

# 快速上手Jmeter性能测试工具-第12讲

## ---分布式Jmeter

- 1、压力不够大
- 2、单台机瓶颈
- 3、网络瓶颈

分布式概念的引入

什么是负载生成器？

使用分布式负载生成器的特点：

- 1、真实的性能测试而不存在网络瓶颈问题
- 2、快速响应的GUI
- 3、将测试结果存储到本地的一台机器上
- 4、使用一台机器管理多个JMeter Engines

同时，测试脚本会被自动发送到作为负载生成的机器上并执行，不用使用者单独执行拷贝文件操作。

## 需要注意的：

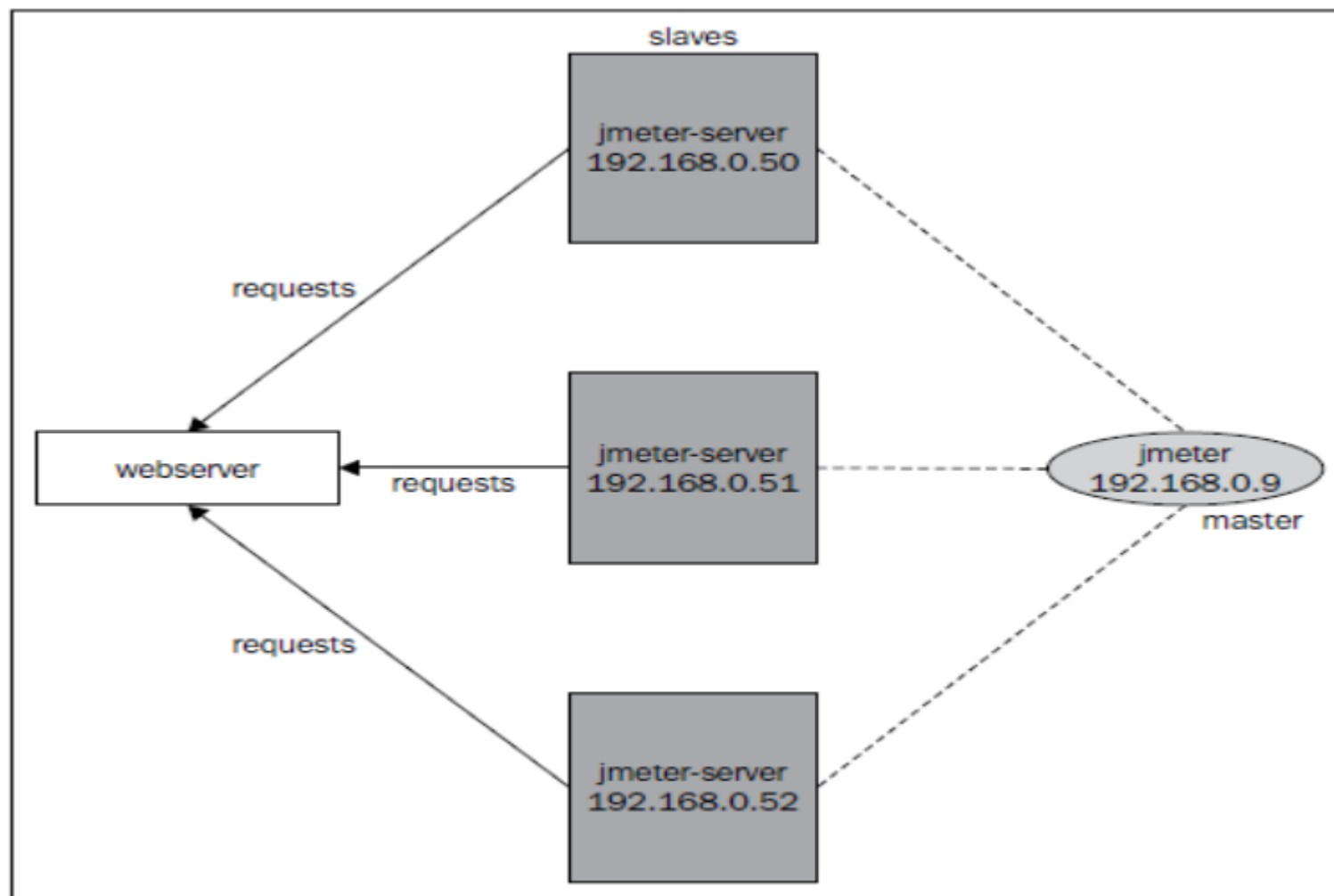
- 1、注意控制台的负载
- 2、所有负载生成器上安装相同版本的Jmeter
- 3、所有负载生成器上安装相同版本的Java
- 4、防火墙和杀毒软件
- 5、确保所有机器在同一个子网

1台机器到底可以模拟多少并发用户？

A single JMeter client running on a 2-3Ghz CPU (recent cpu) can handle 300-600 threads depending on the type of test.

特别是对于XML文件的解析，更是CPU密集型。

# Jmeter中的分布式



JMeter distributed testing architecture

- 1、在负载生成器（ slaves ）启动 jmeter-server ；
- 2、在控制台（ master ）编辑 jmeter.properties 文件。查找 remote\_hosts= 并根据实际情况添加 slave 的 IP 地址到列表中 ；
- 3、在 master 端启动 Jmeter ；
- 4、开始运行。

带有其他外部文件的例子

结论：如果测试用例中使用了其他外部数据文件，这些文件是不会随着测试计划一起发送到负载生成器上的，因此需要确认测试计划所使用的数据文件是否存在于运行测试计划的远程机器上。



- 1、都可以实现分布式负载。相对来说LoadRunner更强大一些。
- 2、都支持在windows和Linux环境的负载生成器。控制台方面，Jmeter跨平台，而LoadRunner不是。
- 3、**LoadRunner可以指定每个负载生成器运行不同数量的并发用户，而目前Jmeter不行。**
- 4、Jmeter中测试计划之外的数据文件不会一起发送到负载生成器，而LoadRunner中的文件可以通过选择纳入LoadRunner的管理而一起发送到远端。