Mahanaxar: Quality of Service Guarantees in High-Bandwidth, Real-Time Streaming Data Storage

David Bigelow, Scott Brandt, John Bent, HB Chen

Systems Research Laboratory
University of California, Santa Cruz

Los Alamos National Laboratory

26th IEEE (MSST2010) Symposium on
Massive Storage Systems and Technologies
May 3-7
Incline Village, NV

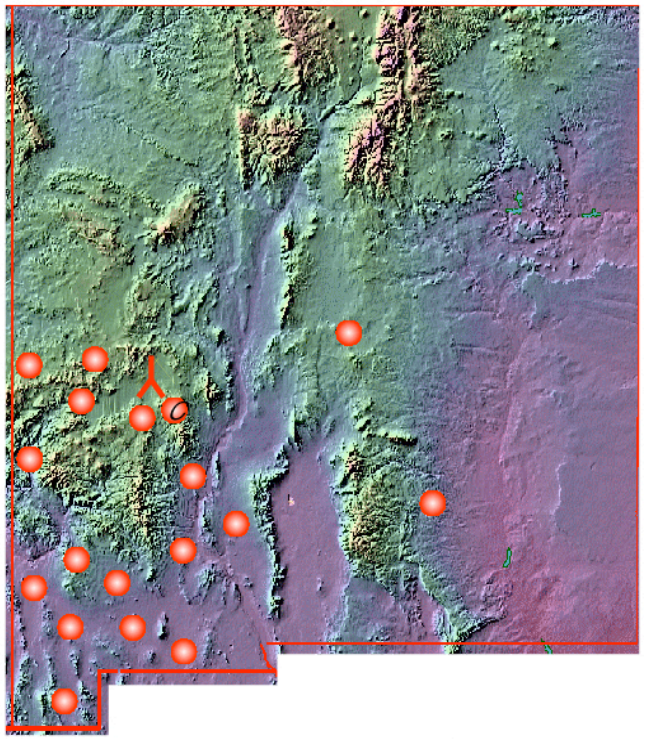
Overview

- **Problem:**
  - Certain applications need to capture and temporarily store “lots” of real time data
- **Example Applications:**
  - Astronomical observation
  - Network traffic capture
  - Trivially, TiVo
- **Our Solution: Mahanaxar**
  - A prototype system for high-speed data capture and management, with quality of service guarantees

Motivation: Long Wavelength Array

- **Low Frequency Radio Telescope**
- **Geographically distributed but synchronized**
- **Most collected data is “useless”**

- **Basic statistics:**
  - 53 stations (planned)
  - 72.5 MB/s data rate per station
  - ~3.75 GB/s data rate total



Right: Locations of LWA stations over southwestern New Mexico

Data Characteristics

- **Most data is worthless in the long run**
- **But sometimes the data is actually worthwhile**
  - ...and so were the last ten minutes of it, but we only found that out just now
- **There's too much data to keep long term**
  - LWA generates 1 PB of data in just over 3 days
- **The data is highly structured**

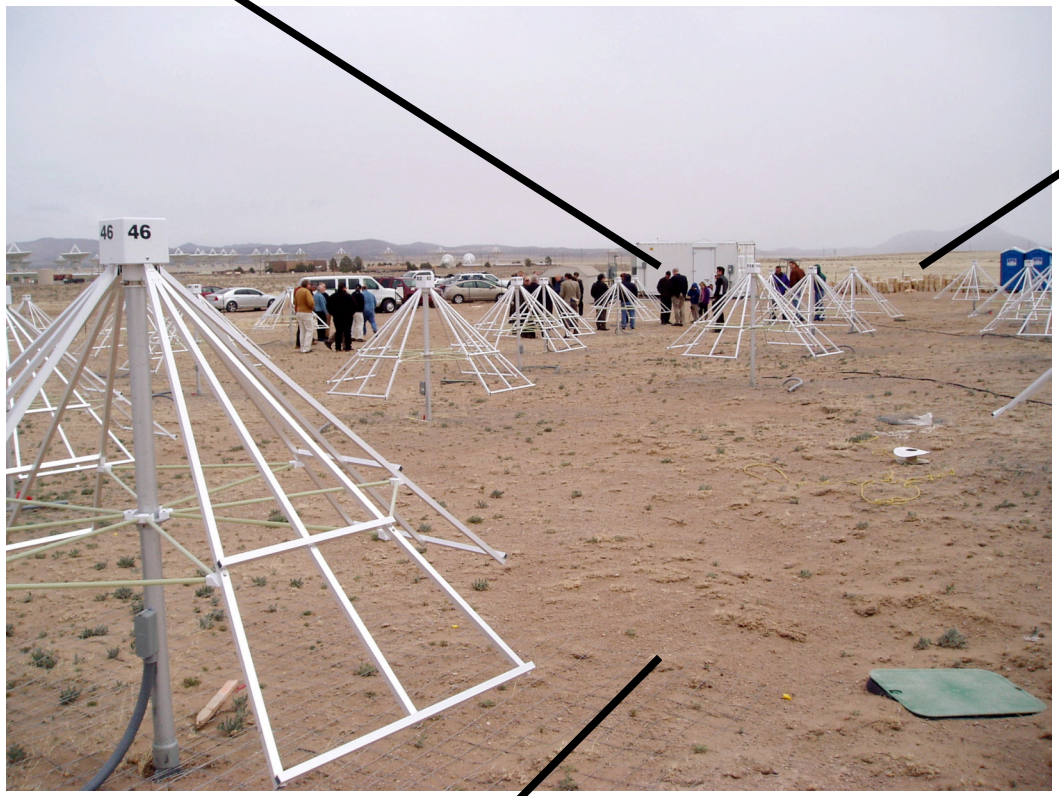
Basic Requirements

- **Quality of Service guarantees**
  - Incoming data *must* be captured on first (and only) transmission
  - Need to be able to read data off again
- **Never lose data**
  - Data cannot be regenerated
  - Reliability mechanisms cannot compromise QoS guarantees
- **Commodity components**
  - Avoid “throwing disks” at the problem
  - Required to work in non-ideal operating conditions

Potential Operating Environment

Example “machine room”

“Fat” network pipe may be unavailable



Desert Environment

Generalization

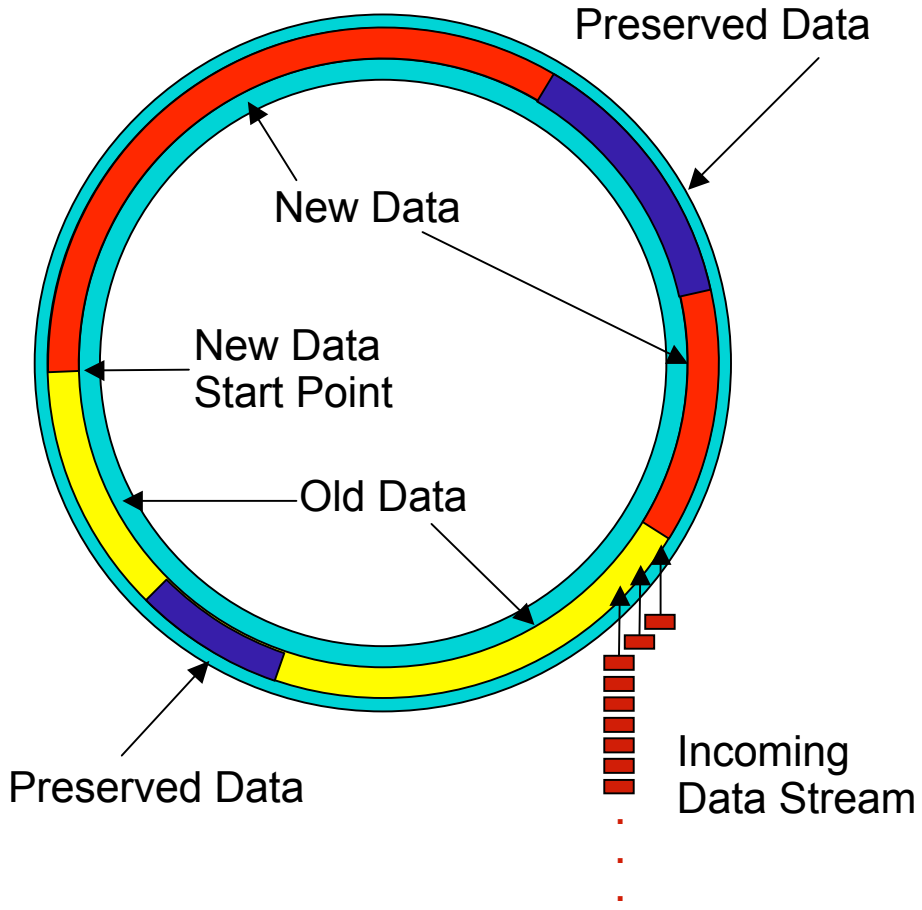
- **Must handle large and small data elements**
  - 60 MB chunk of binary data
  - 20 byte IP packets
- **Variable indexing complexity**
  - Simple sequence (time) indexing
  - Multiple indices for each (small?) element
- **Massive data rate**
  - GB/s in even a “small” system
- **Must manage data relationships**
  - Parallel data
  - Reliability scheme data relation

Observations

- **Many filesystem features useless**
  - No need for file creation, deletion, stat, etc.
  - Only one writing process, total
  - Very little filesystem based indexing or metadata
- **A system which never "shuts down"**
  - Does a file system structure need to be kept on-disk?
- **Large block operations are ideal**
  - Aggregate data into large blocks for maximal I/O performance
  - Minimize fragmentation
  - Minimize disk head movement

Our Solution: Ring Buffer

- **Fixed size**
  - Very little bookkeeping
- **Limited lifetime**
  - Automatic expiration of data
  - No data “cleanup”
- **Highly predictable**
- **Preservation in-place**
- **Limited indexing**

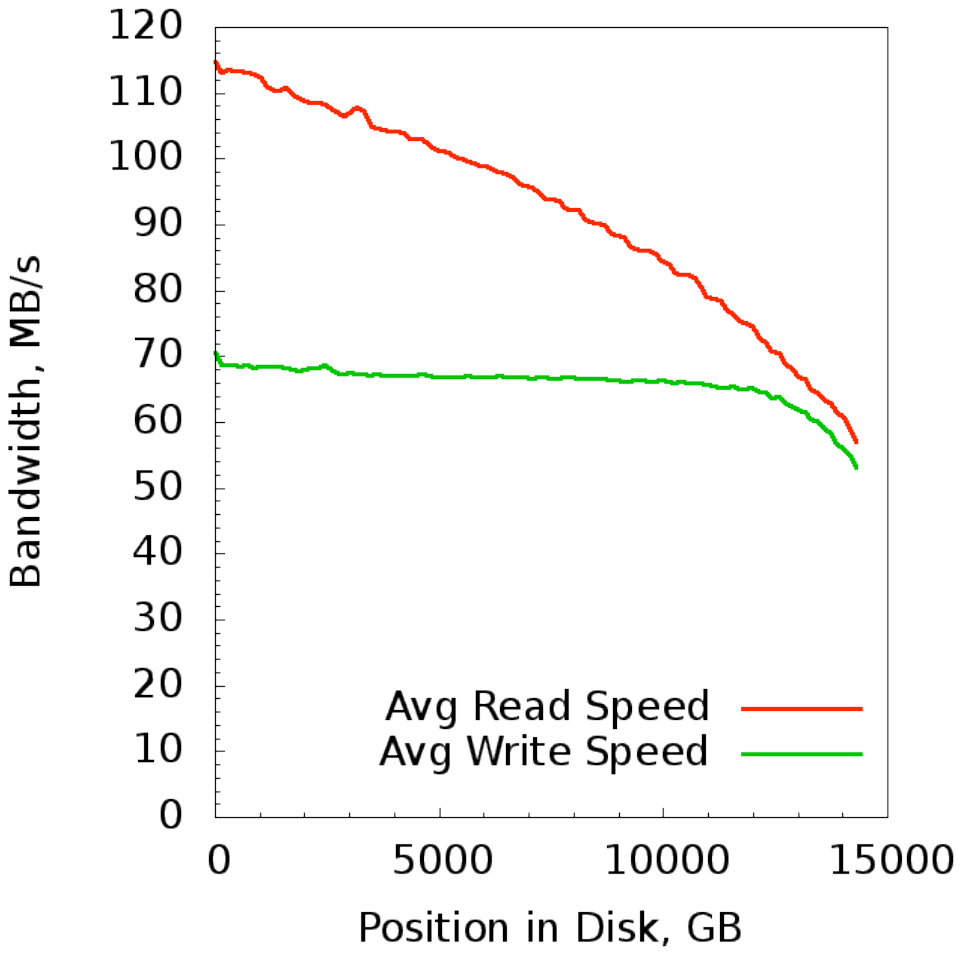


System Design

- **Stay close to the hardware for maximum performance**
  - Need to understand individual hard drives
- **Restrict data layout to large chunks**
  - Maximize performance by strictly controlling data placement
- **Maintain index in memory, not on-disk**
  - System never goes offline (barring errors)
- **Reliability and recovery mechanisms must not interfere with QoS guarantees**

Disk Profiling

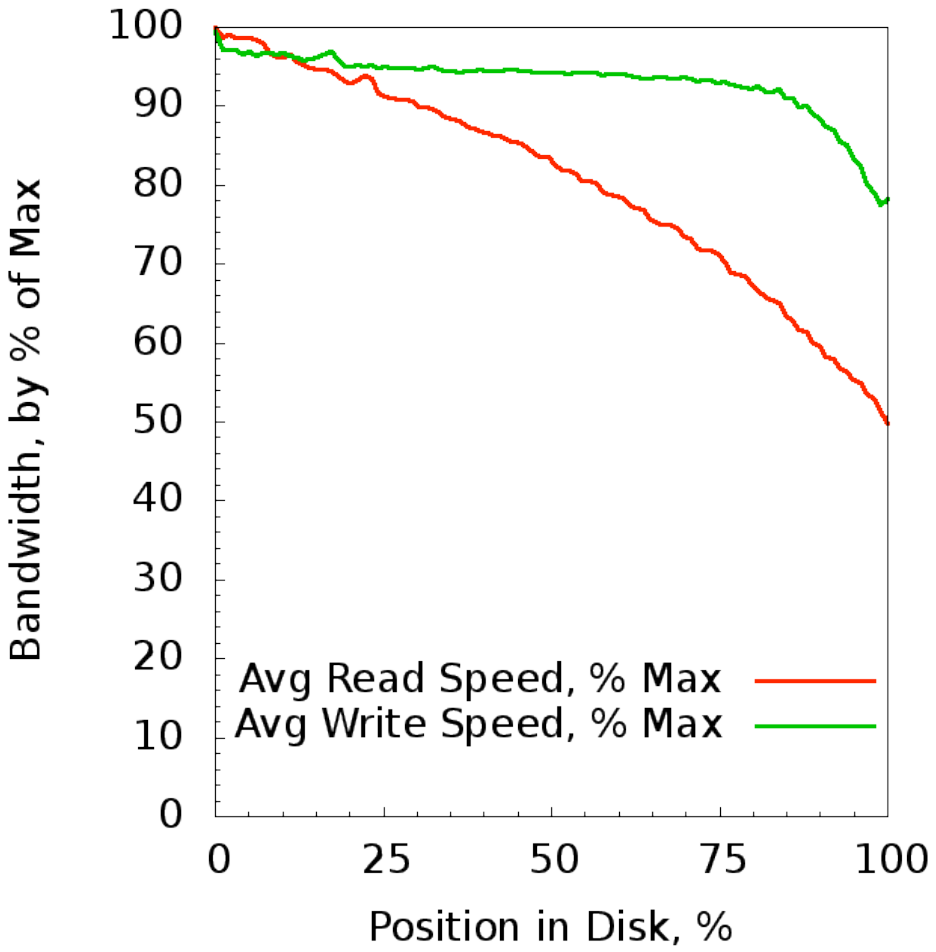
- Performance degrades over course of the disk
- Sharper performance degradation towards end of the disk
- May only want to use portions of the disk to maintain performance



Drive: Western Digital 1.5 TB Caviar Green, Model WD15EARS

Disk Profiling

- Performance degrades over course of the disk
- Sharper performance degradation towards end of the disk
- May only want to use portions of the disk to maintain performance

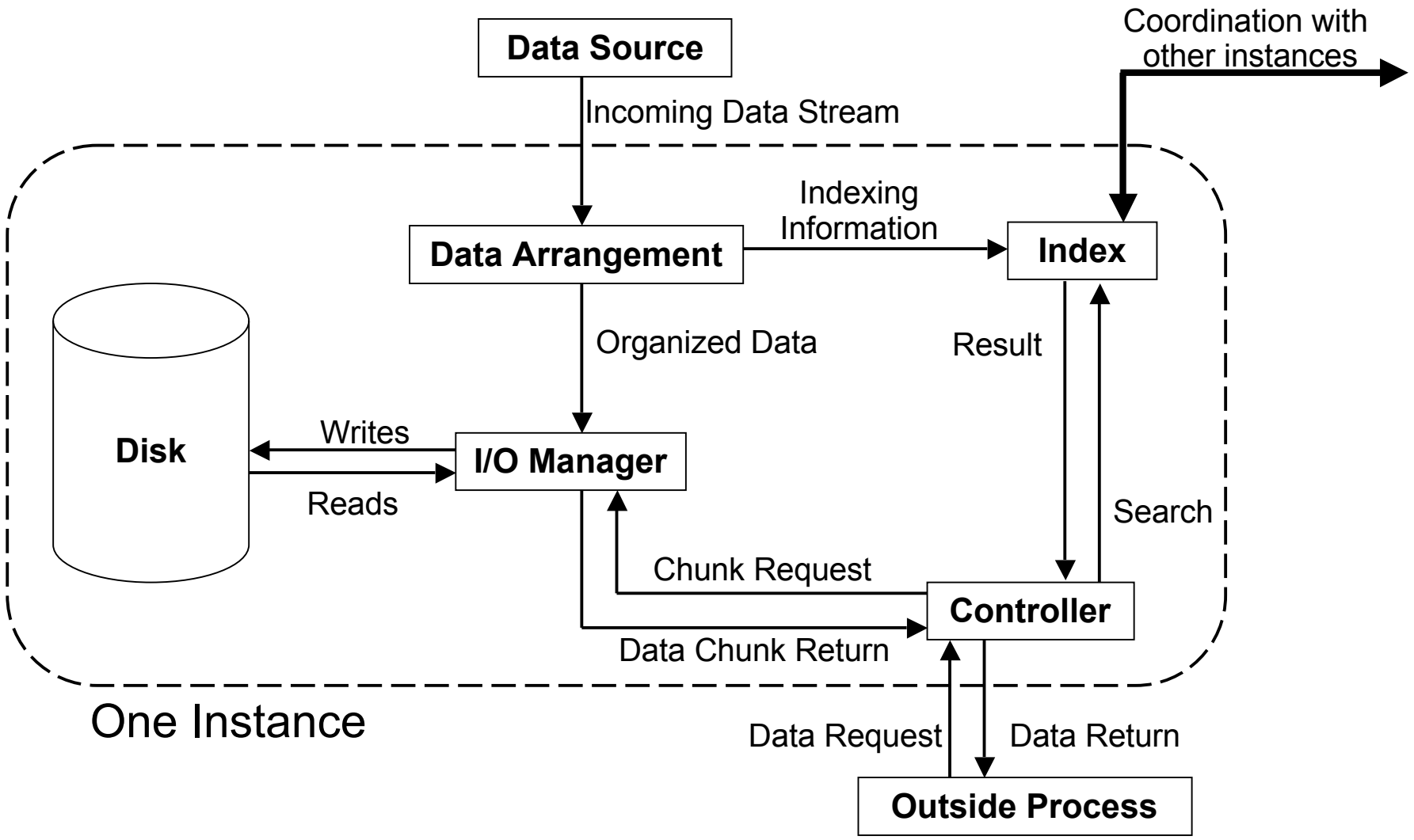


Drive: Western Digital 1.5 TB Caviar Green, Model WD15EARS

Prototype: Mahanaxar

- **Multithreaded userspace program**
  - Runs on single hard drives for big and small data
  - Can act in RAID-4 mode for reliability purposes
- **Can guarantee a minimum bandwidth for the write process (user specified)**
- **Automatically expires old data**
- **Customizable index for data search**
- **Preserves data in place when requested**

Architecture



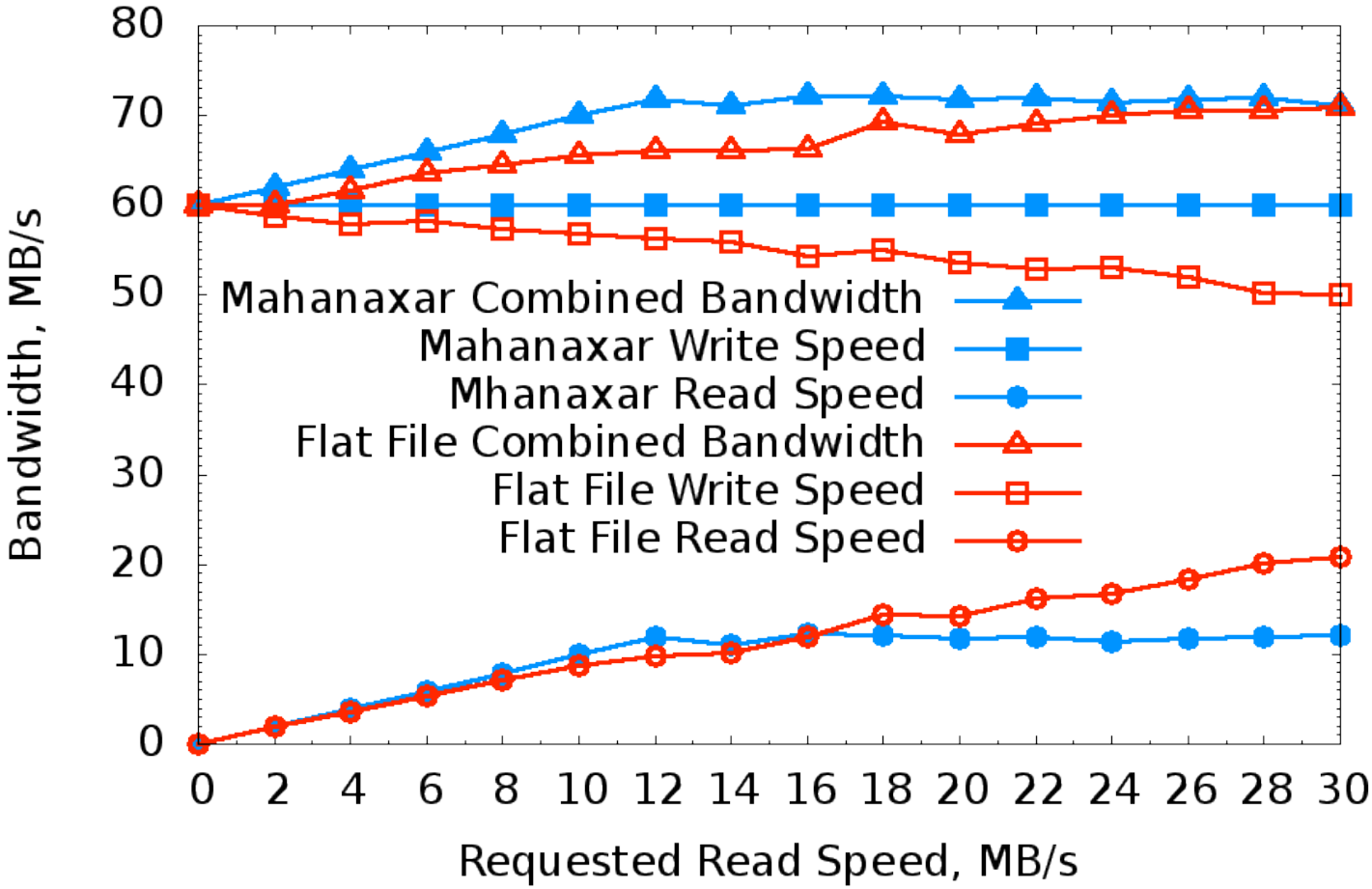
Testing Procedure

- **Primary comparison: flat file system (ext2)**
  - ext2 had best performance of all tested filesystems
- **Databases had poor performance**
  - Database performance collapses when the system is constantly at 99.9%+ capacity
- **Performance testing over multiple hard drives**
  - Results presented here are from one particular drive (the previously modeled one) in order to make the most accurate comparisons
  - Unless otherwise noted, results are from an “aged” system which has some segments preserved

Mahanaxar v. plain ext2

Element size: 60 MB

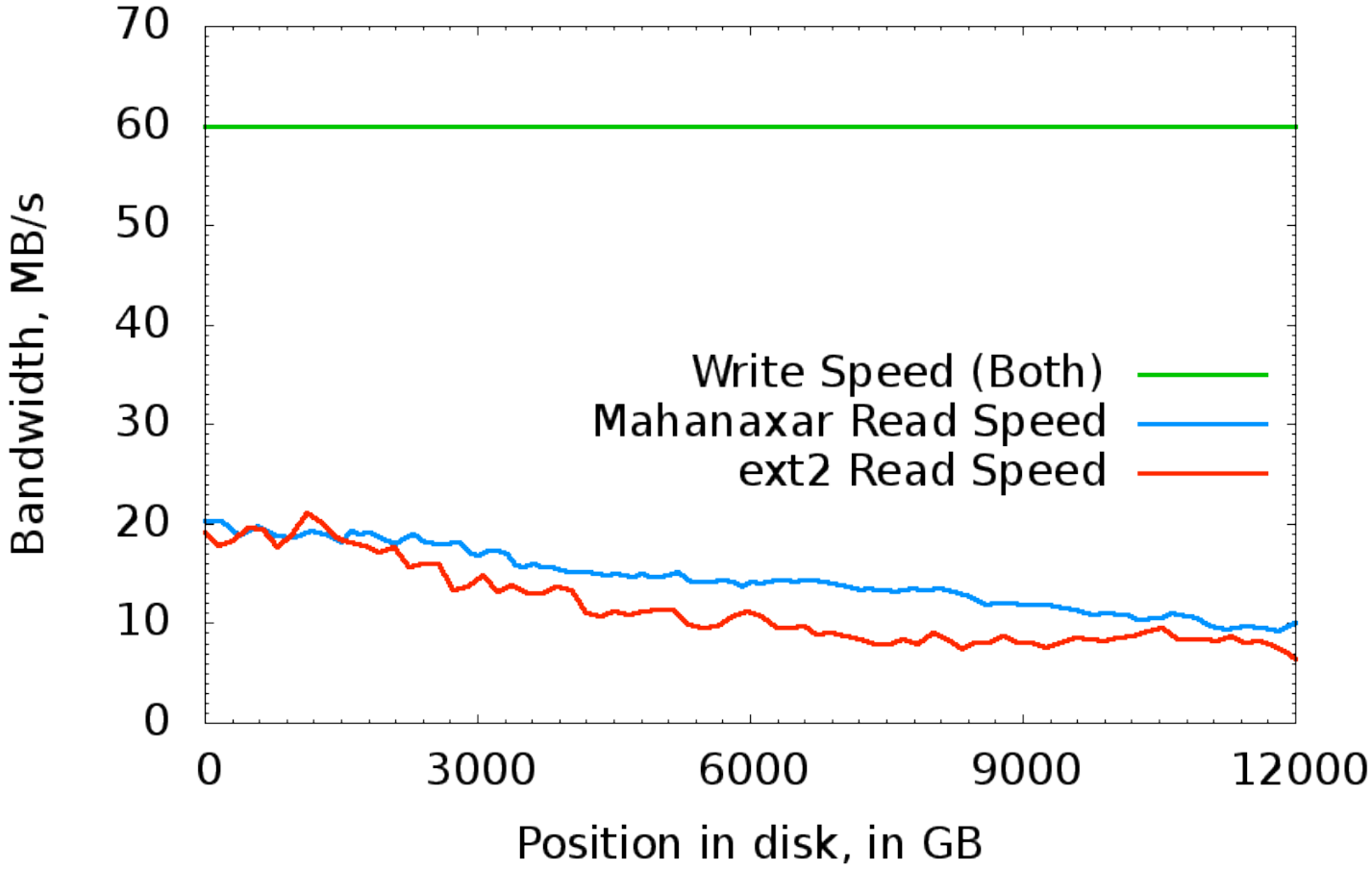
Requested write speed: 60 MB/s



Mahanaxar v. prioritized ext2 (first cycle)

Element size: 60 MB

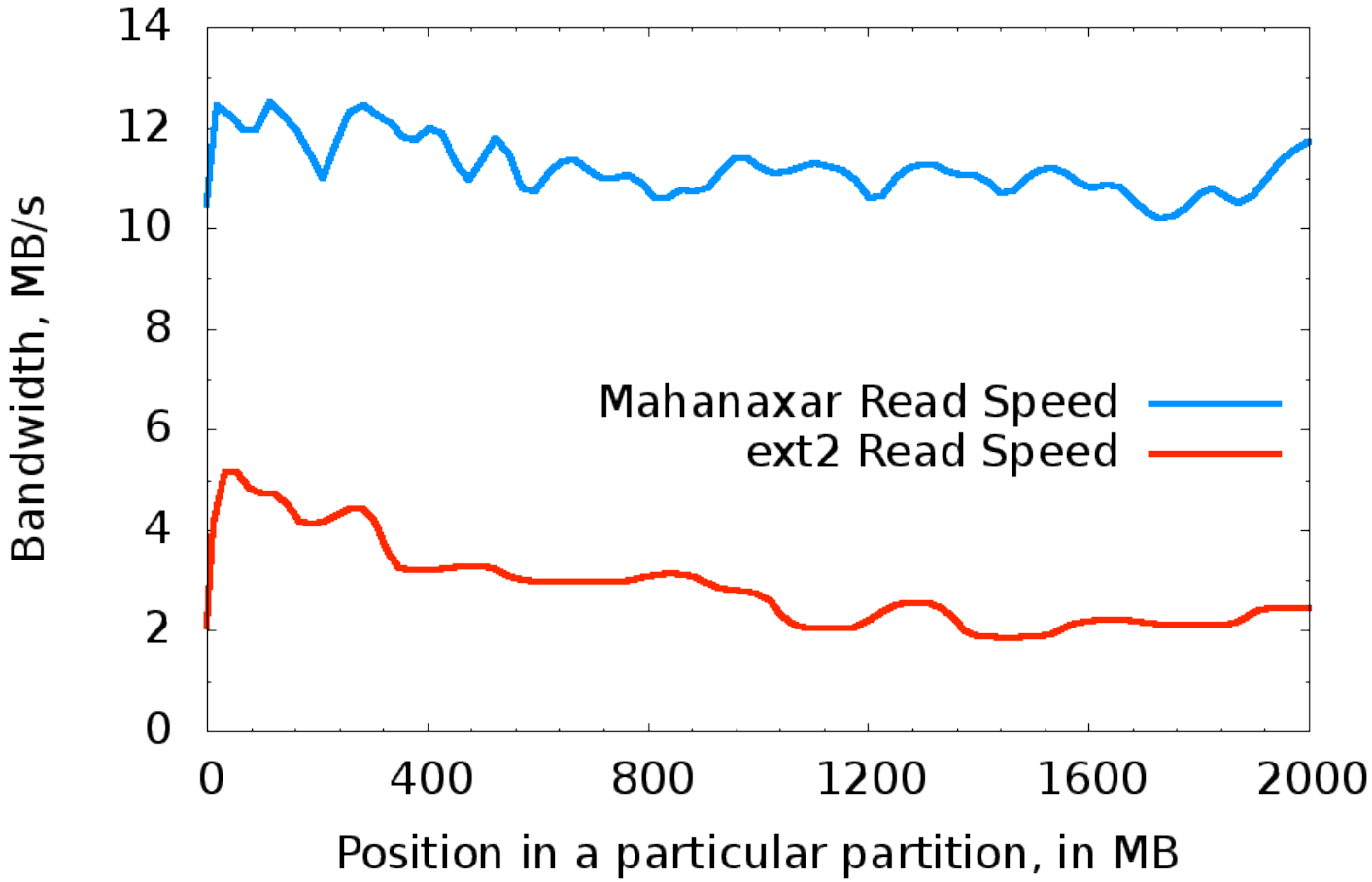
Requested write speed: 60 MB/s



Mahanaxar v. ext2, aged cycle (closeup)

Element size: 60 MB

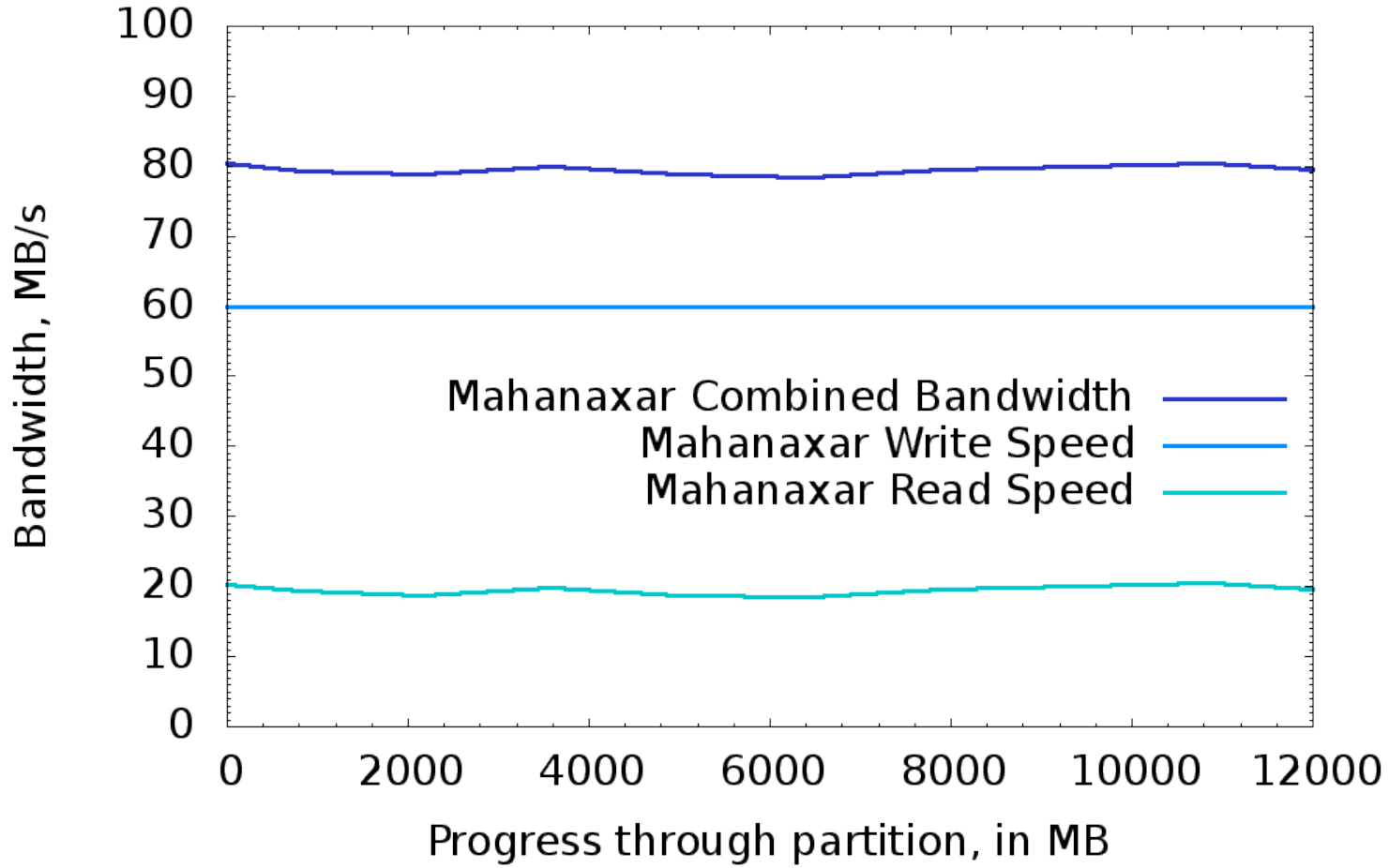
Write speed, both: 60 MB/s (not shown on graph)



Mahanaxar v. ext2 (small elements)

Element size: 1 MB

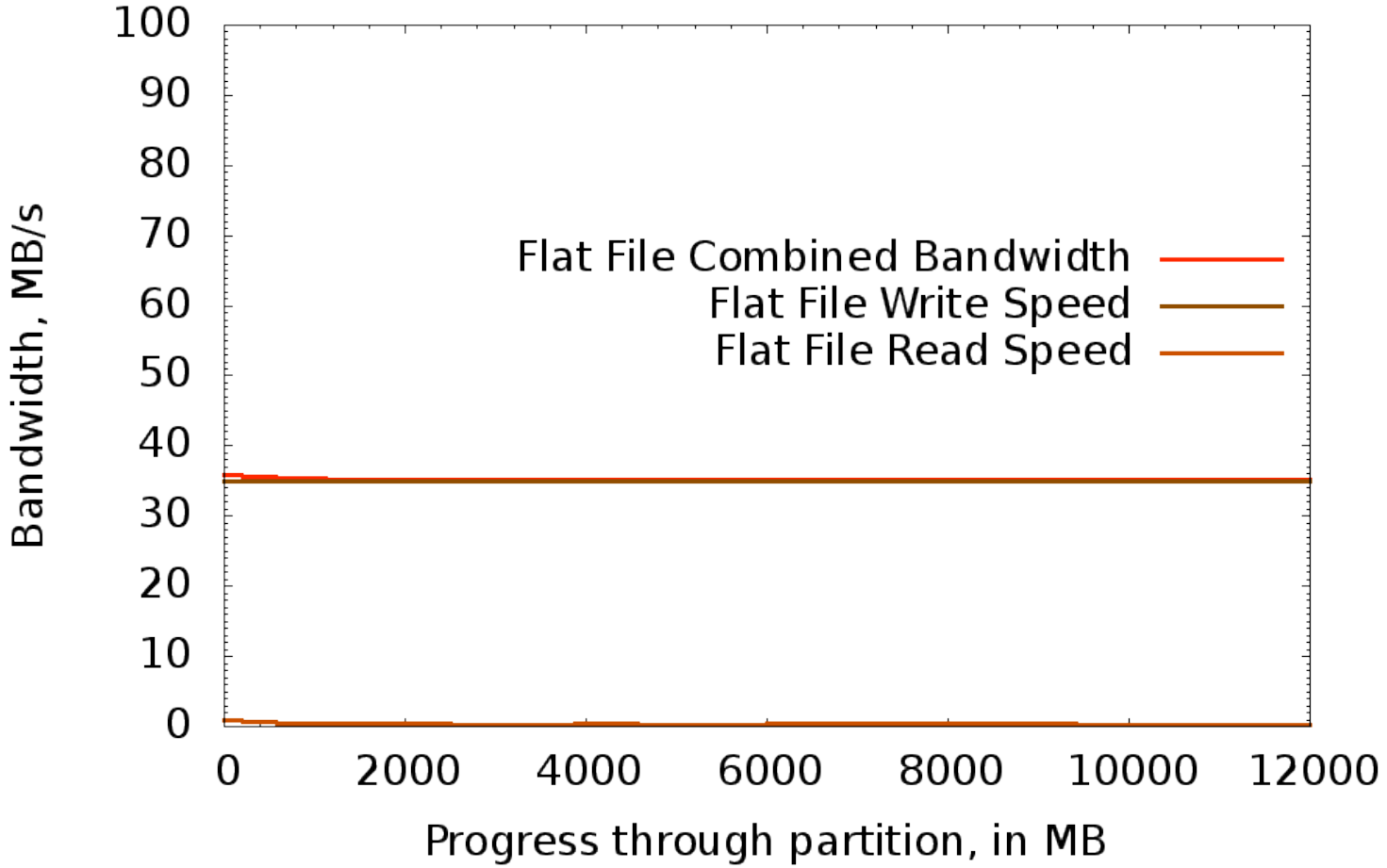
Requested write speed: 60 MB/s



Mahanaxar v. ext2 (small elements)

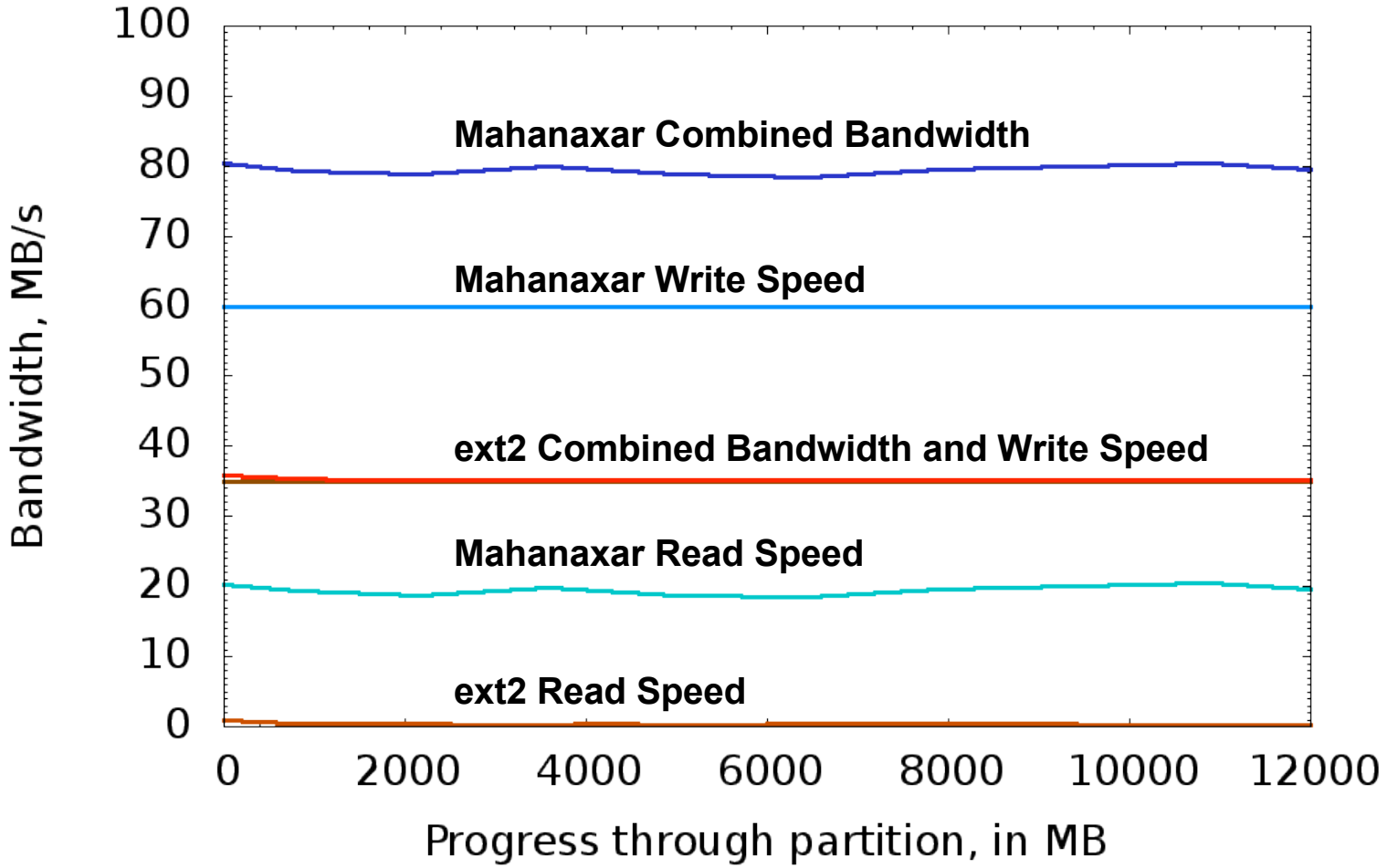
Element size: 1 MB

Requested write speed: 35 MB/s (all it can handle!)



Mahanaxar v. ext2 (small elements)

Combined graphs



Explanation of small elements in ext2

- **Elements get scattered around hard drive**
- **Disk head constantly seeking**
- **These tests overwrote in place**
  - When relying on the file system to expire based on metadata, ext2 starts to fragment extensively
- **Variable element size leads to utter collapse**
  - No “bottoming out” in experimentation

Conclusions

- **Mahanaxar can make QoS guarantees**
- **Mahanaxar provides performance close to raw disk capabilities**
- **Mahanaxar has superior performance to ext2 (and other standard filesystems)**
  - Higher available bandwidth
  - Built-in indexing
  - No “lower limit” to data element size
  - Minimal fragmentation
- **Future work: scalability, data rebuilding, search performance**

Acknowledgments and Questions

- **Questions?**

This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. This work received funding from Los Alamos National Laboratory LDRD Project #20080729DR.