Cloud Data Migration

Performance White Paper

Issue 01

Date 2022-09-30

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

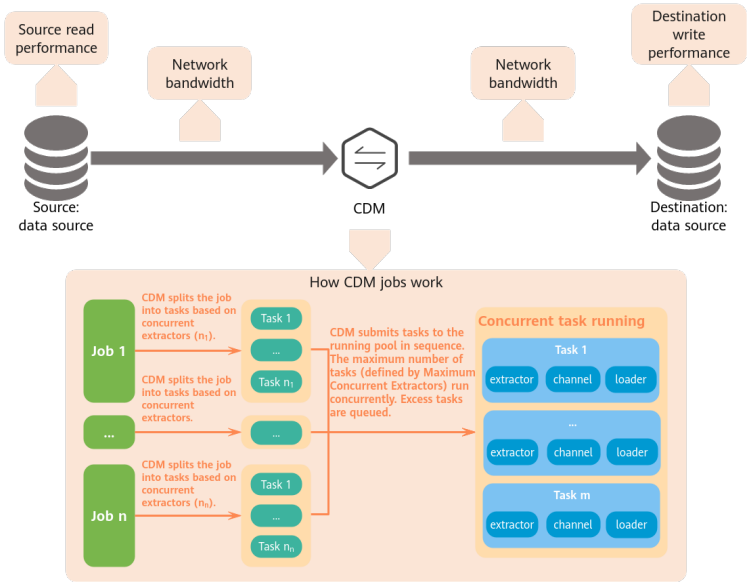
Contents

1 Factors Affecting Performance

Data Migration Model

Figure 1-1 shows the simplified migration model used by CDM.

Figure 1-1 Migration model used by CDM



CDM migrates data through data migration jobs. It works in the following way:

1. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration.

   NOTE

   Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter.

2. CDM submits the tasks to the running pool in sequence. The maximum number of tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued.

Factors Affecting Migration Performance

According to the migration model, the migration speed is affected by factors such as the source read speed, network bandwidth, destination write performance, and CDM cluster and job configuration.

Table 1-1 Factors affecting migration performance

Factor		Description
Service-related factors	Concurrent extractors of a job	The number of concurrent extractors can be set for a CDM job during the job creation. Setting a proper value for this parameter can effectively improve the migration speed. If the value is too small, migration will be too slow. If the value is too large, the migration job is overloaded and may fail. • When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data. • If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended that data be extracted in a single thread.
	Maximum concurrent extractors of a cluster	Setting a proper value for this parameter can effectively improve the migration speed. If the value is too small, migration will be too slow. If the value is too large, the source is overloaded and the system may be unstable. The maximum concurrent extractors vary depending on the CDM cluster flavor. The upper limit is twice the number of vCPUs. The following are the maximum concurrent extractors of some flavors: • cdm.large: 16 • cdm.xlarge: 32 • cdm.4xlarge: 64
	Service model	If the number of CDM jobs that run concurrently exceeds the maximum concurrent extractors for the CDM cluster, some jobs will be queued, and the migration will be prolonged. Avoid running too many jobs simultaneously, which may cause slow migration due to insufficient resources.
	Data model	The migration speed is also affected by the data structure. The following are some examples: • The wider a table is and the more string types the table has, the slower the migration is. • A large file is migrated more quickly than multiple small files whose total size is the same as the large file. • The more content a message has and the higher bandwidth it uses, the less transactions per second (TPS) are.

Factor	Description
Source read speed	It depends on the performance of the data source at the source. For details about how to increase the read speed, see the documents of data sources at the source.
Network bandwidth	The CDM cluster can communicate with the data source through an intranet, public network VPN, NAT, or Direct Connect. ● If they communicate through an intranet, the network bandwidth varies depending on the CDM instance flavor. – For cdm.large instances, the baseline and maximum bandwidths of the CDM cluster NIC are 0.8 and 3 Gbit/s, respectively. – For cdm.xlarge instances, the baseline and maximum bandwidths of the CDM cluster NIC are 4 and 10 Gbit/s, respectively. – For cdm.4xlarge instances, the baseline and maximum bandwidths of the CDM cluster NIC are 36 and 40 Gbit/s, respectively. ● If they communicate through the Internet, the network bandwidth is subject to the Internet bandwidth. The bandwidth for the CDM cluster depends on the EIP bound to the CDM cluster, and the bandwidth for the data source depends on the Internet bandwidth. ● If they communicate through a VPN, NAT, or Direct Connect, the network bandwidth is subject to the VPN, NAT, or Direct Connect bandwidth.
Destination write performance	It depends on the performance of the data source at the destination. For details about how to improve the performance, see the documents of data sources at the destination.

2 Performance Tuning

Overview

In addition to increasing the source read speed, improving the destination write performance, and increasing the bandwidth, you can accelerate migration using the following methods:

- **Use a CDM cluster of higher specifications**

  The NIC bandwidth and maximum number of concurrent extractors vary depending on the CDM cluster specifications. If you want to migrate data faster, or the metrics of your CDM cluster (such as the CPU usage, disk usage, and memory usage) are often high, you may need a CDM cluster with higher specifications for data migration.

- **Use multiple CDM clusters**

  In some scenarios, you are advised to use multiple CDM clusters to share workloads to improve migration efficiency and stability. The following are some examples:

  – Multiple CDM clusters are required for different purposes or by multiple business departments. For example, you may need one CDM cluster for running data migration jobs and another one as an agent for DataArts Studio Management Center.

  – You want to migrate a large number of tables. In this case, you can use multiple CDM clusters to run jobs simultaneously to improve migration efficiency.

  – The CPU usage, disk usage, and memory usage of the in-use CDM cluster are often high. In this case, you are advised to use multiple CDM clusters to shared workloads.

- **Avoid running too many CDM jobs simultaneously**

  If the number of CDM jobs that run concurrently exceeds the maximum concurrent extractors for the CDM cluster, some jobs will be queued, and the migration will be prolonged.

  Avoid running too many jobs simultaneously, which may cause slow migration due to insufficient resources.

- **Change concurrent extractors**

  If the number of tasks is small, adjusting the number of concurrent extractors is the best way to improve performance. You can set the number of

concurrent extractors for a job and the maximum number of concurrent extractors for a cluster.

CDM migrates data through data migration jobs. It works in the following way:

a. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration.

   NOTE

   Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter.

b. CDM submits the tasks to the running pool in sequence. The maximum number of tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued.

By setting appropriate values for parameters **Concurrent Extractors** and **Maximum Concurrent Extractors**, you can accelerate migration. For details about how to change **Concurrent Extractors**, see **Changing Concurrent Extractors**.

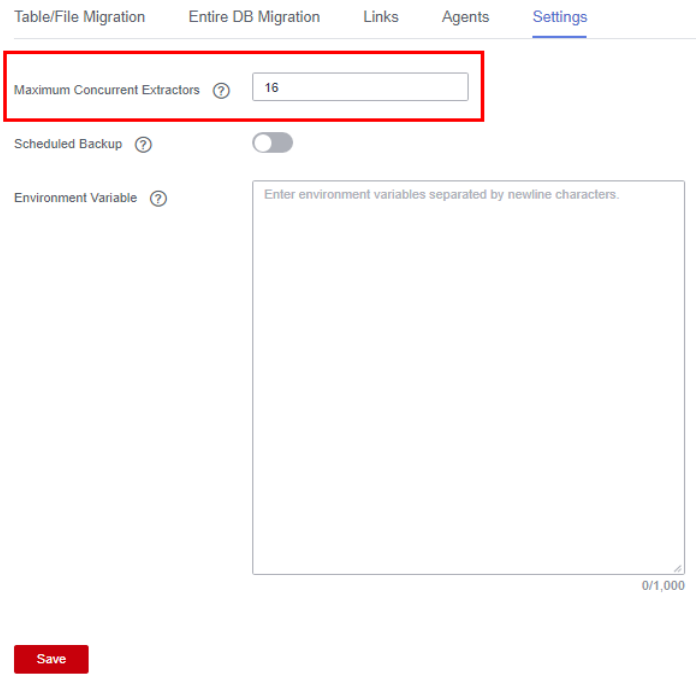
Changing Concurrent Extractors

1. The maximum number of concurrent extractors for a cluster varies depending on the CDM cluster flavor. You are advised to set the maximum number of concurrent extractors to twice the number of vCPUs of the CDM cluster.

   Table 2-1 Maximum number of concurrent extractors for a CDM cluster

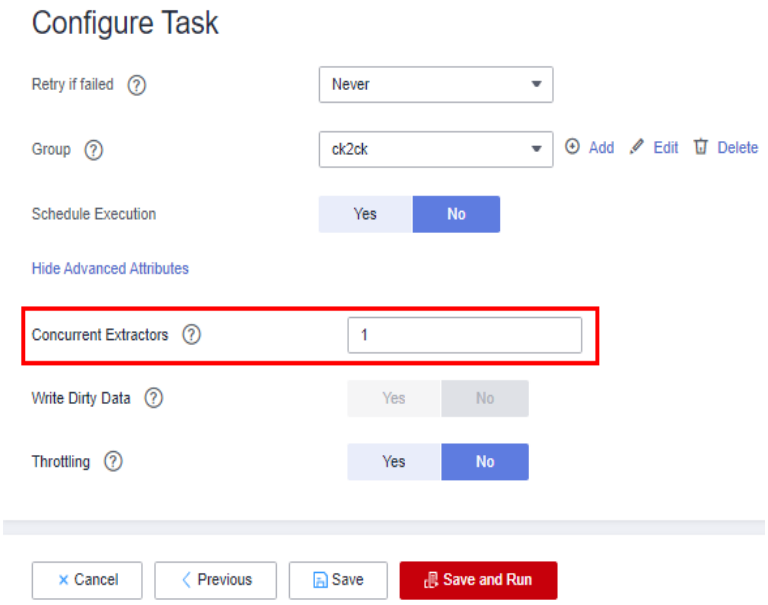
Flavor	vCPUs/Memory	Maximum Concurrent Extractors
cdm.large	8 vCPUs, 16 GB	16
cdm.xlarge	16 vCPUs, 32 GB	32
cdm.4xlarge	32 vCPUs, 64 GB	64

Figure 2-1 Setting Maximum Concurrent Extractors for a CDM cluster



2. Configure the number of concurrent extractors based on the following rules:
   a. When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data.
   b. If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended that data be extracted in a single thread.
   c. Set **Concurrent Extractors** for a job based on **Maximum Concurrent Extractors** for the cluster. It is recommended that **Concurrent Extractors** is less than **Maximum Concurrent Extractors**.

Figure 2-2 Setting Concurrent Extractors for a job



3 Reference: Job Splitting Dimensions

CDM splits jobs for different data sources based on different dimensions. **Table 3-1** lists the splitting dimensions.

Table 3-1 Job splitting dimensions for different data sources

Data Source Category	Data Source	Job Splitting Rule
Data warehouse	GaussDB(DWS)	• Jobs can be split based on table fields. • Jobs cannot be split based on table partitions.
	Data Lake Insight (DLI)	• Jobs can be split based on the partitioning information of partitioned tables. • Jobs cannot be split based on non-partitioned tables.
Hadoop	MRS HDFS	Jobs can be split based on files.
	MRS HBase	Jobs can be split based on HBase regions.
	MRS Hive	• When the read mode is HDFS, jobs can be split based on Hive files. • When the read mode is JDBC, jobs cannot be split.
	FusionInsight HDFS	Jobs can be split based on files.
	FusionInsight HBase	Jobs can be split based on HBase regions.
	FusionInsight Hive	• When the read mode is HDFS, jobs can be split based on Hive files. • When the read mode is JDBC, jobs cannot be split.

Data Source Category	Data Source	Job Splitting Rule
	Apache HDFS	Jobs can be split based on files.
	Apache HBase	Jobs can be split based on HBase regions.
	Apache Hive	• When the read mode is HDFS, jobs can be split based on Hive files. • When the read mode is JDBC, jobs cannot be split.
Object storage	Object Storage Service (OBS)	Jobs can be split based on files.
File system	FTP	Jobs can be split based on files.
	SFTP	Jobs can be split based on files.
	HTTP	Jobs can be split based on files.
Relational database	RDS for MySQL	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	RDS for PostgreSQL	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	RDS for SQL Server	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	MySQL	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	PostgreSQL	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	Microsoft SQL Server	• Jobs can be split based on table fields. • Jobs cannot be split based on table partitions.

Data Source Category	Data Source	Job Splitting Rule
	Oracle	• Jobs can be split based on table fields. • Jobs can be split based on table partitions only when **Extract by Partition** is configured.
	SAP HANA	• Jobs can be split based on table fields. • Jobs cannot be split based on table partitions.
	Database shard	Each backend connects to a subjob, which can be split based on primary keys.
NoSQL	Distributed Cache Service (DCS)	Jobs cannot be split.
	Redis	Jobs cannot be split.
	Document Database Service (DDS)	Jobs cannot be split.
	MongoDB	Jobs cannot be split.
	Cassandra	Jobs can be split based on the token range of Cassandra.
Message system	Apache Kafka	Jobs can be split based on topics.
	DMS Kafka	Jobs can be split based on topics.
	MRS Kafka	Jobs can be split based on topics.
Search	Elasticsearch	Jobs cannot be split.
	Cloud Search Service (CSS)	Jobs cannot be split.

4 Reference: CDM Performance Test Data

Background

The performance metrics provided in this document are for reference only. The performance at your site may be affected by factors such as the data source performance at the source or destination, network bandwidth, latency, and the data and service model. It is recommended that you test the speed with a small amount of data before migration.

Environment

- A xlarge CDM cluster of the 2.9.1 200 version
- A table which has 50 million rows and 100 columns, and three HDFS binary files which have 35.97 million rows and 100 columns, 66.67 million rows and 100 columns, and 100 million rows and 100 columns, respectively.
- Number of concurrent extraction jobs for determining the maximum extraction/write rate: 1, 10, 20, 30, and 50

Data Source Extraction and Write Performance Test Data

Table 4-1 and Table 4-2 provide the data extraction and write performance, respectively.

Table 4-1 Data extraction performance

Data Source	Specifications	Version	Extraction Rate for a Single Job (Lines per Second)	Extraction Rate for Multiple Jobs (Lines per Second)
RDS for MySQL	8 vCPUs, 32 GB	MySQL 5.7	42,052	195,313 (concurrency: 40)
Oracle	8 vCPUs, 16 GB	19C	18,539	18,706 (concurrency: 10)

Data Source	Specifications	Version	Extraction Rate for a Single Job (Lines per Second)	Extraction Rate for Multiple Jobs (Lines per Second)
MRS HBase	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	6,296	69,156 (concurrency: 30)
MRS Hive	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	22,321	170,068 (concurrency: 30)
MRS HDFS (binary files)	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	138,727	141,468 (concurrency: 20)
			125,556	126,990 (concurrency: 10)
			120,919	120,919 (concurrency: 10)
DWS	8 vCPUs, 16 GB	8.1.1.300	13,434	/
DLI	16 vCPUs	SQL queue	71,023	19,290 (concurrency: 20)

Table 4-2 Data write performance

Data Source	Specifications	Version	Write Rate for a Single Job (Rows per Second)	Optimal Write Rate (Rows per Second)
RDS for MySQL	8 vCPUs, 32 GB	MySQL 5.7	2,658	/
Oracle	8 vCPUs, 16 GB	19C	/	/

Data Source	Specifications	Version	Write Rate for a Single Job (Rows per Second)	Optimal Write Rate (Rows per Second)
MRS Hbase	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	3,959	4,120 (concurrency: 10)
MRS Hive	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	25,813	26,882 (concurrency: 10)
MRS HDFS (binary files)	Master: 16 vCPUs, 64 GB x 3 Node: 8 vCPUs, 32 GB x 3	MRS 3.1.0	65,075	90,155 (concurrency: 10)
			86,248	86,248 (concurrency: 1)
			76,687	76,687 (concurrency: 1)
DWS	8 vCPUs, 16 GB	8.1.1.300	26,624	27,902 (concurrency: 10)
DLI	16 vCPUs	SQL queue	15,211	18,430 (concurrency: 10)

