

speeding up ext3 with usb flash drives

Robbie Hott
Advanced Operating Systems
December 2, 2005

Motivation

- Gain speedup over standard ext3
 - Reduce seeks

For Linux's ext3, the journal is typically stored at the beginning of the disk, with normal data to follow. This is, in theory, just as fast as Soft Updates and Logging Systems, however switching back and forth to the journal and then to the data can sometimes require many seeks, which can be avoided by using a flash drive.

- Keep integrity of ext3

Using non-volatile RAM, which is essentially what a flash drive is, since it doesn't require seeks, should be faster than using a hard drive, plus, since the data is on a drive and not the system RAM, if the system crashes the journal will still be in tact.

- Cheap Increase

Flash drives are extremely cheap today, most can be bought at 128MB for \$30 or less, and they should be much faster than using traditional hard disks. So, having this cheap medium between Main Memory and disk to store the journal is a cheap upgrade. Even if flash drives have limited lifespan, \$30 every year or so is still a cheap upgrade.

System Setup

- 2.26GHz P4 Processor
- 512MB RAM
- 2 40GB IDE Hard Drives
 - hda: operating system
 - hdb: two 19GB test storage spaces
- SuSE Linux 9.3
- Kernel 2.6.11.4-20a-default
- No swap

Test Drives

- ext3 using the following journaling devices
 - 128MB low end USB2.0 Flash Drive
 - 512MB Kingston USB2.0 Flash Drive
 - 128MB RAM drive(upper bound for flash journal)
- ext3 using standard ordered journal setup

Test Suites

- Bonnie

Bonnie is a simple data read/write test that writes data character by character, then does random reads and writes as well as reads by chunk. This test is expected to be slower on the flash drive because data is not reused much.

- IO Zone

IOZone is a system test that tests performance with multiple block and chunk sizes as well as testing each chunk size with multiple file sizes. It is more comprehensive than Bonnie because it tests multiple reads (rereads), multiple writes (rewrites), as well as random and in-order reads and writes.

- Custom C Benchmark

The custom coded benchmark I wrote in C was a test of multiple reads/writes to the same 128MB of data. This test wrote 128MB of data, character by character to disk, then called fsync to ensure that it is on disk. The system then read the chunk as one call to read() and wrote as one call to write(). The reading and writing were in a loop to test multiple consecutive executions.

Results

- Bonnie

The Bonnie benchmark, as expected, performs much worse for all systems with a flash drive, versus the setups that do not have a flash drive.

- IO Zone

The IOZone benchmark has some unexpected results. At some points, the systems with the flash drives are faster, but this is mainly in the cases of small files. This could be useful, but it is highly outweighed by similar or slower performance at higher file sizes, since files today are growing larger and larger.

- Custom C Benchmark

The custom benchmark had the most surprising results of all. The flash drive used in this test, the 512MB drive, is much larger than the data used in this experiment, however, using a data size of 128MB had a significant slowdown (3x) using the flash drive versus not using the drive. Even using the RAM drive is slower, although only .3 sec slower than the normal ext3 drive.

Conclusions

- Is it really worth it?

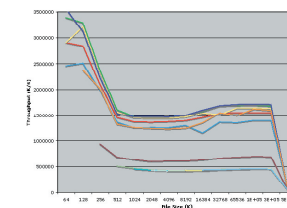
Sadly, after many trials, and with today's memory, where 1GB of RAM is fairly cheap, it is not worth it. The performance using the flash drive is slower, except in cases of really small file size, less than a few MB, and mainly for Re-Reading as seen on the right. But, the speed attained there does not outweigh the slowdown from all other cases; and with multimedia and streaming files on the horizon, only touching data once, and growing file sizes, this system is even less feasible.

| Machine | Sequential Output | | | | | | Sequential Input | | | | | | Random | |
|-------------------|-------------------|-------|---------|----------|-------|-------|------------------|-------|-------|-------|-----|--------|--------|--|
| | Per Char | Block | Rewrite | Per Char | Block | Seeks | Per Char | Block | Seeks | %CPU | | | | |
| 512MB Flash Drive | 420 | 17619 | 51.7 | 23176 | 6.7 | 3920 | 1.2 | 21639 | 60.3 | 32679 | 3.5 | 6066.2 | 11.8 | |
| Hard Disk Drive | 420 | 32948 | 96.8 | 64802 | 20.6 | 20006 | 5.6 | 21407 | 58.4 | 54513 | 5.2 | 1475.2 | 2.4 | |

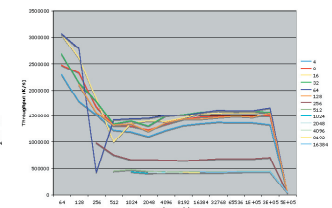
Chart: Bonnie results on each of the drives. This could explain why using the flash drive is so much slower.

| Machine | Sequential Output | | | | | | Sequential Input | | | | | | Random | |
|---------------------------------------|-------------------|-------|---------|----------|-------|-------|------------------|-------|-------|-------|-----|-------|--------|--|
| | Per Char | Block | Rewrite | Per Char | Block | Seeks | Per Char | Block | Seeks | %CPU | | | | |
| Ordered Journal with No Flash | 2000 | 30837 | 90.1 | 35760 | 12.0 | 15702 | 4.6 | 20508 | 56.2 | 41432 | 3.9 | 115.8 | 0.4 | |
| Journal with 128MB Flash | 2000 | 5554 | 16.4 | 6833 | 2.7 | 5719 | 2.6 | 19542 | 53.2 | 35734 | 3.3 | 119.8 | 0.4 | |
| Journal with 512MB Kingston Flash | 2000 | 10659 | 31.9 | 13813 | 5.4 | 9891 | 3.0 | 19197 | 52.4 | 36165 | 3.5 | 116.4 | 0.5 | |
| Ordered Journal with No Flash or Swap | 2000 | 30827 | 90.2 | 36088 | 12.3 | 16036 | 4.7 | 20629 | 56.4 | 40590 | 3.7 | 119.0 | 0.4 | |

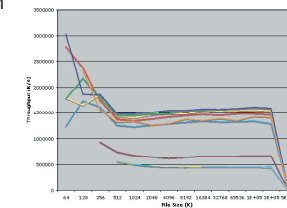
Chart: Bonnie results for the different journaling modes, including one with no swap partition used.



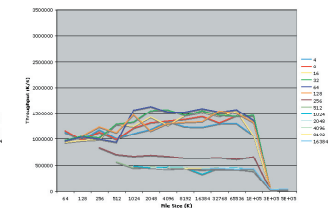
Graph: IOZone Re-Reader Report for 128MB RAM Drive as Journal.



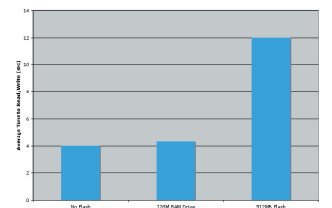
Graph: IOZone Re-Reader Report for 512MB Kingston Flash as Journal.



Graph: IOZone Re-Reader Report for 128MB Flash as Journal.



Graph: IOZone Re-Reader Report for Normal Ordered-Journal ext3.



Graph: Average time to read and write a chunk of data (128MB) without a flash drive, with the 128MB RAM drive and the 512MB Kingston flash drive.