

TencentDB for TcaplusDB

Perguntas frequentes

Product Documentation



Tencent Cloud

[Declaração de direitos autorais]

Direitos autorais ©2013–2026 Tencent Cloud. Todos os direitos reservados.

Os direitos autorais deste documento pertencem exclusivamente à Tencent Cloud. Sem a autorização prévia por escrito da Tencent Cloud, nenhuma entidade pode copiar, modificar, plagiar ou disseminar todo ou parte do conteúdo deste documento, sob qualquer forma.

[Declaração de marca registrada]

e outras marcas registradas relacionadas aos Serviços da Tencent Cloud são de propriedade das entidades relevantes de empresas sob o Grupo Tencent. Além disso, as marcas registradas de entidades terceirizadas envolvidas neste documento são de propriedade dos detentores de direitos, de acordo com a lei.

[Declaração de serviço]

Este documento tem como objetivo fornecer aos clientes uma visão geral de todos ou parte dos produtos e serviços da Tencent Cloud no momento. O conteúdo de alguns produtos e serviços pode ser ajustado. Os tipos de produtos, serviços e os padrões de serviço da Tencent Cloud que você adquirir deverão ser acordados pelo contrato comercial entre você e a Tencent Cloud. Salvo acordo em contrário entre ambas as partes, a Tencent Cloud não assume nenhuma promessa ou garantia, expressa ou implícita, com relação ao conteúdo deste documento.

Contents

Perguntas frequentes

Funcionalidades do banco de dados

Uso do banco de dados

Princípios do banco de dados

Perguntas frequentes

Funcionalidades do banco de dados

Last updated: 2024-12-04 10:27:05

- [O TcaplusDB permite a remoção de dados?](#)
- [Quais são as estruturas de dados do TcaplusDB?](#)
- [Qual é a capacidade de memória de instância única e a utilização da CPU do SDK do TcaplusDB?](#)
- [Quantas tabelas há em um grupo de tabelas no TcaplusDB?](#)
- [Quais são as restrições nos campos de chave e valor no TcaplusDB?](#)
- [Por quanto tempo o arquivo de backup do TcaplusDB é retido?](#)
- [O servidor do jogo está conectado a todos os tcaproxy \(camada de acesso\)?](#)
- [Como realizo a análise dos dados do TcaplusDB?](#)
- [Qual é o tamanho máximo de uma única tabela no TcaplusDB? Qual é o limite da quantidade de registros?](#)
- [O TcaplusDB permite operações de transações de várias tabelas e operações de gravação em lote?](#)
- [A atualização da API do TcaplusDB é compatível com versões anteriores?](#)

O TcaplusDB permite a remoção de dados?

O TcaplusDB permite a remoção de dados em nível de tabela, em que os dados são removidos de acordo com a última vez que foram gravados.

Quais são as estruturas de dados do TcaplusDB?

O TcaplusDB aceita estruturas de dados como matriz de lista, consultas por parte de chaves (índices), chave-valor, chave-objeto (isto é, o valor de uma única chave pode ser estruturas de dados arbitrárias, por exemplo, o servidor do jogo pode serializar `lua table` no campo de valor).

Qual é a capacidade de memória de instância única e a utilização da CPU do SDK do TcaplusDB?

O consumo máximo de memória de instância única é 73 MB e a utilização máxima da CPU é 30%.

Quantas tabelas há em um grupo de tabelas no TcaplusDB?

No TcaplusDB, um grupo de tabelas pode ter até 256 tabelas. Se houver mais de 256 tabelas em um grupo, você pode adicionar um novo grupo de tabelas ou mesclar as tabelas. Se você precisar de ajuda técnica, [envie um tíquete](#) e selecione **Other (Outro)** na página **Select the related product (Selecione o produto relacionado)**.

Quais são as restrições nos campos de chave e valor no TcaplusDB?

A quantidade de campos de chave da tabela genérica é 4, a quantidade de campos de chave da tabela de lista é 3 e o tamanho de um único campo de chave é 1.024 B. A quantidade de campos de valor da tabela genérica é 128, a quantidade de campos de valor da tabela de lista é 127, o tamanho de um único campo de valor é 256 KB e o tamanho máximo de um registro é 1 MB.

Por quanto tempo o arquivo de backup do TcaplusDB é retido?

Os arquivos do mecanismo com backup pelo TcaplusDB são retidos por 7 dias, e o Ulog é salvo por 7 dias. Os períodos de retenção variam com o ambiente do TcaplusDB. Para obter o período de retenção de um ambiente do TcaplusDB específico, você pode [entrar em contato com o atendimento ao cliente](#).

O servidor do jogo está conectado a todos os tcaproxy (camada de acesso)?

Para reduzir o custo de manutenção das conexões TCP entre o servidor do jogo e o tcaproxy (camada de acesso), o servidor do jogo permite a seleção de alguns tcaproxy (camada de acesso) para estabelecer as conexões.

Como realizo a análise dos dados do TcaplusDB?

Os dados do TcaplusDB podem ser exportados em qualquer formato, incluindo json, pb e outros formatos, e podem ser importados para sistemas de análise de dados como o TDW. O TcaplusDB permite a importação de dados em tempo real em bancos de dados MySQL, etc.

Qual é o tamanho máximo de uma única tabela no TcaplusDB? Qual é o limite da quantidade de registros?

Uma única tabela pode ser subdividida em 10.000 fragmentos de dados, cada fragmento de dados tem 256 GB, ou seja, o tamanho total de uma única tabela é $10.000 * 256$ GB. Uma única tabela não tem limite para a quantidade de registros, e a quantidade de registros em uma única tabela está relacionada ao tamanho de um único registro.

O TcaplusDB permite operações de transações de várias tabelas e operações de gravação em lote?

O TcaplusDB não permite operações de transações de várias tabelas nem operações de gravação em lote. Para realizar operações de transações de várias tabelas, as alterações devem ser feitas no seu negócio. No TcaplusDB, as operações devem ser feitas em sequência. Por exemplo, quando várias operações são enviadas ao mesmo tempo, todas as operações concluídas são revertidas se uma das operações posteriores falhar. Após a reversão, você pode enviar essas operações novamente. Para operações importantes, recomendamos manter os logs.

A atualização da API do TcaplusDB é compatível com versões anteriores?

A atualização da API do TcaplusDB é compatível com versões anteriores, e as APIs, as palavras de comando e as funcionalidades existentes não serão modificadas.

Uso do banco de dados

Last updated: 2025-10-16 21:20:48

- Como devo obter a definição dos códigos de erro em um pacote de resposta?
- Como funciona o bloqueio otimista do TcaplusDB e como usá-lo?
- Quais são os casos de uso e as precauções para a tabela LIST?
- Qual é a diferença entre INSERT, UPDATE e REPLACE?
- Como obtenho a quantidade de registros em uma tabela?
- O TcaplusDB aceita operações de travessia?
- O TcaplusDB aceita a atualização e a obtenção de campos parciais?
- O TcaplusDB está preservando a ordem para as operações contínuas de uma única chave primária?
- O TcaplusDB aceita alterações na definição da tabela?
- Como posso saber se o empacotamento do pacote de resposta terminou?
- Qual é a diferença entre GetRecordCount e GetRecordMatchCount?
- O TcaplusDB tem um campo de passagem?
- Qual é a função de setResultFlag?
- Posso realizar operações de aumento em vários campos de chave não primária de uma vez? E se a chave primária não existir?
- Como faço para reduzir os custos de tráfego quando o TcaplusDB lê os registros?
- O TcaplusDB aceita reversões? Qual é a granularidade de reversão aceita?
- Qual é o nível de eficiência das consultas por chaves parciais (índices) com o TcaplusDB?
- Qual é o mecanismo de tempo limite para a API do TcaplusDB? O que significa o erro "it is timeout (o tempo limite foi atingido)"?
- Quais são os papéis das funções SendRequest, OnUpdate e ReceiveResponse da API do TcaplusDB?
- Como o TcaplusDB realiza aumento automático global de campos?
- Quais regras são definidas para consultas por chaves parciais (índices)?
- Como faço para obter uma consulta bidirecional de ID e nome de uma única tabela?
- A tcaplus_client aceita a exibição de campos no segundo nível de aninhamento ou acima?
- O que deve ser observado ao alterar a tabela no TcaplusDB?
- O que deve ser observado ao definir a tabela no TcaplusDB?

Como devo obter a definição dos códigos de erro em um pacote de resposta?

Recomendamos chamar as funções de `TcapErrCode::TcapErrCodeInit` e `TcapErrCode::GetErrStr` no servidor do jogo para obter os códigos de erro, ou pesquisar os arquivos de cabeçalho da API do

TcaplusDB localmente.

Como funciona o bloqueio otimista do TcaplusDB e como usá-lo?

Vamos usar a compra de passagens de trem como exemplo:

- 100 pessoas querem comprar a mesma passagem de trem. O número da versão registrada da passagem de trem é 10, e as 100 pessoas usam o mesmo número de versão registrado para comprar a passagem de trem.
- 100 pessoas realizam operações de gravação nesta passagem. Depois que a operação de gravação for concluída, o número da versão registrada aumentará. Portanto, para a operação de uma única chave, as threads de trabalho tcapsvr são enfileiradas. Quando a primeira pessoa comprou a passagem com êxito, o número da versão registrada da passagem de trem mudou para 11, e o número da versão registrada das 99 pessoas restantes ainda é 10.
- Quando o tcapsvr lidar com as solicitações de gravação das 99 pessoas, ocorrerá um erro porque o número da versão registrada no final de tcapsvr é inconsistente com o número da versão na solicitação. Os princípios concorrentes são os seguintes:
 - N solicitações. Se apenas a primeira solicitação for bem-sucedida e as N-1 solicitações restantes falharem, as regras de proteção de versão serão usadas.
 - N solicitações. Se todas as solicitações N tiverem que ser executadas, nenhuma regra de proteção de versão será necessária. O TcaplusDB opera na mesma chave.

Enfileire e chame a função `SetCheckDataVersionPolicy`, cujos valores incluem:

- `CHECKDATAVERSION_AUTOINCREASE`: o número da versão registrada é detectado, o que só aumentará automaticamente se for o mesmo que o número da versão do servidor.
- `NOCHECKDATAVERSION_OVERWRITE`: o número da versão registrada não é detectado e o número da versão registrada do cliente é gravado de forma forçada no servidor.
- `NOCHECKDATAVERSION_AUTOINCREASE`: o número da versão registrada não é detectado e o número da versão registrada do servidor aumentará automaticamente.
- É recomendável que você use o tipo padrão `CHECKDATAVERSION_AUTOINCREASE`.

Quais são os casos de uso e as precauções para a tabela LIST?

Onde há um caso de uso 1:N, quando $N < 1024$, a tabela LIST é priorizada, como armazenar os 100 e-mails mais recentes do jogador, os 100 registros de batalha mais recentes e assim por diante. A tabela LIST aceita a inserção no início da fila e a remoção no final da fila, a inserção no final da fila e a remoção no início da fila, assim como as N operações principais classificadas por tempo de inserção. A quantidade de unidades em uma única chave pode ser aumentada modificando a tabela, porque os dados antigos precisam ser compatíveis e não podem ser modificados para se tornarem menores. Você pode obter a quantidade total de registros em uma única chave usando `listgetall`. É recomendável obter dados de acordo com o deslocamento e um limite que você definiu. Para `listreplace`, `listdelete` e `listdeletebatch`, você precisa especificar o índice correto. Para `listaddafter`, você precisa especificar a regra de

eliminação quando a quantidade de unidades de elemento em uma única chave atinge o limite superior. Se a função `SetListShiftFlag` for chamada, a tabela LIST terá duas direções crescentes e duas direções de obtenção. Existem 4 possibilidades:

1. Consulte A, B, C, D, E como deslocamento = número positivo e limite = 2, e o resultado é A, B; C, D; E.
2. Consulte A, B, C, D, E como deslocamento = número negativo e limite = 2, e o resultado é D, E; B, C; A.
3. Consulte E, D, C, B, A como deslocamento = número positivo e limite = 2, e o resultado é E, D; C, B; A.
4. Consulte E, D, C, B, A como deslocamento = número negativo e limite = 2, e o resultado é B, A; D, C; E.

Além disso, chamar `GetRecordMatchCount` pode obter a quantidade total de registros.

Qual é a diferença entre INSERT, UPDATE e REPLACE?

Para a operação INSERT, quando a chave não existe, uma operação INSERT é executada; quando a chave existe, um código de erro é retornado: `TcapErrCode::SVR_ERR_FAIL_RECORD_EXIST`.

Para a operação REPLACE, quando a chave não existe, uma operação INSERT é executada; quando a chave existe, se o bloqueio otimista é usado, operações diferentes são executadas com base no resultado do bloqueio otimista. Se a operação for bem-sucedida, a operação REPLACE será executada e, se falhar, um código de erro será retornado: `TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION;`. Se o bloqueio otimista não for usado, a operação REPLACE será executada.

Para a operação UPDATE, quando a chave existe, se o bloqueio otimista é usado, diferentes operações são executadas com base no resultado do bloqueio otimista. Se a operação for bem-sucedida, a operação UPDATE será executada e, se falhar, um código de erro será retornado: `TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION`. Se o bloqueio otimista não for usado, a operação UPDATE será executada. Quando a chave não existe, um código de erro é retornado: `TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST`.

Como obtenho a quantidade de registros em uma tabela?

Há uma palavra de comando de contagem na API do TcaplusDB. Se `tcaplus_client` for usado, você pode usar o comando de nome da tabela `count` para obter a quantidade de registros na tabela.

O TcaplusDB aceita operações de travessia?

O TcaplusDB aceita operações de travessia, incluindo operações de travessia para tabelas GENERIC e LIST. Você pode usar a API `SetOnlyReadFromSlave(bool flag)` para percorrer os dados do tcapsvr secundário, o que não afetará os serviços fornecidos pelo tcapsvr principal.

O TcaplusDB aceita a atualização e a obtenção de campos parciais?

O TcaplusDB aceita a atualização de campos parciais. Ao atualizar e obter registros, é recomendado chamar explicitamente a API `SetFieldNames(IN const char* field_name[], IN const unsigned field_count)` para determinar os campos desta operação de leitura e gravação e reduzir a sobrecarga de tráfego de rede causada por campos inválidos.

O TcaplusDB está preservando a ordem para as operações contínuas de uma única chave primária?

Para o mesmo servidor de jogo, as operações da mesma chave primária preservam a ordem, já as operações de chaves primárias diferentes não preservam a ordem. Para servidores de jogo diferentes, a ordem não é preservada.

O TcaplusDB aceita alterações na definição da tabela?

O TcaplusDB aceita alterações na definição da tabela. Para adicionar campos de chave não primária e alterar macros, basta alterar a tabela. Para alterações mais complexas, você pode alterar a estrutura da tabela continuamente usando a migração de dados e os logs. Para usar essa funcionalidade, [envie um tíquete](#) e selecione **Other (Outro)** na página **Select the related product (Selecione o produto relacionado)**.

Como posso saber se o empacotamento do pacote de resposta terminou?

Para a travessia, verifique se a travessia termina de acordo com o estado, ou seja, a API `GetState`. Para o resto dos cenários de empacotamento, verifique se o empacotamento termina de acordo com a função `HaveMoreResPkgs`.

Qual é a diferença entre `GetRecordCount` e `GetRecordMatchCount`?

Uma solicitação pode ter N pacotes de resposta. Se houver vários pacotes, `GetRecordCount` se refere à quantidade de registros no pacote de resposta, e `GetRecordMatchCount` se refere aos registros de dados armazenados no tcapsvr (camada de armazenamento) (quantidade total de registros para uma única chave).

O TcaplusDB tem um campo de passagem?

O protocolo CS do TcaplusDB está dividido em duas partes: cabeçalho e corpo. `UserBuff` (tamanho máximo é 1 KB), `AsyncID` e `Sequence` no cabeçalho são todos campos de passagem. Você pode usá-los adequadamente.

Qual é a função de `SetResultFlag`?

Ao executar operações de gravação, o pacote de resposta permite o retorno de registros. Ao realizar operações de leitura, chamar esta função é inválido. Os valores específicos de `result_flag` são os seguintes:

0: retorna apenas se a operação foi bem-sucedida ou não sem retornar o campo de valor.

1: retorna dados consistentes com o campo de solicitação.

2: retorna os dados mais recentes para todos os campos do registro alterado.

3: retorna os dados antigos para todos os campos do registro alterado.

A API `SetResultFlagForSuccess` pode ser usada para retornar os dados se a operação for bem-sucedida; a API `SetResultFlagForFail` pode ser usada para retornar os dados se a operação falhar.

Posso realizar operações de aumento em vários campos de chave não primária de uma vez? E se a chave primária não existir?

Para aumentar vários campos de chave não primária, esses campos precisam ser atribuídos com valores pela solicitação do servidor do jogo. Se uma das chaves não existir, você pode usar a função `SetAddableIncreaseFlag` para executar operações de aumento. Se não existir nenhuma chave, insira uma chave e, em seguida, execute operações de aumento, em que os campos não aumentados da chave não serão armazenados e assumirão o valor padrão quando o registro for lido. Se a chave existir, a operação de aumento pode ser executada imediatamente.

Como faço para reduzir os custos de tráfego quando o TcaplusDB lê os registros?

Quando o TcaplusDB lê um registro, ele pode ser configurado para não retornar o campo de valor se o registro não tiver alteração em um período fixo, nem retornar o campo de valor se o número da versão do registro não mudar. Para obter mais informações, consulte a função `SetFlags`.

O TcaplusDB aceita reversões? Qual é a granularidade de reversão aceita?

O TcaplusDB aceita reversões, incluindo reversão de todos os servidores/regiões, reversão de tabela única e pode reverter N registros de 100 bilhões de registros. Ele também aceita reversão manual de tempo de espera (mais recente 01:05:00), reversão de tempo precisa (em segundos) e reversão difusa (você pode especificar as regras de reversão). Para referência de velocidade, a reversão de tempo precisa leva cerca de 2 horas para uma reversão de dados de 300 GB e Ulog de 200 GB. O princípio da reversão manual de tempo de espera é substituir o arquivo do mecanismo. O princípio da reversão de tempo precisa é reverter o arquivo do mecanismo de tempo de espera manual + Ulog para o ponto de tempo necessário. Uma reversão baseada em chave requer que você bloqueie essas chaves primeiro e, em seguida, desbloqueie-as após o TcaplusDB terminar a reversão.

Qual é o nível de eficiência das consultas por chaves parciais (índices) com o TcaplusDB?

É recomendável usar consultas por chaves parciais (índices) em cenários de aplicação 1:N (N > 1024). A quantidade de chaves primárias em uma única chave de índice é igual a 10 GB/o tamanho da chave primária de um único registro. Uma única operação de índice de leitura e gravação leva cerca de 100 ms (há mais de 100.000 partes de registros de dados em uma única chave de índice).

Qual é o mecanismo de tempo limite para a API do TcaplusDB? O que significa o erro "it is timeout (o tempo limite foi atingido)"?

A API do TcaplusDB atribui um ID a cada solicitação. Depois de ser enviado com sucesso, o ID é enviado para a estrutura de avaliação de tempo limite. Se o pacote de resposta para a solicitação retornar, o ID será removido da estrutura. Se o pacote de resposta da solicitação não tiver sido processado pela camada de aplicativos em 3 segundos, um log de erro com a palavra-chave "it is timeout (o tempo limite foi atingido)" é

impresso. Você precisará verificar se o servidor do jogo está bloqueado. Pode ser que o tcaproxy (camada de acesso) tenha removido os pacotes ao devolvê-los ao servidor do jogo, ou o pacote de resposta chegou ao servidor do jogo, mas o servidor do jogo não o processou a tempo.

O TcaplusDB recomenda que você mesmo implemente o mecanismo de tempo limite, que permite realizar novas tentativas e outros processos para solicitações de tempo limite.

Quais são os papéis das funções `SendRequest`, `OnUpdate` e `ReceiveResponse` da API do TcaplusDB?

O papel de `SendRequest` é enviar solicitações. Esta solicitação pode ter sido enviada para a rede, ou pode estar bloqueada no canal de envio do servidor do jogo e um tcaproxy (camada de acesso). O papel de `ReceiveResponse` é obter o pacote de resposta da fila de recebimento local. O `OnUpdate` é responsável por enviar as solicitações da fila de envio para a rede e receber os pacotes de resposta da rede para a fila de recebimento. No modo de programação orientado por mensagens, o `OnUpdate` é recomendado para ser chamado uma vez a cada milissegundo.

Como o TcaplusDB realiza aumento automático global de campos?

Você precisa definir uma única tabela e definir o tipo de campo de uma única chave e um único valor para `int64_t` (vários campos de valor podem implementar matrizes de contador). Vários servidores de jogos podem aumentar uma única chave simultaneamente (sem definir as regras de proteção de versão) e obter o resultado de aumento retornado para esta operação, então o resultado do aumento será um aumento automático global.

Quais regras são definidas para consultas por chaves parciais (índices)?

A chave do índice deve fazer parte da chave primária, a chave do índice deve conter a `shardkey` e a chave do índice não pode ser a chave primária.

Como faço para obter uma consulta bidirecional de ID e nome de uma única tabela?

O terceiro campo de chave `x` é usado como `shardkey`, e o índice é criado em `ID` e `x`, nome e `x` para obter essa funcionalidade. Por exemplo, ao armazenar informações do jogador, o distrito do jogador pode ser adicionado, ou seja, a chave primária é `uin`, nome, distrito e a chave de índice é `uin` e distrito, e nome e distrito.

A `tcaplus_client` aceita a exibição de campos no segundo nível de aninhamento ou acima?

Sim. Você pode executar `help select` para ver como `select * into a.xml` é usado.

O que deve ser observado ao alterar a tabela no TcaplusDB?

1. Você só pode adicionar novos campos. Você não pode modificar o tipo e o nome de um campo existente ou excluir um campo existente. Para modificá-los, você precisa modificar a estrutura da tabela dinamicamente.
2. O comprimento de uma matriz ou string em um campo de chave não primária pode ser aumentado, mas não reduzido. Para modificá-lo, você precisa modificar a estrutura da tabela dinamicamente.
3. O número da versão deve ser incrementado em 1 para cada novo campo, e o número da versão definido na parte superior do arquivo XML também deve ser incrementado em 1 (ou seja, o número da versão do arquivo XML deve ser o mesmo do novo campo). A tabela no TcaplusDB precisa ser alterada antes que a tabela no servidor do jogo seja alterada. O TcaplusDB permite a modificação dinâmica da estrutura da tabela e, para usar essa funcionalidade, [envie um tíquete](#) e selecione **Other (Outro)** na página **Select the related product (Selecione o produto relacionado)**.

O que deve ser observado ao definir a tabela no TcaplusDB?

1. O campo `refer` precisa ser adicionado ao campo `count`.
2. A shardkey da tabela deve ser altamente discreta.
3. A chave do índice não pode ser igual à chave primária.

Princípios do banco de dados

Last updated: 2024-12-04 10:27:05

- Como o servidor do jogo elimina um nó tcaproxy (camada de acesso) inválido?
- Como o servidor do jogo escolhe o nó tcaproxy (camada de acesso)?
- O TcaplusDB tem a funcionalidade de compactação?
- A API do TcaplusDB é thread-safe?
- Como o tcapsvr (camada de armazenamento) realiza a recuperação de desastres?
- Como o tcaproxy (camada de acesso) realiza a recuperação de desastres?
- O TcaplusDB possui proteção contra sobrecarga?
- Qual é o princípio de troca de dados de acesso frequente e não frequente para o TcaplusDB?
- Como o tcaproxy (camada de acesso) escolhe o tcapsvr (camada de armazenamento)?
- Qual é o nível de bloqueio no TcaplusDB?

Como o servidor do jogo elimina um nó tcaproxy (camada de acesso) inválido?

A API do TcaplusDB permite a recuperação de desastres no caso de qualquer exceção de tcaproxy. Existem duas maneiras principais de a API eliminar processos tcaproxy inválidos:

1. A API considera fisicamente que um tcaproxy não está disponível. A API envia pacotes de detecção de pulsação a todos os tcaproxy conectados a cada segundo. Se um servidor de jogo não receber os pacotes de retorno de pulsação correspondentes do tcaproxy em 10 segundos, a API desconectará ativamente a conexão TCP com o tcaproxy e conectará ativamente o tcaproxy na próxima atualização.
2. A API considera logicamente que um tcaproxy não está disponível. A API calcula a taxa de solicitação e resposta de um tcaproxy a cada 10 segundos como base para a avaliação. O tempo limite máximo para um pacote de solicitação é de três segundos. Se um tcaproxy expirar mais de três vezes, o tcaproxy será considerado indisponível e nenhuma solicitação será enviada a ele. Uma solicitação `getmetadata` é enviada ao tcaproxy 60 segundos depois. Se o tcaproxy pode lidar corretamente com a solicitação `getmetadata`, a API considera o tcaproxy disponível e as solicitações serão enviadas para ele novamente.

Se o servidor do jogo descobrir que um tcaproxy não está disponível em 10 segundos, ele não enviará dados para o nó tcaproxy.

Como o servidor do jogo escolhe o nó tcaproxy (camada de acesso)?

O servidor do jogo mantém um anel Hash consistente localmente. Depois que um nó tcaproxy (camada de acesso) for verificado, ele será adicionado ao anel Hash. Se um nó tcaproxy (camada de acesso) reduzir a capacidade ou o link TCP entre o servidor do jogo e o tcaproxy (camada de acesso) for desconectado devido a anormalidades na máquina, o servidor do jogo removerá o nó tcaproxy (camada de acesso) do anel Hash. O servidor do jogo calcula os valores de hash com base na chave primária na solicitação (se for uma

solicitação `batchget`, ele seleciona aleatoriamente um único nó tcaproxy (camada de acesso) e, em seguida, seleciona um único nó tcaproxy (camada de acesso) do anel de Hash consistente.

O TcaplusDB tem a funcionalidade de compactação?

O TcaplusDB tem uma funcionalidade de compactação baseada no algoritmo de compactação ágil do Google, incluindo compactação de protocolo (ou seja, compactação do pacote de solicitação/resposta entre o servidor do jogo e o tcaproxy (camada de acesso) e compactação de dados (ou seja, compactação dos dados que precisam ser armazenados por tcapsvr (camada de armazenamento)). Se você deseja reduzir os custos de tráfego de rede entre o servidor do jogo e o tcaproxy (camada de acesso), recomendamos que você habilite a compactação de protocolo. Você também pode chamar a função `SetCompressSwitch` da API do TcaplusDB para habilitar a compactação do tcapsvr (camada de armazenamento), que pode economizar espaço em disco, melhorar o desempenho do disco de E/S e reduzir os recursos da CPU usados para compactação e descompactação.

A API do TcaplusDB é thread-safe?

A API do TcaplusDB não é thread-safe, principalmente porque componentes como tlog e tdr não são thread-safe. Recomenda-se que uma única thread use um único objeto de API e uma única região do jogo use um único objeto de API. Se você precisar interagir entre regiões do jogo, é recomendável manter vários objetos de API com um único servidor do jogo.

Como o tcaproxy (camada de acesso) realiza a recuperação de desastres?

O tcaproxy (camada de acesso) adota um esquema de design ponto a ponto, ou seja, todos os nós tcaproxy (camada de acesso) em uma única região do jogo contêm informações de roteamento de todas as tabelas em uma única região do jogo. Se um tcaproxy (camada de acesso) falhar, desde que os nós tcaproxy (camada de acesso) restantes não estejam sobrecarregados, o servidor do jogo vai eliminar o nó tcaproxy (camada de acesso) anormal, que não afetará o uso do servidor do jogo. Não existe um ponto único de risco de falha para o tcaproxy (camada de acesso).

Como o tcapsvr (camada de armazenamento) realiza a recuperação de desastres?

O tcapsvr (camada de armazenamento) é executado em um modo primário-secundário (tcapsvr primário e tcapsvr secundário). Os tcapsvr primários e secundários sincronizam dados em tempo real e são implantados em diferentes IDCs na mesma cidade, garantindo que a latência de sincronização primário-secundário seja inferior a 10 ms. Se o tcapsvr secundário estiver anormal, não afetará o uso do servidor do jogo (se o descarregamento da solicitação de leitura não estiver habilitado, as solicitações do servidor do jogo serão processadas pelo tcapsvr primário. Se o descarregamento da solicitação de leitura estiver habilitado, o tcapsvr secundário ajudará no processamento de parte das solicitações de leitura e o DBA recriará o tcapsvr secundário; se o tcapsvr primário for anormal, o tcapsvr secundário executará a recuperação de falhas e o DBA solicitará uma nova máquina para recriar o tcapsvr secundário. Não existe um ponto único de risco de falha para o tcapsvr (camada de armazenamento).

O TcaplusDB possui proteção contra sobrecarga?

A camada de acesso e a camada de armazenamento têm medidas de proteção contra sobrecarga no nível dos processos para garantir que os serviços não entrem em colapso durante os horários de pico.

Qual é o princípio de troca de dados de acesso frequente e não frequente para o TcaplusDB?

O TcaplusDB usa memória + armazenamento em disco SSD. Os primeiros GB de dados de um único arquivo de mecanismo são mapeados na memória. Os dados de acesso frequente são colocados na memória e os dados de acesso não frequente são colocados no disco. O algoritmo LRU é usado para troca de dados de acesso frequente e não frequente. A operação `get` do servidor do jogo dispara a operação de swap-in LRU, e a thread LRU de `tcapsvr` (camada de armazenamento) é responsável pela operação de swap-out LRU. Tente garantir que os dados de acesso frequente sejam armazenados na memória, garantindo assim uma alta taxa de acertos do cache e baixa latência de leitura e gravação única.

Como o `tcaproxy` (camada de acesso) escolhe o `tcapsvr` (camada de armazenamento)?

Cada tabela define uma `shardkey`. Se nenhuma `shardkey` for definida, a `shardkey` padroniza para a chave primária. O `tcaproxy` (camada de acesso) seleciona o `tcapsvr` (camada de armazenamento) correspondente de acordo com `hash (shardkey) % 10000` (em que `%` é o operador restante), então a `shardkey` deve ser altamente discreta.

Qual é o nível de bloqueio no TcaplusDB?

A granularidade do bloqueio no TcaplusDB é o nível de registro em log.