

Παράλληλος Προγραμματισμός 2020

Προγραμματιστική Εργασία #1

Αποτελέσματα no-sse και sse κώδικα

	N=12, M=12	N=252, M=252	N=500, M=500	N=700, M=700	N=1000, M=1000
no-sse	Time: 0.000001 mFlops/s: 301.9898888	Time: :0.000169 mFlops/s: 751.354249	Time: 0.000709 mFlops/s: 701.176769	Time: 0.001475 mFlops/s: 664.363653	Time: 0.002750 mFlops/s: 727.293914
sse	Time: 0.000001 mFlops/s: 92.919966	Time: 0.004287 mFlops/s: 29.626281	Time: 0.005080 mFlops/s: 98.425494	Time: 0.009493 mFlops/s: 103.235331	Time: 0.018424 mFlops/s: 108,553859

Συμπέρασμα

Ο κώδικας χωρίς τις sse εντολές μου βγήκε αρκετά πιο γρήγορος, πράγμα που σημαίνει ότι κάτι δεν πήγε καλά στον sse κώδικα. Προσπάθησα να να φορτώσω τις τιμές που μας ενδιαφέρουν για τον υπολογισμό των pixel σε κάθε επανάληψη με έναν πίνακα τον num, και μετά φόρτωσα κάθε τιμή σε μια μεταβλητή __m128 ώστε να μπορέσουν να πραγματοποιηθούν οι προσθέσεις και το τελικό σύνολο να πάει στο τρέχον κελί του β πίνακα.

Επίσης δοκίμασα να το κάνω και με την εντολή _mm_set1_ps παρόλα αυτά το αποτέλεσμα δεν είχε διαφορά.

```
get_walltime(&ts);
for (i=1;i<N-1;i++) {
    for (int j=1;j<M-1;j+=4) {
        *vb = _mm_set1_ps(a[M*i+j]*K4 + a[(i-1)*M+(j-1)]*K + a[(i-1)*M+j]*K + a[(i-1)*M+(j+1)]*K + a[M
        vb++;
        va++;
    }
}
get_walltime(&te);
```