MaxCompute

Introduction

Introduction

MaxCompute Summary

MaxCompute is a big data processing platform developed by Alibaba independently. It is mainly used for batch structural data storage and processing, which can provide massive data warehouse solution and big data modeling service.

Along with the diversified data collection, more and more industrial data has been accumulated. The data size has grown up to a massive level (TB, even PB), which the traditional software industry cannot take care of. Under the analysis of massive data scenarios, the data analysts usually adopt distributed computing mode due to the limited processing capacity of the single server. But the distributed computing model demands more to the data analysis and is difficult to be maintained. Using the distributed model, data analysts not only need to understand the service requirements, but also need to be familiar with the underlying computing model.

The purpose of MaxCompute is to provide a convenient way to analyze and process big data for the user. The user is able to analyze big data without concerning details of distributed computing.

MaxCompute Ecosystem and Functional Components

MaxCompute provides tunnel for data upload and download, SQL and MapReduce for calculation and analysis service. Besides, it also provides a completed security solution.

MaxCompute Components

MaxCompute TUNNEL: provides high concurrency data upload and download services. Tunnel is a kind of service to upload and download data. User can use the Tunnel service to upload or download the data to MaxCompute. MaxCompute Tunnel only provides the Java programming interface for users.

Computing and Analysis:

MaxCompute SQL: In MaxCompute, data is stored in forms of tables. MaxCompute provides a SQL query function for the external interface. You can operate MaxCompute just like traditional database software, but still be able to process the massive data to TB or PB level. It is worth to mention that MaxCompute SQL does not support transactions, index and Update/Delete operations. MaxCompute SQL syntax differs from Oracle and MySQL, so the user cannot migrate SQL statements of other databases into MaxCompute seamlessly. In addition, MaxCompute SQL can complete the query in minutes even seconds but unable to return to result in millisecond. The advantage of MaxCompute SQL is to reduce users' learning cost and the user does not need to understand the concept of distribution. MaxCompute SQL could be understood easily by users who are familiar with database operations.

MapReduce: MapReduce is the first distributed data processing model put forward by Google. It has drawn a lot of attention and has been applied to all kinds of business scenarios. In this document, we will make a brief introduction of MapReduce model to help users quickly familiar with and understand the model. Users who use MaxCompute MapReduce must have a basic understanding of the concept of distribution and the corresponding programming experience. MaxCompute MapReduce provides Java programming interface for the users.

Graph: the graph function provided by MaxCompute is a set of iterative graph computing and processing framework. Graph computing job is modeled by graph. Graph is composed of Vertex and Edge; the vertex and edge contain weights (Value). Through iteration, edit and evolute the graph and then get the result finally. The typical apllications include: PageRank, SSSP, K-Means, etc.

SDK: Toolkit provided for the developers. For details, please refer to MaxCompute SDK.

Security: MaxCompute provides a powerful security services and provides protection for the user's data. For a full description of each function model, please refer to MaxCompute Security Manual.

Definition

Project is the basic unit of operation in MaxCompute, which sets the boundary for MaxCompute multi-users isolation and access control.In MaxCompute, all objects belong to a project. A user can have multiple project privileges. User can access the objects of another project in their own project,

such as Table, Resource, Function and Instance. User can use the command 'Use Project' to enter a project. For example:

```
use my_project  -- Enter a project named 'my_project'.
```

After running this command, you can enter a project named "my project" and all objects in this project can be operated, such as Table, Resource, Function, Instance and so on. "Use Project" is a command provided by MaxCompute client.The file will briefly describe these commands before describing this part in detail.

Table is a data storage unit in MaxCompute and all data in MaxCompute is stored in tables. Table is a two-dimensional data structure composed of rows and columns. Each row represents a record; each column represents a field with the same data type. One record can contain one or more columns. Each column name and data type consist of the schema of this table.

In MaxCompute, all data is stored in tables. The columns in the table can be any data type supported by MaxCompute (Bigint, Double, String, Boolean and Datetime). The operating objects (input, output) of various computational tasks in MaxCompute are tables. User can create a table, delete a table and import data into a table.

In order to improve the processing efficiency, you can specify the partition when creating a table. That is, several fields in the table are specified as partition columns. In most cases, user can consider the partition to be the directory under the file system. User can specify multiple partitions, that is, multiple fields of the table are considered as the partitions of the table, and the relationship among partitions is similar to that of multiple directories. If the partition columns to be accessed are specified when using data, then only corresponding partitions are read and full table scan is avoided, which can improve the processing efficiency and save costs. For detailed description, please see the description in Partition.

Data types are associated with the columns in the tables. The following types are supported:

Type	Description	Value Range
Bigint	8-byte signed integer. Please don't use the minimum value (-9223372036854775808), which is system reserved value.	-9223372036854775807 ~ 9223372036854775807
String	Support UTF-8 coding.	The threshold of single string length is allowed 2MB.
Boolean	Boolean type	True/False
Double	8-byte, double precision floating number.	-1.0 *10308* ~ *1.0* 10308

Datetime	Date type, use UTC+8 as the standard time system.	0001-01-01 00:00:00 ~ 9999-12-31 23:59:59

Note:

- All data types can be NULL.

The lifecycle of a MaxCompute table is counted from the last time the table (partition) data was updated. If the table (partition) remains unchanged after a specified time, MaxCompute automatically recycles it. The **specified time** is lifecycle.

Lifecycle units: days, positive integers only.

When a lifecycle is specified for a non-partition table, the lifecycle is counted from the last time the table data was modified (LastDataModifiedTime). If the table data is not changed after days, MaxCompute recycles the table automatically without manual operation (similar to the drop table operation).

When a lifecycle is specified for a partition table, MaxCompute determines whether to recycle the partition based on its LastDataModifiedTime. Unlike non-partition tables, a partition table cannot be deleted after all its partitions have been recycled.

You can only set the lifecycle for tables, instead of partitions. The lifecycle can be specified when the table is created.

- If no lifecycle is specified, the table (partitions) cannot be recycled by MaxCompute automatically according to lifecycle rules.

For information on specifying/modifying lifecycle during table creation and modifying a table’s LastDataModifiedTime, see DDL documentation.

Resource is a particular concept of MaxCompute. If you want to use user-defined function UDF or MapReduce, resource is needed. For example:

MaxCompute SQL UDF: After you have prepared UDF, you must upload the compiled jar package to MaxCompute as resource. While running this UDF, MaxCompute automatically downloads the jar package and gets the user’s code. To upload the jar package is a course to create MaxCompute resource, so jar package is a kind of MaxCompute recource.

MaxCompute MapReduce: After you have prepared MapReduce program, you must upload the compiled jar package as a resource to MaxCompute. While running the MapReduce job,

MapReduce framerwork automatically downloads the jar resource and gets the user’ s code. You also can upload the text file and tables as different types of resources to MaxCompute. UDF or MapReduce program can read these resources in running worker. MaxCompute provides an interface to read and use the resource. For more information, see Use Resourse Example and UDTF Usage . Note that the user-defined function UDF of MaxCompute or MapReduce has certain application restrictions on the reading of resources. See Application Restriction.

Types of MaxCompute resources include:

- File.
- Table: Tables in MaxCompute.
- Jar: Compiled Java jar package;
- Archive: Recognize the compression type according to the postfix in the resource name. The supported compressed file types include: .zip/.tgz/.tar.gz/.tar/jar.

For more information of recource, see Add Resource, Drop Resource, List Resources and Describe Resource.

MaxCompute provides SQL computational function for users. You can use System Built-in Functions in MaxCompute SQL to complete computation or count. But when the built-in functions are unable to meet the requirements, you can use Java to delevelop user defined function UDF. The user defined function (UDF) can be further divided into scalar valued function (UDF), user defined aggregation function (UDAF) and user defined table valued function (UDTF).

After UDF coding is finished, the code needs to be compiled into a jar package and uploaded to MaxCompute. While using UDF, you only need to specify the function name of UDF and input the parameters.The use method is the same as built-in functions provided by ODPS.

Note:

- For function operations, refer to Function Introduction.

Task is a basic computing unit of MaxCompute. SQL and MapReduce functions are completed by task.

For most tasks submitted by users, especially the computing tasks, such as SQL DML, MapReduce and so on, MaxCompute will analyze them and generate the task execution plan. The execution plan is composed of multiple execution stages which are dependent each other. Now, the execution plan is displayed as a directed acyclic graph; vertex in the graph designates the execution phase; while edges of graph indicate the dependence of each execution phase. MaxCompute will follow the dependency of execute plan to execute each pahse.

In an execution stage, there are muitiple processes, aslo called Worker, to complete the computting

work. Different Workers deal with different data, but the execution logic is the same. Computational tasks will be instantiated. User can operate this instance. For example, Status Instance, Kill Instance and so on.

On the other hand, some MaxCompute tasks are not computational tasks, such as DDL statement in SQL. These tasks only read and modify the metadata information in MaxCompute. Therefore, no execution plan is needed.

Note:

- Not all the requests will be converted into Tasks in MaxCompute, for example, the operations of Project, Resource, UDF and Instance can be completed without MaxCompute tasks.

In MaxCompute, some tasks will be instantiated at running time, which are existent in form of MaxCompute instances. Instances have two phases: Running and Terminated. The status of the running phase is 'Running', while the status of the end phase will be 'Success', 'Failed' or 'Canceled'. User can query or change the status according to instance ID. For example:

```
    status <instance_id>;          --Query the satatus of a certain instance.
kill <instance_id>; --Stop an instance and set its status to be 'Canceled'.
```

Are you a first-time MaxCompute user?

If you are a first-time MaxCompute user, we recommend that you begin by reading the following sections:

- MaxCompute Summary — This chapter introduces a general introduction of MaxCompute including its main function modules. By reading this chapter, you can have a general understanding of MaxCompute.

- Quick Start — This chapter adopts detailed instances step by step to guide you how to apply for an account, how to install the client, how to create a table, how to authorize for a user, how to export/import data, how to run SQL tasks, how to run UDF, and how to run Mapreduce programs, etc.

  Basic Introduction — This chapter mainly introduces some essential terms and common used commands of MaxCompute. You can be further familiar with how to operate the MaxCompute.

Tools — Before analyzing the data, you may need to master the method to dowmload, configure, and use tools. We provide the following tool:

- MaxCompute Client: You can operate the MaxCompute through this tool.

After you are familar with those modules that mentioned preceding, you are recommended to perform a further study on other modules.

For data analysts?

If you are a data analyst, you must read the following modules:

MaxCompute SQL: You can query and analyze massive data that stored on MaxCompute. The main function includes:

It supports DDL. You can manage tables and partitions through **Create**, **Drop**, and **Alter** syntaxes.

You can select a few records from a table through **Select** clause. You can query records which meet the conditions in **Where** clause.

You can achieve the association of the two tables by the equivalent connection of **Join**.

You can achieve the aggregation operation through **Group By** clause.

You can insert the result records into another table through **Insert overwrite/into** syntax.

You can achieve a series of calculations by using built-in functions and user-defined functions (UDF).

For Developers?

If you are an experienced developer and understand the concept of distributed system and some data analyzing cannot be achieved by SQL, we recommend that you learn more advanced modules of MaxCompute:

MapReduce: MaxCompute provides MapReduce programming interface. You can use the

Java API, which is provided by MapReduce, to write MapReduce program for processing data in MaxCompute.

Graph: A set of framework for iterative graph computing. Use the graph for modeling, which is composed of Vertex and Edge. Vertex and Edge include weight value (Value). Edit and evolute the graph through the iteration and get the final result.

Eclipse Plugin: Facilitate users to use Java SDK of MapReduce, UDF, and Graph.

Tunnel: You can use the Tunnel service to upload batch offline data to MaxCompute or download batch offline data from MaxCompute.

DataHub Service: You can use the DataHub service to publish and subscribe real time data.

SDK: A toolkit is available to developers.

For project owners or project administrators?

If you are a project owner or administrator, you must read:

- Security: Through reading this chapter, you can understand how to grant privileges to a user, how to share resource span projects, how to set project protection, and how to grant privilege by policy, etc.