

Energy Efficiency of Lossless Data Compression on a Mobile Device: An Experimental Evaluation

Armen Dzhagaryan and Aleksandar Milenković
Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Huntsville, AL, U.S.A.
{aad0002, milenka}@uah.edu

Martin Burtscher
Department of Computer Science
Texas State University-San Marcos
San Marcos, TX, U.S.A.
burtscher@txstate.edu

Abstract— Lossless compression and decompression are routinely used in mobile computing devices to reduce the costs of communicating and storing data. This paper presents the results of an experimental evaluation of common compression utilities on Pandaboard, a development platform similar to current commercial mobile devices. We study the compression ratio, compression and decompression throughput, and energy efficiency of different usage scenarios typical for mobile computing. We observe a wide variety of energy costs associated with data compression and provide practical guidelines for selecting the most energy-efficient configurations.

Keywords—mobile computing; measurement techniques; data compression; energy-aware systems

I. INTRODUCTION

Mobile computing devices such as smartphones, tablet computers, and e-readers are becoming the dominant platforms for consuming digital information. The number of devices shipped was estimated to be 700 million smartphones and 106 million tablets in 2012, and it is forecast to reach 1.4 billion and 326 million in 2015, respectively. Global mobile data traffic continues to grow exponentially and reached 885 petabytes per month at the end of 2012.

Minimizing storage capacity and energy costs of data communication is of great interest for mobile devices. Energy efficiency is a prime design requirement for manufacturers and smart application developers alike. It is driven by several key factors, including (i) limited energy capacity of batteries, (ii) cost considerations favoring less expensive packaging, and (iii) user convenience favoring lightweight designs with small form factors that operate for long periods without recharging.

Data compression utilities are critical in helping achieve energy-efficient data communication, reducing communication latencies, and making effective use of available storage. The general goal of data compression is to reduce the number of bits needed to represent information. Whereas lossy compression approximates the original data and is acceptable for audio and video files, lossless compression enables the exact reconstruction of the original data by the decompressor. We focus on lossless compression, which is crucial for data such as program code, text input, medical information, and email.

The choice of compression algorithm, the compression lev-

el, and the quality of the implementation affect the energy consumption and can be a decisive factor in battery-powered devices. Achieving a higher compression ratio requires more computation and therefore more energy, but better compression reduces the number of bytes, thus saving energy when transmitting the data. Hence, it is important to take a close look at the energy efficiency of lossless compression algorithms on a state-of-the-art mobile platform. In particular, we want to answer several questions not addressed in a prior study [1] and determine whether compression is useful for reducing energy consumption, which common compression algorithms should be used, what configurations result in the best energy efficiency, and whether parallel execution can save energy.

We perform a comparative study of the most recent versions of several popular compression utilities, including gzip, lzop, bzip2, xz, pigz (a parallel implementation of gzip) and pbzip2 (a parallel implementation of bzip2) on Pandaboard, a state-of-the-art mobile development platform running a Linaro distribution of Ubuntu 12.04. For each utility, we analyze the effectiveness of all supported compression levels. We examine several performance metrics, including the compression ratio and the compression and decompression throughputs. Using our experimental setup for energy measurements, we also study the amount of energy consumed by the compression and decompression tasks and report the energy efficiency. We evaluate the compression utilities in three typical use scenarios. *Local* involves compression and decompression tasks performed locally on Pandaboard. *Wired* and *Wireless* involve compression tasks that stream data to and from a remote server over a secure communication channel using an Ethernet and a wireless LAN interface, respectively.

II. EXPERIMENTAL METHODOLOGY

A. Measuring setup

We developed the following setup for measuring the energy expenditure during program execution on Pandaboard. Pandaboard is connected to a power supply ($V_{SP} = 5$ V) via a low-resistance shunt resistor ($R = 0.1$ Ω). The voltage over the shunt resistor ($V_{SHUNT} = R \times I$) is sampled using a data acquisition (DAQ) device connected to a development workstation. The current I can be calculated from the voltage drop over the shunt resistor as $I = V_{SHUNT}/R$. The workstation runs a custom mPowerProfile program that controls both Pandaboard (via a serial link terminal) and the DAQ (via a USB port). mPower-

Profile starts collecting voltage samples and, after a head delay, a Linux command is issued to Pandaboard. It collects samples during application execution as well as for a tail delay after the application has completed. mPowerProfile provides utilities for configuring the measurements and calculating the total energy consumed by a compression or decompression task (ET.C or ET.D). In addition, we also calculate the energy overhead of the compression/decompression task alone, which excludes the energy needed to run the platform when idle, thus decoupling our findings from Pandaboard, which draws a relatively high idle current ($I_{\text{idle}}=0.56$ A).

B. Metrics and Experiments

Table I shows the metrics that are used in the experimental evaluation. In Local experiments, the input file (comprised of text, code, image, health, and binary data) is read from tmpfs, a temporary Linux file system stored in main memory and the output files are redirected to the Linux null device (/dev/null) to eliminate the latencies due to read and write operations from the SD card. In Wired and Wireless experiments, the raw input file is read from tmpfs, compressed on Pandaboard, and streamed to a remote server over a secure channel. The output files are redirected to the null device of the remote server. For the decompression tasks, the compressed files are retrieved from the temporary file system of the remote server through a secure channel and decompressed on Pandaboard. In Wired and Wireless experiments we also evaluate the throughput and energy efficiency for uncompressed transfers (UUP – uncompressed upload, UDW – uncompressed download).

TABLE I. METRICS

Symbol	Description	Unit	Definition
US	Uncompressed file size	MB	Measured
CS	Compressed file size	MB	Measured
CR	Compression ratio	-	US/CS
T.C [T.D]	Time to [de]compress	s	Measured
ET.C [ET.D]	Total energy for [de]compression	J	Measured
ET.C(0) [ET.D(0)]	Overhead energy for [de]compression	J	ET.C – $I_{\text{idle}} \times V_{\text{SP}} \times T.C$ ET.D – $I_{\text{idle}} \times V_{\text{SP}} \times T.D$
Th.C [Th.D]	[De]Compression Throughput	MB/s	US/T.C US/T.D
EE.C [EE.D]	[De]Compression energy efficiency	MB/J	US/ET.C US/ET.D

III. RESULTS AND CONCLUSIONS

Fig. 1 shows the compression ratio and throughputs for uncompressed and compressed data transfers in the Wireless experiment. Table II lists the most energy efficient utilities and their compression levels both compression and decompression tasks as a function of the idle current (0, 0.25, and 0.5 A). Our main findings from the experimental analysis are as follows.

- The effectiveness varies widely across different utilities and compression levels, often spanning two orders of magnitude.
- For local compression and compression with upload over the wired network, lzop with compression levels -1 to -6 performs best in compression throughput and energy efficiency. The second best utility is pigz with low compression levels.

- For local decompression, lzop performs best.
- For decompression after download over the wired network, lzop and pigz perform best, regardless of the compression level that was used for generating the input files.
- For compression with upload over the wireless interface, pigz with a low compression level (-1) performs best.
- For decompression after download over the wireless network, xz with the highest compression levels (-4 to -6) achieves the best decompression throughput and energy efficiency.

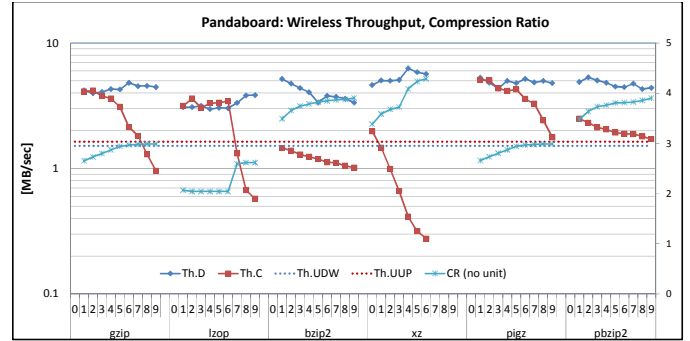


Fig. 1. Compression ratio and throughputs for compressed and uncompressed transfers in Wireless experiment

TABLE II. ENERGY-EFFICIENCY OF COMPRESSION/DECOMPRESSION UTILITIES

Experiment	Compression		Decompression	
	Best Utility	EE.C [MB/J]	Best Utility	EE.D [MB/J]
LOCAL				
$I_{\text{idle}} = 0$ A	lzop -1 to -6	55	lzop -6 to -9	137
$I_{\text{idle}} = 0.25$ A	lzop -1 to -6	14.5	lzop -6 to -9	40
$I_{\text{idle}} = 0.5$ A	lzop -1 to -6	8.5	lzop -1 to -9	23
WIRED				
$I_{\text{idle}} = 0$ A	lzop -1 to -6	12.4	lzop -7 to -9	23.5
$I_{\text{idle}} = 0.25$ A	lzop -1 to -6	5.1	pigz -6 to -9	10
$I_{\text{idle}} = 0.5$ A	lzop -1 to -6	3.2	pigz -6 to -9	6.5
WIRELESS				
$I_{\text{idle}} = 0$ A	pigz -1	2.5	pigz -4 to -7	4.6
$I_{\text{idle}} = 0.25$ A	pigz -1	1.5	xz -4 to -6	2.4
$I_{\text{idle}} = 0.5$ A	pigz -1	1.1	xz -4 to -6	1.6

Our findings may guide energy optimizations of data transfers in mobile applications and encourage the development of data transfer frameworks that are aware of the mobile device’s energy status. For example, a server could easily store multiple copies of the same file, compressed with different utilities and compression levels, to allow the mobile device to choose which version of a file to download based on its capabilities, currently available network bandwidth, energy status, and user preferences. Based on similar criteria, the mobile device could select which format to use for uploading a file, and the server could then convert the file to the best download formats.

REFERENCES

- [1] K. C. Barr and K. Asanović, “Energy-aware lossless data compression,” *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, Aug. 2006.