ICS 35.080
L 77

中华人民共和国国家标准

GB/T 29835.3—2013

系统与软件效率　第3部分：测试方法

Efficiency of system and software—Part 3: Testing method

2013-11-12 发布　　2014-02-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

GB/T 29835 在《系统与软件效率》总标题下，分为如下三部分：

——第 1 部分：指标体系；

——第 2 部分：度量方法；

——第 3 部分：测试方法。

本部分为 GB/T 29835 的第 3 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本部分起草单位：上海鲁齐信息科技有限公司、上海浦东软件平台有限公司、国家应用软件产品质量监督检验中心、深圳市科脉技术有限公司、中国电子技术标准化研究院、上海宝信软件股份有限公司、广东软件评测中心、北京邮电大学、珠海南方软件网络评测中心、上海市计算机软件评测重点实验室、上海嵌入式系统应用工程技术研究中心、南昌金庐软件园软件评测培训有限公司、广州广软信息系统管理咨询有限公司。

本部分主要起草人：张露莹、李家宏、崔岩、苏盼、欧阳树生、张旸旸、袁玉宇、曾昭志、张苏利、肖正坤、侯建华、蔡立志、丁志刚、左家平、黄万民、刘新、袁肃蓉、杨金翠、万方、陈芳芳。

引　言

GB/T 29835 的本部分提出了系统与软件效率的测试方法。为了使效率指标体系适用于不同的测试目标和不同的测试要求，本部分扩展了效率指标体系及度量指标的应用方法，提出了效率指标体系测试应用框架。同时，本部分基于效率指标体系测试应用框架对每个效率指标给出了相应的测试方法和测试流程，并对效率指标体系测试应用框架在不同的应用情况下的应用方法和裁剪给出指导性建议，便于用户或者评测人员进行操作。

本部分所列的测试方法并非适用于每个测试场合，进行效率测试的人员可以根据待测试的系统的特点以及测试目标从本部分中选择适合的指标及其约束进行测试，同时也鼓励测试人员根据测试场合的不同需要增减或修改度量指标、测试约束、测试方法，从而适应特定的测试需要。

本标准预期的主要使用者包括：

a) 软件供方，当：

1) 需要声明软件产品效率特性时；

2) 对照声明的效率特性自行评估其软件产品和系统时；

3) 对软件进行效率相关的产品设计和实现时；

b) 为效率符合性证书或标志进行测试时的第三方评测机构；

c) 潜在的需方，当：

1) 对即将采购的软件产品的效率要求和现有产品的说明信息进行比较；

2) 需要对产品的效率做进一步的改进或者完善、产品的潜在差错而作必需的更改，实际运行的环境和采购的环境存在差异时；

3) 检验效率要求是否被满足。

GB/T 29835.1《系统与软件效率　第 1 部分：指标体系》给出了效率指标体系，GB/T 29835.2《系统与软件效率　第 2 部分：度量方法》给出了如何获得效率指标测量值的度量方法。本部分旨在与 GB/T 29835.1和 GB/T 29835.2 联合使用。

系统与软件效率　第3部分：测试方法

1　范围

GB/T 29835的本部分规定了系统与软件效率的测试方法。本方法是在GB/T 29835.1的指标体系和GB/T 29835.2的度量方法的基础上，给出效率指标体系中每个效率指标的测试方法。本部分对效率指标体系的应用方法进行了扩展，提出了效率指标体系测试应用框架，使效率指标体系尽可能满足各种不同的测试目标和测试需要。

本部分适用于系统与软件的效率测评。

注1：本部分中所指的系统主要是软件系统。

注2：本部分适用于系统与软件通用的效率特性和测试方法，针对特殊类型的系统（如：Web应用系统、嵌入式系统等），可在本部分提出的效率指标体系测试应用框架和测试方法的基础上进行扩充或剪裁。有关剪裁的方法参见附录A。

2　规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 29835.1　系统与软件效率　第1部分：指标体系

GB/T 29835.2　系统与软件效率　第2部分：度量方法

3　术语和定义

GB/T 29835.1和GB/T 29835.2中界定的以及下列术语和定义适用于本文件。

3.1

测试约束　test restriction

影响效率指标测试结果的限制条件。

3.2

结果视图　result view

测试结果的不同展现形式。

注：在本部分中，结果视图表示效率指标测试结果的不同展现形式，反映测试结果不同方面的特性（例如，平均值、最小值、最大值等）。

3.3

测试周期　test period

从启动测试到结束测试之间的时间段。

3.4

有效测试周期　valid test period

在测试周期中，满足所规定的测试约束条件的测试时间段。

注：在效率测试中，有效测试周期中获取的测试数据才能够用于计算所规定测试约束条件下的效率指标值。

4 效率指标体系测试应用框架

4.1 框架体系

系统与软件效率指标体系(见 GB/T 29835.1)说明了构成系统与软件效率的各个关键特性,效率指标的度量值可用于反映系统和软件目前达到的效率水平。但对大部分效率测试来说,每个效率指标的度量值不是唯一的,它依赖于所在环境、用户数量、所针对的操作功能等约束条件,在不同的约束条件下效率指标将产生不同的度量结果,为了获得不同约束条件下的效率指标度量结果,效率测试需要针对各种不同的约束条件进行多次测试。另一方面,一个效率指标的测试结果数据通常为一个统计分布或按时间顺序采集的数据,为了适应不同的结果形式需要,效率指标的度量结果需要不同的统计形式,如最大值、最小值、平均值等,在具有明确的效率目标的情况下,还需要给出效率指标是否满足效率目标的判断,即需要判断效率指标的满足度。因此,效率指标体系中的每个效率指标不再对应一个度量公式和一个度量结果,而是在不同约束条件和结果形式作用下对应为不同的度量公式和度量结果。

效率指标体系测试应用框架是对效率指标体系的扩展应用方法,在应用效率指标体系进行系统和软件的效率评价时,可将效率指标扩展为针对不同测试对象、不同测试条件的度量项,并可通过不同角度来展现效率指标测试结果,从而使效率指标体系满足不同的系统类型、不同的效率目标和不同的测试需求。

效率指标体系测试应用框架的组成要素包括:

a) 效率指标体系——反映系统和软件效率的特性及子特性的集合,并通过效率指标进行效率度量。效率指标体系是本框架的基础,在进行效率测试和评价时,首先应确定要测试的效率指标。

注:有关效率指标体系的详细内容见 GB/T 29835.1。

b) 测试约束——影响效率指标测试结果的限制条件。测试约束产生于对效率指标测试结果产生直接影响的因素。不同的影响因素形成不同类型的测试约束,不同的测试约束交互作用于效率指标,给效率指标限定了各种不同的约束条件。每个测试约束可以有不同的测试约束值,不同测试约束值下的效率指标具有不同的效率指标度量结果。典型的测试约束包括:

   1) 对象约束;
   2) 环境约束;
   3) 负载约束。

c) 结果视图——规定效率指标度量结果的不同展现形式。结果视图通过不同的角度和侧面展现度量结果的不同特征(如,平均值、最小值、最大值、满足效率目标的程度等)。效率指标可选择一个或多个结果视图来展现。

d) 效率指标测试——效率指标的测试方法。在本部分中,效率指标测试首先确定所需要测试的效率指标,确定效率指标所对应的测试约束以及所需要的结果视图,然后针对每个效率指标在不同测试约束值条件下进行效率指标的测试,并按照需要的结果视图计算并展现效率指标的测试结果。

4.2 测试约束

4.2.1 综述

很多因素对效率指标的度量结果产生影响,在不同的系统运行条件和不同的用户访问方式下系统会有不同的运行效率。测试约束限定对效率指标测试结果产生直接影响的因素。

在效率测试中,对测试结果产生影响的因素很多,测试约束对测试结果产生内在的、本质的影响,如

需测试的对象、系统运行环境、用户负载等，这些因素的变化作用于效率指标而产生不同的度量结果；其他一些因素作用于测试过程，对测试结果的影响是外在的、非本质的，例如：测试方法、测试工具等。测试约束不考虑作用于测试过程的外在或随机的影响因素。

测试约束典型地包括对象约束、环境约束和负载约束。但效率指标应用框架的测试约束并不局限于对象约束、环境约束和负载约束三种。当某个因素对测试结果产生直接的内在的影响，测试需求要求反映该因素的变化对效率指标度量结果的影响时，可在测试约束中增加该约束。

其他可增加的测试约束的例子包括用户类型约束、访问方式约束、访问数据量约束等。

4.2.2 对象约束

4.2.2.1 对象约束概念

对象约束规定需要进行效率指标度量与评测的对象。例如，在测试时间特性下的“响应时间”度量指标时，需要明确是针对什么处理请求的响应时间；在测试资源利用性特性下的“CPU 利用率”度量指标时，需要明确是针对系统中哪台计算机设备的 CPU。不同的对象的效率指标具有不同的度量结果，不同的对象作用于效率指标可扩展为多个度量项。

4.2.2.2 确定对象约束

每个计算机系统和软件都具有数量庞大的众多对象。在进行效率评价时，不需要对所有对象的效率指标都进行度量和测试，确定需要测试的对象的基本原则是：

——效率测试明确需要测试的对象；

——选择系统的效率需求中明确了效率目标的对象；

——选择系统中典型的具有代表性的对象；

——选择系统中对运行条件(如负载)最为敏感的对象。

不同效率指标所对应的可加以限定的对象类型不同，表 1 描述了每个效率指标对应的对象约束的类型。

表 1 效率指标适应的对象约束类型

特性	子特性	度量指标	适用的对象约束类型
时间	时间效率	响应时间	不同请求
		周转时间	不同事务
	处理效率	吞吐率	不同请求或事务
容量	用户容量	最大并发用户数	不同用户类型
		最大并发请求数	不同请求
	处理容量	事务吞吐容量	不同事务
		数据吞吐容量	不同数据类型
		数据处理容量	不同数据处理或存储功能
资源利用性	CPU 利用性	CPU 利用率	不同 CPU
	内存利用性	内存利用率	不同内存
		内存错误率	
	外存利用性	外存时间利用率	不同外存设备
		外存空间利用率	

表 1（续）

特性	子特性	度量指标	适用的对象约束类型
资源利用性	传输利用性	传输能力利用率	不同传输设备
		传输出错率	
	I/O 设备利用性	I/O 设备利用率	不同 I/O 设备
		I/O 错误率	
		I/O 等待时间	

效率指标的测试对象由高到低可以形成不同的层次，高层对象可分解为低层对象，低层对象可组成高层对象，高层对象可以是若干低层对象组成的综合对象。

一些对象可以直接进行效率指标度量和评测，这些对象为直接对象；另一些对象由其他对象综合而成，综合对象由低层对象组合而成，这些低层对象可以是其他综合对象，也可以是直接对象。综合对象不能直接进行效率指标测试，当需要对综合对象进行效率指标测试时，需要对组成综合对象的低层对象或直接对象进行效率指标测试，或者选择典型的低层对象或直接对象进行效率指标测试，然后对低层对象或直接对象的效率指标度量结果进行加权平均得到综合对象的效率指标度量结果。

例如，Web 网站的页面由各种不同的页面组成，每个页面都可以是一个测试对象，这些页面是直接对象，可直接度量响应时间等效率指标。不同类型的页面可以组成综合对象，如动态页面、静态页面。动态页面和静态页面就是综合对象。如果要测试动态页面的响应时间，应选择若干典型的动态页面，分别测试响应时间，然后计算各个动态页面响应时间平均（或加权平均）值作为动态页面响应时间度量值。原则上说，一个效率指标可综合所有下层对象为一个最高层对象，对最高层对象的效率指标度量可获得被测系统的综合的效率评价。

4.2.3 环境约束

4.2.3.1 环境约束概念

环境约束规定需要在哪些系统运行环境下进行效率指标的度量和测试。这里的系统运行环境包括系统的硬件环境、软件环境、网络环境等，也可称为目标环境。一些效率测试只需要考察一种目标环境下的系统效率，典型地是系统将运行的实际运行环境；另一些效率测试需要考察系统在不同目标环境下的效率，这时环境约束需要规定多种需测试的目标环境。目标环境的变化直接影响效率指标度量结果。反之，任何效率指标的度量值都是针对了某种特定的目标环境。

注：特别需要注意的是目标环境与测试环境的不同，有些场合测试环境使用了实际目标环境或与目标环境相同，另一些测试场合由于条件的限制测试环境不能完全达到目标环境的要求，这时测试环境的误差可能导致测试结果产生一定误差。测试条件导致的测试环境的不同虽然也导致了不同的效率指标度量结果，但由于这种影响是作用于测试过程而不是测试需求，因此通常不认定为环境约束。

4.2.3.2 确定环境约束

确定适用的环境约束需要考虑效率测试的目标，与环境相关的测试目标的例子包括：

——在指定的目标运行环境下，系统的执行效率是否能够满足用户需求；

——需要怎样的系统软硬件配置，才能够满足用户的基本效率需求；

——什么样的系统软硬件配置具有最高性价比等。

通过对测试目标的分析，确定要对哪些不同的目标环境进行测试，并分别在每个不同的目标环境下进行效率指标的测试。

4.2.4 负载约束

4.2.4.1 负载约束概念

负载约束规定需要在什么样的系统负载下进行效率指标的度量和测试。一次效率测试通常需要对多种不同的系统负载进行测试，从而获得不同负载条件下系统和软件效率。

4.2.4.2 确定负载约束

原则上说，效率测试可针对任意可能的负载进行测试，即：可测试获得各种不同负载条件下的效率指标度量值，并可获得随负载变化情况下效率指标的变化情况。在实际系统中，用户通常更关注具有特定意义的负载下的效率指标度量结果，例如，需要测试“在正常情况下，用户网上购票请求的响应时间是多少”，或“在春运开始前 10 天内，用户网上购票请求的响应时间是多少”，“最坏情况下，系统网上购票能够承受的并发用户数是多少，其订票请求的响应时间是多少?”等。

常用的负载约束包括(但不限于)以下 3 种：

——常规负载：常规运行情况下，系统的并发用户数量。

——峰值负载：在特定时期，系统面临大量并发用户的情况下，系统的并发用户数量。在负载量不均衡的系统中，峰值负载也常常用来表示系统所可能面临的用户负载量的最高值。

——极限负载：系统所能够承受的最大负载量，超过该负载量，可能导致系统失效或运行效率急剧下降，并达到用户难以承受的程度。

注：常规负载和峰值负载的负载大小可以在测试之前确定，而极限负载的负载大小在测试之前可能是未知的，因此对极限负载下的效率指标的测试应首先确定极限负载的大小。可以先通过容量特性中的“最大并发用户数”等效率指标得到系统的极限用户负载，再进行极限负载下的效率指标测试，也可以在进行容量测试的同时测试极限负载下的效率指标。

不同效率指标适用的典型负载约束值见表 2。

表 2 效率指标适用的负载约束

特性	子特性	度量指标	负载约束
时间	时间效率	响应时间	常规负载、峰值负载、极限负载
		周转时间	常规负载、峰值负载、极限负载
	处理效率	吞吐率	常规负载、峰值负载、极限负载
容量	用户容量	最大并发用户数	极限负载
		最大并发请求数	极限负载
	处理容量	事务吞吐容量	极限负载
		数据吞吐容量	极限负载
		数据处理容量	常规负载、峰值负载、极限负载
资源利用性	CPU 利用性	CPU 利用率	常规负载、峰值负载、极限负载
	内存利用性	内存利用率	常规负载、峰值负载、极限负载
		内存错误率	常规负载、峰值负载、极限负载
	外存利用性	外存时间利用率	常规负载、峰值负载、极限负载
		外存空间利用率	常规负载、峰值负载、极限负载
	传输利用性	传输能力利用率	常规负载、峰值负载、极限负载

表 2（续）

特性	子特性	度量指标	负载约束
资源利用性	传输利用性	传输出错率	常规负载、峰值负载、极限负载
	I/O 设备利用性	I/O 设备利用率	常规负载、峰值负载、极限负载
		I/O 错误率	常规负载、峰值负载、极限负载
		I/O 等待时间	常规负载、峰值负载、极限负载

4.3 结果视图

4.3.1 综述

本部分中，效率指标度量结果并不是一个具体的值，而是包括了度量结果完整信息的数据集合。展示该度量结果的方式不是唯一的，可以采用不同的展现形式，这些不同的展现形式从不同的角度不同的侧面揭示度量结果的不同特征，可分别满足对度量结果的不同应用需要。效率指标度量结果的不同展现形式称为结果视图。

效率指标结果视图的主要形式包括：

——单值视图；

——统计视图；

——分时视图；

——满足度视图。

效率指标采用什么类型的结果视图与测试需求及测试方法等因素有关。当测试需求或测试方法规定仅经过一次测试尝试，只取得一个测试结果值，则结果视图为单值视图；当测试需求或测试方法规定需经过多次测试尝试，获得了多个测试结果值，结果视图可以是统计视图；当测试需求或测试方法规定按照时间顺序获得测试结果，则结果视图可以是分时视图，同时还可以将分时值进行统计，得到统计视图；如果测试结果需要和用户期望的效率目标进行比较，则可以采用满足度视图。

效率度量可选择一个或多个结果视图来展现。

4.3.2 单值视图

当效率指标的测试结果为单一值时，该度量指标只有唯一的一个单值视图。

例如，“最大并发用户数”“最大并发请求数”效率指标的结果视图为单值视图。

单值视图不可以同时具有分时视图和统计视图，但可以具有相对应的满足度视图（见 4.3.5）。

4.3.3 统计视图

4.3.3.1 统计视图概念

当效率指标的测试结果为一组测试尝试的结果数据，这组结果数据通常呈现为一组统计分布值，则可用该统计分布的特征值来反映测试结果。用统计分布的特征值来反映测试结果的视图称为统计视图。

典型的统计视图包括但不限于以下类型：

——最大：针对一组结果数据，计算最大值；

——平均：针对一组结果数据，计算平均值；

——最小：针对一组结果数据，计算最小值；

——-90%：针对一组结果数据，90%的测试数据小于该值；

——10%：针对一组结果数据，10%的测试数据小于该值；

——方差：针对一组结果数据，计算方差值。

统计视图可以用以下形式表示为：[效率指标](统计视图类型)。

例如：响应时间(平均)、周转时间(最大)。

4.3.3.2 确定统计视图

决定采用哪些结果视图的因素包括：

——对效率指标度量结果需要的深入程度，如果需要全面的和详细的效率指标度量结果，则需要更多的统计视图。

——是否具有对应的效率目标，如果具有明确的效率目标，则需要具有相同形式的统计视图并由此计算满足度。

4.3.4 分时视图

当效率指标的测试结果为一组按照时间顺序进行采样或测试得到的结果数据，则可以用分时视图来表示测试结果，分时视图是按照规定的时间间隔来展示测试结果的变化情况的结果视图。

分时视图可提供测试过程中的效率指标的变化情况，特别适用于需要进行系统运行状态分析、效率瓶颈分析等测试需求。

分时视图可分为以下两种形式：

——分时采样：按照固定的时间间隔进行采样，获得按照时间序列排列的测试结果序列。

分时采样度量公式：

[效率指标名](t_i)＝t_i 时刻采样或测试获得的测量值。

式中：

$t_i = T_s \cdots T_e$，T_s 为测试开始时间，T_e 为测试结束时间；

——分时平均：按照固定时间间隔对一组测试结果数据进行分组统计，计算每个时间单位分组内的结果数据平均值。

分时平均度量公式：

[效率指标名](t_i)＝AVG(在[$t_i \cdots t_{i-1}$)区间所有测试尝试的测试结果数据)

式中：

$t_i = T_s \cdots T_e$；T_s 为测试开始时间，T_e 为测试结束时间。

分时视图可同时具有统计视图，如果将分时视图的结果序列当作一组统计分布，则可以计算获得对应的统计视图。

分时视图特别适合于采用图形的方式展示，因此分时视图的展现形式往往不是数据序列而是图表。

4.3.5 满足度视图

4.3.5.1 满足度视图的概念

效率指标的单值视图、统计视图和分时视图都是对系统本身的效率特性进行度量，而未考虑效率是否满足用户期望的效率目标。如果需要考察效率水平是否满足用户的期望的效率目标，则可采用满足度视图。

满足度视图是将效率指标的测试结果与用户期望的效率指标目标值相比较，从而提供相关效率指标在多大程度上满足用户效率需求的信息。

应用满足度视图的前提是需要提供效率指标的期望值。如果不能获得效率指标的期望值，则无法采用满足度视图。

通常待测系统只在部分效率指标上具有明确的效率目标需求，因此可在部分效率指标上采用满足度视图。如果待测试的系统未提出任何效率目标需求，则无需采用满足度视图。

在使用满足度视图时，可以将指标名称修改为：[效率指标名]满足度。

例如，针对以下响应时间效率指标的效率目标：

—— 网上订票系统的票务查询请求响应时间平均小于 30 s；

——网上订票系统的订票请求响应时间最大不超过 2 min；

——网上订票系统 90%的订票请求响应时间不超过 1 min。

可以分别使用以下满足度视图：响应时间满足度(平均)、响应时间满足度(最大)、响应时间满足度(90%)。

4.3.5.2 满足度计算公式

效率指标满足度的值反映效率指标满足用户效率目标的程度，满足度计算公式为：

效率指标满足度＝效率指标测试值/效率指标期望值

满足度取值的含义根据不同的效率目标的形式有所不同：

a) 当效率目标形式为“测试值小于效率指标期望值”，则满足度小于 1 时满足用户期望，且越小越好。

例：效率目标为“某请求响应时间平均不高于 A。”则响应时间满足度(平均)＝响应时间(平均)/期望的平均响应时间 A。

当响应时间满足度(平均)小于 1 时，表示目标被满足，且越小越好；

b) 当效率目标形式为“测试值大于效率指标期望值”，则满足度大于 1 时满足用户期望，且越大越好。

例：效率目标为“某事务吞吐率最少不低于 B”。则吞吐率满足度(最小)＝吞吐率(最小)/期望的最小吞吐率 B。

当吞吐率满足度(最小)大于 1 时，表示目标被满足，且越大越好。

注：如果需要对系统和软件的效率进行整体评价，则需要对各个指标的效率满足度进行汇总，这就需要对效率满足度进行归一化处理。归一化保证满足度取值具有相同的意义，方便不同指标之间进行满足度汇总和比较。

4.4 效率指标测试流程

效率指标的度量结果可通过效率指标测试取得。面向效率指标体系应用框架的效率指标测试方法包括以下主要步骤：

a) 选取效率指标：根据测试需求从效率指标体系中选取需要测试的系统和软件的效率特性及子特性，并确定要测试的效率指标；

b) 确定测试约束：根据效率测试需求确定每种效率指标所适用的测试约束，包括对象约束、环境约束、负载约束以及其他适用的测试约束，确定每种测试约束中需要测试的约束值；

c) 确定结果视图：根据效率测试需求或效率目标所适用的形式确定测试结果适用的结果视图；

d) 设计测试用例：在满足效率指标、测试约束要求的基础上设计效率测试的测试环境、测试场景、测试工具、测试过程、数据收集和统计方法等，要求所设计的测试用例应覆盖以上所有需要测试的效率指标及其对应的测试约束的组合；数据收集和统计的方法应考虑能够取得结果视图所需要的数据；

e) 执行效率测试：按照测试用例规定的要求、方法及过程执行效率测试，监控测试运行，获取测试结果；

f) 结果计算与展现：按照所需的结果视图计算并展示效率指标度量结果。

基于效率指标体系应用框架的效率指标测试流程见图 1。

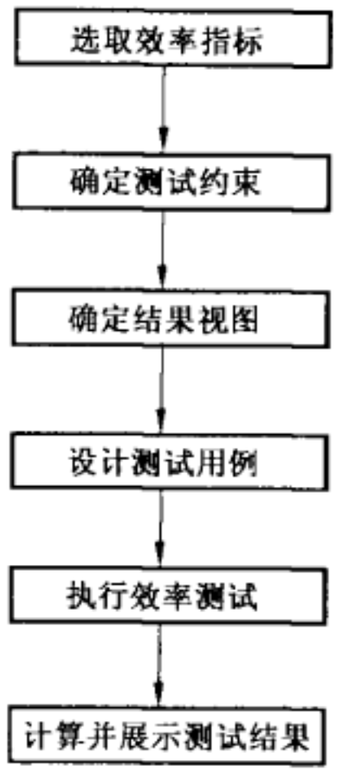

图 1　基于效率指标体系应用框架的效率指标测试流程

本部分第 5 章～第 7 章描述了每个效率指标的具体测试方法，按照效率指标测试流程，每个效率指标分以下条目加以描述：

——指标名称：效率指标名称；

——测试说明：描述效率指标测试方法的总体说明；

——测试约束：描述适用的测试约束；

注：这里列出的测试约束为常用的可选项，不同测试项目可以按照测试需求进行剪裁(详见附录 A)。

——测试设计：按照效率指标及其测试约束要求设计效率测试用例；

——测试实施：描述效率指标测试执行的主要步骤；

——测试结果计算：描述适用的结果视图以及每个结果视图的计算方法。

注：本部分给出了效率指标的测试原理和计算方法，当使用效率测试工具进行测试时，可结合测试工具的功能完成效率测试和结果计算。

5　时间特性

5.1　时间效率

5.1.1　响应时间

指标名称		响应时间
测试说明		响应时间的测试是通过用户(或虚拟用户)向服务器发出并发请求，测量服务器响应这些请求所需的时间
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——效率需求中明确对响应时间具有效率需求的操作请求； ——应用系统中典型的操作请求； ——应用系统中可能面临较大负载压力的操作请求

指标名称	响应时间
测试设计	1. 确定测试环境，按照环境约束设计测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括对象约束中要求的操作请求，一次测试的测试场景中可包括多个操作请求，如果一个测试不能包括全部要测试的操作请求，可进行多次测试。 3. 确定测试负载，按照负载约束确定要达到的负载，一次测试面向一种负载，当要求测试多种不同负载下的响应时间时，则分别进行测试。 注 1：响应时间测试也可以在一次测试中不断增加或减少系统负载，并获得响应时间随负载变化情况发生的变化的信息。 注 2：如果要测试极限条件下的响应时间，应与容量测试相结合，首先获得极限负载的并发用户数，再测试极限负载下的响应时间。
测试实施	1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 ——自动化效率测试可通过运行测试场景启动负载发生器发出并发请求； ——自编制测试程序可在程序中启动对服务器发出并发请求； ——人工测试通过组织大量测试人员或用户共同访问被测系统。 2. 监控测试过程，监控内容： ——负载增加、保持、下降的过程和达到的负载量，确定是否达到测试规划目标； ——测试过程异常终止，未能完成规定的测试； ——请求是否正常完成，结果是否正确，记录出错数量。 3. 收集测试数据：收集每个请求的发出时间 START 和完成时间 END。 ——自动化测试工具可自动收集测试结果； ——自编制测试程序可在每个请求的前后增加时间戳，计算两者的差得到响应时间； ——人工测试可采用秒表等计时工具收集每个请求从发出到收到结果所经过的时间。 注 1：由于人工操作的限制，人工测试不必收集全部的请求响应时间，可以让部分用户仅用于产生负载，部分用户在产生负载的同时收集所需的请求响应时间。 注 2：由于人工测试不能保证测试的精度，通常情况下，不建议采用人工测试方法。
测试结果计算	1. 计算测试周期内每个请求的响应时间 X_i。 $X_i = \mathrm{END}_i - \mathrm{START}_i, i = 1 \cdots N$ 2. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 3. 按照测试需求及效率目标，计算需要的结果视图 a) 计算统计视图： 响应时间(平均) $X_{avg} = \mathrm{AVG}(X_i)$，其中：$X_i$ 满足 START_i IN $[\mathrm{VT}_s .. \mathrm{VT}_e]$ 响应时间(最大) $X_{max} = \mathrm{MAX}(X_i)$，其中：$X_i$ 满足 START_i IN $[\mathrm{VT}_s .. \mathrm{VT}_e]$ b) 必要时，计算满足度视图 响应时间满足度(平均) = X_{avg}/期望的平均响应时间响应时间满足度(最大) = X_{max}/期望的最大响应时间 c) 必要时，计算分时视图，分时响应时间 $X(t_i)$ 为在 t_i 单位时间发出的请求的响应时间的平均值 响应时间 $X(t_i) = \mathrm{AVG}(X_k)$ 式中：$t_i = \mathrm{VT}_s \cdots \mathrm{VT}_e - 1$，$X_k$ 满足 START_k IN $[t_i, t_{i+1})$ 注：分时响应时间 $X(t_i)$ 也可以定义为在 t_i 单位时间完成的请求的平均响应时间。

5.1.2 周转时间

指标名称		周转时间
测试说明		1. 周转时间包括整个事务处理的交互过程，其中包括了用户操作所占用的时间和客户端程序的处理时间。在自动化测试中，可通过设置思考时间来模拟这些所需要的时间。 2. 在自动化效率测试工具中，事务是通过在测试脚本中设置事务开始与事务结束标志来定义的
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——效率需求中明确对事务周转时间具有效率需求的事务； ——应用系统中典型的事务； ——应用系统中可能面临较大负载压力的事务
测试设计		1. 确定测试环境，按照环境约束准备测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括对象约束中要求的事务。 3. 确定测试负载，一次测试面向一种系统负载，当要求测试多种不同负载下的周转时间时，则分别进行测试 **注**：如果测试极限条件下的周转时间，应与容量测试相结合。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，执行事务操作。 ——自动化效率测试可通过运行测试场景启动负载发生器产生事务处理操作请求； ——自编制测试程序可在程序中模拟事务向服务器发出事务操作请求； ——人工测试通过组织大量测试人员或用户模拟实际使用情况进行事务操作。 **注**：事务的处理过程中需要考虑适当的思考时间，从而与实际用户的请求的密度相类似。 2. 监控测试过程，监控内容： ——负载增加、保持、下降的过程和达到的负载量，确定是否达到测试规划目标； ——测试过程异常终止，未能完成规定的测试； ——事务是否正常完成，结果是否正确，记录出错数量。 3. 收集测试数据：收集每个事务的开始时间 START 和结束时间 END。 ——自动化测试工具应标识每个事务的开始位置和结束位置，测试工具可自动收集每个事务的周转时间； ——自编制测试程序可在每个事务的前后增加时间戳，计算两者的差得到周转时间； ——人工测试可采用秒表等计时工具收集每个事务从事务开始到事务结束所经过的时间。 **注 1**：由于人工操作的限制，人工测试不必收集全部的周转时间，可以让部分用户仅用于产生负载，部分用户在产生负载的同时收集所需的周转时间。 **注 2**：由于人工测试不能保证测试的精度，通常情况下，不建议采用人工测试方法。

指标名称	周转时间
测试结果计算	1. 计算测试周期内每个事务的周转时间 X_i。 $X_i = \mathrm{END}_i - \mathrm{START}_i, i = 1 \cdots N$ 2. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 3. 按照测试需求及效率目标，计算需要的结果视图。 a) 计算统计视图： 周转时间(平均)$X_{avg} = \mathrm{AVG}(X_i)$，其中：$X_i$ 满足 START_i IN $[\mathrm{VT}_s \cdots \mathrm{VT}_e]$ 周转时间(最大)$X_{max} = \mathrm{MAX}(X_i)$，其中：$X_i$ 满足 START_i IN $[\mathrm{VT}_s \cdots \mathrm{VT}_e]$ b) 必要时，计算满足度视图 周转时间满足度(平均)$= X_{avg}$/期望的平均周转时间 周转时间满足度(最大)$= X_{max}$/期望的最大周转时间 c) 必要时，计算分时视图： 分时周转时间 $X(t_i)$为在 t_i 单位时间开始的事务的周转时间的平均值 周转时间 $X(t_i) = \mathrm{AVG}(X_k)$ 式中：$t_i = \mathrm{VT}_s \cdots \mathrm{VT}_e$，$X_k$ 满足 START_k IN $[t_i, t_{i+1})$ 注：分时周转时间 $X(t_i)$也可以定义为在 t_i 单位时间完成的事务的平均周转时间。

5.2 处理效率

5.2.1 吞吐率

指标名称		吞吐率
测试说明		执行一批并发任务，测量完成这些任务所需要的时间，并计算单位时间内所完成的任务数。为了去除随机测试的误差，可进行多次测试尝试，获得吞吐率分布数据
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——效率需求中明确对吞吐率具有效率目标的请求、事务或数据处理等； ——应用系统中典型的请求、事务或数据处理等； ——应用系统中可能面临较大负载压力的请求、事务或数据处理等
测试设计		1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括对象约束中要求的请求和/或事务。 3. 确定测试负载 吞吐率测试可一次测试面向一种负载，不同负载下的吞吐率测试可通过多次测试获得。 注：吞吐率测试也可以在一次测试中不断增加或减少系统负载，并获得吞吐率随负载变化情况发生的变化情况的信息。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，执行事务操作。 2. 监控测试过程，监控内容： ——负载增加、保持、下降的过程和达到的负载量，确定是否达到测试规划目标；

指标名称	吞吐率
测试实施	——测试过程异常终止，未能完成规定的测试； ——事务是否正常完成，结果是否正确，记录出错数量。 3. 收集测试数据： ——收集每次测试中完成的任务个数 NUM_i，所使用的时间 T_i； 注：任务个数可以是响应请求数、完成事务数或处理数据量。 ——当需要测试分时吞吐率时，应收集每个单位时间 t_i 内所完成的任务个数 NUM_n
测试结果计算	按照测试需求及效率目标，计算需要的结果视图 a) 计算每次测试的吞吐率 X_i $X_i = Num_i / T_i$ b) 计算平均吞吐率及最大吞吐率： 吞吐率(平均) $X_{avg} = AVG(X_i)$，式中 X_i IN $[VT_s \cdots VT_e]$ 吞吐率(最小) $X_{min} = MIN(X_i)$，式中 X_i IN $[VT_s \cdots VT_e]$ c) 计算满足度视图 吞吐率满足度(平均)＝X_{avg}/期望的平均吞吐率 吞吐率满足度(最小)＝X_{min}/期望的最小吞吐率 d) 必要时，计算分时吞吐率 分时吞吐率 $X(t_i) = NUM_n$ 其中 NUM_n 为单位时间 t_i 到 t_{i+1} 时间内完成的任务数

6 容量

6.1 用户容量

6.1.1 最大并发用户数

指标名称		最大并发用户数
测试说明		1. 最大并发用户数的测试是通过对一个逐渐增大的并发用户数的序列来进行测试尝试，并获得每次尝试的请求响应时间，通过分析请求响应时间在什么时候突然显著增加或变得不可接受，从而得到系统可承受的最大并发用户数。 2. 系统负载的实际大小与并发用户数及用户使用方式(事务类型、操作的速度、频率等)相关。因此，考虑并发用户数的使用方式是必要的。缺省情况下，本指标最大并发用户数的含义是：在系统实际运行方式下的并发用户，具有常规的使用方式和平均的操作速度及思考时间。 3. 如果要在测试中使并发用户数等于实际系统的并发用户数，将可能导致测试工具的并发用户授权权限不够，可以采用增加操作频度来减少所需的虚拟并发用户的数量
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——极限负载。 注：由于极限负载也就是本指标的测试目标，因此也可以认为本指标无需负载约束。
	对象约束	适用的对象约束： ——效率需求中明确指定的用户类型或操作类型； ——系统实际运行过程中典型的用户类型及其典型的操作类型

指标名称	最大并发用户数
测试设计	1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括对象约束中规定的用户类型和操作类型。 3. 设计要尝试的并发用户数序列 $Nu[1]$，$Nu[2]$，…，$Nu[N]$， 式中：$Nu[1]=N_0$，$Nu[2]=Nu[1]+a$，…$Nu[i]=Nu[i-1]+a$；$i=2\cdots N$； N_0 为尝试测试的并发用户数最小值，a 为每次尝试增加的并发用户数。 注 1：尝试测试的并发用户数范围应恰当，其中应该包括估计的最大并发用户数，如果测试发现最大并发用户数不在其中，则调整并发用户数范围保证能够测试获得最大并发用户数。 注 2：应注意每次测试的用户数增量 a 应适当，a 太大将不能保证最大并发操作用户数的测试结果的精度在可接受的范围之内，a 太小将导致测试次数的增加，从而增加测试成本。
测试实施	1. 针对每个需要尝试的并发用户数 $Nu[i]$，执行并发测试。 模拟具有 $Nu[i]$个用户同时在线执行规定的事务或请求。 注：每个事务的请求之间需要考虑适当的思考时间，从而与实际用户的请求的密度相类似。 2. 针对每个 $Nu[i]$，测试响应时间 $Tu[i]$或吞吐率 $Pu[i]$。 注：响应时间及吞吐率的测试方法见 5.1.1 和 5.2.1。 3. 监控测试结果，如果出现以下情况，可停止测试。 ——测试过程出现异常，测试无法正常完成； ——响应时间达到超过用户可容忍的范围； 4. 如果测试完 $Nu[N]$仍然未得到最大并发用户数，应增加尝试次数或重新调整尝试范围 $Nu[1]\cdots Nu[N]$与递增量 a
测试结果计算	1. 对每次尝试得到的结果$\{Nu[i],Tu[i],Pu[i]\}$，以并发用户数为横坐标，响应时间或吞吐率为纵坐标作图，获得响应时间 $Tu[i]$和 $Pu[i]$随并发请求数 $Nu[i]$变化的关系图。 2. 选择并发用户数 $Nu[j]$. 满足： $Tu[1]\cdots Tu[j]$持平滑上升趋势，$Tu[j+1]\cdots Tu[N]$出现急剧上升趋势 或 $T[j+1]$超过了用户能够容忍的响应时间的极限 或：$Pu[1]\cdots Pu[j]$持平滑上升趋势，$Pu[j+1]\cdots Pu[N]$不再上升，出现处理瓶颈 则 $Nu[j]$为最大并发用户数。 3. 必要时，计算最大并发用户数满足度视图 最大并发用户数满足度＝最大并发用户数/期望的最大并发用户数

6.1.2 最大并发请求数

指标名称		最大并发请求数
测试说明		1. 最大并发请求数的测试考虑并发环境中的一种极限情况，即所有的在线用户都是同时或连续发出用户请求，这时最大并发请求数的测试也可以看作是这种情况下的并发用户数的测试。 2. 最大并发请求数的测试不需要考虑用户在发出请求之间的思考时间。用户可以连续地不间断地发送服务请求。在这种情况下，可以通过不太多的虚拟用户尽快地测试到系统的极限负载。通过这种方式测试得到的并发用户数并不是实际环境中真正的客户端能够接受的并发用户数，它应低于实际的客户端并发用户数(见“最大并发用户数”测试方法)。 3. 最大并发请求数的测试是通过对一个逐渐增大的并发请求数的序列来进行测试尝试，并获得每次尝试的请求响应时间，通过分析请求响应时间在什么时候突然显著增加或变得不可接受，从而得到系统可承受的最大并发请求数
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境

指标名称		最大并发请求数
测试约束	负载约束	适用的负载约束： ——极限负载。 注：由于极限负载也就是本指标的测试目标，因此也可以认为本指标无需负载约束。
	对象约束	适用的对象约束： ——效率需求中明确对最大并发请求数具有需求的操作请求； ——应用系统中可能面临较大负载压力的并发请求
测试设计		1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括要测试最大并发请求数的请求。 3. 设计要尝试的并发请求数序列 $Nr[1], Nr[2], \cdots, Nr[N]$， 式中：$Nr[1]=N_0$，$Nr[2]=Nr[1]+a$，$\cdots Nr[i]=Nr[i-1]+a$；$i=2\cdots N$； N_0 为尝试测试的并发请求数最小值，a 为每次尝试增加的并发用户数
测试实施		1. 针对每个需要尝试的并发请求数 $Nr[i]$，执行并发测试。 根据测试条件、测试工具、测试方式的不同，可以选择以下负载生成方式 ——一个用户一个时刻只发出一个请求。 ——一个用户可同时发出若干个请求，例如通过启动多个线程同时发出多个请求。 ——一个用户连续发出多个请求，例如通过循环语句。 注：最大并发请求数的测试无需考虑思考时间，如果是自动化测试，可在测试脚本中增加集合点产生同时发出的并发请求。 2. 针对每个 $Nr[i]$，测试并发请求的响应时间 $Tr[i]$或吞吐率 $Pr[i]$。 注：响应时间及吞吐率的测试方法详见“5.1.1 响应时间”和“5.2.1 吞吐率”。 3. 监控测试结果，如果出现以下情况，可停止测试。 ——测试过程出现异常，测试无法正常完成； ——响应时间变得不可测试。 4. 如果测试完 $Nr[N]$仍然未得到最大并发请求数，应增加尝试次数或重新调整尝试范围 $Nr[1]\cdots Nr[N]$与递增量 a
测试结果计算		1. 对每次尝试得到的结果$(Nr[i], Tr[i], Pr[i])$，以并发请求数为横坐标，响应时间，吞吐率为纵坐标作图，获得响应时间 $Tr[i]$和吞吐率 $Pr[i]$随并发请求数 $Nr[i]$变化的关系图。 2. 选择并发请求数 $Nr[j]$，满足： $Tr[1]\cdots Tr[j]$持平滑上升趋势，$Tr[j+1]\cdots Tr[N]$出现急剧上升趋势　或 $Tr[j+1]$超过了用户能够容忍的响应时间的极限 或：$Pu[1]\cdots Pu[j]$持平滑上升趋势，$Pu[j+1]\cdots Pu[N]$不再上升，出现处理瓶颈 则 $Nr[j]$为最大并发请求数。 3. 必要时，计算最大并发请求数满足度视图 最大并发请求满足度＝最大并发请求数/期望的最大并发请求数

6.2 处理容量

6.2.1 事务吞吐容量

指标名称	事务吞吐容量
测试说明	1. 事务吞吐容量用于表示一定时间内系统能够处理最大事务数，表达系统在事务处理上的最大能力，事务吞吐容量的核心是确定系统的极限负载。

指标名称		事务吞吐容量
测试说明		2. 事务吞吐容量的测试是通过对一个逐渐增大的并发用户数的序列来进行测试尝试，并获得每次尝试的事务吞吐率，通过分析事务吞吐率在什么时候不再增加，从而得到系统可承受的最大事务吞吐率，并进过时间换算得到一定时间的事务吞吐容量。 3. 应考虑保证事务吞吐容量测试中每次测试尝试的持续时间，从而考察在较长的的时间周期内系统总体的事务处理能力。 注：由于事务吞吐容量测试的是服务器端的极限负载，因此可不考虑客户端请求的特征，在测试事务吞吐容量时可减少或忽略用户的思考时间。
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——极限负载 注：由于极限负载也就是本指标的测试目标，因此也可以认为本指标无需负载约束。
	对象约束	适用的对象约束： ——效率需求中明确对事务吞吐容量具有效率需求的事务； ——应用系统中典型的事务； ——应用系统中可能面临较大负载压力的事务
测试设计		1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括对象约束中要求的事务。 3. 设计要尝试的并发用户数序列 $Nu[1],Nu[2],\cdots,Nu[N]$， 式中：$Nu[1]=N_0$，$Nu[2]=Nu[1]+a$，$\cdots Nu[i]=Nu[i-1]+a$；$i=2\cdots N$； N_0 为尝试测试的并发用户数最小值，a 为每次尝试增加的并发用户数 注：这里的并发用户数是指减少或忽略思考时间的并发用户数，因此应比考虑思考时间的实际的并发用户数要少得多。
测试实施		1. 针对每个需要尝试的并发用户数 $Nu[i]$，执行并发测试，模拟具有 $Nu[i]$个用户并发执行指定的事务。 2. 针对每个 $Nu[i]$，测试事务处理的吞吐率 $Pu[i]$。 注：测试吞吐率的方法见 5.2.1。 3. 监控测试结果，如果出现以下情况，可停止测试。 ——测试过程出现异常，测试无法正常完成； ——事务不能正常完成并获得正确处理结果。 4. 如果测试完 $Nu[N]$仍然未得到最大事务容量，应增加尝试次数或重新调整尝试范围与递增量
测试结果计算		1. 对每次尝试得到的结果$\{Nu[i],Pu[i]\}$，以并发用户数为横坐标，吞吐率为纵坐标作图，获得吞吐率 $Pu[i]$随并发用户数 $Nu[i]$变化的关系图。 2. 选择最大事务吞吐率 $Pu[j]$，满足： ——$Pu[1]\cdots Pu[j]$持平滑上升趋势并达到最高，$Pu[j+1]\cdots Pu[N]$不再呈现上升趋势； ——则 $Pu[j]$为最大事务吞吐率，即事务吞吐容量。 3. 必要时，计算事务吞吐容量满足需求的程度 事务吞吐容量满足度＝事务吞吐容量/期望的事务吞吐容量

6.2.2 数据吞吐容量

指标名称		数据吞吐容量
测试说明		1. 数据吞吐容量的测试是通过对一个逐渐增大的数据处理请求量来进行测试尝试，通过测试获得每次尝试的系统实际的数据吞吐率，通过分析数据吞吐率在什么时候不再增加，从而得到系统可承受的最大数据吞吐率，并经过时间换算得到一定时间的数据吞吐容量。 注：数据处理请求量是单位时间内用户发出的数据处理请求。 2. 增加数据请求量可通过以下方式达到： ——增加并发用户数； ——增加单位时间内并发用户数发出的事务数； ——增加事务中要求处理的数据量。 注：由于数据吞吐容量测试的是服务器端的极限负载，因此可不考虑客户端请求的特征，在测试数据吞吐容量时用户可连续产生数据处理请求。
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——极限负载 注：由于极限负载也就是本指标的测试目标，因此也可以认为本指标无需负载约束。
	对象约束	适用的对象约束： ——效率需求中需要测试数据吞吐容量相关的数据类型
测试设计		1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。 2. 设计测试场景，测试场景中应包括要求测试的数据处理事务。 3. 设计要尝试的并发用户数序列 $Nu[1], Nu[2], \cdots, Nu[N]$， 其中：$Nu[1]=N_0$，$Nu[2]=Nu[1]+a$，$\cdots Nu[i]=Nu[i-1]+a$；$i=2\cdots N$； N_0 为尝试测试的并发用户数最小值，a 为每次尝试增加的并发用户数 注：这里假设增加数据请求量的方法是增加并发用户数，每个并发用户连续产生一定数量的数据处理请求。如果通过增加事务数或增加数据量来增加数据请求量，应相应调整需尝试的数据序列。
测试实施		1. 针对每个需要尝试的并发用户数 $Nu[i]$，执行并发测试，每个用户可循环发出若干个数据处理请求。 2. 针对每个 $Nu[i]$，测试数据处理的吞吐率 $Pd[i]$。 注：测试数据吞吐率的方法见 5.2.1。 3. 监控测试结果，如果出现以下情况，可停止测试。 ——测试过程出现异常，测试无法正常完成； ——数据处理功能不能正常完成，并获得正确处理结果。 4. 如果测试 $Nu[N]$仍然未得到最大数据吞吐率，应增加尝试次数或重新调整尝试范围与递增量
测试结果计算		1. 对每次尝试得到的结果$\{Nu[i], Pd[i]\}$。以并发用户数为横坐标，数据吞吐率为纵坐标作图，获得数据吞吐率 $Pd[i]$随并发用户数 $Nu[i]$变化的关系图。 2. 选择最大数据吞吐率 $Pd[j]$，满足 $Pd[1]\cdots Pd[j]$持平滑上升趋势并达到最高，$Pd[j+1]\cdots Pd[N]$不再呈现上升趋势 则 $Pd[j]$为最大数据吞吐率。 3. 必要时，计算数据吞吐容量满足度视图 数据吞吐容量满足度＝数据吞吐容量/期望的数据吞吐容量

6.2.3 数据处理容量

<table>
<tr><th colspan="2">指标名称</th><th>数据处理容量</th></tr>
<tr><td colspan="2">测试说明</td><td>1. 造成数据处理量限制的原因根据不同的数据处理类型而不同，可能的原因包括：a)系统环境限制、b)程序设计限制、c)并发用户分享资源等，应在了解其发生原因的基础上确定测试方法。
注：在本部分中，由于无法针对具体原因，因此采用数据量尝试的方法进行测试。
2. 数据处理容量的测试是通过对一个逐渐增大的数据量序列来进行测试尝试，通过对该数据量的处理、存储或访问，检查数据量在什么时候变得不可接受，从而得到系统可承受的最大数据处理容量。
3. 最大数据处理容量的测试可以在不同的负载条件下进行，从而考察负载变化对数据处理容量的影响</td></tr>
<tr><td rowspan="3">测试约束</td><td>环境约束</td><td>适用的环境约束：
——系统规定的目标运行环境；
——其他需要进行效率测试的系统运行环境</td></tr>
<tr><td>负载约束</td><td>适用的负载约束：
——单一用户：测试单一用户下指定数据的处理容量；
——常规负载：测试在常规负载条件下的数据处理容量；
——峰值负载：测试在峰值负载条件下的数据处理容量；
——极限负载：测试在极限负载条件下的数据处理容量</td></tr>
<tr><td>对象约束</td><td>适用的对象约束：
——效率需求中需要测试数据处理容量的数据类型或存储空间</td></tr>
<tr><td colspan="2">测试设计</td><td>1. 确定测试环境，按照环境约束确定测试环境。一次测试面向一种测试环境，如果要求多种不同的测试环境，则分别进行多次测试。
2. 设计测试场景，测试场景中包括对需要测试的数据的处理、访问、存储功能。
3. 确定系统负载，如果是单一用户，则不考虑多用户并发，如果考虑并发负载环境，则对分别测试知道负载约束条件下的数据处理容量。
4. 设计要尝试的数据量序列 $D[1],D[2],\cdots,D[N]$。
其中：$D[1]=D_0$，$D[2]=D[1]+a$，$\cdots D[i]=D[i-1]+a$；$i=2\cdots N$；
D_0 为尝试测试的数据量最小值，a 为每次尝试增加的数据量。
注：不同负载下的数据处理容量可能不同，因此，应采用不同的尝试数据量序列。</td></tr>
<tr><td colspan="2">测试实施</td><td>1. 按照规定的测试环境、测试场景和负载启动并发测试。
2. 针对每个 $D[i]$，测试数据处理、访问或存储操作的平均响应时间或周转时间 $T[i]$。
3. 监控测试结果，如果出现以下情况，可停止测试。
——测试过程出现异常，测试无法正常完成；
——数据处理、访问、存储功能不能正确完成。
4. 如果测试完 $D[N]$仍然未得到最大数据处理量，应增加尝试次数或重新调整尝试范围与递增量</td></tr>
<tr><td colspan="2">测试结果计算</td><td>1. 对每次尝试得到的结果$\langle D[i],T[i]\rangle$作图。获得数据量与响应时间或周转时间的关系图。
2. 选择 $D[j]$，满足：
$T[1]\cdots T[j]$持平滑上升趋势，$T[j+1]\cdots T[N]$出现急剧上升趋势。
或 $T[j]$用户可接受，$T[j+1]$超过了用户能够容忍的响应时间或周转时间的极限。
则 $D[j]$为最大数据存储量，即数据处理容量。
3. 必要时，计算数据处理容量满足需求的程度
数据处理容量满足度＝数据处理容量/期望的数据处理容量</td></tr>
</table>

7 资源利用性

7.1 CPU 利用性

7.1.1 CPU 利用率

指标名称		CPU 利用率
测试说明		1. CPU 利用率可反映在指定的测试约束条件下,CPU 资源对系统运行效率造成的影响以及 CPU 资源是否得到充分利用。 2. CPU 利用率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试。 3. CPU 利用率的数据可通过操作系统性能计数器获取
测试约束	环境约束	适用的环境约束: ——系统规定的目标运行环境; ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束: ——常规负载; ——峰值负载; ——极限负载
	对象约束	适用的对象约束: ——系统配置中负载压力较大的服务器 CPU; ——其他需要测试的 CPU
测试设计		1. 按照测试约束规定,确定所需的测试环境、测试场景和系统负载。 注:CPU 利用率的测试通常在测试时间特性及容量特性时同时进行,因此只需要在测试这些特性时同时考虑本指标的收集即可。 2. 针对要测试的系统设备及相关资源,规划监控和数据收集设施 注:收集资源的设施包括:操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统,向服务器发出大量并发请求。 注:如果在测试时间特性指标和容量指标的同时测试本指标,无需专门启动本测试,可与处理效率指标和容量指标的同时测试。 2. 收集测试数据:收集每个单位时间内 CPU 执行非空指令的时间 注:CPU 执行非空指令的时间可通过操作系统提供的相关性能计数器值取得。
测试结果计算		1. 识别有效测试周期,设 VT_s 为有效测试周期开始时间,VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时 CPU 利用率 CPU 利用率 $X(t_i)=t_i$ 时刻的平均 CPU 利用率,$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内的 CPU 利用率 CPU 利用率(平均)$X_{avg}=AVG(X(t_i))$,$t_i=VT_s\cdots VT_e$ CPU 利用率(最大)$X_{max}=MAX(X(t_i))$,$t_i=VT_s\cdots VT_e$ 4. 必要时,分析 CPU 利用率的变化情况,画出分时分析图 ——有效测试周期内(或测试周期内)CPU 利用率的分时图{t_i,CPU 利用率(t_i)}; ——有效测试周期内(或测试周期内)CPU 利用率与其他分时效率指标(响应时间、周转时间、吞吐率)的同步分时图{t_i,CPU 利用率(t_i),…}

7.2 内存利用性

7.2.1 内存利用率

指标名称		内存利用率
测试说明		1. 内存利用率可反映在指定的测试约束条件下，内存资源对系统运行效率造成的影响以及内存资源是否得到充分利用。 2. 内存利用率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试。 3. 内存利用率的数据可通过操作系统性能计数器获取
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——系统配置中负载压力较大的计算机内存； ——其他需要测试的计算机内存
测试设计		1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 注：内存利用率的测试通常在测试时间特性及容量特性时同时进行，因此只需要在测试这些特性时同时考虑本指标的收集即可。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 2. 收集测试数据：在每个单位时间，收集已使用内存大小以及总的内存容量，或直接收集每个时间单位的已使用内存百分比 注：已使用内存大小或已使用内存百分比可通过操作系统提供的相关性能计数器值取得。
测试结果计算		1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时内存利用率 内存利用率 $X(t_i)=t_i$ 时刻内存利用率，$t_i=VT_s,\cdots VT_e$。 3. 计算有效测试周期内的内存利用率 内存利用率(平均)$X_{avg}=AVG(内存利用率(t_i))$，$t_i=VT_s,\cdots VT_e$ 内存利用率(最大)$X_{max}=MAX(内存利用率(t_i))$，$t_i=VT_s,\cdots VT_e$ 4. 必要时，分析内存利用率的变化情况，画出分时分析图 ——测试周期内(或有效测试周期内)内存利用率的分时图$\{t_i,X(t_i)\}$； ——测试周期内(或有效测试周期内)分时内存利用率与其他分时效率指标(响应时间、周转时间、吞吐率)的同步分时图$\{t_i,X(t_i),\cdots\}$

7.2.2 内存错误率

指标名称	内存错误率
测试说明	1. 内存错误是指需访问的内存数据不在物理内存中，需要从虚拟内存(即外存)交换到物理内存的情况，内存错误率的高低可反映内存资源是否充分或虚拟内存的参数设置是否合理。 2. 内存错误率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试。 3. 内存错误率的数据可通过操作系统性能计数器获取

指标名称		内存错误率
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——系统配置中负载压力较大的计算机内存； ——其他需要测试的计算机内存
测试设计		1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 **注**：内存错误率的测试通常在测试时间特性及容量特性时同时进行，因此只需要在测试这些特性时同时考虑本指标的收集即可。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设 **注**：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 2. 收集测试数据：在每个单位时间，收集所发生的内存错误数或内存错误率 **注**：内存错误数或内存错误率可通过操作系统提供的相关性能计数器值取得。
测试结果计算		1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时内存错误率 内存错误率(t_i)=t_i 时刻内存错误率，$t_i=VT_s,\cdots VT_e$ 3. 计算有效测试周期内的内存错误率 内存错误率(平均)=AVG(内存错误率(t_i))，$t_i=VT_s,\cdots VT_e$ 内存错误率(最大)=MAX(内存错误率(t_i))，$t_i=VT_s,\cdots VT_e$

7.3 外存利用性

7.3.1 外存时间利用率

指标名称		外存时间利用率
测试说明		1. 外存时间利用率可反映在指定的测试约束条件下，外存的读写时间对系统运行效率所造成的影响。 2. 外存时间利用率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试。 3. 外存时间利用率的数据可通过操作系统性能计数器获取
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——需要测试的计算机的外部存储设备，如指定的硬盘设备

指标名称	外存时间利用率
测试设计	1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 注：外存时间利用率的测试通常在测试时间特性及容量特性时同时进行，因此只需要在测试这些特性时同时考虑本指标的收集即可。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施	1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 注：如果在测试处理效率指标和容量指标的同时测试本指标，无需专门启动本测试，可与处理效率指标和容量指标的同时测试。 2. 收集测试数据：在每个单位时间，收集外存设备忙于读写的时间的百分比。 注：外存设备读写时间百分比可通过操作系统提供的相关性能计数器值取得。
测试结果计算	1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时外存时间利用率 外存时间利用率 $X(t_i)=t_i$ 时刻外存时间利用率，$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内的外存时间利用率 外存时间利用率(平均) $X_{min}=AVG(X(t_i))$，$t_i=VT_s\cdots VT_e$ 外存时间利用率(最大) $X_{max}=MAX(X(t_i))$，$t_i=VT_s\cdots VT_e$

7.3.2 外存空间利用率

指标名称		外存空间利用率
测试说明		1. 外存空间利用率可反映在指定的测试约束条件下，外存空间大小对系统运行效率所造成的影响。 2. 该指标应用于外存空间对系统效率的影响比较重要的场合，应充分考虑该指标的适用性。 3. 外存空间利用率还可以用于测试外存空间利用率的增长趋势，从而推算什么时间外存利用率将达到极限。适用时，可使用外存空间利用增长率来代替外存空间利用率。 4. 外存空间利用率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试，也可以单独进行，外存空间利用率易使用专门的测试场景考察外存空间利用率的增长情况。 5. 根据不同的系统特征，外存空间利用率指标的测试周期可能相对较长
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——需要测试的计算机的外部存储设备，如指定的硬盘设备
测试设计		1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 注：如果外存空间利用率的测试和时间特性或容量特性时同时进行，只需要在测试这些特性时同时考虑本指标的收集即可。 2. 如果需要考察系统运行过程中外存空间增长率，则需要设计采集时间周期和时机。 3. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注1：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。 注2：对采集周期比较长的测试，也可以采用人工方式采集方式。

指标名称	外存空间利用率
测试实施	1. 按照设计的测试环境、测试场景和系统负载模拟用户对被测系统的访问。 2. 收集测试数据：按照设计的采集时机，采集已使用外存空间以及总的外存空间大小
测试结果计算	1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算分时外存空间利用率 外存空间利用率 $X(t_i)=t_i$ 时刻已使用外存空间/总外存空间，$t_i=VT_s\cdots VT_e$ 3. 计算统计视图 外存空间利用率(平均) $X_{avg}=AVG(X(t_i))$，$t_i=VT_s\cdots VT_e$ 外存空间利用率(最大) $X_{max}=MAX(X(t_i))$，$t_i=VT_s\cdots VT_e$ 4. 必要时，计算外存空间利用率增长率 外存空间利用增长率=(外存空间利用率(VT_e)−外存空间利用率(VT_s))/(VT_e-VT_s)

7.4 传输利用性

7.4.1 传输能力利用率

指标名称		传输能力利用率
测试说明		1. 传输能力利用率可反映在指定的测试约束条件下，传输设备的传输性能对系统运行效率造成的影响以及传输设备的传输能力是否得到充分利用。 2. 传输能力利用率指标可在进行其他效率指标(如时间特性指标、容量特性指标)时同时进行测试，也可根据需要设计专门的传输功能相关测试场景。 3. 传输能力利用率指标关注网络传输性能，因此在设计测试环境时应关注对网络传输造成压力的功能
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——系统配置中负载压力较大的网络传输设备； ——其他需要测试的网络传输设备 **注**：必要时，测试不同网段之间的网络传输效率。
测试设计		1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 **注**：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 2. 收集测试数据：在每个单位时间，收集传输吞吐率。 **注 1**：传输吞吐率可通过操作系统提供的相关性能计数器值取得。 **注 2**：传输吞吐率还可以通过网络测试工具取得。

指标名称	传输能力利用率
测试结果计算	1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时传输能力利用率 传输能力利用率 $X(t_i)=t_i$ 时刻传输吞吐率/传输设备最大允许吞吐率。$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内的传输能力利用率 传输能力利用率(平均)=AVG(传输能力利用率(t_i))，$t_i=VT_s\cdots VT_e$ 传输能力利用率(最大)=MAX(传输能力利用率(t_i))，$t_i=VT_s\cdots VT_e$

7.4.2 传输错误率

指标名称		传输错误率
测试说明		1. 传输错误率可反映在指定的测试约束条件下，传输错误是否对系统运行效率造成影响。 2. 传输错误率指标可与传输能力利用率指标同时进行测试
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——系统配置中负载压力较大的网络传输设备； ——其他需要测试的网络传输设备 注：必要时，测试不同网段之间的网络传输效率。
测试设计		1. 按照测试约束规定，确定所需的测试环境、测试场景和系统负载。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，向服务器发出大量并发请求。 2. 收集测试数据：在每个单位时间，收集传输错误 注 1：传输错误率可通过操作系统提供的相关性能计数器值取得。 注 2：传输错误率还可以通过网络测试工具取得。
测试结果计算		1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时传输错误率 传输错误率 $X(t_i)=t_i$ 时刻传输错误率。$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内的传输错误率 传输错误率(平均)=AVG(传输错误率(t_i))，$t_i=VT_s\cdots VT_e$ 传输错误率(最大)=MAX(传输错误率(t_i))，$t_i=VT_s\cdots VT_e$

7.5 I/O 设备利用性

7.5.1 I/O 设备利用率

指标名称	I/O 设备利用率
测试说明	1. I/O 设备利用率可反映在指定的测试约束条件下，现有的 I/O 设备资源对系统运行效率造成的影响以及 I/O 设备是否得到充分利用。 2. I/O 设备利用率可在进行其他效率指标(如时间特性、容量特性)时同时进行测试，也可以针对 I/O 设备的使用场景设计专门的测试用例

指标名称		I/O 设备利用率
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——在效率需求中具有效率要求的 I/O 设备； ——系统配置中关键的 I/O 设备； ——其他需要测试的 I/O 设备
测试设计		1. 按照条件约束规定的测试环境、测试场景和系统负载准备测试用例，测试场景中应包括对待测试 I/O 设备的 I/O 请求。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，并产生对 I/O 设备的I/O 请求。 2. 收集测试数据：对指定 I/O 设备，收集每个单位时间内 I/O 设备的繁忙时间。 如无需收集分时数据，则收集 I/O 设备在有效测试周期内总的繁忙时间
测试结果计算		1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内每个单位时间 t_i 内分时 I/O 设备利用率 I/O 设备利用率 $X(t_i)=t_i$ 单位时间内设备忙时间/单位时间。$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内 I/O 设备利用率 I/O 设备利用率(平均)=AVG(I/O 设备利用率(t_i))，$t_i=VT_s\cdots VT_e$ 或 I/O 设备利用率(平均)=有效测试周期内 I/O 设备忙时间/有效测试周期时间

7.5.2 I/O 错误率

指标名称		I/O 错误率
测试说明		1. I/O 错误率可反映在指定的测试约束条件下，I/O 设备错误是否对系统运行效率造成影响。 2. I/O 错误率指标可与 I/O 设备利用率指标同时进行测试
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——在效率需求中具有效率要求的 I/O 设备； ——系统配置中关键的 I/O 设备； ——其他需要测试的 I/O 设备

指标名称	I/O 错误率
测试设计	1. 按照条件约束规定的测试环境、测试场景和系统负载准备测试用例，测试场景中应包括对待测试 I/O 设备的 I/O 请求。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施	1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，并产生对 I/O 设备的I/O 请求。 2. 收集测试数据：对指定 I/O 设备，收集每个单位时间 I/O 设备的出错次数。 如无需收集分时数据，则收集有效测试周期内出现的 I/O 设备的总出错次数。
测试结果计算	1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内每个单位时间 t_i 内的 I/O 出错率 I/O 出错率 $X(t_i)=t_i$ 单位时间内 I/O 出错次数。$t_i=VT_s\cdots VT_e$ 3. 计算有效测试周期内 I/O 出错率 I/O 出错率(平均)$X_{avg}=AVG(X(t_i))$，$t_i=VT_s\cdots VT_e$ 或 I/O 出错率(平均)X_{max}=有效测试周期内 I/O 出错数总数/有效测试周期时间

7.5.3 I/O 等待时间

指标名称		I/O 等待时间
测试说明		1. I/O 等待时间可反映在指定的测试约束条件下，I/O 的等待时间是否对效率目标造成影响。 2. I/O 等待时间指标可与 I/O 设备利用率指标同时进行测试
测试约束	环境约束	适用的环境约束： ——系统规定的目标运行环境； ——其他需要进行效率测试的系统运行环境
	负载约束	适用的负载约束： ——常规负载； ——峰值负载； ——极限负载
	对象约束	适用的对象约束： ——在效率需求中具有效率要求的 I/O 设备； ——系统配置中关键的 I/O 设备； ——其他需要测试的 I/O 设备
测试设计		1. 按照条件约束规定的测试环境、测试场景和系统负载准备测试用例，测试场景中应包括对待测试 I/O 设备的 I/O 请求。 2. 针对要测试的系统设备及相关资源，规划监控和数据收集设施 注：收集资源的设施包括：操作系统性能监视工具、专用测试工具、自行编写的数据收集程序。
测试实施		1. 按照设计的测试环境、测试场景和系统负载模拟大量用户访问被测系统，并产生对 I/O 设备的I/O 请求。 2. 收集测试数据：收集测试中针对指定 I/O 设备的每个 I/O 的发出时间及 I/O 处理完成并返回结果的等待时间
测试结果计算		1. 识别有效测试周期，设 VT_s 为有效测试周期开始时间，VT_e 为有效测试周期结束时间。 2. 计算有效测试周期内分时 I/O 等待时间，即在 t_i 时刻发出的 I/O 请求的等待时间 I/O 等待时间 $X(t_i)$=AVG(I/O 请求完成时间(j)−I/O 请求发出时间(j))。$t_i=VT_s\cdots VT_e$，j 为在 $t_i\cdots t_{i+1}$ 时间段内发出的 I/O 请求 3. 计算有效测试周期内的 I/O 设备等待时间 I/O 等待时间(平均)$X_{avg}=AVG(X(t_i))$，$t_i=VT_s\cdots VT_e$ I/O 等待时间(最大)$X_{max}=MAX(X(t_i))$，$t_i=VT_s\cdots VT_e$

附 录 A
（资料性附录）
效率指标体系应用框架裁剪指南

A.1 概述

本部分包含一组效率测试指标、各种测试约束和测试方法，旨在对计算机系统和软件的效率特性给出通用的度量指标体系及其测试方法。在实际应用中，不同的项目具有不同的特点，建议不同类型的使用者在使用本指标体系和应用框架时，应根据测试需求进行适当裁剪，以达到不同的效率测试目标。裁剪的类型包括：

——删除：如果本部分中规定包括的测试项或方法不适用于指定的测试，则可删除该测试项或方法；

——增加：如果测试中需要的测试项或方法未包括在本部分中，可根据需要增加新的测试或方法；

——修改：如果本部分中规定的测试项或方法与测试需求不一致，可在本部分基础上进行修改。

A.2 指标体系剪裁

对指标体系的剪裁：

——删除特性/子特性：当指标体系中的某个特性/子特性不适用于特定的测试，可删除相关特性/子特性。

例如，被测系统中未使用外部存储器，则可删除子特性“外存利用性”及相关度量指标；

——删除度量指标：当指标体系中的某个度量指标不适用于特定的项目，可删除相关度量指标。

例如，被测系统条件中不存在数据存储功能，则可删除“数据存储容量”度量指标；

——增加度量指标：如果效率需求中包括指标体系中未包括的度量指标，可增加相关测试指标。

例如，如果待测试系统为网络管理软件，则可能需要在容量特性中增加“网络容量”相关度量指标；

——修改度量指标：当度量指标定义与测试系统及效率需求存在差异，可修改本标准度量指标。

例如，本标准定义的度量指标“吞吐率”在 web 环境下可以用“点击率”“页面连接率”来代替。

A.3 测试约束剪裁

对测试约束的剪裁：

——删除环境约束：如果待测试系统的目标环境是确定的，并不需要考察不同环境下的指标的不同，则可删除环境约束，仅作为基本测试环境来考虑；

——增加测试约束：如果某测试条件的变化直接影响测试结果，则可增加该条件为测试约束。

例如，当需要测试不同类型的用户在执行某事务的周转时间和吞吐率，则可以增加“用户类型”为新的测试约束。

A.4 测试约束取值剪裁

对测试约束取值的剪裁：

——删除测试约束取值：每个测试约束可以包括不同的测试约束取值，测试时可选择适用的测试约束取值，删除不需要的测试约束取值。例如：当只需要测试常规和峰值负载条件下的效率指标，不需要测试极限负载条件下的效率指标，可选择负载约束取值为常规负载和峰值负载，删除极限负载；

——增加测试约束取值：当测试需求需要测试某种特定条件下的效率指标，而该条件未包括在本部分列出的适用的测试约束值中，则可增加该测试约束取值。例如：当需要测试指定负载条件下的效率指标，可增加该指定负载的负载约束。

A.5 结果视图剪裁

对结果视图的剪裁：

——选择统计视图：统计视图包括最小值、平均值、最大值等。如果测试需求不需要详细的分布信息，可以只选择平均值，删除最小值、最大值等统计视图；

——增加统计视图：如果测试需求需要更详细的分布信息，可增加统计视图。

例如：可增加 10%、90%或类似统计视图；

——删除满足度视图：如果待测试系统未能给出度量指标的期望结果，则可删除满足度视图；

——修改满足度视图：可选择不同的满足度计算方法。例如归一化满足度；

——删除分时视图：如果不需要对测试过程进行分析，则可删除分时视图；

——修改分时间隔：可修改分时视图的分时间隔。例如，当测试时间比较长时，可规定分时视图的时间间隔超过常规的 1 s，比如每 10 s、每 30 s。

A.6 测试方法剪裁

对测试方法的剪裁：

对测试方法和步骤的剪裁应适用于所处的测试条件和环境，本部分旨在给出一个基本的广泛适用的测试方法和过程，而未考虑不同类型的系统特征、所借助工具、先进技术和方法等因素，因此，本部分的使用者可根据需要增加、修改、删除本部分中所描述的测试活动和任务。

参 考 文 献

[1] GB/T 5271.1—2000 信息技术 词汇 第1部分:基本术语(eqv ISO/IEC 2382-1:1993)

[2] GB/T 8566—2007 信息技术 软件生存周期过程(ISO/IEC 12207:1995, IDT)

[3] GB/T 11457—2006 软件工程术语

[4] GB/T 16260.1—2006 软件工程 产品质量 第1部分:质量模型(ISO/IEC 9126-1:2001, IDT)

[5] GB/T 16260.2—2006 软件工程 产品质量 第2部分:外部度量(ISO/IEC TR 9126-2:2003,IDT)

[6] GB/T 16260.3—2006 软件工程 产品质量 第3部分:内部度量(ISO/IEC TR 9126-3:2003,IDT)

[7] GB/T 16260.4—2006 软件工程 产品质量 第4部分:使用质量度量(ISO/IEC TR 9126-4:2004,IDT)

中 华 人 民 共 和 国
国 家 标 准
系统与软件效率 第3部分：测量方法
GB/T 29835.3—2013
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100013)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室：(010)64275323 发行中心：(010)51780235
读者服务部：(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 2.25 字数 56 千字
2014年1月第一版 2014年1月第一次印刷
*
书号：155066·1-48013 定价 33.00 元

GB/T 29835.3-2013

打印日期：2014年2月19日 F009A