



The best reverse proxy around

Kristian Lyngstøl

Who we are

- **Redpill Linpro** is a Scandinavian free software company offering development services, operations, support, training and more.
- **Varnish Software** is a new spin-off company from Redpill Linpro that offer global support, training, migration and development on Varnish.
- **Our Varnish Customers** are located in Denmark, Norway, The USA, India, Brazil, Argentina, New Zealand, Netherlands and Sweden.

What Varnish is

- A reverse proxy
 - Also known as a “HTTP accelerator”
- A cache put in front of a potentially slow web server
- Because most of the time, a web page looks the same as the last time it was generated

Varnish at a glance

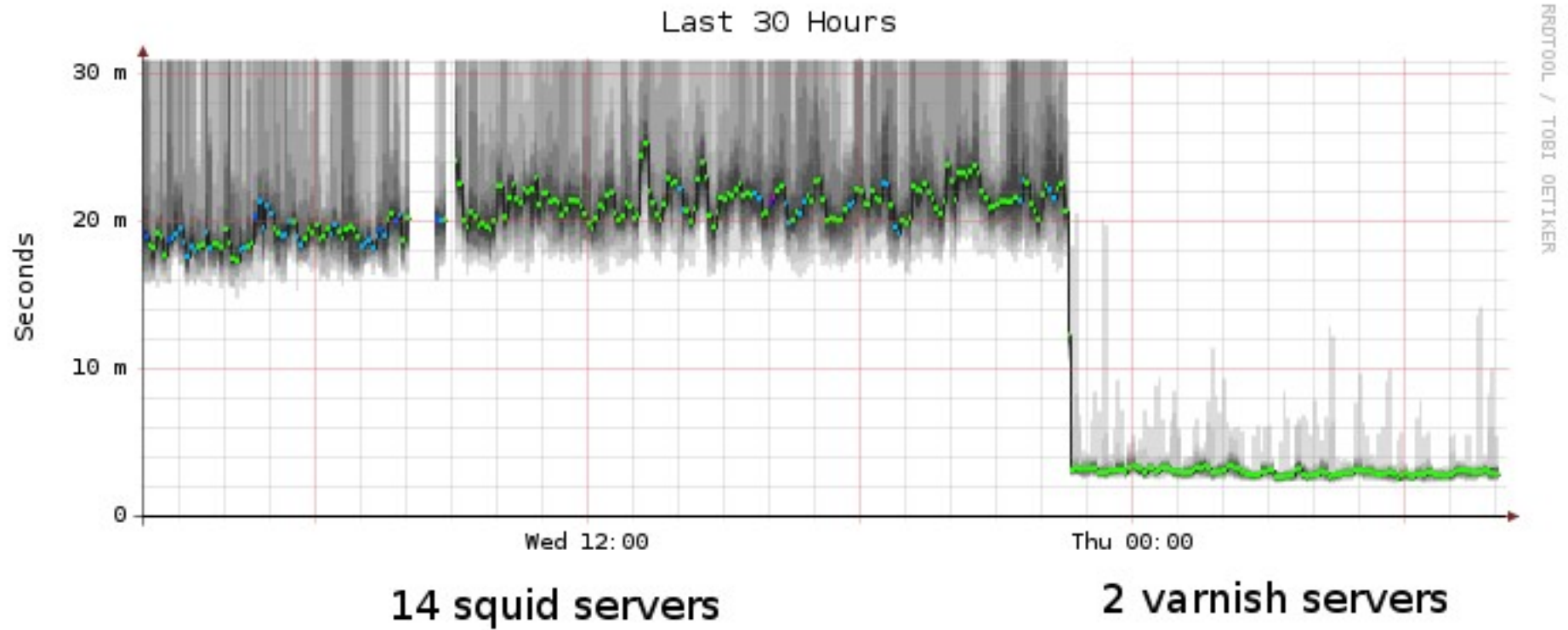
Key philosophies

- Modern design
- Flexibility
- Standardized

Result

- Speed
- Scalability
- Local Control
- Stability

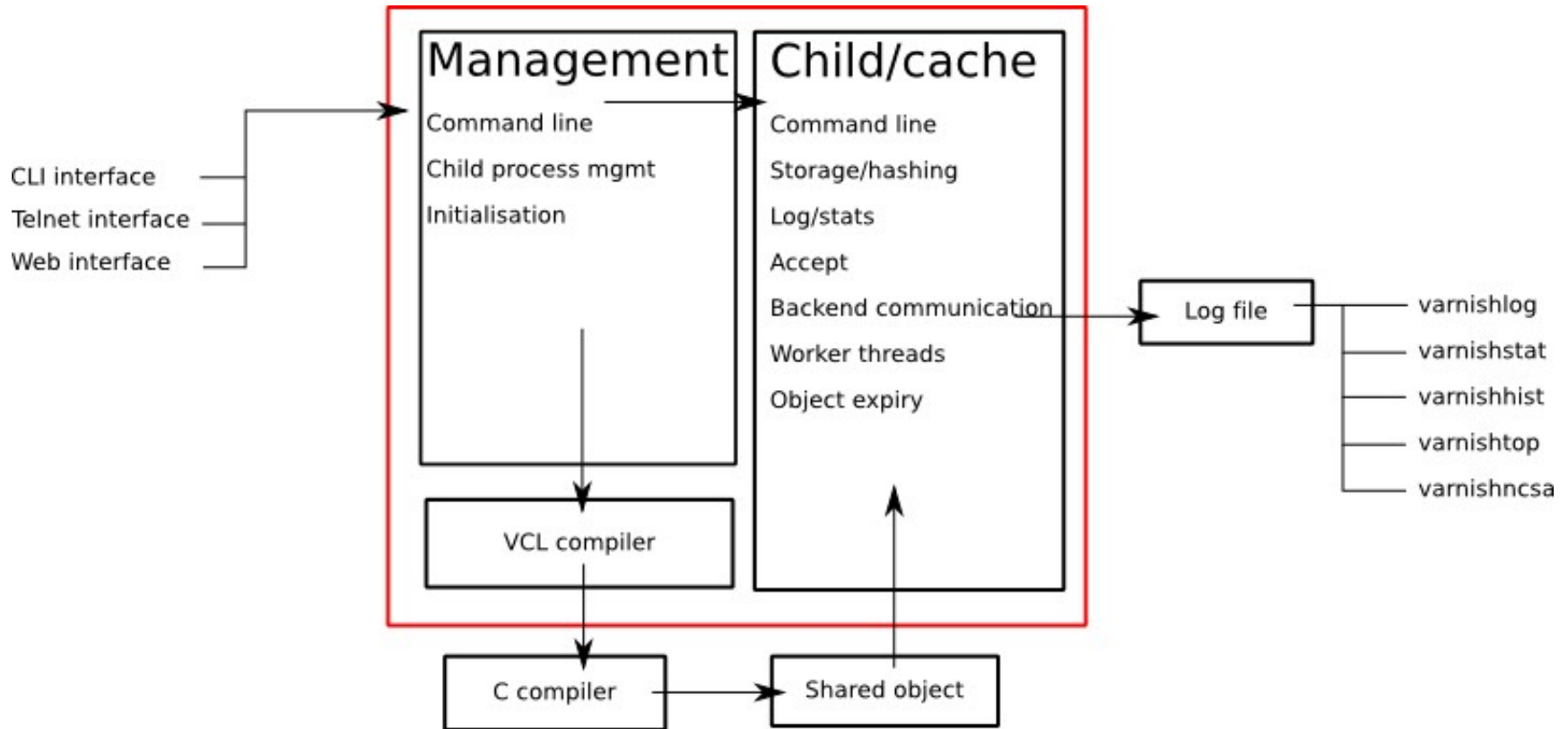
Speed



Architecture

- 64 bit - but support 32 bit
- Heavily multi-threaded
- Work with the OS
 - Let the OS figure out what to store to disk and what to keep in memory
 - `madvise(2) MADV_RANDOM` tells the OS how we expect to access data.
- Avoid memory operations if possible
- Let the VCL figure out the logic

Architecture



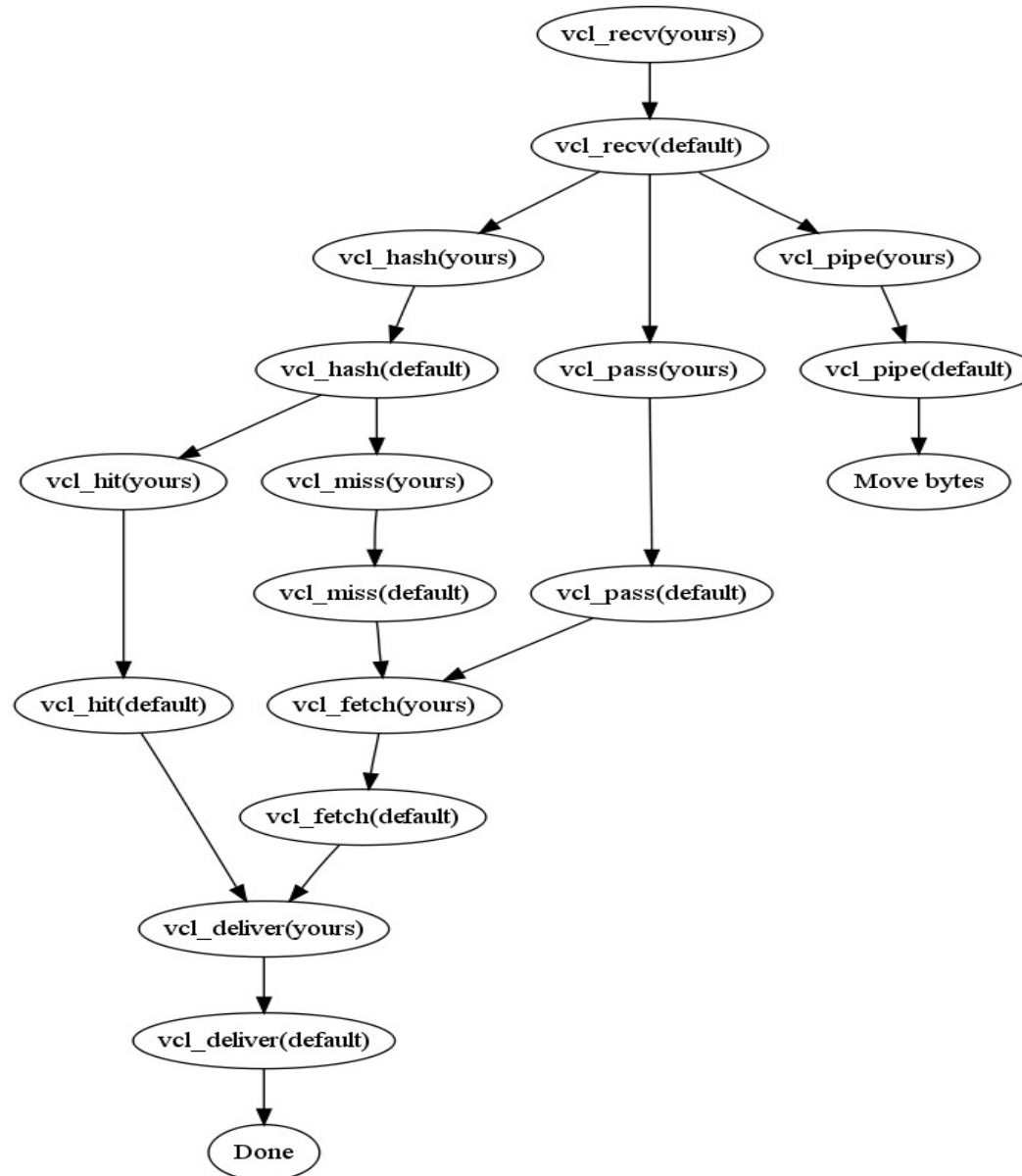
Varnish Configuration Language

- **VCL** offer a well-defined interface to the inner workings of Varnish, and allows the configuration syntax to be both simpler and more flexible.
- **VCL** is translated to C, compiled and linked directly into Varnish. It can be changed at run-time at the cost of switching 4 pointers around.
- **Varnish** allows the administrator to implement features by providing the mechanisms necessary to do so in VCL.

VCL Code

- No loops, no “proper” variables.
- recv, hash, pass, miss, hit, pipe, fetch, error and deliver
- In-line C for the brave of heart
- man vcl

VCL flow





«Demo»

Putting it to use

- Install it on a random machine
- Edit the default vcl-file in `/etc/varnish/default.vcl` to use your web server.
- Start it!
- Point your browser at it.
- Twiddle with the VCL until you're happy with it.
- Point your domain to your brand new varnish
 - Not recommended if the machine in the first step was your laptop.

Why doesn't it cache!

- Because you've got cookies

It still doesn't cache

- Does your backend send Cache-Control and/or Expires?
- Varnish honors - in prioritized order:
 - Cache-Control: s-maxage and max-age
 - Expires:
- Varnish does NOT honor:
 - Everything else, including:
 - Pragma:
 - Cache-Control: private, no-cache
- Unless you tell it to! Use VCL.

But.... it still doesn't cache

- Fire up varnishlog
- Send a request to Varnish
- Review the log
- Review the VCL
-
- PROFIT!

Tips and tricks - Varnishlog

- Use the -o, and one of -b or -c.
- VCL_Call tells you how your request traverse the VCL
- «varnishlog -o -c VCL_call pass»
- The TTL-tag reveals why the TTL was set
 - RFC - Either from a header or from default_ttl
 - VCL - Based on VCL logic
- Tx* is what Varnish transmits
- Rx* is what Varnish receives

Tips and tricks - Varnishtop

- «varnishtop -i TxURL»
- «varnishtop -i TxStatus»

Beyond the basics

- Smart purges
 - Purge on any combination of headers
 - Instantly added, no blocking
- ESI - Edge Side Include
- Grace
 - Avoid swamping backends and piling up threads when a popular page expires
 - Deliver expired objects instead of error messages
- Directors and health probes

Purge examples

```
if (req.request == "PURGE") {  
    purge("req.http.Cache-Channel ~" req.url);  
}
```

...

```
if (req.request == "PURGE" && client.ip ~ purgers) {  
    purge("req.http.Cache-Channel ~" req.url);  
}
```

...

```
$ varnishadm -T localhost:6082 purge req.url == /
```

- Use health probes
- ```
if (req.backend.healthy) {
 set req.grace = 30s;
} else {
 set req.grace = 30h;
}
```
- ```
set obj.grace = 30h;
```

Logging Varnish

- Varnish logs to a shared memory log, often kept on a ramfs
- Extensive information at virtually no cost
- Separate tools analyse the data
- Flexible
- Management information goes to syslog

The tools

- Varnishlog
- Varnishstat
- Varnishtop
- Varnishhist
- Varnishncsa

```

37+13:04:10
Hitrate ratio:      10      100      125
Hitrate avg:      0.9875  0.9858  0.9858

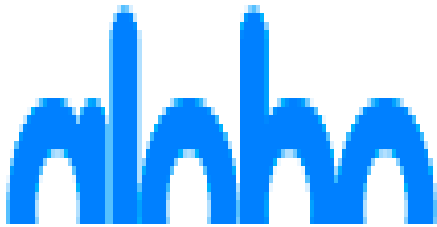
436566336          247.99          134.58 Client connections accepted
3738238196        2149.92          1152.41 Client requests received
3647901706        2113.92          1124.56 Cache hits
 2995698           2.00             0.92 Cache hits for pass
 71691301          25.00            22.10 Cache misses
 83240848          36.00            25.66 Backend connections success
  35457            0.00             0.01 Backend connections failures
 70714372          31.00            21.80 Backend connections reuses
 82628227          35.00            25.47 Backend connections recycles
   2475            .              .      N struct srcaddr
   1097            .              .      N active struct srcaddr

```

Who use Varnish?

wikia

Find and collaborate with people who love what you love.



 creative
commons



twitter™

funny OR DIE

