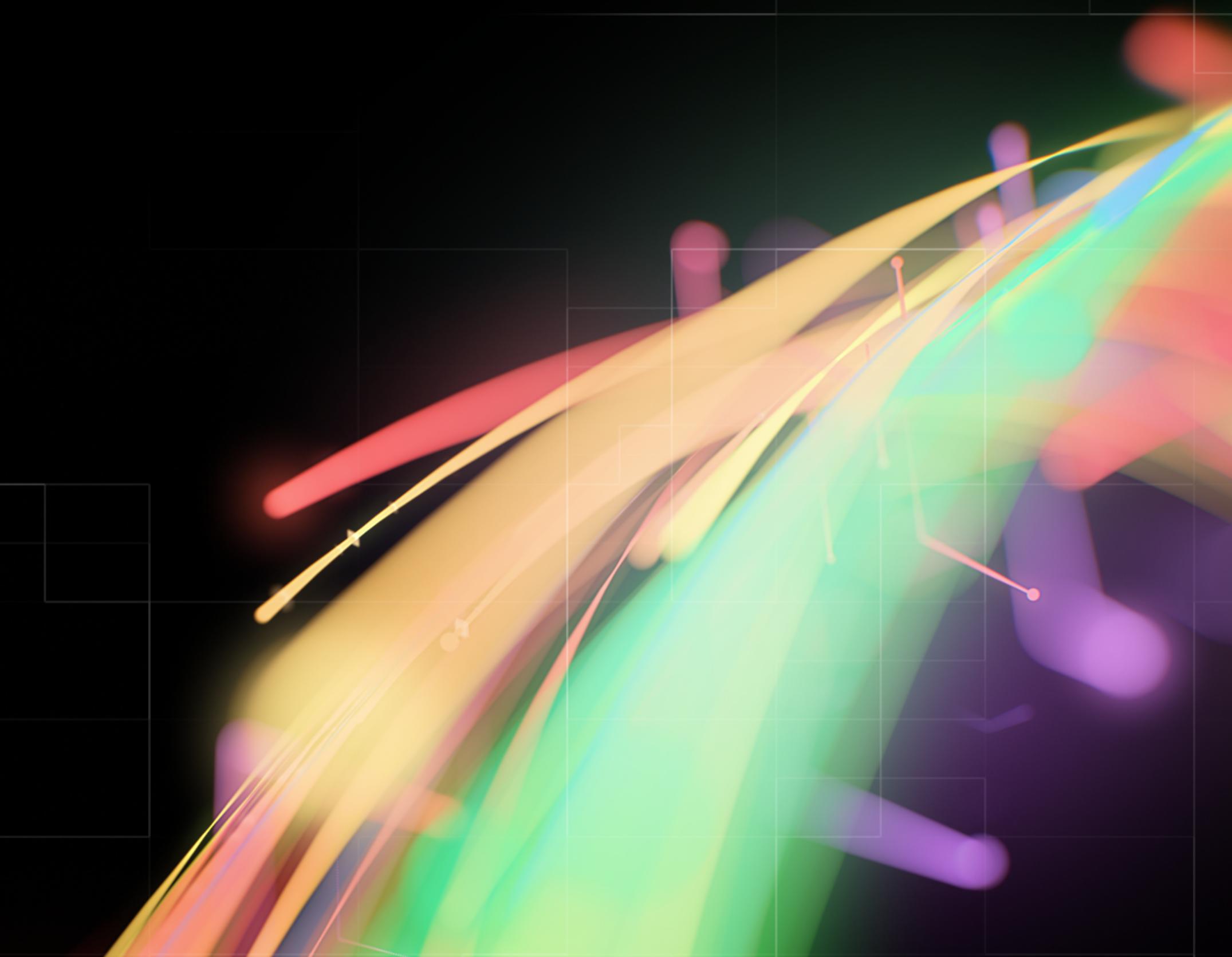




How to keep Git monorepos manageable



@droidpl & @selkins13 / How to keep Git monorepos manageable
Universe 2020



Presenters



Sarah Elkins

@selkins13



Javier de Pedro López

@droidpl

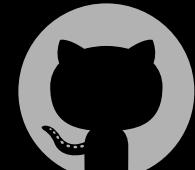


Prerequisites

<https://github.com/githubuniverseworkshops/grafting-monorepos#mega-prerequisites>

Hopefully you brought...

- A clone or fork of a repository you would like to reduce in size
- Time it takes to clone the repository
`time git clone [url]`
- Install git-sizer
- Install git-filter-repo
- Clone the workshop repository



Workshop Sections

History

How Git handles history

- Components of a commit
- Contributors to the bloat
- Git LFS
- Grafting
- 10 minute activity

Analyzing a Git Repository

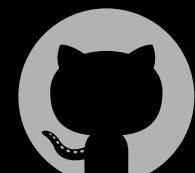
Why history matters

- Effects of a large repository
- Tools to analyze the repository
- Customer examples
- 20 minute activity
- Prevention tips

Grafting A monorepo

Reducing the size

- Cleaning the legacy repository
- Preparing the new repository
- Grafting the old and the new
- 20 minute activity
- Expected Outcomes



Workshop Sections

History

How Git handles history

- Components of a commit
- Contributors to the bloat
- Git LFS
- Grafting
- 10 minute activity

Analyzing a Git Repository

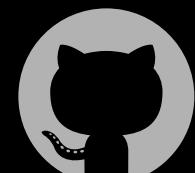
Why history matters

- Effects of a large repository
- Tools to analyze the repository
- Customer examples
- 20 minute activity
- Prevention tips

Grafting A monorepo

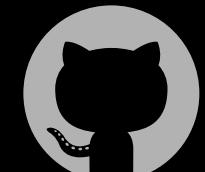
Reducing the size

- Cleaning the legacy repository
- Preparing the new repository
- Grafting the old and the new
- 20 minute activity
- Expected Outcomes



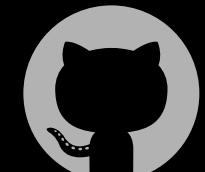
Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**



Components of a commit

- ▶ **commit message**
 - A short description of the changes that are being made that is saved at the time of the commit action with the changeset
 - The pattern of the commit message subject line often completes the sentence “This changeset will ...”
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**



Components of a commit

- ▶ **commit message**
 - ▶ **committer**
 - ▶ **commit date**
 - ▶ **author**
 - ▶ **author date**
 - ▶ **tree**
 - ▶ **parents**
- The identity of the person or account making the commit
 - This can be different than the person that suggests/makes the changes to the code base



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
 - The date/time the commit is made
 - Coordinated Universal Time (UTC)
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**
 - The identity of the author of the committed changeset
 - Can be different from the committer



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
 - The date/time the changes where made
 - Coordinated Universal Time (UTC)
- ▶ **tree**
- ▶ **parents**



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
 - Corresponds to UNIX directory entries
 - Represents a particular directory state of a working directory
 - Contains one or more entries
 - Each entry contains the SHA-1 hash of a blob or subtree
- ▶ **parents**



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**
 - A commit object may have multiple parents
 - One parent indicates an ordinary commit
 - More than one parent indicates the commit is a merge between several lines of history
 - Initial commits have no parents



Components of a commit

- ▶ **commit message**
- ▶ **committer**
- ▶ **commit date**
- ▶ **author**
- ▶ **author date**
- ▶ **tree**
- ▶ **parents**
 - A commit object may have multiple parents
 - One parent indicates an ordinary commit
 - More than one parent indicates the commit is a merge between several lines of history
 - **Initial commits have no parents**



Each time there is a commit or a merge, it is associated with a HASH.

```
> git log --oneline

c2e7554e1b85 (HEAD -> master, origin/master, origin/HEAD) Merge tag 'gfs2-v5.10-rc4-fixes' of
git://git.kernel.org/pub/scm/linux/kernel/git/gfs2/linux-gfs2
ce228d459424 Merge tag 'nfsd-5.10-2' of git://linux-nfs.org/~bfields/linux
f86fee1845ee Merge tag 'linux-kselftest-kunit-fixes-5.10-rc5' of git://git.kernel.org/pub/scm/linux/kernel/git/shuah/linux-kselftest
20b329129009 gfs2: Fix regression in freeze_go_sync
0fa8ee0d9ab9 Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/dtor/input
111e91a6df50 Merge tag 's390-5.10-4' of git://git.kernel.org/pub/scm/linux/kernel/git/s390/linux
ed129cd75ac1 Merge tag 'mips_fixes_5.10_1' of git://git.kernel.org/pub/scm/linux/kernel/git/mips/linux
be1dd6692adb Merge tag 'perf-tools-fixes-for-v5.10-2020-11-17' of git://git.kernel.org/pub/scm/linux/kernel/git/acme/linux
9dacf44c3837 Merge branch 'urgent-fixes' of git://git.kernel.org/pub/scm/linux/kernel/git/paulmck/linux-rCU
ac3b57adf87a MIPS: Alchemy: Fix memleak in alchemy_clk_setup_cpu
61a2f1aecf60 MIPS: kernel: Fix for_each_memblock conversion
9c87c9f41245 Merge tag 'arm-soc-fixes-v5.10-2' of git://git.kernel.org/pub/scm/linux/kernel/git/soc/soc
a5698b3835f5 Merge tag 'hyperv-fixes-signed' of git://git.kernel.org/pub/scm/linux/kernel/git/hyperv/linux
a08f4523243c Merge tag 'for_linus' of git://git.kernel.org/pub/scm/linux/kernel/git/mst/vhost
568beb27959b perf test: Avoid an msan warning in a copied stack.
1c756cd429d8 perf inject: Fix file corruption due to event deletion
cc05af8e2e91 Merge tag 'imx-fixes-5.10-4' of git://git.kernel.org/pub/scm/linux/kernel/git/shawnguo/linux into arm/fixes
39c8d39c04bb Merge tag 'renesas-fixes-for-v5.10-tag1' of git://git.kernel.org/pub/scm/linux/kernel/git/geert/renesas-devel into
arm/fixes
09162bc32c88 (tag: v5.10-rc4) Linux 5.10-rc4
efd838fec17b vhost scsi: Add support for LUN resets. [This line is highlighted]
18f1becb6948 vhost scsi: add lun parser helper
47a3565e8bb1 vhost scsi: fix cmd completion race
25b98b64e284 vhost scsi: alloc cmds per vq instead of session
```

**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <mst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>
Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b

commit efd838fec17bd8756da852a435800a7e6281bfbc
Author: Mike Christie <michael.christie@oracle.com>
AuthorDate: Mon Nov 9 23:33:23 2020 -0600
Commit: Michael S. Tsirkin <msst@redhat.com>
CommitDate: Sun Nov 15 17:30:55 2020 -0500

vhost scsi: Add support for LUN resets.
```

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

```
Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>
```



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500  
  
vhost scsi: Add support for LUN resets.
```

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

```
Signed-off-by: Mike Christie <michael.christie@oracle.com>  
Link: https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com  
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>  
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>
```



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

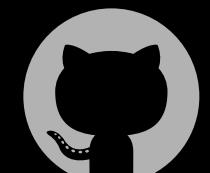
vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b

commit efd838fec17bd8756da852a435800a7e6281bfbc
Author: Mike Christie <michael.christie@oracle.com>
AuthorDate: Mon Nov 9 23:33:23 2020 -0600
Commit: Michael S. Tsirkin <msst@redhat.com>
CommitDate: Sun Nov 15 17:30:55 2020 -0500

vhost scsi: Add support for LUN resets.
```

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

```
Signed-off-by: Mike Christie <michael.christie@oracle.com>
Link: https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com
Signed-off-by: Michael S. Tsirkin <msst@redhat.com>
Acked-by: Stefan Hajnoczi <stefanha@redhat.com>
```



**That HASH is a point in time for
that collection of changes that can
be referenced in the HISTORY of
the repository.**

```
> git log --pretty=fuller efd838fec17b  
  
commit efd838fec17bd8756da852a435800a7e6281bfbc  
Author: Mike Christie <michael.christie@oracle.com>  
AuthorDate: Mon Nov 9 23:33:23 2020 -0600  
Commit: Michael S. Tsirkin <msst@redhat.com>  
CommitDate: Sun Nov 15 17:30:55 2020 -0500
```

vhost scsi: Add support for LUN resets.

In newer versions of virtio-scsi we just reset the timer when an a command times out, so TMFs are never sent for the cmd time out case. However, in older kernels and for the TMF inject cases, we can still get resets and we end up just failing immediately so the guest might see the device get offlined and IO errors.

For the older kernel cases, we want the same end result as the modern virtio-scsi driver where we let the lower levels fire their error handling and handle the problem. And at the upper levels we want to wait. This patch ties the LUN reset handling into the LIO TMF code which will just wait for outstanding commands to complete like we are doing in the modern virtio-scsi case.

Note: I did not handle the ABORT case to keep this simple. For ABORTs LIO just waits on the cmd like how it does for the RESET case. If an ABORT fails, the guest OS ends up escalating to LUN RESET, so in the end we get the same behavior where we wait on the outstanding cmds.

Signed-off-by: Mike Christie <michael.christie@oracle.com>

Link: <https://lore.kernel.org/r/1604986403-4931-6-git-send-email-michael.christie@oracle.com>

Signed-off-by: Michael S. Tsirkin <msst@redhat.com>

Acked-by: Stefan Hajnoczi <stefanha@redhat.com>



That HISTORY is a collection of all those commits and merges.

```
> git log --oneline --graph
| | * | | | | | | | | fdc24d722f35 MAINTAINERS: Add Jernej Škrabec as a reviewer for Allwinner SoCs support
| | / / / / / / /
| * | | | | | | | 7e47a750116d Merge tag 'stm32-dt-for-v5.10-fixes-2' of
git://git.kernel.org/pub/scm/linux/kernel/git/atorgue/stm32 into arm/fixes
| \ \ \ \ \ \ \ \ \
| * | | | | | | | f4c7fa39415d ARM: dts: stm32: Keep VDDA LD01 always on on DHC0M
| * | | | | | | | e5ace7f62695 ARM: dts: stm32: Enable thermal sensor support on stm32mp15xx-dhc0r
| * | | | | | | | 1f3d7fc279b1 ARM: dts: stm32: Define VIO regulator supply on DHC0M
| * | | | | | | | 7e5f3155dcbb ARM: dts: stm32: Fix LED5 on STM32MP1 DHC0M PDK2
| * | | | | | | | 52d9edbe6efc ARM: dts: stm32: Fix TA3-GPIO-C key on STM32MP1 DHC0M PDK2
| * | | | | | | | 0461a1ae98bc Merge tag 'amdtee-fixes-for-5.10' of git://git.linaro.org:/people/jens.wiklander/linux-tee into
arm/fixes
| \ \ \ \ \ \ \ \ \
| * | | | | | | | be353be27874 tee: amdtee: synchronize access to shm list
| * | | | | | | | ff1f855804cd tee: amdtee: fix memory leak due to reset of global shm list
* | | | | | | | 53bf2776e313 ARM: dts: exynos: revert "add input clock to CMU in Exynos4412 Odroid"
| * | | | | | | | 53bbff266880 Merge tag 'socfpga_fix_for_v5.10' of
git://git.kernel.org/pub/scm/linux/kernel/git/dinguyen/linux into arm/fixes
| \ \ \ \ \ \ \ \ \
| * | | | | | | | f126b6702e73 arm64: dts: agilex/stratix10: Fix qspi node compatible
| | | | | | | |
| * | | | | | | | e57523fa64d6 Merge tag 'samsung-fixes-5.10' of git://git.kernel.org/pub/scm/linux/kernel/git/krzk/linux into
arm/fixes
| \ \ \ \ \ \ \ \ \
| * | | | | | | | cd12e4f14f7c CREDITS: remove trailing white spaces
| * | | | | | | | d5a69b6ba186 MAINTAINERS: remove Jeongtae Park from Samsung MFC entry
| * | | | | | | | 215f06d7efc2 MAINTAINERS: move Kyungmin Park to credits
| * | | | | | | | 424f5ca7c851 MAINTAINERS: move Kamil Debski to credits
| / / / / / / / /
| * | | | | | | | 47cd1eac336c Merge tag 'imx-fixes-5.10-3' of git://git.kernel.org/pub/scm/linux/kernel/git/shawnguo/linux
into arm/fixes
| \ \ \ \ \ \ \ \ \
| | | | | | | |
| * | | | | | | | e402599e5e5e ARM: dts: imx6q-prti6q: fix PHY address
| * | | | | | | | cf5abb013219 arm64: dts imx8mn: Remove non-existent USB OTG2
| * | | | | | | | 587258edd94c arm64: dts: imx8mm-beacon-som: Fix Choppy BT audio
| * | | | | | | | 054b5d974487 arm64: dts: fsl: DPAA FMan DMA operations are coherent
| * | | | | | | | d92454287ee2 arm64: dts: fsl: fix endianness issue of rcpm
| * | | | | | | | 6efb099a1da4 arm64: dts: imx8mn-evk: fix missing PMIC's interrupt line pull-up
| * | | | | | | | 4d20fa1dac2e arm64: dts: imx8mn-ddr4-evk: fix missing PMIC's interrupt line pull-up
| * | | | | | | | 34a1c5e39b67 arm64: dts: imx8mn-var-som: fix missing PMIC's interrupt line pull-up
| * | | | | | | | ce6fc31f388d arm64: dts: imx8mm-evk: fix missing PMIC's interrupt line pull-up
| * | | | | | | | 0710e4385c9c arm64: dts: imx8mm-beacon-som: fix missing PMIC's interrupt line pull-up
| * | | | | | | | 00203737867c arm64: dts: imx8mm-var-som: fix missing PMIC's interrupt line pull-up
| * | | | | | | | a5698b3835f5 Merge tag 'hyperv-fixes-signed' of git://git.kernel.org/pub/scm/linux/kernel/git/hyperv/linux
| \ \ \ \ \ \ \ \ \
| * | | | | | | | 92e4dc8b0566 Drivers: hv: vmbus: Allow cleanup of VMBUS_CONNECT_CPU if disconnected
* | | | | | | | a08f4523243c Merge tag 'for_linus' of git://git.kernel.org/pub/scm/linux/kernel/git/mst/vhost
| \ \ \ \ \ \ \ \ \
| * | | | | | | | ef838fec17b vhost scsi: Add support for LUN resets.
| | | | | | | |
| * | | | | | | | 47a3565e8bb1 vhost scsi: fix cmd completion race
| * | | | | | | | 25b98b64e284 vhost scsi: alloc cmds per vq instead of session
| * | | | | | | | 6bcf34224ac1 vhost: add helper to check if a vq has been setup
```

```
> git log --oneline | wc -l
967467
```



4GIFs
.com

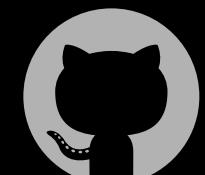
It will start to pile up



Causes of repository bloat other than just history

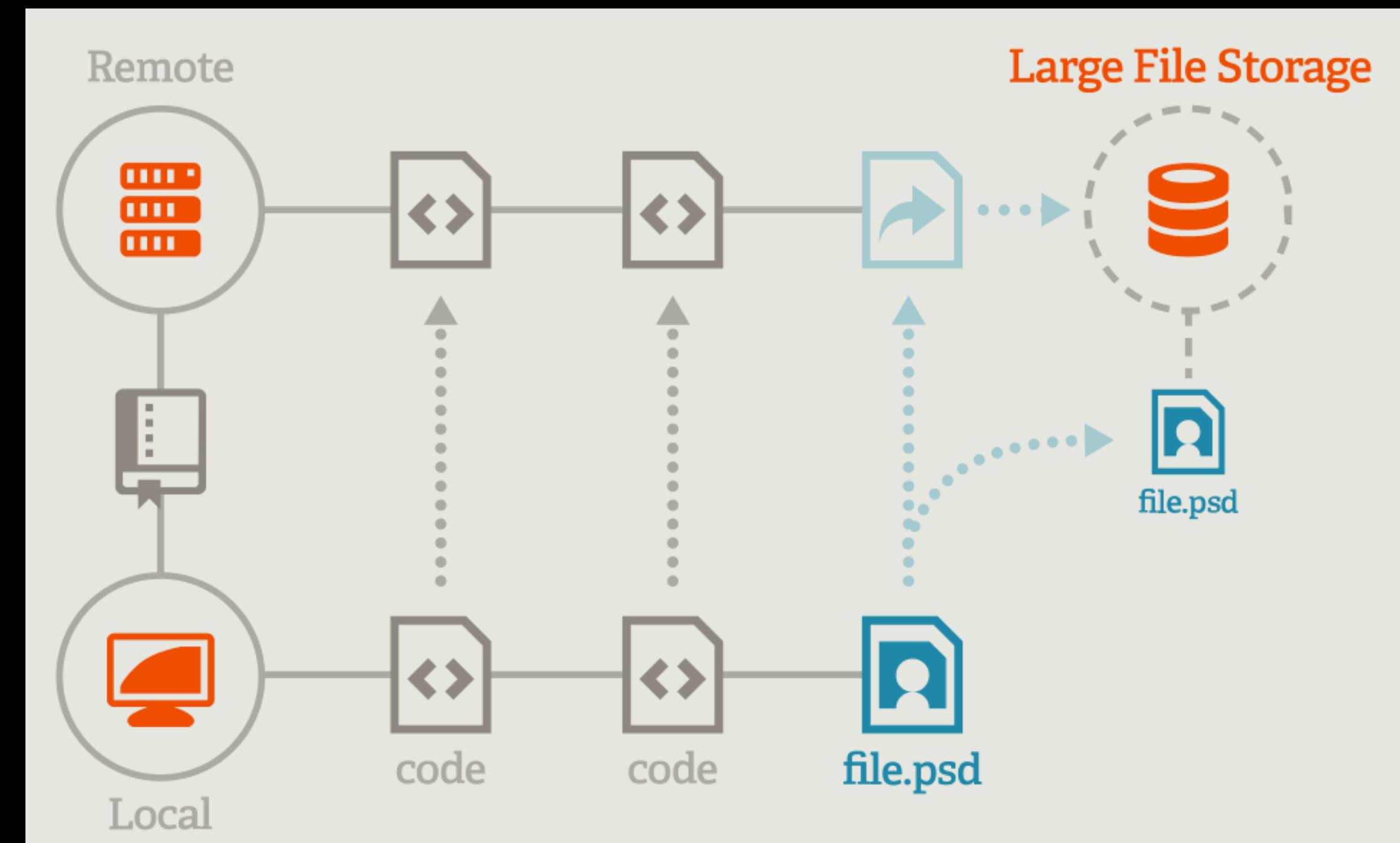
Large files can add to the size of your repository.

- Test data
- Excel/Word files
- Image/Video files
- Actual binaries
- Log Files (shouldn't be there)



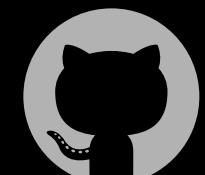
Git Large File Storage

- Git LFS handles large files by storing references to the file in the repository in the form of a pointer file
- GitHub manages this pointer file in your repository
- When you clone the repository down, GitHub uses the pointer file as a map to go and find the large file



Grafting

Enable two otherwise different **lines of development to be joined together**. It works by letting users **record fake ancestry information** for commits. This way you can make git **pretend the set of parents a commit has is different** from what was recorded when the commit was created.



Activity 1: History (10 minutes)

<https://github.com/githubuniverseworkshops/grafting-monorepos/issues/1>

1. Go to the “Activity 1: History” issue
2. Execute the commands in the order specified
3. Report your findings in comments section of the “Activity 1: History” issue
 - Be sure to obfuscate or remove any sensitive data/information before posting
4. We will discuss some of the results to get more insights

*Please reach out to us in chat or the issue
if you have questions or trouble.*

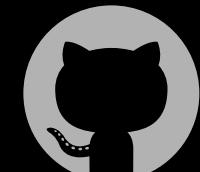
10:00

Stop



What did you learn?

- What were some of your findings?
- Was there anything that surprised you?
- What can you take away from this?



Workshop Sections

History

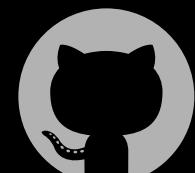
- Components of a commit
- Contributors to the bloat
- Git LFS
- 10 minute activity

Analyzing a Git Repository

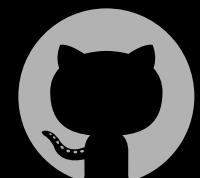
- ### Why history matters
- Effects of a large repository
 - Tools to analyze the repository
 - Customer examples
 - 20 minute activity
 - Prevention tips

Grafting A monorepo

- Grafting definition
- Cleaning the legacy repository
- Preparing the new repository
- Grafting the old and the new
- 20 minute activity
- Expected Outcomes

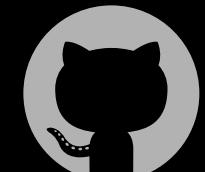


What happens in Git stays in Git



As a rule of 👍 for big repositories

If your repo is reaching 1GB you may be in trouble



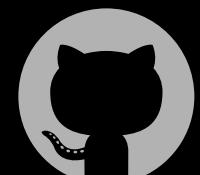
And what is the problem?

Reason

- You have a large repository history
- You are storing non-source code files

Problem

- User experience and waiting times
- Stress to the hosting machine
- CI/CD clones being slow
- In the end, it's just harder to work with it



UX example of a big repository

```
time git clone git@github.com:torvalds/linux.git

Cloning into 'linux'...
remote: Enumerating objects: 7759304, done.
remote: Total 7759304 (delta 0), reused 0 (delta 0), pack-reused 7759304
Receiving objects: 100% (7759304/7759304), 2.93 GiB | 13.79 MiB/s, done.
Resolving deltas: 100% (6437448/6437448), done.
Updating files: 100% (70638/70638), done.

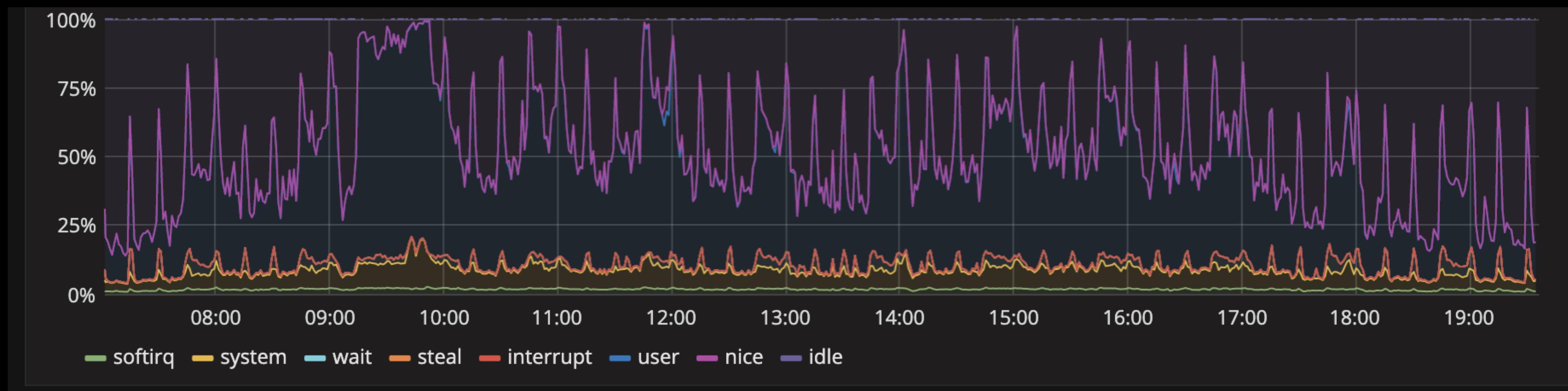
/usr/local/bin/hub clone git@github.com:torvalds/linux.git 265.11s user 65.86s system 48% cpu
11:18.77 total
```

11m 18s to clone linux

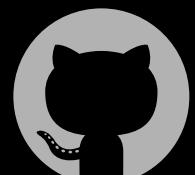


One example for some 2.5GB repos

CPU

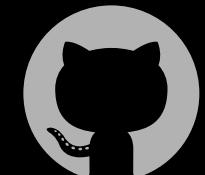


Sometimes you can't buy
bigger hardware



How to analyze the repository

- Use **git-sizer** to compute metrics and identify your repo issues [git-sizer \(https://git.io/Jki2I\)](https://git.io/Jki2I)
- From platform samples run: [platform-samples \(https://git.io/Jki2B\)](https://git.io/Jki2B)
 - **find-lfs-extensions**: lfs file extensions that should be in lfs
 - **git-find-dirs-many-files**: can find 3rd party components candidate to dependency management
 - **git-find-dirs-unwanted**: find candidates for unwanted dirs (build, temp...)
- Use **git-filter-repo** to analyze multiple parameters on your repositories with [git-filter-repo \(https://git.io/Jki2S\)](https://git.io/Jki2S)



Analyzing the repository (git-sizer)

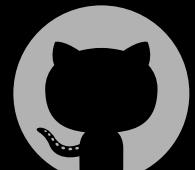
linux kernel

```
./git-sizer-1.3.0-darwin-amd64/git-sizer --verbose
Processing blobs: 2173019
Processing trees: 4613547
Processing commits: 967035
Matching commits to trees: 967035
Processing annotated tags: 671
Processing references: 676
```



Analyzing the repository (git-sizer)

Name	Value	Level of concern
<hr/>		
Overall repository size		
* Commits		
* Count	967 k	*
* Total size	752 MiB	***
* Trees		
* Count	4.61 M	***
* Total size	12.8 GiB	*****
* Total tree entries	373 M	*****
* Blobs		
* Count	2.17 M	*
* Total size	77.6 GiB	*****
* Annotated tags		
* Count	671	
* References		
* Count	676	
 Biggest objects		
* Commits		
* Maximum size	[1] 72.7 KiB	*
* Maximum parents	[2] 66	*****
* Trees		
* Maximum entries	[3] 2.19 k	**
* Blobs		
* Maximum size	[4] 13.5 MiB	*



Analyzing the repository (git-sizer)

Name	Value	Level of concern
<hr/>		
History structure		
* Maximum history depth	162 k	
* Maximum tag depth [5]	1	
<hr/>		
Biggest checkouts		
* Number of directories [6]	4.74 k	**
* Maximum path depth [7]	13	*
* Maximum path length [8]	134 B	*
* Number of files [9]	70.7 k	*
* Total size of files [9]	918 MiB	
* Number of symlinks [10]	40	
* Number of submodules	0	
<hr/>		
[1] 91cc53b0c78596a73fa708cceh7313e7168hh146		
[2] 2cde51fb0f310c8a2c5f977e665c0ac3945b46d		
[3] 648d5c2e006/2dfe3c21/d25b8c/3802/44t3d3b (refs/heads/master:arch/arm/boot/dts)		
[4] 29af5167cd0057fcfbfab150378322008d3d2667 (refs/heads/master:drivers/gpu/drm/amd/include/asic_reg/nbio/nbio_6_1_sh_mask.h)		
[5] 5dc01c595e6c6ec9ccda4f6f69c131c0dd945f8c (refs/tags/v2.6.11)		
[6] 47bb244d15374c342ffe8edelfcf15b46f08827f (8e4c309f9f33b76c09ada02b796ef87918eee494^{tree})		
[7] 78a269635e76ed927e17d7883f2d90313570fdb (dae09011115133666e47c35673c0564b0a702db7^{tree})		
[8] b0da5ce619daec8138cf92dfcf00e7a51ce856a9 (d8763340d2cb6262fb86424315a1f92cab0e23c^{tree})		
[9] 5bf53b0c610a5abf0cd874d0687f4e731009fda4 (9c75b68b91ff010d8d4c703b93954f605e2ef516^{tree})		
[10] f29a5ea76884ac37e1197bef1941f62fda3f7b99 (f5308d1b83eba20e69df5e0926ba7257c8dd9074^{tree})		



Analyzing the repository (git-sizer)

linux kernel

```
git show 2cde51fb0f310c8a2c5f977e665c0ac3945b46d | less
commit 2cde51fb0f310c8a2c5f977e665c0ac3945b46d
Merge: 7471c5c9f58e c097d5fdf3b5 74c375cb85d7 04c3a852f51f 5095f55d7cc3 4f534777c130
2f54d2a1cf7e 56d37d85438d 192043cf6089 f467a0f513ad bbe580302d33 3990c516de66
d754fa9ad18d 516ea4b58433 69ae8489076f 25c1a63f43ca f52c91921553 111bd7b18e13
aafab85e71a75 dd407a324323 71467e46414d 0f7f3d1f17c2 8778ac6be25a 0406a40a095c
308a0f3f24db 2650bc4f6d0c 8cb7a36eb3a8 323702b4e06d ef749400434c 3cec159cfb3f
72aa62bed3ea 328089a47112 11db0da831b1 e1771bcf99b0 f60e5473e678 a010ff628c09
5e8154332f48 58381da68774 626bcacb89f9 38136bde7691 06b2bd23057f 8c5178fca4ce
8e6ad35a31e7 008ef947d0c5 f58c4fc4a3bf 2309d6757900 5c1537163ce7 b65ab73e5d62
26090a834b49 9ea6fbc66d15 2c4864334c4d 1769267bb013 f3f9a60f7947 f25cf3496982
3f3002692ce8 fbbf7fea8e80 c3e8494c001c e40e0b5da87b 50c969732043 63587116811b
0112b62b12e1 a0a05916cf67 b888edbc68fb d44008b35858 9a199b8e9933 784cbf8ab464
Author: Mark Brown <broonie@linaro.org>
Date: Thu Jan 2 13:01:55 2014 +0000
```

```
git show d5530d82efc8 | less
commit d5530d82efc8631beff20480b1168b1c44294fe1
Merge: f4b936f5d6fd b4c00e797663
Author: Linus Torvalds <torvalds@linux-foundation.org>
```



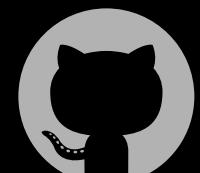
Analyzing the repository (git-find-lfs-extensions)

linux kernel

./grafting-monorepos/scripts/git-find-lfs-extensions

Type Max	Extension	LShare	LCount	Count	Size	Min
--	--	--	--	--	--	--
all	*	0 %	73	70569	917	0 13
text	h	0 %	62	21370	304	0 13
text	c	0 %	7	29461	523	0 0
text	json	1 %	2	332	7	0 0
text	s	0 %	1	1290	8	0 0
text w/o ext	maintainers	50 %	1	2	0	0 0

Add to .gitattributes:



@droidpl & @selkins13 / How to keep Git monorepos manageable
Universe 2020

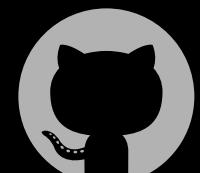
Analyzing the repository (git-find-lfs-extensions)

Other repository

Type	Extension	LShare	LCount	Count	Size	Min	Max
all	*	0 %	30	28943	288	0	90
binary	gz	100 %	2	2	143	52	90
binary	png	12 %	9	75	18	0	1
binary	exe	40 %	2	5	4	0	2
binary	key	100 %	1	1	4	4	4
binary	zip	100 %	1	1	1	1	1
text	js	0 %	6	15557	61	0	2
text	ts	0 %	4	4125	10	0	1
text	map	0 %	3	2911	15	0	1
text	html	5 %	2	41	1	0	0

Add to .gitattributes:

```
*.[Ee][Xx][Ee] filter=lfs diff=lfs merge=lfs -text
*.[Gg][Zz] filter=lfs diff=lfs merge=lfs -text
*.[Kk][Ee][Yy] filter=lfs diff=lfs merge=lfs -text
*.[Pp][Nn][Gg] filter=lfs diff=lfs merge=lfs -text
*.[Zz][Ii][Pp] filter=lfs diff=lfs merge=lfs -text
```



Analyzing the repository (git-find-dirs-many-files)

linux kernel

```
./grafting-monorepos/scripts/git-find-dirs-many-files | head -n 20
```

```
70602
28698 ./drivers
15946 ./arch
7486 ./Documentation
5416 ./include
5085 ./drivers/gpu
4996 ./drivers/gpu/drm
4922 ./drivers/net
4569 ./tools
4492 ./arch/arm
4071 ./Documentation/devicetree
4064 ./Documentation/devicetree/bindings
2453 ./include/linux
2438 ./drivers/media
2366 ./drivers/net/ethernet
2279 ./sound
2214 ./arch/arm/boot
2204 ./drivers/staging
2196 ./tools/testing
2184 ./arch/arm/boot/dts
```



Analyzing the repository (git-find-dirs-many-files)

Other repository

```
..../grafting-monorepos/scripts/git-find-dirs-many-files | head -n 20

29909 .
19818 ./bots
19784 ./bots/probot
19728 ./bots/probot/onboard-new-members
19676 ./bots/probot/onboard-new-members/node_modules
8758 ./scripts
...
...
```



Analyzing the repository (git-find-dirs-unwanted)

linux kernel

```
./grafting-monorepos/scripts/git-find-dirs-unwanted | head -n 15

4569 tools/
4064 Documentation/devicetree/bindings/
2088 tools/testing/selftests/
1385 arch/x86/
1268 tools/perf/
632 tools/testing/selftests/bpf/
448 lib/
368 arch/x86/include/asm/
327 tools/perf/util/
326 Documentation/devicetree/bindings/sound/
325 tools/testing/selftests/bpf/progs/
313 tools/perf/pmu-events/
283 include/dt-bindings/clock/
269 Documentation/devicetree/bindings/clock/
240 arch/x86/kernel/
```

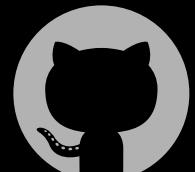


Analyzing the repository (git-find-dirs-unwanted)

Other repository

```
./grafting-monorepos/scripts/git-find-dirs-unwanted | head -n 15

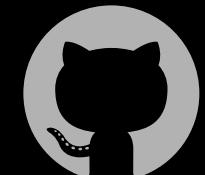
16 libs/npm/octotest/
12 scripts/utils/
10 scripts/utils/labview-git/
10 scripts/github_admin/org-admin-notification-utility/
7 scripts/jenkins scripts/shared-libraries/topic-repo-job/
5 scripts/utils/labview-git/bin/
2 docs/github/onboarding-templates/
2 .github/ISSUE_TEMPLATE/
1 scripts/jenkins scripts/shared-libraries/topic-repo-job/vars/
1 scripts/backup/
1 docs/monitoring/ghes-elk/elk-stack/outputs/
deleted templates/
```



Analyzing the repository (git-filter-repo --analyze)

linux kernel

```
./git-filter-repo/git-filter-repo --analyze
Processed 7754272 blob sizes
Processed 967008 commits warning: inexact rename detection was skipped due to too many files.
Processed 967035 commits
Writing reports to .git/filter-repo/analysis...done.
```



Analyzing the repository (git-filter-repo --analyze)

Results in the workshop repo: [/linux-filter-repo](#)

Generates files under .git/filter-repo/analysis:

- **blob-shas-and-paths.txt**: look at specific blobs
- **directories-all-sizes.txt**: find directories that shouldn't be in the repo
- **directories-deleted-sizes.txt**: deleted things that are still taking space in the repo
- **extensions-all-sizes.txt**: similar to the previous script, meant for LFS findings
- **extensions-deleted-sizes.txt**: same but for files in the history (LFS migrate)
- **path-all-sizes.txt**: files that have been changed many many times (finds logs or automatic committed files)
- **path-deleted-sizes.txt**: deleted files that were committed a lot
- **renames.txt**: renamed files and patterns that are not common



Customer examples

- Customer A - ~1GB emoji packs in the repo that were not used anymore (ownership?)
- Customer B - language files causing packing issues while calculating compression deltas
- Customer C - autogenerated code files to keep the state of the platform
- Customer D - testing database stored in the repository without LFS
- Customer E - Storing open source libraries, binaries, build tools or files that should be ignored (gitignore.io)



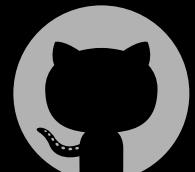
Activity 2: Analysis (20 minutes)

<https://github.com/githubuniverseworkshops/grafting-monorepos/issues/2>

1. Go to the “Activity 2: Analysis” issue
2. Execute the commands in the order specified
3. Report your findings in comments section of the “Activity 2: Analysis” issue
 - Make sure you obfuscate or remove any sensitive data/information before posting
4. **Post your results** in the issue

*Please reach out to us in chat or the issue
if you have questions or trouble.*

20:00



Preventing these issues

- .gitattributes for Git LFS
- Git submodules
- Pull request reviews
- Checks for the size of the files in your CI/CD
<https://github.com/actionsdesk/lfs-warning>
- Repo size < 1GB
- Restrict object size (only server right now, only through support request on cloud)
- Pre-commit hooks (requires some training, local only)

But if your problem goes out of hand... **Grafting**



Agenda

History

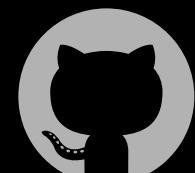
- Components of a commit
- Contributors to the bloat
- Git LFS
- 10 minute activity

Analyzing a Git Repository

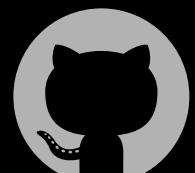
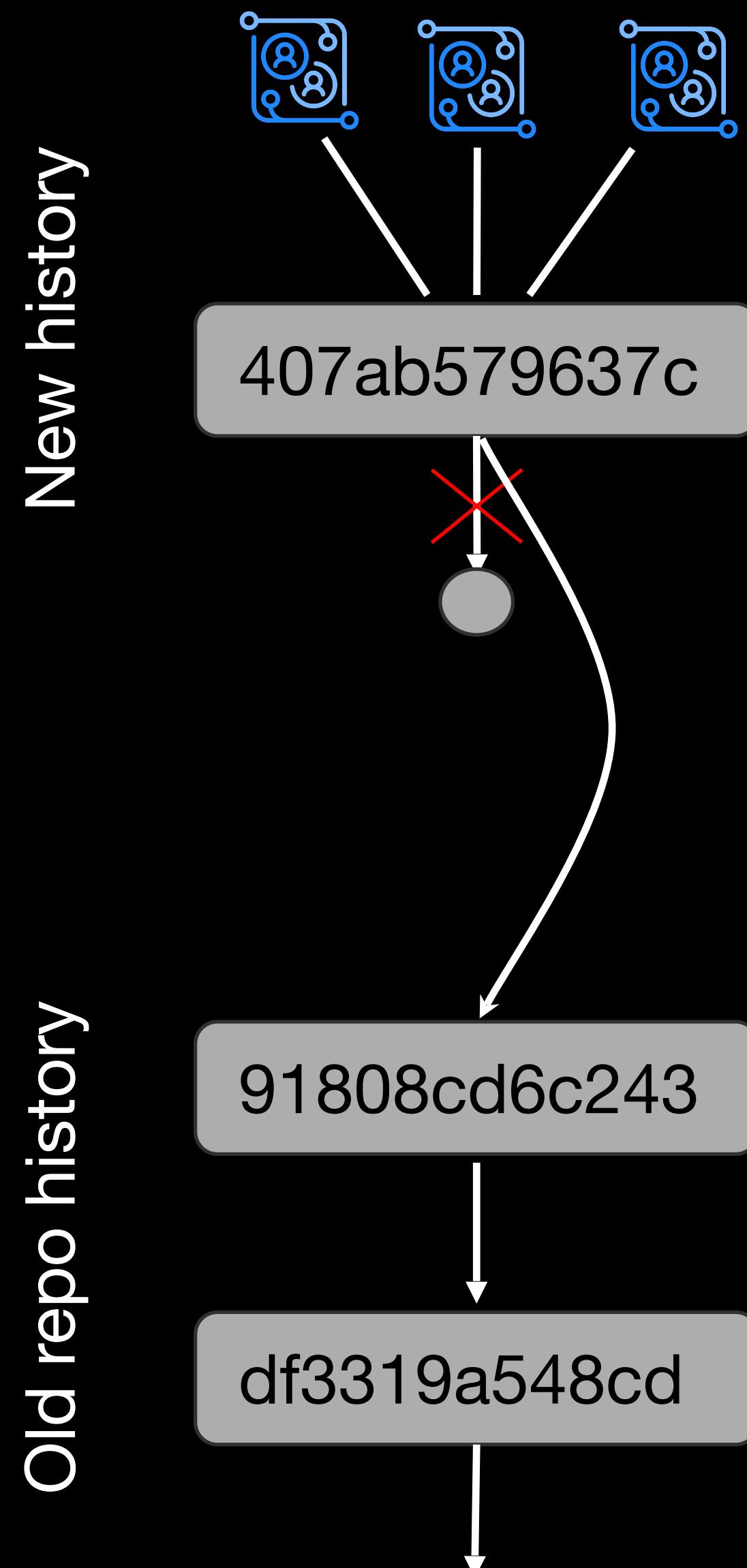
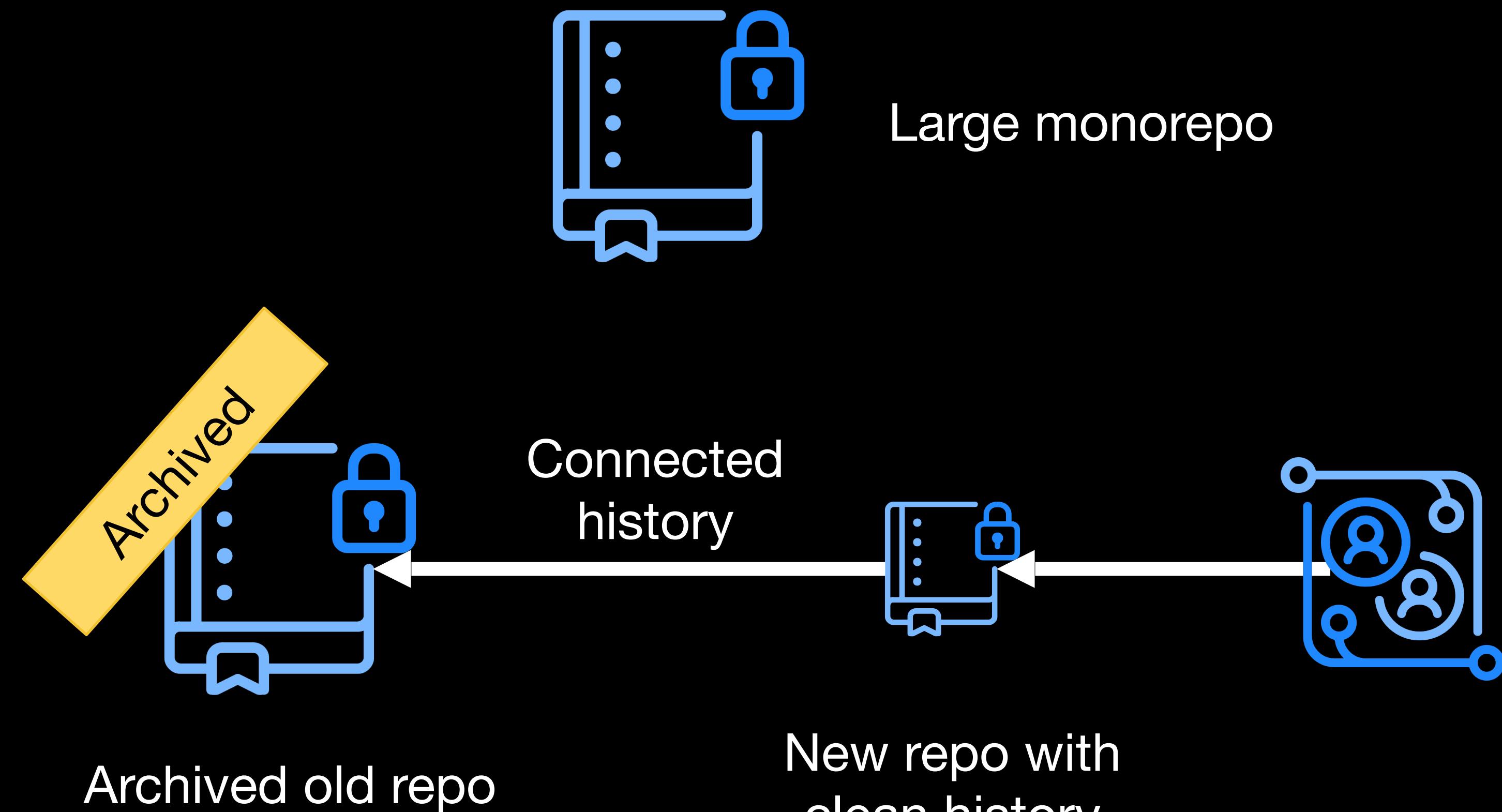
- Effects of a large repository
- Tools to analyze the repository
- Customer examples
- 20 minute activity
- Prevention tips

Grafting A monorepo

- Grafting definition
- Cleaning the legacy repository
- Preparing the new repository
- Grafting the old and the new
- 20 minute activity
- Expected Outcomes



What is grafting?

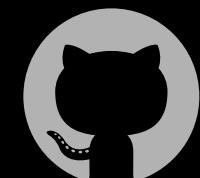


Step 1: Clean your current repo

Grafting the repository without cleaning it is like
taking a dirty rug and putting it in your clean house.

Make sure you:

1. Remove **unneeded** or unwanted files (emojis, logs...)
2. Add big files to **LFS**
3. Add **dependencies** to dependency management
4. Remove **independent components** or migrate to Git submodules
5. Remove any **binaries** or files that should be ignored with .gitignore
6. Review **automations** that write automatically to the repository



Step 2: Create a new repository

Create a new repository on GitHub to store the cleaned up code. Optionally, you can rename the old repo now or later to keep consistent names.

Owner * Repository name *

 githubuniverseworkshops / grafting-repo ✓

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-bassoon](#)?

Description (optional)

 Public
Anyone on the internet can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

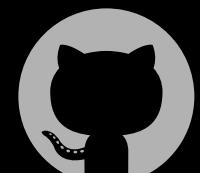
Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository



Step 3: Prepare for the grafting

We need to start from scratch and avoid leaving open work. You should:

- Communicate with your team regarding the cutoff for any new work or changes to the repository
- **Merge all** the work in progress as pull requests are not moved from one repository to the other
- If **not merged, it will not be moved**
- Call **GitHub Professional Services** if you need help



Step 4: Delete the history

linux kernel

```
# Archive your old repository
# Check all the repository is clean locally
git status

# Delete the git folder that contains git objects
rm -rf .git

# Initialize a new history
git init

# Set the new repository
git remote add origin git@github.com:githubuniverseworkshops/grafting-repo.git
```



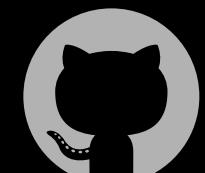
Step 5: Write the beginning of the end

linux kernel

```
# Add all files to the stage
git add --all

# Add changes to history
git commit -m "Previous repo can be found on
https://github.com/torvalds/linux/tree/xxxxxxxxxxxxxxxxxxxxxx"

# Submit your changes to upstream
git push --set-upstream origin main
```



@droidpl & @selkins13 / How to keep Git monorepos manageable
Universe 2020

(Optional) Working in the grafted repo

linux kernel

```
# Fetch the old history
git fetch git@github.com:torvalds/linux.git

# See you only have one commit in it
git log --oneline

# See the commits we are replacing
git rev-parse --short HEAD
git rev-parse --short FETCH_HEAD

# Perform the grafting operation replacing HEAD with FETCH_HEAD
git replace HEAD FETCH_HEAD
```



@droidpl & @selkins13 / How to keep Git monorepos manageable
Universe 2020

Verify you can work in the grafted repo

linux kernel

```
# Check that the new commit goes to the right repo
# Modify a file
echo "Test" > test.txt
git add --all
git commit -m "Adding a test commit"

# Check that you can navigate the history
git log --oneline | head -n 10

# Push the change and see the number of commits is still 2
git push
```

⚠ Make sure you don't push the old code to the new repo, or the problem would be replicated
You can setup a pre-receive hook to prevent from that



Verify you can work in the grafted repo

Grafted linux kernel

```
git log --oneline | head -n 10

3fd90e0a58ed Linux 5.10-rc5
d5530d82efc8 Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/hid/hid
f4b936f5d6fd Merge tag 'sched-urgent-2020-11-22' of
git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
48da33058975 Merge tag 'perf-urgent-2020-11-22' of
git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
855cf1ee4726 Merge tag 'locking-urgent-2020-11-22' of
git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
68d3fa235fd8 Merge tag 'efi-urgent-for-v5.10-rc3' of
git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
7d53be55c9d7 Merge tag 'x86_urgent_for_v5.10-rc5' of
git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip
4a51c60a1115 Merge branch 'akpm' (patches from Andrew)
d27637ece80f Merge tag 'staging-5.10-rc5' of
git://git.kernel.org/pub/scm/linux/kernel/git/gregkh/staging
de7580357025 Merge tag 'tty-5.10-rc5' of
git://git.kernel.org/pub/scm/linux/kernel/git/gregkh/tty
```



Verify you can work in the grafted repo

Grafted linux kernel

master

Commits on Nov 19, 2020

Git replace really did a change?
droidpl committed 4 days ago

Verified 3fd90e0 <>

Previous repository can be found on <https://github.com/torvalds/linux>
droidpl committed 4 days ago

Verified 06f982d <>

Newer Older



Fun fact on the Linux Kernel

<https://archive.org/details/git-history-of-linux>

Files for git-history-of-linux

Name	Last modified	Size
⬆ Go to parent directory		
full-history-linux.git.tar (View Contents)	01-Aug-2011 02:50	594.4M
full-history-linux.git.tar.txt	01-Aug-2011 02:50	4.1K
git-history-of-linux.json	15-Sep-2011 20:54	899.0B
git-history-of-linux_archive.torrent	18-Aug-2018 10:07	25.4K
git-history-of-linux_files.xml	18-Aug-2018 10:07	2.9K
git-history-of-linux_files_all.torrent	15-Sep-2011 20:55	6.4K
git-history-of-linux_meta.xml	10-May-2016 01:43	2.2K
git-history-of-linux_reviews.xml	18-Aug-2018 10:06	880.0B
make-full-linux-history.tgz	01-Aug-2011 02:50	210.1K
make-full-linux-history.tgz.txt	01-Aug-2011 02:50	546.0B



Activity 3: Graft a repository (20 minutes)

<https://github.com/githubuniverseworkshops/grafiting-monorepos/issues/4>

1. Execute the commands explained above to graft your monorepo
2. Run the analysis tools again to see the improvements

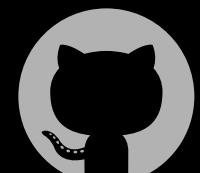
*Please reach out to us in chat or the issue
if you have questions or trouble.*

20:00



git-sizer

	New repo	Old repo
Processing blobs	68974 (- 97 %)	2173019
Processing trees	4703 (- 99 %)	4613547
Processing commits	2 (- ~100 %)	967035
Matching commits to trees	2 (- ~100 %)	967035
Processing annotated tags	0	671
Processing references	3 (- ~100 %)	676



git-sizer

	New repo	Old repo
Commits size	2.18 KiB (- ~100 %)	752 MiB
Tree size	2.81 MiB (- ~100 %)	12.8 GiB
Blob size	908 MiB (- 98 %)	77.6 GiB
Maximum history depth	2 (- ~100 %)	162 k
Script runtime	0.356 secs (- ~100 %)	3:22.24 secs



git-find-lfs-extensions

Still no relevant changes - the Linux repository is very well maintained and contributors make sure nothing that shouldn't be in Git goes into it

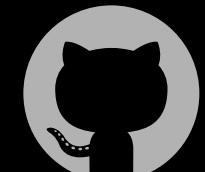
git-find-many-dirs

No relevant changes - All the changes we see are due to the difference between the date we checked out the repository and today (Linux is a very active repository)



git-find-dirs-unwanted

Still no relevant changes - Many of what are considered bad folders, like x86 or tools, are present in the Linux code, but for good reasons. A cleanup was not performed so the results remain the same.



Git-filter-repo --analyze

	New repo	Old repo
Blobs analyzed	73679 (- 95 %)	7754272
Commits analyzed	2 (- ~100 %)	967035
Renames	0	26902
Paths deleted	0	37925
Blobs file size	7.07 mb (- 97 %)	209.48 Mb
Script runtime	7.865 secs (- 99 %)	18:11.0 secs



Expected Outcomes

1. Clear improvement on the **operations speed** for Git interactions
2. **Faster clone** speed (from 11m to 31 secs -96%)
3. **Faster CI** and employee **productivity**
4. Improved **developer experience**
5. **Fewer server resources** needed
6. **Legacy git history** locally, so blaming is still possible 😈
7. Scalable process for **future code generations**

