

Xilinx AML Application Userguide

(Version 1.0)

1. Introduction

Xilinx AML Application is designed for anti-money laundering solution to help users quickly and easily accelerate watch list management. And the Watch List fields can be direct string matched, substring matched and edit distance matched.

Now the application can be run on both on-premises Alveo cards and in FPGA instances in the cloud with Alveo U50/U200/U250.

2. Application Usage

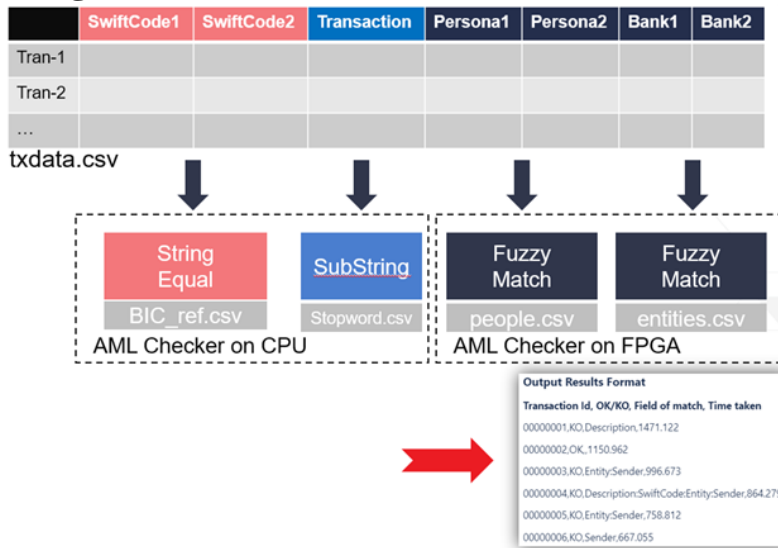


Figure 1. Diagram of AML Design

The application accelerates SWIFT MT103 transaction checking against given watch lists. For each transaction, this checker will return its match results for every comparing field.

Now the application is containerized and can be easily run in a few minutes in the cloud or on premises. Details can be found at link <https://www.xilinx.com/products/acceleration-solutions/anti-money-laundering-application.html>.

2.1 Sample

Sample 1. Try the demo case on U250

```
$ /opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe --demo
```

By the end of the license message print, it will ask for user's input "yes/no" to acknowledge the agreement. Type "yes" to acknowledge the licence agreement. Parameter "--accept-EULA" can be used to bypass the license message.

```

Terminal - nimbix@JARVICENAE-0A0A1851: ~/Desktop
File Edit View Terminal Tabs Help
PATH : /opt/xilinx/xrt/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
LD_LIBRARY_PATH : /opt/xilinx/xrt/lib:
PYTHONPATH : /opt/xilinx/xrt/python:
Type
/opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe -h
to get command list:
--accept-EULA (acknowledge the licence agreement) type: bool default: false
--demo (Run demo) type: bool default: false
-i, --in-dir (input directory of deny/match list) type: string default: ""
-m, --mode (0:FPGA-only, 1:CPU-only, 2:both and comparing results) type: int32 default: 0
# FPGA-only mode
/opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe -i /path/to/deny&match_list -m 0 --accept-EULA
# CPU-only mode
/opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe -i /path/to/deny&match_list -m 1 --accept-EULA
# FPGA-CPU mode
/opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe -i /path/to/deny&match_list -m 2 --accept-EULA
# Demo mode
/opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe --demo
Demo datasets are located in /home/nimbix/demo_data/
nimbix@JARVICENAE-0A0A1851:~/Desktop

```

Figure 2. Startup Screen of AML App

Note: when run in a NUMA machine, suggest installing and using “numactl” command to make performance stable.

When the execution is done, it will print results shown in Figure 1.

```

For FPGA, 100 transactions were processed.
Min(ms)      Max(ms)      Avg(ms)
-----
3.141        6.145        4.62566
-----

```

Figure 3. Output result for Demo Mode

Sample 2. Run test cases on U250 with CPU checker

```

$ /opt/xilinx/apps/vt_data_analytis/aml/bin/vt_data_analytis_aml.exe --accept-EULA -i
/home/nimbix/demo_data/ -m 2

```

When the job is completed, it prints the statistic report of execution time below. And all checked results will be saved in /home/nimbix/results. And final output will print “Check Passed!”.

```

For FPGA, 100 transactions were processed.
Min(ms)      Max(ms)      Avg(ms)
-----
3.416        6.237        4.78226
-----

Start to check...

For CPU, 100 transactions were processed.
Min(ms)      Max(ms)      Avg(ms)
-----
194.403     1371.57     587.542
-----

Check passed!

```

Figure 4. Output Result for FPGA&CPU mode

2.2 Prerequisites

2.2.1 Device and Software

This application supports Xilinx FPGA Alveo U50/U200/U250 card. To run this application on users' machines, please make sure:

- For Alveo U50, Xilinx FPGA Alveo U50 (shell xilinx_u50_gen3x16_xdma_201920_3) card is installed correctly. (default device id is 0)
- For Alveo U200, Xilinx FPGA Alveo U200 (shell xilinx_u200_xdma_201830_2) card is installed correctly. (default device id is 0)
- For Alveo U250, Xilinx FPGA Alveo U250 (shell xilinx_u250_xdma_201830_2) card is installed correctly. (default device id is 0)
- Docker (with sudo access)

When deployed in Nimbix, PushToCompute flow will deploy the application in an instance with ubuntu18.04, U50/U200/U250, and XRT 2020.1.

2.2.2 Datasets

The application provides demo watch lists inside the docker images. As details, people.csv and entities.csv has 6 million and 9474 records, respectively.

2.3 Run Application

The application provides 2 task modes, demo and normal mode.

In demo mode, one demo watch list and transaction list are given as default input.

In normal mode, both watch list and transaction list (-i) can be assigned by user. The name of each list must be same as demo. And the output result will be written into your home directory.

Table 1. list the options in this application.

Command	Default value	Function
--accept-EULA	False	acknowledge the license agreement and skip printing license acknowledgement. If false, it will print out license file to user console and ask for user's input "yes/no" to acknowledge the agreement
--demo	False	If true, run demo case
-i, --in-dir	""	Normal mode, the directory of user's watch and transaction list
-m, --mode	0	Mode setting, 0 for FPGA-only, 1 for CPU-only, 2 for both and comparing results

2.4 Performance Specification

In demo watch list, people.csv and entities.csv has 6 million and 9474 records, respectively. For given 100 transaction as example, the performance is:

- On U50, 2 fuzzy match kernels, achieved 6.7ms per record.
- On U200, 2 fuzzy match kernels, achieved 8.2ms per record.
- On U250, 4 fuzzy match kernels, achieved 4.7ms per record.

3 Limitation

- Current kernel will only handle input field string whose length is less than 36 and greater than 0. Longer field string will be truncated to 35 characters.
- Nested double quote is forbidden in the field of input transaction CSV file.