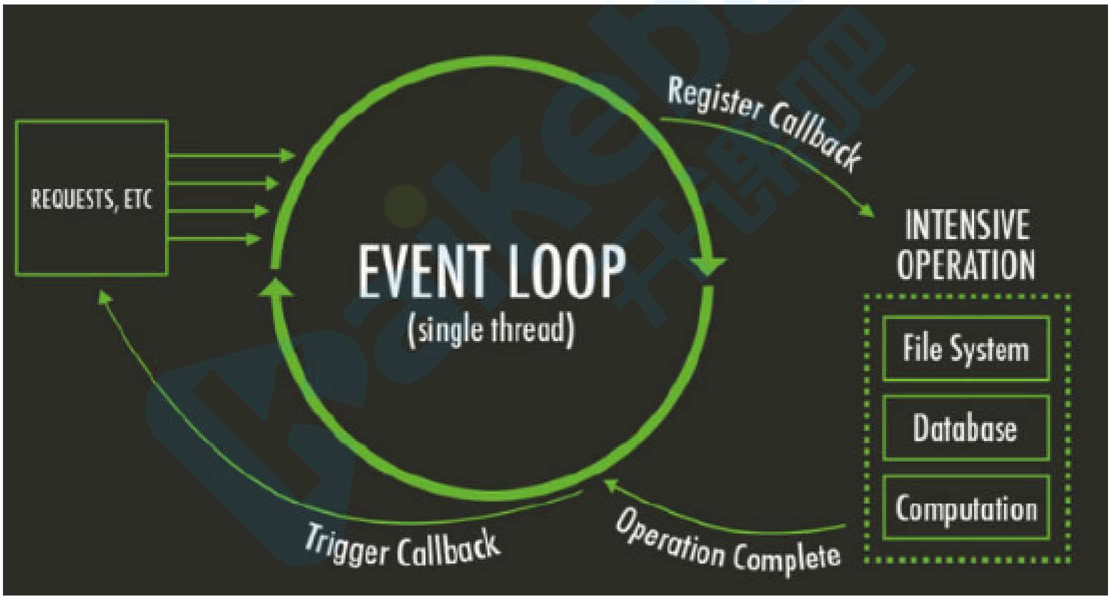
事件 EventLoop

> https://www.imooc.com/article/40020 一次搞懂Event loop
>
> setTimeout/setImmediate/process.nexTick的区别
>
> https://blog.csdn.net/hkh_1012/article/details/53453138

EventLoop是什么

> 一个循环 每次循环叫tick 每次循环的代码叫task

- V8引擎单线程无法同时干两件事
- 文件读取、网络IO缓慢且具有不确定性
- 要通过异步回调方式处理又称为异步IO
- 先同步再异步 异步放入队列等同步完成后在执行 每次循环叫一个tick (process.nextTick())



```
while (eventLoop.waitForTask()) {
  eventLoop.processNextTask()
}
```

异步任务的区分

microtasks(微任务):

> 唯一，整个事件循环当中，仅存在一个；执行为同步，同一个事件循环中的microtask会按队列顺序，串行执行完毕；

- process.nextTick
- promise

- Object.observe
- MutationObserver

tasks(宏任务):

- setTimeout
- setInterval
- setImmediate
- I/O
- UI渲染

  先执行微任务 再执行宏任务

> 最后我么思考一下

```
 // 等待下一下事件队列
(new Promise(resolve => {
    console.log('resolve')
    resolve()
}))
.then(() => console.log('promise then...'))

setImmediate(() => {
    console.log('set Immediate ...')
})
// setTimeout，放入Event Table中，1秒后将回调函数放入宏任务的Event Queue中
setTimeout(() => {
    console.log('setTimeout ...')
}, 0)

process.nextTick(() => {
    console.log('nextTick ...')
})
```

###

```
process.env.NODE_ENV || 'development',
```

// package.json

```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "SET NODE_ENV=production  node index.js"
  },
```

pm2多种模式解读

- fork模式

  单实例多进程，常用于多语言混编，比如php、python等，不支持端口复用，需要自己做应用的端口分配和负载均衡的子进程业务代码。 缺点就是单服务器实例容易由于异常会导致服务器实例崩溃。

- cluster模式

  多实例多进程，但是只支持node，端口可以复用，不需要额外的端口配置，0代码实现负载均衡。 优点就是由于多实例机制，可以保证服务器的容错性，就算出现异常也不会使多个服务器实例同时崩溃。

node高性能与nginx

https://www.cnblogs.com/hongcaomao/articles/nodejs-cluster.html

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}


#mail {
#   # See sample authentication script at:
#   # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#   # auth_http localhost/auth.php;
#   # pop3_capabilities "TOP" "USER";
#   # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#   server {
#       listen     localhost:110;
#       protocol   pop3;
#       proxy      on;
#   }
#
#   server {
#       listen     localhost:143;
#       protocol   imap;
#       proxy      on;
#   }
#}
```