Exercises from third DLI module

Workshop
"Fundamentals of Accelerated Computing with CUDA C/C++"

Departure point for the profiler: `addVectorsInto`

```
cudaMallocManaged(&a, size);
cudaMallocManaged(&b, size);
cudaMallocManaged(&c, size);

initWith(3, a, N);
initWith(4, b, N);
initWith(0, c, N);

addVectorsInto<<<numberOfBlocks, threadsPerBlock>>>(c, a, b, N);

checkElementsAre(7, c, N);

cudaFree(a);
cudaFree(b);
cudaFree(c);
```

Exercise 1:
Compare with a prefetch in kernel

```
cudaMallocManaged(&a, size);
cudaMallocManaged(&b, size);
cudaMallocManaged(&c, size);

initWith(3, a, N);
initWith(4, b, N);
initWith(0, c, N);

cudaMemPrefetchAsync(a, size, deviceId);
cudaMemPrefetchAsync(b, size, deviceId);
cudaMemPrefetchAsync(c, size, deviceId);
addVectorsInto<<<numberOfBlocks, threadsPerBlock>>>(c, a, b, N);

checkElementsAre(7, c, N);

cudaFree(a);
cudaFree(b);
cudaFree(c);
```

Exercise 2: Compare with initialize on GPU (and prefetch earlier)

```
cudaMallocManaged(&a, size);
cudaMallocManaged(&b, size);
cudaMallocManaged(&c, size);
```

```
cudaMemPrefetchAsync(a, size, deviceId);
cudaMemPrefetchAsync(b, size, deviceId);
cudaMemPrefetchAsync(c, size, deviceId);
initWith<<<numberOfBlocks, threadsPerBlock>>>(3, a, N);
initWith<<<numberOfBlocks, threadsPerBlock>>>(4, b, N);
initWith<<<numberOfBlocks, threadsPerBlock>>>(0, c, N);
cudaDeviceSynchronize();
```

```
addVectorsInto<<<numberOfBlocks, threadsPerBlock>>>(c, a, b, N);
```

```
checkElementsAre(7, c, N);
```

```
cudaFree(a);
cudaFree(b);
cudaFree(c);
```

Exercise 3: Compare with a GPU initialization and prefetch when sending data back to CPU

```
cudaMallocManaged(&a, size);
cudaMallocManaged(&b, size);
cudaMallocManaged(&c, size);
```

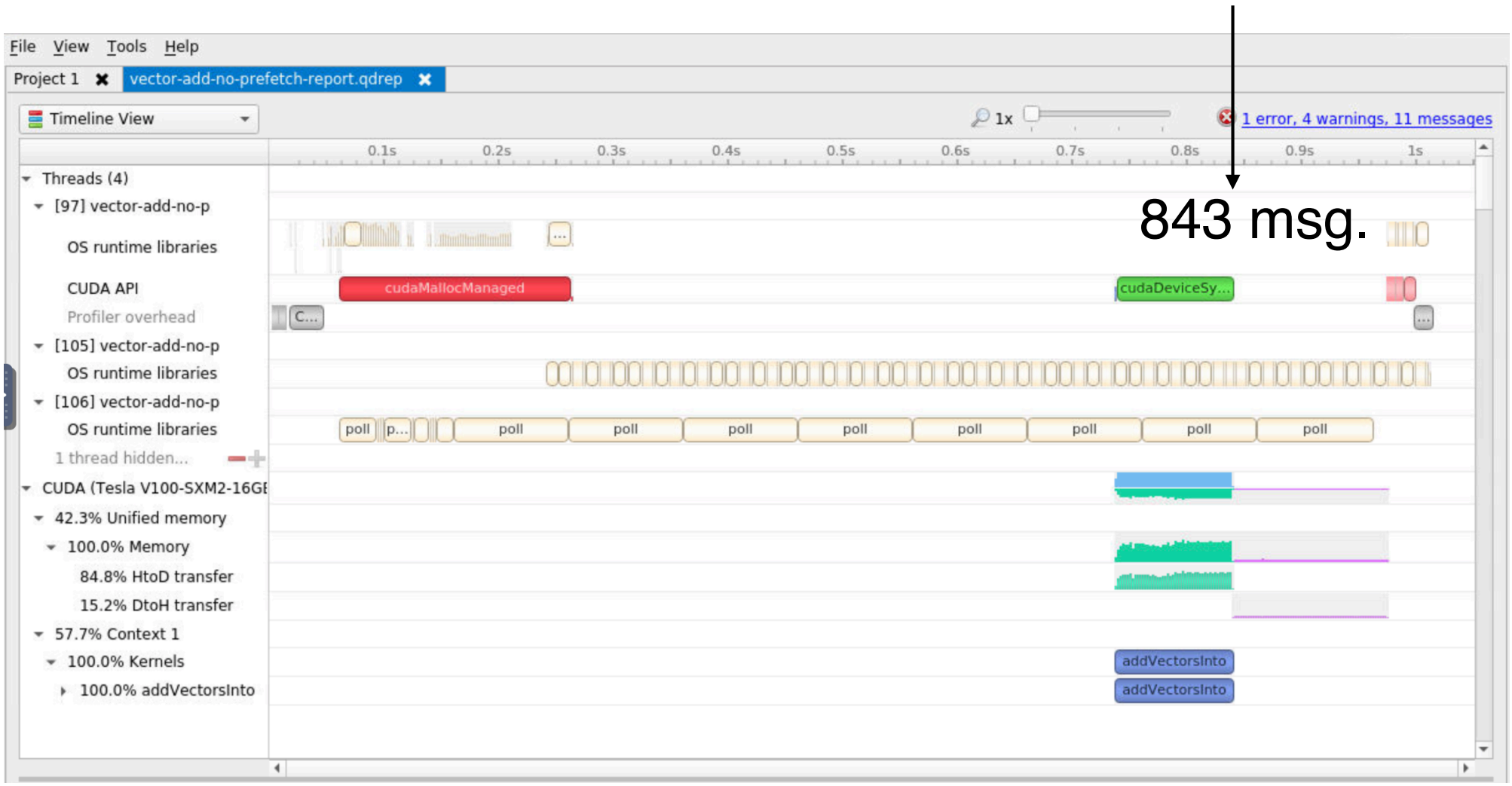
```
cudaMemPrefetchAsync(a, size, deviceId);
cudaMemPrefetchAsync(b, size, deviceId);
cudaMemPrefetchAsync(c, size, deviceId);
initWith<<<numberOfBlocks, threadsPerBlock>>>(3, a, N);
initWith<<<numberOfBlocks, threadsPerBlock>>>(4, b, N);
initWith<<<numberOfBlocks, threadsPerBlock>>>(0, c, N);
cudaDeviceSynchronize();
```

```
addVectorsInto<<<numberOfBlocks, threadsPerBlock>>>(c, a, b, N);
```

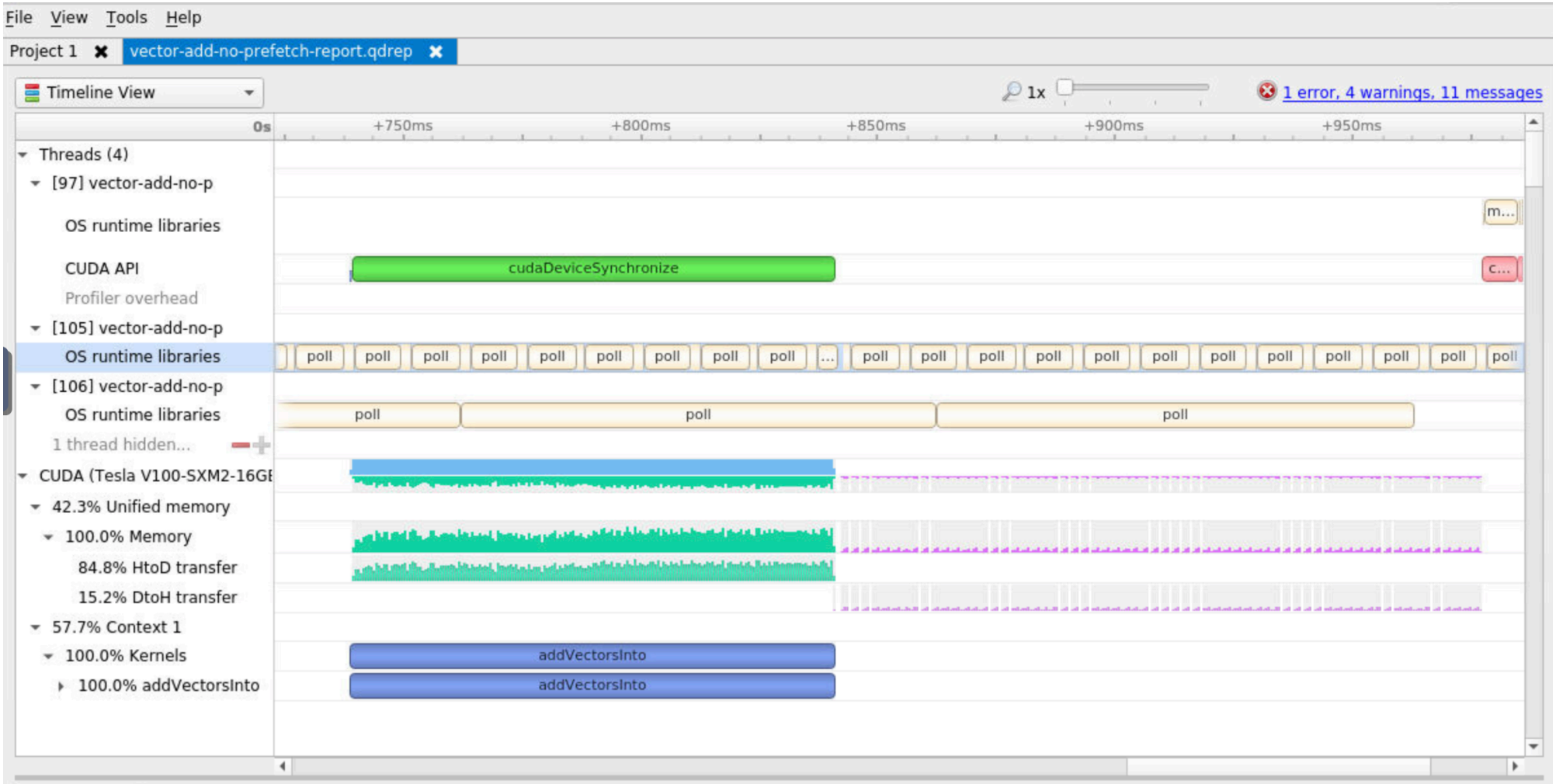
```
cudaMemPrefetchAsync(c, size, cudaCpuDeviceId);
checkElementsAre(7, c, N);
```

```
cudaFree(a);
cudaFree(b);
cudaFree(c);
```

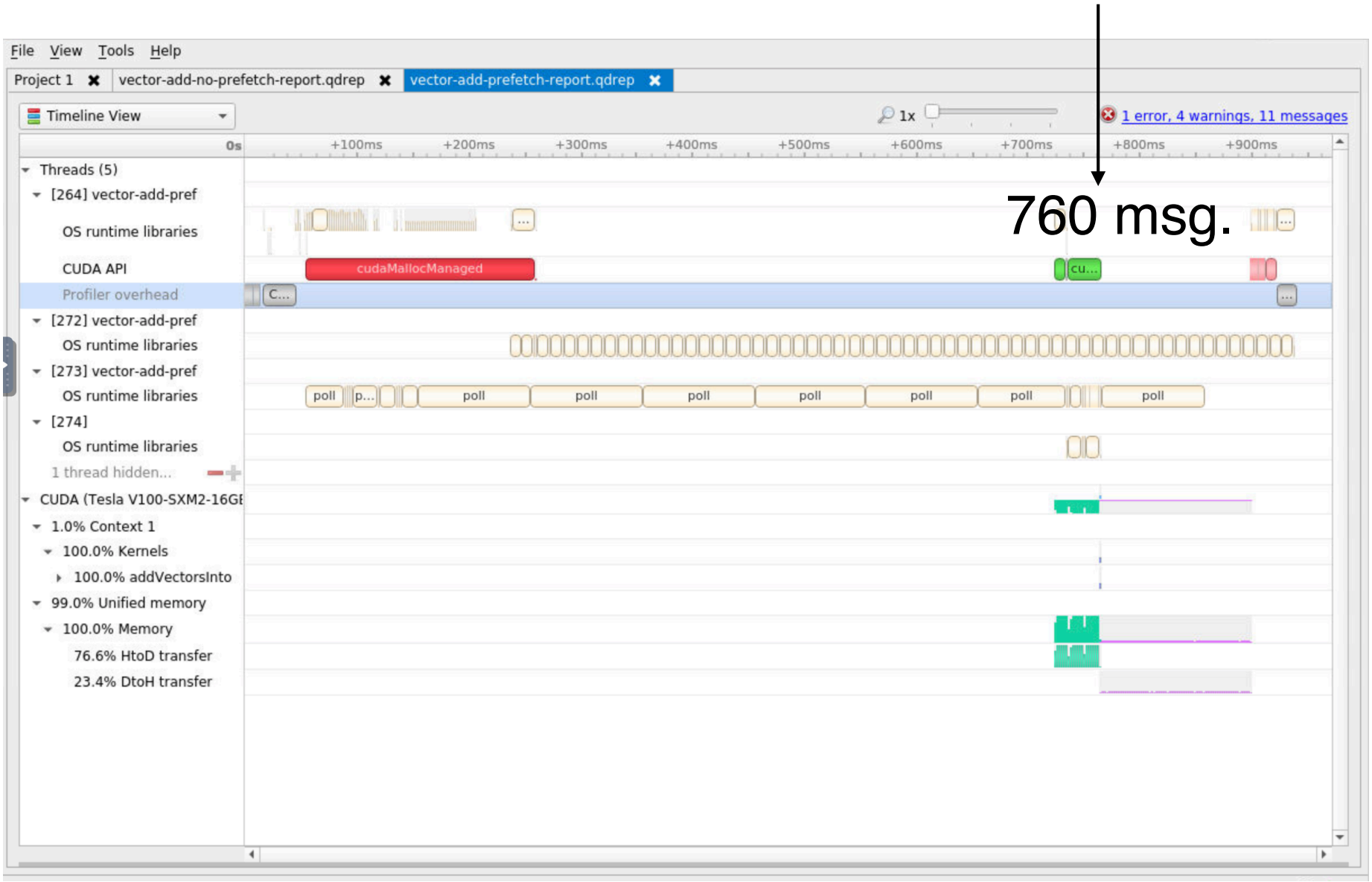
Profiler for our departure point



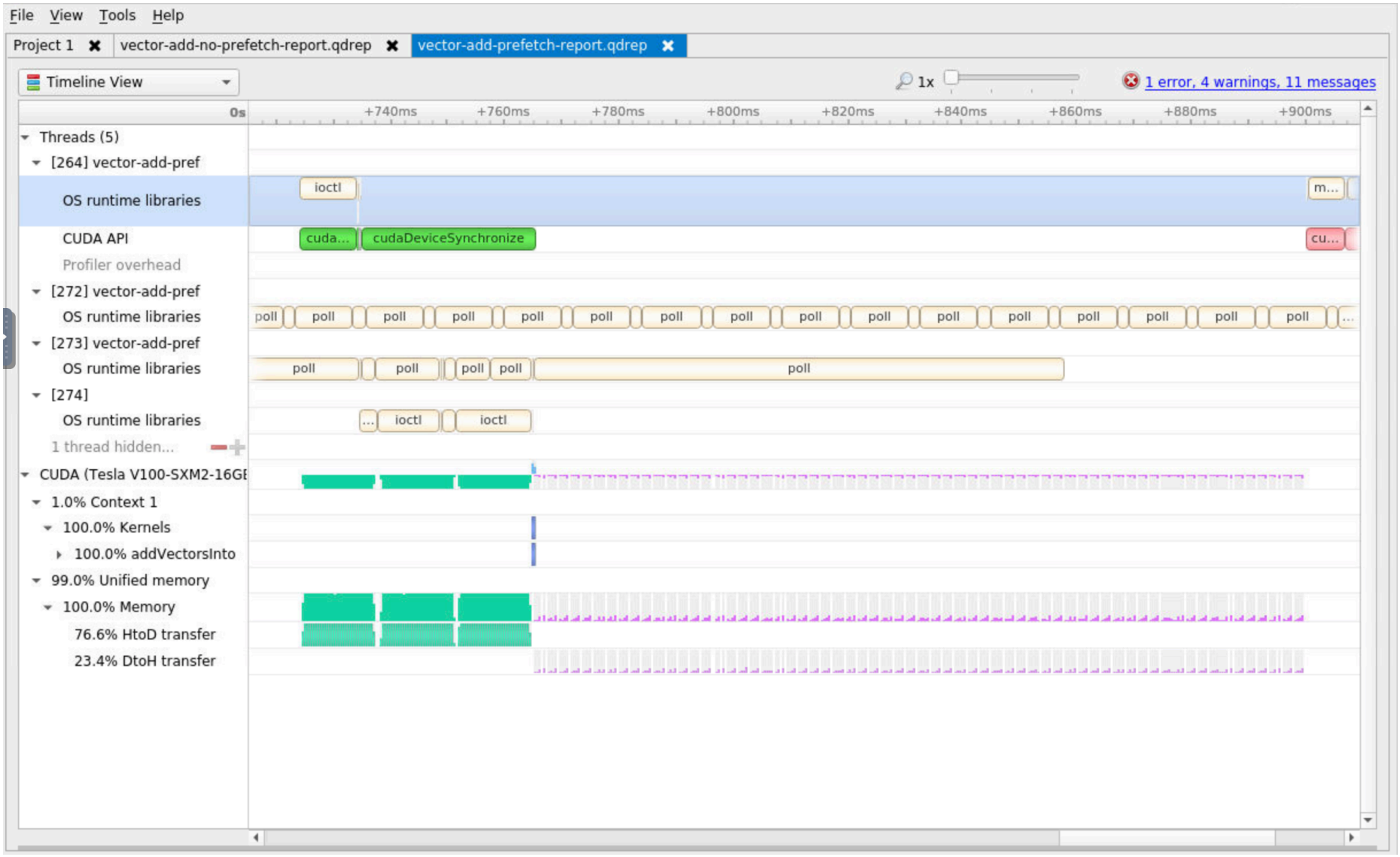
Profiler for our departure point (zoom)



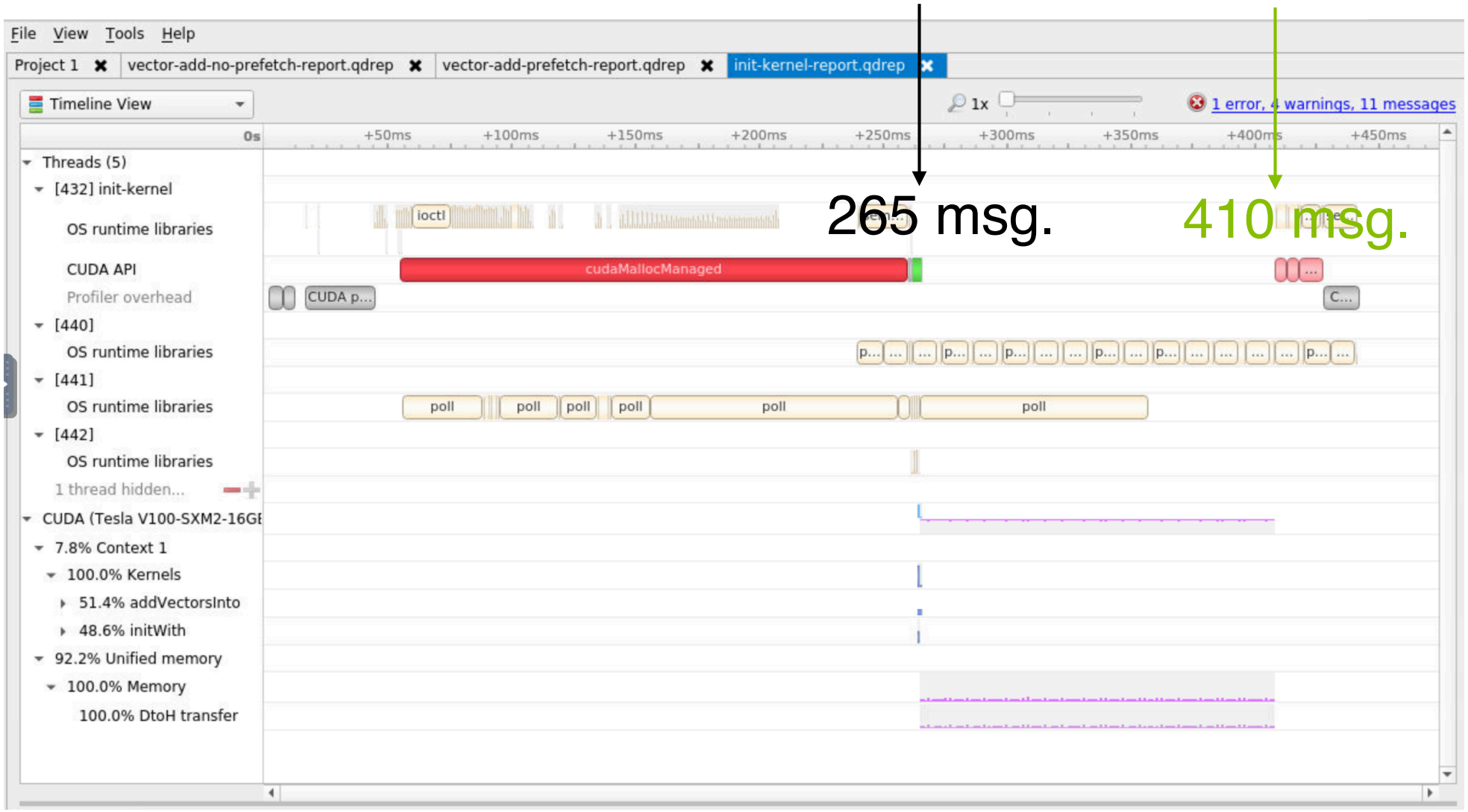
Exercise 1: Profiler for a prefetch in kernel



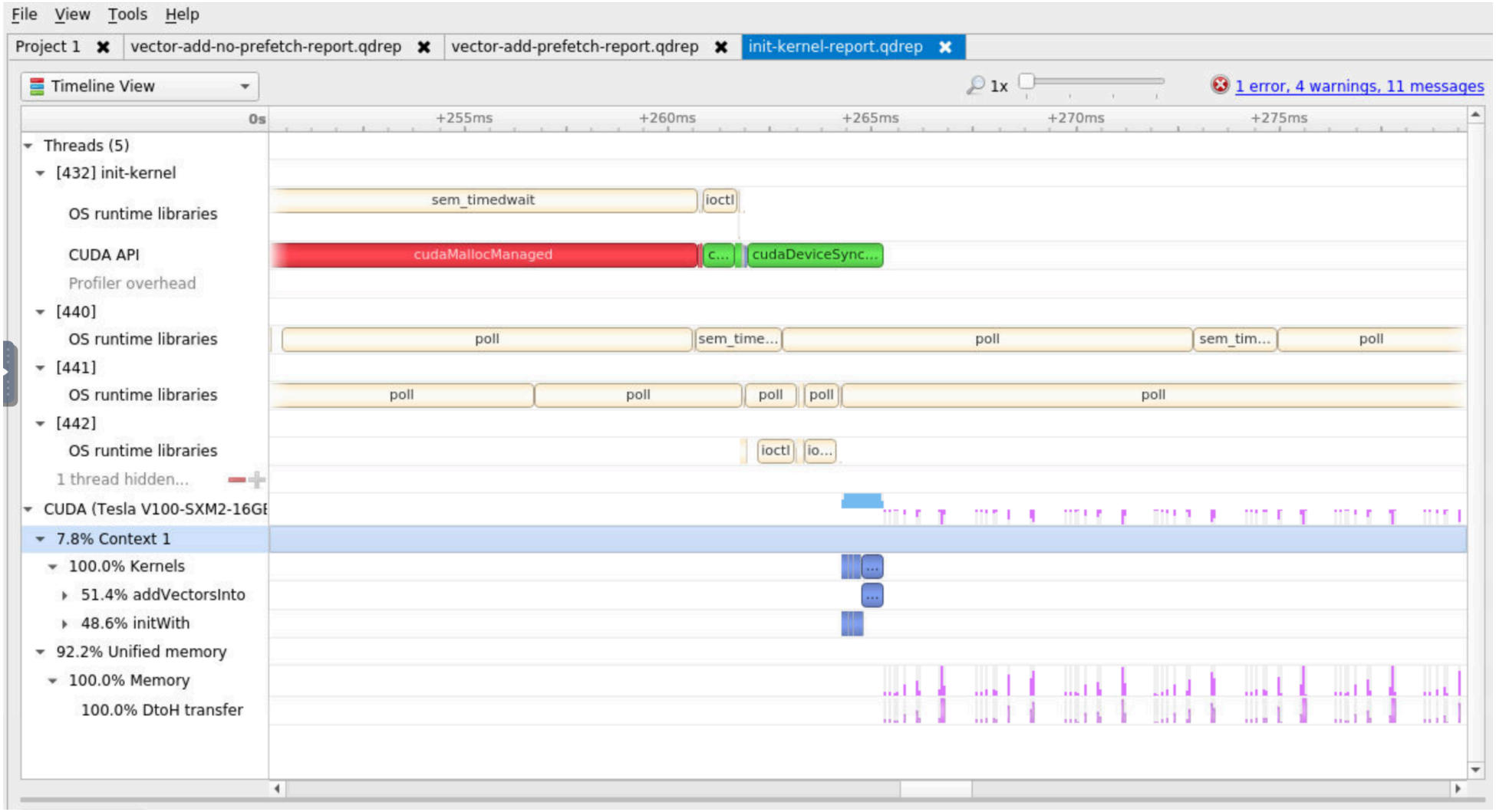
Exercise 1: Profiler for a prefetch in kernel (zoom)



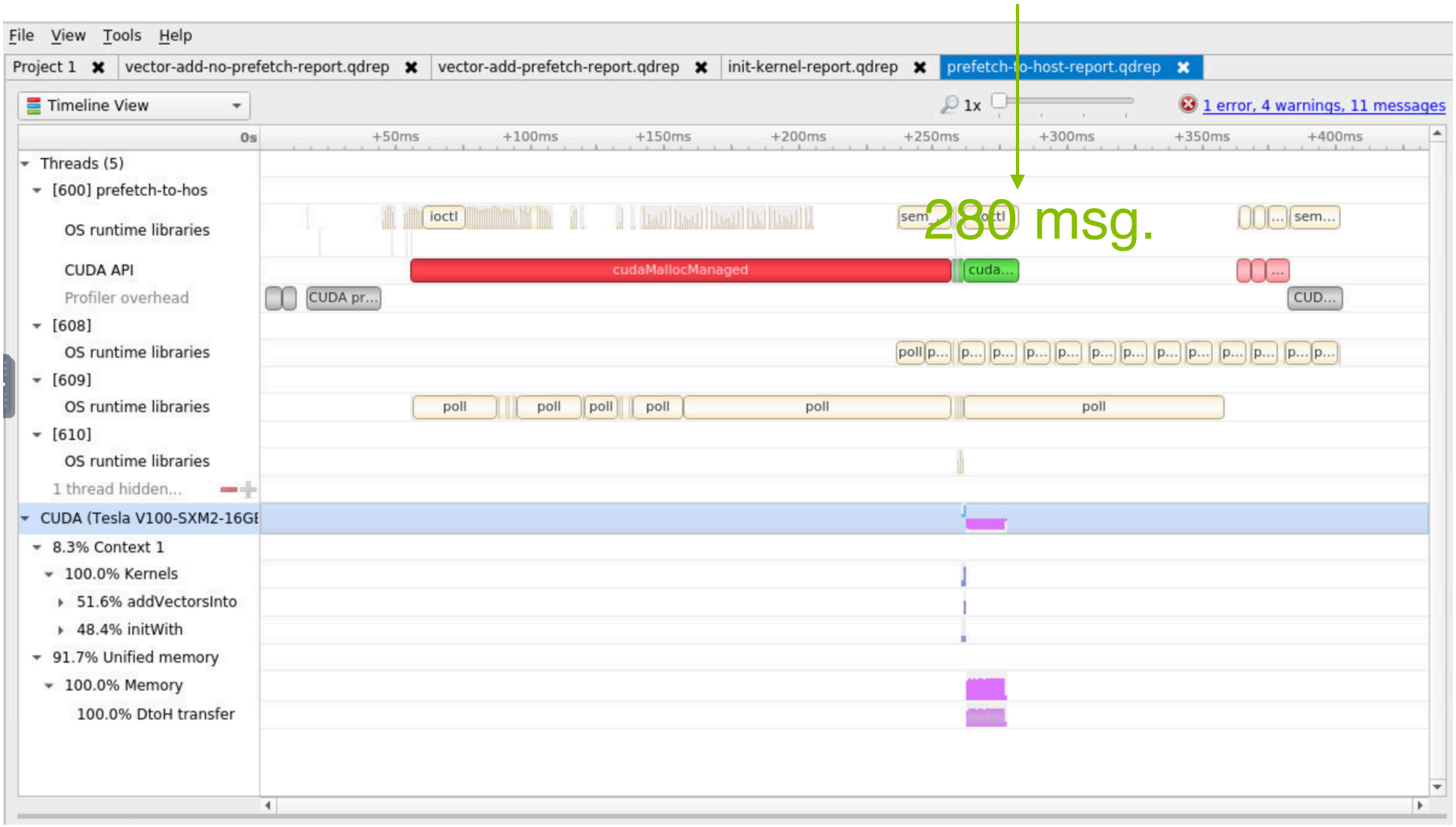
Exercise 2: Profiler for a initialization (and prefetch) on the GPU



Exercise 2: Profiler for a initialization (and prefetch) on the GPU (zoom)



Exercise 3: Profiler for the initialization on the GPU and final prefetch on the CPU



Exercise 3: Profiler for the initialization on the GPU and final prefetch on the CPU (zoom)