# Exploring Performance of Memory Mapping

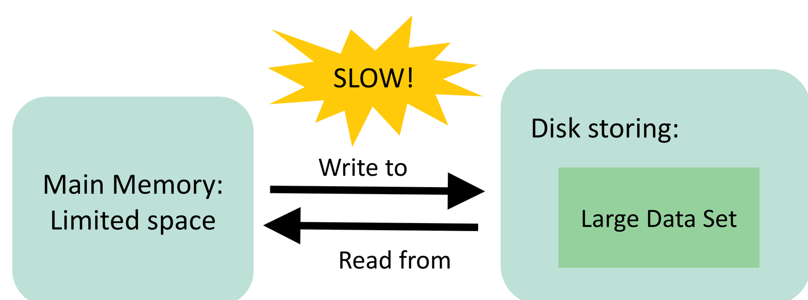
Bonnie Dai, Darya Verzhbinsky, Clair Li, Shayan Raisi
*Advisors:* Jishen Zhao, Xiao Liu

UC San Diego JACOBS SCHOOL OF ENGINEERING · UCSD CSE Computer Science and Engineering · ERSP UC San Diego Early Research Scholars Program · NSF · STABLE

## Background

Traditional data-processing applications have become inadequate to dealt with big-data. With the increase in dataset sizes, data processing performance has drastically decreased.

Current Memory Model:
Inefficient for large data set due to too many copies and pastes.




## Problem Statement

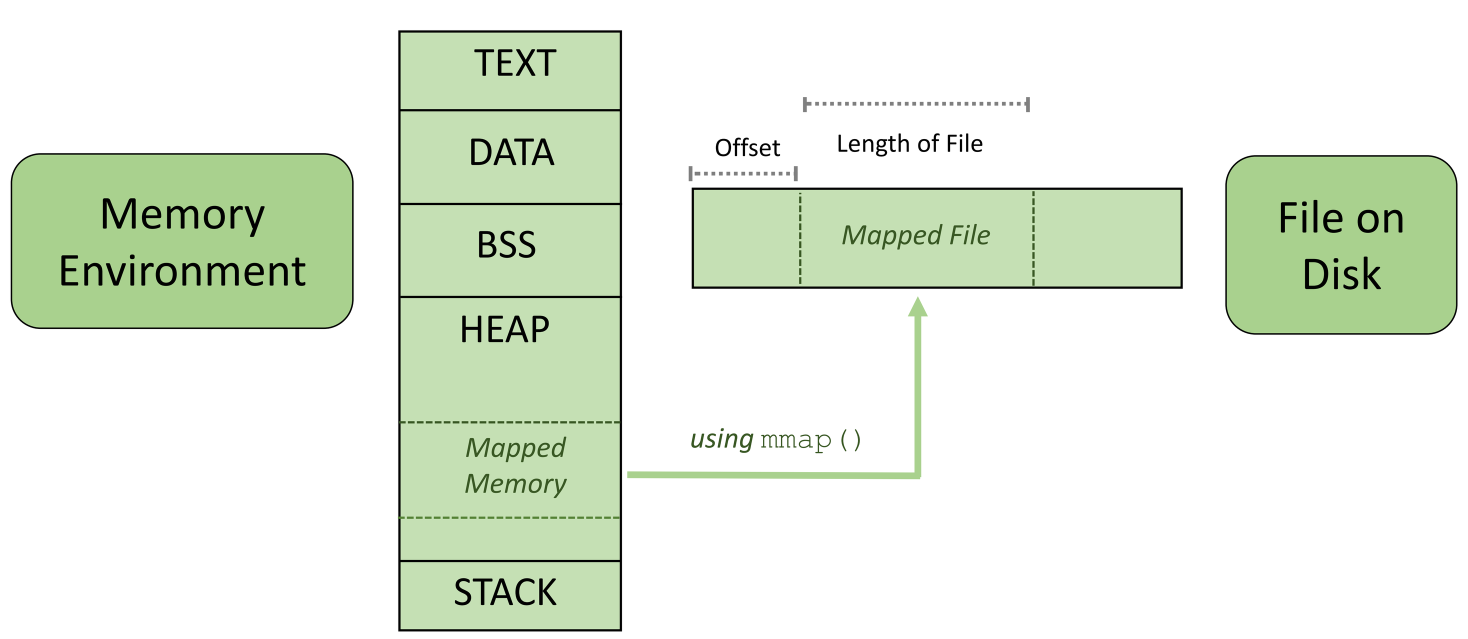
Though `mmap()` is an improvement, there is still room to grow. However, before implementing changes on the current memory mapping mechanism, we first need to better understand it. In order to identify the bottlenecks of `mmap()`, we conducted a series of tests across different Linux kernel versions. Through comparing the performance of the two kernels, we seek to determine which specific aspects of the system are negatively affecting performance.

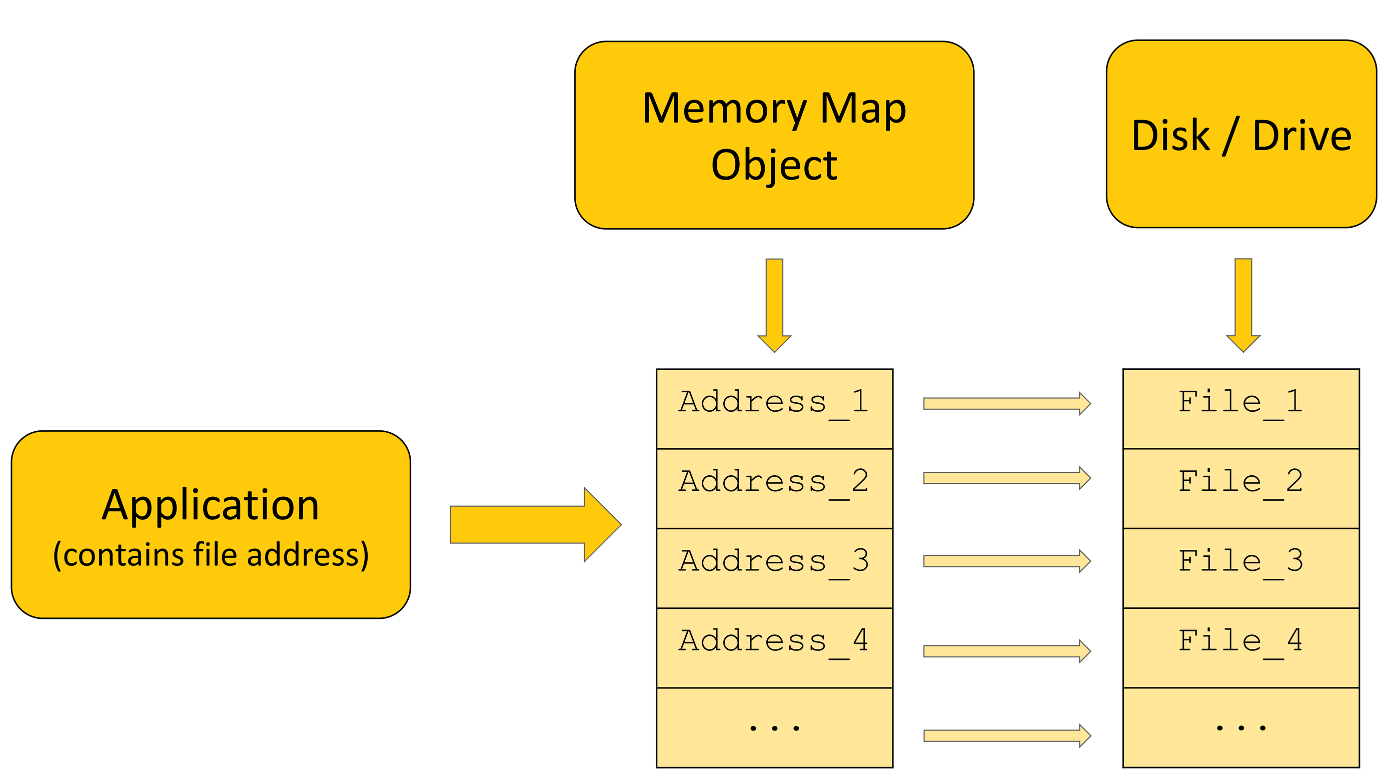
What we used:
`LevelDB` - database containing key-value pairs that includes in-house benchmark implementations.

## Memory Mapping

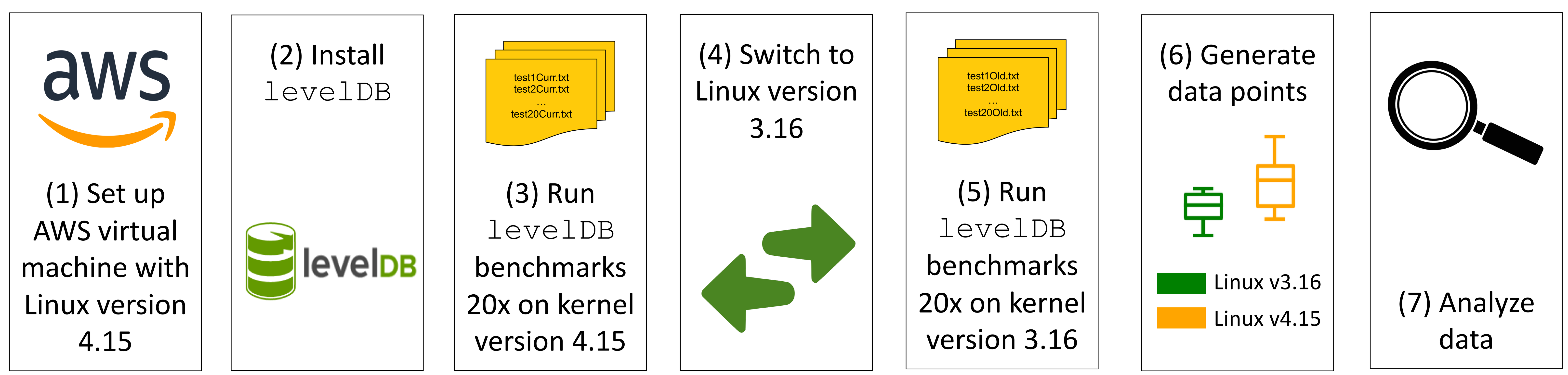
One solution: `Memory Mapping`

Memory mapping allows data on the disk to be accessed like dynamic memory. Instead of searching through the disk for the desired data, copying it into memory, and manipulating it, the system can simply access and change the data directly on the disk. This technique avoids the costly, time consuming disk I/O operations and thus accelerates the overall performance.

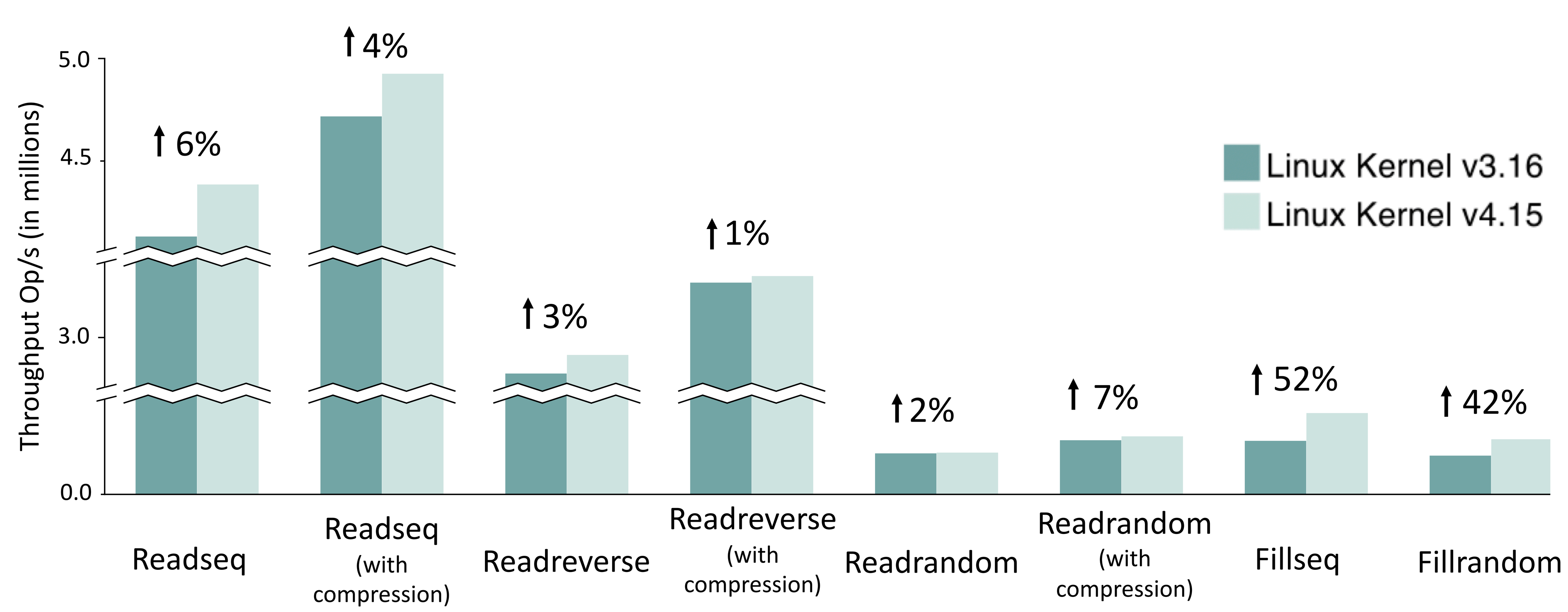



## Project Workflow




## Analysis




Overall, kernel version 4.15 performed better and more consistent than kernel version 3.16. This is most likely related to memory management and file systems updates. For example, in the update for kernel version 4.11, the `per-cpu` allocator was altered, improving performance on allocation/deallocation by roughly 30%.

## Future Work

- Run more benchmarks that mimic real world accessing patterns.
- Implement changes to the current kernel version to verify whether or not the analysis we made was accurate through new rounds of testing on that modified kernel.

## Acknowledgements

We would like to thank Dr. Christine Alvarado, Vignesh Gokul, Dr. Jishen Zhao, and Xiao Liu for all of their guidance and support, and the ERSP program for this great opportunity.