[engine-x]

— Somewhat Advanced NGINX —

require('lx')

Carlos Rodrigues — October 2016

About NGINX*

- Asynchronous/event-based architecture

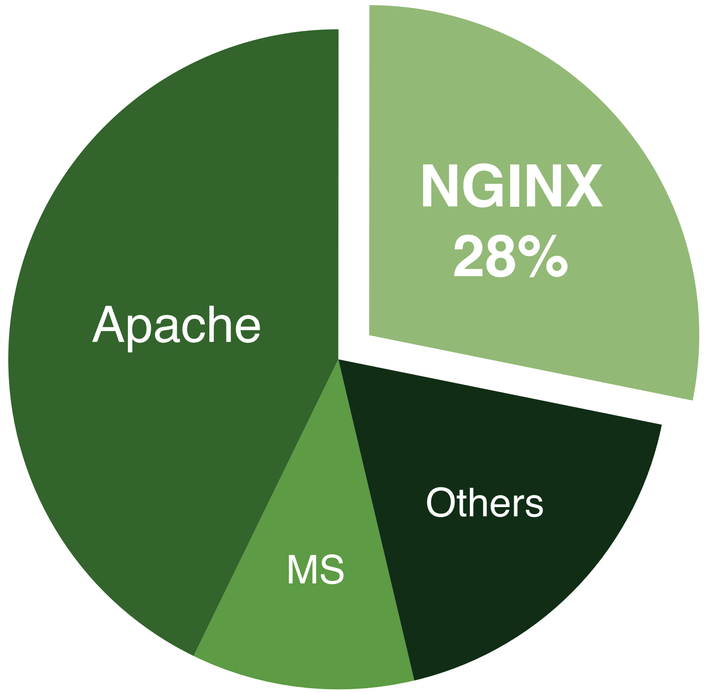
  …handles many thousands of **simultaneous** connections
  …handles **high loads** on modest hardware

- Open-source (BSD-licensed)

  …but NGINX, Inc. sells a beefed up version

- Runs on Linux, FreeBSD, etc.

  …also runs on Windows, but not yet production-ready



top million busiest websites
(Netcraft, September 2016)

About Branches

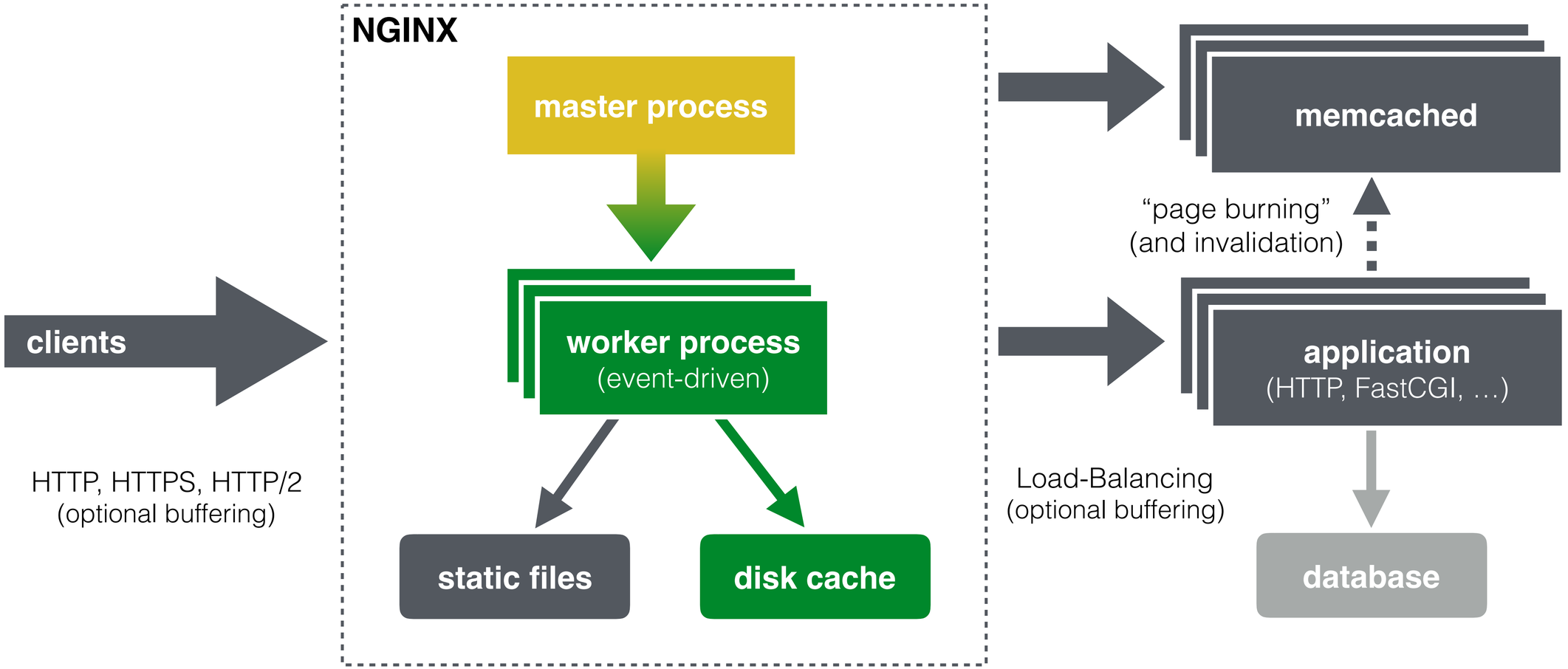
- Two **production-ready** branches

  …**mainline** gets new features and bugfixes
  …**stable** is only updated for security issues
  …there are no development releases

- Official packages for major Linux distros

  …updated quickly after each release
  …also available as Docker images

Example…


NGINX
master process
worker process
(event-driven)
static files
disk cache
clients
HTTP, HTTPS, HTTP/2
(optional buffering)
memcached
“page burning”
(and invalidation)
application
(HTTP, FastCGI, …)
Load-Balancing
(optional buffering)
database

Example…

```
http {
  upstream appserver_farm {
    server 10.0.0.1:8080;
    server 10.0.0.2:8080;
    least_conn;
  }

  server {
    listen *:80 default_server;
    server_name _;

    # [...]
  }
}
```

```
location / {
  default_type text/html;

  expires 5m;
  add_header X-Cache "HIT";

  set $memcached_key "$uri$is_args$args";
  memcached_pass 10.0.0.3:11211;
  error_page 404 502 504 = @backends;
}

location @backends {
  proxy_pass http://appserver_farm;
  proxy_redirect default;
}
```

Subtle Pitfalls

- **Hierarchical** Structure

  Nested blocks inherit (most) directives from their parents.

- **Declarative** Semantics

  Directives are **not** executed in the order they're specified.

```
http {
  server {  # virtual host
    location / {
      if ($request_method !~ ^(HEAD|GET)$ {
        return 403;
      }

      location /static/ {
      }
    }

    location ~ \.jpg$ {
    }
  }
}
```

Another Example…

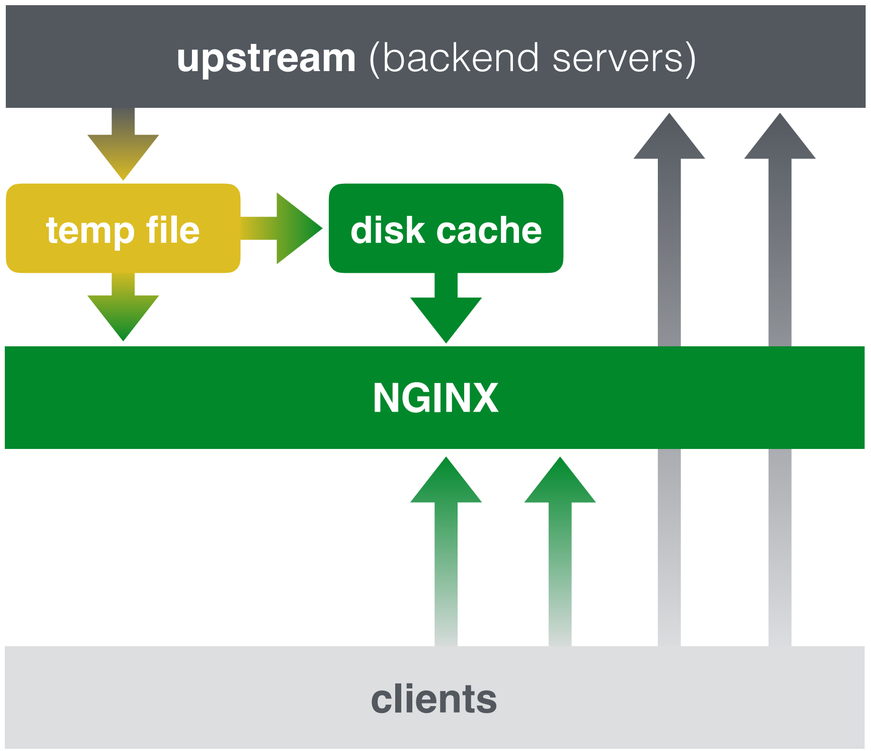
```
http {
  resolver 127.0.0.1 valid=30s ipv6=off;

  # [...]

  server {
    listen *:80 default_server;
    server_name _;

    location / {
      proxy_pass "http://example.com$uri$is_args$args";
      proxy_redirect ~^https?://(www\.)?example\.com(:\d+)?([/?]|$) $scheme://$host$3;
    }
  }
}
```

```
# Already on by default...
proxy_buffering on;
proxy_request_buffering on;
```

Adding Cache…

```
http {
  ...
  proxy_temp_path /var/cache/nginx/proxy_temp 1 2;  # default
  proxy_cache_path /var/cache/nginx/proxy_cache
                   levels=1:2 keys_zone=CACHE:16m
                   inactive=8h max_size=20g;

  server {
    ...
    location / {
      ...
      proxy_cache CACHE;
      add_header X-Cache $upstream_cache_status;  # debugging
      ...
    }
  }
}
```

Cache Fill Concurrency




```
http {
    ...
    proxy_cache_lock on;
    proxy_cache_lock_age 10s;
    proxy_cache_lock_timeout 5s;
    ...
}
```

Asynchronous I/O

- Disk I/O **isn't** asynchronous by default

  …and useful asynchronous I/O isn't actually available on all platforms.

- Workers **will block** on large/cold files

  …this includes both **static files** and files **cached** by any available method.

```
http {
    ...
    sendfile on;
    sendfile_max_chunk 512k;
    directio 8m;

    aio threads;

    tcp_nopush on;
    tcp_nodelay on;
    ...
}
```

The Future…

- **Lua*** simplifies complex scenarios

  …with a more imperative approach.

- **OpenResty**** is easier to get started

  …and bundles lots of extra modules.

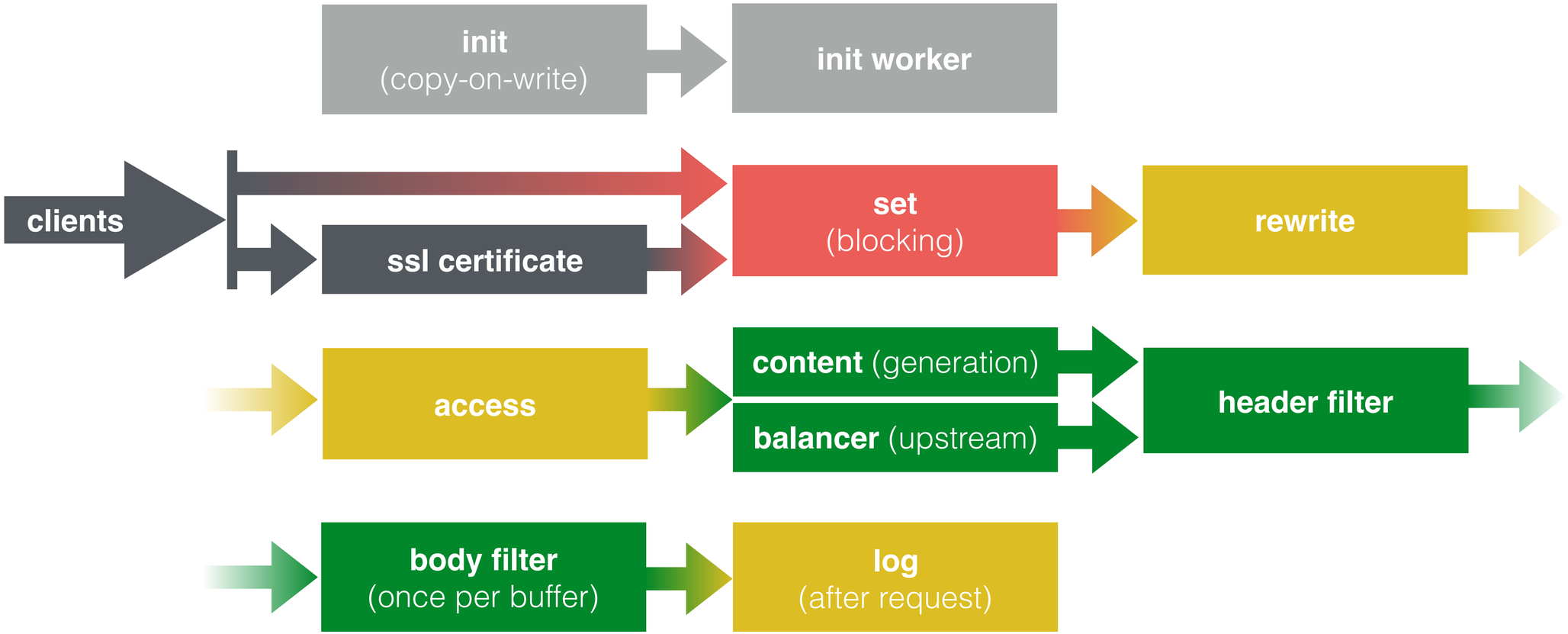
- **nginScript** may become an alternative

  …but is still experimental and not so flexible.

```
http {
  lua_shared_dict counters 1m;

  server {
    location / {
      access_by_lua_block {
        local c = ngx.shared.counters
        local ip = ngx.var.remote_addr

        c:add(ip, 0, 20)  -- 20s
        local hits = c:incr(ip, 1)

        if hits > 10 then
          return ngx.exit(429)
        end
      }
    ...
```


*github.com/openresty/lua-nginx-module // **openresty.org

The Future…



But this would be a whole new presentation…

Thanks!

Any questions?

Carlos Rodrigues
cer@brpx.com