



集群部署及参数实践

提纲

- ❑ TiDB 集群硬件规划
- ❑ TiDB 集群部署方式
- ❑ TiDB 集群日常操作
- ❑ TiDB 集群常见参数
- ❑ 题目

Part 1 - TiDB 集群硬件规划

单实例硬件推荐

测试环境							
序号	服务器数量	CPU	内存	磁盘	网络	部署组件	备注
1	2	16vc	64 GB	500 GB SAS	万兆网	tidb * 1	
2	3	4vc	8 GB	500 GB SSD	万兆网	pd * 1	
3	3	16vc	64 GB	1.5 T PCIE SSD	万兆网	tikv * 1	
4	1	16vc	64 GB	500 GB SAS	万兆网	prometheus * 1 grafana * 1 alertmanager * 1	

TiDB 线上生产环境硬件推荐

生产环境							
序号	服务器数量	CPU	内存	磁盘	网络	部署组件	备注
1	3	2 * 12 * 2vc	256 GB	300G SAS *2 (os) 600G SAS * 10 (log、binlog、backup) 500G SSD * 2(pd)	万兆网	tidb * 2 pd * 1	
2	4	2 * 12 * 2vc	256 GB	300G SAS *2 (os) 4T PCIE SSD * 2(tikv)	万兆网	tikv * 4	
3	1	2 * 12 * 2vc	256 GB	300G SAS *2 (os) 600G SAS * 10 (monitor、log)	万兆网	tidb-ansible prometheus grafana alertmanager	可看情况缩减些配置

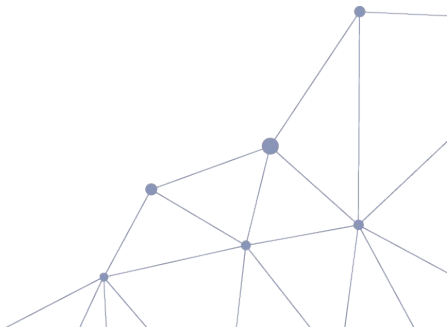
注意事项

- ❑ TiKV 对磁盘要求较高, 建议使用 PCIE SSD;
- ❑ TiDB 集群内部要求万兆网;
- ❑ 生产集群建议 TiKV 至少 4 实例以上;
- ❑ 两地三中心建议同城机房不高于 2ms, 异地机房不高于 16ms;
- ❑ 两地三中心服务器数量需与机房呈对应倍数关系;
- ❑ TiDB 可在云上环境部署, 云上环境需使用本地 SSD;
- ❑ 单服务器硬件能力较强, 建议使用多实例而非虚拟机;
- ❑ 生产环境不建议 TiDB 和 TiKV 混合部署;
- ❑ TiDB 节点建议关闭 swap 。

Part 2 - TiDB 集群部署方式

TiDB 常见部署方式

- ❑ 测试版(功能体验 + 开发)
 - ❑ 单机部署模式
 - ❑ 适合个人观摩各组件之间的联系
 - ❑ 不方便管理, 不推荐使用
 - ❑ docker-compose 快速部署集群
 - ❑ 简单快捷, 一键启停, 组件完全, 可测试功能
 - ❑ 基本不用考虑性能
- ❑ 正式版
 - ❑ 基于 ansible 构建 TiDB 集群
 - ❑ 当前主流多节点管理工具, tidb 主流部署工具
 - ❑ 受制于 ansible, 不够智能, 节点较多时, 管理复杂
 - ❑ 基于 k8s 构建一套 TiDB 集群
 - ❑ cloud is future, 未来 TiDB 商业版部署工具



The Next. . .

- ❑ TiUP
- ❑ TiOPS

Part 3 - TiDB 集群日常操作

Ansible 简介

基于Python开发, 集合了众多运维工具(puppet、cfengine、chef、func、fabric)的优点, 实现了批量系统配置、批量程序部署、批量运行命令等功能

- ❑ 部署简单, 无需 agent, 只要求有 ssh 和 python
- ❑ 模块化: 本身只提供一个框架, 靠丰富的模块来完成各种工作, 插件可扩展
- ❑ 集中的 inventory.ini 完成集群拓扑定义和编排
- ❑ 通过称作 playbook 的 yaml 脚本, 完成各种复杂任务
- ❑ 幂等性, 可重复执行, 使集群达到目标状态

Ansible 操作示例

- 一个简单的 ansible 部署命令

```
- name: deploying TiDB cluster
  hosts: tidb_servers
  tags:
    - tidb
  roles:
    - tidb
    - { role: tispark, when: "(groups.get('spark_master', []) | length == 0 or groups.get('spark_slaves', []) | length == 0) and (deployment_method == 'binary')" }
```

- 一个简单的 ansible role 目录结构

```
drwxr-xr-x  3 dinafer staff   96B  3  18  16:42 defaults
drwxr-xr-x 12 dinafer staff  384B  3  18  16:42 files
drwxr-xr-x  3 dinafer staff   96B  3  18  16:42 meta
drwxr-xr-x  7 dinafer staff  224B  3  18  16:42 tasks
drwxr-xr-x  5 dinafer staff  160B  3  18  16:42 templates
```

Inventory.ini 示例

```
# TiDB Cluster Part
[tidb_servers]
192.168.0.2

[tikv_servers]
192.168.0.3
192.168.0.4
192.168.0.5

[pd_servers]
192.168.0.6
192.168.0.7
192.168.0.8
```

```
# Monitoring Part
[monitoring_servers]
192.168.0.10

[grafana_servers]
192.168.0.10

[monitored_servers:children]
tidb_servers
tikv_servers
pd_servers

## Binlog Part
[pump_servers:children]
tidb_servers
```

```
# Monitoring Part
[monitoring_servers]
192.168.0.10

[grafana_servers]
192.168.0.10

[monitored_servers:children]
tidb_servers
tikv_servers
pd_servers

## Binlog Part
[pump_servers:children]
tidb_servers
```

<https://pingcap.com/docs-cn/stable/how-to/deploy/orchestrated/ansible/#概述>

常见使用脚本

功能	执行命令	说明
集群初始化	ansible-playbook bootstrap.yml	初始化系统环境，检查 inventory.ini 中所有机器的操作系统、磁盘性能，优化系统内核参数等
准备本地安装包	ansible-playbook local_prepare.yml	下载最新 binary 包到本地 downloads 目录（需要外网） 也可以手动下载解压放置在 resources/bin 目录下
集群部署	ansible-playbook deploy.yml	分发 binary 和配置文件到相应节点部署
启动集群	ansible-playbook start.yml	启动 TiDB，启动所有节点的服务，包含监控
停止集群	ansible-playbook stop.yml	停止 TiDB，停止所有节点的服务
滚动升级	ansible-playbook rolling_update.yml	依次停每个节点上的每个服务，更新 binary 和配置文件后重启 期间数据库可以正常提供服务（会有连接闪断，需要客户端有重连机制）

扩容的必要性

- TiDB 节点
 - 一般默认部署两个节点用于高可用
 - 用于可支持更高的连接数和吞吐
 - 用于较为基础的资源隔离
- PD 节点
 - 一般是奇数个节点
 - 常规情况下基本无扩容必要
- TiKV 节点
 - 最少 3 节点, 一般 4 节点, 多机房需要跟机房数成比例关系
 - 当存储不够用时, 用于扩充存储
 - 当 CPU 不够时支持更高的写入和查询

Part 4 - TiDB 集群常见配置

常见压测调节参数

☐ TiDB

- ☐ 日志级别
- ☐ 执行计划缓存
- ☐ 使用的 CPU 数量
- ☐ 冲突检测

☐ TiKV

- ☐ 日志级别
- ☐ sync-log
- ☐ 块缓存

常见压测调节参数

- ❑ TiDB 扫描相关
 - ❑ `tidb_distsql_scan_concurrency`
 - ❑ `tidb_index_lookup_size`
 - ❑ `tidb_index_lookup_concurrency`
 - ❑ `tidb_index_lookup_join_concurrency`
 - ❑ `tidb_hash_join_concurrency`
 - ❑ `tidb_index_serial_scan_concurrency`
 - ❑ `tidb_projection_concurrency`。
 - ❑ `tidb_hashagg_final_concurrency`。
 - ❑ `tidb_index_join_batch_size`

其他注意参数

- ❑ batch 类参数
 - ❑ batch insert
 - ❑ batch delete
- ❑ 增加索引速度 类参数
 - ❑ tidb_ddl_reorg_worker_cnt
 - ❑ tidb_ddl_reorg_batch_size
- ❑ PD 调度参数
 - ❑ "leader-schedule-limit"
 - ❑ "region-schedule-limit"
 - ❑ "replica-schedule-limit"
 - ❑ "merge-schedule-limit"

<https://pingcap.com/docs-cn/stable/reference/tools/pd-control/#pd-control-使用说明>

附录



问答1:

- ❑ 推荐一套支持高可用的 TiDB 最小性能测试环境配置, 并说明原因。

选择2:

- ☐ 以下哪个端口不是 TiDB 集群组件常用端口
 - ☐ A、4000
 - ☐ B、2379
 - ☐ C、3508
 - ☐ D、20160

选择3:

- ❑ 以下选择中, 修改 TiDB 配置文件不生效的操作是:
 - ❑ A、修改 tidb-ansible/conf 下的 tidb 配置文件, 然后执行 rolling-update.yml 命令
 - ❑ B、修改 tidb 节点的配置文件并重启节点上 tidb-server 的进程
 - ❑ C、修改 tidb 节点的配置文件, 然后执行 rolling-update.yml 命令
 - ❑ D、修改 tidb-ansible/roles/tidb/templates/tidb.toml.j2 文件, 然后执行 rolling-update.yml 命令

选择4:

- ☐ 以下修改集群副本数为 5 的操作中, 不生效的有:
 - ☐ A、通过 pd-ctl 设置副本数为 5
 - ☐ B、通过 pd http 接口设置副本数为 5
 - ☐ C、新建集群通过设置 pd 配置文件设置副本数为 5
 - ☐ D、已有集群通过设置 pd 配置文件设置副本数为 5

选择5:

- ☐ 正常部署的 tidb 高可用集群中, 以下哪个 tidb 组件出现扩容的概率最低:
 - ☐ A、tidb
 - ☐ B、pd
 - ☐ C、tikv
 - ☐ D、pump