SYNCROD: AN ENERGY-MINIMIZATION METHOD FOR DISTRIBUTED FILE SYNCHRONIZATION ALGORITHMS ON MOBILE DEVICES

Zach Rattner[1], Hamilton Turner[1], Danny Guymon[1], Brian Dougherty[1], Jules White[1] and Sean Eade[2]
[1]*Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia, U.S.A.*
[2]*Siemens Corporate Research, Princeton, New Jersey, U.S.A.*

Keywords: Energy Optimization, File Synchronization, Mobile, Wireless Networking.

Abstract: As mobile computing platforms become ubiquitous, the need to keep data synchronized between multiple devices becomes increasingly common. However, mobile devices have limited battery capacity, and file synchronization requires extensive use of power-hungry network interfaces. This paper introduces Syncrod, an approach for optimizing file synchronization algorithm parameters in order to minimize total energy consumption. This paper presents a formal model for describing mobile file synchronization energy consumption, and an example of fitting a general file synchronization algorithm to the provided model. Empirical results of running an energy optimization on a general file synchronization algorithm are shown to provide rapid file transfer while using near 0.003% of a standard smartphone battery energy per MB. The provided approach can be used to find the the most energy-efficient parameters for any file synchronization algorithm that can be fit into the provided formal model.

1 INTRODUCTION

Gartner research shows that smartphone sales have increased 96% from 2009 to 2010(Gartner, 2010). As smartphones continue to grow in popularity, consumers are creating, editing, and sharing data from many computing devices, such as cellular devices, laptop computers, and tablets. This surge in the viewing and editing of content from both mobile devices and home computers has increased the need for distributed file synchronization software. As files are edited on a mobile device or home computer, distributed file synchronization software automatically transfers the changes between devices so that all computers have an up-to-date copy of the data.

Distributed file synchronization approaches transfer data using the wireless network interfaces, such as Wi-Fi and 3G on a mobile device, which consumes a significant amount of energy (Lagerspetz et al., 2010; Mummert et al., 1999). Energy consumption is a critical concern on mobile devices due to their limited battery capacity (Agarral et al., 2002; Balasubramanian et al., 2009; Mudge, 2001; Shye et al., 2010). Prior work on distributed file synchronization has focused on optimizing synchronization quality of service (QoS) properties, such as end-to-end synchronization time.

Open Problem ⇒ Determining how to Minimize Power Consumption of Distributed File Synchronization Algorithms for Mobile Devices.

Due to the high importance of energy consumption on mobile devices, there is a need for approaches to aid in optimizing distributed file synchronization algorithms to consume less energy on mobile devices. Distributed file synchronization algorithms rely on a number of configurable parameters, such as the block and hash size used to track and transfer changes, which have a direct impact on power consumption. Moreover, depending on the context of a device (*e.g.*, on a 3G or Wi-Fi network), the optimal values for these parameters vary.

Solution Approach ⇒ A Modeling and Optimization Approach for Deriving Distributed File Synchronization Parameter Values to Minimize Energy Consumption.

In order to address the challenges of deriving algorithm parameters that result in lowest overall energy usage, this paper presents Syncrod i.e. **sync**h**ro**nization **d**aemon. Syncrod is a formal model for estimating the energy consumption of a distributed

Rattner Z., Turner H., Guymon D., Dougherty B., White J. and Eade S..
SYNCROD: AN ENERGY-MINIMIZATION METHOD FOR DISTRIBUTED FILE SYNCHRONIZATION ALGORITHMS ON MOBILE DEVICES.
DOI: 10.5220/0003822601350141
In *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2012), pages 135-141
ISBN: 978-989-8565-00-6

file synchronization algorithm with varying block, hash size, compression ratio, and other algorithmic parameters. Syncrod also provides an optimization method that can be used to automatically select file synchronization algorithm parameters, such as file compression ratio, in order to assist distributed file synchronization algorithm developers in creating energy-efficient file synchronization algorithms. Syncrod uses a formal model of data synchronization outlined in Section 3.

This paper provides the following contributions to the study of distributed file synchronization algorithms optimized for mobile applications:

- This paper presents a formal model, built from empirical data, for estimating the energy consumption of distributed file synchronization algorithms.
- This paper describes Syncrod, a method for deriving the algorithm parameters values for a distributed file synchronization algorithm to minimize energy consumption.
- This paper presents an example of using the Syncrod optimization method to locate the most energy-efficient algorithm parameters for a specific distributed file synchronization algorithm.

The remainder of this paper is organized as follows: Section 2 describes the challenges that make it difficult to derive the distributed file synchronization algorithm parameters that result in minimal total energy usage; Section 3 formally defines the energy used during the file synchronization process; Section 4 empirically evaluates the impact of Syncrod on execution time and energy consumption; Section 5 presents related work; and Section 6 presents concluding remarks and lessons learned.

2 CHALLENGES OF DETERMINING ENERGY-EFFICIENT PARAMETER COMBINATIONS FOR FILE SYNCHRONIZATION ALGORITHMS

Determining the most energy-efficient combination of algorithm parameters for a distributed file synchronization algorithm is challenging due to the difficulty of understanding how algorithm software parameter values impact power consumption. Algorithm parameters values have complex compounding effects upon lower software layers, such as the operating system, network stack, and device drivers. Moreover, dynamically changing mobile environment properties, such as available signal strength or network types, can cause significant changes in synchronization algorithm operation, thereby changing the parameter values that will result in minimal power consumption. The remainder of this section describes, in detail, the challenges of identifying the file synchronization algorithm parameter values that will result in minimal energy consumption for various environments.

2.1 Challenge 1: Determining the Impact of File Synchronization Parameters on Energy Consumption is Difficult due to Software Abstraction

In a distributed file synchronization algorithm for mobile devices, the algorithm's input parameters are the primary method of impacting total energy consumption. For example, increasing the amount of compression applied before transmitting the file across a network interface may increase the total CPU time while decreasing the total transmission time. Both CPU time and wireless data transfer impact power consumption, and thus precisely quantifying the impact of these types of trade-offs on energy consumption is hard.

Similarly, there are many layers of software abstraction between developer-written software and energy-consuming hardware, and there are often no built-in methods for tracking the energy used by a specific piece of software. Additionally, run-time events, such as a user choosing to synchronize a large audio file, can result in different software branch executions with different energy consumption characteristics. This hidden complexity between algorithm parameters and power consumption, coupled with the possibility of run-time events, makes it difficult to determine what algorithm parameter values will result in minimal total energy usage. Section 3.3 identifies multiple input parameters that can be used to modify the behavior and energy consumption of a distributed file synchronization algorithm, and Section 3.2 derives general equations useful for measuring energy consumption of each stage of a distributed file synchronization algorithm.

2.2 Challenge 2: Discerning the Most Energy-efficient Network for a Specific Context is Difficult

Mobile devices transmit and receive data over a variety of wireless network types, such as Wi-Fi, 3G, and Bluetooth. A key complexity for choosing parameter values to minimize energy consumption is that each type of network has a different energy consumption profile that is most power efficient for a specific type of data transfer.

For example, if a large file is being synchronized for the first time, an energy consuming high-bandwidth network interface, such as Wi-Fi, may actually require less overall energy consumption than a lower-bandwidth and less energy consuming alternative, such as 3G. Energy consumption is calculated as the instantaneous energy consumption of the network interface multiplied by the amount of time taken to transfer the data. For large files, the increase in transfer time of the lower-bandwidth 3G interface can end up in a higher total energy consumption versus Wi-Fi. Conversely, if a small amount of changed data is being transmitted, then a low-power low-speed network may be appropriate.

3 FORMAL DEFINITION OF ENERGY CONSUMPTION IN DATA SYNCHRONIZATION ALGORITHMS

To address the challenges described in Section 2, we developed a formal method for estimating energy consumption of distributed file synchronization algorithms. This model allows our optimization method to be easily applied to other work in the field of distributed file synchronization. The overall energy equations presented here be used as the objective functions in Section 4 to derive parameter values that minimize energy consumption.

3.1 Syncrod & Power Optimization Overview

Creating a method of deriving the most power-efficient parameters for a distributed file synchronization algorithm requires an in-depth understanding of how file synchronization components impact power consumption on a mobile device. This understanding is hard to attain, due to the vast amount of complexity encapsulated between application-level software and power-consuming hardware. This significant complexity prompted the use of a formal model that would provide an objective function capable of being optimized. Section 1 presents a formal model for distributed file synchronization energy consumption. The completed formal model of the distributed file synchronization algorithm described in Section 3.2 is empirically evaluated in Section 4 to determine the algorithm parameters that resulted in the least overall energy usage. The algorithm parameters derived show that the algorithm presented in 3.2 is able to transfer a 1MB file using less than 0.003% of a standard smartphone battery capacity. The formal model presented can be directly used with other distributed file synchronization algorithms and use the optimization methods presented in Section 4 to derive energy-efficient values for their algorithm's parameters.

3.2 Formal Definition of Data Synchronization Process

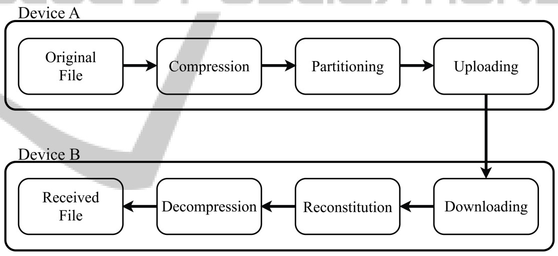

Figure 1: Overview of upload procedure on mobile device.

An example distributed file synchronization algorithm is presented here for motivation. This algorithm is outlined in Figure 1. The algorithm operates by first compressing large files, and then breaking these compressed files into blocks during the partitioning stage. In the initial synchronization, all of these blocks are sent across the network to the remote device. In subsequent synchronizations, only the blocks that have modifications must be sent across the network. The receiving device can 'stitch' the received block into its existing set of blocks to reconstitute the entire file and then decompress the completed set of blocks to form the original file. We use the following subscripts to refer to specific steps in the synchronization algorithm:

- *com*, compression stage,
- *part*, partitioning stage,
- *up*, upload stage,
- *down*, download stage,
- *rec*, reconstitution stage,

- *dec*, decompression stage.

3.2.1 Total Energy Usage

The total energy usage e can be defined as the sum of the energies used at each step of the algorithm, e.g. the energy used during compression e_{com}, the energy used during partitioning e_{part}, etc. The equation for total energy is defined as follows:

$$e(F,C,B,H,N) = \tag{1}$$
$$e_{com}(F,C) + e_{part}(C,B) + e_{up}(F,B,H,N) +$$
$$e_{down}(F,B,H,N) + e_{rec}(F,C) + e_{dec}(F,C)$$

3.2.2 Total Energy Represented as Power over Time

It makes sense to represent the total energy usage in terms of power draw, by integrating power in each stage of the synchronization algorithm over the amount of time spent in that stage, as shown in Equation 2.

$$e(F,C,B,H,N) = \tag{2}$$
$$\int_0^{t_{com}(F,C)} P_{com}(t)\,dt + \int_0^{t_{part}(C,B)} P_{part}(t)\,dt +$$
$$\int_0^{t_{up}(F,B,H,N)} P_{up}\,dt + \int_0^{t_{down}(F,B,H,N)} P_{down}\,dt +$$
$$\int_0^{t_{rec}(F,C)} P_{rec}(t)\,dt + \int_0^{t_{dec}(F,C)} P_{dec}(t)\,dt$$

If we can derive the specific power equations and time values for a distributed file synchronization algorithm, then this high-level energy function is usable as the objective function for minimization optimization.

3.2.3 Execution Time

The overall time taken by the synchronization algorithm is the sum of the time taken by each stage of the algorithm. For simplification, we assume that the uploading and downloading (t_{up} and t_{down}) are completely separate stages and do not occur simultaneously. More complex equations that model simultaneous upload/download can be defined in future works.

$$t(F,C,B,H,N) = \tag{3}$$
$$t_{com}(F,C) + t_{part}(C,B) + t_{up}(F,B,H,N) +$$
$$t_{down}(F,B,H,N) + t_{rec}(F,C) + t_{dec}(F,C).$$

3.3 Distributed File Synchronization Design Decisions that Impact Energy Consumption on Mobile Devices

The purpose of our optimization approach was to find values for the following algorithm parameters that resulted in minimal energy consumption due to distributed file synchronization:

- B - The size of each data block size, in bytes. Because only complete blocks are sent across the network, overly large values for B will result in extraneous network traffic when only a few bits of a file are changed, and the extra network traffic will result in increased energy consumption. Conversely, overly small values of B will result in a need to transfer many hash values across the network, again resulted in extraneous network traffic.
- H - The size of each data block hash, in bytes. Overly small values for H will increase the probability of a hash collision, which would fool the synchronization algorithm into believing a file has not been changed when in fact is has. Overly large values for H result in increased network traffic and increased CPU load, both of which increase energy consumption.
- C - Represents the amount of time and processing power that will be used in attempting to compress the file. Overly large values for C will increase CPU load when compressing and decompressing files, while values of C that are too small will result in increased network traffic. Both of these error types can increase energy consumption.
- N - The network type to use to transmit the data. The chosen network has a significant effect on both data transfer speed and energy used to transmit the files.

Additionally, F represents the file size, in bytes, of the file to be synchronized. This parameter is not a design variable, since the algorithm has no choice over the file that will be synchronized. However, a value for F is required to calculating the values of the objective function e.g. Equation 2.

4 USING Syncrod TO LOCATE DISTRIBUTED FILE SYNCHRONIZATION ALGORITHM PARAMETERS THAT RESULT IN MINIMAL ENERGY USAGE ON MOBILE DEVICES

We attempted to minimize the time spent in each of the six sections of the distributed file synchronization algorithm. Since energy consumption is the result of power usage over time, we reasoned that minimizing the time taken would result in lower energy usage.

4.1 Experimental Platform

The platform used to execute these experiments was a Dell Inspiron 530 with Intel Core 2 Duo E7200 executing at 2.53GHz, with 1 GB of memory and 32 GB of available disk space.

4.2 Derivation of Algorithm-specific Optimization Functions

In order to leverage the Syncrod formal model for estimating energy consumption, a developer must fit specific power and time equations into the objective function in Equation 2. Due to space constraints we cannot outline all of our synchronization algorithm-specific functions, but we emperically measured our algorithm's time and power consumption for each stage of synchronization across a range of file types and sizes.

4.3 Experiment 1: Execution Time Minimization with Syncrod

As file synchronization involves both uploading and downloading files, it is important to not bias the optimization in favor of one phase over the other. Equation 3 corresponds to uploading a file using the synchronization algorithm, and later downloading the same file back to the same device under the same network conditions. This procedure is illustrated in Figure 2. In this test, a filesize of $F = 1$ MB was chosen.

The result of a distributed file synchronization algorithm minimization is a set of configuration parameters for that algorithm. In our chosen algorithm e.g. Syncrod, the parameters are B, H, C, and N (see Section 3.3 for explanations). C and N have a fairly limited set size (10 and 2 respectively), and multiple time equations (t_{com}, t_{rec}, and t_{dec}) depend solely upon C, N, or the derived parameter $R(C)$. Therefore, we decided to enumerate all possible input parameter combinations of C and N and generate solutions for t_{com}, t_{rec}, and t_{dec}. The results are shown in Table 1.

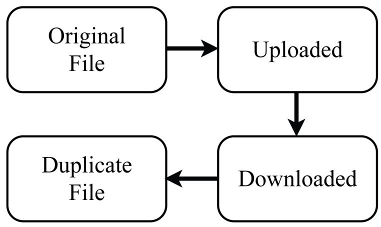

Figure 2: Overview of test procedure used to measure energy usage.

Table 1: Empirical values determined for t_{com}, t_{rec}, and t_{dec} for varying levels of compression C and N.

C	N	t_{com}	t_{rec}	t_{dec}
0	Wi-Fi	0.000000	0.068900	0.000000
0	3G	0.000000	0.068900	0.000000
1	Wi-Fi	0.219298	0.068350	0.111936
1	3G	0.219298	0.068350	0.111936
2	Wi-Fi	0.175132	0.068343	0.103453
2	3G	0.175132	0.068343	0.103453
3	Wi-Fi	0.179364	0.068336	0.105835
3	Wi-Fi	0.179364	0.068336	0.105835
4	3G	0.200769	0.068334	0.104855
4	Wi-Fi	0.200769	0.068334	0.104855
5	3G	0.201235	0.068327	0.114478
5	Wi-Fi	0.201235	0.068327	0.114478
6	3G	0.210243	0.068324	0.113710
6	Wi-Fi	0.210243	0.068324	0.113710
7	3G	0.214303	0.068322	0.122977
7	Wi-Fi	0.214303	0.068322	0.122977
8	3G	0.207665	0.068321	0.120345
8	Wi-Fi	0.207665	0.068321	0.120345
9	3G	0.200630	0.068321	0.123269
9	Wi-Fi	0.200630	0.068321	0.123269

Given that we enumerated all possible combinations of C and N, each of the rows in Table 1 provides a partial solution to the optimization. However, for each partial solution, the remaining parameters B and H must also be determined. Each solution is an instance of the objective Equation 1, which consists of a constrained nonlinear multivariable function. Matlab's `fmincon` program was used to minimize the value of the objective function across the range of possible values for B and H.

5 RELATED WORK

File synchronization, file compression and mobile computing are prevalent areas of research that are increasingly overlapping in problem domain space because smartphones, with limited storage capacity, are being used as data storage for documents, songs and other file types. We categorize the related work of this problem domain into two types: (1) distributed file systems, and (2) file synchronization algorithms.

Distributed File Systems. Cloud distributed file systems offer an alternative to peer-to-peer networks (Aymerich et al., 2008; Popovic and Hocenski, 2010; Uppoor et al., 2010). Recent research on the Cloud and smartphones show that file updating for software-as-a-service applications is an efficient way to synchronize software on all devices that house the program (Cusumano, 2010). Our work is complementary to this approach, in that our approach can be used to optimize the parameters of distributed file system synchronization algorithms that can be fit into our formal model in Equation 2.

Data Synchronization Algorithms. As smartphones move in location, the devices will experience fluctuations in network signal strength (Fukushima et al., 2005). Fukushima et al. discuss an algorithm for scheduling automatic file synchronization based upon the received signal strength indication of the network. Specifically, smaller files synchronize in areas of lower bandwidth and vice versa for larger files. Our approach is complementary to this work in that it provides an alternate method of minimizing energy consumption. We also compare total energy consumption over Wi-Fi and 3G, with results found in Section 4.

Yan et al. present a low-latency algorithm that first compartmentalizes files into fixed-byte blocks, then hashes each block for fast equality comparison when synchronization is done (Yan et al., 2008). The low-latency algorithm is very similar to our approach, except for the fact that Yan et al. focused on minimizing network traffic while we worked to reduce energy consumption.

6 CONCLUSIONS

As mobile devices have increased in popularity, it is relatively common for users to have multiple computing devices, such as a desktop computer, a laptop, and a smartphone, creating the need for distributed file synchronization. The existing approaches for building distributed file synchronization algorithms were developed for desktop platforms, where battery capacity is not a concern. Due to the high importance of energy consumption on mobile devices, there is a need for file synchronization algorithms that attempt to minimize total energy consumption.

This paper presented a method of deriving the distributed file synchronization algorithm parameters that would result in the lowest overall energy usage. In order to do this, Section 3 derived a formal model for energy usage of a distributed file synchronization algorithm. This formal model was then used in Section 4 to derive the most energy-efficient distributed file synchronization algorithm parameter values. The approaches presented in this paper can easily be extended to cover other specific distributed file synchronization algorithms.

From our research on this topic, we learned the following important lessons:

1. **Network Choice is the Single Most Important Decision to when Attempting to Minimize Energy Usage.** Using a 3G network results in a 10x increase in both time and energy over a Wi-Fi network.
2. **Ratio of File Compression is the Second-most Important Design Consideration when Attempting to Minimize Energy Usage.** It is clearly worth examining file extensions to see if the files are likely to experience no benefits from being minimally compressed (such as synchronization of files already stored in a compressed format) and avoiding compression for those files.
3. **Dynamic Block Sizing Should be Investigated.** The optimum block size is different for 3G and Wi-Fi. This encourages research into storing multiple versions of the block-hash table and using the version most appropriate for the context of the mobile device. As mobile storage increases, this approach of sacrificing data to gain energy could provide a notable improvement over the current traditional methodology of consistently using one block size.

REFERENCES

Agarral, S., Starobinski, D., and Trachtenberg, A. (2002). On the scalability of data synchronization protocols for pdas and mobile devices. *IEEE Network*, pages 22–28.

Aymerich, F., Fenu, G., and Surcis, S. (2008). An approach to a cloud computing network. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, pages 113 –118.

Balasubramanian, N., Balasubramanian, A., and Venkataramani, A. (2009). Energy Consumption in Mobile

Phones: A Measurement Study and Implications for Network Applications. In *ICM '09*. ACM.

Cusumano, M. (2010). Cloud computing and saas as new computing platforms. *Commun. ACM*, 53:27–29.

Fukushima, T., Takahashi, E., Narazaki, H., and Tamura, N. (2005). A wireless agent for autonomous file synchronization in a mobile environment.

Gartner (2010). Gartner says worldwide mobile phone sales grew 35 percent in third quarter 2010; smartphone sales increased 96 percent. http://www.gartner.com/it/page.jsp?id=1466313.

Lagerspetz, E., Tarkoma, S., and Lindholm, T. (2010). Dessy: Search and synchronization on the move. *Mobile Data Management*, pages 215–217.

Mudge, T. (2001). Power: A first-class architectural design constraint. *Computer*, 34(4):52–58.

Mummert, L. B., Ebling, M. R., and Satyanarayanan, M. (1999). Exploiting weak connectivity for mobile file access. *Mobility*, pages 306–321.

Popovic, K. and Hocenski, Z. (2010). Cloud computing security issues and challenges. In *MIPRO, 2010 Proceedings of the 33rd International Convention*, pages 344 –349.

Shye, A., Scholbrock, B., Memik, G., and Dinda, P. (2010). Characterizing and modeling user activity on smartphones: summary. *Proceedings of the ACM SIGMETRICS conference on Measurement and Modeling of Computer Systems*.

Uppoor, S., Flouris, M. D., and Bilas, A. (2010). Cloud-based Synchronization of Distributed File System Hierarchies.

Yan, H., Irmak, U., and Suel, T. (2008). Algorithms for Low-Latency Remote File Synchronization. In *IEEE INFOCOM 2008*. IEEE.