Performance and Cost Tradeoffs in Web Search

Nick Craswell [†] **Francis Crimmins** [†] **David Hawking** [†] **Alistair Moffat** [‡]

[†] CSIRO – ICT Centre,
GPO Box 664, Canberra, ACT 2601, Australia
http://es.cmis.csiro.au/people.shtml

[‡] Department of Computer Science and Software Engineering,
The University of Melbourne, Victoria 3010, Australia
http://www.cs.mu.oz.au/~alistair

Abstract

Web search engines crawl the web to fetch the data that they index. In this paper we re-examine that need, and evaluate the network costs associated with data acquisition, and alternative ways in which a search service might be supported. As a concrete example, we make use of the Research Finder search service provided at http://rf.panopticsearch.com, and information derived from its crawl and query logs. Based upon an analysis of the Research Finder system we introduce a hybrid arrangement, in which queries are evaluated partially by reference to a centrally maintained index representing a subset of the collection, and partially by referring them on to the local search services maintained by the balance of the collection. We also examine various ways in which crawling costs can be reduced.

Keywords: Search engine, metasearch, world-wide web, information retrieval, web crawling.

1 Introduction

There are two standard approaches to the problem of providing a single search service derived from data that is located on a range of different hosts. The first is to periodically crawl the various sites that are to be included, and build a comprehensive central index to be used for subsequent querying. This is the arrangement used by generic web search engines such as www.google.com and is by far the most common approach. It allows fast query processing time based on an index local to the search server. However, crawling to download every single page is expensive, and if it is not done often enough the crawl will no longer match the information available on the original hosts – the index becomes stale.

The second approach is that of metasearch, and is possible only if the contributing hosts offer localized search services that index their data. In a metasearcher, the query is forwarded from the central server to each of the component sites, and their returned answers are merged and presented as if they had been identified centrally. The advantages of this model are that the computational (and implementation) effort required at the central site is reduced, and that answers are more likely to be fresh. The disadvantages are that query processing is slowed due to extra network communication, and that an effective method is needed to merge the results of disparate services and possibly remove duplicates. Effective merging may be particularly difficult if the metasearcher has incomplete information about the documents being merged.

Metacrawler (http://www.metacrawler.com) is the best known example of a metasearcher, though the sets of documents indexed by the primary search engines it harnesses (such as Google and Fast) completely overlap, at least in principle [Selberg and Etzioni, 1995]. An experimental metasearcher indexing 15 non-overlapping current news services (http://peace.anu.edu.au/Yves/Metasearch) is described by Rasolofo et al. [2003].

Another approach, which so far has been rarely used in practice, can be thought of as selective metasearch. A selective metasearcher sends its queries to a subset of the satellite servers, to reduce the effort required at query time. It uses a collection selection mechanism to direct queries to the sites most likely to be able to answer them. In a sense, selective metasearch is a compromise between the first two approaches, since the central site must hold some information about the servers in order to perform selection, but retrieves its results via metasearch. A range of authors have examined such systems, including mechanisms for identifying relevant subcollections and the extent to which the volume and type of central information affects retrieval effectiveness [Hawking and Thistlewaite, 1999, D'Souza et al., 2003].

In this paper we examine the network cost of these and other alternatives. We also consider the freshness of crawls in various approaches, based on change data from real web pages.

In particular, we examine a new alternative – a system in which some of the satellite sites are crawled, and their data retained centrally, while the others are accessed via metasearch. If the largest contributing sites are metasearched, then the crawl volume can be significantly reduced. Also, high-turnover sites can be handled via metasearch, to reduce the need for the frequent crawling required to keep the index fresh. This hybrid model translates into reduced network costs and increased freshness of search results. However, it also faces the metasearch drawbacks of extra network communication at query time, and the need for results merging based on incomplete information.

Our initial analysis is based upon an operational crawled system called Research Finder (http://rf.panopticsearch.com). Section 2 describes the current form of that service, and summarizes the usage of it. Section 3 then introduces a cost model that allows us to compare alternative retrieval structures, and measure querying costs in dollars. Section 4 introduces several alternative

Number of organizations	175
Number of hosts	3,405
Number of distinct pages indexed	2.383 million
Total volume of data indexed	33.3 GB

Table 1: Research Finder as at 30 June 2003.

approaches including a hybrid model which provides superior cost-effectiveness to both the central index and full metasearch alternatives. Finally, in Section 5, we generalize the cost model to other applications.

2 An operational system

This section describes the operational requirements of the web search engine used as our principal case study.

2.1 Research Finder

Research Finder is a searchable full-text index-based retrieval system built from a regular crawl of the public websites operated by a range of Australian research institutions, including universities, government agencies, cooperative research centers, and technology transfer organizations. It was commissioned in mid-2000 by the then commonwealth government Department of Industry Science and Resources (DISR) as a service to Australian industry. DISR and its successors maintain a list of sites to be included. They have also classified these sites by state, research field and institution type. This allows users to narrow their search via this "external metadata" (metadata which applies to pages but is not embedded in pages). Table 1 provides some statistics about the Research Finder service.

Many of the research organizations included in Research Finder operate their own localized search service, although Research Finder does not make use of them. In our experience larger institutions such as universities are most likely to support a search service. A variety of search engine products are in use, for example Panoptic (`search.csiro.au`), UltraSeek (`search.unimelb.edu.au`), ht://dig (`www.wehi.edu.au/search/`), and site-restricted Google (`www.find.curtin.edu.au/searchcurtin.html`). Large institutions typically update their local search indexes at least once each week. At the other end of the spectrum, some of the smaller sites (for example the Asia Pacific Research Institute, Austin Health, Australian Technology Park, Federation of Australian Scientific and Technical Societies) provide no local search service at all.

Organizations contribute widely varying amounts of data to the Research Finder crawl. Figure 1 illustrates this, by showing the cumulative growth in crawled data size when the sites are ordered from largest to smallest. The eight largest Australian Universities occupy eight of the first nine places, and account for almost 50% of the indexed data by volume. The smallest 50 sites are typically Cooperative Research Centers and technology transfer organizations, making up just 0.40 MB (0.001%) of the data indexed. Organizations range from being very costly to crawl and index, to very inexpensive. It is this continuum that we seek to exploit in our hybrid system.

2.2 Research Finder queries

The Research Finder query load is relatively low, typically only a few thousand queries per month. We suspect that some users who know about the service find it more convenient to use a whole-of-web search engine, and only use Research Finder for certain queries – for example, searching by geographical region within Australia, or by research field.

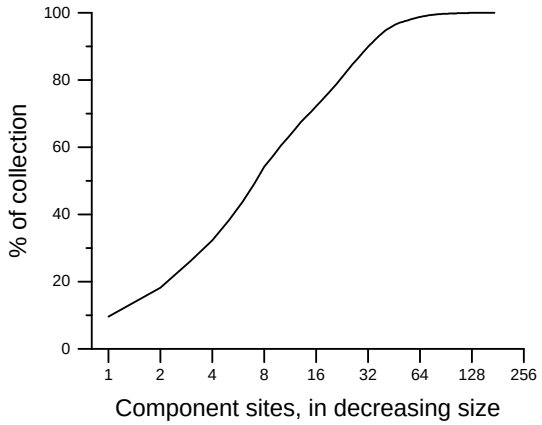

Figure 1: Cumulative collection size as a function of rank, where organizations are ordered by decreasing size. The first nine organizations include the eight largest Australian Universities, and between them account for more than half of the total volume of data indexed.

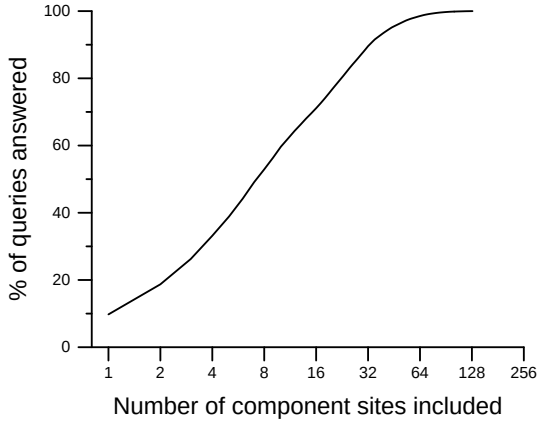

Figure 2: Origins of query answers, ordered by decreasing usefulness of the corresponding server, for 3,658 Research Finder queries. Approximately half of the suggested answers are located in the eight most useful component collections.

To characterize the query load and the distribution of query answers across the indexed domains, a contiguous sample of 3,658 recent queries was extracted from the Research Finder logs and used as the basis for a retrospective analysis. Of the 3,658 queries, 47.8% made use of Research Finder's ability to restrict results to a particular geographic area, research discipline or institution type. Counting each such restriction as a query word, the average query length in the sample was 2.88 words, somewhat longer than the 2.35 words reported for whole-of-web search engines by Silverstein et al. [1999]. An illustrative sample of queries is given in Table 2.

Each of the 3,658 queries was then re-run against the Research Finder index to determine the distribution of query results by organization. A maximum of twenty results per query were requested, and an average of 9.77 suggested answers per query were identified by Research Finder, 35,749 URLs in total. Figure 2 plots the origins of those URLs, ordered by decreasing likelihood of a collec-

Query	Research Field	Institution Type	Region
bisphosphonate veterinary	AgVetEnv		
AQTF			
			NSW
MRET media release			
safeguarding future			
disabilities	MedHealth		Tas
ELISA	BiolSci		
competitive ELISA	BiolSci		
competitive ELISA	BiolSci	MedRes	Qld
competitive ELISA	BiolSci		Qld
ELISA	MedHealth		Qld
ELISA	MedHealth		Vic
disabilities	HumArts		Tas
EAMENS DHORDEVIC		Government	NSW
lugton			
appropriation			
	ICT		NSW
"op op" rat			
physicotherapy	MedHealth	Universities	SA
campril	MedHealth		
"op op" mouse			
internet pornography law	ICT		
toothless rat			
medicolegal documentation	MedHealth	MedRes	
fauna survey proposal	AgVetEnv		
skull bone graphic			
skull bone graphic	MedHealth	MedRes	
soil			
homoeostasis			
anatomy skull bone graphic	MedHealth	MedRes	
dryland salinity		SA	
impacts dryland salinity		SA	

Table 2: Examples of queries submitted to Research Finder. The set of sample queries was an arbitrarily-chosen contiguous sequence from July 2003, with repetitions removed.

tion supplying an answer.

The resultant curve shows a similar skew to Figure 1, and the eight largest universities again all appear in the top ten servers. Those top ten answer providers account for 59.7% of all suggested answers, and the top 50 providers cover 96.6% of the suggested answers. There is also a significant correlation between proportion of data crawled and proportion of answers contributed (Pearson $r = 0.864$, with $p < 0.05$). However, as stipulated by DISR, the service must cover even the very small organizations, despite the fact that they rarely provide answers.

2.3 Rate of change

In a crawled service, pages may change or disappear any time after the crawler downloads them. The freshness of a crawl is usually defined as the proportion of the cached pages that still match their on-the-web sources [Cho and Ntoulas, 2002]. To analyze the average freshness of the Research Finder index, we decompose non-matching pages into three classes:

(a) pages which have disappeared;

(b) pages which have changed so much that they are likely to be bad search results; and

(c) pages where the content has changed but they are still good answers.

Changes (a) and (b) are most important in the case of a search system, and as a crawl becomes stale, the users of the service are more likely to see an embarrassing result of either type (a) or (b).

A different type of failure is the absence from the index of new pages. This is most often noticed by the publisher

	Number	Percent
URLs in initial subset	28,389	100.0%
URLs not valid on day 44	1,273	4.5%
Pages which changed slightly	1,940	6.8%
Pages which changed more	4,558	16.1%
Unchanged pages	20,618	72.6%
Pages which had downtime	5,398	19.0%

Table 3: Responses returned on day 44 of our observations. A "slight change" was a change in the MD5 checksum for the page, but where the page length changed by less than five bytes. This often corresponds to automatically generated pages containing the current time or date. Other changes in checksum were recorded as being more significant alterations. The downtime count of $5{,}398$ was the number of pages which disappeared at least once during the observation period, but were valid again on day 44.

of the page, but can also affect a user's impression of a search service if they know that the page exists, but fail to locate it via the search service.

To explore the rate of change of Research Finder data, we selected slightly more than 1% of the URL set comprising the July 2003 crawl. The pages indicated by those URLs were fetched between 2am and 3am every day for 44 consecutive days. (There was a single exception, when a technical problem caused the overnight process to fail on one day. To allow the experiment to continue, we duplicated the data from the previous day.) Table 3 summarizes the changes over the 44 days.

Each time a page was fetched, it was compared against versions from previous fetches. This gave 43 observations of "one day change rate", 42 observations of "two day change rate", and so on, up to 12 observations with a 32 day window. Across each available window, we measured whether the page disappeared or changed. The set of pages which had changed was divided into small changes, where page length changed by less than five bytes, and large changes. This allowed us to assess the level of minor changes, such as a dynamic page which contains the current date or time, without having to do a full evaluation of text difference. Small change pages are likely to correspond to change class (c) described previously.

Figure 3 shows the likelihood of alteration for pages over the various windows. For example, over the study period the change rate over an eight day period was on average 1.6% for disappearance, 6.4% for large changes and 8.2% for small changes. In the remaining 83.8% of cases, the page was the same on the first and last day of the period. (We ignored periods where the page in question was down at the beginning.)

A larger study in 1999 of page change rates [Cho and Garcia-Molina, 2000] involved downloading $720{,}000$ pages from 270 sites every day over a period of four months. It did not differentiate between large and small changes in pages, but based on an "average interval of change" measure, pages in the `.edu` domain had a monthly change rate of 26%, and a daily change rate of 2%. Research Finder pages, which are largely from the `.edu.au` domain, had corresponding figures of 24% and 10% (Figure 3). One possible reason for the increase in the daily change rate is that in 1999 fewer pages were dynamically generated.

Fetterly et al. [2003] also analyse the evolution of web pages. In their study a set of 151 million pages was

Symbol	Units	Interpretation	Value
S_d	GB	Combined data size of all subcollections	33.3
S_o		Crawling overhead (ratio of bytes fetched to bytes indexed)	1.7
C_t	$/GB	Transfer costs associated with Internet usage	22.5
F_c	crawls/month	Crawl frequency	1
F_q	queries/month	Query arrival rate	10,000
S_q	GB/query	Volume of data associated with a query response page	2×10^{-5}
N_c		Number of subcollections/servers being federated	175

Table 4: Assumed parameters of a multi-host information retrieval system. The numbers in the column headed "Value" represent the current parameters of the Research Finder system. The network traffic cost reflects the 2003 AARNet rate for traffic outside the RNO (Regional Network Organization) and applies to incoming packets only.

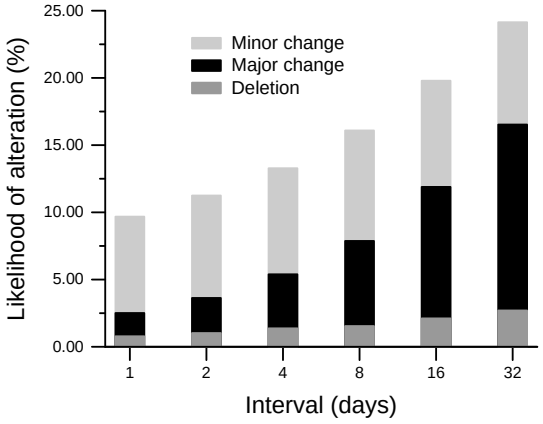

Figure 3: The probability that a randomly selected page in the Research Finder collection is deleted or altered within the specified interval.

	Method			
	Crawled	FullMerge	Selective	Hybrid
Crawling cost	$1273	$0	$43	$584
Querying cost	$5	$792	$85	$41
Total cost	$1278	$792	$128	$625
Cents per query	12.8	7.9	1.3	6.3

Table 5: Network cost per month of central index, metasearch, and two hybrid alternatives for Research Finder. In "FullMerge", there is no central index and all organizations are accessed for each query. In "Selective", a one-fifth size index is updated once every six months, and used for each query to select eighteen (approximately 10%) of the organizations to be accessed to determine answers. In "Hybrid", the eight largest servers are metasearched and the rest centrally indexed after a monthly crawl. The costs are based upon the values in the final column of Table 4. All amounts are in Australian dollars, except for the last row.

crawled once a week for 11 weeks in late 2002. Their weekly change rate in the `.edu` domain was 10%, compared to our eight day rate of 16%. Fetterly et al. did not count situations where the markup was altered, but the content unchanged, a possible reason for the difference in edit rate. Both of these previous studies found that pages in the `.com` domain change more frequently than those in the `.edu` domain.

3 Cost models

Table 4 shows the assumed values for a range of parameters that reflect the current usage of the Research Finder system.

Using the parameters established in Table 4, the monthly cost of crawling is given by

$$F_c \times S_d \times S_o \times C_t \,,$$

which, for the Research Finder collection, amounts to approximately $\$1{,}273$ (56.6 GB of traffic). In a central-index system, this represents a cost that must be paid before the first query can even be processed. The factor S_o incorporates the overheads of crawling which arise because it is impossible to know what is at a URL until it is requested. Requesting dead URLs or redirect URLs leads to some overhead. Also, sometimes a URL is requested that seems likely to return indexable data but actually returns unindexable binary data, in which case our crawler stops the transfer, but some traffic is still generated. By far the largest effect comes from downloading duplicate or near duplicate documents. Duplicate elimination post-crawl often eliminates one third of the data retrieved. The Research Finder figure of 33.3 GB (Table 4) does not include such traffic overheads, which is why we require the multiplier S_o.

The variable cost associated with answering queries must also be factored in. Allowing an outward network traffic of

$$F_q \times S_q \times (N_c + 1) \times C_t$$

gives a cost (in the crawled Research Finder system where N_c is zero) of less than $\$5$ per month. Query processing costs are likely to become an issue when the number of queries per month (F_q) is high or in a metasearch system federating many servers (N_c). (Note that in this calculation we assume that outgoing packets must also be paid for. This is not currently true for Research Finder, but may be true in some environments.)

The other costs are less interesting with respect to our study, and rather harder to quantify. They include the cost of the hardware needed to run the service, the cost of the software used to support it, and the cost of the personnel that manage the service. In general these costs are high initially, and then involve some lesser amount as a maintenance levy. We do not attempt to estimate the hardware, software and personnel costs for each of the models described below, but we do note which alternatives are likely to require significant personnel cost. For example, a metasearch or selective metasearch solution for Research Finder would require "wrappers" for querying the 175 organizations. Keeping these up to date would be a non-trivial ongoing cost.

The second column of Table 5 summarizes the costs associated with running the Research Finder service in its current form. Figure 4 extends the numeric results in Ta-

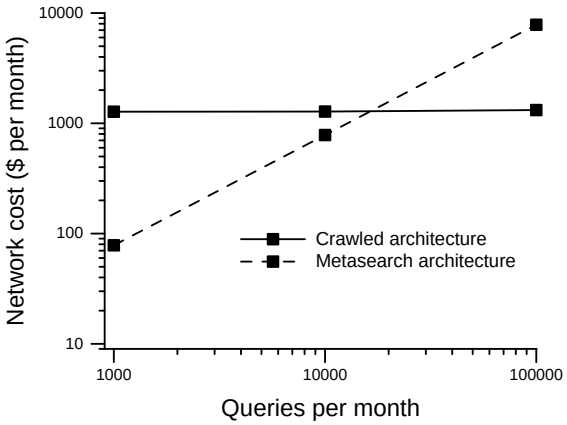

Figure 4: Monthly network cost as a function of the number of queries per month for Research Finder, using the collection statistics listed in Table 4. When the number of queries per month is low, metasearch implementations (the "FullMerge" method is shown) are cost effective, at least in terms of network traffic.

ble 5, and uses the same cost model to show that when the monthly query load is small, a metasearch-based architecture could be attractive.

Note that, while the values in Table 5 and Figure 4 have been presented in terms of crawls per month and queries per month, in many situations it is helpful to consider the load in terms of queries per crawl.

Finally in this section, it is worth commenting that while crawls gradually become stale and result in poor query effectiveness, comprehensive crawling does have the advantage that the age of each part of the index is known to the query engine, and a "probably best before" date can be assigned to each page. In a metasearch system the freshness of the indexes being employed cannot be certified, and poor results might be returned from servers that for technical or financial reasons have stopped building their internal indexes.

4 Alternative approaches

In this section, we discuss alternatives to the current Research Finder model and outline the likely implications both for cost and quality of service. We consider a range of metasearch models; a hybrid of metasearch and crawling; a range of centralized index alternatives in which the network cost of fetching data is reduced; and "piggybacking" off existing global indexes.

4.1 Pure metasearch

Creating a full metasearch or selective metasearch alternative to Research Finder presents a number of significant challenges. Nevertheless, we consider these pure metasearch models first, before moving on to other more practical models.

The first hurdle is that, for a pure metasearch approach to achieve the same level of coverage as the present Research Finder service, all the 175 included organizations must operate local search services. Currently this is not true, although we could "fake" a service for the ones that do not, by building and updating weekly a crawled index of those institutions.

Second, it would be necessary to find and write wrappers for querying all 175 services, potentially increasing personnel costs of running the service. Some software support is becoming available for this [Cope et al., 2003, Wu et al., 2003]. However, automation of such processes on the web is a current research area, rather than a common feature of search products.

Third, it is necessary to have effective methods for merging search results (and for server selection in the case of selective metasearch). In the case of full metasearch the merged results must be synthesized from 175 results lists. Doing so efficiently (without downloading the documents for reranking) and maintaining early precision in the combined list is difficult. Merging remains an area of active research.

The final drawback of full metasearch is the cost at query time of metasearching. For full metasearch, the cost is high because $N_c = 175$. Sending each user query to 175 search services, which return results pages averaging 20 kB each, would lead to 3.4 megabytes of traffic per query, and $\$792$ of traffic per month. Worse, this traffic happens after the user enters a query and before they receive their answer, slowing query response times significantly. For this reason alone, full merging will be frustratingly slower than other options, even if severe timeouts are set to eliminate slow server responses. Gray [2003] examines this observation in detail, and suggests that in the absence of other factors, data should be stored as close as possible to the host on which it is to be used.

In addition, the full query load F_q is applied to every server, regardless of the chances that it will return any matches, a load transfer that is unlikely to win us any popularity awards.

4.2 Selective metasearch

In the selective metasearch case, the list of 175 servers is reduced to a smaller number, without sacrificing (it is hoped) the quality of the results. Merging remains an issue. Selection might be made easier by using an old crawl, which could indicate which institutions used to have matching documents and therefore are candidates to be metasearched for the current query.

The cost of querying in a selective metasearch system is less than in a full metasearch architecture, leading to less network traffic and query delay. In the specific case of Research Finder, the dropdown lists (see the examples in Table 2) for organization type and location could be used to augment any query-based selection process. For example, if the user chooses to query "Cooperative Research Centers" or "NSW", the possible N_c is already significantly reduced from 175.

The retrieval effectiveness of selective metasearch is determined by the effectiveness of the server selection heuristic. Figure 2 shows that the most simple-minded approach – a fixed selection of servers – would be quite effective. Always selecting the 18 servers which contribute the most answers covers nearly 75% of the results returned by the current Research Finder at less than 10% of the network cost. This "bluff" strategy would, however, have the unfortunate consequence of completely hiding all Australian research outside a handful of large institutions, and cannot be accepted.

The compromise we suggest here, and in the amounts listed in the "Selective" column of Table 5, is for a relatively shallow crawl to be undertaken, at relatively infrequent intervals. The query would be run first against the reduced-scale Research Finder index, and then against local search services at each of the institutions contributing highly ranked results. We conjecture that the reference

crawls would need to be carried out only once or twice a year, and that fetching as few as 20% of the documents at each site would give a fair overview of server content.

We have not attempted to evaluate the effectiveness of this approach compared to central indexing or to a full metasearch, and erosion of retrieval effectiveness is a real risk. We hypothesize that selective metasearch is likely to be viable in environments where there are many answers, and where high precision is more important than high recall. For casual web searching, both of these conditions tend to be true. However it is not clear that the same assumptions should be made about portals such as Research Finder, and the low network cost (Table 5) of providing such a service might correspond to low user satisfaction with the returned answers.

4.3 A crawl-metasearch hybrid

A full index is the most efficient way of supporting the query load when there are many queries to be handled. But when the query arrival rate is lower, it becomes economically advantageous to metasearch some sites.

In particular, this advantage depends on an organization's crawl volume compared to the query volume. Assuming $F_q = 10{,}000$ queries per month, the cost of metasearching a single organization is 0.19 GB of traffic ($\$4.29$) per month. The cost of crawling the smallest organization is less than five cents per month, so it is clearly cheaper to crawl than metasearch. On the other hand, we index 3.9 GB from UNSW, which suggests a crawl cost (with S_o overhead) of 6.63 GB or $\$149.18$, and it is clearly cheaper to metasearch UNSW than crawl it.

This observation, combined with the knowledge that higher N_c values leads to slower query processing and more maintenance of metasearch wrappers, suggests a hybrid model with the largest organizations metasearched, and all others crawled. To answer a query, the results from the local index are merged with those from a modest number of the largest servers.

We set up a proof of concept demonstration at `http://thylacine.panopticsearch.com/hybriddemo/index.cgi`. It metasearches the eight largest universities, combined with a crawl of the 167 remaining organizations. This gives a full coverage of the 175 organizations, while reducing the crawl cost by approximately half (Figure 1).

Because it involves metasearch, the cost of the hybrid model depends on query volume. For example, when the query load is $10{,}000$ queries per month, a partial metasearch that combines an index for approximately half the volume of data with a metasearch of the eight largest subcollections costs 6.3 cents per query for network traffic, compared to the 12.8 cents per query for a fully crawled approach (Table 5).

Several drawbacks to metasearch, such as the need to write wrappers, response time issues, and the need to rely on quality local search services also apply to hybrid methods. However, because they are large organizations, it is more likely that the eight universities have the staffing levels and software budget to run a search service of sufficient quality and robustness that the transferred query load will have only a small amount of impact. Answer freshness is also likely to improve – within an institution, crawling is typically performed once a week. On the other hand, freshness also becomes less predictable, and a minority of institutions might crawl infrequently, or not at all.

Figure 5 shows how the monthly network cost varies as a function of the query load, and of the number of servers that are metasearched rather than crawled. Each curve reflects an assumed query load, and shows the cost (dollars per month) of supporting that query load with a varying number of the servers searched for each query. At the point labelled "1", a full central index is prepared and all subcollections are crawled; at the point labelled "2", the largest subcollection is removed from the crawl and is metasearched instead; and so on. The two cost formulations described above were blended to get the overall cost. As can be seen from the graph, even for high query loads, there is an advantage (in terms of network traffic costs) of metasearching some of the large servers.

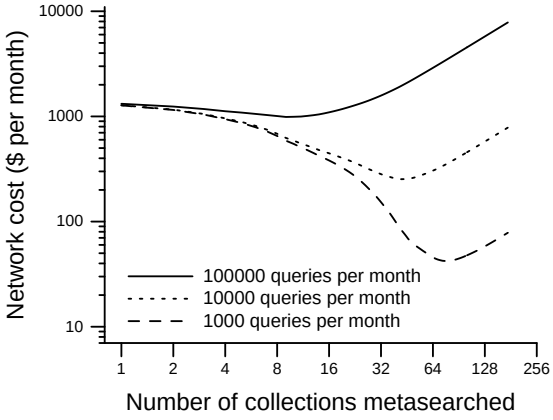

Figure 5: Network cost per month when using a hybrid crawled/metasearch system, for three different monthly query rates. A hybrid system can be cheaper than either a crawled system or a FullMerge metasearch system.

4.4 Reducing crawling costs

The cost of operating a crawled central-index model is heavily dependent on the frequency of crawling and on the amount of traffic generated by each crawl. Costs could be dramatically reduced by crawling more intelligently.

An important first step is to eliminate duplicate pages from the crawl. The Research Finder crawler looks for mirror sites using a checksum-based similarity measure, applied to the root page of each HTTP site. If a site is found to match an existing site, it is not crawled. Even with this in place, a recent test found that 31% of the remaining pages are duplicates (ignoring markup and whitespace). Implementing more sophisticated duplicate detection would not reduce the crawl by 31%, because a page must be downloaded before it can be checked. However, making some effort to eliminate the root page of a duplicate hierarchy significantly reduces crawl costs.

A related mechanism is to eliminate pages unlikely to be useful answers to queries. DISR carefully listed only institutions of interest to those seeking Australian research expertise, but some of these contain a large amount of non-research material. For example, AARNet is the second largest institution at 3.5 GB, because of its extensive software mirrors. Much of this content could be eliminated from the crawl, if it were deemed to be irrelevant to those seeking Australian research.

As noted in Section 2.3 a large majority of pages indexed by Research Finder do not change in the course of any given month. Many of those pages which did change, did so frequently or in very small ways. If it were possible to check whether a page had significantly changed, using substantially less network traffic than actually fetching the

page (10% might be a reasonable estimate), then Incremental Crawling [Cho and Garcia-Molina, 2000] would allow a dramatic saving in cost.

Based on the data presented in Figure 3, if a perfect incremental crawl were performed each month, rather than a full one, the amount of data traffic generated would be 13.9% for fetching the "Large Change" pages, plus 7.8% for the "Small Change" pages. In other words, network costs would be reduced to less than a quarter. In addition, the overhead of S_o would be reduced or eliminated for incremental crawls, because URLs requested were not duplicates, binaries, redirects or dead links in the previous crawl, so are much less likely to be so in this crawl. Applying both of these, we estimate that the monthly crawl cost would drop from $1273 to less than $200.

Incremental crawlers come in two flavors. Adaptive crawlers use a page's past rate of change to predict its future change, and freshen those pages likely to change by refetching them [Edwards et al., 2001]. An interesting twist would be to preferentially update pages based on likelihood of changing and likelihood of being returned as a search result.

Less common is a header based incremental crawler. This would use the HTTP headers provided by the server to check cache consistency [Fielding et al., 1999], and only pages that had changed would be revisited. Note, however, that this approach relies on the fidelity of the HTTP header, a factor beyond our control.

Although we have not implemented a header based crawler, we made HEAD requests for a 11,000 page sample of Research Finder to examine the available information. The most useful header seemed to be Content-Length, which was available and accurate in 46% of responses. Since a HEAD request only generates 300 bytes of traffic, it is an inexpensive way of finding out which pages (in the 46%) have changed significantly in length and therefore should be downloaded again.

Crawl costs might also be reduced by an intelligent assessment of content. For example, it seems unlikely that DISR intended that all of the catalog pages of the University of Melbourne library, and of the Student Information Service (lecture and exam timetables, and so on) should be part of Research Finder.

4.5 Alternatives to crawling

Crawling requires minimal cooperation from the organizations being indexed. In general, unless the `robots.txt` protocol is used to exclude crawlers, any web site which can be browsed by a human can also be crawled.

A rather higher level of cooperation is required if incremental crawling is to work effectively. Web servers must supply accurate last-modified dates and preferably checksums in response to `HEAD` requests from the crawler, as otherwise the page must be fetched in order to determine whether it has changed.

Even greater network efficiency is achievable if a high level of cooperation and trust exists between the indexing and indexed organizations. In such cases, the organization whose data is being indexed might agree to run a software slave on its webservers to supply data in bulk to the indexing organization. The simplest form of this cooperation is a "pull" model, in which the slave accepts a monthly request from the indexing organization, and supplies all webpages regardless of whether they have changed. Even with this approach, the network traffic should drop to around one tenth the previous level, as a result of pre-elimination of duplicate content, and the high compressibility of web pages. As well as frequently repeating simple HTML tags, modern web sites typically repeat large amounts of navigational structure, style sheets and JavaScript code in every page.

The pull model slave can be further refined to allow pre-extraction of text from binary files such as PostScript and PDF documents. Pre-indexing could also be contemplated but requires agreement on indexing structures (unlikely in a heterogeneous environment) and is not likely to reduce data traffic very much.

"Push" slaves are also possible. These routinely monitor the target website, and send updates back to the indexing service at agreed intervals. Such updates could be sent in the form of compressed differences, to maximize network traffic savings. The biggest advantage of the push model is the ability to maintain a very fresh master index. Large scale savings in network traffic are achievable, particularly if overly-frequent updates of trivially changing pages are suppressed.

Unfortunately, trust is a rare commodity in these days of viruses and spam. The indexing organization would need to trust the slaves not to return undesirable content, and the organization being indexed would need to trust the slave software it hosted not to collect private information or to attack systems.

4.6 Crawl piggybacking

A large amount of traffic to a typical web site is generated by web crawlers. Sites are crawled by the local search service, several global search services, and quite possibly by a range of portals such as Research Finder. If the number of active crawlers could be significantly reduced, website operators could potentially save money by deferring hardware upgrades of their servers, and deferring upgrades to the bandwidth of their Internet link.

One could imagine a scenario (a "Global Data Vendor" model) in which a single company provided a global web crawling service, and sold partial data or indexes to search service operators like AllTheWeb and Research Finder. From the point of view of a service such as Research Finder, this model offers network traffic savings comparable to those described in the immediately preceding section.

Another way of piggybacking off a global crawler is by forwarding queries to a global search service and requesting that results be restricted to domains of interest (a "Global Query Forward" model). A query forwarding system incurs almost zero operating costs, but would require Research Finder to enter into a business arrangement with the global search provider. Such an arrangement is likely to require either the payment of a service fee, or the willingness to place advertisements for the global search provider or their clients.

Costs of piggybacking arrangements are difficult to quantify because they depend upon market forces rather than technical factors.

One potential issue for portals like Research Finder is how accurately it is possible to specify the domains to be indexed in a piggybacking model. The definition of what is indexed by Research Finder is currently a list of hundreds of domains (and sub-directories of domains). Such a complex restriction list is not supported by current global search services.

Research Finder could be approximated by a shorter list such as `.edu.au`, or `.edu.au + .gov.au`, but result coverage would be harmed both by the omission of significant content, and by the inclusion of extraneous material.

5 Applying the cost model

Different types of multi-server web search systems sit at different locations in the space of tradeoffs canvassed in this paper. To complete our discussion, we now consider several different search scenarios, including a large centralized organization, a large distributed organization, a topic specific portal, and a whole-of-web search service such as Google. First, we look again at the implementation options for Research Finder.

5.1 Research Finder

Regrettably, many of the canvassed avenues for reducing network cost are infeasible in the present circumstances. The on-paper gains for metasearch and selective metasearch are unachievable, because of the cost of developing and maintaining wrappers for a large number of search interfaces. Furthermore, many organizations do not provide local search.

Improving the intelligence of crawling through incremental techniques and through heuristic-driven variable crawling regimes seem to offer the most feasible cost-reduction mechanisms. These improvements would not harm coverage, freshness or quality, provided that web servers provide the necessary header information. More effective exclusion of low value content such as duplicate pages and non-research content would also cut costs.

A hybrid approach in which a small number of organizations are metasearched might also be worth investigating. The best candidates for metasearch would be those with large data holdings and with effective, unchanging, high quality local search interfaces.

User evaluation via relevance assessments would be needed to determine whether the result merging problem introduced by the hybrid model could be addressed in such a way that ranking quality could be assured.

5.2 A large centralized organization

Within a predominantly single-campus university such as the University of Sydney, networks are typically of very high capacity and internal traffic is essentially free.

Even if internal network traffic were charged, a crawled solution would be the cheapest alternative, as query volumes on the external search service at Sydney University (`http://search.usyd.edu.au`) are much higher than those seen by Research Finder (of the order of $100{,}000$ per month, or $25{,}000$ per crawl) and the amount of data indexed is much smaller (approximately 7.5 GB). Metasearch would seem impractical as there are several hundred individual web servers in the University, many of which provide search only via the central index.

Incremental crawl techniques (or software slave models) would pay off not in reduced cost but in improved freshness or coverage. Global query forwarding might also be a plausible option.

5.3 A large distributed organization

Within a geographically distributed organization such as CSIRO, some network traffic is free and some is subject to network traffic charges.

CSIRO's external search service (`http://search.csiro.au`) has similar parameters to the University of Sydney: around $50{,}000$ queries per month, or $6{,}500$ per crawl given a twice-weekly update cycle; hundreds of webservers; and a total of a few gigabytes of data. Again, crawling is preferable to metasearch and hybrid models. Incremental crawling and software slave models would be more strongly recommended if practicable due to potential cost saving as well as to ensure freshness and coverage.

5.4 A subject-specific portal

BluePages (`bluepages.anu.edu.au`) is a set of online resources for sufferers of depression. It provides a search service covering hundreds of depression sites around the world, although the total amount of data indexed is only of the order of half a gigabyte. The sites are currently recrawled weekly and the number of queries per crawl is very low (about 100) and the per-query cost of network traffic correspondingly high.

If it were feasible, metasearch would clearly be economically attractive. However many depression "sites" comprise only a handful of pages within a large general-health or pharmaceutical site, and many do not provide a search interface.

Considerable cost savings would arise from incremental crawling, if feasible. On the other hand, the software slave alternative is totally implausible, because of the number and nature of the sites involved. Many of the depression resources are published via a technically naive individual person's website, hosted by a commercial ISP. Neither the ISP nor the person in question are likely to agree to download an indexing slave.

5.5 A whole-of-web search service

Google (`http://www.google.com`) reports that its index includes over three billion pages. Since the average web page is now around 15 kB in size, the network traffic generated by a full Google crawl is likely to be of the order of 7×10^4 GB, and the network cost, using the rate quoted in Table 5, would be over $\$1.5$ million.

Google's query load exceeds 260 million per day [Malone, 2003], or eight billion per month. Assuming a once-a-month update frequency, the network crawl cost per query is then approximately $\$2 \times 10^{-4}$. In other words, Google is processing approximately 50 queries for each cent of crawl-time network traffic, and is operating on a quite remarkable scale.

Considering the hundreds of millions of webservers accessible on the Internet [Internet Software Consortium, 2003], a FullMerge alternative to Google is laughably implausible, and it is completely unsurprising that global search engines base their services on crawling. However, there are strong commercial incentives for further cost reduction, and there is clearly a role – if they are not already being employed – for heavy use of intelligent crawling techniques similar to those described above.

5.6 Portals indexing ephemeral or protected content

The Current News Metasearcher [Rasolofo et al., 2003] is a portal search service which allows search of current news content from fifteen independent news sites. It is difficult to compare the differences in cost between the FullMerge option actually deployed and a crawled alternative, because we have no details about how rapidly the content of the sites changes, and how many pages they contain. However, to provide "instant" access to breaking news, full crawling at the necessary interval is likely to be expensive, and unless query volume is high, metasearching will be more cost-effective. Cost reductions might be achieved by employing Selective rather than FullMerge

metasearch, particularly if an increased number of news sites were included.

Metasearch (or a hybrid system) is obviously necessary where all (or some) of the included sites deny access to crawlers. This is likely to be the case for legal or medical portals.

6 Where next?

In this study we have observed and reported the operational and economic parameters of the Research Finder index of Australian research web sites and used them to develop a general framework for web search services that span multiple web servers. We have explored a range of approaches to providing the Research Finder service at lower cost without harming coverage, freshness or effectiveness.

Many of the options we have considered are not presently feasible. For example, it is not possible to assume provision of local search services by all indexed organizations, nor any willingness of organizations to run foreign indexing software on their systems. No global data vendor has yet emerged in the marketplace, and there are challenging IP issues to be resolved before one could safely do so.

The most promising cost-reduction approach in the present circumstances is to move from a fixed-cycle complete-recrawl model to incremental, variable frequency crawling. This could be incorporated into a hybrid metasearch model with further savings if result merging can be done sufficiently well that result quality can be protected.

Finally, we examined several other types of multi-server web search services in the light of our framework, and found that centralized crawling (preferably incremental and adaptive) is indeed an economically sensible model both for high-volume global web search and for a range of organizations. In the case of specialized portals like Research Finder and BluePages, more intelligent crawling is likely to considerably reduce the per-query network cost. Only in cases where content is highly ephemeral (making freshness an issue) or protected against crawlers is full metasearch truly attractive.

Acknowledgement This work was supported by a University of Melbourne-CSIRO Collaborative Research Grant.

References

J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In A. El Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K.-Y. Whang, editors, *Proc. 26th International Conference on Very Large Databases*, pages 200–209, Cairo, Egypt, September 2000.

J. Cho and A. Ntoulas. Effective change detection using sampling. In *Proc 28th International Conference on Very Large Databases*, pages 514–525, Hong Kong, August 2002.

J. Cope, N. Craswell, and D. Hawking. Automatic discovery of search interfaces on the web. In *Proc. Fourteenth Australasian Database Conference*, volume 17 of *Conferences in Research and Practice in Information Technology*, pages 181–189, Adelaide, Australia, 2003. `http://www.ted.cmis.csiro.au/~nickc/pubs/adc03.pdf`.

D. D'Souza, J. A. Thom, and J. Zobel. Collection selection for managed distributed document databases. *Information Processing & Management*, 2003. In press.

J. Edwards, K. McCurley, and J. Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In *Proc. 10th International Conference on the World Wide Web*, pages 106–113. ACM Press, 2001.

D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *Proc. 12th International Conference on the World Wide Web*, pages 669–678. ACM Press, 2003.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1, 1999. RFC 2616. Available from `http://www.ietf.org/rfc/rfc2616.txt`.

J. Gray. Distributed computing economics. Technical Report MSR-TR-2003-24, Microsoft Research, March 2003. `http://research.microsoft.com/research/pubs/view.aspx?tr_id=655`.

D. Hawking and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems*, 17(1):40–76, 1999.

Internet Software Consortium. Internet domain survey host count, January 2003. `http://www.isc.org/ds/hosts.html`.

M. S. Malone. Inside the soul of the web. *Wired Magazine*, 11.05, May 2003. `http://www.wired.com/wired/archive/11.05/google.html`.

Y. Rasolofo, D. Hawking, and J. Savoy. Result merging strategies for a current news metasearcher. *Information Processing & Management*, 39:581–609, 2003.

E. Selberg and O. Etzioni. Multi-service search and comparison using the meta-crawler. In *Proc. 4th International Conference on the World Wide Web*, Boston MA, December 1995. `http://www.w3.org/Conferences/WWW4/Papers/169/`.

C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

Z. Wu, V. Raghavan, C. Du, K. Sai C, W. Meng, H. He, and C. Yu. SELEGO: Creating metasearch engines on demand. In *Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 464–464. ACM Press, 2003.