

Wrestling a Giant C++ desktop app into your browser and making LibreOffice sing

Michael Meeks
CEO

michael.meeks@collabora.com



Collabora
Online



“Stand at the crossroads and look; ask for the ancient paths, ask where the good way is, and walk in it, and you will find rest for your souls...” - Jeremiah 6:16





FLOSS is uniquely fascinating !

You can see everything ...

- The mass of code underneath is quite amazing
- You can read & understand & become an expert easily.

Wrestling this code-base & platform for 25+ years

- Stories – up & down the stack.

OpenOffice.org → LibreOffice & then Collabora Online (COOL)

- A browser-based tool ...

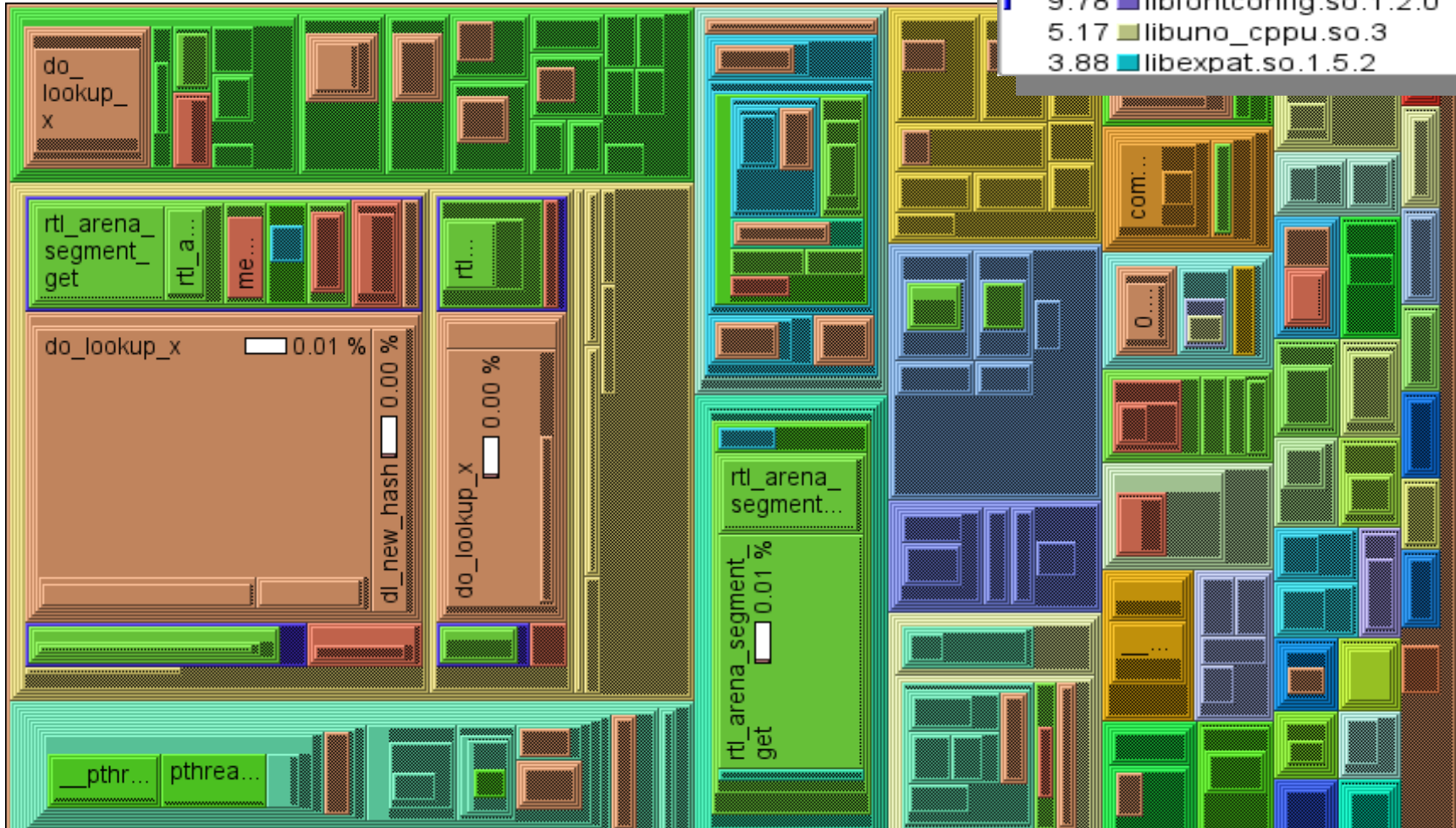
Gritty technical detail...

Why is OO.o slow to start ? #1

the easy to measure bit: CPU



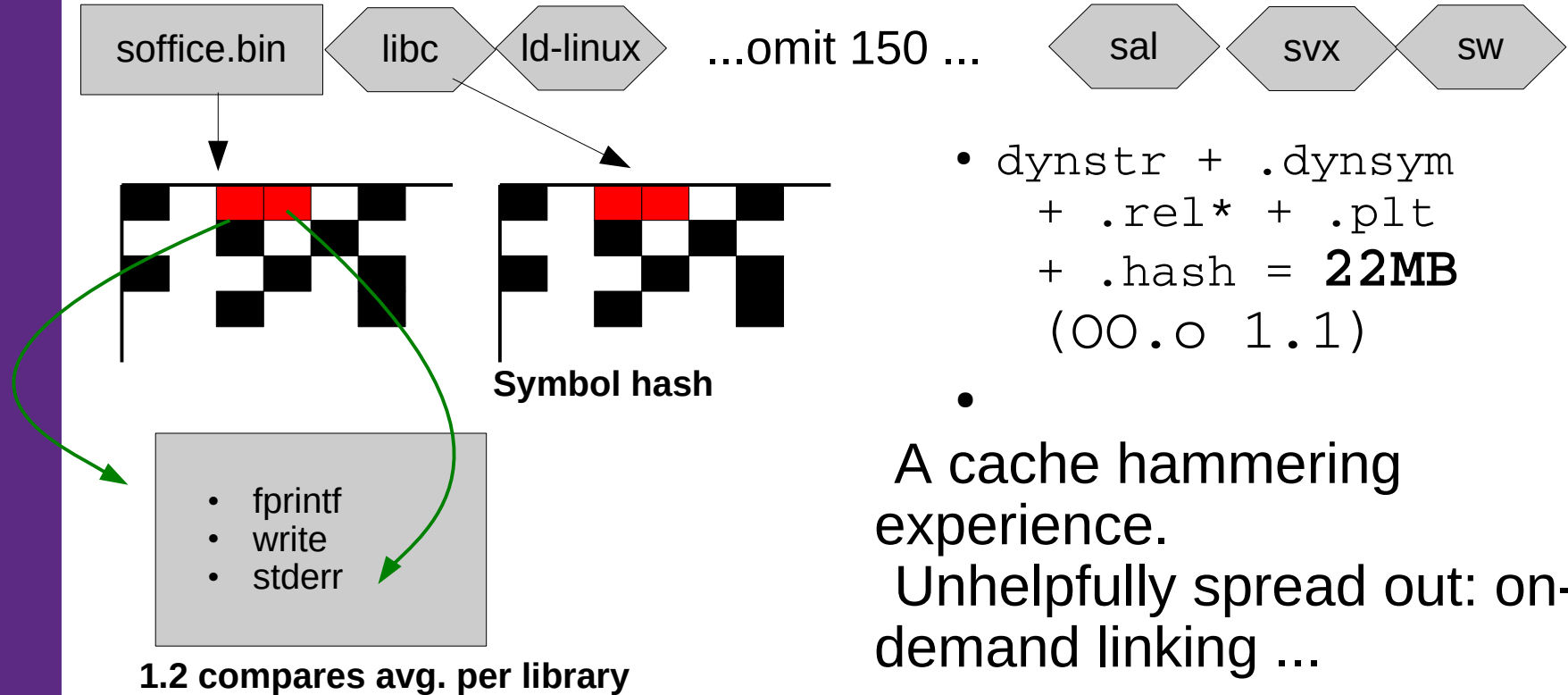
Valgrind ! ...





ELF run-time linking: ld-linux.so.2

For every dynamic symbol we lookup:





Towards a fix:

Prototyped several fixes:

- -Bdirect → at compile-time remember the library to look up symbols in
 - Except for 'Vague' symbols etc. etc. remove a 100x factor.
- Various improved symbol hash storage schemes
- Avoiding lots of named-relocations ... → not finalized.

Large apps:

- Default is not to have the symbol – so optimize for that.
- Much sparser & smaller 'bloom filter' in front of symbol lookup
 - A big bit-field mask – for “do we have an entry”
 - Only if so – go on to touch the symbol.

.gnu.hash

- New linking standard

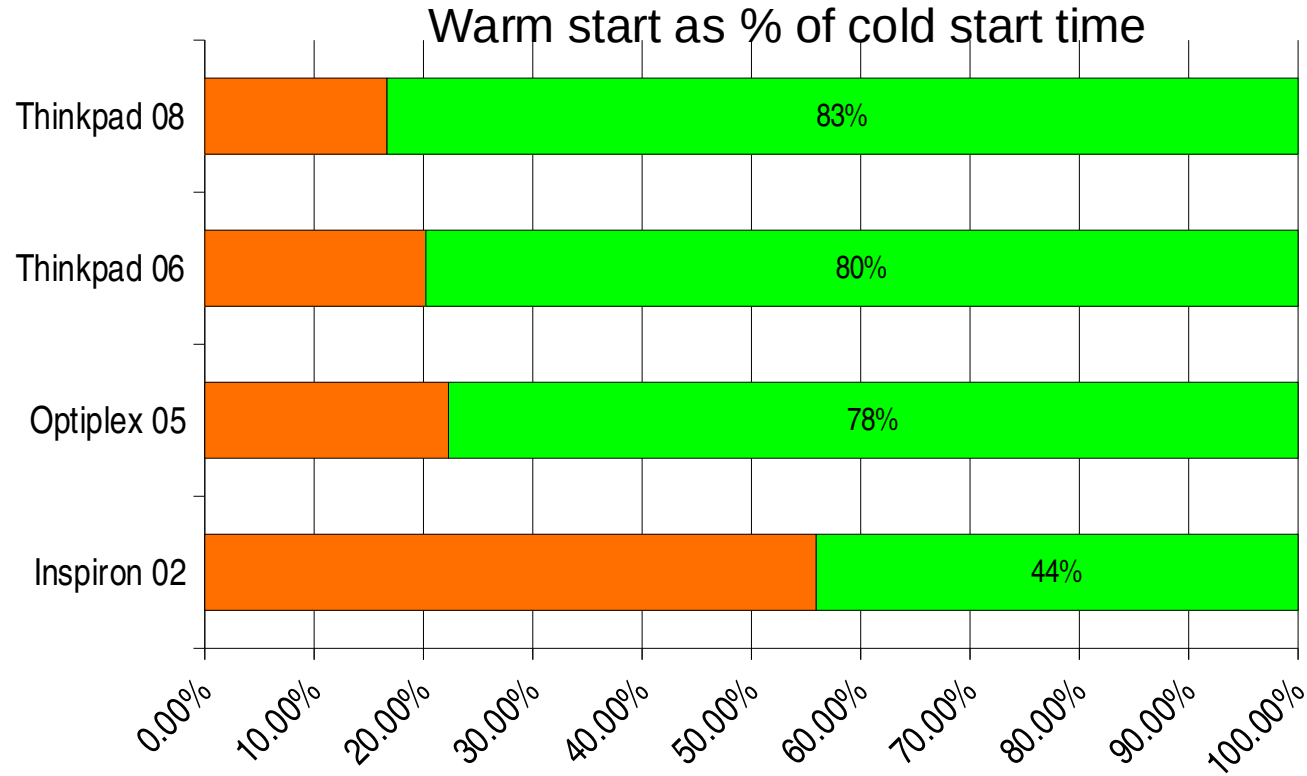
Why is OO.o slow to start ? #2

oh dear, no good tools exist: I/O



I/O → cold vs. warm start:

eg. OpenOffice.org startup really sucks: Amdahl asks Why ?





Happy page-cache users chip-in:

Google: "openoffice startup graphics cache"

~about 1,830,000 hits ...

Tom Adelstein - "Linux Journal" (or earlier?)

re-printed in Linux Format, the Inquirerer.net, PC Pro forums...

"Under Ubuntu, I found that OO Writer opened in three seconds, and in Fedora it opened in less than six seconds. Previously, it took 30 and 26 seconds"

How is it achieved ?!

*Increase the **graphics cache ram**, I upped mine to 64 for "Use for OO.org" and 5 for "memory per object"*

Loading times will *increase*[sic.] dramatically. (20 seconds to 7 seconds on one computer I've used)"



More satisfied page-cache users ...

Is it just Gentoo syndrome run wild ? ...

“very effective change. Having run knoppix **from the cd drive** for over a year on a few computers, this increase in memory setting cuts anywhere from 30 seconds to more than several minutes”

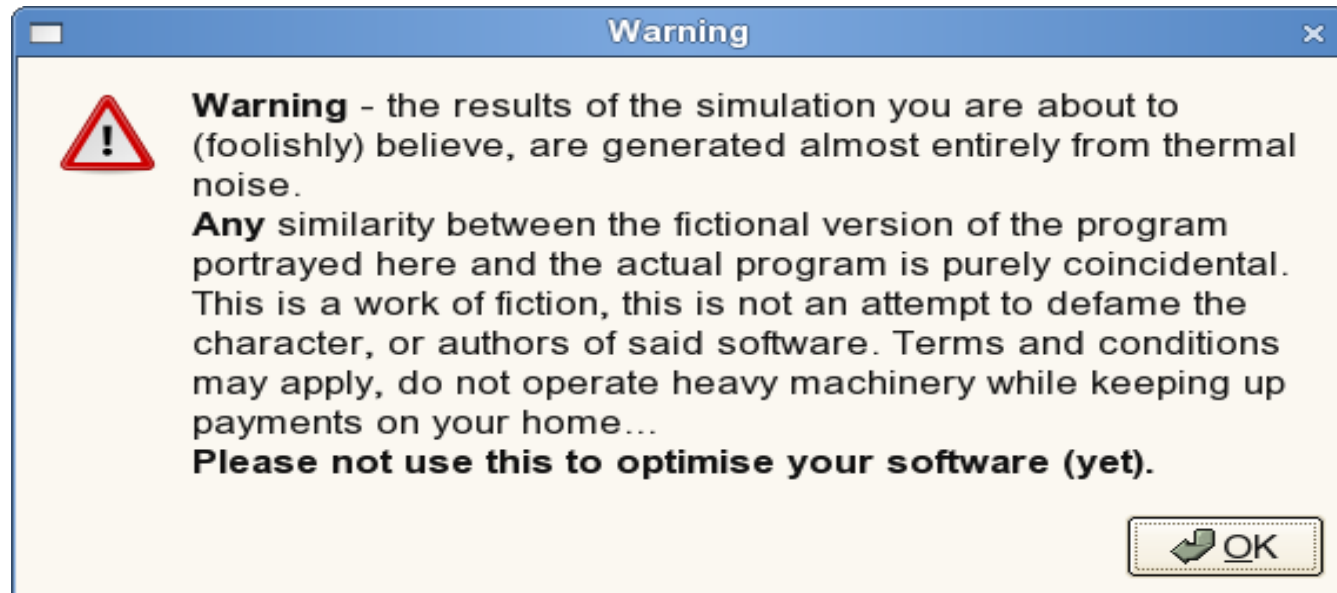
“Ubuntu Forums HOWTO: Speed up OO.o start up”

Not bad My **speed about doubled**, and I've only got 128 MB of RAM.
- Hikaru79

I did see a speed up of about **50%**, or 3 seconds to startup on my PC. -
wallijonn

... this has sped up OOo loading tremendously ... it now loads up in **4 seconds** instead of the old **8 seconds**. ... [this should be] "Howto of the Year". - theepoch

Never profile without measuring: so write a (~throw-away) measuring tool:





logrind – a valgrind tool + GUI

Now Obsolete / Under-used but fun:

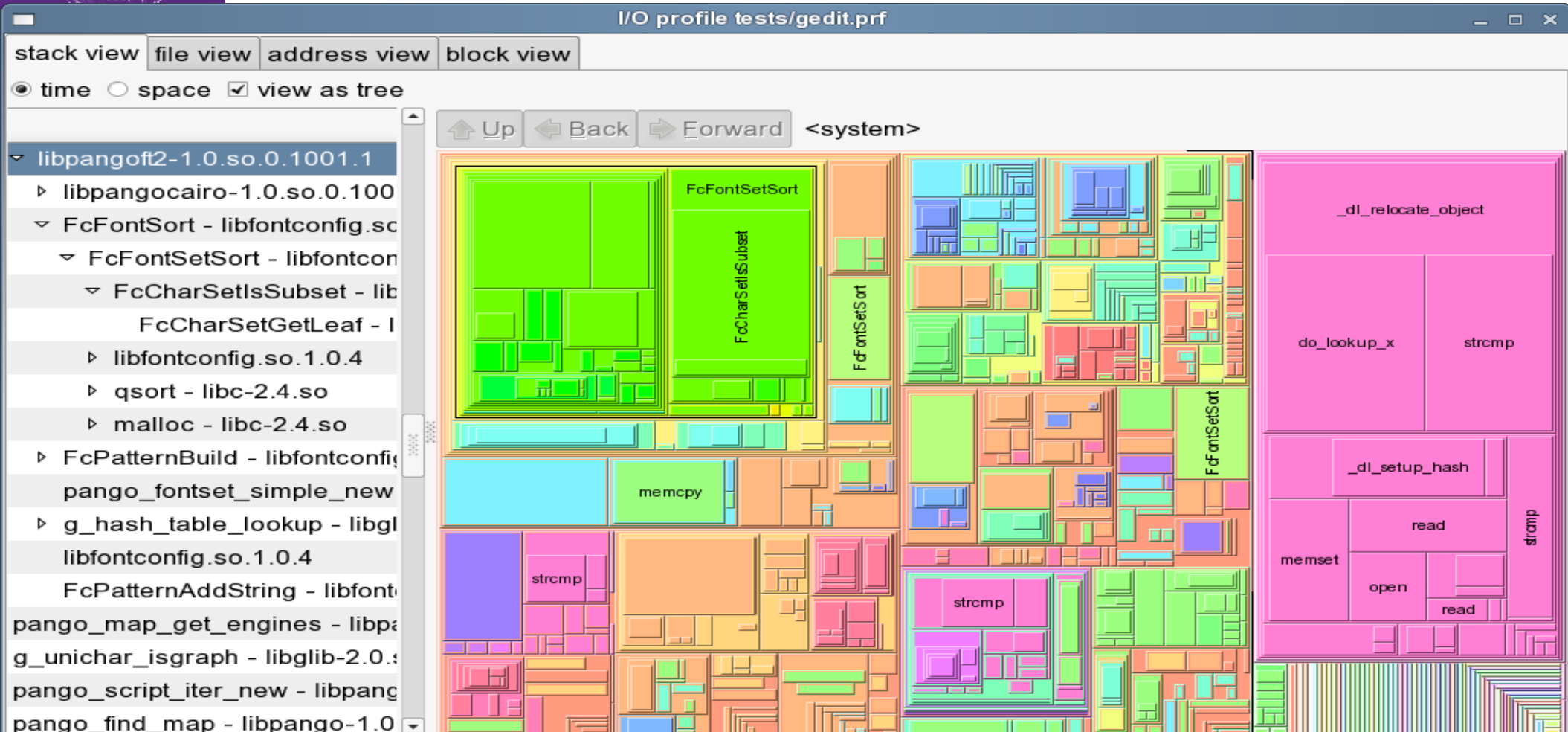
- Traced system-call I/O
- Traced memory-map first-page touching
- Generated a trace.

Disk-simulator / re-run-trace ...

- Aggregate stack traces, and visualize result:
- Split the trace from the file-system layout:
 - Different layouts → better or worse etc.

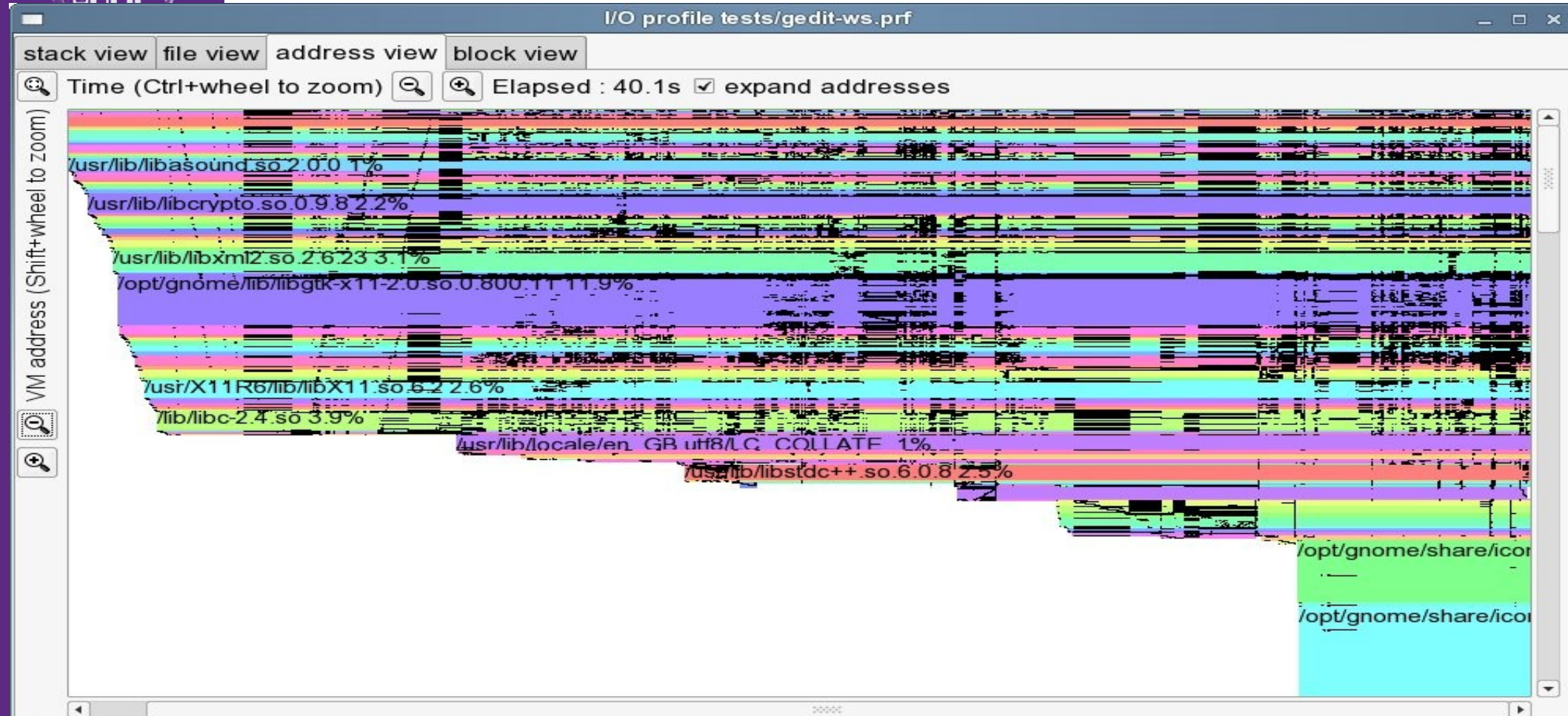


What can we see: stack mode



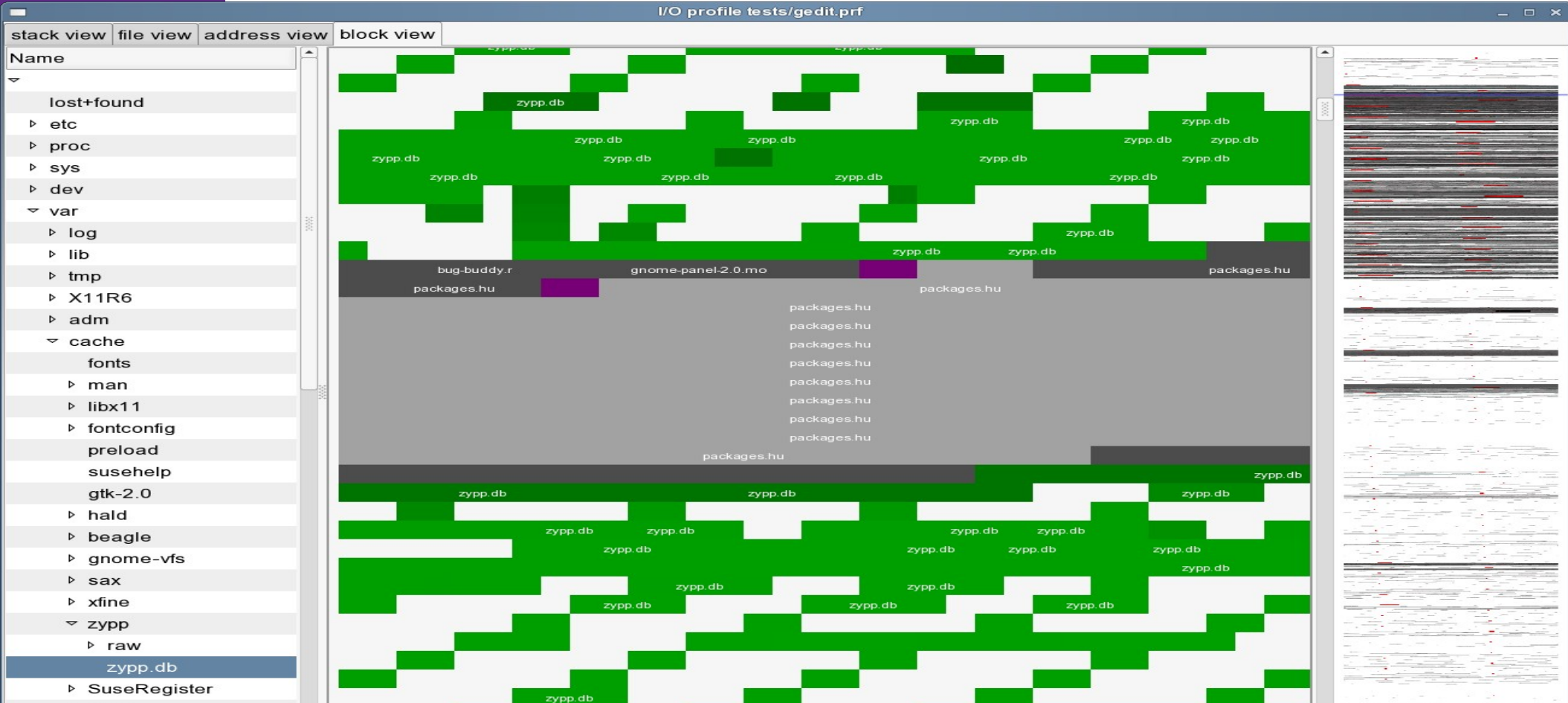


What can we see: address view





What can we see: disk view





Towards a fix:

Pre-loading sorted by inode ...

libmerged → major component re-work ...

Linux Hackers

- Plumbers talk → pointed at their file-system problems ...
- Ultimately: SSDs ...
- Empty writer: today: 2.37s cold, 2.17sec warm
 - **80% I/O to 8% I/O ...**
 - psychologically referent: web-pages

Perceived slowness to start ? #3

Where is the splash-screen ?



splashx.c ...

Split out the 'draw a splash'

- Draws splash
- Renders progress-bar
 - No bevel, no AA text, no CSS animated progress, no theming ...
 - data from a pipe to the main app
- `$ ldd oosplash | wc -l`
11 (shared libraries)



LibreOffice
Development Build



**LibreOffice happened: freeing
developers to re-factor &
improve aggressively**

shortly afterwards TDF setup

Code re-structuring ...



Why re-structure at all ?

A ~40 year old code-base

- Object Orientation mania
 - The silver-bullet
- OO toolkit - less interesting
- Demo / Office apps → very interesting.

What languages should we use ?

What platforms should we target ?

“Technology – the only industry that is more fashion driven than women’s fashion” – Larry Ellison

Lets re-write all 8 million lines in
<insert-language>:

*Java, Javascript, Rust,
Haskell ... !!!*

*#1 indicator of engineering
immaturity*

Tip#1 - don't buy a silver bullet:

*"there is **no** single development, in either technology or management technique, which by itself promises even one **order of magnitude improvement** within a decade in productivity, in reliability, in simplicity."* Fred Brooks – *The Mythical*

Man Month



Sample 10x productivity wins

Object Oriented programming !

Java → develop 10x faster

- Garbage collection ! It rocks.

C# → develop 10x faster

- Syntactic sugar! It rocks.

*"As such, whilst **Vala** is a modern language with all of the features you would expect, it gains its power from ..."*

- Language augmenting pre-processors are great

And on ... and on.



A personal favourite

“We encourage the OpenOffice group to quickly build their version of a spread sheet or a word app using JavaFX,”

- Larry Ellison (according to theregister) - June 2009

- Fatal mistake: using a unique name
- **Hot Tip** – your vanity language / platform project should have a non-google-trend-able name eg. ‘ruby’, ‘rust’, ‘go’ - not eg. ‘zsquat’
- JavaFX’s Classic Hype-Cycle trajectory / google trend





Lucking out on language ?

"The point I'm trying to make is that **the only reason** why all of this was possible, where we build ... [**Microsoft Office**] ... for 30 years, not only ship those applications but **evolve** them - **picked C/C++ - tremendously lucked out**. All the other languages - two categories:

1. language would have been re-invented 3 times and had to re-write the thing 3 times.
2. authors of language & industry lost interest and it would have stagnated"

- Igor Zaika (Microsoft, Distinguished Eng.) [CppCon 2014](#)



Language bits.

Language bigotry:

- Almost totally pointless
- **The** leading driver of pointless duplication
 - Wins over even license choice ... ?
- Lots of it out there.

We're still slowly writing out Java

- can't guarantee it is on the platform in the same way that binaries are.
- The embedded Rhino JS-in-Java impl ? ...
- Default HSQLDB format → in databases ...



Cross Platform-ness: Churn

Windows rendering APIs:

- **GDI**: basic version with [windows 1.0](#)~1985
 - Made the 16bit → 64bit transition well.

GDI+ in XP - 2001

DirectWrite Win7/2007

- finally a physical font API.

Direct2D: 2012

Easy to churn faster than big apps can handle

Linux toolkits

- gtk+1, Qt2 - 1999
- **gtk+2**, Qt3 - 2000
- **Qt4** - 2005
- **Gtk+3** - 2011
- Qt5 - 2012
- **Gtk+4** - 2021
- Qt6 - 2023

Apple: churn to match

Bold: in-use ...

Object Orientation !



Historic Object Orientation mania ...

“...let's take the desktop computer revolution to the next stage. What we need are true object-oriented environments.”

Philippe Kahn, Borland founder & CEO, 1991 ..

The New York Times, August 18, 1991, Section 3, Page 9.



Object Oriented programming #2 +1 year

Borland's - Philippe Kahn:

"Creating the initial "objects" is difficult, especially for programmers who are used to designing the old-fashioned way."

"But ultimately the payoff will be faster, less costly development of new programs and upgrades."

The New York Times (July 26, 1992, Section 3, Page 1)

"If the objects approach is so great, Mr. Gates has said, why are Borland's Windows products so late?"



How (not) to Design:

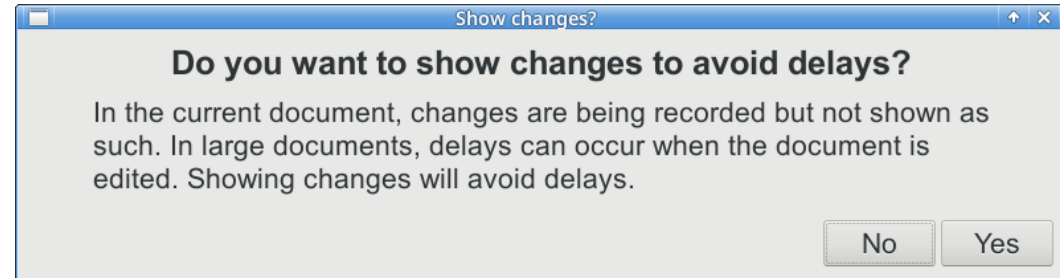
Look at the screen: what you see is objects !

- **Epic Object Oriented blunders:**

- I see a spreadsheet cell ! → object
- I see a paragraph ! → object
- I see a table ! → object

- **Power ideas:**

- Our file format can just be a recursive serialization of all our objects !!!!
- Change-tracking makes this too complex: so tack it on the side & mutate the model.



- **Oh dear ! Oh dear ...**



Fixed many of these things in recentish time.

- **Binfilter** → changing object graph ...
 - hideous hack ... removed in LibreOffice 4.0
- **Spreadsheets**
 - Massive re-factoring for AMD of the Calc core.
 - Sensible, contiguous type span data structures (MDDS)
 - Huge memory reductions, significant performance improvements
 - Thanks to Kohei Yoshida (Collabora)
- **Writer**
 - Red-lining turned into a view state – not model fiddling festival
 - Thanks to Michael Stahl (CIB) for Munich City.



Top Tip:
try to spot the panaceas of today and critically examine their claims. Hold onto the good.

Threading

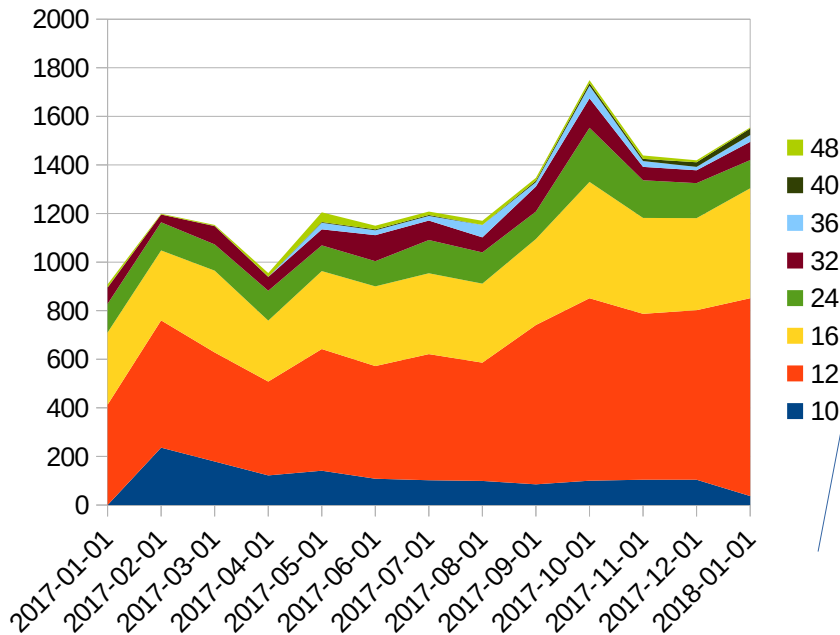


2017 Crash reporting trend

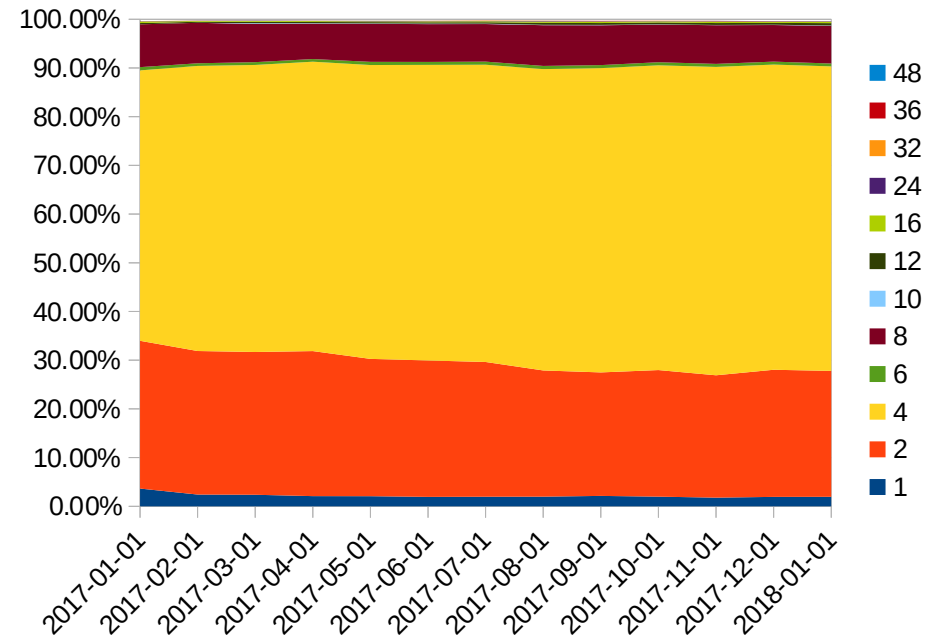


Frustratingly 'cores' not threads.

Reports from large core count machines.



Crash report % by CPU core count over time.





Threading fun:

Config Management

- Hideous example of 2nd system problem
- A path/key – value store
 - Plus some smarts.
- 55 mutexes taken & released on each key fetch.
 - And not thread-safe.

Replaced with a single big code-lock

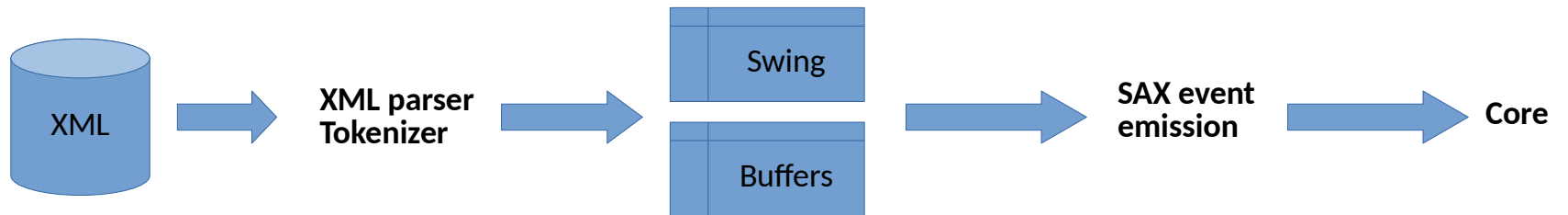
- Beware of granular locking ...
- Faster, smaller, simpler, reasonable code.



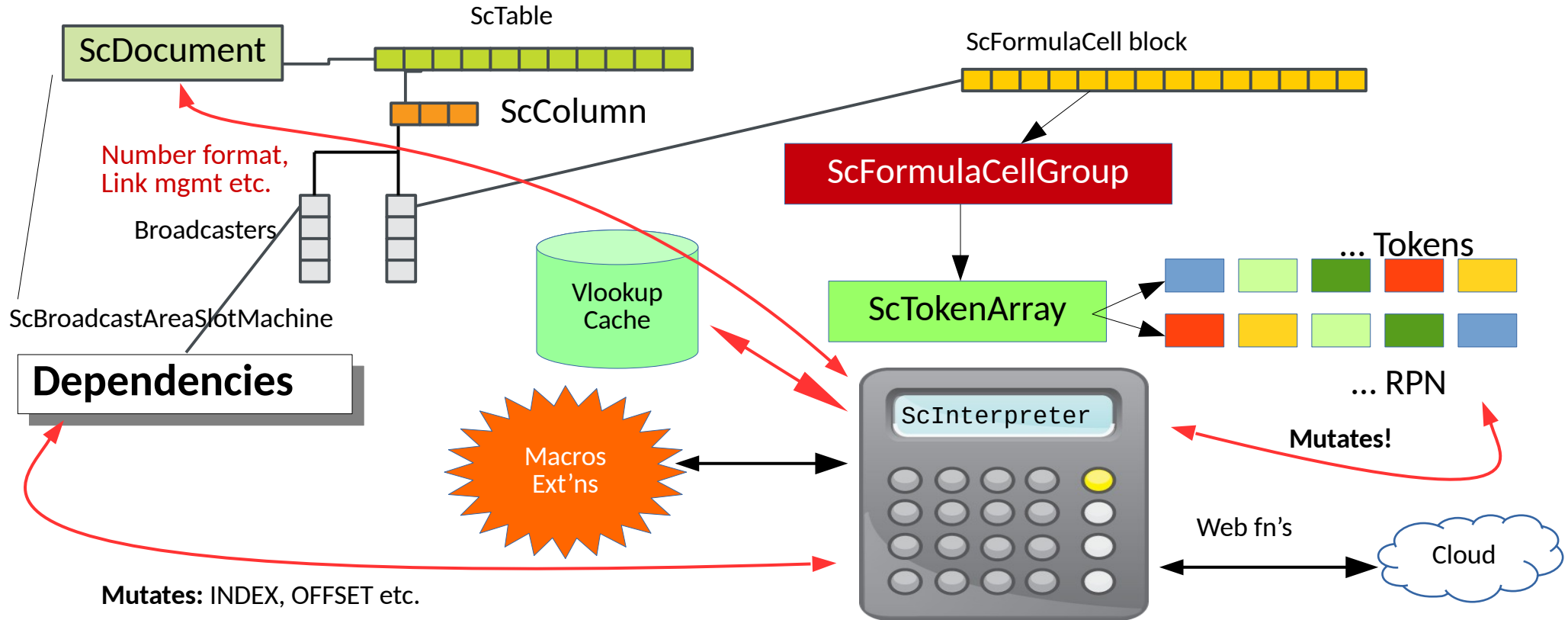
O(~1) XML parsing ...

Parsing XML is a significantly costly pain

- Format is needlessly complex; cf. JSON.
- Namespace handling & tokenization is a pain.
- Double checking for duplicate attributes bad too ...
- SAX API really poor for CPU cache usage:
 - a heavily fragmented workload.



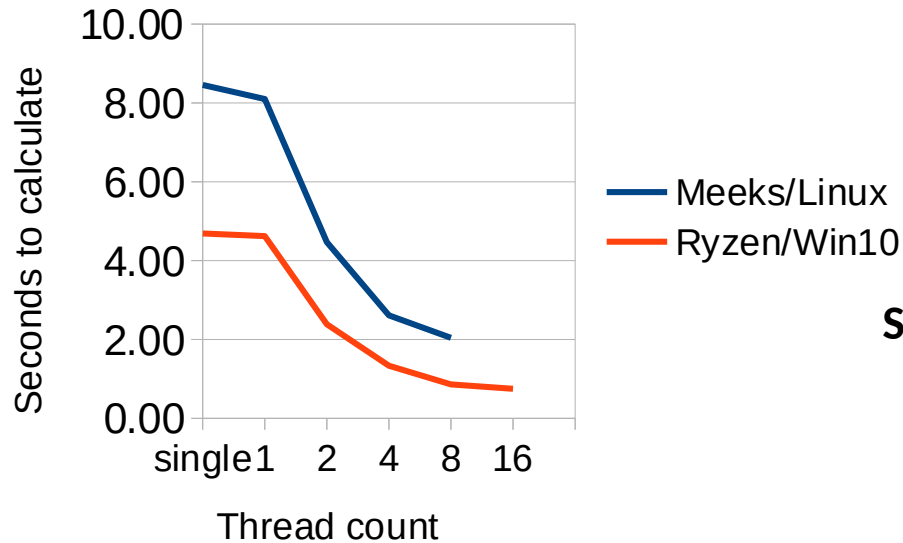
Calc: extreme coupling & threads.





Calc parallelism results ...

re-calculating 100k formulae on 1m doubles



Extreme care & choice of threading granularity

- Mostly lock-less, but lots of assertions.
- Would love more language support to help; C++ weak here.

Some good improvements

- Plenty more to do to expand the scope & reliability
- Embarrassing atomic operation thrash.

LibreOffice Cleanups ...



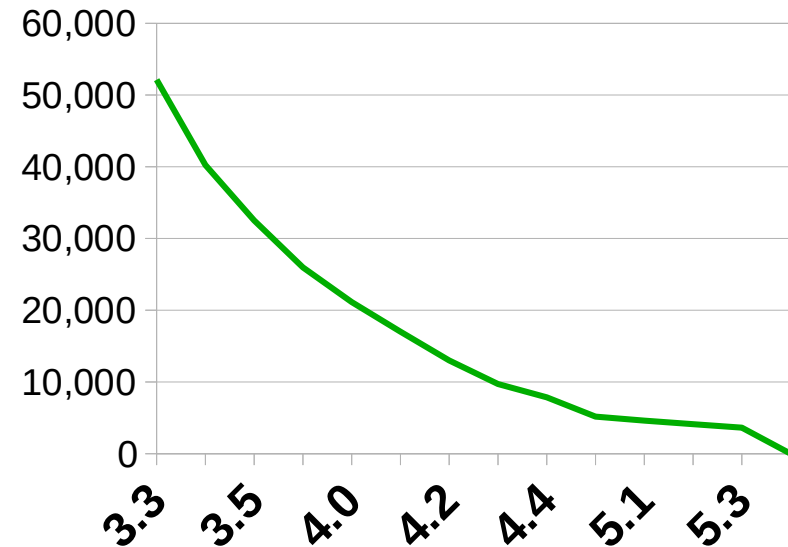
Tastier code - easier to chew

Seven years in - all German comments were gone

Thanks to:

Jens Carl, Johnny_M, Michael Stahl (RedHat), Katerina Behrens (CIB), Thomas Beck (CIB), Lukas Röllin, Gabor Kelemen, Stephan Bergmann (RedHat), dennisroczek, and many others.

Detected lines of German comment



Security: Coverity ... score 0.00

Excellent static checker.

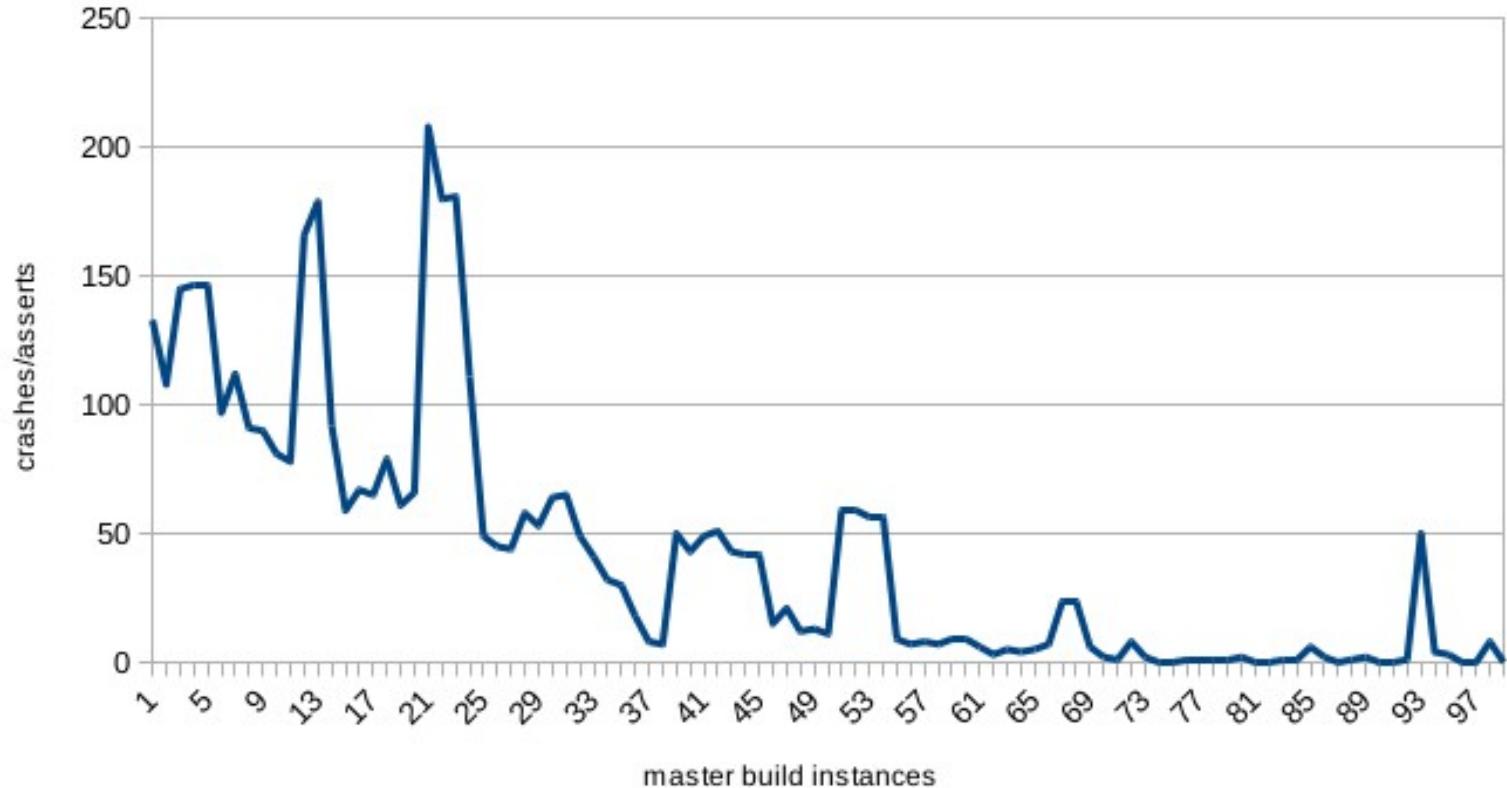
Outstanding vs Fixed defects over period of time





Systematic load crash testing: 100k+ files (now 350k+)

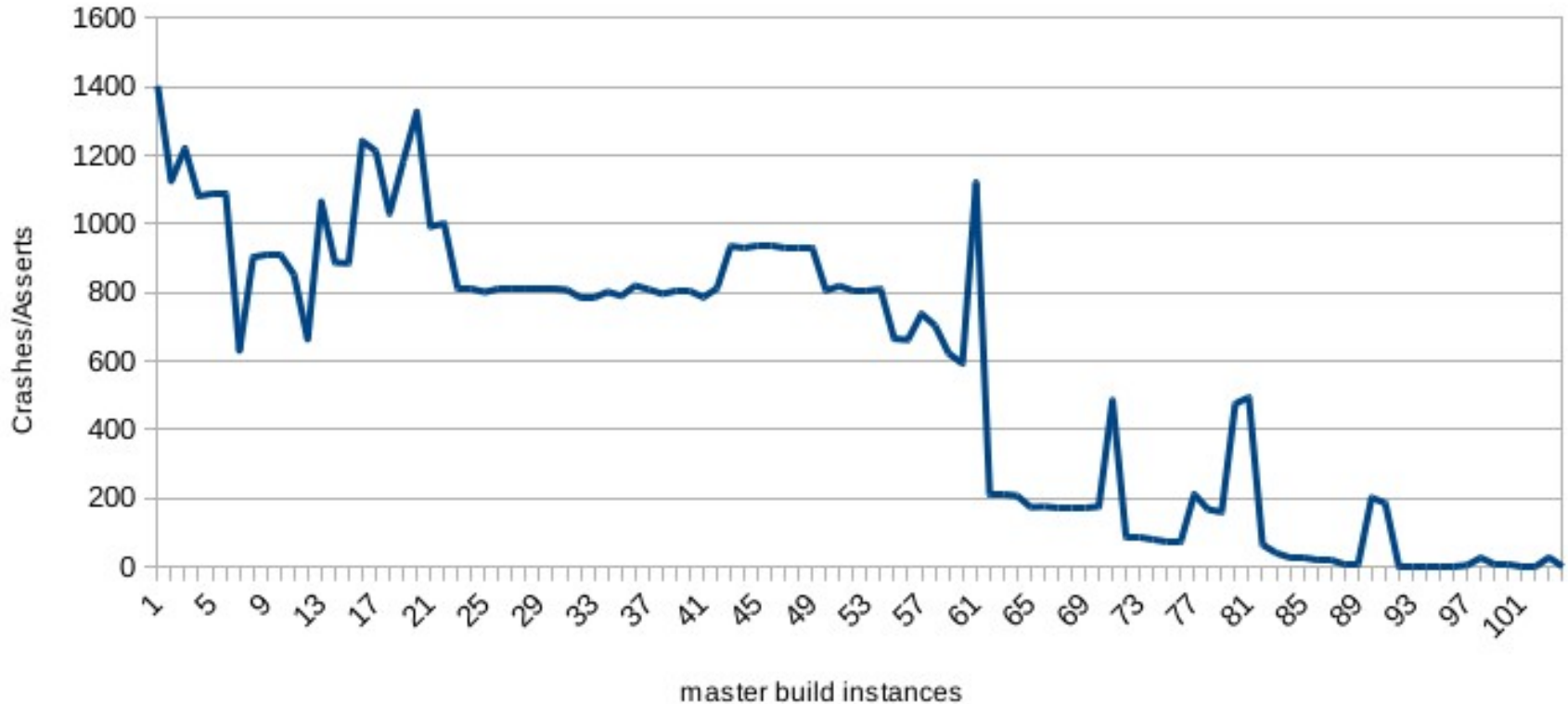
Import crashes over time





Systematic save crash testing: 100k+ files

Export crashes over time



Callgrind based automated performance testing.



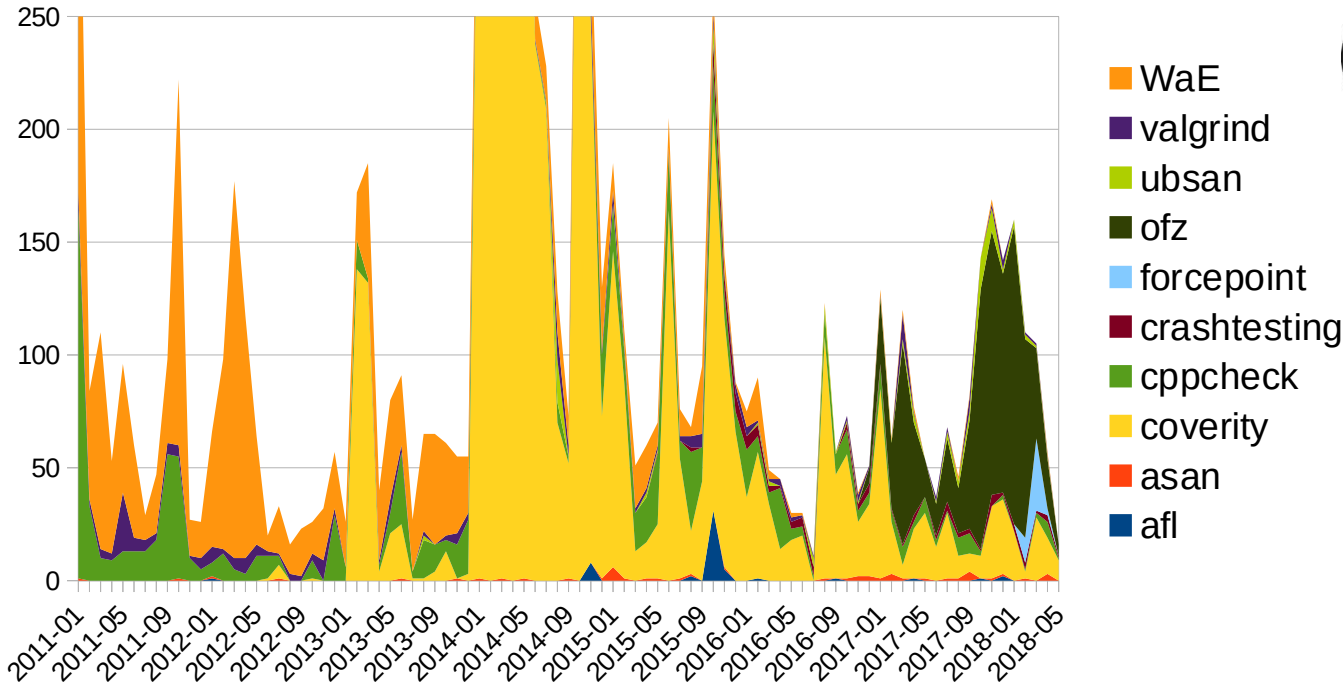


Code quality improvement ...

New tools find new bugs - and over time that reduces

Use lots of
different fuzzers
/ checkers !

Commits per month easily attributable to various tools



Google:
OSS-Fuzz





Less eye-strain horror

Code-base pre-dates templates: C-pre-processor generics instead left & right.

enum cleanup & strings ...

```
-case BUTTON_ABORT: aText = rtl::OUString(  
-     RTL_CONSTASCII_USTRINGPARAM( "Abort" ) );  
+case StandardButtonType::Abort: aText = "Abort";
```

Pretty iterators & auto

```
- for( std::unordered_map< Atom, Selection* >::iterator it =  
This->m_aSelections.begin(); it != This->m_aSelections.end();  
++it )  
+ for (auto const& selection : This->m_aSelections)
```

And tons more - still ongoing ...



Clang plugins ...

Expanding C++ checking:

- unexpected bool conversion hunting
- flatten: look for large if statement at end of function
- cast cleanups / re-writing

Wider scope of understanding

- Unused-fields checking
- count all users of default params
- un-necessary-virtual detection
- Use-unique-ptr

And tons more: thanks to

- *Noel Grandin (Collabora) & Stephan Bergman (RedHat)*



Real Native Widgets / Layout ...

Cartoon - Native Widgets ...

- Used for years on Win/Mac/Linux
- **Twist** widgets into rendering bits of themselves
- Missing: theming details, animations

Plus Native Menu-bar, File-selector, Tool-tips

A human can always lay-out a dialog better than auto-layout ! (or not)

- Now: finished converting all dialogs to glade / XML years back: **1200+** .ui files ...

Now starting to load native gtk+ dialogs, widgets & wrapping API - WxWidgets style

- *Thanks to Caolan McNamara (RedHat)*



Core Infrastructure Initiative

OSS-Fuzz

- Huge Testing infrastructure provided by Google
 - Used for Chrome & many other OSS projects.
 - ~1000 core cluster to hugely accelerate testing.
- Manual auditing – killed by AI ?
 - Finds newly introduced problems rapidly before they can escape (CVEs)

 THE **LINUX** FOUNDATION

Coverity – still a score of ~0.00

Thanks to Caolan McNamara (was at RedHat)



Avoiding deja-vu in bug-fixing

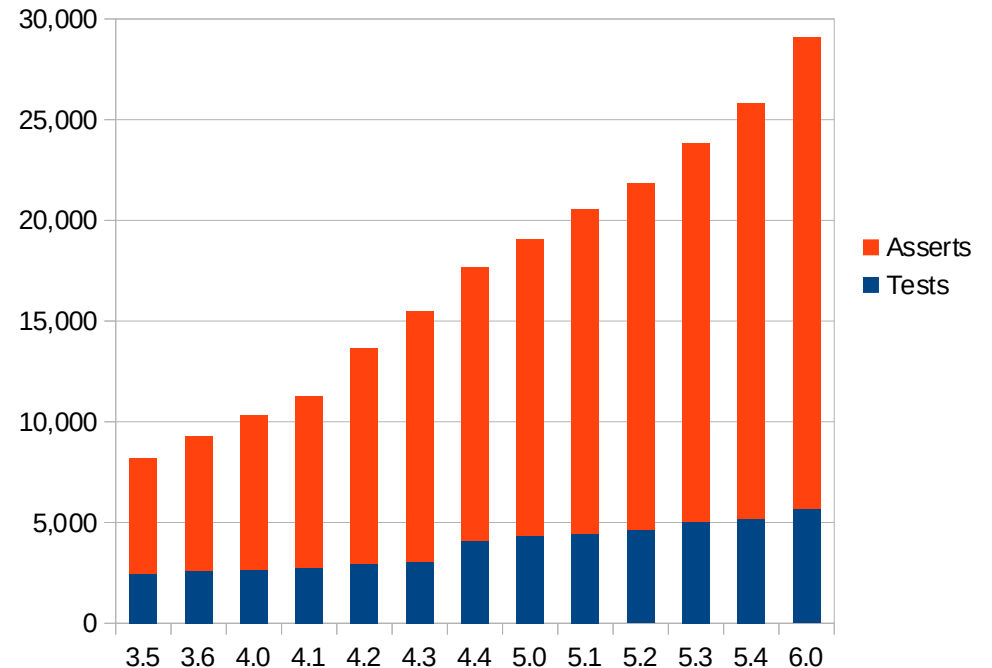
Fix each bug just once;

*Thanks to: Miklos Vajna (Collabora),
Stephan Bergmann (RedHat), Noel
Grandin (Collabora), Zdeněk Crhonek,
Caolán McNamara (RedHat), Ashod
Nakashian (Collabora), Justin Luth (SIL),
Tamás Zolnai (Collabora), Andrea
Gelmini, Jan Holesovsky (Collabora),
Markus Mohrhard, Eike Rathke
(RedHat), Mike Kaganski (Collabora),
Jens Carl, Michael Stahl (RedHat),
Szymon Kłos (Collabora), Tor Lillqvist
(Collabora), Chris Sherlock, Pranav Kant
(Collabora), Winfried Donkers, David
Tardon (RedHat)*

*with >20 unit test commits in
5.4/6.0*

Growth in unit tests over time

count of various CPPUNIT macros



The Web as a platform:



The web as a platform freak-show:

We know how to create languages

- To enable quality software development.

Javascript !

- No types
 - Not just un-checked & un-safe
 - Consider the JIT { 'x': 3, 'y': 5 }

No class / object orientation

- No encapsulation

Different between browsers

“Built with Global Variables !”

- Monster global: the DOM

“Access it through regular expressions !”

- All the worst features of perl ...

Split your JS code into modules:

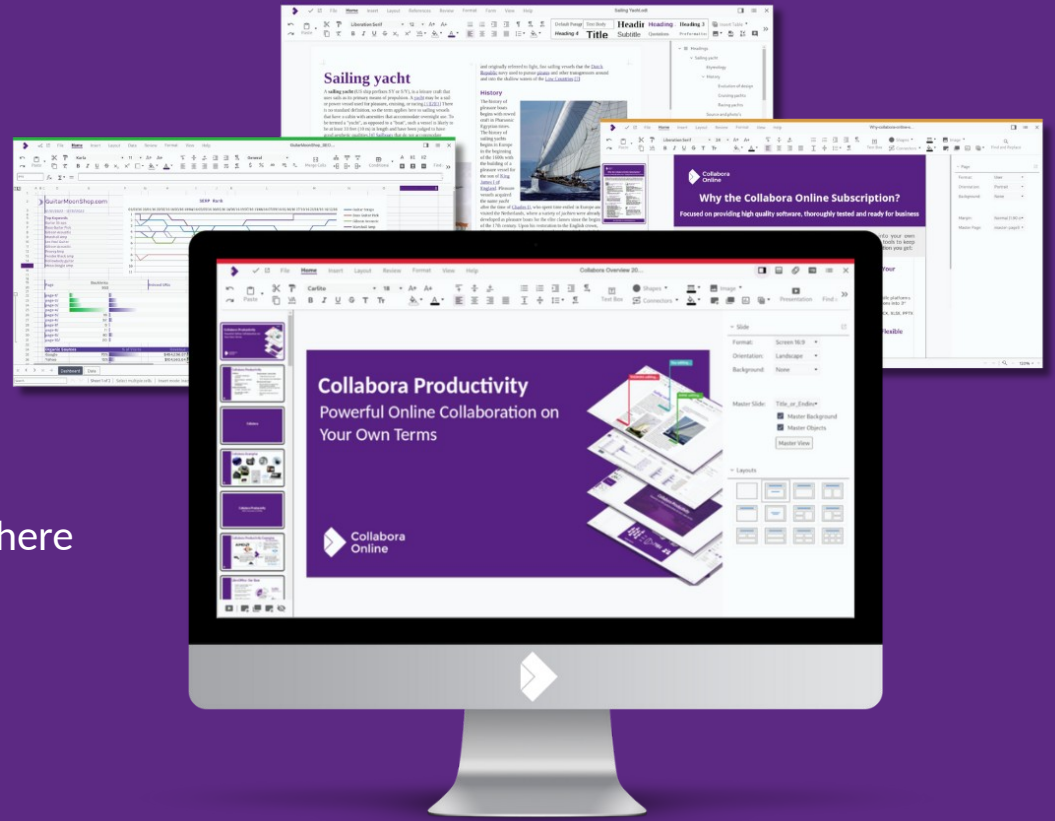
- Then you can't tell which executes in what order
- So bundle-ify them together again

Can't draw hair-lines in firefox:

- Canvas sizes beyond 127 x 127 can anti-alias paths [when they shouldn't](#)

But still – ubiquity is a killer feature

Built with LibreOffice technology
Rich, interoperable, collaborative editing, everywhere



~105 million docker image downloads



What is Collabora Online?

Documents + Spreadsheets + Slides + Drawings

- Viewing and collaborative editing
- Feature rich

Interoperability with Microsoft formats

- docx, doc, rtf, xlsx, xls, pptx, ppt

Import filters and viewing for

- PDF, Visio, Publisher and many more


Powerful WYSIWYG rendering

On-premise integration

Architecture - a bet on CPU threads and network getting cheaper over time



What is Collabora Online?



The screenshot displays the Collabora Online interface for editing a document titled "Sailing Yacht.odt". The top menu bar includes options like File, Home, Insert, Layout, References, Review, Format, Form, View, and Help. The ribbon below the menu contains various editing tools such as Paste, Bold, Italic, Underline, and text alignment options. The main editing area shows a document with a heading "Sailing yacht" and several paragraphs of text. A sidebar on the right contains a "Headings" panel with a tree view showing "Sailing yacht" and its sub-sections: "Etymology", "History", "Evolution of design", "Cruising yachts", and "Racing yachts". Below this is an "Images" panel with thumbnails for "Image3", "Image2", "Image1", and "Image4". A "Bookmarks" panel is also visible with entries like "firstHeading", "cite_ref-Webster_1-0", "cite_ref-2", "cite_ref-3", "cite_ref--3_4-0", "cite_ref-Perry_5-0", "cite_ref-6", "Etymology", and "cite_ref--1_7-0". The bottom status bar shows "Page 1 of 3", "1,050 words, 6,614 characters", and "English (UK)".

Sailing yacht

A **sailing yacht** (US ship prefixes SY or S/Y), is a leisure craft that uses sails as its primary means of propulsion. A **yacht** may be a sail or power vessel used for pleasure, cruising, or racing.[1][2][3] There is no standard definition, so the term applies here to sailing vessels that have a cabin with amenities that accommodate overnight use. To be termed a "yacht", as opposed to a "boat", such a vessel is likely to be at least 33 feet (10 m) in length and have been judged to have good aesthetic qualities.[4] **Sailboats** that do not accommodate overnight use or are smaller than 30 feet (9.1 m) are not universally called yachts. Sailing yachts in excess of 130 feet (40 m) are generally considered to be **superyachts**.^[5]

Sailing **yachts** are actively used in sport and are among categories recognized by the governing body of sailing sports, [World Sailing](#).^[6]

Etymology



Sailing yachts are actively used in sport and are among categories recognized by the governing body of sailing sports, [World Sailing](#).^[6]

The term, *yacht*, originates from the [Dutch](#) word *jacht* (pl. *jachten*, which means "hunt"),

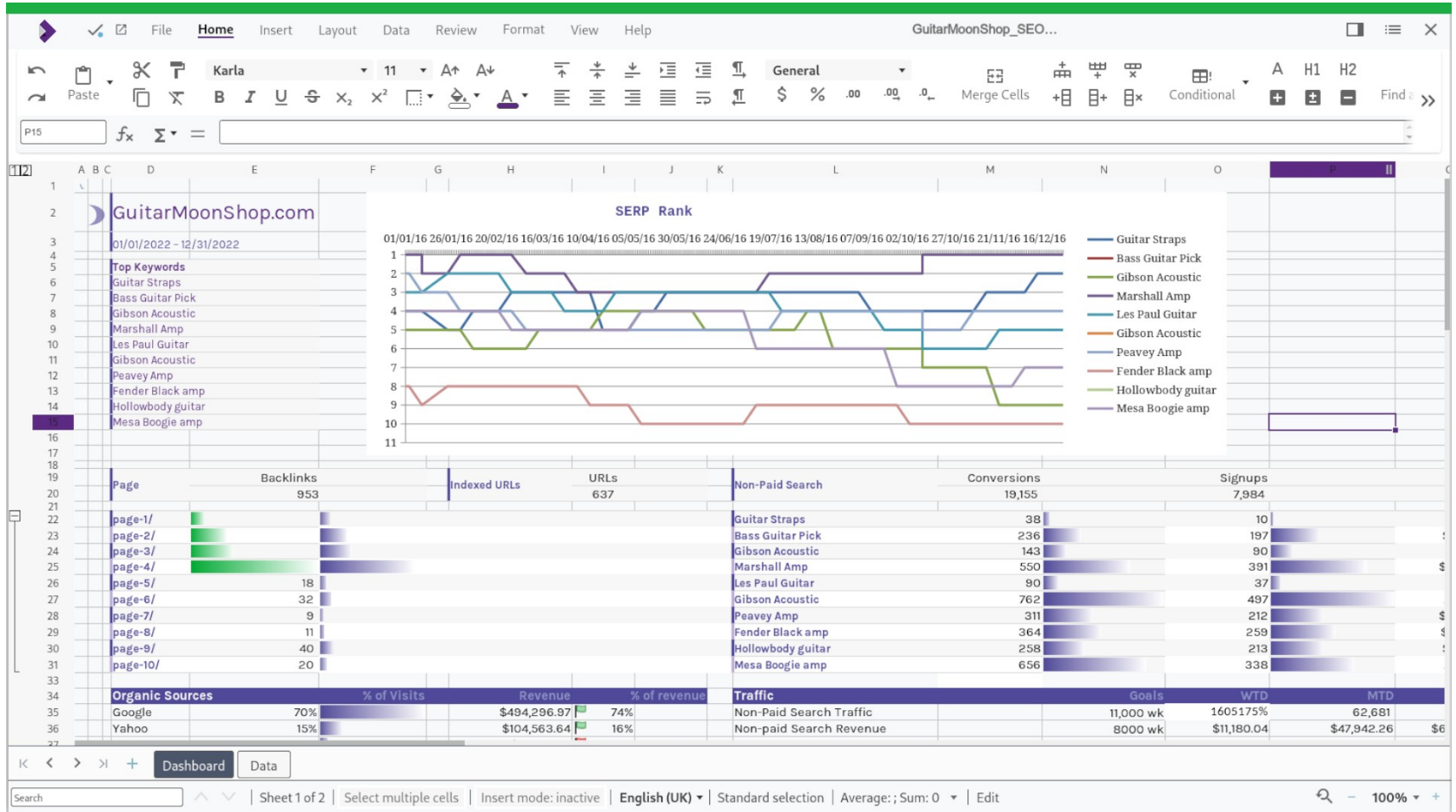
and originally referred to light, fast sailing vessels that the [Dutch Republic](#) navy used to pursue [pirates](#) and other transgressors around and into the shallow waters of the [Low Countries](#).^[7]

History

The history of pleasure boats begins with rowed craft in Pharaonic Egyptian times. The history of sailing yachts begins in Europe in the beginning of the 1600s with the building of a pleasure vessel for the son of [King James I of England](#). Pleasure vessels acquired the name *yacht* after the time of [Charles II](#), who spent time exiled in Europe and visited the Netherlands, where a variety of *jachten* were already well developed as pleasure boats for the elite classes since the beginning of the 17th century. Upon his restoration to the English crown, Charles commissioned a series of naval vessels which included at least one experimental https://en.wikipedia.org/wiki/Charles_II_of_England between two vessels occurred in 1604, followed by the first open sailing competition in 1663 in English waters.^[8]



Collaborative Spreadsheets



Collaborative Drawing

Collabora Online

Why the Collabora Online Subscription?

Focused on providing high quality software, thoroughly tested and ready for business

Collabora Online is a powerful online office suite, which you can integrate into your own infrastructure. We take digital sovereignty seriously and provide you with all the tools to keep your data secure, without compromising on features. With our supported subscription you get:

- Assured Digital Sovereignty & Security**
 - On-premise / self hosted - control your data, app & network
 - No ties to any storage or authentication system
 - Documents are not broadcasted to all clients, but shared as pixels
 - Each document session gets its own private jail - with no local copy between document editing sessions
 - Simple integration based on a WOPI API providing storage & authentication
- No Limitations for Your Productivity**
 - All code is Open Source
 - Full featured, rapid improvement
 - Available for all major PC and mobile platforms
 - Flexible - many different integrations into 3rd party solutions
 - Excellent interoperability with DOCX, XLSX, PPTX and other Microsoft files
- A Cost Effective & Flexible Plan**
 - Simple pricing plan
 - High Availability as standard

Product Management Relationship

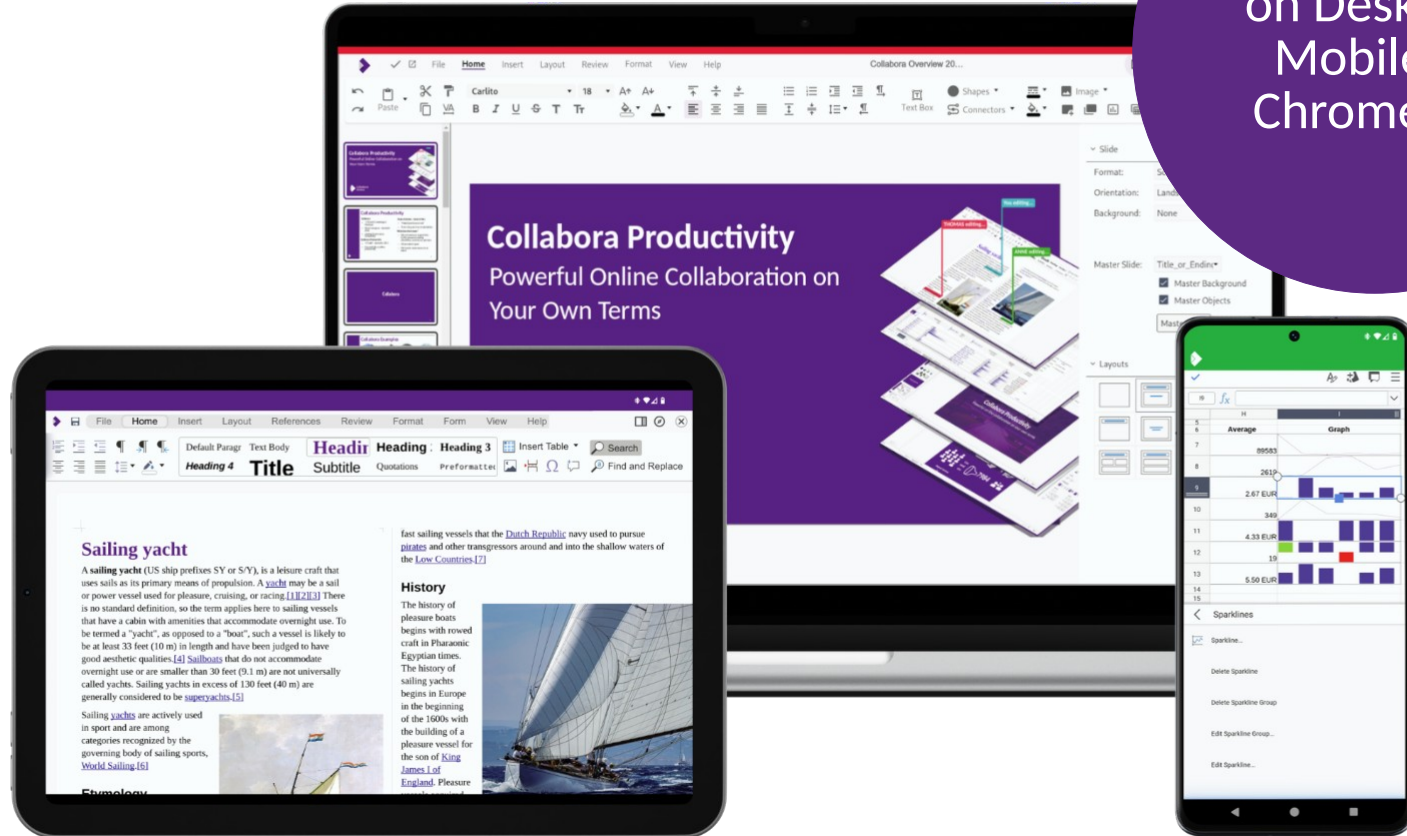
Page 1 of 1 | English (UK) | Edit | 120%

Collaborative Presentations

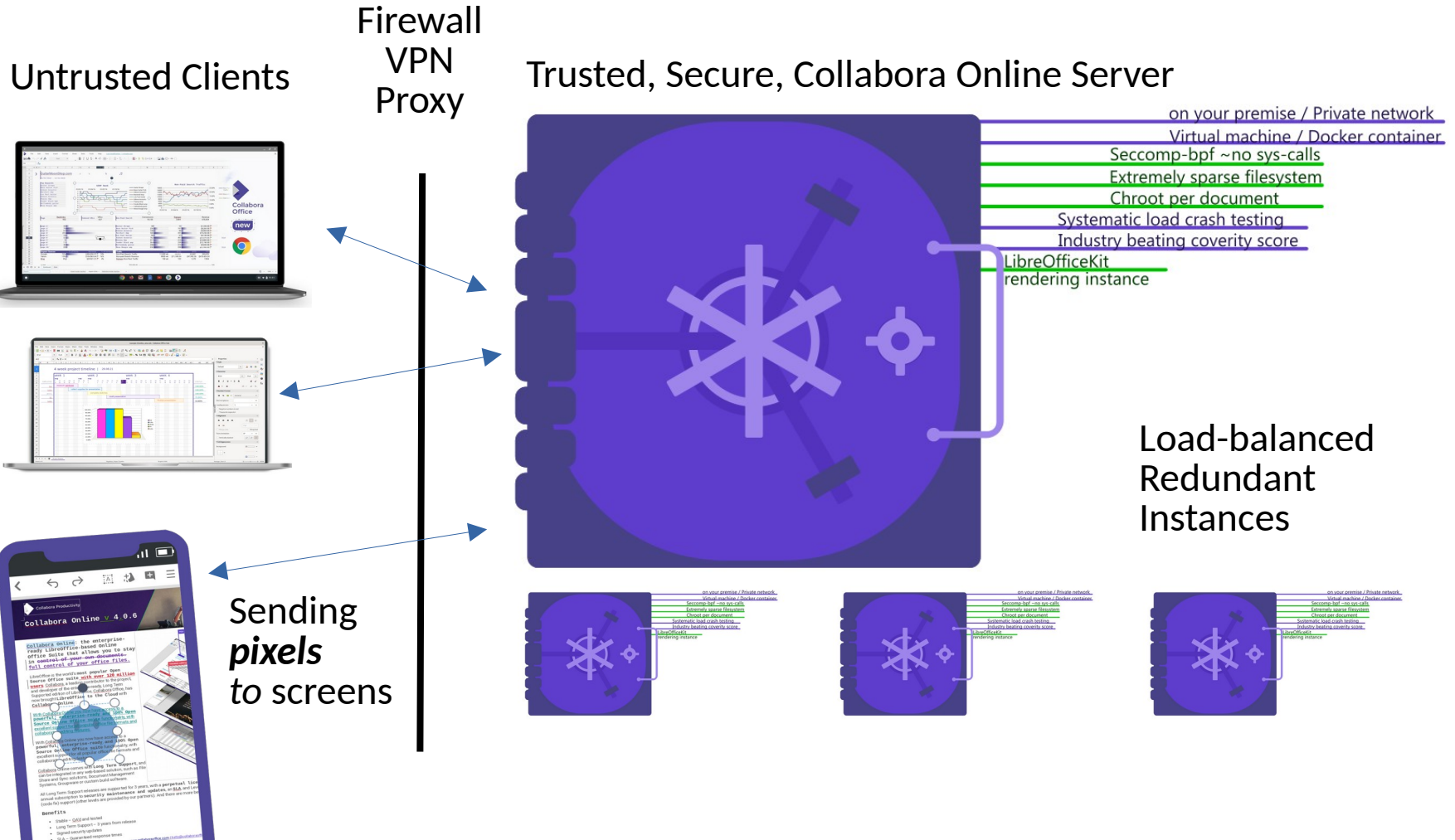
The screenshot displays the Collabora Online presentation editor interface. The main slide features the title "Collabora Productivity" and the subtitle "Powerful Online Collaboration on Your Own Terms". The Collabora Online logo is visible in the bottom left corner of the slide. The interface includes a top menu bar with options like File, Home, Insert, Layout, Review, Format, View, and Help. A ribbon-style toolbar below the menu bar contains various editing tools such as Paste, Bold, Italic, Underline, Text, Text Box, Connectors, and Image. On the right side, there is a "Slide" panel with settings for Format (Screen 16:9), Orientation (Landscape), Background (None), and Master Slide (Title_or_Ending). Below the slide panel is a "Layouts" section showing a grid of different slide layouts. The main slide content includes a stack of overlapping presentation slides, with callouts indicating "THOMAS editing" and "ANNE editing" on specific elements.

Collabora for Android, iOS & Chromebooks

Also available
on Desktop,
Mobile &
Chrome OS



Document Security





Different optimizations

Online

- dlopens ~world (RTLD_BIND_NOW)
- Loads ~all dictionaries & hyphenation
- then forks children
 - Copy-on-write-ness everywhere.
- waste memory in pre-init to save in children
- Waste CPU time on startup to save it later.

Catching bad behaviour:

- Touching / un-sharing pages:



Compare memory: loolmap

Useful tool for eg. kdeinit, webservers?

```

heap page: 0x011e9000 (310/967) - touched: 17 - was shared - from [heap]
...
0x0130  03 00 00 00 0a 00 00 00 4d 00 61 00 74 00 68 00 | .....M.a.t.h.
-par't-  1
0x0140  4d 00 4c 00 20 00 32 00 2e 00 30 00 00 00 65 00 | M.L. .2...0...e.
0x0150  00 00 00 00 00 00 00 00 61 00 00 00 00 00 00 00 | .....a.....

```

Ref-count in a Unicode string incremented

Also dumps all strings it can find with -strings

Drove pre-init string staticization

Totals for heap

- shared 23040 (92160kB)
- unshared 982 (3928kB)
- same but unshared 191 (764kB)
- dirtied bytes touched 30718 per page 31.28

Collabora Online

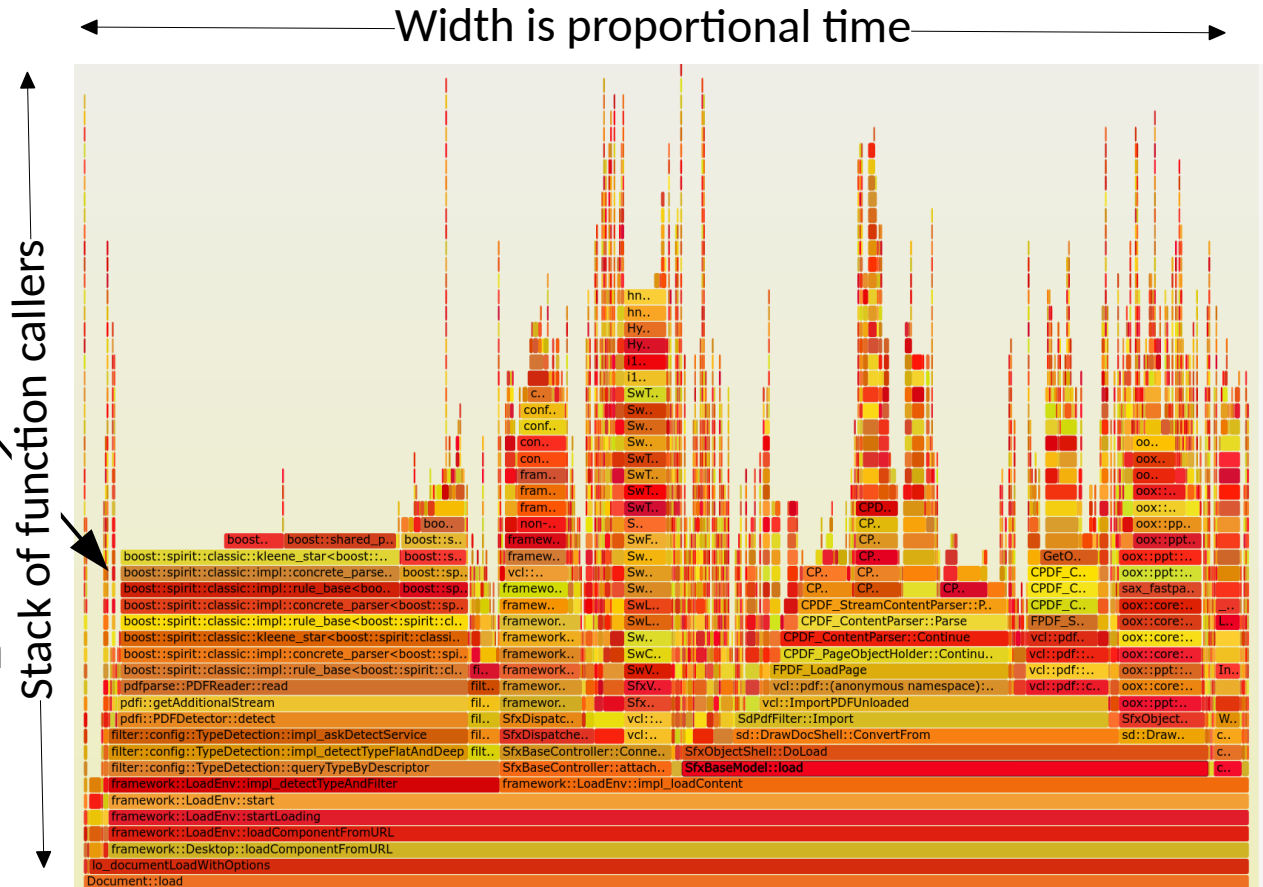
Powerful Online Collaboration



Long Running Demo Profiles: A Week

What does it mean?

- 17% of a week of profiling:
- Detecting: “is it a Hybrid PDF”?
- Unbelievably wasteful
- Scan last 4k block ‘AdditionalStreams’ +17%





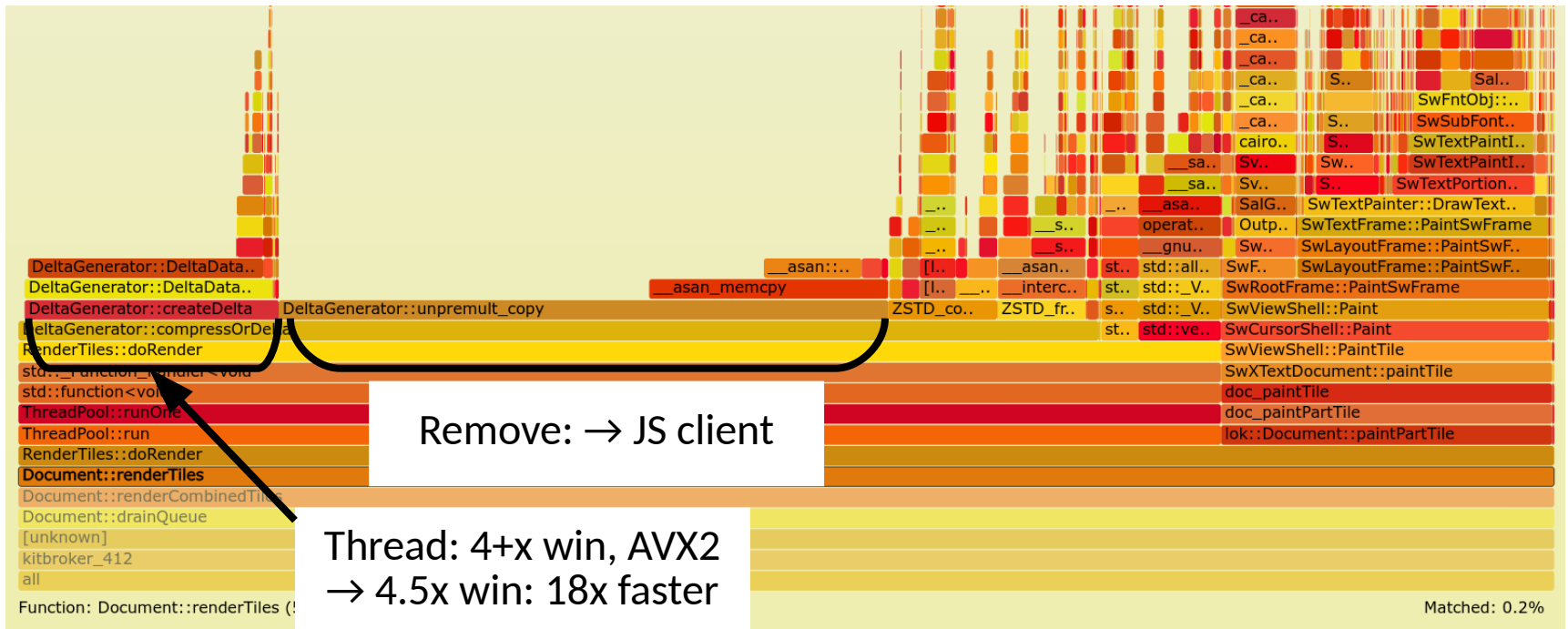
Rendering Tiles 40% of Your Profile ?

Lots of unnecessary re-rendering

- Pruning the simple cases out

Silliness in various places

- Now much improved





The Magic of AVX2 – Branch Free RLE

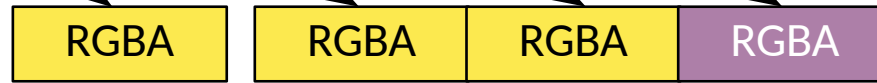
prepend 'last' pixel of the last block

Block of next four pixels to RLE ...

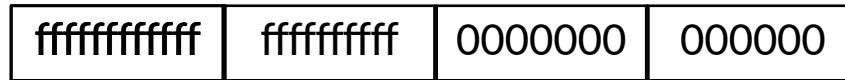


Next time's 'last' pixel:

Build comparison register:

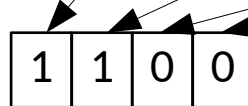


Compare:
cmpeq_epi32



NB. really 8 pixels at a time, not four and more 0's and f's needed ...

Magic: floating point sign mask:
movemask_ps



This is our RLE mask.

How many pixels to copy? popcount(RLE mask)
Which ones? RLE Mask → LUT + AVX2 gather
permutevar8x32_epi32 ... is your friend

Automatic Documents

- Create final PDFs
- or
- Create prefilled ODP/DOCX documents to continue working on

The screenshot shows the LibreOffice Writer interface with a form titled "INWARD REPORT (SHIPS AND BOATS OTHER THAN SMALL CRAFT)". The form is part of a document from the New Zealand Customs Service. The form is titled "PART A: ARRIVAL DETAILS" and contains the following fields:

PART A: ARRIVAL DETAILS	
Name of ship	HMB Endeavour
Date of arrival	10/01/1769
Time of arrival	Click here to enter text
Port of arrival	Poverty Bay
Name of person in charge	James Cook
Name and address of ship's agent in New Zealand	Click here to enter text
IMO number	Click here to enter text
Gross register tonnes	Click here to enter text
Period of stay in New Zealand	Choose a date
IF CARRYING CARGO, BRIEF DESCRIPTION	
LIST SUBSEQUENT PORTS OF CALL WITHIN 100 NAUTICAL MILES OF NEW ZEALAND	
Next New Zealand ports	
1. Auckland	
2. Wellington	
3. Click here to enter text	
4. Click here to enter text	
5. Click here to enter text	

The calendar for September 2024 is open, showing the 24th as the selected date.

Automatic Documents

Properties of "docStructureChartExampleOriginal"

General Description Custom Properties Font Statistics

Title: _____

Subject: _____

Keywords: _____

Contributor: _____

Coverage: _____

Identifier: _____

Publisher: _____

Relation: _____

Rights: _____

Source: _____

Type: _____

Comments: _____

Help

```
1 {  
2   "Transforms": {  
3     "DocumentProperties": {  
4       "Title": "Title TxT",  
5       "Subject": "Subject TxT",  
6       "Keywords": [ ],  
7       "Contributor": ["Contributor1 TxT",  
8       "Coverage": "Coverage TxT",  
9       "Identifier": "Identifier TxT",  
10      "Publisher": ["Publisher TxT", "Re  
11      "Relation": ["Relation TxT", "Re  
12      "Rights": "Rights TxT",  
13      "Source": "Source TxT",  
14      "Type": "Type TxT"  
15      "Description": "Description TxT"  
16    }  
17  }  
18 }
```

Properties of "temp2"

General Description Custom Properties Font Statistics

Title: Title TxT

Subject: Subject TxT

Keywords: _____

Contributor: Contributor1 TxT, Contributor2 TxT

Coverage: Coverage TxT

Identifier: Identifier TxT

Publisher: Publisher TxT, Publisher2 TxT

Relation: Relation TxT, Relation2 TxT

Rights: Rights TxT

Source: Source TxT

Type: Type TxT

Comments: Description TxT

Help Cancel OK

- With simple JSON
 - Add/remove file properties
 - Create 'Custom Properties'

Other Recent Updates

Thanks to our Clients and Partners eg.



Zentrum
Digitale
Souveränität



LibreOffice
The Document Foundation



openDesk
Der Souveräne Arbeitsplatz



Adfinis



KOLAB

COLLABORATE IN CONFIDENCE



EGroupware
SMART ONLINE OFFICE



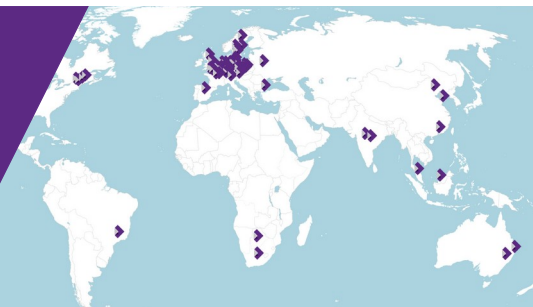
SWISSVON
voice over networks services and solutions



Find your local partner:

[www.col.la/
partners](http://www.col.la/partners)

Over 230 partners
offering professional support
in over 60 countries
Find your partner today!





Need smart people to help us create the future

- An **Artificial Intelligence** of stunning subtlety, of conversational brilliance
- That can understand what you're doing
- And can suggest cool new things to do.
- “Trained on Documents” (TM)



Conclusions



FLOSS is amazing

Such a lot of interesting code

- No barriers to finding & fixing the real problems

Increasingly – Open Source Rocks

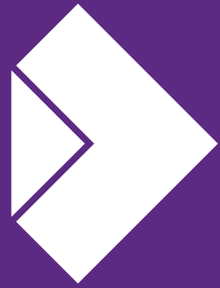
- Now two platforms to wrestle
 - Linux on the server
 - FLOSS browsers

Real heros work on the giant & fascinating problems

- One-code-change to ~200million LibreOffice users + ~100m COOL users

Join us to help recover people's Digital Sovereignty

- Control over and ownership of their data
- Easy hacks to get involved COOL, LibreOffice



Collabora Online

Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27

Thank you!



@CollaboraOffice
hello@collaboraoffice.com
www.collaboraoffice.com