

Cloud-Assisted Mobile Desktop Search

Eemil Lagerspetz

Helsinki Institute for Information Technology HIIT

University of Helsinki

P.O. Box 68 FI-00014 University of Helsinki, Finland

Email: eemil.lagerspetz@hiit.fi

Sasu Tarkoma

Helsinki Institute for Information Technology HIIT

University of Helsinki

P.O. Box 68 FI-00014 University of Helsinki, Finland

Email: sasu.tarkoma@hiit.fi

Abstract—With the advent of the Internet, searching for information became a fundamental part of our daily computing. When large storage devices were integrated to our personal computers, desktop search solved the problem of locating information. With larger storage integrated into mobile devices, a solution for the problem is required in the mobile environment. Users often possess multiple devices, such as laptops, PDAs and smartphones, making information synchronization more important. The Dessy desktop search and synchronization system for mobile devices strives to answer these challenges. This paper introduces some of Dessy’s features and demonstrates the Dessy prototype for mobile devices.

I. INTRODUCTION

The wealth of information on the Internet has shifted the main problem in obtaining information to locating it. This is currently solved by Internet search engines with varying degrees of success. The search of information stored in our personal computers became relevant as the storage space of commodity computers grew from a few hundred megabytes to thousands of gigabytes. This problem is being solved by desktop search software, such as Beagle, Google Desktop, Copernic Desktop Search, and the Tracker project.

As today’s smartphones, PDAs and other mobile devices grow in storage capacity towards several hundred gigabytes, the problem of locating information becomes apparent. The fact that many mobile phone and computer users have multiple devices further compounds the problem, and motivates file synchronization between mobile devices. Furthermore, the limited connectivity and connection bandwidth of the mobile environment coupled with much larger storage capacities on desktop machines motivate a search and synchronization platform. We propose using the same desktop search techniques in the mobile environment, coupled with a mobile file synchronization system.

Desktop search techniques cannot be applied unchanged, however. The limited battery life, memory capacity and processing power of smartphones and other mobile platforms places certain restrictions on software that is to run on them. An application needs to conserve energy by carefully managing CPU and connectivity hardware use. Memory management is crucial for smooth running of the application. The heterogeneous connectivity of the mobile environment

means that software must be able to operate in low bandwidth or disconnected mode.

The rise of cloud services and pervasive if low-bandwidth connectivity of modern mobile platforms provide further sources of personal information to search and synchronize. The Dessy system is designed to be energy-efficient and to allow offloading a portion of its tasks to remote desktops and into the cloud. Storage and indexing-related tasks can be performed on a remote host, while the local device simply queries and synchronizes information of interest.

The contributions of this paper are:

- 1) A desktop search and synchronization architecture for mobile devices with a working implementation.
- 2) Search and synchronization with mobile devices, desktop computers and the cloud, capable of offloading tasks to remote hosts.

This paper is divided into three sections. Section II discusses the Dessy system and related research. Section III points out the highlights of the system. Section IV introduces the demonstration scenario. Section V lists the technical requirements of the demonstration.

II. DESSY AND RELATED WORK

Desktop search, or content retrieval on a local computer, has been widely studied on personal computers. Systems such as Google Desktop¹, Microsoft’s SiS [1], Apple’s Spotlight², Novell’s open-source project Beagle³, the Tracker project⁴, and Copernic Desktop Search⁵ are known to many computer users. Finding files through search folders or *virtual directories* [2] and directory namespaces [3] has also been demonstrated. To enable search with more than keywords, the use of context information and user-friendly metadata was explored by Cuttrell et al [4] and Hess et al [3] by using application and user-defined *tags*. A tag is an arbitrary keyword or a property — value pair assigned to an object, such as `tag:important` for an email or `location:france` on a set of photos.

¹<http://desktop.google.com>

²<http://www.apple.com/macosx/features/spotlight/>

³http://beagle-project.org/Main_Page

⁴<http://projects.gnome.org/tracker/>

⁵<http://www.copernic.com>

While Copernic Desktop Search enables searching the computer through a web-based client, to our knowledge the indexing and search of information stored on a mobile device and personal storage services has not been demonstrated.

SyncML [5] has been used extensively for synchronizing personal information. Its use for synchronization of regular files has been limited. Operation shipping [6]–[8]. has also been explored using PDAs. However, regular synchronization is the most common approach. Automatic reconciliation of concurrent changes during file synchronization is done in some computer-based synchronization solutions, such as Unison⁶ [9]. Dessy uses the Syxaw file synchronizer with XML-awareness [10], since it is open-source, allows changeable conflict reconcilers, and is ported to the mobile Java platform. Furthermore, Syxaw allows linking files stored on Dessy instances with different remote servers. Its XML-aware merging algorithm [11] allows automatic detection and reconciliation of changes in the directory tree, such as moves, renames and deletions.

Dessy was first introduced in [12]. Dessy has been written in Java for multi-platform support and fast prototyping. Dessy enables indexing, search, and file synchronization of a mobile device with connected online services and remote desktops. It uses virtual directories for search and supports free-form custom metadata tags. Users can tag local and remote files, and synchronize tags between endpoints.

The design of Dessy is modular, separating indexing, querying, and synchronization. This allows disabling parts of the framework on a mobile device in order to offload their tasks to a desktop machine. Particularly, a mobile client may synchronize newly created or changed files in batches with a desktop machine, and let the desktop index the files. The user can then query for local and remote files without expending CPU time for indexing. Dessy uses desktop search techniques, such as inverted file indexing, BM25 result relevance ranking, and stemming and lemmatization. The system has been ported to the Mobile Information Device Profile 2.0 / Connected Limited Device Configuration 1.1 Mobile Java specification. This allows running Dessy on Java-enabled smartphones.

III. SYSTEM HIGHLIGHTS

Figure 1 illustrates common usage of Dessy. A mobile Dessy client connects to a number of online services and a remote desktop running Dessy. The client searches for a file using keywords, property — value pairs, file metadata and user-defined tags. The interfaces to cloud services may reside on the mobile itself, or they can be accessed remotely through another Dessy host. The searches are executed in parallel on all connected endpoints. Local indexing and local search on the mobile may be disabled in order to conserve battery life. These tasks can be accomplished by remote

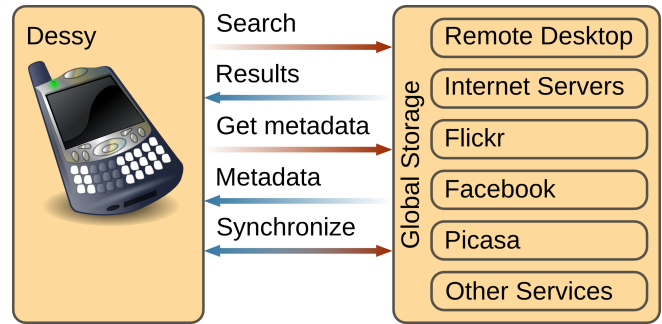


Figure 1. An illustration of Dessy search and synchronization.

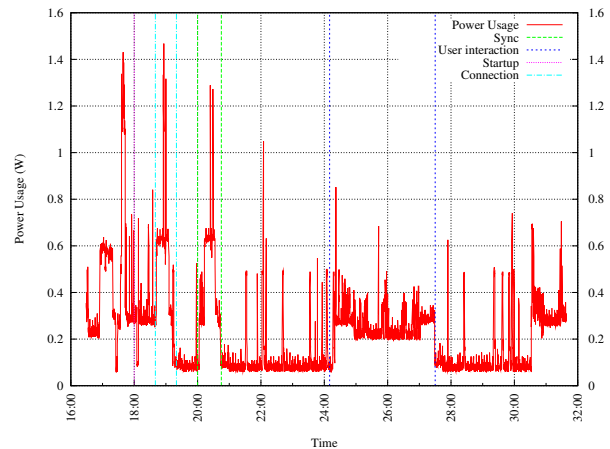


Figure 2. Power measurements of Dessy operations.

hosts when files are synchronized with them. The client can choose to receive extended metadata on all the results, request metadata on specific result files, or forgo receiving extra metadata in order to conserve battery life. The client may then synchronize all files or a number of selected files. Synchronization with remote Dessy instances and some storage services is bidirectional, while synchronization with HTTP servers merely pulls changed files from the server and merges the changes on the client.

Dessy has been designed with energy awareness in mind. Figure 2 shows power measurements of Dessy in use, split up into relevant operations. From the left of the Figure to Startup shows the power use of the system when starting up. This involves selecting the Dessy Midlet on the phone menu and the startup of the Java VM and the Dessy system and its graphical user interface. Connection corresponds to the system connecting to the remote Dessy server. This includes an RSA key exchange, encrypting and decrypting cryptography test messages, and receiving, parsing and showing the Dessy server’s root folder on the phone. Sync delimits synchronizing a single file with the remote server. Finally, User interaction delimits a period

⁶<http://www.cis.upenn.edu/~bcpierce/unison/>

Operation	Battery capacity	Average power
System startup	395 times	0.43 W
Initial connection	743 times	0.49 W
WLAN Synchronization	980 times	0.52 W
User interaction	15 hours	0.26 W
System idle	61 hours	0.069 W

Table I
BATTERY CAPACITY IN TERMS OF SYSTEM OPERATIONS.

of constant user interaction. During this period, the user is constantly scrolling up and down the list of search results.

Table I summarizes the power usage measurements of different Dessy operations with numbers and durations permitted by a full standard battery of a Nokia E51 smartphone. The durations are calculated from the measurements of Figure 2 and experimental synchronization power usage measurements. `WLAN Synchronization` refers to using the Dessy software to synchronize a file 0.5 MB in size. This is the average size of PDF files in the synchronization experiment. The 980 synchronizations therefore amounts to 490 MB of data. This is not much, but considering text or PDF documents and synchronizing their changes only, it is difficult to fill this amount of data in a short time after the initial full download synchronization.

Our measurements give Dessy an idle lifetime of more than two days when mostly idle, and more than twelve hours in active use. This suggests that the energy usage of Dessy is reasonable. Nevertheless, we will continue to measure and improve the energy usage of the system on different mobile platforms.

IV. DEMONSTRATION SCENARIO

In the demonstration, the audience will see mobile cloud-based data synchronization in action, and have the possibility to interact with the system. The techniques and solutions used are also explained with sequence diagrams and with experimental performance measurements. To further illustrate the potential benefits of mobile synchronization, a second interaction involves several mobile phones and allows users to share photos and videos, for example, pertaining to Percom events.

In another scenario, two Dessy instances are running, one on a mobile phone, a Nokia N97, and another on a remote Linux desktop. The file system of the mobile is empty at the beginning of the scenario. The mobile carries out a number of searches to find a certain PDF file. The searches are performed on the mobile, the desktop, and in the Internet, through Google. This illustrates the access to information distributed globally across different services, and finding such information based on extended metadata in addition to simple keywords. The mobile then synchronizes a PDF file with an Internet server, and another with the desktop. Synchronization of files and automatic reconciliation of

synchronization conflicts are key features of the system. After the synchronization, the mobile opens the PDF file, examines the PDF metadata, and proceeds to reading it.

The interactions in our demonstration show that synchronization is an essential component when fully utilizing cloud-based services on mobile devices. Distributed mobile desktop search can help the users in their daily lives and make it easier to find information and keep it up to date. The current challenges include energy efficiency and being able to make intelligent synchronization decisions, essentially what to synchronize and when. We discuss these challenges during the demonstration.

V. TECHNICAL REQUIREMENTS

The technical requirements for the demonstration are:

- A projector with VGA and component video support to project the laptop and smartphone screens. We can also bring our own.
- Electric sockets for mobile phone and laptop chargers.

REFERENCES

- [1] 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.
- [2] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. James W. O'Toole, "Semantic file systems," in *SOSP '91*. ACM Press, 1991, pp. 16–25.
- [3] C. K. Hess and R. H. Campbell, "An application of a context-aware file system," in *CHI '03*, Apr. 2003, pp. 339–352.
- [4] E. Cutrell, D. Robbins, S. Dumais, and R. Sarin, "Fast, flexible filtering with phlat," in *SIGCHI '06*. ACM Press, Apr. 2006, pp. 261–270.
- [5] *SyncML Sync Protocol, version 1.1*, SyncML Initiative, Feb. 2002. [Online]. Available: http://www.syncml.org/docs/syncml_sync_protocol_v11_20020215.pdf
- [6] H. Mei and J. Lukkien, "A remote personal device management framework based on syncml dm specifications," in *MDM '05*. New York, NY, USA: ACM, 2005, pp. 185–191.
- [7] 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.
- [8] 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.
- [9] E. I. Bolso, "File synchronization with unison," *Linux Journal*, vol. 2005, no. 132, pp. 6–6, 2005.
- [10] 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>
- [11] T. Lindholm, "A three-way merge for XML documents," in *ACM Symposium on Document Engineering*, E. V. Munson and J.-Y. Vion-Dury, Eds. ACM Press, Oct. 2004, pp. 1–10. [Online]. Available: <http://www.hiit.fi/files/fi/fc/papers/doceng04-pc.pdf>
- [12] E. Lagerspetz, T. Lindholm, and S. Tarkoma, "Dessy: Towards flexible mobile desktop search," in *Proceedings of the Fourth ACM SIGACT-SIGOPS International Workshop on Foundations of Mobile Computing*, 2007.