# Energy and Performance Evaluation of Lossless File Data Compression on Server Systems

Rachita Kothiyal, Vasily Tarasov,
Priya Sehgal, and Erez Zadok

***File systems and Storage Lab***
***Stony Brook University***
**http://www.fsl.cs.sunysb.edu**

NetApp



## Motivation

- Claim: Data compression is a way to save energy in computer systems [SNIA GSI, AERTC]
- Study the performance and energy implications of compression
- Summary: Compression not universally good
  - compression tool
  - hardware
  - input data type
  - read/write ratio of workload

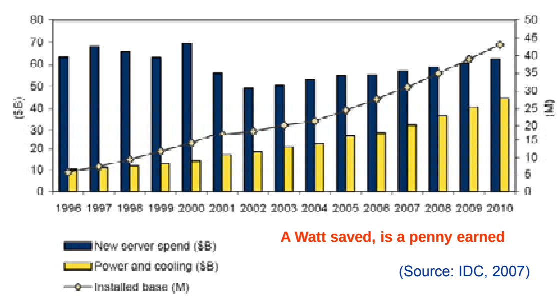


## Why Save Server Energy?

- Earlier focus was only on mobile devices.
- Servers worldwide have large energy requirements
  - 2% of national electricity use in the US [Cameron 2009]
- More power ➔ More cooling
  - Billions of dollars annually



## Worldwide Cost to Power and Cool Installed Servers




A Watt saved, is a penny earned

(Source: IDC, 2007)



## Overview

- Motivation
- **Related Work**
- Experimental Methodology
- Evaluation Results
- Conclusion and Future Work



## Related Work Categories

- Right-sizing:
  - Transition device to lower power state
    - CPU DVFS: voltage/frequency scaling
    - CPU Clock gating
    - Machine ACPI states: standby, hibernate, off, etc.
    - Disk spin-down, DRPM
- Work reduction:
  - Caching, aggregation, localization
  - Compression, DeDUP

## Techniques for Storage (1)

- Spindown disks on idle
  - Predict **when** to spindown
  - Predict **how long** to keep spun down
  - Techniques to increase idle time, to sleep for longer
    - Redirect future requests elsewhere
      - Popular Data Concentration (PDC) [Pinheiro 2004]
      - Massive Array of Idle disks (MAID) [Colarelli 2002]
      - Write off-loading [Narayanan 2008]



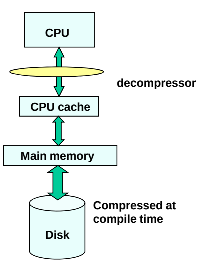
## Techniques for Storage (2)

- Dynamically change rotation speed of disk
  - DRPM: similar to DVFS
- Reduce disk head seeks
  - Replicate data blocks to improve locality
    - Predictive Data Grouping [Essary 2008]
    - FS2 [Huang 2005]
  - **Bonus**: Locality improves both energy and performance



## Code Compression

- [Benini 1999]
- Embedded CPUs
- Compress instructions
- Decompress before Execution
- Less bus activity
- Less storage used




We evaluate both compression & decompression costs



## Wireless Compression (1)

- Energy aware lossless data compression [Barr 2006]
  - Evaluated various compression tools on a **handheld**
  - Wireless transmission energy costlier than CPU
  - Compress data before transmitting over wireless
  - Found *compress* and *lzop* to be beneficial

Our focus is on server, desktop, and laptops



## Wireless Compression (2)

- Dynamic compression in multi-hop wireless networks [Sharma 2009]
  - Wireless sensor networks
  - Make transmission and compression decisions to minimize energy consumption
  - Maintain a static record of energy costs for each compression algorithm
  - Dynamically decide whether to compress data or not, and which compression algorithm to use

We include data with various levels of entropy



## Overview

- Motivation
- Related Work
- **Experimental Methodology**
- Evaluation Results
- Conclusion and Future Work

# Experimental Methodology

- Three dimensions of the evaluation:
  - Hardware type (testbed)
    - Server, desktop, and laptop machines
  - Compression algorithm
    - gzip, lzop, bzip, compress, and ppmd
    - Fastest and best compression levels included for each tool
  - Input data
    - 2GB sized files with varying data redundancy
      - Highly redundant file (zero)
      - Text file (text)
      - Binary file (binary)
      - Highly random file (random)



# Testbed

| | Server | Desktop | Laptop |
|---|---|---|---|
| **CPU Model** | Intel Xeon | Intel Pentium 4 | Intel Core Duo |
| **CPU Speed** | 2.8 GHz | 1.7 GHz | 1.6 GHz |
| **# of CPUs** | 2 dual core | 1 single core | 1 dual core |
| **DVFS** | No | No | Yes |
| **C states support** | No | No | Yes |
| **L1 cache size** | 16KB | 8KB | 16KB |
| **L2 cache size** | 2MB | 256KB | 2MB |
| **FSB speed** | 400 MHz | 400 MHz | 533 MHz |
| **RAM size** | 2048 MB | 1152 MB | 2560 MB |
| **RAM type** | DIMM | RIMM | SODIMM |
| **Disk RPM** | 10000 RPM | 7200 RPM | 5400 RPM |
| **Disk Transfer rate** | 320 Mbps | 133 Mbps | 100 Mbps |
| **Machine Age** | 3 years | 6 years | 2.5 years |
| **SPEC CPU2006** | 6.89 | 4.47 | 8.54 |
| **Average Idle Power** | 218 W | 91 W | 17 W |

All machines use only 1 CPU and 1GB RAM for the tests



# Benchmarks (1)

- plain-write:
  1. read an uncompressed file
  2. write uncompressed file to disk
- compress-write:
  1. read an uncompressed file
  2. compress
  3. write compressed file to disk

(plain-write and compress-write: Compare)

- plain-read:
  1. read uncompressed file from disk
- decompress-read:
  1. read compressed file from disk
  2. uncompress

(plain-read and decompress-read: Compare)



# Benchmarks (2)

```
for each machine in (server, desktop, laptop) {
    for each file in (text, binary, zero, random) {
        for each tool in (gzip, lzop, bzip, compress, ppmd) {
            run plain-write
            run compress-write
            run plain-read
            run decompress-read
        }
    }
}
```



# Benchmarking Setup

- Auto-pilot tool [Wright 2005]
  - Benchmarking suite
  - Run each test minimum 5 iterations, until half-widths were less than 5% of mean
    - 95% conf. interval (student-*t* distribution)
  - Evaluation Metrics:
    - Performance (T): elapsed, user, system, wait times
    - Energy (E) : auto-pilot plugin to obtain logged data from the energy measuring device
    - Energy-delay (ET): combined metric of E and T [Gonzalez 1996]



# Power Measurement Setup

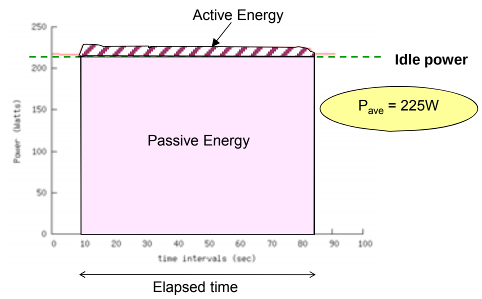
- Plug-in style wattmeter (Wattsup Pro ES)
- Inline between the wall AC power and the test machine
- Provides Volts, Amps, Watts, **energy** (watt-hours), etc.
- Accuracy: 1.5% ± 3 digits
- Resolution: 1 Watt-hour
- Sampling frequency: once per second
- Script to download data from the meter to the test machine

## Overview

- Motivation
- Related Work
- Experimental Methodology
- **Evaluation Results**
  - **Server results**
  - Desktop results
  - Laptop results
- Conclusion and Future Work



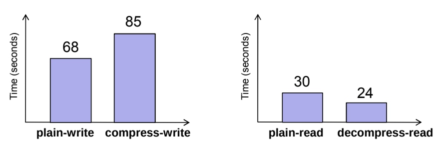
## Power Profile: plain-write






## Energy/Performance Savings Hypothesis

- compress-write generally more expensive than plain-write
- decompress-read cheaper than plain-read
- Multiple decompress-reads can compensate for the compress-write






## Read-Write Model

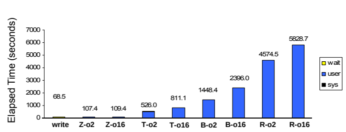
- Best case: both compress-write and decompress-read do better than plain-write/read
- Common case: compress-write loses to plain-write, while decompress-read wins over plain-read
- R/W ratios 4/1 [Roselli 2000], 2/1 [Leung 2008]
- Break-even value: minimum number of decompress-reads to compensate for extra cost of compress-write:

$$(M_c - M_w) \leq n \times (M_r - M_d)$$

$$n_{be} = (M_c - M_w) / (M_r - M_d)$$



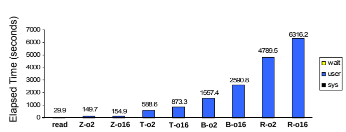
## PPMd Server Performance (all files)




plain-write vs. compress-write

- Best compression ratio
- Significant cost of time
- Similar operations/costs during compression and decompression

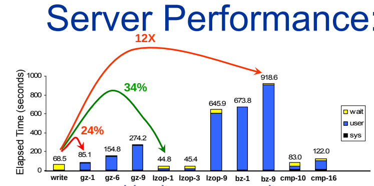
| | |
|---|---|
| all | × |




plain-read vs. decompress-read

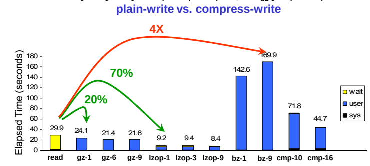


## Server Performance: Text File



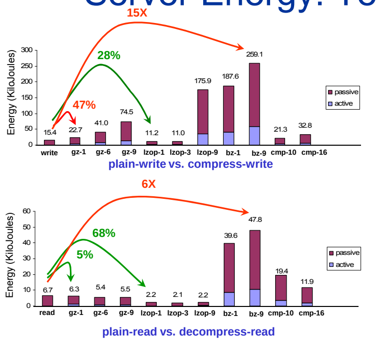

plain-write vs. compress-write

| | |
|---|---|
| lzop-1,3 | ∀ |
| bzip | × |
| gzip | n reads/writes |




plain-read vs. decompress-read

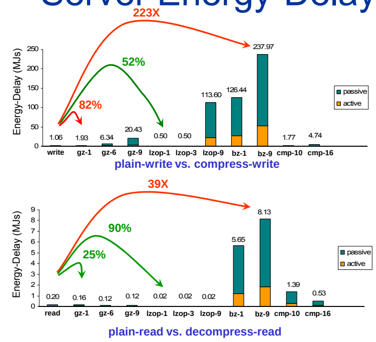
## Server Energy: Text File




| | |
|---|---|
| lzop-1,3 | ∀ |
| bzip | × |
| gzip | n reads/writes |



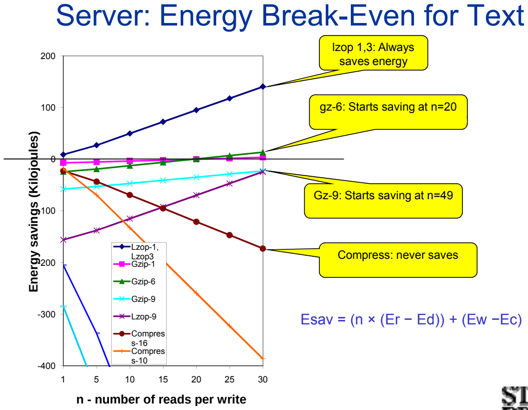
## Server Energy-Delay: Text File




| | |
|---|---|
| lzop-1,3 | ∀ |
| bzip | × |
| gzip | n reads/writes |



## Server: Energy Break-Even for Text




Esav = (n × (Er − Ed)) + (Ew −Ec)



## Server Break-Even Values (E, T)

| Tool | Text | Binary | Zero | Random |
|---|---|---|---|---|
| gz-1 | 20.2 / 2.8 | X | ∀ | X |
| gz-6 | 19.7 / 10.9 | X | ∀ | X |
| gz-9 | 49.0 / 24.9 | X | ∀ | X |
| lzop-1 | ∀ | 2.1 / 0.8 | ∀ | X |
| lzop-3 | ∀ | 2.1 / 1.0 | ∀ | X |
| lzop-9 | 35.4 / 26.8 | 172 / 130 | 5.4 / 3.6 | X |
| bzip-1 | X | X | 0.28 / ∀ | X |
| bzip-9 | X | X | 1.3 / 0.2 | X |
| c-10 | X | X | ∀ | X |
| c-16 | X | X | ∀ | X |

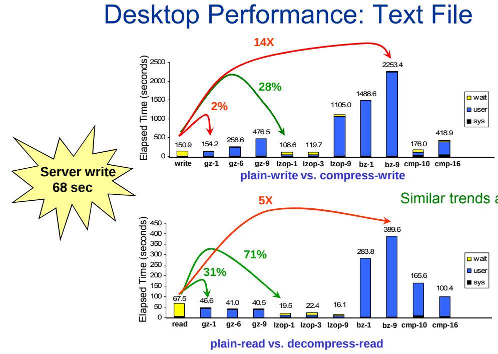


## Overview

- Motivation
- Related Work
- Experimental Methodology
- **Evaluation Results**
  - Server results
  - **Desktop results**
  - Laptop results
- Conclusion and Future Work



## Desktop Performance: Text File




Similar trends as on server

## Desktop Break-Even Values (E, T)

| Tool | Text | Binary | Zero | Random |
|---|---|---|---|---|
| gz-1 | X / 0.2 | X | ∀ | X |
| gz-6 | 33.2 / 4.1 | X | ∀ | X |
| gz-9 | 78.7 / 10.8 | X | ∀ | X |
| lzop-1 | ∀ | 3.8 / 1.0 | ∀ | X |
| lzop-3 | ∀ | 3.5 / 0.9 | ∀ | X |
| lzop-9 | 31.6 / 18.6 | 158 / 89 | 4.2 / 1.9 | X |
| bzip-1 | X | X | 1.3 / ∀ | X |
| bzip-9 | X | X | 1.7 / ∀ | X |
| c-10 | X | X | ∀ | X |
| c-16 | X | X | ∀ | X |

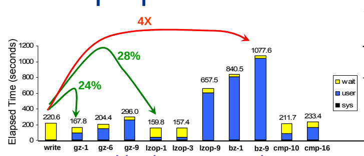


## Overview

- Motivation
- Related Work
- Experimental Methodology
- **Evaluation Results**
  - Server results
  - Desktop results
  - **Laptop results**
- Conclusion and Future Work



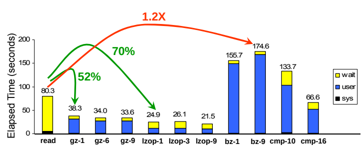
## Laptop Performance: Text File




plain-write vs. compress-write

Server write 68 sec

Desktop write 151 sec



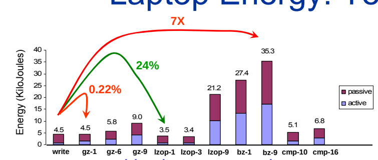

plain-read vs. decompress-read

| | |
|---|---|
| lzop-1,3 | ∀ |
| bzip | X |
| gzip-1,6 | ∀ |

Disk much slower than CPU

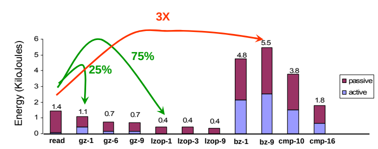


## Laptop Energy: Text File




plain-write vs. compress-write

| | |
|---|---|
| lzop-1,3 | ∀ |
| bzip | X |
| gzip | n reads/writes |




plain-read vs. decompress-read



## Energy and Time on Laptop

- Non linearity between E and T in plain-write vs. compress-write
- Power management
- plain-write:
  - I/O bound, CPU switched to lower frequency ➔ lower power state
- compress-write:
  - Exercises both CPU and disk
  - Higher average power



## Laptop Break-Even Values (E, T)

| Tool | Text | Binary | Zero | Random |
|---|---|---|---|---|
| gz-1 | 0.03 / ∀ | X / 0.7 | ∀ | X |
| gz-6 | 1.86 / ∀ | X / 7.2 | ∀ | X |
| gz-9 | 6.3 / 1.6 | X / 38.3 | ∀ | X |
| lzop-1 | ∀ | 0.75 / ∀ | ∀ | X |
| lzop-3 | ∀ | 0.86 / ∀ | ∀ | X |
| lzop-9 | 15.5 / 7.4 | 84.2 / 34.6 | 0.29 / ∀ | X |
| bzip-1 | X | X | ∀ | X |
| bzip-9 | X | X | ∀ | X |
| c-10 | X / 0.2 | X | ∀ | X |
| c-16 | X / 0.9 | X | ∀ | X |

## Break-Even Text Comparison (E, T)

| Tool | Server | Desktop | Laptop |
| --- | --- | --- | --- |
| gz-1 | 20.2 / 2.8 | X / 0.2 | 0.03 / ∀ |
| gz-6 | 19.7 / 10.9 | 33.2 / 4.1 | 1.86 / ∀ |
| gz-9 | 49.0 / 24.9 | 78.7 / 10.8 | 6.3 / 1.6 |
| lzop-1 | ∀ | ∀ | ∀ |
| lzop-3 | ∀ | ∀ | ∀ |
| lzop-9 | 35.4 / 26.8 | 31.6 / 18.6 | 15.5 / 7.4 |
| bzip-1 | X | X | X |
| bzip-9 | X | X | X |
| c-10 | X | X | X / 0.2 |
| c-16 | X | X | X / 0.9 |



## Overview

- Motivation
- Related Work
- Experimental Methodology
- Evaluation Results
- **Conclusion and Future Work**



## Conclusions

- Compression *can* benefit energy and performance
  - Energy/performance best case 10–40%
  - Worse cases 10–100x+
- Varied results based on file types
  - Compression never helps for random files
  - Always helps for zero files
- Different compression tools behave differently
  - Faster tools tend to do better (e.g., *lzop*)
- Generally, similar trends across the 3 machines
- Hardware compression devices need to be more intelligent



## Future Work

- Expand study to different:
  - workloads
  - file system types, formats, and options
  - kernel options (e.g., I/O scheduler)
  - cluster/storage configs: DAS, NAS, distrib.
- Custom file systems and I/O schedulers
- Cluster auto-configuration for workload
  - Calibrate to specific hardware (compression, etc.)



## Energy and Performance Evaluation of Lossless File Data Compression on Server Systems

Rachita Kothiyal, Vasily Tarasov, Priya Sehgal, and Erez Zadok

# Q&A

*File systems and Storage Lab*
Stony Brook University
http://www.fsl.cs.sunysb.edu