# Filesystem Aging: It's More Usage than Fullness

**Alex Conway**, Eric Knorr, Yizheng Jiao, Michael A. Bender, William Jannen, Rob Johnson, Donald Porter, and Martin Farach-Colton

# What is filesystem aging?

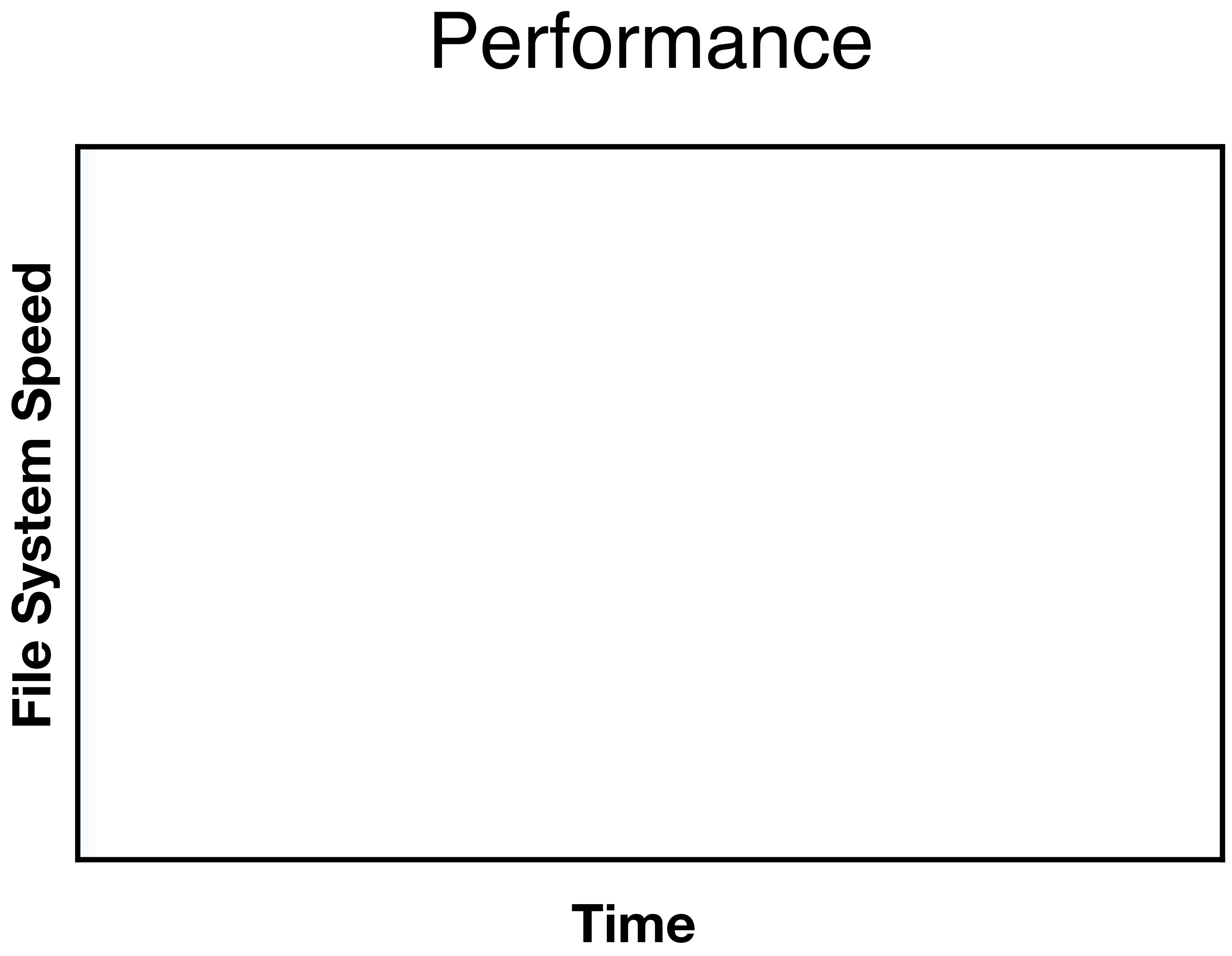
Aging is fragmentation over time

# What is filesystem aging?

Aging is fragmentation over time

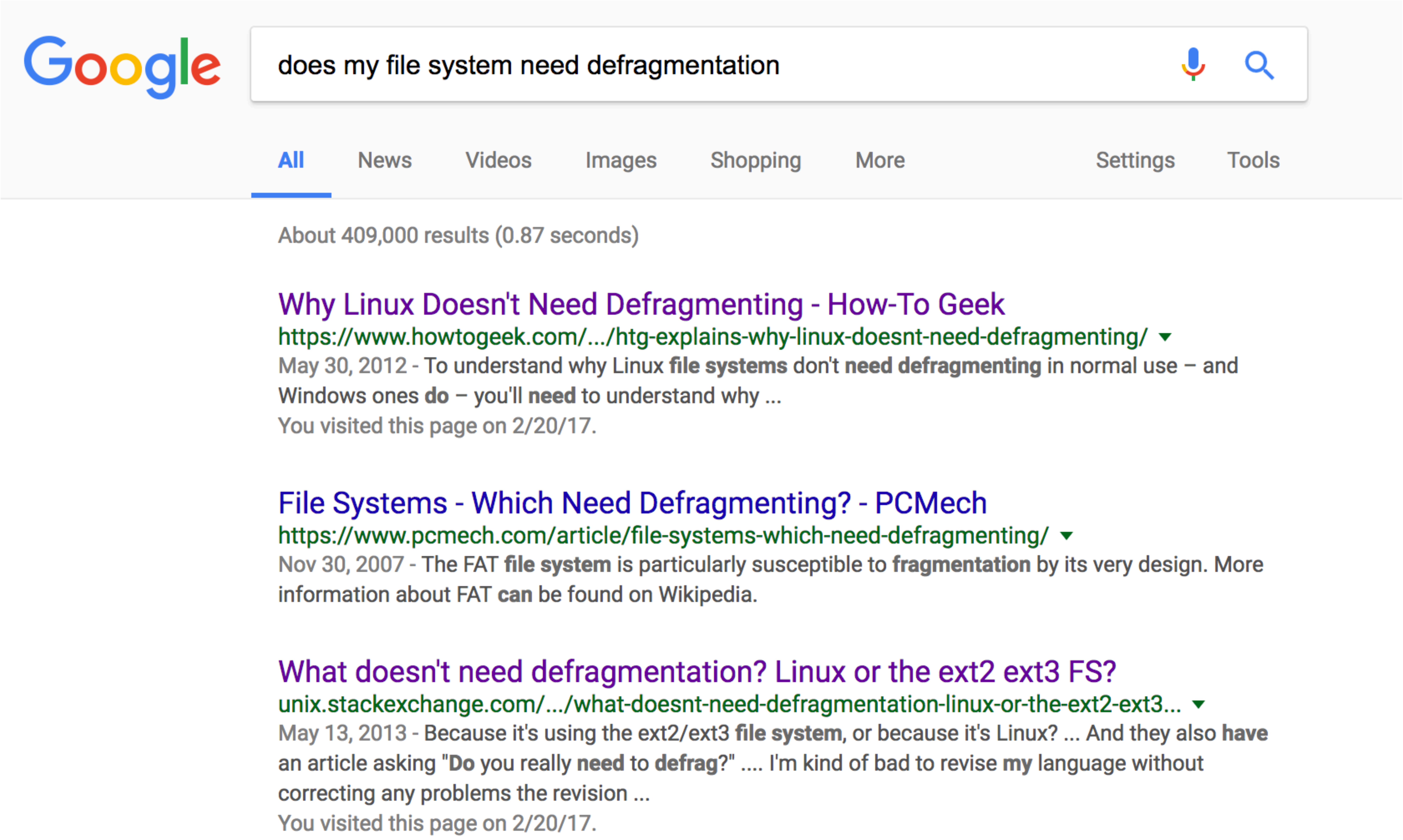
# What is filesystem aging?

Aging is fragmentation over time



# Is aging a problem?

# Is aging a problem?



does my file system need defragmentation

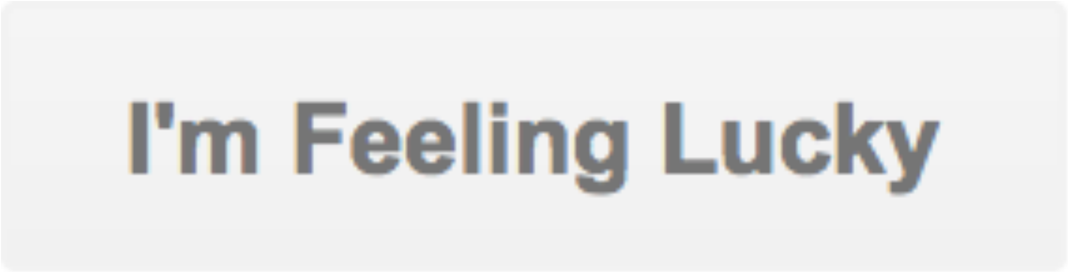
About 409,000 results (0.87 seconds)

Why Linux Doesn't Need Defragmenting - How-To Geek
https://www.howtogeek.com/.../htg-explains-why-linux-doesnt-need-defragmenting/
May 30, 2012 - To understand why Linux **file systems** don't **need defragmenting** in normal use – and Windows ones **do** – you'll **need** to understand why ...
You visited this page on 2/20/17.

File Systems - Which Need Defragmenting? - PCMech
https://www.pcmech.com/article/file-systems-which-need-defragmenting/
Nov 30, 2007 - The FAT **file system** is particularly susceptible to **fragmentation** by its very design. More information about FAT **can** be found on Wikipedia.

What doesn't need defragmentation? Linux or the ext2 ext3 FS?
unix.stackexchange.com/.../what-doesnt-need-defragmentation-linux-or-the-ext2-ext3...
May 13, 2013 - Because it's using the ext2/ext3 **file system**, or because it's Linux? ... And they also **have** an article asking "**Do** you really **need** to **defrag**?" .... I'm kind of bad to revise **my** language without correcting any problems the revision ...
You visited this page on 2/20/17.

# Is aging a problem?

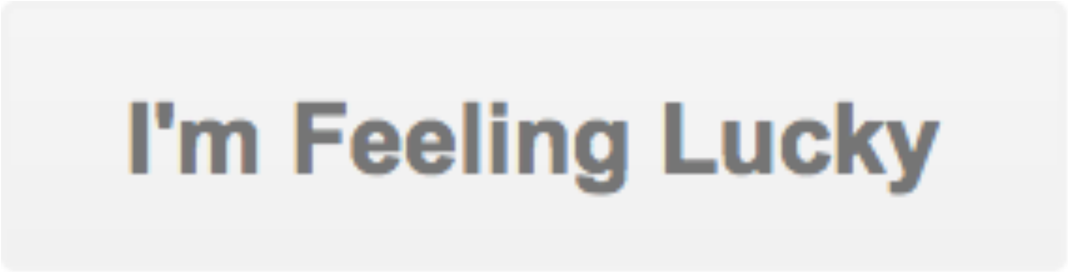


Chris Hoffman at howtogeek.com says:

"Linux's ext2, ext3, and ext4 file systems… [are] designed to avoid fragmentation in normal use."

"If you do have problems with fragmentation on Linux, you probably need a larger hard disk."

# Is aging a problem?



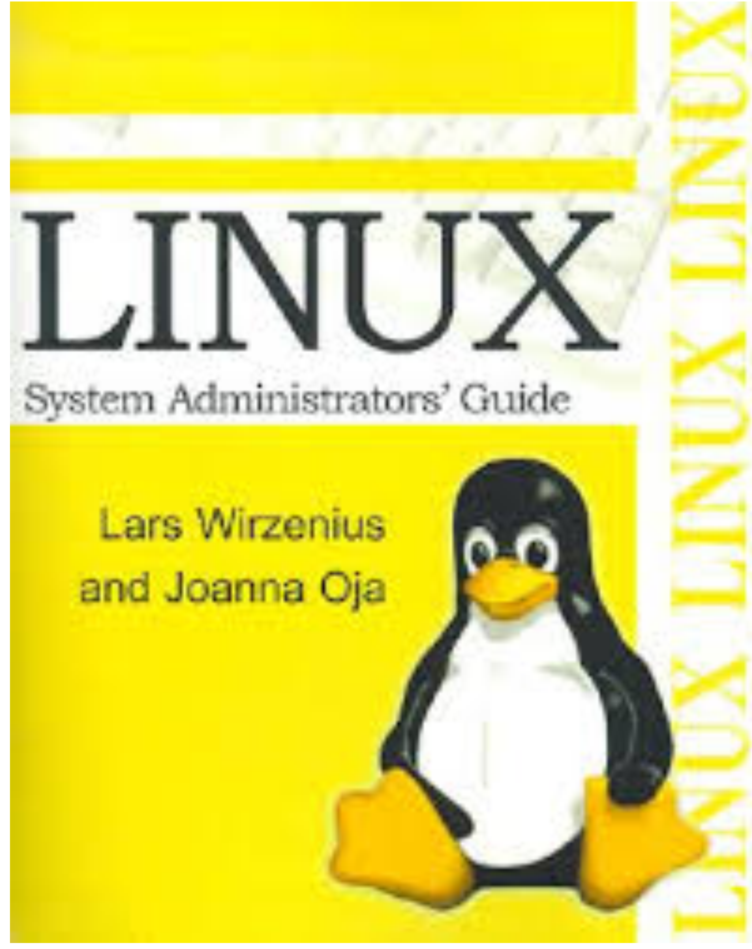
Chris Hoffman at howtogeek.com says:

“Linux’s ext2, ext3, and ext4 file systems… [are] designed to avoid fragmentation in normal use.”

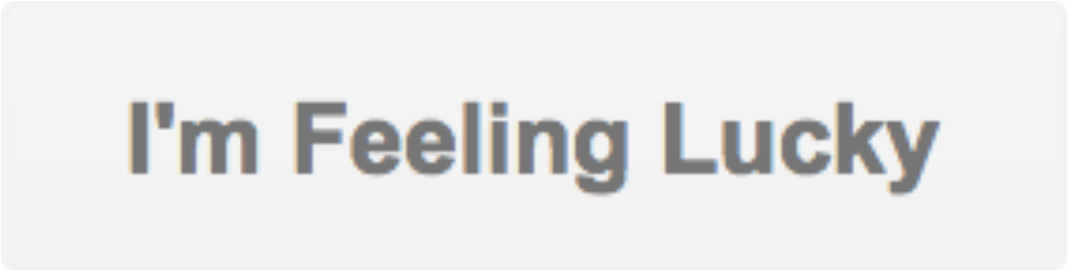
“If you do have problems with fragmentation on Linux, you probably need a larger hard disk.”

---

“Modern Linux filesystems keep fragmentation at a minimum…Therefore it is not necessary to worry about fragmentation in a Linux system.”



# Is aging a problem?



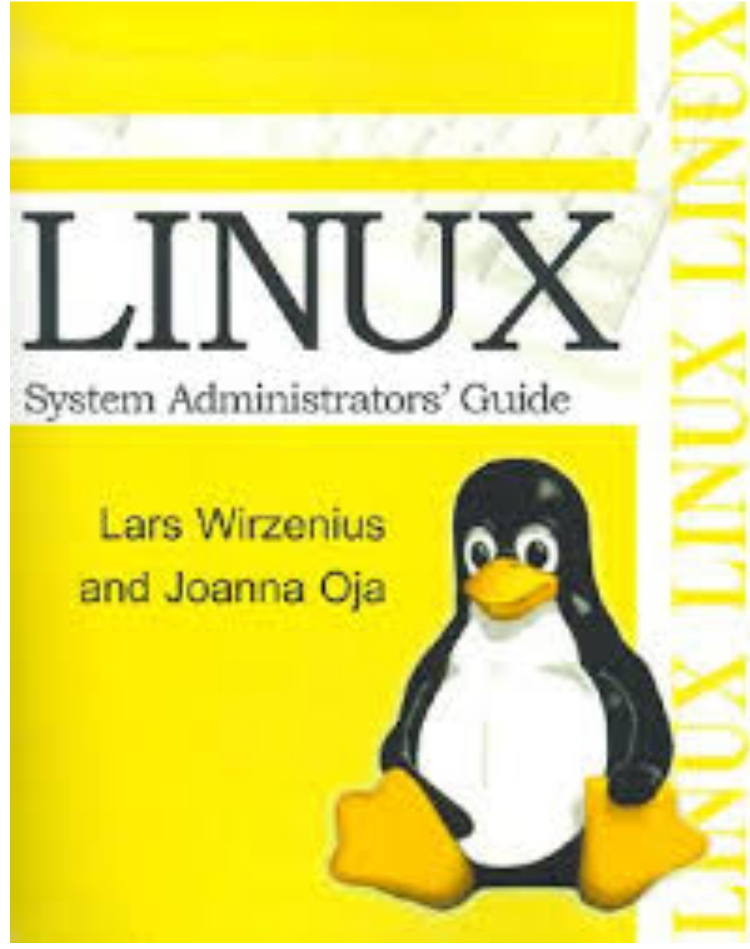
Chris Hoffman at howtogeek.com says:

“Linux’s ext2, ext3, and ext4 file systems… [are] designed to avoid fragmentation in normal use.”

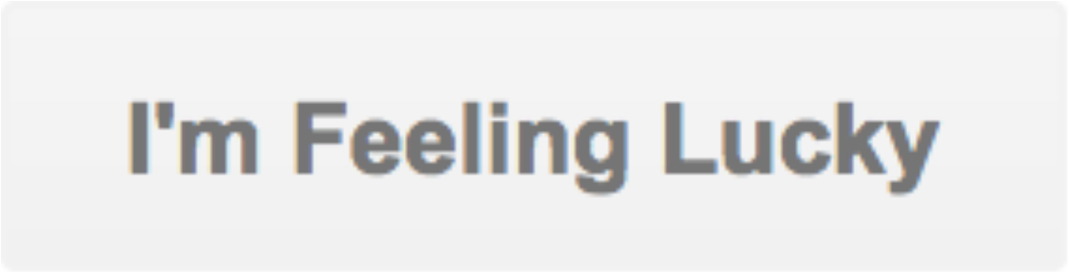
**Aging is not a problem**

“If you do have problems with fragmentation on Linux, you probably need a larger hard disk.”

“Modern Linux filesystems keep fragmentation at a minimum…Therefore it is not necessary to worry about fragmentation in a Linux system.”



# Is aging a problem?



I'm Feeling Lucky

Chris Hoffman at howtogeek.com says:

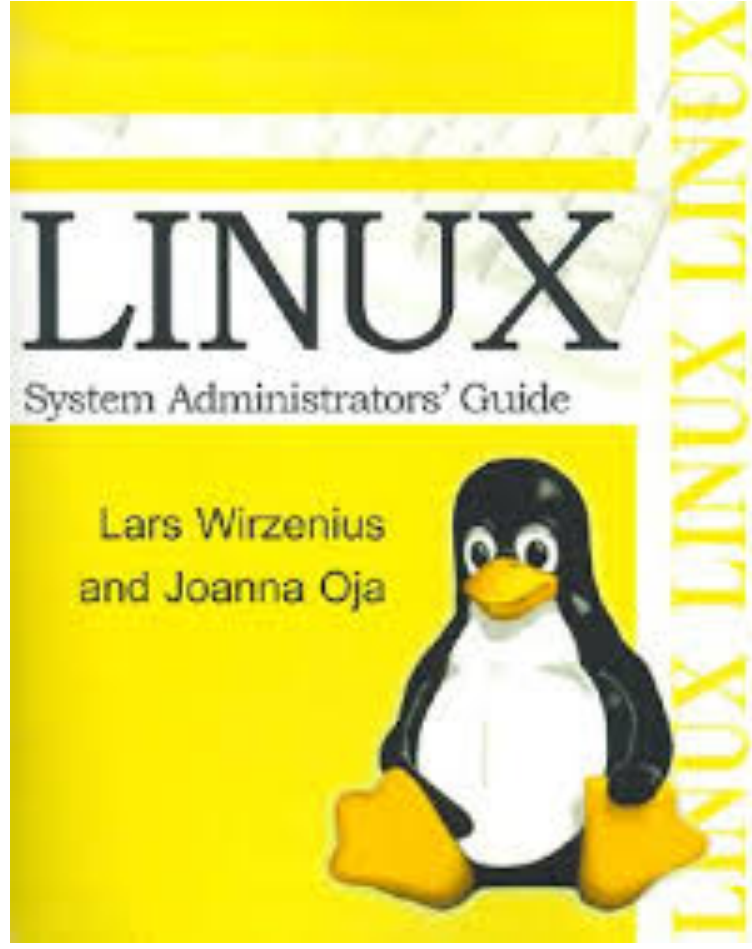
"Linux's ext2, ext3, and ext4 file systems… [are] designed to avoid fragmentation in normal use."

"If you do have problems with fragmentation on Linux, you probably need a larger hard disk."

**Aging is not a problem**

**(unless your disk is full)**

"Modern Linux filesystems keep fragmentation at a minimum…Therefore it is not necessary to worry about fragmentation in a Linux system."



# Recent work:
# Aging is a problem!

# Recent work on aging

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and
happens quickly even if your disk is almost empty.

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and
happens quickly even if your disk is almost empty.

**“Geriatrix: Aging what you see and what you don’t see. A file system aging approach for modern storage systems”**
Kadekodi et. al.   ATC 2018

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and happens quickly even if your disk is almost empty.

**“Geriatrix: Aging what you see and what you don’t see. A file system aging approach for modern storage systems”**
Kadekodi et. al.   ATC 2018

Real world file systems show a lot of aging, as well as free space fragmentation.

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and happens quickly even if your disk is almost empty.

**“Geriatrix: Aging what you see and what you don’t see. A file system aging approach for modern storage systems”**
Kadekodi et. al.   ATC 2018

Real world file systems show a lot of aging, as well as free space fragmentation. ←

**Claim:**
**Only happens when the disk is full**

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and
happens quickly even if your disk is almost empty.

Aging is a problem

**“Geriatrix: Aging what you see and what you don’t see. A file system aging approach for modern storage systems”**
Kadekodi et. al.   ATC 2018

Real world file systems show a lot of aging,
as well as free space fragmentation.

**Claim:**
**Only happens when the disk is full**

# Recent work on aging

**“File Systems Fated for Senescence? Nonsense, says Science!”**
Conway et. al.   FAST 2017

On modern filesystems, aging is severe and happens quickly even if your disk is almost empty.

**“Geriatrix: Aging what you see and what you don’t see. A file system aging approach for modern storage systems”**
Kadekodi et. al.   ATC 2018

Real world file systems show a lot of aging, as well as free space fragmentation. ←

Aging is a problem

Disk fullness ????

**Claim:**
**Only happens when the disk is full**

# Flavors of Aging

# Flavors of Aging

**3 Flavors of Aging**

Read Aging

Write Aging

Free Space Fragmentation

# Flavors of Aging

## 3 Flavors of Aging

### Read Aging

Fragmentation of pages which are read together

### Write Aging

Fragmentation of pages which are written together

OR

Additional work when writing

### Free Space Fragmentation

Fragmentation of the available free space

# Flavors of Aging

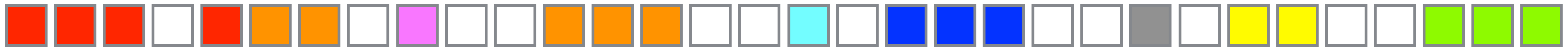
Free Space Fragmentation

Fragmentation of the available free space

**Different types of aging interact**

# Flavors of Aging

Free Space Fragmentation



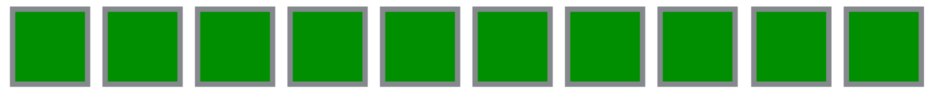
A filesystem:
each square represents a page,
different colors are different files

Fragmentation of the available free space

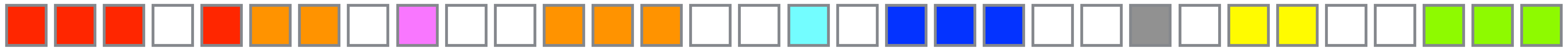
**Different types of aging interact**

# Flavors of Aging

How do we write this large file?



Free Space
Fragmentation



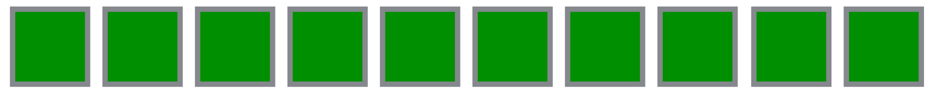
Fragmentation of the
available free space

A filesystem:
each square represents a page,
different colors are different files

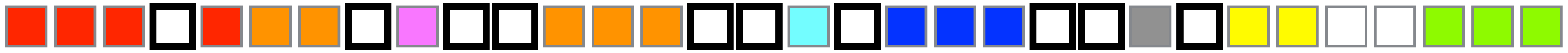
**Different types of aging interact**

# Flavors of Aging

How do we write this large file?



write it to available blocks



A filesystem:
each square represents a page,
different colors are different files

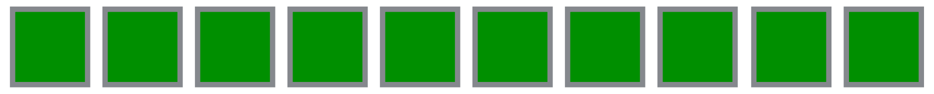
Free Space
Fragmentation

Fragmentation of the
available free space

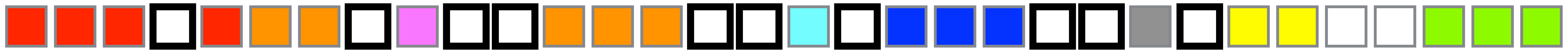
**Different types of aging interact**

# Flavors of Aging

How do we write this large file?



write it to available blocks → both read and write aging



A filesystem:
each square represents a page,
different colors are different files

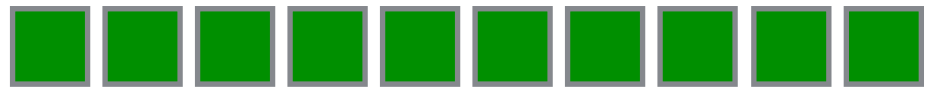
Free Space Fragmentation

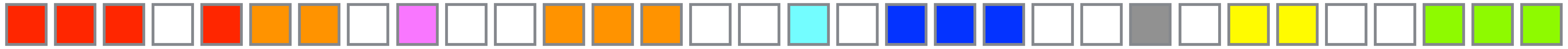
Fragmentation of the available free space

**Different types of aging interact**

# Flavors of Aging

How do we write this large file?





A filesystem:
each square represents a page,
different colors are different files

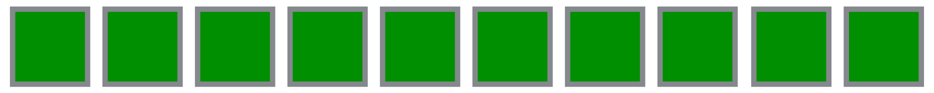
Free Space
Fragmentation

Fragmentation of the
available free space

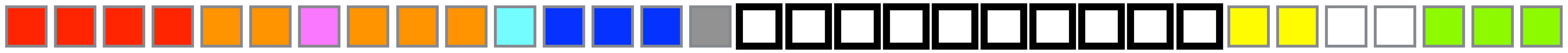
**Different types of aging interact**

# Flavors of Aging

How do we write this large file?



defragment the free space



A filesystem:
each square represents a page,
different colors are different files

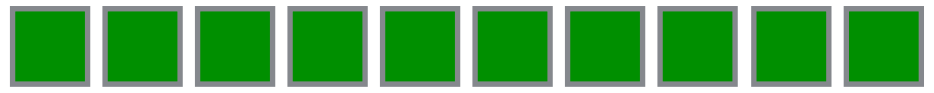
Free Space Fragmentation

Fragmentation of the available free space

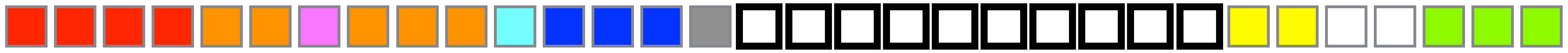
**Different types of aging interact**

# Flavors of Aging

How do we write this large file?



defragment the free space → lots of write aging



A filesystem:
each square represents a page,
different colors are different files

Free Space Fragmentation

Fragmentation of the available free space

**Different types of aging interact**

# How much aging is caused by disk fullness?

# This Work

This work tries to answer:
How much aging is caused by disk fullness?

# This Work

This work tries to answer:
How much aging is caused by disk fullness?

Hypothesis:
A lot

# This Work

This work tries to answer:
How much aging is caused by disk fullness?

Hypothesis:
A lot

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

# Git Replay Benchmark

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**

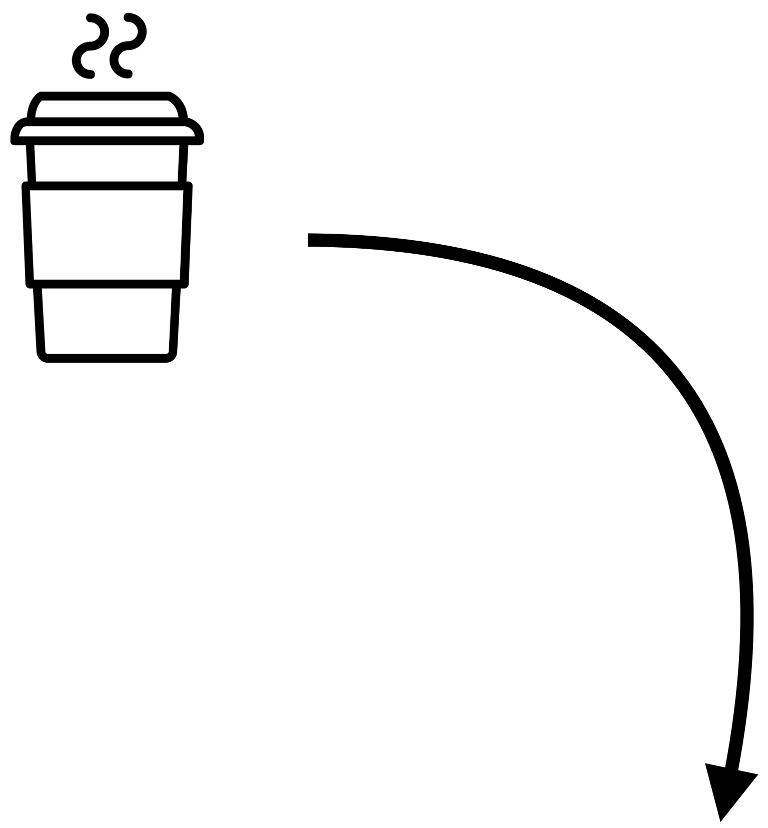
```
get coffee
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**



```
git pull
```

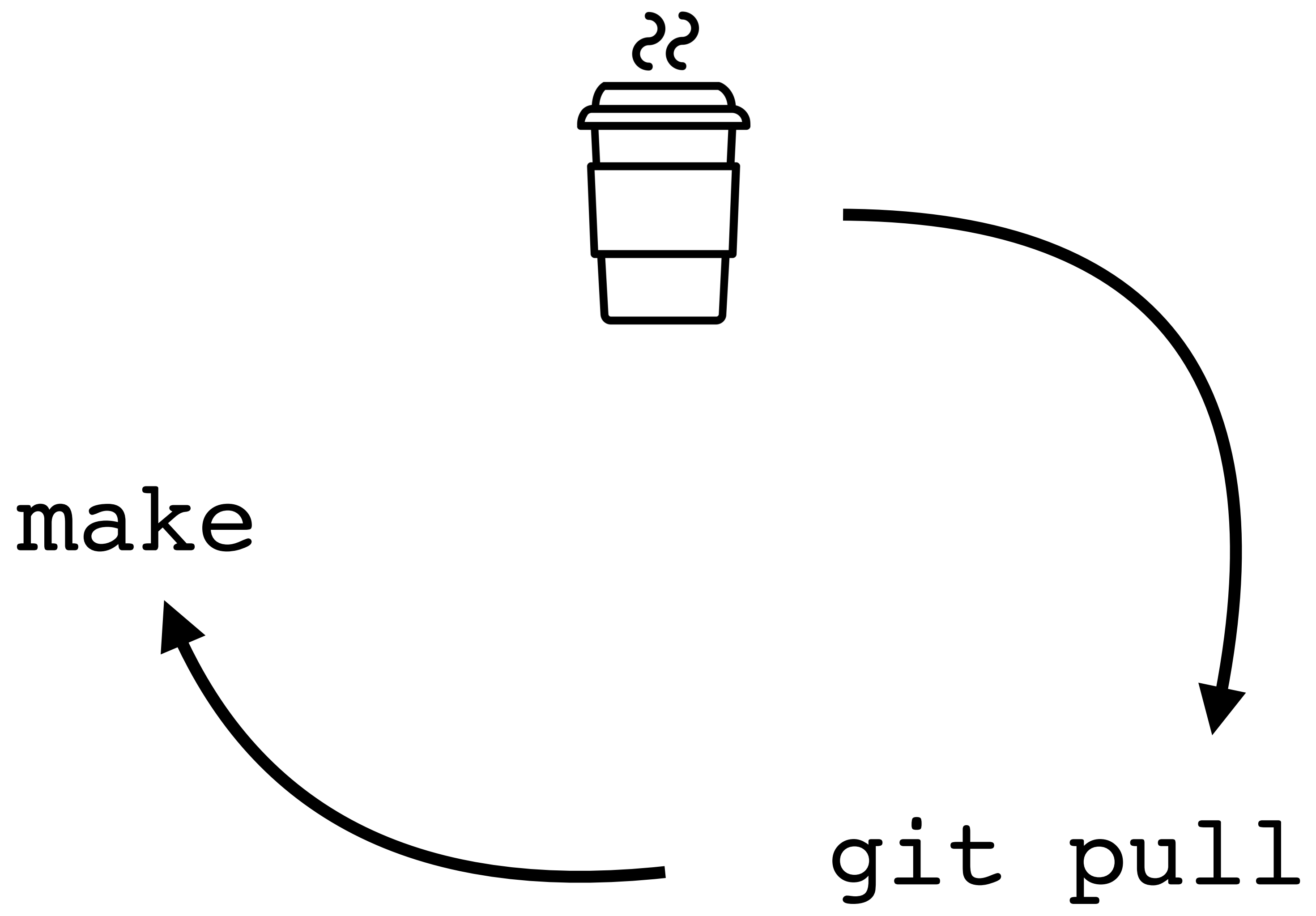
```
get coffee
git pull
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**



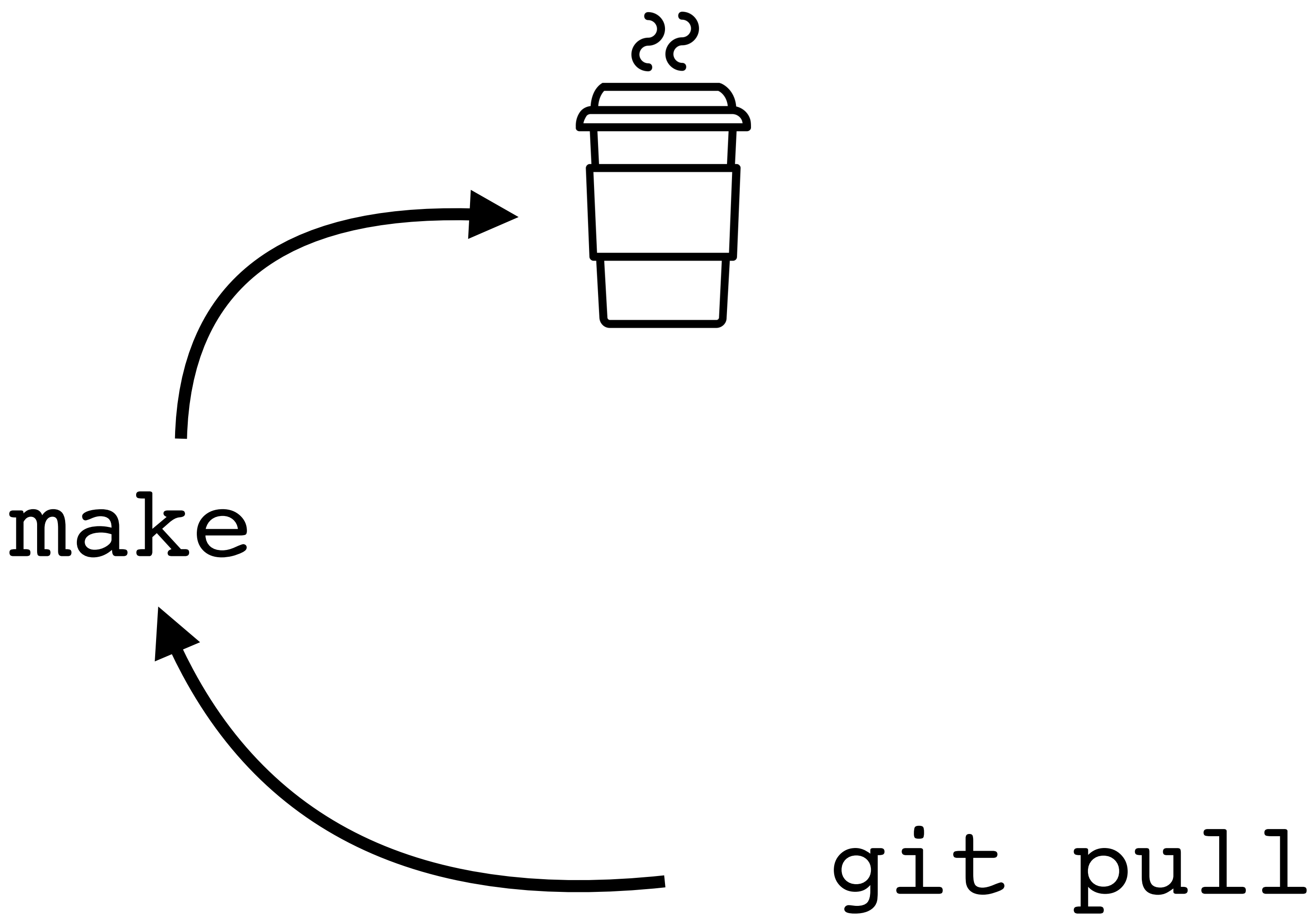
```
get coffee
git pull
make
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**



make

git pull

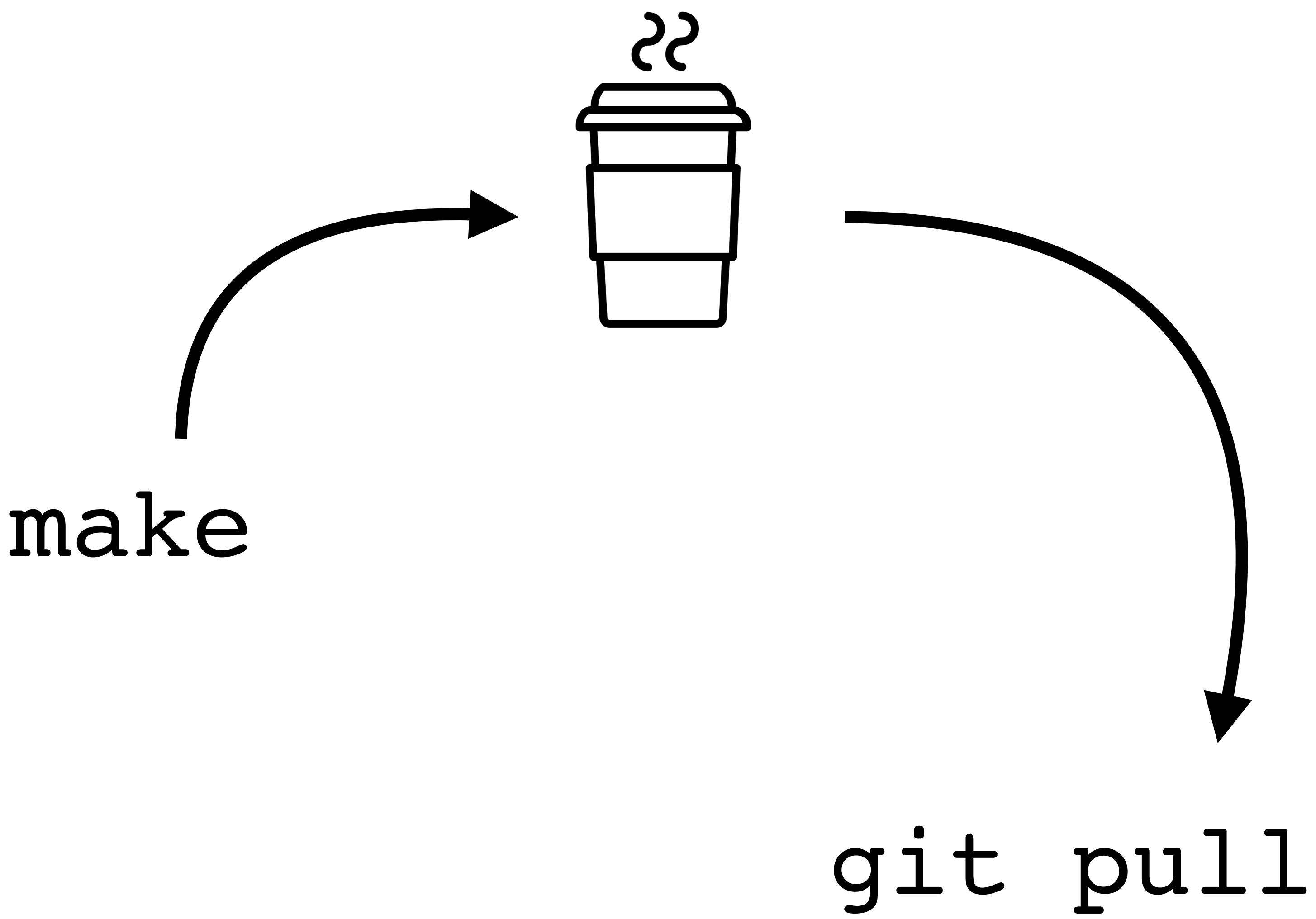
```
get coffee
git pull
make
get coffee
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let's model a very simple case: Developers**



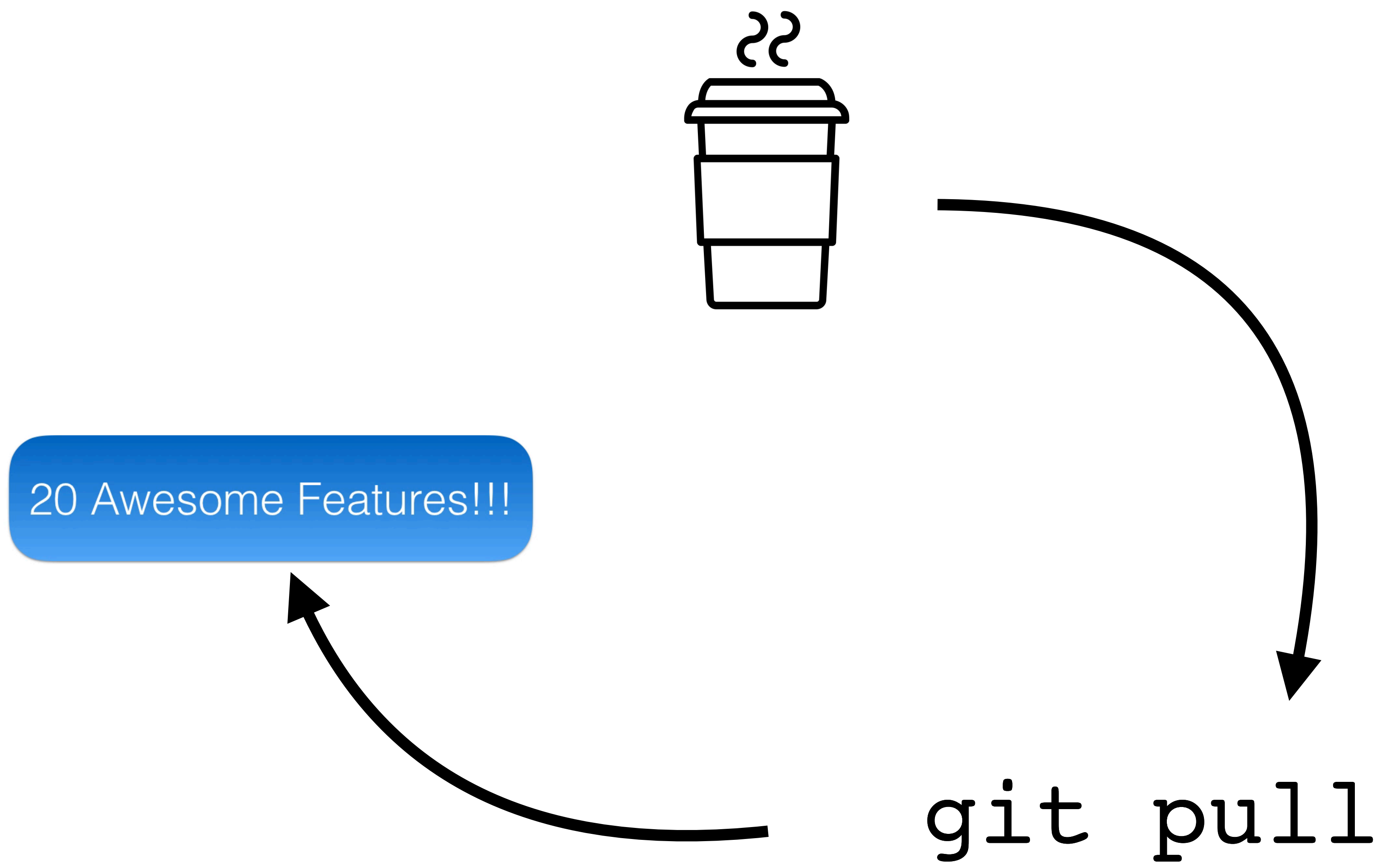
```
get coffee
git pull
make
get coffee
git pull
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let's model a very simple case: Developers**



20 Awesome Features!!!

git pull

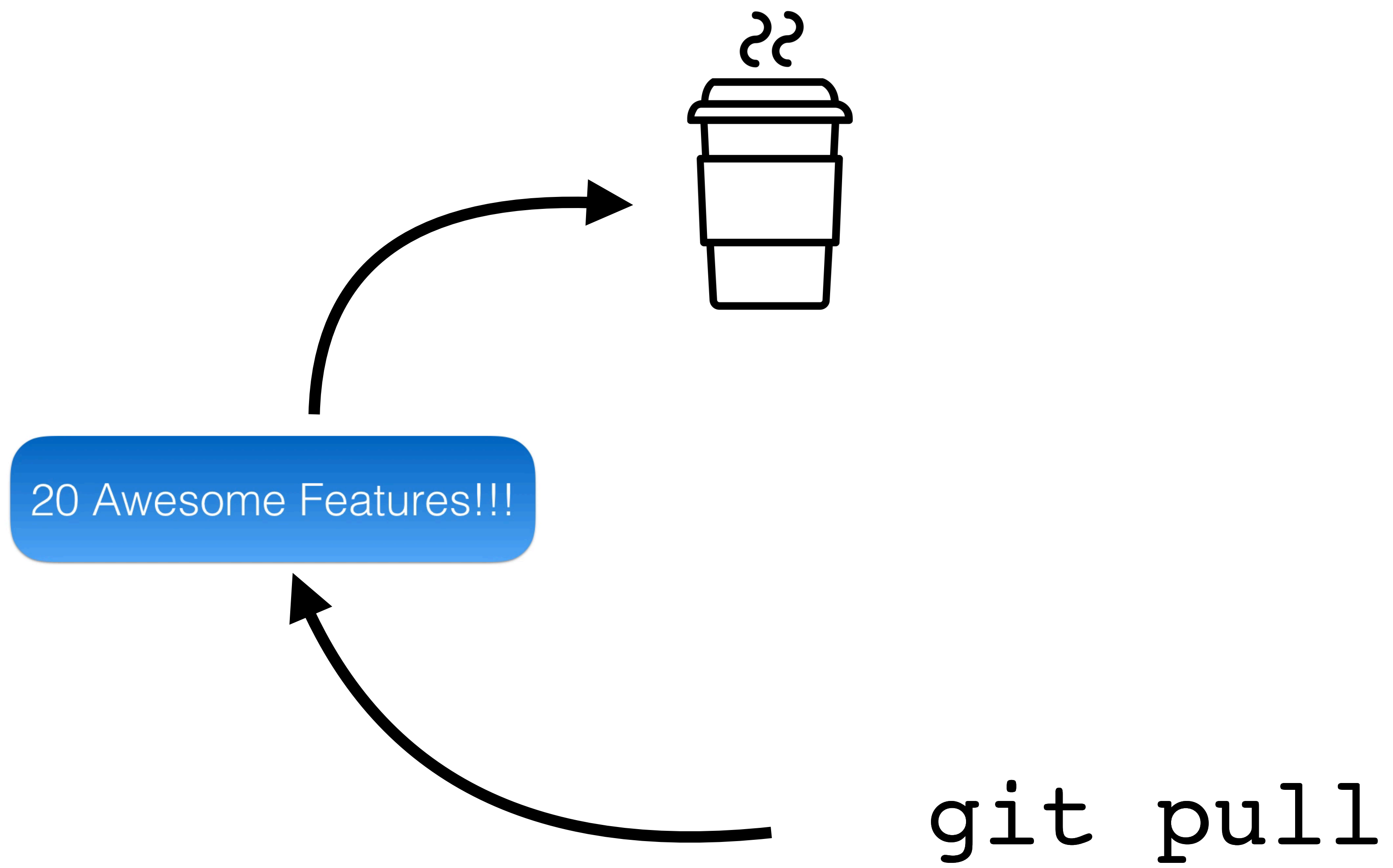
```
get coffee
git pull
make
get coffee
git pull
add awesome features
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let's model a very simple case: Developers**



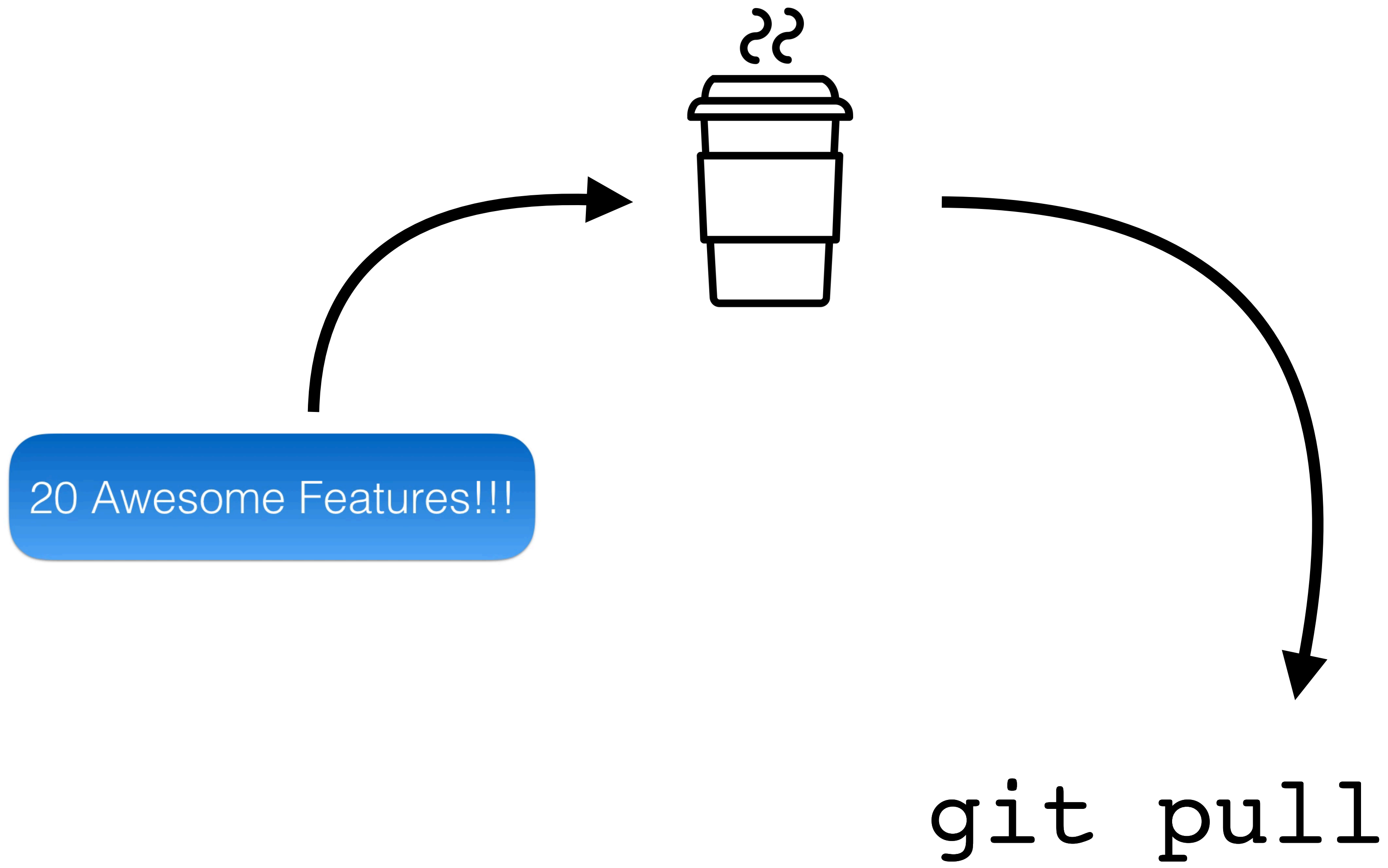
```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let’s model a very simple case: Developers**



20 Awesome Features!!!

git pull

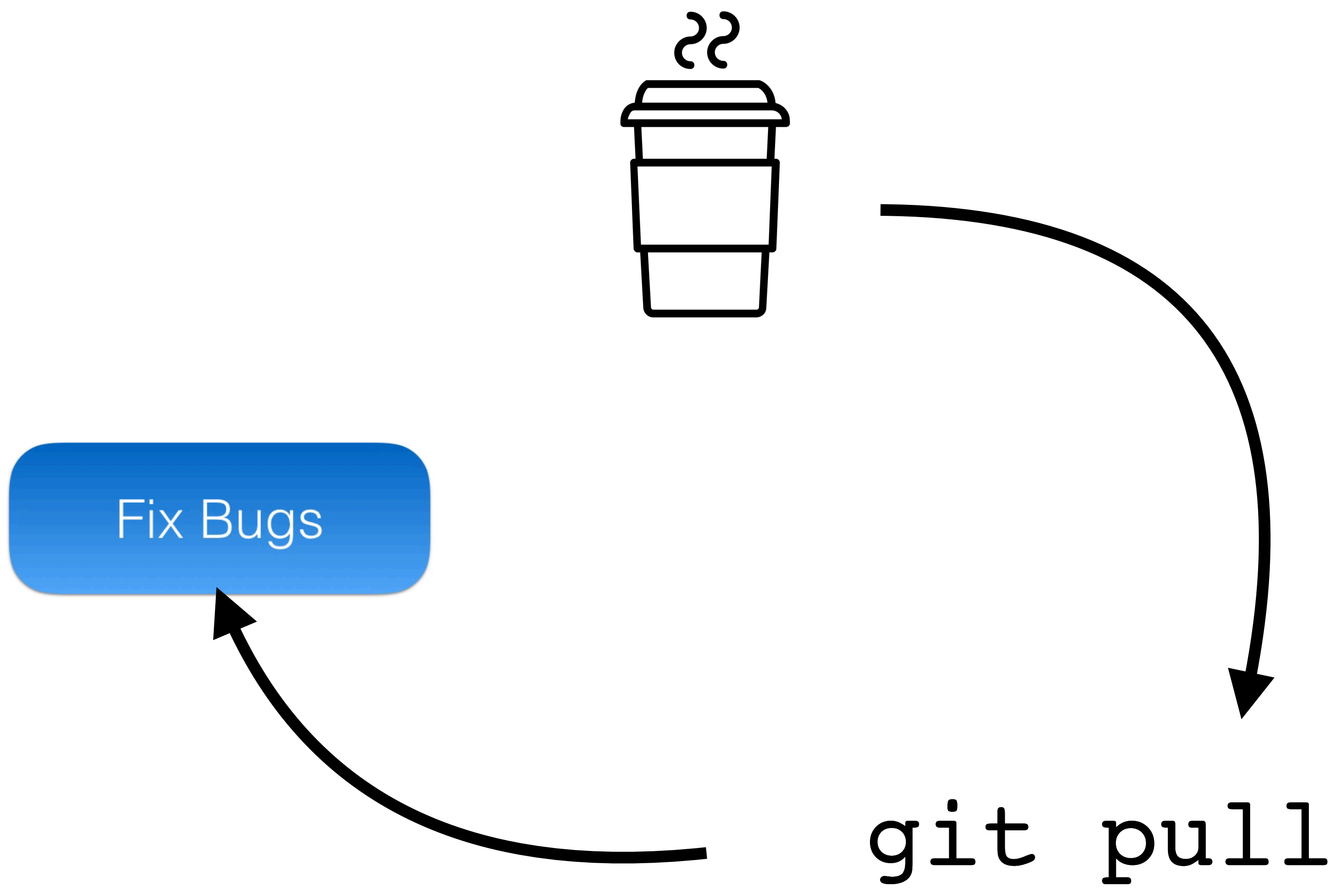
```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
git pull
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let's model a very simple case: Developers**



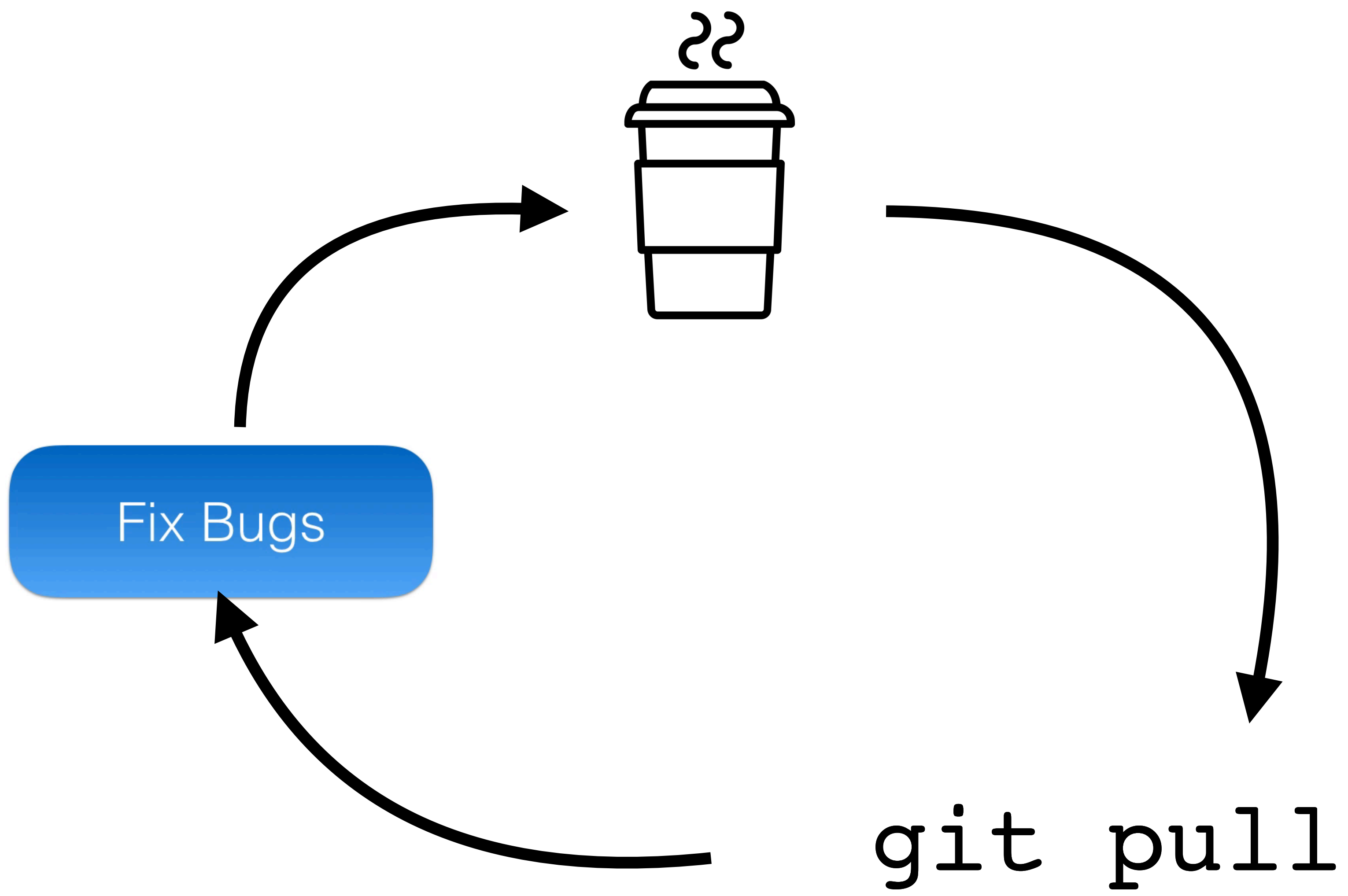
```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
git pull
fix bugs
. . .
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

**Let's model a very simple case: Developers**



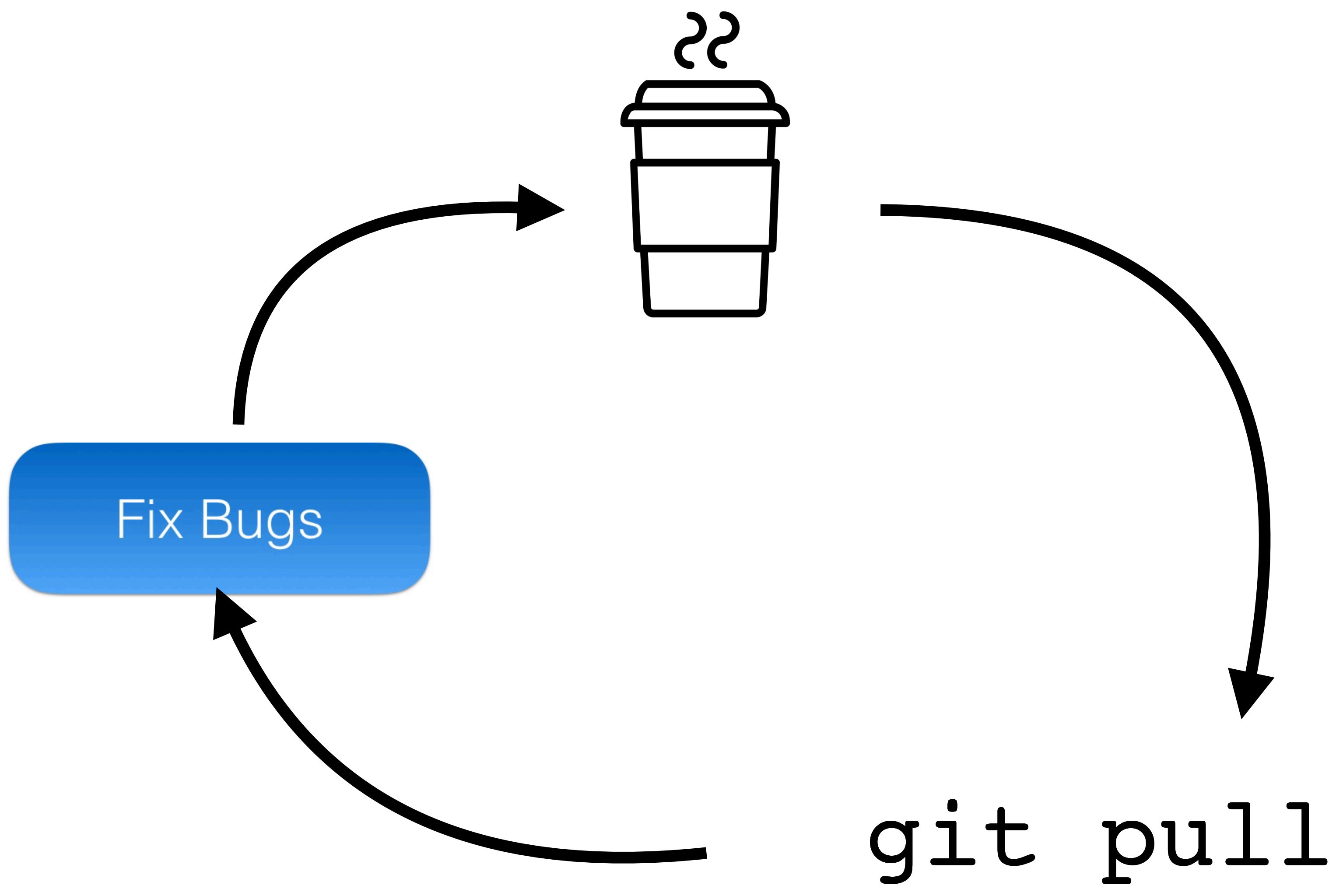
```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
git pull
fix bugs
. . .
```

# Git Replay Benchmark

**We need a workload that:**

- reflects actual use over many years
- can be generated and replayed quickly
- can operate on a nearly full disk

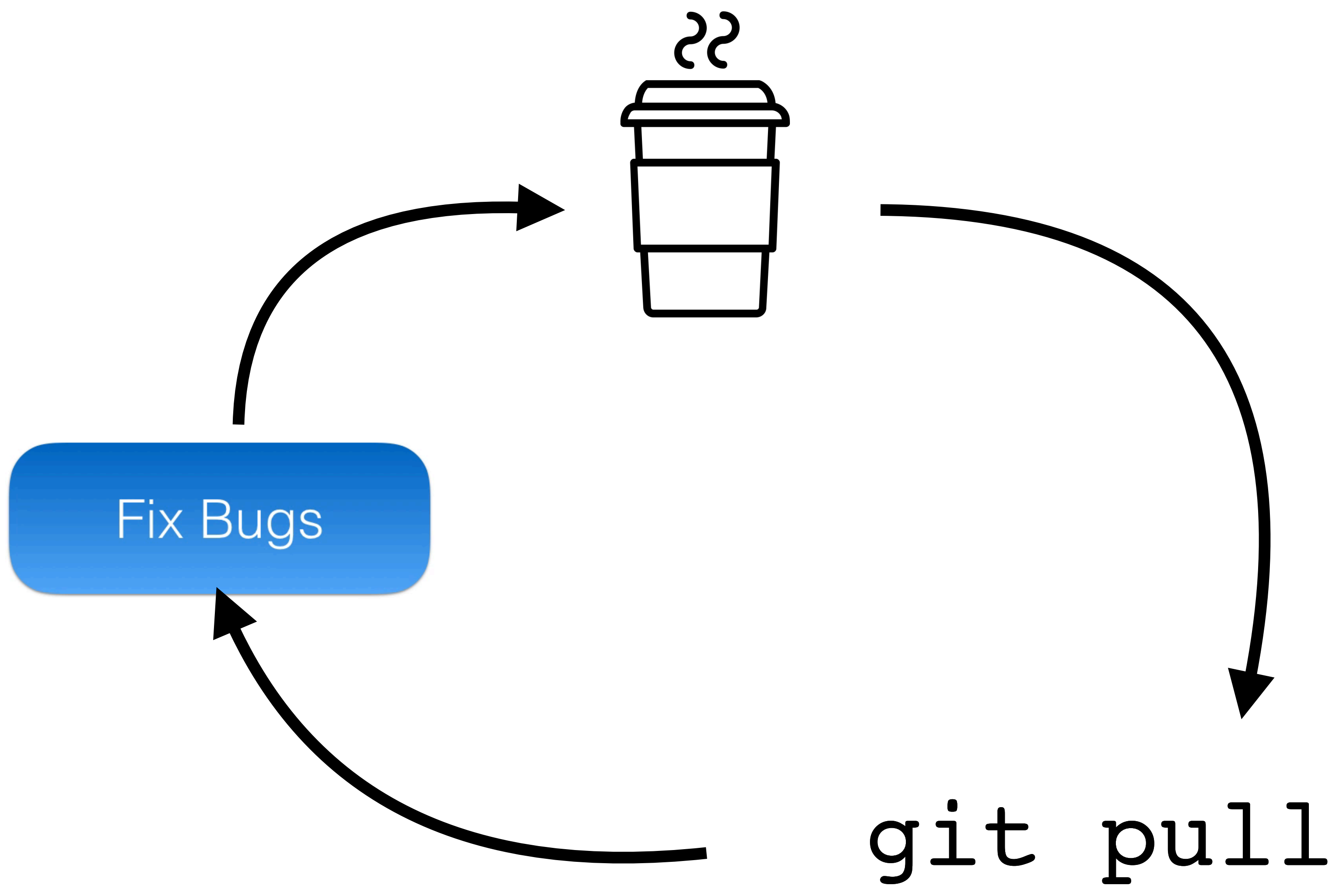
**Let's model a very simple case: Developers**



```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
git pull
fix bugs
. . .
```

# Git Replay Benchmark

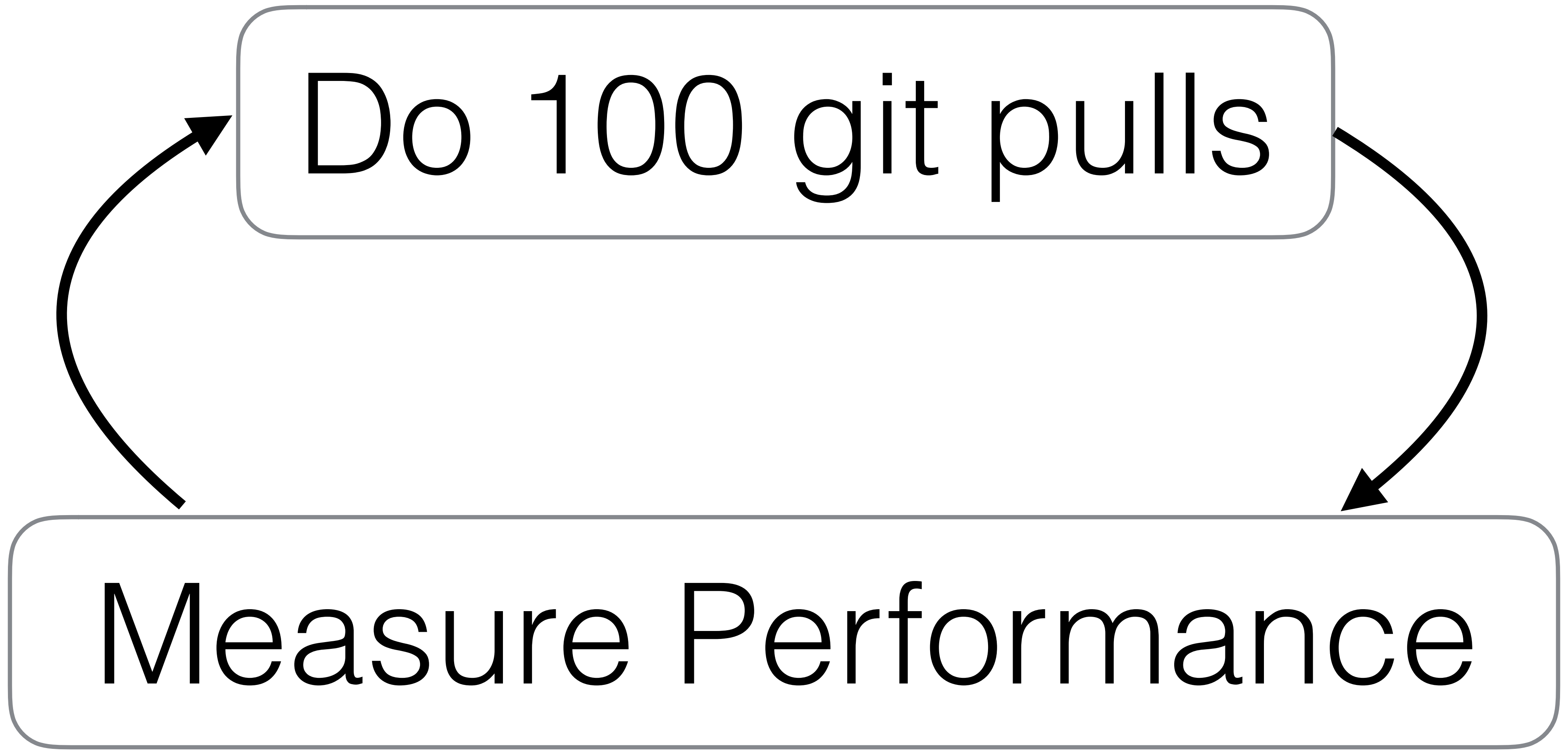
We can simulate a developer by replaying Git histories



```
get coffee
git pull
make
get coffee
git pull
add awesome features
get coffee
git pull
fix bugs
. . .
```

# Git Replay Benchmark

Use the Linux kernel repo from github.com

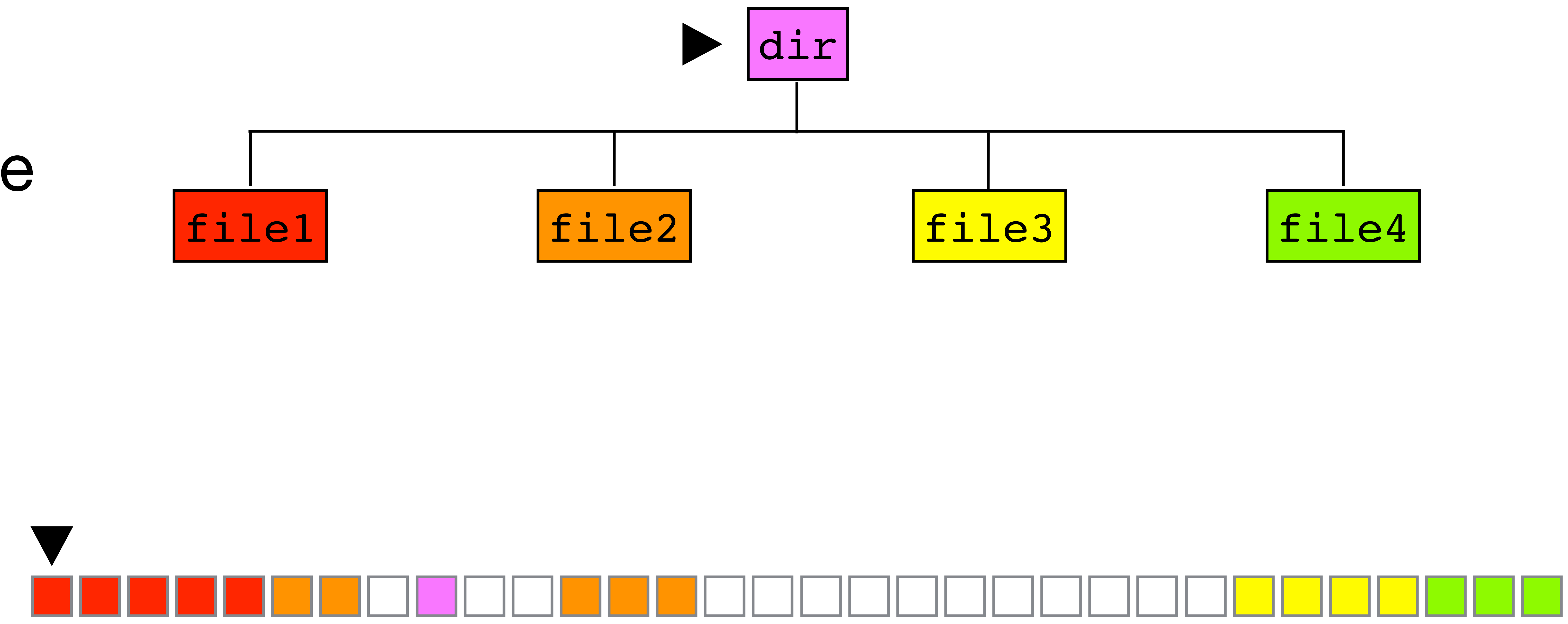


# How to Measure Aging

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

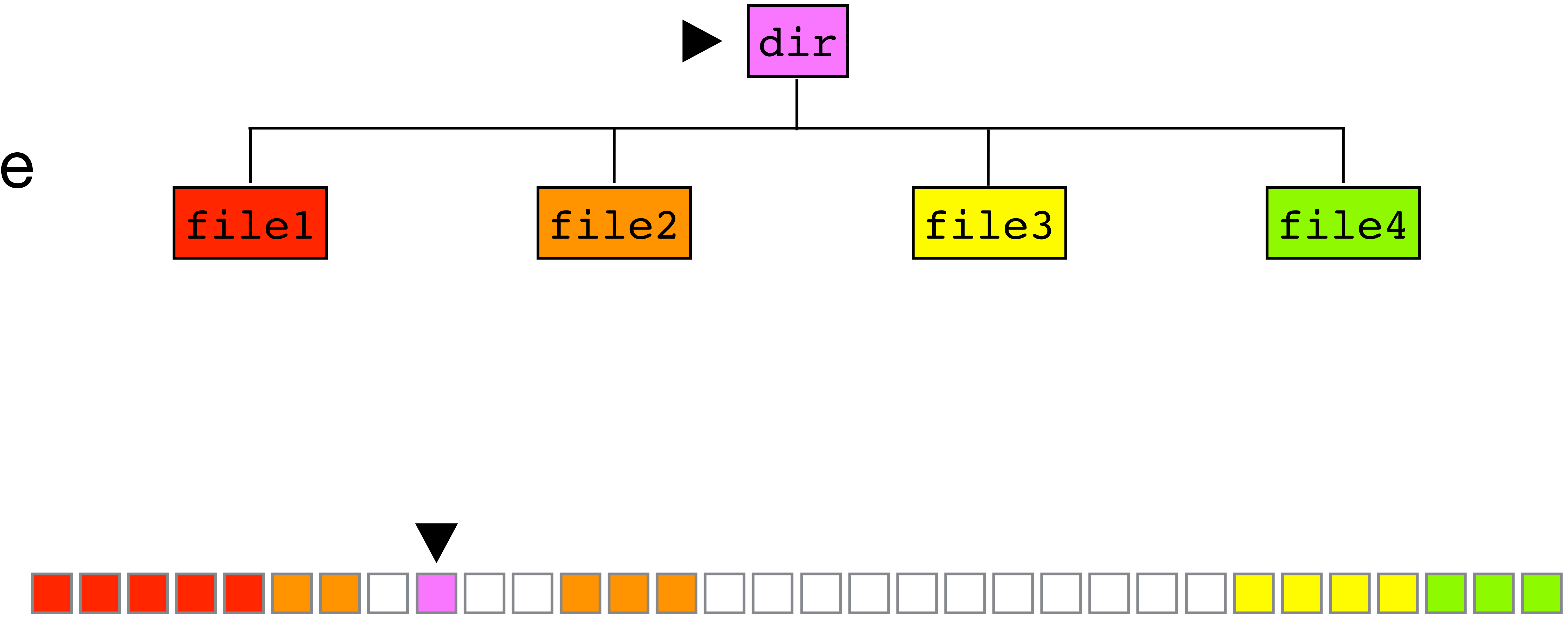


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

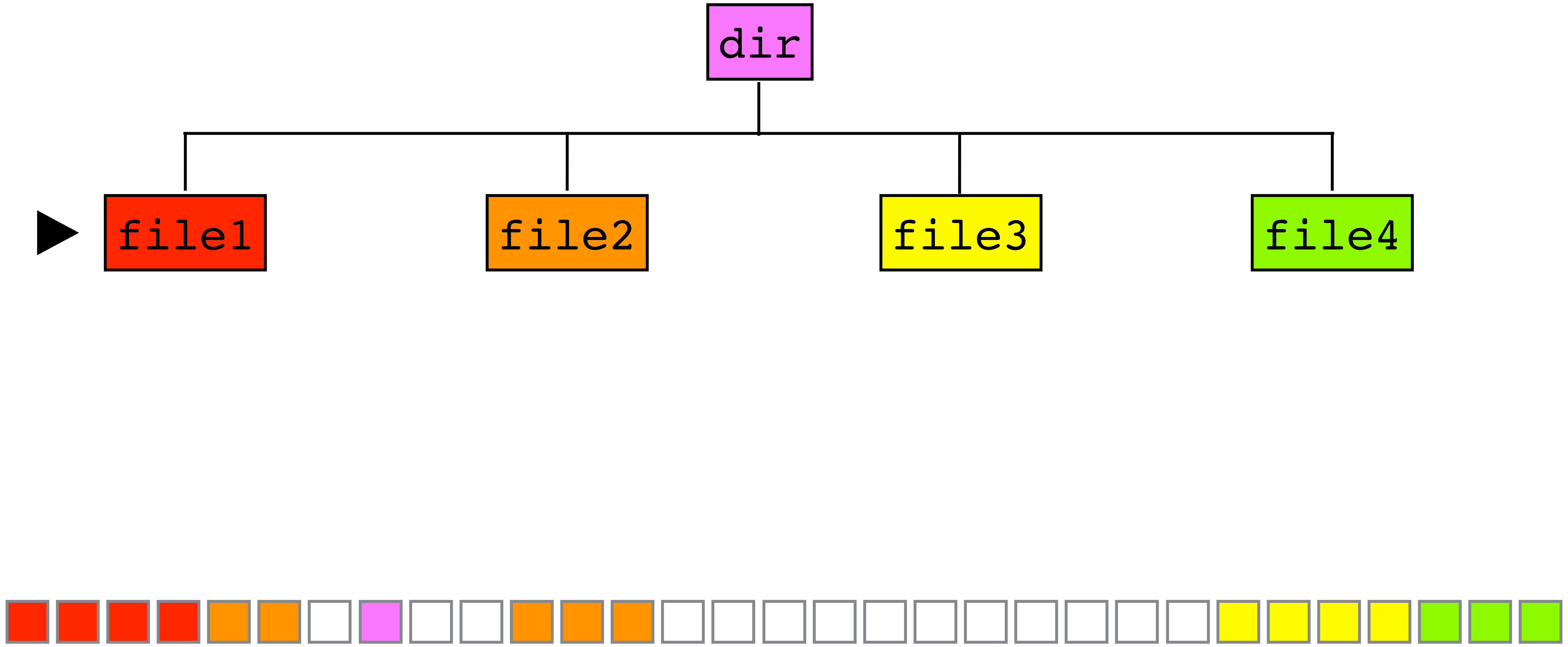


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

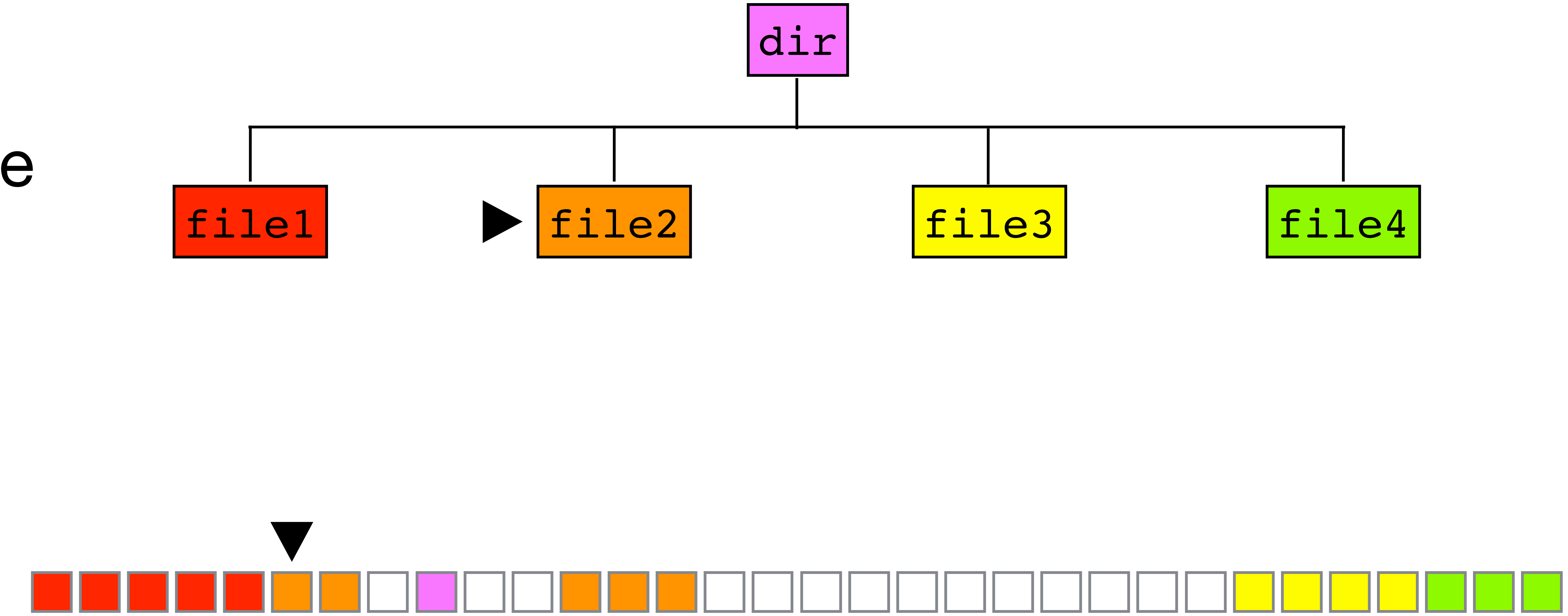


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

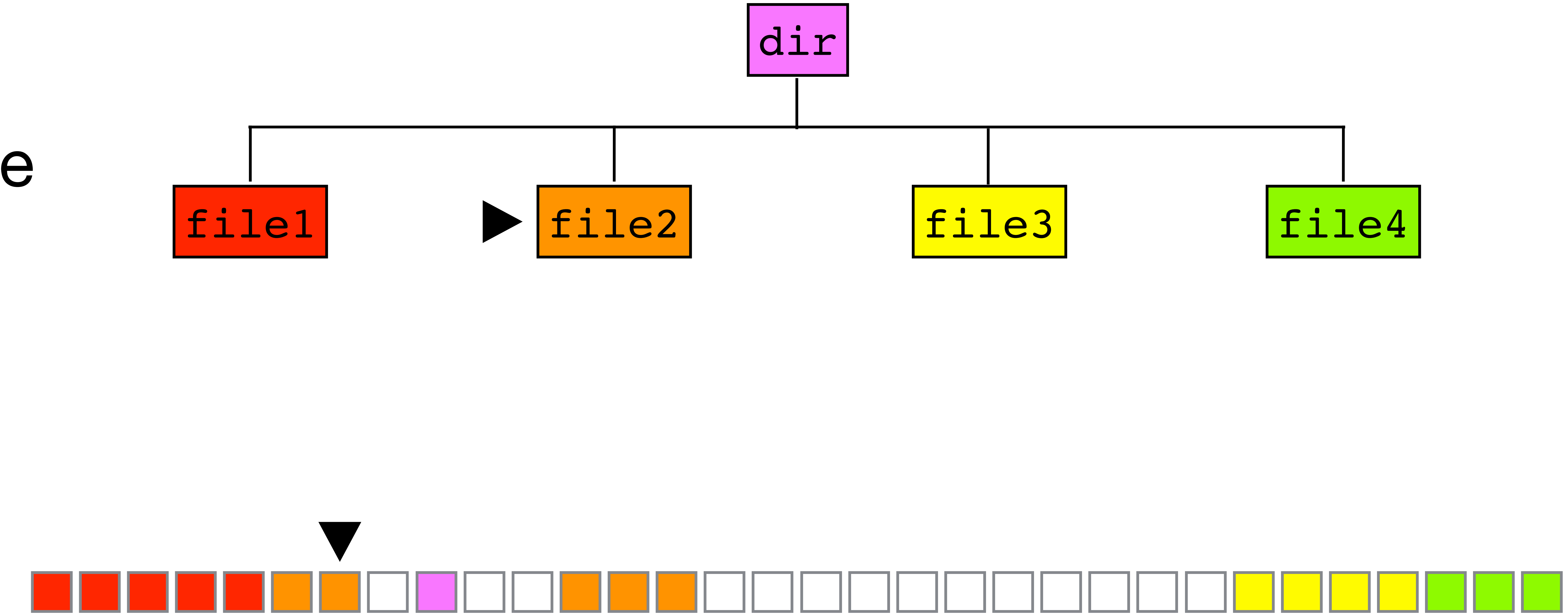


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

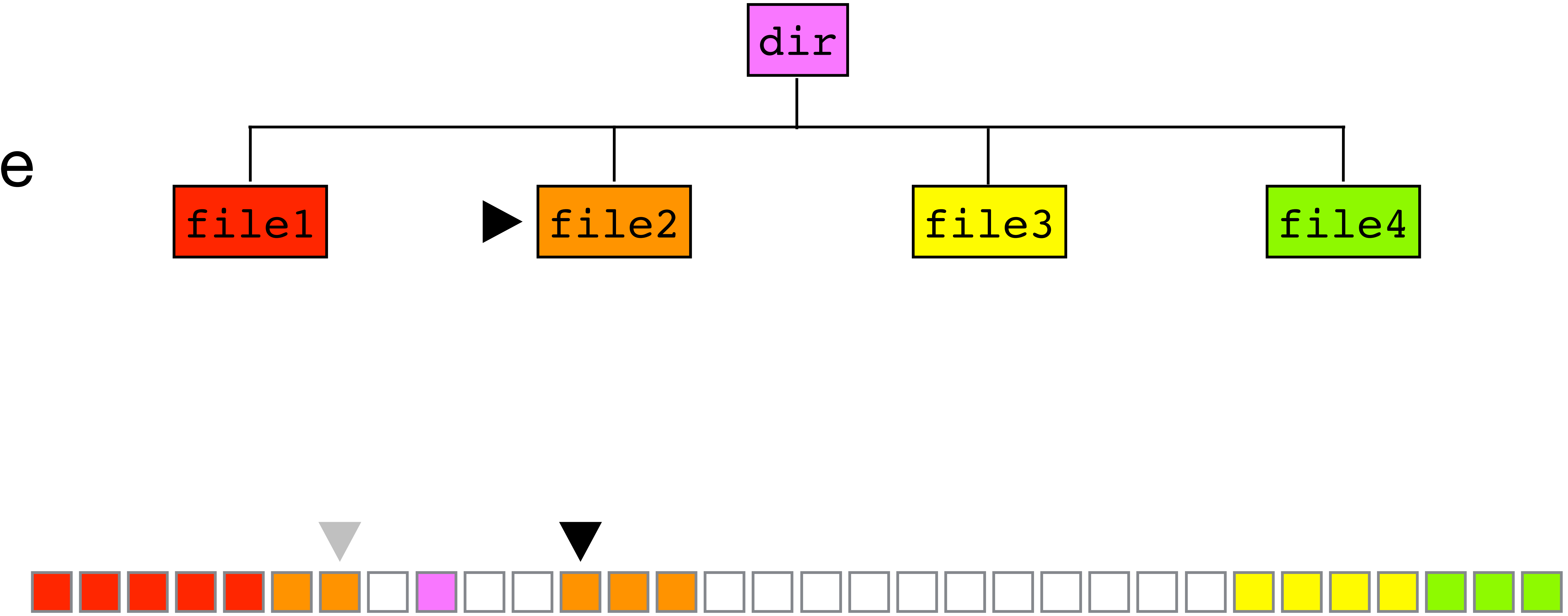


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

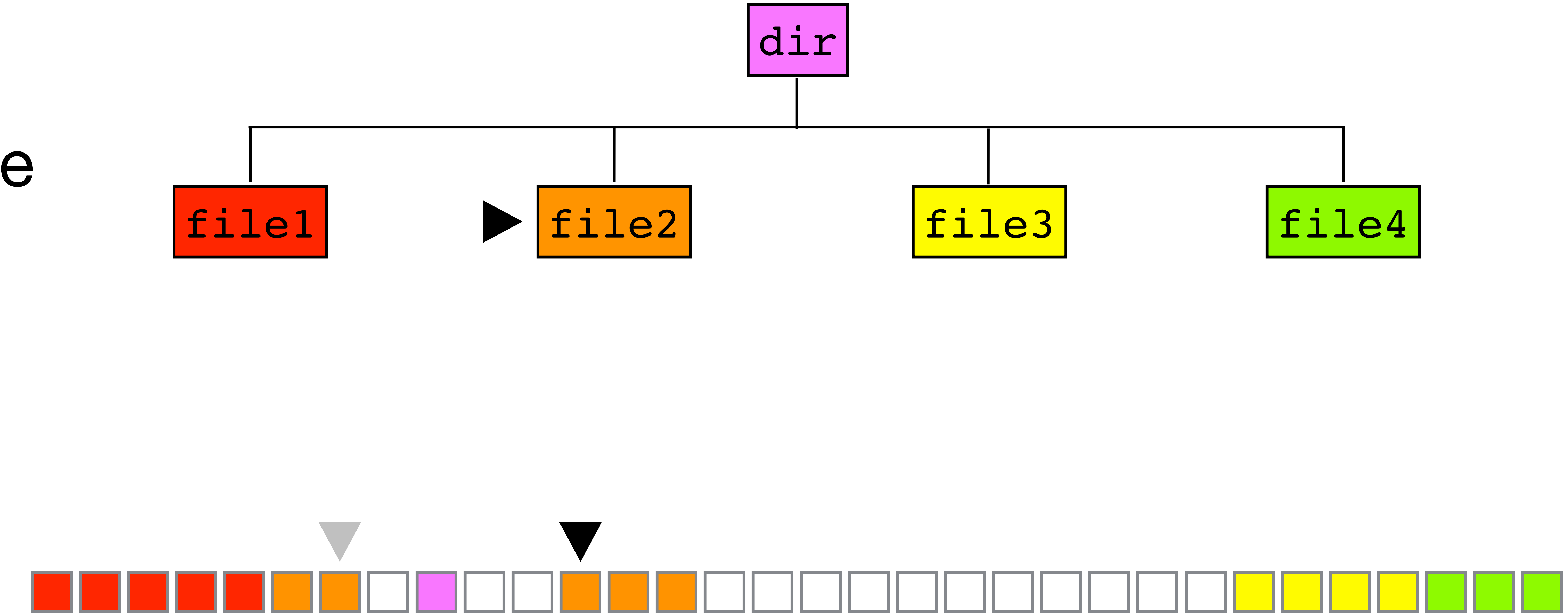


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

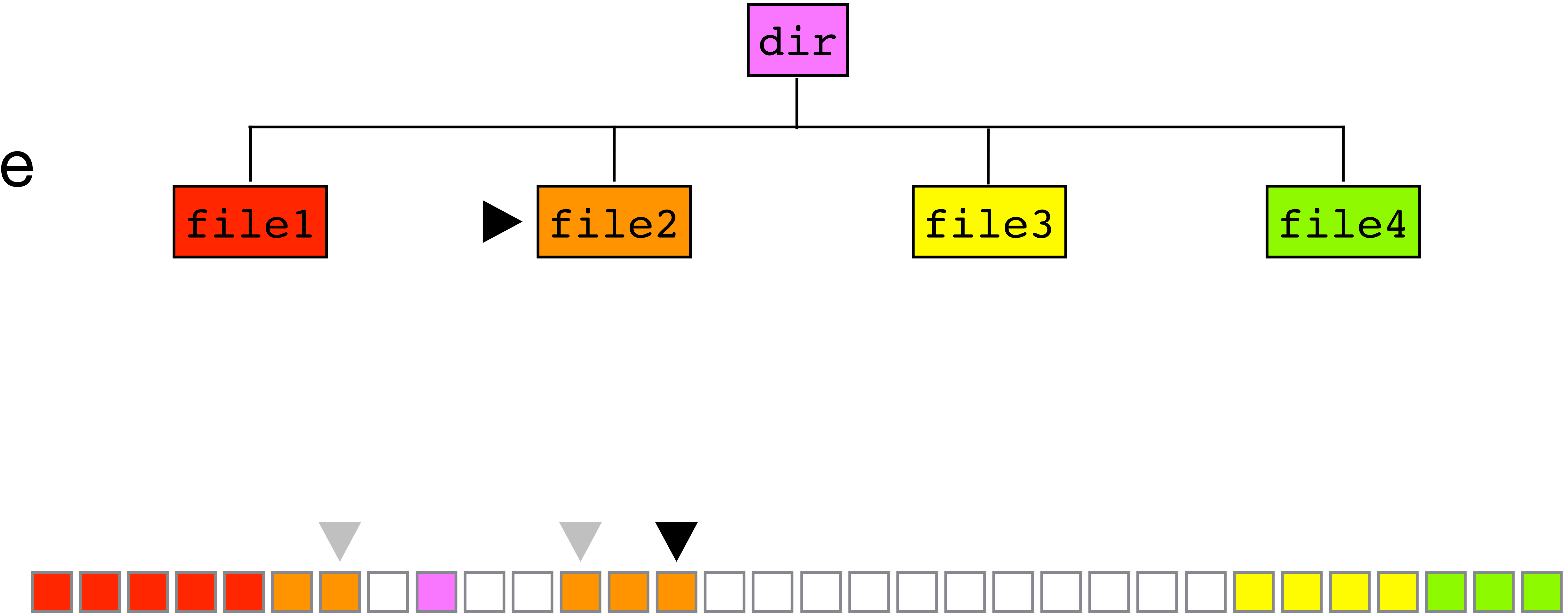


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

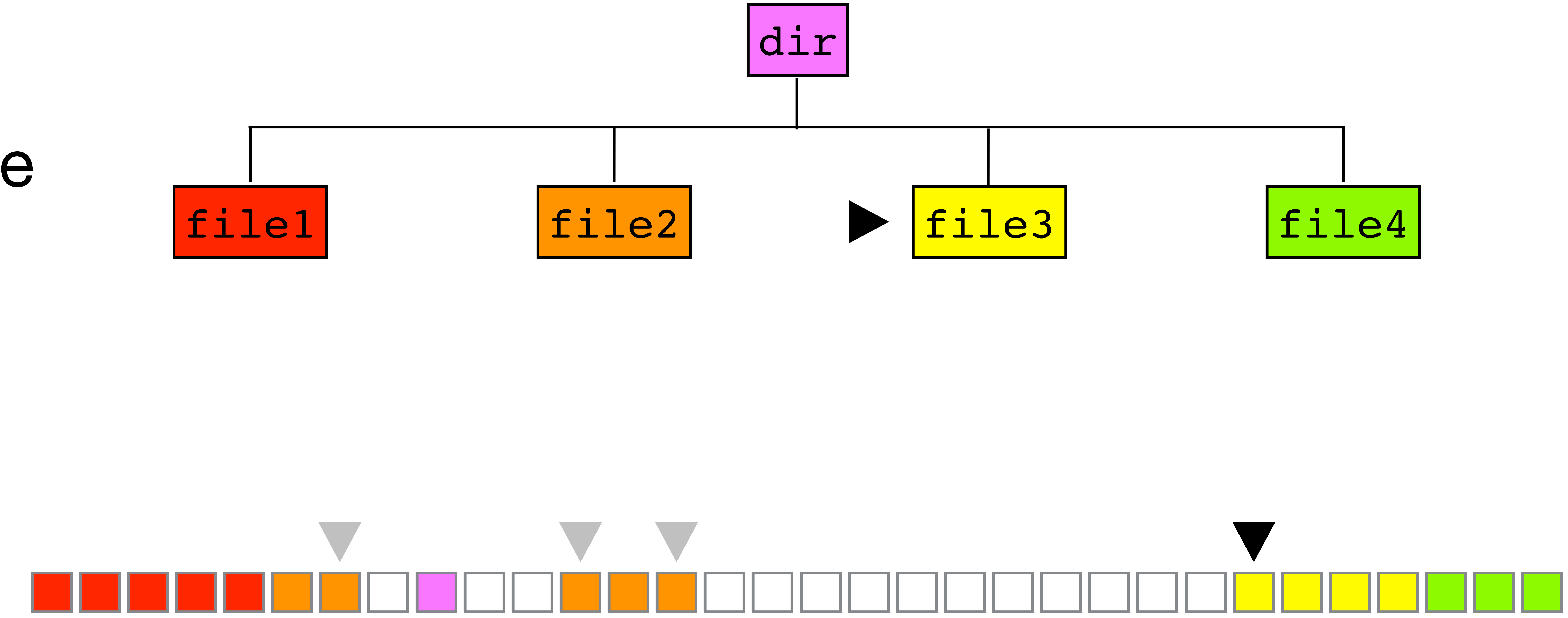


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

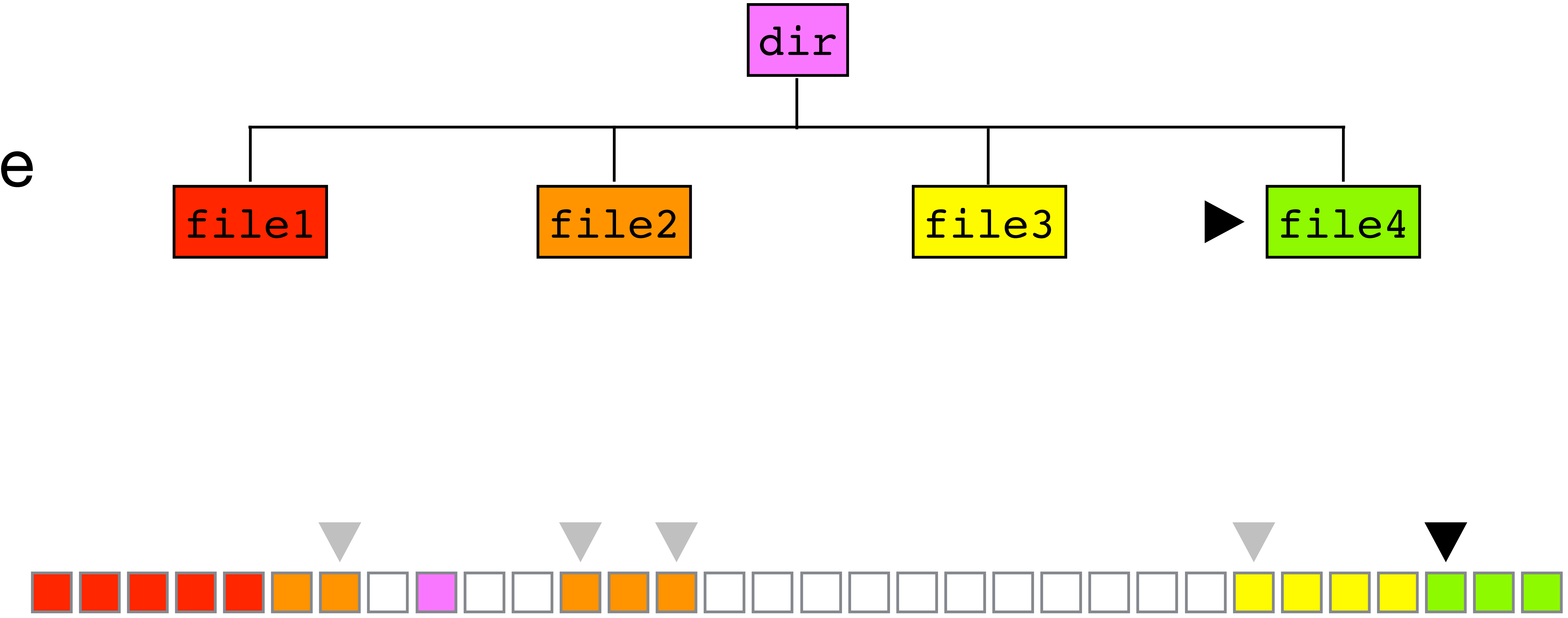


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

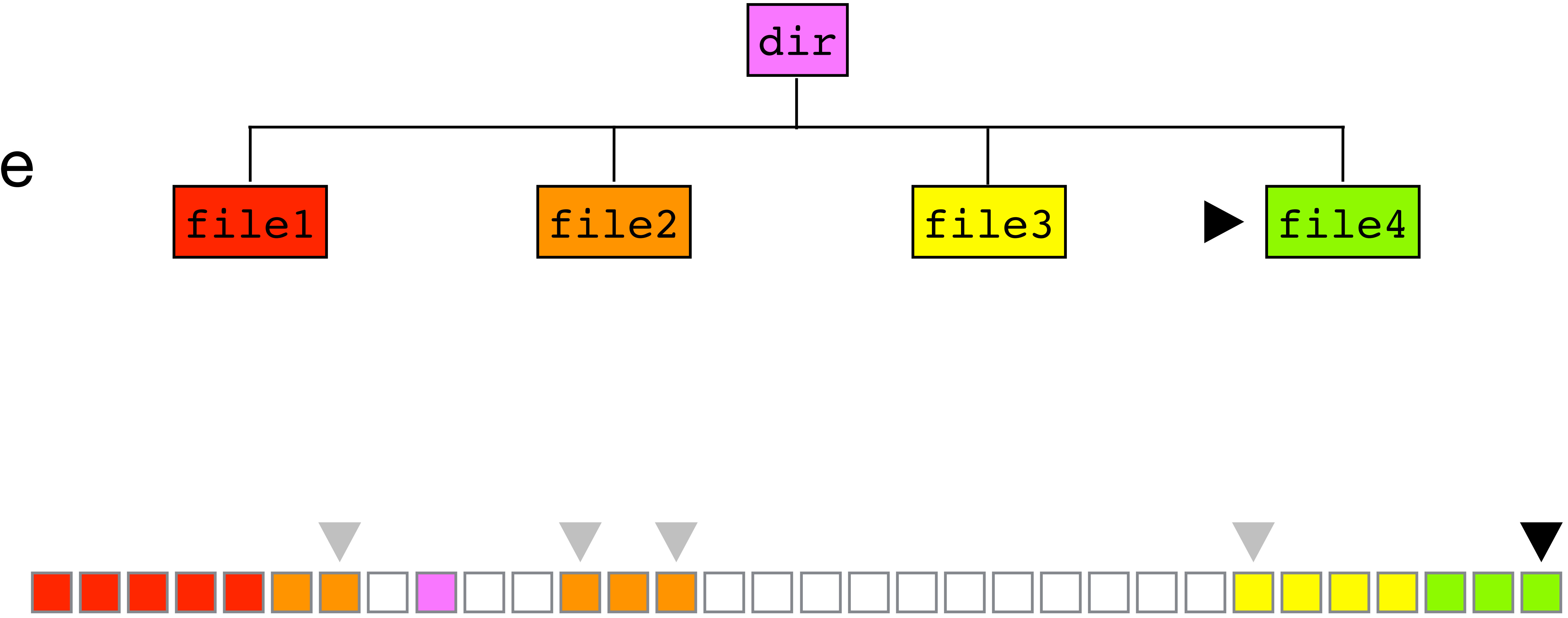


Measure read aging by reading through all the files in directory order

# Measuring Aging

```
time grep -r random_string /path/to/filesystem
```

How to measure read aging

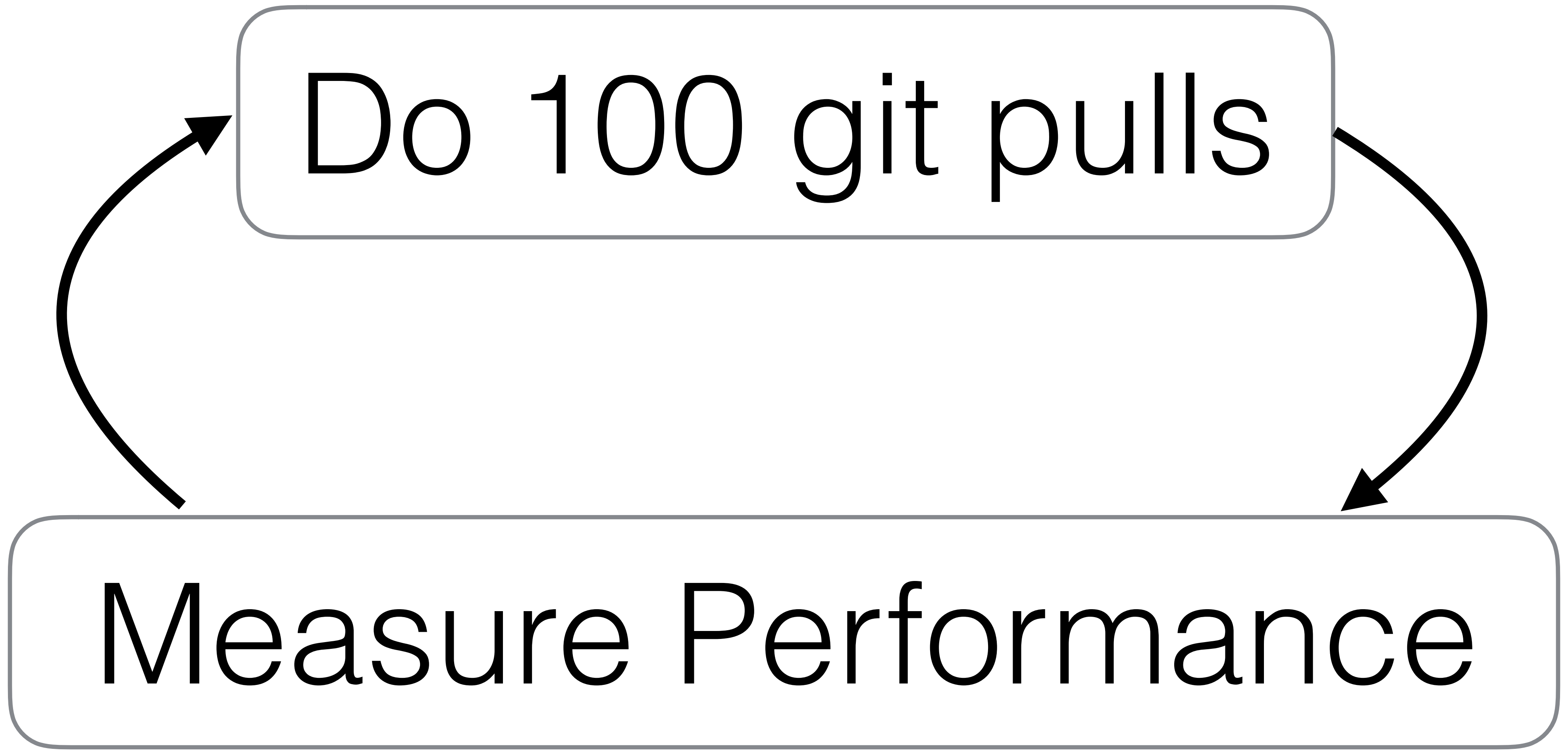


Normalize by filesystem size

Measure read aging by reading through all the files in directory order

# Git Replay Benchmark

Use the Linux kernel repo from github.com

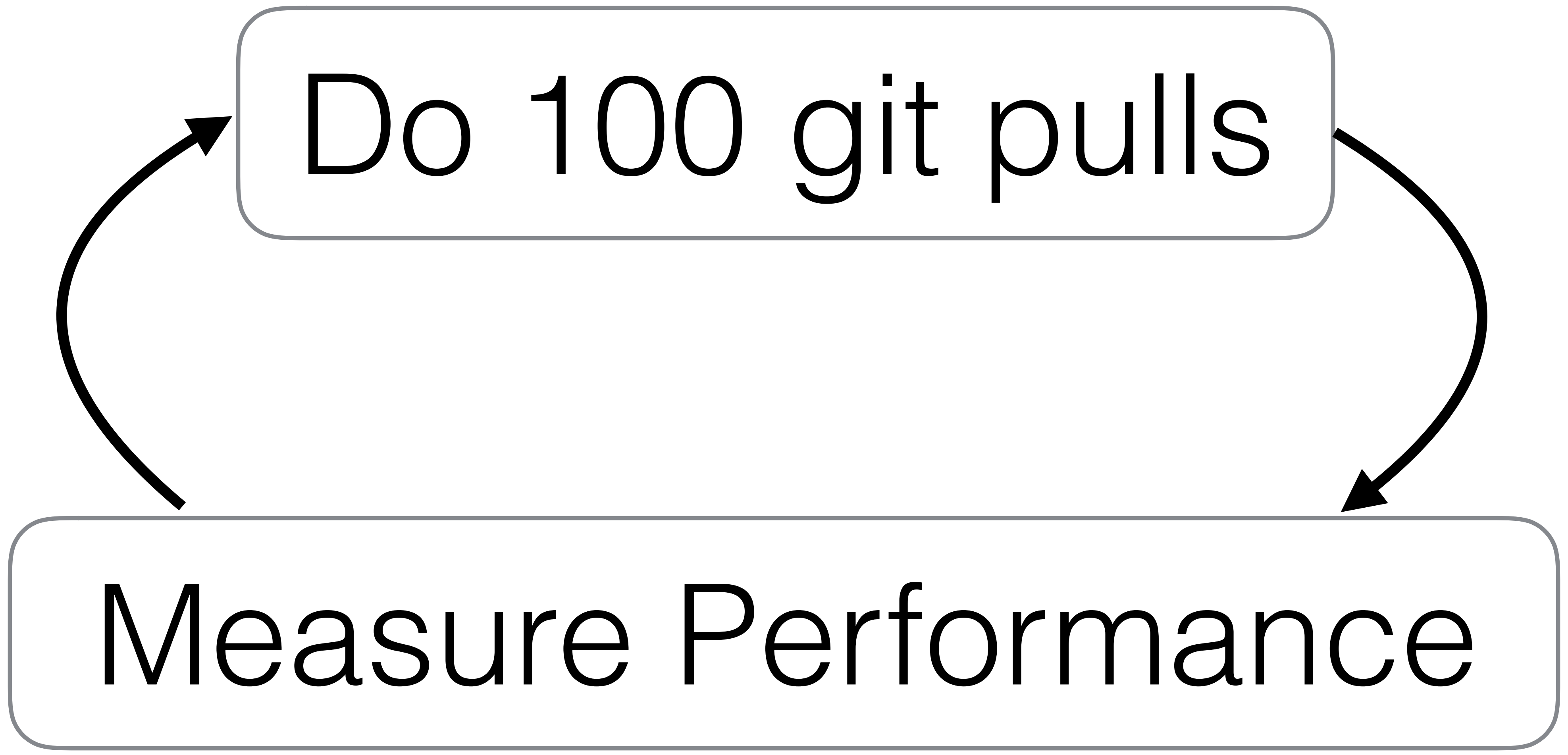


Do 100 git pulls

Measure Performance

# Git Replay Benchmark

Use the Linux kernel repo from github.com



Do 100 git pulls

Measure Performance

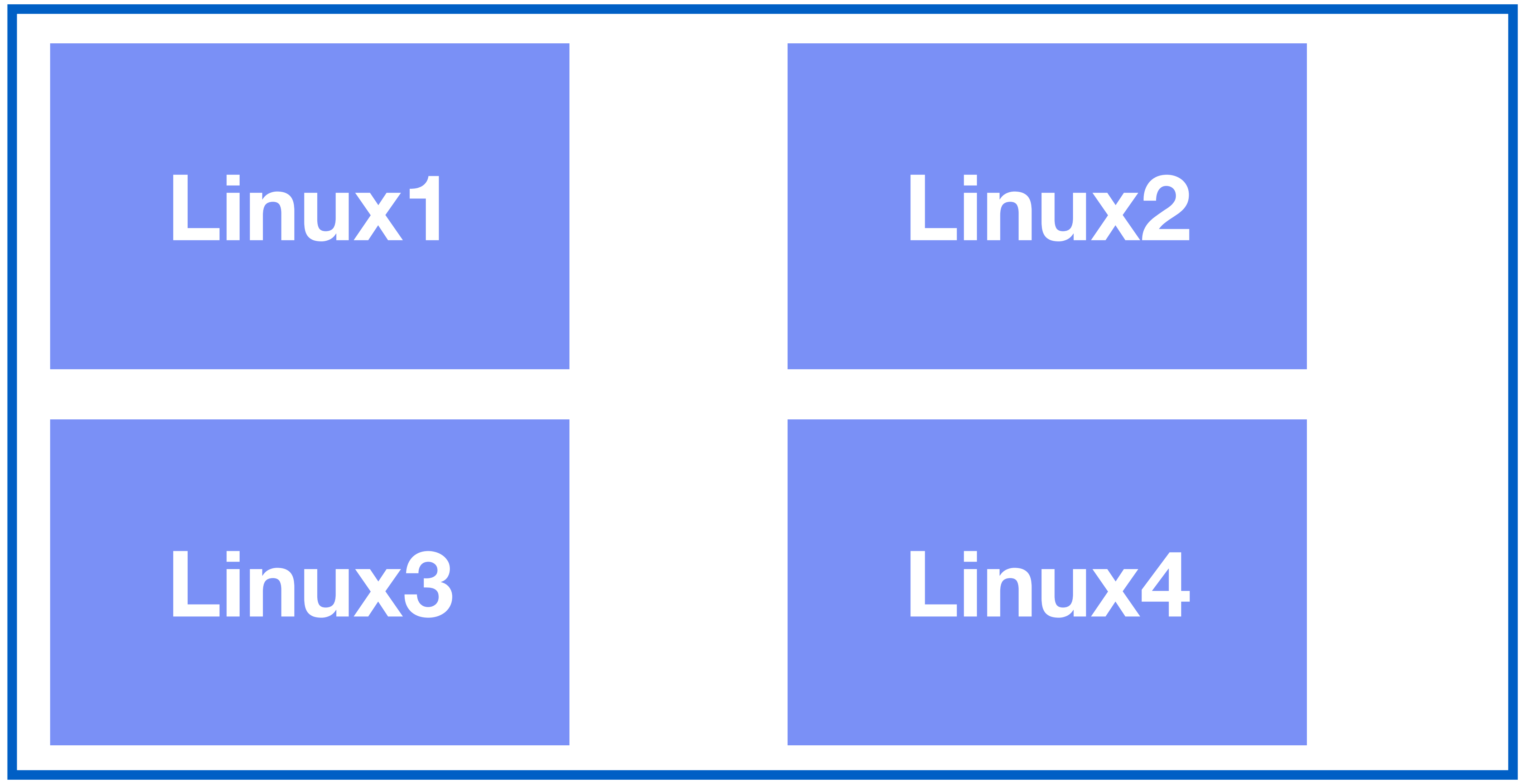
But what about when the disk fills up?

# Git Replay Benchmark

(full disk edition)

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo



Linux1

Linux2

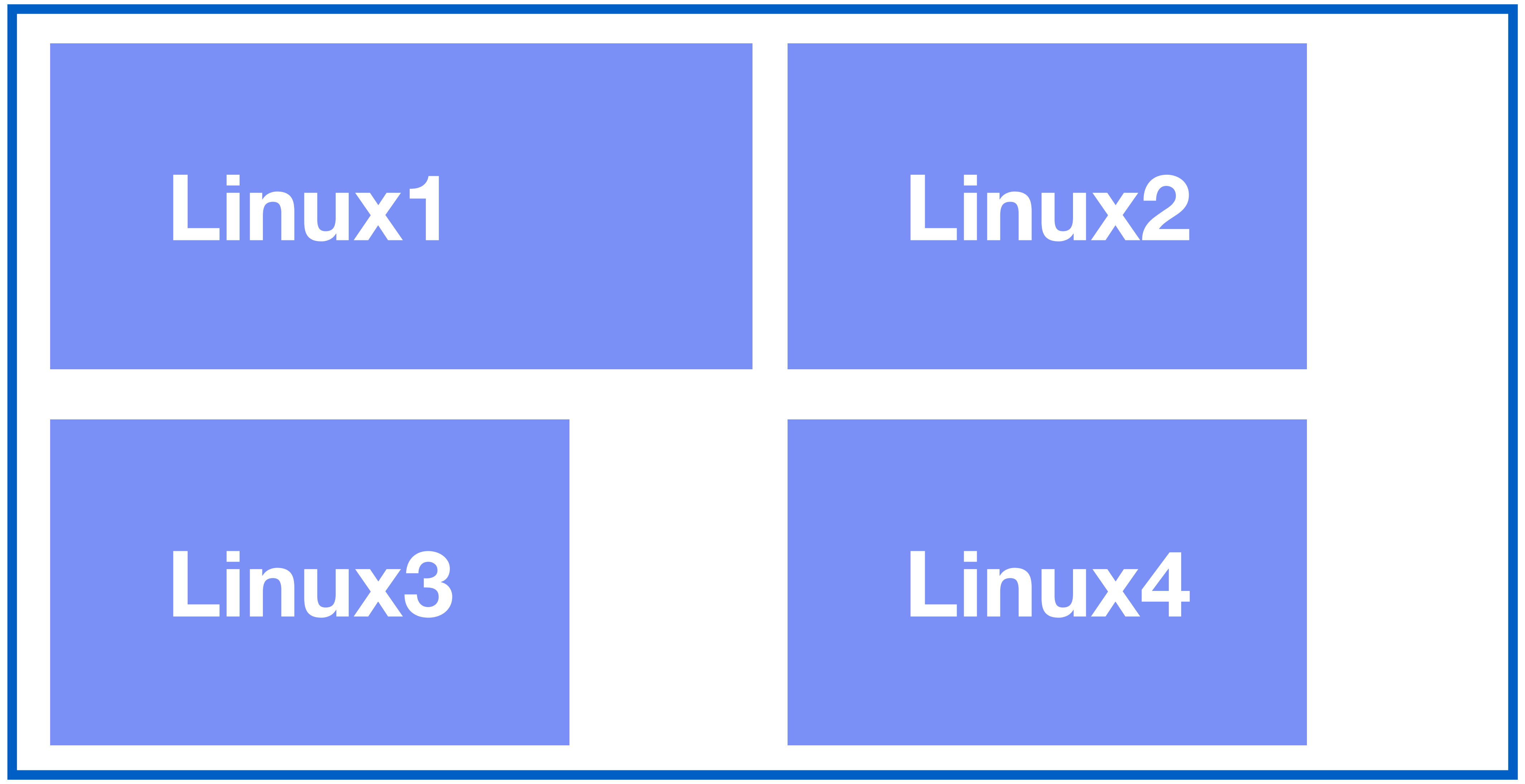
Linux3

Linux4

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo

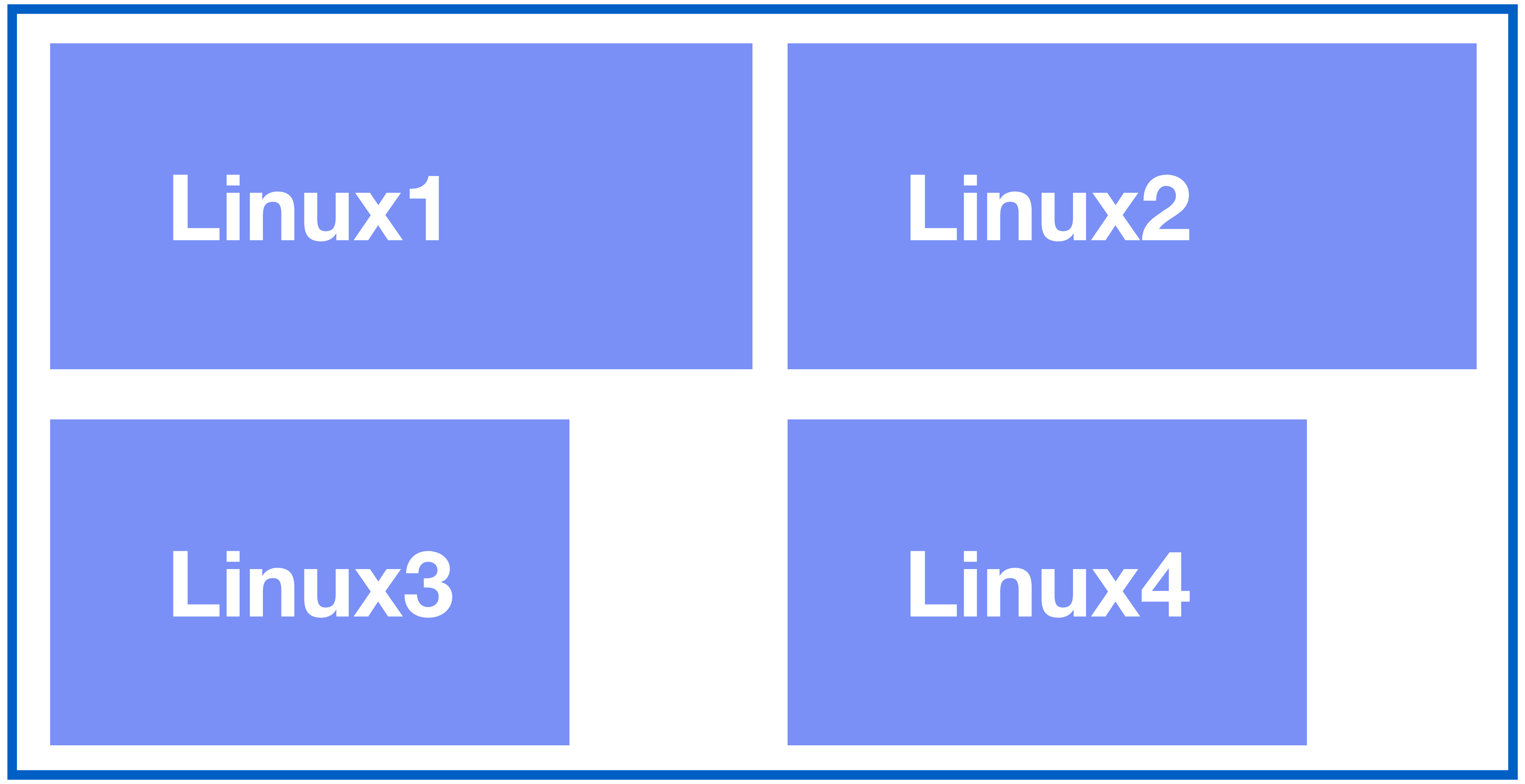


git pull

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo



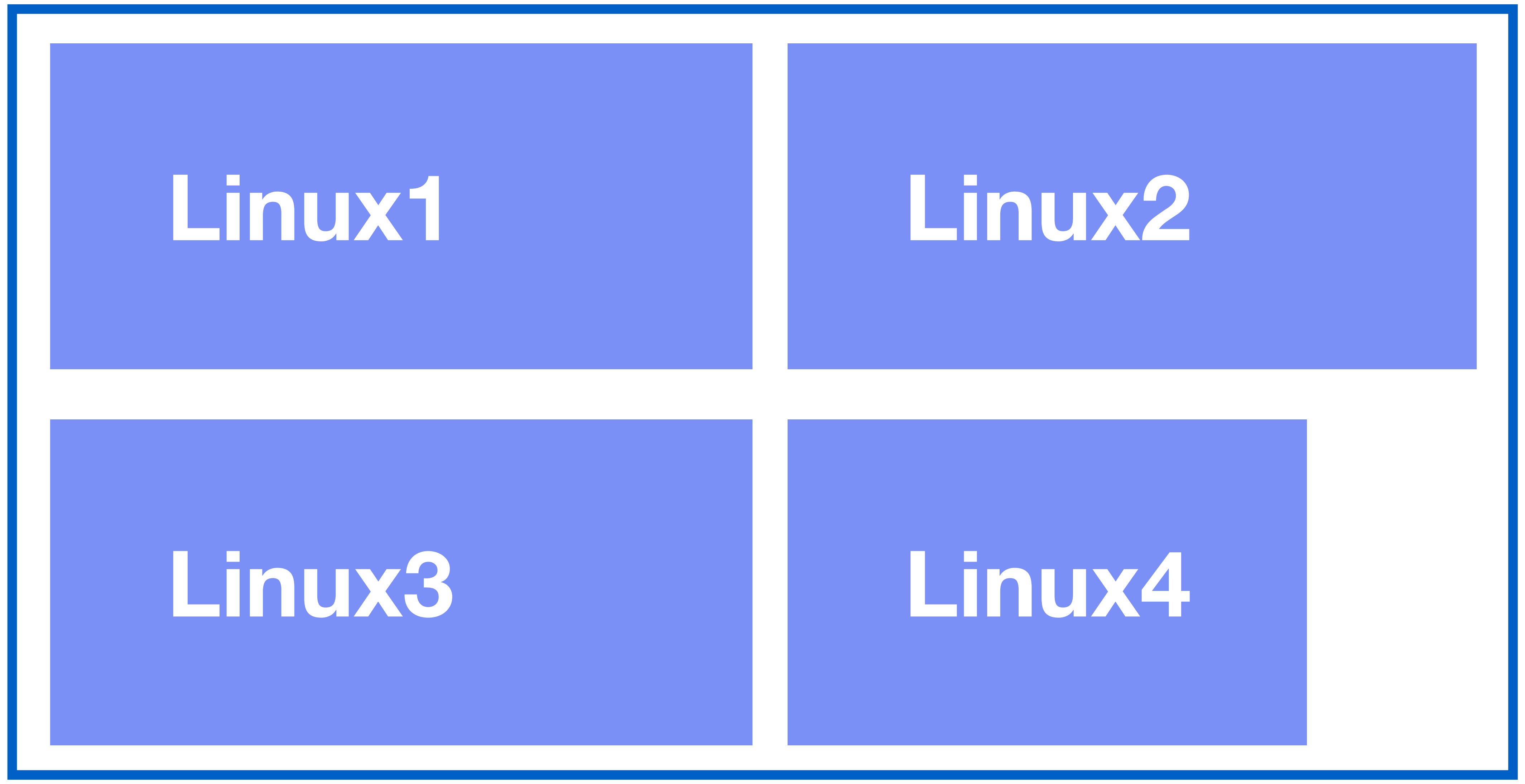
git pull

git pull

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo



Linux1

Linux2

Linux3

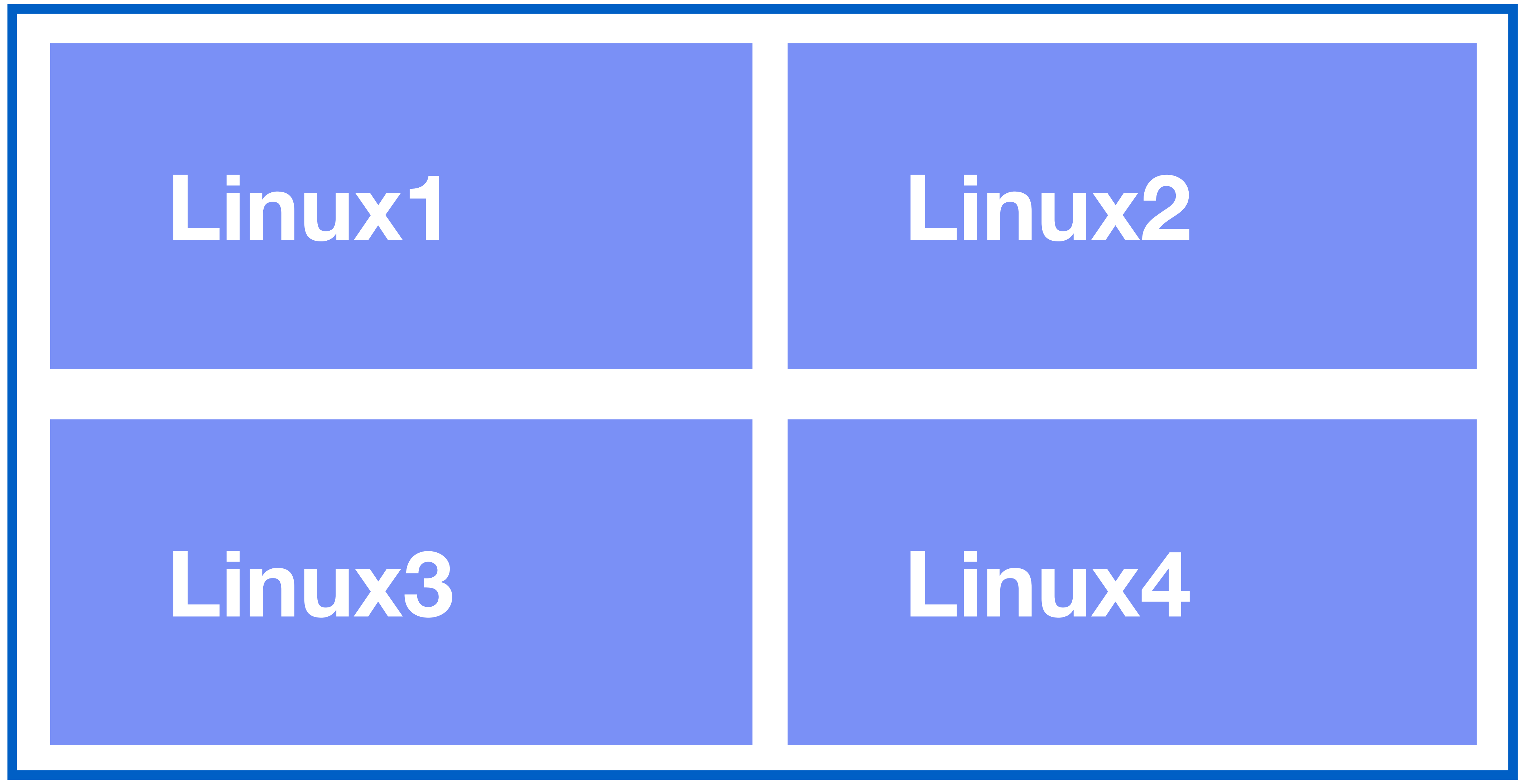
Linux4

```
git pull
git pull
git pull
```

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo

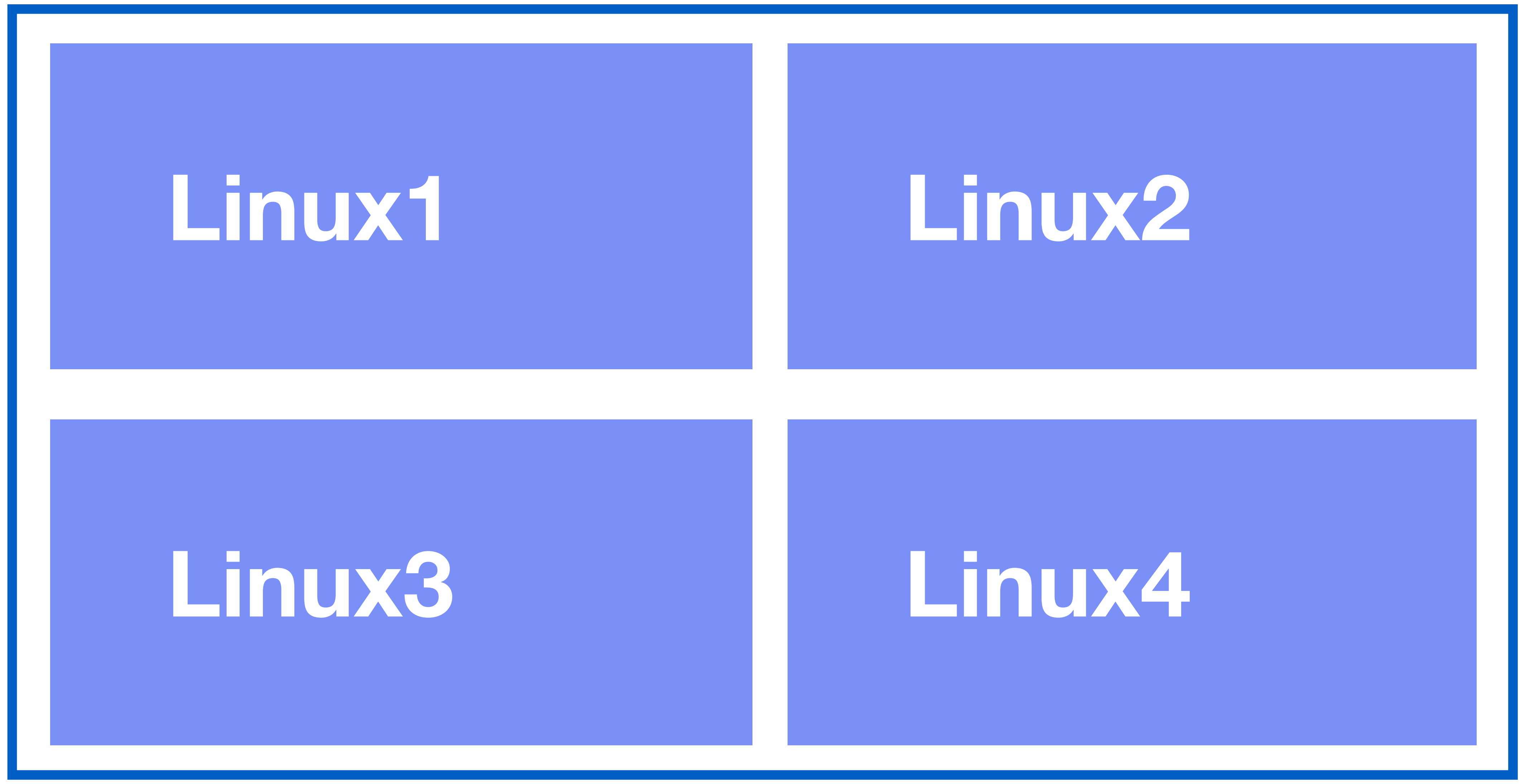


git pull
git pull
git pull
git pull

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo



Linux1

Linux2

Linux3

Linux4

git pull

git pull

git pull

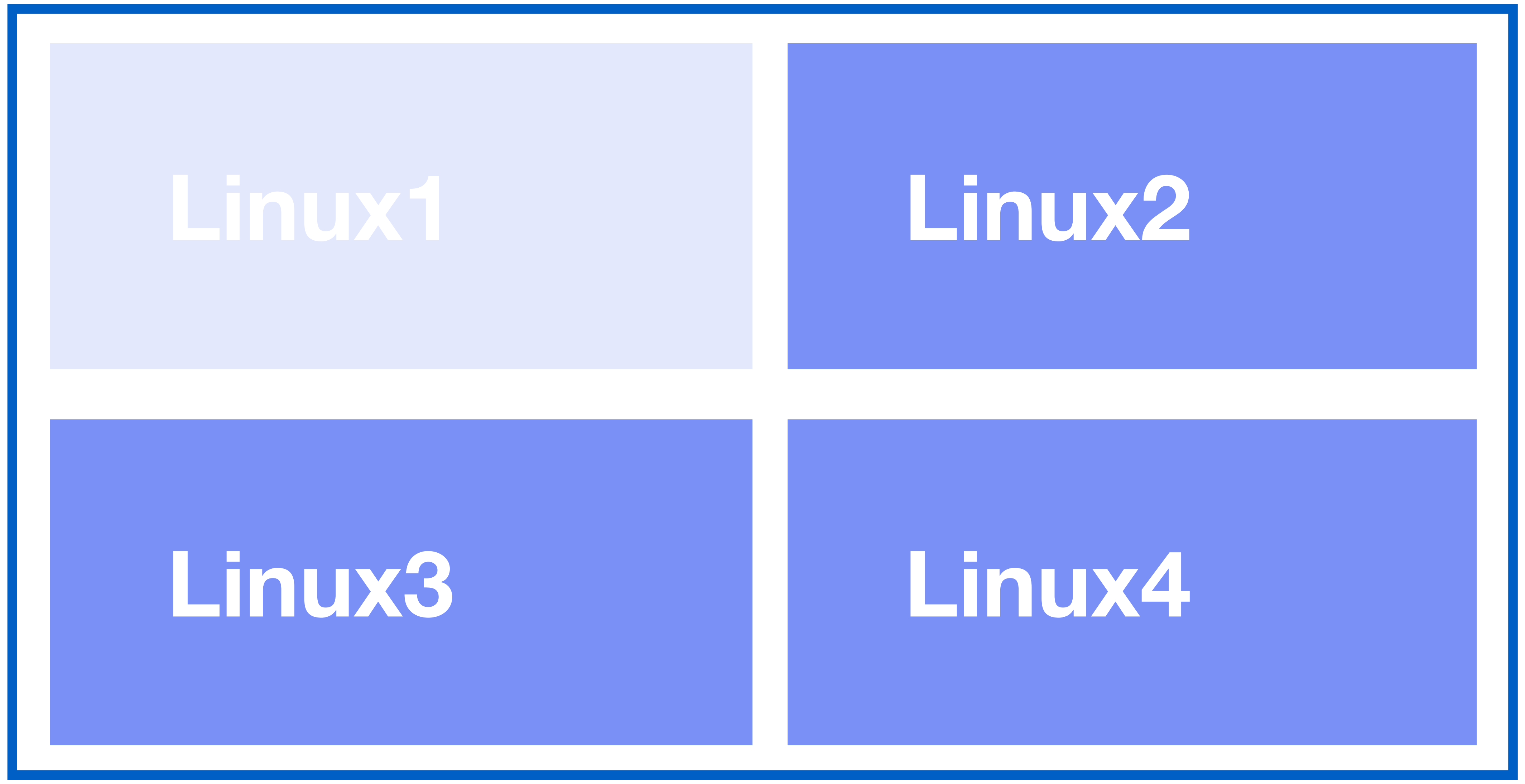
git pull

git pull **FAIL**

But what about when the disk fills up?

# Git Replay Benchmark (full disk edition)

Use multiple copies of the repo



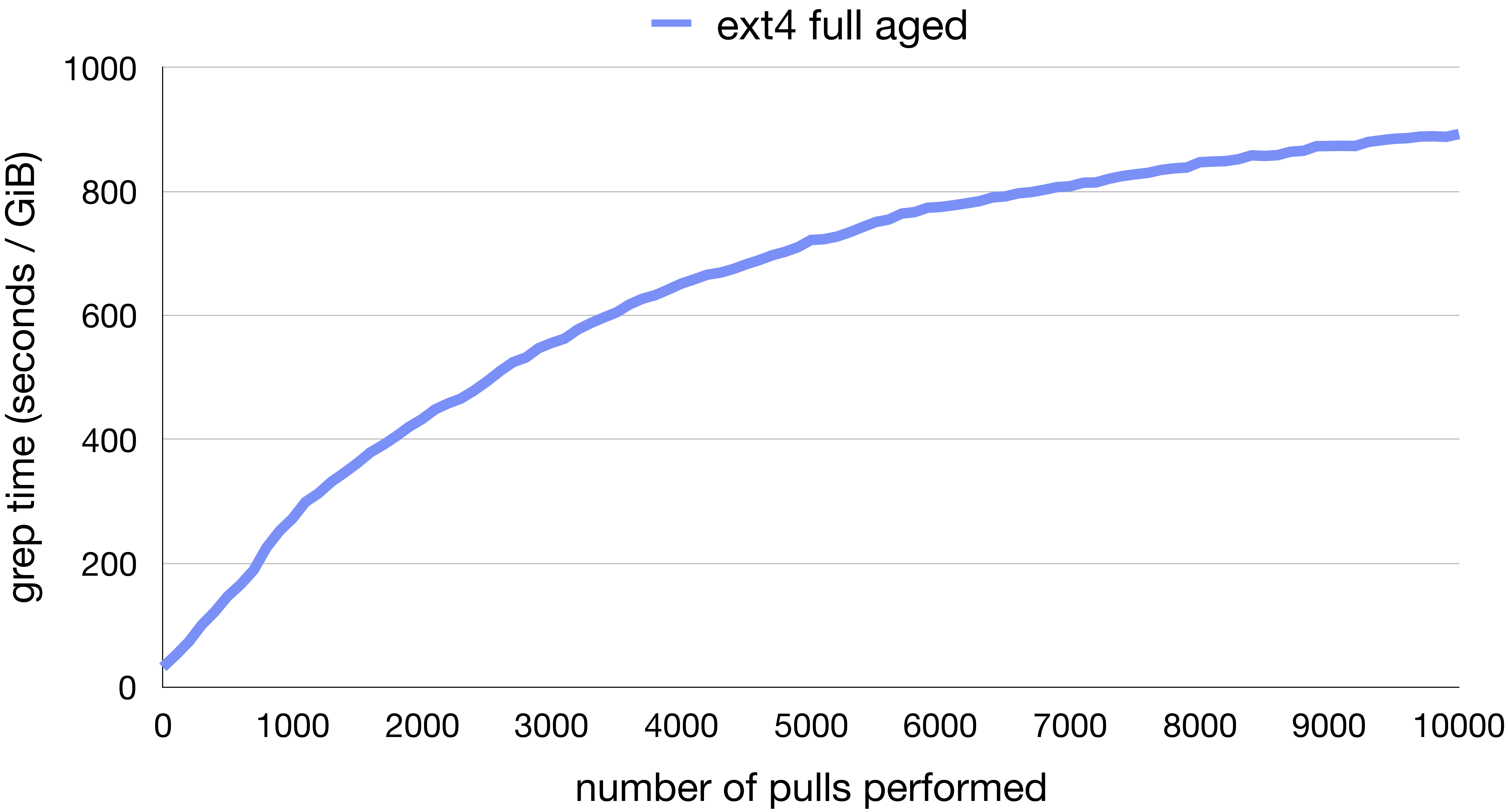
git pull
git pull
git pull
git pull
git pull **FAIL**

When the disk fills and a pull fails, delete one

But what about when the disk fills up?

# Full Disk Git Aging on HDD

# Full Disk Git Aging on HDD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

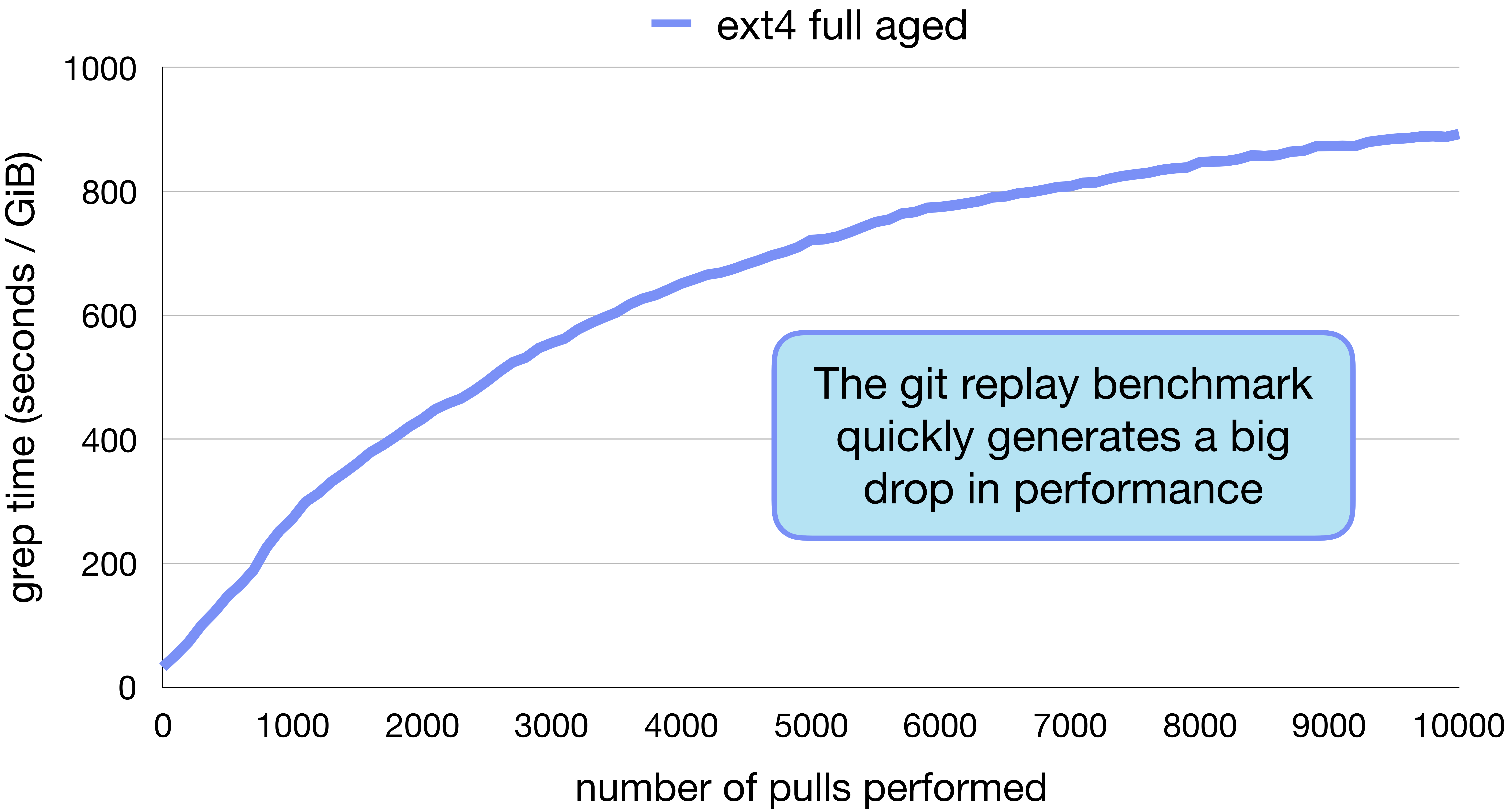
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



The git replay benchmark quickly generates a big drop in performance

| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

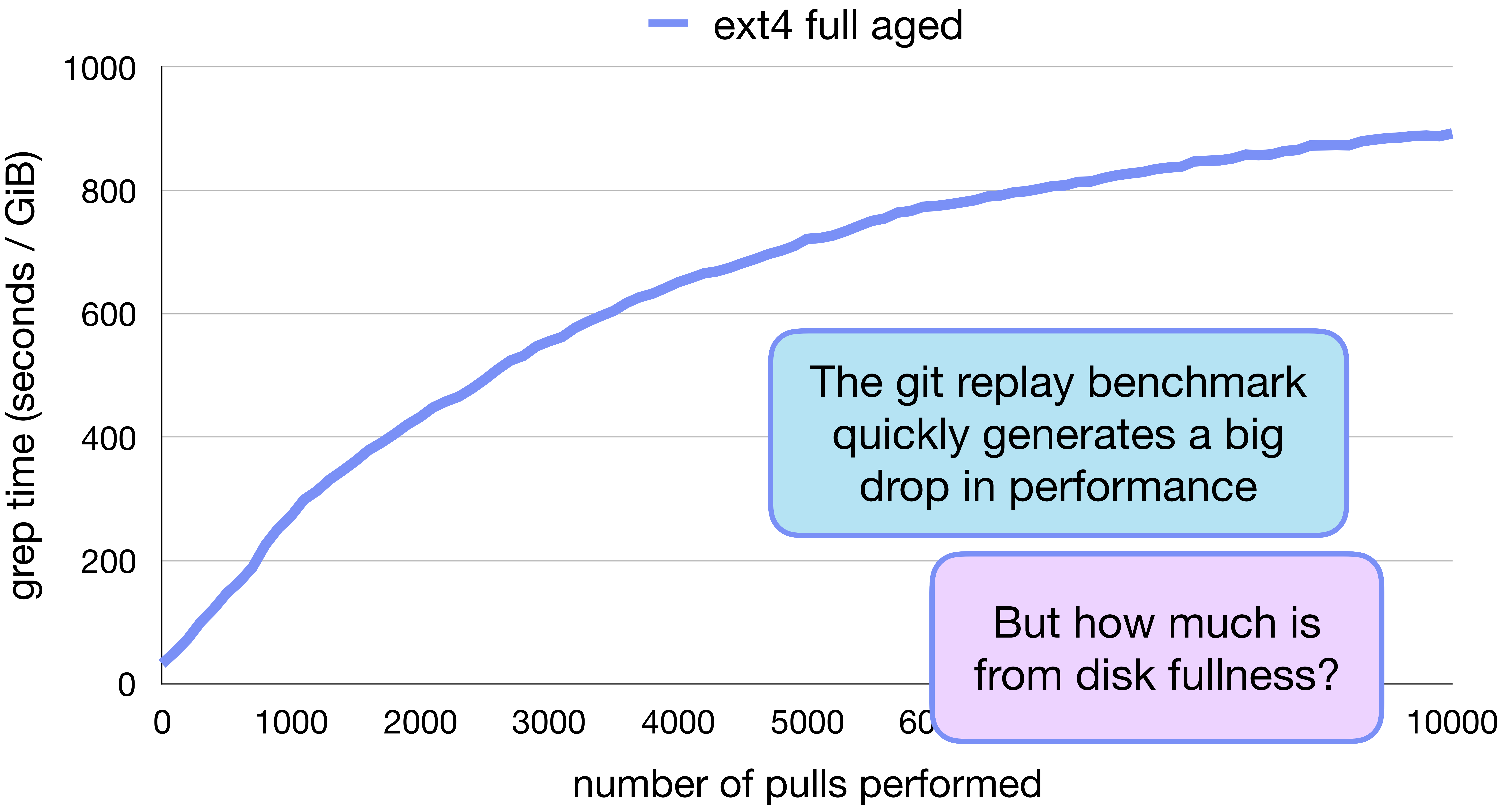
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



The git replay benchmark quickly generates a big drop in performance

But how much is from disk fullness?

| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

4-core 3.00 GHz Intel® Xeon(R) E3-1220 v6 CPU

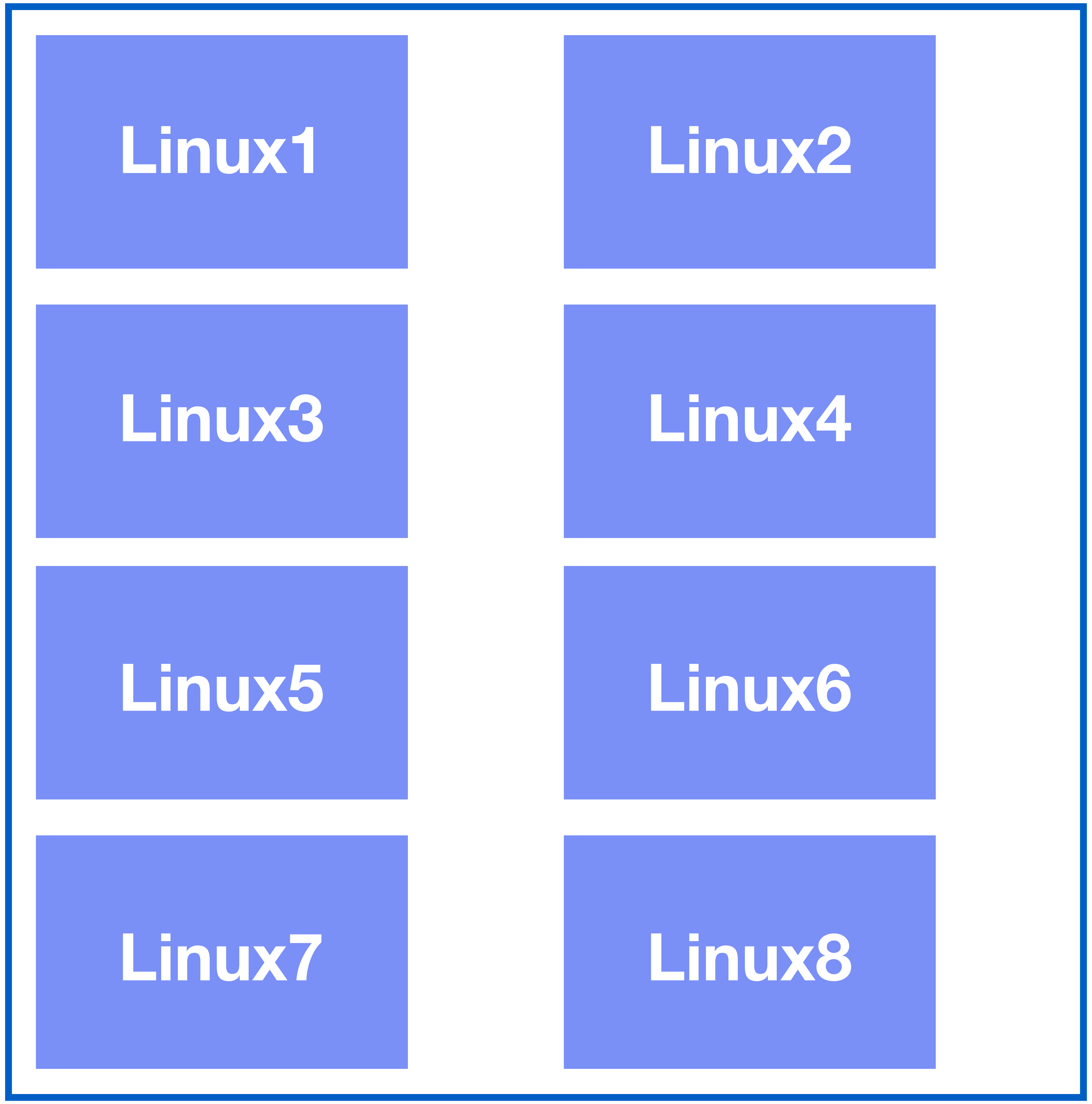
16 GiB RAM

500GiB 7200 RPM Toshiba HDD

Ubuntu version 18.04 LTS Linux kernel v 4.15

Need to compare
to non-full disks

# How to compare to non-full disks



Linux1 Linux2

Linux3 Linux4

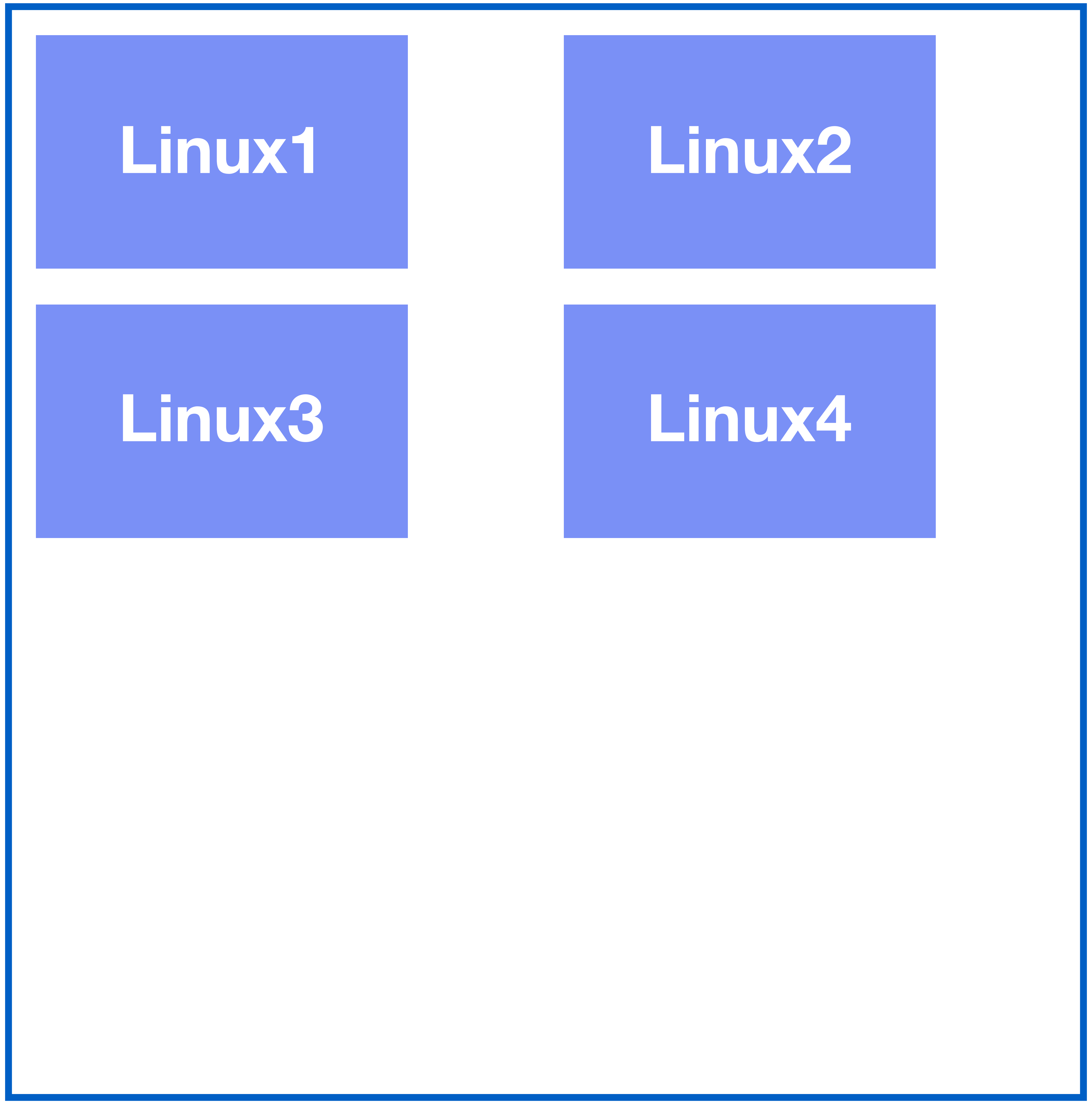
Linux5 Linux6

Linux7 Linux8

We can’t just use a larger disk with the git replay benchmark

It’ll still just be full!

# How to compare to non-full disks

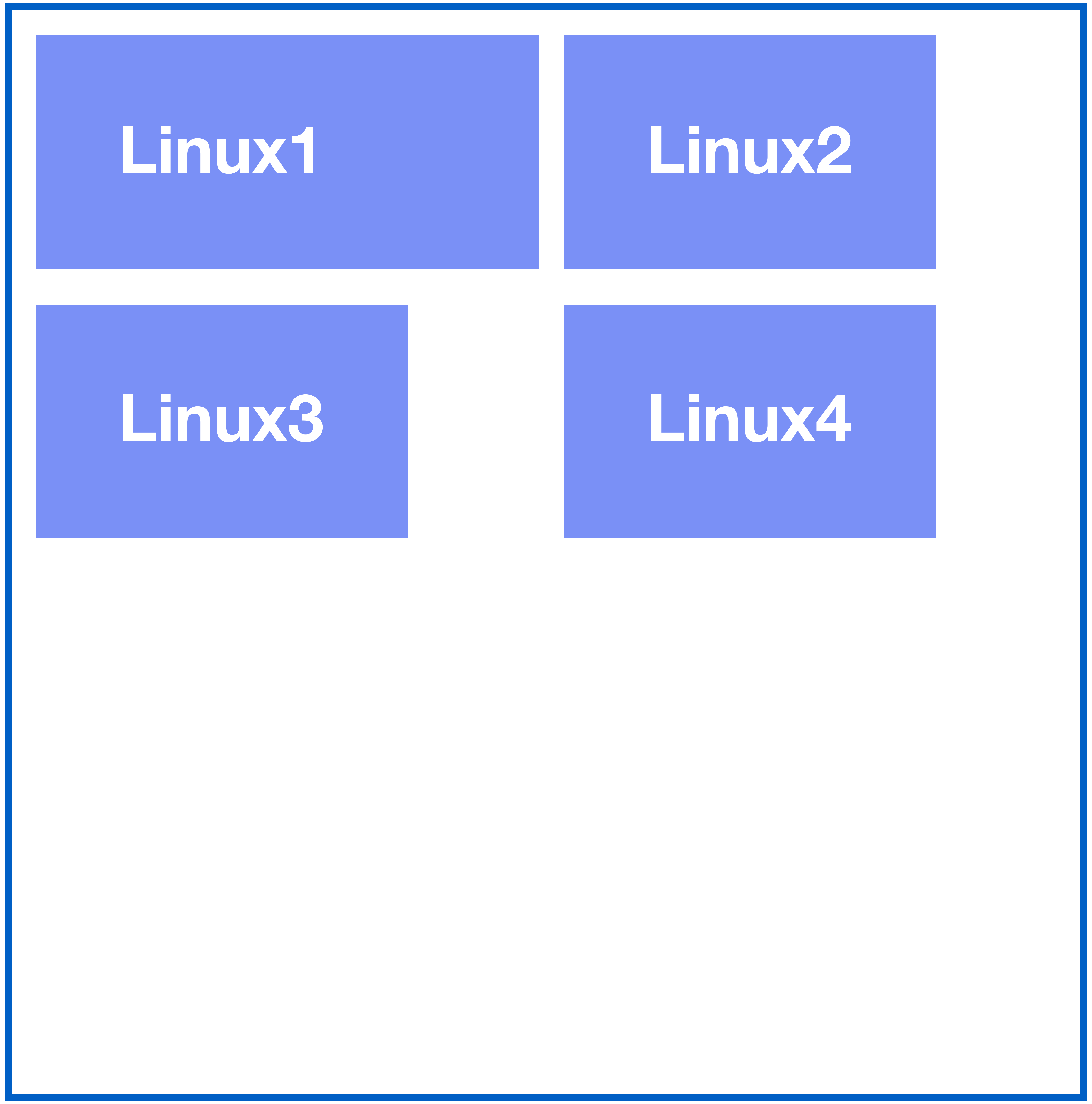


We can't just use a larger disk with the git replay benchmark

It'll still just be full!

We record the operations on the small disk and replay them on the larger disk

# How to compare to non-full disks

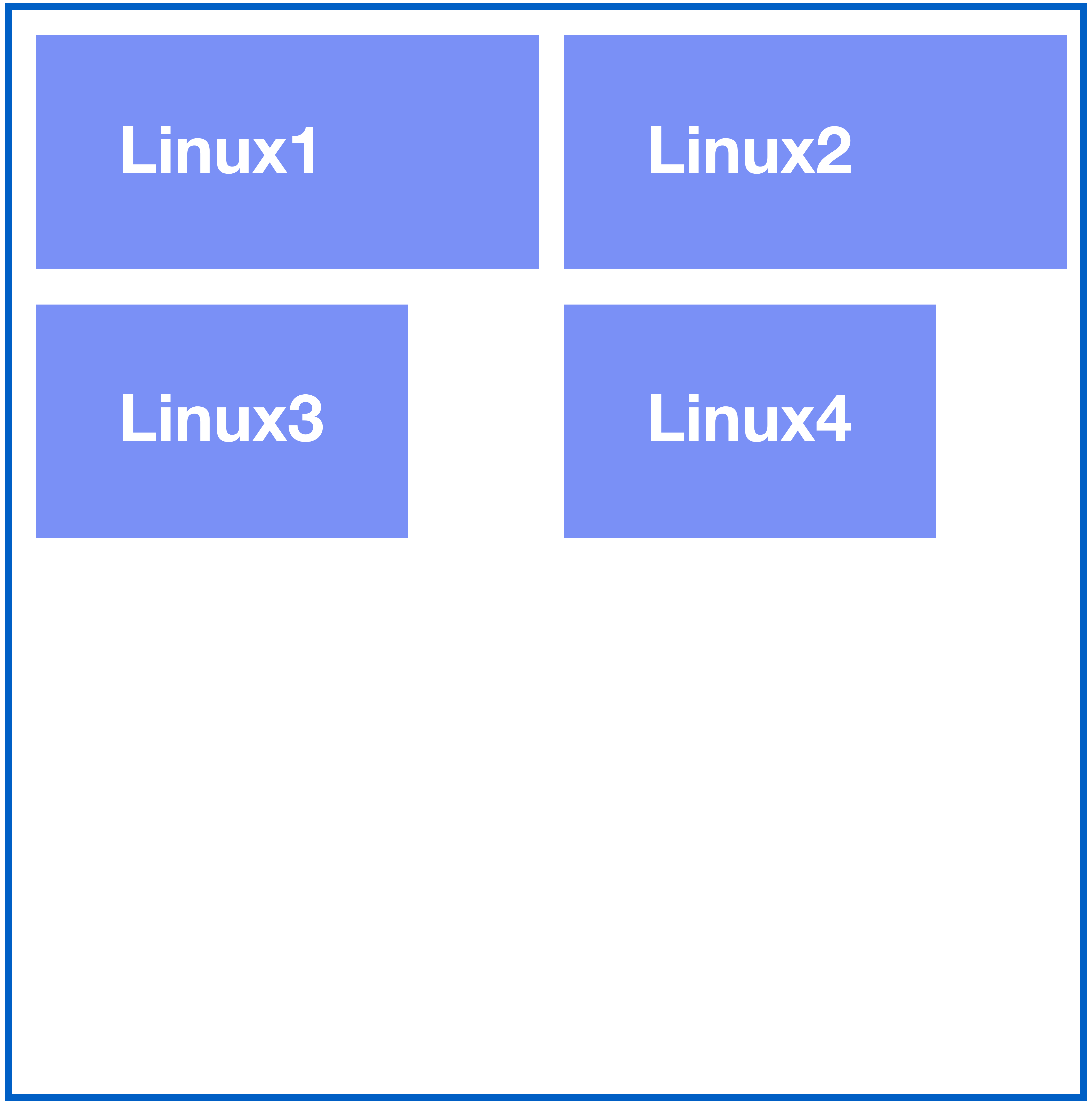


We can’t just use a larger disk with the git replay benchmark

It’ll still just be full!

We record the operations on the small disk and replay them on the larger disk

# How to compare to non-full disks

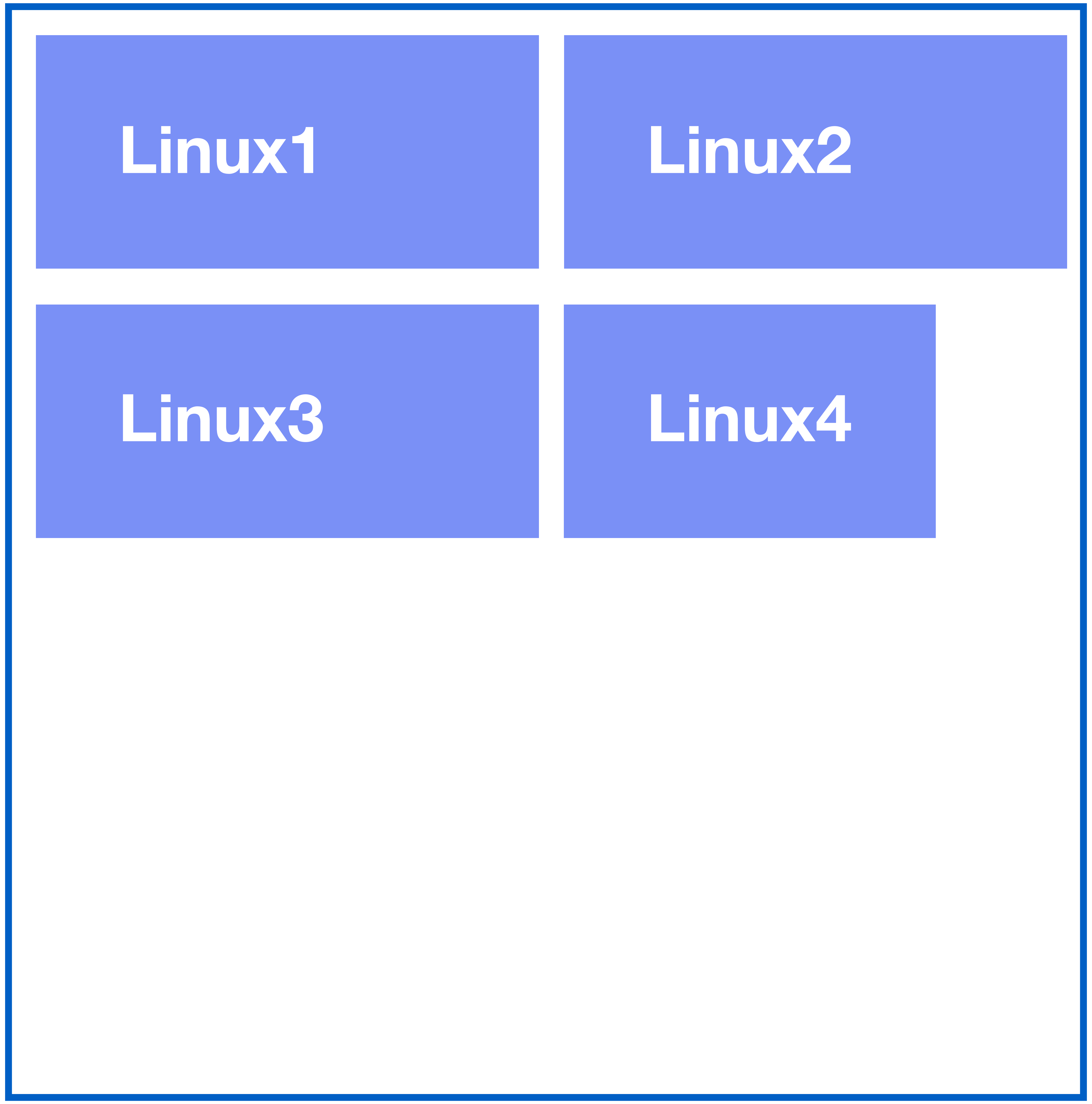


We can't just use a larger disk with the git replay benchmark

It'll still just be full!

We record the operations on the small disk and replay them on the larger disk

# How to compare to non-full disks

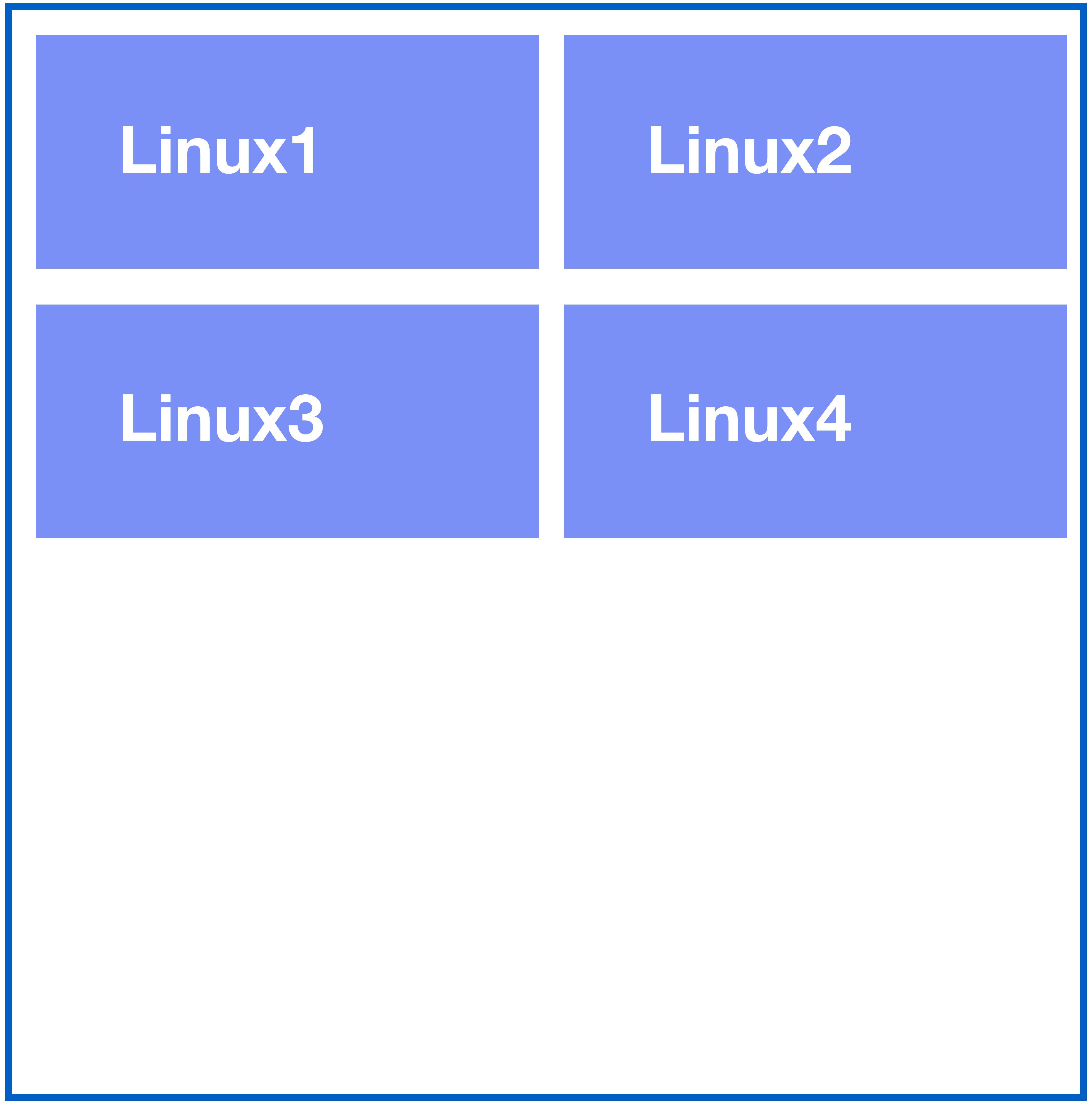


We can't just use a larger disk with the git replay benchmark

It'll still just be full!

We record the operations on the small disk and replay them on the larger disk

# How to compare to non-full disks

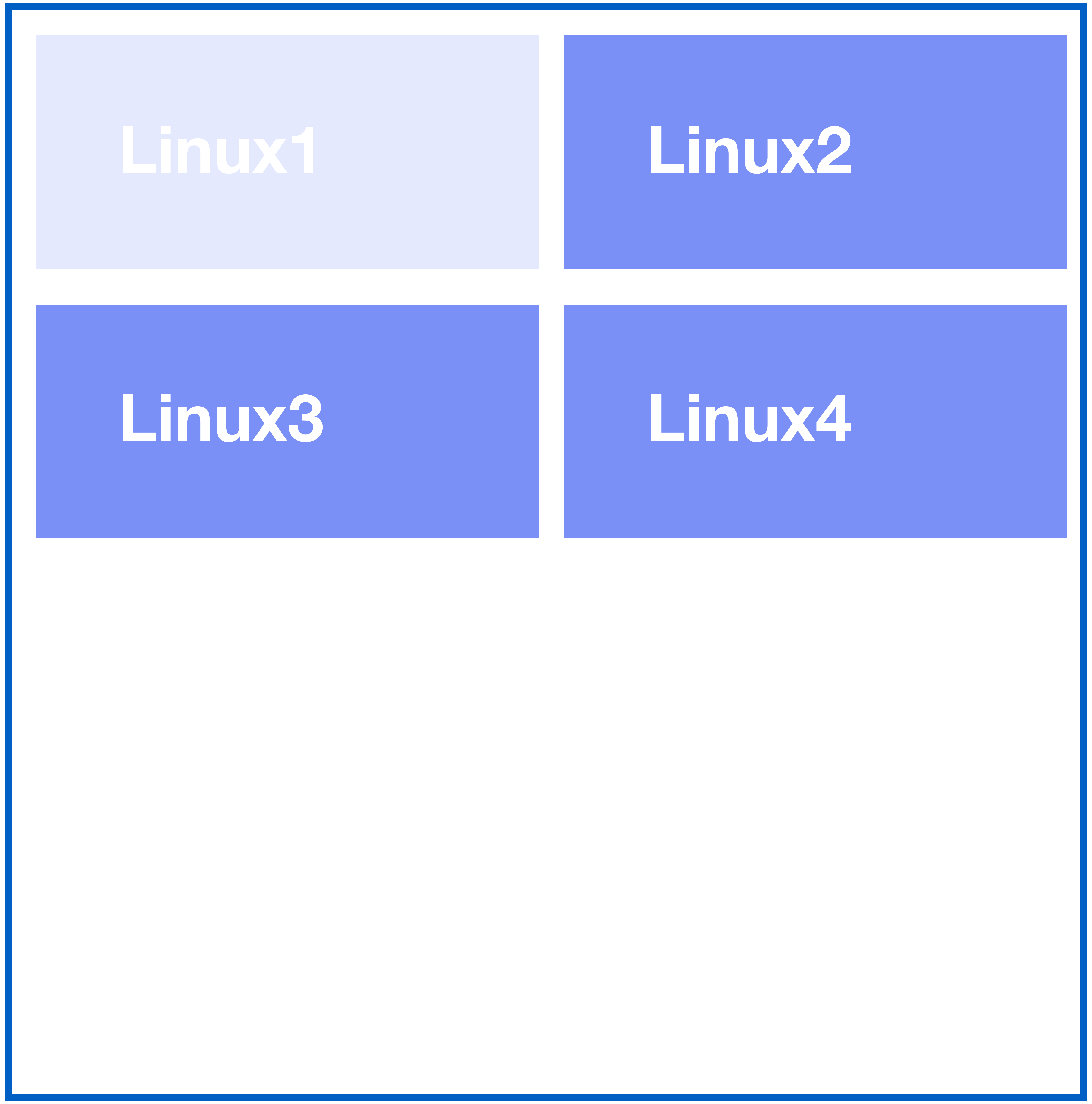


We can’t just use a larger disk with the git replay benchmark

It’ll still just be full!

We record the operations on the small disk and replay them on the larger disk

# How to compare to non-full disks

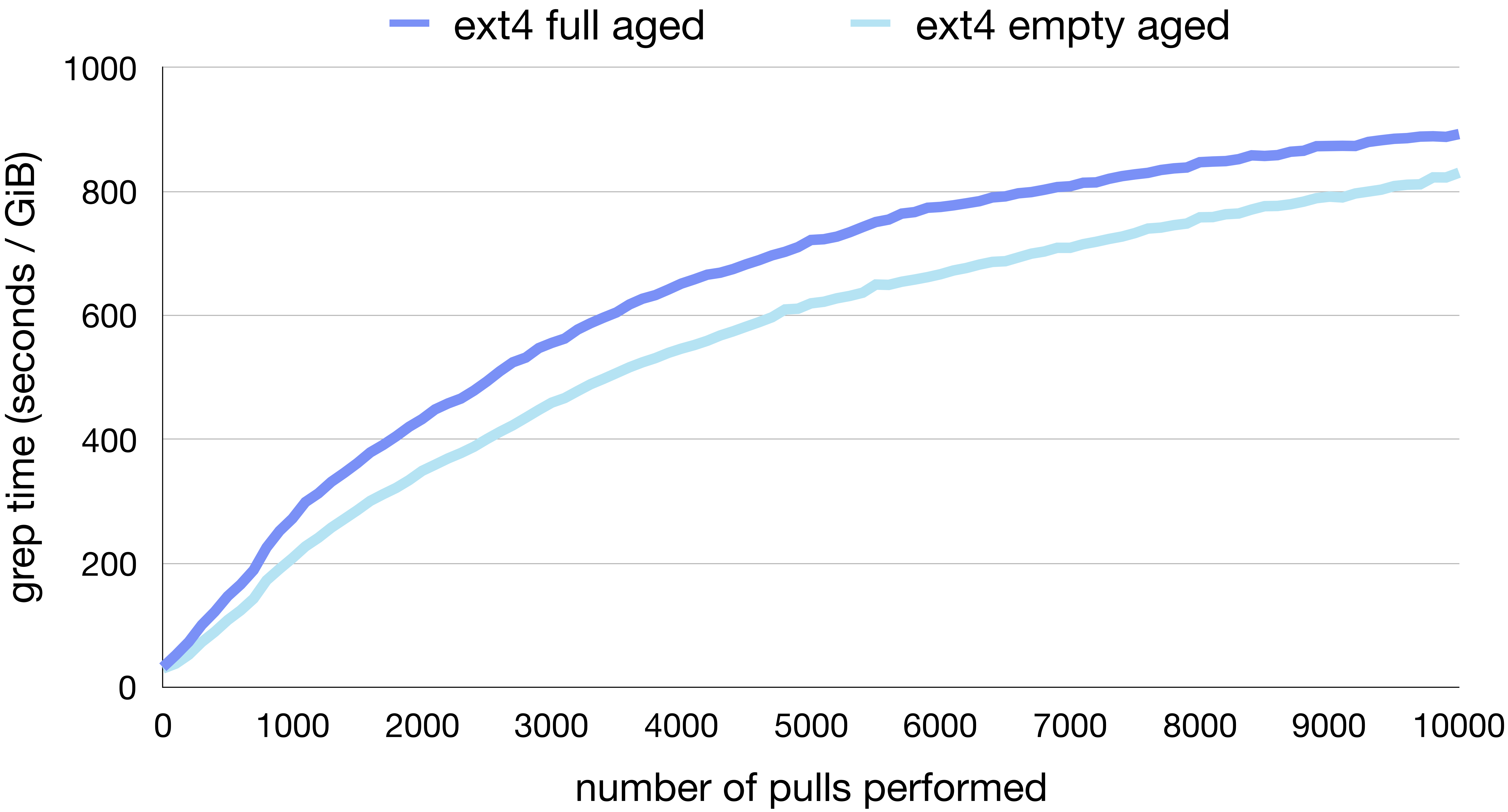


We can’t just use a larger disk with the git replay benchmark

It’ll still just be full!

We record the operations on the small disk and replay them on the larger disk

# Full Disk Git Aging on HDD (ext4)



Full: 5GiB partition
Empty: 50GiB partition
Unaged: 50GiB partition

System Details:

Dell PowerEdge T130

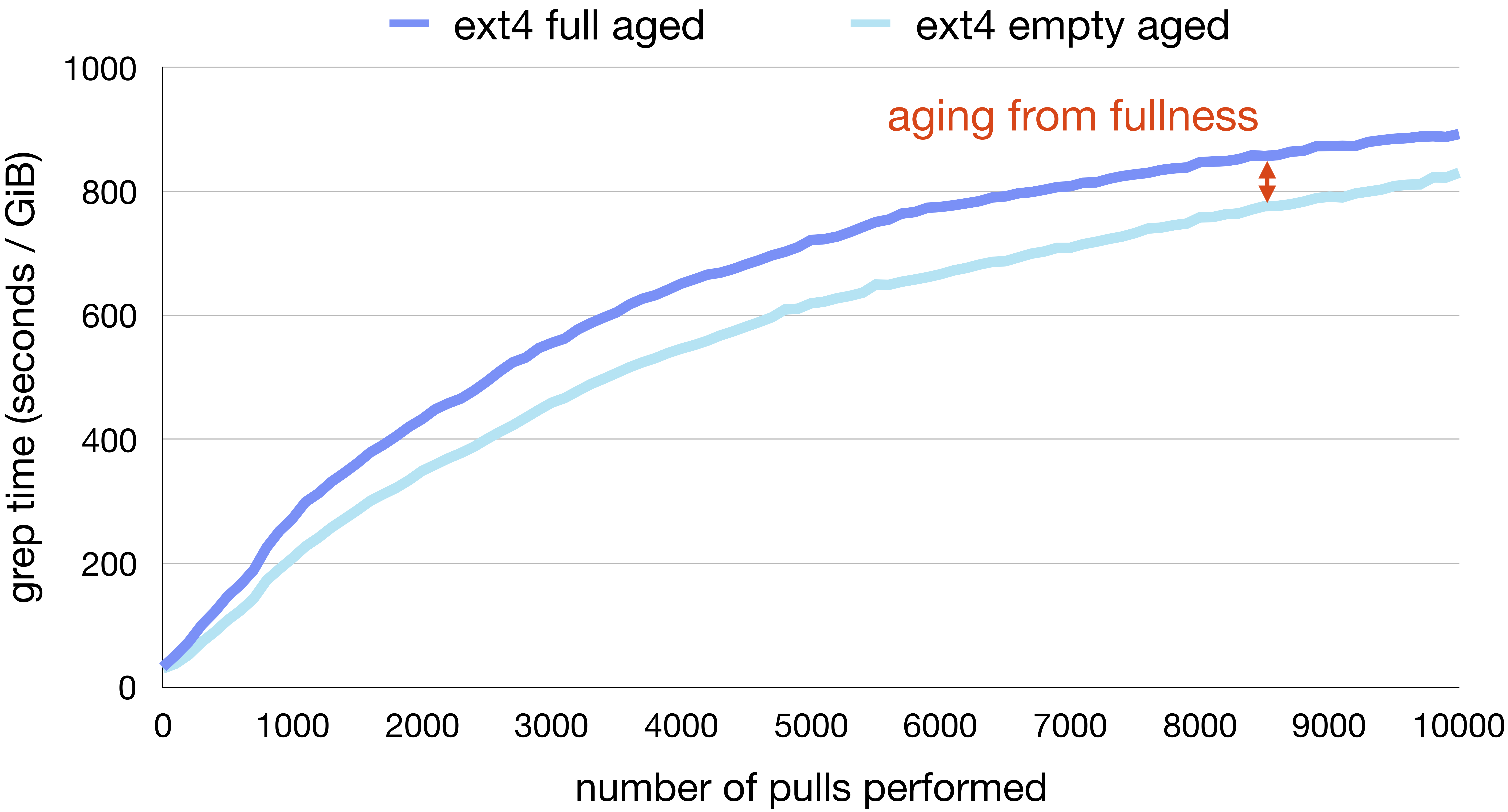
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

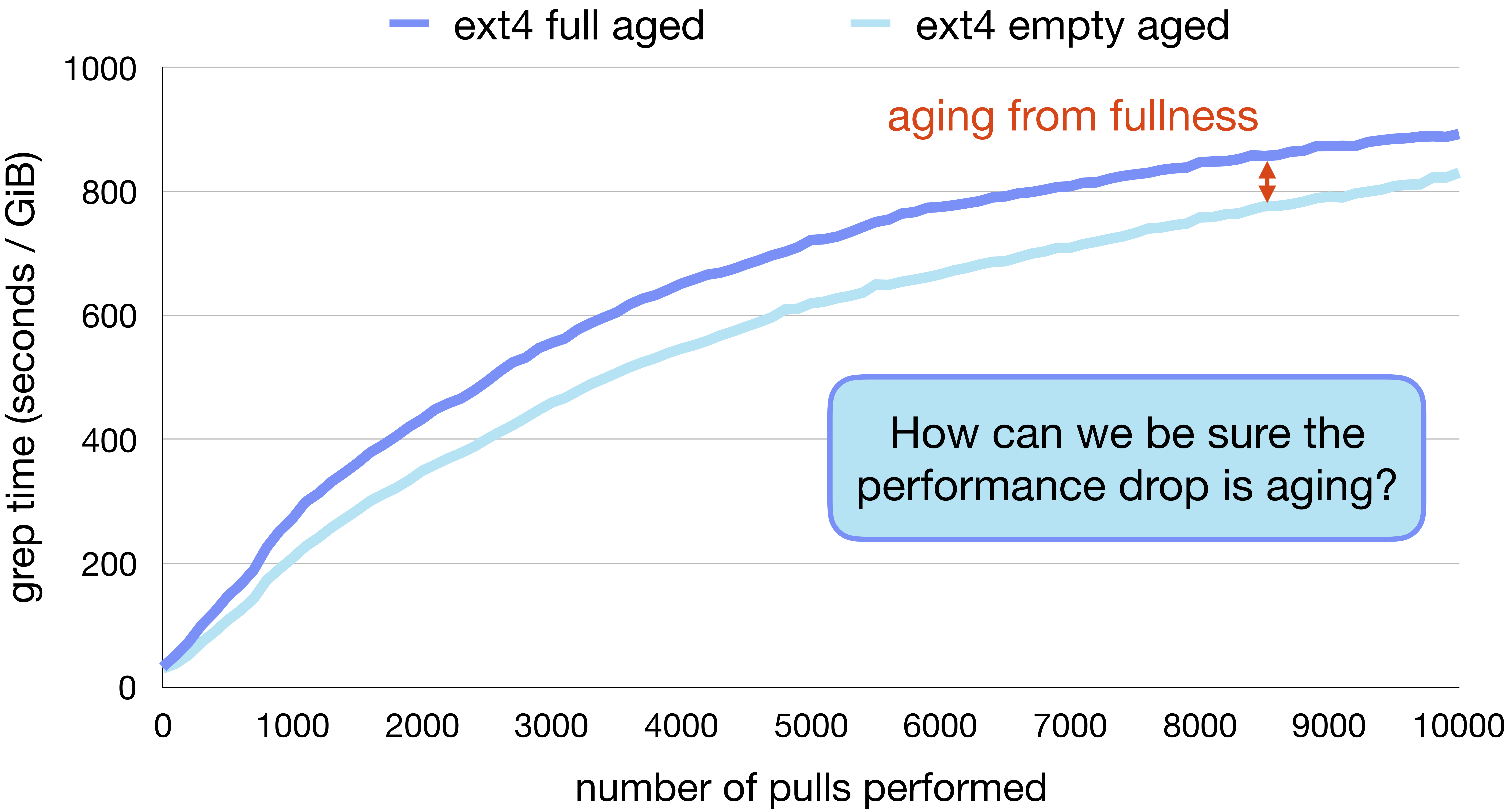
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



How can we be sure the performance drop is aging?

| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# How to isolate aging

# Unaged Baseline

Want an “unaged” baseline

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

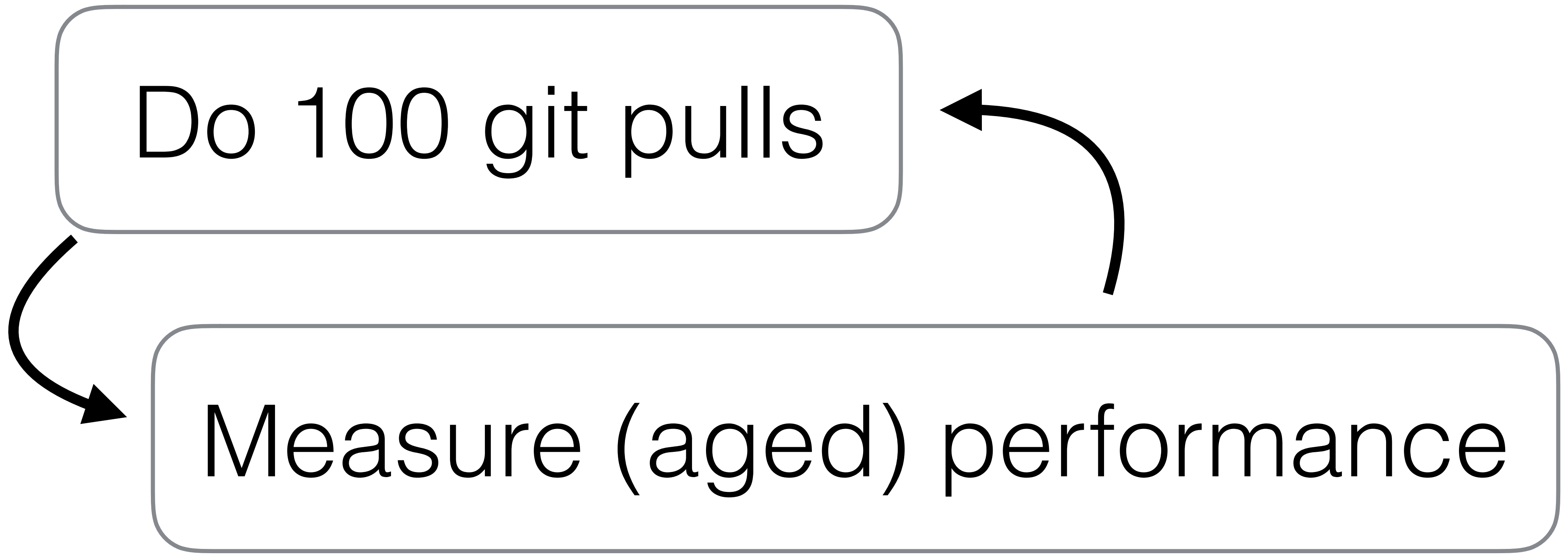
**Unaged baseline:**
Copy logical state to empty filesystem

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

**Unaged baseline:**
Copy logical state to empty filesystem



Do 100 git pulls

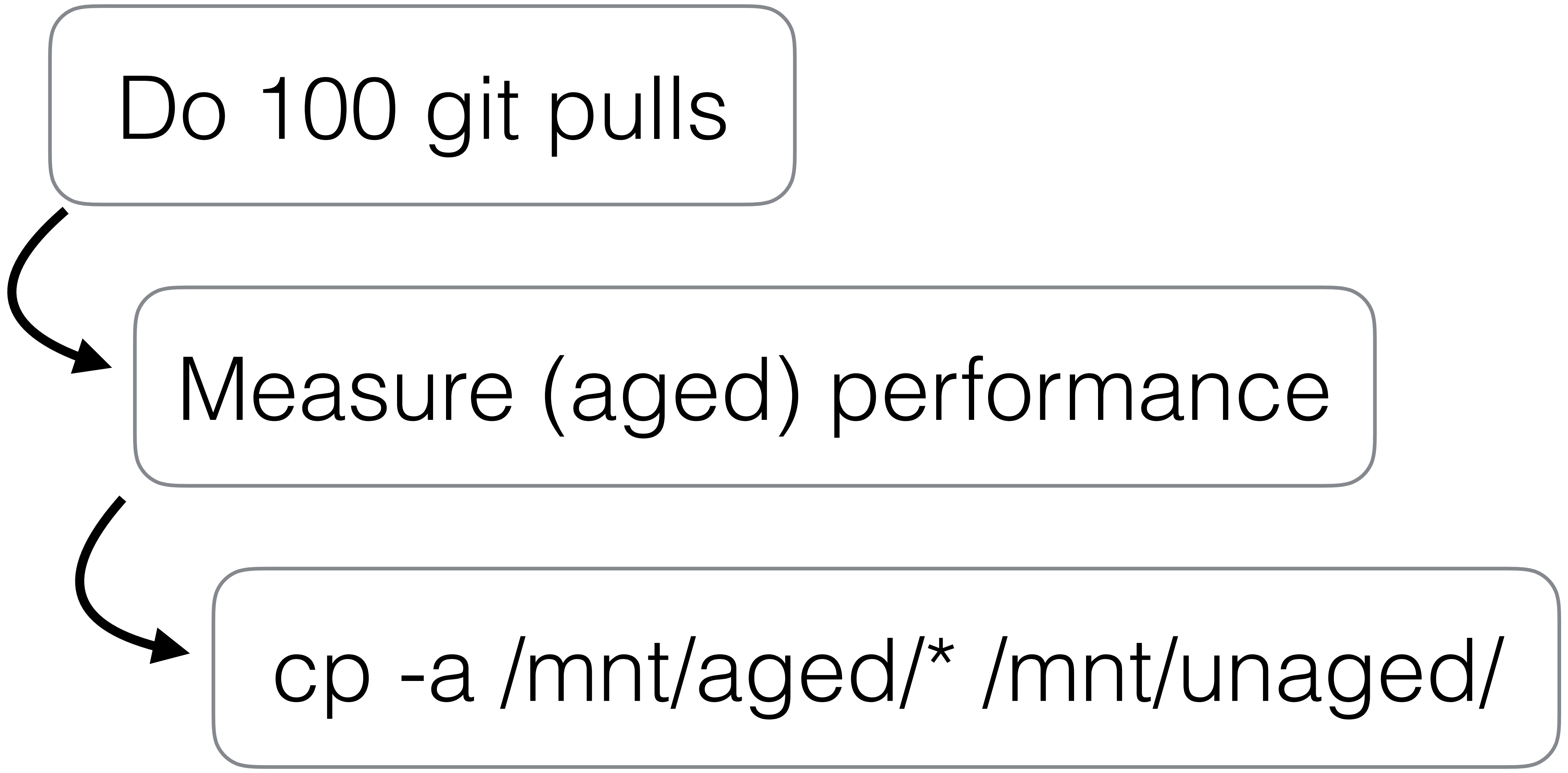
Measure (aged) performance

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

**Unaged baseline:**
Copy logical state to empty filesystem



Do 100 git pulls

Measure (aged) performance

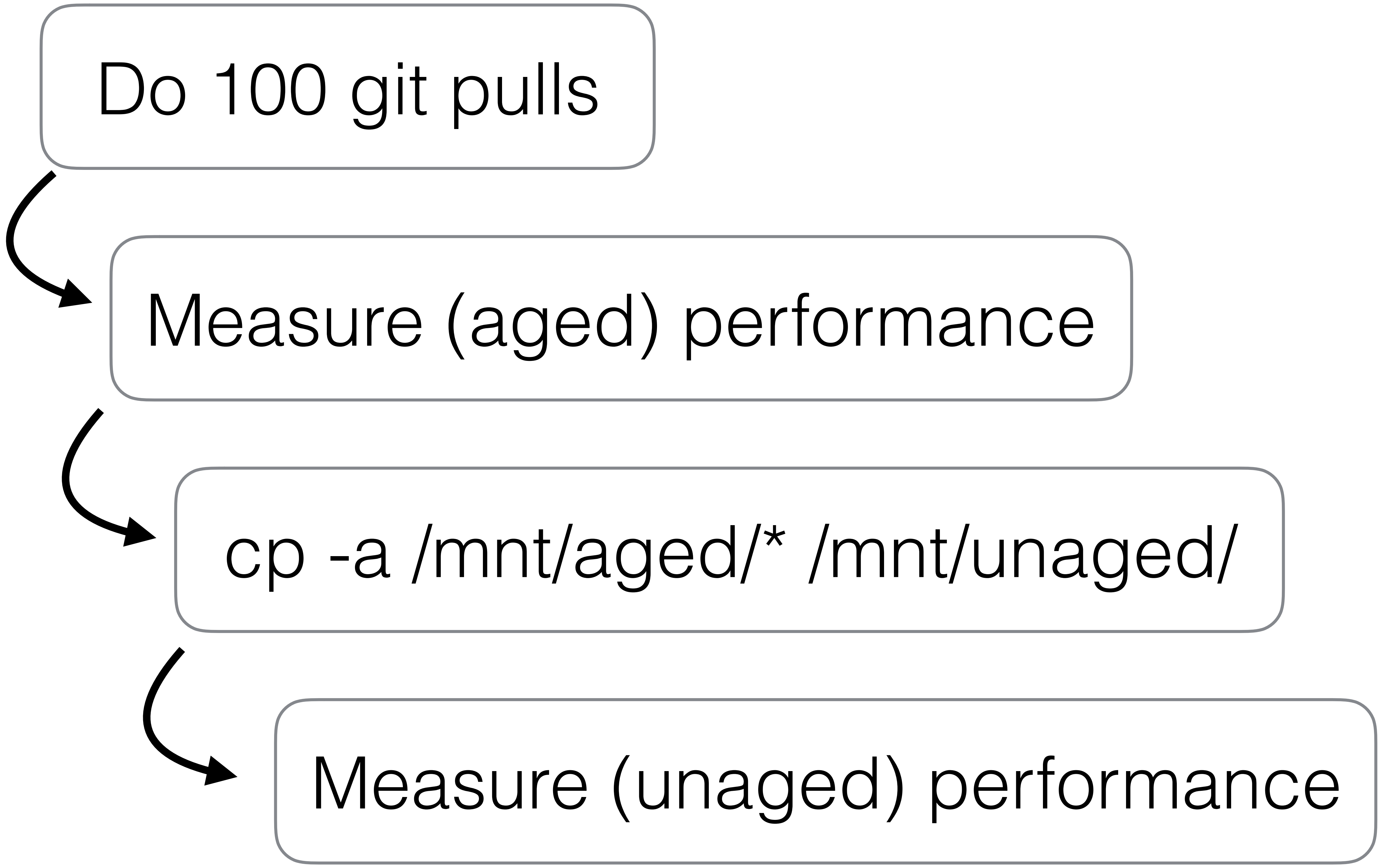
```
cp -a /mnt/aged/* /mnt/unaged/
```

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

**Unaged baseline:**
Copy logical state to empty filesystem



Do 100 git pulls

Measure (aged) performance

cp -a /mnt/aged/* /mnt/unaged/

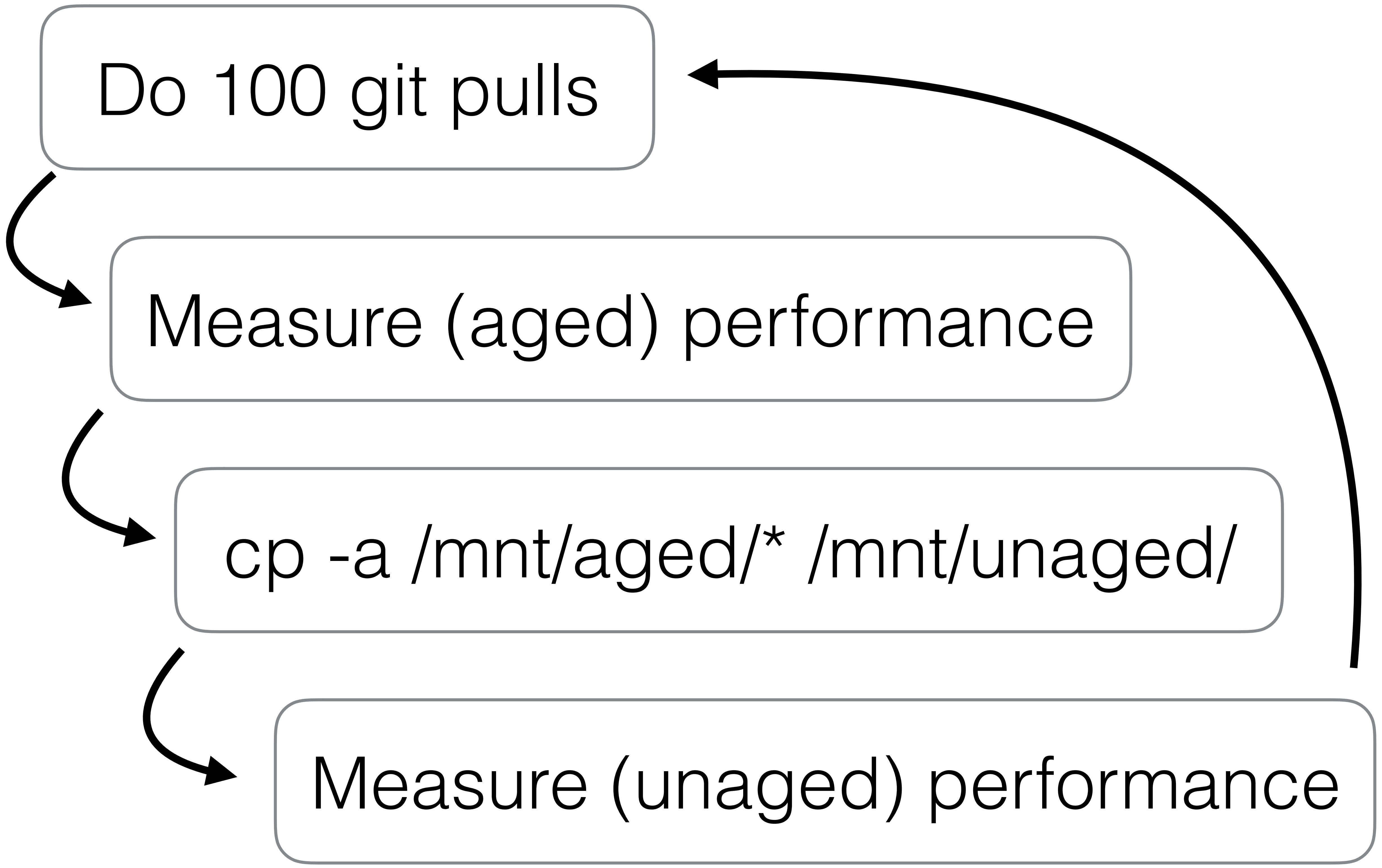
Measure (unaged) performance

# Unaged Baseline

Want an “unaged” baseline

“What the file system would do if the data had always been there”

**Unaged baseline:**
Copy logical state to empty filesystem



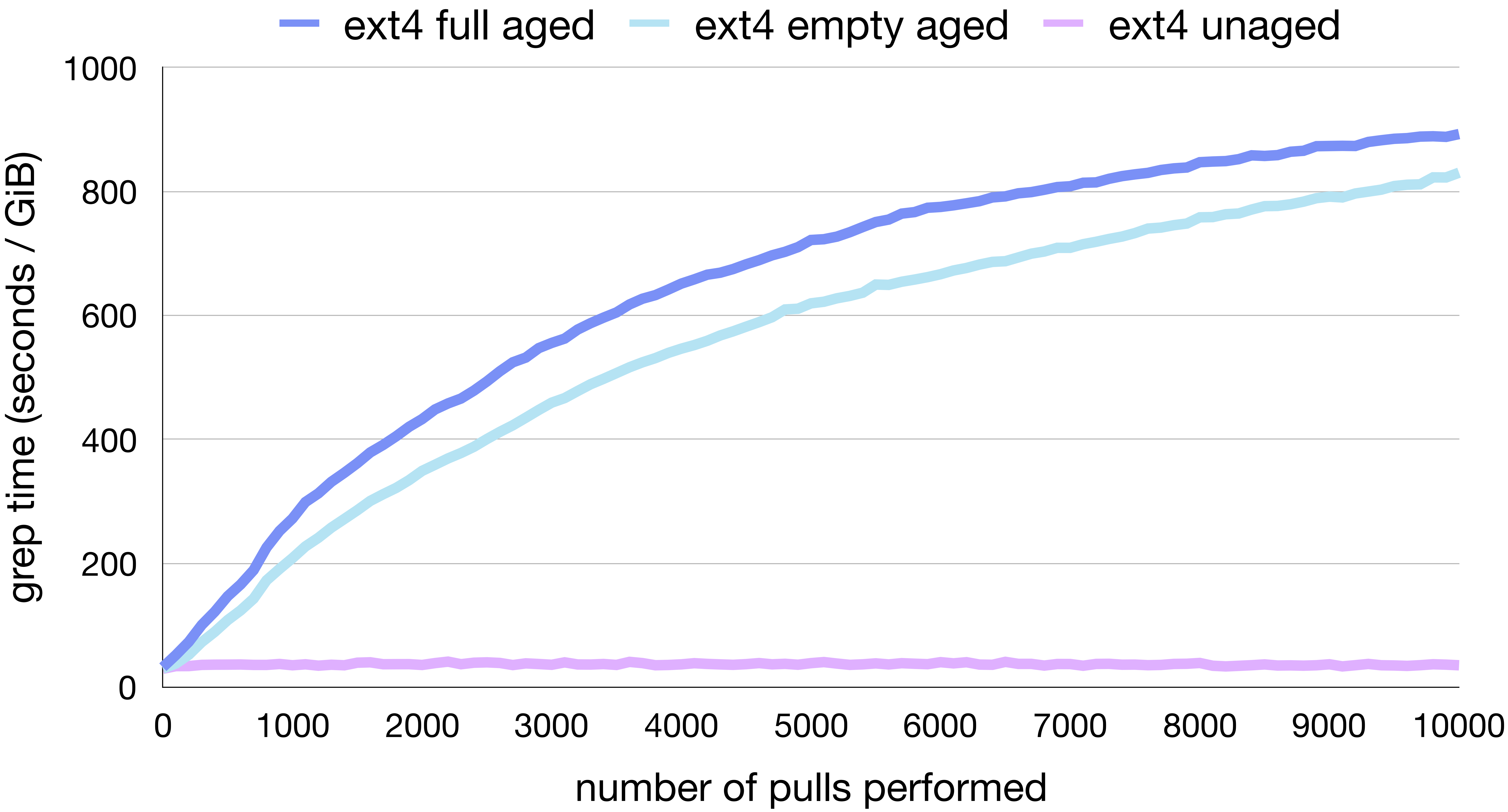
Do 100 git pulls

Measure (aged) performance

```
cp -a /mnt/aged/* /mnt/unaged/
```

Measure (unaged) performance

# Full Disk Git Aging on HDD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

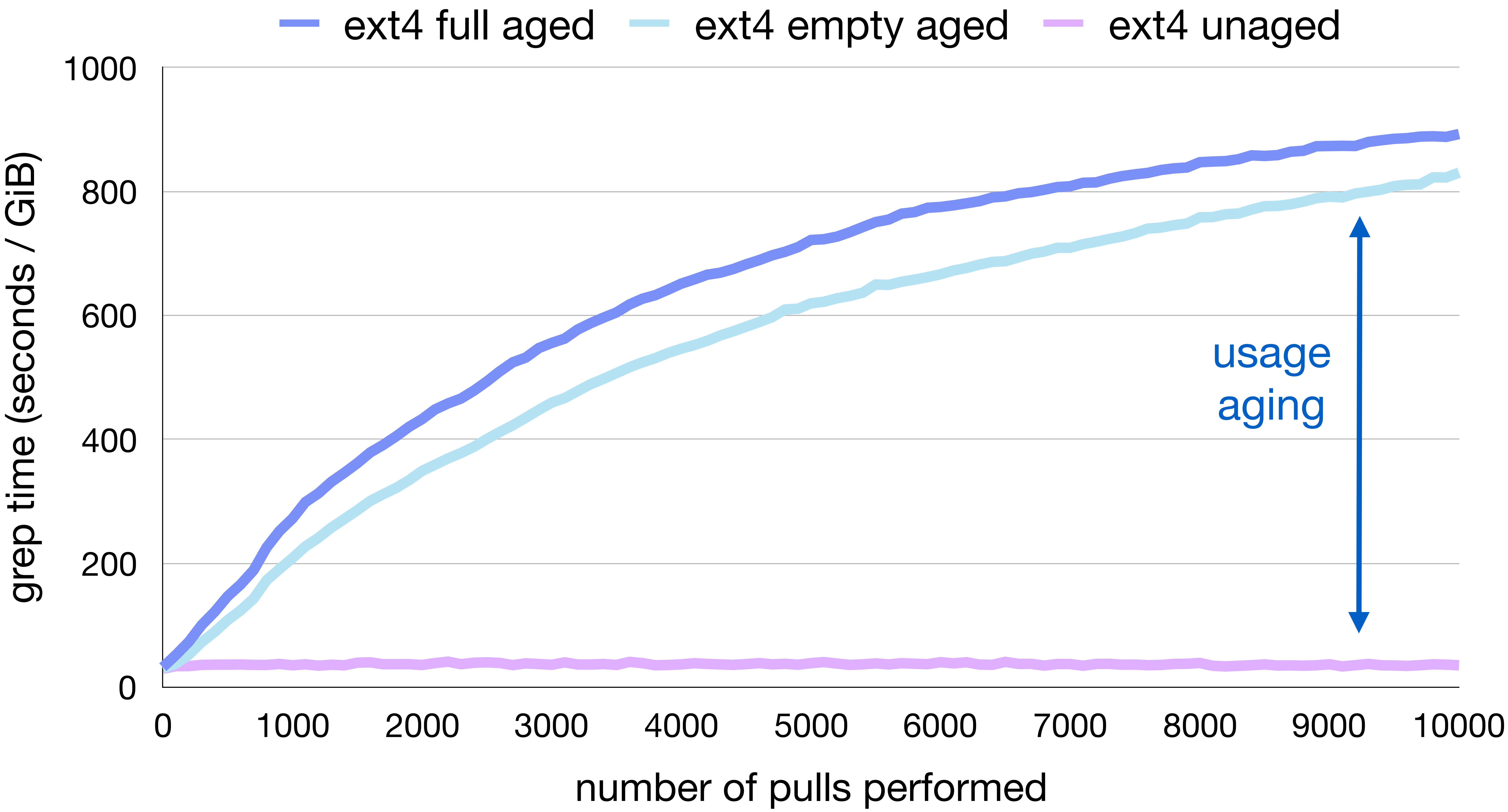
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

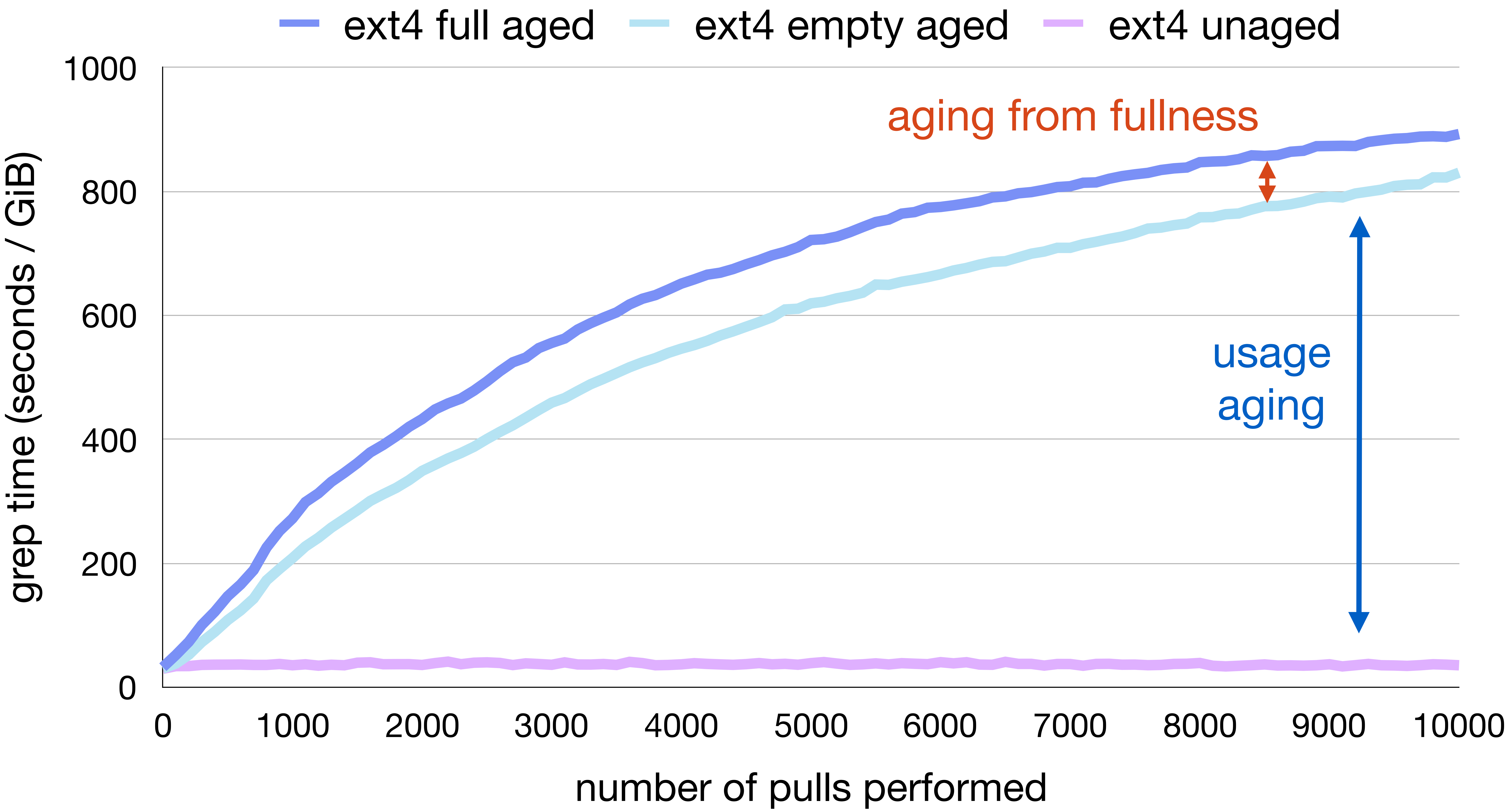
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

500GiB 7200 RPM
Toshiba HDD

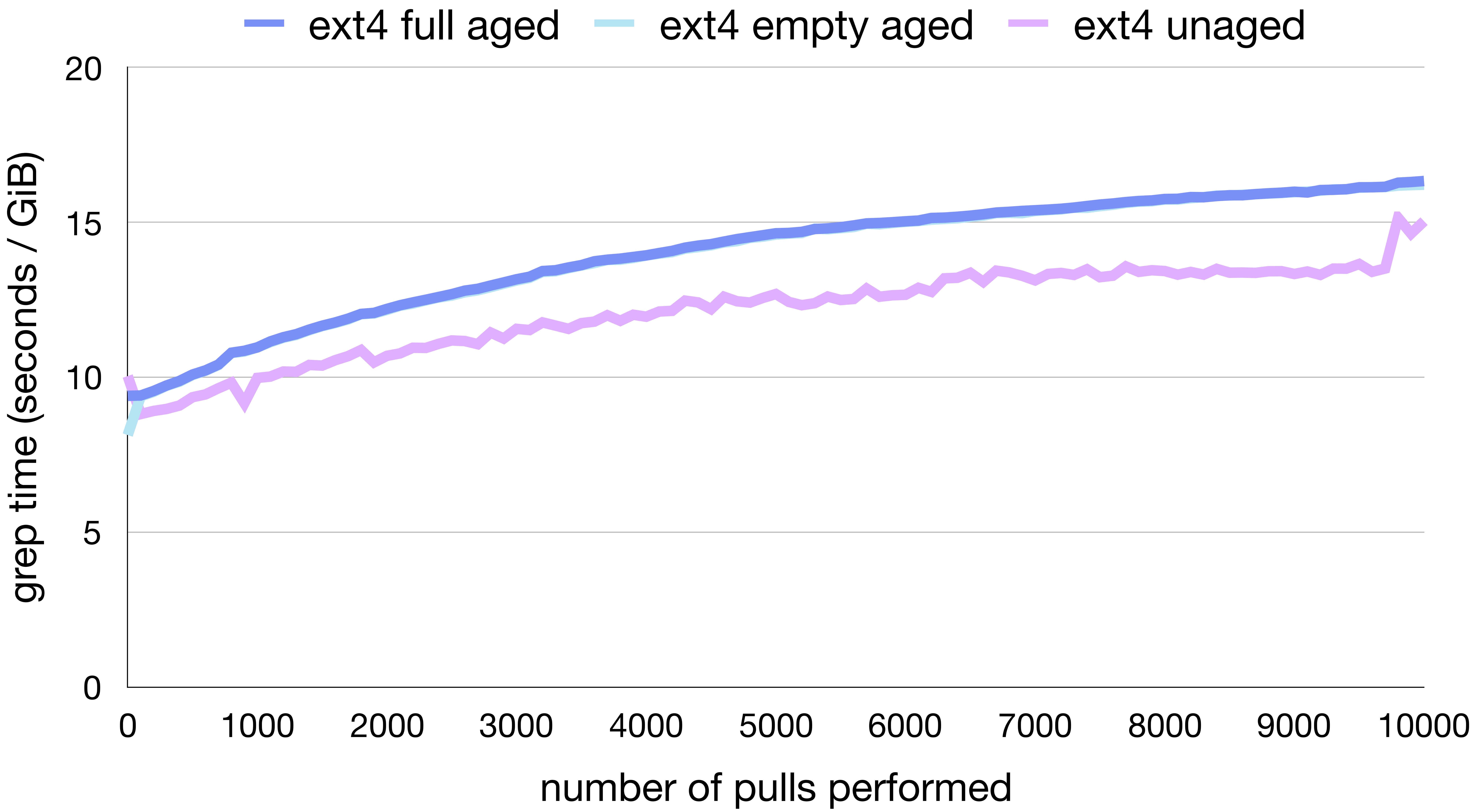
Ubuntu version 18.04 LTS
Linux kernel v 4.15

# Full Disk Git Aging on HDD (XFS and BTRFS)

XFS
BTRFS
grep time (seconds / GiB)
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000
number of pulls performed
XFS full aged
XFS empty aged
XFS unaged
BTRFS full aged
BTRFS empty aged
BTRFS unaged

# Full Disk Git Aging on SSD

# Full Disk Git Aging on SSD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

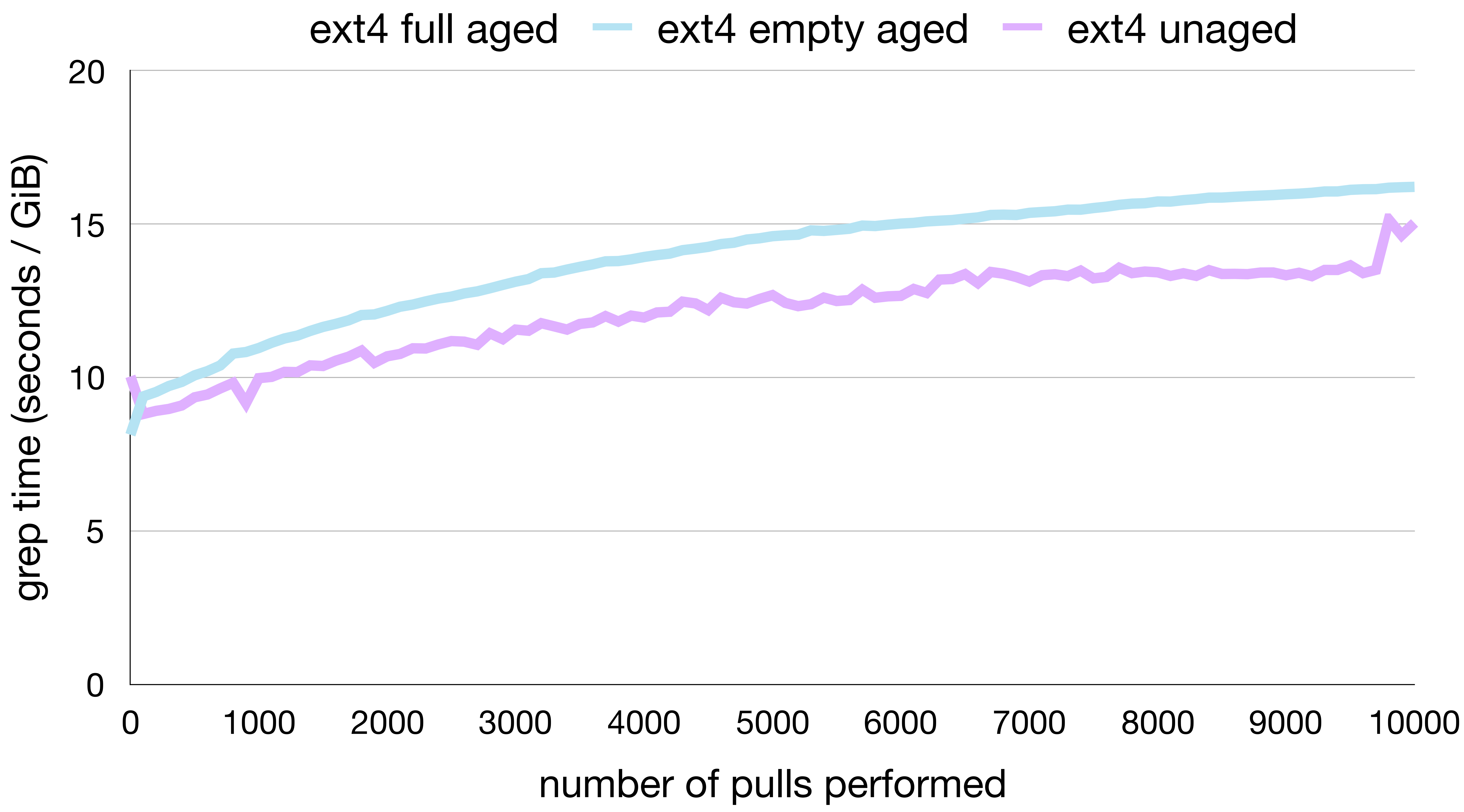
4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

250 GiB Samsung
860 EVO SSD

Ubuntu version 14.04 LTS
Linux kernel v 3.11

# Full Disk Git Aging on SSD (ext4)



| | |
|---|---|
| Full: | 5GiB partition |
| Empty: | 50GiB partition |
| Unaged: | 50GiB partition |

System Details:

Dell PowerEdge T130

4-core 3.00 GHz Intel®
Xeon(R) E3-1220 v6 CPU

16 GiB RAM

250 GiB Samsung
860 EVO SSD

Ubuntu version 14.04 LTS
Linux kernel v 3.11

# Full Disk Git Aging on SSD (ext4)

ext4
20
15
10
5
0
0
2000
4000
6000
8000
10000
ext4 full aged
ext4 empty aged
ext4 unaged
XFS
20
15
10
5
0
0
2000
4000
6000
8000
10000
XFS full aged
XFS empty aged
XFS unaged
BTRFS
20
15
10
5
0
0
2000
4000
6000
8000
10000
BTRFS full aged
BTRFS empty aged
BTRFS unaged
F2FS
20
15
10
5
0
0
2000
4000
6000
8000
10000
F2FS full aged
F2FS empty aged
F2FS unaged

# What About Free Space Fragmentation?

# Measuring Aging

How to measure free space fragmentation

# Measuring Aging

How to measure free space fragmentation

```
200 Device: /dev/sdc1
Blocksize: 4096 bytes
Total blocks: 1048576
Free blocks: 259934 (24.8%)

Min. free extent: 4 KB
Max. free extent: 653308 KB
Avg. free extent: 11680 KB
Num. free extent: 89

HISTOGRAM OF FREE EXTENT SIZES:
Extent Size Range :  Free extents   Free Blocks  Percent
    4K...    8K-  :            20            20    0.01%
    8K...   16K-  :             9            21    0.01%
   16K...   32K-  :             4            23    0.01%
   32K...   64K-  :             4            52    0.02%
   64K...  128K-  :             4            82    0.03%
  128K...  256K-  :             2            74    0.03%
  256K...  512K-  :             1            66    0.03%
    1M...    2M-  :            38         13576    5.22%
    4M...    8M-  :             1          1096    0.42%
    8M...   16M-  :             1          2160    0.83%
   32M...   64M-  :             1         13605    5.23%
   64M...  128M-  :             3         65832   25.33%
  512M... 1024M-  :             1        163327   62.83%
```

e2freefrag on ext4

# Measuring Aging

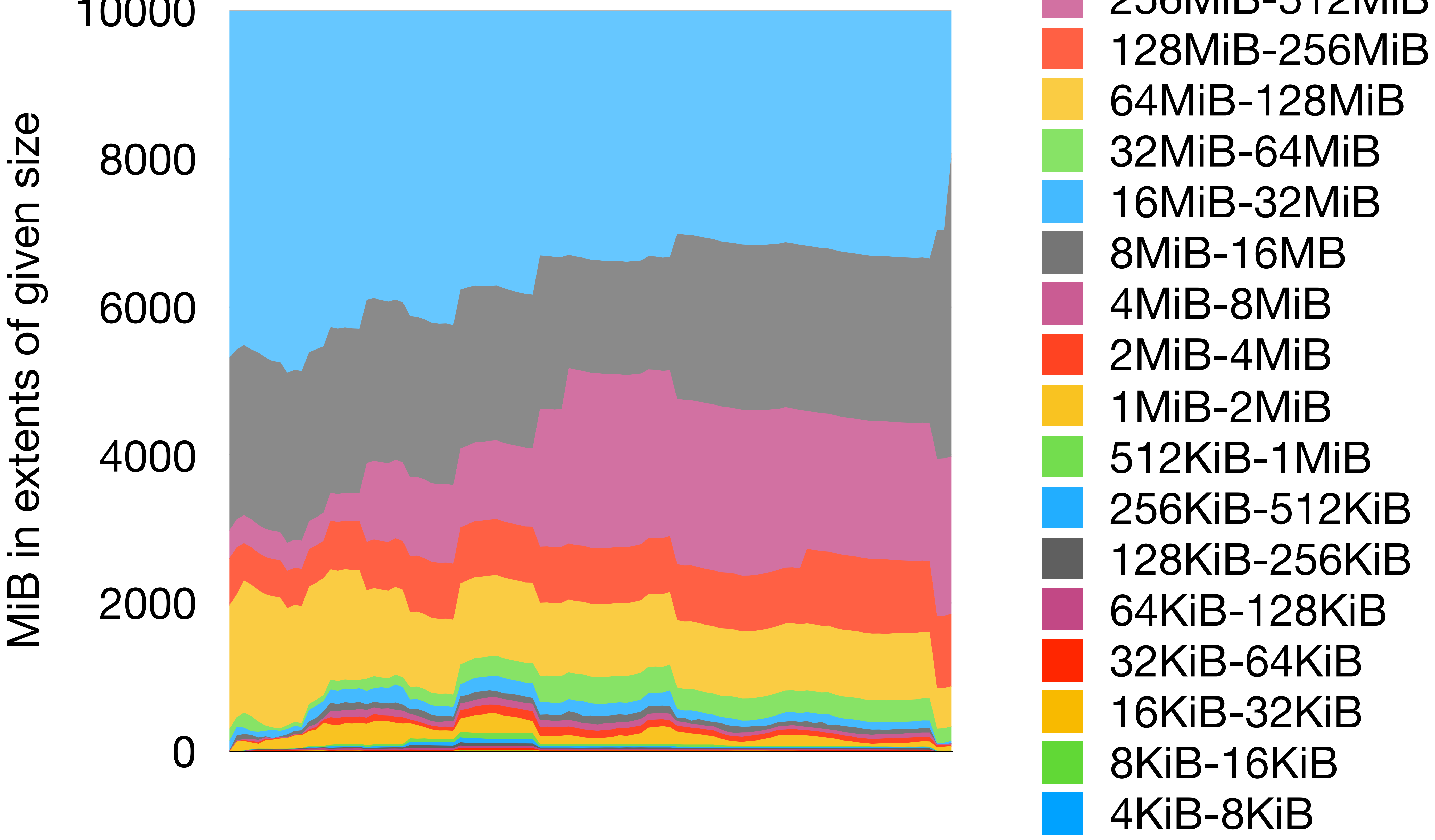
```
200 Device: /dev/sdc1
Blocksize: 4096 bytes
Total blocks: 1048576
Free blocks: 259934 (24.8%)

Min. free extent: 4 KB
Max. free extent: 653308 KB
Avg. free extent: 11680 KB
Num. free extent: 89

HISTOGRAM OF FREE EXTENT SIZES:
Extent Size Range :  Free extents   Free Blocks  Percent
    4K...    8K-  :            20            20    0.01%
    8K...   16K-  :             9            21    0.01%
   16K...   32K-  :             4            23    0.01%
   32K...   64K-  :             4            52    0.02%
   64K...  128K-  :             4            82    0.03%
  128K...  256K-  :             2            74    0.03%
  256K...  512K-  :             1            66    0.03%
    1M...    2M-  :            38         13576    5.22%
    4M...    8M-  :             1          1096    0.42%
    8M...   16M-  :             1          2160    0.83%
   32M...   64M-  :             1         13605    5.23%
   64M...  128M-  :             3         65832   25.33%
  512M... 1024M-  :             1        163327   62.83%
```

e2freefrag on ext4

## How to measure free space fragmentation



“histogram over time”

# Measuring Aging

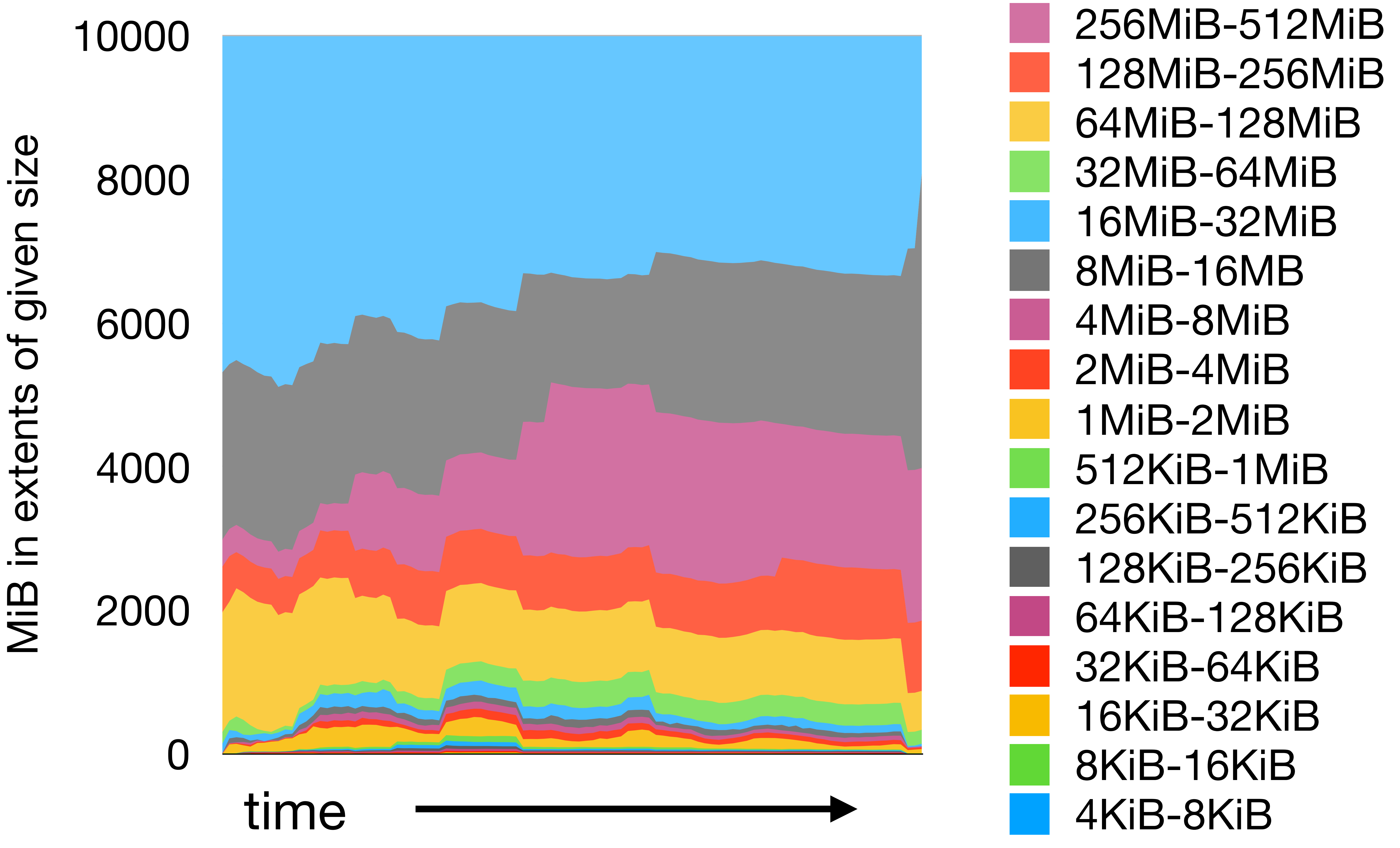
```
200 Device: /dev/sdc1
Blocksize: 4096 bytes
Total blocks: 1048576
Free blocks: 259934 (24.8%)

Min. free extent: 4 KB
Max. free extent: 653308 KB
Avg. free extent: 11680 KB
Num. free extent: 89

HISTOGRAM OF FREE EXTENT SIZES:
Extent Size Range :  Free extents   Free Blocks  Percent
    4K...    8K-  :            20            20    0.01%
    8K...   16K-  :             9            21    0.01%
   16K...   32K-  :             4            23    0.01%
   32K...   64K-  :             4            52    0.02%
   64K...  128K-  :             4            82    0.03%
  128K...  256K-  :             2            74    0.03%
  256K...  512K-  :             1            66    0.03%
    1M...    2M-  :            38         13576    5.22%
    4M...    8M-  :             1          1096    0.42%
    8M...   16M-  :             1          2160    0.83%
   32M...   64M-  :             1         13605    5.23%
   64M...  128M-  :             3         65832   25.33%
  512M... 1024M-  :             1        163327   62.83%
```

e2freefrag on ext4

## How to measure free space fragmentation



“histogram over time”

# Measuring Aging

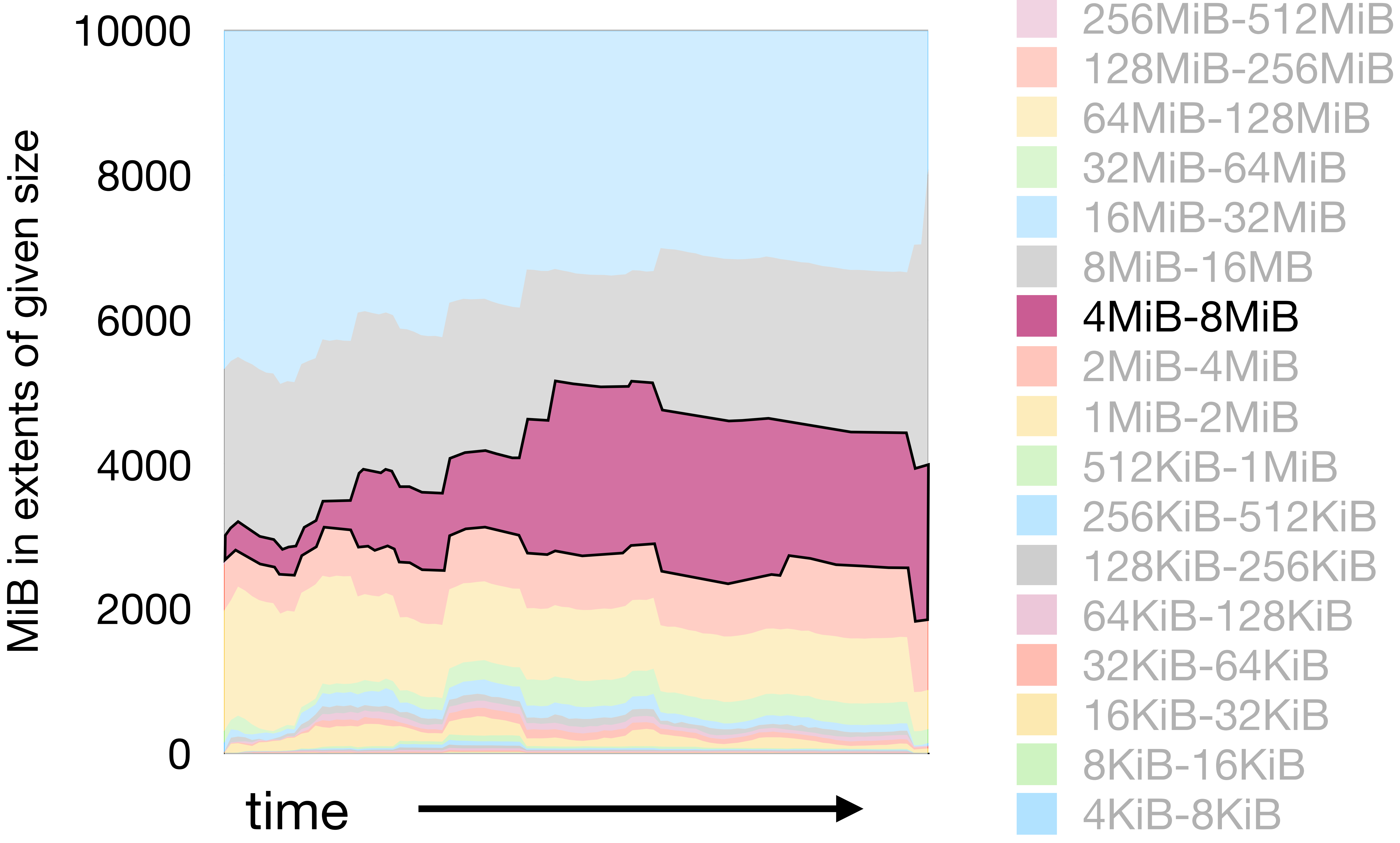
```
200 Device: /dev/sdc1
Blocksize: 4096 bytes
Total blocks: 1048576
Free blocks: 259934 (24.8%)

Min. free extent: 4 KB
Max. free extent: 653308 KB
Avg. free extent: 11680 KB
Num. free extent: 89

HISTOGRAM OF FREE EXTENT SIZES:
Extent Size Range :  Free extents   Free Blocks  Percent
    4K...    8K-  :            20            20    0.01%
    8K...   16K-  :             9            21    0.01%
   16K...   32K-  :             4            23    0.01%
   32K...   64K-  :             4            52    0.02%
   64K...  128K-  :             4            82    0.03%
  128K...  256K-  :             2            74    0.03%
  256K...  512K-  :             1            66    0.03%
    1M...    2M-  :            38         13576    5.22%
    4M...    8M-  :             1          1096    0.42%
    8M...   16M-  :             1          2160    0.83%
   32M...   64M-  :             1         13605    5.23%
   64M...  128M-  :             3         65832   25.33%
  512M... 1024M-  :             1        163327   62.83%
```

e2freefrag on ext4

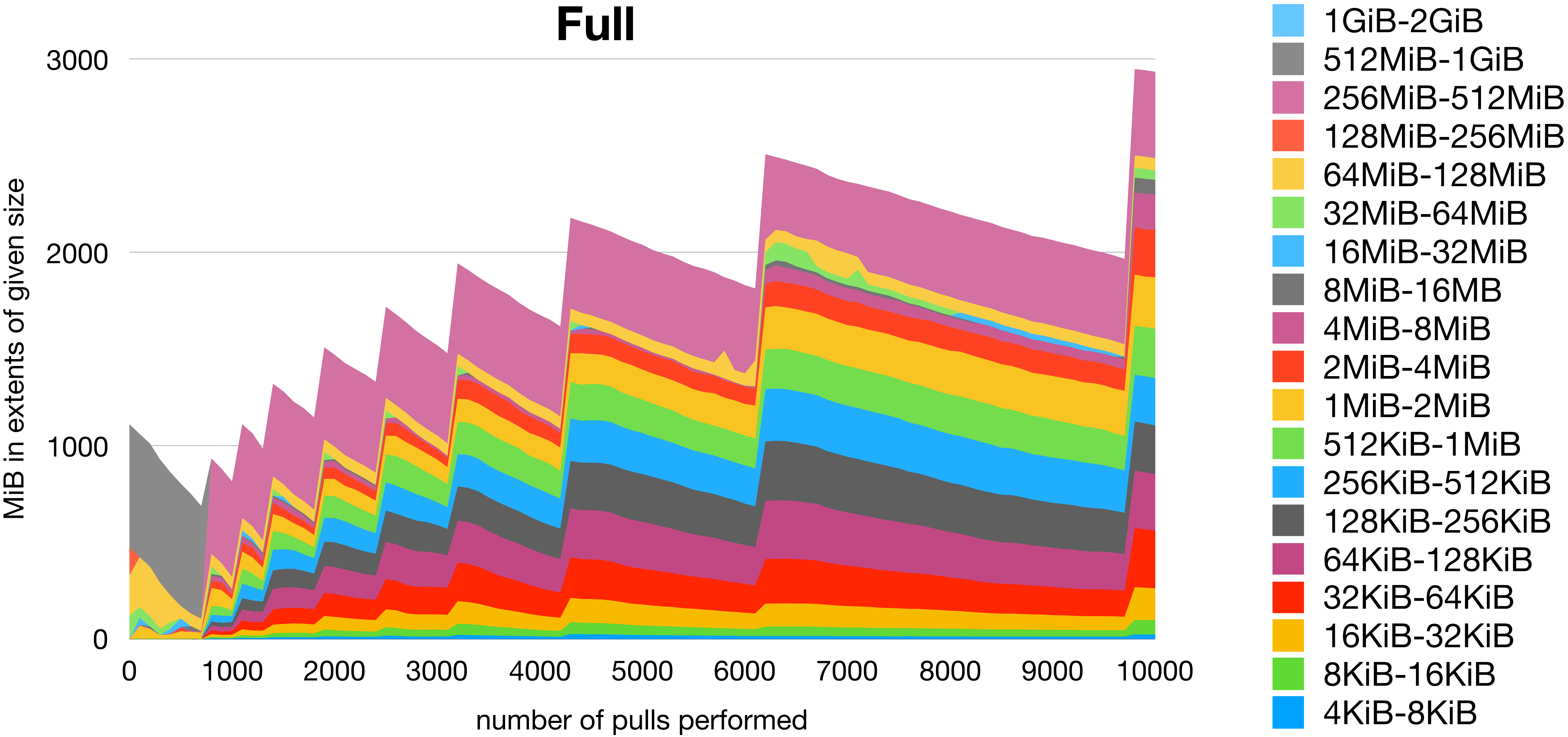
## How to measure free space fragmentation



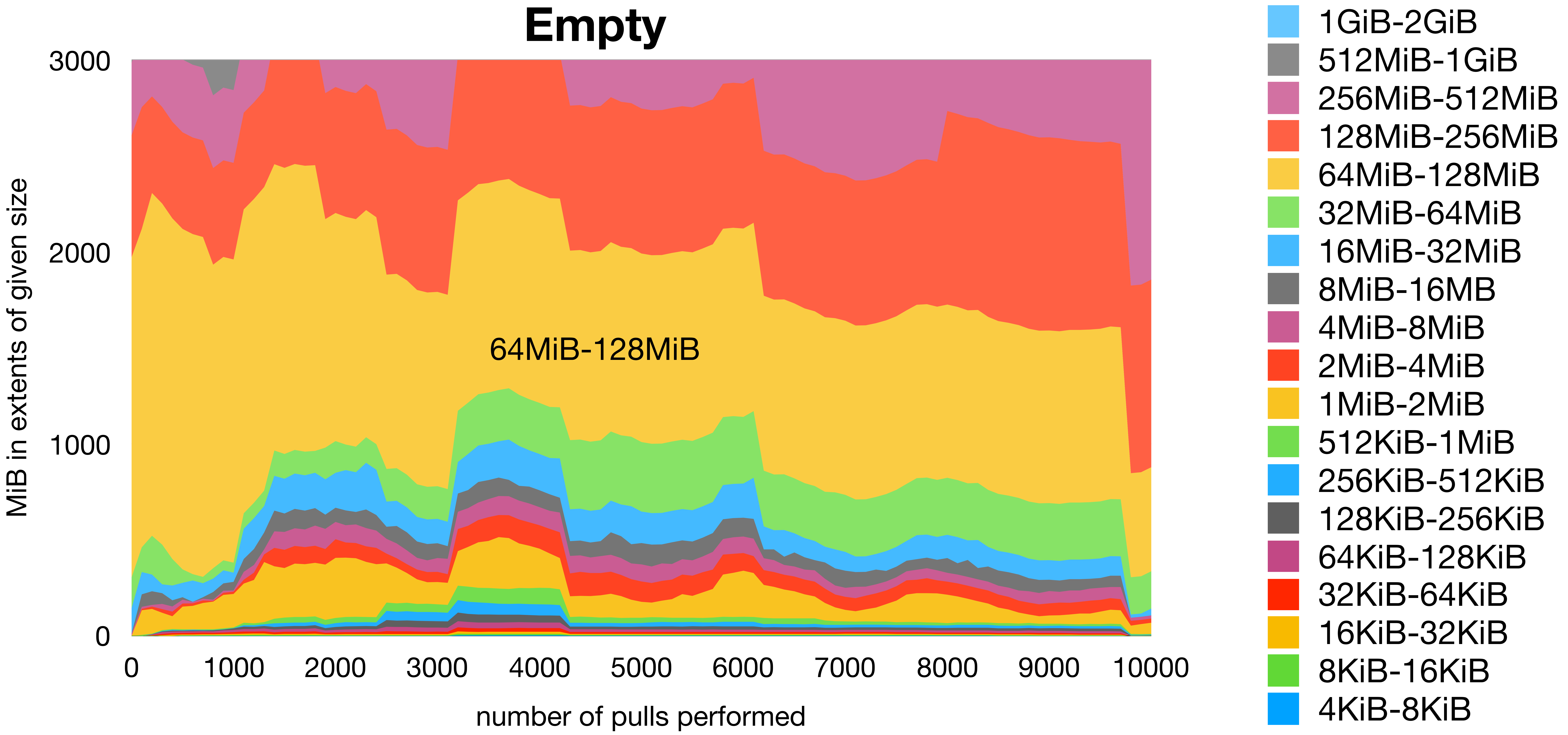
“histogram over time”

# Git Replay Benchmark:
# Free Space Fragmentation

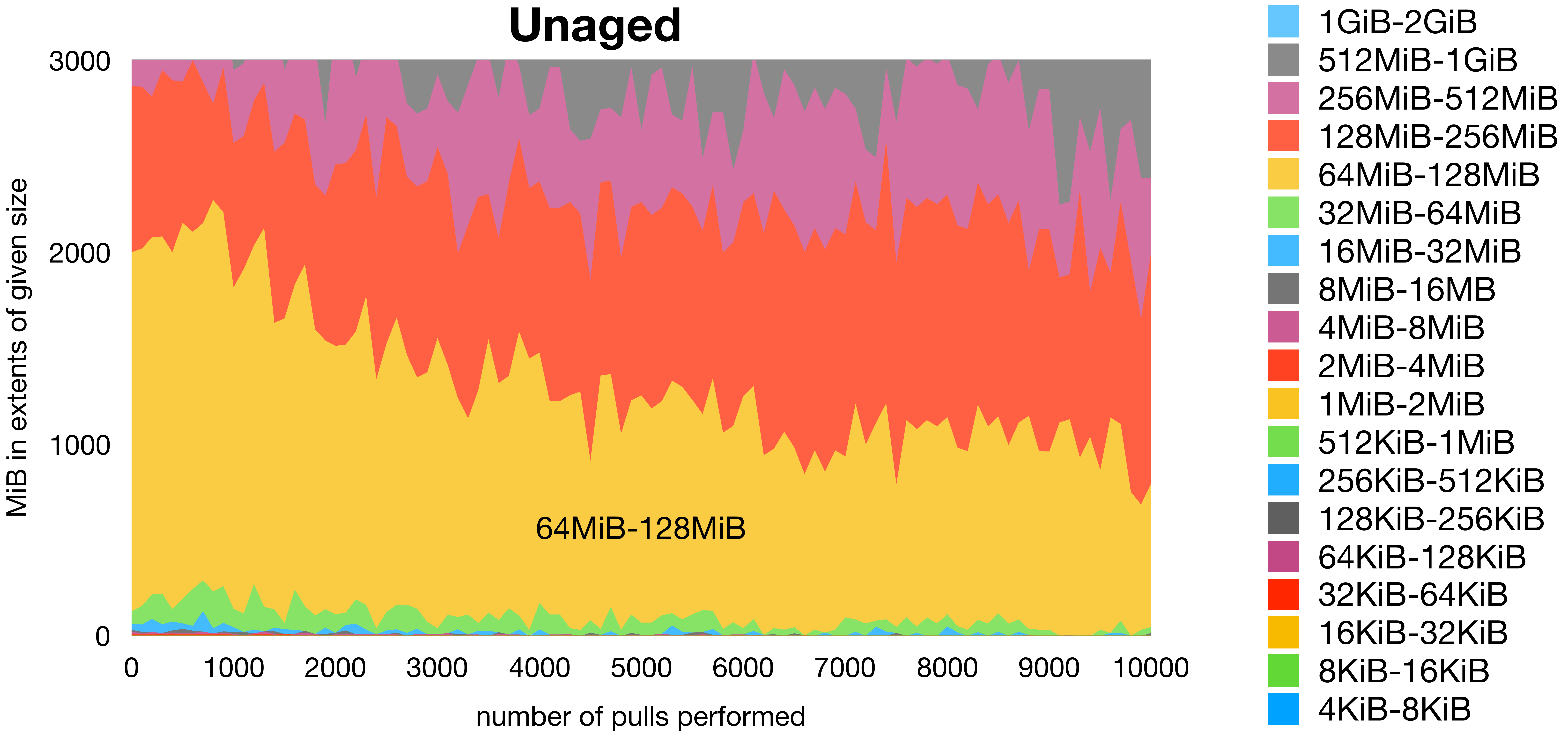
# Full Disk Git Aging Free Space Fragmentation on ext4



# Full Disk Git Aging Free Space Fragmentation on ext4



# Full Disk Git Aging Free Space Fragmentation on ext4



# What we learned

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

Worst case benchmark (see paper):

Aging due to use >= aging due to fullness

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

Worst case benchmark (see paper):

Aging due to use >= aging due to fullness

Suggests that for most workloads, use aging is the first-order effect

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

Worst case benchmark (see paper):

Aging due to use >= aging due to fullness

Suggests that for most workloads, use aging is the first-order effect

Disk Fullness

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

Worst case benchmark (see paper):

Aging due to use >= aging due to fullness

Suggests that for most workloads, use aging is the first-order effect

Disk Fullness ⇛ Free-space fragmentation

# What we learned

Git benchmark:

Aging due to use >> aging due to fullness

Worst case benchmark (see paper):

Aging due to use >= aging due to fullness

Suggests that for most workloads, use aging is the first-order effect

Disk Fullness ⇒ Free-space fragmentation $\overset{?}{\Rightarrow}$ File System Aging