# **Appendix C**

The purpose of this report is to document the limitations of the Transmission Control Protocol (TCP) when using the File Transfer Protocol (FTP) application in a controlled network environment.

Using various testing methods, we tried to isolate each component influencing the throughput for FTP transfers. We were also able to sort each test by sections. First, we'll characterize what the different Operating Systems' TCP performances are. Next, we will show how multiple access to files can affect performance of the FTP application. We will also look at how different TCP windows sizes can affect performance. The file size and the transfer time will be evaluated to see their influence on the throughput. We will end with the file access impact on the Central Processing Unit (CPU) utilization for both the server and the client.

# 1. Test Configuration

The tests were conducted at the Communication Research Centre (CRC). Figure 2.1 shows the local configuration used in the following tests. In the tests sections we will refer to the computer name. More information on the machine's configuration can be found in Appendix A.



Figure 2.1 Test network configuration

# 2. Platform Influence on Transfer Rate

#### **Test Overview**

The purpose of this test is to determine what influence the Operating System (OS) of the server/client can have on the network performance. The test consists in transferring a file from/to a server using wu\_ftp (FTP server for Linux) and a client using ncftp (FTP client for Linux) on Linux and Cute FTP (FTP client for Windows) and SERV-U FTP (FTP server for

Windows) on Windows. The same file is sent between computers using different operating systems. These tests were done with a 100MB File (to make sure the transfer rate has stabilized to its maximum) on the Gamma and the Lambda machines going through a 100Mbps link. The measurements were also taken with different transmit/receive buffer sizes within the ftp client application (It's not the same thing as TCP Window Sizes). These buffers refer to the space in RAM where the application puts the information from the disk before sending/saving it. The last result was taken from Kappa (dual CPU) to Gamma to test if symmetric multi-processors (SMP) usage affects the transfer speed.

#### Results

Linux-Linux 24 Mbps (Disk) 80 Mbps (RAM) Linux-Windows 8 Mbps (with 1k Buffer) Linux-Windows 5.6 Mbps (with 4k Buffer) Linux-Windows 5.8 Mbps (with 16k Buffer) Windows-Linux 8 Mbps (with 16k Buffer) Windows-Linux 7.2 Mbps (with 4k Buffer) Windows-Linux 3.2 Mbps (with 16k Buffer) Windows-Windows 16 Mbps (Disk) 60 Mbps (RAM) Linux-Linux (SMP) 5 Mbps

## Analysis

One can denote a big difference when looking at the results of Linux-Linux and Windows-Windows situation. Having a slightly more efficient file system (ext2fs) and somewhat different caching algorithms probably gives Linux the edge here. If we use an analogy by comparing the running application as a pile of dirt that needs to be carried away to fill a hole, caching corresponds to the size of the shovel used to carry the dirt. Indeed, caching acts as a transport medium from the memory to the Central Processing Unit (CPU) so bigger cache means more operations going to the CPU at the same time, thus increased performance. There are different types of caching, (RAM, Disk, and Pipeline), the difference is that they are links (buffers) between different devices. The Linux disk caching algorithms are caching methods used to write or read the disk.

When we look at Windows-Linux / Linux-Windows transfer rates we can see that they are even slower than the Windows-Windows one due to different TCP implementation. Operating systems like Rhapsody, AIX, and Windows behave like each other and presume that the slow start feature of TCP will initially send two segments (information packets). According to RFC 2001(Request For Comment for TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms) TCP slow start should send only one segment. Linux follows this RFC while Windows does not and the following situation unfolds:

non-RFC OS	sends ftp request.
Linux	sends the first segment and waits for an ACK.
non-RFC OS	waits 200ms for the seconds segment.
non-RFC OS	gives up and sends ACK.
Linux	sends more data

The 200ms delay is a long time for a machine that is less than 1ms away. You can test this by writing a little application in Linux that listens to a port with a streaming socket. Streaming sockets are also known as "connected sockets" mean that you have to "connect" a port before sending information. Many applications use this kind of socket including the "telnet" and "FTP" application. If you launch a telnet session from the Windows environment to that application's port, you will notice that when you write a string and press the CR, that string will appear on the server application. But, if you do the same thing in Windows, characters will appear to be read one at a time because of the TCP stack (buffer where received data is processed) delay under windows.

In Linux, the major performance problems are also with the TCP stack, because it is a single threaded in the 2.2.x Linux kernels, and has large-grained kernel locks that degrade

multiprocessor performance. That means that it creates a virtual file (lock file) to reserve the resource. While this might be an advantage over other Oses, thanks to its guaranty that multiple applications cannot access the device at the same time; it also degrades performance. The Linux community chose to keep the stability instead of trying to achieve higher performance. If you look at Appendix C (Open Benchmark's results) you will see that Windows NT server achieved a higher performance. This is because SMPs systems have a better performance under Windows NT while SMPs kernel support under Linux was experimental. This is supposed to be fixed with the kernel 2.4+ release because they had to change the whole network socket handling code to improve performance and generate faster threads.

# 3. File Access Influence on Transfer Rate

## **Test Overview**

The purpose of this test is to determine the influence of multiple file access on the bandwidth of a system. The test was done on Saturn downloading from Gamma going through a 10Mbps link, transferring a 100MB file, using optimal TCP window sizes, and with both computers running Linux.

## Graphs

3.1.1 Download Average Throughput VS Number of users on a 10Mbps link



Figure 4.2.1 Download Throughput VS Number of Users on a 10Mbps link

This graphic shows that average throughput decreases significantly with the number of users. This throughput is the throughput that each user has. It's not surprising when we see the shape of the curve in Figure 4.2.1, because the total bandwidth is shared between each user and that is why the average bandwidth gradually decreases.

#### 3.1.1. Total Throughput VS Number of users on a 10Mbps link



#### Figure 4.2.2 Download Throughput VS Number of Users on a 10Mbps link

For the total throughput we can see that the transfer takes its peak between 2 and 3 users. The same phenomenon can be observed on the results from the Mindcraft Open Benchmark tests (Appendix C)

#### Analysis

As we expect, the bandwidth is shared between each user (Figure 4.2.1). The results (Figure 4.2.2) also show that maximum throughput (transfer speed) is achieved when there are 2 hits to the file. After that, you see a lot of decrease in the average bandwidth as well as in the total bandwidth. Total bandwidth may be higher than single hit bandwidth because combined bandwidth takes up all available bandwidth alternating between each packet received. Also, when you have only one connection you must ensure that this connection is the fastest and, because FTP takes time to send an acknowledgement (ACK) packet, you can't achieve optimal transfer rates.

# 4. Settings Influence on Transfer Rate

The purpose of this test is to see if different settings on the machine can influence the performance for multiple hits and different file size results. This test consists of transferring files of different sizes between computers using the windows auto-sizing routines. This means that it detects the optimal window size for the transfer. By doing that, TCP transfers can get up to 10% more of speed (See test results by Mindcraft in Appendix C). The tests were done between Saturn and Gamma on a 10Mbps link.

#### Results



Figure 5.1.1 Throughput VS TCP Window Size

## Analysis

By looking at the results, we can tell that the best performance is achieved when we have a window size around 8KB-12KB but theses results will change depending on the latency, the maximum bandwidth availability and the network load. We can also notice that, usually, when transferring small files the window size is smaller than the size of the window used to transfer large files. We can also see that when there are multiple users the TCP window size seems to be bigger than when there's only one user transferring a file. On the local interface we can see that the maximum achieved throughput around 60.36Mbps for total and 35.76Mbps for average this is due to file size (next section) and total bandwidth distribution (like we saw earlier). The 60.36Mbps throughput corresponds to a 7.52 MB/s rate that is critical to the HD speed (7 MB/s sequential read/write). This limit is imposed by the disk speed and the total throughput will never be higher if we don't change the disks to have improved transfer rates.

# 5. File Size and Time Influence on transfer rate

#### **Test Overview**

These tests were also done between Saturn and Gamma on a 10Mbps link. They were also done locally on Saturn to verify the true limitations without going through the network.

# Graphs

# 5.1.1. Throughput vs Time for a 10Mbps link



Figure 6.2.1 Throughput Vs Time for a 10Mbps link

5.1.2 Throughput vs Time for local loopback



Figure 6.2.2 Throughput Vs Time for local loopback





Figure 6.2.3 Throughput Vs FTP File Size on a 10 Mbps link

## Analysis

If we look at the bandwidth for a certain time scale (Figure 6.2.1 and Figure 6.2.2) we can see that it starts slowly and goes faster to finally decrease a little in speed and achieve a stable transfer rate. File size has an indirect influence on the transfer rate because of the way FTP is designed. As we can see in this graphic (Figure 6.2.3), FTP has an optimal transfer window for files between 1MB and 7MB. For files 10KB and less, the transfer doesn't have the time to reach its peak. On the Bandwidth VS Time graphs, FTP starts at a very low speed than increases to reach a maximum. After that, it stabilizes to a specific throughput. File size, itself, doesn't affect the throughput. If the time it takes to transfer the file matches the peak on the curve than that file size will be the optimal file size to transfer because the transfer will not have the time to slow down before the file transfer ends.

# 6. File Access Influence on CPU Utilisation



Graph

Figure 7.1.1 CPU Utilization VS Number of users

#### Analysis

The CPU usage for the client application gets higher and higher to a point that it stabilizes to certain levels. It is because Linux knows how to do multitasking (unlike Windows) and the priority of the ftp client is low to take only the available CPU resources. We can see that once we've reached 4 sessions the CPU usage doesn't go any lower this is because whatever number of sessions are running the kernel keeps only 4-6 sessions running at a time while the others are sleeping. This causes a drop in the throughput but prevents CPU saturation. The server on the other end takes up all available CPU usage because it's a high priority process. CPU use for 1,2,3,4 sessions are the same because the CPU's portion taken by the ftp server daemon is shared between sessions when there are more than one. After 4 sessions the CPU usage starts to rise again because CPU is shared between each session and it keeps only 4-6 sessions running at a time. (Ex. 50% total usage with 4 sessions = 12.5% per process 4 running process = 0% (idling) and the same 50% total usage with 10 sessions = 5% per sessions with 4 running process = 30% idling (50%(total) - 20%(running))

# 7. Limitations

#### Local Loopback Limits

When a socket is created on this interface it exists with theses characteristics

- Theoretical
- Speed = Infinite
  MTU= Infinite
- Practical
- 1. Bus speed = 100 MHz (400 MB/s max.)
- 2. Finite memory access time (285 MB/s max.)
- 3. Finite hard disk access limited

- 4. Finite kernel interruption responses limited
- 5. Finite CPU usage limited

The speed of a loopback can differ from a computer to another. Indeed, I got different speeds (usually for only 1 file around 10 Mb is where you see the biggest difference) with my results when I downloaded at Gamma, which is faster CPU/HD and runs less background tasks, I got better results than what I got on Saturn (my workstation). The difference was around 10Mbps between the two for a 10Mb file and for a 100Mb file the difference was unnoticeable. That's because the peak on the time transfer is really dependent on the CPU/HD Speed. The following results are examples taken from Saturn. The other results can be found in Appendix A (Machine Information).

## **CPU** Limitations

Current CPU:	Dhrystone ALU 1487 MIPS Whetstone FPU 735 MFLOPS
Reference CPU 1:	Dhrystone ALU 1080 MIPS
Intel Pentium II 400MHz	Whetstone FPU 536 MFLOPS
Reference CPU 2:	Dhrystone ALU 1350 MIPS
Intel Pentium III 500MHz	Whetstone FPU 670 MFLOPS
Reference CPU 3:	Dhrystone ALU 1683 MIPS
AMD Duron 600MHz	Whetstone FPU 825 MFLOPS
Reference CPU 4:	Dhrystone ALU 3366 MIPS
AMD Athlon 1.2GHz	Whetstone FPU 1650 MFLOPS

#### Figure 8.2.1 CPU Benchmark

This is a benchmark for the CPU used in the tests. CPU saturation is what can be limiting the bandwidth if there are too much forked processes or threads.

The Arithmetic Logic Unit (ALU) is measured in MIPS (Million instructions per second) and is responsible for executing arithmetic and logical operations such as (ADD, XOR, AND, etc.) The Floating Point Unit (FPU) is measured is MFLOPS (Million float operation per second) and is responsible of all floating-point operations. So the higher these are, the faster the computer can process the information.

## **Memory Access Speed**

Int ALU/RAM Bandwidth 285 MB/s		
Float FPU/FAM Bandwidth 325 MB/s		
Int ALU/RAM Bandwidth 1374 MB/s		
Float FPU/RAM Bandwidth 1400 MB/s		
Int ALU/RAM Bandwidth 1030 MB/s		
Float FPU/RAM Bandwidth 1050 MB/s		
Int ALU/RAM Bandwidth 573 MB/s		
Float FPU/RAM Bandwidth 700 MB/s		
Int ALU/RAM Bandwilth 553 MB/s		
Float FPU/RAM Bandwidth 599 MB/s		

Figure 8.3.1 Memory Access Benchmark

Memory is not what's limiting the bandwidth but it always helps to managing multiple tasks if you have more memory. When you have multiple processes loaded into the memory, the memory usage can be maxed out. This wasn't tested because I had problems saturating memory with Linux. Linux does multitasking too well to let a process consume too much memory. The problem occurs more often under Windows because this OS cannot manage memory and leaves the responsibility to the applications. Since Windows has no control over the memory, if the application leaks or fails to release memory then you can get out of memory messages. Also, the faster the memory is the easier it is to transfer information. When doing transfers from ram to ram you can easily reach the networking media limit before reaching the memory limit. Memory is not considered a priority in TCP transfer because it does not affect the speed if it is not full. If it becomes full (out of "working" memory), the whole system will slow down because and so will the throughput.

## Network Performance for RAW sockets

S ACI
Senterprise
Network Throughput 928kB/s
Network Throughput 1200kB/s
Network Throughput 1000kB/s
Network Throughput 400kB/s
Network Throughput 400kB/s

Figure 8.4.1 Network Benchmark

You can see that the network performance results differ from the FTP results. This in fact is the real network performance if we only look at the Protocol/Transport layers without the application influence. These results give a good overview of what's the available bandwidth on that device. Results like these can also be obtained using Iperf on Linux. I ran Iperf on the machines used for tests and the results were around 9.1Mbps for my workstation to the gamma FTP Server. Also it was around 85Mbps from lambda to gamma. We also tested the link from CRC to NRC, which gave a result of 8.2Mbps. With these results we can testify that network performance is fine and it's the application/OS layer that are imposing the limitations.

Field	Value	
🔍 Sequential Read	8 MB/s	
🔍 Random Read	4 MB/s	
🍳 Buffered Write	7 MB/s	_
🍳 Sequential Write	7 MB/s	
🔍 Random Write	7 MB/s	
🍳 Average Access Time	7 ms (estimated)	

Hard Drive access performance

Figure 8.5.1 Hard Drive Access Benchmark

These are the hard drive access limitations. As we saw in the results, the Hard Disk has a big role in the file transfer limitations. It is what is limiting the transfer speed from going higher than 24Mbps on the test bed PCs.

# 8. Possible Solutions

- Testing File Transfers between RAMdisks would eliminate the HD limitations and could provide further testing information.
- Streaming Myth: Streaming will not improve the server's performance if its performance is limited by the hard disk speed. Indeed, the streaming process buffers data to the memory before sending it. It is dependent on the hard disk speed because it has to move this data to memory. It also needs to deal with multiple file access and that why it usually doesn't keep much of that data in memory before streaming the rest. Where it does make a difference is to the user's point of view. When you stream data, you don't have to write the data to the disk so it is only the server's HD and link speed that sets the maximum rate.

# 9. Conclusion

In conclusion, our bandwidth is limited at 24Mbps – 32Mbps range (5 MB/s), due to the hard drive read/write speed. In solution to this, it is recommended to update to another technology either 7200rpm EIDE drives (16MB/s) or Ultra-SCSI (40MB/s) drives. The first one can be plugged in directly while the second one needs a SCSI controller card if your computer doesn't have one already. Faster transfer speed will mean more expenses. If you use streaming, you only need to upgrade the server's harddisks and that might be the best solution so far.

# **10. Machine Information**

## Saturn

Name Saturn Processor(s) Processor(s) Intel Pentium III @ 550MHz ALU: 1487 MIPS FPU: 735 MFLOPS Installed Memory 256MB (72% true allocated load) Memory Access Speed: ALU: 285MB/s FPU: 325MB/s Network Raw Socket Throughput: 928kB/s = 7.42Mbps Hard Drive Access Speed: Sequential Read: 8MB/s = 64Mbps Random Read: 4MB/s = 32Mbps Sequential Write: 7MB/s = 56Mbps Random Write: 7MB/s = 56Mbps

## Gamma

Name Gamma Processor(s) Processor(s) Intel Pentium III @ 600MHz ALU: 1629 MIPS FPU: 806 MFLOPS Installed Memory 256MB (52% true allocated load) Memory Access Speed: ALU: 286MB/s FPU: 329MB/s Network Raw Socket Throughput: 10MB/s = 80Mbps Hard Drive Access Speed: Sequential Read: 8MB/s = 64Mbps Random Read: 4MB/s = 32Mbps Sequential Write: 7MB/s = 56Mbps Random Write: 7MB/s = 56Mbps

## Lambda

Name Lambda Processor(s) Processor(s) Intel Pentium III @ 750MHz ALU: 2050 MIPS FPU: 1018 MFLOPS Installed Memory 256MB (50% true allocated load) Memory Access Speed: ALU: 193MB/s FPU: 250MB/s Network Raw Socket Throughput: 78Mbps Hard Drive Access Speed: Sequential Read: 8MB/s = 64Mbps Random Read: 3.5MB/s = 32Mbps Sequential Write: 7MB/s = 56Mbps Random Write: 7MB/s = 56Mbps

## Карра

Name Kappa Processor(s) Processor(s) Intel Pentium III @ 866MHz ALU: 2250 MIPS FPU: 1234 MFLOPS Installed Memory 256MB (50% true allocated load) Memory Access Speed: ALU: 285MB/s FPU: 325MB/s Network Raw Socket Throughput: 82Mbps Hard Drive Access Speed: Sequential Read: 8MB/s = 64Mbps Random Read: 4MB/s = 32Mbps Sequential Write: 7MB/s = 56Mbps Random Write: 7MB/s = 56Mbps

# **11. Test Results**

## **Results For File Access Influence on Transfer Rates**

9.1.1. Inroughput for a juivipps link vs number of cile	9.1.1.	Throughput for a	10Mbps link	Vs number	of clients
---	--------	------------------	-------------	-----------	------------

1 User	2 Users	3 Users	4 Users
Upload:5.13Mbps	Upload:	Upload:	Upload:
	avg:3.11Mbps	avg:2.11Mbps	avg:2.07Mbps
	total:6.30Mbps	total:6.33Mbps	total:8.22Mbps
Download: 4.79 Mbps	Download:	Download:	Download:
	avg:2.36Mbps	avg:1.53Mbps	avg:1.22Mbps
	total:4.72Mbps	total:4.59Mbps	total:4.89Mbps

These results were obtained on a 10Mbps link with a Pentium III 550 MHz running Linux on both ends. The file transferred was a binary file of 500 MB.

9.1.2.	Throughput for a	a local loopback	Vs number of clients

<b>e</b> .			
1 User	2 Users	3 Users	4 Users
Upload: 14.96 Mbps	Upload:	Upload:	Upload:
	avg: 7.18 Mbps	avg: 7.12 Mbps	avg:7.148Mbps
	total: 14.36 Mbps	total:21.4Mbps	total: 28.6Mbps
Download:14.80Mbps	Download:	Download:	Download:
	avg:4.2Mbps total:	avg:4.38Mbps	avg:3.060Mbps
	8.4Mbps	total:13.14Mbps	total:12.24Mbps

These results were obtained on the local loopback interface in Linux. The file transferred was a binary file of 500 MB and the tests were done on Saturn. This interface is network card independent and is used to test the application capabilities. The limitations of this kind of interface are listed in the limitation section of this document.

# **Results for Settings Influence on Transfer Rate**

10Mbps	1 User	2 User	3 User	4 User
10 MB File	Upload:5.62Mbps	Upload:	Upload:	Upload:
		Avg:3.58Mbps	avg:2.37Mbps	avg: 2.51Mbps
		total:6.45Mbps	total:7.12Mbps	total: 9.2Mbps
	Download: 4.99 Mbps	Download:	Download:	Download:
		avg:2.99Mbps	avg:1.94Mbps	avg:2.03Mbps
		total: 5.98 Mbps	total:5.83Mbps	total:8.140Mbps
100 MB File	Upload:5.13Mbps	Upload:	Upload:	Upload:
		avg:3.2Mbps	avg:2.14Mbps	avg:2.08Mbps
		total:6.03Mbps	total:6.41Mbps	total: 6.26Mbps
	Download: 4.8 Mbps	Download:	Download:	Download:
		avg:2.47Mbps	avg:1.68Mbps	avg:1.47Mbps
		total:4.94Mbps	total:5.03Mbps	total: 5.88Mbps
1000 MB File	Upload:5.13Mbps	Upload:	Upload:	Upload:
		avg:3.11Mbps	avg:2.11Mbps	avg:2.07Mbps
		total:6.30Mbps	total:6.33Mbps	total: 6.22Mbps
	Download: 4.79 Mbps	Download:	Download:	Download:
		avg:2.36Mbps	avg:1.53Mbps	avg:1.22Mbps
		total:4.72Mbps	total:4.59Mbps	total: 4.89Mbps

9.1.3. Throughput results for different file sizes / hits on a 10 Mbps link.

9.1.4.	Throughput results fo	r different file sizes	/ hits on a loo	al loopback link.

Local	1 User	2 User	3 User	4 User
10 MB File	Upload: 35.76Mbps	Upload:	Upload:	Upload:
		avg:14.8Mbps	avg:11.84Mbps	avg:15.09Mbps
		total:29.6Mbps	total: 34.52Mbps	total: 60.36Mbps
	Download:	Download:	Download:	Download:
	25.28Mbps	avg:20.16Mbps	avg:10Mbps	avg:14.18Mbps
		total:40.32Mbp	total: 30Mbps	total: 56.72Mbps
		S		
100 MB File	Upload:	Upload:	Upload:	Upload:
	15.44Mbps	avg:13.84Mbps	avg:14.4Mbps	avg:5.44Mbps total:
		total:27.68Mbp	total: 43.2Mbps	21.74Mbps
		S		
	Download:	Download:	Download:	Download:
	14.88Mbps	avg:8.8Mbps	avg:5.45Mbps	avg:7.83Mbps
		total:17.6Mbps	total: 16.34Mbps	total:31.33 Mbps
1000 MB File	Upload: 14.96Mbps	Upload:	Upload:	Upload:
		avg:7.18Mbps	avg:12.8Mbps	Avg:7.154Mbps
		total:14.36Mbp	total: 38.4Mbps	total: 28.6Mbps
		S		
	Download:	Download:	Download:	Download:
	14.80Mbps	avg:14.2Mbps	avg:4.38Mbps	avg:3.060Mbps
		total:28.4Mbps	total: 13.14Mbps	total:12.242Mbps

Results for File Size a	and Time Influence	on transfer rate
-------------------------	--------------------	------------------

Time(s)	Throughput(Mbps)
1	1.7
2	4.18
3	4.13
4	4.07
5	3.97
6	3.89
7	3.86
8	3.83
9	3.73
10	3.69
20	3.69
30	3.68
40	3.67
50	3.67
60	3.65
70	3.64
80	3.63
90	3.58
100	3.57
200	3.57
300	3.57
400	3.56
500	3.57
600	3.58
700	3.56
800	3.57
900	3.56
1000	3.56

9.1.5. Table for Throughput vs Time on a 10 Mbps link

9.1.6. Table of results for Throughput VS Time for a local loopback

Time	BandWidth
1	3.2
2	9.76
3	10.72
4	9.36
5	8.88
6	9.2
7	8.8
8	9.04
9	8.88
10	8.56
20	8.56

30	8.56
40	8.64
50	8.64
60	8.56
70	8.56
80	8.56
90	8.64
100	8.56
200	8.64
300	8.56
400	8.56
500	8.56
600	8.64
700	8.64
800	8.56
900	8.56
1000	8.56

Table of results for the Influence of File Size on Throughput

10 Mbps link				
File Size (Kb)	BandWidth 1st	BandWidth 2nd Time	BandWidth 3rd	Average
	Time		Time	
10	2.53	3.41	3.58	3.173333
50	4.85	5.96	5.93	5.58
100	5.64	5.97	6.29	5.966667
500	6.37	6.21	6.421	6.333667
1000	6.57	6.39	6.42	6.46
5000	6.49	6.48	6.51	6.493333
10000	5.75	6.45	6.41	6.203333
50000	5.63	5.606	5.76	5.665333
100000	5.432	5.53	5.46	5.474

# Results for File Access Influence on CPU Utilization

9.1.7. Table of CPU Usage (Idling percentage) vs Number of users

Number of users	CPU utilisation Server(idle)	CPU Utilisation Client(idle)
	%	%
1	3.1	82.3
2	1.7	78.5
3	3.3	68.7
4	3.4	60.5
5	12.2	62.4
6	15.6	66.7
7	18.1	54.8
8	19.2	63.1
9	21	62.7
10	38.6	66.2