# MySQL现有架构对插入性能的制约

## 一、两种语句的执行

insert table\_name values (v1, v2,…vn);

insert table\_name values (v1, v2,…vn), (v1, v2,…vn);

这两类的语句都将作为一条SQL语句进行执行，都将通过一系列函数调用进入到Sql\_cmd\_insert::mysql\_insert的函数中，完成插入功能。

由于存储引擎接口只支持单行记录的插入，因此即使在函数Sql\_cmd\_insert::mysql\_insert中，接受多行数据的输入，也只能将这些行数逐一通过引擎接口，传递给具体的存储引擎。在这个过程中，有很多重复的函数调用，这些函数调用完成相同功能。

## 附录

### 测试

通过对10w行数据的测试

例子：Create table t1(f1 int, f2 varchar(10))

极限速度（注1），省去所有重复操作内容，只进行索引插入操作

insert t1 values (1, '1');

5.63 sec

一次语法解析的单条SQL

insert t1 values (1, '1') , (1, '1')…, (1, '1');

7.35 sec

10w次语法解析的多条SQL

insert t1 values (1, '1');…insert t1 values (1, '1');

40.35 sec

注1：修改源码，去掉重复的调用部分，对同一条记录进行反复插入

