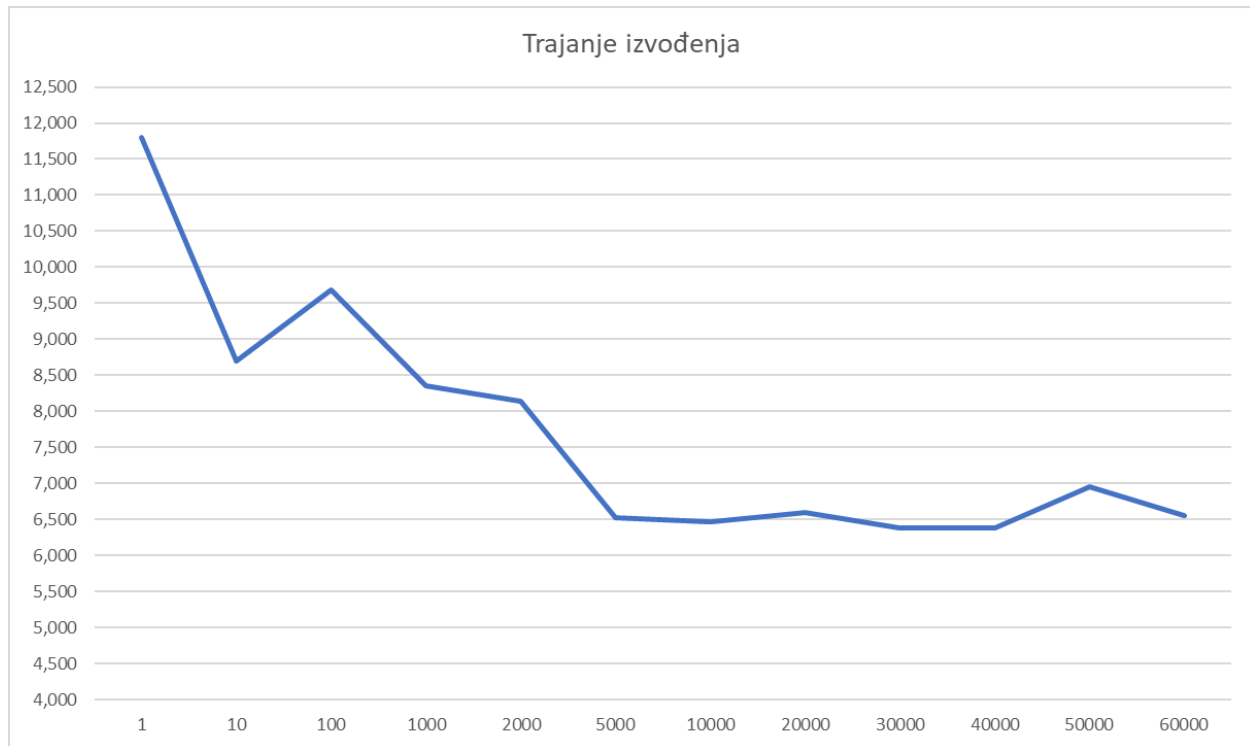


Laboratorijska vježba br. 3

Izvorni kod je dostupan na [Github repozitoriju](#).

Prva optimizacija jest generalna organizacija koda. Bitnija optimizacija jest zamjena običnih paraleliziranih petlji sa **task** mehanizmom, gdje pomoću **grain size** direktive određujemo broj iteracija dodijeljenih individualnim taskovima.



Na slici vidimo kako trajanje izvođenja pada s veličinom taskova, dok ne konvergira oko **6,5 sekundi**. S obzirom da smo u prethodnoj laboratorijskoj vježbi na istoj konfiguraciji (*16 jezgri sa 2048MB radne memorije*) uočili trajanje od 7,384 sekunde, vidimo **ubrzanje faktora 1,14**.



LAB03

```
void parallelMapRGBtoYUV() {  
    #pragma omp parallel  
    {  
        #pragma omp single  
        {  
            #pragma omp taskloop grainsize(GRAIN_SIZE_NUM)  
            for (int i = 0; i < TOTAL_PIXELS; ++i) {  
                {  
                    mapRGBtoYUV(i);  
                }  
            }  
        }  
    }  
}
```



LAB02

```
void parallelMapRGBtoYUV() {  
    #pragma omp parallel for  
    for (int i = 0; i < TOTAL_PIXELS; ++i) {  
        {  
            mapRGBtoYUV(i);  
        }  
    }  
}
```