NEC Laboratories America
Relentless passion for innovation

Using Eager Strategies to Improve NFS I/O Performance

Stephen Rago, Aniruddha Bohra, Cristian Ungureanu
NEC Laboratories America

Introduction

- Background
  - Backup appliance development
  - NFS Version 3
  - Backup over NFS was slower than expected
    - With storage system capable of 400 MB/s, couldn't saturate a 1Gb Ethernet
    - With 10Gb Ethernet, can't approach throughput of storage subsystem
  - Built server testbed with conventional storage subsystem: ext3 on top of striped, 15K RPM disks
    - Server capable of 300 MB/s throughput to storage subsystem

NFS Performance Problems

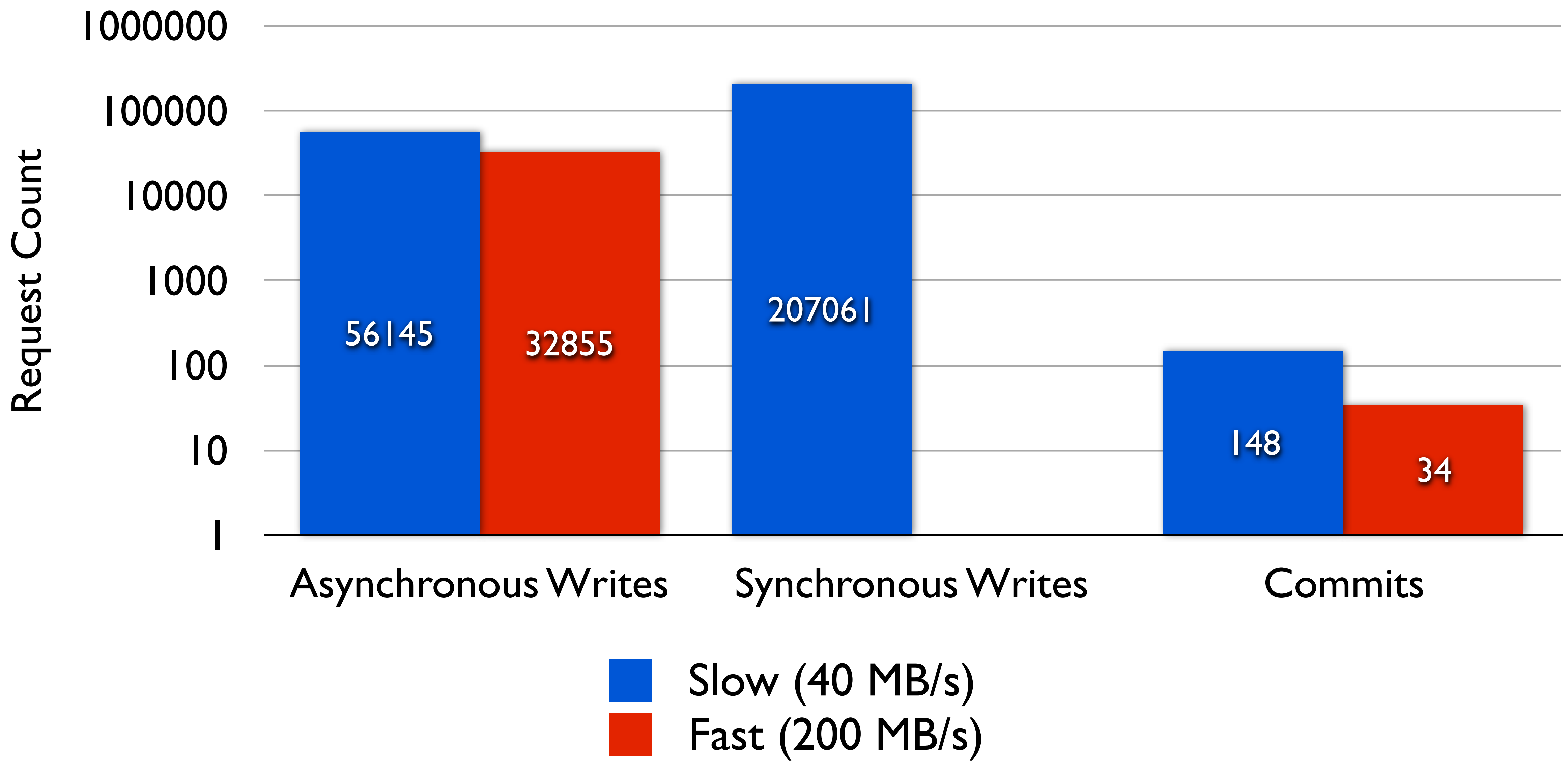
- Streaming <u>write</u> performance erratic
  - Tuning the system to cache more data caused write throughput to vary from 40 MB/s to 200 MB/s on our test systems *for the same set of tunable values*
  - Slow performance results from:
    - Multiple contexts writing generate out-of-order requests
    - Memory pressure leads to small, synchronous writes
    - Memory pressure also increases commits
- Streaming <u>read</u> performance lower than expected
  - Less than 100 MB/s on 10Gb Ethernet
    - Out-of-order requests defeat kernel read-ahead logic

Concurrency = Out-of-Order NFS Requests

	Reads	Writes
Client	Read-ahead	Multiple writers (background flushing, pageout, and application) plus asynchronous writes
Server	Multiple NFS threads	Multiple NFS threads

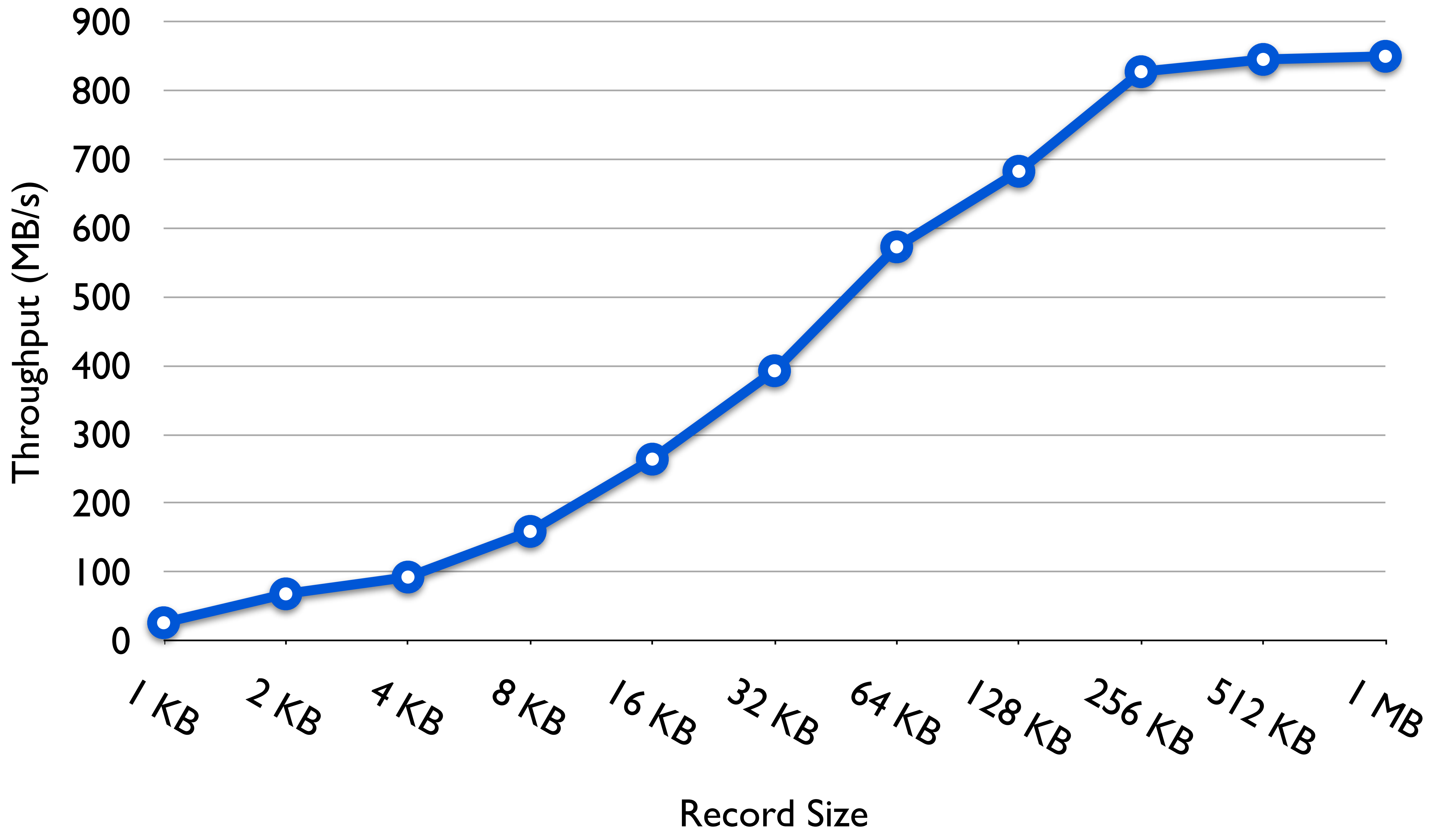
NEC Laboratories America
Relentless passion for innovation

Problem 1: Synchronous Operations - Base System, Slow Run vs. Fast Run for Same Amount of Data Written



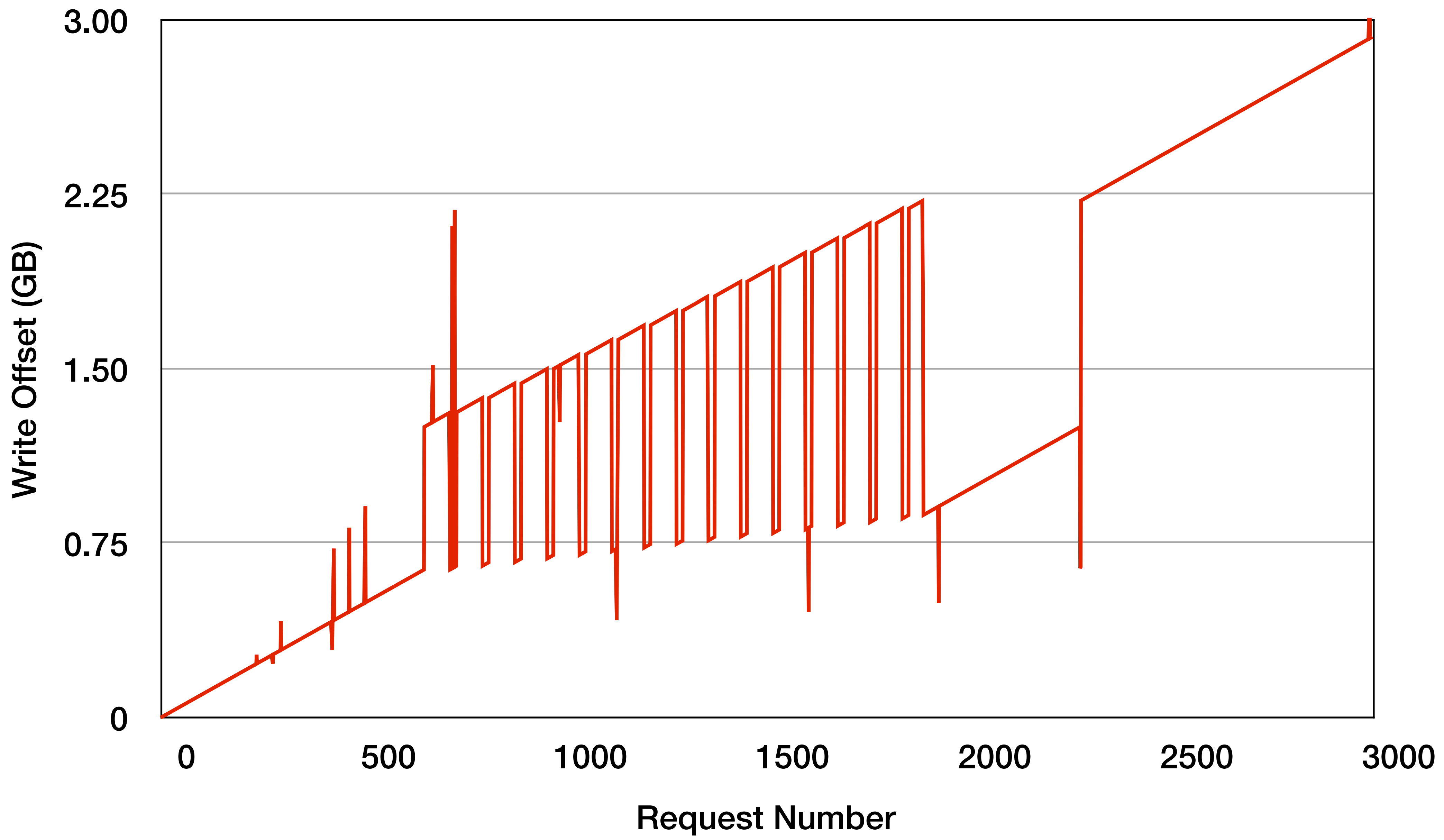
NEC Laboratories America
Relentless passion for innovation

Problem 2: Small Record Sizes - Idealized NFS Write Throughput



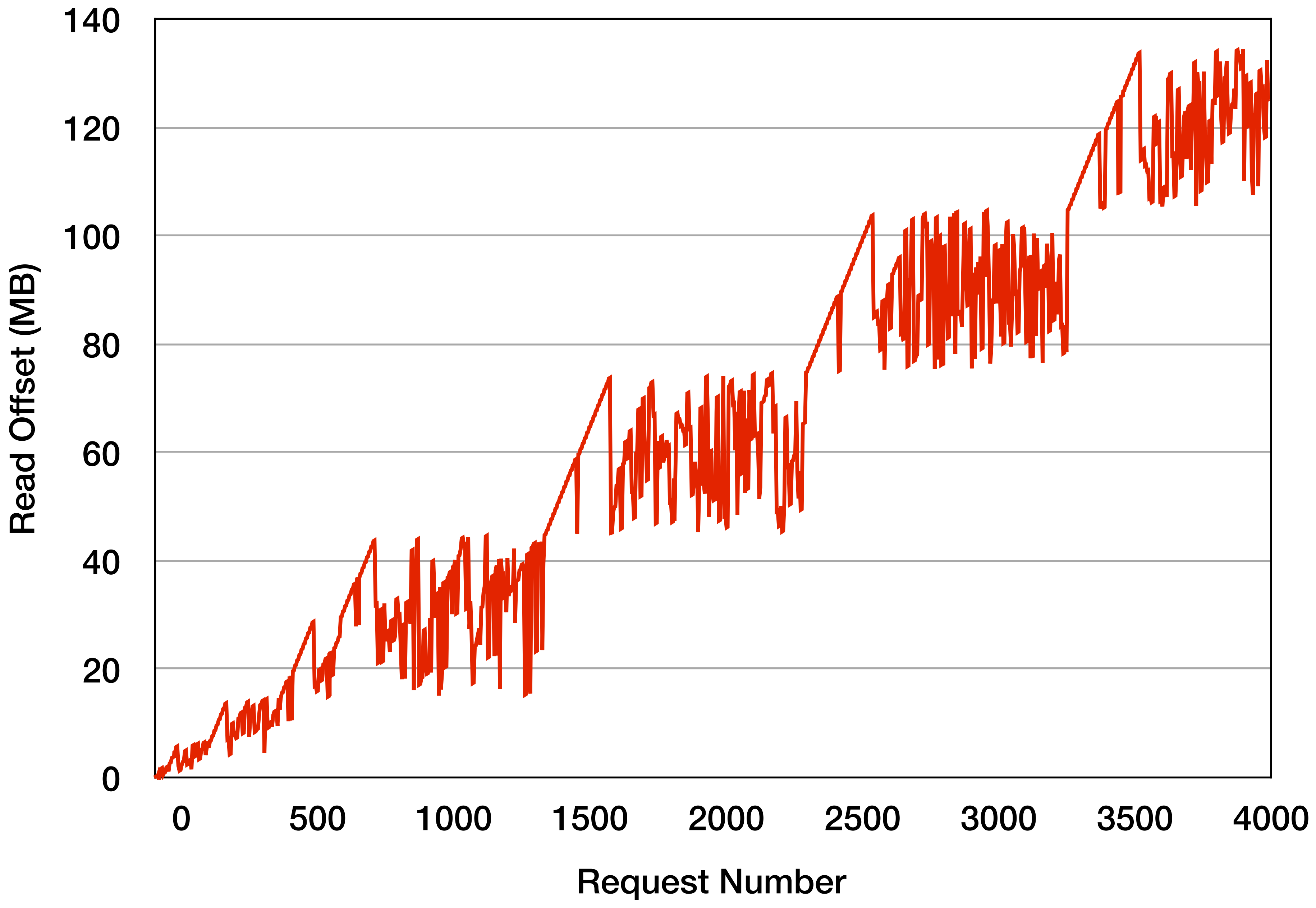
NEC Laboratories America
Relentless passion for innovation

Problem 3: NFS Write Offset Ordering
(Writing a 32 GB File)



NEC Laboratories America
Relentless passion for innovation

Problem 4: NFS Read Offset Ordering
(Reading a 32 GB File)



Solutions

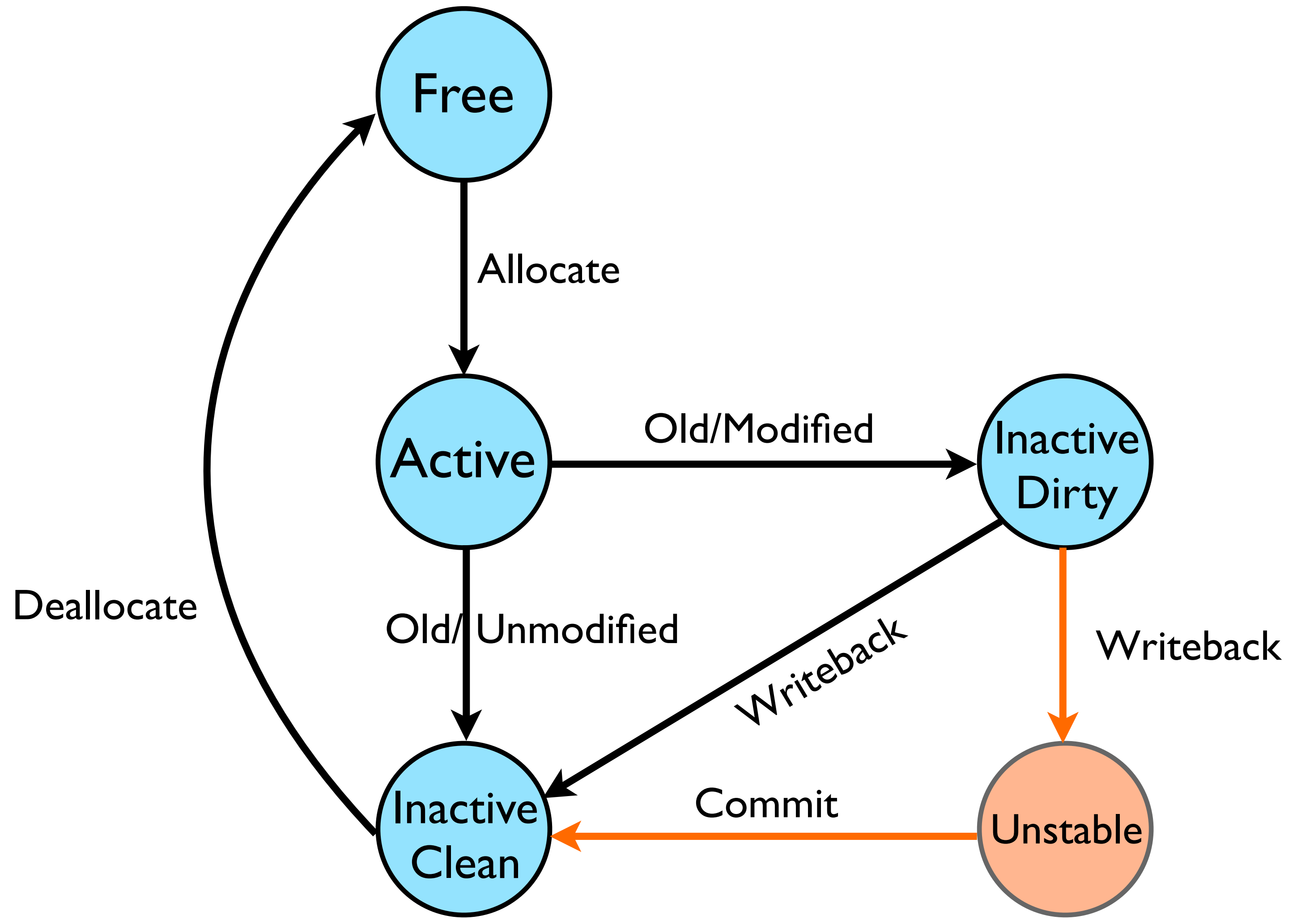
- Three general techniques
  - Eager Writeback
    - Reduces concurrency on client and maintains sequentiality
  - Eager Page Laundering
    - Reduces client memory pressure
  - Request Ordering
    - Prevents out-of-order operations on a single file
- Implemented on Linux 2.6.36
- Techniques applicable to other operating systems

Technique 1: Eager Writeback

- Client-side mechanism
- Prevents application from creating dirty pages quickly
  - Pages written eagerly to server
  - Client waits for outstanding requests to complete before continuing
- Advantages
  - Starts sending dirty pages earlier -- better server utilization
  - Only one thread writes a file's pages to the server
  - Better flow control
- Disadvantages
  - Starts sending dirty pages earlier -- limited page reuse for overwriting patterns

NEC Laboratories America
Relentless passion for innovation

Simplified Page State Diagram

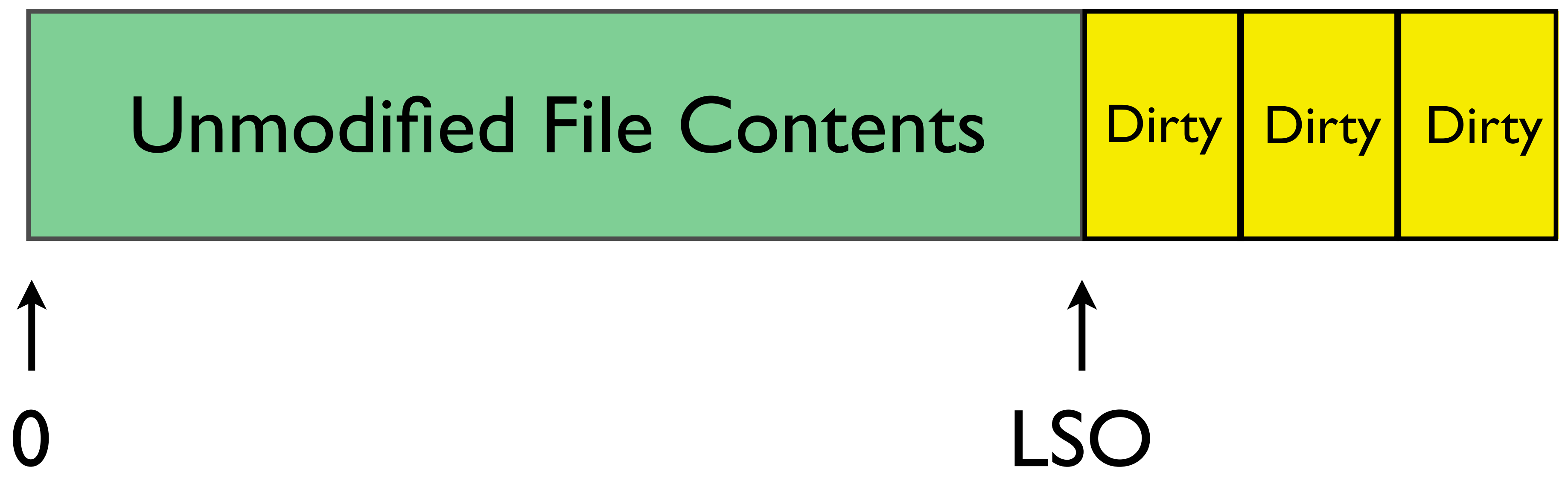


Technique 2: Eager Page Laundering

- Client & Server mechanism
- Dirty pages on server eventually become clean
- Communicate *largest stable offset* from server to client
    - Piggybacked in NFS write response (takes half of verifier)
    - Negotiated at mount time
- Client reclaims ("launders") pages eagerly
- Advantages
    - Reduces memory pressure on client
    - No commits or synchronous writes needed
- Disadvantages
    - Small protocol change

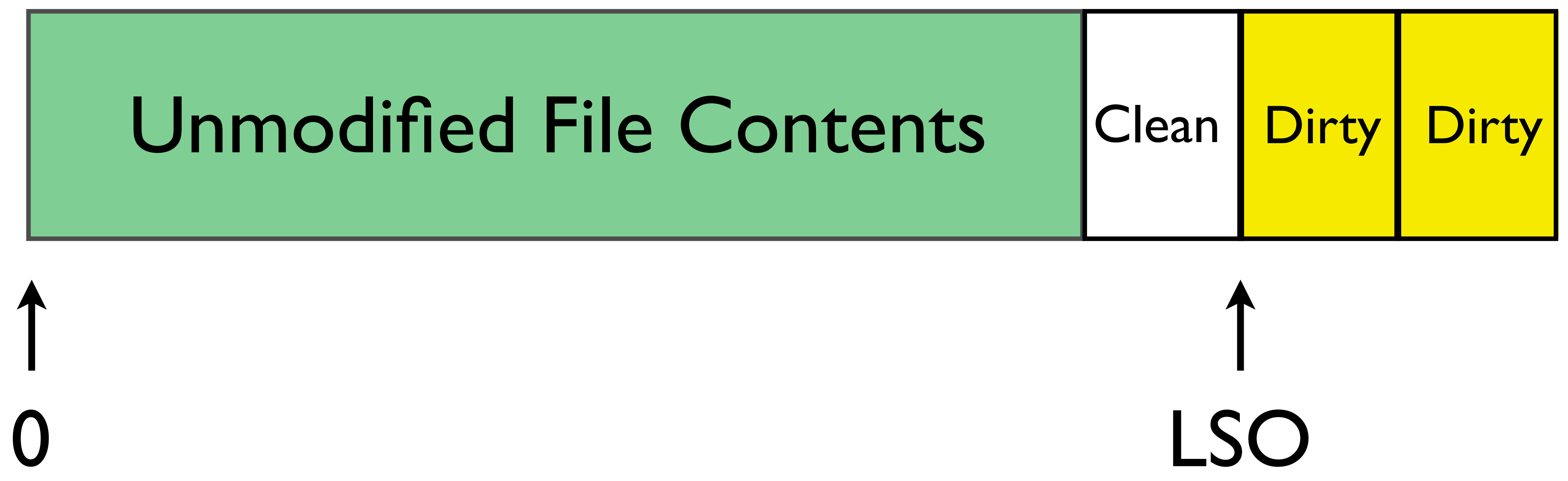
NEC Laboratories America
Relentless passion for innovation

Largest Stable Offset (LSO)



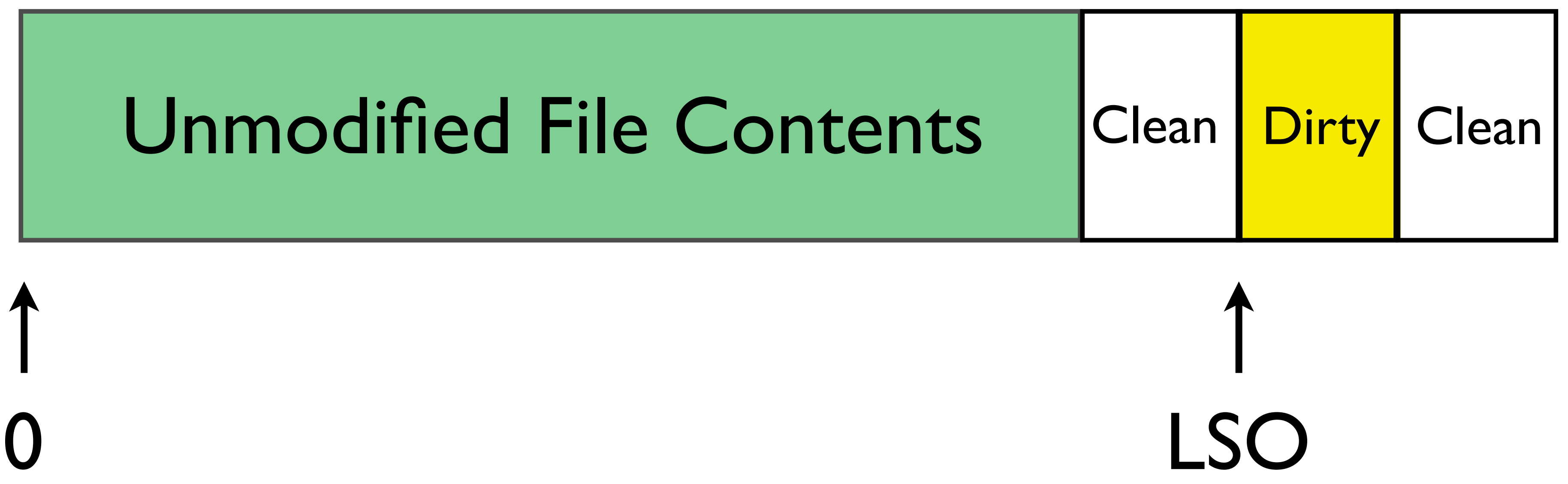
NEC Laboratories America
Relentless passion for innovation

Largest Stable Offset (LSO)

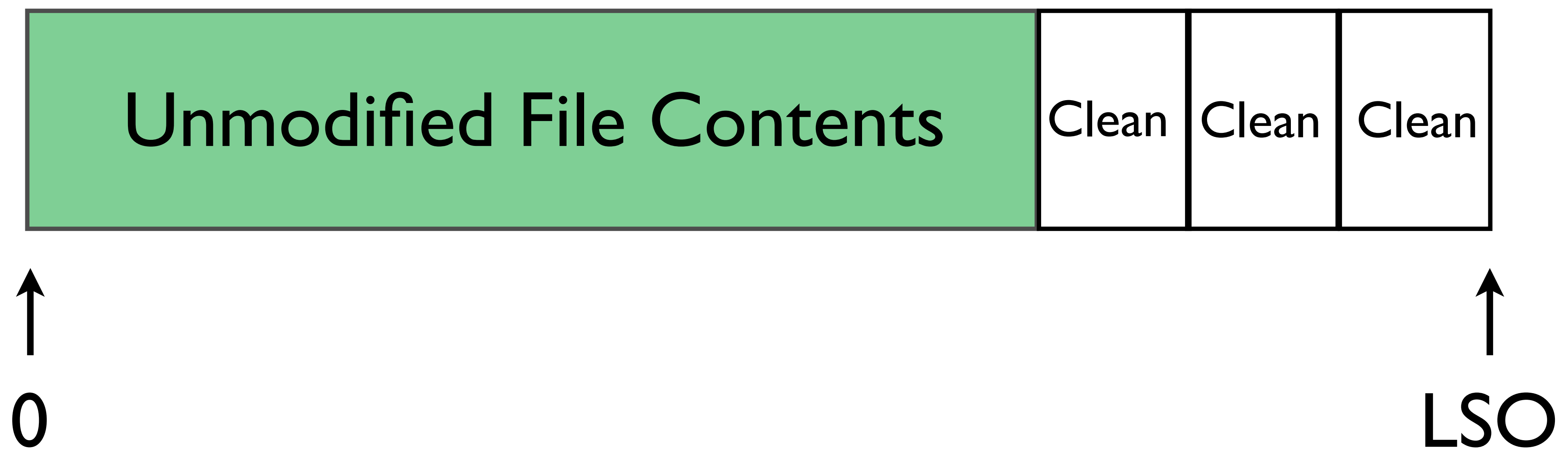


NEC Laboratories America
Relentless passion for innovation

Largest Stable Offset (LSO)



Largest Stable Offset (LSO)

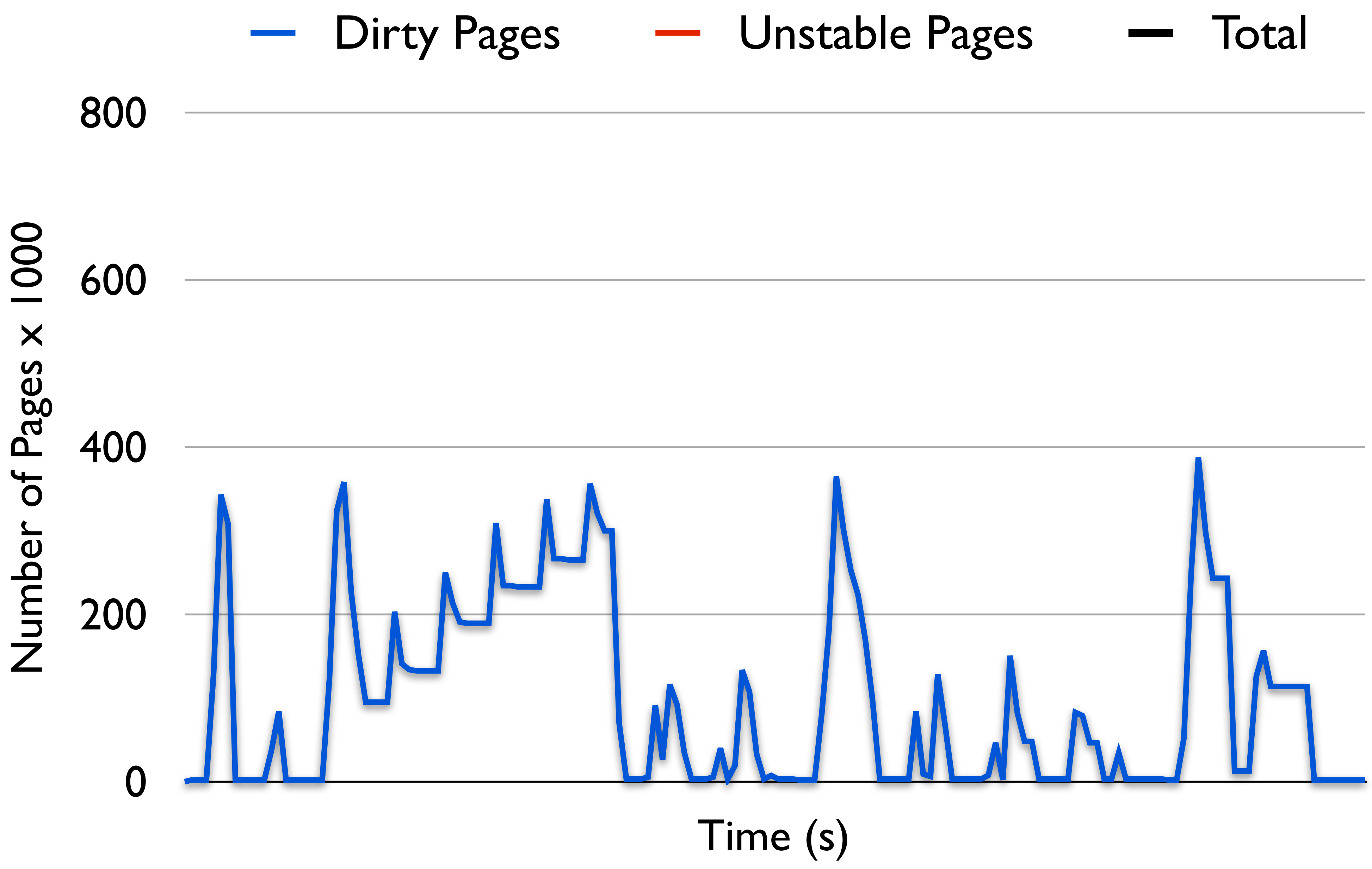


NEC Laboratories America
Relentless passion for innovation

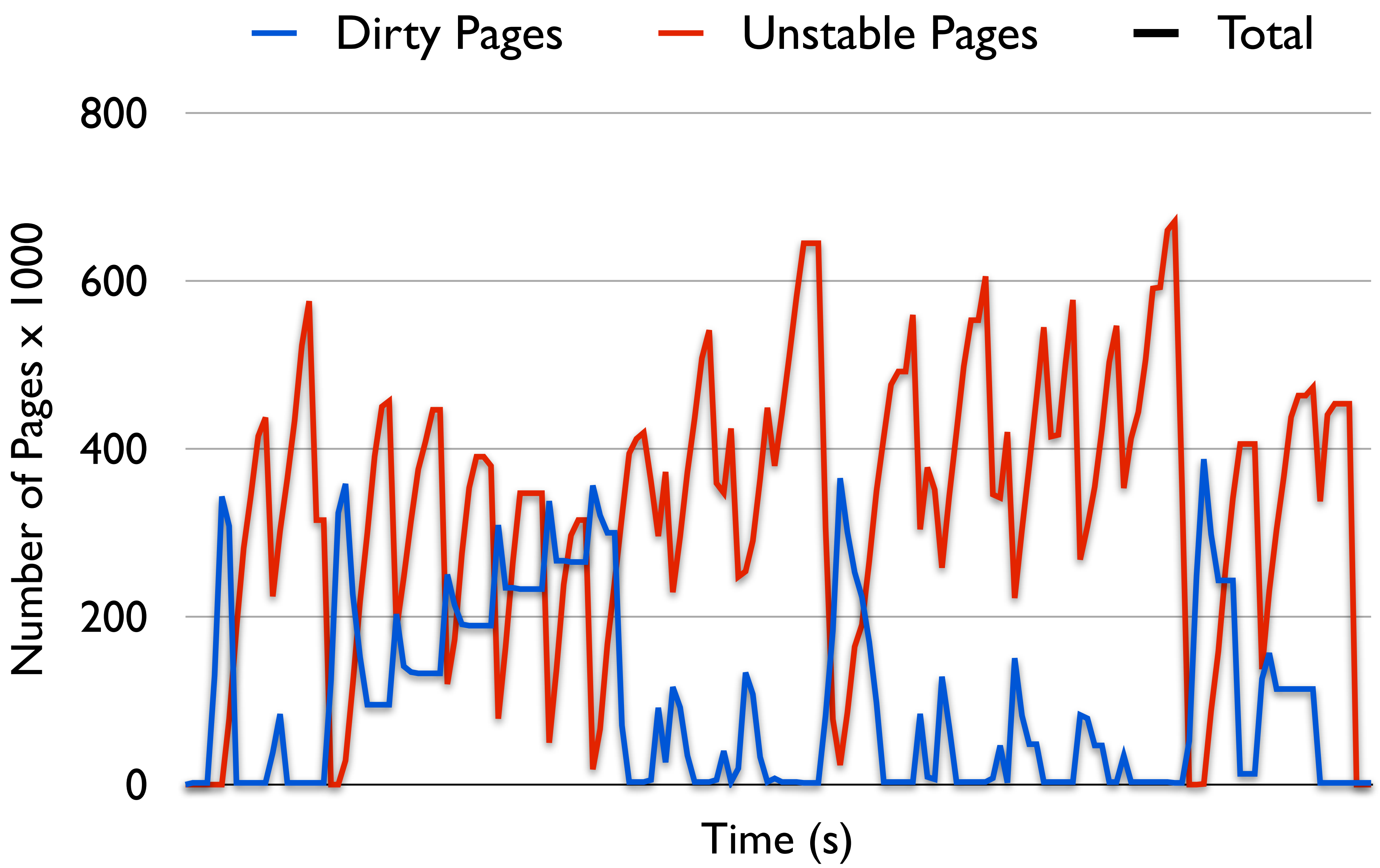
Base Client Page Counts

— Dirty Pages — Unstable Pages — Total

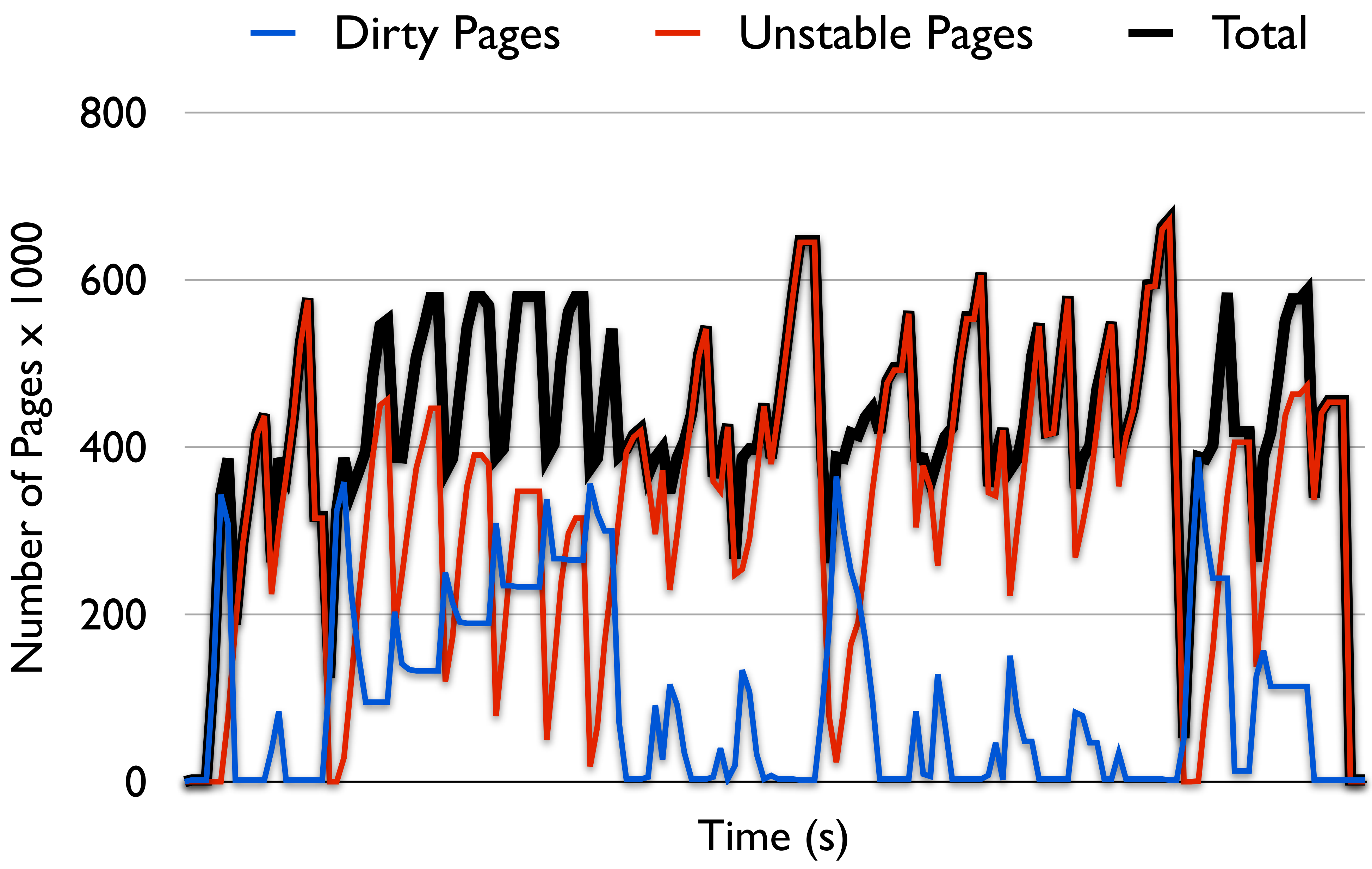
Base Client Page Counts



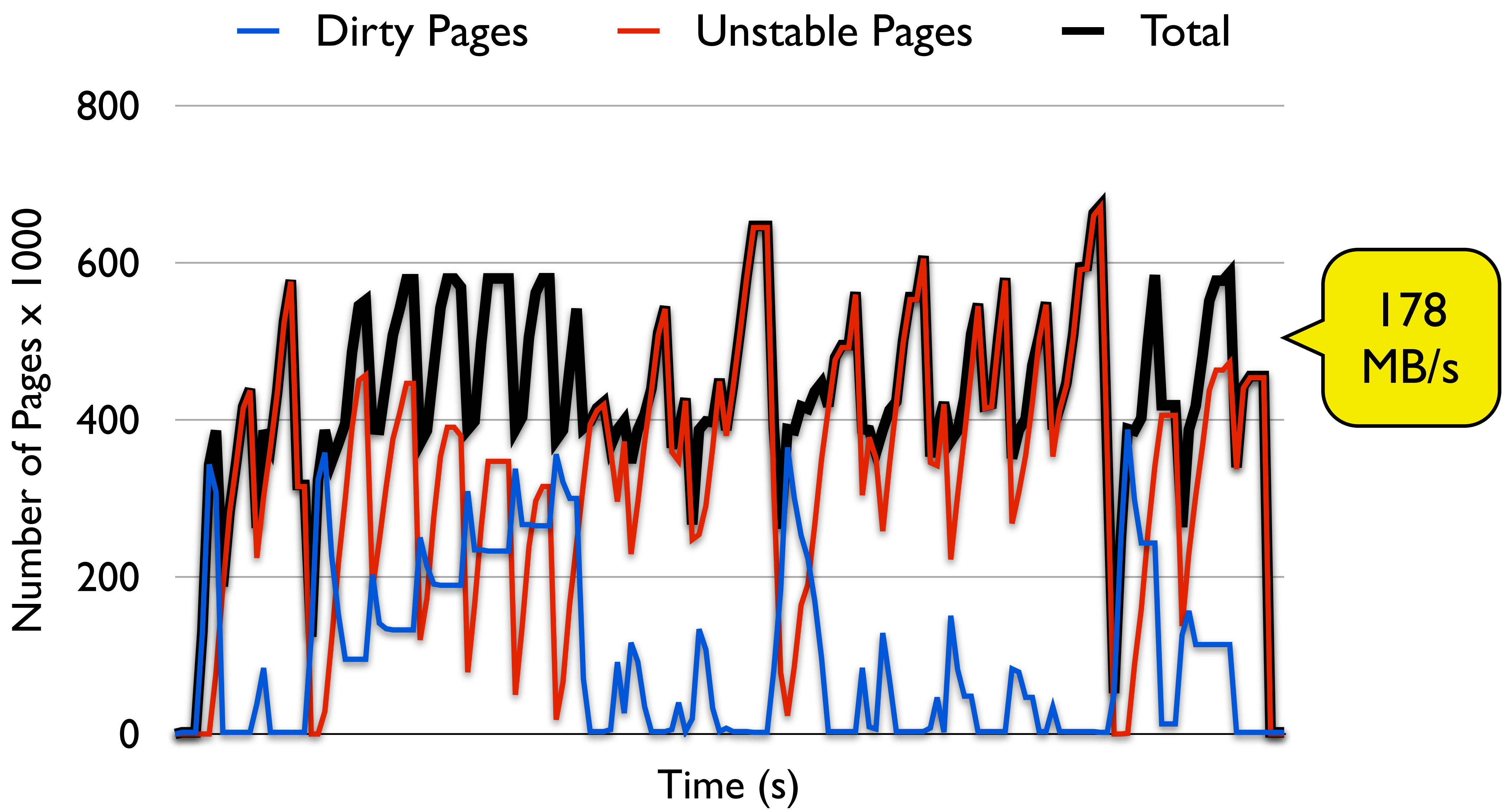
Base Client Page Counts



Base Client Page Counts



Base Client Page Counts



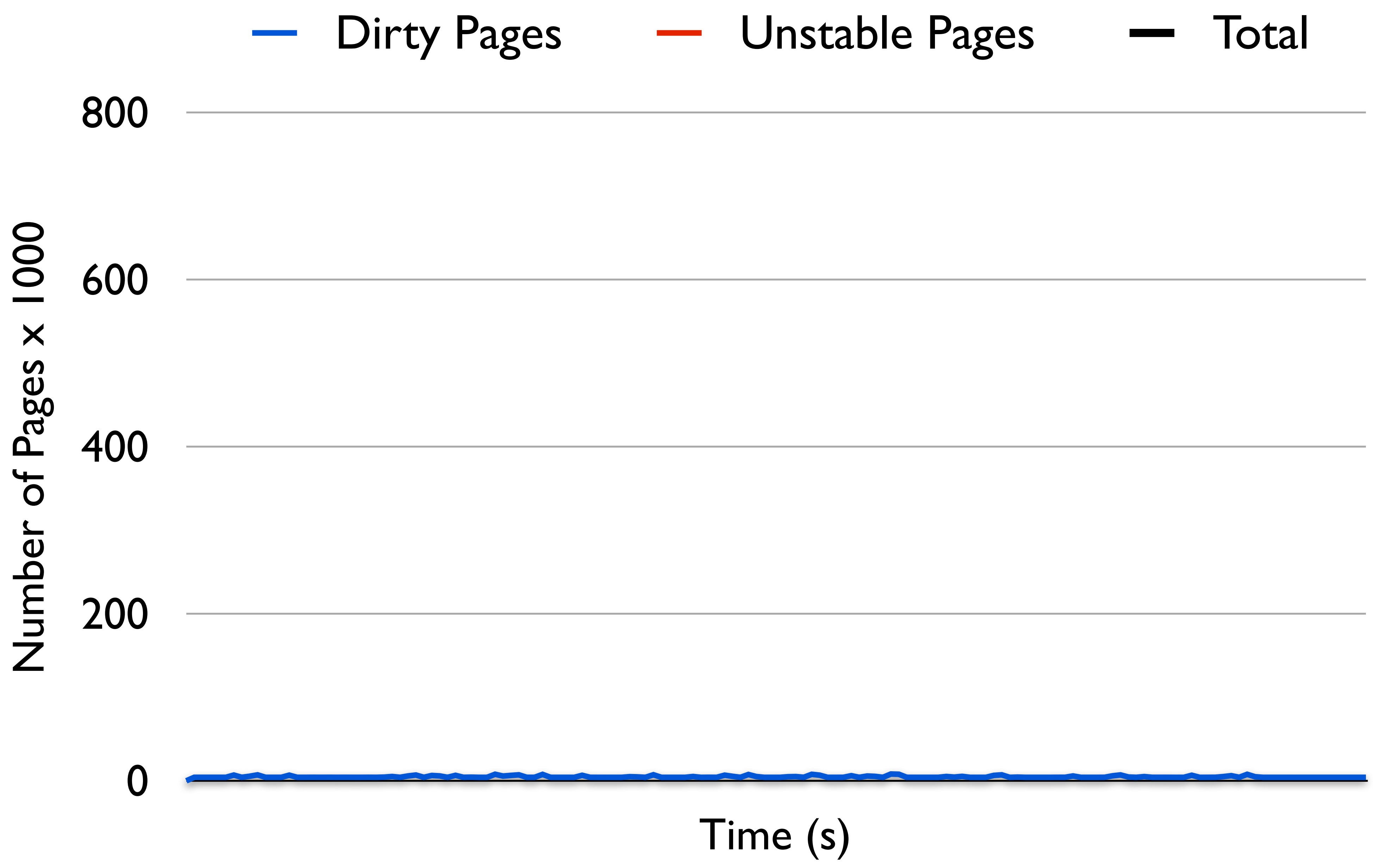
NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback Only

— Dirty Pages — Unstable Pages — Total

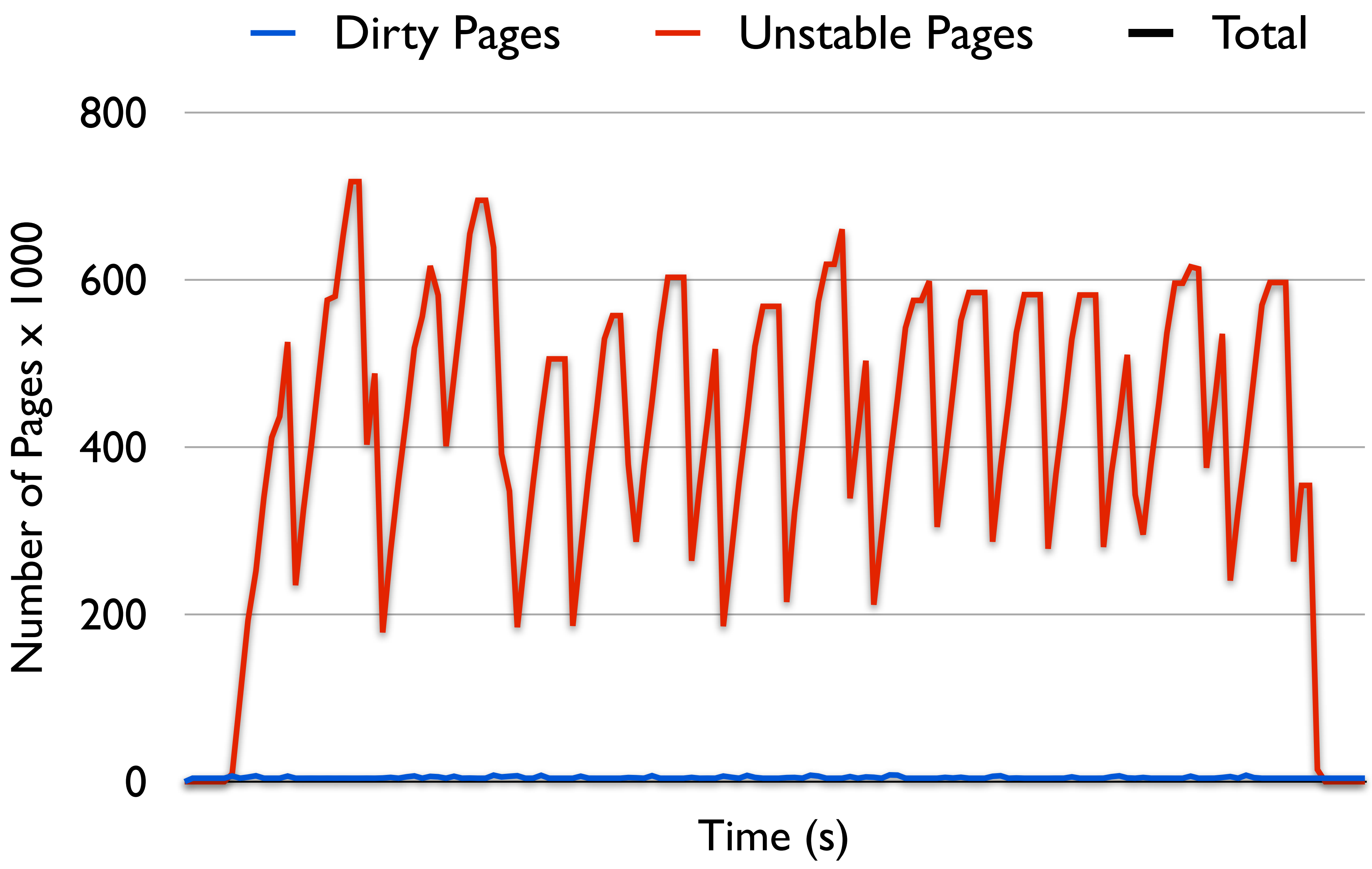
NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback Only



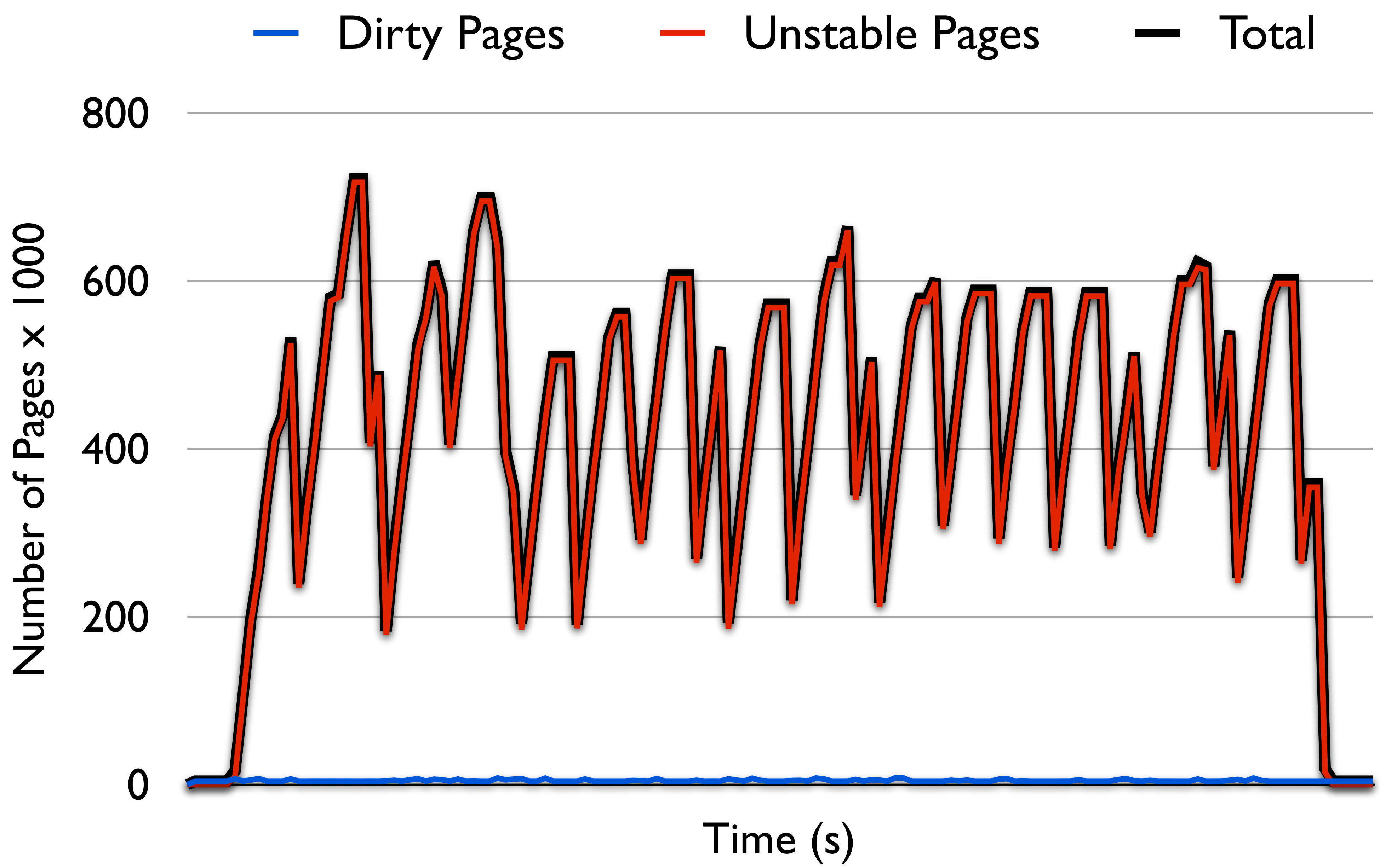
NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback Only

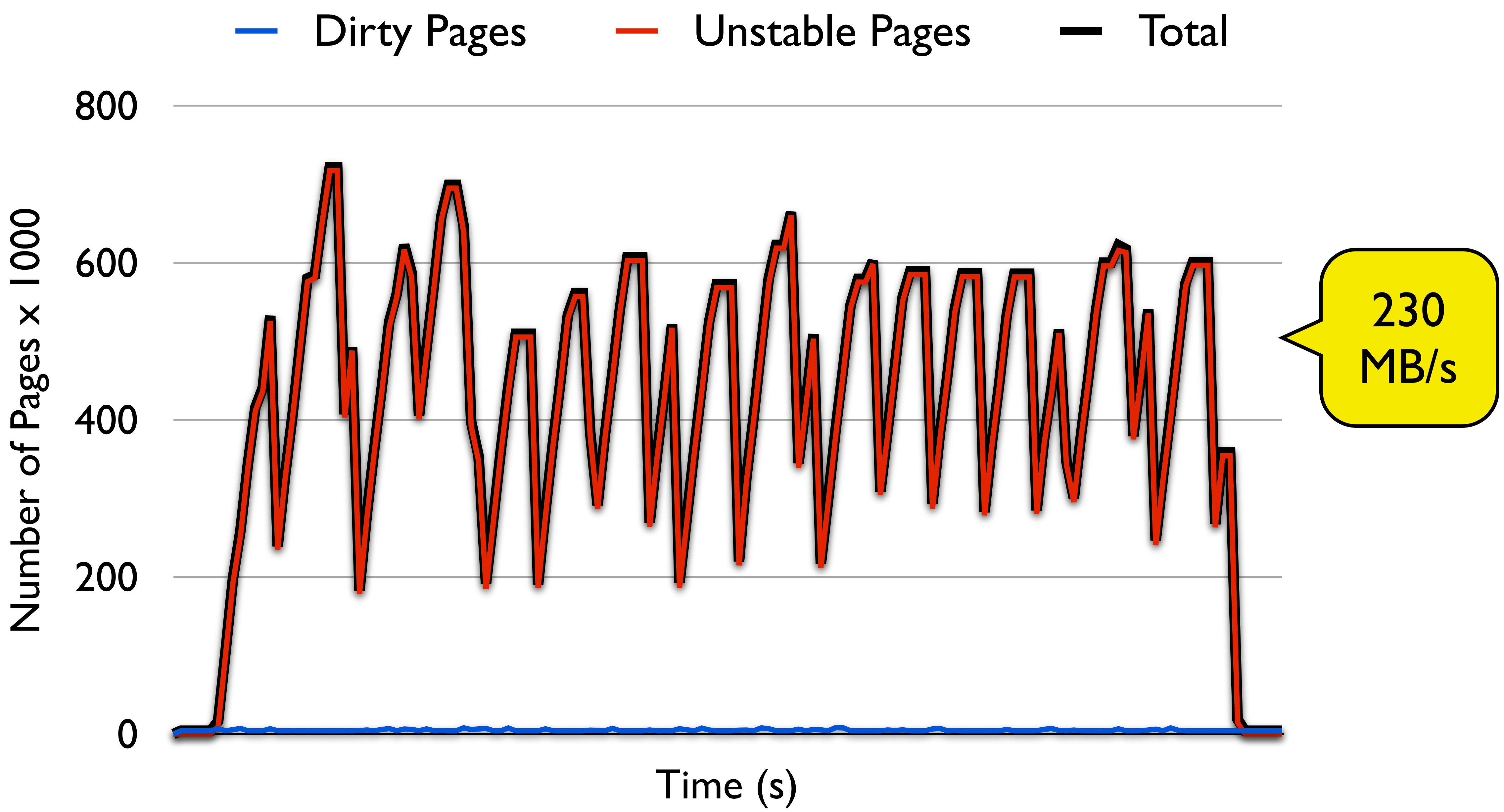


NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback Only



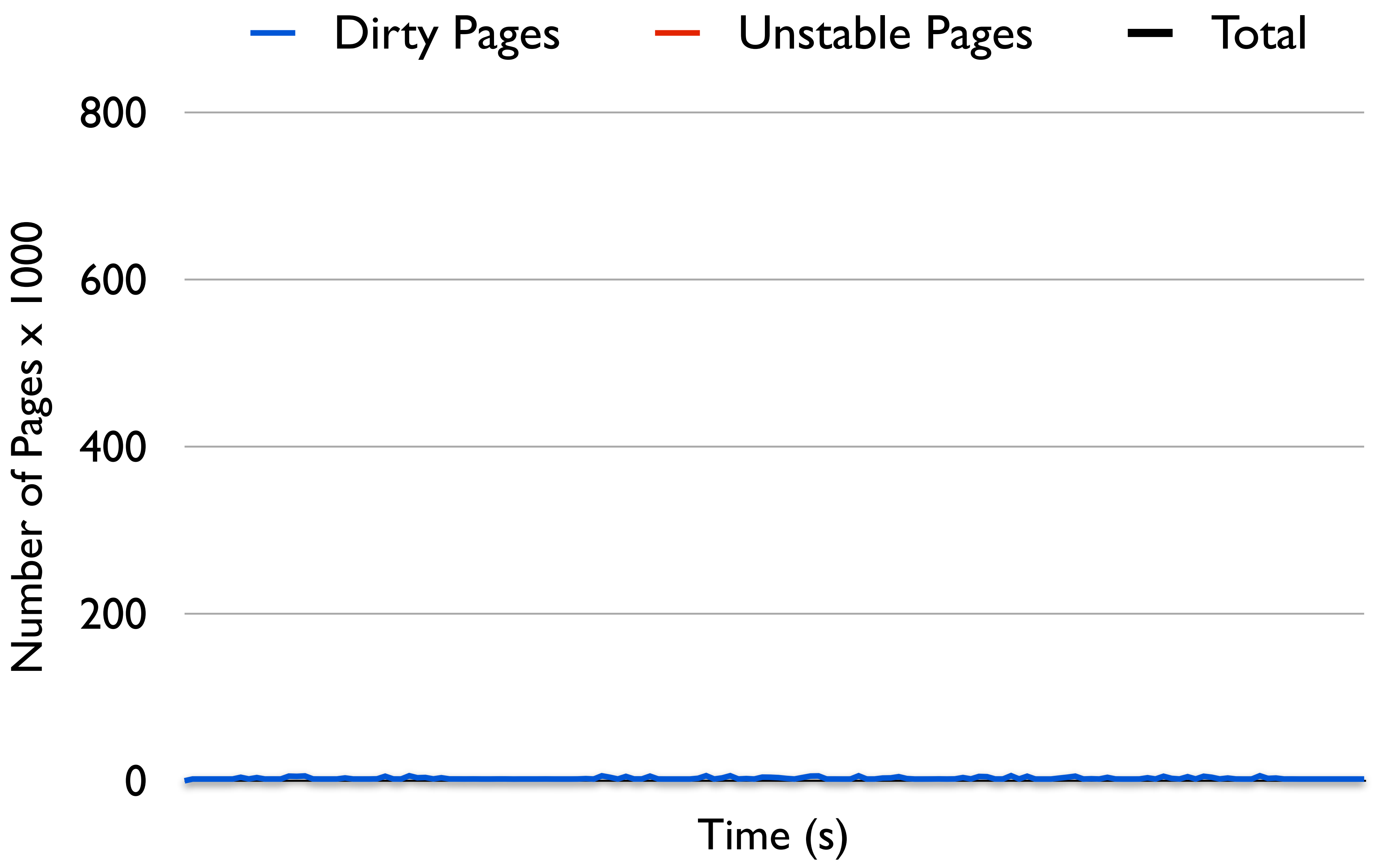
Client Page Counts - Eager Writeback Only



Client Page Counts - Eager Writeback & Eager Page Laundering

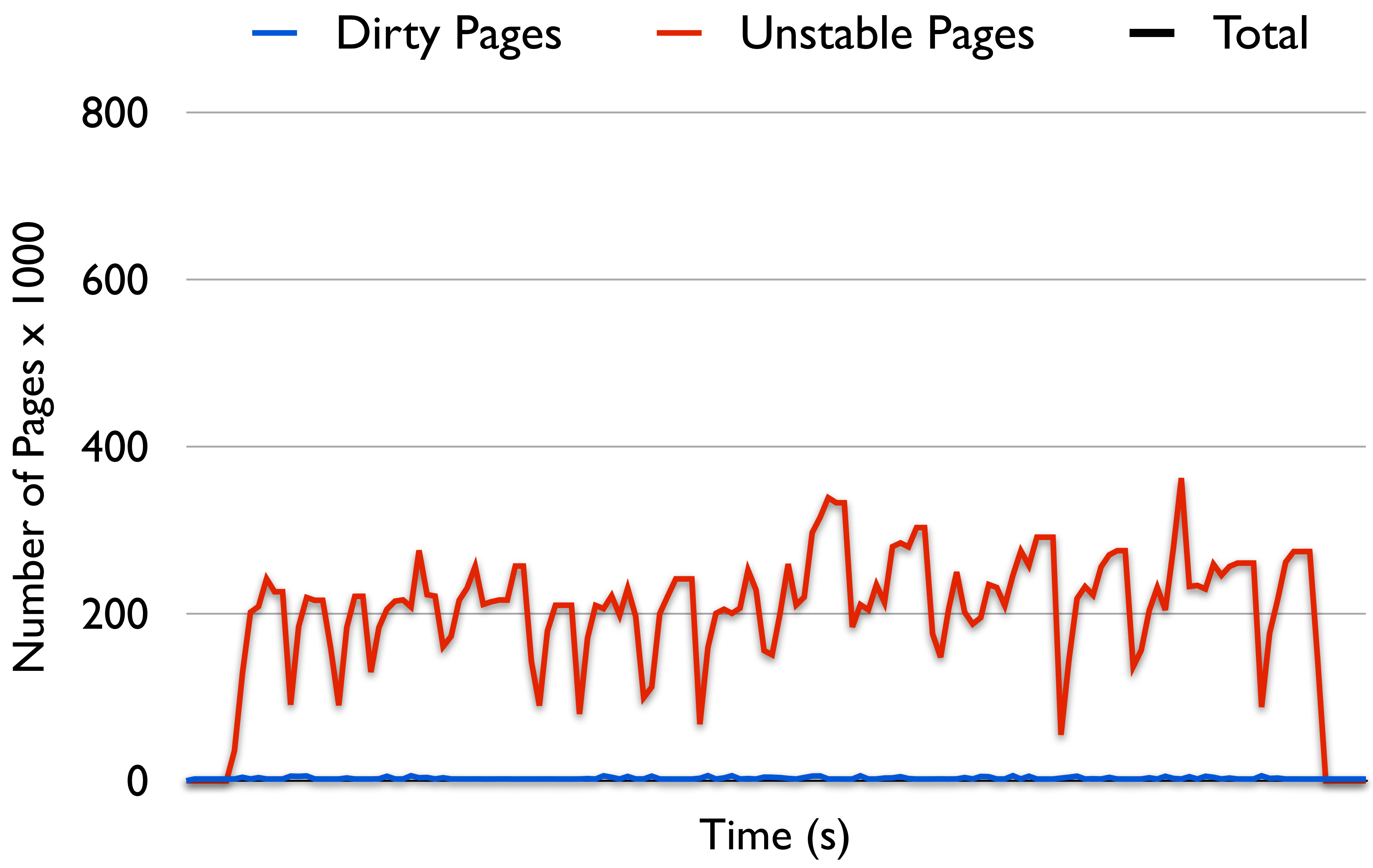
— Dirty Pages — Unstable Pages — Total

Client Page Counts - Eager Writeback & Eager Page Laundering



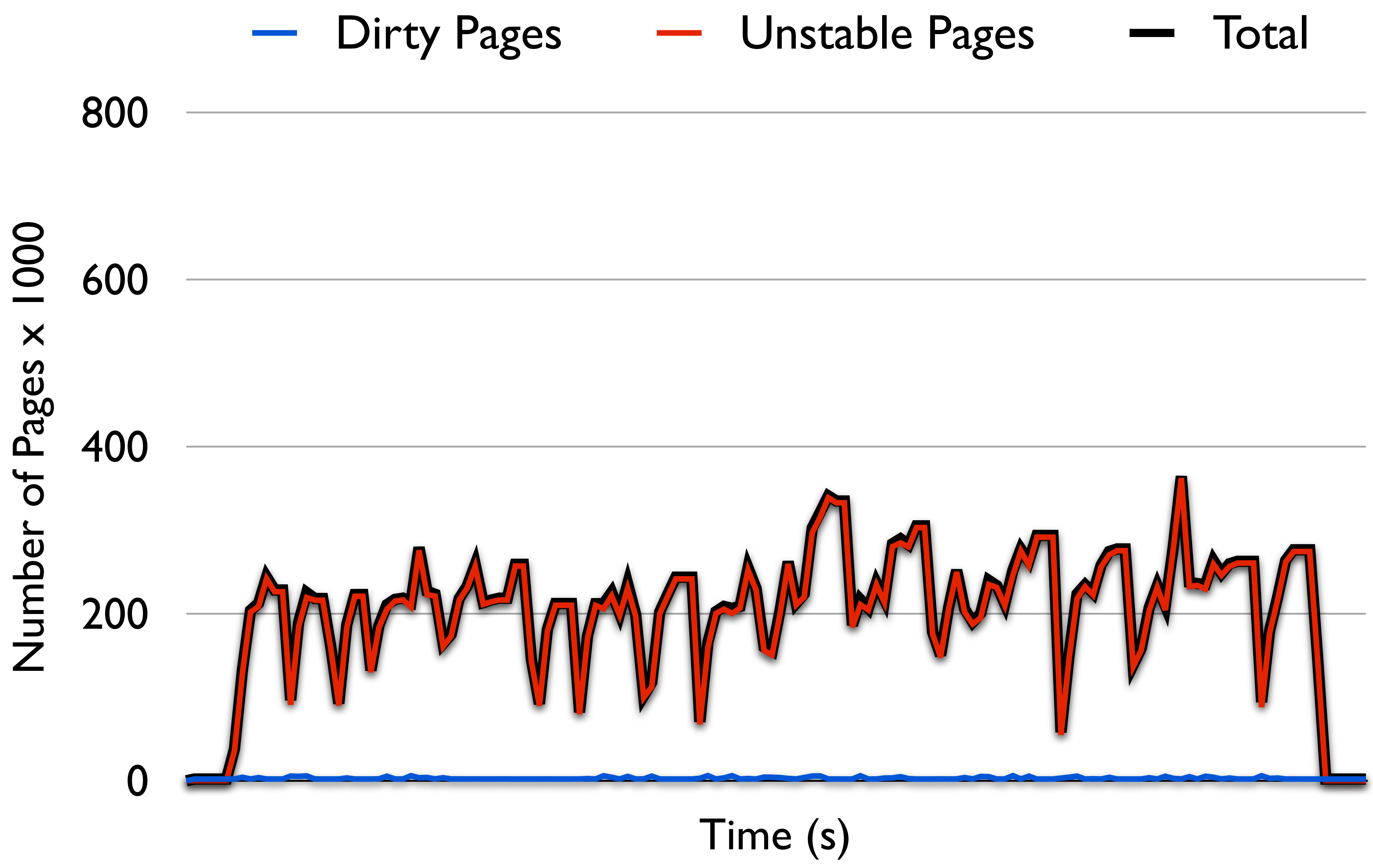
NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback & Eager Page Laundering



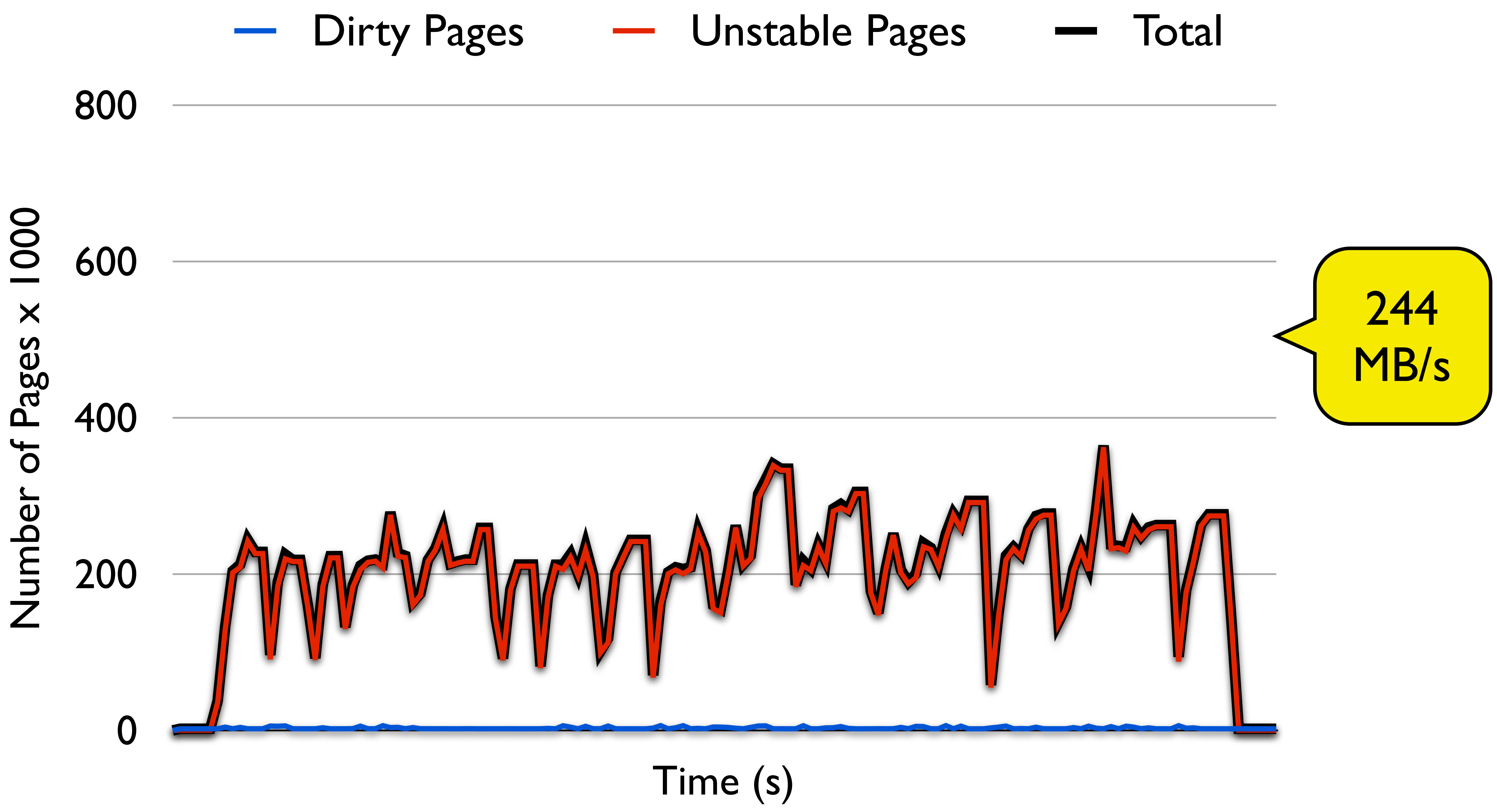
NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback & Eager Page Laundering



NEC Laboratories America
Relentless passion for innovation

Client Page Counts - Eager Writeback & Eager Page Laundering



Technique 3: Request Ordering

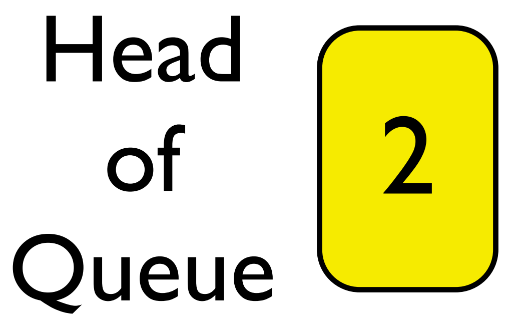
- Server sorts requests based on RPC transmission ID
- Server-side mechanism
- Prevents out-of-order completion of requests from competing threads
- Advantages
  - Improves sequential read performance
  - When used during writes, can further improve read performance (depending on file system implementation)
- Disadvantages
  - Adds a small delay (50 ns) on reads to facilitate sorting, but only for sequential reads on files where the queue is empty

NEC Laboratories America
Relentless passion for innovation

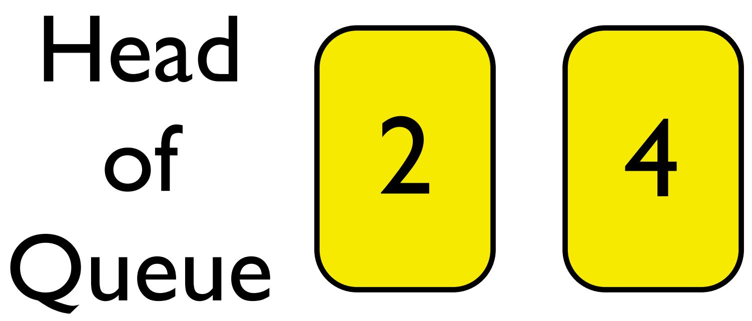
Sorting Request on the Server

Head
of
Queue

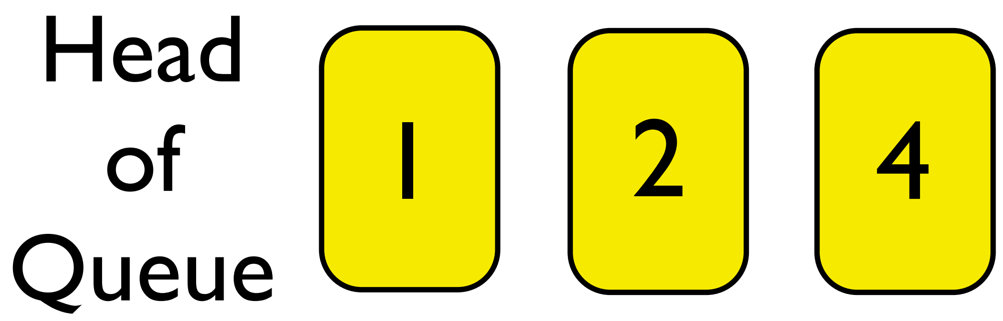
Sorting Request on the Server



Sorting Request on the Server

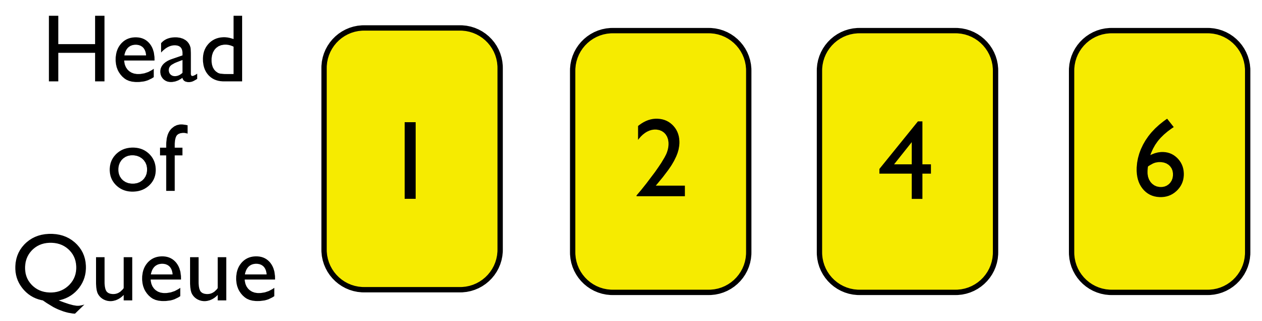


Sorting Request on the Server



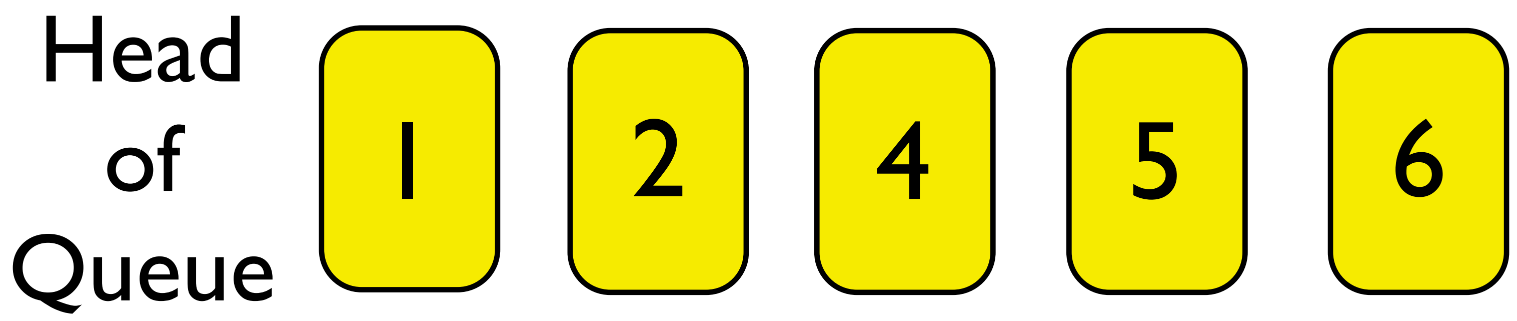
NEC Laboratories America
Relentless passion for innovation

Sorting Request on the Server

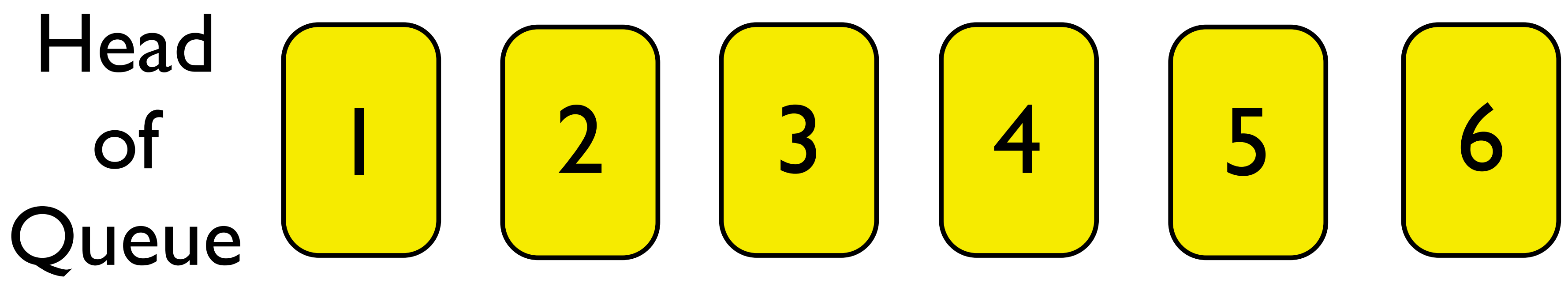


NEC Laboratories America
Relentless passion for innovation

Sorting Request on the Server



Sorting Request on the Server

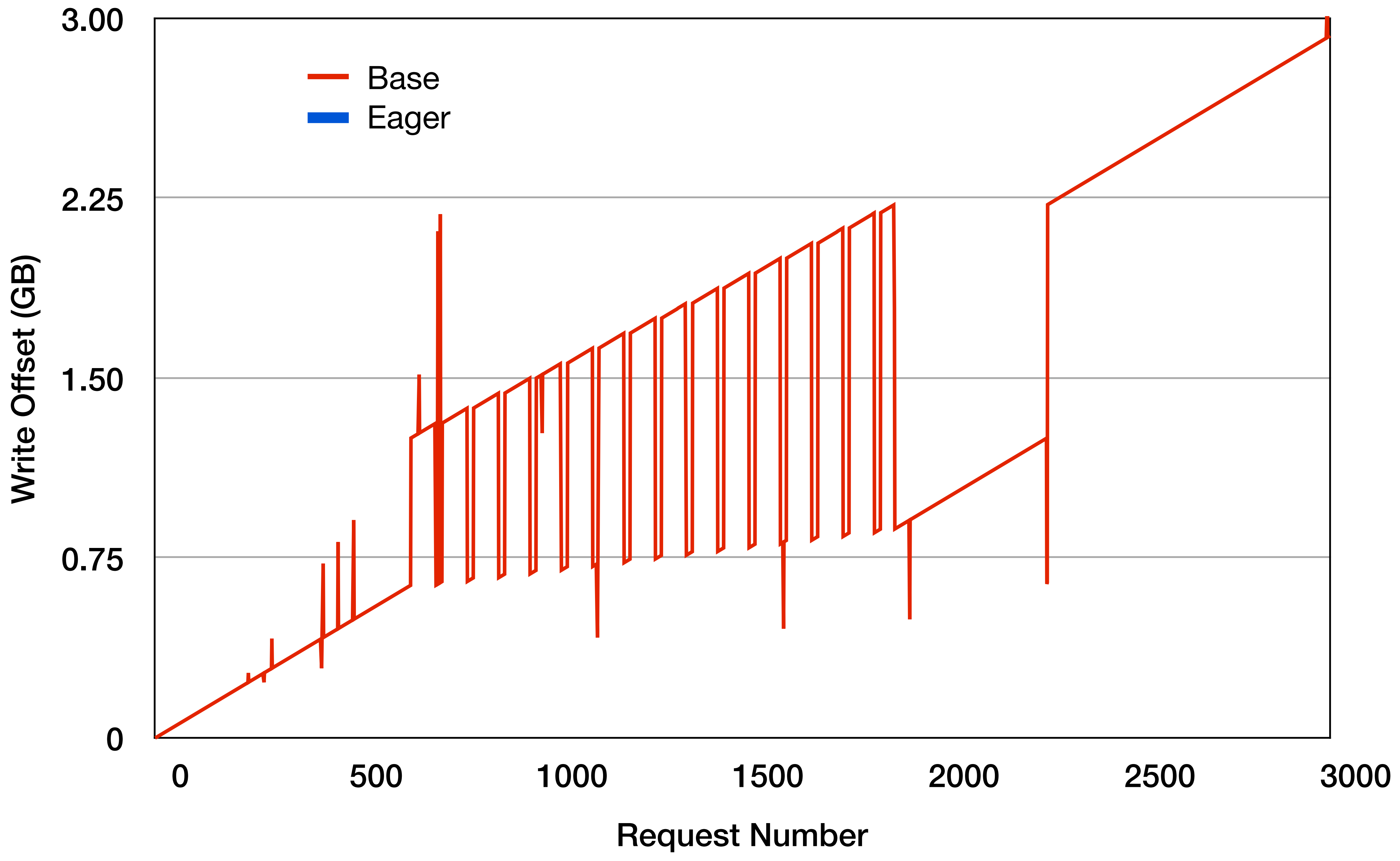


NFS Write Offset Ordering

🟥 Base
🟦 Eager

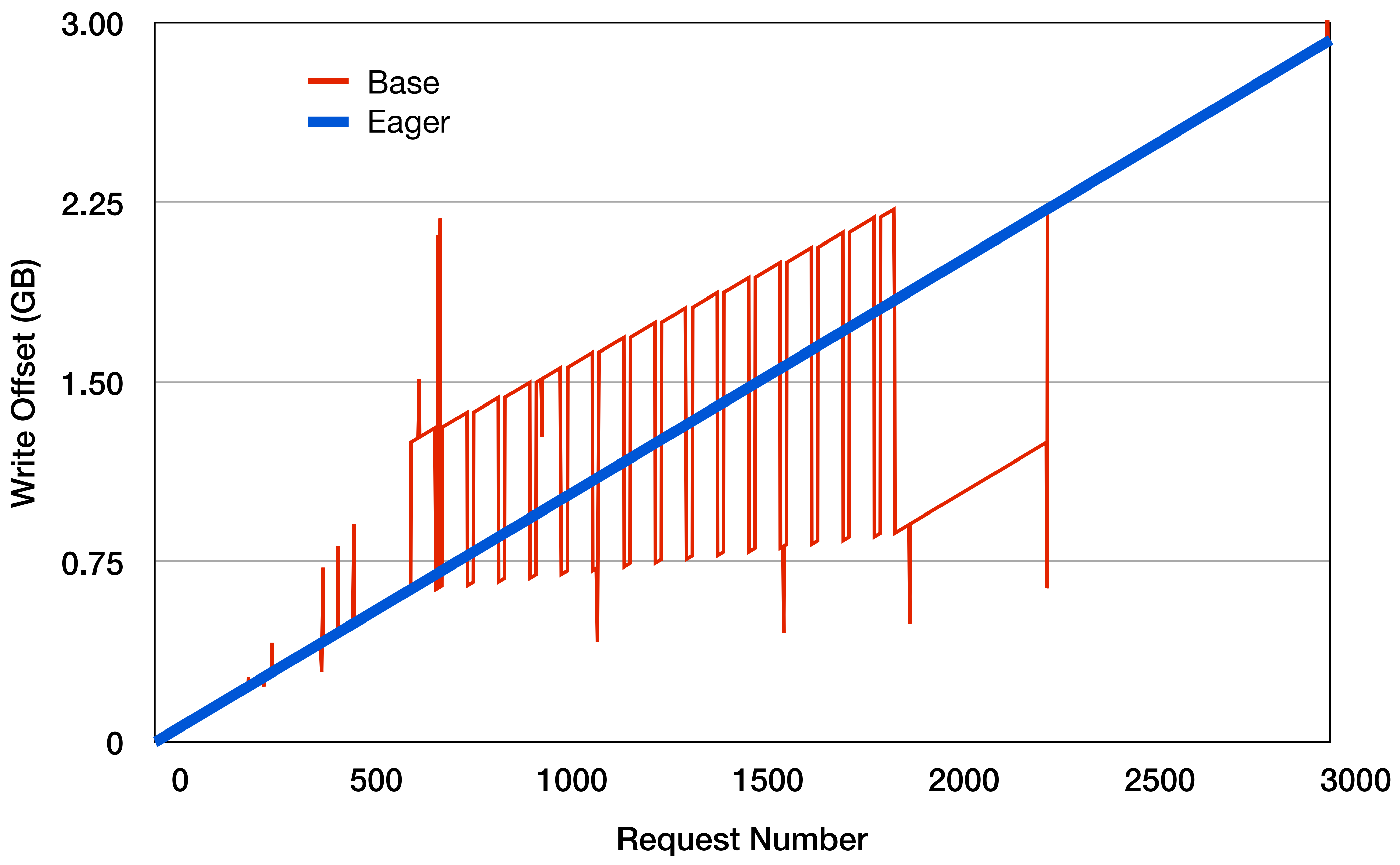
NEC Laboratories America
Relentless passion for innovation

NFS Write Offset Ordering



NEC Laboratories America
Relentless passion for innovation

NFS Write Offset Ordering

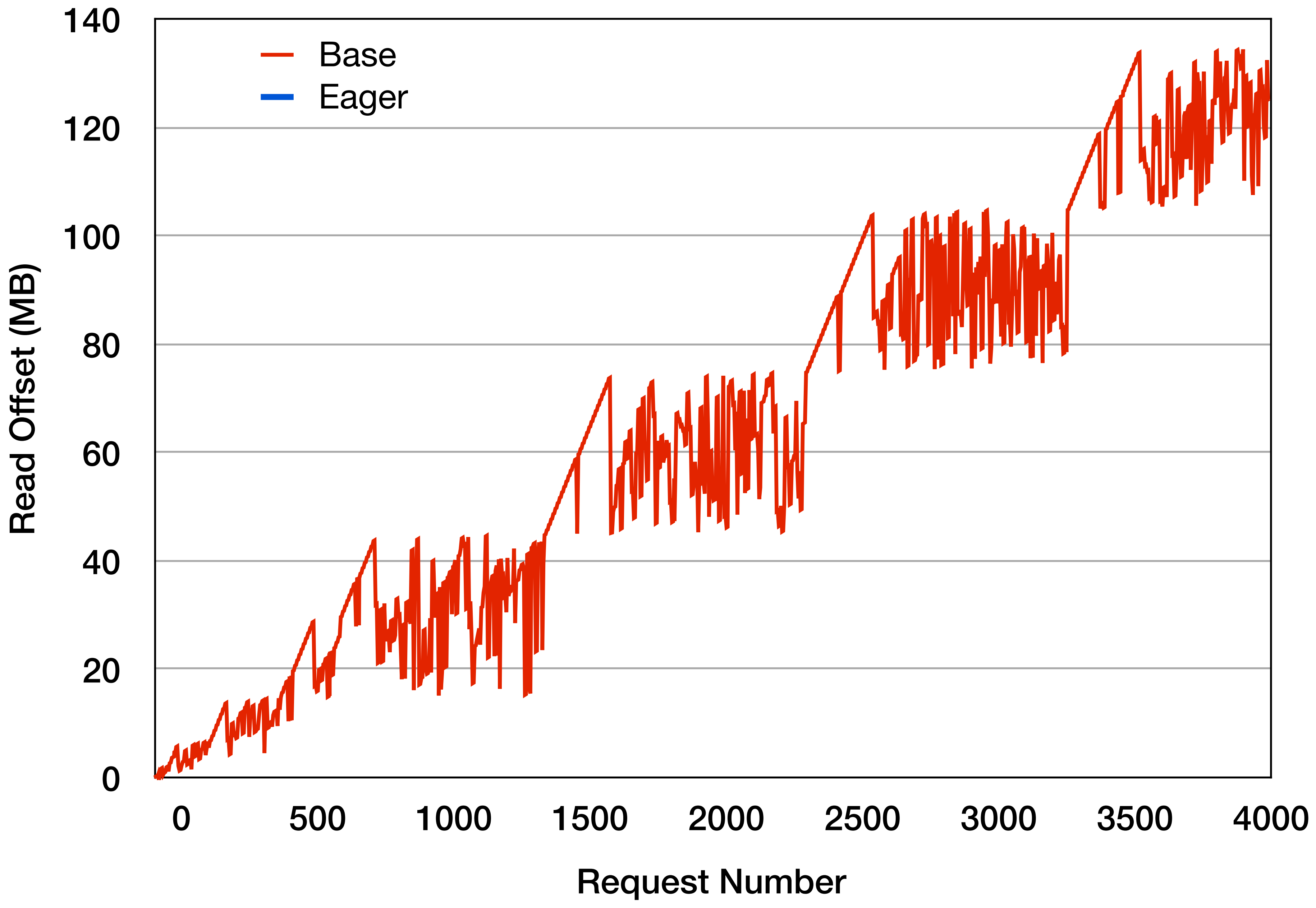


NFS Read Offset Ordering

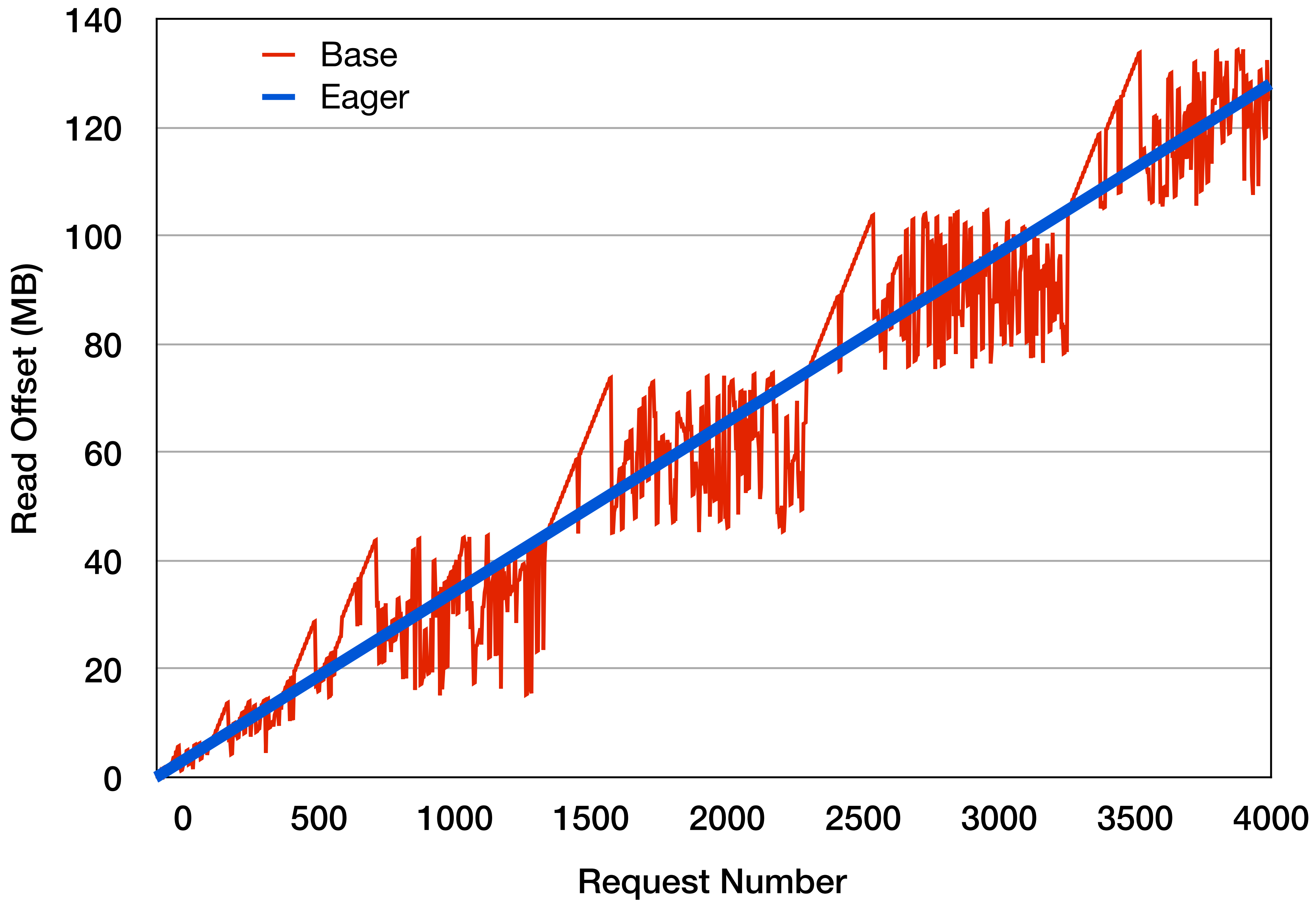
- 🟥 Base
- 🟦 Eager

NEC Laboratories America
Relentless passion for innovation

NFS Read Offset Ordering



NFS Read Offset Ordering



Performance Comparisons

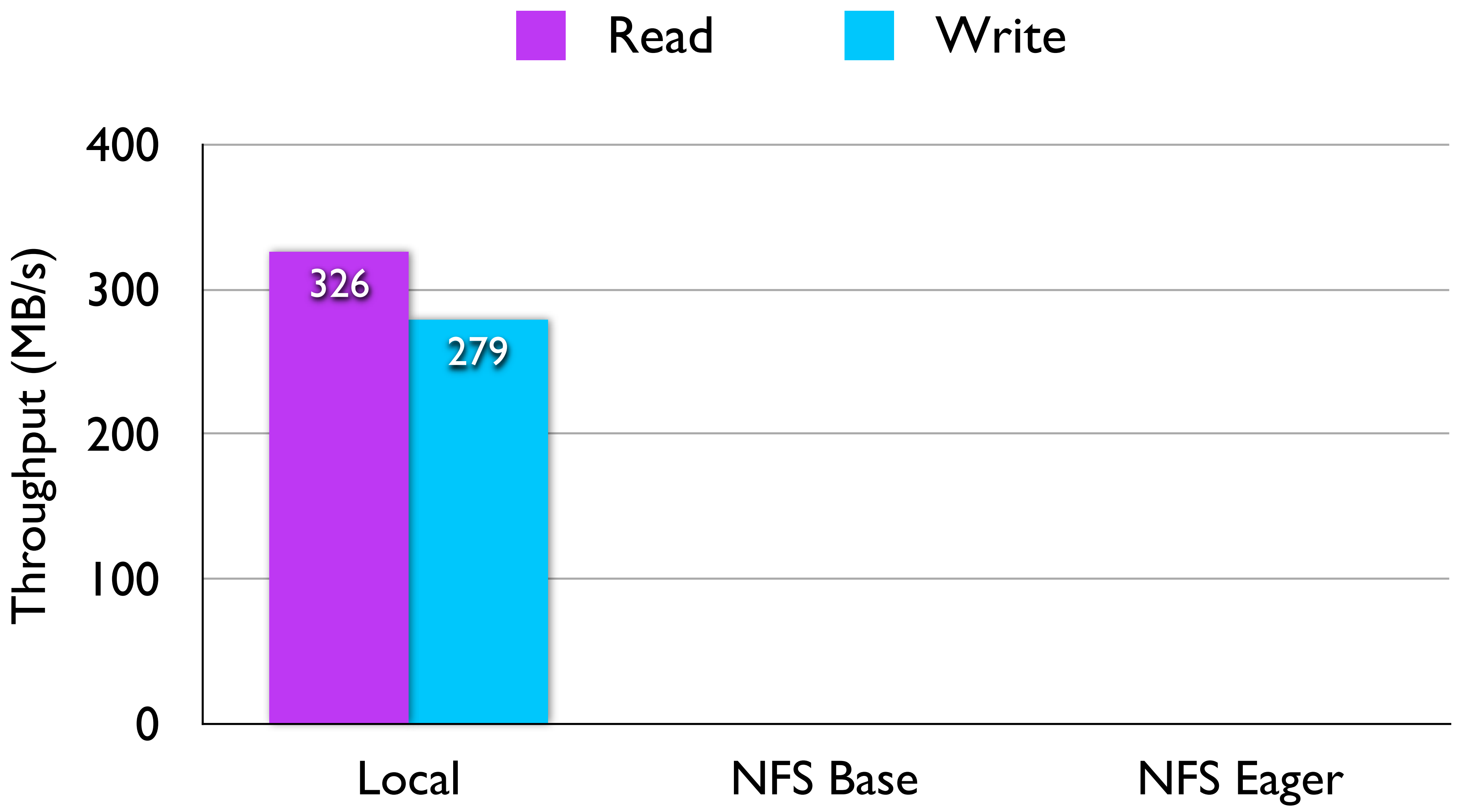
- Micro benchmarks
  - Streaming I/O
  - Random Writes
  - Non-sequential Writes
  - Adversarial Page Reuse
- Macro benchmarks
  - Filebench Fileserver
  - Filebench Videoserver

Streaming I/O Performance

🟪 Read 🟦 Write

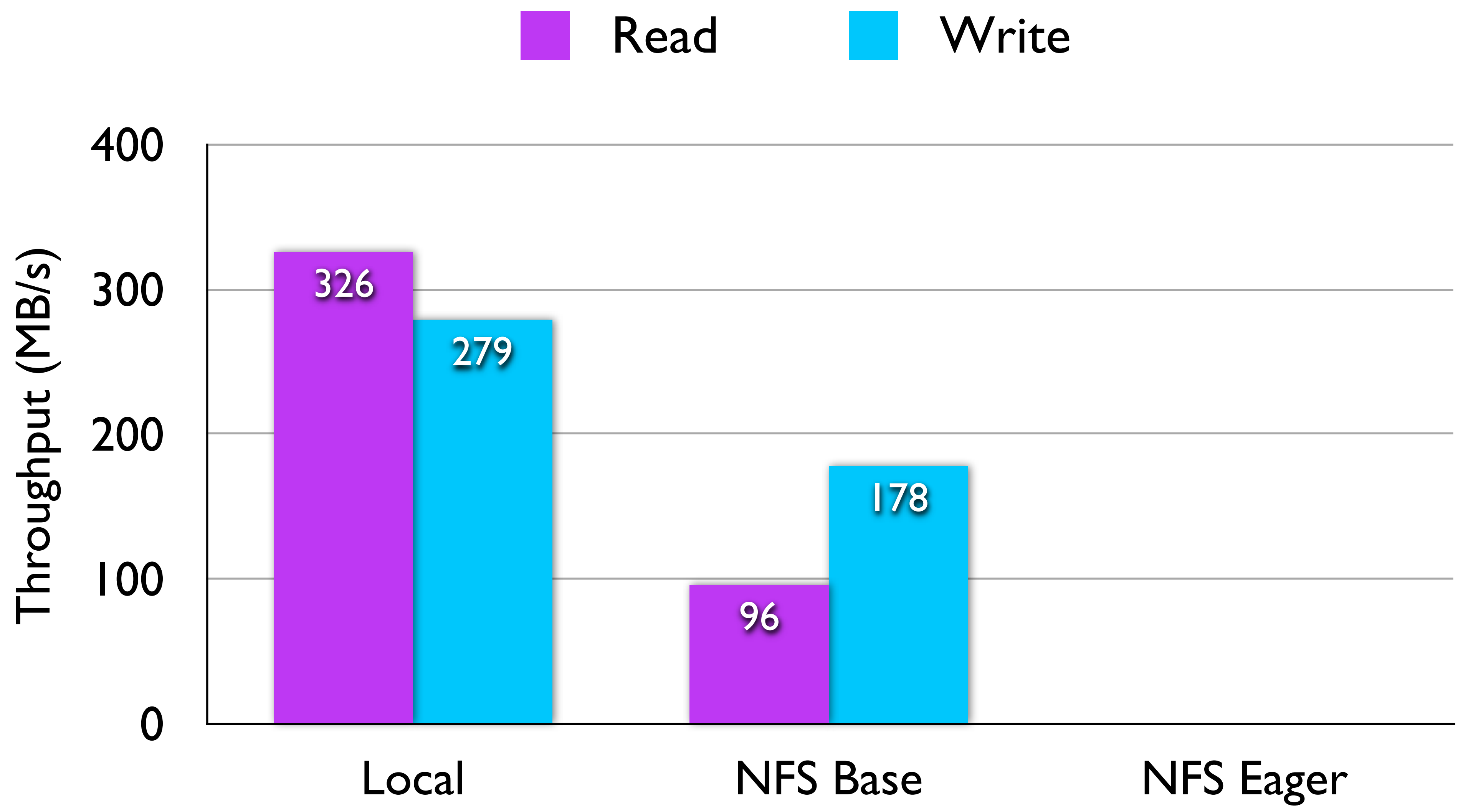
NEC Laboratories America
Relentless passion for innovation

Streaming I/O Performance

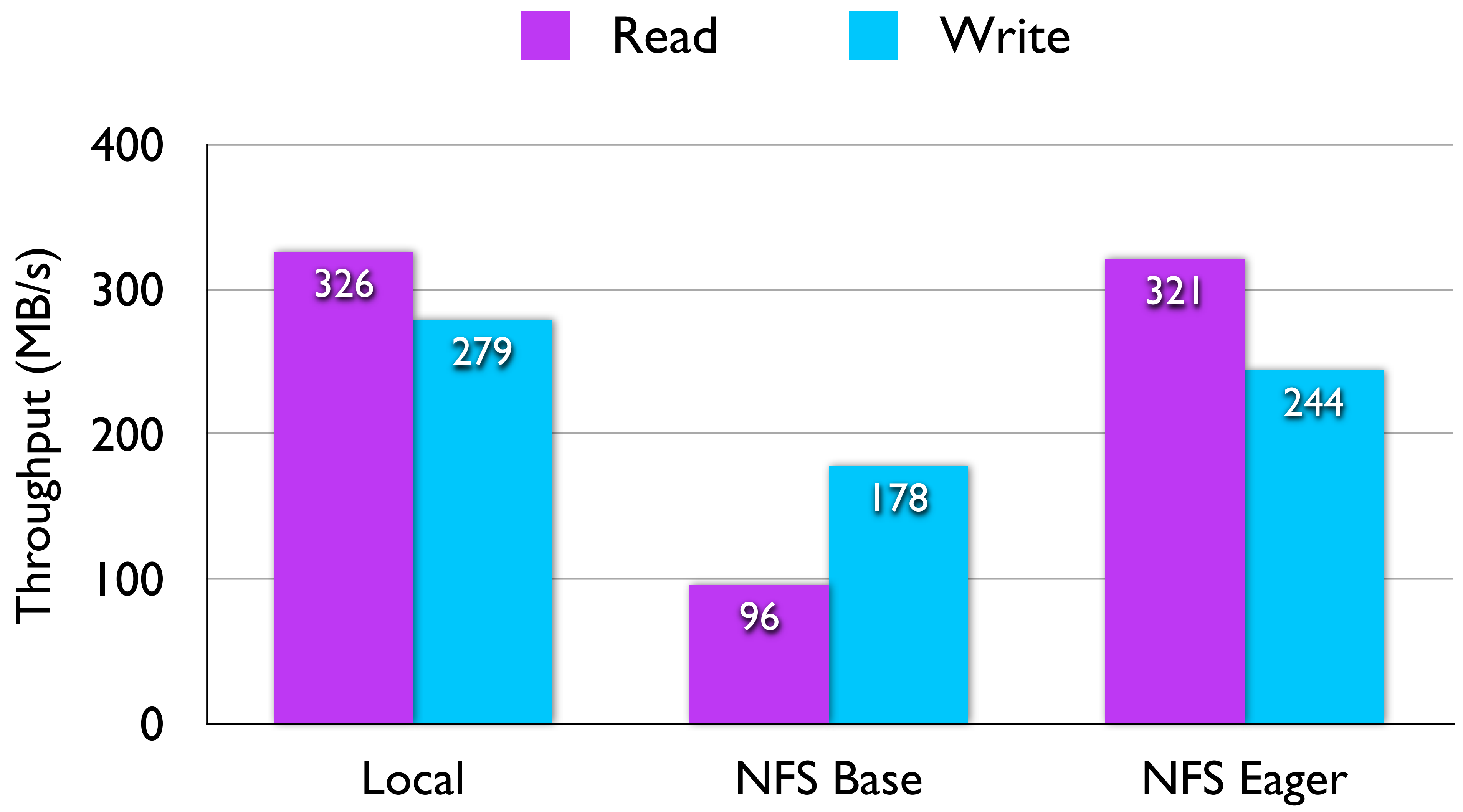


NEC Laboratories America
Relentless passion for innovation

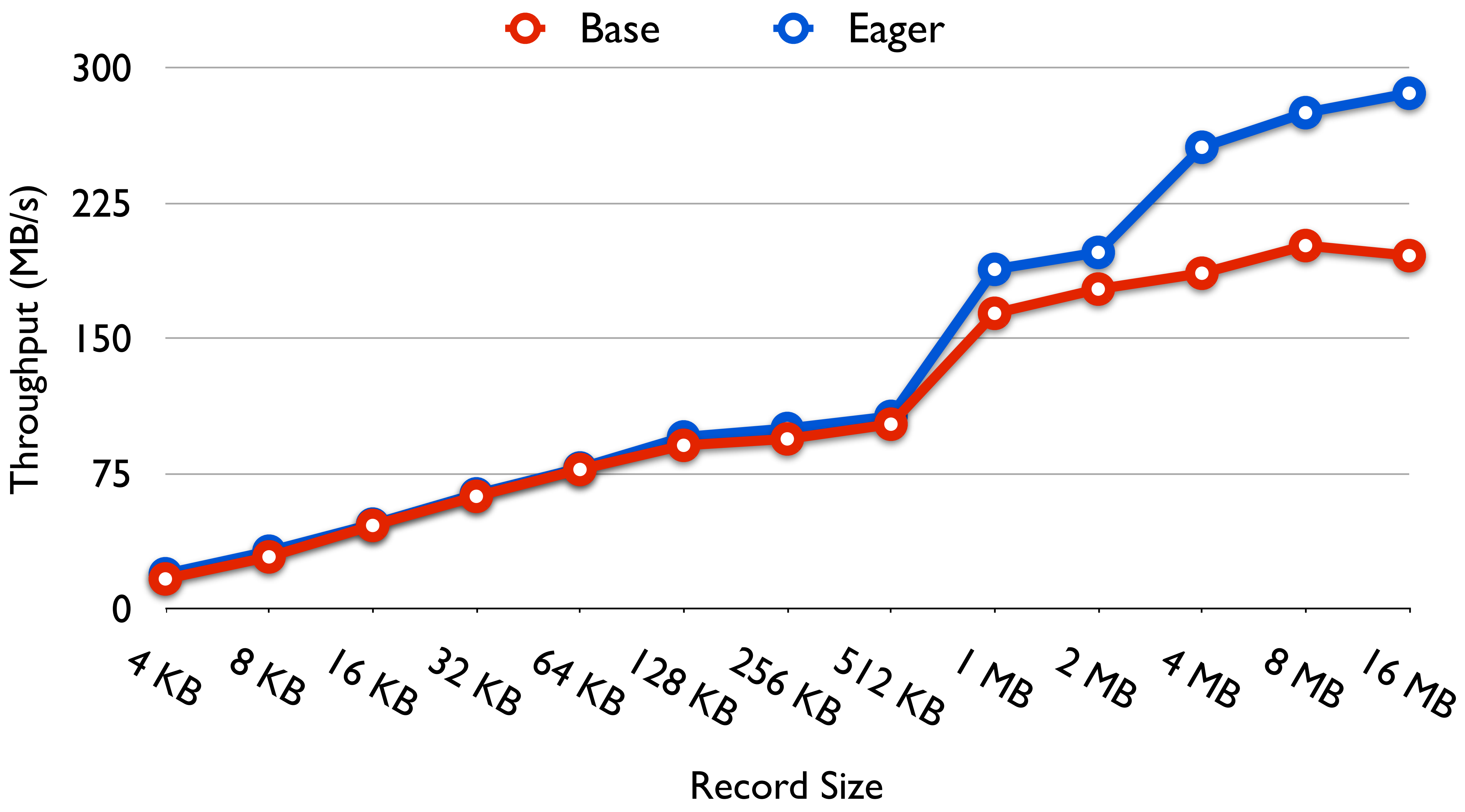
Streaming I/O Performance



Streaming I/O Performance

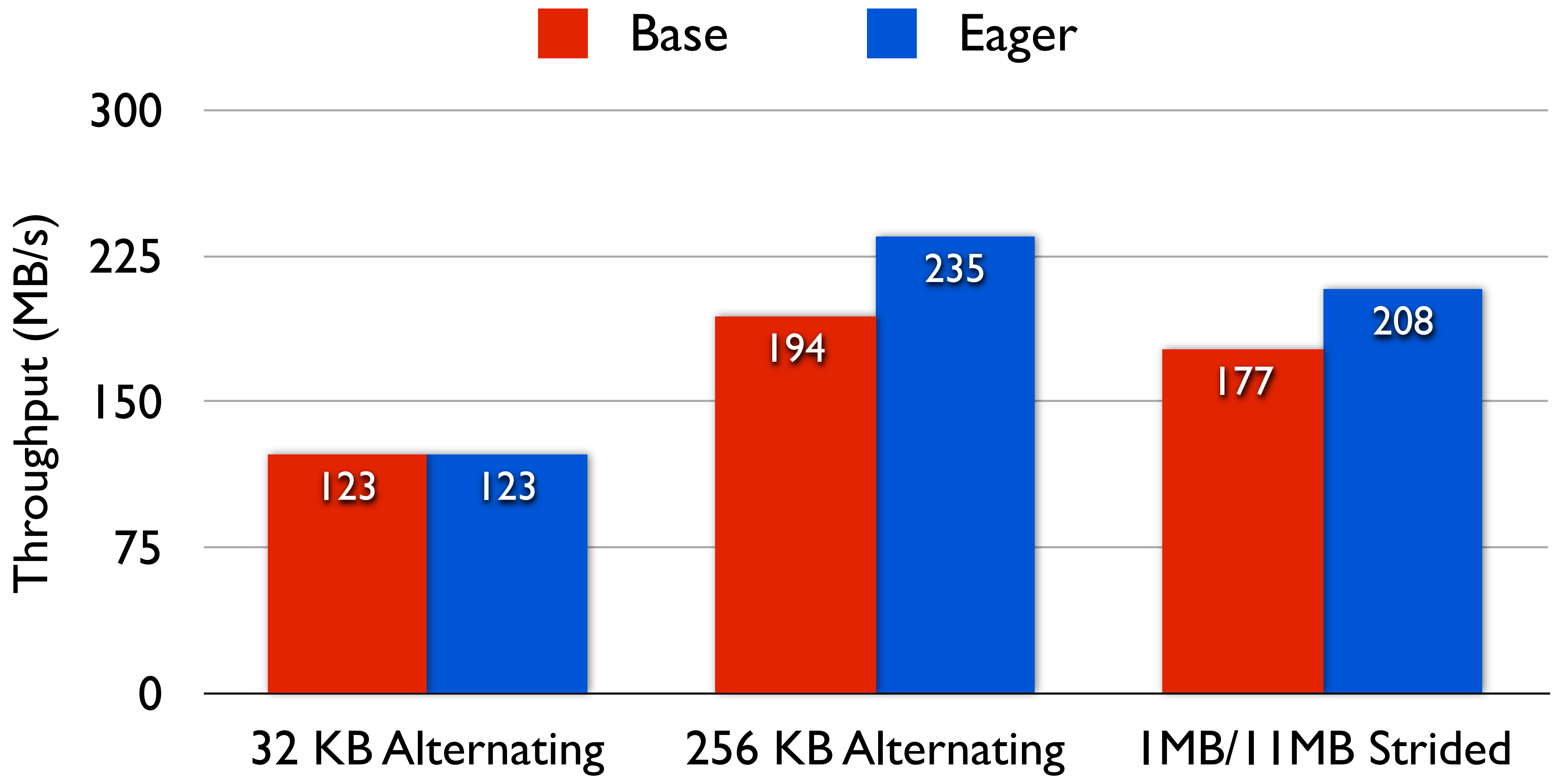


Random Write Performance



NEC Laboratories America
Relentless passion for innovation

Nonsequential Write Performance



Adversarial Example

1.5 GB --

1.4 GB -- `dirty_background_ratio`

0 GB --

Adversarial Example

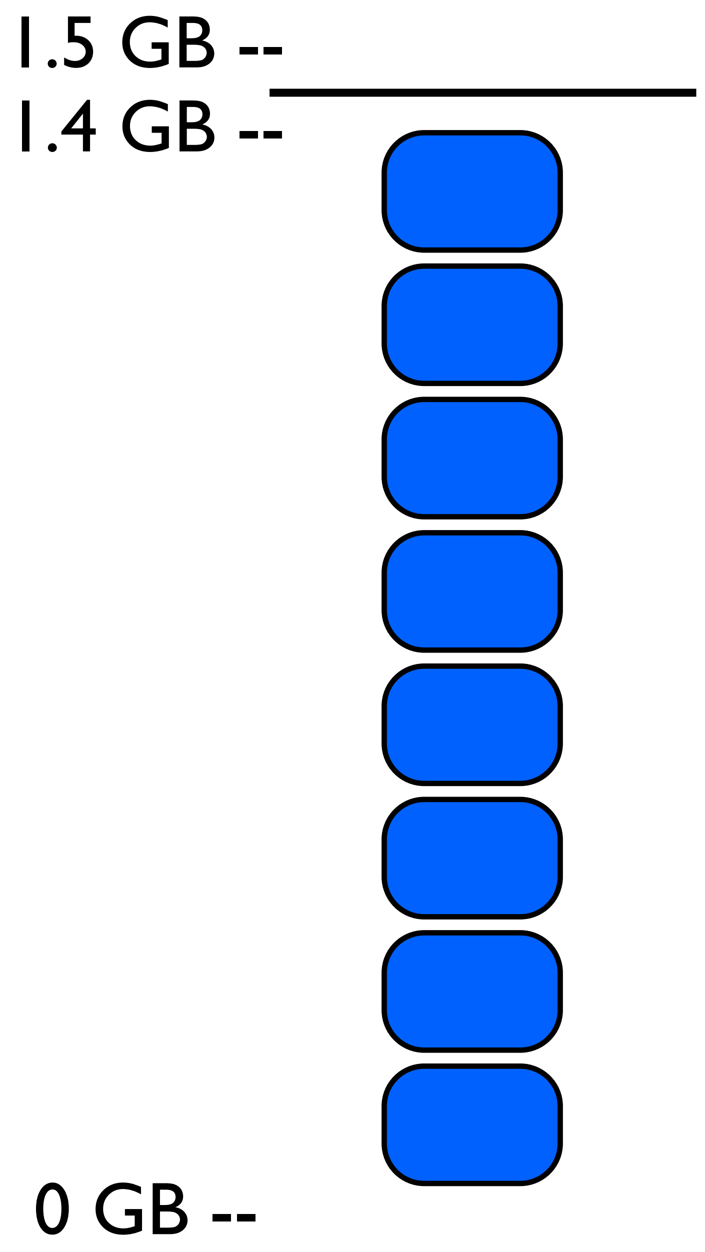
1.5 GB --
1.4 GB -- `dirty_background_ratio`

Footprint	Base	Eager

0 GB --

NEC Laboratories America
Relentless passion for innovation

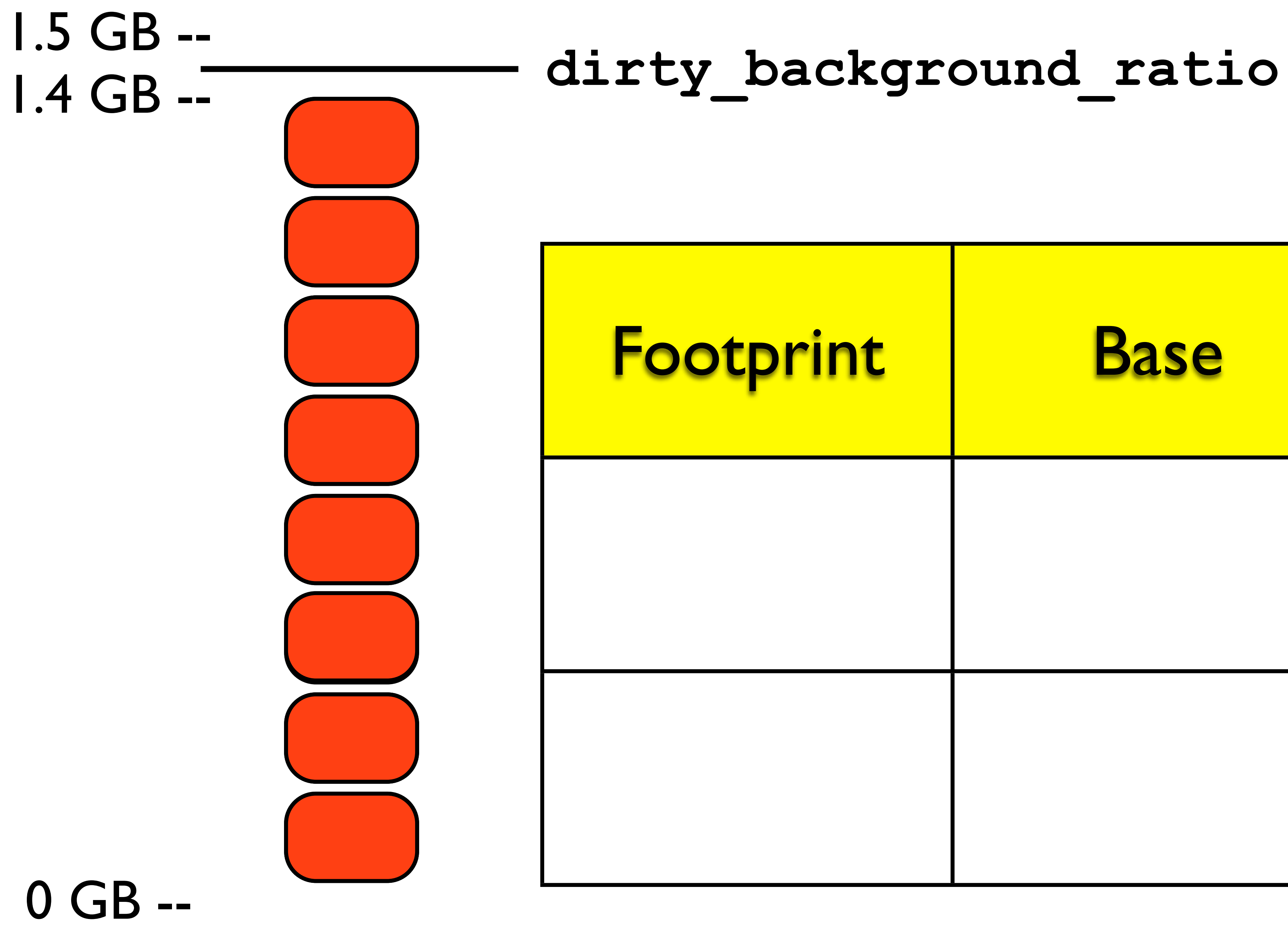
Adversarial Example



`dirty_background_ratio`

Footprint	Base	Eager

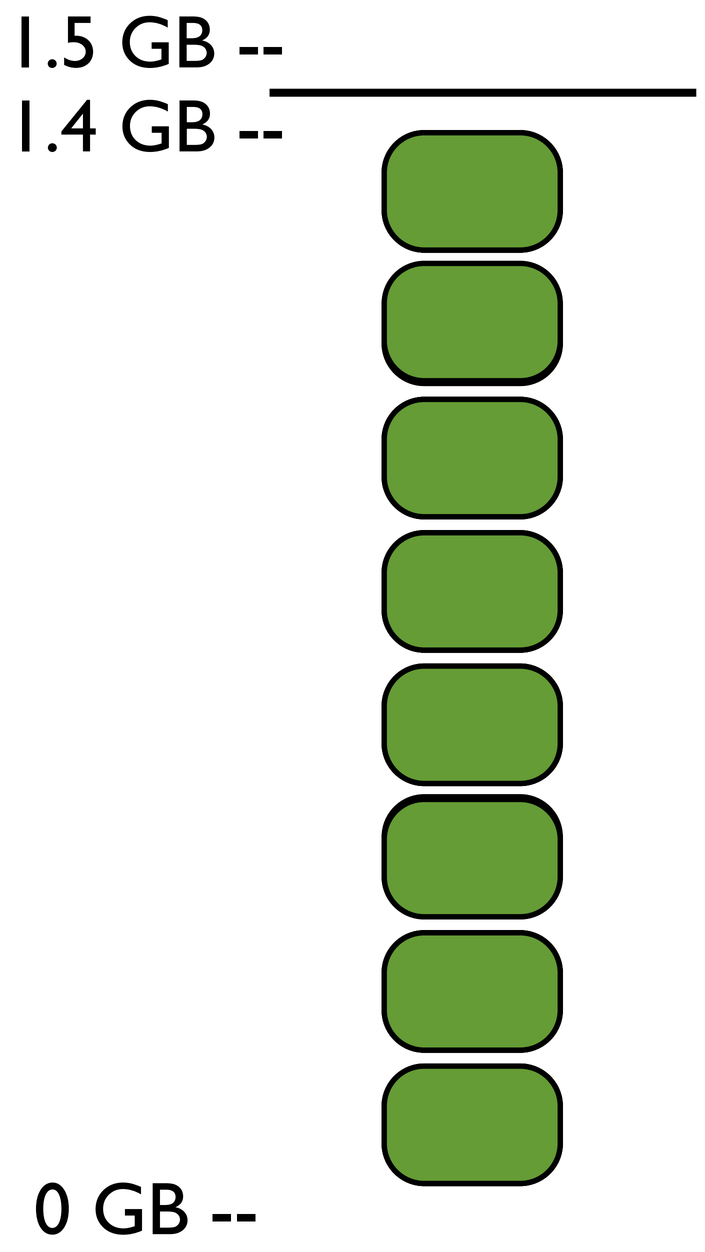
Adversarial Example



Footprint	Base	Eager

NEC Laboratories America
Relentless passion for innovation

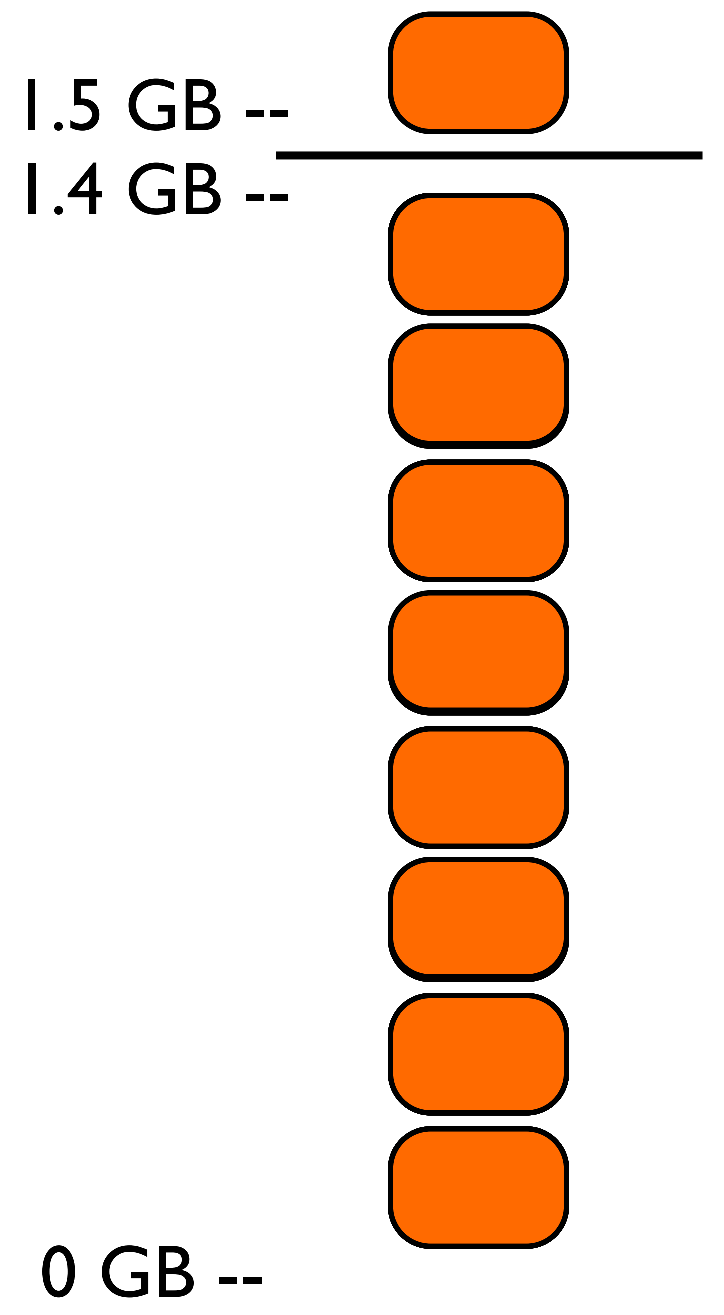
Adversarial Example



`dirty_background_ratio`

Footprint	Base	Eager
1.4 GB (32 GB total)	1093 MB/s (1.4 GB to disk)	513 MB/s (18 GB to disk)

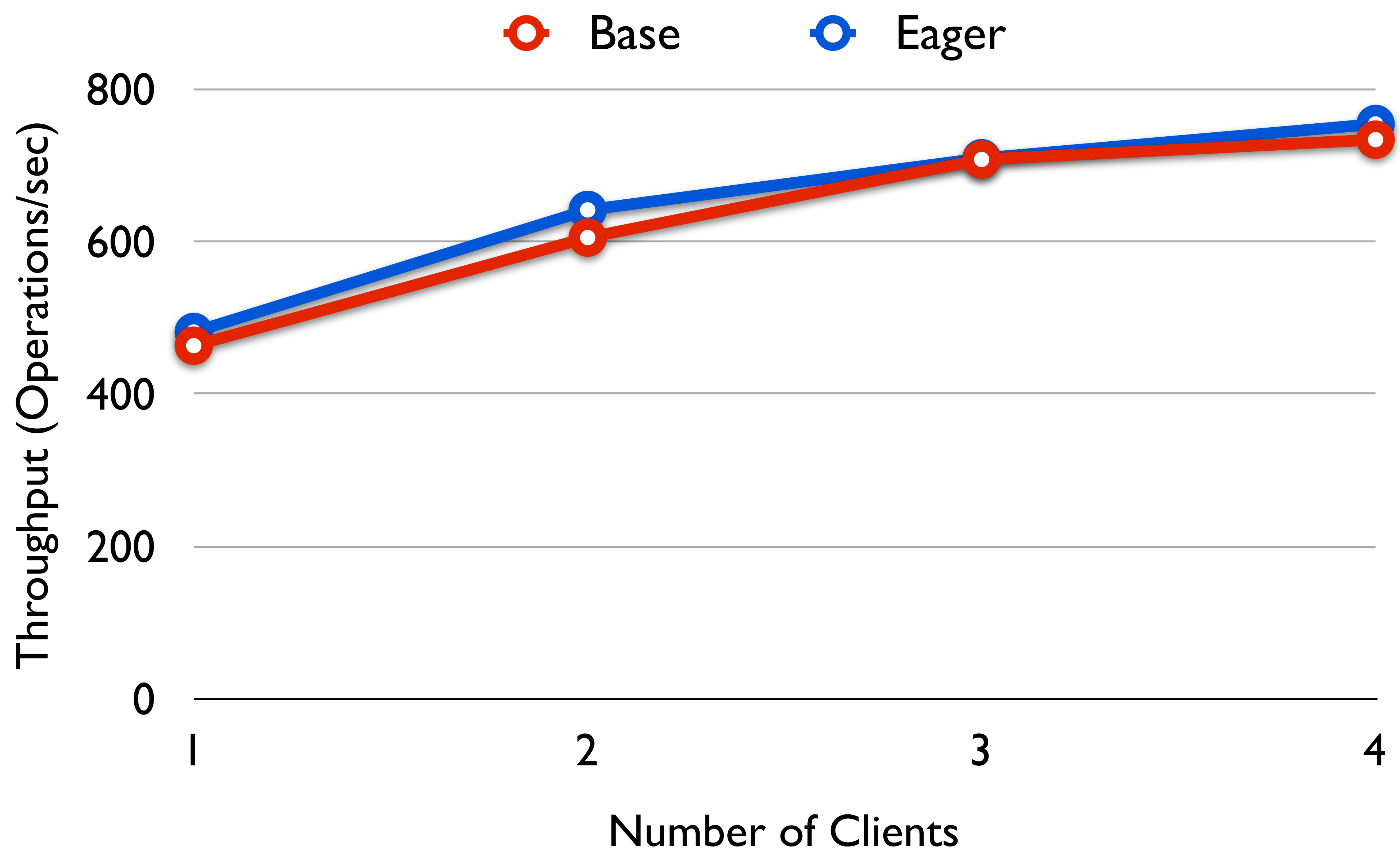
Adversarial Example



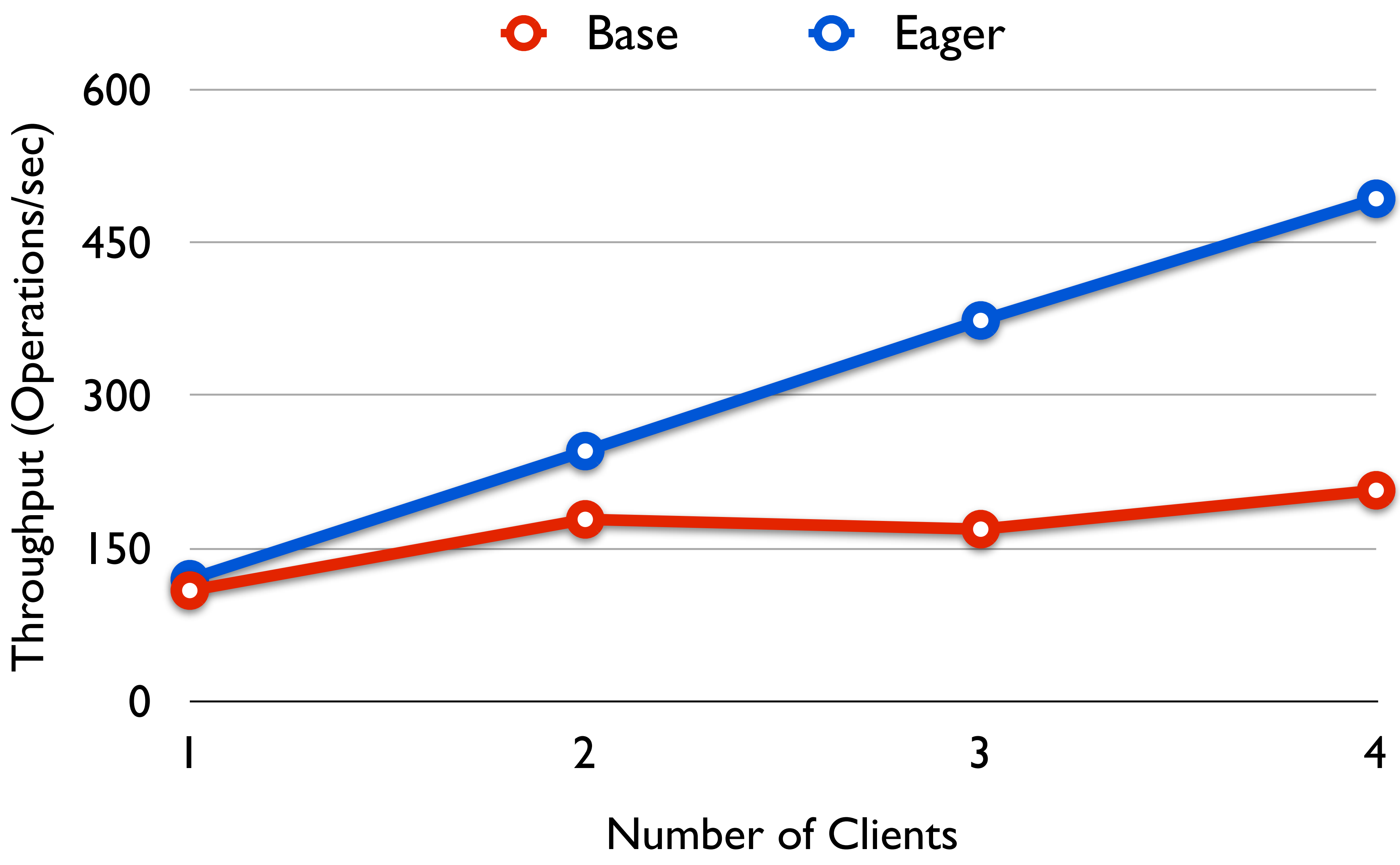
Footprint	Base	Eager
1.4 GB (32 GB total)	1093 MB/s (1.4 GB to disk)	513 MB/s (18 GB to disk)
1.5 GB (32 GB total)	469 MB/s (18 GB to disk)	253 MB/s (32 GB to disk)

NEC Laboratories America
Relentless passion for innovation

Filebench Fileserver Workload



Filebench Videoserver Workload



Implementation

Technique	Scope	Lines of Code
Eager Writeback	NFS Client	100
Eager Page Laundering	NFS Client & Server	150
Request Ordering	NFS Server	120

Related Work

- Lee, et al. 2000
  Eager Writeback - A Technique for Improving Bandwidth Utilization
  (33rd ACM/IEEE Symposium on Microarchitecture)
- Ellard & Seltzer 2003
  NFS Tips and Benchmarking Traps (USENIX ATC)
- Ellard, et al. 2003
  Passive NFS Tracing of Email and Research Workloads (FAST '03)
- Batsakis, et al. 2009
  CA-NFS: a Congestion-Aware Network File System (FAST '09)

Summary

- For writes, memory pressure leads to performance problems
- For reads, out-of-order requests disable read-ahead
- Eager writeback, eager page laundering, and request ordering improve sequential throughput
- No harm for many nonsequential workloads
  - May even improve throughput when clients experience memory pressure