Dessy: demonstrating mobile search and synchronization

Eemil Lagerspetz, Sasu Tarkoma
Helsinki Institute for Information Technology HIIT
University of Helsinki
P.O. Box 68 FI-00014 University of Helsinki, Finland
Email: firstname.lastname@hiit.fi

Tancred Lindholm
Helsinki Institute for Information Technology HIIT
Aalto University
P.O. Box 5400 FI-02150 Espoo, Finland
Email: tancred.lindholm@hiit.fi

***Abstract*—The storage capacity of smartphones has reached tens of gigabytes, while the search functionality remains simple. We have designed a search and synchronization framework for mobile devices, called Dessy. Dessy has been designed with mobility and device constraints in mind. It requires only MIDP 2.0 Mobile Java with FileConnection support, and Java 1.5 on desktop machines. This paper demonstrates the application in practice, using multiple devices and synchronizing files between a desktop computer, a laptop, smartphones, and the Internet. Smartphones and laptops are able to search for files hosted on other devices as well as on the Internet. Unnecessary search operations are avoided using Bloom filters.**

I. Introduction and Background and Related Work

The storage capacity of smartphones is growing, while file search functionality on mobile platforms remains simple. Existing desktop search software allows searching the content of files on desktop computers, but not mobile platforms. Constraints such as limited battery life, CPU power and memory constraints pose additional challenges to the design of software on mobile platforms. Indexing the file system on the mobile phone may not be practical. Desktop search refers to finding local files without iterating the file system, in reasonable time, and by at least their content in addition to their names. This can be accomplished by use of an index. Recently, Copernic Desktop Search[1] enabled searching the desktop's files from a web page, allowing mobile devices to search them.

In order for a desktop search and synchronization system to gain acceptance on smartphones, energy awareness is required. In addition to ease of use, such an application must conserve energy by using communication hardware, persistent storage and the CPU sparingly. This paper demonstrates the Dessy search and synchronization framework for mobile devices, such as smartphones and PDAs. Our system is called *Dessy*. Dessy finds files by their content, metadata, and context information. It enables searching for files on the local file system, remote computers, and the Internet. Dessy supports remote searching and index synchronization. This offloads the indexing and crawling task from the mobile device to a more capable machine. Dessy uses Bloom filters to avoid searches that would return no results for both local and remote searches. The modular architecture of the system allows disabling energy–hungry parts of the system on different platforms.

To allow access to the user's data on other devices, file synchronization can be used. The user may synchronize files as they are required on the phone. This mitigates the storage space gap between the desktop and the smartphone. By syncronization, we mean bidirectional file synchronization. Changes at either end of the synchronization are to be propagated to the other, possibly causing change reconciliation. After a final version has been agreed on, both ends will have the same final version.

The following use–case illustrates Dessy. Mr. Smith is commuting, and reading a computer science article, *Dessy: Towards Flexible Mobile Desktop Search*, on his smartphone. The article refers to another article, titled *A three-way merge for XML documents*, by Mr. Lindholm. Mr. Smith decides that he should read *A three-way merge for XML documents* to obtain a proper understanding of *Dessy: Towards Flexible Mobile Desktop Search*. Mr. Smith may have the article already on his computer, so he decides to use Dessy to search for it, instead of an Internet search engine. After all, Dessy can also search the Internet. To find *A three-way merge for XML documents*, Mr. Smith types the search terms `three-way merge for XML documents` and `author:Lindholm` into Dessy, and clicks `Search`. Dessy reports that there are no local results on the smartphone, and none on Mr. Smith's computer either. However, Dessy shows promising results via the Internet: the result description contains the title, *A three-way merge for XML documents* and the name Lindholm. So, Mr. Smith chooses `Synchronize` on the Internet search result, and *A three-way merge for XML documents* is downloaded to the smartphone. Mr. Smith then opens it in his PDF reader and proceeds to read it.

Dessy searches are executed in multiple, user–defined locations. Possible locations include the phone running Dessy, remote desktop machines, and Internet search sites, such as Google. All resulting files can be synchronized; remote Dessy hosts synchronize files bidirectionally with the phone, while files on Internet hosts are downloaded if they have changed since last synchronization. Dessy provides an interface for locating files for both users and applications. Dessy uses the Syxaw file synchronizer with XML-awareness [1] to synchro-

[1] http://www.copernic.com/

nize found files. Syxaw is designed for limited devices, and follows an efficient synchronization protocol that reduces the number of network round-trips required for synchronization. Syxaw enables separate synchronization of file metadata and data, useful for index synchronization. It also provides Dessy with unique file identifiers suitable for efficient index metadata storage. Dessy is able to find files by their names, content, user-assigned tags, metadata, and context information, such as EXIF data of JPG files, author, subject, and keywords of PDF files, and so forth. Search results can be synchronized when connectivity is available.

Dessy differs from most desktop search software in being mobile. Dessy can be used on desktop computers, MIDP 2.0[2] / CLDC 1.1[3] smartphones, and Java Foundation Profile PDAs. While the MIDP 2.0 / CLDC 1.1 is very widespread among mobile handsets, in terms of desktop search, the platform is very limited. It does not allow accessing files in any way. The JSR 75 PDA optional packages for J2ME provide a very basic stream–based method for file access[4]. The JSR 75 API is used in Dessy.

The open–source Tracker project[5] allows searching for contacts and multimedia files on the Nokia N900 mobile platform. Further search options will be added in future releases of the Maemo (to be MeeGo) platform.

Existing desktop search software for desktop computers includes systems such as Apple's Spotlight[6], the Tracker project, Google Desktop[7], Microsoft's SiS [2], and Windows Search[8].

To provide users with an easy to use search and synchronization system, one needs to study the users and how they use search and synchronization applications. Searching for local files is in principle similar to searching for documents on the Internet. The subject of searching the Web has been increasingly studied recently. In [3] and [4], the distribution of Internet search queries was analyzed. In both studies, queries consisted of 2.3 and 2.7 words on average for phones and PDAs, respectively. The latter study suggests that mobile users spend 56 – 63 seconds inputting a query on a mobile phone, and 27 – 35 seconds on a PDA. It also noted that users rarely look beyond the first page of results. Based on this, an application that displays a concise list of ranked results from multiple locations, together with content snippets would probably be appropriate for most users. For example Google displays results with snippets; it shows a few sentences from each result web page, and a thumbnail picture for image results.

Desktop search has also been examined using virtual directories [5] and directory namespaces [6]. Dessy uses virtual directory paths to represent queries.

[2]http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html
[3]http://jcp.org/aboutJava/communityprocess/final/jsr139
[4]JSR 75, http://jcp.org/aboutJava/communityprocess/final/jsr075
[5]http://projects.gnome.org/tracker/
[6]http://www.apple.com/macosx/features/spotlight/
[7]http://desktop.google.com
[8]http://www.microsoft.com/windows/products/winfamily/desktopsearch

Fig. 1. A screenshot of Dessy running on the Nokia N900.

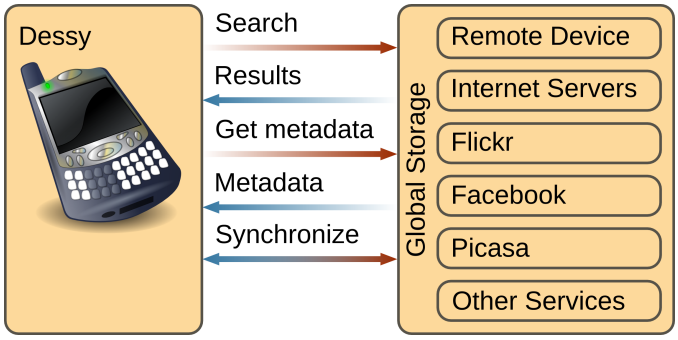

Fig. 2. A view of Dessy search and synchronization.

Synchronization systems for mobile devices have been developed [7]–[9]. These use operation shipping, which may not be practical for everyday users. The SyncML synchronization protocol [10] has mostly been used for personal information synchronization.

A lot of software has been developed for gathering context data on mobile platforms. Dessy could easily be coupled with, e.g. the BeTelGeuse [11] data–gathering software. A camera application combined with Dessy and BeTelGeuse could tag new photos with the current context variable values, such as the current location, nearby Bluetooth devices, and calendar events. The photos could later be synchronized to the user's desktop computer, Google Picasa, Flickr, and other cloud services using Dessy.

II. Demo Scenario

The Dessy search and synchronization framework runs on smartphones, PDAs and laptop and desktop computers. It can also connect to various online services. In effect, Dessy can use cloud–based indexes to help maintain a user's personal file space. In the demonstration scenario, the Dessy system runs on smartphones, a laptop, and a desktop machine. It will also search the Internet via Google. Figure 1 shows the Dessy GUI running on a N900 smartphone. The top of the interface contains a search bar used to search all connected sources. The connection and synchronization options are accessible

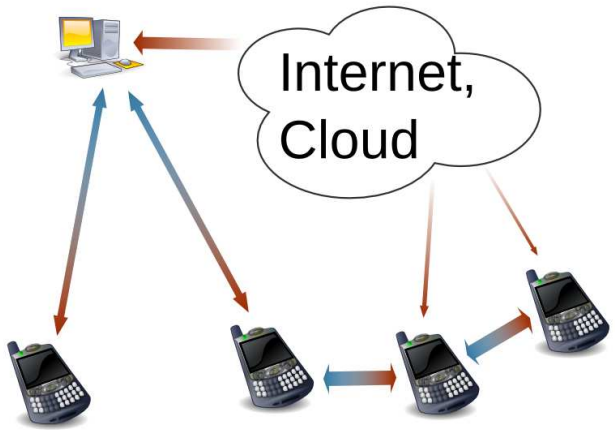

Fig. 3. a dessy search and synchronization scenario.

from the menu bar. The left side of the interface shows the properties and values of metadata that correspond to the currently selected file on the right side, if any. The text field below allows adding custom metadata, or tags, to local files. Finally, The statusbar at the bottom shows recent events.

Figure 2 shows a high–level view of a Dessy search and synchronization exchange. In the demonstration scenario, the device that initiates the search and synchronization operations is in the role of the Dessy device on the right side, while the other endpoint is either a remote device running Dessy or an HTTP server. Bidirectional synchronization is enabled between two Dessy instances. With HTTP servers, files are fully downloaded, but only if the target file is newer than the local copy. Searches in the Internet are conducted through Google. Remote and local device searches are handled by Dessy. Remote and local device searches that would have no results are avoided using a Bloom filter.

The scenario depicted in Figure 3 demonstrates Dessy. In the Figure, the smartphones on the right side synchronize files with the Cloud. In our demonstration, the devices will download files from the Internet, using the Dessy Internet search interface. The rightmost device in the Figure will synchronize its files with the one beside it, which will in turn synchronize files with the third smartphone from the right. The two leftmost smartphones will not synchronize with each other directly, but will synchronize their files with a desktop machine. The desktop machine also receives files from the Internet and lets the two smartphones synchronize those files when they see fit.

REFERENCES

[1] T. Lindholm, J. Kangasharju, and S. Tarkoma, "Syxaw: Data synchronization middleware for the mobile web," *Mobile Networks and Applications*, vol. 14, no. 5, pp. 661–676, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11036-008-0146-1

[2] E. Cutrell, S. T. Dumais, and J. Teevan, "Searching to eliminate personal information management," *Communications of the ACM*, vol. 49, no. 1, pp. 58–64, Jan. 2006.

[3] J. Yi, F. Maghoul, and J. Pedersen, "Deciphering mobile search patterns: a study of yahoo! mobile search queries," in *The Seventeenth World Wide Web Conference*, Apr. 2008, pp. 257–266.

[4] M. Kamvar and S. Baluja, "A large scale study of wireless search behavior: Google mobile search," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, Eds. ACM Press, Apr. 2006, pp. 701–709.

[5] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. James W. O'Toole, "Semantic file systems," in *SOSP '91: Proceedings of the thirteenth ACM symposium on Operating systems principles*. ACM Press, 1991, pp. 16–25.

[6] C. K. Hess and R. H. Campbell, "An application of a context-aware file system," in *CHI '03 extended abstracts on Human factors in computing systems*, G. Cockton and P. Korhonen, Eds., Apr. 2003, pp. 339–352.

[7] H. Mei and J. Lukkien, "A remote personal device management framework based on syncml dm specifications," in *MDM '05: Proceedings of the 6th international conference on Mobile data management*. New York, NY, USA: ACM, 2005, pp. 185–191.

[8] Y.-W. Lee, K.-S. Leung, and M. Satyanarayanan, "Operation shipping for mobile file systems," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1410–1422, Dec. 2002.

[9] T.-Y. Chang, A. Velayutham, and R. Sivakumar, "Mimic: raw activity shipping for file synchronization in mobile file systems," in *Mobisys 2004 Workshop on Context Awareness*, June 2004, pp. 165–176.

[10] *SyncML Sync Protocol, version 1.1*, SyncML Initiative, Feb. 2002. [Online]. Available: http://www.syncml.org/docs/syncml_sync_protocol_v11_20020215.pdf

[11] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson, "BeTelGeuse: A Platform for Gathering and Processing Situational Data," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 49–56, 2009.