**苏建飞：**

并发版本 http服务器

**丁昌茂：**

**python适合的场景有哪些？当遇到计算密集型任务怎么办？**

适用场景:网站运维,金融分析,服务器编写,爬虫当遇到io密集型任务时,涉及到的大多是网络,磁盘等任务,这一类任务的特性是cpu小号低,使用多线程.计算密集型的任务主要是消耗cpu性能,谁要运用多进程,当然运用python语言的运行效率很低,所以一般对于计算密集型任务,可以使用c语言编写

**说一下 MySQL 数据库存储的原理?**

回答技巧:先回答一下mysql的原理，拓展一下它的有点，或者mysql你是怎么用的？回答：储存过程是一个可编程的函数，它在数据库中创建并保存。它可以有 SQL 语句和一些特殊的控制结构组成。当希望在不同的应用程序或平台上执行相同的函数，或者封装特定功能时，存储过程是非常有用的。数据库中的存储过程可以看做是对编程中面向对象方法的模拟。它允许控制数据的访问方式。 存储过程通常有以下优点:1)存储过程能实现较快的执行速度。2)存储过程允许标准组件是编程。3)存储过程可以用流控制语句编写，有很强的灵活性，可 以完成复杂的判断和较复杂的运算。4)存储过程可被作为一种安全机制来充分利用。5)存储过程能过减少网络流量

**王福成：**

数据库和事务相关的两个面试题

1、事务的并发问题

从理论上来说, 事务应该彼此完全隔离, 以避免并发事务所导致的问题，然而, 那样会对性能产生极大的影响, 因为事务必须按顺序运行， 在实际开发中, 为了提升性能, 事务会以较低的隔离级别运行， 事务的隔离级别可以通过隔离事务属性指定。

1、脏读：事务A读取了事务B更新的数据，然后B回滚操作，那么A读取到的数据是脏数据

2、不可重复读：事务 A 多次读取同一数据，事务 B 在事务A多次读取的过程中，对数据作了更新并提交，导致事务A多次读取同一数据时，结果因此本事务先后两次读到的数据结果会不一致。

3、幻读：幻读解决了不重复读，保证了同一个事务里，查询的结果都是事务开始时的状态（一致性）。

例如：事务T1对一个表中所有的行的某个数据项做了从“1”修改为“2”的操作 这时事务T2又对这个表中插入了一行数据项，而这个数据项的数值还是为“1”并且提交给数据库。 而操作事务T1的用户如果再查看刚刚修改的数据，会发现还有跟没有修改一样，其实这行是从事务T2中添加的，就好像产生幻觉一样，这就是发生了幻读。

小结：不可重复读的和幻读很容易混淆，不可重复读侧重于修改，幻读侧重于新增或删除。解决不可重复读的问题只需锁住满足条件的行，解决幻读需要锁表。

**陈中原：**

**mysql 优化：**

1、应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。

2、对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。

3、应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描

4、尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描

5、前置百分号，查询也将导致全表扫描。

6、in 和 not in 也要慎用，否则会导致全表扫描

7、如果在 where 子句中使用参数，也会导致全表扫描。因为SQL只有在运行时才会解析局部变量，但优化程序不能将访问计划的选择推迟到运行时；它必须在编译时进行选择。然 而，如果在编译时建立访问计划，变量的值还是未知的，因而无法作为索引选择的输入项。如下面语句将进行全表扫描

8、应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描

9、应尽量避免在where子句中对字段进行函数操作，这将导致引擎放弃使用索引而进行全表扫描

10、不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引