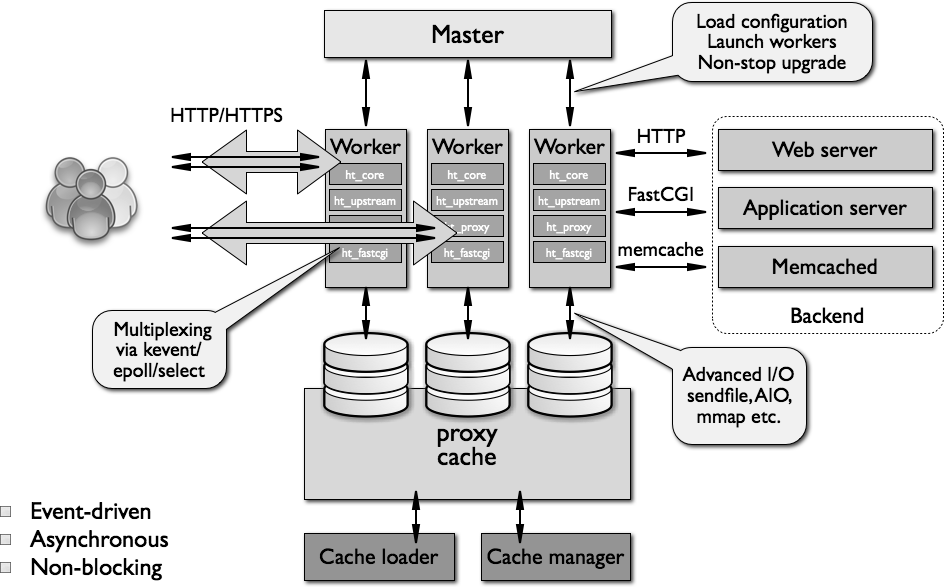
服务器的方方面面 – 如何实现服务器

1. 进程：master- worker模式



Master: 解析配置文件，创建worker进程，信号处理（热启动）等，

创建listen socket

Worker: 继承master环境变量，包括listen socket，

创建accept socket，处理请求。

1. 事件机制

**epoll**, aio, kqueue, poll, select

epoll\_create(…);

epoll\_ctl(…); // EPOLL\_CTL\_ADD, EPOLL\_CTL\_MOD, EPOLL\_CTL\_DEL

epoll\_wait(…);

1. 超时机制：红黑树

特点：插入、删除、查找

1. 读写操作

listen socket : read-event, LT模式(level trigger)

accept socket : read-event [, write-event], ET模式(edge trigger)

连接成功：触发写操作

连接失败：触发读操作

1. 异步无阻塞：永远不能阻塞

connect(…);

A, B, C

=>

connect(…, function()) {

A, B, C

}

var net = require('net');

var HOST = '127.0.0.1';

var PORT = 6969;

var client = new net.Socket();

client.connect(PORT, HOST, function() {

client.write('nonblock demo!');

});

6、资源

<http://www.aosabook.org/en/nginx.html>

<http://libevent.org/>

<http://nodejs.org/>