπίδρασης J μπορεί να πάρει τιμές,
 - $J > 0$, σιδηρομαγνητικό πρότυπο.
 - $J < 0$, αντισιδηρομαγνητικό πρότυπο..
 - $J = 0$, δεν αλληλεπιδρούν μεταξύ τους.
- Στην περίπτωση μας, $J > 0$ με αλληλεπιδράσεις πλησιέστερων γειτόνων και $B = 0$.

Ιδιότητες και μεγέθη

- Παρουσιάζει μετάβαση φάσης 2^{ος} τάξης.
- Η μετάβαση φάσης συμβαίνει για θερμοκρασία $T = T_{\text{Curie}}$ η οποία έχει βρεθεί αναλυτικά: $T_c = \frac{2}{\log(1 + \sqrt{2})} \approx 2,27$
- Ως παράμετρος τάξης ορίζεται η μέση απόλυτη μαγνήτιση:

$$\langle |M| \rangle = \left\langle \sum_i |s_i| \right\rangle = \frac{1}{N} \sum_{i=1}^N |s_i|$$

- Για $T \leq T_c$ το σύστημα είναι σε αταξία με $m > 0$.
- Για $T > T_c$ το σύστημα είναι σε τάξη με $m = 0$.
- Ενώ η θερμοκρασία Curie εξαρτάται από τις διαστάσεις και τον τύπο του πλέγματος, στην κρίσιμη περιοχή τα μεγέθη συμπεριφέρονται ως:
 - $m \sim |t|^\beta$
 - $C \sim |t|^{-\alpha}$
 - $\chi \sim |t|^\gamma$
 - $\xi \sim |t|^{-\nu}$

ανεξάρτητα πλέγματος και αλληλεπιδράσεων, παγκοσμιότητα.

Ιδιότητες και μεγέθη

- Συνάρτηση επιμερισμού:

$$Z = \sum_{s_1 \pm 1} \sum_{s_2 \pm 1} \dots \sum_{s_N \pm 1} e^{-\beta H[\{s_i\}]} \equiv \sum_{\{s_i\}} e^{\beta J \sum_{i,j} s_i s_j + \beta B \sum_i s_i}$$

- Ενέργεια:

$$\langle E \rangle = \frac{1}{Z} \sum_{\mu} E_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial}{\partial \beta} \sum_{\mu} e^{-\beta E_{\mu}} = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = -\frac{\partial \log Z}{\partial \beta}$$

- Μαγνήτιση:

$$\langle M \rangle = \frac{1}{Z} \sum_{\mu} M_{\mu} e^{-\beta M_{\mu} + \beta B M_{\mu}} = \frac{1}{\beta Z} \frac{\partial Z}{\partial B} = -\frac{\partial F}{\partial B}$$

- Ειδική θερμότητα:

$$C = \frac{\partial U}{\partial T} = \frac{\partial \beta}{\partial T} \frac{\partial U}{\partial \beta} = k\beta^2 \frac{\partial^2 \ln Z}{\partial \beta^2} = k\beta^2 (\langle E^2 \rangle - \langle E \rangle^2)$$

- Μαγνητική επιδεκτικότητα:

$$\chi = \frac{1}{N} \frac{\partial \langle M \rangle}{\partial B} = \frac{\beta}{N} (\Delta M)^2$$

- Πιθανότητα κατάληψης καταστάσεως:

$$p_{\mu} = \frac{1}{Z} e^{-\beta E_{\mu}}$$

Διαδικασίες Monte Carlo

- Δειγματοληψία με κριτήριο σημαντικότητας.

$$Q_M = \frac{\sum_{i=1}^M Q_{\mu_i} (e^{-\beta E_{\mu_i}})^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^M (e^{-\beta E_{\mu_i}})^{-1} e^{-\beta E_{\mu_i}}} = \frac{1}{M} \sum_{i=1}^M Q_{\mu_i}$$

- Επιλογή συνόλου καταστάσεων μέσω αλυσίδας Markov.
- Εργοδοτικότητα.
- Συνθήκη λεπτομερούς ισοζύγησης.

$$\sum_{\nu} p_{\mu} P(\mu \rightarrow \nu) = \sum_{\nu} p_{\nu} P(\nu \rightarrow \mu)$$

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_{\nu}}{p_{\mu}} = e^{-\beta(E_{\nu} - E_{\mu})}$$

- Πιθανότητα επιλογής και λόγοι αποδοχής.

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)$$

Αλγόριθμος Metropolis (2D)

- Δυναμική single-spin-flip.
- Ισοπίθανη πιθανότητα επιλογής.
- Λόγοι αποδοχής:
$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & E_\nu - E_\mu > 0 \\ 1 & \text{αλλιώς} \end{cases}$$
- Παρουσιάζει το φαινόμενο της κρίσιμης επιβράδυνσης.
- Ο αλγόριθμος:
 1. Επιλογή παραμέτρων και αρχικών συνθηκών.
 2. Επιλογή κατάλληλης αρχικής διάταξης.
 3. Υπολογισμός των λόγων αποδοχής.
 4. Για κάθε Monte Carlo βήμα, επιλογή ενός spin βάσει της πιθανότητας επιλογής.
 5. Μέτρηση της μεταβολής της ενέργειας από το flip των γειτόνων. Αποδοχή ή απόρριψη της νέας κατάστασης βάσει των λόγων αποδοχής.
 6. Επανάληψη των 4 και 5 για N Monte Carlo βήματα.
 7. Μέτρηση ενέργειας και μαγνήτισης.
 8. Επανάληψη των 4,5,6 και 7 (βήμα Metropolis).

Αλγόριθμος Wolff

- Πιθανότητα επιλογής: $P_{add} = 1 - e^{-2\beta}$
- Λόγος αποδοχής μονάδα.
- Δεν παρουσιάζει το φαινόμενο της κρίσιμης επιβράδυνσης, αλλά είναι εξειδικευμένος.
- Ο αλγόριθμος:
 1. Επιλογή παραμέτρων και αρχικών συνθηκών.
 2. Επιλογή κατάλληλης αρχικής διάταξης.
 3. Υπολογισμός της πιθανότητας επιλογής.
 4. Επιλογή τυχαίου αρχικού γεννήτορα και προσθήκη του στο cluster.
 5. Για κάθε Monte Carlo βήμα, επιλογή ενός spin γεννήτορα από το cluster.
 6. Έλεγχος του προσανατολισμού ενός γείτονα. Αν είναι ίδιος προστίθεται στο cluster βάσει της πιθανότητας επιλογής.
 7. Επανάληψη των 5 και 6 έως ότου έχουν ελεχθεί όλα τα δυνητικά μέλη του cluster.
 8. Flip των spin του cluster.
 9. Μέτρηση ενέργειας και μαγνήτισης.
 10. Επανάληψη των 5,6,7,8 και 9.

GPU Computing

- Ως αμιγώς παράλληλη αρχιτεκτονική, ραγδαία αύξηση υπολογιστικής ισχύος με πολλαπλασιασμό πυρήνων σε κάθε νέα γενιά.
- Αφιερώνει περισσότερα κυκλώματα στην επεξεργασία παρά την cache και flow control.

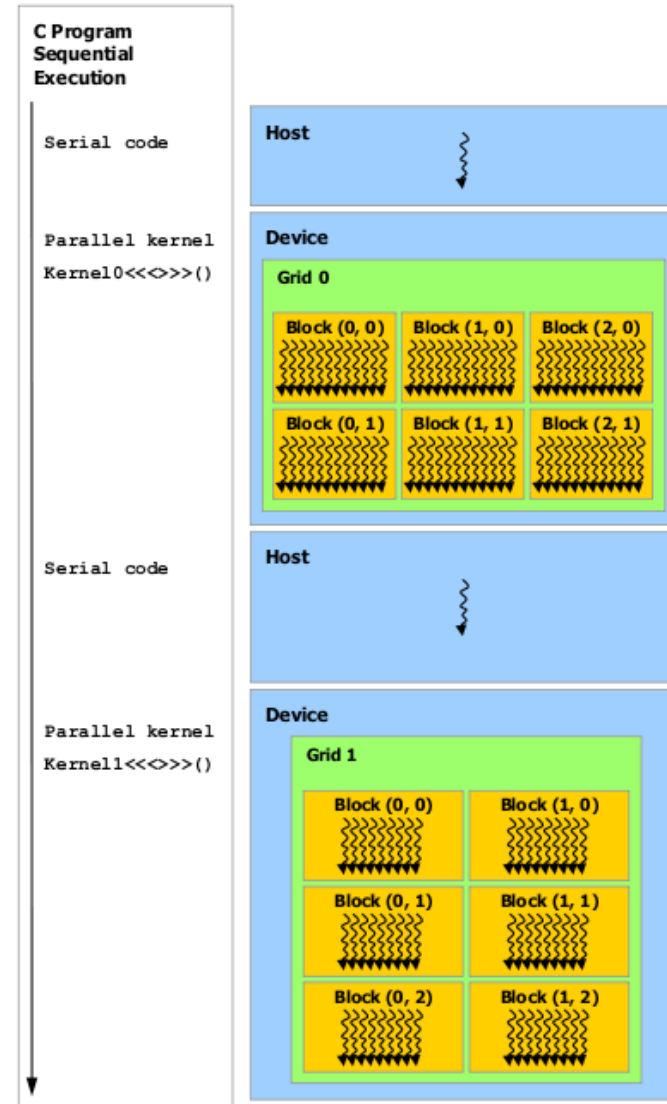
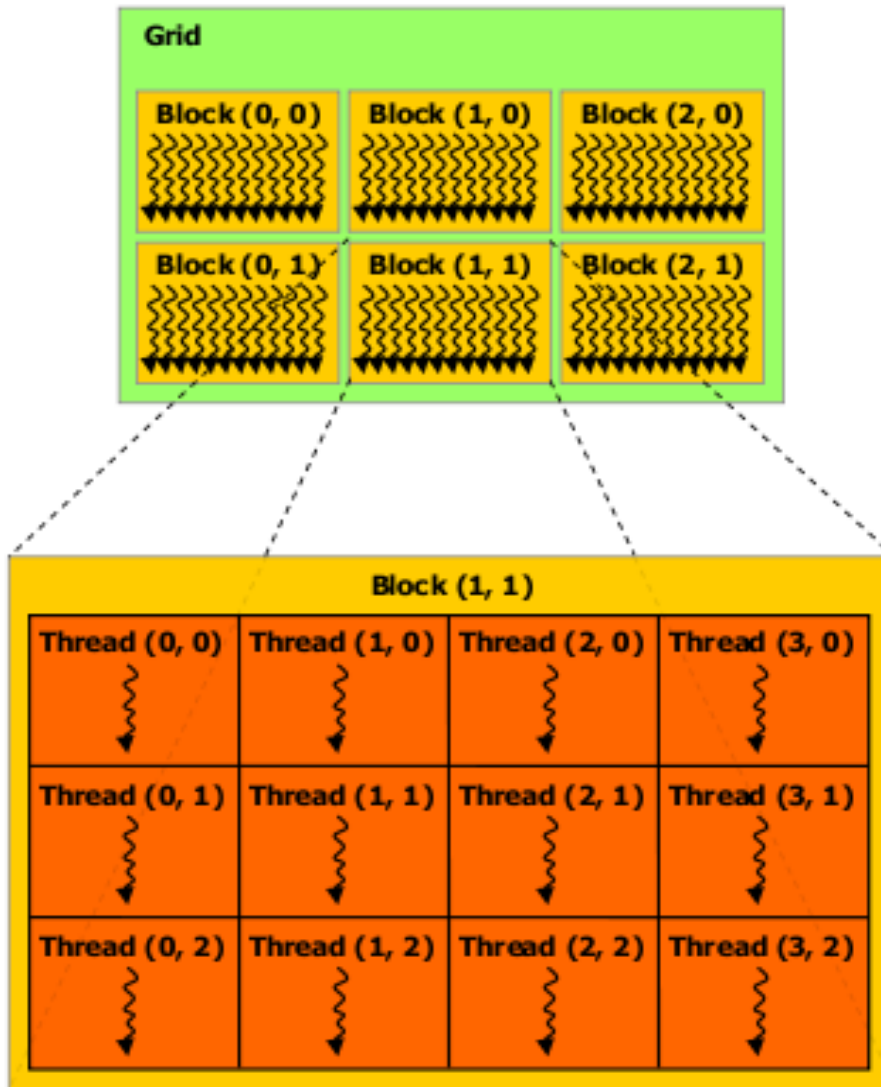


- Κλειστό σύστημα. Δεν απαιτείται συμβατότητα μνημών ► Bleeding edge τεχνολογία μνημών (GDDR5) με μία τάξη μεγέθους διαφορά στην ταχύτητα από τις αντίστοιχες ενός υπολογιστή (DDR3).
- Εφόσον γίνεται μία παρόμοια εργασία σε ένα σύνολο δεδομένων το οποίο μπορεί να χωριστεί σε μεγάλο πλήθος γενικά ανεξάρτητων υποσυνόλων τότε θα μπορεί να επιταχυνθεί δραματικά από μία GPU.
- Επιταχύνσεις έως και τρεις τάξεις μεγέθους. Από σειριακό κώδικα σε 4 πυρήνες θα έχουμε ~4x. Από σειριακό κώδικα σε GPU ένα σύνηθες speedup είναι ~60-70x. Σε ιδανικές περιπτώσεις μπορεί να ξεπεράσει το 200x.
- Μπορεί να χρησιμοποιηθεί μέσω MPI και OpenMP χωρίς καμία μετατροπή.

Αρχιτεκτονική και δομή της GPU

- Array από “multiprocessors” (MP).
- Μνήμη DRAM.
- Δίαυλος επικοινωνίας (Pci Express).
- Κάθε MP αποτελείται από:
 - 8 επεξεργαστικούς πυρήνες (Streaming processors – Shaders).
 - Shared memory χωρισμένη σε banks. Υποστηρίζει παράλληλη ανάγνωση και εγγραφή ► Ταχύτητα της τάξης Tbyte/s.
 - Registers.
- Out of order Execution.
- Ετερογενής και ασύγχρονη εκτέλεση CPU - GPU.
- Τα δεδομένα οργανώνονται σε ένα grid. Στην συνέχεια χωρίζονται σε blocks τα οποία αποτελούνται από threads.
- Δεν ορίζεται “ελεύθερο” thread όπως στην CPU. Θα πρέπει να ανήκει σε ένα block.
- Ο συγχρονισμός threads διαφορετικών blocks είναι αδύνατος.

Αρχιτεκτονική και δομή της GPU

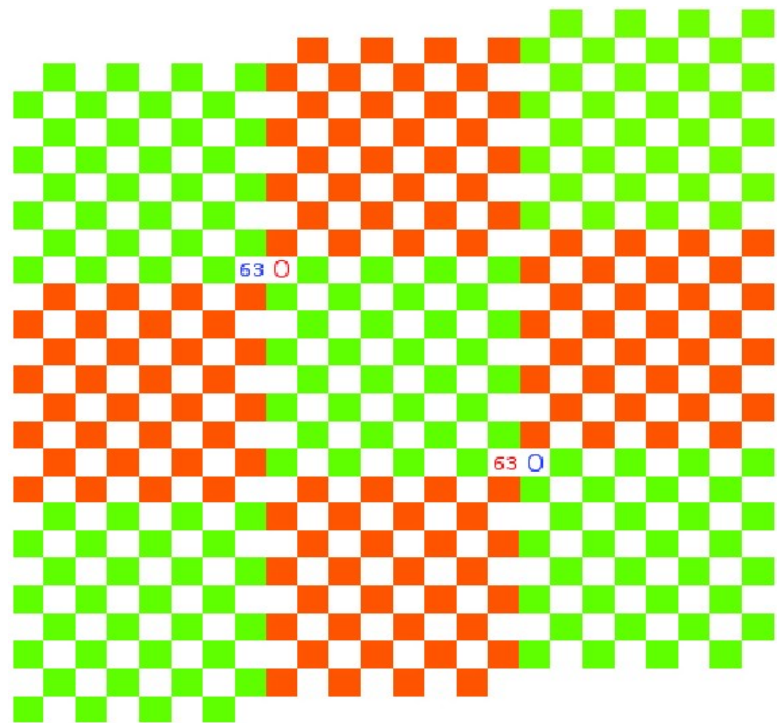
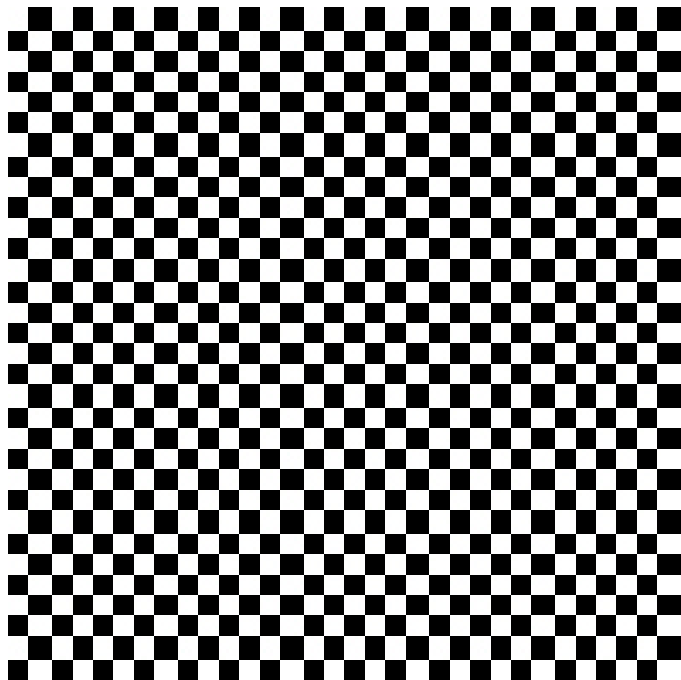


GPU Computing - Πότε;

- Τα δεδομένα μπορούν να χωριστούν (domain decomposition) σε μεγάλο πλήθος γενικά ανεξαρτήτων δεδομένων.
- Θα πρέπει σε κάθε multiprocessor της GPU να μπορούμε να έχουμε πολλά ενεργά threads.
- Οι αντιγραφές μνήμης από CPU σε GPU και αντίστροφα μπορούν να είναι περιορισμένες.
- Περιορισμένες αλλαγές ροής στον αλγόριθμο.
- Δεν απαιτούνται συχνοί συγχρονισμοί των threads ενός block.

GPU Metropolis

- Το τετραγωνικό πλέγμα με κοντινές αλληλεπιδράσεις (τα δεδομένα μας), μπορεί να χωριστεί σε υποσύνολα γενικά ανεξάρτητα μεταξύ τους – πλέγμα checkerboard.
- Σε κάθε πλεγματική θέση εκτελείται η ίδια περίπτωση διεργασία (αποδοχή ή απόρριψη μίας κατάστασης βάσει των λόγων αποδοχής).
- Στην περίπτωση των ελικοειδών συνοριακών υπάρχει εξάρτηση στα σύνορα, αλλά δεν αποτελεί πρόβλημα.



GPU Metropolis

- Ο αλγόριθμος χωρίζεται σε “λευκό” και “μαύρο” και εκτελούνται εναλλάξ.
- 1 thread αντιστοιχεί σε ένα 2x2 υπόπλεγμα για την αποφυγή αδρανών threads.
- Κάθε thread έχει το δικό του seed για την γεννήτρια ψευδοτυχαίων.
- Στον μαύρο αλγόριθμο σε ένα υποπλέγμα θα ανανεωθούν τα spin (0,0) και (1,1) ενώ στον λευκό τα (0,1) και (1,0).
- Για τα threads στο σύνορο χρησιμοποιούνται ειδικές συναρτήσεις φραγής της μνήμης για αποφυγή απροσδιόριστων αποτελεσμάτων.
- Ο υπολογισμός της ενέργειας και της μαγνήτισης (reduction) γίνεται στην GPU εκτός από ειδικές περιπτώσεις.

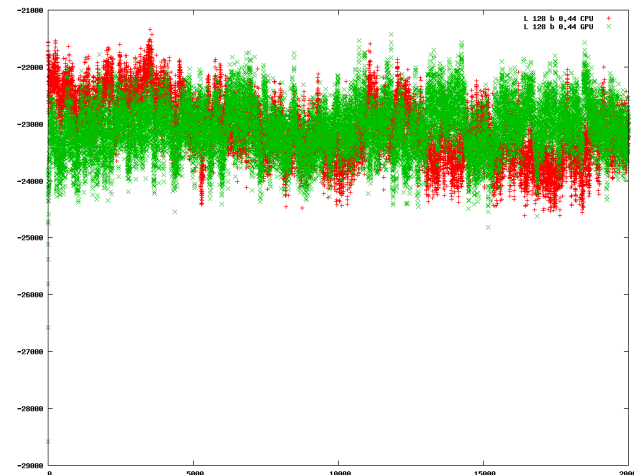
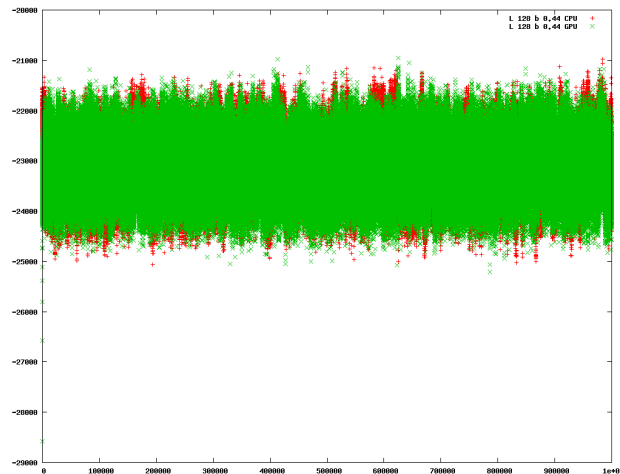
GPU Metropolis

Ο αλγόριθμος:

1. Επιλογή παραμέτρων και αρχικών συνθηκών.
2. Επιλογή κατάλληλης αρχικής διάταξης.
3. Υπολογισμός των λόγων αποδοχής.
4. Αρχικοποίηση της GPU, δέσμευση μνήμης DRAM.
5. Ορισμός του μεγέθους του grid και των blocks.
6. Αντιγραφή του πλέγματος, των διαστάσεων του και των λόγων αποδοχής στην DRAM.
7. Εκτέλεση του “μαύρου” kernel και στην συνέχεια του “λευκού”.
8. Εκτέλεση αναδρομικού reduction αλγορίθμου για την μέτρηση της ενέργειας και της μαγνήτισης.
9. Αντιγραφή της τιμής της ενέργειας και της μαγνήτισης στην RAM.
10. Επανάληψη των βημάτων 7,8 και 9.
11. Αντιγραφή του πλέγματος και των seeds πίσω στην RAM για άλλες χρήσεις.

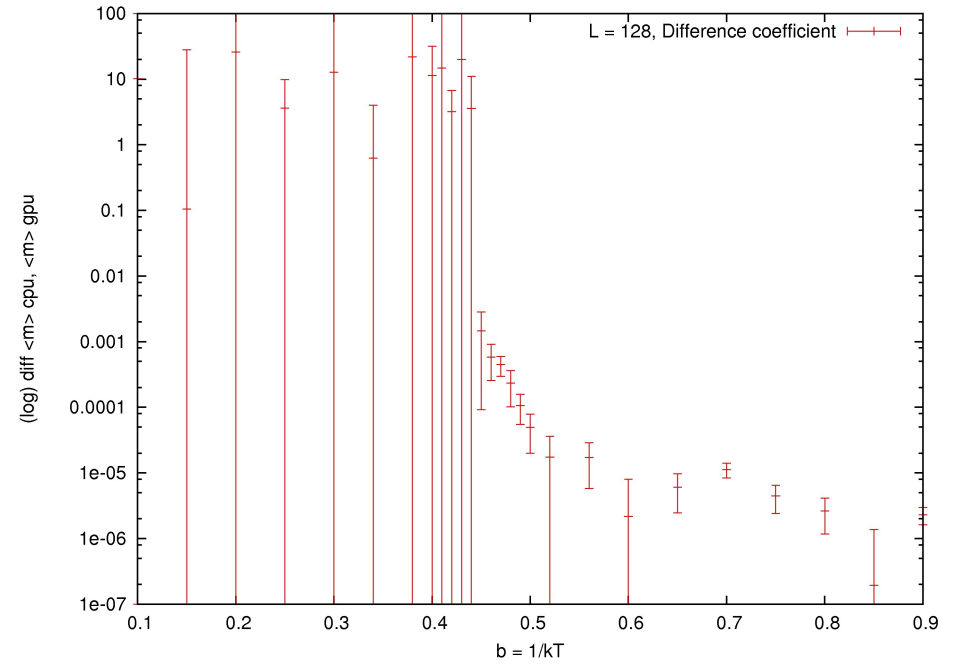
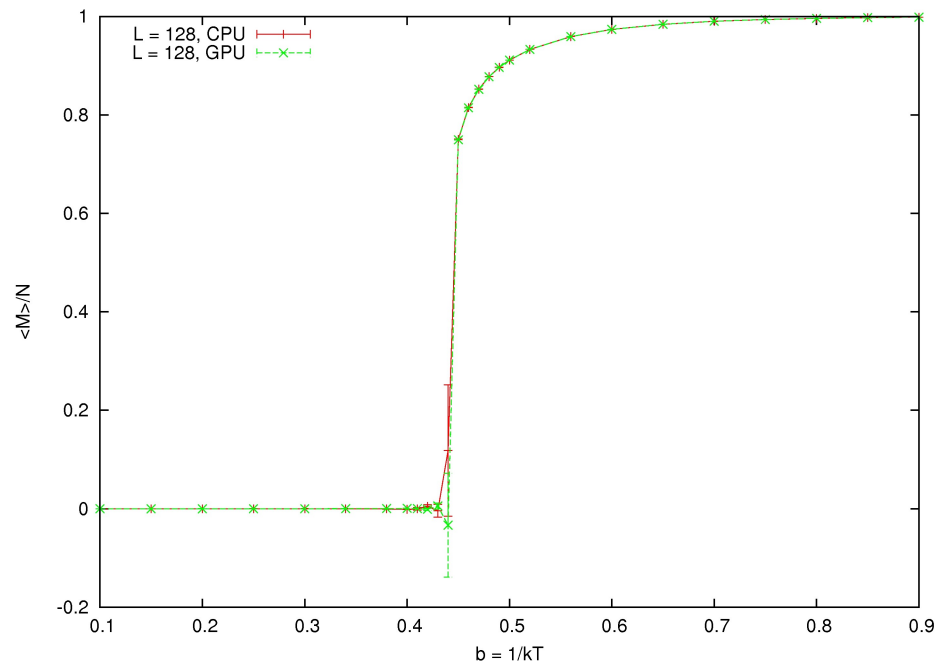
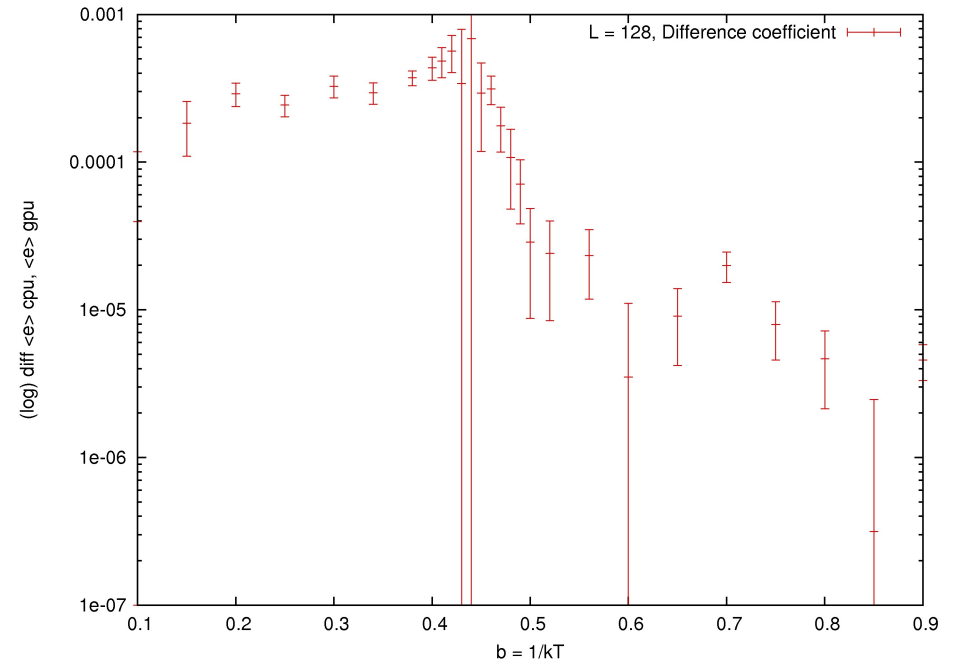
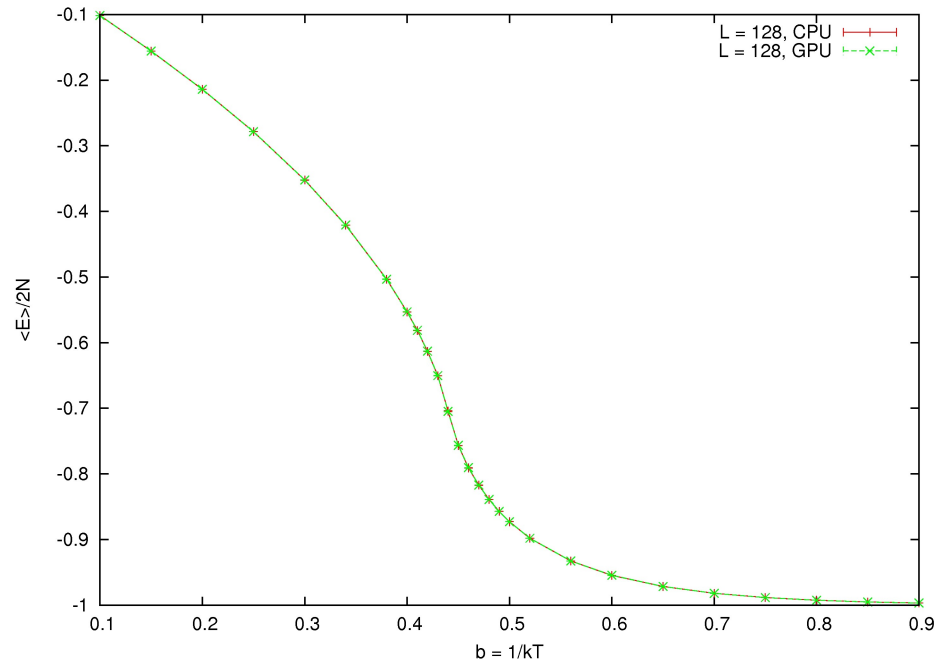
GPU Metropolis - Προσομοίωση

- Εκτέλεση σειριακού CPU αλγορίθμου και GPU αλγορίθμου σε υπολογιστή GPU Tesla.
- Μήκος πλέγματος L 8 έως 128, με βήμα πολλαπλάσιο του 8.
- Θερμοκρασίες β 0.100 έως 0.900.
- 10^6 βήματα Metropolis (sweeps).
- Σε κάθε L έλεγχος τότε το σύστημα είναι σε θερμική ισορροπία μέσω των time histories.

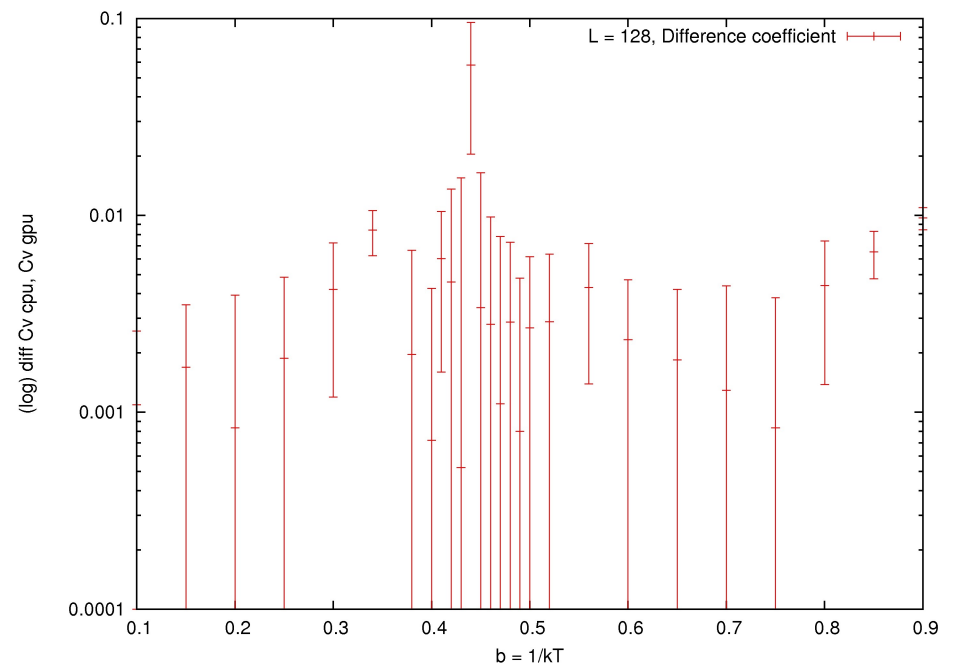
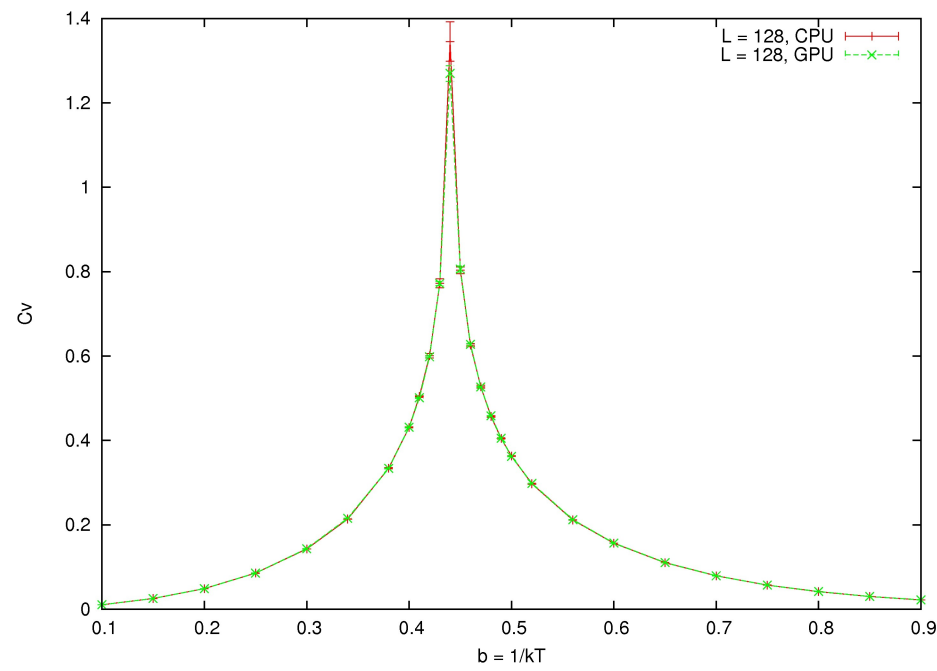
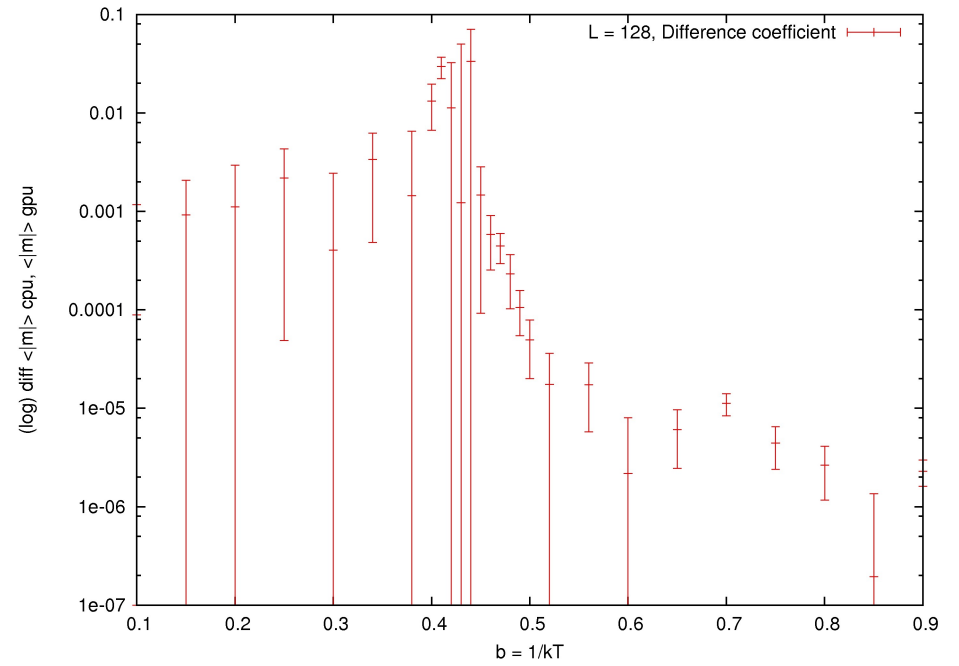
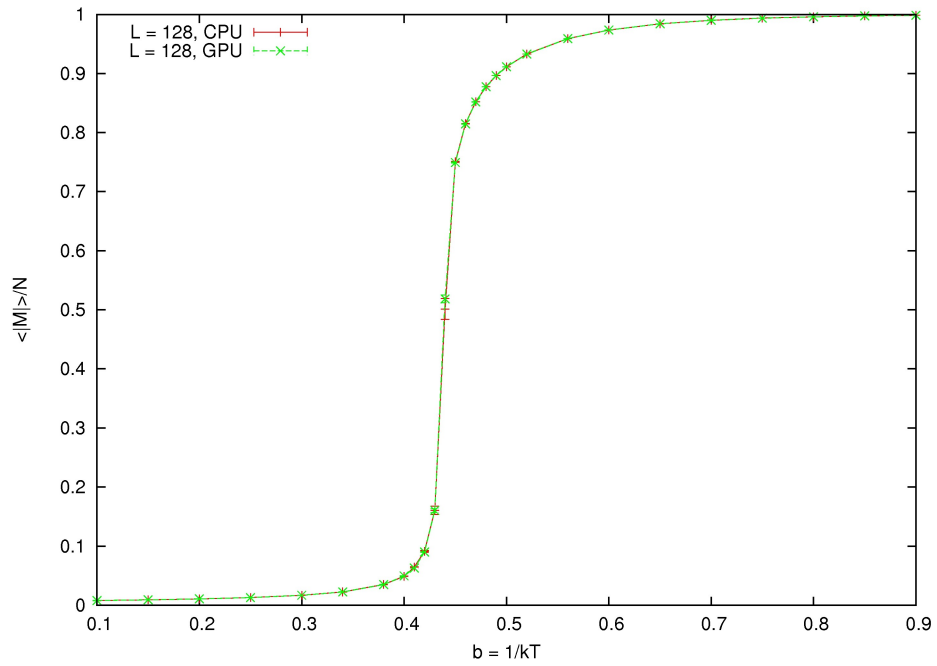


- Υπολογισμός ποσοτήτων και των σφαλμάτων τους.
- Σύγκριση των αποτελεσμάτων.

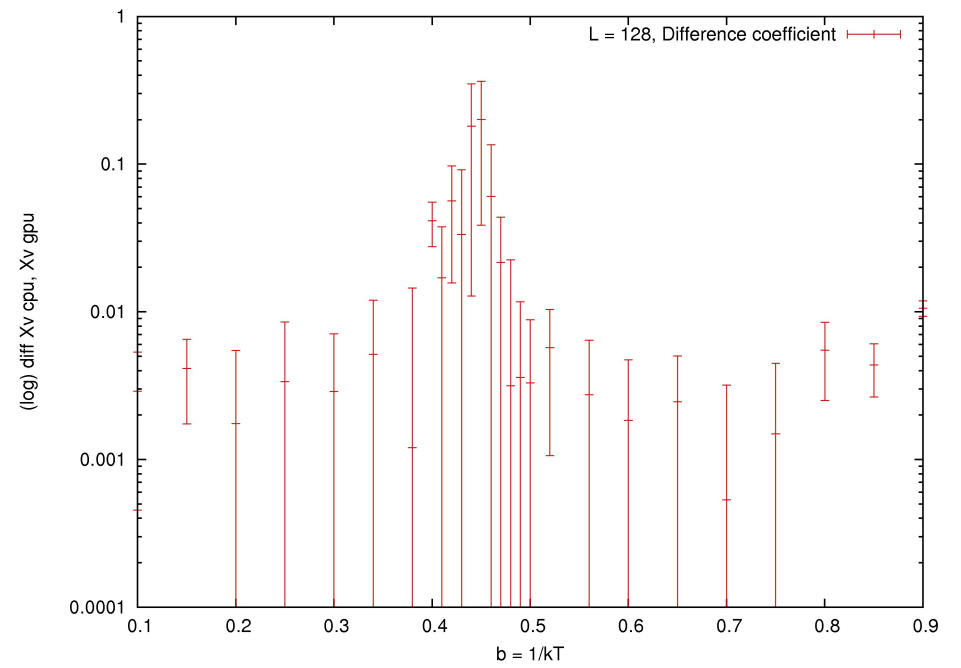
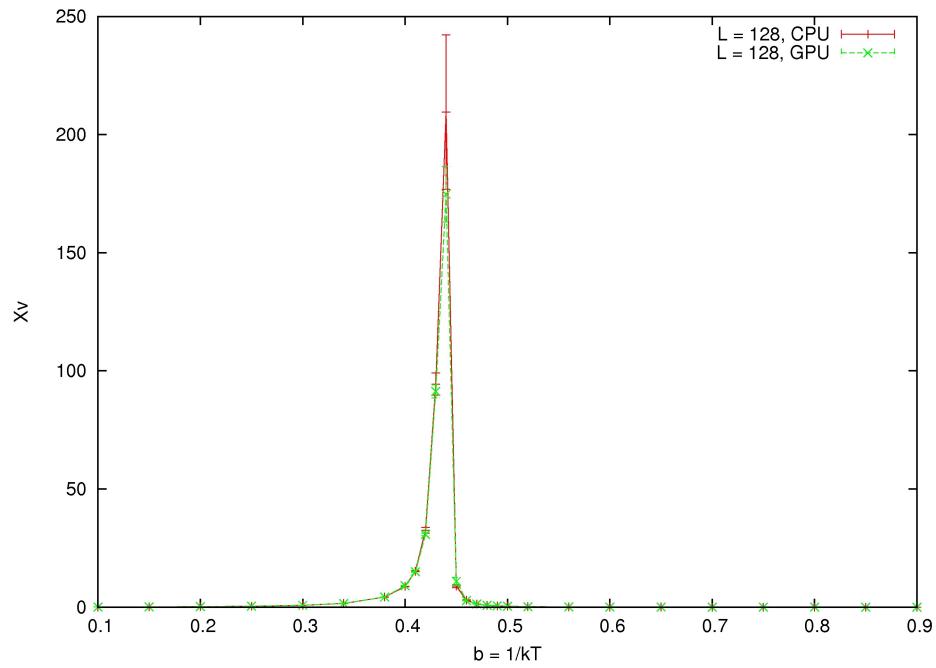
GPU Metropolis – Σύγκριση



GPU Metropolis – Σύγκριση

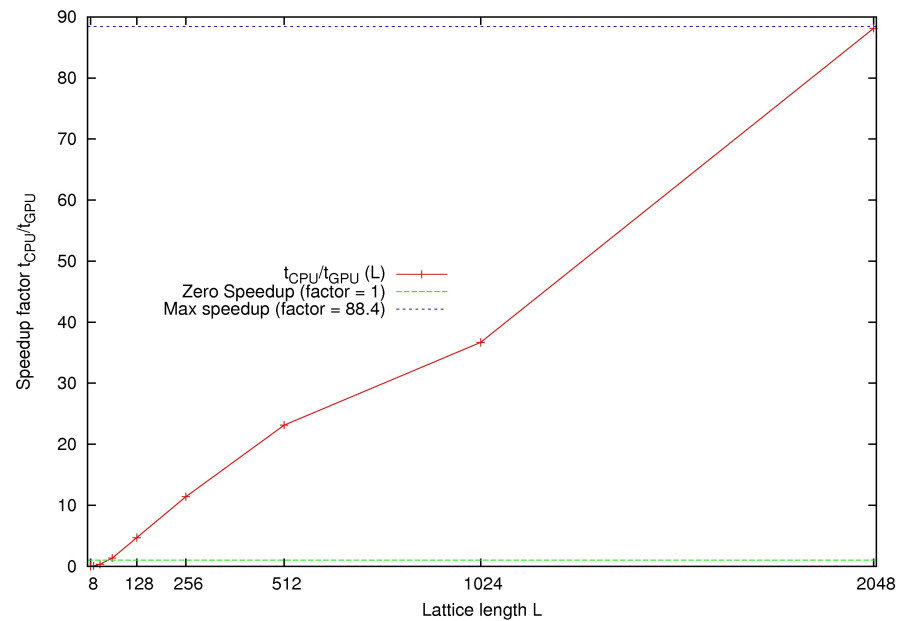
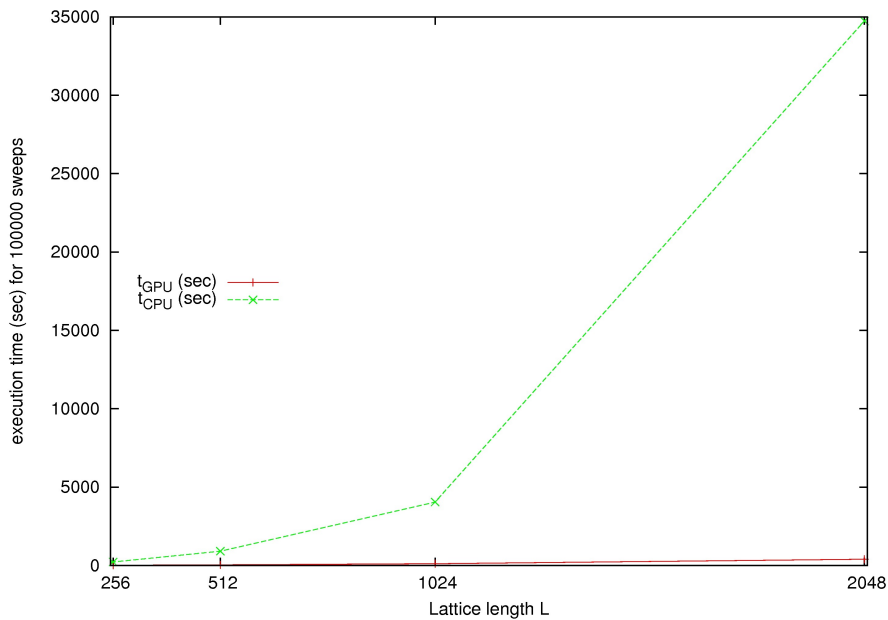
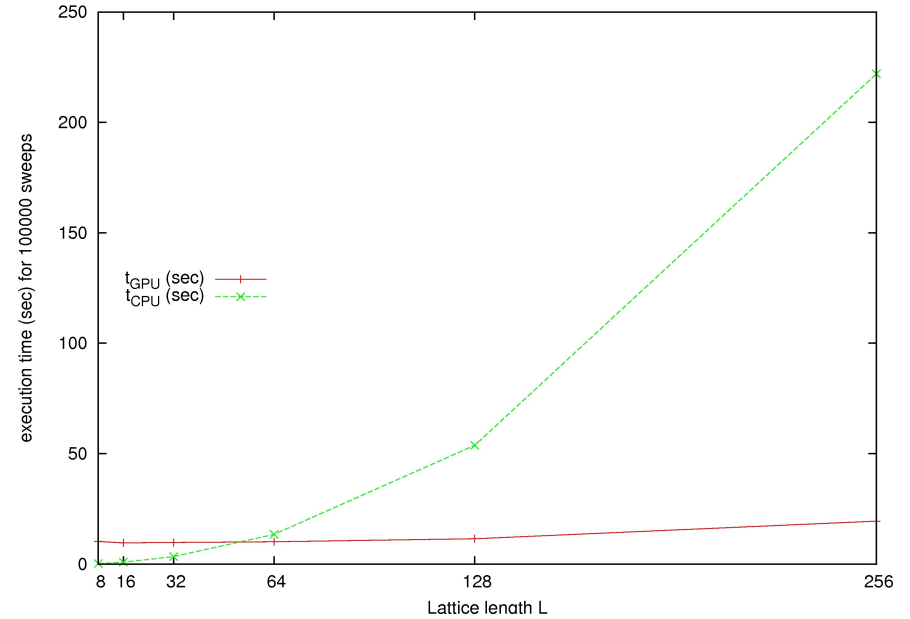
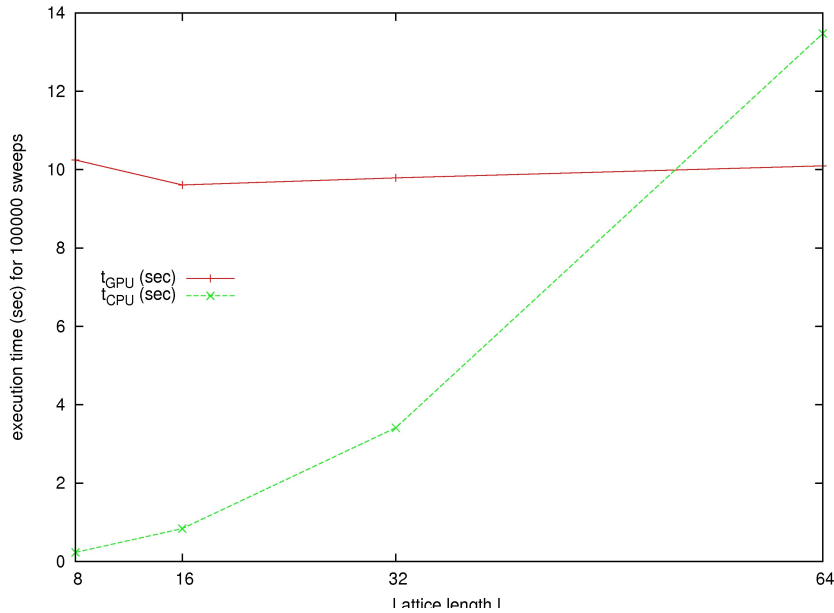


GPU Metropolis – Σύγκριση



GPU Metropolis - Benchmark

- L 8 έως 2048 με β 0.44 και 10^5 sweeps.

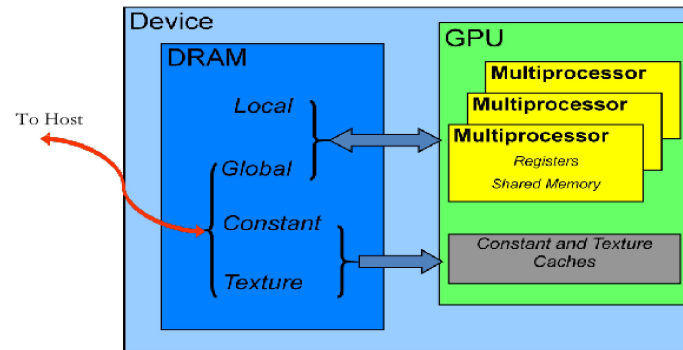


GPU Computing - Ειδικά

- Απαιτεί γνώση και κατανόηση της αρχιτεκτονικής και του τρόπου εκτέλεσης.
- Τα threads δομούνται σε blocks και grids. Κατά την εκτέλεση όμως οργανώνονται διαφορετικά.
- Αρχικά φορτώνονται κάποια blocks σε έναν multiprocessor και τα threads του καθενός οργανώνονται σε warps των 32 threads.
- Σε κάθε ενεργό warp εκτελείται το πάνω ή το κάτω μισό κάθε φορά ταυτόχρονα (16 threads).
- Η ροή μπορεί να αποκλίνει ανάμεσα σε διαφορετικά warps χωρίς πρόβλημα, αλλά μέσα σε ένα warp η ροή δεν πρέπει να αποκλίνει αλλιώς αντί για ταυτόχρονη εκτέλεση θα έχουμε σειριακή για τα threads με διαφορετική ροή.
- Τα warps εκτελούνται γενικά με μη προσδιορίσιμη σειρά. Οι προσβάσεις στην μνήμη πρέπει να είναι όσο το δυνατόν πιο ανεξάρτητες.
- Εκτελούνται με αυτό τον τρόπο ώστε ένας multiprocessor να μπορεί να κρύψει τις καθυστερήσεις που οφείλονται σε διεργασίες μνήμης ή αριθμητικούς υπολογισμούς. Εναλλάσσει συνέχεια warps που περιμένουν να ολοκληρωθεί μία διαδικασία με warps τα οποία μπορούν να συνεχίσουν να εκτελούνται.

GPU Computing - Ειδικά

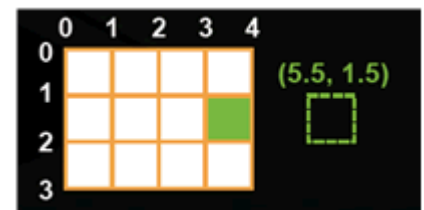
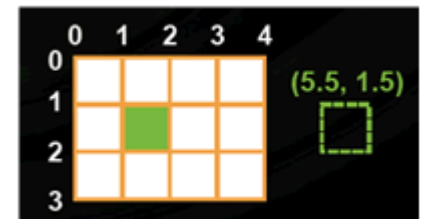
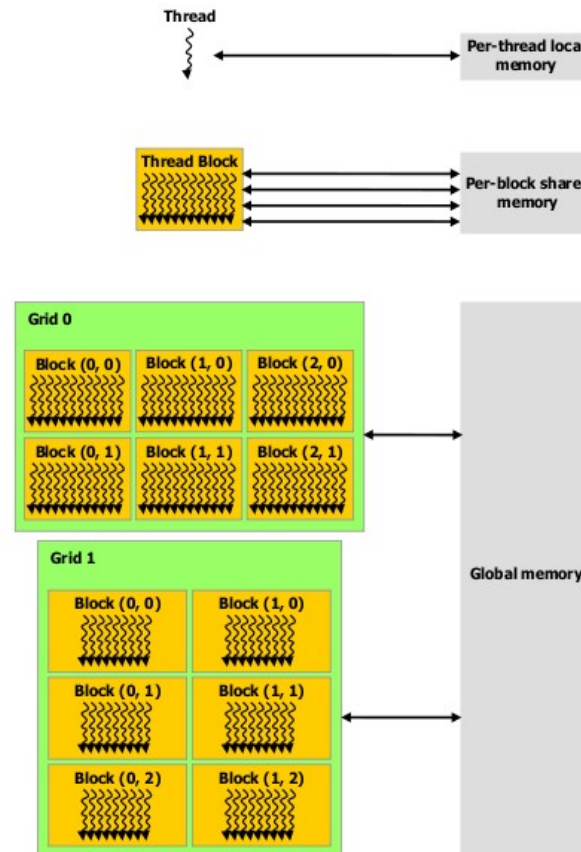
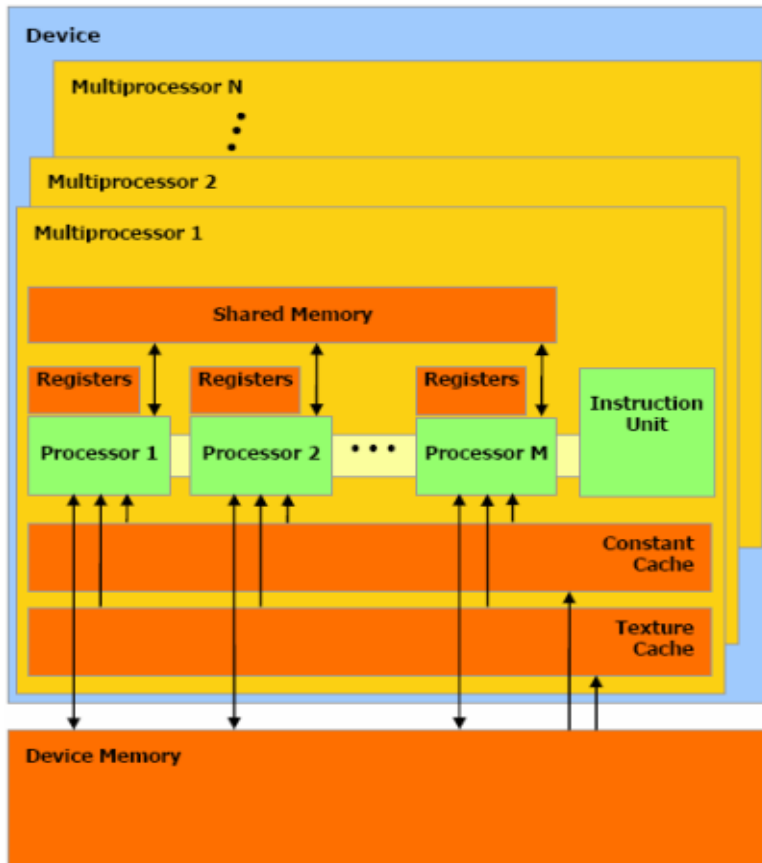
- Πολλοί τύποι μνήμης:



- Κάθε τύπος μνήμης έχει τα πλεονεκτήματα και τα μειονεκτήματα του αλλά και ειδική χρήση. Στις περισσότερες περιπτώσεις πρέπει να ακολουθούνται συγκεκριμένοι κανόνες για να έχουμε γρήγορη πρόσβαση.
 - Global memory: Η βασική μνήμη, μέρος της DRAM. Όποια μεταφορά μνήμης μεταξύ CPU και GPU περνάει από αυτή. Είναι σχετικά αργή και με μεγάλο latency αλλά μεγάλη σε μέγεθος. Οι προσβάσεις σε αυτή πρέπει να είναι ευθυγραμμισμένες και συνεχείς διαφορετικά θα υπάρχουν καθυστερήσεις λόγω του latency.
 - Shared memory: On-chip και ταχύτερη, μέρος του κάθε multiprocessor. Έχει όμως την εμβέλεια και την ζωή ενός block. Είναι πολύ περιορισμένη σε μέγεθος και κατάχρηση της μπορεί να οδηγήσει στην αδυναμία ενός multiprocessor να εκτελέσει πολλά warps ταυτόχρονα.
 - Registers: On-chip και ταχύτατοι, μέρος του κάθε multiprocessor. Έχουν την εμβέλεια και την ζωή ενός thread. Χρησιμοποιούνται ως “προσωπική” μνήμη ενός thread αλλά και για βελτιστοποίηση από τον compiler αποθηκεύοντας σε αυτές σταθερές παραστάσεις. Επίσης περιορισμένοι.

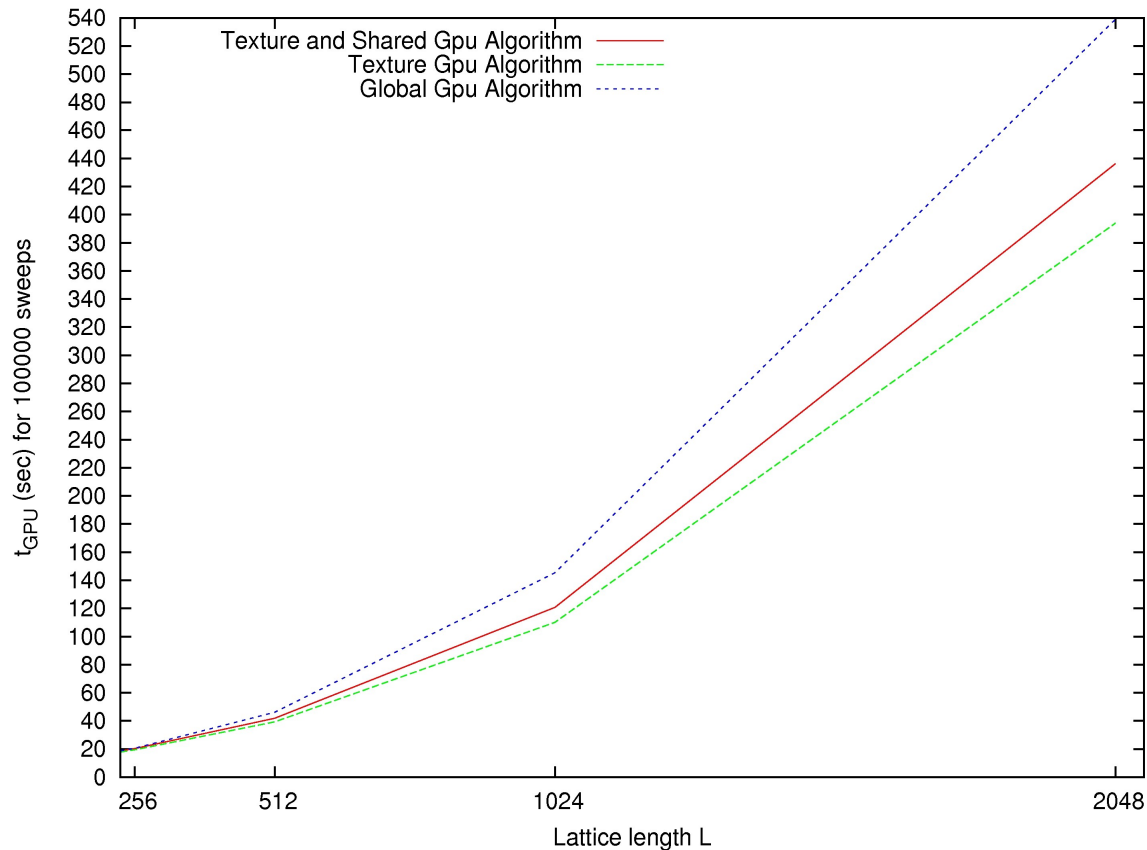
GPU Computing - Ειδικά

- Texture memory: Ειδική περίπτωση. Δεν αποτελεί ακριβώς μνήμη, είναι Global memory η οποία την οποία έχουμε θέσει να ελέγχεται μέσα από τα κυκλώματα texture ενός multiprocessor. Είναι πολύ πιο γρήγορη από την global επειδή είναι cached και υποστηρίζει υπολογισμό δεικτών σε hardware, hardware interpolation, κανονικοποιημένες συντεταγμένες αλλά και “ελεγχόμενο” overflow. Έχει επίσης το πλεονέκτημα ότι είναι βελτιστοποιημένη για spatial locality. Είναι όμως μνήμη ανάγνωσης μόνο!



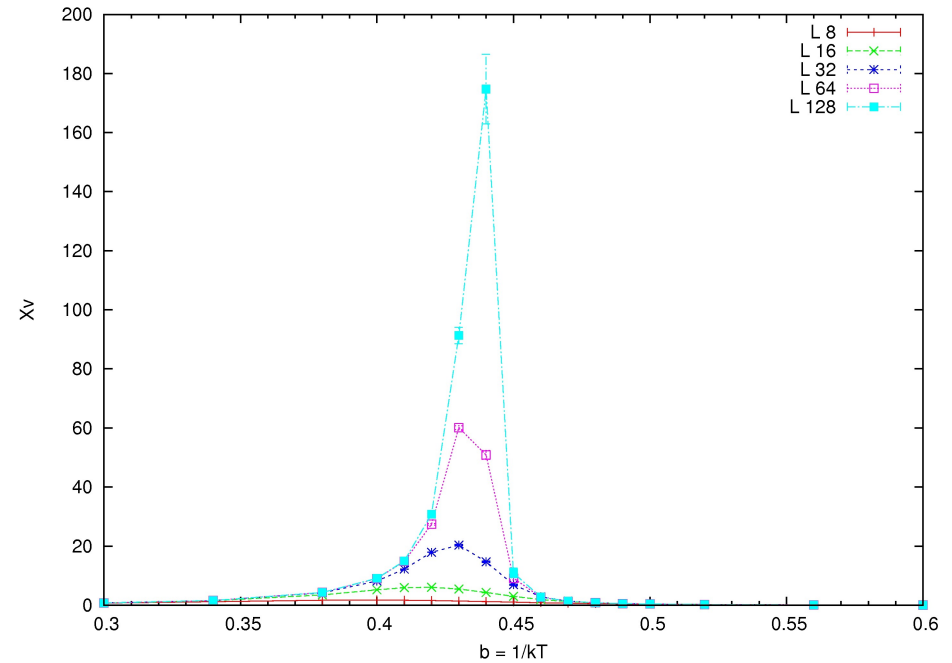
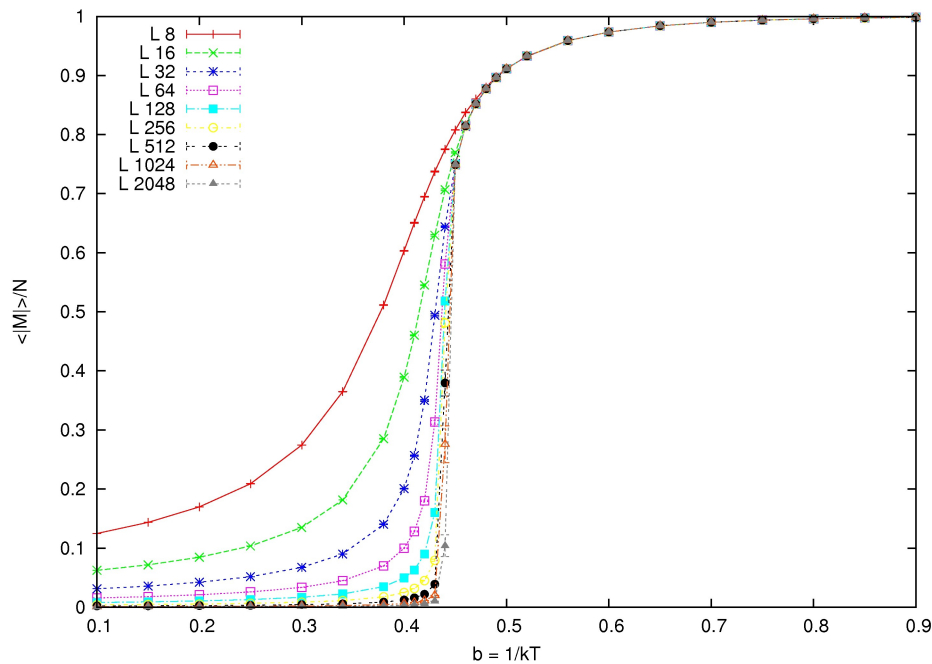
GPU Metropolis - Ειδικά

- Αναπτύχθηκαν 3 εκδόσεις του αλγορίθμου.
 1. Χρήση global memory μόνο.
 2. Χρήση global και texture memory.
 3. Χρήση texture και shared memory.



GPU Metropolis - Ειδικά

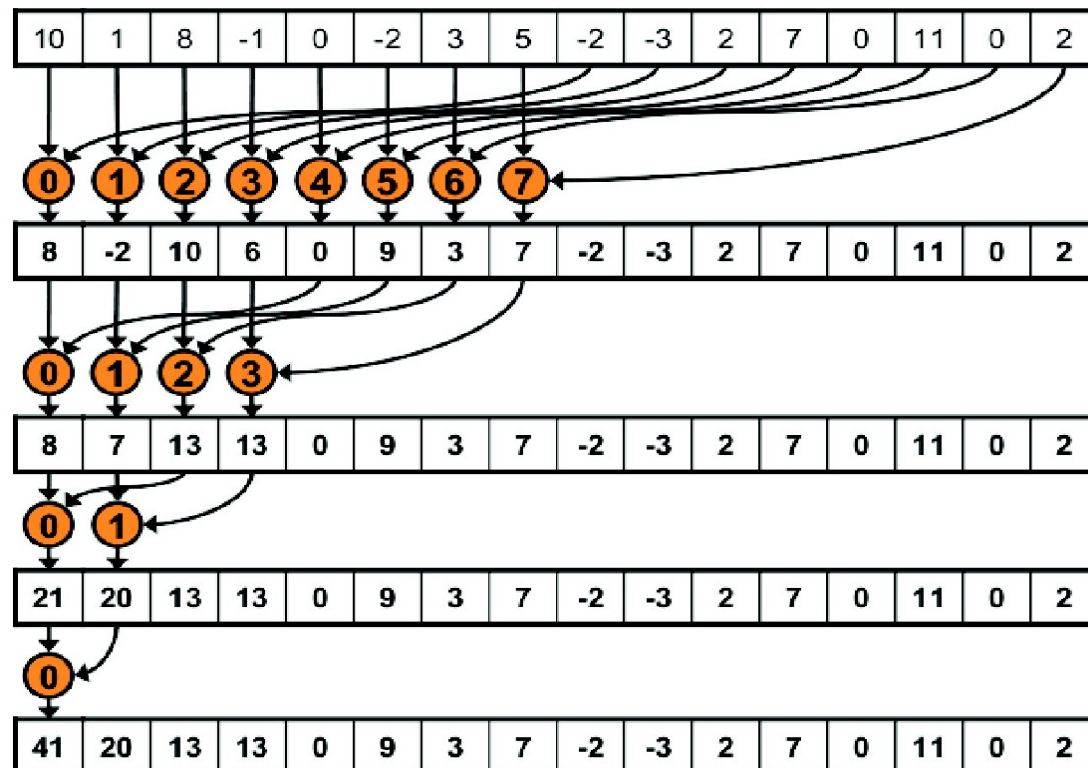
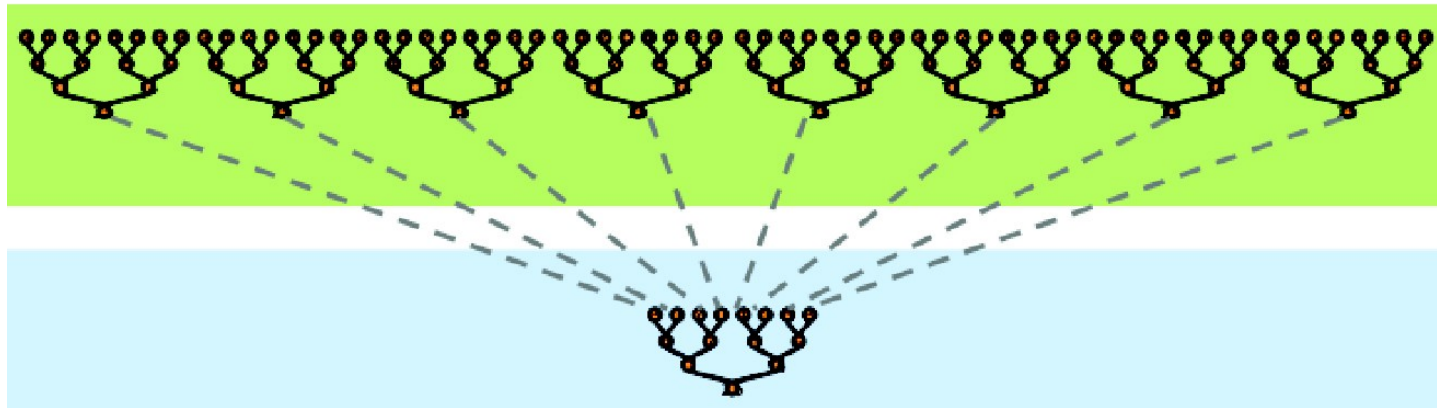
- Για μήκος πλέγματος L 2048 και 10^6 sweeps θα χρειάζονταν 4 μέρες για κάθε θερμοκρασία με τον σειριακό αλγόριθμο. Για τον αλγόριθμο GPU απαιτήθηκε μόλις ~ 1 ώρα ανά θερμοκρασία.
- Μερικά αποτελέσματα για L 8 έως 2048 και 10^6 sweeps.



GPU Tree Reduction

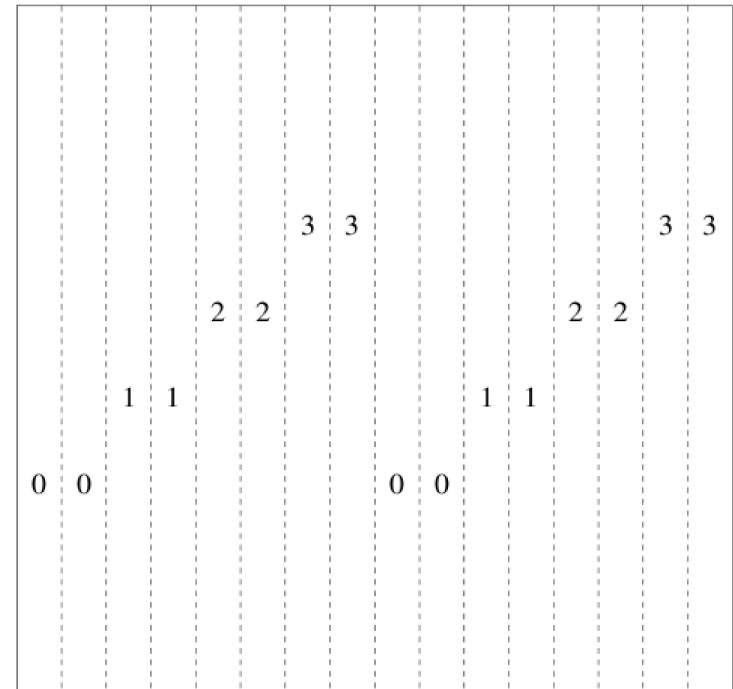
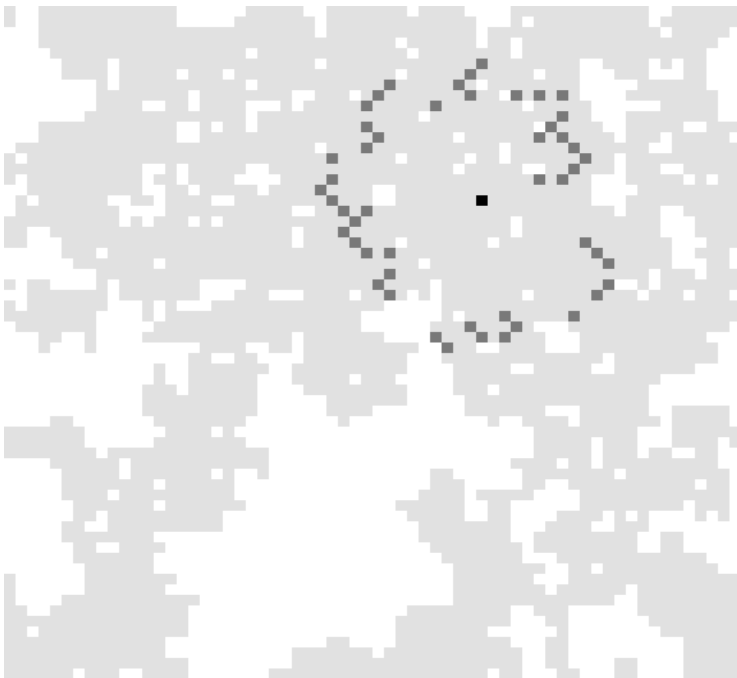
- Tree reduction: Αναδρομικός αλγόριθμος.
- Τα δεδομένα χωρίζονται σε δύο τμήματα.
- Τα threads τρέχουν πάνω στο πρώτο τμήμα και προσθέτει το καθένα ένα στοιχείο από το δεύτερο τμήμα στον εαυτό του.
- Επαναλαμβάνεται η διαδικασία έως ότου τα threads ενός block έχουν αθροιστεί σε έναν αριθμό.
- Το μέγεθος του block της προηγούμενης εκτέλεσης γίνεται το μέγεθος του grid και επαναλαμβάνεται η προηγούμενη διαδικασία έως ότου το μέγεθος του grid γίνει μονάδα.
- Τότε εκτελείται ο αλγόριθμος για μία τελευταία φορά και αθροίζει όλα τα προηγούμενα αποτελέσματα και επιστρέφει το τελικό αποτέλεσμα.

GPU Tree Reduction



Parallel Wolff

- Σε αντίθεση με τον αλγόριθμο Metropolis, σε ένα βήμα Wolff δεν ανανεώνεται όλο το πλέγμα.
- Το πλήθος των spin που θα ανανεωθούν είναι τυχαίο.
- Τα spin που ανανεώνονται σε κάθε γενιά του cluster είναι λίγα.
- Ένα spin γεννήτορας μπορεί να αλλάξει την τιμή των γειτονικών spin, επομένως λίγες μόνο γενικά ανεξάρτητες θέσεις στο πλέγμα και κατά κανόνα σε μία γενιά του cluster τα spin θα εξαρτώνται μεταξύ τους.
- Δύσκολο να υλοποιηθεί παράλληλα με αποδοτικό τρόπο.

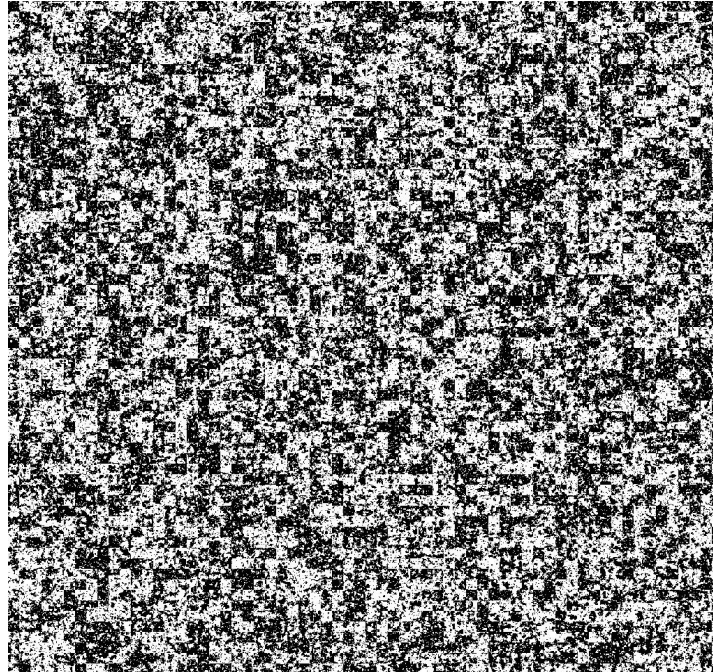


GPU Wolff

- Μέχρι τώρα στην βιβλιογραφία υπάρχει αποδοτική υλοποίηση μόνο σε λίγους πυρήνες CPU με domain decomposition του Lee.
- Η υλοποίηση σε GPU είναι ακόμα πιο δύσκολη:
 - Μικρό πλήθος ανεξάρτητων δεδομένων προς επεξεργασία.
 - Δεν υπάρχει συνοχή στις προσβάσεις στην μνήμη.
 - Το cluster μπορεί να είναι μικρό, σε αυτή την περίπτωση το overhead της εκτέλεσης στην GPU θα είναι μεγαλύτερο από την ίδιο τον χρόνο εκτέλεσης.
 - Γενικά δύσκολο να απασχολήσουμε πολλά threads.

GPU Wolff

- Μπορούμε να δοκιμάσουμε να εκτελέσουμε πολλές προσομοιώσεις ταυτόχρονα στην ίδια GPU.
- Θα έχουμε τώρα πρόβλημα λόγω τυχαίας εξάπλωσης του cluster.
- Η χρήση stack είναι πρόβλημα στις GPU καθώς δεν υπάρχει εγγύηση για την σειρά των εγγραφών. Η ενημέρωση του stack θα γίνεται από ένα thread αποκλειστικά.
- Γίνεται μόνο ο έλεγχος των γειτόνων παράλληλα ► λίγα threads ενεργά και πάλι.
- Ο αλγόριθμος δεν παρουσιάζει επιτάχυνση και περιορίζεται από το τεχνικά μικρό μέγεθος των blocks.



Στην συνέχεια τι;

- Για τον αλγόριθμο Metropolis,
 - Χρήση των textures για την επιτάχυνση περιοδικών συνοριακών συνθηκών στο πλέγμα.
- Για τον αλγόριθμο Wolff,
 - Προσομοίωση μεγάλου πλέγματος αντί πολλών μικρών.
 - Χρήση βοηθητικού πλέγματος αντί για stack.
 - Πιθανώς υλοποίηση domain decomposition του Lee.
 - Εκτέλεση μίας γενιάς στην GPU όταν το cluster έχει ένα σεβαστό μέγεθος μόνο.
 - Αποφυγή των συγχρονισμών στο wavefront μίας γενιάς χωρίζοντας τον έλεγχο γειτόνων σε τέσσερα βήματα.
- Για τον αλγόριθμο Tree reduction,
 - Εκμετάλευση των αδρανών threads για τον υπολογισμό των ποσοτήτων ταυτόχρονα.

