

## Data Movement Research Project Fall '05

Jain, Parul

*pj52@cornell.edu*

Shtokman, Dmitriy

*ds346@cornell.edu*

Tiwari, Harsh

*hot2@cornell.edu*

*Dept. of Computer Science, Cornell University*

*16<sup>th</sup> December 2005*

### Abstract

How do we transfer the data from the Internet Archive site to the local system at Cornell? What is the best way to make the files transfers most efficient? How can the network over which the files are being transferred be monitored to improve the performance of the download? What is the best approach to monitor the local system at Cornell? This paper discusses the various requirements and specifications that are essential to download data efficiently from the Internet Archive to Cornell. The first sets of data are downloaded from the Internet Archive using variety of File Transfer Protocol (FTP) clients, network monitoring tools and local system performance monitors are tested and evaluated to meet vital requirements.

### Introduction

The work done by the data movement group broadly is classified into three major parts:

1. Data Movement: the task of transferring the crawl files from the Internet Archive to the local system at Cornell. The 'raw' data is downloaded using file transfer protocol (FTP). FTP allows users to copy files between their local system and any system they can reach on the network [10].
2. Network Monitoring: the task of monitoring the performance of the Internet2 network over which the data is to be transferred.
3. Local System Monitoring: the task of monitoring the performance of the Scidata machine at the Cornell Theory Center.

Our assigned task was to develop strategies of how best to fulfill the requirements of each part. All the various statistics were collected and data analyzed to determine the most

efficient methods to accomplish the data transfers. The objective was to begin the first data movement from the Internet Archive and run tests on them to evaluate the effectiveness of the various strategies developed by the team. The data movement from the Internet Archive site to the local system at Cornell was to be done using the Internet2 connection. The targeted rate of download was around 250 GB/day.

With respect to the entire project, our work is most crucial in terms of making enough downloaded data available for the other teams to be able to run and test their systems with real files. As a result, the rate at which we transfer data drastically impacts the performance of the entire project, since inadequate amounts of raw data would mean that other parts of the system could not function optimally.

### Scope

The scope of the project was restricted by the following:

1. There were delays from Internet Archive in providing authorization for downloading files.
2. Nodes at the Internet Archive had some problems and their servers required restarting. Pinpointing and resolving these problems caused delays in data transfer.
3. The period from August to November was the preliminary research phase as the Internet2 connection was not set-up. The Internet2 connection was only made available by around Thanksgiving.
4. Once the Internet2 connection was set-up, it had some configuration issues with potential problems relating to the hardware at the Cornell Theory Center. Thus, this prevented the

network from working at its optimal bandwidth.

5. With regard to network monitoring, the selection of the ideal monitoring tool was restricted due to the compatibility of tools with the IA-64 architecture of the Scidata machine. The WinPcap packet capture protocol used by all the commercially available monitoring tools is not configured to work with the IA-64 architecture.

The above unavoidable circumstances prevented the file downloads from being efficient. In the absence of the Internet2 connection, the transfer speeds achieved were slower and the overall analysis was restricted. However, the preliminary design and specification strategies developed can directly be applied to evaluate the Internet2 connection.

### **Background Information**

The Internet Archive maintains the Internet crawls that are to be downloaded. Parts of each crawl are placed on different nodes and each directory on the nodes contains a pair archive (ARC) and data (DAT) files. An ARC file is an archive file of its corresponding data (DAT) file. Typically an ARC file is about 100 MB and a DAT file is 15 MB. The crawls are named using a unique nomenclature scheme. This semester, parts of the EA (January 2005) an EB (February 2005) crawls from node 300906 were downloaded totaling 1,488 files of ARC and corresponding DAT files. To download files, first a connection to a node in the US cluster of the Internet Archive has to be established.

The following sections discuss the three parts of the project: Data Movement, Network Monitoring and Local System Monitoring.

### **Data Movement**

The first task for the data movement task, before bringing in the data, was to determine how best to achieve this. The Internet Archive required that the file downloads be done using the FTP protocol and the network connection should use transport layer security (TLS). TLS is a protocol that guarantees privacy and data integrity between client/server applications

communicating over the Internet [10]. Keeping these criteria in mind we prepared a detailed study on the TLS connection, and various other specifications were developed to evaluate the best strategy for downloading the files.

### **TLS Research**

The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications. The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. TLS Record Protocol provides connection security that has two basic properties [8]:

- The connection is private
- The connection is reliable

The TLS Handshake Protocol provides connection security that has three basic properties:

- The peer's identity can be authenticated using asymmetric or public key, cryptography
- The negotiation of a shared secret is secure
- The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties to the communication

### **Goals of TLS Protocol**

- Cryptographic security- TLS should be used to establish a secure connection between two parties.
- Interoperability- Independent programmers should be able to develop applications utilizing TLS that will then be able to successfully exchange cryptographic parameters without knowledge of one another's code.
- Extensibility- TLS seeks to provide a framework into which new public key and bulk encryption methods can be incorporated as necessary.
- Relative-efficiency- Cryptographic operations tend to be highly CPU intensive, particularly public key operations.

## FTP Tool Testing

Keeping these requirements in mind we tested a number of FTP tools. Some of the tools include:

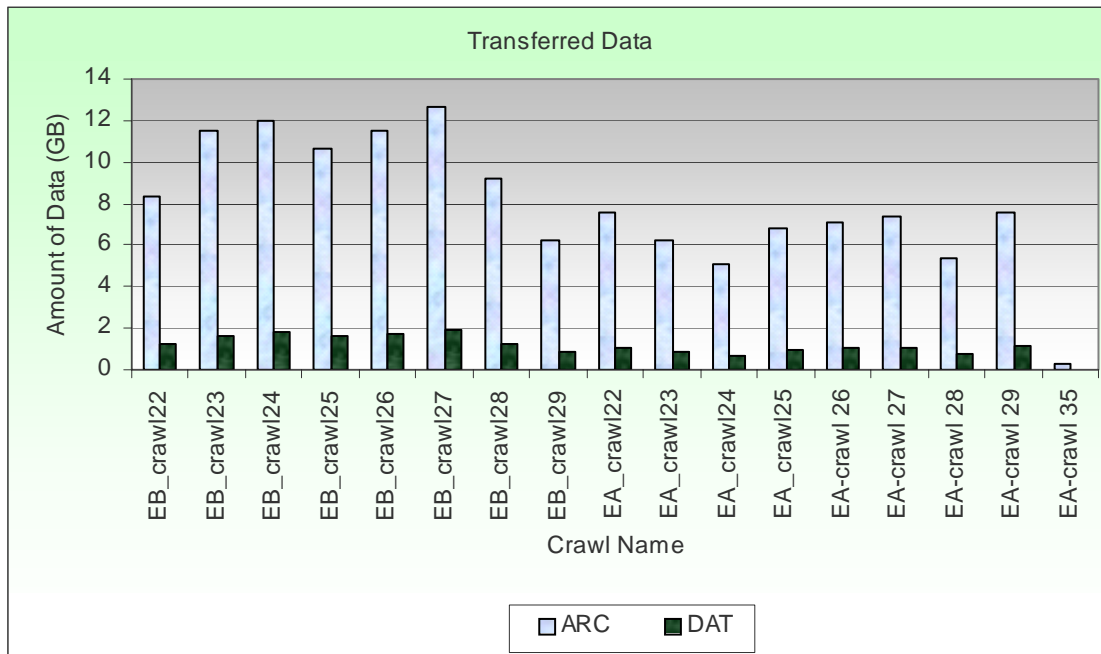
- 1- Core FTP
- 2- Igloo
- 3- Turbo FTP
- 4- Smart FTP
- 5- FTP Voyager
- 6- FileZilla

Some of the important requirements that we tried to satisfy while testing are [3, 9]:

Among the various tools, SMART FTP best fit the requirements list and hence was chosen for further studies.

The total data transferred during the semester was 150 GB, with 135 GB of ARC and 20 GB of corresponding DAT files.

As described previously, various unavoidable events prevented file downloads from being more efficient. After some level of initial troubleshooting, we resolved the issues, downloading will be more efficient in the future. Figure 1 summarizes the download



**Figure 1- Data download Results Summary**

- 1- The tools should allow scheduling. The user should be able to create a list or queue of files for download and the software should process the list automatically.
- 2- In case of error in the transfer of broken links, the client should automatically reconnect and resume download.
- 3- The final data packets downloaded should be complete and error free.
- 4- The client should provide a log of various statistics to measure the efficiency of the download.

results. Node 300906 EA and EB crawls corresponding to January and February 2005 respectively were completely downloaded with a total of 1,448 ARC and corresponding DAT files. The average speed of the downloads was estimated to be around 300 kb/s, ranging from a very slow 80 kb/s to the fastest at 600 kb/s. Based on estimations, downloading 1 terabyte of data at an average of 300 kb/s would take about one month. The project's ideal target is to be able to download the data at 1 terabyte / day. Once the Internet2 connection is configured

to work properly, we estimate the rate would increase up to 250 GB/day. At the moment it takes 7 days to download 250 GB. Thus, using the Internet2 would in effect increase the rate by a factor of 7.

Some of the experiments performed on the Internet2 connection include running separate instances of the FTP clients to take advantage of parallelism in file transfers. Also, hardware configuration problems at the Cornell Theory center are being looked at for getting the Internet2 connection working at optimal bandwidth.

### **Network Monitoring**

Similar to the Data movement part, our first task was to develop a set of requirements for analyzing the network.

The main aim was to monitor the distribution of the network's resources among the users and to study the traffic on the network other than the IA files downloads. Keeping these requirements in mind we tested a number of Networking Monitoring tools.

The following network managers have been used for downloading data and testing:

- 1- NTOP
- 2- Multi Router Traffic Grapher (MRTG)
- 3- Net SMTP- XTRA
- 4- MRTG - XTRA

Various specifications and requirements were short-listed for the evaluation of various tools; of these options available the following are applicable to this project [2, 7]:

- 1- Sort network traffic according to many protocols

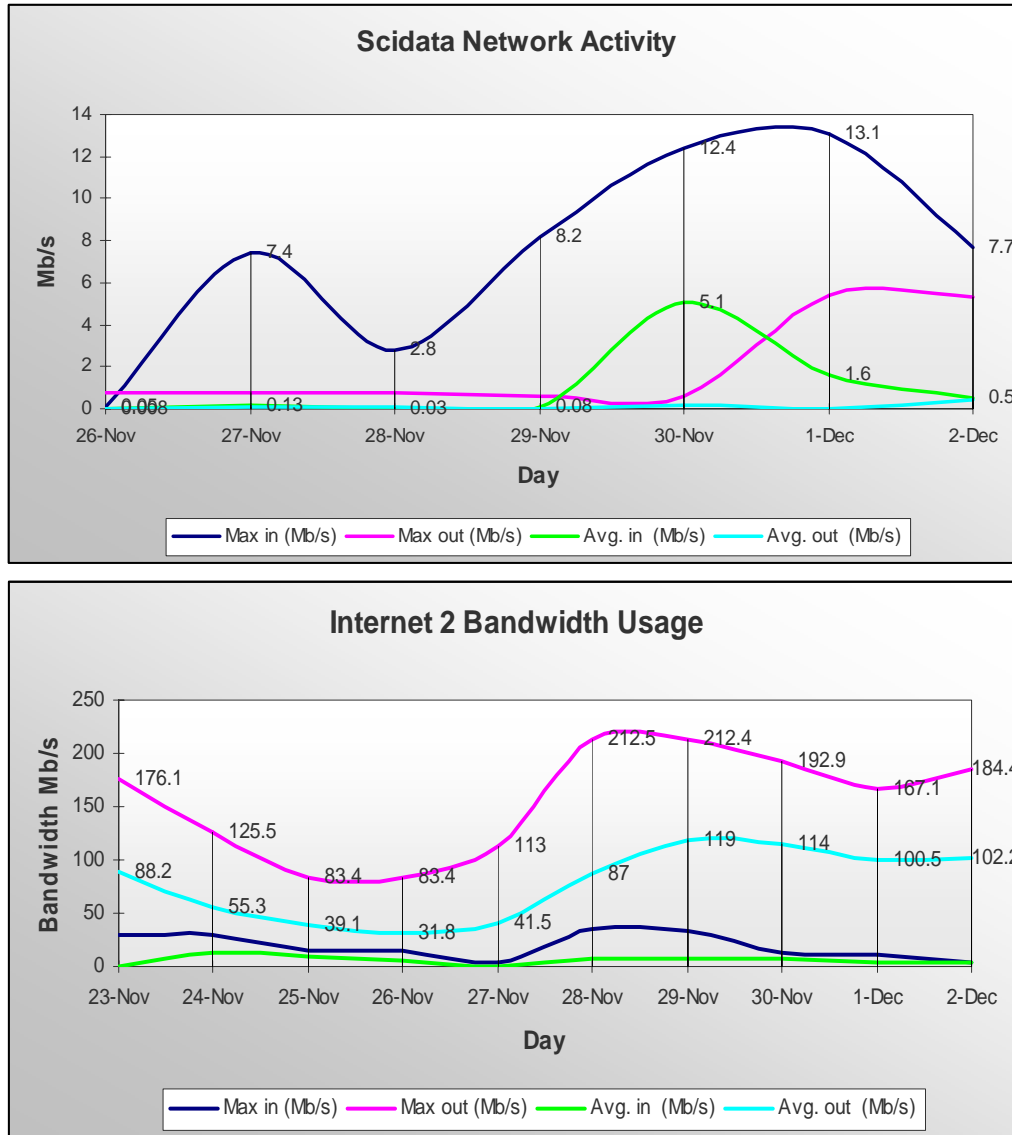
- 2- Show network traffic sorted according to various criteria
- 3- Display traffic statistics
- 4- Identify the identity (e.g. email address) of computer users
- 5- Passively (i.e. without sending probe packets) identify the host OS
- 6- Show IP traffic distribution among the various protocols
- 7- Analyze IP traffic and sort it according to the source/destination
- 8- Display IP Traffic Subnet matrix (who's talking to who?)
- 9- Report IP protocol usage sorted by protocol type
- 10- Collect information flows generated by routers or switches

The N-top software is optimal to provide essential network statistics required to monitor the transfer of data from IA to Cornell. Other software such as MRTG is available for analysis, however, the web interface and the ease of configuration/setup makes N-top a better choice. MRTG produces very similar results to N-top and might be useful for a script level analysis.

For all the previously mentioned tools, the industry standard windows packet capture library WinPcap, which allows the operating system to access low-level network information, was found to be incompatible with any 64-bit operating system. Thus, none of these above tools can be used for analyzing the network on Scidata1. Most of the commercial networks monitoring software are not designed to run in a 64-bit environment and hence no compatible version of WinPcap exists.

MRTG is already running on the systems at Cornell and results are being generated and stored as HTML pages. Since MRTG is already running on the system and no other network monitoring tools are available for the IA-64, it is the option we will focus on for the

results that was collected while evaluating various analytical strategies. These statistics are generated from monitoring the network over a period when no files were being downloaded from the Internet Archive. Thus, the data here represents the traffic by other



**Figure 2- Network activity for Scidata (top) and Internet2 (bottom)**

time being. Using the available MRTG graphs for the Internet2 connection and the Scidata machine network, an estimate of the total traffic on the networks was made. Figure 2 provides a snapshot of the network activity over a week's period. This is a set of sample

users on the network. These graphs were generated using the data collected from monitoring various statistics through running experiments. As shown, for the Internet2 connection, the outgoing traffic on the network was at an average of 77.86 Mb/s,

equivalent to 8% of the available bandwidth. This implies that around 8% of the bandwidth is being utilized by external users for transferring data other than the Internet Archive files. As this number starts increasing, it limits the maximum speed that we can achieve from the Internet2 connection for the file downloads. On the other hand, the incoming traffic on the Scidata machine was a negligible 1% of the available bandwidth (1.7 Mb/s). Thus, the Scidata machine network traffic is minor and does not need much optimization with respect to improving the efficiency of the Crawl downloads. Table 1 summarizes the findings from the testing. Soon Cornell is expecting to set-up a faster network: the TERAGRID network, which is expected to provide direct link to the Internet Archive. Once this is in place, then the bandwidth problem highlighted above will not be of any concern.

- 2- Interrupt Time Counter: % Interrupt Time is the time the processor spends receiving and servicing hardware interrupts during sample intervals.
- 3- User Time Counter: % User Time is the percentage of elapsed time the processor spends in the user mode.

**Logical Disk Object**

The Logical Disk performance object consists of counters that monitor logical partitions of hard or fixed disk drives. Performance Monitor identifies logical disks by their drive letter, such as C.

- 1- Disk Time Counter: % Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or writes requests.
- 2- Disk Write Time Counter: % Disk Write Time is the percentage of elapsed time that the selected disk

| Machine   | Average Incoming (Mb/s) | Average Incoming Bandwidth | Average Outgoing (Mb/s) | Average Outgoing Bandwidth |
|-----------|-------------------------|----------------------------|-------------------------|----------------------------|
| Scidata   | 1.7                     | 1%                         | 0.17                    | 0.1%                       |
| Internet2 | 4.86                    | 0.5%                       | 77.86                   | 8%                         |

**Table 1- Network Analysis Result Summary**

**Local System Monitoring**

We studied perfmon in detail and documented all relevant counters that will be used for the Weblab project. Some of the more important counters include [4]:

**Processor Object**

The Processor performance object consists of counters that measure aspects of processor activity the processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes.

- 1- Idle Time Counter: % Idle Time is the percentage of time the processor is idle during the sample interval

drive was busy servicing write requests.

- 3- Free Space Counter: % Free Space is the percentage of total usable space on the selected logical disk drive that was free.
- 4- Idle Time Counter: % Idle Time reports the percentage of time during the sample interval that the disk was idle.
- 5- Avg. Disk Bytes/Transfer Counter: Avg. Disk Bytes/Transfer are the average number of bytes transferred to or from the disk during write or read operations.
- 6- Avg. Disk Bytes/Write Counter: Avg. Disk Bytes/Write is the average

number of bytes transferred to the disk during writes operations.

- 7- Avg. Disk Queue Length Counter: is the average number of both read and writes requests that were queued for the selected disk during the sample interval.
- 8- Avg. Disk sec/Transfer Counter: is the time, in seconds, of the average disk transfer.
- 9- Disk Write Bytes/sec Counter: is rate at which bytes are transferred to the disk during writing operations.

### **Memory Object**

The Memory performance object consists of counters that describe the behavior of physical and virtual memory on the computer.

- 1- Available Bytes Counter: is the amount of physical memory, in bytes, immediately available for allocation to a process or for system use.
- 2- Available Kbytes Counter and Available Mbytes Counter
- 3- Pages Output/sec Counter: is the rate at which pages are written to disk to free up space in physical memory. Similar to this we have Pages Writes/sec Counter
- 4- Pages/sec Counter: is the rate at which pages are read from or written to disk to resolve hard page faults.
- 5- Packets/sec Counter: is the rate at which packets are sent and received on the network interface.

### **Network Interface Object**

The Network Interface performance object consists of counters that measure the rates at which bytes and packets are sent and received over a TCP/IP network connection. It includes counters that monitor connection errors.

- 1- Bytes Total/sec Counter: is the rate at which bytes are sent and received over each network adapter, including framing characters.
- 2- Current Bandwidth Counter: is an estimate of the current bandwidth of the network interface in bits per second (BPS).
- 3- Packets Received Errors counter: is the number of inbound packets that

contained errors preventing them from being deliverable to a higher-layer protocol.

- 4- Packets Received/sec Counter: is the rate at which packets are received on the network interface.

### **Objects Performance Object**

Processes Counter: is the number of processes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval.

### **Physical Disk Object**

The Physical Disk performance object consists of counters that monitor hard or fixed disk drive on a computer.

- 1- % Disk Time Counter: is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests. Similar to this we have Disk Read Time Counter and Disk Write Time Counter.
- 2- Avg. Disk Bytes/Transfer Counter: is the average number of bytes transferred to or from the disk during write or read operations. Similar to this we have Avg. Disk Bytes/Read Counter and Avg. Disk Bytes/Write Counter.

### **Process Object**

The Process performance object consists of counters that monitor running application program and system processes.

- 1- Processes Object: is the number of processes in the computer at the time of data collection.
- 2- System up Time Object: is the elapsed time (in seconds) that the computer has been running since it was last started.

### **Results**

The first set of data was transferred from the Internet Archive site to Cornell. Parts of the EA and EB crawls corresponding to the January and February 2005 Internet crawls were downloaded during the preliminary phase. In addition, the download process was tested and evaluated using a variety of FTP download clients, networking monitoring tools and local system performance monitors.

Based on the results, strategies for improving the efficiency of the file transfers were analyzed and the most optimal of the group was selected. A total of 150 GB of ARC and DAT files were downloaded. Most importantly, a strong foundation for future work was laid by troubleshooting various problematic initialization issues with respect to establishing connection between the Internet Archive Site and the Cornell Scidata machine.

### Future work

Further work in this area involves configuring the Internet2 connection for working optimally with FTP downloads. Possible ways to do this involve introducing parallelism, i.e. running multiple instances of the FTP clients and downloading different files simultaneously as well as tuning and optimizing the FTP client. Also, additional hardware configuration needs to be done at the Cornell Theory center for faster transfer speeds. These leverage the bandwidth of Internet2 network. In addition, the speed of the transfers needs to be increased so that one terabyte of data can be downloaded in about three to four days. Once the Internet2 connection is in place for downloading data at around 250 GB per day, the next step is to automate the file downloads from the Internet Archive so that a constant stream of data is maintained.

### References

[1] M. Gandhi, K. Jeyabalan, J. Kallukalam, A. Rabkin, P. Reilly, N. Widodo, "Web research infrastructure project semester research report", Fall 2004.

[2] MRTG web-site - <http://mrtg.hdl.com/mrtg.html>

[3] SmartFTP website - [www.smartftp.com/](http://www.smartftp.com/)

[4] Perfmon reference site - <http://perfmon.sourceforge.net/>

[5] Internet Archive site - <http://www.archive.org/web/researcher/>

[6] Web Laboratory Project - <https://gforge.cis.cornell.edu/projects/wri/>

[7] N-TOP network monitor site - <http://www.ntop.org/ntop.html>.

[8] TLS website - [www.javvin.com/protocolTLS.html](http://www.javvin.com/protocolTLS.html)

[9] Core FTP Lite site - [www.coreftp.com/](http://www.coreftp.com/)

[10] Webopedia Definitions - <http://isp.webopedia.com/TERM/>

### Acknowledgements

This work is part of the Web Laboratory, which is a joint project of Cornell University and the Internet Archive. Other members of the team are: William Arms, Blazej Kot, Pavel Dmitriev, Selcuk Aya, Chris Sosa, Jimmy Sun, Min-Daou Gu, Jimmy Yanbo Sun, Serena Kohli, Swati Singhal, Megha Siddavanahalli, Nick Gerner, Samuel Benzaquen, Shantanu Shah, Nicholas S Gerner, Nicolas Hamatake, Wei Guo and Lipi Sanghi.

We would also like to acknowledge the guidance and assistance provided by Ruth Mitchell and Lucia Walle of the Cornell Theory Center. Without their prompt actions and troubleshooting much of the work would have been incomplete.

Furthermore, this work would not be possible without the forethought and longstanding commitment of the Internet Archive to capture and preserve the content of the Web for future generations. The laboratory is one of several projects that depend on the Petabyte Storage Services for Data-Driven Science, which is funded in part by National Science Foundation grant 0403340, 0127308, and 0537606, with equipment support from Unisys. Additional support comes from Microsoft and Dell, and from Cornell University.

## Updates

### Work during the last week using the Internet2 Network: Dec 3-11

EC-crawl 22, 23, and 24 files were transferred from the ia301018 node.

There were 13 ARC files (1.22 GB) and 13 DAT files (225 MB) corresponding to EC-crawl\_22. There were 17 ARC files (1.58 GB) and 17 DAT files (291 MB) corresponding to EC-crawl\_23. There were 15 ARC files (1.40 GB) and 15 DAT files (267 MB) corresponding to EC-crawl\_23.

### Experimentation

On December 3, setting up a parallel upload using a second instance of Smart FTP caused the download speed to go down immediately. Originally the speed was about 240 Kbps, and with two parallel uploads the speed became about 130 Kbps for both transfers. Perhaps this was a general decrease in transfer speed as such decreases have been noticed earlier and possible are due to some hardware configuration issues at the theory center.

On December 6, increasing the number of threads in Smart FTP resulted in simultaneous file downloads (from the queue) without loss of the speed. Nine threads were supported in parallel resulting in faster download speeds:

- On December 6 the transfer speed went up to between 1 Mbps and 2 Mbps, and the average speed was around 1.5 Mb/s. With a transfer rate of 1.5 Mb/s, 116 GB can be downloaded in 22 hours. With this rate more than 3 terabytes of data can be downloaded per month.
- On December 7 the transfer speed was between 2 and 5 Mb/s, and the average speed was around 3 Mb/s. With this transfer rate more than 6 terabytes of data can be downloaded per month.

When the number of threads is increased beyond 9, the number of simultaneous transfers does not increase. Also, when Smart FTP is operating with 9 threads, running a second copy of the software causes the software to crash. Thus, only up to a maximum of 9 threads can be executed

in parallel using Smart FTP. In addition, running an additional instance of Core FTP in parallel to Smart FTP decreased the transfer rate of Smart FTP keeping the aggregate speed constant.

### Areas of concern

There were still problems connecting to the Internet Archive solo nodes. For example, the following nodes were still not accessible for downloads: ia300511, ia302013, and ia300528. An email on this subject has been sent to the Internet Archive contact: Tracey Jaquith.



ia300928.us.archive.org  
ia300929.us.archive.org  
ia300930.us.archive.org  
ia300931.us.archive.org  
ia300932.us.archive.org  
ia300933.us.archive.org  
ia300934.us.archive.org  
ia300935.us.archive.org  
ia300936.us.archive.org  
ia300937.us.archive.org  
ia300938.us.archive.org  
ia300939.us.archive.org  
ia300940.us.archive.org  
ia300941.us.archive.org  
ia300942.us.archive.org  
ia300943.us.archive.org  
ia301002.us.archive.org  
ia301003.us.archive.org  
ia301004.us.archive.org  
ia301006.us.archive.org  
ia301007.us.archive.org  
ia301008.us.archive.org  
ia301009.us.archive.org  
ia301010.us.archive.org  
ia301011.us.archive.org  
ia301012.us.archive.org  
ia301013.us.archive.org  
ia301014.us.archive.org  
ia301015.us.archive.org  
ia301016.us.archive.org  
ia301017.us.archive.org

ia301018.us.archive.org  
ia301019.us.archive.org  
ia301020.us.archive.org  
ia301021.us.archive.org  
ia301024.us.archive.org  
ia301025.us.archive.org  
ia301027.us.archive.org  
ia301028.us.archive.org  
ia301029.us.archive.org  
ia301030.us.archive.org  
ia301031.us.archive.org  
ia301032.us.archive.org  
ia301033.us.archive.org  
ia301034.us.archive.org  
ia301035.us.archive.org  
ia301036.us.archive.org  
ia301037.us.archive.org  
ia301038.us.archive.org  
ia301039.us.archive.org  
ia301040.us.archive.org  
ia301041.us.archive.org  
ia301042.us.archive.org  
ia301043.us.archive.org  
ia302002.us.archive.org  
ia302003.us.archive.org  
ia302004.us.archive.org  
ia302005.us.archive.org  
ia302006.us.archive.org  
ia302007.us.archive.org  
ia302008.us.archive.org  
ia302009.us.archive.org

ia302010.us.archive.org  
ia302011.us.archive.org  
ia302012.us.archive.org  
ia302013.us.archive.org  
ia302014.us.archive.org  
ia302015.us.archive.org  
ia302016.us.archive.org  
ia302017.us.archive.org  
ia302018.us.archive.org  
ia302019.us.archive.org  
ia302020.us.archive.org  
ia302021.us.archive.org  
ia302024.us.archive.org  
ia302026.us.archive.org  
ia302027.us.archive.org  
ia302028.us.archive.org  
ia302029.us.archive.org  
ia302030.us.archive.org  
ia302031.us.archive.org  
ia302032.us.archive.org  
ia302033.us.archive.org  
ia302034.us.archive.org  
ia302035.us.archive.org  
ia302036.us.archive.org  
ia302037.us.archive.org  
ia302038.us.archive.org  
ia302039.us.archive.org  
ia302040.us.archive.org  
ia302041.us.archive.org  
ia302042.us.archive.org  
ia302043.us.archive.org