Recent Improvements in the NASA Technical Report Server

Ming-Hokng Maa <m.h.maa@larc.nasa.gov>
Michael L. Nelson <m.l.nelson@larc.nasa.gov>

NASA Langley Research Center
October, 1995

Abstract

The NASA Technical Report Server (NTRS), a World Wide Web report distribution service, is modified to allow: 1) Parallel database queries, significantly decreasing user access times by an average factor of 2.3; 2) access from clients behind firewalls and/or proxies which truncate excessively long Uniform Resource Locators (URLs); 3) Access to non-Wide Area Information Server (WAIS) databases, and compatibility with the Z39-50.3 protocol.

1.0 Introduction

The original NASA Technical Report Server (NTRS) went on-line June 6, 1994 to facilitate greater distribution of NASA technical publications via the World Wide Web [Nelson, 1995]. This implementation, however, had three shortcomings:

* Slow user access times.
* Non-compatibility with clients behind firewalls/proxies.
* Non-compatibility with non-WAIS databases.

Because the original NTRS queried each Wide Area Information Server (WAIS) database sequentially, users of NTRS often experienced slow performance, often of 2 minutes or more for querying all available NTRS databases. Databases such as the Astrophysics Data System (ADS) and the CASI Technical Report Server (RECONselect) were especially slow due to large databases and other factors. In addition, users behind firewalls or proxies were unable to access abstracts in NTRS. Specifically, the CERN httpd proxy server canonicalizes Uniform Resource Locators (URLs), with the effect of limiting the length of URLs that can be passed through the firewall [Frystyk, 1995]. This results in clients attempting to access incorrect NTRS URLs that have been truncated by their proxy server. This problem is especially relevant with WAIS URLs, which can exceed 400 characters. Finally, the original NTRS was not compatible with non-WAIS query syntax. Protocols such as Z39-50.3, used in ADS, are not supported [Eichorn, 1995]. Because of these shortfalls, the original NTRS is modified to allow parallel database searches, compatibility with clients behind proxies/firewalls, and gateway compatibility with non-WAIS databases. The advantages of WAIS and non-WAIS databases are not discussed here, but can be found in [Accomazzi, 1995] and [Marchionini, 1994]. NTRS is still available at:

```
http://techreports.larc.nasa.gov/cgi-bin/NTRS
```

2.0 Original NTRS Architecture

The original version of NTRS is a simple Perl script acting as a single interface to the many WAIS databases implemented by various NASA programs and centers. These databases are based on the Langley Technical Report Server [Nelson, 1994]. The fundamental principle of NTRS is to have a logically central interface, but physically distributed implementation. Thus the single NTRS script provided the illusion of integrated access to various WAIS databases.

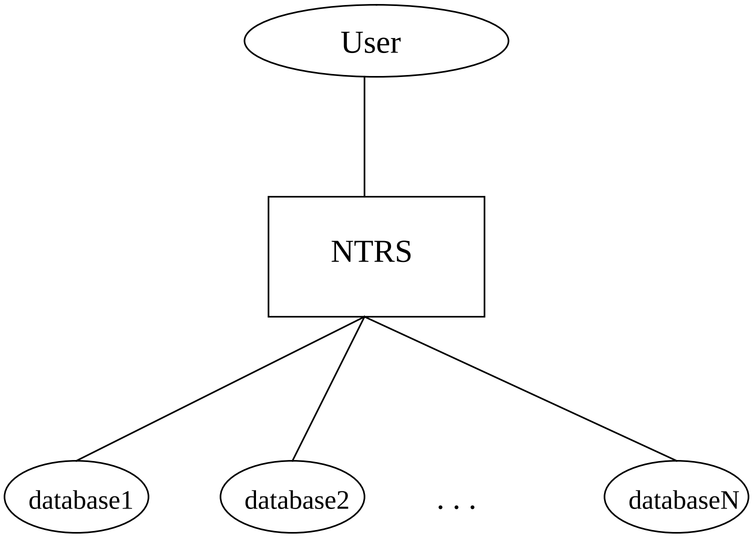

Figure 1: Architecture of sequential NTRS

This implementation searches the databases sequentially. Originally, this was not a problem since the number of databases was small. But as additional NASA centers began to add their report servers, there was a need for parallel searching. Originally, few users were behind firewalls or proxies, so the URL limit went undiscovered for some time. Increased commercial usage (".com" addresses, which are typically behind firewalls) brought this problem to light. Finally, although ADS supports an experimental WAIS database, the bulk of their development is in Z39-50.3 (non-WAIS).

3.0 Revised NTRS Architecture

NTRS was revised to address the current state of: more databases, users behind firewalls, and non-WAIS databases. NTRS has lost some of its simplicity, but it now has greater flexibility.

NTRS presently consists of three main parts (All written in Perl 5.000):

* User interface cgi-bin script
* 10 Database Servers, one for each database
* URL Decompression cgi-bin script

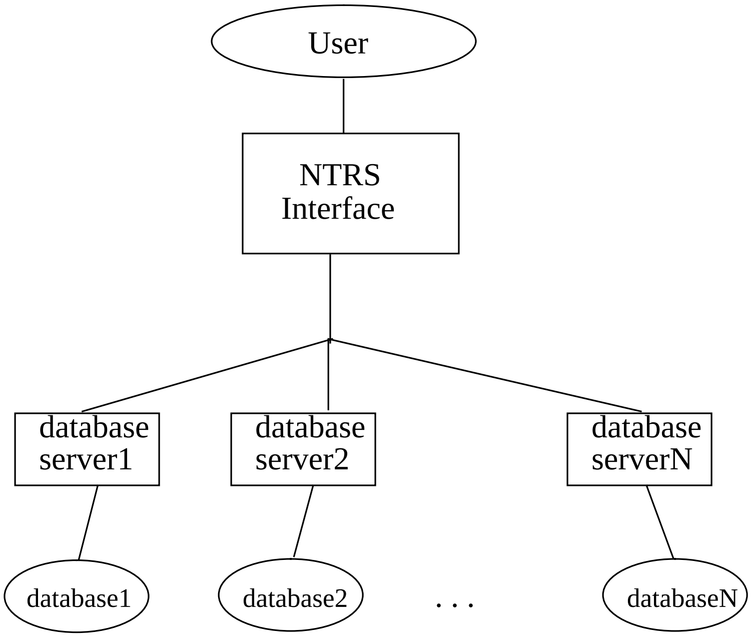

Figure 2: Architecture of parallel NTRS

3.1 User interface cgi-bin script

The user interface script is the front-end to NTRS. Database queries for each database are passed sequentially by the user interface script to the appropriate database server via sockets. The user interface script then waits for the database servers to return the query results in the same order in which the queries were sent and prints the results to the user.

3.2 Database Servers

At first, two different methods were evaluated to implement the parallel search method: *threads* and *forks*. Although a threaded version had greater aesthetic appeal, it is more complicated to implement and maintain, and thus a forking version was chosen.

Currently, CERN's httpd 3.0 has the canonicalizing feature built in, resulting in URL truncation. A patch is available to turn off the canonicalizing feature and future releases of CERN httpd will remove URL truncation. However, NTRS will keep URL compression in operation since it is not known when all copies CERN httpd will be upgraded. Also, other firewall/proxy implementations may have similar problems.

During testing, there were 10 database servers, one for each searchable database used by NTRS. The databases servers accept client requests from the interface script. Once a request has been made, the server forks off the query to the host database and waits for the database to return the results. Once the results are collected and parsed, the server passes the query results back to the interface script for the user to view. By forking off the user queries, a query will be searched across several databases in parallel.

The server also compresses the WAIS URLs returned by the query search for users behind proxies or firewalls. Because typical WAIS URLs often contain a great deal of redundancy such as similar filepaths common to all elements in the databases, parts of the URL that are common to all documents can be replaced with a shorter token. On average, this compresses the size of the URLs by half, short enough to pass through proxies unharmed. The compression script also allows the retrieval to be done on the standard http port (80), thus bypassing other difficulties.

Another advantage of separating the database servers into separate components is that it allows non-WAIS handling to be built in. This would have been too difficult to build into the original NTRS script.

3.3 URL Decompression Script

When users attempt to access a URL that has been compressed, the compressed version of the real URL is detoured first to the decompression script, which takes the compressed URL and decompresses the URL back to its original length and content. The decompression script then fetches and displays the document for the user. All of this is transparent to the user.

4.0 Results

Timings were taken for both the original NTRS and the new NTRS to confirm that the parallel search method produced faster results than the original sequential search method. The search words: “*engine turbulence*” were used. The average access time will differ depending on the number of search words queried, and the frequency of the words in the database. Ideally, if all of the NTRS databases were approximately the same speed, then the parallel method should show marked improvement. In contrast, if there were one or two slow databases while the remaining databases were relatively fast, then the parallel method would be bottlenecked by the one or two slow databases, and performance would not be as great. Tests showed that both ADS and RECON were especially slow databases, around 40+ seconds for access (Figure 3). The rest of the databases were relatively fast, mostly under 10 seconds. Therefore, timing tests for both the sequential and parallel methods were undertaken both with and without ADS and RECON. Table 1 shows the averages for each test. In total, six tests were performed through two weeks every hour:

1. Sequential NTRS accessing all 10 databases
2. Sequential NTRS accessing all databases except ADS and RECON
3. Parallel NTRS accessing all 10 databases
4. Parallel NTRS accessing all 10 databases except ADS and RECON
5. Parallel NTRS accessing all 10 databases with proxy compression
6. Parallel NTRS accessing all 10 databases except ADS and RECON with proxy compression

Table 1: Data Summary: Average user access times

Method	All Databases	Without ADS & RECON
Sequential	142.4 s	52.2 s
Parallel	60.3 s	22.1 s
Parallel with Proxy	87.8 s	19.7 s

The results are graphed in figures 4 - 7. The tests were run between July 17, 1995 - July 31, 1995. The data presented does not include the optional URL compression. The tests made no attempt to compensate for network and transient conditions which would impact the timings. Table 2 gives information about the various databases.

Table 2: Information about WAIS databases

database	hostname of WAIS server	location of WAIS server	abstracts in database
NAS	www.larc.nasa.gov	Hampton, VA	150+
ADS	adswais.harvard.edu	Cambridge, MA	160,000+
Dryden	www.dfrc.nasa.gov	Edwards, CA	650+
GISS	www.larc.nasa.gov	Hampton, VA	550+
ICASE	www.icase.edu	Hampton, VA	150+
Langley	techreports.larc.nasa.gov	Hampton, VA	550+
Lewis	letrs.lerc.nasa.gov	Cleveland, OH	950+
NACA	www.sti.nsa.gov	Linthicum Heights, MD	13,000+
RECON	www.sti.nsa.gov	Linthicum Heights, MD	2,000,000+
SCAN	www.sti.nsa.gov	Linthicum Heights, MD	2,000+

5.0 Discussion

* There is variation between maximum and minimum access times. This could be due to many different reasons:

- Timings were only collected for 2 weeks.
- The access hours for the databases unfortunately are not very well distributed.
- Each database has its own characteristics including size of database, connection speed, host machine load, etc.

* There seems to be a peak around 4 a.m. Other than that, all other high load times are during the afternoon as expected. The peak around 4 AM may be due to cron processes running on the server or maybe even accesses from Europe, though the actual reason for the peak remains unknown.

* The differences in minimum and maximum access times for parallel NTRS without ADS and RECON is fairly small. This positive result is expected from the search algorithm.

* Parallel methods without the proxy compression have consistently improved access times by a factor of 2.3 for timings without ADS and RECON and timings with all 10 databases.

* The fastest sequential search is slower than the slowest parallel search. This holds for searches including ADS and RECON, and those excluding them.

6.0 Future Directions

NTRS continues to evolve. Since the time of the redesign and testing, Goddard Space Flight Center, Kennedy Space Center and Stennis Space Center have joined NTRS. In addition, the WAIS version of ADS is no longer linked from NTRS; the Z39-50.3 version of ADS is used in its place. Future Z39-50.3 databases, such as the Space Instrumentation Abstract Service, will be added to NTRS in the near future.

The new database server components now allow for tailoring the pre- and post-processing for each database. Small syntax differences between freeWAIS, commercial WAIS, databases with fielded searches, non-WAIS databases, etc. can now be easily hidden from the user. Additionally, post-processing of the data, such as highlighting the keyword search terms is now also possible.

7.0 Conclusions

The parallel query method is faster than the original sequential query method by a factor of 2.3, or less than half the access time required. In addition, the compression method used to solve the proxying/firewall problem was implemented and user feedback indicates that it performs satisfactorily. NTRS also now includes an interface with its first non-WAIS database. The completed redesign of NTRS provides many performance enhancements and has the necessary hooks for future improvements. Contact m.l.nelson@larc.nasa.gov to obtain source code for NTRS.

Acknowledgments:
Many thanks to the ICE team (http://ice-www.larc.nasa.gov/) at NASA Langley Research Center.

References

Accomazzi, A.; Murtagh, F.; Rasmussen, B. F.: “Information Retrieval Tools and Techniques,” Library & Information Services in Astronomy II Conference Proceedings, European Southern Observatory, Garching, Germany, May 10-12, 1995.

Eichorn, G.; Accomazzi, A.; Grant, C. S.; Kurtz, M. J.; Murray, S. S.: “Access to the Astrophysics Science Information and Abstract System,” Library & Information Services in Astronomy II Conference Proceedings, European Southern Observatory, Garching, Germany, May 10-12, 1995.

Frystyk, Henrik: “Known Bugs in httpd”, http://www.w3.org/hypertext/WWW/Daemon/Bugs.html , April 1995.

Marchionini, G.; Barlow, D.: “A Comparison of Boolean-Based Retrieval to the WAIS System for Retrieval of Aeronautical Information,” NASA CR-4569, March 1994.

Nelson, M. L.; Gottlich, G. L.; Bianco, D. J.: “World Wide Web Implementation of the Langley Technical Report Server,” NASA TM-109162, September 1994.

Nelson, M. L.; Gottlich, G. L.; Bianco, D. J.; Binkley, R. L.; Kellogg, Y. D.; Paulson, S. S.; Beaumont, C. J.; Schmunk, R. B.; Kurtz M. J.; and Accomazzi, A.: “The Widest Practicable Dissemination: The NASA Technical Report Server,” Computers in Aerospace 10, AIAA-95-0964, San Antonio TX, March 28-30, 1995.

Timing Graphs

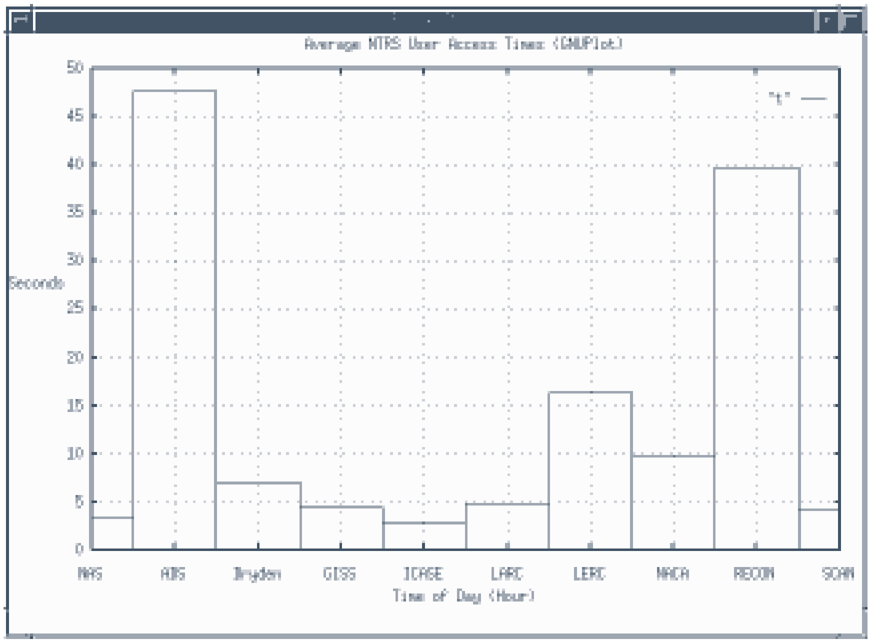

Figure 3. Individual access times for each NTRS database.

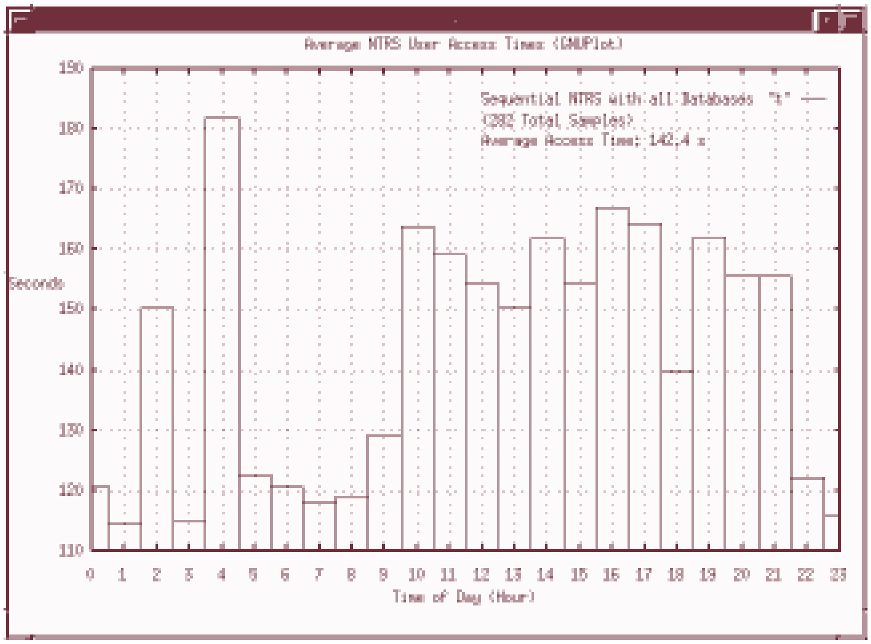

Figure 4. Sequential Search w/ all Databases

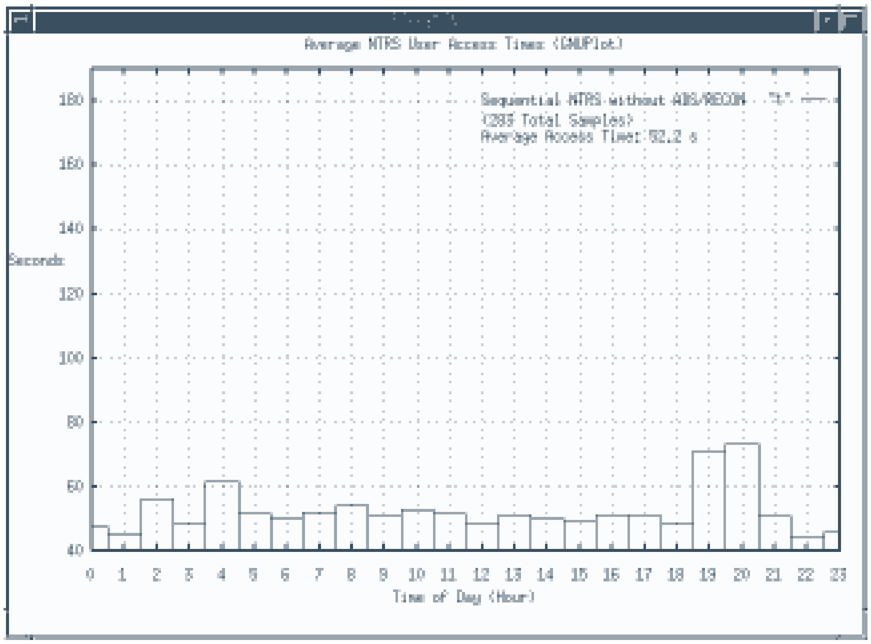

Figure 5. Sequential Search w/ all Databases except ADS and RECON

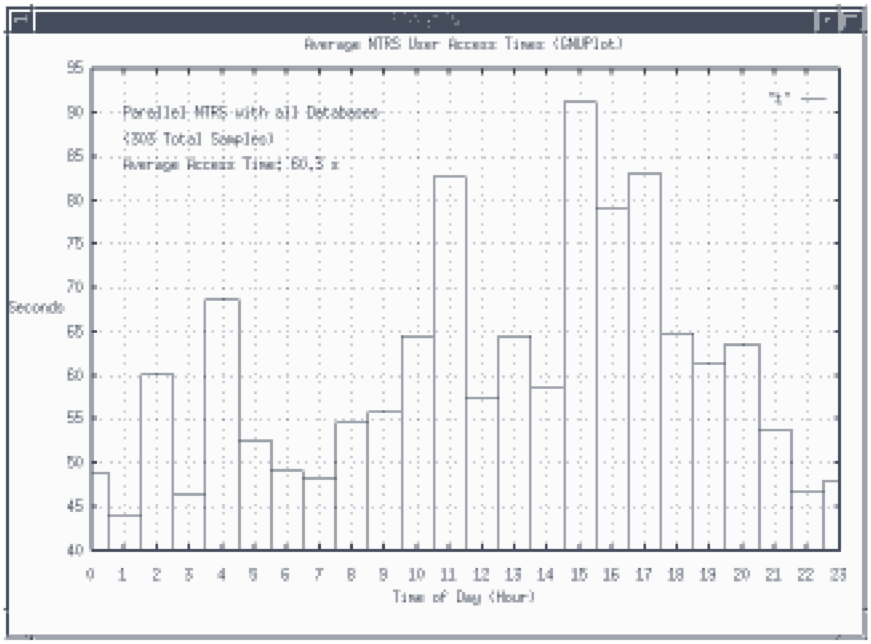

Figure 6. Parallel Search w/ all Databases

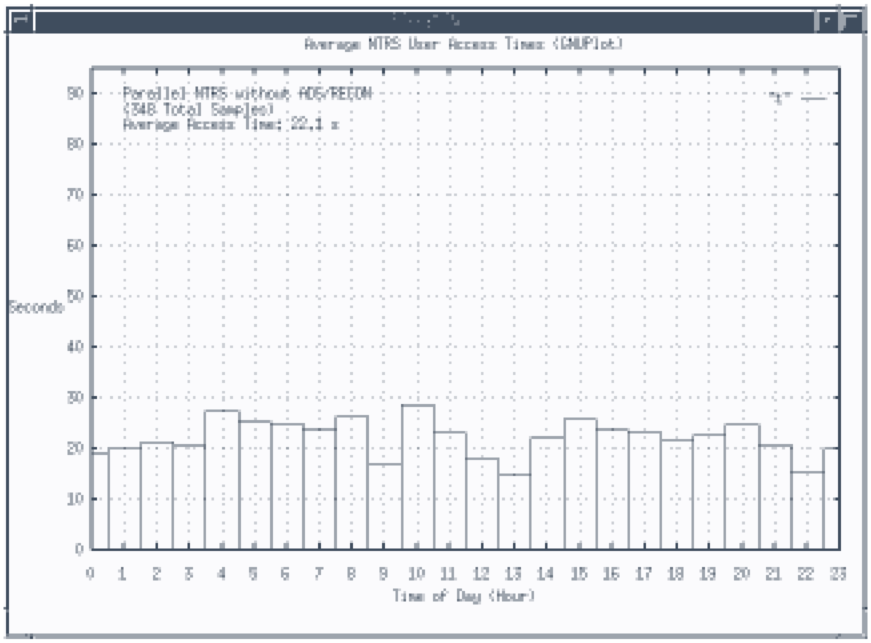

Figure 7. Parallel Search w/ all Databases except ADS and RECON