HAProxyConf 2021 Virtual

Using HAProxy Peers for Realtime Quota Tracking

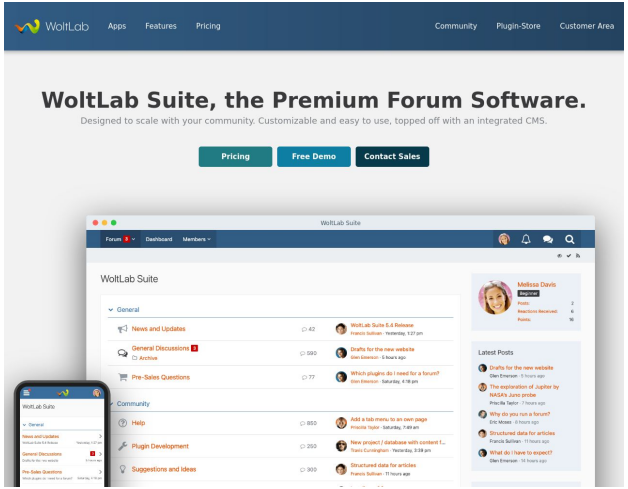
Tim Düsterhus

WoltLab GmbH
Cloud Architect

duesterhus@woltlab.com

WoltLab

- Software Suite for Online Communities
- We’re doing this for 20 years
- Available on-premises only for most of the time
- Very extensible and customizable
- New SaaS offering should not compromise on known features and behavior



The Problem

Creating Different Plans

- Per-seat billing does not work for us
  - Registration is open to the public
  - Team members are not clearly defined
- Reflect real costs instead
  - Dynamic Requests
  - Disk Storage

Tracking Disk Quota is Easy

- Does not rapidly change
- Is implicitly persisted on-disk
- ➜ Ask the file system in regular intervals

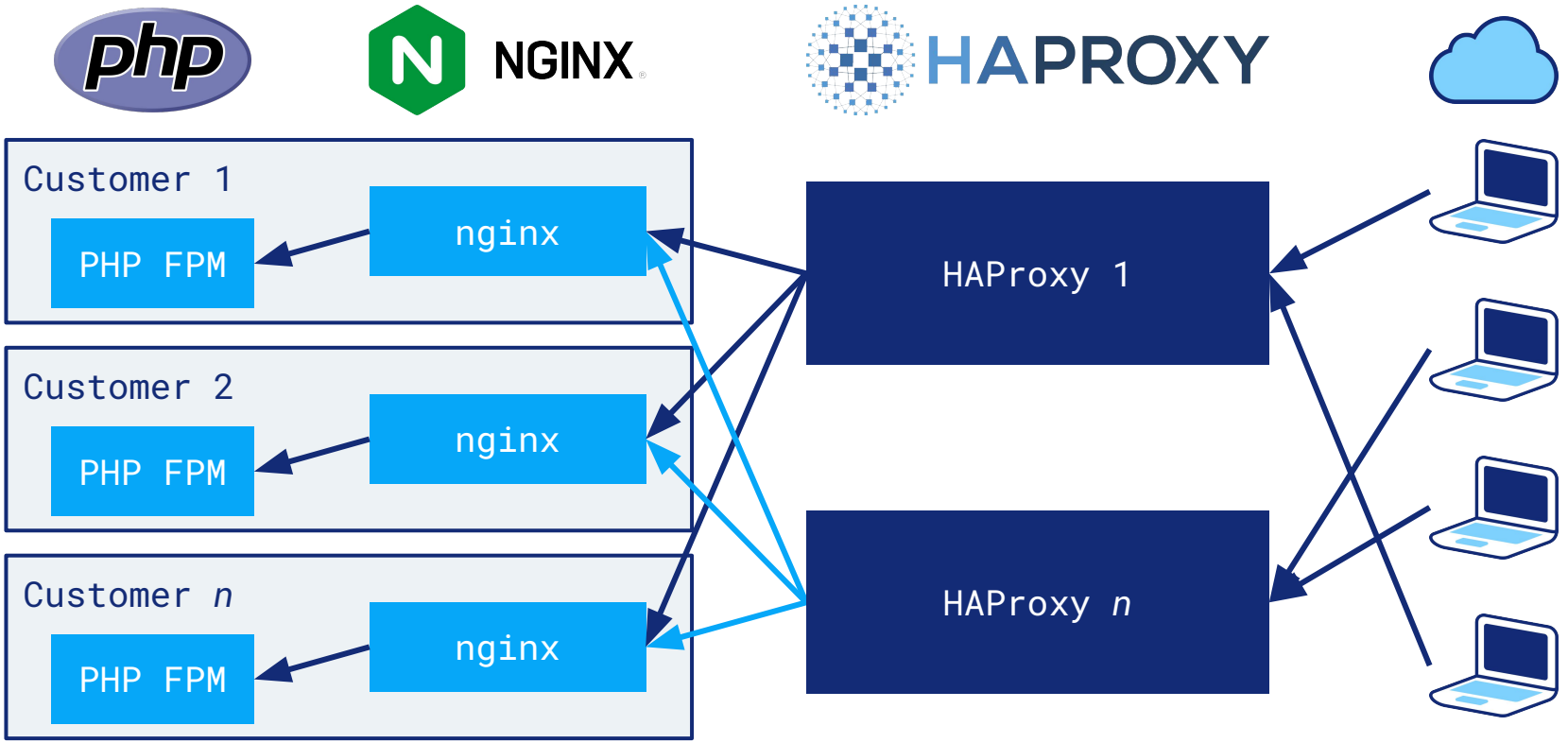
Tracking Requests is Not Easy

- Requests can come in rapidly
- Requests are ephemeral
- If not acted upon in-flight, the request is lost
- ➜ How do we do this?

Quota Tracking Requirements

- **Must not** affect regular request processing
  - Must not add a noticeable delay
  - Must not break processing if tracking fails (fail-open)
- **Must not** overcount
  - Must not charge customers for non-existent requests
- **Should not** undercount
  - Customers should be able to rely on accurate metrics
  - Tracker failures are on us

Architecture



Access Logs?

- Inefficient
  - Needs much disk storage
  - Needs to be parsed
- Fragile
  - Needs careful handling to maintain the offset
- Questionable with regard to privacy
  - PII (IP addresses) need to be deleted ASAP
  - Request tracking needs to be preserved for billing purposes
- ➜ Not a solution

PHP?

- Application is not trustworthy
  - Customer is able to extend and customize it
- Store counts in MySQL
  - Increased database load
- Ping request to a tracking service
  - Persistent connections are impossible in PHP
  - Non-blocking requests are non-trivial in PHP
- ➜ Not a solution

nginx?

- Open Source nginx is a black box
- ngx_http_auth_request_module to ping a tracking service
  - Blocks processing
- Custom Lua
  - Not explored
- ➜ Not a solution

HAProxy?

- Does not know the difference between static and dynamic requests
  - Application relies on nginx' `try_files`
- ➜ Not a solution

HAProxy?

- Share information using HTTP response headers
- Efficient dynamic in-memory storage using stick tables
- Can easily be extracted using the stats socket
- Unfortunately reloads lose state
  - Unless copied into the new worker using peers protocol
  - Wait … why just copy into a new worker?
- ➜ Our solution?

The Solution

nginx

```
location index.php {
    include /etc/nginx/php-common.conf;
    add_header X-WoltLab-Cloud-Dynamic yes always;
    set $upstream […];
    fastcgi_pass $upstream:9000;
}
```

HAProxy

```
backend bk_${customer_id}
    […]

    acl is_dynamic res.hdr(x-woltlab-cloud-dynamic) -m found

    http-response capture str(+PHP) id 0 if is_dynamic
    http-response capture str(-PHP) id 0 unless is_dynamic

    http-response del-header  X-WoltLab-Cloud-Dynamic

    […]
```

HAProxy (2)

```
peers tracker
    […]

    table quota type string len 8 size 10000 store gpc0
    bind :::20000 v4v6 ssl crt […] verify required

    server {{ inventory_hostname }}
    server tracker 127.0.0.1:20001
```

HAProxy (3)

```
backend bk_${customer_id}
    […]

    acl is_dynamic res.hdr(x-woltlab-cloud-dynamic) -m found

    http-response capture str(+PHP) id 0 if is_dynamic
    http-response capture str(-PHP) id 0 unless is_dynamic

    http-response track-sc0 str(${customer_id}) table tracker/quota if is_dynamic
    http-response sc-inc-gpc0(0)                                    if is_dynamic

    http-response del-header  X-WoltLab-Cloud-Dynamic

    […]
```

Quota Tracker

```
const socket = tls.connect(20000, domain, { key: …, cert: …, ca: … });
const connection = new PeerConnection(socket, {
    myName: "tracker",
    peerName: domain,
    direction: PeerDirection.OUT,
});
connection.on("entryUpdate", (update: EntryUpdate, definition: TableDefinition) =>
{
    this.handleUpdate(domain, update, definition);
});
```

Quota Tracker (2)

```
private handleUpdate(domain: string, update: EntryUpdate, definition:
TableDefinition): void {
    const key = (update.key as StringTableKey).key;
    const value = (update.values.get(DataType.GPC0) as
UnsignedInt32TableValue).value;

    […]
}
```

Our solution!

- Implementation was bumpy
  - Peers Protocol not documented well
  - Uncovered a few bugs in HAProxy
- Result is absolutely stable
  - Not a single change after it went into production
- ➜ Our solution to request tracking

Other Use Cases?

- Peers are very powerful
  - Realtime
  - Bidirectional
- For now just used for quota tracking
- Maybe you can think of a use case where this would help?

Open Source

- We rely on Open Source HAProxy
- We’re giving back our TypeScript implementation of HAProxy’s Peers protocol
- Not full-featured
  - Includes just the features we need
- Includes a simple demo
- https://github.com/WoltLab/node-haproxy-peers

Thank you

Tim Düsterhus

WoltLab GmbH
Cloud Architect

duesterhus@woltlab.com

Questions & Answers