Performance Comparison of NTFS and ext4

Akshay Uttamani, Vaibhav Patel
Department of Computer Sciences, University of Wisconsin-Madison

Introduction

- The performance of disk I/O intensive applications depends on the underlying file system
- Benchmarking thus becomes important to choose the best file system for different application requirements
- Two of the most popular file systems used today: NTFS and ext4
- We compared the I/O performance of these two file systems under different workloads

Conclusion

- Compilation benchmarking experiment shows that compilation in ext4 is about 1.5 times faster than in NTFS
- While benchmarking web server, we found that file transfer rate in ext4 is about 8 times faster than in NTFS
- ext4 performs better in most of our microbenchmark experiments
- File system performance is broadly dependent on two type of factors: data storage related and memory related

Discussion and Future Work

- A good benchmark should include in-memory, disk layout, cache warmup/eviction, and metadata operations performance evaluation components
- On NTFS if file is small enough, it can be stored in MFT record itself, further experiments can be performed to determine if NTFS outperforms ext4 for these file sizes
- We used Cygwin library in Windows so as to get a POSIX environment that adds an overhead
- We plan to perform these experiments on workloads that are platform independent

Macrobenchmarks

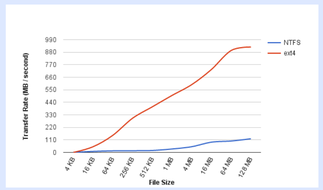
- These exercise multiple file system operations
- Good for an overall view of the system's performance
- Macrobenchmark results are explained using microbenchmark results

gcc compilation

Command	Elapsed Time (Minutes)	
	NTFS	ext4
configure	0.5	0.3
make install	0.7	0.5
make	250.6	171.5

- gcc version: 4.9.4
- Build process requires sequential read, metadata creation and sequential write operations
- 'make' command execution 1.5x faster in ext4 than NTFS
- ext4 performs better than NTFS when reading and writing files sequentially

Web-server Performance

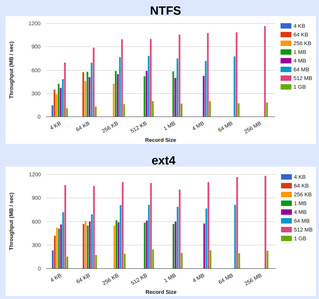


- Apache web-server version: 2.4
- Number of concurrent requests: 1
- ext4 performs significantly better than NTFS
- Transfer rate increases exponentially with file size in ext4

Microbenchmarks

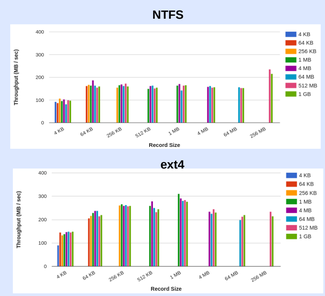
- Benchmarks designed for measuring the performance of a specific piece of code
- Useful for better understanding the results of a macrobenchmark
- These benchmarks are more meaningful when presented together with other benchmarks

Sequential Write



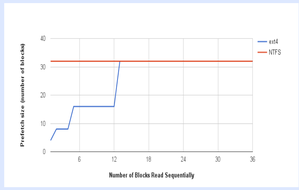
- Writes are first buffered to memory cache
- Throughput drops for 1GB file due to memory cache overflow

Sequential Read



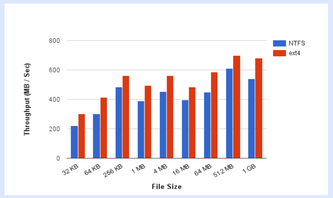
- Throughput drops by a large factor compared to sequential write
- At least the first read goes to disk
- NTFS and ext4 use the concept of prefetching to improve read throughput

Prefetch Size



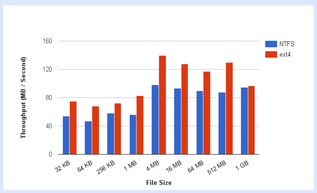
- NTFS prefetches 32 blocks on every read operation
- ext4 follows a dynamic prefetching scheme, prefetch size starts with 4 blocks reaching to a maximum of 32 blocks, growing in multiples of 2

Random Write



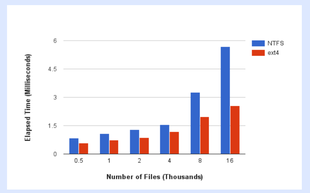
- Random writes are buffered in memory cache
- Throughput is of the same order as sequential write
- Fixed length records (8KB) were used

Random Read



- Significant drop in throughput compared to sequential read
- Prefetching not possible because of random block access
- Used FILE_FLAG_NO_BUFFERING and O_DIRECT flags while opening files in Windows and Linux respectively

Metadata Update Operation



- Updating permissions of each file in a directory
- NTFS stores file metadata in MFT and ext4 in inodes
- Data locality in ext4: all inodes in a directory are placed in the same block group as the directory