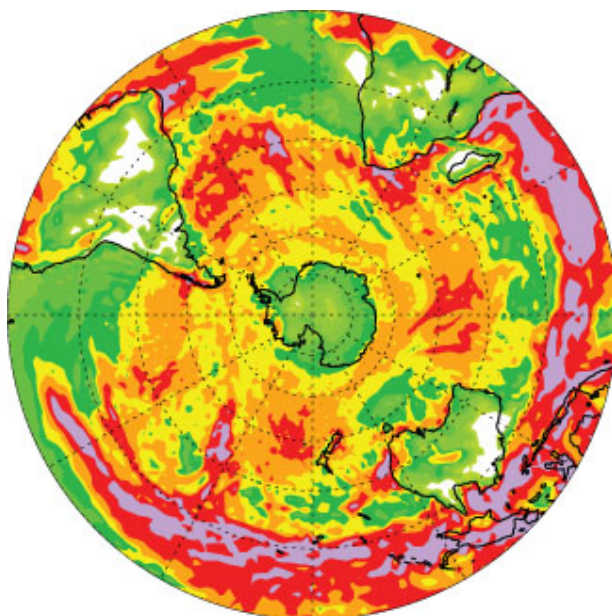


**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



Μάιος 2009

Ατομική Διπλωματική Εργασία

**ΕΦΑΡΜΟΓΗ  
ΤΟΥ ΜΟΝΤΕΛΟΥ ΠΡΟΒΛΕΨΗΣ ΚΑΙΡΟΥ  
WRF  
ΣΕ ΚΑΡΤΑ ΓΡΑΦΙΚΩΝ ΜΕ ΤΗΝ ΧΡΗΣΗ  
COMPUTE UNIFIED DEVICE ARCHITECTURE  
(C.U.D.A.)**

**Παναγιώτης Μάρκου**  
**Αρ. Ταυτότητας : Z0763955**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Μάιος 2009**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Εφαρμογή του μοντέλου πρόβλεψης καιρού WRF σε κάρτα γραφικών με την χρήση  
Compute Unified Device Architecture (C.U.D.A.)**

**Παναγιώτης Μάρκου**  
**Αρ. Ταυτότητας : Z0763955**

Επιβλέπων Καθηγητής  
Pedro Trancoso

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων  
απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου  
Κύπρου

Μάιος 2009

## Περίληψη

Σκοπός της εργασίας είναι η εκτέλεση του μοντέλου πρόβλεψης καιρού WRF σε κάρτα γραφικών με την χρήση CUDA και η επεξήγηση των λογισμικών που βοηθούν την εκτέλεση αυτή. Το μοντέλο πρόβλεψης καιρού είναι ένα μοντέλο που λειτουργεί με βάση κάποια συγκριτικά δεδομένα αλλά που μπορεί να εφαρμόζεται με ελάχιστη αλλαγή στο κώδικα και σε απαιτήσεις διαφόρων χρηστών. Έχουμε αναπτύξει την φύση του μοντέλου αυτού, τα ιδιαίτερα χαρακτηριστικά του και την σημαντικότητά του για τους ερευνητικούς οργανισμούς, τους χρήστες αλλά και τους ανθρώπους γενικότερα. Επίσης έχουμε αναλύσει τα αποτελέσματά του μοντέλου, την συγκρισιμότητα και την αξιοπιστία τους καθώς και την ταχύτητα που το μοντέλο παράγει τα αποτελέσματα.

Για την επιτυχία της γρήγορης ταχύτητας του μοντέλου πρόβλεψης καιρού καθοριστικό ρόλο παίζουν οι κάρτες γραφικών. Έχουμε αναφερθεί στο πώς η εξέλιξη στον τομέα των καρτών γραφικών βοήθησε στην βελτίωση του μοντέλου και στην κατανόηση των αποτελεσμάτων του από περισσότερους χρήστες. Οι κάρτες γραφικών έχουν γίνει ευρεία γνωστές και η χρήση και εφαρμογή τους με το χρόνο αυξάνεται. Στην εργασία αυτή θα αναφερθούμε στις εφαρμογές τους σε μερικούς τομείς και στις δυνατότητες που προσφέρει η εξέλιξή τους στο μέλλον για την βελτίωση της καθημερινής μας ζωής αλλά και στην έρευνα.

Για την εκτέλεση του μοντέλου προσαρμόσαμε το κώδικα έτσι ώστε αντί για FORTRAN, που είναι και η βασική γλώσσα που είναι γραμμένο το μοντέλο, να μπορεί να εκτελείτε κώδικας CUDA. Η CUDA (Compute Unified Device Architecture) είναι μια μέθοδος που αναπτύχθηκε από την εταιρία NVIDIA για να δώσει την δυνατότητα στους χρήστες να γράφουν εύκολα κώδικα ο οποίος μπορεί να τρέξει σε κάρτα γραφικών. Η εκτέλεση σε κάρτα γραφικών δίνει στο μοντέλο WRF την δυνατότητα να κάνει παράλληλα ακόμα περισσότερους υπολογισμούς, επιταχύνοντας έτσι την εκτέλεσή του. Θα αναφερθούμε στο πώς η CUDA επηρεάζει την εκτέλεση του μοντέλου μας, ποια είναι τα πλεονεκτήματα και πια τα μειονεκτήματα στην εφαρμογή αυτή.

Μέσα από την διαδικασία επεξήγησης του μοντέλου πρόβλεψης καιρού, των καρτών γραφικών και της CUDA αναλύουμε τα αποτελέσματα του μοντέλου που εμείς εκτελέσαμε. Περιγράφουμε επίσης τα καθοριστικά στάδια στην εκτέλεση του μοντέλου, και πώς μέσα από της δυνατότητες που μας πρόσφεραν τα λογισμικά που χρησιμοποιήσαμε αλλά

και μέσα από τους περιορισμούς που αντιμετωπίσαμε φτάσαμε στα συμπεράσματά μας. Καθώς έχουν γίνει πολλές έρευνες και μελέτες για το μοντέλο πρόβλεψης καιρού, τα συμπεράσματά μας θα προσπαθήσουμε να τα ερμηνεύσουμε και μέσα από τις έρευνες αυτές για να διαπιστώσουμε πως αυτά συγκρίνονται και υλοποιούνται στην πράξη.

# Περιεχόμενα

|                   |                                                                        |           |
|-------------------|------------------------------------------------------------------------|-----------|
| <b>Κεφάλαιο 1</b> | <b>Εισαγωγή</b>                                                        | <b>10</b> |
| <b>Κεφάλαιο 2</b> | <b>Το Μοντέλο Πρόβλεψης Καιρού WRF</b>                                 | <b>14</b> |
|                   | 2.1 Περιγραφή του WRF                                                  | 14        |
|                   | 2.2 Πώς δημιουργήθηκε το WRF                                           | 15        |
|                   | 2.3 Χρήσης του μοντέλου WRF                                            | 15        |
|                   | 2.4 Εκδόσεις και τεχνική διάσταση του WRF                              | 16        |
|                   | 2.5 Χαρακτηριστικά και Περιορισμοί                                     | 17        |
|                   | 2.6 Γραφικά εργαλεία του μοντέλου WRF                                  | 19        |
|                   | 2.6.1 NCL                                                              | 19        |
|                   | 2.6.2 RIP4                                                             | 19        |
|                   | 2.6.3 ARWpost                                                          | 20        |
|                   | 2.6.4 VAPOR                                                            | 21        |
| <b>Κεφάλαιο 3</b> | <b>Κάρτες Γραφικών</b>                                                 | <b>23</b> |
|                   | 3.1 Περιγραφή και γενική χρήση                                         | 23        |
|                   | 3.2 Οι ενσωματωμένες λύσεις γραφικών καρτών                            | 26        |
|                   | 3.3 Υβριδικές λύσεις                                                   | 27        |
|                   | 3.4 Nvidia Tesla                                                       | 28        |
| <b>Κεφάλαιο 4</b> | <b>CUDA (Compute Unified Device Architecture)</b>                      | <b>30</b> |
|                   | 4.1 Περιγραφή                                                          | 30        |
|                   | 4.2 Συσκευές που υποστηρίζουν GPU                                      | 31        |
|                   | 4.3 Πλεονεκτήματα                                                      | 32        |
|                   | 4.4 Περιορισμοί                                                        | 33        |
|                   | 4.5 Εφαρμογές της CUDA                                                 | 34        |
|                   | 4.5.1 Εφαρμογές της CUDA στην βιομηχανία πετρελαίου και φυσικού αερίου | 34        |
|                   | 4.5.2 Εφαρμογές της CUDA στην γεωγραφική πληροφόρηση                   | 35        |
|                   | 4.5.3 Εφαρμογές της CUDA στην ιατροφαρμακευτική βιομηχανία             | 37        |
|                   | 4.5.4 Εφαρμογές της CUDA στην βιομηχανία ενδυμάτων                     | 38        |

|                     |                                                                                  |           |
|---------------------|----------------------------------------------------------------------------------|-----------|
|                     | 4.5.5 Εφαρμογές της CUDA στην χρηματοοικονομική                                  | 39        |
|                     | 4.5.6 Εφαρμογές της CUDA στην ατομική δυναμική                                   | 40        |
| <b>Κεφάλαιο 5</b>   | <b>Σχετικές Μελέτες</b>                                                          | <b>41</b> |
|                     | “GPU Acceleration of Numerical Weather Prediction”                               | 41        |
|                     | “Weather Research and Forecast Model Port to Windows”                            | 42        |
| <b>Κεφάλαιο 6</b>   | <b>Εκτέλεση του μοντέλου WRF με ενσωματωμένο κώδικα CUDA</b>                     | <b>45</b> |
|                     | 6.1 Διαδικασία εκτέλεσης του μοντέλου WRF                                        | 45        |
|                     | 6.2 Περιγραφή του WSM5                                                           | 47        |
|                     | 6.3 Εργαλεία που χρησιμοποιήσα στην υλοποίηση                                    | 52        |
|                     | 6.3.1 Fortran                                                                    | 52        |
|                     | 6.3.2 gfortrun                                                                   | 52        |
|                     | 6.3.3 Linux                                                                      | 52        |
|                     | 6.3.4 AMD Code Analyst                                                           | 53        |
|                     | 6.3.5 AMD Athlon 3200 +                                                          | 54        |
|                     | 6.3.6 GCC                                                                        | 54        |
|                     | 6.3.7 GEFORCE 9600 GT                                                            | 55        |
|                     | 6.4 Χρόνοι Μέτρησης                                                              | 57        |
|                     | 6.4.1 Χρόνοι Μέτρησης - Test Case “EM_grav2d_x”                                  | 57        |
|                     | 6.4.2 Χρόνοι Μέτρησης - - Test Case “EM_heldsuarez”                              | 60        |
|                     | 6.4.3 Χρόνοι Μέτρησης - Test Case “EM_hill2d”                                    | 62        |
|                     | 6.4.4 Χρόνοι Μέτρησης - Τέταρτη εκτέλεση Χρόνοι Μέτρησης - Test Case “EM_b_wave” | 64        |
|                     | 6.5 Συμπεράσματα                                                                 | 67        |
| <b>Appendix</b>     | <b>Αναλυτικές μετρήσεις για όλα τα test cases</b>                                | <b>69</b> |
| <b>Βιβλιογραφία</b> | <b>A1. Άρθρα και Παρουσιάσεις</b>                                                | <b>78</b> |
|                     | <b>A2. Ιστοσελίδες</b>                                                           | <b>79</b> |

## ΕΙΚΟΝΕΣ

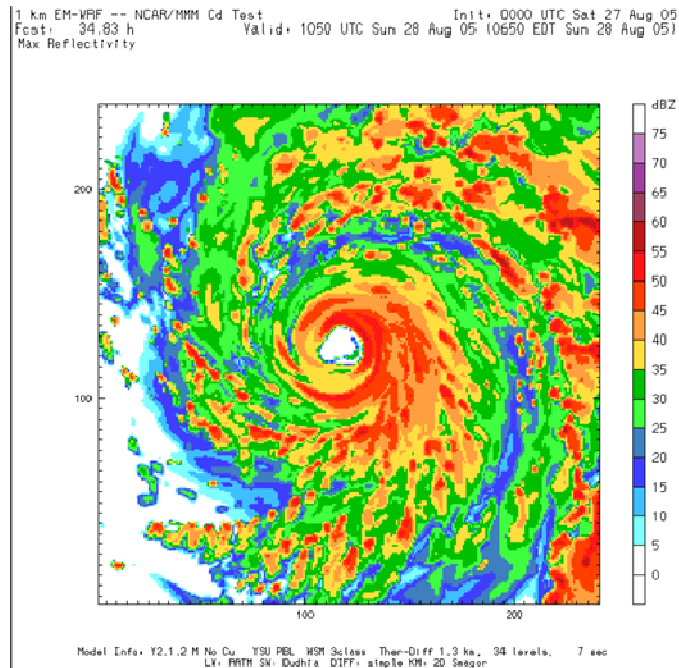
|                                |                                                                                                                                                  |           |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Εικόνα 1<sup>η</sup> :</b>  | Παρουσίαση της ατμόσφαιρας με το μοντέλο WRF                                                                                                     | <b>10</b> |
| <b>Εικόνα 2<sup>η</sup> :</b>  | Ο τυφώνας Gustav όταν έφτασε στην μεγαλύτερη έντασή του πλησιάζοντας την Νότια Κούβα                                                             | <b>11</b> |
| <b>Εικόνα 3<sup>η</sup> :</b>  | Ο τυφώνας Gustav σε WRF                                                                                                                          | <b>11</b> |
| <b>Εικόνα 4<sup>η</sup> :</b>  | Εκτόξευση του δορυφόρου Fengyun-3 (FY-3) για σκοπούς πρόβλεψης καιρού                                                                            | <b>12</b> |
| <b>Εικόνα 5<sup>η</sup> :</b>  | Το μοντέλο WRF και ο τρόπος λειτουργίας του (1)                                                                                                  | <b>17</b> |
| <b>Εικόνα 6<sup>η</sup> :</b>  | Το μοντέλο WRF και ο τρόπος λειτουργίας του (2)                                                                                                  | <b>18</b> |
| <b>Εικόνα 7<sup>η</sup> :</b>  | Κάρτα γραφικών XFX GeForce 9600 GT                                                                                                               | <b>23</b> |
| <b>Εικόνα 8<sup>η</sup> :</b>  | Σύγκριση στην απόδοση GPU και CPU                                                                                                                | <b>24</b> |
| <b>Εικόνα 9<sup>η</sup> :</b>  | Νέα μοντέλα GPUs σε σύγκριση με την θερμική τους απόδοση                                                                                         | <b>25</b> |
| <b>Εικόνα 10<sup>η</sup> :</b> | Νέα μοντέλα GPUs σε σύγκριση με τα επίπεδα θορύβου                                                                                               | <b>25</b> |
| <b>Εικόνα 11<sup>η</sup> :</b> | Νέα μοντέλα GPUs σε σύγκριση με την κατανάλωση ενέργειας                                                                                         | <b>26</b> |
| <b>Εικόνα 12<sup>η</sup> :</b> | Intel GMA X3000 IGP (under heatsink).                                                                                                            | <b>26</b> |
| <b>Εικόνα 13<sup>η</sup> :</b> | Υβριδική κάρτα EN9800GT                                                                                                                          | <b>27</b> |
| <b>Εικόνα 14<sup>η</sup> :</b> | Κάρτα γραφικών γενικής εφαρμογής NVIDIA TESLA                                                                                                    | <b>28</b> |
| <b>Εικόνα 15<sup>η</sup> :</b> | Grid of Thread Blocks                                                                                                                            | <b>31</b> |
| <b>Εικόνα 16<sup>η</sup> :</b> | Εφαρμογή της CUDA για γεωλογικούς σκοπούς                                                                                                        | <b>34</b> |
| <b>Εικόνα 17<sup>η</sup> :</b> | Εφαρμογή της CUDA για γεωγραφικούς σκοπούς                                                                                                       | <b>35</b> |
| <b>Εικόνα 18<sup>η</sup> :</b> | Παρουσίαση της επιφάνειας της θάλασσας στην Νότια Αφρική όπως είναι σήμερα                                                                       | <b>36</b> |
| <b>Εικόνα 19<sup>η</sup> :</b> | Παρουσίαση της επιφάνειας της θάλασσας στην Νότια Αφρική όπως ήταν 100 εκατομμύρια χρόνια πριν όταν η Αφρική ήταν πιο κοντά με την Νότια Αμερική | <b>36</b> |
| <b>Εικόνα 20<sup>η</sup> :</b> | Παρουσίαση της επιφάνειας της γης με την χρήση γραφικών καρτών                                                                                   | <b>37</b> |
| <b>Εικόνα 21<sup>η</sup> :</b> | Εφαρμογή της CUDA στη ιατροφαρμακευτική βιομηχανία                                                                                               | <b>37</b> |
| <b>Εικόνα 22<sup>η</sup> :</b> | Εφαρμογή της CUDA στη βιομηχανία ενδυμάτων                                                                                                       | <b>38</b> |
| <b>Εικόνα 23<sup>η</sup> :</b> | Εφαρμογή της CUDA στη Χρηματοοικονομική                                                                                                          | <b>39</b> |
| <b>Εικόνα 24<sup>η</sup> :</b> | Εφαρμογή της CUDA στη Ατομική Δυναμική (1)                                                                                                       | <b>40</b> |
| <b>Εικόνα 25<sup>η</sup> :</b> | Εφαρμογή της CUDA στη Ατομική Δυναμική (2)                                                                                                       | <b>40</b> |
| <b>Εικόνα 26<sup>η</sup> :</b> | Απόδοση του Windows Compute Cluster Server 2003 σε WRF 12km CONUS standard benchmark case                                                        | <b>43</b> |
| <b>Εικόνα 27<sup>η</sup> :</b> | Αποτέλεσμα ενός μοντέλου WRF που τρέχει σε Windows Compute Cluster Server 2003 σε WRF 12km CONUS standard benchmark case                         | <b>43</b> |

## ΠΙΝΑΚΕΣ

|                                 |                                                               |           |
|---------------------------------|---------------------------------------------------------------|-----------|
| <b>Πίνακας 1<sup>ος</sup> :</b> | Προδιαγραφές των καρτών γενικής εφαρμογής της<br>NVIDIA TESLA | <b>28</b> |
| <b>Πίνακας 2<sup>ος</sup> :</b> | Συσκευές που υποστηρίζουν GPUs                                | <b>31</b> |

# Κεφάλαιο 1

## Εισαγωγή



**Εικόνα 1<sup>η</sup> : Παρουσίαση της ατμόσφαιρας με το μοντέλο WRF**

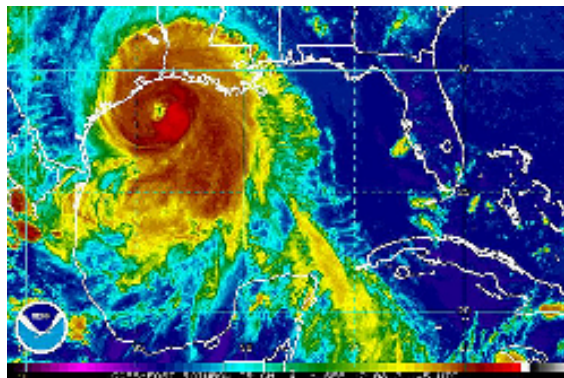
Οι δολοφονικοί τυφώνες και η παγκόσμια αλλαγή του κλίματος, είναι ανεξέλεγκτοι παράγοντες και επηρεάζουν την καθημερινή μας ζωή. Κατά καιρούς τα άσχημα καιρικά φαινόμενα επηρεάζουν ακόμη και την επιβίωσή μας λόγω της σφοδρότητας τους και της ανεξέλεγκτης πορείας τους. Οι προβλέψεις των καιρικών φαινομένων γίνονται τώρα πιο γρήγορα και πιο έγκαιρα, δίνοντας στους ανθρώπους περισσότερο χρόνο για να προετοιμαστούν.

Το 2005, ο τυφώνας Katrina είχε καταστροφικές συνέπειες για την αμερικανική πολιτεία της Νέας Ορλεάνης. Ο τυφώνας προκάλεσε τον θάνατο τουλάχιστον 1800 πολιτών, πλημμύρες και αμέτρητες καταστροφές. Τρία χρόνια αργότερα ο τυφώνας Gustav, απείλησε με χειρότερες καταστροφές οδηγώντας δυο εκατομμύρια πολίτες στην εγκατάλειψη των κατοικιών τους. Το Εθνικό Κέντρο Τυφώνων (NHC – National Hurricane Center) μπόρεσε να κάνει έγκαιρη πρόβλεψη για το επικίνδυνο καιρικό φαινόμενο σχετικά με την ένταση και την ταχύτητά του κάνοντας χρήση του μοντέλου WRF και έτσι οι πολίτες ενημερώθηκαν να

εγκαταλείψουν τα σπίτια τους ή να πάρουν μέτρα προστασίας πριν ο τυφώνας τους απειλήσει άμεσα.



**Εικόνα 2<sup>η</sup> : Ο τυφώνας Gustav όταν έφτασε στην μεγαλύτερη έντασή του πλησιάζοντας την Νότια Κούβα.**



**Εικόνα 3<sup>η</sup> : Ο τυφώνας Gustav σε WRF**

Διάφορες χώρες έχουν εκτοξεύσει δορυφόρους για μετεωρολογικούς σκοπούς με σκοπό να συλλέξουν δεδομένα όπως η Κίνα το 2008 η οποία εκτόξευσε το Fengyun-3 για να βοηθήσει στην πρόβλεψη του καιρού την περίοδο των Ολυμπιακών Αγώνων. Αυτός ήταν ο δεύτερος μετεωρολογικός δορυφόρος που εκτόξευσε η Κίνα. Οι λεπτομερείς προβλέψεις του καιρού ήταν σημαντικές γιατί οι καιρικές συνθήκες θα είχαν σοβαρές επιπτώσεις στις προσπάθειες του Πεκίνου να περιορίσει την ατμοσφαιρική ρύπανση στη διάρκεια των

Αγώνων. Η ρύπανση του αέρα θα είχε ως αποτέλεσμα αγώνες αντοχής να πραγματοποιηθούν σε μέρη που η ρύπανση ήταν χαμηλότερη. Επίσης για την ομαλή έκβαση κάποιων αγώνων ήταν απαραίτητη και η πρόβλεψη βροχοπτώσεων.



**Εικόνα 4<sup>η</sup> : Εκτόξευση του δορυφόρου Fengyun-3 (FY-3) για σκοπούς πρόβλεψης καιρού .**

Ο δορυφόρος επίσης συνέβαλλε στην γεωγραφική έρευνα, στις αερομεταφορές, την πλοήγηση, στη γεωργία, τη δασοκομία, στην ωκεανογραφία και στην πρόβλεψη του καιρού και σε διάφορες άλλες χώρες . Ο δορυφόρος ήταν σε θέση να διεξάγει τρισδιάστατη, παντός καιρού, ανίχνευση ποσοτικών δεδομένων στην επιφάνεια του εδάφους, του ωκεανού, και της ατμόσφαιρας. Τα δεδομένα αυτά αξιοποιήθηκαν από ιδρύματα που χρησιμοποιούν το WRF για την διεξαγωγή προσομοιώσεων και προβλέψεων για τις κλιματολογικές συνθήκες κατά την διάρκεια των αγώνων.

Στο Εθνικό Κέντρο Ατμοσφαιρικής Έρευνας (NCAR), μια ομάδα επιστημόνων παρατηρώντας την σημασία της πρόβλεψης του καιρού στην ζωή μας ανέπτυξε εξελιγμένα μοντέλα άμεσης πρόβλεψης για μακροπρόθεσμες και βραχυπρόθεσμες καιρικές συνθήκες. Το Μοντέλο Πρόβλεψης Καιρού WRF είναι το πιο ευρέως χρησιμοποιούμενο μοντέλο στον κόσμο. Οι χρήστες του, συμπεριλαμβάνουν την National Weather Service, την Air Force Weather Agency, υπηρεσίες καιρικών προγνώσεων σε διάφορες χώρες , εμπορικές εταιρείες που επηρεάζονται άμεσα από τις προβλέψεις , πανεπιστήμια για σκοπούς έρευνας και μελέτης καθώς και διάφορους χρήστες.

Αρχικά τα μετεωρολογικά μοντέλα NCAR μετακινούνταν από Terascale (1 τρις flops) σε Petascale εφαρμογές, φθάνοντας σε σημείο όπου η πρόσθεση περισσότερων CPU δεν ήταν πλέον αποτελεσματική για τη βελτίωση της ταχύτητας των μοντέλων. Το πρόβλημα ήταν ιδιαίτερα έντονο με εφαρμογές που αφορούσαν την πρόβλεψη σε πραγματικό χρόνο και όχι με ιδεατά δεδομένα. Κατά την εξέταση τρόπων για να βελτιωθεί η συνολική ταχύτητα και η ακρίβεια της πρόβλεψης, η NCAR σε συνεργασία με τους ερευνητές στο Πανεπιστήμιο του Κολοράντο στο Boulder χρησιμοποίησε NVIDIA GPU. Η ταχύτητα του μοντέλου αυξήθηκε κατά δέκα φορές στους υπολογισμούς του WRF. Η μετατροπή του μοντέλου σε CUDA οδήγησε σε βελτίωση της ταχύτητας του συνολικού μοντέλου κατά 20 φορές .

Τώρα η NCAR, και οι υπηρεσίες σε όλο τον κόσμο που στηρίζονται στο WRF, είναι σε θέση να παράγουν προβλέψεις σε πραγματικό χρόνο, για τις καιρικές συνθήκες και για συνθήκες που είναι γρήγορα μεταβαλλόμενες . Ως αποτέλεσμα της δημιουργίας μοντέλων πρόβλεψης καιρού και ειδικότερα χρήσης του μοντέλου WRF, οι άνθρωποι είναι σε θέση να προετοιμαστούν για να ξεπεράσουν χωρίς ζημιές τα δύσκολα καιρικά φαινόμενα.

## Κεφάλαιο 2

### Μοντέλο Πρόβλεψης Καιρού WRF

#### 2.1 Περιγραφή του WRF

Το Μοντέλο WRF είναι ένα από τα αριθμητικά πρότυπα που χρησιμοποιούνται παγκοσμίως στην πρόγνωση του καιρού. Η λειτουργία του βασίζεται στην αριθμητική επίλυση μαθηματικών εξισώσεων με την βοήθεια προσεγγιστικών μεθόδων στηριζόμενες σε θεωρίες που καλύπτει ο κλάδος της Φυσικής. Η όλη αυτή διαδικασία προϋποθέτει την ύπαρξη κάποιων αρχικών οριακών συνθηκών οι οποίες λαμβάνονται από μετρήσεις που πραγματοποιούνται σε πραγματικό χρόνο στην ατμόσφαιρα με την βοήθεια των διαφόρων μετεωρολογικών οργάνων. Σε περιοχές όπου δεν είναι δυνατή η πραγματοποίηση μετρήσεων από τον άνθρωπο, όπως είναι για παράδειγμα οι ωκεανοί, χρησιμοποιούνται μετρήσεις από δορυφόρους. Μετά την επεξεργασία αυτών των δεδομένων από το μοντέλο, παράγονται οι προγνωστικοί χάρτες καιρού με την βοήθεια ενός πακέτου γραφικών.

Το μοντέλο πρόβλεψης καιρού WRF είναι ένα σύστημα πρόβλεψης καιρικών φαινομένων μεσαίας κλίμακας, που έχει γνωρίσει μεγάλη εξέλιξη τα τελευταία χρόνια. Είναι σχεδιασμένο να εξυπηρετεί σκοπούς έρευνας της ατμόσφαιρας, της επιφάνειας της γης, των ωκεανών αλλά επίσης βρίσκει απήχηση και σε πολλές άλλες εφαρμογές. Το WRF είναι ένα προσιτό μοντέλο πρόβλεψης καιρού. Η δομή του είναι τέτοια έτσι ώστε να μπορεί να επεκταθεί εύκολα αφού αποτελείται από δυναμικά επιμέρους κομμάτια. Η καινοτομία αυτή το κάνει εύχρηστο όχι μόνο για πραγματική εφαρμογή αλλά και για ευρύ φάσμα πειραματισμού. Κάποιος μπορεί να προσθέσει εύκολα κομμάτια που χρειάζονται για τις ανάγκες του. Επίσης είναι σχεδιασμένο να εκμεταλλεύεται την δυνατότητα παράλληλης επεξεργασίας δεδομένων κάποιων συστημάτων, εκτελώντας πράξεις παράλληλα. Ο αυξανόμενος αριθμός επεξεργαστών αυξάνει και την ταχύτητα του μοντέλου. Επίσης η απόδοση του μοντέλου είναι μεγαλύτερη όσο αυξάνεται και η ταχύτητα του δικτύου όπου λειτουργούν οι επεξεργαστές.

Το μοντέλο WRF δίνει στους ερευνητές την δυνατότητα να κάνουν εξομοιώσεις που να αντιπροσωπεύουν πραγματικά και ιδεατά δεδομένα. Με αυτό τον τρόπο οι ερευνητές μπορούν να έχουν αποτελέσματα για διάφορα ενδεχόμενα και μπορούν να κάνουν καλύτερη

πρόβλεψη για μελλοντικά γεγονότα καθώς επίσης και να επεξηγήσουν καλύτερα φαινόμενα που συνέβησαν στο παρελθόν. Επίσης μπορούν να δουν προσομοιώσεις σε βάθος χρόνου ανακαλύπτοντας συσχετισμούς σε καταστάσεις και διάφορα καιρικά φαινόμενα. Το μοντέλο αυτό πρόβλεψης είναι ευέλικτο και αποδοτικό υπολογιστικά όσον αφορά την επεξεργασία των δεδομένων και επιπλέον αυξάνει τις δυνατότητες της φυσικής και αριθμητικής στην πρόβλεψη. Έτσι το μοντέλο παρέχει πιο αξιόπιστη και έγκαιρη πρόβλεψη. Οι χρήστες του μοντέλου αυτού συνεχώς αυξάνονται καθώς επίσης βελτιώνεται η λειτουργικότητά του και οι δυνατότητές του. Η συνεχής ενημέρωση που παίρνουν οι χρήστες για το πώς να αξιοποιήσουν πιο αποτελεσματικά το μοντέλο αυτό έχει οδηγήσει σε αυξημένες απαιτήσεις και γρήγορη ανάπτυξη.

## **2.2 Πως δημιουργήθηκε το WRF**

Το WRF τώρα είναι σε λειτουργία στο Εθνικό Κέντρο Περιβαλλοντικών Προβλέψεων (NCEP-National Centers of Environmental Prediction). Το μοντέλο είναι ελεύθερο να χρησιμοποιηθεί για κοινή χρήση. Η ανάπτυξή του ήταν ένα αποτέλεσμα της συνεργασίας των πιο κάτω οργανισμών:

- του National Center for Atmospheric Research (NCAR),
- του National Oceanic and Atmospheric Administration
- του Air Force Weather Agency (AFWA),
- του Naval Research Laboratory,
- του University of Oklahoma, και
- του Federal Aviation Administration (FAA).

## **2.3 Χρήσης του μοντέλου WRF**

Το μοντέλο είναι κατάλληλο να χρησιμοποιηθεί για ευρείας χρήσης εφαρμογές που είναι όμως περιορισμένης έκτασης και μπορούν να κυμαίνονται από μερικά μέτρα σε χιλιάδες χιλιόμετρα συμπεριλαμβανομένου και:

- Εξιδανικευμένη εξομοίωση (π.χ. διάδοση θερμότητας με ρεύματα)
- Περιφερικές ή παγκόσμιες εφαρμογές
- Οριοθετημένη έρευνα
- Έρευνες για αφομοίωση δεδομένων
- Έρευνες πρόβλεψης

- Εφαρμογές πολλαπλών μοντέλων
- Εκπαίδευση

Το Μετεωρολογικό Τμήμα Μεσαίας και Μικρής Κλίμακας της NCAR συντηρεί τώρα και υποστηρίζει ένα υποσύνολο του όλου κώδικα WRF, τρίτης έκδοσης το οποίο συμπεριλαμβάνει:

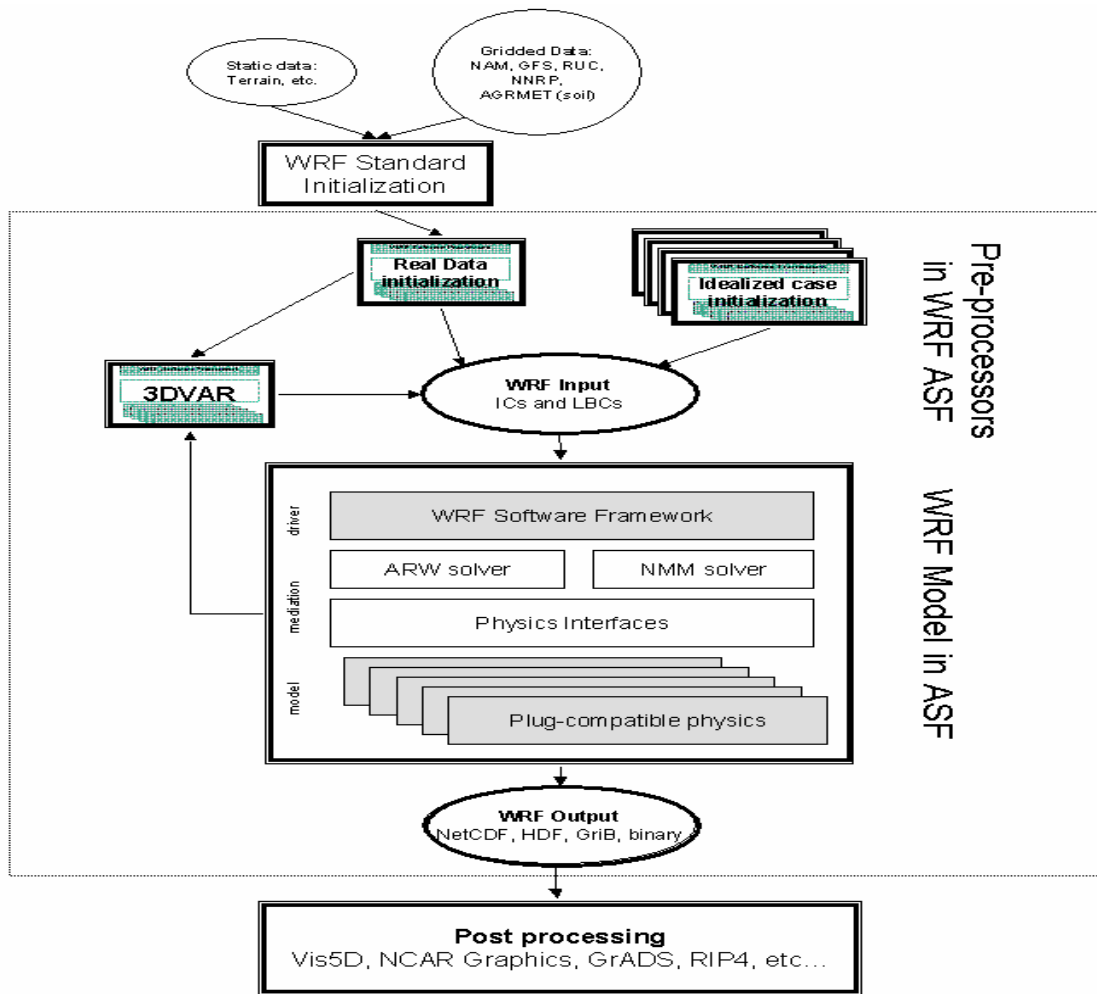
- Advanced Research WRF (ARW) dynamic solver,
- WRF Pre-Processing System (WPS),
- WRF-Var data assimilation system,
- Διάφορα πακέτα φυσικής που έχουν συνεισφέρει στο Εθνικό Κέντρο Ατμοσφαιρικής Έρευνας (NCAR) το οποίο λειτουργείται από το Πανεπιστημιακό Ίδρυμα Ατμοσφαιρικής Έρευνας UCAR (University Corporation for Atmospheric Research).

## 2.4 Εκδόσεις και τεχνική διάσταση του WRF

WRF® είναι ένα εγγεγραμμένο εμπορικό σήμα του Πανεπιστημιακού Ιδρύματος Ατμοσφαιρικής Έρευνας UCAR και η τελευταία έκδοσή του είναι το 3.1. Η έκδοση 2.1 και όλες οι μετέπειτα εκδόσεις υποστηρίζουν δυο τύπους κινήσεων, 2-way διαδραστικά δίκτυα.

- **Καθορισμένη κίνηση :** Ο χρήστης καθορίζει nest moves από μια λίστα με επιλογές.
- **Αυτόματη κίνηση:** Χρησιμοποιεί μεσαίου επιπέδου vortex ακολουθώντας έναν αλγόριθμο το οποίο δίνει την δυνατότητα εξομοίωσης τροπικών καταιγίδων.

Και στους δυο τύπους το nest ενεργοποιείται χρησιμοποιώντας ένα παρεμβαλλόμενο δίκτυο δεδομένων, συμπεριλαμβανομένου του χώρου και άλλων δεδομένων που καθορίζουν το χώρο.



Εικόνα 5<sup>η</sup> : Το μοντέλο WRF και ο τρόπος λειτουργίας του (1)

## 2.5 Χαρακτηριστικά και Περιορισμοί

Χαρακτηριστικά:

- Πλήρης παραλληλισμένο
- Telescoping
- Αποδοτικό

Περιορισμοί:

- Leading edge initialization by interpolation from coarse domain only
- Δεν υπάρχει έλεγχος των κινήσεων
- EM-core only

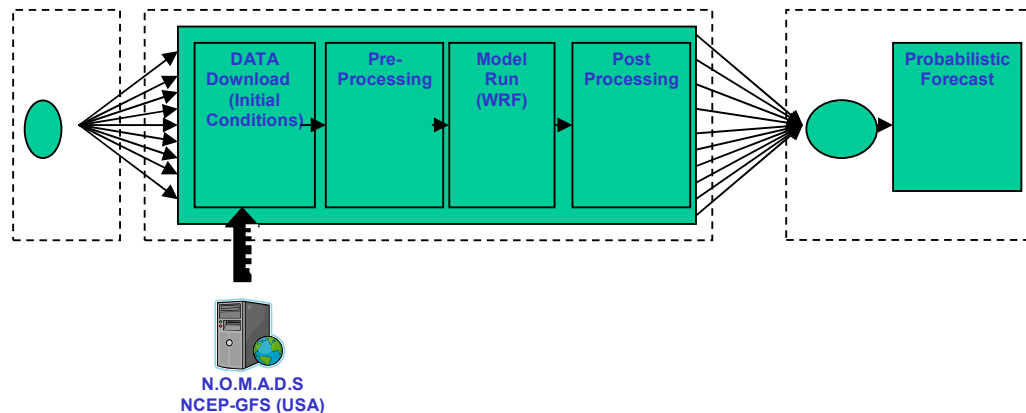
Πιο συγκεκριμένα το μοντέλο αποτελείται από 3 μέρη:

- (1) Τους preprocessors,
- (2) Το WRF Advanced Software Framework (ASF) και,
- (3) Τους postprocessors.

Οι preprocessors ετοιμάζουν τα δεδομένα εισόδου ώστε να τα δεχτεί το κεντρικό κομμάτι για επεξεργασία. Το ASF κάνει τους υπολογισμούς που απαιτούνται σε κάθε περίπτωση. Τέλος οι postprocessors δείχνουν τα αποτελέσματα των υπολογισμών με ένα κατανοητό τρόπο που συνήθως είναι ο γραφικός τρόπος.

Το ASF χωρίζεται και εκείνο σε 3 επίπεδα:

- (1) Το driver layer που ελέγχει την ροή της επεξεργασίας και την μεταφορά δεδομένων μέσα στο ASF ή έξω από αυτό.
- (2) Το meditation layer που είναι υπεύθυνο για την σωστή κλήση όλων των συναρτήσεων που πρέπει να εκτελεστούν σε ένα χρονικό βήμα του μοντέλου.
- (3) Το model layer όπου εκεί βρίσκονται υλοποιημένες όλες οι συναρτήσεις που καλούνται από το meditation layer.



**Εικόνα 6<sup>η</sup> : Το μοντέλο WRF και ο τρόπος λειτουργίας του (2)**

Η περιοχή που εξετάζει το μοντέλο δηλώνεται ως parent domain και χωρίζεται σε πιο μικρά domains. Κάθε domain είναι χωρισμένο σε tiles. Έτσι η περιοχή υπάρχει μέσα στο προγραμματισμό ως ένα δέντρο με κορυφή το parent domain και παιδιά τα πιο μικρά domains. Το ASF μπορεί να εκτελεί παράλληλα τους υπολογισμούς για το κάθε domain. Ο

παράλληλισμός αυτός γίνεται στο meditation layer χρησιμοποιώντας RSL communication library που με την σειρά του χρησιμοποιεί MPI. Το RSL είναι υπεύθυνο για την επικοινωνία των επεξεργαστών κατά την παράλληλη επεξεργασία. Επίσης παράλληλη επεξεργασία γίνεται και στους ARW NMM Solvers με την χρήση OpenMP. Κατά την παράλληλη επεξεργασία στα domains παραχωρείτε distributed memory και το κάθε tile μέσα στο domain έχει πρόσβαση σε μια shared memory που είναι αποκλειστικά για τα tiles του domain αυτού.

## **2.6 Γραφικά εργαλεία του μοντέλου WRF**

Υπάρχει ένας αριθμός διαθέσιμων εργαλείων για την οπτικοποίηση και απεικόνιση των δεδομένων του μοντέλου WRF. Το NCAR στο παρόν υποστηρίζει τα εξής εργαλεία γραφικών, NCL, RIP4, ARWpost (Grads και Vis5d), και VAPOR. Στην εκτέλεση του μοντέλου στην εργασία αυτή χρησιμοποιήσα το RIP4.

### **2.6.1 NCL**

NCL (NCAR Command Language) είναι μια πολύ ισχυρή γλώσσα η οποία έχει πολλά πλεονεκτήματα έναντι των παραδοσιακών NCAR κωδικών γραφικών που χρησιμοποιούν Fortran ή C.

Χαρακτηριστικά και Περιορισμοί:

- Δεν απαιτεί μεγάλο σύνολο δεδομένων. Τα scripts διαβάζουν αρχεία WRF απευθείας
- Κατακόρυφη παρεμβολή στην πίεση, το ύψος των επιπέδων, κ.λπ. χειρίζεται κατά τη λειτουργία (on-the-fly) από plotting scripts, και προσδίδουν προστιθέμενη ευελιξία.
- Μπορεί να επεξεργαστεί δεδομένα WRF εισόδου και εξόδου.
- Μπορεί να επεξεργαστεί δεδομένα από πραγματικό χρόνο, 2-D και ιδανικές προσομοιώσεις.
- Είναι διαθέσιμος ένας σχετικά μεγάλος αριθμός μεθόδων διάγνωσης
- Απαιτεί NCAR Command Language libraries), version 4.3.1 or higher.

### **2.6.2 RIP4**

Το RIP, Read/Interpolate/Plot, είναι ένα πρόγραμμα Fortran που χρησιμοποιεί NCAR Graphics System Plot Package Simulator για την δημιουργία πεδίων (plots)

και μοντέλων παραγωγής μεσαίας κλίμακας. Το RIP είναι ικανό να παράγει οριζόντια ή κάθετη τομή πεδίων (countours) ή vector fields (barbs ή arrows), κάθετα προφίλ και sounding. Πρωταρχικό του πλεονεκτήματα είναι η φορητότητα, ο πλήρης και ενημερωμένος φάκελος και το ευρύ σύνολο των διαγνωστικών δεδομένων που είναι διαθέσιμα για τον υπολογισμό και την σχεδίαση. Το "RIP4" αναφέρεται στην τέσσερις μεγάλες εκδόσεις RIP.

#### Χαρακτηριστικά και Περιορισμοί:

- Απαιτεί μεγάλο ενδιαμέσο σύνολο στοιχείων. Ένα πρόγραμμα που ονομάζεται "ripdr\_wrf" μετατρέπει τα αρχεία δεδομένων του WRF σε αρχεία που το RIP μπορεί να διαβάσει.
- Κατακόρυφη παρεμβολή στην πίεση, το ύψος των επιπέδων, κ.λπ. χειρίζεται κατά τη λειτουργία (on-the-fly) από plotting scripts, και προσδίδουν προστιθέμενη ευελιξία.
- Μπορεί να επεξεργαστεί δεδομένα εισόδου και εξόδου.
- Μπορεί να επεξεργαστεί δεδομένα από πραγματικό καιρός, 2-D και ιδανικές προσομοιώσεις.
- Δεν χρησιμοποιεί το λογισμικό WRF I / O , έτσι μόνο δεδομένα netCDF μπορούν να υποβάλλονται σε επεξεργασία.
- Είναι διαθέσιμος ένας σχετικά μεγάλος αριθμός μεθόδων διάγνωσης
- Απαιτεί NCAR Graphics libraries.

#### 2.6.3 ARWpost

Το εργαλείο αυτό είναι ένας μετατροπέας , που μετατρέπει μόνο τα δεδομένα εισόδου που απαιτούνται από GrADS/Vis5d, δηλαδή, δεν δημιουργούνται γραφικά από αυτό το εργαλείο. Εφόσον τα στοιχεία είναι στη σωστή μορφή, ο χρήστης εξακολουθεί να πρέπει να γνωρίζει πώς να χρησιμοποιεί GrADS/Vis5d για να είναι σε θέση να δημιουργήσει τα αναγκαία plots.

#### Χαρακτηριστικά και Περιορισμοί:

- Δημιουργεί ένα μεγάλο ενδιαμέσο σύνολο δεδομένων
- Κατακόρυφη παρεμβολή στην πίεση, το ύψος των επιπέδων, κλπ., πρέπει να επιλέγεται κατά το στάδιο μετατροπής δεδομένων. Μόλις τα ενδιαμέσα

στοιχεία δημιουργηθούν, μπορούν μόνο να παρουσιάζονται από GrADS/Vis5d.

- Μπορεί να επεξεργαστεί WPS geogrid και metgrid, και WRF-ARW δεδομένα εισόδου και εξόδου.
- Μπορεί να επεξεργαστεί δεδομένα από πραγματικό καιρός, 2-D και ιδανικές προσομοιώσεις.
- Είναι διαθέσιμος μόνο μερικές μέθοδοι διάγνωσης
- Απαιτεί GrADS visualization software
- Απαιτεί Vis5d visualization software

#### 2.6.4 VAPOR

VAPOR είναι η Οπτικοποίηση και η Ανάλυση Πλατφόρμας για Ωκεανούς, την Ατμόσφαιρα, και για Solar Researchers. Το VAPOR αναπτύχθηκε από την NCAR για την παροχή διαδραστικής οπτικοποίησης και για την ανάλυση των αριθμητικά προσομοιωμένων ρευστοδυναμικών (fluid dynamics) Η έκδοση (1.2) του VAPOR υποστηρίζει την οπτικοποίηση των δεδομένων εξόδου του WRF-ARW.

##### **Βασικές ικανότητες του VAPOR με τα δεδομένα εξόδου του WRF-ARW**

- **Direct Volume rendering (DVR)** - Κάθε μεταβλητή 3D στα δεδομένα WRF μπορεί να θεωρηθεί ως πυκνότητα. Οι χρήστες ελέγχουν την διαφάνεια και το χρώμα για να δουν τη θερμοκρασία, την υγρασία, σύννεφα, κ.λπ., σε 3D.
- **Ροή** - Σχεδιάζει 2D και 3D βέλη, τα οποία δείχνουν την κατεύθυνση του ανέμου και την κίνηση, τον τρόπο που αλλάζει ο άνεμος με το χρόνο. Επίσης επιτρέπει την ιχνογράφηση και την οπτικοποίηση της πορείας των σωματιδίων στο χρόνο.
- **Isosurfaces** - Οι isosurfaces μεταβλητών εμφανίζονται διαδραστικά. Οι χρήστες μπορούν να ελέγχουν τις τιμές ISO (ISO-values), το χρώμα και τη διαφάνεια των isosurfaces.

- **Animation** - Ελέγχει το time-stepping των δεδομένων, για την διαδραστική αναπαραγωγή και για την καταγραφή κινούμενων ακολουθιών.
- **Terrain rendering** - Η επιφάνεια του εδάφους μπορεί να παρουσιάζεται έγχρωμη και ως τρισδιάστατη εικόνα για γεω-αναφορά.

## Κεφάλαιο 3

### Κάρτες Γραφικών



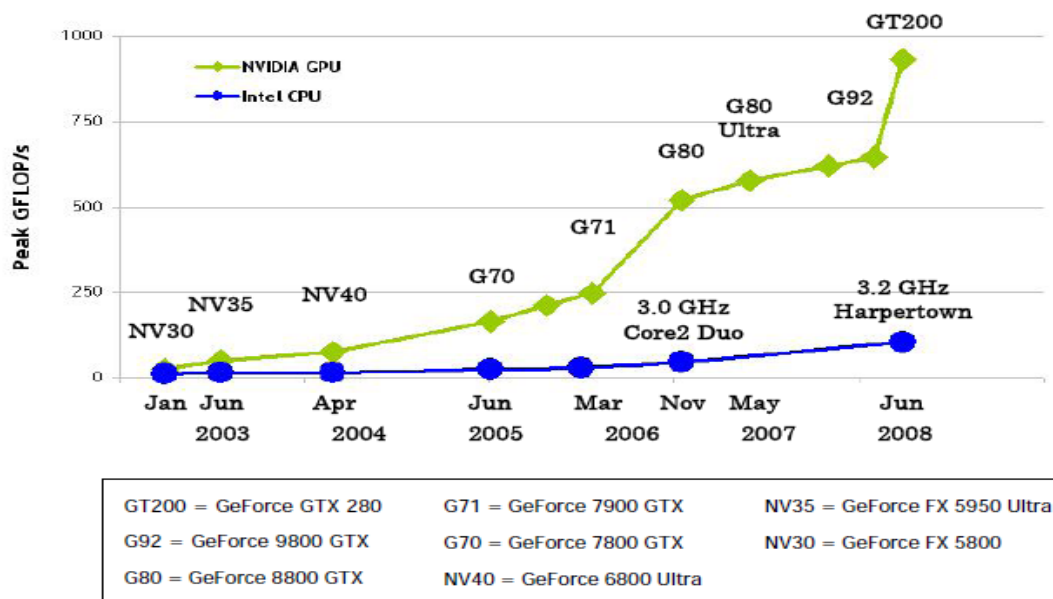
Εικόνα 7<sup>η</sup> : Κάρτα γραφικών XFX GeForce 9600 GT

#### 3.1 Περιγραφή και γενική χρήση

GPU είναι μια μονάδα επεξεργασίας γραφικής παράστασης για ένα προσωπικό Η/Υ, έναν τερματικό σταθμό, ή μια κονσόλα παιχνιδιών. Τα σύγχρονα GPUs είναι πολύ αποδοτικά στο χειρισμό και την επίδειξη της ηλεκτρονικής γραφιστικής, και η ιδιαίτερα παράλληλη δομή τους, τους καθιστά αποτελεσματικότερα από τις γενικής χρήσης CPU για μια σειρά σύνθετων αλγορίθμων. Ένα GPU μπορεί να τρέξει πάνω σε μια τηλεοπτική κάρτα, ή μπορεί να ενσωματωθεί άμεσα στη μητρική κάρτα. Τα GPUs των υπολογιστών γραφείου και των φορητών υπολογιστών είναι συνήθως λιγότερο

Ένα GPU είναι ένας επεξεργαστής που συνδέεται με μια κάρτα γραφικών που αφιερώνεται στον υπολογισμό των διαδικασιών. Ένας επιταχυντής γραφικής παράστασης ενσωματώνει τα μικροτσίπ συνήθειας που περιέχουν τις πρόσθετες μαθηματικές διαδικασίες που χρησιμοποιούνται στην απόδοση γραφικής παράστασης. Η αποδοτικότητα των μικροτσίπ επομένως καθορίζει την αποτελεσματικότητα του επιταχυντή γραφικής παράστασης. Χρησιμοποιούνται κυρίως για τρισδιάστατα παιχνίδια. Ένα GPU εφαρμόζει διάφορες διαδικασίες πρωτόγονων γραφικής παράστασης με τέτοιο τρόπο ώστε τις κάνει να τρέχουν πολύ γρηγορότερα.

Το ενδιαφέρον που υπάρχει για τις GPUs δεν είναι τυχαίο. Τα τελευταία χρόνια οι αποδόσεις τους όλο και αυξάνονται και η τιμή τους όλο και μειώνεται. Όπως φαίνεται και στην πιο κάτω γραφική παράσταση οι επιδόσεις στις κάρτες γραφικών φτάνουν τα 900 GFLOPS/s στην NVIDIA GeForce GTX 280 ενώ ένας CPU της ίδιας χρονολογίας είναι στα 150 GFLOPS/s . Πολλές επιχειρήσεις έχουν παραγάγει GPUs με διάφορα εμπορικά σήματα. Το 2008, η Intel, NVIDIA και AMD/ATI ήταν οι ηγέτες της αγοράς. . Επίσης GPUs παράγουν η VIA Technologies/S3 Graphics και η Matrox.

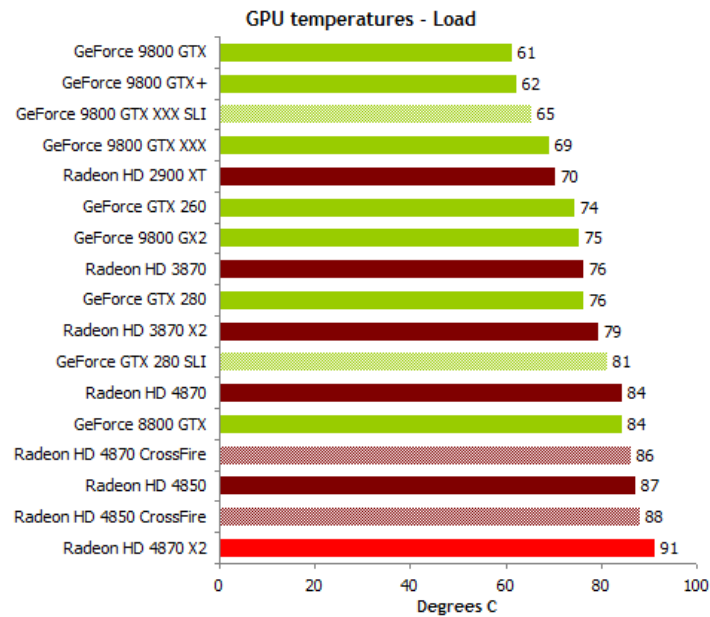


**Εικόνα 8<sup>η</sup> : Σύγκριση στην απόδοση GPU και CPU**

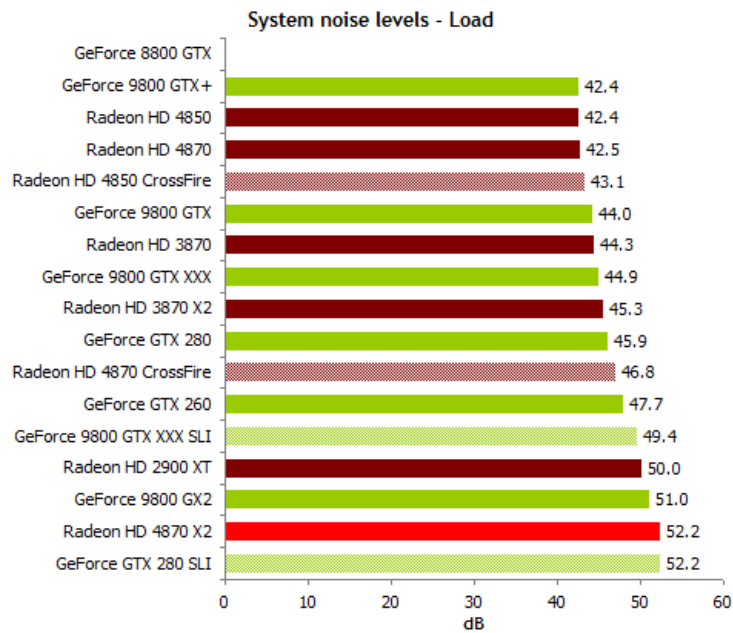
Όλο και περισσότεροι επιστημονικοί τομείς στρέφονται προς τις κάρτες γραφικών για τις αποδόσεις τους. Στην αναζήτησή μου για κάποιο σχετικό με το θέμα μου άρθρο είδα άρθρα για μαθηματικές εφαρμογές, για εφαρμογές βιολογίας, για εφαρμογές για βάσεων δεδομένων κ.α. Αυτό δείχνει το μεγάλο ενδιαφέρον που υπάρχει για τις κάρτες γραφικών. Στο ενδιαφέρον αυτό συμβάλει και η ανάπτυξη ευκολότερων μεθόδων προγραμματισμού των GPUs όπως είναι η CUDA της NVIDIA. Η CUDA προσφέρει στους προγραμματιστές ένα εύκολο μοντέλο προγραμματισμού που δίνει την δυνατότητα εκμετάλλευσης των ισχυρών δυνατοτήτων των νέων καρτών γραφικών.

Οι τεχνολογίες όπως SLI από NVIDIA και ATI επιτρέπουν τον πολλαπλασιασμό των GPUs και την δημιουργία μιας ενιαίας εικόνας , που αυξάνει τη διαθέσιμη δύναμη επεξεργασίας των γραφικών παραστάσεων. Έρευνες που έγιναν για τα GPUs δείχνουν ότι

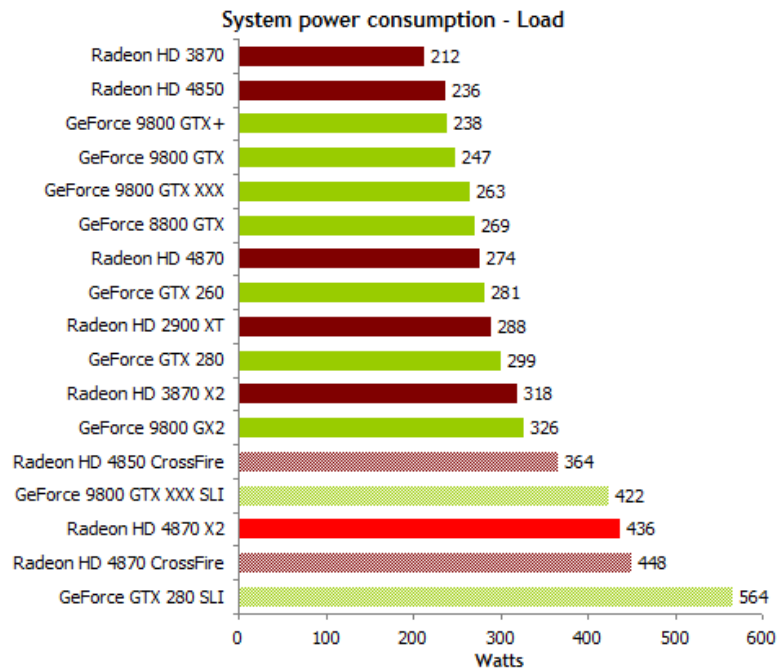
αυτές ξοδεύουν όλο και λιγότερο ενέργεια, ζεσταίνονται πιο αργά και προκαλούν λιγότερο θόρυβο.



**Εικόνα 9<sup>η</sup> : Νέα μοντέλα GPUs σε σύγκριση με την θερμική τους απόδοση**

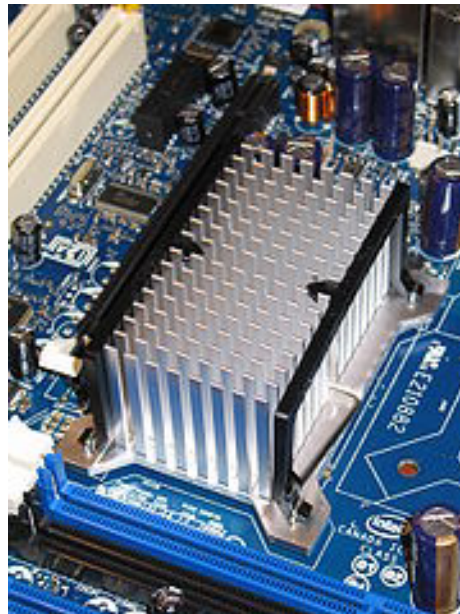


**Εικόνα 10<sup>η</sup> : Νέα μοντέλα GPUs σε σύγκριση με τα επίπεδα θορύβου**



Εικόνα 11<sup>η</sup> : Νέα μοντέλα GPUs σε σύγκριση με την κατανάλωση ενέργειας.

### 3.2 Οι ενσωματωμένες λύσεις γραφικών καρτών



Εικόνα 12<sup>η</sup> : Intel GMA X3000 IGP (under heatsink)

Οι ενσωματωμένες λύσεις γραφικών καρτών, ή οι κοινές λύσεις γραφικών καρτών είναι επεξεργαστές γραφικής παράστασης που χρησιμοποιούν μια μερίδα ενός computer' RAM παρά την αφιερωμένη μνήμη γραφικής παράστασης. Οι υπολογιστές με την

ενσωματωμένη γραφική κάρτα αποτελούν 90% όλων των PC. Αυτές οι λύσεις είναι φτηνότερες για να εφαρμόσουν από τις αφιερωμένες λύσεις γραφικής παράστασης, αλλά είναι λιγότερο ικανές. Ιστορικά, οι ενσωματωμένες λύσεις θεωρήθηκαν συχνά ακατάλληλες για να παίξουν τα τρισδιάστατα παιχνίδια ή τα εντατικά προγράμματα τρεξίματος γραφικών. Εντούτοις, σήμερα, οι ενσωματωμένες λύσεις όπως το Intel' GMA X3000 (Intel G965 chipset), AMD' s Radeon HD 3200 (AMD 780G chipset) και NVIDIA' το GeForce 8200 (NVIDIA nForce 730a) είναι περισσότερο αποδοτικά. Τα τσιπ όπως το Nvidia 9400M στο νέο Macbook έχουν βελτιωμένη απόδοση. Μερικές ενσωματωμένες μητρικές κάρτες υπολογιστών γραφείου περιλαμβάνουν συχνά μια ενσωματωμένη λύση γραφικής παράστασης και έχουν τις αυλακώσεις επέκτασης διαθέσιμες για να προσθέσουν μια αφιερωμένη κάρτα γραφικών αργότερα.

### 3.3 Υβριδικές λύσεις

Αυτή η νεώτερης κατηγορίας GPU ανταγωνίζεται με την ενσωματωμένη γραφική κάρτα στις αγορές υπολογιστών γραφείου και σημειωματάρων. Οι πιο κοινές εφαρμογές αυτού είναι ATI' s HyperMemory και NVIDIA' s TurboCache. Οι υβριδικές κάρτες γραφικών είναι κάπως ακριβότερες από την ενσωματωμένη γραφική κάρτα, αλλά πολύ λιγότερο ακριβές από τις αφιερωμένες κάρτες γραφικών. Η Sony και η Asus έχουν ήδη ενσωματώσει στους υπολογιστές τους τις υβριδικές κάρτες. Οι υβριδικές κάρτες καταναλώνουν λιγότερη ενέργεια και είναι πιο φιλικές προς το περιβάλλον.



**Εικόνα 13<sup>η</sup> : Υβριδική κάρτα EN9800GT**

### 3.4 Nvidia Tesla



**Εικόνα 14<sup>η</sup> : Κάρτα γραφικών γενικής εφαρμογής NVIDIA TESLA**

Η Tesla είναι το GPU που η NVIDIA δημιούργησε για γενικούς σκοπούς.

Λόγω της πολύ υψηλής υπολογιστικής δύναμης (που μετριέται στις διαδικασίες floating points ανά δευτερόλεπτο) έναντι των πρόσφατων μικροεπεξεργαστών, τα προϊόντα Tesla προορίζονται για την αγορά υπολογισμού υψηλής επίδοσης. Η αρχική λειτουργία των προϊόντων tesla είναι να βοηθήσει στις προσομοιώσεις, στους υπολογισμούς μεγάλων κλιμάκων (ειδικά floating-point υπολογισμών), και στην παραγωγή εικόνας για τους επαγγελματικούς και επιστημονικούς τομείς, με τη χρήση CUDA. Οι λύσεις NVIDIA® Tesla™ βοηθούν να λυθούν οι πιο σημαντικές προκλήσεις υπολογισμού πιο γρηγορότερα και αξιόπιστα σε πολλές βιομηχανίες.

| Configuration            | Model | # of GPUs | Core clock in MHz (each) | Shaders                   |                     | Memory               |          |                           |                  |             | Processing Power (GigaFLOPS, total) | Form factor and features     |
|--------------------------|-------|-----------|--------------------------|---------------------------|---------------------|----------------------|----------|---------------------------|------------------|-------------|-------------------------------------|------------------------------|
|                          |       |           |                          | Thread Processors (total) | Clock in MHz (each) | Bandwidth max (GB/s) | Bus type | Bus width (bit, each GPU) | Total size (MiB) | Clock (MHz) |                                     |                              |
| GPU Computing Processor1 | C870  | 1         | 600                      | 128                       | 1350                | 77                   | GDDR3    | 384                       | 1536             | 1600        | 519                                 | Full-height video card       |
| Deskside Supercomputer1  | D870  | 2         | 600                      | 256                       | 1350                | 154                  | GDDR3    | 384                       | 3072             | 1600        | 1037                                | Deskside system or Rack unit |
| GPU Computing Server1    | S870  | 4         | 600                      | 512                       | 1350                | 307                  | GDDR3    | 384                       | 6144             | 1600        | 2074                                | 1U Rack                      |

|                                           |       |   |     |     |      |     |       |     |       |      |      |                                                        |
|-------------------------------------------|-------|---|-----|-----|------|-----|-------|-----|-------|------|------|--------------------------------------------------------|
| C1060<br>Computing<br>Processor 2         | C1060 | 1 | 602 | 240 | 1300 | 102 | GDDR3 | 512 | 4096  | 1600 | 936  | Full-height<br>video card<br>IEEE 754r<br>capabilities |
| S1070 1U<br>GPU<br>Computing<br>Server2,3 | S1070 | 4 | 602 | 960 | 1500 | 410 | GDDR3 | 512 | 16384 | 1600 | 4320 | 1U Rack<br>IEEE 754r<br>capabilities                   |

**Πίνακας 1<sup>ος</sup> : Προδιαγραφές των καρτών γενικής εφαρμογής της NVIDIA  
TESLA**

## Κεφάλαιο 4

### C.U.D.A. Compute Unified Device Architecture

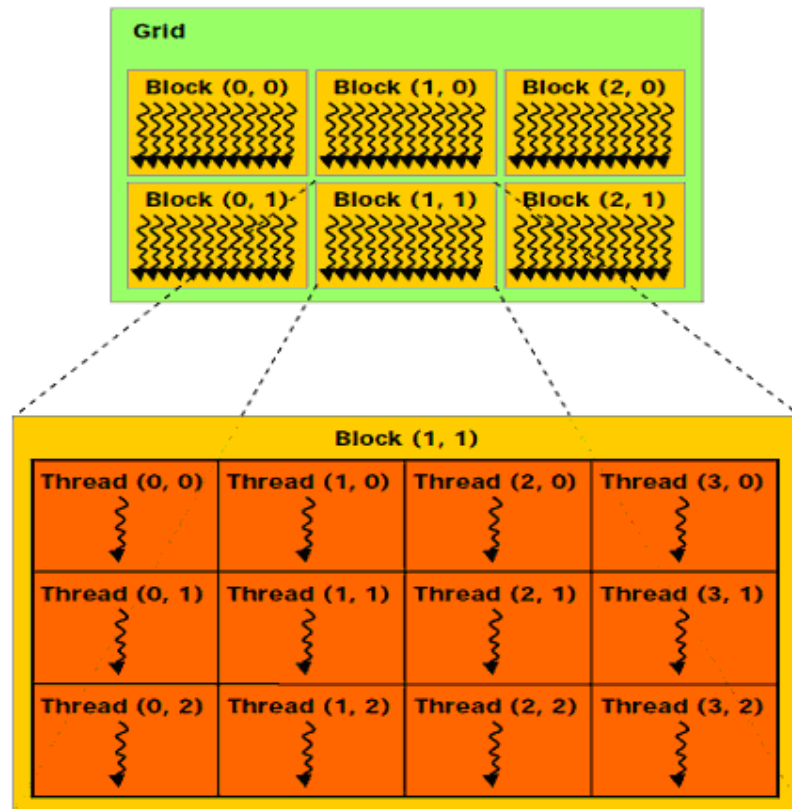
#### 4.1 Περιγραφή

Η κύρια αιτία που δημιούργησε την ανάγκη για ένα μοντέλο σαν τη CUDA είναι η αυξανόμενη προτίμηση της αγοράς σε συστήματα με πολλούς επεξεργαστές. Τέτοια συστήματα μπορεί να βασίζονται είτε σε επεξεργαστές με πολλούς πυρήνες είτε σε κάρτες γραφικών με πολλούς επεξεργαστές. Για να μπορέσουμε να αξιοποιήσουμε όμως τέτοια συστήματα χρειαζόμαστε εφαρμογές που προσαρμόζονται εύκολα σε συστήματα με διαφορετικό αριθμό επεξεργαστών. Αυτό προσπαθεί να κατορθώσει το CUDA.

Το CUDA είναι ένα προγραμματιστικό μοντέλο αλλά και ένα μοντέλο κατασκευής υλικού που αναπτύχθηκε από την NVIDIA. Όσο αφορά το προγραμματιστικό κομμάτι είναι εύκολο στην εκμάθηση γιατί βασίζεται στην γλώσσα προγραμματισμού C προσθέτοντας κάποιες εύκολες επιπλέον εντολές που δίνουν στον προγραμματιστή την δυνατότητα να γράψει “παράλληλα” προγράμματα. Για να μπορέσουν όμως οι εντολές που θα γράψουμε να κάνουν κάτι θέλουν το κατάλληλο υλικό περιβάλλον. Πρέπει το υλικό να έχει κάποια συγκεκριμένη δομή. Προς το παρόν τα μόνα υλικά του υπολογιστή που υποστηρίζουν CUDA είναι οι κάρτες γραφικών της NVIDIA σειράς 8 και πάνω.

Το βασικό γνώρισμα της CUDA είναι η ιεραρχία των εργασιών που εκτελούνται κάθε φορά και η ιεραρχία της διαθέσιμης μνήμης. Οι εργασίες (threads) μοιράζονται σε blocks και τα blocks σε grids. Κάθε thread έχει πρόσβαση στην τοπική δική του μνήμη. Επίσης έχει πρόσβαση στην τοπική μνήμη του block και τέλος έχει πρόσβαση στην καθολική μνήμη όπου έχουν πρόσβαση όλα τα threads. Αυτή η οργάνωση κάνει εύκολη την ανάθεση των threads σε επεξεργαστές και τον συγχρονισμό τους. Η ιδανική εκτέλεση υποθέτει ότι κάθε thread “τρέχει” σε ένα επεξεργαστή. Έτσι αν το πρόγραμμα δημιουργεί αρκετά ανεξάρτητα threads ώστε να αναθέσουμε ένα thread σε κάθε επεξεργαστή τότε έχουμε την ιδανική περίπτωση. Αν δημιουργεί περισσότερα από τους επεξεργαστές τότε τα threads εναλλάσσονται στους επεξεργαστές που τελειώνουν την εκτέλεση τους. Αν δημιουργεί λιγότερα τότε δεν υπάρχει πρόβλημα. Τα threads που δημιουργούνται μπαίνουν αυτόματα στον πρώτο ελεύθερο επεξεργαστή.

Επειδή μόνο οι κάρτες γραφικών υποστηρίζουν προς το παρόν τη CUDA η εκτέλεση ενός προγράμματος αναγκαστικά θα πρέπει να εναλλάσσεται μεταξύ CPU και GPU. Αυτά τα δυο δεν μπορούν να εκτελούν πράξεις παράλληλα. Ο παραλληλισμός συμβαίνει μόνο κατά την εκτέλεση του προγράμματος στο GPU. Επίσης το καθένα τους έχει την δική του μνήμη. Έτσι η καθολική μνήμη που έχουν πρόσβαση όλα τα threads είναι διαφορετική από την μνήμη του επεξεργαστή.



Εικόνα 15<sup>η</sup> : Grid of Thread Blocks

#### 4.2 Συσκευές που υποστηρίζουν GPUs

Ένας πίνακας των συσκευών που υποστηρίζουν επίσημα CUDA είναι αυτός που αναφέρεται πιο κάτω. Πολλές εφαρμογές απαιτούν τουλάχιστον 256 MB αφιερωμένου VRAM).

|                 |                       |                   |
|-----------------|-----------------------|-------------------|
| Nvidia GeForce  | Nvidia GeForce Mobile | Nvidia Quadro     |
| GeForce GTX 295 | GeForce 9800M GTX     | Quadro FX 5800    |
| GeForce GTX 285 | GeForce 9800M GTS     | Quadro FX 5600    |
| GeForce GTX 280 | GeForce 9800M GT      | Quadro FX 4800    |
| GeForce GTX 275 | GeForce 9700M GTS     | Quadro FX 4700 X2 |
| GeForce GTX 260 | GeForce 9700M GT      | Quadro FX 4600    |
| GeForce GTS 250 | GeForce 9650M GS      | Quadro FX 3700    |

|                    |                   |                           |
|--------------------|-------------------|---------------------------|
| GeForce 9800 GX2   | GeForce 9600M GT  | Quadro FX 1700            |
| GeForce 9800 GTX+  | GeForce 9600M GS  | Quadro FX 570             |
| GeForce 9800 GTX   | GeForce 9500M GS  | Quadro FX 370             |
| GeForce 9800 GT    | GeForce 9500M G   | Quadro NVS 290            |
| GeForce 9600 GSO   | GeForce 9400M G   | Quadro FX 3600M           |
| GeForce 9600 GT    | GeForce 9300M GS  | Quadro FX 1600M           |
| GeForce 9500 GT    | GeForce 9300M G   | Quadro FX 770M            |
| GeForce 9400 GT    | GeForce 9200M GS  | Quadro FX 570M            |
| GeForce 9400 mGPU  | GeForce 9100M G   | Quadro FX 370M            |
| GeForce 9300 mGPU  | GeForce 8800M GTS | Quadro Plex 1000 Model IV |
| GeForce 8800 Ultra | GeForce 8700M GT  | Quadro Plex 1000 Model S4 |
| GeForce 8800 GTX   | GeForce 8600M GT  |                           |
| GeForce 8800 GTS   | GeForce 8600M GS  | Nvidia Quadro Mobile      |
| GeForce 8800 GT    | GeForce 8400M GT  | Quadro NVS 360M           |
| GeForce 8800 GS    | GeForce 8400M GS  | Quadro NVS 140M           |
| GeForce 8600 GTS   | GeForce 8400M G   | Quadro NVS 135M           |
| GeForce 8600 GT    | GeForce 8200M G   | Quadro NVS 130M           |
| GeForce 8600 mGT   |                   |                           |
| GeForce 8500 GT    |                   | Nvidia Tesla              |
| GeForce 8400 GS    |                   | Tesla S1070               |
| GeForce 8300 mGPU  |                   | Tesla C1060               |
| GeForce 8200 mGPU  |                   | Tesla C870                |
| GeForce 8100 mGPU  |                   | Tesla D870                |
|                    |                   | Tesla S870                |

**Πίνακας 2ος : Συσκευές που υποστηρίζουν GPUs**

### 4.3 Πλεονεκτήματα της CUDA

Η CUDA έχει διάφορα πλεονεκτήματα πέρα από τον παραδοσιακό υπολογισμό γενικού σκοπού σε GPUs (GPGPU) που χρησιμοποιεί τα APIs της κάρτας γραφικών

- Scattered reads – ο κώδικας μπορεί να διαβάσει από τις αυθαίρετες διευθύνσεις στη μνήμη.
- Κοινή μνήμη – το CUDA εκθέτει μια γρήγορη κοινή περιοχή μνήμης (16KB στο μέγεθος) που μπορεί να μοιραστεί μεταξύ των threads.
- Μεταφορτώνει γρηγορότερα και επαναδιαβάζει από και προς το GPU.

- Υποστηρίζει πλήρως integer and bitwise operations συμπεριλαμβανομένων texture lookups.

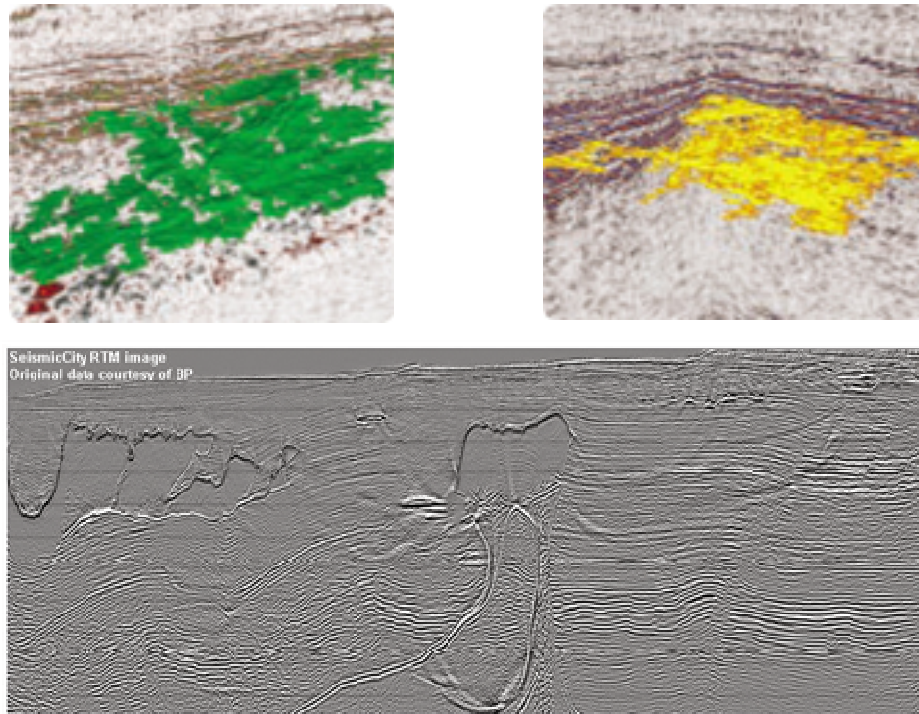
#### 4.4 Περιορισμοί

- Χρησιμοποιεί recursion-free, function-pointer-free υποσύνολο της γλώσσας C, συν μερικές απλές επεκτάσεις. Εντούτοις, μόνο μια διαδικασία μπορεί να τρέξει διασπώντας την εκτέλεσή της σε πολλαπλά διάσπαρτα διαστήματα μνήμης.
- Η απόδοση texture δεν υποστηρίζεται
- Οι επαναλαμβανόμενες λειτουργίες (recursion) δεν υποστηρίζονται και πρέπει να μετατραπούν στους βρόχους (loops).
- Για τη διπλή ακρίβεια δεν υπάρχει καμία απόκλιση από τα IEEE 754 πρότυπα. Στην ενιαία ακρίβεια, Denormals και NaNs δεν υποστηρίζονται.
- Το εύρος μεταξύ της CPU και GPU μπορεί να είναι μια δυσχέρεια.
- Τα threads πρέπει να τρέχουν στις ομάδες τουλάχιστον των 32 για την καλύτερη απόδοση, με το συνολικό αριθμό των threads αριθμώντας στις χιλιάδες. Οι κλάδοι στον κώδικα προγράμματος δεν προσκρούουν στην εκτέλεση σημαντικά, υπό τον όρο ότι κάθε ένα από 32 threads παίρνει την ίδια πορεία εφαρμογής.
- CUDA-GPUs είναι μόνο διαθέσιμα από NVIDIA (GeForce σειρές 8 και ανωτέρω, Quadro και Tesla).

## 4.5 Εφαρμογές της CUDA

### 4.5.1 Εφαρμογές της CUDA στην βιομηχανία πετρελαίου και φυσικού αερίου

Το πετρέλαιο και το φυσικό αέριο καθίστανται όλο και πιο δύσκολο να βρεθούν. Μεγάλες δεξαμενές βρίσκονται τώρα σε μεγαλύτερο βάθος και τα ιζήματα είναι πολύ πιο δύσκολο να αναλυθούν, όπως η ανακάλυψη Jack Field στον Κόλπο του Μεξικού, το οποίο βρέθηκε σε περισσότερο από 20,000 πόδια κάτω από τη θάλασσα. Λόγω της πολυπλοκότητας των στρωμάτων, απαιτείτε καλύτερη επεξεργασία εικόνας, άρα και περισσότερα δεδομένα.



**Εικόνα 16<sup>η</sup> : Εφαρμογή της CUDA για γεωλογικούς σκοπούς**

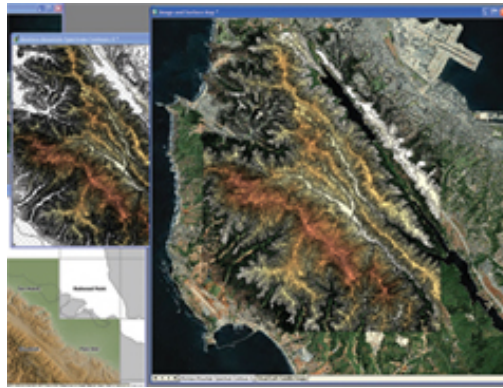
Σε μια εποχή όπου οι περισσότεροι άνθρωποι εξακολουθούν να θεωρηθούν τις GPUs ως μια τεχνολογία μόνο για την αγορά ηλεκτρονικών παιχνιδιών, Houston-based Headwave, μια εταιρεία που ειδικεύεται στην ανάλυση των γεωφυσικών δεδομένων, ξεκίνησε την ανάπτυξη ενός nextgeneration computing platform που θα μπορούσε να αξιοποιήσει την παράλληλη-επεξεργαστική ισχύ των γραφικών καρτών.

Η εφαρμογές της Headwave, που υλοποιήθηκαν σε NVIDIA ® GPU , επιτρέπουν τους γεωφυσικούς να εφαρμόζουν προηγμένα φίλτρα στα δεδομένα τους

με άμεσα αποτελέσματα, ακόμη και σε multi-terabyte dataset. Επιπλέον, οι γεωφυσικοί μπορούν να αναλύσουν τα αρχικά σεισμικά δεδομένα ( «pre-stack”) σε πολλαπλές διαστάσεις, ως μέρος της καθημερινής τους εργασίας.

Η επεξεργασία των multi-terabyte dataset σε πραγματικό χρόνο δεν θα ήταν δυνατή χωρίς τις πρόσφατες εξελίξεις στη NVIDIA GPU computing solutions. Οι εταιρείες πετρελαίου και φυσικού αερίου είναι ήδη εφοδιασμένες με GPUs, πράγμα που σημαίνει ότι μεγάλο μέρος του υλικού υποδομής για την αξιοποίηση αυτής της τεχνολογίας είναι ήδη σε ισχύ.

#### 4.5.2 Εφαρμογές της CUDA στην γεωγραφική πληροφόρηση



**Εικόνα 17<sup>η</sup> : Εφαρμογή της CUDA για γεωγραφικούς σκοπούς**

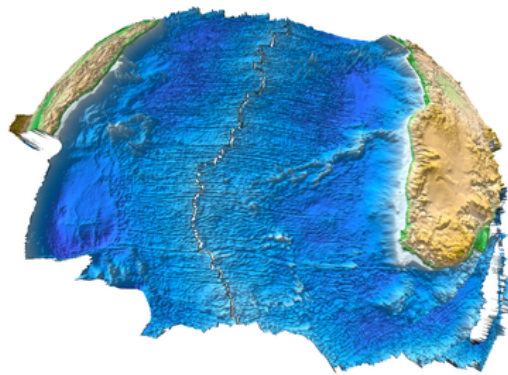
Η τεχνολογία υπηρεσιών γεωγραφικής πληροφόρησης, έχει σημαντικές επιπτώσεις για την ασφάλεια και την ποιότητα της καθημερινής μας ζωής. Πολλές επιχειρήσεις κάνουν χρήση των GIS εφαρμογών για να συνδυάζουν την χαρτογράφηση φυσικών πληροφοριών με τα δημογραφικά δεδομένα, όπως είναι ο πληθυσμός, η ρυμοτομία και οι τοπικοί πόροι.

Καθώς οι χρήστες κάνουν όλο και περισσότερη χρήση των εφαρμογών GIS, οι βάσεις δεδομένων αυξάνονται και γίνονται όλο και πιο πολύπλοκες, με πολλές gigabytes από διαδραστικούς χάρτες που δείχνουν vectors, layers και images. Η Manifold.net που ηγείτο στην ανάπτυξη συστημάτων GIS, προσπαθεί να δημιουργήσει ένα προϊόν το οποίο να είναι ικανό να ανταπεξέρθει στις πολύπλοκες βάσεις δεδομένων. Η εταιρεία κατάλαβε ότι μία CPU-based computing solution δεν είναι πλέον κατάλληλη για την κάλυψη των αναγκών των χρηστών.

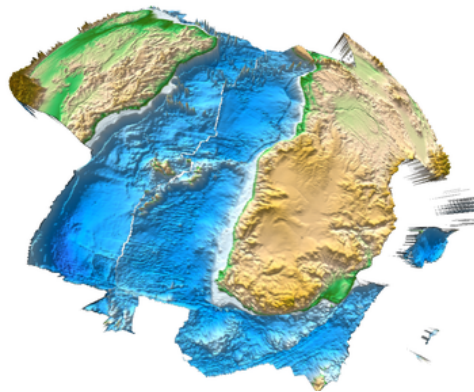
Η Manifold μετέτρεψε το λογισμικό του στην πλατφόρμα NVIDIA® CUDA™, επιτρέποντας στους χρήστες να έχουν πρόσβαση στην παράλληλη επεξεργαστική ισχύ. Με τις ρυθμίσεις της CUDA, οι υπολογισμοί που προηγουμένως χρειαζόνταν 20 λεπτά για να ολοκληρωθούν τώρα ολοκληρώνονται σε 30 δευτερόλεπτα. Επιπλέον, οι υπολογισμοί που προηγουμένως ήθελαν 30 με 40 δευτερόλεπτα τώρα πια ολοκληρώνονται πλέον σε πραγματικό χρόνο.

Με τη νέα επιτάχυνση της CUDA, η Manifold βοηθά στην εξόρυξη εφεδρικών καυσίμων και στην παρακολούθηση της προόδου των ρύπων στον αέρα.

Ένα άλλο παράδειγμα χρήσης καρτών γραφικών είναι αυτό του “Simula Research Laboratory” το οποίο έχει αναπτύξει ένα λογισμικό πρόγραμμα το οποίο τρέχει σε GPU και αναλύει την γεωλογική περιεκτικότητα της γης ανά στρώματα και πώς αυτό έχει αλλάξει ή θα αλλάξει με το χρόνο.

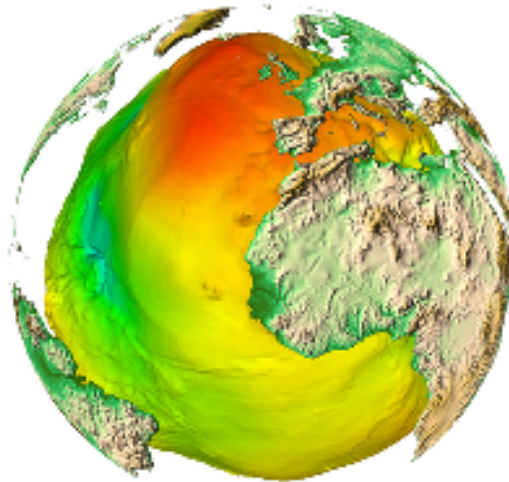


**Εικόνα 18<sup>η</sup> : Παρουσίαση της επιφάνειας της θάλασσας στην Νότια Αφρική  
όπως είναι σήμερα**



**Εικόνα 19<sup>η</sup> : Παρουσίαση της επιφάνειας της θάλασσας στην Νότια Αφρική  
όπως ήταν 100 εκατομμύρια πριν όταν η Αφρική ήταν πιο κοντά με  
την Νότια Αφρική.**

Η παρουσίαση αυτή έχει γίνει σε 4D. Με την χρήση των state-of-the art αλγόριθμων οι γεωλόγοι μπορούν να έχουν εικόνα της επιφάνειας της γης και να αναλύσουν τα δεδομένα που λαμβάνουν από την χρήση των διάφορων μοντέλων. Η γης είναι χωρισμένη σε τεκτονικές πλάκες τα οποία κινούνται με πολύ αργό ρυθμό όμως προκαλούν πολλά γεωλογικά φαινόμενα. Η εφαρμογή των διαφόρων μοντέλων σε GPU δίνει την δυνατότητα στους γεωλόγους να έχουν μια εικόνα και να αναλύουν πιο αποτελεσματικά την κίνηση των πλακών σε βάθος χρόνου.



**Εικόνα 20<sup>η</sup> : Παρουσίαση της επιφάνειας της γης με την χρήση γραφικών καρτών**

#### **4.5.3 Εφαρμογές της CUDA στην ιατροφαρμακευτική βιομηχανία**



**Εικόνα 21<sup>η</sup> : Εφαρμογή της CUDA στη ιατροφαρμακευτική βιομηχανία**

Όλοι οι άνθρωποι είναι συνηθισμένοι στα αυστηρά χρονοδιαγράμματα στις σύγχρονες ιατρικές υπηρεσίες, όπου για μισή ώρα επίσκεψη μπορεί να καθυστερήσει και όλη την ημέρα. Έτσι, μια νέα τεχνολογία για να είναι πρακτική σε αυτό το

περιβάλλον, πρέπει να εργαστεί σύμφωνα με την σημερινή ταχύτητα των γρήγορων ρυθμών ιατρικής. Οι ασθενείς θέλουν μια γρήγορη, άνετη και επακριβή διάγνωση και οι γιατροί θέλουν ένα διαγνωστικό εργαλείο που να είναι αποτελεσματικό. Στο Techniscan Medical System ανέπτυξαν ένα καινούριο σύστημα απεικόνισης για αυτές τις εξετάσεις, Whole Breast Ultrasound (WBU™) system.

Στα πρώτα στάδια της ανάπτυξης, ωστόσο, τα αποτελέσματα χρειάζονταν πολλούς υπολογιστές να συνδέονται μαζί και έρχονταν με μεγάλη καθυστέρηση. Για να επιλύσει το πρόβλημα αυτό η Techniscan χρησιμοποίησε NVIDIA ® CUDA ™ αντί για MPI και FORTRAN. Το σύστημα έτρεχε πιο γρήγορα χωρίς να επηρεάζεται η ποιότητα των αποτελεσμάτων.

#### 4.5.4 Εφαρμογές της CUDA στην βιομηχανία ενδυμάτων



**Εικόνα 22<sup>η</sup> : Εφαρμογή της CUDA στη βιομηχανία ρούχων**

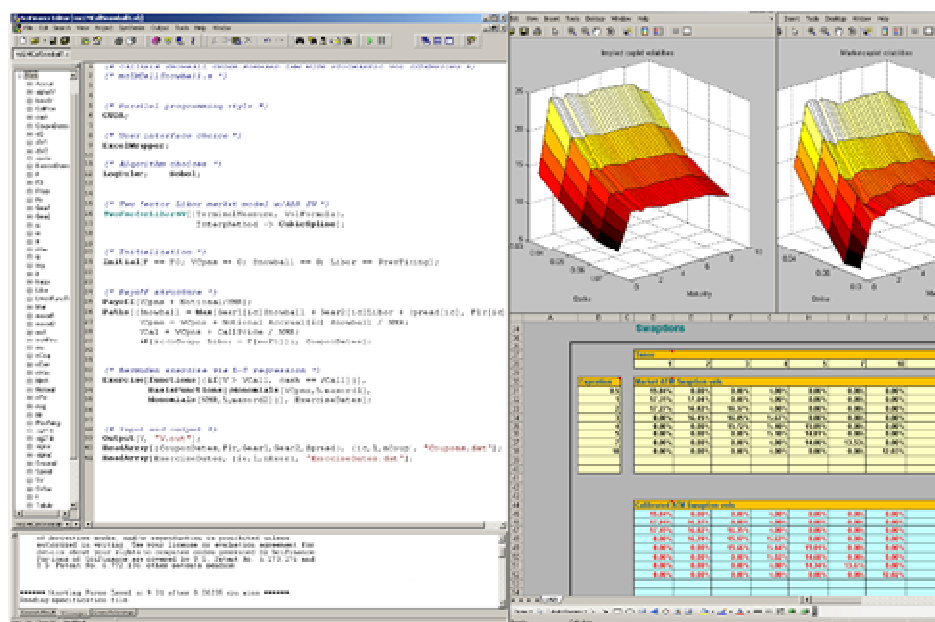
Η OptiTex Ltd, μία εταιρεία από Ισραήλ, έχει αναλάβει το σχεδιασμό της βιομηχανίας σε ένα εντελώς νέο επίπεδο με την 3D CAD / CAM τεχνολογία.

Παραδοσιακά, οι σχεδιαστές πρέπει να δημιουργήσουν δείγματα των πραγματικών γραμμών ρούχων για να τα παρουσιάσουν στους δυνητικούς επενδυτές. Η διαδικασία αυτή είναι εξαιρετικά χρονοβόρα και δαπανηρή. Η OptiTex 3D εκσυγχρονίζει αυτή τη διαδικασία, επιτρέποντας στους σχεδιαστές να προσομοιώνουν την εμφάνιση και την κίνηση των ειδών ένδυσης με εικονικά μοντέλα, να επανεξετάσουν, να βελτιώσουν και να κάνουν καλύτερη μέτρηση των δειγμάτων πριν αρχίσουν να κόβουν τα υφάσματα.

Για την αντιμετώπιση αυτής της πρόκλησης, η OptiTex χρησιμοποιεί την NVIDIA® CUDA™ για την προσομοίωση της μηχανής δεδομένων και αλγορίθμων για να τρέχει τα σχέδια σε GPUs. Η GPU επέτρεψε δέκα φορές καλύτερη επίδοση από την CPU. Η ανάπτυξη του προϊόντος για μια εποχιακή συλλογή είναι συνήθως 190 ημέρες, αλλά με την ανασχηματισμένες OptiTex 3D λύσεις, ο χρόνος έχει μειωθεί σε μόλις 35 ημέρες.

Ως αποτέλεσμα των επιδόσεων σε πραγματικό χρόνο με GPU, οι σχεδιαστές έχουν τη δυνατότητα να μειώσουν το κόστος παραγωγής και να συντομεύσουν τον κύκλο σχεδιασμού του προϊόντος. Οι χρήστες σε μια σειρά βιομηχανιών, συμπεριλαμβανομένων των ειδών ένδυσης, την αυτοκινητοβιομηχανία και την αεροναυπηγική μπορούν να εξαλείψουν έξοδα για υλικά, για ηλεκτρική ενέργεια και για ώρες εργασίας.

#### 4.5.5 Εφαρμογές της CUDA στη Χρηματοοικονομική

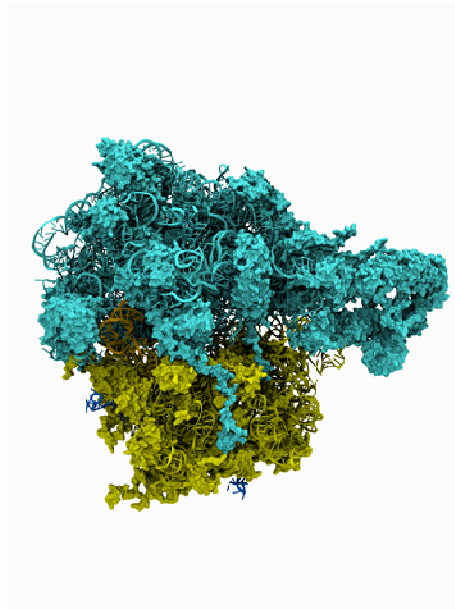


**Εικόνα 23<sup>η</sup> : Εφαρμογή της CUDA στη Χρηματοοικονομική**

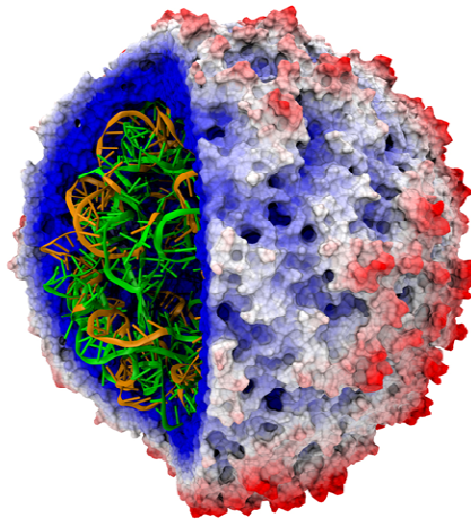
Η εταιρεία Sci Comp., Inc. έχει δημιουργήσει το πρόγραμμα SciFinance το οποίο τρέχει μοντέλα για την αξιολογήσει τιμών όπως τα derivatives σε κάρτα γραφικών. Η οπτικοποίηση των δεδομένων καθώς και η παράλληλη επεξεργασία τους, δίνει την δυνατότητα στους χρήστες να έχουν άμεσα αποτελέσματα και να τα κατανοήσουν πιο σύντομα. Οι νέες δυνατότητες που προσφέρει η CUDA ανοίγουν νέους δρόμους στη

Χρηματοοικονομική αλλά και στα μαθηματικά. Η βελτίωση στον τρόπο επεξεργασίας των δεδομένων βελτιώνει την λήψη αποφάσεων και επομένως την απόδοση.

#### 4.5.6 Εφαρμογές της CUDA στη Ατομική Δυναμική



Εικόνα 24<sup>η</sup> : Εφαρμογή της CUDA στη Ατομική Δυναμική (1)



Εικόνα 25<sup>η</sup> : Εφαρμογή της CUDA στη Ατομική Δυναμική (2)

Nanoscale Molecular Dynamics (NAMD) και Visual Molecular Dynamics (VMD), είναι δύο εφαρμογές που χρησιμοποιούνται ευρέως από επιστήμονες για έρευνες για προσομοίωση και οπτικοποίηση των ατόμων.

## Κεφάλαιο 5

### Σχετικές μελέτες

#### 5.1 «Άρθρο 1» *GPU Acceleration of Numerical Weather Prediction*

Στο άρθρο GPU Acceleration of Numerical Weather Prediction γίνεται περιγραφή παρόμοιας εργασίας που είχε γίνει από τους John Michalakes και Manish Vachharajani. Ο πρώτος εργάζεται στο National Center for Atmospheric Research παράρτημα Boulder Colorado και ο δεύτερος εργάζεται στο University of Colorado at Boulder. Οι συγγραφείς υποστηρίζουν ότι πλέον για να έχουμε πιο σίγουρη και έγκυρη πρόβλεψη καιρού δεν φτάνει μόνο η ταχύτητα των επεξεργαστών. Πρέπει ταυτόχρονα να εκσυγχρονίσουμε και τα μοντέλα πρόβλεψης. Οι συγγραφείς λοιπόν, χρησιμοποίησαν CUDA για να αλλάξουν ένα συγκεκριμένο κομμάτι από το μοντέλο WRF. Κατάφεραν να μειώσουν τον χρόνο εκτέλεσης του κομματιού αυτού κατά 20 φορές και τον χρόνο εκτέλεσης όλου του μοντέλου κατά 1.3 φορές για κάποιο συγκεκριμένο ιδεατό σενάριο.

Το κομμάτι που αποφάσισαν να αλλάξουν είναι το WRF Single Moment 5-trace (WSM5). Το WSM5 υπολογίζει την αλλαγή κατάστασης της ατμόσφαιρας την πιθανότητα αυτή η νέα κατάσταση να προκαλέσει πτώση βροχής χιονιού, χαλαζιού ή άλλων παρόμοιων φαινομένων. Ανήκει στο model layer που αναφέρεται στο δεύτερο κεφάλαιο. Όπως επίσης αναφέρεται η περιοχή που μελετάμε χωρίζεται σε domains. Το κάθε domain χωρίζεται σε tiles. Το κάθε tile έχει μια τρίτη διάσταση για να αναπαριστά το υψόμετρο. Κάθε tile μπορεί να διαφέρει από τα άλλα ως προς αυτήν την τρίτη διάσταση. Για κάθε tile λοιπόν το WSM5 κάνει κατά μέσο όρο 2400 floating point πράξεις ισοδύναμες του πολλαπλασιασμού σε κάθε πέρασμα. Για να δούμε τελικά αποτέλεσμα γίνονται πολλές επαναλήψεις έτσι οι πράξεις που γίνονται συνολικά είναι πάρα πολλές. Το κάθε CUDA thread κάνει τους υπολογισμούς του WSM5 για 1 tile.

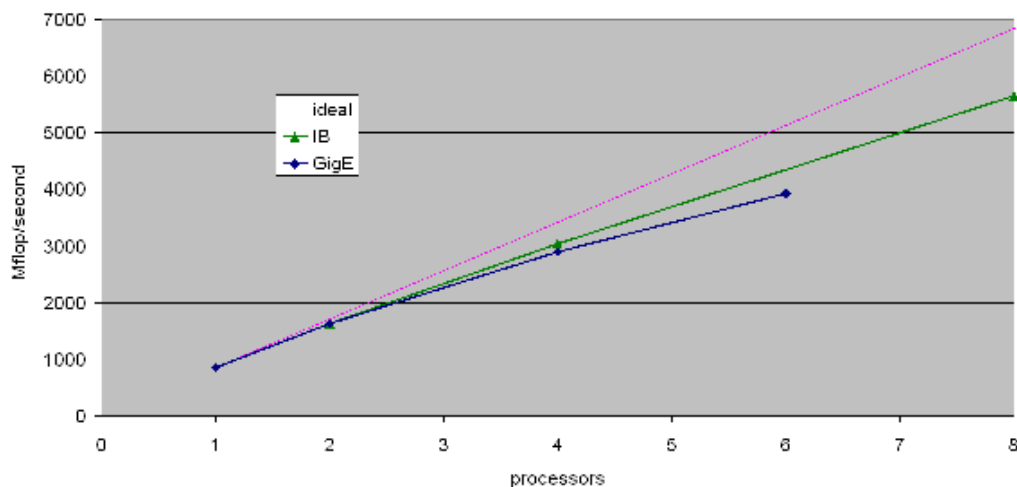
Μεγάλη προσοχή έδειξαν οι συγγραφείς στο πως θα μοιράσουν τον φόρτο εργασίας στα threads και στο ποια πράγματα πρέπει να φυλάσσονται στην περιορισμένη μνήμη του block και ποια πρέπει να πάνε στην καθολική μνήμη. Επίσης χρησιμοποίησαν κάποιον preprocessor για να αναγνωρίζουν το είδος του πίνακα και να γίνεται αυτόματα η ανάθεση σε threads και η ανάθεση στην μνήμη. Για να γίνεται το compile πιο εύκολα απομόνωσαν το κομμάτι αυτό του WRF και στην συνέχεια έδιναν κάποιες έτοιμες εισόδους και σύγκριναν την έξοδο με το τη έβγαζε το ολοκληρωμένο μοντέλο.

Όπως ανάφερα και στην αρχή τα αποτελέσματα τους ήταν αρκετά θεαματικά. Κατάφεραν speed up 17-20 για το κομμάτι αυτό που μεταφράζεται όπως αναφέρουν σε 1.2-1.3 για ολοκληρωμένο το μοντέλο. Κατά την γνώμη μου έχουν δίκαιο στο τέλος όταν αναφέρουν ότι ίσως στο μέλλον τέτοιου είδους προγράμματα να τρέχουν μόνο σε κάρτες γραφικών. Διαφωνώ όταν λένε ότι η δουλειά τους ήταν σύντομη και ίσως και όχι όσο προσεγμένη έπρεπε. Διαφωνώ γιατί έγραψαν preprocessor μελέτησαν την δομή της κάρτας γραφικών που θα χρησιμοποιούσαν, ήξεραν ήδη πολύ καλά την υλοποίηση του κομματιού που αποφάσισαν να αλλάξουν. Ίσως τα λένε αυτά για να δώσουν έμφαση στο ότι υπάρχει χώρος για παραπάνω μελέτη και ίσως για να καλύψουν την έλλειψη ,κατά την γνώμη μου, λεπτομερής περιγραφής της διαδικασίας που ακλούθησαν. Παρόλα αυτά είναι ένα καλό άρθρο που μου έδωσε κατευθυντήριες γραμμές για την δική μου εργασία. Δεν ακολούθησα όμως στη δική μου εργασία κατά γράμμα αυτά που έκαναν οι συγγραφείς λόγο έλλειψης χρόνου και λόγο έλλειψης πείρας στον μετεωρολογικό τομέα.

## **5.2 «Άρθρο 2» *Weather Research and Forecast (WRF) Model Port to Window***

Το NCAR η Microsoft Corporation, η Advances Micro Devices, Inc. και το Portland Group, Inc. συνεργάστηκαν για την δημιουργία μιας πρωτότυπης έκδοσης του WRF. Η προσπάθεια αυτή ήταν να μπορεί το WRF να τρέξει παράλληλα χρησιμοποιώντας MPI – Message Passing Interface σε επεξεργαστή AMD Opteron TM που τρέχει πάνω σε Windows Compute Cluster Server 2003. Το WRF υποστηρίζεται από πολλά συστήματα υψηλής απόδοσης που το καθένα από αυτά τρέχει κάποια έκδοση του λειτουργικού συστήματος UNIX. Τα Windows είναι ένα λειτουργικό σύστημα που παρουσιάζει εξαιρετικές ευκαιρίες για τους χρήστες του WRF.

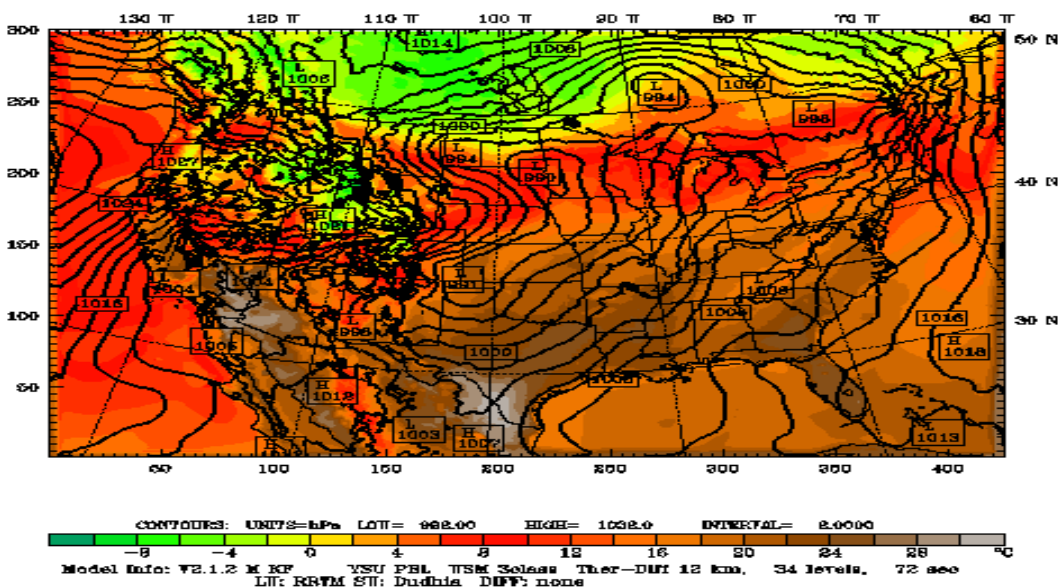
Η μελέτη για την εισαγωγή του WRF σε ένα σύμπλεγμα συστημάτων υψηλής απόδοσης της Microsoft άρχισε μετά το συνέδριο για την σουπερυπολογιστική το 2005. Το WRF έτρεξε με επιτυχία σε παράλληλη εφαρμογή MPI σε επεξεργαστή AMD Opteron TM που τρέχει πάνω σε Windows Compute Cluster Server 2003. Στην παρακάτω εικόνα φαίνεται η απόδοση του μοντέλου WRF σε MFlops/second από έναν επεξεργαστή έως οχτώ επεξεργαστές (IB line) . Παρατηρούμε ότι όσο αυξάνεται ο αριθμός των επεξεργαστών αυξάνεται και η απόδοση του μοντέλου. Η παράλληλη εφαρμογή του μοντέλου αυξάνει επιπλέον κατά 82.6% την απόδοση του μοντέλου (Ideal line). Η γραμμή (GigE) παρουσιάζει την απόδοση του μοντέλου χρησιμοποιώντας Gigabit Ethernet Network. Μπορούμε να δούμε ότι η απόδοση του μοντέλου είναι πολύ πιο χαμηλή. Η ταχύτητα δικτύου επηρεάζει ριζικά την παράλληλη εφαρμογή του μοντέλου.



**Εικόνα 27<sup>η</sup> : Απόδοση του Windows Compute Cluster Server 2003 σε WRF 12km CONUS standard benchmark case.**

Η παρακάτω εικόνα παρουσιάζει πως το μοντέλο WRF τρέχει σε Windows βασισμένο σε στατιστικά στοιχεία που χρησιμοποιήθηκαν ως μέτρο σύγκρισης για την αξιολόγηση μετεωρολογικών στοιχείων.

Dataset: CONUS RIP: john Valid: 0300 UTC Wed 24 Oct 01 Init: 0000 UTC Wed 24 Oct 01  
 Fcst: 3.00 h at k-index = 34  
 Temperature  
 Sea-level pressure



**Εικόνα 28<sup>η</sup>: Αποτέλεσμα ενός μοντέλου WRF που τρέχει σε Windows Compute Cluster Server 2003 σε WRF 12km CONUS standard benchmark case**

Για να μπορεί το μοντέλο να τρέχει σε Windows χρειάστηκαν κάποιες αλλαγές όπως στις καταλήξεις των αρχείων π.χ. η αλλαγή από .f σε .f90 καθώς και κάποιες τροποποιήσεις στα WRF Makefiles λόγω των αλλαγών στις καταλήξεις. Επίσης άλλαξαν κάποια commands και libraries. Με ελάχιστες αλλαγές στο τρόπο που κτίστηκε το WRF έγινε κατορθωτό το μοντέλο να τρέξει σε Windows Compute Cluster Server 2003.

## Κεφάλαιο 6

### Εκτέλεση του μοντέλου WRF με ενσωματωμένο κώδικα CUDA

#### 6.1 Διαδικασία Εκτέλεσης

Η αρχική σκέψη για τον τρόπο εργασίας μου ήταν να ακολουθήσω την πορεία που είχα μάθει στο μάθημα του παραλληλισμού « Παράλληλος Προγραμματισμός, Αρχιτεκτονικές και Γλώσσες». Η διαδικασία αυτή συμπεριλαμβάνει τα ακόλουθα βήματα:

- Μέτρηση του χρόνου εκτέλεσης του μοντέλου WRF,
- Έλεγχος του μοντέλου για να εντοπίσω πιο τμήμα του κώδικα παίρνει τον περισσότερο χρόνο να εκτελεστεί,
- Αλλαγή ή αντικατάσταση του τμήματος αυτού χρησιμοποιώντας κώδικα CUDA,
- Έλεγχος των αποτελεσμάτων και των χρόνων εκτελέσεων. .

Η ακολουθία αυτής της πορείας ήταν πολύ δύσκολη λόγω του ότι δεν γνώρισα σε βάθος το λειτουργικό σύστημα Linux το οποίο ήταν απαραίτητο για να εκτελέσω το μοντέλο, πράγμα που με δυσκόλεψε πολύ στην εγκατάσταση των υπολοίπων εργαλείων που χρειάζονται για την εργασία. Εδώ πρέπει να πω ότι το κύριο πρόβλημα σε όλη την πορεία της εργασίας ήταν ακριβώς αυτό γιατί με δυσκόλεψε όχι μόνο στην εγκατάσταση αλλά και στην χρήση των εργαλείων. Το λειτουργικό σύστημα Linux δεν είναι ενοποιημένο και σταθερό από κατασκευαστή σε κατασκευαστή. Ο κάθε κατασκευαστής μιας έκδοσης του Linux βάζει τις δικές του εντολές και τον δικό του τρόπο λειτουργίας του περιβάλλοντος του λειτουργικού συστήματος, έτσι ακόμα και αν έβρισκα κάποια λύση στα προβλήματα που παρουσιάζονταν συνήθως δεν μπορούσαν να εφαρμοστούν στη δική μου έκδοση του Linux. Η έκδοση του Linux που χρησιμοποίησα σαίτα την εργασία είναι το Fedora8.

Στην αρχή εγκατέστησα το CUDA driver για Linux και το μοντέλο WRF. Βεβαιώθηκα ότι ξεχωριστά τα δυο αυτά εργαλεία δούλευαν ικανοποιητικά. Εκτέλεσα με επιτυχία τα test cases που ήταν μέσα στο πακέτο του WRF και τις εφαρμογές που ήταν μέσα στο Software Development Kit της CUDA. Στην συνέχεια προσπάθησα να βρω ένα καλό εργαλείο για μέτρηση χρόνων εκτέλεσης. Βρήκα το CodeAnalyst της AMD το οποίο ήταν κατάλληλο για τον επεξεργαστή που χρησιμοποιώ ο υπολογιστής μου, ο οποίος είναι AMD Athlon 64.3200+.

Επόμενο βήμα ήταν να βρω τον τρόπο να μεταφέρω την εκτέλεση από τον επεξεργαστή του υπολογιστή στην κάρτα γραφικών. Κατά την αναζήτηση μεθόδων αλλαγής κώδικα Fortran σε κώδικα CUDA βρήκα την ιστοσελίδα όπου δημοσιεύονται οι εργασίες που έχουν γίνει σχετικά με WRF και CUDA. Εκεί υπάρχουν τρεις εργασίες που έχουν γίνει ήδη και που παρέχουν τον πηγαίο κώδικα.. Στην ίδια σελίδα υπάρχει και ο πηγαίος κώδικας για την εργασία που δημοσιεύτηκε στο άρθρο «*GPU Acceleration of Numerical Weather Prediction*» καθώς και οδηγίες για το πώς μπορείς να το ενσωματώσεις αυτόν τον κώδικα στο μοντέλο. Ακολουθώντας τις οδηγίες αυτές παρατήρησα ότι ο πηγαίος κώδικας αυτής της εργασίας παρέχεται και με το πακέτο 3.0.1 αλλά και με το επόμενο πακέτο (3.1.0) του WRF. Ο κώδικας για το συγκεκριμένο άρθρο είναι γραμμένος από ανθρώπους που δουλεύουν στο NCAR. Γράφτηκε όταν το ίδρυμα αποφάσισε να δημιουργήσει ένα πειραματικό κομμάτι κώδικα για το WRF το οποίο θα εκτελούνταν σε GPU με την χρήση CUDA. Το θεώρησα ένα καλό παράδειγμα για το πώς μπορείς να αλλάξεις κώδικα του WRF σε CUDA και αρχικός σκοπός ήταν να το χρησιμοποιήσω σαν βοήθημα για να γράψω δικό μου κώδικα.

Ο συγκεκριμένος κώδικας υλοποιεί το υπολογιστικό μοντέλο WSM5. Γι' αυτό και πρώτα έλεγξα κατά πόσο επηρεάζει η χρήση του υπολογιστικού μοντέλου αυτού την εκτέλεση του WRF. Άλλαξα τα input files των test cases του WRF πακέτου έτσι ώστε να χρησιμοποιούν WSM5 στους υπολογισμούς τους αλλά και να μην διαρκούν υπερβολική ώρα. Το θέμα της διάρκειας επηρέαζε κυρίως το CodeAnalyst. Τις περισσότερες περιπτώσεις αν μια εφαρμογή κρατούσε παραπάνω από μια ώρα να τελειώσει, το CodeAnalyst διέκοπτε απότομα την διαδικασία μέτρησης με αποτέλεσμα να μην παίρνω τελικά σωστούς χρόνους εκτέλεσης. Έτσι έκανα τους χρόνους εκτέλεσης της κάθε περίπτωσης να είναι μεταξύ 10 και 20 λεπτά. Η αλλαγή αυτή από μερικές ώρες χρόνο εκτέλεσης σε είκοσι λεπτά χρόνο εκτέλεσης πιστεύω δεν επηρεάζει την ποιότητα των μετρήσεων. Αυτό που άλλαξα ήταν η διάρκεια του χρόνου που θα εκτελείτε η προσομοίωση κάτι που είναι μια από τις παραμέτρους εισόδου του μοντέλου (π.χ. αντί για 1000 'μέρες' εφαρμογής, η προσομοίωση να γίνει για 10 'μέρες'). Η διάρκεια της προσομοίωσης δεν αφορά βεβαίως πραγματικό χρόνο. Τα αποτελέσματα της προσομοίωσης για 10 'μέρες' εμφανίζονταν είκοσι λεπτά αργότερα. Έτσι μέτρησα χρόνους εκτέλεσης και είδα τις τιμές που παρουσιάζονται πιο κάτω.

Αποφάσισα τελικά να χρησιμοποιήσω τον έτοιμο κώδικα για το WSM5 που είχε στην σελίδα και να μην γράψω δικό μου κώδικα για δυο λόγους. Ο πρώτος λόγος είναι ότι κατά την διαδικασία ανάλυσης του Fortran κώδικα έτσι ώστε να τον αλλάξω σε CUDA παρατήρησα ότι λόγο του όγκου και της περιπλοκότητας της εφαρμογής θα έπρεπε τελικά να

εξαρτηθώ κατά μεγάλο βαθμό από τον έτοιμο κώδικα που θεωρούσα έως εκείνο το σημείο ως απλό βοήθημα. Επιπλέον τα χρονικά περιθώρια που είχα δεν μου επέτρεπαν να ασχοληθώ σε όσο βάθος χρειαζόταν έτσι ώστε να επιτύχω τον στόχο της μετατροπής μέρους του WRF σε CUDA χωρίς να αντιγράψω ουσιαστικά τον κώδικα που είχα ήδη έτοιμο. Ο δεύτερος λόγος είναι ότι ο κώδικας αυτός είναι γραμμένος από ανθρώπους που εργάζονται στην NCAR για την συντήρηση του κώδικα του μοντέλου, και συνεπώς ξέρουν καλύτερα την δομή του και το πώς πρέπει να αλλάξει σε κώδικα CUDA και πώς τελικά να ενσωματωθεί στο μοντέλο. Σε συνδυασμό με το γεγονός ότι όταν το WSM5 μετέφερε την εκτέλεση του στην κάρτα γραφικών έβλεπα σημαντικές διαφορές στο χρόνο εκτέλεσης θα αναδεικνύονταν καλύτερα η χρησιμότητα της κάρτας γραφικών στην εκτέλεση των υπολογισμών που είναι δηλαδή ο δεύτερος στόχος της εργασίας.

## 6.2 Περιγραφή του WSM5

Το WSM5 είναι ένα υπολογιστικό μοντέλο μικροφυσικής για το WRF που υπολογίζει την αλλαγή κατάστασης της ατμόσφαιρας, την πιθανότητα αυτή η νέα κατάσταση να προκαλέσει πτώση βροχής χιονιού, χαλαζιού ή και άλλων παρόμοιων φαινομένων. Ανήκει στο model layer που αναφέρεται στο δεύτερο κεφάλαιο. Όπως επίσης αναφέρεται στο άρθρο η περιοχή που μελετάμε χωρίζεται σε domains. Το κάθε domain χωρίζεται σε tiles. Το κάθε tile έχει μια τρίτη διάσταση για να αναπαριστά το υψόμετρο. Κάθε tile μπορεί να διαφέρει από τα άλλα ως προς αυτήν την τρίτη διάσταση. Για κάθε tile λοιπόν το WSM5 κάνει κατά μέσο όρο 2400 floating point πράξεις ισοδύναμες του πολλαπλασιασμού σε κάθε πέρασμα. Για να δούμε τελικά αποτέλεσμα γίνονται πολλές επαναλήψεις έτσι οι πράξεις που γίνονται συνολικά είναι πάρα πολλές. Το κάθε CUDA thread κάνει τους υπολογισμούς του WSM5 για 1 tile. Όταν χρησιμοποιείται αυτό το υπολογιστικό μοντέλο έχουμε καλύτερα αποτελέσματα στις προσομοιώσεις και στις προβλέψεις από το WRF.

Σαν κώδικας το WSM5 είναι 1500 γραμμές κώδικα Fortran90. Αυτό είναι μόλις το 0.4 του συνολικού κώδικα του WRF. Στο άρθρο «GPU Acceleration of Numerical Weather Prediction» αναφέρεται ότι για να μετατραπεί ο κώδικας από Fortran90 σε CUDA ξαναγράφηκε από την αρχή σε C αφού η CUDA είναι βασισμένη στην C. Πιο κάτω βλέπουμε πως είναι ο κώδικας Fortran για το κομμάτι του WSM5 που υπολογίζει την πτώση κρυστάλλων πάγου. Αυτό είναι ένα μικρό κομμάτι του WSM5.

```

mstepmax = 1
mstep = 1
numdt = 1
do k = kte, kts, -1
  do i = its, ite
    if(qci(i,k,2).le.0.) then
      work2c(i,k) = 0.
    else
      xmi = den(i,k)*qci(i,k,2)/xni(i,k)
      diameter = min(dicon * sqrt(xmi),dimax)
      diameter = max(min(dicon * sqrt(xmi),dimax), 1.e-25)
      work1c(i,k) = 1.49e4*exp(log(diameter)*(1.31))
      work2c(i,k) = work1c(i,k)/delz(i,k)
    endif
    numdt(i) = max(nint(work2c(i,k)*dtcld+.5),1)
    if(numdt(i).ge.mstep(i)) mstep(i) = numdt(i)
  enddo
enddo
do i = its, ite
  if(mstepmax.le.mstep(i)) mstepmax = mstep(i)
enddo
!
do n = 1, mstepmax
  k = kte
  do i = its, ite
    if(n.le.mstep(i)) then
      falkc(i,k) = den(i,k)*qci(i,k,2)*work2c(i,k)/mstep(i)
      fallc(i,k) = fallc(i,k)+falkc(i,k)
      qci(i,k,2) = max(qci(i,k,2)-falkc(i,k)*dtcld/den(i,k),0.)
    endif
  enddo
  do k = kte-1, kts, -1
    do i = its, ite
      if(n.le.mstep(i)) then
        falkc(i,k) = den(i,k)*qci(i,k,2)*work2c(i,k)/mstep(i)
        fallc(i,k) = fallc(i,k)+falkc(i,k)
        qci(i,k,2) = max(qci(i,k,2)-(falkc(i,k)-falkc(i,k+1)
          *delz(i,k+1)/delz(i,k))*dtcld/den(i,k),0.) &
      endif
    enddo
  enddo
enddo
enddo

```

Πιο κάτω βλέπουμε πως είναι ο κώδικας CUDA για το κομμάτι του WSM5 που υπολογίζει την πτώση κρυστάλλων πάγου. Αυτό είναι ένα αντιπροσωπευτικό παράδειγμα για το πώς έχει αλλάξει ο κώδικας Fortran σε CUDA. Βλέπουμε ότι έχει γραφτεί ολόκληρο σε C από την αρχή.

```

mstep = 1 ;
numdt = 1 ;
for ( k = kpe-1 ; k >= kps-1 ; k-- ) {
  if (qi[k] <= 0.) {
    w2[k] = 0. ;
  } else {
    xmi = den[k]*qi[k]/xni[k] ;
    diameter = MAX(MIN(dicon * sqrt(xmi),dimax), 1.e-25) ;
    w1[k] = 1.49e4*exp(log(diameter)*(1.31)) ;
    w2[k] = w1[k]/delz[k] ;
  }
  numdt = MAX( (int) trunc(w2[k]*dtcld+.5+.5),1) ;
  if(numdt > mstep) mstep = numdt ;
}

```

```

}
rmstep = 1./mstep ;

float falkc_k, falkc_kp1, fallc_k, fallc_kp1 ;
for ( n = 1 ; n <= mstep ; n++ ) {
    k = kpe - 1 ;
    den_k = den[k] ;
    falkc_kp1 = den_k*qi[k]*w2[k]*rmstep ;
    fallc_kp1 = fallc_kp1+falkc_kp1 ;
    qi[k] = MAX(qi[k]-falkc_kp1*dtcld/den_k,0.) ;
    delz_kp1 = delz[k] ;
    for ( k = kpe-2 ; k >= kps-1 ; k-- ) {
        den_k = den[k] ;
        falkc_k = den_k*qi[k]*w2[k]*rmstep ;
        fallc_k = fallc_k+falkc_k ;
        delz_k = delz[k] ;
        qi[k] = MAX(qi[k]-(falkc_k-falkc_kp1
            *delz_kp1/delz_k)*dtcld/den_k,0.) ;
        delz_kp1 = delz_k ;
        falkc_kp1 = falkc_k ;
        fallc_kp1 = fallc_k ;
    }
}

float fallsum = fall1_k+fall2_k+fallc_k ;
float fallsum_qsi = fall2_k+fallc_k ;

rainncv = 0. ;
if(fallsum > 0.) {
    rainncv = fallsum*delz[1]/denr*dtcld*1000. ;
    rain = fallsum*delz[1]/denr*dtcld*1000. + rain ;
}

snowncv = 0. ;
if(fallsum_qsi > 0.) {
    snowncv = fallsum_qsi*delz[0]/denr*dtcld*1000. ;
    snow = fallsum_qsi*delz[0]/denr*dtcld*1000. + snow ;
}

sr = 0. ;
if ( fallsum > 0. ) sr =
fallsum_qsi*delz[0]/denr*dtcld*1000./(rainncv+1.e-12) ;

```

Ολόκληρος ο κώδικας του WSM5 τοποθετήθηκε μέσα σε ένα CUDA kernel. Έτσι ολόκληρη η εκτέλεση του κώδικα γίνεται συνέχεια μέσα στην κάρτα γραφικών. Κατά την μετατροπή χρειάστηκε καταρχάς αλλαγή στον τρόπο που φυλάσσονται τα δεδομένα από το WSM5 κώδικα λόγω της διαφορετικότητας της μνήμης της κάρτας γραφικών. Στην έκδοση για Fortran90 χρησιμοποιούνται πολυδιάστατοι πίνακες δηλωμένοι ως global ενώ στον κώδικα CUDA τους άλλαξαν σε μονοδιάστατους πίνακες που είναι δηλωμένοι τοπικά για να μπορούν να μοιράζονται πιο εύκολα στους διάφορους επεξεργαστές. Αυτή η μετατροπή δημιούργησε την ανάγκη για επιπλέον έλεγχο της πρόσβασης στις θέσεις των πινάκων αυτών. Επιπλέον πρόκληση που δημιουργήθηκε ήταν το ότι έπρεπε να δηλώνονται και να χειρίζονται διαφορετικά οι πίνακες που έρχονται σαν παράμετροι από τους πίνακες του WSM5 που υπάρχουν μέσα στον κώδικα. Ο λόγος είναι για βελτιστοποίηση διαχείρισης μνήμης. Μερικοί από αυτούς τους πίνακες θα ήταν καλύτερα να βρίσκονται μόνιμα στο device memory και μερικοί να αποθηκεύονταν σε thread shared memory της κάρτας

γραφικών. Ο διαχωρισμός γίνεται με βάση τις προσβάσεις που γίνονται σε κάθε πίνακα κατά την διάρκεια της εκτέλεσης. Δόθηκε μεγάλη προσοχή ως προς το ποιοι πίνακες θα αποθηκεύονταν σε thread shared memory καθώς δεν υπάρχει μεγάλος χώρος διαθέσιμος για φύλαξη δεδομένων. Οι πίνακες που τελικά τοποθετήθηκαν στην thread shared memory ήταν οι πίνακες που φυλάσσουν δεδομένα σχετικά με την υγρασία, οι οποίοι χρησιμοποιούνταν περισσότερο κατά την διάρκεια εκτέλεσης του κώδικα του WSM5. Οι πίνακες αυτοί φυλάσσονταν στο thread shared memory στην αρχή της δήλωσης του CUDA Kernel και επανατοποθετούνταν στο device memory στο τέλος. Αυτό ήταν και η μονή βελτιστοποίηση διαχείρισης μνήμης που εφαρμόστηκε κατά την μετατροπή του κώδικα.

Για να γίνει πιο εύκολη η μετατροπή και η ανάθεση των πινάκων στο σωστό μέρος μνήμης δημιουργήθηκε ένας προεπεξεργαστής σε Perl. Ο προεπεξεργαστής αυτός διαβάζει έναν ενδιάμεσο κώδικα CUDA C με κάποια directives. Τα directives καθορίζουν την διάσταση των πινάκων και το που θα τοποθετηθούν στην μνήμη. Αφού περάσει αυτός ο κώδικας από τον προεπεξεργαστή παράγεται κώδικας CUDA C με τα σωστά CUDA directives και με τις σωστές αναφορές στις θέσεις μνήμης των πινάκων. Πιο κάτω βλέπουμε ένα παράδειγμα από αυτήν την μέθοδο που χρησιμοποιήθηκε. Αυτή είναι μια δομή επανάληψης σε Fortran όπου γίνεται πρόσβαση στους τρισδιάστατους πίνακες t, Q, DEN.

```
DO j = jts, jte
  DO k = kts, kte
    DO i = its, ite
      IF (t(i,k,j).GT. t0c) THEN
        Q(i,k,j) = T(i,k,j) * DEN(i,k,j)
      ENDIF
    ENDDO
  ENDDO
ENDDO
```

Ο πιο πάνω κώδικας ξαναγράφεται και προστίθενται τα directives `_def`. Το `//_def_ arg i k j : q , t , den` λέει στον προεπεξεργαστή ότι τα q, t, den είναι τρισδιάστατοι πίνακες με διαστάσεις τα i, k, j και ότι είναι παράμετροι. Το `_def_ copy_up_memory i k j: q` λέει στον προεπεξεργαστή ότι το q πρέπει να πάει στο thread shared memory πριν την δομή επανάληψης και να επιστρέψει στο device memory όταν αυτή τελειώσει. Ένα επιπλέον μέτρο που διευκολύνει την μετατροπή είναι ότι οι δομές επανάληψης για τους τρισδιάστατους πίνακες είναι πλέον γραμμένες μόνο ως προς την κάθετη διάσταση των πινάκων.

```

//_def_arg i k j : q , t , den
//_def_copy_up_memory i k j:q
[ . . . ]
for (k = kps-1; k <= kpe-1; k+=J
    if (t[k] > tOc) {
        q[k] = t [ k ] * den[k] ;
    }
}
[ . . . ]
//_def_copy_down_memory ikj:q

```

Τέλος, στην τελική μορφή του κώδικα αφού περάσει από τον προεπεξεργαστή βλέπουμε στην πρώτη δομή επανάληψης τα δεδομένα του πίνακα q να μεταφέρονται στον πίνακα q\_s(q shared) ο οποίος φυλάσσεται στην thread shared memory. Στην επόμενη δομή επανάληψης γίνονται οι πράξεις και στην συνέχεια στην τρίτη δομή επανέρχονται τα δεδομένα του q στο device memory. Οι άλλες δυο διαστάσεις των πινάκων καθορίζονται από τις μεταβλητές ti και tj. Οι πίνακες S3 και D3 φυλάσσουν τις αναφορές στις σωστές θέσεις των πινάκων. Το S3 έχει τις αναφορές για την thread shared memory και το D3 για το device memory.

```

__shared__ float * q_s; int k;
[ . . . ]
for(k=kps-1;k<kpe;k++) {
    q_s [ S 3 ( t i , k , t j ) ] = q [ D 3 ( t i , k , t j ) ] ;
}
[ . . . ]

for ( k = kps-1 ; k <= kpe-1 ; k++ )
    if ( t [ k ] > tOc ) {
        q_s [ S 3 ( t i , k , t j ) ] = t [ D 3 ( t i , k , t j ) ] * d e n [ D 3 ( t i , k , t j ) ] ;
    }
}
[ . . . ]

for(k=kps-1;k<kpe;k++) {
    q [ D 3 ( t i , k , t j ) ] = q_s [ S 3 ( t i , k , t j ) ] ;
}

```

## 6.3 Εργαλεία που χρησιμοποιήσα στην υλοποίηση

### 6.3.1 FORTRAN

Η γλώσσα FORTRAN (από τα αρχικά FORmulae TRANslator - μεταφραστής τύπων) είναι μία από τις πρώτες γλώσσες υψηλής απόδοσης, η οποία χρησιμοποιήθηκε κυρίως σε επιστημονικές αλλά και σε εμπορικές εφαρμογές. Δημιουργήθηκε τη δεκαετία του 1950 από την IBM και χρησιμοποιείται μέχρι και σήμερα. Αρχικά η FORTRAN ήταν προσανατολισμένη στην επίλυση μαθηματικών προβλημάτων. Υπάρχουν οι εξής 5 τυποποιημένες εκδόσεις της FORTRAN:

- FORTRAN-66
- FORTRAN-77
- FORTRAN-90
- FORTRAN-95
- FORTRAN 2000
- FORTRAN 2003
- FORTRAN 2008

Ο περισσότερος κώδικας του WRF είναι γραμμένος σε fortran90

### 6.3.2 gfortran

gfortran είναι το όνομα του compiler GNU Fortran, το οποίο είναι μέρος του GNU Compiler Collection (GCC). gfortran αντικατέστησε το g77 compiler. Υποστηρίζει την γλώσσα Fortran 95 και είναι συμβατός με όλες τις γλώσσες που υποστήριζαν το g77. Συμπεριλαμβάνεται για πρώτη φορά στην έκδοση 4.1 του gcc.

### 6.3.3 LINUX

Το Linux (Λίνουξ) είναι ένας πυρήνας λειτουργικού συστήματος που μοιάζει με τον πυρήνα του AT&T UNIX. Είναι μία πρωτότυπη υλοποίηση πυρήνα λειτουργικού συστήματος και δεν χρησιμοποιεί κώδικα του UNIX. Μπορεί να θεωρηθεί κλώνος του UNIX, αφού διαθέτει τις περισσότερες εντολές του, ενώ η φιλοσοφία της σχεδίασής του πλησιάζει περισσότερο το UNIX από οποιοδήποτε άλλο λειτουργικό σύστημα. Το Linux αναπτύσσεται με βάση το POSIX πρότυπο, το οποίο είναι μία προσπάθεια τυποποίησης όλων των κλώνων του UNIX.

Παρ' όλο που το Linux είναι ο πυρήνας του λειτουργικού συστήματος, πολλές φορές αναφερόμαστε σε αυτό εννοώντας όλο το λειτουργικό σύστημα, που περιλαμβάνει και το περιβάλλον εργασίας, και το συνοδευτικό λογισμικό.

Η ανάπτυξη του πυρήνα Linux ξεκίνησε περίπου το 1991 από τον Φιλανδό Linus Torvalds , ο οποίος με βοήθεια πολλών εθελοντών προγραμματιστών μέσω του Internet, κατάφερε να δημιουργήσει έναν πυρήνα που ανταγωνίζεται πυρήνες λογισμικού μεγάλων εταιριών. Αρχικά ο πυρήνας αυτός ονομάστηκε FreaX (από τους όρους free και freak, με την κατάληξη X να υποδηλώνει ένα σύστημα τύπου Unix) αλλά αργότερα έλαβε την ονομασία Linux, ένα όρο που επινόησε ο Άρι Λέμκε (Ari Lemmke).

Σήμερα το Linux παρέχει όλα όσα θεωρούνται αναγκαία για ένα σύγχρονο πυρήνα λειτουργικού, όπως:

- υποστήριξη πολυεπεξεργαστικών συστημάτων (SMP)
- πραγματική πολυδιεργασία
- εικονική μνήμη
- διαμοιραζόμενες βιβλιοθήκες
- σωστή διαχείριση μνήμης
- δικτύωση μέσω TCP/IP κ.α.
- Ο πυρήνας Linux αρχικά σχεδιάστηκε για επεξεργαστές της οικογένειας x86 (386/486/Pentium), αλλά σήμερα τρέχει σε πολύ μεγάλη ποικιλία επεξεργαστών, όπως οι Alpha (64 bit), οι Motorola 68000 (Amiga), PowerPC, MIPS κ.α.

Χρησιμοποίησα την έκδοση Fedora8 του Linux

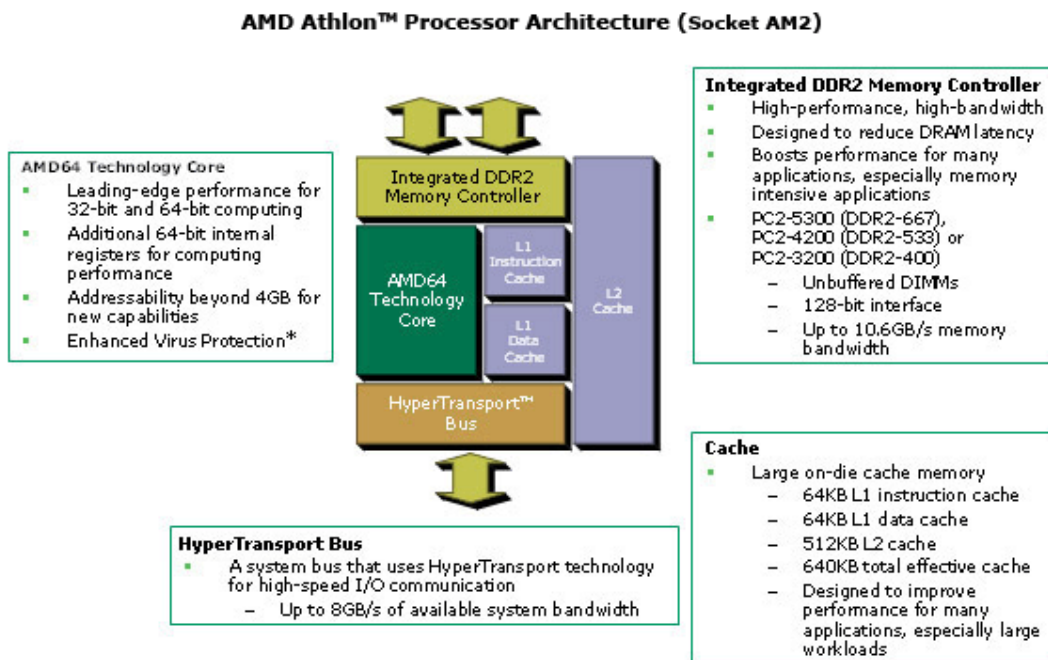
#### **6.3.4 AMD Code Analyst**

AMD CodeAnalyst Performance Analyzer είναι σύνολο εργαλείων για την ανάλυση της απόδοσης των λογισμικών σε AMD επεξεργαστές. Επίσης βοηθάει στην βελτίωση της λειτουργίας των εφαρμογών. Χρησιμοποιώντας αυτό το σύνολο εργαλείων οι χρήστες μπορούν να μεταφερθούν από το ένα σημείο της εφαρμογής σε άλλο εντοπίζοντας διάφορα προβλήματα, χρησιμοποιώντας hotspots.

### 6.3.5 AMD Athlon64 3200+

Ο επεξεργαστής AMD Athlon™ είναι ο πρώτος επεξεργαστής της Windows®-compatible 64-bit PC processor και έχει δημιουργηθεί κυρίως για desktop computers. Δημιουργεί ένα σύστημα για προστασία από virus αρκετά ενισχυμένο και βοηθάει στην φύλαξη των δεδομένων στους υπολογιστές. Επίσης αυξάνει την απόδοση των λογισμικών.

Στην εικόνα παρουσιάζεται η αρχιτεκτονική του AMD Athlon™



### 6.3.6 GCC

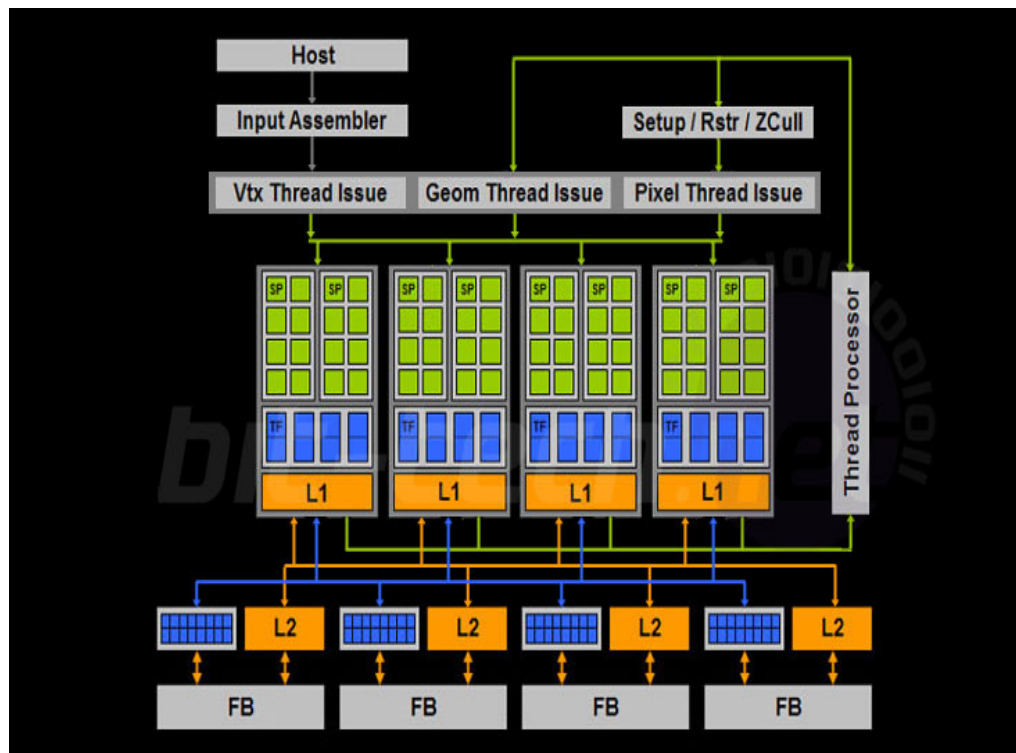
Το GCC (GNU Compiler Collection ) είναι ένα σύστημα από compilers το οποίο δημιουργήθηκε από το GNU Project για να υποστηρίξει διάφορες γλώσσες προγραμματισμού. Αρχικά ονομαστικέ GNU C Compiler γιατί μπορούσε να χειριστεί μόνο την γλώσσα προγραμματισμού C, GCC 1.0 το οποίο εκδόθηκε το 1987. Το compiler επεκτάθηκε στην C++ την ίδια χρονιά και αργότερα στην Fortran, Pascal, Objective-C, Java, και Ada. Έχει υιοθετηθεί ως το επίσημο compiler των λειτουργικών συστημάτων που μοιάζουν στην UNIX, συμπεριλαμβανομένου GNU/Linux, BSD και Mac OS X. Το GCC είναι διαθέσιμος για πολλές πλατφόρμες όπως Symbian, AMCC, Freescale Power Architecture, videogame consoles, Playstation 2 και Sega Dreamcast.

### 6.3.7 GEFORCE 9600GT

**Core Clock:** 650MHz  
**Shader Clock:** 1,625MHz  
**Memory Clock:** 1,800MHz (effective)

Η GeForce® 9600 GT GPU έχει σχεδιαστεί για υψηλή ανάλυση γραφικών. Βγήκε στην αγορά το 2007 από την NVIDIA σε αρκετά χαμηλή τιμή. Η υψηλή ανάλυση γραφικών και η χαμηλή τιμή της την κάνουν προσιτή κυρίως από τους χρήστες ηλεκτρονικών παιχνιδιών και video playback.

Το πιο κάτω διάγραμμα παρουσιάζει το Nvidia's G94 graphics chip (GeForce® 9600GT)



GeForce® 9600 GT GPU specifications

|                                 |          |          |
|---------------------------------|----------|----------|
| Processor Cores                 | 64       | 64       |
| Graphics Clock (MHz)            | 650 MHz  | 600 MHz  |
| Processor Clock (MHz)           | 1625 MHz | 1500 MHz |
| Texture Fill Rate (billion/sec) | 20.8     | 19.2     |

**Memory Specs:**

|                           |              |              |
|---------------------------|--------------|--------------|
| Memory Clock (MHz)        | 900 MHz      | 900 MHz      |
| Standard Memory Config    | 512 MB GDDR3 | 512 MB GDDR3 |
| Memory Interface Width    | 256-bit      | 256-bit      |
| Memory Bandwidth (GB/sec) | 57.6         | 57.6         |

**Feature Support:**

|                               |               |           |
|-------------------------------|---------------|-----------|
| NVIDIA SLI®-ready*            | 2-Way         | 2-Way     |
| GeForce 3D Vision Ready       | ✓             | ✓         |
| NVIDIA PureVideo® Technology* | HD            | HD        |
| NVIDIA PhysX™-ready           | ✓             | ✓         |
| NVIDIA CUDA™ Technology       | ✓             | ✓         |
| Microsoft DirectX             | 10            | 10        |
| OpenGL                        | 2.1           | 2.1       |
| Bus Support                   | PCI-E 2.0 x16 | PCI-E 2.0 |
| Certified for Windows Vista   | ✓             | ✓         |

**Display Support:**

|                             |                           |                           |
|-----------------------------|---------------------------|---------------------------|
| Maximum Digital Resolution  | 2560x1600                 | 2560x1600                 |
| Maximum VGA Resolution      | 2048x1536                 | 2048x1536                 |
| Standard Display Connectors | HDTV<br>Two Dual Link DVI | HDTV<br>Two Dual Link DVI |
| Multi Monitor               | ✓                         | ✓                         |
| HDCP*                       | ✓                         | ✓                         |
| HDMI*                       | Via adapter               | Via adapter               |
| Audio Input for HDMI        | SPDIF                     | SPDIF                     |

**Standard Graphics Card Dimensions:**

|        |              |              |
|--------|--------------|--------------|
| Height | 4.376 inches | 4.376 inches |
| Length | 9 inches     | 7.8 inches   |
| Width  | Single-slot  | Single-slot  |

**Thermal and Power Specs:**

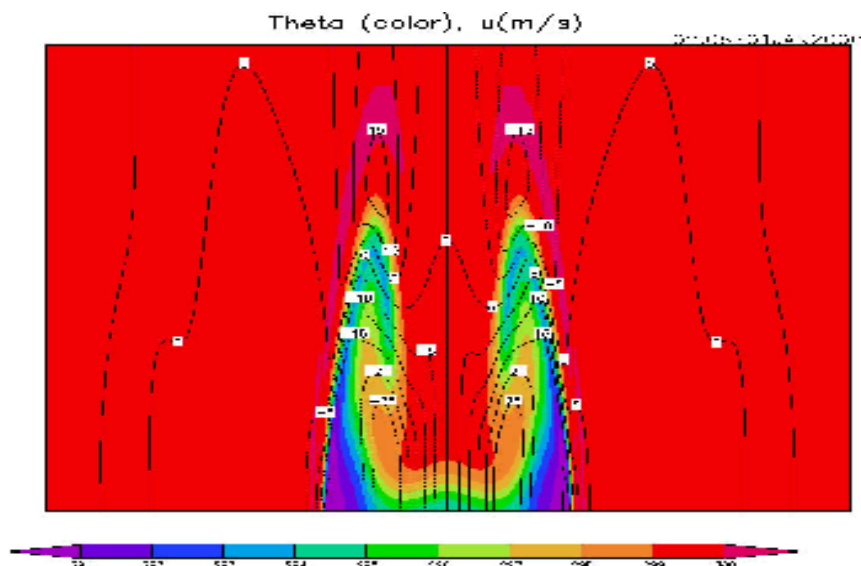
|                                      |       |       |
|--------------------------------------|-------|-------|
| Maximum GPU Temperature (in C)       | 105 C | 105 C |
| Maximum Graphics Card Power (W)      | 96 W  | 59 W  |
| Minimum System Power Requirement (W) | 400 W | 300 W |
| Supplementary Power Connectors       | 6-pin |       |

## 6.4 Χρόνοι μέτρησης

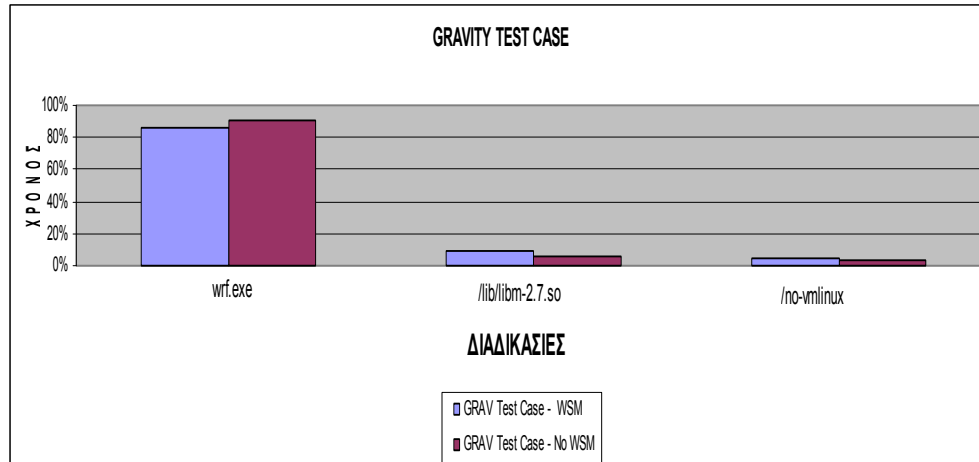
Η διαδικασία που ακλούθησα για να μετρήσω τους χρόνους ήταν καταρχάς να κάνω compile το μοντέλου συμφωνά με τις οδηγίες που βρίσκονται στο read me αρχείο του μοντέλου. Δηλαδή `./configure` για να επιλέξω το σωστό configuration για την μηχανή που έτρεξα. Στη συνέχεια `./compile wrf` για να μεταγλωττίσω το μοντέλου συμφωνά με το configuration που επέλεξα. Για να μπορέσω να τρέξω το κάθε test case υπάρχουν τρία επιπλέον βήματα. Πρώτον `./compile test case name` για να δημιουργηθούν τα εκτελέσιμα του test case. Δεύτερον `./ideal.exe` για να φορτώσω στο μοντέλο τις ρυθμίσεις για την αρχική κατάσταση της προσομοίωσης και το τι θέλω να προσομοιώσω και τρίτον `./wrf.exe` για να τρέξω την προσομοίωση. Το τελευταίο βήμα το εκτελούσε το CodeAnalyst και έτσι πήρα τις μετρήσεις για CPU. Τους χρόνους της GPU τους μέτρησα με την εντολή `time` του Linux καθώς το CodeAnalyst μπορεί να μετρήσει και να αναλύσει μόνο εφαρμογές που τρέχουν αποκλειστικά στο CPU.

### 6.4.1 Χρόνοι Μέτρησης - Test Case “EM\_grav2d\_x”

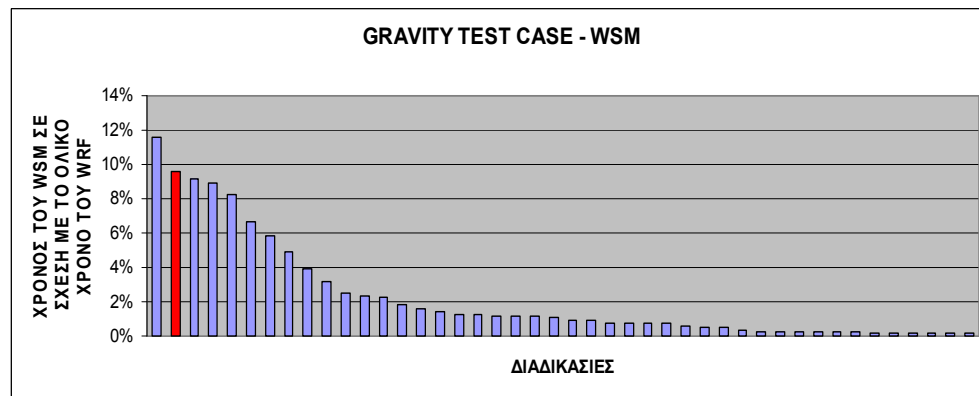
Το test case gravity αναφέρεται στην παρουσίαση πεδίων βαρύτητας στην επιφάνεια της γης. Παράδειγμα γραφικού αποτελέσματος του test case χρησιμοποιώντας NCL βλέπουμε πιο κάτω



Στον πιο κάτω πίνακα βλέπουμε πως είναι μοιρασμένος ο συνολικός χρόνος εκτέλεσης του WRF ανάμεσα στις διάφορες διεργασίες του συστήματος. Βλέπουμε ότι το wrf.exe απασχολεί το 85% του συνολικού χρόνου όταν χρησιμοποιείται το WSM5 και 90% του συνολικού χρόνου όταν δεν χρησιμοποιείται το WSM5. Φαίνεται ότι ανάλογα με την ποιότητα των μετρήσεων που θέλεις το WRF να σου δείξει αυξάνεται και ο φόρτος εργασίας του συστήματος.



Στην συνέχεια βλέπουμε πως μοιράζεται ο χρόνος του wrf.exe στις διαφορές διεργασίες του για την περίπτωση που χρησιμοποιείται το WSM5. Παρατηρούμε ότι το wsm5 module παίρνει το 7% του συνολικού χρόνου του wrf.exe.



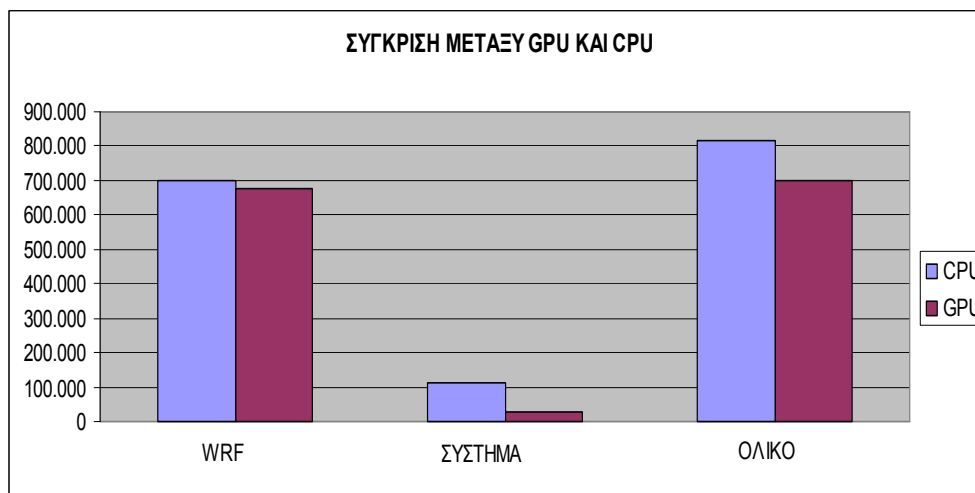
Τέλος βλέπουμε στους χρόνους εκτέλεσης σε κάρτα γραφικών καθώς και την σύγκριση μεταξύ του συνολικού χρόνου εκτέλεσης του WRF χωρίς το CUDA κώδικα και του συνολικού χρόνου εκτέλεσης του WRF που χρησιμοποιεί τον CUDA κώδικα. Παρατηρούμε πως υπάρχει μείωση χρόνου και στο χρόνο που παίρνει το WRF και στο χρόνο του συστήματος. Αφού η εκτέλεση γίνεται από την GPU ελαφρύνεται και ο φόρτος εργασίας του συστήματος. Αυτό συμβαίνει και στα επόμενα σενάρια με πιο εμφανή αποτελέσματα.

| gpu  |            |             |
|------|------------|-------------|
|      |            | Χρονος msec |
| real | 11m53.850s | 713.850     |
| user | 11m15.588s | 675.588     |
| sys  | 0m25.860s  | 25.860      |

| Σύγκριση με τα αποτελεσμάτων όταν το μοντέλο τρέχει σε κάρτα γραφικών |         |      |         |     |                                |     |
|-----------------------------------------------------------------------|---------|------|---------|-----|--------------------------------|-----|
|                                                                       | CPU     |      | GPU     |     | ΠΟΣΟ ΠΙΟ ΓΡΗΓΟΡΑ ΤΡΕΧΕΙ ΣΕ gpu |     |
| WRF                                                                   | 699.750 | 86%  | 675.588 | 95% | 24.162                         | 3%  |
| Σύστημα                                                               | 114.185 | 14%  | 25.860  | 4%  | 88.325                         | 77% |
| ΣΥΝΟΛΟ                                                                | 813.935 | 100% | 701.448 | 98% | 112.487                        | 14% |

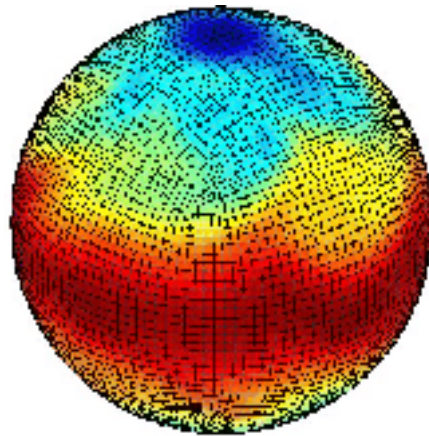
Το τελικό speedup για την εφαρμογή em\_grav2d\_x είναι 1.03 στο μοντέλο και 1.14 στο σύνολο. Αυτό το μικρό speedup είναι δικαιολογημένο καθώς το θέμα του σεναρίου και ο σκοπός του WSM5 δεν ανήκουν σε παρόμοιους τομείς. Το σενάριο έχει να κάνει με βαρύτητα και το WSM5 με αλλαγές στην ατμόσφαιρα. Παρόλο που το wsm5 module παίρνει μόνο 7% το τελικό speedup εξαρτάται και από τον χρόνο του συστήματος.

Τέλος βλέπουμε σε γραφική μορφή του χρόνου εκτέλεσης της κάθε έκδοσης του WSM5 καθώς και τον χρόνο του συστήματος. Φαίνεται και γραφικά αυτό που είπαμε πιο πάνω ότι παρόλο την μικρή βελτίωση του χρόνου εκτέλεσης του μοντέλου το τελικό speedup είναι μεγαλύτερο λόγω του ότι μειώνεται σημαντικά ο χρόνος του συστήματος.

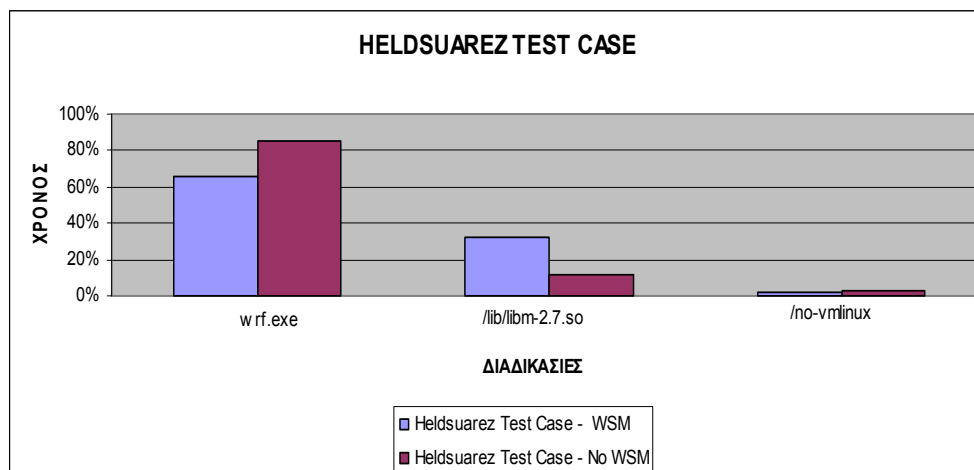


#### 6.4.2 Χρόνοι Μέτρησης - - Test Case “EM\_heldsuarez”

Το test case αναφέρεται στην παρουσίαση της εικόνας της επιφάνειας της γης και της ατμόσφαιρας που την περιβάλλει. Στην πιο κάτω εικόνα, που είναι παράδειγμα γραφικού αποτελέσματος για το test case, φαίνεται ότι η θερμοκρασία είναι πιο χαμηλή στους πόλους και πιο θερμή στον ισημερινό.

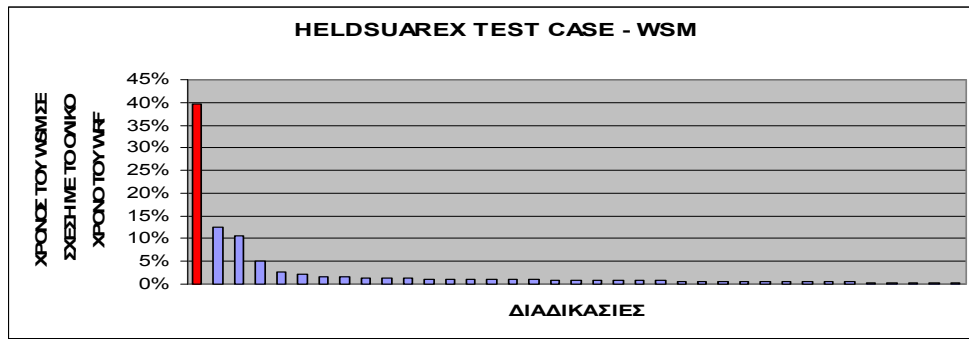


Στον πιο κάτω πίνακα βλέπουμε πως είναι μοιρασμένος ο συνολικός χρόνος εκτέλεσης του WRF ανάμεσα στις διάφορες διεργασίες του συστήματος. Βλέπουμε ότι το wrf.exe απασχολεί το 65% του συνολικού χρόνου όταν χρησιμοποιείται το WSM5 και 85% του συνολικού χρόνου όταν δεν χρησιμοποιείται το WSM5. Πάλι βλέπουμε πόσο πολύ επηρεάζει η ποιότητα των μετρήσεων που θέλεις να σου δείξει το WRF στην αύξηση του φόρτου εργασίας του συστήματος.



Στην συνέχεια βλέπουμε πως μοιράζεται ο χρόνος του wrf.exe στις διάφορες διεργασίες του όταν χρησιμοποιείται το WSM5. Παρατηρούμε ότι το wsm5 module παίρνει το 39% του συνολικού χρόνου του wrf.exe καθώς όπως είπα πιο πάνω, στην περιγραφή του WSM5, η δουλειά του είναι να υπολογίζει αλλαγές στην ατμόσφαιρα.

Αφού αυτό το σενάριο έχει να κάνει αποκλειστικά με αλλαγές στην ατμόσφαιρα είναι λογικό η εφαρμογή να περνά τον πιο πολύ χρόνο εκτέλεσης της στο wsm5 module.



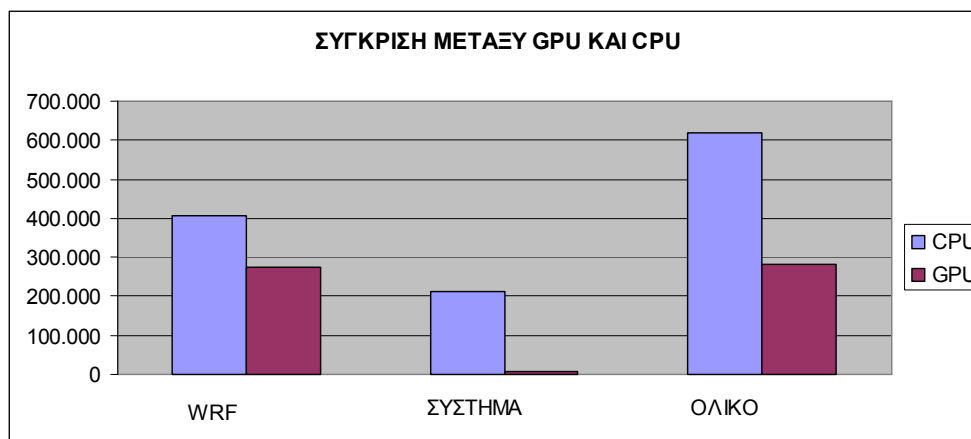
Η εκτέλεση του WRF σε συνδυασμό με τον κώδικα CUDA εμφάνισε τις πιο κάτω μετρήσεις.

| gpru        |           |         |
|-------------|-----------|---------|
| Χρονος msec |           |         |
| real        | 4m53.726s | 293.726 |
| user        | 4m34.543s | 274.543 |
| sys         | 0m8.134s  | 8.134   |

Συγκρίνοντας τους χρόνους για CPU και GPU καταλήγουμε στον πιο κάτω πίνακα. Αυτό που φαίνεται ξεκάθαρα είναι το πόσο μπορεί να βοηθήσει η εκτέλεση μιας εφαρμογής σε κάρτα γραφικών τον χρόνο εκτέλεσης αλλά και τον φόρτο εργασίας στον επεξεργαστή,

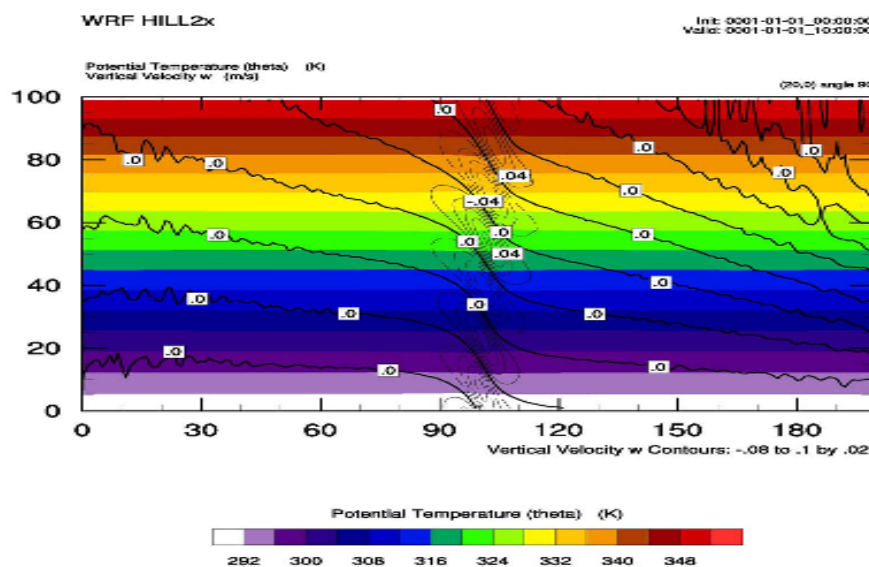
| Σύγκριση με τα αποτελέσματα όταν το μοντέλο τρέχει σε κάρτα γραφικών |         |               |            |               |            |                                |
|----------------------------------------------------------------------|---------|---------------|------------|---------------|------------|--------------------------------|
| Τύπος                                                                | Μοντέλο | CPU           |            | GPU           |            | ΠΟΣΟ ΠΙΟ ΓΡΗΓΟΡΑ ΤΡΕΧΕΙ ΣΕ gpu |
|                                                                      |         | Χρόνος (msec) | Φόρτος (%) | Χρόνος (msec) | Φόρτος (%) |                                |
| WRF                                                                  |         | 405.114       | 65%        | 274.543       | 93%        | 32%                            |
| Σύστημα                                                              |         | 214.325       | 35%        | 8.134         | 3%         | 96%                            |
| ΣΥΝΟΛΟ                                                               |         | 619.439       | 100%       | 282.677       | 96%        | 54%                            |

βλέπουμε την γραφική σύγκριση μεταξύ του συνολικού χρόνου εκτέλεσης του WRF χωρίς το CUDA κώδικα και του συνολικού χρόνου εκτέλεσης του WRF που χρησιμοποιεί τον CUDA κώδικα. Σε αυτό το σενάριο είχαμε 1.32 speedup στο μοντέλο και 1.54 τελικό speedup.

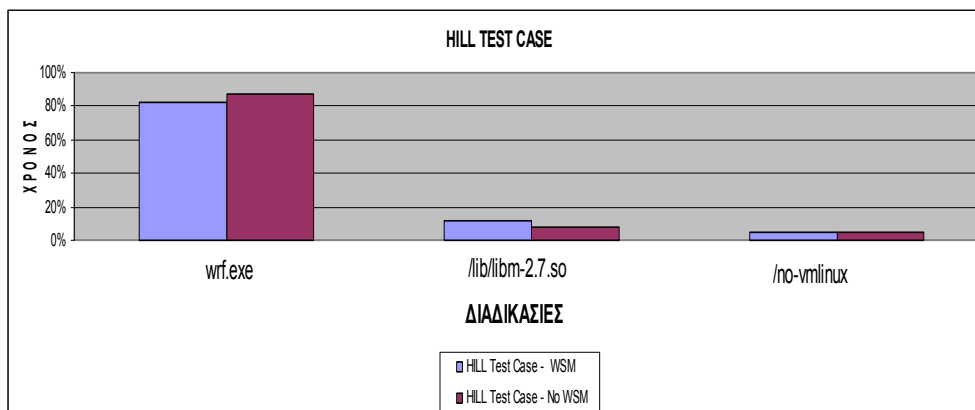


#### 6.4.3 Χρόνοι Μέτρησης - Test Case “EM\_hill2d”

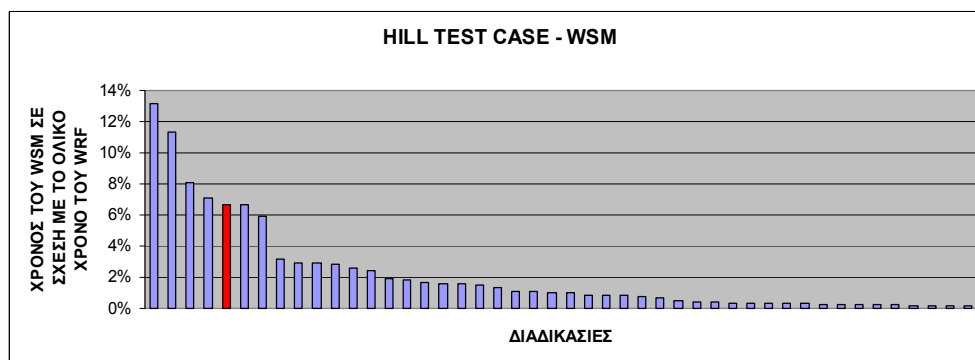
Το test case EM\_hill2d απεικονίζει τις ατμοσφαιρικές συνθήκες πάνω στην επιφάνεια ενός βουνού. Παράδειγμα γραφικού αποτελέσματος για το test case χρησιμοποιώντας NCL



Στον πιο κάτω πίνακα βλέπουμε πως είναι μοιρασμένος ο συνολικός χρόνος εκτέλεσης του WRF ανάμεσα στις διάφορες διεργασίες του συστήματος. Βλέπουμε ότι το wrf.exe απασχολεί το 83% του συνολικού χρόνου όταν χρησιμοποιείται το WSM5 και 88% του συνολικού χρόνου όταν δεν χρησιμοποιείται το WSM5. Πάλι βλέπουμε πόσο πολύ επηρεάζει η ποιότητα των μετρήσεων που θέλεις να σου δείξει το WRF στην αύξηση του φόρτου εργασίας του συστήματος.



Στην συνέχεια βλέπουμε πως μοιράζεται ο χρόνος του wrf.exe στις διάφορες διεργασίες του όταν χρησιμοποιείται το WSM5. Παρατηρούμε ότι το wsm5 module παίρνει το 7% του συνολικού χρόνου του wrf.exe. Σε συνδυασμό με την πιο πάνω γραφική παράσταση αλλά και με τις γραφικές παραστάσεις των άλλων δυο σεναρίων που είδαμε ήδη παρατηρούμε ότι το wsm5 module επηρεάζει κατά πολύ τον χρόνο του συστήματος. Όταν ο χρόνος του wsm5 module, σε σχέση με τον χρόνο εκτέλεσης του μοντέλου, είναι πολύς τότε και ο χρόνος του συστήματος είναι πολύς και το αντίστροφο.



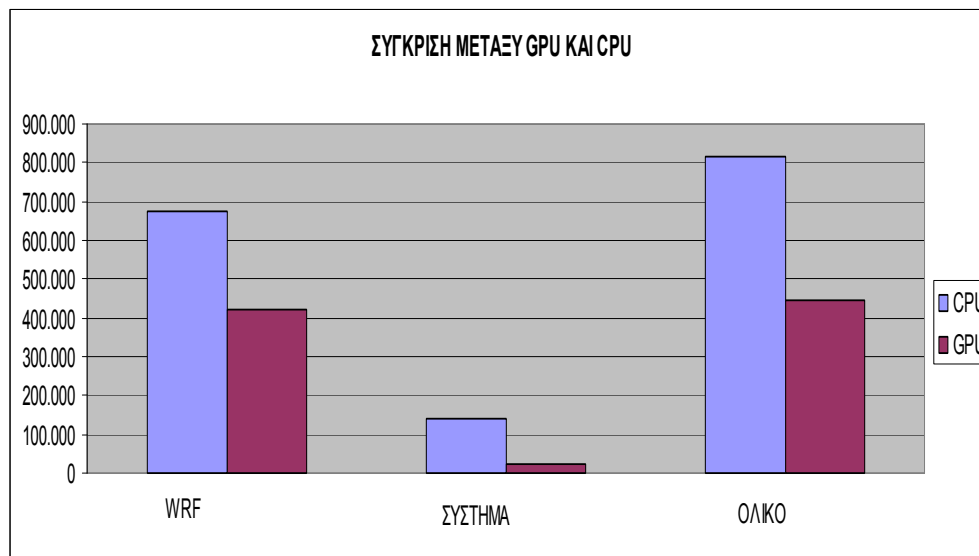
Η εκτέλεση του WRF σε συνδυασμό με τον κώδικα CUDA εμφάνισε τις πιο κάτω μετρήσεις.

| gpu  |           |             |
|------|-----------|-------------|
|      |           | Χρονος msec |
| real | 7m42.723s | 462.723     |
| user | 7m2.804s  | 422.804     |
| sys  | 0m22.523s | 22.523      |

Συγκρίνοντας τους χρόνους για CPU και GPU καταλήγουμε στον πιο κάτω πίνακα. Εφόσον το wsm5 module είναι μόνο το 7% το speedup στο μοντέλο είναι υπερβολικά ψηλό. Η εξήγηση είναι ότι το wsm5 module καλείται από τα πιο πάνω modules άρα όταν μειώνεται ο χρόνος του wsm5 μειώνεται ο χρόνος και για αυτά που το καλούν.

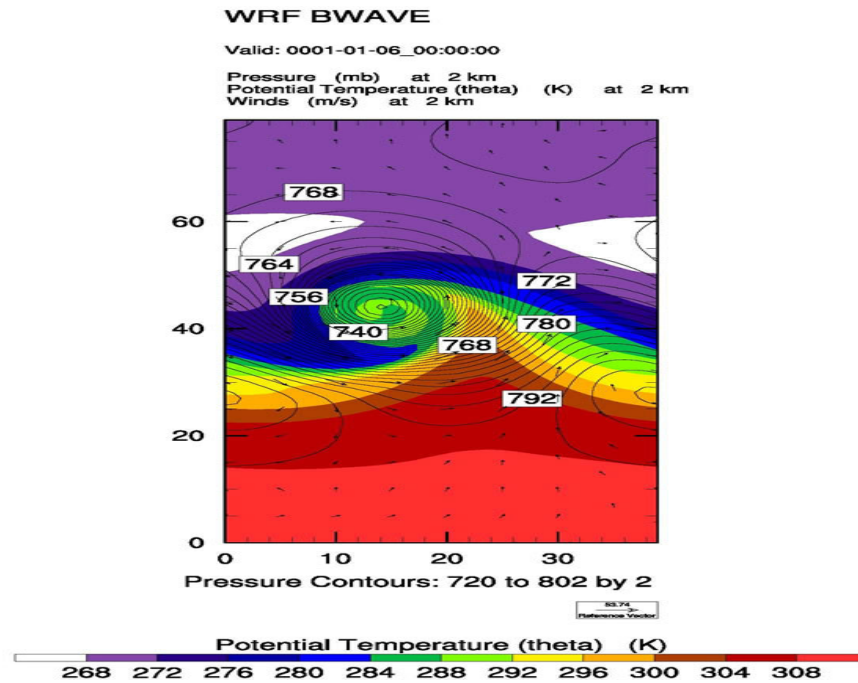
| Σύγκριση με τα αποτελέσματα όταν το μοντέλο τρέχει σε κάρτα γραφικών |         |      |         |     |                                |     |
|----------------------------------------------------------------------|---------|------|---------|-----|--------------------------------|-----|
|                                                                      | CPU     |      | GPU     |     | ΠΟΣΟ ΠΙΟ ΓΡΗΓΟΡΑ ΤΡΕΧΕΙ ΣΕ gpu |     |
| WRF                                                                  | 673.577 | 83%  | 422.804 | 91% | 250.773                        | 37% |
| Σύστημα                                                              | 140.355 | 17%  | 22.523  | 5%  | 117.832                        | 84% |
| ΣΥΝΟΛΟ                                                               | 813.932 | 100% | 445.327 | 96% | 368.605                        | 45% |

Η γραφική αναπαράσταση των χρόνων εκτέλεσης μας δείχνει καλύτερα τα αποτελέσματα του πιο πάνω πίνακα. Και σε αυτό το σενάριο είχαμε σχεδόν 1.5 speedup. 1.37 στο μοντέλο και 1.45 τελικό speedup.

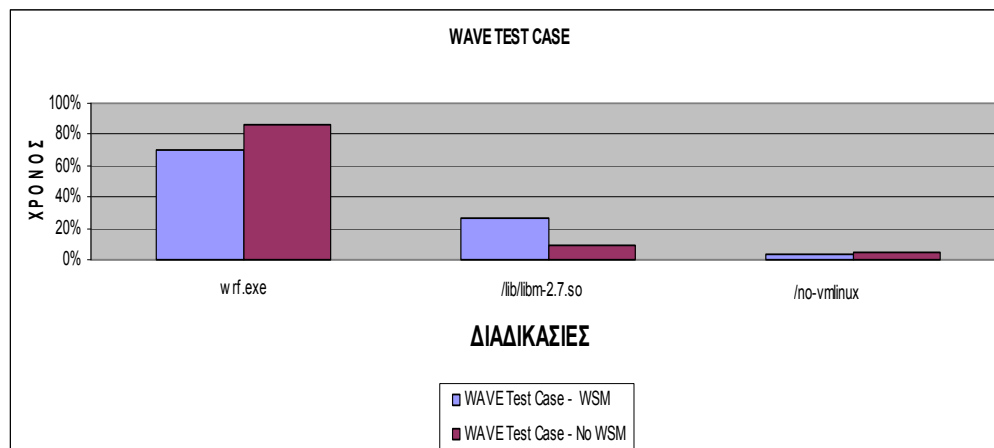


#### 6.4.4 Χρόνοι Μέτρησης - Test Case “EM\_b\_wave”

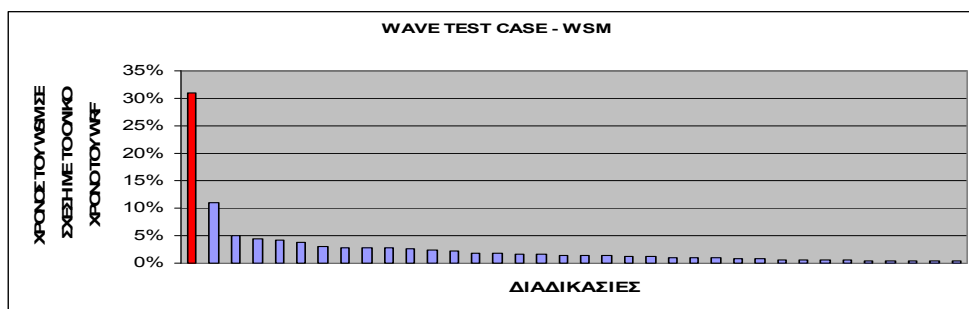
Το test case EM\_b\_wave απεικονίζει τις συνθήκες που επικρατούν σε ένα κύμα κατά την διάρκεια που κινείται. Παράδειγμα γραφικού αποτελέσματος γι’ αυτό το test case βλέπουμε πιο κατω.



Στον πιο κάτω πίνακα βλέπουμε πως είναι μοιρασμένος ο συνολικός χρόνος εκτέλεσης του WRF ανάμεσα στις διάφορες διεργασίες του συστήματος. Βλέπουμε ότι το wrf.exe απασχολεί το 70% του συνολικού χρόνου όταν χρησιμοποιείται το WSM5 και 86% του συνολικού χρόνου όταν δεν χρησιμοποιείται το WSM5.



Στην συνέχεια βλέπουμε πως μοιράζεται ο χρόνος του wrf.exe στις διάφορες διεργασίες του όταν χρησιμοποιείται το WSM5. Παρατηρούμε ότι το wsm5 module παίρνει το 31% του συνολικού χρόνου του wrf.exe. Παρατηρούμε ξανά ότι το wsm5 module επηρεάζει κατά πολύ τον χρόνο του συστήματος.



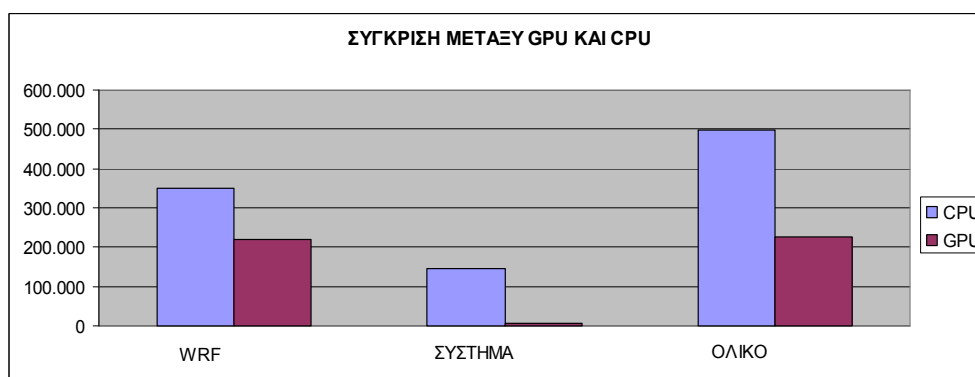
Η εκτέλεση του WRF σε συνδυασμό με τον κώδικα CUDA εμφάνισε τις πιο κάτω μετρήσεις.

| gpu         |           |         |
|-------------|-----------|---------|
| Χρονος msec |           |         |
| real        | 3m47.019s | 227.019 |
| user        | 3m39.930s | 219.930 |
| sys         | 0m6.043s  | 6.043   |

Συγκρίνοντας τους χρόνους για CPU και GPU καταλήγουμε στον πιο κάτω πίνακα. Πάλι βλέπουμε μια μείωση του χρόνου που είναι μεγαλύτερη από την αναμενόμενη. Επειδή δεν είναι τόσο μεγάλη διάφορα όσο το προηγούμενο σενάριο πιθανώς εδώ να έχει παίξει ρολό και το πόσο το Code Analyst επηρεάζει τους χρόνους εκτέλεσης σε CPU καθώς τρέχει ταυτόχρονα με την εφαρμογή για να κάνει τις μετρήσεις.

| Σύγκριση με τα αποτελέσματα όταν το μοντέλο τρέχει σε κάρτα γραφικών |         |      |         |      |                                |     |
|----------------------------------------------------------------------|---------|------|---------|------|--------------------------------|-----|
|                                                                      | CPU     |      | GPU     |      | ΠΟΣΟ ΠΙΟ ΓΡΗΓΟΡΑ ΤΡΕΧΕΙ ΣΕ gpu |     |
| WRF                                                                  | 349.977 | 70%  | 219.930 | 97%  | 130.047                        | 37% |
| Σύστημα                                                              | 146.468 | 30%  | 6.043   | 3%   | 140.425                        | 96% |
| ΣΥΝΟΛΟ                                                               | 496.445 | 100% | 225.973 | 100% | 270.472                        | 54% |

Η γραφική αναπαράσταση των χρόνων εκτέλεσης μας δείχνει καλύτερα τα αποτελέσματα του πιο πάνω πίνακα. Και σε αυτό το σενάριο είχαμε σχεδόν 1.5 speedup. 1.37 στο μοντέλο και 1.54 τελικό speedup.



## 6.5 Συμπεράσματα

Το WRF είναι ένα πολύ καλό μοντέλο προσομοίωσης συνθηκών περιβάλλοντος και ένα χρήσιμο εργαλείο για τον επιστημονικό τομέα που απευθύνεται. Μπορεί να χρησιμοποιηθεί για πολλές διαφορετικές προσομοιώσεις όπως είδαμε, από προσομοίωση καιρικών συνθηκών μέχρι προσομοίωση κινήσεων των κυμάτων με αρκετά καλή ακρίβεια. Για ένα κανονικό PC όμως είναι αρκετά «βαρύ» πρόγραμμα καθώς όταν προσθέσεις επιλογές στο μοντελο που θα κάνουν την πρόβλεψη πιο ακριβής σπαταλείται αρκετός χρόνος και από διεργασίες του συστήματος. Καλύτερο περιβάλλον για να τρέξει το μοντελο πιστεύω ότι είναι ένα συστημα cluster με χρήση more. Επιπλέον με το πακέτο του μοντέλου θα πρέπει να παρέχονται όλα τα εργαλεία που είναι απαραίτητα για να εκτελέσεις μια προσομοίωση και να δεις γραφικά αποτελέσματα. Εργαλεία που έχουν να κάνουν με τον τύπο των αρχείων που διαβάζει το WRF καθώς και τα εργαλεία που βλέπεις τελικά το γραφικό αποτέλεσμα της προσομοιώσεις δεν παρέχονται με αποτέλεσμα να χάνεις χρόνο ψάχνοντας τα και συνδέοντας τα με το μοντελο.

Όσον αφορά την μετατροπή του κώδικα πρέπει να τονίσω ότι η μετατροπή οποιουδήποτε κώδικα από μια γλώσσά σε άλλη απαιτεί καλή γνώση και των δυο γλωσσών που θα χρησιμοποιήσεις. Είναι απαραίτητο να γνωρίζεις περισσότερο από τα βασικά καθώς κάθε γλώσσα έχει τα δικά της πλεονεκτήματα που πρέπει να εκμεταλλευτείς και τα δικά της μειονεκτήματα που πρέπει να προσέχεις. Για μετατροπή κώδικα σε CUDA απαιτείται πολύ καλή γνώση του αρχικού κώδικα γιατί όταν θα μετατρέψεις τον κώδικα πρέπει να δώσεις προσοχή στην διαχείριση της διαθέσιμης μνήμης. Επιπλέον έχεις να κάνεις με παράλληλη επεξεργασία έτσι ,παρόλο τον αυτοματισμό που προσφέρει η CUDA, πρέπει να δείξεις προσοχή στην δημιουργία και διαχείριση διεργασιών. Τέλος απαιτείται γνώση της κάρτας όπου θα τρέξει ο κώδικας για να μπορέσεις να κάνεις σωστά τα δυο πιο πάνω. Δηλαδή διαχείριση μνήμης και διαχείριση διεργασιών. Εφόσον όμως η Cuda χρησιμοποιείται σωστά και μονό για υπολογισμούς το speedup της εφαρμογής μπορεί να είναι πάρα πολύ καλό.

Όσο αφορά το WRF, η μετατροπή μέρους κώδικα του WRF σε κώδικα για GPU θα έχει πάρα πολύ καλές συνέπειες όσο αφορά την ταχύτητα που οι επιστήμονες θα βλέπουν τα αποτελέσματα των προσομοιώσεων. Όπως είδαμε αλλάζοντας μόνο ένα κομμάτι του είχαμε speedup μέχρι και . Το ίδιο μπορεί να γίνει και με άλλα κομμάτια του WRF αφού το ενδιαφέρον και για τις GPUs και για το WRF είναι αρκετά μεγάλο από τον επιστημονικό τομέα. Ίσως θα ήταν καλό να δημοσιευτεί και μια έκδοση του WRF με ενσωματωμένο κώδικα που τρέχει σε GPU αφού δεν υπάρχει μόνο μια προσπάθεια

υλοποίησης μέρους του WRF σε CUDA. Αυτό θα βοηθούσε και στην ταχύτητα που θα εργάζεται το μοντέλο αλλά και στην επιστημονική έρευνα κυρίως λόγω των νέων συστημάτων που χρησιμοποιούν τεχνολογία όπως η Tesla.

## Appendix

grav wsm5

| Module Name                      | CPU clocks |     |
|----------------------------------|------------|-----|
| /root/Desktop/WRFV3/main/wrf.exe | 699.750    | 86% |
| /lib/libm-2.7.so                 | 77.137     | 9%  |
| /no-vmlinux                      | 37.048     | 5%  |
|                                  | 813.935    |     |

| CS:EIP                                                      | CPU clocks |     |
|-------------------------------------------------------------|------------|-----|
| __module_bc_set_physical_bc3d                               | 79.980     | 12% |
| __module_mp_wsm5_wsm52d                                     | 66.396     | 10% |
| __module_advect_em_advect_scalar                            | 63.360     | 9%  |
| __module_small_step_em_advance_uv                           | 61.700     | 9%  |
| __module_small_step_em_advance_w                            | 57.046     | 8%  |
| __module_small_step_em_advance_mu_t                         | 45.993     | 7%  |
| __module_diffusion_em_horizontal_diffusion_s                | 40.235     | 6%  |
| __module_big_step_utilities_em_horizontal_pressure_gradient | 33.817     | 5%  |
| __module_small_step_em_calc_p_rho                           | 27.194     | 4%  |
| __module_small_step_em_sumflux                              | 21.973     | 3%  |
| __module_advect_em_advect_v                                 | 17.056     | 2%  |
| __module_advect_em_advect_u                                 | 15.887     | 2%  |
| __module_advect_em_advect_w                                 | 15.636     | 2%  |
| __module_big_step_utilities_em_rhs_ph                       | 12.779     | 2%  |
| __module_big_step_utilities_em_curvature                    | 10.912     | 2%  |
| __module_diffusion_em_cal_deform_and_div                    | 10.062     | 1%  |
| __module_big_step_utilities_em_coriolis                     | 8.866      | 1%  |
| __module_big_step_utilities_em_calc_cq                      | 8.583      | 1%  |
| __module_small_step_em_small_step_prep                      | 7.986      | 1%  |
| __module_big_step_utilities_em_zero_tend                    | 7.846      | 1%  |
| __module_em_rk_update_scalar                                | 7.808      | 1%  |
| __module_em_rk_addtend_dry                                  | 7.369      | 1%  |
| __module_big_step_utilities_em_calc_p_rho_phi               | 6.132      | 1%  |
| __module_small_step_em_small_step_finish                    | 6.111      | 1%  |
| __module_diffusion_em_vertical_diffusion_s                  | 5.241      | 1%  |
| __module_big_step_utilities_em_w_damp                       | 5.206      | 1%  |
| __module_small_step_em_calc_coef_w                          | 5.074      | 1%  |
| __module_mp_wsm5_wsm5                                       | 4.968      | 1%  |
| __module_diffusion_em_compute_diff_metrics                  | 3.834      | 1%  |
| __module_big_step_utilities_em_phy_prep                     | 3.336      | 0%  |
| __module_diffusion_em_calculate_n2                          | 3.307      | 0%  |
| __module_diffusion_em_horizontal_diffusion_v_2              | 2.176      | 0%  |
| __module_diffusion_em_vertical_diffusion_v_2                | 1.860      | 0%  |
| __module_big_step_utilities_em_couple_momentum              | 1.805      | 0%  |
| __module_big_step_utilities_em_calc_ww_cp                   | 1.804      | 0%  |
| __module_diffusion_em_horizontal_diffusion_w_2              | 1.775      | 0%  |
| __module_big_step_utilities_em_pg_buoy_w                    | 1.651      | 0%  |
| __module_diffusion_em_horizontal_diffusion_u_2              | 1.601      | 0%  |
| __module_big_step_utilities_em_moist_physics_prep_em        | 1.345      | 0%  |
| __module_diffusion_em_cal_tau_12_21                         | 1.303      | 0%  |
|                                                             | 691.662    |     |

grav no wsm

| Module Name                      | CPU clocks |     |
|----------------------------------|------------|-----|
| /root/Desktop/WRFV3/main/wrf.exe | 542.896    | 90% |
| /lib/libm-2.7.so                 | 34.812     | 6%  |
| /no-vmLinux                      | 23.558     | 4%  |
|                                  | 601.266    |     |

| CS:EIP                                                       | CPU clocks |     |
|--------------------------------------------------------------|------------|-----|
| __module_bc__set_physical_bc3d                               | 72.096     | 13% |
| __module_small_step_em__advance_uv                           | 61.023     | 11% |
| __module_small_step_em__advance_w                            | 60.807     | 11% |
| __module_small_step_em__advance_mu_t                         | 51.057     | 10% |
| __module_big_step_utilities_em__horizontal_pressure_gradient | 32.847     | 6%  |
| __module_advect_em__advect_scalar                            | 28.680     | 5%  |
| __module_small_step_em__calc_p_rho                           | 24.276     | 5%  |
| __module_small_step_em__sumflux                              | 21.668     | 4%  |
| __module_advect_em__advect_v                                 | 16.939     | 3%  |
| __module_advect_em__advect_u                                 | 15.756     | 3%  |
| __module_advect_em__advect_w                                 | 15.650     | 3%  |
| __module_diffusion_em__horizontal_diffusion_s                | 14.928     | 3%  |
| __module_big_step_utilities_em__rhs_ph                       | 12.728     | 2%  |
| __module_big_step_utilities_em__curvature                    | 11.116     | 2%  |
| __module_diffusion_em__cal_deform_and_div                    | 9.790      | 2%  |
| __module_big_step_utilities_em__coriolis                     | 8.859      | 2%  |
| __module_small_step_em__small_step_prep                      | 8.270      | 2%  |
| __module_em__rk_addtend_dry                                  | 7.509      | 1%  |
| __module_small_step_em__small_step_finish                    | 6.385      | 1%  |
| __module_big_step_utilities_em__calc_p_rho_phi               | 6.200      | 1%  |
| __module_big_step_utilities_em__zero_tend                    | 5.547      | 1%  |
| __module_big_step_utilities_em__w_damp                       | 5.226      | 1%  |
| __module_small_step_em__calc_coef_w                          | 5.201      | 1%  |
| __module_diffusion_em__compute_diff_metrics                  | 3.847      | 1%  |
| __module_big_step_utilities_em__phy_prep                     | 3.582      | 1%  |
| __module_big_step_utilities_em__calc_cq                      | 3.549      | 1%  |
| __module_diffusion_em__calculate_n2                          | 2.930      | 1%  |
| __module_diffusion_em__horizontal_diffusion_v_2              | 2.150      | 0%  |
| __module_diffusion_em__vertical_diffusion_v_2                | 1.866      | 0%  |
| __module_big_step_utilities_em__calc_ww_cp                   | 1.817      | 0%  |
| __module_big_step_utilities_em__couple_momentum              | 1.811      | 0%  |
| __module_diffusion_em__vertical_diffusion_s                  | 1.800      | 0%  |
| __module_diffusion_em__horizontal_diffusion_w_2              | 1.726      | 0%  |
| __module_diffusion_em__horizontal_diffusion_u_2              | 1.550      | 0%  |
| __module_em__rk_update_scalar                                | 1.468      | 0%  |
| __module_big_step_utilities_em__pg_buoy_w                    | 1.464      | 0%  |
|                                                              | 536.844    |     |

| gpu  |            | Χρονος msec |  |
|------|------------|-------------|--|
| real | 11m53.850s | 713.850     |  |
| user | 11m15.588s | 675.588     |  |
| sys  | 0m25.860s  | 25.860      |  |

| Module Name                      | CPU<br>clocks | Percentage |
|----------------------------------|---------------|------------|
| /root/Desktop/WRFV3/main/wrf.exe | 405.114       | 65%        |
| /lib/libm-2.7.so                 | 203.043       | 33%        |
| /no-vmlinux                      | 11.282        | 2%         |
|                                  | 619.439       |            |

| Basic Block : ( Load / Store )                               | CPU<br>clocks | Percentage |
|--------------------------------------------------------------|---------------|------------|
| __module_mp_wsm5__wsm52d                                     | 160.316       | 40%        |
| mradf4_                                                      | 50.450        | 12%        |
| mradb4_                                                      | 43.630        | 11%        |
| __module_advect_em__advect_scalar                            | 20.895        | 5%         |
| __module_small_step_em__advance_mu_t                         | 10.769        | 3%         |
| __module_small_step_em__advance_uv                           | 8.794         | 2%         |
| __module_small_step_em__calc_p_rho                           | 6.746         | 2%         |
| __module_small_step_em__sumflux                              | 5.938         | 1%         |
| __module_big_step_utilities_em__horizontal_diffusion         | 5.644         | 1%         |
| __module_advect_em__advect_u                                 | 5.190         | 1%         |
| __module_advect_em__advect_v                                 | 5.060         | 1%         |
| __module_bc_set_physical_bc3d                                | 4.544         | 1%         |
| __module_big_step_utilities_em__horizontal_pressure_gradient | 4.521         | 1%         |
| __module_big_step_utilities_em__rhs_ph                       | 4.096         | 1%         |
| mrftb1_                                                      | 4.069         | 1%         |
| __module_polarfft__polar_filter_fft_2d_ncar                  | 3.915         | 1%         |
| __module_big_step_utilities_em__calc_p_rho_phi               | 3.867         | 1%         |
| mrftf1_                                                      | 3.700         | 1%         |
| __module_em__rk_update_scalar                                | 3.300         | 1%         |
| __module_big_step_utilities_em__curvature                    | 2.955         | 1%         |
| __module_diffusion_em__cal_deform_and_div                    | 2.936         | 1%         |
| __module_small_step_em__small_step_prep                      | 2.864         | 1%         |
| vssqrt_                                                      | 2.853         | 1%         |
| __module_damping_em__held_suarez_damp                        | 2.654         | 1%         |
| __module_big_step_utilities_em__coriolis                     | 2.543         | 1%         |
| __module_em__rk_addtend_dry                                  | 2.527         | 1%         |
| __module_big_step_utilities_em__calc_cq                      | 2.473         | 1%         |
| __module_small_step_em__small_step_finish                    | 2.289         | 1%         |
| vsrec_                                                       | 2.080         | 1%         |
| __module_big_step_utilities_em__zero_tend                    | 2.079         | 1%         |
| __module_big_step_utilities_em__w_damp                       | 1.896         | 0%         |
| __module_polarfft__polar_filter_3d                           | 1.708         | 0%         |
| __module_big_step_utilities_em__vertical_diffusion           | 1.565         | 0%         |
| __module_big_step_utilities_em__phy_prep                     | 1.370         | 0%         |
| __module_big_step_utilities_em__diagnose_w                   | 1.363         | 0%         |
| __module_diffusion_em__calculate_n2                          | 1.294         | 0%         |
| __module_diffusion_em__compute_diff_metrics                  | 1.063         | 0%         |
|                                                              | 393.956       |            |

held no wsm

| Module Name                      | CPU clocks | Percentage |
|----------------------------------|------------|------------|
| /root/Desktop/WRFV3/main/wrf.exe | 367.857    | 86%        |
| /lib/libm-2.7.so                 | 51.157     | 12%        |
| /no-vmlinux                      | 10.798     | 3%         |
|                                  | 429.812    |            |

| Basic Block : ( Load / Store )                               | CPU clocks | Percentage |
|--------------------------------------------------------------|------------|------------|
| mradf4_                                                      | 76.068     | 21%        |
| mradb4_                                                      | 65.862     | 18%        |
| __module_small_step_em__advance_mu_t                         | 20.765     | 6%         |
| __module_small_step_em__advance_uv                           | 16.408     | 4%         |
| __module_small_step_em__calc_p_rho                           | 14.757     | 4%         |
| __module_advect_em__advect_scalar                            | 14.422     | 4%         |
| __module_small_step_em__sumflux                              | 11.709     | 3%         |
| __module_advect_em__advect_u                                 | 9.726      | 3%         |
| __module_advect_em__advect_v                                 | 9.401      | 3%         |
| __module_big_step_utilities_em__horizontal_pressure_gradient | 9.333      | 3%         |
| __module_big_step_utilities_em__rhs_ph                       | 9.133      | 2%         |
| __module_small_step_em__small_step_prep                      | 7.851      | 2%         |
| __module_bc__set_physical_bc3d                               | 7.230      | 2%         |
| __module_big_step_utilities_em__curvature                    | 7.057      | 2%         |
| __module_small_step_em__small_step_finish                    | 6.422      | 2%         |
| mrftb1_                                                      | 6.057      | 2%         |
| __module_polarfft__polar_filter_fft_2d_ncar                  | 5.906      | 2%         |
| mrftf1_                                                      | 5.749      | 2%         |
| __module_big_step_utilities_em__coriolis                     | 5.547      | 2%         |
| __module_big_step_utilities_em__horizontal_diffusion         | 5.507      | 1%         |
| __module_diffusion_em__cal_deform_and_div                    | 5.325      | 1%         |
| __module_damping_em__held_suarez_damp                        | 4.926      | 1%         |
| __module_em__rk_addtend_dry                                  | 4.709      | 1%         |
| __module_big_step_utilities_em__calc_p_rho_phi               | 3.993      | 1%         |
| __module_big_step_utilities_em__w_damp                       | 3.533      | 1%         |
| __module_big_step_utilities_em__phy_prep                     | 2.735      | 1%         |
| __module_big_step_utilities_em__zero_tend                    | 2.718      | 1%         |
| __module_polarfft__polar_filter_3d                           | 2.572      | 1%         |
| __module_big_step_utilities_em__diagnose_w                   | 2.567      | 1%         |
| __module_diffusion_em__calculate_n2                          | 2.150      | 1%         |
| __module_diffusion_em__compute_diff_metrics                  | 2.010      | 1%         |
| __module_big_step_utilities_em__calc_cq                      | 1.956      | 1%         |
| __module_big_step_utilities_em__horizontal_diffusion_3dmp    | 1.477      | 0%         |
| __module_em__rk_update_scalar                                | 1.189      | 0%         |
| __module_big_step_utilities_em__calc_ww_cp                   | 1.187      | 0%         |
| __module_big_step_utilities_em__couple_momentum              | 1.177      | 0%         |
|                                                              | 359.134    |            |

| gpu  |           |                |
|------|-----------|----------------|
|      |           | Χρονος<br>msec |
| real | 4m53.726s | 293.726        |
| user | 4m34.543s | 274.543        |
| sys  | 0m8.134s  | 8.134          |

| Module Name                      | CPU clocks |     |
|----------------------------------|------------|-----|
| /root/Desktop/WRFV3/main/wrf.exe | 673.577    | 83% |
| /lib/libm-2.7.so                 | 98.307     | 12% |
| /no-vmlinux                      | 42.048     | 5%  |
|                                  | 813.932    |     |

| Basic Block : ( Load / Store )                               | CPU clocks |     |
|--------------------------------------------------------------|------------|-----|
| __module_bc__set_physical_bc3d                               | 87.244     | 13% |
| __module_advect_em__advect_scalar                            | 75.209     | 11% |
| __module_small_step_em__advance_uv                           | 53.776     | 8%  |
| __module_diffusion_em__horizontal_diffusion_s                | 46.936     | 7%  |
| __module_mp_wsm5__wsm52d                                     | 44.096     | 7%  |
| __module_small_step_em__advance_w                            | 44.044     | 7%  |
| __module_small_step_em__advance_mu_t                         | 39.154     | 6%  |
| __module_advect_em__advect_v                                 | 20.754     | 3%  |
| __module_advect_em__advect_u                                 | 19.576     | 3%  |
| __module_advect_em__advect_w                                 | 19.332     | 3%  |
| __module_small_step_em__calc_p_rho                           | 18.534     | 3%  |
| __module_big_step_utilities_em__horizontal_pressure_gradient | 17.104     | 3%  |
| __module_big_step_utilities_em__rhs_ph                       | 15.818     | 2%  |
| __module_big_step_utilities_em__curvature                    | 12.779     | 2%  |
| __module_small_step_em__sumflux                              | 12.134     | 2%  |
| __module_big_step_utilities_em__coriolis                     | 10.856     | 2%  |
| __module_big_step_utilities_em__calc_cq                      | 10.536     | 2%  |
| __module_diffusion_em__cal_deform_and_div                    | 10.351     | 2%  |
| __module_small_step_em__small_step_prep                      | 9.724      | 1%  |
| __module_big_step_utilities_em__zero_tend                    | 9.021      | 1%  |
| __module_small_step_em__small_step_finish                    | 7.375      | 1%  |
| __module_em__rk_update_scalar                                | 7.205      | 1%  |
| __module_big_step_utilities_em__rk_rayleigh_damp             | 6.574      | 1%  |
| __module_big_step_utilities_em__w_damp                       | 6.466      | 1%  |
| __module_big_step_utilities_em__calc_p_rho_phi               | 5.578      | 1%  |
| __module_diffusion_em__vertical_diffusion_s                  | 5.475      | 1%  |
| __module_em__rk_addtend_dry                                  | 5.352      | 1%  |
| __module_diffusion_em__compute_diff_metrics                  | 5.075      | 1%  |
| __module_diffusion_em__calculate_n2                          | 4.293      | 1%  |
| __module_small_step_em__calc_coef_w                          | 3.185      | 0%  |
| __module_big_step_utilities_em__phy_prep                     | 2.826      | 0%  |
| __module_big_step_utilities_em__couple_momentum              | 2.589      | 0%  |
| __module_diffusion_em__horizontal_diffusion_v_2              | 2.376      | 0%  |
| __module_diffusion_em__vertical_diffusion_v_2                | 2.313      | 0%  |
| __module_big_step_utilities_em__calc_ww_cp                   | 2.257      | 0%  |
| __module_mp_wsm5__wsm5                                       | 2.074      | 0%  |
| __module_diffusion_em__horizontal_diffusion_w_2              | 2.005      | 0%  |
| __module_big_step_utilities_em__pg_buoy_w                    | 1.835      | 0%  |
| __module_diffusion_em__horizontal_diffusion_u_2              | 1.638      | 0%  |
| __module_big_step_utilities_em__moist_physics_prep_em        | 1.585      | 0%  |
| __module_diffusion_em__vertical_diffusion_u_2                | 1.549      | 0%  |
| __module_diffusion_em__cal_tau_11_22_33                      | 1.269      | 0%  |
|                                                              | 663.020    |     |

| Module Name                     | CPU<br>clocks |     |
|---------------------------------|---------------|-----|
| /root/Desktop/WRV3/main/wrf.exe | 304.998       | 88% |
| /lib/libm-2.7.so                | 27.441        | 8%  |
| /no-vmlinux                     | 15.954        | 5%  |
|                                 | 348.393       |     |

| Basic Block : ( Load / Store )                              | CPU<br>clocks |     |
|-------------------------------------------------------------|---------------|-----|
| __module_bc_set_physical_bc3d                               | 48.223        | 16% |
| __module_small_step_em_advance_uv                           | 31.155        | 11% |
| __module_small_step_em_advance_w                            | 25.142        | 9%  |
| __module_small_step_em_advance_mu_t                         | 23.116        | 8%  |
| __module_advect_em_advect_scalar                            | 16.063        | 5%  |
| __module_advect_em_advect_v                                 | 12.012        | 4%  |
| __module_advect_em_advect_u                                 | 11.684        | 4%  |
| __module_small_step_em_calc_p_rho                           | 11.325        | 4%  |
| __module_advect_em_advect_w                                 | 11.313        | 4%  |
| __module_big_step_utilities_em_horizontal_pressure_gradient | 9.765         | 3%  |
| __module_diffusion_em_horizontal_diffusion_s                | 9.291         | 3%  |
| __module_big_step_utilities_em_rhs_ph                       | 9.210         | 3%  |
| __module_big_step_utilities_em_curvature                    | 7.434         | 3%  |
| __module_small_step_em_sumflux                              | 6.908         | 2%  |
| __module_big_step_utilities_em_coriolis                     | 6.310         | 2%  |
| __module_diffusion_em_cal_deform_and_div                    | 6.015         | 2%  |
| __module_small_step_em_small_step_prep                      | 5.636         | 2%  |
| __module_small_step_em_small_step_finish                    | 4.299         | 1%  |
| __module_big_step_utilities_em_zero_tend                    | 4.221         | 1%  |
| __module_big_step_utilities_em_rk_rayleigh_damp             | 3.858         | 1%  |
| __module_big_step_utilities_em_w_damp                       | 3.778         | 1%  |
| __module_big_step_utilities_em_calc_p_rho_phi               | 3.189         | 1%  |
| __module_em_rk_addtend_dry                                  | 3.133         | 1%  |
| __module_diffusion_em_compute_diff_metrics                  | 2.993         | 1%  |
| __module_big_step_utilities_em_calc_cq                      | 2.627         | 1%  |
| __module_diffusion_em_calculate_n2                          | 2.320         | 1%  |
| __module_small_step_em_calc_coef_w                          | 1.902         | 1%  |
| __module_big_step_utilities_em_phy_prep                     | 1.796         | 1%  |
| __module_big_step_utilities_em_couple_momentum              | 1.569         | 1%  |
| __module_diffusion_em_vertical_diffusion_v_2                | 1.394         | 0%  |
| __module_diffusion_em_horizontal_diffusion_v_2              | 1.377         | 0%  |
| __module_big_step_utilities_em_calc_ww_cp                   | 1.336         | 0%  |
| __module_diffusion_em_horizontal_diffusion_w_2              | 1.134         | 0%  |
| __module_diffusion_em_vertical_diffusion_s                  | 1.107         | 0%  |
| __module_big_step_utilities_em_pg_buoy_w                    | 1.078         | 0%  |
|                                                             | 293.713       |     |

| gpu  |           | Xpovos msec |
|------|-----------|-------------|
| real | 7m42.723s | 462.723     |
| user | 7m2.804s  | 422.804     |
| sys  | 0m22.523s | 22.523      |

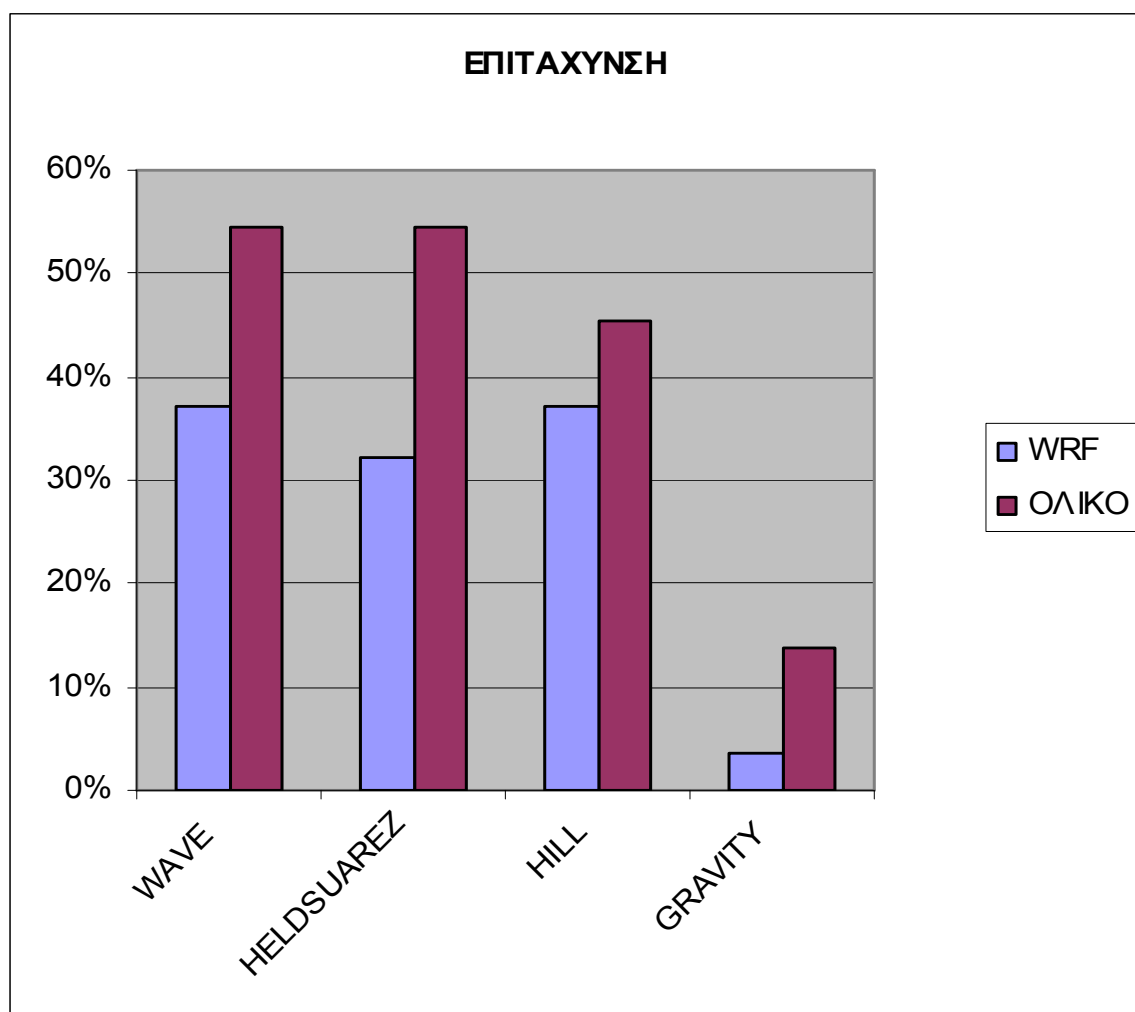
| Module Name                      | CPU clocks |     |
|----------------------------------|------------|-----|
| /root/Desktop/WRFV3/main/wrf.exe | 349.977    | 70% |
| /lib/libm-2.7.so                 | 131.559    | 27% |
| /no-vmLinux                      | 14.909     | 3%  |
|                                  | 496.445    |     |

| CS:EIP                                                      | CPU clocks |     |
|-------------------------------------------------------------|------------|-----|
| __module_mp_wsm5_wsm52d                                     | 105.872    | 31% |
| __module_advect_em_advect_scalar                            | 37.393     | 11% |
| __module_small_step_em_advance_w                            | 16.917     | 5%  |
| __module_small_step_em_advance_uv                           | 14.882     | 4%  |
| __module_small_step_em_advance_mu_t                         | 14.200     | 4%  |
| __module_bc_set_physical_bc3d                               | 12.715     | 4%  |
| __module_big_step_utilities_em_horizontal_diffusion         | 10.384     | 3%  |
| __module_small_step_em_calc_p_rho                           | 9.882      | 3%  |
| __module_advect_em_advect_u                                 | 9.548      | 3%  |
| __module_advect_em_advect_w                                 | 9.303      | 3%  |
| __module_advect_em_advect_v                                 | 8.728      | 3%  |
| __module_big_step_utilities_em_rhs_ph                       | 7.906      | 2%  |
| __module_big_step_utilities_em_horizontal_pressure_gradient | 7.172      | 2%  |
| __module_em_rk_update_scalar                                | 6.268      | 2%  |
| __module_small_step_em_sumflux                              | 5.960      | 2%  |
| __module_big_step_utilities_em_curvature                    | 5.679      | 2%  |
| __module_big_step_utilities_em_calc_cq                      | 5.150      | 2%  |
| __module_small_step_em_small_step_prep                      | 5.098      | 1%  |
| __module_diffusion_em_cal_deform_and_div                    | 4.958      | 1%  |
| __module_big_step_utilities_em_coriolis                     | 4.753      | 1%  |
| __module_small_step_em_small_step_finish                    | 4.129      | 1%  |
| __module_big_step_utilities_em_zero_tend                    | 3.801      | 1%  |
| __module_big_step_utilities_em_vertical_diffusion           | 3.573      | 1%  |
| __module_big_step_utilities_em_w_damp                       | 3.244      | 1%  |
| __module_big_step_utilities_em_calc_p_rho_phi               | 3.143      | 1%  |
| __module_em_rk_addtend_dry                                  | 2.991      | 1%  |
| __module_diffusion_em_calculate_n2                          | 2.677      | 1%  |
| __module_big_step_utilities_em_phy_prep                     | 2.257      | 1%  |
| __module_diffusion_em_compute_diff_metrics                  | 2.015      | 1%  |
| __module_small_step_em_calc_coef_w                          | 1.915      | 1%  |
| vssqrt_                                                     | 1.769      | 1%  |
| __module_big_step_utilities_em_horizontal_diffusion_3dmp    | 1.549      | 0%  |
| __module_big_step_utilities_em_pg_buoy_w                    | 1.373      | 0%  |
| vsrec_                                                      | 1.301      | 0%  |
| __module_big_step_utilities_em_couple_momentum              | 1.175      | 0%  |
| __module_big_step_utilities_em_calc_ww_cp                   | 1.139      | 0%  |
|                                                             | 340.819    |     |

| Module Name                      | CPU clocks |     |
|----------------------------------|------------|-----|
| /root/Desktop/WRFV3/main/wrf.exe | 192.806    | 86% |
| /lib/libm-2.7.so                 | 21.182     | 9%  |
| /no-vmlinux                      | 9.619      | 4%  |
|                                  | 223.607    |     |

| CS:EIP                                                      | CPU clocks |    |
|-------------------------------------------------------------|------------|----|
| __module_small_step_em_advance_w                            | 16.625     | 9% |
| __module_small_step_em_advance_uv                           | 14.997     | 8% |
| __module_small_step_em_advance_mu_t                         | 14.276     | 8% |
| __module_advect_em_advect_scalar                            | 13.100     | 7% |
| __module_bc_set_physical_bc3d                               | 10.800     | 6% |
| __module_advect_em_advect_u                                 | 9.564      | 5% |
| __module_advect_em_advect_w                                 | 9.316      | 5% |
| __module_advect_em_advect_v                                 | 8.768      | 5% |
| __module_small_step_em_calc_p_rho                           | 8.278      | 4% |
| __module_big_step_utilities_em_rhs_ph                       | 7.919      | 4% |
| __module_big_step_utilities_em_horizontal_pressure_gradient | 7.086      | 4% |
| __module_small_step_em_sumflux                              | 6.446      | 3% |
| __module_big_step_utilities_em_curvature                    | 5.624      | 3% |
| __module_big_step_utilities_em_horizontal_diffusion         | 5.283      | 3% |
| __module_small_step_em_small_step_prep                      | 5.180      | 3% |
| __module_diffusion_em_cal_deform_and_div                    | 4.998      | 3% |
| __module_big_step_utilities_em_coriolis                     | 4.758      | 3% |
| __module_small_step_em_small_step_finish                    | 4.178      | 2% |
| __module_big_step_utilities_em_w_damp                       | 3.252      | 2% |
| __module_big_step_utilities_em_calc_p_rho_phi               | 3.035      | 2% |
| __module_em_rk_addtend_dry                                  | 2.977      | 2% |
| __module_big_step_utilities_em_zero_tend                    | 2.880      | 2% |
| __module_big_step_utilities_em_phy_prep                     | 2.522      | 1% |
| __module_diffusion_em_calculate_n2                          | 2.124      | 1% |
| __module_big_step_utilities_em_calc_cq                      | 2.073      | 1% |
| __module_diffusion_em_compute_diff_metrics                  | 2.015      | 1% |
| __module_small_step_em_calc_coef_w                          | 1.925      | 1% |
| __module_big_step_utilities_em_horizontal_diffusion_3dmp    | 1.466      | 1% |
| __module_em_rk_update_scalar                                | 1.261      | 1% |
| __module_big_step_utilities_em_calc_ww_cp                   | 1.152      | 1% |
| __module_big_step_utilities_em_couple_momentum              | 1.126      | 1% |
| __module_big_step_utilities_em_pg_buoy_w                    | 1.012      | 1% |
|                                                             | 186.016    |    |

| gpu  |           |  | Χρονος<br>msec |
|------|-----------|--|----------------|
| real | 3m47.019s |  | 227.019        |
| user | 3m39.930s |  | 219.930        |
| sys  | 0m6.043s  |  | 6.043          |



# Βιβλιογραφία

## A.1. Άρθρα και Παρουσιάσεις

John Michalakes, Manish Vachharajani " *GPU Acceleration of Numerical Weather Prediction*", (2007)

John Michalakes, Josh Hacker, Richard Loft, Michael O. McCracken, Allan Snavely, Nicholas J. Wright, Tom Spelce, Brent Gorda and Robert Walkup "*WRF Nature Run*", (2007)

John Michalakes, Josh Hacker, Richard Loft, Michael O. McCracken, Allan Snavely, Nicholas J. Wright, Tom Spelce, Brent Gorda and Robert Walkup "*WRF Nature Run*", (2007)

John Michalakes "*Weather Research and Forecast (WRF) Model Port to Windows: Preliminary Report*" (2007)

John Michalakes "*High Performance Computing and Atmospheric Modeling*" Colorado State University, November 26, 2007.

Sergio Barrachina, Maribel Castillo, Francisco D. Igual, Rafael Mayo, Enrique S. Quintana-Ort<sup>1</sup>, Gregorio Quintana-Ort<sup>1</sup> "*Exploiting the capabilities of modern GPUs for dense matrix computations*" November 2008

Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W. Sheaffer, Kevin Skadron "*A Performance Study of General-Purpose - Applications on Graphics Processors Using CUDA*" University of Virginia, Department of Computer Science, Charlottesville, VA, USA

Dinesh Manocha & Ming C. Lin Naga Govindaraju "*GPGPU to Many Core Processing: Higher Performance for Mass Market Applications*" University of North Carolina at Chapel Hill Microsoft Corporation

Zhe Fan, Feng Qiu, Arie Kaufman, Suzanne Yoakum-Stover "*GPU Cluster for High Performance Computing*" Center For Visual Computing and Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400

Kalyan S. Perumalla "*Discrete-event Execution Alternatives on General Purpose Graphical Processing Units (GPGPUs)*" Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA

Dale Barker, "*Overview Of Recent Developments And Results From WRF-Var*" WRF-Var Development Team (ESSL/MMM Division), Data Assimilation Testbed Center (RAL/JNT), National Center For Atmospheric Research, 12th June 2007

Tom R. Halfhill "*Parallel Processing with CUDA*" (2008)

## **A2. Ιστοσελίδες**

“About The Weather Research and Forecasting Model” from  
<http://wrf-model.org/index.php>

“WRF Model Users Page” from  
<http://www.mmm.ucar.edu/wrf/users>

“CUDA Zone” from  
<http://nvidia.com/cuda>

“CUDA” from  
<http://en.wikipedia.org/wiki/CUDA>

“Τμήμα Φυσικής - infogenesis” from  
<http://www.infogenesis.gr>

“My Meteo – WRF information” from  
<http://www.mymeteo.gr/index>

“ Νέα προσθήκη - Wrf Model - Προγνωστικό μοντέλο ” from  
<http://www.meteoclub.gr/index>

“ NVIDIA Tesla to Improve Search for Oil and Gas” from  
<http://www.news.softpedia.com/news/NVIDIA-Tesla-to-Improve-Search-for-Oil-and-Gas-110500.shtml>

“nVidia Tesla processors boost oil industry” from  
<http://www.itpro.co.uk/604017/nvidia-tesla-processors-boost-oil-industry>

“NVIDIA Tesla High Performance Computing - GPUs Take a New Life” from  
<http://www.pcper.com/article.php?aid=424>

“nVidia Tesla Technology” from  
<http://www.hardwaresecrets.com/article/495>

“nVidia Tesla Technology” from  
<http://www.hardwaresecrets.com/article/495>

GPU Acceleration of NWP: Benchmark Kernels Web Page  
<http://www.mmm.ucar.edu/wrf/WG2/GPU/>