Server Side Cache Performance Analysis

Rayan Zachariassen, CTO, NEVEX
rayan@nevex.com

What determines performance

- Your Application
- The Kind of Cache: write-through or write-back
- Cache Software
  - What goes in
  - **How fast it is**
  - What comes out
- Other control points:
  - Hardware
  - Size
  - Load

What you are comparing to

- The performance of the application using the primary storage? NO
- The performance of the application using the cache device as primary storage!!!
- Remember, this is technical analysis, not an ROI exercise
- Use I/O as experienced by the application

Marketing benchmarks don't help

- tpmC: ~70/30 read/write mix
- Assuming reads and writes "cost" the same, maximum performance increase for a write-through cache is ~3.

The Math

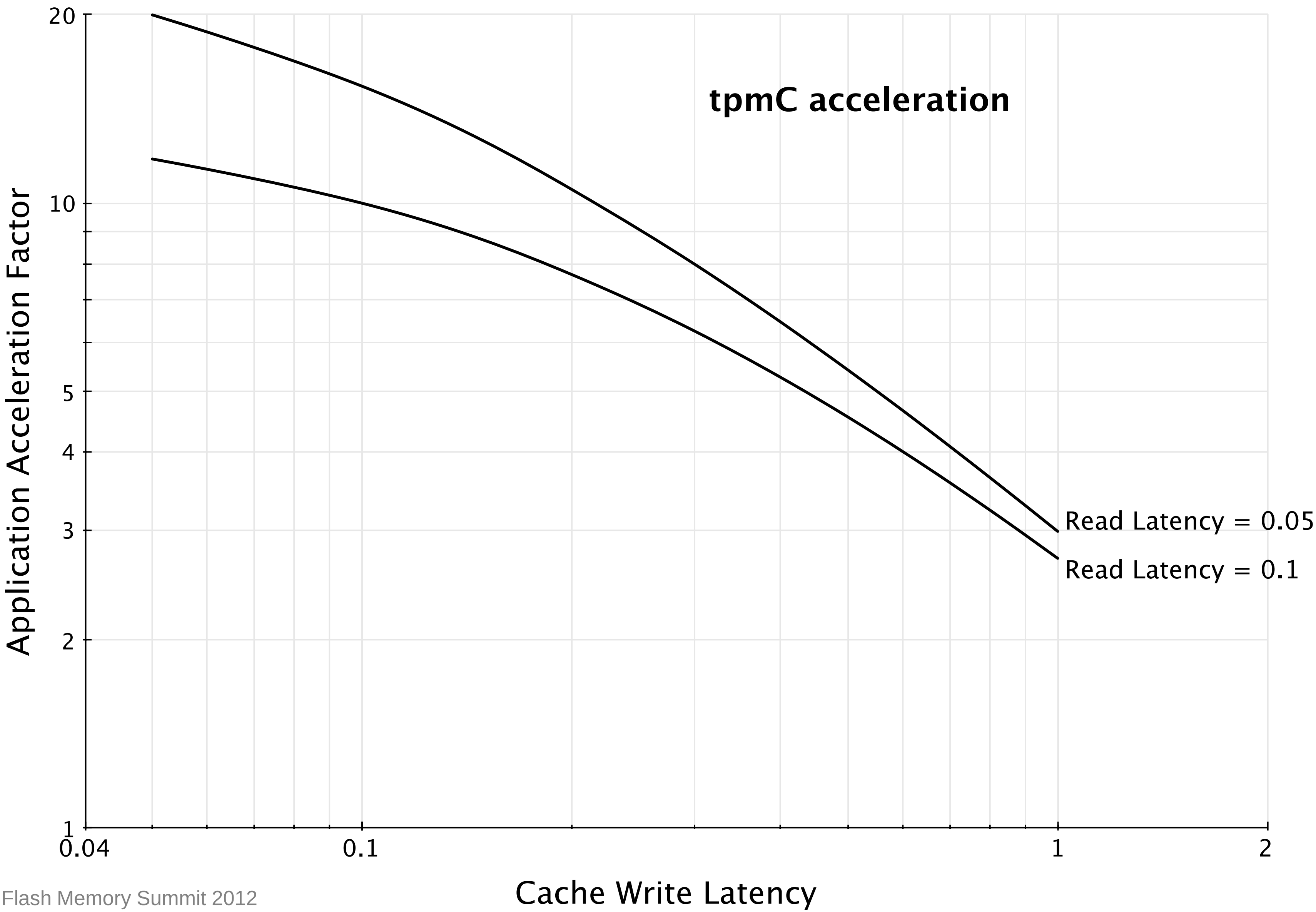
- I/O bound application
- Assume reads and writes both cost: 1
- 70 reads + 30 writes = 100
- Write-through caching accelerates reads
- Say read cost goes to 0.1
- 70 reads + 30 writes = 7 + 30 = 37
- Application acceleration = 100/37 = **2.7x**

Write-back calculation

- Write-back caching accelerates reads and writes
- Lets say write cost now goes to 0.1
- 70 reads + 30 writes = 7 + 3 = 10
- Application acceleration = 100/10 = **10x**
- Write-back can be done in the controller (very common) or in software

Hidden write-back caching

- Lets say the baseline tpmC test already uses write acceleration (due to BB hardware)
- 70 reads + 30 writes = 70 + 3 = 73, baseline
- We now accelerate reads
- 70 reads + 30 writes = 7 + 3 = 10
- Application acceleration = 73/10 = **7.3x**


tpmC acceleration
Application Acceleration Factor
20
10
5
4
3
2
1
0.04
0.1
1
2
Read Latency = 0.05
Read Latency = 0.1
Cache Write Latency

To relate an application benchmark
to your situation,

you need to know
EVERYTHING

about the application I/O pattern
and the platform

“It goes faster”

is not analysis

Micro benchmarks for analysis

Application benchmarks for validation

Micro benchmarks

- The Full Sweep
- Triangle test
- Latency curves
- Noise Rejection

A Sweep

Disk	IO Size	QD	IO Type	Read/ Write	IOPS	MB/s	Latency (usec)	Max Latency (msec)	CPU Util (%)
I:	4096		random	read					
I:	8192		random	read					
I:	65536		random	read					
I:	65536		sequential	read					
I:	1048576		sequential	read					
I:	4096		random	write					
I:	8192		random	write					
I:	65536		random	write					
I:	65536		sequential	write					
I:	1048576		sequential	write					

The Full Sweep

- Run baseline sweeps for primary storage and cache drive, as an application would use them
- Run a sweep for each possible cache option:
  - Primed
  - Noise
  - 1-level or 2-level
  - Other

The Full Sweep

- Run with select queue depths: 1 & 16
  - 16 was picked to not run into out-of-CPU issues for a single thread on the hardware
- Understand what the cache has to do
  - Read hits, read misses, write hits, write misses
- Pay attention to what nominal performance is:
  - Like the cache device
  - Like primary storage

Triangle test

- Maintain a constant queue depth but vary the number of threads
- For example use QD 1x16, 2x8, 4x4, 8x2, 16x1
- Use it to characterize effect of application multi-threading, or lack thereof
- Ideal is to see good and consistent performance

Latency

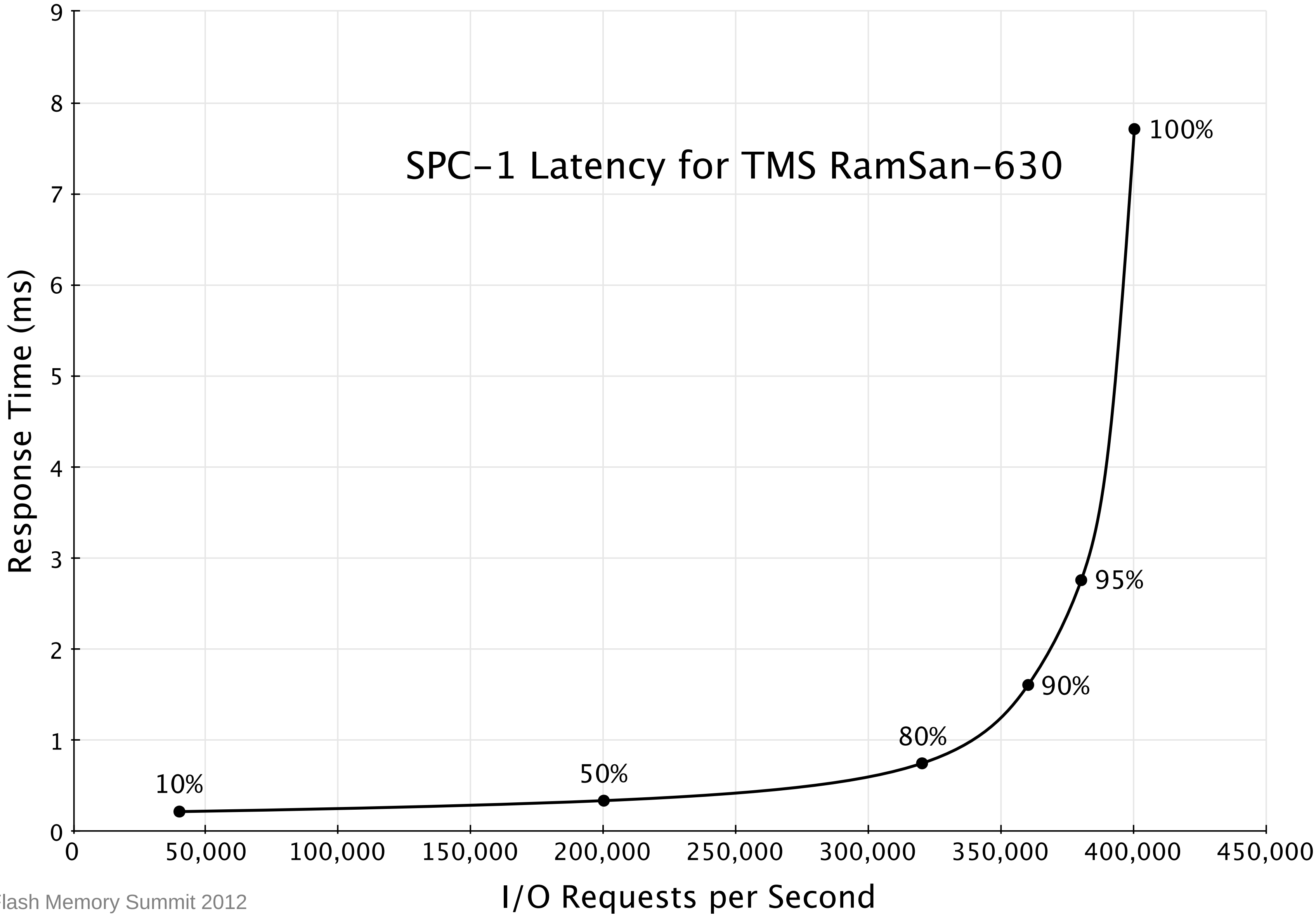
SPC-1

- Measure latency at 10%,50%,80%,90%,95%, 100% of max. throughput
- I/O requests are read/write mix
- Includes mirroring, etc.

IOPS-Latency micro benchmark

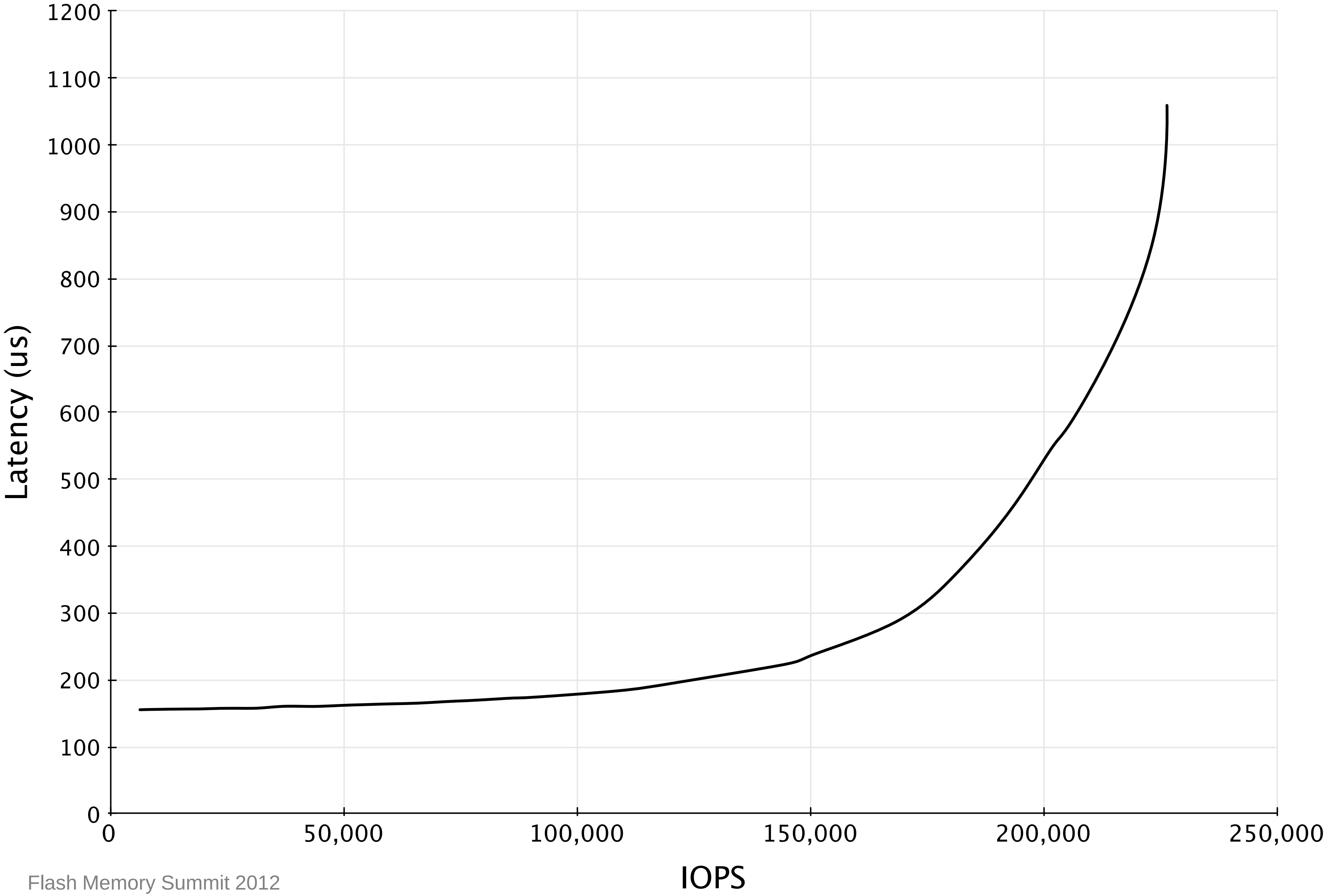
- Measure latency at each level of concurrency
- One type of I/O at a time, e.g. 4K Random Reads
- No partitioning, no mirroring

We are NOT trying to test the I/O system
We ARE trying to test the caching system

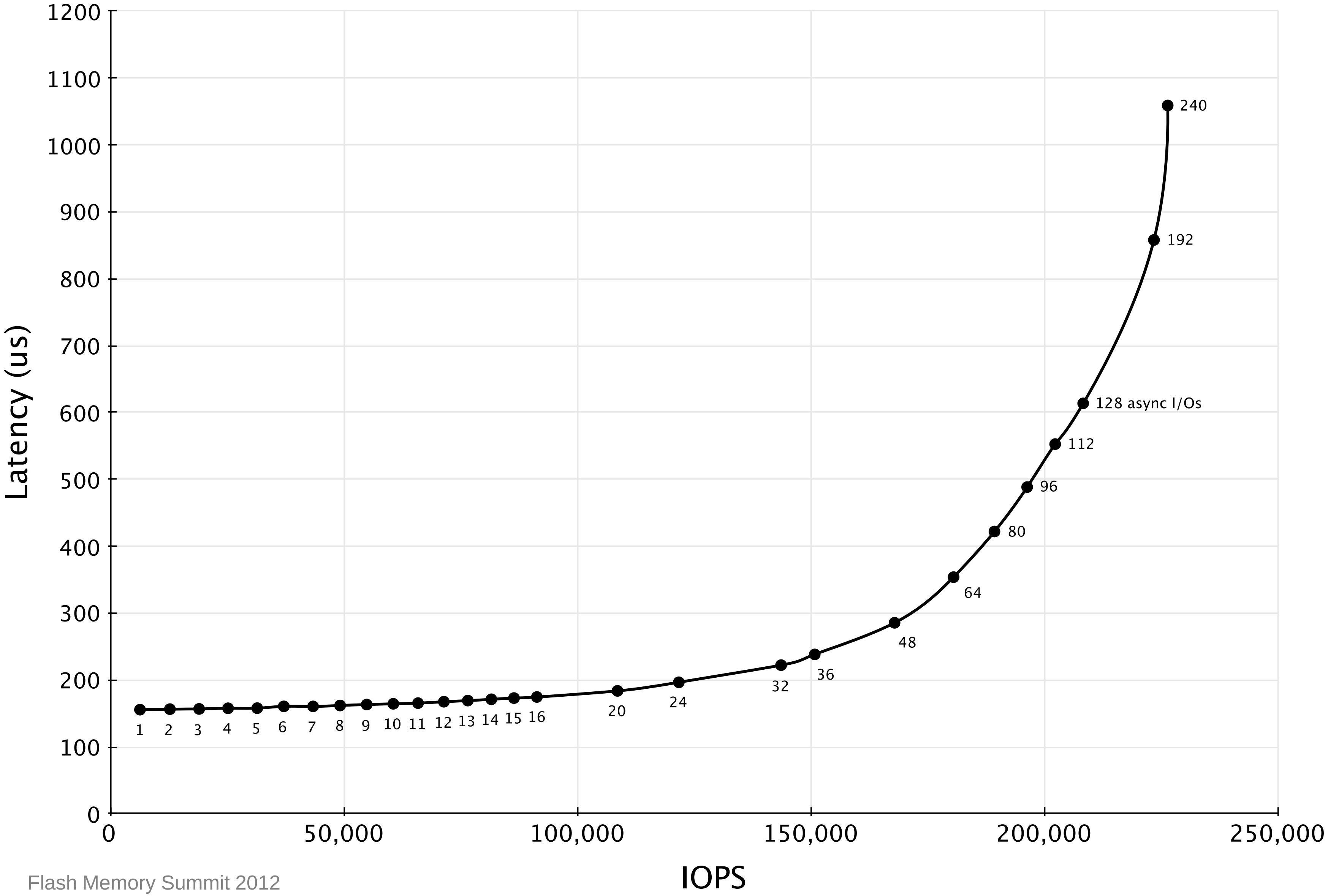


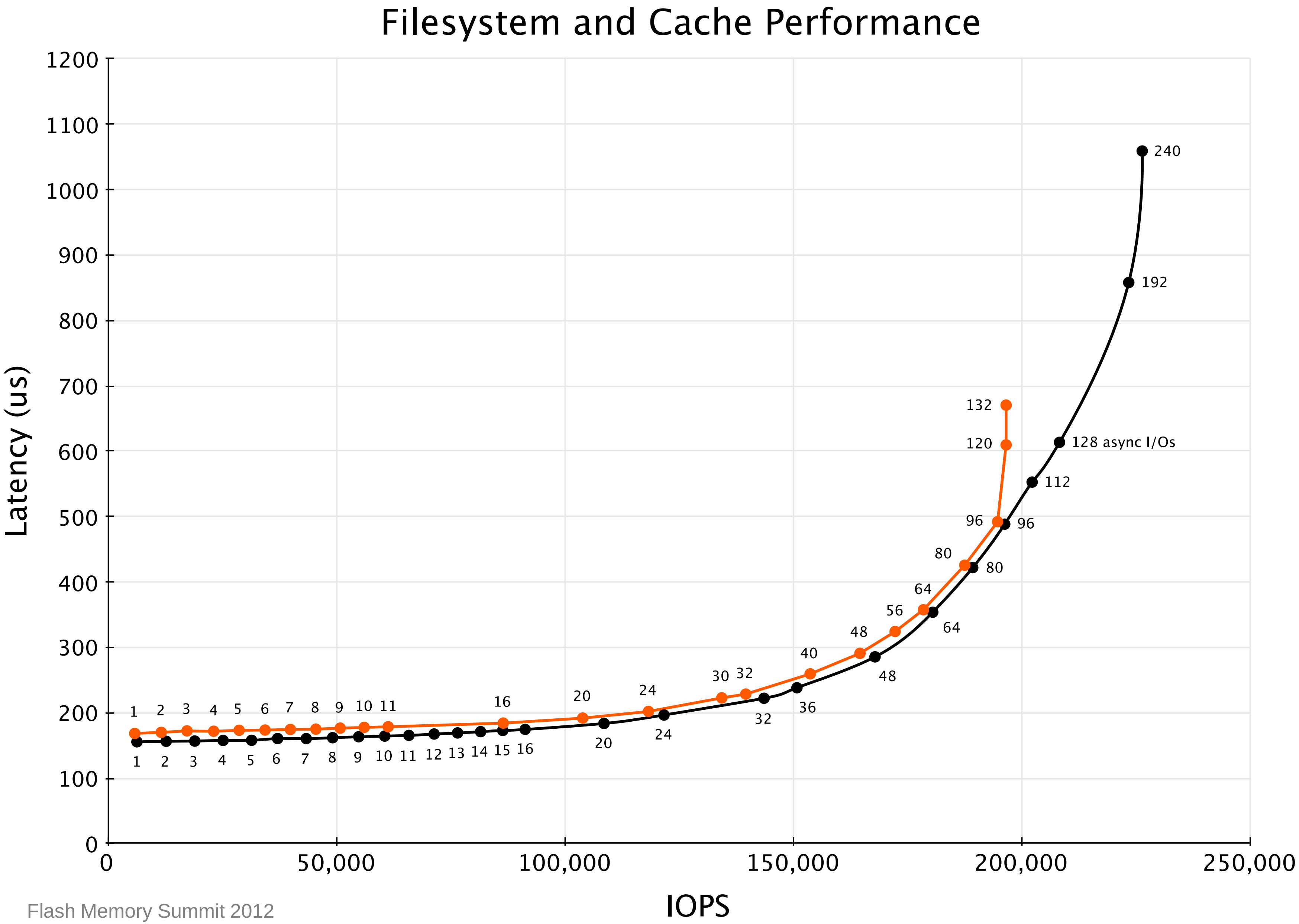
SPC-1 Latency for TMS RamSan-630
Response Time (ms)
I/O Requests per Second
0
1
2
3
4
5
6
7
8
9
0
50,000
100,000
150,000
200,000
250,000
300,000
350,000
400,000
450,000
10%
50%
80%
90%
95%
100%

Filesystem Performance

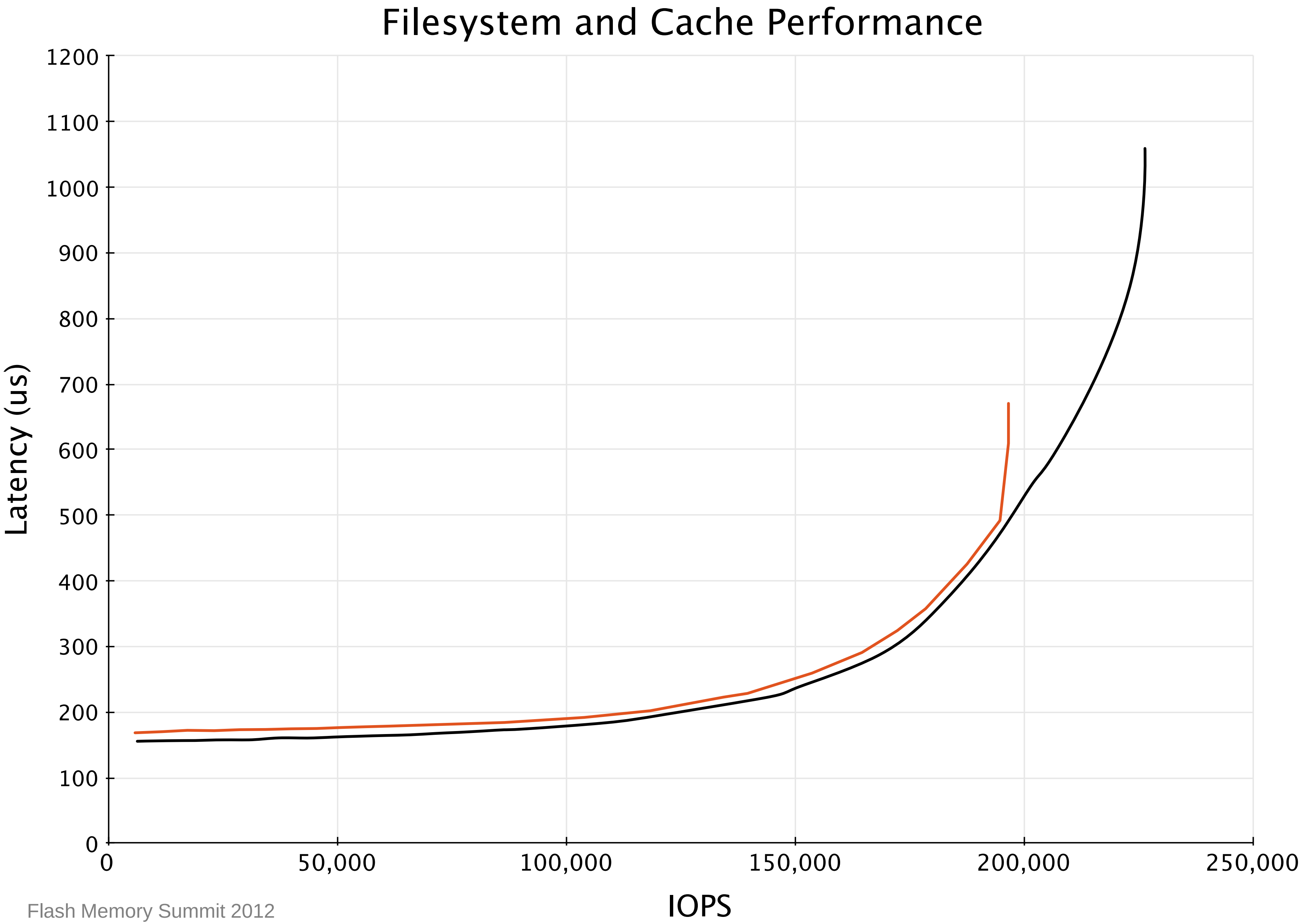


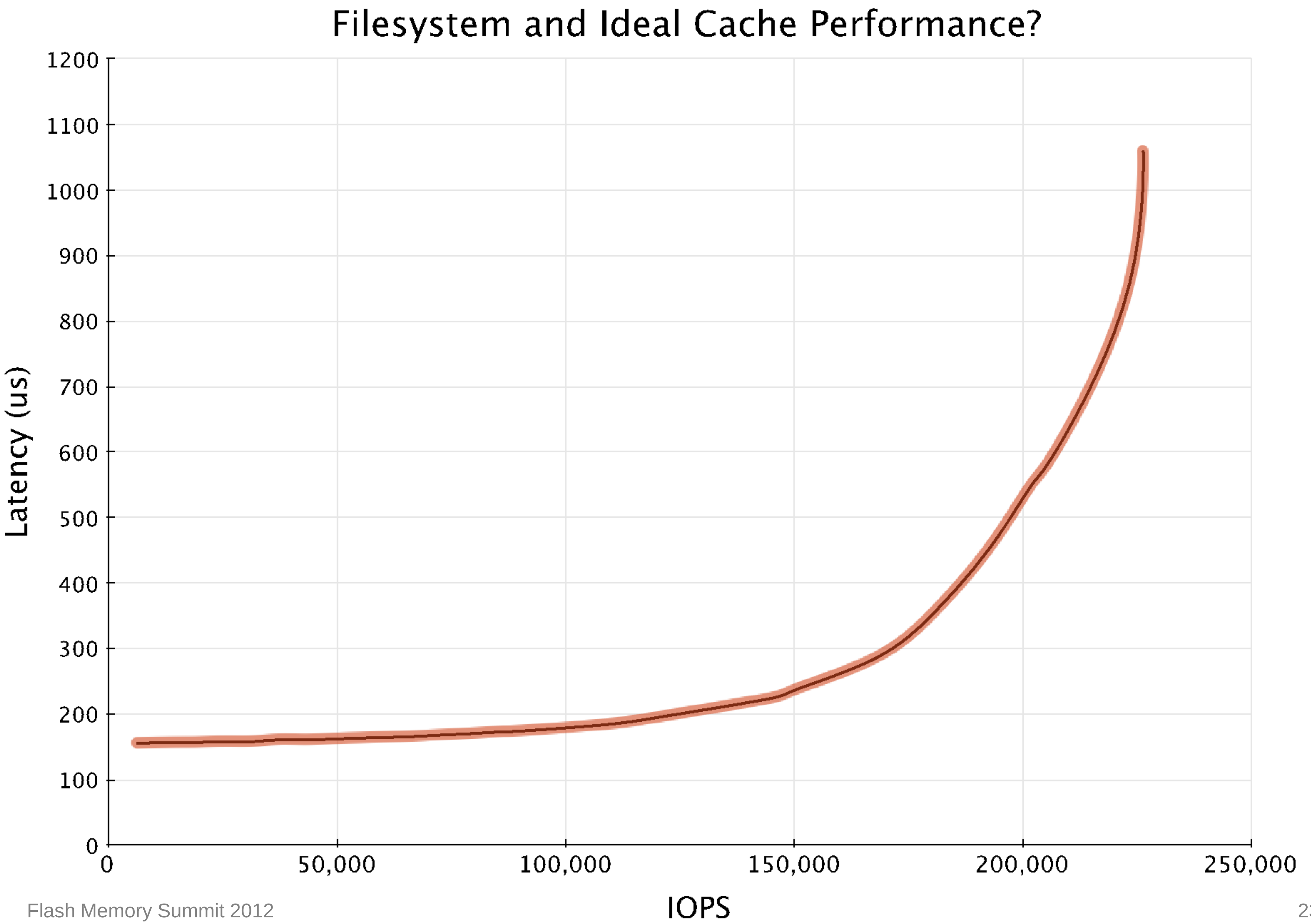
Filesystem Performance

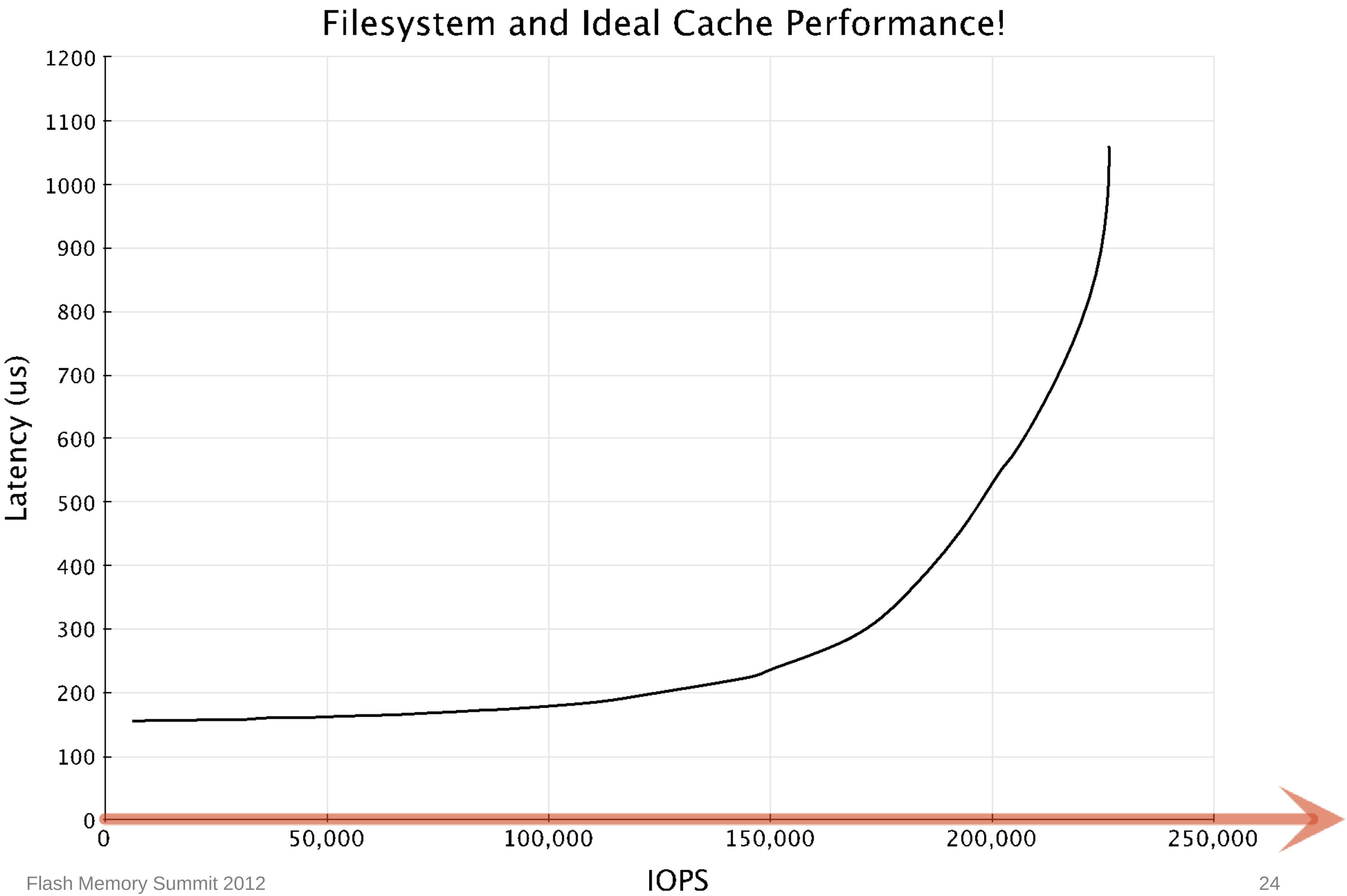


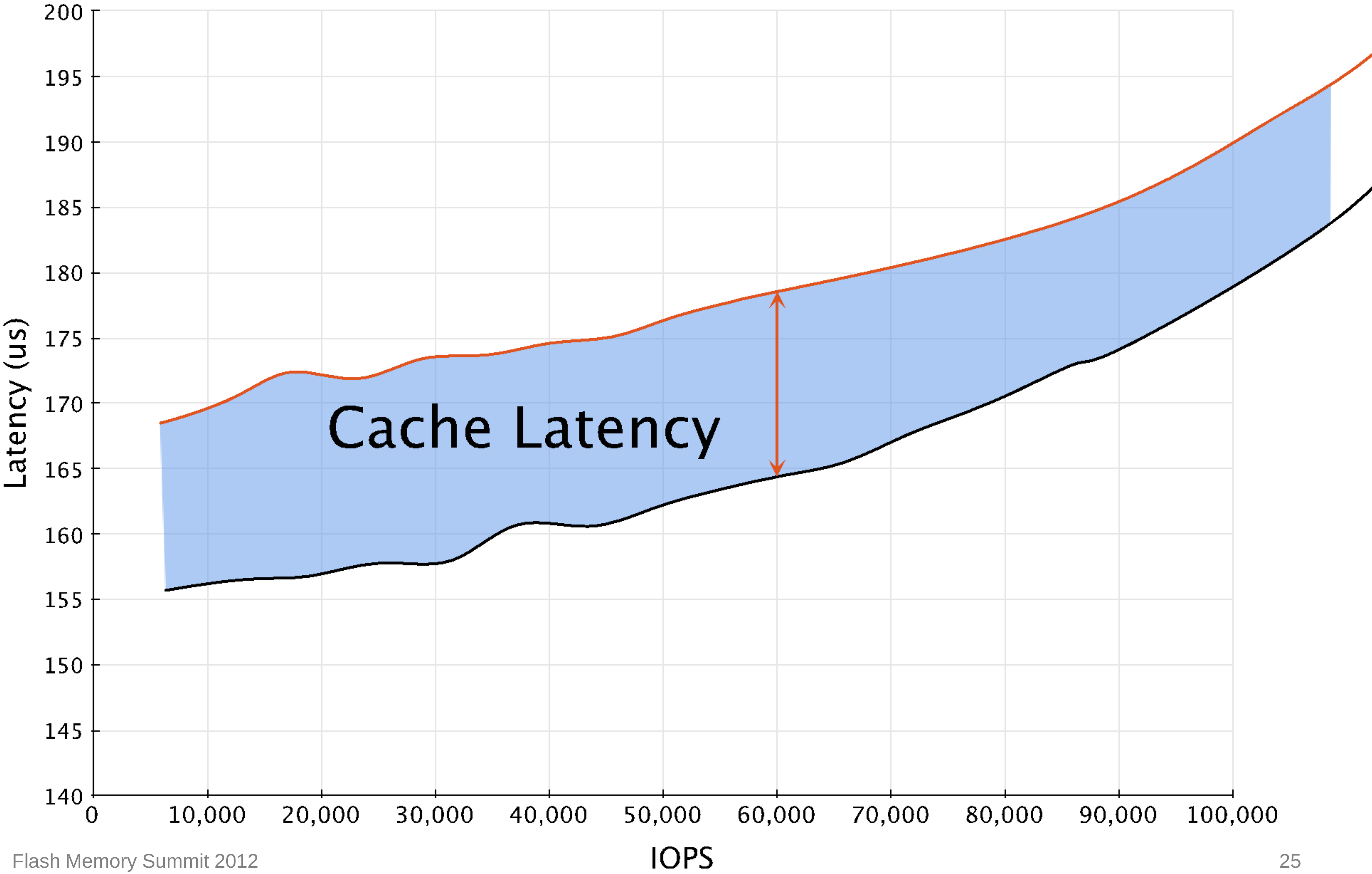


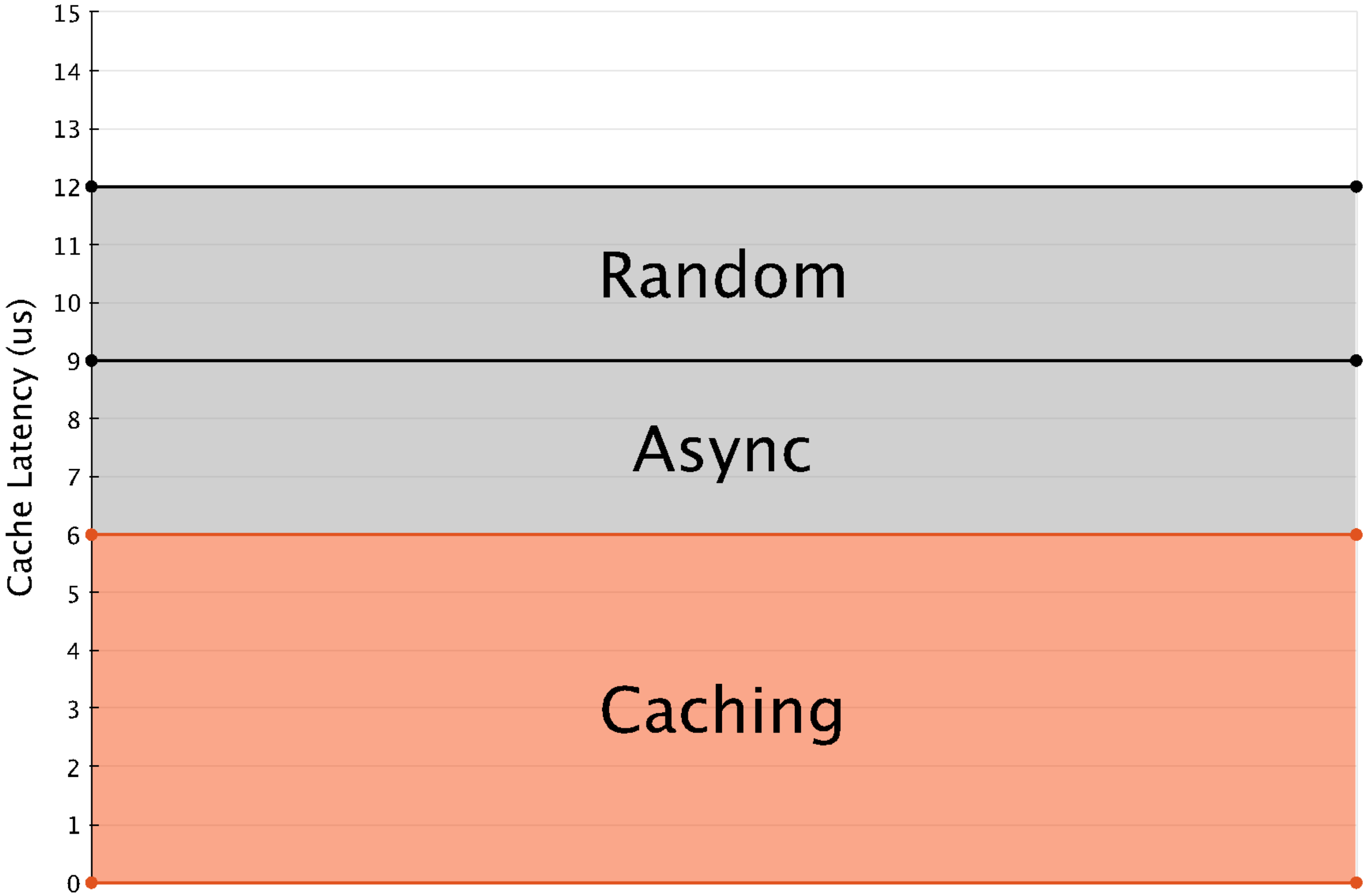
Filesystem and Cache Performance
Latency (us)
IOPS
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
0
50,000
100,000
150,000
200,000
250,000
128 async I/Os
240
192
112
132
120
96
80
64
56
48
40
36
32
30
24
20
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1


Filesystem and Cache Performance
Latency (us)
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
0
50,000
100,000
150,000
200,000
250,000
IOPS


Filesystem and Ideal Cache Performance?
Latency (us)
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
0
50,000
100,000
150,000
200,000
250,000
IOPS


Filesystem and Ideal Cache Performance!
Latency (us)
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
0
50,000
100,000
150,000
200,000
250,000
IOPS


200
195
190
185
180
175
170
165
160
155
150
145
140
Latency (us)
Cache Latency
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
100,000
IOPS


15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Cache Latency (us)
Random
Async
Caching

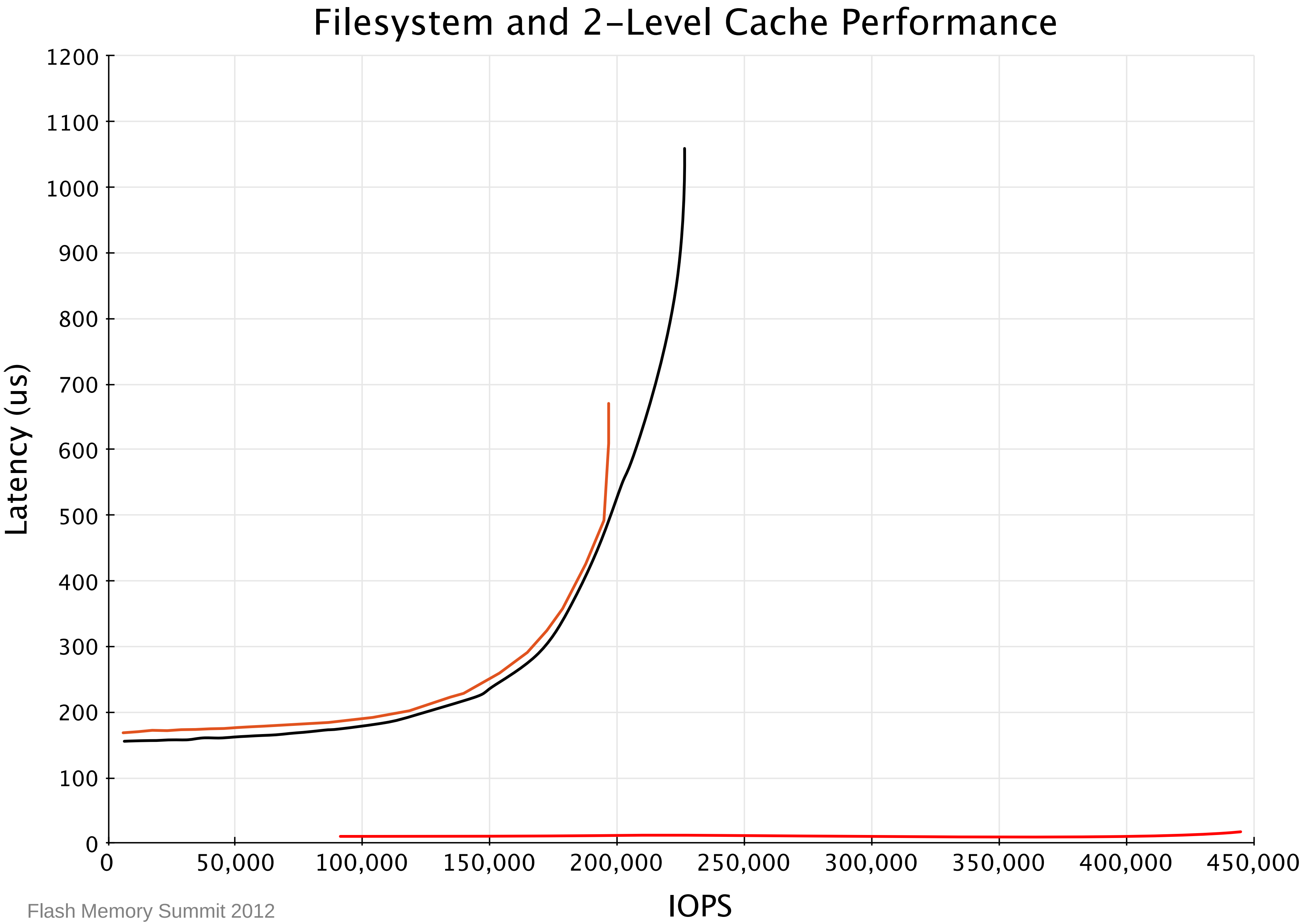
Laws of Latency

Sync < Async

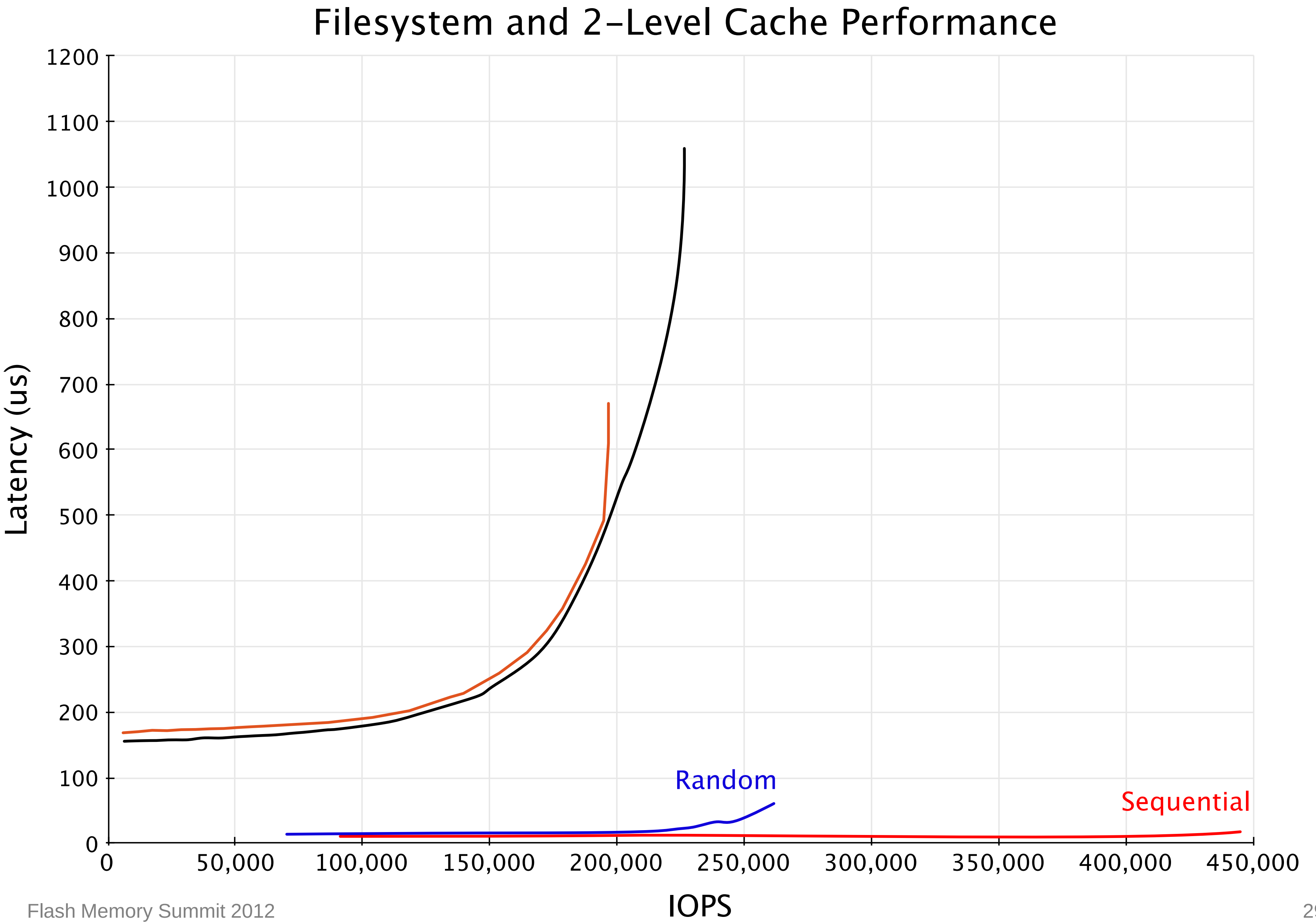
Sequential < Random

Delta ~ CPU time

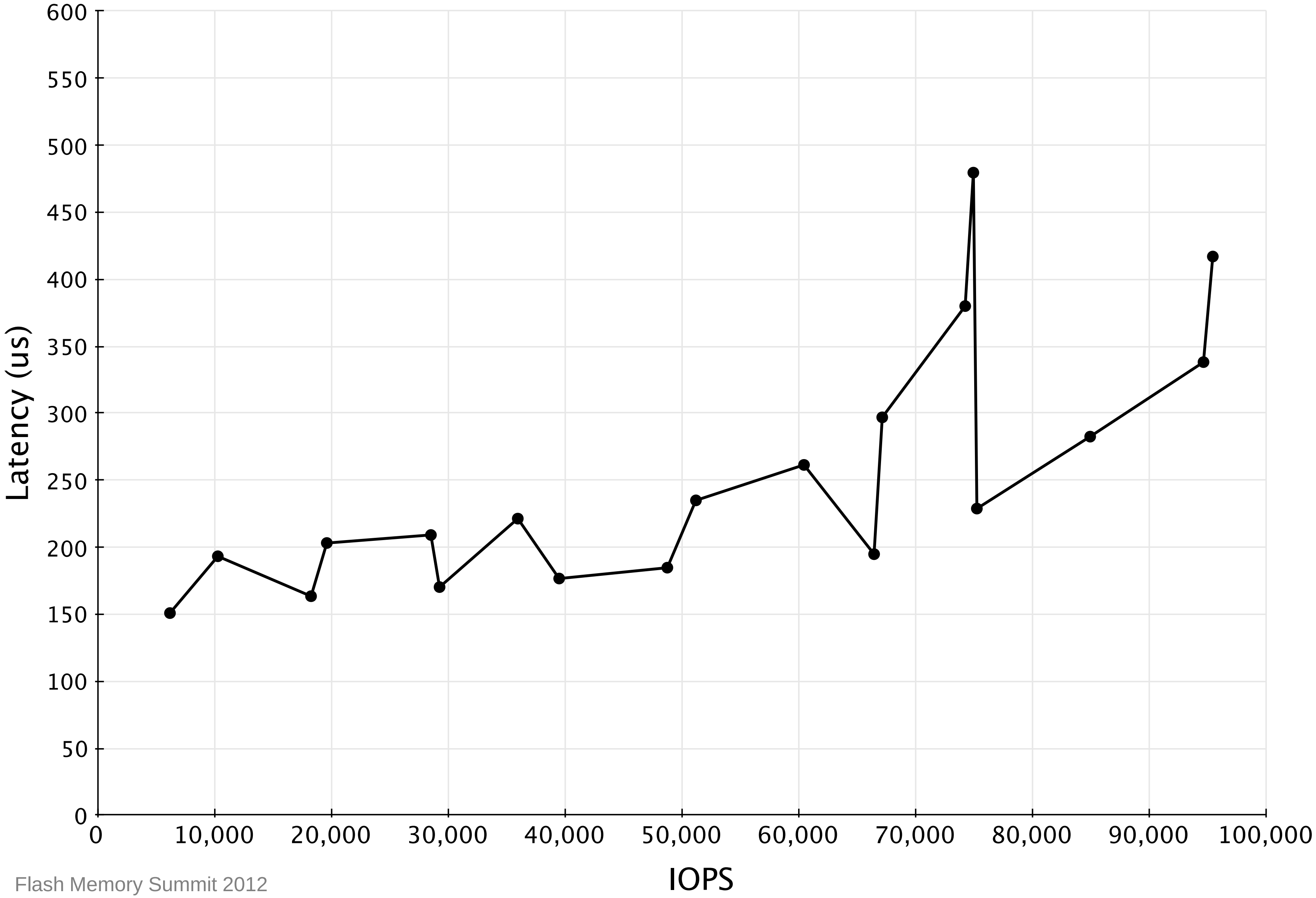
0 is best!



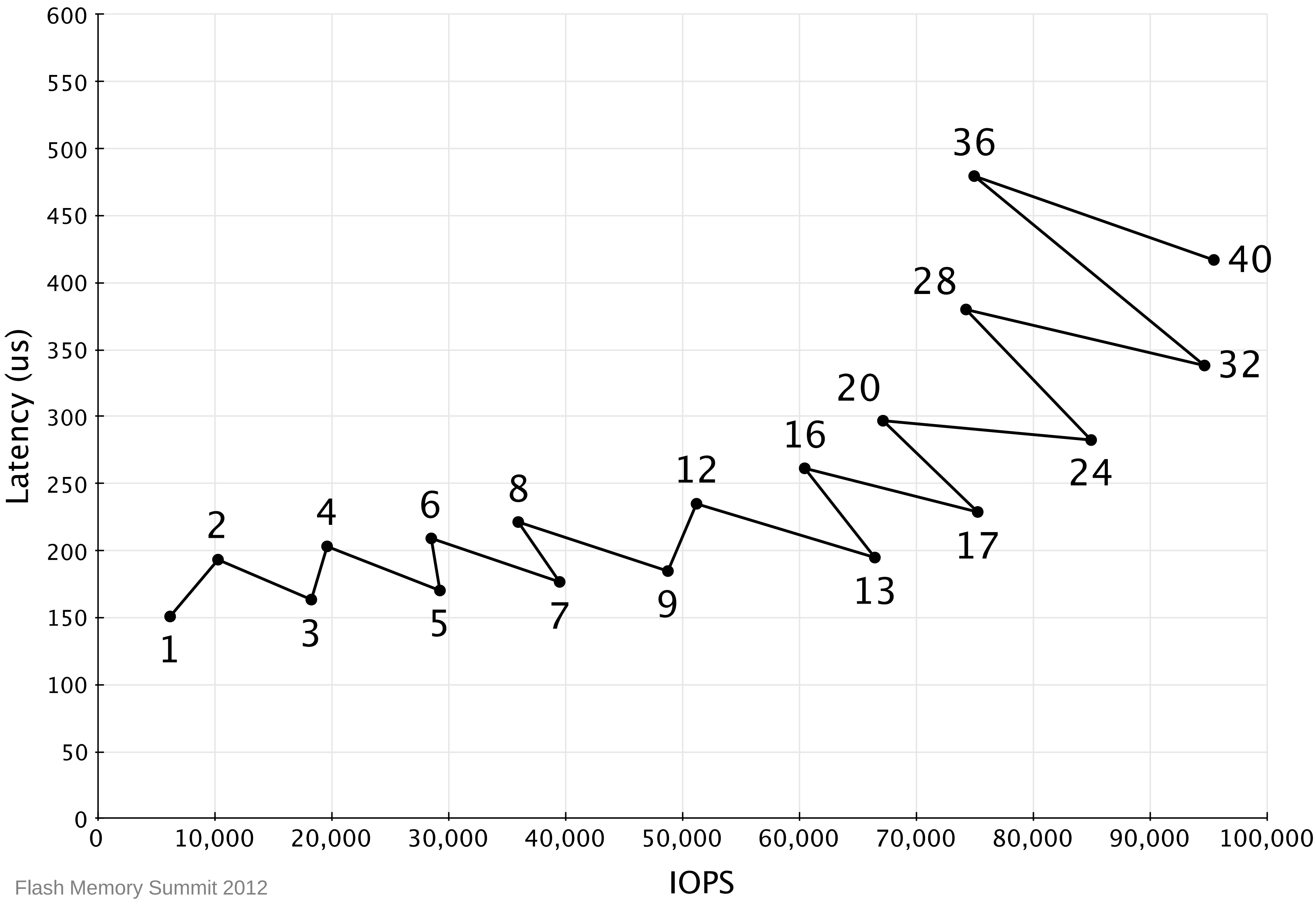
Filesystem and 2-Level Cache Performance
Latency (us)
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
0
50,000
100,000
150,000
200,000
250,000
300,000
350,000
400,000
450,000
IOPS


Filesystem and 2-Level Cache Performance
Latency (us)
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
0
50,000
100,000
150,000
200,000
250,000
300,000
350,000
400,000
450,000
IOPS
Random
Sequential

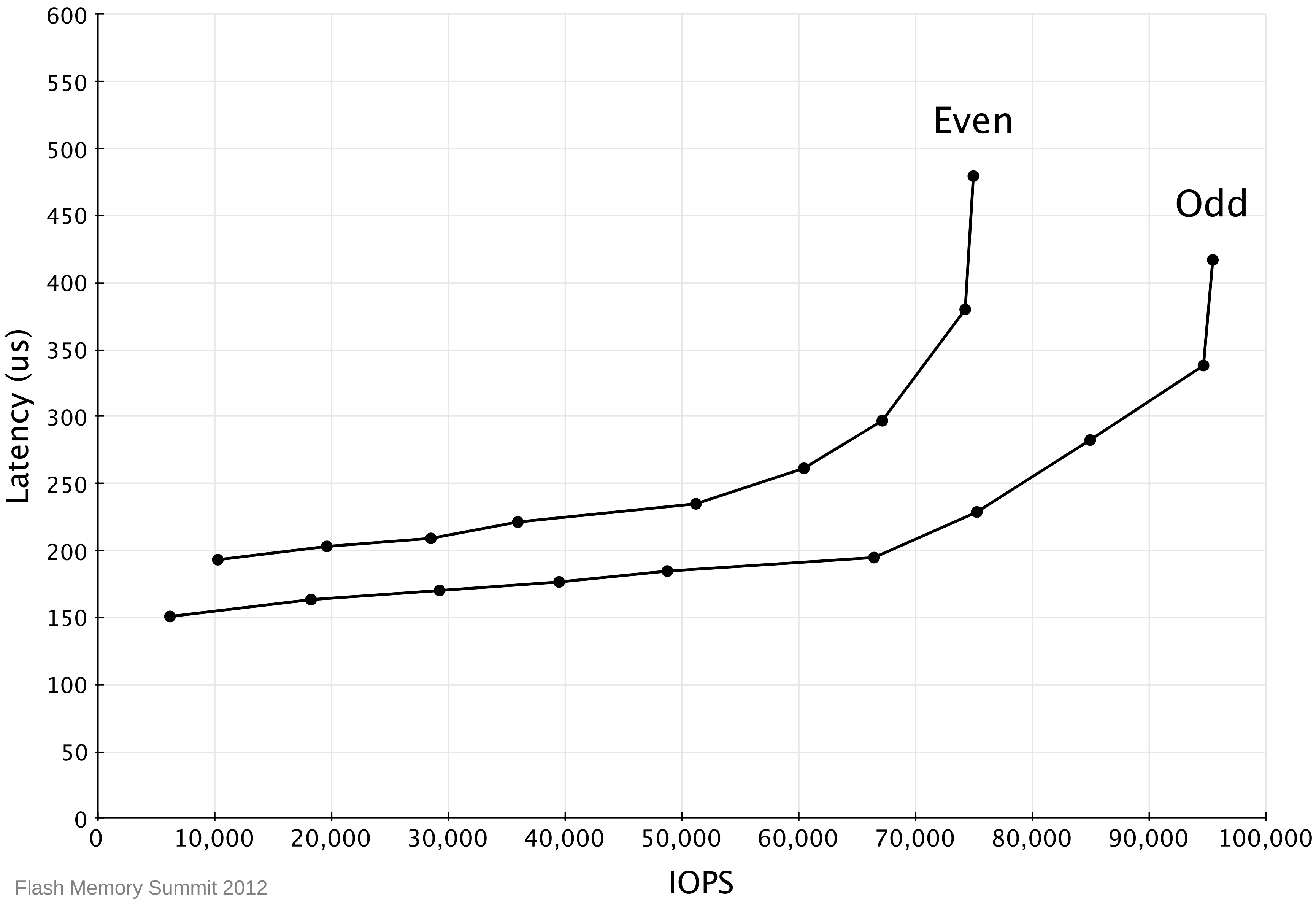
Not good


Latency (us)
600
550
500
450
400
350
300
250
200
150
100
50
0
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
100,000
IOPS

Not good


Latency (us)
600
550
500
450
400
350
300
250
200
150
100
50
0
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
100,000
IOPS
1
2
3
4
5
6
7
8
9
12
13
16
17
20
24
28
32
36
40

Aha!


Even
Odd
Latency (us)
IOPS
0
50
100
150
200
250
300
350
400
450
500
550
600
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
100,000

There be traps

- Bi-modal performance
- The CPU runs Faster when caching
- Modern multi-core processors are Too Slow
- Device conditioning
- Sync I/O is different from Async I/O
- Hidden caches
- I/O alignment

Noise Rejection

Points Made

- What determines performance
- Know what you are comparing to
- Most published benchmarks don't help
- Use micro benchmarks for analysis
- Latency curves are good
- There be traps

Server Side Cache Performance

- Analyze
  - Latency
  - Noise resistance

- Application
- Cache type, size and policies
- Cache device
- Platform OS, Hardware
- Load

It’s the **difference** that matters

Server Side Cache Performance Analysis

Rayan Zachariassen, CTO, NEVEX
rayan@nevex.com