Sergiu Buhatel, Research Assistant – Computing/ML

Migration from MySQL on windows to OmniSci on Ubuntu (NVIDIA DGX A100)

Table of Contents

[1 Introduction 1](#_Toc197768542)

[2 MySQL 2](#_Toc197768543)

[3 OmniSci 3](#_Toc197768544)

[3.1 NVIDIA DGX A100 3](#_Toc197768545)

[3.2 Install 3](#_Toc197768546)

[3.3 Configuration 3](#_Toc197768547)

[3.3.1 omnisci.conf 3](#_Toc197768548)

[3.4 Run 4](#_Toc197768549)

[3.4.1 launch.sh 4](#_Toc197768550)

[4 Run 4](#_Toc197768551)

[5 SQL 4](#_Toc197768552)

[6 Import 5](#_Toc197768553)

[7 Basic Commands 5](#_Toc197768554)

[8 omnisql 6](#_Toc197768555)

[9 References 6](#_Toc197768556)

# Introduction

This paper outlines the process of migrating a MySQL database from a Windows environment to an OmniSci database running on Ubuntu. The target system is an NVIDIA DGX A100 machine equipped with 8 GPUs.

The primary motivation for this migration is to take advantage of GPU acceleration to significantly improve query performance. The current MySQL setup on Windows suffers from slow query execution times. By transitioning to OmniSci on the DGX A100, the system can leverage the parallel processing power of GPUs, resulting in markedly faster query performance.

# MySQL

The MySQL database is currently hosted on a Windows machine with the IP address 134.117.176.72. This server is equipped with 768 GB of RAM and runs Windows Server 2022 Standard. Despite its substantial hardware resources, query performance remains suboptimal.

The database can be accessed using MySQL Workbench by connecting to the blazing\_sql schema. This schema name reflects an earlier migration attempt to BlazingSQL. However, due to certain limitations encountered with BlazingSQL, OmniSci was chosen as a more reliable and performant alternative.

A screenshot of a computer

AI-generated content may be incorrect.

The connection details for the MySQL database are as follows:

* **Username:** root
* **Password:** root
* **Port:** 3307
* **Database Name:** blazing\_sql

# OmniSci

## NVIDIA DGX A100

The NVIDIA DGX A100 machine is accessible at IP address **10.128.20.12**. From the Windows machine, you can connect to it using the following SSH command:

ssh <username>@10.128.20.12

The DGX A100 runs **Ubuntu 20.04.6 LTS**, which can be verified by executing:

lsb\_release -a

To transfer files from the Windows machine to the Ubuntu system, you can use the scp command. For example:

scp \* <username>@10.128.20.12:/home/<username>

This command copies all files from the current Windows directory to the /home/<username> directory on the DGX A100. Ensure that the target directory exists before executing the command; if it does not, create it beforehand to avoid errors.

## Install

To install OmniSci using Docker, run the following command:

sudo docker pull omnisci/core-os-cuda:latest

To confirm that the image has been pulled successfully, list available Docker images and filter for OmniSci:

docker images | grep omnisci

## Configuration

cd /home/ra2/omnisci/omnisci-docker-storage

vi omnisci.conf

### omnisci.conf

allowed-import-paths = ["/data/twitter\_tables"]

This configuration enables bulk data import from the Docker container directory /data/twitter\_tables when using the COPY FROM SQL command within omnisql. If additional directories are required, they can be added as a comma-separated list. To apply the changes, the OmniSci Docker container must be restarted. As the configuration file has already been created, no further modifications are currently needed.

## Run

cd /home/ra2/omnisci

./launch.sh

This will launch OmniSci, provided is not running yet.

### launch.sh

sudo -E docker run --runtime=nvidia \

--name omnisci \

-v /home/ra2/omnisci/omnisci-docker-storage:/omnisci-storage \

-v /home/ra2:/data \

-p 6273-6280:6273-6280 \

omnisci/core-os-cuda:latest

This configuration mounts the local directory /home/ra2/omnisci/omnisci-docker-storage into the Docker container as /omnisci-storage, and the local directory /home/ra2 as /data.

The OmniSci database is stored persistently outside the container at /home/ra2/omnisci/omnisci-docker-storage, ensuring that data remains intact even if the container is removed or recreated.

# Run

cd /home/ra2/omnisci

./launch.sh

If is running already, stop it and remove the container and re-launch again.

docker ps -a

docker kill <container>

docker rm <container>

# SQL

Tap into the running docker container

docker exec -it <container> bash

./bin/omnisql

# Import

Tap into the running docker container

docker exec -it <container> bash

./bin/omnisql

# Basic Commands

# Install omnisci on Ubuntu (This can be done once, and it is already done)

sudo docker pull omnisci/core-os-cuda:latest

# List docker images

docker images

# Stop existing OmniSci server in case it is running and you want to running it again

sudo docker ps | grep omnisci

sudo docker kill <container id>

# Check whether OmniSci server is till running

sudo docker ps

# Launch OmniSci server

cd /home/ra2/omnisci

./launch.sh

# If getting an error remove the existing container

sudo docker rm <container id>

# This is the content of launch.sh

sudo -E docker run --runtime=nvidia \

--name omnisci \

-v /home/ra2/omnisci/omnisci-docker-storage:/omnisci-storage \

-v /home/ra2:/mnt/data \

-p 6273-6280:6273-6280 \

omnisci/core-os-cuda:latest

# It mounts /home/ra2/omnisci/omnisci-docker-storage into /omnisci-storage for

# storing the database, and /home/ra2 to /mnt/data for

# having access to the twitter csv files in the /home/ra2/twitter\_tables directory

# Launch a bourne shell inside the running docker container

# Find first the running container id

sudo docker ps | grep omnisci

# Launch the bourse shell inside the docker container

sudo docker exec -it <container id> bash

# This will change the prompt to #

# Change the directory to /omnisci/bin and launch omnisql

cd /omnisci/bin

./omnisql

# Password: HyperInteractive

# Create a dummy table in omnisci and quary the data

# Open a new commad prompt and chnage directory to /home/ra2/omnisci

cd /home/ra2/omnisci

# Activate the virtual enviroment previously created

source env/bin/activate

# If you don't have the virtual enviroment, created as following

python3 -m venv env

# and then activate the virtual environment

source env/bin/activate

# Create a table and display the results

python create\_table.py

python add\_data.py

python display\_table.py

# omnisql

# Launch the bourse shell inside the docker container

sudo docker exec -it <container id> bash

# This will change the prompt to #

# Change the directory to /omnisci/bin and launch omnisql

cd /omnisci/bin

./omnisql

# Password: HyperInteractive

|  |  |
| --- | --- |
| **Command** | **Description** |
| \t | List tables |
| \h | Help |

At omnisql> command prompt, you ca issue SQL commands.

# References

<https://docs.omnisci.com/v5.1.0/4_docker_gpu_os_apt_recipe.html>