Reduce Performance Impact of Compaction Process by Designing a Global Format for LSM

Jinghuan YU
City University of Hong Kong

Chun Jason. Xue
City University of Hong Kong

Abstract

Your abstract text goes here. Just a few facts. Whet our appetites. Not more than 200 words, if possible, and preferably closer to 150.

1 Introduction

2 Background

This section provides some background on LSM (Log Structured Merged Tree) and NVM (Non-Volatile Memory). It first provides a high level overview of the LSM technology which is designed for better sequential write operations; Then, it describes some feature Non-Volatile Memory; Finally, it provides several challenges and advantages when combining NVM with LSM.

2.1 Log-Structured-Merged Tree

LSM Tree is a high warm writing performance data structure proposed in 1996 [8], gets widely used in many products like Hbase [2], Cassandra [1], BigTable [5] and WiredTiger [3]. The most important characteristics LSM has are list as following:

Sequential Write Optimized LSM's basic idea is converting random writes to sequential writes. Most of the storage devices has the characteristics that can perform much better in sequential operations than in random access operations, no matter read or write. LSM use a small amount of memory to buffer write operations.

Periodic Garbage Collection For using log-structure to organize persistent data, which may be outdated due to deletions and updates. This garbage collection process is known as "Compaction", which may cause very severe performance impact and resource occupying. This problem are studied from many years ago, bLSM [9] proposed a "Gear Scheduler" to dispersion pressure caused by compaction. This inspired a lot of following works studied on this, like Monkey [6], DostoevsKey [7], GearDB [11].

2.1.1

3 Footnotes, Verbatim, and Citations

Footnotes should be places after punctuation characters, without any spaces between said characters and footnotes, like so.[1] And some embedded literal code may look as follows.

```
int main(int argc, char *argv[])
{
    return 0;
}
```

Now we're going to cite somebody. Watch for the cite tag. Here it comes. Arpachi-Dusseau and Arpachi-Dusseau coauthored an excellent OS book, which is also really funny [4], and Waldspurger got into the SIGOPS hall-of-fame due to his seminal paper about resource management in the ESX hypervisor [10].

The tilde character (~) in the tex source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

And the 'cite' package sorts your citations by their numerical order of the corresponding references at the end of the paper, ridding you from the need to notice that, e.g, "Waldspurger" appears after "Arpachi-Dusseau" when sorting references alphabetically [4, 10].

It'd be nice and thoughtful of you to include a suitable link in each and every bibtex entry that you use in your submission, to allow reviewers (and other readers) to easily get to the cited work, as is done in all entries found in the References section of this document.

[1] Remember that USENIX format stopped using endnotes and is now using regular footnotes.

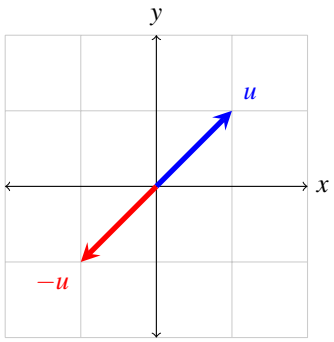

Figure 1: Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text.

Now we're going take a look at Section 4, but not before observing that refs to sections and citations and such are colored and clickable in the PDF because of the packages we've included.

4 Floating Figures and Lists

Here's a typical reference to a floating figure: Figure 1. Floats should usually be placed where latex wants then. Figure1 is centered, and has a caption that instructs you to make sure that the size of the text within the figures that you use is as big as (or bigger than) the size of the text in the caption of the figures. Please do. Really.

In our case, we've explicitly drawn the figure inlined in latex, to allow this tex file to cleanly compile. But usually, your figures will reside in some file.pdf, and you'd include them in your document with, say, \includegraphics.

Lists are sometimes quite handy. If you want to itemize things, feel free:

fread a function that reads from a `stream` into the array `ptr` at most `nobj` objects of size `size`, returning returns the number of objects read.

Fred a person's name, e.g., there once was a dude named Fred who separated usenix.sty from this file to allow for easy inclusion.

The noindent at the start of this paragraph in its tex version makes it clear that it's a continuation of the preceding paragraph, as opposed to a new paragraph in its own right.

4.1 LaTeX-ing Your TeX File

People often use `pdflatex` these days for creating pdf-s from tex files via the shell. And `bibtex`, of course. Works for us.

Acknowledgments

The USENIX latex style is old and very tired, which is why there's no \acks command for you to use when acknowledging. Sorry.

Availability

USENIX program committees give extra points to submissions that are backed by artifacts that are publicly available. If you made your code or data available, it's worth mentioning this fact in a dedicated section.

References

[1] Apache cassandra. http://cassandra.apache.org/. (Accessed on 03/07/2019).

[2] Apache hbase – apache hbase™ home. https://hbase.apache.org/. (Accessed on 03/07/2019).

[3] Wiredtiger: making big data roar. http://www.wiredtiger.com/. (Accessed on 03/07/2019).

[4] Remzi H. Arpaci-Dusseau and Arpaci-Dusseau Andrea C. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, LLC, 1.00 edition, 2015. http://pages.cs.wisc.edu/~remzi/OSTEP/.

[5] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):4, 2008.

[6] Niv Dayan, Manos Athanassoulis, and Stratos Idreos. Monkey: Optimal navigable key-value store. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 79–94. ACM, 2017.

[7] Niv Dayan and Stratos Idreos. Dostoevsky: Better space-time trade-offs for lsm-tree based key-value stores via adaptive removal of superfluous merging. In *Proceedings of the 2018 International Conference on Management of Data*, pages 505–520. ACM, 2018.

[8] Patrick O'Neil, Edward Cheng, Dieter Gawlick, and Elizabeth O'Neil. The log-structured merge-tree (lsm-tree). *Acta Informatica*, 33(4):351–385, 1996.

[9] Russell Sears and Raghu Ramakrishnan. blsm: a general purpose log structured merge tree. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 217–228. ACM, 2012.

[10] Carl A. Waldspurger. Memory resource management in VMware ESX server. In *USENIX Symposium on Operating System Design and Implementation (OSDI)*, pages 181–194, 2002. https://www.usenix.org/legacy/event/osdi02/tech/waldspurger/waldspurger.pdf.

[11] Ting Yao, Jiguang Wan, Ping Huang, Yiwen Zhang, Zhiwen Liu, Changsheng Xie, and Xubin He. Geardb: A gc-free key-value store on hm-smr drives with gear compaction. In *17th {USENIX} Conference on File and Storage Technologies ({FAST} 19)*, pages 159–171, 2019.