



链滴

# Node 学习记录之 cluster 模块

作者: [shealtiel](#)

原文链接: <https://ld246.com/article/1591309087205>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在如今机器的CPU都是多核的背景下，Node的单线程设计已经没法更充分的“压榨”机器性能了。所以从v0.8开始，Node新增了一个内置模块——“cluster”，顾名思义，它可以通过一个父进程管理一子进程的方式来实现集群的功能。

```
var cluster = require('cluster');
var http = require('http');
var numCPUs = require('os').cpus().length; // 获取CPU的个数

if (cluster.isMaster) {
  for (var i = 0; i < numCPUs; i++) {
    cluster.fork();
  }

  cluster.on('exit', function(worker, code, signal) {
    console.log('worker ' + worker.process.pid + ' died');
  });
} else {
  http.createServer(function(req, res) {
    res.writeHead(200);
    res.end("hello world\n");
  }).listen(8000);
}
```

通过isMaster属性，判断是否Master进程，是则fork子进程，否则启动一个server。每个HTTP server都能监听到同一个端口。但是在实际项目中，我们的启动代码一般都已经封装在了app.js中，要把整个启动逻辑嵌在上面的if else中实在不优雅。所以，我们可以这样：

```
var cluster = require('cluster');
var numCPUs = require('os').cpus().length;

if (cluster.isMaster) {
  for (var i = 0; i < numCPUs; i++) {
    cluster.fork();
  }
  // 其它代码
} else {
  require("./app.js");
}
```

简单之处就在于原本的应用逻辑根本不需要知道自己是在集群还是单边。（当然，如果应用在内存中护了某些状态，比如session，就需要运用某些机制来共享了，这里不详说）