Tcaplus Database

FAQs

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

FAQs

General FAQ

Last updated : 2020-05-22 10:37:15

What is TcaplusDB?

TcaplusDB is a distributed database developed by Tencent independently that features low cost, high availability, high reliability and high performance. TcaplusDB's design is based on the development characteristics and OPS requirements of games and provides 7x24 game data storage service that is easy to use and features high performance, low cost, high scalability, 99.999% availability and data security, so that game developers no longer worry about data storage.

What business scenarios is TcaplusDB suitable for?

TcaplusDB is created for games, including mobile games, PC games and browser games. It is also suitable for other storage scenarios, such as social network, e-commerce, and government websites and apps.

Which client languages are supported by TcaplusDB?

Linux platform: C++, Java, Erlang; Windows platform: C++, Java, Erlang. Tdr and Protobuf are provided to define app data table structure, and the object-oriented method is adopted. API network communication modes include: message-driven asynchronous operation mode, function callback asynchronous operation mode, coroutine operation mode, and dbproxy proxy mode. Game developers who adopt a language other than C++/Java can access TcaplusDB via RestAPI.

How is the performance of TcaplusDB?

When the single record size of TcaplusDB is 1 KB, the read/write hybrid model is adopted with a read and write operations ratio of 9:1, memory and disk operations ratio of 50:1, and a performance up to 150,000 QPS.

How do I use TcaplusDB?

For how to use TcaplusDB, see TcaplusDB product documents Getting Started.

How do I get the latest TcaplusDB SDK?

Please download the latest version of Tcaplus API according to your platform, such as Windows and Linux.

Does TcaplusDB support capacity expansion/reduction?

TcaplusDB's access layer and storage layer support capacity expansion/reduction, services provided will not be affected during capacity expansion/reduction.

How is the charge calculated for TcaplusDB?

See TcaplusDB Purchase Guide.

Who are using TcaplusDB?

TcaplusDB is Tencent's preferred game data storage service. Tencent's game project teams, including Arena of Valor, PUBG Mobile, CFM and QQ Speed Mobile, are all using it.

Is TcaplusDB stable?

TcaplusDB has a high availability of 99.999%. You can use it without worry.

Which regions can TcaplusDB be used?

TcaplusDB is deployed with Tencent Cloud. TcaplusDB is supported in all regions where Tencent Cloud is supported.

How is the data consistency of TcaplusDB?

The tcapsvr master and slave of TcaplusDB synchronize their data in real time, and are deployed in different IDCs in the same city, with a sync difference within 10 ms. TcaplusDB is being optimized for stronger consistency (implemented by Paxos algorithm).

How does TcaplusDB implement monitoring statistics?

TcaplusDB has a special process to collect and calculate the metrics of access layer and storage layer nodes and generate alarms, including physical machine metrics, process metrics, access traffic metrics, and latency metrics. With these metrics, problems can be identified quickly and accurately.

Usage FAQ

Last updated : 2019-02-13 10:03:09

How do I set up the read/write separation of TcaplusDB storage layer?

TcaplusDB storage layer supports read/write separation. If needed, submit a ticket to apply to use this feature.

Does TcaplusDB support data removal?

TcaplusDB supports table-level data removal based on the last write time; TcaplusDB supports key-based data removal. If needed, submit a ticket to apply to use this feature.

How should I get the definition of the error codes in a response packet?

We recommend to calling the functions of `TcapErrCode::TcapErrCodeInit` and `TcapErrCode::GetErrStr` in the gameserver to get error codes, or searching the header files of the TcaplusDB API locally.

What are the data structures of TcaplusDB?

TcaplusDB supports data structures such as list array, queries by part of keys (indexes), key-value, key-object (that is, the value of a single key can be arbitrary data structures, such as gameserver can serialize lua table into the value field).

What is TcaplusDB's optimistic locking principle and usage method?

Let's use the purchase of train tickets as an example:

1. 100 people want to snap up the same train ticket. The recorded version number of the train ticket is 10, and the 100 people use the same recorded version number to snap up the train ticket.
2. 100 people conduct write operations to this ticket. After the write operation is completed, the recorded version number will increase. Therefore, for the operation of a single key, the tcapsvr working threads are queued. When the first person successfully purchased the ticket, the recorded version number of the train ticket changed to 11, and the recorded version number of the remaining 99 people is still 10.
3. When tcapsvr handles the write requests of the 99 people, an error will occur because the recorded version number at tcapsvr end is inconsistent with the version number in the request. The concurrent principles are as follows:
   - N requests. If only the first request succeeds and the remaining N-1 requests fail, the version protection rules are used.
   - N requests. If all N requests are required to be executed, no version protection rules are required. TcaplusDB operates on the same key.

Queued to execute the call function `SetCheckDataVersionPolicy` , whose values include:

- `CHECKDATAVERSION_AUTOINCREASE` : The recorded version number is detected, which will only automatically increase if it is the same as the server-side version number.
- `NOCHECKDATAVERSION_OVERWRITE` : The recorded version number is not detected, and the recorded version number of the client is forcibly written to the server.
- `NOCHECKDATAVERSION_AUTOINCREASE` : The recorded version number is not detected, and the recorded version number of the client will automatically increase.
- You are recommended to use the default type `CHECKDATAVERSION_AUTOINCREASE` .

What are the usage scenarios and precautions for the list table?

Where there is a 1:N usage scenario, when N < 1024, the list table is prioritized, such as storing the player's most recent 100 emails, the most recent 100 battle records, and so on. The list table supports insertion at the head of the queue and removal at the end of the queue, insertion at the end of the queue and removal at the head of the queue, as well as the Top N operations sorted by insertion time. The number of units under a single key can be increased by modifying the table, because the old data needs to be compatible and cannot be modified to become smaller. You can obtain the total number of records under a single key by using listgetall. It is recommended to obtain data according to the offset and limit of a certain threshold that you set. For listreplace, listdelete, and listdeletebatch, you need to specify the correct index. For listaddafter, you need to specify the elimination rule when the number of element units under a single key reaches the upper limit. Since the SetListShiftFlag function settings are called, the list table has two increasing and two obtaining directions. There are 4 possibilities:

1. Query A, B, C, D, E as offset = positive number and limit = 2, and the result is A, B; C, D; E.
2. Query A, B, C, D, E as offset = negative number and limit = 2, and the result is D, E; B, C; A.
3. Query E, D, C, B, A as offset = positive number and limit = 2, and the result is E, D; C, B; A.
4. Query E, D, C, B, A as offset = negative number and limit = 2, and the result is B, A; D, C; E.

In addition, calling `GetRecordMatchCount` can get the total number of records.

What is the difference between insert, update and replace?

For insert operation, when the key does not exist, an insert operation is performed; when the key exists, an error code is returned: `TcapErrCode::SVR_ERR_FAIL_RECORD_EXIST` .
For replace operation, when the key does not exist, an insert operation is performed; when the key exists, if the optimistic lock is used, different operations are performed based on the result of the optimistic lock. If the operation is successful, the replace operation is performed, and if it fails, an error code is returned: `TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION;` . If optimistic locking is not used, the key will be replaced.

For update operation, when the key exists, if the optimistic lock is used, different operations are performed based on the result of the optimistic lock. If the operation is successful, the update operation is performed, and if it fails, an error code is returned: `TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION` . If optimistic locking is not used, the key will be updated. when the key does not exist, the error code is returned: `TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST` .

How do I get the number of records in a table?

There is a count command word in the TcaplusDB API. If tcaplus_client is used, you can use the count table name command to get the number of records in the table.

Does TcaplusDB support traversal operations?

TcaplusDB supports traversal operations, including traversal operations for generic and list tables. When traversing, set to traverse the data from the tcapsvr slave (traversing the data from tcapsvr slave will not affect the external services provided by tcapsvr master), i.e. API: `SetOnlyReadFromSlave(bool flag)` .

Does TcaplusDB support updating and obtaining partial fields?

TcaplusDB support updating partial fields. When updating and obtaining records, it is recommended to explicitly call API `SetFieldNames(IN const char* field_name[], IN const unsigned field_count)` to determine the fields of this read and write operation, and reduce the network traffic overhead caused by invalid fields.

Is TcaplusDB order-preserving for the continuous operations of a single key?

For the same gameserver, the operations of the same key are order-preserving, while the operations of different keys are not order-preserving. For different gameservers, the order is not preserved.

Does TcaplusDB support table definition changes?

TcaplusDB supports table definition changes. If you simply add value fields and modify macros, use table change operations; for the rest of the scenarios, you need to dynamically modify the table structure. That is, use the data migration + Ulog process to achieve table definition changes. Submit a ticket to apply to use this feature.

How does the gameserver kick out an invalid tcaproxy (access layer) node?

TcaplusDB API does disaster recovery processes for tcaproxy exceptions here. There are two main ways in which the API kicks out invalid tcaproxy processes:

1. The API physically considers that a tcaproxy is not available. The API sends heartbeat detection packets to all connected tcaproxy every 1 second. If a gameserver does not receive the

corresponding heartbeat return packets from tcaproxy within 10 seconds, the API will actively disconnect the TCP link to tcaproxy and actively link the tcaproxy at the next onupdate.

2. API logically considers that a tcaproxy is not available. It calculates the request and response ratio of a tcaproxy every 10 seconds as a basis for judgment. The time out threshold for the API to a request packet is 3s. If it times out more than 3 times, the tcaproxy is considered unavailable and the request will not be sent to the tcaproxy. A getmetdata request is sent after 60s. If tcaproxy can correctly handle the getmetadata request , the API considers the tcaproxy available and the request is sent to the tcaproxy again.

If the gameserver finds that a tcaproxy is not available within 10s, it will not send data to the tcaproxy node.

How does the gameserver choose the tcaproxy (access layer) node?

The gameserver maintains a consistent Hash ring locally. Once a tcaproxy (access layer) node has been verified, it will be added to the Hash ring. If a tcaproxy (access layer) node reduces capacity or the TCP link between gameserver and tcaproxy (access layer) has been disconnected due to machine abnormalities, the gameserver will remove the tcaproxy (access layer) node from the Hash ring. The gameserver calculates hash values based on the primary key in the request (if it is a batchget request, it randomly selects a single tcaproxy (access layer) node), and then selects a single tcaproxy (access layer) node to send out on the consistent Hash ring.

How do I judge whether the packeting of response packet has ended?

For traversal, judge whether the traversal ends according to state, that is, API GetState. For the rest of packeting scenarios, judge whether the packeting ends according to the function `HaveMoreResPkgs` .

What is the difference between GetRecordCount and GetRecordMatchCount?

A request may have N response packets. If there are multiple packets, GetRecordCount refers to the number of records in the response packet, and GetRecordMatchCount refers to the data records stored on the tcapsvr (storage layer) end (total number of records for a single key).

Does TcaplusDB have a pass through field?

The CS protocol of TcaplusDB is divided into two parts: Head and Body. UserBuff (maximum size is 1 KB), AsyncID and Sequence in Head are all pass through fields. You can use them accordingly.

What is the role of SetResultFlag?

When performing write operations, the response packet supports returning records. When performing read operations, calling this function is invalid. The description of the specific value of

result_flag is as follows:
0 refers to: Simply return whether the operation is successful or not without returning the value field.
1 refers to: Return data consistent with the request field.
2 refers to: The latest data for all fields of the change record must be returned.
3 refers to: The old data for all fields of the change record must be returned.

The SetResultFlagForSuccess API can set the data returned if the operation succeeded; the SetResultFlagForFail API can set the data returned if the operation failed.

Can I perform an increase operation on multiple value fields at a time? What if the key does not exist?

The increase operation can increase multiple value fields at a time, which requires the request passed by gameserver to assign values to multiple fields. If one of the keys does not exist when the increase operation is performed, it can be set by the SetAddableIncreaseFlag function. If the key does not exist, it will insert the key and perform increase operation. The non-increase field of the key will not be actually stored, and the default value of the non-increase field will be used when the record is read; if the key exists, the increase operation will be performed directly.

Does TcaplusDB have compression function?

TcaplusDB has a compression feature, and the compression algorithm used is Google snappy, including protocol compression, which is the compression feature of request packet/response packet between gameserver <--> tcaproxy (access layer), and data compression, that is, tcapsvr (storage layer) compresses the data that needs to be stored when the data is stored. If you want to reduce the network traffic between gameserver <--> tcaproxy, it is recommended that you enable protocol compression. If you want to call the SetCompressSwitch function of TcaplusDB API, it is recommended that you enable tcapsvr (storage layer) compression, which can save disk space and improve IO disk performance, while the CPU used for compression and decompression is also controllable.

What are the ways to reduce traffic when TcaplusDB reads records?

When TcaplusDB reads a record, it can be set so it does not return the value field if the record has no change in a fixed period of time, neither does it return the value field if the record version number does not change. For more information, see the SetFlags function.

Is the TcaplusDB API thread-safe?

TcaplusDB API is non-thread-safe, mainly because components such as tlog and tdr are not thread-safe. It is recommended that a single thread uses a single API object and a single game area uses a single API object. If you need to interact across game zones, it is recommended to maintain multiple API objects with a single gameserver.

How does tcaproxy (access layer) achieve disaster recovery?

Tcaproxy (access layer) adopts a peer-to-peer design scheme, that is, all tcaproxy (access layer) nodes under a single game zone contain routing information of all tables under a single game zone. If a tcaproxy (access layer) fails, as long as the remaining tcaproxy (access layer) nodes are not overloaded, the gameserver will kick out the abnormal tcaproxy (access layer) nodes, which will not affect the use of gameserver. There is no single point of failure risk for the tcaproxy (access layer).

How does tcapsvr (storage layer) achieve disaster recovery?

The tcapsvr (storage layer) runs in a master-slave mode (tcapsvr master - tcapsvr slave). Tcapsvr master/slave synchronize data in real time, and are deployed at the different IDCs in the same city, ensuring that master-slave synchronization latency is less than 10 ms. If the tcapsvr slave is abnormal, it will not affect the use of gameserver (if the read traffic diversion is not enabled, the requests of gameserver are processed by the tcapsvr master. If the read traffic diversion is enabled, the tcapsvr slave will assist in processing part of the read requests), and the DBA will rebuild the tcapsvr slave; if the tcapsvr master is abnormal, the tcapsvr slave will perform failure recovery and the DBA will apply for a new machine to rebuild the tcapsvr slave. There is no single point of failure risk for the tcapsvr (storage layer).

Does TcaplusDB support rollbacks? What is the supported rollback granularity?

TcaplusDB supports rollbacks, including all-server/all-region rollback, single-table rollback, and can roll back N records from 100 billion records. It also supports cold standby time rollback (most recent 01:05:00 am), precise time rollback (to the second), and fuzzy rollback (you can specify the rollback rules). For speed reference, precise time rollback takes about 2 hours for a rollback of 300 GB data and 200 GB Ulog processes. The principle of the cold standby time rollback is to replace the engine file. The principle of precise time rollback is to roll back the cold standby engine file + Ulog process to the needed time point. A key-based rollback requires your cooperation. You need to block these keys first and then unblock them after TcaplusDB has finished the rollback.

What are the security measures for TcaplusDB?

If TcaplusDB is deployed in an IDC environment, it will be deployed directly in the private network with physical security isolation. If TcaplusDB is deployed in a non-IDC environment, the API performs identity verification through AppKey to prevent illegal data access. The Protobuf API provides the ability to encrypt, decrypt, and store String and Bytes fields.

Does TcaplusDB have overload protection?

Both the access layer and the storage layer have process-level overload protection measures to ensure that services will not collapse during peak hours.

What is the single instance memory capacity and CPU utilization of the TcaplusDB API?

The API's maximum single-instance memory consumption is 73 MB, and the maximum CPU utilization is 30%.

What is the number of tables in the single game zone of TcaplusDB?

The maximum number of tables in the single game zone of TcaplusDB is 256. If the number of tables under a single game zone is greater than 256, you can add new game zones or merge the tables. If technical support is required, please submit a ticket.

What are the restrictions on the key and value fields in TcaplusDB?

The number of key fields of the generic table is 4, the number of key fields of the list table is 3, and the size of a single key field is 1024 B. The number of value fields of the generic table is 128, the number of value fields of the list table is 127, the size of a single value field is 256 KB, and the maximum value of the overall record is 1 MB.

How efficient is queries by partial keys (indexes) with TcaplusDB?

It is recommended to use queries by partial keys (indexes) in 1:N (N > 1024) app scenarios. The number of primary keys under a single index key is equal to 10 GB/the size of primary key of a single record. A single read-write index operation takes about 100 ms (there are more than 100,000 pieces of data records under a single index key).

What is the principle of cold and hot data exchange for TcaplusDB?

TcaplusDB uses memory + SSD disk storage, a single engine file, and the first 1 GB is mapped in memory. The hot data is placed in the memory, the cold data is placed on the disk, and the LRU algorithm is used for hot and cold data exchange. The get operation of gameserver triggers the LRU swap-in operation. The LRU thread of tcapsvr (storage layer) is responsible for the LRU swap-out operation. Try to ensure that the hot data is stored in the memory, thus ensuring high cache hit rate and low single read and write latency.

How long is the TcaplusDB backup file stored?

The engine files backed up by TcaplusDB are saved for 25 days, and the Ulog process is saved for 15 days. The storage periods in overseas environment and other TcaplusDB environments are different. For the storage period of the specific TcaplusDB environment, you can contact customer service.

Is the gameserver connected to all tcaproxy (access layer)?

In order to reduce the cost of maintaining the TCP connections between the gameserver and tcaproxy (access layer), the gameserver supports selecting some tcaproxy (access layer) to establish connections.

How do I perform data analysis of TcaplusDB data?

TcaplusDB data can be exported in any format, including json, pb and other formats, and can be imported into data analysis systems such as tdw. TcaplusDB supports the import of real-time data into MySQL database and so on.

What is the timeout mechanism for the TcaplusDB API? What does "it is timeout" error mean?

The TcaplusDB API assigns an ID to each request. After it is successfully sent, the ID is pushed into the judgment timeout data structure. If the response packet for the request returns, the ID is removed from the data structure. If the response packet of the request has not been processed by the gameserver within 3 seconds, an error log with the keyword "it is timeout" is printed. You will then need to check whether the gameserver is blocked. It may be that tcaproxy (access layer) dropped the packets when returning packets to the gameserver, or the response packet has arrived at the gameserver, but the gameserver did not process it in time.
TcaplusDB recommends that you implement the timeout mechanism yourself, which allows you to perform retry and other processes to timeout requests.

What are the roles of SendRequest, OnUpdate, and ReceiveResponse functions of TcaplusDB API?

The role of SendRequest is to send requests. This request may have been sent to the network, or it may be blocked in the sending channel of the gameserver and a tcaproxy (access layer). The role of ReceiveResponse is to get the response packet from the local receive queue. OnUpdate is responsible for sending the requests of the send queue to the network and receiving response packets from the network to the receive queue. In message-driven programming mode, the recommended call frequency for OnUpdate calls is 1 ms.

How should I resolve TcaplusDB write operation process?

TcaplusDB records Ulog processes for all write operations. If you need to resolve Ulog processes, please submit a ticket.

ow does TcaplusDB achieve global auto increase of fields?

You need to define a single table, and set the field type of a single key and a single value to int64_t (multiple value fields can implement counter arrays). Multiple gameservers can increase a single key concurrently (without setting the version protection rules), and get the increase result returned for this operation, then the result of the increase will be a global auto increase.

What rules are defined for queries by partial keys (indexes)?

The index key must be a part of the primary key, the index key must contain the sub-table factor, and the index key cannot be the primary key.

How do I achieve two-way query of the id and name of a single table?

The third key field x is used as the sub-table factor, and the index is built on id and x, name and x to achieve this feature. For example, when storing player information, the player's provincial address can be added, that is, the primary key is uin, name, provincial address, and the index key is uin and provincial address, and name and provincial address.

Does table cleanup delete data?

Table cleanup deletes data. If the table is configured to have cache write data, you can select whether to delete the cache write data at the same time.

How does tcaproxy (access layer) choose tcapsvr (storage layer)?

Each table defines a sub-table factor. If no sub-table is defined, the sub-table factor defaults to the primary key. Tcaproxy (access layer) selects the corresponding tcapsvr (storage layer) according to hash (sub-table factor) %1w, so the dispersion of the sub-table factor needs to be high.

What is the maximum size of a single table in TcaplusDB? What is the limit on the number of records?

A single table can be subdivided into 10,000 data shard, each data shard is 256 GB, that is, the total size of a single table is 10,000 * 256 GB. A single table has no limit on the number of records, and the number of records in a single table is related to the size of a single record.

Does tcaplus_client support the display of secondary fields and above?

Yes. You may execute help select to see how select * into a.xml is used.

What is the level of the lock in TcaplusDB?

The granularity of the lock in TcaplusDB is logging level.

Does TcaplusDB support multi-table transaction operations and batch write operations?

TcaplusDB does not support multi-table transaction operations, that is, when multiple operations are submitted at the same time, all completed operations are rolled back if the subsequent operations fail. In such situations, changes has to be made on how app operations are performed. Operations has to be done separately in sequence, for example, the deduction operation is performed first and then the add operation is performed, or record a reconciliation log of important operations. Currently, TcaplusDB also does not support batch write operations.

What should be noted when changing the table in TcaplusDB?

1. You can only add new fields, but you cannot modify the type and name of an existing field, or delete an existing field. If you need to modify these, you need to dynamically modify the table structure.
2. The length of an array or string in the value field can be increased but not reduced. If you need to modify these, you need to dynamically modify the table structure.
3. The version number corresponding to the new field needs to be incremented by 1 from the original version number. In correspondence, the version number of the entire xml file (defined at the top of the file) also needs to be incremented by 1 (consistent with the version number of the new field). You need to make changes to the table in TcaplusDB first, and then make changes to the table in the gameserver. You can dynamically modify the table structure in TcaplusDB if needed, please submit a ticket to do this.

What should be noted when defining the table in TcaplusDB?

Note the following when defining a table in TcaplusDB:

1. The refer field needs to be added to the count field.
2. The sub-table factors requires high dispersion.
3. The index key cannot be the same as the primary key.

What preparations are needed for TcaplusDB before the launch of the app?

Before the launch of the app, you will need to perform stress tests, failure recovery of tcapsvr (storage layer), expansion and reduction of tcaproxy (access layer), expansion and reduction of tcapsvr (storage layer), rollback and overload protection.

What is the routine OPS time of TcaplusDB?

TcaplusDB carries out data migration, data removal, data backup and other OPS operations at 01:00-07:00 daily. We recommend that you ignore the physical indicator warnings of the server during these periods.

Is the TcaplusDB API upgrade forward compatible?

TcaplusDB API upgrade is forward compatible, and existing APIs, command words, and features will not be modified.

What is the speed of expansion and reduction of tcaproxy (access layer)?

Tcaproxy (access layer) expansion and reduction does not require data migration, so the expansion and reduction time is only the time required to start and stop the service, which only takes seconds to complete.