Atitit Mysql性能提升 去除tcp协议直连模式

目录

[1.1. 使用127.0.0.1连接 1](#_Toc31553)

[1.2. Tcp协议大概损失20--50%性能。。 1](#_Toc3665)

[1.3. 嵌入式MySQL服务器库 3](#_Toc30003)

[1.4. 压缩协议： 不要认为使用压缩协议总是会提高性能。 4](#_Toc17675)

[2. 一、通信协议 4](#_Toc23038)

[2.1.1. 驗證已啟用命名管道 5](#_Toc6110)

## 使用127.0.0.1连接

最开始 127.0.0.1也是走 tcp协议栈的，很多冗余的东西，而 AF\_UNIX 更像跨进程管道，因此会快很多。但是后面 Linux持续优化，给 127.0.0.1加了很多 fast path。使得本机内部的tcp性能和 AF\_UNIX差不多，也和管道一个数量级了，所以今天 Linux下，保证代码的简单性，没有非要用 AF\_UNIX的必要。

而 Windows下，就没有这样的 fast path，你用 127.0.0.1通信会走长长的一串协议栈代码，性能比管道差远了。所以说，高性能服务端，Windows始终是一个玩具。

## Tcp协议大概损失20--50%性能。。

我测试是命名管道大概损失30%

 使用命名管道方式连接 MySQL 只适合 连接本机的 MySQL ，性能可比一般的TCP/IP方式提升30%~50%。

本地调用 嵌入式db

Linux平台环境下主要有两种连接方式，一种是TCP/IP连接方式，另一种就是socket连接。

在Windows平台下，有name pipe和share memory（不考虑）两种。

TCP/IP连接是网络中用得最多的一种方式。

不知道你怎么测试的，有个[rigtorp/ipc-bench · GitHub](https://link.zhihu.com/?target=https://github.com/rigtorp/ipc-bench" \t "https://www.zhihu.com/question/_blank) 比较了各种IPC的效率。  
我测试的结果是：  
UDS的QPS是loopback(127.0.0.1)的564%，吞吐是568%

Unix Domain Socket 3.57ms、3.57ms、3.35ms

UDS直接写入的是对端的socket buffer，要比loopback少非常多的“累赘”：

1. checksum
2. 各种Header
3. IP路由
4. etc...

===================  
还要多说一句，现在很少会有程序瓶颈出现在这个层面了，考虑用UDS还是IP还是要从功能方面出发：

* UDS基于文件系统，可以做基于文件系统权限的ACL
* IP方便做scaling，改个配置就变成分布式了

最开始 127.0.0.1也是走 tcp协议栈的，很多冗余的东西，而 AF\_UNIX 更像跨进程管道，因此会快很多。但是后面 Linux持续优化，给 127.0.0.1加了很多 fast path。使得本机内部的tcp性能和 AF\_UNIX差不多，也和管道一个数量级了，所以今天 Linux下，保证代码的简单性，没有非要用 AF\_UNIX的必要。

而 Windows下，就没有这样的 fast path，你用 127.0.0.1通信会走长长的一串协议栈代码，性能比管道差远了。所以说，高性能服务端，Windows始终是一个玩具。

MySQL中有个profiler的开关，打开之后query profiler能看到所有执行过的语句的执行时间。先看看这个执行时间到底是多久。如果这个执行时间本身就3毫秒左右的话，你换什么样的传输方法也不会有太大的提升的。

12

如果要将UNIX套接字与Mysql JDBC Connector / J一起使用，则需要提供一个*[socketFactory](https://dev.mysql.com/doc/connector-j/en/connector-j-reference-configuration-properties.html)*。

jdbc:mysql:///?user=test&password=test&socketFactory=<classname>&<socket>=/tmp/mysql.sock

因此，这将随您使用的实现而有所不同。默认情况下，Mysql不附带任何实现，仅在源代码中提供了此类工厂的示例。

现有一个名为***[junixsocket的](https://github.com/kohlschutter/junixsocket)***UNIX套接字Java库，它也具有这样的*socketFactory*类实现。*[通过Unix域套接字连接到MySQL数据库中](https://code.google.com/p/junixsocket/wiki/ConnectingToMySQL)*概述了一个示例，该示例是其文档的一部分。

[另一个答案](https://dba.stackexchange.com/a/74858)指出，这是可能的（至少在的IntelliJ IDE），以使用套接字连接[MySQL的JDBC连接器/ J](https://dev.mysql.com/doc/connector-j/5.1/en/)（我使用的版本5.1） ****，而不****不必提供的SocketFactory，

jdbc：mysql：// localhost：3306 / database\_name？socket = / tmp / mysql.sock

## 嵌入式MySQL服务器库

[27.7.1使用libmysqld编译程序](http://devdoc.net/database/MySQL-refman-5.7/connectors-apis.html" \l "libmysqld-compiling)

[27.7.2使用嵌入式MySQL服务器时的限制](http://devdoc.net/database/MySQL-refman-5.7/connectors-apis.html" \l "libmysqld-restrictions)

[27.7.3嵌入式服务器的选项](http://devdoc.net/database/MySQL-refman-5.7/connectors-apis.html" \l "libmysqld-options)

[27.7.4嵌入式服务器示例](http://devdoc.net/database/MySQL-refman-5.7/connectors-apis.html" \l "libmysqld-example)

嵌入式MySQL服务器库使在客户端应用程序中运行功能齐全的MySQL服务器成为可能。主要优点是可以提高速度，并简化嵌入式应用程序的管理。

**注意**

## ****压缩协议：**** 不要认为使用压缩协议总是会提高性能。

使用压缩协议将减少要通过网络传输的数据量，但将导致服务器和客户端上的额外负载（压缩和解压缩）。在本地计算机，本地网络或非常快速的Internet连接上连接MySQL时，最好不要使用该选项。但是，如果网络速度较慢，则可能会提高整体性能。由于大多数Internet连接具有非对称属性，这在Internet提供商和电信公司之间是不同的，因此不可能更具体。

# ****一、通信协议****

****1、TCP/IP协议****

    ﻿> 通常我们通过这个协议来连接MySQL，各种主要编程语言都是根据这个协议实现了连接模块

****2、Unix Socket协议****

    ﻿> 通常我们登入MySQL服务器中使用这个协议，因为要使用这个协议连接MySQL需要一个物理文件，文件的存放位置在配置文件中有定义，值得一提的是，这是所有协议中最高效的一个.

****3、Share Memory协议****

    ﻿> 这个协议一般人不知道，肯定也没用过，因为这个只有windows可以使用，使用这个协议需要在配置文件中在启动的时候使用–shared-memory参数，注意的是，使用此协议，一个host上只能有一个server，所以这个东西一般没啥用的，除非你怀疑其他协议不能正常工作，实际上微软的SQL Sever也支持这个协议

****4、Named Pipes协议****

    ﻿> 这个协议也是只有windows

### **驗證已啟用命名管道**

要確保啟用命名管道，請運行以下代碼：SHOW GLOBAL VARIABLES LIKE 'named\_pipe'

enable-named-pipe參數添加到該mysqld部分。考慮以下mysqld部分的示例：

[mysqld]# The next three options are mutually exclusive to SERVER\_PORT below.# skip-networking

enable-named-pipe# shared-memory# shared-memory-base-name=MYSQL# The Pipe the MySQL Server will usesocket=MYSQL