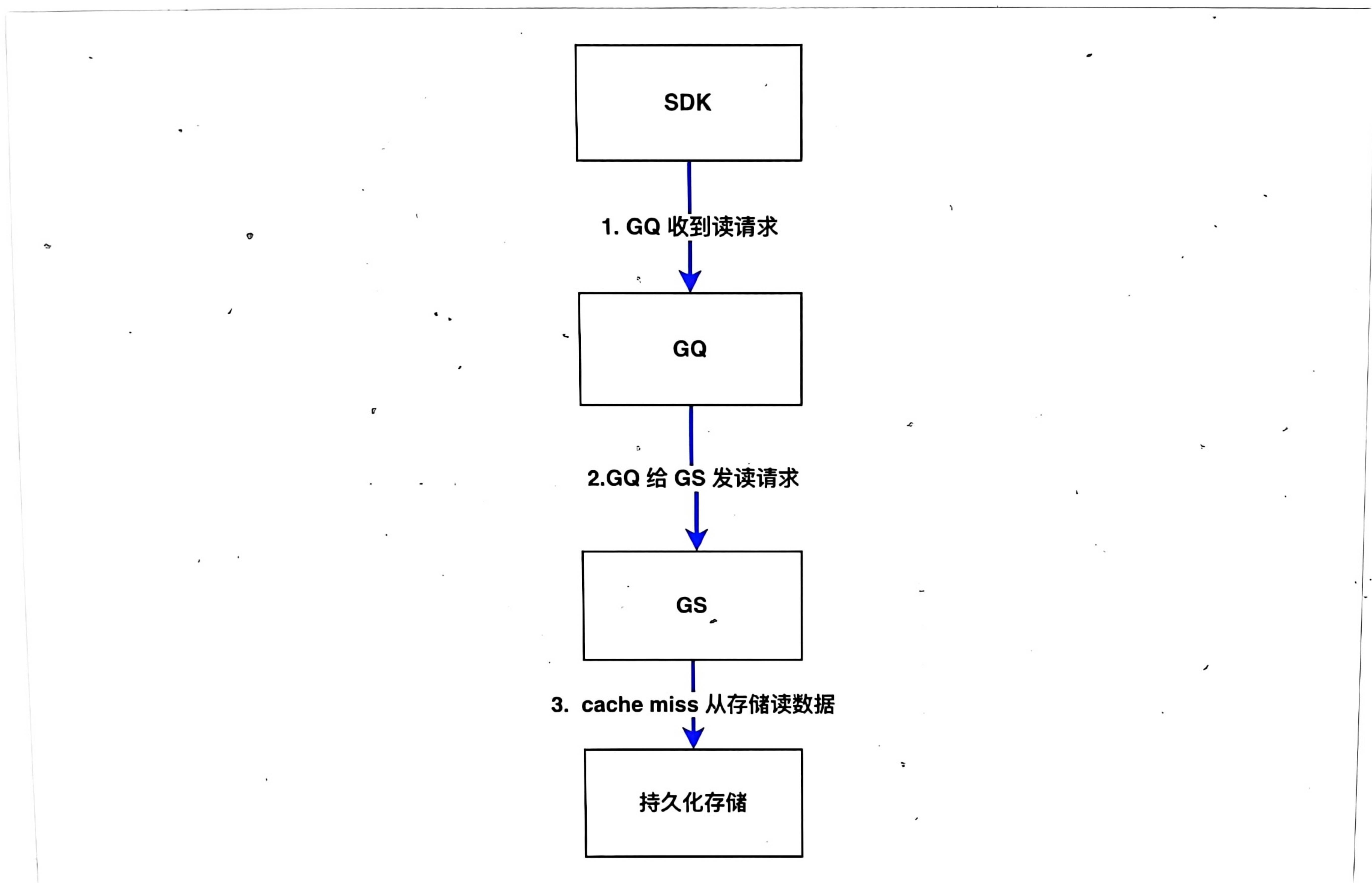


## 读请求

无论收到用户请求的 GQ 处于哪个机房，处理的流程都是一样的：

1. GQ 解析收到的请求后从同机房的 GS 读取数据
2. GS 会维护底层存储数据的缓存，如果要读取的数据命中缓存则返回 GQ，读取流程结束
3. 如果发生 cache miss，GS 从同机房的底层存储中读取数据，读取流程结束



## 三、读写一致性

ByteGraph 在多机房场景下提供最终一致性的保证。在用户收到 ByteGraph 返回的写入成功的响应后：

- 在主机房立即能读到最新的数据。
- 正常情况下(99%)：在一个跨机房的 **rtt(ms级)** 内所有机房都能读到最新的数据。
- 异常情况下(1%)：如 GQ 间数据同步通信失败、GQ GS 间缓存更新失败，需要分钟级所有的机房才能读到最新的数据。

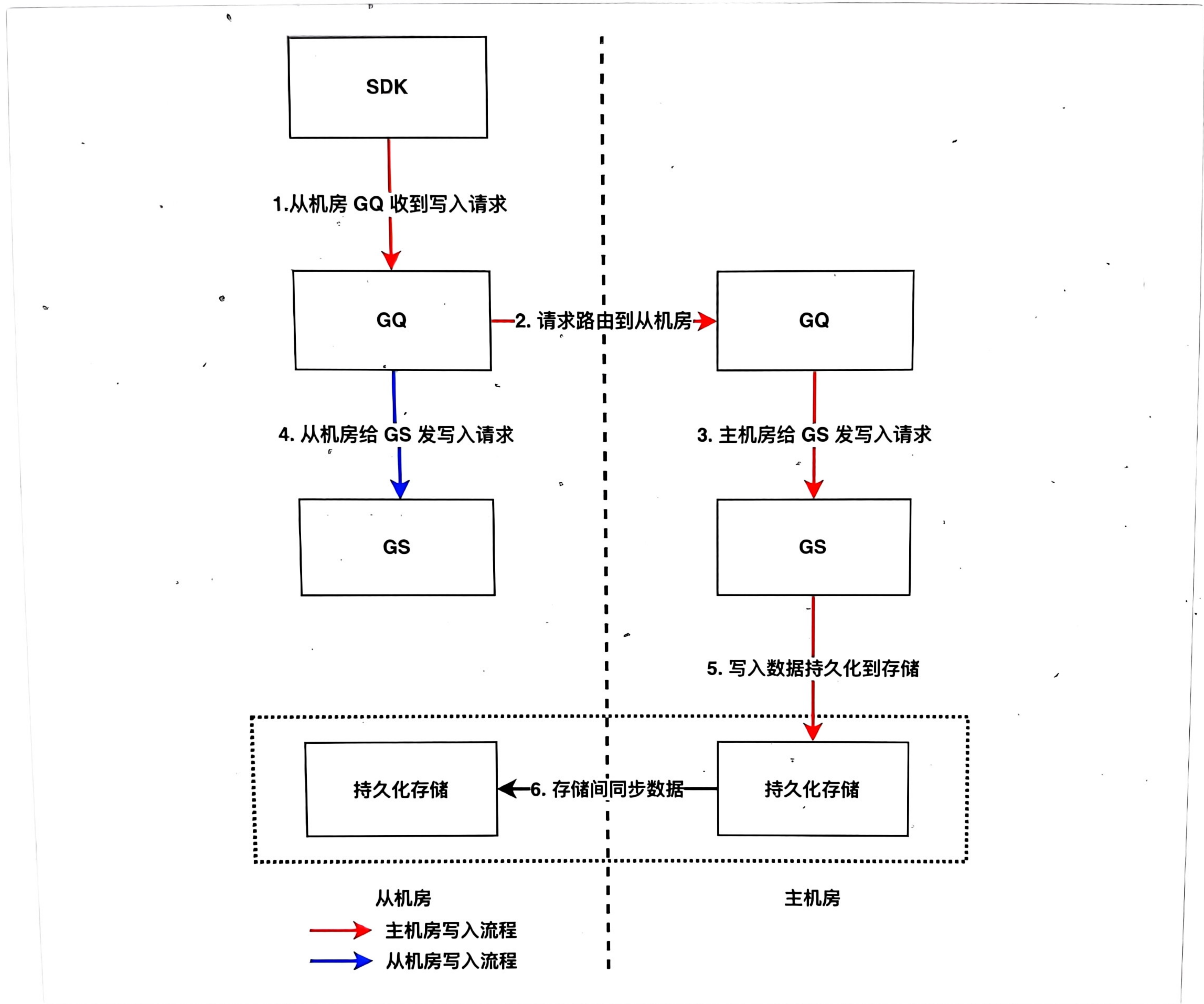
## 异地多活/跨大洲同步功能



6. 底层存储的多机房之前会同步数据

其中步骤2、3 是并发执行的

当从机房收到用户的写入请求

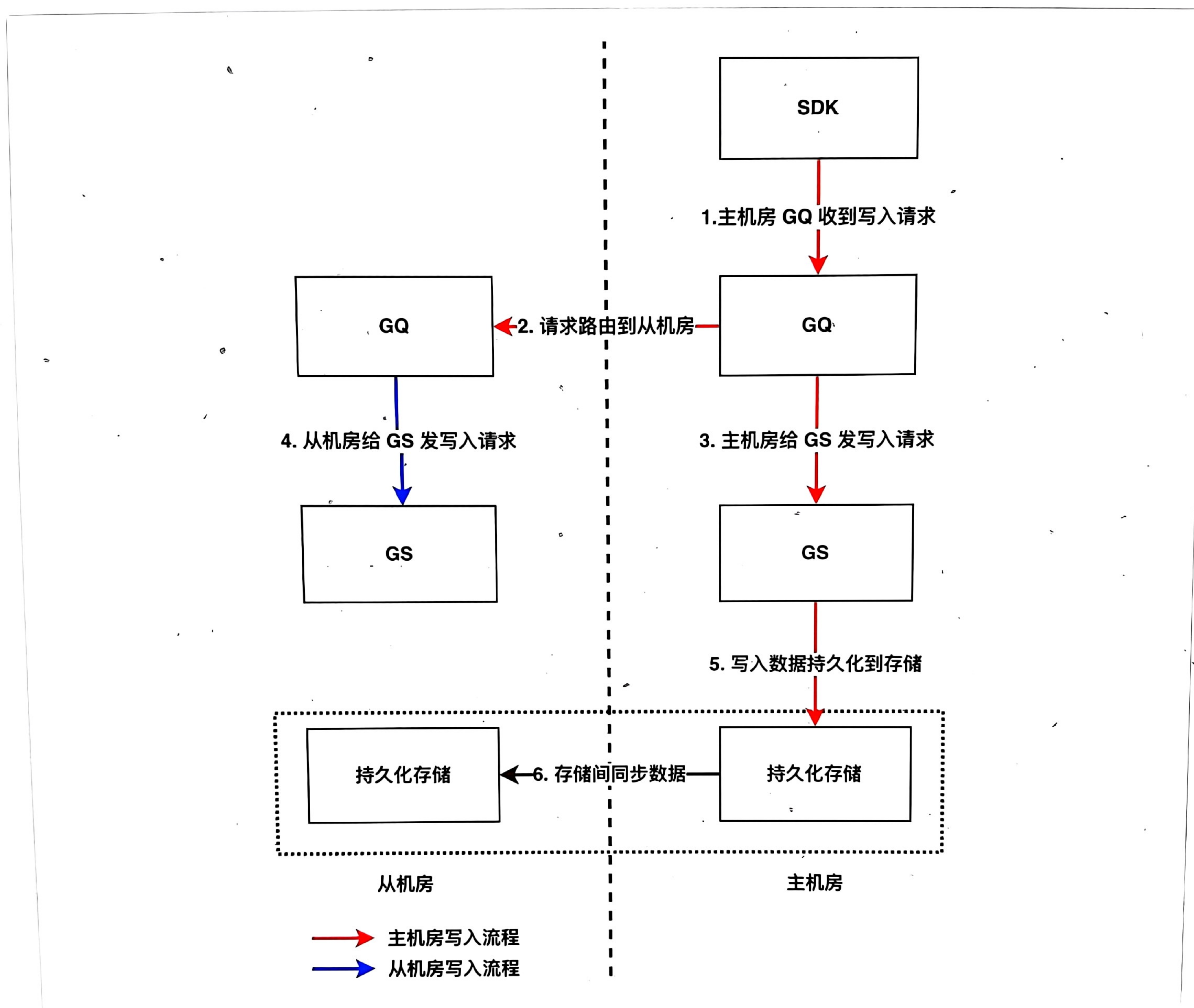


1. 用户将请求发送给从机房的 GQ
2. 从机房的 GQ 将请求转发给主机房的 GQ
3. 主机房的 GQ 将请求解析后发送给主机房的 GS
4. 从机房的 GQ 将请求解析后发送给从机房的 GS，GS 将数据写入到内存，从机房的写入流程结束
5. 主机房的 GS 将数据写入到底层存储，主机房的写入流程结束，这时收到用户请求的主机房的 GQ 在收到从机房的 GQ 写入成功的响应后，会返回 SDK 写入成功
6. 底层存储的多机房之间会同步数据



## 二、以两机房为例说明读写的流程

当主机房收到用户的写入请求时



1. 用户将请求发送给主机房的 GQ
2. 主机房的 GQ 将请求转发给从机房的 GQ
3. 主机房的 GQ 将请求解析后发送给主机房的 GS
4. 从机房的 GQ 将请求解析后发送给从机房的 GS，GS 将数据写入到内存，从机房的写入流程结束
5. 主机房的 GS 将数据写入到底层存储，主机房的写入流程结束，这时收到请求的主机房的 GQ 会返回 SDK 写入成功
6. 底层存储的多机房之前会同步数据

其中步骤2、3 是并发执行的



## 名词解释

GQ (Graph Query Engine): ByteGraph 的查询引擎层, 负责解析请求、多机房间转发请求等

GS (Graph Store Engine): ByteGraph 的存储引擎层, 负责缓存、持久化数据、维护索引等

持久化存储: ByteGraph 依赖的存储组件, 一般是 Abase 或 ByteKV

## 主从多机房同步功能

### 一、功能描述

ByteGraph 支持城域网多机房部署, 正常情况下多个机房都能对外提供读写服务, 业务可以直接在所在的机房访问 ByteGraph。当某个机房、或机房间网络等发生故障时, ByteGraph 能够将主机房切换到其他正常的机房对外提供服务。ByteGraph 的主、从机房对业务是透明的。

ByteGraph 典型的部署方式是三机房部署, 每个机房间有链路进行异步的数据同步, 保证最终一致性。

