|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |
| 调优指导书模板 |
|  | | |

目录

[1 调优概述 1](#_Toc80903847)

[1.1 方案介绍 1](#_Toc80903848)

[1.2 调优思路 1](#_Toc80903849)

[1.2.1 业务流程 1](#_Toc80903850)

[1.2.2 调优思路 2](#_Toc80903851)

[2 调优方案 3](#_Toc80903852)

[2.1 BIOS配置(示例--硬件调优部分) 3](#_Toc80903853)

[2.2 操作系统调优(示例--OS调优部分) 4](#_Toc80903854)

[2.2.1 文件系统调优 4](#_Toc80903855)

[2.2.2 网卡中断绑核 4](#_Toc80903856)

[2.2.3 网络参数调优 5](#_Toc80903857)

[2.2.4 IO参数调优 5](#_Toc80903858)

[2.2.5 缓存参数调优 5](#_Toc80903859)

[2.3 产品调优(示例--业务本身调优) 6](#_Toc80903860)

[2.3.1 增加tidb和tikv实例(对应示例1：基础软件) 6](#_Toc80903861)

[2.3.2 增加Kafka分区数(对应示例2：行业应用软件) 6](#_Toc80903862)

[3 调优效果汇总 7](#_Toc80903863)

# 调优概述

## 方案介绍

*介绍方案的基本信息，*

*内容包括：对产品的描述，在行业的应用*

*格式要求：产品哪个公司的什么方案，解决了哪些行业的什么痛点，应用范围如何，技术亮点是什么(可选)*

*示例：*

*XXX有限公司设计开发的XXX解决方案，聚焦XX、XX行业的XX痛点问题，解决了XXX，在XX方面获得了XX。*

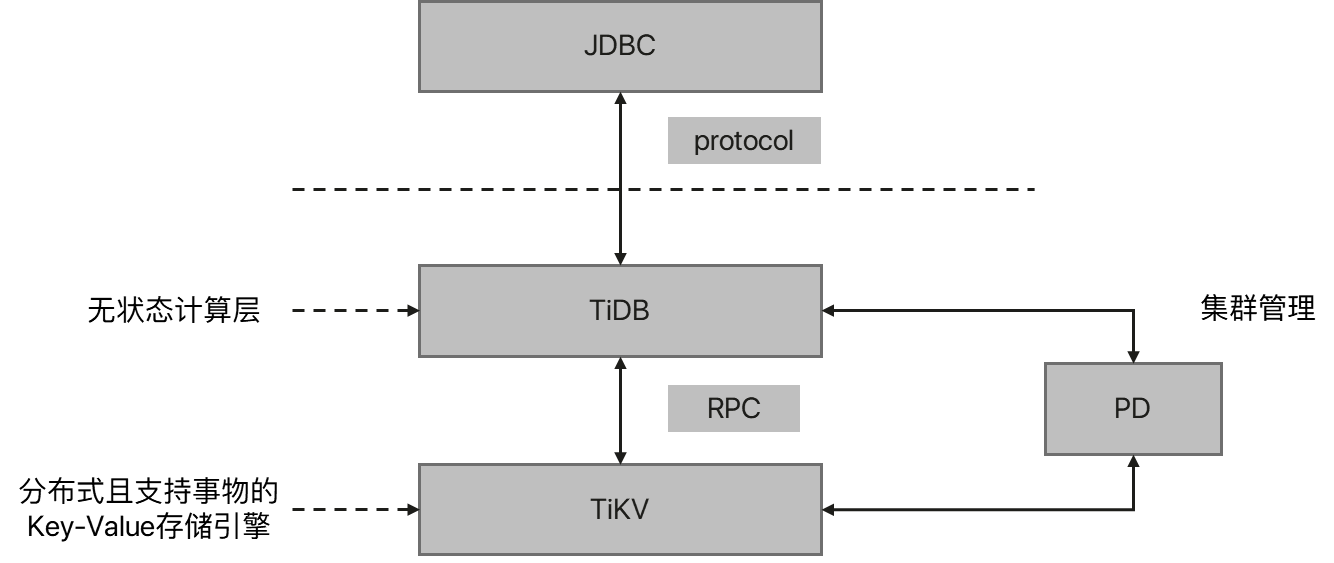
## 调优思路

### 业务流程

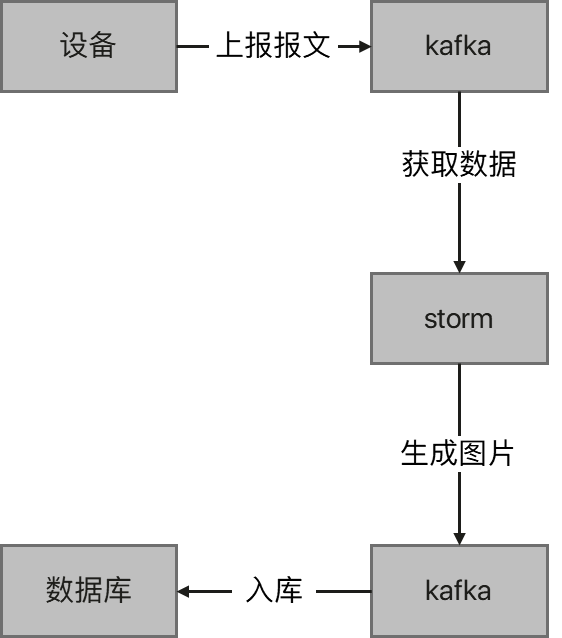
*梳理出业务的数据流、业务流框图，分析业务流程中可能存在的性能瓶颈点，然后结合调优手段或BoostKit特性等进行性能优化。*

*注：这里给出的是一个调优的指导思路，并且是跟产品本身业务相关的方面。对硬件、OS等基础层面的东西，这里可以不用具体体现，但是要能明白硬件、OS的调优措施可能会影响到业务流的那一个或哪些关键环节。*

*示例1：TiDB业务流分析（基础软件）*

**

*示例2：行业应用软件*

**

### 调优思路

*根据业务流程图，分析能存在的瓶颈点，并给出可能解决的调优方案(可选)*

*示例1：基础软件*

*外部请求通过Tikv下发至TiDB，再由TiDB将请求发送至后端数据库驱动做相应操作。当每台机器只有2个TiDB和TiKV时，单台服务器的CPU还有较多空间，增加TiDB和TiKV的实例个数可以提升CPU的使用率，从而提升性能。*

*示例2：行业应用软件*

*瓶颈点分析：*

*1）接收端接受速度比较慢，分析后发现消费端速度一直处于比较慢的水平。*

*2）当数据量大时，在数据传入后spout运行时间明显增加，且CPU利用率低，分析发现可能有数据堆积。*

*解决方案：*

1. *XXX*

*2）YYY*

# 调优方案

*本章节具体描述在硬件、OS、基础软件、应用等各个层面的调优手段以及调优效果；*

*要求：*

*1）说明每一种调优手段的目的是什么，即这个调优手段可以提交哪些方面的性能*

*2）说明每一种调优手段如何实施，可以写出实施的具体操作步骤；*

*3）说明调优手段的提升效果，如TPS性能提升5%*

## BIOS配置(示例--硬件调优部分)

目的

关闭CPU预取：CPU将内存中的数据读到CPU的高速缓冲Cache时，会根据局部性原理，除了读取本次要访问的数据，还会预取本次数据的周边数据到Cache里面，如果预取的数据是下次要访问的数据，那么性能会提升，如果预取的数据不是下次要取的数据，那么会浪费内存带宽。对于数据比较集中的场景，预取的命中率高，适合打开CPU预取，反之需要关闭CPU预取。

方法

1. 在BIOS中，选择“Advanced>MISC Config”，按Enter进入。
2. 将“CPU Prefetching Configuration”设置为“Disabled”，按F10保存退出。

## 操作系统调优(示例--OS调优部分)

### 文件系统调优

目的

对于不同的IO设备，通过调整文件系统相关参数配置，可以有效提升服务器性能。

方法



以xfs文件系统为例，解释文件系统调优步骤。

建议在文件系统的mount参数上加上noatime，nobarrier两个选项。命令为（其中数据盘以及数据目录以实际为准）：

**mount -o noatime,nobarrier /dev/sdb /data**

1. 一般来说，Linux会给文件记录了三个时间，change time, modify time和access time。

* access time指文件最后一次被读取的时间。
* modify time指的是文件的文本内容最后发生变化的时间。
* change time指的是文件的inode最后发生变化（比如位置、用户属性、组属性等）的时间。

一般来说，文件都是读多写少，而且我们也很少关心某一个文件最近什么时间被访问了。所以，我们建议采用noatime选项，文件系统在程序访问对应的文件或者文件夹时，不会更新对应的access time。这样文件系统不记录access time，避免浪费资源。

1. 现在的很多文件系统会在数据提交时强制底层设备刷新cache，避免数据丢失，称之为write barriers。但是，其实我们数据库服务器底层存储设备要么采用RAID卡，RAID卡本身的电池可以掉电保护；要么采用Flash卡，它也有自我保护机制，保证数据不会丢失。所以我们可以安全的使用nobarrier挂载文件系统。

* 对于ext3, ext4和 reiserfs文件系统可以在mount时指定barrier=0。
* 对于xfs可以指定nobarrier选项。

### 网卡中断绑核

目的

手动绑定网卡中断，根据网卡所属CPU将其进行分配，从而优化系统网络性能。

方法

查询网卡所在的CPU，将网络中断绑定到该CPU的所有核上。

关闭irqbalance。

停止irqbalance服务

systemctl stop irqbalance.service

关闭irqbalance服务

systemctl disable irqbalance.service

查看irqbalance服务状态，确认服务已经关闭

systemctl status irqbalance.service

查询中断号

cat /proc/interrupts | grep enp6s0 | awk -F ':' '{print $1}'

根据中断号，将每个中断各绑定在一个核上。

echo $cpunum > /proc/irq/$irq/smp\_affinity\_list

----结束

### 网络参数调优

目的

调大tcp\_max\_syn\_backlog参数，由2048调整为8192，增加网络吞吐量

方法

echo 8192 > /proc/sys/net/ipv4/tcp\_max\_syn\_backlog

### IO参数调优

目的

调大nr\_requests参数，提升磁盘吞吐量。

方法

echo 2048 > /sys/block/${device}/queue/nr\_requests

### 缓存参数调优

目的

调小swappiness参数，更加积极的使用内存，而非swap分区。

方法

执行命令 vi /etc/sysctl.conf ，将 vm.swappiness = 1添加到文件底部，保存退出，执行命令sysctl -p使其生效。

## 产品调优(示例--业务本身调优)

### 增加tidb和tikv实例(对应示例1：基础软件)

目的

当每台机器只有2个TiDB和TiKV时，单台服务器的CPU还有较多空间，增加TiDB和TiKV的实例个数可以提升CPU的使用率，从而提升性能。另外无限水平扩展是 TiDB 的一大特点，这里说的水平扩展包括两方面：计算能力和存储能力。TiDB Server 负责处理 SQL 请求，随着业务的增长，可以简单的添加 TiDB Server 节点，提高整体的处理能力，提供更高的吞吐。TiKV 负责存储数据，随着数据量的增长，可以部署更多的 TiKV Server 节点解决数据 Scale 的问题。

方法

1. 对于实例个数测试，单节点部署4tidb和4tikv性能最优。
2. 4tikb和4tikv进程均需要按numa进行范围绑核，即4个tidb进程分别分布在4个numa节点，同理，4个tikv进程也分别分布在4个numa节点。

### 增加Kafka分区数(对应示例2：行业应用软件)

目的

因为XXX，所以XXX，

方法

1. XXX
2. YYY

# 调优效果汇总

*描述每个调优措施达成的效果（负向想过可以不体现），正向显著效果重点标注。*

*示例：*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 调优方案 | 具体措施 | 性能指标1(主要) | 性能指标2(次要) | 性能提升 |
| 0 |  |  | **26** | **28** | **基准数据** |
| 1 | BIOS | 关闭CPU预存取 | 12.8 | 13 | ！负向！ |
| 2 | **BIOS** | **打开支持aio** | **36** | **38** | **提升较大** |
| 3 | OS-IO调优 | IO调度:none | 35.1 | 38 | 基本不变 |
| 4 | OS-中断 | 关闭irqbalance服务 | 35.3 | 38 | 基本不变 |
| 5 | OS-网卡调优 | PCIE Max Payload Size 128B | 35 | 38 | 基本不变 |
| 6 | **OS-vCPU pin** | **40AI,vCPU pin** | **39.1** | **39** | **提升较大** |
| 7 | OS-倒序绑核 | 中断倒序绑核 | 38.8 | 38 | 基本不变 |
| 8 | OS-透明大页 | 关闭透明大页，开启内存大页 | 35.9 | 37 | 略有下降 |
| 9 | 绑核 | 40AI,vCPU pin,中断绑核，none | 39.7 | 38 | 略有提升 |
| 10 | BoostKit-VTurbo | vm 2线程 | **40.5** | **39** | **最终数据** |