

Exploring HPSS Bandwidth - NERSC Production Experience

Harvard Holmes (hhholmes@lbl.gov)

Mass Storage Group
Nancy Meyer (Group Lead),
Matthew Andrews, Shreyas Cholia, Damian Hazen,
Wayne Hurlbert, Nancy Johnston

National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

November 2003

Abstract

We have developed tools to help us analyze transfer logs from a production High Performance Storage System (HPSS) system at the National Energy Research Scientific Computing Center (NERSC). These tools provide graphical displays of data transfer performance over relatively short (24 hour or less) time spans.

One set of tools prepares graphs of individual transfers showing the data transfer rate versus the size of the file. Filters allow various classes of clients to be singled out to display traffic according to their network connectivity. Performance can be isolated down to individual users if desired. The effects of network links that limit transfer rates are easily observed on these graphs. The effect of TCPIP flow control algorithms is seen in the lower transfer rates of small files.

A second set of tools displays the transfers in time sequence. These tools assume a constant (average) rate of each transfer from the beginning to the ending time of each transfer. A typical day includes 40,000 to 100,000 transfers, and the display can be zoomed in to get time resolution down to 1 second or so. Additional tools summarize the individual transfers to derive overall system transfer rates when multiple transfers occur simultaneously. For these overall rates, both the peak and averages are shown. We are exploring these presentations to understand the amount of concurrency typical of our installation, and to gain insight into the peak performance required to sustain our desired average performance.

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division, U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Introduction to HPSS

HPSS was developed by a consortium of government, industrial, and educational sites [1] and is currently deployed at many large supercomputing sites. NERSC is a developer site within the consortium. HPSS provides a high performance data storage architecture based on the IEEE Mass Storage Reference Model [3, 4], which is designed to operate in a multiple-computer parallel configuration. The HPSS facility is a key component among the computational resources provided to NERSC users, responding to the increasingly data-intensive aspects of modern computational science.

NERSC has two HPSS systems ("Archive" and "Hpss"), both of which are accessible interactively from all NERSC supercomputers and auxiliary systems, as well as from off-site computers. Both systems are also accessible to batch jobs from all NERSC systems [2]. Of the two systems, user archive activity is encouraged on "Archive," and backup and other system activities are encouraged on "Hpss." Each system has an IBM AIX SP Winterhawk II node for file system management and other metadata operations, and a mixture of Winterhawk, Nighthawk and 6M1 nodes for data movement. The systems had 20 TB (Archive) and 9 TB (HPSS) of disk cache at the time of this study. The Archive system has 40 tape drives and the Hpss system has 35 tape drives. The drives used are StorageTek 9840 and 9940B drives in 8 StorageTek tape libraries. Collectively, the systems store about 800 TB of data in about 24 million files, and perform about 3 TB of client I/O per day [5].

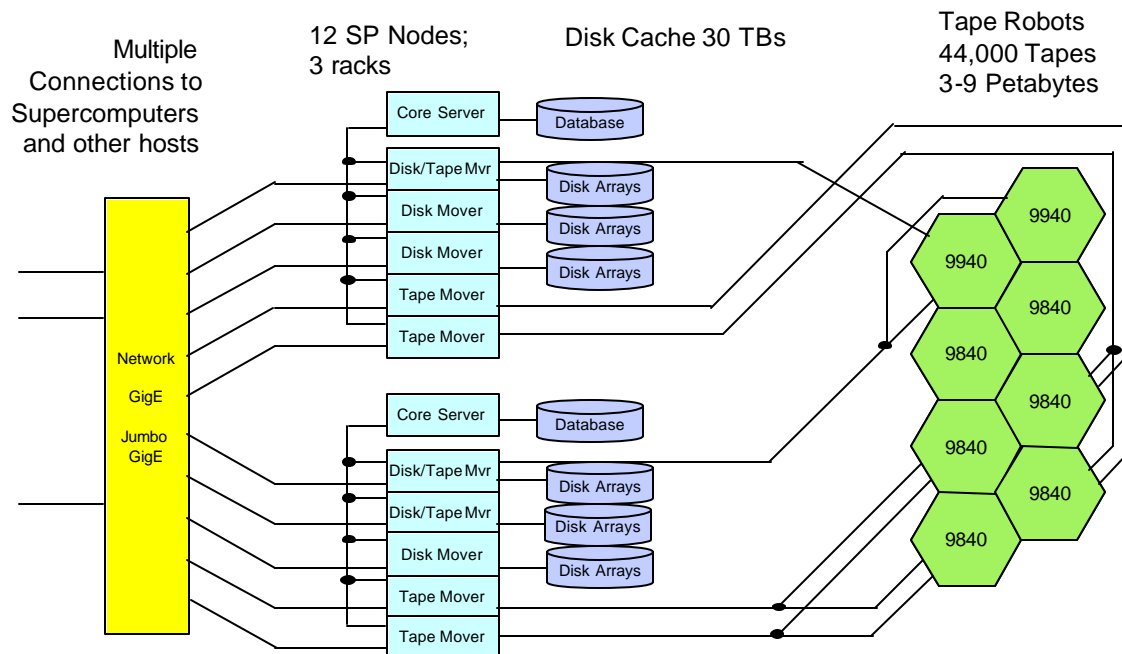


Figure 1. NERSC's Hardware Architecture

Data Collection

The primary interfaces to our HPSS systems are PFTP and HSI. PFTP is the HPSS supplied parallel FTP client, and HSI is the Hierarchical Storage Interface developed by Mike Gleicher with funding from several of the development sites. Both of these interfaces create both detailed log files and transfer log files as text files. At our site, we cycle the logs every 24 hours, so a 24 hour period is a very convenient analysis/display period. The detail logs record a great deal of internal operational data, while the transfer logs record one entry for each file transferred to or from a client. The transfer logs are saved and used for statistical summaries and to extract individual usage information. Each interface type creates its own distinct log file on the machine where it runs and the log files of different types and from the various machines are combined later. The transfer logs were deemed most accessible for this investigation. Information in the transfer logs includes the date and time of the transfer, the time taken for the transfer or the data rate, the host that was the client, the size and name of the file transferred, and the user who transferred the file (not necessarily the owner of the file).

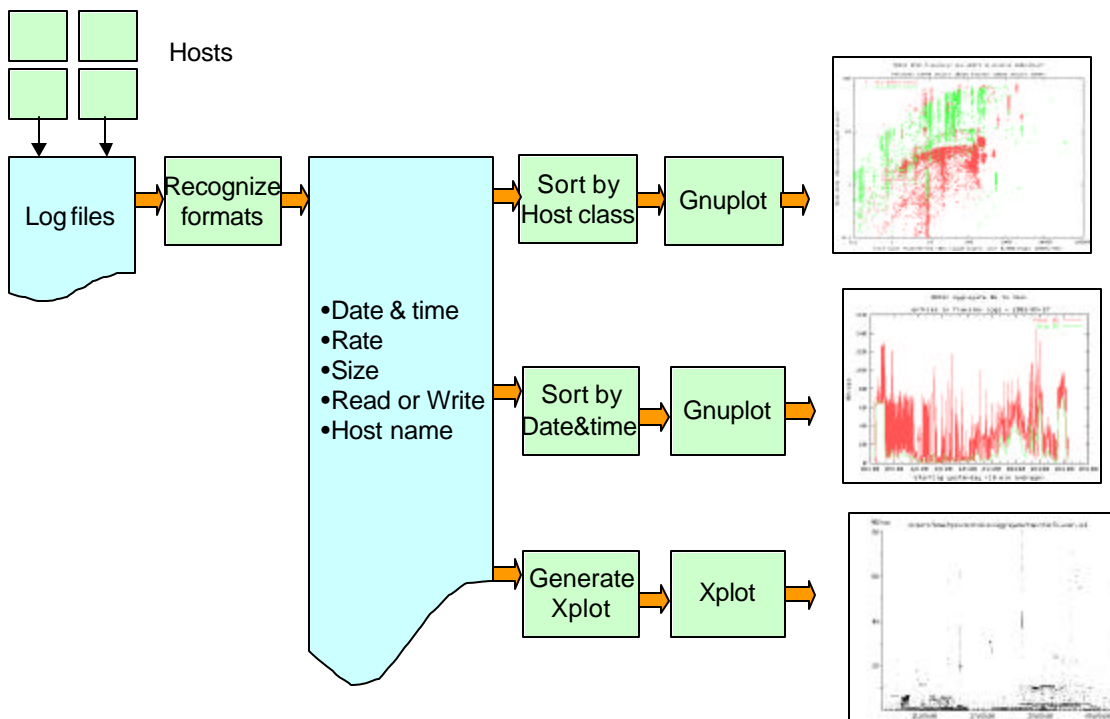


Figure 2. Data Processing Flow

Over time, the formats of the transfer records have had several changes and revisions. A PERL script was written to examine each record, deduce the format, and capture selected information in a standard form, including:

- the date and time of the transfer,
- the rate of the transfer,
- the size of the transfer,
- whether the transfer was a read or write,
- the name of the client,

- the name of the user, and
- the full path name of the transfer (occasionally only the name is available).

Data Display

For the display of data transfer rate versus file size, the data rates and file sizes are extracted from the transfer records, filtered according to the class of host and then graphed using gnuplot. Classes of hosts are established by specifying one or more IP network subnets. Most of our clients have multiple hosts that are assigned IP addresses from a contiguous block of addresses, so IP subnets are a convenient notation to describe sets of related hosts. File transfers are typically color coded to indicate whether they are fetching or storing data. Limits on the minimum and maximum file sizes and transfer rates may be specified to allow close examination of specific performance regions. Typically, only lower limits are specified for both rates and sizes to eliminate very small files and rates that are very low. Very low transfer rates are usually due to some misconfiguration and rarely provide any insight into high performance operations. The data graphs use logarithmic scales for both the file sizes and the transfer rates, due to the wide dynamic range of both of these data values. File sizes range from 1 byte up to several hundred gigabytes, over eleven orders of magnitude in range. Our lower limit for displaying file sizes is usually set to 0.1 MB. Transfer rates range from 1 byte per second up to 150 MB/sec for individual transfers. Our lower limit for displaying transfer rates is usually set to 0.1 MB/sec.

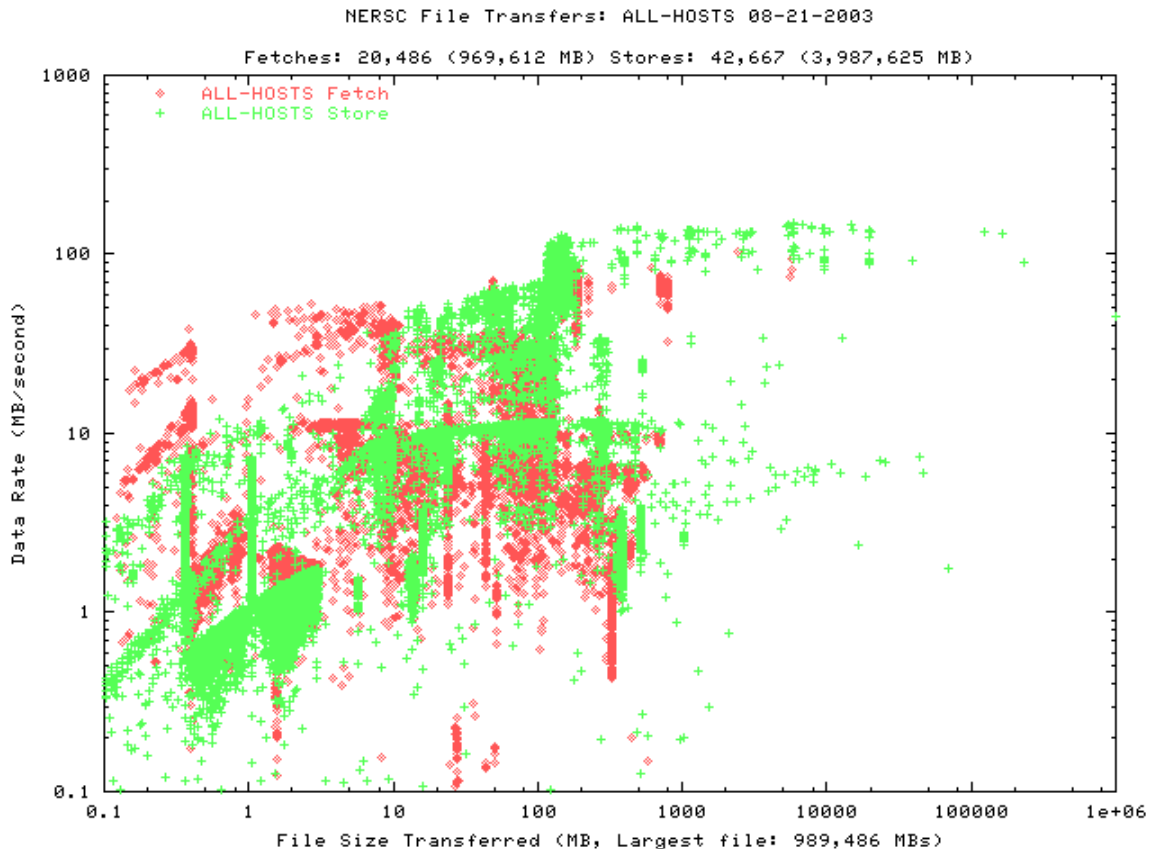


Figure 3. Example of Transfer Rate vs. File Size Plot

For a time based display, the records are formatted for input to Xplot, an interactive plotting program which can zoom in and pan to interesting portions of the display. Each transfer is represented as a horizontal line that starts at the beginning time of the transfer and ends at the ending time of the transfer. The y-value of the line corresponds to the average data rate during the entire transfer. From a 24 hour plot, one can observe a constant background transfer activity at a fairly low data rate, and occasional bursts of high data rate activity.

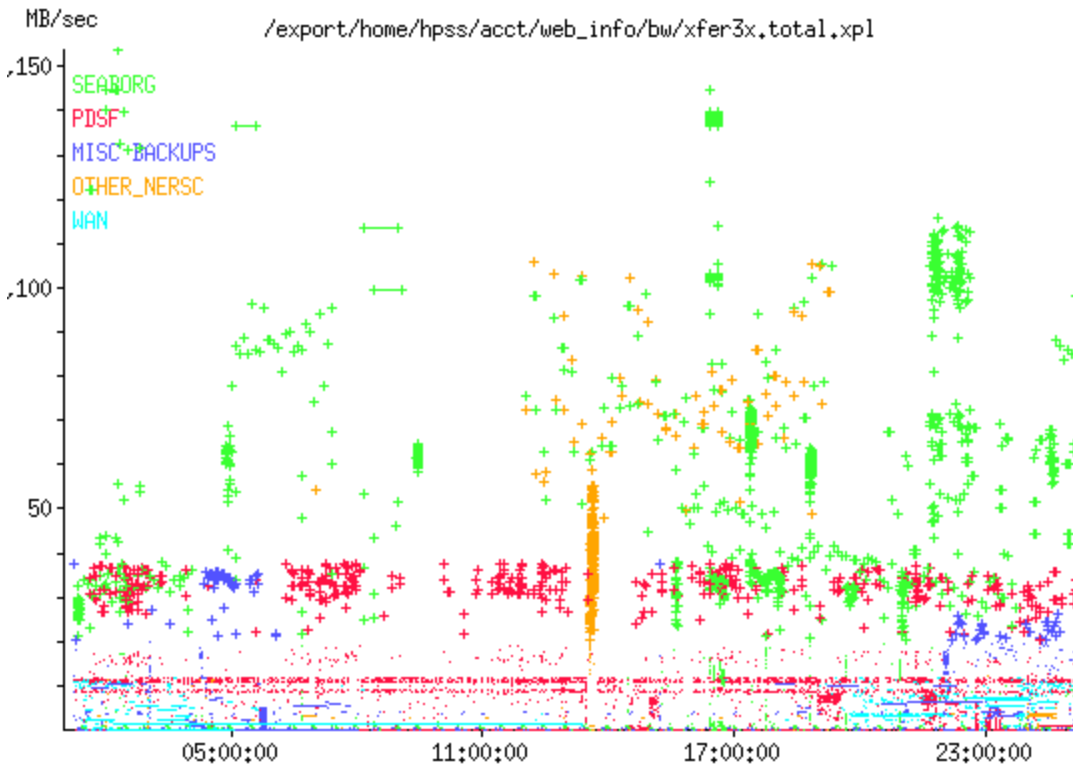


Figure 4. Example of Transfer Rate vs. Time Plot

A closer view of the activity along the bottom in the right half of the display above is shown below:

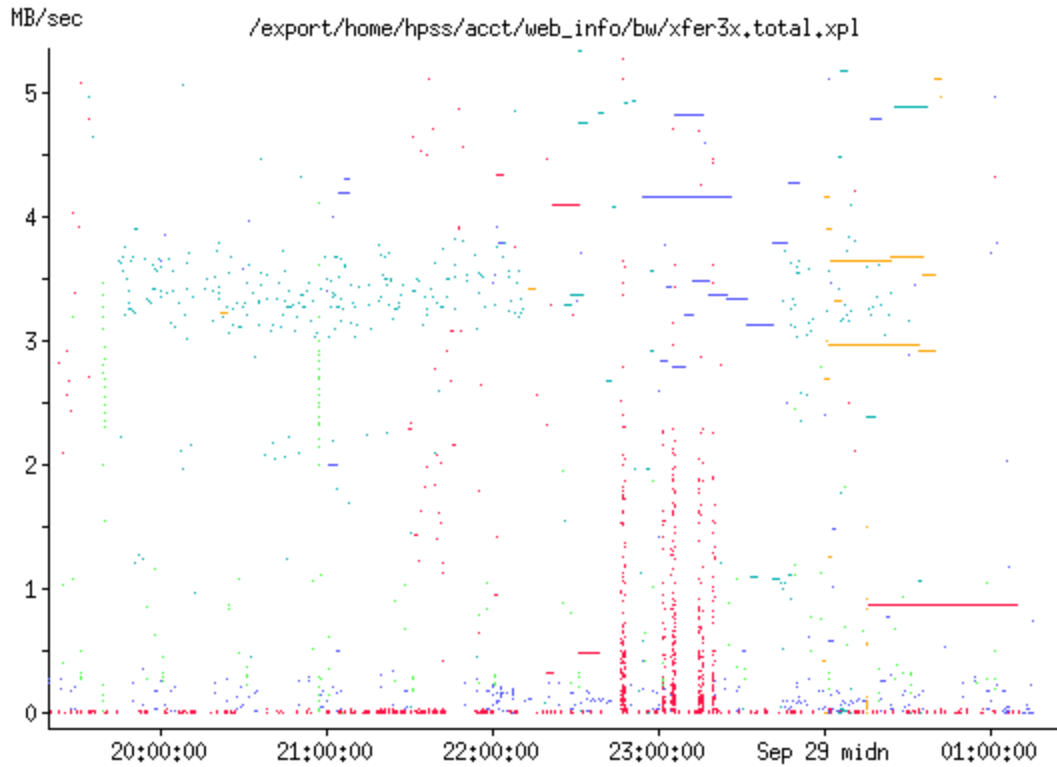


Figure 5. Zoomed View of Transfer Rate vs. Time Plot

From the more detailed view, it is possible to see a series of transfers at a data rate of about 3.5 MB/sec, and a few longer transfers at similar data rates.

The vertical spike at 16:25 can also be expanded horizontally to reveal that it is a 1 to 2 minute burst of short (1 to 2 seconds, 200 to 400 MB) file transfers that occur at a high data rate. At this magnification, it can also be seen that the files are all about the same size. The green color indicates that the transfers are from Seaborg, the IBM SP compute engine. Lower down, around 30 MB/s, we can see a second series of transfers from Seaborg taking place in parallel.

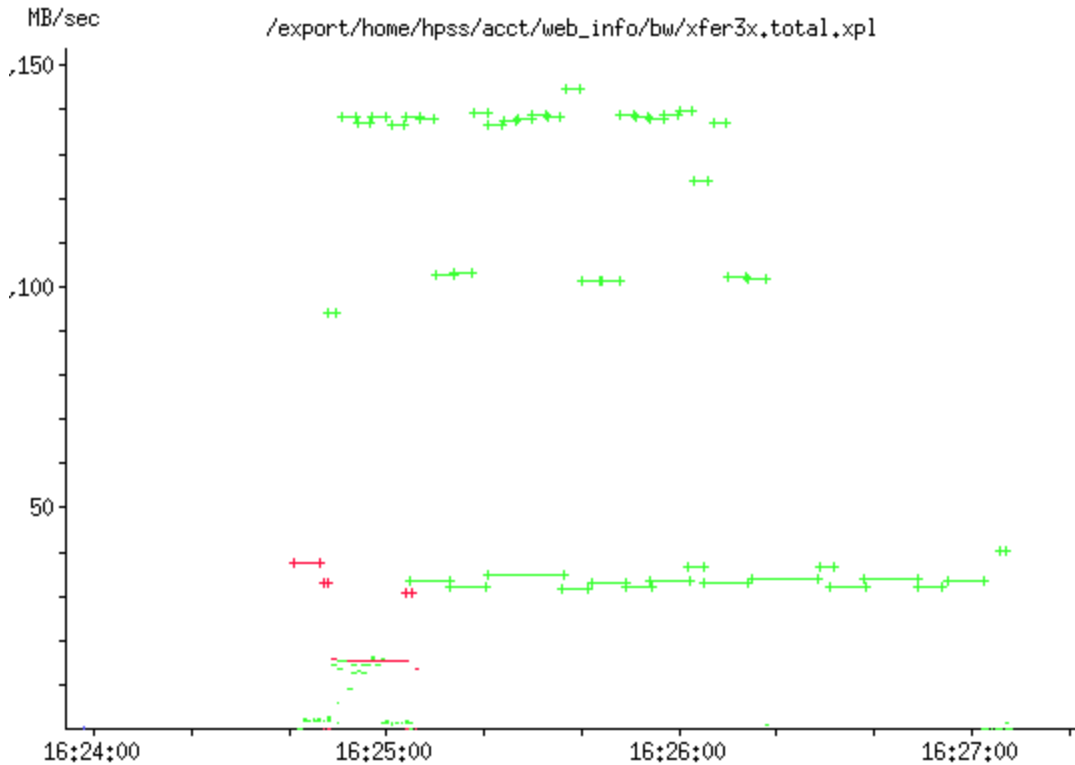


Figure 6. Horizontally Expanded View of Transfer Rate vs. Time Plot

Analysis of Bandwidth

The data shown in the plots of discrete transfers can be difficult to quantify due to the large number of individual transfers. Another tool reads the individual data transfers and provides a number of summary measures over user chosen intervals. This tool will keep track of all the transfers in progress at a given time and keep a running total of several measures of the transfers. For example, it keeps track of the aggregate transfer rate into and out of the storage system, adding to the rate when transfers start, and subtracting from the aggregate rate when transfers end. It maintains minimums, maximums and averages over user chosen intervals. Figure 7 is an overlay of two summary runs. It is produced by using a sampling interval of 1 minute, and then plotting the peak bandwidths seen during each minute (in red); together with a sampling interval of 10 minutes, and plotting the average bandwidth for each 10 minute interval (in green). Given the limited resolution of computer display systems, a 1 minute summary interval over a 24 hour span (1440 points) gives more resolution than can be displayed on most monitors.

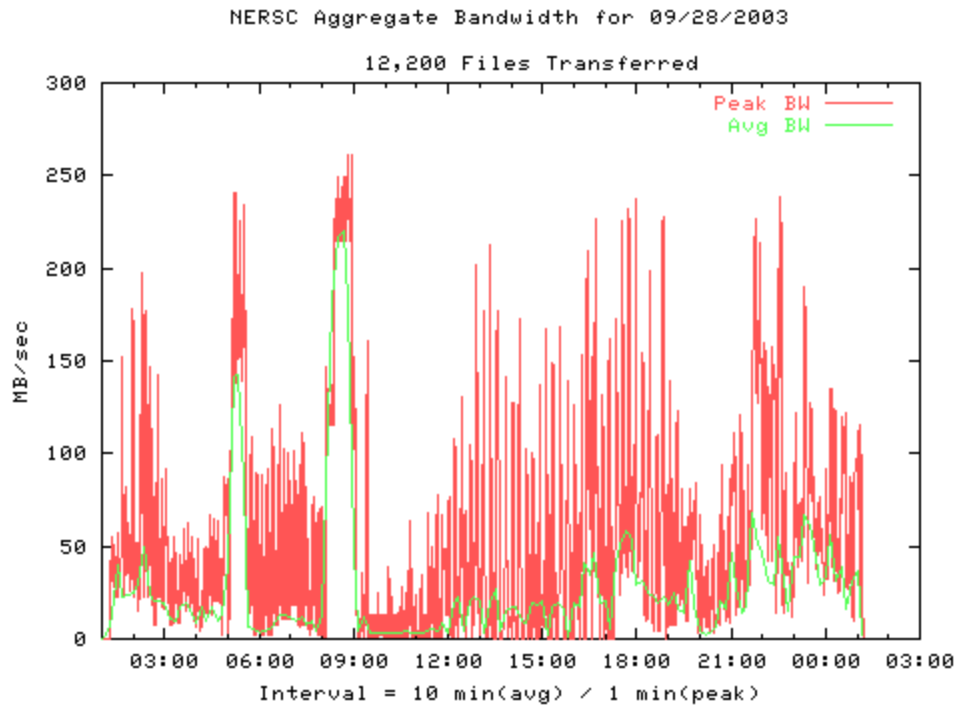


Figure 7. Statistics Based on the Transfer Rate vs. Time Plot

The detailed analysis of the discrete transfers is performed in an event driven manner. Figure 8 shows a simplified example of the data rate versus time plots. The plot has been greatly magnified so that only 7 transfers are shown. Event numbers are shown associated with the start and end of each transfer within this 10 second slice, and at the start and end of the 10 second interval itself. The last transfer does not end within the 10 second analysis period, so no event marker is given for the end of this transfer.

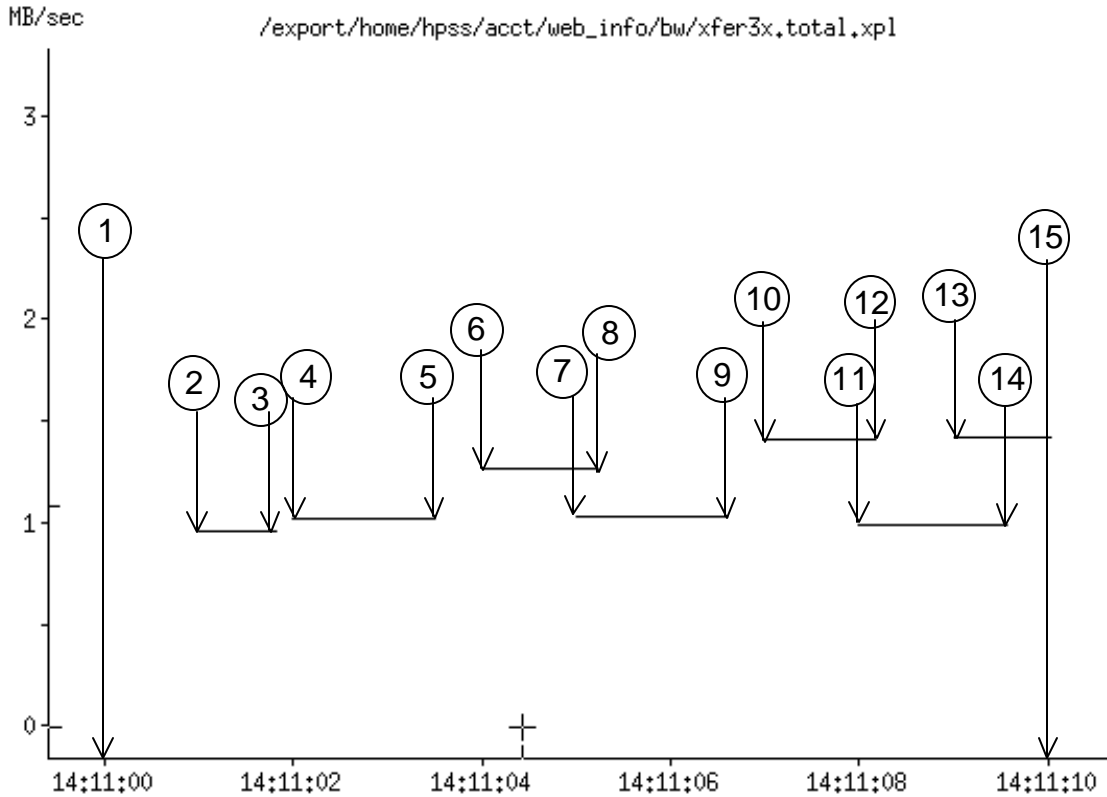


Figure 8. Simplified data rate versus time plot (greatly magnified).

The analysis is driven by a list of the start and end times of each transfer in the interval of interest (the summary interval). In this case, at time 1, the summary measures are all initialized. Event 2 is then fetched from the time ordered list of events. Summary measures are then computed for the time interval from event 1 to event 2. In this case:

- the count of transfers in progress is 0;
- the data rate of transfers in progress is 0;
- the minimum count of transfers is updated;
- the maximum count of transfers is updated;
- the minimum data rate is updated if necessary;
- the maximum data rate is updated if necessary, and if updated, the count of simultaneous transfers at peak rate is also updated.

Partial averages are also computed:

- the number of transfers in progress is weighted by the length of the interval, to be combined with other weighted intervals to get an average over the summary interval (this provides the average concurrency);
- the number of bytes transferred in this interval is calculated, to be combined with other counts, and then divided by the summary interval at the end to get the average rate.

Then the basic counts are updated from the event just fetched:

- if the event is a transfer start, then the count of transfers in progress is incremented;
- if the event is a transfer end, the count of transfers in progress is decremented;
- depending on whether the event is a transfer start or end, the current rate is increased or decreased;
- if the event is a transfer start, the count of transfers begun in an interval is incremented.

The analysis is then ready for the next event.

At the end of each summary interval, the summary statistics are written out, and the counter for transfers begun in a summary interval is reset to zero.

Observations

The effects of network bottlenecks can be observed in Figure 9. The bandwidth to the WAN is strongly limited to ~12 MB/sec from about 6 am to 8 am and a similar effect is seen in the later part of the bandwidth chart. The most likely explanation is that a particular user is behind a link with this speed, and he is performing sequential transfers. His average transfer performance is about half of the peak observed here; this would be consistent with smaller files with some interval between them.

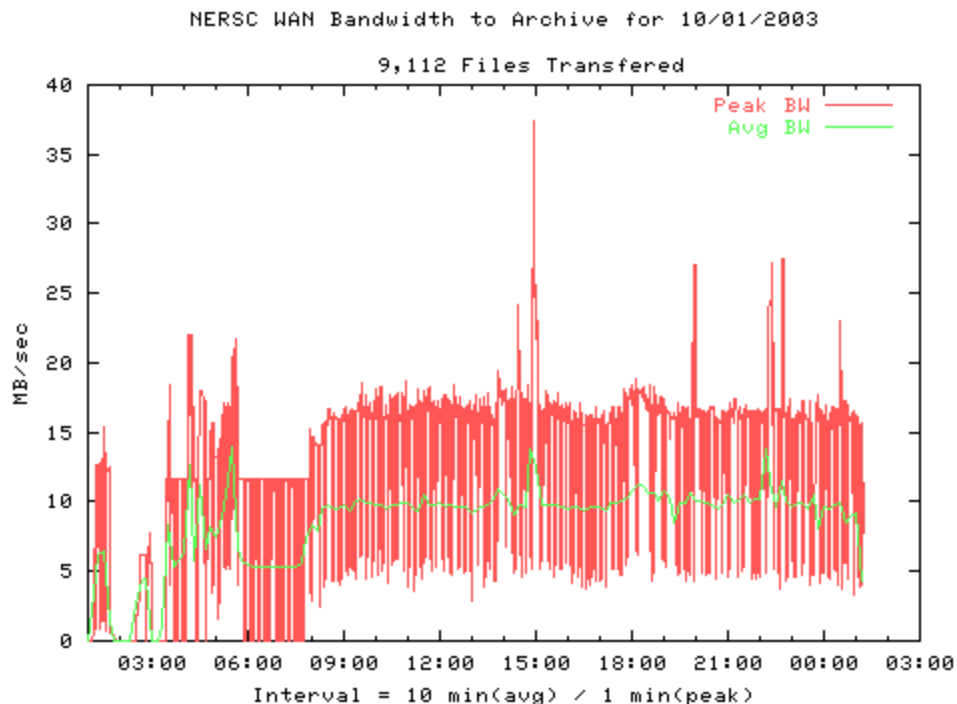


Figure 9. Network bottlenecks observed in WAN traffic.

Looking at the scatter plot of data rate versus file size for the same day, we see the same effect in figure 10, where the transfers (stores in this case) are limited at about 12MB/sec.

It only takes a file size of about 10 MB to reach this limit, which represents a transfer time of only 1 second at this rate.

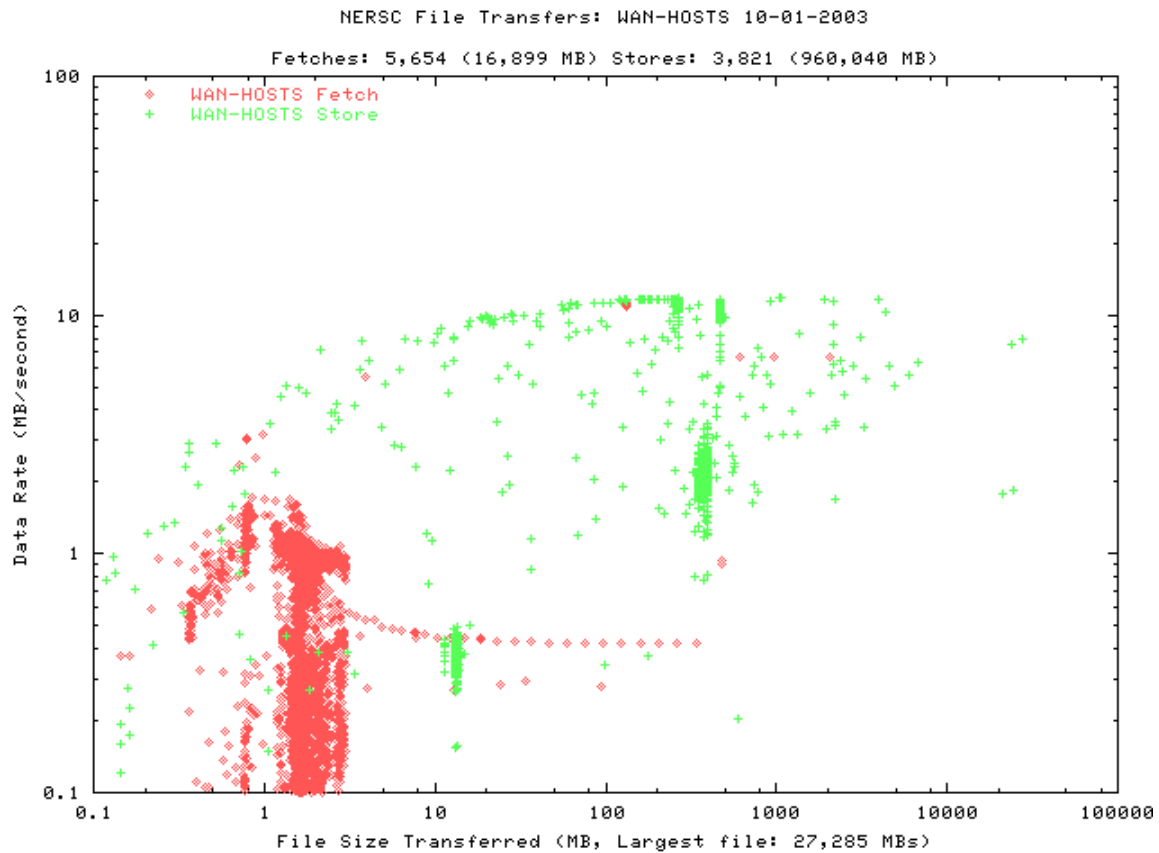


Figure 10. WAN transfer scatter plot, showing network limitations at 12 MB/sec.

A second type of observation that can be made is to confirm that the storage system has enough "headroom" for future growth demands. For that, we look at bandwidth plots for the overall system, especially on days with heavy loads. In Figure 11, we see that the peaks in the bandwidth are up near 500MB/sec, while the average transfer rates tend to hover at or below 200MB/sec. This indicates adequate bandwidth to meet the demand.

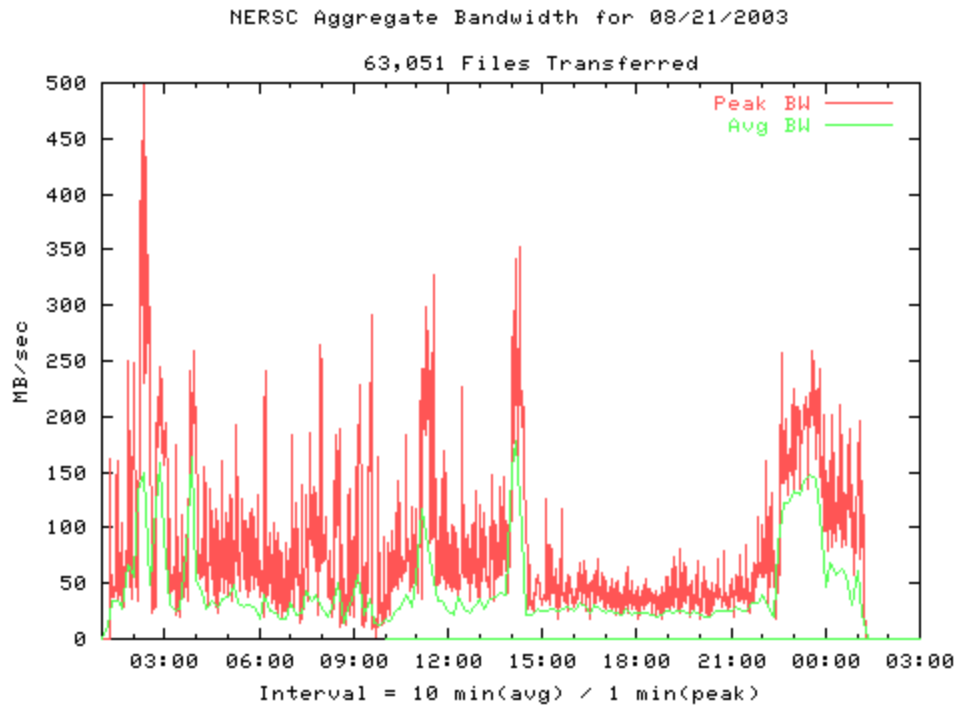


Figure 11. Aggregate bandwidth on a heavily loaded day.

To confirm that we are able to meet the demand, we can also look at concurrencies, where, if the systems were not able to keep up with demand, the number of concurrent transfers would gradually increase during periods of heavy load, as the systems fell behind the demand. Figure 12 shows no evidence of this.

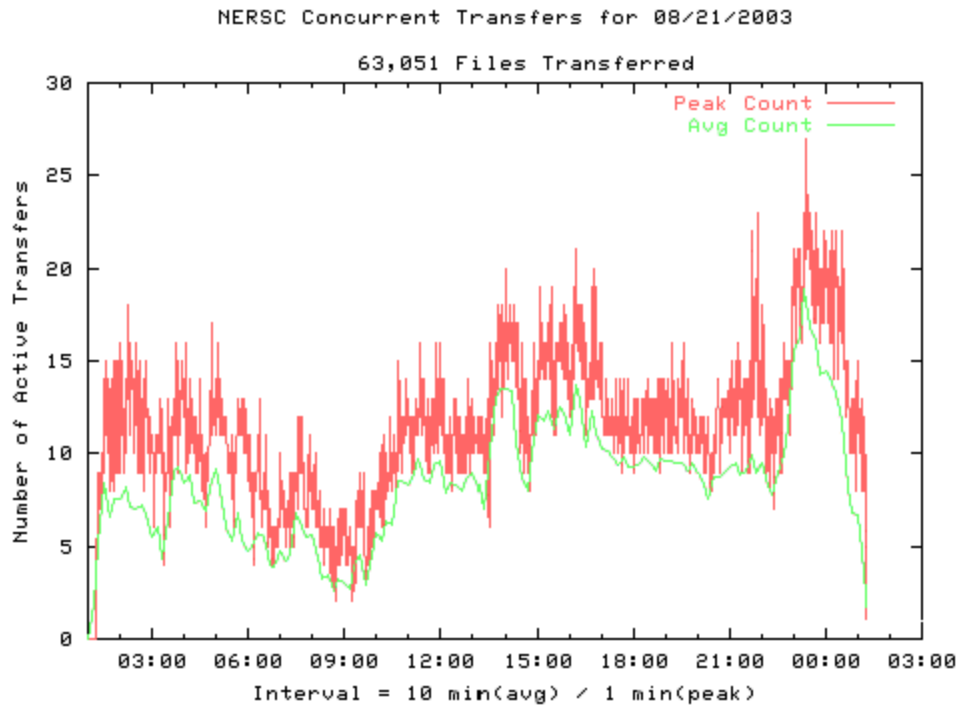
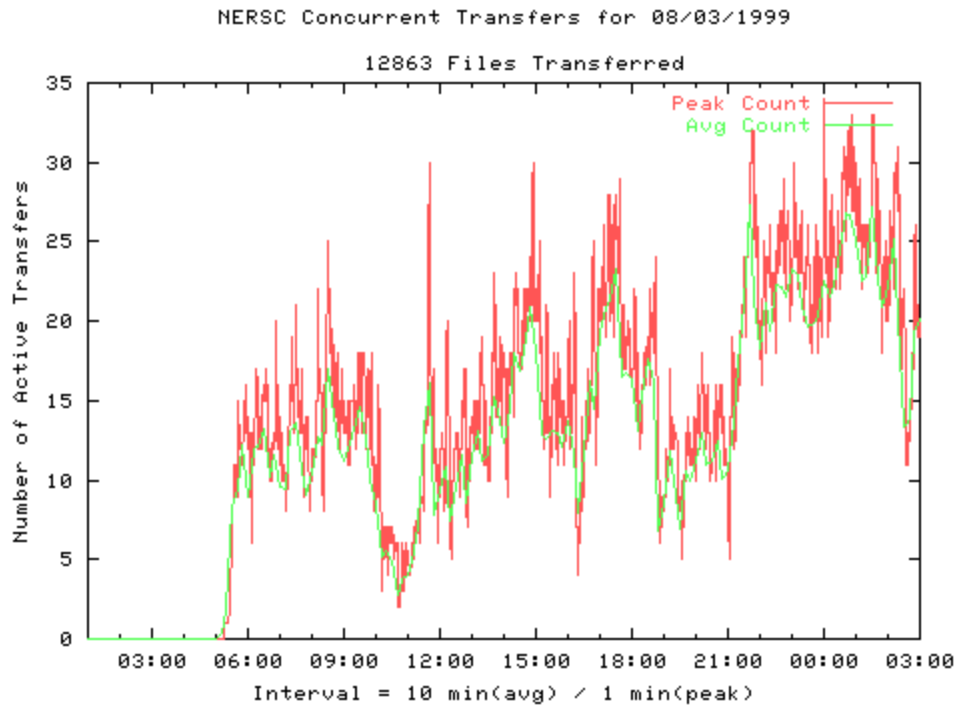


Figure 12. Concurrency under heavy loads.

A third type of observation comes from the historical record. The aggregate bandwidth plots over the years look remarkably similar to the aggregate band width shown in Figure 11, with the data rate declining by almost 2x per year as we look back in time. The degree of concurrency has also remained remarkably stable over the years: the number of concurrent users has remained stable, while a very minor tendency is observed for the average concurrency to be slightly higher relative to the peak concurrency for earlier years. This effect is shown in Figure 13, for 1999, which can be compared to Figure 12 for 2003.



Conclusions

These graphs, together with other performance characterizations, provide a good way to monitor performance to ensure it meets the demand; and they also provide some amount of problem determination support as well. We use them to monitor overall performance as well as to identify specific networks that are troubled by low performance.

Acknowledgement

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division, U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

References

1. <http://www.sdsc.edu/hpss/>
2. <http://hpcf.nersc.gov/storage/hpss/>
3. R. A. Coyne and H. Hulen, "An Introduction to the Mass Storage Reference Model, Version 5," *Proc. 12th IEEE Symp. Mass Storage*, Monterey, CA, IEEE Computer Society Press, April 1993.
4. IEEE Storage Systems Standards Working Group public documents, http://www.ssswg.org/public_documents/MSSRM/V5toc.html
5. <http://hpcf.nersc.gov/storage/hpss/stats/>