

Performance of Virtual Desktops in a VMware® Infrastructure 3 Environment

VMware ESX 3.5 Update 2

The benefits of virtualization for enterprise servers have been well documented. These include reduced hardware, power, and cooling costs; improved manageability; and higher availability. Increasingly, enterprises are turning to virtualization to bring these benefits to their desktop infrastructure. Virtual desktops deployed on VMware Infrastructure 3 provide the benefits of virtualization on a robust and proven virtualization platform. When preparing for virtual desktop deployments, typical performance questions include the number of desktops that can be supported on a given system and the interactive performance that can be expected by the users of those desktops.

In this paper we examine the performance of virtual desktops running a typical mix of interactive applications on VMware ESX 3.5 Update 2. These include office application tasks such as editing documents, spreadsheets, and presentations, as well as browsing the Internet and reading documents. Each virtual desktop ran in a separate virtual machine and ran a workload consisting of a set of identical applications. The order in which the individual applications were exercised was randomized in each virtual desktop. The number of virtual machines was scaled up from 16 to 160 virtual machines, equivalent to scaling from one to 10 virtual machines per processor core on our 16-core system. We present performance results showing the effect on performance as we scale up the number of virtual desktops on a single server. In addition, we present some of the performance characteristics of the workload. We also investigate the ability of VMware ESX 3.5 to provide fairness among all of the virtual desktops.

The results show a small increase in overall response time up to seven virtual machines per core. After seven virtual machines per core, the increase in overall response time becomes more rapid, and the response time of some individual operations becomes long enough to be perceived as slow by some users. At all load levels, the virtual machines were given fair treatment by VMware ESX.

These results are specific to the workload and configuration used in our tests. However, they can be used to gain a basic insight into the scalability of virtual desktop workloads on VMware Infrastructure 3.

This paper covers the following topics:

- [“Workload”](#) on page 1
- [“Testbed Configuration”](#) on page 3
- [“Performance Results”](#) on page 5
- [“Conclusions”](#) on page 8

Workload

In order to examine the performance of interactive workloads on VMware Infrastructure 3 with VMware ESX 3.5, we constructed a workload intended to simulate the behavior of a typical desktop user. This behavior includes editing documents, browsing Web pages, and other similar operations. The simulated user has a specific set of tasks to complete. The time taken to complete all tasks, excluding time the user spends thinking

between actions, is measured and represents the overall response time of the workload. The workload runs on Windows XP within a virtual machine on ESX 3.5. In a virtual desktop deployment, each virtual machine is assigned to a single user. As a result, our workload simulates the behavior of only a single desktop user in each virtual machine. A number of interactive users can be simulated by running the workload in multiple virtual machines simultaneously.

The remainder of this section gives additional details about the workload. We discuss the complete list of tasks performed by a simulated user in “[Desktop User Tasks](#)” on page 2. We describe the execution of the workload in “[Workload Harness](#)” on page 2. We discuss additional details about the performance metrics captured by the workload in “[Performance Metrics](#)” on page 3.

Desktop User Tasks

The user simulated by our desktop workload has a set amount of work to perform. A number of different applications are used to perform these tasks. The tasks are completed by each simulated user in a random order. At the beginning of a task, the associated application is started. Depending upon the work to be completed, the user then edits or views an existing document, creates a new document, or performs other work appropriate to the application. Once the task is completed, the application is left open to simulate the memory demands placed on a system by a user working on multiple simultaneous applications. The tasks included in the simulated workload are as follows:

- Edit a text document using Microsoft Word. This task includes starting the application, opening an existing document, typing additional text into the document, and then saving the document. The typing speed was set to 40 words per minute, to model a relatively fast typist.
- Edit a spreadsheet using Microsoft Excel. This task includes starting the application with a new spreadsheet, typing new data into the spreadsheet, performing some operations on the data, and then saving the spreadsheet. This is done twice with two different sets of data. One data set is used for numerical calculations; the other is sorted.
- Browse Web pages with Microsoft Internet Explorer. This task includes starting the application, opening a Web site, and browsing a number of Web pages on that site. This is done twice with two different Web sites, one primarily containing text and the other primarily images.
- Read a document using Adobe Reader. This task includes starting the application, opening a document, and reading a number of pages of the document.
- View and edit a presentation using Microsoft PowerPoint. This includes starting the application, opening an existing presentation, watching the presentation as a slide show, editing the contents of the presentation, and saving the edited version as a new file.
- Browse a local file using Mozilla Firefox. This task includes starting the application and opening an existing document.
- Compress a group of files into a compressed folder (ZIP file). This task includes starting the compression program on a predefined set of files.

While working on a task, the simulated user pauses between actions. These pauses simulate time taken for thinking between steps. In this workload, we set the think times to an average of 20 seconds.

All of the applications are closed after the simulated user has completed all tasks. If multiple iterations of the workload are to be run, the user repeats the entire set of tasks on each iteration.

Workload Harness

The execution of the workload is controlled by a workload harness that includes components that execute both on the virtual machines and on an external system.

Within the virtual machines, we used AutoIT v3 to create scripts that automate the portion of the workload that simulates the behavior of an interactive user. AutoIT v3 is a scripting language and interpreter that can be used to automate interactions with GUI and text applications in Microsoft Windows (<http://www.autoitscript.com>). The AutoIT scripts control all interactions of the simulated user with the

applications. The response times are also measured within the AutoIT scripts. However, in order to avoid problems of clock skew within the virtual machines, we used a custom-developed module to allow the virtual machine to directly read the hardware time-stamp counter.

The operation of the workload is controlled by a test controller node, which runs on a machine separate from the ESX 3.5 host running the virtual machines. The test controller has two functions. It provides a workload parameter file to the virtual machines, and it signals them to start the workload. It also hosts the Web server that the simulated users access during the execution of the workload and provides an FTP site that the virtual machines use to upload the results.

Performance Metrics

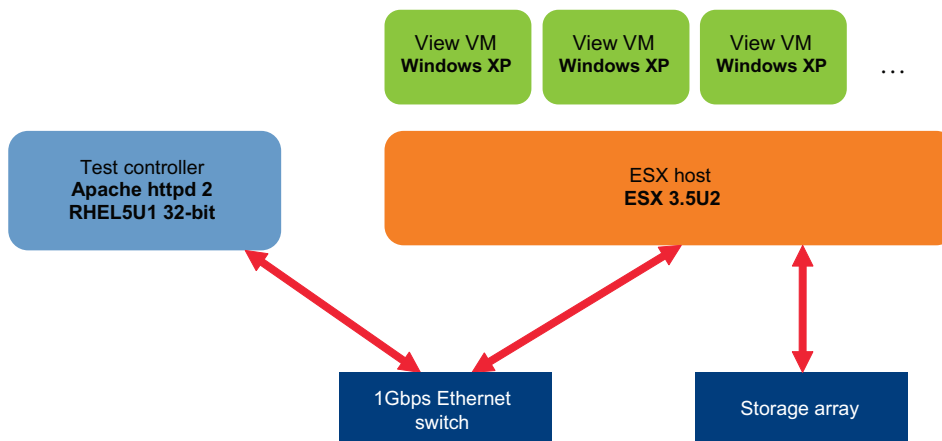
During the execution of the workload, the workload harness running within the virtual machine captures all response times as they would be observed by the desktop user. These include the time to open an application, load a Web page, and save a document. The sum of these times represents all of the time that the user spent waiting for a response while performing the set of tasks. This overall response time is the primary metric collected from the workload. In our test, we ran six iterations of the workload, using the first iteration as a warm-up period (see “Performance Results” on page 5). In this case, the reported response time is the average of five iterations. When multiple virtual machines run the workload simultaneously, we computed the average using the results from all virtual machines.

Testbed Configuration

This section describes the testbed used in these tests. As shown in Figure 1, the testbed consists of the test controller, the ESX host, which runs the virtual desktops, a storage array, and networking infrastructure. The test controller and the ESX host were connected to the same 1Gbps Ethernet switch. The storage array was connected to the ESX host by a single 4Gbps Fibre Channel link. The storage array hosted the virtual machines on a single 2TB LUN laid out over 22 disk spindles. We configured the LUN with the VMFS3 file system.

When creating the virtual desktops, we first created and configured a single virtual machine, designated the “golden” virtual machine. The virtual disk from this golden virtual machine was used as a base disk for all of the virtual desktops used in these tests. We created the remaining virtual machines used to host the virtual desktops as linked clones using VMware View Composer, a component of VMware View 3.

Figure 1. Testbed Configuration



The following sections give the details of the hardware and software configuration used in the testbed.

Hardware Configuration

Details of the server hardware configurations are given below.

Test Controller

- System model: Dell PowerEdge 2950

- Processor model: Intel Xeon X5355
- Processor speed: 2.66GHz
- Number of processors: two (four cores per processor)
- Total Memory: 32GB

ESX Host

- System model: HP ProLiant DL580 G5
- Processor model: Intel Xeon X7350
- Processor speed: 2.93GHz
- Number of processors: four (four cores per processor)
- Total memory: 128GB

Virtual Machine

- Number of virtual CPUs: one
- Total memory: 512MB
- Network adapters: one
- Hard Disks: two
 - Operating system drive (C:\): 10GB
 - Data drive (E:\): 500MB
- SCSI controller: LSI Logic

Software Configuration

The software configuration of the physical and virtual servers is given below.

Test Controller

- Operating system: Red Hat Enterprise Linux 5 Update 1 64-bit
- Web server: Apache httpd 2
- Tuning parameters: All left at defaults

Virtual Desktop Host

- Operating system: VMware ESX 3.5 Update 2
- Tuning parameters: All left at defaults

Virtual Desktop

The following configuration was used for the virtual desktop:

- Operating system: Windows XP SP2 32-bit
- Applications:
 - Microsoft Office 2003
 - Microsoft Internet Explorer 6
 - Mozilla Firefox 2
 - Adobe Reader 8.1.1

Performance Results

In this section we present the performance results from the interactive workload tests using multiple virtual machines. In these tests we used virtual machines running the workload discussed in [“Workload”](#) on page 1. Each virtual machine ran an identical workload, although the order in which the individual applications were exercised was randomized in each virtual machine. The number of virtual machines was scaled up from one to 10 virtual machines per core, for a maximum of 160 virtual machines on a single system. We used the run with one virtual machine per core as the basis for comparison. We kept all configuration parameters and system components identical to this base run. In each run, all simulated users went through six iterations of the workload, with a total time per run of approximately six hours. The initial iteration was treated as a warm-up period and was not included in the results.

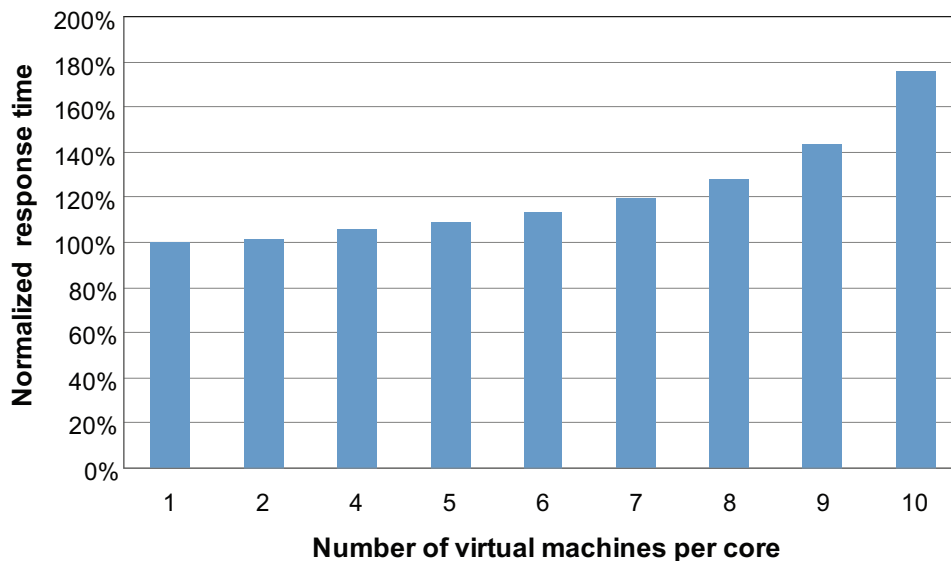
The results of this test are presented in the following sections. [“Overall Response Time”](#) on page 5 looks at the increase in overall response time as the number of virtual machines was increased. [“Operation Response Times”](#) on page 6 shows how this translates into the increase of actual response times experienced by a user on specific operations. [“Fairness”](#) on page 6 looks at whether individual virtual machines are given fair treatment when many virtual machines are running. [“CPU Utilization”](#) on page 7 shows the CPU utilization on our ESX host for different numbers of View virtual machines.

Overall Response Time

As discussed in [“Performance Metrics”](#) on page 3, we collected response times for operations performed by the simulated users on the applications running in the virtual machine. These response times are summed for each iteration of the workload to get an overall per-iteration response time. The overall response time for the workload is the average of the per-iteration response times. This represents the average time taken by a user to perform the given amount of work, excluding think time.

In this section we examine the increase in the overall response time as the number of virtual desktops was increased. [Figure 2](#) shows overall response time normalized to the response time when running one virtual machine per core, which for our system was equivalent to 16 virtual machines.

Figure 2. Normalized Overall Response Time per Virtual Desktop User



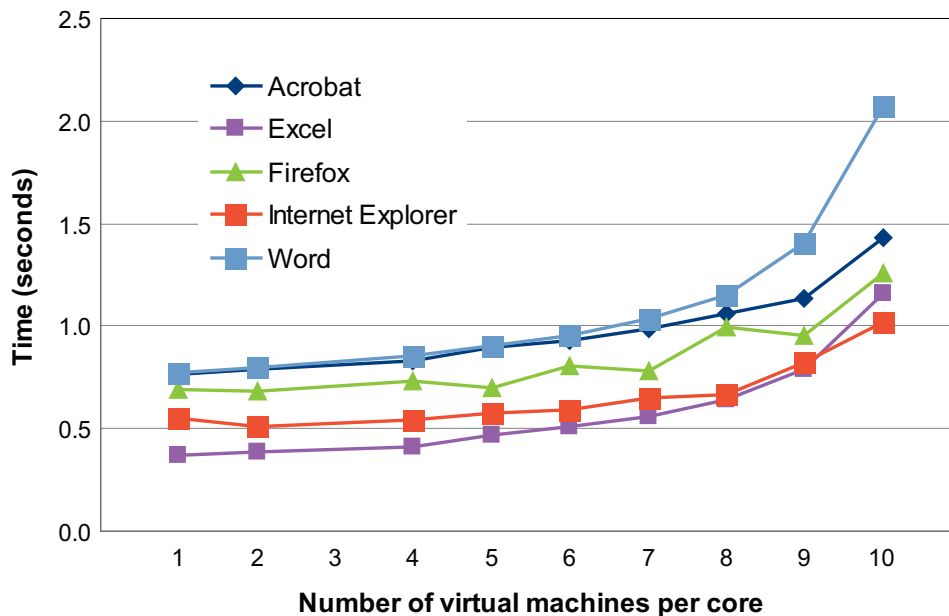
These results show that the increase in overall response time is relatively low out to seven virtual machines per core (112 virtual machines). After seven virtual machines per core, we start to see exponential increases in response time as the system begins to reach saturation. However, the absolute response times of most of the individual operations remains acceptable even at 10 virtual machines per core, as we discuss in [“Operation Response Times”](#) on page 6.

Operation Response Times

The increase in overall response time gives only a part of the overall performance picture. Even large increases in relatively fast operations may still give acceptable performance as observed by the virtual desktop user. In order to understand the usability of the system at different loads, it is necessary to look at the time observed by the user for the individual operations.

In [Figure 3](#), we show the time taken to start individual programs used in the workload and load the documents to be read or edited. These represent the longest times taken by any individual operations in the workload and would be experienced directly by the user as waiting time.

Figure 3. Application Start Times



In the 16 virtual machine case, all of the start times are below one second. At 112 virtual machines, all but Word still start in less than one second, and even at 128 virtual machines, only two of the application start times are just above one second. At 160 virtual machines, the start times have taken a larger jump, and the time to start Word is more than two seconds. It is clear that at this point the system is near saturation, and any additional increase in load will likely lead to a much larger increase in response times.

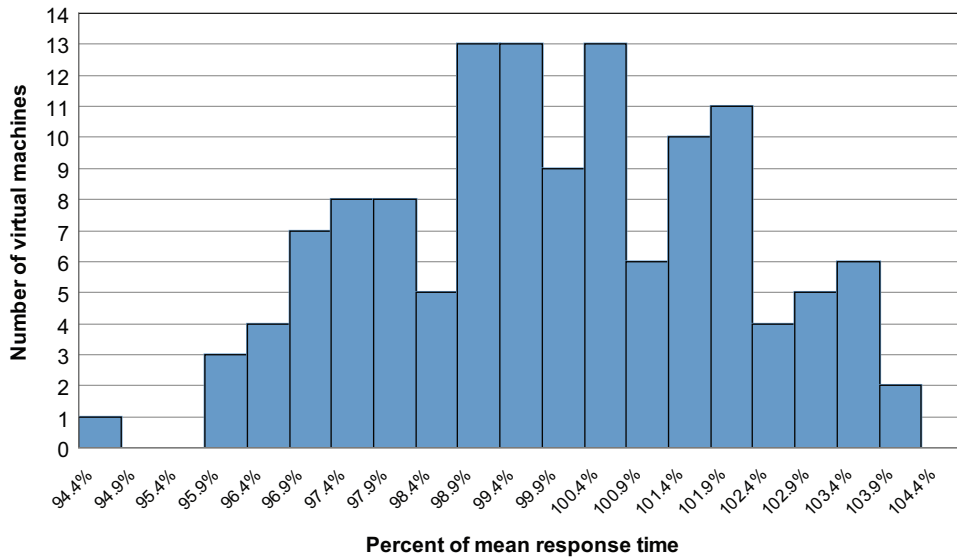
Fairness

When a large number of virtual machines is running on a single server, it is important that all virtual machines are treated fairly within the limits of their resource allocations. This is particularly true for interactive workloads, where response time is a key performance metric. In this section we look at data from a run of the workload with 128 virtual machines to examine how fairly the virtual machines are treated.

The metric that we use to look at fairness is the total response time of each individual virtual machine, normalized to the mean response time of all virtual machines. If the virtual machines are treated unfairly, the distribution of the normalized response times per virtual machine would have long tails, with some virtual machines having a response time much lower than the mean while others have response times much higher than the mean.

[Figure 4](#) shows a histogram of the normalized response times per virtual machine. Each bar represents the number of virtual machines whose response time is in a specific range relative to the mean response time. For example, one virtual machine had an average response time between 94.4 percent and 94.9 percent of the overall mean response time. From this graph we can see that the response times of 98 percent of the virtual machines are within 4 percent of the mean, and the response times of all of the virtual machines are within 5.6 percent of the mean. The standard deviation of the measured response times per virtual machine is 2.1 percent of the mean. The absence of outliers in the data and the low variation in response times indicate that the virtual machines are being treated fairly.

Figure 4. Histogram of Response Time Per Virtual Machine for a Run with 128 Virtual Machines

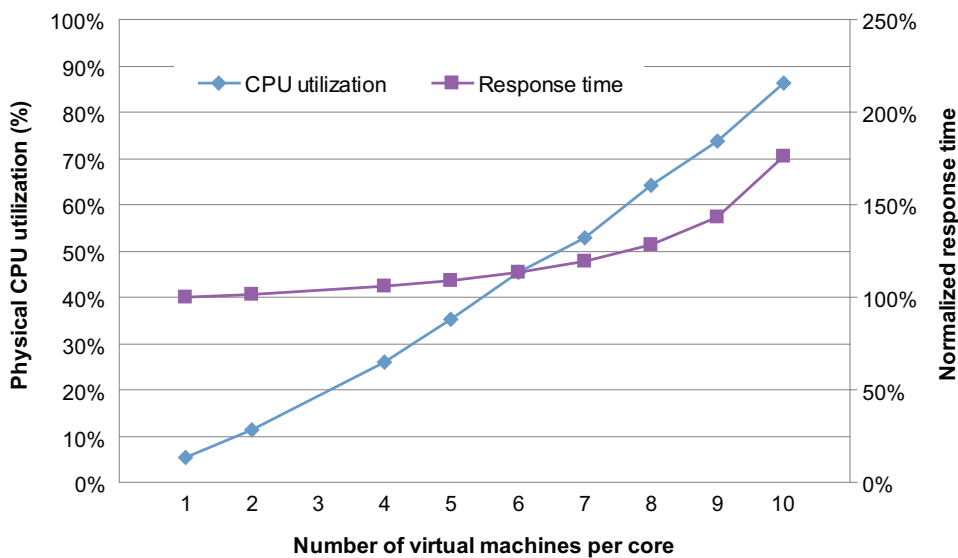


CPU Utilization

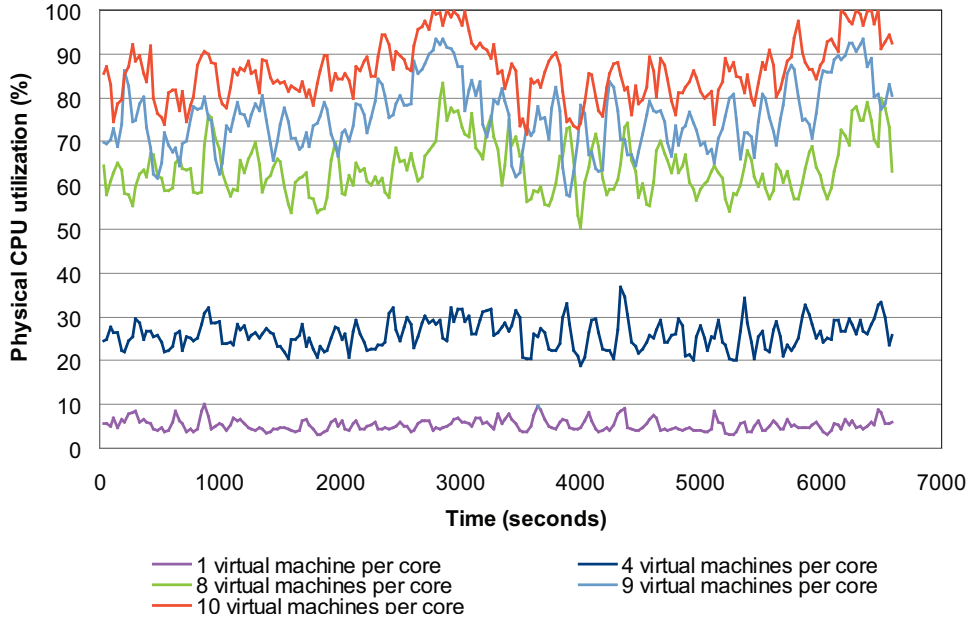
In this section we examine the trends in CPU usage as the number of virtual machines was increased. In these tests, the utilization of other system resources (memory, disk, and network) was relatively low even at 160 virtual machines.

Figure 5 shows the increase in average CPU utilization as the number of virtual machines was increased. The normalized response times are included for reference. Figure 5 shows the CPU utilization over the course of two iterations of the workload.

Figure 5. CPU Utilization with Increasing Numbers of Virtual Machines



In Figure 5 we see that the increase in average CPU utilization is close to linear as the number of active virtual machines increases. The normalized response time shows a similar linear trend out to seven virtual machines per core. Beyond 70 percent average CPU utilization, the response time begins to increase exponentially. The reason for this change can be seen in Figure 6. At large numbers of virtual machines per core, the View workload experiences spikes in CPU utilization. These spikes cause a disproportionate increase in the overall response times. At 10 virtual machines per core, the system experiences significant periods of CPU saturation, even though the average CPU utilization is still below 90 percent.

Figure 6. CPU Utilization Over Time

Conclusions

In this paper we have looked at the performance impact on interactive workloads of running a large number of virtual desktop users on a single VMware Infrastructure 3 host. For our workload and system configuration, we were able to run seven View virtual machines per processor core with less than a 20 percent increase in response time. At this load the absolute response times of the individual operations were still in a range that would be acceptable to virtual desktop users. These results are specific to the workload and configuration used in our tests. However, they can be used to gain a basic insight into the scalability of virtual desktop workloads on VMware Infrastructure 3.

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Copyright © 2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware, the VMware "boxes" logo and design, Virtual SMP, and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Revision: 20090113 Item: PS-083-PRD-01-01