

Article

[Piyush Adhikari](#) · Oct 19, 2022 3m read

Ingestion and Querying Speed Test

The capacity of taking numerous records every second while also facilitating real-time queries simultaneously in real time is called Hybrid Transactional Analytical Processing (HTAP). It is also called Transactional analytics or Transanalytics or Translytics and is a very useful element in scenarios where there is constant flow of real time data coming from IIOT sensors or data on fluctuations in stock market, and supporting the need for querying these data sets in real-time or near real-time.

I am sharing my experience on running a demo with test streaming data, having constant input and constant queries at the very same time, on different platforms and how differently each one reacted to it, as in, the speed of input and output of data in each platform and their performance. The platforms I have experimented with in this demo are: InterSystems IRIS, MariaDB and MySQL.

Walkthrough video of the demo: <https://www.loom.com/share/8e30930232db4b18ac455e310226575d>

Prerequisites

Docker and Docker Compose

Running it locally

We are running the speed tests for ingestion of records and querying them on three different platforms, MySQL, MariaDB and InterSystems IRIS Community Edition.

For the purpose, YAML file of containers for each platform needs to be downloaded as follows:

1. Run the speed test with MySQL
 1. Download YAML file: <https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-htap/master/docker-compose-mysql.yml>
 2. Open command prompt in the folder the above YAML file is in and run the command, to build the docker image for MySQL: `docker-compose -f ./docker-compose-mysql.yml up`
 3. Open a browser tab at <http://localhost:10000> to see the demo UI.
2. Run the speed test with MariaDB
 1. Download YAML file: <https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-htap/master/docker-compose-mariadb.yml>
 2. Open command prompt in the folder the above YAML file is in and run the command, to build the docker image for MariaDB: `docker-compose -f ./docker-compose-mariadb.yml up`
 3. Open a browser tab at <http://localhost:10000> to see the demo UI.
3. Run the speed test with InterSystems IRIS
 1. Download YAML file: <https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-htap/master/docker-compose.yml>
 2. Open command prompt in the folder the above YAML file is in and run the command, to build the docker image for InterSystems IRIS: `docker-compose up`
 3. Open a browser tab at <http://localhost:10000> to see the demo UI.

After the tests are run from the localhost, the results can be seen in forms of records per second ingested and queried.

InterSystems IRIS has more impressive numbers compared to MySQL and MariaDB.

Exception/s

While trying to run the docker containers for each of the platforms, one might get following error:

‘ Missing Linux capabilities ’

```
htapisdb      | [INFO] Writing status to file: /home/irisowner/irissys/iscagent.status
htapisdb      | Reading configuration from file: /home/irisowner/irissys/iscagent.conf
htapisdb      | ISCAgent[19]: Starting
htapisdb      | ISCAgent[24]: Starting ApplicationServer on *:2188
htapisdb      | [INFO] ...executed command /home/irisowner/irissys/startISCAgent.sh 2188
htapisdb      | [ERROR] Required Linux capability cap_setuid is missing.
htapisdb      | [ERROR] Required Linux capability cap_dac_override is missing.
htapisdb      | [ERROR] Required Linux capability cap_fowner is missing.
htapisdb      | [ERROR] Required Linux capability cap_setgid is missing.
htapisdb      | [ERROR] Required Linux capability cap_kill is missing.
htapisdb      | [FATAL] Your IRIS container is missing one or more required Linux capabilities.
htapisdb exited with code 1
```

One can skip the checking for Linux capabilities by adding the line:

command: --check-caps false

([@Bob Kuszewski](#), 2022)

in the htapisdb section of the YAML file as shown below:

```
htapisdb:
  # We can't use clean IRIS Community because its HEALTHCHECK period is too long.
  # So we are going to use our Iridemo-base-db image.
  #image: store/intersystems/iris-community:2019.3.0.309.0
  # version-1.2 is based on IRIS Community 2019.3
  # version-1.4 is based on IRIS Community 2019.4
  # version-1.5.1 is based on IRIS Community 2020.1.0.197.0
  # version-1.5.2 is based on IRIS Community 2020.2.0.196.0
  # version-1.7.1 is based on IRIS Community 2020.3.0.200.0
  # version-1.8.0 is based on IRIS Community 2021.1.0.215.0
  # version-1.8.1 is based on IRIS Community 2021.1.0.215.3
  # version-1.9.1 is based on IRIS Community 2021.2.0.619.0
  # version-1.9.2 is based on IRIS Community 2021.2.0.637.0
  image: intersystemsdc/iridemo-base-irisdb-community:version-1.9.3
  hostname: htapisdb
  init: true
  container_name: htapisdb
  command: --check-caps false
  ports:
    - "10001:52773" # 52773 is the webserver/management portal port: http://localhost:9092/csp/sys/UtilHome.csp
```

Real-time test run on each platform: <https://www.loom.com/share/859866ac97a6476fbb367112fc71c2c2>

Speed testing results from each platforms

InterSystems IRIS:

Ingestion and Querying Speed Test

Published on InterSystems Developer Community (<https://community.intersystems.com>)



MariaDB:



MySQL:

Ingestion and Querying Speed Test

Published on InterSystems Developer Community (<https://community.intersystems.com>)



References

1. [intersystems.com](https://openexchange.intersystems.com/package/Ingestion-Speed-Test?tab=details). (Amir, S.). InterSystems Open Exchange. [online] Available at: <https://openexchange.intersystems.com/package/Ingestion-Speed-Test?tab=details> [Accessed 20 Sep. 2022].
2. Kuszewski, B. (2022). Using InterSystems IRIS containers with Docker 20.10.14+. [online] InterSystems Developer Community. Available at: <https://community.intersystems.com/post/using-intersystems-iris-containers-docker-201014> [Accessed 20 Oct. 2022].

[#Analytics](#) [#Big Data](#) [#Database Transaction Processing](#) [#Data Import and Export](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/ingestion-and-querying-speed-test>