Who am I?

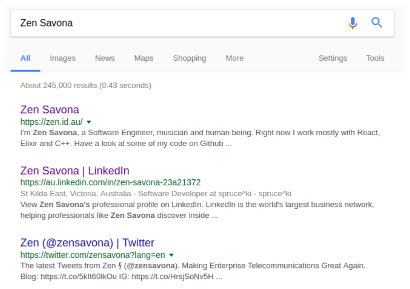
My name is Zen

I work for an enterprise telecommunications startup: spruce^ki

I think Elixir is pretty cool

@zensavona on Twitter

How do websites rank in Google? (simplified)

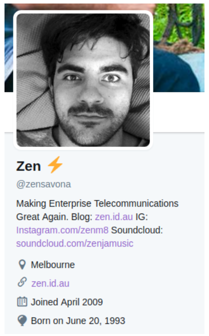


About 245,000 results (0.43 seconds)

Zen Savona
https://zen.id.au/
I'm **Zen Savona**, a Software Engineer, musician and human being. Right now I work mostly with React, Elixir and C++. Have a look at some of my code on Github ...

Zen Savona | LinkedIn
https://au.linkedin.com/in/zen-savona-23a21372
St Kilda East, Victoria, Australia - Software Developer at spruce^ki - spruce^ki
View **Zen Savona's** professional profile on LinkedIn. LinkedIn is the world's largest business network, helping professionals like **Zen Savona** discover inside ...

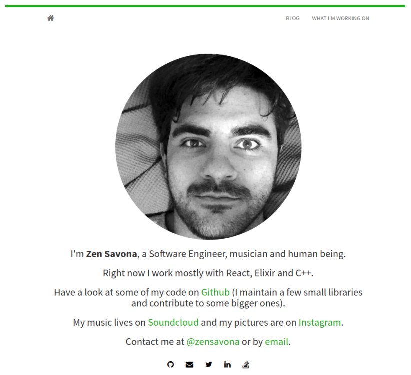
Zen (@zensavona) | Twitter
https://twitter.com/zensavona?lang=en
The latest Tweets from Zen ϟ (@**zensavona**). Making Enterprise Telecommunications Great Again. Blog: https://t.co/5kIt60lkOu IG: https://t.co/HrsjSoNv5H ...



Zen ⚡
@zensavona

Making Enterprise Telecommunications Great Again. Blog: zen.id.au IG: Instagram.com/zenm8 Soundcloud: soundcloud.com/zenjamusic

Melbourne
zen.id.au
Joined April 2009
Born on June 20, 1993



I'm **Zen Savona**, a Software Engineer, musician and human being.

Right now I work mostly with React, Elixir and C++.

Have a look at some of my code on Github (I maintain a few small libraries and contribute to some bigger ones).

My music lives on Soundcloud and my pictures are on Instagram.

Contact me at @zensavona or by email.

- Domains with more high quality links have more Link Juice™
- Link Juice™ flows from page to page
- More relevant link juice == higher ranking (sort of)
- Proprietary metrics to judge domains (TF, CF, DA, PA, etc)

The problem:

- There are lots of domains which expire and are full of Link Juice™.
- If we register these and link them to our site, we can hack ranking

but how can we find them?

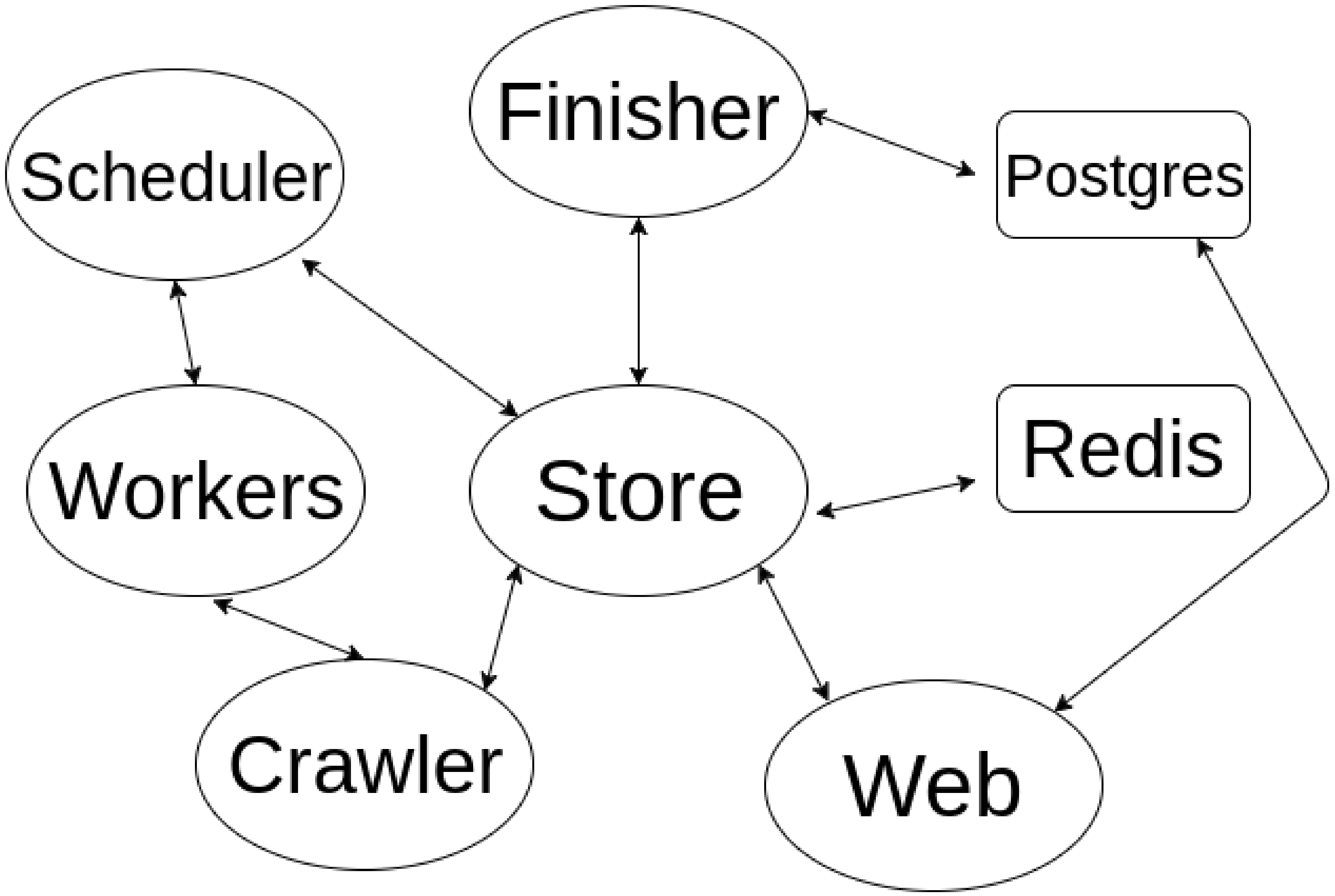
The solution:

- Find a big website with a lot of Link Juice™
- Crawl it
- Find outward linked domains, throw away registered ones
- Throw away ones with shit metrics
- ?????
- Profit

Overview of architecture

- **Store**: Get stuff in and out of Redis
- **Scraper**: Core scraping and domain checking functionality
- **Workers**: Do the actual work and scale concurrency
- **Scheduler**: Dish each crawl an equal slice of the resource pie
- **Finisher**: Finish things up and put them in Postgres
- **Web**: Show me what's going on

Granular MicroService Driven Architectual Design Specification

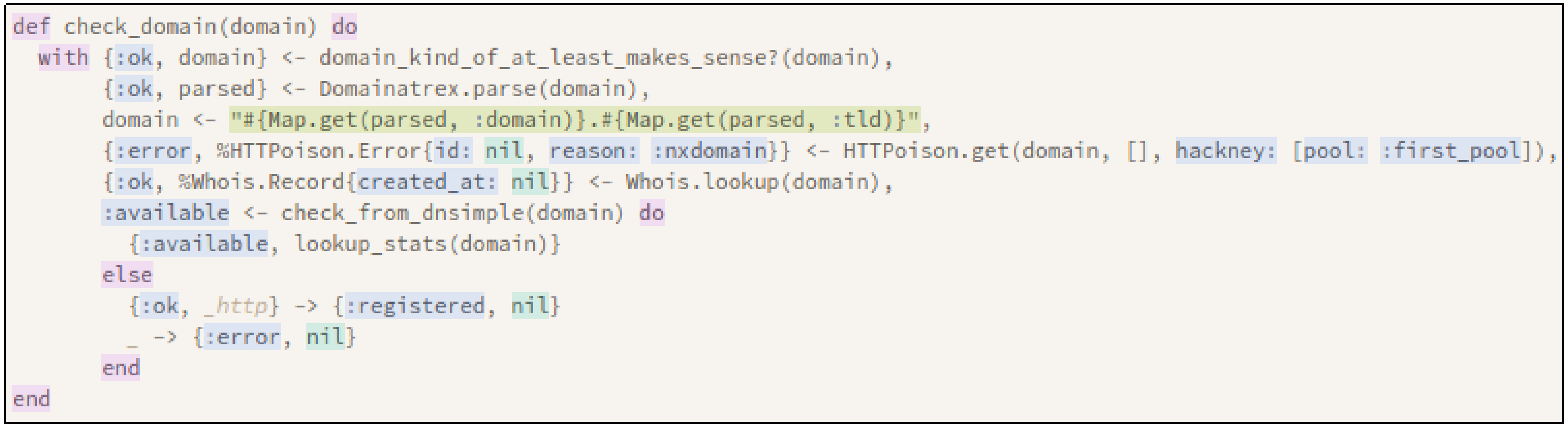


Store

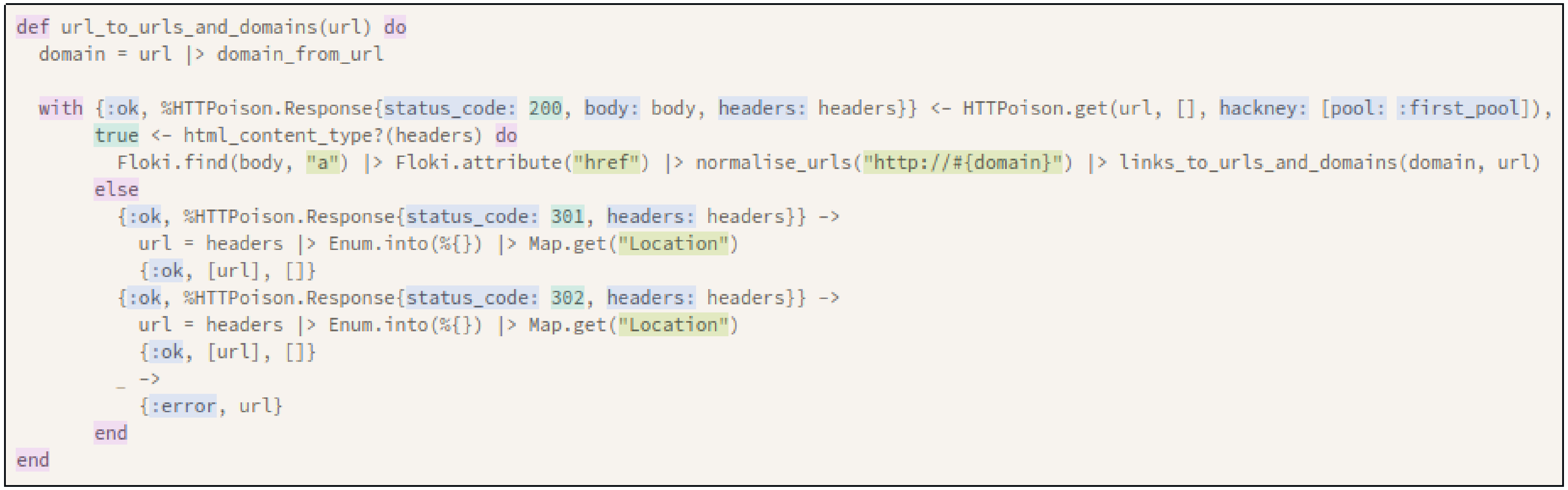
```
pool_size = 500
redix_workers = for i <- 0..(pool_size - 1) do
  worker(
   Redix,
   [
     [
       host: Application.get_env(:store, :redis_host),
       port: Application.get_env(:store, :redis_port)
     ],
     [name: :"redix_#{i}"]
   ],
   id: {Redix, i}
  )
end
```

```
def command(command) do
  Redix.command(:"redix_#{random_index()}",
                command, timeout: :infinity)
end
```

Scraper



```
def check_domain(domain) do
  with {:ok, domain} <- domain_kind_of_at_least_makes_sense?(domain),
       {:ok, parsed} <- Domainatrex.parse(domain),
       domain <- "#{Map.get(parsed, :domain)}.#{Map.get(parsed, :tld)}",
       {:error, %HTTPoison.Error{id: nil, reason: :nxdomain}} <- HTTPoison.get(domain, [], hackney: [pool: :first_pool]),
       {:ok, %Whois.Record{created_at: nil}} <- Whois.lookup(domain),
       :available <- check_from_dnsimple(domain) do
         {:available, lookup_stats(domain)}
       else
         {:ok, _http} -> {:registered, nil}
         _ -> {:error, nil}
       end
end
```

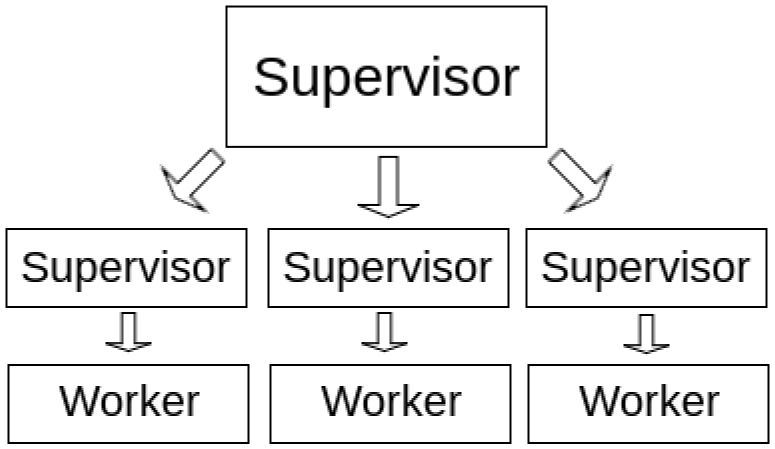


```
def url_to_urls_and_domains(url) do
  domain = url |> domain_from_url

  with {:ok, %HTTPoison.Response{status_code: 200, body: body, headers: headers}} <- HTTPoison.get(url, [], hackney: [pool: :first_pool]),
       true <- html_content_type?(headers) do
         Floki.find(body, "a") |> Floki.attribute("href") |> normalise_urls("http://#{domain}") |> links_to_urls_and_domains(domain, url)
       else
         {:ok, %HTTPoison.Response{status_code: 301, headers: headers}} ->
           url = headers |> Enum.into(%{}) |> Map.get("Location")
           {:ok, [url], []}
         {:ok, %HTTPoison.Response{status_code: 302, headers: headers}} ->
           url = headers |> Enum.into(%{}) |> Map.get("Location")
           {:ok, [url], []}
         _ ->
           {:error, url}
       end
end
```

Workers

- One for one supervision, with a parent supervisor
- Simple, but works just fine



- Workers are just tail recursive Tasks, which ask the Store for a new thing when they finish working on the current thing

```
worker(Task,
       [&Workers.Url.worker/0],
       [id: {Workers.Url, id},
       restart: :permanent]
)
```

What do these workers do?

```elixir
def worker do
  case Scheduler.pop_url do
    :empty ->
      :timer.sleep(1000)
    {crawl_id, url} ->
      case Scraper.Core.url_to_urls_and_domains(url) do
        {:error, url} ->
          Store.Crawled.push(crawl_id, url)
        {:ok, urls, domains} ->
          Store.Crawled.push(crawl_id, url)
          Store.ToCrawl.push(crawl_id, urls)
          domains
            |> Enum.each(
                 &(Store.Domains.push(crawl_id, &1))
               )
      end
  end
  worker()
end
```

Scheduler

Problems:

- If we pop a random url, bigger crawls would get more time
- If we pop the first/last, older crawls might never finish

Solution:

- Select a random crawl
- Select a random url from that crawl's `to_crawl` store

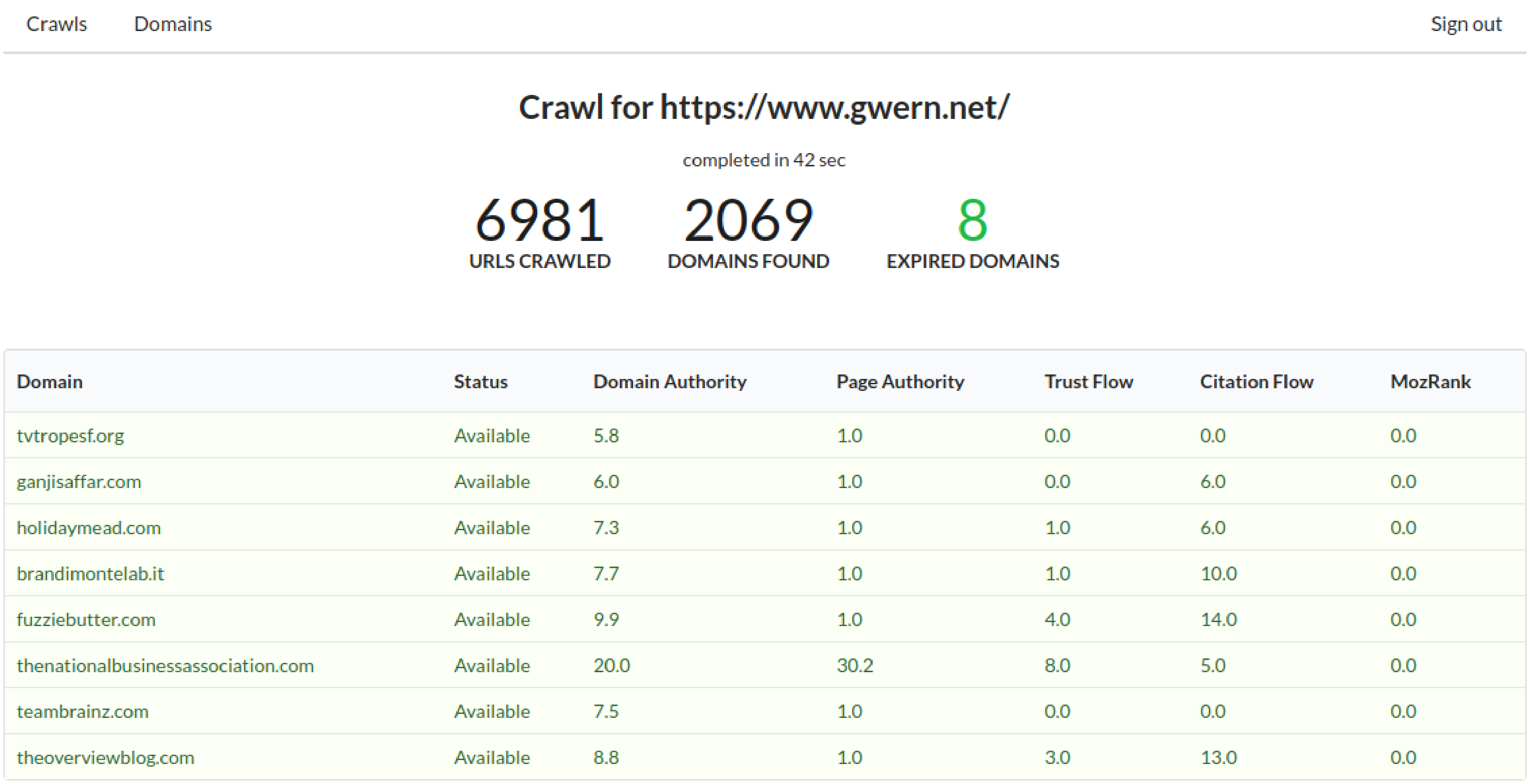
Finisher(s)

- How do we know when a crawl is finished?
- Domain metrics APIs cost $500/mo (each), too expensive
- There is a "pirated" domain metrics API
- This API is really crap, needs retries
- The Finishers are just a `:timer`. Simple, but works fine

```
:timer.apply_interval(5000, Finisher, :finish, [])
```

Web

- Just a Phoenix app which adds crawls and seed urls to Redis



Crawl for https://www.gwern.net/

completed in 42 sec

6981 URLS CRAWLED

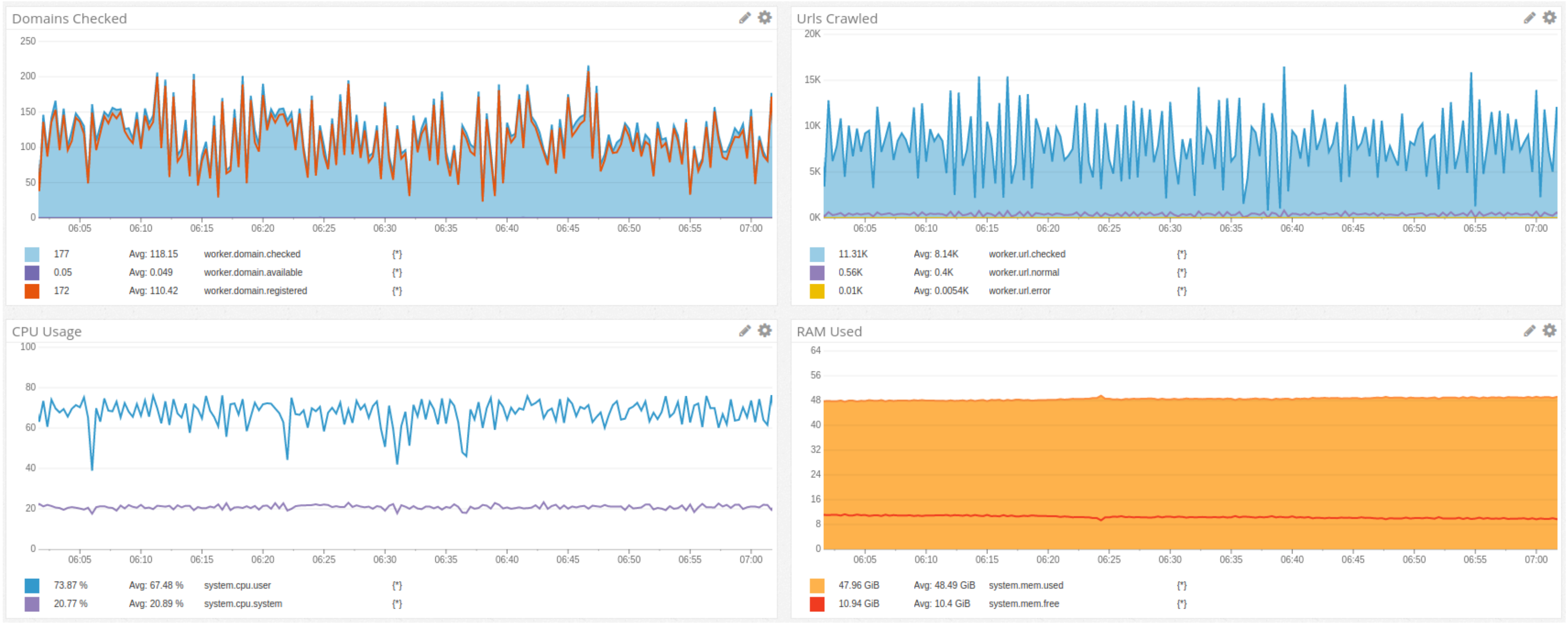
2069 DOMAINS FOUND

8 EXPIRED DOMAINS

Domain	Status	Domain Authority	Page Authority	Trust Flow	Citation Flow	MozRank
tvtropesf.org	Available	5.8	1.0	0.0	0.0	0.0
ganjisaffar.com	Available	6.0	1.0	0.0	6.0	0.0
holidaymead.com	Available	7.3	1.0	1.0	6.0	0.0
brandimontelab.it	Available	7.7	1.0	1.0	10.0	0.0
fuzziebutter.com	Available	9.9	1.0	4.0	14.0	0.0
thenationalbusinessassociation.com	Available	20.0	30.2	8.0	5.0	0.0
teambrainz.com	Available	7.5	1.0	0.0	0.0	0.0
theoverviewblog.com	Available	8.8	1.0	3.0	13.0	0.0

Instrumentation

- Datadog for graphs
- DogstatsD for metrics collection
- Looks pretty sweet



```
177       Avg: 118.15    worker.domain.checked       {*}
0.05      Avg: 0.049     worker.domain.available     {*}
172       Avg: 110.42    worker.domain.registered    {*}
```

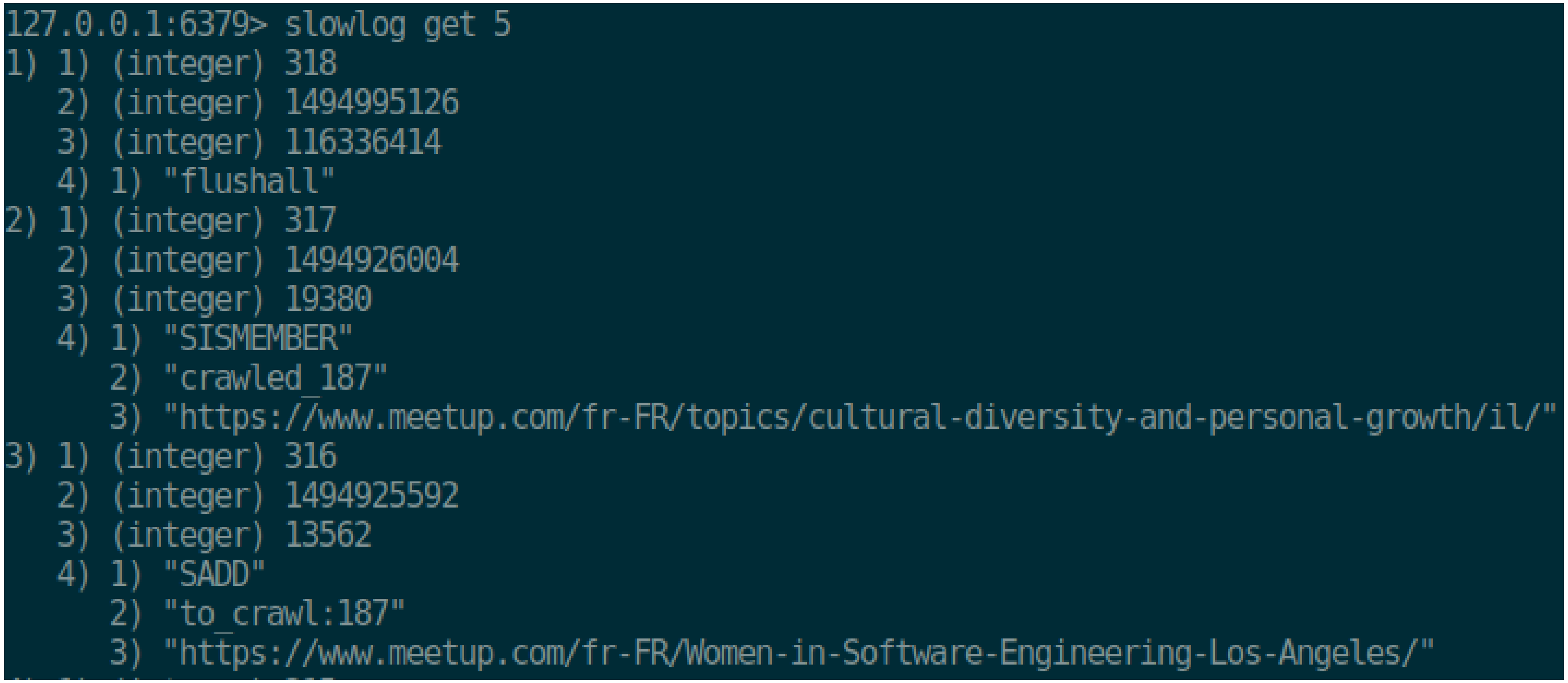
```
11.31K    Avg: 8.14K     worker.url.checked    {*}
0.56K     Avg: 0.4K      worker.url.normal     {*}
0.01K     Avg: 0.0054K   worker.url.error      {*}
```

```
73.87 %   Avg: 67.48 %   system.cpu.user     {*}
20.77 %   Avg: 20.89 %   system.cpu.system   {*}
```

```
47.96 GiB   Avg: 48.49 GiB   system.mem.used   {*}
10.94 GiB   Avg: 10.4 GiB    system.mem.free   {*}
```

Problems / Solutions

- Redis is single threaded
- Redis is not as fast as you'd think with 100m+ Sets
- RAM is expensive, SSD is slow(?)



```
127.0.0.1:6379> slowlog get 5
1) 1) (integer) 318
   2) (integer) 1494995126
   3) (integer) 116336414
   4) 1) "flushall"
2) 1) (integer) 317
   2) (integer) 1494926004
   3) (integer) 19380
   4) 1) "SISMEMBER"
      2) "crawled_187"
      3) "https://www.meetup.com/fr-FR/topics/cultural-diversity-and-personal-growth/il/"
3) 1) (integer) 316
   2) (integer) 1494925592
   3) (integer) 13562
   4) 1) "SADD"
      2) "to_crawl:187"
      3) "https://www.meetup.com/fr-FR/Women-in-Software-Engineering-Los-Angeles/"
```

Questions / Comments / Improvements