AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

STG403

Amazon EFS performance best practices

Geert Jansen
Principal Product Manager
Amazon Web Services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“All problems in computer science can be solved by another level of indirection.”

David Wheeler

Professor of Computer Science, University of Cambridge

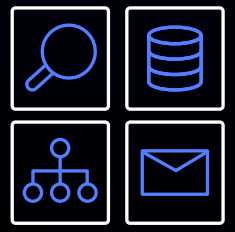
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Any performance problem can be solved by removing a level of indirection.”

Unknown

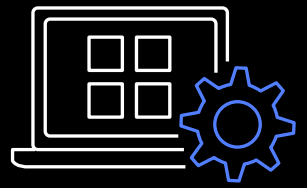
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Who is this session for? What will you get?



You are migrating an existing application to Amazon Elastic File System (Amazon EFS)

- Basic understanding of your application's FS performance
- Best practices for configuring your file system and OS



You are developing a new application for Amazon EFS

- Specific considerations for developing high-performance applications

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Thinking about performance

Recent and upcoming performance improvements

Amazon EFS storage classes, performance and throughput modes

OS configuration and mount options

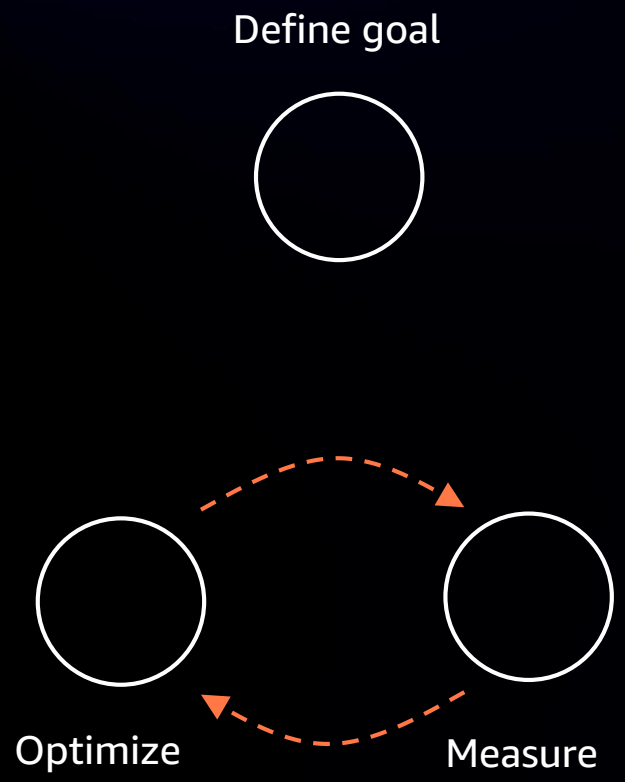
NFS caching and small-file performance

Considerations for using directories

Summary of performance best practices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to think about performance?



Performance is process

Establish goals

Iterative process

Measure: actual performance vs. goal

Optimize: change configuration or software

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS performance since 2020

Amazon Elastic File System announces 400% increase in read operations for General Purpose mode file systems

Posted On: Apr 1, 2020

Starting today, Amazon Elastic File System
systems support up to 35,000 read operati
the previous limit of 7,000. Maximum writ
second.

Amazon Elastic File System increases per-client throughput by 100%

Posted On: Jul 23, 2020

Amazon Elastic File System (Ama
client throughput, a 100% increa
throughput for an EFS file syste

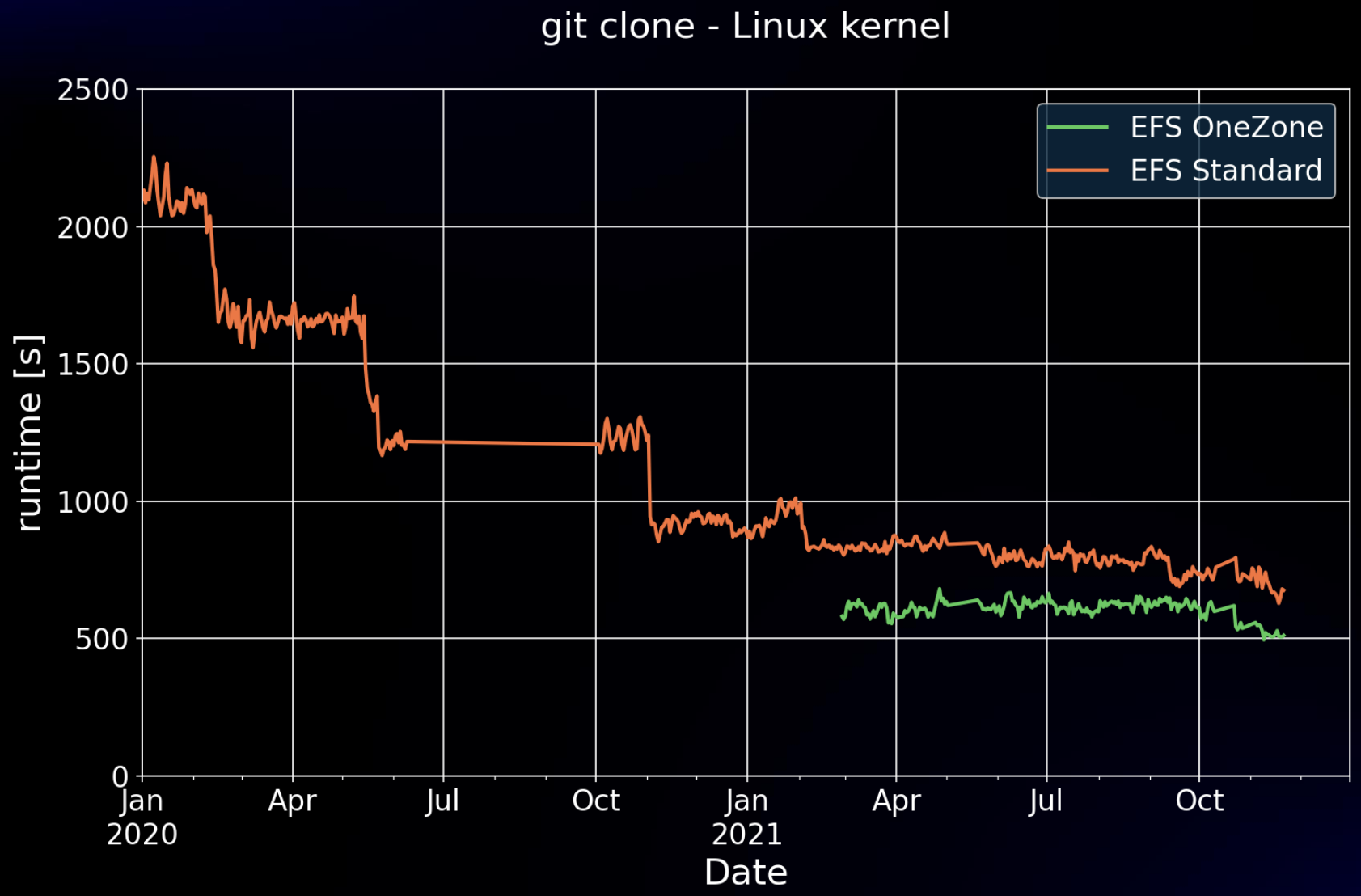
Amazon Elastic File System triples read throughput

Posted On: Jan 29, 2021

Amazon Elastic File System (Amazon EFS) now allows you to drive up to 3x higher read throughput on your file system. For example, bursting mode file systems now provide 300MB/s of bursting read throughput, or 300MB/s per TiB of data stored in Amazon EFS standard, whichever is higher. If you have configured 1 GB/s of

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS latency improvements: Writes

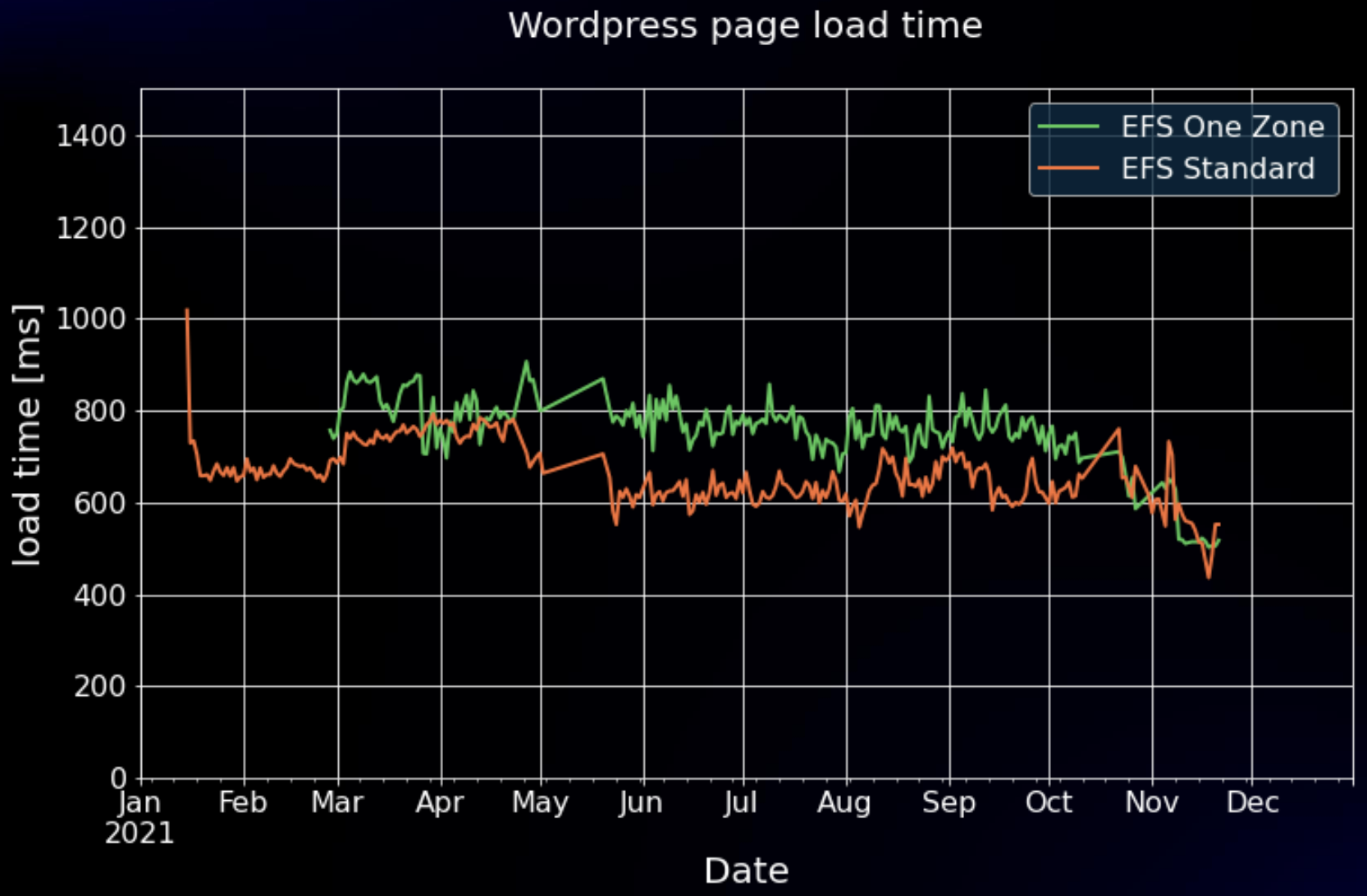


4.5x
improvement

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS latency improvements: Reads



2x
improvement

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS storage classes, performance and throughput modes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS storage classes

	One Zone	Standard
Main use	Highly available and durable file system that replicates data within a single availability zone	Highly available and durable file system that replicates data across multiple availability zones
IOPS	35,000	35,000
Throughput	3–5 GB/s (reads) 1–3 GB/s (writes)	3–5 GB/s (reads) 1–3 GB/s (writes)
Latency[1]	1.0–1.5 ms (reads) 2.4–7.4 ms (writes	1.0–2.4 ms (reads) 3.6–11.5 ms (writes)

[1] *Average read/write latencies for metadata and 4 KiB data operations in us-east-1. Results may vary based on workload, operating system, and other factors. Standard uses the General Purpose performance mode.*

aws

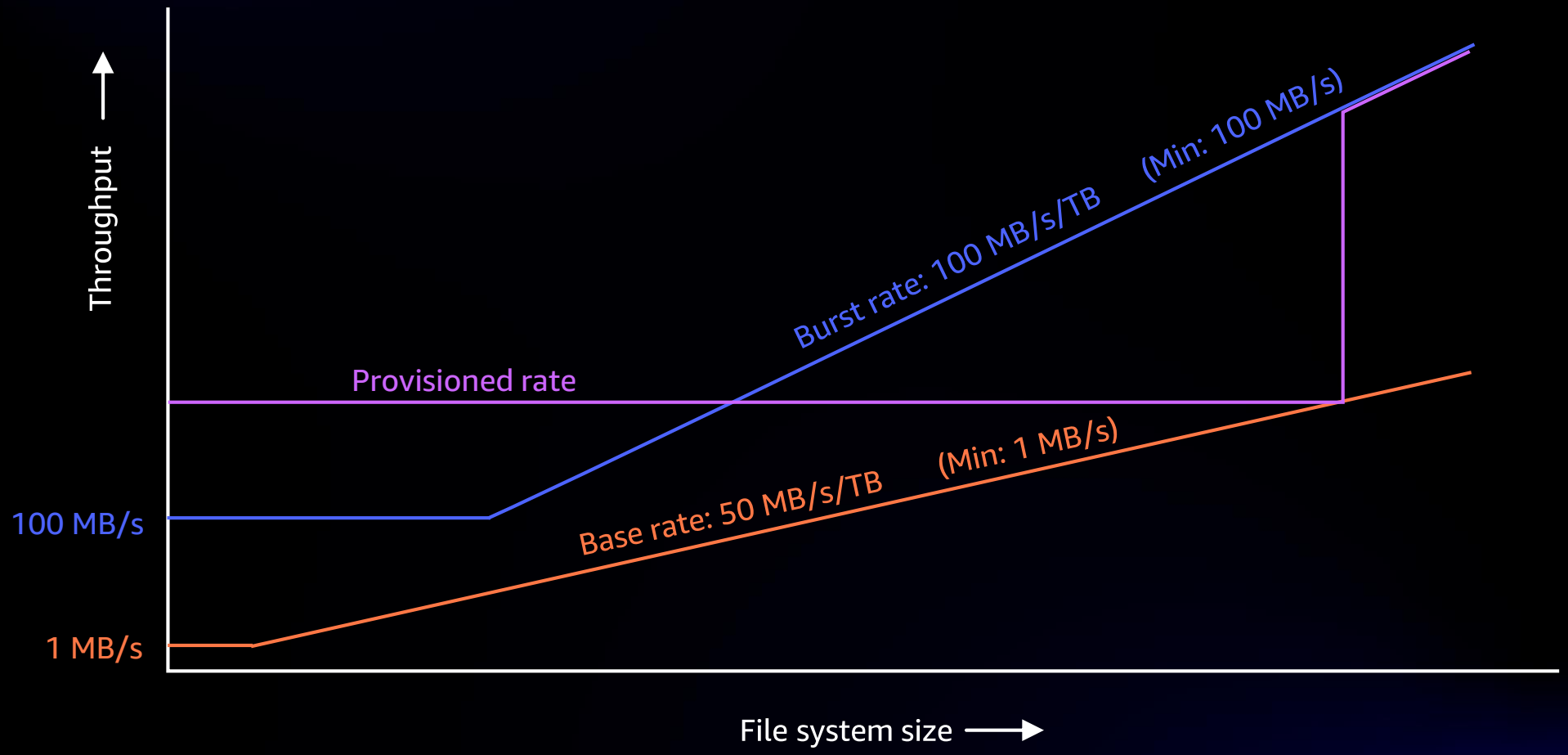
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS performance mode

	General Purpose	Max I/O (Standard only)
Main use	Majority of workloads (95%)	Scale-out workloads
Main benefit	Lowest latencies	Higher number of IOPS
IOPS	35,000 per second	500,000+ per second

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EFS throughput modes



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

OS configuration and mount options

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use a recent OS version

```
# Use latest available kernel

$ sudo yum update kernel
$ sudo reboot
```

Good OS versions

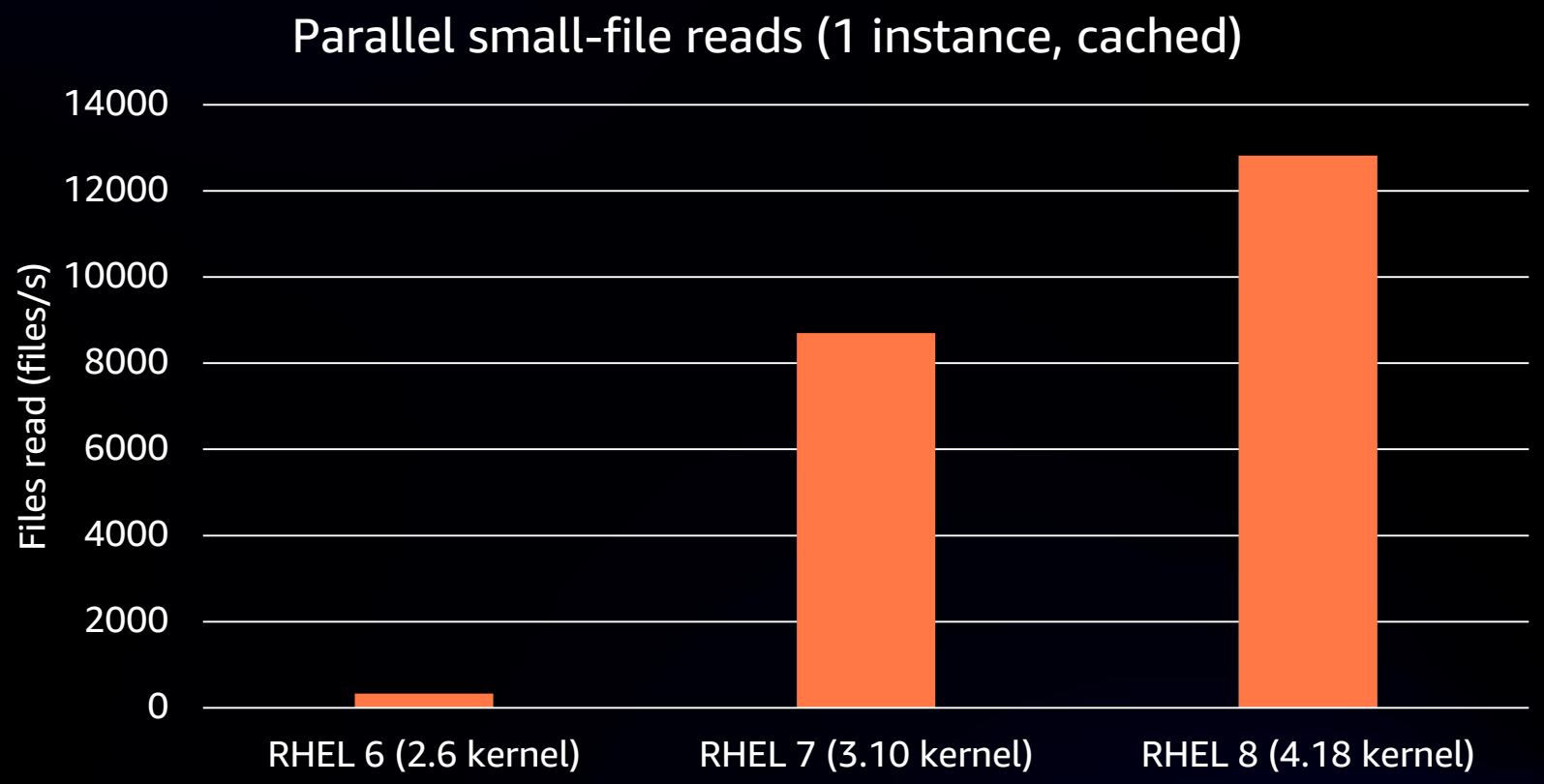
- Amazon Linux 1
- RHEL 7.3+
- Ubuntu 16.04

Better OS versions

- Amazon Linux 2+
- RHEL 8+
- Ubuntu 18.04+

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Performance with different OS versions



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use recommended mount options

```
$ sudo yum install amazon-efs-utils
$ sudo mount -t efs fs-01234578 /efs
```

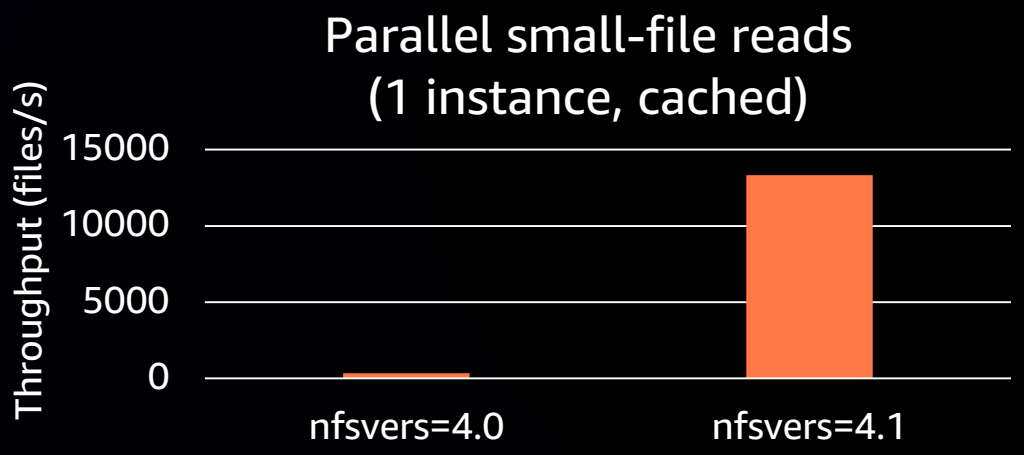
Option	Description
nfsvers=4.1	Use NFSv4.1
rsize=1048576 wsize=1048576	Use 1 MB read and write buffer sizes

aws

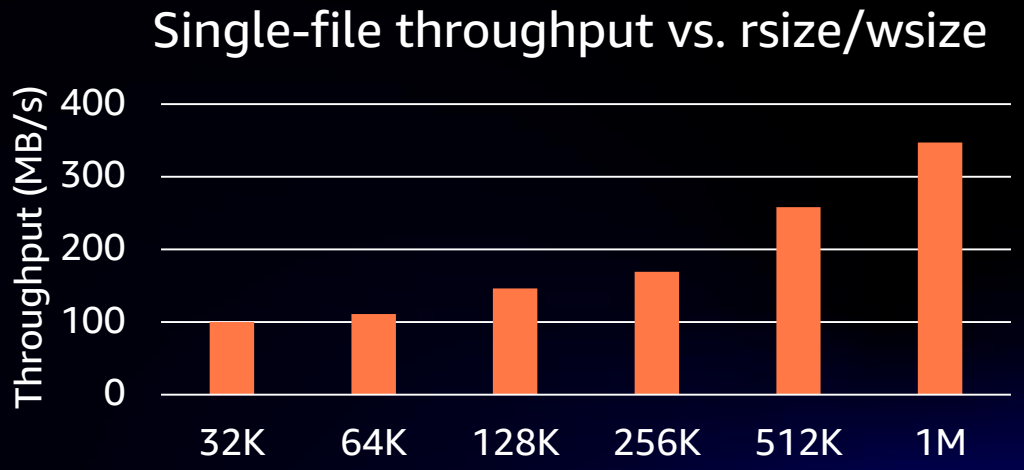
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Performance with different mount options

nfsvers=4.0 / 4.1



rsize / wsize



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

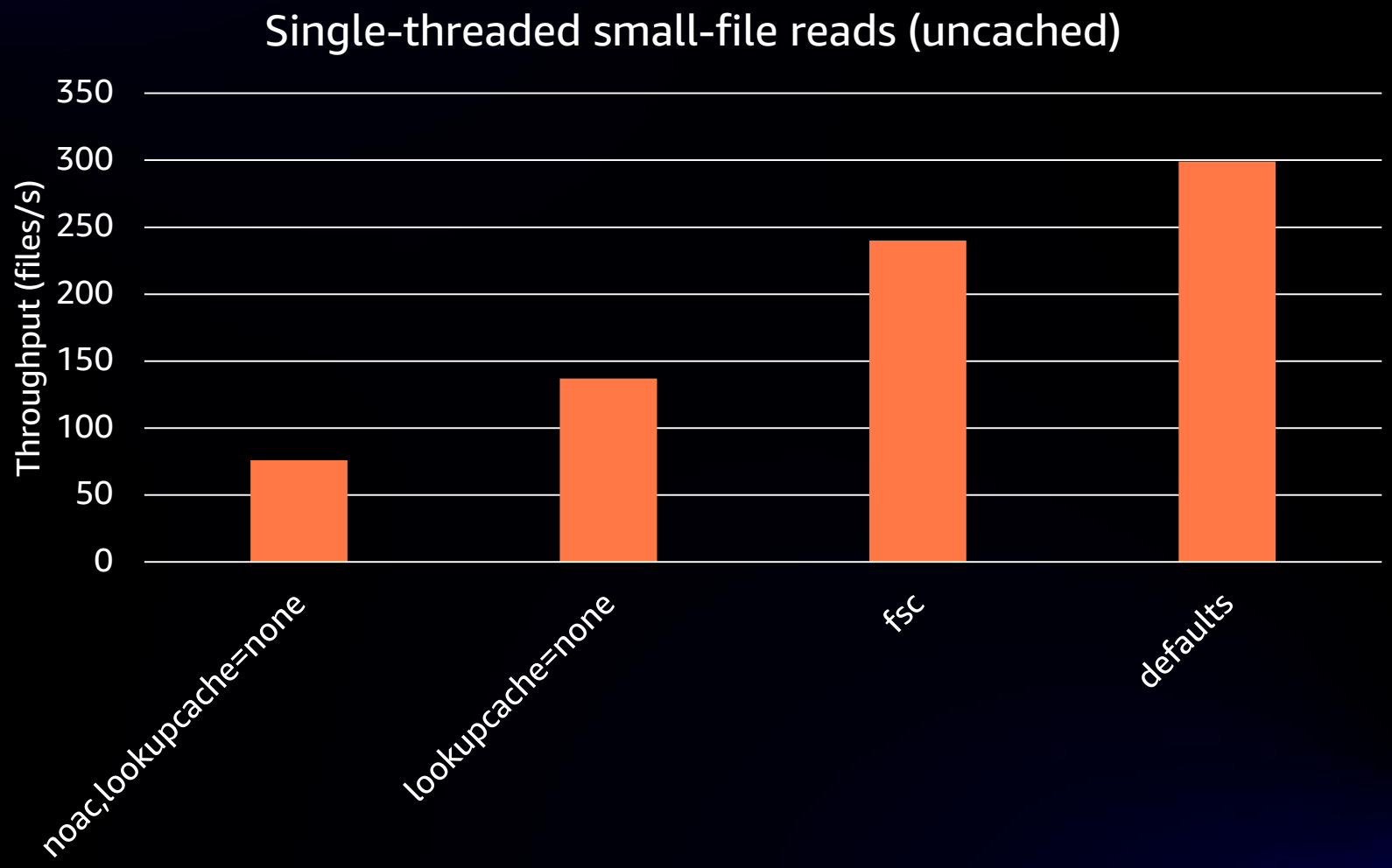
Mount options to avoid

Option	Description	Recommendation
noac, actimeo=0, acregmax=0, acdirmax=0	Disables attribute cache	**Avoid**; options have a very large performance impact
lookupcache=none lookupcache=pos	Disables file name lookup cache	**Avoid**; option has a large impact on performance Some vendors require ‘pos’
fsc	Enables local file caching	**Avoid**; does not change NFS cache coherency and will not reduce latencies Larger cache size may increase throughput for some workloads

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Impact of nonrecommended mount options

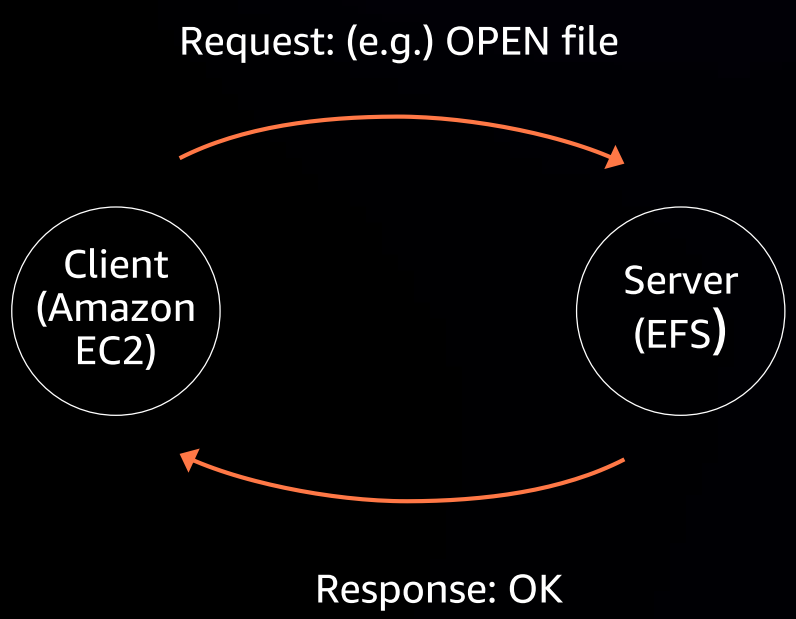


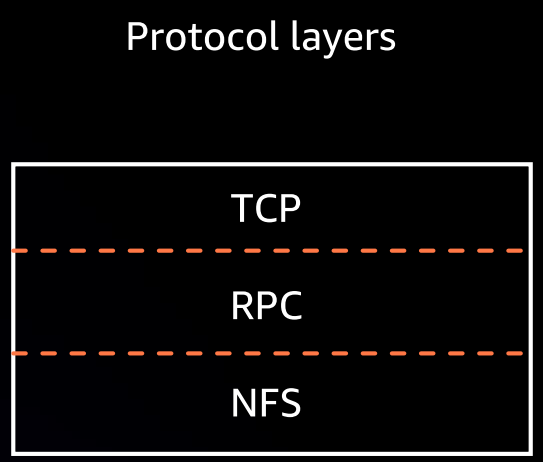
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

NFS caching and small-file performance

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

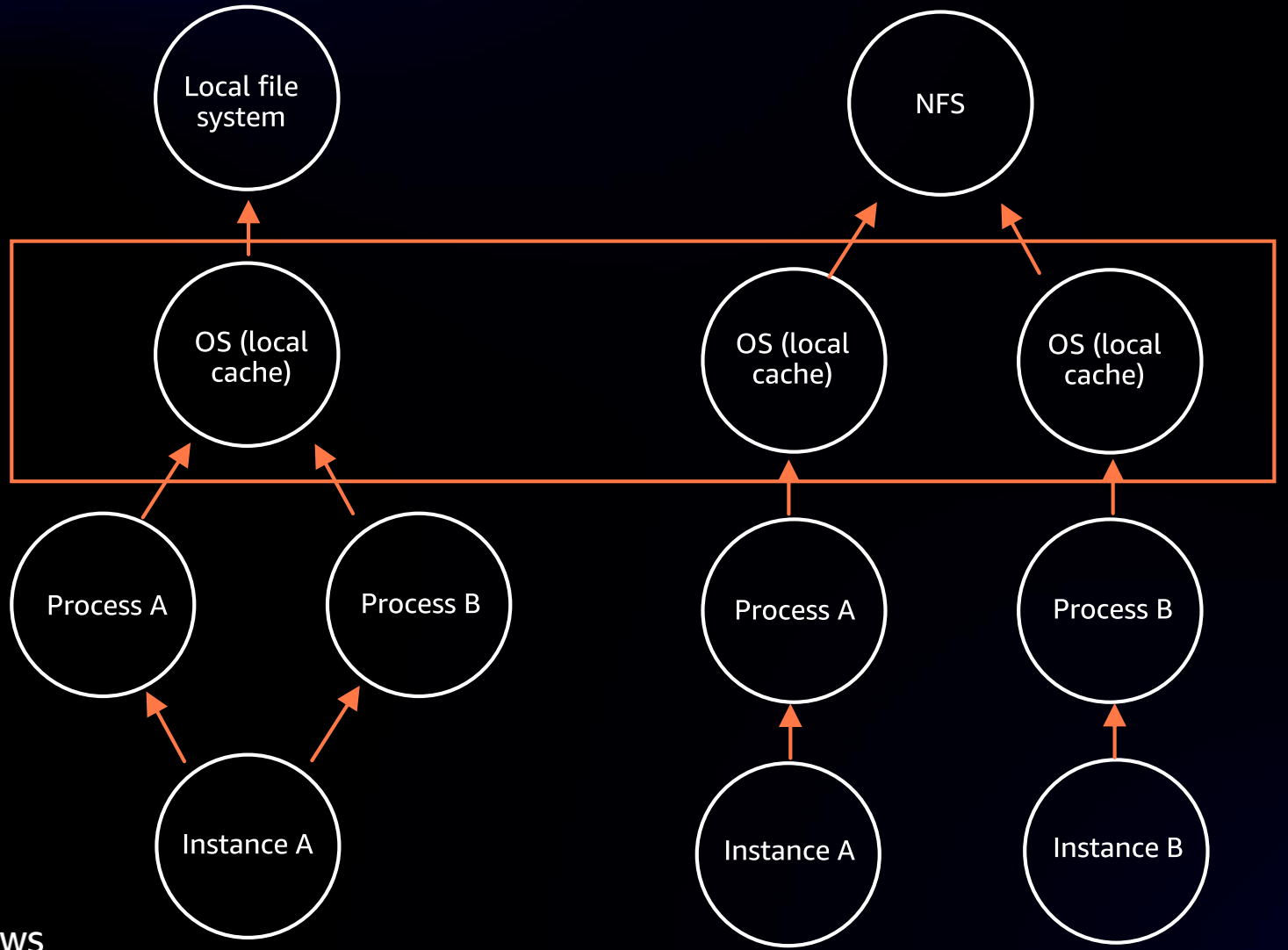
NFS overview





© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

File system caching



Q: “How and when does process B notice changes made by process A?”

Local FS: shared cache

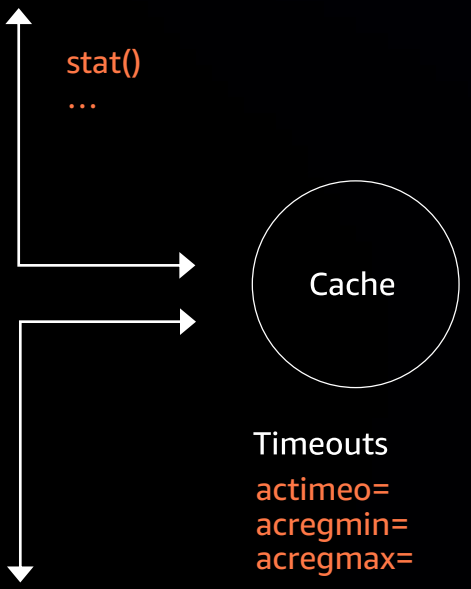
NFS: ?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

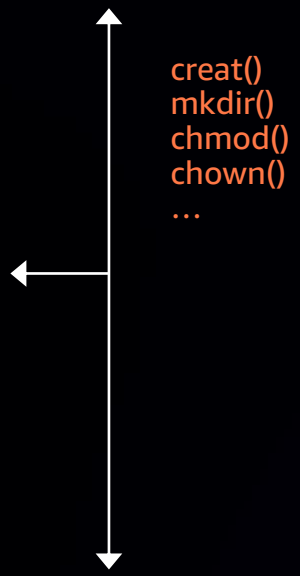
NFS caching

Metadata: time based

Metadata reads: cached



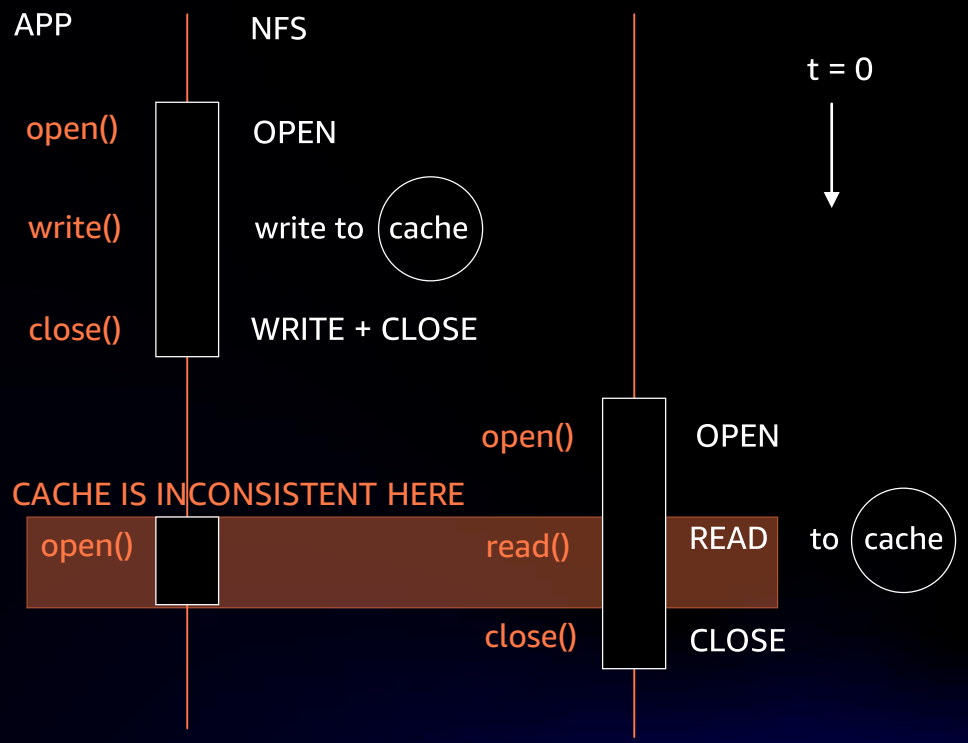
Metadata writes: synchronous



Data: “close to open”

Client A: write “foo.txt”

Client B: read “foo.txt”



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

NFS caching and small-file performance

- On local file systems, open() and close() are typically memory-based operations and take O (10 µs)
- Programmers are not used to thinking opening and closing files is expensive
- NFS “close-to-open” cache consistency means that both open() and close() require an O (ms) round trip
- When reading or writing small files, the two additional round trips are significant

File system	File opens/creates
Local file system (e.g., xfs, ext4)	O (100,000) files per second
NFS	O (100) files per second

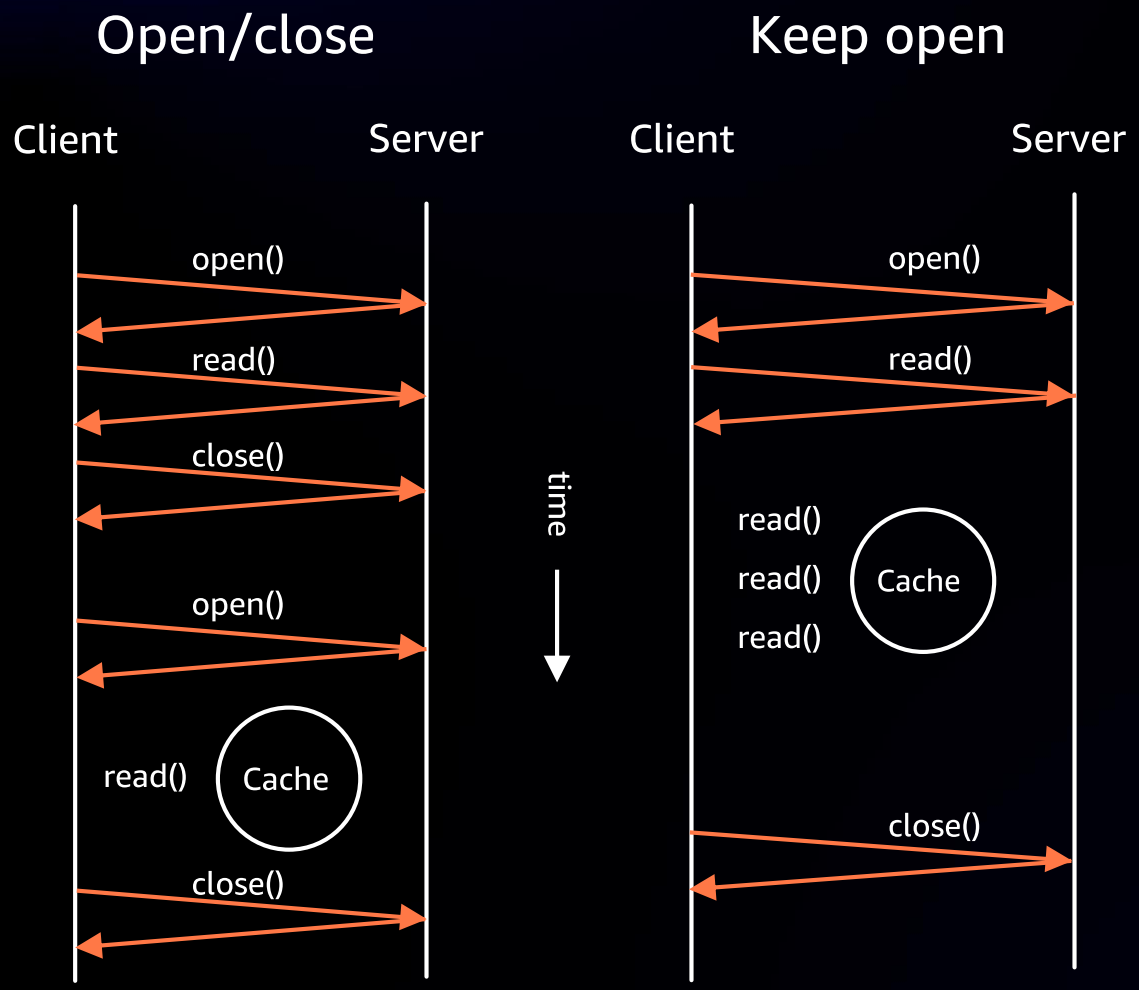
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing open/close

1. Don’t unnecessarily close files if you need them later
2. Use parallelism to reduce impact of round-trip times
3. Consider bundling reference data

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Don’t unnecessarily close files



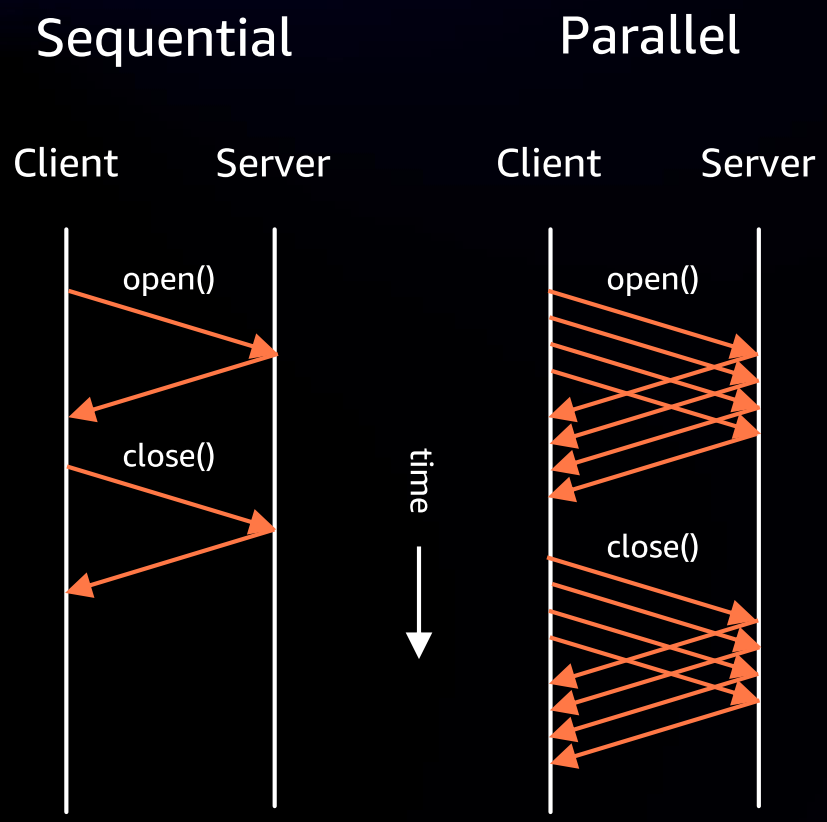
- Files are often read multiple times
- Keeping file descriptors open enables direct access to the local copy in the cache
- May need to update max file descriptors

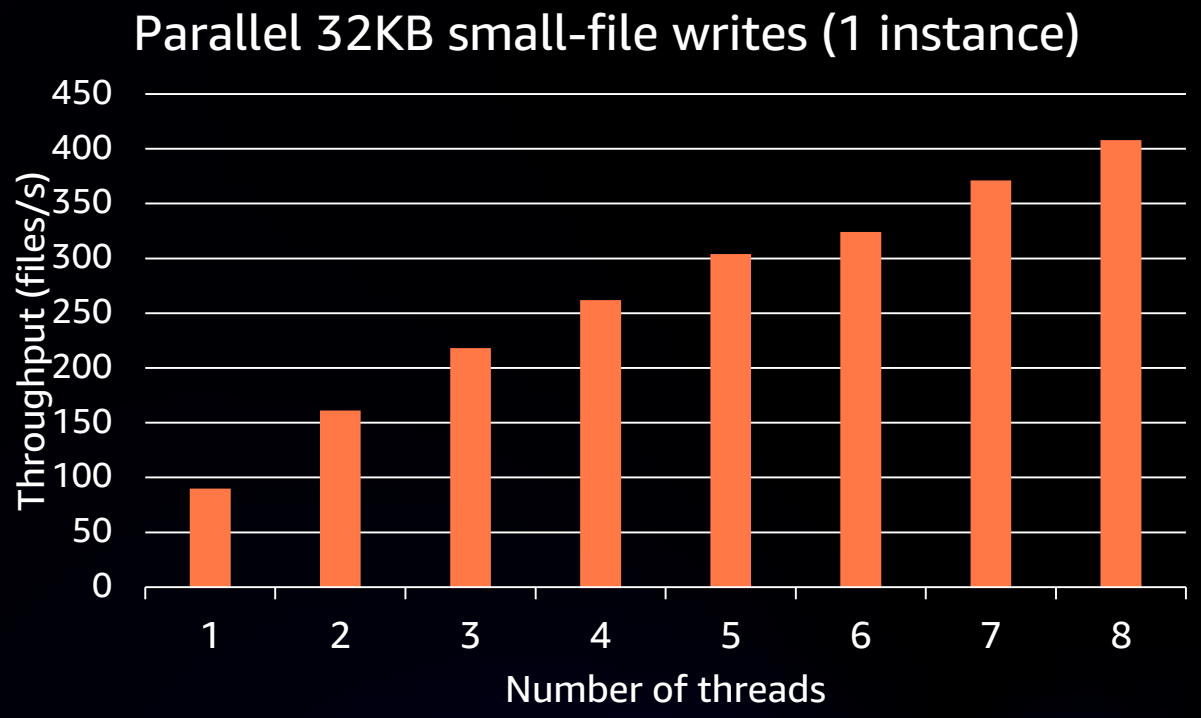
```
# Global limit
$ sudo sysctl -w fs.file-max=100000

# Per-user limit
$ sudo vi /etc/security/limits.conf
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimize open/close using parallelism



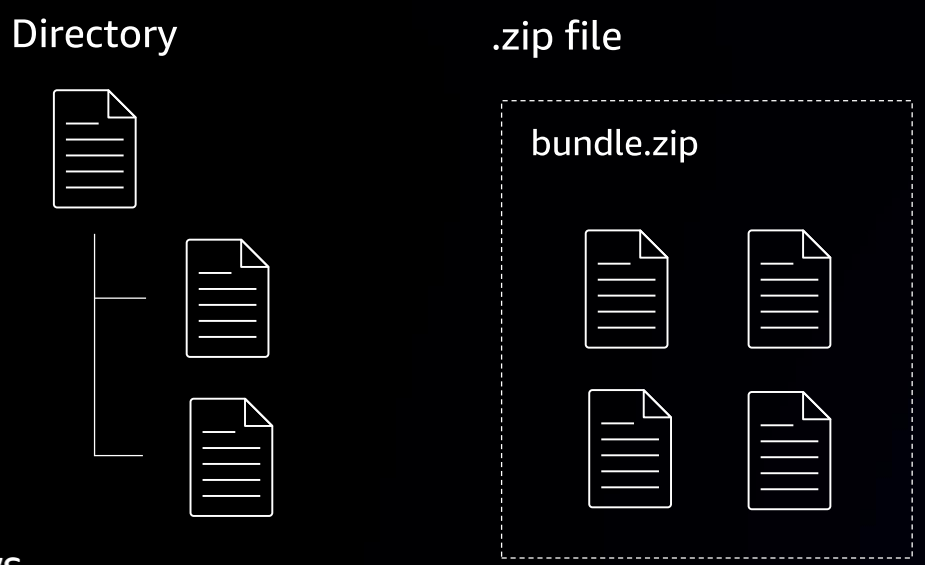


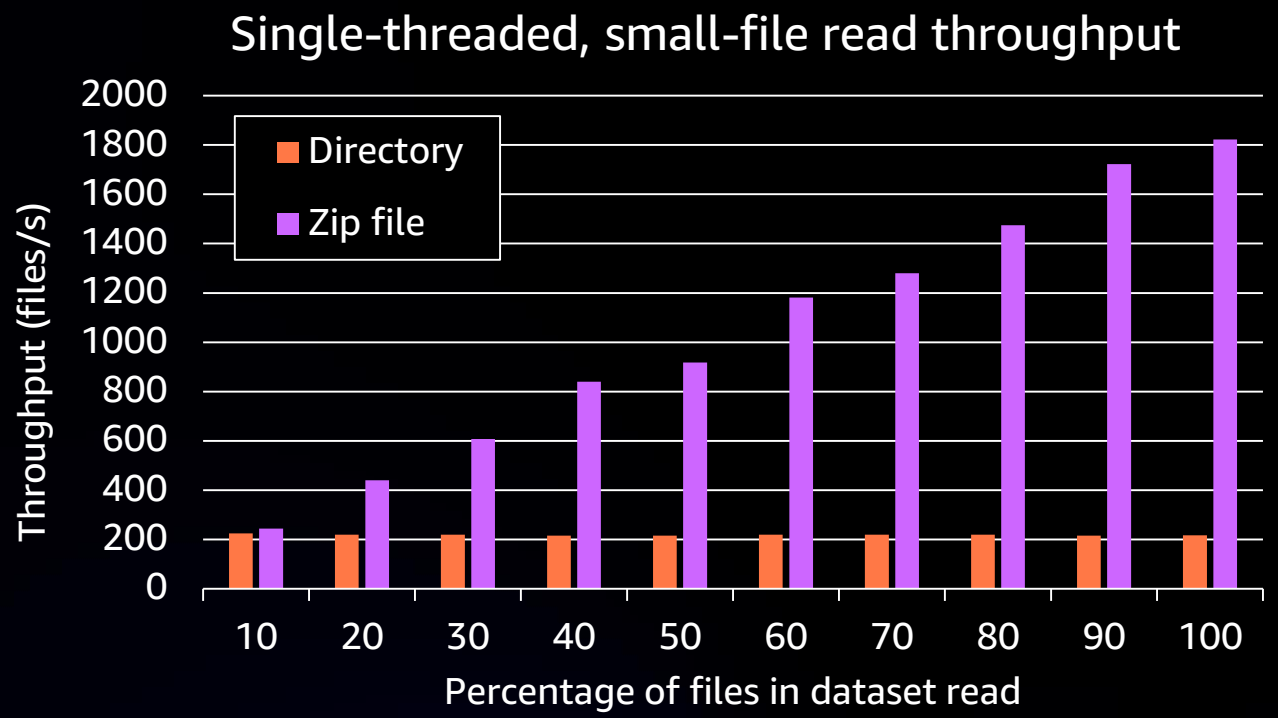

```
# xargs -n<number> -P<parallel> is your friend
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimize open/close by bundling files

- Some applications use a large set of small, mostly read-only reference files
- Bundling these in a .zip file allows you to read many files with one open/close
- .zip is a great format: central directory allows random access to individual files





© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Considerations for using directories

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Directory performance

Operation	Performance
Lookup by name	Constant time (ms)
List entries	25K–200K entries per second (readdirplus vs. readdir)
Add entry	340* entries per second
Delete entry	260* entries per second

- Lookups are fast for directories of any size
- Large directories are hard to manage (copy, list, delete)
- Linux NFS client can get stuck when listing very large directories that are being concurrently modified (fixed in kernel 5.11, backported to AL2 kernels 4.14, 5.4, and 5.10)

** Performance measured in November 2021 for EFS One Zone in us-east-1.*

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing directories

```
$ ls -l d/e/a/deadbeef.bin
```

- Keep directory sizes reasonable (under 10,000, if possible)
- Use (nested) subdirectories to store any number of files

- Do not get file attributes when listing a directory if not required, as they are not stored in the directory itself

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary of performance best practices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary of performance best practices

1. Select the desired storage class, performance and throughput modes
2. Use the most recent OS possible
3. Use default mount options
4. Improve small-file performance by reducing reopens, increasing parallelism, and bundling reference files where possible
5. Keep directories below 10,000 items, if possible

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Geert Jansen

gerardu@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.