Front. Earth Sci. 2011, 5(2): 111–119
DOI 10.1007/s11707-011-0152-7

RESEARCH ARTICLE

Federated or cached searches: Providing expected performance from multiple invasive species databases

Jim GRAHAM (✉)[1], Catherine S. JARNEVICH[2], Annie SIMPSON[3], Gregory J. NEWMAN[4], Thomas J. STOHLGREN[2]

1 Natural Resource Ecology Laboratory, Colorado State University, Fort Collins, CO 80523-1062, USA
2 United States Geological Survey Fort Collins Science Center, Fort Collins, CO 80523-1062, USA
3 United States Geological Survey Headquarters, Reston, VA 11750, USA
4 Natural Resource Ecology Laboratory, Colorado State University, Fort Collins, CO 80523-1062, USA

Abstract Invasive species are a universal global problem, but the information to identify them, manage them, and prevent invasions is stored around the globe in a variety of formats. The Global Invasive Species Information Network is a consortium of organizations working toward providing seamless access to these disparate databases via the Internet. A distributed network of databases can be created using the Internet and a standard web service protocol. There are two options to provide this integration. First, federated searches are being proposed to allow users to search "deep" web documents such as databases for invasive species. A second method is to create a cache of data from the databases for searching. We compare these two methods, and show that federated searches will not provide the performance and flexibility required from users and a central cache of the datum are required to improve performance.

Keywords design, performance, invasive species, web services, databases, federated search, search engine

1 Introduction

The United States loses an estimated $120 billion per year in control programs and in reduced agricultural productivity due to invasive species (Pimentel et al., 2005). Invasive species are also viewed as the second greatest cause of decline in species diversity (Wilcove et al., 1998). Invasive diseases impact human health and are a global problem (Mack et al., 2000), as evidenced by frequent news reports of Asian bird flu, West Nile Virus, or Severe Acute Respiratory Syndrome (SARS). Information on the distribution of these non-native plants, animals, and pathogens is largely held in widely disparate formats ranging from taxa-specific or region-specific databases to paper documents (e.g., treatment records and field data sheets).

With the technological improvements over the last decades, the amount of electronic data has greatly increased. Great advances have been made recently in making biologic data sets interoperable and discoverable over the Internet (e.g., see the Global Biodiversity Information Facility, GBIF, at http://www.gbif.org). Databases with invasive species information have also proliferated in the last few years. For example, over 150 online databases with data on invasive species have been identified in the United States alone (Crall et al., 2006). However, these databases are stored in different database programs (e.g., Microsoft Access, SQLServer, PostgreSQL), with different database structures, and are located throughout the world both on and off line (Halpin et al., 2006; Graham et al., 2008).

There is a need among invasive species scientists to cross-search of these disparate data sources (Casal 2006; Graham et al., 2008). Search engines such as Google and Yahoo allow users to search a large number of Web pages quickly and with great flexibly. However, with these types of searches it is difficult to extract information from a specific domain, such as invasive species locations or invasive species classifications for an area. Additionally, since these engines only search Web pages they can miss data within a database or can cause significant loading on Websites to index each unique page. For example, if there are Web pages with information for 10000 species, it would index the Web pages 10000 times with one index for each of the species.

Received February 14, 2011; accepted March 2, 2011

E-mail: jim@nrel.colostate.edu

Another approach to provide access to disparate databases is to create a Web service that allows one computer to query a database on another computer and extract information from that database. Web services can then be used to exchange data throughout the Internet between different databases.

Within the public sector it is common to use Web Services to exchange data over the Internet (Fan and Kambhampati, 2005). These services use a common language, or protocol, to allow for the exchange of data in a reliable manner. Protocols have also been developed within the life sciences, especially to exchange genetic data (Curcin et al., 2005). Several protocols have been developed to share biologic data, such as the Distributed Generic Information Retrieval (DiGIR) system (Vieglais, 2006), TDWG Access Protocol for Information Retrieval (TAPIR) (de Giovanni et al., 2006), and the Global Invasive Species Information Network (GISIN) Protocol (Global Invasive Species Information Network; Graham et al., 2008).

While the protocol and design of data exchange has been explored (Martens, 2005), the performance of these approaches has not. Since biologic data can amount to tens of millions of records (e.g., see GBIF) and users on the Internet expect very fast responses, the performance of any system is critical to its success. Some existing portals, such as NISBase (http://www.nisbase.org), use a federated search, while others, such as GBIF (Global Biologic Information Facility; Edwards 2004), cache data to improve performance. In a federated search, each time a user makes a request, all the individual databases are searched sequentially and the combined results presented (Fig. 1(a)) (Jacsó, 2004). In a cached approach, key information describing the data from the databases is regularly queried and stored in a central location (Fig. 1(b)) (Fox et al., 2004). This allows a user's request to be queried from a single location.

We hypothesize that a cached approach should provide better performance with large numbers of databases, but at the cost of having to build and maintain the cache. Either approach can provide searches based on a particular knowledge domain and the results displayed to the user can include links to the original source of the data. Here, we analyze the performance of these two approaches using a Web service developed for invasive species data as a part of the GISIN.

The GISIN is being developed to exchange invasive species data between existing online databases throughout the world. This integration is critical to improving management of invasive species, and to prevent further invasions because a species invasive in one country may be invasive in other countries (Committee on the Scientific Basis for Predicting the Invasive Potential of Nonindigenous Plants and Plant Pests in the United States, 2002). Thus, the decision makers and resource managers in one country need information from other countries on how invasive species have been effectively managed in other areas.

One of the key desired features for the GISIN is the ability to search across databases for information regardless of where the datum are housed or how it is structured. This user requirement includes searching for the known locations of invasive species, management information for an invasive species, and information on identifying species. The users of the GISIN will include researchers, ecological informatics professionals, managers of natural resources and the general public. These users expect Websites to work quickly and support a variety of features in searching and browsing data (Nielsen 2000, Pearson and Pearson 2008).

The Internet, the database, and the software connecting the two can influence the performance of a distributed database system. The Internet performance is based on overall usage and its reliability can vary. The design of a

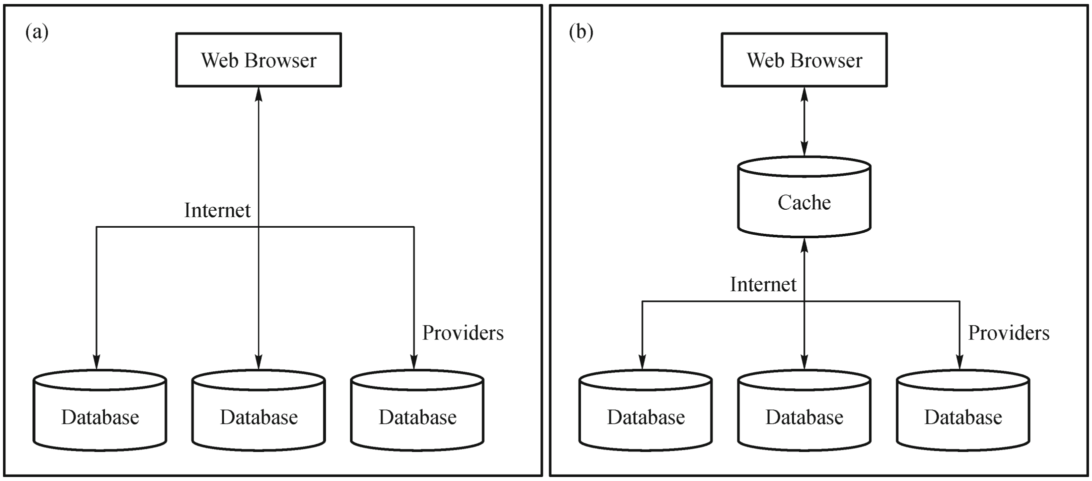

Fig. 1 Diagram of: (a) a system using a federated search over the Internet and; (b) a database to cache the results from providers

database schema and the quantity of data within the database can significantly affect the performance of the system (Baeza-Yates et al., 2008). The software between the database and the Internet can include a database driver, a Web service toolkit, a Web server, and a number of network software components. Two examples of database drivers include Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC). A Web server is software that turns a computer in to a server allowing it to "serve" data (typically Web pages) to browsers on the Internet. Each of these components can affect the performance of the system.

The goal of this study was to compare a federated search and a cached search to determine which would be optimal for GISIN. The basic difference between these two approaches is searching a large number of smaller databases versus searching a larger single database. The GISIN currently has identified 270 databases that are currently online and could form a part of the network (http://www.gisin.org/GISINlist.htm). The experimental design was created to evaluate the tradeoffs between a federated and cached search, in as close to a real-life situation as possible with the resources available.

2 Methods

The GISIN network is in early development stages with a limited number of available databases. At the same time, decisions needed to be made to move forward with the design and implementation of the network. This is not an unusual situation in technical development and require experiments that adapt to the available resources. Three databases were available within the GISIN network at the time of this study. Two additional databases of invasive species field data were available locally to the development team. Using these databases three tests where completed: 1) a federated search on the Internet; 2) a federated search within a local network; and 3) a cached search where the five databases were aggregated into a single database representing a cache.

All tests performed a "Count" operation followed by a "Search" operation where 1000 records were requested. A count is typically executed once for a given search to determine the number of records available for a given search criteria. The total count is then used to determine the number of blocks of a specified size that should be requested. The simplest possible search operation, simply requesting the first 1000 records, was used to help limit the influence of various database designs. The time from sending the operations until the requested information was returned was recorded. Both operations used the GISIN "Occurrence" data model representing the location of a species, its scientific name, and when it was identified. Fields requested were CountryCode, DecimalLatitude, DecimalLongitude, HorizontalDatum, LanguageCode, ScientificName, Kingdom, StartCollectionDate, and EndCollectionDate (see http://www.gisin.org, "Technical" and "Protocol" for details). For each test the minimum, maximum, mean, and standard deviation for the length of time it took to execute the count and search operations were computed. Communication failures where also tallied. Communication failures could include dropped requests (where the operation returns immediately with no results) or a timeout (where the requesting computer gives up on the request).

The first test evaluated the performance and reliability of a federated search over the Internet. This search most closely simulated the characteristics of the future GISIN network and included performance issues introduced by the Internet. To detect these issues this test was executed each hour over the course of a week. Three providers were used for this test: 1) FishBase, a large database housed in the Philippines; 2) Inter-American Biodiversity Information Network's (IABIN) Invasives Information Network (I3N) of Argentina, a relatively small database housed in Costa Rica; and 3) the National Institute of Invasive Species Science database (NIISS), a mid-sized database housed in the United States at Colorado State University where the tests were completed (Table 1). The first two databases provided a test of the reliability of an internationally distributed network while the third provided a control on these effects since it was located in the same location the tests were completed (i.e., it was isolated from the Internet).

The second test provided additional information about the differences between databases independent of problems that may arise when databases are accessed via the Internet. For this test, data were requested from three different Structured Query Language (SQL) Server databases, including the NIISS Website used in the first test, that were all housed locally on the same server. The test was executed 100 times on each of the three databases and we compared average response times to get the full record set. Since these databases were not accessed over the Internet and were on a server that was not performing other operations, the test was executed in one day.

The third test was based on a cached system. Data were requested from all five databases (Table 1) and stored in a single database table. The purpose of this test was to compare the performance of a federated search with a cached search. Because a cache includes data from a large number of databases, additional data were added to this cache to simulate from 500000 to 16 million records. Assuming an average sized database of 500000 this would represent from one to 32 databases stored in the cache. In a federated search, the number of matches to any given query would be expected to increase as the number of available databases increased. To provide a valid comparison with a federated search on the Internet, the number of records requested was increased as the number of records in the cache increased. Thus, 1000 records were requested when there were 500000 records in the database, 2000 records with one million records in the database, and so on. A query for records based on country code was also executed as part of this test to determine the overall performance of the cache based on more challenging queries.

The first and second tests were executed using a new toolkit developed specifically for the GISIN protocol that provides a link between the provider's database and the GISIN protocol. The toolkit was implemented to provide good search performance to the supported databases. Since the focus of the study was on federated versus cached searches, we used real GISIN providers who had implemented the toolkit and two local databases in which we implemented the toolkit. The cache for the third test was placed in a SQL Server database on a server that was not in normal use during the tests to simulate a server dedicated as a cache.

The tests were executed at the Colorado State University's Natural Resource Ecology Laboratory. The university provided a high-performance connection to the Internet and a MS-Windows-based server for the tests.

3 Results

The federated search tests on a database accessed through the Internet showed times to count the number of records available from a minimum of 0.2s to a maximum of 54s (Table 2). The local database, NIISS, had the greatest minimum, maximum and mean times to respond. However, it had zero failures, while FishBase had a 79% failure rate, or 137 records, and I3N had a failure rate of 1.1%, or 2 records. The failed transfers for I3N timed out after 180s. Times to request data from these three databases varied from 0.1s to 63.5s with averages from 0.3s to 24.3s (Table 3). This translates to an average time to acquire 1000 records of 9.2s (Table 3). The same pattern from the counts held true, with the NIISS database taking more time to respond but having zero failures while the others had the same exact failure rates as in the counts. The performance for each database varied in different ways over the course of the week (Fig. 2).

For the second test, the federated search tests on local databases, the time to count the number of records available ranged from a minimum of 0.04s to a maximum of 2.6s (Table 4). The time to acquire 1000 records from

Table 1 GISIN providers and databases used for the tests. The first three were used for the first test and the first database and the last two were used for the second test

Name	URL	Location	Record count	Database software
NIISS	http://www.niiss.org	USA	309879	SQL Server
FishBase	http://www.fishbase.org	Philippines	1245936	MySQL
I3N-Costa Rica	http://invasoras.acebio.org	Costa Rica	3875	Microsoft Access
Local 1	[none]	USA	5628	SQL Server
Local 2	[none]	USA	253239	SQL Server

Table 2 Database response time to return a count of records for the first test in seconds. The mean record count varied for NIISS and FishBase as records were added over the course of the test. The failures for FishBase were all dropped requests with a time of zero seconds, while the failures for I3N were all timeouts near 180s

	Mean record count	Minimum	Maximum	Mean	Standard deviation	Percent failures
NIISS	396948	14.9	54.1	18.4	7.8	0.0%
FishBase	266986	0.2	8.6	1.0	0.9	79.0%
I3N	3829	0.2	180.1	3.0	19.9	1.1%

Table 3 Database response times and failure rates to return the first 1000 records for the first test. Times are in seconds

	Minimum	Maximum	Mean	Standard deviation	Time per record	Percent failures
NIISS	16.5	63.5	24.3	11.1	0.0243	0.0%
FishBase	0.1	0.9	0.3	0.3	0.0003	79.0%
I3N	0.9	180.8	3.3	19.5	0.0033	1.1%

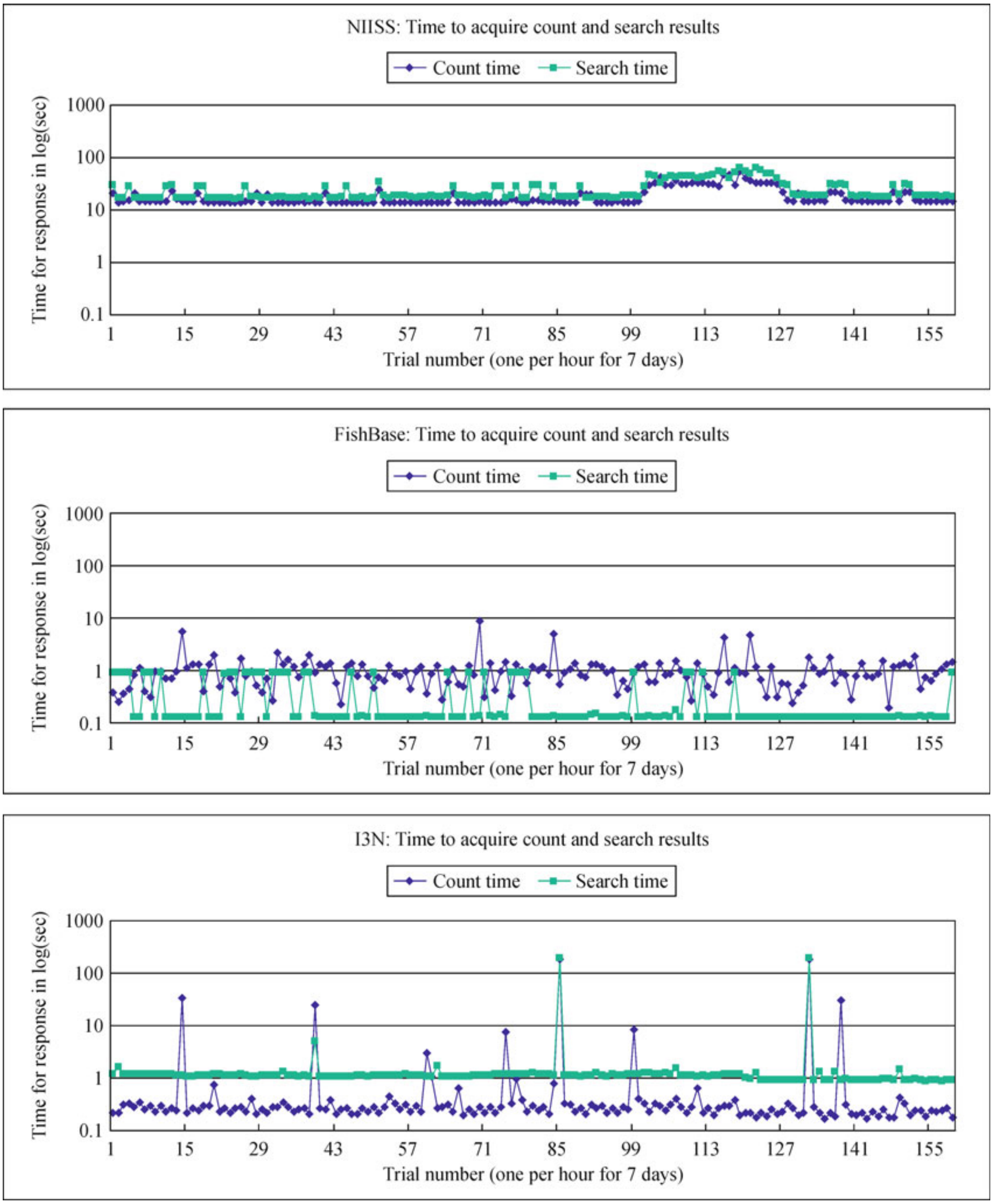

Fig. 2 Time, in seconds, to acquire a count and a search from two remote databases accessed through the Internet (FishBase and I3N) and one local database (NIISS) over the course of a week. The *x*-axis indicates the trial number with one trial executed per hour. To make the other results more visible on the graphs, the requests that timed out where removed from the graphs

local databases ranged from 0.5s to 7.4s for an average time of 1.9s (Table 5). While the mean acquire times where close to federated searches, the maximum times where much smaller (Fig. 3). There were no failures from these databases and all the standard of deviations were less than one second.

The count and search operations performed on a cache showed times to count the number of records available from a minimum of 1.10s to a maximum of 7.03s based on table sizes from 500000 to 16 million (Table 6). Times to request data from these databases varied from 0.45s to 0.48s (Table 7). This translates to an average time to request 1000 records of about 0.45s. A simple query requesting all the data for a particular country was executed on the cache to determine performance under realistic searching. This query showed a sharp increase in the time to acquire 16 million records (44 seconds) (Fig. 4).

Table 4 Times in seconds to acquire a count of the number of records matching a query. Failure rates were 0 in all cases

	Record count	Minimum	Maximum	Mean	Standard deviation
Local 1	5628	0.04	0.1	0.1	0.0089
Local 2	253239	0.15	1.2	0.2	0.10
NIISS	166563	0.04	2.6	2.4	0.24

Table 5 Times in seconds to acquire 1000 records that matched a query. Failure rates were 0 in all cases

	Minimum	Maximum	Mean	Standard deviation	Time per record
Local 1	0.5	0.8	0.6	0.06	0.00060
Local 2	1.3	3.9	1.9	0.44	0.00044
NIISS	0.5	7.4	3.4	0.65	0.00065

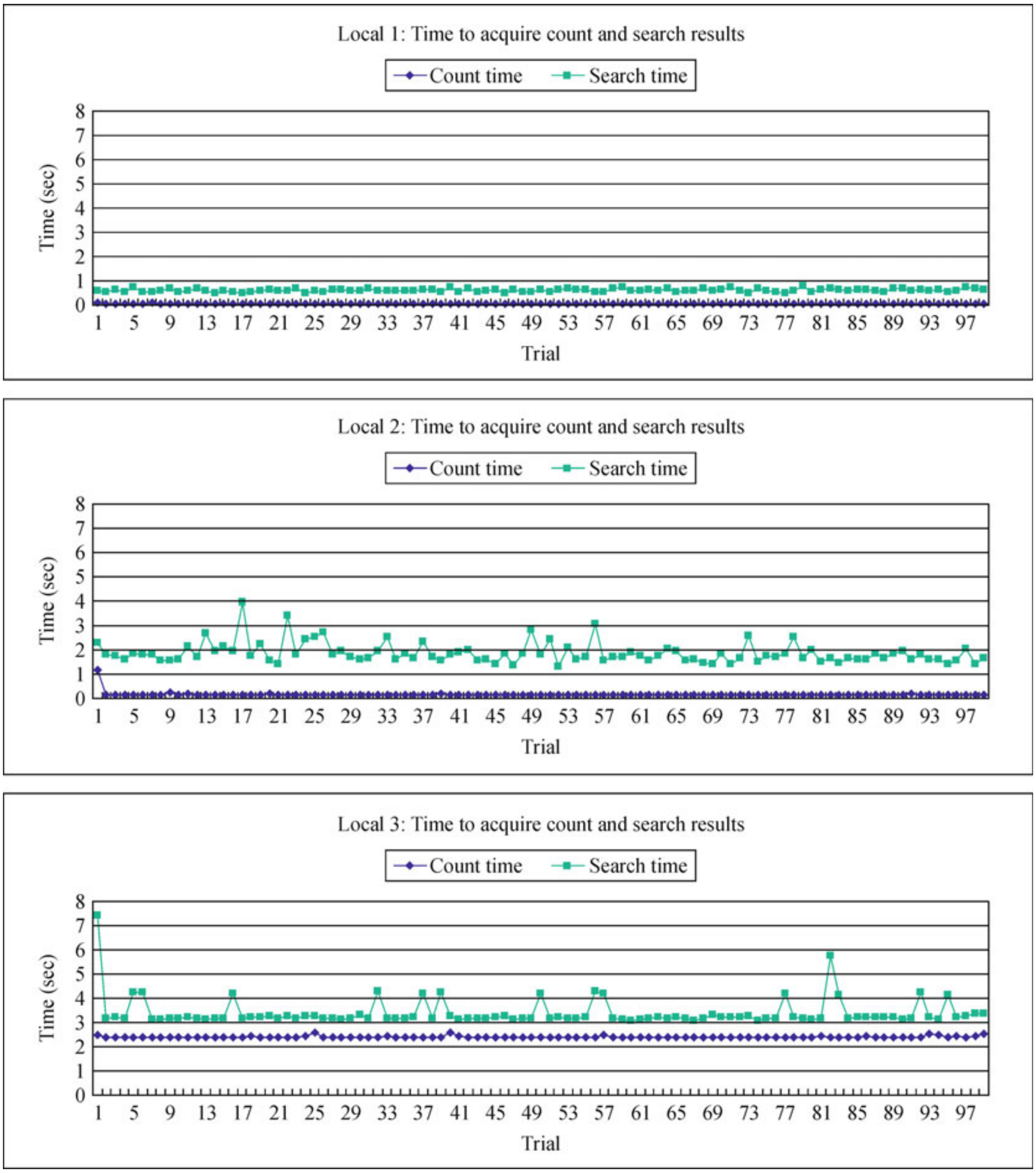

Fig. 3 Times in seconds to count and search local databases through a federated search. The *x*-axis is the trial being performed with 100 trials in all

Table 6 Times in seconds to acquire the number of records in the database cache as the number of records increased. The failure rate was 0 in all cases

Record count	Minimum	Maximum	Mean	Standard Deviation
500000	1.10	1.41	1.14	0.06
1 million	0.81	1.12	0.82	0.49
2 million	1.15	1.91	1.21	0.12
4 million	1.83	1.99	1.83	0.10
8 million	3.19	19.65	3.18	0.29
16 million	2.24	7.03	2.68	0.80

Table 7 Times in seconds to acquire 1000 records that matched a query. Failure rates were 0 in all cases

Records in database	Minimum	Maximum	Mean	Standard deviation	Time per record
500000	0.44	0.76	0.47	0.06	0.00047
1 million	0.45	0.51	0.45	0.01	0.00045
2 million	0.45	0.68	0.47	0.04	0.00047
4 million	0.23	0.45	0.24	0.03	0.00024
8 million	0.23	0.46	0.24	0.03	0.00024
16 million	0.24	0.87	0.54	0.11	0.00054

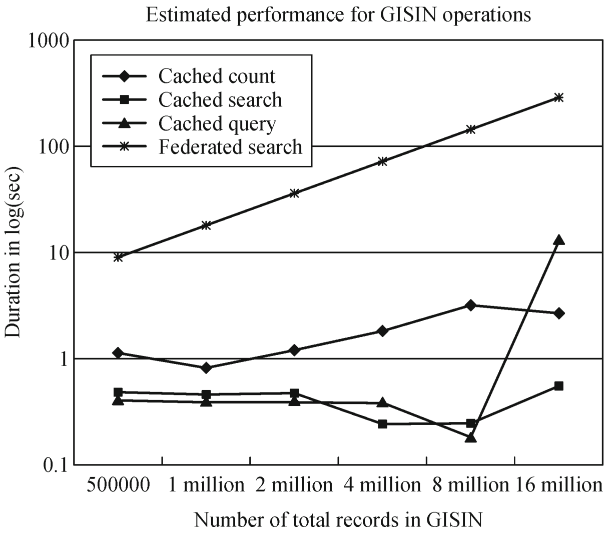

Fig. 4 Times to query the cache and estimated times to search on a federated network based on different numbers of records available in the network. The durations for federated searches are approximated by multiplying the results above times the number of records specified, while the durations for cached searches are directly from the tables

4 Discussion

While the range of provider's performance for individual searches to providers on the Internet almost overlaps, we must consider this with the expectation of a large number of providers in mind to predict the performance of the mature GISIN network. Averaging the mean search times for the three Internet databases provides us with a mean time of nine seconds. If, in the future, there are 200 databases in the GISIN, then the time to search all of them sequentially using this method would be 1800s (30 min). The cached performance is predicted to be about 0.5s. This shows a clear advantage for the cached system based on performance across a large number of providers (Fig. 4). There is the possibility to improve a federated search with such approaches as querying multiple databases simultaneously but this would require additional complexity in managing the requests and integrating the results. Without further research into these approaches these results show the cached count, search, and query operations outperform the federated search in all cases except with small numbers of providers. This indicates that in a large network with end-users expecting fast access to data, a cache should be added to the system.

Surprisingly, the maximum time for a search did not involve the remote database or the largest database, but instead was a local database with an intermediate amount of data. The NIISS database is a relatively complex relational database that was designed to communicate to the Web service through a "View." A view in a database is a query that the database updates on a periodic basis. The other remote databases contained a single table that was mapped directly to the GISIN protocol. The local databases were all connected to the Web service through traditional SQL query statements. The cached searches were executed on the same databases, but on a single table. This result supports the idea of a cache to remove performance differences between databases related to database design and illustrates the need to work with providers to ensure their databases are connected in a manner that allows them to be accessed quickly. In addition, the results show that count operations took far longer than search operations and should not be included in protocols unless there is a clear requirement for them.

Some Internet connection failures were expected, but not at the level found with the FishBase database in the Philippines. The FishBase information technology staff warned us that their Internet connection to the United States was not always reliable. This study showed that the connections between countries could be a serious problem with a federated database search. A cache can help with this problem because at least some version of the datum are readily available, and regular datum provider updates that fail can be programmed to retry until successful. The failures were almost synchronous (i.e., a count failed and then a search failed right afterward), and this may indicate that there are times when the Internet is more reliable than others. An additional problem for effective searches will be users in the Philippines and other countries accessing the cache if it is in the United States. This may require that the caches are replicated and distributed around the globe.

Other unexpected problems with a distributed system may arise that were not encountered here. The data providers' servers will most likely be used for other functions. These functions could include backups, data processing, and serving Websites. For example, the NIISS database was backed up every night at midnight, causing the observed large decrease in the server performance at midnight. Thus, these other functions on remote data providers could also negatively influence the performance of a distributed search.

It is also somewhat unclear how to provide data from cross-database searches in a manner that is compatible with user needs. The simple search operation used in this study is the simplest example, but users may wish to query data in a more complex manner. For example, a user may want an alphabetical list of invasive species from around the world and view the first 100 species. A federated search would have to obtain the top 100 records from each database and alphabetize them to assure we have the top 100. A cached system can perform this query on an internal table of all available invasive species and, because the number of tables searched is reduced and the internal table can be indexed within the database, the performance will be much higher than a federated search. The cached system could more easily provide this kind of flexibility in queries for the end user.

A network of distributed databases and multiple caches may be the optimal solution. Although caching data could improve the performance and flexibility of searches, if there is only one cached system, then there is a single point of failure for the entire system. A system would include all 200 GISIN participants placing their databases on the Web, and with multiple caches of the entire data set to provide performance, reliability, and flexibility to the users. Different caches could also provide different Web interfaces geared toward different types of users (e.g., one that produces multilayered maps for end-users, another that specializes in data downloads for modeling).

There are also issues with a cache system. Cached systems poll their provider's databases on a periodic basis, often monthly. Thus, data from each provider may not be available at the cache until a month after the provider has made it available online. Additionally, some types of data, such as links to species profile information, are not as complex as the occurrence information used in these analyses and may have different results.

5 Conclusions

Our tests show that a cross-database search based on a purely distributed system of any significant size will not provide the end-users may the required levels of performance and that a cached system has other distinct benefits. The cached system may be able to: 1) remove performance issues caused by differences in database design; 2) provide flexibility in querying specific data and in ordering data; 3) enable a wider range of display options for data results; and 4) provide higher reliability for searching than a federated search. However, a system with a cache will require additional support costs to maintain it.

There were some indications that a schedule for updating the cache should be set by the data providers, but additional research is needed to determine the best times and quantities of records to request at a time. Caching data is a computationally intensive process and should probably be executed on servers dedicated to this purpose, leaving other servers free to provide users with access to the data. Resolving these issues will require ongoing communication between the technical group maintaining the cache and the personnel managing the provider databases. Additional information on performance and reliability problems of the system should also be communicated to data providers. Much of this work could be automated with providers notified through email. Most importantly, this research shows that the GISIN network, with the addition of a cache, can provide worldwide access to a huge variety of information on invasive species.

Acknowledgements Funding was partly provided by the US Geological Survey National Biologic Information Infrastructure (NBII) Invasive Species information Node. Logistic support was provided by Colorado State University and the US Geological Survey. Additional support and advice was provided by Dr. Wei GAO and the Integrated Bioclimatic-Dynamic Modeling of Climate Change Impacts on Agricultural & Invasive Plant Distributions in the United States project funded by USDA CSREES-NRI-2008-35615-04666. Dr. Thomas Stohlgren's contribution supported by U.S. Geologicd Invasive Species Program, USGS Fort Collins Science Program, and USDA CSREES–NRI-2008-35615-04666. We thank Michael Browne and Kyle Braak for comments on the manuscript. Any use of trade, product,

or firm names is for descriptive purposes only and does not imply endorsement by the US government.

References

Baeza-Yates R, Gionis A, Junqueira F P, Murdock V, Plachouras V, Silvestri F (2008). Design Trade-Offs for Search Engine Caching. Acm Transactions on the Web (**TWEB**), 2(4)

Casal C M V (2006). Global documentation of fish introductions: the growing crisis and recommendations for action. Biol Invasions, 8(1): 3–11

Committee on the Scientific Basis for Predicting the Invasive Potential of Nonindigenous Plants and Plant Pests in the United States (2002). Predicting Invasions of Nonindigenous Plants and Plant Pests. National Research Council of the National Academy of Sciences, 198

Crall A W, Meyerson L A, Stohlgren T J, Jarnevich C S, Newman G J, Graham J (2006). Show me the numbers: What data currently exist for non-native species in the USA? Front Ecol Environ, 4(8): 414–418

Curcin V, Ghanem M, Guo Y (2005). Web services in the life sciences. Drug Discov Today, 10(12): 865–871

de Giovanni R, Doering M, de la Torre J (2006). TAPIR 1.0. In: Proceedings of TDWG

Edwards J L (2004). Research and societal benefits of the Global Biodiversity Information Facility. Bioscience, 54(6): 486

Fan J C, Kambhampati S (2005). A snapshot of public web services. SIGMOD Rec, 34(1): 24–32

Fox E A, Gonçalves M A, Luo M, Chen Y, Krowne A, Zhang B, McDevitt K, ñones M P Q, Richardson R, Cassel L N(2004). Harvesting: Broadening the field of distributed information retrieval. Distributed Multimedia Information Retrieval, 2924: 1–20

Graham J, Simpson A, Crall A, Jarnevich C, Newman G, Stohlgren T J (2008). Vision of a cyberinfrastructure for nonnative, invasive species management. Bioscience, 58(3): 263–268

Halpin P N, Read A J, Best B D, Hyrenbach K D, Fujioka E, Coyne M S, Crowder L B, Freeman S A, Spoerri C (2006). OBIS-SEAMAP: developing a biogeographic research data commons for the ecological studies of marine mammals, seabirds, and sea turtles. Mar Ecol Prog Ser, 316: 239–246

Jacsó P (2004). Thoughts About Federated Searching. Information Today 21(9) October, 2004, p.17

Mack R N, Simberloff D, Lonsdale W M, Evans H, Clout M, Bazzaz F A (2000). Biotic invasions: Causes, epidemiology, global consequences, and control. Ecol Appl, 10(3): 689–710

Martens A (2005). Analyzing Web service based business processes. Fundamental Approaches to Software Engineering. Proceedings, 3442: 19–33

Nielsen J (2000). Designing Web Usability. New Riders Publishing

Pearson J M, Pearson A M (2008). An exploratory study into determining the relative importance of key criteria in Web usability: A multi-criteria approach. J Comput Inf Syst, 48: 115–127

Pimentel D, Zuniga R, Morrison D (2005). Update on the environmental and economic costs associated with alien-invasive species in the United States. Ecol Econ, 52: 273–288

Vieglais D (2006). The Big Dig. In: Proceedings of TDWG2006

Wilcove D S, Rothstein D, Dubow J, Phillips A, Losos E (1998). Quantifying threats to imperiled species in the United States. Bioscience, 48(8): 607–615