



(12) 发明专利申请

(10) 申请公布号 CN 104866408 A

(43) 申请公布日 2015. 08. 26

(21) 申请号 201410058364. 5

(22) 申请日 2014. 02. 20

(71) 申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛资本大厦一座
四层 847 号邮箱

(72) 发明人 李恩领 童庭坚

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 任苏亚 王宝筠

(51) Int. Cl.

G06F 11/34(2006. 01)

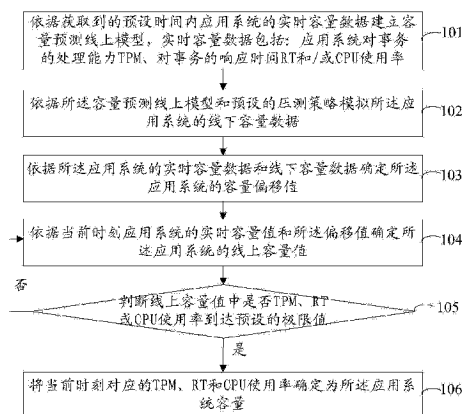
权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种应用系统的容量预测方法和装置

(57) 摘要

本申请实施例公开了一种应用系统的容量预测方法和装置,该方法包括:预处理过程和容量预测过程,其中,预处理过程包括:依据应用系统的实时容量数据建立容量预测线上模型,依据容量预测线上模型和预设的压测策略模拟线下容量数据;依据应用系统的实时容量数据和线下容量数据确定应用系统的容量偏移值;容量预测过程包括:依据当前时刻应用系统的实时容量值和偏移值确定应用系统的线上容量值;判断是否 TPM、RT 或 CPU 使用率到达预设的极限值,如果是,则将当前时刻对应的 TPM、RT 和 CPU 使用率确定为应用系统的容量。采用本申请实施例进行容量预测,比现有技术更为准确。



1. 一种应用系统的容量预测方法,其特征在于,包括:

预处理过程:

依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率;

依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据;

依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值;

容量预测过程:

依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值;

判断所述线上容量值中是否 TPM、RT 或 CPU 使用率达到预设的极限值,如果是,则将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

2. 根据权利要求 1 所述的方法,其特征在于,所述依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,包括:

获取所述应用系统的应用场景中各事务的比例关系;

在预设时间内,依据所述各事务的比例关系及采样周期确定所述应用系统的 RT 值,并采集所述应用系统的实时 TPM 和 CPU 使用率;

依据所述应用系统在预设时间内的 RT 值、实时 TPM 和 CPU 使用率生成所述应用系统的容量走势。

3. 根据权利要求 1 所述的方法,其特征在于,所述依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据,包括:

获取预设的压测策略,所述压测策略包括:压测时长、梯度运行时长和容量上限,所述压测时长表示压测的时间,所述梯度运行时长表示每一个梯度的压测时间,容量上限表示压测过程中的容量最大值;

依据所述容量预测线上模型和所述压测策略生成所述线下容量数据的走势;

依据所述线下容量数据的走势对所述应用系统进行压力测试,以得到线下容量数据的压测结果。

4. 根据权利要求 1 所述的方法,其特征在于,所述依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值,包括:

分别确定与所述实时容量数据对应的线上容量基准值,以及与线下容量数据对应的线下容量基准值;

比对所述应用系统的线上容量基准值和线下容量基准值从而获得应用系统的容量偏移值。

5. 根据权利要求 4 所述的方法,其特征在于,所述比对所述应用系统的线上容量基准值和线下容量基准值从而获得应用系统的容量偏移值,具体为:将所述应用系统的线上容量基准值和线下容量基准值的差作为所述应用系统的容量偏移值。

6. 一种应用系统的容量预测装置,其特征在于,包括预处理单元和容量预测单元,其中预处理单元包括:

模型建立模块,用于依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率;

压测模块,用于依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据;

偏移值确定模块,用于依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值;

所述容量预测单元包括:

第一确定模块,用于依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值;

判断模块,用于判断所述线上容量值中是否 TPM、RT 或 CPU 使用率到达预设的极限值;

第二确定模块,用于在所述判断模块的结果为是的情况下,将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

7. 根据权利要求 6 所述的装置,其特征在于,所述模型建立模块包括:

获取子模块,用于获取所述应用系统的应用场景中各事务的比例关系;

确定子模块,用于在预设时间内,依据所述各事务的比例关系及采样周期确定所述应用系统的 RT 值,并采集所述应用系统的实时 TPM 和 CPU 使用率;

生成子模块,用于依据所述应用系统在预设时间内的 RT 值、实时 TPM 和 CPU 使用率生成所述应用系统的容量走势。

8. 根据权利要求 6 所述的装置,其特征在于,所述压测模块包括:

获取子模块,用于获取预设的压测策略,所述压测策略包括:压测时长、梯度运行时长和容量上限,所述压测时长表示压测的时间,所述梯度运行时长表示每一个梯度的压测时间,容量上限表示压测过程中的容量最大值;

生成子模块,用于依据所述容量预测线上模型和所述压测策略生成所述线下容量数据的走势;

测试子模块,用于依据所述线下容量数据的走势对所述应用系统进行压力测试,以得到线下容量数据的压测结果。

9. 根据权利要求 6 所述的装置,其特征在于,所述容量偏移值确定模块包括:

确定子模块,用于分别确定与所述实时容量数据对应的线上容量基准值,以及与线下容量数据对应的线下容量基准值;

比对子模块,用于比对所述应用系统的线上容量基准值和线下容量基准值从而获得应用系统的容量偏移值。

10. 根据权利要求 9 所述的装置,其特征在于,所述比对子模块具体用于:将所述应用系统的线上容量基准值和线下容量基准值的差作为所述应用系统的容量偏移值。

一种应用系统的容量预测方法和装置

技术领域

[0001] 本申请涉及互联网数据处理领域,特别是涉及一种应用系统的容量预测方法和装置。

背景技术

[0002] 目前,互联网不管是常规的性能测试,还是应用系统的容量评估,一般需要先通过生产运维提供的业务场景参考(大多数是单一场景)比例,凭借开发、业务、测试人员的经验进行分析,得出经验性测试模型,再通过性能测试工具在测试环境中模拟压测,最后分析测试结果来判断应用系统的最大负载能力。

[0003] 但是发明人在研究过程中发现,这种传统的实施方法中,生产运维提供的业务场景参考数据是否准确,其代表的业务场景是否全面都是未知的,因此,现有技术中进行应用系统容量预测的时候准确性较差。

发明内容

[0004] 本申请所要解决的技术问题是,提供一种应用系统的容量预测方法,以解决现有技术中预测应用系统的容量时准确性较差的技术问题。

[0005] 本申请的另一个目的是将上述构思应用于具体的应用环境中,提供一种应用系统的容量预测装置,从而保证该方法的实现和应用。

[0006] 为解决上述技术问题,本申请实施例提供了一种应用系统的容量预测方法,该方法包括:

[0007] 预处理过程:

[0008] 依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率;

[0009] 依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据;

[0010] 依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值;

[0011] 容量预测过程:

[0012] 依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值;

[0013] 判断所述线上容量值中是否 TPM、RT 或 CPU 使用率到达预设的极限值,如果是,则将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

[0014] 本申请实施例还提供了一种应用系统的容量预测装置,包括预处理单元和容量预测单元,其中预处理单元包括:

[0015] 模型建立模块,用于依据获取到的预设时间内应用系统的实时容量数据建立容量

预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率;

[0016] 压测模块,用于依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据;

[0017] 偏移值确定模块,用于依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值;

[0018] 所述容量预测单元包括:

[0019] 第一确定模块,用于依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值;

[0020] 判断模块,用于判断所述线上容量值中是否 TPM、RT 或 CPU 使用率到达预设的极限值;

[0021] 第二确定模块,用于在所述判断模块的结果为是的情况下,将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

[0022] 从上述的技术方案可以看出,在本申请实施例中,采用应用系统中的实时变化的容量数据进行流量建模和容量压测,再通过对实时容量数据的比对分析进行容量预测,从而通过建立生产环境与测试环境的关联关系,来得出应用系统在两者间的性能对比差异,而依靠这种对比关系实现容量预测就因为考虑了容量偏移而更准确。因此,解决了现有技术中存在的业务场景的参考数据可能不够准确,其代表的业务场景可能不够全面等问题,最终实现的容量预测也比现有技术更准确。

附图说明

[0023] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请中记载的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0024] 图 1 为本申请方法实施例的流程图;

[0025] 图 2 为本申请方法实施例中步骤 101 的流程图;

[0026] 图 3 为本申请方法实施例中被测的应用系统的事务流量走势的示意图;

[0027] 图 4 为本申请方法实施例中步骤 102 的流程图;

[0028] 图 5 为本申请方法实施例中步骤 103 的流程图;

[0029] 图 6 为本申请方法实施例中容量偏移量的示意图;

[0030] 图 7 为本申请云装置实施例的结构示意图。

具体实施方式

[0031] 为了使本技术领域的人员更好地理解本申请方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0032] 参见图 1,示出了本申请的一种应用系统的容量预测方法实施例的流程图,其中,

步骤 101 ~ 步骤 103 为预处理过程,而步骤 104 和步骤 105 为容量预测过程。本实施例可以包括以下步骤:

[0033] 步骤 101 :依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率。

[0034] 在本申请实施例中,需要采集到应用系统某一个时段内各业务场景的实时容量数据。其中,应用系统是指在线实时服务于用户的某种事务(业务或请求)的服务器,例如:Linux 或者 Windows,等等。某个时段是指应用系统提供服务的一个持续的时间段,一般可以选择 2 个小时。当然,也可以选择其他时间。业务场景是指应用系统对用户或其他系统提供的某种服务,也可以理解为对请求的应答。不同的应用系统对外提供的服务不同业务场景也就不同。例如在互联网第三方支付中的收银台系统中提供了几种不同的事务类型,收银台系统就可以称之为业务场景,例如,其中,页面展现的事务类型(用户可以看到自己的余额,和不同的支付类型),以及查询余额、账户管理、信用支付、余额宝支、余额支付或者网银支付的事务类型,等等。具体在采集数据的时候,是把在线的应用系统在 2 个小时内的每一分钟内是如何处理业务场景和性能表现数据进行收集存储的过程。

[0035] 具体的,参考图 2 所示,为步骤 101 的一个具体实施方式,本实施方式具体可以包括:

[0036] 步骤 201 :获取所述应用系统的应用场景中各事务的比例关系。

[0037] 根据业务场景中各事务比例得出容量预测线上模型。

[0038] 在本申请实施例中,最终建立的容量预测线上模型有三个维度:时间、事务比例和应用系统的性能指标,如表 1 所示。那么,就需要获取到再每一分钟内,各个业务场景内的各个事务的比例,例如对于第一分钟来讲,余额支付、网银支付和信用支付的比例为:500、300、200。

[0039] 表 1

[0040]

时间	第一分钟	第二分钟	第三分钟	第 120 分钟
业务场景	余额支付: 500	余额支付: 800	余额支付: 1500
	网银支付: 300	网银支付: 700	网银支付: 500	
	信用支付: 200	信用支付: 500	信用支付: 300	
			余额宝支付: 700	
性能	TPM:10000	TPM:20000	TPM:30000
	CPU:30%	CPU:40%	CPU:50%	
	RT{余额支付: 100ms,	RT{余额支付: 110ms,	RT{余额支付: 120ms,	

[0041]

指标	网银支付: 200ms,	网银支付: 225ms,	网银支付: 255ms,	
	信用支付: 150ms}	信用支付: 160ms}	信用支付: 170ms,	
			余额宝支付:: 120ms}	

[0042] 在表 1 所示的容量预测线上模型中,容量预测线上模型可以是一个多维的数组,

其中,CPU 代表应用系统的资源使用率,TPM (Transaction Per Minute)是指应用系统对事务的处理能力,RT 表示应用系统处理某一个事务的响应时间。

[0043] 步骤 202 :在预设时间内,依据所述各事务的比例关系及采样周期确定所述应用系统的 RT 值,并采集所述应用系统的实时 TPM 和 CPU 使用率。

[0044] 在表 1 所示的容量预测线上模型中,采样周期就是 1 分钟。当然,采样周期也可以依据实际需求采用其他的值,例如半分钟,或者两分钟等。在本步骤中,即是在预设的两个小时的采集时间内,应用系统对余额支付事务的响应时间为 100ms,而对网银支付事务的响应时间为 200ms,对信用支付事务的响应时间为 150ms。

[0045] 步骤 203 :依据所述应用系统在预设时间内的 RT 值、实时 TPM 和 CPU 使用率生成所述应用系统的容量走势。

[0046] 在本实施例中,实现本步骤可以通过将表 1 中的数据生成一副容量走势图的方式。参考图 3 所示,为被测的应用系统的事务流量走势的示意图,其中,X 轴代表时间,Y 轴代表事务量(在图 3 中以 TPM 值为例),图 3 中的每一个曲线代表一个业务或者一个请求。

[0047] 在步骤 101 建立了容量预测线上模型之后,则后续可以利用容量预测线上模型和预设的压测策略来进行线下压力测试,进入步骤 102 :依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据。

[0048] 在本步骤实现压力测试的时候,可以建立一个压测模型。具体的,根据步骤 101 建立的容量预测线上模型,开发人员可以使用模拟压测工具编写压测脚本,并将所编写出的压测脚本和容量预测线上模型之间建立映射关系。例如 :工具中的 eBankPay () 对应于容量预测线上模型中的网银支付事务。

[0049] 具体的,参考图 4 所示所示,为步骤 102 的一个具体实施方式,本实施方式具体可以包括 :

[0050] 步骤 401 :获取预设的压测策略,所述压测策略包括 :压测时长、梯度运行时长和容量上限,所述压测时长表示压测的时间,所述梯度运行时长表示每一个梯度的压测时间,容量上限表示压测过程中的容量最大值。

[0051] 在实施本申请时,需要开发人员预先设置线下模拟压测的压测策略。其中,压测策略主要包括 :压测时长,梯度运行时长和容量上限这三个参数。其中,压测时长表示压测的时间,例如,压力测试的时间为 5 个小时 ;而梯度运行时长表示每一个梯度的压测时间,例如,一个梯度的运行时长为 20 分钟 ;容量上限表示压测过程中的容量最大值,例如,CPU 的使用率不超过 75%。

[0052] 步骤 402 :依据所述容量预测线上模型和所述压测策略生成所述线下容量数据的走势。

[0053] 在建立压测模型之后,具体需要回放压测模型下创建的压力,以获得线下容量数据的走势。因为根据容量预测线上模型的数据可以计算出生成系统在压测时长内的线下容量数据的走势(即是压测模型),但是该线下容量数据的走势仅仅是根据容量预测线上模型的数据推理出来的,因此需要进行实际中应用系统的压力测试。

[0054] 步骤 403 :依据所述线下容量数据的走势对所述应用系统进行压力测试,以得到线下容量数据的压测结果。

[0055] 得出压测模型之后,再根据压测策略可以计算出应用系统的线下容量数据的走

势,那么后续就可以装载容量预测线上模型和线下容量数据的走势进行压力测试。其中,线下容量数据的走势中有两个参数,一个是时间趋势,一个是事务量,然后即可触发应用系统的压力测试。具体的,在实施压力测试的过程,流量控制器(Alipay Throughput Replayer,也称流量回放器)在压测工具上的实现如下:

[0056] 获取当前秒向应用系统已发送的事务量;

[0057] 如果已发送事务量小于预期事务量(该预期事务量的计算方式为:从压测容量趋势中获得的当前时间所对应的事务量),则计算延时,等待延时结束后再发送事务请求;其中,延时的算法为:当前秒剩余毫秒数(当前秒(1000 毫秒) - 当前秒已过毫秒数)/ 当前秒未完成事务量(当前秒预期事务量 - 当前秒已完成事务量),如果计算结果小于或等于 1,则不延时,否则等待相应时长(延时的计算结果)后再发起请求。例如,当前秒预期完成事务量为 100,假设,当前秒刚刚开始(也就是当前秒为 0 毫秒的时候),延时 = (1000 毫秒 - 0 毫秒) / (预期事务量 100 - 已完成事务量 0) = 10 毫秒,那么,需要延时 10 毫秒后再向应用系统发送事务的请求。

[0058] 最终得到的容量测试结果,也是和表 1 所示的容量预测线上模型的结果对应的一个数组,参考表 2 所示。

[0059] 表 2

[0060]

时间	第一分钟	第二分钟	第三分钟	第 120 分钟
业务场景	余额支付: 500	余额支付: 800	余额支付: 1500
	网银支付: 300	网银支付: 700	网银支付: 500	
	信用支付: 200	信用支付: 500	信用支付: 300	
			余额宝支付: 700	
性能指标	TPM:10000	TPM:20000	TPM:30000
	CPU:31%	CPU:42%	CPU:53%	
	RT{余额支付: 105ms,	RT{余额支付: 115ms,	RT{余额支付: 120ms,	
	网银支付: 210ms,	网银支付: 235ms,	网银支付: 265ms,	
	信用支付: 155ms}	信用支付: 175ms}	信用支付: 180ms,	
			余额宝支付: 125ms}	

[0061] 在进行压力测试之后得到了和容量预测线上模型类似的线下容量数据,进入步骤 103:依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值。

[0062] 因为应用系统的线下容量数据是进行压测得出来的,和在建立容量预测线上模型的时候采集的应用系统的实时容量数据是存在区别的,而这个区别就是应用系统的容量偏移值。在进行应用系统的容量预测的时候,需要将容量偏移值也考虑在内。本领域技术人员可以通过已经获取的容量预测线上模型的实时容量数据和线下压测结果的值进行比对,通过比对得出容量偏移值,其中,容量比对可以包括系统 TPM、CPU 使用率和 RT,当然也可以比对其他容量参数。

[0063] 具体的,在确定容量偏移值的时候,可以参考图 5 所示的流程图,可以包括:

[0064] 步骤 501:分别确定与所述实时容量数据对应的线上容量基准值,以及与线下容量数据对应的线下容量基准值。

[0065] 下参考图 6 所示,图 6 为比对 CPU 使用率的界面示意图。在图 6 中,共有两条直线,其中斜率大的 1 号直线是线上 CPU 使用率基准线,该线上 CPU 使用率基准线上对应的 Y 坐标即是线上容量基准值。斜率小的 2 号直线是线下 CPU 使用率基准线,该线下 CPU 使用率基准线对应的 Y 坐标即是线下容量基准值。其中,3 号曲线表示线上 CPU 使用率,而 4 号曲线表示线上 CPU 使用率预测,5 号曲线表示的是线下 CPU 使用率。其中,1 号直线是根据 3 号曲线确定的,而 2 号直线是根据 5 号曲线确定的。

[0066] 步骤 502:比对所述应用系统的线上容量基准值和线下容量基准值从而获得应用系统的容量偏移值。

[0067] 在本步骤中,即是计算图 6 所示的两条直线所表示的线上和线下的 CPU 使用率的差值,即是线上 CPU 使用率和线下 CPU 使用率的斜率差。根据标准斜率 m 的公式

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

即可获得线上和线下两条 CPU 使用率的斜率差。

[0068] 具体的,本步骤在实现是可以将所述应用系统的线上容量基准值和线下容量基准值的差作为所述应用系统的容量偏移值。例如,如果想获得 x 轴上的任意一个点的 y 值,通过上面得到的斜率可以得出线上的 CPU 使用率为 $y_{\text{线上}} = m_{\text{线上}}x + b_{\text{线上}}$,线下的 CPU 使用率为 $y_{\text{线下}} = m_{\text{线下}}x + b_{\text{线下}}$ (b 是常量),因此,线上和线下的 CPU 使用率在某一点的差异值是 $y_{\text{差}} = y_{\text{线上}} - y_{\text{线下}}$,即是: $y_{\text{差}} = (m_{\text{线上}} - m_{\text{线下}})x + b_{\text{线上}} - b_{\text{线下}}$ 。

[0069] 步骤 104:依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值。

[0070] 基于步骤 103 得到的容量偏移值,再依据线下的压测结果可以预测出应用系统在线的容量,具体的,根据之前得出的容量偏移值和 x 、 y 坐标,预测的容量值计算方式为: $y_x = y_{\text{线下}} + y_{\text{差}}$ 。

[0071] 需要说明的是,在本申请实施例中,进行容量预测需要考虑应用系统的极限值的问题。例如,某个应用系统的 CPU 使用率上限为 75%,处理事务的响应时间不超过 150 毫秒,等等,那么在进行容量预测的时候就需要考虑这些因素,预测的容量值不能超过预先设定的应用系统的极限值。此外,在本申请实施例中,还有影响应用系统的健康度的其他指标,例如,错误率,是否内存有泄露,虽然在本申请实施例中只给出了 TPM、RT 或者 CPU 使用率的方案,但是本领域技术人员可以依据本申请实施例的方式在需要的时候也可以进行这些指标的预测。

[0072] 步骤 105:判断所述线上容量值中是否 TPM、RT 或 CPU 使用率到达预设的极限值,如果是,则进入步骤 106,如果否,则返回步骤 104。

[0073] 在进行压测过程中,实时判断应用系统的 TPM、RT 或 CPU 使用率是否达到了预设的极限值。判断应用系统是否达到了最大容量,就是在应用系统还处于健康状态情况下,无论是 CPU 使用率、TPM 或者 RT 中的某一项达到了预设的极限值。在该时刻对应的应用系统的吞吐量就是应用系统的最大容量,例如响应时间 120 毫秒的时候,CPU 的使用率达到了极限值 75%,或,CPU 的使用率为 50% 的时候响应时间达到了极限值 150 毫秒,那么此刻的应用系统就是达到了最大容量。在应用系统的某一项性能指标,TPM、RT 或 CPU 使用率,达到了极限值的时候,就将当前时刻对应的其他两项的性能指标值确定为预测的应用系统的容量值。如果当前未达到极限值,则继续进行压力测试即可。在本申请实施例中,可以返回步骤

104 重新计算应用系统当前时刻的线上容量值。

[0074] 可以理解的是,在实际的应用中,为了最大程度的优化应用系统的吞吐量,开发人员可以采取用牺牲应用系统的 RT 换取 CPU 使用率的相抵,从而使应用系统更加稳定,或者为了获取最大的吞吐量而增加系统的 CPU 使用率。本领域技术人员可以通过设置不同对应关系的 CPU 使用率、RT 和 TPM 的极限值来实现上述策略。

[0075] 步骤 106 :将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

[0076] 可以理解的是,当某一项的性能指标达到极限值的时候,则当前时刻对应的其他两项的性能指标就是应用系统的容量,因为应用系统不可能再承受再大的压力了。例如,某个应用系统预设的 CPU 的使用率的极限值是 75%, TPM 是 50000, RT 是 100ms,而某个时刻 CPU 使用率达到了 75%,此时刻对应的 TPM 是 45000, RT 是 90ms,那么该应用系统的容量即为 :CPU 使用率为 75%, TPM 是 45000, RT 是 90ms。

[0077] 在本申请实施例中,采用应用系统中的实时变化的容量数据进行流量建模和容量压测,再通过对实时容量数据的比对分析进行容量预测,从而通过建立生产环境与测试环境的关联关系,来得出应用系统在两者间的性能对比差异,而依靠这种对比关系实现容量预测就因为考虑了容量偏移而更准确。因此,解决了现有技术中存在的业务场景的参考数据可能不够准确,其代表的业务场景可能不够全面等问题,最终实现的容量预测也比现有技术更准确。

[0078] 通过以上的方法实施例的描述,所属领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的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)执行本申请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:只读存储器(ROM)、随机存取存储器(RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0079] 相应于上面的方法实施例,本申请实施例还提供一种应用系统的容量预测装置。参见图 7,示出了一种基于应用系统的容量预测实施例的结构示意,该装置可以包括:预处理单元 70 和容量预测单元 71,其中预处理单元可以包括:

[0080] 模型建立模块 701,用于依据获取到的预设时间内应用系统的实时容量数据建立容量预测线上模型,所述实时容量数据包括:应用系统对事务的处理能力 TPM、应用系统对事务的响应时间 RT 和 / 或 CPU 使用率;

[0081] 可选的,所述模型建立模块 701 具体可以包括:

[0082] 获取子模块,用于获取所述应用系统的应用场景中各事务的比例关系;

[0083] 确定子模块,用于在预设时间内,依据所述各事务的比例关系及采样周期确定所述应用系统的 RT 值,并采集所述应用系统的实时 TPM 和 CPU 使用率;

[0084] 生成子模块,用于依据所述应用系统在预设时间内的 RT 值、实时 TPM 和 CPU 使用率生成所述应用系统的容量走势。

[0085] 压测模块 702,用于依据所述容量预测线上模型和预设的压测策略模拟所述应用系统的线下容量数据;

[0086] 可选的,所述压测模块 702 具体可以包括:

[0087] 获取子模块,用于获取预设的压测策略,所述压测策略包括:压测时长、梯度运行时长和容量上限,所述压测时长表示压测的时间,所述梯度运行时长表示每一个梯度的压测时间,容量上限表示压测过程中的容量最大值;

[0088] 生成子模块,用于依据所述容量预测线上模型和所述压测策略生成所述线下容量数据的走势;

[0089] 测试子模块,用于依据所述线下容量数据的走势对所述应用系统进行压力测试,以得到线下容量数据的压测结果。

[0090] 容量偏移值确定模块 703,用于依据所述应用系统的实时容量数据和线下容量数据确定所述应用系统的容量偏移值;

[0091] 可选的,所述容量偏移值确定模块 703 具体可以包括:

[0092] 确定子模块,用于分别确定与所述实时容量数据对应的线上容量基准值,以及与线下容量数据对应的线下容量基准值;

[0093] 比对子模块,用于比对所述应用系统的线上容量基准值和线下容量基准值从而获得应用系统的容量偏移值。

[0094] 所述比对子模块具体用于:将所述应用系统的线上容量基准值和线下容量基准值的差作为所述应用系统的容量偏移值。

[0095] 其中,所述容量预测单元 71 可以包括:

[0096] 第一确定模块 711,用于依据当前时刻应用系统的实时容量值和所述偏移值确定所述应用系统的线上容量值;

[0097] 判断模块 712,用于判断所述线上容量值中是否 TPM、RT 或 CPU 使用率到达预设的极限值;

[0098] 第二确定模块 713,用于在所述判断模块的结果为是的情况下,将当前时刻对应的 TPM、RT 和 CPU 使用率确定为所述应用系统容量。

[0099] 在本实施例中,采用应用系统中的实时变化的容量数据进行流量建模和容量压测,再通过对实时容量数据的比对分析进行容量预测,从而通过建立生产环境与测试环境的关联关系,来得出应用系统在两者间的性能对比差异,而依靠这种对比关系实现容量预测就因为考虑了容量偏移而更准确。因此,解决了现有技术中存在的业务场景的参考数据可能不够准确,其代表的业务场景可能不够全面等问题,最终实现的容量预测也比现有技术更准确。

[0100] 可以理解的是,本申请可用于众多通用或专用的计算系统环境或配置中。例如:个人计算机、服务器计算机、手持设备或便携式设备、平板型设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络 PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境等等。

[0101] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0102] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实

体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0103] 对于装置实施例而言,由于其基本对应于方法实施例,所以相关之处参见方法实施例的部分说明即可。以上所描述的云平台 and 系统实施例仅仅是示意性的,其中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模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0104] 以上所述仅是本申请的具体实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本申请的保护范围。

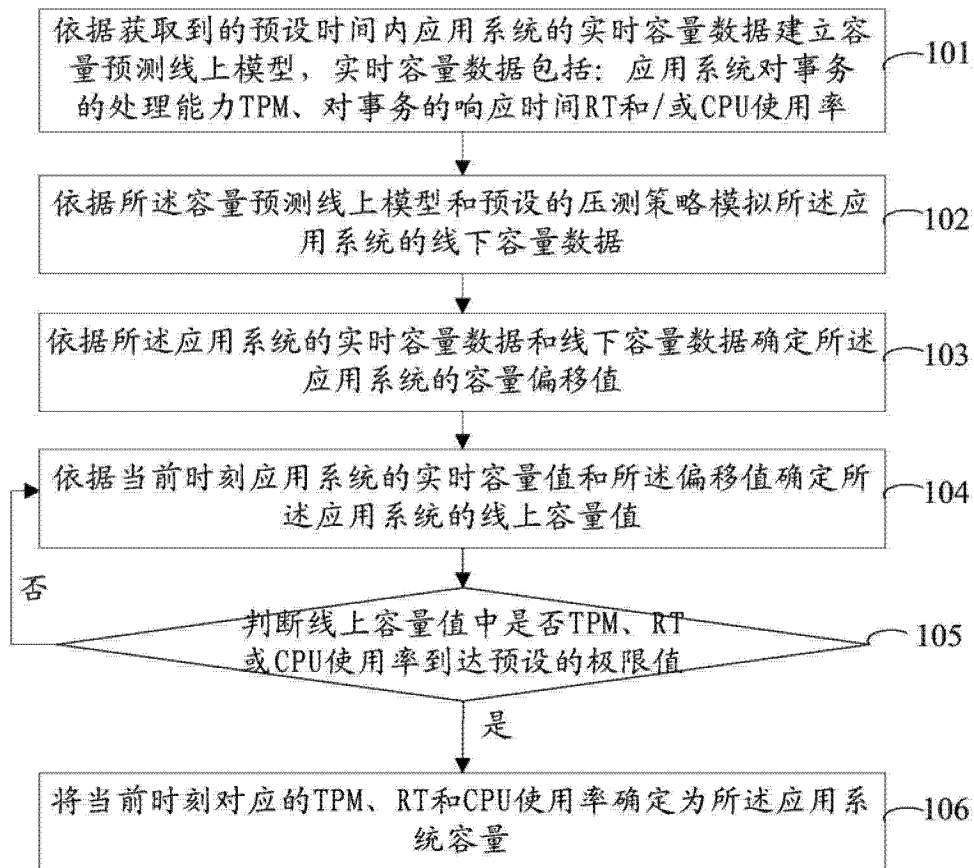


图 1

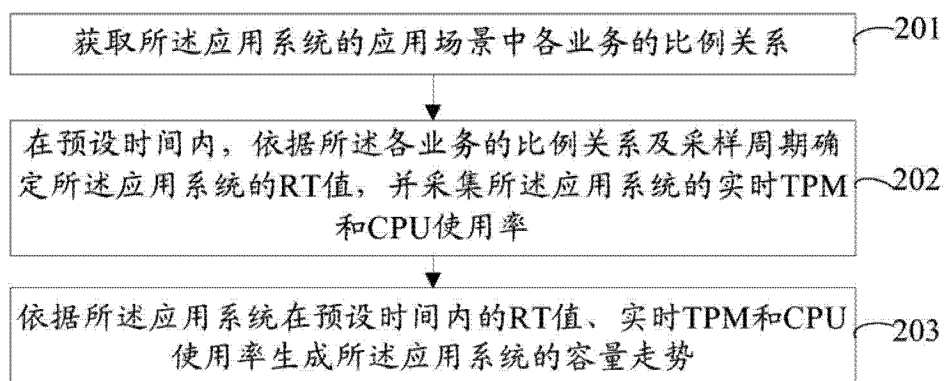


图 2

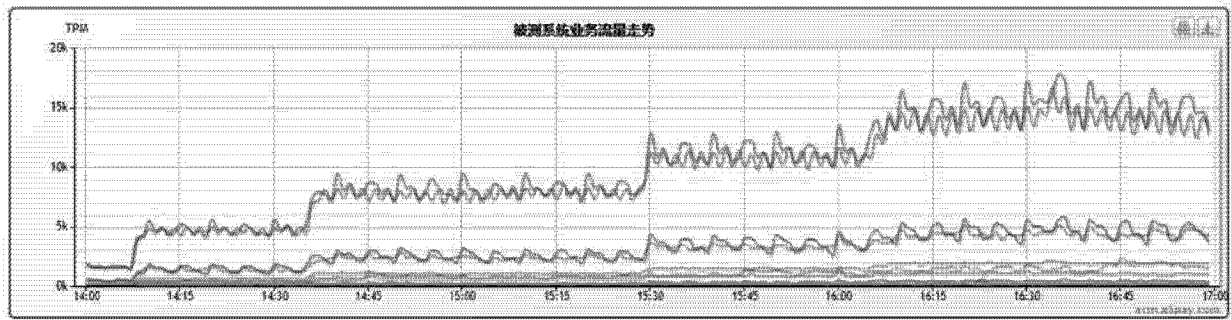


图 3

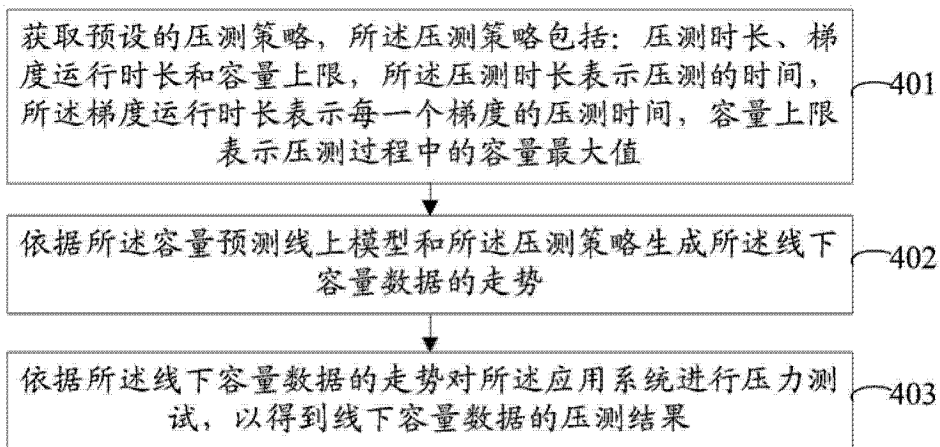


图 4

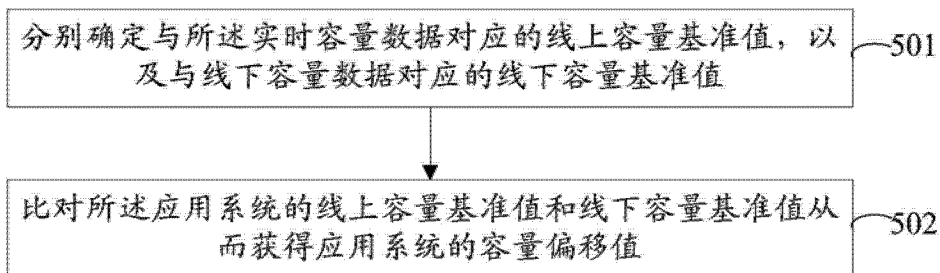


图 5

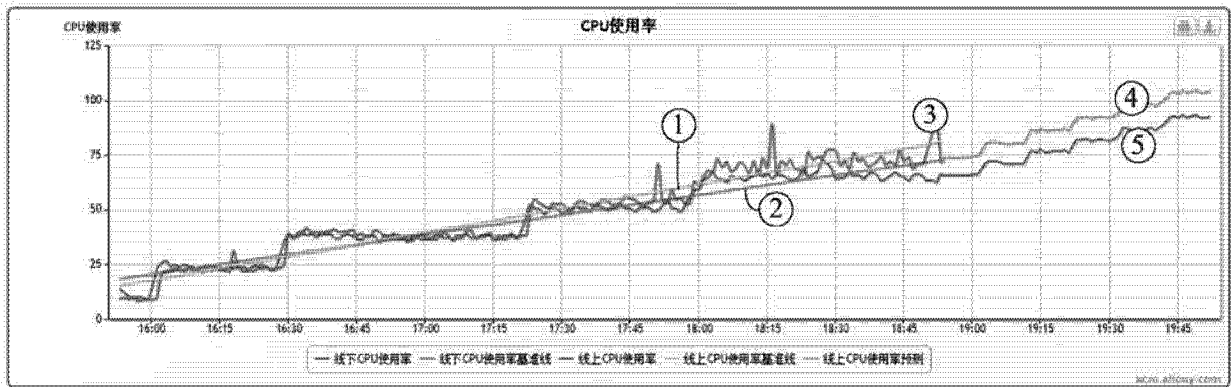


图 6

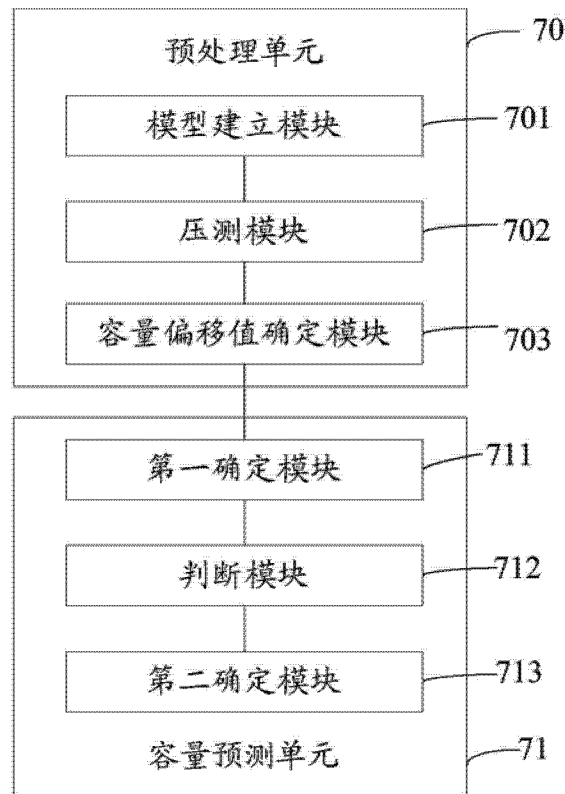


图 7