



## Chapter 3.2: Matrix Transpose LDS Kernel Analysis

ROCm Tutorial | AMD 2020

---

## Table of Contents

<b>CHAPTER 3.2: MATRIX TRANSPOSE LDS KERNEL ANALYSIS .....</b>	<b>2</b>
PREPARATION.....	2
COMPILING AND EXECUTING.....	2
PROFILING.....	3

# Chapter 3.2: Matrix Transpose LDS Kernel Analysis

In this tutorial, we will analyze the matrix transpose implementation that uses LDS using the rocProf profiler tool from ROCm

## Preparation

1. First in the tutorial repository go to the directory

```
cd 02_Matrix_Transpose
```

2. The application code is in `matrix_transpose_lds.cpp`

## Compiling and Executing

1. Compile the program

```
hipcc matrix_transpose_lds.cpp -o matrix_transpose_lds
```

2. Execute the program without profiler

```
./matrix_transpose_lds
```

3. Note that we are not printing any output from the matrices as the matrices are large. But you can add print code if desired

## Profiling

1. Now we will analyze the application through the profiler
2. First let us collect the kernel execution time using the performance measurement mode. Run the following command:

```
rocprof --stats ./matrix_transpose_lds
```

You will get the output in a file results.csv. Note down the kernel duration(ns)

3. Now we will be collecting the HW performance counters. For this application we have provided this file in “metrics\_matrix\_transpose\_lds\_kernel.txt
4. On a closer look the only thing different in the metrics file for this experiment when compared to the metrics file of the copy kernel is the kernel name which is set to transpose\_lds\_kernel.
5. Now we will run the application in performance counter mode using our defined metric file:

```
rocprof -i metrics_matrix_transpose_lds_kernel.txt -o metrics_matrix_transpose_lds.csv  
./matrix_transpose_lds
```

This will output the results of the HW performance counters in metrics\_matrix\_transpose\_lds.csv.

6. Keep a record of the results obtained for this kernel. For our case, we obtained the results below. Your results might be different depending on the GPU you are on:

```
Kernel time(ns):3908787  
TCC_EA_RDREQ_sum: 524313  
TCC_EA_WRREQ_sum: 524288
```