(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2016 年 1 月 21 日 (21.01.2016)

WIPO | PCT

(10) 国际公布号
WO 2016/008398 A1

(51) 国际专利分类号:
G06F 11/36 (2006.01)

(21) 国际申请号: PCT/CN2015/083886

(22) 国际申请日: 2015 年 7 月 13 日 (13.07.2015)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201410338809.5 2014 年 7 月 16 日 (16.07.2014) CN

(71) 申请人: 北京奇虎科技有限公司 (BEIJING QIHOO TECHNOLOGY COMPANY LIMITED) [CN/CN]; 中国北京市西城区新街口外大街 28 号 D 座 112 室（德胜园区）, Beijing 100088 (CN)。 奇智软件（北京）有限公司 (QIZHI SOFTWARE (BEIJING) COMPANY LIMITED) [CN/CN]; 中国北京市朝阳区酒仙桥路 6 号院 2 号楼 B 座 2 层、3 层 301-306 室, Beijing 100015 (CN)。

(72) 发明人: 郭振华 (GUO, Zhenhua); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。

(74) 代理人: 北京润泽恒知识产权代理有限公司 (BEIJING RISEHIGH INTELLECTUAL PROPERTY LAW FIRM); 中国北京市海淀区中关村南大街 31 号神舟大厦 702, Beijing 100081 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

[见续页]

(54) Title: PROGRAM PERFORMANCE TEST METHOD AND DEVICE

(54) 发明名称 : 程序性能测试方法和装置

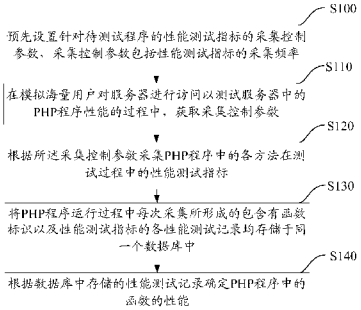


图 1 / Fig. 1

S100 Presetting a collection control parameter of a performance test index for a program to be tested, wherein the collection control parameter comprises a collection frequency of the performance test index
S110 In the process of simulating a large number of users accessing a server so as to test the performance of the PHP program in the server, acquiring the collection control parameter
S120 According to the collection control parameter, collecting a performance test index of each function in the PHP program in the test process
S130 Storing each performance test record formed by each collection during the operation of the PHP program and containing a function identifier and the performance test index in the same database
S140 According to the performance test record stored in the database, determining the performance of the function in the PHP program

(57) Abstract: A program performance test method and device, which are used for testing the performance of an Xhprof-based PHP program. The method comprises: presetting a collection control parameter of a performance test index for a PHP program, wherein the collection control parameter at least comprises a collection frequency of the performance test index (S100); in the process of simulating a large number of users accessing a server so as to test the performance of the PHP program in the server, acquiring the collection control parameter (S110); according to the collection control parameter, collecting a performance test index of each function in the PHP program in the test process (S120); storing each performance test record formed by each collection during the operation of the PHP program and containing a function identifier and the performance test index in the same database (S130); and according to the performance test record stored in the database, determining the performance of the function in the PHP program (S140).

(57) 摘要: 一种程序性能测试方法和装置，用于基于 Xhprof 的 PHP 程序性能测试，其中的方法包括：预先设置针对 PHP 程序的性能测试指标的采集控制参数，且所述采集控制参数至少包括：性能测试指标的采集频率（S100）；在模拟海量用户对服务器进行访问以测试服务器中的 PHP 程序性能的过程中，获取所述采集控制参数（S110）；根据所述采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标（S120）；将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中（S130）；根据所述数据库中存储的性能测试记录确定 PHP 程序中的函数的性能（S140）。

WO 2016/008398 A1

本国际公布:

— 包括国际检索报告(条约第 21 条(3))。

程序性能测试方法和装置

技术领域

本发明涉及性能测试技术，特别是涉及一种程序性能测试方法以及相应的程序性能测试装置，用于基于 Xhprof 的 PHP 程序的性能测试。

背景技术

PHP（Hypertext Preprocessor，超文本预处理器）是一种通用的开源脚本语言。目前，PHP 主要适用于Web开发领域。

PHP 程序在开发过程中通常需要进行性能测试。目前，通常会使用 XHProf 工具对 PHP 程序进行性能测试。

发明人在实现本发明过程中发现，XHProf 工具目前提供的 PHP 程序性能测试方式无论是在 PHP 程序性能测试的准确性方面，还是在 PHP 程序性能测试的易用性方面，亦或是在 PHP 程序性能测试的多样性方面，都存在进一步提升的空间。

发明内容

鉴于上述问题，提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的程序性能测试方法和装置。

依据本发明的一个方面，提供了一种程序性能测试方法，用于基于 Xhprof 的 PHP 程序性能测试，包括：预先设置有针对 PHP 程序的性能测试指标的采集控制参数，且所述采集控制参数至少包括：性能测试指标的采集频率；在模拟海量用户对服务器进行访问以测试服务器中的 PHP 程序性能的过程中，获取所述采集控制参数；根据所述采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标；将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中；根据所述数据库中存储的性能测试记录确定 PHP 程序中的函数的性能。

根据本发明的另一方面，提供了一种计算机程序，包括计算机可读代码，当所述计算机可读代码在计算机上运行时，将执行前述的程序性能测试方法。

根据本发明的再一方面，提供了一种计算机可读介质，其中存储了前述的计算机程序。

根据本发明的又一方面，提供了一种程序性能测试装置，用于基于Xhprof的PHP程序性能测试，包括：存储模块，配置为预先设置有针对PHP程序的性能测试指标的采集控制参数，且所述采集控制参数至少包括：性能测试指标的采集频率；获取模块，配置为在模拟海量用户对服务器进行访问以测试服务器中的PHP程序性能的过程中，获取所述采集控制参数；采集模块，配置为根据所述采集控制参数采集PHP程序中的各函数在测试过程中的性能测试指标；记录模块，配置为将PHP程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中；性能模块，配置为根据所述数据库中存储的性能测试记录确定PHP程序中的函数的性能。

本发明的程序性能测试方法以及装置通过预先设置包含有性能测试指标的采集频率的采集控制参数，并根据该采集频率以及采集控制参数中的其他参数采集PHP程序中的各函数在测试过程中的性能测试指标，这样，在模拟海量用户对服务器进行访问，以对服务器中的PHP程序性能测试过程中，可以避免PHP程序中的每个函数在被调用时均进行性能测试指标采集的现象，进而可以减小性能测试自身对服务器资源的消耗；本发明的采集控制参数可以根据实际测试需求来设置相应的参数项，这样，可以使PHP程序性能测试结果满足实际的多种测试需求；本发明通过在PHP程序性能测试过程中将每次获取的各性能测试记录均存储于同一个数据库中，在进行PHP程序性能分析过程中，可以方便的基于该数据库中存储的所有性能测试记录而进行各种形式的数据统计，以从PHP程序的所有函数中方便的定位出有待于进一步改进的函数，有利于PHP程序性能测试；从而本发明提高了PHP程序性能测试的准确性、易用性以及多样性。

上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。说明书附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

图 1 示出了根据本发明实施例一的程序性能测试方法流程图；

图 2 示出了根据本发明实施例二的程序性能测试装置示意图；

图 3 示意性地示出了用于执行根据本发明的方法的服务器的框图；以及

图 4 示意性地示出了用于保持或者携带实现根据本发明的方法的程序代码的存储单元。

具体实施方式

下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

实施例一、程序性能测试方法。

下面结合图 1 对本实施例的方法进行详细说明。

图 1 中，S100、预先设置针对待测试 PHP 程序的性能测试指标的采集控制参数，且采集控制参数至少包括：性能测试指标的采集频率。

具体的，本实施例中的性能测试指标的采集频率主要用于控制性能测试指标的采集频度。

本实施例中的采集控制参数还可以根据实际测试需求而包括：总采集控制开关、实时监控性能测试指标的控制开关、函数占用 CPU 情况的采集控制开关、函数占用内存情况的采集控制开关、函数在预定时间间隔内调用次数的采集控制开关以及函数响应时间的采集控制开关中的一个或者多个（甚至全部）。

另外，本实施例中的采集控制参数还可以包括：文本文件的路径信息和文件名称信息、以及数据库的路径信息和数据库名称信息中的一个或者两个。当然，本实施例也可以根据实际应用中的具体测试需求而添加或者删减相应的参数项。

上述总采集控制开关主要用于决定是否执行 PHP 程序性能测试指标的采集操作，如总采集控制开关设置为开，表示在 PHP 程序运行过程中执行性能测试指标的采集操作；总采集控制开关设置为关，表示在 PHP 程序运行过程中不执行 PHP 程序性能测试指标的采集操作。

上述实时监控性能测试指标的控制开关主要用于决定是否需要对采集到的性能测试指标进行实时监控，如实时监控性能测试指标设置为开，表示在每次采集到性能测试指标时均应实时显示该性能测试指标；实时监控性能测试指标设置为关，表示采集到的所有性能测试指标均不需要实时显示。

上述函数占用 CPU 情况的采集控制开关主要用于决定是否对函数占用 CPU 情况这一性能测试指标进行采集操作，如函数占用 CPU 情况的采集控制开关设置为开，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均应采集函数占用 CPU 情况这一性能测试指标；函数占用 CPU 情况的采集控制开关设置为关，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均不需要采集函数占用 CPU 情况这一性能测试指标。

上述函数占用内存情况的采集控制开关主要用于决定是否对函数占用内存情况这一性能测试指标进行采集操作，如函数占用内存情况的采集控制开关设置为开，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均应采集函数占用内存情况这一性能测试指标；函数占用内存情况的采集控制开关设置为关，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均不需要采集函数占用内存情况这一性能测试指标。

上述函数在预定时间间隔内调用次数的采集控制开关主要用于决定是否对函数在预定时间间隔内调用次数这一性能测试指标进行采集操作，如函数在预定时间间隔内调用次数的采集控制开关设置为开，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均应采集函数在预定时间间隔内调用次数这一性能测试指标；函数在预定时间间隔内调用次数的采集控制开关设

置为关，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均不需要采集函数在预定时间间隔内调用次数这一性能测试指标。

上述函数响应时间的采集控制开关主要用于决定是否对所有的函数响应时间这一性能测试指标进行采集操作，如函数响应时间的采集控制开关设置为开，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均应采集所有的函数响应时间这一性能测试指标；函数占用 CPU 情况的采集控制开关设置为关，表示在 PHP 程序运行过程中每次执行性能测试指标的采集操作时均不需要采集函数占用 CPU 情况这一性能测试指标。

上述文本文件的路径信息和文件名称信息主要用于描述在每次采集到的性能测试指标均先暂时存储于文本文件的情况下，应依据该文本文件的路径信息和文件名称信息将性能测试指标暂时存储于相应的文本文件中；或者在每次采集到的性能测试指标均最终存储于文本文件的情况下，应依据该文本文件的路径信息和文件名称信息将性能测试指标存储于相应的文本文件中。

上述数据库的路径信息和数据库名称信息主要用于描述在每次采集到的性能测试指标直接存储于数据库的情况下，应依据该数据库的路径信息和数据库名称信息将性能测试指标直接存储于相应的数据库中；或者在每次采集到的性能测试指标先存储于文本文件中且在性能测试结束时再将文本文件中的性能测试指标存储于数据库中的情况下，应依据该数据库的路径信息和数据库名称信息将相应的文本文件中的性能测试指标存储于相应的数据库中。

S110、在模拟海量用户对服务器进行访问，以测试服务器中的 PHP 程序性能的过程中，获取预先设置的采集控制参数。

具体的，本实施例中的 PHP 程序运行于服务器中，且该服务器会面对海量用户（如数以百万计的用户）的访问，这样，运行于服务器中的 PHP 程序所包含的任意一个函数都有可能在一个很短的时间间隔内被多次（如数以万计，甚至数以百万计）调用执行。

获取预先设置的采集控制参数的操作通常是在进行 PHP 程序性能测试的开始阶段执行。采集控制参数可以采用配置文件的形式，且该配置文件可以预先设置于被测试的 PHP 程序所在的服务器中，这样，在 PHP 程序性能测试过程中，可以方便的从服务器本地存储的配置文件中获得采集控制参数。

采集控制参数中的各参数项的具体取值可以根据 PHP 程序性能测试需求的不同而被更改。为了方便设置采集控制参数中各参数项的取值，可以在 PHP 程序性能测试的开始阶段显示采集控制参数的设置界面，该界面中显示有采集控制参数所包含的所有参数项的名称以及各参数项的当前取值（如缺省值），用户可以通过该界面来设置各参数项的具体取值，如通过该界面将性能测试指标的采集频率设置为预定时间间隔、将总采集控制开关的取值设置为开、将实时监控性能测试指标的控制开关的取值设置为关、将函数占用 CPU 情况的采集控制开关设置为开、将函数占用内存情况的采集控制开关设置为开、将函数在预定时间间隔内调用次数的采集控制开关设置为开、将函数响应时间的采集控制开关设置为开、设置文本文件的路径信息和文件名称信息以及数据库的路径信息和数据库名称信息，然后，通过点击采集控制参数设置界面中的保存按钮等方式，使上述设置的参数项的具体取值本地存储（如存储于本地的配置文件中）。

一个配置文件的具体例子如下所示：

1、define('XHPROF_ENABLE',true); //是否开启 PHP 程序性能测试指标的采集操作，即设置总采集控制开关的取值；

if(XHPROF_ENABLE == true && function_exists('xhprof_enable')){

xhprof_enable();

}

其中，上述语句中的“function_exists('xhprof_enable')”主要用于判断 xhprof_enable（）是否存在，以使程序逻辑更严谨且程序运行更稳定；

2、define('XHPROF_TYPE,0); //0：是性能，1：是功能（其中的性能是指性能测试应用场景，即针对大批量用户访问的情况；其中的功能是指手工性能测试应用场景，即针对测试人员手工点击实现用户访问的情况，也即针对用户访问量很小的情况）；

if(XHPROF_TYPE==1){

$percent=1；调整性能测试指标的采集频率为每次用户访问都进行采集；

$run_id = $xhprof_runs->save_run($xhprof_data, "hx");开启页面监控，并且将采集数据生成一个链接在测试页面展现；

echo "view";（该语句表示在性能测试完成后，所有采集数据会在测试页面自动生成一个链接，以方便用户的查看性能测试结果）；

}

If（XHPROF_TYPE==1）{

}

3、define('XHPROF_LOG_DATA','/usr/local/apache2/htdocs/xhprof/');//日志信息存放路径;

4、define('PERCENT',1);//记录日志信息的频率，即设置性能测试指标的采集频率的取值;

通过该函数$percent= mt_rand(1,PERCENT)生成随机数;

If（$percent==1）{

xhprof_enable(XHPROF_FLAGS_CPU + XHPROF_FLAGS_MEMORY);

}（该语句表示每次采集的性能测试指标应包含有函数占用 CPU 的详细信息和函数占用 MEMORY 的详细信息）

5、define('PAGEVIEW',true);//是否开启页面监控，即每次采集的数据都会生成一个包含详细采集数据的页面，该页面就是上面所提及的可以查看性能测试结果的测试页面;

if(开启性能测试 & PAGEVIEW==true){

$run_id = $xhprof_runs->save_run($xhprof_data, "hx");

}（该语句表示采集的性能测试指标所存储的文件 id，该文件 id 可以用来自动生成展示性能测试结果的测试页面）

Else

不保存监控页面;

6、define('BUILTINS',true);// 是否监控 PHP 内部函数性能数（需要特别说明的是，PHP 内部函数是 PHP 自带的一些常用公共函数，不是 PHP 程序开发人员编写的，通常情况下，在进行 PHP 程序性能测试时，不会对 PHP 内部函数的代码进行核查）

if（BUILTINS==true）{

xhprof_enable(XHPROF_FLAGS_NO_BUILTINS);

}（该语句表示关闭对 PHP 内部函数性能测试指标的采集操作）

7、define('CPU_MONITOR',true);//是否监控函数占用 CPU 的详细信息(如在包含 PHP 内部函数的情况下函数占用 CPU 的时间以及在不包含 PHP 内部函数的情况下函数占用 CPU 的时间），默认只监控在不包含 PHP 内部函数的情况下函数占用 CPU 的时间，此选项可用于统计更详细的 CPU 使用信息，本实施例通过对函数占用 CPU 的时间进行采集，从而可以通过函数占用 CPU 的时间计算出函数占用 CPU 的百分比;

if（CPU_MONITOR ==true）{

xhprof_enable(XHPROF_FLAGS_CPU); }（该语句表示监控在包含 PHP 内部函数的情况下函数占用 CPU 的详细信息）

8、define('MEM_MONITOR',true);// 是否监控函数占用内存的详细信息（如在包含 PHP 内部函数的情况下函数占用内存情况以及在不包含 PHP 内部函数的情况下函数占用内存情况，也可以统计内存占用峰值等数据），默认只监控在不包含 PHP 内部函数的情况下函数占用内存的字节数，此选项可用于统计更详细的内存占用信息;

if（MEM_MONITOR ==true）{

xhprof_enable(XHPROF_FLAGS_MEMORY); （该语句表示监控在包含 PHP 内部函数的情况下函数占用内存的详细信息）

S120、根据采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标。

具体的，本实施例应根据采集控制参数中的各参数项的设置值来执行性能测试指标的采集操作，一个具体的例子:

在采集控制参数中的总采集控制开关的设置值为开的情况下，应根据性能测试指标的采集频率来执行性能测试指标的采集操作;在采集控制参数中的总采集控制开关的设置值不为开的情况下，则不会执行任何性能测试指标的采集操作。

在实时监控性能测试指标的控制开关的设置值为开的情况下，应在根据性

能测试指标的采集频率每次执行了性能测试指标的采集操作时，将采集到的性能测试指标实时显示给用户，如根据采集到的性能测试指标生成一个链接，并在监控页面中实时显示该链接的内容；在实时监控性能测试指标的控制开关的设置值不为开的情况下，则不会执行上述实时显示的操作。

在函数占用 CPU 情况的采集控制开关的设置值为开的情况下，应在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均应采集函数占用 CPU 情况这一性能测试指标；在函数占用 CPU 情况的采集控制开关的设置值不为开的情况下，则在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均不需要采集函数占用 CPU 情况这一性能测试指标。

在函数占用内存情况的采集控制开关的设置值为开的情况下，应在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均应采集函数占用内存情况这一性能测试指标；在函数占用内存情况的采集控制开关的设置值不为开的情况下，则在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均不需要采集函数占用内存情况这一性能测试指标。

在函数在预定时间间隔内调用次数（如在单位时间内调用次数）的采集控制开关的设置值为开的情况下，应开启 Xhprof 的方法调用次数统计功能，这样，在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均可以采集到函数在预定时间间隔内调用次数这一性能测试指标；在函数在预定时间间隔内调用次数（如函数在单位时间内调用次数）的采集控制开关的设置值不为开的情况下，则可以不开启 Xhprof 的调用次数统计功能，这样，在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，也不会采集到函数在预定时间间隔内调用次数这一性能测试指标。

在函数响应时间的采集控制开关的设置值为开的情况下，应开启 Xhprof 的函数响应时间统计功能，这样，在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均可以采集到各函数的响应时间这一性能测试指标；在函数响应时间的采集控制开关的设置值不为开的情况下，则可以不开启 Xhprof 的函数响应时间统计功能，这样，在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，也不会采集到各函数的响应时间这一性能测试指标。

本实施例可以通过调用 Xhprof 提供的相应的 API（Application Programming Interface，应用程序编程接口）来采集函数标识、函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内调用次数以及函数响应时间等性能参数指标。

获取性能参数指标的一个具体的例子为：

$mu = $PerformaceInfo['mu']; （mu 表示通过采集获取到的函数占用内存的字节数，且单位为字节）

$wt = $PerformaceInfo['wt']; （wt 表示通过采集而获取到的函数响应的时间）

$ct = $PerformaceInfo['ct']; （ct 表示通过采集而获取到的函数在预定时间间隔内调用次数）

$cpu= $PerformaceInfo['cpu']; （cpu 表示通过采集而获取到的函数花费的 CPU 时间(单位为微秒，即百万分之一秒)）

$wt = $wt/1000;（通过该计算式可以获得函数占用内存的字节数，且单位为 KB）

由于在 PHP 程序性能测试过程中通常会模拟海量用户（如一百万用户甚至更多）对服务器进行访问，因此，在 PHP 程序中的每个函数在被调用时均进行性能测试指标采集的话，则会给服务器带来较大的资源消耗，从而给服务器带来较大的压力，这会影响到 PHP 程序性能测试的准确性。本实施例通过根据性能测试指标的采集频率来采集性能测试指标，可以避免 PHP 程序中的每个函数在被调用时均进行性能测试指标采集的现象，从而可以使获得的 PHP 程序性能测试的结果更准确。

S130、将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中。

具体的，本实施例多次采集获得的包含有函数标识以及性能测试指标的所有性能测试记录最终会存储于同一个数据库中，以便于执行后续的性能分析操作；而且，以数据库的方式存储本次 PHP 程序性能测试过程中所获得的所有性能测试记录还可以方便的比较该 PHP 程序在优化前后的性能测试结果，有利于测试人员权衡优化 PHP 程序的利弊。

采集到的所有的性能测试记录存储于同一个数据库中的具体实现方式主要有如下两种:

实现方式一、在配置文件中既设置有文本文件的路径信息和文件名称信息，又设置有数据库的路径信息和数据库名称信息的情况下，在 PHP 程序运行过程中，根据性能测试指标的采集频率以及采集控制参数中的其他相应的参数项采集到的包含有函数标识以及性能测试指标的各性能测试记录会先根据配置文件中的文本文件的路径信息和文件名称信息存储于文本文件中，在 PHP 程序运行结束后，再根据配置文件中的数据库的路径信息和数据库名称信息将该文本文件中的所有性能测试记录转存储于数据库中。

实现方式二、在配置文件中没有设置文本文件的路径信息和文件名称信息，但是设置有数据库的路径信息和数据库名称信息的情况下，在 PHP 程序运行过程中，根据性能测试指标的采集频率以及采集控制参数中的其他相应的参数项采集到的包含有函数标识以及性能测试指标的各性能测试记录会直接根据配置文件中的数据库的路径信息和数据库名称信息存储于数据库中。

上述性能测试记录可以为 Xhprof 所提供的 log（日志）的形式。性能测试记录所包含的函数标识可以为函数名称等能够于唯一标识 PHP 程序中的一个函数的标识信息；性能测试记录所包含的除了函数标识之外的其他内容取决于采集控制参数中的参数项。从 PHP 性能全面分析的角度考虑，性能测试记录通常会包括函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内调用次数以及函数响应时间。

上述函数占用 CPU 情况可以具体为：函数对 CPU 占用率和/或者函数对 CPU 的占用时间等；上述函数占用内存情况可以具体为：函数对内存占有率和/或者函数对内存占用时间等；上述函数在预定时间间隔调用次数可以具体为：函数在单位时间内被其他函数调用的次数和/或函数在单位时间内调用其他函数的次数等；上述函数响应时间可以具体为：从函数被其他函数调用开始到函数返回相应的结果所花费的时间。

S140、根据数据库中存储的性能测试记录确定 PHP 程序中的函数的性能。

具体的，本实施例是对数据库中存储的各性能测试记录进行统计，并对统计结果进行分析来确定 PHP 程序中的函数的性能的。对数据库中存储的各性

能测试记录进行统计可以具体为：对属于同一个函数的性能测试指标分别进行统计计算（如求均值）。对统计结果进行分析确定 PHP 程序中的函数的性能可以具体为：将性能测试指标的统计计算结果与其对应的预先设置的阈值进行比较，选取出超过阈值的性能测试指标，将选取出的性能测试指标以及其所属的函数可以展示给测试用户，这些选取出的性能测试指标所属的函数是测试人员需要关注的函数；例如，可以通过对数据库中的性能测试记录的统计分析而选取出分别占用 CPU 最多、占用内存最多、被调用次数最多以及响应时间最慢的前 10 位的函数，实现函数定位，测试人员可以有针对性的对定位出的函数进行分析改进。上述预先设置的阈值可以是根据测试经验预先设置的阀值。如果在 PHP 程序性能测试过程中，采集到的性能测试记录需要实时的展现给测试人员，则超出预先设置的阈值的性能测试记录可以展现为不同的颜色，如字体为红色或者采用高亮方式显示。

另外，选取出的性能测试指标所属的函数所对应的所有的性能测试指标可以以折线图、统计表或者文本文件等方式呈现给测试人员；其中的折线图可以使测试人员更加直观方便的对需要改进的函数进行定位。

在采用折线图进行函数性能分析的情况下，二维坐标系的横坐标可以表示时间，二维坐标系的纵坐标可以表示性能测试指标的数值大小，这样，一个函数的一个性能测试指标可以根据其对应的采集时间而被表示于二维坐标轴中，从而该函数的多个性能测试指标形成一条折线。

一个函数的多个性能测试指标所形成的多条折线可以表示在同一个二维坐标系中；例如，在同一个二维坐标系中示出了一个函数的四条折线，这四条折线分别是针对该函数的所有函数占用 CPU 情况而形成的第一条折线、针对该函数的所有函数占用内存情况而形成的第二条折线、针对该函数的所有函数在预定时间间隔内调用次数而形成的第三条折线、以及针对该函数的所有函数响应时间而形成的第四条折线。

另外，多个函数的同一性能测试指标所形成的多条折线可以表示在同一个二维坐标系中；例如，可以在同一个二维坐标系中示出四个函数的四条折线，这四条折线分别是属于第一个函数的所有函数占用 CPU 情况所形成的第一折线、属于第二个函数的所有函数占用 CPU 情况所形成的第二折线、属于第三

个函数的所有函数占用 CPU 情况所形成的第三折线以及属于第四个函数的所有函数占用 CPU 情况所形成的第四折线。

本实施例可以利用现有的绘图插件来绘制上述折线图，例如，将从数据库的性能测试记录中选取出的性能测试记录按照预先设定的方式提供给绘图插件，从而绘图插件可以根据其接收到的数据绘制出上述折线图。

本实施例通过对数据库中记录的所有性能测试记录的分析，可以使测试人员很方便的定位出 PHP 程序性能瓶颈，且数据库中记录的性能测试记录还可以很方便的和采用其他测试工具所获得的测试结果进行合并分析，以满足多种 PHP 程序的测试需求。

实施例二、基于 Xhprof 的 PHP 程序性能测试装置，本实施例的装置可以设置于运行需要进行性能测试的 PHP 程序的服务器中。

下面结合图 2 对本实施例的装置进行详细说明。

图 2 中示出的基于 Xhprof 的 PHP 程序性能测试装置通常设置于服务器中，且该装置主要包括：存储模块 200、获取模块 210、采集模块 220、记录模块 230 以及性能模块 240。

存储模块 200 主要配置为预先设置有针对 PHP 程序的性能测试指标的采集控制参数，且这里的采集控制参数至少包括：性能测试指标的采集频率。

具体的，存储模块 200 中预先设置的性能测试指标的采集频率主要用于控制性能测试指标的采集频度。存储模块 200 中存储的采集控制参数还可以根据实际测试需求而包括：总采集控制开关、实时监控性能测试指标的控制开关、函数占用 CPU 情况的采集控制开关、函数占用内存情况的采集控制开关、函数在预定时间间隔内调用次数的采集控制开关以及函数响应时间的采集控制开关中的一个或者多个（甚至全部）。另外，存储模块 200 中存储的采集控制参数还可以包括：文本文件的路径信息和文件名称信息、以及数据库的路径信息和数据库名称信息中的一个或者两个。当然，本实施例也可以根据实际应用中的具体测试需求而添加或者删减存储模块 200 中存储的相应的参数项。

上述总采集控制开关、实时监控性能测试指标的控制开关、函数占用 CPU 情况的采集控制开关、函数占用内存情况的采集控制开关、函数在预定时间间隔内调用次数的采集控制开关、函数响应时间的采集控制开关、文本文件的路

径信息和文件名称信息、以及数据库的路径信息和数据库名称信息具体如上述方法实施例中的描述，在此不再重复说明。

存储模块 200 可以采用配置文件的形式来存储采集控制参数。

获取模块 210 主要配置为在模拟海量用户对服务器进行访问以测试服务器中的 PHP 程序性能的过程中，获取采集控制参数。

具体的，获取模块 210 可以从存储模块 200 中获取采集控制参数，且采集控制参数中的各参数项的具体取值可以根据 PHP 程序性能测试需求的不同而被更改。为了方便设置采集控制参数中各参数项的取值，获取模块 210 可以在 PHP 程序性能测试的开始阶段显示采集控制参数的设置界面，该界面中显示有采集控制参数所包含的所有参数项的名称以及各参数项的当前取值（如缺省值），用户可以通过该界面来设置各参数项的具体取值，如通过该界面将性能测试指标的采集频率设置为预定时间间隔、将总采集控制开关的取值设置为开、将实时监控性能测试指标的控制开关的取值设置为关、将函数占用 CPU 情况的采集控制开关设置为开、将函数占用内存情况的采集控制开关设置为开、将函数在预定时间间隔内调用次数的采集控制开关设置为开、将函数响应时间的采集控制开关设置为开、设置文本文件的路径信息和文件名称信息以及数据库的路径信息和数据库名称信息，然后，通过点击采集控制参数设置界面中的保存按钮等方式，使上述设置的参数项的具体取值本地存储于存储模块 200 中的配置文件中。

采集模块 220 主要配置为根据采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标。

具体的，采集模块 220 应根据获取模块 210 获得到的采集控制参数中的各参数项的设置值来执行性能测试指标的采集操作，一个具体的例子：

在采集控制参数中的总采集控制开关的设置值为开的情况下，采集模块 220 应根据性能测试指标的采集频率来执行性能测试指标的采集操作；在采集控制参数中的总采集控制开关的设置值不为开的情况下，则采集模块 220 不会执行任何性能测试指标的采集操作。

在实时监控性能测试指标的控制开关的设置值为开的情况下，采集模块 220 应在根据性能测试指标的采集频率每次执行了性能测试指标的采集操作时，将采集到的性能测试指标实时显示给用户，如根据采集到的性能测试指标

生成一个链接，并在监控页面中实时显示该链接的内容；在实时监控性能测试指标的控制开关的设置值不为开的情况下，则采集模块 220 不会执行上述实时显示的操作。

在函数占用 CPU 情况的采集控制开关的设置值为开的情况下，采集模块 220 应在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均应采集函数占用 CPU 情况这一性能测试指标；在函数占用 CPU 情况的采集控制开关的设置值不为开的情况下，则采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均不需要采集函数占用 CPU 情况这一性能测试指标。

在函数占用内存情况的采集控制开关的设置值为开的情况下，采集模块 220 应在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均应采集函数占用内存情况这一性能测试指标；在函数占用内存情况的采集控制开关的设置值不为开的情况下，则采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均不需要采集函数占用内存情况这一性能测试指标。

在函数在预定时间间隔内调用次数（如在单位时间内调用次数）的采集控制开关的设置值为开的情况下，采集模块 220 应开启 Xhprof 的方法调用次数统计功能，这样，采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均可以采集到函数在预定时间间隔内调用次数这一性能测试指标；在函数在预定时间间隔内调用次数（如函数在单位时间内调用次数）的采集控制开关的设置值不为开的情况下，则采集模块 220 可以不开启 Xhprof 的调用次数统计功能，这样，采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，也不会采集到函数在预定时间间隔内调用次数这一性能测试指标。

在函数响应时间的采集控制开关的设置值为开的情况下，采集模块 220 应开启 Xhprof 的函数响应时间统计功能，这样，采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的采集操作时，均可以采集到各函数的响应时间这一性能测试指标；在函数响应时间的采集控制开关的设置值不为开的情况下，则采集模块 220 可以不开启 Xhprof 的函数响应时间统计功能，这样，采集模块 220 在根据性能测试指标的采集频率每次执行性能测试指标的

采集操作时，也不会采集到各函数的响应时间这一性能测试指标。

采集模块 220 可以通过调用 Xhprof 提供的相应的 API（Application Programming Interface，应用程序编程接口）来采集函数标识、函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内调用次数以及函数响应时间等性能参数指标。

由于在PHP程序性能测试过程中通常会模拟海量用户对服务器进行访问，因此，在 PHP 程序中的每个函数在被调用时采集模块 220 均进行性能测试指标采集的话，则会给服务器带来较大的资源消耗，从而给服务器带来较大的压力，这会影响到 PHP 程序性能测试的准确性。本实施例中的采集模块 220 通过根据性能测试指标的采集频率来采集性能测试指标，可以避免 PHP 程序中的每个函数在被调用时均进行性能测试指标采集的现象，从而可以使获得的 PHP 程序性能测试的结果更准确。

记录模块 230 主要配置为将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中。

具体的，采集模块 220 多次采集获得的包含有函数标识以及性能测试指标的所有性能测试记录最终会被记录模块 230 存储于同一个数据库中，以便于性能模块 240 执行后续的性能分析操作；而且，记录模块 230 以数据库的方式存储本次 PHP 程序性能测试过程中所获得的所有性能测试记录还可以方便测试人员比较该 PHP 程序在优化前后的性能测试结果，有利于测试人员权衡优化 PHP 程序的利弊。

记录模块230将采集模块220采集到的所有的性能测试记录存储于同一个数据库中的具体实现方式主要有如下两种：

实现方式一、在配置文件中既设置有文本文件的路径信息和文件名称信息，又设置有数据库的路径信息和数据库名称信息的情况下，在 PHP 程序运行过程中，采集模块 220 根据性能测试指标的采集频率以及采集控制参数中的其他相应的参数项采集到的包含有函数标识以及性能测试指标的各性能测试记录会被记录模块 230 先根据配置文件中的文本文件的路径信息和文件名称信息存储于文本文件中，在 PHP 程序运行结束后，记录模块 230 再根据配置文件中的数据库的路径信息和数据库名称信息将该文本文件中的所有性能测试记录转存储于数据库中。

实现方式二、在配置文件中没有设置文本文件的路径信息和文件名称信息，但是设置有数据库的路径信息和数据库名称信息的情况下，在 PHP 程序运行过程中，采集模块 220 根据性能测试指标的采集频率以及采集控制参数中的其他相应的参数项采集到的包含有函数标识以及性能测试指标的各性能测试记录会被记录模块 230 直接根据配置文件中的数据库的路径信息和数据库名称信息存储于数据库中。

上述性能测试记录可以为 Xhprof 所提供的 log（日志）的形式。性能测试记录所包含的函数标识可以为函数名称等能够于唯一标识 PHP 程序中的一个函数的标识信息；性能测试记录所包含的除了函数标识之外的其他内容取决于采集控制参数中的参数项。从 PHP 性能全面分析的角度考虑，性能测试记录通常会包括函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内调用次数以及函数响应时间。

性能模块 240 主要配置为根据数据库中存储的性能测试记录确定 PHP 程序中的函数的性能。

具体的，性能模块 240 是对数据库中存储的各性能测试记录进行统计，并对统计结果进行分析来确定 PHP 程序中的函数的性能的。性能模块 240 对数据库中存储的各性能测试记录进行统计可以具体为：性能模块 240 对属于同一个函数的性能测试指标分别进行统计计算（如求均值）。性能模块 240 对统计结果进行分析确定 PHP 程序中的函数的性能可以具体为：性能模块 240 将性能测试指标的统计计算结果与其对应的预先设置的阈值进行比较，选取出超过阈值的性能测试指标，性能模块 240 将选取出的性能测试指标以及其所属的函数可以展示给测试人员，这些选取出的性能测试指标所属的函数是测试人员需要关注的函数；例如，性能模块 240 可以通过对数据库中的性能测试记录的统计分析而选取出分别占用 CPU 最多、占用内存最多、被调用次数最多以及响应时间最慢的前 10 位的函数，实现函数定位，测试人员可以有针对性的对定位出的函数进行分析改进。上述预先设置的阈值可以测试人员是根据测试经验预先设置的阀值。如果在 PHP 程序性能测试过程中，采集模块 220 采集到的性能测试记录需要实时的展现给测试人员，则超出预先设置的阈值的性能测试记录可以展现为不同的颜色，如字体为红色或者采用高亮方式显示。

另外，性能模块 240 选取出的性能测试指标所属的函数所对应的所有的性

能测试指标可以由性能模块240以折线图、统计表或者文本文件等方式呈现给测试人员；其中的折线图可以使测试人员更加直观方便的对需要改进的函数进行定位。

测试人员性能模块 240 可以利用现有的绘图插件来绘制上述折线图，例如，性能模块240将从数据库的性能测试记录中选取出的性能测试记录按照预先设定的方式提供给绘图插件，从而绘图插件可以根据其接收到的数据绘制出上述折线图。具体的有关折线图的说明请参见上述方法实施例中的描述，在此不再重复说明。

性能模块240通过对数据库中记录的所有性能测试记录的分析，可以使测试人员很方便的定位出 PHP 程序性能瓶颈，且数据库中记录的性能测试记录还可以很方便的和采用其他测试工具所获得的测试结果进行合并分析，以满足多种 PHP 程序的测试需求。

本发明的各个部件实施例可以以硬件实现，或者以在一个或者多个处理器上运行的软件模块实现，或者以它们的组合实现。本领域的技术人员应当理解，可以在实践中使用微处理器或者数字信号处理器（DSP）来实现根据本发明实施例的客户端或服务器中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者设备程序（例如，计算机程序和计算机程序产品）。这样的实现本发明的程序可以存储在计算机可读介质上，或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到，或者在载体信号上提供，或者以任何其他形式提供。

例如，图 3 示出了可以实现根据本发明上述方法的服务器，例如搜索引擎服务器。该服务器传统上包括处理器 310 和以存储器 330 形式的计算机程序产品或者计算机可读介质。存储器 330 可以是诸如闪存、EEPROM（电可擦除可编程只读存储器）、EPROM、硬盘或者 ROM 之类的电子存储器。存储器 330 具有用于执行上述方法中的任何方法步骤的程序代码 351 的存储空间 350。例如，用于程序代码的存储空间 350 可以包括分别用于实现上面的方法中的各种步骤的各个程序代码 351。这些程序代码可以从一个或者多个计算机程序产品中读出或者写入到这一个或者多个计算机程序产品中。这些计算机程序产品包

括诸如硬盘，紧致盘（CD）、存储卡或者软盘之类的程序代码载体。这样的计算机程序产品通常为如参考图 4 所述的便携式或者固定存储单元。该存储单元可以具有与图 3 的服务器中的存储器 330 类似布置的存储段、存储空间等。程序代码可以例如以适当形式进行压缩。通常，存储单元包括计算机可读代码 351’，即可以由例如诸如 310 之类的处理器读取的代码，这些代码当由服务器运行时，导致该服务器执行上面所描述的方法中的各个步骤。

本文中所称的“一个实施例”、“实施例”或者“一个或者多个实施例”意味着，结合实施例描述的特定特征、结构或者特性包括在本发明的至少一个实施例中。此外，请注意，这里“在一个实施例中”的词语例子不一定全指同一个实施例。

在此处所提供的说明书中，说明了大量具体细节。然而，能够理解，本发明的实施例可以在没有这些具体细节的情况下被实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制，并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中，不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中，这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

此外，还应当注意，本说明书中使用的语言主要是为了可读性和教导的目的而选择的，而不是为了解释或者限定本发明的主题而选择的。因此，在不偏离所附权利要求书的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。对于本发明的范围，对本发明所做的公开是说明性的，而非限制性的，本发明的范围由所附权利要求书限定。

权 利 要 求 书

1、一种程序性能测试方法，用于基于 Xhprof 的 PHP 程序性能测试，该方法包括:

预先设置针对待测试 PHP 程序的性能测试指标的采集控制参数，且所述采集控制参数至少包括：性能测试指标的采集频率;

在模拟海量用户对服务器进行访问以测试服务器中的 PHP 程序性能的过程中，获取所述采集控制参数;

根据所述采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标;

将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中;

根据所述数据库中存储的性能测试记录确定 PHP 程序中的函数的性能。

2、如权利要求 1 所述的程序性能测试方法，其中，所述采集控制参数还包括：总采集控制开关以及实时监控性能测试指标的控制开关;

且所述根据所述采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标包括:

在所述总采集控制开关为开且所述实时监控性能测试指标的控制开关为开的情况下，基于所述采集控制参数中的其他参数设置值采集 PHP 程序中的各函数在测试过程中的性能测试指标，并将采集到的性能测试记录实时呈现。

3、如权利要求 1 或 2 所述的程序性能测试方法，其中，所述性能测试指标包括：函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内的调用次数以及函数响应时间中的至少一个。

4、如权利要求 3 所述的程序性能测试方法，其中，所述采集控制参数还包括：函数占用 CPU 情况的采集控制开关、函数占用内存情况的采集控制开关、函数在预定时间间隔内调用次数的采集控制开关以及函数响应时间的采集控制开关中的至少一个。

5、如权利要求 1 所述的程序性能测试方法，其中，所述将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记

录均存储于同一个数据库中包括:

在 PHP 程序运行过程中，将每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录先存储于同一个文本文件中，且在 PHP 程序运行结束后，将所述文本文件中的所有性能测试记录存储于同一数据库中；或者在 PHP 程序运行过程中，将每次采集所形成的包含有函数标识以及性能测试指标的性能测试记录均直接存储于同一个数据库。

6、如权利要求 5 所述的程序性能测试方法，其中，所述采集控制参数还包括：文本文件的路径信息以及文件名称信息；和/或，数据库的路径信息以及数据库名称信息。

7、如权利要求 1 所述的程序性能测试方法，其中，所述根据所述数据库中存储的性能测试记录确定 PHP 程序中各函数的性能包括:

从数据库存储的性能测试记录中选取性能测试指标超出预定阈值的性能测试记录，并根据选取的性能测试记录中的性能测试指标展现性能测试指标图示。

8、一种计算机程序，包括计算机可读代码，当所述计算机可读代码在计算机上运行时，将执行根据权利要求 1 至 7 中的任一项所述的程序性能测试方法。

9、一种计算机可读介质，其中存储了如权利要求 8 所述的计算机程序。

10、一种程序性能测试装置，用于基于 Xhprof 的 PHP 程序性能测试，该装置包括:

存储模块，配置为预先设置针对待测试 PHP 程序的性能测试指标的采集控制参数，且所述采集控制参数至少包括：性能测试指标的采集频率;

获取模块，配置为在模拟海量用户对服务器进行访问以测试服务器中的 PHP 程序性能的过程中，获取所述采集控制参数;

采集模块，配置为根据所述采集控制参数采集 PHP 程序中的各函数在测试过程中的性能测试指标;

记录模块，配置为将 PHP 程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中;

性能模块，配置为根据所述数据库中存储的性能测试记录确定 PHP 程序中的函数的性能。

11、如权利要求 10 所述的程序性能测试装置，其中，所述采集控制参数还包括：总采集控制开关以及实时监控性能测试指标的控制开关；

且所述采集模块还配置为：

在所述总采集控制开关为开且所述实时监控性能测试指标的控制开关为开的情况下，基于所述采集控制参数中的其他参数设置值采集 PHP 程序中的各函数在测试过程中的性能测试指标，并将采集到的性能测试记录实时呈现。

12、如权利要求 10 或 11 所述的程序性能测试装置，其中，所述性能测试指标包括：函数占用 CPU 情况、函数占用内存情况、函数在预定时间间隔内的调用次数以及函数响应时间中的至少一个；

且所述采集控制参数还包括：函数占用 CPU 情况的采集控制开关、函数占用内存情况的采集控制开关、函数在预定时间间隔内调用次数的采集控制开关以及函数响应时间的采集控制开关中的至少一个。

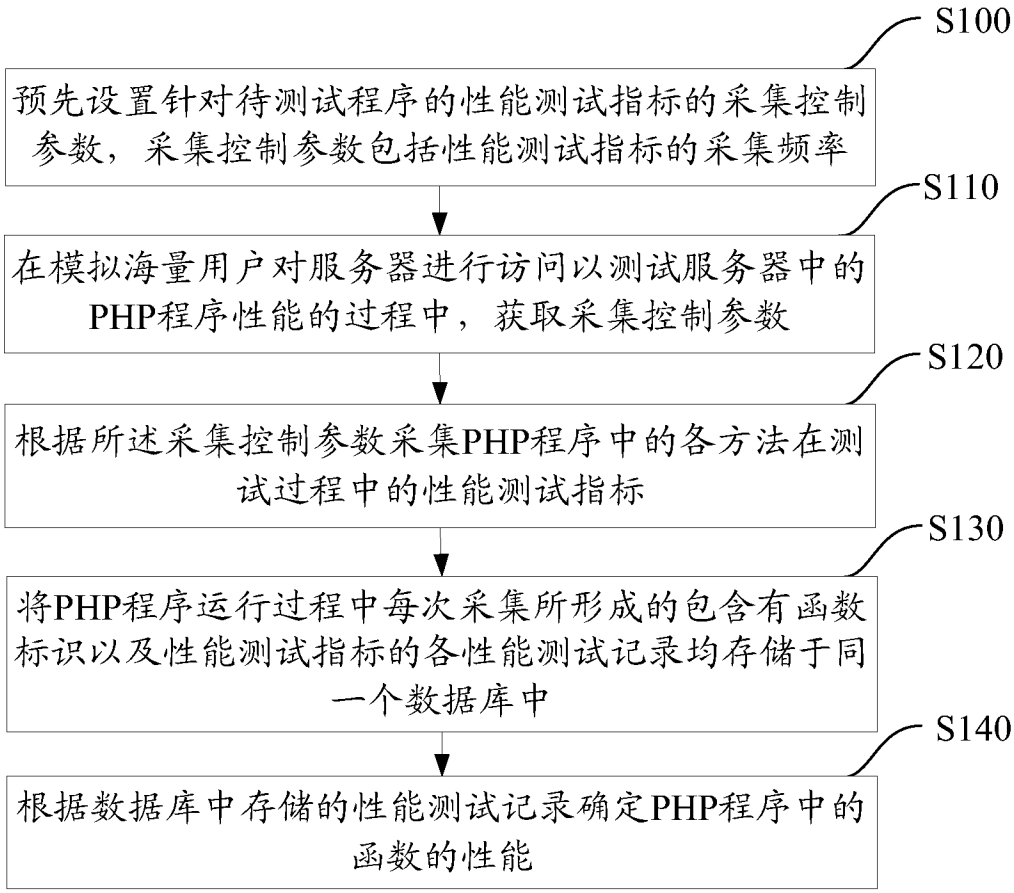

S100
预先设置针对待测试程序的性能测试指标的采集控制参数，采集控制参数包括性能测试指标的采集频率
S110
在模拟海量用户对服务器进行访问以测试服务器中的PHP程序性能的过程中，获取采集控制参数
S120
根据所述采集控制参数采集PHP程序中的各方法在测试过程中的性能测试指标
S130
将PHP程序运行过程中每次采集所形成的包含有函数标识以及性能测试指标的各性能测试记录均存储于同一个数据库中
S140
根据数据库中存储的性能测试记录确定PHP程序中的函数的性能

图 1

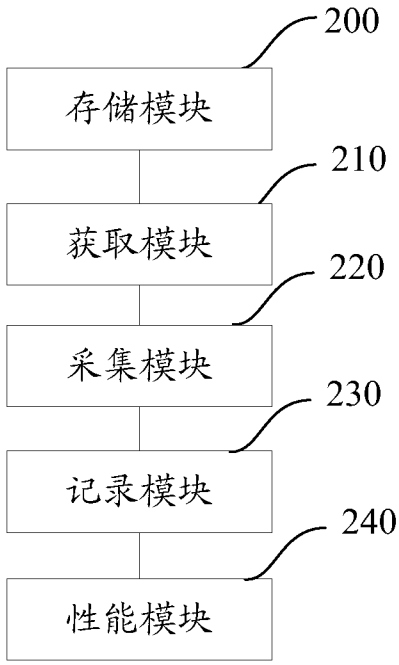

200
存储模块
210
获取模块
220
采集模块
230
记录模块
240
性能模块

图 2

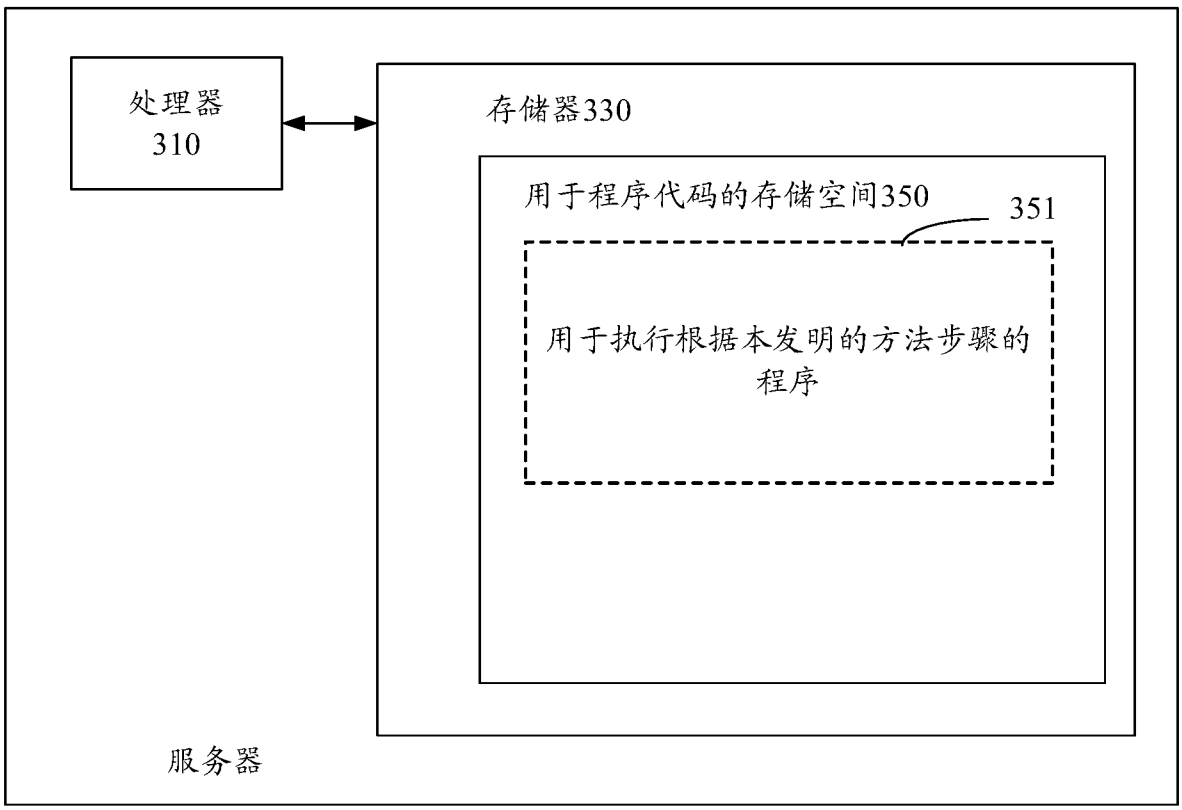

处理器
310
存储器330
用于程序代码的存储空间350
351
用于执行根据本发明的方法步骤的
程序
服务器

图 3

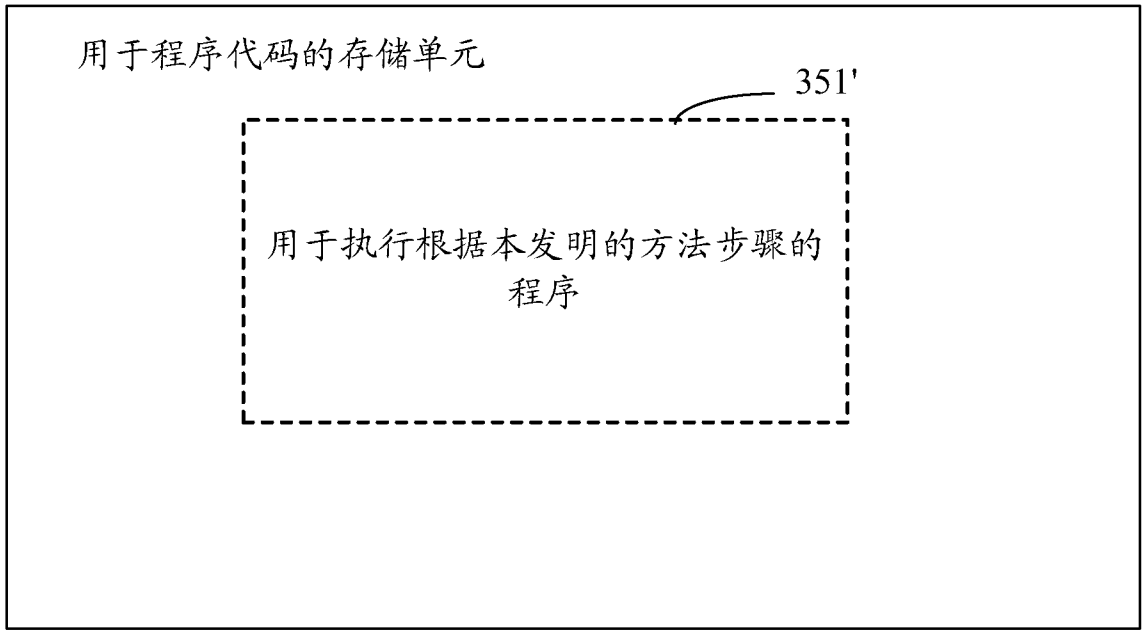


图 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2015/083886

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/36 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, GOOGLE: program, frequency, interval, PHP, xhprof, function, software, hierarchical profiler, performance, test

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"Manual for Xhprof Php Lightweight Performance Analysis Tool (Chinese Version) XHProf php", http: //www. seven2. com. cn/archives/831/, 01 June 2011 (01.06.2011), sections 1, 5 and 7	1-12
A	CN 103049385 A (BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.), 17 April 2013 (17.04.2013), the whole document	1-12
A	US 2009192761 A1 (INTUIT INC.), 30 July 2009 (30.07.2009), the whole document	1-12

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 September 2015 (10.09.2015)	**29 September 2015 (29.09.2015)**
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer **SUN, Guohui** Telephone No.: (86-10) **62413599**

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2015/083886

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103049385 A	17 April 2013	None	
US 2009192761 A1	30 July 2009	None	

国际检索报告

国际申请号
PCT/CN2015/083886

A. 主题的分类

G06F 11/36(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNPAT,WPI,EPODOC,GOOGLE:函数，程序，频度，频率，间隔，性能，PHP，测试,xhprof, function, software, hierarchical profiler,performance,test

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
X	"XHProf php轻量级的性能分析工具使用手册（中文版）" http://www.seven2.com.cn/archives/831/, 2011年 6月 1日 (2011-06-01), 第1,5,7部分	1-12
A	CN 103049385 A (百度在线网络技术北京有限公司) 2013年 4月 17日 (2013-04-17) 全文	1-12
A	US 2009192761 A1 (INTUIT INC.) 2009年 7月 30日 (2009-07-30) 全文	1-12

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2015年 9月 10日	2015年 9月 29日

ISA/CN的名称和邮寄地址	受权官员
中华人民共和国国家知识产权局(ISA/CN) 北京市海淀区蓟门桥西土城路6号 100088 中国 传真号 (86-10)62019451	孙国辉 电话号码 (86-10)62413599

表 PCT/ISA/210 (第2页) (2009年7月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2015/083886

检索报告引用的专利文件	公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN 103049385 A	2013年 4月 17日	无	
US 2009192761 A1	2009年 7月 30日	无	