

v3.0 >

> Introduction

> Concepts

> How-to

- > Get Started
- > Deploy
  - Hardware Recommendations
  - > From Binary Tarball
  - > Orchestrated Deployment
- > Geographic Redundancy
  - Overview
  - Schedule Replicas by Topology Labels
  - Data Migration with Ansible

> Configure

> Secure

> Monitor

> Migrate


> Maintain

> Scale

> Upgrade

> Troubleshoot

> Reference

 Important

You are viewing the archived documentation of TiDB, which no longer receives updates. It is recommended that you use [the latest LTS version](#) of the TiDB database.

# Cross-DC Deployment Solutions

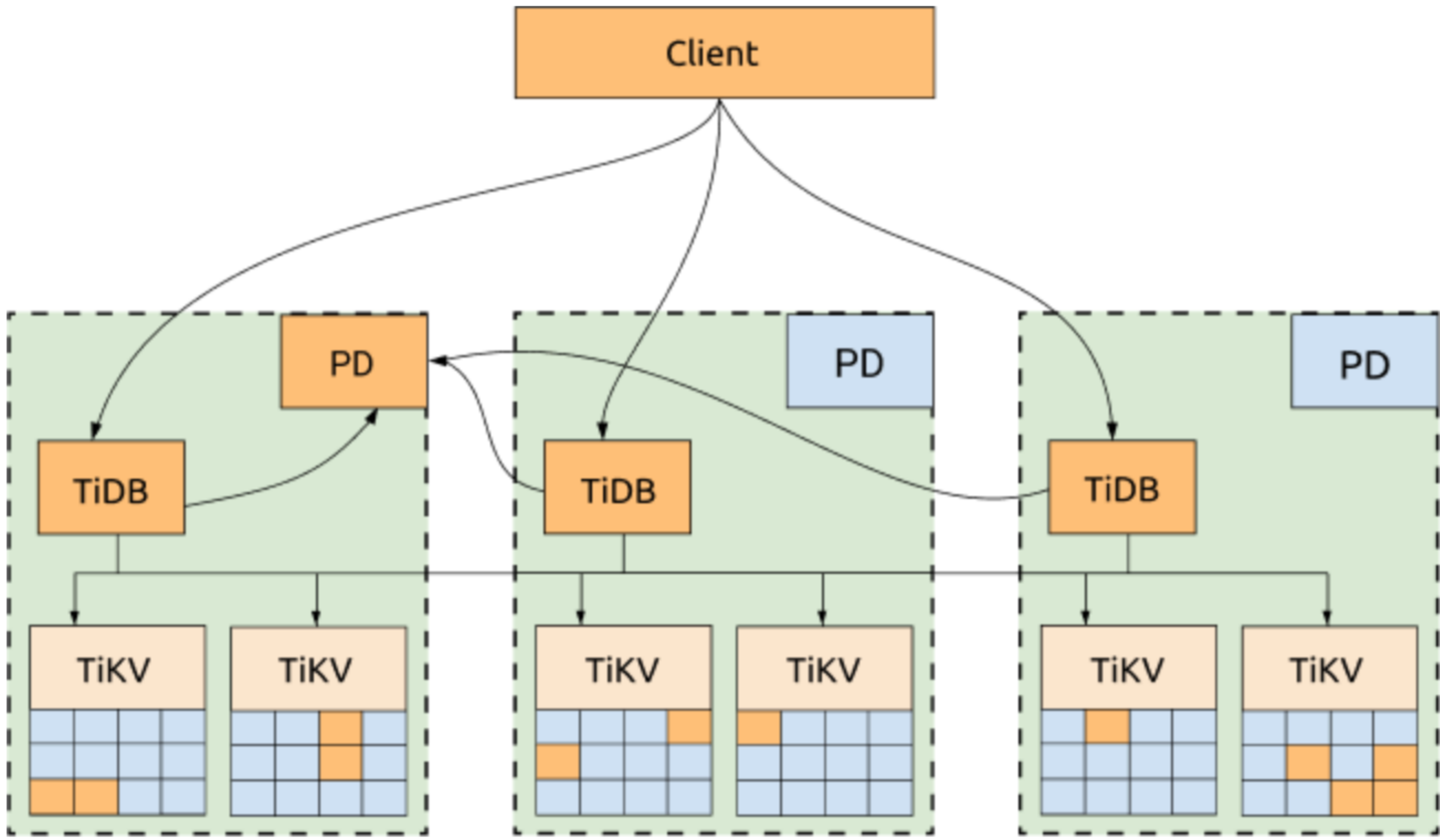


4 Contributors

As a NewSQL database, TiDB excels in the best features of the traditional relational database and the scalability of the NoSQL database and is of course, highly available across data centers (hereinafter referred to as DC). This document is to introduce different deployment solutions in cross-DC environment.

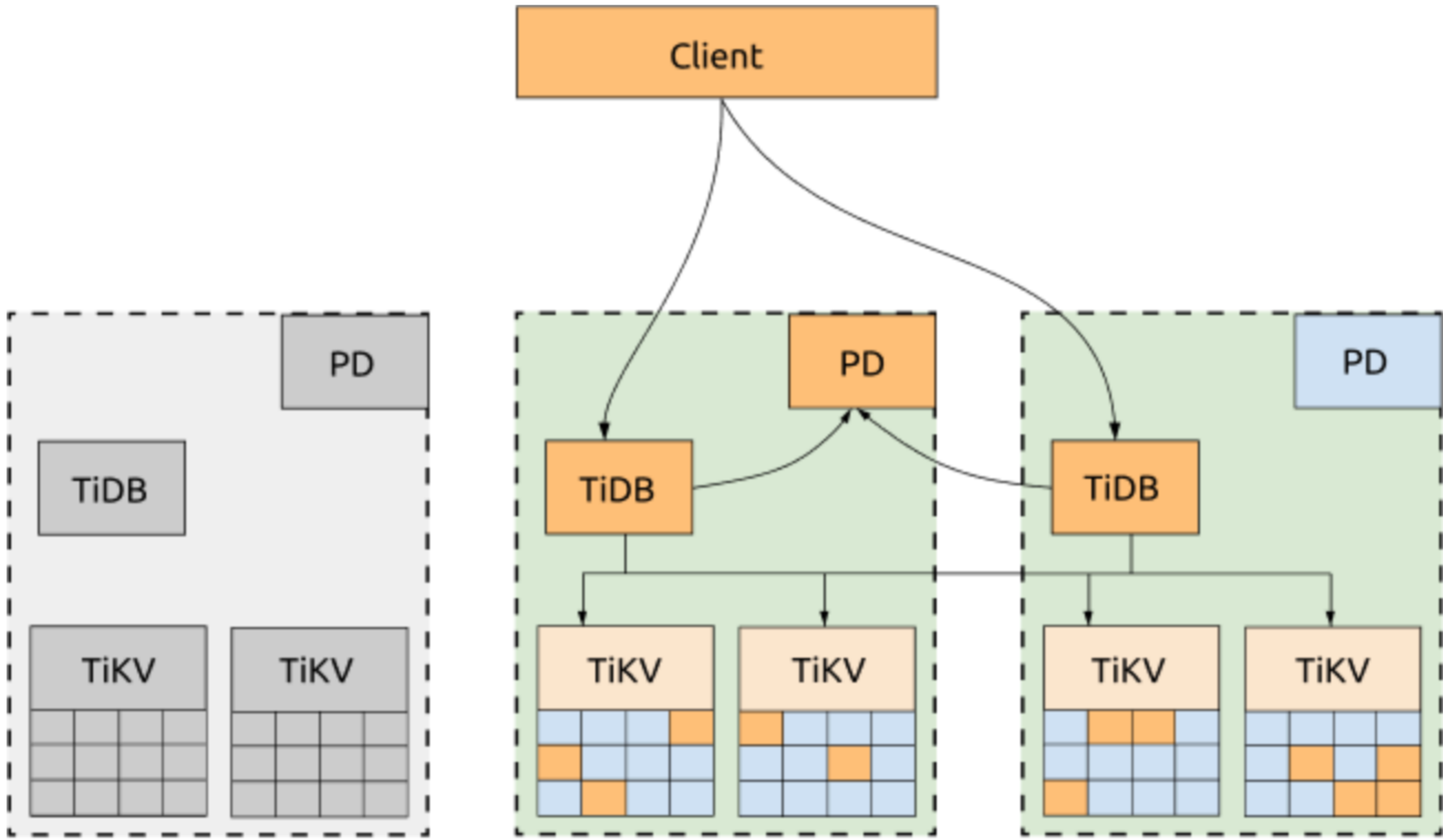
## 3-DC deployment solution

TiDB, TiKV and PD are distributed among 3 DCs, which is the most common deployment solution with the highest availability.



### Advantages

All the replicas are distributed among 3 DCs. Even if one DC is down, the other 2 DCs will initiate leader election and resume service within a reasonable amount of time (within 20s in most cases) and no data is lost. See the following diagram for more information:



### Disadvantages

The performance is greatly limited by the network latency.

- For writes, all the data has to be replicated to at least 2 DCs. Because TiDB uses 2-phase commit for writes, the write latency is at least twice the latency of the network between two DCs.
- The read performance will also suffer if the leader is not in the same DC as the TiDB node with the read request.
- Each TiDB transaction needs to obtain TimeStamp Oracle (TSO) from the PD leader. So if TiDB and PD leader are not in the same DC, the performance of the transactions will also be impacted by the network latency because each transaction with write request will have to get TSO twice.

### Optimizations

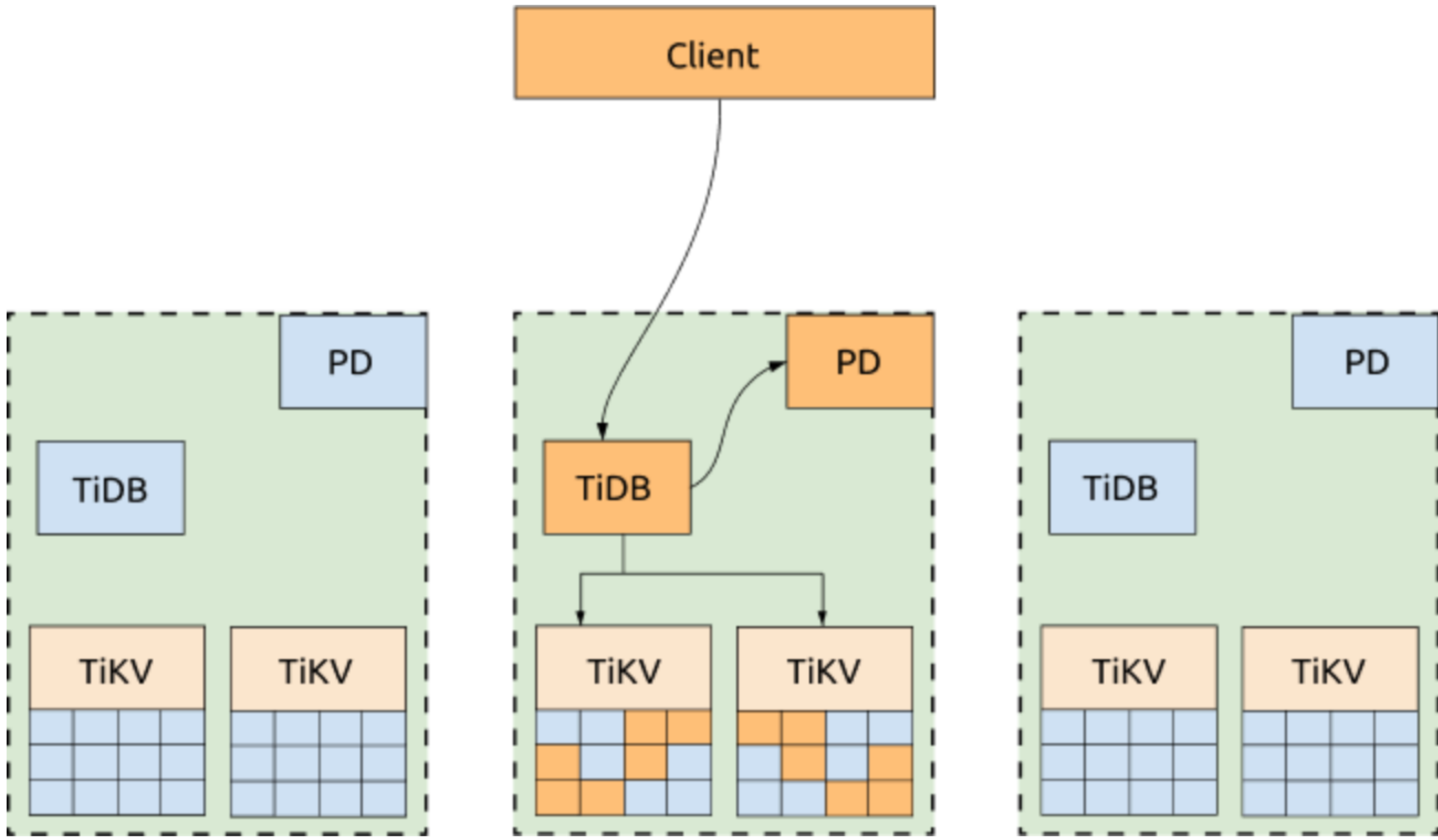
If not all of the three DCs need to provide service to the applications, you can dispatch all the requests to one DC and configure the scheduling policy to migrate all the TiKV Region leader and PD leader to the same DC, as what we have done in the following test. In this way, neither obtaining TSO or reading TiKV Regions will be impacted by the network latency between DCs. If this DC is down, the PD leader and Region leader will be automatically elected in other surviving DCs, and you just need to switch the requests to the DC that are still online.

 Download PDF

#### What's on this page

- 3-DC deployment solution
- Advantages
- Disadvantages
- Optimizations
- 3-DC in 2 cities deployment solution
- 2-DC + Binlog replication deployment solution
- Overall analysis for HA and DR





### 3-DC in 2 cities deployment solution

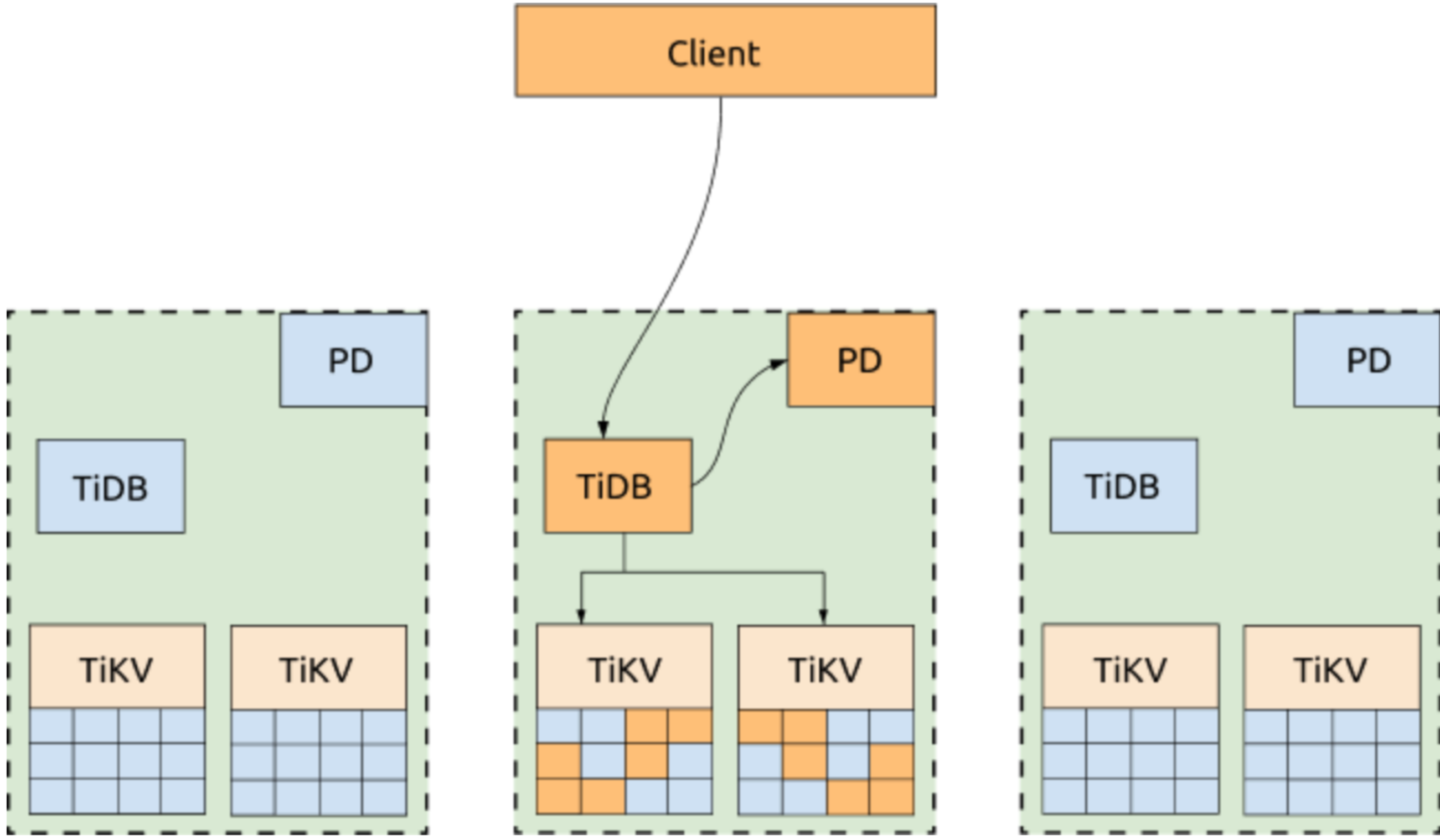
This solution is similar to the previous 3-DC deployment solution and can be considered as an optimization based on the business scenario. The difference is that the distance between the 2 DCs within the same city is short and thus the latency is very low. In this case, we can dispatch the requests to the two DCs within the same city and configure the TiKV leader and PD leader to be in the 2 DCs in the same city.

client

in the same DC, the performance of the transactions will also be impacted by the network latency because each transaction with write request will have to get TSO twice.

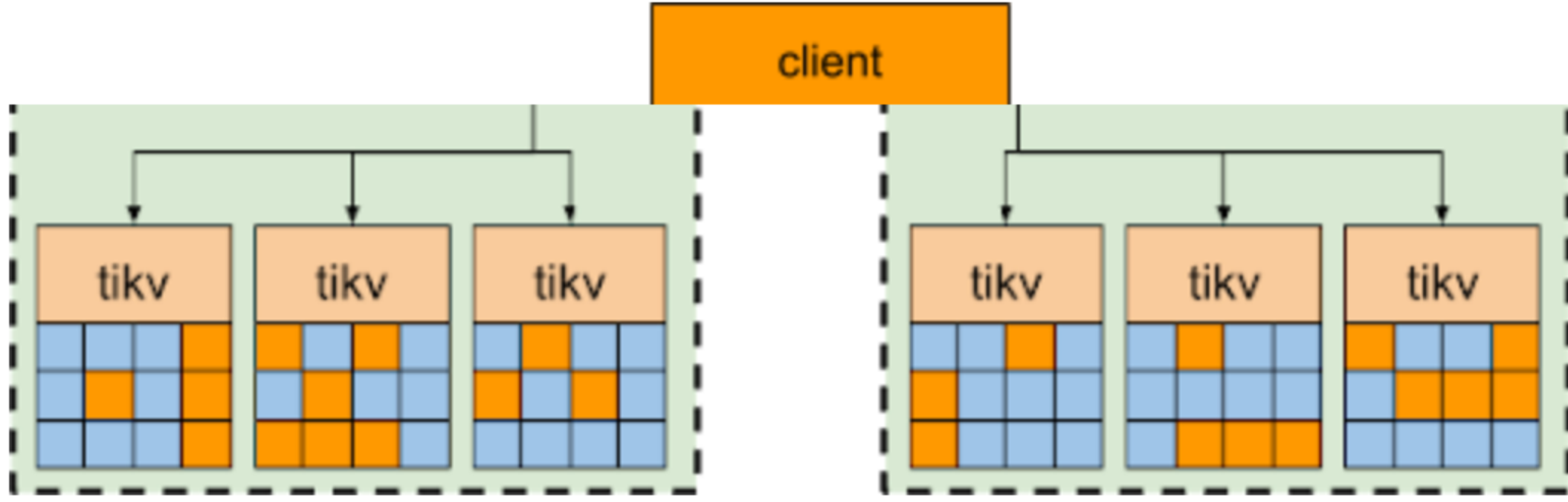
#### Optimizations

If not all of the three DCs need to provide service to the applications, you can dispatch all the requests to one DC and configure the scheduling policy to migrate all the TiKV Region leader and PD leader to the same DC, as what we have done in the following test. In this way, neither obtaining TSO or reading TiKV Regions will be impacted by the network latency between DCs. If this DC is down, the PD leader and Region leader will be automatically elected in other surviving DCs, and you just need to switch the requests to the DC that are still online.

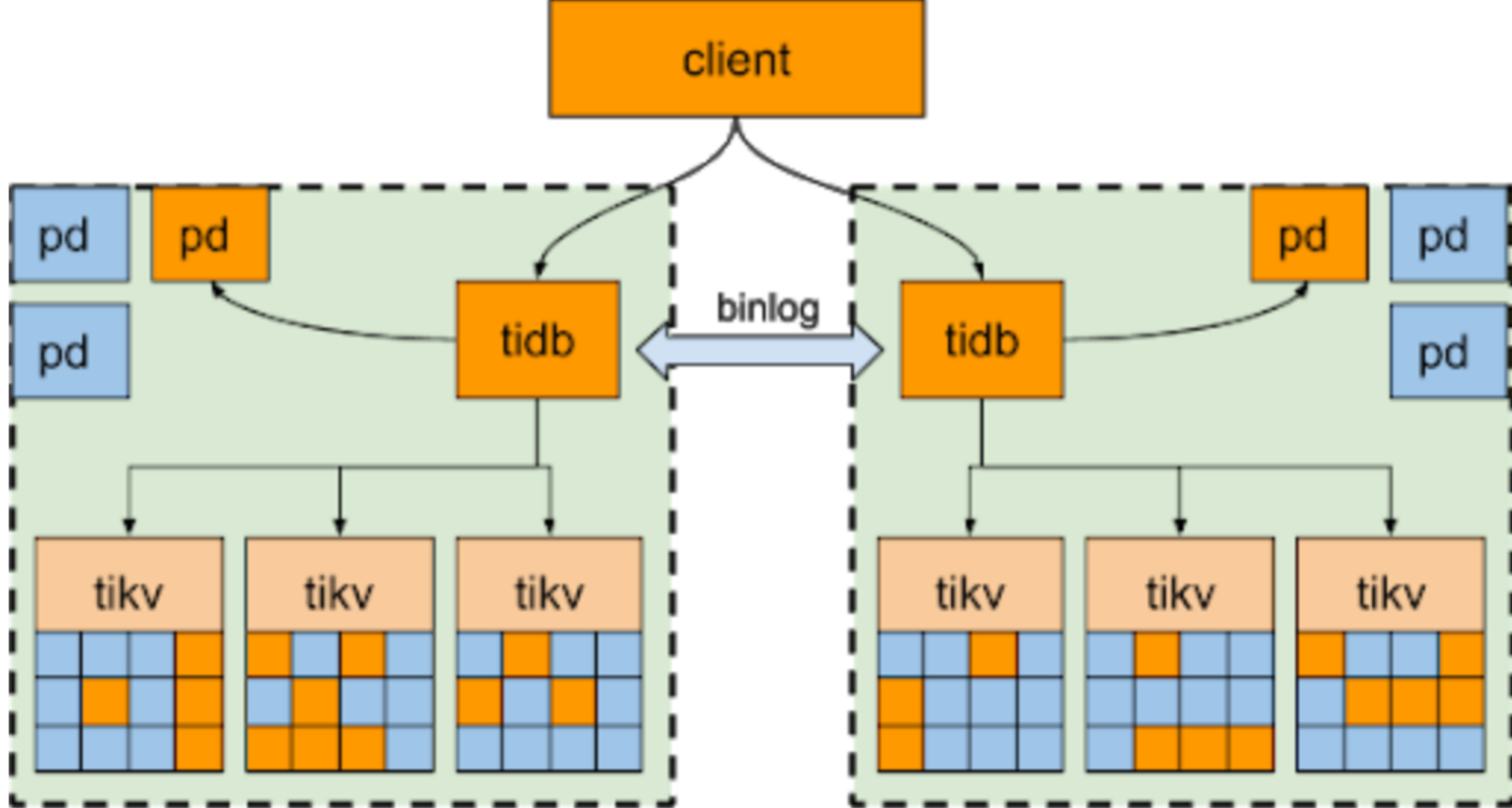


### 3-DC in 2 cities deployment solution

This solution is similar to the previous 3-DC deployment solution and can be considered as an optimization based on the business scenario. The difference is that the distance between the 2 DCs within the same city is short and thus the latency is very low. In this case, we can dispatch the requests to the two DCs within the same city and configure the TiKV leader and PD leader to be in the 2 DCs in the same city.



If the primary DC goes down, the requests can be switched to the secondary cluster. Similar to MySQL, some data might be lost. But different from MySQL, this solution can ensure the high availability within the same DC: if some nodes within the DC are down, the online workloads won't be impacted and no manual efforts are needed because the cluster will automatically re-elect leaders to provide services.



Some of our production users also adopt the 2-DC multi-active solution, which means:

1. The application requests are separated and dispatched into 2 DCs.



2. Each DC has 1 cluster and each cluster has two databases: A primary database to serve part of the application requests and a secondary database to act as the backup of the other DC's primary database. Data written into the primary database is replicated via Binlog to the secondary database in the other DC, forming a loop of backup.

Please be noted that for the 2-DC + Binlog replication solution, data is asynchronously replicated via Binlog. If the network latency between 2 DCs is too high, the data in the secondary cluster will fall much behind of the primary cluster. If the primary cluster goes down, some data will be lost and it cannot be guaranteed the lost data is within 5 minutes.

### Overall analysis for HA and DR

For the 3-DC deployment solution and 3-DC in 2 cities solution, we can guarantee that the cluster will automatically recover, no human interference is needed and that the data is strongly consistent even if any one of the 3 DCs goes down. All the scheduling policies are to tune the performance, but availability is the top 1 priority instead of performance in case of an outage.

For 2-DC + Binlog replication solution, we can guarantee that the cluster will automatically recover, no human interference is needed and that the data is strongly consistent even if any some of the nodes within the primary cluster go down. When the entire primary cluster goes down, manual efforts will be needed to switch to the secondary and some data will be lost. The amount of the lost data depends on the network latency and is decided by the network condition.

Products

TiDB Cloud Serverless

TiDB Cloud Dedicated

TiDB Self-Managed

Pricing

Ecosystem

Integrations

TiKV

TiFlash

OSS Insight

Resources

Blog

Forum

Events & Webinars

HTAP Summit

Docs

Developer Guide

FAQs

Support

Company

About Us

News

Careers

Contact Us


Partners


Trust Center


Security


Release Support


Stay Connected

















© 2025 PingCAP. All Rights Reserved.

/ Privacy Policy

/ Legal