LinnOS
Predictability on Unpredictable Flash Storage with a Light Neural Network

Mingzhe Hao, Levent Toksoz, Nanqinqin Li, Edward Edberg Halim, Henry Hoffmann, and Haryadi S. Gunawi

THE UNIVERSITY OF CHICAGO
CERES Center for Unstoppable Computing
SURYA UNIVERSITY

1

Doubling data every ***2 years***
30 zettabytes in 2018
175 zettabytes in 2025

Large-volume data with **microsecond-level** access latencies!

15% annual growth rate.
47 to 80 billion market by 2025!

30% data access will be ***real-time by 2025***

More **real-time** data? More **flash storage**!

SSD

[1] Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018
[2] Solid State Drive (SSD) Market – Growth, Trends, and Forecasts (2020 - 2025), mordorintelligence.com

2

Do these SSDs solve real-time data access challenge?

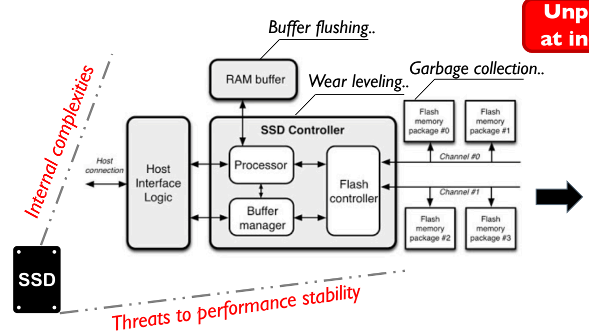
2014 "Most of the interesting devices that we are dealing with are in the microsecond-level, and we suck at microsecond-level"
[Google Research: Three things that MUST BE DONE to save the data center of the future]

Destructive latency interruptions inside flash storage devices

2017 "A new breed of I/O devices motivates greater interest in microsecond-scale latencies, and new technologies are needed"
[Attack of the Killer Microseconds, CACM]

3

Unpredictable Latency

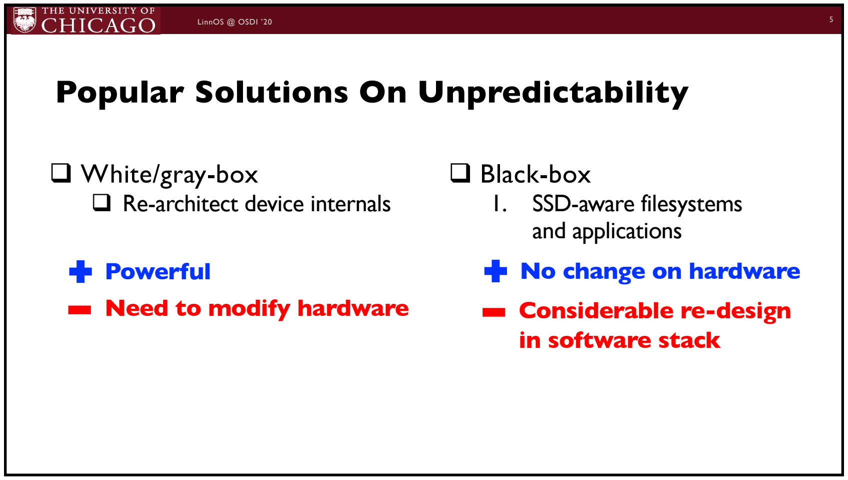


Unpredictable latency at individual-I/O level!

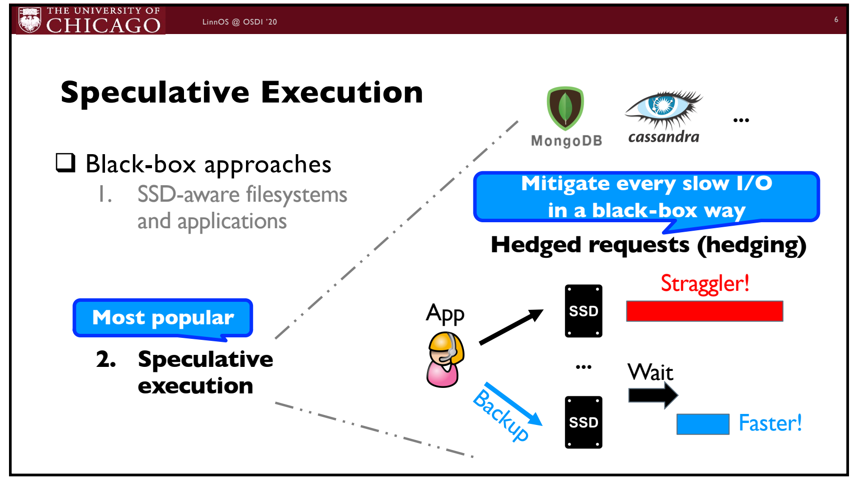
SSD ... SSD

1st I/O	100µs	120µs
2nd I/O	80µs	150µs
3rd I/O	110µs	800µs
4th I/O	1.5ms	200µs
⋮	⋮	⋮

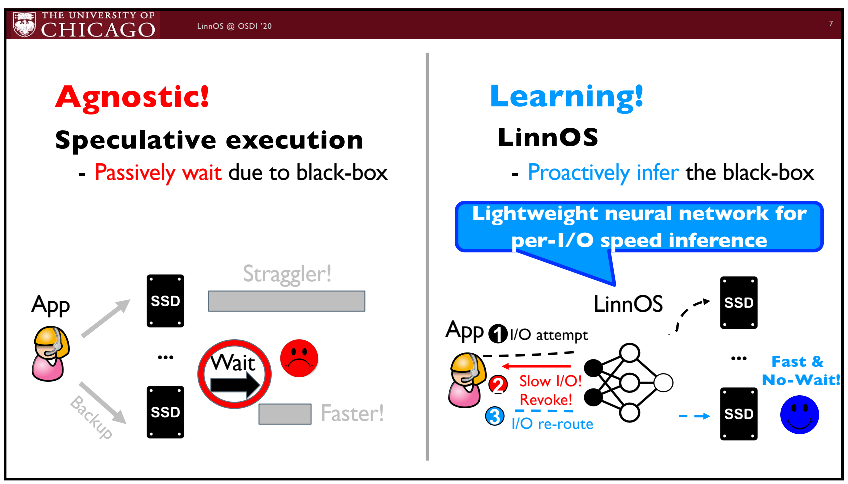
4


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
Popular Solutions On Unpredictability
White/gray-box
Re-architect device internals
Powerful
Need to modify hardware
Black-box
1. SSD-aware filesystems and applications
No change on hardware
Considerable re-design in software stack

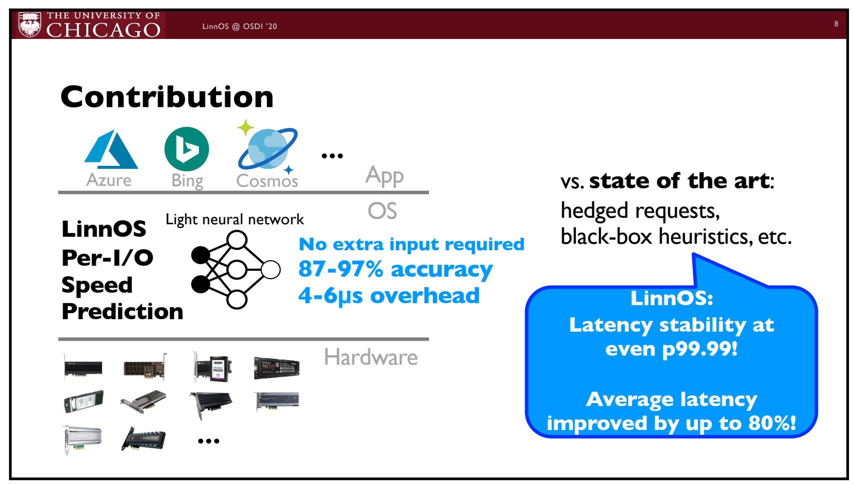
5


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
Speculative Execution
Black-box approaches
1. SSD-aware filesystems and applications
Most popular
2. Speculative execution
MongoDB
cassandra
...
Mitigate every slow I/O in a black-box way
Hedged requests (hedging)
App
SSD
Straggler!
...
Wait
Backup
SSD
Faster!

6


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
Agnostic!
Speculative execution
- Passively wait due to black-box
App
SSD
Straggler!
...
Wait
Backup
SSD
Faster!
Learning!
LinnOS
- Proactively infer the black-box
Lightweight neural network for per-I/O speed inference
LinnOS
App
1 I/O attempt
2 Slow I/O! Revoke!
3 I/O re-route
SSD
...
Fast & No-Wait!
SSD

7


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
Contribution
Azure
Bing
Cosmos
...
App
OS
LinnOS Per-I/O Speed Prediction
Light neural network
No extra input required
87-97% accuracy
4-6μs overhead
Hardware
...
vs. state of the art:
hedged requests,
black-box heuristics, etc.
LinnOS:
Latency stability at even p99.99!
Average latency improved by up to 80%!

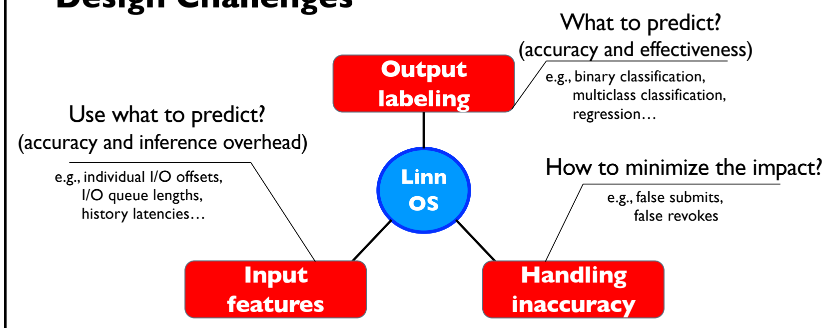
8

Outline

- ❑ Introduction
- ❑ **Challenges & Solutions**
- ❑ Evaluations
- ❑ Conclusion

9

Design Challenges



10

Output labeling

Ideal labels:

Only 60-70% accuracy

Exact latency value
(e.g. 120μs, 80μs..) ➕ **Flexible**

Latency ranges
(e.g., 100-200μs,
200-400μs)

➖ **Difficult to achieve decent accuracy**

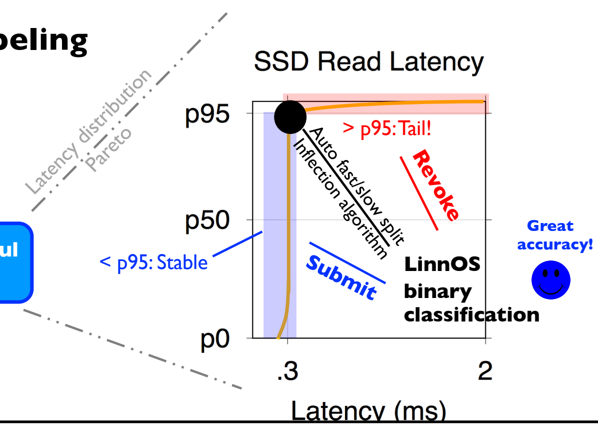
50-100μs
100-200μs **Mis-Prediction**
200-400μs **Truth**
…

11

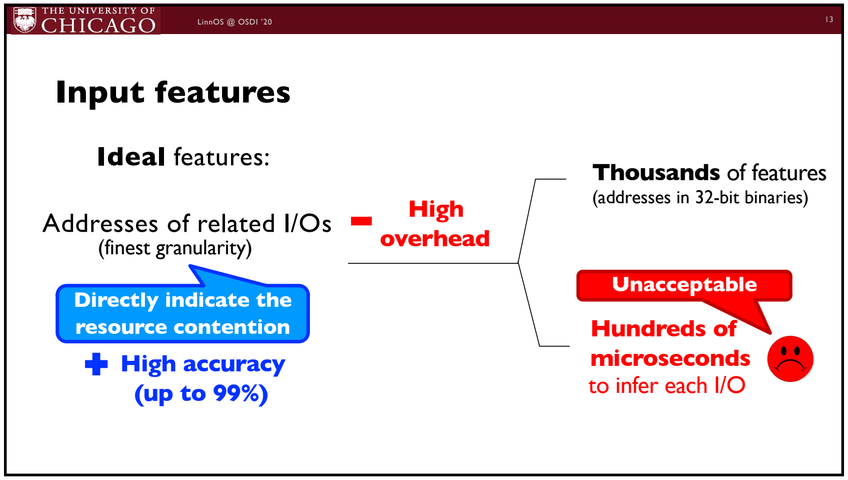
Output labeling

Precise prediction **is difficult!**

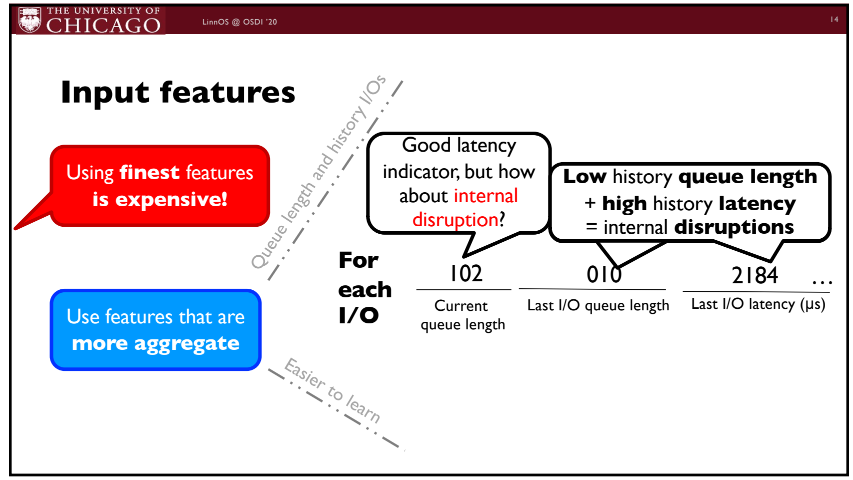
Simpler and **helpful** alternatives?



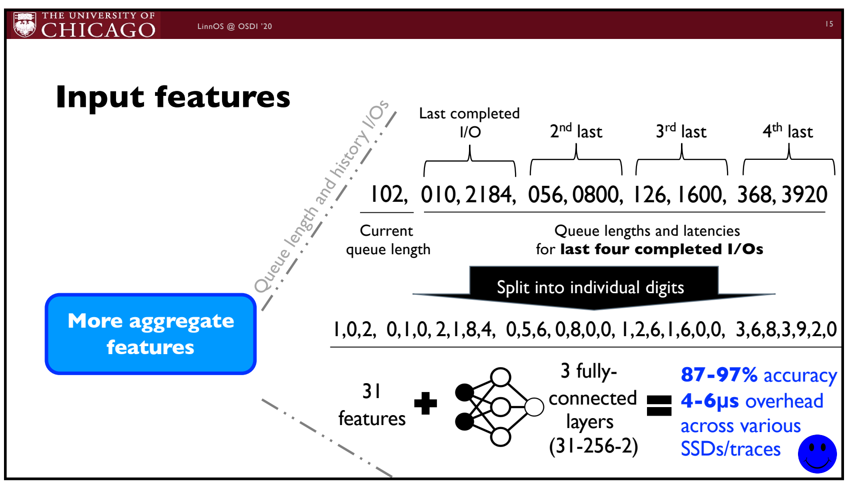
12


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
13
Input features
Ideal features:
Addresses of related I/Os
(finest granularity)
Directly indicate the resource contention
High accuracy (up to 99%)
High overhead
Thousands of features
(addresses in 32-bit binaries)
Unacceptable
Hundreds of microseconds to infer each I/O

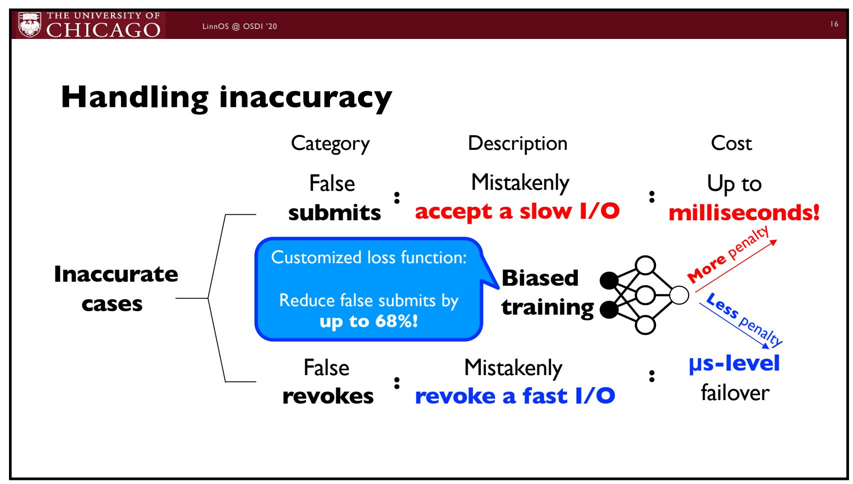
13


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
14
Input features
Using finest features is expensive!
Use features that are more aggregate
Queue length and history I/Os
Easier to learn
Good latency indicator, but how about internal disruption?
Low history queue length + high history latency = internal disruptions
For each I/O
102
Current queue length
010
Last I/O queue length
2184
Last I/O latency (μs)
…

14

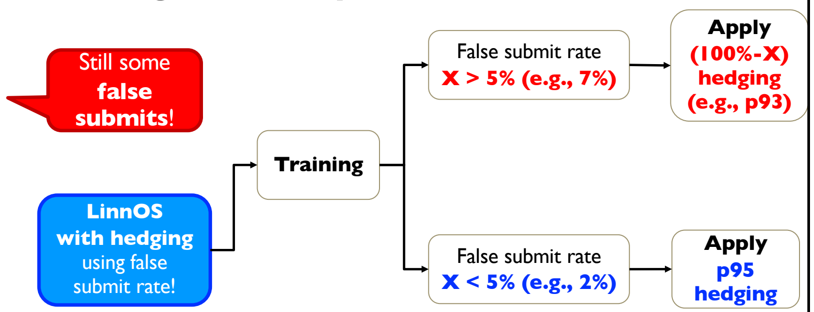

THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
15
Input features
Queue length and history I/Os
Last completed I/O
2nd last
3rd last
4th last
102, 010, 2184, 056, 0800, 126, 1600, 368, 3920
Current queue length
Queue lengths and latencies for last four completed I/Os
Split into individual digits
1,0,2, 0,1,0, 2,1,8,4, 0,5,6, 0,8,0,0, 1,2,6,1,6,0,0, 3,6,8,3,9,2,0
More aggregate features
31 features
3 fully-connected layers (31-256-2)
87-97% accuracy
4-6μs overhead
across various SSDs/traces

15


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
16
Handling inaccuracy
Category
Description
Cost
Inaccurate cases
False submits
Mistakenly accept a slow I/O
Up to milliseconds!
Customized loss function:
Reduce false submits by up to 68%!
Biased training
More penalty
Less penalty
False revokes
Mistakenly revoke a fast I/O
μs-level failover

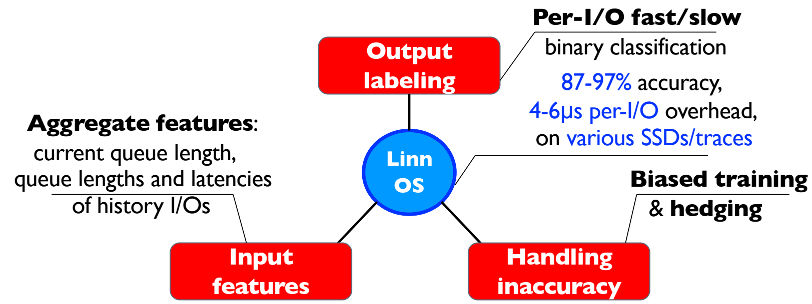
16

Handling inaccuracy



17

Recap



18

Other designs

More detail in the paper!

User-space offline training + kernel-space online inference

… + OS

Can utilize additional to further reduce inference overhead

Support re-tracing, re-training, and re-uploading weights

19

Outline

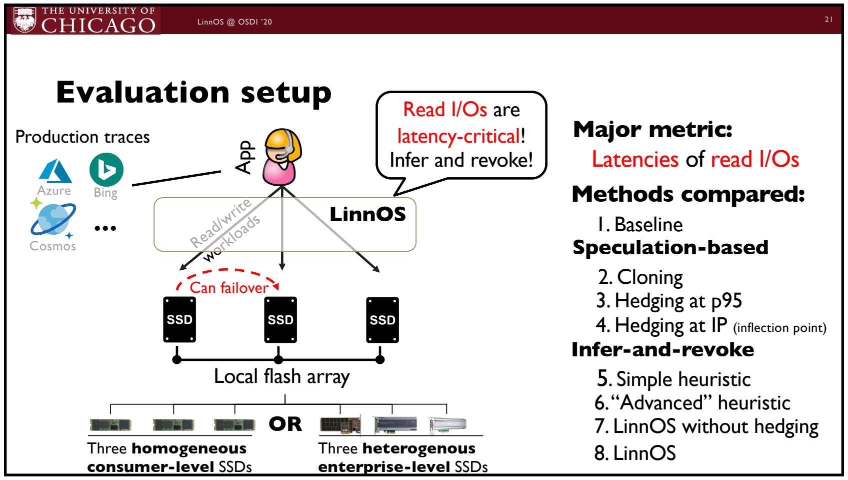
❑ Introduction

❑ Challenges & Solutions

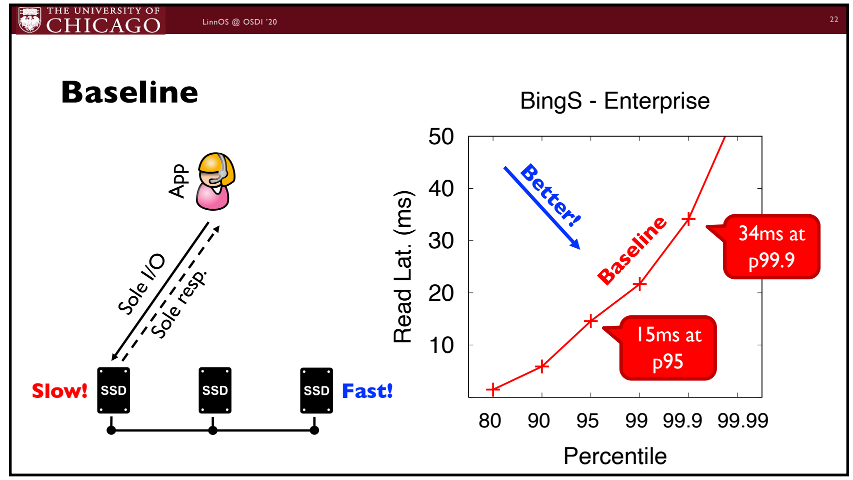
❑ **Evaluations**

❑ Conclusion

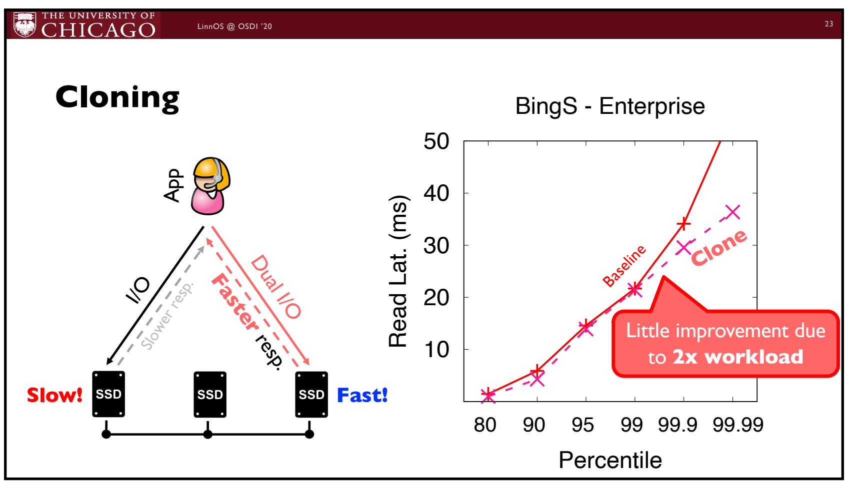
20


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
21
Evaluation setup
Production traces
Azure
Bing
Cosmos
...
App
Read/write workloads
LinnOS
Can failover
SSD
SSD
SSD
Local flash array
OR
Three homogeneous consumer-level SSDs
Three heterogenous enterprise-level SSDs
Read I/Os are latency-critical! Infer and revoke!
Major metric:
Latencies of read I/Os
Methods compared:
1. Baseline
Speculation-based
2. Cloning
3. Hedging at p95
4. Hedging at IP (inflection point)
Infer-and-revoke
5. Simple heuristic
6. "Advanced" heuristic
7. LinnOS without hedging
8. LinnOS

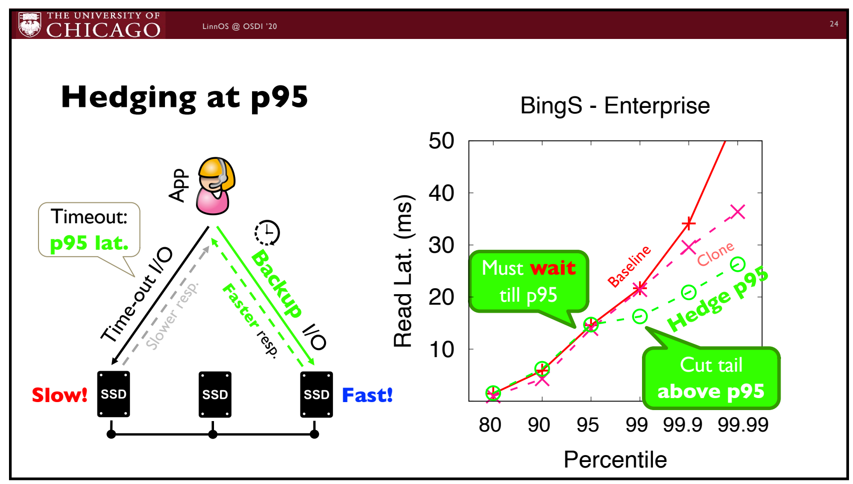
21


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
22
Baseline
App
Sole I/O
Sole resp.
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Read Lat. (ms)
Percentile
Better!
Baseline
34ms at p99.9
15ms at p95

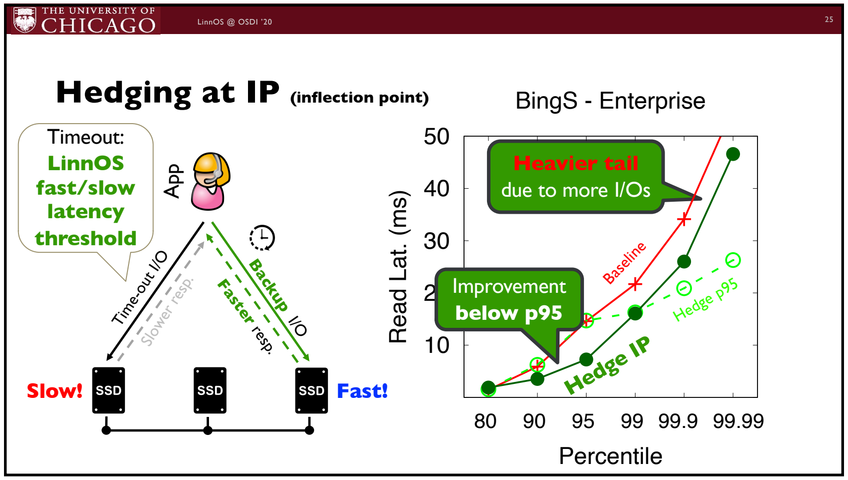
22


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
23
Cloning
App
I/O
Slower resp.
Dual I/O
Faster resp.
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Read Lat. (ms)
Percentile
Baseline
Clone
Little improvement due to 2x workload

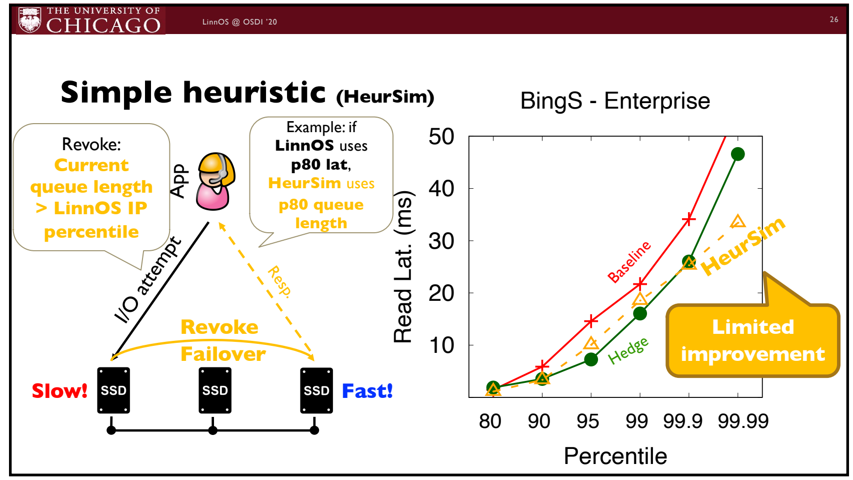
23


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
24
Hedging at p95
App
Timeout: p95 lat.
Time-out I/O
Slower resp.
Backup I/O
Faster resp.
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Read Lat. (ms)
Percentile
Must wait till p95
Baseline
Clone
Hedge p95
Cut tail above p95

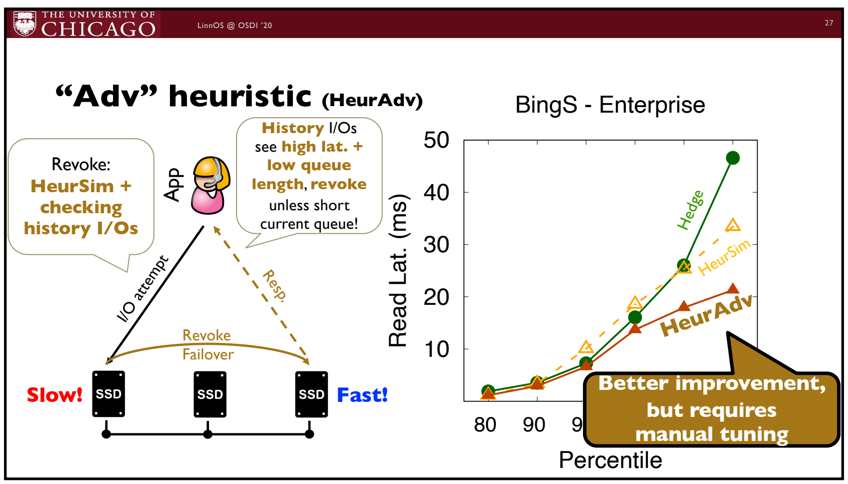
24


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
25
Hedging at IP (inflection point)
Timeout: LinnOS fast/slow latency threshold
App
Time-out I/O
Slower resp.
Backup I/O
Faster resp.
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Heavier tail due to more I/Os
Improvement below p95
Baseline
Hedge p95
Hedge IP
Read Lat. (ms)
50
40
30
10
80
90
95
99
99.9
99.99
Percentile

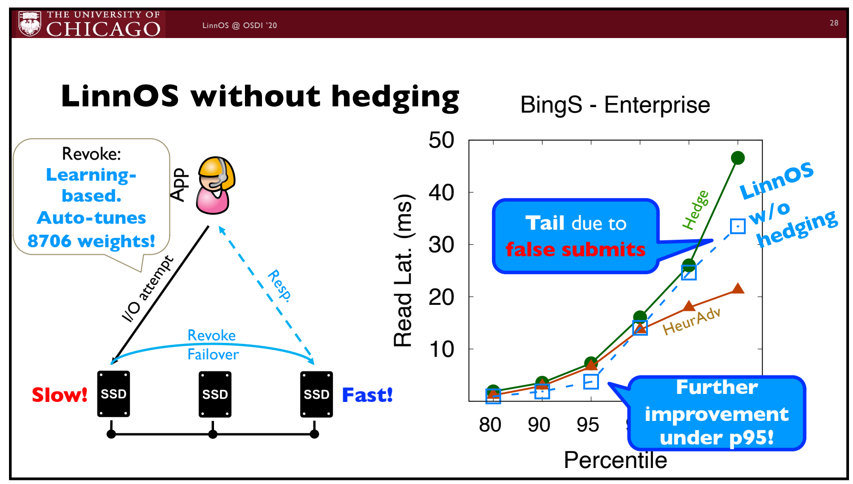
25


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
26
Simple heuristic (HeurSim)
Revoke: Current queue length > LinnOS IP percentile
Example: if LinnOS uses p80 lat, HeurSim uses p80 queue length
App
I/O attempt
Resp.
Revoke Failover
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Baseline
HeurSim
Hedge
Limited improvement
Read Lat. (ms)
50
40
30
20
10
80
90
95
99
99.9
99.99
Percentile

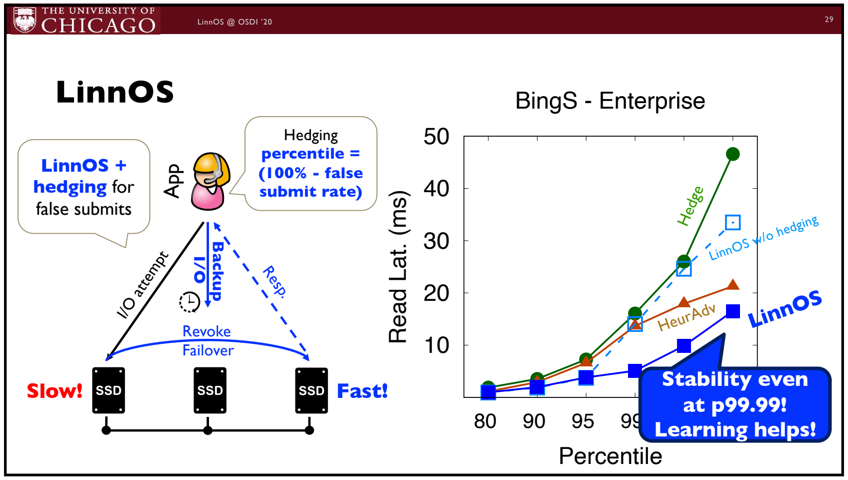
26


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
27
"Adv" heuristic (HeurAdv)
Revoke: HeurSim + checking history I/Os
History I/Os see high lat. + low queue length, revoke unless short current queue!
App
I/O attempt
Resp.
Revoke Failover
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Hedge
HeurSim
HeurAdv
Better improvement, but requires manual tuning
Read Lat. (ms)
50
40
30
20
10
80
90
Percentile

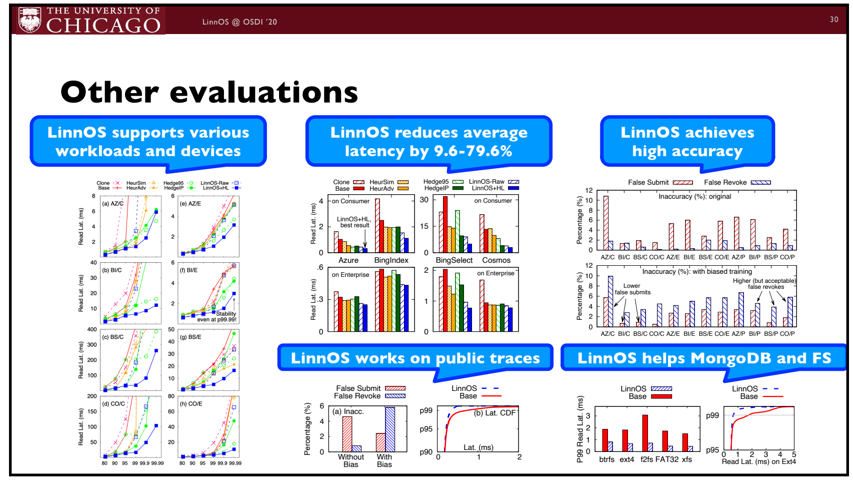
27


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
28
LinnOS without hedging
Revoke: Learning-based. Auto-tunes 8706 weights!
App
I/O attempt
Resp.
Revoke Failover
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Tail due to false submits
Hedge
LinnOS w/o hedging
HeurAdv
Further improvement under p95!
Read Lat. (ms)
50
40
30
20
10
80
90
95
Percentile

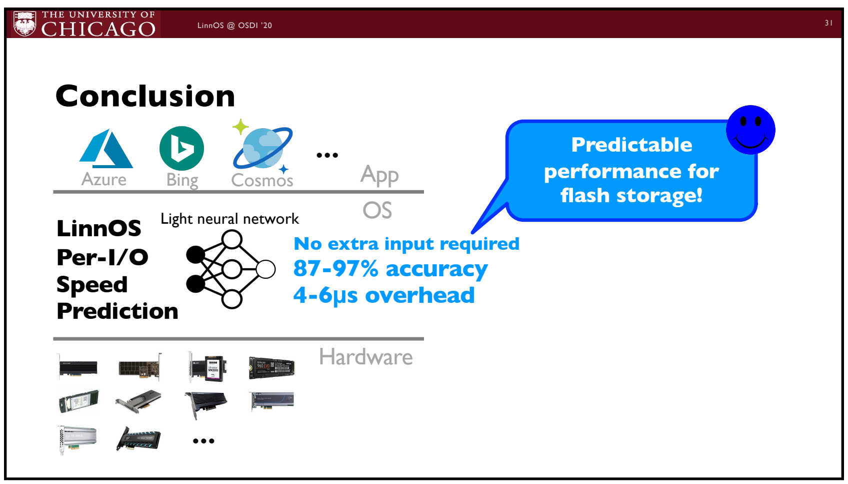
28


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
29
LinnOS
LinnOS + hedging for false submits
App
Hedging percentile = (100% - false submit rate)
I/O attempt
I/O
Backup
Resp.
Revoke Failover
Slow!
SSD
SSD
SSD
Fast!
BingS - Enterprise
Read Lat. (ms)
50
40
30
20
10
80
90
95
99
Percentile
Hedge
LinnOS w/o hedging
HeurAdv
LinnOS
Stability even at p99.99! Learning helps!

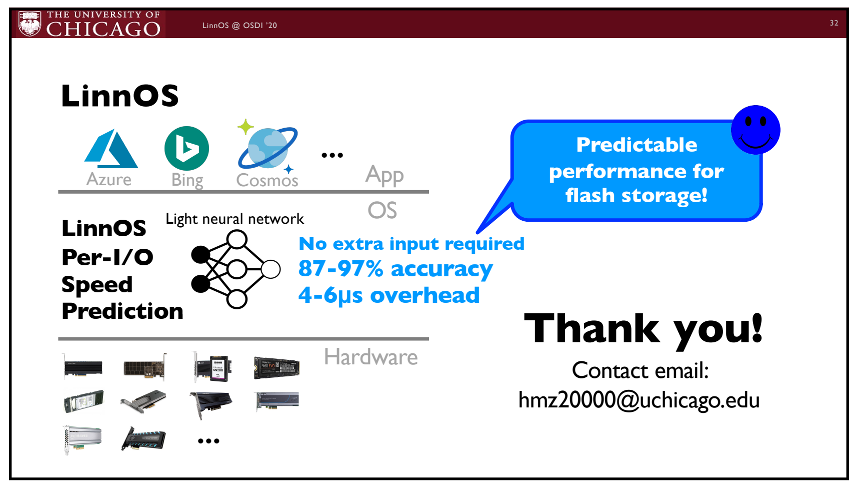
29


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
30
Other evaluations
LinnOS supports various workloads and devices
LinnOS reduces average latency by 9.6-79.6%
LinnOS achieves high accuracy
LinnOS works on public traces
LinnOS helps MongoDB and FS

30


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
31
Conclusion
Azure
Bing
Cosmos
...
App
OS
LinnOS Per-I/O Speed Prediction
Light neural network
No extra input required
87-97% accuracy
4-6μs overhead
Hardware
...
Predictable performance for flash storage!

31


THE UNIVERSITY OF CHICAGO
LinnOS @ OSDI '20
32
LinnOS
Azure
Bing
Cosmos
...
App
OS
LinnOS Per-I/O Speed Prediction
Light neural network
No extra input required
87-97% accuracy
4-6μs overhead
Hardware
...
Predictable performance for flash storage!
Thank you!
Contact email:
hmz20000@uchicago.edu

32