

# ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2012/2013

## Εισαγωγή

Στα πλαίσια της άσκησης θα ασχοληθούμε με το πρόβλημα της ομαδοποίησης δεδομένων που λαμβάνονται σε συνεχή ροή, όπως για παράδειγμα πολυμεσικά δεδομένα, χρηματοοικονομικές συναλλαγές, κλπ. Ο σκοπός είναι να κατασκευαστεί μια καλή ομαδοποίηση (ορίζοντας κάποια ή κάποιες μετρικές) με μικρό κόστος σε χρόνο και χώρο καθώς έρχονται συνεχώς νέα δεδομένα.

## Η εφαρμογή “Streamcluster”

Η εφαρμογή “Streamcluster” ανήκει στην κατηγορία των εφαρμογών Recognition, Mining and Synthesis (RMS) και λύνει το πρόβλημα της «Ομαδοποίησης σε πραγματικό χρόνο» (Online clustering problem [1]). Αναπτύχθηκε στο Πανεπιστήμιο του Princeton και είναι μέρος της σουίτας προγραμμάτων PARSEC [2]. Με λίγα λόγια, για μια συνεχόμενη ροή σημείων εισόδου η εφαρμογή υπολογίζει ένα προκαθορισμένο πλήθος από μέσους (medians) και αναθέτει κάθε σημείο εισόδου στον πλησιέστερο μέσο. Η ποιότητα της ομαδοποίησης μετριέται από το άθροισμα των τετραγώνων των αποστάσεων (Sum of Squared distances ή SSQ). Οι μέσοι και ποια στοιχεία ανήκουν σε κάθε μέσο καθορίζονται ώστε να ελαχιστοποιείται η μετρική SSQ.

Η ομαδοποίηση σε πραγματικό χρόνο αποτελεί βασική πράξη όταν μεγάλο πλήθος δεδομένων ή δεδομένα που παράγονται συνεχώς πρέπει να οργανωθούν σε ομάδες σε πραγματικό χρόνο, για παράδειγμα για την ανίχνευση εισβολής σε κάποιο δίκτυο (network intrusion detection), για την αναγνώριση προτύπων (pattern recognition) και την εξόρυξη δεδομένων (data mining). Το πρόγραμμα καταναλώνει τον περισσότερο χρόνο αξιολογώντας το πιθανό όφελος από την ανακήρυξη κάποιου νέου στοιχείου ως μέσο και κατά συνέπεια την επαναδιάταξη των στοιχείων σε ομάδες. Ο χρόνος εκτέλεσης της εφαρμογής εξαρτάται σε μεγάλο βαθμό από την απόδοση της μνήμης όταν τα δεδομένα εισόδου έχουν μικρό πλήθος διαστάσεων (μονοδιάστατα, δισδιάστατα), αλλά αρχίζει να εξαρτάται περισσότερο από την υπολογιστική ισχύ του επεξεργαστή καθώς οι διαστάσεις των δεδομένων εισόδου αρχίζουν να μεγαλώνουν. Τέλος, επειδή πρόκειται για αλγόριθμο πραγματικού χρόνου, το μέγεθος του συνόλου εργασίας (working set) του αλγορίθμου μπορεί να επιλεγεί ανεξάρτητα από τα δεδομένα εισόδου.

## Αναφορές

- [1] L. O’Callaghan, A. Meyerson, R. M. N. Mishra, and S. Guha. High-Performance Clustering of Streams and Large Data Sets. In Proceedings of the 18th International Conference on Data Engineering, February 2002.
- [2] <http://parsec.cs.princeton.edu>

## Ζητούμενα της άσκησης

Στα πλαίσια της άσκησης θα σας δοθεί μια σειριακή έκδοση της εφαρμογής “Streamcluster” γραμμένη σε C++. Σκοπός σας θα είναι:

- 1) Να αναλύσετε την συμπεριφορά της εφαρμογής, εντοπίζοντας τις συναρτήσεις του κώδικα που παίρνουν τον περισσότερο χρόνο για να εκτελεστούν.
- 2) Να παραλληλοποιήσετε το πρόγραμμα και να δημιουργήσετε δύο διαφορετικές παράλληλες εκδόσεις:
  - a. Μια με χρήση του προτύπου OpenMP.
  - b. Μια που θα χρησιμοποιεί την έκδοση με OpenMP και επιπλέον θα χρησιμοποιεί εντολές τύπου SIMD.

Θα πρέπει κατ’ αρχάς να αναλύσετε την απόδοση του σειριακού κώδικα και να δείτε σε ποιες συναρτήσεις καταναλώνεται ο περισσότερος χρόνος εκτέλεσης. Αυτές οι συναρτήσεις είναι προφανώς οι πρώτες υποψήφιες για παραλληλοποίηση. Για τον σκοπό αυτό θα χρησιμοποιήσετε το εργαλείο Scalasca, ένα εργαλείο ανάλυσης της απόδοσης εφαρμογών. Στο Παράρτημα Α της εκφώνησης παρατίθενται οδηγίες για την εγκατάσταση των εργαλείων που απαιτούνται στα πλαίσια της εργασίας. Αναλύστε την σειριακή εφαρμογή για τον χρόνο εκτέλεσης και προσπαθείστε να καταλάβετε γιατί οι συναρτήσεις αυτές δεν έχουν καλή απόδοση (π.χ., πολλά cache misses, TLB misses, απλά πολλές περισσότερες πράξεις σε σχέση με άλλες συναρτήσεις της εφαρμογής, κλπ) με χρήση των Performance Counters που προσφέρουν οι σύγχρονοι επεξεργαστές (δείτε το Παράρτημα Β για λεπτομέρειες). **Συμπεριλάβετε όλα τα συμπεράσματα σας στην αναφορά σας. Συμπεριλάβετε εικόνες (screenshots) από τις αναλύσεις που θα κάνετε με χρήση του εργαλείου Scalasca.**

Στην συνέχεια θα πρέπει να παραλληλοποιήσετε των κώδικα σας πρώτα με χρήση του προτύπου OpenMP. Ξεκινήστε από τις συναρτήσεις που εντοπίσατε νωρίτερα. Επιπλέον όμως είσαστε ελεύθεροι να παραλληλοποιήσετε τον κώδικα με όποιον τρόπο θέλετε. Αν θεωρήσετε πως μια οποιαδήποτε αναδιοργάνωση του κώδικα μπορεί να οδηγήσει σε καλύτερη παραλληλοποίηση και απόδοση είστε ελεύθεροι να το κάνετε. Ωστόσο, θεωρείται αυτονόητο πως η υλοποίηση θα πρέπει να είναι σωστή (να δίνει **πάντα** τα ίδια αποτελέσματα με την ακολουθιακή έκδοση του προγράμματος) και πως **στην αναφορά που θα παραδώσετε θα πρέπει να αιτιολογήσετε γιατί παραλληλοποιήσατε με τον συγκεκριμένο τρόπο την εφαρμογή.**

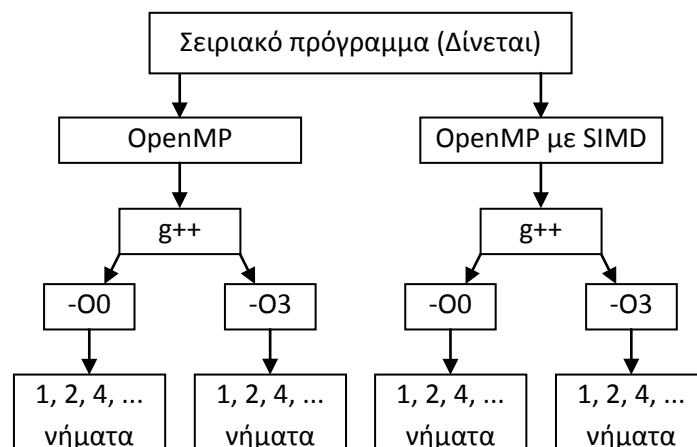
Τέλος, χρησιμοποιώντας ως βάση την έκδοση του προγράμματος με χρήση του OpenMP, θα δημιουργήσετε μια ακόμα παράλληλη έκδοση του προγράμματος, προσθέτοντας **επιπλέον** εντολές τύπου SIMD. Το Παράρτημα Γ έχει αναφορές που μπορείτε να χρησιμοποιήσετε για να δείτε πως αξιοποιούνται οι εντολές αυτές στα πλαίσια ενός προγράμματος. **Συμπεριλάβετε στην αναφορά τις παρατηρήσεις σας για τις αλλαγές στους χρόνους εκτέλεσης από την εισαγωγή των εντολών SIMD.**

Οδηγίες για την μεταγλώττιση και εκτέλεση του σειριακού κώδικα που σας δίνεται και του παράλληλου κώδικα που θα φτιάξετε δίνονται στο Παράρτημα Β της εκφώνησης. Εκεί επίσης αναφέρεται για ποιες παραμέτρους θα πρέπει να τρέξετε την κάθε έκδοση του προγράμματος και ποια αποτελέσματα θα συμπεριλάβετε στην αναφορά σας.

Συγκεντρωτικά, θα πρέπει στην αναφορά σας να παραθέσετε μετρήσεις και να σχολιάσετε τις παρακάτω περιπτώσεις:

- 1) Πρόγραμμα με OpenMP
- 2) Πρόγραμμα με OpenMP και εντολές SIMD.
- 3) Μεταγλώττιση και εκτέλεση κάθε προγράμματος χωρίς βελτιστοποιήσεις (-O0) και με μέγιστες βελτιστοποιήσεις (-O3).
- 4) Εκτέλεση κάθε περίπτωσης με 1, 2 και 4 νήματα τουλάχιστον. Αν το σύστημα σας διαθέτει περισσότερους πυρήνες, ακόμα καλύτερα!

Για κάθε έναν από τους παραπάνω συνδυασμούς συμπεριλάβετε στην αναφορά σας πίνακες με τους χρόνους εκτέλεσης του βασικού αλγόριθμου (υπάρχει έτοιμο στον κώδικα που σας δίνεται) και διαγράμματα της χρονοβελτίωσης. Συγκεντρωτικά, όλες οι περιπτώσεις που θα πρέπει να συμπεριλάβετε φαίνονται στο παρακάτω σχήμα:



## Διαδικαστικά

Η εργασία θα πρέπει να γίνει σε ομάδες των 3 ή 4 ατόμων **ακριβώς**. Δεν θα γίνει δεκτή ομάδα με λιγότερα ή περισσότερα άτομα **για κανέναν λόγο**. Αν δεν συμπληρώσετε ομάδα 3 ή 4 ατόμων, τότε θα προστεθούν άτομα στην ομάδα σας από εμάς.

Ένα άτομο από κάθε ομάδα θα αναλάβει να δηλώσει την ομάδα του μέχρι την **Τετάρτη, 10/04/2013 και ώρα 23:59:59**. Το άτομο αυτό θα είναι επίσης υπεύθυνο για όλη την επικοινωνία της ομάδας μαζί μας, καθ' όλη την διάρκεια του εξαμήνου και μέχρι την παράδοση της άσκησης. Η ομάδα θα δηλωθεί μέσω e-mail στην διεύθυνση [iev@hpclab.ceid.upatras.gr](mailto:iev@hpclab.ceid.upatras.gr). Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, το e-mail θα πρέπει να έχει τον εξής τίτλο (χωρίς κενά στο τμήμα [ParPro12-13]):

[ParPro12-13] Δήλωση ομάδας

Το περιεχόμενο του e-mail θα πρέπει να είναι ο Α.Μ. και το ονοματεπώνυμο των μελών της ομάδας. **Κάθε επικοινωνία μαζί μας θα πρέπει να απευθύνεται στην προαναφερθείσα διεύθυνση e-mail και ο τίτλος να ξεκινάει με [ParPro12-13].**

## Παραδοτέα

Η άσκηση θα πρέπει να παραδοθεί μέχρι την **Παρασκευή, 19/05/2013 και ώρα 23:59:59** και θα αποτελείται από μια γραπτή αναφορά και τον κώδικα της άσκησης που θα αναπτύξετε. Τα παραπάνω θα παραδωθούν μέσω e-mail στην διεύθυνση [iev@hpclab.ceid.upatras.gr](mailto:iev@hpclab.ceid.upatras.gr). Ο τίτλος θα πρέπει να έχει την μορφή:

[ParPro12-13] Παράδοση Άσκησης-Ομάδα ΧΧ

Ο αριθμός της ομάδας σας θα σας γνωστοποιηθεί μετά την ολοκλήρωση των δηλώσεων των ομάδων και θα πρέπει να αντικαταστήσετε τα "ΧΧ" στον τίτλο με τον αριθμό της ομάδας.

Στην αναφορά **δεν** θα πρέπει να περιλαμβάνεται επεξήγηση του ακολουθιακού αλγόριθμου που σας δώθηκε. Επικεντρωθείτε στην επεξήγηση της παραλληλοποίησης που κάνατε, στις μετρήσεις σας και στα διαγράμματα που θα προσθέσετε. Σχολιάστε τις διαφορές από την χρήση βελτιστοποιήσεων στον χρόνο εκτέλεσης της εφαρμογής και (κυρίως) στην χρονοβελτίωση. Γενικότερα, δώστε ιδιαίτερο βάρος στην αναφορά σε αυτά που κάνατε εσείς.

Η άσκηση θα μετρήσει στην τελική βαθμολογία ως εξής:

- 1) 30% για όσους την παραδώσουν τον Ιούνιο. Το υπόλοιπο 70% θα είναι η τελική εξέταση.
- 2) 20% για όσους την παραδώσουν τον Σεπτέμβριο (ακριβής ημερομηνία θα ανακοινωθεί στο μέλλον). Το υπόλοιπο 80% θα είναι η τελική εξέταση.
- 3) Για τους επί πτυχίω φοιτητές που θα δώσουν γραπτή εξέταση κατά την άτυπη εξεταστική του Φεβρουαρίου 2014, το ποσοστό της άσκησης θα είναι 20%, ανεξάρτητα από το πότε θα έχουν παραδώσει την άσκηση (Ιούνιο ή Σεπτέμβριο). Το υπόλοιπο 80% θα είναι η τελική εξέταση. Δεν θα δοθεί δυνατότητα παράδοσης της άσκησης μετά τον Σεπτέμβριο.

## **Παράρτημα Α**

Για την εγκατάσταση των προγραμμάτων που θα χρειαστείτε στα πλαίσια της άσκησης, θα πρέπει να ακολουθήσετε τις παρακάτω οδηγίες. Σε περίπτωση που κάτι δεν πάει καλά (ιδιαίτερα για τα πακέτα που θα πρέπει να εγκαταστήσετε μόνοι σας), διαβάστε καλά τα μηνύματα λάθους. Είναι πολύ πιθανό να πρέπει να εγκαταστήσετε κάποιο επιπλέον πακέτο στην διανομή Linux, για να μπορέσουν να μεταγλωττιστούν και να εγκατασταθούν τα υπόλοιπα πακέτα.

### 1) Εγκατάσταση Linux

Για να εγκαταστήσετε μια διανομή Linux θα πρέπει πρώτα να αποφασίσετε ποια διανομή θέλετε. Στα πλαίσια της εργασίας συστήνεται να εγκαταστήσετε μια από τις Ubuntu (<http://www.ubuntu.com>), Fedora (<http://fedoraproject.org>) ή openSUSE (<http://www.opensuse.org>). Φυσικά μπορείτε να εγκαταστήσετε όποια άλλη διανομή θέλετε. Σε κάθε περίπτωση επιλέξτε την τελευταία έκδοση που είναι διαθέσιμη για κάθε μια. Αφού κατεβάσετε την διανομή θα πρέπει να την γράψετε σε ένα CD/DVD/USB stick. Ακολουθήστε τις αντίστοιχες οδηγίες της διανομής που επιλέξατε. Στην συνέχεια, η εγκατάσταση της διανομής μπορεί να γίνει με δύο τρόπους:

#### a. Απευθείας στον σκληρό δίσκο του μηχανήματος σας:

Για να μπορέσετε να ακολουθήσετε την μέθοδο αυτή θα πρέπει να φτιάξετε ένα ξεχωριστό partition στον σκληρό σας δίσκο. Όταν ξεκινήσετε την εγκατάσταση της διανομής θα πρέπει να επιλέξετε το συγκεκριμένο partition. Ακολουθήστε αυτή τη μέθοδο αν θέλετε να κρατήσετε την διανομή που θα εγκαταστήσετε και για αργότερα.

#### b. Μέσω Virtual Machine:

“Virtual Machine” είναι στην πραγματικότητα οποιοδήποτε πρόγραμμα που μπορούμε να εγκαταστήσουμε σε ένα λειτουργικό σύστημα που διαθέτουμε και προσομοιώνει έναν υπολογιστή. Αν για παράδειγμα διαθέτουμε Windows, μπορούμε να εγκαταστήσουμε ένα Virtual Machine και στην συνέχεια να εγκαταστήσουμε στον προσομοιούμενο υπολογιστή ένα άλλο λειτουργικό σύστημα, όπως για παράδειγμα Linux.

Αν ακολουθήσετε αυτή τη μέθοδο θα πρέπει κατ’ αρχάς να κατεβάσετε και να εγκαταστήσετε στο λειτουργικό σύστημα που διαθέτετε ένα Virtual Machine. Υπάρχουν αρκετά ελεύθερα προγράμματα διαθέσιμα για αυτό, όπως το VMware Player (<http://www.vmware.com/products/player/overview.html>), το Virtual Box (<https://www.virtualbox.org>) και το QEMU (<http://wiki.qemu.org>). Επιλέξτε ένα και εγκαταστήστε το. Επειδή θα χρειαστεί να τρέξουμε παράλληλα προγράμματα, διαβάστε τις οδηγίες εγκατάστασης και δημιουργίας Virtual Machine, ώστε η Virtual Machine στην οποία θα εγκαταστήσετε την διανομή Linux που θα επιλέξετε να υποστηρίζει πολλαπλούς πυρήνες. Τέλος, εγκαταστήστε την διανομή Linux που επιλέξατε στην Virtual Machine που φτιάξατε.

### 2) Εγκατάσταση PAPI-5.1.0.2

Η βιβλιοθήκη PAPI δίνει την δυνατότητα πρόσβασης στους Performance Counters που διαθέτουν οι σύγχρονοι επεξεργαστές. Οι Performance Counters είναι ειδικόι καταχωρητές που μετράνε ένα πλήθος χαρακτηριστικών μιας εφαρμογής που εκτελεί ο χρήστης, όπως για παράδειγμα L1 cache misses ή hits, L2 cache misses ή hits, TLB misses, κλπ. Το τι θα καταμετρηθεί μπορεί να το καθορίσει ο χρήστης. Έτσι δίνεται η

δυνατότητα να αναλυθεί με ακρίβεια η απόδοση μιας εφαρμογής και να καθοριστεί σε ποια σημεία της και γιατί δεν έχει την καλύτερη δυνατή απόδοση.

Αν και η βιβλιοθήκη PAPI μπορεί να χρησιμοποιηθεί απευθείας στον πηγαίο κώδικα μιας εφαρμογής με την χρήση του API (Application Programming Interface) που προσφέρει, στα πλαίσια της εργασίας θα χρησιμοποιηθεί μόνο σε συνδυασμό με την αυτοματοποιημένη δυνατότητα αξιοποίησης της από το εργαλείο Scalasca (βλέπε παρακάτω).

Αν και πολλές διανομές Linux προσφέρουν ως πακέτο την βιβλιοθήκη αυτή, συνιστάται να την εγκαταστήσετε μόνοι σας από τον πηγαίο της κώδικα, ακολουθώντας τις οδηγίες που παρατίθενται αμέσως μετά. Ο λόγος είναι πως οι περισσότερες διανομές παρέχουν παλιότερες εκδόσεις της βιβλιοθήκης και είναι πολύ πιθανό νεότεροι επεξεργαστές να μην αναγνωρίζονται από την βιβλιοθήκη και να μην μπορείτε να πάρετε μετρήσεις.

Για να εγκαταστήσετε μόνοι σας την βιβλιοθήκη PAPI:

- a. Μεταβείτε στην ιστοσελίδα <http://icl.cs.utk.edu/papi> και στο μενού αριστερά μεταβείτε στο "Software". Κατεβάστε την τρέχουσα έκδοση της βιβλιοθήκης (5.1.0.2).
- b. Πριν την εγκατάσταση **αλλά και πριν από κάθε μέτρηση με χρήση της βιβλιοθήκης** θα πρέπει να απενεργοποιείτε το NMI watchdog:

```
echo 0 > /proc/sys/kernel/nmi_watchdog
```

Κάθε φορά που κάνετε επανεκκίνηση του Linux που έχετε εγκαταστήσει, θα πρέπει να εκτελείται την παραπάνω εντολή.

- c. Αποσυμπιέστε τον πηγαίο κώδικα της βιβλιοθήκης και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd papi-5.1.0.2.tar.gz | tar -x  
cd papi-5.1.0/src
```

- d. **Υποθέτουμε για την συνέχεια πως θα εγκαταστήσετε όλα τα πακέτα στον κατάλογο "opt" του Home Directory του λογαριασμού σας στο Linux που έχετε εγκαταστήσει. Αν αλλάξετε τους καταλόγους εγκατάστασης, θα πρέπει να προσαρμόσετε αντίστοιχα όλες τις εντολές από αυτό το σημείο και πέρα.**

Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης, μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
./configure --prefix=${HOME}/opt/papi-5.1.0.2  
make  
make install
```

Για να δείτε ποιες μετρήσεις με χρήση των Performance Counters είναι διαθέσιμες στο σύστημα σας, μπορείτε να εκτελέσετε την εντολή:

```
${HOME}/opt/papi-5.1.0.2/bin/papi_avail
```

### 3) Εγκατάσταση OPARI2-1.0.7

Άλλη μια βιβλιοθήκη που χρειάζεται για την εγκατάσταση του Scalasca. Αυτή δεν δίνεται σαν πακέτο από τις διανομές Linux και θα πρέπει να εγκατασταθεί.

- a. Μεταβείτε στην ιστοσελίδα <http://www.vi-hps.org/projects/score-p> και κοντά στο τέλος της σελίδας ακολουθήστε τον σύνδεσμο για να κατεβάσετε την τρέχουσα έκδοση της βιβλιοθήκης (1.0.7).
- b. Αποσυμπιέστε τον πηγαίο κώδικα της βιβλιοθήκης και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd opari2-1.0.7.tar.gz | tar -x
cd opari2-1.0.7
```

- c. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης, μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
./configure --prefix=${HOME}/opt/opari2-1.0.7
make
make install
```

#### 4) Εγκατάσταση Scalasca-1.4.3

Φτάνουμε έτσι στην εγκατάσταση του κυρίως εργαλείου που θα χρειαστούμε στα πλαίσια της εργασίας, το Scalasca-1.4.3. Εκτός από το ίδιο το πρόγραμμα, μπορείτε να κατεβάσετε από το site που δίνεται παρακάτω και το “User’s Guide”, το οποίο περιέχει πιο αναλυτικά τον τρόπο χρήσης του εργαλείου.

- a. Μεταβείτε στην ιστοσελίδα <http://www.scalasca.org> και στο μενού που εμφανίζεται στην αρχή της σελίδας επιλέξτε το “Download”. Κατεβάστε την τρέχουσα έκδοση της εφαρμογής (1.4.3). Στο μενού που εμφανίζεται αριστερά μπορείτε να μεταβείτε στο “Documentation” για να κατεβάσετε και το User’s Guide.
- b. Βεβαιωθείτε πως έχετε εγκαταστήσει στην διανομή Linux που επιλέξατε το πακέτο που περιέχει το πρόγραμμα “qmake” (ή “qmake-qt4” όπως το ονομάζουν μερικές διανομές). **Ανάλογα με το που είναι εγκατεστημένο το συγκεκριμένο πρόγραμμα προσαρμόστε την αντίστοιχη παράμετρο στην μεθεπόμενη οδηγία.**
- c. Αποσυμπιέστε τον πηγαίο κώδικα της εφαρμογής και μεταβείτε στο κατάλογο που θα δημιουργηθεί:

```
gzip -cd scalasca-1.4.3.tar.gz | tar -x
cd scalasca-1.4.3
```

- d. Ρυθμίστε τις παραμέτρους εγκατάστασης της βιβλιοθήκης:

```
./configure --prefix=${HOME}/opt/scalasca-1.4.3
--with-opari2=${HOME}/opt/opari2-1.0.7
--with-papi=${HOME}/opt/papi-5.1.0.2
--with-qmake=/usr/bin/qmake-qt4 --disable-mpi
```

- e. Μεταβείτε στον κατάλογο “build-linux-gomp-nompi”:

```
cd build-linux-gomp-nompi
```

- f. Με έναν κειμενογράφο ανοίξτε το αρχείο “Makefile.defs”, βρείτε την γραμμή που ξεκινάει με “BFDLIB” και προσθέστε στο τέλος της το κείμενο “-ldl” (Χωρίς τα εισαγωγικά, με το κενό πριν την παύλα).

g. Μεταγλωττίστε και εγκαταστήστε την βιβλιοθήκη:

```
make  
make install
```

h. Προσθέστε στην μεταβλητή περιβάλλοντος PATH τον κατάλογο εγκατάστασης:

```
export PATH=${HOME}/opt/scalasca-1.4.3/bin:$PATH
```

5) Αρχεία/Κατάλογοι που δεν χρειάζονται μετά την εγκατάσταση

Αφού εγκαταστήσετε τα παραπάνω εργαλεία μπορείτε να σβήσετε τους καταλόγους που δημιουργήθηκαν μετά την αποσυμπίεση των αρχείων που κατεβάσατε.

**ΠΡΟΣΟΧΗ: Μην σβήσετε τους καταλόγους στους οποίους εγκαταστήσατε τα εργαλεία (\$HOME/opt).**



## Παράρτημα Β

- 1) Για την εκτέλεση του σειριακού προγράμματος που σας δίνεται και των παράλληλων προγραμμάτων που θα φτιάξετε απαιτείται το πέρασμα παραμέτρων από την γραμμή εντολής. Κατά την διάρκεια ανάπτυξης και αποσφαλμάτωσης των προγραμμάτων σας μπορείτε να χρησιμοποιείται τις παρακάτω παραμέτρους, οι οποίες οδηγούν σε σχετικά μικρούς χρόνους εκτέλεσης:

```
10 20 32 4096 4096 1000 none output_small.txt
10 20 64 8192 8192 1000 none output_medium.txt
10 20 128 16384 16384 1000 none output_large.txt
```

Με αυτές μπορείτε να ελέγχετε γρήγορα αν οι αλλαγές που κάνετε για την παραλληλοποίηση είναι σωστές. Για τις μετρήσεις που θα περιλαμβάνονται στην αναφορά σας θα χρησιμοποιήσετε τις παρακάτω παραμέτρους, που απαιτούν περισσότερο χρόνο:

```
10 20 256 32768 32768 1000 none output_small.txt
10 20 512 65536 65536 1000 none output_medium.txt
10 20 1024 131072 131072 1000 none output_large.txt
```

Μόλις μεταγλωττίσετε το σειριακό πρόγραμμα (βλέπε παρακάτω) μπορείτε να το τρέξετε χωρίς παραμέτρους για να δείτε μια σύντομη περιγραφή των παραμέτρων που χρησιμοποιούμε.

- 2) Για την μεταγλώττιση του σειριακού προγράμματος που σας δίνεται μπορείτε να χρησιμοποιήσετε την παρακάτω εντολή:

```
g++ -O3 (ή -O0) -Wall -Wextra -o streamcluster streamcluster.cpp
```

Αυτή θα δημιουργήσει ένα εκτελέσιμο αρχείο με το όνομα “streamcluster”, το οποίο μπορείτε να εκτελέσετε με την εντολή:

```
./streamcluster <Παράμετροι γραμμής εντολής>
```

Οι παράμετροι γραμμής εντολής είναι κάποιο από τα σύνολα που δώθηκαν παραπάνω. **Συνιστάται να τρέξετε την σειριακή εφαρμογή με όλα τα παραπάνω σύνολα παραμέτρων (ορίζοντας το κατάλληλο όνομα αρχείου εξόδου κάθε φορά), ώστε να έχετε τα αποτελέσματα σαν αναφορά για τις αλλαγές που θα κάνετε κατά την παραλληλοποίηση. Μην ξεχνάτε πως θα πρέπει να παίρνετε τα ίδια αποτελέσματα από το σειριακό και το παράλληλο πρόγραμμα!**

- 3) Για την μεταγλώττιση του παράλληλου προγράμματος με OpenMP που θα φτιάξετε μπορείτε να χρησιμοποιήσετε την παρακάτω εντολή (θεωρώντας πως το όνομα του αρχείου που φτιάξατε είναι “streamcluster\_omp.cpp”):

```
g++ -O3 (ή -O0) -fopenmp -Wall -Wextra -o streamcluster_omp
streamcluster_omp.cpp
```

Αυτή θα δημιουργήσει ένα εκτελέσιμο αρχείο με το όνομα “streamcluster\_omp”. Μπορείτε να ορίσετε το πλήθος των νημάτων που θα δημιουργούνται σε κάθε παράλληλη περιοχή και στην συνέχεια να το εκτελέσετε με τις εντολές:

```
Export OMP_NUM_THREADS=<Πλήθος νημάτων ανά παράλληλη περιοχή>
./streamcluster_omp <Παράμετροι γραμμής εντολής>
```

4) Για να μεταγλωττίσετε, να τρέξετε και να δείτε μετρήσεις με χρήση του εργαλείου Scalasca, πρέπει να δώσετε αντίστοιχα τις παρακάτω εντολές:

a. Μεταγλώττιση:

```
scalasca -instrument g++ -O3 (ή -O0) -Wall -Wextra -o  
streamcluster streamcluster.cpp
```

b. Πρέπει να θέσετε την μεταβλητή περιβάλλοντος "EPK\_TRACE":

```
export EPK_TRACE=1
```

c. Πρέπει να θέσετε την μεταβλητή περιβάλλοντος "ELG\_BUFFER\_SIZE":

```
export ELG_BUFFER_SIZE=1500000000
```

d. Αν θέλετε να χρησιμοποιήσετε τους Performance Counters πρέπει να θέσετε την μεταβλητή περιβάλλοντος "EPK\_METRICS":

```
export EPK_METRICS=<Τιμή από το πρόγραμμα pari_avail>
```

**Για την λειτουργία των παραπάνω (και πολλών άλλων) μεταβλητών περιβάλλοντος που επηρεάζουν την εκτέλεση του Scalasca μπορείτε να δείτε το Appendix B στο User's Guide του Scalasca.**

e. Εκτέλεση:

```
scalasca -analyze ./streamcluster <Παράμετροι γραμμής  
εντολής>
```

Η παραπάνω εντολή εκτελεί το πρόγραμμα και δημιουργεί έναν νέο κατάλογο που περιέχει αρχεία με τα αποτελέσματα των μετρήσεων του Scalasca. Το όνομα του καταλόγου αυτού χρησιμοποιείται στην επόμενη εντολή.

f. Εμφάνιση αποτελεσμάτων:

```
scalasca -examine <Όνομα καταλόγου>
```

## Παράρτημα Γ

- 1) Μπορείτε να βρείτε το manual του gcc στην ιστοσελίδα <http://gcc.gnu.org/onlinedocs>. Εκτελώντας την εντολή “gcc -v” βλέπετε ποια έκδοση του gcc έχετε εγκατεστημένη. Δείτε το αντίστοιχο manual από την παραπάνω ιστοσελίδα. Οι πληροφορίες για τις εντολές τύπου SIMD βρίσκονται στην παράγραφο με τίτλο “X86 Built-in Functions”. Προσέξτε τις παραμέτρους που μπορεί να χρειαστείτε όταν θα μεταγλωτίζετε το πρόγραμμα σας (π.χ., -msse2).
- 2) [http://www.360doc.com/content/12/0327/11/1016783\\_198217072.shtml](http://www.360doc.com/content/12/0327/11/1016783_198217072.shtml)
- 3) <http://www.cs.uiuc.edu/class/sp07/cs232/section/Discussion13/disc13.pdf>
- 4) <http://www.linuxjournal.com/content/introduction-gcc-compiler-intrinsics-vector-processing> (Δείτε στην τελευταία σελίδα τις αναφορές)
- 5) [http://www.cs.fsu.edu/~xyuan/cis4930-cda5125/lect4\\_simd\\_sse.pptx](http://www.cs.fsu.edu/~xyuan/cis4930-cda5125/lect4_simd_sse.pptx)
- 6) <http://ds9a.nl/gcc-simd>