CLAIMS

1. A coordination server for controlling traffic buffering during database migration between a source database and a destination database, wherein the coordination server is configured to control traffic buffering of data by:

coordinating migration of data between the source database and the destination database by identifying overlap data and non-overlap data between the source database and the destination database;

instructing one or more clients to buffer the non-overlap data for data migration;

buffering the non-overlap data at the one or more clients to obtain buffered traffic data, wherein the one or more clients are configured to allow existing traffic to complete before sending a traffic buffered acknowledgement to the coordination server after buffering the non-overlap data;

checking whether the destination database is synchronized with the source database; and

releasing the buffered traffic data to the destination database after receiving the traffic buffered acknowledgement and determining that the destination database is synchronized with the source database.

2. The coordination server of claim 1, wherein the coordination server is configured to migrate the non-overlap data between the source database and the destination database.

3. The coordination server of claim 1 or claim 2, wherein the coordination server is configured to check whether the destination database is synchronized with the source database by checking a unique Global transaction identifier (GTID) of the source database.

4. The coordination server of claim 3, wherein the GTID comprises a latest data replication state of the source database.

5. The coordination server of claims 1 to 4, wherein the buffered traffic data release to the destination database is rate limited.

6. The coordination server of any one of claims 1 to 5, wherein the one or more clients are configured to check for existing traffic buffering before buffering the non-overlap data.

7. The coordination server of claim 1 to 6, wherein the coordination server is developed using Golang programming language.

8. A method for using a coordination server to control traffic buffering during database migration between a source database and a destination database comprising:

controlling traffic buffering of data using the coordination server by:

coordinating the migration of data between the source database and the destination database by identifying overlap data and non-overlap data between the source database and the destination database;

instructing one or more clients to buffer the non-overlap data for data migration;

buffering the non-overlap data at the one or more clients to obtain buffered traffic data, wherein the one or more clients are configured to allow existing traffic to complete before sending a traffic buffered acknowledgement to the coordination server after buffering the non-overlap data;

checking whether the destination database is synchronized with the source database; and

releasing the buffered traffic data to the destination database after receiving the traffic buffered acknowledgement and determining that the destination database is synchronized with the source database.

9. The method of claim 8, further comprising using the coordination server to migrate the non-overlap data between the source database and the destination database.

10. The method of claim 8 or claim 9, further comprising using the coordination server to check whether the destination database is synchronized with the source database by checking a unique Global transaction identifier (GTID) of the source database.

11. The method of claim 10, wherein the GTID comprises a latest data replication state of the source database.

12. The method of claims 8 to 11, wherein the buffered traffic data release to the destination database is rate limited.

13. The method of any one of claims 8 to 12, further comprising using the one or more clients to check for existing traffic buffering before buffering the non-overlap data.

14. A data migration device comprising a communication interface, a memory interface and a processing unit configured to perform the method of any one of claims 8 to 13.

15. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 8 to 13.

16. A computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 8 to 13.