Rapid Prototyping and Evaluation of Intelligence Functions of Active Storage Devices

Yongsoo Joo

Embedded Software Research Center

Ewha Womans University

This research was supported by Basic Science Research Program through NRF (2012-0003366)

Active Storage Device (ASD)

- Key idea
  - Offload computation (data processing) to the storage device
- A more general definition
  - Storage devices that actively perform “**something**” more than just handling the I/O requests that they receive
  - Goal: to improve storage performance
- We call “**something**” intelligence functions

Intelligence Function (IF)

- Application-specific intelligence functions
  - Query operations in database systems
  - Data mining for multimedia applications
  - Gene sequence matching in biological data
- Object storage devices (OSDs)
  - Support various types of applications and IFs
  - Object are managed by the storage device
  - Cf.) conventional systems: object -> file -> LBA -> PBA

Requirement of OSDs

- A new, innovative I/O interface
  - OSD SCSI T10 specification (implemented over iSCSI)
- OS kernel support
  - Support for the OSD protocol added in Linux 2.6.30
- A new programming model for applications
  - Stream based, RPC based, etc.
- Technically feasible, but facing difficulty in practice

Difficulties in Deployment

- **Researchers**: hard to set up an evaluation platform
  - ASDs not available as commodity hardware
  - Applications should be ASD-aware as well
- **Manufacturers**: need confidence before migration to ASDs
  - Find good applications (with intelligence functions)
  - Feedback from user experience
- **Users**: hard to gain user experiences
  - Users have little way to experience ASD-based systems

Chicken-and-egg problem!

Alternative Way

- What about intelligence functions compatible with commodity systems?
- Some IFs can be implemented on a file system
  - MVSS (multi-view storage systems), QuFiles, etc.
- Modern HDDs and SSDs have potential to be an ASD
  - Lookahead read, data deduplication, etc.
- Less flexible, but immediately deployable

File-based Intelligence Functions

- Intelligence functions running at file level
  - Multiple views of a file (e.g., a video clip at various resolutions)
  - Context-aware adaptation
- How to evaluate?
  - Implement a new file system from scratch
  - Stackable file system (e.g., FUSE)

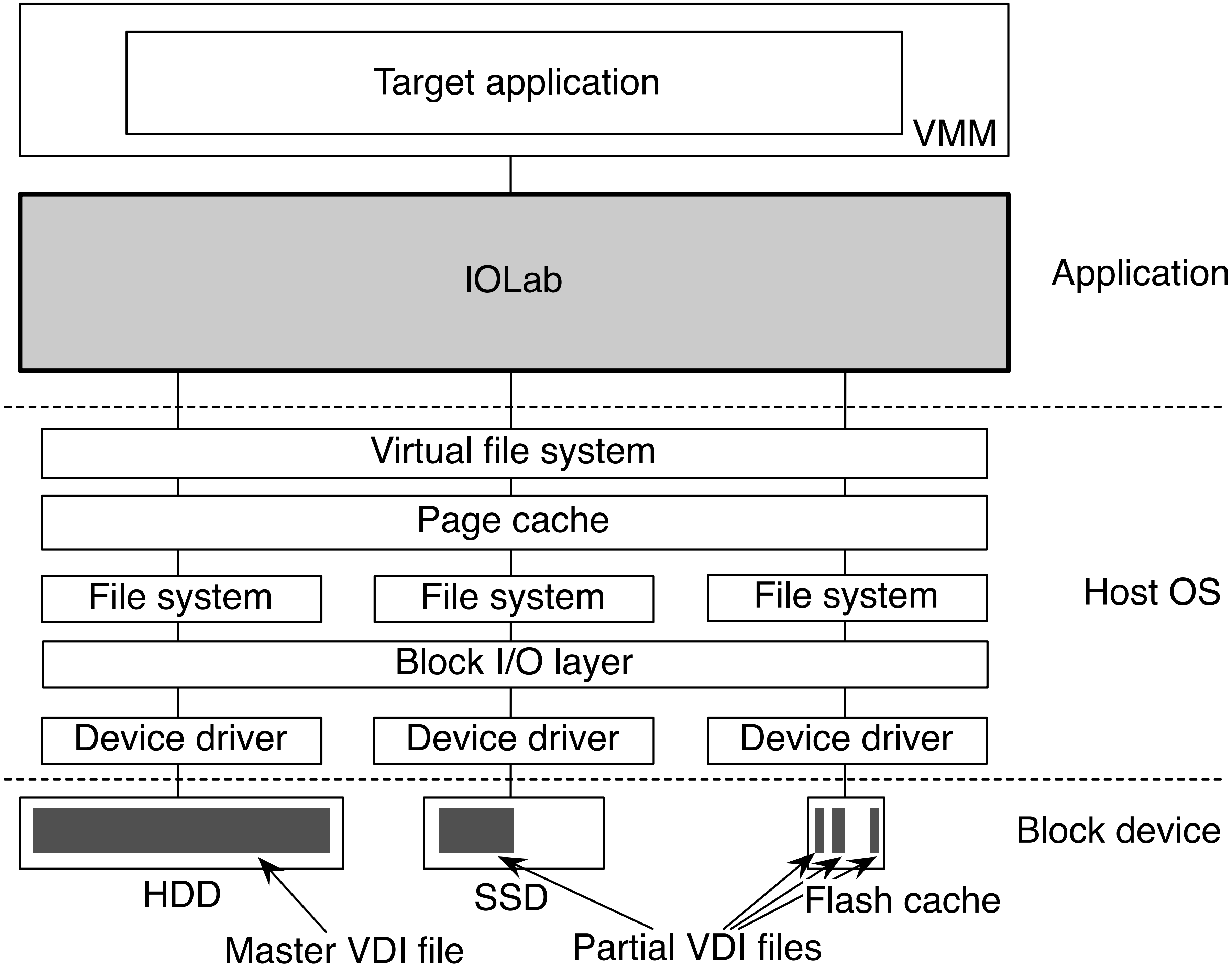
Block-based Intelligence Functions

- Intelligence functions running at block level
  - Prefetching / hot data clustering / block replication
  - Data pinning / NVRAM write cache / block deduplication
- How to evaluate?
  - Block device simulation (e.g., disksim)
  - Hack the OS block layer
  - No tool like FUSE for block-based IFs

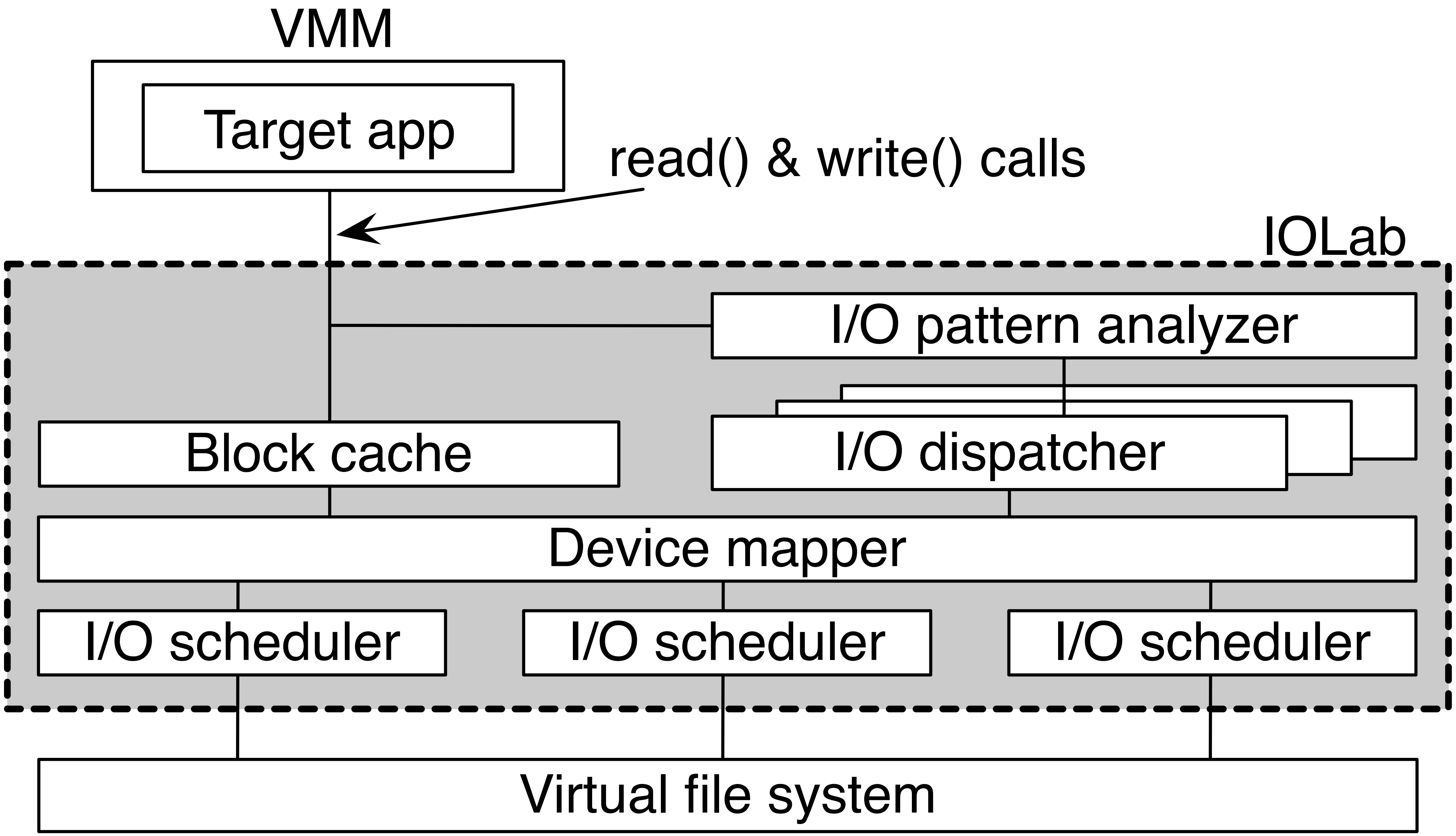
Proposed Evaluation Platform

- **IOLab**: A VM-based evaluation platform for ASDs
- The role of the VM
  - Run target applications to generate input I/O requests
- Key Idea
  - Intercept I/O requests between the VM and the host OS
- Implementation
  - A userspace module running on the host OS

Structure of IOLab



Structure of IOLab



Advantage of IOLab

- Easy prototyping of intelligence functions
  - No customized hardware
  - No need to hack the OS kernel
- Real-time execution
  - IFs are running on real HDDs or SSDs
  - Immediate benefit to VM users
- Extensibility
  - Able to use any block device attachable to the host machine
  - Easy to combine heterogeneous block devices

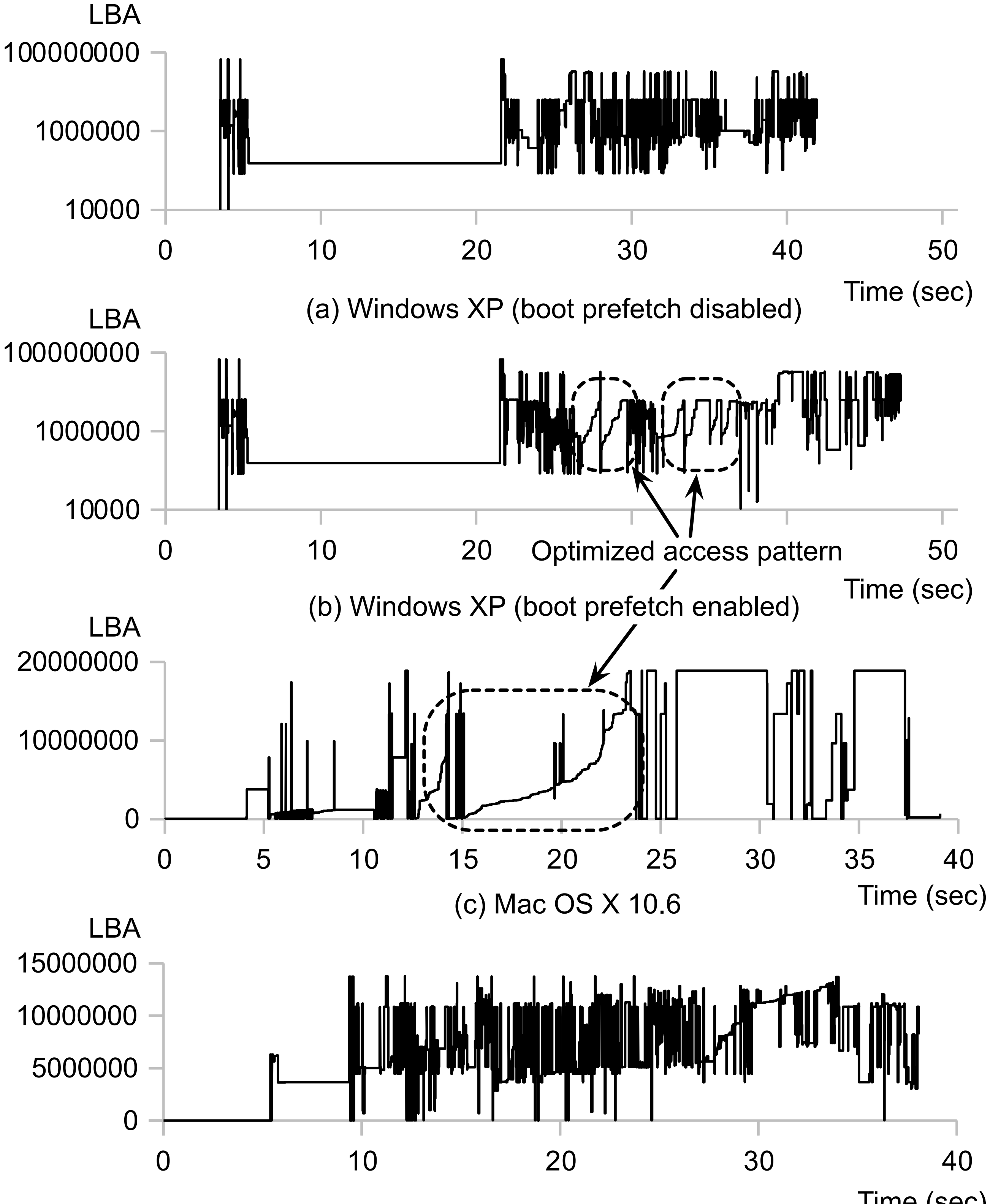
OS Boot Observation

Windows XP

Windows XP
(with boot prefetch)

Mac OS X

Linux Fedora



(a) Windows XP (boot prefetch disabled)

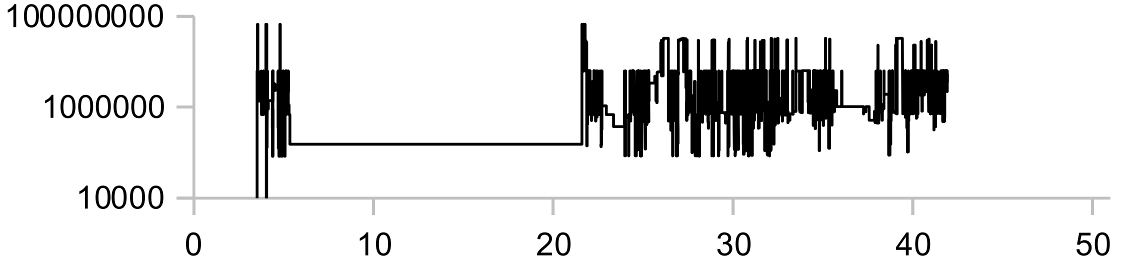
(b) Windows XP (boot prefetch enabled)

(c) Mac OS X 10.6

(d) Linux Fedora 14 x64

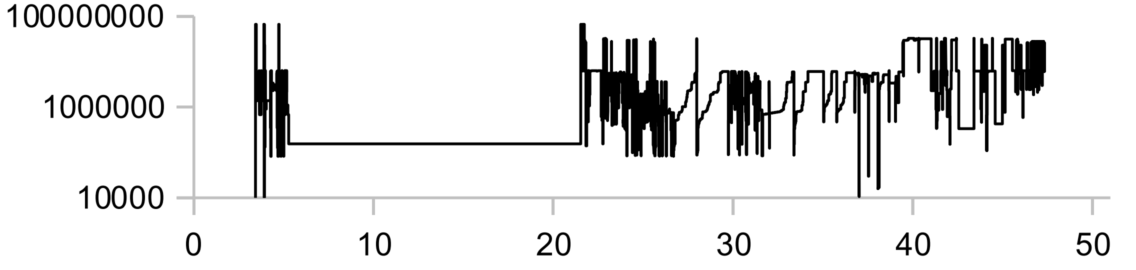
OS Boot Optimization (Windows XP)

No prefetcher (baseline)



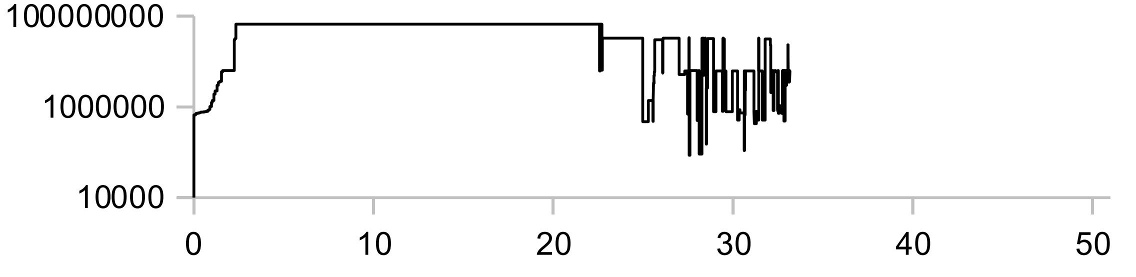
(a) No prefetcher

Built-in boot prefetcher



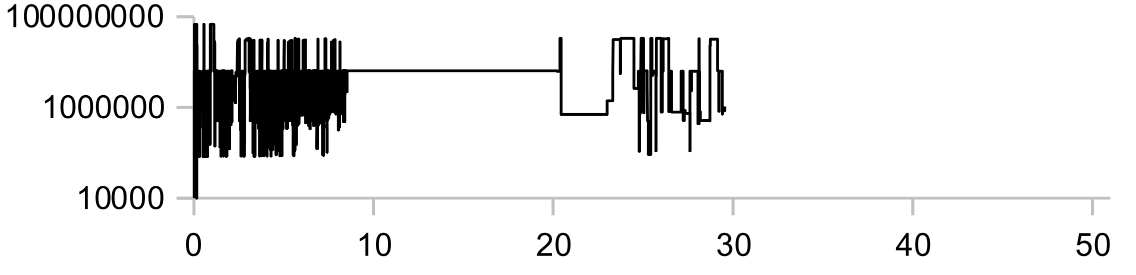
(b) Windows prefetcher

IOLab prefetcher
(sorted by LBA)



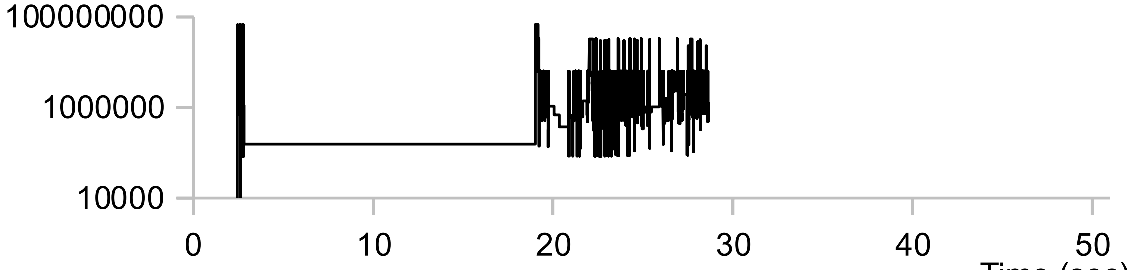
(c) IOLab prefetcher (sorting)

IOLab prefetcher
(keep the LBA order)



(d) IOLab prefetcher (CDP)

Warm start
(100% hit on the page cache of
the host OS)



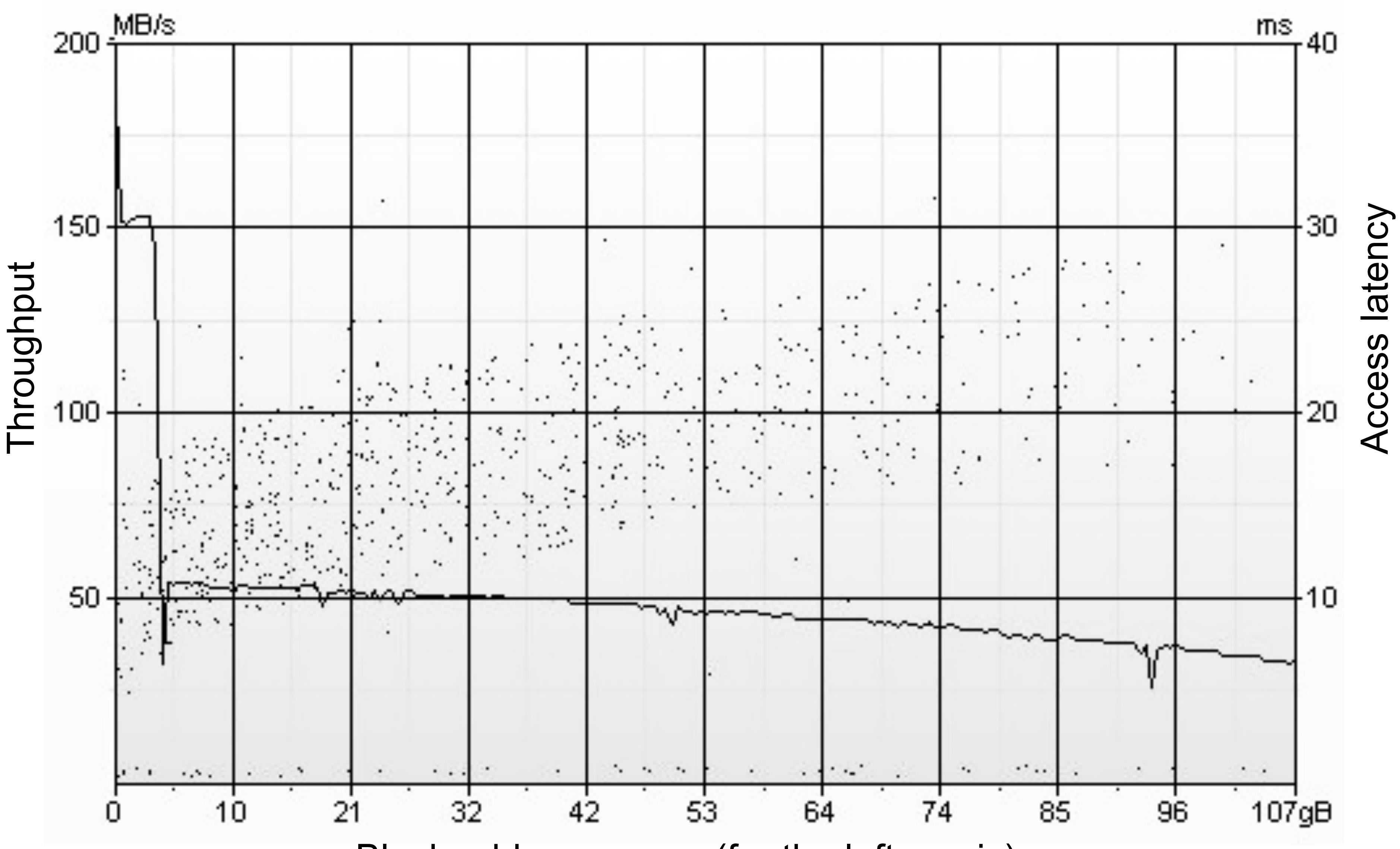
(e) Warm start

Hybrid Disk

- Rapid prototyping of a hybrid disk
  - Combination of commodity block devices
- SSD+HDD hybrid disk
  - SSD: Intel X25-V (40GB MLC)
  - HDD: Fujitsu MHZ2120BH (120GB, 2.5")
- Block mapping
  - First 4GB mapped to the SSD
  - The rest to the HDD

Hybrid HDD

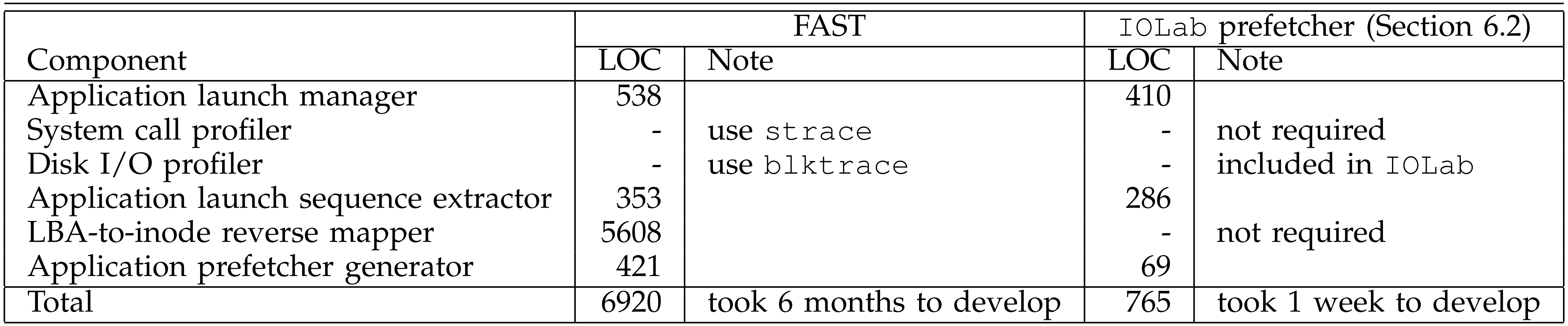
- Measured throughput and latency
  - HD Tune Pro (a HDD benchmarking tool running on Windows OS)



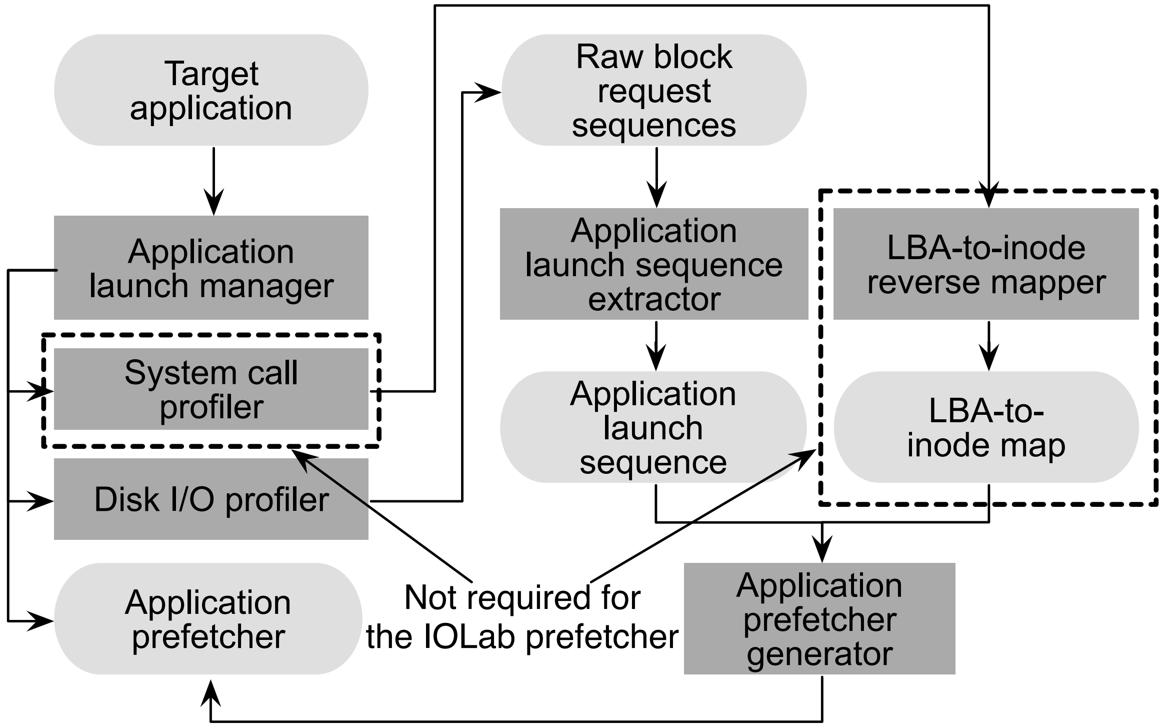
Block address space (for the left y-axis)
Distance btw. adjacent I/O requests (for the right y-axis)

Prototyping Effort

- Real implementation vs. IOLab
  - Target IF: application prefetcher



	FAST		IOLab prefetcher (Section 6.2)	
Component	LOC	Note	LOC	Note
Application launch manager	538		410	
System call profiler	-	use strace	-	not required
Disk I/O profiler	-	use blktrace	-	included in IOLab
Application launch sequence extractor	353		286	
LBA-to-inode reverse mapper	5608		-	not required
Application prefetcher generator	421		69	
Total	6920	took 6 months to develop	765	took 1 week to develop

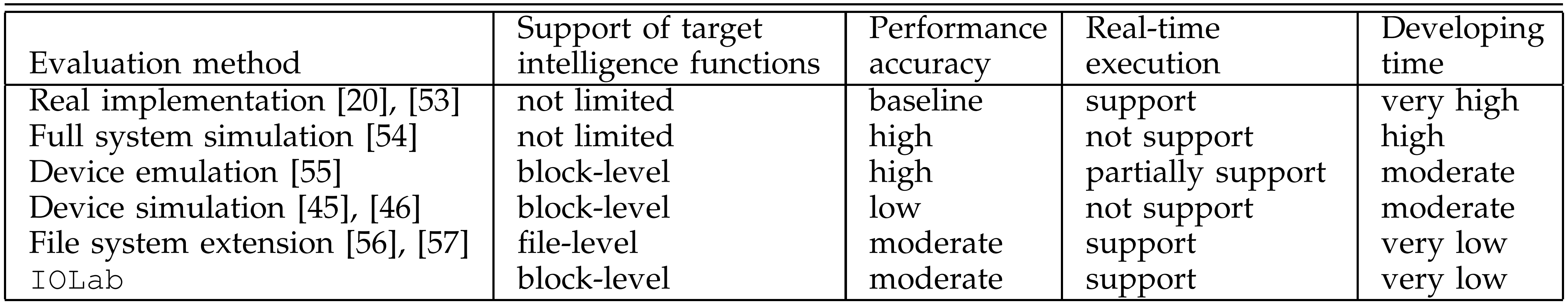


Target application
Application launch manager
System call profiler
Disk I/O profiler
Application prefetcher
Raw block request sequences
Application launch sequence extractor
Application launch sequence
LBA-to-inode reverse mapper
LBA-to-inode map
Application prefetcher generator
Not required for the IOLab prefetcher

Summary

- IOLab supports rapid prototyping of block-based intelligence functions
- Once a new IF is confirmed to be effective on IOLab, we can move to the next step without much risk

Comparison with other prototyping methods



Evaluation method	Support of target intelligence functions	Performance accuracy	Real-time execution	Developing time
Real implementation [20], [53]	not limited	baseline	support	very high
Full system simulation [54]	not limited	high	not support	high
Device emulation [55]	block-level	high	partially support	moderate
Device simulation [45], [46]	block-level	low	not support	moderate
File system extension [56], [57]	file-level	moderate	support	very low
IOLab	block-level	moderate	support	very low

Q&A