Zoekt, improved

Code search, one year later

Han-Wen Nienhuys (hanwen@google.com)

github.com/google/zoekt, a code source engine

Seek (and ye shall eat spinach)

- Fast!
- Open source!
- Regular expressions, AND/OR/NOT-queries
- Git aware indexing
- Easy to deploy
- Instance at https://cs.bazel.build

Why?

- Understand/navigate code
- Especially outside of IDE

Overview

- Intro
- Quality improvements
  - search Unicode
  - better thresholds
- Secure deployment
- Closing words
- Bonus: gerrit ACL support

Unicode support

- Substring search used ASCII/bytes
  - Works, kinda
- But regular expression engine uses Unicode/UTF-8
  - A bug, kinda
- Gerrit repo is not ASCII only

(wikipedia)

Substring search: positional trigram

Input

```
code model
0123 45678
```

File 1 File 2

Index

```
cod: 0
ode: 1,5
mod: 4
del: 6
```

Candidates

Query “temp 50C max”

➡

```
find
  “tem”, ”max”
  10 chars apart
```

Case insensitive

Query “temp 50C max”

➡

```
find
  {tem,Tem,tEm,..}, {max,..}
  10 chars apart
```

Match

String comparison at found offset

Unicode

Unicode is a number ➜ meaning map

Example: 25991 = 文

- ASCII (1963), 128 characters 7-bit
  - A, b, DEL, ~
- Unicode 1.0 (1988) 65336 code points 16-bit
  - **ã, 文 , ➜**
- Unicode 2.0 (1996) 2M code points 21-bit
  - □ , 💩

Storing Unicode: UTF-8

bits	Byte 1	Byte 2	Byte 3	Byte 4
7	0xxxxxxx			
11	110xxxxx	10xxxxxx		
16	1110xxxx	10xxxxxx	10xxxxxx	
21	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- Pro: ASCII stays ASCII
- Con: no random access
  - Most apps don’t need it
  - But we do :-(

The Unicode Case

Find “`temp 323K max`”

• k = 107	ASCII lowercase k	1 byte
• K = 75	ASCII uppercase K	1 byte
• K = U+212A	Kelvin symbol	3 bytes


```
temp 323K max
0123456789012
```

```
temp 323K  max
012345678901234
```

- 10 bytes apart?
- 12 bytes apart?

Unicode: case closed

- Trigrams are unicode
- 3 x 21-bits = 63. Fits in int64
- Offsets are for code points
- Translate to byte offsets for final match

Input

```
code 💩odel
0123 4 5678
```

Index

```
cod:  0
ode:  1,5
💩od: 4
del:  6
```

Offsets table

codepoint	byte
3	3
6	9

Where is ImmutableList?

Search some code: r:guava immutablelist Max results: 50 Search

Found 5240 results in 723 files (26KB index data, 6426 ngram matches, 785 docs considered, 785 docs (12MB) loaded,

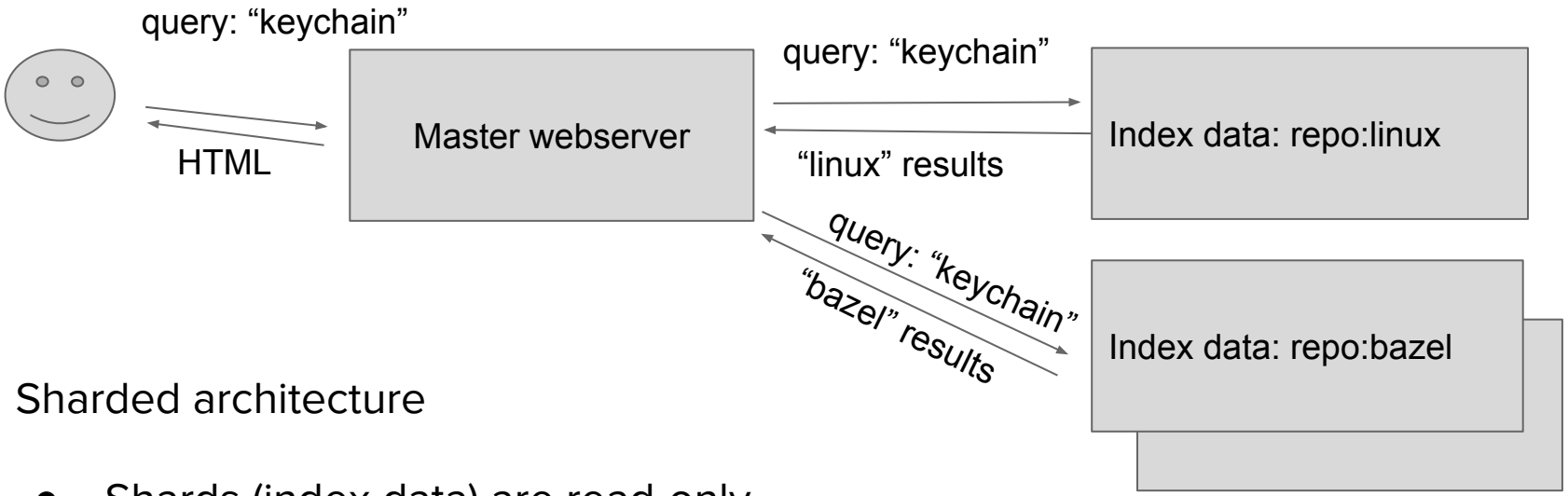
```
(and repo:guava substr:"immutablelist") with options &zoekt.SearchOptions{EstimateDocCount:false,
```

in 15.582855ms (queued: 4.256µs)

github.com/google/guava:guava-gwt/src-super/com/google/common/collect/super/com/google/common/collect/ImmutableL

```
41: public abstract class ImmutableList<E> extends ImmutableCollection<E>
46:   ImmutableList() {}
49:   public static <E> Collector<E, ?, ImmutableList<E>> toImmutableList() {
195:   static <E> ImmutableList<E> asImmutableList(Object[] elements) {
36:  * GWT emulated version of {@link com.google.common.collect.ImmutableList}. TODO(cpovirk): more doc
43:   static final ImmutableList<Object> EMPTY =
55:   public static <E> ImmutableList<E> of() {
56:     return (ImmutableList<E>) EMPTY;
59:   public static <E> ImmutableList<E> of(E element) {
```

Sharded architecture



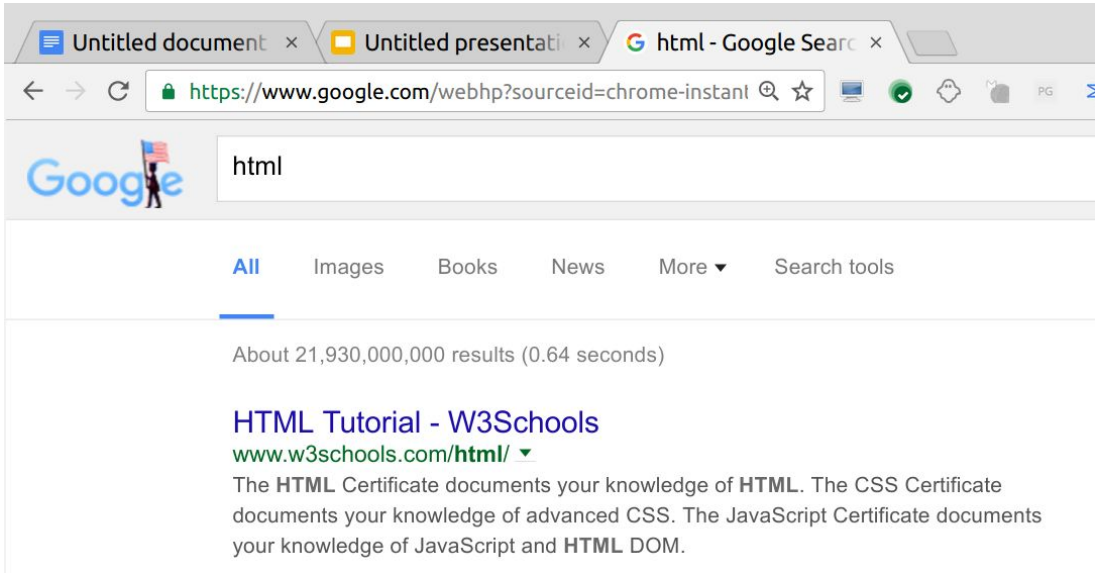
Sharded architecture

- Shards (index data) are read-only
- Shards don’t communicate
- Can search in parallel (“embarrassingly parallel”)

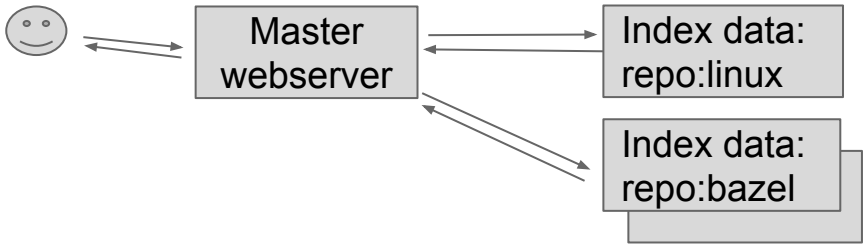
Works great for rare strings

Searching common terms

- Showing all results too expensive/unnecessary
- Stop after finding ***enough*** matches



Stop after “enough” matches



- Web UI sets N (default 50)
  - N total important matches
  - 10 * N total matches
- Per shard maximums
  - N/10 important matches per shard
  - 5 * N matches per shard

```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

xkcd.com/221

Where is ImmutableList?

Search some code: r:guava immutablelist Max results: 50 Search

Found 5240 results in 723 files (26KB index data, 6426 ngram matches, 785 docs considered, 785 docs (12MB) loaded,

```
(and repo:guava substr:"immutablelist") with options &zoekt.SearchOptions{EstimateDocCount:false,
```

in 15.582855ms (queued: 4.256µs)

CHEAP!

github.com/google/guava:guava-gwt/src-super/com/google/common/collect/super/com/google/common/collect/ImmutableL

```
41: public abstract class ImmutableList<E> extends ImmutableCollection<E>
46:   ImmutableList() {}
49:   public static <E> Collector<E, ?, ImmutableList<E>> toImmutableList() {
195:   static <E> ImmutableList<E> asImmutableList(Object[] elements) {
36:  * GWT emulated version of {@link com.google.common.collect.ImmutableList}. TODO(cpovirk): more doc
43:   static final ImmutableList<Object> EMPTY =
55:   public static <E> ImmutableList<E> of() {
56:     return (ImmutableList<E>) EMPTY;
59:   public static <E> ImmutableList<E> of(E element) {
```

Smarter thresholds

`ImmutableList`

6,000,000 files

`repo:"guava"` `ImmutableList`

3,000 files.

Idea:

1. Quick search
   - Evaluate `repo:` queries
   - Estimate file counts
2. Adjust thresholds
3. Real search

Now the real Guava ImmutableList gets returned too.

Deployment

- Deploy at http://cs.bazel.build
  - 6000 open-source repositories
  - Many bugs found & fixed
  - Use Google Cloud
- Project for fun; no headaches
  - Cookie troubles
  - Security breaches

Threat model

What is there to get?

● Private code	leak private code	n/a, public only
● SSL keys	MITM users	n/a, in Load Balancer
● Git access tokens	obtain private code	✘ github read token
● Machine access	abuse	✘

How to get in?

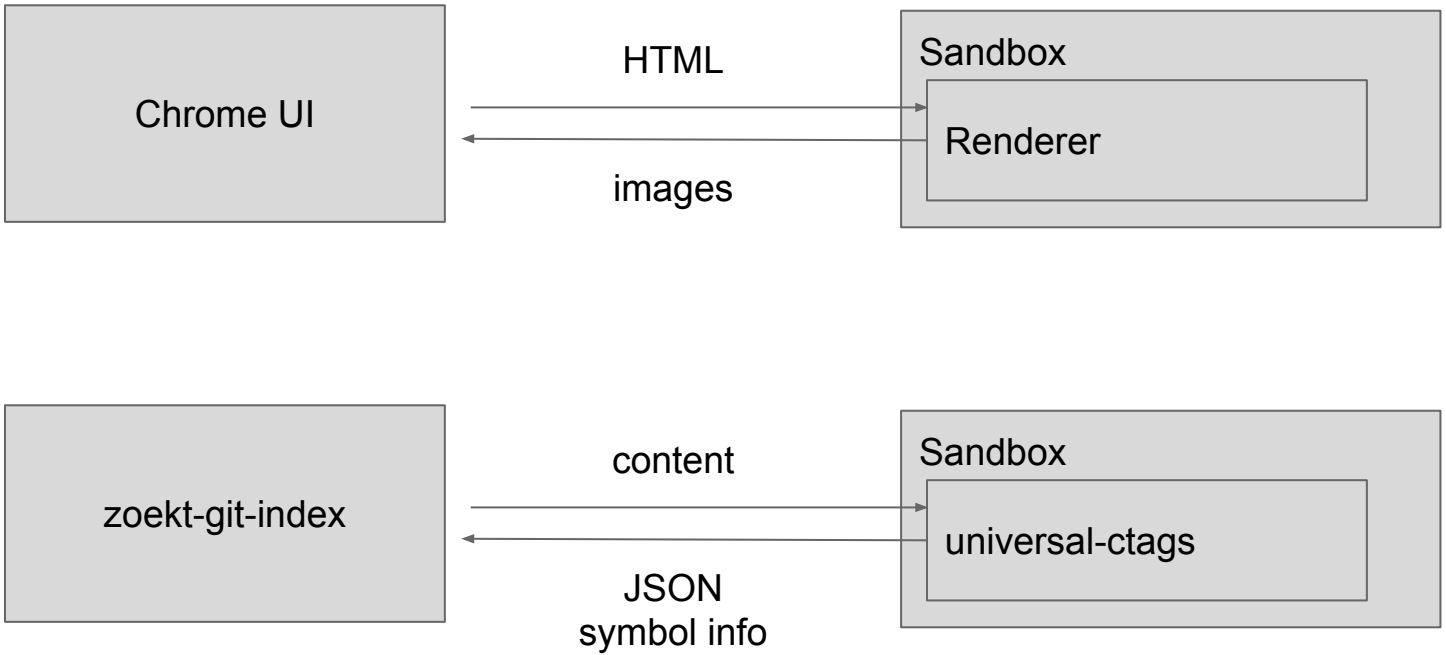
- Zoekt written in Go - safe language
- Symbol search: CTags
  - Identifier definitions are the best matches
  - Find definitions with Universal-CTags

Universal-ctags

- 100,000 lines of C
- Parse 87 programming languages
  - Common: Java, C++, Go
  - Obscure: Zephir, Cobol, Myrddin
- Parsing attacker controlled data is risky

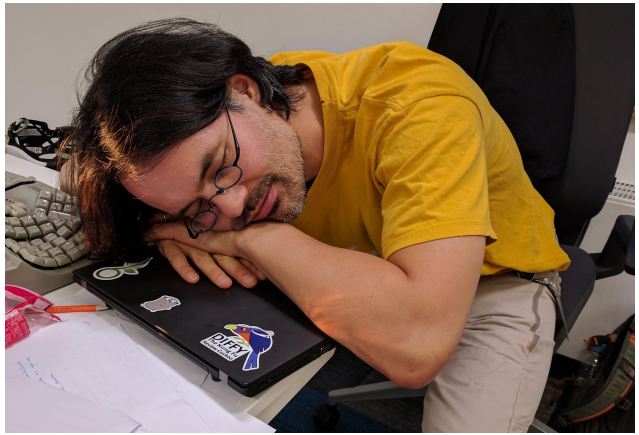
```
struct
Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxATDT 0900-12345
```

Making the best of insecure code



Sandboxing with seccomp-bpf

- Start program
- Declare allowed system calls:
  - Allocate memory
  - Read (stdin)
  - Write (stdout)
  - Exit
- call seccomp()
- illegal system call ➜ process killed

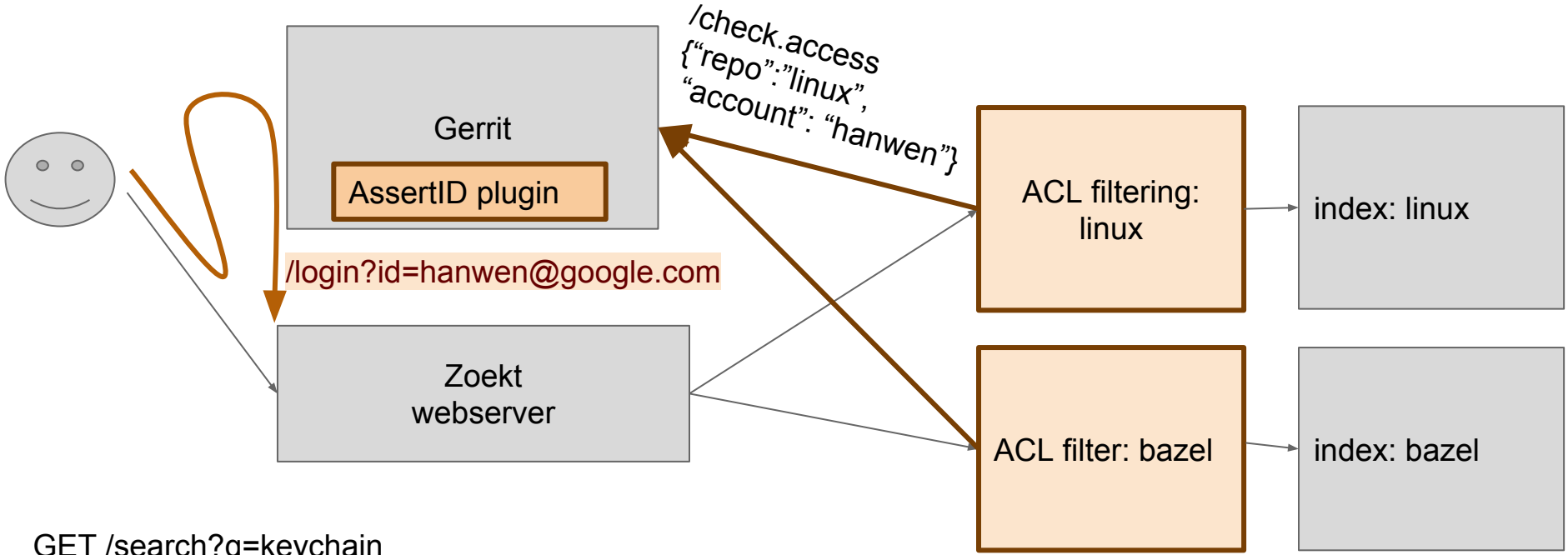


Closing thoughts

- Building search engines is fun!
- Productionized Zoekt
  - Support for UTF-8
  - Many bugfixes
  - Truly secure now
- Feedback is welcome

https://github.com/google/zoekt

Bonus: Gerrit ACL support



GET /search?q=keychain
Cookie: gerritID=abcdef

Bonus: ACL support

- Per-index access checks: seems OK
- Identity cookie dance:
  - Needs a plugin
  - Needs to synchronize with Gerrit login/logout
- Should integrate with SSO/LDAP/...
- **Help**: I don't run LDAP installation