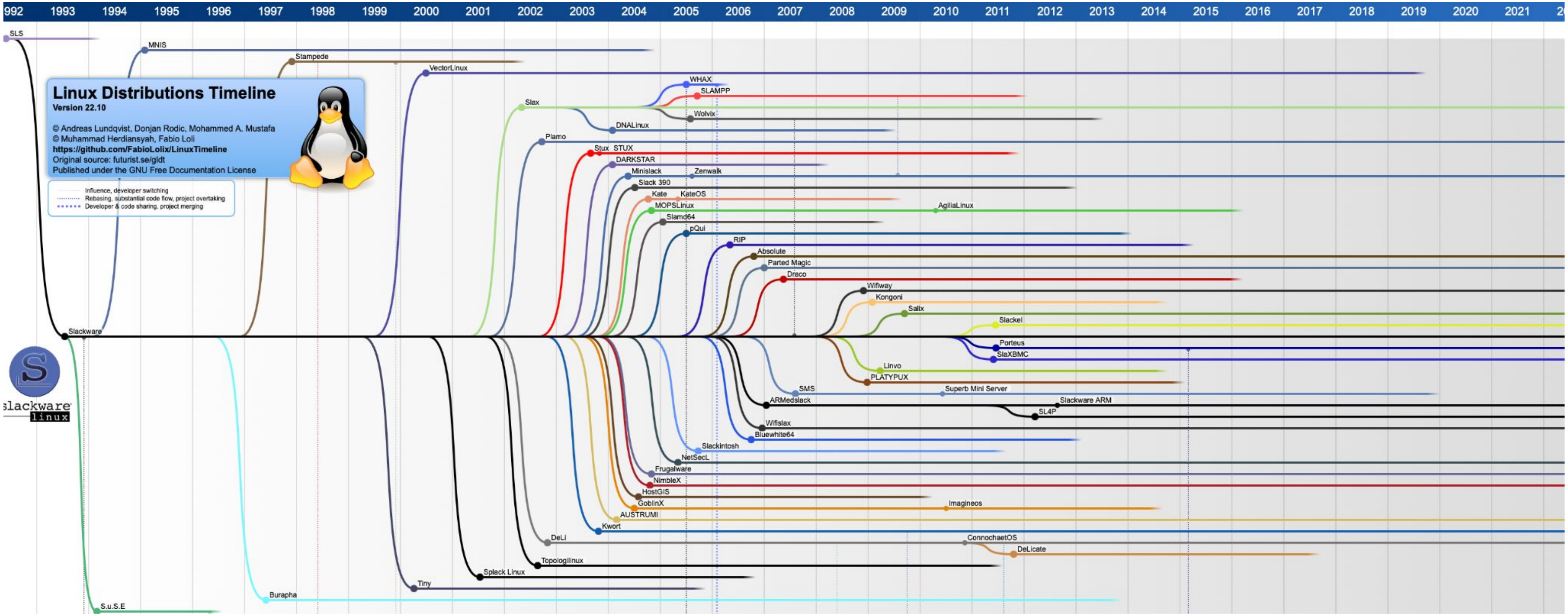
Bridging the gap in back-porting Linux kernel patches

- Akshith G

The Linux Distro Map



https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

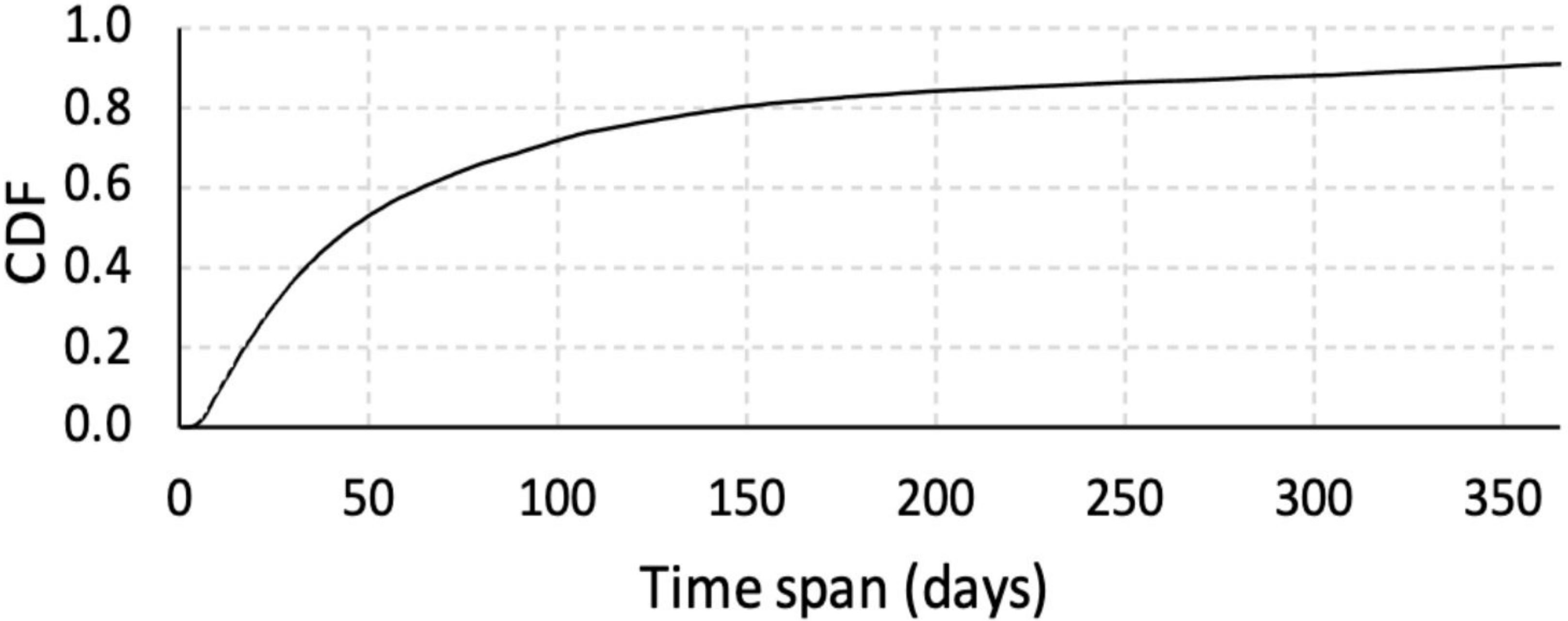
Mainline vs Downstream

1. Mainline: All developments happen here.

2. Distribution: But do I keep up with the new Kernel? ☹️

3. Security, Performance, New features.

Patch Porting Time (Cumulative)



Two main tasks

1. What patches should I port?

2. Is the patch relevant to my kernel?

Two main tasks

1. What patches should I port?

2. Is the patch relevant to my kernel?

Is the patch relevant to my kernel?

The gap is largely due to lack of understanding of the context of the patch

a. File names do not match

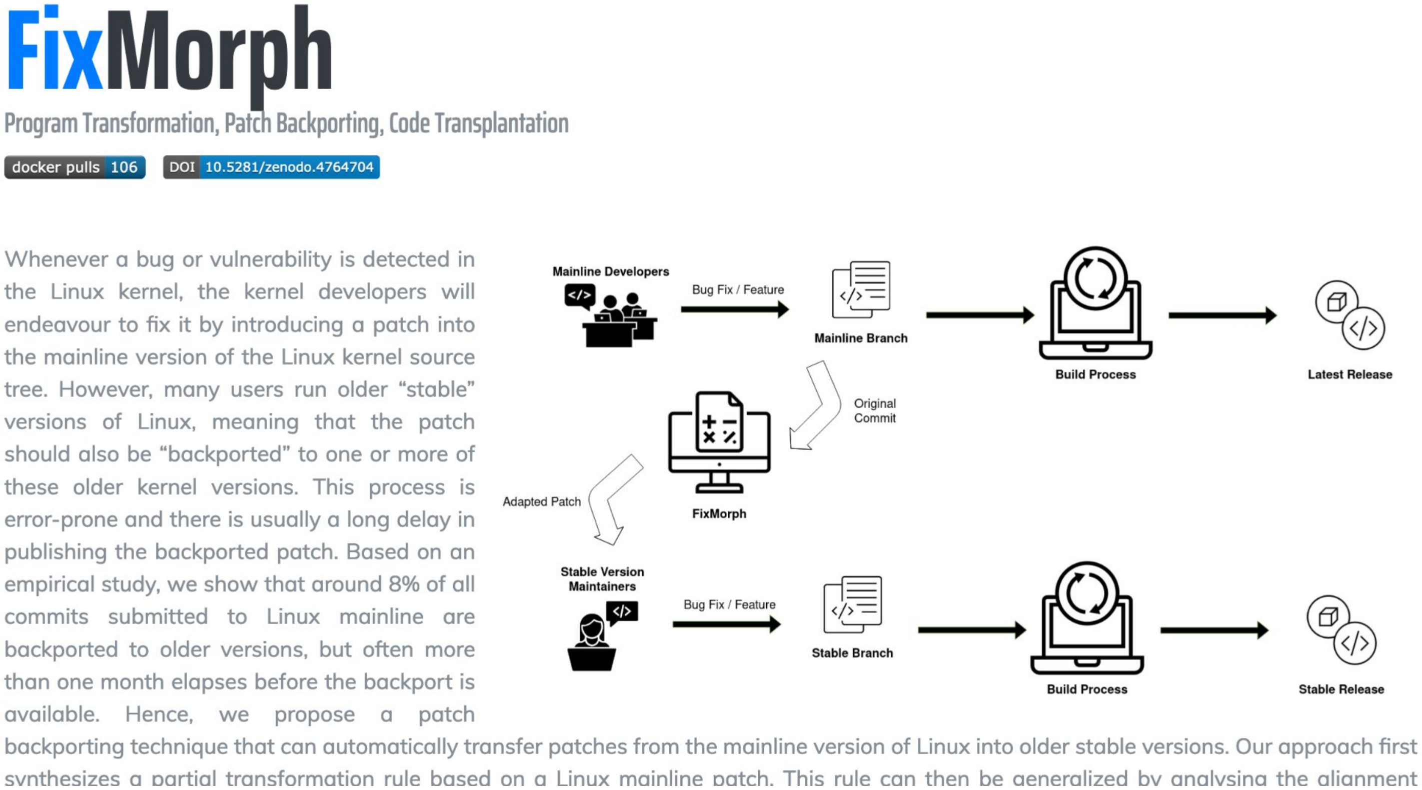
b. Hard to locate the context of the patch in earlier versions.

Solution?

LLMs have shown to be effective in understanding large contexts:

1. Code completion
2. Code summarization
3. Code search

Standing on the shoulder of giants



FixMorph

Program Transformation, Patch Backporting, Code Transplantation

docker pulls 106 DOI 10.5281/zenodo.4764704

Whenever a bug or vulnerability is detected in the Linux kernel, the kernel developers will endeavour to fix it by introducing a patch into the mainline version of the Linux kernel source tree. However, many users run older "stable" versions of Linux, meaning that the patch should also be "backported" to one or more of these older kernel versions. This process is error-prone and there is usually a long delay in publishing the backported patch. Based on an empirical study, we show that around 8% of all commits submitted to Linux mainline are backported to older versions, but often more than one month elapses before the backport is available. Hence, we propose a patch backporting technique that can automatically transfer patches from the mainline version of Linux into older stable versions. Our approach first synthesizes a partial transformation rule based on a Linux mainline patch. This rule can then be generalized by analysing the alignment

Research Questions

RQ1. Can LLMs be used to understand the context of the patch?

RQ2. Can LLMs be used to automate patch transplantation?

Datasets

1. Patches from the mainline kernel and the corresponding downstream kernel patches.

2. Dataset of CVEs and the patches that fix them.

Evaluation (Standard ML metrics)

1. Success in understanding the context of the patch. (File locations)
2. Success in porting the patch.

Precision

Recall

F1 score