



(12)发明专利申请

(10)申请公布号 CN 110417613 A

(43)申请公布日 2019.11.05

(21)申请号 201910523487.4

G06F 8/61(2018.01)

(22)申请日 2019.06.17

(71)申请人 平安科技(深圳)有限公司

地址 518033 广东省深圳市福田区福田街
道福安社区益田路5033号平安金融中
心23楼

(72)发明人 李润妮

(74)专利代理机构 北京市京大律师事务所

11321

代理人 刘挽澜

(51)Int.Cl.

H04L 12/26(2006.01)

H04L 29/08(2006.01)

G06F 9/455(2006.01)

G06F 8/71(2018.01)

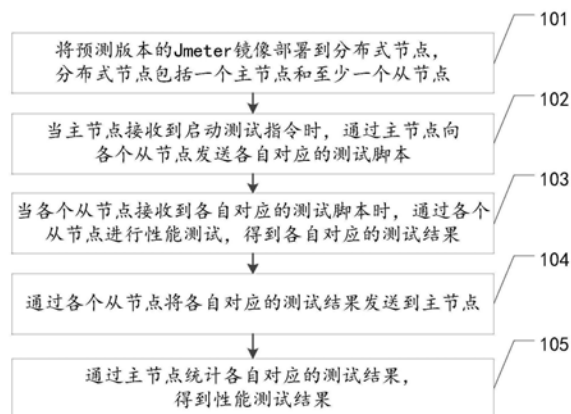
权利要求书2页 说明书12页 附图5页

(54)发明名称

基于Jmeter的分布式性能测试方法、装置、
设备及存储介质

(57)摘要

本发明涉及云测试领域,公开了一种基于Jmeter的分布式性能测试方法、装置、设备及存储介质。基于Jmeter的分布式性能测试方法包括:将预测版本的Jmeter镜像部署到分布式节点,分布式节点包括一个主节点和至少一个从节点;当主节点接收到启动测试指令时,通过主节点向各个从节点发送各自对应的测试脚本;当各个从节点接收到各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;通过各个从节点将各自对应的测试结果发送到主节点;通过主节点统计各自对应的测试结果,得到性能测试结果。本发明对Jmeter镜像进行版本控制,采用基于容器的分布式节点部署Jmeter镜像进行性能测试,部署简单快捷,提高测试效率。



1. 一种基于Jmeter的分布式性能测试方法,其特征在于,包括:

将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点;

当所述主节点接收到启动测试指令时,通过所述主节点向各个从节点发送各自对应的测试脚本;

当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;

通过各个从节点将所述各自对应的测试结果发送到所述主节点;

通过所述主节点统计所述各自对应的测试结果,得到性能测试结果。

2. 根据权利要求1所述的基于Jmeter的分布式性能测试方法,其特征在于,所述将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点包括:

从镜像仓库中选择预测版本的Jmeter镜像;

通过弹性计算服务ECS实例设置所述分布式节点,所述分布式节点包括所述一个主节点和所述至少一个从节点;

判断所述主节点和各个从节点是否同时存在所述预测版本的Jmeter镜像;

若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果;

若所述主节点和各个从节点同时存在所述预测版本的Jmeter镜像,则确定部署成功。

3. 根据权利要求2所述的基于Jmeter的分布式性能测试方法,其特征在于,所述若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果包括:

若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则将所述预测版本的Jmeter镜像部署到所述ECS实例上,得到Jmeter应用;

通过容器docker运行所述Jmeter应用;

将各个从节点对应的配置信息添加到所述主节点的控制列表中,所述各个从节点对应的配置信息包括各自对应的IP地址和各自对应的端口;

重启所述主节点的Jmeter应用,得到部署结果。

4. 根据权利要求2所述的基于Jmeter的分布式性能测试方法,其特征在于,所述若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果之后,所述基于Jmeter的分布式性能测试方法还包括:

判断所述部署结果是否为目标值,所述目标值用于指示成功部署所述预测版本的Jmeter镜像;

若所述部署结果不为所述目标值,则重新部署;

若所述部署结果为所述目标值,则确定部署成功。

5. 根据权利要求3所述的基于Jmeter的分布式性能测试方法,其特征在于,所述当所述主节点接收到启动测试指令时,通过所述主节点向各个从节点发送各自对应的测试脚本包括:

将所述主节点设置为远程方法调用RMI的客户端;

将各个从节点设置为所述RMI的服务器；

当所述主节点接收到启动测试指令时,通过所述主节点执行测试计划,所述测试计划包括总测试数据和测试脚本；

解析所述主节点的控制列表,得到所述各个从节点对应的配置信息；

根据所述各个从节点对应的配置信息对所述总测试数据进行拆分,得到各个从节点对应的测试数据；

根据所述各个从节点对应的测试数据替换所述测试脚本中的关键字,得到所述各自对应的测试脚本；

通过所述主节点以所述RMI方式向各个从节点推送所述各自对应的测试脚本。

6. 根据权利要求5所述的基于Jmeter的分布式性能测试方法,其特征在于,所述当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果包括：

当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点解析所述各自对应的测试脚本,得到各自对应的测试请求；

通过各个从节点执行所述各自对应的测试请求,得到所述各自对应的测试结果；

通过各个从节点记录所述各自对应的测试结果。

7. 根据权利要求1至6中任一项所述的基于Jmeter的分布式性能测试方法,其特征在于,所述将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点之前,所述基于Jmeter的分布式性能测试方法还包括：

将所述预测版本的Jmeter镜像上传到所述镜像仓库；

通过打标签方式对所述预测版本的Jmeter镜像进行版本管理。

8. 一种基于Jmeter的分布式性能测试装置,其特征在于,所述基于Jmeter的分布式性能测试装置包括：

部署单元,用于将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点；

控制单元,当所述主节点接收到启动测试指令时,用于通过所述主节点向各个从节点发送各自对应的测试脚本；

测试单元,当各个从节点接收到所述各自对应的测试脚本时,用于通过各个从节点进行性能测试,得到各自对应的测试结果；

发送单元,用于通过各个从节点将所述各自对应的测试结果发送到所述主节点；

统计单元,用于通过所述主节点统计所述各自对应的测试结果,得到性能测试结果。

9. 一种基于Jmeter的分布式性能测试设备,其特征在于,所述基于Jmeter的分布式性能测试设备包括:存储器和至少一个处理器,所述存储器中存储有指令,所述存储器和所述至少一个处理器通过线路互联；

所述至少一个处理器调用所述存储器中的所述指令,以使得所述基于Jmeter的分布式性能测试设备执行如权利要求1-7中任意一项所述的方法。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于:所述计算机程序被处理器执行时实现如权利要求1-7中任意一项所述方法的步骤。

基于Jmeter的分布式性能测试方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及云测试领域,尤其涉及基于Jmeter的分布式性能测试方法、装置、设备及存储介质。

背景技术

[0002] 服务器端性能测试的主流工具有Jmeter和LoadRunner,原理都是通过中间代理,监控收集并发客户端发送的指令,生成脚本发送到应用服务器,再监控服务器反馈的结果。

[0003] 对应用程序做接口性能测试时,由于单机CPU和内存的限制,单机部署的Jmeter无法满足测试需求,需要使用Jmeter分布式部署。传统的应用部署方式是通过插件或脚本来安装应用,这样做的缺点是应用程序的运行、配置、管理、所有生存周期将与当前操作系统绑定,这样做并不利于应用的升级更新和版本回滚等操作。

[0004] 通过创建虚拟机的方式部署Jmeter时,则部署过程复杂,效率低,并不利于可移植性,同时基于大量虚拟机的Jmeter集群有个缺点,就是不方便管理,比如需要重启Jmeter服务,还需要连接到每个虚拟机上进行操作,而且安装部署不方便,也不方便将集群封装成整体应用对外提供服务。

发明内容

[0005] 本发明的主要目的在于解决单机部署Jmeter无法满足测试需求,而采用虚拟机部署Jmeter过程复杂、效率低和移植性差的问题,同时解决了测试脚本对版本控制支持不友好的问题。

[0006] 为实现上述目的,本发明第一方面提供了一种基于Jmeter的分布式性能测试方法,包括:将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点;当所述主节点接收到启动测试指令时,通过所述主节点向各个从节点发送各自对应的测试脚本;当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;通过各个从节点将各自对应的测试结果发送到所述主节点;通过所述主节点统计所述各自对应的测试结果,得到性能测试结果。

[0007] 可选的,在本发明第一方面的第一种实现方式中,所述将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点包括:从镜像仓库中选择预测版本的Jmeter镜像;通过弹性计算服务ECS实例设置所述分布式节点,所述分布式节点包括所述一个主节点和所述至少一个从节点;判断所述主节点和各个从节点是否同时存在所述预测版本的Jmeter镜像;若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果;若所述主节点和各个从节点同时存在所述预测版本的Jmeter镜像,则确定部署成功。

[0008] 可选的,在本发明第一方面的第二种实现方式中,所述若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果包括:若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜

像,则将所述预测版本的Jmeter镜像部署到所述ECS实例上,得到Jmeter应用;通过容器docker运行所述Jmeter应用;将各个从节点对应的配置信息添加到所述主节点的控制列表中,所述各个从节点对应的配置信息包括各自对应的IP地址和各自对应的端口;重启所述主节点的所述Jmeter应用,得到部署结果。

[0009] 可选的,在本发明第一方面的第三种实现方式中,所述若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果之后,所述基于Jmeter的分布式性能测试方法还包括:判断所述部署结果是否为目标值,所述目标值用于指示成功部署所述预测版本的Jmeter镜像;若所述部署结果不为所述目标值,则重新部署;若所述部署结果为所述目标值,则确定部署成功。

[0010] 可选的,在本发明第一方面的第四种实现方式中,所述当所述主节点接收到启动测试指令时,通过所述主节点向各个从节点发送各自对应的测试脚本包括:将所述主节点设置为远程方法调用RMI的客户端;将各个从节点设置为所述RMI的服务器端;当所述主节点接收到启动测试指令时,通过所述主节点执行测试计划,所述测试计划包括总测试数据和测试脚本;解析所述主节点的控制列表,得到所述各个从节点对应的配置信息;根据所述各个从节点对应的配置信息对所述总测试数据进行拆分,得到各个从节点对应的测试数据;根据所述各个从节点对应的测试数据替换所述测试脚本中的关键字,得到所述各自对应的测试脚本;通过所述主节点以所述RMI方式向各个从节点推送所述各自对应的测试脚本。

[0011] 可选的,在本发明第一方面的第五种实现方式中,所述当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果还包括:当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点解析所述各自对应的测试脚本,得到各自对应的测试请求;通过各个从节点执行所述各自对应的测试请求,得到所述各自对应的测试结果;通过各个从节点记录所述各自对应的测试结果。

[0012] 可选的,在本发明第一方面的第六种实现方式中,所述将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点之前,所述基于Jmeter的分布式性能测试方法还包括:将所述预测版本的Jmeter镜像上传到所述镜像仓库;通过打标签方式对所述预测版本的Jmeter镜像进行版本管理。

[0013] 本发明第二方面提供了一种基于Jmeter的分布式性能测试装置,包括:部署单元,用于将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点;控制单元,当所述主节点接收到启动测试指令时,用于通过所述主节点向各个从节点发送各自对应的测试脚本;测试单元,当各个从节点接收到所述各自对应的测试脚本时,用于通过各个从节点进行性能测试,得到各自对应的测试结果;发送单元,用于通过各个从节点将各自对应的测试结果发送到所述主节点;统计单元,用于通过所述主节点统计所述各自对应的测试结果,得到性能测试结果。

[0014] 可选的,在本发明第二方面的第一种实现方式中,所述部署单元还包括:选择子单元,用于从镜像仓库中选择预测版本的Jmeter镜像;设置子单元,用于通过弹性计算服务ECS实例设置所述分布式节点,所述分布式节点包括所述一个主节点和所述至少一个从节点;第一判断子单元,用于判断所述主节点和各个从节点是否同时存在所述预测版本的Jmeter镜像;部署子单元,若所述主节点和各个从节点不同时存在所述预测版本的Jmeter

镜像,则用于通过预置方式部署所述预测版本的Jmeter镜像,得到部署结果;确认子单元,若所述主节点和各个从节点同时存在所述预测版本的Jmeter镜像,则用于确定部署成功。

[0015] 可选的,在本发明第二方面的第二种实现方式中,所述部署子单元具体用于:若所述主节点和各个从节点不同时存在所述预测版本的Jmeter镜像,则将所述预测版本的Jmeter镜像部署到所述ECS实例上,得到Jmeter应用;通过容器docker运行所述Jmeter应用;将各个从节点对应的配置信息添加到所述主节点的控制列表中,所述各个从节点对应的配置信息包括各自对应的IP地址和各自对应的端口;重启所述主节点的所述Jmeter应用,得到部署结果。

[0016] 可选的,在本发明第二方面的第三种实现方式中,所述部署单元还包括:第二判断子单元,用于判断所述部署结果是否为目标值,所述目标值用于指示成功部署所述预测版本的Jmeter镜像;第一处理子单元,若所述部署结果不为所述目标值,则用于重新部署;第二处理子单元,若所述部署结果为所述目标值,则用于确定部署成功。

[0017] 可选的,在本发明第二方面的第四种实现方式中,所述控制单元具体用于:将所述主节点设置为远程方法调用RMI的客户端;将各个从节点设置为所述RMI的服务器端;当所述主节点接收到启动测试指令时,通过所述主节点执行测试计划,所述测试计划包括总测试数据和测试脚本;解析所述主节点的控制列表,得到各个从节点对应的配置信息;根据所述各个从节点对应的配置信息对所述总测试数据进行拆分,得到各个从节点对应的测试数据;根据所述各个从节点对应的测试数据替换所述测试脚本中的关键字,得到所述各自对应的测试脚本;通过所述主节点以所述RMI方式向各个从节点推送所述各自对应的测试脚本。

[0018] 可选的,在本发明第二方面的第五种实现方式中,所述测试单元具体用于:当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点解析所述各自对应的测试脚本,得到各自对应的测试请求;通过各个从节点执行所述各自对应的测试请求,得到所述各自对应的测试结果;通过各个从节点记录所述各自对应的测试结果。

[0019] 可选的,在本发明第二方面的第六种实现方式中,所述基于Jmeter的分布式性能测试装置还包括:上传单元,用于将所述预测版本的Jmeter镜像上传到所述镜像仓库;管理单元,用于通过打标签方式对所述预测版本的Jmeter镜像进行版本管理。

[0020] 本发明第三方面提供了一种基于Jmeter的分布式性能测试设备,包括:存储器和至少一个处理器,所述存储器中存储有指令,所述存储器和所述至少一个处理器通过线路互联;所述至少一个处理器调用所述存储器中的所述指令,以使得所述基于Jmeter的分布式性能测试设备执行上述第一方面所述的方法。

[0021] 本发明的第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述第一方面所述的方法。

[0022] 从以上技术方案可以看出,本发明具有以下优点:

[0023] 本发明提供的技术方案中,将预测版本的Jmeter镜像部署到分布式节点,所述分布式节点包括一个主节点和至少一个从节点;当所述主节点接收到启动测试指令时,通过所述主节点向各个从节点发送各自对应的测试脚本;当各个从节点接收到所述各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;通过各个从节点将各自对应的测试结果发送到所述主节点;通过所述主节点统计所述各自对应的测试结果,

得到性能测试结果。本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,同时相对于虚拟机,容器部署过程简单化,提高测试效率。

附图说明

- [0024] 图1为本发明实施例中基于Jmeter的分布式性能测试方法的一个实施例示意图;
[0025] 图2为本发明实施例中基于Jmeter的分布式性能测试方法的另一个实施例示意图;
[0026] 图3为本发明实施例中基于Jmeter的分布式性能测试装置的一个实施例示意图;
[0027] 图4为本发明实施例中基于Jmeter的分布式性能测试装置的另一个实施例示意图;
[0028] 图5为本发明实施例中基于Jmeter的分布式性能测试设备的另一个实施例示意图。

具体实施方式

[0029] 本发明实施例提供了一种基于Jmeter的分布式性能测试方法、装置、设备及存储介质,本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,支持版本回滚,同时采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,并且相对于虚拟机,容器能快速部署。由于容器与底层设施和机器文件系统是解耦的,能在不同云和不同版本操作系统间进行迁移,部署过程简单化,提高测试效率。

[0030] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明实施例进行描述。

[0031] 本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第四”等(如果存在)是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外,术语“包括”或“具有”及其任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0032] 为便于理解,下面对本发明实施例的具体流程进行描述,请参阅图1,本发明实施例中基于Jmeter的分布式性能测试方法的一个实施例包括:

[0033] 101、将预测版本的Jmeter镜像部署到分布式节点,该分布式节点包括一个主节点和至少一个从节点;

[0034] 服务器将预测版本的Jmeter镜像部署到分布式节点,该分布式节点包括一个主节点和至少一个从节点,主节点和从节点都采用容器docker.docker是一个开源的应用容器引擎,docker完全使用沙箱机制,相互之间不会有任何接口,docker使用客户端-服务器C/S架构模式。

[0035] 需要说明的是,服务器从镜像仓库中选择预测版本的Jmeter镜像。服务器通过

Jmeter镜像,则没必要再去设置诸如Java虚拟机和属性文件的环境,而只需要专注于创建测试脚本和测试资源,如数据文件便可。一个Jmeter镜像包含测试任务中所有的脚本,根据测试任务和测试脚本名字的对应关系,选择某一版本的Jmeter镜像作为预测版本的Jmeter镜像。Jmeter镜像并不过滤任何Jmeter的命令参数,允许多种运行模式,在基于容器的分布式模式下,该镜像可以被用来建立容器集群。例如,服务器通过容器即服务(containers as a service,CAAS)平台的配置管理设置Jmeter镜像,将配置好的模版进行部署。CAAS是一种由IT负责管理和保护的应用环境,可以通过CAAS构建和部署应用,通过基于容器的虚拟化管理和部署Jemeter容器分布式集群。

[0036] 102、当主节点接收到启动测试指令时,通过主节点向各个从节点发送各自对应的测试脚本;

[0037] 当主节点接收到启动测试指令时,服务器通过主节点向各个从节点发送各自对应的测试脚本。具体的,服务器将主节点设置为远程方法调用RMI的客户端;服务器将各个从节点设置为RMI的服务器端;当主节点接收到启动测试指令时,服务器通过主节点执行测试计划,该测试计划包括总测试数据和测试脚本;服务器解析主节点的控制列表,得到各个从节点对应的配置信息;服务器根据各个从节点对应的配置信息对总测试数据进行拆分,得到各个从节点对应的测试数据;服务器根据各个从节点对应的测试数据替换测试脚本中的关键字,得到各自对应的测试脚本;服务器通过主节点以RMI方式向各个从节点推送各自对应的测试脚本。例如,并发数为700的测试请求数据,分布式节点包括主节点A和从节点B、C、D和E,服务器根据从节点B、C、D和E各自的配置信息情况进行计算,服务器分配从节点B、C、D和E的测试请求数据分别为180、170、190和160,服务器通过主节点A将包含有各自对应的测试请求数据的测试脚本发送到从节点B、C、D和E。

[0038] 需要说明的是,远程方法调用(remote method invocation,RMI)是一种用于实现远程过程调用的应用程序编程接口,支持客户机上运行的程序可以调用远程服务器上的对象,也就是存储于不同地址空间的程序级对象之间彼此进行通信,RMI特性支持在网络环境中分布式操作。RMI包括运行远程方法调用服务的服务器和运行远程方法调用服务的客户端。例如,设置包括Jmeter应用的主节点A作为RMI的客户端,设置包括Jmeter应用的从节点B、C、D和E作为RMI服务的服务器,主节点A和从节点B、C、D和E之间通过RMI的方式进行通讯。

[0039] 103、当各个从节点接收到各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;

[0040] 当各个从节点接收到各自对应的测试脚本时,服务器通过各个从节点进行性能测试,得到各自对应的测试结果。具体的,当各个从节点接收到各自对应的测试脚本时,服务器通过各个从节点解析各自对应的测试脚本,得到各自对应的测试请求;通过各个从节点执行各自对应的测试请求,得到各自对应的测试结果;服务器通过各个从节点记录各自对应的测试结果。

[0041] 需要说明的是,性能测试指标包含并发数、平均响应时长、请求成功率、每秒钟请求或者事务数量的并发数,即QPS/TPS。其中,QPS/TPS与并发数和平均响应时长的关系为QPS/TPS等于并发数与平均响应时间相除。例如,一个上班签到系统,早上9点上班,从8点半到9点的30分钟时长内,用户会登录签到系统进行签到。公司员工为2000人,平均每个员工登录签到系统的时长为5分钟,具体的计算过程为,将30分钟转换为1800秒,QPS为2000除以

1800,平均响应时间为3分钟,换算为秒为180秒,那么计算得到的并发数为200。

[0042] 104、通过各个从节点将各自对应的测试结果发送到主节点;

[0043] 服务器通过各个从节点将各自对应的测试结果发送到主节点。具体的,服务器通过各个从节点接收各自对应的测试结果,服务器通过各个从节点以RMI服务方式将各自对应的测试结果发送到主节点,同时各个从节点记录各自对应的测试结果,该各自对应的测试结果为同一测试计划下的测试请求,但是各自对应的测试结果并不相同,该各自对应的测试结果包括正常测试数据结果和异常测试数据结果。

[0044] 105、通过主节点统计各自对应的测试结果,得到性能测试结果。

[0045] 服务器通过主节点统计各自对应的测试结果,得到性能测试结果。具体的,服务器通过主节点接收各个从节点的各自对应的测试结果,服务器根据预置的统计脚本对各个从节点的各自对应的测试结果进行计算和分类,得到性能测试结果。

[0046] 本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,支持版本回滚,同时采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,并且相对于虚拟机,容器能快速部署。由于容器与底层设施和机器文件系统是解耦的,能在不同云和不同版本操作系统间进行迁移,部署过程简单化,提高测试效率。

[0047] 请参阅图2,本发明实施例中基于Jmeter的分布式性能测试方法的另一个实施例包括:

[0048] 201、从镜像仓库中选择预测版本的Jmeter镜像;

[0049] 服务器从镜像仓库中选择预测版本的Jmeter镜像,镜像是一个包含了软件及必要配置的弹性云服务器模板,至少包含操作系统,还可以包含应用软件和私有软件,例如,应用软件包括数据库软件。镜像分为公共镜像和私有镜像,公共镜像为系统默认提供的镜像,私有镜像为用户创建的镜像。例如,若应用程序是网站或Web服务,则镜像可能会包含Web服务器、相关静态内容和动态页面代码,通过该镜像创建弹性云服务器之后,Web服务器将启动,并且应用程序已准备好接受请求。

[0050] 需要说明的是,服务器从镜像仓库中选择预测版本的Jmeter镜像。服务器通过Jmeter镜像,则没必要再去设置诸如Java虚拟机和属性文件的环境,而只需要专注于创建测试脚本和测试资源,如数据文件便可。一个Jmeter镜像包含测试任务中所有的脚本,根据测试任务和测试脚本名字的对应关系,选择某一版本的Jmeter镜像作为预测版本的Jmeter镜像。Jmeter镜像并不过滤任何Jmeter的命令参数,允许多种运行模式,在基于容器的分布式模式下,该镜像可以被用来建立容器集群。

[0051] 可选的,服务器将预测版本的Jmeter镜像上传到镜像仓库,该镜像仓库是存储docker镜像的仓库,不局限于Jmeter镜像,分为公有仓库和私有仓库;服务器通过打标签方式对预测版本的Jmeter镜像进行版本管理,版本管理就是对不同测试任务的Jmeter镜像根据预置规则打标签,同一测试任务根据测试脚本的不同,也可以有不同的分支和标签,同时Jmeter镜像部署之后还支持版本回滚,使测试部署更加方便快捷。

[0052] 202、通过弹性计算服务ECS实例设置分布式节点,该分布式节点包括一个主节点和至少一个从节点;

[0053] 服务器通过弹性计算服务ECS实例设置分布式节点,该分布式节点包括一个主节

点和至少一个从节点。该分布式节点是由一组通过网络进行通信、为了完成共同的任务而协调工作的计算机节点组成的系统。分布式节点的出现是为了用廉价的并且普通的机器完成单个计算机无法完成的计算、存储任务。例如,服务器通过容器即服务(containers as a service,CAAS)平台的配置管理设置Jmeter镜像,将配置好的模版进行部署。CAAS是一种由IT负责管理和保护的应用环境,可以通过CAAS构建和部署应用,通过基于容器的虚拟化管理和部署Jemeter容器分布式集群。

[0054] 需要说明的是,在实际部署时,服务器根据性能测试指标的并发数,动态调整从节点数量,例如,性能测试指标的并发数为1000,每个从节点最大的并发数为200,那么至少需要5个从节点。

[0055] 203、判断主节点和各个从节点是否同时存在预测版本的Jmeter镜像;

[0056] 服务器判断主节点和各个从节点是否同时存在预测版本的Jmeter镜像,具体的,服务器判断主节点和各个从节点指定目录中是否包含目标镜像安装文件,若主节点和各个从节点指定目录中不包含目标镜像安装文件,则确定不同时存在预测版本的Jmeter镜像;若主节点和各个从节点指定目录中包含目标镜像安装文件,则读取目标镜像安装文件,得到版本号;判断版本号是否与预测版本的Jmeter镜像的版本号匹配,若Jmeter镜像安装文件的版本号与预测版本的Jmeter镜像的版本号匹配,则确定存在预测版本的Jmeter镜像;若版本号与预测版本的Jmeter镜像的版本号不匹配,则确定不同时存在预测版本的Jmeter镜像。例如,从节点B中,目标镜像安装文件中的版本号为j20190122,同时预测版本的Jmeter镜像的版本号为j20190123,那么目标镜像安装文件中的版本号与预测版本的Jmeter镜像的版本号不匹配,则从节点B中不存在预测版本的Jmeter镜像。

[0057] 204、若主节点和各个从节点不同时存在预测版本的Jmeter镜像,则通过预置方式部署预测版本的Jmeter镜像,得到部署结果;

[0058] 若主节点和各个从节点不同时存在预测版本的Jmeter镜像,则通过预置方式部署预测版本的Jmeter镜像,得到部署结果,具体的,若主节点和各个从节点不同时存在预测版本的Jmeter镜像,则服务器将预测版本的Jmeter镜像部署到ECS实例上,得到Jmeter应用;服务器通过容器docker运行Jmeter应用;服务器将各个从节点添加到主节点的控制列表中;服务器重启主节点的Jmeter应用,得到部署结果。其中,一个ECS实例等同于一台虚拟机,包含CPU、内存、操作系统、网络和磁盘等基础的计算组件,根据业务场景和使用场景,ECS实例可以分为多种规格族。同一业务场景下,还可以选择新旧多种规格族。同一个规格族里,根据CPU和内存的配置,可以分为多种不同的规格。ECS实例规格定义了实例的CPU和内存的配置,包括CPU型号和主频等。

[0059] 可选的,服务器判断部署结果是否为目标值,目标值指示成功部署预测版本的Jmeter镜像;若部署结果不为目标值,则服务器重新部署预测版本的Jmeter镜像;若部署结果为目标值,则确定部署成功。具体的,服务器通过kubernetes平台查询部署Jmeter镜像的状态,并据此判断该Jmeter镜像部署是否成功。其中,kubernetes是一个开源的容器编排引擎,它支持自动化部署、大规模可伸缩和应用容器化管理。在生产环境中部署一个应用程序时,通常要部署该应用的多个实例以便对应用请求进行负载均衡。例如,在kubernetes中创建多个容器,每个容器里面运行一个Jmeter应用,通过内置的负载均衡策略,实现对这一组Jmeter应用的管理和访问。

[0060] 205、若主节点和各个从节点同时存在预测版本的Jmeter镜像,则确定部署成功;

[0061] 若主节点和各个从节点同时存在预测版本的Jmeter镜像,则服务器确定部署成功。具体的,服务器循环遍历部署预测版本的Jmeter镜像到主节点和各个从节点,并逐个排查部署结果,直到全部部署成功后,执行步骤206。

[0062] 206、当主节点接收到启动测试指令时,通过主节点向各个从节点发送各自对应的测试脚本;

[0063] 当主节点接收到启动测试指令时,服务器通过主节点向各个从节点发送各自对应的测试脚本。具体的,服务器将主节点设置为远程方法调用RMI的客户端;服务器将各个从节点设置为RMI的服务器端;当主节点接收到启动测试指令时,服务器通过主节点执行测试计划,该测试计划包括总测试数据和测试脚本;服务器解析主节点的控制列表,得到各个从节点对应的配置信息;服务器根据各个从节点对应的配置信息对总测试数据进行拆分,得到各个从节点对应的测试数据;服务器根据各个从节点对应的测试数据替换测试脚本中的关键字,得到各自对应的测试脚本,例如,根据测试脚本1扩展出来各个从节点各自对应的测试脚本2、3、4和5,各自对应的测试脚本2、3、4和5执行的测试任务相同,只是测试数据有所不同;服务器通过主节点以RMI方式向各个从节点推送各自对应的测试脚本。其中,主节点与各个从节点之间进行通讯,但是各个从节点之间互相之间并不进行通讯。例如,并发数为700的测试请求,分布式节点包括主节点A和从节点B、C、D和E,服务器根据从节点B、C、D和E的配置信息情况进行计算,服务器分配从节点B、C、D和E各自的测试请求数量分别为180、170、190和160,服务器通过主节点A将包含有该测试请求数量的各自对应的测试脚本发送到从节点B、C、D和E。

[0064] 需要说明的是,远程方法调用(remote method invocation,RMI)是一种用于实现远程过程调用的应用程序编程接口,支持客户机上运行的程序可以调用远程服务器上的对象,也就是存储于不同地址空间的程序级对象之间彼此进行通信,RMI特性支持在网络环境中分布式操作。RMI包括运行远程方法调用服务的服务器和运行远程方法调用服务的客户端。例如,设置主节点A最为RMI的客户端,设置从节点B、C、D和E作为RMI的服务器,主节点A和从节点B、C、D和E之间通过RMI的方式进行通讯。

[0065] 207、当各个从节点接收到各自对应的测试脚本时,通过各个从节点进行性能测试,得到各自对应的测试结果;

[0066] 当各个从节点接收到各自对应的测试脚本时,服务器通过各个从节点进行性能测试,得到各自对应的测试结果。具体的,当各个从节点接收到各自对应的测试脚本时,服务器通过各个从节点解析各自对应的测试脚本,得到各自对应的测试请求;通过各个从节点执行各自对应的测试请求,得到各自对应的测试结果;服务器通过各个从节点记录各自对应的测试结果,例如,将各自对应的测试结果记录在数据库或者指定的目录文件中,具体此处不做限定。

[0067] 需要说明的是,性能测试指标包含并发数、平均响应时长、请求成功率、每秒钟请求或者事务数量的并发数,即QPS/TPS。其中,QPS/TPS与并发数和平均响应时长的关系为QPS/TPS等于并发数与平均响应时间相除。例如,一个上班签到系统,早上9点上班,从8点半到9点的30分钟时长内,用户会登录签到系统进行签到。公司员工为2000人,平均每个员工登录签到系统的时长为5分钟,具体的计算过程为,将30分钟转换为1800秒,QPS为2000除以

1800,平均响应时间为3分钟,换算为秒为180秒,那么计算得到的并发数为200。

[0068] 208、通过各个从节点将各自对应的测试结果发送到主节点;

[0069] 服务器通过各个从节点将各自对应的测试结果发送到主节点。具体的,服务器通过各个从节点接收各自对应的测试结果,服务器通过各个从节点以RMI服务方式将各自对应的测试结果发送到主节点,该各自对应的测试结果为同一测试计划下的测试请求,但是各自对应的测试结果并不相同。该测试结果包括正常测试数据和异常测试数据。例如,从节点B、C、D和E各自对应的测试请求数量为180、170、190和160,那么各自对应的测试结果的数量也为180、170、190和160。

[0070] 209、通过主节点统计各自对应的测试结果,得到性能测试结果。

[0071] 服务器通过主节点统计各自对应的测试结果,得到性能测试结果。具体的,服务器通过主节点接收各个从节点的各自对应的测试结果,服务器根据预置的统计脚本对各个从节点的各自对应的测试结果进行计算和分类,得到性能测试结果。进一步地,服务器通过主节点的Jmeter应用对性能测试结果进行展示。

[0072] 需要说明的是,每个从节点的吞吐量与用户请求对CPU的消耗紧密关联。单个用户请求对CPU消耗越高,每个从节点的吞吐能力越低,也就是处理的并发数越少。

[0073] 本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,支持版本回滚,同时采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,并且相对于虚拟机,容器能快速部署。由于容器与底层设施和机器文件系统是解耦的,能在不同云和不同版本操作系统间进行迁移,部署过程简单化,提高测试效率。

[0074] 上面对本发明实施例中基于Jmeter的分布式性能测试方法进行了描述,下面对本发明实施例中基于Jmeter的分布式性能测试装置进行描述,请参阅图3,本发明实施例中基于Jmeter的分布式性能测试装置一个实施例包括:

[0075] 部署单元301,用于将预测版本的Jmeter镜像部署到分布式节点,该分布式节点包括一个主节点和至少一个从节点;

[0076] 控制单元302,当主节点接收到启动测试指令时,用于通过主节点向各个从节点发送各自对应的测试脚本;

[0077] 测试单元303,当各个从节点接收到各自对应的测试脚本时,用于通过各个从节点进行性能测试,得到各自对应的测试结果;

[0078] 发送单元304,用于通过各个从节点将各自对应的测试结果发送到主节点;

[0079] 统计单元305,用于通过主节点统计各自对应的测试结果,得到性能测试结果。

[0080] 本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,支持版本回滚,同时采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,并且相对于虚拟机,容器能快速部署。由于容器与底层设施和机器文件系统是解耦的,能在不同云和不同版本操作系统间进行迁移,部署过程简单化,提高测试效率。

[0081] 请参阅图4,本发明实施例中基于Jmeter的分布式性能测试装置的另一个实施例包括:

[0082] 部署单元301,用于将预测版本的Jmeter镜像部署到分布式节点,该分布式节点包

括一个主节点和至少一个从节点；

[0083] 控制单元302,当主节点接收到启动测试指令时,用于通过主节点向各个从节点发送各自对应的测试脚本；

[0084] 测试单元303,当各个从节点接收到各自对应的测试脚本时,用于通过各个从节点进行性能测试,得到各自对应的测试结果；

[0085] 发送单元304,用于通过各个从节点将各自对应的测试结果发送到主节点；

[0086] 统计单元305,用于通过主节点统计各自对应的测试结果,得到性能测试结果。

[0087] 可选的,部署单元301还可进一步包括：

[0088] 选择子单元3011,用于从镜像仓库中选择预测版本的Jmeter镜像；

[0089] 设置子单元3012,用于通过弹性计算服务ECS实例设置分布式节点,分布式节点包括一个主节点和至少一个从节点；

[0090] 第一判断子单元3013,用于判断主节点和各个从节点是否同时存在预测版本的Jmeter镜像；

[0091] 部署子单元3014,若主节点和各个从节点不同时存在预测版本的Jmeter镜像,则用于通过预置方式部署预测版本的Jmeter镜像,得到部署结果；

[0092] 确认子单元3015,若主节点和各个从节点同时存在预测版本的Jmeter镜像,则用于确定部署成功。

[0093] 可选的,部署子单元3014还可以具体用于：

[0094] 若主节点和各个从节点不同时存在预测版本的Jmeter镜像,则将预测版本的Jmeter镜像部署到ECS实例上,得到Jmeter应用；

[0095] 通过容器docker运行Jmeter应用；

[0096] 将各个从节点对应的配置信息添加到主节点的控制列表中,各个从节点对应的配置信息包括各自对应的IP地址和各自对应的端口；

[0097] 重启主节点的Jmeter应用,得到部署结果。

[0098] 可选的,部署单元301还可进一步包括：

[0099] 第二判断子单元3016,用于判断部署结果是否为目标值,该目标值用于指示成功部署预测版本的Jmeter镜像；

[0100] 第一处理子单元3017,若部署结果不为目标值,则用于重新部署；

[0101] 第二处理子单元3018,若部署结果为目标值,则用于确定部署成功。

[0102] 可选的,控制单元302还可以具体用于：

[0103] 将主节点设置为远程方法调用RMI的客户端；

[0104] 将各个从节点设置为RMI的服务器端；

[0105] 当主节点接收到启动测试指令时,通过主节点执行测试计划,该测试计划包括总测试数据和测试脚本；

[0106] 解析主节点的控制列表,得到各个从节点对应的配置信息；

[0107] 根据各个从节点对应的配置信息对总测试数据进行拆分,得到各个从节点对应的测试数据；

[0108] 根据各个从节点对应的测试数据替换测试脚本中的关键字,得到各自对应的测试脚本；

[0109] 通过主节点以RMI方式向各个从节点推送各自对应的测试脚本。

[0110] 可选的,测试单元303还可以具体用于:

[0111] 当各个从节点接收到各自对应的测试脚本时,通过各个从节点解析各自对应的测试脚本,得到各自对应的测试请求;

[0112] 通过各个从节点执行各自对应的测试请求,得到各自对应的测试结果。

[0113] 可选的,基于Jmeter的分布式性能测试装置还可以进一步包括:

[0114] 上传单元306,用于将预测版本的Jmeter镜像上传到镜像仓库;

[0115] 管理单元307,用于通过打标签方式对预测版本的Jmeter镜像进行版本管理。

[0116] 本发明实施例中,通过镜像仓库对Jmeter镜像进行版本控制,支持版本回滚,同时采用基于容器的分布式节点部署Jmeter镜像并进行性能测试,容器之间进程不会相互影响,能区分计算资源,并且相对于虚拟机,容器能快速部署。由于容器与底层设施和机器文件系统是解耦的,能在不同云和不同版本操作系统间进行迁移,部署过程简单化,提高测试效率。

[0117] 上面图3和图4从模块化功能实体的角度对本发明实施例中的基于Jmeter的分布式性能测试装置进行详细描述,下面从硬件处理的角度对本发明实施例中基于Jmeter的分布式性能测试设备进行详细描述。

[0118] 图5是本发明实施例提供的一种基于Jmeter的分布式性能测试设备的结构示意图,该基于Jmeter的分布式性能测试设备500可因配置或性能不同而产生比较大的差异,可以包括一个或一个以上处理器(central processing units,CPU) 501(例如,一个或一个以上处理器)和存储器509,一个或一个以上存储应用程序509或数据509的存储介质508(例如一个或一个以上海量存储设备)。其中,存储器509和存储介质508可以是短暂存储或持久存储。存储在存储介质508的程序可以包括一个或一个以上模块(图示没标出),每个模块可以包括对基于Jmeter的分布式性能测试中的一系列指令操作。更进一步地,处理器501可以设置为与存储介质508通信,在基于Jmeter的分布式性能测试设备500上执行存储介质508中的一系列指令操作。

[0119] 基于Jmeter的分布式性能测试设备500还可以包括一个或一个以上电源502,一个或一个以上有线或无线网络接口503,一个或一个以上输入输出接口504,和/或,一个或一个以上操作系统505,例如Windows Serve,Mac OS X,Unix,Linux,FreeBSD等等。本领域技术人员可以理解,图5示出的基于Jmeter的分布式性能测试设备结构并不构成对基于Jmeter的分布式性能测试设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0120] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0121] 在本发明所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0122] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0123] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0124] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(read-only memory, ROM)、随机存取存储器(random access memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0125] 以上所述,以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

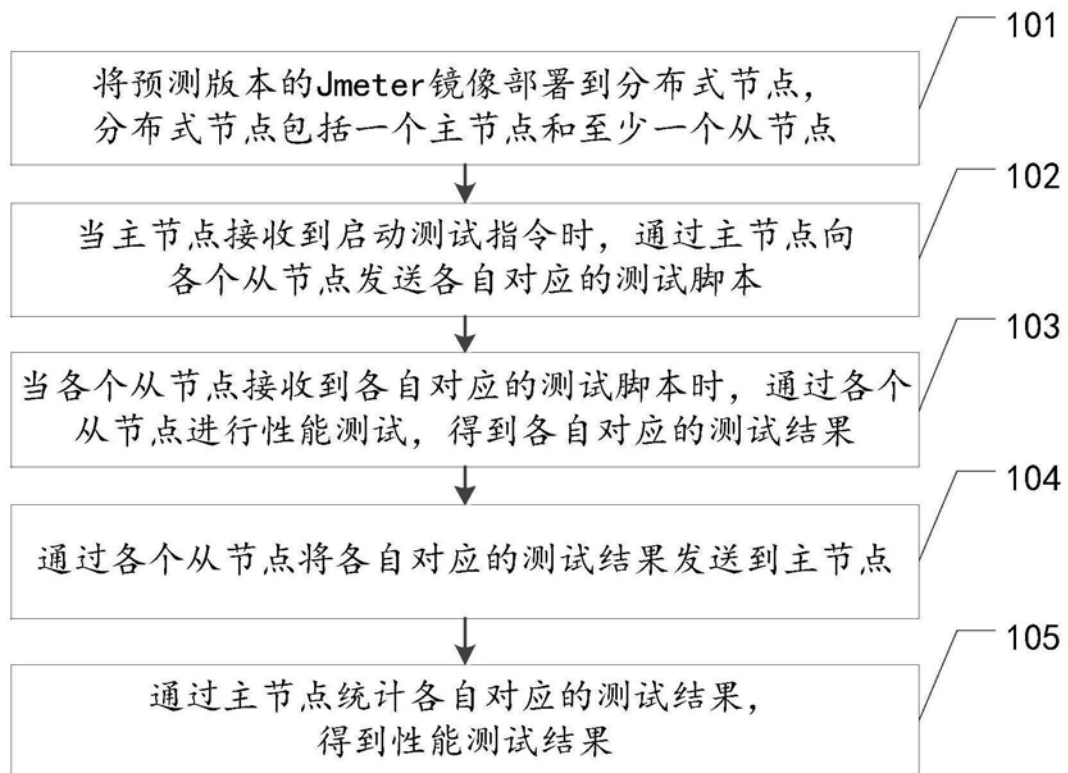


图1

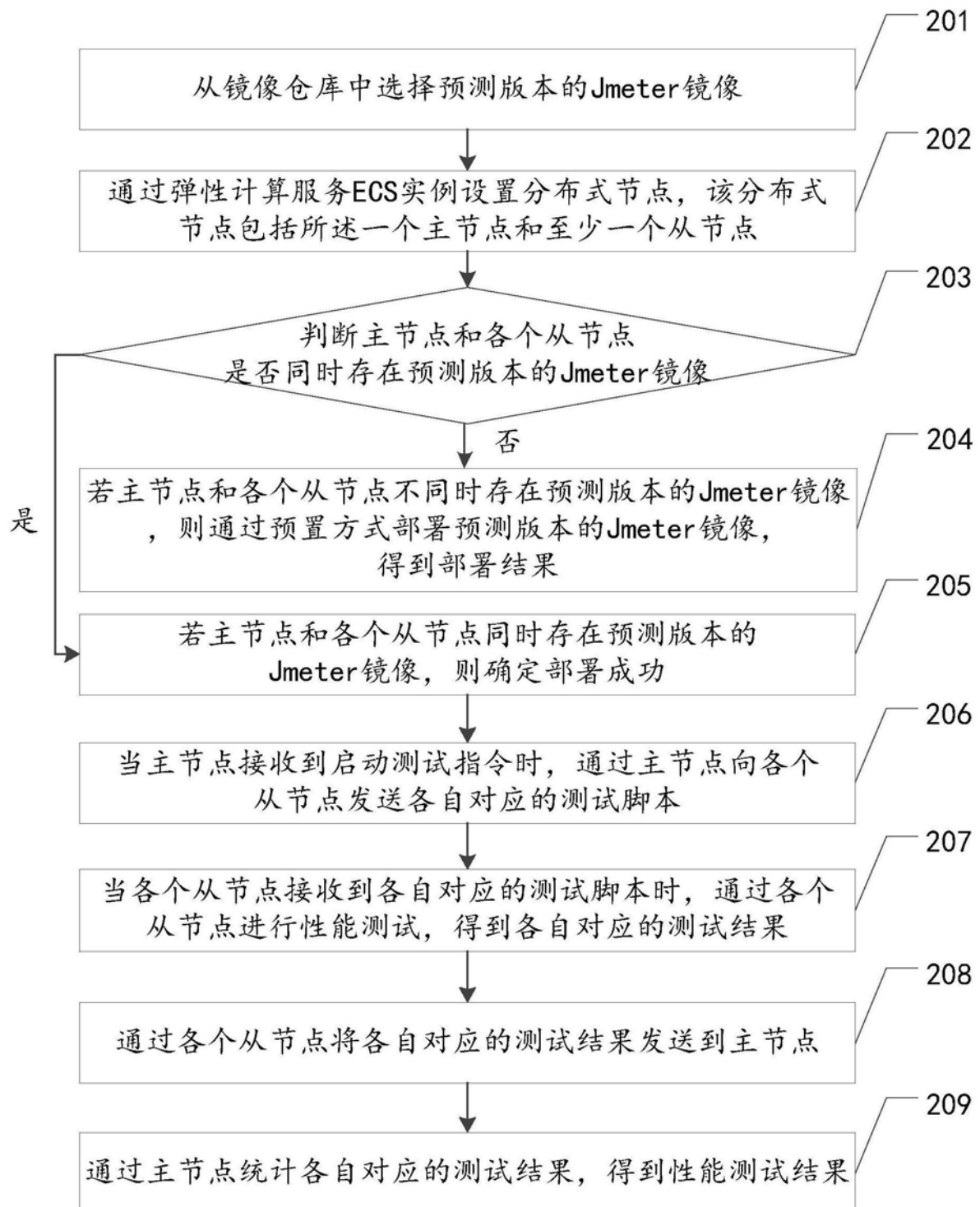


图2

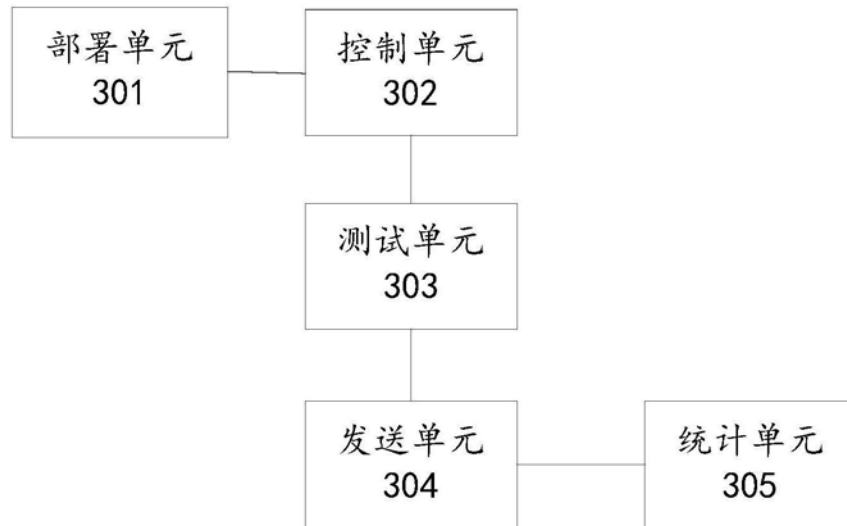


图3

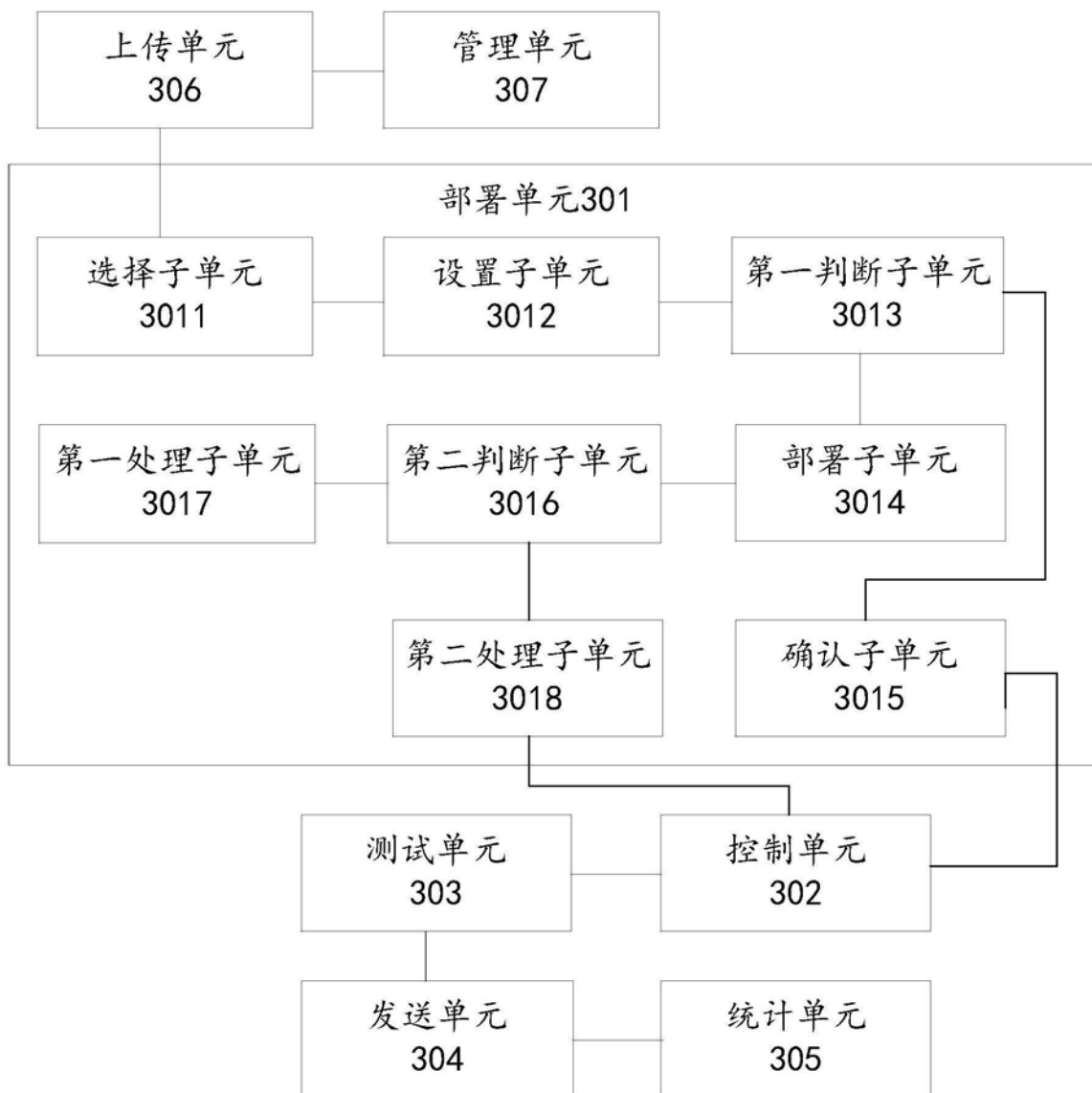


图4

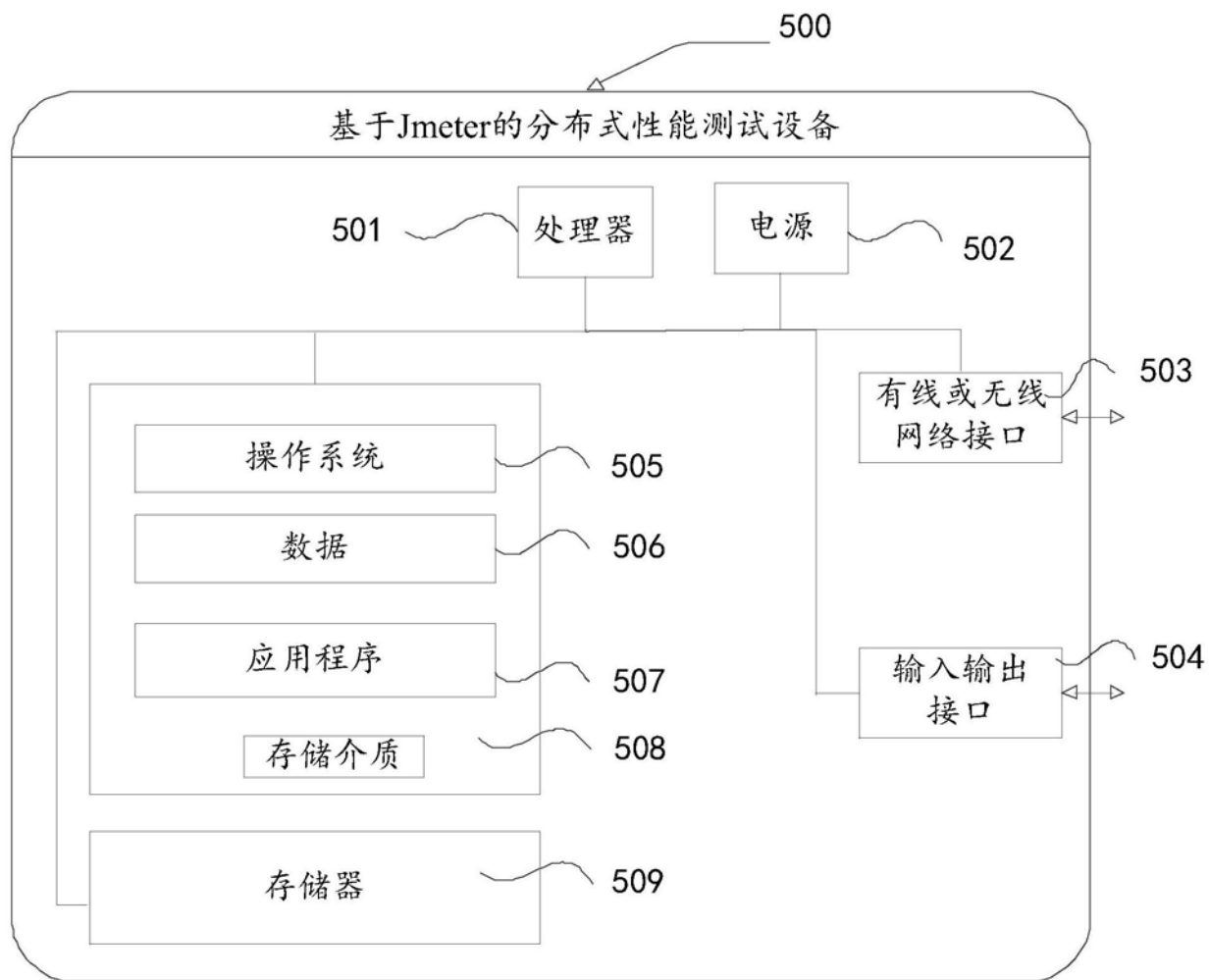


图5